



**UNIVERSITE ABDELMALEK ESSAADI**  
**FACULTÉ DES SCIENCES et TECHNIQUES**  
**TANGER**

**Centre d'Études Doctorales : « Sciences et Techniques de  
l'Ingénieur »**

**Formation Doctorale : « Sciences et Techniques de l'Ingénieur »**

**THÈSE DE DOCTORAT**  
**Présentée**

**Pour l'obtention du**

**DOCTORAT EN SCIENCES ET TECHNIQUES DE L'INGÉNIEUR**

**Par :**

***YAZIDI ALAOUI Otmane***

**Discipline : Système d'Information**

**Spécialité : SIG**

**Titre de la Thèse :**

**ALLIER LE TEMPS À L'ESPACE EN VUE D'UNE BONNE  
GOUVERNANCE DES RÉSEAUX TÉLÉCOMS**

**Soutenue le 3 juin 2016 devant le Jury**

***Président :***

***Pr ABDELHADI FENNAN. (FST-Tanger)***

***Rapporteurs :***

***Pr Omar EL KHARKI (ENCG-Agadir)***

***Pr Mustapha MAATOUK, (FST-Tanger)***

***Pr Mohammed Réda BRITEL, (ENSA-Tanger)***

***Directeur de thèse :***

***Pr Salaheddine HAMDOUNE (FST-Tanger)***

***Co-encadrant :***

***Pr Hassan ZILI, (FST-Tanger)***

***Invité :***

***M. Ahmed ZIANI, (Maroc Telecom-Division Réseau-Tanger)***

***Structure de recherche accréditée d'accueil :***

***Références : Laboratoire d'Informatique et Systèmes de Télécom de la FST de Tanger***

***Exemple : UAE/E14FST : Équipe de Recherche en Transferts Thermiques et Énergétique de la FST de  
Tanger***



**À mes Parents, qui ont cru en moi, et m'ont appris à ne jamais faillir.**

**À mes sœurs et frères, avec lesquels j'ai partagé les bons et les mauvais moments de la vie.**

## **Remerciements**

Je souhaite remercier les honorables membres du Jury qui ont accepté d'examiner avec intérêt ce travail de thèse.

J'adresse aussi mes profonds remerciements aux professeurs Salah Eddine Hamdoun et Hassane Zili :

Le premier pour son investissement humain et scientifique. Durant ces cinq années, il a su me diriger dans ce travail avec capacité et conviction, en m'indiquant les bonnes pistes scientifiques, mais aussi en me laissant une importante liberté d'initiative. Trouvez ici toute ma gratitude pour m'avoir transmis l'amour et la patience que la recherche implique.

Le second qui tout au long de mes études m'a accordé sa confiance et m'a permis d'entreprendre mes études doctorales à la FST de Tanger en me proposant ce sujet de thèse passionnant.

Toute ma gratitude à mon ami, le professeur Michel Leikine pour sa contribution attentive sans laquelle cette thèse n'aurait pas vu le jour.

Merci aussi à toute l'équipe du Laboratoire d'Informatique, Système et Télécommunication.

Merci aux ingénieurs et techniciens de Maroc Télécom pour les conseils judicieux qu'ils m'ont accordés.

La thèse est une aventure de plusieurs années pleines de joies, mais aussi de difficultés et de doutes qu'il est difficile de surmonter seul. Je ne peux donc pas oublier le soutien moral infatigable que m'a accordé la doctorante Hajar Talbi ; mais aussi mes ami(e)s de tous les instants Fatiha, Otmane dont le soutien est inestimable.



# SOMMAIRE

Liste des figures .....	I
Liste des Abréviations.....	II
Résumé.....	III
Abstract.....	IV
INTRODUCTION.....	1
Prémises .....	1
Objectifs de la thèse .....	2
Contexte de la thèse .....	3
Organisation du mémoire.....	5
<b>PARTIE I:ÉTAT DE L'ART .....</b>	<b>7</b>
<b>Chapitre 1 : Systèmes d'Aide à la Décision .....</b>	<b>9</b>
1.1 Outils d'extraction, ETL (ExtractTransformLoad) .....	11
1.2 Entrepôts de données et magasins de données .....	12
1.2.1 Définition de l'entrepôt de données .....	12
1.2.2 Définition du magasin de données .....	13
1.2.3 Modèle multidimensionnel logique .....	13
1.3 Serveur OLAP.....	19
1.3.1 Principe de base.....	19
1.3.2 Fonctionnalité d'un serveur OLAP .....	21
1.3.3 Implémentation physique d'un serveur OLAP.....	23
1.4 Client OLAP .....	23
1.5 Conclusions .....	24
<b>Chapitre 2 : Système d'Information Géographique (SIG) et Géomatique .....</b>	<b>25</b>
2.1 L'espace géographique.....	25
2.2 Les données géographiques.....	26
2.2.1 Composante spatiale.....	26
2.2.2 Composante sémantique.....	26
2.2.3 Échelle spatiale.....	26
2.2.4 Analyse spatiale.....	27
2.2.5 Les relations temporelles.....	29
2.3 Fonctionnalités des Systèmes d'Information Géographique .....	31
2.4 Conclusions .....	33
<b>Chapitre 3 : le concept SOLAP ou OLAP spatial.....</b>	<b>35</b>
3.1 Origines des systèmes d'aide à la décision spatiale .....	36
3.1.1 Définitions et caractéristiques de la technologie SOLAP .....	36
3.2 Principaux concepts du SOLAP. Modélisation logique .....	38
3.2.1 Dimensions spatiales .....	39
3.2.2 Mesures spatiales.....	41

3.2.3 Niveau d'agrégation spatial .....	43
3.2.4 Opérateur SOLAP.....	44
3.2.5 Agrégation dans le SOLAP .....	45
3.2.6 Modélisation conceptuelle des données spatiales.....	46
3.3. Typologie des approches pour le développement d'applications SOLAP .....	48
3.3.1 L'OLAP dominant.....	48
3.3.2 Le « SIG dominant » .....	50
3.3.3 La solution « Hybride ou intégrée » .....	50
<b>Chapitre 4 : La technologie Open Source .....</b>	<b>53</b>
4.1. Définition open source .....	53
4.1.1 Libre redistribution .....	53
4.1.2 Code source .....	53
4.1.3 Travaux dérivés .....	53
4.1.4 Intégrité du code source de l'auteur.....	54
4.1.5 Pas de discrimination entre les personnes ou les groupes.....	54
4.1.6 Pas de discrimination entre les domaines d'application .....	54
4.1.7 Distribution de la licence .....	54
4.1.8 La licence ne doit pas être spécifique à un produit.....	54
4.1.9 La licence ne doit pas contaminer d'autres logiciels .....	55
4.1.10 Exemples de licences .....	55
4.2 Justification de choix de l'Open Source.....	55
4.3 Informatique décisionnelle Open Source .....	56
4.3.1 Pourquoi, les entreprises choisissent-elles, BI Open Source .....	56
4.3.2 Justification de choix des outils open BI .....	57
4.3.3 Absence ou droits de licence réduits.....	57
4.3.4 Mises à jour fréquentes .....	57
4.3.5 Le logiciel peut être personnalisé .....	57
4.4. Inconvénient de BI Open Source .....	58
4.4.1 Cette solution exige des développeurs hautement qualifiés. ....	58
4.4.2 Peut être coûteux .....	58
4.4.3 L'interface utilisateur n'est pas attrayante .....	58
<b>Chapitre 5 : Principe de la téléphonie mobile.....</b>	<b>59</b>
5.1 Structure générique d'un réseau de téléphonie mobile .....	59
5.1.1 Réseau d'accès .....	59
5.1.2 Réseau cœur (Figure 23).....	61
5.2.1 Le NSS.....	63
5.2.2 OMC (Operations and Maintenance Center) et NMC sous-systèmes d'exploitation et de maintenance.....	64
5.3 Les entités fonctionnelles du réseau GSM .....	64
5.3.1 La station mobile (MS).....	65
5.3.2 Le sous-système radio (BSS, Base Station Subsystem).....	65

<b>PARTIE II :DÉVELOPPEMENT D'UN PROTOTYPE .....</b>	<b>67</b>
<b>Chapitre 6 : Conception de l'Entrepôt de données spatial « GouvData ».....</b>	<b>71</b>
6.1 Outils de modélisation et de développement.....	71
6.1.1 UML.....	71
6.1.2 PostgreSQL.....	71
6.1.3 Talend Open Studio.....	72
6.1.4 Pantaho.....	72
6.2 Structure de l'Entrepôt de Donnée proposée.....	73
6.2.1 Magasin Temps.....	74
6.2.2 Magasin Technique.....	75
6.2.3 Magasin planification urbaine.....	76
6.2.4 Magasin Découpage Administratif.....	77
6.3 Création de la structure vide.....	78
6.4 Validation de l'EDS avec Pantaho.....	79
<b>Chapitre 7 : Conception de l'outil Spatial OLAP « GouvRes » .....</b>	<b>85</b>
7.1 Outils.....	85
7.1.1 QTCreator.....	85
7.1.2 Cubes.....	85
7.1.3 SQLAlchemy.....	86
7.1.4 Matplotlib.....	86
7.2 Création de la plateforme SIG.....	87
7.2.1 Téléchargement et adaptation du logiciel Qgis.....	87
7.3 Création d'un plug-in « Moteur ».....	89
7.3.1 Création d'un fichier d'extension « *. Ui ».....	90
7.3.2 Fichier d'extension « *. Py ».....	91
7.3.3 Fichier d'extension « *. Json ».....	92
7.4 Intégration du plug-in de couplage sous Qgis modifié.....	94
7.5 Connexion à l'entrepôt de données spatial « GouvData ».....	95
7.6 Conclusion.....	99
<b>Chapitre 8 : fonctionnements du Prototype Géodécisionnel et Résultats.....</b>	<b>101</b>
8.1 requête relative au choix d'un site pour l'installation d'un émetteur –récepteur BTS.....	101
8.1.1 Description de la page principale.....	102
8.1.2 Fenêtre de Planification urbaine.....	105
8.2 Requête de nature statistique.....	115
8.2.1 Fenêtre des statistiques.....	115
8.2.2 Résultats.....	117
8.3 Fenêtre MXD.....	119
8.3.1 Rédaction des Requêtes Personnalisées.....	120
8.4 Affichage des résultats sur différentes plateformes.....	127
8.5 Conclusion.....	129
<b>PARTIE III :CONCLUSION .....</b>	<b>131</b>
<b>Chapitre 9 : conclusion générale et perspective .....</b>	<b>133</b>
9.1 Discussion.....	133
9.1.1 Open source.....	133

9.1.2 Conception de l'entrepôt de données Spatial « GouvData » .....	134
9.1.3 Développement de « GouvRes » .....	134
9.1.4 Résultats .....	135
9.1.5 Évolution du Prototype .....	137
9.2 Perspectives .....	137
<b>ANNEXES .....</b>	<b>139</b>
<b>Annexe 1 : Création de l'Entrepôt de Données Spatial sous PostgreSQL .....</b>	<b>141</b>
<b>Annexe 2 : Ajout des données à l'entrepôt de données Spatial.....</b>	<b>143</b>
<b>Annexe 3 : Installation des Outils SqlAlchemy : .....</b>	<b>146</b>
<b>Annexe 4 : Création du cubes de données .....</b>	<b>148</b>
<b>Annexe 5 : Compilation du code source QGIS.....</b>	<b>154</b>
<b>Annexe 6 : les Opérateurs de cube de donnée .....</b>	<b>186</b>
<b>Annexe 7 : Création du Plugin .....</b>	<b>192</b>
<b>Références.....</b>	<b>215</b>

# Liste des figures

<i>Figure 1 : Architecture typique d'un OLAP. Adaptée de (Malinowski et Zimányi 2008).</i> .....	10
<i>Figure 2 : Dimensions et leur hiérarchie respective. Source : (Dinimant 2009)</i> .....	14
<i>Figure 3 : Niveaux et membres de la dimension temporelle</i> .....	15
<i>Figure 4 : Représentation multidimensionnelle suivant la métaphore du cube lequel représente trois perspectives d'analyse. Source : (Meier 2006).</i> .....	16
<i>Figure 5 : Structure en étoile. Source : (Bédard 2008)</i> .....	17
<i>Figure 6 : Structure en flocon. Source : (Bédard 2008)</i> .....	18
<i>Figure 7 : Structure mixte. Source : (Bédard 2008)</i> .....	18
<i>Figure 8 : Structure en constellation. Source : (Bédard 2008)</i> .....	19
<i>Figure 9 : Opérateurs OLAP, roll-up drill-down. Source : (Bouadi 2014)</i> .....	21
<i>Figure 10 : Opérateur OLAP Slice. Source : (Bouadi 2014)</i> .....	22
<i>Figure 11 : Opérateur OLAP Dice Source : (Bouadi 2014)</i> .....	22
<i>Figure 12 : Topologie : exemple de situations équivalentes du point de vue topologique</i> .....	28
<i>Figure 13 : Adjacence</i> .....	28
<i>Figure 14 : Relation entre intervalles temporels. Source : (Allen 1983)</i> .....	30
<i>Figure 15 : Système de coordonnées géographiques pour localiser une entité.</i> .....	31
<i>Figure 16 : Organisation des informations géographiques en couches thématiques. Source : (Bimonte 2007)</i> .....	33
<i>Figure 17 : Exemple de modèle logique spatiomultidimensionnel. Source : (Boulil 2014)</i> .....	38
<i>Figure 18 : Les trois modes de représentation des dimensions spatiales supportées par le SOLAP.</i> ...	40
<i>Figure 19 : Schéma de l'entrepôt « Météo ». Source : (Stefanovic et al. 2000)</i> .....	42
<i>Figure 20 : Modèle spatiomultidimensionnel avec des dimensions alphanumériques et une mesure spatiale. Source : (Malinowski et Zimányi 2004)</i> .....	43
<i>Figure 21 : Structure générique d'un réseau de téléphonie mobile. Source : (<a href="http://www.efort.com">http://www.efort.com</a>)</i> .	59
<i>Figure 22 : Schéma des réseaux d'accès 2 G, 3 G, 4 G de la téléphonie mobile. Source : (<a href="http://www.efort.com">http://www.efort.com</a>)</i> .....	61
<i>Figure 23 : Schéma des différents « réseaux cœur » de la téléphonie mobile. Source : (<a href="http://www.efort.com">http://www.efort.com</a>)</i> .....	62
<i>Figure 24 : Modèle en flocon de l'Entrepôt de Données Spatial</i> .....	74

<i>Figure 25 : Magasin dimension temps</i> .....	75
<i>Figure 26 : Magasin dimension techniques</i> .....	76
<i>Figure 27 : Magasin planification urbaine</i> .....	77
<i>Figure 28 : Magasin découpage administratif</i> .....	77
<i>Figure 29 : Structure vide de l'entrepôt de données spatial sous PostgreSQL</i> .....	78
<i>Figure 30 : Cube de données sous Workbench</i> .....	79
<i>Figure 31 : Onglet préférence sous Workbench</i> .....	80
<i>Figure 32 : Affichage des données du trafic mensuel de quelques BTS de la région de Rabat, pour l'année 2013</i> .....	81
<i>Figure 33 : Projection sur une image satellitaire, du trafic de quelques BTS</i> .....	81
<i>Figure 34 : Projection des données sur un fond cartographique avec affichage des informations relatives à chaque BTS</i> .....	82
<i>Figure 35 : Différents types de graphes obtenus à l'aide de Pantaho (c)</i> .....	83
<i>Figure 36 : Exemple de graphes disponibles dans Matplotlib</i> .....	86
<i>Figure 37 : Vue partielle du code source QGIS</i> .....	87
<i>Figure 38 : Fichiers de classes (*. h,*. Cpp)</i> .....	88
<i>Figure 39 : code source remodelé</i> .....	88
<i>Figure 40 : Ensemble des fichiers du plug-in Python</i> .....	89
<i>Figure 41 : Fenêtre homme-machine de notre Plug-in (extention.*ui)</i> .....	90
<i>Figure 42 : Fichier contenant la classe (extention.*py)</i> .....	91
<i>Figure 43 : Cube de données</i> .....	92
<i>Figure 44 : Connexion au Cube de données</i> .....	93
<i>Figure 45 : Installation du nouveau Plug-in</i> .....	94
<i>Figure 46 : Installation de nouveau Plug-in</i> .....	94
<i>Figure 47 : : Le Plug-in Développé</i> .....	95
<i>Figure 48 : Icône de connexion Entrepôt de données spatial</i> .....	95
<i>Figure 49 : Paramètre de connexion à l'Entrepôt de données spatial « GouvData »</i> .....	96
<i>Figure 50 : Test :la connexion à l'EDS est réassie</i> .....	96
<i>Figure 51 : connexion à l'entrepôt de données spatial</i> .....	97

<i>Figure 52 : Projection de l'ensemble des données spatiales figurant sur l'EDS.....</i>	<i>98</i>
<i>Figure 53 : Affichage de la partie attributaire associée aux tables .....</i>	<i>98</i>
<i>Figure 54 : Architecture de notre Prototype géodécisionel .....</i>	<i>99</i>
<i>Figure 55 : Prototype développé.....</i>	<i>102</i>
<i>Figure 56 : Fenêtre principale du prototype.....</i>	<i>103</i>
<i>Figure 57 : Fenêtre de Planification urbaine .....</i>	<i>105</i>
<i>Figure 58 : Définition de l'objet étudié par l'intermédiaire de .....</i>	<i>106</i>
<i>Figure 59 : Figure montrant le zoning du centre urbain Tiflet.....</i>	<i>106</i>
<i>Figure 60 : Critères de recherche de la zone considérée.....</i>	<i>107</i>
<i>Figure 61 : Liste des items affichés relatifs à la couche Équipement .....</i>	<i>107</i>
<i>Figure 62 : Fenêtre permettant de renseigner les critères relatifs au type centre commercial.....</i>	<i>108</i>
<i>Figure 63 : définition d'une zone tampon et des caractéristiques d'un antenne BTS en souffrance...108</i>	<i>108</i>
<i>Figure 64 : Zone tampon du BTS_OULMES_2.....</i>	<i>109</i>
<i>Figure 65 : Fenêtre affichant l'absence de résultat .....</i>	<i>110</i>
<i>Figure 66 : fenêtre affichant un résultat unique.....</i>	<i>111</i>
<i>Figure 67 : représentation de l'entité unique sur fond satellitaire Google Earth.....</i>	<i>112</i>
<i>Figure 68 : Fenêtre affichant des résultats multiples.....</i>	<i>113</i>
<i>Figure 69 : Représentation de l'une des zones sélectionnées à partir du tableau .....</i>	<i>114</i>
<i>Figure 70 : Fenêtre des statistiques .....</i>	<i>115</i>
<i>Figure 71 : Choix de la ville .....</i>	<i>116</i>
<i>Figure 72 : Choix du quartier .....</i>	<i>116</i>
<i>Figure 73 : Dimension temporelle .....</i>	<i>117</i>
<i>Figure 74 : Affichage des résultats de la requête « moyenne du trafic mensuel » pour chacun des BTS analysés.....</i>	<i>117</i>
<i>Figure 75 : Exemple d'une ligne Sélectionnée (ligne 3).....</i>	<i>118</i>
<i>Figure 76 : Affichage d'un graphe en bâtons sur un fond satellite. ....</i>	<i>118</i>
<i>Figure 77 : Affichage d'un graphe en 3D en fonction des années.....</i>	<i>119</i>
<i>Figure 78 : Figure du langage MXD .....</i>	<i>120</i>

<i>Figure 79 : Ensemble des Dimensions existant sur l'entrepôt de données spatial</i> .....	121
<i>Figure 80 : Liste des champs</i> .....	121
<i>Figure 81 : Liste des valeurs uniques</i> .....	122
<i>Figure 82 : Rédaction Semi-automatique des requêtes MXD</i> .....	122
<i>Figure 83 : Affichage d'une requête MXD suivi du tableau de résultats</i> .....	123
<i>Figure 84:Requête MXD Affichant la valeur moyenne de la densité de trafic des BTS de la ville de Rabat, pendant l'année 2013</i> .....	124
<i>Figure 85 : Exportation des résultats dans la table attributaire</i> .....	125
<i>Figure 86 : La surface des cercles est proportionnelle à la densité relative du trafic des BTS de la région de Rabat</i> .....	125
<i>Figure 87 : Affichage sur une Image satellitaire « BingMap » de la densité relative du trafic des BTS de la région de Rabat pour l'année 2013</i> .....	126
<i>Figure 88 : Liste des fonds satellites disponibles</i> .....	127
<i>Figure 89 : Affichage des résultats sur un fond satellite BingMap</i> .....	127
<i>Figure 90 : Les résultats précédents sont affichés sur un fond OpenStreetMap</i> .....	128
<i>Figure 91 : Les résultats sont affichés sur un fond « Bing Aerial with labels »</i> .....	128
<i>Figure 92 : Tableau comparatif des propriétés respectives du Solap « GouvRes » développé et du Solap Pantaho(C)</i> .....	129
<i>Figure 93 : Intégration des données avec PostGIS</i> .....	143
<i>Figure 94 : Connexion à l'Entrepôt de données</i> .....	144
<i>Figure 95 : Résultat de l'integration</i> .....	145
<i>Figure 96 : Création d'un Schéma Workbench</i> .....	148
<i>Figure 97 : Choix d'une source de données</i> .....	148
<i>Figure 98 : Fenêtre de schéma</i> .....	149
<i>Figure 99 : Ajout du cube</i> .....	149
<i>Figure 100 : Ajout de la table</i> .....	149
<i>Figure 101 : ajout d'une dimension</i> .....	150
<i>Figure 102 : Ajout d'une table</i> .....	150
<i>Figure 103 : Ajout d'un niveau</i> .....	151
<i>Figure 104 : Définition des nouveaux niveaux</i> .....	152

<i>Figure 105 : création des mesures .....</i>	<i>153</i>
<i>Figure 106 : Cube .....</i>	<i>186</i>
<i>Figure 107 : modèle logique et physique de données.....</i>	<i>188</i>
<i>Figure 108 : Agrégation.....</i>	<i>189</i>
<i>Figure 109 : Slicer .....</i>	<i>190</i>
<i>Figure 110 : Installation du Plugin-Builder.....</i>	<i>194</i>
<i>Figure 111 : Creation du Plugin.....</i>	<i>195</i>
<i>Figure 112 : Enregistrement du Plugin.....</i>	<i>196</i>
<i>Figure 113 : Fenêtre Résultat .....</i>	<i>197</i>
<i>Figure 114 : Fenêtre OSGeo4W Shell.....</i>	<i>198</i>
<i>Figure 115 : Acces au Dossier Créé .....</i>	<i>198</i>
<i>Figure 116 : Compilation de fichier.....</i>	<i>199</i>
<i>Figure 117 : Intégration de Plug-in .....</i>	<i>199</i>
<i>Figure 118 : Fenêtre Principale .....</i>	<i>200</i>
<i>Figure 119 : Fenêtre QtCreator.....</i>	<i>200</i>
<i>Figure 120 : Fenêtre de plug-in .....</i>	<i>201</i>
<i>Figure 121: Fenêtre en mode creation sous QtCreator .....</i>	<i>201</i>
<i>Figure 122 : Ajout des composants sur la fenêtre.....</i>	<i>202</i>
<i>Figure 123 : Ajout d'un comosant "QwidgetComboBox" .....</i>	<i>203</i>
<i>Figure 124 : Sauvegarde de la fenêtre .....</i>	<i>204</i>
<i>Figure 125 : Rechargement du Plugin Developpé .....</i>	<i>205</i>
<i>Figure 126 : choix du plugin à recharger .....</i>	<i>205</i>
<i>Figure 127 : Intégration définitif du plugin .....</i>	<i>206</i>
<i>Figure 128 : Partie du code Python .....</i>	<i>207</i>
<i>Figure 129 : Affichage des données .....</i>	<i>208</i>
<i>Figure 130 : Exemple des résultats .....</i>	<i>208</i>
<i>Figure 131 : Ajout de composant .....</i>	<i>209</i>
<i>Figure 132 : Ajout de QFileDialog.....</i>	<i>210</i>

<i>Figure 133 : Ajout de code Python correspondant.....</i>	<i>211</i>
<i>Figure 134 : le code Python à ajouter.....</i>	<i>212</i>
<i>Figure 135 : résultat du code source .....</i>	<i>212</i>
<i>Figure 136 : partie du code Python .....</i>	<i>213</i>

# Liste des Abréviations

SIG : Système d'Information Géographique

BTS: Base Transceiver System

ED : Entrepôt de Données

EDS : Entrepôt de Données Spatial

BI: Business Intelligence

ETL : Extract-Transforme-Load

OLAP: OnLine Analytical Processing

SOLAP: Spatial OnLine Analytical Processing

ROLAP: Relational OnLine Analytical Processing

MOLAP : Multidimensional OnLine Analytical Processing

TIC : Technique de l'Information et de Communication

SAD : Système d'Aide à la Décision

SADS : Système d'Aide à la Décision Spatiale

SI : Système d'Information

UML : Unified Modeling Language

OCL : Object Constant Language

SQL : Structured Query Language

GPL: General Public License

BSD: Berkeley Software Distribution

MPL: Mozilla Public License

OSBI: Open Source Business Intelligence

BSS: Base Station Subsystem

BSC: Base Station Controller

RNC: Radio Network controller

LTE: Long Tern Evolution

RAN: Radio Access Network

OMC: Operating and Maintenance Centre

PDU : Plan de Développement Urbain

PDAR : Plan de Développement des Agglomérations Rurales

PA : Plan d'Aménagement

MXD : Multi-Dimensional Expressions

TI : Techique de l'Information

# Résumé

Les opérateurs de téléphonie mobile accumulent dans leurs systèmes de gestion de base de données relationnelle, un volume considérable de données supportant une information géographique implicite qui est indispensable aux prévisions de développement de l'entreprise.

1. Ce travail a donc eu pour objectif l'élaboration d'un prototype SOLAP afin de réaliser des statistiques multidimensionnelles et d'établir des prévisions d'extension du parc d'antennes radio. À cette fin, nous nous sommes appuyés sur les six concepts suivants.
2. **Le Big-Data** : permettant de stocker toutes les données de l'entreprise de manière historique ;
3. **La modélisation multidimensionnelle** : permettant de structurer les données autour des notions de « fait » et de « dimension » ;
4. **OLAP** : Permettant de naviguer dans le Big Data afin de réaliser des statistiques multidimensionnelles ;
5. **Le SIG** : Les données géoréférencées peuvent être représentées à toutes les échelles sur des fonds topographiques ou satellitaires ;
6. **SOLAP** : Combinaison des propriétés des SIG et des OLAP permettant :
  - a. la représentation des analyses statistiques sur des fonds topographiques et satellitaires ;
  - b. la représentation des résultats de scénarios de prévision d'extensions du parc d'antennes.
7. **l'Open Source** : nous a permis de développer un prototype SOLAP opérationnel à partir de logiciel libre.

La conception de notre prototype s'est effectuée en 4 étapes :

1. Conception du modèle multidimensionnel de la base de données spatiale ;
2. Création sous PostgreSQL de l'Entrepôt de données Spatial « GouvData » ;
3. Développement de notre SOLAP « GouvRes » réalisant le couplage entre :
  - a. une plateforme SIG (sous QGIS)
  - b. et un OLAP « cube » développé sur une plateforme « Python »
4. Réalisation d'une interface « personne-machine » permettant de formuler des requêtes statistiques et des scénarios de planification urbaine.

Mots Clés : SIG, BI, OLAP, SOLAP, Entrepôt de Données, Entrepôt de Données Spatial, Datawarehouse, Magasin de Données, DataMart, Open Source.



# Abstract

Mobile operators amass in their relational database management system (RDBMS), a considerable volume of data including geographic implied information that is essential to the forecast of the business development.

This work, therefore, aimed to the development of a prototype SOLAP to carry out multidimensional statistics and to make forecasts of extension of the Park of radio antennas. To this end, we relied on six concepts:

1. **The Big Data** allowing storing all the data of the business in a chronological way
2. **The multidimensional modelling** for structuring the data around the notions of ‘fact’ and ‘dimension’.
3. **OLAP** For navigating in Big Data to perform multidimensional statistics
4. **GIS** spatial data can be represented at all scales on satellite and topographical background.
5. **SOLAP**: Combination of GIS and the OLAP properties for:
  - A** – The representation of statistical analysis on satellite and topographical background.
  - b** – The representation of forecast scenarios for antenna farm extensions.
6. **Open Source**: enabled us to develop an operational SOLAP prototype from free software.

The design of our prototype is done in 4 steps:

1. The multidimensional model of the spatial database design
2. The Establishment, under PostgreSQL, from the data warehouse space ‘GouvData’
3. The Development of our SOLAP ‘GouvRes’ realising the coupling between:
  - a** – Platform SIG (under QGIS)
  - b** – And an OLAP ‘cube’ developed on a platform ‘Python’.
4. Creation of a ‘man-machine interface’ to formulate statistical queries and urban planning scenarios.

Key Word : SIG, BI, OLAP, SOLAP, Entrepôt de Données, Entrepôt de Données Spatial, Datawarehouse, Magasin de Données, DataMart, Open Source.



# INTRODUCTION

## Prémisses

Les Techniques de l'information et Communication (TIC) sont de plus en plus utilisées, non comme de simples supports de communication ou d'outils améliorant les conditions de travail ou plus généralement la vie quotidienne ; mais comme de réels facteurs de développement et de promotion d'un territoire, d'enrichissement de la qualité des systèmes éducatifs et du développement économique et social des pays. Dans cette perspective, le Maroc en tant que pays en voie de développement a lancé un projet courageux dans cette direction « Maroc numérique 2020 ».

La concrétisation des objectifs ambitieux fixés par cette stratégie repose sur la mise en œuvre de quatre (4) priorités stratégiques : (transformation sociale, services publics orientés usagers, productivité des PME [Petites et Moyennes Entreprises], industrie des TI [Technologie de l'Information]); de deux (2) mesures d'accompagnement (confiance numérique, capital humain) ; et de deux (2) modalités d'implémentation jugées essentielles pour réussir sa mise en œuvre (pilotage de la stratégie et ressources financières).

Le projet Maroc numérique s'appuie notamment sur le dynamisme du secteur des télécommunications lié au déploiement du service 4G afin d'assurer un service à la hauteur des attentes des citoyens. Dans ce contexte, les opérateurs doivent gouverner leurs services et adopter en particulier une stratégie de décision rigoureuse relative au développement du parc d'antennes émettrices-réceptrices. C'est dans cette perspective que s'inscrit la réalisation de cette thèse : « Allier le temps à l'espace en vue d'une bonne gouvernance des réseaux télécom » qui propose un prototype géodécisionnel créé à partir d'outil open source permettant ainsi de répondre aux préoccupations économiques et stratégiques des opérateurs de téléphonie mobile, permettant de rationaliser les prévisions d'extension des infrastructures tout en réalisant de substantielles économies d'échelle.

Un système de téléphonie mobile est composé de 3 parties essentielles : un réseau d'accès de nature cellulaire, composé d'antennes radio relais. Ce réseau est raccordé au cœur du système, par un réseau d'amenée ou backhaul.

L'analyse spatio-temporelle des fluctuations du trafic dans le réseau d'accès constitue un enjeu primordial tant en ce qui concerne la maintenance opérationnelle du réseau que pour les décisions à prendre dans les perspectives d'extension à long terme de ses infrastructures.

Depuis le début des années 1970, l'informatique propose aux entreprises et aux chercheurs plusieurs types d'applications favorisant l'aide à la décision. Parmi celle-ci nous avons distingué 3 groupes de technologie : 1 — Les Systèmes d'Information Géographique (SIG) conçus pour recueillir, stocker, traiter, analyser, gérer et représenter sur des cartes, tous les types de données ayant trait à l'infrastructure spatiale. 2— Les OLAP (en anglais Online Analytical Processing, «OLAP») qui permettent l'analyse en ligne des données en produisant des rapports de synthèse qui offrent une vue transversale de l'activité de l'entreprise. 3— Les SOLAP qui résultent du couplage des fonctions propres aux deux outils précédents.

## **Objectifs de la thèse**

Les opérateurs de téléphonie mobile accumulent au cœur de leur système et dans différentes bases de données opérationnelles (Oracle, etc.) une masse considérable de données numériques et alphanumériques géoréférencées ou assorties d'attributs géographiques. Ces données contiennent une information potentielle latente inexploitable dans les systèmes transactionnels.

Cette thèse a pour objet de proposer un prototype géodécisionnel (SOLAP) permettant d'extraire ces données et de les stocker afin d'exploiter cette information masquée en la représentant sur des cartes interactives facilement accessibles et interprétables par les décideurs.

Les capacités fonctionnelles de ce prototype seront testées, validées et illustrées à partir des données relatives aux antennes émettrices-réceptrices d'un réseau de téléphonie mobile de la ville de Rabat.

## Contexte de la thèse

L'informatique décisionnelle propose des systèmes d'Entrepôts de Données (ED) et des OLAP (On-Line Analytical Processing) qui permettent l'analyse en ligne de très gros volumes de données numériques et alphanumériques provenant de sources de données multiples et hétérogènes (Imnon, 1992).

Cœur du système, l'entrepôt de données est l'espace de stockage, d'intégration et d'historisation de toutes les données de l'entreprise qui sont utiles à l'analyse décisionnelle.

Les outils OLAP permettent l'exploration interactive et simplifiée de cet espace à travers une schématisation logique multidimensionnelle s'appuyant sur le modèle métaphorique de l'hypercube. Ce concept est organisé à partir des notions de fait et de dimension.

Les faits représentent les sujets d'analyse et sont décrits par des mesures généralement numériques.

Les dimensions définissent les axes d'analyse des faits et sont organisées en hiérarchies ayant plusieurs niveaux d'agrégation favorisant la visualisation des mesures à différentes échelles.

L'exploration en ligne de l'hypercube de données est réalisée à l'aide d'opérateurs effectuant des fonctions d'agrégation telles que : Sum, Max, etc. Ainsi : les opérateurs de forage vers le haut (Roll-Up) et vers le bas (Drill-Down) supportent respectivement de monter/descendre dans les hiérarchies des dimensions ; les opérateurs de coupe slice et dice choisissent une partie des données de l'hypercube ; tandis que l'opérateur de projection permet de sélectionner une ou plusieurs mesures, etc.

Ainsi, dans une situation standardisée une analyse multidimensionnelle portant par exemple sur un fait « ventes » d'un groupe de magasins, pourra être réalisée en définissant comme mesures « le volume de la vente » et « le montant de la vente ». Les opérateurs OLAP (Roll-Up, Drill-Down, Slice, etc.) permettent de visualiser les mesures pour des ensembles de membres à des niveaux de granularité sélectionnés par l'utilisateur. Les opérations de forage (Roll-Up, Drill-Down) sont développées à l'aide de fonctions d'agrégation

imputées aux mesures, par exemple la somme appliquée au volume de produits vendus.

Des interfaces (tableaux de bord, tableaux multidimensionnels, graphes) complètent l'arsenal des outils décisionnels.

Mais l'information géographique, pourtant essentielle dans une prise de décision stratégique n'est pas prise en compte par les systèmes OLAP. Seul le SIG (Système d'Information Géographique) permet d'acquérir, de structurer, de mémoriser, d'analyser et d'afficher les données géographiques.

Partant de ce constat, une solution mixte, connue sous le sigle SOLAP ou « Spacial OLAP », a été définie par Yvan Bédard comme « une plateforme visuelle conçue spécialement pour supporter une analyse spatio-temporelle rapide et efficace à travers une approche multidimensionnelle incluant des niveaux d'agrégation cartographiques, graphiques et tabulaires » [Bédard, 1997].

Le SOLAP enrichit donc les capacités d'analyse des systèmes OLAP ; car la visualisation des mesures sur une carte permet de comprendre la distribution géographique d'un fait et de mettre en relation les différents phénomènes spatiaux par rapport aux axes d'analyse alphanumériques et de comparer ces phénomènes à diverses échelles spatiales en outre, le SOLAP offre à l'utilisateur l'accès aux opérations de navigations multidimensionnelles à partir d'une carte interactive.

# **Organisation du mémoire**

Ce mémoire de thèse comporte 3 parties décomposées en neuf chapitres.

## **PARTIE I — ÉTAT DE L'ART**

### ***Chapitre 1***

Dans ce chapitre sont présentés les 4 éléments constituant l'architecture standard des systèmes décisionnels. Les principaux concepts des entrepôts de données et de l'analyse en ligne sont également exposés. Les principes de la modélisation multidimensionnelle sont également abordés, car la formalisation de notre modèle Solap « GouvRes » est conçue à partir de ces concepts. Ce chapitre se termine par la description des principales fonctionnalités d'un serveur OLAP et des différentes modalités de leur implantation.

### ***Chapitre 2***

Ce chapitre est dédié à la définition des caractéristiques fondamentales de l'information géographique. Une attention particulière est apportée aux aspects spatiaux et sémantiques qui la caractérisent. Ainsi, nous introduisons les concepts principaux des systèmes d'information géographiques : les bases de données spatiales ; les deux modes de représentation de l'espace (raster et vectoriel) sont explicités. Les méthodes d'analyse spatiales sont également abordées. Dans cette perspective, nous détaillons les principes de la topologie, car notre modèle Solap « GouvRes » est construit sur le modèle vectoriel qui autorise uniquement une représentation relative des objets géographiques.

### ***Chapitre 3***

Dans ce chapitre sont introduits les concepts principaux de l'OLAP Spatial : mesures, faits et dimensions spatiales. Nous effectuons une classification de ces définitions et un état de l'art des différents modèles formels SOLAP et nous présentons la typologie des approches pour le développement des applications SOLAP.

### ***Chapitre 4***

Ce chapitre est consacré à la définition du concept « open source ». Il offre la possibilité d'accéder au code source des applications open sources, afin de les modifier en fonction des besoins de l'utilisateur.

## ***Chapitre 5***

Ce chapitre expose les principaux concepts de la téléphonie mobile. Il met en particulier l'accent sur la structure du réseau d'accès de la téléphonie mobile qui constitue l'objet de notre étude.

## **PARTIE II — CONCEPTION DU MODÈLE SOLAP**

Dans la deuxième partie de ce mémoire, nous décrivons les différentes étapes relatives à la création d'un prototype géodécisionnel (S-OLAP), spécifiquement dédié aux besoins des opérateurs de téléphonie mobile. Ce prototype est mis au point à l'aide d'outils Open Source dont nous avons décrit les caractéristiques au chapitre 4.

## ***Chapitre 6***

Ce chapitre a pour objet la conception, la mise en œuvre et la validation de notre Entrepôt de Données Spatial (EDS) spécialement dédié aux besoins spécifiques des opérateurs de téléphonie mobile.

## ***Chapitre 7***

Ce chapitre est consacré aux différentes étapes du développement de notre prototype SOLAP « GouvRes ».

## ***Chapitre 8***

Ce chapitre est dédié à la mise en œuvre de notre prototype géodécisionnel à travers la formulation de 3 types de requêtes et à la visualisation des résultats sur des fonds topographiques et satellitaires.

## **PARTIE III — CONCLUSIONS**

## ***Chapitre 9***

Ce chapitre ouvre une discussion sous forme de conclusion et envisage les perspectives de cette thèse.

A decorative L-shaped line consisting of a vertical line on the left and a horizontal line on the bottom, intersecting at the top-left corner of the text.

# **PARTIE I**

## **ÉTAT DE L'ART**



# Chapitre 1 : Systèmes d'Aide à la Décision

Dans ce chapitre seront décrits les différents niveaux des systèmes d'aide à la décision ; mais l'accent sera essentiellement mis sur les entrepôts de données, l'analyse multidimensionnelle et les serveurs OLAP.

Dans un Entrepôt de données, l'information représente un capital immatériel dont la bonne gestion est un facteur primordial pour la réussite de toute organisation. Les systèmes d'aide à la décision ou encore BI (pour Business Intelligence) ont pour objectif de supporter la réalisation des activités d'une organisation. Ils sont construits à partir des exigences des métiers et des processus définis par l'entreprise afin de stocker, traiter et communiquer les informations.

Les Systèmes d'Information (SI) des entreprises accumulent au fil du temps un volume important de données stockées sur plusieurs sites internes à l'entreprise ou provenant de son environnement externe (partenaires, Web, etc.). Le problème des entreprises est d'exploiter efficacement ces données, afin de permettre aux décideurs d'optimiser et d'anticiper leurs choix. Ainsi, le besoin d'une exploitation efficace des données dans une perspective décisionnelle a donné lieu à l'élaboration de nouveaux systèmes, dits systèmes d'aide à la décision (SAD), facilitant le stockage et le traitement synthétique de grands volumes de données. Ils permettent de sélectionner des informations opérationnelles pertinentes pour l'entreprise et en particulier pour les décideurs (Alter 1980). Ils regroupent des informations et d'outils qui permettent de supporter de manière efficace la prise de décision (Codd *et al.* 1993) (Imnon 1994) (Chaudhuri et Dayal 1997).

Les SAD réunissent, normalisent et coordonnent les bases de données, les modèles d'analyses et les techniques de visualisation.

Les systèmes d'aide à la décision sont employés dans tous les domaines où la prise de décision est nécessaire, à savoir, les domaines du commerce (marketing, ventes), de la logistique, de la santé (aide à la décision médicale), de la science (par exemple en bio-informatique), des télécommunications, des transports (trafic autoroutier)...

Afin d'offrir une vision transversale de l'activité de l'entreprise, les systèmes d'aide à la décision collectent et stockent des données en provenance des bases de données des différents métiers de l'entreprise et de sources externes (sites web, emails...). L'architecture des systèmes d'aide à la décision comporte 4 niveaux : deux niveaux de stockage et deux niveaux opérationnels (outils ETL et outils OLAP) (Figure 1).

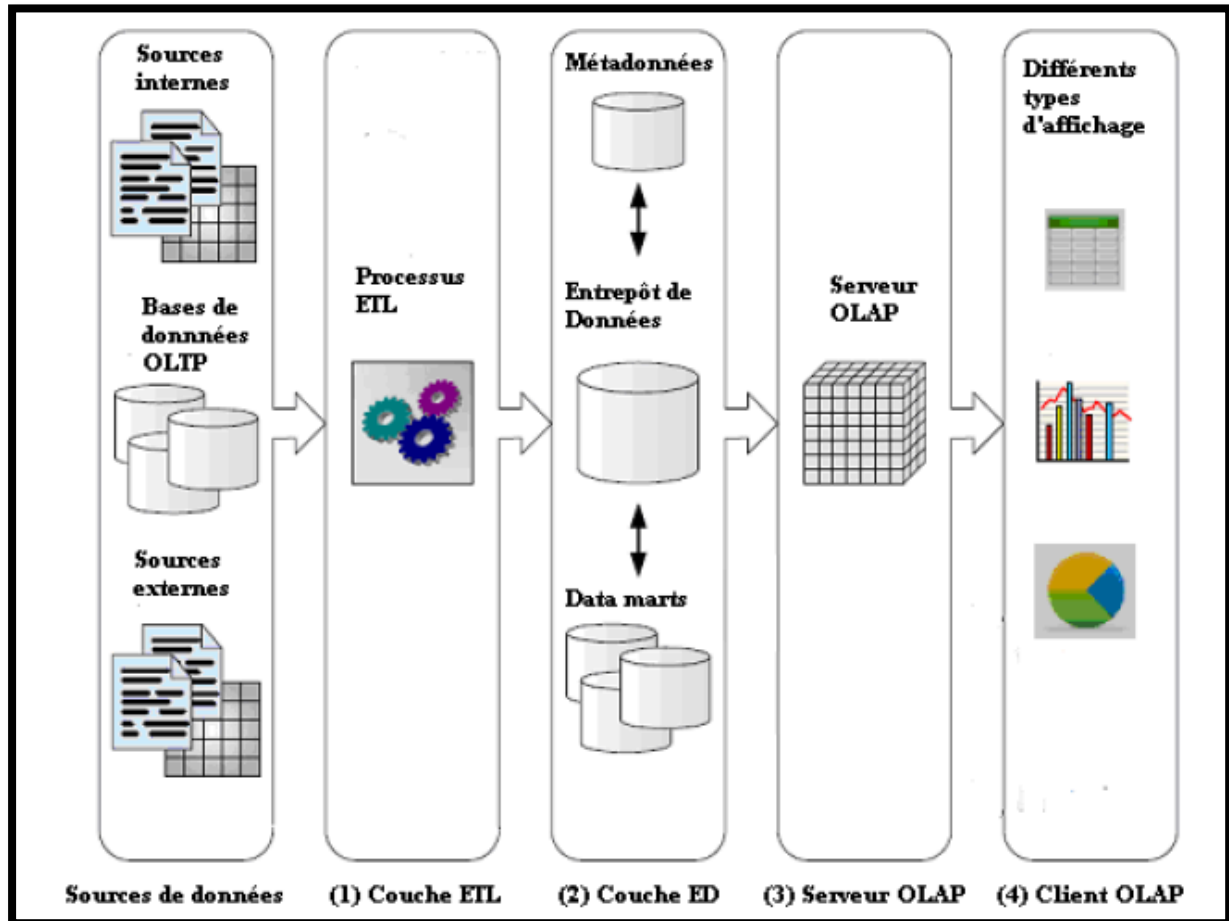


Figure 1 : Architecture typique d'un OLAP. Adaptée de (Malinowski et Zimányi 2008).

(1) Les outils d'extraction, de transformation et de chargement ou ETL (Extract, Transform, Load) permettent d'alimenter et de rafraîchir les données contenues dans les entrepôts et les magasins de données à partir des sources.

(2) L'entrepôt regroupe toute l'information décisionnelle. C'est le lieu de stockage centralisé d'un extrait des sources. Son organisation doit faciliter l'intégration des données et la conservation de leurs évolutions. Le magasin est un extrait de l'entrepôt. L'organisation des données d'un magasin suit un modèle spécifique qui facilite les analyses décisionnelles.

(3) Les outils OLAP (OnLine Analytical Processing) permettent d'interroger et d'analyser les données sous une forme adaptée aux décideurs.

(4) Le Client OLAP permet de gérer les différents types d'affichage et de reporting.

## 1.1 Outils d'extraction, ETL (ExtractTransformLoad)

Les sources de données sont nombreuses et variées. Ce sont des bases de production de nature relationnelle interne ou externe (Internet, bases des partenaires). Leur acquisition par les outils ETL est la source de nombreux aménagements dans le système d'information : changement d'encodage, ouverture de liens d'accès aux SGBD, adaptation du réseau et mise en place d'authentifications (Tranchant 2011).

Ces outils sont indispensables afin de réaliser l'alimentation et la mise à jour des entrepôts de données. Après avoir extrait les données des différentes sources, l'ETL leur applique des transformations afin de pouvoir les intégrer dans une structure multidimensionnelle.

Les processus couverts par l'ETL sont complexes et doivent faire l'objet d'une attention particulière : il en va en effet de la crédibilité des données qui seront ensuite exploitées par l'OLAP et qui serviront de base aux décisions des analystes. Tranchant rappelle les différentes opérations qui doivent être réalisées (Tranchant, 2011) :

- **Filtrage** : identifie les données aberrantes ;
- **Dé-doublonnage** : empêche l'insertion en de multiples exemplaires de la même information, issue de plusieurs sources de données différentes ;
- **Formatage** : arrange les données afin qu'elles partagent toutes le même référentiel (même unité, même multiplicateur) ; transforme les informations codifiées (abréviations) pour les unifier ; standardise les dates ;
- **Dé-normalisation** : redondance de l'information. Les sources étant généralement issues de bases relationnelles s'appuyant sur la troisième forme normale, l'outil « aplanit » les relations entre entités afin de n'en garder que des données pertinentes (les clés primaires des SGBDR sont par exemple inutiles) ;

- **Synchronisation** : garantit la cohérence des agrégats. Les mêmes éléments peuvent avoir des représentations différentes selon leurs sources, il s'agit ici de les détecter et de les traiter en conséquence ;
- **Désagrégation** : transforme les données en changeant leur support. On peut ainsi passer d'un indicateur de haut niveau vers un niveau plus petit en le pondérant, grâce à d'autres informations. Par exemple : Déterminer le taux de pollution d'une commune à partir de celui d'un bassin versant, en le pondérant par la superficie de ladite commune.
- **Agrégation** : effectue une ou plusieurs opérations sur les données, parmi lesquelles on retrouve la somme, cumul des sommes, la moyenne, le comptage, le comptage distinct, etc.

## 1.2 Entrepôts de données et magasins de données

### 1.2.1 Définition de l'entrepôt de données

Pivot du système décisionnel, l'entrepôt de données est « une base comportant un très grand nombre de données, organisées de telle sorte que les utilisateurs puissent obtenir rapidement des réponses à des requêtes et des analyses (Noirault 2006) (Bédard *et al.* 1998). Cette organisation est réalisée en fonction des besoins analytiques des utilisateurs. Selon Imnon (Imnon 1994), un entrepôt de données est « une collection de données, intégrées, non volatiles, historiées, résumées et disponibles pour l'interrogation et l'analyse » :

**Orienté sujet** : l'entrepôt est organisé pour répondre à une problématique métier, par exemple, les habitudes d'achat des clients, ou l'efficacité d'une équipe...;

**Intégrées** : l'intégration consiste à résoudre les problèmes d'hétérogénéité des systèmes de stockages opérationnels, des modèles de données, de sémantiques des données. Afin que toute entrée de l'entrepôt soit uniforme, les valeurs sont toutes converties dans la même unité par exemple : les centimètres, les heures, etc. ;

**Historié** : contrairement aux données opérationnelles qui ont une durée limitée, les entrepôts de données doivent permettre une analyse historique. De ce fait, chaque donnée est horodatée et ne sera jamais effacée ni modifiée (voir

« non volatile »). Ceci permet à l'utilisateur du système de constater une progression, comprendre les tendances, etc. ;

**Non volatile :** ici synonyme de « lecture seule ». Pouvoir modifier une donnée reviendrait à avoir la possibilité de modifier le passé, ce qui n'est jamais une bonne chose. Les données sont associées à un instant, valables uniquement à un moment donné. Elles ne peuvent pas être modifiées par l'utilisateur ;

**Résumée :** Les informations provenant des sources de données doivent être agrégées et réorganisées afin de simplifier le processus de décision.

## **1.2.2 Définition du magasin de données**

Les magasins de données sont des extraits de l'entrepôt. Dans le magasin les données concernent soit un groupe spécifique d'utilisateurs soit sont adaptées à un usage particulier. Les données y sont organisées suivant un modèle spécifique ayant pour objectif de simplifier les analyses décisionnelles.

## **1.2.3 Modèle multidimensionnel logique**

Du point de vue conceptuel, la modélisation multidimensionnelle est à l'origine des concepts de faits et de dimensions (Kimball 1996). Dans cette optique, le sujet analysé est considéré comme un point dans un espace à plusieurs dimensions. Cette vision correspond à une structuration des données selon plusieurs axes d'analyses (ou dimensions) pouvant représenter des notions variées telles que : le temps, la localisation géographique, le code identifiant des produits...

### **1.2.3.1 Dimensions et hiérarchie**

Une dimension modélise une perspective de l'analyse. Elle se compose de paramètres correspondants aux informations faisant varier les mesures de l'activité. Ces paramètres sont textuels et discrets. Ils ont pour but de limiter la portée des requêtes afin de restreindre la taille des réponses.

Les dimensions constituent les axes de l'analyse. Elles sont organisées en schéma hiérarchique dont voici quelques exemples (Figure 2) :

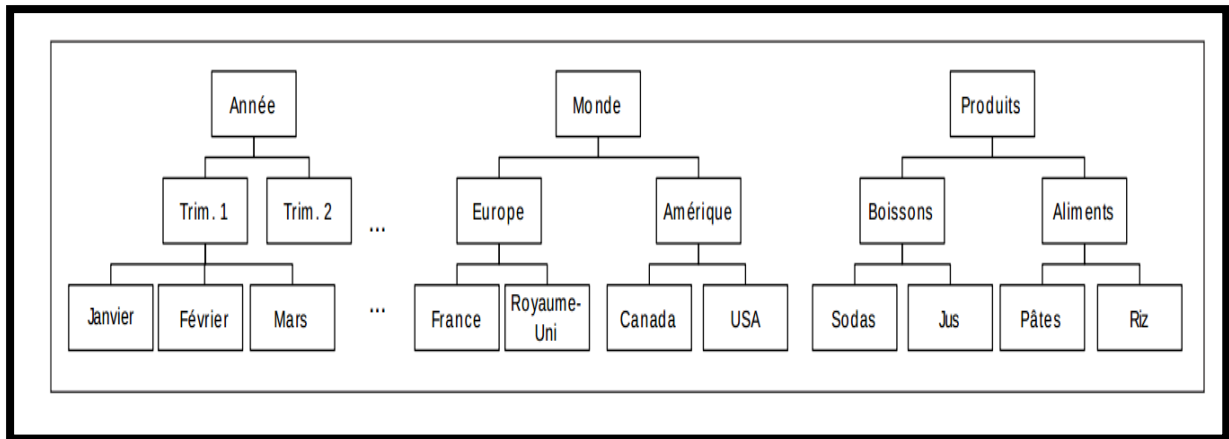


Figure 2 : Dimensions et leur hiérarchie respective. Source : (Dinimant 2009)

La figure 2 représente trois dimensions (temporelle, spatiale, produit). Chacune possède sa propre hiérarchie :

- Temporelle : les années sont divisées en trimestres ; eux-mêmes subdivisés en mois ;
- Spatiale : le monde est partagé en continent, lui-même découpé en pays ;

Produit : les produits sont segmentés en catégories puis en sous-catégories.

La figure 3 représente un exemple détaillé de la dimension temporelle. Tous les rectangles sont les membres de la dimension divisée en niveaux.

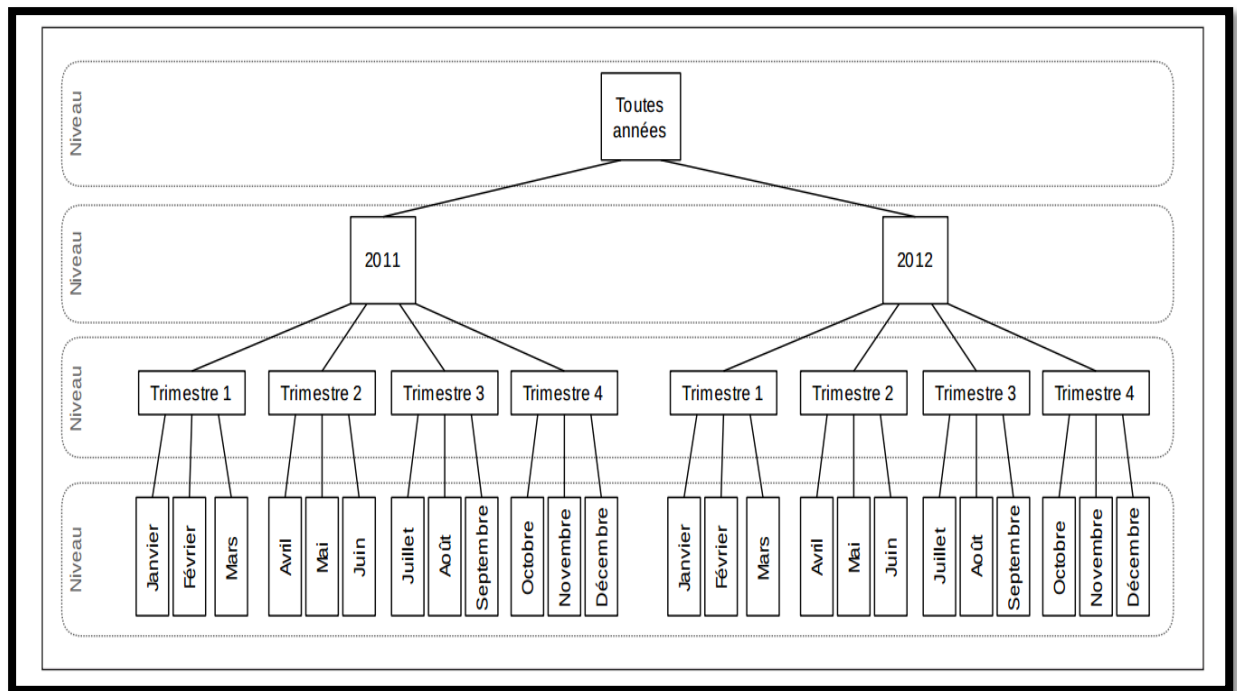


Figure 3 : Niveaux et membres de la dimension temporelle

Le niveau le plus bas de la hiérarchie impose la granularité maximale.

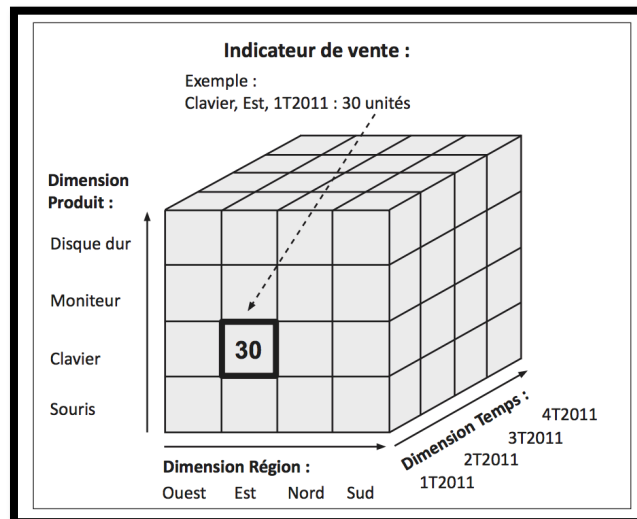
Chaque niveau de la dimension peut présenter des attributs qui ne sont pas utilisés dans la définition du schéma hiérarchique (Hüsemann *et al.* 2000) par exemple, un produit peut avoir un attribut représentant le prix, la couleur, etc... Ces attributs peuvent être utilisés dans l'analyse multidimensionnelle.

Notons enfin que certaines dimensions peuvent posséder plusieurs hiérarchies. Ainsi concernant la dimension temporelle un utilisateur peut être intéressé par la visualisation en année solaire ou en année lunaire. La hiérarchie permet d'analyser efficacement sous plusieurs angles le même type d'information.

### 1.2.3.2 Faits et mesures

Un fait modélise le sujet analysé. Il est défini par plusieurs mesures. Les mesures représentent généralement des valeurs numériques qui apportent une description quantitative du fait (Kimball 1996). Une ou plusieurs associations de membres des dimensions peuvent être associées à un fait. Par ailleurs (Blaschka *et al.* 1998) indiquent que certaines mesures peuvent être calculées à partir d'autres mesures ou propriétés de membres.

Les dimensions s'organisent autour de faits afin de présenter des perspectives d'analyses multiples. Ainsi, un fait référencé par trois dimensions peut être représenté suivant la métaphore du cube (Grey et al 1996), comme l'illustre la figure 4.



*Figure 4 : Représentation multidimensionnelle suivant la métaphore du cube lequel représente trois perspectives d'analyse. Source : (Meier 2006).*

La question, « combien d'unités de clavier ont été vendues dans la région au cours du premier trimestre 2011 », est illustrée (Figure 4)

Un modèle avec plus de trois dimensions est nommé soit « hypercube » soit « n-cube ». Par commodité, on parle simplement de « cube ».

Le mode de stockage multidimensionnel favorise un accès rapide à l'information, les fonctions (somme, moyenne, minimum, maximum, comptage, comptage distinctif) étant plus facilement réalisables lorsque les données sont normalisées et précalculées.

À partir du fait et des dimensions, il est possible d'établir un dispositif simple des entrepôts de données qui correspond au besoin de la modélisation multidimensionnelle :

### 1.2.3.3 Modèle en étoile

La structure en étoile (Kimball 1996) tire son nom de sa configuration : au centre : la table des faits ; autour les dimensions contenant les attributs (Figure 5). Ce choix est généralement justifié par la simplicité de sa mise en place, le recours à un nombre limité de jointures (restreint à celles entre le fait et chacune des dimensions). Les jointures sont limitées à celles existant entre le fait et chacune des dimensions).

Cette disposition implique une forte dé-normalisation ; ce qui exige la duplication des attributs de niveaux supérieurs. En effet, pour introduire les niveaux d'une hiérarchie, les données doivent être répétées autant de fois qu'il y a de lignes dans cette dimension.

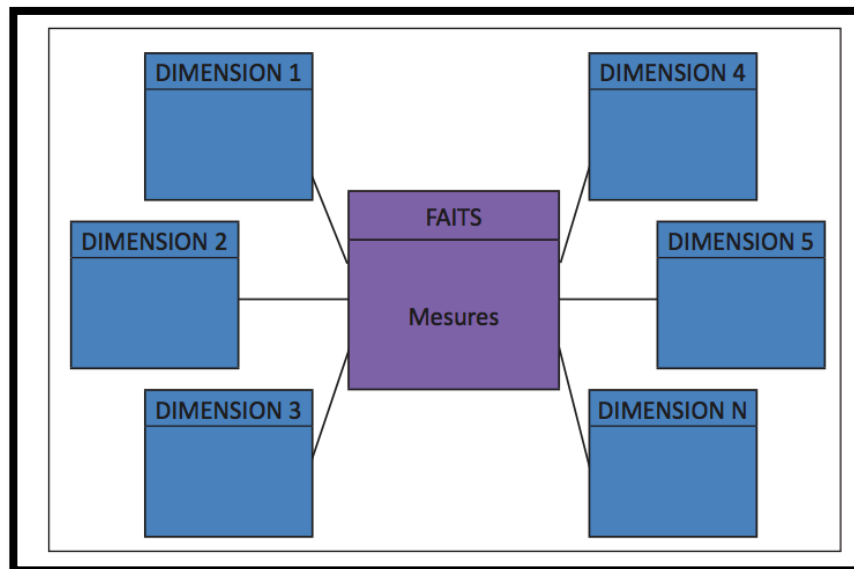


Figure 5 : Structure en étoile. Source : (Bédard 2008)

### 1.2.3.4 Modèle en flocon

La disposition en flocon (Figure 6) découle du dispositif en étoile ; mais dans ce cas, les dimensions sont décomposées suivant les niveaux de leur hiérarchie et sont plus normalisées que dans le schéma étoile. Ainsi, dans chaque dimension, une table représente un niveau.

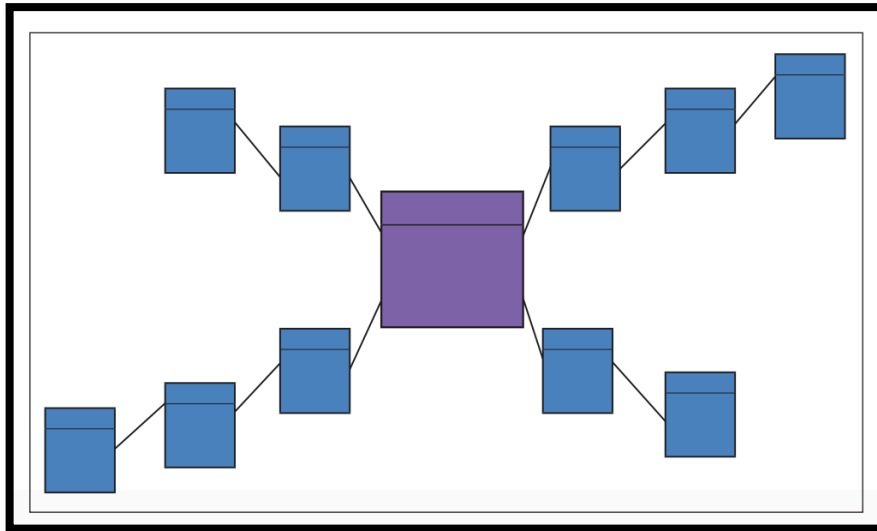


Figure 6 : Structure en flocon. Source : (Bédard 2008)

Ce modèle exige de nombreuses jointures afin d'être exploité. Cette disposition représente malgré tout, une réponse afin de limiter la redondance des informations.

### 1.2.3.5 Modèle mixte

Ce modèle (Figure 7) est un compromis entre les deux modèles précédents. Dans ce cas, seules les plus grandes tables sont normalisées.

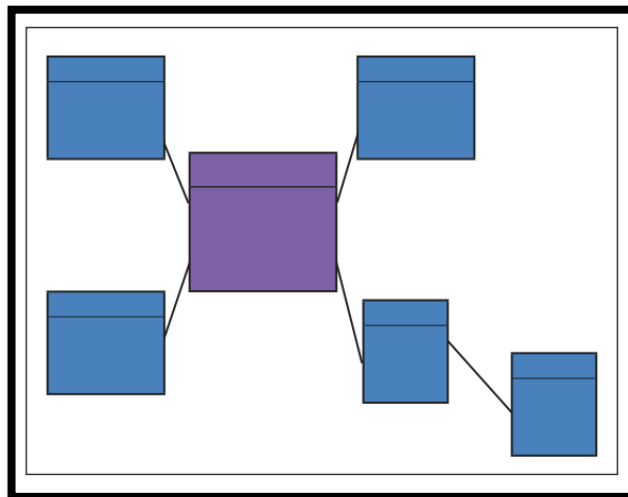


Figure 7 : Structure mixte. Source : (Bédard 2008)

### 1.2.3.6 Modèle en constellation

Dans ce type de structure, plusieurs faits partagent certaines tables de dimensions (Figure 8). Ces dimensions sont qualifiées de conformes. Autour de ces faits, le schéma utilisé peut-être en étoile, en flocon ou mixte. L'utilisation de dimensions conformes permet d'obtenir une grande cohérence entre les vues destinées aux utilisateurs et oblige à une homogénéité des attributs et des mesures.

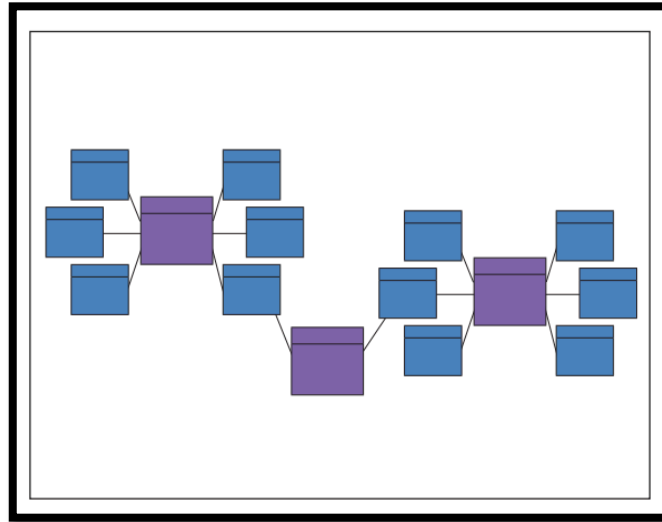


Figure 8 : Structure en constellation. Source : (Bédard 2008)

## 1.3 Serveur OLAP

### 1.3.1 Principe de base

Le serveur OLAP (On-Line Analytical Processing) permet d'effectuer une analyse de données. C'est un outil de médiation entre les logiciels éditant des rapports et les différentes sources auxquelles l'utilisateur peut avoir un accès indirect grâce à l'utilisation systématique des vues multidimensionnelles.

Le principe de base de cet outil de décision a été défini par Codd (Codd *et al.* 1993) à partir de 12 règles relatives à la conception de l'Entrepôt de données et à la conception du serveur OLAP :

1) Vue multidimensionnelle : les données sont structurées en dimensions métiers.

2) **Transparence** : l'utilisateur doit pouvoir utiliser les logiciels habituels (tableur...) sans percevoir la présence de l'outil OLAP.

3) **Accessibilité** : l'outil OLAP doit se charger d'accéder aux données stockées dans n'importe quel type de données (interne+externe) et le faire simultanément.

4) **Performance continue dans la restitution** : à mesure que le nombre de dimensions ou la taille de la base augmente, l'utilisateur ne doit pas subir de baisse sensible de performance.

5) **Architecture client-serveur** : tout produit OLAP doit fonctionner en mode C/S, avec une répartition des traitements.

6) **Dimension générique** : chaque dimension (avec l'analyse) doit être équivalente aux autres à la fois dans sa structure et dans ces capacités opérationnelles. Une seule structure logique dans l'ensemble des dimensions.

7) **Gestion dynamique des matrices creuses** : OLAP doit gérer les cellules non renseignées de manière optimale.

8) **Support multi-utilisateur** : OLAP doit assurer un accès simultané aux données, gérer l'intégrité et la sécurité de ses données.

9) **Opérations entre dimensions** : OLAP doit gérer des calculs associés entre dimensions sans faire appel à l'utilisateur pour définir le contenu du calcul.

10) **Manipulation intuitive** : Minimiser le recours à des menus ou bien les allers et retours avec l'interface utilisateur.

11) **Flexibilité des restitutions** : convivialités des états de gestion ou des états de sortie-ergonomie.

12) **Nombre illimité de dimensions et niveaux de hiérarchie** : l'outil doit gérer au moins quinze dimensions et ne pas limiter le nombre de niveaux hiérarchiques.

Ce concept a été appliqué à un modèle virtuel de représentation de données, appelé cube ou hypercube OLAP. L'hypercube OLAP donne accès à des fonctions d'extraction de l'information (pour visualisation, analyse ou

traitement) et à des fonctions de requête en langage MDX comparable à SQL pour une base de données relationnelle.

### 1.3.2 Fonctionnalité d'un serveur OLAP

Un serveur OLAP procure aux utilisateurs une vue multidimensionnelle des données qui peuvent être analysées grâce à un ensemble d'opérateurs qui permettent d'explorer les données multidimensionnelles en utilisant les concepts de dimensions et de hiérarchie. Un survol des opérateurs OLAP est présenté par (Rafanelli 2003). Les plus courants sont les suivants :

Les opérateurs d'agrégation qui permettent de naviguer entre différentes granularités d'une ou plusieurs dimensions :

**Roll-Up** permet de remonter dans la hiérarchie des dimensions et d'agréger les mesures.

**Drill-Down** permet de descendre dans une hiérarchie et d'afficher une granularité plus fine ; par exemple sur la Figure 9, on observe l'effet du roll-up sur la dimension « Localisation » du cube « Rendement agricole ».

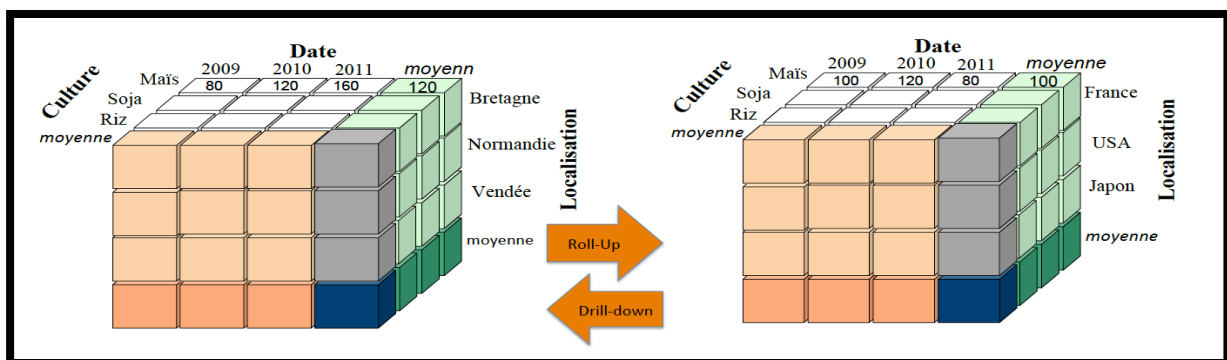


Figure 9 : Opérateurs OLAP, roll-up drill-down. Source : (Bouadi 2014)

Sur le cube « rendement agricole », l'utilisateur réalise un drill-down sur la valeur France de la dimension localisation. Le cube résultant présente les régions ayant pour pays « France ». Le roll-up est l'opération inverse qui permet de remonter dans une hiérarchie vers un niveau plus agrégé.

- **Opérateurs de sélection : Slice et Dice**

L'opérateur slice (littéralement, trancher), permet de sélectionner un sous-ensemble du cube, selon une ou plusieurs valeurs d'une dimension particulière. Slice est un opérateur de sélection portant sur les valeurs d'une seule dimension. La figure 10 montre un exemple de Slice : à partir du cube « Rendement agricole » initial, on sélectionne un sous-ensemble de ce cube tel que la « Localisation » soit « France » ou « USA ».

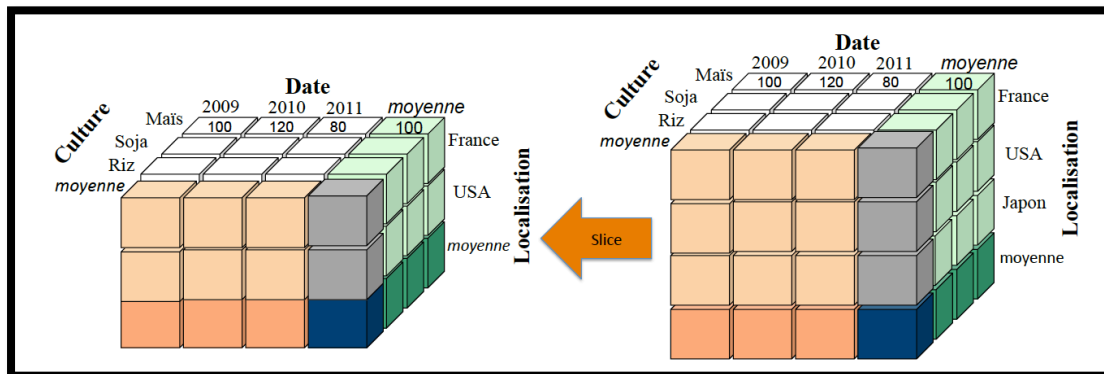


Figure 10 : Opérateur OLAP Slice. Source : (Bouadi 2014)

L'opérateur Dice permet de faire une projection du cube. L'application de Dice équivaut à l'application de l'opérateur Slice à plusieurs dimensions. La Figure 11 montre un exemple de Dice : à partir du cube « Rendement agricole » initial, on sélectionne un sous-ensemble de ce cube tel que la « Localisation » soit « France » ou « USA », et que la « Date » soit « 2009 » ou « 2011 ». Slice, ne peut produire que des tranches du cube, du fait qu'une sélection n'est possible que sur une dimension, alors que Dice peut produire des sous-cubes arbitraires.

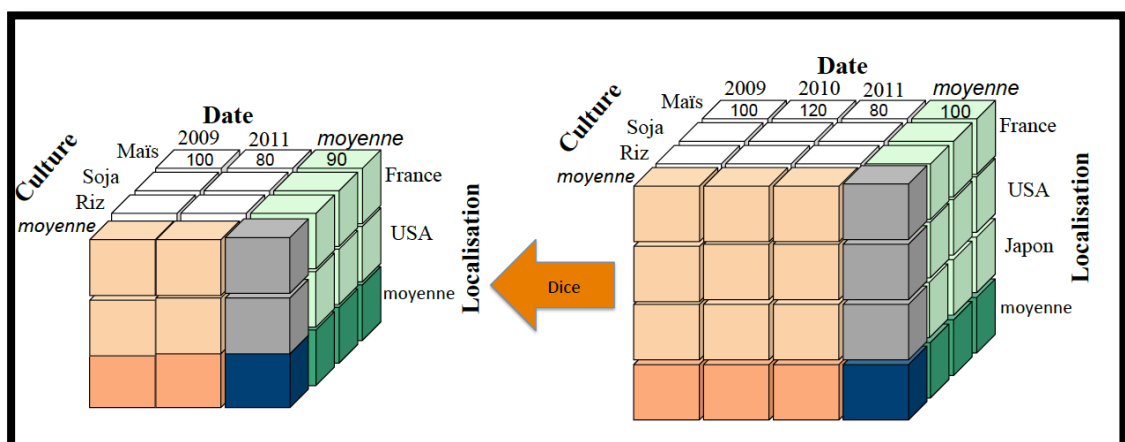


Figure 11 : Opérateur OLAP Dice Source : (Bouadi 2014)

### 1.3.3 Implémentation physique d'un serveur OLAP

- **Approche ROLAP (Relational OLAP)**

Les serveurs ROLAP (Kimball 1996) utilisent une base de données relationnelle pour le stockage des données et des agrégations destinées à l'analyse. La majorité des fonctions nécessaires à l'utilisation d'OLAP sont gérées par un système de gestion de base de données relationnelle (SGBDR). Afin d'obtenir des performances satisfaisantes, ces systèmes utilisent des structures d'indexations particulières et les vues matérialisées (Winter 1998). Une vue matérialisée est la traduction relationnelle d'un sous-cube qui est pré calculé et stocké dans le magasin de données. Le schéma logique le plus utilisé est le schéma en étoile (cf. supra). Par contre, le modèle en flocon est préconisé quand les tables de dimensions contiennent un nombre important de données.

- **Approche MOLAP (Multidimensional On-Line Analytical Processing).**

Cette approche permet de stocker les données de manière multidimensionnelle. Les temps d'accès sont de la sorte optimisés. Cependant, cette approche nécessite de redéfinir des opérations pour manipuler les structures multidimensionnelles.

## 1.4 Client OLAP

Le client OLAP offre une interface utilisateur avec des outils de reporting, d'analyse interactive et parfois de fouille de données. Son rôle est de rendre « visible » l'information multidimensionnelle, c'est-à-dire de permettre de mettre en exergue des connaissances grâce à la seule visualisation et l'interaction avec les données. Parfois, client et serveur OLAP sont intégrés. (Stolte *et al.* 2003) définissent les caractéristiques indispensables de cet outil :

- 1) Affichage de grandes quantités de données : nombreuses mesures et dimensions ;
- 2) Présence de multiples types d'affichages graphiques : histogrammes pour les ventes ou des couleurs pour les produits les plus vendus, etc.
- 3) Emploi d'une interface pour l'exploration : comparer les données et changer rapidement les données et leur mode de visualisation dans une interface familière et intuitive.

## 1.5 Conclusions

Ce chapitre nous a permis de faire un tour d'horizon des principaux concepts relatifs d'une part aux entrepôts de données et d'autre part aux systèmes OLAP. Nous avons également présenté les notions sur lesquelles l'analyse multidimensionnelle repose. Par ailleurs, les trois niveaux de l'architecture classique des entrepôts de données ont été explicités ainsi que les caractéristiques essentielles des modèles formels des bases de données multidimensionnelles. Par contre, cette analyse fait ressortir par défaut l'incapacité des systèmes OLAP à gérer l'information spatiale ; donc la nécessité de concevoir de nouveaux concepts permettant l'intégration de l'information spatiale dans l'OLAP.

C'est dans cette perspective que se situent les travaux que nous examinerons dans les chapitres suivants en commençant dans un premier temps par la présentation des systèmes d'information géographique.

## **Chapitre 2 : Système d'Information Géographique (SIG) et Géomatique**

La géomatique, née dans les années 80, a vu au cours du temps sa définition et ses objectifs évolués, depuis les propositions de (Burrough 1986). «Un SIG est un équipement instrumental puissant pour l'acquisition, le traitement, la transformation et la représentation de données localisées relatives au monde réel ».

(Paegelow 2004) effectuant une synthèse des définitions successivement proposées entre autres par (Tomlinson 1987) (Cowen 1988) (Goodchild 1988) (Clarke 1990) (Didier 1990) (Hewlett Packard 1993) et (Joliveau 1996). (Pornon 2007), en proposa la formulation suivante: On peut qualifier le SIG comme un ensemble de matériels (hardware) et de logiciels (software), de plus en plus imbriqués dans des réseaux de données et de personnels qualifiés, capable d'acquérir, de stocker, de traiter, d'analyser, de modéliser, de représenter des données géoréférencées numériques, et utiles dans de nombreux domaines d'application manipulant l'information spatialisée où cet outil apporte une aide à la prise de décision ».

L'objectif de la géomatique est donc d'intégrer les moyens d'acquisition de gestion et de diffusion de données à références spatiales en vue d'aboutir à une information d'aide à la décision. Ces données sont appelées indifféremment « données spatiales », « données géospaciales » ou « données géographiques » (Yazidi Alaoui *et al.* 2014).

### **2.1 L'espace géographique**

L'espace géographique est défini par (Pumain et Saint-Julien 1997) comme l'ensemble des lieux et des relations entre les lieux déterminés par les acteurs sociaux localisés. L'espace géographique peut être conçu et représenté selon (plumejeaud 2011) soit :

- comme un simple contenant, repère immuable dans lequel on situe les objets et analyse leurs relations : c'est l'espace support, ou espace absolu. Il est isotrope et homogène, c'est-à-dire qu'il a les mêmes propriétés dans toutes les directions,

- comme un ensemble de relations, dont les propriétés variables, dans le temps et dans l'espace sont définies par la nature et la forme des interactions entre les objets et les unités spatiales : c'est l'espace relatif ou espace produit.

## **2.2 Les données géographiques**

L'information géographique est caractérisée par une composante purement spatiale associée à une composante sémantique (Denègre et Salgé 1996).

### **2.2.1 Composante spatiale**

Une information géographique : relative à un objet ou un phénomène réel, définis par des données numériques, textuelles ou visuelles, doit être positionnée dans un référentiel spatial, par exemple de type latitude/longitude : UTM (Universal Transvers Mercator) ou encore GPS (Global Position System) permettant de calculer sa position sur le globe.

### **2.2.2 Composante sémantique**

La composante sémantique représente l'information relative à la nature, l'aspect, et les propriétés descriptives d'un objet ou un phénomène du monde terrestre, par exemple une ville est décrite par son nom (Rissani) et sa population (20 469) et la date du recensement (2004), etc. Cette information peut aussi inclure des relations avec d'autres objets par exemple, la ville de Rissani appartient à la région d'Er Rachidia. Un des aspects sémantiques caractéristiques des données spatiales est leur représentation multiple à différentes échelles nommées généralisation.

### **2.2.3 Échelle spatiale**

L'échelle spatiale est définie par la modélisation et l'analyse selon plusieurs niveaux de granularité. Il n'existe pas, pour l'espace, de grain élémentaire consensuel ou naturel (Lardon *et al.* 1999). L'échelle spatiale est un ensemble ordonné de grandeurs spatiales. Cette échelle peut être comprise soit comme l'échelle de représentation, soit comme le niveau d'observation et d'analyse. Concernant l'échelle de représentation, un nombre est utilisé afin d'exprimer le

rapport entre une distance sur la représentation et une distance réelle, dans la même unité de distance. Ainsi, dire qu'une carte est au 25 000 millièmes signifie qu'un centimètre sur la carte représente 25 000 centimètres dans la réalité. Par contre, l'échelle d'observation (ou niveaux d'analyse) sous-tend l'usage d'une structuration de l'espace en niveaux hiérarchiques :

- Le niveau micro : la maison, la rue, le quartier.
- le niveau méso : le canton, le pays, la région, l'état.
- le niveau macro : cadre national, supranational (comme l'Afrique) et mondial.

## **2.2.4 Analyse spatiale**

Le principe de l'analyse spatiale énonce que les caractéristiques d'une entité dépendent des relations de proximité de cette entité par rapport à d'autres entités (Pumain et Saint-Julien 1997). La proximité est évaluée par la distance. La distance peut être évaluée de manière quantitative ou qualitative. De manière quantitative, la distance s'exprime par une métrique par exemple le kilomètre ; de manière qualitative, la distance est évaluée par la topologie, mais le milieu naturel est anisotrope ; la distance quantitative entre deux villes séparées par une montagne est plus courte si elle est exprimée à « vol d'oiseau » que par la mesure du tracé d'une route traversant le relief ou le contournant. La distance qualitative évoque l'écartement entre deux entités selon qu'elles sont « proches » ou « éloignées ».

### **2.2.4.1 Topologie, concept de base**

Les relations géométriques de proximité entre des entités peuvent être décrites de façons quantitatives par une métrique : la distance, et de manière qualitative par la topologie.

La topologie est un élément fondamental de l'analyse spatiale. Sans elle, il serait impossible d'extraire d'une base de données des informations sur le voisinage d'une entité, sur les branchements d'un réseau de canalisation, d'un réseau ferroviaire ou encore sur le sens d'écoulement de l'eau dans une rivière.

Selon (Schneuwly et Caloz 2013) la topologie est l'ensemble des relations perçues permettant de situer les objets les uns par rapport aux autres. Le

« voisinage » est donc une notion spatiale « qu'est-ce qui est à côté de quoi ». Pour les réseaux, la question devient « qu'est-ce qui est connecté à ? »

Dans le cadre des SIG, on fait appel à la topologie pour rendre compte de façon sommaire, mais synthétique de la proximité entre les entités. Les relations topologiques exploitées dans ce contexte sont l'adjacence, la connectivité, l'inclusion et l'intersection, la connexité.

Les situations « a » « b » « c » (Figure 12) sont équivalentes du point de vue topologique.

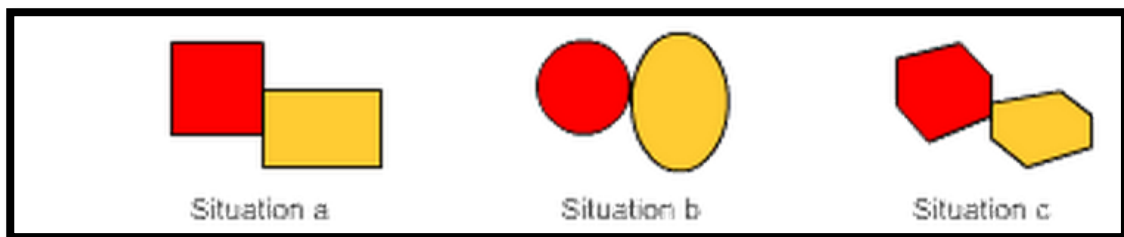


Figure 12 : Topologie : exemple de situations équivalentes du point de vue topologique

### ▪ L'adjacence

L'adjacence répond à la question « qu'est-ce qui est à côté de quoi ? ». La notion d'adjacence (ou contiguïté ou voisinage), implique que les unités spatiales possèdent en commun un côté : (adjacence stricte), ou un sommet : (adjacence large) comme l'illustre la figure 13. Dénuée de métrique, la topologie prend en compte seulement l'ordre dans lequel des unités spatiales se rencontrent en s'éloignant de l'unité de référence. On définit ainsi une adjacence de : 1er ordre si les deux entités sont en contact ; de 2ème ordre si une entité s'intercale et ainsi de suite.

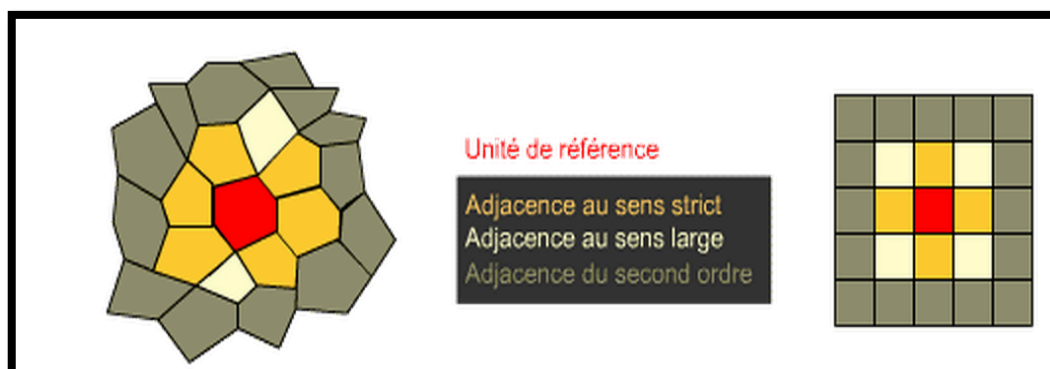


Figure 13 : Adjacence

- **La connectivité**

La connectivité exprime l'adjacence pour des réseaux linéaires. Il s'agit alors de lier les différents segments constitutifs du réseau tout en leur attribuant ou non une orientation. Cette notion est en particulier utilisée pour décrire les réseaux de distribution (eau, électricité, gaz).

- **L'inclusion**

L'inclusion se rencontre lorsqu'une unité spatiale est totalement située à l'intérieur d'une autre, telle une enclave, les départements ou provinces à l'intérieur d'un pays. En termes topologiques il s'agit d'un cas particulier d'adjacence.

- **L'intersection**

L'intersection exprime le point ou une surface commune à deux entités.

- **La connexité**

La connexité formalise le concept d'« être à un seul tenant ».

La connexité est décrite au moyen de graphe et de matrice pour indiquer au moment de la conception du système, ce qui est relié à quoi.

- **L'orientation**

L'orientation permet de définir la position d'un objet par rapport à un autre à l'aide de huit valeurs linguistiques : nord, nord-est, est, sud-est, sud, sud-ouest, ouest, nord-ouest. La sémantique de chaque valeur est donnée par une fonction d'appartenance trapézoïdale, sur l'axe des angles.

## **2.2.5 Les relations temporelles**

Quelle que soit la nature d'un phénomène se produisant sur un espace géographique, son analyse doit s'effectuer dans un espace de temps déterminé.

Dans cette perspective, la représentation du temps peut être réalisée suivant deux approches ; l'une quantitative, « temps mesuré et positionné avec sa

coordonnée sur l'axe temporel», l'autre qualitative (temps représenté par des évènements ordonnés en fonction de leurs situations relatives, c'est-à-dire topologiques). Ainsi le temps peut être représenté comme :

- un flux mesurable et quantifiable par des dates exprimées sur une échelle de granularité en fonction de la précision recherchée (siècles, années, mois, jours, heures, minutes, etc.),
- une succession d'évènements exprimée sur une échelle ordinale (quelque chose se passe avant, pendant ou après autre chose).

Concernant la structure ordinale du temps, les travaux de Allen (Allen 1983) fondent une algèbre temporelle permettant de définir les relations entre objets datés.

Il s'agit d'opérateurs booléens de base pour le positionnement relatif dans le temps ; « précède, succède, égale, rencontre, chevauche, pendant, commence et finit et leur réciproque ».

Treize relations de base, d'Allen sont illustrées (Figure 14). Ce tableau montre toutes les relations possibles que deux intervalles définis peuvent avoir. Chacun est défini graphiquement par un schéma relatif à deux intervalles définis : a et b, avec le temps presse → de gauche à droite. Par exemple, le premier diagramme montre qu'« a précède b » signifie qu'une extrémité avant b commence, avec un écart qui les sépare, la seconde montre que « a rencontre b » signifie que b se termine quand a commence.

précède	répond	chevauchements	terminé par	contient	départs	égal	commencé par	pendant	finitions	conteneurs superposés ped par	rencontré par	précédée par
p	m	o	Fa	Ré	s	e	S	ré	fa	O	M	P

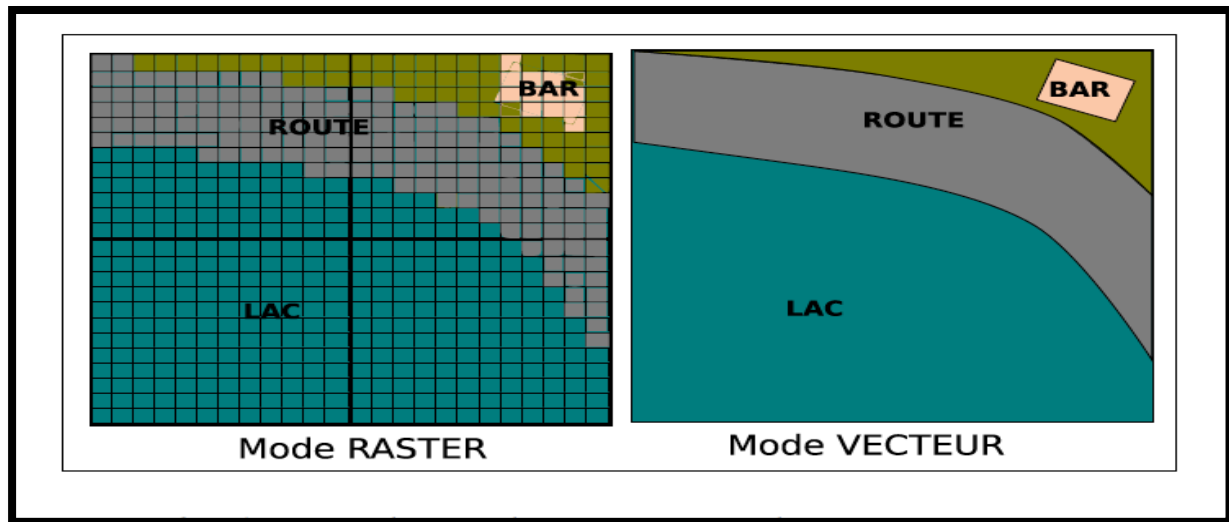
Figure 14 : Relation entre intervalles temporels. Source : (Allen 1983)

Ces relations sont l'équivalent temporel des relations topologiques appliquées au domaine spatial.

## 2.3 Fonctionnalités des Systèmes d'Information Géographique

Dans les SIG les données sont représentées suivant deux types de structure et sont organisées suivant la métaphore des couches.

Les deux modes de représentations : matriciel ou « raster » et vectoriel sont illustrés par la figure 15.



*Figure 15 : Système de coordonnées géographiques pour localiser une entité.  
Source : (Plumejeaud 2011)*

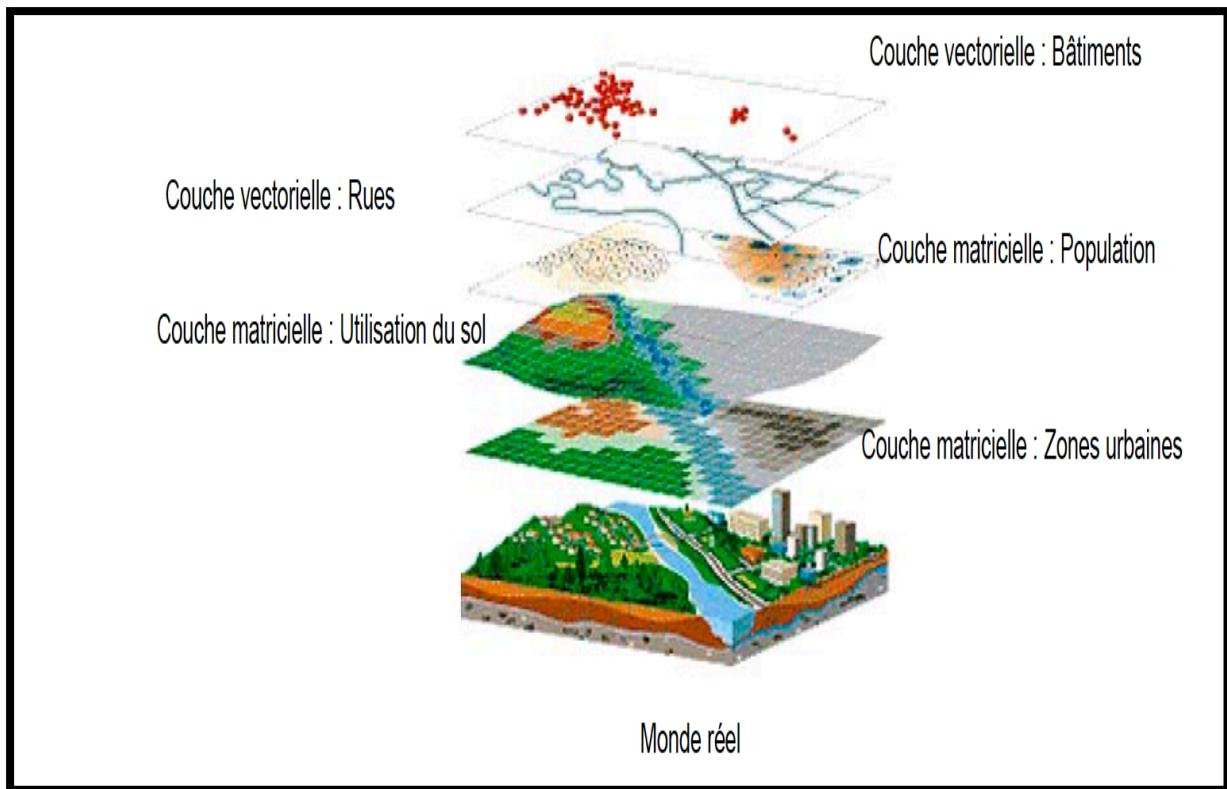
En mode « raster », une surface est représentée par un ensemble de cellules, également appelées pixels dotés d'une couleur et une luminosité donnée. Chaque pixel représente une surface, à laquelle des attributs thématiques sont rattachés. Le modèle matriciel sert à représenter aussi bien des variables qualitatives que quantitatives, avec ce type de représentation, la résolution dépend de la taille de la cellule et la topologie est toujours implicite. L'avantage est que les opérations sur les surfaces sont aisées (calcul d'aire, superposition de couches).

En mode vectoriel, la surface contient des objets géographiques, qui sont définis chacun par leur description géométrique ; lignes, polygones (formes géométriques élémentaires) ou une combinaison de ces formes élémentaires. À chaque entité représentée est attaché un ensemble d'attributs descriptifs : des variables qualitatives ou bien quantitatives, comme en mode raster. Les objets peuvent être observés à différents niveaux de granularité (ou échelles), et cette

granularité affecte le type de la géométrie utilisée pour représenter la position de l'objet. En effet, de loin, une ville peut se résumer à un point, mais en zoomant, à plus grande échelle, les formes de la ville se dessinent (elle occupe alors une surface non nulle). Elle est alors mieux représentée par un polygone. Il peut donc exister plusieurs représentations d'un même objet géographique, qui dépendent à la fois de l'usage souhaité et du niveau de généralisation de l'objet (Ruas 1999), mais aussi du niveau d'analyse de l'espace considéré (Mathian et Piron 2001).

Le modèle raster est particulièrement utilisé comme fond de carte, car il réussit à communiquer beaucoup d'informations (Longley *et al.* 2005). L'avantage du modèle vectoriel est qu'il est plus adapté à l'abstraction et au raisonnement sur les positions relatives des objets.

L'organisation en couche des données géographiques est illustrée (Figure 16). Une couche est une carte qui représente un ensemble d'information géographique de même nature (routes, bâtiments, limites de communes, types de sol, types d'équipements, etc.). Cette structuration permet de relier et de croiser donc d'analyser différentes couches thématiques indépendantes, sous forme de superpositions d'informations géographiques.



*Figure 16 : Organisation des informations géographiques en couches thématiques.  
Source : (Bimonte 2007)*

## 2.4 Conclusions

Un système d'information géographique (SIG) est un système d'information conçu pour recueillir, stocker, traiter, analyser, gérer et visualiser tous les types de données spatiales et géographiques.

La caractéristique primordiale des SIG tient au fait que les données numériques, textuelles ou visuelles sont géoréférencées permettant ainsi d'enrichir le pouvoir expressif des cartes.

L'intégration de l'information géographique au sein d'un système d'aide à la décision de type OLAP, implique une redéfinition des propriétés fondamentales de ces derniers, en termes de modèles d'analyse et de techniques de visualisation et d'interaction.



## Chapitre 3 : le concept SOLAP ou OLAP spatial

Solution efficace pour l'analyse spatiomultidimensionnelle de données spatiales, les Entrepôts de Données Spatiales (EDS) permettent d'intégrer, d'organiser de façon multidimensionnelle, de stocker et d'historier de très gros volumes de données spatiales et non spatiales provenant de multiples sources pour supporter le processus de prise de décision au sein d'une organisation (Stefanovic *et al.* 2000).

Les systèmes SOLAP sont une catégorie d'outils logiciels qui permettent l'exploration interactive basée sur approche spatiomultidimensionnelle à plusieurs niveaux de détail des EDS (Bédard *et al.* 2007). Ces systèmes accroissent les capacités d'analyse et de représentation des systèmes OLAP par de nouveaux concepts/opérateurs (concepts/opérateurs SOLAP) qui permettent d'enrichir l'analyse OLAP par des analyses spatiales. En plus de la navigation spatiomultidimensionnelle par l'intermédiaire d'opérateurs spécifiques au SOLAP (p. ex. Drill-Down spatial), ces systèmes permettent également la visualisation des résultats de requêtes décisionnelles sous forme de cartes interactives, ce qui permet une meilleure compréhension du phénomène analysé.

Les EDS et les systèmes SOLAP se basent sur le modèle spatiomultidimensionnel. Ce modèle modifie le modèle multidimensionnel classique par l'adjonction d'autres concepts, tels que la mesure et la dimension spatiale qui permettent ainsi l'intégration de l'information spatiale dans l'analyse multidimensionnelle.

Dans ce chapitre, nous présenterons en premier lieu les caractéristiques essentielles des systèmes d'aides à la décision spatiale ; puis nous décrirons les principaux concepts des entrepôts de données spatiales et des SOLAP en focalisant notre propos sur les notions de dimensions et de mesures spatiales et sur la typologie des différents SOLAP.

### **3.1 Origines des systèmes d'aide à la décision spatiale**

Selon (Franklin 1992) les données transactionnelles ayant une référence spatiale représentent 80 % des données accumulées dans les différentes bases de données d'une entreprise. Or la composante spatiale, de ces données, reste souvent inutilisée dans les processus décisionnels des technologies OLAP dépourvues de visualisation et de navigation cartographique (Caron 1998) in (Bédard *et al.* 2005). Ceci limite donc considérablement l'analyse des phénomènes géographiques et spatio-temporels.

De leur côté, les SIG ne sont pas par essence d'authentiques SAD (Keenan 1996). En effet, bien que possédant des fonctionnalités avancées d'analyse, ils sont dépourvus d'une interface simple et intuitive pour visualiser et rechercher les données. De plus, ils ne sont pas conçus pour fournir une vision agrégée des données.

Afin de combler cette lacune, une nouvelle catégorie de Système d'Aide à la Décision Spatiale (SADS) a été élaborée dont la plus connue est l'OLAP Spatial ou SOLAP.

#### **3.1.1 Définitions et caractéristiques de la technologie SOLAP**

Le SOLAP est caractérisé par le couplage des techniques d'analyse de l'OLAP à celles des systèmes d'information géographique (SIG). Ce couplage augmente donc les capacités d'analyse des systèmes OLAP. Cela implique une reformulation des concepts des entrepôts de données et de l'analyse en ligne d'un point de vue formel et de l'implémentation (Bédard *et al.* 2005) (Bédard *et al.* 2001) (Fidalgo *et al.* 2004) (Pestana *et al.* 2005); mais on peut dire aussi, selon Armstrong (Armstrong et Densham 1990) et Goodchild (Goodchild et Densham 1990), que le SOLAP accroît les capacités des SIG (gestion de la donnée spatiale, affichage cartographique, outils d'analyse spatiale) en apportant des modèles d'analyse, des interfaces simples et en gérant des structures complexes de données spatiales.

La vocation du SOLAP consiste à assister les décideurs afin d'élucider des problèmes et à prendre des décisions dans les situations où la composante spatiale joue un rôle primordial comme le géomarketing, l'environnement, la téléphonie mobile, etc.

La technologie SOLAP est définie par (Bédard 2004) comme « un type de logiciel qui permet la navigation rapide et facile dans les bases de données spatiales, qui offre plusieurs niveaux de granularité d'information, plusieurs thèmes, plusieurs époques et plusieurs modes d'affichage synchronisés, ou non : cartes, tableaux et diagrammes ».

Cet auteur précise par ailleurs : « que la technologie SOLAP supporte la structure multidimensionnelle telle qu'utilisée en informatique décisionnelle, ce qui lui confère un immense avantage sur les logiciels de déploiement d'application cartographique sur le Web, car ces derniers sont basés sur une structure transactionnelle. La technologie SOLAP offre aussi de nouvelles fonctions d'aide à la décision, non disponibles dans les SIG traditionnels ni dans les outils OLAP.

La technologie SOLAP permet la visualisation cartographique des données, la navigation cartographique dans la carte elle-même ou dans les symboles affichés sur cette carte et ceci selon différents types de forage. Dans une technologie SOLAP la création des cartes résultantes des analyses est dynamique, contrairement à certains logiciels de visualisation OLAP où chacune des opérations de navigation spatiale (par exemple : forage) doit être prédéfinie dans l'application et associée à une carte. Cette limitation complexifie la mise à jour des données géométriques en répartissant l'information sur plusieurs cartes. De plus, un tel outil SOLAP gère adéquatement les règles de représentation cartographique des résultats des analyses sur les cartes. Par conséquent, l'utilisation d'un tel outil ne nécessite pas le support d'un expert en cartographie même s'il permet à l'utilisateur de créer des centaines de milliers de cartes différentes par quelques clics de souris. Dans la présentation des résultats, la technologie SOLAP utilise les mêmes règles sémiologiques (par exemple : couleur, trame, contour) pour l'ensemble des affichages, c.-à-d. tableaux, graphiques et cartes. Cela permet d'avoir une synchronisation visuelle entre les différents modes de présentation de l'information et d'avoir un panorama homogène. La sémiologie graphique utilisée pour les différents types d'affichage (c.-à-d. tableaux, graphiques et cartes) demeure synchronisée lors d'un forage ou lors d'autres opérations, conservant ainsi une continuité perceptuelle nécessaire à la découverte de corrélations ».

### 3.2 Principaux concepts du SOLAP. Modélisation logique

En paraphrasant Imnon (Imnon 1994) ; Stefanovic (Stefanovic *et al.* 2000) définissent cet entrepôt comme un ensemble de données spatiales et thématiques intégrées, non volatiles et historisés pour la prise de décisions spatiales.

Le modèle multidimensionnel spatial contrairement au modèle conventionnel comporte à côté de faits et de dimensions non spatiales des dimensions spatiales de différents types, des mesures spatiales et des opérations spatiales que nous décrivons succinctement. Pour une meilleure compréhension de la définition du modèle spatio — multidimensionnel, nous emprunterons à Boulil (Boulil 2014), un exemple d'analyse de ventes auquel sont associés des attributs et relation spatiale topologique modélisée en utilisant les notations graphiques du modèle conceptuel spatiomultidimensionnel : Spatial MultiDimER (Malinowski et Zimányi 2008) (Figure 17 ).

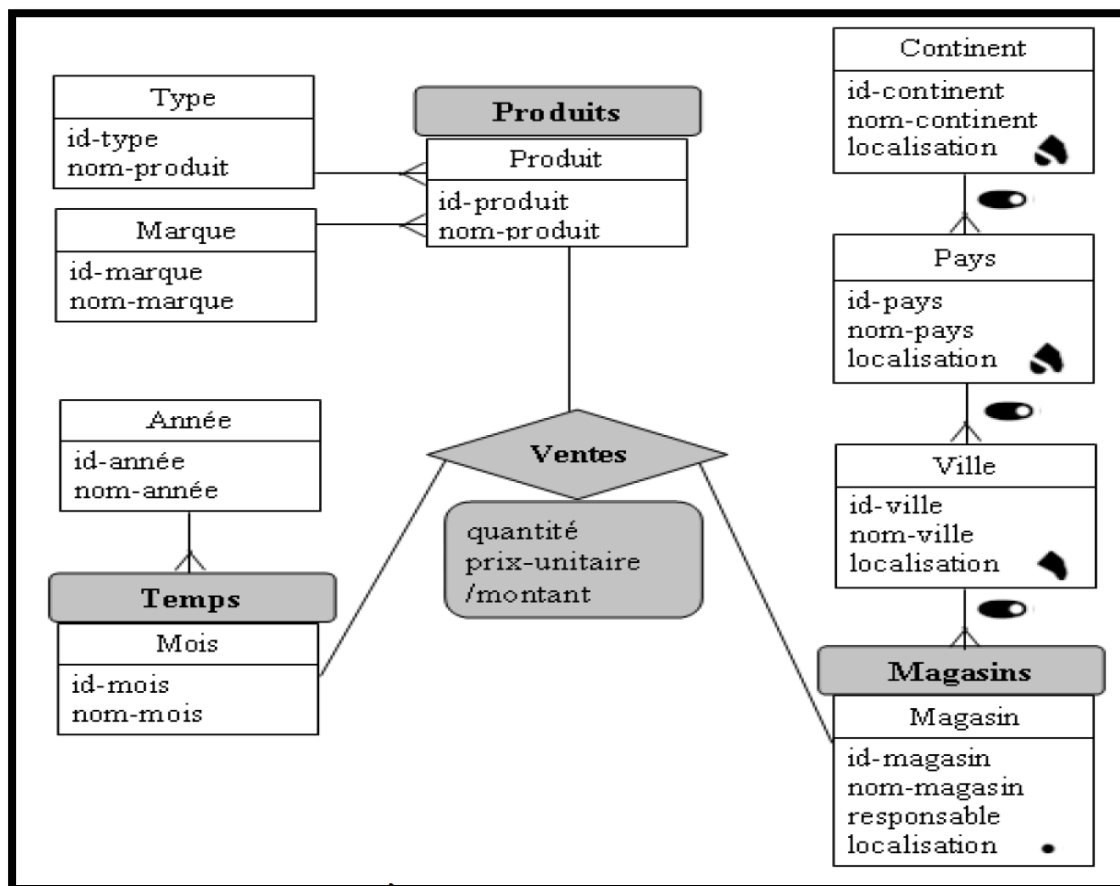


Figure 17 : Exemple de modèle logique spatiomultidimensionnel. Source : (Boulil 2014)

### 3.2.1 Dimensions spatiales

La dimension spatiale permet la représentation en axe d'analyse de l'information spatiale relative aux localisations géographiques et non géographiques des faits. OÙ formulé d'une autre manière : la dimension spatiale consiste à utiliser l'information spatiale comme axe d'analyse dans une application décisionnelle. Plusieurs définitions ont été proposées :

#### 3.2.1.1 les différentes démarches

Le concept de dimension spatiale supporte 3 types de dimensions spatiales : descriptives, géométriques et mixtes en plus des dimensions non spatiales (Bédard *et al.* 2001) et (Rivest *et al.* 2003) :

- 1) La dimension « descriptive » utilise la référence spatiale nominale seulement (par exemple les noms des lieux) et aucune représentation cartographique n'est associée aux membres de la dimension. Ce type de dimension spatiale est couramment utilisé dans les outils OLAP conventionnels.
- 2) La dimension « géométrique » quand chaque niveau hiérarchique comprend un ensemble de formes géométriques (polygones, points, lignes, etc.).
- 3) La dimension « mixte » quand des niveaux à composante spatiale sont associés à des niveaux purement textuels.

Un exemple des dimensions spatiales descriptive, géométrique et mixte est proposé (Figure 18) afin de représenter la subdivision administrative du Maroc (par exemple pays, région, ville).

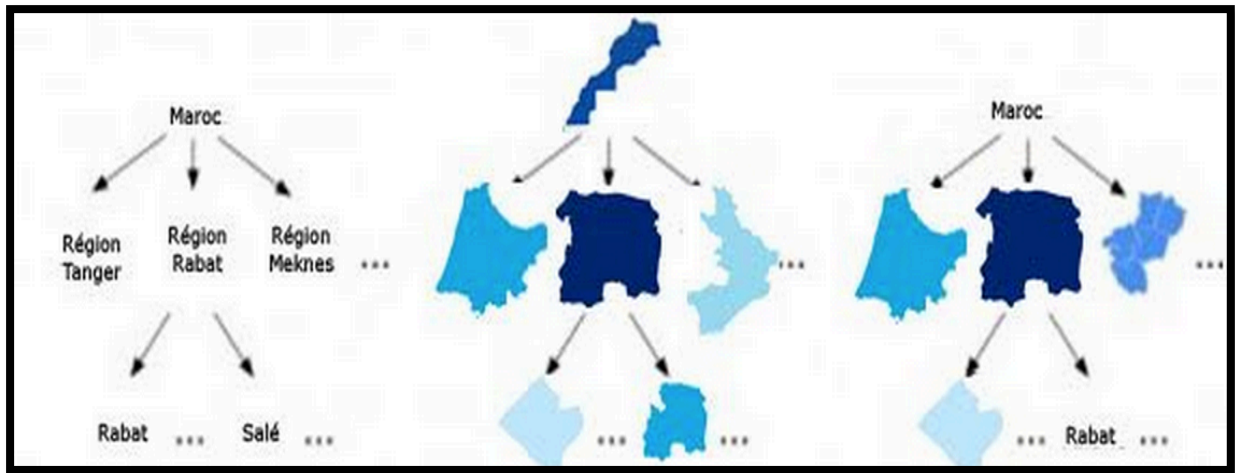


Figure 18 : Les trois modes de représentation des dimensions spatiales supportées par le SOLAP.

Le concept de dimension spatiale vu comme ensemble de hiérarchies spatiales est proposé par (Malinowski et Zimányi 2005). Une hiérarchie est spatiale si au moins l'un de ses niveaux comporte une composante spatiale. Ce niveau est qualifié niveau spatial. De plus, entre les membres de deux niveaux spatiaux doit exister une relation topologique d'inclusion ou d'intersection.

Une hiérarchie spatiale peut être totalement spatiale si tous les niveaux sont spatiaux, partiellement spatiale s'il y a au moins un niveau partiellement spatial.

Une autre vision est proposée par Fidalgo (Fidalgo *et al.* 2004). Ces auteurs distinguent deux types de dimension : géographique et hybride. La dimension géographique contient uniquement des niveaux spatiaux. Une dimension est hybride si elle comporte simultanément des échelons spatiaux et alphanumériques. Par ailleurs en tenant compte de la présence ou non d'attributs descriptifs dans un même échelon, ces auteurs proposent une sous-classification de ces dimensions : primitives dans le cas où seuls les attributs spatiaux sont présents, sinon composés si des attributs alphanumériques sont compris.

### 3.2.1.2 Conclusion partielle

Pour conclure en paraphrasant Jansen (Jansen *et al.* 2004) et Malinowski (Malinowski et Zimányi 2005) ; nous dirons que l'introduction des relations topologiques dans les processus d'agrégation de l'OLAP permet l'utilisation d'attributs spatiaux dans les perspectives d'analyse (dimensions).

Cette composante spatiale de l'information géographique permet donc d'enrichir l'analyse multidimensionnelle en ajoutant une composante cartographique aux perspectives d'analyse et en conséquence de bénéficier du pouvoir métaphorique et d'analyse des cartes.

### 3.2.2 Mesures spatiales

Dans un contexte d'OLAP spatial, les dimensions spatiales ne sont pas les seules à posséder une composante géométrique ; les mesures sont également concernées. La mesure spatiale permet la représentation de l'information spatiale relative aux localisations géographiques et non géographiques des faits dans les cellules de l'hypercube. Ainsi, en plus des mesures conventionnelles supportées dans les systèmes OLAP, il existe les mesures spatiales (Rivest *et al.* 2001).

Dans la littérature, de nombreuses définitions explicitent le concept de mesure spatiale. Par exemple :

- Un ensemble de pointeurs vers des objets spatiaux (stockés dans le cube de données) vers la géométrie d'un objet spatial stockée dans une autre structure que la structure multidimensionnelle. C'est la méthode utilisée par les technologies SIG pour gérer la composante géométrique des objets spatiaux. (Han *et al.* 1998) ; (Stefanovic *et al.* 2000).
- Le résultat d'opérations spatiales topologiques ou métriques par exemple la surface d'une région géographique, la distance entre régions spatiales, la distance minimale avec l'objet le plus proche, le cumul de longueurs sur un réseau, etc. métriques (Han *et al.* 1998) ; (Bédard *et al.* 2005), (Malinowski et Zimányi 2008)
- Une mesure, dont le type de données est spatial (géométrique) par exemple, dans le domaine de la téléphonie mobile : la zone couverture des émetteurs — récepteurs BTS (Base Transceiver

Station). (Bédard *et al.* 2005), (Damiani et Spaccapietra 2006) (Malinowski et Zimányi 2008).

- Une mesure complexe ayant des attributs alphanumériques descriptifs et une géométrie (Bimonte *et al.* 2005).

Pour finir, citons une définition proposée par (Boulil 2014) : « Une mesure spatiale est une mesure dont le type de données est spatial (c.-à-d. géométrique) et dont la représentation cartographique représente un intérêt pour l’analyse décisionnelle ».

Pour une meilleure compréhension de la définition de la mesure spatiale, nous emprunterons à la littérature deux exemples de mesures spatiales. Ainsi (Bimonte 2007), distingue deux types de modèles :

- la mesure spatiale vue comme une liste d’objets spatiaux.
- La mesure spatiale vue comme le résultat des opérateurs spatiaux.

1) le premier modèle est illustré (Figure 19) à l’aide d’une étude multidimensionnelle permettant d’analyser la localisation de phénomènes météorologiques en fonction du temps, des régions et de la température (Stefanovic *et al.* 2000).

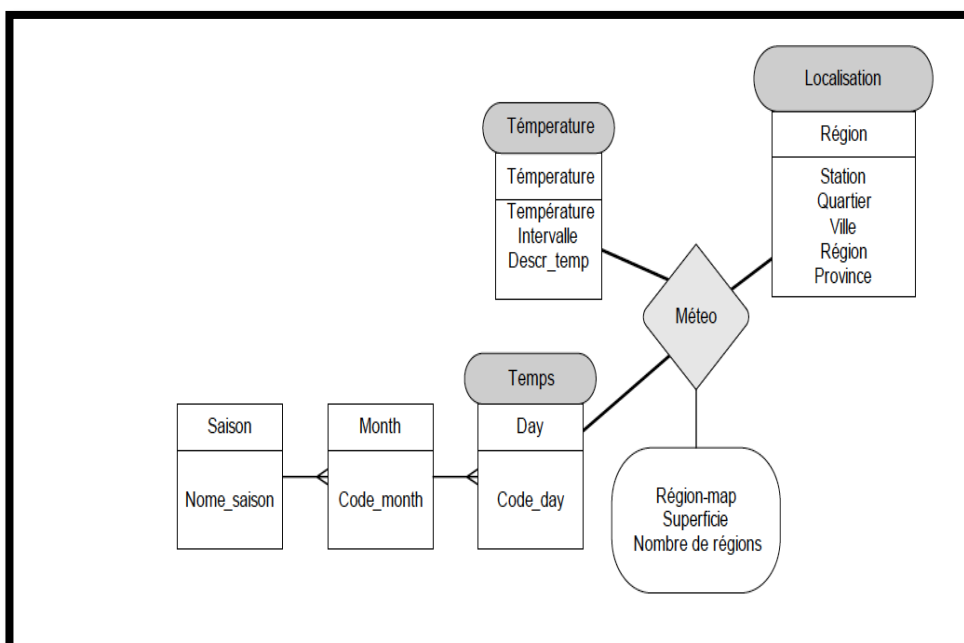


Figure 19 : Schéma de l’entrepôt « Météo ». Source : (Stefanovic *et al.* 2000)

Dans ce cas de figure, la mesure est un objet spatial défini par sa fraction géométrique et par des attributs numériques qui en découlent à savoir la superficie et le nombre de régions. Afin d'intégrer aux mesures spatiales leurs caractéristiques alphanumériques (nom de la région où se situe la station météo) etc., les auteurs dupliquent la mesure spatiale dans le niveau le plus détaillé de la dimension spatiale. De ce fait, les attributs de la dimension représentent les informations alphanumériques des mesures spatiales (détaillées ou agrégées).

2) le deuxième modèle (Figure 20) est illustré par une étude relative aux coûts de réparation des autoroutes (Malinowski et Zimányi 2004).

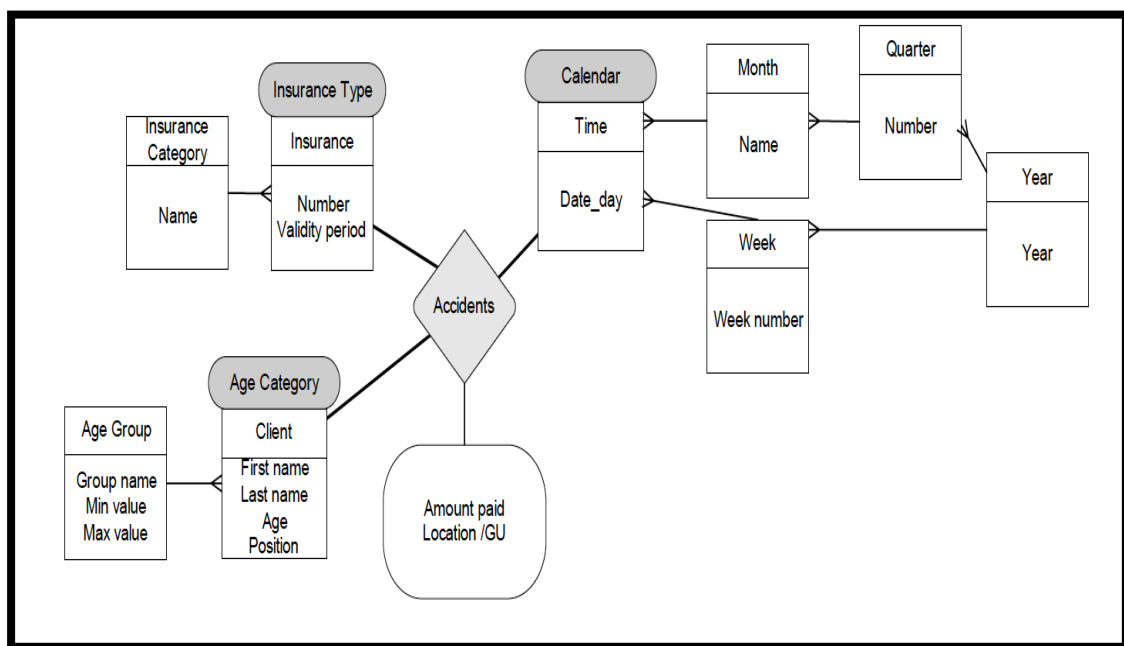


Figure 20 : Modèle spatiomultidimensionnel avec des dimensions alphanumériques et une mesure spatiale. Source : (Malinowski et Zimányi 2004)

Dans ce cas de figure, « La table de fait définit une jointure spatiale entre les dimensions spatiales. Elle permet d'avoir une vision multidimensionnelle des relations spatiales. Elle permet le calcul des mesures spatiales pour les membres de tous les niveaux des hiérarchies des dimensions » (Bimonte 2007).

### 3.2.3 Niveau d'agrégation spatial

Un niveau d'agrégation spatial est défini comme un niveau qui contient un attribut spatial (c.-à-d. dont le type est géométrique) permettant la représentation cartographique de ses membres de dimension (Han *et al.* 1998) ;

(Bimonte 2007) ; (Malinowski et Zimányi 2008) ; (Rivest *et al.* 2005) (F Pinet et Schneider 2010).

Les niveaux d'agrégation spatiaux sont classés selon le type de représentation de leur spatialité en :

- Géométriques (la spatialité des membres est représentée par une géométrie).
- Non géométriques (représentation textuelle).

### 3.2.4 Opérateur SOLAP

L'intégration de l'information spatiale en tant que dimension d'analyse mène à la définition d'opérateurs spécifiques au SOLAP pour l'exploration des hypercubes de données spatiales.

Ces opérateurs SOLAP reformulent les opérateurs OLAP traditionnels.

Les principaux opérateurs SOLAP ont été définis par (Bédart *et al.* 2007) (Ruiz et Times 2009) de la manière suivante :

**Roll-UP spatial** permet la navigation dans une hiérarchie de dimension d'un niveau d'agrégation spatial vers un autre niveau d'agrégation spatial moins détaillé.

**Drill Down spatial**: permet la navigation dans une hiérarchie de dimension d'un niveau d'agrégation spatial vers un autre niveau plus détaillé d'agrégations spatiales.

**Slice spatial** : cet opérateur permet de sélectionner un sous-ensemble des cellules de l'hypercube spatial en appliquant un prédicat spatial (métrique, d'ordre ou topologique) sur les membres d'une dimension spatiale.

**Dice spatial** : cet opérateur permet de sélectionner un sous-ensemble des données en appliquant des prédicats spatiaux à des membres spatiaux de deux dimensions spatiales ou plus.

### 3.2.5 Agrégation dans le SOLAP

Les mesures spatiales sont agrégées en utilisant des fonctions d'agrégation spatiale (union, intersection, etc.).

L'agrégation spatiale peut retourner un objet spatial (géométrie simple) ou un ensemble d'objets spatiaux (une géométrie complexe), selon la connectivité des objets à agréger et de la fonction d'agrégation utilisée. Par exemple, si les objets spatiaux agrégés sont disjoints et si la fonction d'agrégation utilisée est l'union spatiale, alors l'agrégation retournera une géométrie complexe. L'agrégation des mesures dans le SOLAP dépend de plusieurs paramètres :

- de l'additivité de la mesure « stock, flux ou valeur par unité,
- de la structuration thématique de la hiérarchie (type de hiérarchie stricte ou non stricte,
- du type de fonction d'agrégation « applicable à des données pouvant être additionnées, etc. » (Pedersen *et al.* 2001).
- de la distributivité de la fonction d'agrégation,
- des relations spatiales topologiques entre les membres spatiaux « pour une hiérarchie de dimension spatiale » (Pedersen et Tryfona 2001).

Les fonctions d'agrégation numériques ont été classées en fonction des catégories de données auxquelles elles peuvent s'appliquer en trois types :

- applicables aux données pouvant être additionnées (Sum, Count, Avg, Min, Max).
- applicables aux données pouvant être moyennées (Count, Avg, Min, Max).
- applicables aux données pouvant être uniquement comptées (Count) ; (Pedersen *et al.* 2001).

Par rapport aux types de données auxquels les fonctions d'agrégation SOLAP peuvent être appliquées, les catégories suivantes ont été identifiées dans la littérature :

- Fonctions d'agrégation numérique : s'appliquent seulement aux données de type numérique ; par exemple Sum et Avg.
- Fonctions d'agrégation spatiale, s'appliquent seulement aux données de type géométrique (Point, Ligne, Polygone, etc.), par exemple l'union spatiale, le centroïde, et l'équipartition ; (Silva *et al.* 2008), ((Ruiz et Times 2009).

- Fonctions d'agrégation booléenne, s'appliquent seulement aux données booléennes, par exemple, Or et And (Golfarelli *et al.* 1998).
- Fonctions d'agrégation textuelle, s'appliquent seulement aux données textuelles (type String), par exemple, Topic et Top Keywords ; (Park *et al.* 2005), ((Ravat *et al.* 2007) (Ravat *et al.* 2008).
- Fonctions d'agrégation temporelle, s'appliquent aux données de type temporel (Instant ou Intervalle de dates), par exemple l'union temporelle.
- Fonctions d'agrégation génériques : s'appliquent à plusieurs types de données, par exemple Count, Distinct Count et Mode (Ravat *et al.* 2008).

Suivant leur distributivité, les fonctions d'agrégation SOLAP (numérique, spatiale ou autre) sont classées de manière formelle en trois catégories :

- 1) Distributives (Sum, Union spatiale, etc.
- 2) algébriques (Avg, Centroid, etc.)
- 3) Holistiques (général)

Les fonctions distributives permettent de réutiliser les agrégats d'un niveau d'agrégation pour calculer des agrégats corrects à un niveau d'agrégation plus large. Les fonctions algébriques sont des expressions algébriques finies de fonctions distributives. Ces fonctions requièrent donc une manipulation supplémentaire pour pouvoir réutiliser correctement des agrégats. Finalement, les fonctions holistiques qui ne sont ni distributives ni algébriques nécessitent de recalculer le total en utilisant les données à un niveau le plus détaillé.

### **3.2.6 Modélisation conceptuelle des données spatiales**

De nombreux travaux ont été réalisés afin d'adapter le modèle multidimensionnel aux concepts spatiomultidimensionnels.

Les principaux modèles UML (Unified Model Language) proposés sont ceux de Boulil (Boulil 2014) ;Pinet (François Pinet et Schneider 2010) ; Glorio (Glorio et Trujillo 2008).

### 3.2.6.1 Proposition de Glorio et Trujillo

Le déploiement du modèle multidimensionnel proposé permet la création automatique d'un modèle logique de données implantable dans un SGBDS depuis le modèle conceptuel. La mesure spatiale et le niveau d'agrégation spatial sont définis en utilisant des stéréotypes UML sur lesquels sont définies des contraintes OCL (Object Constant Language) qui garantissent leur bon usage. Les stéréotypes :

- « Spatial Measure » caractérisent les mesures spatiales.
- « Spatial Level» représente les niveaux d'agrégation spatiaux.
- Rolls-upTo est défini comme une association UML qui permet de relier les niveaux d'agrégation d'une dimension de manière hiérarchique. Chaque niveau d'agrégation est composé de différents attributs (identifiant, des attributs spatiaux, des attributs descriptifs, etc.).

### 3.2.6.2 Proposition de François Pinet et Schneider

Dans ce cas, la conceptualisation des composantes de l'analyse multidimensionnelle et la définition des contraintes d'intégrité OCL sont conçues à partir d'un profil UML générique (François Pinet et Schneider 2010). Le profil proposé définit :

- les niveaux spatiaux avec des classes identifiées spatiales disposant chacune d'un attribut géométrique.
- les mesures spatiales avec des attributs UML

### 3.2.6.3 Proposition de Boulil

Un Framework basé sur les langages standard UML et OCL permet la représentation conceptuelle des modèles spatiomultidimensionnels et leurs contraintes d'intégrité (Boulil 2014). Le profil UML comporte 3 métamodèles :

- 1- Un métamodèle qui permet de définir la structure spatiomultidimensionnelle des données.
- 2- Un métamodèle d'agrégation qui permet de spécifier les contraintes d'agrégation, ainsi que le modèle d'analyse.
- 3- Un métamodèle de contraintes d'intégrité sur les requêtes SOLAP.

### **3.3. Typologie des approches pour le développement d'applications SOLAP**

La proposition de Boulil permet une meilleure définition des contraintes d'intégrité relatives à la qualité d'analyse SOLAP, ainsi qu'un formalisme plus complet des concepts et caractéristiques de l'analyse multidimensionnelle.

Le système SOLAP repose sur l'intégration des capacités des SIG au système OLAP. Cette extension de l'OLAP avec la composante cartographique et les opérateurs qui permettent de la manipuler autorise la visualisation des membres spatiaux et alphanumériques sur une carte thématique (Bédard *et al.* 2005). Trois familles de solutions sont décrites :

1. Les solutions « OLAP dominant ».
2. Les solutions « SIG dominant ».
3. Les solutions « intégrées ou hybrides » faisant autant appel aux fonctions OLAP que SIG.

Dans chaque catégorie, c'est l'outil dominant qui fait appel aux fonctionnalités du moins dominant. Ainsi, chaque catégorie répond à un besoin d'analyse particulier. Dans la première catégorie, la visualisation cartographique est considérée comme accessoire. Dans la deuxième, ce sont les fonctions OLAP qui sont considérées comme accessoires. Dans la dernière, les deux solutions sont essentielles pour permettre une exploitation totale des deux applications.

#### **3.3.1 L'OLAP dominant**

Ce type de solution intègre la totalité des fonctions OLAP et un sous-ensemble des fonctionnalités des SIG. En général, les fonctionnalités SIG dont fait appel le OLAP dominant sont les fonctions d'affichage, les fonctions de déplacement dans la carte et de changement d'échelle, la sélection d'objets géométriques, ainsi que quelques fonctions de forage (Bédard *et al.* 2005).

L'inconvénient de cette solution dans l'analyse spatiomultidimensionnelle est qu'elle ne fournit pas les outils suffisants pour une analyse spatiale pertinente. En effet, la composante spatiale n'est pas exploitée, ce qui limite la solution OLAP dominant. L'utilisateur a besoin des outils SIG pour pouvoir manipuler les membres spatiaux et les mesures spatiales et ainsi pouvoir les intégrer aux requêtes multidimensionnelles.

Les solutions OLAP dominantes peuvent être réparties sur 2 catégories :

- Celles qui utilisent les cartes statiques.
- Celles qui utilisent des cartes interactives (Bimonte 2007).

### **Polaris**

**Polaris** est un exemple de solutions OLAP dominant qui utilise les cartes statistiques. Cet outil général peut être utilisé pour acquérir d'une part une compréhension initiale de l'entrepôt de données et d'autre part, l'exploiter visuellement en explorant de façon interactive son contenu. Polaris peut être utilisée directement comme un outil d'exploration visuelle (Stolte, Tang et Hanrahan 2002).

### **Pentaho**

**Pentaho** est un produit de Business Intelligence (BI), open source qui permet l'intégration des données, l'utilisation des fonctions OLAP, le reporting et l'exploration des données, tout en mettant à disposition de l'utilisateur une visualisation cartographique des données géographiques.

### **KMapX**

**KMapX** est un progiciel basé sur la technologie MapX de MapInfo qui permet le forage et la visualisation des membres spatiaux des dimensions spatiales du cube SOLAP (Bédard *et al.* 2005).

### **CognosVisualizer**

**CognosVisualizer** exploite la perception visuelle humaine pour permettre aux utilisateurs de visualiser de grandes quantités de données issues de plusieurs dimensions de données alphanumériques ou de données avec des composants géographiques. Entièrement intégré à Cognos Entreprise Reporting et aux solutions de traitement analytique en ligne (OLAP), CognosVisualizer permet aux utilisateurs d'acquérir rapidement et intuitivement la perspicacité fournie par les cartes, les aidants à prendre de meilleures décisions.

### 3.3.2 Le « SIG dominant »

Le serveur OLAP peut être simulé dans une base de données relationnelle grâce à la modélisation en étoile. Ceci permet d'effectuer des agrégations en utilisant des requêtes SQL sur la base de données de manière contrôlée. Cependant, cette solution ne permet pas d'utiliser des opérateurs OLAP tels que le forage et le remontage, ou l'utilisation des concepts OLAP avancés tels que les mesures dérivées, à cause de l'absence du serveur OLAP. Les solutions SIG dominant offrent toutes la panoplie des fonctionnalités SIG et un sous-ensemble des fonctionnalités OLAP. Cette solution couple une base de données relationnelle simulant un serveur OLAP à un logiciel SIG ou à un outil de visualisation de données spatiales (Bédard *et al.* 2005).

#### *CommonGIS*

*CommonGIS* est un exemple de solutions « SIG dominant ». C'est un système puissant qui combine des méthodes de SIG traditionnelles avec des outils innovants pour l'analyse visuelle de données et la prise de décision. CommonGIS peut être utilisé en combinaison avec le logiciel SIG commercial (ESRI ArcGIS, MapInfo, etc.). Ceci est assuré par sa capacité à traiter les données spatiales dans de nombreux formats standard. Cependant, au-delà des différentes techniques de géovisualisation que fournit CommonGIS, celui-ci ne propose aucun affichage tabulaire.

### 3.3.3 La solution « Hybride ou intégrée »

Ce type de solutions intègre les fonctionnalités SIG et OLAP dans un environnement qui permet d'utiliser constamment les composantes spatiales dans le processus d'exploration et d'analyse des données. Ce type de solutions est donc soit le mieux adapté pour l'analyse spatiomultidimensionnelle pertinente.

La solution hybride est idéale pour les applications décisionnelles qui disposent d'un grand nombre de composantes spatiales et qui requièrent l'utilisation des fonctionnalités des SIG (ex. des opérateurs topologiques) dans l'analyse multidimensionnelle.

Pour que la solution hybride soit performante, les fonctionnalités OLAP doivent être adaptées de manière à intégrer les opérateurs spatiaux qui

caractérisent les SIG. L'utilisateur dispose de l'affichage tabulaire et graphique de l'OLAP, couplé à l'affichage cartographique des SIG.

L'exploration multidimensionnelle des données est effectuée de manière transparente, grâce à l'interaction de l'utilisateur avec la carte ; sans qu'il ait besoin de connaître un langage de « requête » particulier.

### **Map4Decision**

**Map4Decision** est la première technologie Web qui intègre complètement les dimensions géospatiales dans un environnement d'aide à la décision et de business intelligence. Cette technologie a l'avantage d'être compatible avec la majorité des systèmes de gestion de bases de données relationnelles, ainsi que la plupart des systèmes d'informations géographiques. Map4Decision dispose d'une interface utilisateur intuitive qui permet aux utilisateurs non techniques d'accéder facilement à leurs données spatiales, de les visualiser et de les analyser. Son affichage peut inclure plusieurs cartes thématiques, des tables et des diagrammes statistiques (graphiques à barres, camemberts, etc.). Map4Decision supporte la structure multidimensionnelle des données, l'affichage cartographique et non cartographique, les dimensions et mesures spatiales, ainsi que les mesures calculées. Il effectue aussi une synchronisation de l'exploration interactive entre les différents types d'affichage pour faciliter l'identification et l'interprétation des données.

## **3.4 Conclusions**

L'introduction de l'information spatiale et de l'information géographique dans l'analyse multidimensionnelle implique une redéfinition des concepts des entrepôts de données et de l'analyse en ligne. La composante spatiale est alors utilisée dans la littérature soit comme axe soit comme sujet de l'analyse sous les termes de dimension spatiale et de mesure spatiale.

En nous appuyant sur la littérature, nous avons décrit dans ce chapitre les différentes approches conceptuelles relatives aux entrepôts de données spatiodimensionnelles et dressé un panorama typologique des SOLAP. Il en découle que les solutions SOLAP hybrides sont les plus aptes à réaliser l'analyse spatiale en ligne.



# Chapitre 4 : La technologie Open Source

## 4.1. Définition open source

Bien plus qu'un simple copyright, la terminologie Open Source (également connue sous l'appellation Logiciel libre) reflète une certaine philosophie. Richard STALLMAN, le père fondateur de la Free Software Foundation a coutume de résumer ce qu'est le Logiciel libre par : « Liberté, Égalité, Fraternité ».

« Open Source » implique bien plus que la simple diffusion du code source. La licence d'un programme « Open-Source » doit correspondre aux critères suivants :

### 4.1.1 Libre redistribution

La licence ne doit pas empêcher de vendre ou de donner le logiciel en tant que composant d'une distribution d'un ensemble contenant des programmes de diverses origines. La licence ne doit pas exiger que cette vente soit soumise à l'acquittement de droits d'auteur ou de royalties (s. d.).

### 4.1.2 Code source

Le programme doit inclure le code source, et la distribution sous forme de code source comme sous forme compilée doit être autorisée. Quand un produit n'est pas distribué avec le code source correspondant, il doit exister un moyen clairement indiqué de télécharger le code source, depuis l'Internet, sans frais supplémentaires. Le code source est la forme la plus adéquate pour qu'un programmeur modifie le programme. Il n'est pas autorisé de proposer un code source volontairement incompréhensible. Il n'est pas autorisé de proposer des formes intermédiaires, comme ce qu'engendre un préprocesseur ou un traducteur automatique (s. d.).

### 4.1.3 Travaux dérivés

La licence doit autoriser les modifications et les travaux dérivés, et leur distribution sous les mêmes conditions que celles qu'autorise la licence du programme original(s. d.).

#### **4.1.4 Intégrité du code source de l'auteur**

La licence ne peut restreindre la redistribution du code source sous forme modifiée sauf si elle autorise la distribution de fichiers « patch » aux côtés du code source dans le but de modifier le programme au moment de sa construction. La licence doit explicitement permettre la distribution de logiciel construit à partir du code source modifié. La licence peut exiger que les travaux dérivés portent un nom différent ou un numéro de version distinct de ceux du logiciel original (s. d.).

#### **4.1.5 Pas de discrimination entre les personnes ou les groupes**

La licence ne doit opérer aucune discrimination à l'encontre de personne ou de groupes de personnes.

#### **4.1.6 Pas de discrimination entre les domaines d'application**

La licence ne doit pas limiter le champ d'application du programme. Par exemple, elle ne doit pas interdire l'utilisation du programme pour faire des affaires ou dans le cadre de la recherche génétique.

#### **4.1.7 Distribution de la licence**

Les droits attachés au programme doivent s'appliquer à tous ceux à qui le programme est redistribué sans que ces parties doivent remplir les conditions d'une licence supplémentaire.

#### **4.1.8 La licence ne doit pas être spécifique à un produit**

Les droits attachés au programme ne doivent pas dépendre du fait que le programme fait partie d'une distribution logicielle spécifique. Si le programme est extrait de cette distribution et utilisé ou distribué selon les conditions de la licence du programme, toutes les parties auxquelles le programme est redistribué doivent bénéficier des droits accordés lorsque le programme est au sein de la distribution originale de logiciels.

### **4.1.9 La licence ne doit pas contaminer d'autres logiciels**

La licence ne doit pas apposer de restriction, sur d'autres logiciels distribués avec le programme qu'elle couvre. Par exemple, la licence ne doit pas exiger que tous les programmes distribués grâce au même médium soient des logiciels « Open Source ».

### **4.1.10 Exemples de licences**

Les licences suivantes sont des exemples de licences que nous considérons conformes à la définition de l'« Open Source » : GNU-GPL, BSD, X Consortiums, et Artistic. C'est aussi le cas de la MPL.

## **4.2 Justification de choix de l'Open Source**

Audaxis a basé sa stratégie de développement sur l'intégration de solutions Open Source, pour les raisons suivantes :

Les licences étant gratuites, le risque d'investissement logiciel, qui représente en général 20 % du budget du projet, est écarté.

100 % de l'investissement est affecté aux services (personnalisation, formation, support) ce qui permet de maximiser les chances de réussite du projet.

L'accès libre aux sources permet de personnaliser librement l'application au métier de l'entreprise et augmente l'adéquation des solutions aux besoins des utilisateurs.

La culture Recherche et Développement des éditeurs de logiciels libres et le modèle collaboratif de développement permettent une évolution rapide et fiable du logiciel Open Source.

Les solutions Open Source sont basées sur des composants objet interopérables.

Les clients sont maîtres de l'évolution de leur système d'information puisqu'il n'y a pas de dépendance vis-à-vis d'un éditeur ni d'obligation à migrer vers de nouvelles versions.

La logique de l'Open Source veut que la pérennité du logiciel soit assurée par l'attrait de la communauté vis-à-vis de l'outil et non par des lois économiques.

La logique de libre utilisation des logiciels apporte de la transparence et leur permet d'arriver plus rapidement à maturité qu'avec un éditeur-proprétaire.

### **4.3 Informatique décisionnelle Open Source**

Avant de discuter de la raison pour laquelle une entreprise peut choisir (OSBI) et mettre en évidence ses avantages et inconvénients, nous en exposerons les caractéristiques (s. d.).

- Comme les autres modèles Open Source, OSBI est axé sur la collectivité.

BI Open Source : est soutenue par une communauté de développeurs qui travaillent en collectivité. Open Source fournit aux développeurs un accès complet (et gratuit) pour le code des logiciels de BI, afin qu'ils puissent créer leurs propres extensions, colmater les bugs, etc.

- C'est un logiciel gratuit

Open Source, offre une « licence libre de coût. » Cela signifie que vous n'aurez pas l'obligation d'acheter une licence de logiciel afin d'utiliser « BI Open Source », mais de nombreux fournisseurs OSBI font payer pour des fonctionnalités, telles que des services de soutien.

- OSBI n'est pas une version gratuite d'une solution commerciale.

Certaines entreprises commerciales de business intelligence, tels que « Microsoft Puissance BI », offrent une version gratuite (mais moins robuste) de leur logiciel de BI « à la vente » ; mais cela ne signifie pas qu'il est « open source ». Cela signifie simplement qu'il est libre. Les développeurs ne peuvent pas pirater le code afin de corriger des bugs, ou en aucune façon ne peuvent modifier le produit BI comme ils peuvent le faire avec OSBI.

#### **4.3.1 Pourquoi, les entreprises choisissent-elles, BI Open Source**

Les principales raisons pour lesquelles les entreprises choisissent de tirer parti d'une solution « BI open » au lieu d'une solution traditionnelle sont les suivantes :

- L'entreprise a une équipe compétente de développement qui peut exploiter le code et contribuer à la communauté Open Source en manipulant le logiciel afin de soutenir le BI, dont l'entreprise a besoin.
- L'entreprise essaie d'économiser de l'argent, et une « licence libre de coût » est séduisante. (Cependant, nous signalons ci-dessous que les licences de logiciels ne sont pas le seul coût associé à OSBI.)
- L'entreprise peut être elle-même une société Open Source ou promouvoir une politique de soutien aux initiatives open source.

### **4.3.2 Justification de choix des outils open BI**

L'entreprise est fondamentalement contre l'octroi de licences commerciales de logiciels ; elle pense que le logiciel devrait être libre et transparent.

### **4.3.3 Absence ou droits de licence réduits**

BI Open Source permettant d'installer dans votre environnement la plateforme de base.

### **4.3.4 Mises à jour fréquentes**

L'inconvénient majeur de la BI traditionnelle tient au fait qu'elle nécessite un long et complexe cycle de mise à niveau, avant la mise en place des nouvelles fonctionnalités. Par contre, si vous choisissez une solution OSBI soutenue par une communauté de développeurs compétents, vous bénéficierez de mises à jour fréquentes de logiciels et de nouvelles fonctionnalités, des corrections régulières de bugs. De plus, vos développeurs auront la possibilité de participer et de contribuer à l'amélioration du logiciel.

### **4.3.5 Le logiciel peut être personnalisé**

Un autre avantage notable « d'open source BI » (en particulier par rapport aux solutions traditionnelles de BI) est qu'il peut être adapté afin de répondre aux besoins spécifiques de votre entreprise. Votre développeur ou consultant peuvent créer une solution logicielle hautement personnalisée beaucoup moins

onéreuse que si vous deviez payer cette adaptation au fournisseur de BI commerciale. De plus, il n'y a aucune obligation pour vous de partager votre code personnalisé avec la communauté OSBI.

## **4.4. Inconvénient de BI Open Source**

### **4.4.1 Cette solution exige des développeurs hautement qualifiés.**

Bien que OSBI offre de nombreux avantages, son principal inconvénient tient au fait que sa manipulation nécessite une grande expertise en matière de développement ; ce qui implique la présence au sein de l'entreprise d'un staff expérimenté.

### **4.4.2 Peut être coûteux**

Comme nous l'avons mentionné, «BI, open source» vous offre un ensemble assez mature de fonctionnalités de business intelligence gratuitement, mais vous aurez généralement à payer le propriétaire principal de la communauté Open Source pour le soutien, ou des fonctionnalités avancées (communément appelés Entreprise Édition). Par exemple : le fournisseur open source, de Pentaho BI offre un ensemble noyau pouvant être téléchargé et utilisé gratuitement ; mais Pentaho BI, a développé des extensions payantes, pour dédommager leur effort. Dans ce contexte avant de s'engager, il y a tout intérêt à évaluer la balance coût/bénéfice si l'on est une petite entreprise.

### **4.4.3 L'interface utilisateur n'est pas attrayante**

Les logiciels de business intelligence sont créés par (et pour) les développeurs et non pas pour des utilisateurs néophytes.

## Chapitre 5 : Principe de la téléphonie mobile

Les réseaux des opérateurs mobiles furent conçus originellement pour le transport de la voix sur le modèle de la téléphonie filaire commutée. Progressivement, grâce à la création de nouveaux types de modulations et de technologies d'accès au réseau, l'offre de service s'est enrichie permettant l'envoi de données tel que : les SMS, l'accès à Internet ou encore la télévision mobile.

Par la suite, l'élargissement de la bande passante a hâté le passage de l'univers mobile GSM, vers le tout IP à travers la mise en œuvre progressive de la 4 G.

### 5.1 Structure générique d'un réseau de téléphonie mobile

Un réseau de téléphonie mobile est subdivisé en 3 parties (Figure 21) :

- Un réseau d'accès assurant la modulation du trafic,
- Un réseau garantissant la mobilité de l'utilisateur,
- Un backhaul (réseau d'amenée) qui relie le réseau d'accès au réseau cœur. Les liaisons du réseau d'amenée peuvent être en fibre optique, en cuivre ou supportées par des faisceaux hertziens.

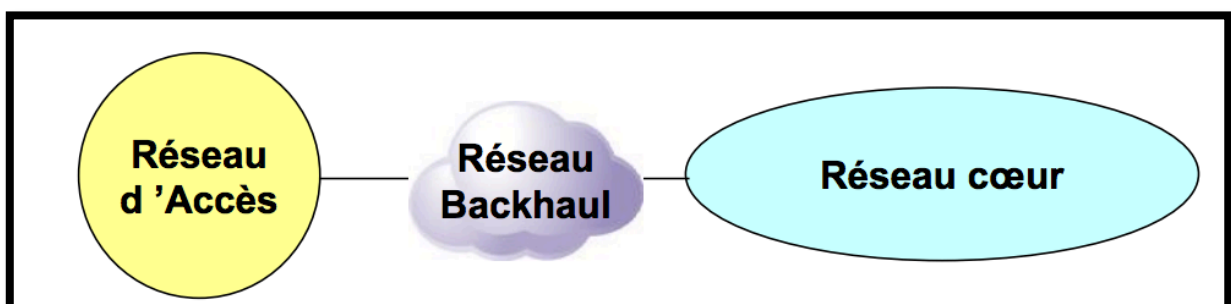


Figure 21 : Structure générique d'un réseau de téléphonie mobile. Source : (<http://www.efort.com>)

#### 5.1.1 Réseau d'accès

Le réseau d'accès assure la couverture de zones géographiques appelées cellules. Ces cellules ont une forme hexagonale. Leur taille peut varier d'une trentaine de kilomètres à quelques mètres en fonction de l'intensité du trafic, et du type d'antenne utilisée. Par ailleurs, la taille d'une cellule n'est pas fixe. En

effet, elle se contracte ou se dilate, en fonction de la météorologie, de la présence d'obstacles et de l'intensité du trafic. Chaque cellule comporte les matériels et logiciels nécessaires pour communiquer avec les stations mobiles (Figure 22).

#### **5.1.1.1 GSM (2 G)**

Le réseau d'accès GSM (2G) est appelé BSS (Base Station Subsystem). Il est constitué de BTS (Base Transceiver Station) et de BSC (Base Station Controller) :

Les BTS (Base Transceiver Station) sont des émetteurs-récepteurs ayant un minimum « d'intelligence ».

La BSC (Base Station Controller) contrôle un ensemble de BTS (une cinquantaine). Pour le trafic venant des BTS, la BSC joue un rôle de concentrateur ; alors qu'elle joue le rôle d'aiguilleur pour le trafic venant du réseau cœur, vers le BTS dont dépend le destinataire.

#### **5.1.1.2 UMTS (3 G)**

Le réseau d'accès UMTS (3 G) est appelé UTRAN (UMTS Terrestrial Radio Access Network). Il est constitué de NodeB et de RNC (Radio Network Controller) :

Le NodeB est équivalent au BTS du réseau d'accès 2G

Le RNC contrôle une centaine de NodeB. Il est l'équivalent du BSC du réseau d'accès 2 G.

Certains fournisseurs proposent les fonctions NodeB/RNC combinées dans le même équipement pour la mise en œuvre du réseau d'accès 3 G.

#### **5.1.1.3 LTE (4 G)**

Le réseau d'accès LTE 4G (Long Term Evolution of 3G) ou eUTRAN (Evolved UTRAN). Il ne comporte qu'une seule entité, l'eNodeB. L'eNodeB est fonctionnellement équivalent à un NodeB et un RNC.

Le RAN (Radio Access Network) d'un réseau LTE se limite donc eNodeB, aux antennes et aux liaisons en fibres optiques vers les antennes distantes (liens CPRI) et celles reliant les eNodeB entre eux et avec le cœur de réseau.

Tout comme le GSM et l'UMTS, le LTE est une technologie cellulaire qui offre la possibilité de réutiliser les mêmes fréquences hertziennes dans des cellules radio mitoyennes, grâce aux codages radio OFDMA et SC-FDMA et à un encodage permettant d'identifier les cellules. Cela permet d'affecter à chaque cellule une largeur spectrale variant de 5 à 20 MHz et donc d'avoir une bande passante plus importante et plus de débit dans chaque cellule.

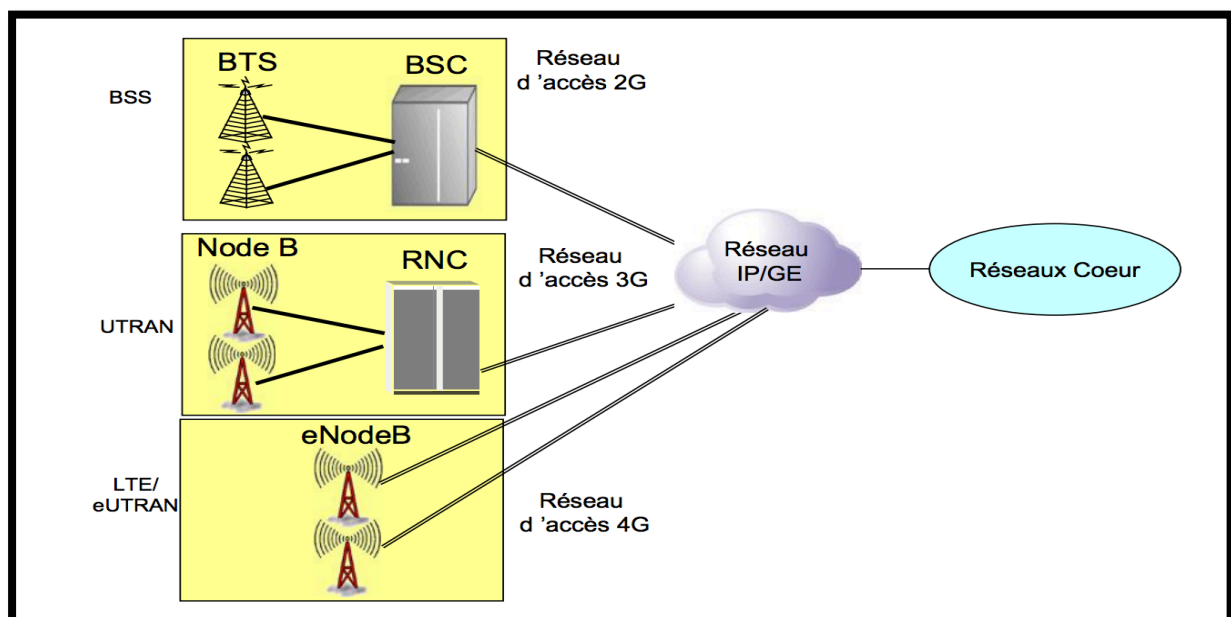


Figure 22 : Schéma des réseaux d'accès 2 G, 3 G, 4 G de la téléphonie mobile. Source : (<http://www.efort.com>)

## 5.1.2 Réseau cœur (Figure 23)

### 5.1.2.1 Le réseau cœur 2G/3G.

Il se subdivise en deux ensembles :

- 1) Le domaine circuit, qui permet le transport de la voix. Initialement constitué de « commutateurs voix », il a évolué vers une structure « NGN mobile » (Next Generation Network) dénommée R4.

- 2) Le domaine paquet appelé GPRS (General Packet Radio Service) ou EDGE (Enhanced Data Rates for GSM Evolution). Il permet l'accès à l'univers IP et à ses services. Il est constitué de commutateurs qui transfèrent des paquets émis par les accès 2G et 3G vers l'internet et en sens inverse transfèrent à l'utilisateur mobile, des paquets en provenance des accès 2G et 3G.

### 5.1.2.2 Le réseau 4G

Le réseau 4G est basé sur un nouveau concept « réseau cœur » incluant la téléphonie et le service des données. Ce concept est appelé ePC (Evolved PacketCore). Il utilise des technologies « full IP » ; c'est-à-dire basées sur les protocoles Internet pour la signalisation, le transport de la voix et des données. Ce cœur de réseau permet l'interconnexion via des routeurs avec les autres eNodeB distants, les réseaux des autres opérateurs mobiles, les réseaux de téléphonie fixe et le réseau Internet (Figure 23).

L'utilisation du protocole IP de bout en bout dans le cœur de réseau permet des temps de latence réduits pour l'accès internet et les appels vocaux LTE.

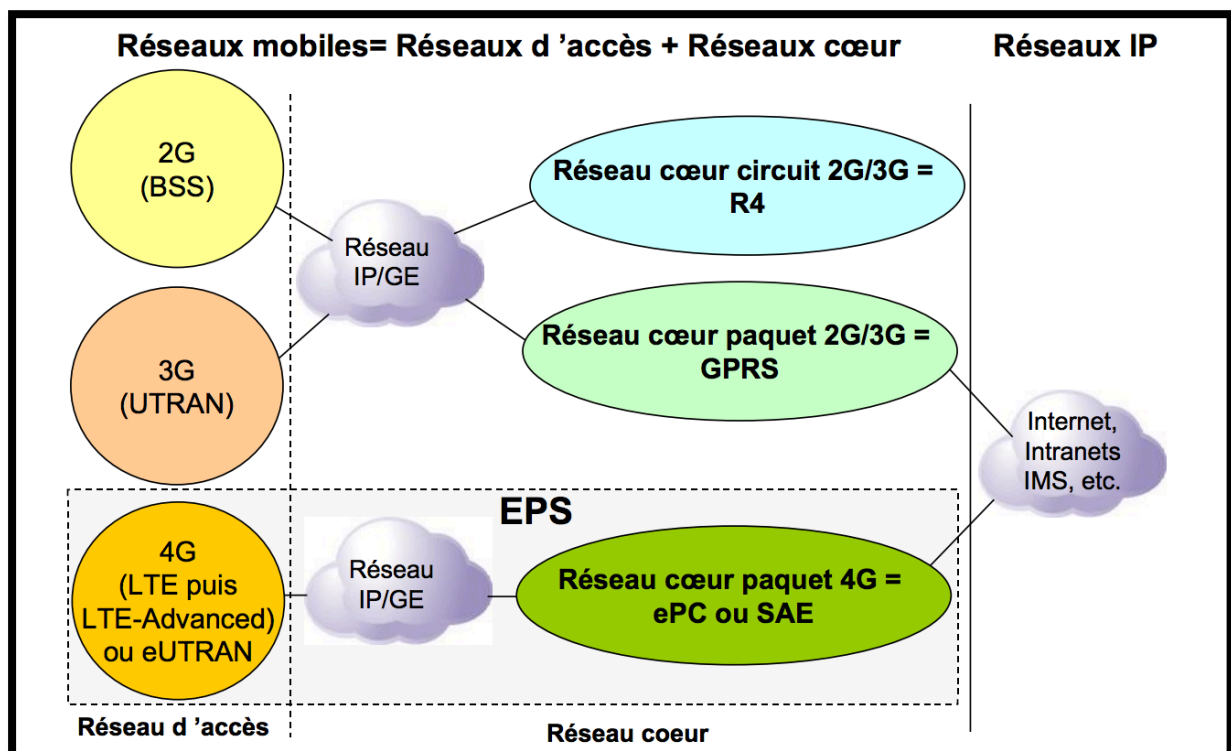


Figure 23 : Schéma des différents « réseaux cœur » de la téléphonie mobile. Source : (<http://www.efort.com>)

## **5.2 Les entités fonctionnelles du « réseau cœur »**

Il comporte 2 sous-systèmes :

- 1) Le sous-système d'acheminement connu sous la dénomination NSS (Network Sub-system).
- 2) L'OMC (Operations and Maintenance Center)

### **5.2.1 Le NSS**

Il comprend l'ensemble des fonctions nécessaires à l'établissement des appels et à la mobilité. Il est constitué de commutateurs et de 3 bases de données.

#### **5.2.1.1 Le MSC**

Le MSC réalise la commutation entre le système radio et les réseaux fixes. Il réalise toutes les fonctions de connexion et de localisation indispensables à l'acheminement des appels, de et vers les mobiles, dans une zone géographique appelée zone de localisation du MSC, à l'aide des procédures suivantes :

- Procédures pour l'enregistrement des localisations.
- Procédures requises pour les handovers.

#### **5.2.1.2 Les bases de données**

- 1) Base de données HLR (Home Location Register). Cette base de données contient les informations relatives aux abonnés du réseau. Chaque abonné y est décrit par un enregistrement contenant le détail des options d'abonnement et des services complémentaires accessibles à l'abonné. À ces informations statiques se rajoutent des informations dynamiques telles que la dernière localisation connue du mobile (localisation permettant la taxation et le routage des appels vers le MSC sous lequel le mobile est localisé), et son état. Le HLR contient par ailleurs la clé secrète de l'abonné qui permet au service d'authentifier l'abonné. Cette clé est inscrite sous un format codé que seul l'AUC (Authentication Center) peut décrypter.
- 2) Base de données VLR (Visitor location Register). Le réseau doit connaître à chaque instant, la localisation des abonnés

présents. Cette base a donc pour but l'enregistrement des informations dynamiques relatives aux abonnés connectés au moment t. Dans le VLR, chaque abonné est décrit par un identifiant et une localisation. Grâce à ces informations, le réseau est apte à acheminer un appel vers un abonné mobile. A chaque changement de zone de localisation d'un abonné, le VLR du MSC, auquel est rattaché le mobile, doit être mis à jour, ainsi que l'enregistrement de cet abonné dans le HLR. En effet, le HLR est immédiatement interrogé dès le début d'un appel afin de connaître la dernière localisation connue de l'abonné.

- 3) AUC (authentication Center). L'AUC (Authentication Center) est une fonction associée à un HLR. Elles sauvegardent dans le HLR une clé d'identification pour chaque abonné mobile. Cette clé est utilisée afin d'élaborer les données nécessaires à l'authentification de l'abonné dans le réseau GSM.

### **5.2.2 OMC (Operations and Maintenance Center) et NMC sous-systèmes d'exploitation et de maintenance.**

L'OMC comporte deux parties :

- 1) OMC-R (Operations and Maintenance Centre Radio), dont la fonction consiste à gérer les éléments du BSS,
- 2) OMC-S (OMC Switching), qui a pour fonction de gérer les éléments du NSS.

Le NMC (network Management center) assure l'administration générale centralisée du réseau. Les fonctions suivantes peuvent être assurées :

- Fonctions liées à la gestion commerciale ou administrative du réseau.
- Gestion de la sécurité.
- Gestion des performances.
- Gestion de la configuration.
- Maintenance et gestion des alarmes.

## **5.3 Les entités fonctionnelles du réseau GSM**

Nous ne détaillerons dans la suite de ce chapitre que les entités fonctionnelles relatives au système GSM, sur l'architecture duquel notre modèle de gestion sera établi. Le système GSM est composé de 4 entités fonctionnelles.

### **5.3.1 La station mobile (MS)**

C'est le matériel, utilisé par l'utilisateur du réseau pour accéder aux services de télécommunication. Une station mobile (MS, Mobile Station) est constituée de deux éléments séparables :

- 1) Un équipement mobile qui fournit les capacités radio et logiciels nécessaires au dialogue avec le réseau et demeure indépendant de l'abonné-utilisateur.
- 2) Une carte SIM (Subscriber Identification Module) qui contient les caractéristiques de l'abonné et de ses droits.

### **5.3.2 Le sous-système radio (BSS, Base Station Subsystem).**

Il assure la couverture de zones géographiques appelées cellules qui contiennent les matériels et logiciels nécessaires pour communiquer avec les stations mobiles.

#### **5.3.2.1 Base Transceiver Station (BTS)**

Le BTS (Base Transceiver Station) relie les stations mobiles à l'infrastructure fixe du réseau. Le BTS est composé d'un ensemble d'émetteurs/récepteurs. Il assure :

- 1) la gestion du multiplexage temporel (une porteuse est divisée en 8 slots dont 7 sont alloués aux utilisateurs),
- 2) la gestion des sauts de fréquence,
- 3) les opérations de chiffrement,
- 4) les mesures radio permettant de vérifier la qualité de service ; ces mesures sont transmises directement au BSC,
- 5) la gestion de la liaison de données (données de trafic et de signalisation) entre les mobiles et la BTS,
- 6) la gestion de la liaison de trafic et de signalisation avec le BSC.

La capacité maximale d'un BTS est de 16 porteuses, soit 112 communications simultanées. En zone urbaine où le diamètre de couverture d'un BTS est réduit, cette capacité peut descendre à 4 porteuses soit 24 communications.

### **5.3.2.2 Base Station Controller (BCS)**

Une BSC gère un ou plusieurs BTS et n'est reliée qu'à un seul MSC. La BSC joue le rôle de concentrateur pour le trafic provenant des BTS. Par ailleurs, il joue le rôle d'aiguilleur vers le BTS dont dépend le destinataire, pour le trafic venant du commutateur.

Une BSC utilise les mesures radio des BTS pour gérer la signalisation des « Handover » entre les cellules dont elle a la charge.



# **PARTIE II**

**DÉVELOPPEMENT**

**D'UN**

**PROTOTYPE**



# PRÉLUDE

Dans la deuxième partie de ce mémoire, nous présenterons les différentes étapes relatives à la création de notre « **Prototype Géodécisionnel** », spécifiquement dédié aux besoins des opérateurs de téléphonie mobile. Ce prototype est mis au point à l'aide d'outils Open Source dont nous avons décrit les caractéristiques au chapitre 4.

Il ressort, des chapitres : 1 — 3, que l'entrepôt de données (ED) et l'analyse en ligne (OLAP) constituent une technologie stratégique incontournable pour l'informatique décisionnelle. Par ailleurs, les systèmes d'information géographiques (SIG) en s'appuyant sur des bases de données géographiques, permettent quant à eux de traiter des quantités très importantes de données à référence géographique. Afin de pouvoir bénéficier dans un unique outil des fonctions spécifiques des SIG et des OLAP, (Bédard 2004) propose de marier les deux méthodologies. Ce couplage a ainsi donné naissance à une nouvelle famille d'outils mieux adaptée aux analyses spatiales et spatio-temporelles : les technologies Spatial-OLAP (S-OLAP) alimentées par des Entrepôts de Données Spatial (EDS). Yvan Bédard décrit la solution du S-OLAP comme « une plateforme visuelle conçue spécialement pour supporter une analyse spatio-temporelle rapide et efficace à travers une approche multidimensionnelle qui comprend des niveaux d'agrégation cartographiques, graphiques et tabulaires ».

En d'autres termes, l'intérêt des architectures S-OLAP est de pouvoir utiliser conjointement les outils OLAP (décision... etc.) ainsi que les outils géographiques (représentation cartographique, agrégateurs géographiques, etc.). Les systèmes S-OLAP permettent de gérer trois types de dimensions spatiales (voir section 3.2.1.1) :

- 1) Descriptives : les références spatiales sont textuelles (nom du lieu, par exemple).
- 2) Géométriques : chaque niveau hiérarchique comprend un ensemble de formes géométriques (polygones, points, etc.)
- 3) Mixtes : combinaison des deux autorisant à la fois les références textuelles et les références géométriques.

Le développement et la mise en œuvre de notre « **Prototype Géodécisionnel** » se réalisent suivant 3 phases :

- 1) Création en « open source », à l'aide de PostgreSQL, d'un entrepôt de données spatiales « **GouvData** » dans lequel les données du réseau d'accès de la téléphonie mobile sont organisées de manière multidimensionnelle (**chapitre 6**).
- 2) Création en « open source » d'un outil S-OLAP « **GouvRes** », en combinant les capacités d'un outil **SIG (QGIS)** à celles d'une bibliothèque **OLAP sous Python**). Ce nouvel outil est conçu selon le mode « SIG hybride » défini (section 3.3.2). Cette solution offre toute la panoplie des fonctionnalités SIG et un sous-ensemble de services OLAP (**chapitre 7**).
- 3) Mise en Œuvre de notre « **Prototype Géodécisionnel** » permettant la Visualisation des résultats des analyses multidimensionnelles sur des cartes ou des images satellitaires interactives (**Chapitre 8**).

# Chapitre 6 : Conception de l'Entrepôt de données spatial « GouvData »

L'objectif de ce chapitre est de créer un Entrepôt de Données Spatial (EDS), spécialement dédié aux besoins spécifiques des opérateurs de téléphonie mobile, en utilisant des outils Open Source.

## 6.1 Outils de modélisation et de développement

Les outils Open Source, comme nous l'avons spécifié au chapitre 4, sont désormais de qualité équivalente aux outils propriétaires existants sur le marché. De plus, ce choix est justifié par les économies d'échelle qu'ils permettent de réaliser, par rapport aux coûts extrêmement élevés des plateformes propriétaires, lesquelles ne sont pas parfaitement adaptées aux exigences des différentes activités de l'entreprise.

### 6.1.1 UML

UML (Unified Modelling Language), que l'on peut traduire par « langage de modélisation unifié ») est une notation permettant de modéliser un problème de façon standard. Ce langage est né de la fusion de plusieurs méthodes préexistantes. Ce langage est désormais devenu la référence en matière de « modélisation objet », à tel point que sa connaissance est souvent nécessaire pour obtenir un poste de développeur objet.

### 6.1.2 PostgreSQL

PostgreSQL est un puissant système de base de données relationnelle. Il a une solide réputation de fiabilité et de respect de l'intégrité des données. Il fonctionne sur tous les principaux systèmes d'exploitation, y compris Linux, UNIX (AIX, BSD, HP-UX, SGIIRIX, Mac OS X, Solaris, Tru64), et Windows. Il prend également en charge le stockage de grands objets binaires, y compris les images, les sons ou de la vidéo. Il dispose d'interfaces de programmation natives : pour C/C++, Java .net, Perl, Python, Ruby, Tcl, ODBC.

PostgreSQL est un SGBD Objet relationnel qui fournit un support natif pour les données spatiales (PostGIS).

Grâce à PostgreSQL, l'entrepôt de donnée s'enrichit de tous les avantages d'une solution fortement intégrée pour la mémorisation des données spatiales : sécurité, accès standard aux données, efficacité et passage à l'échelle, possibilité de mémoriser une grande quantité de données spatiales et d'obtenir des temps de réponse optimisés. Ceci représente les caractéristiques incontournables de l'analyse spatiale en ligne surtout au cas où l'entrepôt de données contiendrait une mesure avec la composante spatiale (Marchand *et al.* 2004) et (Maryvonne *et al.* s. d.).

De plus, PostgreSQL offre la possibilité de définir des données complexes, et des fonctions d'agrégation « ad hoc ». Ces fonctionnalités sont nécessaires à la modélisation et à l'agrégation des objets géographiques. Les attributs des mesures, comme nous l'avons mis en évidence aux chapitres 4 et 5, peuvent être des données complexes ou les fonctions d'agrégations utilisées pour les agréger et qui peuvent être différentes des agrégations SQL classiques, par exemple une moyenne pondérée sur la surface.

### **6.1.3 Talend Open Studio**

L'architecture ouverte de Talend génère une flexibilité inégalée qui permet de trouver nos propres réponses aux défis d'intégration. Talend réduit le délai d'apprentissage et facilite le processus d'adoption pour l'intégration de données, le profiling, le Big Data, l'intégration d'applications, etc..

### **6.1.4 Pantaho**

Pantaho, du fait de son héritage Open Source, contribue à une innovation continue, en matière de plate-forme d'analyse intégrable nécessitant de stocker une grande quantité de données. Ce logiciel lie étroitement l'intégration des données dans une plate-forme moderne. L'entreprise souhaite en effet que les nouveaux outils et processus puissent collecter rapidement et facilement tout type de données : stocker, gérer, manipuler, agréger, analyser et intégrer toutes ces données de manière à ce qu'elles engendrent un impact positif sur l'activité de l'entreprise.

## **6.2 Structure de l'Entrepôt de Donnée proposée**

Le modèle de notre entrepôt de données a été réalisé dans le but d'exploiter les représentations cartographiques de manière interactive, de telle sorte que les informations géographiques qui sont habituellement mises en exergues par les SIG soient complétées par des informations géométriques et des mesures qui s'afficheront sur les cartes. Pour ce faire, le modèle proposé prend en charge les données spatiales à travers la dimension « Planification urbaine » qui constitue l'un des facteurs prépondérants de l'analyse ultérieure.

Du point de vue conceptuel (voir section 1.2.3), notre entrepôt de données spatial a été créé sur le principe de la modélisation multidimensionnelle à l'origine des notions de faits et de dimensions (Kimball 1996). Dans cette optique, le sujet analysé est considéré comme un point dans un espace à plusieurs dimensions. Cette vision correspond à une structuration des données selon plusieurs axes d'analyses (ou dimensions) pouvant représenter des notions variées telles que le temps, la localisation géographique, la planification urbaine, etc..

Le stockage des données est conçu sur le mode relationnel (voir section 1.3.3.1). Cet entrepôt est structuré en suivant le schéma en flocon dérivant du schéma en étoile (sections 1.2.3.1 & 1.2.3.4). Dans ce cas, la table de fait est maintenue, tandis que les dimensions sont divisées en plusieurs tables en fonction de leurs hiérarchies. Le modèle en flocon est préconisé lorsque les tables de dimensions contiennent un très grand volume de données.

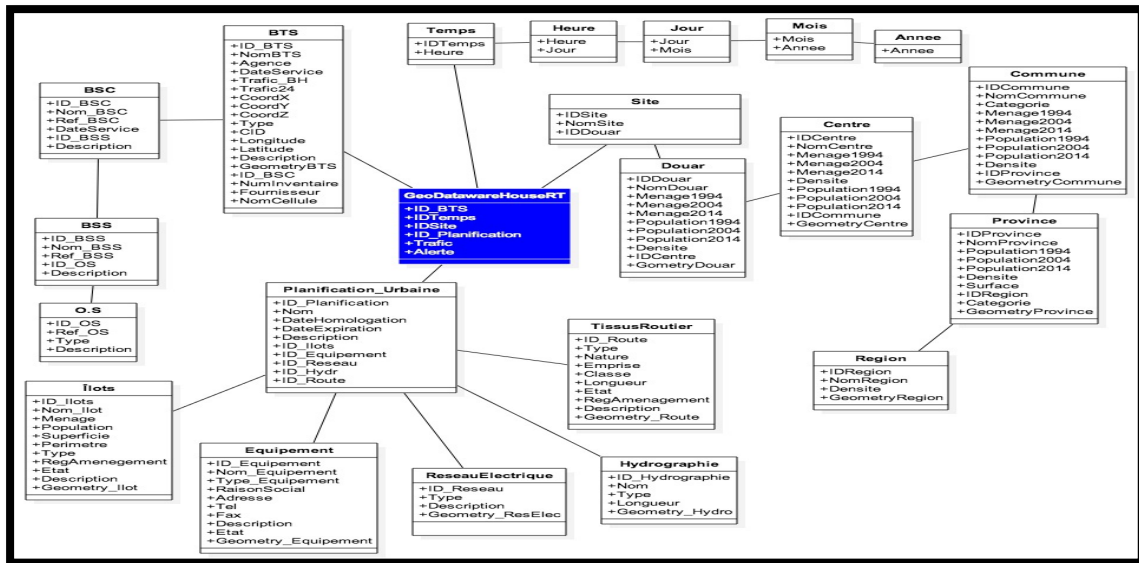


Figure 24 : Modèle en flocon de l'Entrepôt de Données Spatial

Le modèle de notre EDS (Figure 24) est représenté par la table de fait (table bleue) qui contient les informations nécessaires (Mesure) dans la description d'un réseau de téléphonie mobile tel que le trafic.

Le modèle comporte 4 magasins (datamart) représentant chacun un sujet déterminé :

### 6.2.1 Magasin Temps

Ce magasin (Figures 24 et 25) contient les informations temporelles. Il représente la division du temps suivant la hiérarchie :

- Dimension Année,
- Dimension Mois,
- Dimension Jour,
- Dimension heure,

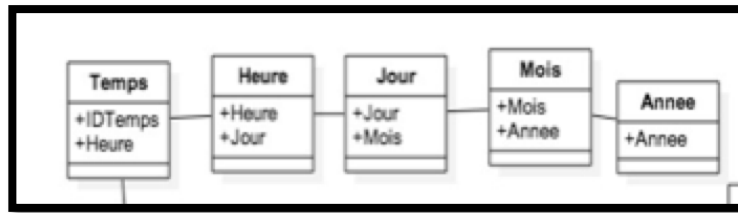


Figure 25 : Magasin dimension temps

## 6.2.2 Magasin Technique

Ce magasin (Figures 24 et 26) est hiérarchisé suivant les dimensions suivantes :

- Dimension BTS : (Base Transcrive System) : il représente le cœur d'EDS, puisqu'il constitue l'élément primordial dans la prise de décision. Cette dimension contient des informations relatives à un BTS, à savoir : le Nom, l'Agence, Trafic max, les coordonnées X, Y, Z avec un champ géométrie BTS de nature ponctuelle qui représente la partie spatiale de ce dernier.
- Dimension Cellule : une cellule est composée par un ensemble de BTS.
- Dimension BSC : Base Station Controller. Son rôle est de commander un certain nombre de BTS jusqu'à plusieurs centaines. Il contient les informations suivantes : Nom et date de mise en service.
- Dimension BSS : (Base Station Subsystem). Cette dimension contient une description relative au BSS.
- Dimension OS : (Operating System). Il contient toutes les informations sur le système d'exploitation.

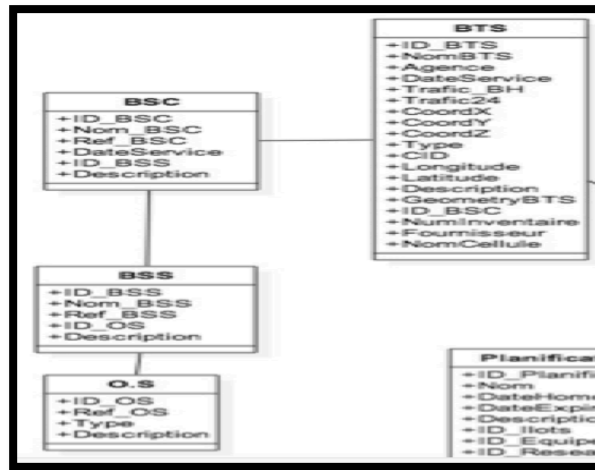


Figure 26 : Magasin dimension techniques

### 6.2.3 Magasin planification urbaine

Ce magasin (Figures 24 et 27) comprend la dimension spatiale prévue par des plans d'aménagement ou des PDU (Plan de développement urbain) ou des PDAR (Plan de Développement des Agglomérations Rurales) avec les dates d'homologation et d'expirations. Ils représentent les différentes entités urbaines existantes ou projetées. Pour une meilleure lisibilité, ce magasin a été hiérarchisé en 5 dimensions :

- 1) Dimension ilot : elle contient les ilots de populations, le nombre de ménages, une description des règlements urbains en vigueur dans cet ilot, et sa partie spatiale sous forme de polygone.
- 2) Dimension Équipement : elle contient les administrations, les agences et les différents équipements existants ou en projet.
- 3) Dimension Réseau électrique : sa partie spatiale Polyline représente les différentes entités du réseau électrique HT (Haute Tension), MT (Moyenne Tension), BT (Basse Tension).
- 4) Dimension Hydrographie : représente les Bassins versants et les tracés des différents cours d'eau.
- 5) Dimension Tissus routiers : représente les réseaux routiers et leurs typologies (route nationale, route secondaire, non classée, ferroviaire, autoroute) et avec une partie spatiale sous forme de polyLine.

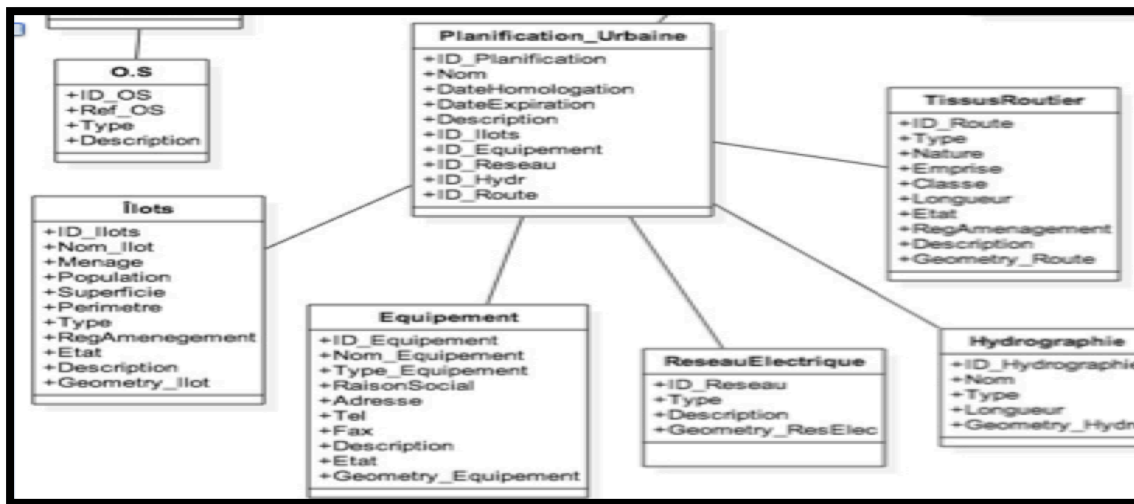


Figure 27 : Magasin planification urbaine

### 6.2.4 Magasin Découpage Administratif

Il représente la hiérarchie résultant du découpage administratif (région, province, commune, centre, douar) (Figures 24 et 28).

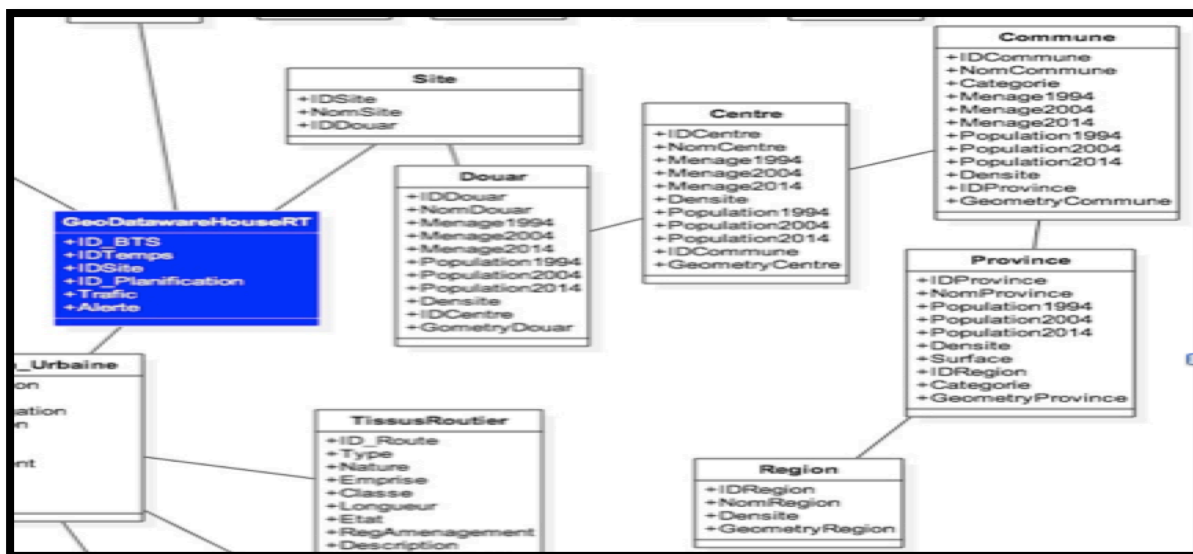


Figure 28 : Magasin découpage administratif

## 6.3 Création de la structure vide

L'entrepôt de données spatial a été implémenté en utilisant l'outil Open Source PostgreSQL (Figure 29). Les données spatiales et alphanumériques de l'application multidimensionnelle sont stockées en utilisant une modélisation en flocon.

De plus, l'entrepôt de données contient aussi un ensemble de métadonnées, comme le nom de l'application, le projet d'appartenance, le type d'application. Il mémorise aussi les informations concernant d'une part les dimensions qui doivent être cartographiées et d'autre part, les fonctions d'agrégation relatives aux mesures.

La structure vide de notre EDS réalisée sous PostgreSQL (PostgreSQL, 2014) contient la table de faits, toutes les dimensions ainsi que l'ensemble des mesures et des relations (Figure 29 et annexe 1).

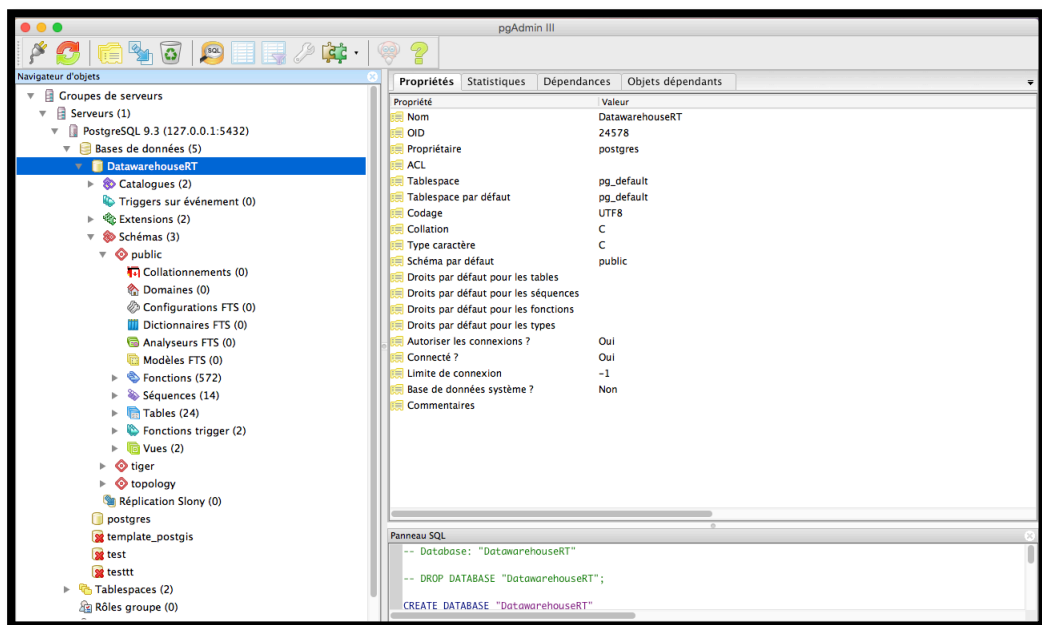


Figure 29 : Structure vide de l'entrepôt de données spatial sous PostgreSQL

## 6.4 Validation de l'EDS avec Pantaho

Nous avons testé la pertinence de notre modèle d'entrepôt avec l'outil Open Source Pantaho. À cet effet, nous avons utilisé à titre de matériel pilote des données issues d'un réseau de téléphonie mobile de la ville de Rabat au Maroc. Ces données étaient primitivement conservées sous des formats Access. L'alimentation de l'EDS vide a été réalisée à l'aide de l'outil Open source Talend Open Studio (Yazidi Alaoui *et al.* 2016). Nous ne donnerons pas ici de détails sur la procédure de sélection des données (Annexe 2 et Figure 54).

### 6.4.1 Création d'un cube (Schéma Workbench)

La création d'un cube au sens Mondrian est tout simplement la rédaction d'un fichier XML. Ce fichier permet de lier les informations du cube que nous souhaitons faire apparaître à partir de la base de données. La réalisation de ce fichier est effectuée à l'aide de l'outil Schéma Workbench qui offre une interface graphique pour effectuer cette tâche (Figure 30).

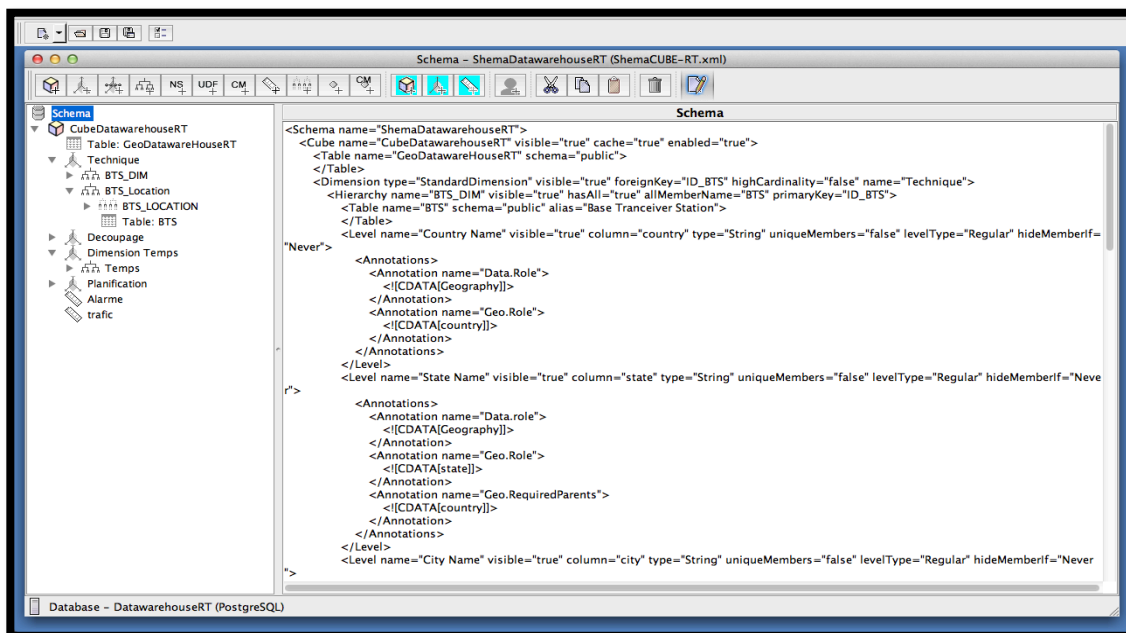


Figure 30 : Cube de données sous Workbench

Le cube de données vide est ensuite créé après l'avoir configuré à partir de l'onglet « préférences » (Figure 31) et en complément d'information (voir annexe 4).

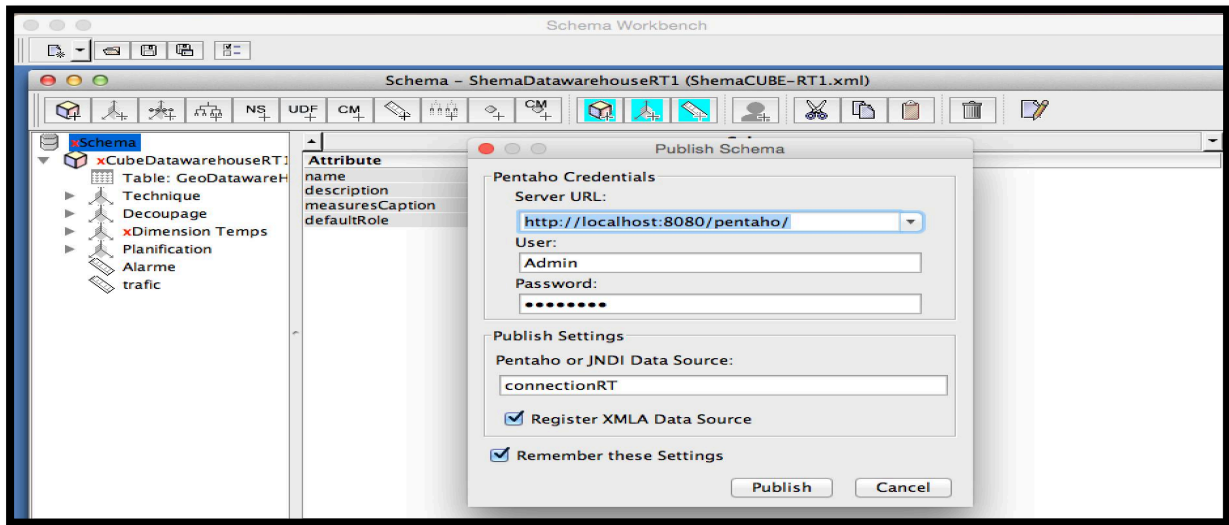


Figure 31 : Onglet préférence sous Workbench

## 6.4.2 Résultats obtenus avec Pantaho

Afin d'illustrer la résilience, la rapidité et la souplesse des performances de notre entrepôt de données nous allons fournir les différents types d'affichage qu'il permet de réaliser.

### 6.4.2.1 Affichage des données

Les données sélectionnées pour alimenter le cube sont relatives d'une part au trafic horaire enregistré pour chaque BTS et d'autre part, aux différents signaux d'alarme. Le tableau (Figure 32) présente uniquement la valeur du trafic moyen mensuel pour chaque BTS pour l'année 2013.

Annee compris dans 2013								
Annee								
Mois								
2013								
BTS_LOCATION	Aout	Avril	Decembre	Fevrier	Janvier	Juillet	Juin	Mai
	trafic	trafic	trafic	trafic	trafic	trafic	trafic	trafic
AGDALGENDARMO	24 272,4	24 272,4	23 292	22 654,24	23 430,64	25 081,48	24 272,4	24 272,4
CENTRALAGDALO	44 889	44 889	46 821,6	41 896,4	43 457,12	46 385,3	44 889	44 889
CUAGDALO	23 676,6	23 676,6	23 680,8	22 098,16	22 887,52	24 465,82	23 676,6	23 676,6
GAREAGDALO	46 821,6	46 821,6	46 821,6	43 700,16	45 260,88	48 382,32	46 821,6	46 821,6
MASSIRAO	46 821,6	46 821,6	46 821,6	43 700,16	45 260,88	48 382,32	46 821,6	46 821,6
NAKHIL0	45 841,2	45 841,2	46 821,6	42 785,12	44 345,84	47 369,24	45 841,2	45 841,2
RIADNAKHIL0	25 516,8	25 516,8	23 580	23 815,68	24 601,68	26 367,36	25 516,8	25 516,8

Figure 32 : Affichage des données du trafic mensuel de quelques BTS de la région de Rabat, pour l'année 2013

#### 6.4.2.2 Affichage des données sur une image satellitaire

L'intérêt primordial des SOLAP est de représenter des données sur un fond géographique afin d'offrir une vision régionale synthétique. La figure 33 représente une image satellitaire récente de la région de Rabat sur laquelle, la surface des points rouges représentant chaque BTS est proportionnelle à l'intensité du trafic annuel enregistrée pour chacun des BTS.

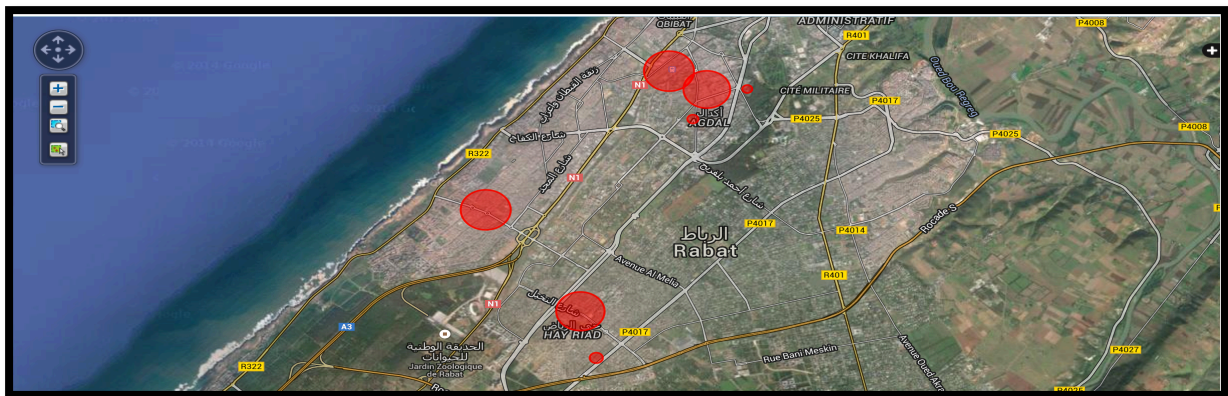


Figure 33 : Projection sur une image satellitaire, du trafic de quelques BTS de la région de Rabat. Le diamètre de chaque cercle est proportionnel à l'intensité du trafic.

### 6.4.2.3 Affichage sur un fond cartographique des informations relatives aux BTS.

Il est par ailleurs possible d'afficher les informations détaillées relatives à chaque BTS en cliquant sur le point rouge le représentant (Figure 34).

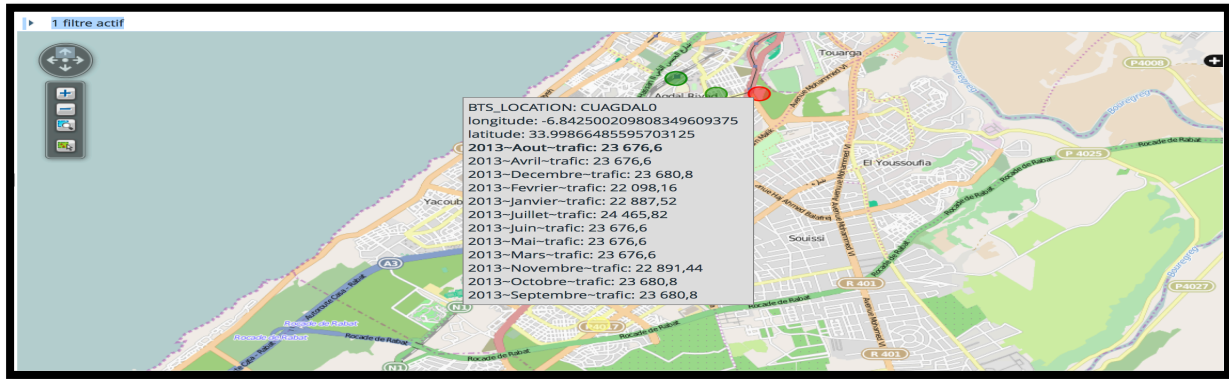
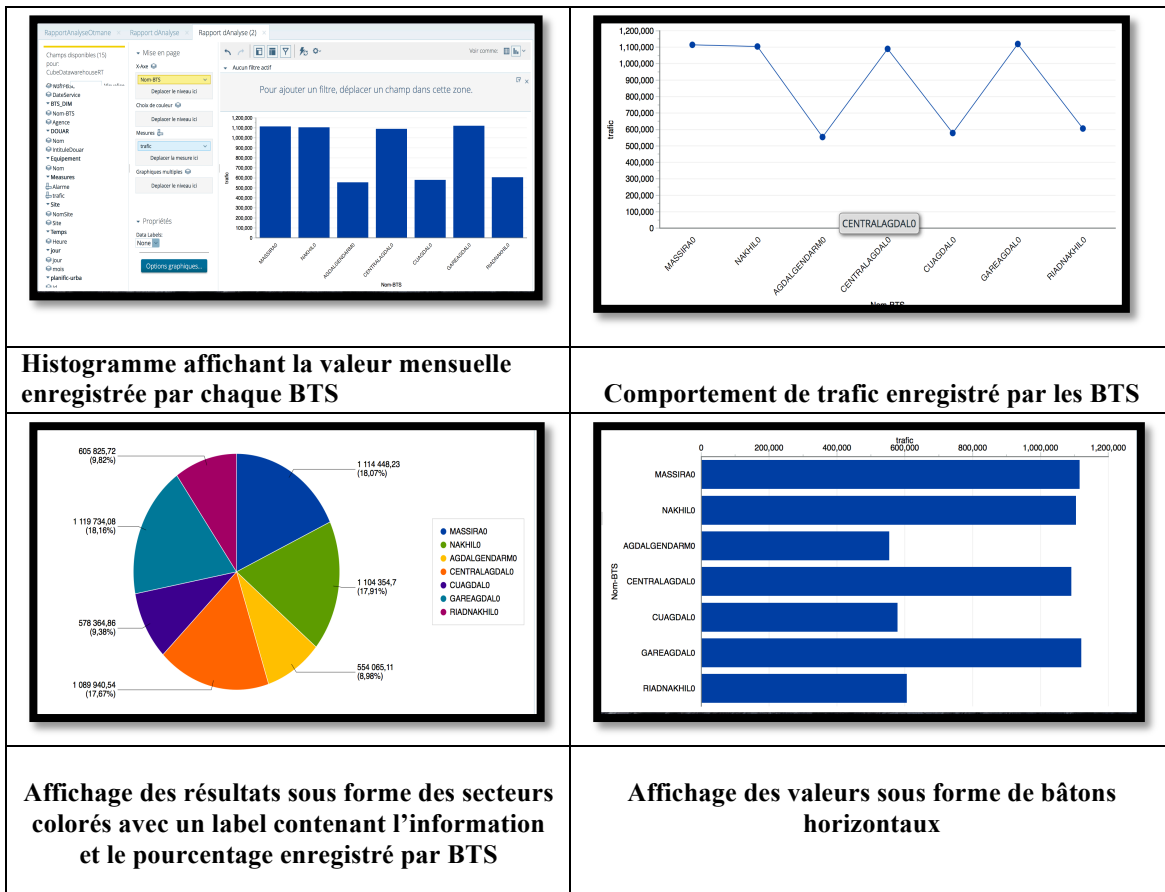


Figure 34 : Projection des données sur un fond cartographique avec affichage des informations relatives à chaque BTS

### 6.4.2.4 Affichage sous forme des graphes



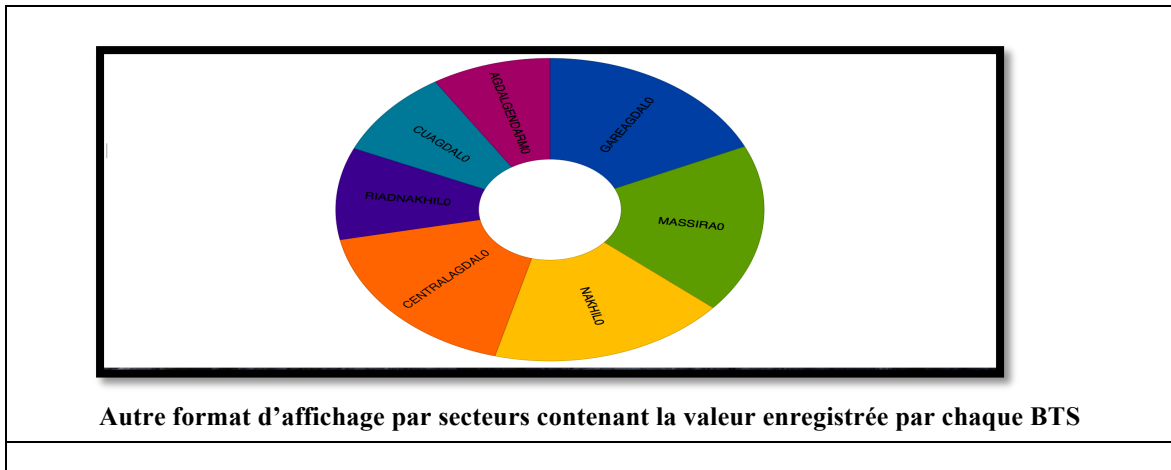


Figure 35 : Différents types de graphes obtenus à l'aide de Pantaho (c)

#### 6.4.2.5 Conclusion sous forme d'une discussion des résultats

La visualisation, à l'aide de Pantaho, des données stockées dans notre entrepôt de données spatial « GouvData » (Figures 32 à 35) montrent que notre EDS est parfaitement opérationnel et permet une navigation souple à l'intérieur des données à l'aide de Pantaho dont la puissance graphique permet un affichage des résultats sous différents types de graphiques (Figure 35). Par ailleurs, Pantaho permet une projection sur certains fonds satellitaires ou cartographiques à différentes échelles, des résultats des analyses réalisées pour chaque BTS (Figures 33 et 34). Ces résultats illustrent la puissance graphique et de représentation, spatiale de Pantaho. Mais Pantaho exige une connaissance profonde en matière d'informatique et en matière de modélisation de cube de donnée avec l'outil WorkBensh. C'est la raison pour laquelle nous avons choisi de développer un outil faisant office de SOLAP sous la forme d'un plug-in. Cet outil a été conçu afin d'être adapté aux besoins d'utilisateurs novices en informatique. En d'autres termes, cet outil a pour vocation d'afficher des résultats identiques à ceux obtenus à l'aide de Pantaho sans que l'utilisateur possède la moindre connaissance en informatique.

Dans le chapitre suivant, nous présentons notre méthode de développement de l'OLAP géographique (SOLAP), réalisée à partir des modules de langage de programmation Python et de l'outil Open Source Qgis.



# Chapitre 7 : Conception de l'outil Spatial OLAP « GouvRes »

La conception de « GouvRes » est conçue sur le mode « SIG hybride » défini (section 3.3.2). Cette solution a l'avantage d'offrir toute la panoplie des fonctionnalités SIG en les associant à un ensemble d'opérateurs OLAP.

La réalisation en open source de « GouvRes » résulte de la combinaison d'une part, des capacités du SIG (**QGIS**), auquel nous avons apporté des modifications et d'autre part, celles des bibliothèques (**OLAP sous Python**). Le développement de « GouvRes » s'est opéré en deux étapes :

- 1) Téléchargement et modification de QGIS
- 2) Exploitation de la bibliothèque OLAP sous Python.

## 7.1 Outils

Afin de pouvoir développer « GouvRes », nous avons utilisé les outils et les plateformes suivants :

### 7.1.1 QtCreator

Cet outil est nécessaire afin d'accéder au code source de QGIS. C'est une bibliothèque multiplateforme permettant de créer des GUI (programme employant des fenêtres). Il est initialement conçu pour être utilisé en C++ ; mais il est également possible de l'employer avec d'autres langages comme Java, Python, etc.

### 7.1.2 Cubes

Cubes est une architecture Python légère. Il se présente comme un ensemble d'outils pour le traitement analytique en ligne (OLAP), l'analyse multidimensionnelle et la navigation de données agrégées. Il permet :

- 1- Effectuer une étude multidimensionnelle.
- 2- Fournir une vue logique des données analysées.
- 3- Définir la manière dont les utilisateurs considèrent les données, et non pas leur implémentation physique dans l'EDS.

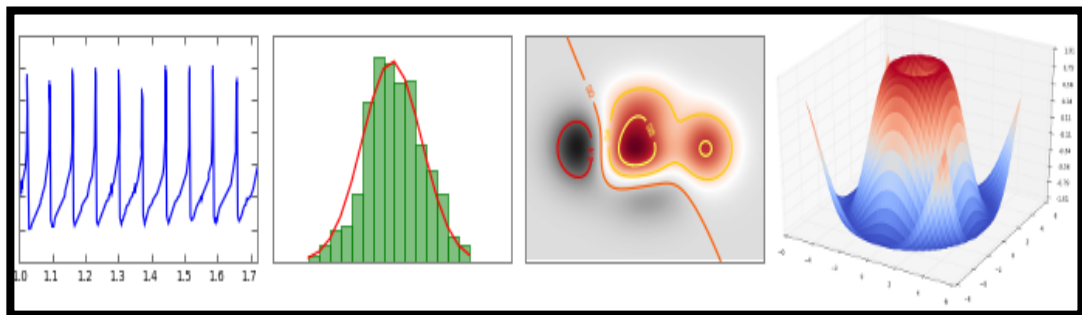
#### 4- Préciser les dimensions hiérarchiques

### 7.1.3 SQLAlchemy

C'est une bibliothèque de Mapping Objet-Relationnel (ORM) qui permet aux développeurs d'applications de disposer de la puissance et de la flexibilité de SQL. Elle fournit une gamme complète de modèles, autorisant un accès efficace et performant aux bases de données (voir annexe 4).

### 7.1.4 Matplotlib

C'est une bibliothèque de traçage 2D/3D qui produit des graphes de qualité pouvant être exploités dans des scripts Python (Figure 36). À la faveur d'une interface orientée objet et à travers un ensemble de fonctions familières aux utilisateurs MATLAB. La bibliothèque Matplotlib permet également de disposer d'un contrôle total des styles de lignes, des propriétés de police, des propriétés d'axes, etc. .



*Figure 36 : Exemple de graphes disponibles dans Matplotlib*

## 7.2 Création de la plateforme SIG

Afin de prendre en compte les besoins spécifiques des opérateurs de téléphonie mobile, la création de la plateforme SIG nécessite d'apporter des modifications au logiciel QGIS. Les caractéristiques de ce logiciel sont décrites sur le site web officiel de QGIS. Son code source y est également disponible : (<http://www.qgis.org/fr/site/forusers/download.html>).

### 7.2.1 Téléchargement et adaptation du logiciel Qgis

La mise en place opérationnelle de la plateforme SIG nécessite de procéder à la compilation du logiciel « QGIS » ; afin d'accéder à l'ensemble des fichiers représentant les classes (extension \*.cpp) et la bibliothèque (extension \*.h).

Le code source est ouvert avec QtCreator (<http://www.qt.io/>). Son remodelage est réalisé à l'aide du langage de programmation orienté objet C++.

À titre d'exemple nous donnons un aperçu sur l'ensemble des codes sources QGIS (Figure 37) et des fichiers de classes (Figure 38).

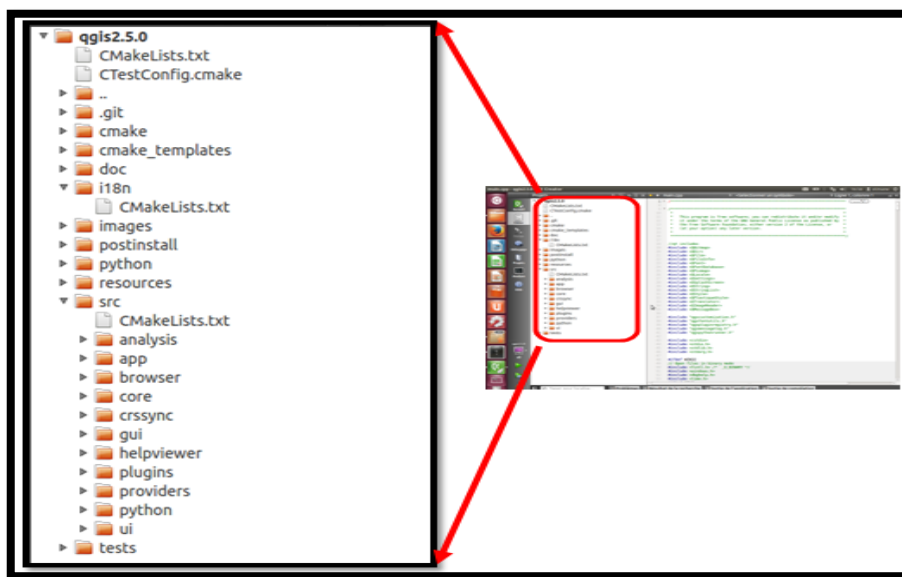


Figure 37 : Vue partielle du code source QGIS

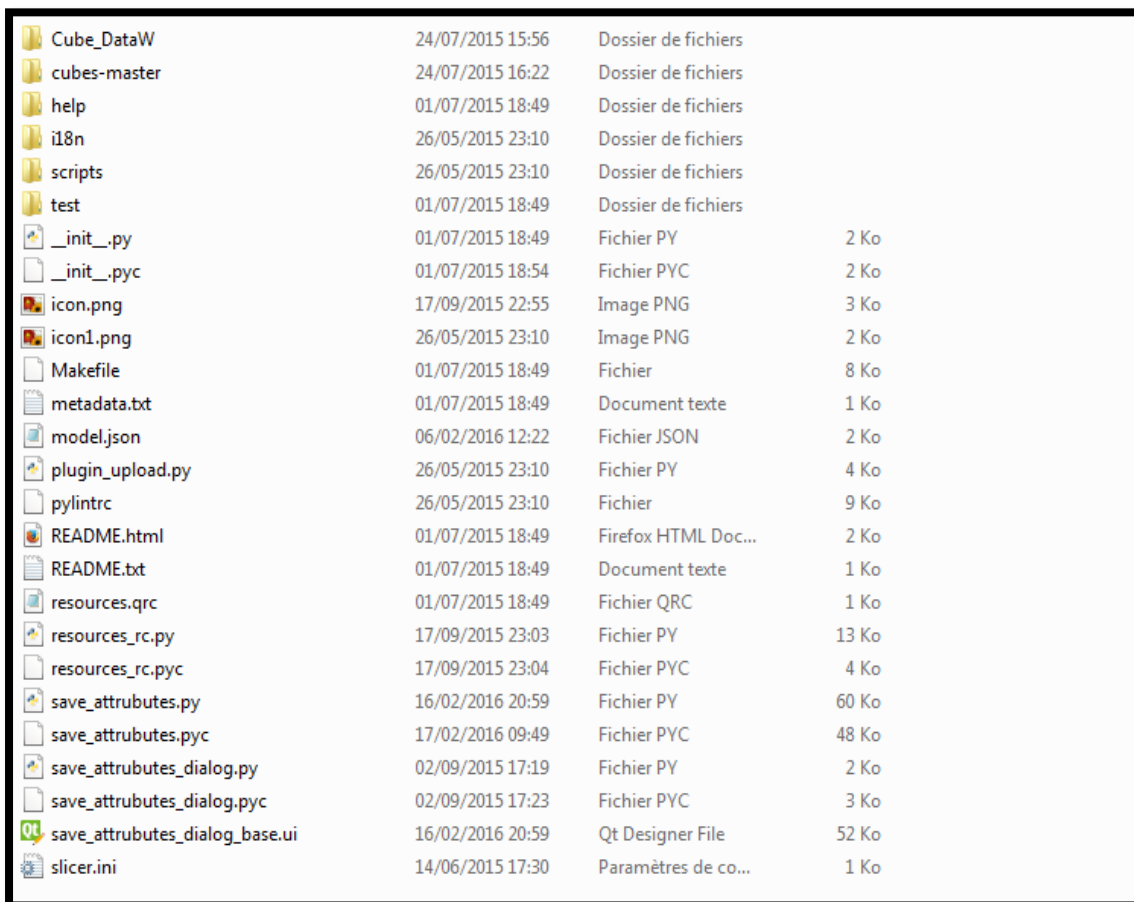


Les informations inhérentes à ces modifications sont rapportées dans l'annexe 5.

### 7.3 Création d'un plug-in « Moteur »

Ce plug-in, écrit en langage de programmation orienté objet Python, met en œuvre les différents éléments des bibliothèques Python afin de les intégrer dans la plateforme Qgis remodelée. (Voir annexe 7).

Ce plug-in est en réalité un dossier établi en respectant une structure reconnaissable par Qgis, (Figure 40).



Nom	Date	Type	Taille
Cube_DataW	24/07/2015 15:56	Dossier de fichiers	
cubes-master	24/07/2015 16:22	Dossier de fichiers	
help	01/07/2015 18:49	Dossier de fichiers	
i18n	26/05/2015 23:10	Dossier de fichiers	
scripts	26/05/2015 23:10	Dossier de fichiers	
test	01/07/2015 18:49	Dossier de fichiers	
__init__.py	01/07/2015 18:49	Fichier PY	2 Ko
__init__.pyc	01/07/2015 18:54	Fichier PYC	2 Ko
icon.png	17/09/2015 22:55	Image PNG	3 Ko
icon1.png	26/05/2015 23:10	Image PNG	2 Ko
Makefile	01/07/2015 18:49	Fichier	8 Ko
metadata.txt	01/07/2015 18:49	Document texte	1 Ko
model.json	06/02/2016 12:22	Fichier JSON	2 Ko
plugin_upload.py	26/05/2015 23:10	Fichier PY	4 Ko
pylintrc	26/05/2015 23:10	Fichier	9 Ko
README.html	01/07/2015 18:49	Firefox HTML Doc...	2 Ko
README.txt	01/07/2015 18:49	Document texte	1 Ko
resources.qrc	01/07/2015 18:49	Fichier QRC	1 Ko
resources_rc.py	17/09/2015 23:03	Fichier PY	13 Ko
resources_rc.pyc	17/09/2015 23:04	Fichier PYC	4 Ko
save_attributes.py	16/02/2016 20:59	Fichier PY	60 Ko
save_attributes.pyc	17/02/2016 09:49	Fichier PYC	48 Ko
save_attributes_dialog.py	02/09/2015 17:19	Fichier PY	2 Ko
save_attributes_dialog.pyc	02/09/2015 17:23	Fichier PYC	3 Ko
save_attributes_dialog_base.ui	16/02/2016 20:59	Qt Designer File	52 Ko
slicer.ini	14/06/2015 17:30	Paramètres de co...	1 Ko

Figure 40 : Ensemble des fichiers du plug-in Python

### 7.3.1 Création d'un fichier d'extension « \*. Ui »

Ce fichier représente l'interface personne-machine de notre plug-in. Il est construit à partir de différents composants nommés « widget » par exemple : les boutons (QWidgetButton), les zones de texte (QWidgetTextEdit), les listes de choix et les tableaux d'affichage (QWidgetTable).

Cette interface homme — machine, en mode création sous QtDesigner est représentée (Figure 41).

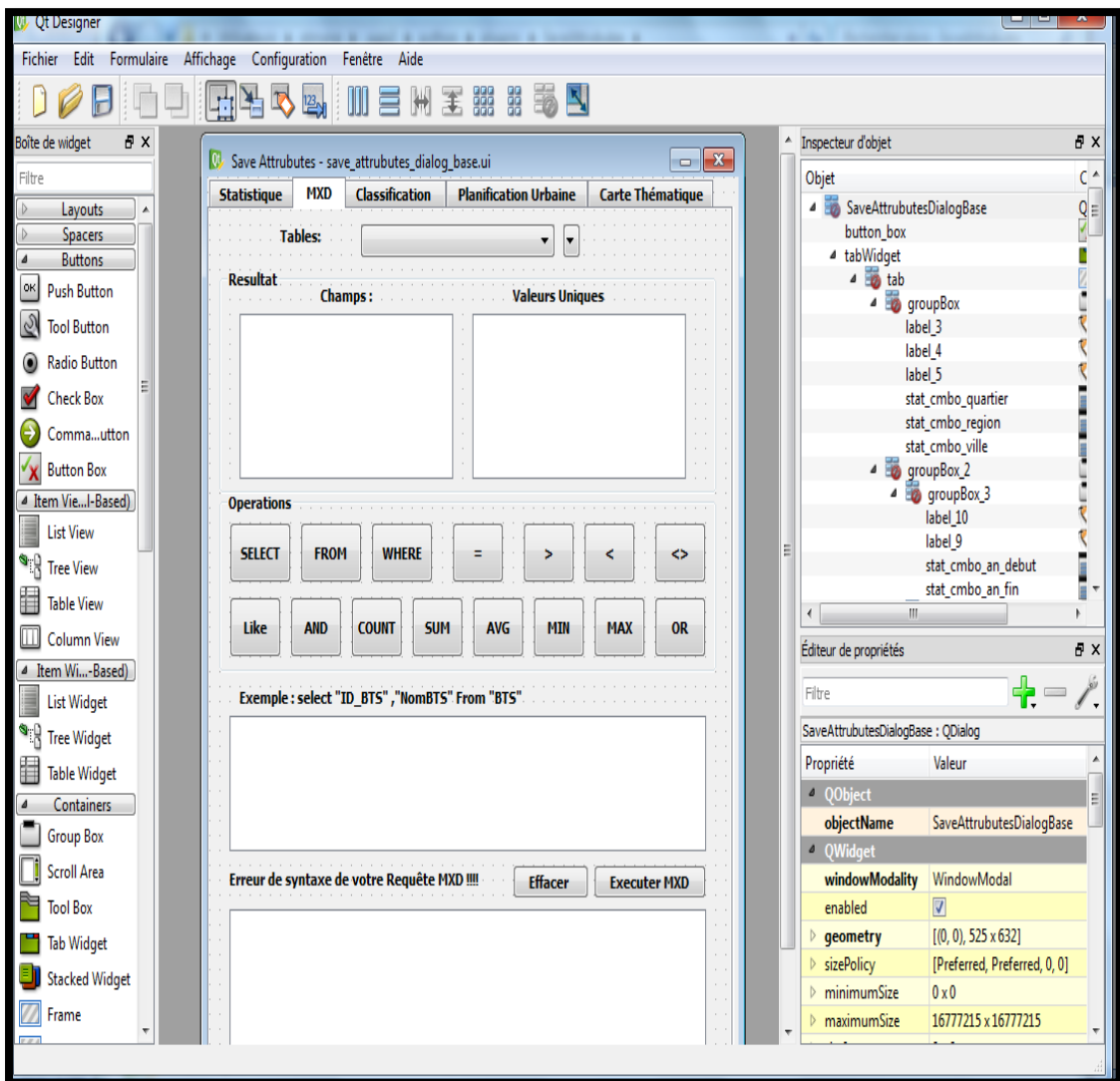


Figure 41 : Fenêtre homme-machine de notre Plug-in (extension. \*.ui)

### 7.3.2 Fichier d'extension « \*. Py »

Le fichier d'extension « Py » est le fichier principal comportant la classe et l'ensemble des fonctions et des signaux. (Une fonction est appelée quand une action est produite sur un Qwidget). Ce fichier contient la totalité des scénarios de connexion à l'Entrepôt de données spatial ; et l'ensemble des opérations automatisées.

Une partie du code, écrit sur le fichier d'extension « \*. Py », est représentée (Figure 42).

```
from PyQt4.QtCore import *

"""
Importation des Bibliotheque de BI
-----
"""

import sqlalchemy
import cubes
import cubes.tutorial.sql as tutorial
from cubes import Workspace, Cell, PointCut, RangeCut
from qgis.core import *
from qgis.core import QgsVectorLayer, QgsMapLayerRegistry, QgsGraduatedSymbolRendererV2, QgsSymbolV2, QgsRendererRangeV2

import matplotlib.pyplot as plt; plt.rcParams.update({'font.size': 14})
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import numpy as np
import matplotlib.pyplot as plt
import processing
import psychopg2

# Initialize Qt resources from file resources.py
import resources_rc
# Import the code for the dialog
from save_attributes_dialog import SaveAttributesDialog
import os.path

class SaveAttributes:
    """QGIS Plugin Implementation."""

    def __init__(self, iface):
        """Constructor.

        :param iface: An interface instance that will be passed to this class
            which provides the hook by which you can manipulate the QGIS

```

Figure 42 : Fichier contenant la classe (extention. \*py)

### 7.3.3 Fichier d'extension « \*. Json »

Afin de réaliser le couplage entre SOLAP « GouRes » et l'Entrepôt de données spatial « GouvData »; il est nécessaire de créer un fichier portant l'extension \*. json (Figure 43).

```
{
  "cubes": [
    {
      "name": "GeoDatawareHouseRT",
      "dimensions": ["BTS", "Site", "Temps"],
      "measures": [{"name": "Trafic", "label": "Trafic", "aggregation": ["Sum"]}],
      "joins": [
        {"master": "ID_BTS", "detail": "BTS.ID_BTS"},
        {"master": "ID_Site", "detail": "Site.ID_Site"},
        {"master": "ID_Temps", "detail": "Temps.ID_Temps"}
      ]
    }
  ],
  "dimensions": [
    { "name": "BTS", "attributes": ["NomBTS", "ID_BSC"] },
    { "name": "Site", "attributes": ["ID_Site", "Nom_Site"] },
    { "name": "Temps",
      "attributes": ["ID_Temps", "annee", "Month", "day", "hour"],
      "levels": [
        {"name": "annee", "attributes": ["annee"]},
        {"name": "Month", "attributes": ["Month"]},
        {"name": "day", "attributes": ["day"]},
        {"name": "hour", "attributes": ["hour"]}
      ]
    }
  ],
  "hierarchies": [
    { "name": "y", "label": "annee", "levels": ["annee"] },
    { "name": "ym", "label": "mois", "levels": ["annee", "Month"] },
    { "name": "ymd", "label": "jour", "levels": ["annee", "Month", "day"] },
    { "name": "ymdh", "label": "heure2", "levels": ["annee", "Month", "day", "hour"] }
  ],
  "default_hierarchy_name": "ymdh"
}
]
```

Figure 43 : Cube de données

Ensuite, il est nécessaire de créer une fonction de connexion à l'entrepôt de données spatial « GouvData » en précisant le nom du cube et l'emplacement du fichier \*. Json (Figure 44).

```
workspace = Workspace()
workspace.register_default_store("sql", url="postgres://postgres:alaoui@localhost:5432/DatawareHouseRTFinal")
workspace.import_model("C:\Users\otmane\.qgis2\python\plugins\SaveAttributes\model.json")
cube = workspace.cube("GeoDatawareHouseRT")
browser = workspace.browser(cube)
an_deb=self.dlg.stat_cmbo_an_debut.currentText()
an_fin=self.dlg.stat_cmbo_an_fin.currentText()
cuts = [
    PointCut("BTS", ["MASSIRAO"]),
    RangeCut("Temps", [an_deb],[an_fin],hierarchy="ymdh")
]
```

*Figure 44 : Connexion au Cube de données*

## 7.4 Intégration du plug-in de couplage sous Qgis modifié

Pour intégrer notre plug-in de couplage développé sous la plateforme SIG, il suffit d'effectuer une recherche sur la base des extensions de QGIS (Figure 45).

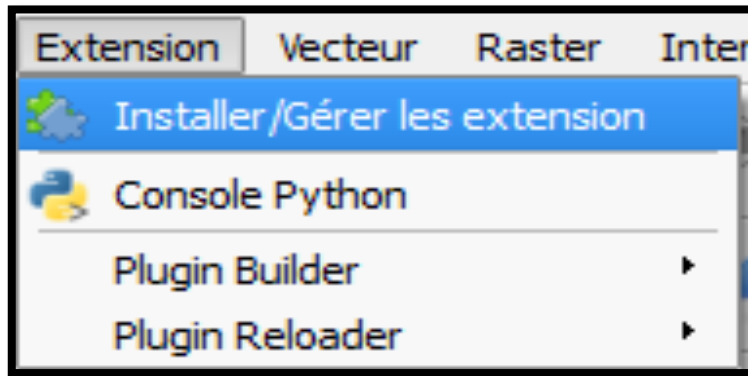


Figure 45 : Installation du nouveau Plug-in

On trouve le plug-in développé en écrivant sur la barre de recherche les premières lettres du nom du plug-in (Figure 46).

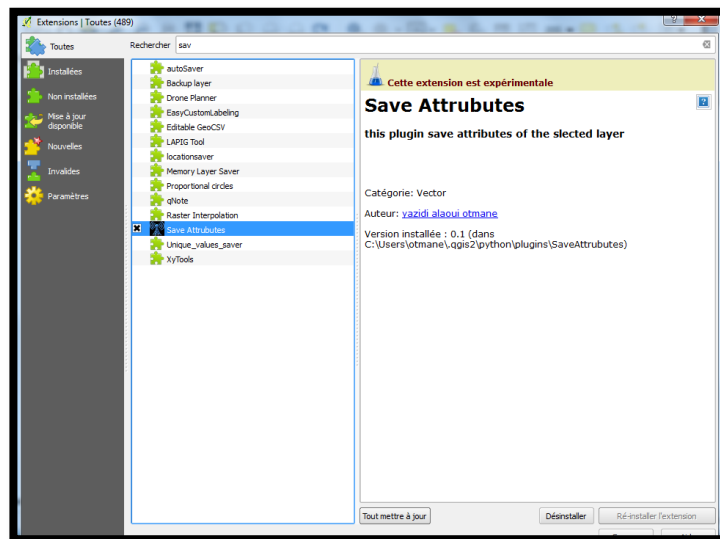


Figure 46 : Installation de nouveau Plug-in

Après avoir coché et actualisé, le plug-in s'intègre instantanément dans la barre des outils (Figure 47).



Figure 47 : : Le Plug-in Développé

## 7.5 Connexion à l'entrepôt de données spatial « GouvData »

La connexion initiale à notre Entrepôt de données spatial, est réalisée en cliquant sur l'icône suivant (Figure 48).



Figure 48 : Icône de connexion Entrepôt de données spatial

Les paramètres de connexion à notre serveur de données doivent être ensuite définis. Il s'agit : de l'adresse IP du serveur, du port de connexion, du nom de l'entrepôt de données spatial, du mode de connexion, du nom de l'utilisateur et du mot de passe du serveur (Figure 49).

Information de connexion

Nom: ServerDataWarehouseRT

Service:

Hôte: 127.0.0.1

Port: 5432

Base de données: DatawarehouseRT2016

mode SSL: désactive

Nom d'utilisateur: postgres

Mot de Passe: ●●●●●●

Enregistrer le nom d'utilisateur

Sauvegarder le mot de passe

Afficher seulement les couches dans la liste de couches

Ne pas résoudre le type pour les géométries non restreintes (GEOMETRY)

Uniquement regarder dans le schéma 'public'

Lister les tables sans géométries

Utiliser la table de métadonnées estimées

Tester la connexion

OK Annuler Aide

Figure 49 : Paramètre de connexion à l'Entrepôt de données spatial « GouvData »

Un test de connexion permet de vérifier la validité des paramètres renseignés. Si le test est positif, un message de confirmation affiche : « Connexion à l'entrepôt de données spatial : réussie » (Figure 50).

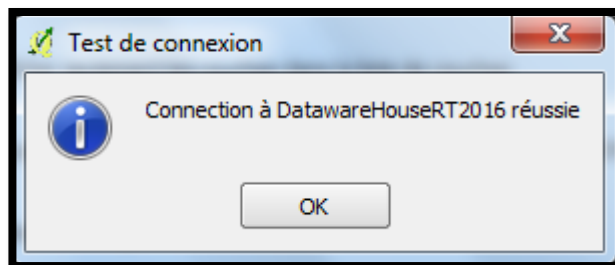


Figure 50 : Test :la connexion à l'EDS est réussie

Après avoir cliqué sur le bouton « connecter » (Figure 51), le système affiche l'ensemble des tables de dimensions et la table de faits existant dans l'entrepôt de données spatial (Figure 51). L'utilisateur peut alors sélectionner les dimensions à afficher.

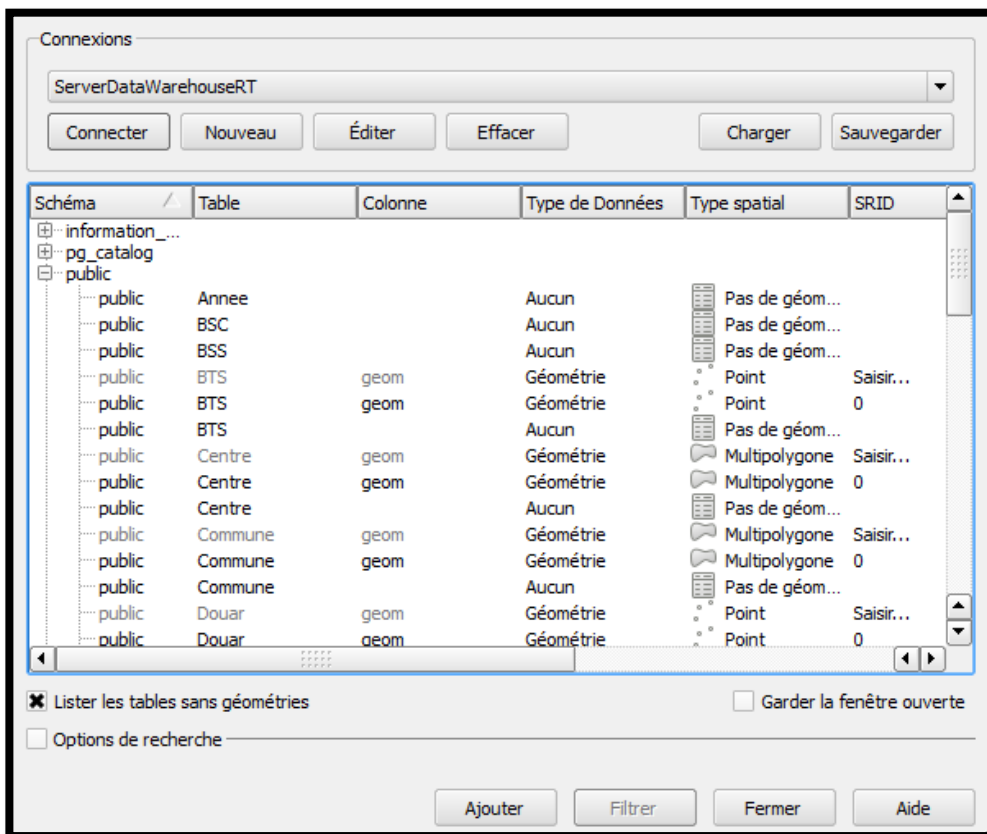


Figure 51 : connexion à l'entrepôt de données spatial

À la fin et après avoir défini le système de coordonnées et le système de projection, dans notre cas de figure, nous choisissons la projection Lambert relative à la zone 2 du Maroc. Une vision globale de l'ensemble des données spatiales (des tables attributaires) s'affichera alors sur le fond topographique (Figure 52 et 53).

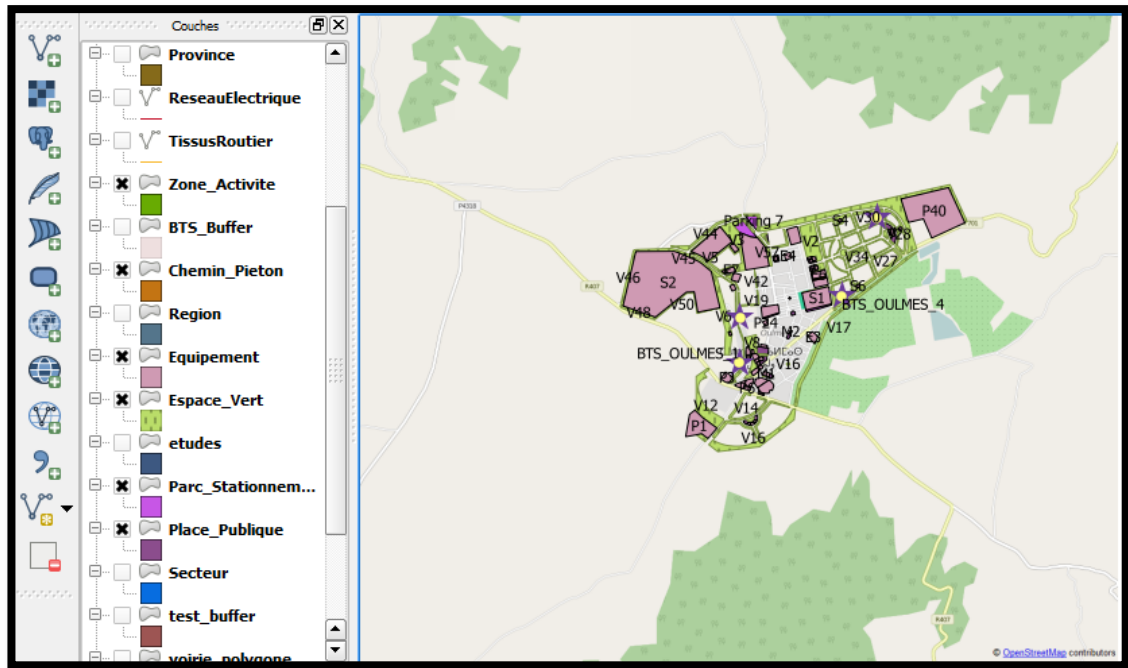


Figure 52 : Projection de l'ensemble des données spatiales figurant sur l'EDS

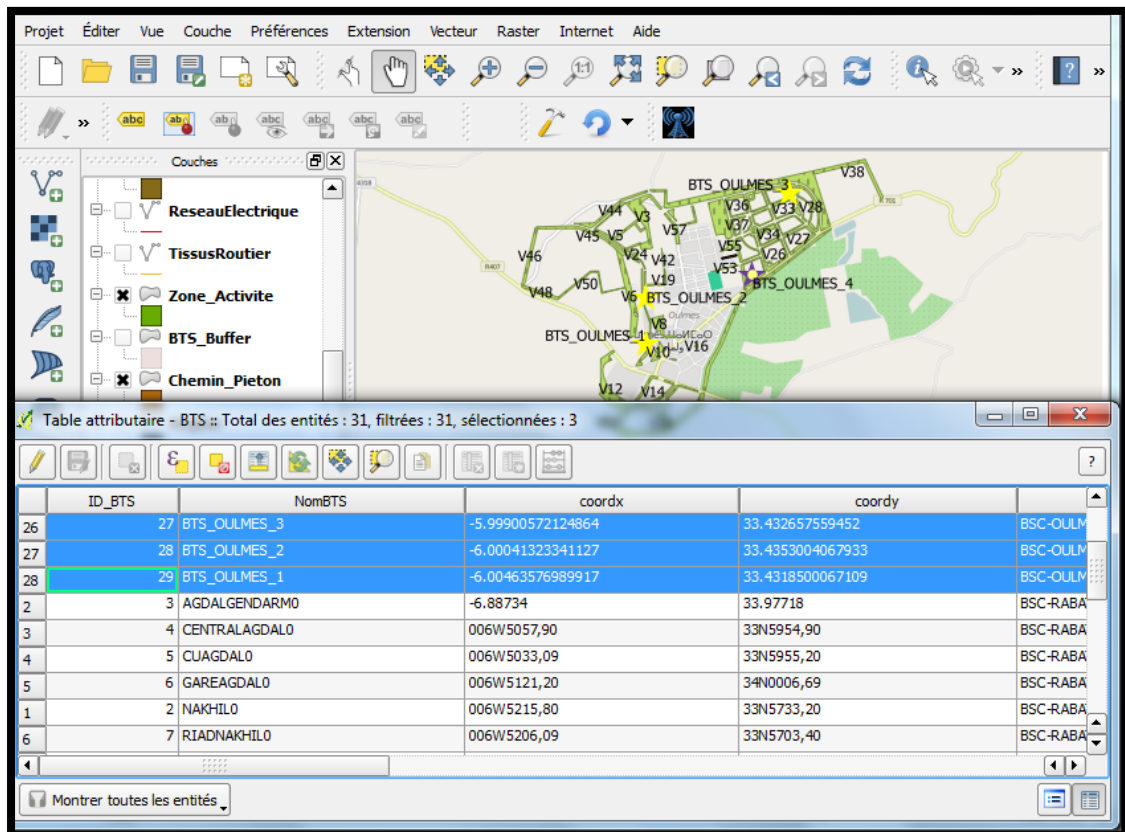


Figure 53 : Affichage de la partie attributaire associée aux tables

## 7.6 Conclusion

Le principe actif de notre SOLAP « GouvRes » réside dans le plug-in « Moteur ». Il constitue l'élément central de notre prototype géodécisionnel (Figure 54). Il en anime et coordonne en effet tous les modules, à savoir :

- Il assure la jointure entre notre SIG et les bibliothèques OLAP sous Python
- Il met en œuvre le cube de données pour interroger notre entrepôt de données spatial « GouvData »
- Il coordonne les actions des différentes requêtes et leur visualisation.

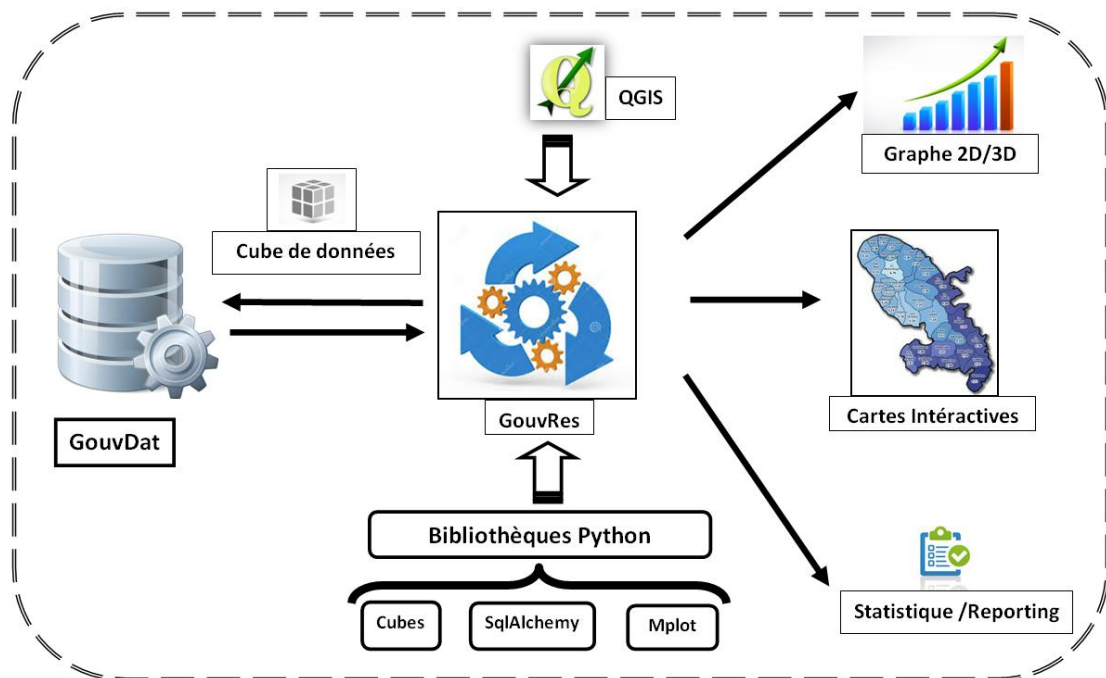


Figure 54 : Architecture de notre Prototype géodécisionnel



## **Chapitre 8 : fonctionnements du Prototype Géodécisionnel et Résultats**

L'objectif de ce travail est de proposer à l'utilisateur de notre prototype géodécisionnel un outil pouvant répondre à des requêtes comportant un nombre quelconque de dimensions.

Sachant qu'il n'existe actuellement aucun système (même parmi les plus puissants, tel que le moteur de recherche Google) capable d'effectuer l'analyse sémantique d'une requête complexe ; il est donc nécessaire de décomposer la requête primitive en un nombre suffisant de subrequêtes élémentaires dont la concaténation permettra de répondre à la requête initiale.

Dans notre SOLAP, ce sont les 4 principaux opérateurs d'agrégations spatiaux (décrits dans la section 1.3.2) qui vont œuvrer à cette tâche afin d'établir un cube logique qui permettra d'interroger la base de données.

### **8.1 requête relative au choix d'un site pour l'installation d'un émetteur –récepteur BTS.**

Voici à titre d'exemple une requête complexe dont l'objectif est de déterminer l'emplacement d'un nouveau BTS dans la ville de Tiflet, région de Rabat, dans une zone d'habitat rural, dont la surface est à 1100 m<sup>2</sup>, possédant un centre commercial, et dont la population est de l'ordre de 700 Personnes, les immeubles d'habitations sont de type R4 ; ce BTS doit se situer à une distance inférieure à 1Km d'un BTS existant et ayant enregistré 10 alarmes durant l'année 2012.

Nous allons présenter en détail les différentes fenêtres que notre prototype géodécisionnel propose. De plus, à chaque étape, nous exposerons les actions qui s'accomplissent en arrière-plan de manière transparente.

### 8.1.1 Description de la page principale.

L'icône relative au Prototype géodécisionnel s'affiche au centre de la fenêtre d'accueil (Figure 55).



Figure 55 : Prototype développé

Après le démarrage de notre prototype géodécisionnel, la fenêtre principale est affichée (Figure 56) et propose un menu permettant un choix entre 3 types de requêtes.

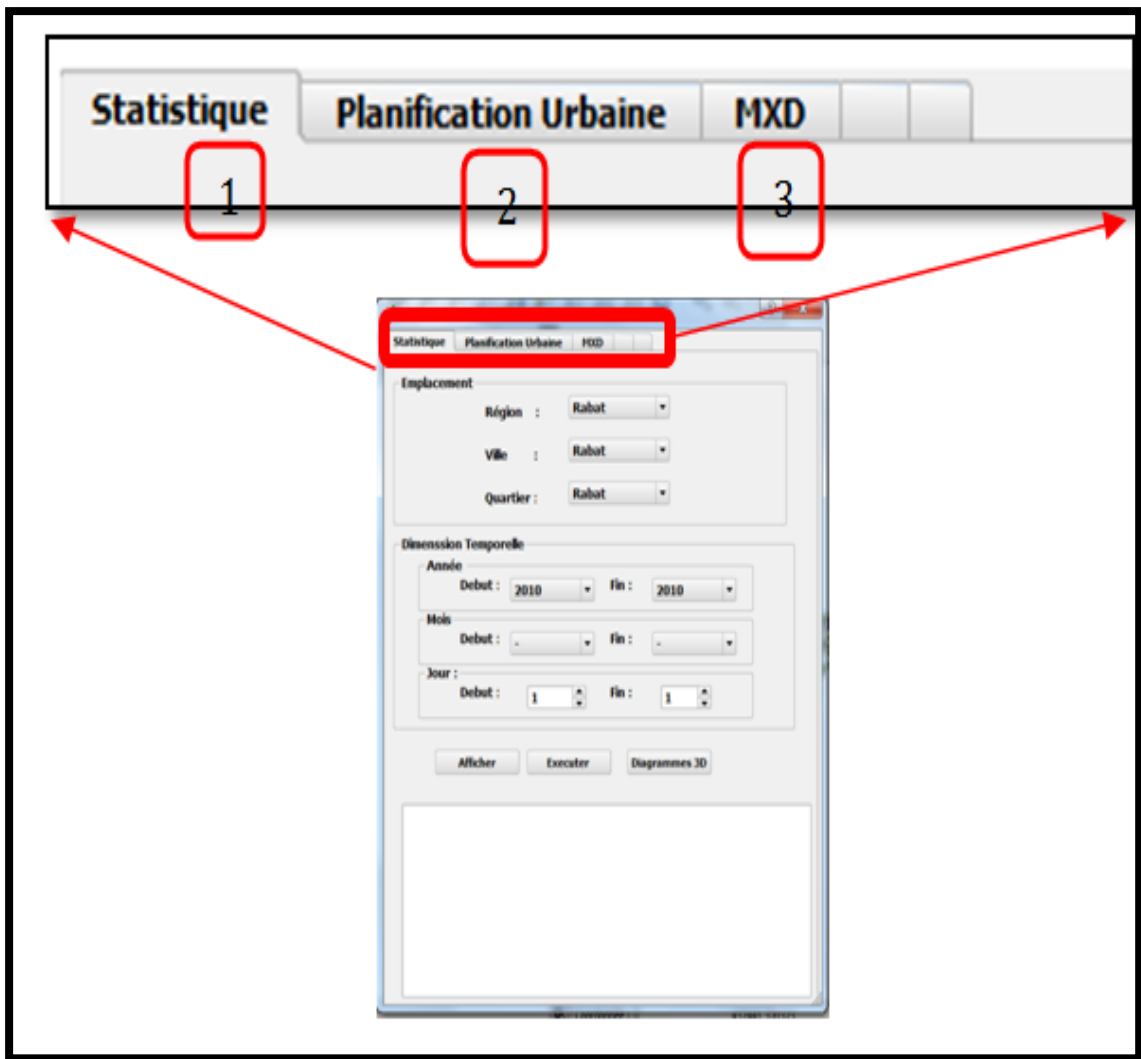


Figure 56 : Fenêtre principale du prototype

*La fenêtre statistique (1) : est la page dédiée aux différentes statiques. La fenêtre Planification urbaine (2) : est la page de la planification urbaine. La fenêtre MXD (3) : est la page consacrée à l'exécution des requêtes multidimensionnelles libres rédigées par l'utilisateur.*

Afin d'illustrer notre exemple, l'option (2) planification urbaine est sélectionnée ; car tout nouvel aménagement impactant le territoire astreint l'opérateur à se référer au plan d'urbanisme du secteur considéré. Dans cette perspective, les données concernant cette planification ont donc été préalablement modélisées dans un magasin « Planification urbaine » de l'EDS « GouvData » décrit dans la section 6.2.3. Rappelons que ce magasin comprend la dimension spatiale prévue par des plans d'aménagement ou des PDU (Plan de Développement Urbain) ou des PDAR (Plan de Développement des

Agglomérations Rurales) avec les dates d'homologation et d'expirations. Ils représentent les différentes entités urbaines existantes ou projetées. Par souci de clarté, ce magasin a été divisé en 5 dimensions : Îlot, équipement, réseau électrique, tissus urbains et hydrographie.

## 8.1.2 Fenêtre de Planification urbaine

La fenêtre Planification urbaine est divisée en 4 parties (Figure 57).

The screenshot shows a software window titled 'Planification Urbaine' with four distinct sections highlighted by red boxes and labeled on the right as 'Partie 1' through 'Partie 4'.  
**Partie 1:** 'Zones Urbanisables' section with the question 'Où voudrez installer votre nouveau BTS ??'. It includes a dropdown menu for 'Etude/Centre' (selected: TIFLET) and another for 'Couche' (selected: Secteur).  
**Partie 2:** 'Critères de la Zones à cherchée' section with the instruction 'Donner les paramètres de recherche de la zone'. It contains fields for 'Type' (dropdown), 'Nombre Etages' (R + 0), 'Population Projetée' (0 Hab), and 'Surface Min' (0 m2).  
**Partie 3:** 'Spatiale' section with the title 'les Criteres de Recherches de votre BTS'. It features a 'Buffer De Recherche' (0 m), 'Nbr Alarmes' (0), and a date range 'le Nombre des Alarmes Enregistré entre Les Deux Dates suivants :' with 'Debut' (2012-01-01) and 'Fin' (2012-01-01).  
**Partie 4:** 'Resultat' section, which is currently empty, with 'Bu' and 'Planifier' buttons.

Figure 57 : Fenêtre de Planification urbaine

**Partie 1 :** permet de choisir la zone urbanisable sur laquelle l'installation d'un BTS est souhaitable par exemple : dans une zone « Habitat économique de la ville de "TIFLET" »

**Partie 2 :** permet de raffiner de la recherche en précisant : le type, le nombre d'étages, population projetée, et la surface minimale en mètre carré

**Partie 3 :** est consacrée à la définition du Buffer (la zone tampon de recherche) et nombre d'alarmes enregistrées entre deux dates bien définies

**Partie 4 :** affichage des résultats

La formulation des requêtes (qui seront ensuite concaténées) s'effectue à l'aide des fenêtres suivantes.

### 8.1.2.1 Partie 1 de la fenêtre de planification urbaine

Cette partie (Figure 58) permet de sélectionner le centre urbain qui est l'objet de notre étude, au sein des centres urbains couverts par le document d'urbanisme enregistré dans notre Entrepôt de données spatial. À cet effet, deux listes déroulantes permettent de renseigner d'une part le nom du centre et d'autre part, la couche.

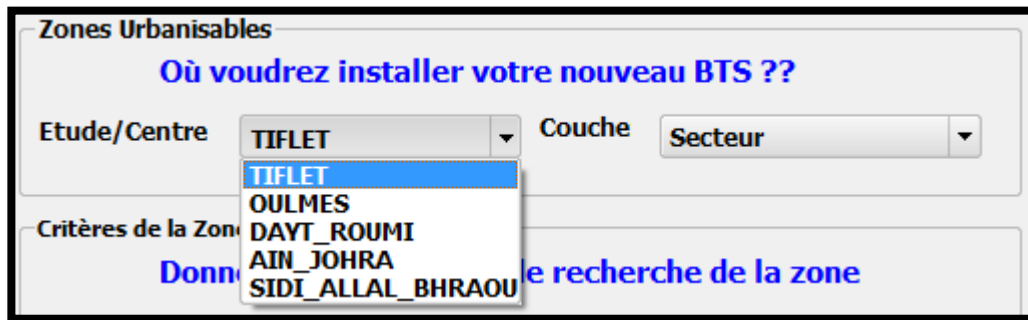


Figure 58 : Définition de l'objet étudié par l'intermédiaire de 2 listes déroulantes permettant de sélectionner le nom du centre urbain et la couche correspondante

Après avoir renseigné la première liste, la seconde liste affiche le zoning disponible sur le document d'urbanisme correspondant au centre urbain sélectionné sur la première liste (Figure 59)

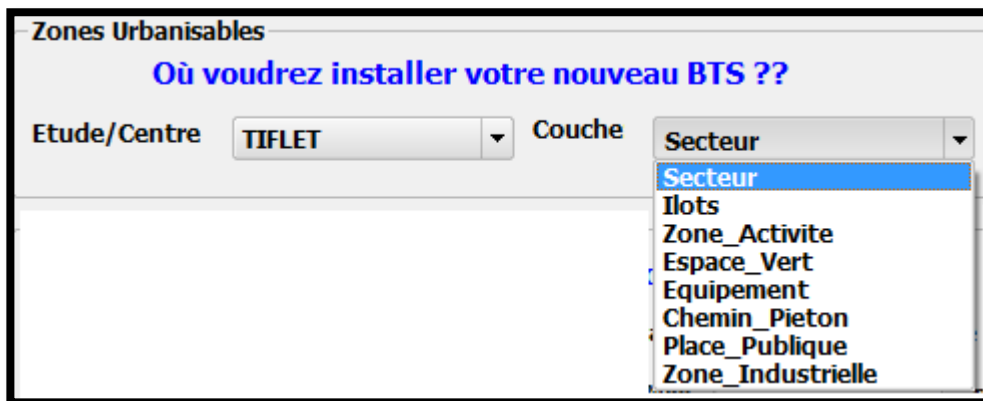


Figure 59 : Figure montrant le zoning du centre urbain Tiflet

Le renseignement de ces deux premiers champs (centre urbain et couche) entraîne la création d'une instruction de sélection dans le code de fichiers principal Python en fonction de ces deux paramètres.

Les autres paramètres de l'instruction de recherche seront renseignés, dans la deuxième partie de la fenêtre.

### 8.1.2.2 Partie 2 de la fenêtre de planification urbaine

Cette fenêtre permet de compléter les critères relatifs à la zone précédemment définie à savoir le nombre d'étages, la nature de l'habitat, la population prévisionnelle, la surface minimale de la zone (Figure 60).

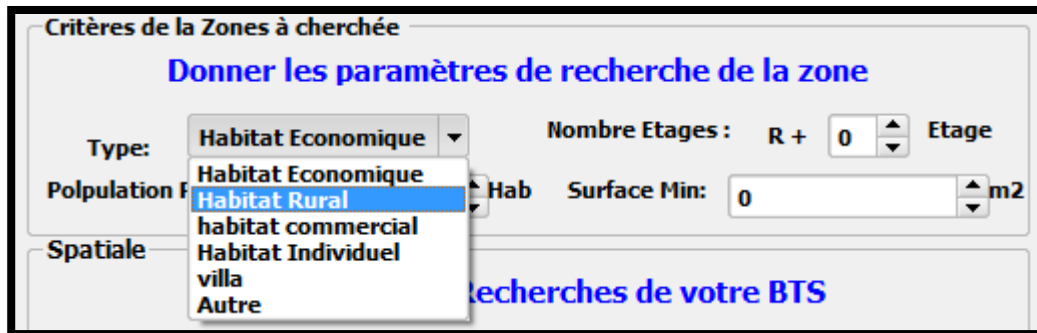


Figure 60 : Critères de recherche de la zone considérée

Dans le cas par exemple où l'utilisateur a décidé de choisir « Équipement » dans la liste déroulante relative à couches (Figure 59), le système lui propose la liste type suivante (Administration, centre commercial, Foyer féminin) (Figure 61).

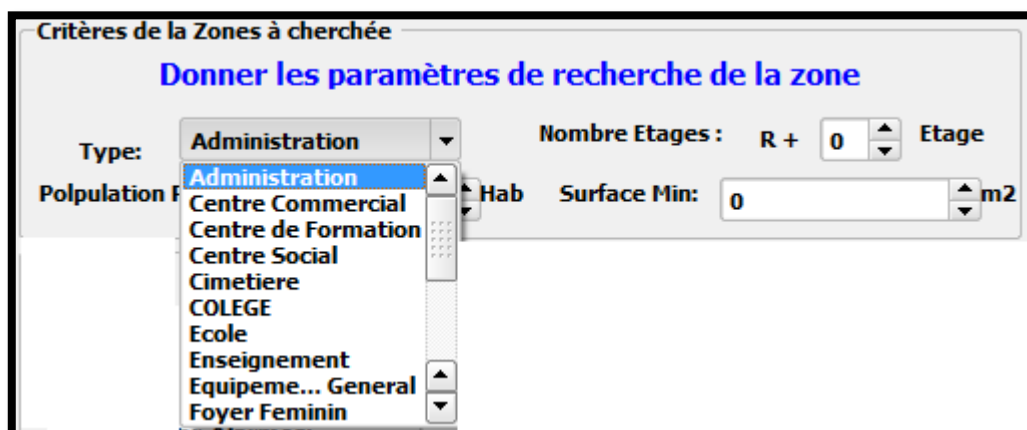


Figure 61 : Liste des items affichés relatifs à la couche Équipement

Afin de compléter les informations relatives au type « Centre commercial » à savoir : le nombre d'étages, la population et la surface de la zone ; la figure 62 permet de renseigner les valeurs relatives à ses sous types.

Critères de la Zones à cherchée

**Donner les paramètres de recherche de la zone**

Type: Centre Commercial

Nombre Etages : R + 4 Etage

Polpulation Projetée : 700 Hab

Surface Min: 1100 m2

Figure 62 : Fenêtre permettant de renseigner les critères relatifs au type centre commercial

### 8.1.2.3 Partie 3 de la fenêtre de la planification urbaine

On vient de définir les caractéristiques de la zone susceptible d'accueillir le nouveau BTS. Il convient maintenant de préciser le rayon de la zone autour du BTS en souffrance ; la souffrance étant caractérisée par le nombre d'alarmes enregistrées durant une période déterminée. Ces valeurs sont renseignées (Figure 63).

Spatiale

**les Criteres de Recherches de votre BTS**

Buffer De Recherche : 1000 m

Nbr Alarmes: 10

le Nombre des Alarmes Enregistré entre Les Deux Dates suivants :

Debut: 2012-01-01

Fin: 2013-01-01

Figure 63 : définition d'une zone tampon et des caractéristiques d'un antenne BTS en souffrance

Le renseignement de ces paramètres provoque la mise en œuvre des 3 opérations dans le cube :

- 1) un slice qui tranche l'année 2012 au sein de la dimension « Temps » du cube.
- 2) Un rolle UPS agrège ensuite les alarmes pour l'ensemble des BTS concernant la partie tranchée.
- 3) Un deuxième slice tranche les BTS dont les alarmes en 2012 ont été égales ou supérieures à 10.

Quatre BTS sont ainsi retenus : BTS\_OULMES\_2 ; BTS\_OULMES\_9, BTS\_OULMES\_12, BTS\_OULMES\_13.

De plus, un cercle tampon d'un kilomètre de rayon circonscrit chaque BTS. À titre d'exemple, la zone tampon relative au BTS\_OULMES\_2 est représentée (Figure 64). Cette zone est projetée sur un fond satellitaire.

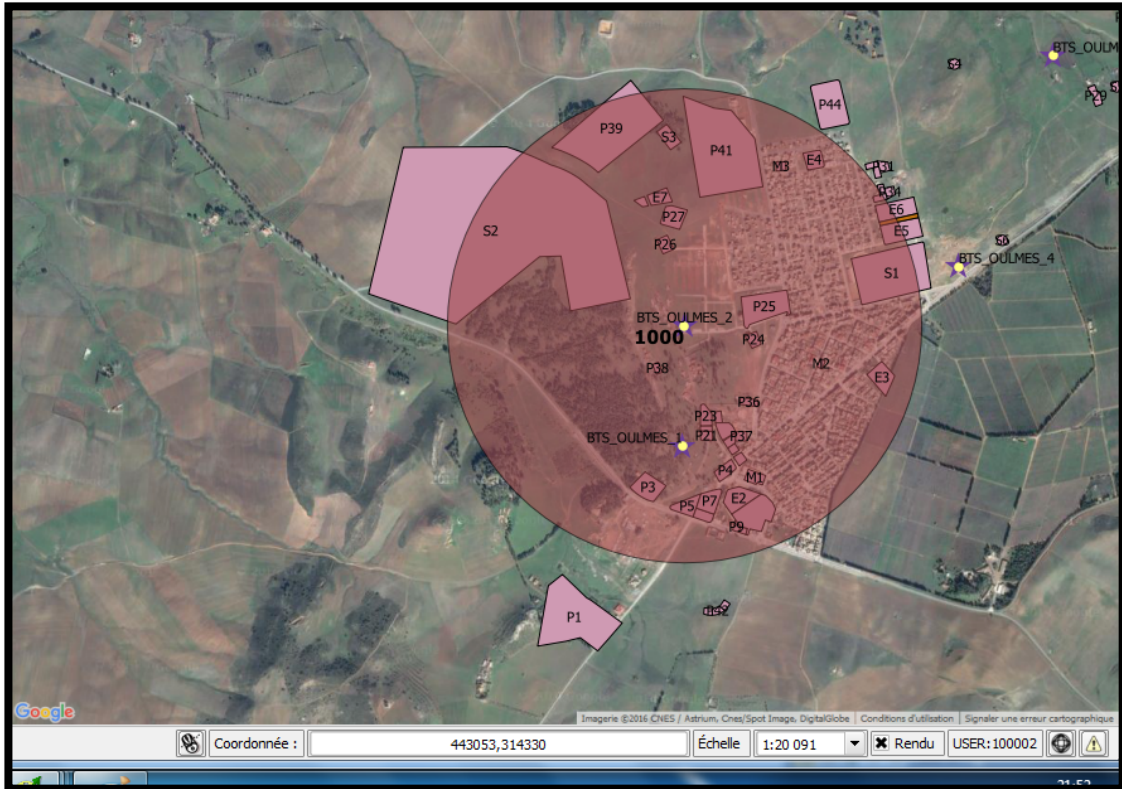


Figure 64 : Zone tampon du BTS\_OULMES\_2

#### 8.1.2.4 Partie 4 de la fenêtre planification urbaine

La dernière partie est consacrée à l'affichage des résultats, et selon les cas on peut définir trois types d'affichage :

- ❖ **1<sup>er</sup> Cas : Pas de résultat s'affiche, quand la requête définie par l'utilisateur ne correspond à aucune zone (Figure 65).**

**Exemple de requête sans résultat :**

Dans le centre de TIFLET, afficher les équipements de nature « Centre Commercial », avec « R +4 » ayant une surface minimale de 1000 m<sup>2</sup> ; situés dans un périmètre de 1Km d'un BTS qui a enregistré 10 alarmes pendant l'année 2012

The screenshot shows a software interface with the following sections:

- Zones Urbanisables:** Title "Où voudrez installer votre nouveau BTS ??". Fields: "Etude/Centre" (TIFLET), "Couche" (Equipement).
- Critères de la Zones à cherchée:** Title "Donner les paramètres de recherche de la zone". Fields: "Type" (Centre Commercial), "Nombre Etages" (R + 4 Etage), "Polpulation Projetée" (700 Hab), "Surface Min" (1100 m2).
- Spatiale:** Title "les Criteres de Recherches de votre BTS". Fields: "Buffer De Recherche" (1000 m), "Nbr Alarmes" (10), "le Nombre des Alarmes Enregistré entre Les Deux Dates suivants": "Debut" (2012-01-01), "Fin" (2013-01-01).
- Resultat:** A large red text area displaying "Pas de Résultat !!!!".

Figure 65 : Fenêtre affichant l'absence de résultat

## ❖ 2eme cas : Une seule entité répond à la requête

Dans ce cas le prototype sélectionne directement l'entité répondant à l'ensemble des critères définis par l'utilisateur.

### Exemple Requête aboutissant à un résultat unique :

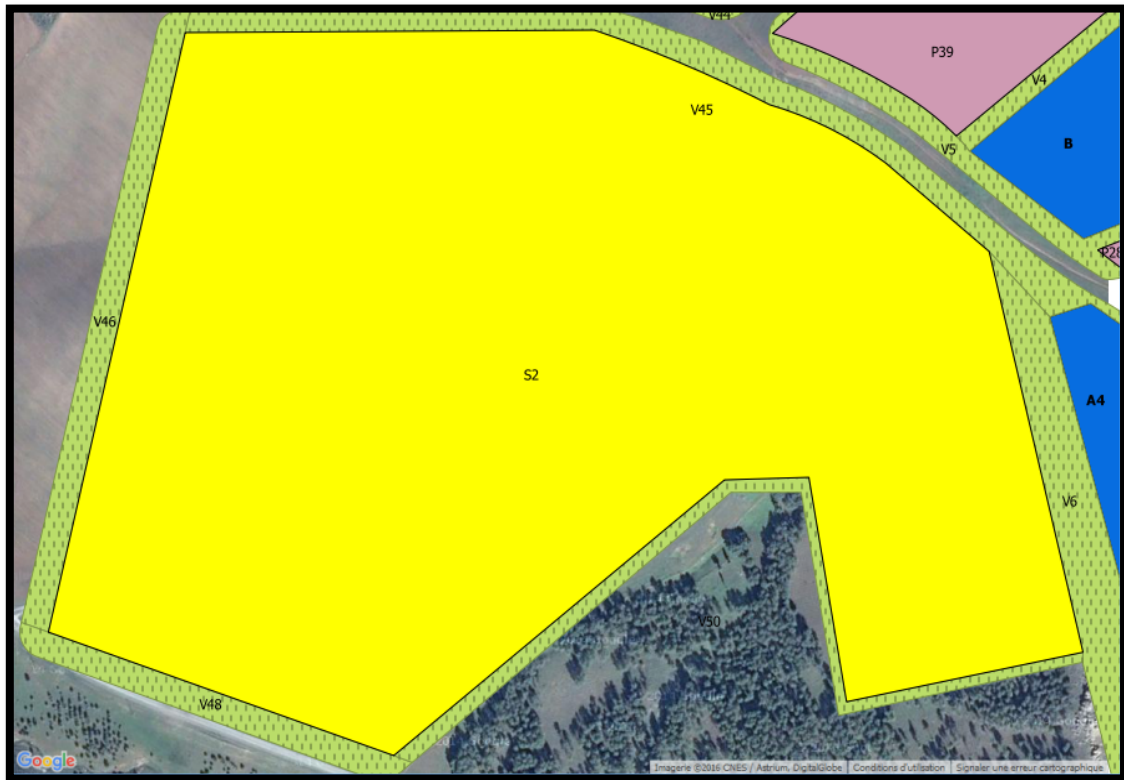
Dans le « centre OULMES », afficher les équipements de type « Santé » ayant une surface minimale de 70 000 m<sup>2</sup> ; situés dans un périmètre de 1Km d'un BTS qui a enregistré 10 alarmes pendant l'année 2012 (Figure 66)

The screenshot shows a software interface with the following sections and controls:

- Statistique** | **Planification Urbaine** | **MXD**
- Zones Urbanisables**  
**Où voudrez installer votre nouveau BTS ??**  
Etude/Centre: **OULMES** | Couche: **Equipement**
- Critères de la Zones à cherchée**  
**Donner les paramètres de recherche de la zone**  
Type: **Sante** | Nombre Etages: **R + 0** | Etage  
Polpulation Projetée: **0** Hab | Surface Min: **70000** m<sup>2</sup>
- Spatiale**  
**les Criteres de Recherches de votre BTS**  
Buffer De Recherche: **1000** m | **Jff**  
Nbr Alarmes: **10** | **per**  
le Nombre des Alarmes Enregistré entre Les Deux Dates suivants :  
Debut: **2012-01-01** | Fin: **2013-01-01**
- Resultat** | **Bu** | **Planifier**

Figure 66 : fenêtre affichant un résultat unique

La visualisation des résultats projetés sur un fond satellitaire Google Earth est représentée (Figure 67).



*Figure 67 : représentation de l'entité unique sur fond satellitaire Google Earth*

### ❖ 3eme cas : Plusieurs entités répondent à la requête

Dans le cas où plusieurs entités correspondent aux critères, et dans le but de réduire le nombre d'interrogations de données (couteuses en temps et en mémoire) tous les résultats sont récupérés dans un seul tableau. Il suffit alors de cliquer sur une ligne du tableau pour visualiser la zone correspondante.

#### Exemple Requête à résultats multiples

Dans le centre d'Oulmes, afficher les Espaces verts de type V (Verdure) ayant une surface minimale de 3300 m2 située sur une zone tampon de 1000 m de rayon autour de BTS ayant enregistré 10 alarmes dans le premier semestre de l'année 2012 (Figure 68)

Statistique Planification Urbaine MXD

Zones Urbanisables  
Où voudrez installer votre nouveau BTS ??

Etude/Centre: OULMES Couche: Espace\_Vert

Critères de la Zones à cherchée  
Donner les paramètres de recherche de la zone

Type: V Nombre Etages: R + 0 Etage  
Polpulation Projetée: 0 Hab Surface Min: 3300 m2

Spatiale  
les Criteres de Recherches de votre BTS

Buffer De Recherche: 1000 m  
Nbr Alarmes: 10

le Nombre des Alarmes Enregistré entre Les Deux Dates suivants :  
Debut: 2012-01-01 Fin: 2012-06-30

Resultat

Bu Planifier

Votre Requete a donner  
**43 Ent.**  
Raffiner la Recherche ou  
Selectionner une ligne su.

V	10536.8366954	331
V	4948.46073859	332
V	3418.18678975	333

Figure 68 : Fenêtre affichant des résultats multiples

La visualisation d'une ligne du tableau est représentée (Figure 69).

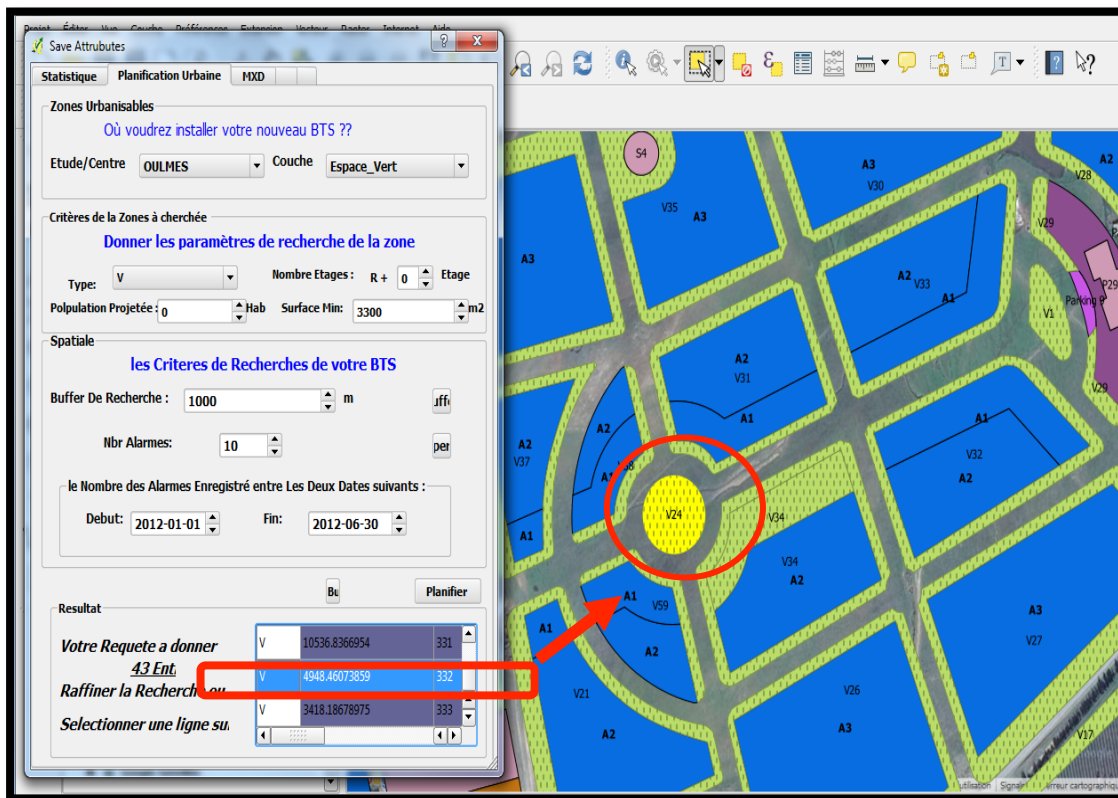


Figure 69 : Représentation de l'une des zones sélectionnées à partir du tableau

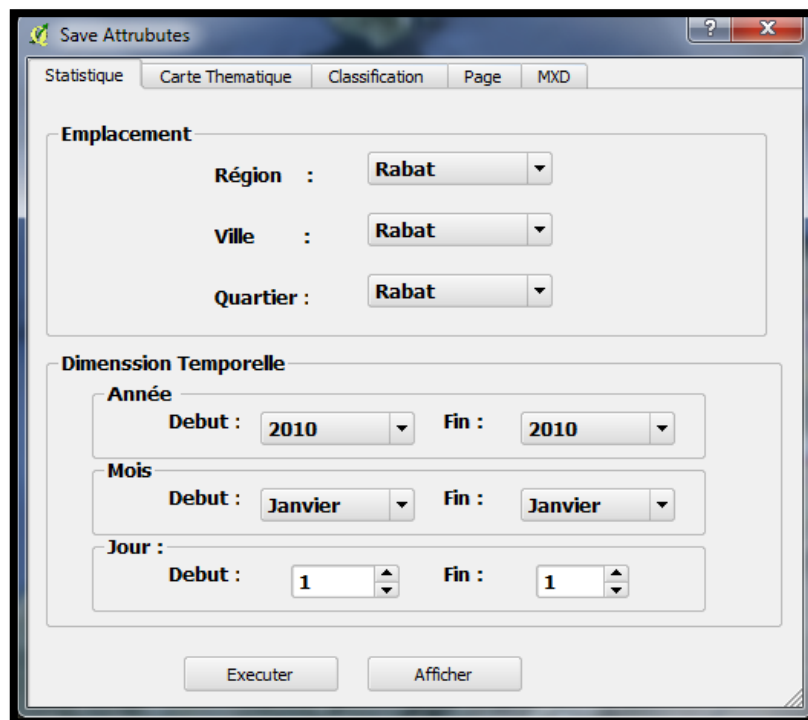
## 8.2 Requête de nature statistique

L'établissement de statistique s'effectue à partir de la « fenêtre statistique ».

### 8.2.1 Fenêtre des statistiques

Cette page (Figure 70) est consacrée aux différentes statistiques

Cette page, permet d'établir des requêtes statistiques relatives d'une part, au comportement des BTS et d'autres parts, à l'intensité du trafic dans différents environnements géographiques : région, ville quartier et dans des périodes de temps dont la granularité peut varier, de l'heure à l'année. Ces requêtes mettent en œuvre les opérateurs, Slice et Drill-Down.



The screenshot shows a software window titled "Save Attributes" with a tabbed interface. The active tab is "Statistique". Below the tabs, there are two main sections: "Emplacement" and "Dimension Temporelle".

**Emplacement:**

- Région : Rabat (dropdown menu)
- Ville : Rabat (dropdown menu)
- Quartier : Rabat (dropdown menu)

**Dimension Temporelle:**

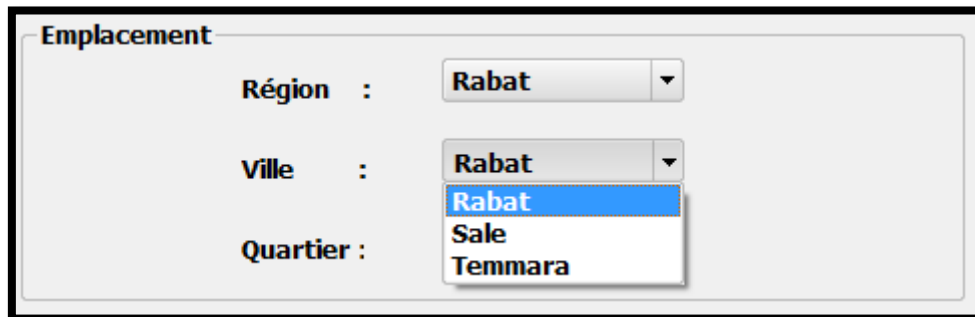
- Année:** Debut : 2010 (dropdown), Fin : 2010 (dropdown)
- Mois:** Debut : Janvier (dropdown), Fin : Janvier (dropdown)
- Jour:** Debut : 1 (spin box), Fin : 1 (spin box)

At the bottom of the window, there are two buttons: "Executer" and "Afficher".

Figure 70 : Fenêtre des statistiques

### 8.2.1.1 Choix de la dimension Spatiale

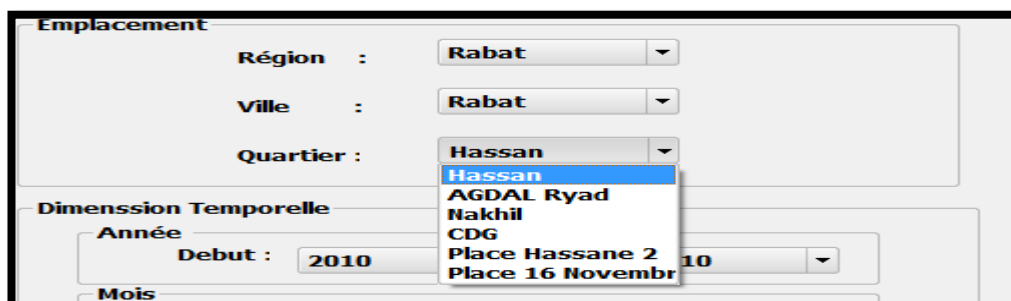
Dans la fenêtre « Emplacement », on choisit dans un premier temps la région « Rabat » ; puis la ville « Rabat » (Figure 71).



The screenshot shows a window titled "Emplacement" with three dropdown menus. The "Région" menu is set to "Rabat". The "Ville" menu is also set to "Rabat", and its dropdown list is open, showing "Rabat" as the selected item, with "Sale" and "Temmara" as other options. The "Quartier" menu is currently empty.

Figure 71 : Choix de la ville

Dans un deuxième temps, on sélectionne le quartier Hassan (Figure 72).



The screenshot shows the "Emplacement" window with the "Région" and "Ville" menus still set to "Rabat". The "Quartier" menu is now set to "Hassan", and its dropdown list is open, showing "Hassan" as the selected item, with "AGDAL Ryad", "Nakhil", "CDG", "Place Hassane 2", and "Place 16 Novembr" as other options. Below the "Emplacement" section, there is a "Dimension Temporelle" section with "Année" and "Mois" labels, and a "Debut" field set to "2010".

Figure 72 : Choix du quartier

### 8.2.1.2 Choix de la dimension temporelle

Par exemple, pour calculer la valeur moyenne du trafic mensuel pour l'année 2014, les champs début et fin de la figure 73 sont renseignés.

Dimension Temporelle

Année  
Debut : 2014 Fin : 2014

Mois  
Debut : Janvier Fin : Decembre

Jour :  
Debut : 1 Fin : 1

Figure 73 : Dimension temporelle

### 8.2.2 Résultats

#### 8.2.2.1 Affichage des résultats de la requête

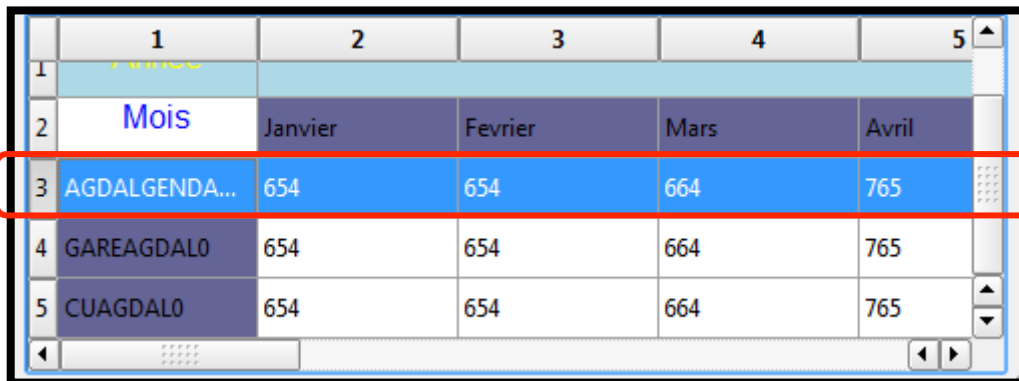
Après connexion à l'entrepôt de données spatial afin d'effectuer les opérations nécessaires en réponse à la requête, le résultat s'affiche sous forme d'un tableau (Figure 74).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Année	2014												
2	Mois	Jan...	Fev...	M...	Avril	Mai	Juin	Juillet	A...	Se...	Oc...	N...	D...	
3	AGDALGENDARMO	654	654	664	765	343	765	123	764	765	543	876	454	Graphe :0
4	GAREAGDALO	654	654	664	765	343	765	123	764	765	543	876	454	Graphe :1
5	CUAGDALO	654	654	664	765	343	765	123	764	765	543	876	454	Graphe :2

Figure 74 : Affichage des résultats de la requête « moyenne du trafic mensuel » pour chacun des BTS analysés

### 8.2.2.2. Zoom sur un BTS

En sélectionnant ensuite une ligne dans ce tableau de résultats (Figure 75).



	1	2	3	4	5
1					
2	Mois	Janvier	Fevrier	Mars	Avril
3	AGDALGENDA...	654	654	664	765
4	GAREAGDALO	654	654	664	765
5	CUAGDALO	654	654	664	765

Figure 75 : Exemple d'une ligne Sélectionnée (ligne 3)

### 8.2.2.3 Affichage du graphe sur un support satellitaire

Afin d'afficher un graphe bâtons sur l'image satellite, il suffit de cliquer sur le bouton Graphe (Figure 76).

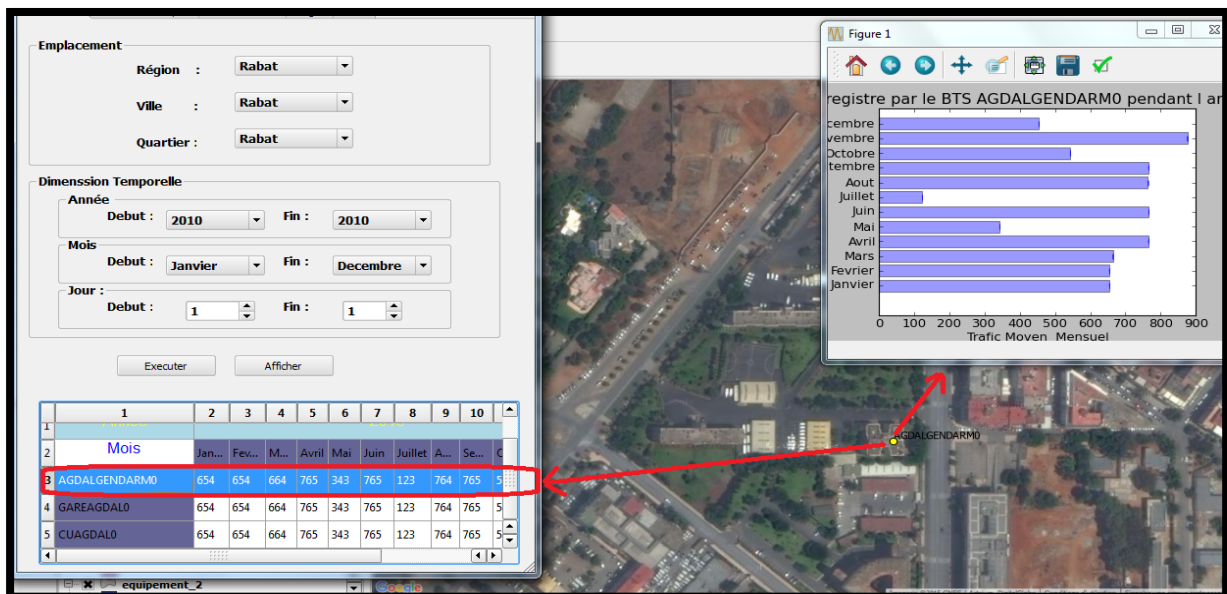


Figure 76 : Affichage d'un graphe en bâtons sur un fond satellite.

#### 8.2.2.4 Affichage d'un graphe en 3D pour une requête portant sur 4 années successives

Ce type de graphe permet d'effectuer une comparaison entre les valeurs d'un même mois sur une séquence de plusieurs années (Figure 77).

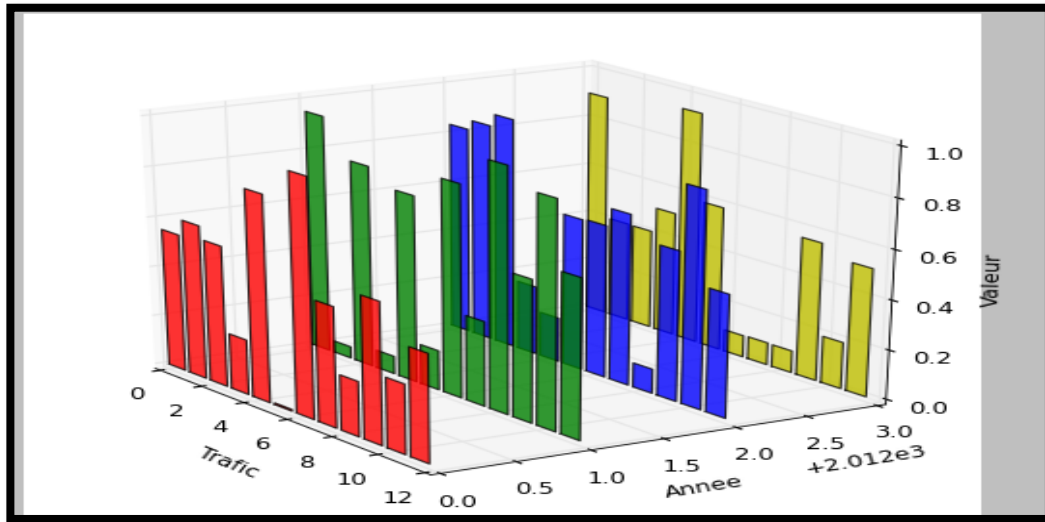


Figure 77 : Affichage d'un graphe en 3D en fonction des années

### 8.3 Fenêtre MXD

Cette page permet de formuler des requêtes personnalisées avec le langage spécialisé pour les requêtes multidimensionnelles MXD (Figure 78)

Le langage MDX, est un langage d'interrogation des bases multidimensionnelles plus adapté que le classique SQL pour le traitement des requêtes de type OLAP. MDX signifie « Multi-Dimensional Expressions ».

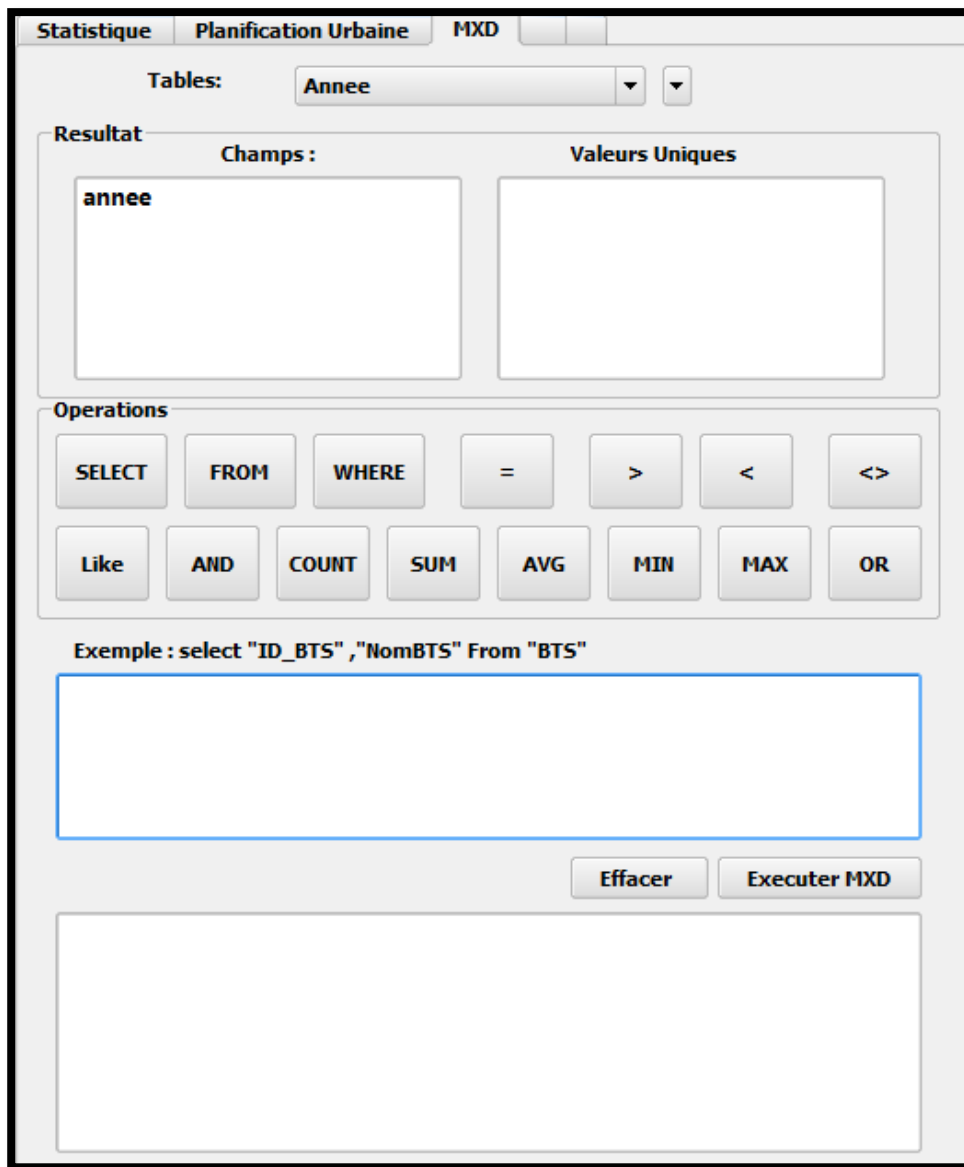


Figure 78 : Figure du langage MXD

### 8.3.1 Rédaction des Requêtes Personnalisées

Dans un premier temps, on liste l'ensemble des dimensions existant sur l'entrepôt de données spatial (Figure 79).

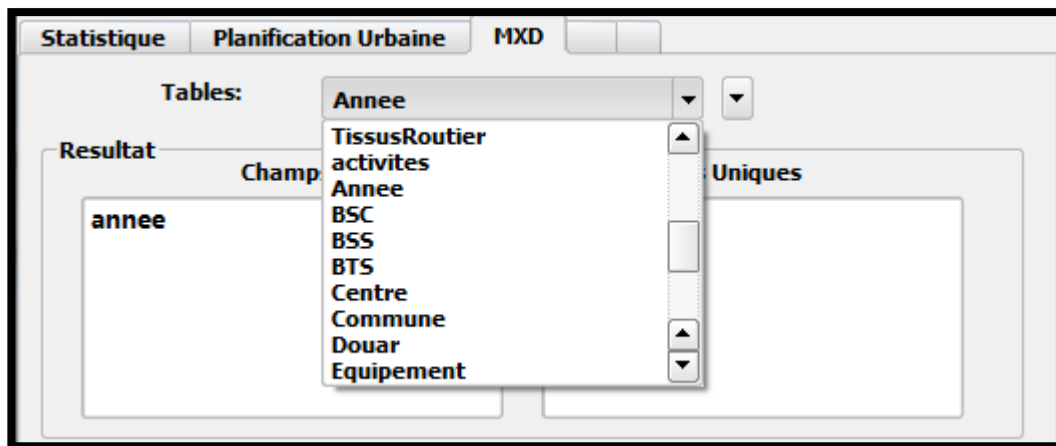


Figure 79 : Ensemble des Dimensions existant sur l'entrepôt de données spatial

Après avoir choisi la dimension souhaitée, dans ce cas « BTS », le système affiche l'ensemble des champs qui existent sur la dimension BTS (Figure 80).

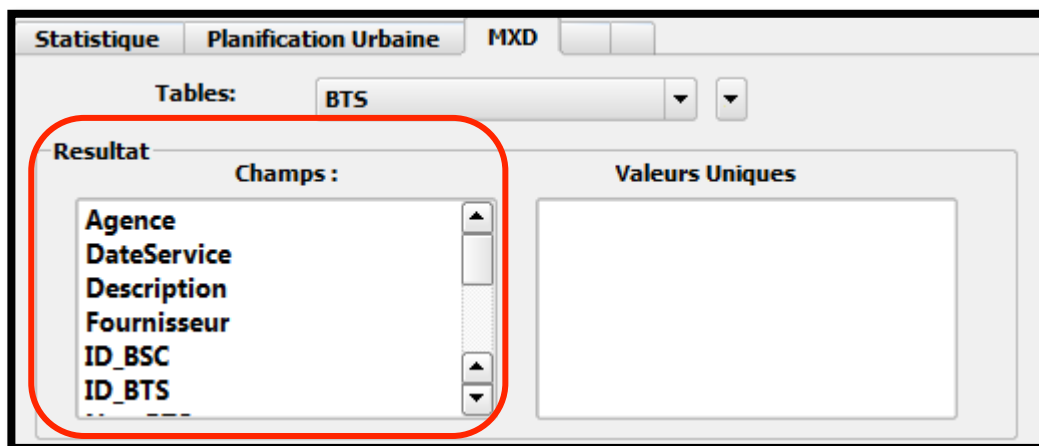


Figure 80 : Liste des champs

En double cliquant sur un champs (par exemple : NomBTS) une liste des valeurs uniques correspondant à ce champs s'affiche (Figure 81).

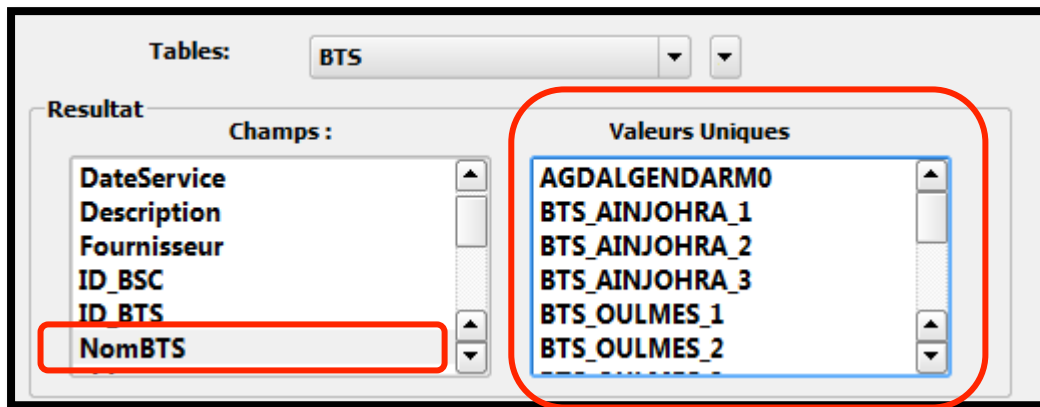


Figure 81 : Liste des valeurs uniques

Cette partie est un rédacteur semi automatique des requêtes MXD. Elle permet à un utilisateur ayant des notions de base dans le langage SQL, de rédiger une requête complexe en tout sécurité. Il suffit de choisir le bouton correspondant afin que l'instruction s'insère automatiquement (Figure 82).

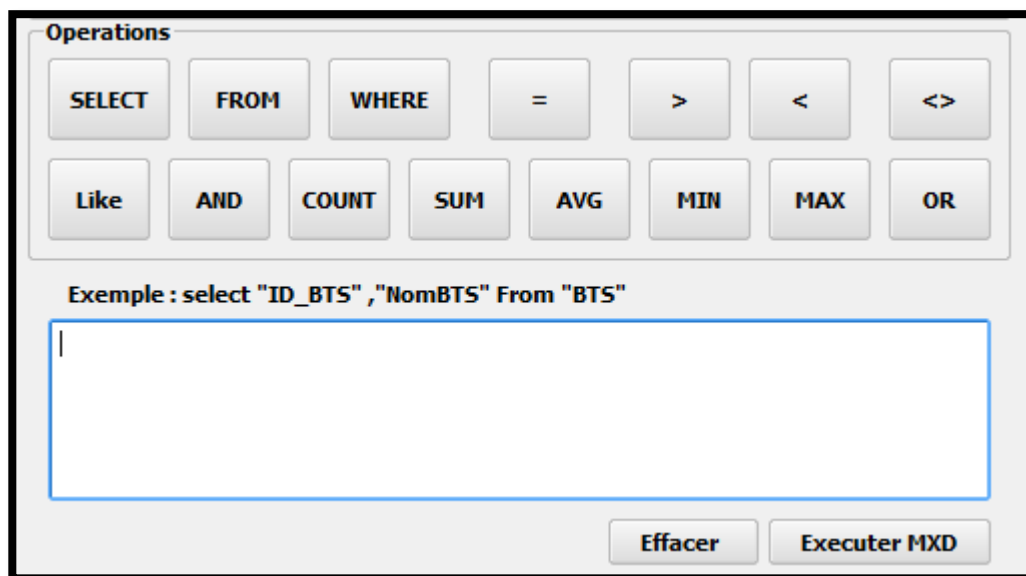


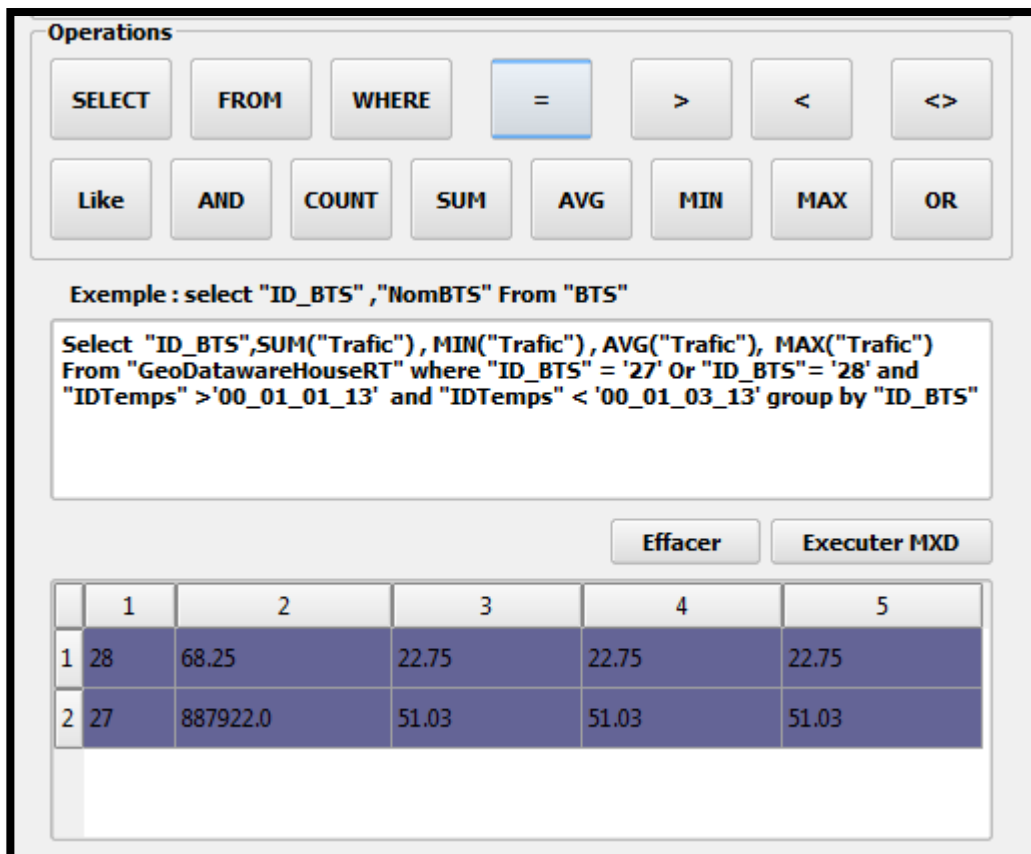
Figure 82 : Rédaction Semi-automatique des requêtes MXD

### 8.3.1.1 Requête relative au comportement des BTS

#### Exemple Requête

Dans la période comprise entre le premier janvier 2013 à minuit et le premier mars 2013 à minuit ; afficher : la somme, la moyenne et les valeurs minimales et maximales de la densité du trafic concernant les BTS dont les identifiants sont 27 et 28 (Figure 83)

Le résultat de cette requête apparaît sur la figure 83.



The screenshot shows a query execution interface with a toolbar of SQL operators (SELECT, FROM, WHERE, =, >, <, <>, Like, AND, COUNT, SUM, AVG, MIN, MAX, OR). Below the toolbar, an example query is displayed in a text area. At the bottom right, there are buttons for 'Effacer' and 'Executer MXD'. Below the interface is a table with 5 columns and 2 rows of data.

```
Exemple : select "ID_BTS" , "NomBTS" From "BTS"

Select "ID_BTS",SUM("Trafic") , MIN("Trafic") , AVG("Trafic") , MAX("Trafic")
From "GeoDatawareHouseRT" where "ID_BTS" = '27' Or "ID_BTS"= '28' and
"IDTemps" >'00_01_01_13' and "IDTemps" <'00_01_03_13' group by "ID_BTS"
```

	1	2	3	4	5
1	28	68.25	22.75	22.75	22.75
2	27	887922.0	51.03	51.03	51.03

Figure 83 : Affichage d'une requête MXD suivi du tableau de résultats

### 8.3.1.2 Requête relative à la densité de trafic

Requête :

Effectuer, une requête MXD permettant de récupérer la valeur moyenne de la densité du trafic pour les BTS de la région de Rabat, pendant l'année 2013 (Figure 84)

The screenshot shows a software interface for creating queries. At the top, there are tabs for 'Statistique', 'Planification Urbaine', and 'MXD'. Below the tabs, a 'Tables:' dropdown menu is set to 'BTS'. The interface is divided into two main sections: 'Resultat' and 'Operations'. The 'Resultat' section has two columns: 'Champs:' and 'Valeurs Uniques'. Under 'Champs:', a list of fields is shown: 'Agence', 'DateService', 'Description', 'Fournisseur', 'ID\_BSC', and 'ID\_BTS'. The 'Operations' section contains buttons for SQL keywords and operators: 'SELECT', 'FROM', 'WHERE', '=', '>', '<', '<>', 'Like', 'AND', 'COUNT', 'SUM', 'AVG', 'MIN', 'MAX', and 'OR'. Below the operations section, there is an 'Exemple:' section with a text box containing the following SQL query:

```
select "GeoDataawareHouseRT"."ID_BTS",AVG("Trafic")
from "GeoDataawareHouseRT","Temps","BTS"
where "GeoDataawareHouseRT"."IDTemps"="Temps"."IDTemps" and
"GeoDataawareHouseRT"."ID_BTS"="BTS"."ID_BTS" and "BTS"."city" like 'Rabat'
and "Temps"."year"='2013'
group by "GeoDataawareHouseRT"."ID_BTS"
```

Figure 84: Requête MXD Affichant la valeur moyenne de la densité de trafic des BTS de la ville de Rabat, pendant l'année 2013

Afin de pouvoir visualiser les résultats de cette requête, il est nécessaire d'insérer dynamiquement ces résultats dans une nouvelle colonne « densMoy2013 » créée dans la table attributaire relative à la couche BTS (Figure 85).

		NomBTS	sord	sord	_BS	cid	fic_2	fic_1	n_r	ongi	lati	genc	miss	!Ser	crip	unt	state	city	latitude	longitude	DenMoy2013
0	1	MASSTRAO	0...	3...	B...	4...	4...	1...	T...	-6...	3...	R...	N...	N...	N...	M...	Rabat Salé Zemmour Zaer	R...	-6.88...	33.977	623
1	2	NAKHIL0	0...	3...	B...	4...	1...	1...	R...	-6...	3...	T...	N...	N...	N...	M...	Oriental	O...	-6.87...	33.959	289
2	3	AGDALGENDARMO	-6...	3...	B...	4...	4...	N...	R...	-6...	3...	T...	N...	N...	N...	USA	New York	J...	-6.85...	33.993	373
3	4	CENTRALAGDALO	0...	3...	B...	4...	9...	1...	R...	-6...	3...	Ai...	N...	N...	N...	M...	Rabat Sale Zemmour Zaer	R...	-6.84...	33.998	43
4	5	CJAGDALO	0...	3...	B...	4...	7...	N...	R...	-6...	3...	S...	N...	N...	N...	USA	Texas	S...	-6.8425	33.998	25
5	6	GAREAGDALO	0...	3...	B...	4...	2...	1...	R...	-6...	3...	B...	N...	N...	N...	M...	Khemisset	R...	-6.85...	34.001	90
6	7	RIADNAKHIL0	0...	3...	B...	4...	7...	1...	R...	-6...	3...	Si...	N...	N...	N...	M...	Rabat	T...	-6.86...	33.950	133

Figure 85 : Exportation des résultats dans la table attributaire

Sur l'onglet propriété de la couche, on indique que la surface des cercles représentant l'intensité du trafic sera proportionnelle aux valeurs figurant dans la colonne « DenMoy2013 » (Figure 86).

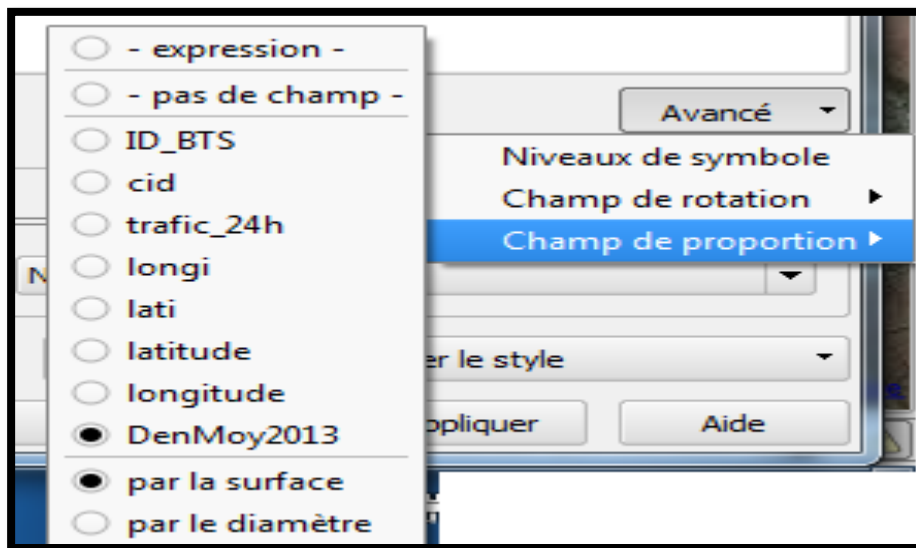


Figure 86 : La surface des cercles est proportionnelle à la densité relative du trafic des BTS de la région de Rabat

Le résultat de cette requête est affiché sur un fond satellitaire BigMap (Figure 87).

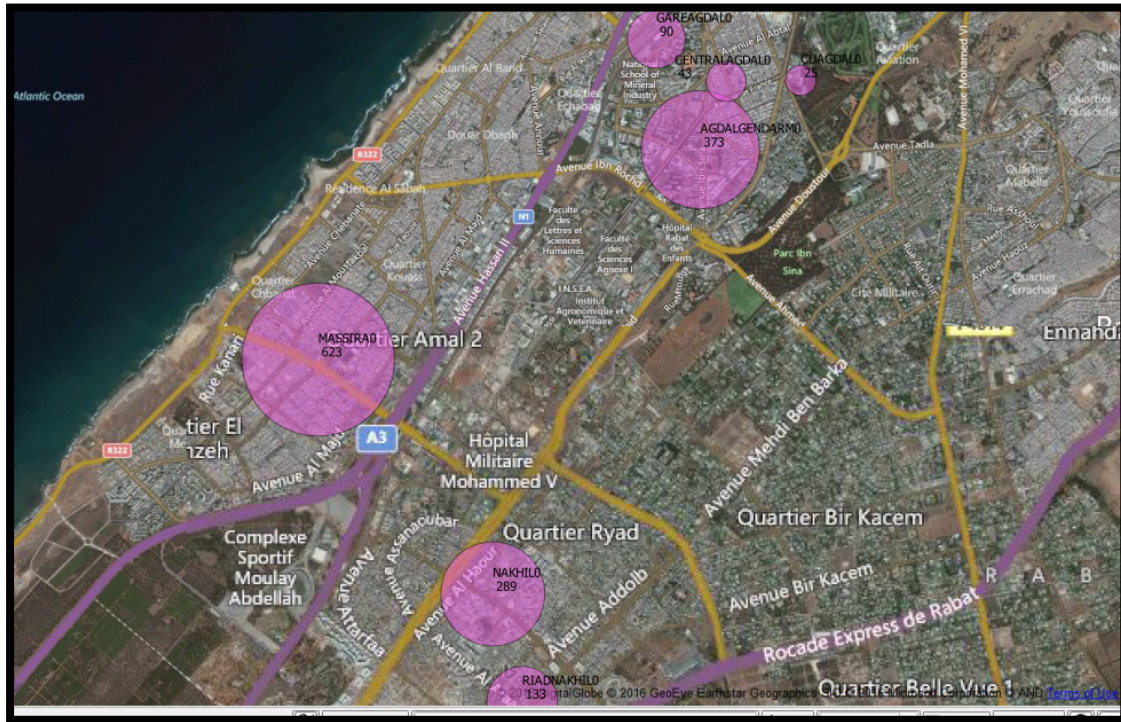


Figure 87 : Affichage sur une Image satellitaire « BingMap » de la densité relative du trafic des BTS de la région de Rabat pour l'année 2013

## 8.4 Affichage des résultats sur différentes plateformes

Notre prototype géodécisionnel est capable d'afficher l'ensemble des résultats sur tous les fonds satellitaires existant sur le marché à savoir : « Google », « Bing », « Apple Maps » (Figure 88). L'utilisation de l'un de ces fonds n'est pas limitative, ce qui permet d'utiliser sur un même document d'autres supports comme nous l'illustrons dans les figures 89 à 91.

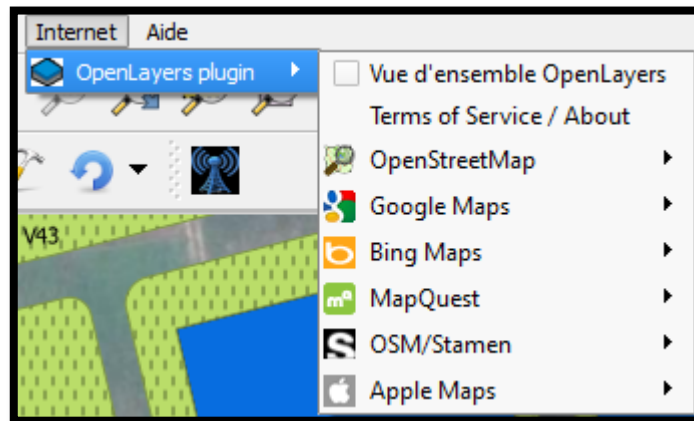


Figure 88 : Liste des fonds satellites disponibles



Figure 89 : Affichage des résultats sur un fond satellite BingMap

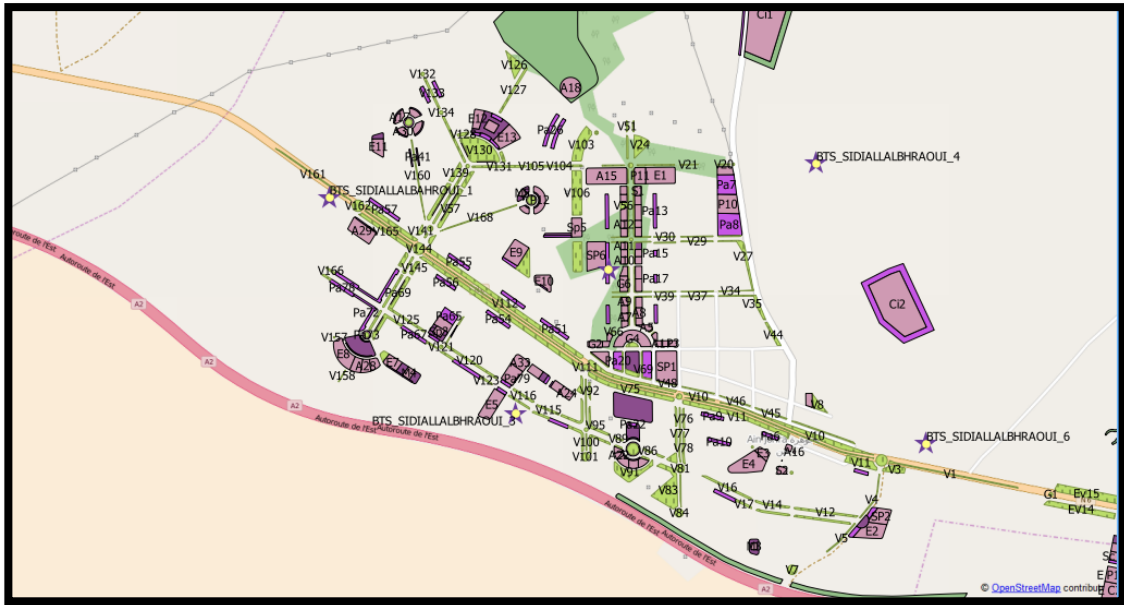


Figure 90 : Les résultats précédents sont affichés sur un fond OpenStreetMap



Figure 91 : Les résultats sont affichés sur un fond « Bing Aerial with labels »

## 8.5 Conclusion

À titre de conclusion, nous allons comparer les résultats obtenus à l'aide de notre SOLAP à ceux fournis par Pantaho (Figures 32 à 35).

<b>« GouvRes »</b>	<b>Pantaho</b>
Intègre les règles topologiques	N'intègre pas les règles topologiques
La représentation satellitaire peut se faire sur l'ensemble des fonds satellites existant sur le marché	Représentation sur deux fonds satellitaires prédéfinis
Complètement ouvert donc modifiable	Plus puissant en matière de graphes
Complètement ouvert (souple)	Complètement fermé (non modifiable)
Logiciel libre	Commercial à partir de la version 8.
SOLAP pouvant toujours être modifié en fonction des besoins de l'entreprise	SOLAP totalement fermé
SOLAP d'utilisation facile par des utilisateurs novices	SOLAP nécessitant des connaissances approfondies en informatique

Figure 92 : Tableau comparatif des propriétés respectives du Solap « GouvRes » développé et du Solap Pantaho(C)

- 1- Pantaho est un SOLAP généraliste qui exige que l'utilisateur potentiel possède un minimum de connaissance en informatique afin d'établir une connexion à une base de données spatiale. Par contre, avec « GouvRes », un utilisateur sans compétence particulière en informatique peut interroger notre EDS et obtenir immédiatement le résultat de sa requête.
- 2- En matière d'affichage graphique, « GouvRes » propose une offre moins large que Pantaho.
- 3- « GouvRES » est un SOLAP de type « SIG intégré ». Il bénéficie donc de la puissance complète d'un SIG, y compris les règles topologiques entre les entités de l'EDS, ce qui n'est pas le cas de Pantaho.

4— « GouvRes » offre la possibilité de projeter les résultats sur tous les supports existants (google/apple/Bing..), tandis que Pantaho est uniquement orienté googlemap et googlestreet.

5- « GouvRes » est issu de l'open source, de ce fait l'accès aux codes source permet de remodeler le système en fonction des besoins spécifiques de l'entreprise. Tandis que Pantaho est un système clos donc non modifiable.

A decorative graphic consisting of a vertical line on the left and a horizontal line below the text, intersecting at the start of the text.

# **PARTIE III**

## **CONCLUSION**



## **Chapitre 9 : conclusion générale et perspective**

Cette thèse avait pour propos de mettre à la disposition des opérateurs de téléphonie mobile un prototype permettant une bonne gouvernance du réseau d'accès de la téléphonie mobile. Dans cette perspective, notre prototype permet d'énoncer et de visualiser des requêtes complexes, à toutes les échelles sur des fonds cartographiques ou satellitaires. Ces requêtes sont de deux ordres :

- Réalisation d'analyses statistiques relatives aux fluctuations spatiotemporelles du trafic radio transitant par les antennes émettrices-réceptrices du réseau d'accès à l'échelle d'un quartier, d'une ville, d'une région et par extension d'un pays entier.
- Prévisions d'amélioration et d'implantation de nouvelles antennes en tenant compte des différents plans d'aménagement du territoire.

### **9.1 Discussion**

Cette section a pour objet d'éclairer les motivations tactiques et stratégiques qui ont sous-tendu la conception de notre « Prototype Géodécisionnel ».

#### **9.1.1 Open source**

Sans les outils disponibles en Open Source, cette thèse n'aurait pas pu être réalisée ; car les outils propriétaires proposés sur le marché ne peuvent subir aucune modification. De plus, leur coût prohibitif excluait à priori que nous puissions les acquérir. En conséquence, nous avons sélectionné dans la nébuleuse Open Source l'ensemble des outils nécessaires à la création de notre EDS « GouvData » et de notre SOLAP « GouvRes ». De ce fait, nous avons pu réaliser un prototype géodécisionnel aussi souple et performant que ceux commercialisés par des sociétés de service. De plus, notre outil étant libre de frais commerciaux, son utilisation permet de réaliser sur le long terme d'importantes économies d'échelles ; à condition que les mises à jour soient réalisées par les informaticiens de l'entreprise.

### 9.1.2 Conception de l'entrepôt de données Spatial « GouvData »

Dans un contexte multidimensionnel, la conception logique et la réalisation d'un entrepôt de données spatial, nécessite en premier lieu de définir l'architecture générale de l'entrepôt. Cette architecture est essentiellement conditionnée par le volume de données à enregistrer et par le nombre d'axes d'analyse (dimensions) envisagés.

Dans cette optique il a paru souhaitable d'opter pour une architecture comportant 4 dimensions ; elles-mêmes subdivisées en sous-dimensions ; le tout organisé sur le modèle en flocon autour d'une table de faits.

Les spécifications de chaque dimension sont enregistrées dans 4 magasins.

Ce dispositif offre l'avantage de pouvoir adjoindre de nouvelles dimensions au modèle logique initial de l'entrepôt, en ajoutant simplement de nouveaux magasins. Ceci évite de bouleverser l'architecture générale de l'EDS.

Par ailleurs, la structure en flocon réduit sensiblement le temps de réponse aux requêtes.

Après avoir créé l'entrepôt de données spatial, il était indispensable de vérifier ses capacités opérationnelles avant d'entreprendre le développement de notre SOLAP. Ce test a été réalisé à l'aide du SOLAP Pantaho dont l'utilisation complète en open source est limitée par des brevets commerciaux.

### 9.1.3 Développement de « GouvRes »

La mise au point de notre SOLAP représente le cœur de notre prototype décisionnel ; car cette plateforme en est le chef d'orchestre (Figure 54). Son rôle consiste à interroger notre EDS à travers un ensemble de requêtes dont nous avons organisé les marches à suivre ; et à visualiser le résultat des requêtes. Elle repose sur l'assemblage d'un outil SIG, en l'occurrence **QGIS remodelé**, et des bibliothèques Python contenant toutes les fonctions d'un OLAP.

QGIS, est une plateforme SIG, totalement, Open Source. C'est l'une des plateformes SIG les plus abouties de la nébuleuse Open Source. Ses performances sont équivalentes à celle d'Arcgis et de Mapinfo. C'est la raison pour laquelle nous l'avons retenue afin de développer notre SOLAP

« GouvRes ». Ce sont les mêmes motifs qui nous ont incités à faire appel aux bibliothèques Python, d'autant que le langage Python est le seul compatible avec QGIS.

### **9.1.3.1 Compilation et remodelage de QGIS**

La compilation et les modifications apportées à ce logiciel ont constitué la partie la plus ardue et la plus longue de ce travail.

La compilation consiste à télécharger le code source sur internet et ensuite à choisir une plateforme de compilation. À la suite d'une série de tentatives infructueuses, la plateforme QtCreator a été retenue afin d'accéder à l'ensemble des fichiers constitutifs de ce logiciel. Les détails de cette compilation sont illustrés (annexe 5).

Les modifications ont été portées aux classes et aux fonctions ; par exemple suppression de commandes inutiles ; ajouts de classes avec leurs attributs et leurs méthodes, suppressions et remaniement des attributs et des instructions de certaines classes.

### **9.1.3.2 Bibliothèques Python**

La prise en main de l'ensemble de ces bibliothèques a également nécessité un travail d'élaboration long et conséquent, en particulier la mise en œuvre de la bibliothèque « Cube » contenant les opérateurs de navigation OLAP, et la création d'un fichier d'extension (Json). Ce dernier contient le format logique de données et requiert une écriture extrêmement rigoureuse des différentes dimensions et hiérarchies qui représentent le cube multidimensionnel de données.

### **9.1.4 Résultats**

Au regard des aspects pratiques relatifs au maniement de ce prototype, nous avons porté une attention particulière à l'élaboration :

- d'un assistant secondant par palier l'utilisateur dans l'énonciation de requêtes consistantes et transparentes ;
- une visualisation des résultats très souples et sur de multiples supports cartographiques et satellitaires.

#### **9.1.4.1 Assistant de requêtes**

Cet assistant permet d'introduire des requêtes dans 3 contextes différents :

- 1) Établissement de statistiques spatiotemporelles relatives au trafic radio, permettant par exemple d'identifier les BTS qui atteignent trop fréquemment un seuil de saturation qui les rend impropres à soutenir un trafic constant et régulier. Les résultats sont affichés sous forme de tableaux.
- 2) Détermination des sites propices à l'implantation de nouvelles antennes émettrices-réceptrices en tenant compte d'une manière extrêmement détaillée de tous les aspects des plans d'aménagement du territoire.
- 3) Libre établissement de requêtes en ayant recours au langage MXD (Langage de Requêtes multidimensionnelles).

#### **9.1.4.2 Visualisation**

La visualisation des résultats constitue un atout primordial à la prise de décision. Notre prototype offre de multiples possibilités d'affichage par exemple :

- Les résultats statistiques sous forme de graphes en bâtons en 2D ou 3 D. Dans ce dernier cas, l'intensité mensuelle moyenne du trafic d'un BTS peut être observée simultanément pour chaque mois d'une année, et sur plusieurs années consécutives ;
- Le tableau de résultats statistiques ou les graphes en bâton peuvent être projetés et localisés sur des fonds topographiques ou satellitaires à différentes échelles ;
- L'intensité relative du trafic des BTS pour une période et un secteur géographique déterminés peut être projetée sur des fonds satellitaires;
- Les sites propices à l'installation de nouveaux BTS peuvent être visualisés sur le plan d'aménagement à différentes échelles et projetés sur des fonds satellitaires.

### **9.1.5 Évolution du Prototype**

Ce prototype étant un système open source conçu avec des langages de programmation orientés objet « Python » et C++ ; il possède donc une très large souplesse. Ce qui permet son adaptation immédiate aux nouvelles contraintes imposées par les orientations stratégiques impliquant des réformes en profondeur des infrastructures. Ainsi le passage à la technologie tout internet (full IP), nécessite une transformation des infrastructures aériennes, à savoir le remplacement des « BTS » par les « noteB » et les « eNoteB ».

## **9.2 Perspectives**

S'agissant d'un prototype, de nombreuses perspectives sont envisageables pour son enrichissement.

En ce qui concerne l'enrichissement des dimensions permettant d'affiner l'analyse multidimensionnelle, on peut prévoir l'adjonction de nouvelles couches de données telles que le MNT (Modèle Numérique de Terrain) et le MNE (Modèle Numérique d'Élévation) permettant ainsi de représenter le patrimoine en 3D.

Sur un plan conceptuel, on pourrait songer à la refonte complète du concept d'analyse multidimensionnelle en introduisant le paradigme de l'OLAP géographique qui prend en compte la composante spatiale et sémantique de l'information géographique. Il est également possible d'enrichir ce travail en affectant à la dimension spatiale la valeur d'un fait.

Par ailleurs, il serait possible de développer une version Web et une version légère pour les Smartphones et les tablettes.



# **ANNEXES**



# Annexe 1 : Création de l'Entrepôt de Données Spatial sous PostgresSQL

```
-- Database: "DataWarehouseReseauTelecom"
```

```
-----  
-- DROP DATABASE "DataWarehouseReseauTelecom";
```

```
CREATEDATABASE "DataWarehouseReseauTelecom"
```

```
WITHOWNER = postgres
```

```
ENCODING = « UTF8'
```

```
TABLESPACE = pg_default
```

```
LC_COLLATE = « French_France.1252'
```

```
LC_CTYPE = « French_France.1252'
```

```
CONNECTIONLIMIT = -1 ;
```

```
CREATE TABLE « DOUAR »  
(  
  « Nom_Dour » charactervarying (100),  
  « Id_Ville » charactervarying (50),  
  "Id_Douar" charactervarying (50) NOT NULL,  
  CONSTRAINTprimary_keyPRIMARY KEY ("Id_Douar")  
)  
WITH (  
  OIDS=FALSE  
)  
);  
ALTER TABLE « DOUAR »  
OWNER TO postgres;
```

## 2— Création de la table Année

```
CREATE TABLE "Annee" (annee charactervarying (20) NOT NULL,  
  CONSTRAINT "primarykeyAnnee" PRIMARY KEY (annee) ) WITH (  
  OIDS=FALSE ); ALTER TABLE "Annee" OWNER TO postgres;
```

### 3— Création de la table BTS

```
CREATE TABLE « BTS » ("ID_BTS" integer NOT NULL DEFAULT
nextval('bts_rabat_gid_seq'::regclass), "NomBTS" character_varying(254),
coordx character_varying(254), coordy character_varying(254), "ID_BSC"
character_varying(254), cid numeric, trafic_24h numeric, trafic_bh date,
nom_msc character_varying(254), longi numeric, latin numeric,
geom geometry(Point), "Agence" character_varying(20), "Fournisseur"
character_varying(50), "DateService" date, "Description"
character_varying(100), country character_varying(50), state
character_varying(50), city character_varying(50), latitude real, longitude real,
CONSTRAINT bts_rabat_pkey PRIMARY KEY ("ID_BTS"), CONSTRAINT
"cleEtrangeBTS" FOREIGN KEY ("ID_BSC") REFERENCES "BSC"
("ID_BSC") MATCH SIMPLE ON UPDATE NO ACTION ON DELETE
NO ACTION ) WITH ( OIDS=FALSE ); ALTER TABLE "BTS" OWNER
TO postgres; -- Index: bts_rabat_geom_gist -- DROP INDEX
bts_rabat_geom_gist ; CREATE INDEX bts_rabat_geom_gist ON "BTS"
USING gist (geom);
```

## Annexe 2 : Ajout des données à l'entrepôt de données Spatial

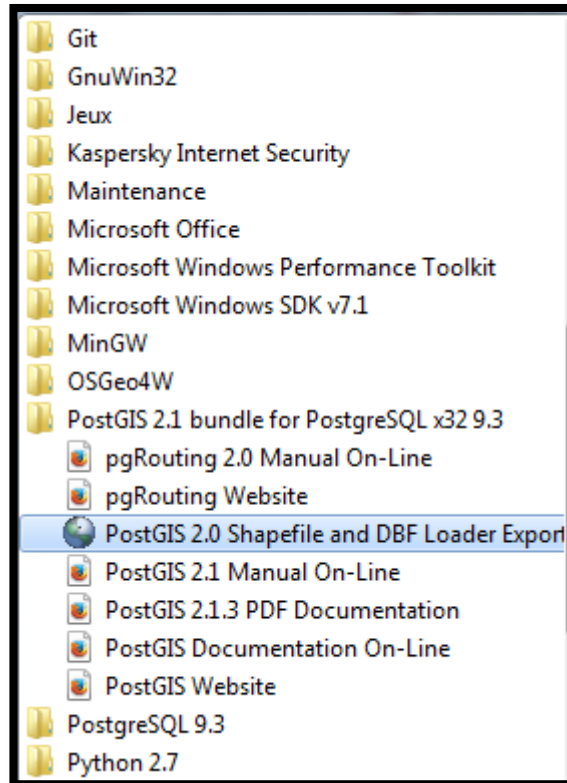


Figure 93 : Intégration des données avec PostGIS

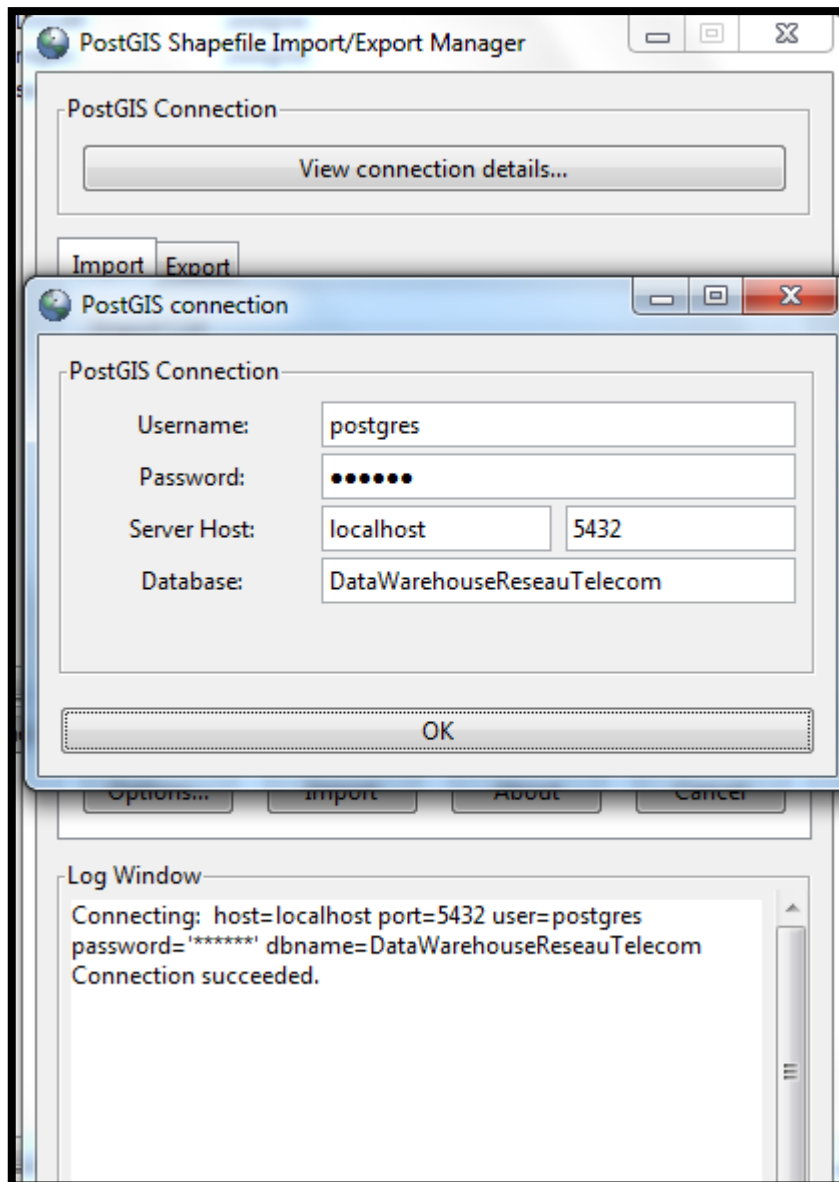


Figure 94 : Connexion à l'Entrepôt de données

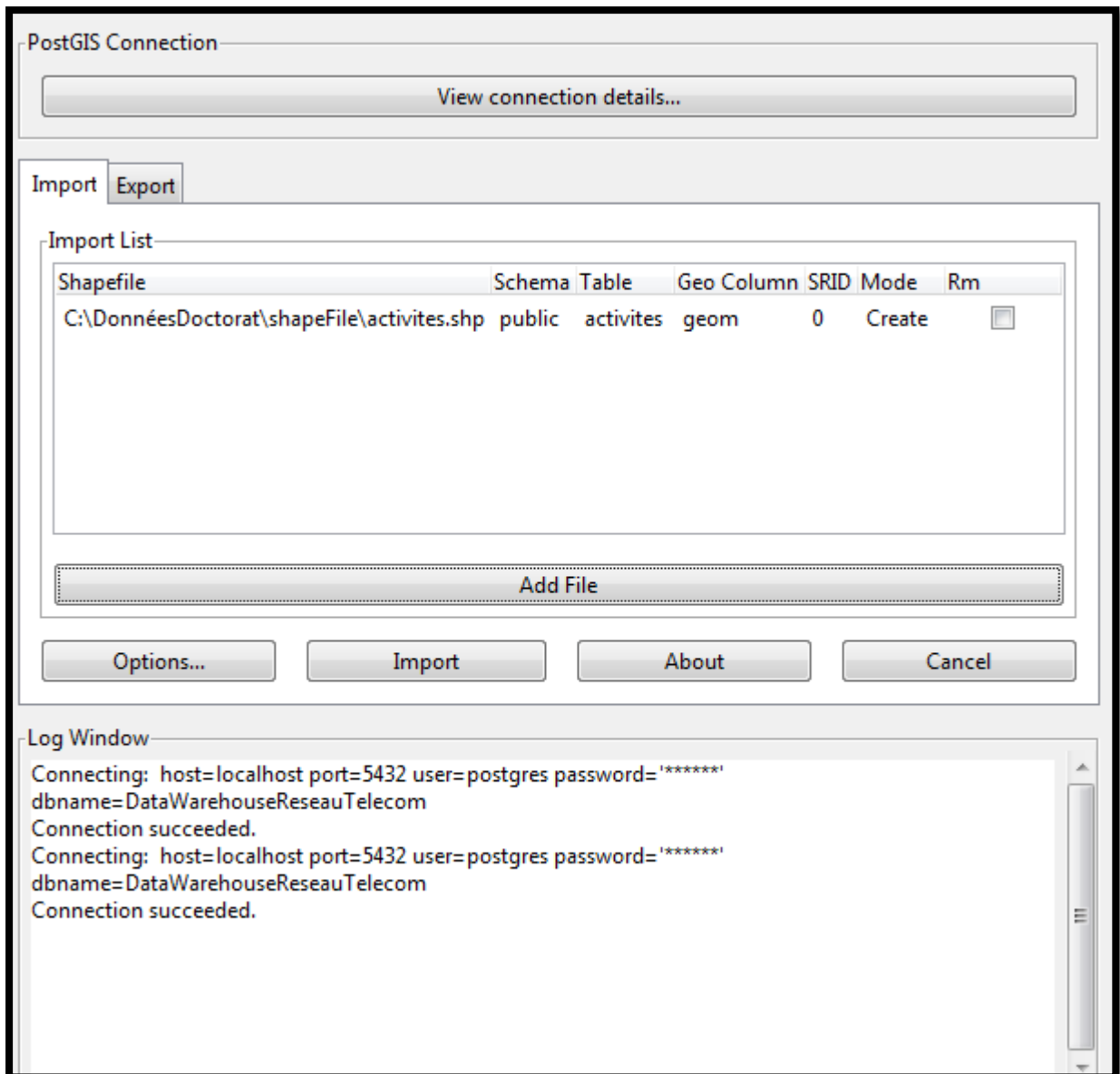


Figure 95 : Résultat de l'intégration

## Annexe 3 : Installation des Outils SQLAlchemy :

### Installation

There are two options how to install cubes: basic common installation - recommended mostly for users starting with Cubes. Then the reiscustomized installation with requirements explained.

### Basic Installation

The cubes has optional requirements:

- [SQLAlchemy](#) for SQL data base aggregation browsing backend (version  $\geq 0.7.4$ )
- [Flask](#) for Slicer OLAP HTTP server

Note

If you never used Python before, you might have to get the [pipinstaller](#) first, if you do not have it already.

Note

The command-line tool *Slicer* does not require know ledge of Python. You do not need to know the language if you just want to *serve* OLAP data.

You may install Cubes with the minimal dependencies,

Pipinstall cubes with certain extras (html, sql, mongo, mongo2, or slicer),  
Pipinstall cubes [slicer] or with all of the extras pipinstall cubes [all]

If you are developing cubes, you should install cubes [all].

### Quick Start or Hello World!

Download the sources from the [Cubes Githubrepository](#). Go to the examples/hello\_world folder:

```
git clone git://github.com/DataBrewery/cubes.git cd cubes cd
examples/hello_world
```

Prepare data and run the *OLAP server*:

```
Python prepare_data.py slicer serve slicer.ini
```

And try to do some queries:

```
curl "http://localhost:5000/cube/irbd_balance/aggregate" curl
« http://localhost:5000/cube/irbd_balance/aggregate?drilldown=year" curl
« http://localhost:5000/cube/irbd_balance/aggregate?drilldown=item" curl
« http://localhost:5000/cube/irbd_balance/aggregate?drilldown=item &
cut=item : e »
```

## Customized Installation

The project sources are stored in the [Github repository](#).

Download from Github:

```
git clone git://github.com/DataBrewery/cubes.git
```

Install :

```
cd cubes pipinstall — r requirements.txt pipinstall — r requirements-optional.txt
python setup.py install
```

Note

The requirements for [SQLAlchemy](#) and [Flask](#) are optional and you do not need them if you are going to use another kind of back end or don't going to use the Slicer server.

## Annexe 4 : Création du cubes de données Schéma

1. Click on **File > New > New Schema**.



Figure 96 : Création d'un Schéma Workbench

2. Choose **Foodmart** data source (see [Adding a Database Connection](#))



Figure 97 : Choix d'une source de données

3. On the left, click on **New Schema1**. This item's attributes appear on the right. Write **Foodmart** for **name** attribute.

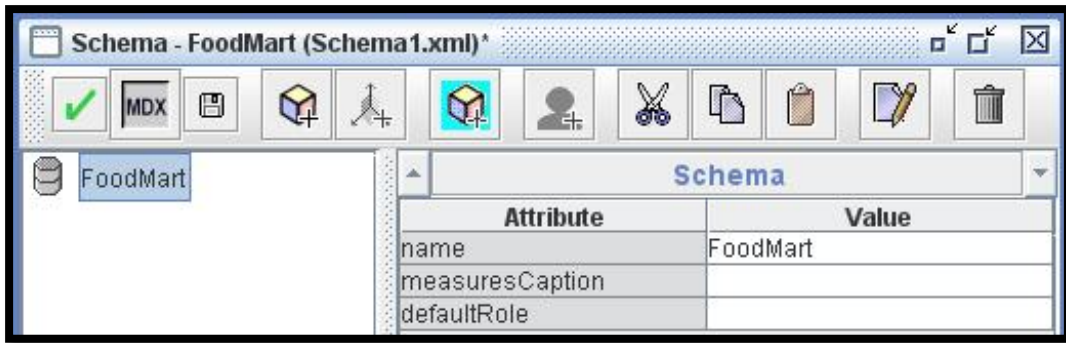


Figure 98 : Fenêtre de schéma

## Cube

- Right click on **Foodmart** schema and do **Add cube**. Click on **New Cube 1** and write **Products** for **name** attribute.

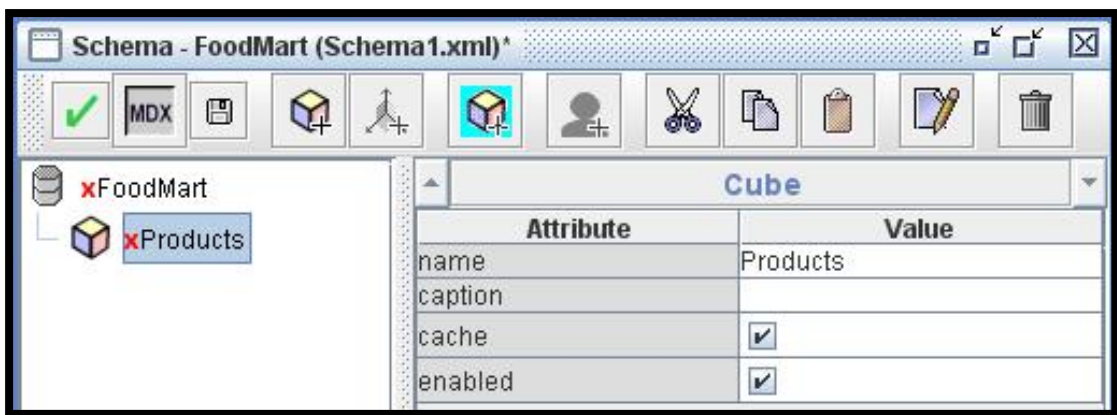


Figure 99 : Ajout du cube

- Right click on **Products** cube and do **Add Table**. Click on **Table** and choose **Product** table for **name** attribute.

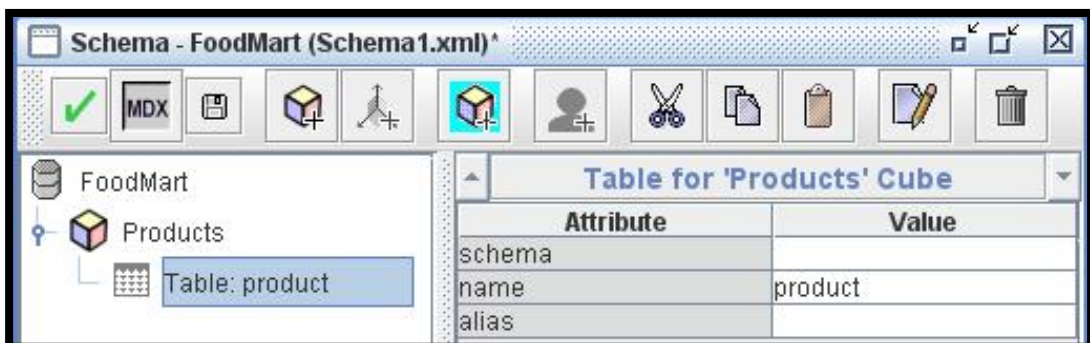


Figure 100 : Ajout de la table

## Dimension

- Right click on **Products** cube and do **Add Dimension**. Click on **New Dimension 1** and write **Product** for name attribute. Choose **product\_id** column for **foreign Key** attribute.

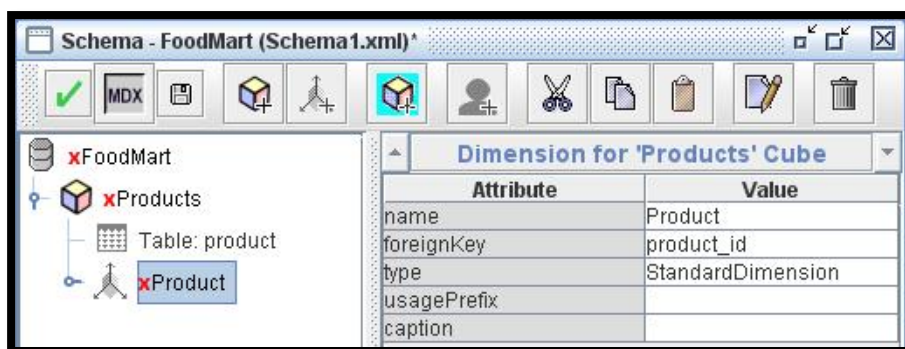


Figure 101 : ajout d'une dimension

- Right click on **default** hierarchy and do **Add table**. Click on **Table** and write **product** for name attribute.

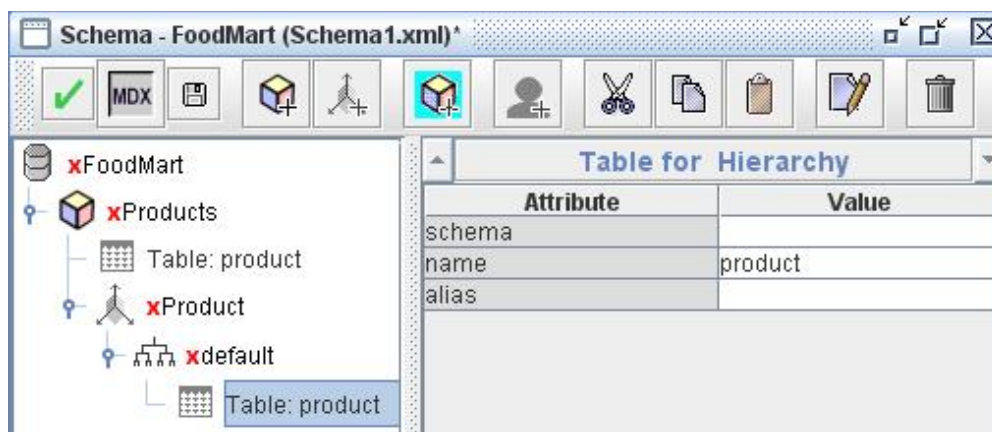


Figure 102 : Ajout d'une table

## Level

- Right click on **default** hierarchy and do **Add Level**. Click on **New Level 1** and fill out the following fields:

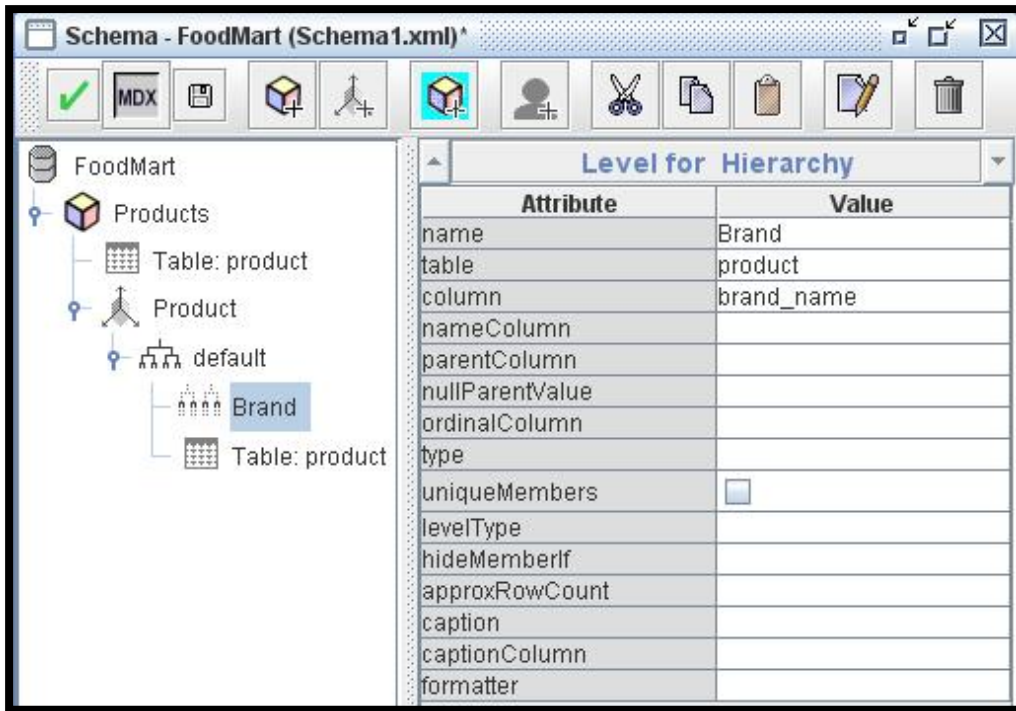


Figure 103 : Ajout d'un niveau

9. Repeat the previous step 8 to create two levels as below:

Name and SKULevelsAttributes		
<b>name</b>	Name	SKU
<b>table</b>	product	product
<b>column</b>	product_name	SKU

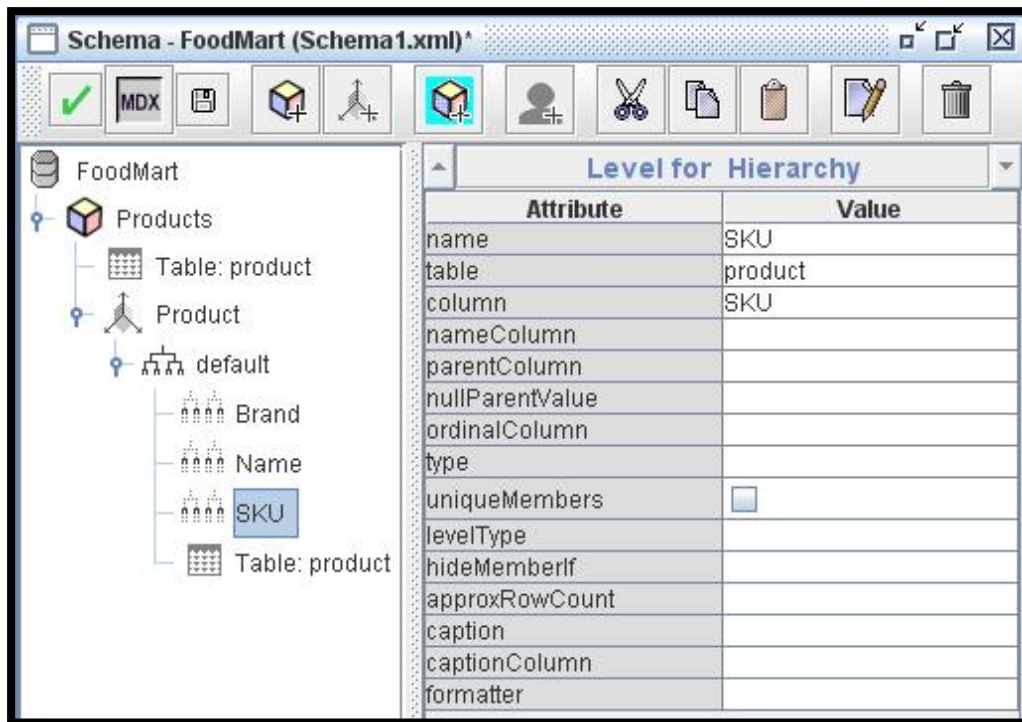
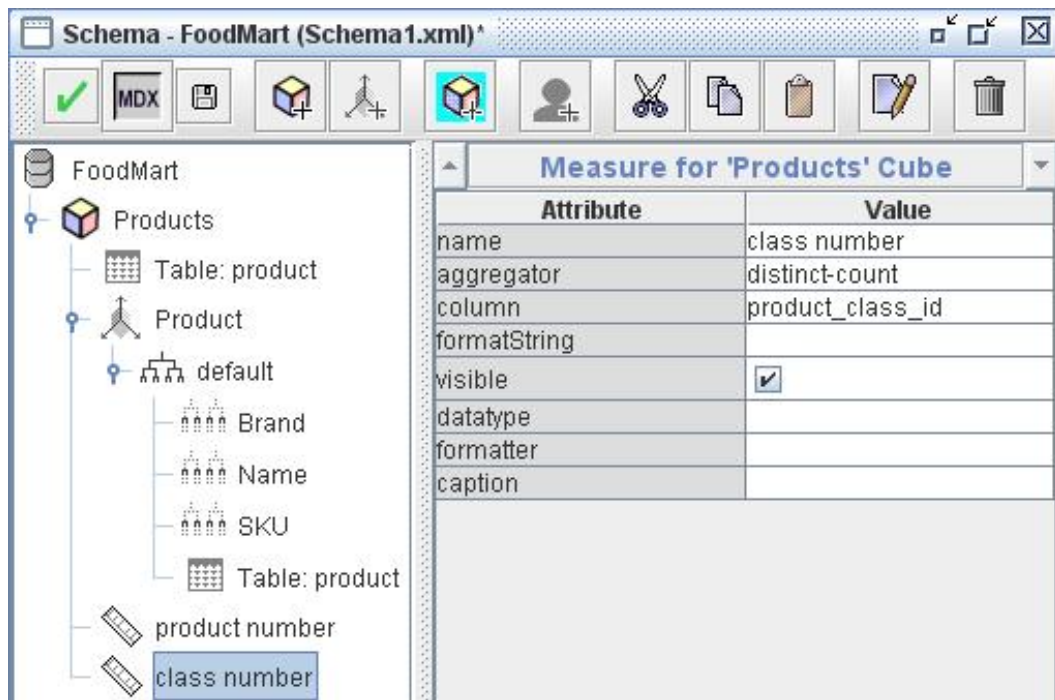


Figure 104 : Définition des nouveaux niveaux

## Measure

10. Right click on **Products** cube and do **Add Measure** twice. Click on **New Measure 0** and **New Measure 1** then fill out the following fields:

Product and Class NumbersAttributes		
<b>name</b>	productnumber	class number
<b>aggregator</b>	distinct-count	distinct-count
<b>column</b>	product_id	product_class_id



*Figure 105 : création des mesures*

## Save your Schema

11. Click on **File > Save As ...** . Choose a path and a name for your schema.

# Annexe 5 : Compilation du code source QGIS

## Comment compiler le code sources :

### QGIS

Building QGIS from source - step by step

Wednesday June 24, 2015

Last Updated: Wednesday June 24, 2015

Last Change : Wednesday June 24, 2015

1. Introduction
2. Overview
3. Building on GNU/Linux
  - 3.1. Building QGIS with Qt 4.x
  - 3.2. Prepare apt
  - 3.3. Install build dependencies
  - 3.4. Setup ccache (Optional)
  - 3.5. Prepare your development environment
  - 3.6. Check out the QGIS Source Code
  - 3.7. Starting the compile
  - 3.8. Building Debian packages
4. Building on Windows
  - 4.1. Building with Microsoft Visual Studio
  - 4.2. Building using MinGW
  - 4.3. Creation of MSYS environment for compilation of QGIS
5. Building on MacOS X
  - 5.1. Install Developer Tools
- 6.1. Préparation
- 6.2. Setup mapserver
- 6.3. Create a home page
- 6.4. Now deploy it
- 6.5. Debugging
7. Setting up a Jenkins Build Server
8. Debug output and running tests
9. Authors and Acknowledgments
- 5.2. Install Qt4 from disk image
- 5.3. Install CMake for OSX
- 5.4. Install development frameworks for QGIS dependencies
- 5.5. API documentation
- 5.6. QGIS source
- 5.7. Configure the build
- 5.8. Building
- 5.9. Post-Install

### 1. Introduction

=====

This document is the original installation guide of the described software

QGIS. The software and hardware descriptions named in this document are in most cases registered trademarks and are therefore subject to the legal requirements. QGIS is subject to the GNU General Public Licence. Find more information on the QGIS Homepage:

<http://qgis.org>

The details, that are given in this document have been written and verified to the best of knowledge and responsibility of the editors. Nevertheless, mistakes concerning the content are possible. Therefore, all data are not liable to any duties or guarantees. The editors and publishers do not take any responsibility or liability for failures and their consequences. You are always welcome for indicating possible mistakes.

You can download this document as part of the QGIS « User and Installation Guide » in HTML and PDF format via <http://qgis.org>. A current version is also available at:

<http://htmlpreview.github.io/?https://raw.githubusercontent.com/qgis/QGIS/master/doc/INSTALL.html>

Translations of this document can also be downloaded at the documentation area of the QGIS project at <http://qgis.org>. More information is available

via <http://qgis.org/en/site/getinvolved/governance/organisation/governance.html#community-resources>.

Please visit <http://qgis.org> for information on joining our mailing lists and getting involved in the project further.

source directory.

## 2. Overview

=====

QGIS, like a number of major projects (eg. KDE 4.0), uses CMake (<http://www.cmake.org>) for building from source.

Following a summary of the required dependencies for building:

Required build tools:

— CMake  $\geq 2.8.6$

— Flex  $\geq 2.5.6$

— Bison  $\geq 2.4$

Required build dependencies:

— Qt  $\geq 4.8.0$

— Proj  $\geq 4.4.x$

— GEOS  $\geq 3.0$

— Sqlite3  $\geq 3.0.0$

— GDAL/OGR  $\geq 1.4.x$

— Qwt  $\geq 5.0$  & ( $< 6.1$  with internal QwtPolar)

— expat  $\geq 1.95$

— QScintilla2

Optional dependencies:

- for GRASS providers and plugin - GRASS  $\geq 6.0.0$ . QGIS may be compiled with GRASS 6 or GRASS 7.

It can also be compiled with both GRASS versions in a single build but only if QGIS

is not installed with rpath. The desired GRASS version is chosen on runtime by setting

LD\_LIBRARY\_PATH or PATH.

- for georeferencer - GSL >= 1.8
- for postgis support and SPIT plugin - PostgreSQL >= 8.0.x
- for gps plugin — gpsbabel
- for mapserver export and PyQGIS - Python >= 2.3 (2.5+ preferred)
- for python support — SIP >= 4.12, PyQt >= 4.8.3 must match Qt version, Qscintilla2
- for qgis mapserver — FastCGI
- for oracle provider - Oracle OCI library

Indirect dependencies:

Some proprietary formats (eg. ECW and MrSid) supported by GDAL require proprietary third party libraries. QGIS doesn't need any of those itself to build, but will only support those formats if GDAL is built accordingly. Refer to [http://gdal.org/formats\\_list.html](http://gdal.org/formats_list.html) for instructions how to include those formats in GDAL.

### 3. Building on GNU/Linux

#### 3.1. Building QGIS with Qt 4.x

Requires: Ubuntu / Debian derived distro

!\ Note: Refer to the section Building Debian packages for building debian packages. Unless you plan to develop on QGIS, that is probably the easiest option to compile and install QGIS.

These notes are for Ubuntu - other versions and Debian derived distros may require slight variations in package names.

These notes are for if you want to build QGIS from source. One of the major aims here is to show how this can be done using binary packages for \*all\* dependencies - building only the core QGIS stuff from source. I prefer this approach because it means we can leave the business of managing system packages

to apt and only concern ourselves with coding QGIS!

This document assumes you have made a fresh install and have a 'clean' system.

These instructions should work fine if this is a system that has already been in use for a while, you may need to just skip those steps which are irrelevant to you.

#### 3.2. Prepare apt

The packages QGIS depends on to build are available in the "universe" component

of Ubuntu. This is not activated by default, so you need to activate it:

1. Edit your /etc/apt/sources.list file.
2. Uncomment all the lines starting with "deb"

Also you will need to be running Ubuntu 'precise' or higher in order for all dependencies to be met.

Now update your local sources database:

```
sudo apt-get update
```

#### 3.3. Install build dependencies

```
|| Distribution | install command for packages |
| wheezy | `apt-get install bison cmake doxygen flex git graphviz grass-dev
libexpat1-dev libfcgi-dev libgdal1-dev libgeos-dev libgsl0-dev
```

```

libopencscenograph-dev libosgearth-dev libpq-dev libproj-dev libqscintilla2-dev
libqt4-dev libqt4-opengl-dev libqt4-sql-sqlite libqtwebkit-dev libqwt-dev
libspatialindex-dev libspatialite-dev libsqlite3-dev lighttpd locales pkg-config
poppler-utils pyqt4-dev-tools python python-dev python-qscintilla2 python-qt4
python-qt4-dev python-sip python-sip-dev qt4-doc-html spawn-fcgi txt2tags xauth
xfonts-100dpi xfonts-75dpi xfonts-base xfonts-scalable xvfb cmake-curses-gui` |
| jessie | `apt-get install bison cmake doxygen flex git graphviz grass-dev
libexpat1-dev libfcgi-dev libgdal-dev libgeos-dev libgs10-dev libopencscenograph-
dev libosgearth-dev libpq-dev libproj-dev libqscintilla2-dev libqt4-dev libqt4-
opengl-dev libqt4-sql-sqlite libqtwebkit-dev libqwt-dev libspatialindex-dev
libspatialite-dev libsqlite3-dev lighttpd locales pkg-config poppler-utils pyqt4-
dev-tools pyqt4.qsci-dev python-all python-all-dev python-pyspatialite python-
qscintilla2 python-qt4 python-qt4-dev python-sip python-sip-dev qt4-doc-html
spawn-fcgi txt2tags xauth xfonts-100dpi xfonts-75dpi xfonts-base xfonts-scalable
xvfb cmake-curses-gui` |
| stretch | `apt-get install bison cmake doxygen flex git graphviz grass-dev
libexpat1-dev libfcgi-dev libgdal-dev libgeos-dev libgs10-dev libopencscenograph-
dev libosgearth-dev libpq-dev libproj-dev libqscintilla2-dev libqt4-dev libqt4-
opengl-dev libqt4-sql-sqlite libqtwebkit-dev libqwt-dev libspatialindex-dev
libspatialite-dev libsqlite3-dev lighttpd locales pkg-config poppler-utils pyqt4-
dev-tools pyqt4.qsci-dev python-all python-all-dev python-pyspatialite python-
qscintilla2 python-qt4 python-qt4-dev python-sip python-sip-dev qt4-doc-html
spawn-fcgi txt2tags xauth xfonts-100dpi xfonts-75dpi xfonts-base xfonts-scalable
xvfb cmake-curses-gui` |

```

```

| precise | `apt-get install bison cmake doxygen flex git graphviz grass-dev
libexpat1-dev libfcgi-dev libgdal-dev libgeos-dev libgs10-dev libopencscenograph-
dev libosgearth-dev libpq-dev libproj-dev libqscintilla2-dev libqt4-dev libqt4-
opengl-dev libqt4-sql-sqlite libqtwebkit-dev libqwt5-qt4-dev libspatialindex-dev
libspatialite-dev libsqlite3-dev lighttpd locales pkg-config poppler-utils pyqt4-
dev-tools python python-qscintilla2 python-qt4 python-qt4-dev python-sip
python-sip-dev qt4-doc-html spawn-fcgi txt2tags xauth xfonts-100dpi xfonts-
75dpi xfonts-base xfonts-scalable xvfb cmake-curses-gui` |
| trusty | `apt-get install bison cmake doxygen flex git graphviz grass-dev
libexpat1-dev libfcgi-dev libgdal-dev libgeos-dev libgs10-dev libopencscenograph-
dev libosgearth-dev libpq-dev libproj-dev libqscintilla2-dev libqt4-dev libqt4-
opengl-dev libqt4-sql-sqlite libqtwebkit-dev libqwt5-qt4-dev libspatialindex-dev
libspatialite-dev libsqlite3-dev lighttpd locales pkg-config poppler-utils pyqt4-
dev-tools python-all python-all-dev python-pyspatialite python-qscintilla2 python-
qt4 python-qt4-dev python-sip python-sip-dev qt4-doc-html spawn-fcgi txt2tags
xauth xfonts-100dpi xfonts-75dpi xfonts-base xfonts-scalable xvfb cmake-curses-
gui` |
| utopic | `apt-get install bison cmake doxygen flex git graphviz grass-dev
libexpat1-dev libfcgi-dev libgdal-dev libgeos-dev libgs10-dev libopencscenograph-
dev libosgearth-dev libpq-dev libproj-dev libqscintilla2-dev libqt4-dev libqt4-
opengl-dev libqt4-sql-sqlite libqtwebkit-dev libqwt5-qt4-dev libspatialindex-dev
libspatialite-dev libsqlite3-dev lighttpd locales pkg-config poppler-utils pyqt4-
dev-tools python-all python-all-dev python-pyspatialite python-qscintilla2 python-
qt4 python-qt4-dev python-sip python-sip-dev qt4-doc-html spawn-fcgi txt2tags

```

```
xauth xfonts-100dpi xfonts-75dpi xfonts-base xfonts-scalable xvfb cmake-curses-
gui` |
| vivid | `apt-get install bison cmake doxygen flex git graphviz grass-dev
libexpat1-dev libfcgi-dev libgdal-dev libgeos-dev libgsl0-dev libopencscenograph-
dev libosgearth-dev libpq-dev libproj-dev libqscintilla2-dev libqt4-dev libqt4-
opengl-dev libqt4-sql-sqlite libqtwebkit-dev libqwt5-qt4-dev libspatialindex-dev
libspatialite-dev libsqlite3-dev lighttpd locales pkg-config poppler-utils pyqt4-
dev-tools python-all python-all-dev python-pyspatialite python-qscintilla2 python-
qt4 python-qt4-dev python-sip python-sip-dev qt4-doc-html spawn-fcgi txt2tags
xauth xfonts-100dpi xfonts-75dpi xfonts-base xfonts-scalable xvfb cmake-curses-
gui` |
```

```
| sid | `apt-get install bison cmake doxygen flex git graphviz grass-dev libexpat1-
dev libfcgi-dev libgdal-dev libgeos-dev libgsl0-dev libopencscenograph-dev
libosgearth-dev libpq-dev libproj-dev libqscintilla2-dev libqt4-dev libqt4-opengl-
dev libqt4-sql-sqlite libqtwebkit-dev libqwt-dev libspatialindex-dev libspatialite-
dev libsqlite3-dev lighttpd locales pkg-config poppler-utils pyqt4-dev-tools
pyqt4.qsci-dev python-all python-all-dev python-pyspatialite python-qscintilla2
python-qt4 python-qt4-dev python-sip python-sip-dev qt4-doc-html spawn-fcgi
txt2tags xauth xfonts-100dpi xfonts-75dpi xfonts-base xfonts-scalable xvfb
cmake-curses-gui` |
```

(extracted from the control.in file in debian/)

### 3.4. Setup ccache (Optional)

=====

You should also setup ccache to speed up compile times:

```
cd /usr/local/bin
sudo ln -s /usr/bin/ccache gcc
sudo ln -s /usr/bin/ccache g++
```

### 3.5. Prepare your development environment

=====

As a convention I do all my development work in \$HOME/dev/<language>, so in this case we will create a work environment for C++ development work like this:

```
mkdir -p $ {HOME}/dev/cpp
cd $ {HOME}/dev/cpp
```

This directory path will be assumed for all instructions that follow.

### 3.6. Check out the QGIS Source Code

=====

There are two ways the source can be checked out. Use the anonymous method if you do not have edit privileges for the QGIS source repository, or use the developer checkout if you have permissions to commit source code changes.

1. Anonymous Checkout

```
cd $ {HOME}/dev/cpp
git clone git://github.com/qgis/QGIS.git
```
2. Developer Checkout

```
cd $ {HOME}/dev/cpp
git clone git@github.com:qgis/QGIS.git
```

### 3.7. Starting the compile

I compile my development version of QGIS into my ~/apps directory to avoid conflicts with Ubuntu packages that may be under /usr. This way for example you can use the binary packages of QGIS on your system along side with your development version. I suggest you do something similar:

```
mkdir -p $ {HOME}/apps
```

Now we create a build directory and run cmake:

```
cd QGIS
mkdir build-master
cd build-master
cmake ..
```

When you run cmake (note the .. is required!) , a menu will appear where you can configure various aspects of the build. If you want QGIS to have debugging capabilities then set CMAKE\_BUILD\_TYPE to Debug. If you do not have

root access or do not want to overwrite existing QGIS installs (by your packagemanager for example), set the CMAKE\_INSTALL\_PREFIX to somewhere you

have write access to (I usually use \${HOME}/apps). Now press 'c' to configure, 'e' to dismiss any error messages that may appear. and 'g' to generate the make files. Note that sometimes 'c' needs to be pressed several times before the 'g' option becomes available. After the 'g' generation is complete, press 'q' to exit the cmake interactive dialog.

Now on with the build:

```
make
make install
```

It may take a little while to build depending on your platform.

After that you can try to run QGIS:

```
~/apps/bin/qgis
```

If all has worked properly the QGIS application should start up and appear on your screen. If you get the error message "error while loading shared libraries", execute this command in your shell.

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH : $ {HOME}/apps/lib/
```

### 3.8. Building Debian packages

---

Instead of creating a personal installation as in the previous step you can also create debian package. This is done from the QGIS root directory, where you'll find a debian directory.

First you need to install the debian packaging tools once:

```
apt-get install build-essential
```

First you need to create a changelog entry for your distribution. For example for Ubuntu Lucid:

```
dch -l ~precise --force-distribution --distribution precise "precise build"
```

The QGIS packages will be created with:

```
dpkg-buildpackage -us -uc -b
```

#!/ Note: Install devscripts to get dch.

#!/ Note: If dpkg-buildpackage complains about unmet build dependencies you can install them using apt-get and re-run the command.

#!/ Note: If you have libqgis1-dev installed, you need to remove it first

using `dpkg --r libqgis1-dev`. Otherwise `dpkg-buildpackage` will complain about a build conflict.

/!\ Note: By default tests are run in the process of building and their results are uploaded to <http://dash.orfeo-toolbox.org/index.php?project=QGIS>. You can turn the tests off using `DEB_BUILD_OPTIONS=nocheck` in front of the build command. The upload of results can be avoided with `DEB_TEST_TARGET=test`.

The packages are created in the parent directory (ie. one level up).

Install them using `dpkg`. E.g.:

```
sudo debi
```

#### 4. Building on Windows

---

##### 4.1. Building with Microsoft Visual Studio

---

This section describes how to build QGIS using Visual Studio on Windows. This is currently also how the binary QGIS packages are made (earlier versions used MinGW).

This section describes the setup required to allow Visual Studio to be used to build QGIS.

###### 4.1.1. Visual C++ Express Edition

---

The free (as in free beer) Express Edition installer is available under:

[http://download.microsoft.com/download/c/d/7/cd7d4dfb-5290-4cc7-9f85-ab9e3c9af796/vc\\_web.exe](http://download.microsoft.com/download/c/d/7/cd7d4dfb-5290-4cc7-9f85-ab9e3c9af796/vc_web.exe)

You also need the Windows SDK for Windows 7 and .NET Framework 4:

[http://download.microsoft.com/download/A/6/A/A6AC035D-DA3F-4F0C-ADA4-37C8E5D34E3D/winsdk\\_web.exe](http://download.microsoft.com/download/A/6/A/A6AC035D-DA3F-4F0C-ADA4-37C8E5D34E3D/winsdk_web.exe)

###### 4.1.2. Other tools and dependencies

---

Download and install following packages:

|| Tool | Website |

| CMake | <http://www.cmake.org/files/v3.0/cmake-3.0.2-win32-x86.exe> |

| GNU flex, GNU bison and GIT | <http://cygwin.com/setup-x86.exe> (32bit) or [http://cygwin.com/setup-x86\\_64.exe](http://cygwin.com/setup-x86_64.exe) (64bit) |

| OSGeo4W | <http://download.osgeo.org/osgeo4w/osgeo4w-setup-x86.exe> (32bit) or [http://download.osgeo.org/osgeo4w/osgeo4w-setup-x86\\_64.exe](http://download.osgeo.org/osgeo4w/osgeo4w-setup-x86_64.exe) (64bit) |

OSGeo4W does not only provide ready packages for the current QGIS release and nightly builds of master, but also offers most of the dependencies needs to build it.

For the QGIS build you need to install following packages from cygwin:

— bison

— flex

— git

and from OSGeo4W (select Advanced Installation):

— expat

— fcgi

— gdal

— grass

- gsl-devel
- iconv
- pyqt4
- qt4-devel
- qwt5-devel-qt4
- sip
- spatialite
- libspatialindex-devel
- python-qscintilla

This will also select packages the above packages depend on.

Earlier versions of this document also covered how to build all above dependencies. If you're interested in that, check the history of this page in the Wiki

or the SVN repository.

#### 4.1.3. Setting up the Visual Studio project with CMake

---

/!\ Consider this section as example. It tends to outdate, when OSGeo4W and SDKs move on. ms-windows/osgeo4w/package-nightly.cmd is used for the nightly builds and constantly updated and hence might contain necessary updates that are not yet reflected here.

To start a command prompt with an environment that both has the VC++ and the OSGeo4W

variables create the following batch file (assuming the above packages were installed in the default locations):

```
@echo off
set VS90COMNTOOLS=%PROGRAMFILES%\Microsoft Visual Studio 9.0\Common7\Tools\
call « % PROGRAMFILES%\Microsoft Visual Studio 9.0\VC\vcvarsall.bat » x86
set INCLUDE=% INCLUDE% ; % PROGRAMFILES%\Microsoft SDKs\Windows\v7.1\include
set LIB=% LIB% ; % PROGRAMFILES%\Microsoft SDKs\Windows\v7.1\lib
set OSGEO4W_ROOT=C : \OSGeo4W
call « % OSGEO4W_ROOT%\bin\o4w_env.bat »
path % PATH % ; %PROGRAMFILES%\CMake\bin;c:\cygwin\bin
@set GRASS_PREFIX=c:/OSGeo4W/apps/grass/grass-6.4.4
@set INCLUDE=% INCLUDE% ; % OSGEO4W_ROOT%\include
@set LIB=% LIB% ; % OSGEO4W_ROOT%\lib ; % OSGEO4W_ROOT%\lib
@cmd
```

Start the batch file and on the command prompt checkout the QGIS source from git to the source directory QGIS:

```
git clone git://github.com/qgis/QGIS.git
```

Create a 'build' directory somewhere. This will be where all the build output will be generated.

Now run cmake-gui (still from cmd) and in the Where is the source code: box, browse to the top level QGIS directory.

In the Where to build the binaries: box, browse to the 'build' directory you created.

If the path to bison and flex contains blanks, you need to use the short name for the directory (c.-à-d. C : \Program Files should be rewritten to

C : \Progra~n, where n is the number as shown in `dir /x C : \`).

Verify that the 'BINDINGS\_GLOBAL\_INSTALL' option is not checked, so that python bindings are placed into the output directory when you run the INSTALL target. Hit Configure to start the configuration and select Visual Studio 9 2008 and keep native compilers and click Finish. The configuration should complete without any further questions and allow you to click Generate. Now close cmake-gui and continue on the command prompt by starting vcexpress. Use File / Open / Project/Solutions and open the qgis-x.y.z.sln File in your project directory. Change Solution Configuration from Debug to RelWithDebInfo (Release with Debug Info) or Release before you build QGIS using the ALL\_BUILD target (otherwise you need debug libraries that are not included). After the build completed you should install QGIS using the INSTALL target. Install QGIS by building the INSTALL project. By default this will install to c:\Program Files\qgis<version> (this can be changed by changing the CMAKE\_INSTALL\_PREFIX variable in cmake-gui). You will also either need to add all the dependency DLLs to the QGIS install directory or add their respective directories to your PATH.

#### 4.1.4. Packaging

---

To create a standalone installer there is a perl script named 'creatensis.pl' in « qgis/ms-windows/osgeo4w ». It downloads all required packages from OSGeo4W and repackages them into an installer using NSIS. The script can be run on both Windows and Linux. On Debian/Ubuntu you can just install the 'nsis' package. NSIS for Windows can be downloaded at: <http://nsis.sourceforge.net> And Perl for Windows (including other requirements like 'wget', 'unzip', 'tar' and 'bzip2') is available at: <http://cygwin.com>

#### 4.1.5. Packaging your own build of QGIS

---

Assuming you have completed the above packaging step, if you want to include your own hand built QGIS executables, you need to copy them in from your windows installation into the ms-windows file tree created by the creatensis script.

```
cd ms-windows/  
rm — rf osgeo4w/unpacked/apps/qgis/*  
cp — r /tmp/qgis1.7.0/* osgeo4w/unpacked/apps/qgis/
```

Now create a package.

```
./quickpackage.sh
```

After this you should now have a nsis installer containing your own build of QGIS and all dependencies needed to run it on a windows machine.

#### 4.1.6. Osgeo4w packaging

---

The actual packaging process is currently not documented, for now please take a

look at:

[ms-windows/osgeo4w/package.cmd](#)

## 4.2. Building using MinGW

---

Note: This section might be outdated as nowadays Visual C++ is used to build the « official » packages.

Note: For a detailed account of building all the dependencies yourself you can visit Marco Pasetti's website here:

<http://www.webalice.it/marco.pasetti/qgis+grass/BuildFromSource.html>

Read on to use the simplified approach with pre-built libraries...

### 4.2.1. MSYS

---

MSYS provides a unix style build environment under windows. We have created a

zip archive that contains just about all dependencies.

Get this:

<http://download.osgeo.org/qgis/win32/msys.zip>

and unpack to `c : \msys`

If you wish to prepare your msys environment yourself rather than using our pre-made one, detailed instructions are provided elsewhere in this document.

### 4.2.2. Qt

---

Download Qt opensource precompiled edition exe and install (including the download and install of mingw) from here:

<http://qt.nokia.com/downloads/>

When the installer will ask for MinGW, you don't need to download and install it, just point the installer to `c:\msys\mingw`

When Qt installation is complete:

Edit `C:\Qt\4.8.0\bin\qtvars.bat` and add the following lines:

```
set PATH=% PATH % ; C:\msys\local\bin ; c:\msys\local\lib
```

```
set PATH=% PATH % ; »C:\Program Files\Subversion\bin »
```

I suggest you also add `C:\Qt\4.8.0\bin\` to your Environment Variables Path in the windows system preferences.

If you plan to do some debugging, you'll need to compile debug version of Qt:

```
C:\Qt\4.8.0\bin\qtvars.bat compile_debug
```

Note: there is a problem when compiling debug version of Qt 4.7, the script ends with

this message « `mingw32-make: *** No rule to make target `debug`. Stop.` ». To compile the debug version you have to go out of `src` directory and execute the following command:

```
c:\Qt\4.8.0 make
```

### 4.2.3. Flex and Bison

---

Get Flex

[http://sourceforge.net/project/showfiles.php?group\\_id=23617&package\\_id=16424](http://sourceforge.net/project/showfiles.php?group_id=23617&package_id=16424)  
(the zip bin) and extract it into `c:\msys\mingw\bin`

### 4.2.4. Python stuff (optional)

---

Follow this section in case you would like to use Python bindings for QGIS. To

be able to compile bindings, you need to compile SIP and PyQt4 from sources as their installer doesn't include some development files which are necessary.

#### 4.2.4.1. Download and install Python — use Windows installer

---

==

(It doesn't matter to what folder you'll install it)

<http://python.org/download/>

#### 4.2.4.2. Download SIP and PyQt4 sources

---

<http://www.riverbankcomputing.com/software/sip/download>

<http://www.riverbankcomputing.com/software/pyqt/download>

Extract each of the above zip files in a temporary directory. Make sure to get versions that match your current Qt installed version.

#### 4.2.4.3. Compile SIP

---

```
c:\Qt\4.8.0\bin\qtvars.bat
python configure.py — p win 32 — g++
make
make install
```

#### 4.2.4.4. Compile PyQt

---

```
c:\Qt\4.8.0\bin\qtvars.bat
python configure.py
make
make install
```

#### 4.2.4.5. Final python notes

---

/!\ You can delete the directories with unpacked SIP and PyQt4 sources after a successful install, they're not needed anymore.

CMake is build system used by QGIS. Download it from here:

<http://www.cmake.org/files/v2.8/cmake-2.8.2-win32-x86.exe>

#### 4.2.7. QGIS

---

Start a cmd.exe window ( Start -> Run -> cmd.exe ) Create development directory and move into it

```
md c:\dev\cpp
```

```
cd c:\dev\cpp
```

Check out sources from GIT:

```
git clone git://github.com/qgis/QGIS.git
```

#### 4.2.8. Compiling

---

As a background read the generic building with CMake notes at the end of this document.

Start a cmd.exe window ( Start -> Run -> cmd.exe ) if you don't have one already. Add paths to compiler and our MSYS environment:

```
c:\Qt\4.8.0\bin\qtvars.bat
```

For ease of use add c:\Qt\4.8.0\bin\ to your system path in system properties so you can just type qtvars.bat when you open the cmd console.

Create build directory and set it as current directory:

```
cd c:\dev\cpp\qgis
```

```
md build
```

```
cd build
```

#### 4.2.9. Configuration

---

```
cmakesetup ..
```

Note: You must include the « .. » above.

Click « Configure » button. When asked, you should choose 'MinGW Makefiles' as generator.

There's a problem with MinGW Makefiles on Win2K. If you're compiling on this platform, use 'MSYS Makefiles' generator instead.

All dependencies should be picked up automatically, if you have set up the Paths correctly. The only thing you need to change is the installation destination (CMAKE\_INSTALL\_PREFIX) and/or set 'Debug'.

For compatibility with NSIS packaging scripts I recommend to leave the install prefix to its default c:\program files\

When configuration is done, click 'OK' to exit the setup utility.

#### 4.2.10. Compilation and installation

```
make make install
```

4.2.11. Run qgis.exe from the directory where it's installed (CMAKE\_INSTALL\_PREFIX)

---

Make sure to copy all .dll:s needed to the same directory as the qgis.exe binary is installed to, if not already done so, otherwise QGIS will complain about missing libraries when started.

A possibility is to run qgis.exe when your path contains c:\msys\local\bin and c : \msys\local\lib directories, so the DLLs will be used from that place.

#### 4.2.12. Create the installation package: (optional)

---

Download and install NSIS from ([http://nsis.sourceforge.net/Main\\_Page](http://nsis.sourceforge.net/Main_Page))

Now using windows explorer, enter the win\_build directory in your QGIS source

on qgis.nsi and choose the option 'Compile NSIS Script'.

### 4.3. Creation of MSYS environment for compilation of QGIS

---

#### 4.3.1. Initial setup

---

##### 4.3.1.1. MSYS

---

This is the environment that supplies many utilities from UNIX world in Windows and is needed

by many dependencies to be able to compile.

Download from here:

<http://puzzle.dl.sourceforge.net/sourceforge/mingw/MSYS-1.0.11-2004.04.30-1.exe>

Install to c : \msys

All stuff we're going to compile is going to get to this directory (resp. its subdirs).

#### 4.3.1.2. MinGW

---

Download from here:

<http://puzzle.dl.sourceforge.net/sourceforge/mingw/MinGW-5.1.3.exe>

Install to c:\msys\mingw

It suffices to download and install only g++ and mingw-make components.

#### 4.3.1.3. Flex and Bison

---

Flex and Bison are tools for generation of parsers, they're needed for GRASS and also QGIS compilation.

Download the following packages:

<http://gnuwin32.sourceforge.net/downlinks/flex-bin-zip.php>

<http://gnuwin32.sourceforge.net/downlinks/bison-bin-zip.php>

<http://gnuwin32.sourceforge.net/downlinks/bison-dep-zip.php>

Unpack them all to c:\msys\local

#### 4.3.2. Installing dependencies

---

##### 4.3.2.1. Getting ready

---

Paul Kelly did a great job and prepared a package of precompiled libraries for GRASS.

The package currently includes:

— zlib-1.2.3

— libpng-1.2.16-noconfig

— xdr-4.0-mingw2

— freetype-2.3.4

— fftw-2.1.5

— PDCurses-3.1

— proj-4.5.0

— gdal-1.4.1

It's available for download here:

<http://www.stjohnspoint.co.uk/grass/wingrass-extralibs.tar.gz>

Moreover he also left the notes how to compile it (for those interested):

<http://www.stjohnspoint.co.uk/grass/README.extralibs>

Unpack the whole package to c:\msys\local

##### 4.3.2.2. GRASS

---

Grab sources from CVS or use a weekly snapshot, see:

<http://grass.itc.it/devel/cvs.php>

In MSYS console go to the directory where you've unpacked or checked out sources

(e.g. c:\msys\local\src\grass-6.3.cvs)

Run these commands:

```
export PATH="/usr/local/bin:/usr/local/lib : $ PATH »
```

```
./configure --prefix=/usr/local --bindir=/usr/local --with-  
includes=/usr/local/include --with-libs=/usr/local/lib --with-cxx --without-jpeg \  
--without-tiff --with-postgres=yes --with-postgres-includes=/local/pgsql/include --  
with-pgsql-libs=/local/pgsql/lib --with-opengl=windows --with-fftw \  

```

```
--with-freetype -- with-freetype-includes=/mingw/include/freetype2 --without-x --
without-tcltk --enable-x11=no -- enable-shared=yes \
--with-proj-share=/usr/local/share/proj
```

make

make install

It should get installed to c:\msys\local\grass-6.3.cvs

By the way, these pages might be useful:

— [http://grass.gdf-hannover.de/wiki/WinGRASS\\_Current\\_Status](http://grass.gdf-hannover.de/wiki/WinGRASS_Current_Status)

— <http://geni.ath.cx/grass.html>

#### 4.3.2.3. GEOS

=====

Download the sources:

<http://geos.refractor.net/geos-2.2.3.tar.bz2>

Unpack to e.g. c:\msys\local\src

To compile, I had to patch the sources : in file source/headers/timeval.h line 13.

Change it from:

```
#ifdef _WIN32
```

to:

```
#if defined (_WIN32) && defined(_MSC_VER)
```

Now, in MSYS console, go to the source directory and run:

```
./configure --prefix=/usr/local
```

make

make install

#### 4.3.2.4. SQLITE

=====

You can use precompiled DLL, no need to compile from source:

Download this archive:

[http://www.sqlite.org/sqlitedll-3\\_3\\_17.zip](http://www.sqlite.org/sqlitedll-3_3_17.zip)

and copy sqlite3.dll from it to c:\msys\local\lib

Then download this archive:

[http://www.sqlite.org/sqlite-source-3\\_3\\_17.zip](http://www.sqlite.org/sqlite-source-3_3_17.zip)

and copy sqlite3.h to c:\msys\local\include

#### 4.3.2.5. GSL

=====

Download sources:

<ftp://ftp.gnu.org/gnu/gsl/gsl-1.9.tar.gz>

Unpack to c:\msys\local\src

Run from MSYS console in the source directory:

```
./configure
```

```
make
```

```
make install
```

#### 4.3.2.6. EXPAT

=====

Download sources:

<http://dfn.dl.sourceforge.net/sourceforge/expat/expat-2.0.0.tar.gz>

Unpack to c:\msys\local\src

Run from MSYS console in the source directory:

```
./configure  
make  
make install
```

#### 4.3.2.7. POSTGRES

---

We're going to use precompiled binaries. Use the link below for download:  
<http://wwwmaster.postgresql.org/download/mirrors-ftp?file=%2Fbinary%2Fv8.2.4%2Fwin32%2Fpostgresql-8.2.4-1-binaries-no-installer.zip>

copy contents of pgsq directory from the archive to c:\msys\local

#### 4.3.3. Cleanup

---

We're done with preparation of MSYS environment. Now you can delete all stuff in c:\msys\local\src - it takes quite a lot of space and it's not necessary at all.

### 5. Building on MacOS X

---

In this approach I will try to avoid as much as possible building dependencies from source and rather use frameworks wherever possible.

"Universal", SDK and non-default arch builds require more complex options and some fiddling with the system. It is best to stick with a single, default, architecture build and follow these instructions for an initial build.

Included are notes for building on Mac OS X 10.5 (Leopard), 10.6 (Snow Leopard), 10.7 (Lion), 10.8 (Mt. Lion) and 10.9 (Mavericks) (These names will be used throughout the instructions.)

Make sure to read each section completely before typing the first command you see.

General note on Terminal usage: When I say "cd" to a folder in a Terminal, it means type « cd » (without the quotes, make sure to type a space after) and then type the path to said folder, then <return>. A simple way to do this without having to know and type the full path is, after type the "cd " part, drag the folder (use the icon in its window title bar, or drag a folder from within a window) from the Desktop to the Terminal, then tap <return>.

Parallel Compilation: On multiprocessor/multicore Macs, it's possible to speed up compilation, but it's not automatic. Whenever you type "make" (but NOT "make install"), instead type:

```
make -j [#cpus]
```

Replace [#cpus] with the number of cores and/or processors your Mac has. On recent

models with hyperthreading processors this can be double the physical count of processors and cores.

ie: Mac Pro "8 Core" model (2 quad core processors) = 8

ie: Macbook Pro i5 (hyperthreading) = 2 cores X 2 = 4

To find out how many CPUs you have available, run the following in Terminal:

```
/usr/sbin/sysctl -n hw.ncpu
```

which can be used in build shell scripts like:

```
make — j $ (/usr/sbin/sysctl -n hw.ncpu)
```

Note: if you get an error in parallel compilation, try removing the -j # flag, so it's just 'make', or using a smaller number. Sometimes make can hiccup on too

many threads.

## 5.1. Install Developer Tools

---

Developer tools are not a part of a standard OS X installation. Up through Snow Leopard, the Developer Tools, later called Xcode, were included with the system install disks, though it's best to download the latest version compatible with your system to get important updates fixing various issues. Starting with Lion, Xcode is available as a download and from the App Store. Downloading Xcode/Developer Tools for up through Snow Leopard requires a free developer account at [developer.apple.com](http://developer.apple.com). Up through Snow Leopard, get the latest Xcode that is supported for your system. For Lion and above, you can get Xcode from either a free developer account or for a minimal fee from the app store. When installing Xcode up through Snow Leopard, make sure to do a custom install and install the Unix Development or Command Line Tools option.

On Lion, if you have installed Xcode 4.0 - 4.2 and are upgrading to 4.3, it's a good idea to uninstall the old version first with:

```
sudo /Developer/Library/uninstall-devtools
```

On Lion and Mt. Lion, using Xcode 4.4+, the developer command line tools can be installed via the Xcode preferences.

Xcode 4.3+ also introduces the clang frontend to the LLVM compiler as default.

Note: In XCODE 4.5 installed from the app store, you need to install the command line tools from XCode -> Preferences -> Downloads and choose command line tools.

<http://clang.llvm.org/>

The supplied clang version 4 can compile QGIS, but presents many warnings compared to just using LLVM. You can specifically use LLVM by exporting paths to

the compilers in Terminal, or shell scripts, prior to building QGIS:

```
export CC=/usr/bin/llvm-gcc
```

```
export CXX=/usr/bin/llvm-g++
```

If you have trouble building some of the dependencies listed below with clang (e.g. OSG & osgEarth), try using only the LLVM compilers.

## 5.2. Install Qt4 from disk image

---

You need a minimum of Qt-4.4.0. I suggest getting the latest (Qt 4, not 5). There is no need

for the full Qt SDK, so save yourself some download time and get the frameworks only. This is available in the Libraries section of the Qt download page.

Snow Leopard+ note: If you are building on Snow Leopard+, you will need to decide between 32-bit support in the older Qt Carbon branch, or 64-bit support in the Qt Cocoa branch. Appropriate installers are available for both as of Qt-4.5.2, though they stopped making Carbon packages at Qt 4.7.4.

Qt 4.6+ is recommended for Cocoa.

Starting with Lion, Carbon may not work properly, if at all.

Starting with Qt 4.8, only 64bit Cocoa installers are available.

General note: Support for new system versions in any given Qt version may not be present and may cause a 'This version of Mac OS X

is unsupported' error when building QGIS. Try the next Qt version.

PPC note: The readymade Qt Cocoa installers don't include PPC support, you'd have to compile Qt yourself. But, there appear to be issues with Qt Cocoa on PPC Macs anyways. Qt Carbon is recommended on PPC Macs.

<http://qt-project.org/downloads>

If you want debug frameworks, Qt also provides a separate download with these.

These are in addition to the non-debug frameworks.

Earlier OS X systems may need an old Qt version - check the requirements of the current Qt version. To get old Qt downloads, there is an FTP link at the bottom of the download page. Files are in the qt/source (yes, even the binary packages). Once downloaded open the disk image and run the installer. Note you need admin privileges to install.

Leopard+ note: Qt includes a couple non-framework libraries in /usr/lib.

When using a system SDK these libraries will not be found. To fix this problem, add symlinks to /usr/local:

```
sudo ln -s /usr/lib/libQtUiTools.a /usr/local/lib/
```

```
sudo ln -s /usr/lib/libQtCLucene.dylib /usr/local/lib/
```

These should then be found automatically. Earlier systems

may need some help by adding '-L/usr/local/lib' to CMAKE\_SHARED\_LINKER\_FLAGS, CMAKE\_MODULE\_LINKER\_FLAGS and CMAKE\_EXE\_LINKER\_FLAGS in the cmake build.

### 5.3. Install CMake for OSX

=====  
Get the latest source release from here:

<http://www.cmake.org/cmake/resources/software.html>

Binary installers are available for OS X, but they are not recommended (2.4 versions install in /usr instead of /usr/local, and 2.6+ versions are a strange application). Instead, download the source.

NOTE: 2.8.5 is broken for detecting part of Qt. Fixed in 2.8.6.

Double-click the source tarball to unpack it, then cd to the source folder and:

```
./bootstrap --docdir=/share/doc/CMake --mandir=/share/man
```

```
make -j [#cpus]
```

```
sudo make install
```

#### 5.3.1. Optional setup: ccache

=====  
Xcode 4.4+ note: You will probably not need to install ccache if you are using the clang frontend to LLVM compiler, a setup that already provides fairly quick compile times.

Setup ccache to significantly speed up compile times after initial build.

(Switching git branches will again cause longer initial build times unless separate build directories are used for each branch.)

Get the latest source release from here:

<http://ccache.samba.org/>

Double-click the source tarball to unpack, then, in Terminal.app, cd to the source folder and:

```
./configure
```

```
make
```

sudo make install

After install, symbolically link compilers to /usr/local/bin/ccache.

(Note: this differs from instructions at <http://ccache.samba.org/manual.html>

Changing the /usr/bin:/usr/local/bin order in PATH is not recommended on OS X.

```
sudo mkdir /usr/local/bin/compilers && cd /usr/local/bin/compilers
```

```
sudo ln -s ../ccache gcc
```

```
sudo ln -s ../ccache g++
```

```
sudo ln -s ../ccache cc
```

```
sudo ln -s ../ccache c++
```

Add the following to the end of your ~/.bash\_profile (and optionally ~/.bashrc) to allow your login shell to discover the symbolically linked compilers before /usr/bin compilers and to easily toggle using ccache off, by commenting out the line and starting a new login session in Terminal.

```
export PATH=/usr/local/bin/compilers : $ PATH
```

If you have trouble building some of the dependencies listed below (e.g. OSG & osgEarth), try bypassing ccache.

#### 5.4. Install development frameworks for QGIS dependencies

---

Download William Kyngesburye's excellent GDAL Complete package that includes

PROJ, GEOS, GDAL, SQLite3, Spatialite, and image libraries, as frameworks.

There are also GSL and FreeType frameworks.

<http://www.kyngchaos.com/software/frameworks>

Once downloaded, open and install the frameworks.

William provides an additional installer package for Postgresql (for PostGIS support). QGIS just needs the libpq client library, so unless you want to setup the full Postgres + PostGIS server, all you need is the client-only package. It's available here:

<http://www.kyngchaos.com/software/postgres>

Also available is a GRASS application:

<http://www.kyngchaos.com/software/grass>

Old versions of these packages for older systems are available in the software archive section.

##### 5.4.1. Additional dependencies: General compatibility note

---

There are some additional dependencies that, at the time of writing, are not provided as frameworks or installers so we will need to build these from source.

If you are wanting to build QGIS as a 64-bit application, you will need to provide the appropriate build commands to produce 64-bit support in dependencies.

Likewise, for 32-bit support on Snow Leopard, you will need to override the default system architecture, which is 64-bit, according to instructions for individual dependency packages.

Stable release versions are preferred. Beta and other development versions may have problems and you are on your own with those.

##### 5.4.2. Additional dependencies: Expat

---

Snow Leopard+ note: Snow Leopard includes a usable expat, so this step is

not necessary on Snow Leopard or above.

Get the expat sources:

[http://sourceforge.net/project/showfiles.php?group\\_id=10127](http://sourceforge.net/project/showfiles.php?group_id=10127)

Double-click the source tarball to unpack, then, in Terminal.app, cd to the source folder and:

```
./configure  
make  
sudo make install
```

#### 5.4.3. Additional dependencies: Spatialindex

---

Get the libspatialindex sources:

<http://download.osgeo.org/libspatialindex/>

Double-click the source tarball to unpack, then, in Terminal.app, cd to the source folder and:

```
./configure --disable-dependency-tracking CFLAGS= — Os  
make  
sudo make install
```

#### 5.4.4. Additional dependencies: Python

---

Leopard+ note: Starting with Leopard a usable Python is included in the system. This is Python 2.5, 2.6 and 2.7, respectively for Leo, Snow and Lion+.

So there is no need to install Python on Leopard and newer.

You can still install Python from python.org if preferred.

If installing from python.org, make sure you install the latest Python 2.x from

<http://www.python.org/download/>

Python 3 is a major change, and may have compatibility issues, so try it at your own risk.

#### 5.4.5. Additional dependencies: SIP

---

Mt Lion note: SIP 4.15.7 appears to not work on Mt Lion. Install either a prior version to 4.14.6 or a later version 4.16.3+

Retrieve the python bindings toolkit SIP from

<http://www.riverbankcomputing.com/software/sip/download>

Double-click the source tarball to unpack it, then, in Terminal.app, cd to the source folder. Then for your chosen Python:

```
python.org Python  
python configure.py  
make  
sudo make install
```

Leopard system Python

SIP wants to install in the system path -- this is not a good idea.

More configuration is needed to install outside the system path:

```
python configure.py — n — d /Library/Python/2.5/site-packages — b
```

```
/usr/local/bin \  
— e /usr/local/include — v /usr/local/share/sip — s MacOSX10.5.sdk  
Snow Leopard system Python
```

Similar to Leopard, you should install outside the system Python path.

Also, you need to specify the architecture you want and make sure to run the versioned python binary (this one responds to the 'arch' command, 'python' does not). Substitute « 2.7' for python version and 10.7 for SDK version below for Lion.

If you are using 32-bit Qt (Qt Carbon):

```
python2.6 configure.py — n — d /Library/Python/2.6/site-packages — b  
/usr/local/bin \  
— e /usr/local/include — v /usr/local/share/sip -- arch=i386 — s  
MacOSX10.6.sdk
```

For 64-bit Qt (Qt Cocoa), use this configure line:

```
python2.6 configure.py — n — d /Library/Python/2.6/site-packages — b  
/usr/local/bin \  
— e /usr/local/include — v /usr/local/share/sip -- arch=x86_64 — s  
MacOSX10.6.sdk
```

Lion+ system Python

Similar to Snow Leopard, you should install outside the system Python path.

The SDK option should match the system you are compiling on:

for Lion:

```
python2.7 configure.py — d /Library/Python/2.7/site-packages — b /usr/local/bin  
\  
— e /usr/local/include — v /usr/local/share/sip -- arch=x86_64 — s  
MacOSX10.7.sdk
```

for Mt. Lion:

```
python2.7 configure.py — d /Library/Python/2.7/site-packages — b /usr/local/bin  
\  
— e /usr/local/include — v /usr/local/share/sip -- arch=x86_64 — s  
MacOSX10.8.sdk
```

for Mavericks:

```
python2.7 configure.py — d /Library/Python/2.7/site-packages — b /usr/local/bin  
\  
— e /usr/local/include — v /usr/local/share/sip -- arch=x86_64 — s  
MacOSX10.9.sdk
```

continue...

Then continue with compilation and installation:

```
make
```

```
sudo make install
```

5.4.6. Additional dependencies: QScintilla2

Retrieve the Qt version of the Scintilla-based text editor widget from

<http://www.riverbankcomputing.co.uk/software/qscintilla/download>

Double-click the tarball to unpack it. Then, cd to the QScintilla2.x.x source folder in a Terminal.

QScintilla2 wants to install in the system path -- with libraries going into /Library/Frameworks and headers into /usr/include/Qsci -- this is not a good

idea, and it also basically breaks the QtDesigner plugin. More configuration is needed to install outside the system path, in /usr/local/:

```
cd Qt4Qt5
```

Edit QScintilla-gpl-2.x.x/Qt4Qt5/qscintilla.pro in the following manner:

current line --> new line

```
target.path = $$[QT_INSTALL_LIBS] --> target.path = /usr/local/lib
```

```
header.path = $$[QT_INSTALL_HEADERS] --> header.path = /usr/local/include
```

Save the qscintilla.pro file and build the QScintilla2 C++ library:

```
qmake -spec macx-g++ qscintilla.pro
```

```
make -j [#cpus]
```

```
sudo make install
```

adjust the install\_name\_tool command for the version installed of QScintilla installed:

```
sudo install_name_tool — id /usr/local/lib/libqscintilla2.11.dylib \
```

```
/usr/local/lib/libqscintilla2.11.dylib
```

This installs QScintilla2's dylib in /usr/local/lib/ and the header files in /usr/local/include/Qsci/, both of which should be automatically found when building QGIS.

#### 5.4.6.1. Optional setup : QScintilla2 QtDesigner plug-in

=====

The plugin allows QScintilla2 widgets to be used within QtDesigner.

```
cd <QScintilla2 source directory>
```

```
cd designer-Qt4Qt5
```

```
qmake -spec macx-g++ designer.pro
```

```
make
```

```
sudo make install
```

Installs in /Developer/Applications/Qt/plugins/designer/

#### 5.4.7. Additional dependencies: PyQt

=====

Retrieve the python bindings toolkit for Qt from

<http://www.riverbankcomputing.com/software/pyqt/download>

Double-click the source tarball to unpack it, then, in Terminal.app,

cd to the source folder. Then for your chosen Python:

python.org Python

```
python configure.py — n /usr/local/Qt4.8/qsci
```

```
yes
```

Leopard system Python

PyQt wants to install in the system path -- this is not a good idea.

More configuration is needed to install outside the system path:

```
python configure.py — d /Library/Python/2.5/site-packages — b /usr/local/bin —
```

```
n /usr/local/Qt4.8/qsci — v /usr/local/share/sip/PyQt4
```

Snow Leopard system Python

Similar to Leopard, you should install outside the system Python path.

Also, you need to specify the architecture you want (requires at least PyQt 4.6),

and make sure to run the versioned python binary (this one responds to the 'arch' command, which is important for pyuic4, 'python' does not).

Substitute « 2.7' for python version and 10.7 for SDK version below for Lion.

#### 4.2.5. git

In order to check out QGIS sources from the repository, you need a git client.  
This installer should work fine:

<http://msysgit.googlecode.com/files/Git-1.7.4-preview20110204.exe>

#### 4.2.6. CMake

If you are using 32-bit Qt (Qt Carbon):

```
python2.6 configure.py — d /Library/Python/2.6/site-packages — b /usr/local/bin  
\  
— n /usr/local/Qt4.8/qsci — v /usr/local/share/sip/PyQt4 -- use-arch i386
```

For 64-bit Qt (Qt Cocoa), use this configure line:

```
python2.6 configure.py — d /Library/Python/2.6/site-packages — b /usr/local/bin  
\  
— n /usr/local/Qt4.8/qsci — v /usr/local/share/sip/PyQt4 -- use-arch x86_64
```

Lion, Mt. Lion, and Mavericks system Python

Similar to Snow Leopard, you should install outside the system Python path.

But you don't need the use-arch option:

```
python2.7 configure.py — d /Library/Python/2.7/site-packages — b /usr/local/bin  
— n /usr/local/Qt4.8/qsci — v /usr/local/share/sip/PyQt4
```

continue...

```
make -j [#cpus]
```

```
sudo make install
```

If there is a problem with undefined symbols in QtOpenGL on Leopard, edit  
QtOpenGL/makefile and add -undefined dynamic\_lookup to LFLAGS.

Then make again.

#### 5.4.8. Additional dependencies: QScintilla2 Python Module

---

This will create the Qsci.so module in /Library/Python/2.x/site-packages/PyQt4.

Like PyQt, it needs help to not install in system locations.

Snow Leopard: substitute '2.6' for Python version

```
cd <QScintilla2 source dir>
```

```
cd Python
```

```
python2.7 configure.py — o /usr/local/lib — n /usr/local/include \
```

NOTE: PyQt is not compatible with PyQt 4.9, so we will skip that.

Download the latest Qwt 6.0 source (6.1 does not work with the QwtPolar in  
QGIS) from:

<http://sourceforge.net/projects/qwt>

Double-click the tarball to unpack it. Now, cd to the qwt source folder in a  
Terminal.

Type these commands to build and install 6.0.x (assumes v6.0.2, adjust commands  
for other version as needed):

```
cat >> qwtconfig.pri <<EOF
```

```
QWT_CONFIG -= QwtFramework
```

```
EOF
```

```
qmake -spec macx-g++
```

```
make -j [#cpus]
```

```
sudo make install
```

```
sudo install_name_tool — id /usr/local/qwt-6.0.2/lib/libqwt.6.dylib \  
/usr/local/qwt-6.0.2/lib/libqwt.6.dylib
```

The Qwt shared library is now installed in /usr/local/qwt-6.0.x (x is the point version). Remember this for QGIS configuration.

#### 5.4.10. Additional dependencies: Bison

---

The version of bison available by default on Mac OS X is too old so you need to get a more recent one on your system. Download at least version 2.4 from: [ftp.gnu.org/gnu/bison/](http://ftp.gnu.org/gnu/bison/)

Now build and install it to a prefix of /usr/local. Double-click the source tarball to unpack it, then cd to the source folder and:

```
./configure --disable-dependency-tracking CFLAGS= — Os
```

```
make
```

```
sudo make install
```

#### 5.4.11. Additional dependencies: gpsbabel

---

For integrated GPS Tools functions, a gpsbabel executable is required. You can find this at:

<http://www.gpsbabel.org/>

Download the GPSBabel OS X package, and copy GPSBabelFE.app from the disk image to

/Applications.

#### 5.4.12. Optional dependencies: libfcgi

---

If you want to use the QGIS Mapserver, you need libfcgi. This is included on systems up through Snow Leopard, but was dropped at Lion. So, on Lion you need

to get the source from:

<http://www.fastcgi.com/dist/>

Grab the latest fcgi SNAP package there. Double-click the source tarball to unpack it, then cd to the source folder and:

```
./configure --disable-dependency-tracking CFLAGS= — Os
```

```
make
```

```
sudo make install
```

#### 5.4.13. Optional dependencies: OSG & osgEarth

---

If you want the Globe plugin in QGIS (default OFF), OSG and osgEarth are needed.

First, OpenSceneGraph. The main site is very out of date, just go to github:

<http://github.com/openscenegraph/osg/tags>

Download the latest 3.1 version (you can select a tarball when you hover over the entry). Double-click the source tarball to unpack it.

(There is a version numbering oddity in the source, but since we'll be bundling OSG as it's meant to be, it really doesn't matter).

Installation is a bit out of touch with OS X standards, so we'll stage it to a temporary location first. You could stage it to the folder that the OSG source folder is in, or a common staging area like /Users/Shared/unix/osg. Pick a folder not hidden and that doesn't need admin permissions to write to for simplicity.

If you are building on Leopard, its configure forces a old ppc/i386 32bit build. If you want 64bit you need to fix CMakeLists.txt - in a text editor, find the if-block that starts with:

```
ELSEIF(${OSG_OSX_SDK_NAME} STREQUAL "macosx10.6" OR  
${OSG_OSX_SDK_NAME} STREQUAL "macosx10.5")
```

In that section before the next ELSEIF, change:

```
ppc ; i386
```

to:

```
i386 ; x86_64
```

and change:

```
macosx-version-min=10.5
```

to:

```
macosx-version-min=10.6
```

In a new Terminal cd to the source folder and:

```
mkdir build
```

```
cd build
```

```
cmake -D CMAKE_INSTALL_PREFIX=/path/to/some/staging/folder \  
— D OSG_COMPILE_FRAMEWORKS=ON \  
— D OSG_PLUGIN_SEARCH_INSTALL_DIR_FOR_PLUGINS=OFF \  
— D JASPER_LIBRARY=/Library/Frameworks/UnixImageIO.framework \  
— D
```

```
JASPER_INCLUDE_DIR=/Library/Frameworks/UnixImageIO.framework/Heade
```

```
rs \  
— D TIFF_LIBRARY=/Library/Frameworks/UnixImageIO.framework \  
— D
```

```
TIFF_INCLUDE_DIR=/Library/Frameworks/UnixImageIO.framework/Headers \  
..
```

```
make
```

```
make install
```

```
sudo mkdir -p "/Library/Application Support/OpenSceneGraph/PlugIns"
```

Open the staging folder you chose for the CMAKE\_INSTALL\_PREFIX option

above.

Now move all .frameworks from the lib/ folder in the staging area to

/Library/Frameworks. Move the files in the osgPlugins folder in the lib/ folder

to /Library/Application Support/OpenSceneGraph/PlugIns. The bin/ executables

can be left where they are.

Next up is libzip. Get the latest tarball at:

<http://nih.at/libzip/>

Double-click the source tarball to unpack it.

In a new Terminal cd to the source folder and:

```
./configure --disable-dependency-tracking -- disable-shared CFLAGS=— Os
```

```
make
```

```
sudo make install
```

Then it's time for osgEarth. Downloads are also on github:

<http://github.com/gwaldron/osgearth/tags>

Download a tarball for the latest stable release (sorting can be confusing here).

Double-click the source tarball to unpack it.

Note: for now stick with version 2.3. There are compile errors in 2.4 that need

attention.

This one also needs an intermediate staging area. Choose a folder similar to OSG.

In a new Terminal cd to the source folder and:

```
mkdir build
cd build
export PATH="/path/to/osg/staging/folder/bin : $ PATH »
cmake -D CMAKE_INSTALL_PREFIX=/path/to/some/staging/folder \
— D CMAKE_BUILD_TYPE=MinSizeRel \
— D OSGEARTH_BUILD_FRAMEWORKS=true \
..
make
make install
sudo mkdir -p "/Library/Application Support/OpenSceneGraph/Headers"
```

Open the staging folder you chose for the CMAKE\_INSTALL\_PREFIX option above.

Also open the OSG staging path /bin folder from the OSG build.

Move all the .frameworks from the lib/ folder to /Library/Frameworks.

Move the files in the osgPlugins folder in the lib/ folder to /Library/Application Support/OpenSceneGraph/PlugIns. Move the osgEarthDrivers

folder in the include/ folder to /Library/Application Support/OpenSceneGraph/Headers.

(you may need to create this folder)

And as for OSG, you can leave the bin/ executables where they are.

## 5.5. API documentation

=====  
If you want to build a local copy of the API docs (like those at <http://doc.qgis.org/api>) you will need Graphviz and Doxygen installed:

[http://www.graphviz.org/Download\\_macos.php](http://www.graphviz.org/Download_macos.php)

<http://www.stack.nl/~dimitri/doxygen/download.html>

Graphviz is simply installed via a regular Mac package installer. Install it first. It will place some of its binaries in /usr/local/bin/.

For Doxygen, compiling the source is recommended over installing the app.

Double-click the source tarball to unpack it, then cd to the source folder and:

```
./configure
```

```
make -j [#cpus]
```

```
sudo make install
```

The documentation will be output to the build directory, and if using more complete

QGIS.app bundling on install, inside the app in:

QGIS.app/Contents/Resources/doc

## 5.6. QGIS source

=====  
Unzip the QGIS source tarball to a working folder of your choice (/usr/somewhere is not a good choice as it's hidden and requires root privileges). If you are reading this from the source, you've already done this.

If you want to experiment with the latest development sources, go to the github QGIS project page:

<http://github.com/qgis/QGIS>

It should default to the master branch. Click the Downloads button and select Download .tar.gz. Double-click the tarball to unzip it. Alternatively, install git from <http://git-scm.com> and do the following. Make a specific repository directory somewhere, e.g. ~/QGIS/QGIS, and cd into it. The following will read-only clone the master branch to the directory:

```
git init
git remote add — f — t master — m master qgisupstream
git://github.com/qgis/QGIS.git
git merge qgisupstream
```

## 5.7. Configure the build

=====

CMake supports out of source build so we will create a 'build' dir for the build process. OS X uses \$ {HOME}/Applications as a standard user app folder (it gives it the system app folder icon). If you have the correct permissions you may want to build straight into your /Applications folder. The instructions below assume you are building into a \$ {HOME}/Applications directory. You have two interactive options for configuring the build: ccmake or run Terminal commands. ccmake is a curses interface inside Terminal for CMake and

allows a tabular layout for viewing and editing ALL available QGIS source CMake

options. To get started initially run the Terminal method.

In a Terminal cd to the qgis source folder previously downloaded, then:

```
mkdir build
```

```
cd build
```

```
cmake -D CMAKE_INSTALL_PREFIX=~/.Applications \
— D CMAKE_BUILD_TYPE=MINSIZEREL — D ENABLE_TESTS=FALSE \
-D WITH_PYSPATIALITE=FALSE \
— D SPATIALINDEX_LIBRARY=/usr/local/lib/libspatialindex.dylib \
-D SPATIALINDEX_INCLUDE_DIR=/usr/local/include/spatialindex \
— D QWT_LIBRARY=/usr/local/qwt-6.0.2/lib/libqwt.dylib \
— D QWT_INCLUDE_DIR=/usr/local/qwt-6.0.2/include \
— D BISON_EXECUTABLE=/usr/local/bin/bison \
```

```
..
```

Note: Don't forget the .. on the last line, which tells CMake to look for the source files in one directory up.

After the initial Terminal configure, you can use ccmake to make further changes:

```
cd build
```

```
ccmake ..
```

This will automatically find and use the previously installed frameworks, and the GRASS application if installed. Remember to change the Qwt version if a different version was installed, and possibly paths, e.g. for Qwt 6.0.2 installed

```
— D QWT_LIBRARY=/usr/local/qwt-6.0.2/lib/qwt.framework/qwt \
— D QWT_INCLUDE_DIR=/usr/local/qwt-6.0.2/lib/qwt.framework/Headers \
```

If you want to use a newer PostgreSQL client than the default Mac OS X version, e.g. install from [kyngchaos.com](http://kyngchaos.com), set the following option to pg\_config's path:

```
— D POSTGRES_CONFIG=/usr/local/pgsql/bin/pg_config \
```

To build a local copy of the API docs (see API documentation section above):

```

-D WITH_APIDOC=TRUE \
Snow Leopard note: To handle 32-bit Qt (Carbon), create a 32bit python wrapper
script and add arch flags to the configuration:
sudo cat >/usr/local/bin/python32 <<EOF
#! /bin/sh
exec arch — i386 /usr/bin/python2.6 \$ {1+"\$@"}
EOF
sudo chmod +x /usr/local/bin/python32
cmake -D CMAKE_INSTALL_PREFIX=~/.Applications \
— D CMAKE_BUILD_TYPE=MINSIZEREL — D ENABLE_TESTS=FALSE \
-D WITH_INTERNAL_SPATIALITE=FALSE -D
WITH_PYSPATIALITE=FALSE \
— D SPATIALINDEX_LIBRARY=/usr/local/lib/libspatialindex.dylib \
-D SPATIALINDEX_INCLUDE_DIR=/usr/local/include/spatialindex \
— D QWT_LIBRARY=/usr/local/qwt-5.2.2/lib/libqwt.dylib \
— D QWT_INCLUDE_DIR=/usr/local/qwt-5.2.2/include \
— D BISON_EXECUTABLE=/usr/local/bin/bison \

— D CMAKE_OSX_ARCHITECTURES=i386 —
D PYTHON_EXECUTABLE=/usr/local/bin/python32 \

— d /Library/Python/2.7/site-packages/PyQt4 — v /usr/local/share/sip/PyQt4 \
--sip-incdir=/usr/local/include -- pyqt-sipdir=/usr/local/share/sip/PyQt4
cat >>Qsci.pro <<EOF
QMAKE_LFLAGS_PLUGIN -= -dynamiclib
QMAKE_LFLAGS_PLUGIN += -bundle
EOF
qmake -spec macx-g++ Qsci.pro
make -j [#cpus]
sudo make install
The -o and -n options should match the QScintilla2 C++ dylib install options.
5.4.9. Additional dependencies: Qwt
=====
The GPS tracking feature uses Qwt.
Mapserver note: The QGIS Mapserver feature requires fastcgi support. This is
included in
Leopard and Snow Leopard, but was dropped at Lion. To build the Mapserver
component on Leopard and Snow, add the following line before the last line in
the above configuration:
-D WITH_SERVER=TRUE \
On Lion you are on your own to figure out how to install libfcgi and add fcgi
support to the system Apache. Not recommended for the average user.
Globe plugin note: If you want the Globe plugin (and you compiled and installed
OSG/osgEarth),
add the following lines before the last line in the above configuration:
-D WITH_GLOBE=true \
-D OSGEARTH_INCLUDE_DIR="/Library/Application
Support/OpenSceneGraph/Headers » \
-D OSG_PLUGINS_PATH="/Library/Application

```

Support/OpenSceneGraph/PlugIns » \

Bundling note: Older Qt versions may have problems with some Qt plugins and QGIS. The way to handle this is to bundle Qt inside the QGIS application. The default is to bundle Qt (and osg/osgEarth, if configured).

Even better for distribution purposes, to also bundle any extra non-framework, non-standard, libs (ie postgres' libpq) set the bundle value to 2:

— D QGIS\_MACAPP\_BUNDLE=2 \

## 5.8. Building

---

Now we can start the build process (remember the parallel compilation note at to the QGIS application. Assuming you installed QGIS in your home folder:

```
cp — fp /Applications/GPSBabelFE.app/Contents/MacOS/gpsbabel  
~/QGIS.app/Contents/MacOS/bin/
```

If you installed in /Applications, adjust the path accordingly and prefix the whole command with 'sudo '.

### QGIS Mapserver

See the QGIS Mapserver wiki page at:

[http://hub.qgis.org/projects/quantum-gis/wiki/QGIS\\_Server\\_Tutorial](http://hub.qgis.org/projects/quantum-gis/wiki/QGIS_Server_Tutorial)

for instructions on setting up Apache fastcgi and testing Mapserver, including installing the mod-fastcgi that is missing on Lion.

## 6. Setting up the WCS test server on GNU/Linux

---

Requires: Ubuntu / Debian derived distro

These notes are for Ubuntu - other versions and Debian derived distros may require slight variations in package names.

### 6.1. Préparation

---

Note the git repo below will change to the default QGIS repo once this work is integrated into master.

```
git remote add blazek git : //github.com/blazek/Quantum-GIS.git
```

```
git fetch blazek
```

```
git branch --track wcs2 blazek/wcs2
```

```
git checkout wcs2
```

```
cd /var/www/
```

```
sudo mkdir wcs
```

```
sudo chown timlinux wcs
```

```
cd wcs/
```

```
mkdir cgi-bin
```

```
cd cgi-bin/
```

### 6.2. Setup mapserver

---

```
`sudo apt-get install cgi-mapserver`
```

Set the contents of cgi-bin/wcstest-1.9.0 to:

```
#!/bin/sh
```

```

MS_MAPFILE=/var/www/wcs/testdata/qgis-1.9.0/raster/wcs.map
export MS_MAPFILE
/usr/lib/cgi-bin/mapserv
Then do:
chmod +x cgi-bin/wctest-1.9.0
mkdir -p /var/www/wcs/testdata/qgis-1.9.0/raster/
cd /var/www/wcs/testdata/qgis-1.9.0/raster/
cp -r /home/timlinux/QGIS/tests/testdata/raster/* .
Edit wcs.map and set the shapepath to this:
SHAPEPATH « /var/www/wcs/testdata/qgis-1.9.0/raster »

```

Then create /var/www/wcs/7-wcs.qgis.org.conf setting the contents to this:

```

<VirtualHost * : 80>
ServerName wcs.qgis.org
ServerAdmin tim@linfiniti.com
LogLevel warn
LogFormat « %h %l %u %t \ » %r \ » %>s %b \ » % {Referer} i \ » \ » %
{User-Agent} i \ » \"%{forensic-id}n\" combined
CustomLog /var/log/apache2/wcs_qgis.org/access.log combined
ErrorLog /var/log/apache2/wcs_qgis.org/error.log
DocumentRoot /var/www/wcs/html
ScriptAlias /cgi-bin/ /var/www/wcs/cgi-bin/
<Directory « /var/www/wcs/cgi-bin »>
AllowOverride None
Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
Order allow,deny
Allow from all
</Directory>
RewriteEngine on
RewriteRule /1.9.0/wcs /cgi-bin/wctest-1.9.0 [PT]
</VirtualHost>

```

### 6.3. Create a home page

```
=====
```

```

mkdir html
vim html/index.html
Set the contents to:
This is the test platform for QGIS' wcs client. You can use these services
from QGIS directly (to try out WCS for example) by pointing your QGIS to:
http://wcs.qgis.org/1.9.0/wcs

```

### 6.4. Now deploy it

```
=====
```

```

sudo mkdir /var/log/apache2/wcs_qgis.org
sudo chown www-data /var/log/apache2/wcs_qgis.org
cd /etc/apache2/sites-available/
sudo ln -s /var/www/wcs/7-wcs.qgis.org.conf .
cd /var/www/wcs/
sudo a2ensite 7-wcs.qgis.org.conf
sudo /etc/init.d/apache2 reload

```

### 6.5. Debugging

```
=====
```

sudo tail — f /var/log/apache2/wcs\_qgis.org/error.log

## 7. Setting up a Jenkins Build Server

=====

Assumption: You know how to make a working build environment and want to deploy it under Jenkins for continuous integration testing now.

These notes are terse, I will expand on them later as the need arises. The procedure is:

- Install Jenkins and get it configured according to your own preferences
- Make sure you have the git, github, junit etc plugins installed. A complete list of the plugins I have installed follows (note that you almost certainly don't need every plugin listed here):
- External Monitor Job Type Plugin
- LDAP Plugin
- pam-auth
- javadoc
- ant
- Jenkins Subversion Plug-in
- Git Plugin
- Maven 2 Project Plugin
- Jenkins SLOCCount Plug-in
- Jenkins Sounds plugin
- Jenkins Translation Assistance plug-in
- ruby-runtime
- Jenkins CVS Plug-in
- Coverage/Complexity Scatter Plot PlugIn
- Status Monitor Plugin
- Git Parameter Plug-In
- github-api
- GitHub plug-in
- Jenkins Violations plug-in
- git-notes Plugin
- Twitter plug-in
- Jenkins Cobertura Plugin
- Jenkins Gravatar plug-in
- Jenkins SSH Slaves plugin
- Create a Job called 'QGIS'
- Use the following options for your job:
- Job Name: QGIS
- Job Type: Build a free-style software project
- Tick enable project based security (you need to elsewhere configure your Jenkins security to per project settings)
- Allow Anonymous user Read and Discover access
- Set the github project to <https://github.com/qgis/QGIS/>
- Set source code management to Git
- Set repository url to `git://github.com/qgis/QGIS.git`
- In advanced repository url settings set refspec to :  
+refs/heads/master:refs/remotes/origin/master
- Set branch to build to master
- Repository Browser: Auto

— Build triggers: set to Poll SCM and set schedule to \* \* \* \* \* (polls every minute)

- Build - Execute shell and set shell script to:

```
cd build
cmake ..
xvfb-run --auto-servernum -- server-num=1 \
--server-args=" — screen 0 1024x768x24 » \
make Experimental || true
if [ — f Testing/TAG ] ; then
xsltproc ../tests/ctest2junix.xsl \
Testing/^head — n 1 < Testing/TAG`/Test.xml > \
CTestResults.xml
fi
```

— Add Junit post build action and set 'Publish Junit test result report' to:

build/CTestResults.xml

— Email notification: Send separate e-mails to individuals who broke the build

- Jenkins sounds - set up sounds for Failure, Success and Unstable.

— Save

Now open the Job dash board and push something to QGIS and wait a minute to validate automated builds work.

Note: You will need to log in to the Jenkins user account and go to /var/lib/jenkins/jobs/QGIS/workspace, then make a build directory and run the initial cmake setup and then do test build. This process is the same as described elsewhere in this doc.

I based some of the set up from this nice blog article here:

<http://alexott.blogspot.com/2012/03/jenkins-cmakectest.html>

## 8. Debug output and running tests

---

If you are interested in seeing embedded debug output, change the following CMake option :

-D CMAKE\_BUILD\_TYPE=DEBUG (or RELWITHDEBINFO)

This will flood your terminal or system log with lots of useful output from QgsDebugMsg() calls in source code.

If you would like to run the test suite, you will need to do so from the build directory, as it will not work with the installed/bundled app. First set the CMake option to enable tests:

— D ENABLE\_TESTS=TRUE

Then run all tests from build directory:

```
cd build
```

```
make test
```

To run all tests and report to <http://dash.orfeo-toolbox.org/index.php?project=QGIS>

```
cd build
```

```
make Experimental
```

You can define the host name reported via 'make Experimental' by setting a CMake

option:

— D SITE= » my.domain.nouvelle page »

To run specific test(s) (see 'man ctest'):

```
cd build
```

```
# show listing of tests, without running them
ctest --show-only
# run specific C++ or Python test(s) matching a regular expression
ctest --verbose --tests-regex SomeTestName
```

## Annexe 6 : les Opérateurs de cube de donnée

### Why cubes?

Purpose is to provide a framework for giving analyst or any application end-user understandable and natural way of reporting using concept of data Cubes – multidimensional data objects.

It is meant to be used by application builders that want to provide analytical functionality.

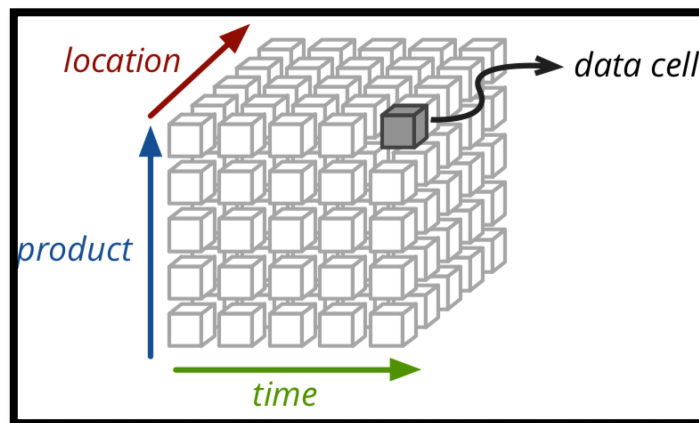


Figure 106 : Cube

Features:

- logical view of analysed data - how analysts look at data, how they think of data, not not how the data are physically implemented in the data stores
- OLAP and aggregated browsing (default backend is for relational database - ROLAP)
- hierarchical dimensions (attributes that have hierarchical dependencies, such as category-subcategory or country-region)
- multiple hierarchies in a dimension
- localizable metadata and data (see *Localization*)
- authentication and authorization of cubes and their data
- pluggable data warehouse – plug-in other cube-like (multidimensional) data sources

The framework is very extensible.

### Cube, Dimensions, Facts and Measures

The framework models the data as a cube with multiple dimensions:

The most detailed unit of the data is a *fact*. Fact can be a contract, invoice, spending, task, etc. Each fact might have a *measure* – an attribute that can be measured, such as: price, amount, revenue, duration, tax, discount, etc.

The *dimension* provides context for facts. Is used to:

- filter queries or reports
- controls scope of aggregation of facts
- used for ordering or sorting
- defines master-detail relationship

Dimension can have multiple *hierarchies*, for example the date dimension might have year, month and day levels in a hierarchy.

## Feature Overview

Core cube features:

- **Workspace** – Cubes analytical workspace (see *docs, reference*)
- **Model** - Description of data (*metadata*): cubes, dimensions, concept hierarchies, attributes, labels, localizations. (see *docs, reference*)
- **Browser** - Aggregation browsing, slicing-and-dicing, drill-down. (see *docs, reference*)
- **Backend** - Actual aggregation implementation and utility functions. (see *docs, reference*)
- **Server** - WSGI HTTP server for Cubes (see *docs, reference*)
- **Formatters** - Data formatters (see *docs, reference*)
- *slicer - Command Line Tool* - command-line tool

## Model

Logical model describes the data from user's or analyst's perspective: data how they are being measured, aggregated and reported. Model is independent of physical implementation of data. This physical independence makes it easier to focus on data instead on ways of how to get the data in understandable form.

More information about logical model can be found in the chapter *Logical Model and Metadata*.

## Browser

Core of the Cubes analytics functionality is the aggregation browser. The browser module contains utility classes and functions for the browser to work.

More information about browser can be found in the chapter *Slicing and Dicing*. See also programming *reference*.

## Backends

Backends provide the actual data aggregation and browsing functionality. Cubes comes with built-in ROLAP backend which uses SQL database using SQLAlchemy.

Framework has modular nature and supports multiple database backends, therefore different ways of cube computation and ways of browsing aggregated data.

Multiple backends can be used at once, even multiple sources from the same backend might be used in the analytical workspace.

More about existing backends can be found in the *backends documentation*. See also the backends programming *reference*.

## Logical Model

business and analyst's point of view on data

- Dimensions with multiple hierarchies
- User oriented metadata
- Dimension templates - define complex dimensions
- Localization of model and data

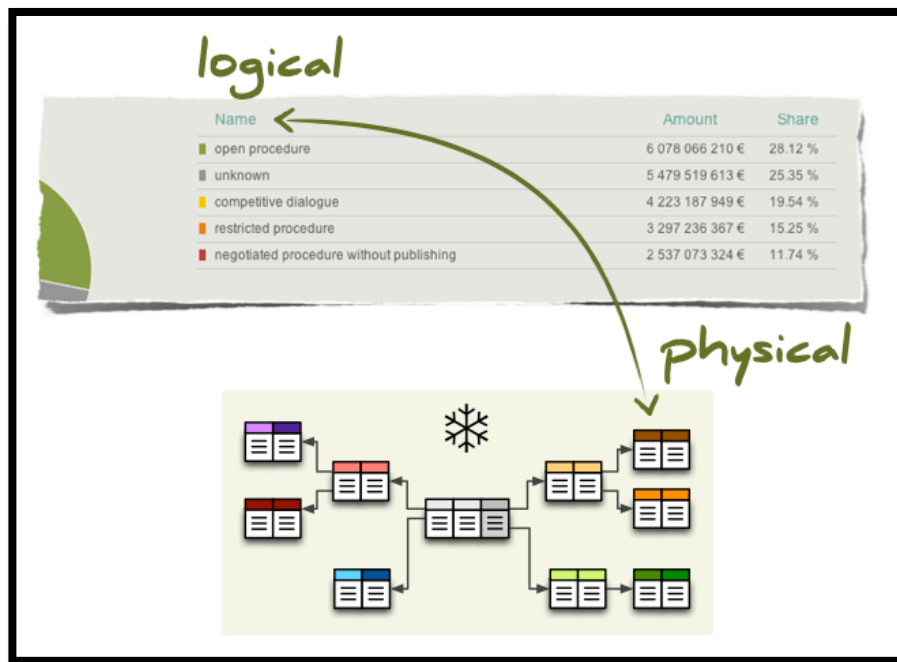


Figure 107 : modèle logique et physique de données

## Aggregated Browsing

easy development of exploration tools

- Slice and dice through dimensions
- Drill-down through any hierarchy
- Automatic next level selection, if desired
- Get dimension values or all facts within a cut

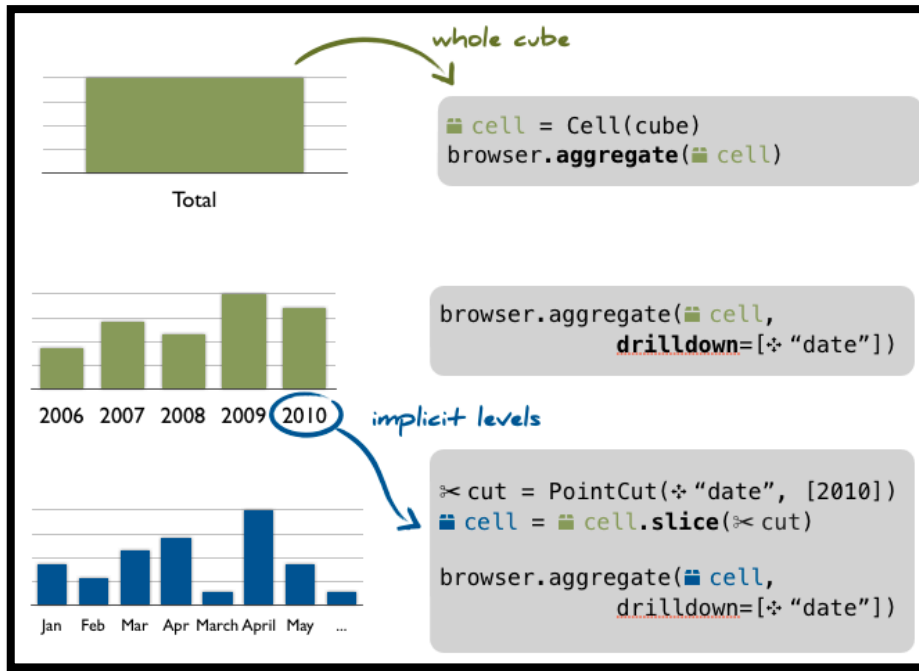


Figure 108 : Agrégation

## Slicer

Slicer is a HTTP OLAP cube server for aggregation queries.

- Easy drilling-down
- Slicing and dicing
- Serves aggregates, dimension details, facts
- Provides all necessary metadata for a reporting application

Structured responses are in JSON format with rich metadata for easier reporting application development or reporting integration.

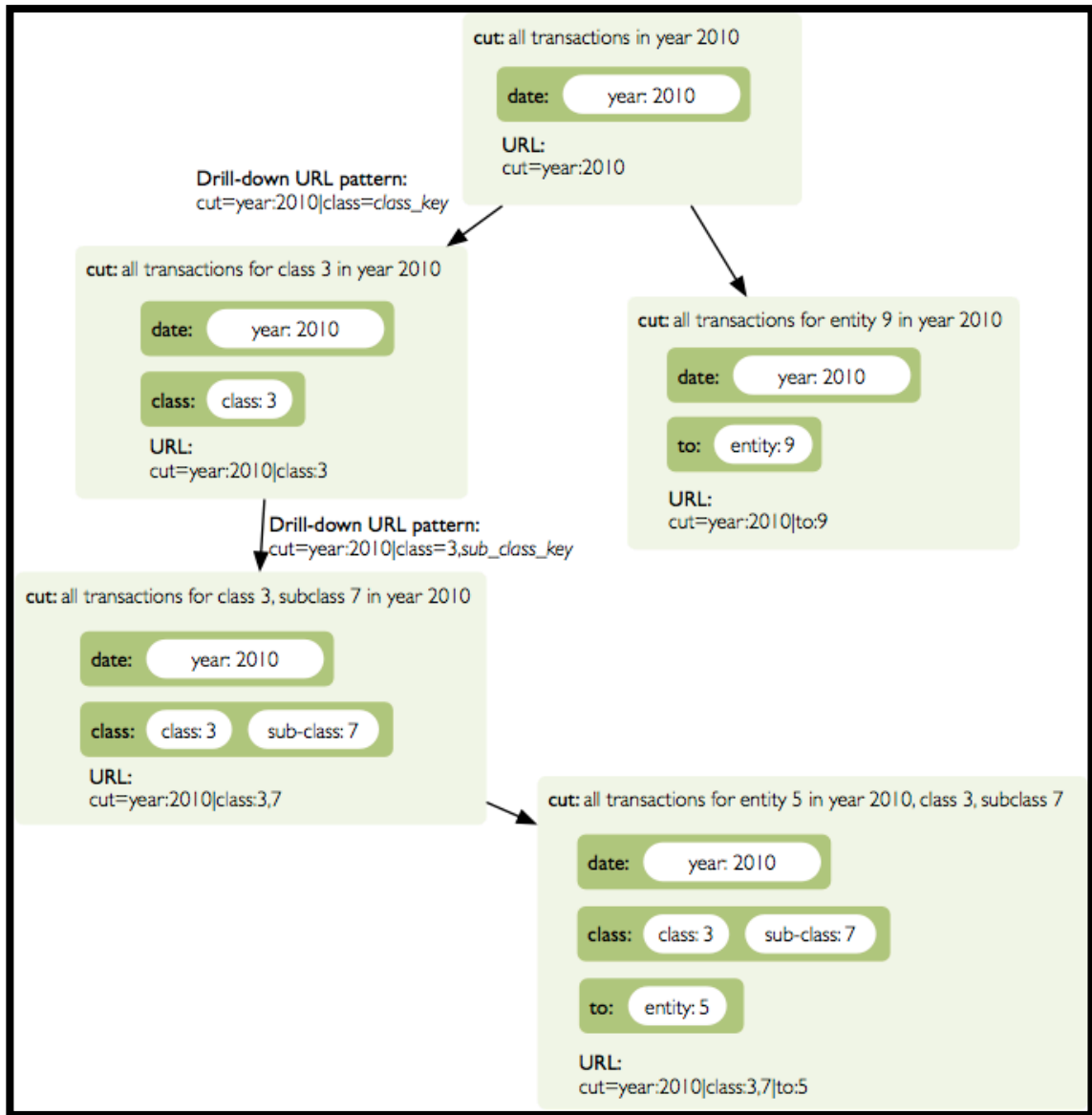


Figure 109 : Slicer

## **Backend and SQL**

One of the backends shipped with the framework is SQL. It is powered by the SQLAlchemy which supports multiple databases including PostgreSQL, MySQL, Oracle, simple sqlite and many others.

Features:

Easy to prototype on top of existing, arbitrary star or snowflake looking schemas.

Logical-to-physical mapping that supports multiple database schemas in databases such as PostgreSQL or Oracle.

Denormalization by view or materialized by table.

## Annexe 7 : Création du Plugin

Source : [http://www.qgistutorials.com/en/docs/building\\_a\\_python\\_plugin.html](http://www.qgistutorials.com/en/docs/building_a_python_plugin.html)

### Building a Python Plugin

Plugins are a great way to extend the functionality of QGIS. You can write plugins using Python that can range from adding a simple button to sophisticated toolkits. This tutorial will outline the process involved in setting up your development environment, designing the user interface for a plugin and writing code to interact with QGIS. Please review the *Getting Started With Python Programming* tutorial to get familiar with the basics.

### Overview of the Task

We will develop a simple plugin called Save Attributes that will allow users to pick a vector layer and write its attributes to a CSV file.

### Get the Tools

#### Qt Creator

Qt is a software development framework that is used to develop applications that run on Windows, Mac, Linux as well as various mobile operating systems. QGIS itself is written using the Qt framework. For plugin development, we will use an application called Qt Creator to design the interface for our plugin.

Download and install the Qt Creator software from SourceForge

### Python Bindings for Qt

Since we are developing the plugin in Python, we need to install the python bindings for Qt. The method for installing these will depend on the platform you are using. For building plugins we need the pyrcc4 command-line tool.

#### Windows

Download the OSGeo4W network installer and choose *Express Desktop Install*. Install the package QGIS. After installation, you will be able to access the pyrcc4 tool via the *OSGeo4W Shell*.

#### Mac

Install the Homebrew package manager. Install PyQt package by running the following command:

```
brew install pyqt
```

#### Linux

Depending on your distribution, find and install the python-qt4 package. On Ubuntu and Debian-based distributions, you can run the following command:

```
sudo apt-get install python-qt4
```

### A Text Editor or a Python IDE

Any kind of software development requires a good text editor. If you already have a favorite text editor or an IDE (Integrated Development Environment), you may use it for this tutorial. Otherwise, each platform offers a wide variety of free or paid options for text editors. Choose the one that fits your needs.

This tutorial uses Notepad++ editor on Windows.

## Windows

Notepad++ is a good free editor for windows. Download and install the Notepad++ editor.

Note

If you are using Notepad++, makes sure to check *Replace by space* at *Settings* › *Preferences* › *Tab Settings*. Python is very sensitive about whitespace and this setting will ensure tabs and spaces are treated properly.

## Plugin Builder plugin

There is a helpful QGIS plugin named Plugin Builder which creates all the necessary files and the boilerplate code for a plugin. Find and install the Plugin Builder plugin. See *Using Plugins* for more details on how to install plugins.

## Plugins Reloader plugin

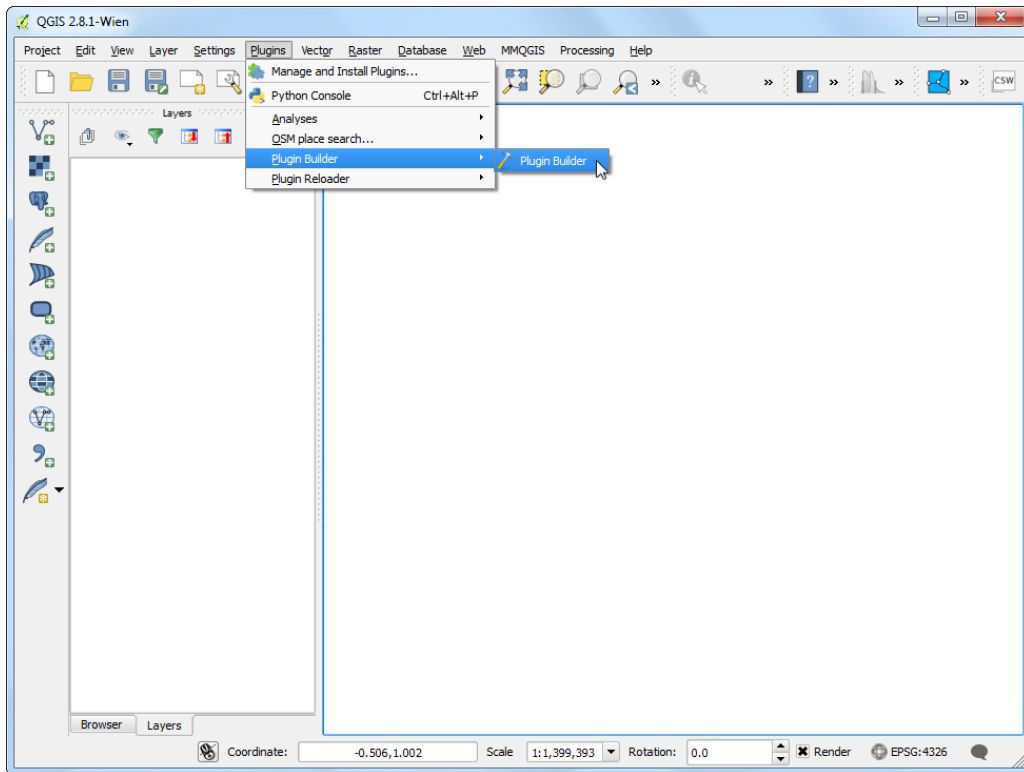
This is another helper plugin which allows iterative development of plugins. Using this plugin, you can change your plugin code and have it reflected in QGIS without having to restart QGIS every time. Find and install the Plugin Reloader plugin. See *Using Plugins* for more details on how to install plugins.

Note

Plugin Reloader is an experimental plugin. Make sure you have checked *Show also experimental plugins* in *Plugin Manager* settings if you cannot find it.

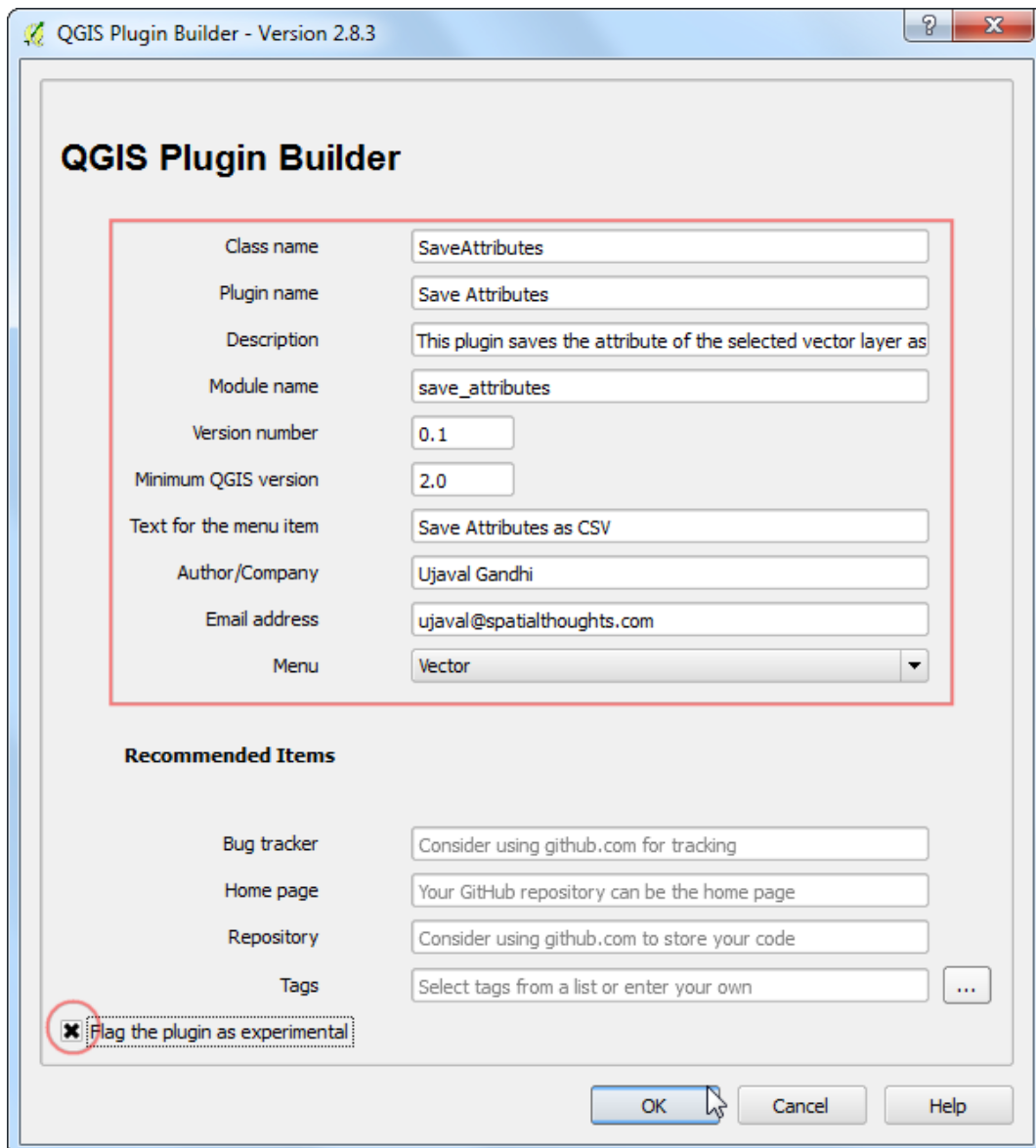
## Procedure

1. Open QGIS. Go to *Plugins* › *Plugin Builder* › *Plugin Builder*.



*Figure 110 : Installation du Plugin-BUILDER*

2. You will see the *QGIS Plugin Builder* dialog with a form. You can fill the form with details relating to our plugin. The *Class name* will be the name of the Python Class containing the logic of the plugin. This will also be the name of the folder containing all the plugin files. Enter *SaveAttributes* as the class name. The *Plugin name* is the name under which your plugin will appear in the *Plugin Manager*. Enter the name as *Save Attributes*. Add a description in the *Description* field. The *Module name* will be the name of the main python file for the plugin. Enter it as *save\_attributes*. Leave the version numbers as they are. The *Text for menu item* value will be how the users will find your plugin in QGIS menu. Enter it as *Save Attributes as CSV*. Enter your name and email address in the appropriate fields. The *Menu* field will decide where your plugin item is added in QGIS. Since our plugin is for vector data, select *Vector*. Check the *Flag the plugin as experimental* box at the bottom. Click *OK*.



*Figure 111 : Creation du Plugin*

- Next, you will be prompted to choose a directory for your plugin. You need to browse to the QGIS python plugin directory on your computer and select Select Folder. Typically, a .qgis2/ directory is located in your home directory. The plugin folder location will depend on your platform as follows: (Replace username with your login name)

#### **Windows**

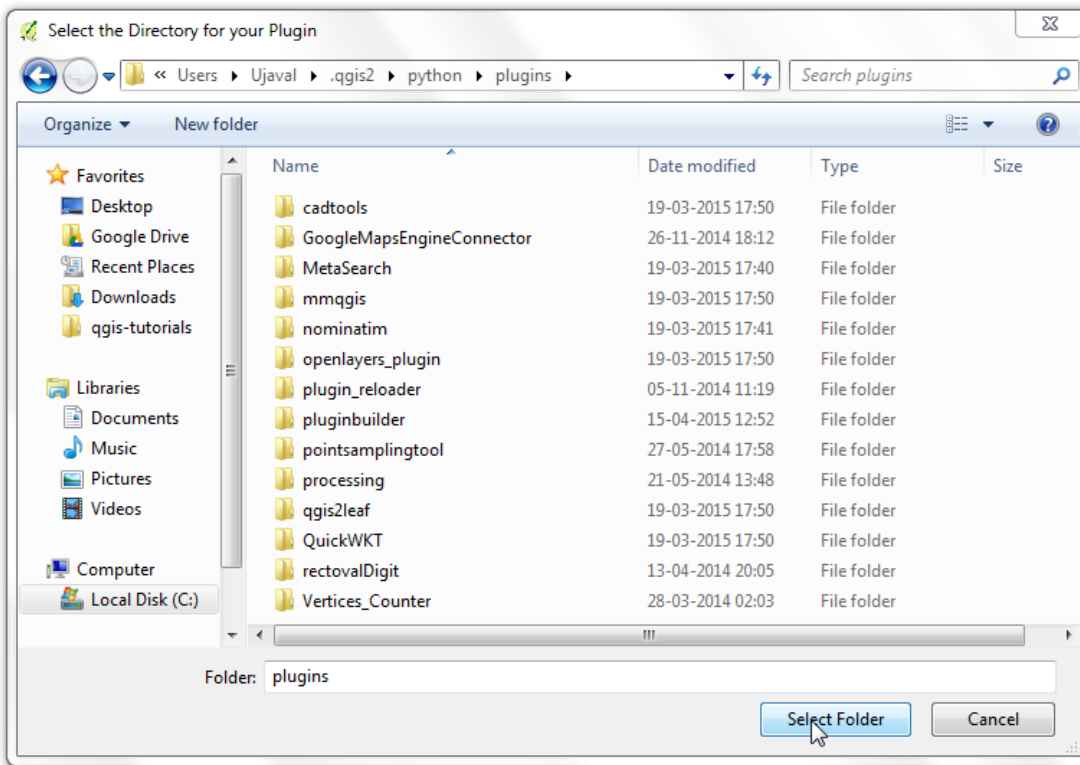
c:\Users\username\.qgis2\python\plugins

#### **Mac**

/Users/username/.qgis2/python/plugins

#### **Linux**

/home/username/.qgis2/python/plugins



*Figure 112 : Enregistrement du Plugin*

4. You will see a confirmation dialog once your plugin template is created. Note the path to the plugin folder.

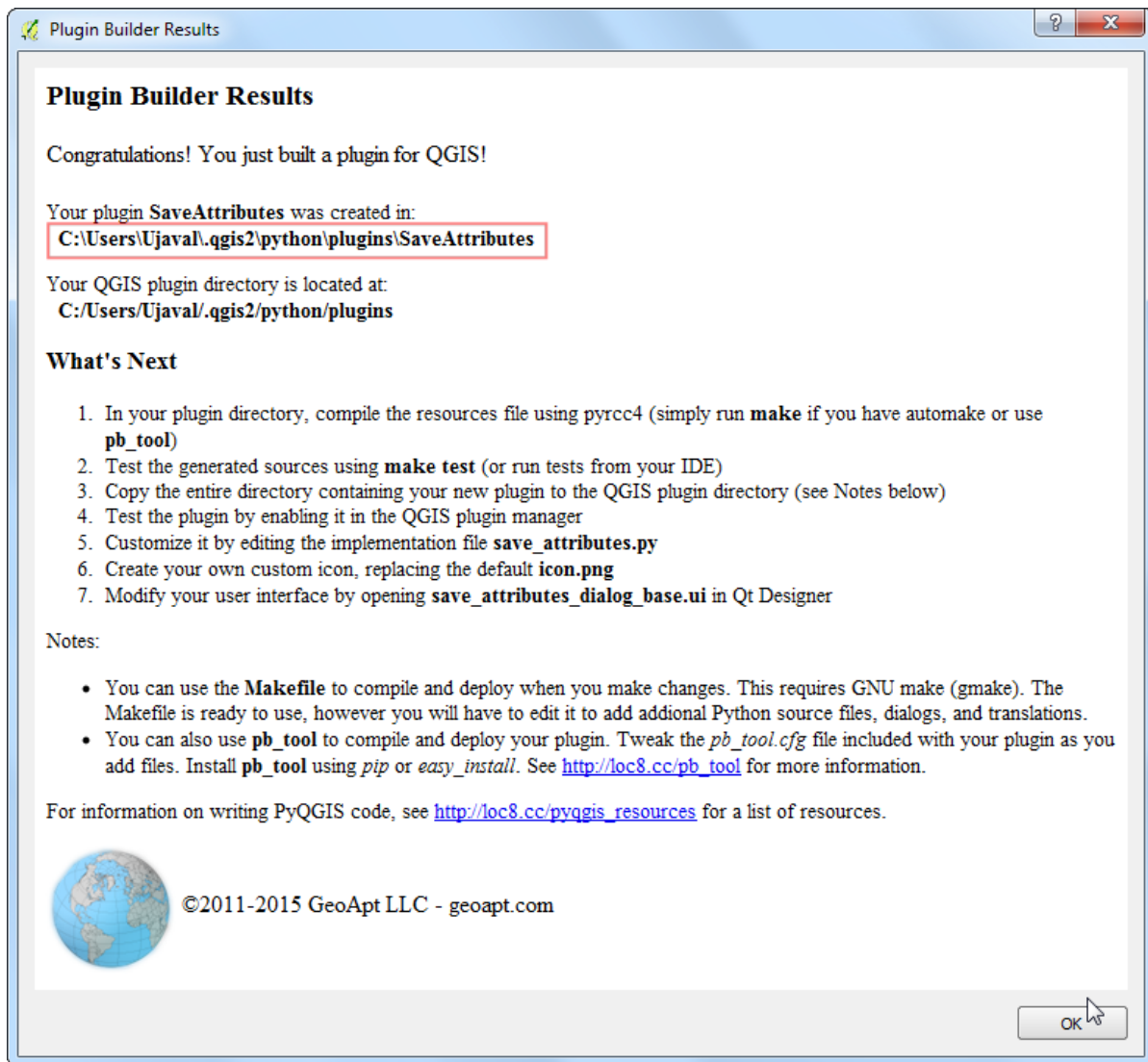


Figure 113 : Fenêtre Résultat

5. Before we can use the newly created plugin, we need to compile the resources.qrc file that was created by Plugin Builder. Launch the OSGeo4W Shell on windows or a terminal on Mac or Linux.

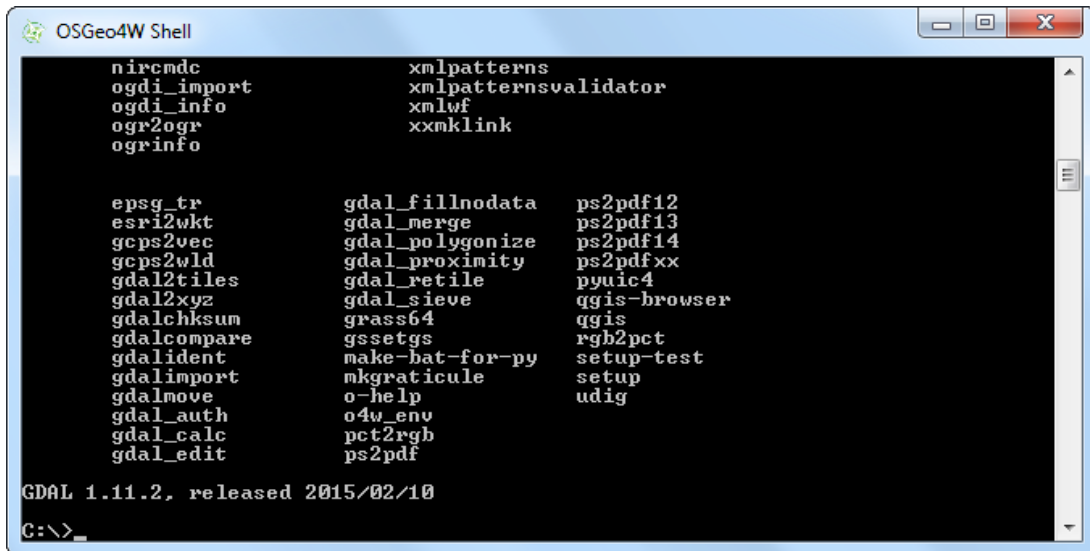


Figure 114 : Fenêtre OSGeo4W Shell

- Browse to the plugin directory where the output of Plugin Builder was created. You can use the cd command followed by the path to the directory.

```
cd c:\Users\username\.qgis2\python\plugins\SaveAttributes
```

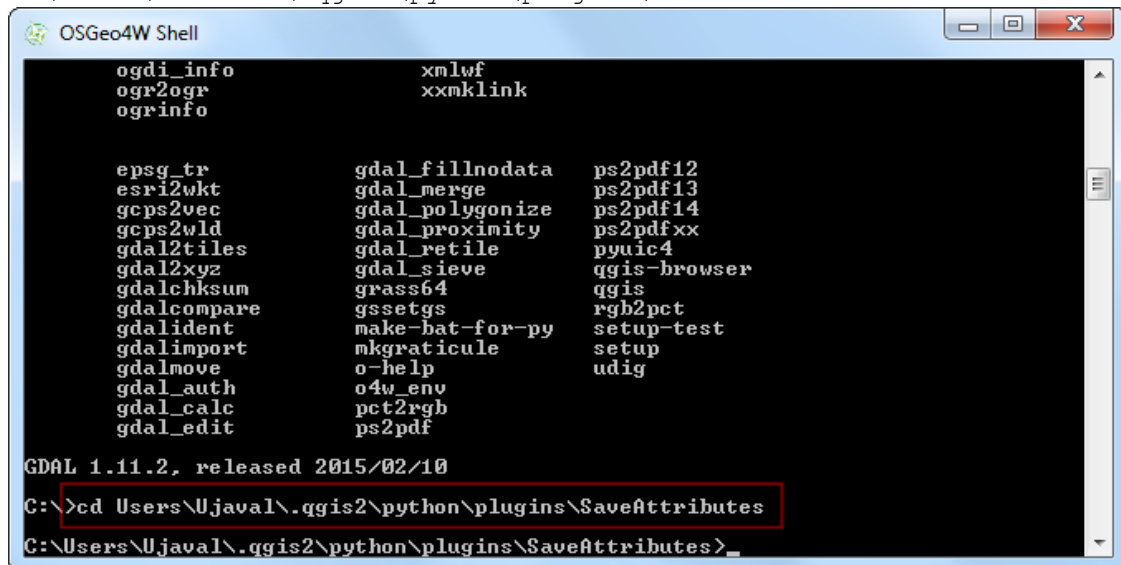


Figure 115 : Acces au Dossier Créé

- Once you are in the directory, type make. This will run the pyrcc4 command that we had installed as part of Qt bindings for Python.

```
make
```

```

OSGeo4W Shell

epsq_tr          gdal_fillnodata ps2pdf12
esri2wkt         gdal_merge       ps2pdf13
gcps2vec        gdal_polygonize  ps2pdf14
gcps2wld        gdal_proximity   ps2pdfxx
gdal2tiles      gdal_retile      pyuic4
gdal2xyz        gdal_sieve       qgis-browser
gdalchksum      grass64          qgis
gdalcompare     gssetgs         rgb2pct
gdalident       make-bat-for-py  setup-test
gdalimport      mkgraticule     setup
gdalmove        o-help          udig
gdal_auth       o4w_env
gdal_calc       pct2rgb
gdal_edit       ps2pdf

GDAL 1.11.2, released 2015/02/10

C:\>cd Users\Ujaval\.qgis2\python\plugins\SaveAttributes
C:\Users\Ujaval\.qgis2\python\plugins\SaveAttributes>make
pyrcc4 -o resources_rc.py resources.qrc
C:\Users\Ujaval\.qgis2\python\plugins\SaveAttributes>_

```

Figure 116 : Compilation de fichier

- Now we are ready to have a first look at the brand new plugin we created. Close QGIS and launch it again. Go to *Plugins* ▶ *Manage and Install plugins* and enable the Save Attributes plugin in the *Installed* tab. You will notice that there is a new icon in the toolbar and a new menu entry under *Vector* ▶ *Save Attributes* ▶ *Save Attributes as CSV*. Select it to launch the plugin dialog.

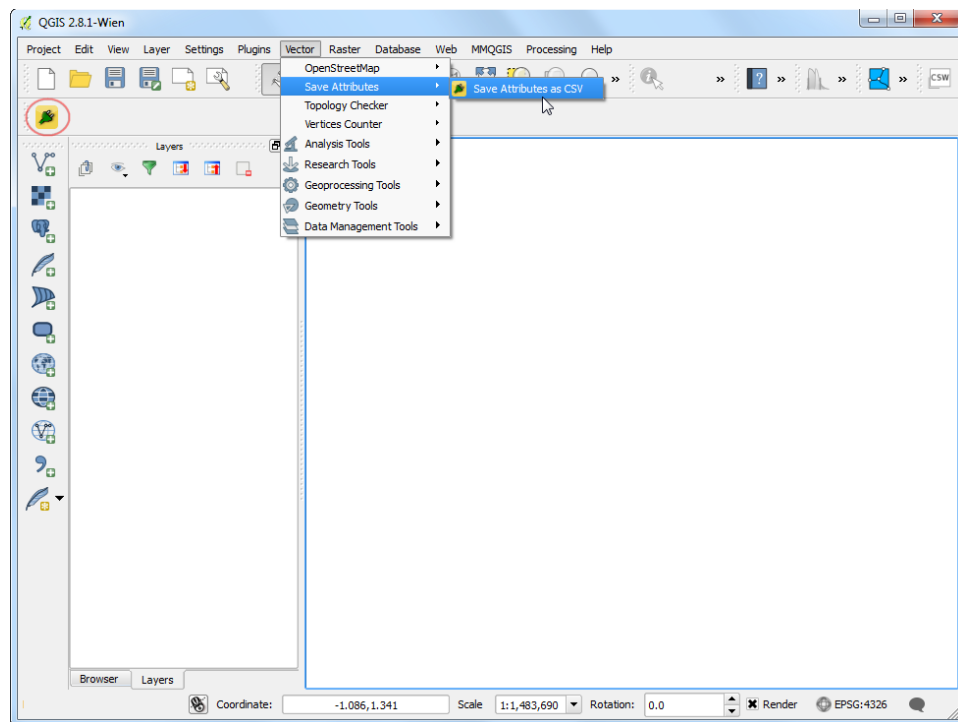
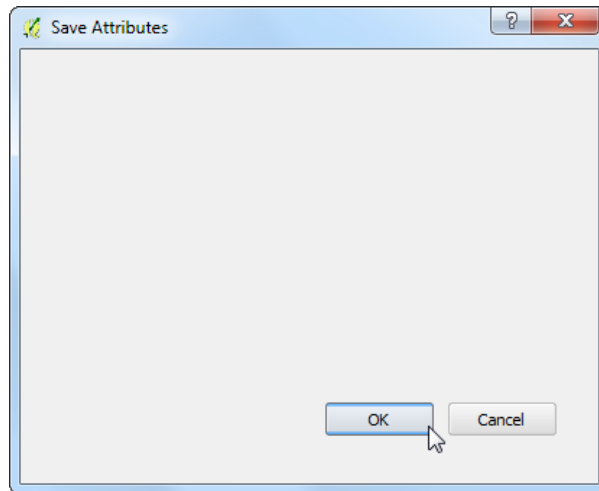


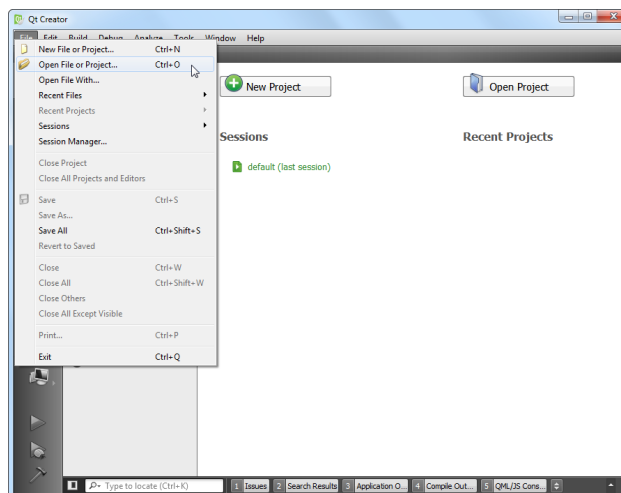
Figure 117 : Integration de Plug-in

- You will notice a new blank dialog named *Save Attributes*. Close this dialog.



*Figure 118 : Fenêtre Principale*

10. We will now design our dialog box and add some user interface elements to it. Open the Qt Creator program and to *File* → *Open File or Project...*



*Figure 119 : Fenêtre QtCreator*

11. Browse to the plugin directory and select the `save_attributes_dialog_base.ui` file. Click *Open*.

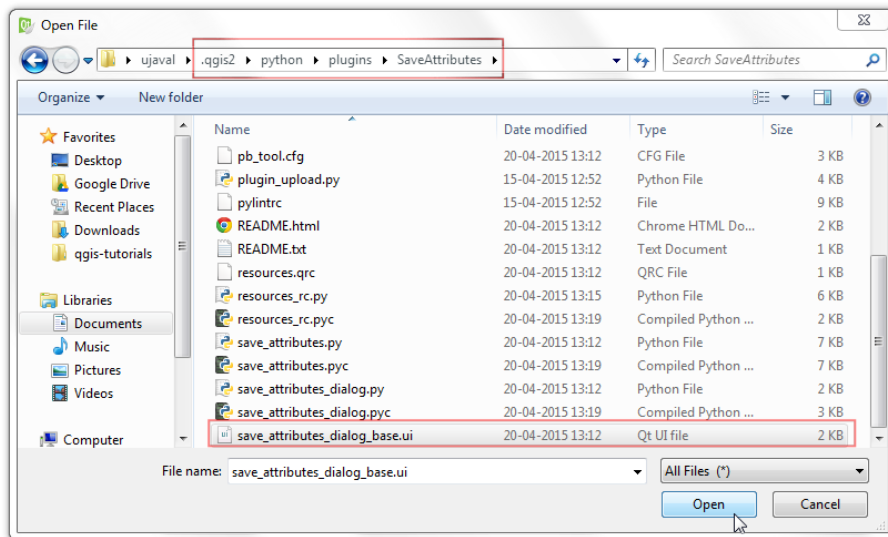


Figure 120 : Fenêtre de plug-in

- You will see the blank dialog from the plugin. You can drag-and-drop elements from the left-hand panel on the dialog. We will add a *Combo Box* type of *Input Widget*. Drag it to the plugin dialog.

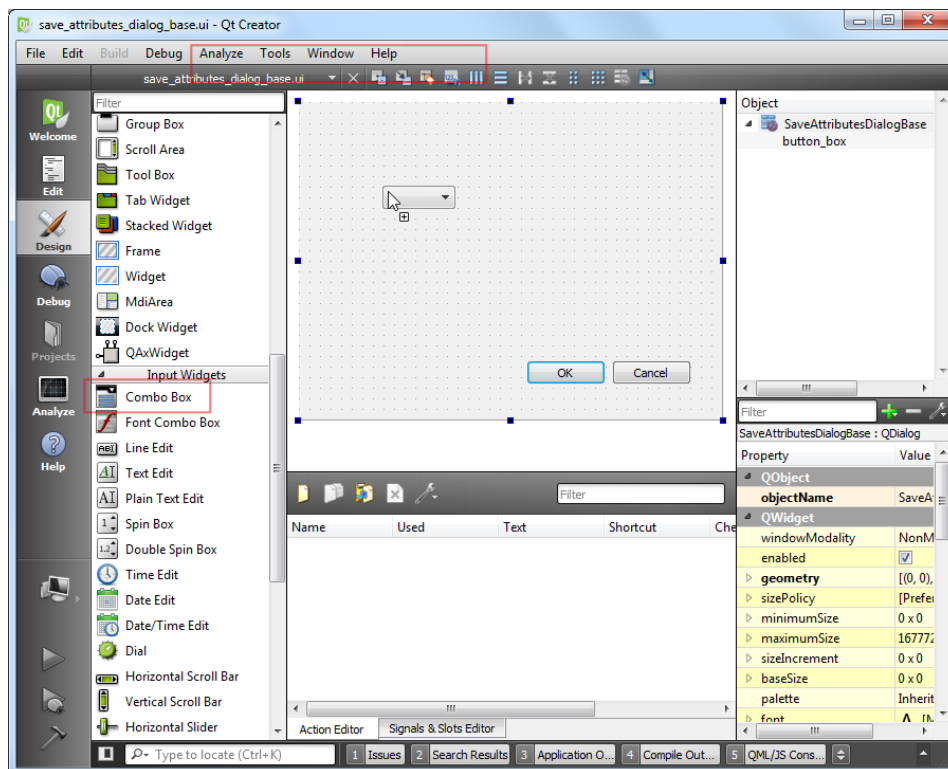


Figure 121: Fenêtre en mode creation sous QtCreator

- Resize the combo box and adjust its size. Now drag a *Label* type *Display Widget* on the dialog.

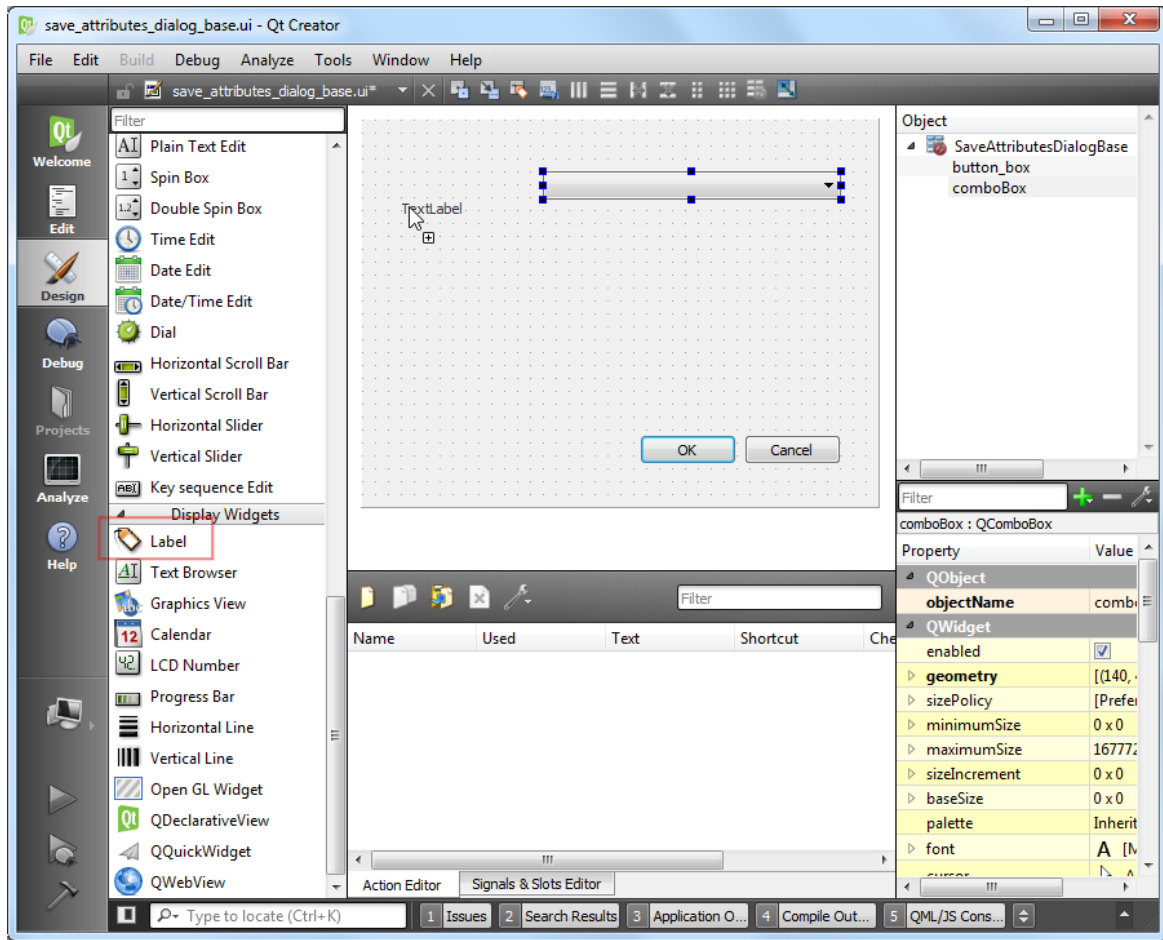


Figure 122 : Ajout des composants sur la fenêtre

14. Click on the label text and enter Select a layer.

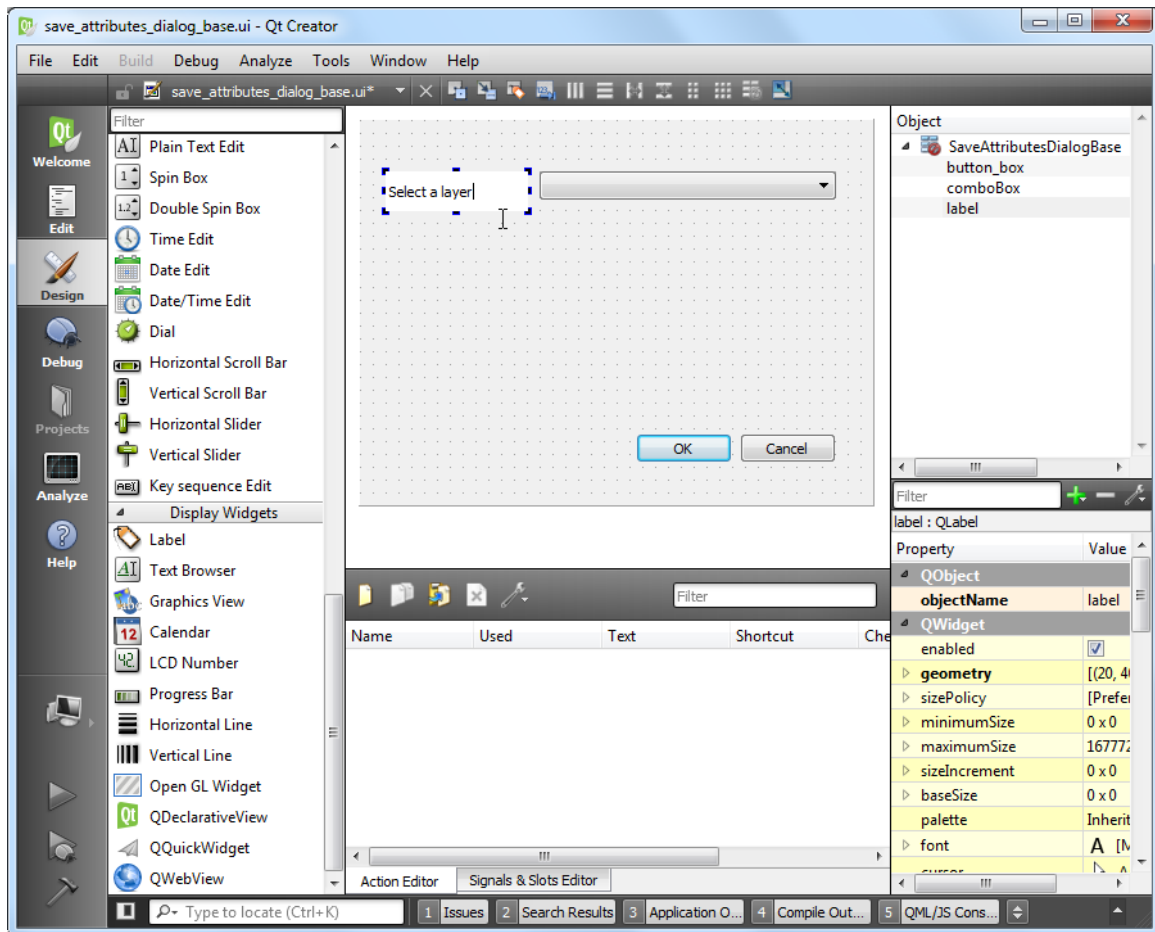


Figure 123 : Ajout d'un comosant "QwidgetComboBox"

15. Save this file by going to *File* ▶ *Save save\_attributes\_dialog\_base.ui*. Note the name of the combo box object is `comboBox`. To interact with this object using python code, we will have to refer to it by this name.

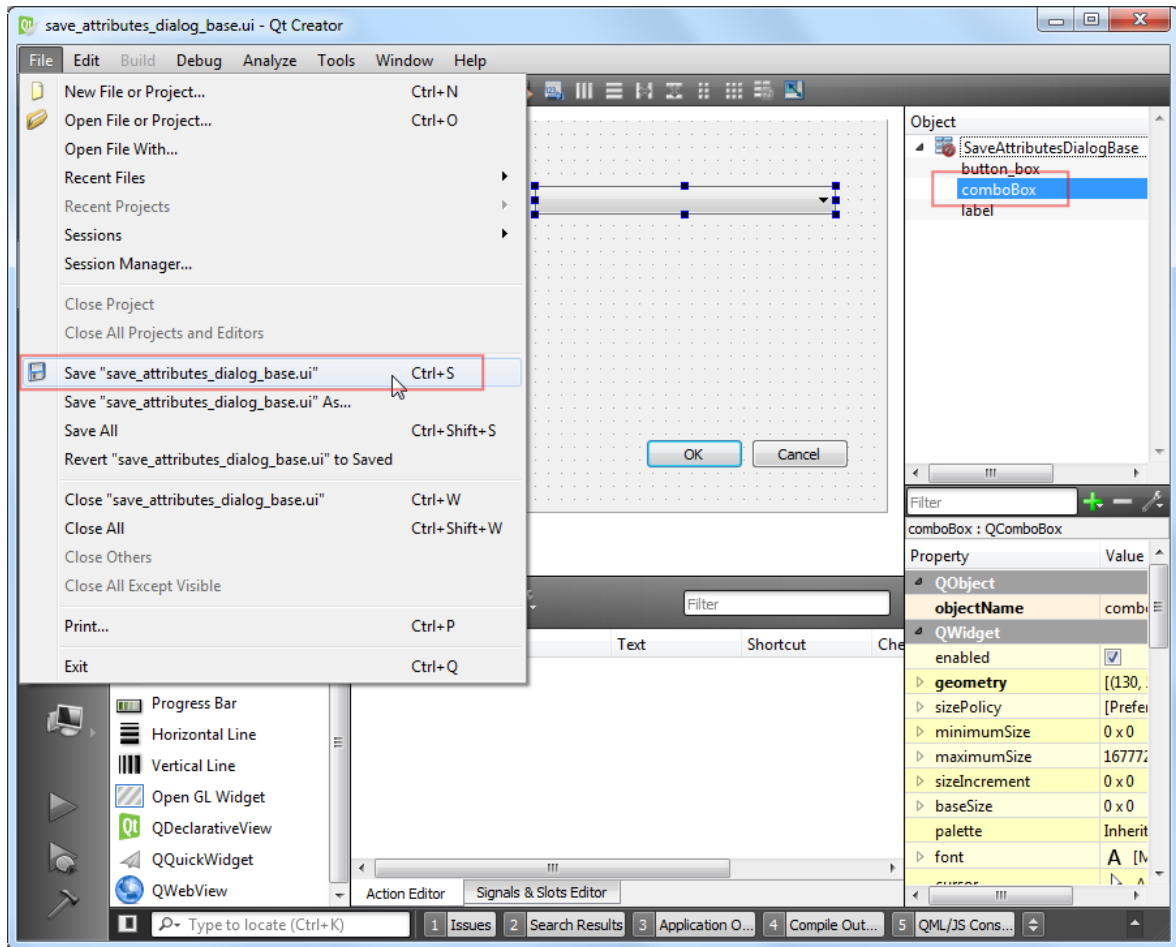
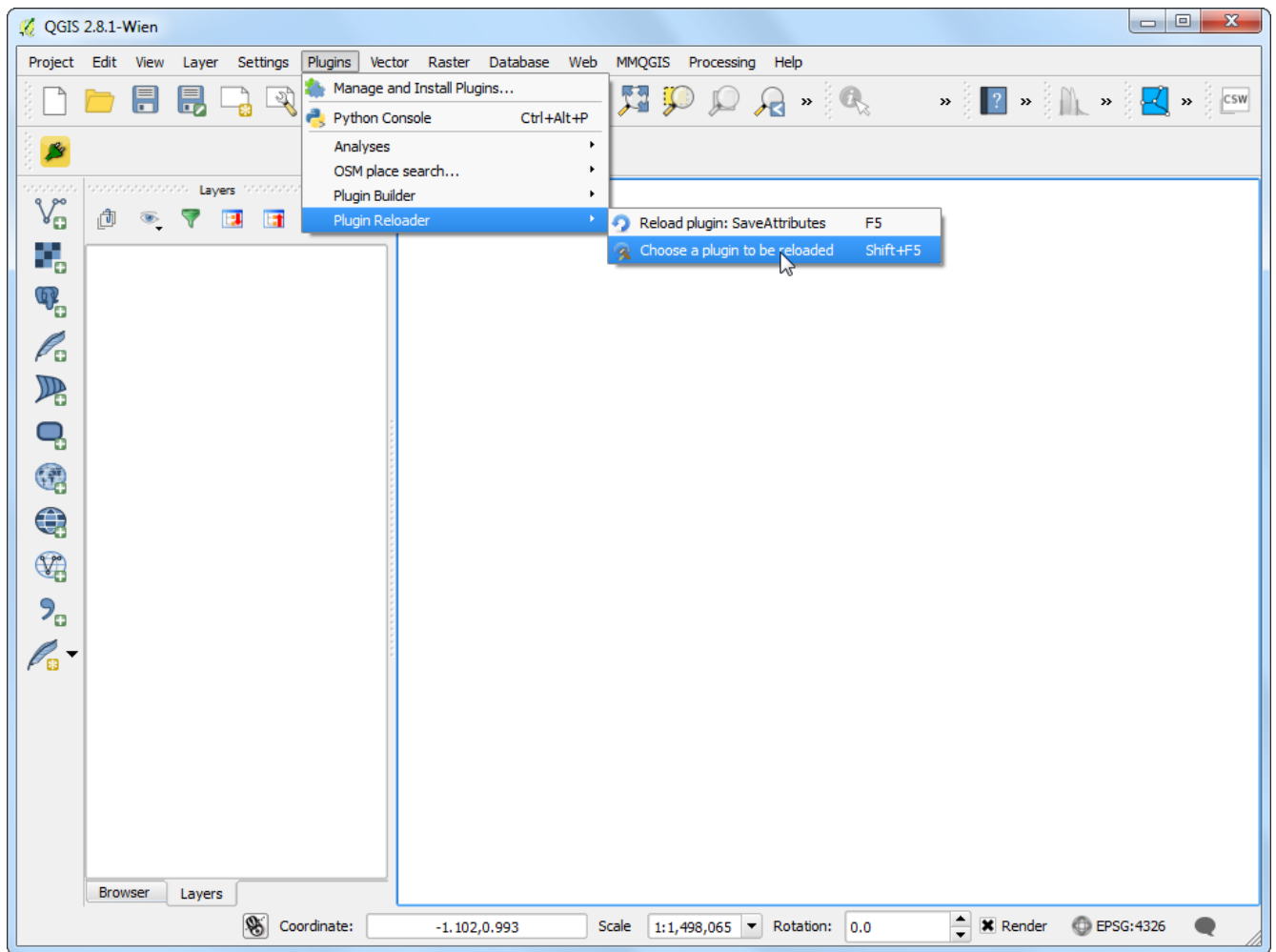


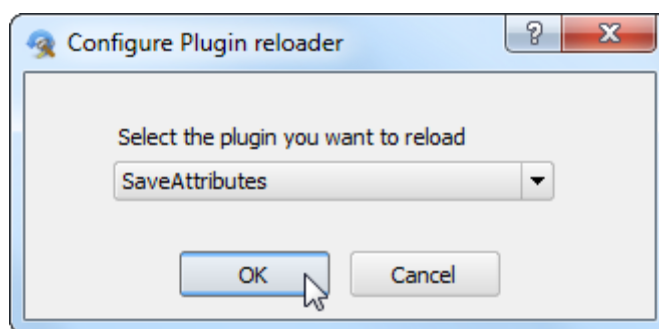
Figure 124 : Sauvegarde de la fenêtre

16. Let's reload our plugin so we can see the changes in the dialog window. Go to *Plugin* ▶ *Plugin Reloader* ▶ *Choose a plugin to be reloaded*.



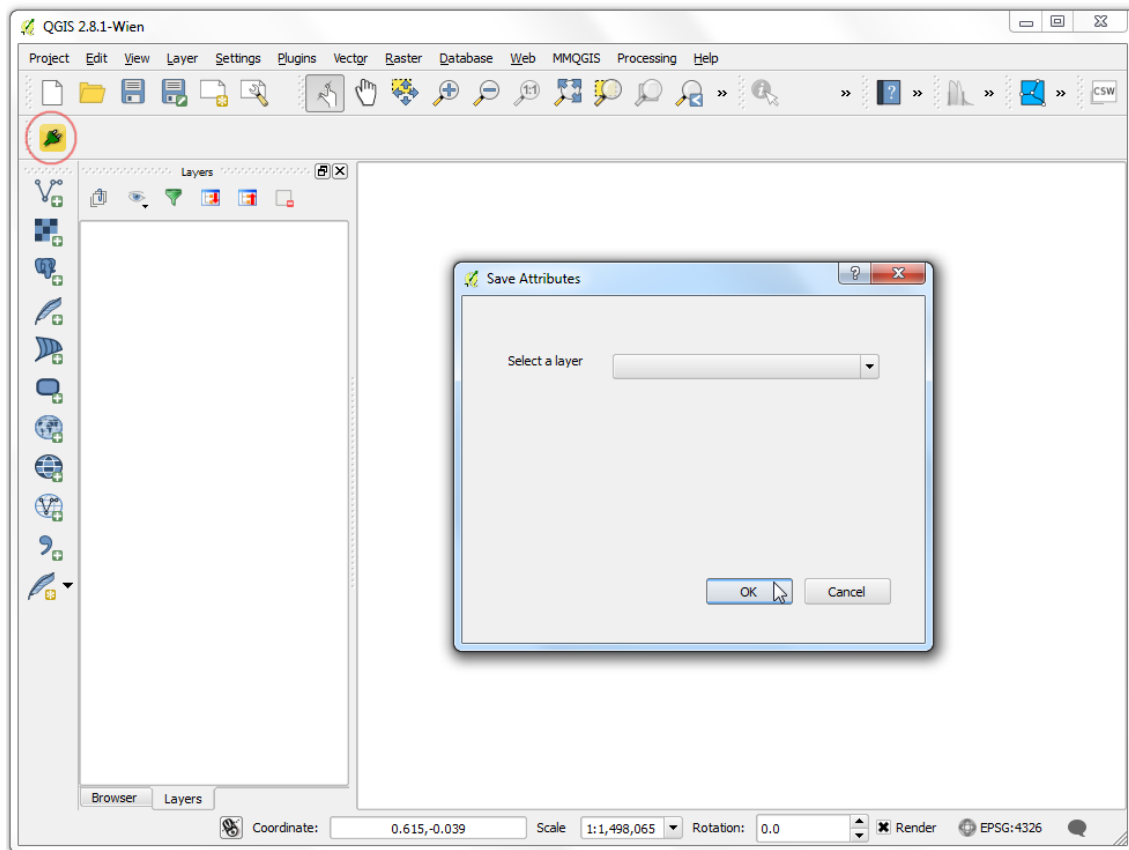
*Figure 125 : Rechargement du Plugin Developpé*

17. Select SaveAttributes in the *Configure Plugin reloader* dialog.



*Figure 126 : choix du plugin à recharger*

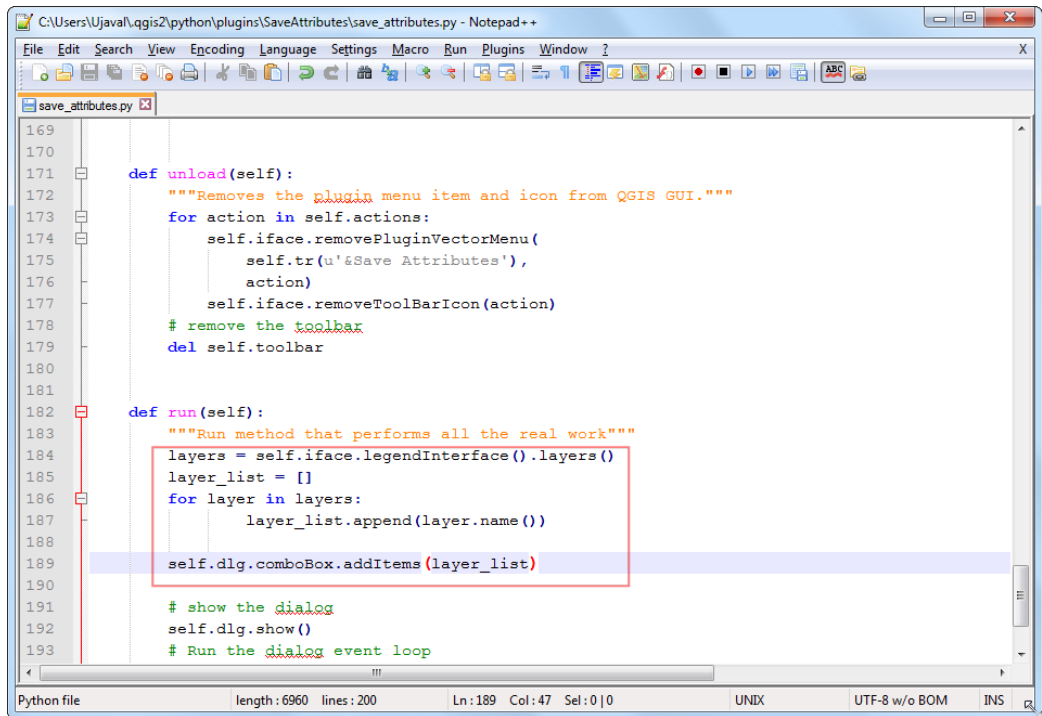
18. Now click the *Save Attributes as CSV* button. You will see the newly designed dialog box.



*Figure 127 : Integration définitif du plugin*

19. Let's add some logic to the plugin that will populate the combo box with the layers loaded in QGIS. Go to the plugin directory and load the file `save_attributes.py` in a text editor. Scroll down and find the `run(self)` method. This method will be called when you click the toolbar button or select the plugin menu item. Add the following code at the beginning of the method. This code gets the layers loaded in QGIS and adds it to the `comboBox` object from the plugin dialog.

```
layers = self.iface.legendInterface().layers()
layer_list = []
for layer in layers:
    layer_list.append(layer.name())
    self.dlg.comboBox.addItem(layer_list)
```



```
169
170
171     def unload(self):
172         """Removes the plugin menu item and icon from QGIS GUI."""
173         for action in self.actions:
174             self.iface.removePluginVectorMenu(
175                 self.tr(u'&Save Attributes'),
176                 action)
177             self.iface.removeToolBarIcon(action)
178         # remove the toolbar
179         del self.toolbar
180
181
182     def run(self):
183         """Run method that performs all the real work"""
184         layers = self.iface.legendInterface().layers()
185         layer_list = []
186         for layer in layers:
187             layer_list.append(layer.name())
188         self.dlg.comboBox.addItem(layer_list)
189
190
191         # show the dialog
192         self.dlg.show()
193         # Run the dialog event loop
```

Figure 128 : Partie du code Python

20. Back in the main QGIS window, reload the plugin by going to *Plugins* ▸ *Plugin Reloader* ▸ *Reload plugin: SaveAttributes*. Alternatively, you can just press F5. To test this new functionality, we must load some layers in QGIS. After you load some data, launch the plugin by going to *Vector* ▸ *Save Attributes* ▸ *Save Attributes as CSV*.

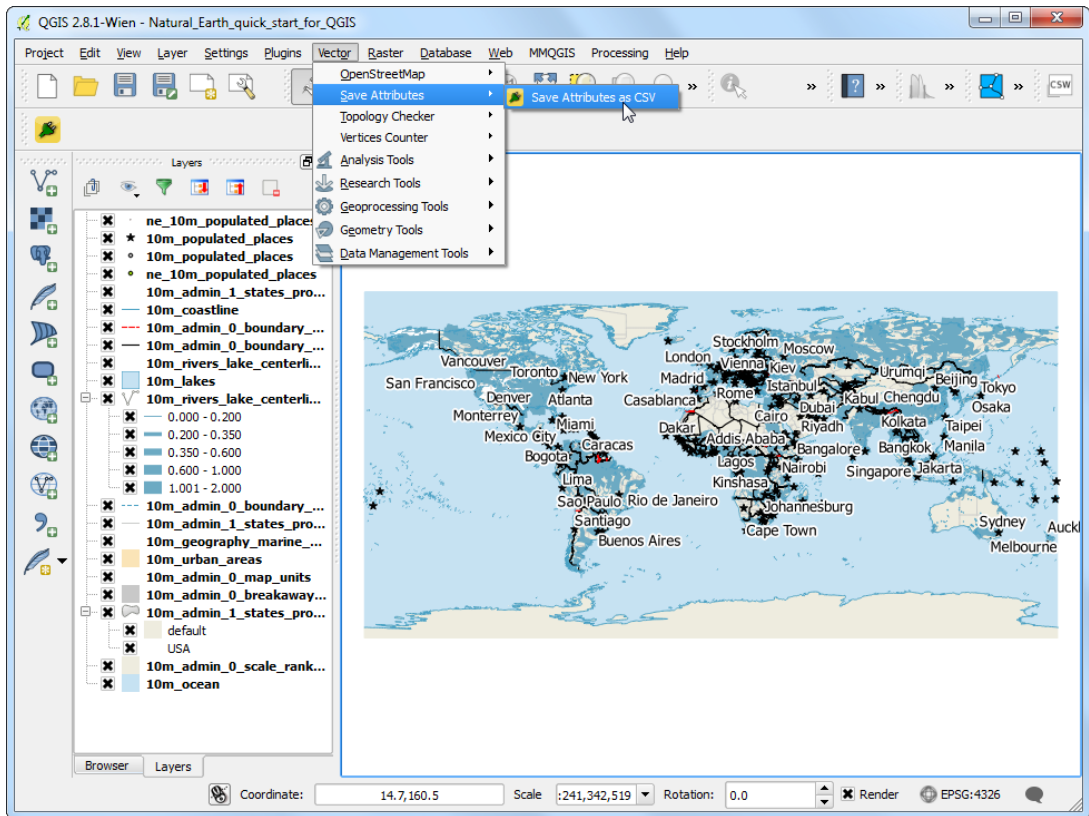


Figure 129 : Affichage des données

21. You will see that our combo box is now populated with the layer names that are loaded in QGIS.

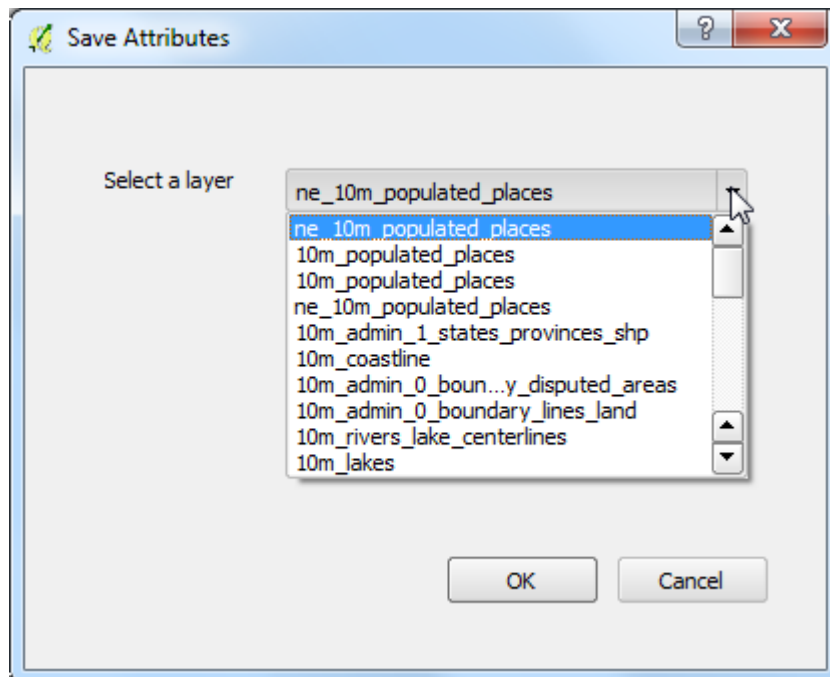


Figure 130 : Exemple des résultats

22. Let's add remaining user interface elements. Switch back to Qt Creator and load the `save_attributes_dialog_base.ui` file. Add a Label *Display Widget* and change the text to Select output file. Add a LineEdit type *Input Widget* that will show the output file path that the user has chosen. Next, add a Push Button type *Button* and change the button label to .... Note the object names of the widgets that we will have to use to interact with them. Save the file.

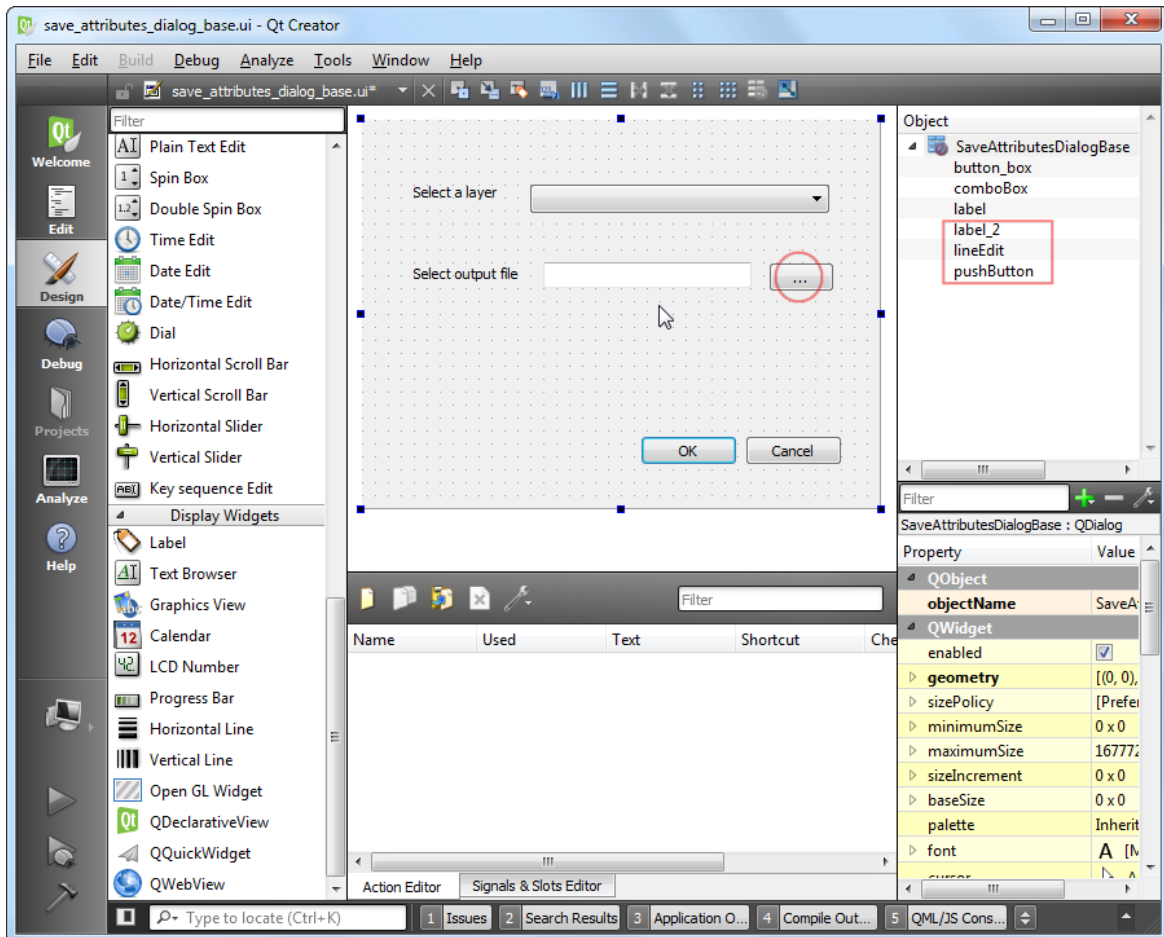


Figure 131 : Ajout de composant

23. We will now add python code to open a file browser when the user clicks the ... push button and show the select path in the line edit widget. Open the `save_attributes.py` file in a text editor. Add `QFileDialog` to our list of imports at the top of the file.

```

1  # -*- coding: utf-8 -*-
2  """
3  /*****
4  SaveAttributes
5  A QGIS plugin
6  This plugin saves the attribute of the selected vector layer as a CSV file.
7  -----
8  begin           : 2015-04-20
9  git sha         : $Format:%H$
10 copyright      : (C) 2015 by Ujaval Gandhi
11 email          : ujaval@spatialthoughts.com
12 *****/
13
14 /*****
15 *
16 * This program is free software; you can redistribute it and/or modify
17 * it under the terms of the GNU General Public License as published by
18 * the Free Software Foundation; either version 2 of the License, or
19 * (at your option) any later version.
20 *
21 *****/
22 """
23 from PyQt4.QtCore import QSettings, QTranslator, qVersion, QCoreApplication
24 from PyQt4.QtGui import QAction, QIcon, QFileDialog
25 # Initialize Qt resources from file resources.py
26 import resources_rc

```

Figure 132 : Ajout de QFileDialog

24. Add a new method called `select_output_file` with the following code. This code will open a file browser and populate the line edit widget with the path of the file that the user chose.

```

def select_output_file(self):
    filename = QFileDialog.getSaveFileName(self.dlg, "Select output file", "", '*.txt')
    self.dlg.lineEdit.setText(filename)

```

```

C:\Users\Ujava\qgis2\python\plugins\SaveAttributes\save_attributes.py - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
save_attributes.py x
173 for action in self.actions:
174     self.iface.removePluginVectorMenu(
175         self.tr(u'&Save Attributes'),
176         action)
177     self.iface.removeToolBarIcon(action)
178     # remove the toolbar
179     del self.toolbar
180
181 def select_output_file(self):
182     filename = QFileDialog.getSaveFileName(self.dlg, "Select output file ", "", '*.txt')
183     self.dlg.lineEdit.setText(filename)
184
185 def run(self):
186     """Run method that performs all the real work"""
187     layers = self.iface.legendInterface().layers()
188     layer_list = []
189     for layer in layers:
190         layer_list.append(layer.name())
191
192     self.dlg.comboBox.addItem(layer_list)
193
194     # show the dialog
195     self.dlg.show()
196     # Run the dialog event loop
197     result = self.dlg.exec_()
198     # See if OK was pressed
199     if result:

```

Python file    length: 7151    lines: 203    Ln: 183    Col: 44    Sel: 0 | 0    UNIX    UTF-8 w/o BOM    INS

Figure 133 : Ajout de code Python correspondant

25. Now we need to add code so that when the ... button is clicked, select\_output\_file method is called. Scroll up to the \_\_init\_\_ method and add the following lines at the bottom. This code will clear the previously loaded text (if any) in the line edit widget and also connect the select\_output\_file method to the clicked signal of the push button widget.

```

self.dlg.lineEdit.clear()
self.dlg.pushButton.clicked.connect(self.select_output_file)

```

```

C:\Users\Ujaval\qgis2\python\plugins\SaveAttributes\save_attributes.py - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
save_attributes.py x
58     if qVersion() > '4.3.3':
59         QApplication.installTranslator(self.translator)
60
61     # Create the dialog (after translation) and keep reference
62     self.dlg = SaveAttributesDialog()
63
64     # Declare instance attributes
65     self.actions = []
66     self.menu = self.tr(u'&Save Attributes')
67     # TODO: We are going to let the user set this up in a future iteration
68     self.toolbar = self.iface.addToolBar(u'SaveAttributes')
69     self.toolbar.setObjectName(u'SaveAttributes')
70
71     self.dlg.lineEdit.clear()
72     self.dlg.pushButton.clicked.connect(self.select_output_file)
73
74
75 # noinspection PyMethodMayBeStatic
76 def tr(self, message):
77     """Get the translation for a string using Qt translation API.
78
79     We implement this ourselves since we do not inherit QObject.
80
81     :param message: String for translation.
82     :type message: str, QString
83
84     :returns: Translated version of message

```

Figure 134 : le code Python à ajouter

- Back in QGIS, reload the plugin and open the *Save Attributes` dialog*. If all went fine, you will be able to click the . . . button and select an output text file from your disk.

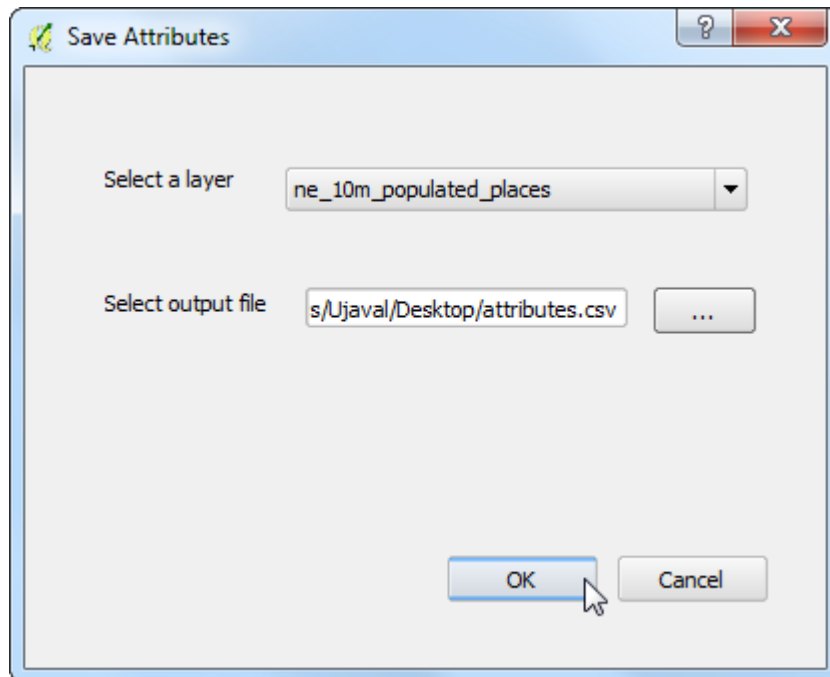


Figure 135 : résultat du code source

27. When you click *OK* on the plugin dialog, nothing happens. That is because we have not added the logic to pull attribute information from the layer and write it to the text file. We now have all the pieces in place to do just that. Find the place in the run method where it says *pass*. Replace it with the code below. The explanation for this code can be found in *Getting Started With Python Programming*.

```
filename = self.dlg.lineEdit.text()
output_file = open(filename, 'w')

selectedLayerIndex = self.dlg.comboBox.currentIndex()
selectedLayer = layers[selectedLayerIndex]
fields = selectedLayer.pendingFields()
fieldnames = [field.name() for field in fields]

for f in selectedLayer.getFeatures():
    line = ','.join(unicode(f[x]) for x in fieldnames) + '\n'
    unicode_line = line.encode('utf-8')
    output_file.write(unicode_line)
output_file.close()
```

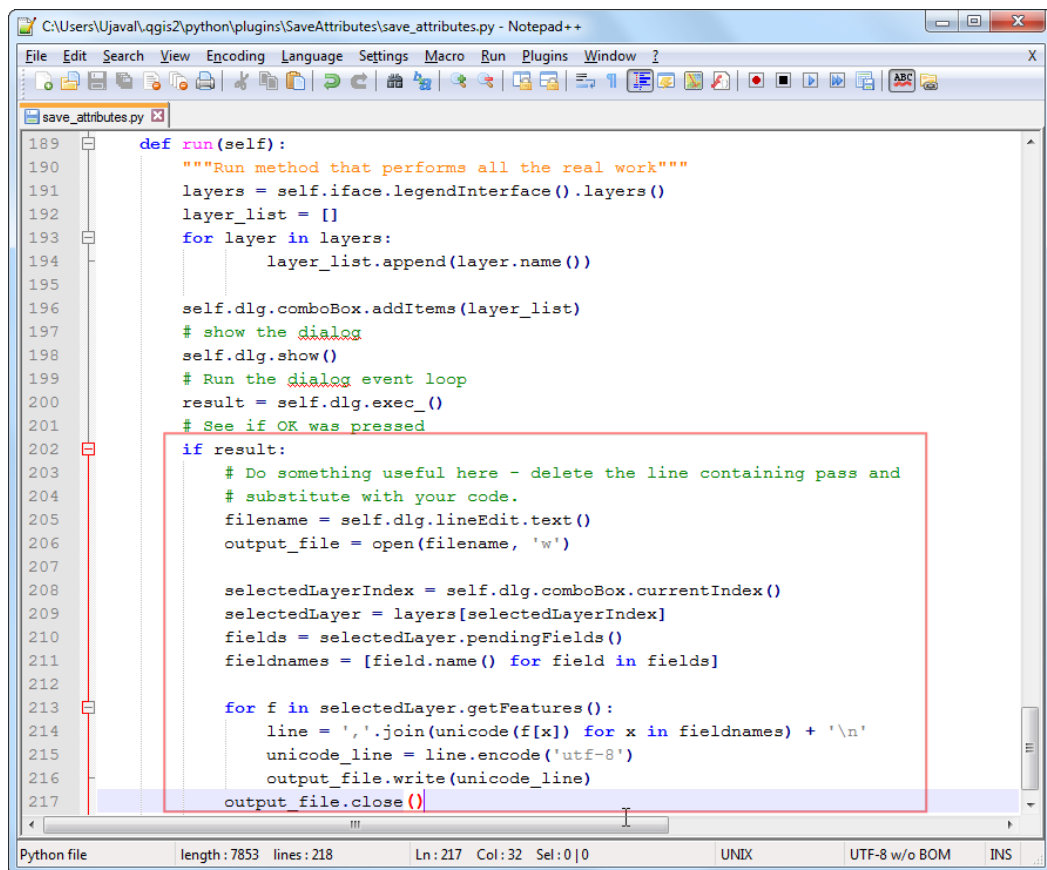


Figure 136 : partie du code Python

28. Now our plugin is ready. Reload the plugin and try it out. You will find that the output text file you chose will have the attributes from the vector layer. You can zip the plugin directory and share it with your users. They can unzip the contents to their plugin directory and try out your plugin. If this was a real plugin, you would upload it to the QGIS Plugin Repository so that all QGIS users will be able to find and download your plugin.



# Références

Allen, James F. 1983. « Maintaining knowledge about temporal intervals », *In Communications of the ACM*, vol. 26, n° 11 : 832-843.

Alter, Steven. 1980. *Decision Support System: Current Practices and Continuing Challenges*. Massachusetts: Addison-Wesley Publishing Co.

Armstrong, Marc P et Paul J Densham. 1990. « Database organization strategies for spatial decision support systems. », *Journal of Geographical Information Systems*, vol. 4, n° 1 : 3-20.

Bédard, Yvan. 2008. *OLAP et SOLAP : notions avancées des bases de données SIG*, <http://www.scribd.com/doc/6964842/OLAP-et-SOLAP-completavec-explication-ppt-univ-laval>.

Bédard, Yvan. 2004. *Amélioration des capacités décisionnelles des SIG par l'ajout d'un module SOLAP.*, Marseille Université de Provence, Centre de Mathématiques et Informatique, LSIS, Marseille, 8 avril.

Bédard, Yvan, François Létourneau et Bernard Moulin. 1998. *Perspectives d'utilisation du concept d'entrepôt de données pour les géorépertoires sur Internet.*, [http://sirs.scg.ulaval.ca/Yvanbedard/article\\_nonprotege/224.pdf](http://sirs.scg.ulaval.ca/Yvanbedard/article_nonprotege/224.pdf).

Bédard, Yvan, T.H. Merrett et Jiawei Han. 2001. « Fundamentals of Spatial Data Warehousing for Geographic Knowledge Discovery », in MILLER Harvey J. et HAN Jiawei. *Geographic Data Mining and Knowledge Discovery*, Taylor & Francis., Londres U.K.: 53-76.

Bédard, Yvan, Marie-Josée Proulx et Sonia Rivest. 2005. « Enrichissement du OLAP pour l'analyse géographique: exemples de réalisation et différentes possibilités technologiques. », Université de Lyon-II, 10 juin 2005.

Bédard, Yvan, Sonia Rivest et Marie-Josée Proulx. 2007. in (Bédard et al., 2007) BÉDARD Yvan, RIVEST Sonia et PROULX Marie-Josée. *Spatial On-Line Analytical Processing (SOLAP): Concepts, Architectures and Solutions from a Geomatics Engineering Perspective*. In : Wrembel Robert et Koncilia Christian. *Data Warehouses and OLAP : Concepts, Architectures and Solutions*. Londres, UK : IRM Press (Idea Group), 2007, 298-319 p.

Bédard, Yvan, Sonia Rivest et Marie-Josée Proulx. 2007. « Spatial On-Line Analytical Processing (SOLAP): Concepts, Architectures and Solutions from a Geomatics Engineering Perspective. In : Londres, UK ;, 298-319 p. », in *Data Warehouses and OLAP : Concepts, Architectures and Solutions.*, IRM Press (Idea Group), Londres U.K. Wrembel Robert et Koncilia Christian.: 298-319.

Bimonte, Sandro. 2007. « Intégration de l'information géographique dans les entrepôts de données et l'analyse en ligne: de la modélisation à la visualisation »,.

Bimonte, Sandro, Anne Tchounikine et Maryvonne Miquel. 2005. « Towards a spatial multidimensional model », Bremen, Germany New York USA : ACM press.

Blaschka, Markus, Carsten Sapia, Gabriele Höfling et Barbara Dinter. 1998. « Finding your way through multidimensional data models », Vienne, Autriche. Washington, DC, USA : IEEE Computer Society.

Bouadi, Tassadit. 2014. « Analyse multidimensionnelle interactive de résultats de simulation. Aide à la décision dans le domaine de l'agro-écologie », Rennes.

Boulil, Kamal. 2014. « Une approche automatisée basée sur des contraintes d'intégrité définies en UML et OCL pour la vérification de la cohérence logique dans les systèmes SOLAP: Applications dans le domaine agri-environnemental », Université Blaise Pascal-Clermont-Ferrand II.

Burrough, P.A. 1986. *Principles of Geographical Information Systems for Land Resources Assessment* ., Oxford, Oxford University Press.

Caron, Pierre-Yves. 1998. *Étude du potentiel de OLAP pour supporter l'analyse spatio-temporelle*, université de Laval, Canada.

Chaudhuri, S. et U. Dayal. 1997. « An Overview of Data Warehousing and OLAP Technology », *ACM SIGMOD Record*, vol. 26, n° 1 : 65-74.

Clarke, K.C. 1990. *Analytical and computer cartography* .Prentice-Hall.

Codd, E.F., S.B. Codd et C.T. Salley. 1993. *Providing OLAP to User-Analysts: An IT Mandate*, [http://www.minet.uni-jena.de/dbis/lehre/ss2005/sem\\_dwh/lit/Cod93.pdf](http://www.minet.uni-jena.de/dbis/lehre/ss2005/sem_dwh/lit/Cod93.pdf).

Cowen, D.J. 1988. « GIS versus CAD versus DBMS : What are the differences? », *Photogrammetric Engineering and remote sensing*, vol. 54, n° 11 : 1551-1555.

Damiani, M.L. et S. Spaccapietra. 2006. « Spatial Data Warehouse Modelling. Processing and Managing Complex Data for Decision Support . », in *In: Darmont Jérôme et Boussaid Omar. Processing and Managing Complex data for Decision Support.*, PA, USAIDEA Group Publishing: 1-27.

Denègre, Jean et François Salgé. 1996. *Les systèmes d'information géographique*, Puf.

Didier, Michel. 1990. *Utilité et valeur de l'information géographique*, Economica.

Dinimant, A. 2009. *Qu'est-ce que la modélisation multidimensionnelle ?*, <http://blog.developpez.com/bi/p7422/informations/modelisation-multidimensionnelle/>.

Fidalgo, Robson N, Valeria Times, Joel Silva et Fernando F Souza. 2004. « GeoDWFrame: A Framework for Guiding the Design of Geographical Dimensional Schemas », in *In: Kambayashi yahiko, mohania mukesh k. et wöß wolfram*, Saragosse, Espagne Berlin Heidelberg : Springer, : 26-37.

Franklin, Carl. 1992. « An Introduction to Geographic Information Systems: Linking Maps to databases. », *Database*, vol. 15, n° 2 : 13-21.

Glorio, Octavo et Juan Trujillo. 2008. « An MDA Approach for the Development of Spatial Data Warehouses », in *In : Data Warehousing and Knowledge Discovery* édité par Il-Yeol Song, Johann Eder et Tho Manh Nguyen.: 23-32.

Golfarelli, M., D. Maio et S Rizzi. 1998. « Conceptual Design of Data Warehouses from E/R Schema », Hawaii IEEE Computer Society: 334.

Goodchild, M.F. 1988. « A spatial analytical perspective on GIS », *International Journal of Geographic Information Systems*, vol. 1: 327-334.

Goodchild, Michael Frank et Paul J Densham. 1990. « Spatial Decision Support Systems: Scientific Report for the Specialist Meeting, TR-90-5. », Santa Barbara California.

Han, Jiawei, Nebolsa Stefanovic et Krzysztof Koperski. 1998. « Selective materialization : An Efficient Method for Spatial Data Cube Construction », in *Research and Development in Knowledge Discovery and Data Mining* .: 144-158.

Hewlett Packard. 1993. « Spatial data processing with computer systems . - Google Scholar ».

Hüsemann, Bodo, Jens Lechtenböcker et Gottfried Vossen. 2000. « Conceptual data warehouse modeling », StockholmCEUR-WS.org,.

Imnon, W.H. 1994. *Building the Data Warehouse*, New YorkWiley.

jensen, Christian, Augustas Kligys, Torben B. Pedersen et Igor Timko. 2004. « Multidimensional data modeling for location-based services », *International Journal on Very Large Data Bases*, vol. 13, n° 1 : 1-21.

Joliveau, Thierry. 1996. « , Gérer l'environnement avec des SIG. Mais qu'est - ce qu'un SIG ? », *Revue de Géographie de Lyon*, n° 71 : 101-110.

Keenan, Peter. 1996. « Using a GIS as a DSS Generator », in *Perspectives on Decision Support System.*, GrèceDarzentas John: 33-40.

Kimball, Ralph. 1996. *The data warehouse toolkit: Practical Techniques for Building Dimensional*, John Wiley and Sons.

Lardon, Sylvie, Thérèse Libourel et Jean-Paul Cheylan. 1999. « Concevoir la dynamique des entités spatio-temporelles. vol. 9, no. 1, pages 67–99, 1999 », *Revue Internationale de Géomatique*, vol. 9, n° 1 : 67-99.

Longley, P.A., M.F. Goodchild, D. Maguire et Rhind. 2005. *Geographic information systems and science*. Wiley, 2005.

Malinowski, Elzbieta et Esteban Zimányi. 2008. *Advanced Data Warehouse Design: From Conventional to Spatial and Temporal Applications, Data- Centric Systems and Applications*, Springer., Berlin Heidelberg.

Malinowski, Elzbieta et Esteban Zimányi. 2005. « Spatial Hierarchies and Topological Relationships in Spatial Multidimer model », Sunderland, UK.Springer Berlin He idelberg.

Malinowski, Elzbieta et Esteban Zimányi. 2004. « Representing spatiality in a conceptual multidimensional model », Washington, DC, USANew York, USA : ACM Press.

Marchand, Pierre, Alexandre Brisebois, Yvan Bédard et Geoffrey Edwards. 2004. « Implementation and evaluation of a hypercube-based method for spatiotemporal exploration and analysis », *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 59, n° 1–2 : 6-20.

Maryvonne, Miquel, Bédard Yvan, Brisebois Alexandre, Pouliot Jacynthe, Marchand Pierre et Brodeur Jean. s. d. *MODELING MULTIDIMENSIONAL SPATIO-TEMPORAL DATA WAREHOUSES IN A CONTEXT OF EVOLVING SPECIFICATIONS - 454.pdf*,  
<http://www.isprs.org/proceedings/XXXIV/part4/pdfpapers/454.pdf>: consulté le 15 septembre 2015.

Mathian, Hélène et Marie Piron. 2001. « Echelles géographiques et méthodes statistiques multidimensionnelles », in Léna Sanders, editeur, *Modèles en analyse spatiale*, Hermès.

Meier, A. 2006. *Introduction pratique aux bases de données relationnelles.*, Seconde édition., FranceSpinger Editions.

Noirault, Claire. 2006. *Business Intelligence avec Oracle 10g: ETL, Data warehouse, Data mining, rapports...*, <http://books.google.fr/books?id=2StX-h64ShgC&lpg=PT9&dq=Entrep%C3%B4ts%20de%20donn%C3%A9es&pg=PT9#>.

Paegelow, Martin. 2004. « Géomatique et géographie de l'environnement. De l'analyse spatiale à la modélisation prospective », Université Toulouse le Mirail-Toulouse II.

Park, B.-K., H Han et I.-Y. Song. 2005. « A Multidimensional Analysis Framework for XML Warehouses », in *Data Warehousing and Knowledge Discovery.*, HeidelbergA. Tjoa and J. Trujillo, Springer Berlin: 32-42.

Pedersen, Torben B., Christian jensen et C.E. Dyreson. 2001. « A foundation for capturing and querying complex multidimensional data », *Inf. Syst.*, vol. 25, n° 5 : 383-423.

Pedersen, Torben B. et N. Tryfona. 2001. « Pre-aggregation in Spatial Data Warehouses »Springer-Verlag.

Pestana, Gabriel, Miguel Silva et Yvan Bédart. 2005. « Spatial OLAP Modeling: An Overview Based on Spatial Objects Changing over Time », Iles Maurice.

Pinet, F et M Schneider. 2010. « Precise design of environmental data warehouses », *Operational Research*, vol. 10, n° 3 : 349-369.

Pinet, François et Michel Schneider. 2010. « Precise Design of Environmental data Warehouses », *Operational Research*, vol. 10, n° 3 : 349-369.

plumejeaud, Christine. 2011. « Modèles et méthodes pour l'information spatio-temporelle évolutive », Université Paris.

Pornon, Henri. 2007. « Bilan et perspectives de 20 années de Géomatique », *Géomatique expert*, vol. 57: 36-46.

Pumain, Denise et Thérèse Saint-Julien. 1997. *L'analyse spatiale, Localisations dans l'espace.*, Armand Colin., Paris.

Rafanelli, Maurizio. 2003. « Operators for Multidimensional Aggregate Data », in *Multidimensional Databases: Problems and Solutions: Problems and Solutions*, Hershey, PA, USA.IGI Publishing: 116-165.

Ravat, F., Olivier Teste et R. Tournier. 2007. « OLAP aggregation function for textual data warehouse. Funchal, Madeira - Portugal », Funchal, Madeira, Portugal.

Ravat, F., Olivier Teste, R. Tournier et G. Zurfluh. 2008. « Top\_Keyword: An Aggregation Function for Textual Document OLAP. », in *Data Warehousing and Knowledge Discovery*, Eidelbergl.-Y. Song, J. Eder and T. Nguyen, Springer Berlin: 55-64.

Rivest, Sonia, Yvan Bédard et Pierre Marchand. 2001. « Toward better support for spatial decision making: defining the characteristics of spatial on-line analytical processing », *Geomatica. Journal of the Canadian Institute of Gemomatics*, vol. 55, n° 4 : 539–555.

Rivest, Sonia, Yvan Bédard, Marie-Josée Proulx, Martin Nadeaum, Frédéric Hubert et Julien Pastor. 2005. « SOLAP technology: Merging business Intelligence with Geospatial Technology for Interactive Spatio-Temporal Exploration and Analysis of Data. », *Elsevier: Journal of Photogrammetry and Remote Sensing*, vol. 60, n° 1 : 17-33.

Rivest, Sonia, Yvan Bédard, Marie-Josée Proulx et Martin Nadeaum. 2003. « SOLAP: a new type of user interface to support spatio-temporal multidimensional data exploration and analysis », Quebec, Canada.

Ruas, Anne. 1999. « Modèle de généralisation de données géographiques à base de contraintes et d'autonomie », Thèse de doctorat en Sciences de l'information géographique, Marne la Vallée.

Ruiz, C.V. et V.C. Times. 2009. « A Taxonomy of SOLAP Operators. »

Schneuwly, Dominique et Regis Caloz. 2013. « Les concepts spatiaux fondamentaux ».

Silva, Joel, Valeria Times, A.C. Salgado, C. Souza, Robson N Fidalgo et A.G. Oliveira. 2008. « A set of aggregation functions for spatial measures », Napa Valley, California, USAACM.

Stefanovic, Nebolsa, Jiawei Han et Krzysztof Koperski. 2000. « Object-Based Selective Materialization for Efficient Implementation of Spatial Data Cubes », *IEEE Transactions on Knowledge and Data Engineering*, vol. 12, n° 6 : 938-958.

Stolte, Chris, Diane Tang et Pat Hanrahan. 2003. « Multiscale visualization using data cubes », *IEEE Transaction on Visualization and Computer Graphics*, vol. 9, n° 2 : 176-187.

Tomlinson, R.F. 1987. « Current and potential uses of geographical information systems. », *The North American experience. International Journal of Geographic Information Systems*, vol. 1, n° 3 : 203-219.

Tranchant, Michaël. 2011. *Capacités des outils SOLAP en termes de requêtes spatiales, temporelles et spatio-temporelles*, GrenobleARTS ET METIERS CENTRE REGIONAL RHÔNE-ALPES CENTRE D'ENSEIGNEMENT DE GRENOBLE UE ENG111 –.

Winter, Richard. 1998. « Databases: back in the OLAP game. », *Intelligent Enterprise Magazine*, vol. 1: 60-64.

Yazidi Alaoui, Otmane, Salaheddine Hamdoune et Hassan Zili. 2016. « La Dynamique d'un réseau Télécom », *Revue Geo Science Publications*, vol. 4, n° 10 : 53-69.

Yazidi Alaoui, Otmane, Salaheddine Hamdoune et Hassan Zili. 2014. « Elaboration d'une solution-SIG adaptée à la gestion des réseaux d'eau du Maroc », *Revue internationale de géomatique*, vol. 24, n° 4 : 501-523.

S. d. *Version 1.0 de la définition de l'OpenSource de Bruce Perens*, <http://www.velic.com/publications/tribunelibre/fr-appb.html>: consulté le 10 septembre 2015a.

S. d. *Pros and Cons of Open Source Business Intelligence*, <https://www.yurbi.com/blog/pros-and-cons-of-open-source-business-intelligence/>: consulté le 10 septembre 2015b.

S. d. *Plateforme logicielle d'intégration Open Studio | Talend*, <https://fr.talend.com/products/talend-open-studio>: consulté le 14 septembre 2015c.

S. d. *Pentaho Business Analytics*, <http://www.pentaho.fr/explore/>: consulté le 14 septembre 2015d.