



BOOCs
EPFL

Systèmes d'Information
Géographique – Partie 1
Stéphane Joost, Marc Soutter,
Fernand Kouamé et Amadou Sall

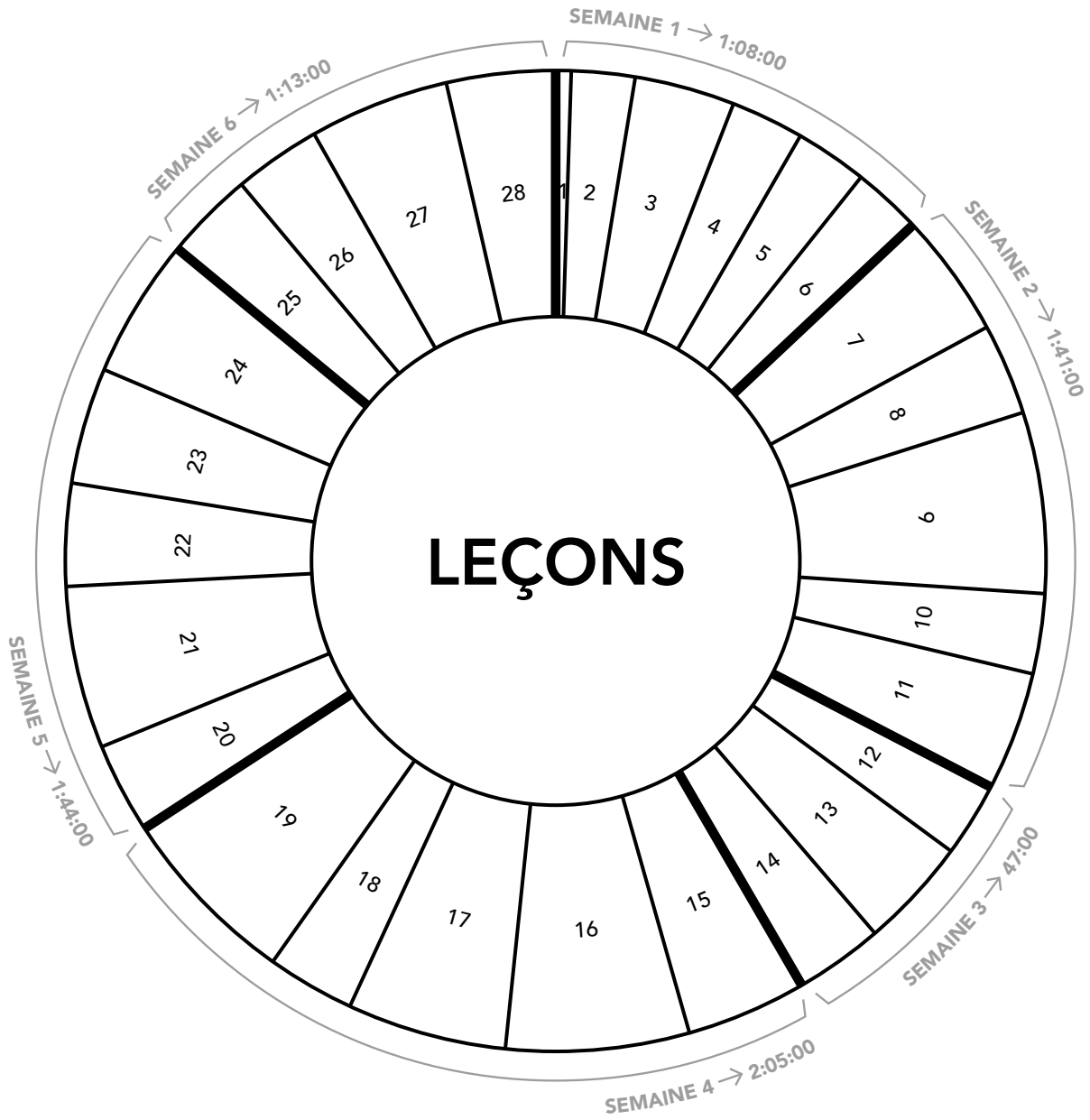


SYSTÈMES D'INFORMATION GÉOGRAPHIQUE

Partie I



**Stéphane Joost, Marc Soutter,
Fernand Kouamé et Amadou Sall**





CONTENU

SEMAINE 1 : NUMÉRISATION - MODÉLISATION DU TERRITOIRE: LES ÉLÉMENTS SPATIAUX ET LEURS CARACTÉRISTIQUES

1. Présentation du cours	4
2. Modélisation du territoire	5
3. Coordonnées et systèmes de projection	8
4. Démo – Introduction à QGIS	11
5. Relations spatiales et topologie	13
6. Mode vecteur et mode image	15

SEMAINE 2 : NUMÉRISATION - SAISIE ET DOCUMENTATION DES GÉODONNÉES

7. Métadonnées	17
8. Sources de géodonnées	18
9. Étude de cas – SIG et planification participative	20
10. Géoréférencement d'images	23
11. Digitalisation d'objets vectoriels	25

SEMAINE 3 : NUMÉRISATION - AUTOMATISATION DE LA SAISIE ET UTILISATION DE GÉODONNÉES EXISTANTES

12. Vectorisation automatique	27
13. Sources de géodonnées	28
14. Étude de cas – Biodiversité Sénégal-Mauritanie	30

SEMAINE 4 : STOCKAGE - ORGANISATION ET STRUCTURATION DES GÉODONNÉES

15. Stockage de l'information géographique	32
16. Bases de données relationnelles	35
17. Modélisation des données	39
18. Création des bases de données test	45
19. Étude de cas – Développement et aménagement urbain à Touba	47

SEMAINE 5 : STOCKAGE - GESTION DE DONNÉES AVEC LE LANGAGE SQL

20. Requêtes et langage SQL	55
21. Requêtes conditionnelles	58
22. Requêtes d'agrégation et tri du résultat	63
23. Fusion de requêtes, requêtes emboîtées	65
24. DDL, DML, Vues	67

SEMAINE 6 : STOCKAGE - SQL SPATIAL ET BASES DE DONNÉES NOSQL

25. Requêtes spatiales géométriques	71
26. Requêtes spatiales topologiques	74
27. Bases de données NoSQL	77
28. Étude de cas – Les feux de brousse au Sahel	81

1. PRÉSENTATION DU COURS

Les systèmes d'information géographique (SIG) permettent de stocker, gérer, afficher et analyser des données spatiales. Ils se révèlent indispensables dans de nombreux domaines comme la planification urbaine, l'analyse spatiale ou encore le suivi de l'évolution d'une thématique localisée.

Ce cours donne un vaste aperçu des différentes phases du cycle de vie des SIG. Il décrit comment l'information spatiale est gérée, à commencer par son acquisition et son stockage dans une base de données puis par sa visualisation.

OUVRAGE DE RÉFÉRENCE EPFL



Les systèmes d'information géographique (SIG) sont des outils d'observation des territoires et d'analyse spatiale puissants. Combinant les informations géographiques et statistiques, ils permettent un suivi cartographié et quantifié des dynamiques territoriales. Leur potentiel en matière d'aide à la décision, notamment dans les domaines de l'urbanisme, de l'aménagement et du développement des territoires, reste toutefois sous-exploité. En la matière, concepteurs, analystes et décideurs sont des acteurs qui s'ignorent trop souvent et qu'il s'agit de mieux réunir. Cet ouvrage les y invite autour du questionnement suivant: comment les SIG et l'analyse spatiale contribuent-ils à la prise de décision en faveur d'un développement territorial durable ?

Différents champs thématiques se croisent: progrès des SIG en tant que technologie, multiplication et diversification des sources de données géographiques et statistiques, méthodes de modélisation et d'analyse spatiale en constante évolution et, dans le même temps, apparition de besoins toujours nouveaux en matière de gouvernance territoriale. Cet ouvrage fait le pari et pose l'hypothèse que, pour servir réellement d'aide à la décision, une mise en perspective conjointe des dimensions techniques, analytiques et décisionnelles des SIG est urgente. Une meilleure compréhension des interdépendances entre ces dimensions est précisément le fil rouge poursuivi.

2. MODÉLISATION DU TERRITOIRE

Toute tentative de description de l'espace géographique nécessite une étape de modélisation, c'est-à-dire de représentation d'une portion réduite de l'information géographique. Cette étape est une étape préliminaire et indispensable à toute forme d'analyse et de représentation spatiale.

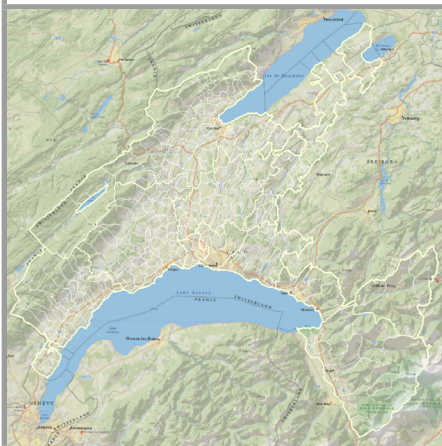


FIGURE 1

4:41 10:00

Morcellement du territoire par les frontières administratives du canton de Vaud.

ESPACE GÉOGRAPHIQUE ET TERRITOIRE

L'**espace géographique** comprend aussi bien les *composantes physiques naturelles* (montagnes, lacs, forêts) que les *infrastructures anthropisées* (zones bâties, routes d'accès). Le géographe Brunet définit l'espace géographique comme «l'étendue terrestre utilisée et aménagée par les sociétés en vue de leur reproduction au sens large, soit non seulement pour se nourrir et s'abriter, mais dans toute la complexité des actes sociaux. Il comprend l'ensemble des lieux et de leurs relations. Il est de ce fait à la fois un système de relations et un produit organisé résultant des interactions entre la nature et les sociétés. Il a des lois et des règles d'organisation et de différenciation universelles, mais exprimées différemment selon les systèmes. Au premier rang d'entre elles se trouvent la gravitation et, plus généralement tous les effets de la distance, de l'agrégation et de l'espacement. On peut y reconnaître cinq usages fondamentaux: l'appropriation, l'exploitation (ou la mise en valeur), l'habitation, l'échange (ou communication) et la gestion».

La notion de **territoire** peut être vue comme une *partie délimitée de l'espace géographique*. La figure 1 montre le canton de Vaud et ses communes.

MODÉLISATION DU TERRITOIRE

La **modélisation du territoire** est le processus par lequel s'effectue la transition d'une *réalité géographique* visible et tridimensionnelle à une *représentation schématique* et planaire, comme une carte. La représentation du territoire (contenu et sémiologie graphique) diffère en fonction de la *discipline* et de l'*échelle de description*. Pour cette raison, on parle de *réduction contrôlée de la réalité*.

CARTOGRAPHIE

La **cartographie** est la première forme de modélisation de l'espace géographique, comme en témoigne la carte d'Eratosthène (figure 2), datant du troisième siècle avant notre ère. Elle sert à représenter un lieu, des objets dans leur environnement ainsi que la distribution et la dynamique de *phénomènes spatiaux*.

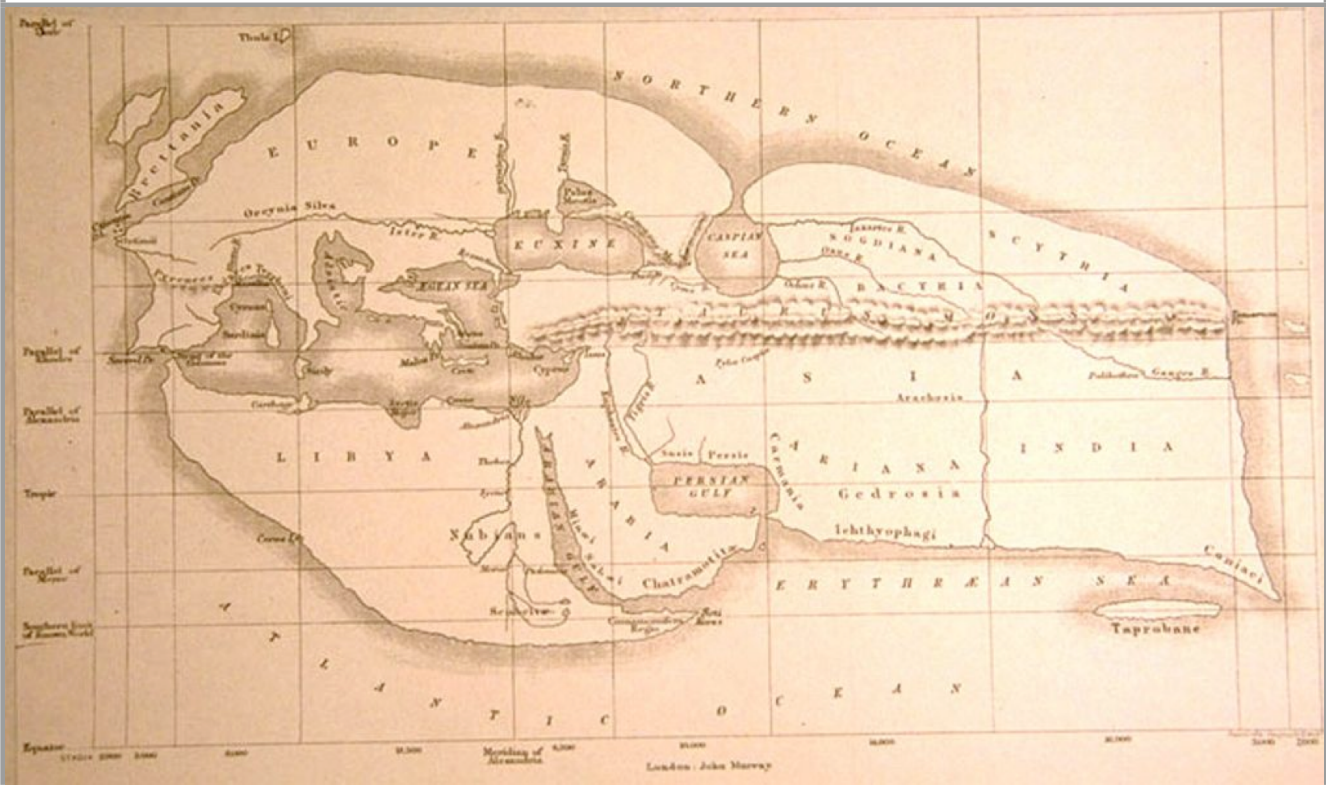


FIGURE 2

6:45

10:00

Carte d'Eratosthène, III^e siècle av. J.-C.

PHÉNOMÈNES DISCRETS ET PHÉNOMÈNES CONTINUS

Le modèle du territoire peut être la représentation d'objets spatiaux, la distribution de phénomènes spatiaux ou l'illustration de processus spatiaux et leur dynamique dans le temps. Le **phénomène spatial discret** est contenu dans ses limites et constitué d'objets homogènes dans toutes leurs parties (zones à bâtir, surfaces boisées, haies, arbres isolés). On parle d'*approche objet*. Au contraire, le **phénomène spatial continu** n'est pas circonscrit et ses propriétés varient dans l'espace (maximum glaciaire, distribution des pentes, profondeur des sols). Il peut être représenté par des *lignes d'isovaleurs* ou par une *grille à maille régulière*. On parle d'*approche image*. La figure 3 est un exemple de représentation de phénomène discret et la figure 4 d'un phénomène continu.

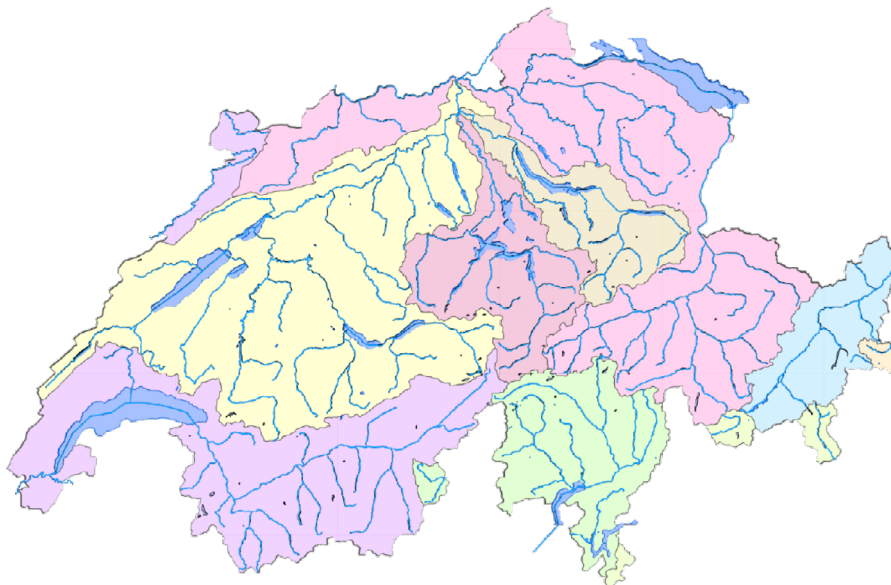


FIGURE 3

8:33

10:00

Lacs, cours d'eau et bassins fluviaux en Suisse.

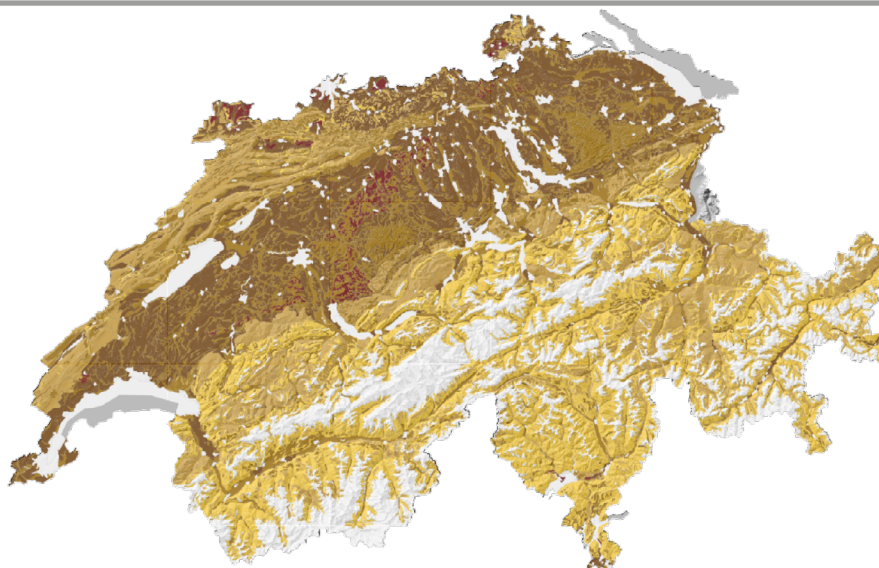


FIGURE 4

9:05

10:00

Profondeur des sols suisses.

3. COORDONNÉES ET SYSTÈMES DE PROJECTION

Les **systemes de coordonnées géographiques** permettent de situer les objets à la surface de la Terre tandis que les **systemes de projection** permettent de les projeter sur une surface plane.

Pour intégrer la **géométrie** des objets dans la représentation du territoire, il est nécessaire de connaître leurs **coordonnées**, ainsi que les **transformations** qu'elles subissent de la réalité au modèle. C'est la raison pour laquelle on s'intéresse à la métrique de la description des éléments du territoire ainsi qu'aux transformations pour passer d'un système de coordonnées à un autre, en particulier, pour passer de l'espace géographique à la représentation cartographique.

ÉCHELLES D'OBSERVATION ET DE REPRÉSENTATION

L'**échelle** est le rapport entre la distance mesurée sur le modèle (la carte) et la distance réelle. Elle dépend de l'objet de la représentation et des contraintes techniques, graphiques ou physiologiques. Le choix de l'échelle repose sur la question du **degré de généralisation** souhaité, lui-même dépendant du phénomène observé. Les géométries pourront par exemple être très précises et complexes pour de grandes échelles, mais elles doivent être simplifiées à de plus petites échelles (représentation des immeubles d'une ville, tracé d'un cours d'eau, réseau routier).

Dans les faits, il s'avère que sur la carte, un carré de moins d'un millimètre de côté est difficilement discernable. De ce fait, cette valeur s'est imposée comme une règle empirique de la représentation cartographique. Ainsi, une échelle inférieure au 1:10000 ne serait pas adaptée pour représenter un objet d'une taille d'environ 10 mètres.

COORDONNÉES ET POSITIONNEMENT

Un objet peut être localisé par la notion de **voisinage**. Par contre, cette notion est très abstraite et l'on sent bien qu'elle manque d'une métrique clairement définie. Le **système de coordonnées euclidien** possède un **système d'axes perpendiculaires** et une **métrique constante**. Il se fonde sur l'**hypothèse d'un espace plan et continu**. Il est utilisé pour référencer les objets spatiaux. Pour ce faire, il est nécessaire de projeter la surface sphérique de la terre sur un plan: il s'agit de l'étape de **projection**.

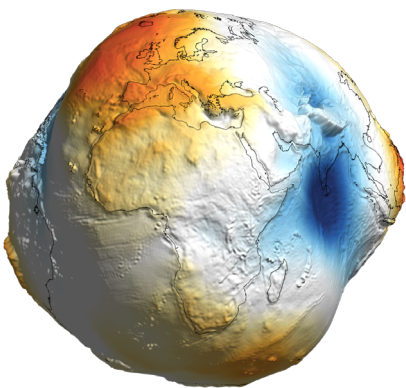


FIGURE 1

6:47

17:00

Modèle du géoïde.

SYSTÈMES DE PROJECTION

La géométrie de la terre est une sphère irrégulière appelée **géoïde** (figure 1). Il s'agit de la surface équipotentielle en gravité qui présente des variations locales (renflements et dépressions) par rapport à un ellipsoïde de révolution.

La projection s'effectue en deux étapes: l'*approximation du géoïde par un ellipsoïde de révolution* (global: WGS84 ou local: en Suisse, l'ellipsoïde de Bessel), puis la *projection des points de l'ellipsoïde de révolution sur le plan*, comme l'illustre la figure 2. Cette dernière étape implique une perte d'information, car on passe d'un système de coordonnées sphérique à trois coordonnées à un système de coordonnées euclidien à deux coordonnées. C'est la raison pour laquelle il existe trois types de projection différents, chacune privilégiant une propriété au détriment des autres.

- La projection **conforme** conserve les *directions*.
- La projection **équivalente** conserve les *surfaces*.
- La projection **équidistante** conserve les *distances*.

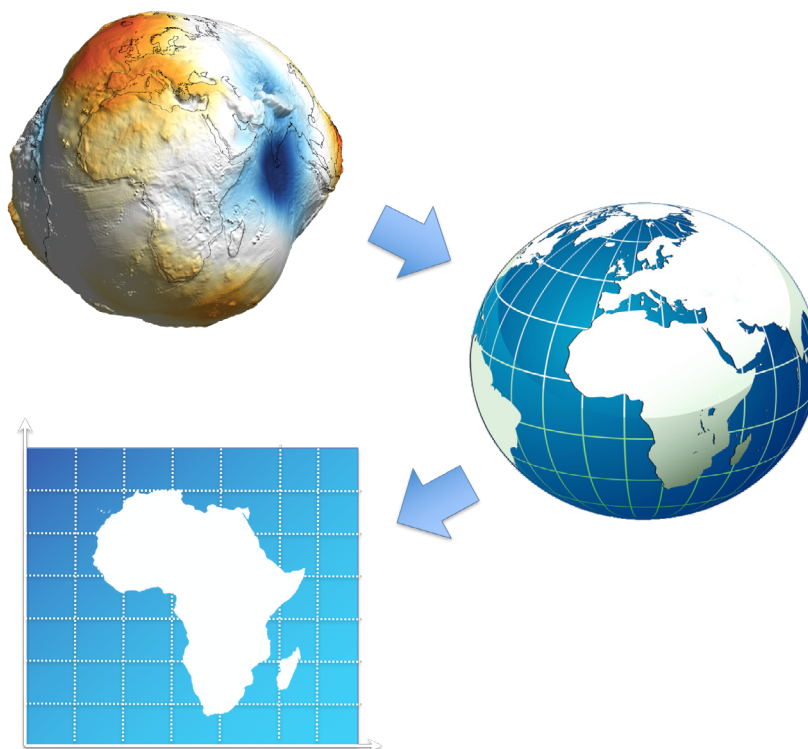


FIGURE 2

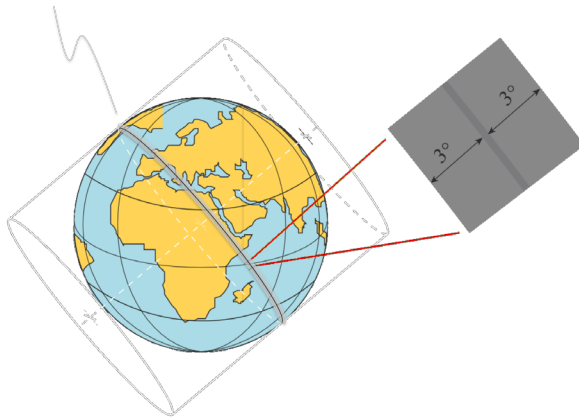
7:30

17:00

Étapes de la projection du géoïde à la carte

La projection peut être **cylindrique**, **conique** ou **azimutale**. Les erreurs de direction, de distance et de surface sont négligeables sur un angle de 3° de part et d'autre du *grand cercle de tangence* (figure 3). Celles de la projection conique se situent autour du *parallèle de tangence* (figure 4) et celles de la projection azimutale autour du *point tangent* (figure 5).

Grand cercle de tangence et zone cartographiable



Cône de projection

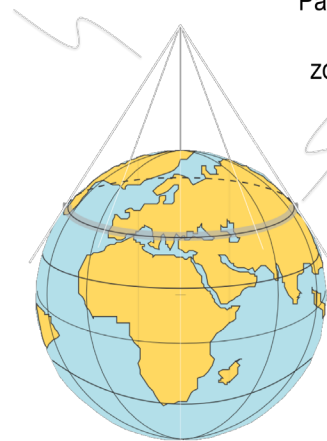
Parrallèle de tangeance
et
zone cartographiable

FIGURE 3

9:42

17:00

Projection cylindrique conforme de Mercator.

FIGURE 4

10:47

17:00

Projection conique conforme de Lambert.

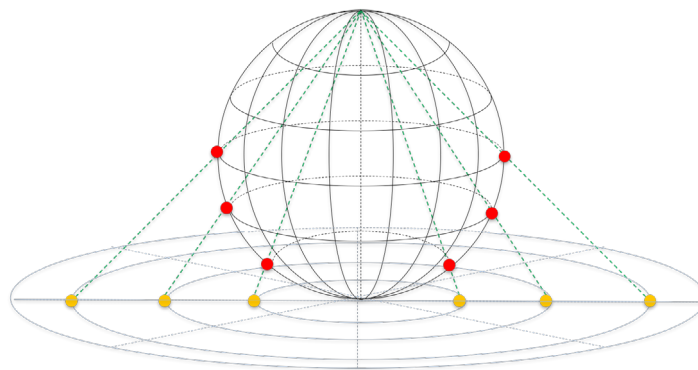


FIGURE 5

11:31

17:00

Projection azimutale stéréographique.

PROJECTION...	CYLINDRIQUE	CONIQUE	AZIMUTALE
Conforme	Mercator, UTM	Lambert	-
Équivalente	Gall-Peters	Albers	Azimutale équivalente de Lambert
Équidistante	Cylindrique équidistante	Conique équidistante	-

Résumé des différents types de projection.

Les projections azimutales diffèrent selon l'emplacement du *centre de projection* choisi. S'il est au centre de la Terre, on parle de projection gnomonique, au pôle de projection stéréographique, à l'infini de projection orthographique, etc.

CODE EPSG

Les différentes projections sont répertoriées dans le code **EPSG** (European Petroleum Survey Group). Par exemple, le système latitude/longitude de WGS84 porte le nom de EPSG 4326, et celui de la projection suisse EPSG 21781. Le site www.spatialreference.org offre la possibilité de rechercher les paramètres propres à chacun des codes EPSG, tels que la zone couverte, l'ellipsoïde de référence, etc.

4. DÉMO – INTRODUCTION À QGIS

QGIS est un logiciel SIG libre permettant la manipulation de données spatiales. L'objectif de cette leçon est de se familiariser avec QGIS. Elle montre comment importer et exporter des données, gérer leur symbologie et créer des cartes.

ÉLÉMENTS DE L'INTERFACE

La figure 1 montre l'interface de QGIS. L'interface QGIS comprend 1) un gestionnaire des couches, 2) une barre d'outils, 3) la zone de carte, 4) une barre d'état (où l'échelle, l'orientation, les coordonnées et le système de coordonnées sont spécifiés) et 5) une barre d'outils latérale. La souris permet de voyager dans la carte et la molette de zoomer sur celle-ci.

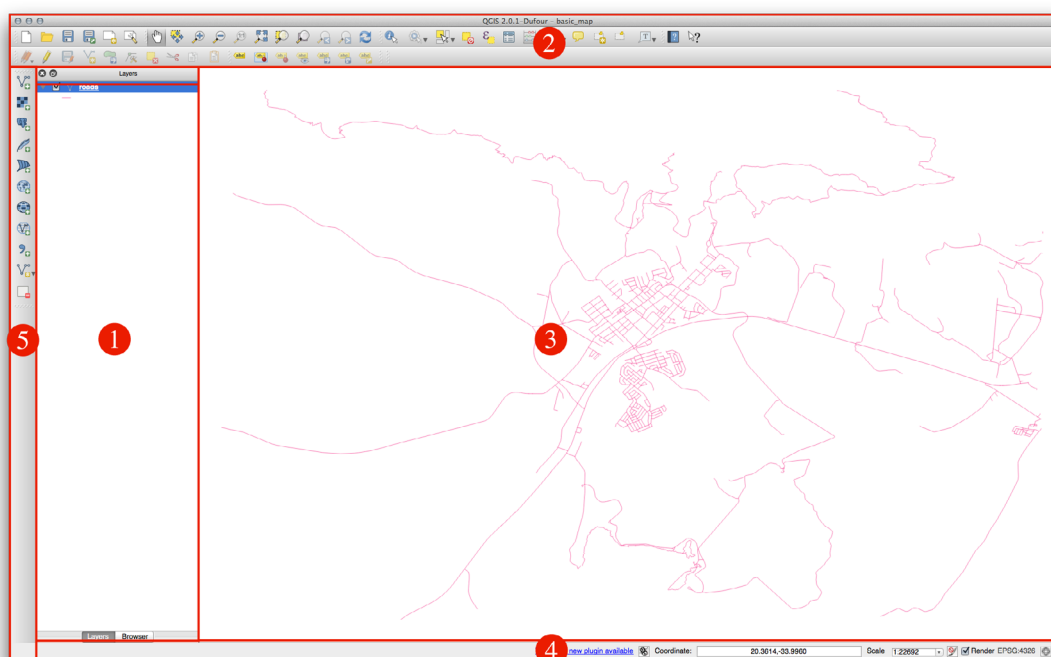


FIGURE 1

2:30

14:00

Interface de QGIS (source : https://docs.qgis.org/2.2/fr/docs/training_manual/introduction/overview.html)

L'information spatiale (les couches et leur symbologie) est enregistrée sous la forme de *projet* dans QGIS et qui peut être enregistré dans un répertoire. Il porte alors l'extension QGIS et contient les chemins d'accès aux couches. Il n'est possible de charger qu'un projet à la fois.

La barre d'outils peut contenir plusieurs sets d'outils différents qui peuvent être activés en cliquant sur le menu **Vue >> Barres d'outils** puis en les sélectionnant dans la liste. Ils peuvent également être sélectionnés en faisant un clic droit sur la barre d'outils, puis en les cochant dans la liste.

La particularité de QGIS est de fonctionner avec des *extensions* (ou **plugins**) pour compléter la liste des fonctionnalités de base du logiciel. Par exemple, le plugin *OpenLayers* permet de charger les cartes de Google, Bing, Open Street Map, etc.

GESTION DU SYSTÈME DE PROJECTION

Pour accéder aux propriétés du géoréférencement d'une couche, il faut accéder aux *propriétés de la couche* en faisant un clic droit dessus dans le panneau latéral. Puis, sous l'onglet **Général**, le système de projection utilisé est indiqué sous **Système de coordonnées de référence (SCR)**. La *projection de la carte* peut être modifiée également, en cliquant sur le bouton **EPSG: XXXXX** à droite dans le pied de page. Il faut cocher sur **Activer la projection 'à la volée'**, ce qui signifie que les couches ouvertes dans le projet vont automatiquement s'adapter au système de projection de la carte.

Il est également possible d'exporter une couche dans un format différent de la source (passer de *ESRI Shapefile* au format *GeoJSON* par exemple), en utilisant un système de projection différent. Pour ce faire, faire un clic droit sur la couche, cliquer sur **Enregistrer sous...** et sélectionner les options désirées.

GESTION DES DONNÉES

Pour importer une couche, cliquer sur le menu **Couche >> ajouter une couche** et sélectionner le type de couche à importer. Lors de l'importation d'une couche, il est nécessaire de préciser son *système de projection*. Il est aussi possible d'ajouter une nouvelle couche en cliquant sur la barre d'outils à gauche de l'écran ou en faisant simplement un glisser-déposer du fichier à importer sur la fenêtre de QGIS.

Un clic droit sur la couche permet d'accéder à sa table d'attributs ou à ses propriétés.

SYMBOLOGIE

La **symbologie** est l'ensemble des paramètres qui définissent l'apparence graphique d'une couche. Elle peut être modifiée sous l'onglet *style* dans les propriétés de la couche. On peut définir le *type de bordure* et le *remplissage* des éléments vectoriels, leur *transparence*, et tout un tas d'autres paramètres concernant le rendu graphique de la couche. Les éléments d'une couche peuvent également être représentés par des couleurs spécifiques selon la valeur d'un attribut.

On peut afficher des *étiquettes* sur les éléments en précisant la police, la taille, la couleur ainsi que d'autres paramètres pour le rendu.

La symbologie d'une couche peut être enregistrée dans un fichier de style. Enfin, ce fichier peut être chargé pour appliquer la même symbologie à une autre couche.

IMPRESSIONS

Le **composeur d'impression**, disponible dans le menu **Projet**, permet de créer des cartes. Il ouvre une page vierge sur laquelle il est possible d'insérer une carte et son échelle, une légende, du texte formaté, une image, etc. Le menu des propriétés de l'objet, à droite, permet d'effectuer le formatage de l'objet sélectionné.

Une fois la carte enregistrée, on peut y accéder via le **gestionnaire de composition**, dans le menu **Projet**. Elle peut également être exportée sous différents formats d'image et en PDF.

5. RELATIONS SPATIALES ET TOPOLOGIE

La modélisation du territoire repose sur la sélection d'objets spatiaux ainsi que leur sémiologie associée. Elle dépend également de leur géométrie, qui est définie par leur position dans un système de coordonnées.

RELATION SPATIALE

La **relation spatiale** étudie le positionnement des objets spatiaux les uns par rapport aux autres, leur **connectivité** ainsi que leurs **relations topologiques**. Les relations topologiques sont essentielles pour garantir l'intégrité d'une base de données spatiale, afin que cette dernière puisse être exploitée par des logiciels SIG.

Les **entités spatiales** sont décrites *individuellement* par leur position, mais également *collectivement* en fonction de leurs relations dans l'espace (contact, connexion, distance, atténuation). La relation spatiale (ou dépendance spatiale) décrit les propriétés partagées par au moins deux entités spatiales. Les propriétés géométriques définissent la notion de *voisinage*, de *proximité*, d'*agencement* et de *réseau*. La relation spatiale peut également s'exprimer au moyen d'un *attribut lié aux entités* (processus de diffusion ou de déplacement par exemple). Dans ce cas, on parle de *relation fonctionnelle*.

NOTION DE TOPOLOGIE

La **topologie** traite des propriétés de l'espace préservées lors de *déformations continues* (étirement, pliage). Elle décrit les relations spatiales de manière *qualitative* en se fondant sur les notions de voisinage, de proximité, de limite, d'agencement et de réseau.

Dans un SIG, la **cohérence topologique** est définie par le respect des règles de topologie. Elle est indispensable à toute forme d'analyse spatiale. Par exemple, une règle topologique consiste en ce que deux communes voisines partagent une même frontière.

RELATIONS TOPOLOGIQUES

Deux entités spatiales qui partagent une frontière commune sont dites **adjacentes** (ou contiguës). Elles sont dites *adjacentes au sens strict* si elles partagent un côté commun, ou *adjacentes au sens large* si elles partagent un sommet commun. Si les deux entités sont en contact direct alors elles sont adjacentes au *premier ordre*. Si elles sont séparées par une entité intercalée, elles sont adjacentes au *second ordre*. Ces propriétés sont illustrées à la figure 1.

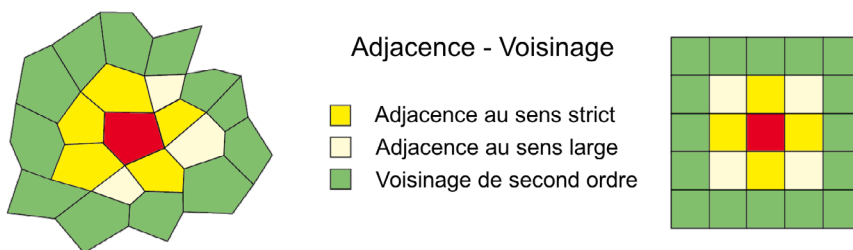


FIGURE 1

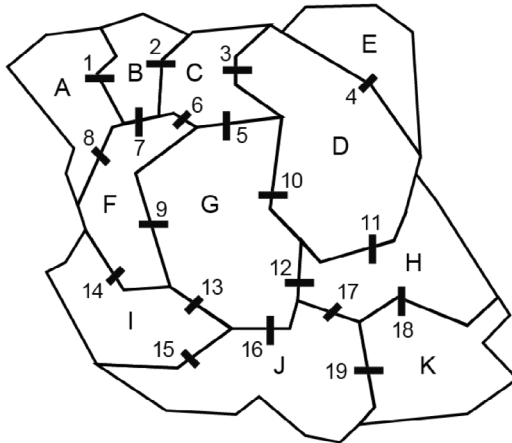
10:11

12:00

Différents types de contiguïté.

La connectivité entre entités spatiales peut être exprimée dans une matrice de connectivité, où un polygone est adjacent avec lui-même d'ordre 0, avec son voisin direct d'ordre 1, etc., comme illustré à la figure 2.

15 zones contiguës avec 19 connexions



Matrice de connectivité

	A	B	C	D	E	F	G	H	I	J	K
A	0	1	0	0	0	1	0	0	0	0	0
B	1	0	1	0	0	1	0	0	0	0	0
C	0	1	0	1	0	1	1	0	0	0	0
D	0	0	1	0	1	0	1	1	0	0	0
E	0	0	0	1	0	0	0	0	0	0	0
F	1	1	1	0	0	0	1	0	1	0	0
G	0	0	1	1	0	1	0	1	1	1	0
H	0	0	0	1	0	0	1	0	0	1	1
I	0	0	0	0	0	1	1	0	0	1	0
J	0	0	0	0	0	0	1	1	1	0	1
K	0	0	0	0	0	0	0	1	0	1	0
nb	2	3	4	4	1	5	6	4	3	4	2

FIGURE 2

10:41

12:00

Matrice de connectivité.

L'**inclusion** définit le cas où une entité spatiale est *entièrement contenue dans une autre*.

L'**intersection** définit l'*espace commun* entre deux entités spatiales.

6. MODE VECTEUR ET MODE IMAGE

L'information géographique peut être représentée soit par le *mode vecteur* pour les *phénomènes discrets*, soit par le *mode image* pour les *phénomènes continus*. Elle est également définie par les notions de *positionnement* et de *topologie*. Phénomènes discrets et continus sont liés aux notions d'*unités d'observation* régulières et irrégulières.

UNITÉ D'OBSERVATION

On définit l'*unité d'observation* comme un élément constitutif élémentaire et indivisible de l'espace géographique. Elle est entièrement définie par sa *géométrie* et par ses *propriétés thématiques* qui sont considérées comme *homogènes sur l'ensemble de l'unité*. La superposition de deux couches thématiques différentes entraîne la démultiplication des unités d'observation et la recherche du plus petit dénominateur commun des deux couches. Par exemple, si l'on considère une route, définie par le volume de trafic, la vitesse autorisée et la largeur des voies, la combinaison de ces trois thématiques conduit à un morcellement de la route, comme l'illustre la figure 1.

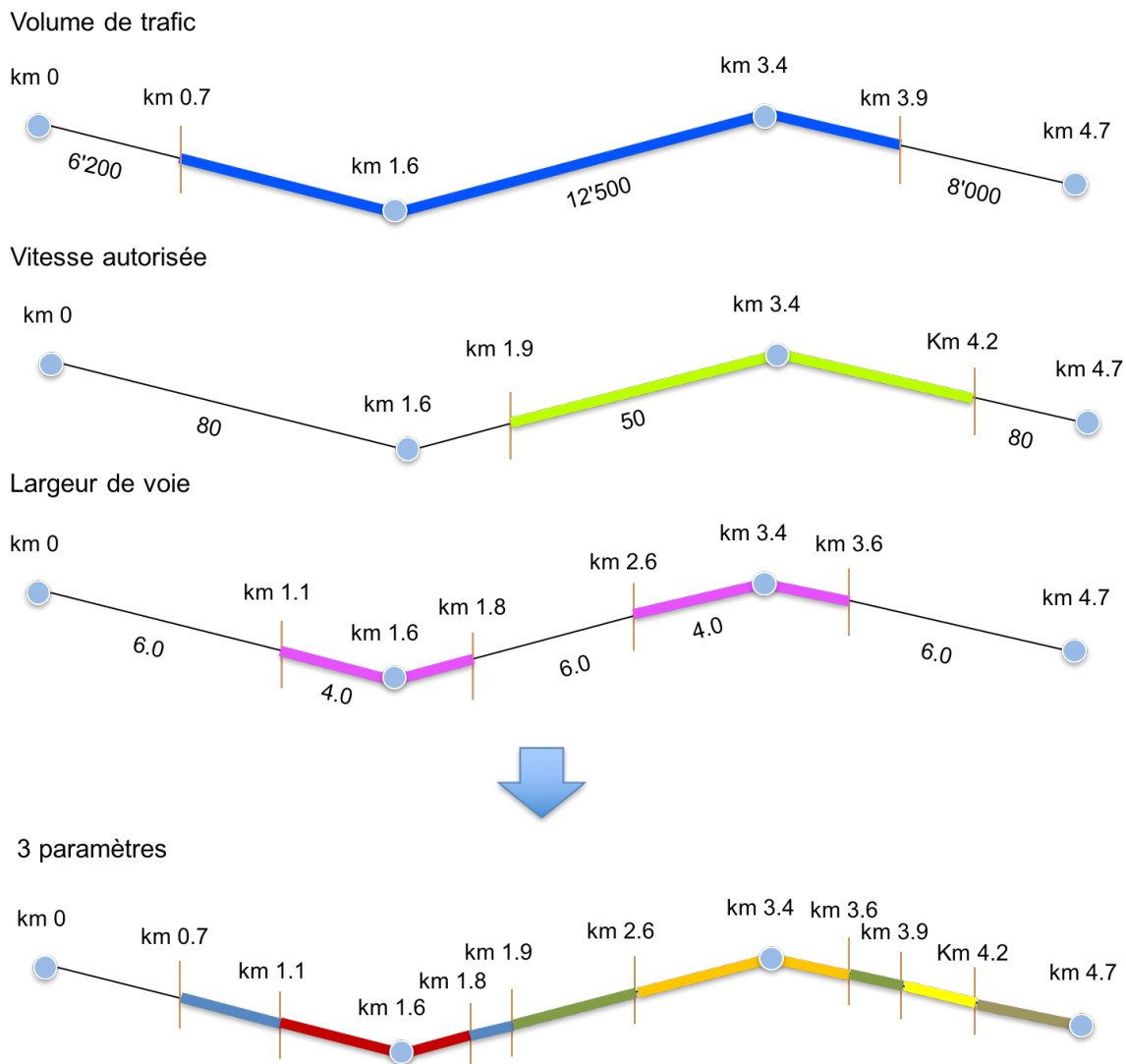


FIGURE 1

4:04

12:00

UNITÉS D'OBSERVATION IRRÉGULIÈRES

Les unités d'observation sont dites *irrégulières* si leurs contours épousent la géométrie de l'objet spatial et sont définies *a priori*. Elles peuvent être ponctuelles, linéaires ou zonales. On parle alors de *mode objet*. Notons que selon l'échelle, une unité ponctuelle peut devenir linéaire ou zonale. Un exemple est la représentation des maisons sur les cartes nationales. Finalement, les unités d'observation irrégulières dont les propriétés thématiques sont les mêmes peuvent être agrégées (les parcelles de zones à bâtir constituent une région).

UNITÉS D'OBSERVATION RÉGULIÈRES

Les unités d'observation *régulières* (aussi appelée maille, cellule ou pixel) correspondent à un élément de *grille régulière*, ou par analogie à une image, à un pixel. On parle alors de *mode image*. Une cellule ne peut contenir la valeur que d'un attribut, c'est la raison pour laquelle il est nécessaire d'avoir autant de grilles qu'il y a de thématiques à représenter. L'attribution de cette valeur peut se faire de plusieurs moyens. On peut par exemple y octroyer sa valeur dominante, sa valeur moyenne ou encore y représenter une information particulière contenue dans le pixel. En mode image, il n'y a pas de définition *a priori* des objets, bien que ceux-ci puissent se révéler par agrégation *a posteriori* d'unités d'observations partageant une propriété thématique commune (forêt, champs, maisons).

DIMENSIONS

Les unités d'observation possèdent trois dimensions fondamentales définies dans le [tableau ci-dessous](#).

DIMENSION	GÉNÉRAL	MODE OBJET	MODE IMAGE
<i>Géométrique</i>	Position, voisinage, forme, taille, proximité	Coordonnées géométriques	Résolution de la maille et géoréférencement de la grille
<i>Thématique</i>	Nature, propriété, état, fonction de l'entité	Attributs thématiques (aussi nombreux que les thèmes)	Attribut unique constituant le thème de la grille
<i>Temporelle</i>	Évolution dans le temps		

Dimensions fondamentales.

La combinaison des dimensions géométrique et thématique donne lieu à la **dimension spatiale**, et la **dimension spatio-temporelle** est obtenue en y ajoutant la dimension temporelle (évolution de la population par commune du canton de Vaud par ex.).

L'information spatiale est l'élément constitutif des systèmes d'information géographique. Elle stocke de l'information localisée tout en exprimant ses propriétés géométriques et thématiques associées. Elle est catégorisée selon l'*échelle de mesure* (nominale, ordinale ou cardinale) et son *origine* (information mesurée, dérivée ou interprétée).



7. MÉTADONNÉES

Les *métadonnées* servent à documenter un jeu de données. Elles permettent de savoir rapidement si le set de données en question répond aux besoins de l'utilisateur ou non.

PRINCIPES GÉNÉRAUX

Les données sont enregistrées dans des bases de données qui brassent des données de type, d'origine et de précision très différentes. Pour garantir leur *traçabilité* et leur *pertinence*, elles sont décrites dans ce que l'on appelle les **métadonnées**. Dans un SIG, l'information spatiale, structurée en couches, est enregistrée dans des bases de données. Les métadonnées décrivent alors toutes les couches, les entités et tous les attributs selon *l'échelle* et la *nature* de l'information.

Les métadonnées (données sur les données) décrivent *objectivement* les données et servent à leur *maintenance*. Elles contiennent notamment leur :

- source (auteur, éditeur, distributeur, contact);
- contenu (nom et description, type de données, échelle de référence et précision, date de saisie);
- quantité (recouvrement spatial, volume de données);
- modalités (format, droit, tarifs, restrictions, adresse).

NORMALISATION

La **normalisation** est le processus qui permet une description universelle et l'échange de données (*compatibilité*). Elle facilite également leur accessibilité. L'*ISA/TC 211* (www.isotc211.org) produit une série de normes concernant les données et géodonnées. Les normes diffèrent légèrement d'un pays à l'autre. Quelques exemples sont cités ci-dessous :

- Union européenne : Directive INSPIRE (inspire.ec.europa.eu).
- États-Unis : Standard du FGDC (www.fgdc.gov).
- Suisse : Norme GM03 (www.geocat.ch).

EXEMPLES

Sur les géocatalogues en ligne (par ex. www.geo.vd.ch), les métadonnées sont généralement disponibles directement avec les données elles-mêmes. Les métadonnées d'une couche sur QGIS peuvent être consultées et éditées en faisant clic droit sur la couche, puis en sélectionnant le menu **Métadonnées**. Entre autres, le fichier contient le format des données, le système de projection utilisé ainsi que le système de coordonnées utilisé.

8. SOURCES DE GÉODONNÉES

DÉFINITION DES DONNÉES PRIMAIRES

Les **données primaires** peuvent être *mesurées* par un instrument ou par une observation, *dérivées* d'informations primaires combinées ou encore *interprétées* si elles ne sont pas mesurables.

MÉTHODES D'ACQUISITION DES DONNÉES

À la base, les données de positionnement (longitude et latitude) sont obtenues par des mesures astronomiques. La latitude est obtenue en mesurant l'angle entre l'horizon et l'étoile Polaire dans l'hémisphère Nord ou l'une des étoiles de la constellation de la Croix du Sud dans l'hémisphère Sud. Elle peut également être obtenue à partir des tables qui contiennent la position des astres, appelées éphémérides. La détermination de la longitude est plus complexe à cause de la rotation de la Terre. Ce n'est qu'à partir du développement de l'horlogerie précise qu'il devient possible d'estimer la longitude par la différence d'heure au midi solaire.

Puis, ces données de positionnement s'affinent par des mesures terrestres. En s'appuyant sur un réseau de points fixes planimétriques dont les coordonnées sont issues de mesures astronomiques régulièrement mises à jour, des mesures de direction et de distances permettent d'établir une **triangulation**. Les directions sont réalisées à l'aide d'un théodolite et les distances à partir d'un tachéomètre. La composante altimétrique est obtenue par des mesures de différences d'altitudes effectuées à l'aide d'un niveau. Un réseau de points fixes altimétrique, différent du réseau planimétrique, sert également de base pour la définition des altitudes sur l'ensemble du territoire.

Aujourd'hui, la position est généralement issue de méthodes satellitaires, comme le GPS (*global positioning system*). Un récepteur interprète les signaux émis par les satellites et, dans le cas du système différentiel tient compte d'une station de référence au sol et indique la position de l'appareil.

APPLICATION : OBTENIR UNE POSITION AVEC UN APPAREIL GPS

Pour obtenir les coordonnées d'un point sur le terrain, on peut se munir d'un appareil GPS. On peut préciser le *système de projection* dans lequel seront données les coordonnées. Généralement, l'appareil permet également de voir la constellation des satellites disponibles pour effectuer les mesures. L'appareil calcule les coordonnées du point en continu. En réalité, les coordonnées obtenues sont entachées de bruit et d'erreurs. C'est la raison pour laquelle en stationnant sur un point, le GPS fournit un nuage de points, plus ou moins étalé. On prend généralement la moyenne géométrique de ces points pour déterminer la position. Dans la pratique, on accumule des mesures en attendant que l'intervalle de confiance sur la position soit assez bon. La durée de la mesure est en général d'une à deux minutes. Il est possible d'afficher les coordonnées obtenues sur un SIG. On observe que l'erreur entre le point mesuré et le point réel est plus faible dans le cas des coordonnées moyennées qu'en ne récoltant qu'une coordonnée, comme le montre l'exemple du MOOC sur les coins du bâtiment du Rolex et à l'entrée du garage souterrain.



LE LIDAR

Ce sont des mesures *laser* (*lidar = light detection and ranging*). Le laser émet un faisceau lumineux qui entre en interaction avec les objets. Une partie de ce signal est réfléchi et captée par un système optique. En filtrant cette information et en connaissant les paramètres avec lesquels a été prise la mesure (orientation et position du laser), il devient alors possible de générer un nuage de points tridimensionnels géoréférencé. Le lidar peut être embarqué dans un avion. On parle alors de *lidar aérien*. La *densité de points* correspond au nombre de points lidar mesurés par unité de surface (généralement le mètre carré).

Les données lidar peuvent servir à la confection du *modèle numérique de terrain* (MNT). Pour cela, les données brutes subissent une série de filtres, de manière à supprimer les points correspondant aux arbres et bâtiments par exemple et à ne conserver plus que les points du sol. Finalement, le lidar peut également être utilisé pour des mesures terrestres, par exemple pour la modélisation d'un carrefour dans une ville.

DONNÉES THÉMATIQUES

Ce sont des données qui représentent un état (qualitatif ou quantitatif) d'une chose ou d'un ensemble de choses. Elles sont acquises par des mesures physiques et ponctuelles. Par exemple, on peut chercher à représenter le débit des rivières, les charges de trafic sur les routes, la météorologie (les mesures sont ponctuelles puis extrapolées à l'ensemble du territoire), etc. Ces mesures sont complétées par de l'imagerie aérienne et satellitaire et définies par le type de capteur utilisé (visible, proche-infrarouge, etc.). Les combinaisons d'images spectrales permettent d'identifier ou de qualifier un paramètre sur le territoire. Par exemple, la combinaison des bandes spectrales 7, 4 et 2 de Landsat 5 permet de mettre en évidence la végétation et l'infrarouge thermique permet de cartographier les températures.

9. ÉTUDE DE CAS – SIG ET PLANIFICATION PARTICIPATIVE

Cette leçon illustre le rôle que jouent les SIG dans la *planification territoriale participative* comme outils d'aide à la décision. L'étude porte sur deux cas dont les problématiques sont différentes, mais les approches sont semblables.

DYNAMIQUES TERRITORIALES

Le premier cas est celui de la ville de Thiès (Sénégal). En 2000, la ville comptait environ 300 000 habitants avec un taux de croissance élevé de 3.3% par an. Le taux d'alphabétisation était faible, surtout chez les femmes et le taux de chômage est élevé (61%), poussant la population à avoir des emplois informels. En 2000, près de 50% de la population vivait en dessous du seuil de pauvreté à Thiès. La ville est gérée par une équipe composée d'un administrateur, un ingénieur et deux techniciens supérieurs. Cela représente bien peu de monde pour la gestion de la ville et pour traiter avec la trentaine d'organisations non gouvernementales (ONG) qui ont un impact direct sur la planification territoriale.

La ville s'est beaucoup développée de 1998 à 2008, notamment dans le Sud-Ouest et le Nord-Est. Par conséquent, ses limites ont dû être retracées. De pair avec sa croissance territoriale, la ville en est venue à déborder de ses limites administratives, et donc à s'étendre sur des terres de la communauté voisine. Des conflits d'intérêts ont éclaté entre les deux communautés et les échanges entre elles se sont rapidement bloqués.

Des problèmes surgissent à l'échelle de la ville, comme la gestion des eaux et des déchets, d'accès et à plus grande échelle comme la déforestation de l'ouest de la ville, qui conduit à une érosion importante des sols, l'alimentation en eau potable, etc.

L'objectif à Thiès est de rétablir le contact entre les communautés et d'optimiser le développement urbain.

Le deuxième cas porte sur les îles des Seychelles, situées dans l'océan indien. Les Seychelles comptent en tout 115 îles (dont 32 granitiques) dispersées sur une surface de 400 000 km² et divisées en 25 districts. En 2006, la population y était de 82 000 habitants dont 95% qui vivaient dans les îles de Mahé, Praslin et la Digue. Le taux de croissance de la population y est faible (0.5% par an). Les ressources des îles sont principalement la pêche et le tourisme (qui crée 30% de l'emploi et 70% du PIB).

Les enjeux territoriaux aux Seychelles sont très fortement liés à la topographie, car les pentes sont fortes sur les îles granitiques. L'espace côtier est donc soumis à de fortes pressions pour les questions d'aménagement du territoire, entre le désir de rentabiliser l'espace en y construisant de nouveaux lotissements par exemple, et la protection de l'environnement qui joue un rôle essentiel dans l'attrait touristique des îles.

L'objectif est donc dans ce cas de concilier la préservation et le développement.

PLANIFICATION PARTICIPATIVE

Une analyse systémique des deux cas a permis de révéler que la planification participative se joue sur trois axes: le *plan physique* (plan matériel, infrastructures), le *plan logique* (où évoluent les acteurs) et le *plan holistique* (où se situent les projets, véritables moteurs des dynamiques territoriales). Le but de la **planification participative** est d'échanger les intérêts de plusieurs groupes de personnes (l'État, les services, les associations, etc.) et de prendre un choix sur la base de ces derniers, en leur conférant des poids selon leur influence et leur importance sur le projet. C'est dans cette phase de forum qu'interviennent les SIG, qui, sur la base d'informations existantes et sur des planifications, évaluent la faisabilité sur la base d'indices précis.

Les acteurs de la décision formalisent le problème, puis le modélisent. On décompose ensuite le processus de gestion en trois étapes:

1. Diagnostic: analyse participative des forces et faiblesses sur le territoire.
2. Décision: fixer les priorités et identifier les actions possibles.
3. Action: mener les actions du point 2, puis évaluer leurs impacts en revenant au point 1.

Dans le cas de Thiès, cela a conduit à l'élaboration d'un arrêté préfectoral portant sur la formalisation d'un cadre de concertation (figure 1). Le cadre de concertation réunit des représentants du Conseil municipal, des délégués du village et des quartiers, des associations, des ONG, etc. Il est dirigé par un comité directeur, et entouré d'une cellule de coordination qui assure le suivi des activités, et une commission technique qui veille à leur faisabilité en fonction des réalités du terrain. Le cadre de concertation se réunit quatre fois par année pour former les utilisateurs, élaborer et valider les données, diagnostiquer les forces et faiblesses du modèle, fixer les priorités et élaborer le projet.

Aux Seychelles, le ministère des collectivités locales a participé à des ateliers de définition du contenu de la plateforme de partage de données. Puis il a fallu mettre en place la plateforme et enseigner son fonctionnement aux différents utilisateurs. Les districts pilotes ont réalisé des ateliers de planification.

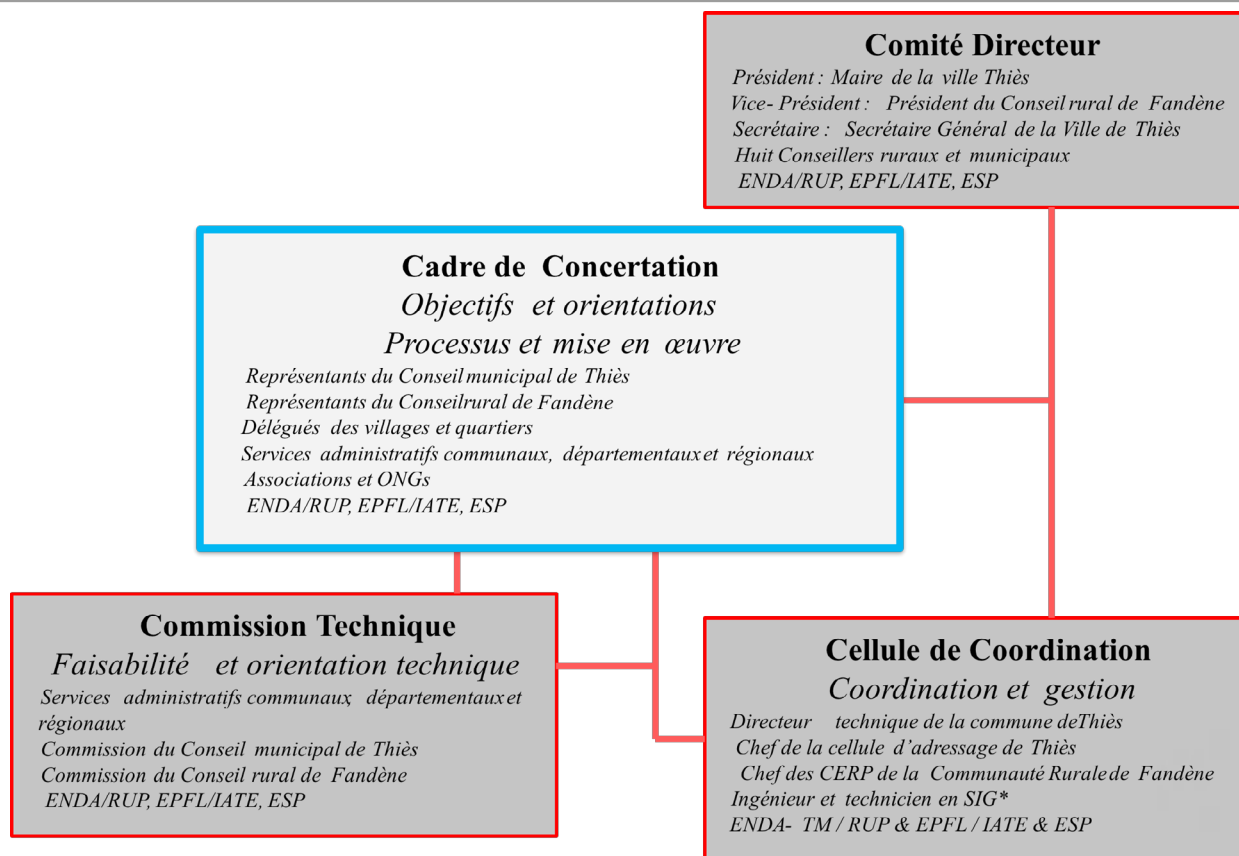


FIGURE 1

RÔLES DES SIG ET CONTRAINTES

Les SIG, dans le processus d'aide à la décision, jouent avant tout un rôle de partage et d'accès à l'information. Les différents acteurs peuvent y échanger leurs idées, afin de trouver ensemble un compromis. Pour cette raison, le SIG doit être simple et facile à prendre en main : il s'adresse à un public élargi. De la même manière, il est nécessaire que sa maintenance soit facile et à faible coût. Raison pour laquelle les jeux de données doivent être aussi ciblés et épurés que possible. Finalement, le processus de mise à jour des données doit être participatif, c'est-à-dire que chacun peut apporter ses modifications, et que celles-ci puissent être validées par un comité spécial qui les enregistrera définitivement dans la base de données.

Les données brutes sont assemblées dans des indicateurs prédéfinis. Ils reprennent les domaines centraux de la problématique (la santé, l'éducation, l'habitat, etc.). Puis les projets sont évalués sur la base de ces indices.

Finalement les SIG permettent d'éditer et d'afficher des éléments, d'évaluer leur influence sur la base d'indices, et de créer des cartes thématiques.

QUELQUES LEÇONS À TIRER

Les deux cas d'étude ont révélé le besoin d'élaborer des modèles simples et sur mesure pour que les différents acteurs puissent s'approprier les outils et partager leurs intérêts.

Il s'est avéré que la photographie aérienne aide à ouvrir la discussion entre les acteurs, qui reconnaissent plus facilement le territoire que sur une carte abstraite.

Finalement, on relève encore la pérennisation du système et l'échange des données via internet. L'utilisation d'une base de données centrale avec une architecture client-serveur facilite considérablement les choses.

10. GÉORÉFÉRENCEMENT D'IMAGES

On souhaite intégrer une image dépourvue de référence spatiale (carte scannée par exemple) dans un SIG. Le géoréférencement est l'étape durant laquelle on lui reconstitue son système de référence spatiale.

MÉTHODOLOGIE

La procédure est divisée en deux étapes. La première consiste en la reconnaissance de points identiques (dits **points de contrôle**) sur l'image et une carte de référence, comme illustré à la figure 1, et la seconde consiste en une série de transformations (de translation, d'échelle et d'orientation) pour minimiser les écarts entre les points de contrôle de l'image et ceux de la carte.

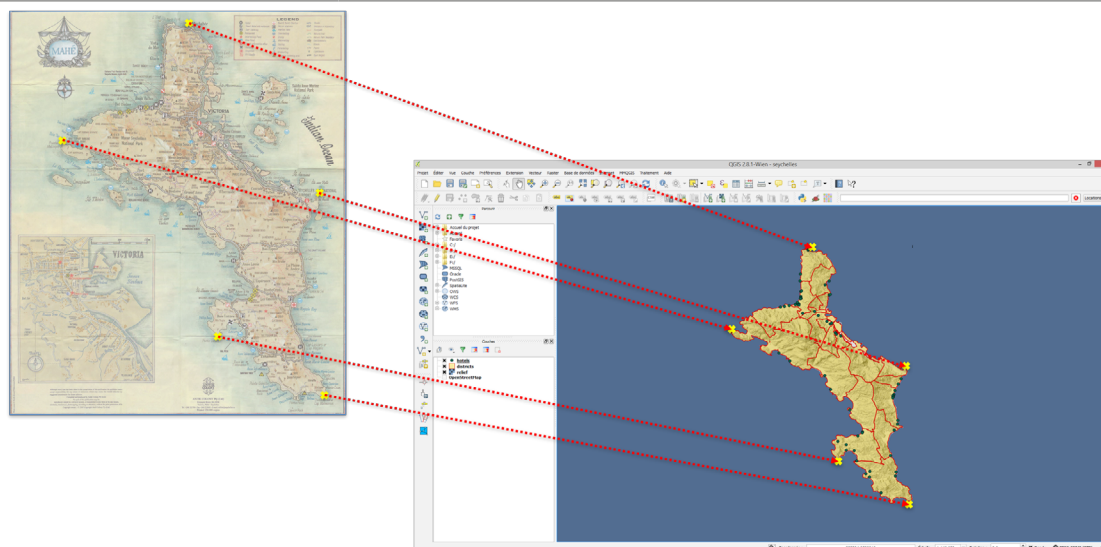


FIGURE 1

2:39

13:00

Points de contrôle sur l'image et sur la carte.

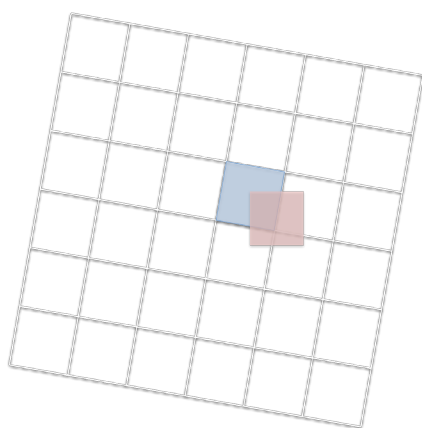


FIGURE 2

4:02

13:00

Ré-échantillonnage d'image selon le plus proche voisin.

Dans la première étape, il est nécessaire de trouver au moins trois *points de contrôle*, de préférence facile à identifier et bien répartis sur l'ensemble de la zone, de manière à apporter une transformation globale de l'image.

La deuxième étape repose sur trois types de transformation: translation, mise à l'échelle et rotation. Elle est généralement basée sur la transformation de Helmert (transformation affine). Il est également possible d'utiliser d'autres transformations (polynomiales, splines et projection) qui sont plutôt utilisées lorsque l'image à projeter contient des déformations locales. Le changement d'échelle et de l'orientation de l'image entraînent tous deux le besoin de *rééchantillonner* l'image. Pour cela, on crée une nouvelle grille raster pour laquelle il faut attribuer des valeurs aux nouveaux pixels. Un moyen de le faire est d'attribuer la valeur du plus proche pixel voisin (figure 2), ou encore une approche bilinéaire ou bicubique si les transformations sont trop importantes.

EXEMPLE SUR QGIS

Ouvrir QGIS avec une carte de référence. Cliquer sur **Raster >> Géoréférencer >> Géoréférencer**. Une fenêtre s'ouvre. Cliquer sur **Fichier >> Ouvrir un raster**. Puis sélectionner l'image que l'on souhaite référencer. Une fenêtre s'ouvre et demande de préciser le système de coordonnées de référence (qui doit être identique à celui de la carte). Désormais, il est possible d'indiquer les points reconnaissables sur l'image et sur la carte. Pour chaque point ajouté correspond une ligne du tableau au bas de la fenêtre de l'outil de géoréférencement. Il faut préciser au moins trois points sur l'image et sur la carte.

Pour lancer la transformation, il faut cliquer sur le triangle vert dans la barre des menus. On précise alors le type de transformation, la méthode de rééchantillonnage de l'image, le fichier de sortie ainsi que son système de coordonnées. Cliquer sur **OK** une fois que tout est prêt. L'image apparaît désormais comme une couche raster sur QGIS, qui devrait être superposée sur la carte. Un bon réflexe est de voir les écarts sur les points de contrôle dans l'outil de géoréférencement pour évaluer la qualité de la transformation.

TRANSFORMATION AFFINE

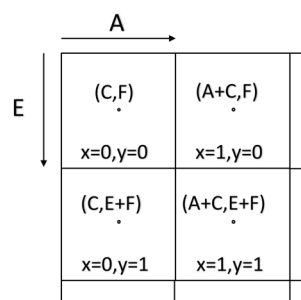
La transformation affine peut s'écrire comme une combinaison linéaire des coordonnées de départ (x,y) . Les nouvelles coordonnées sont notées (x',y') :

Les six paramètres sont :

- A : taille du pixel selon x en unités de carte/pixel
- D : composante y de la hauteur du pixel
- B : composante x de la hauteur du pixel
- E : taille du pixel selon y en unités de carte/pixel
- C : coordonnée x du centre du pixel situé en haut à gauche de l'image
- G : coordonnée y du centre du pixel situé en haut à gauche de l'image

La figure 3 illustre deux cas particuliers. Celui où la transformation affine ne contient pas de rotation et un cas où la transformation comporte une rotation.

Pixels carrés, pas de rotation
 $B=D=0$ et $E=-A$



Pixels carrés, avec rotation
 $B=D \neq 0$ et $E=-A$

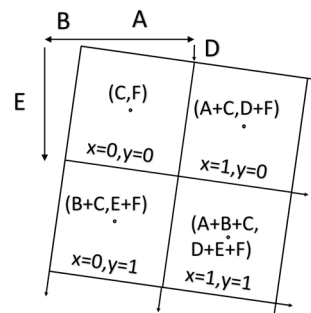


FIGURE 3

9:06

13:00

Deux cas particuliers : pixels carrés sans rotation (à gauche) et avec rotation (à droite)

Les six paramètres de la transformation peuvent être enregistrés dans un fichier (*World file*), qui portera le même nom que le fichier de base et une extension selon le format du fichier de base (TFW si le fichier de base est un TIF, JGW si le fichier de base est un JPG, etc.). Il est également possible d'enregistrer directement cette information dans l'en-tête du fichier de base si celui-ci est un TIF. Son format devient alors *GEOTIFF* et l'extension du fichier reste « .tif ».

L'utilitaire *GeotiffExamine* permet de vérifier si un fichier TIF contient ou non des données de géoréférencement dans son en-tête. S'il en possède un, l'utilitaire permet de l'exporter en World File, et inversement, il permet de créer cet en-tête à partir d'un World File.

Une fois le World File créé, il est possible de modifier son nom et son extension pour le faire correspondre à une image de base que l'on souhaite importer sur QGIS. Il suffit de le renommer convenablement, puis d'importer l'image sur QGIS. Les transformations sont effectuées automatiquement au chargement de la couche.

11. DIGITALISATION D'OBJETS VECTORIELS

Les outils d'édition d'objets vectoriels et de dessin assisté par ordinateur sont très souvent utilisés pour digitaliser des objets à partir d'images aériennes ou satellitaires.

ÉLÉMENTS GÉNÉRAUX

Les objets vectoriels sont organisés en couches thématiques (hôtels, districts, etc.). Ces couches sont généralement organisées par type de géométrie (points, lignes ou polygones). Mais ce n'est pas toujours le cas, et certains logiciels, comme Manifold, sont capables de gérer plusieurs types de géométrie dans une seule couche. Finalement, chaque couche partage une liste d'attributs avec tous ses éléments.

Il est parfois nécessaire de modifier, ajouter ou supprimer des éléments d'une couche ou leurs attributs. La notion de topologie définie à la leçon 5 resurgit ici. Par exemple, lorsque l'on veut ajouter une parcelle dans le cadastre, il est impératif de vérifier la condition que deux parcelles voisines partagent la même frontière. Ce processus s'appelle l'**accrochage**, ou « *snap* » en anglais. Mentionnons encore qu'à chaque nouvelle entité doit correspondre un attribut unique de manière à ce qu'il puisse être identifié sans équivoque. Ce champ est souvent appelé ID, et correspond à une suite de chiffres et parfois à une séquence alphanumérique, correspondant à l'adresse de l'entité.

ÉDITION SIMPLE DANS QGIS

Pour créer une nouvelle couche vectorielle dans QGIS, cliquer sur **Couche >> Créer une couche >> Nouvelle couche shapefile**. Une fenêtre s'ouvre et l'on peut y sélectionner le type de géométrie, le système de coordonnées, le nom des attributs à créer, etc. L'attribut unique ID est déjà présent.

La barre d'outils de numérisation peut être activée sur QGIS en faisant clic droit sur la barre des menus de QGIS, puis en activant le champ **Numérisation**. Un menu d'édition apparaît sur la barre des menus. La figure 1 montre les différentes possibilités d'édition de la couche active.



FIGURE 1

6:05

20:00

Outils de numérisation de QGIS.

Les outils de numérisation de QGIS sont décrits ci-dessous :

1. Effectuer les modifications sur plusieurs couches ou revenir en arrière.
2. Entrer en **Mode édition** sur la couche active.
3. Enregistrer les modifications.
4. Dessiner un nouvel objet de type polygone. Terminer le dessin avec un clic droit.
5. Dessiner un nouvel objet de type ligne. Terminer le dessin avec un clic droit.
6. Déplacer un objet en le sélectionnant.
7. Éditer les géométries et leurs points.
8. Supprimer des objets.
9. Couper un objet.
10. Copier un objet.
11. Coller un objet.

Une barre d'outils concernant les attributs peut être activée de la même manière que celle d'édition des polygones. Pour cela, faire clic droit sur la barre d'outils de QGIS et activer le champ **Attributs**. Cette nouvelle barre d'outils comporte des outils de sélection.

Pour attribuer une valeur unique à l'attribut ID, entrer dans la table attributaire de la couche, cliquer sur la calculatrice de champs et sélectionner la fonction **\$id** dans la rubrique **Enregistrement**.

Il existe un certain nombre de plugins pour la digitalisation du style CAD, notamment le plugin **CADDigitize**. Une nouvelle barre de menus apparaît. On peut l'utiliser pour dessiner des cercles, des rectangles, etc. Dans le menu **CADDigitize**, il est possible de modifier la densité de points (cercles, arcs, etc.).

GESTION DE L'ACCROCHAGE DANS QGIS

Pour afficher les options d'accrochage dans QGIS, cliquer sur **Préférences >> options d'accrochage**. Une fenêtre s'ouvre, sur laquelle on peut sélectionner les couches sur lesquelles on souhaite s'accrocher, si l'on s'accroche aux segments et/ou aux sommets et une tolérance d'accrochage en pixel ou en unité de carte. Si l'édition topologique est activée, alors les sommets partagés par deux objets vont être déplacés simultanément pour les deux objets pour conserver une topologie cohérente.

OUTILS D'ÉDITION AVANCÉS

La barre d'outils d'édition avancée de QGIS (figure 2) peut être affichée en faisant clic droit sur la barre d'outils de QGIS, puis en activant **Numérisation avancée**.



FIGURE 2

13:39

20:00

Barre d'outils d'édition avancée de QGIS.

La barre d'outils comprend les outils suivants :

1. Numérisation avancée (imposer une coordonnée, une distance et/ou un angle).
2. Boutons d'annulation et de rétablissement.
3. Effectuer une rotation d'un objet.
4. Simplifier la géométrie d'un objet.
5. Créer un anneau dans un polygone.
6. Créer une nouvelle partie d'un objet.
7. Créer un nouvel objet dans un objet existant.
8. Supprimer un anneau dans un polygone.
9. Supprimer une nouvelle partie d'un objet.
10. Ajouter une excroissance à un objet.
11. Effectuer une homothétie.
12. Séparer un objet en deux objets distincts.
13. Séparer un objet en deux parties.
14. Fusionner les objets.
15. Fusionner les attributs des objets sélectionnés.
16. Effectuer une rotation des symboles de points.

OUTILS DE DESSIN ASSISTÉ PAR ORDINATEUR (DAO) DANS QGIS

Cet outil permet de construire des géométries en introduisant des conditions manuellement (un point, une distance et/ou un angle fixés).

12. VECTORISATION AUTOMATIQUE

La **vectorisation automatique** est utilisée pour extraire de l'information vectorielle à partir de fonds de plans ou d'images aériennes. Elle peut s'appuyer sur l'utilisation de filtres ou sur la classification supervisée.

PRINCIPES GÉNÉRAUX

Quand elle est appliquée à une image, la vectorisation automatique peut être une *polygonisation* si l'on s'intéresse à des surfaces ou une *détection des segments* si l'on s'intéresse aux structures linéaires. Par opposition, on parle de *traçage automatique* quand on utilise à une couche vectorielle préexistante.

POLYGONISATION

Dans le cas où l'on veut polygoniser une zone homogène clairement délimitée à partir d'une image raster, on peut utiliser la fonction **Polygoniser** de QGIS. Pour cela, cliquer sur **Raster >> Conversion >> Polygoniser**. Il faut préciser le fichier d'entrée (un fichier raster) ainsi que le fichier de sortie (un fichier vecteur).

Pour des structures linéaires, on peut également utiliser la fonction de polygonisation de SAGA, disponible depuis la boîte à outils de QGIS. Pour y accéder: **Traitement >> Boîte à outils**. Dans la fenêtre qui apparaît, sélectionner **SAGA >> Shapes – Grid >> Vectorizing grid classes**. On doit fournir une image monobande à cette fonction pour qu'elle puisse extraire les contours des formes et donner le nom du fichier vectoriel en sortie. Le résultat de la polygonisation produit des polygones qui épousent les contours des pixels, parfois très sinueux. On peut donc être intéressé à simplifier la géométrie des polygones. Pour ce faire, entrer dans **Vecteur >> Outils de géométrie >> Simplifier géométrie**. Dans cette fonction, on précise la couche vectorielle à simplifier, ainsi qu'une tolérance de simplification.

Finalement, si l'on applique l'une de ces fonctions à une image complexe comme une photographie aérienne, la polygonisation va produire un fichier vecteur très complexe également. Seuls les clusters de pixels partageant la même valeur constitueront de grands polygones, tandis que les autres pixels produiront un polygone à eux seuls.

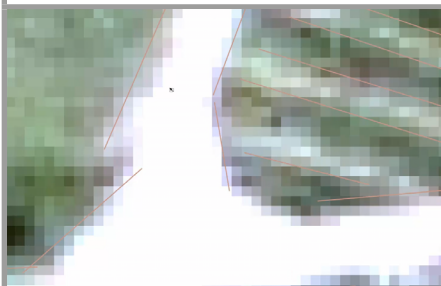


FIGURE 1

11:33 14:00

Produit de la détection de contours automatisée de la fonction **Line segment detection**.

DÉTECTION DES BORDS OU DE SEGMENTS

Les méthodes présentées ci-dessus ne permettent pas de reconstituer des structures linéaires. Raison pour laquelle il existe d'autres fonctions qui recherchent ces structures particulières.

Dans la boîte à outils **Orfeo >> Feature Extraction**, il existe une fonction **Line segment detection**. Il faut lui préciser la couche raster d'entrée. Elle détecte et reproduit automatiquement certains segments linéaires (lignes). En voyant la figure 1, on constate qu'un travail manuel de digitalisation reste souvent nécessaire pour compléter le résultat.

TRAÇAGE AUTOMATIQUE

Si l'on souhaite matérialiser un élément vectoriel, il faut créer une nouvelle couche vectorielle. Le plugin **AutoTrace** permet de construire un polygone qui suit automatiquement le contour du polygone d'une autre couche en y fournissant le point de départ et ceci en cliquant sur le dernier point et en appuyant sur la touche SHIFT. Pour cela, il faut vérifier que la couche active est bien celle du polygone dont on veut suivre le contour dans les options d'accrochage.

13. SOURCES DE GÉODONNÉES

Dans cette leçon, nous allons découvrir qu'il existe une panoplie de *données vectorisées* disponible sur le Net, et directement accessible dans les logiciels SIG. Ces données sont généralement mises à disposition par des organisations internationales ou par des pays. Ces données sont soit disponibles sur des *géoserveurs*, soit il faut les télécharger pour les utiliser.

DONNÉES DISPONIBLES SUR GÉOSERVEURS

Elles sont accessibles moyennant une connexion internet dans une architecture client-serveur. Le client est en général le navigateur WEB et le serveur une machine capable de recevoir des requêtes et d'y répondre. Dans le cas des géodonnées, le géoserveur retourne soit des images, soit des couches vectorielles.

Les fichiers raster sont généralement très lourds en termes de mémoire. Pour faciliter leur transfert, on a souvent recours au *tuilage*. Ce principe repose sur différentes échelles (ou niveau de zoom) pour lesquelles on définit des images de $n \times n$ pixels (par ex. 256×256 pixels). Si l'on zoome sur l'image, cette dernière est divisée en quatre carrés égaux, également composés de $n \times n$ pixels, et ainsi de suite, comme l'illustre la figure 1. L'indexation des tuiles diffère selon le système utilisé (Google, TMS, Quadtree).

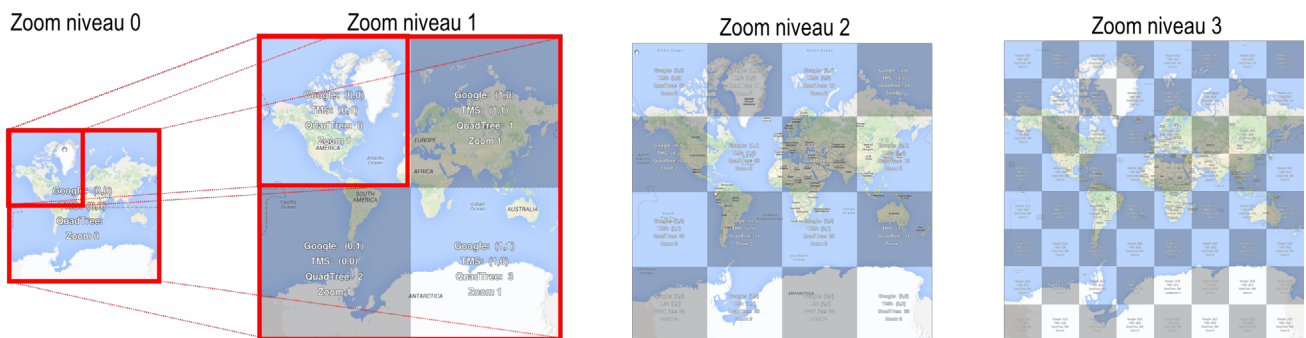


FIGURE 1

3:26

17:00

Tuilage de la carte mondiale (niveaux 0, 1, 2 et 3).

Le tuilage peut être effectué automatiquement par des logiciels géoserveurs via les services WMS (pour les images) et WFS (pour les objets vectoriels), ou peut également être effectué manuellement, par exemple avec la fonction `gdal2tile` de la librairie GDAL.

Exemple de connexion WFS sur QGIS : Sur QGIS, on peut télécharger une couche des communes du canton de Neuchâtel en créant une connexion sur le géoserveur du guichet cartographique neuchâtelois (SITN). Pour y parvenir, il faut se rendre dans le panneau latéral de l'arborescence des fichiers et faire un clic droit sur **WFS >> créer une nouvelle connexion WFS**. Il faut donner un nom à cette connexion ainsi que l'URL du géoserveur. Dans notre cas, l'URL est <http://sitn.ne.ch/ogc-sitn-open/wms>. Une couche *commune* apparaît alors dans l'arborescence des fichiers, que l'on peut intégrer à notre projet en double-cliquant dessus.

Les globes virtuels sont une catégorie particulière de géodonnées, basées sur les cartes de OpenStreetMap, Google Maps (carte ou satellite), Bing Maps, etc. Pour accéder à ces cartes, entrer dans le menu **Internet >> Open layer plugin** et sélectionner le type de carte.

Finalement, il est possible de télécharger des cartes rasters via des outils spécialisés, en précisant une zone d'étude par ses coordonnées et le type de carte (Google Maps, OpenStreetMaps, etc.). L'outil sélectionnera et assemblera automatiquement les tuiles nécessaires.



DONNÉES DISPONIBLES EN TÉLÉCHARGEMENT

Le modèle numérique d'altitude (MNA, ou DEM en anglais) peut être téléchargé à partir de différents sites internet et à différentes résolutions (30 m ou 90 m).

Les images satellitaires à haute résolution sont généralement commercialisées. Celles disponibles gratuitement sont donc de faible résolution, allant de 10 m pour les images SENTINEL à 30 m pour celles de Landsat (15 m pour les images panchromatiques).

Certaines compagnies mettent à disposition différents types de données portant sur la géographie physique. Le tableau ci-dessous présente une liste non exhaustive de géodonnées que l'on peut trouver sur Internet.

TYPE DE DONNÉES	SOURCE
Données générales	http://geoportal.org http://www.naturalearthdata.com
Climatologie	http://www.worldclim.org
Hydrologie	http://hydrosheds.cr.usgs.gov
Glaciologie	http://www.glims.org
Sols	http://soilgrids.org
Couverture du sol	http://due.esrin.esa.int/page_globcover.php http://maps.elie.ucl.ac.be/CCI/viewer
Forêts	http://data.globalforestwatch.org
Environnement (UNEP-PNUE)	http://data.unep.wcmc.org/datasets
EPA (agence américaine de la protection de l'environnement)	https://edg.epa.gov/metadata/catalog/main/home.page
Géologie	http://portal.onegeology.org/OnegeologyGlobal
Énergie	http://globalenergyobservatory.org

Types de données satellitaires et leur source.

Enfin, le tableau ci-dessous résume une sélection de sites permettant de télécharger des données de géographie humaine.

TYPE DE DONNÉES	SOURCE
Données générales Limites administratives Lieux de population Zones urbaines Relations socio-économiques Etc.	http://naturalearthdata.com http://geodata.grid.unep.ch http://maps.worldbank.org http://sedac.ciesin.columbia.edu
Population	http://www.worldpop.org.uk
Toponymie	http://geonames.nga.mil/gns/html
Limites administratives	http://gadm.org

Types de données de géographie humaine et leur source.

14. ÉTUDE DE CAS – BIODIVERSITÉ SÉNÉGAL-MAURITANIE

Cette étude présente les apports des SIG dans un projet de préservation de la biodiversité.

LA VALLÉE DU FLEUVE SÉNÉGAL

La zone d'étude se situe dans le Sahel, dans la vallée du fleuve Sénégal (figures 1 et 2). Aujourd'hui, une attention particulière est portée à cette région qui abrite une biodiversité unique. Les écosystèmes de la région sont intimement liés à la diversité pédologique, aux variations climatiques, au relief et aux structures hydrographiques. Plusieurs ethnies habitent cette vallée, dont les activités sont de fait également diversifiées (agriculture, pêche, élevage extensif). Malheureusement, l'expansion de l'agriculture affecte beaucoup la biodiversité, ainsi que l'utilisation de certaines essences d'arbres pour la construction.

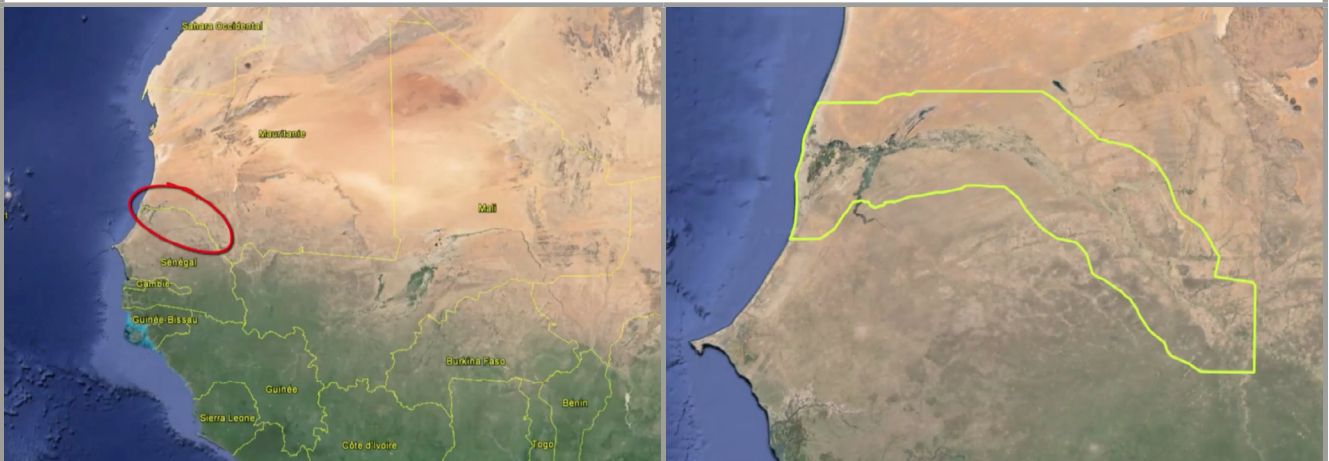


FIGURE 1

1:41

16:00

Vallée du fleuve Sénégal.

FIGURE 2

1:56

16:00

Vallée du fleuve Sénégal.

OBJECTIFS

Les objectifs sont de préserver la biodiversité de la vallée du fleuve Sénégal, ce qui implique la restauration des sols dégradés, apprendre aux communautés locales à gérer durablement les écosystèmes et proposer de nouvelles sources de revenus.

Le rôle des SIG est crucial dans ce type de projet. D'une part ils stockent l'information géographique et socio-économique. Puis c'est une plateforme d'échange de cette information et de planification. Finalement, les SIG permettent d'analyser l'occupation du sol et sa dynamique.

MÉTHODOLOGIE ET SIG

L'analyse de l'occupation des sols est divisée en quatre étapes :

1. sélection et traitement des images ;
2. classification supervisée ;
3. analyse diachronique ;
4. vectorisation, digitalisation et ajout de couches vectorielles.

Pour la première étape, des images multispectrales LANDSAT de 1984 et 2003 dont la résolution est de 30m ont été utilisées. Ces images ont été prises entre la mi-octobre et fin décembre, car c'est à ce moment-là que les végétaux sont le plus facilement identifiables. Il aura fallu sept images pour couvrir la zone d'étude. Il a donc fallu créer la mosaïque de ces sept images.

La deuxième étape est une classification supervisée en combinant les bandes spectrales. Il a fallu identifier les zones témoins sur le terrain. Cette étape permet de définir la signature spectrale des différents éléments au sol (rizière, eau, forêt, etc.). Ensuite, en entrant ces zones témoin dans le logiciel, on a pu procéder à la classification.

L'analyse diachronique, la troisième étape, consiste en la détermination des surfaces de chaque catégorie d'occupation du sol et la détermination des pixels qui ont changé d'occupation entre 1984 et 2003.

La quatrième étape est la vectorisation des zones d'occupation du sol issues de la classification supervisée, et l'intégration de couches comme celles du réseau routier et des limites administratives.

RÉSULTATS

La figure 3 montre l'évolution de l'occupation du sol entre 1984 et 2003.

Globalement, l'occupation du sol liée à l'eau a augmenté, alors que les thèmes plus exposés aux actions anthropiques (comme les formations végétales naturelles) ont sensiblement diminué.

Il faut également tenir compte du Djeri (terres non inondables de la vallée) et du Waalo (terres cultivées dans la partie inondée par les crues annuelles du fleuve). En effet, les zones de cultures sèches sont généralement plus menacées, car elles dépendent entièrement de la pluviométrie.

En conclusion, la biodiversité de la vallée du fleuve Sénégal et soumise à rude épreuve entre les effets combinés des actions anthropiques et les facteurs physiques. Les SIG ont mis en évidence la différence entre les deux types de sols que sont le Djeri et le Waalo, mais surtout une complémentarité entre l'agriculture sous pluie, les cultures irriguées et l'élevage. Finalement, ils ont également permis de quantifier la diminution du potentiel végétal et du desserrement des contraintes d'apport en eau entre 1984 et 2003.

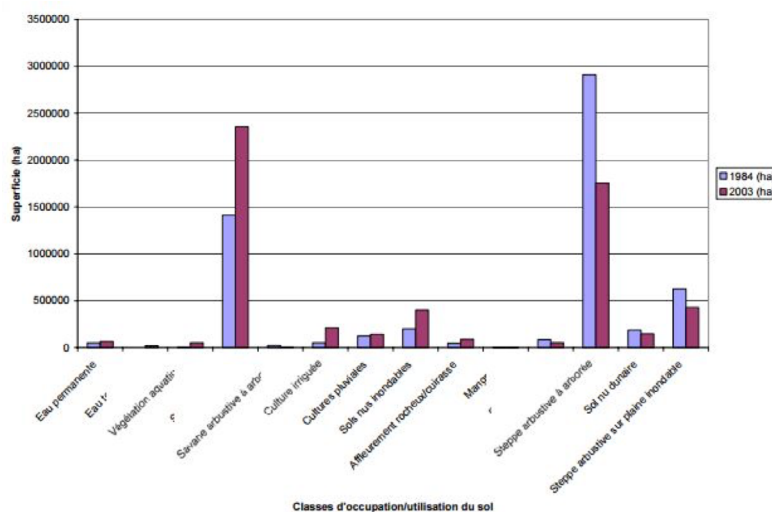


FIGURE 3

15. STOCKAGE DE L'INFORMATION GÉOGRAPHIQUE

Les données géographiques doivent être stockées sur un support numérique pour pouvoir être manipulées à partir d'un SIG. Cette leçon, met en évidence les différences entre les méthodes de stockage des données en mode vecteur et en mode raster.

MODE VECTEUR

Informations à enregistrer : Pour des données de type vecteur, on enregistre pour chaque objet :

- sa *géométrie*: le type de données (polygone, ligne, point) et les coordonnées des sommets qui le composent ;
- ses *attributs*.

Pour une collection d'objets, on enregistre

- le *système de projection* utilisé ;
- la *symbologie* (la représentation graphique des objets) ;
- la *topologie* (les connexions entre différents objets par exemple).

Enjeux du stockage : Le premier objectif du stockage est d'assurer la *persistance* des données, c'est-à-dire qu'une donnée, une fois enregistrée, reste accessible jusqu'à ce qu'elle soit explicitement supprimée. De plus, la *structuration des données* représente un aspect fondamental du stockage. Si l'on enregistre des informations relatives à des parcelles, beaucoup de données vont se répéter selon un même schéma, par exemple le numéro de la parcelle, son propriétaire, sa surface, ce qu'on y cultive, etc. Cette étape de structuration des données est très importante, et fera l'objet de la leçon suivante.

Si l'information est centralisée, des questions de *sécurité* et d'*intégrité* s'imposent. En effet, les utilisateurs peuvent avoir des accès limités aux informations stockées (confidentialité). Finalement, la gestion des données doit reposer sur des outils existants permettant de chercher, d'ajouter, de modifier ou de supprimer des données.

Formats du stockage : Les données vectorielles peuvent être stockées dans des fichiers simples. Si l'information peut être structurée, on préférera l'enregistrer dans des fichiers semi-structurés, ce qui nous amènera à parler des bases de données. Les données centralisées sont accessibles à partir de fichiers ou bases de données selon une architecture client-serveur. Finalement, la gestion des données est faite par l'intermédiaire d'un *système de gestion de bases de données* (SGBD).

Un exemple de **fichier simple** est le format *Well Known Text* (wkt). Il s'agit d'un simple fichier de texte lisible par l'utilisateur. Les géométries sont décrites par un mot-clé (polygone) ainsi que les coordonnées des sommets qui le composent (figure 1). Il s'agit d'un format très répandu grâce à sa flexibilité et sa polyvalence.

Sur QGIS, le plugin *QuickWKT* permet de créer des objets en les décrivant par leur mot-clé et leurs coordonnées. Il est également possible d'exporter une couche au format wkt en faisant click droit sur la couche >> **Enregistrer la couche vectorielle sous...** et en précisant le format du fichier de sortie (csv), et la géométrie (wkt).

Type	Examples	
Point	POINT(30 10)	
LineString	LINestring (30 10, 10 30, 40 40)	
Polygon	POLYGON ((30 10, 10 20, 20 30, 40 30, 30 10))	
	POLYGON ((35 10, 10 20, 15 40, 40 40, 35 10), (20 30, 35 35, 30 20, 20 30))	
Multipoint	MULTIPOINT ((10 40), (40 30), (20 20), (30 10)) MULTIPOINT (10 40, 40 30, 20 20, 30 10)	
MultiLineString	MULTILINestring ((10 10, 20 20, 10 40), (40 40, 30 30, 40 20, 30 10))	
MultiPolygon	MULTIPOLYGON (((30 20, 10 40, 10 40, 45 40, 30 20), (15 5, 40 10, 10 20, 5 10, 15 5)))	
	MULTIPOLYGON (((40 40, 20 45, 45 30, 40 40)), ((20 35, 45 20, 30 5, 10 10, 10 30, 20 35), (30 20, 20 25, 20 15, 30 20)))	

FIGURE 1

16:03

21:00

Type de données et exemple d'information enregistrée.

Un autre type de fichier simple est le *shapefile* de ESRI, très répandu dans le monde des SIG. Il est composé d'au moins trois fichiers :

- shp : contient les entités géographiques ;
- dbf : contient les données attributaires (peut être ouvert par n'importe quel logiciel tableur, comme Microsoft Excel) ;
- shx : contient un index des entités géographiques.

Un troisième type de fichier simple très répandu est le dxf d'Autocad. Il s'agit d'un format de dessin assisté par ordinateur (DAO). Il décrit des géométries simples (points, lignes, polygones), mais sans référence spatiale ni symbologie. Le fichier peut être lu par des logiciels de dessin, comme LibreOffice Draw par exemple. Il est possible d'importer un fichier dxf dans QGIS en ajoutant une couche de type vecteur. QGIS propose de sélectionner les couches vectorielles en fonction de la géométrie (point, lignes, polygones).

Les attributs sont stockés dans des fichiers auxiliaires (csv ou dbf).

Les **fichiers structurés** contiennent à la fois les données géométriques, attributaires et symbologiques. Un exemple de fichier structuré est le XML, où l'information est structurée par *balises*.

QGIS peut exporter des couches vectorielles dans des fichiers structurés comme le GML (*geography markup language*, produit dérivé du XML) ou le GeoJSON. Ces deux formats peuvent être consultés sous la forme de texte (chaque ligne correspond à un objet) ou de *fichier hiérarchisé*.



MODE RASTER

Informations à enregistrer: Dans le cas d'une image, il est important d'enregistrer les coordonnées d'un point de référence. Par convention, on enregistre en général les *coordonnées du coin en haut à gauche de la grille*. Il faut encore préciser le *système de projection*. Chaque valeur de la grille correspond à la variable que l'on souhaite stocker.

Enjeux du stockage: Les enjeux du stockage en mode raster sont similaires à ceux en mode vecteur. Par contre, les conséquences et types de support sont différents. Le stockage sous forme de fichiers simples, d'image ou de table suffit à la persistance des données. La question de la structuration des données est triviale puisque l'on enregistre des images ou des tables de données X, Y et Z. En ce qui concerne la centralisation des accès et les fonctionnalités de gestion, on utilise des *fichiers simples* ou des *tables simples* d'une base de données dans une architecture client-serveur et l'on effectue des modifications grâce à des systèmes de gestion de base de données spécifiques aux données raster.

Formats du stockage: Pour les rasters, on peut utiliser les formats suivants:

- *Grille ou matriciel:* constitué d'un en-tête et d'un contenu sous la forme d'une matrice. Exemples de fichiers: asc, grd, flt et hdr.
- *Liste:* constitué de multiples lignes et de trois colonnes (X, Y et Z). Exemples de fichiers: txt, csv et dat.
- *Image:* constitué d'un en-tête et d'un contenu sous la forme d'un tableau. Exemples: tif, jpeg, png et gif.

L'*information colorimétrique* peut être stockée de deux manières différentes. Soit elle est enregistrée pour chaque pixel (rouge, vert et bleu et la composante alpha si elle existe), soit elle est référencée à une palette des couleurs présentes dans l'image. Dans le deuxième cas, chaque pixel contient l'index de sa couleur dans la palette. Dans QGIS, si l'image est enregistrée selon la première méthode, on peut alors sélectionner les couleurs à attribuer à chaque bande de l'image. Pour une image enregistrée avec un index des couleurs, on peut modifier globalement la couleur de tous les pixels ayant le même index.

Finalement, l'information de positionnement de l'image peut se trouver dans l'en-tête d'un fichier (comme c'est le cas pour le GeoTiff), ou dans un world file annexe accompagnant le fichier (tfw, jgw, pgw, gfw).



16. BASES DE DONNÉES RELATIONNELLES

Les *bases de données relationnelles* sont la forme de bases de données la plus répandue aujourd'hui. Cette leçon décrit de quelle manière l'information, spatiale ou non, est enregistrée dans les bases de données.

DÉFINITION ET FIABILITÉ

Une **base de données** est une *collection de données persistantes*, centralisée ou non, servant pour les besoins d'une ou plusieurs applications, *interrogeables* et *modifiables* par un groupe d'utilisateurs travaillant en parallèle.

Un **système de gestion de bases de données (SGBD)** est un logiciel permettant de créer, structurer, alimenter, mettre à jour et consulter une base de données ; c'est, en fait, l'interface entre la base de données et les utilisateurs ou leurs programmes.

Les modifications d'une base de données sont le résultat de *transactions*. Une **transaction** est une suite d'opération qui fait passer la base de données d'un *état initial intègre et cohérent* à un *état final également intègre et cohérent*. Pour ce faire, les transactions doivent respecter les **propriétés ACID**:

- **Atomicité**: la transaction se fait complètement ou pas du tout. Si une transaction échoue, on revient à l'état initial.
- **Cohérence**: ensemble de contraintes d'intégrité qui doivent être respectées à l'issue de la transaction. Si ce n'est pas le cas, la transaction est annulée.
- **Isolation**: l'exécution des transactions l'une après l'autre doit donner le même résultat que l'exécution simultanée. Sinon on revient à un état valide et cohérent.
- **Durabilité**: l'enregistrement en mémoire de la transaction a lieu au moment de la validation afin d'éviter de passer par une mémoire tampon qui pourrait s'effacer en cas de coupure de courant par exemple.

TYOLOGIE HISTORIQUE

Le stockage sous forme de fichiers s'est développé dans les années 1950 à 1960. Puis de 1960 à 1970, les bases de *données hiérarchiques* (notion de dépendance parent-enfant, un enfant ne peut avoir qu'un seul parent) et les *bases de données réseau* (variante de la BD hiérarchique où un enfant peut avoir plusieurs parents, faisant apparaître des structures cycliques dans les bases de données). Les *bases de données relationnelles* apparaissent à partir des années 1970, dans lesquelles l'information est enregistrée dans des tables. Finalement, depuis les années 1990 sont apparues les *bases de données objet* et *semi-structurée*.

MODÈLE RELATIONNEL

Le *modèle relationnel* a été développé par IBM pour répondre aux *propriétés ACID*. Il a pour but d'éviter les redondances d'information. Il repose sur des bases solides puisqu'il dépend de l'*algèbre relationnelle*. Il s'agit du modèle le plus largement répandu actuellement dans les systèmes de gestion de base de données.

Les données sont organisées sous forme de *tables* que l'on appelle *relations*. Les colonnes correspondent aux *attributs*, caractérisés par un nom et un domaine (*integer*, *boolean*, etc.), et chaque ligne correspond à un enregistrement (ou *tuple*). Si un attribut n'est pas évalué pour un tuple, il prend la valeur NULL.

Dans le modèle relationnel, chaque objet doit obéir à la contrainte qu'il est unique. De ce fait il ne peut pas apparaître deux fois dans la même table. Pour vérifier cette contrainte, on définit un *identifiant unique*, appelé *clé primaire*, qui permet de référencer sans ambiguïté chaque objet. Cette clé primaire peut être créée pour l'occasion ou peut être un attribut ou un ensemble d'attributs tel qu'il n'existe jamais deux tuples qui partagent les mêmes valeurs pour ces attributs. Par définition, une clé primaire ne peut pas contenir la valeur NULL.

Lorsque deux tables sont liées (une table « étudiant » et une table « cours » par exemple), on définit des *clés étrangères* (ou identifiants externes) qui permettent de faire le lien entre les deux tables. La référence doit pointer sur un objet unique de la table référencée. La vérification de l'intégrité référentielle est effectuée par le système de gestion de la base de données.

Le modèle relationnel ne comprend pas la notion d'*attributs multivalués* (une personne qui a plusieurs prénoms par exemple). Pour y remédier, on peut créer une nouvelle relation pour les prénoms qui contient une référence de la personne. La notion d'*attribut complexe* n'existe pas non plus. L'adresse est un bon exemple dans ce cas. Faut-il l'enregistrer dans plusieurs attributs (rue, NPA, localité) ou enregistrer un attribut unique? En fait, cela dépend de l'application. Par exemple, si on a besoin de référer le numéro postal NPA, il faudra opter pour la première solution.

Pour manipuler les tables, on fait appel à un certain nombre d'opérateurs. Ils sont listés ci-dessous :

- *Union*: assemble deux tables sans répéter les doublons.
- *Différence*: soustrait deux tables. Les objets redondants de la première table sont supprimés.
- *Produit croisé*: Tous les tuples de la première table sont associés à tous ceux de la seconde.
- *Sélection*: extrait des tuples (lignes dans les relations).
- *Projection*: extrait des attributs (colonnes dans les relations).
- *Jointure*: associe deux tables sur un attribut commun.

La différence n'est pas une opération commutative.

COMPOSANTE SPATIALE

Pour les *bases de données spatiales*, on ajoute trois informations. La première est le *type de données* (point, lignes, polygones, etc.). La deuxième est l'*indexation* des objets spatiaux qui permet de retrouver rapidement un objet dans un SIG. Finalement, la troisième est l'*ensemble des fonctions spatiales* que l'on peut appliquer aux données.

Indexation: L'indexation permet d'optimiser la recherche d'informations dans une base de données. Il existe de nombreux types d'indexation. Il existe des index non spatiaux, comme le *Hash table* et le *B-tree*. Pour les index spatiaux, on présentera le *R-tree* (similaire au B-tree), le *quadtree* et le *grid*.

L'indexation en Hash table est très efficace pour des recherches d'information de type « égal ». On précise un attribut à trouver dans la table. L'attribut passe dans une fonction de hachage (Hash-fonction) qui détermine la position de l'élément dans la base de données.

L'indexation en B-tree est une arborescence hiérarchique qui trie de manière ordonnée l'information selon un attribut (figure 1). Il est très pratique pour des recherches d'informations du type « plus grand que » ou « plus petit que ».

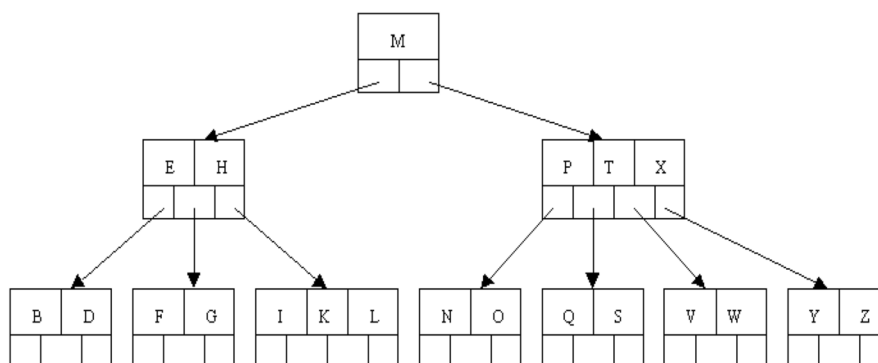


FIGURE 1

L'indexation en R-tree est similaire à la B-tree, à la différence qu'elle gère l'information spatiale (figure 2). Les objets les plus proches sont regroupés au niveau supérieur par leur *enveloppe minimale*, puis les enveloppes sont elles-mêmes regroupées au niveau supérieur par leur *enveloppe minimale*, et ainsi de suite. De cette manière, si une requête spatiale n'intercepte pas une enveloppe minimale, on sait qu'il n'est pas nécessaire d'en inspecter son contenu mais qu'il faut poursuivre dans une autre enveloppe minimale.

L'indexation quadtree possède une *structure arborescente* dans laquelle chaque nœud possède quatre enfants. Elle permet de partitionner un espace bidimensionnel en le divisant récursivement en quatre quadrants ou régions (figure 3). Cette indexation est utilisée par de nombreux systèmes de tuilage de cartes géographiques comme Quadkey, Tile Map Service, Google, etc.

R-tree Hierarchy

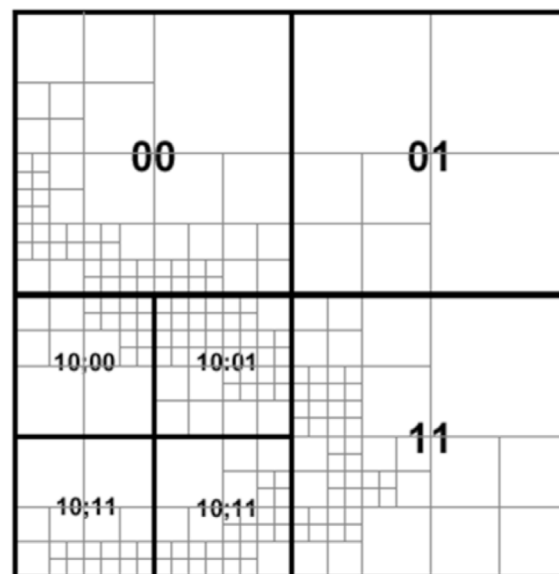
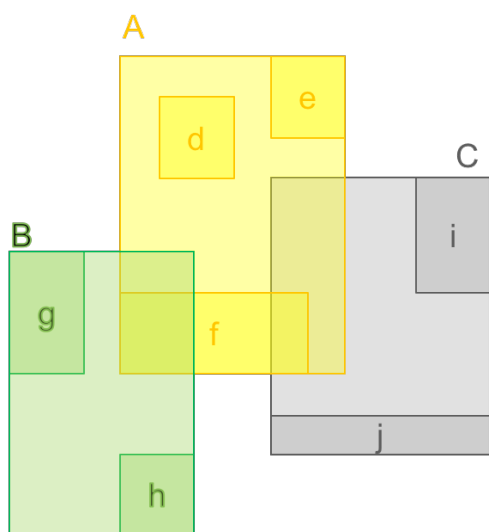


FIGURE 2

22:33 31:00

Illustration de l'indexation spatiale en R-tree.

FIGURE 3

23:33 31:00

Illustration de l'indexation spatiale Quadtree.

L'indexation en grille est similaire à l'indexation quadtree, à l'exception que chaque cellule n'est pas nécessairement divisée en quatre, mais peut être fractionnée en neuf ou en seize sous-cellules régulières par exemple (figure 4).

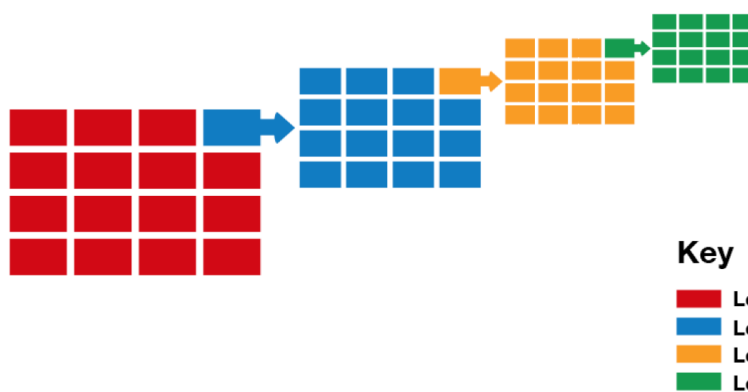


FIGURE 4

23:59

31:00

Illustration de l'indexation spatiale en grille.

Fonctions: Les fonctions spatiales sont principalement réparties dans trois grandes familles: les *fonctions de mesure* (longueur, surface, distance, etc.), les *fonctions spatiales qui créent des entités* (zones tampon, intersection, union, différence, etc., voir figure 5) et les *opérateurs topologiques* qui retournent des valeurs booléennes (disjoint, touche, conjoint).

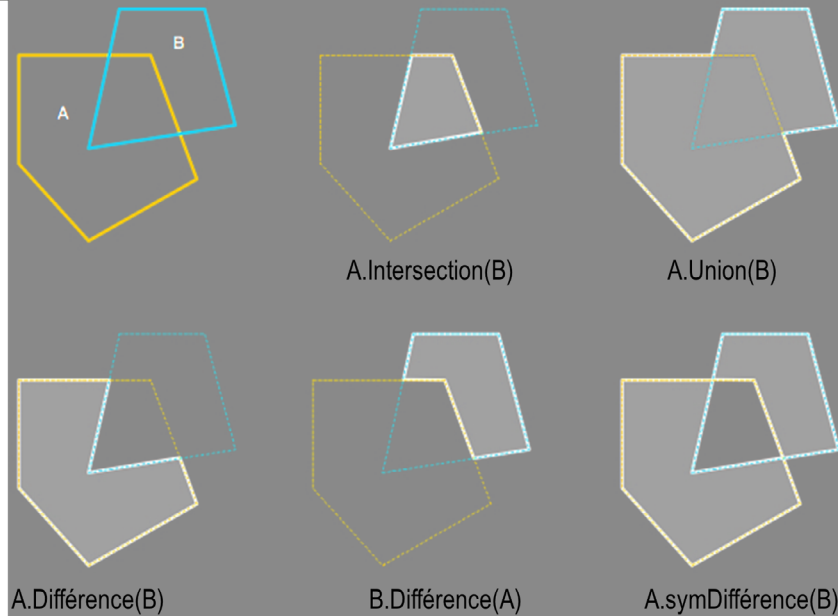


FIGURE 5

24:22

31:00

Exemples de fonctions spatiales.

LOGICIELS DE SYSTÈME DE GESTION DE BASES DE DONNÉES

Sur le marché, il existe de nombreux systèmes de gestion de bases de données consacrées aux bases de données relationnelles (IBM DB2, SQLite, Microsoft AQL Server, Oracle, PostgreSQL, MySQL, etc.). Ils sont accessibles par l'intermédiaire de logiciels payants ou libres qui offrent une interface utilisateur et un certain nombre de fonctionnalités. Dans les fonctionnalités essentielles que ces logiciels doivent offrir, on retrouve la gestion de la structure des données, leur consultation et gestion ainsi que la construction et l'exécution de requêtes.

Importation de données dans un SGBD: Dans cet exemple, on exporte deux fois la couche vectorielle des districts de Mahé à partir de QGIS (sélectionner la couche, clique droit >> **Enregistrer la couche vectorielle sous...**). La première fois on l'exporte dans un format de base de données SpatiaLite (extension .sqlite) avec l'information spatiale et la deuxième fois dans un format texte (.csv) sans l'information spatiale. On utilise ensuite le SGBD *SQLiteStudio* (version 3.0.5) pour ouvrir les deux fichiers exportés précédemment. La première étape consiste en la création d'une base de données, que l'on appelle *mahe_data.sqlite*. Puis il est nécessaire de se connecter à la base de données créée. On peut désormais importer nos deux tables. Pour le fichier texte, cliquer sur le bouton importer, écrire le nom de la table sous **Table**, cliquer sur **Next** et sélectionner le fichier texte dans l'arborescence de fichiers en indiquant que la première colonne contient des en-têtes. Une table apparaît dans la base de données, qui contient les attributs de la couche exportée. Puis on importe la base de données avec les éléments de géométrie. Pour ce faire, cliquer sur le menu **Base de données** puis sur ajout d'une base de données. Indiquer le fichier .sqlite exporté précédemment. Finalement il est nécessaire de se connecter à cette nouvelle base de données. On remarque qu'elle comporte un nombre d'objets beaucoup plus important dû à la composante spatiale. Elle contient également la table avec les attributs des districts de Mahé.

17. MODÉLISATION DES DONNÉES

L'étape de modélisation des données est primordiale dans la conception d'une base de données relationnelle. Elle permet de comprendre rapidement comment l'information est stockée et permet donc de pérenniser la base de données.

RÔLE DE LA MODÉLISATION

L'exemple des parcelles qui a déjà été utilisé illustre très bien le rôle que joue la modélisation des données. Avec les règles du jeu :

- Chaque parcelle a un propriétaire.
- Un propriétaire peut posséder plusieurs parcelles.

On sent bien dans cet exemple qu'il faut créer deux tables: une pour les parcelles, et une pour les propriétaires. En effet, si l'on enregistrerait toute l'information dans une seule table, on répéterait le nom des propriétaires qui possèdent plusieurs parcelles, et cela accroîtrait les chances de se tromper lors de mises à jour des données du propriétaire. La solution dans cet exemple est donc de se référer à un identifiant unique de la table des propriétaires dans celle des parcelles.

La modélisation des données consiste en une schématisation de la structuration des données. Mais elle ne s'arrête pas là, elle peut aussi couvrir des aspects liés à la mise en place du système (comme les fonctionnalités du système par exemple).

Finalement, cette étape ne se limite pas aux SIG, mais s'inscrit dans le contexte général du génie logiciel. Les SIG ne font qu'emprunter ses outils.

MÉTHODOLOGIES DU GÉNIE LOGICIEL

Merise: Il s'agit d'une méthode séquentielle datant des années 1970. Toutes les phases du cycle de vie du développement logiciel sont présentées successivement, comme l'illustre la figure 1.

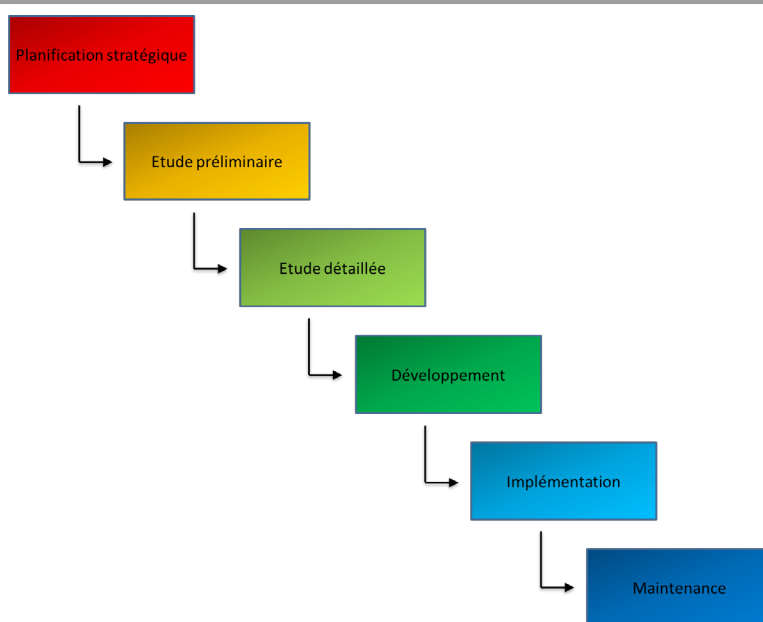


FIGURE 1

RUP: La *méthode RUP* (rational unified process) a la particularité d'être une méthode itérative. Elle repose sur un certain nombre d'activités qui ont lieu lors des différentes phases de développement d'un logiciel. Les activités atteignent des niveaux d'intensité propres à chaque phase, comme le montre la *figure 2*. La modélisation des données a lieu au début du processus, mais sera revue et corrigée à plusieurs étapes ultérieures de la création du logiciel.

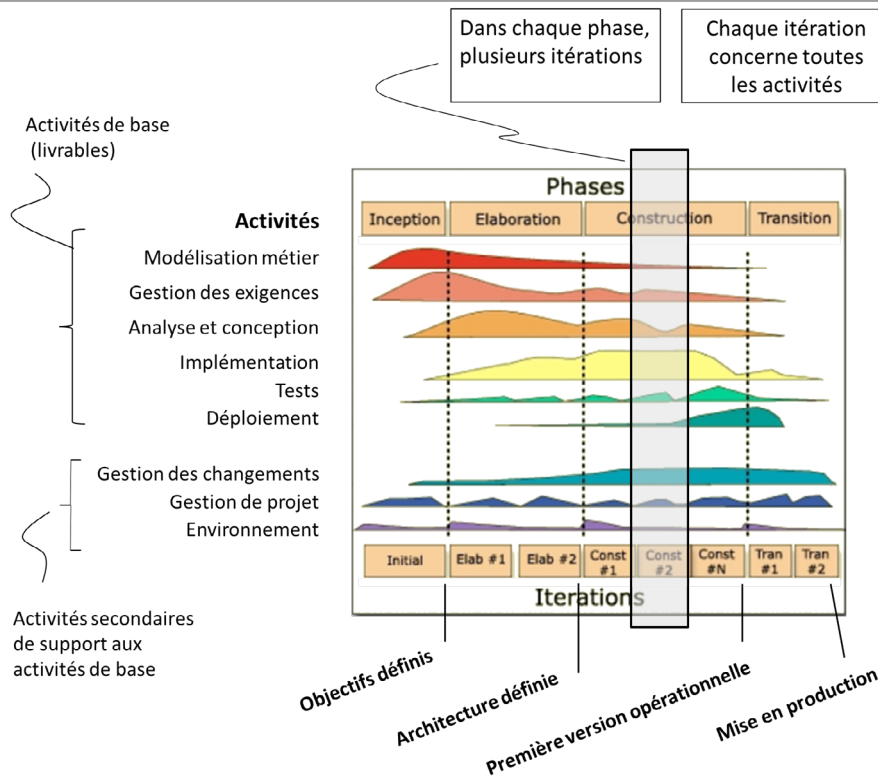


FIGURE 2

3:12

27:00

Illustration de la méthode RUP.

Méthodes « agile »: Les méthodes Merise et RUP sont des méthodes relativement rigides et ne conviennent pas toujours pour la conception de nouveaux logiciels. Les *méthodes agiles*, basées sur le développement adaptatif et centrées sur la production, la validation et le test en continu du logiciel offrent un cadre plus souple. (voir https://fr.wikipedia.org/wiki/Manifeste_agile)

SYNTHÈSE DES MÉTHODES

Les trois méthodes présentées mettent toutes en avant quelques éléments communs :

- *L'analyse des besoins*. En effet, il faut commencer par identifier les utilisateurs et les tâches qu'ils devront effectuer. Il devient donc possible de faire l'inventaire des données nécessaires ainsi que des fonctionnalités à mettre en place pour le logiciel.
- *La formalisation* qui représente les résultats de l'analyse des besoins dans un modèle.

La modélisation des données est une étape qui vise à :

- maîtriser la complexité de la thématique (outil de réflexion);
- faciliter les échanges et la validation des idées (outil de communication);
- améliorer le processus de développement (outil technique);
- faciliter la maintenance du système (outil de documentation).

Il s'agit d'un *modèle conceptuel*, parce qu'il s'accompagne généralement d'un croquis ou d'une schématisation des concepts et idées. Puis il est traduit dans un *modèle logique* qui formalise l'information sous forme de tables, attributs, domaines et formats. Finalement, ce modèle logique est implémenté dans un SGBD et l'on parle alors de *modèle physique*.

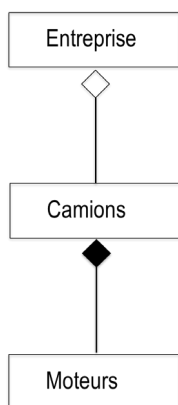


FIGURE 3

10:27

27:00

Exemple d'agrégation simple et d'agrégation composite.

MODÈLE ENTITÉ-ASSOCIATION

Une entité est un objet. Dans l'exemple « l'automobile bleue de Georges », il y a deux entités: l'automobile et Georges. Une classe est un ensemble d'entités de même nature. Les entités sont des réalisations d'une classe « parent » abstraite.

Une *association* est un lien entre deux entités. Par exemple, « Georges est propriétaire d'une automobile bleue » est une association entre deux classes: propriétaire et automobile. Une classe d'association est par extension l'ensemble des liens entre paires d'entités.

La modélisation des données porte donc sur les classes d'objets (entités) et leurs relations (associations). On parle ainsi de *diagramme de classe* pour désigner le modèle entité-association. Une **classe** est principalement caractérisée par son nom (automobile, propriétaire) et par ses propriétés (le nom, l'adresse et l'âge du propriétaire; le type, la marque et la couleur de l'automobile).

De son côté, une **relation** est caractérisée par son nom et sa terminaison. La *terminaison* comprend le rôle de chacune des classes l'une envers l'autre, la *multiplicité* (le nombre d'objets susceptibles d'occuper la position définie par la terminaison d'association) et la *navigabilité* (si la relation est unidirectionnelle).

L'**agrégation simple** définit la situation où un élément plus grand (tout) est composé d'éléments plus petits (parties). Si le tout disparaît, les parties subsistent. Supposons que nous ayons deux classes: l'entreprise et les camions. Les deux classes peuvent être agrégées (les camions sont une partie de l'entreprise). Si l'entreprise venait à disparaître, les camions, eux, existeraient toujours. L'agrégation simple est indiquée par un losange vide sous la classe parente sur le modèle entité-association. Au contraire, une **agrégation composite** (ou composition) définit le cas où si le tout disparaît, les parties disparaissent *de facto*. Considérons une nouvelle classe du moteur à notre modèle. Si le camion n'est plus, il en va de même pour son moteur. Dans ce cas, l'agrégation composite est indiquée au moyen d'un losange rempli sous la classe parente. Cet exemple est illustré à la figure 3.

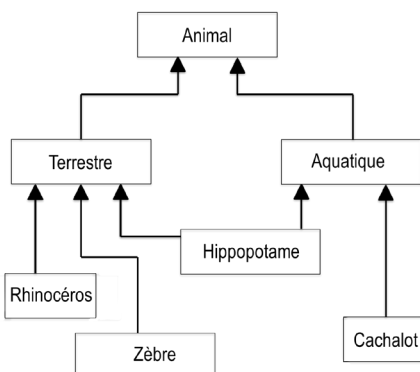


FIGURE 4

10:43

27:00

Exemples d'association généralisation-spécialisation.

Il existe également des **associations de généralisation-spécialisation**.

Il s'agit de la relation entre une classe générale (parent) et une classe spécialisée (sous-classe). La sous-classe hérite des propriétés de la classe parente, et possède des éléments supplémentaires. Un objet de la classe spécialisée peut être utilisé partout où la classe parente est autorisée. Dans le schéma entité-association, on dessine une flèche de la sous-classe à la classe parente. Par exemple la classe animaux est composée des sous-classes animaux terrestres et animaux aquatiques, etc., comme le suggère la figure 4. Dans l'exemple, la sous-classe Hippopotame a deux parents. Dans ce cas, on parle d'héritage multiple.

Le langage le plus répandu pour représenter le modèle entité-association est le langage **UML** (*unified modelling language*). Il s'agit d'une représentation en diagramme de classe. La représentation n'est pas unique, mais peut prendre plusieurs formes (plus ou moins détaillée).

EXEMPLES

La construction d'un modèle de données consiste à définir et représenter schématiquement les classes qui seront utilisées pour décrire un domaine thématique et montrer comment ces différents objets sont liés les uns aux autres.

Hiérarchie des classes de géométrie : Exemple de la hiérarchie des classes utilisée pour décrire les objets spatiaux dans une base de données spatiale. La figure 5 présente le modèle de la hiérarchie des classes géométriques définie par l'OGC en 1999.

L'élément de base est la *géométrie* (*Geometry*) à laquelle est associé un système de référence spatiale. La géométrie peut être des *points*, des *courbes* ou des *surfaces*, ou une collection de géométries. On utilise des pictogrammes pour faciliter la lecture du graphique (à droite de chaque classe). Les courbes peuvent être décrites par des polygones, qui sont elles-mêmes définies par un ensemble de points. Les surfaces peuvent être des polygones. Une collection de géométrie peut être composée de plusieurs surfaces, plusieurs courbes ou plusieurs points. À nouveau, les surfaces peuvent être des polygones et les courbes des polygones. Les multiples polygones, multiples polygones et multiples points sont respectivement une composition de polygones, polygones et points. Finalement, la polygone peut être spécialisée en polygone simple ou en polygone fermée. Si la polygone est fermée, elle est un élément constitutif des polygones. On peut encore préciser la multiplicité sur les multipoints qui sont composés d'au moins un point et les polygones d'une frontière.

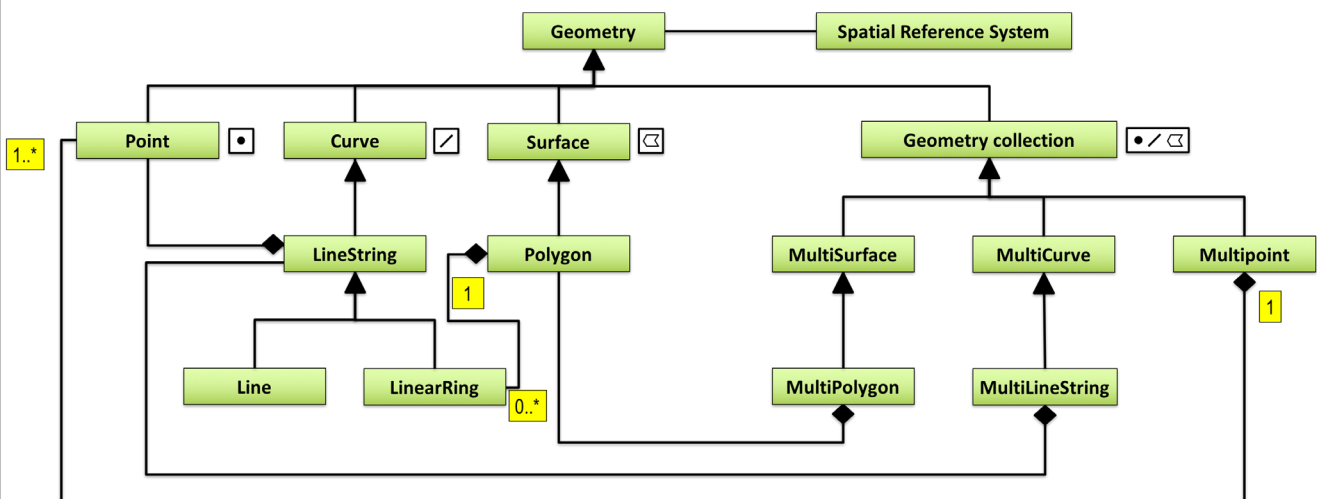


FIGURE 5

13:23

27:00

Geometry class hierarchy.

Donnée de ressources en eau: Le deuxième exemple traite de la modélisation des données des ressources en eau que l'on veut intégrer dans un SIG. On est intéressé à représenter les cours d'eau, les lacs, les bassins-versants, les sections transversales (pour les calculs de débits), les prélèvements (pour l'arrosage, l'électricité, etc.), les sites de mesure des débits et les aspects environnementaux comme l'écomorphologie. Le modèle de données est présenté à la figure 6. On peut encore affiner le modèle en y ajoutant les attributs relatifs à chaque classe du modèle, ce qui rend le modèle complet mais assez complexe.

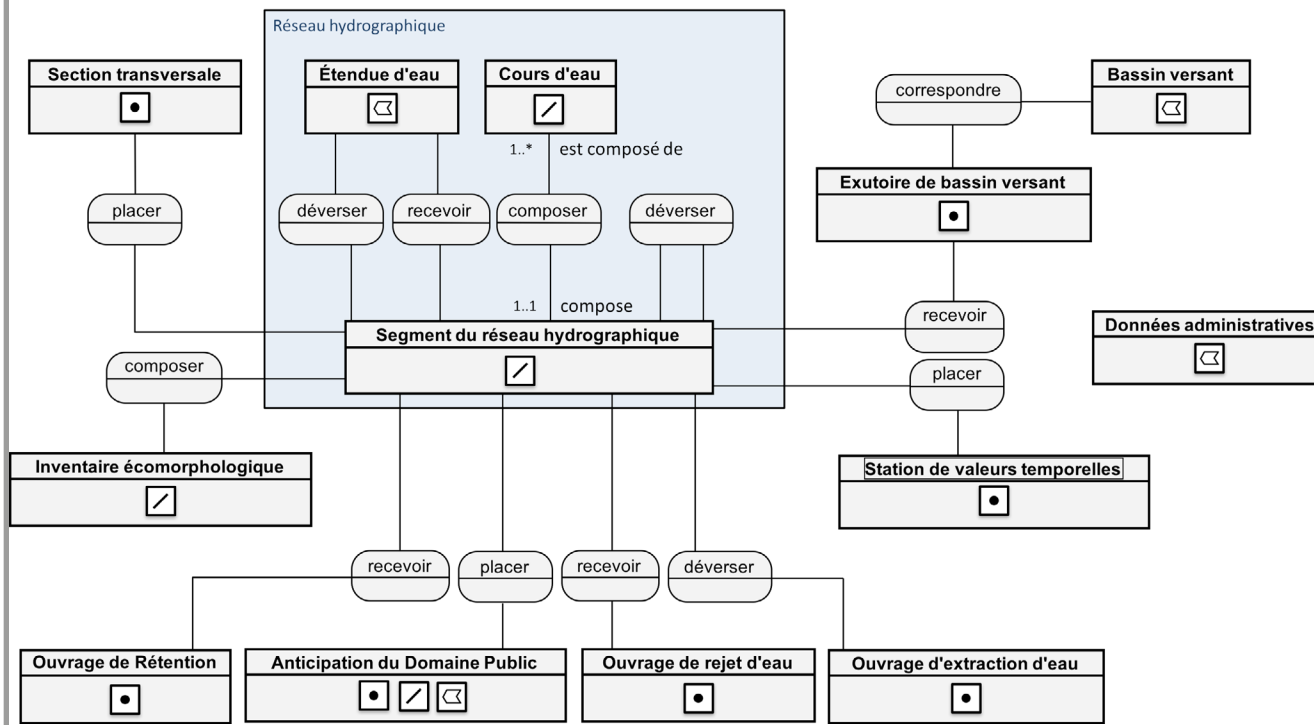


FIGURE 6

15:36

27:00

Modèle de données du réseau hydrographique.

OUTILS CASE

Les **outils CASE** (*computer aided software engineering*) sont des outils de génie logiciel. Ils ont trois fonctions :

- définition de structures de données ;
- création d'une base de données physique à partir d'un modèle logique (*forward engineering*) ;
- génération du modèle de données logique correspondant à une base de données physique (*retro engineering*).

Les SIG possèdent généralement des SGBD capables de définir la structure des données. Dans QGIS, on peut ouvrir le gestionnaire de base de données (menu **Base de données >> Gestionnaire de base de données >> Gestionnaire de base de données**) et ajouter une nouvelle table (menu **Table >> Créer une nouvelle table**). Pour créer la table, on définit son nom, le nom de ses attributs et on précise une clé primaire. Il est encore nécessaire de préciser le type de géométrie ainsi que le système de coordonnées utilisé. Pour créer des liaisons entre différentes bases de données, on peut utiliser l'outil de jointure qui se trouve dans les propriétés de la couche pour lequel il faut donner la couche à joindre et le champ commun aux deux tables. Cette jointure n'est effective que pour le projet QGIS, mais ne sera pas reflétée dans un système de base de données indépendant, quand on ouvrira la couche.

Dans le système de gestion de bases de données pgAdmin, on peut ajouter des clés étrangères sur un attribut. Pour ce faire, faire clic droit sur la table et entrer dans les propriétés puis aller dans l'onglet **Contraintes**. Cliquer sur **Ajouter**, entrer le nom de la clé et préciser la table et l'attribut à référencer. Terminer en cliquant sur **Ajouter** puis **OK**. Ainsi, la clé étrangère fait désormais partie de la couche et sera visible à partir de n'importe quel SGBD.

Les logiciels CASE peuvent être répartis en deux grandes familles : les logiciels spécialisés dans la modélisation de diagrammes UML (pas uniquement les modèles de données) et ceux spécialisés dans la modélisation de données et qui incluent les forward et retro engineering. Dans le deuxième cas, les logiciels se différencient selon le type de bases de données et s'ils gèrent les géométries spatiales ou non.

Reconstruction du modèle de données (*retro engineering*) : Le logiciel pgModeler (de PostgreSQL) permet de reconstituer le modèle de données d'une base de données et d'ajouter des tables comportant des liaisons.



18. CRÉATION DES BASES DE DONNÉES TEST

Cette leçon illustre comment créer les bases de données qui sont utilisées par la suite pour les requêtes SQL (voir leçon 20). Les données sont stockées sous la forme de fichiers simples (shapefile), dans une base de données autonome (Spatialite) ou dans une base de données sur un serveur (PostGis).

IMPORTATION DES FICHIERS SOURCES

Dans QGIS, pour importer une couche vectorielle de type shapefile, cliquer sur **Couche >> Ajouter une couche >> Ajouter une couche vecteur**. Dans l'exemple, on importe les couches des districts des Seychelles, celles des hôtels, des lieux-dits et du réseau routier. Ajuster les paramètres selon vos choix (couleur de fond de carte, échelle) et enregistrer le projet QGIS en cliquant sur **Projet >> Enregistrer** sous le nom de Seychelles.qgs.

Il est possible de regrouper les couches importées dans un *groupe*. Faire un clic droit dans la barre des couches >> **Ajouter un groupe** et le nommer «shapefiles». Puis il suffit de glisser les couches dans le groupe pour les y associer.

CRÉATION DE LA BASE DE DONNÉES SPATIALITE

Pour créer une base de données Spatialite depuis QGIS, il est nécessaire d'installer l'extension *QSpatialite*. Pour ce faire, cliquer sur le menu **Extension >> Installer/Gérer les extensions** puis rechercher l'extension. Cliquer sur **Installer l'extension** si cela n'est pas encore fait. L'extension est désormais disponible sous le menu **Base de données >> Spatialite >> QSpatialite**.

Ouvrir *QSpatialite* puis créer une base de données en cliquant sur l'icône comportant une clé à molette bleue. L'explorateur de fichiers s'ouvre. Sélectionner un dossier. Enregistrer la base de données sous le nom de mahe.sqlite.

Importer les couches vectorielles depuis QGIS en cliquant sur l'icône de QGIS et sélectionner toutes les quatre couches chargées précédemment. Sélectionner l'option **Desination SRID**, choisir **EPSG:32740** et cliquer **OK**.

Les couches apparaissent à gauche sous **My Tables**. En cliquant dessus, on s'aperçoit que lors de l'importation, Spatialite ajoute automatiquement une clé primaire PKUID à chacune des couches. Cet effet involontaire peut être modifié en ouvrant la couche depuis un système de management de bases de données. Ouvrir le logiciel SQLiteStudio (3.0.5). Cliquer sur **Database >> Add a database**. Une fenêtre s'ouvre. Sélectionner la base de données mahe.sqlite et cliquer sur **OK**. La base de données apparaît à gauche. Pour s'y connecter, sélectionner la base de données en cliquant dessus puis cliquer sur le menu **Database >> Connect to the database**. Développer **Districts** et **Columns** puis clic droit sur **PKUID >> Delete the column** et confirmer en cliquant sur **Yes**. Pour définir la colonne ID comme clé primaire, cliquer droit sur **ID >> Edit the column**. Préciser le que le type de la colonne ID est INTEGER et sélectionner la contrainte **Primary Key**. Finir en cliquant sur **OK**. Cliquer sur le vu vert pour valider les modifications. Faire de même pour les tables des hôtels, lieux-dits et routes. Dans la table des hôtels, modifier le type de données des chambres, des lits et l'ID_DIST en INTEGER en précisant que ce dernier est une Foreign Key. À droite de l'option **Foreign Key**, cliquer sur **Configure** et sélectionner la table districts comme **Foreign table**. Cliquer sur le vu vert pour valider les modifications.

Une fois les modifications terminées, fermer le programme SQLiteStudio et retourner sur QGIS. Ouvrir l'extension *QSpatialite*. Il est maintenant possible de charger les couches depuis la base de données dans QGIS. Pour ce faire, cliquer droit sur chacune d'entre elles et cliquer sur **Load in QGIS**. Les couches apparaissent dans le gestionnaire de couches de QGIS. Créer un groupe nommé Spatialite et y insérer les couches de la base de données. Ajouter le suffixe «_S» aux couches pour les différencier des couches shapefile. À ce stade, c'est un bon réflexe de sauvegarder le projet QGIS en cliquant sur la disquette bleue.

CRÉATION DE LA BASE DE DONNÉES POSTGIS

Ouvrir l'interface utilisateur de pgAdmin de PostgreSQL et PostGis (disponible gratuitement sur internet) pour accéder à la base de données. Dans l'**Object browser**, double-cliquer sur **PostgreSQL** puis cliquer droit sur **Database >> New Database**. Définir le nom de la base de données (Seychelles) et préciser le nom de l'utilisateur, cliquer sur **OK**.

Ouvrir la base de données en double-cliquant dessus. Cliquer droit sur **Extensions >> New Extension** et sélectionner **PostGis** dans le menu déroulant. Il s'agit de l'extension spatiale de PostgreSQL. De cette manière, il est possible d'importer dans la base de données PostgreSQL les couches spatiales de QGIS.

Retourner sur QGIS. Si ce n'est pas déjà fait, installer les extensions Processing et DB Manager. Cliquer sur le menu **Traitement >> Boîte à outils** puis sélectionner **Géotraitements QGIS >> Database >> Import into PostGis**. Une fenêtre s'ouvre. Définir la couche à importer dans la base de données PostGis, préciser la clé primaire et terminer en cliquant sur **Run**. Répéter la manipulation pour chacune des quatre couches vectorielles (districts, hôtels, lieux-dits et routes).

Ouvrir le menu **Base de données >> Gestionnaire de base de données >> Gestionnaire de base de données**. Ce gestionnaire donne accès aux bases de données PostGis et Spatialite. Ajouter les couches de la base de données PostGis à QGIS en faisant cliquer droit **>> Ajouter au canevas** sur chacune d'entre elles. Finalement, créer un groupe pour les couches PostGis dans le gestionnaire des couches de QGIS. Appeler ce groupe PostGis. Cette fois-ci, ajouter le suffixe «_P» à chacune des couches exportées.

Les fichiers et nos bases de données sont maintenant prêtes à être testées avec des requêtes SQL.



19. ÉTUDE DE CAS – DÉVELOPPEMENT ET AMÉNAGEMENT URBAIN À TOUBA

Ce chapitre présente une étude d'aménagement à Touba. Il sera question de la gestion des transports en lien avec le développement urbain.

LA VILLE DE TOUBA

Touba se situe au Nord-Est de Thiès au Sénégal. Elle est la capitale de la confrérie mouride. Sa population augmente significativement (25 000 habitants en 1988 contre 500 000 en 2008). Touba est une ville spirituelle. Elle héberge la Grande Mosquée, ainsi que plus de 500 mosquées secondaires situées dans les différents quartiers de la ville. Elle accueille également plusieurs pèlerinages, dont le Grand Magal qui attire plusieurs millions de personnes dans la ville. Touba est dépourvue du statut officiel de ville, parce que ses 33 000 hectares constituent en fait un titre foncier privé. Par conséquent, elle ne bénéficie pas des structures administratives usuelles d'une ville (services techniques, de l'urbanisme, etc.). La gestion de la ville est faite de manière empirique et montre un bon fonctionnement jusqu'à présent.

Les besoins de planification de la ville sont, du côté urbain: les lotissements, l'accès à l'eau potable ainsi que l'assainissement, la gestion des déchets et l'alimentation en électricité. Du côté de la mobilité et des transports: gestion des différents types de transports et de voiries ainsi que la gestion de la période du Grand Magal.

ÉTUDE DES DÉPLACEMENTS ET AMÉNAGEMENTS URBAINS

L'étude présentée ici a été confiée par l'agence autonome de travaux routiers à un consortium de bureaux formé de Transitec (transport et mobilité), Urbaplan (urbanisme), Ipogee (SIG) et Municipalité Service (coordination). Ses objectifs étaient d'établir un diagnostic de la situation, et de planifier des solutions d'aménagements urbains et de transports à court-moyen terme ainsi que les moyens de leur mise en œuvre et définir la stratégie globale d'aménagement à long terme. Deux scénarios sont à considérer: continuer à centraliser le pôle d'activité sur la Grande Mosquée (comme c'était le cas jusqu'à présent) ou en développer de nouveaux à différents endroits de la ville.

L'étude a été divisée en quatre phases:

1. analyse et diagnostic de la situation actuelle;
2. conception d'un modèle global du développement urbain et des transports;
3. planification formelle (à l'aide des SIG);
4. proposition des solutions.

Les SIG étaient indispensables pour faciliter l'accès aux données (notamment cadastrales) pour les différents acteurs ainsi que l'édition de données, l'intégration d'indicateurs statistiques et la création de cartes montrant les différents produits de l'étude.

PRINCIPAUX RÉSULTATS DE L'ÉTUDE

Cette étude a montré l'existence d'une zone à forte densité à l'intérieur de la rocade (140 hab/ha) entourée d'un anneau à plus faible densité (70 hab/ha), comme l'illustre la figure 1.

Principales caractéristiques en matière d'urbanisation

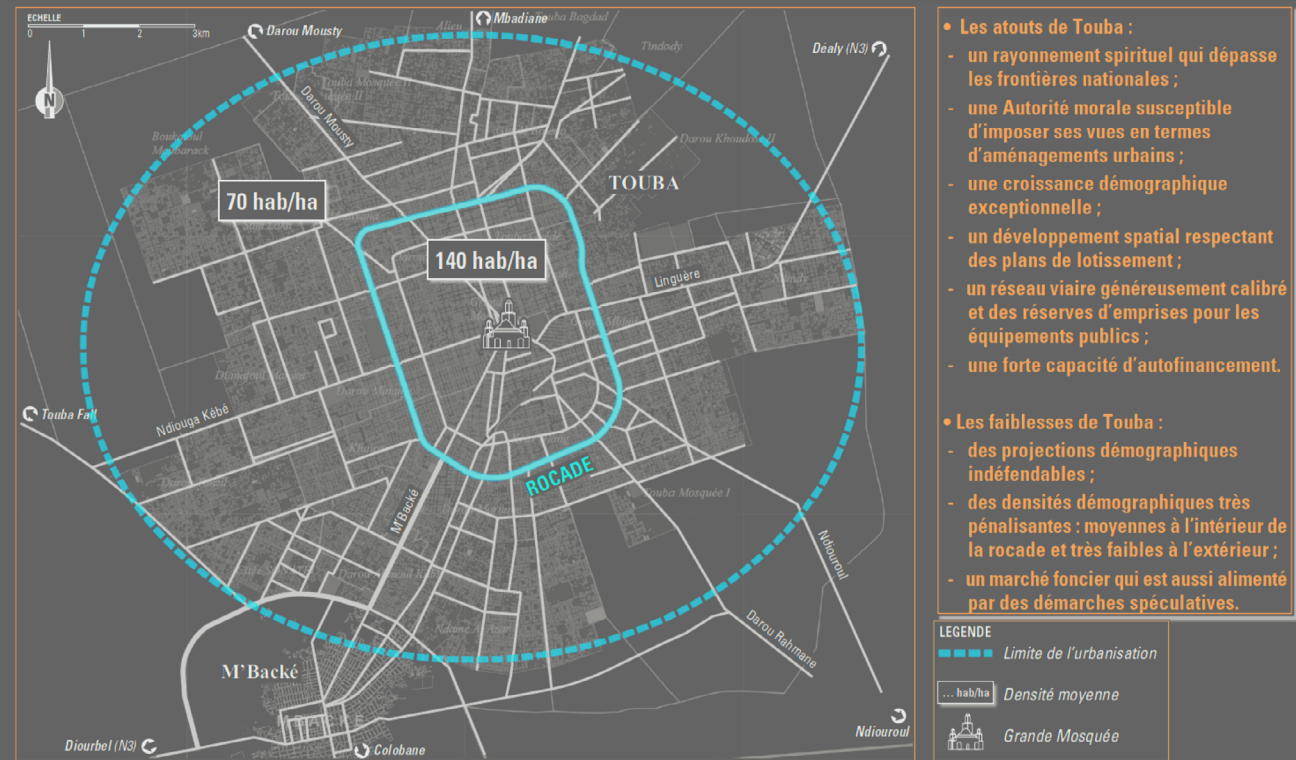


FIGURE 1

9:14

31:00

Principales caractéristiques en matière d'urbanisation de la ville de Touba.

Cette étude relève l'attraction spirituelle de la ville avec la Grande Mosquée et le Grand Magal, une importante croissance démographique, le respect des plans de lotissements, un système de voies de circulation généreux (routes larges). Elle montre aussi la faiblesse des projections démographiques, la faible densité d'habitation au sein de la ville, permettant certes d'accueillir les pèlerins lors des grands pèlerinages annuels mais qui rendent importants les coûts d'extension des réseaux en eau, électricité, etc. Bien que les voies de circulation soient généreuses, quelques problèmes subsistent dans la mixité des types de transports (voitures, poids lourds, piétons, charrettes, etc.), le débordement de constructions et de marchés sur la chaussée, l'organisation peu claire des gares routières et des transports publics en général, etc.

L'étude a démontré que la marche à pied constitue la part principale du mode de transport (50%), suivie par les déplacements hippomobiles (20%), les transports collectifs (15%). Les transports individuels représentent finalement une assez faible part des modes de déplacements (10%). Cette mixité de modes de déplacements sur les routes conduit à des conflits de vitesse et de priorité, la saturation des routes et le stationnement désorganisé.

La figure 2 illustre le cercle vicieux de l'absence de planification des déplacements dans la ville de Touba. La non-maîtrise du développement urbain est caractérisée par l'absence de transports en commun performants, qui induit une perturbation due à la circulation ayant pour effet de dévaloriser les modes de déplacements doux (marche, vélo). La conséquence de cela est le recours à des modes de transports collectifs informels, potentiellement dangereux et qui freinent le développement économique lié à la mobilité.

Les conséquences de la non-planification de l'urbanisation et des déplacements à Touba

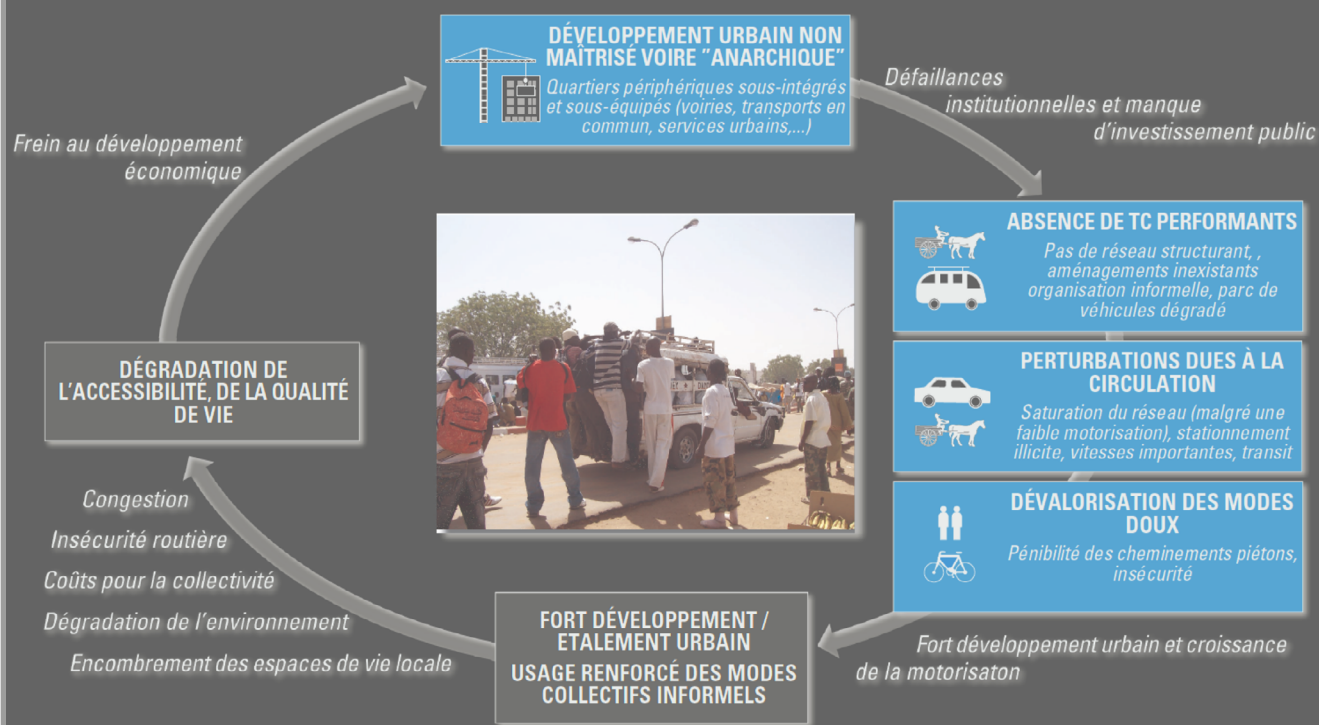


FIGURE 2

12:40

31:00

Conséquences de la non-planification de l'urbanisation et des déplacements à Touba.

Les avantages de la planification de la mobilité (figure 3) seraient une meilleure organisation des transports publics, la mise en place d'un réseau hiérarchisé entre les différents utilisateurs des routes. Cela accompagnerait la maîtrise du développement urbain et de la croissance de la mobilité urbaine améliorant l'accessibilité de la ville, et donc son développement économique.

Les enjeux de la planification de l'urbanisation et des déplacements à Touba

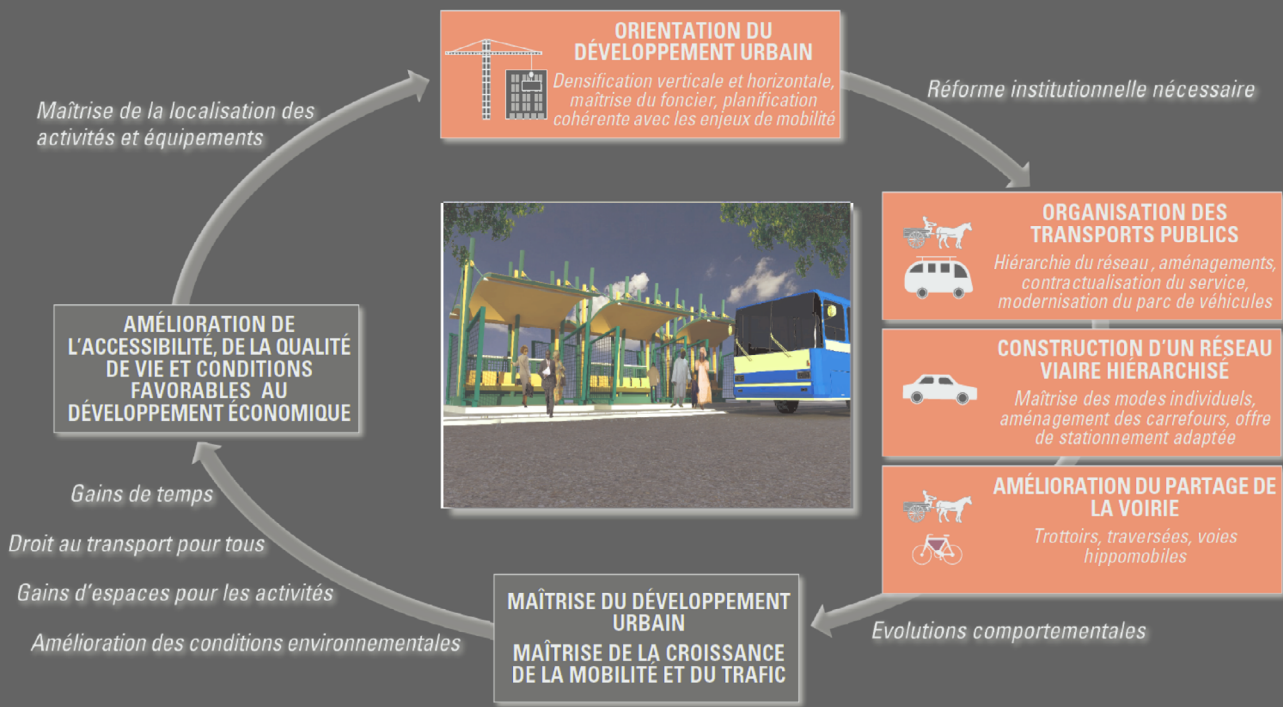


FIGURE 3

13:33

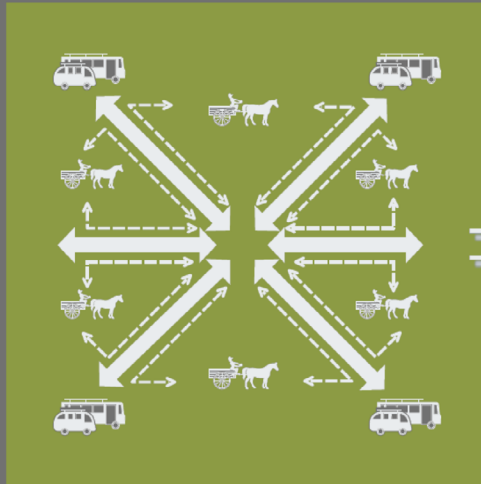
31:00

Avantages de la planification de l'urbanisation et des déplacements à Touba.

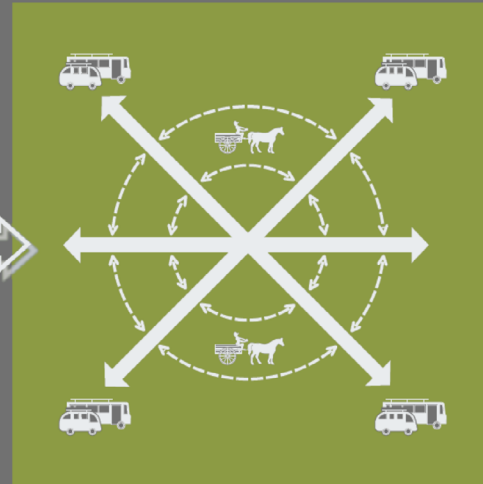
Les objectifs généraux de l'étude montrent un fort accent sur le développement de la mobilité et des transports. Le concept général propose de passer d'un système radial avec mixité entre différents modes de transports à un système diagonal pour les transports publics motorisés et de rabattement pour le mode hippomobile, comme le montre la figure 4.

Concept général proposé en matière de transports publics

PASSER D'UN SYSTEME RADIAL AVEC MIXITE ENTRE TRANSPORTS MOTORISES ET HIPPOMOBILE ...



... A UN SYSTEME DIAGONAL POUR LES TRANSPORTS PUBLICS MOTORISES ET DE RABATEMENT POUR L'HIPPOMOBILE



CONSEQUENCES :



- suppression des arrêts prolongés aux terminus du centre-ville
- suppression des manoeuvres de rebroussement au centre-ville
- suppression de la gêne occasionnée par la mixité "hippomobile - véhicules motorisés"

FIGURE 4

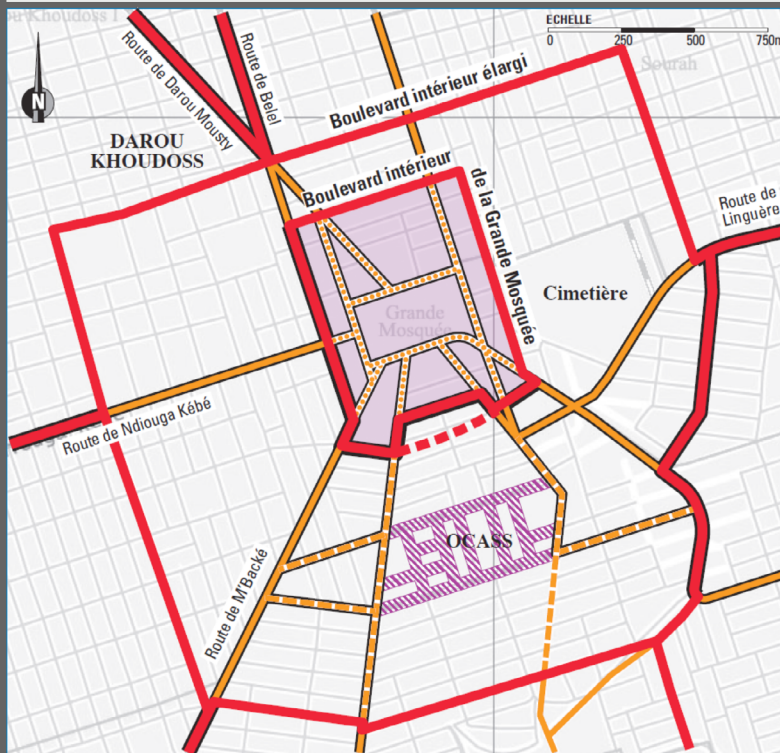
14:42

31:00

Concept général proposé en matière de transports publics.

À l'échelle du centre-ville, l'étude propose de créer un premier boulevard autour de la Grande Mosquée (boulevard intérieur), puis un second en périphérie, au-delà du marché OCASS (boulevard intérieur élargi) qui servirait de moyen de dérouter la circulation de l'hypercentre et de la zone de la mosquée (figure 5).

Organisation des circulations proposée au centre-ville (2012)



- Deux boulevards "intérieurs" protègent l'hypercentre.
- Le boulevard élargi qui permet de reprendre le trafic de transit local ainsi que le trafic à destination des marchés Ocass et Darou Khoudoss.
- Le boulevard de la Grande Mosquée qui constitue le dernier niveau de protection de la sérénité de la Grande Mosquée.
- Le marché Ocass reste accessible aux voitures privées et aux camions pour les livraisons et chargements, mais la circulation Est-Ouest y est coupée : les deux rues principales sont piétonnisées.

▶ Le "double boulevard" permet de mailler le réseau routier de l'hypercentre et propose une protection à deux niveaux et par deux effets (la diffusion et le contrôle d'accès)

LEGENDE

- Réseau existant à requalifier
- Réseau structurant principal
- Libération d'emprise nécessaire
- Réseau structurant secondaire
- Contrôle d'accès TC et autorisations
- Contrôle d'accès livraisons
- Espace à priorité TC et modes doux
- Voie piétonnisée

FIGURE 5

15:13

31:00

Organisation des circulations proposées au centre-ville.

À plus grande échelle et à moyen terme, un réseau de transports publics desservirait la ville le long de ses diagonales accompagnées de ceintures qui entourent l'hypercentre. Des solutions quant à la période de Grand Magal sont également proposées, avec une zone piétonne élargie au centre-ville et des zones de stationnement permettant de réduire la circulation en ville. Les déplacements hippomobiles se feraient sur les voies de desserte dans des zones délimitées par les réseaux de transports en commun et les ceintures entourant l'hypercentre. À long terme, de nouvelles zones d'activités verraient le jour à l'ouest et au sud-ouest (vers la commune de M'Backé), accompagnées de nouvelles ceintures de transport entourant le centre-ville et de plus en plus grands en fonction de la distance de ce dernier. Les transports publics seraient étendus jusqu'à la périphérie de la ville et une ligne ferait le tour de la première ceinture.

Le plan d'action développé à court-moyen terme porte non seulement sur le développement des transports, mais s'accompagne également de suggestions sur l'urbanisme, les infrastructures, le stationnement, le transport de marchandises, sur le volet institutionnel et de mise en œuvre et sur les aspects de communication.

LE MODULE SIG

Le module SIG développé dans cette étude a été développé en C#, en utilisant le composant TatukGIS pour la partie cartographique et le support d'une base de données MySQL. L'outil permet d'afficher les couches de base regroupées dans deux familles: Foncier (le titre foncier, les quartiers, les districts, etc.) et Infrastructures (les routes, les équipements scolaires, spirituels, etc.). Il permet également de consulter les données en cliquant directement dessus (figure 6). L'utilisateur peut modifier la symbologie de chacune des couches à sa guise.

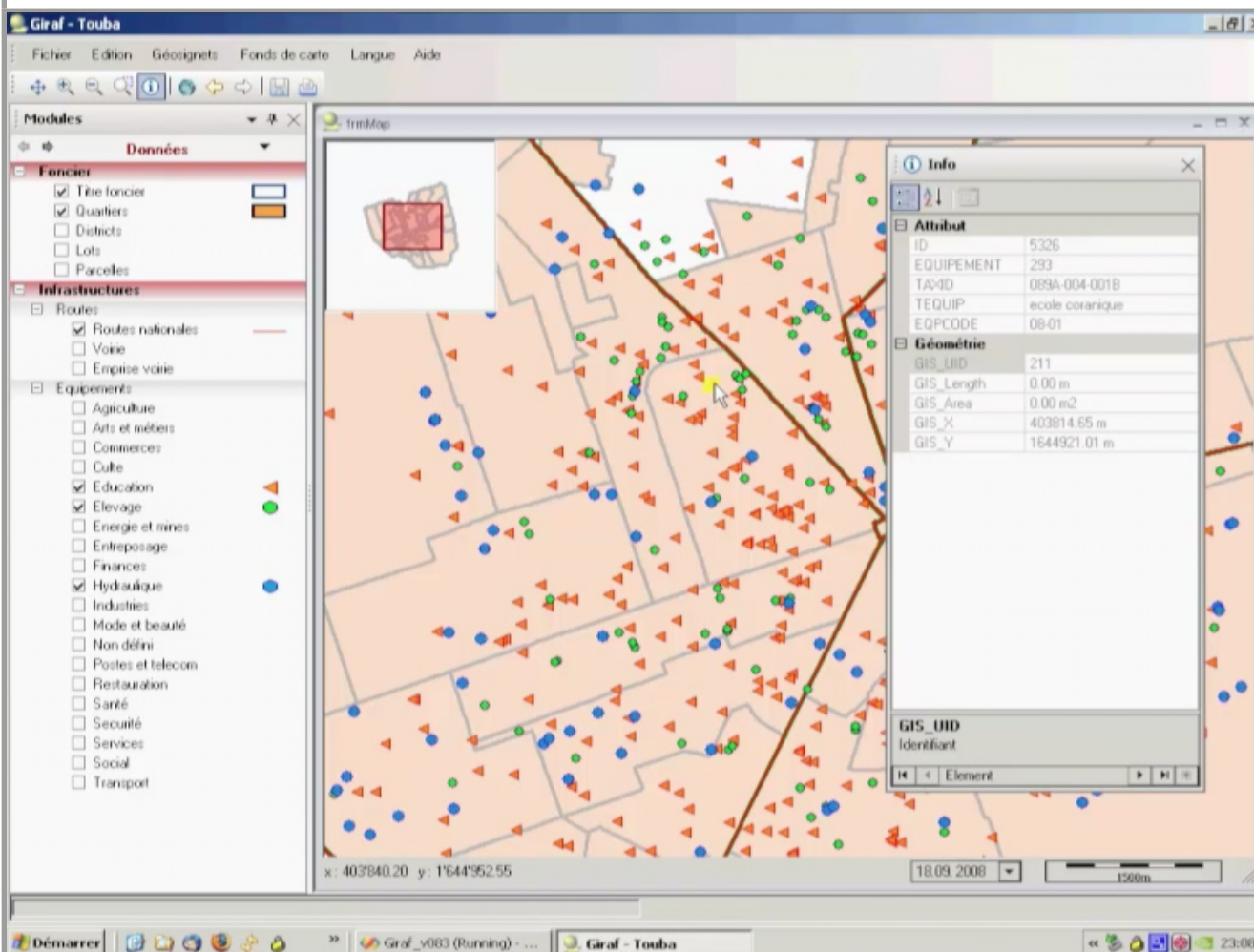


FIGURE 6

20:21

31:00

Consultation des données dans le SIG.

L'affichage des indices est très simple. Si l'on souhaite afficher les quartiers en leur appliquant une colorimétrie relative à leur superficie, il suffit de cliquer sur la couche et de sélectionner l'attribut à afficher (ici, l'aire). Il est possible d'afficher d'autres indicateurs, comme la densité d'habitation ou le pourcentage de maisons à étages par districts directement depuis le menu indicateurs de l'outil SIG. Ces cartes thématiques peuvent encore être configurées en modifiant le nombre de classes, les intervalles utilisés, les échelles de couleurs et la transparence.

Le module offre la possibilité d'éditer les données, par exemple d'ajouter un nouveau quartier. Dans ce cas, il est encore possible d'activer la fonction d'accrochage (snap en anglais) pour créer des polygones jointifs. Il permet également de modifier des données déjà existantes, par exemple pour mettre à jour les attributs d'un objet ou encore de supprimer un objet. Il existe deux modes d'édition. Le mode administrateur dans lequel les modifications de l'utilisateur se font directement dans la base de données, et les modifications des utilisateurs lambda qui sont enregistrées dans des couches provisoires. Les couches provisoires peuvent être affichées dans la barre des outils du logiciel et apparaissent en bleu pour les nouveaux objets, en jaune pour les objets modifiés et en rouge pour les objets effacés (figure 7).

Le module SIG

Edition de données (utilisateur)

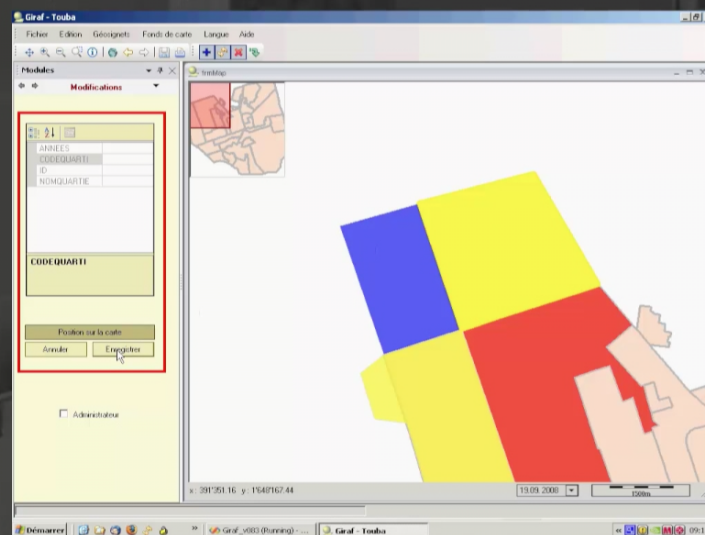


FIGURE 7

25:18

31:00

Interface utilisateur en enregistrement provisoire.

Pour accéder aux données du cadastre, il est nécessaire d'avoir un accès d'administrateur sur le logiciel. Il permet d'afficher toutes les informations sur une parcelle et son propriétaire.

L'étude montre qu'il n'a pas toujours été simple de joindre toutes les données du projet pour les déplacements et l'aménagement urbain. Chacune des couches peut être affichée selon son mode de transport, l'échéance du projet (court-moyen terme ou long terme), etc. Les couches peuvent être affichées en mode vecteur ou en téléchargeant le PDF de la carte du rapport. Il en va de même avec les couches raster. Finalement, le mode vectoriel peut être modifié et mis à jour, en modifiant par exemple la charge de trafic sur un tronçon particulier.

Finalement, l'outil tient compte de la temporalité des objets. Si l'on décide d'ajouter un objet à une date précise, puis de la modifier ou supprimer à une autre date, ces informations sont enregistrées dans la base de données et il est possible de voir le suivi temporel des différents objets.



20. REQUÊTES ET LANGAGE SQL

Cette leçon présente les bases du langage SQL ainsi que la clause SELECT.

LE SQL

L'acronyme **SQL** signifie *Structured Query Language*, ou langage de requêtes structurées en français. Il s'agit du langage destiné à la gestion des *bases de données relationnelles* et permet d'interagir directement avec les données. Il est normalisé et donc indépendant des systèmes de gestion de base de données (SGBD) à quelques exceptions près.

Le SQL est composé de quatre groupes d'instruction complémentaires :

- Le *Data Query Language* (DQL) permet d'interroger et d'extraire des données.
- Le *Data Definition Language* (DDL) permet de définir ou de modifier la structure d'une base de données.
- Le *Data Manipulation Language* (DML) permet d'insérer, modifier ou supprimer des données.
- Le *Data Control Language* (DCL) permet de gérer les droits d'accès à la base de données.

La syntaxe de base du SQL comporte des clauses de sélection (**SELECT**, **FROM**), de filtre conditionnel (**WHERE**), d'agrégation (**GROUP BY**, **HAVING**), de tri (**ORDER BY**, **LIMIT**) et de fusion (**UNION**, **INTERSECT**, **EXCEPT**).

Les exemples présentés utilisent les bases de données tests des Seychelles dont la préparation a fait l'objet de la leçon 18 (formats shapefile, Spatialite et PostGis).

Le tableau ci-dessous résume les attributs pour chacune des tables dans les bases de données.

TABLE	ATTRIBUTS
Districts	Géométrie, identifiant, nom
Hôtels	Géométrie, identifiant, nom, nombre de chambres, nombre de lits, statut, identifiant du district auquel l'hôtel appartient
Lieux-dits	Géométrie, identifiant, nom
Routes	Géométrie, identifiant, nom, lieu, type, surface

Résumé des attributs par table.

Il existe trois outils de requêtes applicables en général à tous les types de couche QGIS sans toutefois utiliser le langage SQL : le query builder, les requêtes sur la table d'attributs et l'extension Requête spatiale.

L'extension Gestion de base de données permet de traiter les bases de données Spatialite et PostGis.

Il existe également des solutions spécifiques à Spatialite (Sqlite Studio qui est hors QGIS, extension QSpatialite) et PostGis (pgAdmin qui est hors QGIS, extensions PostGisQueryBuilder, Traitements ou encore PostGis execute SQL).

Dans ces exemples, l'extension QSpatialite est utilisée pour la base de données Spatialite et le programme pgAdmin pour la gestion de la base de données PostGis.

SÉLECTION SIMPLE

Les deux termes à retenir pour la sélection d'un ou plusieurs attributs dans une base de données sont **SELECT** et **FROM** qui, respectivement, sélectionnent les attributs désirés à partir des tables mentionnées.

Dans QSpatialLite, si l'on veut afficher les noms de tous les hôtels, il faut écrire : **SELECT nom FROM hotels**. Puis cliquer sur le bouton **Run SQL** pour afficher le résultat. La figure 1 montre l'onglet de requête ainsi que celui du résultat pour cette requête.

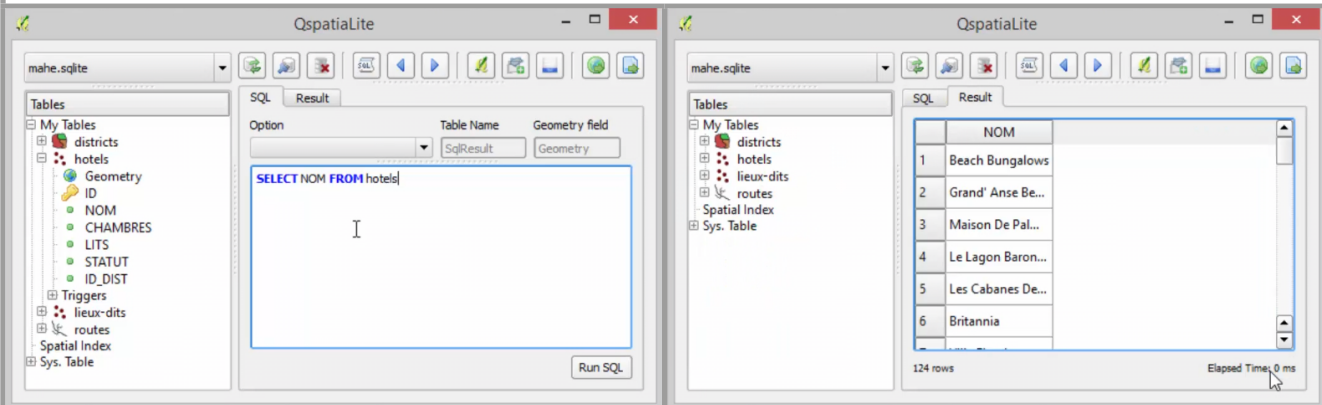


FIGURE 1

6:37

15:00

Interface de requête (à gauche) et résultat (à droite).

Il est possible de créer graphiquement des requêtes dans pgAdmin. Pour ce faire, faire glisser la table des hôtels dans le constructeur de requêtes, puis sélectionner l'attribut nom. En cliquant sur l'onglet **SQL Editor**, la requête a été créée de manière à lever toute ambiguïté. Dans le **Select**, le nom de la table précède le nom de l'attribut. Cela leverait le doute s'il y avait deux fois le même nom d'attribut dans deux tables différentes (figure 2).

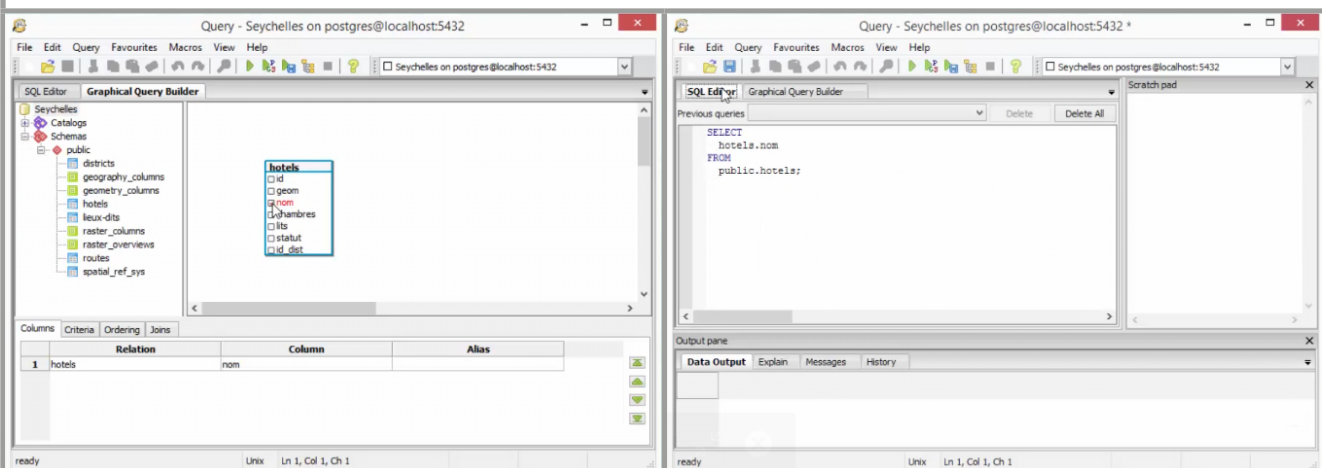


FIGURE 2

7:09

15:00

Créateur de SQL graphique de pgAdmin (à gauche) et interprétation du programme (à droite).

Dans le cas où l'on veut afficher plusieurs attributs d'une même table, il suffit d'écrire tous les noms des attributs après le mot-clé **SELECT** et de les séparer par des virgules. Le résultat affiche les différents attributs en colonne côtes à côtes. Dans le constructeur graphique de requêtes de pgAdmin, il suffit de sélectionner tous les attributs que l'on souhaite afficher.

Tous les attributs d'une table sont affichés si l'on utilise l'étoile après le mot-clé **SELECT** :

SELECT * FROM hotels



SÉLECTION DISTINCTE

Le mot-clé `DISTINCT` peut être placé après le `SELECT` afin de supprimer tous les doublons qui pourraient apparaître lors de l'exécution de la requête. En exécutant la requête `SELECT statut FROM hotels`, on s'aperçoit que beaucoup d'hôtels partagent les mêmes attributs. Or si l'on souhaite savoir quelles sont les valeurs existantes pour un attribut, les doublons sont inutiles. Ils peuvent être supprimés en effectuant la requête `SELECT DISTINCT statut FROM hotels`. Dans pgAdmin, il est nécessaire d'ajouter manuellement le mot-clé `DISTINCT` après le `SELECT`, car aucun raccourci ne permet de le faire depuis le constructeur graphique.

SÉLECTION DEPUIS DEUX TABLES

La force du langage SQL est qu'il peut afficher dans un même résultat des attributs de plusieurs tables différentes. Dans ce cas, il va falloir être vigilant et se prémunir contre toute ambiguïté lors de l'appel des attributs. Ainsi, une requête qui a pour but d'afficher deux attributs de deux tables différentes prendra la forme suivante:

```
SELECT nom_table1.nom_attribut1, nom_table2.nom_attribut2
FROM nom_table1, nom_table2
```

Par exemple, pour afficher l'association de tous les noms d'hôtel avec tous les noms des districts, il faut écrire:

```
SELECT hotels.nom, districts.nom
FROM hotels, districts
```

Dans le constructeur graphique de pgAdmin, le même résultat est obtenu en glissant les deux tables dans la zone de construction puis en cochant les deux attributs.

ALIAS

Il peut être utile de renommer le résultat affiché à l'aide d'un alias (`AS`). Cet alias donnera son nom à l'en-tête de la colonne dans le résultat. Cela s'effectue de la manière suivante:

```
SELECT nom_attribut AS nom_alias
FROM nom_table
```

Dans pgAdmin, l'alias peut être entré dans la troisième colonne sous l'onglet **Columns** de l'éditeur graphique.

L'alias peut également être utilisé pour modifier le nom des tables. Cette fois-ci, on n'utilise pas le mot-clé `AS`.

```
SELECT nom_alias1.nom_attribut1, nom_alias2.nom_attribut2
FROM nom_table1 nom_alias1, nom_table2 nom_alias2
```

On l'utilise dans ce cas pour simplifier la notation. Par exemple, on peut écrire la requête:

```
SELECT H.name, D.name
FROM hotels H, districts D
```

Dans le constructeur graphique de pgAdmin, il faut faire clic droit sur la table >> **Set Alias for table** et entrer l'alias de la table. Puis en retournant dans l'éditeur SQL, on voit bien que les tables ont des alias.

21. REQUÊTES CONDITIONNELLES

Cette leçon présente les requêtes conditionnelles destinées à extraire un sous-ensemble d'objets d'une base de données. Cela est nécessaire pour extraire les voitures rouges d'une base de données d'un parking, ou pour sélectionner les routes d'asphalte du réseau routier.

FILTRE CONDITIONNEL – LA CLAUSE **WHERE**

La syntaxe de base avec la clause **WHERE** est la suivante :

```
SELECT * FROM nom_table WHERE condition
```

Cette requête va afficher tous les attributs de la table vérifiant la *condition* imposée. Une condition est composée d'un attribut de la table, d'un opérateur (=, >, <=, etc.) et d'un critère.

Pour afficher la liste de tous les hôtels qui possèdent 20 lits, la condition s'écrit :

... **WHERE** hotels.lits = 20 Sur la liste des 124 hôtels que compte la table hotels, on extrait un sous-ensemble de 10 hôtels correspondant à la sélection.

OPÉRATEURS DE LA CLAUSE **WHERE**

Les opérateurs que l'on peut utiliser dans la clause **WHERE** sont résumés dans le tableau ci-dessous.

OPÉRATEUR	INTERPRÉTATION
=	Égal à (comparaison)
!=, <>	Différent de
<	Inférieur à
>	Supérieur à
<=	Inférieur ou égal à
>=	Supérieur ou égal à
IN (val1, val2, ...)	Dans la liste {val1, val2, ...}
BETWEEN val1 AND val2	Dans l'intervalle [val1, val2]
IS NULL	Valeur nulle
IS NOT NULL	Valeur non nulle
LIKE	Égal à (comparaison) pour une chaîne de caractères. En général non sensible à la casse
NOT LIKE	Ne contient pas

Opérateurs de la clause **WHERE**.

Quelques exemples de conditions. Ici on teste si la valeur de l'attribut :

- est supérieure ou égale à 35
... `WHERE nom_attribut >= 35`
- est contenu dans la liste des valeurs {val1, val2, val3}
... `WHERE nom_attribut IN (val1, val2, val3)`
- est contenu dans l'intervalle [val1, val2]
... `WHERE nom_attribut BETWEEN val1 AND val2`
- est nulle
... `WHERE nom_attribut IS NULL`
- est égale à la chaîne de caractères 'valeur'
... `WHERE nom_attribut LIKE 'valeur'`
- contient la chaîne de caractères 'valeurs'
... `WHERE nom_attribut LIKE '% valeur %'`

Remarque: L'opérateur `LIKE` n'est pas toujours insensible à la casse. C'est le cas dans PostgreSQL/PostGis, où `LIKE` et `=` sont équivalents (sensibles à la casse). L'opérateur `ILIKE` est insensible à la casse.

Les conditions peuvent être combinées pour effectuer des requêtes complexes. Lorsque deux conditions doivent être vérifiées, on utilise le mot-clé `AND` entre les deux conditions. Si seulement l'une d'entre elles doit être vérifiée, on peut utiliser le mot-clé `OR`. La requête suivante vérifie qu'au moins les conditions 1 et 2 sont vérifiées ou la troisième :

```
... WHERE (condition1 AND condition2) OR condition3
```

La requête suivante sélectionne le nom et le nombre de lits des hôtels dont le nom commence soit par B, soit par C et dont le nombre de lits est supérieur à 100:

```
SELECT hotels.nom, hotels.lits  
FROM hotels  
WHERE (hotels.nom LIKE 'B%' OR hotels.nom LIKE 'C%') AND hotels.lits > 100
```

Dans le constructeur graphique de pgAdmin, sélectionner les tables puis les attributs et cliquer sur l'onglet **Criteria**. Ajouter les conditions à appliquer sur la requête.

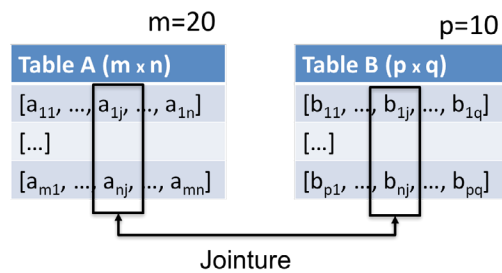
JOINTURES BASÉES SUR LA CLAUSE `WHERE`

Le principe des jointures est d'associer le contenu de plusieurs tables. Si l'on dispose de deux tables A et B, et que l'on souhaite les combiner en affichant toutes les combinaisons possibles, on peut écrire: `SELECT * FROM tableA, tableB`. Le résultat de cette requête est le *produit cartésien* de la table A avec la table B. Cela veut dire que toutes les lignes de B vont être successivement associées à chaque ligne de A. Si les tables A et B ont respectivement 20 et 10 lignes, alors le résultat comptera $20 \times 10 = 200$ lignes. Il s'agit du cas particulier du `CROSS JOIN`.

En règle générale, on effectue une jointure sur un champ commun aux deux tables, c'est-à-dire si des attributs de la table A apparaissent également dans la table B. En reprenant l'exemple d'une table A comportant 20 lignes et d'une table B en comportant 10, on peut se retrouver dans les situations suivantes.

- Si A et B ne possèdent aucun élément commun, alors si l'on affiche tous les résultats de A sauf ceux qui se retrouvent dans B (`AIB = NULL`), on affiche 20 lignes. De même, on affiche 10 lignes si l'on cherche tous les attributs de B qui ne figurent pas dans A (`BIA = NULL`).
- Si A et B possèdent 5 éléments communs, alors `AIB=NULL` affiche 15 résultats et `BIA = NULL` en affiche 5
- Si A et B possèdent 10 éléments communs, alors `AIB=NULL` affiche 10 résultats et `BIA = NULL` en affiche 0

Ces trois cas sont illustrés à la figure 1.



3 groupes

Table Jointe			
A ↔ B	0	5	10
A B=NULL	20	15	10
B A = NULL	10	5	0
Nb de lignes	30	25	20

FIGURE 1

14:31

28:00

Illustration des trois cas de jointure.

La syntaxe générale d'une jointure basée sur la clause **WHERE** est :

```
SELECT T1.nom_attribut1, T2.nom_attribut2
FROM nom_table1 T1, nom_table2 T2
WHERE T1.nom_attribut3 = T2.nom_attribut4
```

Supposons que l'on souhaite afficher pour chaque hôtel son nom, son nombre de lits et le nom du district auquel il appartient. Il faut afficher du contenu de deux tables: hotels et districts et faire une jointure sur l'attribut de l'identifiant du district, disponible dans les deux tables. Voici à quoi doit ressembler la requête :

```
SELECT H.nom, H.lits, D.nom
FROM hotels H, districts D
WHERE H.id_dist = D.id
```

Dans le constructeur graphique de pgAdmin, une jointure entre deux attributs est réalisée en cliquant sur l'attribut de la première table et en maintenant cliqué jusque sur l'attribut de la deuxième table. Une liaison symbolisée par un trait noir apparaît entre les deux tables (figure 2).

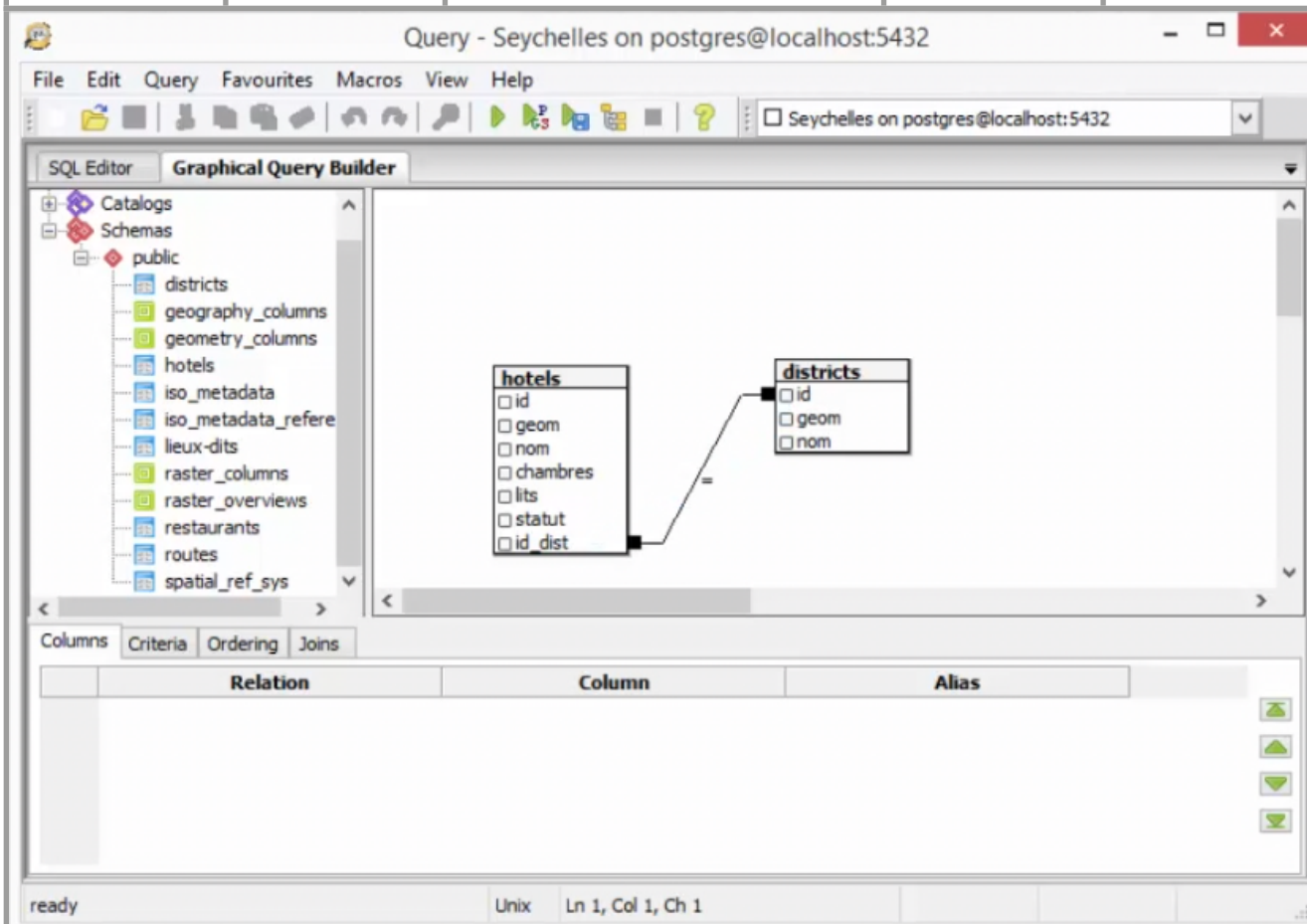


FIGURE 2

17:54

28:00

Jointure attributaire entre deux tables dans le composeur graphique de requêtes de pgAdmin.

JOINTURES BASÉES SUR LA CLAUSE JOIN

La clause `JOIN` permet également d'effectuer des *jointures* entre deux tables. Cette fois-ci, le mot-clé `JOIN` doit être placé après le `FROM`. Il est suivi du mot-clé `ON` après lequel on écrit la condition de jointure :

```
SELECT nom_table1.nom_attribut1, nom_table2.nom_attribut2
FROM nom_table1 JOIN nom_table2
ON nom_table1.nom_attribut3 = nom_table2.nom_attribut4
```

On préfère utiliser la syntaxe avec `JOIN` quand on écrit des requêtes complexes pour des raisons de clarté et de lisibilité.

Il existe plusieurs types de jointures. Considérons les tables 1 et 2 et que nous effectuons les différents `JOIN` entre elles. On écrit table 1 ... table 2 (remplacer le ... par le type de `JOIN` désiré).

- `(INNER) JOIN`: correspond à l'intersection des deux tables.
- `LEFT (OUTER) JOIN`: prend tous les éléments de la table de gauche (table 1) ainsi que l'intersection entre les deux tables. La valeur `NULL` est attribuée aux attributs manquants de la table 2.
- `RIGHT (OUTER) JOIN`: prend tous les éléments de la table de droite (table 2) ainsi que l'intersection entre les deux tables. La valeur `NULL` est attribuée aux attributs manquants de la table 1.
- `FULL (OUTER) JOIN`: prend tous les éléments de la table de gauche (table 1) ainsi que l'intersection entre les deux tables. La valeur `NULL` est attribuée aux attributs manquants dans les deux tables.

Remarque: La syntaxe de la requête reste identique selon le type de jointure que l'on souhaite effectuer :

```
SELECT nom_table1.nom_attribut1, nom_table2.nom_attribut2
FROM nom_table1 LEFT JOIN nom_table2
ON nom_table1.nom_attribut3 = nom_table2.nom_attribut4
```



Exemple : On souhaite afficher le nom des hôtels et le nom du district auquel il appartient :

```
SELECT hotels.nom, districts.nom  
FROM hotels JOIN districts  
ON hotels.id_dist = districts.id
```

Dans le constructeur graphique de requêtes de pgAdmin, la jointure peut être spécifiée dans l'onglet **Joins**. En revanche, dans l'éditeur SQL, on voit que le programme a traduit la jointure dans une clause **WHERE**. La requête peut être modifiée manuellement pour faire la jointure à l'aide de la clause **JOIN**.

Si l'on désire effectuer un **LEFT JOIN** exclusif (sans l'intersection), il est nécessaire d'ajouter une condition sur les attributs non disponibles dans la table de droite :

```
SELECT nom_table1.nom_attribut1, nom_table2.nom_attribut2  
FROM nom_table1 LEFT JOIN nom_table2  
ON nom_table1.nom_attribut3 = nom_table2.nom_attribut4  
WHERE nom_table2.nom_attribut4 IS NULL
```

Il en va de même pour la jointure exclusive à droite (**RIGHT JOIN**), à l'exception que la condition porte sur l'attribut 3 de la table 1. Et pour un **FULL JOIN** auquel on souhaite retirer l'intersection, il faut ajouter deux conditions séparées par le mot-clé **OR** (l'un ou l'autre doit être **NULL**), comme le montre la requête ci-dessous :

```
SELECT nom_table1.nom_attribut1, nom_table2.nom_attribut2  
FROM nom_table1 FULL JOIN nom_table2  
ON nom_table1.nom_attribut3 = nom_table2.nom_attribut4  
WHERE nom_table1.nom_attribut3 IS NULL OR nom_table2.nom_attribut4 IS NULL
```



22. REQUÊTES D'AGRÉGATION ET TRI DU RÉSULTAT

Cette leçon montre comment extraire de l'information synthétique d'un ensemble de données et comment trier et regrouper les résultats d'une requête.

FONCTIONS D'AGRÉGATION

Les fonctions d'agrégation permettent de calculer des statistiques sur un ensemble de données. La fonction vient directement s'insérer dans la clause `SELECT`, et prend un attribut entre parenthèses. Voici sa syntaxe :
`SELECT fonction(nom_attribut) FROM nom_table`

La fonction peut être :

- `sum()` pour la somme ;
- `count()` pour compter le nombre d'éléments ;
- `avg()` pour la moyenne ;
- `min()` pour le minimum ;
- `max()` pour le maximum.

`count(*)` retourne donc le nombre d'enregistrements d'une table et `count(DISTINCT(nom_attribut))` le nombre de valeurs différentes de l'attribut.

Pour savoir combien de chambres d'hôtel comptent les Seychelles, on peut écrire la requête suivante :

```
SELECT sum(hotels.chambres)
FROM hotels
```

Si l'on est intéressé à connaître le nombre de lits moyens par hôtel, un moyen d'y parvenir est de compter le nombre de lits total et de le diviser par le nombre d'hôtels. Comme ces deux arguments sont formatés en nombres entiers dans la base de données, on peut obtenir le ratio en réel en utilisant la fonction `cast()`.

```
SELECT cast(sum(hotels.lits) AS FLOAT)/count(hotels.lits)
FROM hotels
```

Dans pgAdmin, les fonctions d'agrégation ne peuvent pas être implémentées dans le constructeur graphique, mais doivent être écrites à la main dans l'éditeur SQL. Par contre, la division dans ce logiciel de gestion de base de données donne directement un résultat en nombre réel.

REGROUPEMENT

L'objectif du **regroupement** est de calculer de fonctions d'agrégation sur des groupes de données plutôt que sur toutes les données d'une table. La clause de regroupement (selon un attribut) est le `GROUP BY`.

Voici la syntaxe d'un `GROUP BY` :

```
SELECT nom_attribut1, fonction(nom_attribut2)
FROM nom_table
GROUP BY nom_attribut1
```

Pour afficher chaque district avec son nombre de chambres d'hôtels, on peut écrire la requête :

```
SELECT districts.nom, sum(hotels.chambres)
FROM districts, hotels
WHERE hotels.id_dist = districts.id
GROUP BY districts.nom
```

AGRÉGATION CONDITIONNELLE

La clause d'**agrégation conditionnelle** est le **HAVING**. Elle agit de manière similaire à la clause **WHERE** à l'exception qu'elle regroupe / filtre les résultats de la condition. Sa syntaxe est :

```
SELECT nom_attribut1, fonction(nom_attribut2)
FROM nom_table
GROUP BY nom_attribut1
HAVING fonction(nom_attribut2) opérateur critère
```

Si l'on veut afficher le nom des districts qui ont plus de 100 chambres, on peut exécuter la requête suivante :

```
SELECT districts.nom, sum(hotels.chambres)
FROM districts, hotels
WHERE hotels.id_dist = districts.id
GROUP BY districts.name
HAVING sum(hotels.chambres) > 100
```

À nouveau, le constructeur graphique de pgAdmin n'est d'aucune utilité pour effectuer l'agrégation conditionnelle. Par contre, il est possible de le faire depuis le constructeur graphique du logiciel Navicat.

TRI ET LIMITATION DU RÉSULTAT

La clause **ORDER BY** permet de *trier* les lignes dans un résultat et **LIMIT** de *limiter* le nombre d'éléments à afficher.

La clause **ORDER BY** classe les lignes d'un résultat de manière ascendante (**ASC**) ou descendante (**DESC**). Par défaut, l'ordre de tri est ascendant. Elle peut être appliquée sur plusieurs colonnes, comme le montre l'exemple de la syntaxe ci-dessous :

```
SELECT nom_attribut1, nom_attribut2, ...
FROM nom_table
ORDER BY nom_attribut1 DESC, nom_attribut2 ASC
```

La clause **LIMIT** permet d'indiquer le nombre maximum de résultats à afficher. Si l'on souhaite afficher d'autres lignes que les premières, il est possible d'utiliser le mot-clé **OFFSET** à cet effet. Voici sa syntaxe :

```
SELECT nom_attribut1
FROM nom_table
LIMIT nb_ligne1 OFFSET nb_ligne2
```

Pour afficher les 5 hôtels qui ont le plus grand nombre de chambres dans un ordre décroissant, on peut exécuter la requête suivante :

```
SELECT hotels.nom, hotels.lits.
FROM hotels
ORDER BY hotels.lits DESC, hotels.nom ASC
LIMIT 5
```

Le constructeur graphique de requêtes de pgAdmin possède un onglet **Ordering** pour gérer le tri des résultats. Par contre, la limitation doit être indiquée à la main dans l'éditeur de SQL.

23. FUSION DE REQUÊTES, REQUÊTES EMBOÎTÉES

Cette leçon présente la fusion de requêtes, que l'on utilise pour afficher le contenu en commun de deux (ou plusieurs) requêtes ou pour retirer du résultat d'une requête le groupe de résultat d'une autre requête. Elle présente également le principe des requêtes emboîtées, qui permettent par exemple d'effectuer des comparaisons d'attributs par rapport à la moyenne. Ces requêtes peuvent être placées dans les clauses `SELECT`, `FROM`, `WHERE` et `HAVING`.

FUSION DE REQUÊTES

Les clauses de fusion utilisent les mots-clés `UNION`, `INTERSECT` et `EXCEPT`.

L'`UNION` met bout à bout les résultats de plusieurs requêtes. Il est nécessaire que les résultats assemblés aient le même nombre et type de colonnes et ordonnées de la même manière. La syntaxe est la suivante :

```
SELECT * FROM nom_table1 UNION  
SELECT * FROM nom_table2
```

Remarques :

- Bien que l'effet de l'`UNION` semble similaire ici à celle du `FULL JOIN` décrite à la leçon 21, la différence est de taille. Une fusion ne peut être faite qu'entre deux (ou plus) résultats organisés de manière identique (ordre, type et nombre des colonnes).
- Par défaut, le mot-clé `UNION` supprime les doublons qui se trouvent dans les deux résultats. Le mot-clé `UNION ALL` en revanche ne les supprime pas.

Le mot-clé `INTERSECT` permet d'afficher le contenu identique aux résultats de deux requêtes. À nouveau, il est primordial que les deux résultats aient le même nombre et type de colonnes et qu'elles soient disposées dans le même ordre. Sa syntaxe est similaire à l'`UNION`.

```
SELECT * FROM nom_table1 INTERSECT  
SELECT * FROM nom_table2
```

Comme pour l'`UNION`, l'`INTERSECT` possède un type de jointure qui lui ressemble. Il s'agit du `JOIN`. Même première remarque que pour l'`UNION`.

Le mot-clé `EXCEPT` (ou `MINUS`) permet de conserver les résultats de la première requête qui ne sont pas dans le résultat de la seconde. Encore une fois, l'`EXCEPT` ne peut être réalisé que sur deux résultats de requêtes possédant les mêmes colonnes (types, nombre) et dans le même ordre.

```
SELECT * FROM nom_table1 INTERSECT  
SELECT * FROM nom_table2
```

Comme pour l'`UNION` et l'`INTERSECT`, l'`EXCEPT` possède un type de jointure qui lui ressemble. Il s'agit du `LEFT JOIN` (sans intersection). Même première remarque que pour l'`UNION`.

Pour associer (sans doublon) les noms des districts et des lieux-dits dont la première lettre est un T, on peut exécuter la requête suivante :

```
SELECT districts.nom  
FROM districts  
WHERE districts.nom LIKE 'T%'  
UNION  
SELECT "lieux-dits".nom  
FROM "lieux-dits"  
WHERE "lieux-dits".nom LIKE 'T%'
```

Ici, les guillemets servent à ôter toute ambiguïté relative au tiret entre lieux et dits.

REQUÊTES EMBOÎTÉES DANS LA CLAUSE WHERE

Syntaxe générale :

```
SELECT nom_attribut1
FROM nom_table1
WHERE nom_attribut2 = (SELECT nom_attribut3
                       FROM nom_table2
                       WHERE nom_attribut3 = critère)
```

Le critère de la requête extérieure est remplacé par la requête emboîtée.

Pour afficher le nom de l'hôtel qui possède plus de lits que la moyenne, on peut utiliser la requête suivante :

```
SELECT hotels.nom, hotels.lits
FROM hotels
WHERE hotels.lits = (SELECT avg(hotels.lits)
                   FROM hotels)
```

REQUÊTES EMBOÎTÉES DANS LA CLAUSE FROM

Syntaxe générale :

```
SELECT nom_attribut1
FROM (SELECT nom_attribut2
      FROM nom_table1
      WHERE nom_attribut3 = critère)
WHERE nom_attribut2 = critère
```

Il est possible d'emboîter une requête dans la clause **FROM**, car le résultat de cette requête agit comme une table source pour la requête extérieure.

Une alternative à la requête emboîtée dans la clause **WHERE** pour afficher la liste des hôtels qui possèdent plus de lits que la moyenne est donnée ici :

```
SELECT h1.nom, h1.lits
FROM hotels h1, (SELECT avg(h2.lits) AS avg_lits
                FROM hotels h2) s
WHERE h1.lits > s.avg_lits
```

FUSION ET OPÉRATEURS IN ET NOT IN

Les opérateurs **IN** et **NOT IN** sont placés dans la clause **WHERE**. Ils permettent de comparer un attribut dans la liste qui est en fait le résultat de la requête emboîtée. La syntaxe est :

```
SELECT nom_attribut1
FROM nom_table1
WHERE nom_attribut2 IN (SELECT nom_attribut3
                      FROM nom_table 2
                      WHERE nom_attribut3 = critère)
```

Pour chercher les districts qui ont le même nom que les lieux-dits, on peut exécuter la requête suivante :

```
SELECT districts.nom
FROM districts
WHERE districts.nom IN (SELECT "lieux-dits".nom
                       FROM "lieux-dits")
```

24. DDL, DML, VUES

Cette leçon porte principalement sur le *data definition language* (DDL) et le *data manipulation language* (DML). Tous les outils de création, d'édition et de suppression sont en fait des implémentations de requêtes qui utilisent le SQL.

SQL VS STRUCTURATION ET ÉDITION DE DONNÉES

Dans la table d'attributs d'une couche, QGIS offre la possibilité d'éditer les données. La figure 1 montre l'exemple de la création d'une nouvelle colonne. Il en va de même sur SpatialLite et PostGis/Postgres.

Les systèmes de gestion de base de données (SGBD) offrent également cette possibilité. Par un clic droit sur la couche >> **New Column...** . Les outils CASE permettent eux aussi de créer et éditer des tables. Toutes ces fonctionnalités sont basées sur des requêtes SQL préimplémentées.

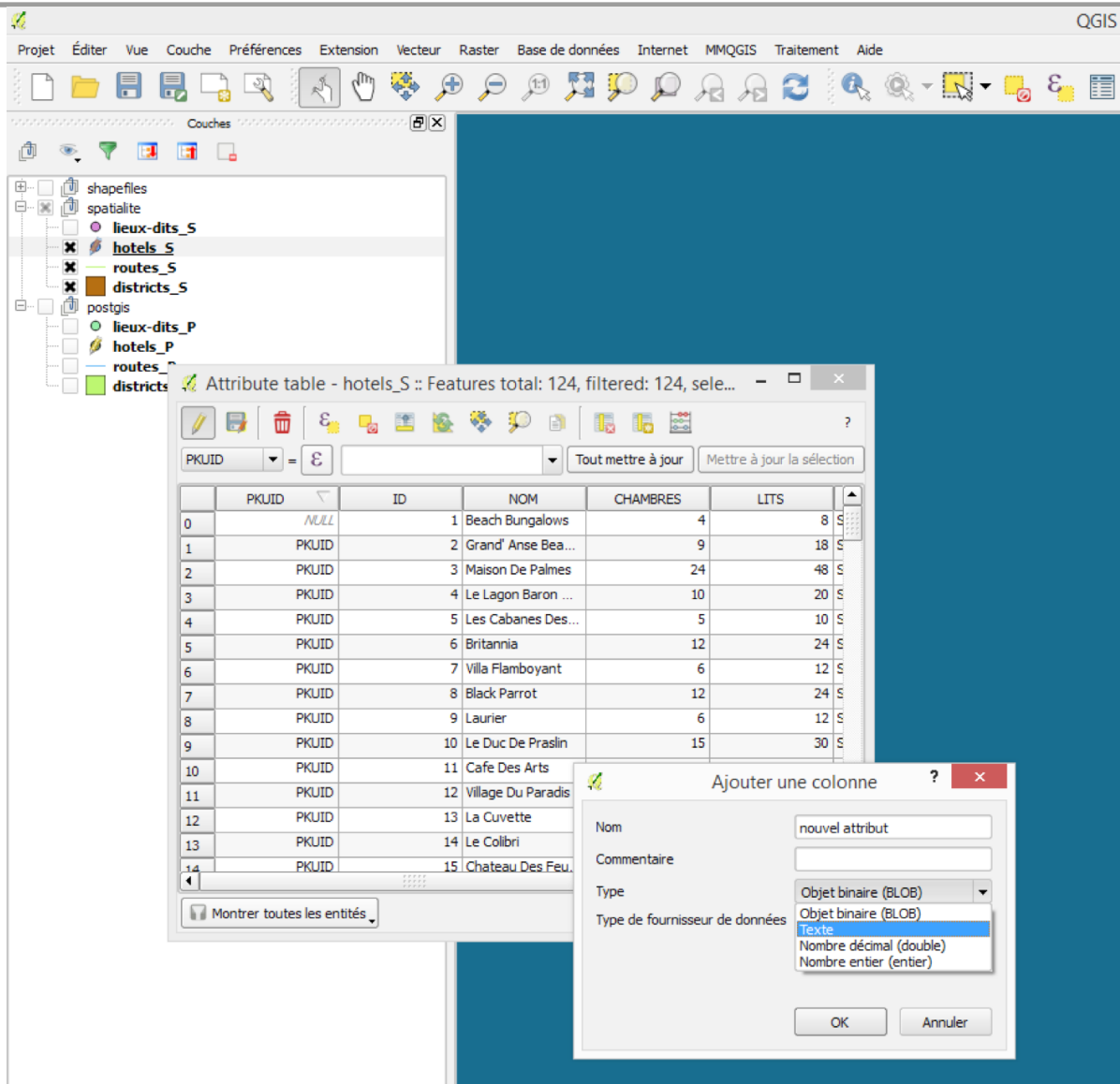


FIGURE 1

DÉFINITION DE DONNÉES

CREATE TABLE

L'instruction `CREATE TABLE` permet de créer des tables. Il est nécessaire de définir ses colonnes avec un nom et le type de données qu'elles contiendront. En voici la syntaxe.

```
CREATE TABLE nom_table (
    colonne_1 type_donné1,
    colonne_2 type_donnée2,
    ...
)
```

Les types de données diffèrent d'un SGBD à un autre.

Créons d'une nouvelle table pour stocker les informations des employés des hôtels. On veut enregistrer trois informations: l'identifiant unique de l'employé (integer), son nom (texte) et l'identifiant unique (integer) de l'hôtel dans lequel il travaille. Voici une requête possible dans QSpatialite:

```
CREATE TABLE employés(
    ID integer,
    nom text,
    hotel_id integer,
    PRIMARY KEY(ID),
    FOREIGN KEY(hotel_id) REFERENCES hotel(id)
)
```

Il est possible de créer graphiquement une telle table dans pgAdmin. Cliquez droit sur la base de données >> **New Table...** puis définir le nom de la table. Aller dans l'onglet **Columns** et cliquer sur **Add** pour ajouter une nouvelle colonne. Donner son nom (ID, nom, hotel_id) ainsi que le type de données qu'elle va contenir (integer, text, integer). Pour ajouter les contraintes, entrer dans l'onglet **Constraints**, sélectionner le type de contrainte (Primary Key ou Foreign Key) puis cliquer sur **Add**. Entrer dans l'onglet **Column** de la fenêtre qui s'ouvre pour sélectionner la colonne à laquelle on souhaite appliquer la contrainte.

L'instruction `CREATE TABLE AS` permet d'enregistrer le résultat d'une requête dans une nouvelle table:

```
CREATE TABLE nom_table AS requête
```

Il existe cependant une alternative qui utilise l'instruction `SELECT INTO`:

```
SELECT nom_attributs INTO nom_table2 FROM nom_table1
```

Pour enregistrer dans une nouvelle table le nombre d'hôtels correspondant à chaque statut, on peut utiliser la requête suivante:

```
CREATE TABLE hotel_statut AS
SELECT statut, count(*) as nb_hotels
FROM hotels
GROUP BY hotels.statut
```

Dans pgAdmin, il est possible de créer la requête graphiquement, mais il est nécessaire d'ajouter manuellement le `CREATE TABLE AS` dans l'éditeur SQL.

Remarque: on aurait également pu utiliser la requête suivante, avec l'instruction `SELECT INTO` pour obtenir un résultat identique.

```
SELECT hotels.statut, count(*)
INTO hotel_statut
FROM hotels
GROUP BY hotels.statut
```

ALTER TABLE

L'instruction `ALTER TABLE` permet de modifier la structure d'une table en y ajoutant ou supprimant des colonnes ou en modifiant le nom ou le type d'une colonne. La syntaxe de base est donnée par : `ALTER TABLE nom_table instruction`

Voici les instructions possibles :

- `ADD` pour ajouter une colonne
`ALTER TABLE nom_table ADD nom_colonne type_donnée`
- `DROP` pour supprimer une colonne
`ALTER TABLE nom_table DROP nom_colonne`
- `ALTER COLUMN TYPE` pour modifier le nom et le type d'une colonne
`ALTER TABLE nom_table ALTER nom_colonne TYPE type_donnée`

Reprenons la table des employés que nous avons créée à l'instant et ajoutons-y une colonne contenant les prénoms des employés.

```
ALTER TABLE employés  
ADD COLUMN prénom text
```

L'interface de création graphique de pgAdmin permet de créer une nouvelle colonne (clique droit sur la table >> **New Column...**). Entrer son nom et son type dans l'onglet **Propriétés**.

DROP TABLE, TRUNCATE TABLE

L'instruction `DROP TABLE` supprime la table tandis que `TRUNCATE TABLE` en efface le contenu tout en conservant sa structure.

- `DROP TABLE nom_table`
- `TRUNCATE TABLE nom_table`

Pour supprimer la table `hotel_statut` créée précédemment :

```
DROP TABLE hotel_statut
```

MODIFICATION DE DONNÉES

INSERT INTO

L'instruction `INSERT INTO` permet d'insérer des données dans une table. Il existe plusieurs possibilités d'utilisation :

- Insertion une ligne à la fois
`INSERT INTO nom_table VALUES ('valeur1', 'valeur2', ...)`
- Insertion en spécifiant les colonnes concernées
`INSERT INTO nom_table (nom_colonne1, nom_colonne2, ...) VALUES ('valeur1', 'valeur2', ...)`
- Insertion de plusieurs lignes à la fois
`INSERT INTO nom_table VALUES`
`('valeur1', 'valeur2', ...),`
`('valeur1', 'valeur2', ...),`
`('valeur1', 'valeur2', ...)`

Pour insérer un champ de type text, il faut entourer la valeur de guillemets. Ceci n'est pas nécessaire quand la valeur est numérique.

Pour ajouter une entrée dans la table employée précédemment, on peut utiliser par exemple :

```
ALTER TABLE employés  
INSERT INTO (1, 'Jean')
```

Cette requête a inséré 1 dans l'attribut ID (clé primaire) et 'Jean' dans l'attribut nom. L'attribut `hotel_id` est resté vide, car non spécifié.

La gestion de la clé primaire peut devenir très embarrassante lorsque l'on gère de nombreuses données. Pour y remédier et attribuer automatiquement une valeur unique à chacune des nouvelles entrées, on peut déclarer le type de la colonne ID avec le terme `serial`. En faisant ainsi, il est possible d'ajouter plusieurs lignes sans se soucier de la clé primaire qui est automatiquement attribuée. Voici un exemple (après avoir modifié le type de données de la colonne ID) avec la table des employés :

```
INSERT INTO employés (nom)
SELECT hotels.nom
FROM hotels
WHERE hotels.nom LIKE 'Chez%'
```

UPDATE

L'instruction `UPDATE` permet de mettre à jour des lignes ou des *tuples* existants. Elle est généralement associée à la clause `WHERE` pour préciser la ligne à modifier :

```
UPDATE nom_table SET nom_colonne = 'valeur' WHERE condition
```

On peut par exemple ajouter l'identifiant de l'hôtel où travaille Jean dans la base de données de la façon suivante :

```
UPDATE employés
SET hotel_id = 6
WHERE nom = 'Jean'
```

DELETE FROM

L'instruction `DELETE FROM` supprime des lignes ou des *tuples*. Elle est également souvent associée à la clause `WHERE` pour sélectionner la ligne à supprimer :

```
DELETE FROM nom_table WHERE condition
```

Si l'on veut supprimer la ligne de Jean dans la base de données, on peut exécuter la requête suivante :

```
DELETE FROM employés
WHERE nom = 'Jean'
```

VUES

Les vues sont des tables virtuelles non éditables qui permettent par exemple d'afficher les résultats de requêtes emboîtées. Voici leur syntaxe :

- Créer une vue :
`CREATE VIEW nom_vue AS requête`
- Supprimer une vue :
`DROP VIEW nom_vue`

Une vue est enregistrée dans le menu **Views** dans pgAdmin.



25. REQUÊTES SPATIALES GÉOMÉTRIQUES

Les *requêtes spatiales* s'intéressent à la géométrie des objets contenus dans une base de données. Elles permettent par exemple d'obtenir les coordonnées d'un point, la longueur d'un segment ou encore la surface d'un polygone. Il existe une très grande variété de fonctions spatiales. Les fonctions principales sont décrites ici pour les systèmes de gestion de base de données SQLite et PostGis. Un inventaire des fonctions est disponible sous <http://www.gaia-gis.it/gaia-sins/Spatialite-sql-4.2.0.html> pour SQLite et sous <http://PostGis.net/docs/index.html> pour PostGis.

La syntaxe des requêtes spatiales est quasi identique à celle des requêtes non spatiales. PostGis différencie bien les fonctions non spatiales des fonctions spatiales qui sont précédées du préfixe ST_ (spatial type).

GÉOMÉTRIE ET RÉFÉRENCE SPATIALE

L'attribut contenant la géométrie spatiale dans une table est généralement nommé *geometry*. Il contient entre autres les coordonnées de l'objet et la référence du système de projection utilisé.

Pour afficher les objets sur la carte, on peut utiliser la requête spatiale :

```
SELECT geometry FROM nom_table
```

Pour extraire la référence du système de projection (code *ESPG*) de chaque objet :

```
SELECT SRID(geometry) FROM nom_table
```

Par exemple, pour afficher tous les districts dont le nom commence par B dans QGIS, on peut ouvrir QSpatialite et entrer la requête suivante :

```
SELECT districts.geometry  
FROM districts  
WHERE districts.nom LIKE 'B%'
```

Finalement, pour afficher la carte sur QGIS, il faut charger le résultat de la requête de QSpatialite dans QGIS. Pour ce faire, sélectionner l'option **Load in QGIS as Spatial Layer**, donner un nom à la table et cliquer sur **Run SQL**. La nouvelle couche apparaît dans le gestionnaire des couches de QGIS.

CONVERSION

Les fonctions `AsText()` et `AsBinary()` permettent de convertir la géométrie dans un format texte (WKT) ou binaire (WKB). Elles facilitent le partage d'une base de données spatiale. Voici leur syntaxe :

– Format texte :

```
SELECT AsText(geometry) FROM nom_table
```

– Format binaire :

```
SELECT AsBinary(geometry) FROM nom_table
```

La fonction `AsText()` renvoie une liste de multipoints (série de coordonnées) tandis que la fonction `AsBinary()` renvoie un objet géométrique. Les résultats peuvent être exportés comme tables dans QGIS ou dans QSpatialite. À l'inverse, pour convertir un format texte en géométrie, on utilise la fonction `GeomFromText()` et la fonction `GeomFromBinary()` si l'entrée est au format binaire.

TYPE DE GÉOMÉTRIE

Il existe des fonctions pour nous renseigner sur le type et les propriétés de la géométrie. Les fonctions sont résumées dans le tableau ci-dessous.

FONCTION SPATIALE	EFFET
<code>GeometryType()</code>	Géométrie de l'objet
<code>NPoints()</code>	Nombre de points
<code>NRings()</code>	Nombre d'anneaux
<code>Envelope()</code>	Enveloppe extérieure

Fonctions d'information sur le type et les propriétés de la géométrie.

La syntaxe de ces fonctions est toujours:

```
SELECT Fonction(geometry) FROM nom_table
```

Par exemple, dans QSpatialite la requête suivante renvoie une liste de 25 multipolygones:

```
SELECT GeometryType(geometry) FROM districts
```

Pour afficher les enveloppes des districts dans QGIS à partir de QSpatialite, on peut exécuter la requête suivante:

```
SELECT Envelope(geometry) AS geometry FROM districts
```

Avant de cliquer sur **Run SQL**, il faut sélectionner l'option **Load in QGIS as Spatial Layer**, nommer la table et préciser que le champ contenant la géométrie dans la nouvelle table est `geometry`. En cliquant sur **Run SQL**, la couche apparaît dans QGIS et donne le résultat visible à la figure 1.

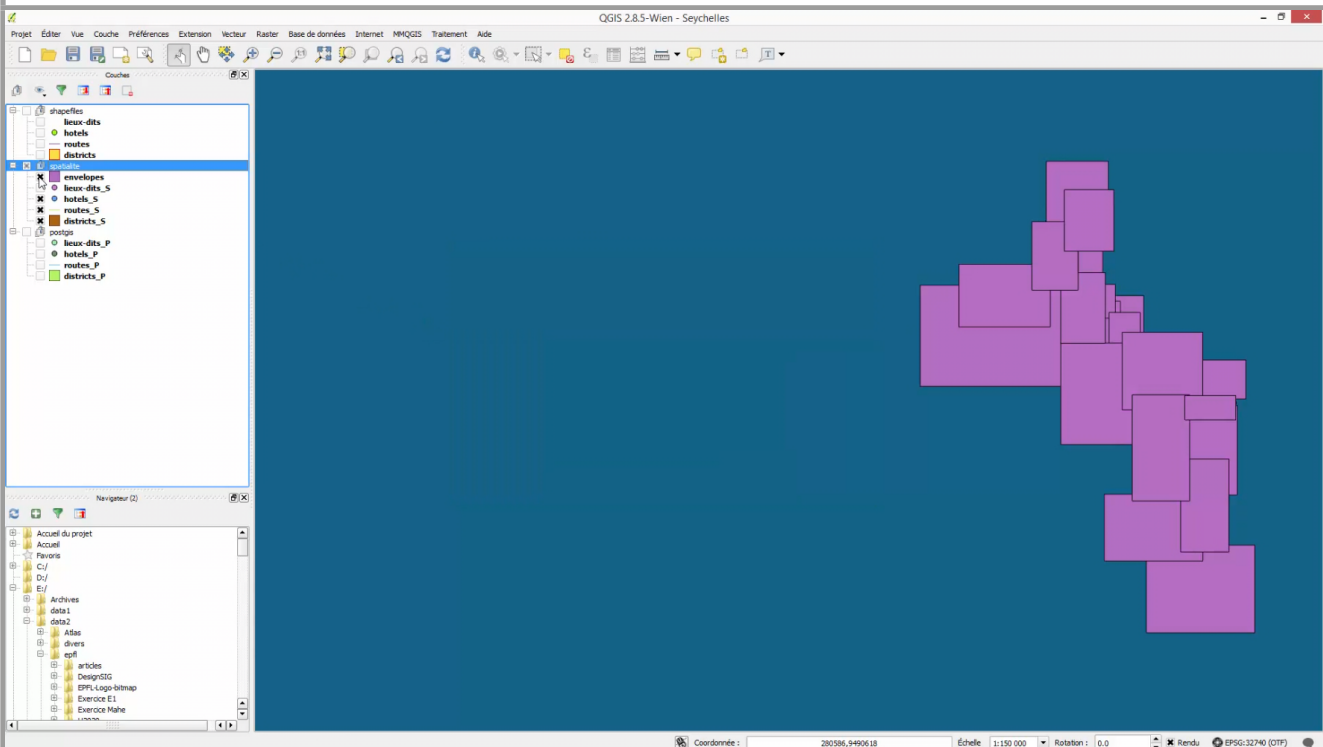


FIGURE 1

10:39

16:00

Enveloppes des districts.



PROPRIÉTÉS DE LA GÉOMÉTRIE

Les principales fonctions qui renseignent sur les propriétés de la géométrie sont listées dans le tableau ci-dessous.

FONCTION SPATIALE	EFFET
<code>X(), Y()</code>	Extrait les coordonnées x et y d'un point
<code>StartPoint()</code>	Retourne le point initial d'une polyligne
<code>EndPoint()</code>	Retourne le point final d'une polyligne
<code>Length()</code>	Retourne la longueur d'une polyligne. Cette fonction est appelée <code>GLength()</code> dans SpatiaLite et <code>ST_Length()</code> dans PostGis
<code>isClosed()</code>	Permet de savoir si une polyligne est fermée
<code>isRing()</code>	Permet de savoir si une polyligne constitue un contour externe ou interne d'un polygone.
<code>Centroid()</code>	Donne le barycentre d'un polygone
<code>Area()</code>	Retourne la surface d'un polygone

Principales fonctions renseignant sur les propriétés de la géométrie.

QUELQUES EXEMPLES :

- Pour calculer la longueur des tronçons des routes aux Seychelles, on utilise la requête suivante :
`SELECT Length(routes.geometry) FROM routes`
- Pour calculer la surface des districts :
`SELECT Area(districts.geometry) FROM districts`
- Pour extraire la géométrie des barycentres des districts :
`SELECT Centroid(districts.geometry) AS centroid_geometry FROM districts`

26. REQUÊTES SPATIALES TOPOLOGIQUES

Les requêtes spatiales topologiques interrogent les relations spatiales entre les objets spatiaux. Elles permettent de calculer des distances entre les objets, mais aussi de récupérer les intersections entre deux polygones, etc.

Les fonctions spatiales topologiques sont recensées sur les sites internet cités en introduction de la leçon précédente. La syntaxe reste identique et l'on doit ajouter le préfixe ST_ devant les fonctions dans PostGis.

NOTION DE REQUÊTE TOPOLOGIQUE

La relation ou *dépendance spatiale* est définie par une (ou plusieurs) propriété partagée par deux entités spatiales. Par exemple, la distance qui sépare deux points.

Les principales *relations topologiques* sont l'*adjacence*, la *connectivité*, l'*inclusion* et l'*intersection*. Ce sont des notions qualitatives et invariantes sous déformation continue.

Les requêtes topologiques sont de trois types :

- requête de test : renvoi vrai ou faux ;
- requête de distance : renvoie des valeurs numériques ;
- requêtes d'action : renvoie des objets spatiaux.

FONCTION DE TEST

Ces fonctions vérifient les *relations topologiques* entre deux entités spatiales en fonction de leur géométrie. Leur syntaxe est simple :

```
SELECT FONCTION(geometry1, geometry2) FROM nom_table
```

Les principales fonctions sont :

- `Equals()` pour tester si deux géométries sont identiques
- `Disjoint()` pour tester si deux géométries sont séparées
- `Touches()` pour tester si deux géométries se touchent
- `Within()` pour tester si la première géométrie est contenue dans la seconde
- `Overlaps()` pour tester si deux géométries se superposent
- `Crosses()` pour tester si deux géométries se croisent
- `Intersects()` pour tester si deux géométries se recoupent
- `Contains()` pour tester si la première géométrie contient la seconde

Pour sélectionner les hôtels qui sont dans le district Takamaka, on peut exécuter la requête suivante :

```
SELECT hotels.nom
FROM hotels, districts
WHERE Within(hotels.geometry, districts.geometry)
      AND districts.nom LIKE 'Takamaka'
```

On peut également vouloir connaître les routes qui intersectent le district de Takamaka et les exporter de QSpatialLite dans QGIS. Dans ce cas, on peut utiliser la requête suivante :

```
SELECT routes.nom, routes.geometry as geometry
FROM routes, districts
WHERE Intersects(routes.geometry, districts.geometry)
      AND districts.nom LIKE 'Takamaka'
```

Il est encore nécessaire de sélectionner l'option **Load in QGIS as Spatial Layer**, donner un nom à la table et identifier le nom du champ contenant la géométrie (ici geometry).

FONCTION DE DISTANCE

La fonction de distance exprime la distance séparant deux géométries. Elle s'utilise de la manière suivante:
`SELECT Distance(geometry1, geometry2) FROM nom_table`

Si l'on veut classer les hôtels en fonction de leur distance minimale à la route la plus proche, on peut utiliser la requête suivante:

```
SELECT hotels.nom, Min(Distance(hotels.geometry, routes.geometry)) as dist
FROM hotels, routes
GROUP BY hotels.nom
ORDER BY dist
```

OPÉRATEURS SPATIAUX

Les opérateurs spatiaux créent de nouveaux objets géométriques à partir d'objets géométriques existants. Les fonctions `Union()`, `Intersect()` et `Difference()` retournent respectivement un objet composé des deux géométries en argument, leur partie commune ou la partie de la première géométrie à laquelle on a soustrait la deuxième.

Dans SQLite, la fonction `Union()` s'écrit `GUnion()`, car la première est déjà utilisée dans le cadre des SQL non spatiales.

Ici, il est important de préciser la variable qui récupérera le champ de géométrie, car on crée un nouvel objet. La syntaxe est alors, dans ce cas:

```
SELECT Fonction(geometry1, geometry2) as geometry FROM nom_table
```

Les fonctions `Simplify()` et `Buffer()` complètent la liste des opérateurs spatiaux. La première est utilisée pour *simplifier* la géométrie d'un objet compte tenu d'une tolérance définie numériquement (algorithme de Douglas-Peucker). La deuxième crée une *zone tampon* autour de la géométrie d'une distance précisée dans les attributs. Les deux fonctions ont une syntaxe similaire:

```
SELECT Simplify(geometry, number) as geometry FROM nom_table
SELECT Buffer(geometry, number) as geometry FROM nom_table
```

Si l'on souhaite simplifier la géométrie des routes aux Seychelles, on peut écrire la requête suivante.

```
SELECT Simplify(geometry, 50) as geometry
FROM routes
```

La figure 1 montre le résultat de cette requête.

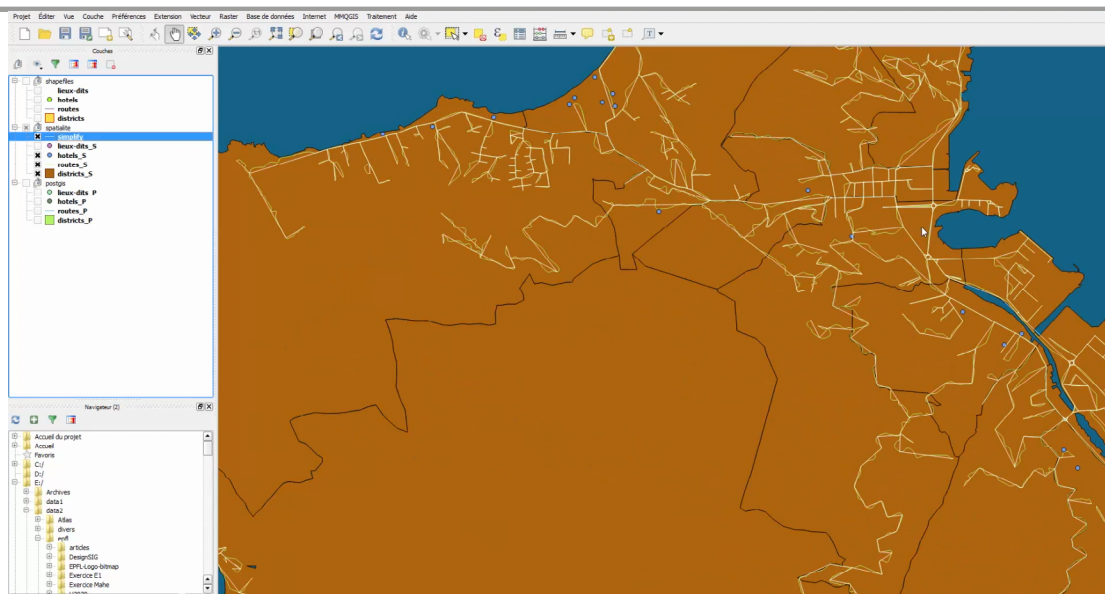


FIGURE 1

Si l'on veut créer une zone tampon de 100m autour de la route Sans Soucis Road, on peut exécuter la requête suivante.

```
SELECT Buffer(routes.geometry, 100) AS geometry
FROM routes
WHERE routes.nom LIKE 'Sans Soucis Road'
```

On vient de créer des polygones à partir des polygones qui composent la route, comme le montre la figure 2.

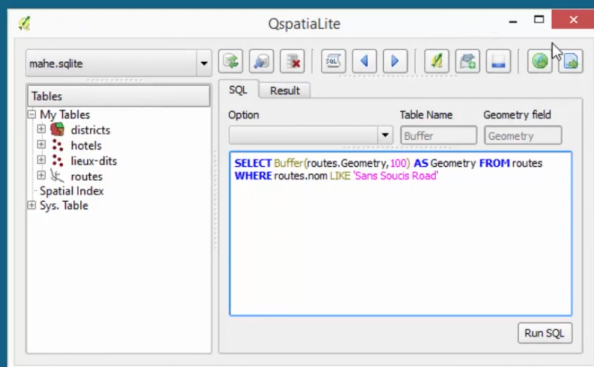


FIGURE 2

13:07

14:00

Zone tampon en rose de 100 m de part et d'autre de la route Sans Soucis Road.

Finalement, si l'on désire fusionner deux districts (Port Glaud et Grand Anse Mahe), on peut le faire à l'aide de la requête suivante:

```
SELECT GUnion(districts.geometry) AS geometry
FROM districts
WHERE districts.nom = 'Port Glaud'
OR districts.nom = 'Grand Anse Mahe'
```

27. BASES DE DONNÉES NOSQL

Les *bases de données NoSQL* (BD NoSQL) se distinguent de bases de *données relationnelles* par le fait qu'à la base, elles n'implémentaient pas le langage SQL. Les BD NoSQL forment une très vaste famille de BD.

ORIGINES DU NOSQL

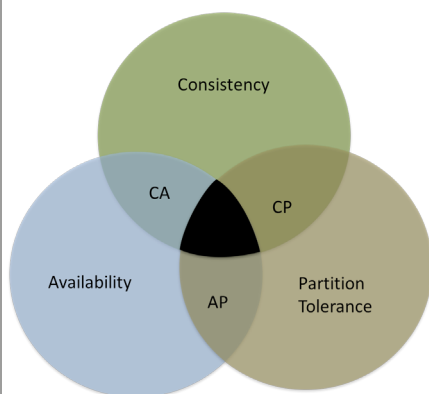
BD relationnelles : À titre de comparaison, voici un résumé de ce que sont les bases de données relationnelles. L'information est stockée dans des tables dont les colonnes définissent les attributs des éléments avec des types clairement définis. Les relations permettent de lier plusieurs tables. La consultation et l'édition des données sont réalisées par le langage standardisé SQL. Les transactions répondent aux *principes ACID*.

BD NoSQL : Elles sont apparues plus récemment que les BD relationnelles, pour répondre à de brusques afflux de trafic. Elles ont été développées d'abord par les sites internet gérant de gros volumes de données, tels que Google, Facebook, Amazon, LinkedIn, etc. à partir des années 1990. L'objectif d'une telle BD est de maintenir la scalabilité (ou évolutivité), c'est-à-dire maintenir les performances du système en cas de forte demande.

Un moyen de gérer ces brusques afflux de trafic est d'augmenter la puissance des ordinateurs (*scalabilité verticale*). Un autre est de mieux gérer la multiplication des nœuds (*scalabilité horizontale*) et distribution des services sur les différents nœuds du réseau. Le traitement des données peut également être accéléré en optimisant le système de gestion de la BD.

Ainsi, les BD NoSQL sont une alternative aux BD relationnelles. Leurs formes de stockage sont différentes, et, à l'origine, elles n'implémentaient pas le langage SQL. Il n'est pas possible d'y effectuer de jointure entre les données. Les données sont distribuées sur plusieurs serveurs. Finalement elles tendent à ne pas répondre aux *principes ACID*.

Théorème CAP



Scalabilité horizontale : La démultiplication des nœuds partageant des données donne naissance aux *systèmes distribués*. Ces derniers ne peuvent pas répondre simultanément à plus de deux des trois critères de *cohérence*, *disponibilité* (availability) et tolérance au *partitionnement* (CAP, figure 1).

- Cohérence: Tous les nœuds du système voient simultanément les mêmes données
- Disponibilité: Garantie que chaque requête reçoit une réponse en cas de succès ou d'échec
- Tolérance au partitionnement: le système reste opérationnel en dépit d'un partitionnement aléatoire dû à des ruptures de communication.

Au contraire, les BD NoSQL qui répondent à la tolérance de partition peuvent soit privilégier la consistance soit la disponibilité des données. Dans le premier cas (CP), on attend une synchronisation des nœuds en dépit de la disponibilité des données. Si la disponibilité n'est pas respectée, alors le système envoie une erreur de timeout. Dans le second cas (AP), la dernière version disponible au niveau du nœud que l'on interroge en acceptant que probablement, tous les nœuds ne sont pas à jour. Cela autorise donc le système à renvoyer des valeurs différentes.

Les BD relationnelles privilégient quant à elles la consistance et la disponibilité en dépit de la scalabilité horizontale (c'est-à-dire la tolérance au partitionnement) qui est un concept absent de ce type de BD.

FIGURE 1

5:37

24:00

Théorème CAP.

TYPES DE BD NOSQL ET SGBD

On distingue quatre types de BD NoSQL. Ils sont brièvement présentés dans le tableau ci-dessous.

TYPE	FONCTIONNEMENT	SGBD
Clé-valeur	Fonctionne comme un dictionnaire. Les valeurs d'un objet sont consultables par l'intermédiaire d'une clé unique.	Redis, Voldemort
Orienté colonne	Similaire aux BD relationnelles, à la différence que le stockage est organisé par colonne par paires clé-valeur.	Cassandra, Hbase, BigTable, Vertica
Orienté document	Repose sur un paradigme clé-valeur. Les données sont enregistrées dans des collections dans des fichiers JSON ou XML.	CouchDB, SimpleDB, MongoDB
Orienté graphe	Basé sur la théorie des graphes (notions de nœuds, relations et propriétés qui sont rattachées aux objets).	Neo4J

Types de Bd NoSQL.

La figure 2 illustre chacun de ces types de BD.

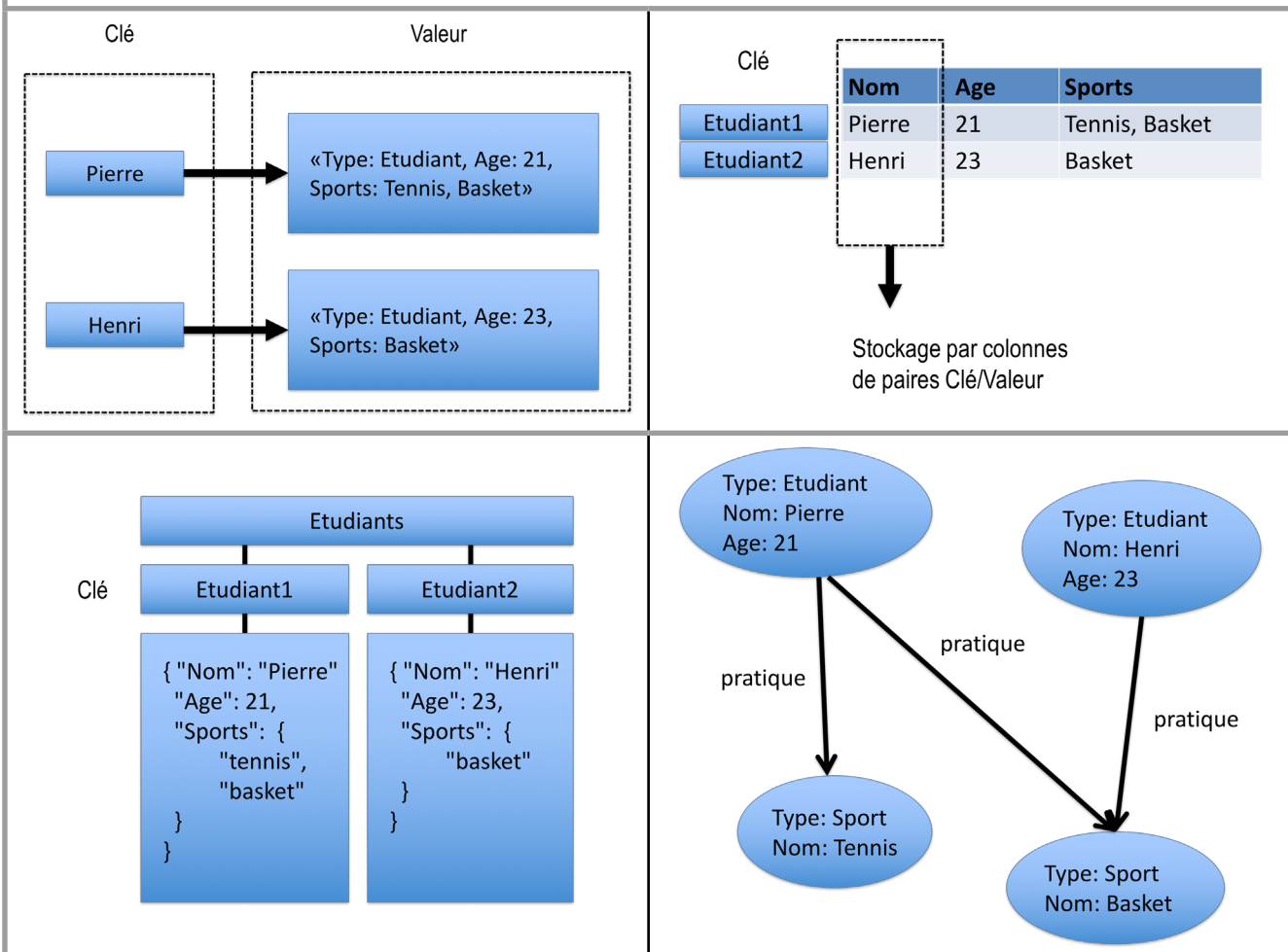


FIGURE 2

8:39

24:00

Illustration des quatre types de bases de données NoSQL. De gauche à droite : Clé-valeur, orienté colonne, orienté document et orienté graphe.



MONGODB

MongoDB est une base de données NoSQL libre orientée document. La notion de table dans les BD relationnelles est remplacée par la notion de *connexion*. Les données peuvent être visualisées sous la forme de représentation arborescente, de table ou au format JSON. Le schéma de données est libre, mais pour qu'il soit interprétable (dans un SIG par exemple), il est préférable de structurer les données.

Alors que dans les BD relationnelles la **dénormalisation** consiste en la suppression de la redondance des informations par l'intermédiaire de la création d'une nouvelle table, dans les BD NoSQL on préférera conserver cette redondance pour d'avantages de clarté. Notons au passage que le problème qui surgit dans les BD relationnelles lorsque l'on crée une table intermédiaire pour lier le responsable d'un hôtel à son établissement lorsqu'un hôtel est géré par plusieurs responsables disparaît dans les BD NoSQL. C'est là l'avantage des BD NoSQL: elles sont très performantes quand il s'agit de stocker de l'information en masse, lorsque l'information n'est pas très complexe. Par contre, lorsque les données sont fortement liées les unes aux autres, le système de BD relationnelle reste très intéressant.

CRUD ET MONGODB

CRUD est l'acronyme de *creating, reading, updating et deleting*. Quelques exemples d'édition de données en comparant les requêtes SQL pour les BD relationnelles et l'insertion de données dans une base de données MongoDB (BD NoSQL) sont présentés.

Insertion de données: Elle requiert l'utilisation des mots-clés `INSERT INTO` dans le monde SQL tandis qu'il existe une fonction `insert()` pour MongoDB. Ci-dessous on montre un exemple d'enregistrement de données dans une BD relationnelle et dans une MongoDB.

```
INSERT INTO hotels ('nom', 'chambres', 'lits', 'statut')
VALUES ('Beach Bungalows', 4, 8, 'Small hotel')
```

SQL

```
db.hotels.insert({
  nom: "Beach Bungalows",
  chambres: 4,
  lits: 8,
  statut: "Small hotel"
});
```

MongoDB

Mise à jour de données: On utilise les mots-clés `UPDATE noma_table SET nom_colonne = value` dans les BD relationnelles, alors qu'on utilise la fonction `update()` dans MongoDB.

```
UPDATE hotels
SET chambres = 6
WHERE nom = "Beach Bungalows"
```

SQL

```
db.hotels.update(
  { nom: "Beach Bungalows" },
  { $set: { chambres: 6 } }
);
```

MongoDB

Suppression de données: La suppression de données dans les BD relationnelles implique l'utilisation des mots-clés `DELETE FROM`, alors que l'on utilise la fonction `remove()` dans MongoDB.

```
DELETE FROM hotels
WHERE nom = "Beach Bungalows"
```

SQL

```
db.hotels.remove({
  "nom": "Beach Bungalows"
});
```

MongoDB



Requêtes: Pour afficher les hôtels dont le nombre de chambres est supérieur à 10, on utilise les requêtes suivantes:

```
SELECT nom
FROM hotels
WHERE chambres > 10
```

```
db.hotels.find(
  { chambres: { >: 10 } },
  { _id: 0, nom: 1 }
);
```

SQL

MongoDB

Dans la syntaxe de MongoDB, les deux lignes d'attributs de la fonction `find()` correspondent respectivement au critère de recherche et aux données à afficher (afficher si 1, ne pas afficher si 0).

Pour compter le nombre d'hôtels qui possèdent 10 chambres, on utilise les requêtes suivantes:

```
SELECT count(nom)
FROM hotels
WHERE chambres = 10
```

```
db.hotels.count(
  { "chambres": 10 }
);
```

SQL

MongoDB

Pour lister le nombre de chambres qu'il y a pour chaque statut, on utilise les requêtes suivantes:

```
SELECT statut, count(1) as total
FROM hotels
GROUP BY statut
```

```
db.hotels.aggregate([
  { $group:
    {
      _id: "$statut",
      total: { $sum: 1 }
    }
  }
]);
```

SQL

MongoDB

Pour consulter et éditer les données contenues dans une base de données MongoDB, on peut utiliser l'invite de commande (sous Windows) ou un terminal (sous Linux et Mac) et taper la commande `mongo`. Cela nécessite l'installation de mongo au préalable. On peut également utiliser une interface utilisateur graphique, comme le programme `MongoChef` Professionnel ces fonctions sont déjà implémentées et un `Query Builder` permet de construire des requêtes.



28. ÉTUDE DE CAS – LES FEUX DE BROUSSE AU SAHEL

Cette étude de cas illustre le rôle des SIG pour le monitoring des feux de brousse au Sahel.

LES FEUX DE BROUSSE

Le Sénégal subit chaque année d'énormes dégâts liés aux feux de brousse. On distingue deux causes des feux : l'agriculture (culture sur brûlis et défrichage) et les déclenchements par négligence des feux précoces. Le phénomène couvre l'ensemble de la zone soudano-sahélienne pendant la saison sèche (octobre à mai) qui précède la mise en culture.

Ces feux provoquent la destruction du patrimoine forestier qui représente un intérêt économique important ainsi qu'un appauvrissement de la diversité biologique. Ils représentent également une menace sur les habitants et leurs biens et contribuent au réchauffement climatique (production de gaz à effet de serre et réduction des capacités de séquestration du carbone atmosphérique). Finalement, leur impact se mesure aussi sur la production du forage qui constitue la base de l'alimentation du bétail.

LA MISSION

Les pouvoirs publics en charge de l'environnement définissent une stratégie de gestion des feux. Ils ont confié cette mission de surveillance au CSE (Centre de Suivi Écologique) qui utilise les données satellitaires pour le suivi et la cartographie des superficies brûlées.

Les différents ministères de l'institut de l'environnement et l'agence nationale de météorologie organisent la collecte, le traitement et la diffusion de l'information sur les feux de brousse afin de véhiculer une information unique et fiable. Cette information est également utilisée pour réaliser des analyses spatiales et des représentations qui sont des outils d'aide à la décision au niveau du ministère de l'environnement.

LE RÔLE DU SIG

La campagne de suivi des feux de brousse a lieu de fin octobre à fin mai (saison sèche). Elle comprend le repérage des feux à partir d'images satellitaires MODIS, notamment les images thermiques, et la caractérisation des sites ayant été dévastés par le feu à partir de ces images.

Les satellites renouvellent l'information deux fois par jour. En analysant les surfaces brûlées, on peut en déduire les courbes d'évolution des feux pendant la saison sèche.

Le rapport saisonnier contient l'analyse de la distribution spatiale et temporelle des feux, des cartes, l'évaluation des impacts économiques et une série de recommandations. Les analyses ne tiennent compte dans ce rapport que des feux de brousse. Il ne tient donc pas compte des feux déclenchés à titre préventif ou les feux liés à l'agriculture (création du brûlis et défrichage). Cette considération est mise en œuvre par l'intermédiaire d'un masque spatio-temporel appliqué sur les données.

ANALYSE TEMPORELLE

L'analyse de la distribution temporelle est réalisée dans QGIS et permet de déterminer le nombre de feux par mois. Une couche Feux_2014_2015 contient les feux recensés ainsi que la date à laquelle ils ont eu lieu. Le nombre de feux qui ont eu lieu par mois doit être déterminé. Pour ce faire, il faut utiliser une requête dans l'extension QSpatialite.

```
SELECT strftime('%m',d.DATE), count(f.PKUID)
FROM Feux_2014_2015 f
GROUP BY strftime('%m',d.DATE)
```

Ensuite, on peut exporter la table créée dans QGIS puis l'enregistrer au format CSV par exemple.

ANALYSE SPATIALE

L'analyse du nombre de feux qui ont eu lieu par district administratif est également une grandeur importante. Elle est obtenue à l'aide de cette commande SQL :

```
SELECT d.nom_dep, count(f.PKUID), d.geometry as geometry
FROM Départements d, Feux_2014_2015 f
WHERE within(f.geometry, d.geometry)
GROUP BY d.nom_dep
```

On exporte ensuite la table dans QGIS. On peut choisir une symbologie adaptée au nombre de feux détectés par district en appliquant une échelle de couleurs graduée par exemple. C'est ce qui est illustré à la figure 1. On constate qu'il y a davantage de feux dans la région sud-est du pays.

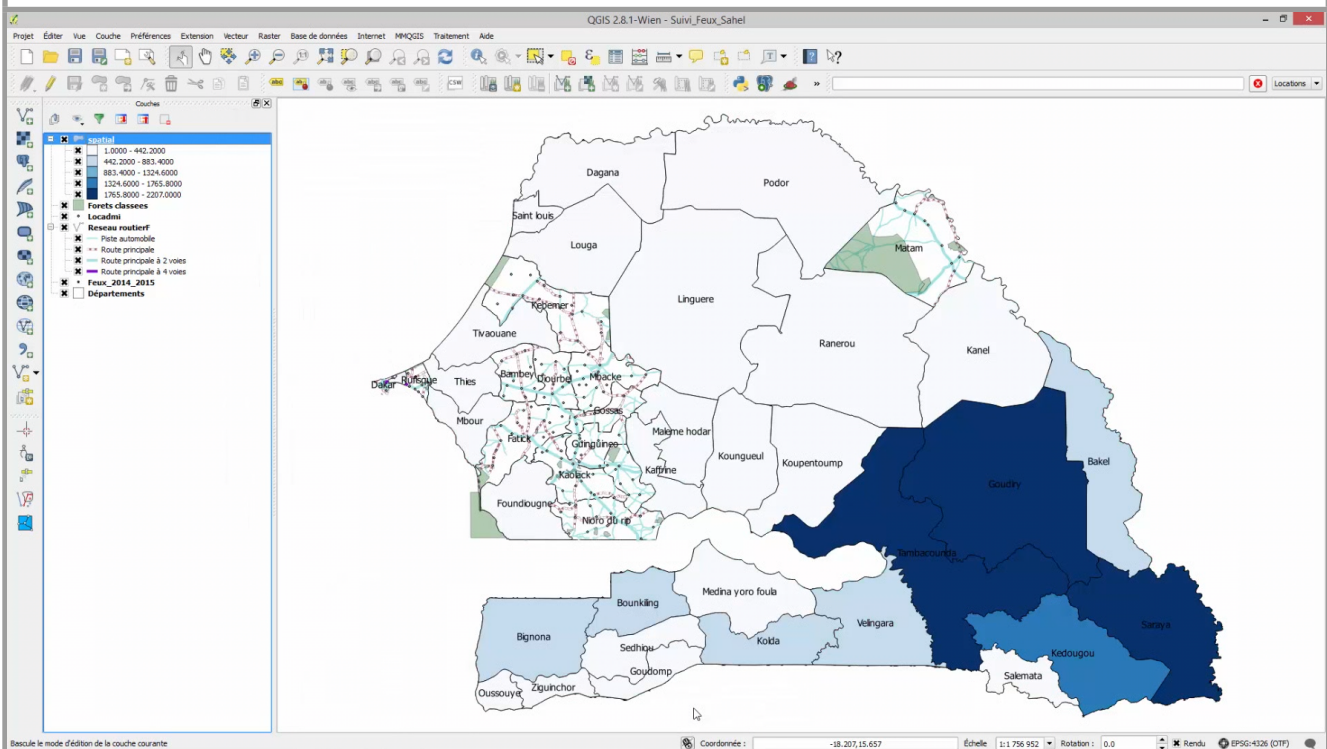


FIGURE 1

15:14

19:00

Affichage des districts en fonction du nombre de feux détectés au Sahel.

RÉSULTATS ET EFFETS

Deux types de résultats sont publiés: les résultats en cours de saison (hebdomadaires) et les résultats de fin de saison qui sont publiés en juin. Le rapport de fin de saison contient les cartes de synthèse, les analyses (notamment l'impact économique) et des recommandations. Les impacts économiques tiennent compte des dommages causés par le feu et des interventions pour lutter contre le feu. La figure 2 montre un exemple de carte où les feux sont reportés sur les districts. On y voit également les routes principales, les cours d'eau, les villes principales et les limites administratives.

Une bonne exploitation des données permettrait d'aider significativement à lutter contre les feux de brousse.

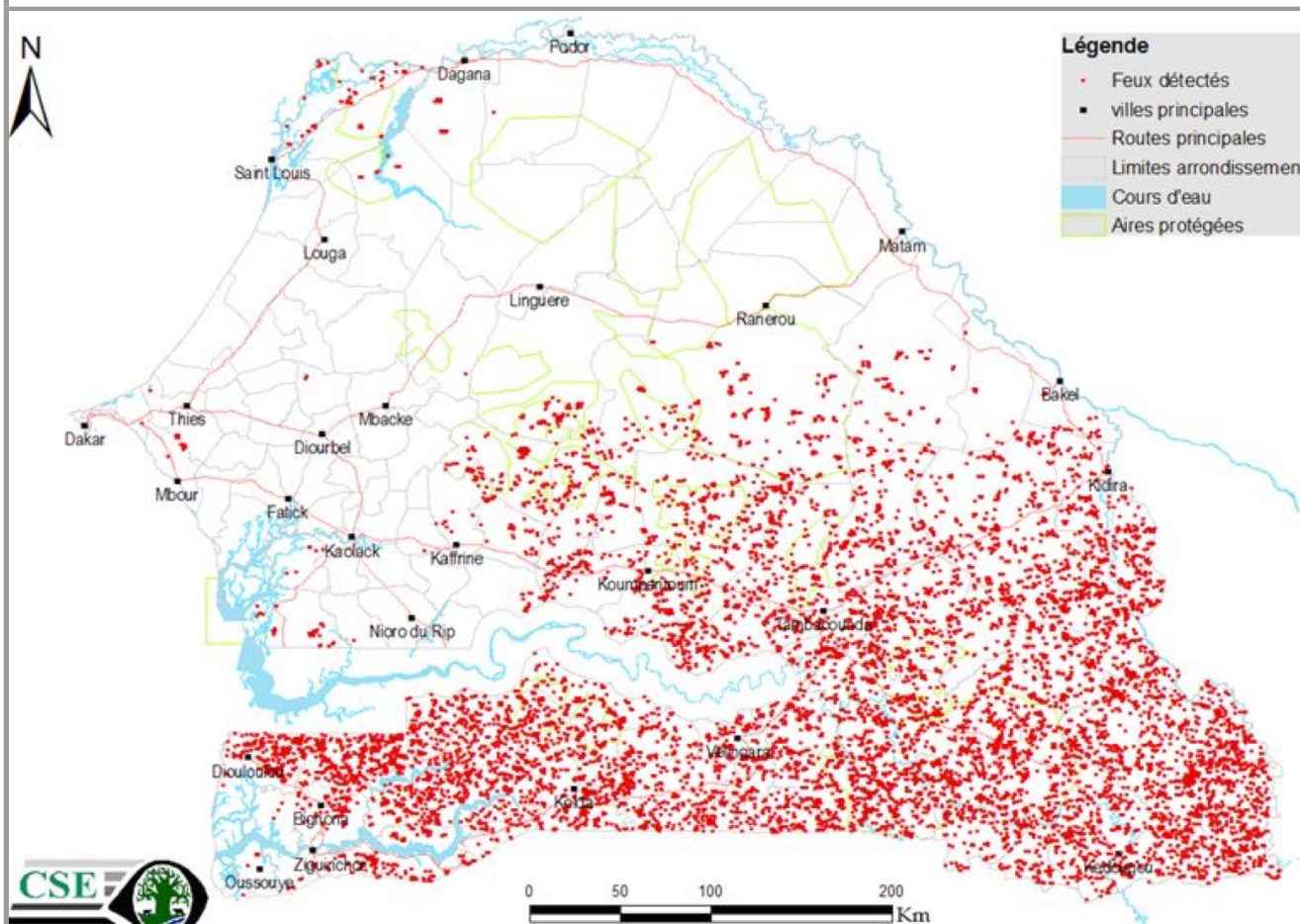


FIGURE 2

15:58

19:00

Carte de synthèse extraite d'un rapport de saison.

CONCLUSION

Les SIG permettent de connaître et de surveiller les mois durant lesquels on observe le plus grand nombre de feux ainsi que de déterminer les départements les plus vulnérables aux feux de brousse. Les autorités peuvent donc prendre les mesures nécessaires pour une meilleure gestion des feux de brousse.



BOOCs
EPFL

Systèmes d'Information
Géographique – Partie 1
Stéphane Joost, Marc Soutter,
Fernand Kouamé et Amadou Sall



IMPRESSUM

© EPFL Press, 2017.
Tous droits réservés.

Graphisme :
Emphase Sàrl, Lausanne

Résumé : Marc Rufener

Développés par EPFL Press, les BOOCs (Book and Open Online Courses) sont le support compagnon des MOOCs proposés par l'École polytechnique fédérale de Lausanne. Valeur ajoutée aux MOOCs, ils rassemblent l'essentiel à retenir pour l'obtention du certificat et constituent un atout pédagogique. Learn faster, learn better. Bonne révision !

ISBN 978-2-88914-413-6