

Requêtes spatiales géométriques



Objectifs de la leçon

- Découvrir les requêtes SQL permettant d'explorer les caractéristiques et propriétés des géométries spatiales

Après cette leçon vous serez capables

- D'extraire d'une base de données des éléments comme les coordonnées, les dimensions ou le système de référence de géométries spatiales qu'elle héberge

Introduction aux systèmes d'information géographique

Bonjour et bienvenue dans cette leçon qui va porter sur les requêtes spatiales géométriques, un ensemble de requêtes donc qui s'intéresse plus spécifiquement à extraire de l'information sur la nature et les propriétés géométriques des objets spatiaux stockés dans une base de données. L'objectif de cette leçon consiste donc à explorer ce monde des requêtes SQL qui cherche à extraire des propriétés caractéristiques géométriques des objets spatiaux stockés dans une base de données, de sorte que vous soyez en mesure de récupérer des informations comme les coordonnées d'un point, le périmètre ou la longueur d'une ligne, la surface d'un polygone, à partir des géométries spatiales hébergées dans une base de données.

Notes

Summary



0m 21s

Requêtes spatiales géométriques

<http://www.gaia-gis.it/gaia-sins/spatialite-sql-4.2.0.html>



<http://postgis.net/docs/index.html>



Introduction aux systèmes d'information géographique

Il existe un grand nombre de requêtes spatiales géométriques différentes avec une grande variété de mots-clés que nous ne pouvons pas passer évidemment tous en revue donc nous allons vous présenter quelques situations, quelques requêtes types qui nous paraissent particulièrement importantes.

Notes

Summary



1m 07s

Requêtes spatiales géométriques



ST = Spatial Type

ST_ + Fonction



- **Area()** >> **ST_Area()**
- **Contains()** >> **ST_Contains()**
- **Intersects()** >> **ST_Intersects()**
- ...

Introduction aux systèmes d'information géographique

C'est toujours une bonne idée de consulter les sites des logiciels spécialisés, en l'occurrence le site de SpatiaLite et celui de PostGIS, pour se faire une idée un peu plus exhaustive de l'ensemble des fonctionnalités disponibles en terme de requêtes spatiales géométriques. En général, la syntaxe est à peu près la même dans tous les cas avec une particularité toutefois dans le monde PostGIS qui, dès lors que la requête porte sur les objets spatiaux, fait précéder en fait le mot-clé des lettres ST souligné, ST pour Spatial Type, qui permet dans une requête SQL de distinguer clairement tout ce qui s'adresse à des propriétés spatiales et qui sont spécifiques au monde PostGIS par rapport monde PostgreSQL sur lequel PostGIS est bâti. Mais pour le reste, la syntaxe est la même que dans SpatiaLite à quelques rares exceptions près.

Notes

Summary



Géométrie et référence spatiale

geometry

- L'attribut geometry représente la géométrie de l'objet spatial et est notamment utilisé dans une requête de sélection pour afficher des objets sur une carte
- ... ou pour extraire la référence du système de projection utilisé
- Syntaxe

```
SELECT geometry FROM nom_table
```

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

Introduction aux systèmes d'information géographique

Dans cette leçon, nous allons aborder d'abord les requêtes qui permettent de sélectionner et d'afficher des géométries, afficher dans une carte, ou de consulter la référence spatiale utilisée, donc le code ESPG. Nous verrons ensuite quelques syntaxes de requêtes SQL, de conversion de données pour transformer des géométries en format texte ou en format binaire, des requêtes qui permettent d'extraire, de consulter le type de géométrie auquel on a affaire et finalement, nous verrons une série de requêtes qui permettent de récupérer les propriétés spécifiques des géométries. Le mot-clé GEOMETRY désigne en général l'attribut géométrie d'une table de données spatiales, donc la géométrie de l'objet spatial et il est utilisé dans les requêtes de sélection pour afficher des objets sur une carte. On peut également l'utiliser pour extraire la référence du système de projection utilisé ou tout un tas d'autres informations que nous verrons par la suite. La syntaxe se base tout simplement sur le mot-clé SELECT, GEOMETRY, on sélectionne les géométries, FROM, le nom de la table pour indiquer la table de laquelle on extrait les géométries.

Notes

Summary



2m 26s

Géométrie et référence spatiale

geometry

- L'attribut geometry représente la géométrie de l'objet spatial et est notamment utilisé dans une requête de sélection pour afficher des objets sur une carte
- ... ou pour extraire la référence du système de projection utilisé
- Syntaxe

```
SELECT geometry FROM nom_table
```

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

Introduction aux systèmes d'information géographique

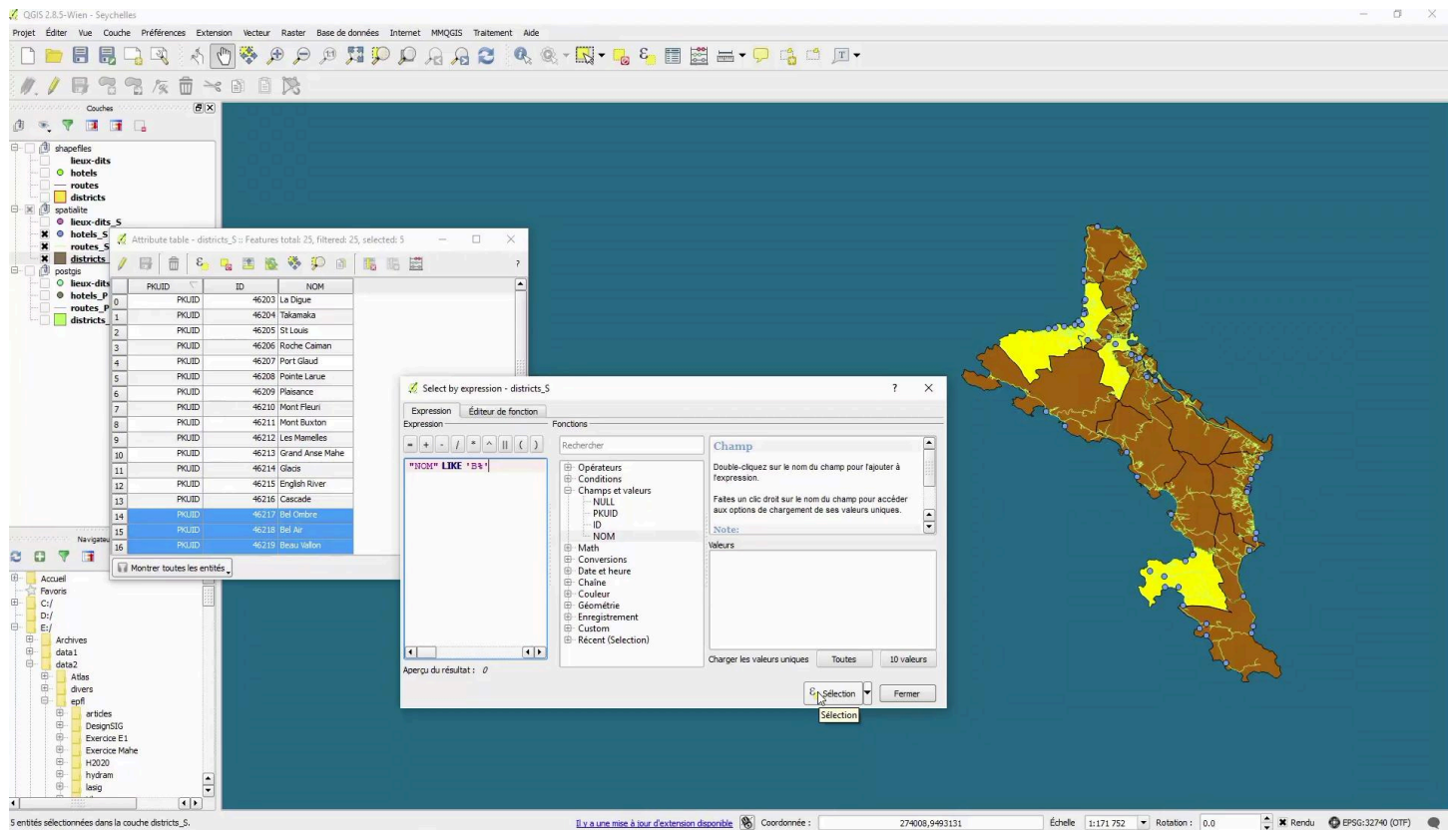
Puis après, on peut imaginer des clause conditionnelles qui permettent de trier, dans les géométries, celles qui répondent à certains critères. On en verra un exemple tout à l'heure. Le mot-clé SRID avec le mot-clé GEOMETRY entre parenthèses pour extraire la référence du système de projection utilisé, en l'occurrence le code EPSG.

Notes

Summary



3m 49s

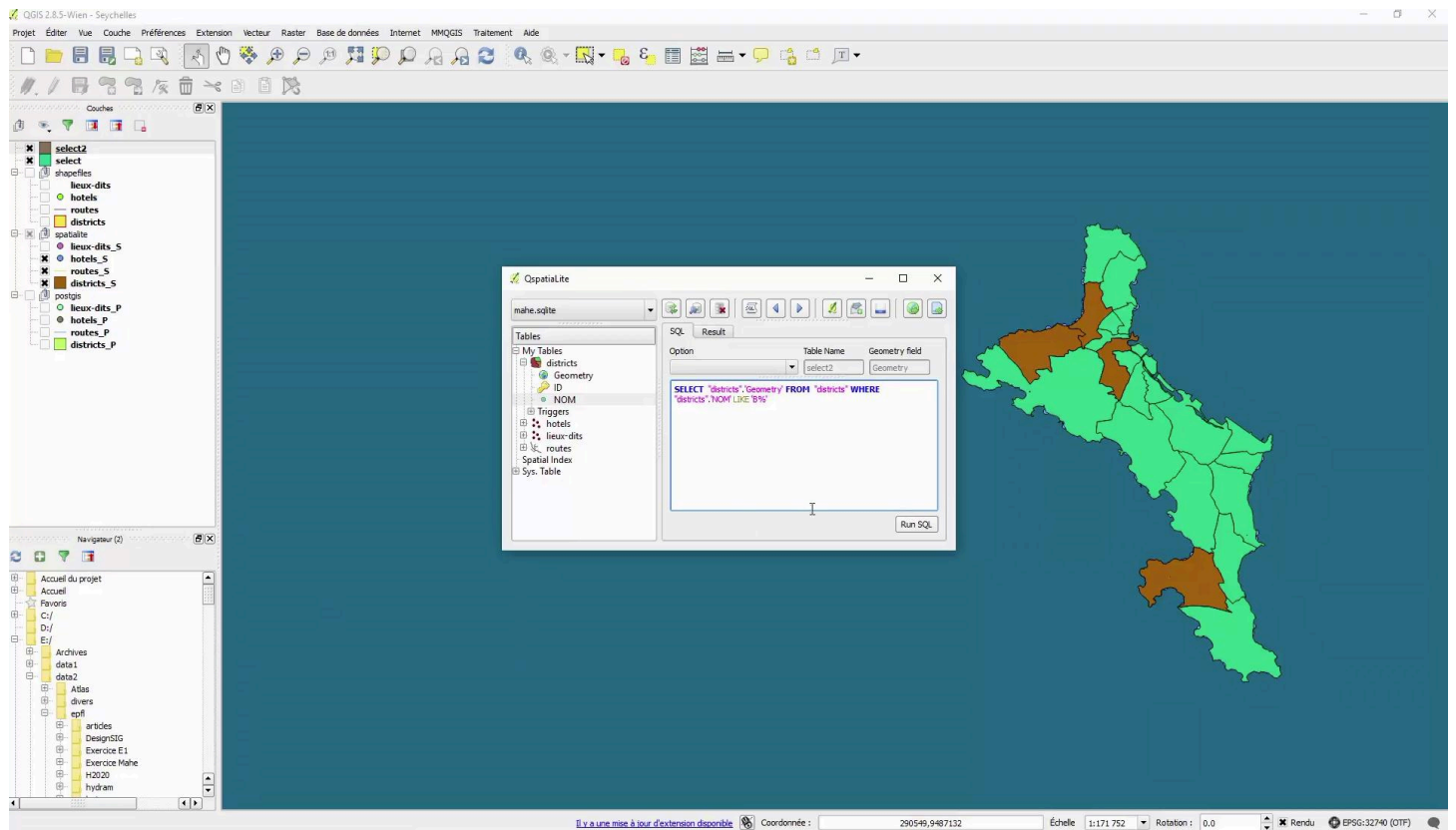


Alors un exemple de l'application de ces éléments. Tout d'abord, lorsqu'il s'agit de sélectionner en fait des objets, le plus simple est d'utiliser l'outil intégré dans QGIS avec un module de sélection qui fait intervenir du langage SQL simplement le `SELECT GEOMETRY FROM` est masqué et on ajoute simplement le critère dans la syntaxe SQL, ici le nom qui commence par d, et on sélectionne l'ensemble des districts dont le nom commence par d.

Notes

Summary





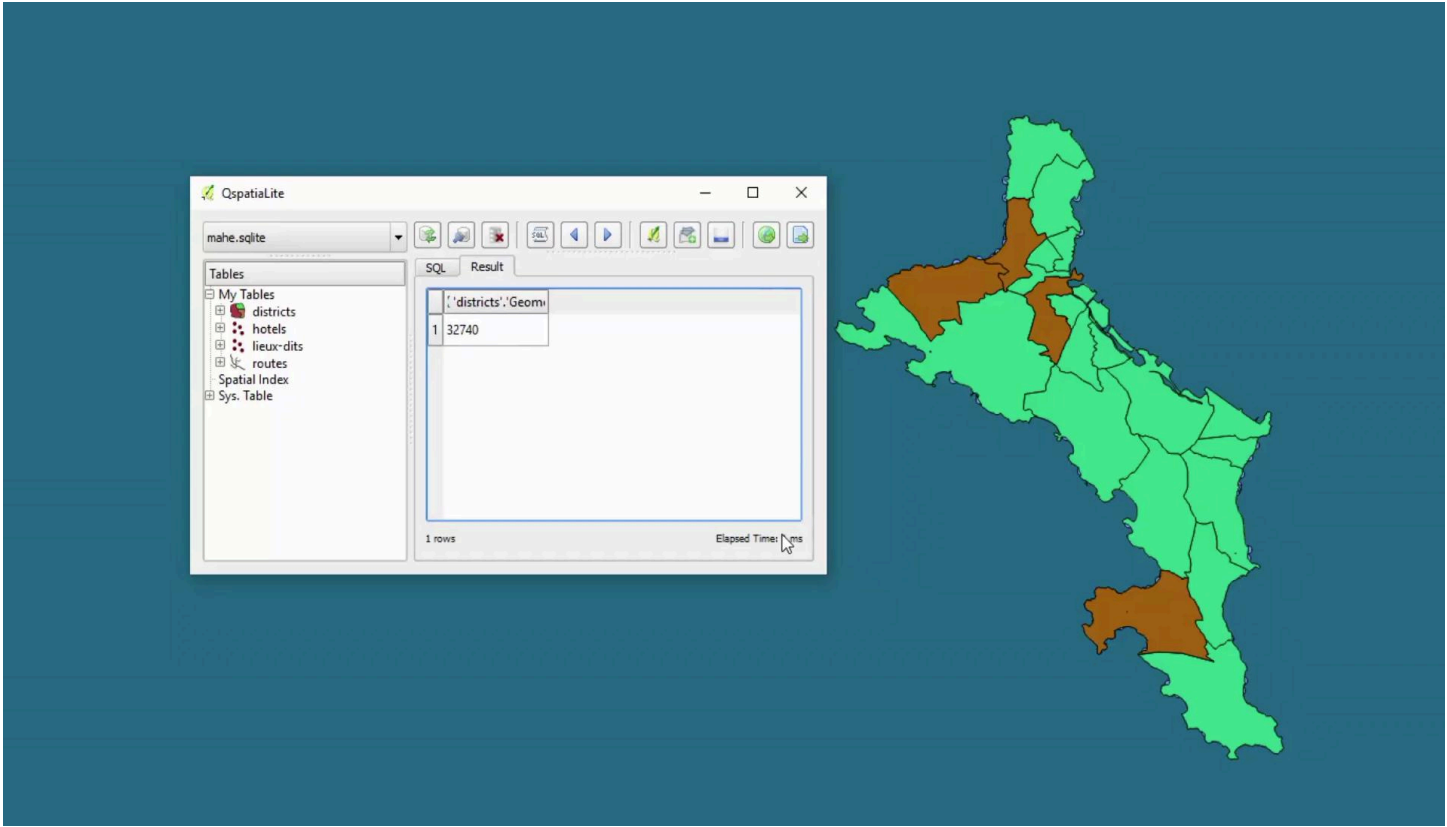
Second exemple, cette fois dans le module Spatialite, avec à nouveau une requête dans la table des districts pour sélectionner l'ensemble des districts de Mahé. Pour afficher les résultats de la requête, il faut en fait les charger comme une nouvelle couche dans QGIS, couche dont il faut définir le nom et le champ qui contient les géométries. Finalement, tout comme on l'a fait avec le requêteur QGIS, on peut modifier la requête Spatialite en introduisant un clause conditionnelle pour limiter la sélection aux districts dont le nom commence par d. Et à nouveau, il faut définir le résultat de la requête comme une nouvelle couche pour afficher cette sélection de districts dans la carte.

Notes

Summary



4m 47s



Pour récupérer le système de projection de ces objets, donc on élimine la partie conditionnelle de la requête et on ajoute la fonction SRID qui s'applique donc aux géométries. En exécutant cette requête, on voit que tous nos objets, enfin une grande partie de nos objets, sont en 32740 qui est donc le système UTM 40 Sud et on voit avec le mot-clé DISTINCT que l'ensemble en fait des objets ont ce même système de référence.

Notes

Summary

5m 38s



Conversion

AsText(), AsBinary()

- Les fonction AsText() et AsBinary() permettent une conversion du format de stockage de la géométrie vers du WKT ou WKB (Well-Known Text / Binary)
- Très utile pour exporter (resp. importer) un jeu de données d'une base de données et les convertir dans un autre format de stockage
- Syntaxe

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

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

Introduction aux systèmes d'information géographique

Les fonctions de conversions, qui s'appliquent dont également à l'attribut GEOMETRY, qui permettent de transformer une géométrie en format texte pour la rendre lisible ou au contraire, en format binaire pour la stocker de manière à ce qu'elle prenne peu de place. Ces deux fonctions sont très utiles pour exporter / importer des jeux de données d'une base de données à une autre et les convertir dans un autre format de stockage. La syntaxe est toujours assez simple, le mot-clé SELECT, la fonction AsText, avec la géométrie être parenthèses, FROM et le nom de la table.

Notes

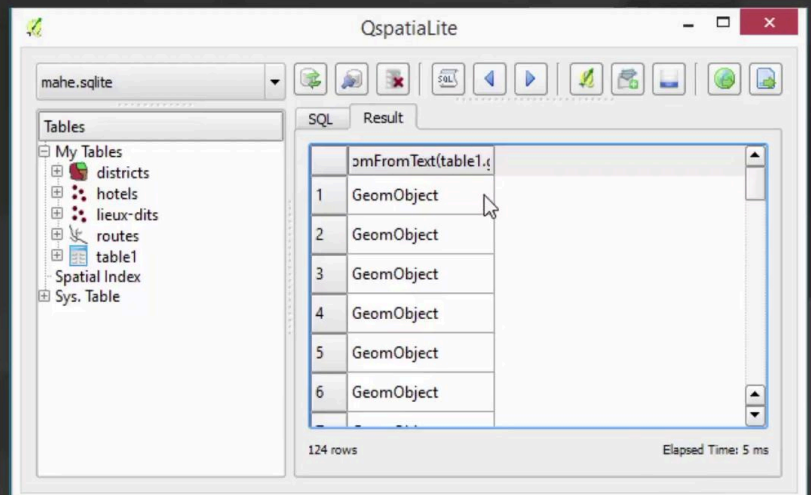
Summary



6m 22s

Conversion

AsText(), AsBinary()



Pour illustrer cette opération, nous reprenons l'interface SpatiaLite de QGIS et l'on écrit simplement cette requête transformation des géométries des hôtels en texte et on donne un alias en fait à cette colonne, on l'appelle gg, donc ça vient de la table des hôtels et on voit que notre colonne gg contient une série de multipoints avec les coordonnées x y. On peut créer une table à partir de ces résultats, qu'on va appeler table 1, et ajouter cette table en fait dans la base de données SpatiaLite. On peut consulter les objets de cette table. On voit qu'on retrouve nos multipoints avec une clé primaire qui a été ajoutée automatiquement. La fonction équivalente pour transformer la géométrie en un nombre binaire permet de constater en fait que ce que l'on fabrique, c'est un objet géométrique que l'on va pouvoir utiliser pour des cartes. Supposons maintenant qu'on ait obtenu, importé cette table qui contient des géométries sous forme texte. Nous pouvons utiliser une fonction GeomFromText, ce qui permet de transformer ces géométries textes en des géométries binaires, donc des objets géométriques que l'on va pouvoir ensuite utiliser pour les afficher dans une carte.

Notes

Summary



7m 00s

Caractéristiques de la géométrie

GeometryType(), NPoints(), NRings(), Envelope()

- Information sur le type de géométrie et sur ses propriétés: le nombre de points, le nombre d'anneaux, l'enveloppe
- Syntaxe

```
SELECT GeometryType(geometry) FROM nom_table
```

[...]

```
SELECT Envelope(geometry) FROM nom_table
```

Introduction aux systèmes d'information géographique

Une série de fonctions qui permette à présent de récupérer les caractéristiques propres de la géométrie spatiale, à commencer par le type de géométrie, le nombre de points ou le nombre d'anneaux qu'elle peut contenir et son enveloppe, donc le bounding box, donc le rectangle qui englobe la géométrie. La syntaxe est toujours du type SELECT, la fonction GeometryType ou Envelope et le mot-clé GEOMETRY, entre parenthèses, FROM et le nom de la table.

Notes

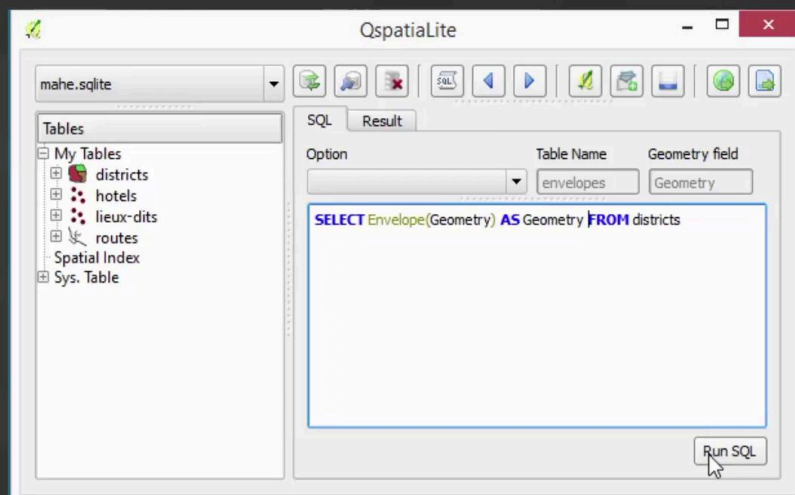
Summary



8m 57s

Caractéristiques de la géométrie

GeometryType(), NPoints(), NRings(), Envelope()



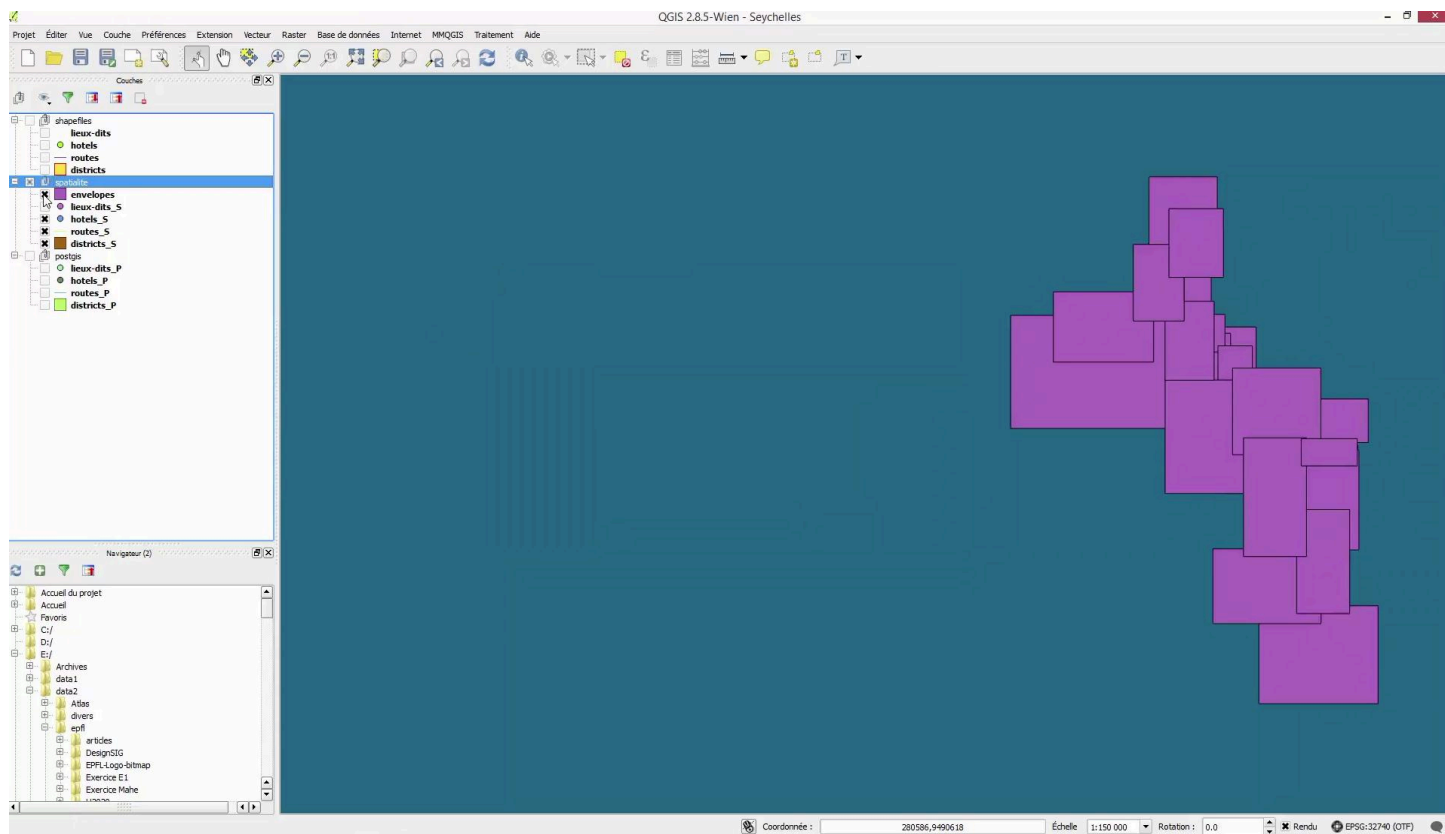
Alors on illustre ce type de requêtes avec la fonction qui permet d'extraire le type de géométrie d'une couche, en l'occurrence, on va s'intéresser à la couche des districts et on voit que la syntaxe SQL est ici assez permissive puisque l'on peut simplement mentionner l'attribut GEOMETRY, le nom de la table, levant toute ambiguïté. Avec le mot-clé DISTINCT, on peut vérifier que l'ensemble de la couche des districts est constitué de multipolygones. Si maintenant on remplace cette extraction de type de géométrie par la fonction d'enveloppe, on voit qu'on crée une série d'objets géométriques et pour ajouter ces objets géométriques à la carte, il faut en faire une colonne GEOMETRY, définir le nom de la table dans laquelle on souhaite stocker la chose, définir le champ GEOMETRY comme contenant la géométrie et exécuter la requête.

Notes

Summary



9m 32s



Donc cette requête crée une couche qui s'appelle On Enveloppe et on retrouve les enveloppes qui englobent les différents districts de l'île de Mahé ici.

Notes

Summary

10m 33s



Propriétés de la géométrie

Points – X(), Y()

- Renvoie les coordonnées X et Y d'un point
- Syntaxe

```
SELECT X(geometry), Y(geometry) FROM nom_table
```

Introduction aux systèmes d'information géographique

Parmi les nombreuses fonctions qui renvoient des propriétés spécifiques des géométries spatiales, nous allons commencer par simplement celles qui renvoient les coordonnées x y d'un point avec une syntaxe tout simple de type SELECT X entre parenthèses la géométrie, virgule Y la géométrie si on veut extraire les deux coordonnées, FROM le nom de la table.

Notes

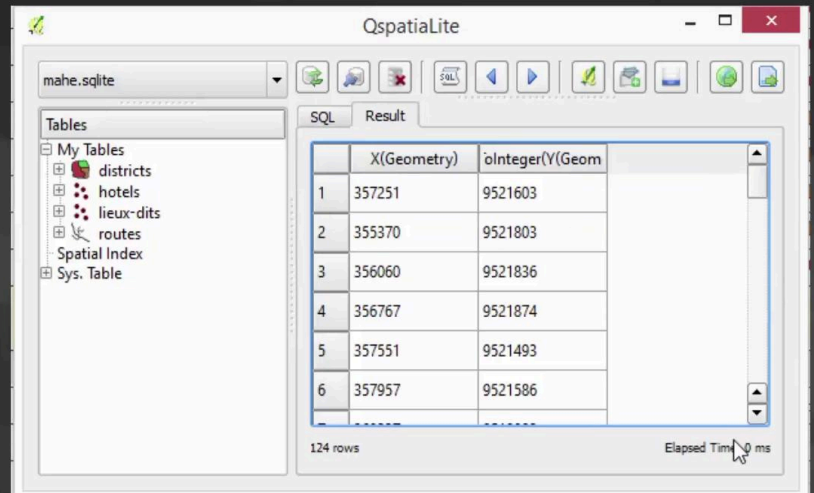
Summary



10m 50s

Propriétés de la géométrie

Points – X(), Y()



The screenshot shows the QspatialLite application window. On the left, a tree view shows the database structure: 'mahe.sqlite' containing 'My Tables' (districts, hotels, lieux-dits, routes, Spatial Index) and 'Sys. Table'. The main window displays a SQL query result for the 'hotels' table. The result table has two columns: 'X(Geometry)' and 'Integer(Y(Geom))'. The data is as follows:

	X(Geometry)	Integer(Y(Geom))
1	357251	9521603
2	355370	9521803
3	356060	9521836
4	356767	9521874
5	357551	9521493
6	357957	9521586

At the bottom of the result window, it indicates '124 rows' and 'Elapsed Time 0 ms'.

On illustre cette syntaxe en sélectionnant les coordonnées x et y des hôtels des Seychelles. Donc ici à nouveau, simplement le mot-clé GEOMETRY, le nom de la table levant tout ambiguïté. On voit que le résultat de la colonne grecque est en notation scientifique et donc on peut utiliser une fonction CastTo vers un entier pour transformer cette notation scientifique en un nombre entier pour rendre simplement la chose plus lisible.

Notes

Summary



11m 15s

Propriétés de la géométrie

Polygones – StartPoint(), EndPoint(), Length(), isClosed(), isRing ()

- StartPoint() et EndPoint() retournent le point initial ou le point final d'une polyligne
- Length() retourne la longueur d'une polyligne
- isClosed() permet de savoir si une polyligne est fermée, et isRing() de savoir si elle constitue un contour externe ou interne d'un polygone

Introduction aux systèmes d'information géographique

Deuxième série de fonctions qui concernent davantage ici les polygones avec l'extraction du point initial et du point final de la polyligne, StartPoint, EndPoint, la longueur, Length et un test pour vérifier si la polyligne est ouverte ou fermée et éventuellement un test pour vérifier s'il s'agit d'un anneau ou pas, donc si elle fait partie d'un polygone.

Notes

Summary



11m 54s

Propriétés de la géométrie

Polygones – StartPoint(), EndPoint(), Length(), isClosed(), isRing ()

- StartPoint() et EndPoint() retournent le point initial ou le point final d'une polyligne
- Length() retourne la longueur d'une polyligne
- isClosed() permet de savoir si une polyligne est fermée, et isRing() de savoir si elle constitue un contour externe ou interne d'un polygone
- Syntaxe, par exemple

```
SELECT GLength(geometry) FROM nom_table
```

NB. Length() est la dénomination OpendGIS, mais GLength() dans spatialite et ST_Length() dans postgis

Introduction aux systèmes d'information géographique

La syntaxe, c'est du type SELECT, la fonction entre parenthèses la géométrie, FROM le nom de la table comme d'habitude avec une particularité, le fait que la fonction de longueur qui renvoie la longueur de la polyligne est précédée d'un G majuscule dans Spatialite pour la simple raison que le mot-clé LENGTH est un mot réservé de Spatialite et dans QGIS, comme je l'ai dit tout au début, en fait cette fonction s'écrit ST pour spatial type, souligné et ensuite le mot-clé LENGTH.

Notes

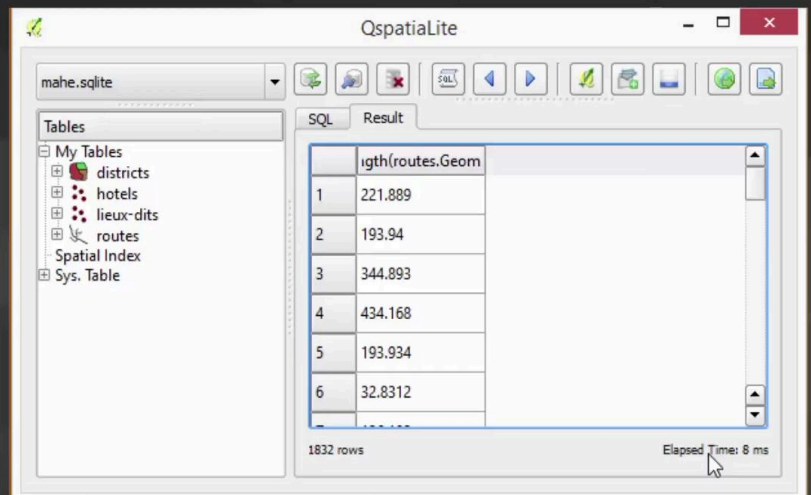
Summary



12m 19s

Propriétés de la géométrie

Polygones – StartPoint(), EndPoint(), Length(), isClosed(), isRing ()



The screenshot shows the QspatialLite application window. On the left, a tree view under 'mahe.sqlite' shows a 'Tables' folder containing 'My Tables' (with sub-items 'districts', 'hotels', 'lieux-dits', 'routes', 'Spatial Index') and 'Sys. Table'. The main area is split into 'SQL' and 'Result' tabs. The 'Result' tab displays a table with 6 rows and 2 columns. The first column is an index (1-6) and the second column is the result of the SQL query 'length(routes.Geom)'. The status bar at the bottom indicates '1832 rows' and 'Elapsed Time: 8 ms'.

	length(routes.Geom)
1	221.889
2	193.94
3	344.893
4	434.168
5	193.934
6	32.8312

Comme exemple de ce type de requête, nous allons extraire la longueur des tronçons de route de la table des routes des Seychelles. Et l'on voit qu'en exécutant cette requête, on obtient bel et bien la longueur en mètres de ces tronçons de route.

Notes

Summary



12m 53s

Propriétés de la géométrie

Polygones – Centroid(), Area()

- Centroid() renvoie le point situé au barycentre du polygone
- Area() retourne la surface du polygone
- Syntaxe

```
SELECT Centroid(geometry) FROM nom_table
```

```
SELECT Area(geometry) FROM nom_table
```

Introduction aux systèmes d'information géographique

Deux fonctions qui concernent à présent plus spécifiquement les polygones, la fonction Centroid qui renvoie en fait le barycentre du polygone, le centre de gravité du polygone, et la fonction Area qui retourne à sa surface. La syntaxe est toujours la même avec le mot-clé SELECT, la fonction, la géométrie entre parenthèses, le mot-clé FROM et le nom de la table.

Notes

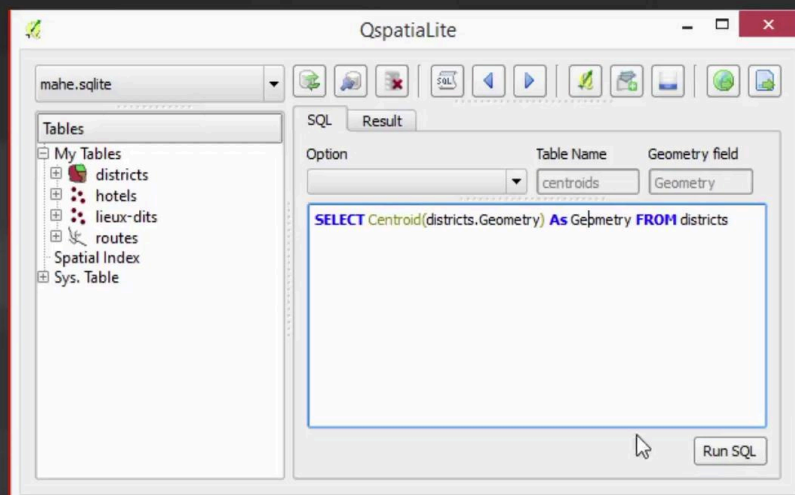
Summary



13m 15s

Propriétés de la géométrie

Polygones – Centroid(), Area()



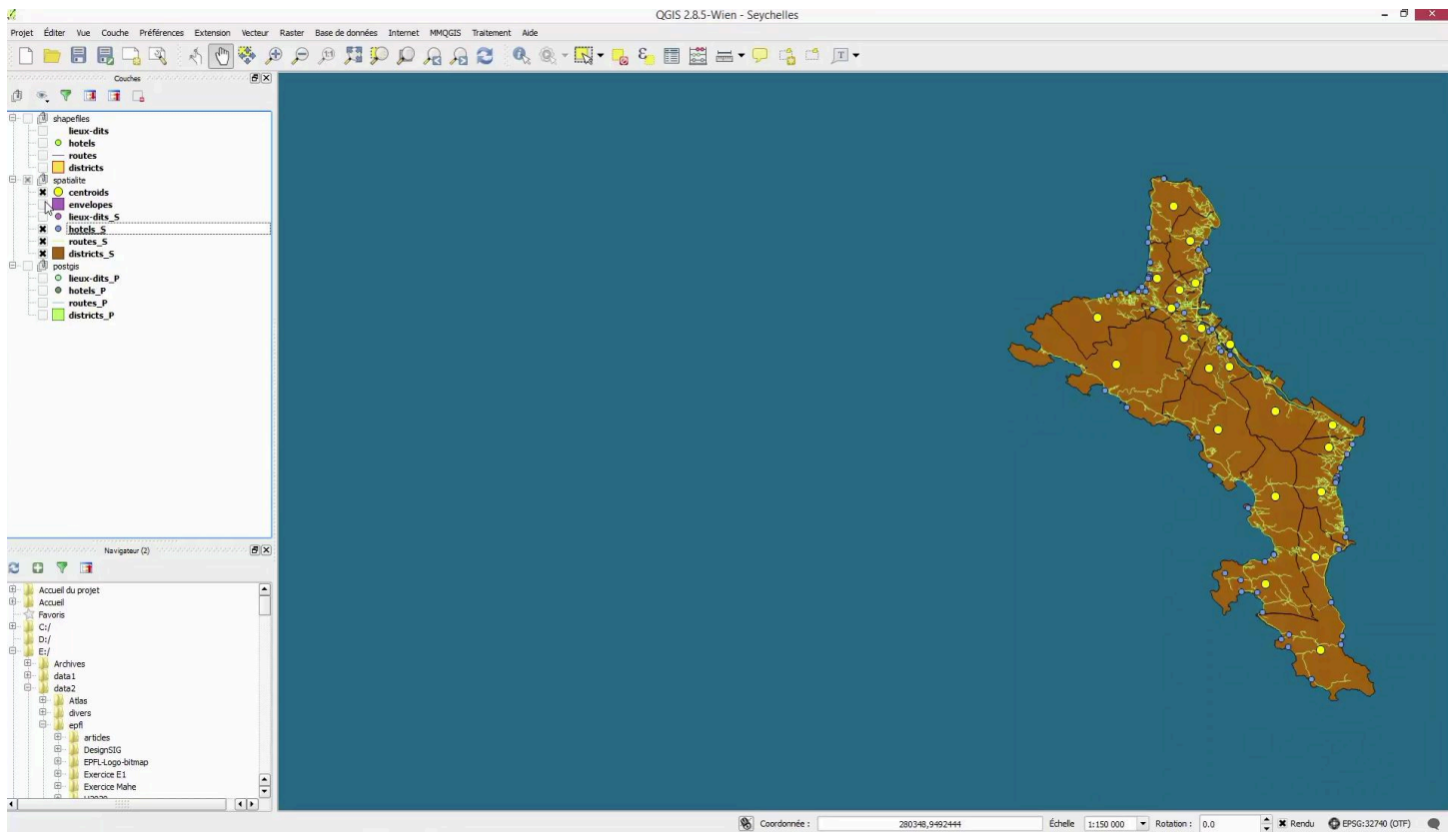
Alors dans notre exemple ici, nous commençons par sélectionner, par extraire les surfaces des districts des Seychelles. Voilà. Donc on a de nouveau une notation de type scientifique. Si l'on s'intéresse maintenant plutôt au Centroid, on obtient en fait des objets géométriques que l'on peut vouloir représenter sur la carte. Donc pour cela, il faut charger le résultat comme une couche dans QGIS, définir la géométrie, le champ de la géométrie avec le mot-clé GEOMETRY, donner un nom à la table et s'assurer que le champ géométrie s'appelle bien GEOMETRY. En exécutant cette requête, on voit que l'on crée donc une couche qu'on a appelé centroid.

Notes

Summary



13m 37s



Et si l'on affiche ces couches centroid, on voit qu'on a des points jaunes qui apparaissent au centre des districts des Seychelles.

Notes

Summary



14m 31s