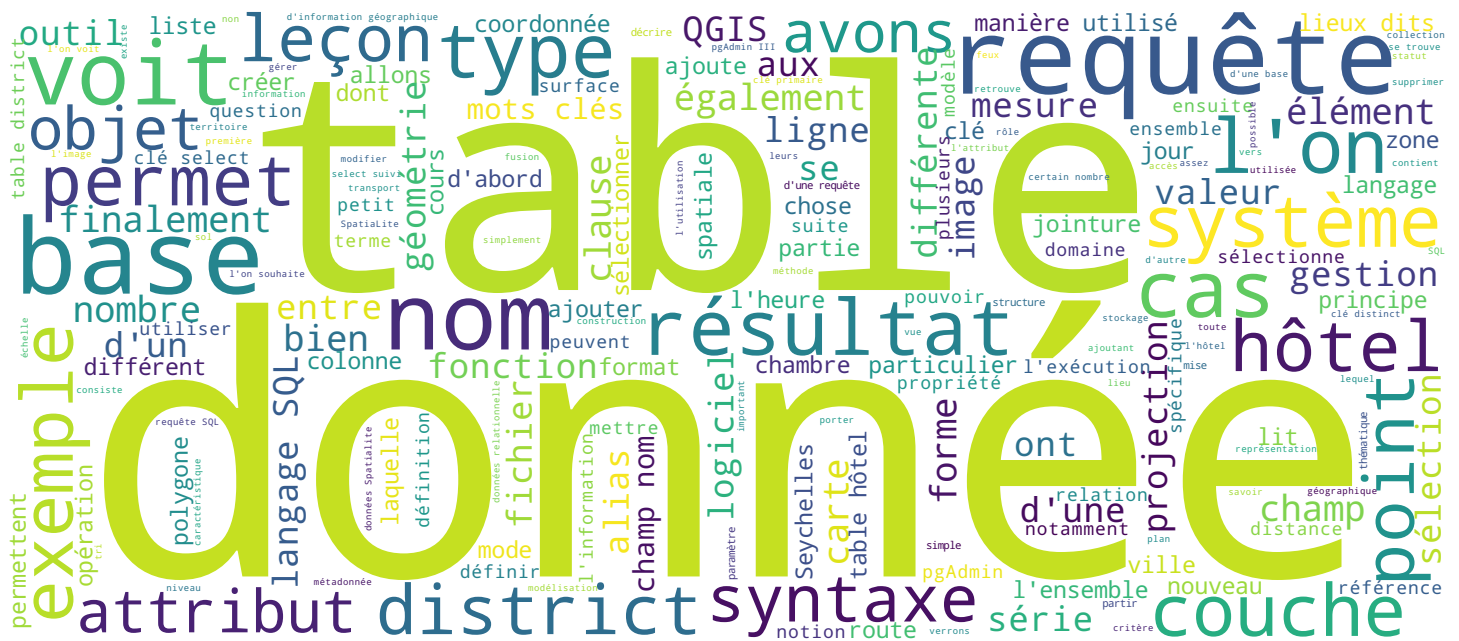


Requêtes et langage SQL

Introduction aux systèmes d'information géographique

Stéphane Joost, Marc Soutter, Fernand Kouamé, Amadou Sall



Search MOOC



Video



Requêtes et langage SQL



Objectifs de la leçon

- Découvrir les composantes du langage SQL
- Apprendre la syntaxe de la clause SELECT

Après cette leçon vous serez capables de

- Sélectionner des attributs dans une ou plusieurs tables
- Enlever les doublons du résultat

Introduction aux systèmes d'information géographique

Bienvenue dans cette leçon qui va donc porter sur l'interrogation des bases de données par des requêtes, et sur le langage SQL qui est utilisé pour écrire ces requêtes. Dans cette leçon, nous allons aborder donc les bases du langage SQL, langage qui va nous occuper pour les 5 ou 6 prochaines leçons. L'objectif de cette première leçon est donc de découvrir les principes les plus fondamentaux du langage SQL et en particulier la fonction ou la clause, la plus basique qui soit qui est la clause "select" et ses différents usages. Au terme de cette leçon, vous devriez être en mesure de sélectionner des attributs dans une table et... d'éliminer les doublons qui pourraient se trouver dans cette sélection.

Notes

Summary



0m 22s

Le SQL

Structured Query Language - SQL

- Langage de gestion des bases de données **relationnelles**
- Langage normalisé, donc **indépendant des SGBD**
- Interaction avec les BD par l'intermédiaire de **requêtes structurées**



Nous verrons donc successivement les principes du langage SQL puis la sélection simple d'un attribut dans une table. Le principe de la sélection distincte qui permet d'éliminer les doublons d'une requête. Ensuite, la sélection à partir de 2 tables et finalement l'utilisation des alias pour les attributs et pour les tables. Le SQL ou Structured Query Language est donc un langage destiné à la gestion des bases de données relationnelles spécifiquement. C'est un langage qui a été normalisé donc qui est en principe indépendant du système de gestion de base de données utilisé même s'il est vrai que chaque système de gestion de base de données a quelques spécialités au niveau de la syntaxe utilisée. C'est un langage qui permet d'interagir avec des bases des données sous forme de requête structurée comme son nom l'indique.

Notes

Summary



1m 11s

Le SQL

Interrogation des données

- Data Query Language – DQL

+ 3 groupes d'instructions complémentaires

- Data Definition Language – DDL
- Data Manipulation Language – DML
- Data Control Language - DCL

	ID /	NAME	ROOMS	MANAGER	ADDRESS
0	1	Beach Bungalows	4	Mr. A. Confait	Grand' Anse
1	2	Grand' Anse Bea...	9	Mrs. M. F. Wartelsteiner	Amitie
2	3	Maison De Palme...	24	Mr. Basil Ferrari	Amitie
3	4	Le Lagon Baron ...	10	Mr. Karl D' Unienville	Grand' Anse
4	5	Les Cabanes Des...	5	Mr. Jose Confait	Grand' Anse
5	6	Britannia *	12	Mrs. L. Lablache	Grand' Anse
6	7	Villa Flamboyant	6	Mr. Martial Adeline	St. Sauveur



	ID /	NAME	ROOMS	MANAGER	ADDRESS
2	3	Maison De Palme...	24	Mr. Basil Ferrari	Amitie
3	4	Le Lagon Baron ...	10	Mr. Karl D' Unienville	Grand' Anse
5	6	Britannia *	12	Mrs. L. Lablache	Grand' Anse

Interroger les données /
Extraire des données

Introduction aux systèmes d'information géographique

Le SQL est composé de 4 groupes d'instructions complémentaires. En premier lieu le Data Query Language, langage d'interrogation des données qui permet d'extraire des données d'une base des données.

Notes

Summary



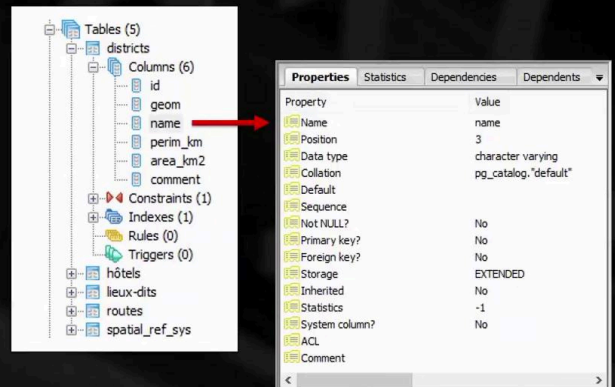
Le SQL

Interrogation des données

- Data Query Language – DQL

+ 3 groupes d'instructions complémentaires

- Data Definition Language – DDL
- Data Manipulation Language – DML
- Data Control Language – DCL



Définir ou modifier la structure de la base de données

Introduction aux systèmes d'information géographique

Le Data Definition Language, langage de définition de données qui permet de modifier ou de définir la structure d'une base de données.

Notes

Summary



2m 27s

Le SQL

Syntaxe de base

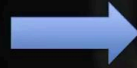
```
SELECT nom_attribut  
FROM nom_table
```

```
WHERE condition
```

```
GROUP BY nom_attribut  
HAVING condition
```

```
ORDER BY nom_attribut  
LIMIT nb_lignes
```

```
UNION/INTERSECT/EXCEPT  
requête2
```



Bases de données test



Introduction aux systèmes d'information géographique

Le langage de manipulation DML qui permet d'insérer, de mettre à jour, de supprimer des données et finalement le langage de contrôle qui permet de gérer les droits et les accès des utilisateurs. Dans le domaine de l'interrogation des données, la syntaxe de base comprend des clauses de sélection avec les mots clés "select" et "from", les clauses de filtre conditionnel avec le mot clé "where", des clauses d'agrégation avec les mots clés "group by" ou "having"; des clauses de tri avec les mots clés "order by" ou "limit" et finalement des clauses de fusion avec les mots clés "union", "intersect", "except". L'utilisation de ces différents mots clés dans des requêtes va être illustrée dans la suite de cette leçon et dans les leçons suivantes. Sur la base de données qui ont été regroupées dans une série de bases de données test, données qui portent sur les îles proches des Seychelles qui comprennent en fait 4 couches de données : les districts sous forme de polygones, les hôtels, les lieux-dits qui sont des points et les routes qui sont des lignes.

Notes

Summary

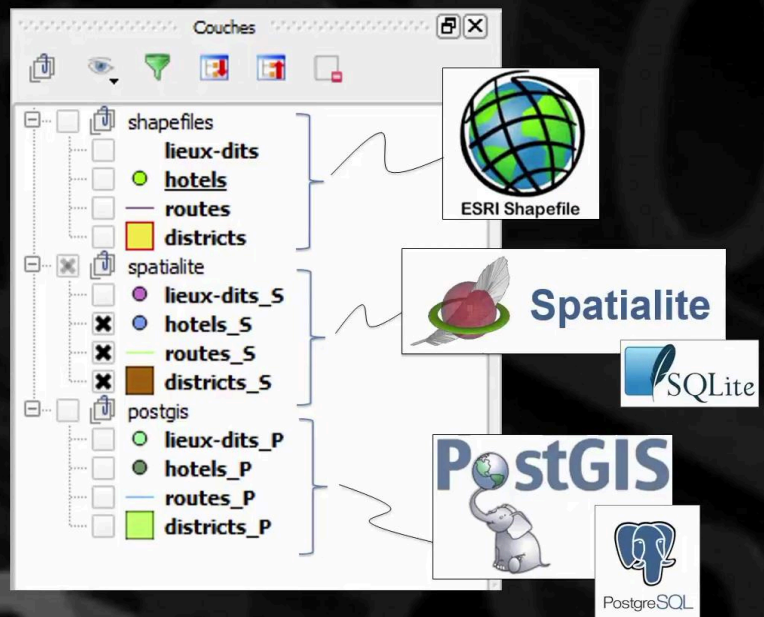


2m 35s

Le SQL – Bases de données test

Seychelles, les îles proches

- 4 couches de données
 - ➔ Districts (polygones)
 - ➔ Hôtels (points)
 - ➔ Lieux-dits (points)
 - ➔ Routes (lignes)
- Dans 3 formats
 - ➔ Shapefile
 - ➔ Spatialite
 - ➔ Postgis



Ces données sont stockées dans 3 formats différents ce qui permettra d'illustrer 3 types d'approches, de gestion différente des requêtes, le format ESRI Shapefile, une base de données Spatialite et une base de données PostGIS.

Notes

Summary

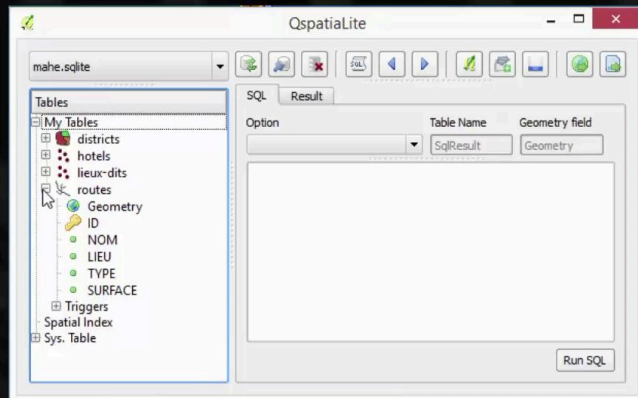


3m 42s

Le SQL – Bases de données test

Seychelles, les îles proches

- 4 couches de données
 - ➔ Districts (polygones)
 - ➔ Hôtels (points)
 - ➔ Lieux-dits (points)
 - ➔ Routes (lignes)
- Dans 3 formats
 - ➔ Shapefile
 - ➔ Spatialite
 - ➔ Postgis



Introduction aux systèmes d'information géographique

On peut voir ici que ces différentes couches de données ont un certain nombre d'attributs. Pour les districts, la géométrie, l'identifiant et le nom, pour les hôtels de nouveau géométrie, identifiant, le nom et une série d'attributs complémentaires (le nombre de chambres, le nombre de lits, le statut) et le district dans lequel se trouve les hôtels. Pour les lieux-dits, géométrie, identifiant et nom de même que pour les routes avec en plus le lieu où elles se trouvent, le type de route et le type de surface. On voit qu'il y a une petite clé autour de l'identifiant qui montre que dans tous ces cas de figure, l'identifiant joue le rôle de clé primaire.

Notes

Summary



3m 59s

Le SQL – Outils de requête

Couches QGIS en général

- Query builder (propriétés de couche)
- Requêtes sur la table d'attributs
- Extension Requête spatiale

Spatialite et PostGis

- Extension Gestion de base de données

Spécifique Spatialite

- Sqlite Studio (hors QGIS)
- Extension QSpatialite

Spécifique PostGis

- pgAdmin III (hors QGIS)
- Extension postGisQueryBuilder
- Extension Traitements / PostGis execute SQL

Introduction aux systèmes d'information géographique

Il existe de nombreux outils de requête et nous allons nous concentrer sur ceux que l'on peut utiliser en lien avec le logiciel QGIS. Il y a tout d'abord dans le logiciel-même une série de 3 outils de requête qui... s'appliquent de manière générale à tous les types de couches. Mais ces 3 solutions ont le défaut de ne pas implémenter de manière explicite le langage SQL, si bien que dans le cas présent, ce n'est pas très intéressant. Nous avons une extension gestion de base de données qui permet de traiter des bases de données Spatialite et PostGIS et sans doute d'autres, et puis une série de solutions spécifiques à Spatialite et spécifiques à PostGIS. Certaines d'entre elles faisant appel à des logiciels hors QGIS comme le SQLite Studio pgAdmin III. Dans les solutions PostGIS PostGisQueryBuilder est également peu intéressante car n'implémentant pas explicitement le langage SQL. Pour les besoins de ce cours, nous allons travailler avec l'extension Spatialite pour QGIS qui permet d'écrire et d'exécuter des requêtes SQL sur une base de données Spatialite dans QGIS. Et nous utiliserons également pgAdmin III qui est un logiciel qui fait partie de la suite PostGreSQL et qui présente l'avantage de proposer un outil de construction de requête graphique.

Notes

Summary

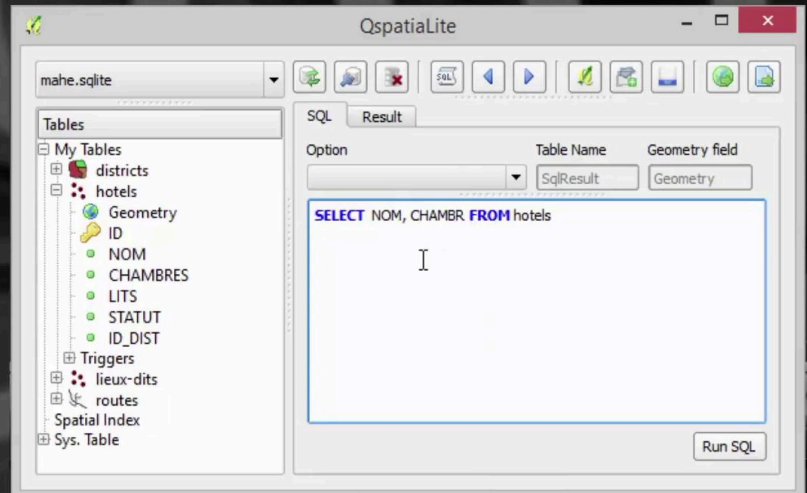


4m 41s

Sélection simple

Sélectionner deux attributs

```
SELECT nom_attribut_1, nom_attribut_2  
FROM nom_table
```



La requête de sélection comprend donc les mots clés "select" suivi du nom de l'attribut recherché et le mot clé "from" qui permet de préciser dans quelle table se trouve l'attribut en question. Dans QGIS, avec l'extension QSpatialite, on voit donc que l'on peut écrire une requête SQL. Dans la fenêtre prévue à cette effet, nous allons écrire une requête pour sélectionner la liste des hôtels par leurs noms. Donc l'attribut recherché c'est l'attribut "nom" et la table, la table "hôtels". L'exécution de la requête donne donc la liste des noms des hôtels des Seychelles. La même opération peut être effectuée graphiquement dans pgAdmin III en ajoutant la table des hôtels dans le champ de construction de la requête en sélectionnant le champ "nom". On voit que ça se traduit par la création d'une requête SQL dont la syntaxe est un tout petit peu différente. Cette syntaxe permet de lever toute ambiguïté dans le cas où on a plusieurs tables avec des attributs de mêmes noms. Par contre, elle n'est pas obligatoire lorsque le doute n'est pas permis. La requête de sélection de 2 attributs comprend le mot clé "select" suivi du nom des attributs séparés par une virgule puis du mot clé "from" et du nom de la table.

Notes

Summary

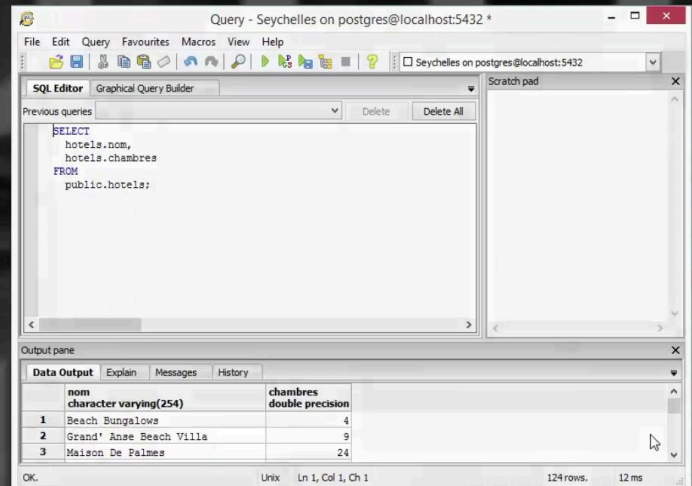


6m 22s

Sélection simple

Sélectionner deux attributs

```
SELECT nom_attribut_1, nom_attribut_2  
FROM nom_table
```



Dans notre exemple, nous ajoutons le nombre de chambres comme attribut recherché et l'exécution de la requête montre donc le nombre de chambres par... en regard du nom de l'hôtel. Dans pgAdmin, il suffit de sélectionner le champ supplémentaire dans le constructeur graphique ce qui met à jour la requête en SQL et son exécution donne bien le résultat escompté.

Notes

Summary

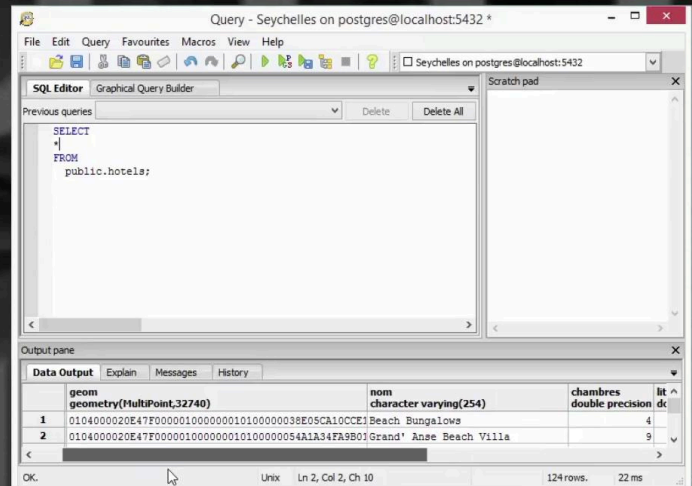


7m 39s

Sélection simple

Sélectionner tous les attributs

```
SELECT *  
FROM nom_table
```



La syntaxe qui permet de sélectionner l'ensemble des attributs d'une table est constituée du mot clé "select" suivi d'un astérisque puis du mot clé "from" et du nom de la table. On voit dans cet exemple que si l'on remplace les noms des champs recherchés par une astérisque, on obtient effectivement une table qui contient l'ensemble des champs de la couche. Pareillement, dans le constructeur graphique de pgAdmin...alors on voit qu'il n'y a pas d'astérisque donc on peut sélectionner l'ensemble des champs manuellement ou alors dans la fenêtre SQL simplement remplacer les champs par une astérisque et le résultat est toujours le même. On peut noter au passage qu'il y a 124 hôtels sélectionnés.

Notes

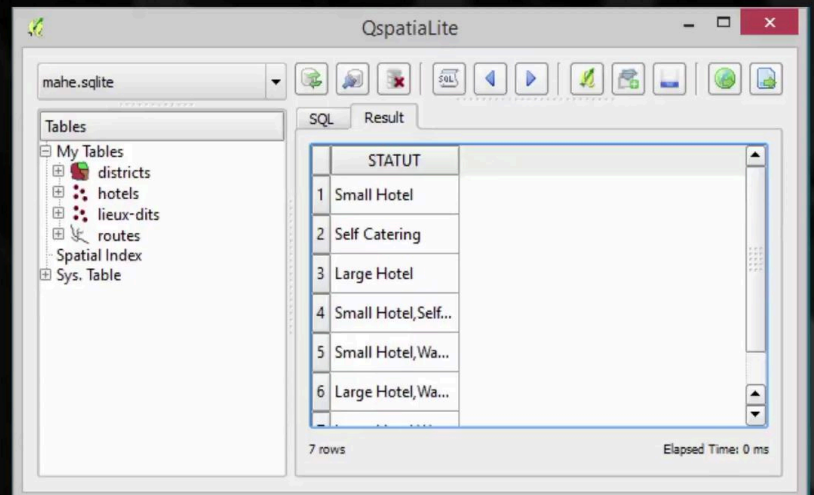
Summary



8m 08s

Sélection distincte

```
SELECT DISTINCT nom_attribut  
FROM nom_table
```



Le mot clé "distinct" placé à la suite du mot clé "select" permet d'éliminer les doublons qui se retrouveraient dans le résultat d'une requête. On voit que si l'on effectue une requête sur le statut des hôtels, on obtient une table dans laquelle on retrouve plusieurs fois la même valeur puisque de nombreux hôtels ont en fait le même statut - petit hôtel, grand hôtel etc. Le mot clé "distinct" ajouté à cette requête permet en fait de filtrer ce résultat et de n'obtenir que la liste des différentes valeurs possibles de l'attribut "statut".

Notes

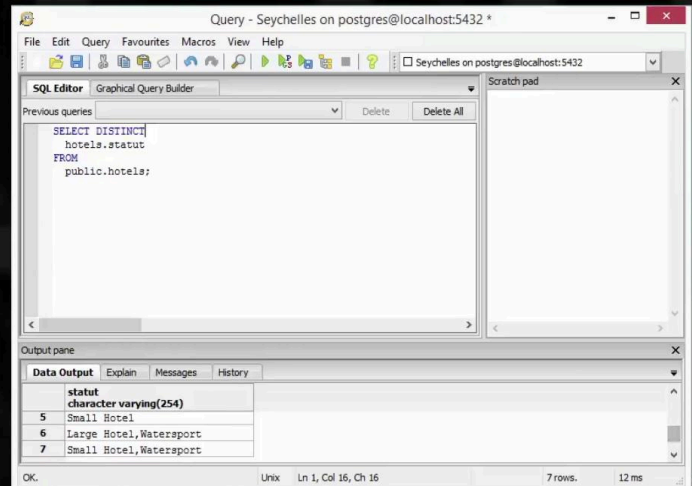
Summary



9m 09s

Sélection distincte

```
SELECT DISTINCT nom_attribut  
FROM nom_table
```



L'opération est la même dans pgAdmin. Et si on ne conserve que le champ "statut", on met à jour la requête et on ajoute le mot clé "distinct" et on voit qu'en exécutant la requête, le résultat obtenu est le même.

Notes

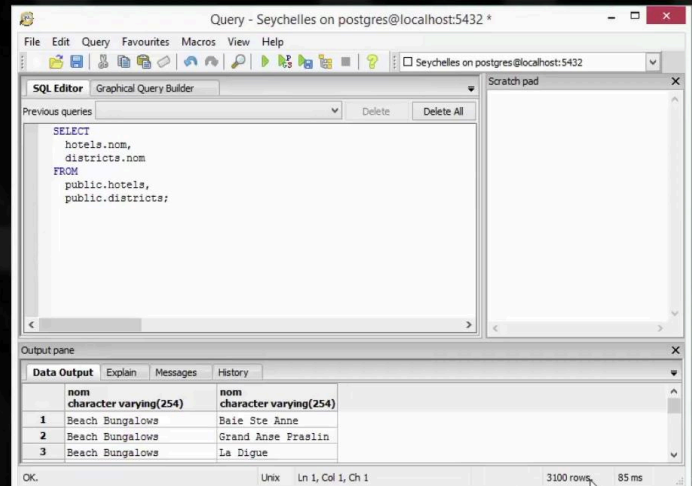
Summary



9m 59s

Sélection depuis deux tables

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



L'un des grands intérêts du langage SQL est de pouvoir associer plusieurs tables au sein d'une seule et même requête pour obtenir un résultat qui combine des données provenant de 2 tables différentes. Cette syntaxe plus explicite implique de décrire un attribut par le nom de la table dont il provient suivi du nom de l'attribut, les 2 éléments étant séparés par un point. Dans notre exemple ici, nous allons sélectionner le champ "nom" de la table "hôtels" donc décrite par `hotels.nom` et dans la table "districts", également le champ "nom" ce qui montre bien tout l'intérêt d'avoir cette syntaxe un peu plus spécifique. Donc ces 2 champs tirés des 2 tables "hôtels" et "districts". On voit que dans le résultat alors on trouve l'association systématique des noms des 2 champs et on voit que pour 124 hôtels et 25 districts, ça nous donne effectivement 3'100 résultats. Dans pgAdmin, les choses se passent de la même manière. Donc, on sélectionne le champ "nom" dans la table "hôtels". On ajoute la table "districts", on sélectionne le champ "nom" également et en passant dans l'onglet requête SQL, on voit que la requête a été rédigée de la bonne façon et l'exécution donne le même résultat avec 3'100 possibilités.

Notes

Summary

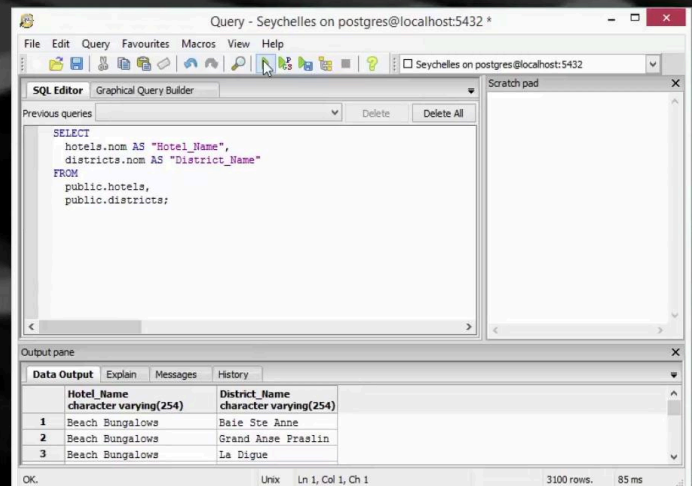


10m 37s

Alias

Alias du nom de colonne

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



Il est parfois utile de pouvoir remplacer le nom d'attribut original par un alias plus lisible ou compréhensible, opération qui s'effectue à l'aide du mot clé "as", nom d'attribut "as" nom d'alias. Dans notre exemple de tout à l'heure, nous remplaçons ici le champ "nom" de l'hôtel par Hotel_Name et le champ "nom" du district par Districts_Name ce qui permet aussi d'éliminer toute ambiguïté dans la table de résultat où l'on voit que les entêtes des colonnes portent bien les noms qui permettent de savoir à quoi on a affaire. Même chose dans pgAdmin, l'alias est précisé dans les critères, les caractéristiques du résultat de la requête et on voit que en passant en mode SQL, la requête a... la syntaxe de la requête est bien celle à laquelle on s'attend et le résultat est bien celui qu'on s'attend à trouver.

Notes

Summary

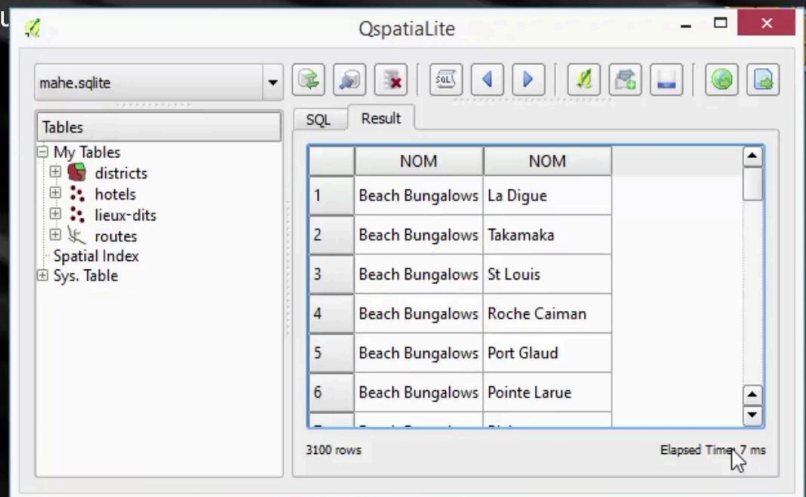


12m 22s

Alias

Alias du nom de table

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



Deuxième possibilité d'utilisation du concept d'alias, cette fois pour changer le nom des tables. Ce qui est souvent intéressant pour simplifier un petit peu la syntaxe des requêtes. Dans le cas des tables, les alias ne font pas appel au mot clé "as" mais sont simplement créés en ajoutant un nom au nom de la table avec un espace entre les 2. Dans notre exemple ici... bon nous supprimons les alias des colonnes, des attributs et nous créons des alias pour les tables, H pour la table "hôtels" et D pour la table "districts", ce qui permet de remplacer dans la définition des attributs recherchés le nom des tables par leurs alias, H pour "hôtels" et D pour "districts". Et on voit que l'exécution de cette requête donne toujours le même résultat.

Notes

Summary



13m 35s

Alias

Alias du nom de table

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

Introduction aux systèmes d'information géographique

Même opération dans pgAdmin, où l'on supprime donc les alias des attributs. Clic droit sur la table pour créer un alias de la table, H pour la table "hôtels", D pour la table "districts" comme tout à l'heure et l'on voit en revenant dans l'éditeur SQL que la syntaxe a été adaptée correctement et l'exécution donne bien le bon résultat.

Notes

Summary



14m 33s