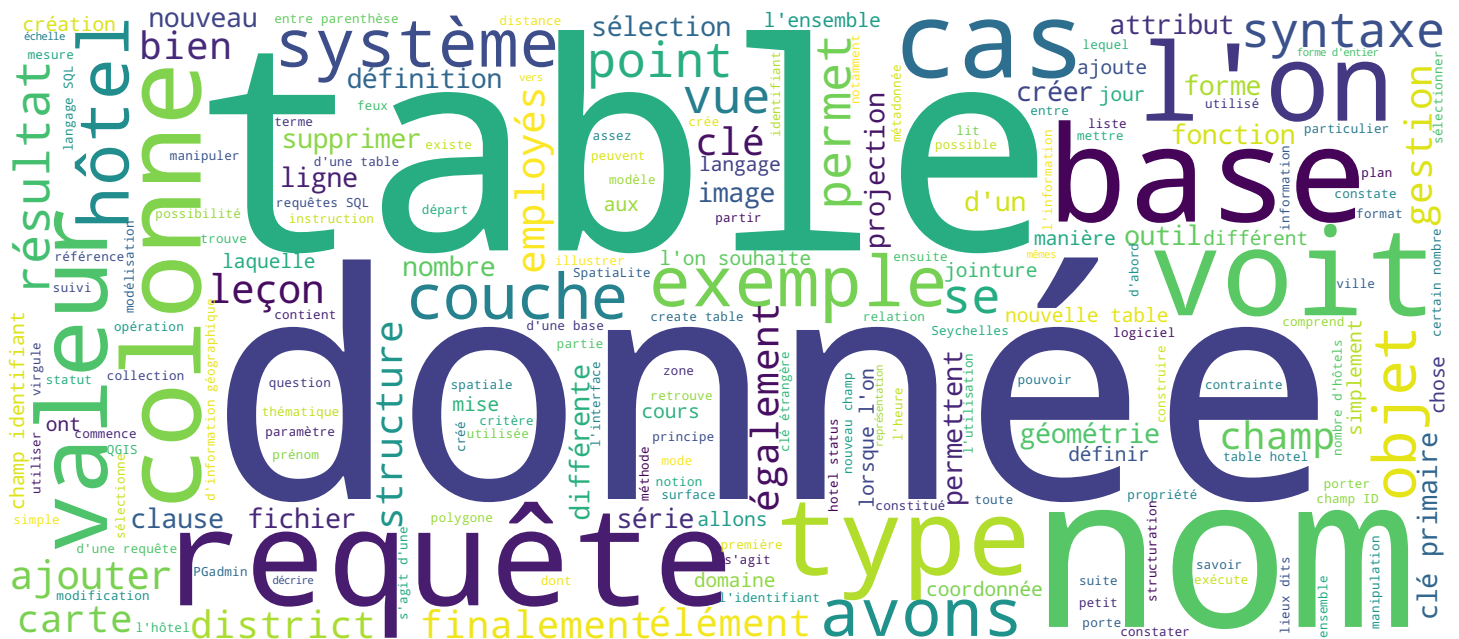


DDL, DML, Vues

Introduction aux systèmes d'information géographique

Stéphane Joost, Marc Soutter, Fernand Kouamé, Amadou Sall



Search MOOC



Video



DDL, DML, Vues

Objectifs de la leçon

- Découvrir les fonctions SQL de définition et de manipulation des données

Après cette leçon vous serez capables

- D'utiliser le langage SQL pour créer et/ou modifier une structure de données
- D'utiliser le langage SQL pour manipuler les données d'une base de données

Introduction aux systèmes d'information géographique

Bienvenue dans cette leçon qui va porter sur ces éléments particuliers du langage SQL que sont le langage de définition de données ou data definition language, DDL, et le langage de manipulation de données ou data manipulation language, DML. Les outils de structuration de données qui permettent de créer des tables avec des colonnes de types bien définis de même que les outils d'édition de données attributaires que nous avons vu dans les leçons précédentes, sont autant d'éléments qui ne font finalement rien d'autre que d'implémenter des requêtes de type DML ou DDL. Requête que nous allons donc examiner au cours de la présente leçon. Les objectifs de cette leçon consistent donc à découvrir les fonctions de ces langages de définition et de manipulation de données, de sorte à être capable ensuite d'utiliser les requêtes SQL pour créer et manipuler des structures de données et d'utiliser des requêtes SQL pour manipuler les données mêmes qui sont contenues dans les bases de données.

Notes

Summary

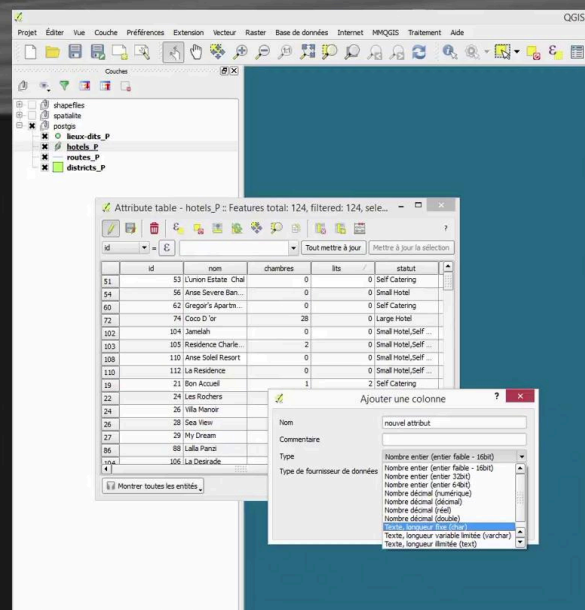


0m 22s

SQL vs structuration et édition de données

Structuration / définition de données

● QGIS



Introduction aux systèmes d'information géographique

Dans cette leçon nous allons commencer par revenir un petit peu sur les outils de structuration et d'édition de données classiques que nous avons vu jusqu'à présent avant d'aborder plus spécifiquement la question du langage de définition de données puis celle du langage de modification de données et nous terminerons par un aspect particulier dont nous n'avons pas encore parlé qui sont les vues, qui sont un autre composant des bases de données. Si l'on revient à l'idée de structuration et de définition de données on se souvient que le logiciel QGIS propose des éléments, des outils qui permettent directement de créer des tables, de modifier ces tables, en créant des champs, des colonnes, de type défini dans le cas d'une base de données SpatiaLite aussi bien que dans le cas d'une base de données PostGIS Postgres.

Notes

Summary

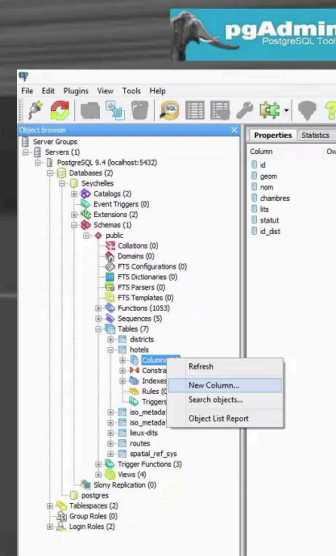


1m 32s

SQL vs structuration et édition de données

Structuration / définition de données

- QGIS
- SGBD



Introduction aux systèmes d'information géographique

On retrouve le même genre de fonctionnalité avec les systèmes de gestion de bases de données, aussi bien pour Spatialite que pour PGAdmin donc des interfaces de gestion de bases de données qui permettent de manipuler la structure de données.

Notes

Summary

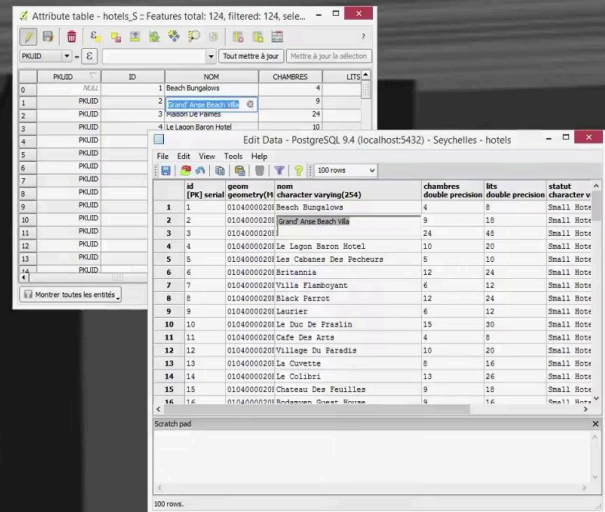


2m 33s

SQL vs structuration et édition de données

Edition de données

- QGIS
- SGBD
- ➔ Implémentation de requêtes SQL
- ➔ Offre davantage de possibilités



Introduction aux systèmes d'information géographique

Finalement, nous avons aussi abordé la question des outils CASE qui permettent en fait de fabriquer et de gérer des structures de données et de les transformer éventuellement en base de données physique réelle. Tous ces divers outils que nous avons vu à différentes occasions dans les précédentes leçons du cours implémentent toujours simplement des requêtes SQL. Il s'agit toujours simplement d'un habillage dont le moteur est constitué par des requêtes SQL. Dans le domaine de l'édition de données attributaires nous avons également vu que les logiciels SIG proposent des outils d'édition de même que les systèmes de gestion de bases de données. Dans ce cas également le moteur de cette édition de données est constitué par des requêtes SQL comme on le verra tout à l'heure dans la leçon. L'avantage du SQL dans le domaine de l'édition de données c'est qu'il permet de développer des approches un peu plus élaborées que l'édition d'un attribut à la fois.

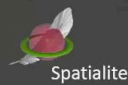
Notes

Summary

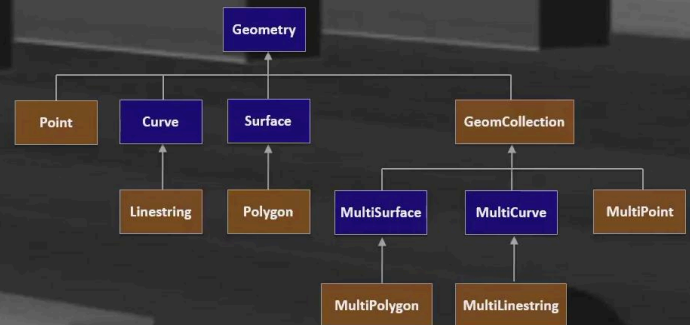


2m 50s

Définition des données



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)))	



Introduction aux systèmes d'information géographique

La première et la plus fondamentale des choses que l'on souhaite faire dans le domaine de la définition des données c'est bien la création de tables avec l'instruction "create table". Cette instruction permet de créer une table sur la base de la définition de ses colonnes par leurs noms et par le type de données qu'elles vont contenir. Donc la syntaxe est du type "create table », le nom de la table entre parenthèses, la série des colonnes par leurs noms et leurs types, séparées par une virgule. Le type de colonne que l'on peut créer est très variable. Nous avons ici à gauche un tableau de synthèse des types utilisables dans une base de données SQLite avec l'intérêt d'avoir résumé en fait une grande variété de types différents en 5 catégories qui sont l'entier, le texte, le blob pour les éléments binaires, comme par exemple les images, les valeurs réelles et les éléments numériques. Du côté droit, on a un tableau des différents types de données que l'on peut définir dans une base de données postSQL ou postGIS et on voit que la liste est assez considérable.

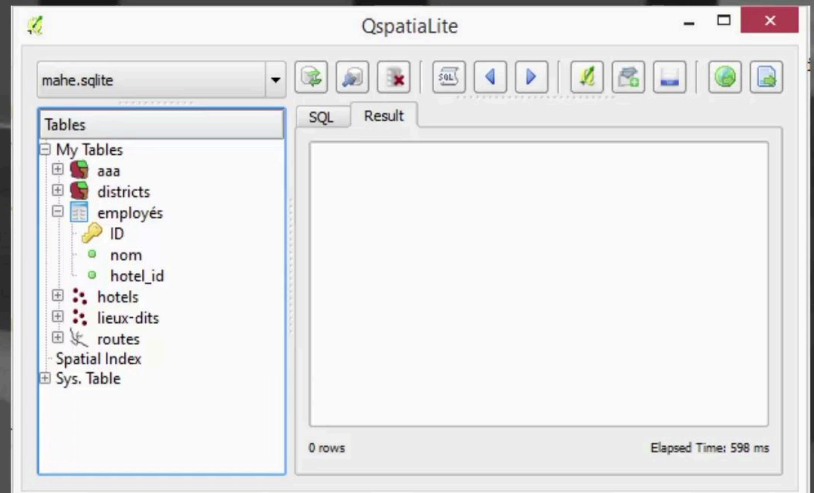
Notes

Summary



Définition des données

CREATE TABLE



Comme exemple nous allons créer une table des employés des hôtels des Seychelles avec quelques champs identifiant le nom, l'hôtel dans lequel l'employé travaille. On va définir une clé primaire et une clé étrangère. Alors tout d'abord l'identifiant sous forme d'entier puis le nom sous forme format texte, l'identifiant de l'hôtel, également sous forme d'entier, après quoi on va définir la clé primaire de cette table. Clé primaire qui sera donc l'identifiant de l'employé, ID. Puis une clé étrangère pour exprimer le fait que le champ hotel_id référence en fait le champ identifiant de la table hotel qui est une autre table. La requête étant écrite en execut et on voit que cela crée effectivement cette table des employés avec les trois champs, avec le champ identifiant comme clé primaire.

Notes

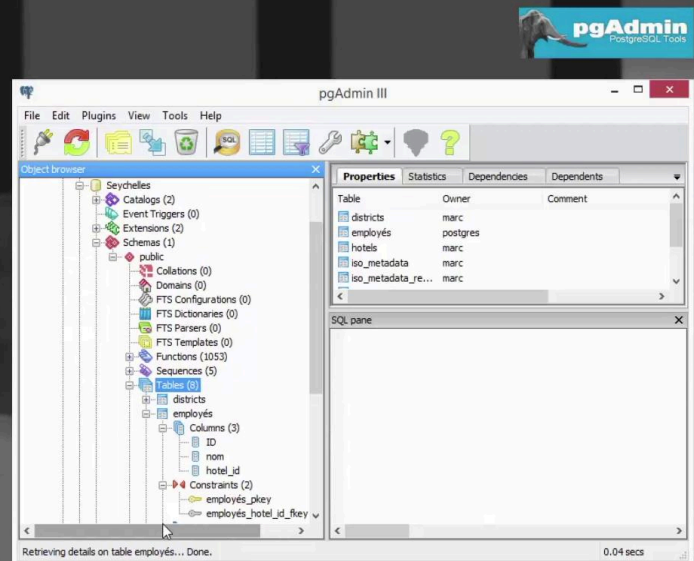
Summary



5m 19s

Définition des données

CREATE TABLE



Le même type d'opération dans le cas de PGadmin se fait de manière graphique au travers de l'interface de manipulation de données donc on crée une nouvelle table des employés auquel on va ajouter un certain nombre de champs, donc d'abord le champ identifiant sous forme d'entier. On ajoute ensuite le nom des employés sous forme de texte puis comme précédemment on va ajouter un champ "identifiant de l'hôtel" pour faire la jointure avec la table des hôtels, donc il s'agit ici également d'un entier. Dans les contraintes, on va ajouter la clé primaire, donc on a dit qu'on prenait le champ ID de cette nouvelle table et puis on va ajouter une clé secondaire que l'on va définir en référençant, en liant le champ ID hotel de la table des employés avec le champ ID de la table des hôtels. On voit que l'onglet SQL permet de consulter la syntaxe de la requête qui a été créée en manipulant ces objets graphiques. L'exécution de cette requête conduit bien à créer la table des employés avec ses trois colonnes et puis ses deux contraintes sous forme de clé primaire et de clé étrangère.

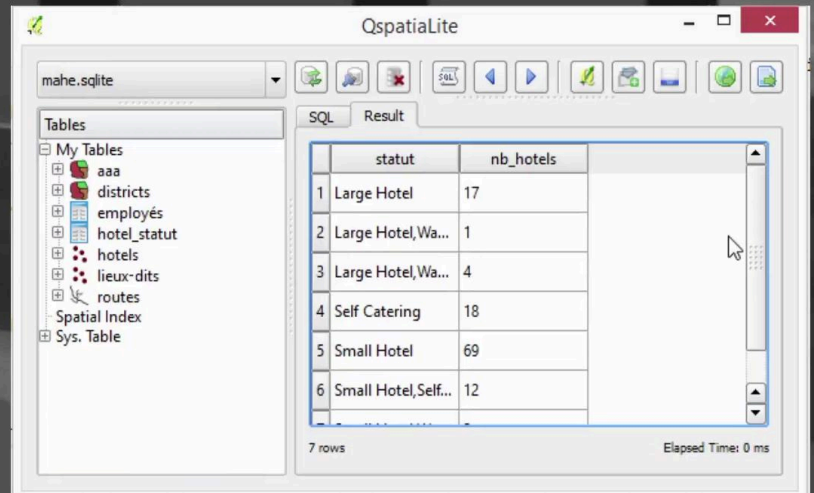
Notes

Summary



Définition des données

CREATE TABLE AS
SELECT INTO



The screenshot shows the QspatialLite application window. On the left, a tree view lists tables: 'My Tables' (containing 'aaa', 'districts', 'employés', 'hotel_statut', 'hotels', 'lieux-dits', 'routes', 'Spatial Index', and 'Sys. Table'). The main area displays a table with two columns: 'statut' and 'nb_hotels'. The table contains 7 rows of data.

	statut	nb_hotels
1	Large Hotel	17
2	Large Hotel, Wa...	1
3	Large Hotel, Wa...	4
4	Self Catering	18
5	Small Hotel	69
6	Small Hotel, Self...	12
7		

7 rows
Elapsed Time: 0 ms

Un cas un peu particulier de création de table avec l'instruction "create table as" qui permet en fait de transformer le résultat d'une requête de sélection en une table nouvelle. La syntaxe générale est "create table ", nom de la table, AS, la requête de sélection que l'on souhaite stocker comme nouvelle table. Il existe cependant une syntaxe alternative sous la forme d'une requête de type "select into" où en fait il s'agit d'une simple requête dans laquelle le fait d'ajouter le mot-clé INTO suivi du nom de la nouvelle table que l'on souhaite créer, conduit au même résultat à savoir de stocker le résultat de la requête dans une nouvelle table. Pour illustrer ces opérations de création de table à partir de requêtes de sélection nous repartons de cette requête qui donnait en fait le nombre d'hôtels dans chaque type de statut. On avait donc sept catégories, sept types de statut d'hôtels avec le nombre d'éléments associés. On ajoute alors simplement la clause, l'instruction "create table". Et on va appeler cette table "hotel status" avec le mot clé AS et puis la requête qui suit. On exécute cette requête et on voit qu'on a créé une nouvelle table qui comprend les deux champs "statut" et "nombre d'hôtels" et lorsque l'on affiche en fait les données de cette table on voit qu'on retrouve en fait nos sept catégories avec le nombre d'hôtels correspondant.

Notes

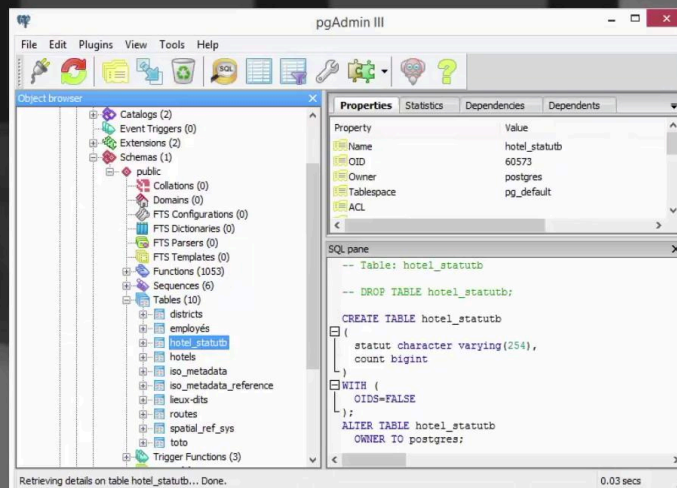
Summary



7m 49s

Définition des données

CREATE TABLE AS SELECT INTO



La même chose dans le monde PGadmin où l'on a la même requête et pareillement on ajoute simplement la clause "create table", le nom de la table "hotel status", AS et la requête originelle de sélection derrière. On voit que lorsqu'on exécute cette requête et que l'on revient dans l'interface de consultation de la base de données, on a créé ce champ "hotel status" qui contient bien ces deux colonnes et lorsque l'on consulte les données on trouve bien les données recherchées. La syntaxe alternative consiste simplement à rajouter le mot-clé INTO avec le nom de la nouvelle table que l'on souhaite créer au milieu de la requête de sélection pour obtenir en fait le même résultat avec ici une table que l'on a baptisé "hotel status B" et qui contient les mêmes champs et les mêmes valeurs.

Notes

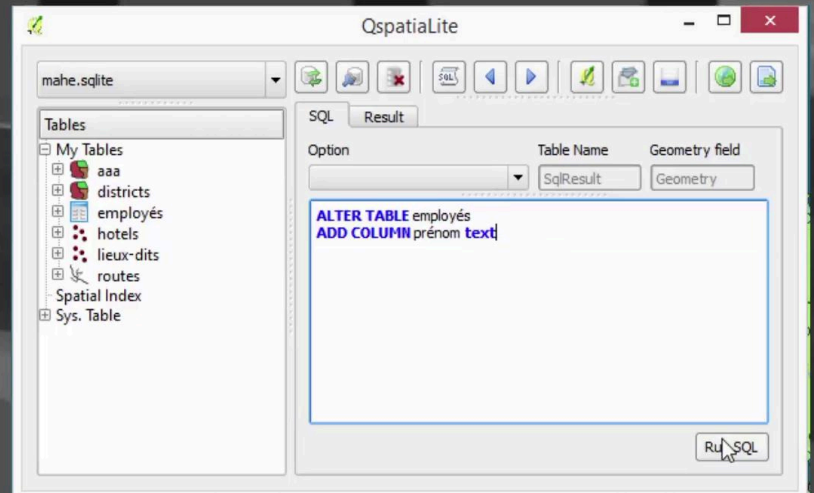
Summary



9m 41s

Définition des données

ALTER TABLE



Seconde opération intéressante dans le domaine de la définition des données lorsqu'une table a été créée, qu'elle existe, on peut vouloir la modifier, modifier sa structure en ajoutant ou en supprimant une colonne ou en modifiant le nom ou le type d'une colonne. On a donc plusieurs types de syntaxe qui sont possibles avec l'instruction "alter table", nom de la table, suivi d'une instruction qui peut être soit ADD, le nom de la colonne et le type de données, soit DROP, le nom de la colonne pour supprimer une colonne, soit "alter column" avec le nom de la colonne que l'on souhaite changer et le nouveau type de données qui suit le mot-clé type. Pour illustrer cette opération on reprend la table des employés et on va ajouter une colonne pour contenir par exemple le prénom au format texte.

Notes

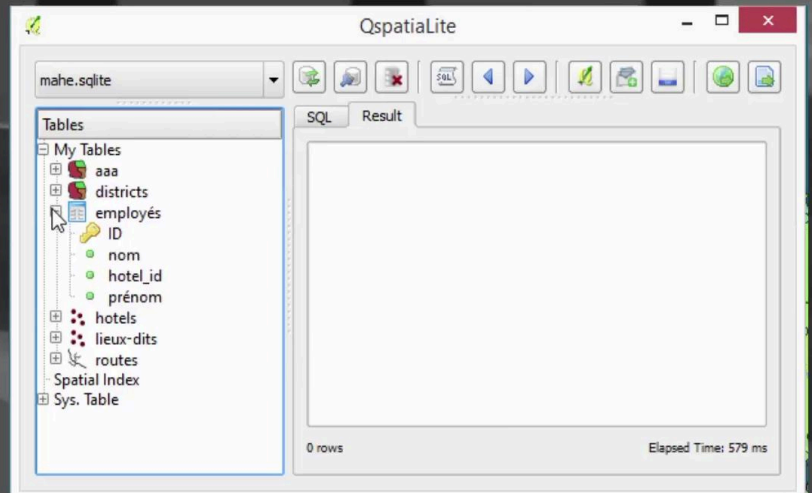
Summary



10m 50s

Définition des données

ALTER TABLE



On exécute la requête et on constate effectivement que ce nouveau champ a été ajouté.

Notes

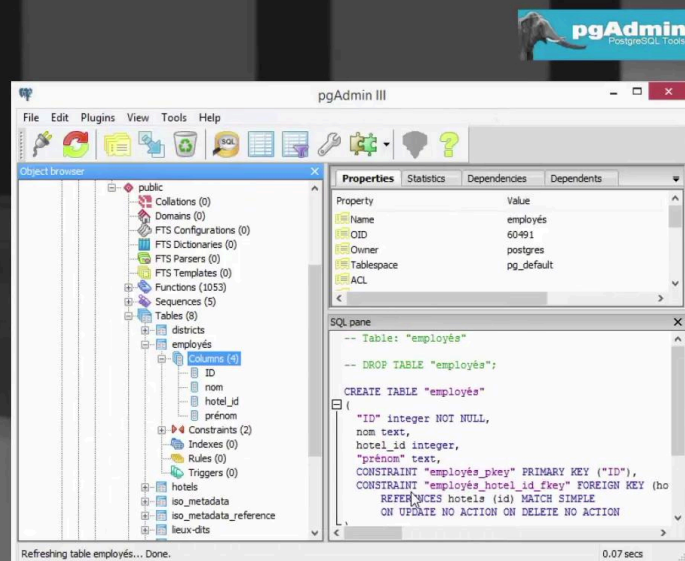
Summary



11m 47s

Définition des données

ALTER TABLE



Même chose dans PGadmin mais cette fois-ci à partir de l'interface d'édition de la structure de données, on ajoute une colonne, on définit son nom, son type, prénom et le type texte et on voit que l'onglet SQL donne l'équivalent SQL de l'instruction qu'on prépare. On exécute cette instruction et on crée ce nouveau champ.

Notes

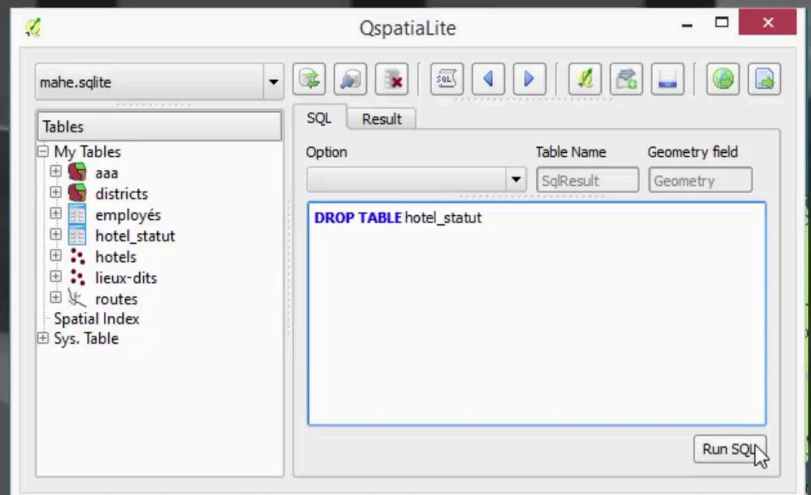
Summary



11m 52s

Définition des données

DROP TABLE, TRUNCATE TABLE



Et finalement, deux instructions qui permettent de supprimer une table avec "drop table" ou d'effacer le contenu d'une table tout en préservant sa structure avec "truncate table". La syntaxe est toute simple, l'instruction suivi du nom de la table sur laquelle elle porte. A titre d'exemple on peut simplement prendre "drop table" pour supprimer la table que l'on a créé tout à l'heure, la table des statuts des hôtels.

Notes

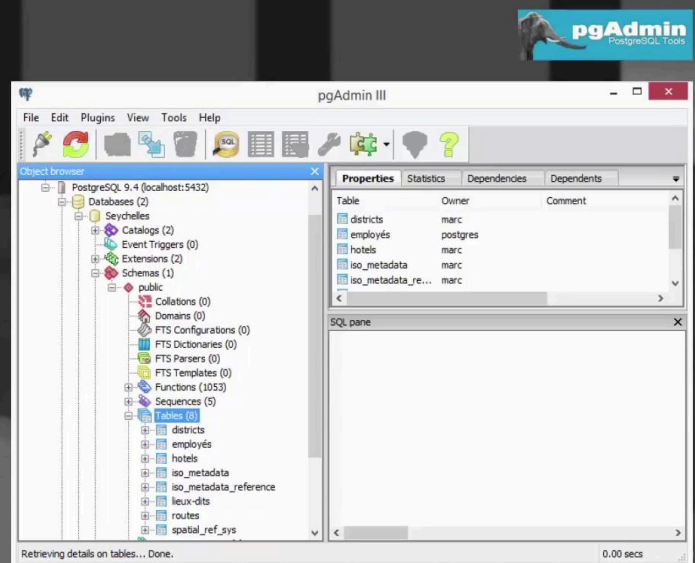
Summary



12m 14s

Définition des données

DROP TABLE, TRUNCATE TABLE



Dans le domaine SQL simplement la fonction "truncate" qui permet de constater que la table a été vidée et l'instruction "drop" pour supprimer la table elle-même.

Notes

Summary



12m 50s

Manipulation des données

INSERT INTO

- Insertion de données dans une table,
- Plusieurs syntaxes

➡ Une ligne à la fois

En spécifiant toutes les colonnes

```
INSERT INTO nom_table VALUES ( 'valeur1', 'valeur2', ... )
```

En ne spécifiant qu'une sélection de colonnes

```
INSERT INTO nom_table ( nom_colonne1, nom_colonne2, ... )  
VALUES ( 'valeur1', 'valeur2', ... )
```

Introduction aux systèmes d'information géographique

Dans le registre de la manipulation des données ce que l'on souhaite faire en tout premier lieu, on a créé une structure de données, on a des tables et on aimerait ajouter des données dans ces tables avec l'instruction "insert into". Cette instruction permet d'insérer des données dans une table. Elle connaît plusieurs syntaxes selon la manière dont on procède. Une première série de syntaxes concerne l'insertion d'une ligne de données à la fois et ça, ça peut se faire de deux manières soit en donnant une valeur pour chacune des colonnes de la table, avec une syntaxe de type "insert into", le nom de la table, le mot-clé VALUES et puis la série des valeurs, une valeur pour chaque colonne de la table, séparée par des virgules et entre parenthèses. Autre possibilité, on ne veut pas documenter, renseigner tous les champs, toutes les colonnes de la table et on peut spécifier en fait à quelle colonne correspondent les valeurs que l'on donne comme argument avec une syntaxe de type "insert into", le nom de la table, la série des noms des colonnes concernés par les valeurs qui vont suivre, entre parenthèses et séparées par des virgules, le mot-clé VALUES et de nouveau comme précédemment entre parenthèses, la série des valeurs qui correspond à l'ordre des colonnes listées précédemment.

Notes

Summary



13m 19s

Manipulation des données

INSERT INTO

➡ Plusieurs lignes à la fois

```
INSERT INTO nom_table VALUES  
( 'valeur1', 'valeur2', ... ),  
( 'valeur1', 'valeur2', ... ),  
( 'valeur1', 'valeur2', ... )
```

- Lorsque le champ ajouté est de type texte, il faut indiquer le contenu entre guillemets simples
- S'il s'agit d'un type numérique les guillemets ne sont pas nécessaire

Introduction aux systèmes d'information géographique

Autre possibilité, le cas où on veut insérer plusieurs lignes à la fois avec une seule instruction, donc avec une syntaxe de type "insert into", le nom de la table, le mot-clé VALUES et une série de vecteurs de valeurs avec une valeur par colonne de la table donc des vecteurs de valeur entre parenthèses, séparés par des virgules. On a vu dans toutes ces instructions que lorsque l'on ajoute un champ de type texte il faut indiquer le contenu entre guillemets, des guillemets simples. Dans le cas où il s'agit de valeurs numériques les guillemets ne sont pas nécessaires.

Notes

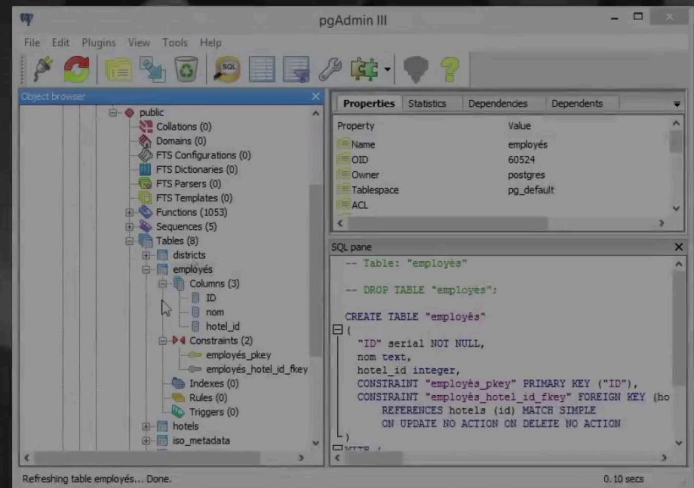
Summary



14m 43s

Manipulation des données

INSERT INTO



Comme exemple de l'utilisation de ces instructions d'insertion de données on va ajouter des données à la table des employés. On commence par ajouter un premier élément avec le dénommé Pierre et l'identifiant 1. On voit qu'on n'a pas mentionné l'ensemble des champs de la table mais leur ordre et par défaut, la valeur manquante a été placée comme nulle. On peut également vouloir introduire les données dans un ordre différent en commençant par le prénom de la personne et ensuite l'identifiant. Alternativement, on peut vouloir, en respectant de nouveau l'ordre des champs dans la table, introduire plusieurs éléments à la suite. On voit qu'avec tout ça dans notre table des employés nous avons maintenant les 4 employés que nous avons ajouté. La principale difficulté qui se pose avec ce type d'ajout tient au choix de l'identifiant. Si on a tout d'un coup une centaine d'éléments qu'on aimerait ajouter à la table, il est un peu fastidieux de définir à chaque fois le bon identifiant. Il faut savoir où on en est dans la liste de ces identifiants pour être sûr qu'il s'agit d'une valeur unique.

Notes

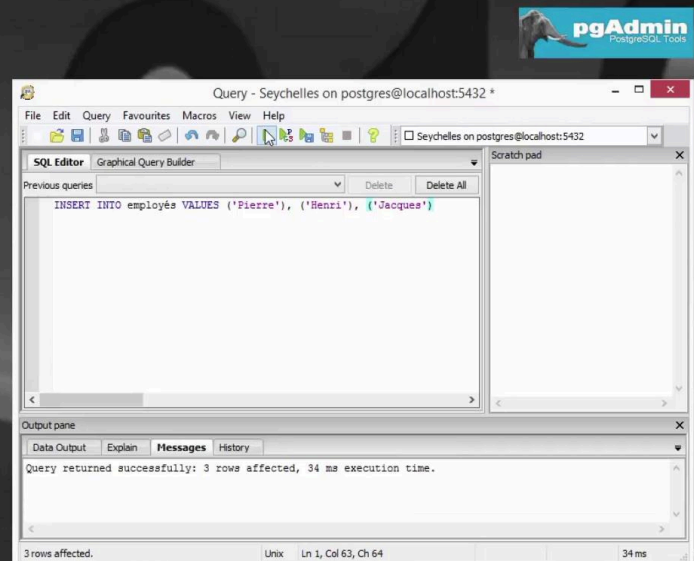
Summary

15m 24s



Manipulation des données

INSERT INTO



Pour pouvoir effectuer ce genre d'opérations il nous faut donc redéfinir le champ identifiant, donc on commence par supprimer le champ existant et on ajoute une nouvelle colonne avec ce nouveau champ identifiant qui ne sera plus de type entier mais de type serial. On ajoute dans la définition de la table ce nouveau champ identifiant comme clé primaire et on peut encore, pour faire bonne mesure, nettoyer la table existante, donc on sélectionne les éléments existants, on les supprime et on repart d'une table vide. Il devient alors possible d'écrire une requête d'insertion de données multiples qui va ajouter en fait à la table des employés, dans la rubrique nom, l'ensemble des résultats d'une autre requête qui va consister à rechercher les noms des hôtels tirés de la table des hôtels pour lesquels le nom de l'hôtel est de type Chez Albert, André François, Paul ou quelqu'un d'autre. Cette requête effectuée, on revient à l'interface de gestion des données, on constate qu'on a ajouté 3 éléments automatiquement et que l'identifiant a été ajouté automatiquement également. On voit qu'avec cette nouvelle définition de l'identifiant non plus comme un entier mais comme une valeur de type serial, il devient possible d'ajouter, d'insérer des employés en donnant simplement la liste des prénoms, séparés par des virgules.

Notes

Summary

16m 55s



Manipulation des données

UPDATE

- Modifications de lignes / tuples existants
- Souvent associée à une clause WHERE pour spécifier sur quelle(s) ligne(s) doit porter la modification.
- Syntaxe

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

Introduction aux systèmes d'information géographique

L'instruction "update" permet de mettre à jour les valeurs de champs existants dans une table. Elle est souvent associée à une clause WHERE afin de spécifier sur quelle ligne doit porter la modification et éventuellement de propager une mise à jour sur plusieurs lignes qui satisferaient un même critère. La syntaxe est donc du type "update", le nom de la table, "set", le nom de la colonne et la valeur à attribuer à cet attribut, et WHERE, la condition qui permet de sélectionner, de filtrer les lignes de la table sur lesquelles il faut mettre à jour l'attribut "nom colonne" avec la valeur donnée en argument.

Notes


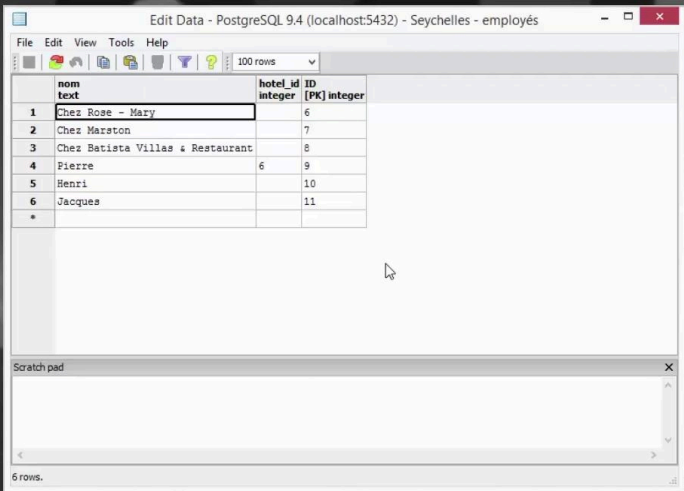
Summary



18m 46s

Manipulation des données

UPDATE

	nom text	hotel_id integer [PK]	ID integer
1	Chez Rose - Mary	6	
2	Chez Marston	7	
3	Chez Batista Villas & Restaurant	8	
4	Pierre	6	9
5	Henri		10
6	Jacques		11
*			

Dans le cas de la table des employés on peut faire porter cette mise à jour en définissant en fait simplement l'identifiant de l'hôtel pour l'employé Pierre, paramètre qui n'a pas encore été ajouté. On constate, si on affiche, après avoir exécuté la requête, la table, on voit que cet identifiant est présent.

Notes

Summary

19m 30s



Manipulation des données

DELETE FROM

- Supression de lignes / tuples
- Souvent associée à une clause WHERE
- Syntaxe

```
DELETE FROM nom_table WHERE condition
```

Introduction aux systèmes d'information géographique

L'instruction "delete from", qui permet de supprimer des enregistrements ou des lignes d'une table de données, qui est à nouveau comme l'instruction de mise à jour souvent associée à une clause WHERE, qui définit la condition de filtre des lignes d'une table que l'on va supprimer. La syntaxe sera du type "delete from", le nom de la table, WHERE, la condition que doivent remplir les lignes qui vont être supprimées.

Notes

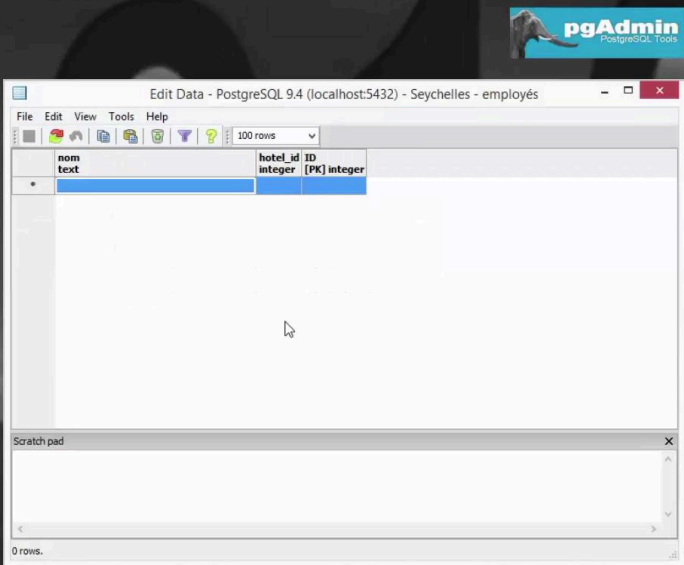
Summary



19m 58s

Manipulation des données

DELETE FROM



Comme exemple tout simple nous reprenons la table des employés de laquelle nous allons supprimer la ligne qui porte le nom Pierre. On voit qu'une seule ligne est affectée et lorsque l'on consulte les données on voit qu'effectivement l'information concernant Pierre a disparu. On voit qu'on peut aussi supprimer des lignes beaucoup plus directement en utilisant les fonctions prévues à cet effet de l'interface PGAdmin.

Notes

Summary

20m 28s



Les vues

CREATE VIEW, DROP VIEW

- Une vue est une table virtuelle
- Regroupe des données provenant de plusieurs tables pour des motifs de représentation
- Pas d'édition de données
- Syntaxe

```
CREATE VIEW nom_vue AS requête
```

```
DROP VIEW nom_vue
```

Introduction aux systèmes d'information géographique

Il est souvent intéressant de pouvoir stocker et réutiliser les résultats d'une requête, notamment pour construire des requêtes emboîtées par exemple, avec l'inconvénient toutefois lorsque l'on utilise les fonctions de créations de tables, "create as" et "select into", de créer une duplication des données ce qui pose des problèmes de mise à jour. L'intérêt de l'objet "vue" c'est qu'il s'agit d'une table virtuelle, donc c'est un extrait non-éditable de données existantes, donc simplement un outil de présentation de données un peu différente. Les données originales restant stockées dans les tables de départ. Il n'y a donc pas d'édition de données dans ces vues. La syntaxe est toute simple "create view", nom de la vue, AS et la requête de sélection qui réunit les différentes données que l'on souhaite présenter dans la vue et "drop view" pour supprimer cette vue.

Notes

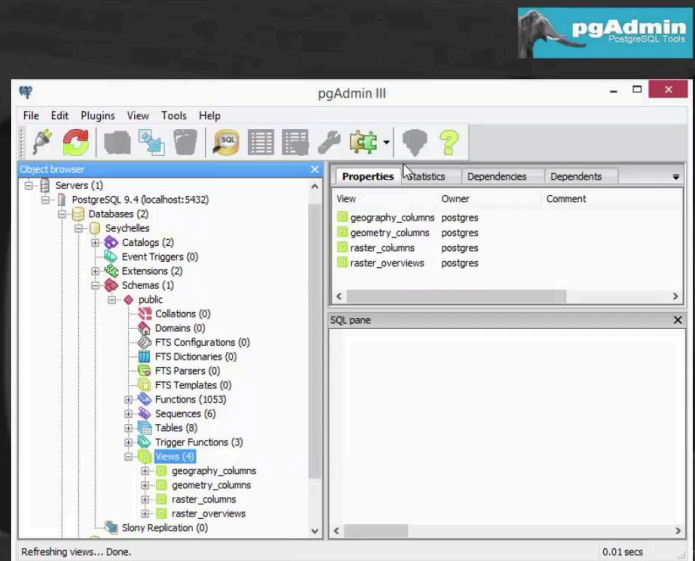
Summary



21m 06s

Les vues

CREATE VIEW, DROP VIEW



Pour illustrer ces instructions de création et de suppression de vues nous allons créer une vue test comme un simple reflet de la table des hôtels, donc on sélectionne l'ensemble des champs de la table hotels et on crée cette vue. On constate que cette nouvelle vue est disponible dans la collection des vues et si on regarde ces données on retrouve les hôtels des Seychelles mais dans une version qui n'est pas éditable. Ce qui est intéressant c'est de constater que l'on peut ensuite, lorsque l'on fait d'autres requêtes, on peut retrouver cette vue dans les éléments utilisables pour construire des requêtes de la même manière qu'on pourrait utiliser des tables. La suppression, soit par l'élément de menu intégré à PGadmin ou bien sous forme d'instruction SQL, "drop view", le nom de la vue et on constate que la vue a bel et bien disparu.

Notes

Summary

