

# Systèmes d'Information Géographique

<https://go.epfl.ch/sig>

**Requêtes et langage SQL** (Structured Query Language)

Stéphane Joost, Gabriel Kathari (GEOME-LGB)

# Langage SQL

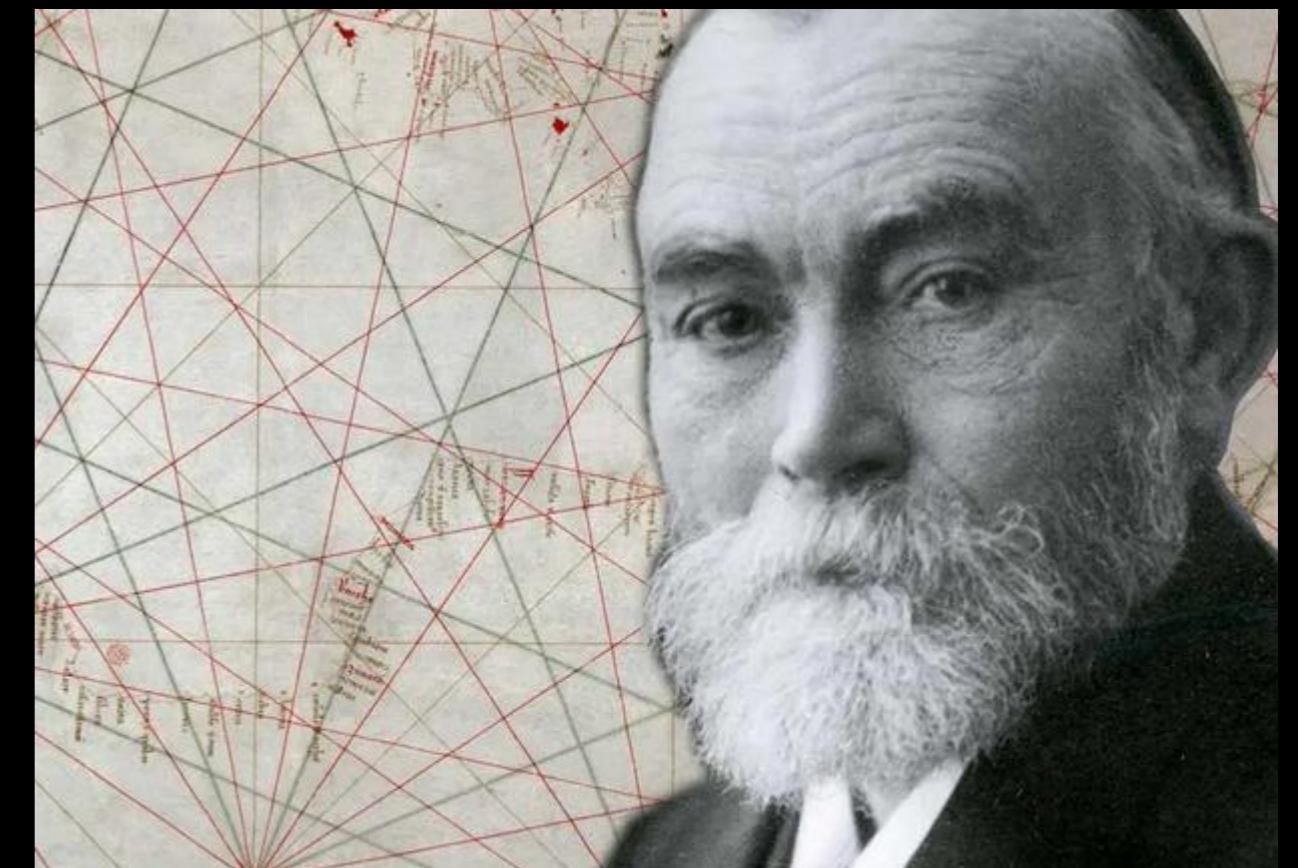
- En juin 1970, Edgar Frank Codd publie un article important
- « Un référentiel de données relationnel pour de grandes banques de données partagées » dans la revue *Communications of the ACM* (Association for Computing Machinery)
- Référentiel relationnel fondé sur la **logique des prédictats du premier ordre**
- Reconnu comme un modèle théorique utile pour l'interrogation des bases de données
- Ce référentiel a inspiré le développement du langage « **Structured English QUERy Language** » (SEQUEL)



# Logique des prédictats

---

- Domaine de la logique mathématique
- Calcul des prédictats du premier ordre (logique du premier ordre, calcul des relations, ou calcul des prédictats)
- Système formel utilisé pour raisonner et décrire des énoncés en mathématiques, informatique, intelligence artificielle, philosophie et linguistique
- Il a été proposé par Gottlob Frege comme une formalisation du langage des mathématiques (fin du XIXe et début du XXe siècle).



# Logique du premier ordre ou FOL - *First-Order Logic*

---

La logique du premier ordre est une extension de la logique propositionnelle qui permet:

- Des variables (ex :  $x, y$ ) représentant des objets dans un domaine.
- Des quantificateurs :
  - $\forall x$  (pour tout  $x$ ) : une assertion doit être vraie pour tous les objets.
  - $\exists x$  (il existe un  $x$ ) : une assertion est vraie pour au moins un objet.
- Des prédictats qui décrivent des relations entre objets  
ex:  $Ami(x, y)$  signifie que  $x$  est ami avec  $y$ .
- Des constantes et des fonctions pour désigner des entités et des transformations.

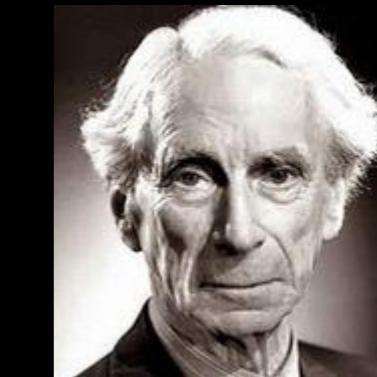
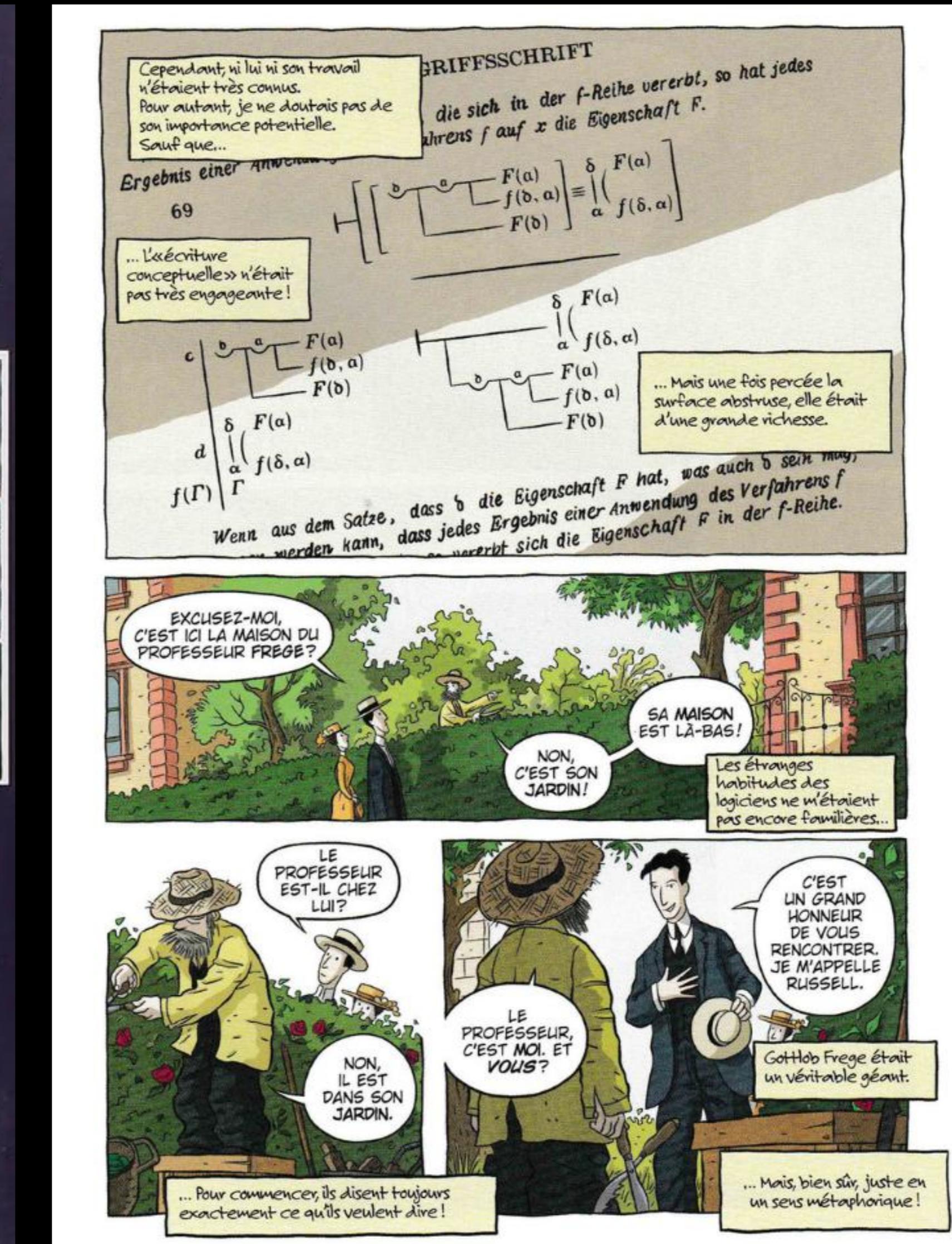
# LOGICOMIX



APOSTOLOS DOXIADIS  
CHRISTOS PAPADIMITRIOU  
ALECOS PAPADATOS ANNIE DI DONNA

Vuibert

La quête des fondements mathématiques



Bertrand Russell

«Principia Mathematica»

Naissance de la logique moderne

Introduction aux systèmes d'information géographique

# Langage SQL

---

- SEQUEL a été développé chez IBM en 1970
- Cette première version a été conçue pour manipuler et éditer des données stockées dans une base de données relationnelle à l'aide du système de gestion de base de données IBM appelé « **System R** »
- SEQUEL renommé SQL pour cause de conflit de marque déposée (mais on devrait toujours prononcer « sikouel »)
- En 1979, Relational Software, Inc. (actuellement Oracle Corporation) présente la première version commercialement disponible de SQL
- SQL a été adopté par l'Institut de normalisation américaine (ANSI) en 1986, puis comme norme internationale par l'ISO en 1987 (Norme internationale SQL)

# Syntaxe générale

---

- Les instructions SQL s'écrivent d'une manière qui ressemble à celle de phrases ordinaires en anglais. Cette ressemblance voulue vise à faciliter l'apprentissage
- C'est un **langage déclaratif**: il permet de décrire le résultat escompté, sans décrire la manière de l'obtenir
- Par exemple, les pages HTML étaient uniquement déclaratives à la base. Elles décrivaient ce que contenait une page (texte, titres, paragraphes, etc.)
- Et non comment les afficher (positionnement, couleurs, etc.)
- Au contraire, en **programmation impérative** (C ou Java), on décrit le comment (c'est-à-dire la structure de contrôle correspondant à la solution)

# Le SQL

---

- Le SQL est un langage destiné à la gestion des **bases de données relationnelles** spécifiquement
- C'est un langage qui a été normalisé et qui est en principe **indépendant du système** de gestion de base de données utilisé
- Toutefois chaque système de gestion de base de données (SGBD) peut avoir des spécificités au niveau de la syntaxe utilisée
- C'est un langage qui permet d'interagir avec des bases des données sous forme de requêtes structurées

# Le SQL

---

Le SQL est composé de 4 groupes d'instructions complémentaires:

1. En premier lieu le **Data Query Language (DQL)**, langage d'interrogation des données, qui permet d'extraire des données d'une base des données
2. Le **Data Definition Language (DDL)**, langage de définition de données qui permet de modifier ou de définir la structure d'une base de données
3. Le **Data Manipulation Language (DML)** qui permet d'insérer, de mettre à jour, de supprimer des données
4. Le **Data Control Language (DCL)** qui permet de gérer les droits et les accès des utilisateurs

# Le SQL

## Interrogation des données

1. Data Query Language – DQL  
permet d'extraire des données d'une base des données
2. Data Definition Language – DDL
3. Data Manipulation Language – DML
4. 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' Unieville	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



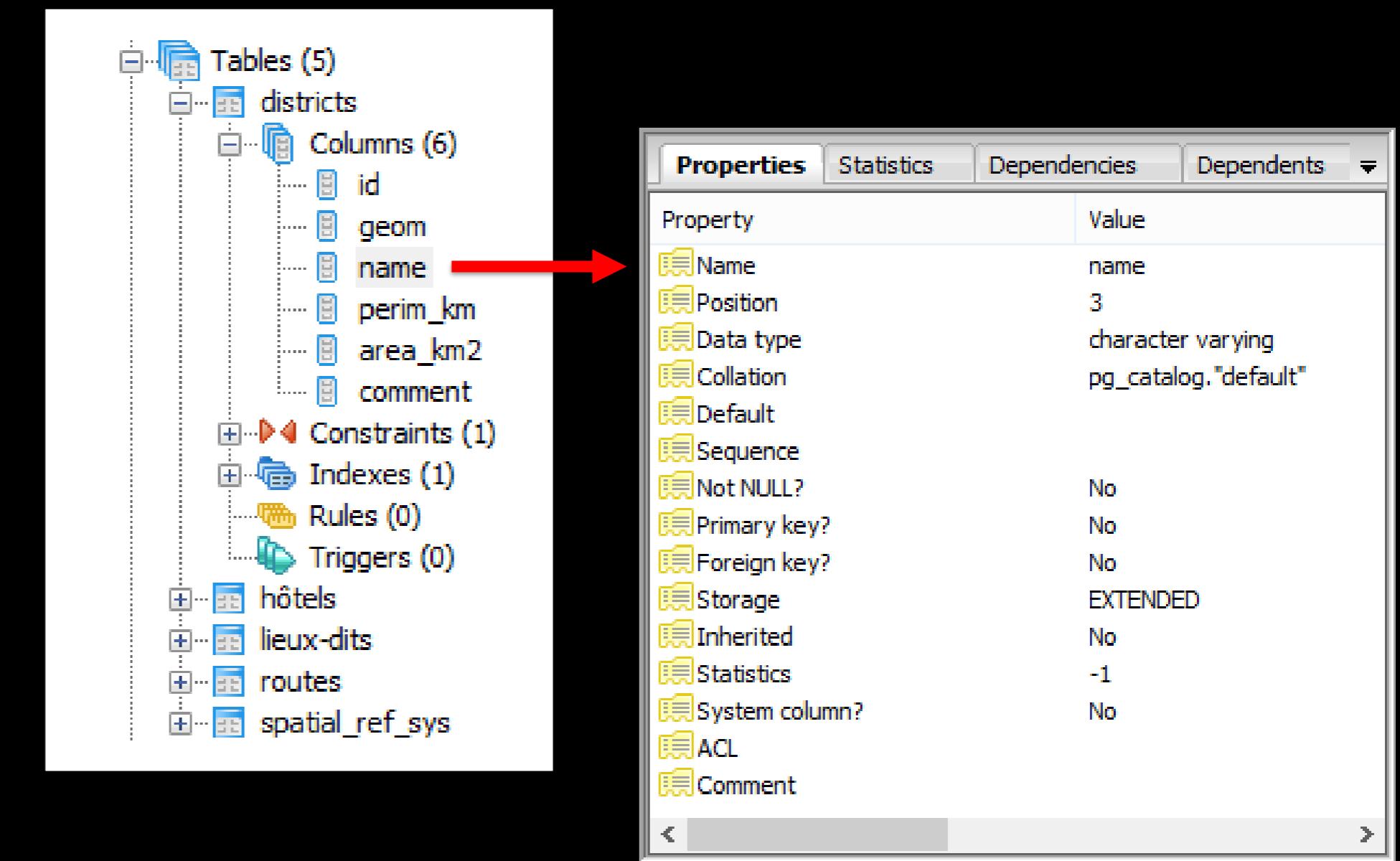
	ID	NAME	ROOMS	MANAGER	ADDRESS
2	3	Maison De Palme...	24	Mr. Basil Ferrari	Amitie
3	4	Le Lagon Baron ...	10	Mr. Karl D' Unieville	Grand' Anse
5	6	Britannia *	12	Mrs. L . Lablache	Grand' Anse

Interroger les données /  
Extraire des données

# Le SQL

## Interrogation des données

1. Data Query Language – DQL
2. Data Definition Language – DDL  
permet de modifier ou de définir la structure d'une base de données
3. Data Manipulation Language – DML
4. Data Control Language - DCL



Définir ou modifier la structure de la base de données

# Le SQL

## Interrogation des données

1. Data Query Language – DQL
2. Data Definition Language – DDL
3. Data Manipulation Language –  
DML permet d'insérer, de mettre à  
jour, de supprimer des données
4. Data Control Language - DCL



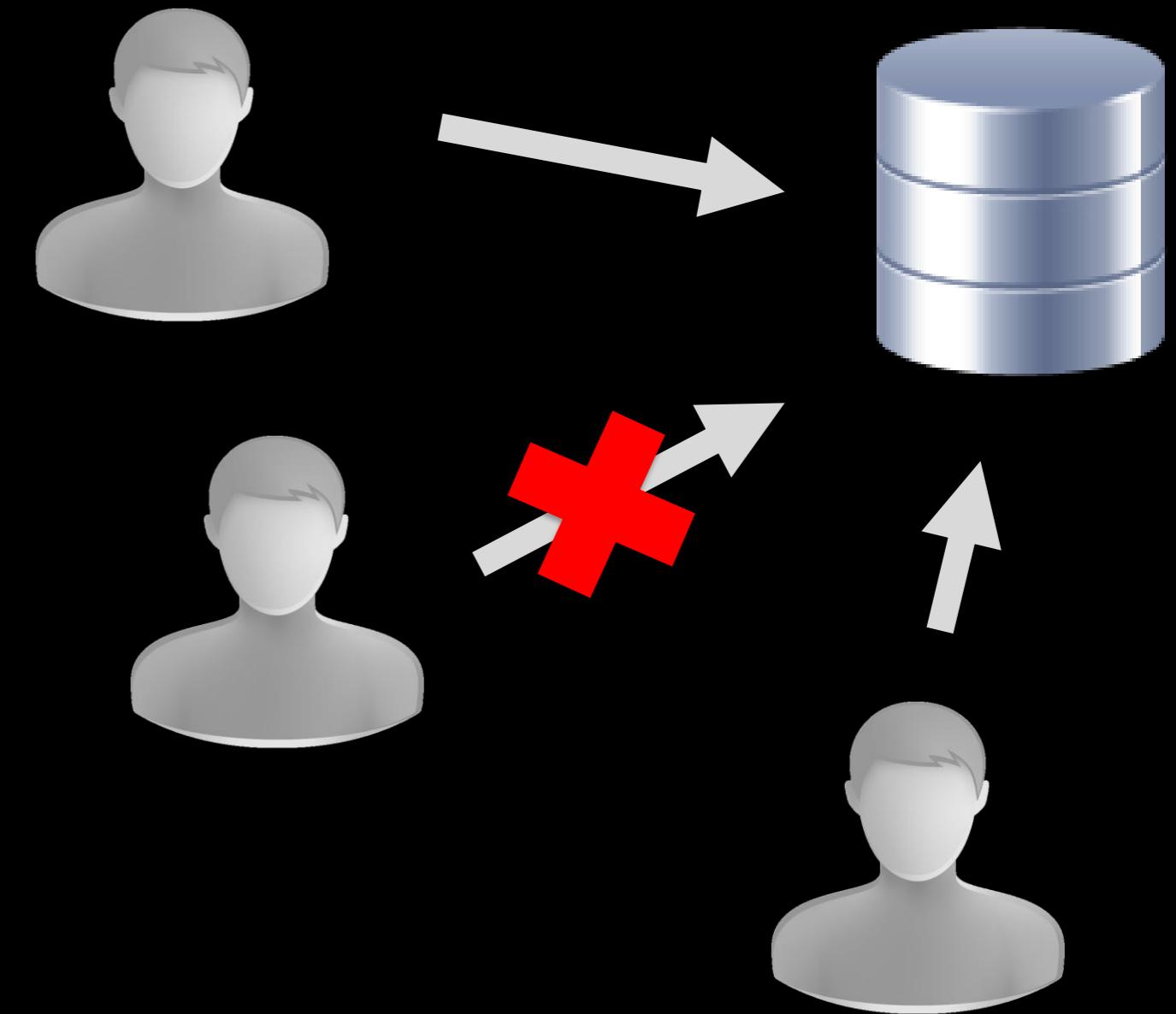
	ID	NAME	ROOMS	MANAGER	ADDRESS
0	1	Beach Bungalows	4	Mr. A. Confait	Grand' Anse
1	2	Grand' Anse Bea...	9	Mrs. M . F . Wartelsteiner	Amitie
2	3	Maison De Palme...	24	Mr. Basil Ferrari	Amitie
3	4	Le Lagon Baron ...	10	Mr. Karl D' Unieville	Grand' Anse
4	5	Les Cabanes Des...	5	Mr. Jose Confait	Grand' Anse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

Insérer, supprimer ou  
modifier des données

# Le SQL

## Interrogation des données

1. Data Query Language – DQL
2. Data Definition Language – DDL
3. Data Manipulation Language – DML
4. Data Control Language – DCL  
permet de gérer les droits et les accès des utilisateurs



Gestion des droits et des accès

# Interrogation des données - Data Query Language (DQL)

## Syntaxe de base (<https://sql.sh/cours>)

```
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
```

→ Clauses de sélection

→ Clauses de filtre conditionnel

→ Clauses d'agrégation

→ Clauses de tri

→ Clauses de fusion

\* Restriction sur le nombre de lignes rentrées

# Illustration

---

- L'utilisation de ces différents mots clés dans les requêtes est illustrée dans les séquences vidéos du MOOC (voir sur Moodle, matériel complémentaire)
- Bases de données Sqlite et PostgreSQL/PostGIS avec exemples sur les Seychelles

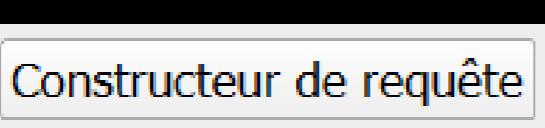
# Le SQL – Bases de données - exemples

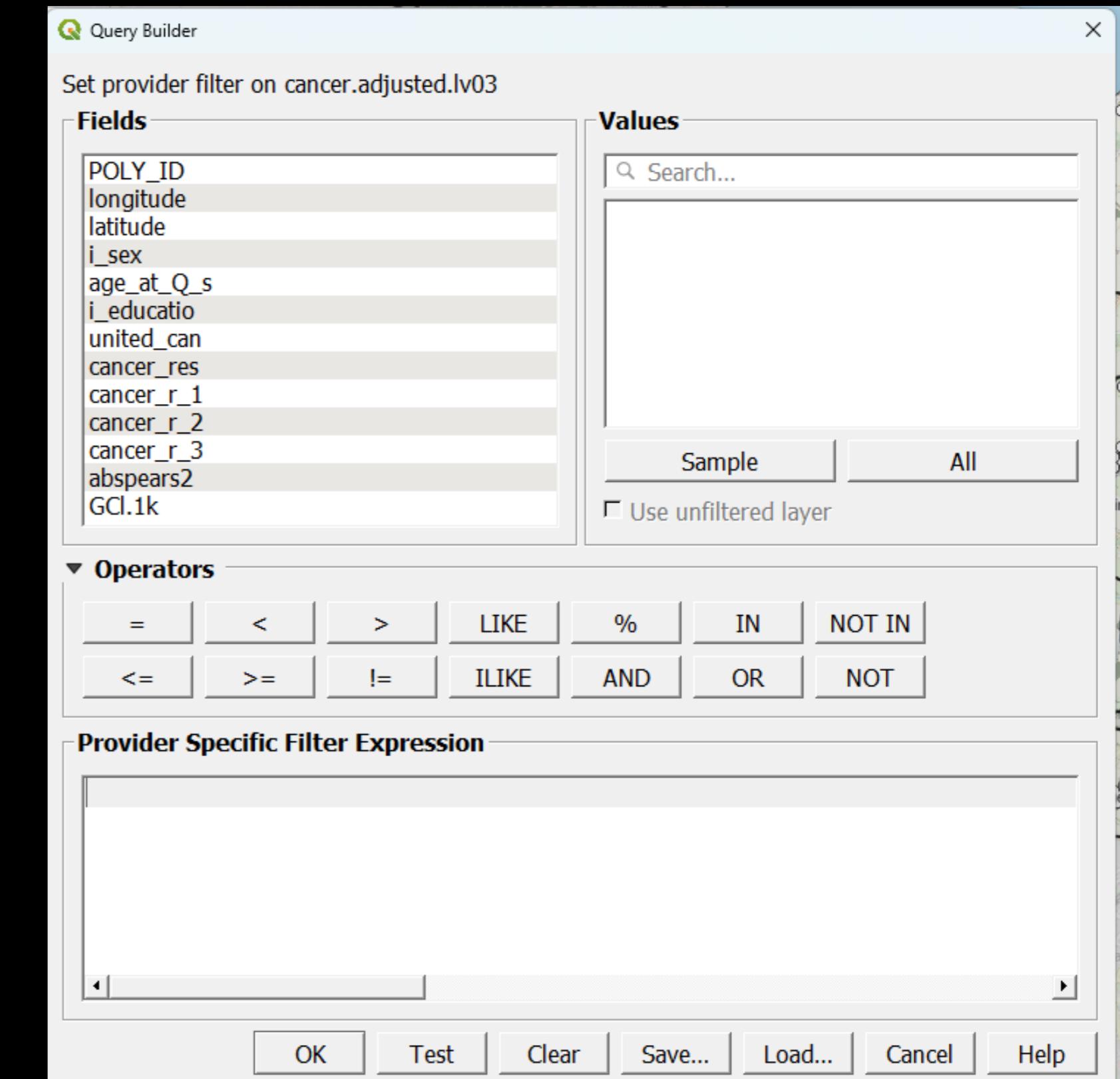
Seychelles, les îles proches

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



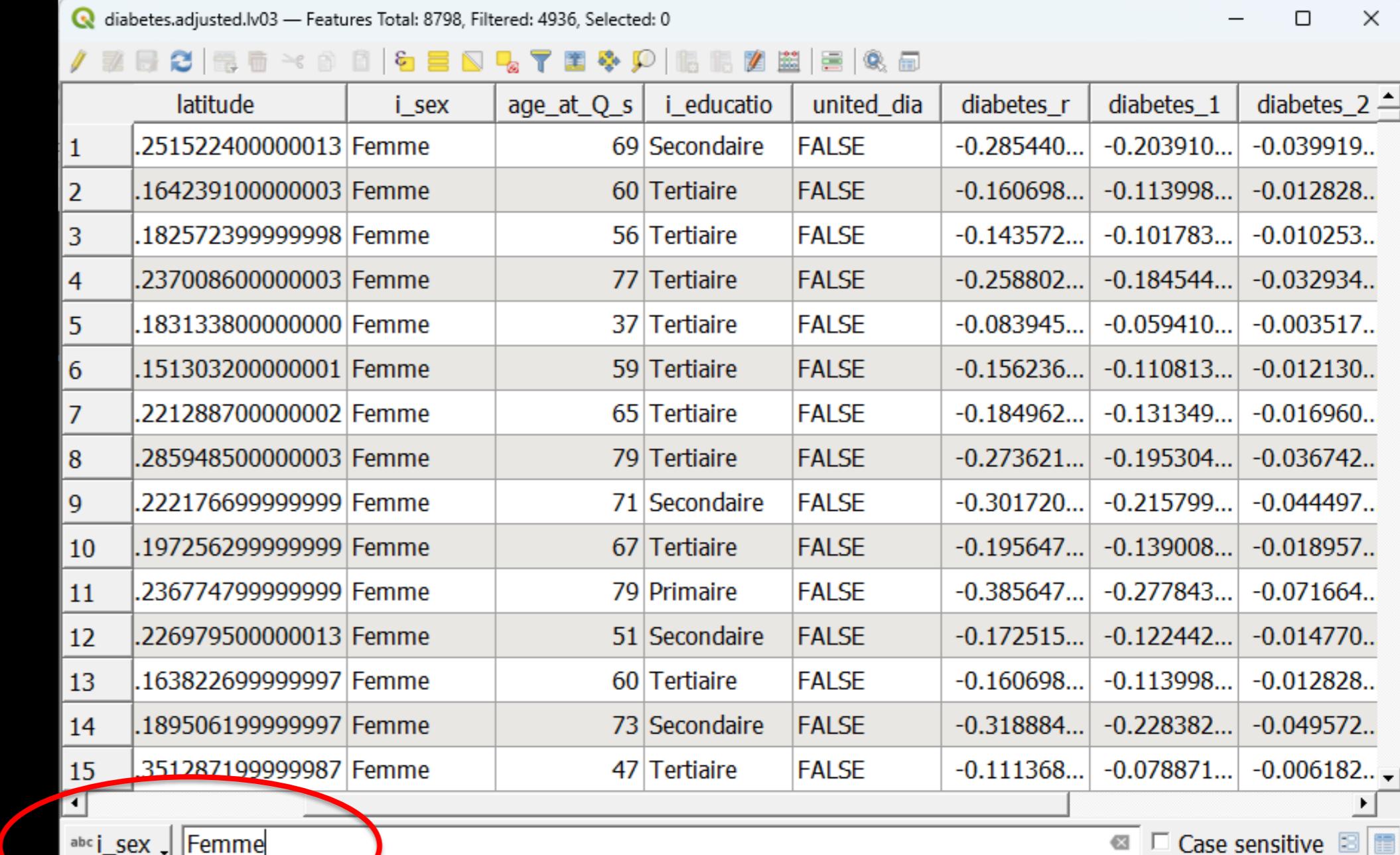
# Outils de requête dans QGIS 1 – sans SQL

- Il existe de nombreux outils de requête
- Nous allons nous concentrer sur ceux que l'on peut utiliser avec QGIS
- Il y a trois outils de requête dans QGIS qui s'appliquent à tous les types de couches
- Mais ces solutions n'implémentent pas de manière explicite le langage SQL
- Constructeur de requêtes (Query builder) (propriétés de couche > Source > Constructeur de requêtes) 
- Requêtes et filtres sur la table d'attributs
- Extension Requête spatiale («Sélection par localisation» dans QGIS3, Processing Toolbox)



# Outils de requête dans QGIS 2 – sans SQL

- Il existe de nombreux outils de requête
- Nous allons nous concentrer sur ceux que l'on peut utiliser avec QGIS
- Il y a trois outils de requête dans QGIS qui s'appliquent à tous les types de couches
- Mais ces solutions n'implémentent pas de manière explicite le langage SQL
- Query builder (propriétés de couche, Source)
- **Requêtes et filtres sur la table d'attributs**
- Extension Requête spatiale («Sélection par localisation» dans QGIS3, Processing Toolbox)

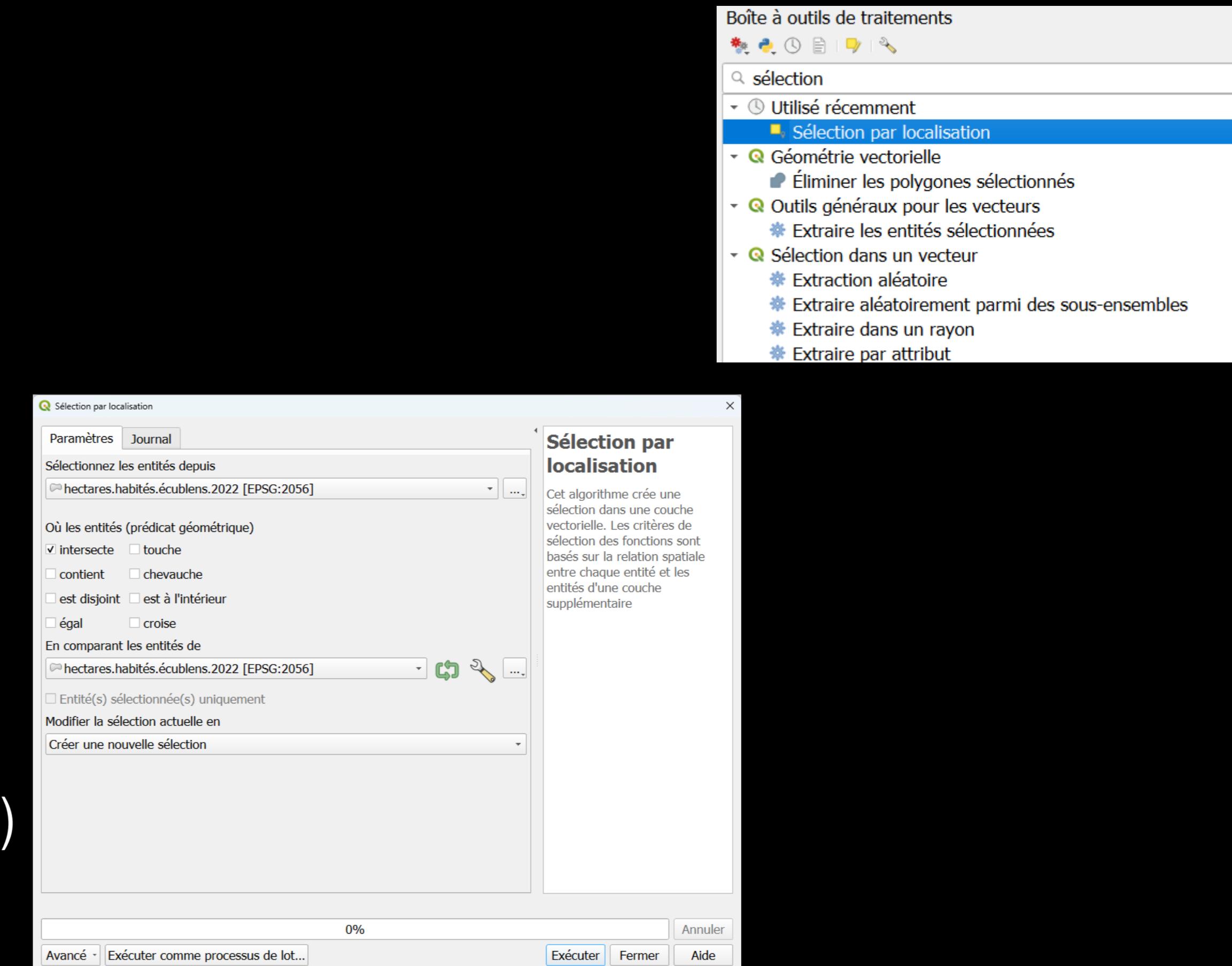


The screenshot shows the QGIS attribute table for the 'diabetes.adjusted.lv03' layer. The table has 15 rows and 9 columns. The columns are: latitude, i\_sex, age\_at\_Q\_s, i\_educatio, united\_dia, diabetes\_r, diabetes\_1, diabetes\_2, and diabetes\_3. The 'i\_sex' column contains values like 'Femme' and 'Homme'. The 'age\_at\_Q\_s' column contains values like 69, 60, 56, 77, 37, 59, 65, 79, 71, 67, 79, 51, 60, 73, and 47. The 'diabetes\_r' column contains numerical values. A red circle highlights the search bar at the bottom of the table, which contains the text 'Femme'. The status bar at the bottom right shows 'Case sensitive'.

	latitude	i_sex	age_at_Q_s	i_educatio	united_dia	diabetes_r	diabetes_1	diabetes_2
1	.251522400000013	Femme	69	Secondaire	FALSE	-0.285440...	-0.203910...	-0.039919...
2	.164239100000003	Femme	60	Tertiaire	FALSE	-0.160698...	-0.113998...	-0.012828...
3	.18257239999998	Femme	56	Tertiaire	FALSE	-0.143572...	-0.101783...	-0.010253...
4	.237008600000003	Femme	77	Tertiaire	FALSE	-0.258802...	-0.184544...	-0.032934...
5	.183133800000000	Femme	37	Tertiaire	FALSE	-0.083945...	-0.059410...	-0.003517...
6	.151303200000001	Femme	59	Tertiaire	FALSE	-0.156236...	-0.110813...	-0.012130...
7	.221288700000002	Femme	65	Tertiaire	FALSE	-0.184962...	-0.131349...	-0.016960...
8	.285948500000003	Femme	79	Tertiaire	FALSE	-0.273621...	-0.195304...	-0.036742...
9	.222176699999999	Femme	71	Secondaire	FALSE	-0.301720...	-0.215799...	-0.044497...
10	.197256299999999	Femme	67	Tertiaire	FALSE	-0.195647...	-0.139008...	-0.018957...
11	.236774799999999	Femme	79	Primaire	FALSE	-0.385647...	-0.277843...	-0.071664...
12	.226979500000013	Femme	51	Secondaire	FALSE	-0.172515...	-0.122442...	-0.014770...
13	.163822699999997	Femme	60	Tertiaire	FALSE	-0.160698...	-0.113998...	-0.012828...
14	.189506199999997	Femme	73	Secondaire	FALSE	-0.318884...	-0.228382...	-0.049572...
15	35128719999987	Femme	47	Tertiaire	FALSE	-0.111368...	-0.078871...	-0.006182...

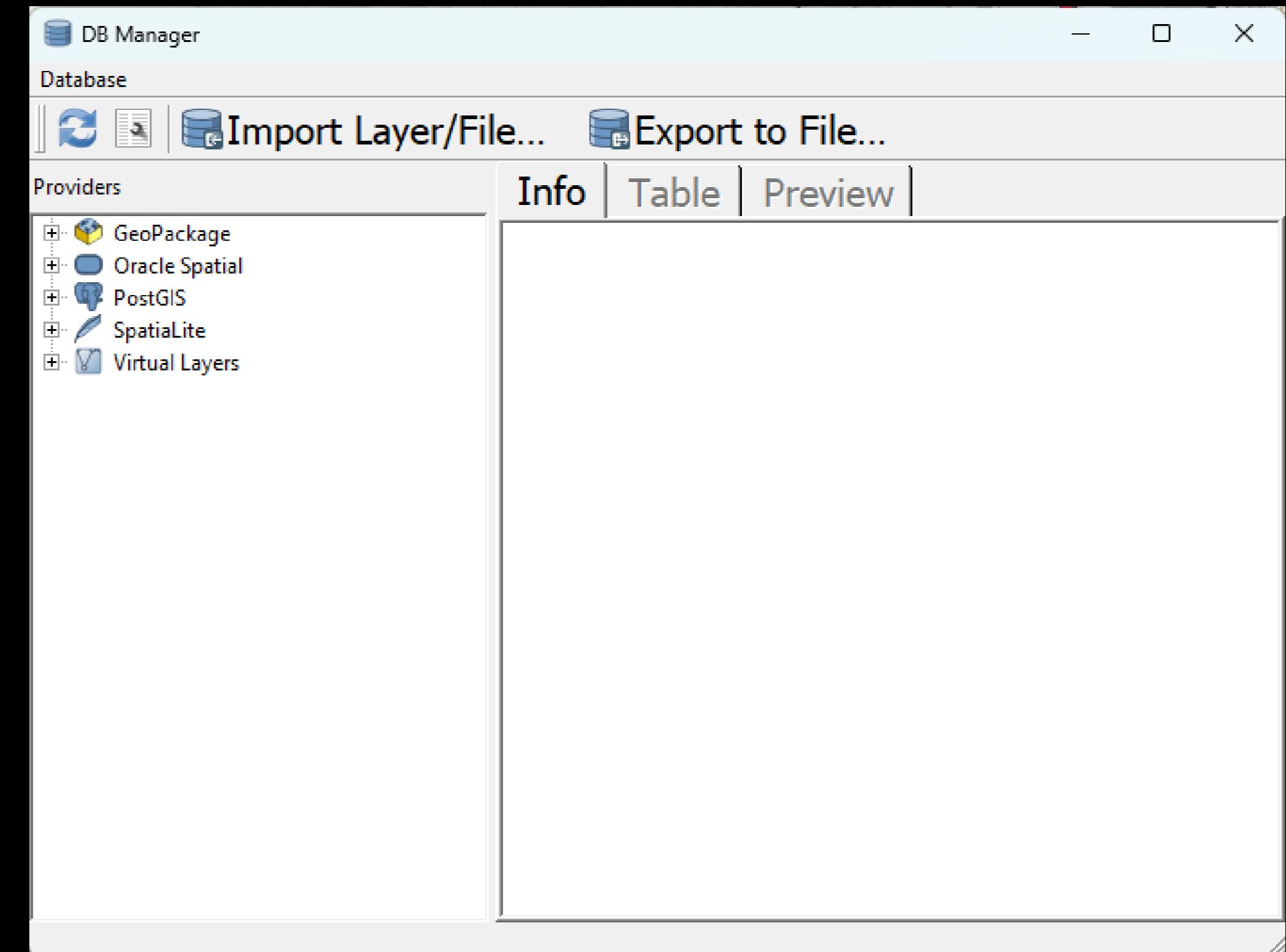
# Outils de requête dans QGIS 3 – sans SQL

- Il existe de nombreux outils de requête
- Nous allons nous concentrer sur ceux que l'on peut utiliser avec QGIS
- Il y a trois outils de requête dans QGIS qui s'appliquent à tous les types de couches
- Mais ces solutions n'implémentent pas de manière explicite le langage SQL
- Query builder (propriétés de couche, Source)
- Requêtes et filtres sur la table d'attributs
- Extension Requête spatiale («**Sélection par localisation**» dans QGIS3, Processing Toolbox)



# Outils de requête – SQL compatibles

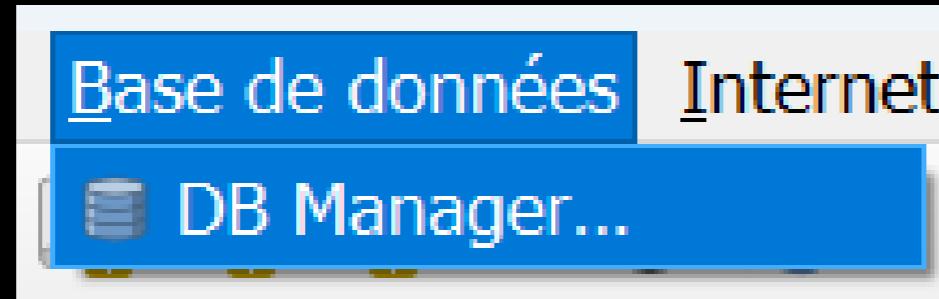
- Gestion de base de données SQL  
«DB Manager»
- Permet de traiter des bases de données  
Spatialite, PostGIS, Oracle Spatial et  
GeoPackage



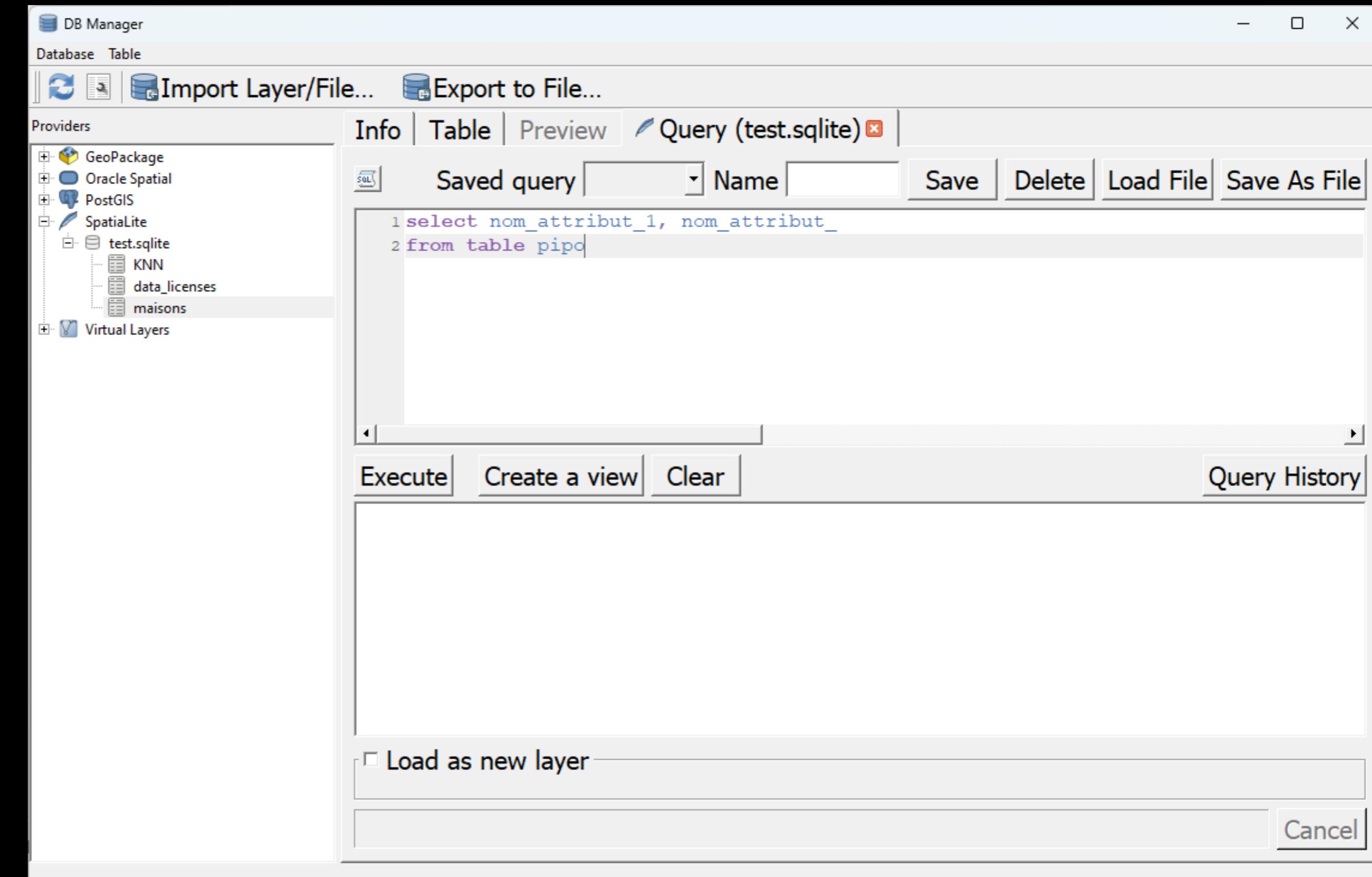
# Outils de requête – SQL compatibles

- Gestion de base de données SQL

«DB Manager»



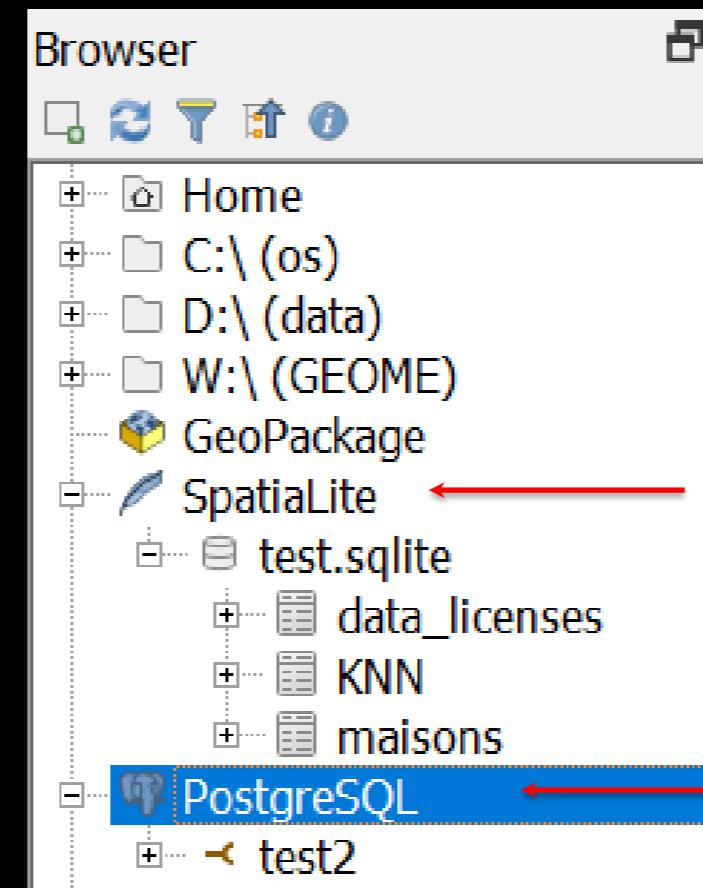
- Permet de traiter des bases de données Spatialite, PostGIS, Oracle Spatial et GeoPackage
- SQL Window



# Le SQL – Outils de requête spécifiques Sqlite et pgAdmin

## Spécifique Spatialite

- **Sqlite Studio** (hors QGIS)
- **Ancienne extension Qspatialite**



The screenshot shows the pgAdmin 4 interface with the 'Browser' tab selected. The tree view on the left lists various PostgreSQL objects: Casts, Catalogs, Event Triggers, Extensions, Foreign Data Wrappers, Languages, and Schemas (2). Under Schemas, the 'public' schema is expanded to show Collations, Domains, FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Sequences, and Tables (6). The 'country\_outlines' table is selected and highlighted in blue. The 'Query Editor' tab is active, showing the following SQL query:

```
1 SELECT * FROM public.country_outlines
2 ORDER BY ogc_fid ASC
```

The 'Data Output' tab shows the results of the query, listing countries with their geometry IDs. The 'Geometry Viewer' tab on the right displays a map of Europe with the country outlines highlighted.

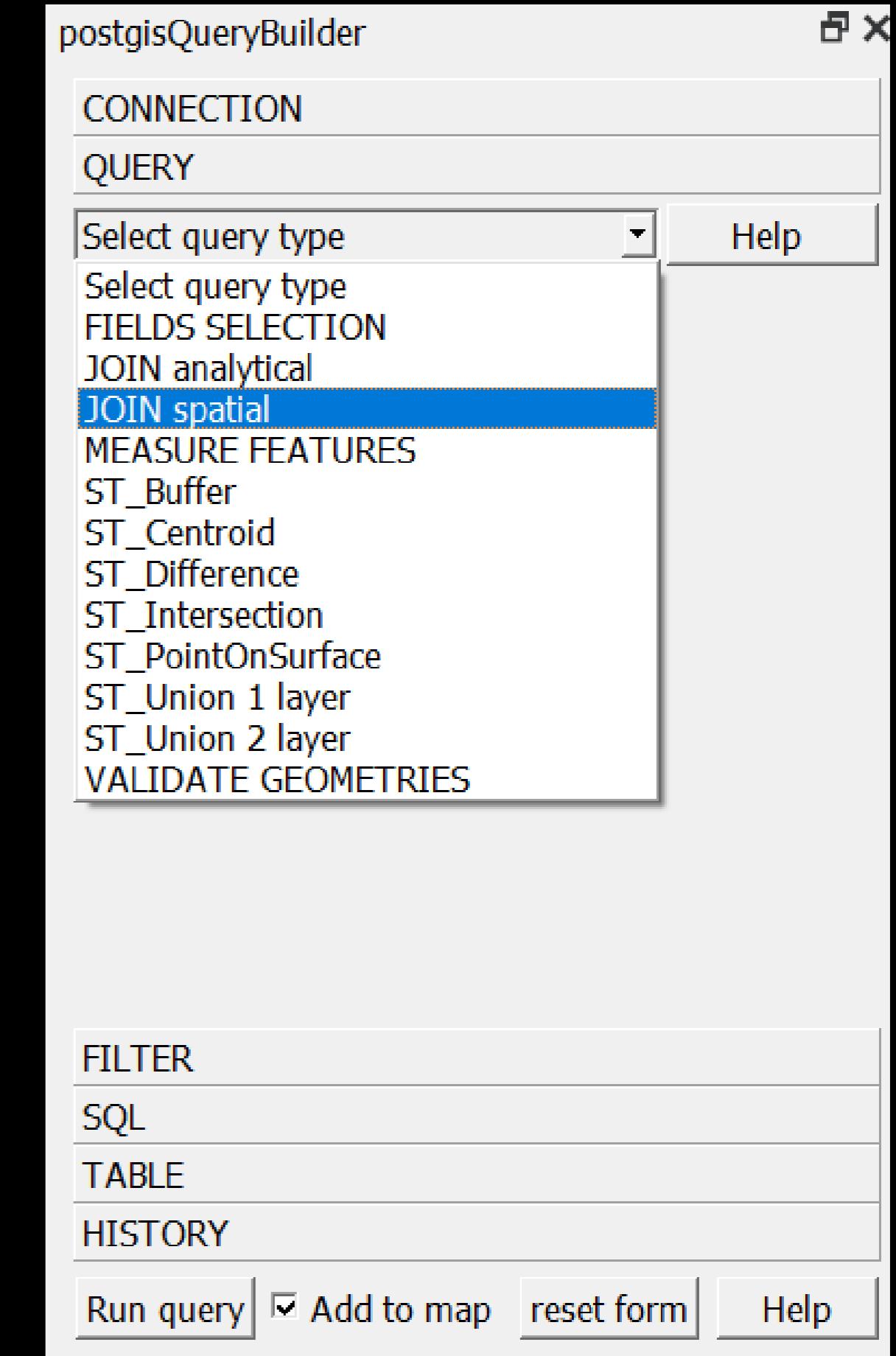
name	geometry4326	geometry3857
Afghanistan	0103000020E610000001...	0103000020110F000001...
Angola	0106000020E610000002...	0106000020110F000002...
Albania	0103000020E610000001...	0103000020110F000001...
United Arab Emirates	0103000020E610000001...	0103000020110F000001...
Argentina	0106000020E610000002...	0106000020110F000002...
Armenia	0103000020E610000001...	0103000020110F000001...
Antarctica	0106000020E610000008...	0106000020110F000008...
French Southern and A...	0103000020E610000001...	0103000020110F000001...
Australia	0106000020E610000002...	0106000020110F000002...
Austria	0103000020E610000001...	0103000020110F000001...
Azerbaijan	0106000020E610000003...	0106000020110F000003...

## Spécifique PostgreSQL/PostGIS

- **pgAdmin 4** (hors QGIS): logiciel qui fait partie de la suite PostgreSQL et qui présente l'avantage de proposer un outil de construction de requête graphique

# Le SQL – Outils de requête - PostGIS Query Builder

- **Extension postGISQueryBuilder**
- Ce plugin QGIS vise à fournir un environnement convivial pour faire de l'analyse spatiale dans postgresql/postgis sans avoir à connaître le code SQL
- Il n'est pas destiné à remplacer le DBManager intégré à QGIS qui couvre tous les aspects de l'interaction QGIS/PostGIS



 postgisQueryBuilder

# Sélection simple

## Sélectionner un attribut

```
SELECT nom_attribut  
FROM nom_table
```

La requête de sélection comprend le mots clé "**select**" suivi du nom de l'attribut recherché puis du mot clé "**from**" qui permet de préciser dans quelle table se trouve l'attribut en question.

The image consists of two side-by-side screenshots of the QspatialLite application interface. Both screenshots show a toolbar at the top with various icons for database management, followed by a 'Tables' panel on the left and a main workspace on the right.

**Left Screenshot (Query Preparation):** The workspace shows the SQL query `SELECT NOM FROM hotels` entered in the SQL editor. The 'Run SQL' button is visible at the bottom of the editor.

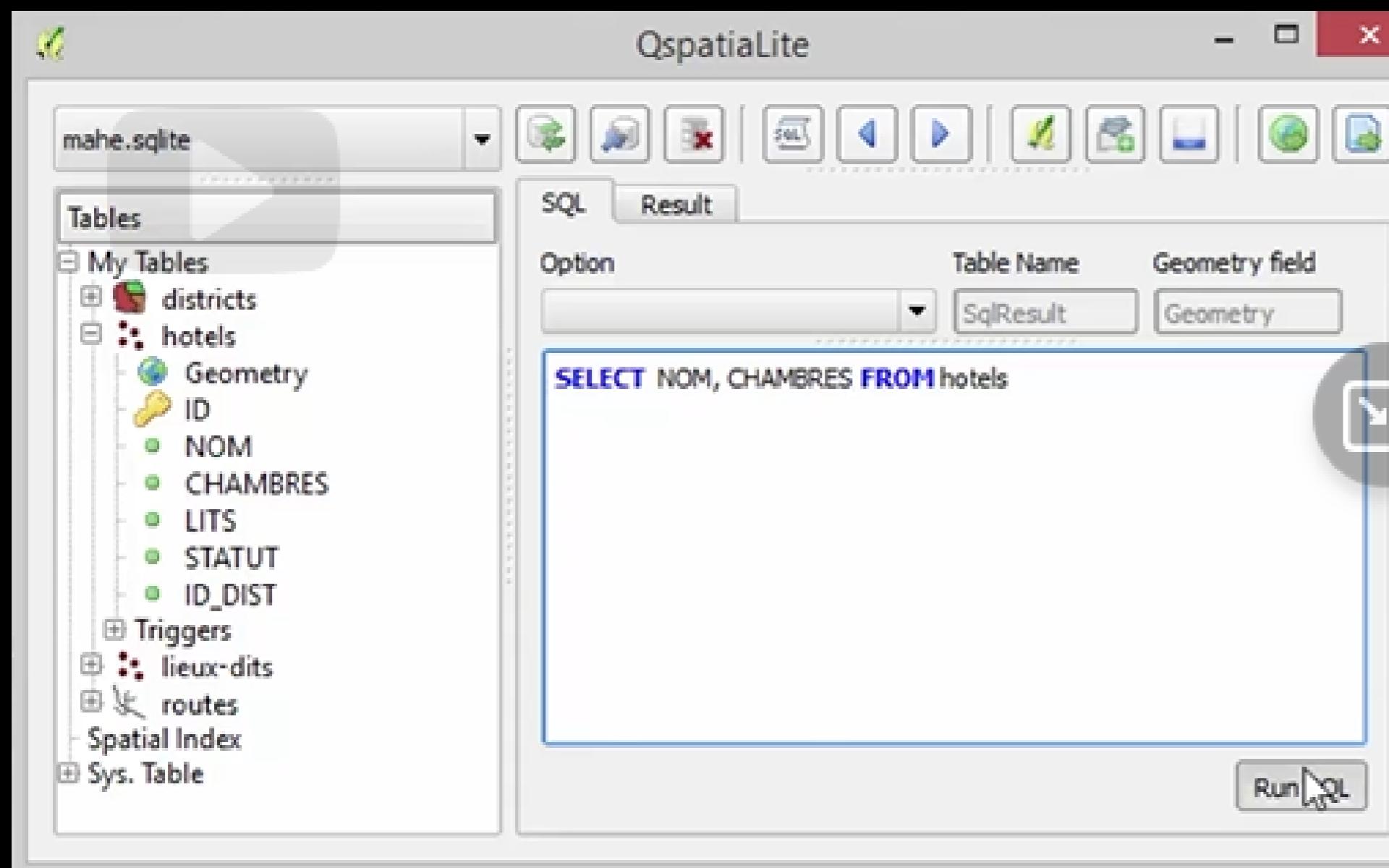
**Right Screenshot (Query Result):** The workspace displays the results of the query as a table. The table has a single column labeled 'NOM' and contains six rows of data: Beach Bungalows, Grand' Anse Be..., Maison De Pal..., Le Lagon Baron..., Les Cabanes De..., and Britannia. The table includes standard database navigation buttons (first, previous, next, last) on the right and bottom.

# Sélection simple

## Sélectionner deux attributs

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

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.



QspatialLite

mahe.sqlite

Tables

My Tables

- districts
- hotels
  - Geometry
  - ID
  - NOM
  - CHAMBRES
  - LITS
  - STATUT
  - ID\_DIST
- lieux-dits
- routes

Spatial Index

Sys. Table

SQL

Result

Option

Table Name

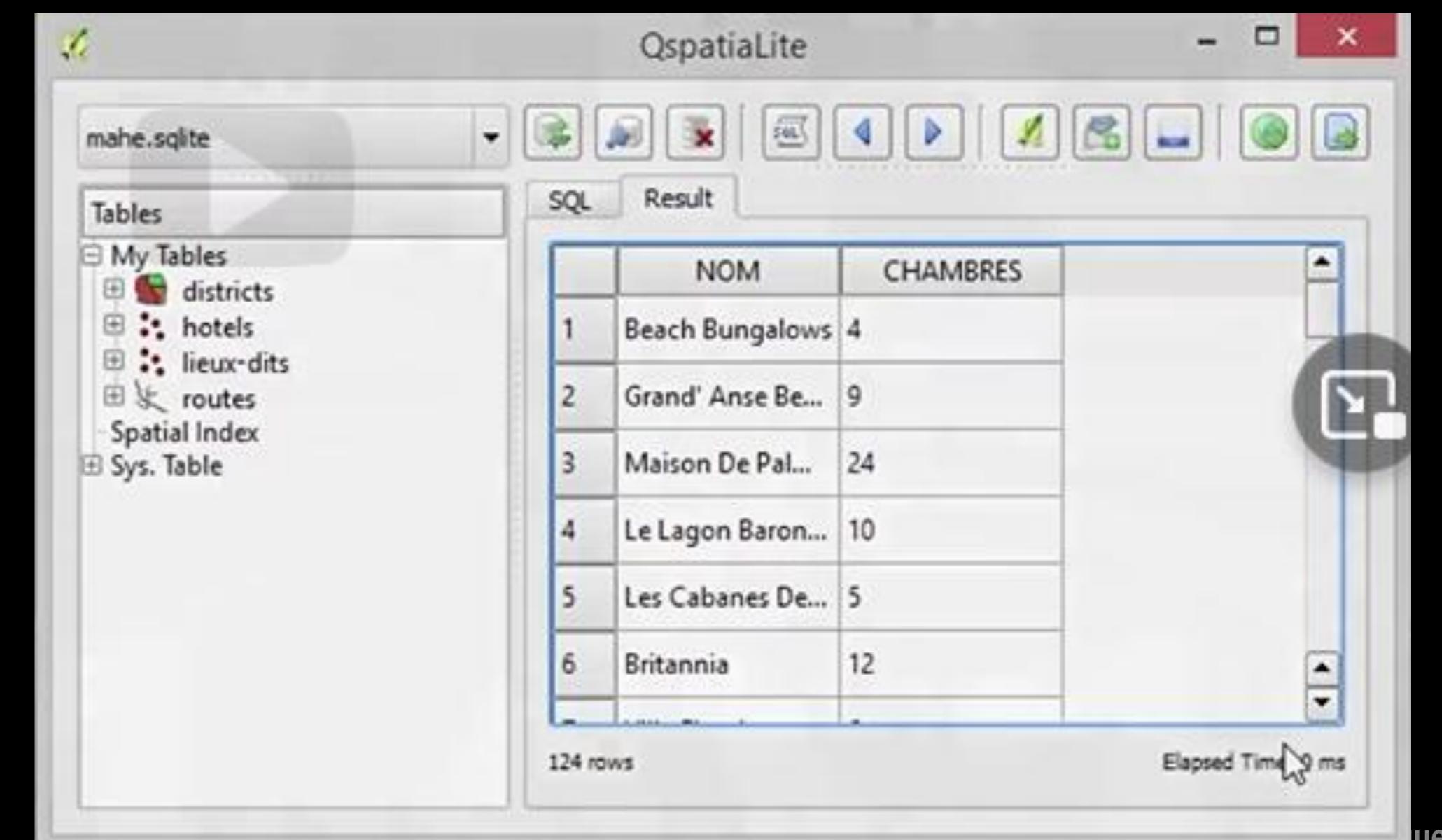
Geometry field

SqlResult

Geometry

SELECT NOM, CHAMBRES FROM hotels

Run SQL



QspatialLite

mahe.sqlite

Tables

My Tables

- districts
- hotels
  - Geometry
  - ID
  - NOM
  - CHAMBRES
  - LITS
  - STATUT
  - ID\_DIST
- lieux-dits
- routes

Spatial Index

Sys. Table

SQL

Result

	NOM	CHAMBRES
1	Beach Bungalows	4
2	Grand' Anse Be...	9
3	Maison De Pal...	24
4	Le Lagon Baron...	10
5	Les Cabanes De...	5
6	Britannia	12

124 rows

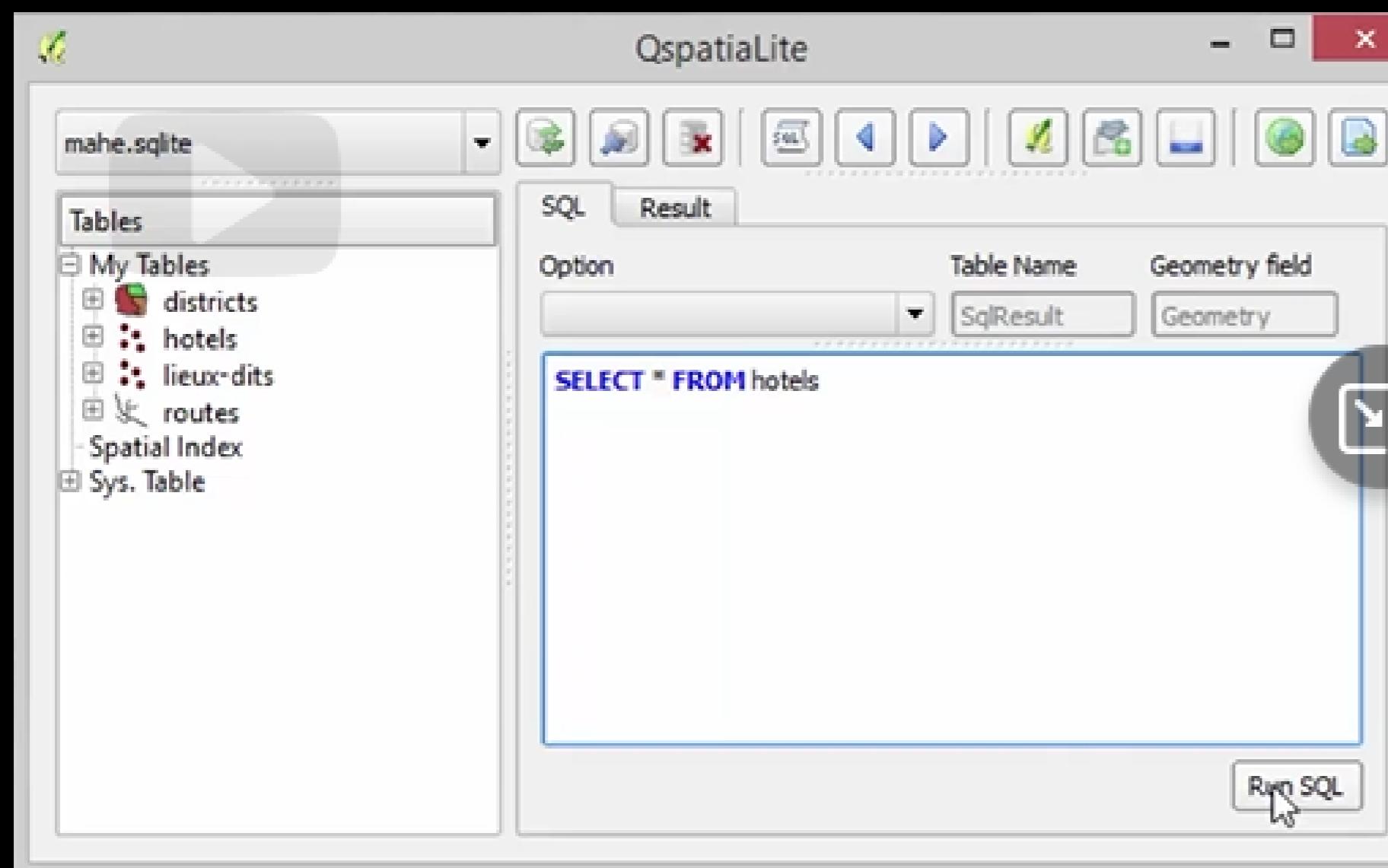
Elapsed Time 9 ms

# 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.



The screenshot shows the QspatialLite application window with the 'Result' tab selected. The results of the query 'SELECT \* FROM hotels' are displayed in a table. The table has columns 'NOM', 'CHAMBRES', and 'LITS'. The data rows are:

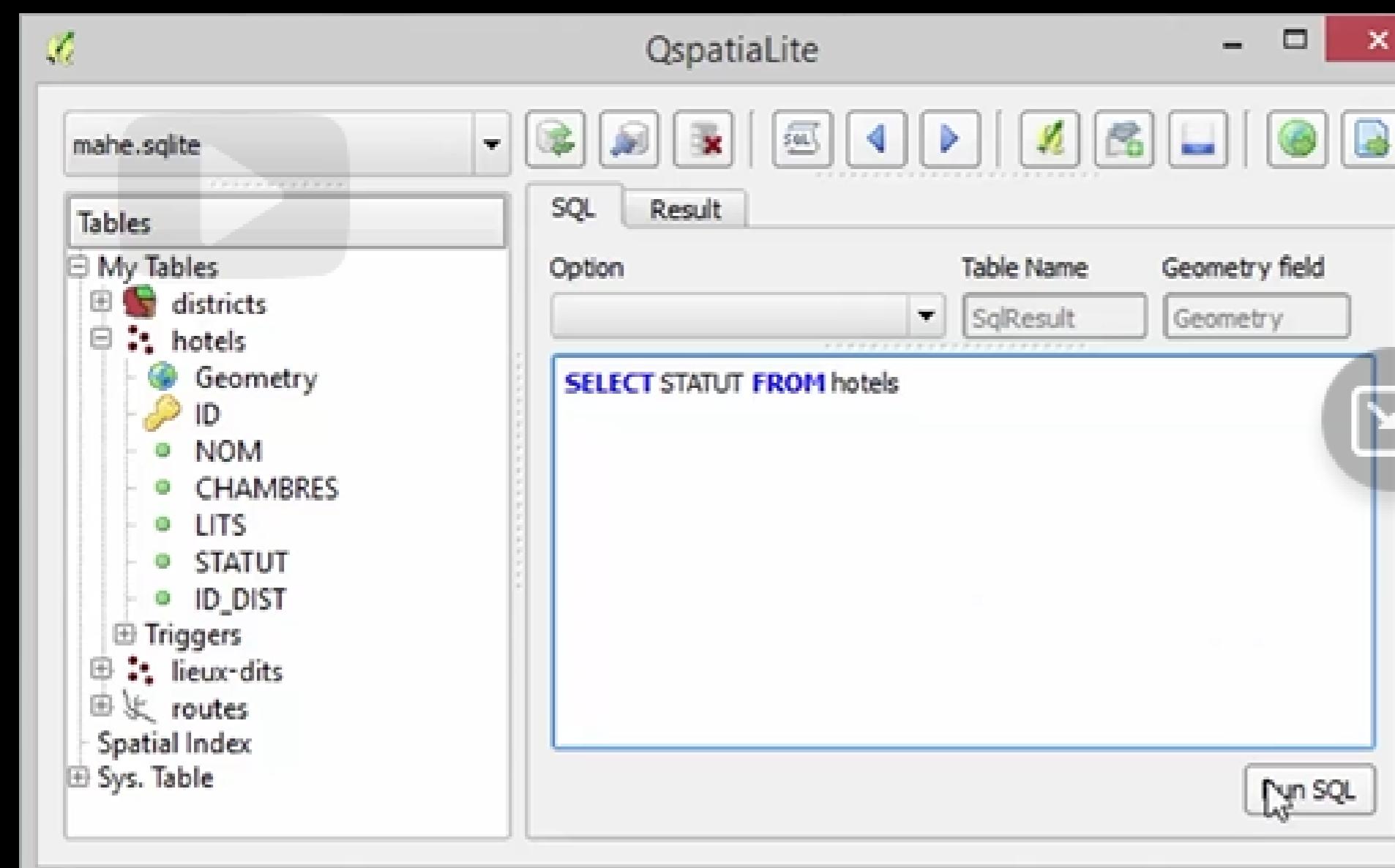
	NOM	CHAMBRES	LITS
1	Beach Bungalows	4	8
2	Grand' Anse Be...	9	18
3	Maison De Pal...	24	48
4	Le Lagon Baron...	10	20
5	Les Cabanes De...	5	10
6	Britannia	12	24

At the bottom of the table, it says '124 rows' and 'Elapsed Time: 0 ms'.

# 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.



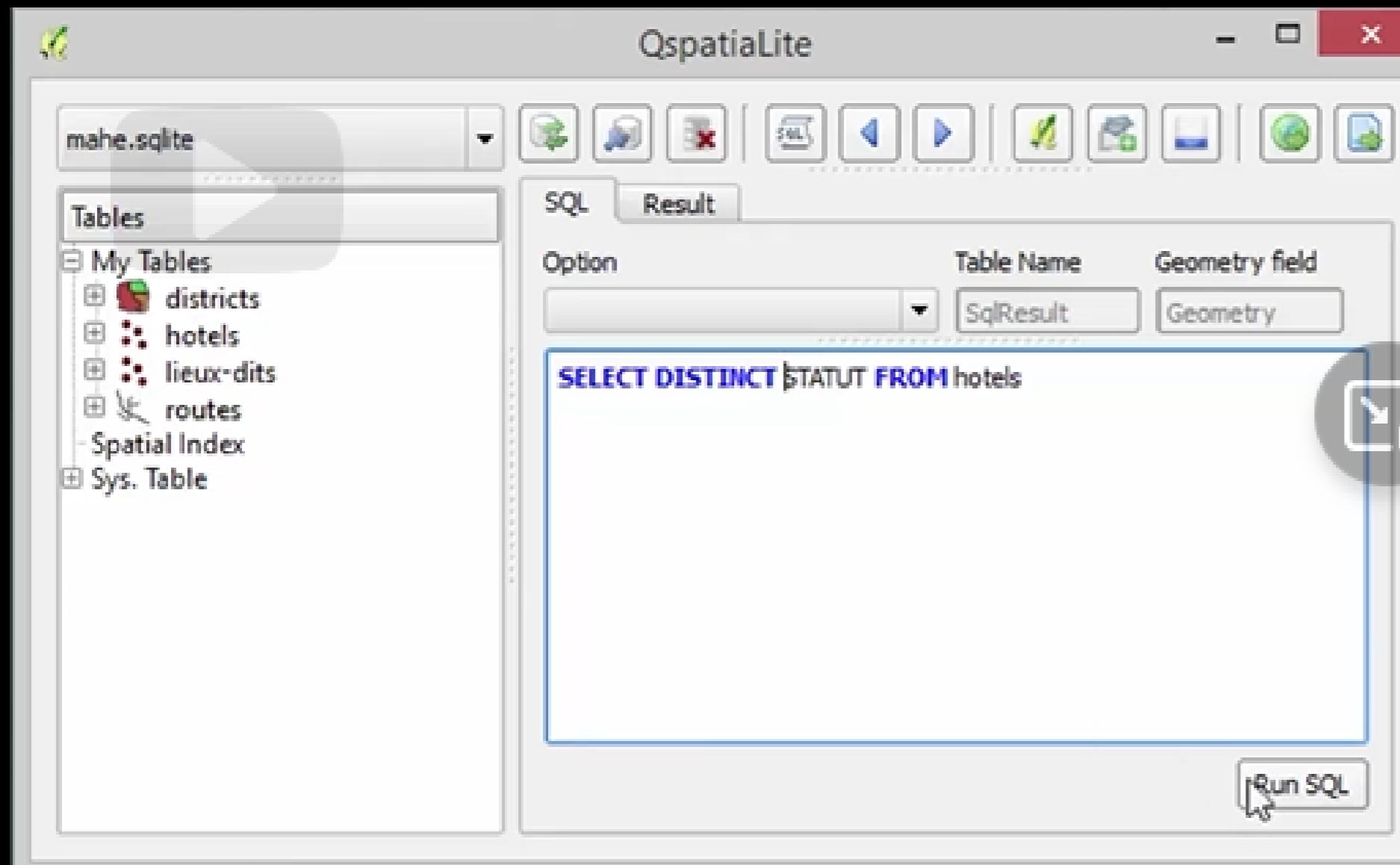
The screenshot shows the QspatialLite application window with the 'Result' tab active. The results of the query 'SELECT STATUT FROM hotels' are displayed in a table. The table has one column labeled 'STATUT' with six rows, all containing the value 'Small Hotel'. The table is surrounded by a light blue border. The status bar at the bottom right shows '124 rows' and 'Elapsed Time: 0 ms'. The 'Tables' pane on the left is identical to the one in the first screenshot.

STATUT
Small Hotel

# 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.



The screenshot shows the Qspatialite application window. On the left, the 'Tables' panel lists 'districts', 'hotels', 'lieux-dits', and 'routes' under 'My Tables'. The 'Result' tab is active, displaying the query results in a table format:

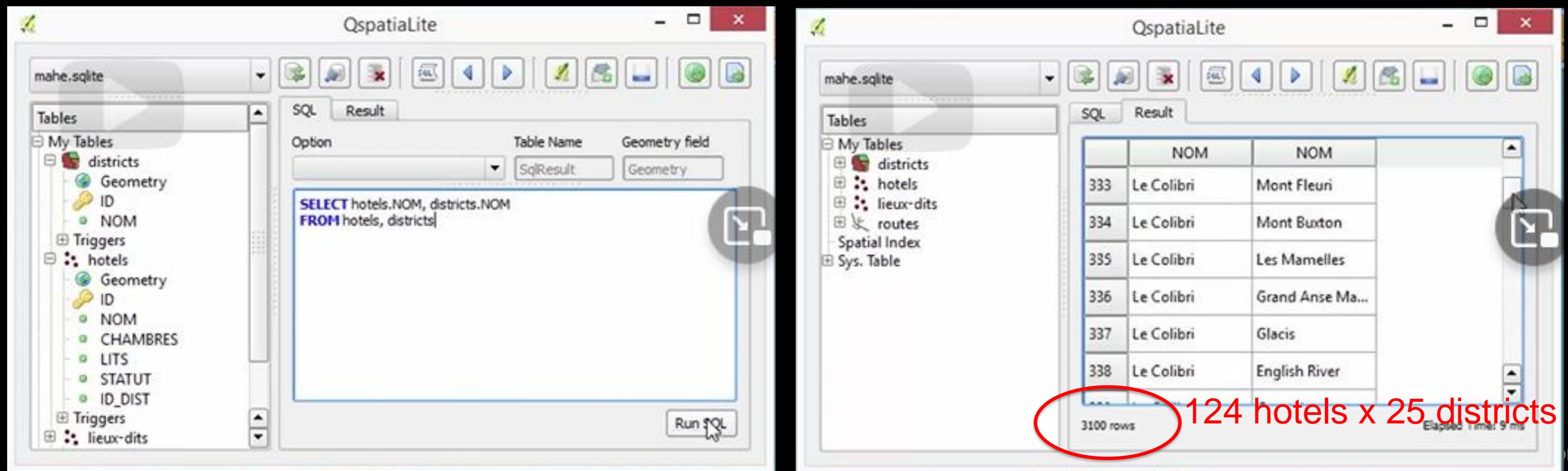
STATUT
1 Small Hotel
2 Self Catering
3 Large Hotel
4 Small Hotel,Self...
5 Small Hotel,Wa...
6 Large Hotel,Wa...

Below the table, it says '7 rows' and 'Elapsed Time: 0 ms'.

# Sélection depuis deux tables

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

L'un des intérêts de 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 décrit 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**.



The image shows two screenshots of the Qspatialite application interface. The left screenshot shows the SQL tab with the following query:

```
SELECT hotels.NOM, districts.NOM  
FROM hotels, districts
```

The right screenshot shows the Result tab displaying the query results in a table:

	NOM	NOM
333	Le Colibri	Mont Fleuri
334	Le Colibri	Mont Buxton
335	Le Colibri	Les Mamelles
336	Le Colibri	Grand Anse Magique
337	Le Colibri	Glacis
338	Le Colibri	English River

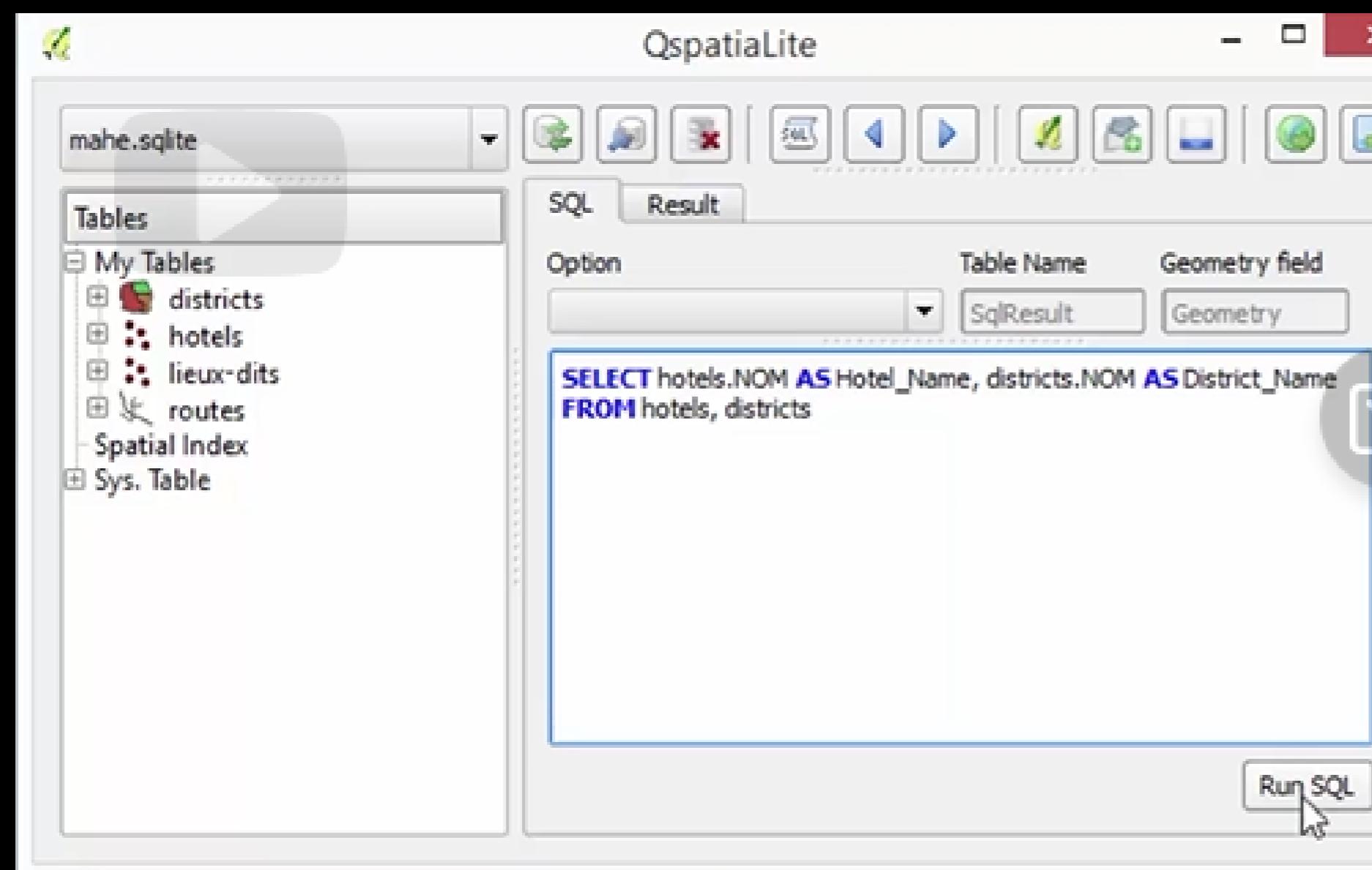
At the bottom of the result table, it says "3100 rows". A red circle highlights this text, and the text "124 hotels x 25 districts" is overlaid in red at the bottom right of the screenshot.

# Alias

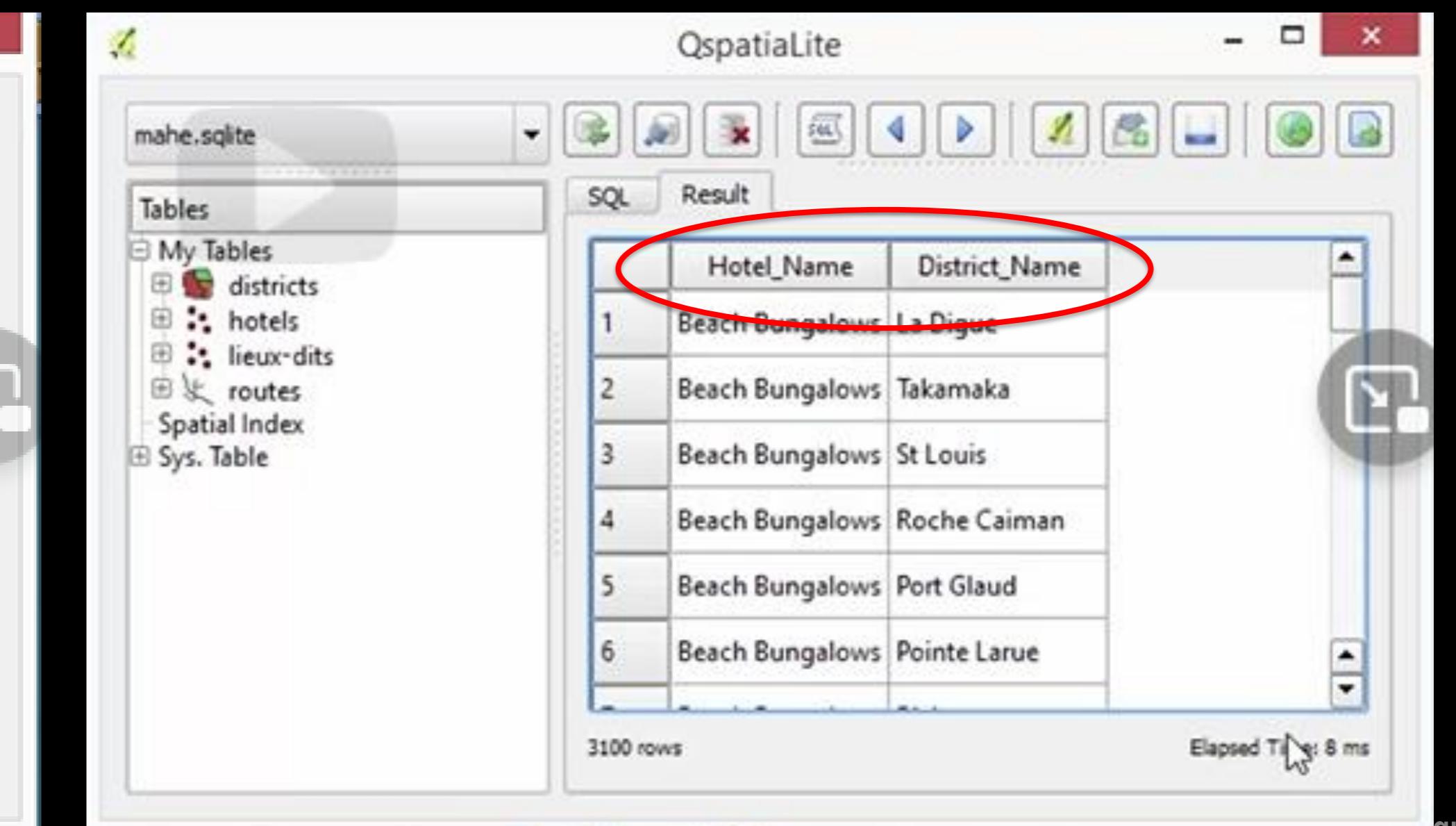
## Alias du nom de colonne

```
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. Cette opération s'effectue à l'aide du mot clé "as".



The screenshot shows the QspatialLite interface. On the left, the 'Tables' panel lists 'My Tables' with 'districts', 'hotels', 'lieux-dits', 'routes', 'Spatial Index', and 'Sys. Table'. The 'SQL' tab is active, showing the query: `SELECT hotels.NOM AS Hotel_Name, districts.NOM AS District_Name FROM hotels, districts`. A 'Run SQL' button is at the bottom.



The screenshot shows the 'Result' tab of QspatialLite. The query has been run, and the results are displayed in a table. The columns are aliased: 'hotels.NOM' is shown as 'Hotel\_Name' and 'districts.NOM' is shown as 'District\_Name'. The table contains 6 rows of data. A red oval highlights the column headers 'Hotel\_Name' and 'District\_Name'. The bottom status bar shows '3100 rows' and 'Elapsed Time: 8 ms'.

	Hotel_Name	District_Name
1	Beach Bungalows	La Digue
2	Beach Bungalows	Takamaka
3	Beach Bungalows	St Louis
4	Beach Bungalows	Roche Caiman
5	Beach Bungalows	Port Glaud
6	Beach Bungalows	Pointe Larue

# Alias

## Alias du nom de table

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

Le concept d'alias peut aussi être utilisé pour changer le nom des tables. Utile pour simplifier la syntaxe. Dans le cas des tables, les alias ne font pas appel au mot clé "as" mais sont créés en ajoutant un nom au nom de la table avec un espace entre les 2.

The figure consists of three side-by-side screenshots of the QspatialLite application interface, showing the process of running a SQL query with table aliases.

- Left Screenshot:** Shows the SQL tab with the following query:

```
SELECT hotels|NOM, districts.NOM  
FROM hotels h, districts d
```
- Middle Screenshot:** Shows the SQL tab with the query modified to use aliases:

```
SELECT h.NOM, d.NOM  
FROM hotels h, districts d
```
- Right Screenshot:** Shows the Result tab displaying the query results in a table format. The table has two columns: NOM and NOM. The data is as follows:

	NOM	NOM
1	Beach Bungalows	La Digue
2	Beach Bungalows	Takamaka
3	Beach Bungalows	St Louis
4	Beach Bungalows	Roche Caiman
5	Beach Bungalows	Port Gaud
6	Beach Bungalows	Pointe Larue

A red circle highlights the text "3100 rows" at the bottom of the result table, and a red arrow points to the "Elapsed Time" text at the bottom right of the result tab.

## Comment ça marche ?

### Données en tableaux



### Requêtes

<Create Database/>

<Select/>

<Create Table/>

<Update/>

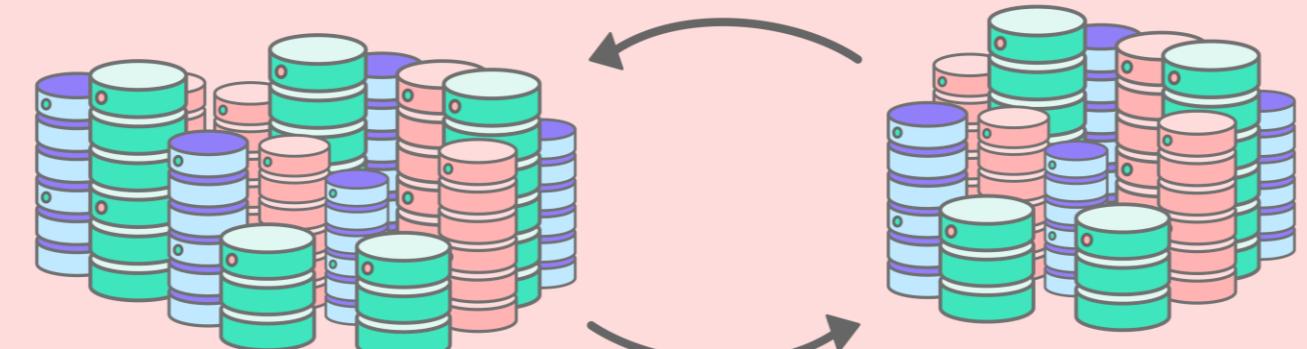
<Delete/>

### Système des "clés"

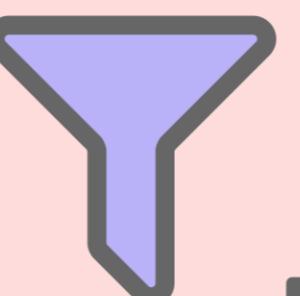


## À quoi ça sert ?

### Communiquer entre bases



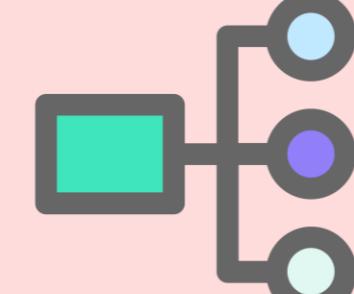
### Filtrer



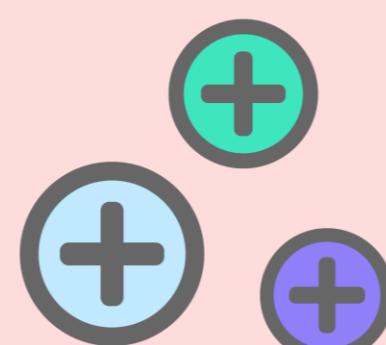
### Mettre à jour



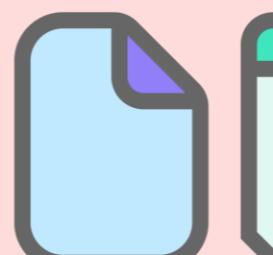
### Réorganiser



### Créer



### Modifier

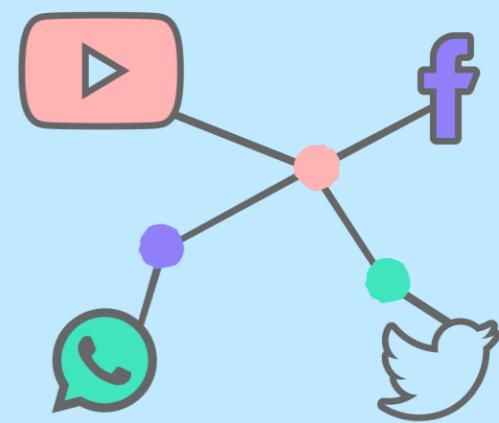


## Où c'est utilisé ?

### Streaming Musical



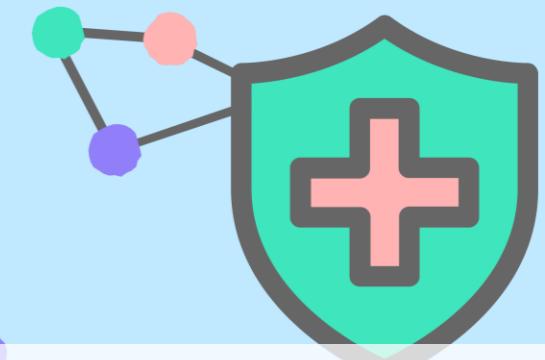
### Réseaux Sociaux



### La Finance



### Assurance



### E-Commerce



+ les bases de données  
spatialement référencées

# Frege et Russell (Logicomix)



Merci pour votre attention !

