

Concepts Avancés de Bases de données

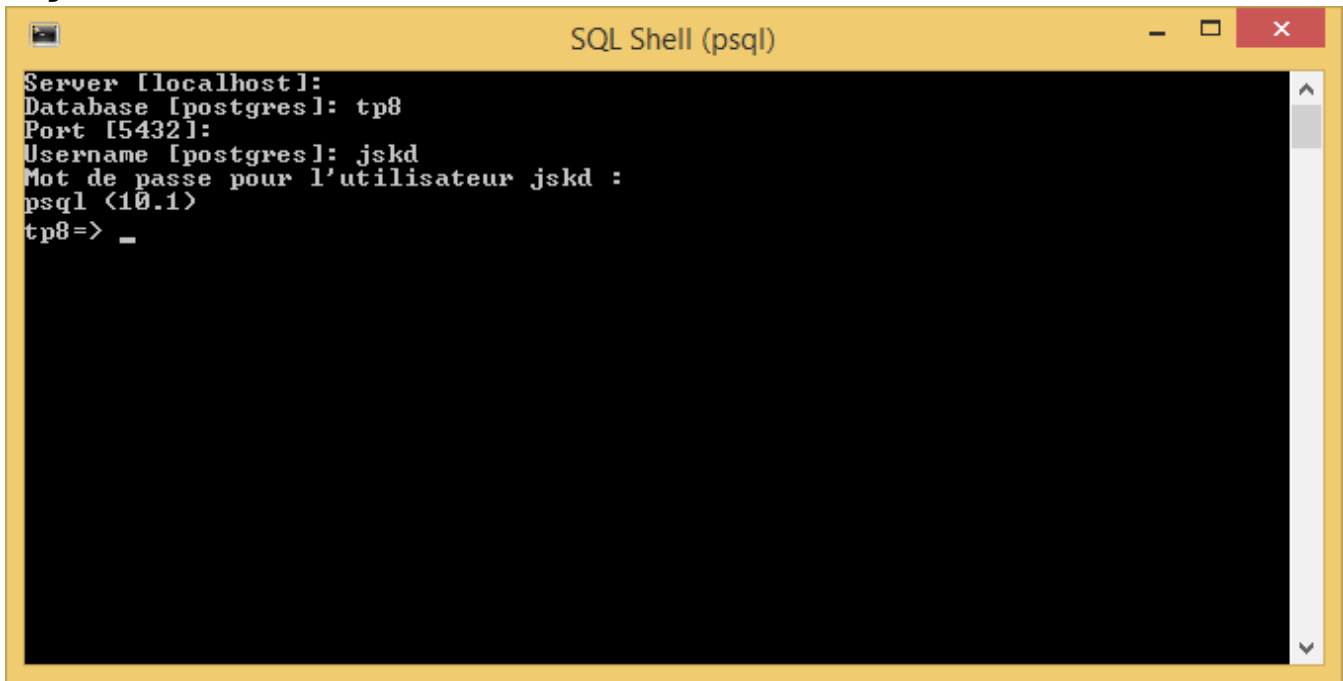
Joaquim LEFRANC et Jérôme SKODA

December 9, 2017

PG admin Explain

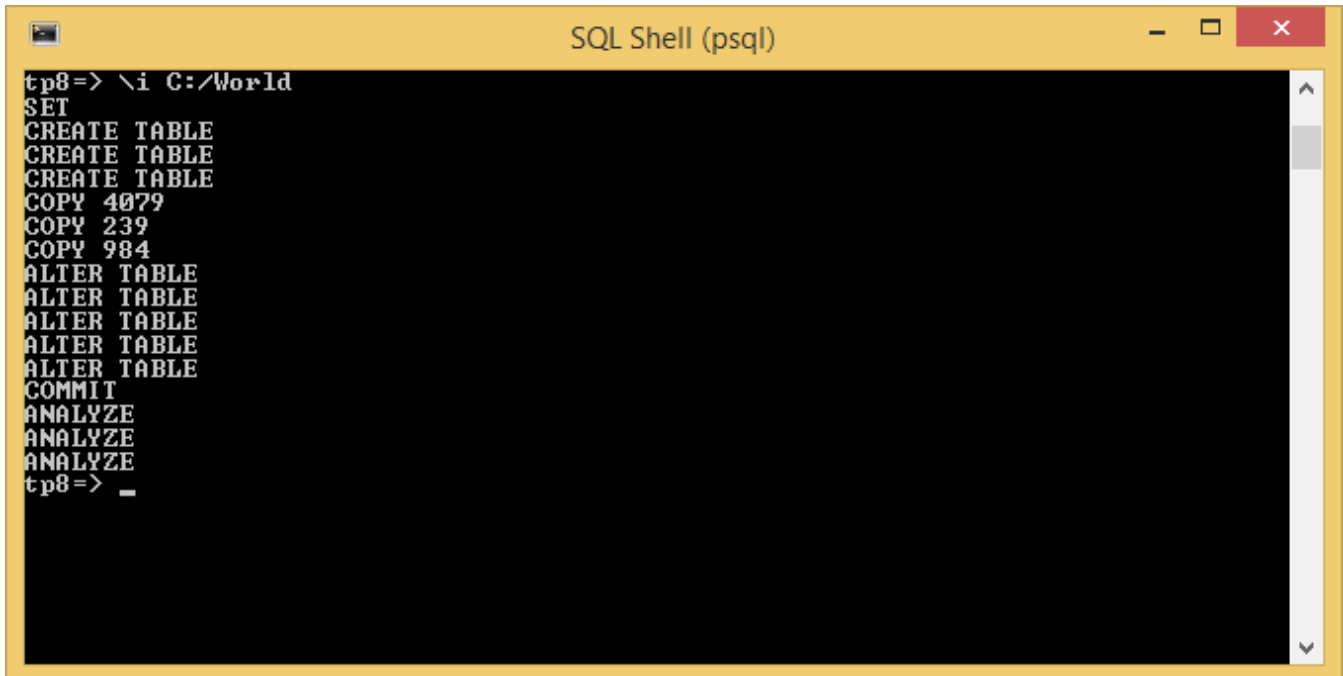
A. World.sql

Login

A screenshot of a terminal window titled "SQL Shell (psql)". The window has a yellow title bar with standard Windows window controls (minimize, maximize, close). The terminal text shows the connection details for a PostgreSQL database named "tp8" on localhost, port 5432, using the username "jskd". It prompts for a password, which is entered as "psql <10.1>". The prompt then changes to "tp8=> _".

```
SQL Shell (psql)
Server [localhost]:
Database [postgres]: tp8
Port [5432]:
Username [postgres]: jskd
Mot de passe pour l'utilisateur jskd :
psql <10.1>
tp8=> _
```

Exéc

A screenshot of a terminal window titled "SQL Shell (psql)". The window has a yellow title bar with standard Windows window controls. The terminal text shows a series of SQL commands being executed in the "tp8" database. The commands include a backslash command to read a file, followed by several CREATE TABLE, COPY, ALTER TABLE, COMMIT, and ANALYZE statements. The prompt "tp8=> _" is visible at the bottom.

```
SQL Shell (psql)
tp8=> \i C:/World
SET
CREATE TABLE
CREATE TABLE
CREATE TABLE
COPY 4079
COPY 239
COPY 984
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
COMMIT
ANALYZE
ANALYZE
ANALYZE
tp8=> _
```

Check

```
SQL Shell (psql)
```

```
tp8-> \dt
```

Schéma	Nom	Type	Propriétaire
public	city	table	jskd
public	country	table	jskd
public	countrylanguage	table	jskd

```
<3 lignes>
```

```
tp8->
```

B. PgAdmin

Réalisez la version graphique de la requête

The screenshot shows the PgAdmin 3 Query Tool interface. The title bar reads "Query - tp8 sur jskd@localhost : 5432 *". The menu bar includes "Fichier", "Édition", "Requêtes", "Favoris", "Macros", "Affichage", and "Aide". The toolbar contains various icons for file operations, query execution, and navigation. The main window is titled "Éditeur SQL" and "Constructeur graphique de requêtes". On the left, a tree view shows the database structure: "tp8" (Catalogues) -> "public" (Schémas) -> "city", "country", "countrylanguage" (Tables). The central area displays a graphical query plan with three tables: "city", "country", and "countrylanguage". The "city" table has columns: id, name (checked), countrycode, district, population. The "country" table has columns: code, name (checked), continent, region, surfacearea, indepyear, population, lifeexpectancy, gnp, gnpold, localname, governmentform, headofstate, capital, code2. The "countrylanguage" table has columns: countrycode, language (checked), isofficial, percentage. Lines with equals signs connect the "countrycode" column of "city" to the "code" column of "country", and the "code" column of "country" to the "countrycode" column of "countrylanguage". Below the query plan, there is a table with columns "Relation", "Colonne", and "Alias". The table contains three rows: 1. "countrylanguage" relation, "language" column. 2. "city" relation, "name" column. 3. "country" relation, "name" column. At the bottom, the status bar shows "OK.", "DOS", "Ligne 1, Col 1, Caract. 1", "30670 lignes.", and "882 msec".

Query - tp8 sur jskd@localhost : 5432 *

Fichier Édition Requêtes Favoris Macros Affichage Aide

tp8 sur jskd@localhost : 5432

Éditeur SQL Constructeur graphique de requêtes

tp8

- Catalogues
- Schémas
 - public
 - city
 - country
 - countrylanguage

city

- ☐ id
- ☒ name
- ☐ countrycode
- ☐ district
- ☐ population

country

- ☐ code
- ☒ name
- ☐ continent
- ☐ region
- ☐ surfacearea
- ☐ indepyear
- ☐ population
- ☐ lifeexpectancy
- ☐ gnp
- ☐ gnpold
- ☐ localname
- ☐ governmentform
- ☐ headofstate
- ☐ capital
- ☐ code2

countrylanguage

- ☐ countrycode
- ☒ language
- ☐ isofficial
- ☐ percentage

Colonne Critère Tri Jointures

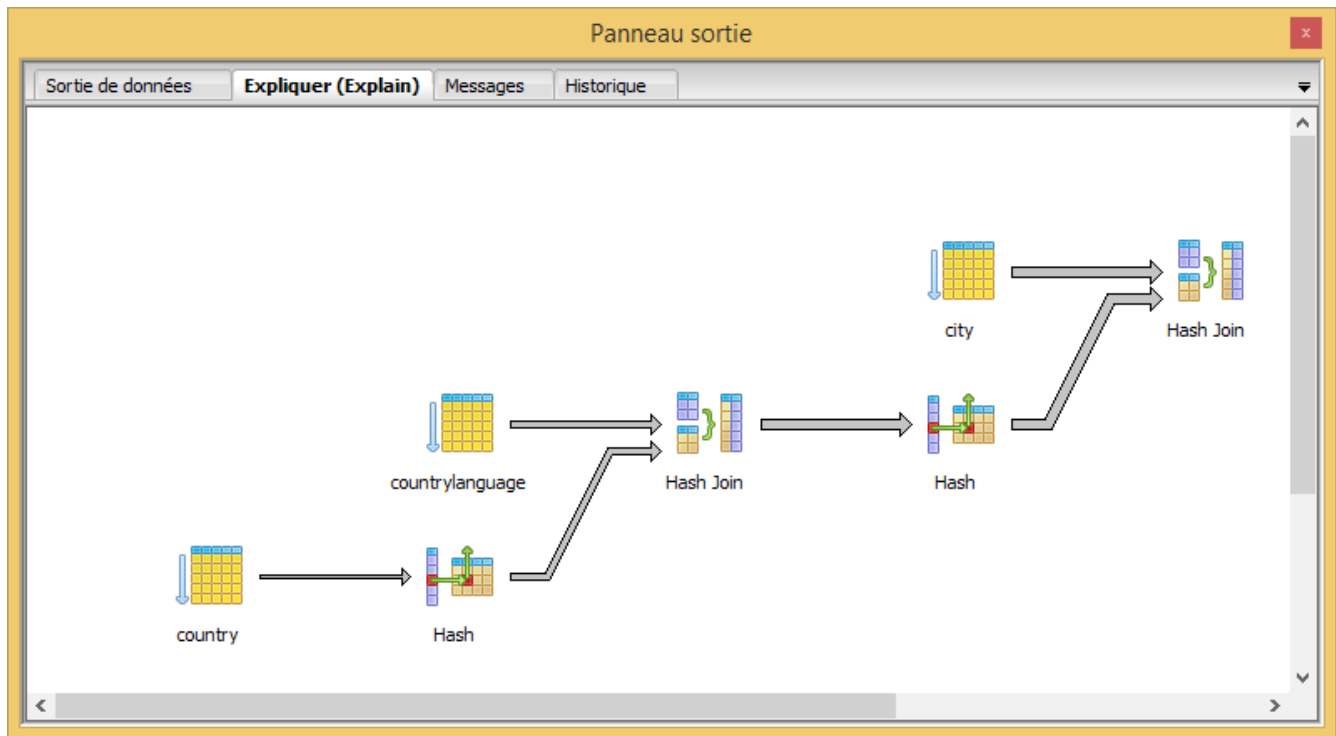
	Relation	Colonne	Alias
1	countrylanguage	language	
2	city	name	
3	country	name	

OK. DOS Ligne 1, Col 1, Caract. 1 30670 lignes. 882 msec

Affichez le résultat de la requête

Panneau sortie			
Sortie de données			
Expliquer (Explain) Messages Historique			
	language text	name text	name text
1	Balochi	Kabul	Afghanistan
2	Turkmenian	Kabul	Afghanistan
3	Uzbek	Kabul	Afghanistan
4	Dari	Kabul	Afghanistan
5	Pashto	Kabul	Afghanistan
6	Balochi	Qandahar	Afghanistan
7	Turkmenian	Qandahar	Afghanistan
8	Uzbek	Qandahar	Afghanistan
9	Dari	Qandahar	Afghanistan
10	Pashto	Qandahar	Afghanistan
11	Balochi	Herat	Afghanistan
12	Turkmenian	Herat	Afghanistan
13	Uzbek	Herat	Afghanistan
14	Dari	Herat	Afghanistan
15	Pashto	Herat	Afghanistan
16	Balochi	Mazar-e-Sharif	Afghanistan
17	Turkmenian	Mazar-e-Sharif	Afghanistan
18	Uzbek	Mazar-e-Sharif	Afghanistan
19	Dari	Mazar-e-Sharif	Afghanistan
20	Pashto	Mazar-e-Sharif	Afghanistan
21	Turkish	Amsterdam	Netherlands
22	Arabic	Amsterdam	Netherlands
23	Fries	Amsterdam	Netherlands
24	Dutch	Amsterdam	Netherlands
25	Turkish	Rotterdam	Netherlands
26	Arabic	Rotterdam	Netherlands
27	Fries	Rotterdam	Netherlands
28	Dutch	Rotterdam	Netherlands
29	Turkish	Haag	Netherlands
30	Arabic	Haag	Netherlands
31	Fries	Haag	Netherlands
32	Dutch	Haag	Netherlands
33	Turkish	Utrecht	Netherlands
34	Arabic	Utrecht	Netherlands
35	Fries	Utrecht	Netherlands
36	Dutch	Utrecht	Netherlands
37	Turkish	Eindhoven	Netherlands
38	Arabic	Eindhoven	Netherlands
39	Fries	Eindhoven	Netherlands
40	Dutch	Eindhoven	Netherlands
41	Turkish	Tilburg	Netherlands
42	Arabic	Tilburg	Netherlands
43	Fries	Tilburg	Netherlands
44	Dutch	Tilburg	Netherlands
45	Turkish	Groningen	Netherlands
46	Arabic	Groningen	Netherlands
47	Fries	Groningen	Netherlands
48	Dutch	Groningen	Netherlands

Affichez la version graphique de l'arbre d'exécution de la requête



Affichez la version texte du résultat de la requête Explain.

```
SQL Shell (psql)

tp8=> EXPLAIN SELECT countrylanguage.language, city.name, country.name
tp8-> FROM public.city, public.country, public.countrylanguage
tp8-> WHERE city.countrycode = country.code AND country.code = countrylanguage.c
countrycode;

               QUERY PLAN

-----
Hash Join  (cost=50.91..322.23 rows=16794 width=28)
  Hash Cond: (city.countrycode = country.code)
    -> Seq Scan on city  (cost=0.00..72.79 rows=4079 width=13)
    -> Hash  (cost=38.61..38.61 rows=984 width=27)
          -> Hash Join  (cost=10.38..38.61 rows=984 width=27)
                Hash Cond: (countrylanguage.countrycode = country.code)
                -> Seq Scan on countrylanguage  (cost=0.00..15.84 rows=984 width
=12)
                -> Hash  (cost=7.39..7.39 rows=239 width=15)
                      -> Seq Scan on country  (cost=0.00..7.39 rows=239 width=15)
    )
  )
(9 lignes)

tp8=>
```

C) Explain Cost

1) requête 1 : plan avec un « seq scan »

1^{er} exemple :

The screenshot shows a database IDE interface. The top menu bar includes 'Fichier', 'Édition', 'Requêtes', 'Favoris', 'Macros', 'Affichage', and 'Aide'. Below the menu is a toolbar with various icons. The main window is titled 'Éditeur SQL' and contains the following SQL query:

```
SELECT
  city.name
FROM
  public.city
WHERE
  city.id > 800;
```

Below the query editor, there is a 'Panneau sortie' (Output Panel) with tabs for 'Sortie de données', 'Exécuter (Explain)', 'Messages', and 'Historique'. The 'Exécuter (Explain)' tab is selected, displaying the execution plan for the query:

Seq Scan
on city
Filter: (id > 800)
(cost=0.00..82.99 rows=3279 width=9)

Below the text, there is a visual representation of the 'city' table as a grid of yellow cells, with a blue arrow pointing downwards, indicating a sequential scan.

2ème Exemple :

Éditeur SQL Constructeur graphique de requêtes

Requêtes précédentes

```
SELECT
  city.name,
  country.name
FROM
  public.city,
  public.country,
  public.countrylanguage
WHERE
  city.countrycode = country.code AND
  country.code = countrylanguage.countrycode AND
  city.id > 800;
```

Panneau sortie

Sortie de données Expliquer (Explain) Messages Historique

Seq Scan
on country
(cost=0.00..7.39 rows=239 width=15)

country

countrylanguage

city

Hash

Hash Join

Hash

Hash Join

OK.

2) requête 2 : plan avec un « index on »

1^{er} exemple :

The screenshot shows a SQL IDE interface. The main window displays the following SQL query:

```
SELECT
  city.name
FROM
  public.city
WHERE
  city.id = 800;
```

Below the query editor, the 'Panneau sortie' (Output Panel) is visible. It contains a tab labeled 'Exécuter (Explain)' which is currently selected. The output shows the execution plan for the query:

Index Scan
using city_pkey on city
Index Cond: (id = 800)
(cost=0.28..8.30 rows=1 width=9)

Below the text, there is a visual representation of the index scan. It consists of a grid of 10 columns and 10 rows. A blue arrow points downwards from the top-left cell to the bottom-left cell, indicating the scan direction. Below the grid, the text 'city_pkey' is displayed.

2ème exemple :

Éditeur SQL Constructeur graphique de requêtes

Requêtes précédentes

```
SELECT
  city.name,
  country.name
FROM
  public.city,
  public.country,
  public.countrylanguage
WHERE
  city.countrycode = country.code AND
  country.code = countrylanguage.countrycode AND
  city.id = 800;
```

Panneau sortie

Sortie de données Expliquer (Explain) Messages Historique

Index Scan
using city_pkey on city
Index Cond: (id = 800)
(cost=0.28..8.30 rows=1 width=13)

```
graph LR
    city_pkey[city_pkey] --> NL1[Nested Loop]
    country_pkey[country_pkey] --> NL1
    NL1 --> NL2[Nested Loop]
    countrylanguage_pkey[countrylanguage_pkey] --> NL2
```