

RAPPORT :

Base de Données et langage SQL

Création d'une base de données : Freedom in the world

SOMMAIRE :

- SCRIPT MANUEL DE CREATION DE BASE DE DONNEES
 - PEUPLEMENT DE TABLES

- **SCRIPT MANUEL DE CREATION DE BASE DE DONNEES**

Voici mon script SQL de création de table :

```
----- SCRIPT DE CREATION DE TABLE DE DONNEES

CREATE TABLE region
(
    region_code INTEGER PRIMARY KEY,
    name VARCHAR
);

CREATE TABLE status
(
    status VARCHAR
);

CREATE TABLE country
(
    id_country SERIAL PRIMARY KEY,
    name VARCHAR,
    region_code INTEGER REFERENCES region(region_code),
    is_ldc BOOLEAN
);

CREATE TABLE freedom
(
    id_country INTEGER REFERENCES country(id_country),
    year INT,
    civil_liberties INT,
    political_rights INT,
    status VARCHAR
);
```

Explication de mon script :

• PEUPLEMENT DES TABLE

Aperçu du fichier « freedom.csv » :

country,year,CL,PR,Status,Region_Code,Region_Name,is_ldc		
Afghanistan,1995,7,7,NF,142,Asia,1		
Afghanistan,1996,7,7,NF,142,Asia,1		
Afghanistan,1997,7,7,NF,142,Asia,1		
Afghanistan,1998,7,7,NF,142,Asia,1		
Afghanistan,1999,7,7,NF,142,Asia,1		
Afghanistan,2000,7,7,NF,142,Asia,1		
Afghanistan,2001,7,7,NF,142,Asia,1		
Afghanistan,2002,6,6,NF,142,Asia,1		
Afghanistan,2003,6,6,NF,142,Asia,1		
Afghanistan,2004,6,5,NF,142,Asia,1		

Etape 1 :

```
CREATE TEMPORARY TABLE data
(
  country VARCHAR,
  year INT,
  cl INT,
  pr INT,
  Status VARCHAR,
  Region_Code SERIAL,
  Region_Name VARCHAR,
  is_ldc BOOLEAN
);
```

```
\COPY data FROM 'C:\Users\dkont\Downloads\freedom.csv' WITH CSV HEADER DELIMITER ',';
```

Pour peupler mes tableaux, j'ai décidé de créer une table temporaire du nom de « data », j'ai fait ce choix pour simplifier mon peuplement. Dans cette table, j'ai mis comme attributs les noms des colonnes du fichier « freedom.csv ». J'ai ensuite peuplé la table data avec le fichier « freedom.csv ».

La table data :

```
postgres=# SELECT * FROM data;
 country | year | cl | pr | status | region_code | region_name | is_ldc
-----+-----+----+----+-----+-----+-----+-----

```

Etape 2 :

```
INSERT INTO region (region_code, name)
SELECT Region_Code, Region_Name
FROM data
ON CONFLICT (region_code) DO NOTHING;
```

Puis à partir de la table
« data », j'ai peuplé mon premier tableau « région ».

J'ai d'abord sélectionné les colonnes que je voulais peupler dans la table région.
Puis j'ai choisi dans quelles colonnes de data j'allais prendre les données pour
peupler la table région. La dernière ligne de mon script est là pour demander à
ne pas s'arrêter sur les messages d'erreur du à la répétition de la clé primaire
region_code dans le fichier « freedom.csv »

Explication de ce script :

1^{ère} ligne : « **INSERT INTO** region (region_code, name) » Ici je demande à insérer des
données dans le tableau region avec entre parenthèse les colonnes que je voudrais
remplir.

2^{ème} et 3^{ème} ligne : « **SELECT** Region_Code, Region_Name » et « **FROM** data » Ici je
sélectionne les colonnes de la table « data » avec « **SELECT** Region_Code,
Region_Name » puis je sélectionne à partir de quelle table extraire ces données ici
avec « **FROM** data ; »

4^{ème} ligne « **ON CONFLICT** (region_code) **DO NOTHING** ; » Elle là pour demander à
ne pas s'arrêter sur les messages d'erreur du à la répétition de la clé primaire
region_code dans le fichier « freedom.csv »

Voici la table que j'obtiens pour région :

```
postgres=# SELECT * FROM region;
```

```
 region_code | name
-----+-----
          142 | Asia
          150 | Europe
             2 | Africa
           19 | Americas
             9 | Oceania
```

```
INSERT INTO status (status)
SELECT Status
FROM data;
```

```
INSERT INTO country (name, region_code, is_ldc)
SELECT country, Region_Code, is_ldc
FROM data;
```

```
INSERT INTO freedom (year, civil_liberties, political_rights, status)
SELECT year, cl, pr, Status
FROM data;
```

```
DROP TABLE data;
```

Etape 3 :

J'ai ensuite fait la même manipulation pour les tables « statuts » puis pour la table « country » et enfin pareil pour la table « Freedom ». Et a la fin, j'ai supprimé ma table temporaire.

Voici respectivement les tables status, country et freedom :

```
db=# select * from status;
```

```
status
-----
NF
NF
NF
NF
NF
NF
NF
NF
NF
NF
PF
PF
PF
NF
NF
---
```

```
db=# select * from country;
```

	id_country	name	region_code
	1	Afghanistan	142
t	2	Afghanistan	142
t	3	Afghanistan	142
t	4	Afghanistan	142
t	5	Afghanistan	142
t	6	Afghanistan	142
t	7	Afghanistan	142
t	8	Afghanistan	142
t	9	Afghanistan	142
t	10	Afghanistan	142
t	11	Afghanistan	142
t	12	Afghanistan	142

```
db=# select * from freedom;
```

	id_country	year	civil_liberties	political_rights	status
		1995	7	7	NF
		1996	7	7	NF
		1997	7	7	NF
		1998	7	7	NF
		1999	7	7	NF
		2000	7	7	NF
		2001	7	7	NF
		2002	6	6	NF
		2003	6	6	NF
		2004	6	5	NF
		2005	5	5	PF
		2006	5	5	PF
		2007	5	5	PF
		2008	6	5	NF
		2009	6	6	NF
		2010	6	6	NF
		2011	6	6	NF
		2012	6	6	NF
		2013	6	6	NF
		2014	6	6	NF
		2015	6	6	NF

J'ai choisi de peupler mes tableaux dans cet ordre car certaines de mes tableaux avaient des clés étrangères. IL ne jamais laisse pointer une Clé étrangère dans le vide. La table « région » a été créée en première suivi de la table « statuts » car ces 2 tables ne contiennent pas de clé étrangère. Puis vient la table « country » qui contient une clé étrangère (region_code) provenant de la clé primaire de la table région(region_code). La table « freedom » a été créée en dernier car elle contient des clés étrangères dont les clés primaires se trouvent dans tous les tableaux créés avant.