

Gnanapragasam
INFORMATIQUE Royston
Euros

BUT 1

Base de Données et langage SQL

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

Ce rapport est réalisé dans le cadre de la SAÉ S104 “**Création d'une base de données**” dont le sujet est “**Freedom in the world**”.

Sommaire:

I.Script manuel de création de la base de données.....	2
II.Modélisation et script de création “avec AGL”.....	3
III.Peuplement des tables.....	5

I.Script manuel de création de la base de données

L'objectif de cette première partie est de réaliser manuellement un script SQL afin de créer les différentes tables présente dans la base de données “Freedom in the world”.

Pour ce faire nous allons utiliser trois type de data différentes:

- **INTEGER:**Pour les data de type numérique
- **VARCHAR:**Pour les data de type chaîne de caractère
- **BOOLEAN:**Pour les data de type vrai ou faux (1 ou 0)

Dans un premier temps nous allons créer une base de donnée propre à la SAÉ:

```
CREATE DATABASE freedom_in_the_world;
```

Puis nous allons créer les différentes tables:
“REGION”, “COUNTRY”, “FREEDOM”, “STATUS”.

Pour ce faire nous allons suivre le schéma relationnel de la base de données:

```
- region(region_code, name)
- status(status)
- country(id_country, name, region_code, is_ldc) où
region_code est une clé étrangère qui fait référence au schéma de la relation
region
- freedom(id_country, civil_liberties, political_rights,
status) où status et id_country sont des clés étrangères qui font
respectivement références aux schémas de relation de status et de country
```

```
CREATE TABLE region(
    region_code INTEGER PRIMARY KEY,      #region_code étant
unique on le met en clé primaire
    name VARCHAR);
```

```
CREATE TABLE country(
    id_country SERIAL UNIQUE PRIMARY KEY,  #SERIAL permet
d'auto-incrémentation des valeur pour chaque ligne unique
    country VARCHAR, #le nom de chaque pays
```

```

region_code INTEGER REFERENCES region, # clé étrangère
faisant référence à la table region
    is_ldc BOOLEAN NOT NULL # on place boolean comme type
car c'est soit 0 soit 1
);

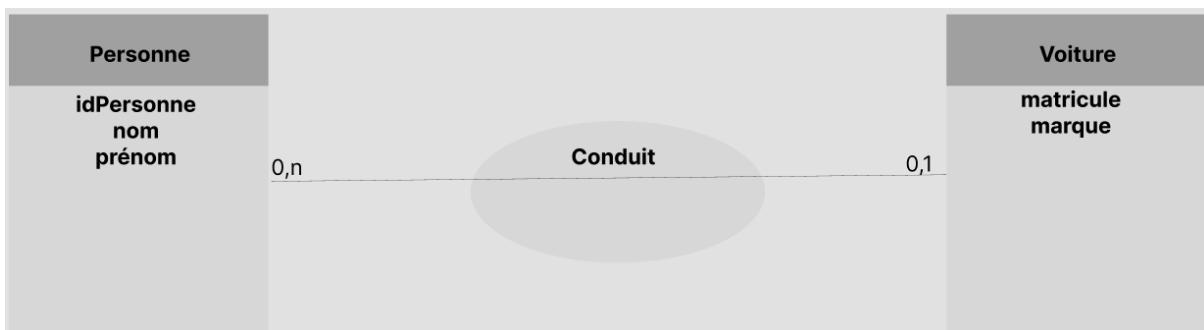
CREATE TABLE status(
    status VARCHAR PRIMARY KEY
);

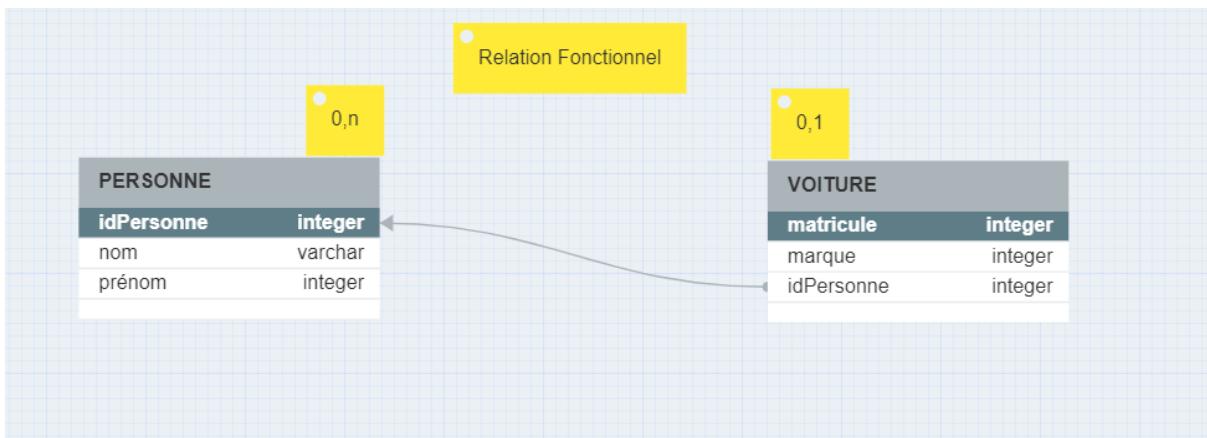
CREATE TABLE freedom(
    year INTEGER,
    id_country INTEGER REFERENCES country,
    civil_liberties INTEGER,
    political_rights INTEGER,
    status VARCHAR REFERENCES status,
    PRIMARY KEY(id_country,year) # la clé primaire étant
l'ensemble id_country,year)
);

```

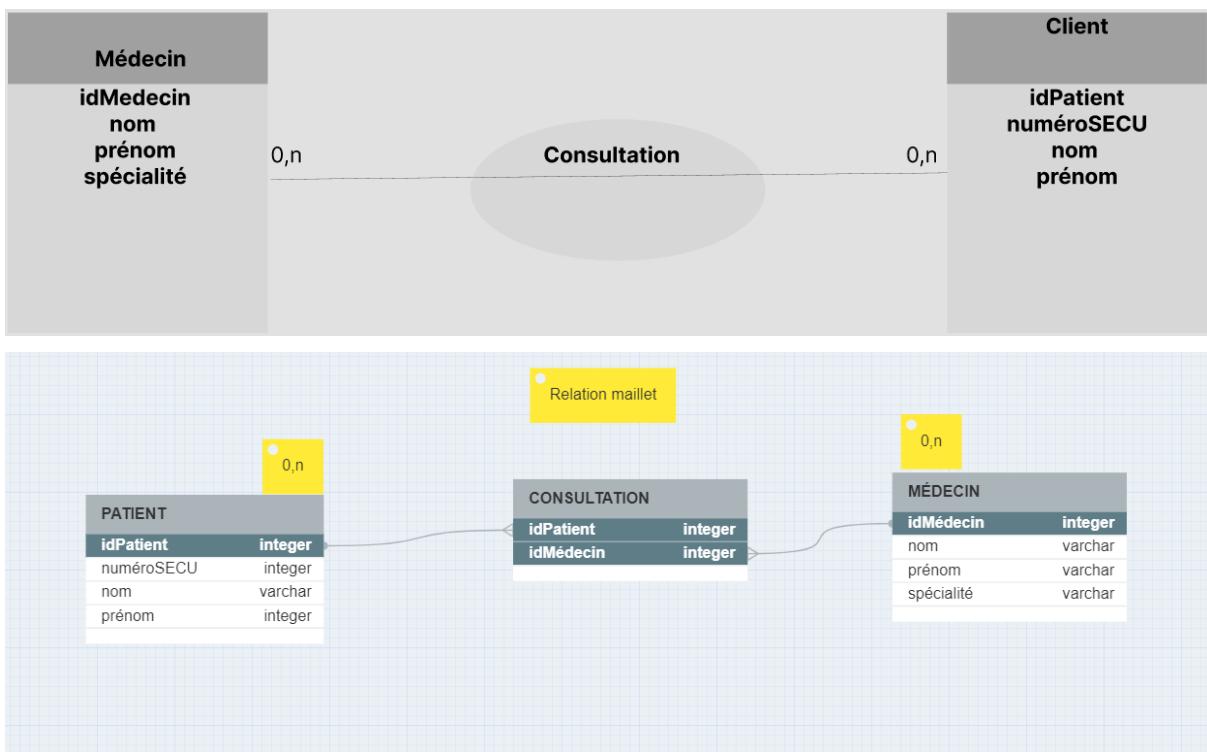
II.Modélisation et script de création “avec AGL”

Voici une comparaison entre une association fonctionnel du cours et de l'AGL:

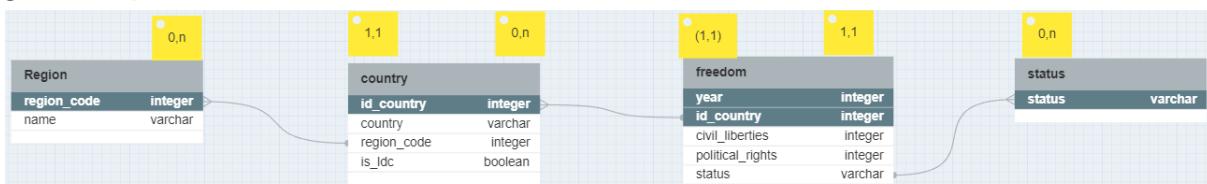




Voici une comparaison entre une association maillée du cours et de l'AGL:



Voici une modélisation physique de donnée réalisé avec l'AGL et le script générer par celui-ci:



```

Region {
    region_code integer pk
    name varchar
}

```

```

country {
    id_country integer pk increments
    country varchar
    region_code integer >* Region.region_code
    is_ldc boolean
}

status {
    status varchar pk
}

freedom {
    year integer pk
    id_country integer pk >* country.id_country
    civil_liberties integer
    political_rights integer
    status varchar >* status.status
}

```

Lorsque l'on compare le script de l'AGL et le notre on retrouve des différence au niveau de l'orthographe (pk pour "primary key") mais aussi les clé étrangère qui diffèrent notamment avec le ">" qui équivaut au "references" et enfin la possibilité de mettre en clé primaire plusieur élément en le signalant sur la ligne même au lieu de créer une nouvelle ligne pour le dire(voir freedom)

III. Peuplement des tables

Afin de peupler la table nous allons copier le fichier plat fourni.Nous devons d'abord créer une table temporaire 'tempo'

```

CREATE TABLE tempo (
    country VARCHAR NOT NULL,
    year INTEGER,
    CL INTEGER,
    PR INTEGER,          # On associe chaque colonne a son
type
    Status VARCHAR,
    region_code INTEGER,
    region_name VARCHAR,

```

```
    is_ldc BOOLEAN  
);
```

Puis en nous mettant dans le répertoire où se trouve le fichier freedom.csv puis on copie les valeur dans notre table temporaire

```
\cd [C:\Users\Royston\Documents\BUT\SQL]
```

```
\COPY tempo FROM 'freedom.csv' DELIMITER ',' CSV HEADER;
```

On peut à présent peupler les tables avec la formule:

```
INSERT INTO [table à peupler] # commande qui permet  
l'insertion de données dans une table  
SELECT DISTINCT [colonne à insérer] #permet de filtrer et  
récupérer les colonnes demander  
FROM [table d'origine]; #défini la table d'origine des  
colonnes
```

Nous avons donc ces différentes formule:

```
INSERT INTO region  
SELECT DISTINCT region_code,region_name  
FROM tempo;
```

```
INSERT INTO country(country,region_code,is_ldc) # la  
parenthèse permet de spécifier les colonnes  
SELECT DISTINCT country,region_code,is_ldc  
FROM tempo;
```

```
INSERT INTO status  
SELECT DISTINCT status  
FROM tempo;
```

```
INSERT INTO freedom  
SELECT year,id_country,cl,pr,status  
FROM tempo  
JOIN country AS new_tab USING (country); #permet de joindre  
la table temporaire et country pour ajouter des élément de  
country et tempo
```

cela permet donc de retrouver respectivement les table
region,status,country et freedom peupler

```
country;
aquabdd:5432 12305804@etudiants=> SELECT*FROM
aquabdd:5432 12305804@etudiants=> SELECT*FROM
+-----+-----+
| region_code |      name   |
+-----+-----+
|      150    | Europe     |
|        9    | Oceania    |
|        2    | Africa     |
|     142    | Asia       |
|      19    | Americas   |
+-----+-----+
(5 rows)
```

```
aquabdd:5432 12305804@etudiants=> █
```

```
| 12305568 | region          | table   | 12305568 |
| 12305568 | status           | table   | 12305568 |
| 12305568 | temp            | table   | 12305568 |
+-----+-----+-----+
(11 rows)
```

```
aquabdd:5432 12305568@etudiants=> SELECT* FROM temp
aquabdd 12305568@etudiants-> ;
aquabdd:5432 12305568@etudiants=> SELECT* FROM country;
aquabdd:5432 12305568@etudiants=> SELECT* FROM region;
```

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

```
aquabdd:5432 12305568@etudiants=> SELECT* FROM status
aquabdd 12305568@etudiants-> ;
+-----+
| status |
+-----+
| PF    |
| F     |
| NF    |
+-----+
(3 rows)
```

```
aquabdd:5432 12305568@etudiants=> █
```

173	Russian Federation
174	Botswana
175	Liechtenstein
176	Viet Nam
177	Venezuela (Bolivarian Republic of)
178	Costa Rica
179	Equatorial Guinea
180	Japan
181	San Marino
182	Marshall Islands
183	Hungary
184	Djibouti
185	Iraq
186	Seychelles
187	Niger
188	Singapore
189	Antigua and Barbuda
190	Jamaica
191	Malawi
192	Lao People's Democratic Republic
193	Ghana

+-----+(193 rows)

2012	193	2		1	F	
2011	193	2		1	F	
2010	193	2		1	F	
2009	193	2		1	F	
2008	193	2		1	F	
2007	193	2		1	F	
2006	193	2		1	F	
2005	193	2		1	F	
2004	193	2		1	F	
2003	193	2		2	F	
2002	193	2		2	F	
2001	193	3		2	F	
2000	193	3		2	F	
1999	193	3		3	PF	
1998	193	3		3	PF	
1997	193	3		3	PF	
1996	193	4		3	PF	
1995	193	4		4	PF	

+-----+(4979 rows)

(END)

Nous avons donc rempli l'objectif de cette SAE qui est de concevoir, mettre en place puis alimenter une base de données permettant de stocker des mesures de sentiments, de degré de liberté et de droits politiques par pays.