JEGATHASAN Amirtavarsini
Groupe: Cérynie

SAE Climat Base de donnée

SOMMAIRE:

2.1- Script manuel de la création de la base de donnée

J’ai rencontrée un problème avec mon programme sql manuel, car quand je lance le script général en entier sur mon psql, je rencontre un problème avec le (\copy), ce qui me mets (ERROR: syntax error at or near "\"
LINE 2: \copy temp_2 FROM /home/jegathasan/SAE_climat/climate.csv DE…),
mais lorsque que je lance mon script petit à petit sois en commençant d'abord par ca:

```
DROP TABLE IF EXISTS CLIMATE_DISASTER;
DROP TABLE IF EXISTS DISASTER;
DROP TABLE IF EXISTS COUNTRY;
DROP TABLE IF EXISTS SUB_REGION;
DROP TABLE IF EXISTS REGION;
DROP TABLE IF EXISTS TEMP_1;
DROP TABLE IF EXISTS TEMP_2;

CREATE TABLE REGION (region_code INT PRIMARY KEY, name VARCHAR NOT NULL);
CREATE TABLE SUB_REGION (sub_region_code INT PRIMARY KEY, name VARCHAR NOT NULL, region_code INT REFERENCES region (region_code) ON DELETE SET NULL);
CREATE TABLE COUNTRY (country_code INT PRIMARY KEY, name VARCHAR NOT NULL, ISO2 CHAR(2), ISO3 CHAR(3), sub_region_code INT REFERENCES sub_region (sub_region_code) ON DELETE SET NULL);
CREATE TABLE DISASTER (disaster_code SERIAL PRIMARY KEY, disaster VARCHAR NOT NULL);
CREATE TABLE CLIMATE_DISASTER (country_code INT REFERENCES country (country_code) ON DELETE SET NULL, disaster_code INT REFERENCES disaster (disaster_code) ON DELETE SET NULL, year INT , number INT );
ALTER TABLE CLIMATE_DISASTER ADD CONSTRAINT CLIMATE_DISASTER_pkey PRIMARY KEY (country_code, disaster_code, year);
CREATE TABLE TEMP_1 (name VARCHAR, alpha_2 VARCHAR, alpha_3 VARCHAR, country_code INT, iso_3166_2 VARCHAR, region VARCHAR, sub_region VARCHAR, intermediate_region VARCHAR, region_code INTEGER, sub_region_code INT, intermediate_region_code VARCHAR );
CREATE TABLE TEMP_2 (country VARCHAR , iso2 CHAR(2), iso3 CHAR(3), region_code INT, region VARCHAR, sub_region_code VARCHAR, sub_region VARCHAR, disaster VARCHAR, year INT, number INT);
```

Voici le résultat obtenue pour cette partie de script:

```
DROP TABLE
DROP TABLE
DROP TABLE
DROP TABLE
DROP TABLE
DROP TABLE
DROP TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
```

CREATE TABLE
CREATE TABLE
ALTER TABLE
CREATE TABLE
CREATE TABLE

puis ca:

\copy temp_1 FROM /home/jegathasan/SAE_climat/github.csv DELIMITER ',' CSV HEADER
résultat: COPY 249

ensuite ca:

\copy temp_2 FROM /home/jegathasan/SAE_climat/climate.csv DELIMITER ',' CSV HEADER
résultat: COPY 6448

et enfin par ca:

INSERT INTO region (name, region_code) SELECT DISTINCT region, region_code FROM temp_1 WHERE region_code IS NOT NULL;
INSERT INTO sub_region (name, sub_region_code, region_code) SELECT DISTINCT sub_region, sub_region_code, region_code FROM temp_1 WHERE region_code IS NOT NULL;
INSERT INTO country (name, country_code, sub_region_code) SELECT DISTINCT name,country_code,sub_region_code FROM temp_1 WHERE sub_region_code IS NOT NULL;
UPDATE country SET iso2=temp_1.alpha_2, iso3=temp_1.alpha_3 FROM temp_1 WHERE country.name = temp_1.name;
INSERT INTO disaster (disaster) SELECT DISTINCT disaster FROM temp_2 WHERE disaster IS NOT NULL;
INSERT INTO climate_disaster (country_code, disaster_code, year,number) SELECT country_code, disaster_code,year, number FROM country, disaster, temp_2 WHERE temp_2.country=country.name AND temp_2.disaster=disaster.disaster AND temp_2.year IS NOT NULL AND temp_2.number IS NOT NULL GROUP BY (country_code, disaster_code, year, number);

résultat:
INSERT 0 5
INSERT 0 17
INSERT 0 247
UPDATE 247
INSERT 0 6
INSERT 0 4518

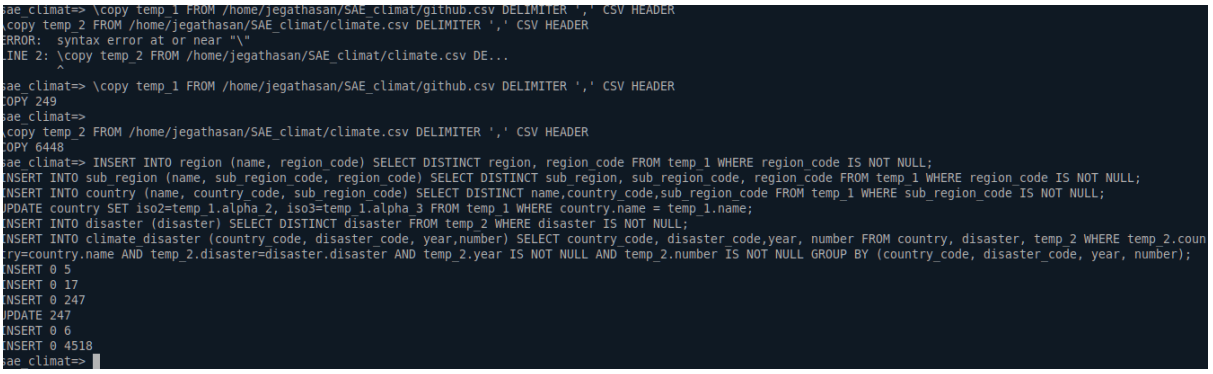
Ça marche correctement, uniquement lorsque que je fais comme ça, mais en lançant le script en entier ça me mets des erreurs et je ne comprends pas pourquoi. Mais je vous mets quand même le script en entier:

```
DROP TABLE IF EXISTS CLIMATE_DISASTER;
DROP TABLE IF EXISTS DISASTER;
DROP TABLE IF EXISTS COUNTRY;
DROP TABLE IF EXISTS SUB_REGION;
DROP TABLE IF EXISTS REGION;
DROP TABLE IF EXISTS TEMP_1;
DROP TABLE IF EXISTS TEMP_2;

CREATE TABLE REGION (region_code INT PRIMARY KEY, name VARCHAR NOT NULL);
CREATE TABLE SUB_REGION (sub_region_code INT PRIMARY KEY, name VARCHAR NOT NULL, region_code INT REFERENCES region (region_code) ON DELETE SET NULL);
CREATE TABLE COUNTRY (country_code INT PRIMARY KEY, name VARCHAR NOT NULL, ISO2 CHAR(2), ISO3 CHAR(3), sub_region_code INT REFERENCES sub_region (sub_region_code) ON DELETE SET NULL);
CREATE TABLE DISASTER (disaster_code SERIAL PRIMARY KEY, disaster VARCHAR NOT NULL);
CREATE TABLE CLIMATE_DISASTER (country_code INT REFERENCES country (country_code) ON DELETE SET NULL, disaster_code INT REFERENCES disaster (disaster_code) ON DELETE SET NULL, year INT , number INT );
ALTER TABLE CLIMATE_DISASTER ADD CONSTRAINT CLIMATE_DISASTER_pkey PRIMARY KEY (country_code, disaster_code, year);
CREATE TABLE TEMP_1 (name VARCHAR, alpha_2 VARCHAR, alpha_3 VARCHAR, country_code INT, iso_3166_2 VARCHAR, region VARCHAR, sub_region VARCHAR, intermediate_region VARCHAR, region_code INTEGER, sub_region_code INT, intermediate_region_code VARCHAR );
CREATE TABLE TEMP_2 (country VARCHAR , iso2 CHAR(2), iso3 CHAR(3), region_code INT, region VARCHAR, sub_region_code VARCHAR, sub_region VARCHAR, disaster VARCHAR, year INT, number INT);
\copy temp_1 FROM /home/jegathasan/SAE_climat/github.csv DELIMITER ',' CSV HEADER
\copy temp_2 FROM /home/jegathasan/SAE_climat/climate.csv DELIMITER ',' CSV HEADER
INSERT INTO region (name, region_code) SELECT DISTINCT region, region_code FROM temp_1 WHERE region_code IS NOT NULL;
INSERT INTO sub_region (name, sub_region_code, region_code) SELECT DISTINCT sub_region, sub_region_code, region_code FROM temp_1 WHERE region_code IS NOT NULL;
INSERT INTO country (name, country_code, sub_region_code) SELECT DISTINCT name,country_code,sub_region_code FROM temp_1 WHERE sub_region_code IS NOT NULL;
UPDATE country SET iso2=temp_1.alpha_2, iso3=temp_1.alpha_3 FROM temp_1 WHERE country.name = temp_1.name;
INSERT INTO disaster (disaster) SELECT DISTINCT disaster FROM temp_2 WHERE disaster IS NOT NULL;
```

INSERT INTO climate_disaster (country_code, disaster_code, year,number) SELECT country_code, disaster_code,year, number FROM country, disaster, temp_2 WHERE temp_2.country=country.name AND temp_2.disaster=disaster.disaster AND temp_2.year IS NOT NULL AND temp_2.number IS NOT NULL GROUP BY (country_code, disaster_code, year, number);

Voici, un screen des résultats:

```
jegathasan=> \c sae_climat
You are now connected to database "sae_climat" as user "jegathasan".
sae_climat=> DROP TABLE IF EXISTS CLIMATE_DISASTER;
DROP TABLE IF EXISTS DISASTER;
DROP TABLE IF EXISTS COUNTRY;
DROP TABLE IF EXISTS SUB_REGION;
DROP TABLE IF EXISTS REGION;
DROP TABLE IF EXISTS TEMP_1;
DROP TABLE IF EXISTS TEMP_2;

CREATE TABLE REGION (region_code INT PRIMARY KEY, name VARCHAR NOT NULL);
CREATE TABLE SUB_REGION (sub_region_code INT PRIMARY KEY, name VARCHAR NOT NULL, region_code INT REFERENCES region (region_code) ON DELETE SET NULL);
CREATE TABLE COUNTRY (country_code INT PRIMARY KEY, name VARCHAR NOT NULL, ISO2 CHAR(2), ISO3 CHAR(3), sub_region_code INT REFERENCES sub_region (sub_region_code) ON DE
LETE SET NULL);
CREATE TABLE DISASTER (disaster_code SERIAL PRIMARY KEY, disaster VARCHAR NOT NULL);
CREATE TABLE CLIMATE_DISASTER (country_code INT REFERENCES country (country_code) ON DELETE SET NULL, disaster_code INT REFERENCES disaster (disaster_code) ON DELETE SE
T NULL, year INT , number INT );
ALTER TABLE CLIMATE_DISASTER ADD CONSTRAINT CLIMATE_DISASTER_pkey PRIMARY KEY (country_code, disaster_code, year);
CREATE TABLE TEMP_1 (name VARCHAR, alpha_2 VARCHAR, alpha_3 VARCHAR, country_code INT, iso_3166_2 VARCHAR, region VARCHAR, sub_region VARCHAR, intermediate_region VARCH
AR, region_code INTEGER, sub_region_code INT, intermediate_region_code VARCHAR );
CREATE TABLE TEMP_2 (country VARCHAR , iso2 CHAR(2), iso3 CHAR(3), region_code INT, region VARCHAR, sub_region_code VARCHAR, sub_region VARCHAR, disaster VARCHAR, year
INT, number INT);
DROP TABLE
DROP TABLE
DROP TABLE
DROP TABLE
DROP TABLE
DROP TABLE
DROP TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
ALTER TABLE
CREATE TABLE
CREATE TABLE
sae_climat=> \copy temp_1 FROM /home/jegathasan/SAE_climat/github.csv DELIMITER ',' CSV HEADER
```

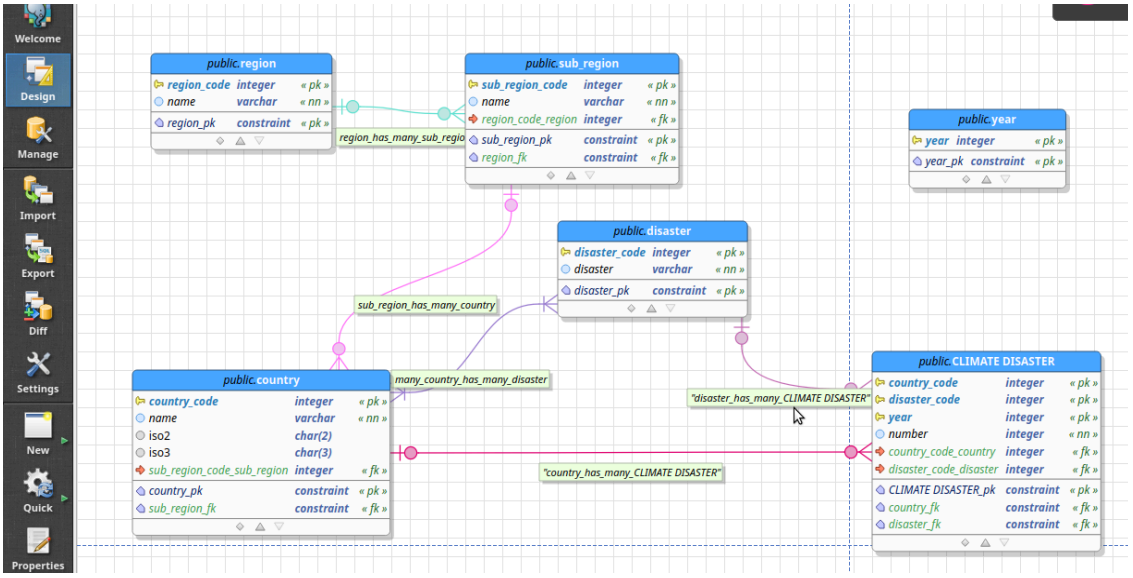


```
sae_climat=> \copy temp_1 FROM /home/jegathasan/SAE_climat/github.csv DELIMITER ',' CSV HEADER
\copy temp_2 FROM /home/jegathasan/SAE_climat/climate.csv DELIMITER ',' CSV HEADER
ERROR:  syntax error at or near "\"
LINE 2: \copy temp_2 FROM /home/jegathasan/SAE_climat/climate.csv DE...
        ^
sae_climat=> \copy temp_1 FROM /home/jegathasan/SAE_climat/github.csv DELIMITER ',' CSV HEADER
COPY 249
sae_climat=>
\copy temp_2 FROM /home/jegathasan/SAE_climat/climate.csv DELIMITER ',' CSV HEADER
COPY 6448
sae_climat=> INSERT INTO region (name, region_code) SELECT DISTINCT region, region_code FROM temp_1 WHERE region_code IS NOT NULL;
INSERT INTO sub_region (name, sub_region_code, region_code) SELECT DISTINCT sub_region, sub_region_code, region_code FROM temp_1 WHERE region_code IS NOT NULL;
INSERT INTO country (name, country_code, sub_region_code) SELECT DISTINCT name,country_code,sub_region_code FROM temp_1 WHERE sub_region_code IS NOT NULL;
UPDATE country SET iso2=temp_1.alpha_2, iso3=temp_1.alpha_3 FROM temp_1 WHERE country.name = temp_1.name;
INSERT INTO disaster (disaster) SELECT DISTINCT disaster FROM temp_2 WHERE disaster IS NOT NULL;
INSERT INTO climate_disaster (country_code, disaster_code, year,number) SELECT country_code, disaster_code,year, number FROM country, disaster, temp_2 WHERE temp_2.coun
try=country.name AND temp_2.disaster=disaster.disaster AND temp_2.year IS NOT NULL AND temp_2.number IS NOT NULL GROUP BY (country_code, disaster_code, year, number);
INSERT 0 5
INSERT 0 17
INSERT 0 247
UPDATE 247
INSERT 0 6
INSERT 0 4518
sae_climat=>
```

2.2-Modélisation et script “avec l’AGL”

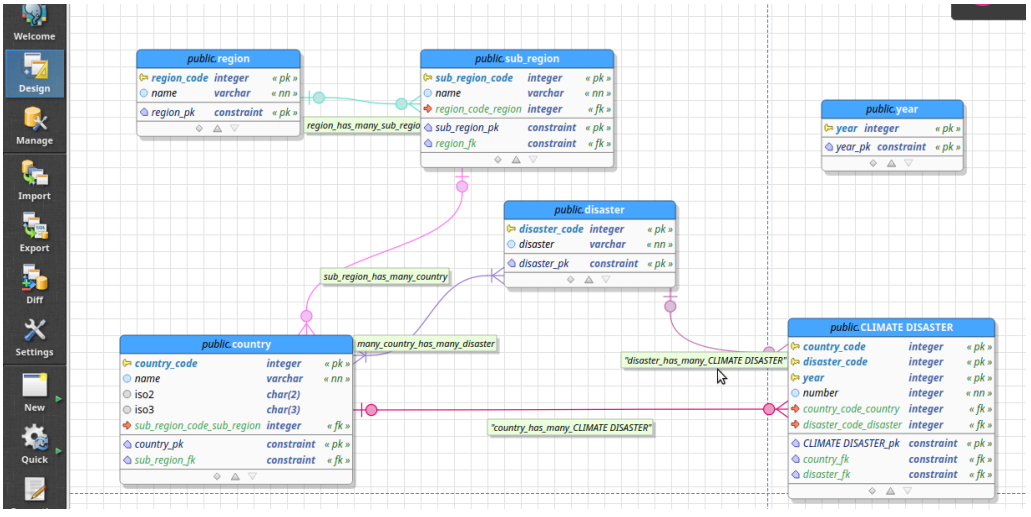
Choix du AGL, j’ai choisi le **pgModeler** car c'était celui que j’ai réussi à installer correctement sur mon ordinateur. Pour comprendre comment l’AGL fonctionne j’ai regardé des tuto sur youtube, mais en je n’avais trouvé que des tuto en anglais. Ensuite j’ai essayé de faire une modélisation fonctionnelle avec pgModeler en regardant les cours en introduction BDD.

Type-association fonctionnelle:



Les relations fonctionnelles soit (one-to-many), sont celles où une entité dépend directement d'une autre. Il s'agit comme une relation simple entre parent-enfant où une clé étrangère pointe vers une clé primaire d'une autre table. Par exemple, dans ce que j'ai pu faire avec l'AGL, j'ai pu fair une relation fonctionnelle entre, REGION —> SUB_REGION car une sous-région appartient à une seule région, SUB_REGION —> COUNTRY car un pays appartient à une seule sous-région, DISASTER —> CLIMATE_DISASTERoù un événement dans la table CLIMATE_DISASTER st lié à un seul type de catastrophes et enfin COURY —> CLIMATE_DISASTER où on a les donnés de catastrophes climatiques se produisant dans un pays spécifique.

Type-association maillées:



Les relations maillées soit (many-to-many), sont des relations complexes où on a une entité qui peut être associée à plusieurs autres entités via des relations croisées. J'ai pu faire une seule relation maillée dans ce sujets avec l'AGL, c'est avec COUNTRY —> DISASTER + CLIMATE_DISASTER, je n'ai pas su faire l'association du climate_disaster dans mon AGL.

J’ai eu beaucoup de mal avec l’utilisation de l’AGL, j’ai fait ce que j’ai pu faire.

2.3 Peuplement des tables

Je n’ai pas réussi à générer le script par l’AGL et faire la partie 2.3, je préfère être honnête avec vous et dire que ce projet était super difficile et je n’ai pas voulu copier sur mes camarades ou autres. J’ai fais ce que j’ai pu faire mais le reste j’ai pas réussi. Je suis consciente de la note que j’aurais, mais je préfère faire ça correctement que de copier et d’avoir une note qui ne me correspond pas. Je vous prie de m'excuser pour ce travail incomplet.