Rapport final

Création et Peuplement de la Base de Données

Catastrophes Climatiques

Nom : *Hachim*

Prénom : *Mohammed*

Groupe : Cérynie

Date : *13 janvier 2025*

Nom de la SAE : *SAE Base de Données (S1.04)*

-Sommaire

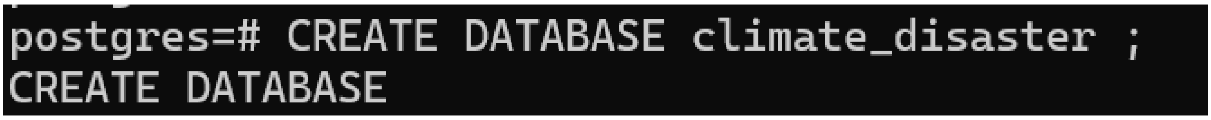
Les scripts sont écrits dans Visual Studio Code, puis je les copie-colle dans mon rapport pour obtenir une police bien lisible et différente de l'Arial

1. Script Manuel de création de la base de données :

```
CREATE DATABASE climate_disaster ;
\c climate_disaster ;

 CREATE TABLE region ( region_code INTEGER PRIMARY KEY , name  VARCHAR
NOT NULL);

 CREATE TABLE sub_region ( name VARCHAR NOT NULL , region_code INTEGER
NOT NULL REFERENCES region (region_code), sub_region_code INTEGER
 PRIMARY KEY);

 CREATE TABLE country ( name VARCHAR NOT NULL , ISO2 CHAR(2) UNIQUE ,
ISO3 CHAR(3)  UNIQUE , sub_region_code INTEGER NOT NULL REFERENCES
sub_region(sub_region_code) , country_code INTEGER PRIMARY KEY);

 CREATE TABLE disaster (disaster_code INTEGER PRIMARY KEY , disaster
VARCHAR NOT NULL UNIQUE );

CREATE TABLE climate_disaster (country_code INTEGER NOT NULL REFERENCES
country (country_code) , disaster_code INTEGER NOT NULL REFERENCES
disaster( disaster_code) , year INTEGER NOT NULL CHECK(year>0) , number
INTEGER CHECK(number>0) , PRIMARY
KEY(country_code,disaster_code,year));
```

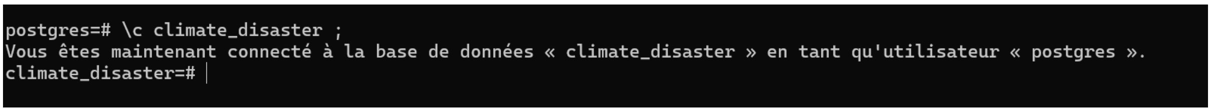
Explication brève du script :

1- On crée une base de données pour effectuer notre travail dedans :



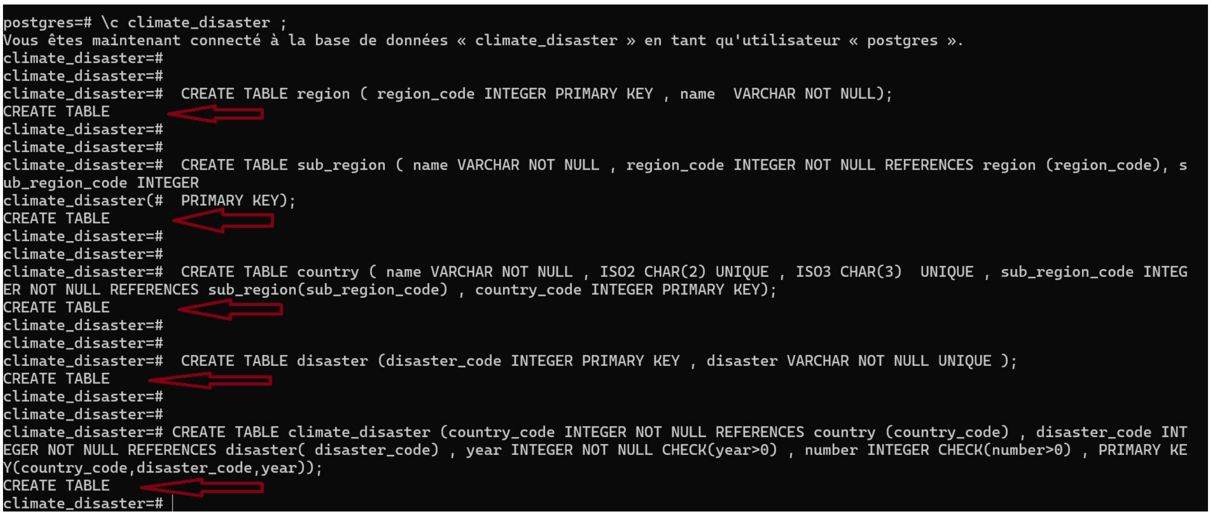
```
postgres=# CREATE DATABASE climate_disaster ;
CREATE DATABASE
```

2- on accède à la base de données qu'on a créée :



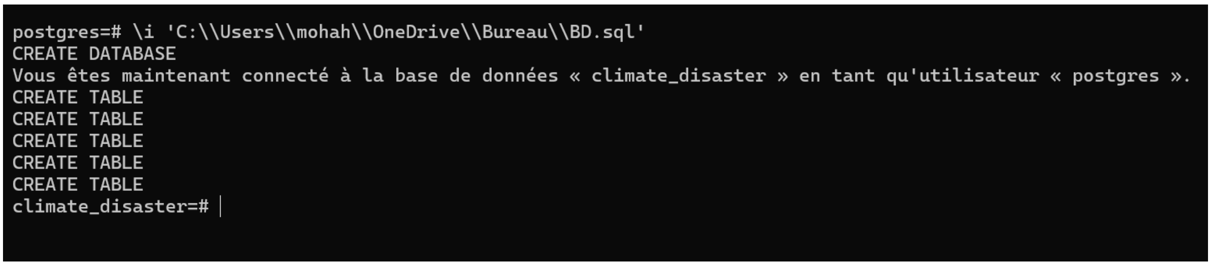
```
postgres=# \c climate_disaster ;
Vous êtes maintenant connecté à la base de données « climate_disaster » en tant qu'utilisateur « postgres ».
climate_disaster=# |
```

3- On crée les tables car il s'agit d'une nouvelle base de données. Normalement, elle ne contient pas de tables existantes (Sinon On peut utiliser la commande : DROP table <nom_du_tableau> IF EXISTS) :



```
postgres=# \c climate_disaster ;
Vous êtes maintenant connecté à la base de données « climate_disaster » en tant qu'utilisateur « postgres ».
climate_disaster=#
climate_disaster=#
climate_disaster=#  CREATE TABLE region ( region_code INTEGER PRIMARY KEY , name  VARCHAR NOT NULL);
CREATE TABLE
climate_disaster=#
climate_disaster=#
climate_disaster=#  CREATE TABLE sub_region ( name VARCHAR NOT NULL , region_code INTEGER NOT NULL REFERENCES region (region_code), s
ub_region_code INTEGER
climate_disaster(#  PRIMARY KEY);
CREATE TABLE
climate_disaster=#
climate_disaster=#
climate_disaster=#  CREATE TABLE country ( name VARCHAR NOT NULL , ISO2 CHAR(2) UNIQUE , ISO3 CHAR(3)  UNIQUE , sub_region_code INTEG
ER NOT NULL REFERENCES sub_region(sub_region_code) , country_code INTEGER PRIMARY KEY);
CREATE TABLE
climate_disaster=#
climate_disaster=#
climate_disaster=#  CREATE TABLE disaster (disaster_code INTEGER PRIMARY KEY , disaster VARCHAR NOT NULL UNIQUE );
CREATE TABLE
climate_disaster=#
climate_disaster=#
climate_disaster=# CREATE TABLE climate_disaster (country_code INTEGER NOT NULL REFERENCES country (country_code) , disaster_code INT
EGER NOT NULL REFERENCES disaster( disaster_code) , year INTEGER NOT NULL CHECK(year>0) , number INTEGER CHECK(number>0) , PRIMARY KE
Y(country_code,disaster_code,year));
CREATE TABLE
climate_disaster=# |
```

Affichage de résultat du script par la commande " \i " :

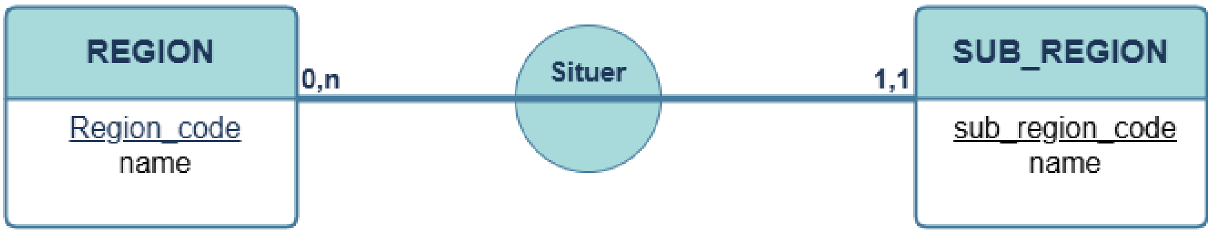


```
postgres=# \i 'C:\\Users\\mohah\\OneDrive\\Bureau\\BD.sql'
CREATE DATABASE
Vous êtes maintenant connecté à la base de données « climate_disaster » en tant qu'utilisateur « postgres ».
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
climate_disaster=# |
```

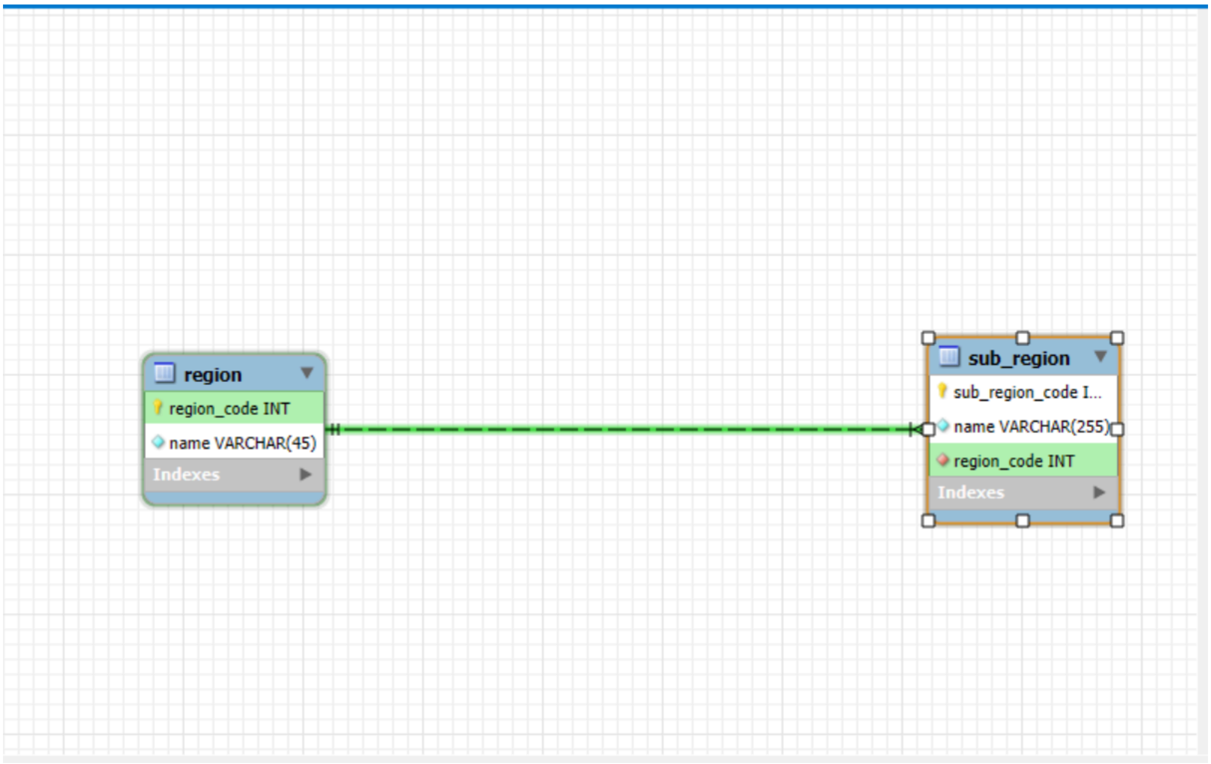
2. Modélisation et script de création avec AGL :

2.1. *Illustrations comparatives cours/AGL commentées d'une association fonctionnelle*

2.1.1. *Représentation d’une association fonctionnelle MCD(modèle conceptuel de données)* :



2.1.2. *Représentation d’une association fonctionnelle MPD(modèle physique de données)⟶AGL* :

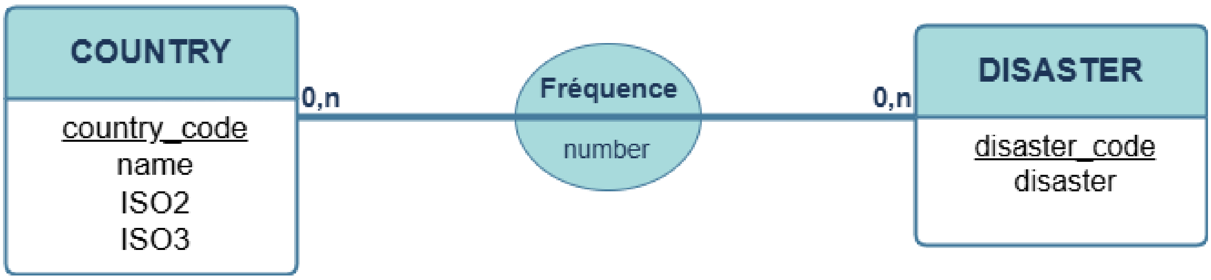


2.1.3. *Comparaison entre la représentation MCD et MPD (cours /AGL) d'une association fonctionnelle :*

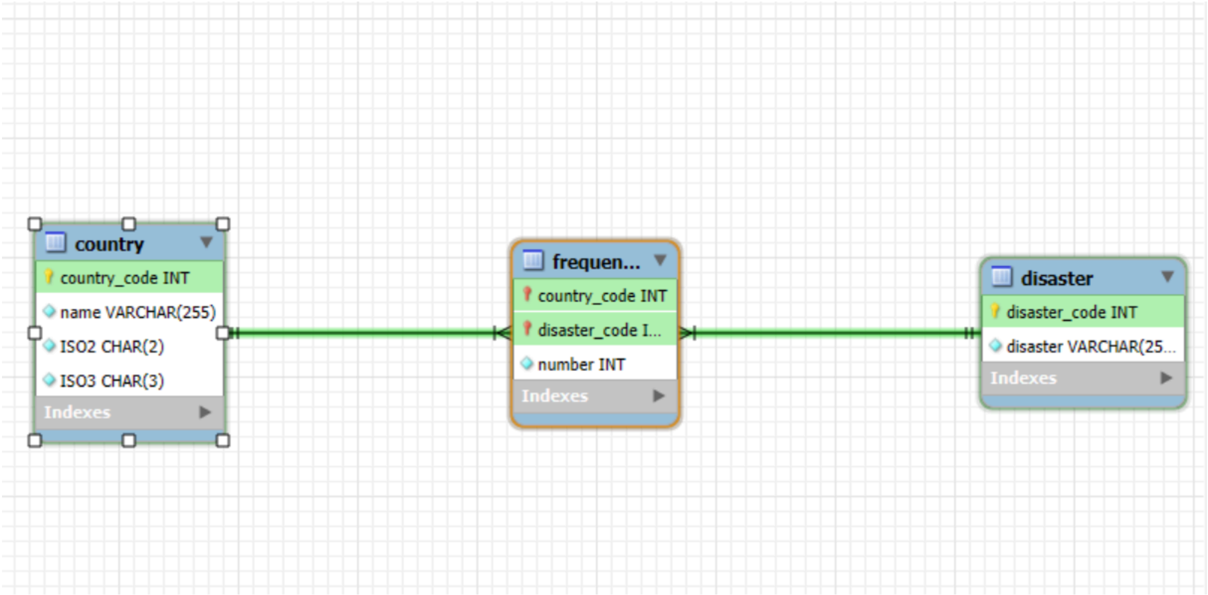
La différence entre le MCD et le MPD est dans la façon dont il représente une association. Nous retrouvons les cardinalités forment une liaison d'entité associative ("1,1" , "0,n"), ainsi l'association entre les deux tableaux est faite par une association lisible ("SITUER" dans notre cas) dans une représentation MCD. Dans le MPD, les entités sont ajoutées grâce à des clés étrangères dans les tables donc l'association n'est pas lisible. Dans certains cas, la représentation MCD met en avant les relations originales alors que le MPC reflète les dépendances.

Ces deux représentations (MCD et MPD) ont des similitudes sur la description des données mais aussi sur la transmission des informations. Le MPD nous offre une compréhension plus technique et complète sur la base de données, alors que la MCD se base sur la facilité conceptuelle

2.2.1. *Représentation d'une association maillée MCD (modèle conceptuel de donnée) :*



2.2.2. *Représentation d'une association maillée MPD(modèle physique de données)⟶AGL :*



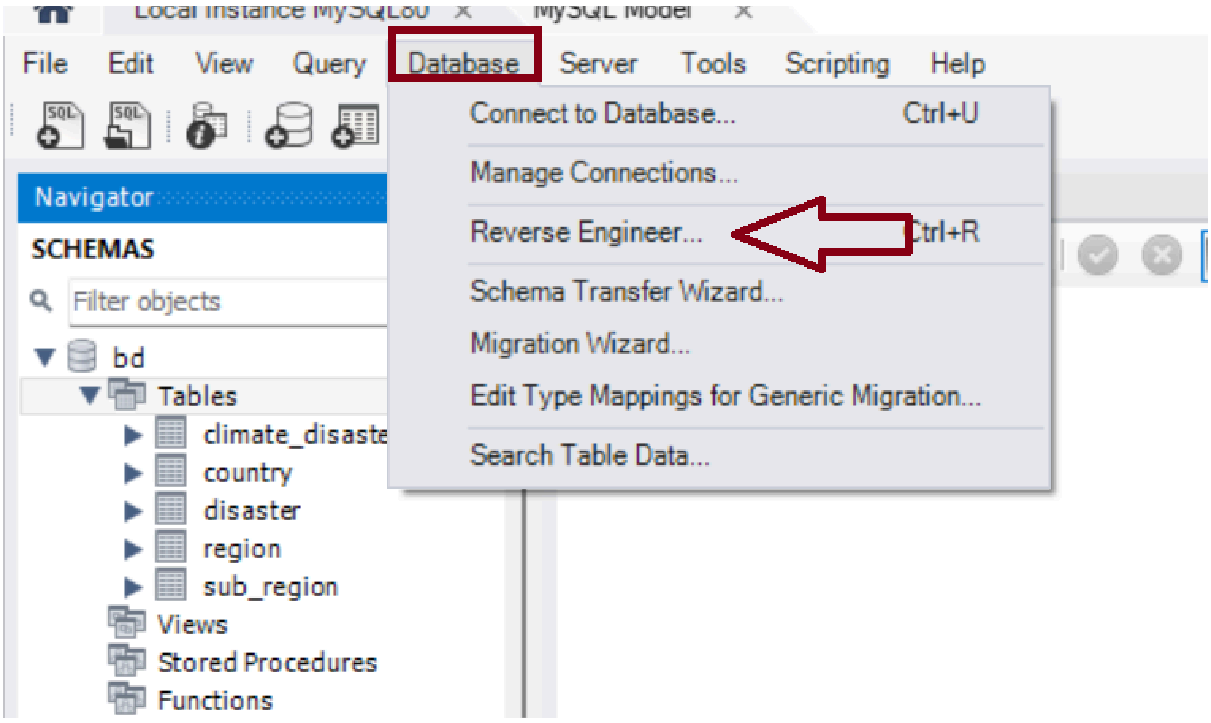
2.1.3. *Comparaison entre la représentation MCD et MPD (cours /AGL) d'une association maillée :*

La représentation MCD a une association Fréquence qui possède l'attribut "number", relie les entités COUNTRY et DISASTER. Dans cette reproduction MCD, on retrouve des cardinalités (0, n), ce qui facilite la compréhension conceptuelle de la représentation : un pays peut avoir plusieurs catastrophes, une catastrophe peut avoir lieu dans plusieurs pays. Cette méthode représentative propose des liens entre les données claires et précises.
Pour la conception MPD, dans cette association, la table fréquence a pour rôle intermédiaire. Les clés étrangères country_code et disaster_code sont présentes dans la table fréquence, pour lier l'attribut "number" ainsi que les tables country et disaster. Dans cette conception, les cardinalités ne figurent pas comme dans le MCD, néanmoins grâce aux clés primaires et étrangères, la représentation est faite via les contraintes d'intégrité référentielle.Dans une
base de données relationnelle le MPD se regroupe sur la réalisation technique en structurant les relations pour qu'elles soient directement exploitables.

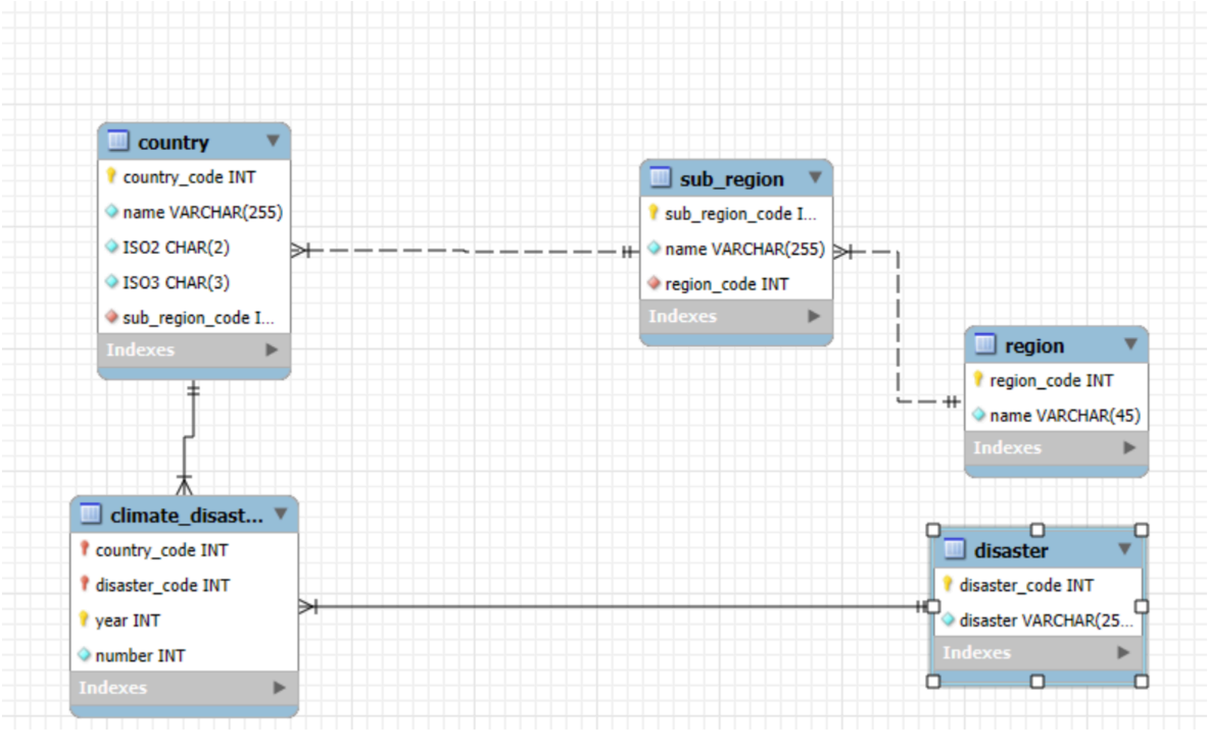
La représentation et organisation des données de façon logique et le lien commun entre les deux représentations . Néanmoins la logique est différente dans les modelé

2.2.3 *MDP réalisé avec l'AGL :*

Après la création des tableaux dans "MySQL Workbench", on peut afficher le Modèle Physique de Données. Il suffit d'appuyer sur l'élément indiqué dans la photo ci-dessous :

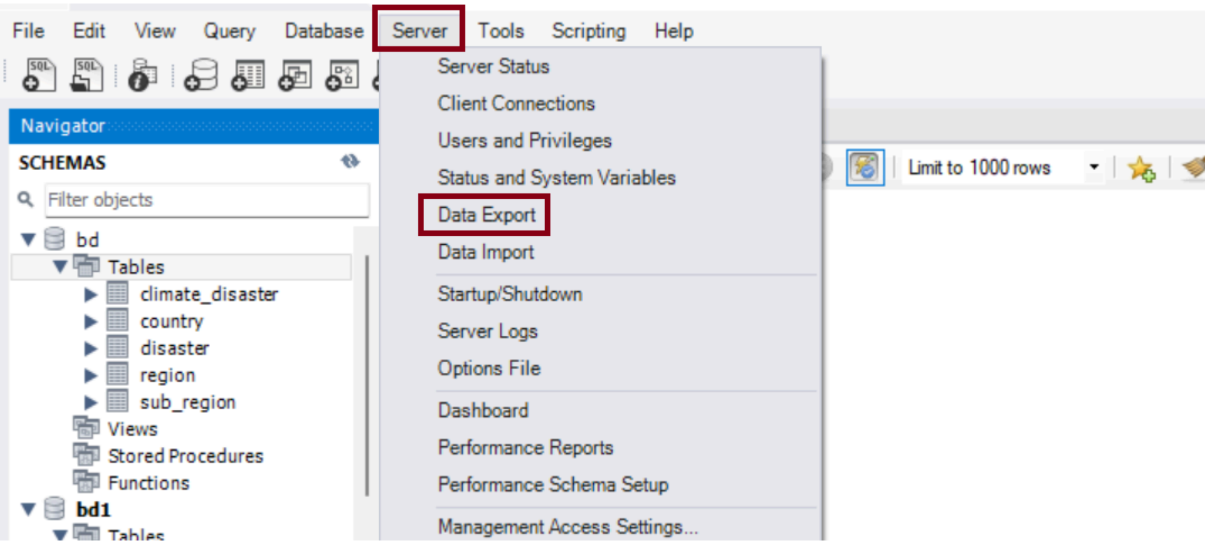


En obtenant les résultats suivants :

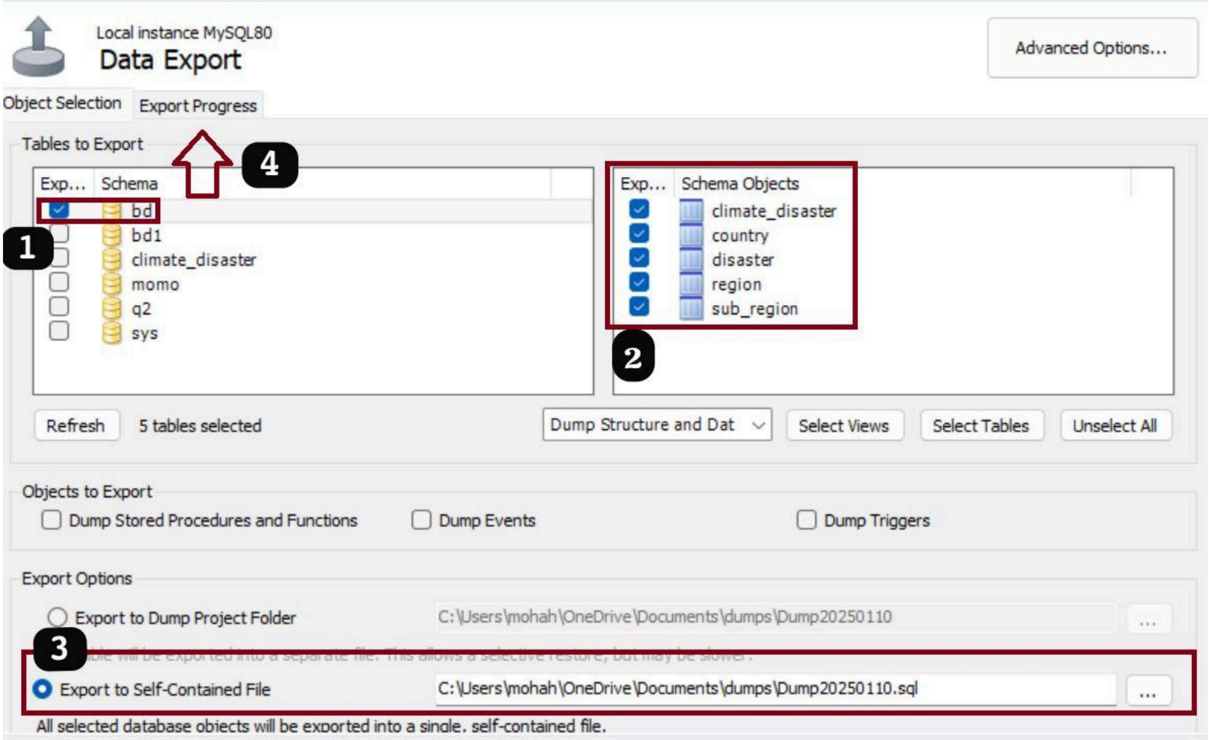


2.4. Script SQL de création des tables généré automatiquement par l'AGL

pour générer le script automatiquement, il suffit de suivre les étapes ci-dessous :



1) On sélectionne la base de données que l'on souhaite exporter sous forme de script.
2) On vérifie que tous les tableaux sont correctement sélectionnés.
3) On exporte le script dans un fichier unique et on choisit le chemin où sera placé ce fichier (fichier sql).
4) On procède à l'exportation du script.



Tables to Export

Exp...	Schema
☑	bd
□	bd1
□	climate_disaster
□	momo
□	q2
□	sys

Exp...	Schema Objects
☑	climate_disaster
☑	country
☑	disaster
☑	region
☑	sub_region

5 tables selected

Dump Structure and Dat

Objects to Export

□ Dump Stored Procedures and Functions □ Dump Events □ Dump Triggers

Export Options

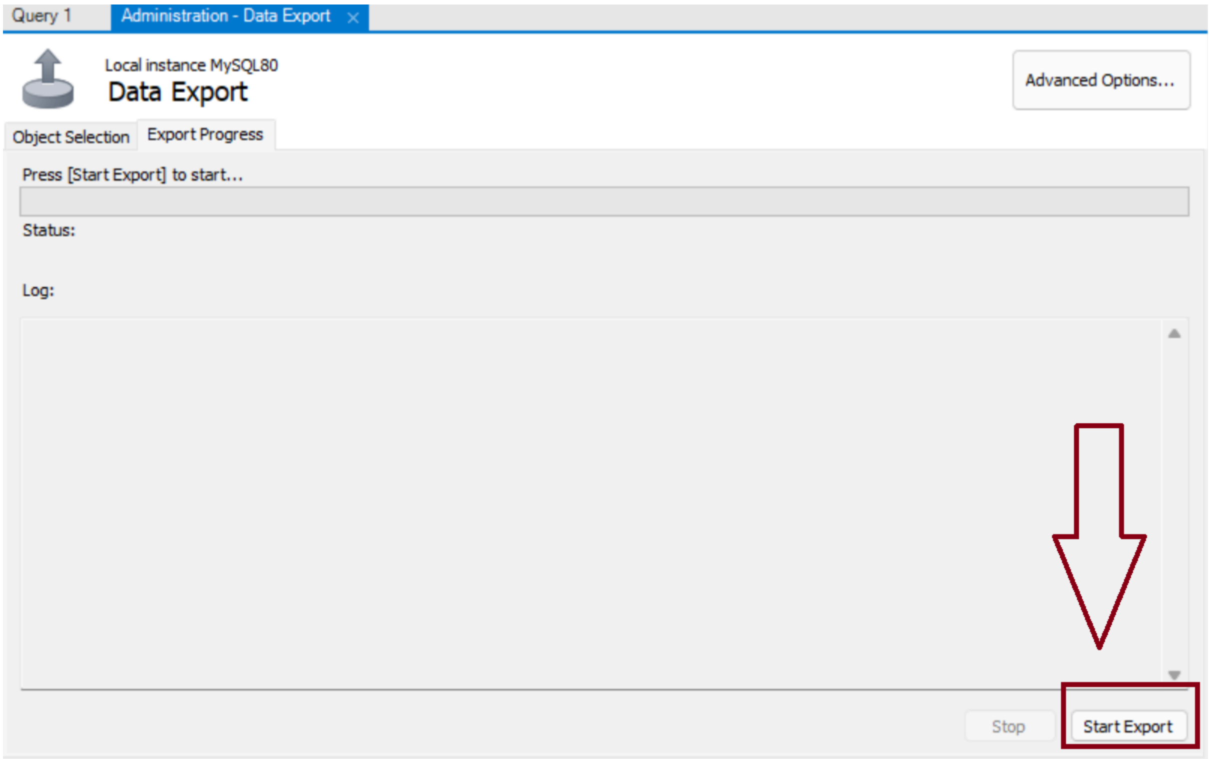
○ Export to Dump Project Folder C:\Users\mohah\OneDrive\Documents\dumps\Dump20250110 ...

able will be exported into a separate file. This allows a selective restore, but may be slower.

● Export to Self-Contained File C:\Users\mohah\OneDrive\Documents\dumps\Dump20250110.sql ...

All selected database objects will be exported into a single, self-contained file.

5) On commence notre exportation :



Press [Start Export] to start...

Status:

Log:

Le résultat obtenu :

```
-- MySQL dump 10.13  Distrib 8.0.40, for Win64 (x86_64)
--
-- Host: localhost    Database: bd
-- ------------------------------------------------------
-- Server version 8.0.40

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!50503 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
```

```
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO'
*/;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `climate_disaster`
--

DROP TABLE IF EXISTS `climate_disaster`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `climate_disaster` (
  `country_code` int NOT NULL,
  `disaster_code` int NOT NULL,
  `year` int NOT NULL,
  `number` int NOT NULL,
  PRIMARY KEY (`country_code`,`disaster_code`,`year`),
  KEY `disaster_code_idx` (`disaster_code`),
  CONSTRAINT `country_code` FOREIGN KEY (`country_code`) REFERENCES
`country` (`country_code`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `disaster_code` FOREIGN KEY (`disaster_code`) REFERENCES
`disaster` (`disaster_code`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `climate_disaster`
--

LOCK TABLES `climate_disaster` WRITE;
/*!40000 ALTER TABLE `climate_disaster` DISABLE KEYS */;
/*!40000 ALTER TABLE `climate_disaster` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `country`
--

DROP TABLE IF EXISTS `country`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
```

```
CREATE TABLE `country` (
  `country_code` int NOT NULL,
  `name` varchar(255) NOT NULL,
  `ISO2` char(2) NOT NULL,
  `ISO3` char(3) NOT NULL,
  `sub_region_code` int NOT NULL,
  PRIMARY KEY (`country_code`),
  UNIQUE KEY `ISO2_UNIQUE` (`ISO2`),
  UNIQUE KEY `ISO3_UNIQUE` (`ISO3`),
  KEY `sub_region_code_idx` (`sub_region_code`),
  CONSTRAINT `sub_region_code` FOREIGN KEY (`sub_region_code`)
REFERENCES `sub_region` (`sub_region_code`) ON DELETE CASCADE ON UPDATE
CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `country`
--

LOCK TABLES `country` WRITE;
/*!40000 ALTER TABLE `country` DISABLE KEYS */;
/*!40000 ALTER TABLE `country` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `disaster`
--

DROP TABLE IF EXISTS `disaster`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `disaster` (
  `disaster_code` int NOT NULL,
  `disaster` varchar(255) NOT NULL,
  PRIMARY KEY (`disaster_code`),
  UNIQUE KEY `disaster_UNIQUE` (`disaster`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `disaster`
--
```

```
LOCK TABLES `disaster` WRITE;
/*!40000 ALTER TABLE `disaster` DISABLE KEYS */;
/*!40000 ALTER TABLE `disaster` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `region`
--

DROP TABLE IF EXISTS `region`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `region` (
  `region_code` int NOT NULL,
  `name` varchar(45) NOT NULL,
  PRIMARY KEY (`region_code`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `region`
--

LOCK TABLES `region` WRITE;
/*!40000 ALTER TABLE `region` DISABLE KEYS */;
/*!40000 ALTER TABLE `region` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `sub_region`
--

DROP TABLE IF EXISTS `sub_region`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `sub_region` (
  `sub_region_code` int NOT NULL,
  `name` varchar(255) NOT NULL,
  `region_code` int NOT NULL,
  PRIMARY KEY (`sub_region_code`),
  KEY `region_code_idx` (`region_code`),
  CONSTRAINT `region_code` FOREIGN KEY (`region_code`) REFERENCES
`region` (`region_code`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `sub_region`
--

LOCK TABLES `sub_region` WRITE;
/*!40000 ALTER TABLE `sub_region` DISABLE KEYS */;
/*!40000 ALTER TABLE `sub_region` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2024-12-31 22:08:28
```

2.2.5. discussion sur les différences entre les scripts produits manuellement et automatiquement :

La création manuelle et automatique des scripts montre des particularités tant sur la forme que sur le fond. Les deux scripts répondent aux objectifs demandés et les deux mènent à des résultats que nous attendons, le script manuel est moins détaillé et moins précis que le script généré. Ce dernier utilise une technique qui vise la précision extrême

Pour ce qui est du caractère mobilité et facilité de lecture, le script manuel paraît en général plus universel et clair, alors que le script produit par l'AGL peut introduire des instructions plus détaillées (des fois non nécessaires) comme nous l'avons déjà mentionné, ainsi que des expressions et des commandes spécifiques à MySQL, ce qui ne permet pas de l'utiliser dans un autre système (PostgreSQL).

De cela, une conclusion idéale donne ce qui suit : le script manuel donne accès à la facilité de compréhension et à l'adaptabilité, quant au script automatique, utilise une technique avec exactitude et minutie afin d'une configuration rapide et, au niveau technique, soit précis.

2.3 . Script de peuplement de la base de données avec la description commentée des différentes étapes de script peuplement

j'ai choisi de travailler avec la figure 1 et le parcours simplifié, mais je vais tenter de réussir le chemin le plus difficile après .

Avant de commencer, nous devons analyser le fichier CSV. Nous constatons qu'aucun des 5 tableaux créés dans la partie 2.1 ne peut regrouper toutes les informations du fichier. Il est nécessaire de créer un tableau temporaire dans lequel toutes les données du fichier seront insérées. Cette étape (insertion) est faite par l'usage de la commande INSERT INTO. On remarque aussi que le fichier ne contient pas certaines entités comme country_code et disaster_code dans les tableaux country et disaster, ce qui risque d'interrompre l'insertion des données. Pour résoudre ce problème, nous devons modifier ou recréer les tableaux (mon cas) pour que le type des entités manquantes soit défini comme SERIAL au lieu de INTEGER .

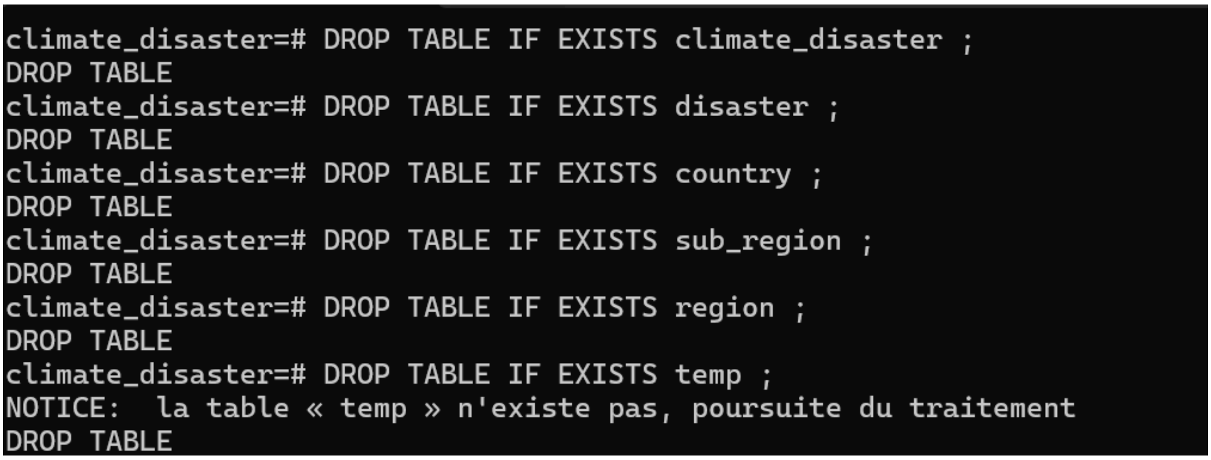
---

1) On commence par supprimer les anciens tableaux et créer les nouveaux tableaux (avec le type SERIAL) :

```
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 ;



 CREATE TABLE region ( region_code INTEGER PRIMARY KEY , name
VARCHAR NOT NULL);


 CREATE TABLE sub_region ( name VARCHAR NOT NULL , region_code
INTEGER NOT NULL REFERENCES region (region_code) ON DELETE
CASCADE, sub_region_code INTEGER

 PRIMARY KEY);


 CREATE TABLE country ( name VARCHAR NOT NULL , ISO2 CHAR(2)
UNIQUE , ISO3 CHAR(3)  UNIQUE , sub_region_code INTEGER NOT NULL
REFERENCES sub_region(sub_region_code) ON DELETE CASCADE,
country_code SERIAL PRIMARY KEY);


 CREATE TABLE disaster (disaster_code  SERIAL PRIMARY KEY ,
disaster VARCHAR NOT NULL UNIQUE );


CREATE TABLE climate_disaster (country_code INTEGER NOT NULL
REFERENCES country (country_code) ON DELETE CASCADE,
disaster_code INTEGER NOT NULL REFERENCES disaster(
disaster_code) ON DELETE CASCADE,  year INTEGER NOT NULL
CHECK(year>0) , number INTEGER CHECK(number>0) , PRIMARY
KEY(country_code,disaster_code,year));
```

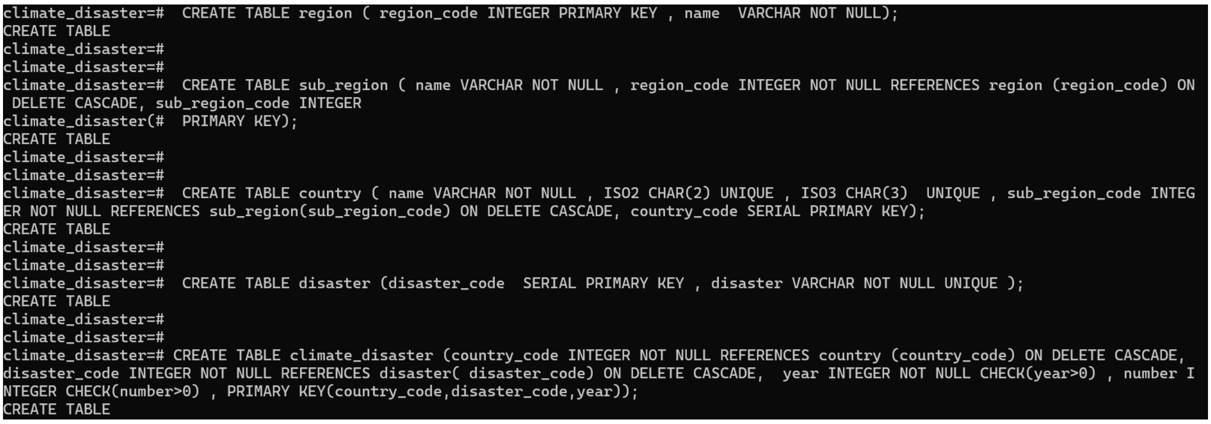
-Résultats des commandes au dessus :

1.1) la suppression des anciens tableaux :



```
climate_disaster=# DROP TABLE IF EXISTS climate_disaster ;
DROP TABLE
climate_disaster=# DROP TABLE IF EXISTS disaster ;
DROP TABLE
climate_disaster=# DROP TABLE IF EXISTS country ;
DROP TABLE
climate_disaster=# DROP TABLE IF EXISTS sub_region ;
DROP TABLE
climate_disaster=# DROP TABLE IF EXISTS region ;
DROP TABLE
climate_disaster=# DROP TABLE IF EXISTS temp ;
NOTICE:  la table « temp » n'existe pas, poursuite du traitement
DROP TABLE
```

1.2)la création des nouveaux tableaux :

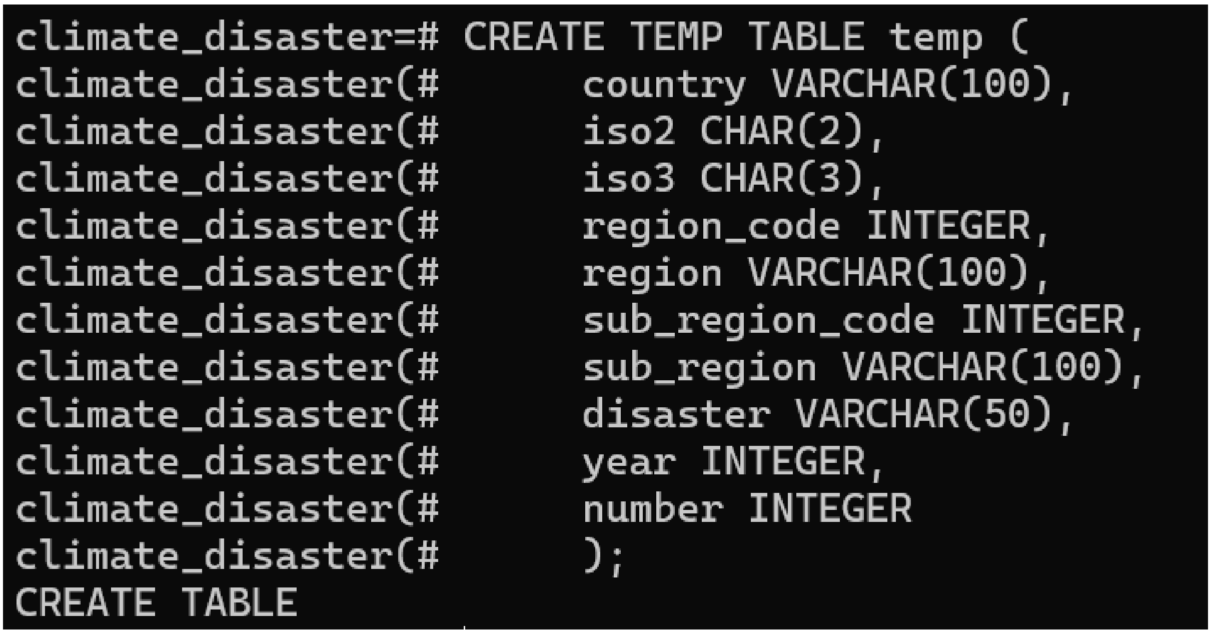


```
climate_disaster=#  CREATE TABLE region ( region_code INTEGER PRIMARY KEY , name  VARCHAR NOT NULL);
CREATE TABLE
climate_disaster=#
climate_disaster=#
climate_disaster=#  CREATE TABLE sub_region ( name VARCHAR NOT NULL , region_code INTEGER NOT NULL REFERENCES region (region_code) ON
 DELETE CASCADE, sub_region_code INTEGER
climate_disaster(#  PRIMARY KEY);
CREATE TABLE
climate_disaster=#
climate_disaster=#
climate_disaster=#  CREATE TABLE country ( name VARCHAR NOT NULL , ISO2 CHAR(2) UNIQUE , ISO3 CHAR(3)  UNIQUE , sub_region_code INTEG
ER NOT NULL REFERENCES sub_region(sub_region_code) ON DELETE CASCADE, country_code SERIAL PRIMARY KEY);
CREATE TABLE
climate_disaster=#
climate_disaster=#
climate_disaster=#  CREATE TABLE disaster (disaster_code  SERIAL PRIMARY KEY , disaster VARCHAR NOT NULL UNIQUE );
CREATE TABLE
climate_disaster=#
climate_disaster=#
climate_disaster=# CREATE TABLE climate_disaster (country_code INTEGER NOT NULL REFERENCES country (country_code) ON DELETE CASCADE,
disaster_code INTEGER NOT NULL REFERENCES disaster( disaster_code) ON DELETE CASCADE,  year INTEGER NOT NULL CHECK(year>0) , number I
NTEGER CHECK(number>0) , PRIMARY KEY(country_code,disaster_code,year));
CREATE TABLE
```

2) On crée la table temporaire par la commande suivante : (on choisit une table de type temporaire car elle est utilisée pour stocker des données temporaires qui ne sont pas conservées de manière permanente. Après la fin de la session le tableau serait supprimé automatiquement)

```
CREATE TEMP TABLE temp (country VARCHAR(100),iso2 CHAR(2),iso3
CHAR(3),region_code INTEGER,region VARCHAR(100),sub_region_code
INTEGER,sub_region VARCHAR(100),disaster VARCHAR(50),year
INTEGER,number INTEGER);
```

-Résultats des commandes au dessus :



```
climate_disaster=# CREATE TEMP TABLE temp (
climate_disaster(#      country VARCHAR(100),
climate_disaster(#      iso2 CHAR(2),
climate_disaster(#      iso3 CHAR(3),
climate_disaster(#      region_code INTEGER,
climate_disaster(#      region VARCHAR(100),
climate_disaster(#      sub_region_code INTEGER,
climate_disaster(#      sub_region VARCHAR(100),
climate_disaster(#      disaster VARCHAR(50),
climate_disaster(#      year INTEGER,
climate_disaster(#      number INTEGER
climate_disaster(#      );
CREATE TABLE
```

3) on copie le fichier dans le tableau temporaire, la commande utilisée est la suivante :

```
\copy temp FROM
C:/Users/mohah/OneDrive/Bureau/Climate_related_disasters_frequency.csv
DELIMITER ',' CSV HEADER;
```

si vous voulez tester cette commande, il est préférable de ne pas utiliser la version de la commande au-dessus avec la police plus grande, mais plutôt celle (au-dessous) avec la police réduite. Cela permet d'éviter des erreurs de syntaxe qui peuvent survenir à cause des espaces .

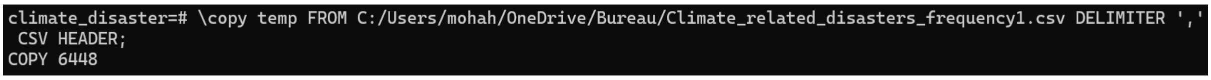
La version utilisable :

```
\copy temp FROM C:/Users/mohah/OneDrive/Bureau/Climate_related_disasters_frequency.csv DELIMITER ',' CSV HEADER;
```

Explication de la commande :

1) \copy temp from → la commande qui permette de copier le fichier
2) on ajoute notre chemin vers le fichier csv .dans mon cas c'est : *"C:/Users/mohah/OneDrive/Bureau/Climate_related_disasters_frequency.csv "*
3) `DELIMITER ','` → cela veut dire que les valeurs dans les lignes du fichier sont séparées par une virgule
4) `CSV HEADER;` → cela veut dire que la première ligne comporte les noms des colonnes

<u>-Résultats des commandes au dessus :</u>

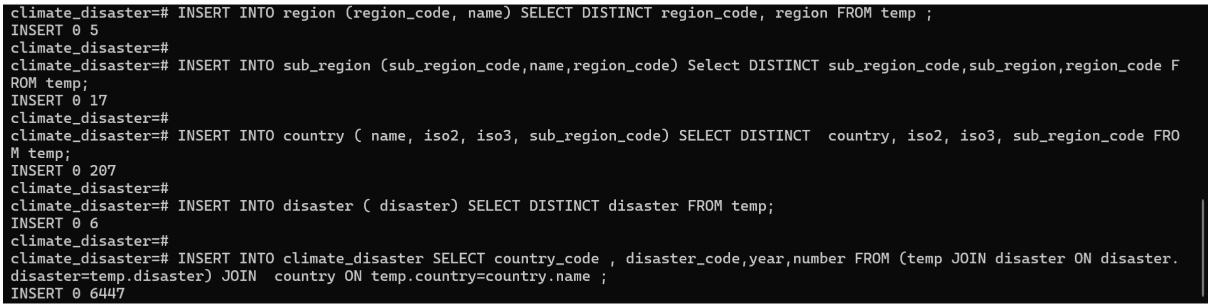


```
climate_disaster=# \copy temp FROM C:/Users/mohah/OneDrive/Bureau/Climate_related_disasters_frequency1.csv DELIMITER ','
 CSV HEADER;
COPY 6448
```

4) on insère les données dans les tableaux (country, region, sub-region, disaster, climate_disaster) par les commandes suivantes :

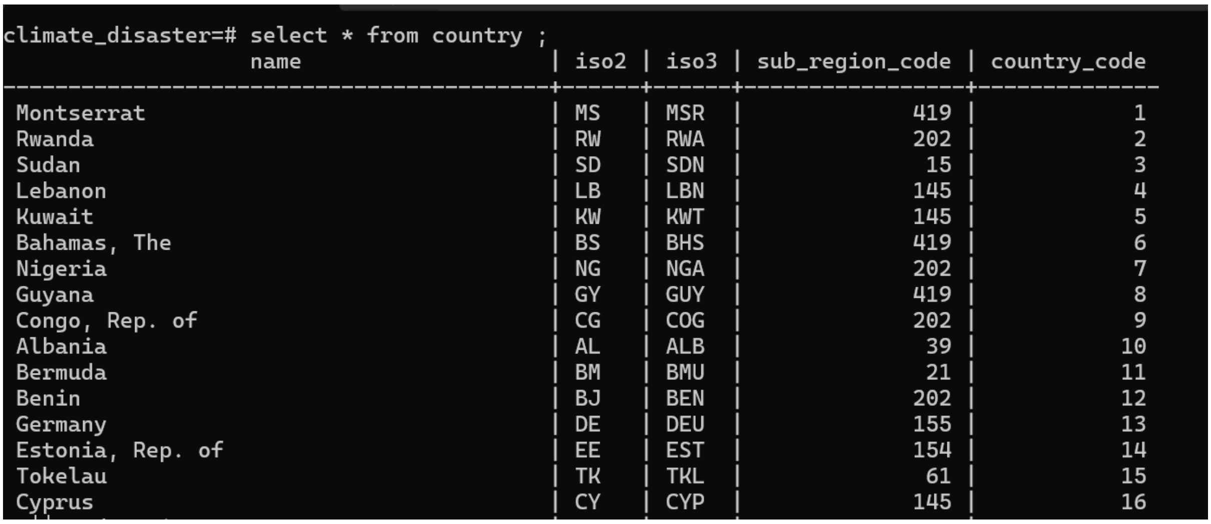
```
INSERT INTO region (region_code, name) SELECT DISTINCT region_code,
region FROM temp ;

INSERT INTO sub_region (sub_region_code,name,region_code) Select
DISTINCT sub_region_code,sub_region,region_code FROM temp;

INSERT INTO country ( name, iso2, iso3, sub_region_code) SELECT
DISTINCT  country, iso2, iso3, sub_region_code FROM temp ;

INSERT INTO disaster ( disaster) SELECT DISTINCT disaster FROM temp;

INSERT INTO climate_disaster SELECT country_code ,
disaster_code,year,number FROM (temp JOIN disaster ON
disaster.disaster=temp.disaster) JOIN  country ON
temp.country=country.name ;
```

-Résultats des commandes au dessus :



```
climate_disaster=# INSERT INTO region (region_code, name) SELECT DISTINCT region_code, region FROM temp ;
INSERT 0 5
climate_disaster=#
climate_disaster=# INSERT INTO sub_region (sub_region_code,name,region_code) Select DISTINCT sub_region_code,sub_region,region_code F
ROM temp;
INSERT 0 17
climate_disaster=#
climate_disaster=# INSERT INTO country ( name, iso2, iso3, sub_region_code) SELECT DISTINCT  country, iso2, iso3, sub_region_code FRO
M temp;
INSERT 0 207
climate_disaster=#
climate_disaster=# INSERT INTO disaster ( disaster) SELECT DISTINCT disaster FROM temp;
INSERT 0 6
climate_disaster=#
climate_disaster=# INSERT INTO climate_disaster SELECT country_code , disaster_code,year,number FROM (temp JOIN disaster ON disaster.
disaster=temp.disaster) JOIN  country ON temp.country=country.name ;
INSERT 0 6447
```

Exemple d'un tableau après l'insertion des données :



```
climate_disaster=# select * from country ;
                   name                   | iso2 | iso3 | sub_region_code | country_code
------------------------------------------+------+------+-----------------+--------------
 Montserrat                               | MS   | MSR  |             419 |            1
 Rwanda                                   | RW   | RWA  |             202 |            2
 Sudan                                    | SD   | SDN  |              15 |            3
 Lebanon                                  | LB   | LBN  |             145 |            4
 Kuwait                                   | KW   | KWT  |             145 |            5
 Bahamas, The                             | BS   | BHS  |             419 |            6
 Nigeria                                  | NG   | NGA  |             202 |            7
 Guyana                                   | GY   | GUY  |             419 |            8
 Congo, Rep. of                           | CG   | COG  |             202 |            9
 Albania                                  | AL   | ALB  |              39 |           10
 Bermuda                                  | BM   | BMU  |              21 |           11
 Benin                                    | BJ   | BEN  |             202 |           12
 Germany                                  | DE   | DEU  |             155 |           13
 Estonia, Rep. of                         | EE   | EST  |             154 |           14
 Tokelau                                  | TK   | TKL  |              61 |           15
 Cyprus                                   | CY   | CYP  |             145 |           16
```

5) On supprime le tableau temporaire (cette étape est facultative, car on sait bien qu'à la fin de la session, le tableau temporaire sera supprimé automatiquement) . la commande utilisée est :

   ```
   Drop table temp ;
   ```

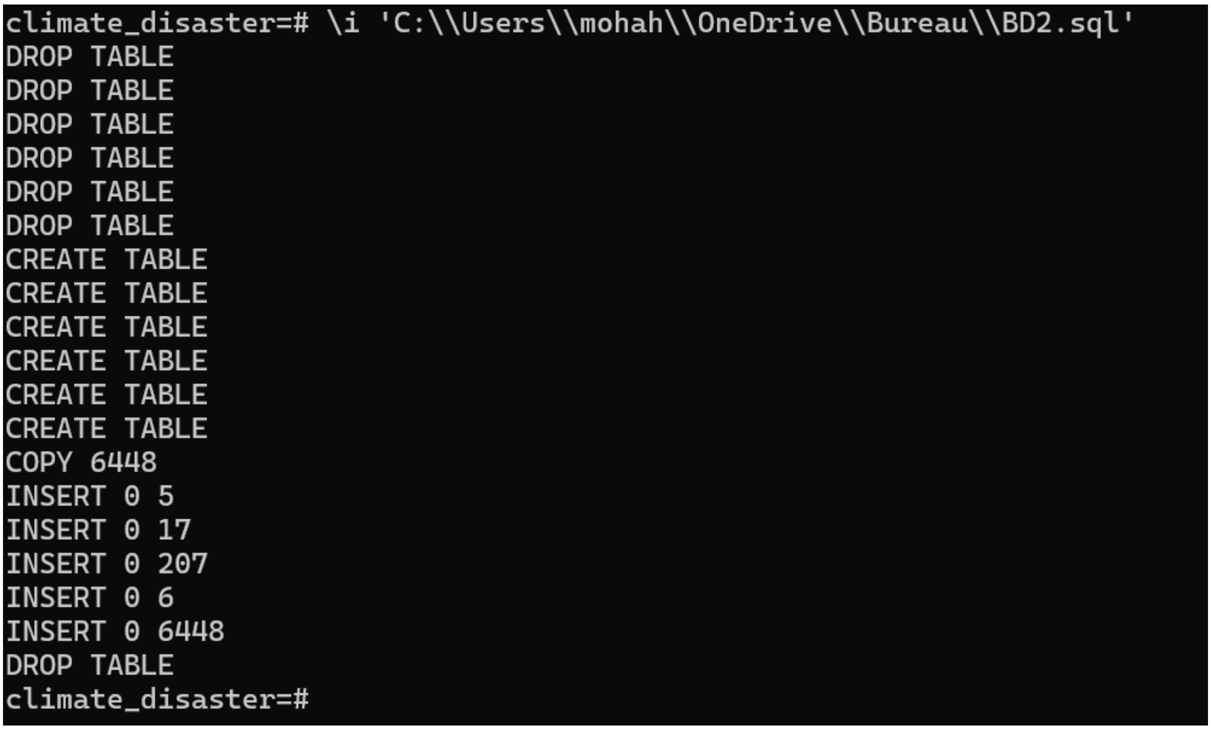
Le script final de peuplement :

```
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 ;

CREATE TABLE region ( region_code INTEGER PRIMARY KEY , name  VARCHAR
NOT NULL);

CREATE TABLE sub_region ( name VARCHAR NOT NULL , region_code INTEGER
NOT NULL REFERENCES region (region_code) ON DELETE CASCADE,
sub_region_code INTEGER PRIMARY KEY);

CREATE TABLE country ( name VARCHAR NOT NULL , ISO2 CHAR(2) UNIQUE ,
ISO3 CHAR(3)  UNIQUE , sub_region_code INTEGER NOT NULL REFERENCES
sub_region(sub_region_code) ON DELETE CASCADE, country_code SERIAL
PRIMARY KEY);

CREATE TABLE disaster (disaster_code  SERIAL PRIMARY KEY , disaster
VARCHAR NOT NULL UNIQUE );
```

```sql
CREATE TABLE climate_disaster (country_code INTEGER NOT NULL REFERENCES
country (country_code) ON DELETE CASCADE, disaster_code INTEGER NOT
NULL REFERENCES disaster( disaster_code) ON DELETE CASCADE,  year
INTEGER NOT NULL CHECK(year>0) , number INTEGER CHECK(number>0) ,
PRIMARY KEY(country_code,disaster_code,year));

CREATE TEMP TABLE temp (country VARCHAR(100),iso2 CHAR(2),iso3
CHAR(3),region_code INTEGER,region VARCHAR(100),sub_region_code
INTEGER,sub_region VARCHAR(100),disaster VARCHAR(50),year
INTEGER,number INTEGER);

\copy temp FROM C:/Users/mohah/OneDrive/Bureau/Climate_related_disasters_frequency1.csv DELIMITER ',' CSV HEADER;
```

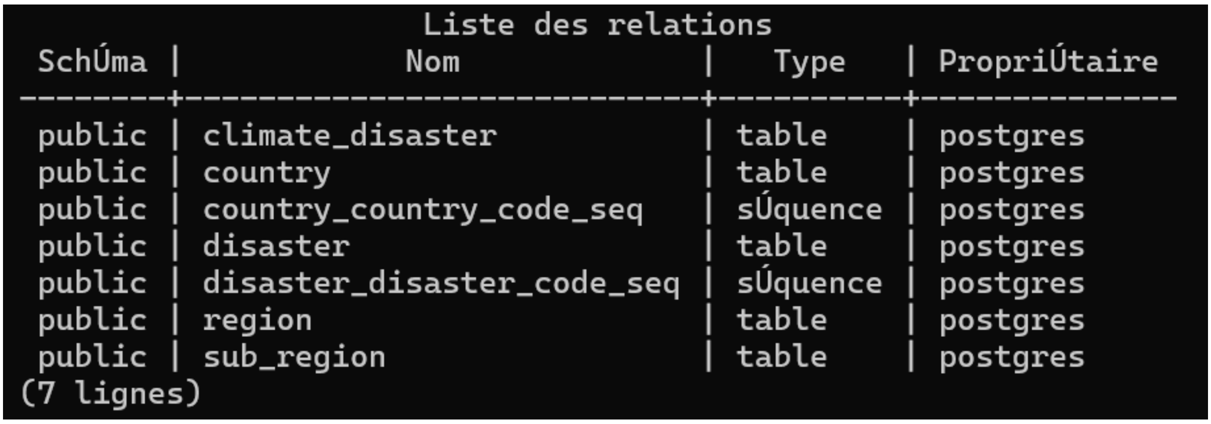
```sql
INSERT INTO region (region_code, name) SELECT DISTINCT region_code,
region FROM temp ;

INSERT INTO sub_region (sub_region_code,name,region_code) Select
DISTINCT sub_region_code,sub_region,region_code FROM temp ;

INSERT INTO country ( name, iso2, iso3, sub_region_code) SELECT
DISTINCT  country, iso2, iso3, sub_region_code FROM temp ;

INSERT INTO disaster ( disaster) SELECT DISTINCT disaster FROM temp ;

INSERT INTO climate_disaster SELECT country_code ,
disaster_code,year,number FROM (temp JOIN disaster ON
disaster.disaster=temp.disaster) JOIN  country ON
temp.country=country.name  ;

Drop table temp ;
```

Affichage de résultat du script par la commande “ \i “ :



```
climate_disaster=# \i 'C:\\Users\\mohah\\OneDrive\\Bureau\\BD2.sql'
DROP TABLE
DROP TABLE
DROP TABLE
DROP TABLE
DROP TABLE
DROP TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
COPY 6448
INSERT 0 5
INSERT 0 17
INSERT 0 207
INSERT 0 6
INSERT 0 6448
DROP TABLE
climate_disaster=#
```

Affichage des tableaux par la commande “ \d ; “ :



```
                      Liste des relations
 SchÚma |             Nom              |   Type   | PropriÚtaire
--------+------------------------------+----------+--------------
 public | climate_disaster             | table    | postgres
 public | country                      | table    | postgres
 public | country_country_code_seq     | sÚquence | postgres
 public | disaster                     | table    | postgres
 public | disaster_disaster_code_seq   | sÚquence | postgres
 public | region                       | table    | postgres
 public | sub_region                   | table    | postgres
(7 lignes)
```

Tentative du peuplement du second jeu

(Le script final est à la fin du rapport)

1)On va commencer par créer les deux tableaux temporaires comme on a déjà fait dans le peuplement du premier jeu, ainsi on copie les fichiers csv dans ces tableaux. Cette étape nécessite les commandes suivantes :

Création du tableau de figure 2 :

```
CREATE TEMPORARY TABLE temp_disasters (ObjectId SERIAL PRIMARY
KEY,Country VARCHAR,ISO2 CHAR(2),ISO3 CHAR(3),Indicator VARCHAR,Unit
VARCHAR,"1980" INT,"1981" INT,"1982" INT,"1983" INT,"1984" INT,"1985"
INT,"1986" INT,"1987" INT,"1988" INT,"1989" INT,"1990" INT,"1991"
INT,"1992" INT,"1993" INT,"1994" INT,"1995" INT,"1996" INT,"1997"
INT,"1998" INT,"1999" INT,"2000" INT,"2001" INT,"2002" INT,"2003"
INT,"2004" INT,"2005" INT,"2006" INT,"2007" INT,"2008" INT,"2009"
INT,"2010" INT,"2011" INT,"2012" INT,"2013" INT,"2014" INT,"2015"
INT,"2016" INT,"2017" INT,"2018" INT,"2019" INT,"2020" INT,"2021"
INT,"2022" INT,"2023" INT,Source VARCHAR);
```

Insertion des données dans le tableau temporaire :

```
\COPY  temp_disasters FROM C:\Users\mohah\OneDrive\Bureau\Figure_2.csv
DELIMITER ',' CSV HEADER;
```

Création du tableau de figure 3 :

```
CREATE TEMPORARY TABLE temp_country_region (name VARCHAR,alpha_2
CHAR(2),alpha_3 CHAR(3),country_code  VARCHAR,iso_3166_2 VARCHAR,region
VARCHAR,sub_region VARCHAR,intermediate_region VARCHAR,region_code
VARCHAR ,sub_region_code  VARCHAR ,intermediate_region_code VARCHAR);
```

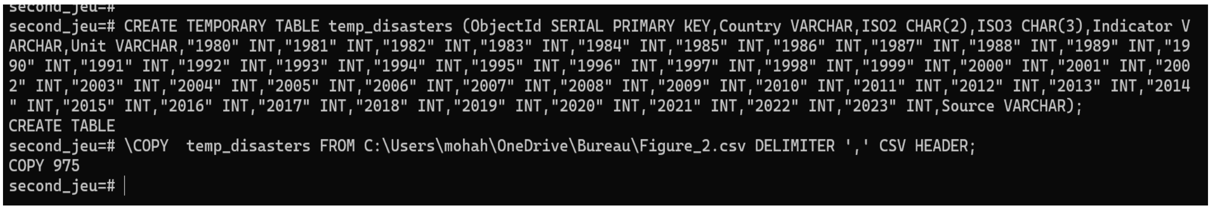
J'ai utilisé le type VARCHAR partout pour éviter les problèmes associés aux colonnes lors de l'insertion des données dans la table temporaire. Par exemple, certaines colonnes qui sont censées contenir uniquement des valeurs entières comme sub_region_code peuvent contenir des valeurs vides dans les données importées. En utilisant VARCHAR, le problème sera résolu et on garantit une importation des données sans aucun problème et difficulté, et par la suite je vais modifier le type de ces colonnes pour les rendre à leur type convenable

Insertion des données dans le tableau temporaire :

```
\COPY   temp_country_region FROM C:\Users\mohah\OneDrive\Bureau\Figure_3.csv  DELIMITER ','
CSV HEADER;
```

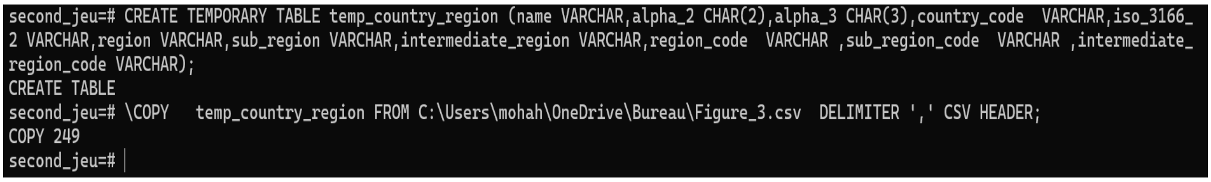
<u>-Résultats des commandes au dessus :</u>

<u>figure 2 :</u>



```
second_jeu=#
second_jeu=# CREATE TEMPORARY TABLE temp_disasters (ObjectId SERIAL PRIMARY KEY,Country VARCHAR,ISO2 CHAR(2),ISO3 CHAR(3),Indicator V
ARCHAR,Unit VARCHAR,"1980" INT,"1981" INT,"1982" INT,"1983" INT,"1984" INT,"1985" INT,"1986" INT,"1987" INT,"1988" INT,"1989" INT,"19
90" INT,"1991" INT,"1992" INT,"1993" INT,"1994" INT,"1995" INT,"1996" INT,"1997" INT,"1998" INT,"1999" INT,"2000" INT,"2001" INT,"200
2" INT,"2003" INT,"2004" INT,"2005" INT,"2006" INT,"2007" INT,"2008" INT,"2009" INT,"2010" INT,"2011" INT,"2012" INT,"2013" INT,"2014
" INT,"2015" INT,"2016" INT,"2017" INT,"2018" INT,"2019" INT,"2020" INT,"2021" INT,"2022" INT,"2023" INT,Source VARCHAR);
CREATE TABLE
second_jeu=# \COPY  temp_disasters FROM C:\Users\mohah\OneDrive\Bureau\Figure_2.csv DELIMITER ',' CSV HEADER;
COPY 975
second_jeu=#
```

<u>figure 3 :</u>



```
second_jeu=# CREATE TEMPORARY TABLE temp_country_region (name VARCHAR,alpha_2 CHAR(2),alpha_3 CHAR(3),country_code  VARCHAR,iso_3166_
2 VARCHAR,region VARCHAR,sub_region VARCHAR,intermediate_region VARCHAR,region_code  VARCHAR ,sub_region_code  VARCHAR ,intermediate_
region_code VARCHAR);
CREATE TABLE
second_jeu=# \COPY   temp_country_region FROM C:\Users\mohah\OneDrive\Bureau\Figure_3.csv  DELIMITER ',' CSV HEADER;
COPY 249
second_jeu=#
```

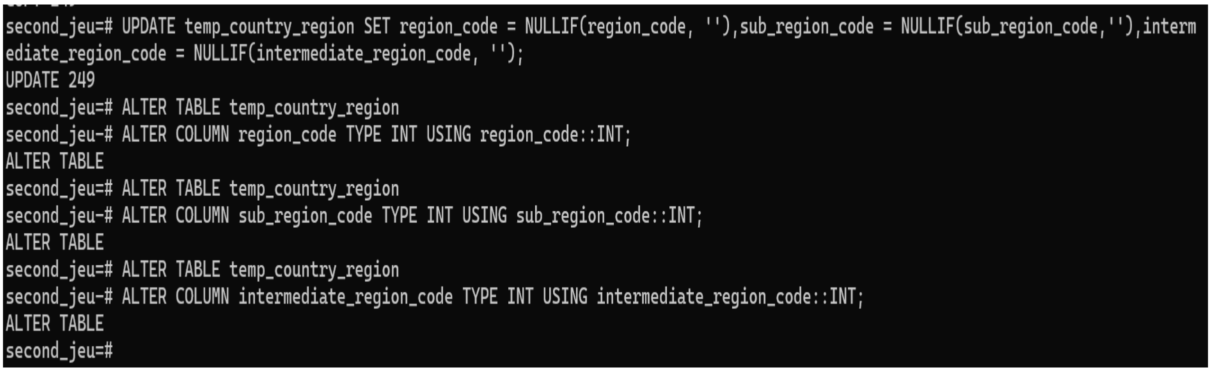
On modifie les valeurs vides (les valeurs qui sont exprimées par : '') dans le tableau temporaire (temp_country_region) par la valeur NULL à l'aide de la commande suivante :

```
UPDATE temp_country_region SET region_code = NULLIF(region_code,
''),sub_region_code =
NULLIF(sub_region_code,''),intermediate_region_code =
NULLIF(intermediate_region_code, '');
```

Comme je l'ai déjà mentionné, je vais ajuster le type des colonnes à celui qui est convenable (INT), la commande utilisée :

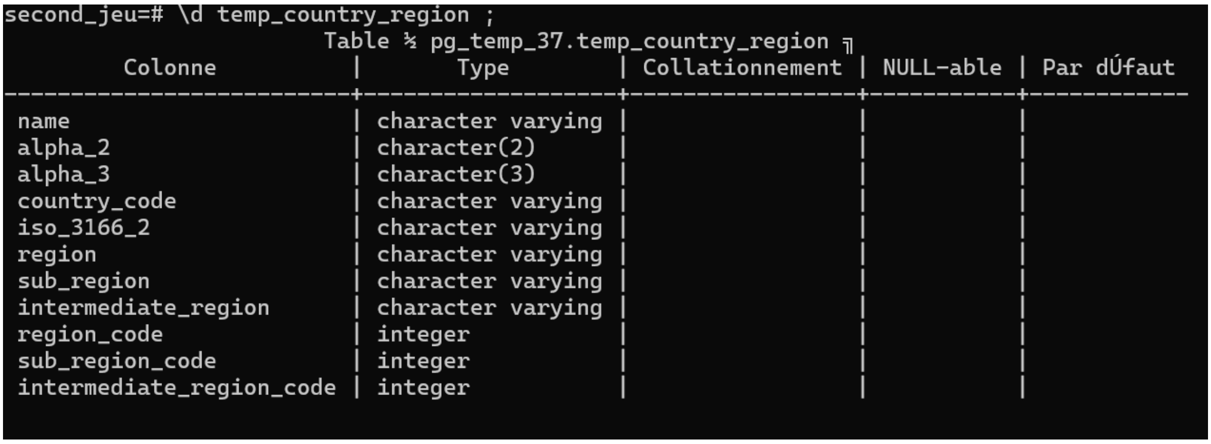
```
ALTER TABLE temp_country_region
ALTER COLUMN region_code TYPE INT USING region_code::INT;
ALTER TABLE temp_country_region
ALTER COLUMN sub_region_code TYPE INT USING sub_region_code::INT;
ALTER TABLE temp_country_region
ALTER COLUMN intermediate_region_code TYPE INT USING
intermediate_region_code::INT;
```

-Résultats des commandes au dessus :



```
second_jeu=# UPDATE temp_country_region SET region_code = NULLIF(region_code, ''),sub_region_code = NULLIF(sub_region_code,''),interm
ediate_region_code = NULLIF(intermediate_region_code, '');
UPDATE 249
second_jeu=# ALTER TABLE temp_country_region
second_jeu-# ALTER COLUMN region_code TYPE INT USING region_code::INT;
ALTER TABLE
second_jeu=# ALTER TABLE temp_country_region
second_jeu-# ALTER COLUMN sub_region_code TYPE INT USING sub_region_code::INT;
ALTER TABLE
second_jeu=# ALTER TABLE temp_country_region
second_jeu-# ALTER COLUMN intermediate_region_code TYPE INT USING intermediate_region_code::INT;
ALTER TABLE
second_jeu=#
```

Après modification des types :



```
second_jeu=# \d temp_country_region ;
                       Table ½ pg_temp_37.temp_country_region ╗
         Colonne          |        Type       | Collationnement | NULL-able | Par dÚfaut
--------------------------+-------------------+-----------------+-----------+------------
 name                     | character varying |                 |           |
 alpha_2                  | character(2)      |                 |           |
 alpha_3                  | character(3)      |                 |           |
 country_code             | character varying |                 |           |
 iso_3166_2               | character varying |                 |           |
 region                   | character varying |                 |           |
 sub_region               | character varying |                 |           |
 intermediate_region      | character varying |                 |           |
 region_code              | integer           |                 |           |
 sub_region_code          | integer           |                 |           |
 intermediate_region_code | integer           |                 |           |
```

2)Comme nous l'avons déjà fait dans le premier jeu (le chemin facile) de la figure 1, nous allons supprimer les tableaux   (region, sub_region, country, disaster, climate_disaster) et les recréer en utilisant le type serial pour certaines colonnes de ces tableaux.
les commandes utilisées :

```
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 ;


CREATE TABLE region ( region_code INTEGER PRIMARY KEY , name  VARCHAR
NOT NULL);

 CREATE TABLE sub_region ( name VARCHAR NOT NULL , region_code INTEGER
NOT NULL REFERENCES region (region_code) ON DELETE CASCADE,
sub_region_code INTEGER
 PRIMARY KEY);

 CREATE TABLE country ( name VARCHAR NOT NULL , ISO2 CHAR(2) UNIQUE ,
ISO3 CHAR(3)  UNIQUE , sub_region_code INTEGER REFERENCES
sub_region(sub_region_code) ON DELETE CASCADE, country_code SERIAL
PRIMARY KEY);

 CREATE TABLE disaster (disaster_code  SERIAL PRIMARY KEY , disaster
VARCHAR NOT NULL UNIQUE );

CREATE TABLE climate_disaster (country_code INTEGER NOT NULL REFERENCES
country (country_code) ON DELETE CASCADE, disaster_code INTEGER NOT
NULL REFERENCES disaster( disaster_code) ON DELETE CASCADE,  year
INTEGER NOT NULL CHECK(year>0) , number INTEGER CHECK(number>0) ,
PRIMARY KEY(country_code,disaster_code,year));
```

---

3) on insère les données dans les tableaux (country, region, sub-region, disaster, climate_disaster) par les commandes suivantes :

-Tableau Region :

```
INSERT INTO region (region_code, name) SELECT DISTINCT
region_code,region FROM temp_country_region WHERE region_code IS NOT
NULL AND region IS NOT NULL;
```

Nous avons utilisé la condition :

```
WHERE region_code IS NOT NULL AND region IS NOT NULL;
```

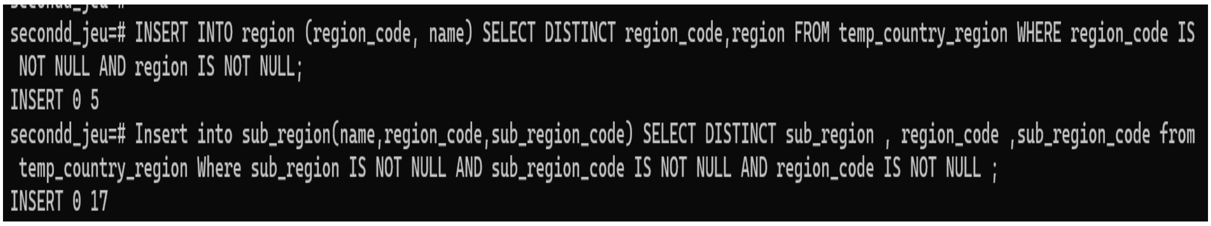
pour éviter les erreurs liées aux pays qui n'appartiennent à aucune région comme Antarctica .

-Tableau sub_region :

```
Insert into sub_region(name,region_code,sub_region_code) SELECT
DISTINCT sub_region , region_code ,sub_region_code from
temp_country_region Where sub_region IS NOT NULL AND sub_region_code IS
NOT NULL AND region_code IS NOT NULL ;
```

Nous avons appliqué `( Where sub_region IS NOT NULL AND sub_region_code IS NOT NULL AND region_code IS NOT NULL )` pour les mêmes raisons que dans le tableau region .

-Résultats des commandes au dessus :



```
secondd_jeu=#
secondd_jeu=# INSERT INTO region (region_code, name) SELECT DISTINCT region_code,region FROM temp_country_region WHERE region_code IS
 NOT NULL AND region IS NOT NULL;
INSERT 0 5
secondd_jeu=# Insert into sub_region(name,region_code,sub_region_code) SELECT DISTINCT sub_region , region_code ,sub_region_code from
 temp_country_region Where sub_region IS NOT NULL AND sub_region_code IS NOT NULL AND region_code IS NOT NULL ;
INSERT 0 17
```

-Tableau Country :

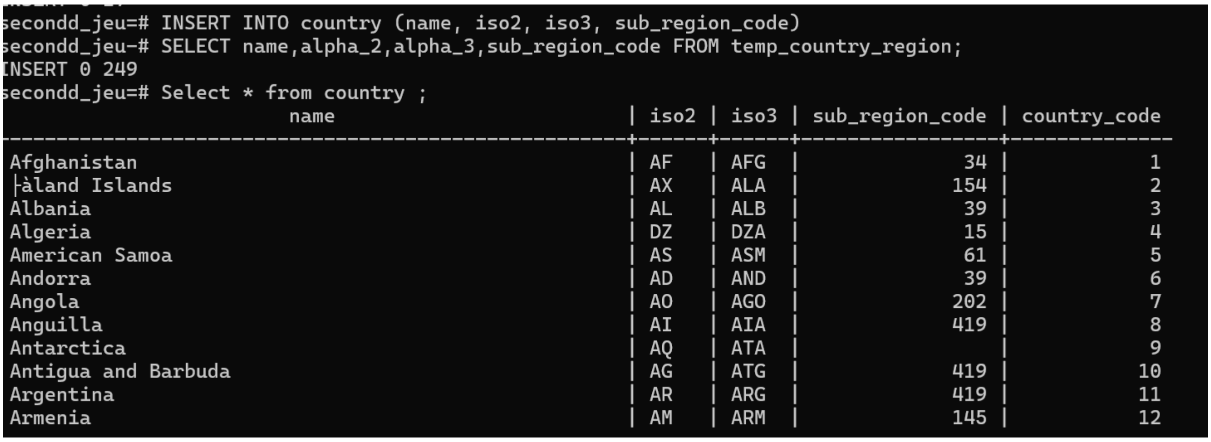
```
INSERT INTO country (name, iso2, iso3, sub_region_code)
SELECT name,alpha_2,alpha_3,sub_region_code FROM temp_country_region;
```

-Tableau Disaster:

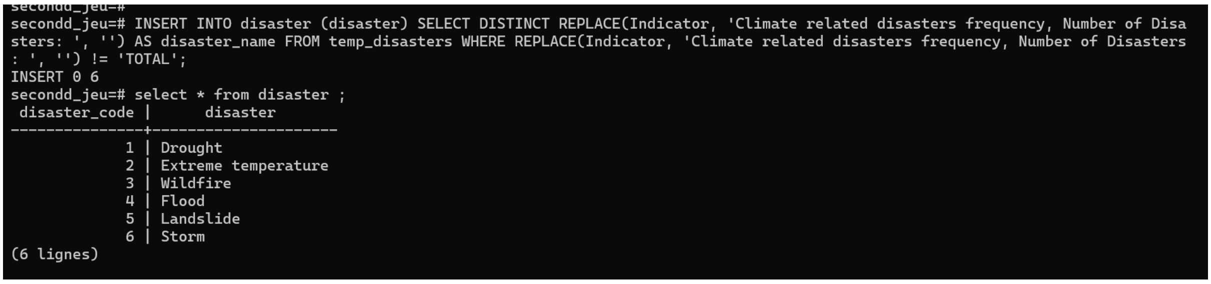
```
INSERT INTO disaster (disaster) SELECT DISTINCT REPLACE(Indicator,
'Climate related disasters frequency, Number of Disasters: ', '') AS
disaster_name FROM temp_disasters WHERE REPLACE(Indicator, 'Climate
related disasters frequency, Number of Disasters: ', '') != 'TOTAL';
```

Nous avons utilisé REPLACE afin de supprimer le prefixe (Climate related disasters frequency, Number of Disasters:) et obtenir uniquement le nom de catastrophe (disaster)

-Résultats des commandes au dessus :



```
secondd_jeu=# INSERT INTO country (name, iso2, iso3, sub_region_code)
secondd_jeu-# SELECT name,alpha_2,alpha_3,sub_region_code FROM temp_country_region;
INSERT 0 249
secondd_jeu=# Select * from country ;
                          name                           | iso2 | iso3 | sub_region_code | country_code
---------------------------------------------------------+------+------+-----------------+--------------
 Afghanistan                                             | AF   | AFG  |              34 |            1
 ├åland Islands                                          | AX   | ALA  |             154 |            2
 Albania                                                 | AL   | ALB  |              39 |            3
 Algeria                                                 | DZ   | DZA  |              15 |            4
 American Samoa                                          | AS   | ASM  |              61 |            5
 Andorra                                                 | AD   | AND  |              39 |            6
 Angola                                                  | AO   | AGO  |             202 |            7
 Anguilla                                                | AI   | AIA  |             419 |            8
 Antarctica                                              | AQ   | ATA  |                 |            9
 Antigua and Barbuda                                     | AG   | ATG  |             419 |           10
 Argentina                                               | AR   | ARG  |             419 |           11
 Armenia                                                 | AM   | ARM  |             145 |           12
```



```
secondd_jeu=#
secondd_jeu=# INSERT INTO disaster (disaster) SELECT DISTINCT REPLACE(Indicator, 'Climate related disasters frequency, Number of Disa
sters: ', '') AS disaster_name FROM temp_disasters WHERE REPLACE(Indicator, 'Climate related disasters frequency, Number of Disasters
: ', '') != 'TOTAL';
INSERT 0 6
secondd_jeu=# select * from disaster ;
 disaster_code |      disaster
---------------+---------------------
             1 | Drought
             2 | Extreme temperature
             3 | Wildfire
             4 | Flood
             5 | Landslide
             6 | Storm
(6 lignes)
```

Tableau climate_disaster :

```
INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1980 AS
year,temp_disasters."1980" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1980" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1981 AS
```

```
year,temp_disasters."1981" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1981" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1982 AS
year,temp_disasters."1982" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1982" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1983 AS
year,temp_disasters."1983" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1983" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1984 AS
year,temp_disasters."1984" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1984" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1985 AS
year,temp_disasters."1985" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1985" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1986 AS
year,temp_disasters."1986" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1986" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1987 AS
year,temp_disasters."1987" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1987" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1988 AS
year,temp_disasters."1988" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1988" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1989 AS
year,temp_disasters."1989" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1989" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1990 AS
year,temp_disasters."1990" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1990" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1991 AS
year,temp_disasters."1991" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1991" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1992 AS
year,temp_disasters."1992" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1992" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1993 AS
year,temp_disasters."1993" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1993" IS NOT NULL;
```

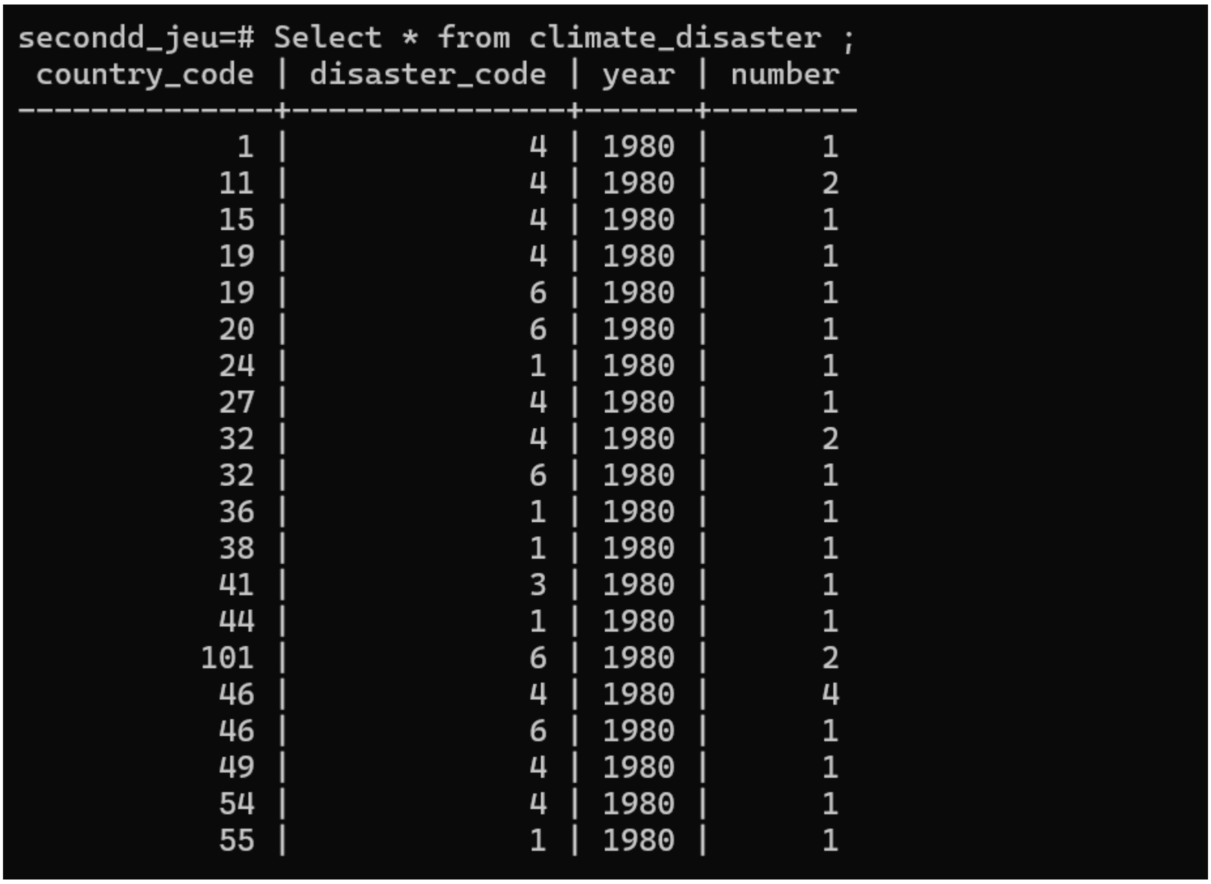
```
INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1994 AS
year,temp_disasters."1994" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1994" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1995 AS
year,temp_disasters."1995" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1995" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1996 AS
year,temp_disasters."1996" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1996" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1997 AS
year,temp_disasters."1997" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1997" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1998 AS
year,temp_disasters."1998" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1998" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1999 AS
year,temp_disasters."1999" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1999" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2000 AS
```

```
year,temp_disasters."2000" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2000" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2001 AS
year,temp_disasters."2001" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2001" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2002 AS
year,temp_disasters."2002" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2002" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2003 AS
year,temp_disasters."2003" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2003" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2004 AS
year,temp_disasters."2004" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2004" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2005 AS
year,temp_disasters."2005" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2005" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2006 AS
year,temp_disasters."2006" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2006" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2007 AS
year,temp_disasters."2007" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2007" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2008 AS
year,temp_disasters."2008" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2008" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2009 AS
year,temp_disasters."2009" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2009" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2010 AS
year,temp_disasters."2010" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2010" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2011 AS
year,temp_disasters."2011" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2011" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2012 AS
year,temp_disasters."2012" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2012" IS NOT NULL;
```

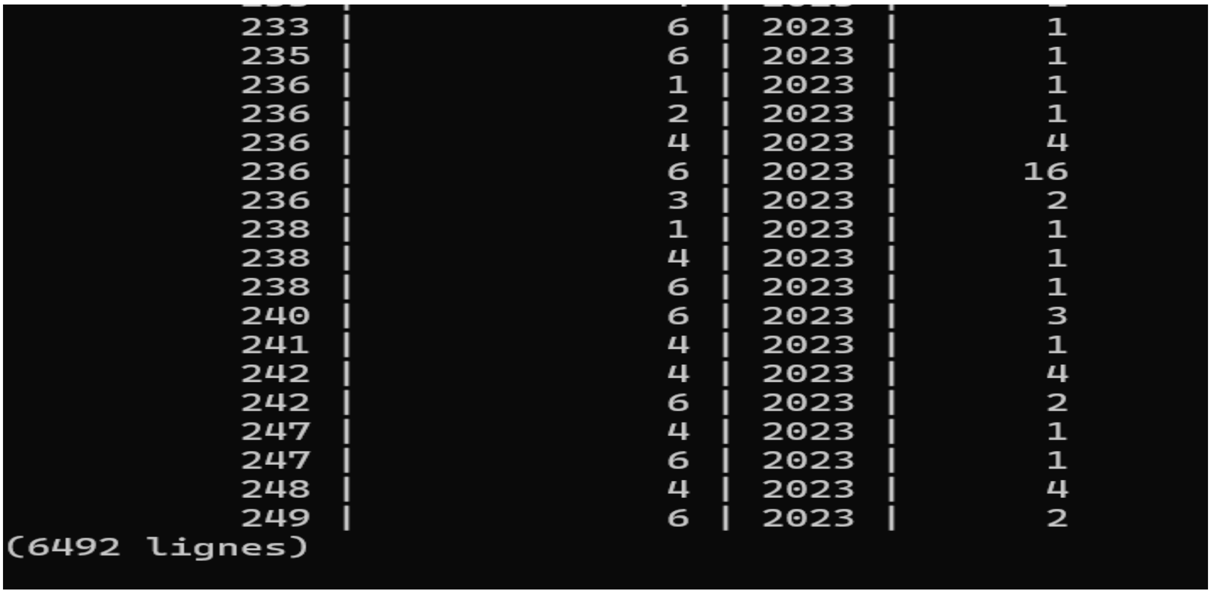
```
INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2013 AS
year,temp_disasters."2013" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2013" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2014 AS
year,temp_disasters."2014" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2014" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2015 AS
year,temp_disasters."2015" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2015" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2016 AS
year,temp_disasters."2016" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2016" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2017 AS
year,temp_disasters."2017" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2017" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2018 AS
year,temp_disasters."2018" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2018" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2019 AS
```

```
year,temp_disasters."2019" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2019" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2020 AS
year,temp_disasters."2020" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2020" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2021 AS
year,temp_disasters."2021" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2021" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2022 AS
year,temp_disasters."2022" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2022" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2023 AS
year,temp_disasters."2023" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2023" IS NOT NULL;
```

Resulats de l’insertion des donnes dans le tableau climate_disaster :



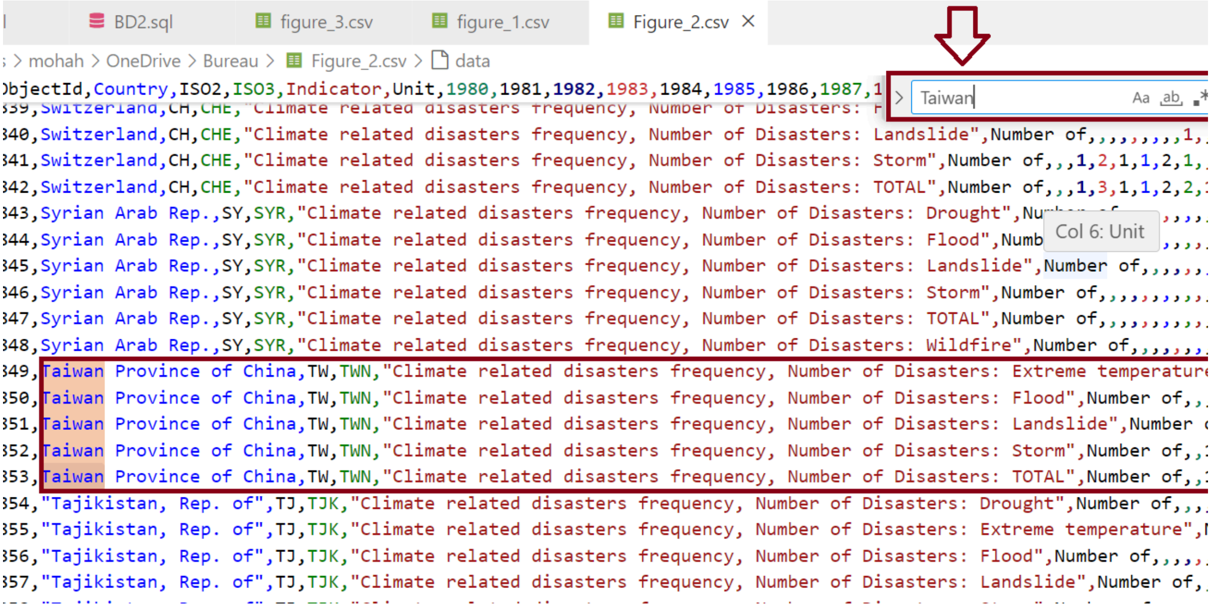
```
secondd_jeu=# Select * from climate_disaster ;
 country_code | disaster_code | year | number
--------------+---------------+------+--------
            1 |             4 | 1980 |      1
           11 |             4 | 1980 |      2
           15 |             4 | 1980 |      1
           19 |             4 | 1980 |      1
           19 |             6 | 1980 |      1
           20 |             6 | 1980 |      1
           24 |             1 | 1980 |      1
           27 |             4 | 1980 |      1
           32 |             4 | 1980 |      2
           32 |             6 | 1980 |      1
           36 |             1 | 1980 |      1
           38 |             1 | 1980 |      1
           41 |             3 | 1980 |      1
           44 |             1 | 1980 |      1
          101 |             6 | 1980 |      2
           46 |             4 | 1980 |      4
           46 |             6 | 1980 |      1
           49 |             4 | 1980 |      1
           54 |             4 | 1980 |      1
           55 |             1 | 1980 |      1
```



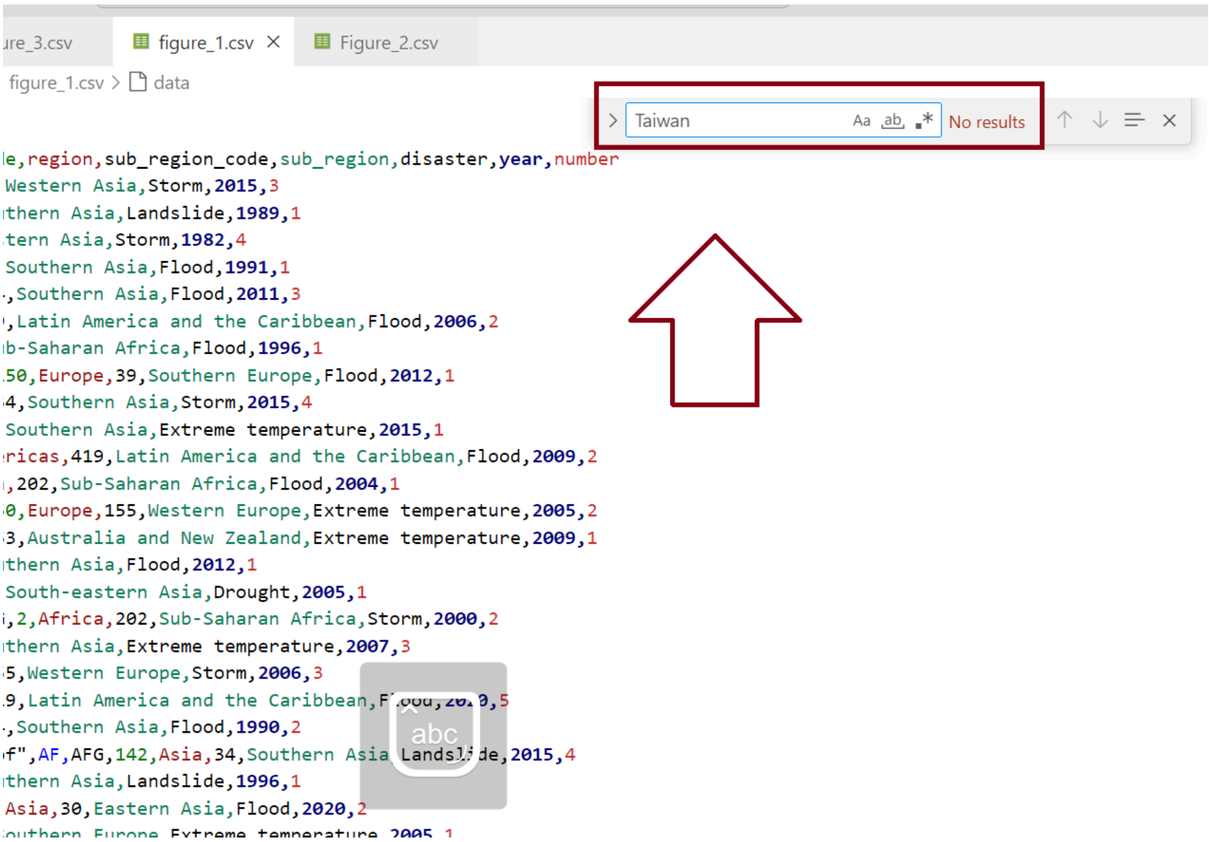
```
          233 |             6 | 2023 |      1
          235 |             6 | 2023 |      1
          236 |             1 | 2023 |      1
          236 |             2 | 2023 |      1
          236 |             4 | 2023 |      4
          236 |             6 | 2023 |     16
          236 |             3 | 2023 |      2
          238 |             1 | 2023 |      1
          238 |             4 | 2023 |      1
          238 |             6 | 2023 |      1
          240 |             6 | 2023 |      3
          241 |             4 | 2023 |      1
          242 |             4 | 2023 |      4
          242 |             6 | 2023 |      2
          247 |             4 | 2023 |      1
          247 |             6 | 2023 |      1
          248 |             4 | 2023 |      4
          249 |             6 | 2023 |      2
(6492 lignes)
```

*Remarque : J'ai bien remarqué une différence dans le nombre de lignes entre le premier jeu (chemin facile) et le second jeu (chemin difficile) après l'exécution de la commande SELECT * FROM climate_disaster ; dont on trouve 6448 lignes dans le chemin facile et 6492 dans le second jeu. Cela est dû au pays TAIWAN qui est absent dans le fichier csv de la figure 1 et il est bien présent dans le fichier de la figure 2 ce qui crée la différence entre les nombres de lignes*

Justificatif :



```
ObjectId,Country,ISO2,ISO3,Indicator,Unit,1980,1981,1982,1983,1984,1985,1986,1987,1
339,Switzerland,CH,CHE,"Climate related disasters frequency, Number of Disasters: F
340,Switzerland,CH,CHE,"Climate related disasters frequency, Number of Disasters: Landslide",Number of,,,,,,,,,1,
341,Switzerland,CH,CHE,"Climate related disasters frequency, Number of Disasters: Storm",Number of,,,1,2,1,1,2,1,
342,Switzerland,CH,CHE,"Climate related disasters frequency, Number of Disasters: TOTAL",Number of,,,1,3,1,1,2,2,
343,Syrian Arab Rep.,SY,SYR,"Climate related disasters frequency, Number of Disasters: Drought",Nu
344,Syrian Arab Rep.,SY,SYR,"Climate related disasters frequency, Number of Disasters: Flood",Numb
345,Syrian Arab Rep.,SY,SYR,"Climate related disasters frequency, Number of Disasters: Landslide",Number of,,,,,,
346,Syrian Arab Rep.,SY,SYR,"Climate related disasters frequency, Number of Disasters: Storm",Number of,,,,,,,,,,
347,Syrian Arab Rep.,SY,SYR,"Climate related disasters frequency, Number of Disasters: TOTAL",Number of,,,,,,,,,,
348,Syrian Arab Rep.,SY,SYR,"Climate related disasters frequency, Number of Disasters: Wildfire",Number of,,,,,,,
349,Taiwan Province of China,TW,TWN,"Climate related disasters frequency, Number of Disasters: Extreme temperature
350,Taiwan Province of China,TW,TWN,"Climate related disasters frequency, Number of Disasters: Flood",Number of,,
351,Taiwan Province of China,TW,TWN,"Climate related disasters frequency, Number of Disasters: Landslide",Number
352,Taiwan Province of China,TW,TWN,"Climate related disasters frequency, Number of Disasters: Storm",Number of,,
353,Taiwan Province of China,TW,TWN,"Climate related disasters frequency, Number of Disasters: TOTAL",Number of,,
354,"Tajikistan, Rep. of",TJ,TJK,"Climate related disasters frequency, Number of Disasters: Drought",Number of,,,
355,"Tajikistan, Rep. of",TJ,TJK,"Climate related disasters frequency, Number of Disasters: Extreme temperature",
356,"Tajikistan, Rep. of",TJ,TJK,"Climate related disasters frequency, Number of Disasters: Flood",Number of,,,,,
357,"Tajikistan, Rep. of",TJ,TJK,"Climate related disasters frequency, Number of Disasters: Landslide",Number of,
```



```
le,region,sub_region_code,sub_region,disaster,year,number
Western Asia,Storm,2015,3
ıthern Asia,Landslide,1989,1
tern Asia,Storm,1982,4
Southern Asia,Flood,1991,1
.,Southern Asia,Flood,2011,3
,Latin America and the Caribbean,Flood,2006,2
ıb-Saharan Africa,Flood,1996,1
.50,Europe,39,Southern Europe,Flood,2012,1
4,Southern Asia,Storm,2015,4
Southern Asia,Extreme temperature,2015,1
ricas,419,Latin America and the Caribbean,Flood,2009,2
,202,Sub-Saharan Africa,Flood,2004,1
0,Europe,155,Western Europe,Extreme temperature,2005,2
3,Australia and New Zealand,Extreme temperature,2009,1
ıthern Asia,Flood,2012,1
South-eastern Asia,Drought,2005,1
,2,Africa,202,Sub-Saharan Africa,Storm,2000,2
ıthern Asia,Extreme temperature,2007,3
5,Western Europe,Storm,2006,3
.9,Latin America and the Caribbean,Flood,20[illegible]0,5
.,Southern Asia,Flood,1990,2
f",AF,AFG,142,Asia,34,Southern Asia,Landslide,2015,4
ıthern Asia,Landslide,1996,1
Asia,30,Eastern Asia,Flood,2020,2
outhern Europe,Extreme temperature,2005,1
```

SCRIPT FINAL :

```
CREATE TEMPORARY TABLE temp_country_region (name VARCHAR,alpha_2
CHAR(2),alpha_3 CHAR(3),country_code  VARCHAR,iso_3166_2 VARCHAR,region
VARCHAR,sub_region VARCHAR,intermediate_region VARCHAR,region_code
VARCHAR ,sub_region_code  VARCHAR ,intermediate_region_code VARCHAR);

\COPY   temp_country_region FROM
C:\Users\mohah\OneDrive\Bureau\Figure_3.csv  DELIMITER ',' CSV HEADER;



UPDATE temp_country_region SET region_code = NULLIF(region_code,
''),sub_region_code = NULLIF(sub_region_code,
''),intermediate_region_code = NULLIF(intermediate_region_code, '');
```

```
ALTER TABLE temp_country_region

ALTER COLUMN region_code TYPE INT USING region_code::INT;

ALTER TABLE temp_country_region

ALTER COLUMN sub_region_code TYPE INT USING sub_region_code::INT;

ALTER TABLE temp_country_region

ALTER COLUMN intermediate_region_code TYPE INT USING
intermediate_region_code::INT;


CREATE TEMPORARY TABLE temp_disasters (ObjectId SERIAL PRIMARY
KEY,Country VARCHAR,ISO2 CHAR(2),ISO3 CHAR(3),Indicator VARCHAR,Unit
VARCHAR,"1980" INT,"1981" INT,"1982" INT,"1983" INT,"1984" INT,"1985"
INT,"1986" INT,"1987" INT,"1988" INT,"1989" INT,"1990" INT,"1991"
INT,"1992" INT,"1993" INT,"1994" INT,"1995" INT,"1996" INT,"1997"
INT,"1998" INT,"1999" INT,"2000" INT,"2001" INT,"2002" INT,"2003"
INT,"2004" INT,"2005" INT,"2006" INT,"2007" INT,"2008" INT,"2009"
INT,"2010" INT,"2011" INT,"2012" INT,"2013" INT,"2014" INT,"2015"
INT,"2016" INT,"2017" INT,"2018" INT,"2019" INT,"2020" INT,"2021"
INT,"2022" INT,"2023" INT,Source VARCHAR);


\COPY  temp_disasters FROM C:\Users\mohah\OneDrive\Bureau\Figure_2.csv
DELIMITER ',' CSV HEADER;


CREATE TABLE region ( region_code INTEGER PRIMARY KEY , name  VARCHAR
NOT NULL);


 CREATE TABLE sub_region ( name VARCHAR NOT NULL , region_code INTEGER
NOT NULL REFERENCES region (region_code) ON DELETE CASCADE,
sub_region_code INTEGER

 PRIMARY KEY);
 CREATE TABLE country ( name VARCHAR NOT NULL , ISO2 CHAR(2) UNIQUE ,
ISO3 CHAR(3)  UNIQUE , sub_region_code INTEGER REFERENCES
sub_region(sub_region_code) ON DELETE CASCADE, country_code SERIAL
PRIMARY KEY);
```

```
 CREATE TABLE disaster (disaster_code  SERIAL PRIMARY KEY , disaster
VARCHAR NOT NULL UNIQUE );
```

```
CREATE TABLE climate_disaster (country_code INTEGER NOT NULL REFERENCES
country (country_code) ON DELETE CASCADE, disaster_code INTEGER NOT
NULL REFERENCES disaster( disaster_code) ON DELETE CASCADE,  year
INTEGER NOT NULL CHECK(year>0) , number INTEGER CHECK(number>0) ,
PRIMARY KEY(country_code,disaster_code,year));
```

```
INSERT INTO region (region_code, name)

SELECT DISTINCT region_code, region

FROM temp_country_region

WHERE region_code IS NOT NULL AND region IS NOT NULL;
```

```
Insert into sub_region(name,region_code,sub_region_code) SELECT
DISTINCT sub_region , region_code ,sub_region_code from
temp_country_region Where sub_region IS NOT NULL AND sub_region_code IS
NOT NULL AND region_code IS NOT NULL ;
```

```
ALTER TABLE temp_country_region
```

```sql
ALTER COLUMN sub_region_code TYPE VARCHAR;


UPDATE temp_country_region

SET sub_region_code = NULL

WHERE sub_region_code = '';


ALTER TABLE temp_country_region

ALTER COLUMN sub_region_code TYPE INTEGER USING sub_region_code::INT;



INSERT INTO country (name, iso2, iso3, sub_region_code)

SELECT name,alpha_2,alpha_3,sub_region_code FROM temp_country_region;


INSERT INTO disaster (disaster) SELECT DISTINCT REPLACE(Indicator,
'Climate related disasters frequency, Number of Disasters: ', '') AS
disaster_name

FROM temp_disasters WHERE REPLACE(Indicator, 'Climate related disasters
frequency, Number of Disasters: ', '') != 'TOTAL';




INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1980 AS
year,temp_disasters."1980" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1980" IS NOT NULL;
```

```
INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1981 AS
year,temp_disasters."1981" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1981" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1982 AS
year,temp_disasters."1982" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1982" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1983 AS
year,temp_disasters."1983" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1983" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1984 AS
year,temp_disasters."1984" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1984" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1985 AS
year,temp_disasters."1985" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1985" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1986 AS
year,temp_disasters."1986" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1986" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1987 AS
```

```
year,temp_disasters."1987" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1987" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1988 AS
year,temp_disasters."1988" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1988" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1989 AS
year,temp_disasters."1989" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1989" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1990 AS
year,temp_disasters."1990" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1990" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1991 AS
year,temp_disasters."1991" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1991" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1992 AS
year,temp_disasters."1992" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1992" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1993 AS
year,temp_disasters."1993" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1993" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1994 AS
year,temp_disasters."1994" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1994" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1995 AS
year,temp_disasters."1995" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1995" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1996 AS
year,temp_disasters."1996" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1996" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1997 AS
year,temp_disasters."1997" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1997" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1998 AS
year,temp_disasters."1998" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1998" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,1999 AS
year,temp_disasters."1999" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."1999" IS NOT NULL;
```

```sql
INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2000 AS
year,temp_disasters."2000" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2000" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2001 AS
year,temp_disasters."2001" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2001" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2002 AS
year,temp_disasters."2002" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2002" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2003 AS
year,temp_disasters."2003" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2003" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2004 AS
year,temp_disasters."2004" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2004" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2005 AS
year,temp_disasters."2005" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2005" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2006 AS
```

```
year,temp_disasters."2006" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2006" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2007 AS
year,temp_disasters."2007" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2007" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2008 AS
year,temp_disasters."2008" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2008" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2009 AS
year,temp_disasters."2009" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2009" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2010 AS
year,temp_disasters."2010" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2010" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2011 AS
year,temp_disasters."2011" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2011" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2012 AS
year,temp_disasters."2012" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
```

```sql
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2012" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2013 AS
year,temp_disasters."2013" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2013" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2014 AS
year,temp_disasters."2014" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2014" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2015 AS
year,temp_disasters."2015" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2015" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2016 AS
year,temp_disasters."2016" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2016" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2017 AS
year,temp_disasters."2017" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2017" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2018 AS
year,temp_disasters."2018" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2018" IS NOT NULL;
```

```
INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2019 AS
year,temp_disasters."2019" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2019" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2020 AS
year,temp_disasters."2020" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2020" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2021 AS
year,temp_disasters."2021" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2021" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2022 AS
year,temp_disasters."2022" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2022" IS NOT NULL;

INSERT INTO climate_disaster (country_code, disaster_code, year,
number) SELECT country.country_code,disaster.disaster_code,2023 AS
year,temp_disasters."2023" AS number FROM temp_disasters JOIN country
USING (ISO3) JOIN disaster ON disaster.disaster =
REPLACE(temp_disasters.Indicator, 'Climate related disasters frequency,
Number of Disasters: ', '') WHERE temp_disasters."2023" IS NOT NULL;
```

FIN.

Hachim mohammed