**Table des matières**

[**Exercice 1 : Chargement du fichier dans une table unique** 2](#_Toc123928929)

[1-création d’une base de données : 3](#_Toc123928930)

[2-Se connecter à la base de données : 3](#_Toc123928931)

[3-Création du rôle avec mdp : 3](#_Toc123928932)

[4- Accordez tous les droits nécessaires pour rendre le schéma *etudiants* opérationnel dans la base pise 3](#_Toc123928933)

[Creation des tables : 4](#_Toc123928934)

[Import des fichiers csv : 7](#_Toc123928935)

[**Exercice 2 : nettoyage des colonnes** 9](#_Toc123928936)

[1. Après avoir exécuté les requêtes ci-dessous, dites quelles actions ont-elles sur les données et la structure de la table ECRITURE : 9](#_Toc123928937)

[2. Proposez une ou plusieurs requêtes qui changent le type de données des colonnes **HT, TVA, TTC** de **TEXT** à **DECIMAL**. 9](#_Toc123928938)

[3. Proposez une ou plusieurs requêtes qui change le type de données des colonnes **IDT\_MODE\_REG, IDT\_STATUT** et **IDT\_SS\_CATEGORIE** de **TEXT** à **INTEGER**. Que constatez-vous sur le cas de la colonne **IDT\_SS\_CATEGORIE** ? Comment y remédier ? 10](#_Toc123928939)

[4. Supprimez toutes les écritures qui n’ont pas de DATE puis renommez la colonne **DATE** en **DT\_ECRITURE**. Enfin proposez une requête qui change le type de **DT\_ECRITURE** de **TEXT** à **DATE**. 11](#_Toc123928940)

[**5.** Supprimez toutes les écritures qui n’ont pas de **NUMERO** 12](#_Toc123928941)

[6. Exécutez et commentez la requête suivante : 12](#_Toc123928942)

[7. Changez le type de données de la colonne **EXERCICE\_COMPTABLE** de TEXT à **INTEGER**. 14](#_Toc123928943)

[**Exercice 3 : création des contraintes sur les colonnes de tables** 15](#_Toc123928944)

[**Exercice 4 : vers l’établissement du bilan comptable** 19](#_Toc123928945)

[1. Proposer une requête SQL qui liste le nombre d’écritures ainsi que la somme des montants HT, TVA, TTC des écritures par année comptable. Cette liste doit être triée sur l’année de l’exercice comptable, de la plus récente à la plus ancienne. 19](#_Toc123928946)

[2. Proposer une requête SQL qui affiche le montant moyen TTC des écritures de catégorie « Fonctionnement Entreprise » 19](#_Toc123928947)

[3. Proposer une requête SQL qui affiche le nombre, la somme HT, la somme TTC par type d’opération 21](#_Toc123928948)

[4. Proposez une requête SQL qui permet d’afficher le bilan comptable de l’année 2022 sous la forme (ne pas tenir compte des montants affichés, seule la forme compte) : 22](#_Toc123928949)

[5. Commenter la requête ci-dessous. Que vous inspire le résultat obtenu : 24](#_Toc123928950)

[**Exercice 5 : Export des données** 26](#_Toc123928951)

# **Exercice 1 : Chargement du fichier dans une table unique**

* Générer un fichier CSV à partir des données de l’onglet ECRITURE du fichier Excel fourni
* Créer la table centrale **ECRITURE** avec toutes les colonnes de type **TEXT**
* Importer dans la table centrale créée l’intégralité du fichier CSV généré
* Créer et alimenter les autres tables

Nettoyage du fichier Excel :

Afin de débuter ce projet, il est nécessaire de mettre en ordre le fichier Excel et donc de le nettoyer correctement.

D’après notre réflexion, pour le nettoyage, plusieurs étapes se sont imposées.

La première étape est de fusionner les onglets en un seul et unique onglet écriture. Pour ce faire il convient de reprendre les informations des autres onglets en faisant correspondre par rapport à la configuration de l’onglet écriture. Toutes les informations sont à récupérer à l’exception des cellules, lignes ou colonnes comportant des calculs (par exemple des sommes ou totaux).

Il est important de converser la mise en page de l’onglet écriture et donc de faire correspondre les nouvelles informations importées.

Il faut ensuite défusionner les cellules. Cela est dans le but de ne pas retrouver des lignes en parties vides lors de la transformation en csv mais également lors de l’import des données.

Il a donc été nécessaire de rechercher toutes les cellules fusionnées et ce dans toutes les colonnes.

Dans certains cas la défusion de cellule a engendré une nouvelle ligne tandis que dans d’autres cas cela a engendré une nouvelle ligne vide qu’il a donc fallu supprimer.

Une fois tout le fichier Excel nettoyé des cellules fusionnées il convient de traiter les potentiels doublons que comporte la colonne n° de pièce. Cela sera nécessaire pour ne pas avoir deux fois un numéro de facture mais dont les montants et type seront différents.

Ensuite, il est nécessaire de supprimer les lignes comportant des calculs comme par exemple les lignes de calculs de fin d’exercice. Il en est de même avec certaines cellules qui sont en dehors du tableau avec des formules de résultats vis-à-vis de plusieurs cellules.

Enfin, une harmonisation de certains noms a été nécessaire en procédant par tri de catégorie dans chaque colonne. Cela a permis de déceler des erreurs de dates avec notamment des dates impossibles. Il est donc nécessaire d’uniformiser le tout. Il convient de faire de même avec chaque colonne. De nombreux noms ont été modifiés dans la colonne type avec par exemple des différences comme péage et péage autoroute. L’uniformisation de tous les noms qui correspondent au même type a permis de réduire le nombre de type mais aussi d’avoir une certaine concordance pour l’import des données. Cela a été également le cas pour des types identiques mais dont des fautes d’orthographes engendre des erreurs (dans le sens où deux types différents existent alors).

Cette tentative de nettoyage a été procédée pour chaque colonne, cellule (par exemple cela a permis de déceler un montant qui comportait une faute de frappe avec un ç à la place du 9).

Pour conclure, il est nécessaire d’ajouter des colonnes idt\_mode\_reg, idt\_statut et idt\_ss\_categorie. Une fois ces colonnes ajoutées et remplies, le fichier est terminé et prêt à être enregistré puis converti au format csv pour importation.

Ensuite nous sommes passés à la création d’une base de données pour entamer le travail sur Postgrés.

## 1-création d’une base de données :

create database pise ;

## 2-Se connecter à la base de données :

Afin de se connecter à la base de données PISE en tant que DBA :

psql -h localhost -p 5432 -U postgres -d pise

## 3-Création du rôle avec mdp :

Création du schéma etudiants avec le mot de passe etu\_pise\_2023

### 4- Accordez tous les droits nécessaires pour rendre le schéma *etudiants* opérationnel dans la base pise

CREATE ROLE etudiants LOGIN PASSWORD 'etu\_pise\_2023’;

CREATE SCHEMA IF NOT EXISTS etudiants AUTHORIZATION etudiants;

GRANT CONNECT ON DATABASE PISE TO etudiants;

GRANT USAGE ON SCHEMA etudiants TO etudiants;

GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA etudiants TO etudiants;

On crée le tablespace dans le tablespace data\_pise dans le repertoire C:\temp :

CREATE TABLESPACE data\_pise

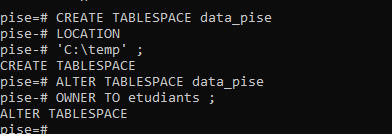
LOCATION

'C:\temp' ;

Attribution comme propriétaire de tablespace le compte etudiants :

ALTER TABLESPACE data\_pise

OWNER TO etudiants ;

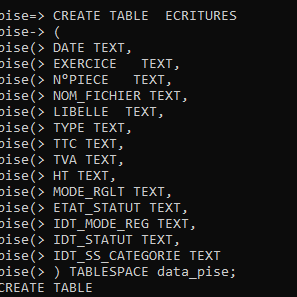


### Creation des tables :

Se connecter en tant que schema pour créer nos six tables :

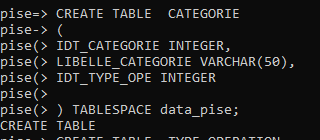
psql -h localhost -p 5432 -U etudiants -d pise

**Création de la table ecriture :**



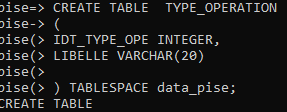
CREATE TABLE  ECRITURE  
(  
DATE TEXT,  
EXERCICE   TEXT,  
N°PIECE   TEXT,  
NOM\_FICHIER TEXT,  
LIBELLE  TEXT,  
TYPE TEXT,  
TTC TEXT,  
TVA TEXT,  
HT TEXT,  
MODE\_RGLT TEXT,  
ETAT\_STATUT TEXT,  
IDT\_MODE\_REG TEXT,  
IDT\_STATUT TEXT,  
IDT\_SS\_CATEGORIE TEXT  
) TABLESPACE data\_pise;

**Création de la table catégorie :**



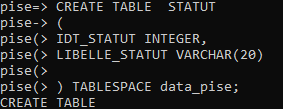
CREATE TABLE  CATEGORIE  
(  
IDT\_CATEGORIE INTEGER,  
LIBELLE\_CATEGORIE VARCHAR(50),  
IDT\_TYPE\_OPE INTEGER  
  
) TABLESPACE data\_pise;

**Création de la table type\_operation :**



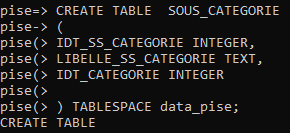
CREATE TABLE  TYPE\_OPERATION  
(  
IDT\_TYPE\_OPE INTEGER,  
LIBELLE VARCHAR(20)  
  
) TABLESPACE data\_pise;

**Création de la table statut :**



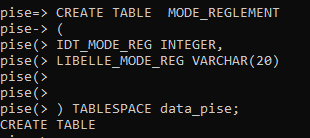
CREATE TABLE  STATUT  
(  
IDT\_STATUT INTEGER,  
LIBELLE\_STATUT VARCHAR(20)  
  
) TABLESPACE data\_pise;

**Création de la table sous\_categorie :**



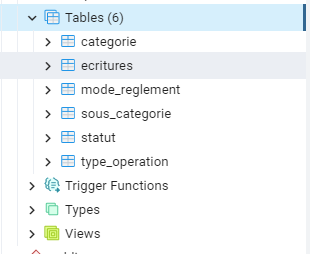
CREATE TABLE  SOUS\_CATEGORIE  
(  
IDT\_SS\_CATEGORIE INTEGER,  
LIBELLE\_SS\_CATEGORIE TEXT,  
IDT\_CATEGORIE INTEGER  
  
) TABLESPACE data\_pise;

**Création de la table mode\_reglement :**



CREATE TABLE  MODE\_REGLEMENT  
(  
IDT\_MODE\_REG INTEGER,  
LIBELLE\_MODE\_REG VARCHAR(20)  
  
  
) TABLESPACE data\_pise;

**Résultat :**



### Import des fichiers csv :

**Se connecter en tant que dba pour importer les fichiers csv :**

psql -h localhost -p 5432 -U postgres -d Pise

**Table ecriture :**

COPY ETUDIANTS.ECRITURE (DATE, EXERCICE, N°PIECE, NOM\_FICHIER, LIBELLE, TYPE, TTC,TVA,HT,MODE\_RGLT, ETAT\_STATUT, IDT\_MODE\_REG, IDT\_STATUT, IDT\_SS\_CATEGORIE)

FROM 'c:/temp/ECRITURE.CSV'

DELIMITER ';' csv header ;

**Table CATEGORIE :**

COPY ETUDIANTS. CATEGORIE (IDT\_CATEGORIE, LIBELLE\_CATEGORIE, IDT\_TYPE\_OPE)

FROM 'c:/temp/CATEGORIE.CSV'

DELIMITER ';' csv header ;

**Table TYPE\_OPERATION :**

COPY ETUDIANTS.TYPE\_OPERATION (IDT\_TYPE\_OPE, LIBELLE)

FROM 'c:/temp/ TYPE\_OPERATION.CSV'

DELIMITER ';' csv header ;

**Table STATUT :**

COPY ETUDIANTS.STATUT (IDT\_STATUT, LIBELLE\_STATUT)

FROM 'c:/temp/STATUT.CSV'

DELIMITER ';' csv header ;

**Table SOUS\_CATEGORIE :**

COPY ETUDIANTS.SOUS\_CATEGORIE (IDT\_SS\_CATEGORIE, LIBELLE\_SS\_CATEGORIE,IDT\_CATEGORIE)

FROM 'c:/temp/ SOUS\_CATEGORIE.CSV'

DELIMITER ';' csv header ;

**Table MODE\_REGLEMENT:**

COPY ETUDIANTS.MODE\_REGLEMENT (IDT\_MODE\_REG, LIBELLE\_MODE\_REG)

FROM 'c:/temp/MODE\_REGLEMENT.CSV'

DELIMITER ';' csv header ;

Après cette étape, les données seront en base et le reste des exercices se déroulent dans la table de données à coup de requêtes SQL.

# **Exercice 2 : nettoyage des colonnes**

## Après avoir exécuté les requêtes ci-dessous, dites quelles actions ont-elles sur les données et la structure de la table ECRITURE :

update ecriture set ht = replace(ht, ' ', '');

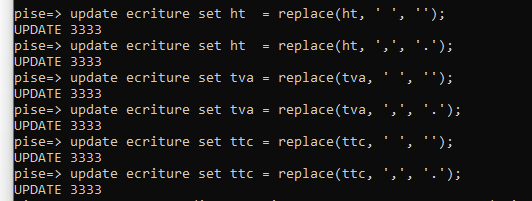
update ecriture set ht = replace(ht, ',', '.');

update ecriture set tva = replace(tva, ' ', '');

update ecriture set tva = replace(tva, ',', '.');

update ecriture set ttc = replace(ttc, ' ', '');

update ecriture set ttc = replace(ttc, ',', '.');



Ces requêtes ont deux effets sur les données et la structure de la table. Dans un premier temps, cela supprime les espaces entre les chiffres et ensuite cela modifie les virgules en points.

## Proposez une ou plusieurs requêtes qui changent le type de données des colonnes **HT, TVA, TTC** de **TEXT** à **DECIMAL**.

Changement du type de données de la colonne HT en DECIMAL :

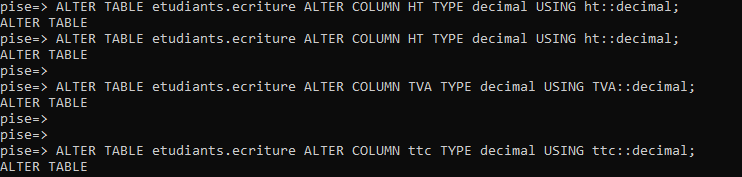
ALTER TABLE etudiants.ecriture ALTER COLUMN HT TYPE decimal USING ht::decimal;

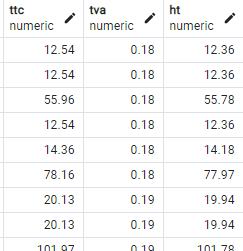
Changement du type de données de la colonne TVA en DECIMAL :

ALTER TABLE etudiants.ecriture ALTER COLUMN TVA TYPE decimal USING TVA::decimal;

Changement du type de données de la colonne TTC en DECIMAL :

ALTER TABLE etudiants.ecriture ALTER COLUMN ttc TYPE decimal USING ttc::decimal;



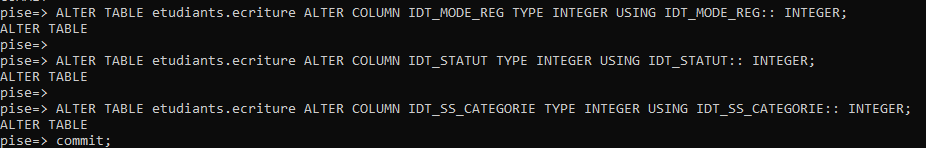


## Proposez une ou plusieurs requêtes qui change le type de données des colonnes **IDT\_MODE\_REG, IDT\_STATUT** et **IDT\_SS\_CATEGORIE** de **TEXT** à **INTEGER**. Que constatez-vous sur le cas de la colonne **IDT\_SS\_CATEGORIE** ? Comment y remédier ?

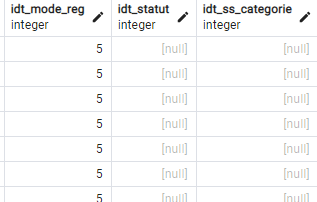
ALTER TABLE etudiants.ecriture ALTER COLUMN IDT\_MODE\_REG TYPE INTEGER USING IDT\_MODE\_REG:: INTEGER;

ALTER TABLE etudiants.ecriture ALTER COLUMN IDT\_STATUT TYPE INTEGER USING IDT\_STATUT:: INTEGER;

ALTER TABLE etudiants.ecriture ALTER COLUMN IDT\_SS\_CATEGORIE TYPE INTEGER USING IDT\_SS\_CATEGORIE:: INTEGER;



Nous ne constatons rien sur le cas de la colonne IDT\_SS\_CATEGORIE et nous ne pouvons donc pas y remédier.



## Supprimez toutes les écritures qui n’ont pas de DATE puis renommez la colonne **DATE** en **DT\_ECRITURE**. Enfin proposez une requête qui change le type de **DT\_ECRITURE** de **TEXT** à **DATE**.

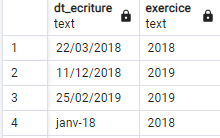
Afin de de supprimer les écritures qui n’ont pas de DATE il convient d’exécuter la requête suivant :

delete from etudiants.ecriture where DATE is null ;

Afin de de changer le type de DT\_ECRITURE de TEXT à DATE il convient d’exécuter la requête suivant :

ALTER TABLE ecriture ALTER COLUMN DT\_ECRITURE TYPE DATE USING DT\_ECRITURE::DATE ;







## Supprimez toutes les écritures qui n’ont pas de **NUMERO**

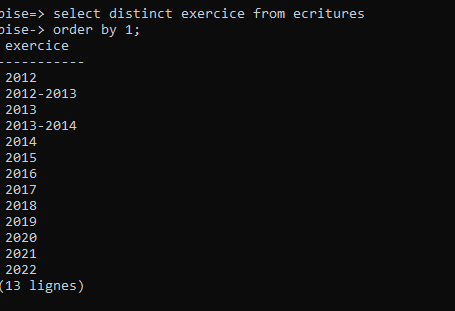
Requête permettant la suppression des écritures qui ne possèdent pas de NUMERO :

delete from etudiants.ecriture where N°PIECE IS NULL;

## Exécutez et commentez la requête suivante :

select distinct exercice from ecriture

order by 1;



Que constatez-vous ? Corriger les anomalies suivantes :

Nous constatons des anomalies d’années avec des lignes composées de deux années séparées d’un tiret.

2012-2013 en 2012

2013-2014 en 2013

null en 2022

Pour corriger les anomalies il est nécessaire d’exécuter les requêtes suivantes :

UPDATE ecriture

SET exercice = '2012'

WHERE exercice ='2012-2013' ;

UPDATE ecriture

SET exercice = '2013'

WHERE exercice ='2013-2014' ;

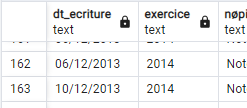
UPDATE ecriture

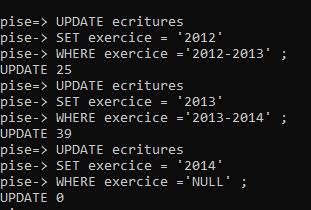
SET exercice = '2022'

WHERE exercice is NULL ;

Après avoir exécuté ces scripts, on a fait un commit.

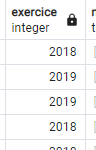
Exemple :



## Changez le type de données de la colonne **EXERCICE\_COMPTABLE** de TEXT à **INTEGER**.

ALTER TABLE etudiants.ecriture ALTER COLUMN EXERCICE TYPE INTEGER USING EXERCICE:: INTEGER;



# **Exercice 3 : création des contraintes sur les colonnes de tables**

Proposez des requêtes sql pour répondre aux besoins de contraintes suivants :

* Chaque table doit disposer d’une contrainte de clé de primaire conformément au MLD

**TABLE ecriture :**

ALTER TABLE ecriture ADD CONSTRAINT ecriture\_PK PRIMARY KEY (N°PIECE) USING INDEX TABLESPACE data\_pise ;

**TABLE TYPE\_OPERATION :**

ALTER TABLE TYPE\_OPERATION ADD CONSTRAINT TYPE\_OPERATION\_PK PRIMARY KEY (IDT\_TYPE\_OPE) ;

**TABLE CATEGORIE ;**

ALTER TABLE CATEGORIE ADD CONSTRAINT CATEGORIE\_PK PRIMARY KEY (IDT\_CATEGORIE) ;

**TABLE SOUS\_CATEGORIE :**

ALTER TABLE SOUS\_CATEGORIE ADD CONSTRAINT SOUS\_CATEGORIE\_PK PRIMARY KEY (IDT\_SS\_CATEGORIE) ;

**TABLE MODE\_REGLEMENT :**

ALTER TABLE MODE\_REGLEMENT ADD CONSTRAINT MODE\_REGLEMENT\_PK PRIMARY KEY (IDT\_MODE\_REG) ;

**TABLE STATUT:**

ALTER TABLE STATUT ADD CONSTRAINT STATUT\_PK PRIMARY KEY (IDT\_STATUT) ;

* Créer les contraintes de clés étrangères si besoin conformément au MLD :

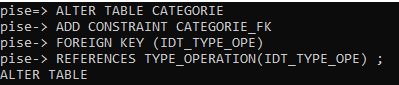
**Table categorie :**

ALTER TABLE CATEGORIE

ADD CONSTRAINT CATEGORIE\_FK

FOREIGN KEY (IDT\_TYPE\_OPE)

REFERENCES TYPE\_OPERATION(IDT\_TYPE\_OPE) ;



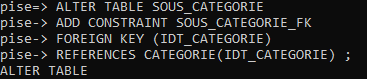
**Table sous\_categorie :**

ALTER TABLE SOUS\_CATEGORIE

ADD CONSTRAINT SOUS\_CATEGORIE\_FK

FOREIGN KEY (IDT\_CATEGORIE)

REFERENCES CATEGORIE(IDT\_CATEGORIE) ;



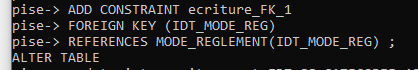
**Table ecriture :**

ALTER TABLE ecriture

ADD CONSTRAINT ecriture\_FK\_1

FOREIGN KEY (IDT\_MODE\_REG)

REFERENCES MODE\_REGLEMENT(IDT\_MODE\_REG) ;

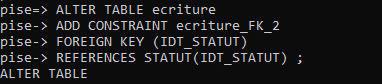


ALTER TABLE ecriture

ADD CONSTRAINT ecriture\_FK\_2

FOREIGN KEY (IDT\_STATUT)

REFERENCES STATUT(IDT\_STATUT) ;

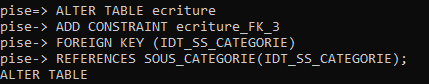


ALTER TABLE ecriture

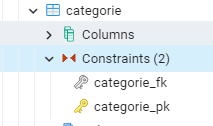
ADD CONSTRAINT ecriture\_FK\_3

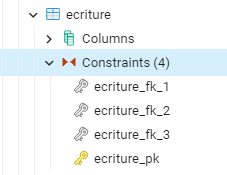
FOREIGN KEY (IDT\_SS\_CATEGORIE)

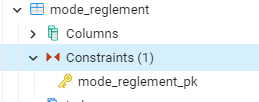
REFERENCES SOUS\_CATEGORIE(IDT\_SS\_CATEGORIE);

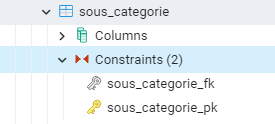


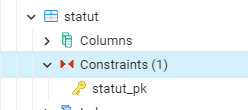
Résultats :

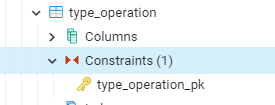












* Les colonnes suivantes ne doivent pas admettre du null :
  + Au niveau de la table ECRITURE : DATE\_ECRITURE, EXERCICE\_COMPTABLE, TTC
  + Au niveau de la table MODE\_REGLEMENT : LIBELLE\_MODE\_REG
  + Au niveau de la table STATUT : LIBELLE\_STATUT
  + Au niveau de la table CATEGORIE : LIBELLE\_CATEGORIE
  + Au niveau de la table SOUS\_CATEGORIE : LIBELLE\_SS\_CATEGORIE

**La table ecriture :**

ALTER TABLE ecriture ALTER COLUMN dt\_ecriture SET NOT NULL;

ALTER TABLE ecriture ALTER COLUMN exercice SET NOT NULL;

ALTER TABLE ecriture ALTER COLUMN ttc SET NOT NULL;

**La table Mode\_reglement :**

ALTER TABLE MODE\_REGLEMENT ALTER COLUMN libelle\_mode\_reg SET NOT NULL;

**La table libelle\_statut:**

ALTER TABLE statut ALTER COLUMN libelle\_statut SET NOT NULL;

**La table libelle\_categorie:**

ALTER TABLE categorie ALTER COLUMN libelle\_categorie SET NOT NULL;

**La table libelle\_ss\_categorie:**

ALTER TABLE sous\_categorie ALTER COLUMN libelle\_ss\_categorie SET NOT NULL;

A ce stade toutes les tables sont correctes en structure et en données. Les contraintes de clés primaires ainsi que les clés étrangères ont été créées.

# **Exercice 4 : vers l’établissement du bilan comptable**

Il vous est demandé à présent de :

## Proposer une requête SQL qui liste le nombre d’écritures ainsi que la somme des montants HT, TVA, TTC des écritures par année comptable. Cette liste doit être triée sur l’année de l’exercice comptable, de la plus récente à la plus ancienne.

select exercice,

count(\*) as "nombre\_ecritures",

sum(HT) as "somme\_ht",

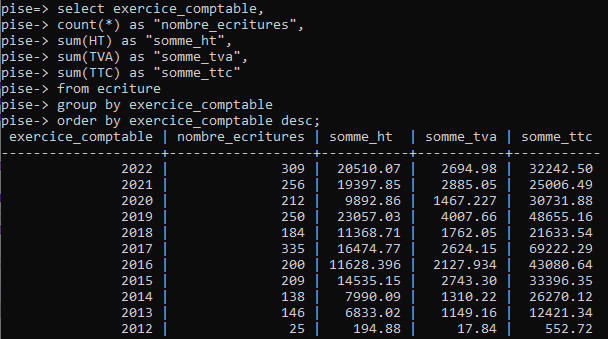
sum(TVA) as "somme\_tva",

sum(TTC) as "somme\_ttc"

from ecriture

group by exercice

order by exercice desc;



## Proposer une requête SQL qui affiche le montant moyen TTC des écritures de catégorie « Fonctionnement Entreprise »

Pour afficher l’idt\_categorie fonctionnement categorie dans la table ecriture :

select \*

from ecriture ecr

inner join sous\_categorie sscat on

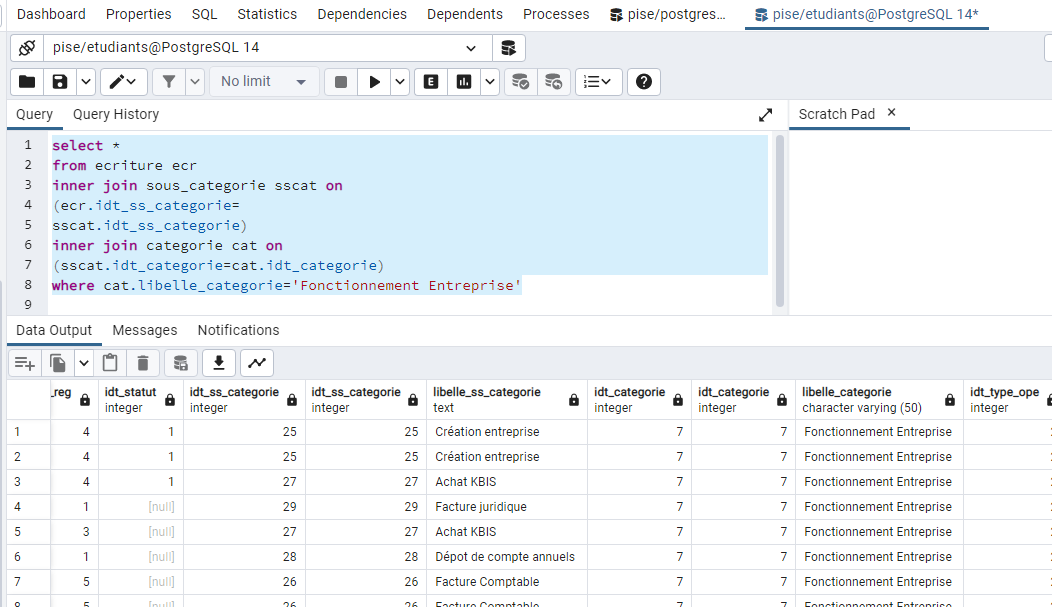
(ecr.idt\_ss\_categorie=

sscat.idt\_ss\_categorie)

inner join categorie cat on

(sscat.idt\_categorie=cat.idt\_categorie)

where cat.libelle\_categorie='Fonctionnement Entreprise';



Pour afficher le montant moyen TTC des écritures de catégorie « Fonctionnement Entreprise », nous avons utilisé la requête suivante :

select

avg (ecr.ttc)

from ecriture ecr

inner join sous\_categorie sscat on

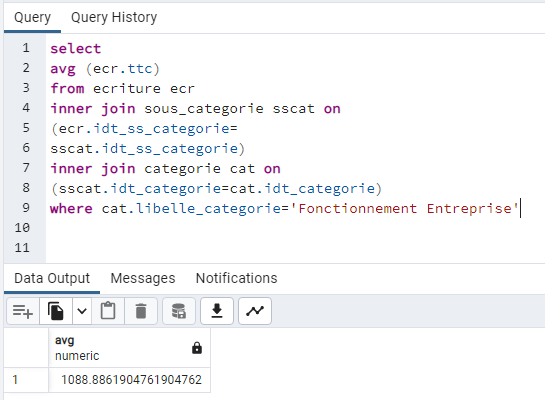
(ecr.idt\_ss\_categorie=

sscat.idt\_ss\_categorie)

inner join categorie cat on

(sscat.idt\_categorie=cat.idt\_categorie)

where cat.libelle\_categorie='Fonctionnement Entreprise';



## Proposer une requête SQL qui affiche le nombre, la somme HT, la somme TTC par type d’opération

select

top.libelle,

count(ecr.\*) as "nombre\_ecriture",

sum(ecr.HT) as "somme\_ht",

sum(ecr.ttc) as "somme\_ttc"

from ecriture ecr

inner join sous\_categorie sscat on

(ecr.idt\_ss\_categorie=

sscat.idt\_ss\_categorie)

inner join categorie cat

on (sscat.idt\_categorie=

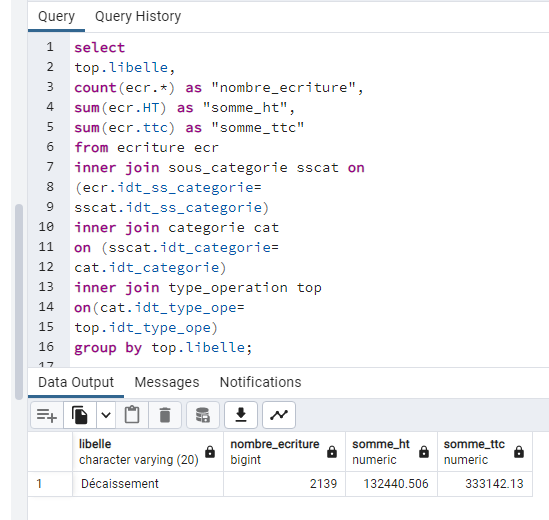
cat.idt\_categorie)

inner join type\_operation top

on(cat.idt\_type\_ope=

top.idt\_type\_ope)

group by top.libelle;



## Proposez une requête SQL qui permet d’afficher le bilan comptable de l’année 2022 sous la forme (ne pas tenir compte des montants affichés, seule la forme compte) :

Une image contenant texte

Description générée automatiquement

select

top.libelle as "libelle\_ope",

sum(ecr.HT) as "somme\_ht"

from ecriture ecr

inner join sous\_categorie sscat on

(ecr.idt\_ss\_categorie=

sscat.idt\_ss\_categorie)

inner join categorie cat

on (sscat.idt\_categorie=

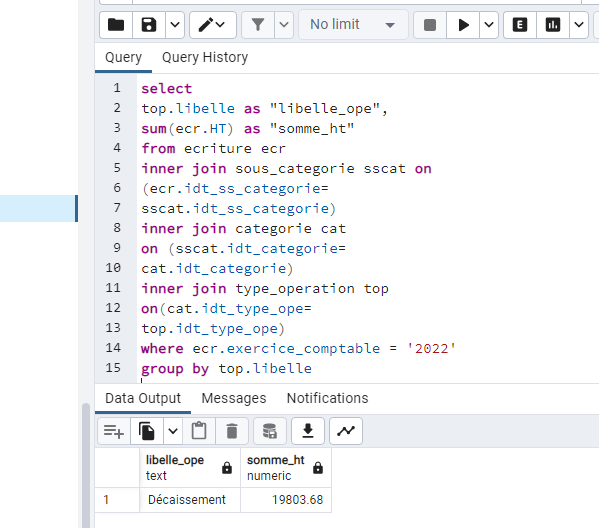
cat.idt\_categorie)

inner join type\_operation top

on(cat.idt\_type\_ope=

top.idt\_type\_ope)

where ecr.exercice = '2022'

group by top.libelle;

## Commenter la requête ci-dessous. Que vous inspire le résultat obtenu :

select

cat."LIBELLE\_CATEGORIE" as "Catégorie",

ss."LIBELLE\_SS\_CATEGORIE" as "Sous-Catégorie",

count(\*) as "nombre",

sum(ec.ht) as "somme\_ht"

from ecriture ec

inner join public."SOUS\_CATEGORIE" ss

on ec.idt\_ss\_categorie = ss."IDT\_SS\_CATEGORIE"

inner join public."CATEGORIE" cat

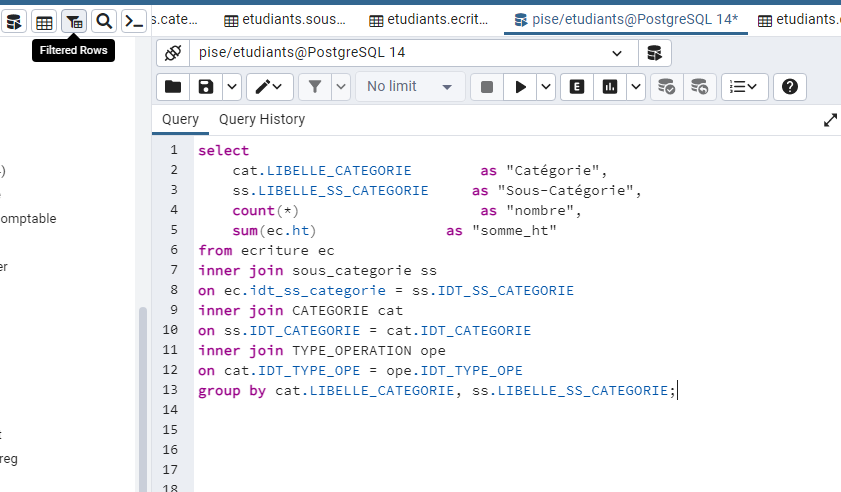
on ss."IDT\_CATEGORIE" = cat."IDT\_CATEGORIE"

inner join public."TYPE\_OPERATION" ope

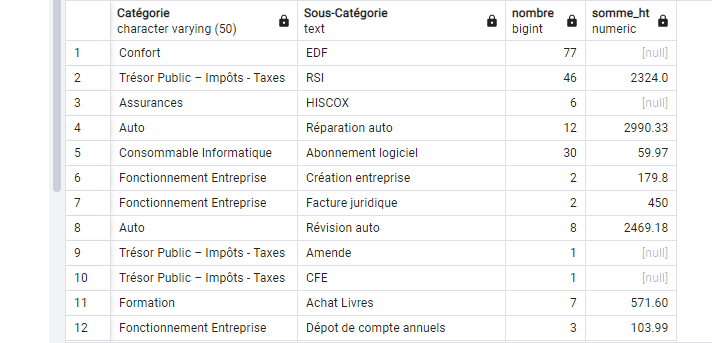
on cat."IDT\_TYPE\_OPE" = ope."IDT\_TYPE\_OPE"

group by cat."LIBELLE\_CATEGORIE", ss."LIBELLE\_SS\_CATEGORIE";

Afin de pouvoir exécuter cette requête nous avons du procéder à quelques modifications. Par conséquent, dans notre contexte, il a été nécessaire de supprimer public et les guillemets.



Résultat :

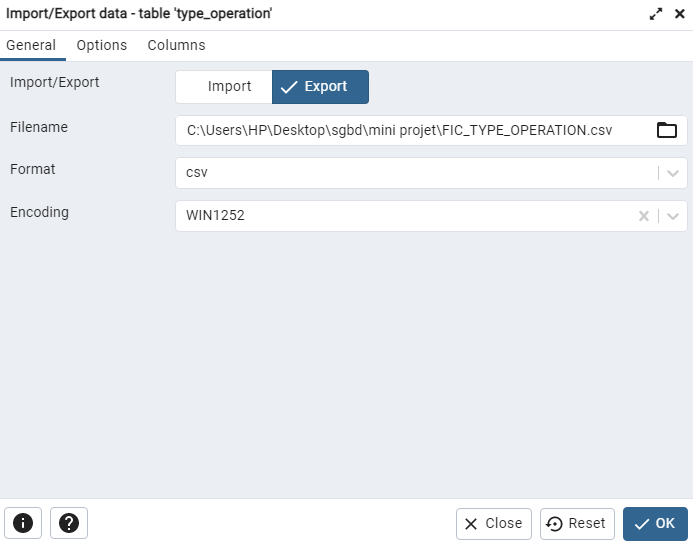


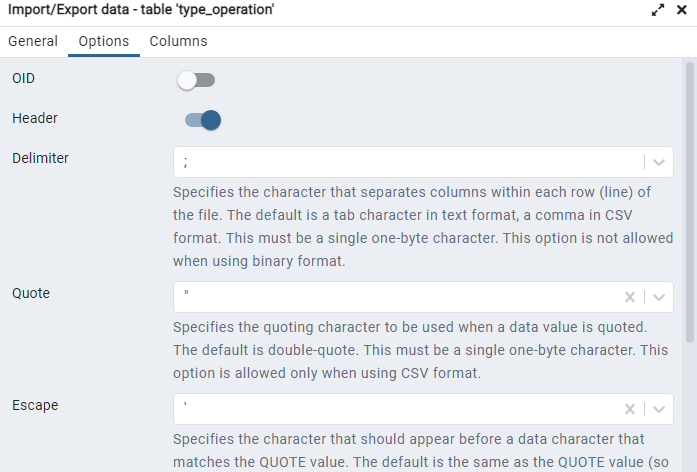
On remarque que cette requête affiche les nombres et la somme\_ht de tous les sous\_catégories.

# **Exercice 5 : Export des données**

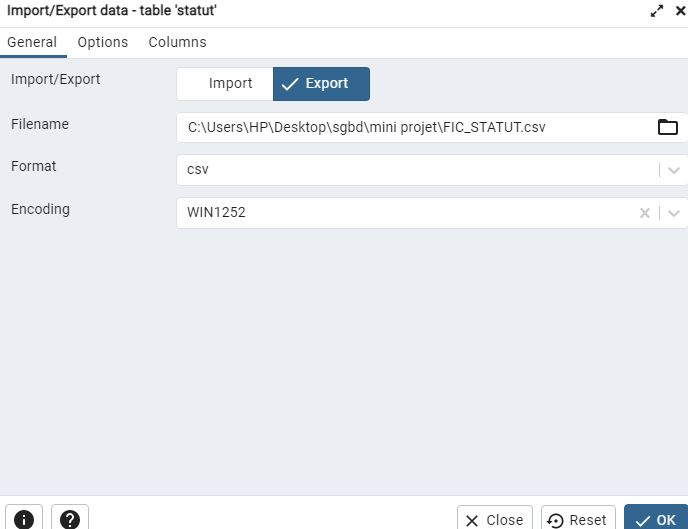
Afin qu’une autre équipe soit capable de tester votre travail, il vous est demande enfin d’exporter dans des fichiers CSV chacune des tables de de votre base de données. Il y’a en tout 6 tables et l’équipe de testeurs attend de votre part 6 fichiers csv nommés de la manière suivante :

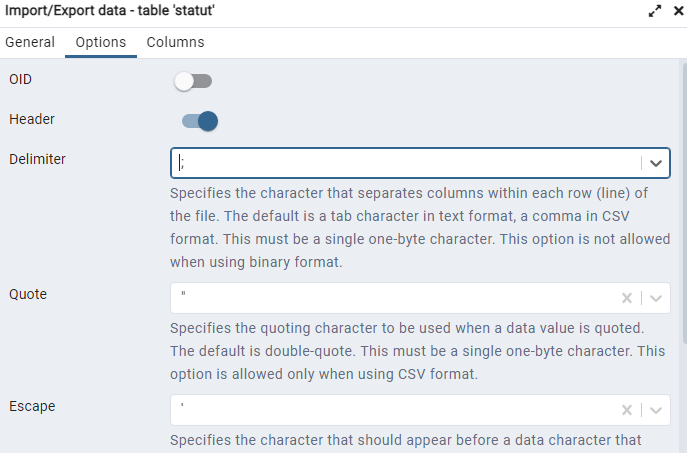
* FIC\_TYPE\_OPERATION.csv



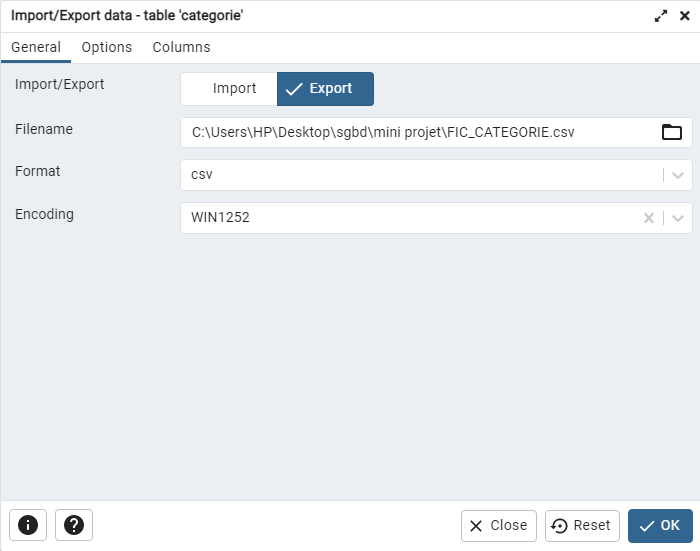


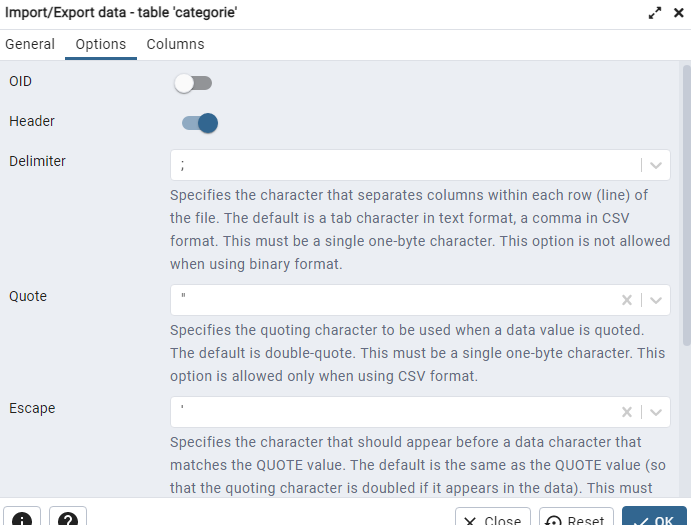
* FIC\_STATUT.csv



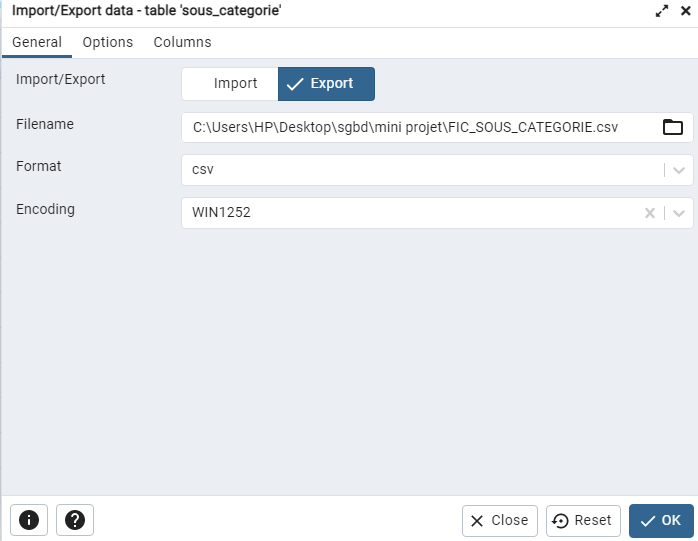


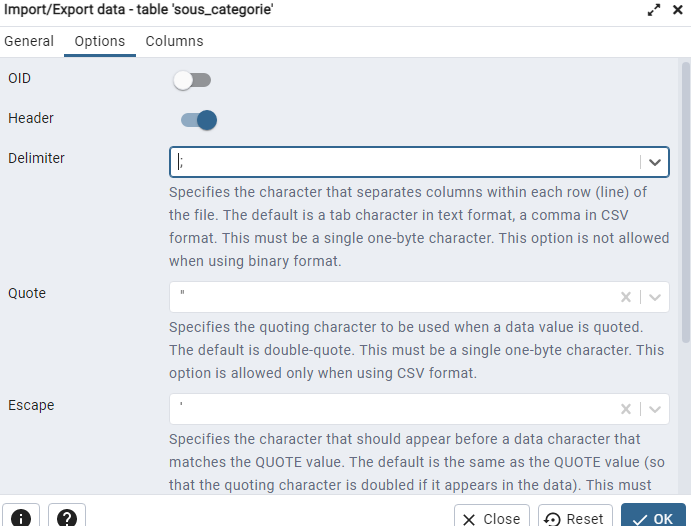
* FIC\_CATEGORIE.csv





* FIC\_SOUS\_CATEGORIE





* FIC\_ECRITURE.csv

Pour la table ecriture , on a utilisé le script suivant pour faire l’import :

\copy etudiants."ecriture" to 'C:\Users\HP\Desktop\sgbd\mini projet\FIC\_ECRITURE.csv' DELIMITER ';' CSV HEADER;



* FIC\_MODE\_REGLEMENT.csv

