## Exercice 1

### Créer les tables du MPD

*-- Création de la table Type\_opération*

CREATE TABLE

    IF NOT EXISTS TYPE\_OPERATION (IDT\_TYPE\_OPE TEXT, LIBELLE TEXT) TABLESPACE data\_pise;

*-- Création de la table Catégorie*

CREATE TABLE

    IF NOT EXISTS CATEGORIE (

        IDT\_CATEGORIE TEXT,

        LIBELLE\_CATEGORIE TEXT,

        IDT\_TYPE\_OPE TEXT

    ) TABLESPACE data\_pise;

*-- Création de la table Sous-catégorie*

CREATE TABLE

    IF NOT EXISTS SOUS\_CATEGORIE (

        IDT\_SS\_CATEGORIE TEXT,

        LIBELLE\_SS\_CATEGORIE TEXT,

        IDT\_CATEGORIE TEXT

    ) TABLESPACE data\_pise;

*-- Création de la table Mode de règlement*

CREATE TABLE

    IF NOT EXISTS MODE\_REGLEMENT (IDT\_MODE\_REG TEXT, LIBELLE\_MODE\_REG TEXT) TABLESPACE data\_pise;

*-- Création de la table Statut*

CREATE TABLE

    IF NOT EXISTS STATUT (IDT\_STATUT TEXT, LIBELLE\_STATUT TEXT) TABLESPACE data\_pise;

*-- Création de la table Ecriture*

CREATE TABLE

    IF NOT EXISTS ECRITURE (

        NUMERO TEXT,

        DT\_ECRITURE TEXT,

        FICHIER TEXT,

        LIBELLE TEXT,

        HT TEXT,

        TVA TEXT,

        TTC TEXT,

        EXERCICE\_COMPTABLE TEXT,

        IDT\_MODE\_REG TEXT,

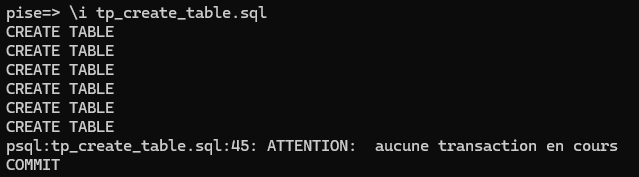
        IDT\_STATUT TEXT,

        IDT\_SS\_CATEGORIE TEXT

    ) TABLESPACE data\_pise;

COMMIT;

Le résultat dans l’invite de commande est :



### Alimenter les tables depuis les CSV

Connecter en tant que DBA, le script suivant est lancé :

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

FROM

    'C:\temp\import\_csv\TypeOperation\_table.csv' DELIMITER ';' HEADER;

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

FROM

    'C:\temp\import\_csv\Categorie\_table.csv' DELIMITER ';' HEADER;

COPY chokri.SOUS\_CATEGORIE (

    IDT\_SS\_CATEGORIE,

    LIBELLE\_SS\_CATEGORIE,

    IDT\_CATEGORIE

)

FROM

    'C:\temp\import\_csv\SousCategorie\_table.csv' DELIMITER ';' HEADER;

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

FROM

    'C:\temp\import\_csv\ModeReglement\_table.csv' DELIMITER ';' HEADER;

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

FROM

    'C:\temp\import\_csv\Statut\_table.csv' DELIMITER ';' HEADER;

COPY chokri.ECRITURE (

    NUMERO,

    DT\_ECRITURE,

    FICHIER,

    LIBELLE,

    HT,

    TVA,

    TTC,

    EXERCICE\_COMPTABLE,

    IDT\_MODE\_REG,

    IDT\_STATUT,

    IDT\_SS\_CATEGORIE

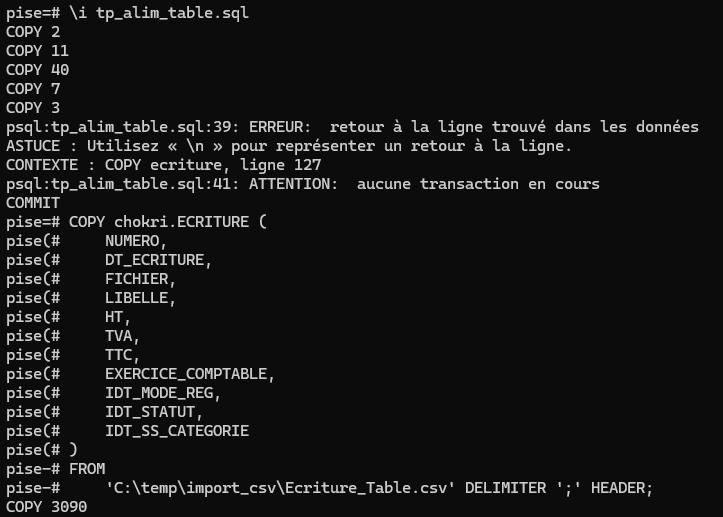
)

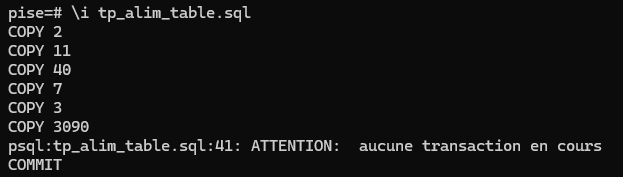
FROM

    'C:\temp\import\_csv\Ecriture\_Table.csv' DELIMITER ';' HEADER;

COMMIT;

Le résultat de ce script dans l’invite de commande est :





Lorsque le fichier source a été corriger de l’erreur ci-dessus

## Exercice 2

### Nettoyage de la table ECRITURE



Les requêtes exécutées ont permis de préparer les données des champs HT, TVA et TTC pour les transformer de type texte vers le type décimal. En informatique, un nombre décimal est composé d’une partie entière, puis d’un « . » et enfin une partie décimale. La suppression des espaces et la modification de virgule vers point est une étape préparant les données à ce changement (car ce sont des données monétaires).

### Changement de type : HT, TTC, TVA

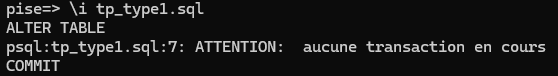
ALTER TABLE ECRITURE

ALTER COLUMN HT TYPE DECIMAL USING HT::decimal,

ALTER COLUMN TVA TYPE DECIMAL USING TVA::decimal,

ALTER COLUMN TTC TYPE DECIMAL USING TTC::decimal;

COMMIT ;



### Changement de type : Mode\_Reg, Statut, Ss\_Categorie

*-- MODIFICATION DES TYPES DE DONNEES*

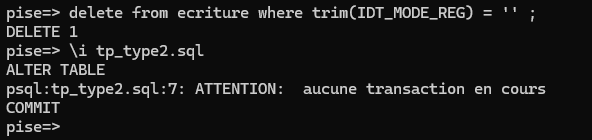
ALTER TABLE ECRITURE

ALTER COLUMN IDT\_MODE\_REG TYPE INTEGER USING IDT\_MODE\_REG::INTEGER,

ALTER COLUMN IDT\_STATUT TYPE INTEGER USING IDT\_STATUT::INTEGER,

ALTER COLUMN IDT\_SS\_CATEGORIE TYPE INTEGER USING IDT\_SS\_CATEGORIE::INTEGER;

COMMIT;



### Suppression des écritures : Pas de date d’écriture

DELETE FROM ECRITURE

WHERE

    LENGTH (NUMERO) = 0

    OR LENGTH (DT\_ECRITURE) = 0

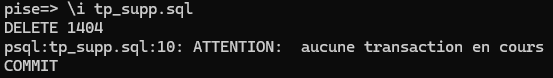
    OR LENGTH (HT) = 0

    OR LENGTH (TTC) = 0

    OR LENGTH (TVA) = 0

    OR LENGTH (IDT\_SS\_CATEGORIE) = 0;

COMMIT;





### Changement de nom : DT\_ECRITURE vers DATE\_ECRITURE

ALTER TABLE IF EXISTS ECRITURE

RENAME COLUMN DT\_ECRITURE TO DATE\_ECRITURE;

COMMIT ;



### Changement de type : Date\_ecriture

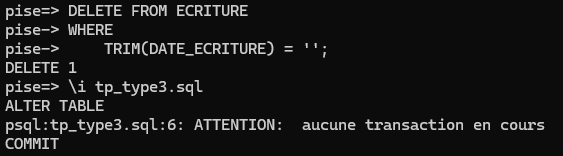
ALTER TABLE ECRITURE

ALTER COLUMN DATE\_ECRITURE

SET

    DATA TYPE date USING TO\_DATE (DATE\_ECRITURE, 'dd/mm/yyyy');

COMMIT;



### Suppression des écritures : Pas de numéro

*Voir requête n°4 ci-dessus.*

### Changement de type : Exercice\_comptable

UPDATE ECRITURE

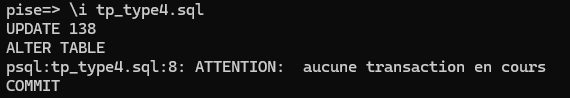
SET EXERCICE\_COMPTABLE = RIGHT(EXERCICE\_COMPTABLE, 4)

WHERE LENGTH(EXERCICE\_COMPTABLE) > 4 ;

ALTER TABLE ECRITURE

ALTER COLUMN EXERCICE\_COMPTABLE TYPE INTEGER USING EXERCICE\_COMPTABLE::INTEGER ;

COMMIT;



### Autres requêtes de changement de type

ALTER TABLE MODE\_REGLEMENT

ALTER COLUMN IDT\_MODE\_REG TYPE INTEGER USING IDT\_MODE\_REG::INTEGER ;

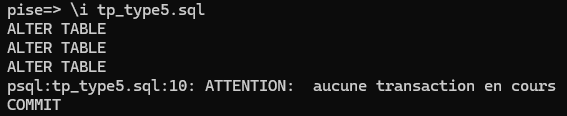
ALTER TABLE STATUT

ALTER COLUMN IDT\_STATUT TYPE INTEGER USING IDT\_STATUT::INTEGER;

ALTER TABLE SOUS\_CATEGORIE

ALTER COLUMN IDT\_SS\_CATEGORIE TYPE INTEGER USING IDT\_SS\_CATEGORIE::INTEGER;

COMMIT;



## Exercice 3

### Contraintes PK

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

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

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

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

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

ALTER TABLE ECRITURE ADD CONSTRAINT ECRITURE\_PK PRIMARY KEY (NUMERO);

COMMIT;



### Contraintes FK

ALTER TABLE CATEGORIE ADD CONSTRAINT CATEGORIE\_FK FOREIGN KEY (IDT\_TYPE\_OPE) REFERENCES TYPE\_OPERATION (IDT\_TYPE\_OPE);

ALTER TABLE SOUS\_CATEGORIE ADD CONSTRAINT SOUS\_CATEGORIE\_FK FOREIGN KEY (IDT\_CATEGORIE) REFERENCES CATEGORIE (IDT\_CATEGORIE);

ALTER TABLE ECRITURE ADD CONSTRAINT ECRITURE\_MODE\_REG\_FK FOREIGN KEY (IDT\_MODE\_REG) REFERENCES MODE\_REGLEMENT (IDT\_MODE\_REG);

ALTER TABLE ECRITURE ADD CONSTRAINT ECRITURE\_STATUT\_FK FOREIGN KEY (IDT\_STATUT) REFERENCES STATUT (IDT\_STATUT);

ALTER TABLE ECRITURE ADD CONSTRAINT ECRITURE\_SS\_CAT\_FK FOREIGN KEY (IDT\_SS\_CATEGORIE) REFERENCES SOUS\_CATEGORIE (IDT\_SS\_CATEGORIE);

COMMIT;



### Changement de structure : Not null

ALTER TABLE ECRITURE

ALTER COLUMN DATE\_ECRITURE

SET NOT NULL,

ALTER COLUMN EXERCICE\_COMPTABLE

SET NOT NULL,

ALTER COLUMN TTC

SET NOT NULL;

ALTER TABLE MODE\_REGLEMENT

ALTER COLUMN LIBELLE\_MODE\_REG

SET NOT NULL;

ALTER TABLE STATUT

ALTER COLUMN LIBELLE\_STATUT

SET NOT NULL;

ALTER TABLE CATEGORIE

ALTER COLUMN LIBELLE\_CATEGORIE

SET NOT NULL;

ALTER TABLE SOUS\_CATEGORIE

ALTER COLUMN LIBELLE\_SS\_CATEGORIE

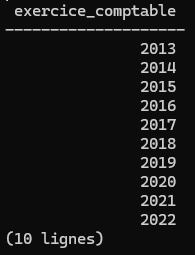
SET NOT NULL;

COMMIT;



## Exercice 4

### Commentaire de requête



La requête affiche les différentes années comptables présent dans la colonne exercice\_comptable. L’instruction DISTINCT permet de n’afficher aucun doublon. L’instruction order by permet de trier de façon croissante.

### Requête : Nombre d’écriture, somme de HT, TVA et TTC par exercice comptable

SELECT

    COUNT(\*) AS "Nombre ecriture",

    SUM(HT) AS "Total HC",

    SUM(TVA) AS "Total TVA",

    SUM(TTC) AS "Total TTC",

    EXERCICE\_COMPTABLE AS "Annee Comptable"

FROM

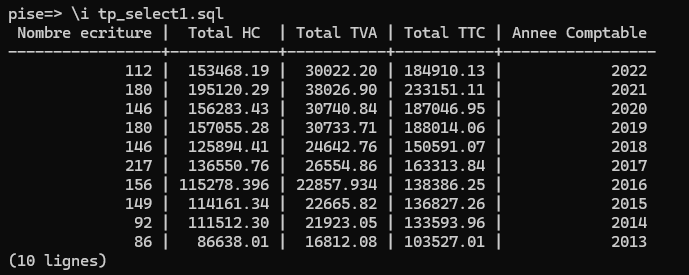
    ECRITURE

GROUP BY

    EXERCICE\_COMPTABLE

ORDER BY

    EXERCICE\_COMPTABLE DESC;



### Requête : Montant moyen TTC de la catégorie « Fonctionnement Entreprise »

SELECT

    AVG(TTC) AS "Montant TTC Moyen"

FROM

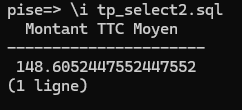
    ECRITURE

    INNER JOIN SOUS\_CATEGORIE ON ECRITURE.IDT\_SS\_CATEGORIE = SOUS\_CATEGORIE.IDT\_SS\_CATEGORIE

    INNER JOIN CATEGORIE ON CATEGORIE.IDT\_CATEGORIE = SOUS\_CATEGORIE.IDT\_CATEGORIE

WHERE

    UPPER(LIBELLE\_CATEGORIE) = 'FONCTIONNEMENT ENTREPRISE';



### Requête : Nombre, Somme HT et TTC par type d’opération

SELECT

    TYPE\_OPERATION.LIBELLE AS "Type Operation",

    COUNT(\*) AS "Nombre",

    SUM(ECRITURE.HT) AS "Total HT",

    SUM(ECRITURE.TTC) AS "Total TTC"

FROM

    ECRITURE

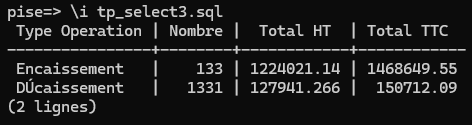
    INNER JOIN SOUS\_CATEGORIE ON ECRITURE.IDT\_SS\_CATEGORIE = SOUS\_CATEGORIE.IDT\_SS\_CATEGORIE

    INNER JOIN CATEGORIE ON CATEGORIE.IDT\_CATEGORIE = SOUS\_CATEGORIE.IDT\_CATEGORIE

    INNER JOIN TYPE\_OPERATION ON TYPE\_OPERATION.IDT\_TYPE\_OPE = CATEGORIE.IDT\_TYPE\_OPE

GROUP BY

    TYPE\_OPERATION.LIBELLE;



### Requête : Bilan comptable 2022

SELECT

    TYPE\_OPERATION.LIBELLE AS "Type Operation",

    SUM(ECRITURE.HT) AS "Total HT"

FROM

    ECRITURE

    INNER JOIN SOUS\_CATEGORIE ON ECRITURE.IDT\_SS\_CATEGORIE = SOUS\_CATEGORIE.IDT\_SS\_CATEGORIE

    INNER JOIN CATEGORIE ON CATEGORIE.IDT\_CATEGORIE = SOUS\_CATEGORIE.IDT\_CATEGORIE

    INNER JOIN TYPE\_OPERATION ON TYPE\_OPERATION.IDT\_TYPE\_OPE = CATEGORIE.IDT\_TYPE\_OPE

GROUP BY

    EXERCICE\_COMPTABLE,

    TYPE\_OPERATION.LIBELLE

HAVING

    EXERCICE\_COMPTABLE = 2022;

