CREATE TABLE Mecanisme(

mecanisme\_id SERIAL PRIMARY KEY,

nom VARCHAR(255)

);

CREATE TABLE Jeux(

id\_jeu SERIAL PRIMARY KEY,

titre VARCHAR(255),

date\_parution\_debut VARCHAR,

date\_parution\_fin VARCHAR,

information\_date VARCHAR(255),

version VARCHAR(255),

nombre\_de\_joueurs VARCHAR(255),

age\_indique VARCHAR(255),

mots\_cles VARCHAR(255),

mecanisme\_id INT REFERENCES Mecanisme(mecanisme\_id)

);

CREATE TABLE JeuMecanisme(

jeu\_id INT REFERENCES Jeux(id\_jeu),

mecanisme\_id INT REFERENCES Mecanisme(mecanisme\_id)

);

CREATE TABLE Editeur(

editeur\_id SERIAL PRIMARY KEY,

nom VARCHAR(255)

);

CREATE TABLE Collection(

collection\_id SERIAL PRIMARY KEY,

nom VARCHAR(255)

);

CREATE TABLE Localisation(

localisation\_id SERIAL PRIMARY KEY,

salle VARCHAR,

etagere VARCHAR

);

CREATE TABLE Boites(

boite\_id SERIAL PRIMARY KEY,

etat VARCHAR(255),

code\_barre VARCHAR(255),

jeu\_id INT REFERENCES Jeux(id\_jeu),

collection\_id INT REFERENCES Collection(collection\_id)

);

CREATE TABLE Auteur(

auteur\_id SERIAL PRIMARY KEY,

nom VARCHAR(255)

);

CREATE TABLE JeuAuteur(

jeu\_id SERIAL REFERENCES Jeux(id\_jeu),

auteur\_id INT REFERENCES Auteur(auteur\_id)

);

CREATE TABLE Emprunteur(

emprunteur\_id SERIAL PRIMARY KEY,

nom VARCHAR(255),

email VARCHAR(255),

telephone VARCHAR(20),

adresse TEXT

);

CREATE TABLE Pret(

pret\_id SERIAL PRIMARY KEY,

boite\_id SERIAL REFERENCES Boites(boite\_id),

emprunteur\_id SERIAL REFERENCES Emprunteur(emprunteur\_id),

date\_emprunt DATE,

date\_retour DATE

);

CREATE TABLE Categorie(

categorie\_id SERIAL PRIMARY KEY,

nom VARCHAR

);

CREATE TABLE JeuCategories(

jeu\_id SERIAL REFERENCES Jeux(id\_jeu),

categorie\_id SERIAL REFERENCES Categorie(categorie\_id)

);

CREATE TABLE jeuediteur(

jeu\_id SERIAL REFERENCES Jeux(id\_jeu),

editeur\_id SERIAL REFERENCES Editeur(editeur\_id)

);

CREATE TABLE utilisateurs (

id SERIAL PRIMARY KEY,

numero\_etu INT,

nom VARCHAR,

prenom VARCHAR,

mot\_de\_passe VARCHAR(255));

CREATE TABLE roles (

role\_id SERIAL PRIMARY KEY,

role\_name VARCHAR(50) NOT NULL UNIQUE

);

CREATE TABLE temp (

id INT,

TITRE VARCHAR(255),

REFERENCE VARCHAR(255),

AUTEURS VARCHAR,

DATE\_DE\_PARUTION\_DEBUT VARCHAR(255),

DATE\_DE\_PARUTION\_FIN VARCHAR(255),

INFORMATION\_DATE VARCHAR(255),

VERSION VARCHAR(255),

NOMBRE\_DE\_JOUEURS VARCHAR(255),

AGE\_INDIQUE VARCHAR(255),

MOTS\_CLES VARCHAR(255),

N\_Boite VARCHAR,

LOCALISATION\_CNJ VARCHAR(255),

MECANISME VARCHAR(255),

Collection\_dorigine VARCHAR(255),

Etat VARCHAR(255),

Code\_barre VARCHAR(255),

REFERENCESS VARCHAR(255)

);

**CA VEUT DIRE QUE DANAS LE FICHIER EXCEL FAUT AUSSI**

**CETTE COMMANDE POUR INSERER TOUTES LES DONNÉES DU FICHIER DANS LA TABLE TEMP :** \COPY temp(id,TITRE,REFERENCE,AUTEURS,DATE\_DE\_PARUTION\_DEBUT,DATE\_DE\_PARUTION\_FIN,INFORMATION\_DATE,VERSION,NOMBRE\_DE\_JOUEURS,AGE\_INDIQUE,MOTS\_CLES,N\_Boîte,LOCALISATION\_CNJ,MECANISME,Collection\_dorigine,Etat,Code\_barre,REFERENCESS) FROM '**METTRE LE CHEMIN DE INVENTAIRE\_2712.XLSX(téléchargeable dans le drive)**' DELIMITER ',' CSV HEADER ENCODING 'ISO-8859-1';

**INSERTIONS À FAIRE :**

INSERT INTO Mecanisme (nom) SELECT MECANISME FROM temp;

INSERT INTO Jeux (id\_jeu,titre,date\_parution\_debut,date\_parution\_fin,information\_date,version,nombre\_de\_joueurs,age\_indique,mots\_cles) SELECT id,TITRE,DATE\_DE\_PARUTION\_DEBUT,DATE\_DE\_PARUTION\_FIN,INFORMATION\_DATE,VERSION,NOMBRE\_DE\_JOUEURS,AGE\_INDIQUE,MOTS\_CLES FROM temp;

INSERT INTO categorie(nom) SELECT DISTINCT mots\_cles FROM temp;

INSERT INTO editeur(nom) SELECT DISTINCT reference FROM temp;

INSERT INTO collection(nom) SELECT DISTINCT collection\_dorigine FROM temp;

INSERT INTO auteur(nom) SELECT DISTINCT auteurs FROM temp;

INSERT INTO jeumecanisme (jeu\_id, mecanisme\_id)

SELECT j.id\_jeu, m.mecanisme\_id

FROM Jeux j

JOIN temp t ON j.titre = t.TITRE -- Lier le jeu via le titre

JOIN Mecanisme m ON m.nom = t.MECANISME; -- Lier le mécanisme via le nom

SELECT \* FROM mecanisme where mecanisme\_id = 40;

SELECT mecanisme from temp where mecanisme is not null;

UPDATE Jeux

SET mecanisme\_id = (

SELECT MIN(m.mecanisme\_id) -- Utilise MIN pour garantir une seule valeur

FROM Mecanisme m

JOIN Temp t ON m.nom = t.MECANISME -- Lie Mecanisme et Temp via leur nom

WHERE t.TITRE = Jeux.titre -- Associe le jeu par le titre

)

WHERE EXISTS (

SELECT 1

FROM Mecanisme m

JOIN Temp t ON m.nom = t.MECANISME

WHERE t.TITRE = Jeux.titre

);

UPDATE Jeux

SET

nombre\_de\_joueurs = COALESCE(jeux.nombre\_de\_joueurs, temp.nombre\_de\_joueurs),

age\_indique = COALESCE(jeux.age\_indique, temp.age\_indique),

mots\_cles = COALESCE(jeux.mots\_cles, temp.mots\_cles)

FROM temp

WHERE Jeux.id\_jeu = temp.id

AND (Jeux.nombre\_de\_joueurs IS NULL OR

Jeux.age\_indique IS NULL OR

Jeux.mots\_cles IS NULL);

INSERT INTO jeuediteur (jeu\_id, editeur\_id)

SELECT j.id\_jeu,e.editeur\_id

FROM Jeux j

JOIN temp t ON j.titre = t.TITRE -- Lier le jeu via le titre

JOIN editeur e ON e.nom = t.reference;

INSERT INTO jeuauteur (jeu\_id, auteur\_id)

SELECT j.id\_jeu, a.auteur\_id

FROM Jeux j

JOIN temp t ON j.titre = t.TITRE -- Lier le jeu via le titre

JOIN auteur a ON a.nom = t.auteurs;

INSERT INTO boites (etat, code\_barre, jeu\_id, collection\_id,nboite)

SELECT DISTINCT

t.etat,

t.code\_barre,

j.id\_jeu,

c.collection\_id,

t.n\_boîte

FROM temp t

LEFT JOIN Jeux j ON t.titre = j.titre -- Associe temp avec Jeux

LEFT JOIN Collection c ON t.collection\_dorigine = c.nom;

INSERT INTO pret (boite\_id) SELECT DISTINCT boite\_id from boites;

INSERT INTO roles (role\_name) VALUES

('Utilisateur'),

('Administrateur'),

('Gestionnaire');

ALTER TABLE utilisateurs ADD COLUMN role\_id INT;

-- Ajouter une clé étrangère pour garantir l'intégrité des données

ALTER TABLE utilisateurs ADD CONSTRAINT fk\_role FOREIGN KEY (role\_id) REFERENCES roles(role\_id);

INSERT INTO utilisateurs (nom, email, mot\_de\_passe, role\_id)

VALUES ('John Doe', 'john@example.com', 'hashed\_password',

(SELECT role\_id FROM roles WHERE role\_name = 'Utilisateur'));

SELECT

utilisateurs.id\_utilisateur,

utilisateurs.nom,

utilisateurs.email,

roles.role\_name

FROM utilisateurs

LEFT JOIN roles ON utilisateurs.role\_id = roles.role\_id;