**INSTITUT SUPERIEUR DE COMMERCE DE GOMA**

**ISC-GOMA**



https://www.iscgoma.ac.cd

**Section  Informatique**

**TRAVAIL PRATIQUE DE LA BASE DE DONNEES WEB**

**ET**

**MOBILE**

**Le Système de gestion d’une bibliothèque universitaire**

**Par :**

1. **BWEMA KAMALA VANNY**
2. **KUKWAKWABAKOLO KANDOLO**
3. **PALUKU MAJIVUNO**
4. **TAMBWE PODA**
5. **MUHINDO MUKUKYO**

**Domaine :** Sciences économiques et de gestion

**Filière :** Gestion Commerciale et Administrative

**Mention :** Gestion Informatique.

Sous la supervision de : Mr l’Assistant Josué KALEMA

**Année Académique : 2024 – 2025**

La réponse complète au Travail Pratique - Examen de Base de Données Web et Mobile concernant le Système de gestion d’une bibliothèque universitaire, en respectant les consignes données.

La bibliothèque souhaite informatiser la gestion des livres, emprunts, étudiants et bibliothécaires. Le système devra permettre :

L’enregistrement des livres et auteurs

L’inscription des étudiants

Le suivi des emprunts et retours

L’association d’un bibliothécaire à chaque emprunt enregistré

**Tables possibles** :

Livre

Auteur

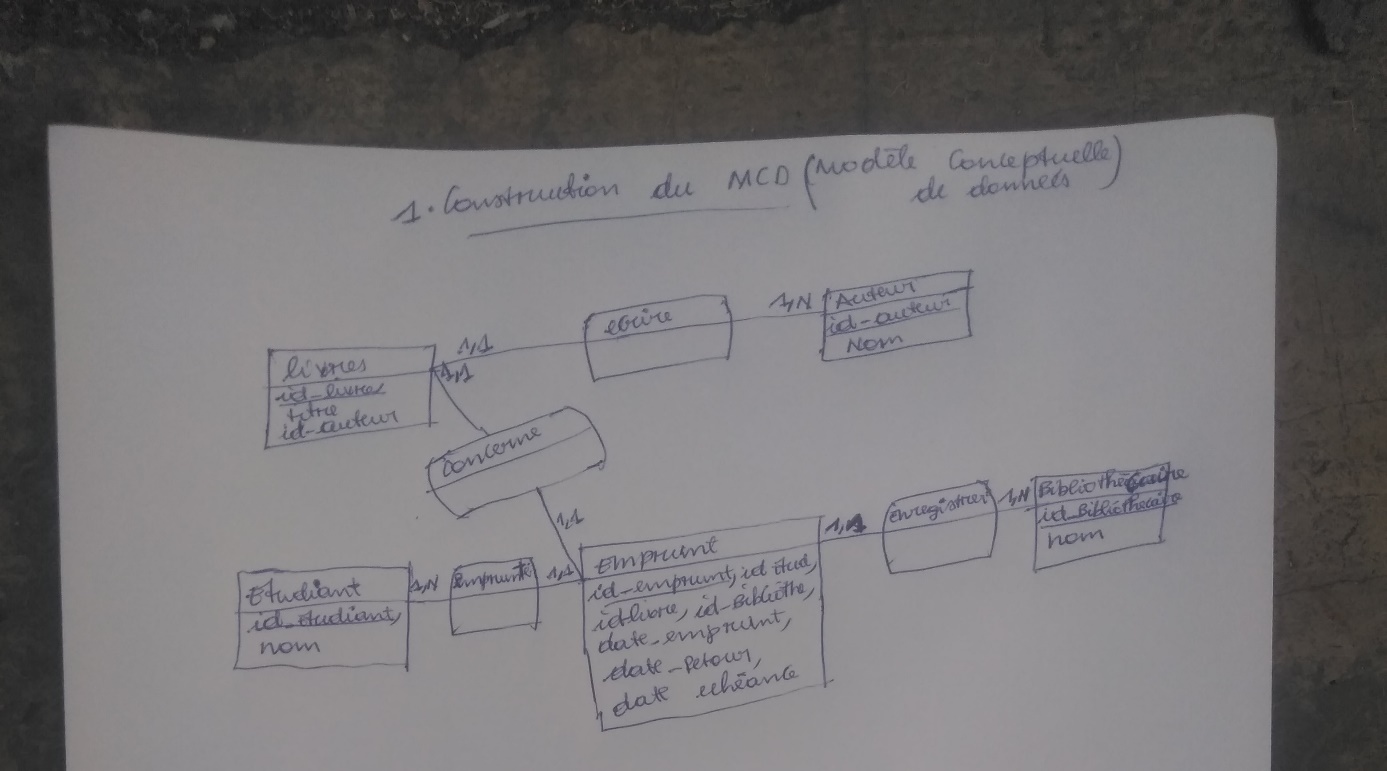
Etudiant

Bibliothecaire

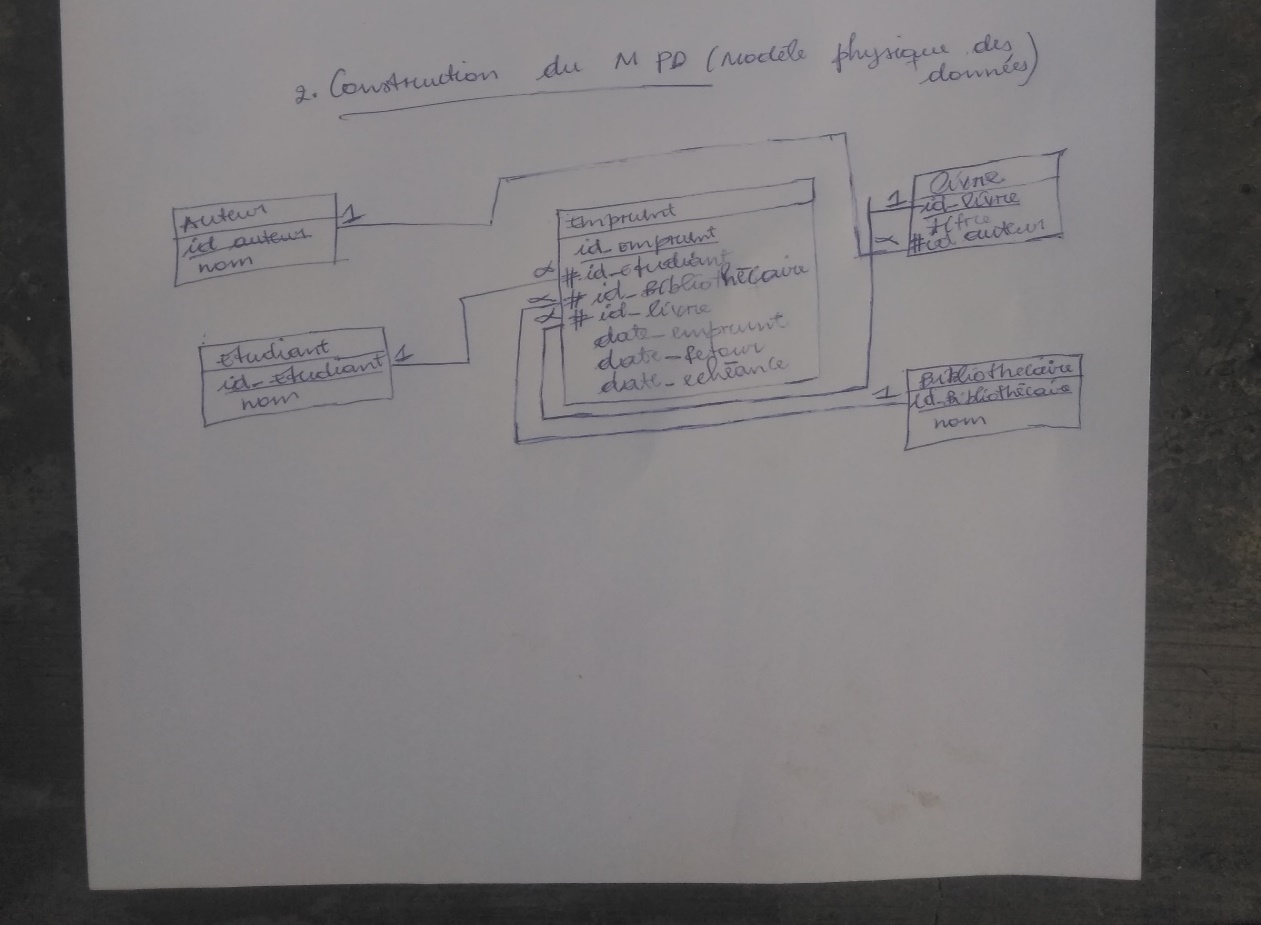
Emprunt

**PARTIE 1 : Conception**

* **MCD/MPD**
* Construction du Modele conceptuel de donnees « MCD »



* Construction du Modele physique de donnees « MPD »



**2. LES INSTRUCTIONS SQL DES CREATIONS**

* Requete de creation de la base de donnees « sqlite3 bibliothecaire .db»

1. Requête de création des tables

**Table auteur**

CREATE TABLE Auteur (

id\_auteur INTEGER PRIMARY KEY,

nom TEXT NOT NULL);

**Table livre**

CREATE TABLE Livre (

id\_livre INTEGER PRIMARY KEY,

titre TEXT NOT NULL,

id\_auteur INTEGER,

FOREIGN KEY(id\_auteur) REFERENCES Auteur(id\_auteur)

);

**TABLE Etudiant**

CREATE TABLE Etudiant (

id\_etudiant INTEGER PRIMARY KEY,

nom TEXT NOT NULL);

**TABLE Bibliothecaire**

CREATE TABLE Bibliothecaire (

id\_bibliothecaire INTEGER PRIMARY KEY,

nom TEXT NOT NULL);

**TABLE EMPRUNT**

CREATE TABLE Emprunt (

id\_emprunt INTEGER PRIMARY KEY,

id\_etudiant INTEGER,

id\_livre INTEGER,

id\_bibliothecaire INTEGER,

date\_emprunt DATE,

date\_retour DATE,

date\_echeance DATE,

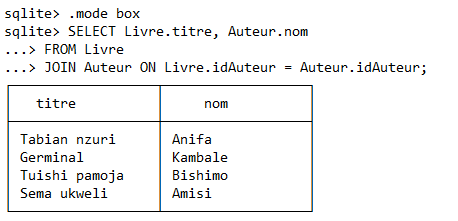
FOREIGN KEY(idetudiant) REFERENCES Etudiant(idetudiant),

FOREIGN KEY(idlivre) REFERENCES Livre(idlivre),

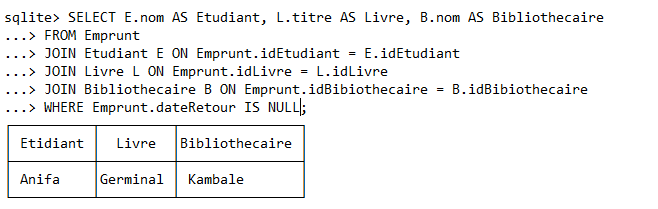
FOREIGN KEY(idbibliothecaire) REFERENCES Bibliothecaire(idbibliothecaire));

**Partie 2**: **REQUETE SQL**

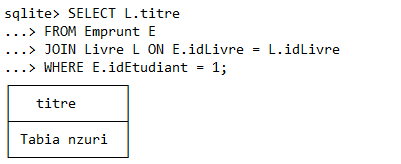
* Afficher la liste de livres avec leurs auteurs



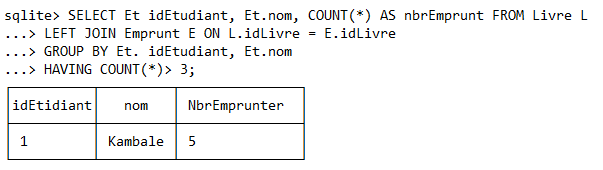
* Lister tous les emprunts en cours avec nom de l’étudiant, titre du livre et nom du bibliothécaire



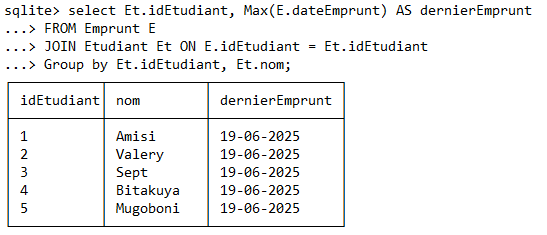
* Afficher les livres emprunter par un étudiant donné



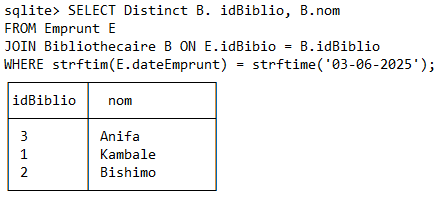
* Lister les étudiants ayant emprunté plus de 3 livres



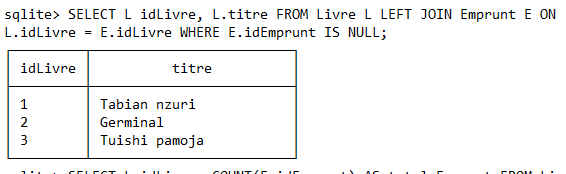
* Afficher la date du dernier emprunt pour chaque étudiant



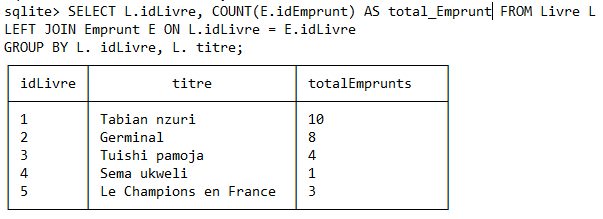
* Lister les bibliothécaires ayant enregistré au moins un emprunt ce mois



* Trouver les livres jamais empruntés

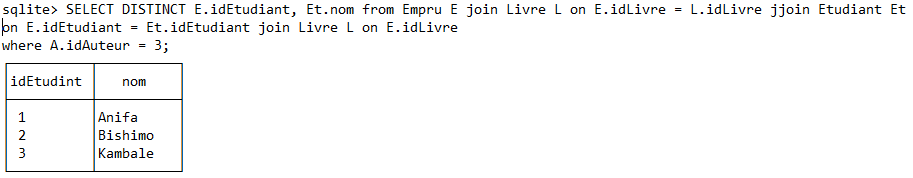


* Afficher pour chaque livre le nombre total d’emprunts



* Lister les étudiants qui ont emprunté des livres d’un auteur spécifique

(ex. id\_auteur = 2)



* Lister les emprunts avec retour en retard (dateretour > dateecheance)

