

REPUBLIQUE TUNISIENNE *** MINISTERE DE L'EDUCATION *** DEVOIR DE CONTROLE N°3		LYCÉE OUED ELLIL Matière : STI Classe:4 SI
DATE : 27-Janvier-2025	DURÉE : 1 heure	ENSEIGNANT: Mr Zouaghi Mohamed
NOM : PRENOM :		NOTE

La présente base de données a pour objectif d'informatiser la gestion des livres et de leurs catégories au sein d'une bibliothèque. L'administrateur chargé de la conception de la base de données de ce système conçoit une première version de la base implémentée et décrite par les huit requêtes SQL suivantes :

<pre> CREATE TABLE auteurs (Id_auteur varchar(4) , Nom_auteur varchar(50) , Pays varchar(30)) ; CREATE TABLE emprunts (Id_membre int(11) , ISBN varchar(13) , Date_emprunt date , Date_retour date) ; CREATE TABLE livres (ISBN varchar(13) , Titre varchar(100) , Annee_publication int(11) , Id_auteur varchar(4) , nomCategorie varchar(50)) ; CREATE TABLE membres (Id_membre int(11) , Nom_membre varchar(50) , Date_inscription date , tel varchar(8) UNIQUE) ; </pre>	<pre> ALTER TABLE auteurs ADD PRIMARY KEY (Id_auteur); ALTER TABLE emprunts ADD PRIMARY KEY (Id_membre,ISBN); ALTER TABLE livres ADD PRIMARY KEY (ISBN); ALTER TABLE membres ADD PRIMARY KEY (Id_membre); ALTER TABLE membres MODIFY Id_membre int(11) AUTO_INCREMENT; ALTER TABLE emprunts ADD CONSTRAINT fk_1 FOREIGN KEY (Id_membre) REFERENCES membres (Id_membre) ON DELETE CASCADE ON UPDATE CASCADE, ADD CONSTRAINT fk_2 FOREIGN KEY (ISBN) REFERENCES livres (ISBN) ON DELETE CASCADE ON UPDATE CASCADE; ALTER TABLE livres ADD CONSTRAINT fk_3 FOREIGN KEY (Id_auteur) REFERENCES auteurs (Id_auteur) ON DELETE CASCADE ON UPDATE CASCADE; </pre>
---	--

Voici une description des colonnes

- **Id_auteur** Identifiant unique de l'auteur, composé de 4 caractères.
- **Nom_auteur** Nom complet de l'auteur, pouvant contenir jusqu'à 50 caractères.
- **Pays** Pays d'origine de l'auteur, pouvant contenir jusqu'à 30 caractères.

- **Date_emprunt**: Date à laquelle le livre a été emprunté.
- **Date_retour**: Date à laquelle le livre doit être retourné.

- **ISBN** : Identifiant du livre
- **Titre**: Titre du livre
- **Annee_publication** : Année de publication du livre
- **nomCategorie**: Nom de la catégorie à laquelle le livre appartient

- **Id_membre**: Identifiant unique du membre
- **Nom_membre**: Nom complet du membre.
- **Date_inscription** : Date à laquelle le membre s'est inscrit à la bibliothèque.
- **tel**: Numéro de téléphone du membre.

1- En s'appuyant sur les requêtes données ci-dessus, et pour chacune des propositions du tableau suivant, répondre par la lettre (V) si la proposition est correcte ou par la lettre (F) dans le cas contraire. Justifier votre réponse par la **clause** de la contrainte correspondante puis citer la contrainte d'intégrité non respectée dans le cas faux. (2.5 points)

Proposition	V / F	Justification par la clause (la partie) de la requête correspondante	Nom de la contrainte non respectée dans la cas Faux
La suppression d'un enregistrement de la table auteurs entraine la suppression automatique des enregistrements qui lui correspondent de la table livres
Il est possible d'insérer un nouvel enregistrement dans la table livres dont id_auteur existe déjà dans cette table auteurs
Il est possible d'insérer un nouvel enregistrement dans la table emprunts dont le ISBN n'existe pas dans la table Livres
Il est possible d'insérer un nouvel enregistrement dans la table membres dont son num tel existe deja dans cette table

2- L'administrateur de la base de données ajoute la table catégories (1.5 points)

```

1 CREATE TABLE categories (
2   Id_categorie varchar(4) ,
3   Nom_categorie varchar(30)
4 ) ;

```

On s'intéresse aux tables **categories** et **livres**, et aux règles de gestion suivantes :

R1 : un livre peut être classé sous plusieurs catégories

R2 : une catégorie peut identifier plusieurs livres

Ecrire la représentation textuelle **de ces deux tables** et toutes autres modifications apportés sur la base de données en prenant en compte les règles **R1** et **R2**

.....

.....

.....

.....

.....

3- Ecrire en SQL la requête qui permet d'ajouter une contrainte à la table **emprunts** pour que la date de retour soit supérieur à la date d'emprunt **(1 point)**

.....
.....
.....

4- Ecrire en SQL les requêtes permettant de respecter cette règle :
R3 : un membre ne peut emprunter le même livre qu'une seule fois par jour **(2 points)**

.....
.....
.....

5- Ajouter à la table **membres** le membre **Tounsi Ali** qui s'est inscrit en **22/01/2025** ayant le numéro de téléphone **99552244** **(1 point)**

.....
.....

6- Donner le rôle de cette requête **(1 point)**

```
SELECT Nom_membre  
FROM membres as M, emprunts as E  
WHERE M.Id_membre = E.Id_membre  
AND Date_emprunt >= ADDDATE(NOW(), INTERVAL -1 MONTH);
```

.....
.....

7- Ecrire les requêtes **SQL** permettant d'afficher : **(9 points)**

a- Afficher les titres des livres ainsi que les noms leurs auteurs par ordre alphabétique des titres.

.....
.....
.....

b- Les titres livres qui n'ont pas été empruntés

.....
.....
.....

c- les noms des membres qui ont emprunté plus que 2 livres

.....
.....
.....

d- les noms des membres qui ont emprunté le livre dont son ISBN= '1234567890123'

.....
.....
.....

e- les auteurs dont leurs livres ont été empruntés

.....
.....
.....

f- les noms des membres qui ont effectué des emprunts au cours du dernier mois

.....
.....
.....

8- l'administration souhaite ajouter le champ **statut** à la table membres qui peut prendre les valeurs 'A', 'I' ou 'S' pour indiquer si le membre est actif, inactif, ou suspendu et qui prend par défaut 'A'.

Donner la requête SQL qui répond à ce besoin : **(1 point)**

.....
.....
.....

9- modifier le statut de « **Tounsi Ali** » en « **Suspendu** » **(1 point)**

.....
.....