

**□ Exercice 1 : Une seule table**

1. Un élève de CPGE souhaite modéliser son travail à faire sous forme d'une liste de tâches. Une tâche peut être de plusieurs types (exercice, DM, révisions, ...), a une description, un état (à faire, en cours ou terminée), une date de création, une date limite d'achèvement et est liée à une matière (maths, physique, ...). Donner le schéma relationnel de cette table.
2. Ecrire les requêtes SQL permettant :
  - a) De lister les tâches en cours.
  - b) D'obtenir toutes les tâches liées à la matière informatique.
  - c) De lister toutes les tâches créées depuis au moins une semaine et pas encore commencées.
  - d) De lister les trois tâches pour lesquelles il reste le moins de temps.
  - e) D'obtenir la tâche dont la date d'achèvement est la plus éloignée dans le temps.

**□ Exercice 2 : Avec plusieurs tables**

1. Un restaurant souhaite modéliser ses réservations. Proposer un schéma relationnel basé sur trois tables : les clients, les tables du restaurant et les réservations. Une réservation est faite pour un client, à une table et pour un jour et une heure donnée. Préciser les clés primaires et étrangères de votre schéma relationnel.
2. Ecrire les requêtes SQL permettant :
  - a) D'obtenir le nombre de client de ce restaurant.
  - b) D'obtenir les dates de toutes les réservations faites par un client spécifique.
  - c) D'obtenir la liste des tables non encore réservés à une date donnée
  - d) De lister les trois tables qui ont été le plus réservé.
  - e) De lister les clients n'ayant jamais fait de réservation.

**□ Exercice 3 : Gestions de randonnées**

 CCINP PC 2025

Les applications de randonnées permettent à un utilisateur de stocker les données de ses randonnées effectuées ou d'avoir accès à celles effectuées par d'autres utilisateurs. Ces données sont stockées dans une base contenant notamment les tables suivantes :

La table **Randonnée** contenant :

- **Id** : entier identifiant de la randonnée.
- **Titre** : chaîne de caractères, titre de la randonnée.
- **Type** : chaîne de caractère du type de la randonnée : "Pied", "VTT", "Cheval".
- **Lieu** : chaîne de caractères, coordonnées GPS du point de départ.
- **Distance** : flottant, longueur en kilomètres de la randonnée.
- **DenP** : entier, dénivelé positif en mètres.
- **DenN** : entier, dénivelé négatif en mètres.
- **Duree** : entier, durée comprise entre 1 (facile) et 5 (difficile).
- **IdAuteur** : entier, identifiant de l'auteur de la randonnée.
- **Trace** : chaîne de caractères, lien internet vers la trace GPX.

La table **Auteur** contenant :

- **Id** : entier, identifiant de l'auteur (le randonneur).
- **Nom** : chaîne de caractères, nom de l'auteur.
- **Prenom** : chaîne de caractères, prénom de l'auteur.
- **Pseudo** : chaîne de caractères, pseudo de l'auteur.
- **Mail** : chaîne de caractères, mail de l'auteur.

1. Expliquer en quoi l'attribut **Titre** ne peut probablement pas être une clé primaire pour la table **Randonnée**. Proposer un attribut de la table **Randonnée** qui puisse être une clé primaire.
2. Identifier un attribut qui soit une clé étrangère de la table **Randonnée**.
3. Ecrire une requête SQL dont l'évaluation renvoie le titre, les coordonnées GPS du point de départ et la longueur des randonnées à pied.

4. Ecrire une requête SQL dont l'évaluation renvoie l'identifiant de l'auteur et son nombre d'activités à pied de niveau 3, classées par ordre décroissant du nombre d'activités de chaque auteur.
5. Ecrire une requête SQL dont l'évaluation renvoie le **Pseudo** de l'auteur et le **Titre** des randonnées stockées dans la base.
6. Ecrire une requête SQL dont l'évaluation renvoie les nom et prénom d'un des auteurs ayant posté le plus de randonnées à cheval.

**□ Exercice 4 : Quelques requêtes SQL**

On considère une base de données des pays et villes du monde, constituée de :

- La table **Pays** (codepays, nom, continent, superficie, population, capitale, monnaie, pib)
- La table **Villes** (codeville, nom, population, #codepays, longitude, latitude)

Le champ codepays de la table Ville fait référence à la clé primaire de la table Pays. Pour chaque question, écrire une requête SQL permettant d'obtenir le résultat :

1. La superficie et la population de la France.
2. Le nombre de pays.
3. La liste alphabétique des pays situés sur le continent américain.
4. Les dix pays les plus peuplés au monde
5. Le nom de la capitale du Togo.
6. La liste des pays dont la capitale est située dans l'hémisphère nord.
7. La liste des pays d'Europe avec leur capitale triée par superficie décroissante.

**□ Exercice 5 : Résultats de requêtes**

On considère le schéma relationnel suivant pour une bibliothèque :

- **Livre** (isbn, titre, #id\_auteur, année, stock)
- **Auteur** (id\_auteur, nom, prenom)
- **Client** (id\_client, nom, prenom, adresse, email, telephone)
- **Emprunt** (id\_emprunt, #id\_client, #isbn, date, retour)

1. Expliquer ce que renvoie chacune des requêtes suivantes :

```
SELECT COUNT(*)
FROM Emprunt
WHERE Emprunt.retour IS NULL;
```

```
SELECT Livre.titre, Emprunt.date, Emprunt.retour
FROM Livre
JOIN Emprunt ON Livre.isbn = Emprunt.isbn
WHERE Emprunt.id_client = 42;
```

```
SELECT Auteur.nom, Auteur.prenom, COUNT(Livre.isbn)
FROM Auteur
JOIN Livre ON Auteur.id_auteur = Livre.id_auteur
GROUP BY Auteur.id_Auteur;
```

2. Ecrire les requêtes permettant de :

- a) Lister les livres dont le stock est inférieur ou égal à 1.
- b) Lister les titres des livres de J. Verne par ordre alphabétique
- c) Lister les noms et email des clients ayant des livres en retard à rendre.
- d) Obtenir le nom de l'auteur ayant écrit le plus de livres.

**□ Exercice 6 : Résultats de requêtes**

Les fabricants de montres offrent la possibilité d'enregistrer les activités dans une base de données afin qu'elles soient accessibles sur PC ou tablettes et qu'elles puissent être partagées avec des amis. La base de données est constituée des tables suivantes :

- Table **activite**

- **Ida** : entier permettant d'identifier l'activité
- **Idm** : entier correspondant à l'identifiant du membre " propriétaire " de l'activité
- **Date** : correspondant à la date (type date) de l'activité
- **Type** : chaîne de caractères correspondant au type d'activité " course ", " marche "...
- **Distance** : entier correspondant à la distance parcourue en mètre de l'activité
- **Temps** : entier correspondant à la durée en secondes de l'activité
- **Fichier** : contient le lien vers le fichier de l'activité

- Table **amis**

- **Id1** : entier, identifiant du lien
- **Membre1** : Idm d'un membre
- **Membre2** : Idm d'un second membre

Il ne peut pas y avoir de doublons dans la table amis : si A est le membre 1 et B le membre 2 d'une relation d'amitiés, la ligne membre 1=B et membre 2=A n'existe pas dans la table.

On considère pour les questions suivantes un membre dont l'identifiant est 1 (Idm = 1).

1. Ecrire une requête SQL permettant de récupérer la liste des identifiants des activités du membre dont l'identifiant est 1.
2. Ecrire une requête SQL permettant de donner la date, la distance parcourue et la vitesse moyenne en km/h des activités de type " course " du membre dont l'identifiant est 1.
3. On donne la requête suivante :

```

SELECT activite.Ida FROM activite
JOIN (SELECT membre1 AS idam1 FROM amis WHERE membre2=1
      UNION
      SELECT membre2 AS idam1 FROM amis WHERE membre1=1 ) AS amis1
ON activite.idm=amis1.idam1
WHERE Type="marche"
    
```

Décrire chaque instruction de la requête et expliquer ce qu'elle renvoie.