

Jour 2: SQL et Python

Job 1

Récupérer votre base de données "LaPlateforme" créée hier. À l'aide du module "mysql-python-connector", connectez-vous à votre base de données "LaPlateforme".

Récupérer l'ensemble des étudiants et afficher le résultat de la requête en console.

Job 2

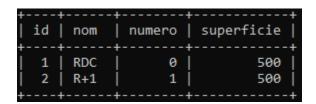
À l'aide de votre terminal SQL, créez les tables suivantes :

- "etage" ayant comme champs :
 - id, int, clé primaire et Auto Incrément
 - nom, varchar de taille 255
 - numero, int
 - superficie, int
- "salles" ayant comme champs :
 - id, int, clé primaire et Auto Incrément
 - nom, varchar de taille 255
 - id_etage, int
 - capacite, int

Job 3

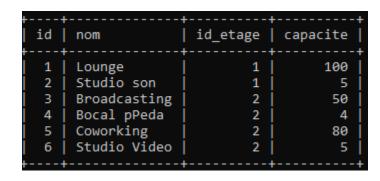
Ajouter les données suivantes a la table "etage" :

- RDC, 0, 500
- R+1, 1, 500



Ajouter les données suivantes a la table "salles" :

- Lounge, 1, 100
- Studio Son, 1, 5
- Broadcasting, 2, 50
- Bocal Peda, 2, 4
- Coworking, 2, 80
- Studio Video, 2, 5



Exporter votre base de données.

Job 4

Écrire un programme qui récupère tous les **noms** et les **capacités** de la table **"salles"** et afficher le résultat en console.

Résultat attendu:

[('Lounge', 100), ('Studio son', 5), ('Broadcasting',
50), ('Bocal pPeda', 4), ('Coworking', 80), ('Studio
Video', 5)]

Job 5

À l'aide d'un requête calculer la superficie de l'ensemble des étages et afficher "La superficie de La Plateforme est de X m2", X étant le résultat de la requête.

Résultat attendu: La superficie de La Plateforme est de 1000 m2

Job 6

À l'aide d'une requête, calculer la somme des capacités des salles et afficher le résultat en console.

Résultat attendu: La capacité de toutes les salles est de : 244

Job 7

Créer une base de données SQL avec une table nommée "employes" contenant les champs suivants :

- id, int, primary key, auto-incrémente
- **nom**, varchar
- **prenom**, varchar
- salaire, decimal
- id_service, int

Insérer des employées dans la table "employes".

Écrire une requête SQL pour récupérer tout les employées dont le salaire est supérieur à 3 000 €. Exécuter la requête et afficher le résultat.

Ajouter la table "services" contenant les champs suivants :

- id, int, primary key, auto-incrémente
- **nom**, varchar

Insérer des services dans votre table.

Récupérer tous les employés et leur service respectif. Afficher le résultat en console.

Créer une classe qui permet d'effectuer différentes opérations (CRUD) sur la table salariée. Vérifier que tout fonctionne correctement.

Job 8

À l'aide du **SQL** et **python**, développer un programme permettant la gestion d'un zoo.

Chaque animal possède un **identifiant** qui l'identifie de façon unique, un **nom**, une **race**, l'**id du type de cage**, une **date de naissance** et un **pays d'origine**.

Une cage peut contenir **un** ou **plusieurs animaux**, mais peu être aussi **vide**. Chaque cage a un **identifiant** unique, une **superficie** et une **capacité** maximum.

À l'aide des informations ci-dessus, créer une base de données nommée **zoo** et créer la table **animal** et la table **cage**.

Votre programme devra:

- Demander au directeur d'ajouter, de supprimer ou de modifier des animaux ou des cages.
- Afficher l'ensemble des animaux présents dans le zoo ainsi la liste des animaux présents dans les cages.
- Calculer la superficie totale de toutes les cages

Rendu

Le projet est à rendre sur https://github.com/prenom-nom/runtrack-bdd-python. Pour chaque jour, créer un dossier nommé "jourXX" et pour chaque job, créer un fichier "jobXX.py" ou XX est le numéro du job.

N'oubliez pas d'envoyer vos modifications dès qu'une étape est avancée ou terminée et utilisez des commentaires explicites.

Pensez à donner les droits sur le répertoire à deepthoughtlaplateforme!

Compétences visées

- SQL
- Utiliser MySQL-python-connector

Base de connaissances

- Base de données et Python
- MySQL-connector-python
- Connexion à une base de données