Bing

La base de données est un système qui permet de stocker et de manipuler des données structurées. Il existe différents types de bases de données, comme les bases de données relationnelles, les bases de données orientées objet, les bases de données NoSQL, etc. SQLite est un exemple de base de données relationnelle, qui utilise le langage SQL pour interroger les données.

Pour utiliser la base de données SQLite en Python, vous pouvez utiliser le module sqlite3, qui fait partie de la bibliothèque standard de Python. Ce module vous permet de créer une connexion à une base de données SQLite, d'exécuter des requêtes SQL, et de récupérer les résultats. Voici quelques étapes pour utiliser le module sqlite3 en Python :

* Importez le module sqlite3 avec la commande `import sqlite3`.
* Créez une connexion à une base de données SQLite avec la méthode `sqlite3.connect(nom\_de\_la\_base\_de\_données)`. Si la base de données n'existe pas, elle sera créée automatiquement. Cette méthode renvoie un objet connexion, qui représente la connexion à la base de données.
* Créez un curseur avec la méthode `connexion.cursor()`. Un curseur est un objet qui permet d'exécuter des requêtes SQL et de parcourir les résultats.
* Exécutez une requête SQL avec la méthode `curseur.execute(requête\_sql)`. Vous pouvez passer des paramètres à la requête SQL avec le second argument de la méthode, sous forme de tuple ou de dictionnaire.
* Récupérez les résultats de la requête SQL avec les méthodes `curseur.fetchone()`, `curseur.fetchmany(n)` ou `curseur.fetchall()`. Ces méthodes renvoient respectivement une seule ligne, n lignes ou toutes les lignes du résultat, sous forme de tuples ou de listes de tuples.
* Validez les modifications dans la base de données avec la méthode `connexion.commit()`. Cette méthode est nécessaire si vous avez effectué des opérations qui modifient les données, comme INSERT, UPDATE ou DELETE.
* Fermez le curseur avec la méthode `curseur.close()`.
* Fermez la connexion avec la méthode `connexion.close()`.

Voici un exemple simple qui crée une table nommée personne avec deux colonnes (nom et âge), insère deux enregistrements dans cette table, et affiche le contenu de la table :

# Importer le module sqlite3

import sqlite3

# Créer une connexion à la base de données test.db

connexion = sqlite3.connect("test.db")

# Créer un curseur

curseur = connexion.cursor()

# Créer une table personne avec deux colonnes (nom et âge)

curseur.execute("CREATE TABLE personne (nom TEXT, âge INTEGER)")

# Insérer deux enregistrements dans la table personne

curseur.execute("INSERT INTO personne (nom, âge) VALUES (?, ?)", ("Alice", 25))

curseur.execute("INSERT INTO personne (nom, âge) VALUES (:nom, :age)", {"nom": "Bob", "age": 30})

# Valider les modifications

connexion.commit()

# Sélectionner tous les enregistrements de la table personne

curseur.execute("SELECT \* FROM personne")

# Afficher le résultat

for ligne in curseur.fetchall():

print(ligne)

# Fermer le curseur

curseur.close()

# Fermer la connexion

connexion.close()

Si vous exécutez ce code Python, vous verrez le résultat suivant :

Output
('Alice', 25)
('Bob', 30)

J'espère que ces informations vous ont été utiles. Si vous avez d'autres questions sur SQLite ou sur la programmation en général, n'hésitez pas à me les poser. Je suis toujours heureux de partager mes connaissances avec vous. 😊

Je suis en train de créer une image pour vous. Elle sera prête dans quelques instants. 🎨