Utiliser les bases de données avec Python

Se connecter à la base de données

Télécharger et installer le connecteur MySQL pour Python à l'adresse http://dev.mysql.com/downloads/connector/python/ (le fichier est dans le dossier partagé Progs)

```
import mysql.connector
serveur = "10.187.36.203"       # nom ou numéro IP du serveur
login = 'btssio'               # login de connexion
pwd = 'btssio'                # Mot de passe pour la connexion
base = 'glpi'                  #nom de la base de données

#Connexion a la base
bdd=mysql.connector.connect(user=login, password=pwd, host=serveur, database=base)
```

Gérer les exceptions

S'il y a une **erreur** à la connexion (serveur indisponible, erreur de login ou de mot de passe, base de données inconnue), le programme python génère une **exception** et **s'arrête**.

Il est donc pertinent de gérer ces erreurs et de gérer quelle erreur est survenue **sans arrêter l'exécution** du programme python.

Pour cela, il faut mettre dans un bloc **try** les instructions à tester et dans **except** les instructions à exécuter en cas d'erreur.

Syntaxe de base :

```
try:
   # Bloc à essayer
except:
    # Bloc qui sera exécuté en cas d'erreur
```

Mise en œuvre

```
#Connexion a la base
try:
    connexion=mysql.connector.connect(user=login, password=pwd, host=serveur, database=base)
except mysql.connector.Error as err:
    if err.errno == mysql.connector.errorcode.ER_DBACCESS_DENIED_ERROR:
        print("Base inconnue")
    elif err.errno == mysql.connector.errorcode.ER_ACCESS_DENIED_ERROR:
        print("Erreur login ou pwd")
    else:
        print(err)

print("connexion réussie")
```

Le bloc **except** permet de gérer précisément le **type d'erreur** en récupérant le **numéro d'erreur** puis en l'utilisant pour préciser l'erreur. Il faut dans ce cas savoir quels sont les numéros d'erreur possibles.

Pour le savoir consultez l'aide du module.

Aide sur le module :

```
help ('mysql.connector')
```

Aide sur les erreurs du module

```
help ('mysql.connector.errorcode')
```

Exécuter une requête

L'exécution d'une requête qui renvoie plusieurs tuples (lignes, enregistrement) nécessite d'utiliser un **curseur**.

```
cursor=connexion.cursor()       #Definition du curseur
requeteSQL= "Select * from eleve"   # Definition de la requête
cursor.execute(requeteSQL)       # Execution de la requête
resultat = cursor.fetchall()        #Recuperation des donnees
```

Si votre requête ne retourne aucun résultat, ce qui est le cas d'une requête de mise à jour (**update**) ou de création d'enregistrement (**insert**), il n'est pas nécessaire d'utiliser la méthode fetchall().

Utilisation des données

Une **boucle** est nécessaire pour récupérer les données ligne par ligne.

```
for row in resultat:
    print(row[0]) # champ 1 du select
    print(row[1]) # champ 2  du select
    print(row[2]) # champ 3  du select
```

Fermeture de la connexion

```
cursor.close() #Fermeture du curseur
connexion.close()  #Deconnexion de la base
```

Projet à réaliser

Modifiez votre programme Python **WOL** afin de récupérer dans la base de données de **GLPI** l'**adresse MAC** d'un ordinateur à partir de son **nom**. De cette manière, il suffira de saisir le nom du poste dans votre programme Python pour récupérer dans la base GLPI, son adresse MAC.

Les tables **GLPI** à utiliser sont ***glpicomputers (liste des ordinateurs) et glpicomputersdevicenetworkcards (choisissez l'adresse MAC de la carte réelle - physique) ===== Utiliser une classe pour simplifier le code ===== La classe de base d'accès aux données : <code python> class connexionBD: “””Connexion à la base de donnee””” def init(self): # Création d'une connexion à la base import mysql.connector self.serveur = “10.187.36.203” self.login = “btssio” self.pwd = “btssio” self.base = “glpi” self.connection = mysql.connector.connect(user=self.login, password=self.pwd, host=self.serveur, database=self.base) self.cursor = self.connection.cursor() def close(self): #! Ferme la connexion self.cursor.close() self.connection.close() def execute(self, requete): #! Execute la requete MySQL et retourne le resultat self.cursor.execute(requete) return self.__cursor.fetchall() </code> ===== Insérer des données ===== Exemple : créer un enregistrement élève : <code python> connexion = connexionBD() requete = “INSERT INTO eleve (nom, prenom) VALUES('Dupond','Raymond')” # La requete est toujours un string connexion.execute(requete) connexion.close() </code> ===== Modifier des donnés ===== Exemple : modifier le prénom : <code python> connexion = connexionBD() requete = “UPDATE eleve SET prenom='Paul' WHERE nom='Dupond' connexion.execute(requete) connexion.close() </code> ===== Lire des données ===== Exemple : Liste des élèves : <code python> connexion = connexionBD() resultat = connexion.execute(“SELECT * FROM eleve”) i = 1 for row in resultat: print(“enregistrement ” + str(i)) for champ in row: # Ecrit tous les champs de chaque ligne print (“Champ :” + str(champ)) i+=1 connexion.close() </code>***