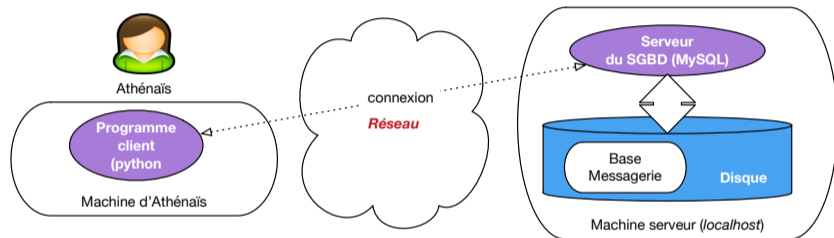


Cours de bases de données,
<http://sql.bdpedia.fr>

Programmation avec la base “Messagerie”

Programmation (Python) la base “Messagerie”

La base Messagerie, gérée par un SGBD (disons, MySQL)



Notre utilisatrice est Athénaïs : elle va écrire quelques scripts Python pour exécuter ses requêtes.

Connectons-nous

Une application doit toujours établir une connexion avec le serveur.

```
connexion = pymysql.connect
            ('localhost',
             'athénaïs',
             'motdepasse',
             'Messagerie',
             cursorclass=pymysql.cursors.DictCursor)
```

Au moins 4 paramètres : machine serveur, nom et mot de passe, nom de la base.

Et souvent des options.

Les curseurs

La connexion nous permet de créer des **curseurs**

```
curseur = connexion.cursor()
```

Les curseurs permettent d'exécuter des requêtes.

```
curseur.execute("select * from Contact")
```

Et de récupérer le résultat

```
for contact in curseur.fetchall():  
    print(contact['prénom'], contact['nom'])
```

Le programme complet

```
import pymysql
import pymysql.cursors

connexion = pymysql.connect('localhost', 'athénaïs',
                           'motdepasse', 'Messagerie',
                           cursorclass=pymysql.cursors.DictCursor)

curseur = connexion.cursor()
curseur.execute("select * from Contact")

for contact in curseur.fetchall():
    print(contact['prénom'], contact['nom'])
```

Important : il faudrait le rendre robuste... À vous de jouer

Une transaction d'envoi des messages

```
# Tous les messages non envoyés
messages = connexion.cursor()
messages.execute("select * from Message where dateEnvoi is null")
for message in messages.fetchall():
    # Ici il faudrait tamponner le message !

    # Ici on envoie les messages à tous les destinataires
    envois = connexion.cursor()
    envois.execute("select * from Envoi as e, Contact as c "
                  +" where e.idDestinataire=c.idContact "
                  + "and e.idMessage = %s", message['idMessage'])
    for envoi in envois.fetchall():
        mail (envoi['email'], message['contenu'])
```

Tamponnage du message

```
# Tous les messages non envoyés
messages = connexion.cursor()
messages.execute("select * from Message where dateEnvoi is null")
for message in messages.fetchall():
    # Marquage du message
    connexion.begin()
    maj = connexion.cursor()
    maj.execute ("Update Message set dateEnvoi='2018-12-31' "
                + "where idMessage=%s", message['idMessage'])

# Ici on envoie les messages à tous les destinataires

# Ici on valide
connexion.commit()
```

À retenir

Transposition en Python des principes de programmation déjà vus pendant la semaine “Programmation”

- Tout se passe par des curseurs
- Séquence systématique exécution-boucle-fermeture (implicite)
- Un langage interprété rend la conversion plus simple

Prochaine question : mon application est-elle à l’abri de tout problème de concurrence ?