

PostgreSQL + Java

Téléchargez le fichier .jar dans la rubrique Download du site dédié:<http://jdbc.postgresql.org>

Pour utiliser le driver → `Class.forName("org.postgresql.Driver");`

il faut qu'une connexion soit effective afin que le programme et la base de données puissent communiquer → `Connection conn = DriverManager.getConnection(url, user, passwd);`

l'URL de connexion ;

le nom de l'utilisateur ;

le mot de passe utilisateur.

L'URL de connexion est indispensable à Java pour se connecter à n'importe quelle BDD.

La figure suivante illustre la manière dont se décompose cette URL.

jdbc:postgresql://localhost:5432/ Passagers Info

URL de connexion à une BDD via JDBC

** ici, nous travaillons en local, nous utilisons donc `//localhost:5432`. En effet, le nom de la machine physique est suivi du numéro de port utilisé.

PostgreSQL+ Java

- création de l'objet Statement ;
- exécution de la requête SQL ;
- récupération et affichage des données via l'objet ResultSet ;
- fermeture des objets utilisés (bien que non obligatoire, c'est recommandé).

Statement - ResultSet

Ces deux objets qui permettent de récupérer des données de la BDD et de travailler avec celles-ci.

L'objet **Statement** permet d'exécuter des instructions SQL, il interroge la base de données et retourne les résultats. Ensuite, ces résultats sont stockés dans l'objet ResultSet, grâce auquel on peut parcourir les lignes de résultats et les afficher. L'objet Statement permet d'exécuter des requêtes SQL. Ces dernières peuvent être de différents types :

- CREATE ;
- INSERT ;
- UPDATE ;
- SELECT ;
- DELETE.;

PostgreSQL+ Java

L'objet `Statement` est fourni par l'objet `Connection` grâce à l'instruction `conn.createStatement()`.

Ce que j'ai fait, ensuite, c'est demander à mon objet `Statement` d'exécuter une requête SQL de type `SELECT` :

```
SELECT * FROM Passagers_Info.
```

Elle demande à la BDD de nous envoyer toutes les passagers.

PostgreSQL+Java

Il existe une méthode `getXXX()` par type primitif ainsi que quelques autres correspondant aux types SQL :

```
getArray(int columnIndex) ;  
getAscii(int columnIndex) ;  
getBigDecimal(int columnIndex) ;  
getBinary(int columnIndex) ;  
getBlob(int columnIndex) ;  
getBoolean(int columnIndex) ;  
getBytes(int columnIndex) ;  
getCharacter(int columnIndex) ;  
getDate(int columnIndex) ;  
getDouble(int columnIndex) ;  
getFloat(int columnIndex) ;  
getInt(int columnIndex) ;  
getLong(int columnIndex) ;  
getObject(int columnIndex) ;  
getString(int columnIndex).
```

Pour fermer mes objets à l'aide des instructions `result.close()` et `state.close()`.

Exercice 4 - Gestion d'une salle de gym

- chaque client doit indiquer nom, âge, adresse et occupation ;
- chaque client souscrit un abonnement (mensuel ou annuel) ;
- chaque abonnement permet l'accès à la salle de poids et au sauna, alors que la piscine et le mur d'escalade sont facultatifs ;
- chaque exercice du programme est associé à une quantité de répétitions et à une fréquence ;
- le programme peut être organisé sur une séance ou sur deux ;

.

Créer les tables pour cette BDD.

Entrer ensuite 3 données dans chaque table avec la commande INSERT et afficher les tables avec la commande SELECT.

Exercice 5 – Restaurant

- un restaurant est caractérisé par le nom et par le type de cuisine (française, chinoise, indienne, japonaise, . . .) ;
- un restaurant est composé de plusieurs salles, chacune caractérisée par ses dimensions (longueur et largeur) ;
- il y a trois types de salles : cuisine, salle de bain, salle de restauration ;
- la cuisine (il y en a une seule) est caractérisée par la liste d’outils qu’elle contient ;
- une salle de bain est caractérisée par son type (hommes, femmes, privé) ;
- une salle de restauration est caractérisée par le nombre de tables et par le nombre de places qu’elle contient ;
- au restaurant il y a un propriétaire et plusieurs employés ;
- les employés et les propriétaires sont des personnes ;
- pour chaque personne on enregistre la date de naissance, la résidence, le code de s’sécurité sociale et le numéro de telephone ;

Créer les tables pour cette BDD.

Entrer ensuite 3 données dans chaque table avec la commande INSERT et afficher les tables avec la commande SELECT.

Exercice 6 – Service de livraison de commandes a la maison d'une pizzeria

- les clients doivent indiquer leur adresse, etage, porte et numero de telephone ;
- les clients peuvent commander des pizzas, des boissons et des frites ;
- une pizza est caractérisée par un nom et un prix ;
- une boisson est caractérisée par un nom, une taille et un prix ;
- des frites sont caractérisées par une taille et un prix ;
- pour chaque commande effectuée les clients doivent indiquer l'heure de livraison souhaitée ;

Créer les tables pour cette BDD.

Entrer ensuite 3 données dans chaque table avec la commande INSERT et afficher les tables avec la commande SELECT.

Exercice 7 - Ventes de bureaux et d'appartements dans de nouveaux batiments.

- Un bâtiment a plusieurs etages , parking
- un etage il peut y avoir des bureaux et des appartements.
- Le parking se compose de plusieurs places numérotées, chacune appartenant a un bureau ou a un appartement.

Créer les tables pour cette BDD.

Entrer ensuite 3 données dans chaque table avec la commande INSERT et afficher les tables avec la commande SELECT.