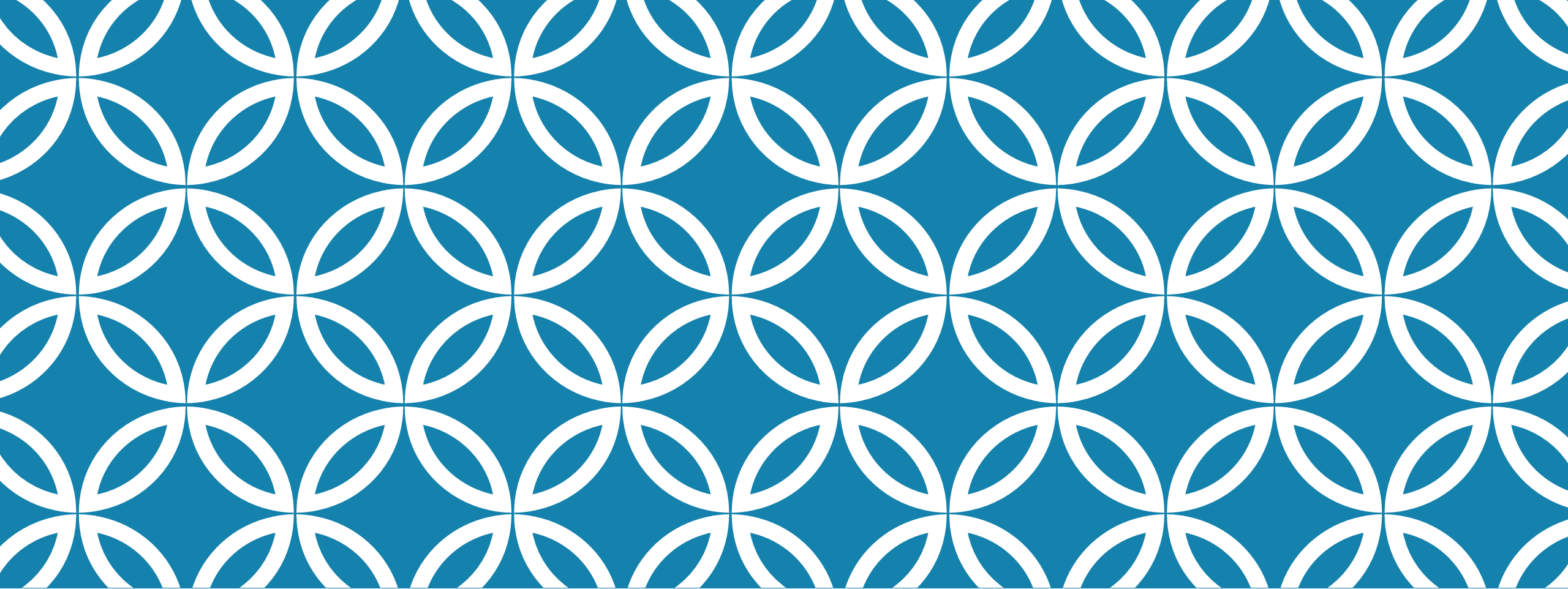


FORMATION JDBC

Jordan ABID





ACCÈS AUX BASES DE DONNÉES VIA JDBC

PILOTES JDBC

Pour qu'une application java puisse communiquer avec un serveur de bases de données, elle a besoin d'utiliser les pilotes JDBC (Java Data Base Connectivity)

- Les Pilotes JDBC est une bibliothèque de classes java qui permet, à une application java, de communiquer avec un SGBD via le réseau en utilisant le protocole TCP/IP
- Chaque SGBD possède ses propres pilotes JDBC.
 - Il existe un pilote particulier « JdbcOdbcDriver » qui permet à une application java de communiquer avec n'importe quelle source de données via les pilotes ODBC (Open Data Base Connectivity)
- Les pilotes ODBC permettent à une application Windows de communiquer avec une base de données quelconque (Access, Excel, MySQL, Oracle, SQL SERVER etc...)
- La bibliothèque JDBC a été conçu comme interface pour l'exécution de requêtes SQL.

CRÉER UNE APPLICATION JDBC

Pour créer une application manipulant une base de données, il faut suivre les étapes suivantes :

- Chargement du Pilote JDBC
- Identification de la source de données
- Création d'un objet **Connection**
- Création d'un objet Instruction **Statement** (ou **PreparedStatement**)
- Exécution d'une requête à l'aide de l'objet **Statement**
 - Récupération de données à partir de l'objet renvoyé **ResultSet** ;
- Fermeture de l'objet **ResultSet** ;
- Fermeture de l'objet **Statement** ;
- Fermeture de l'objet **Connection**.

DÉMARCHE JDBC

Charger les pilotes JDBC

Utiliser la méthode `forName` de la classe `Class`, en précisant le nom de la classe pilote.

- Pour charger le pilote `JdbcOdbcDriver`: **`Class.forName("sun.jdbc.odbc.JdbcOdbcDriver") ;`**
- Pour charger le pilote `jdbc` de `MySQL`: **`Class.forName("com.mysql.jdbc.Driver") ;`**

CRÉER UNE CONNEXION

Pour créer une connexion à une base de données, il faut utiliser la méthode statique `getConnection()` de la classe `DriverManager`.

Pour un pilote `com.mysql.jdbc.Driver` :

```
Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/DB", "user", "pass" );
```

Pour un pilote `sun.jdbc.odbc.JdbcOdbcDriver` :

```
Connection conn=DriverManager.getConnection("jdbc:odbc:dsnSCO", "user", "pass" );
```

OBJETS STATEMENT ET RESULTSET

Pour exécuter une requête SQL, on peut créer l'objet Statement en utilisant la méthode `createStatement()` de l'objet Connection.

```
Statement st=conn.createStatement();
```

Pour exécuter une requête SQL de type select, on peut utiliser la méthode `executeQuery()`

Cette méthode exécute la requête et stocke le résultat de la requête dans l'objet ResultSet :

```
ResultSet rs=st.executeQuery("select * from PRODUITS");
```

Pour exécuter une requête SQL de type insert, update et delete on peut utiliser la méthode `executeUpdate()`

```
st.executeUpdate("insert into PRODUITS (...) values(...)");
```

RÉCUPÉRER LES DONNÉES D'UN RESULTSET

Pour parcourir un ResultSet, on utilise sa méthode `next()` qui permet de passer d'une ligne à l'autre. Si la ligne suivante existe, la méthode `next()` retourne `true`.

Pour récupérer la valeur d'une colonne de la ligne courante du ResultSet, on peut utiliser les méthodes :

`getInt(colonne)`, `getString(colonne)`, `getFloat(colonne)`, `getDouble(colonne)`, `getDate(colonne)`, etc...

colonne représente le numéro ou le nom de la colonne de la ligne courante

```
while(rs.next())
{
    System.out.println(rs.getInt(1));
    System.out.println(rs.getString("NOM_PROD"));
    System.out.println(rs.getDouble("PRIX"));
}
```


OBJET PREPAREDSTATEMENT

Pour exécuter une requête SQL, on peut également créer l'objet PreparedStatement en utilisant la méthode `prepareStatement()` de l'objet `Connection`.

```
PreparedStatement ps=conn.prepareStatement("select * from PRODUITS where NOM_PROD like ? AND PRIX<?");
```

Définir les valeurs des paramètres de la requête:

```
ps.setString(1,"%"+motCle+"%");  
ps.setString(2, p);
```

Exécution d'une requête SQL avec l'objet PreparedStatement :

```
ResultSet rs=ps.executeQuery();  
ps.executeUpdate();
```