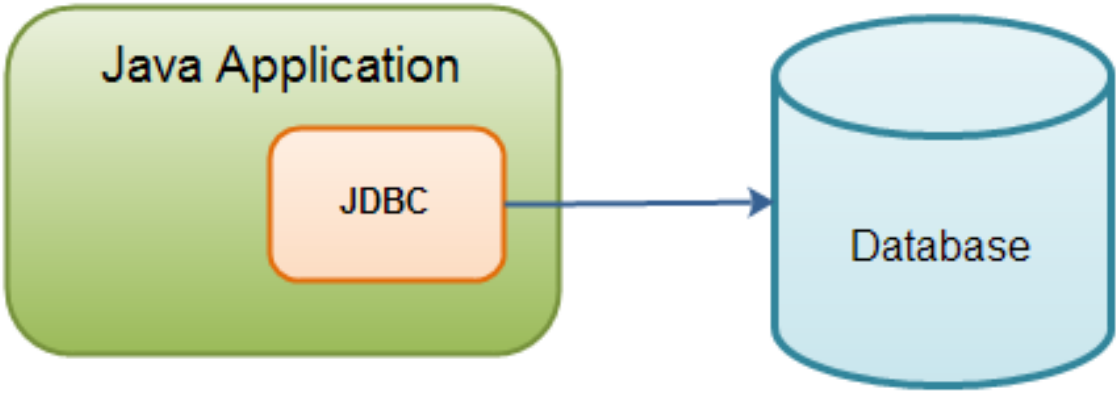
Développement en JAVA

Connexion aux base de données

JDBC

- **J**ava **D**ata **B**ase **C**onnection
- **API** qui standardise l'accès à une base de données
- Une implémentation (Driver) existe pour chaque produit du marché (MySQL, PostgreSQL, Oracle…). L'interface est standard.



JDBC

- Il existe différents type de driver
- JDBC Offre une API simple permettant de requêter une base de données
- Gestion du pooling
- Gestion des transactions

Configuration de JDBC

- JDBC est distribué sous la forme d'une library externe qu'il faut ajouter dans le buildPath du projet
- Afin de vérifier que le jar JDBC est bien présent dans notre projet, on peut forcer le chargement du Driver de la manière suivante:

```
Class.forName("com.mysql.jdbc.Driver").newInstance();
System.out.println("JDBC driver successfuly loaded");
```

La classe DriverManager

- La classe `java.sql.DriverManager` s'occupe de gérer la configuration du driver JDBC
- La méthode `getConnection` permet d'ouvrir une connexion avec la BDD, elle attend en paramètre:
  - L'URL de la base
  - L'utilisateur
  - Le mot de passe
- L'object `Connection` obtenu doit **toujours être fermé** à la fin du programme avec la méthode `close()`

La classe DriverManager

```
Connection conn = null;
try{
 conn = DriverManager.getConnection(
"jdbc:mysql://localhost:3306/test", "root", "");
}catch(Exception e){
  // Gestion des exceptions
}finally{
   // Fermeture de la connexion dans tous les cas
  if(conn != null){
    conn.close();
  }
}
```

La classe Statement

- Un objet de classe `java.sql.Statement` est capable d'exécuter un requête sur la base de données
- Elle possède notamment la méthode `executeQuery(String)` qui est permet de lancer directement une commande SQL

```
Statement stmt = conn.createStatement();
ResultSet result = stmt.executeQuery("SELECT * FROM items");
```

La classe ResultSet

- La classe `java.sql.ResultSet` contient les résultats de la requête
- Elle se parcours à l'aide comme un Iterator à l'aide de la méthode `next()` qui permet d'avancer à la ligne suivante
- On peut récupérer les valeurs en précisant le nom de la colonne ou son indice dans la réponse

```
ResultSet result = stmt.executeQuery("SELECT * FROM items");

// Affichage de toutes les lignes de la table items
while(result.next()){
  System.out.print(result.getInt(1) + result.getString(3) );
  System.out.print(result.getInt("id") + result.getString("name"));
}
```