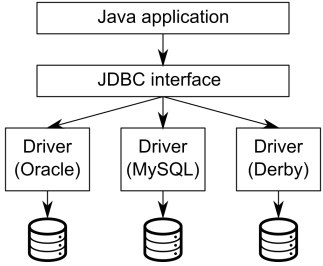
## JDBC v podnikových aplikacích: popis JDBC obecně, popis dat datového zdroje (DataSource), nastavení datových zdrojů v aplikačním serveru (JDBC Connection Pool, JDBC Resources). [BI-TJV]

Java Database Connectivity (JDBC)

- Java **rozhraní pro připojení k relační databázi**
- Umožňuje vytvořit **spojení (connection)** s databází
- Přes toto spojení lze pak spouštět dotazy SELECT, CREATE, INSERT, UPDATE a DELETE.
- **JDBC je pouze obecné rozhraní**. Pro připojení ke konkrétnímu typu databáze (MySQL, PostgreSQL....) je potřeba **driver**.




Existují dvě možnosti jak se připojit k databázi přes JDBC.

1. Použitím třídy **DriverManager**: hodí se pro jednoduché aplikace
2. Použitím rozhraní **DataSource**: obecnější přístup vhodný pro větší aplikace, které potřebují pokročilejší techniky jako recyklování připojení apod.

DataSource vs DriverManager:

- Detaily připojení nemusí být napevno v aplikaci
- Nemusime byt v try bloku

```
try (Connection conn = DriverManager.getConnection(
     "jdbc:mysql://localhost/mydb",
     "myLogin",
     "myPassword")) {

  // zde používáme vytvořené spojení

}  // Java VM se postará o uzavření spojení
```

JDBC používá:

- Statement (metoda createStatement)
- PreparedStatement (prepareStatement)
    - Pokud chceme nějaký dotaz spouštět opakovaně jen s malými změnami parametrů.
    - ```
      PreparedStatement ps = conn.prepareStatement(
            "SELECT * FROM person WHERE name = ? AND age = ?" )
      ```
    - ```
      ResultSet rs = ps.executeQuery()
      ```

JDBC vrací výsledek dotazu jako implementaci rozhranní **ResultSet .**
Po jednotlivých řádcích výsledku mužeme iterovat.

## JDBC Connection pooling

Problém - pro každé vykonání dotazu je třeba se k databázi připojit. Připojení stojí nějaký čas a není dobré se neustále připojovat a odpojovat.

Connection pooling zajišťuje, že připojení proběhne na začátku a zachová se i po vykonání dotazu. Pro další dotaz je **existující připojení znovupoužito**.

Nastavení Connection poolu se v TJV dělalo přes Glassfish. Na localhost webu se zadaly přihlašovací údaje do databáze, adresa databáze atp.
**Nakonfiguruje:** user, password, url, název databáze, počet připojení

## JDBC Resource

Poskytuje prostředky pro připojení k databázi. Pro vytvoření je potřeba speficikovat Connection pool, se kterým bude spjat. Má vlastní jméno (JNDI name), které dle konvence začíná “jdbc/”.

Přístup ke zdroji lze získat přes `InitialContext`:

```
InitialContext ic = new InitialContext();
DataSource source = (DataSource) ic.lookup("jdbc/my_resouce_name");
```