

MySQL mit Maven

1 Dependency

Maven-Repository: <https://mvnrepository.com/artifact/mysql/mysql-connector-java>

```
<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>8.0.18</version>
</dependency>
```

2 Namensraum

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
```

3 Connection (Verbindung mit Datenbank)

```
String databaseName = "MeineDB";
String user = "root";
String password = "mysql";

String dbString = "jdbc:mysql://localhost:3306/" +
    databaseName +
    "?useUnicode=true&" +
    "useJDBCCompliantTimezoneShift=true&" +
    "useLegacyDatetimeCode=false&" +
    "serverTimezone=UTC";

Connection connection = DriverManager.getConnection(dbString, user, password);
```

4 SQL-Anweisung mit Parametern

4.1 Datensätze lesen

```
String sql = "SELECT * FROM PERSON WHERE Vorname = ?";
```

4.2 Datensätze neu, ändern, löschen

```
String sql = "INSERT INTO Tabelle(Vorname, Datum) VALUES(?, ?)";
// String sql = "UPDATE Tabelle SET Datum = ? WHERE Vorname = ?";
// String sql = "DELETE FROM Tabelle WHERE Vorname = ?";
```

5 PreparedStatement

```
PreparedStatement statement = connection.prepareStatement(sql);
```

6 Argumente an Parameter übergeben

```
statement.setString(1, "Horst");
statement.setString(2, "1980-01-30");
```

7 SQL-Anweisung ausführen

7.1 Datensätze lesen

```
ResultSet rs = statement.executeQuery();
while(rs.next()){
    ... = rs.getString("Vorname");
    ... = rs.getDate("Datum");
}
rs.close();
```

7.2 Datensätze neu, ändern, löschen

```
statement.executeUpdate();
```

8 Alles schließen

```
statement.close();
connection.close();
```