



# Java Programmierung

Datenbankverbindung mit ***Eclipse*** und ***MySQL***



# Themenübersicht

**01**

**XAMPP**

- 1. Download und Installation

**02**

**MySQL Connector  
und Eclipse**

- 2.1 Download
- 2.2 Einrichten der Entwicklungsumgebung

**03**

**PHP My Admin  
und Eclipse**

- 3.1 Datenbank in PHP My Admin erstellen
- 3.2 Datenbankabfrage in Java



# 01

## XAMPP

Anwendung zum Betreiben eines Webserver mit MySQL und PHP My Admin ←

*Diesen Schritt können Sie überspringen, wenn Sie bereits XAMPP installiert haben*



 **XAMPP** Apache + MariaDB + PHP + Perl

---

### Was ist XAMPP?

XAMPP ist die beliebteste PHP-Entwicklungsumgebung

XAMPP ist eine vollständig kostenlose, leicht zu installierende Apache-Distribution, die MariaDB, PHP und Perl enthält. Das XAMPP Open-Source-Paket wurde für eine extrem einfache Installation und Nutzung eingerichtet.

  
**XAMPP**

**Herunterladen**  
Klicken Sie hier für weitere Versionen

 **XAMPP für Windows**  
8.0.10 (PHP 8.0.10)

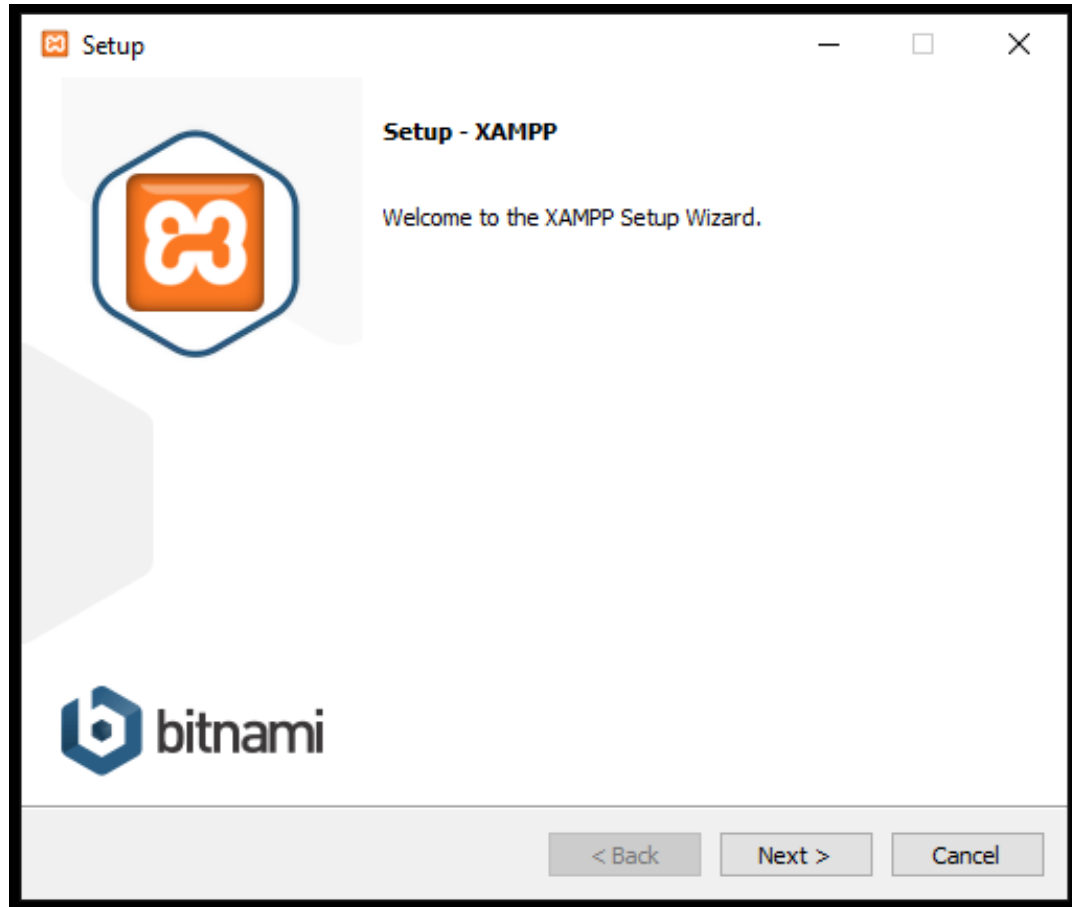
 **XAMPP für Linux**  
8.0.10 (PHP 8.0.10)

 **XAMPP für OS X**  
8.0.10 (PHP 8.0.10)

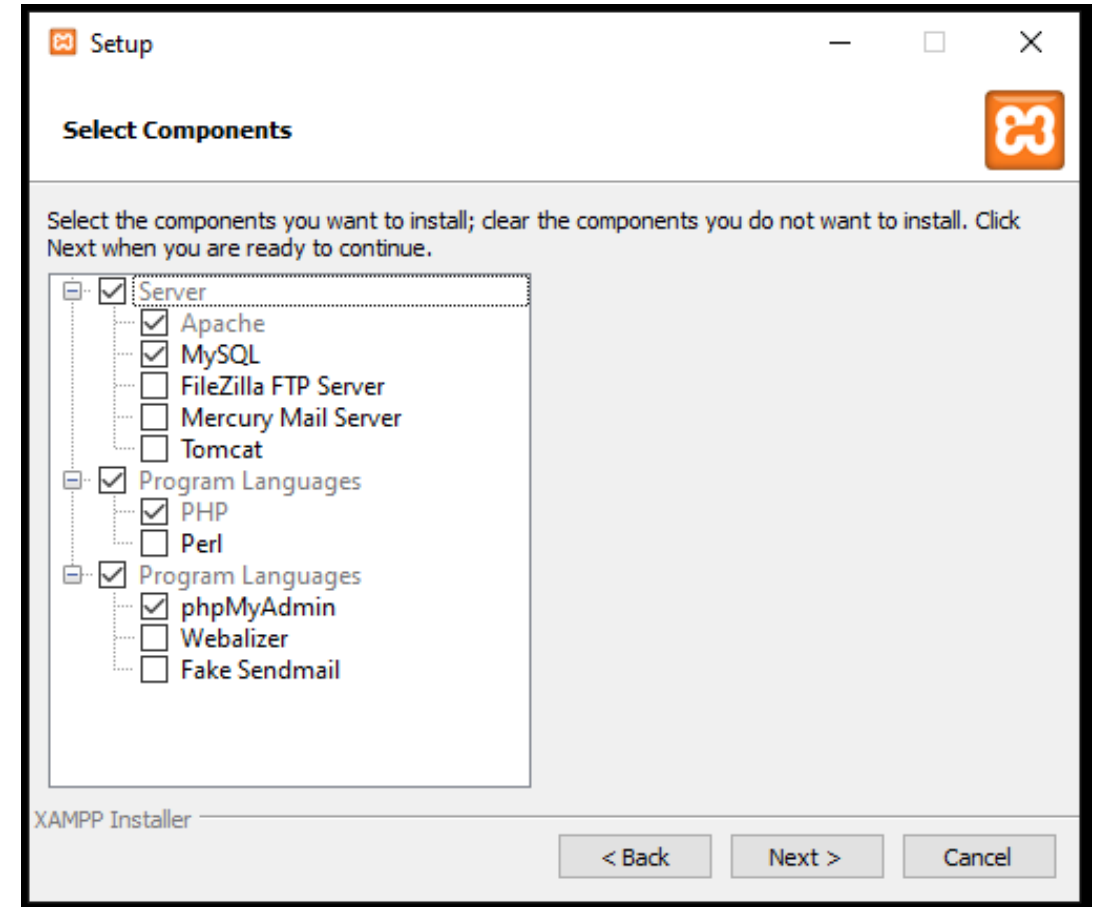
Unter folgendem Link können Sie XAMPP herunterladen.

<https://www.apachefriends.org/de/index.html>

## Installation XAMPP

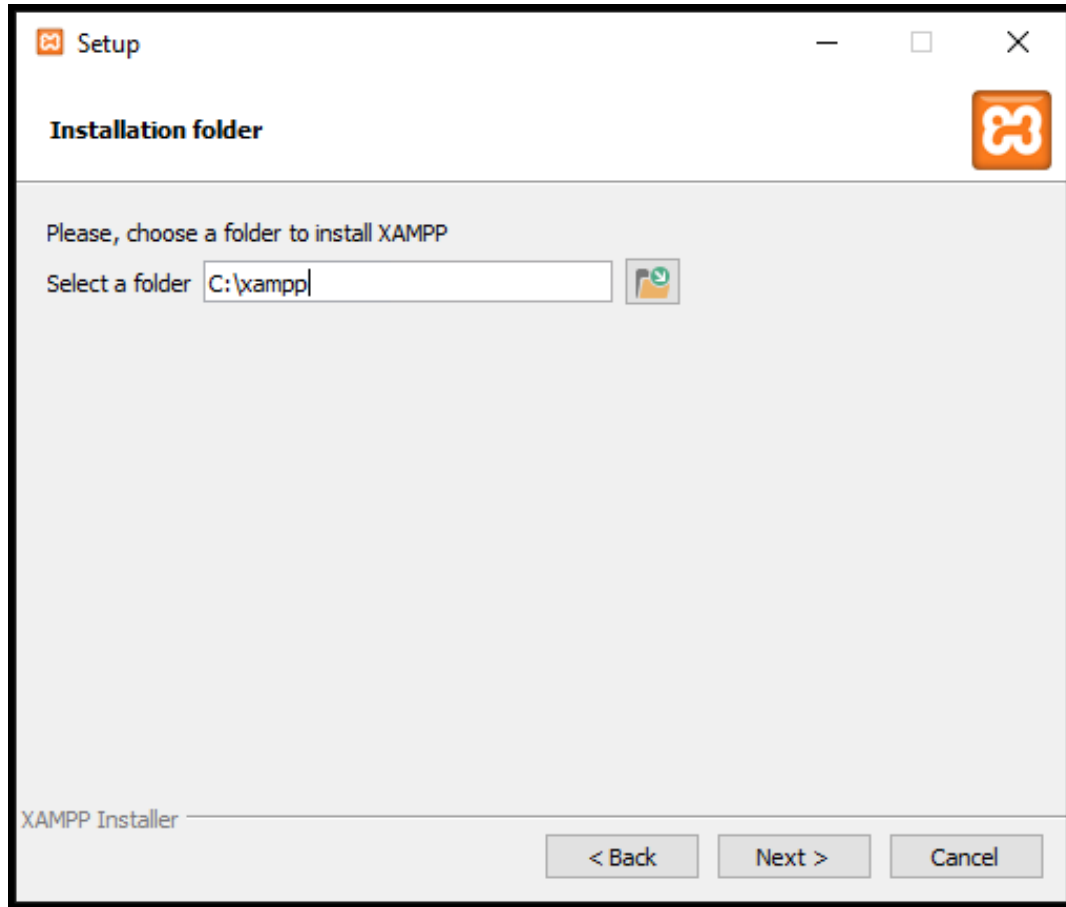


*Starten Sie die Installation durch Doppelklick auf die heruntergeladene Setup Anwendung*

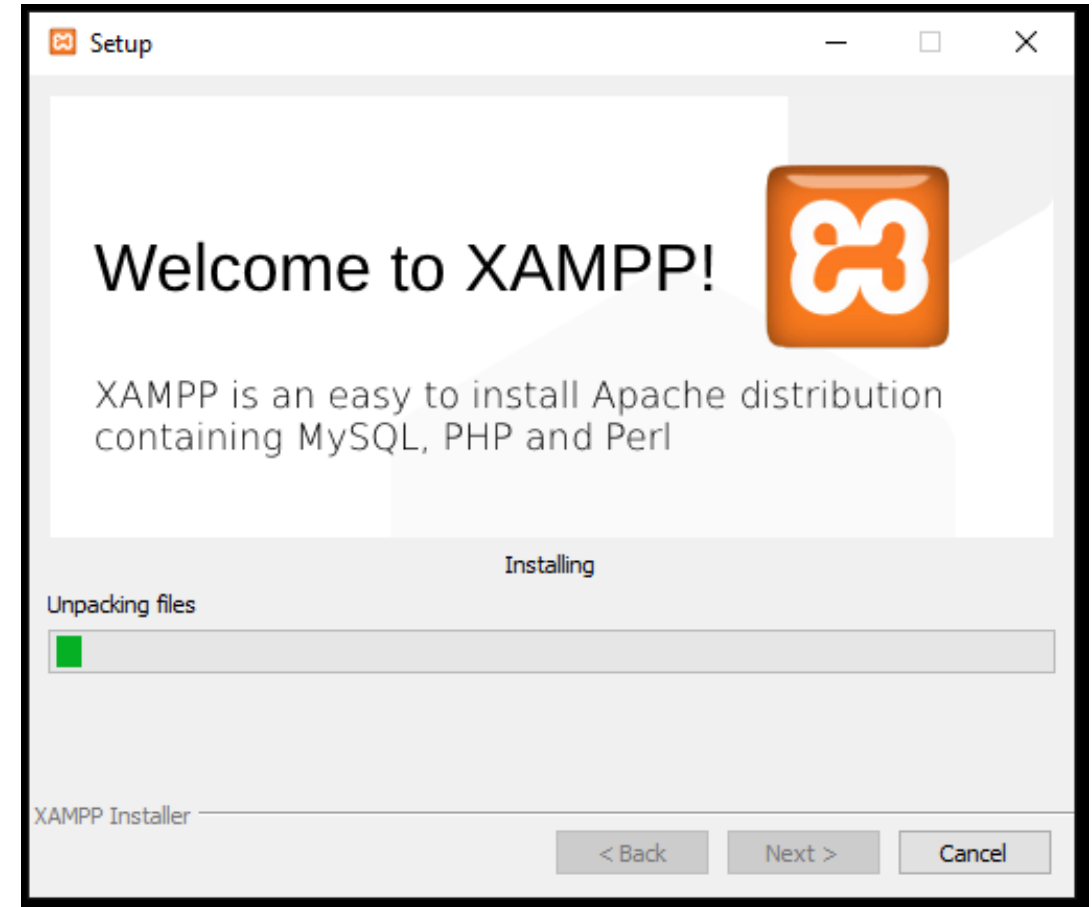


*Die Installation verläuft wie jede andere und Sie können die Einstellungen auf ihren Standard-Werten lassen. Mindestens aber brauchen wir MySQL und phpMyAdmin.*

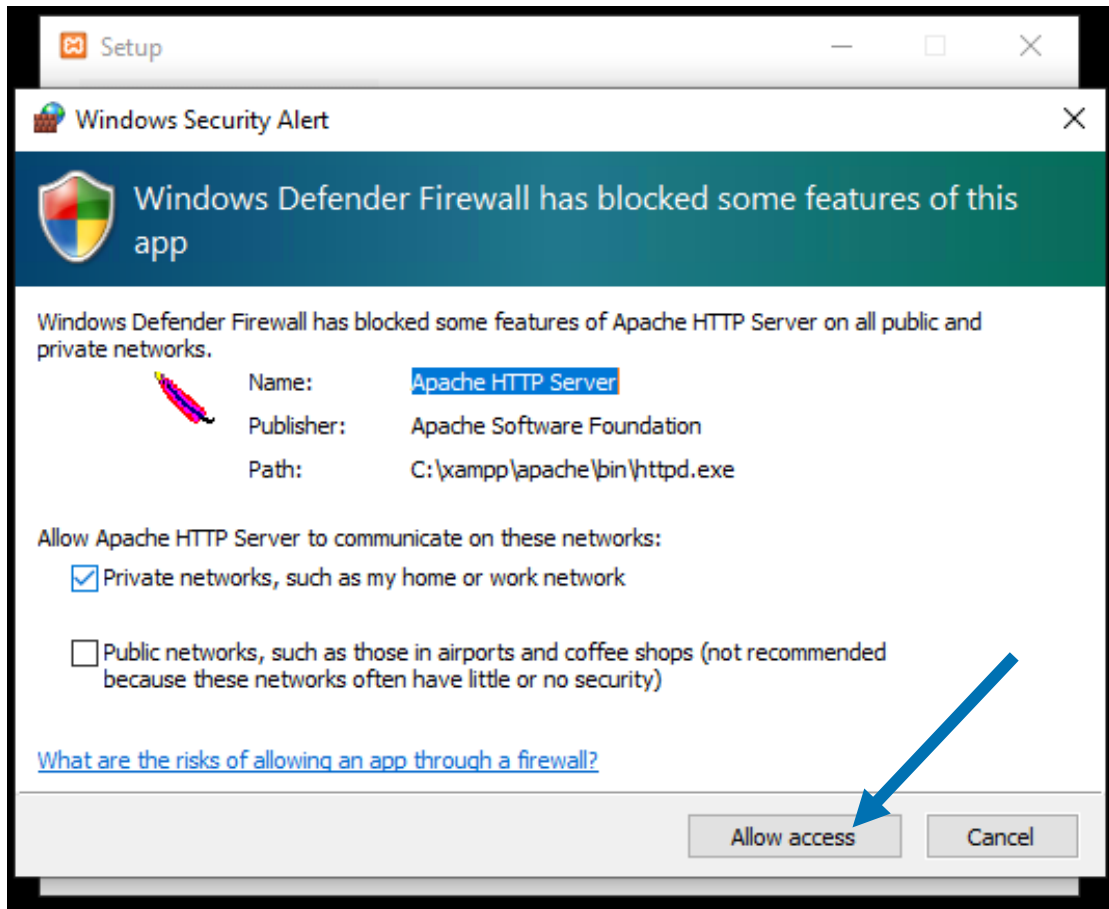
## Installation XAMPP



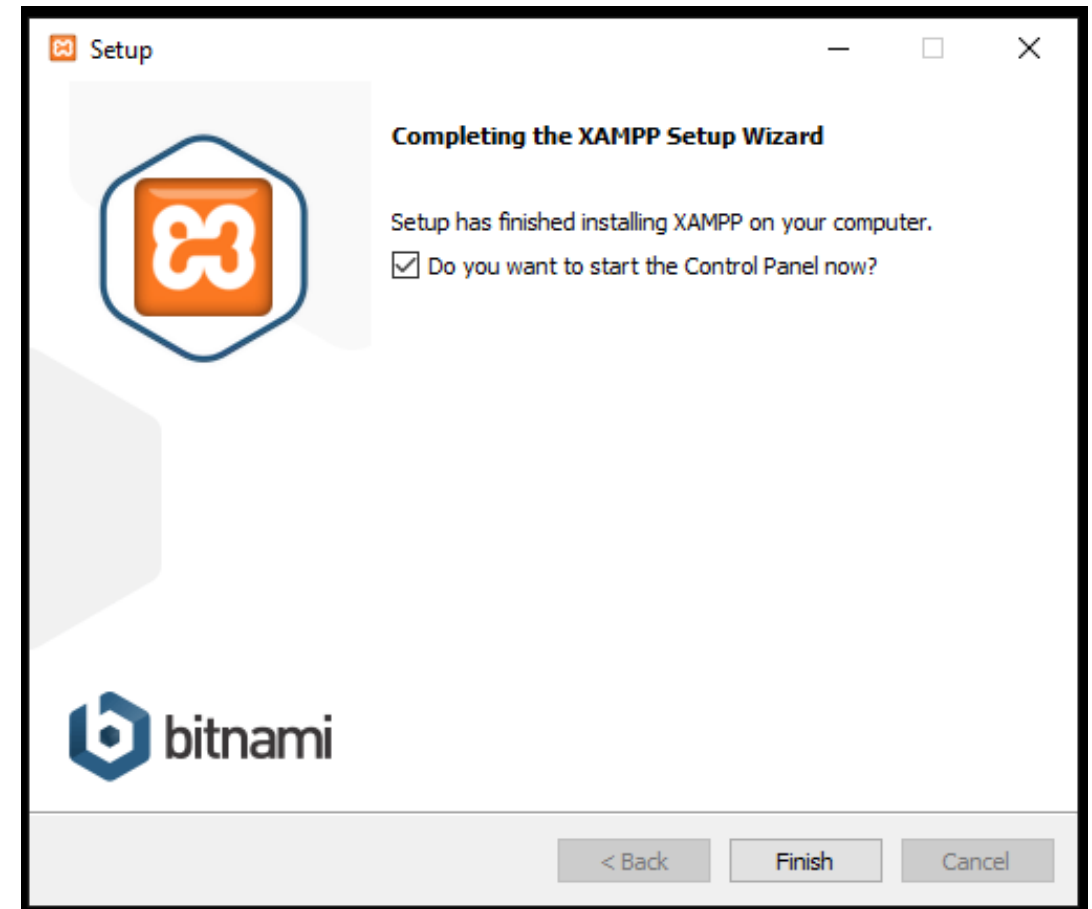
*Installieren Sie XAMPP direkt unter C:\xampp*



## Installation XAMPP



*Eventuell werden Sie darum gebeten, den Apache HTTP Server durch die Windows Firewall zu lassen. Dies können Sie bestätigen.*



„Sie haben XAMPP  
erfolgreich installiert.“





# 02

## MySQL Connector und Eclipse

### 2.1 Download ←

### 2.2 *Einrichten der Entwicklungsumgebung*

*Um in Java auf MySQL Datenbanken zugreifen zu können, muss ein MySQL Connector installiert werden. Auf diesen Connector werden wir in unseren Java Projekten, die Datenbanken verwenden sollen, verweisen.*



## Download von MySQL Connector



General Availability (GA) Releases Archives

### Connector/J 8.0.26

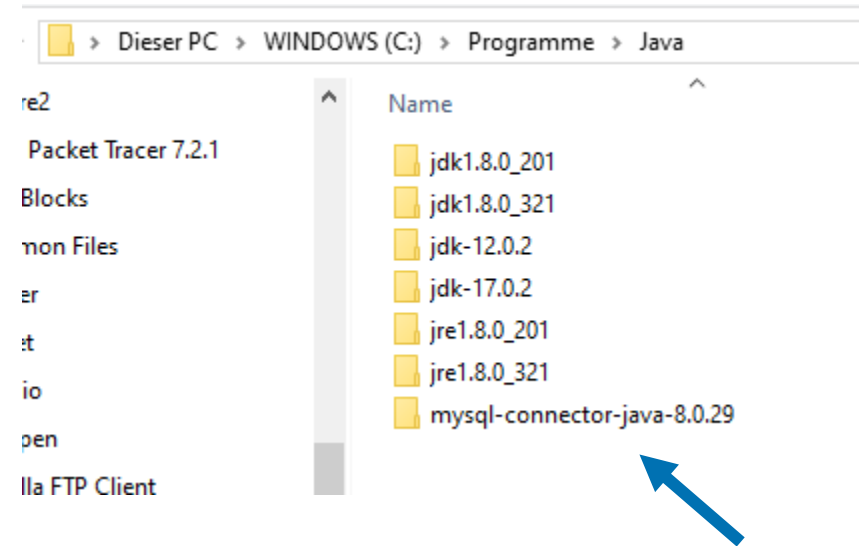
Select Operating System:  
Platform Independent

Platform Independent (Architecture Independent), Compressed TAR Archive (mysql-connector-java-8.0.26.tar.gz)	8.0.26	4.0M	Download
Platform Independent (Architecture Independent), ZIP Archive (mysql-connector-java-8.0.26.zip)	8.0.26	4.7M	Download

MD5: 29df9c3a386684c27b9585d1... | Signature

MD5: 37377584bf16245d6401faedace08067 | Signature

We suggest that you use the [MD5 checksums](#) and [GnuPG signatures](#) to verify the integrity of the packages you download.



Unter folgendem Link können Sie den MySQL Connector herunterladen.

<https://dev.mysql.com/downloads/connector/j/>

Wählen Sie als Betriebssystem „Platform Independent“ aus und laden Sie das Zip-Archiv herunter.

Entpacken Sie das Archiv in Ihren Java Ordner.

# 02

## MySQL Connector und Eclipse

### 2.1 Download

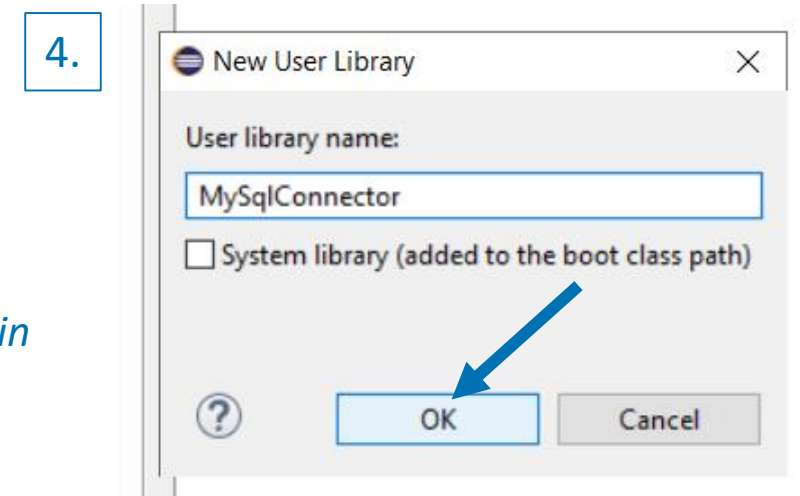
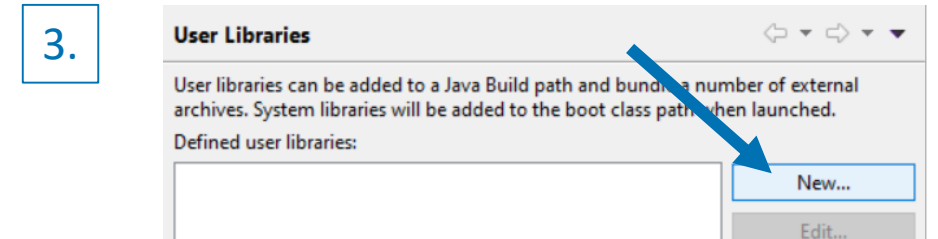
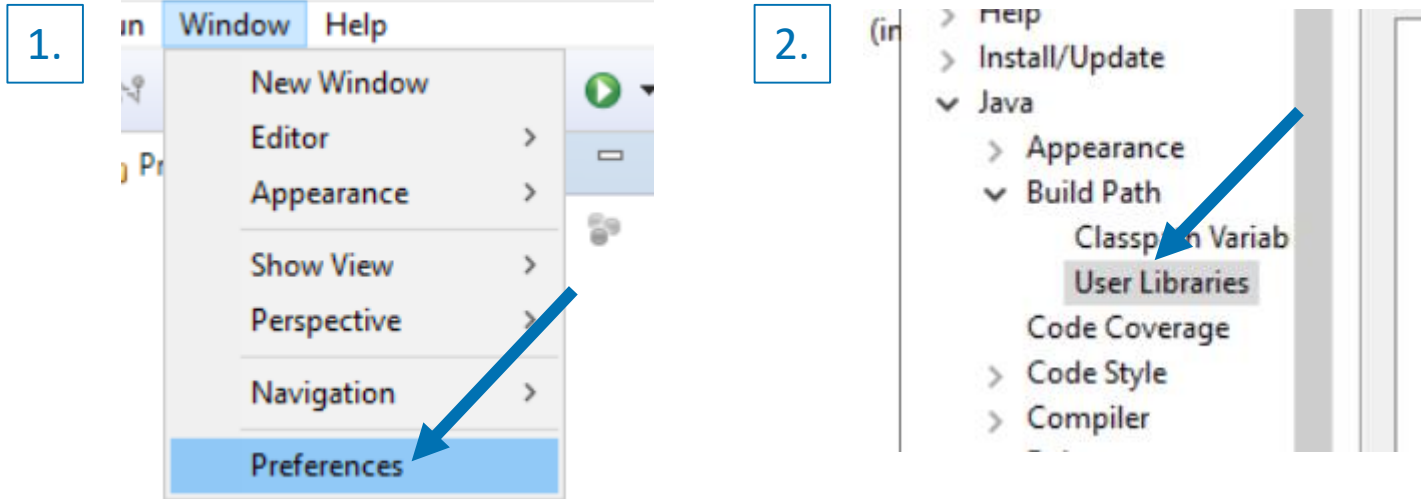
### 2.2 Einrichten der Entwicklungsumgebung ←

*Um in Java auf MySQL Datenbanken zugreifen zu können, muss ein MySQL Connector installiert werden. Auf diesen Connector werden wir in unseren Java Projekten, die Datenbanken verwenden sollen, verweisen.*



## Einrichten der Entwicklungsumgebung

Öffnen Sie Eclipse und arbeiten sie folgende Stichpunkte ab!

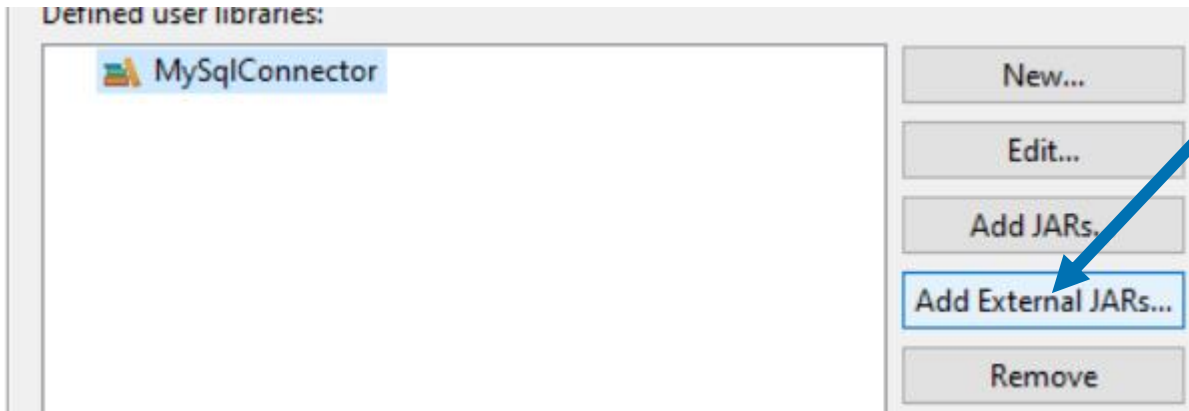


Trage einen beliebigen Namen ein

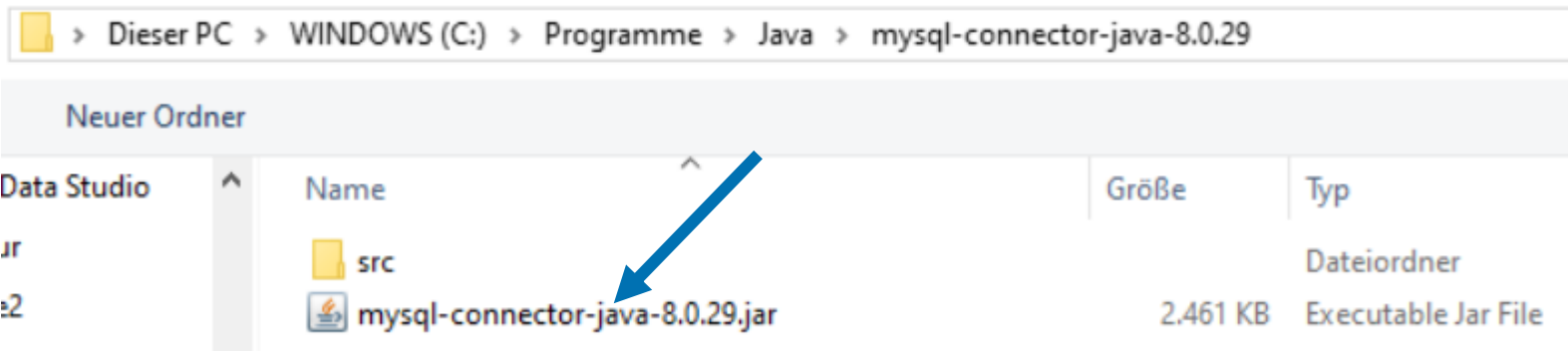


## Einrichten der Entwicklungsumgebung

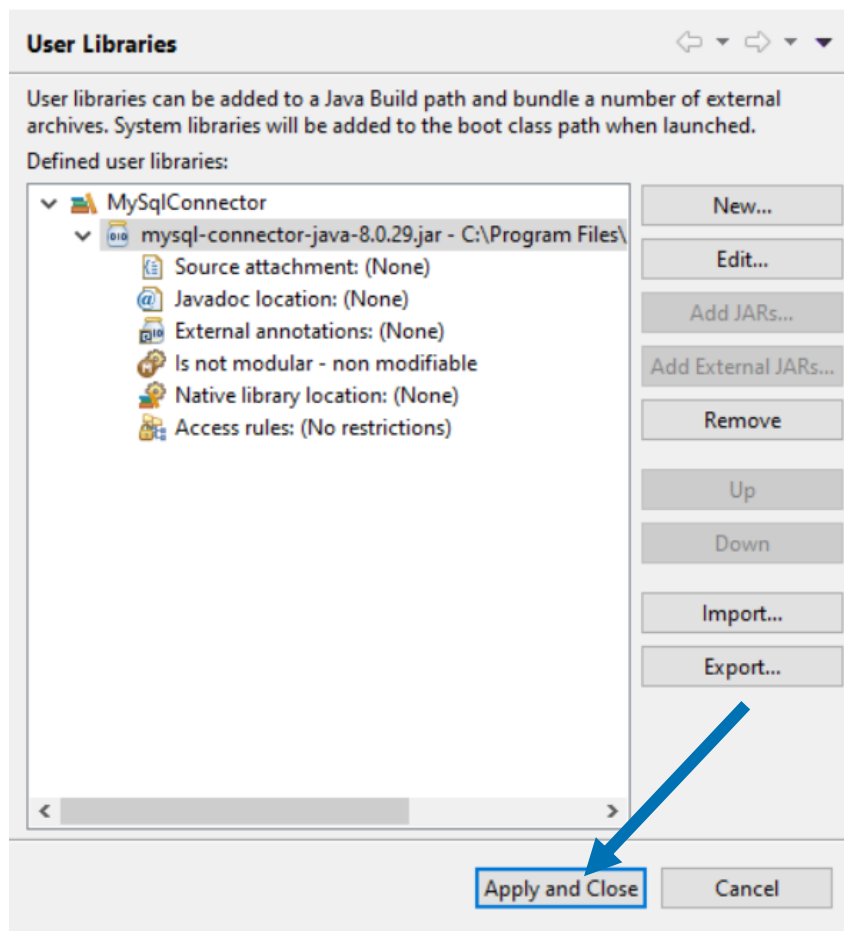
5.



6.



7.



*Füge die .jar File zu einem bestehendem, oder einem neuem Projekt hinzu*

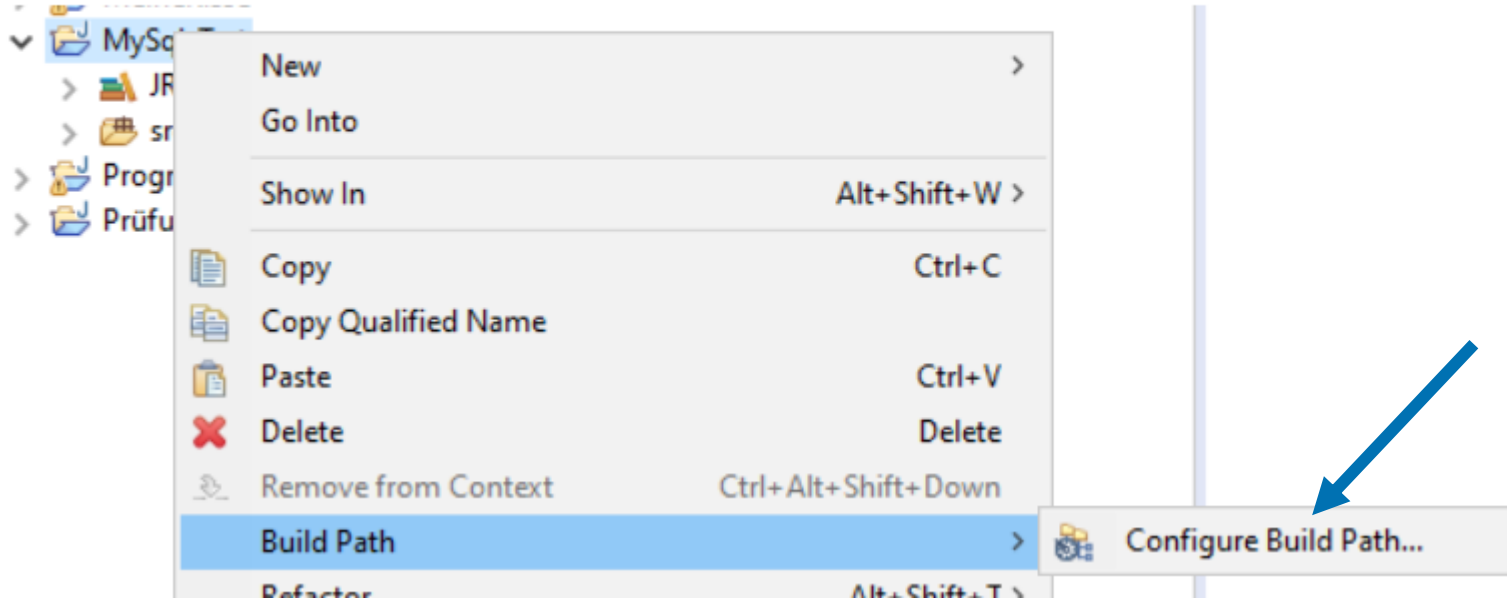


## Einrichten der Entwicklungsumgebung

Füge die .jar File in ein bestehendes oder neues Projekt hinzu

8.

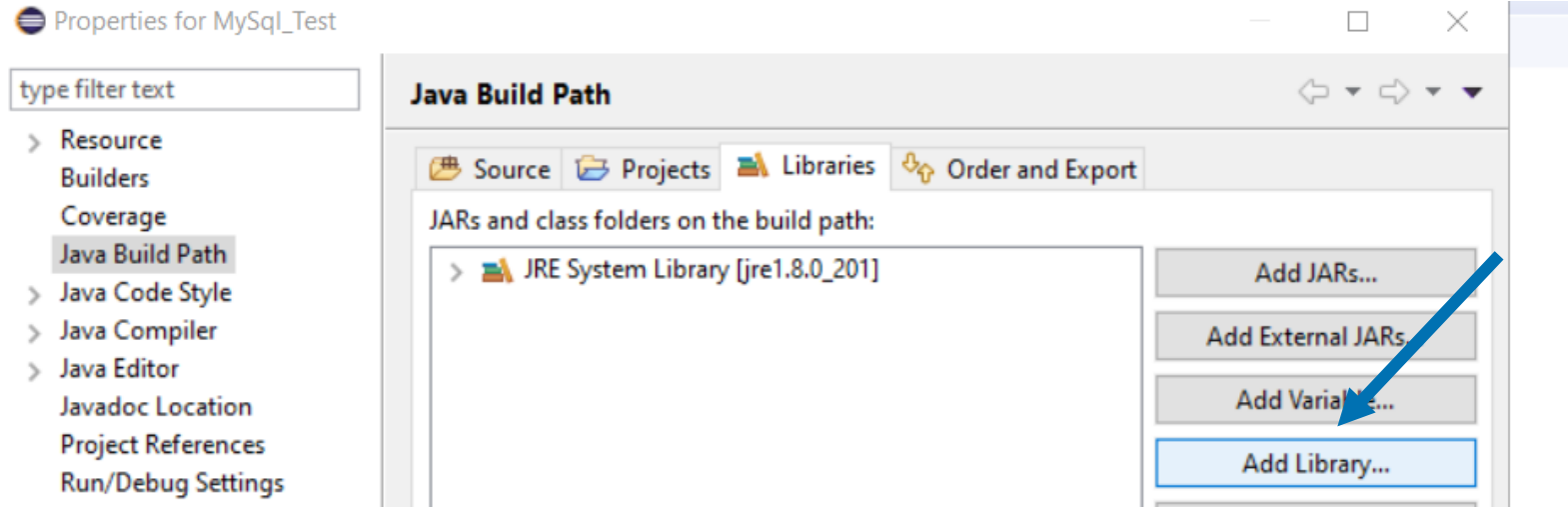
Klicken Sie auf den Wurzelknoten des Projektes



## Einrichten der Entwicklungsumgebung

Füge die .jar File in ein bestehendes oder neues Projekt hinzu

9.

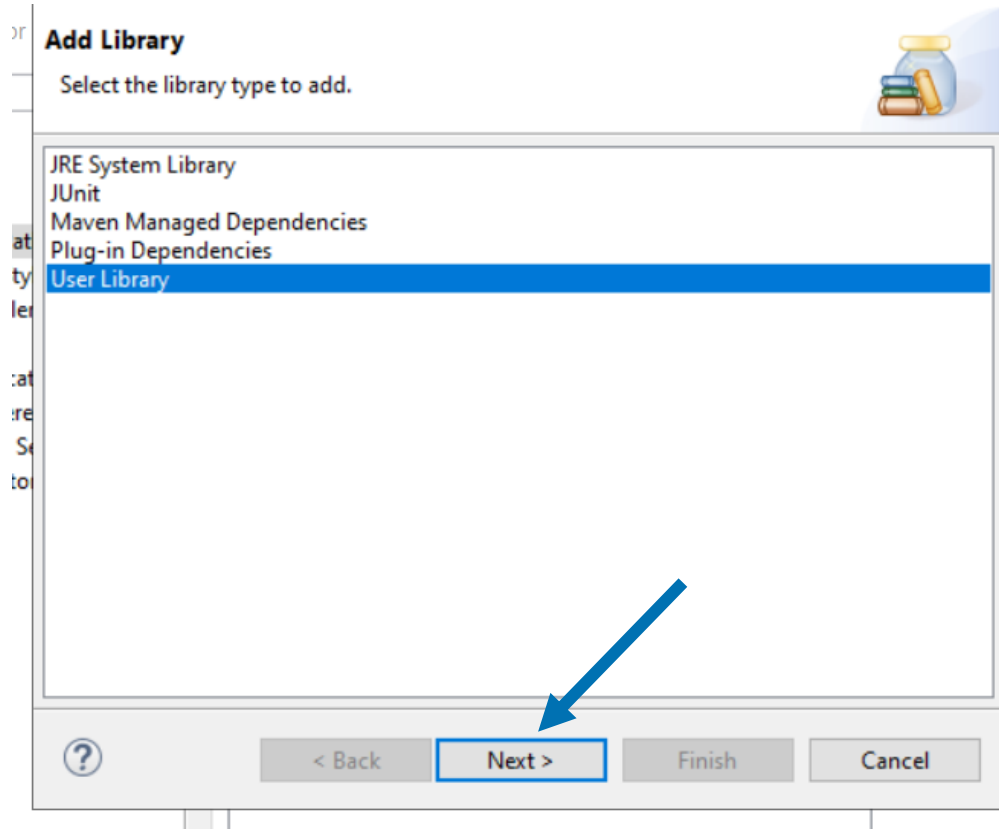




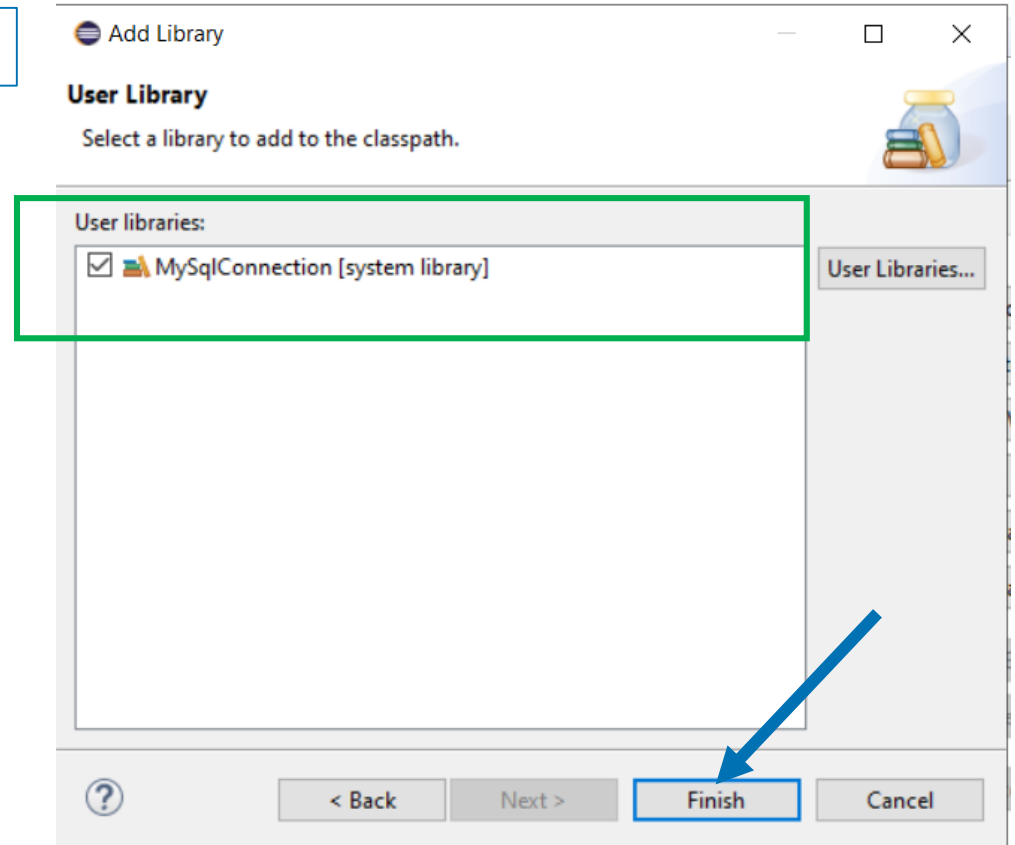
## Einrichten der Entwicklungsumgebung

Füge die .jar File in ein bestehendes oder neues Projekt hinzu

10.



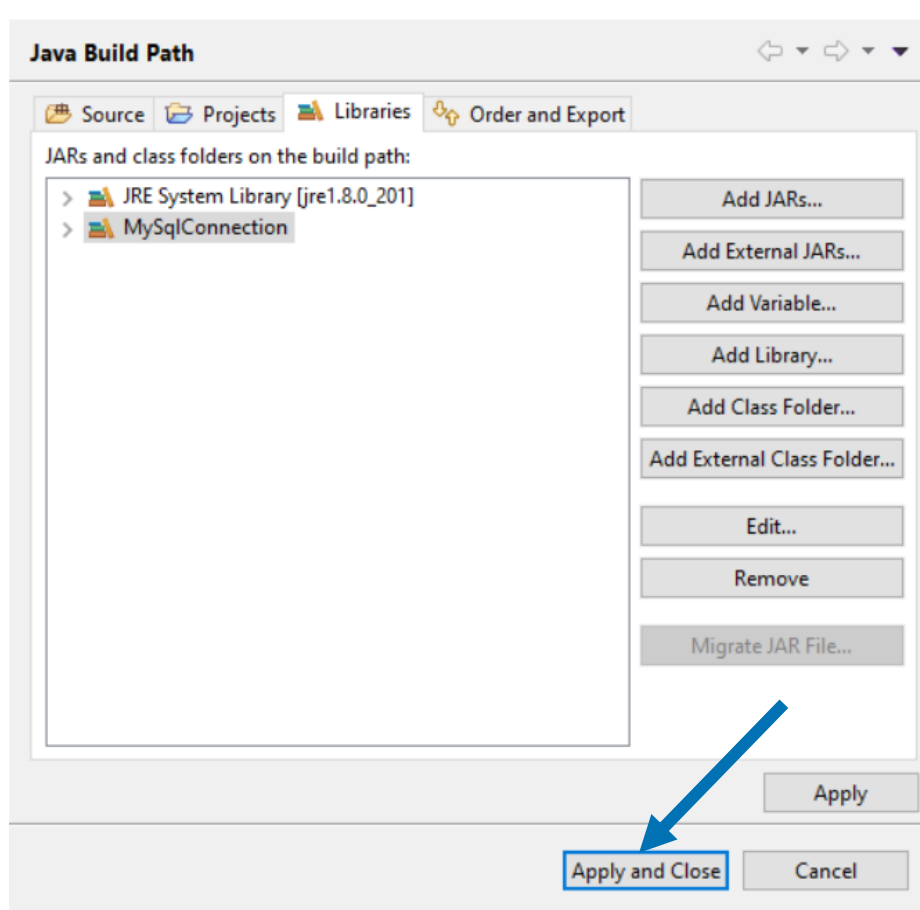
11.



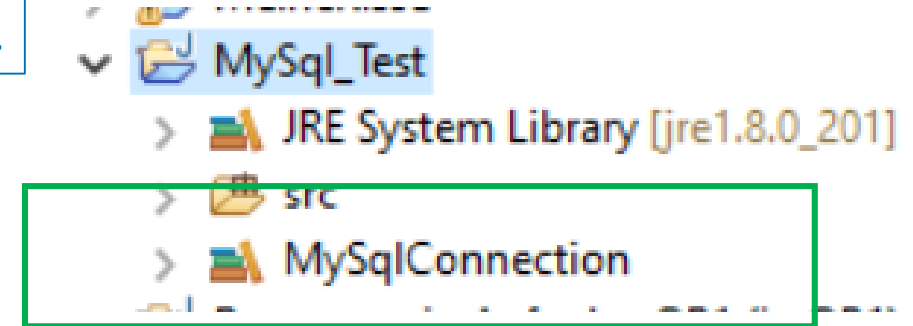
## Einrichten der Entwicklungsumgebung

Füge die .jar File in ein bestehendes oder neues Projekt hinzu

12.



13.



*Fertig!!!!*

„Sie haben erfolgreich  
den MySQL Connector  
installiert.“



# 03

## PHP My Admin und Eclipse

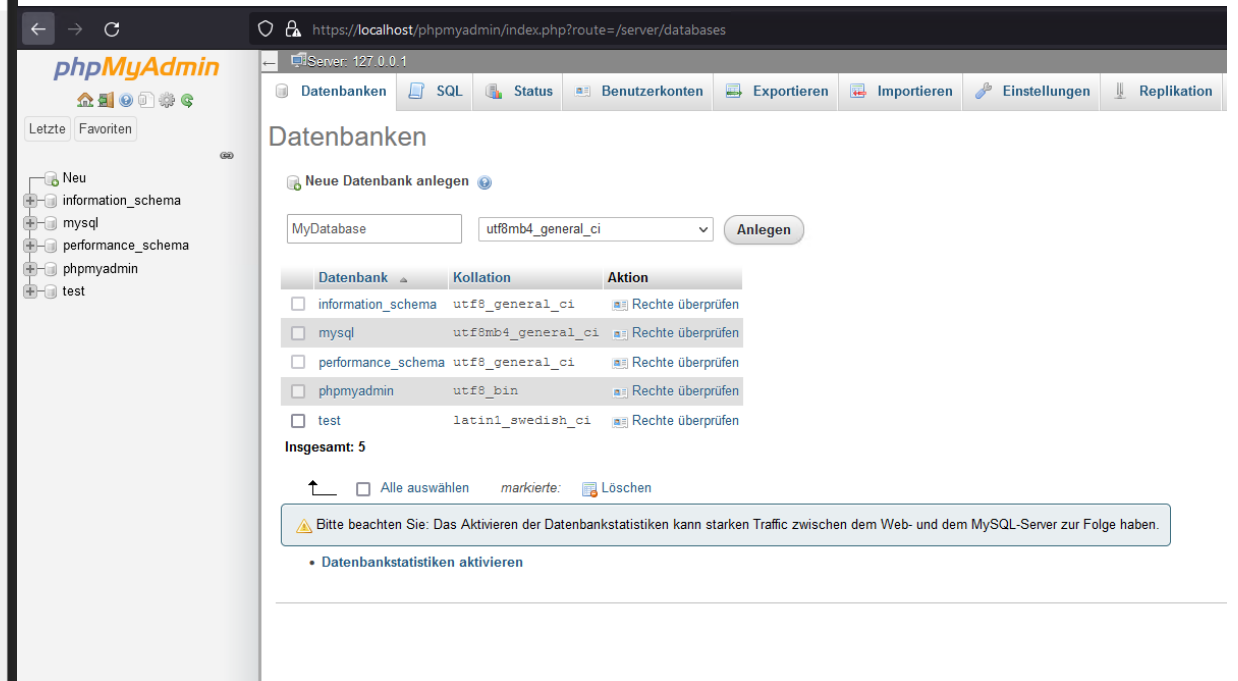
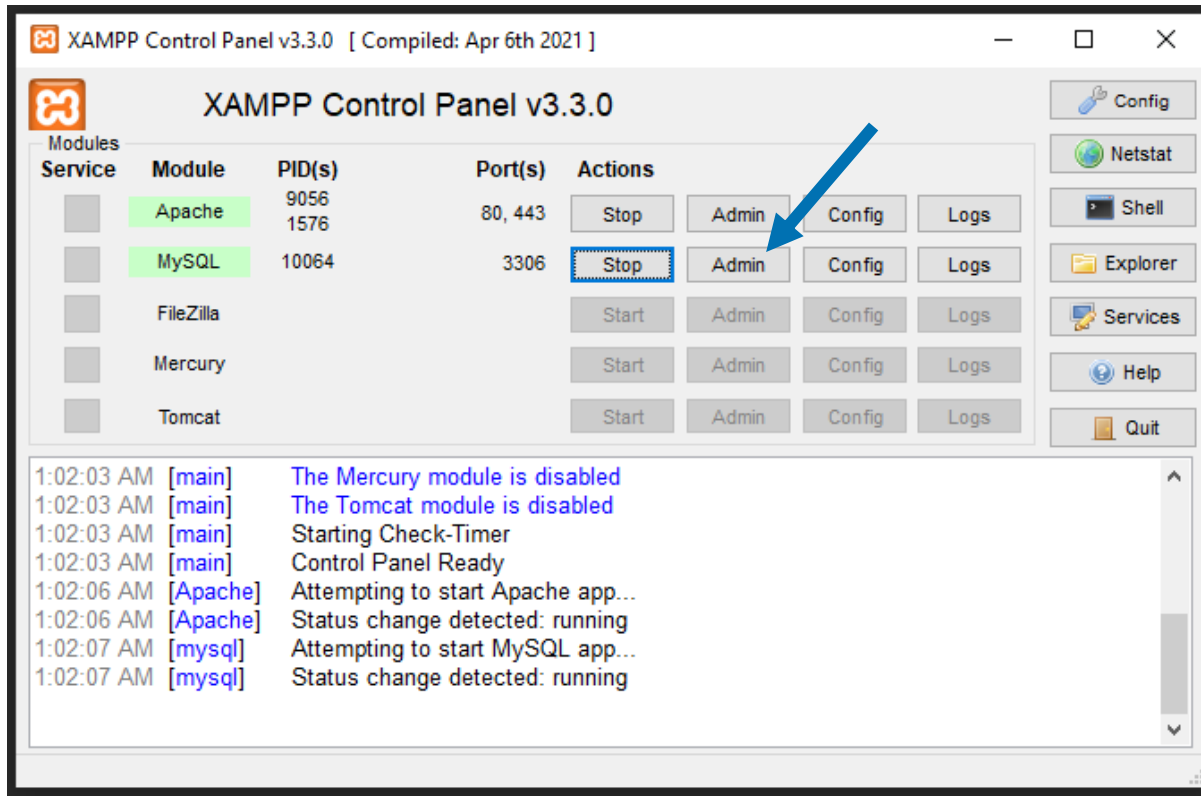
*3.1 Datenbank in PHP My Admin erstellen ←*

*3.2 Datenbankabfrage in Java*

*Zuerst erstellen wir eine Testdatenbank mit einer Tabelle und Beispieldaten in PHP My Admin und fragen diese Tabelle anschließend in Java ab.*



## Datenbank in PHP My Admin erstellen



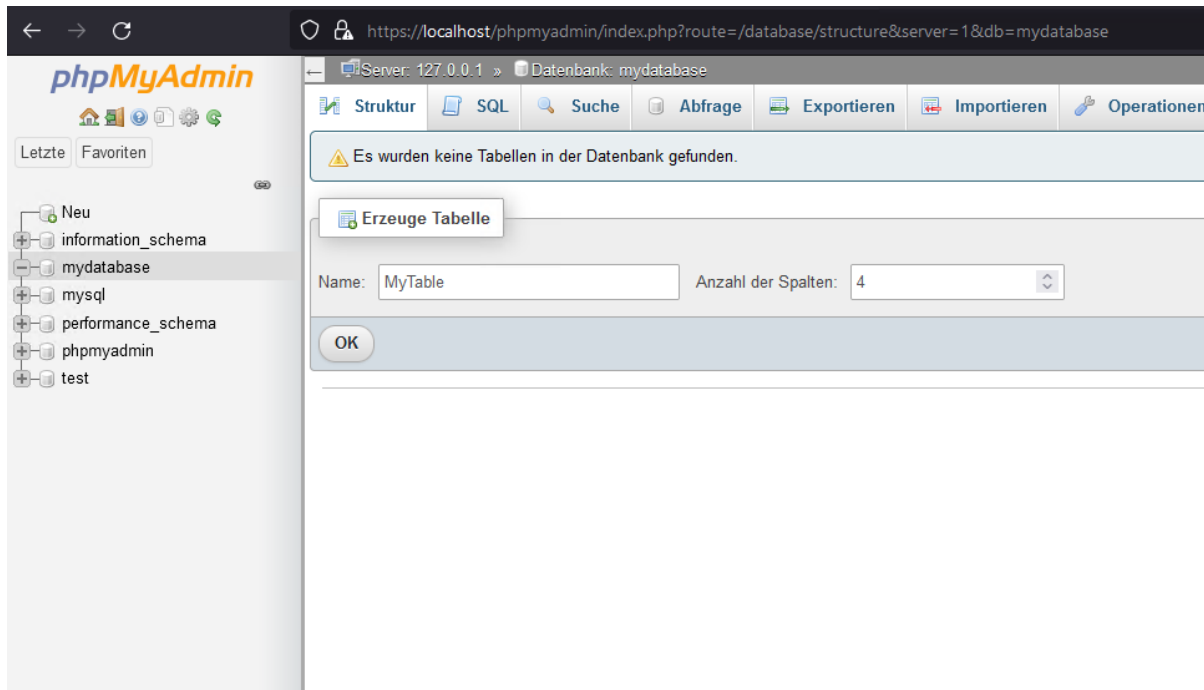
Starten Sie XAMPP, Apache und MySQL

Klicken Sie bei MySQL auf Admin

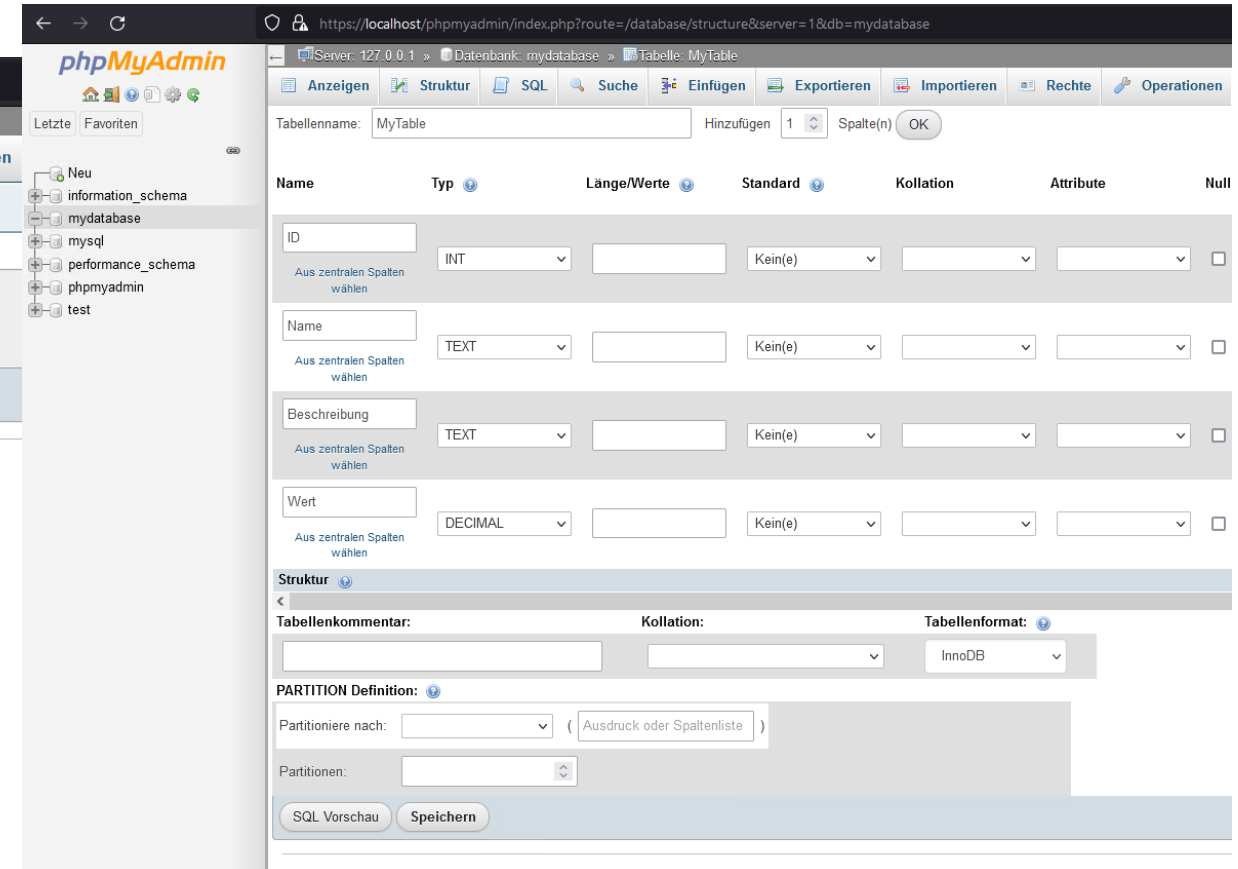
Es öffnet sich ein Browser mit PHP My Admin

Navigieren Sie zu den Datenbanken und erstellen Sie die Datenbank „MyDatabase“

## Datenbank in PHP My Admin erstellen



*Erzeugen Sie die Tabelle „MyTable“ mit 4 Spalten und klicken Sie auf OK*



Name	Typ	Länge/Werte	Standard	Kollation	Attribute	Null
ID	INT		Kein(e)			<input type="checkbox"/>
Name	TEXT		Kein(e)			<input type="checkbox"/>
Beschreibung	TEXT		Kein(e)			<input type="checkbox"/>
Wert	DECIMAL		Kein(e)			<input type="checkbox"/>

*Erstellen Sie die Spalten „ID“, „Name“, „Beschreibung“, „Wert“ mit INT, TEXT, TEXT und DECIMAL entsprechend.*

*Klicken Sie anschließend unten auf Speichern*

## Datenbank in PHP My Admin erstellen

The screenshot displays the PHPMyAdmin interface. On the left, the 'SQL' tab is active, showing the execution of an SQL command in the table 'mytable' of the 'mydatabase' database. The command is:

```
1 INSERT INTO mytable
2 VALUES
3 (1, "Testname", "Testbezeichnung", 9000);
```

On the right, the 'Anzeigen' (Display) tab is active, showing the query results. A message indicates that the current selection does not contain a unique ('unique') column. Below this, a green bar confirms that 1 row is displayed. The SQL query shown is:

```
SELECT * FROM `mytable`
```

Below the query, there are options to 'Messen' (Measure) and 'Inline bearbeiten' (Inline edit). The results are displayed in a table with columns 'ID', 'Name', 'Beschreibung', and 'Wert'.

ID	Name	Beschreibung	Wert
1	Testname	Testbezeichnung	9000

At the bottom, there are options to 'Drucken' (Print), 'In Zwischenablage kopieren' (Copy to clipboard), 'Exportieren' (Export), and 'Diagramm anzeigen' (Show diagram).

*Erzeugen Sie nun einen Eintrag in der Tabelle*

# 03

## PHP My Admin und Eclipse

*3.1 Datenbank in PHP My Admin erstellen*

*3.2 Datenbankabfrage in Java ←*

*Zuerst erstellen wir eine Testdatenbank mit einer Tabelle und Beispieldaten in PHP My Admin und fragen diese Tabelle anschließend in Java ab.*





```
2
3 import java.sql.*;
4
5 import java.sql.Connection;
6 import java.sql.DriverManager;
7 import java.sql.SQLException;
8 import java.sql.ResultSet;
9 import java.sql.Statement;
10
11 public class Main {
12
```

Öffnen Sie wieder IntelliJ mit dem Projekt, welches den Verweis auf den Connector besitzt.

Fügen Sie dem Projekt die Import-Anweisung `java.sql.*` hinzu

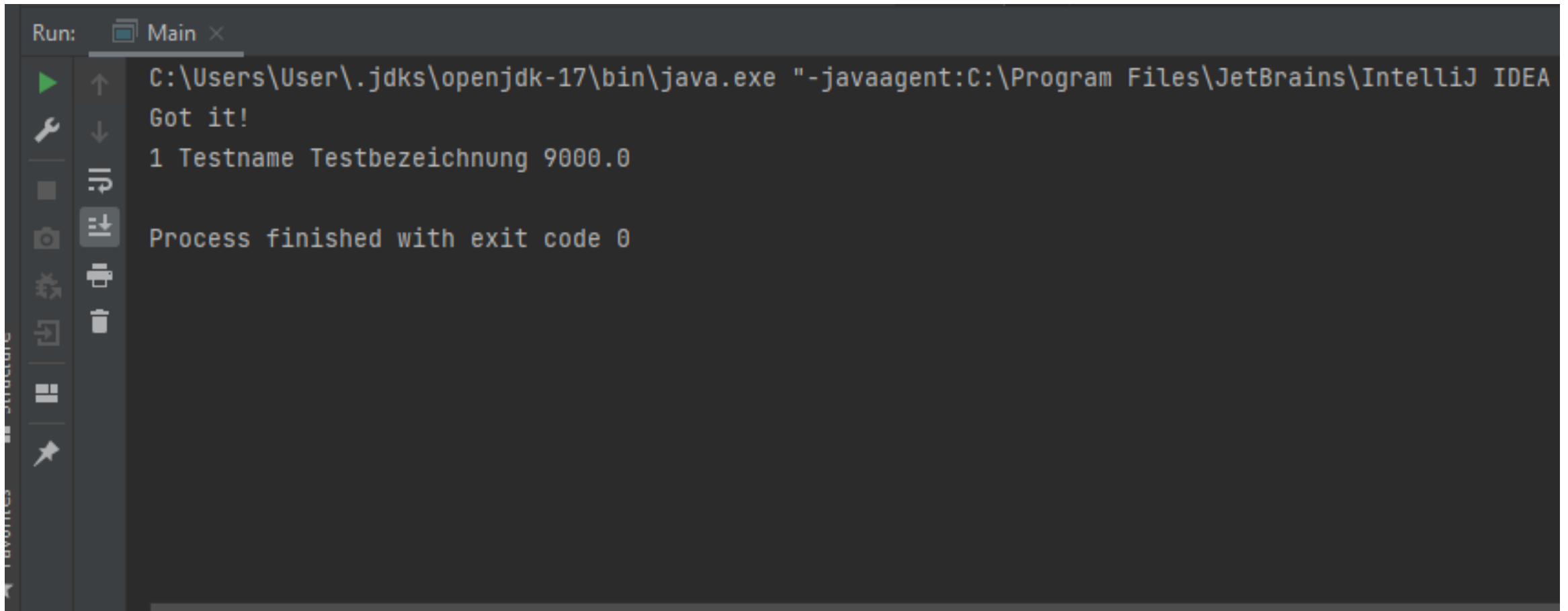
Für dieses Beispiel werden im Detail `.Connection`, `.DriverManager`, `.SQLException`, `.ResultSet` und `.Statement` benötigt.

```
12
13 public static void main(String[] args)
14 {
15     Connection connection = null;
16     try
17     {
18         String connectionString = "jdbc:mysql://localhost:3306/mydatabase";
19         connection = DriverManager.getConnection(connectionString, user: "root", password: "");
20
21         System.out.println("Got it!");
22
23         Statement statement = connection.createStatement();
24         ResultSet rs = statement.executeQuery( sql: "select * from mytable");
25         while(rs.next())
26         {
27             String sb = rs.getInt( columnLabel: "id") +
28                 " " + rs.getString( columnLabel: "name") +
29                 " " + rs.getString( columnLabel: "beschreibung") +
30                 " " + rs.getDouble( columnLabel: "wert");
31             System.out.println(sb);
32         }
33         connection.close();
34     }
35     catch (SQLException e)
36     {
37         System.out.println(e);
38     }
39 }
40
41 }
```

Durch eine Reihe von Anweisungen können wir nun einen Select Befehl auf der Datenbank ausführen und uns die Ergebnisse anzeigen lassen. Die Befehle sind auf der folgenden Folie genauer erläutert.

- `String connectionString` beinhaltet eine Zeichenfolge, mit der die Verbindung zur Datenbank aufgebaut werden kann. Verschiedene Datenbanken erfordern unterschiedliche Connection Strings
- `DriverManager.getConnection(...)` versucht nun, anhand des Connection Strings eine Verbindung zur Datenbank aufzubauen und gibt die Verbindung als Objekt zurück.
- `connection.createStatement()` erzeugt passend zu der Connection ein Statement-Objekt, über welches anschließend SQL Befehle an die Datenbank gesendet werden können.
- `statement.executeQuery(...)` führt die angegebene Query auf der Datenbank aus und gibt das Ergebnis der Abfrage als `ResultSet` zurück.
- mit `rs.next()` in einer Schleife kann Zeile für Zeile das Ergebnis durchlaufen werden.
- Die Getter-Methoden des `ResultSets` geben Zugriff auf die einzelnen Spalten über Index oder Spaltenname.
- Zum Schluss wird die Connection wieder geschlossen.
- Sollte es dabei zu einer Exception kommen, wird diese vom Try-Catch aufgefangen.

## Datenbankabfrage in Java



```
Run: Main x
C:\Users\User\.jdk\openjdk-17\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA
Got it!
1 Testname Testbezeichnung 9000.0
Process finished with exit code 0
```

*Wenn alles geklappt hat, sollten in der Ausgabe nun die Daten der abgefragten Tabelle erscheinen.*

„Sie haben erfolgreich  
eine SQL Abfrage über  
Java ausgeführt.“



# VIELEN DANK!

