Tutorial – MySQL and Task

Inhalt

[Installation 2](#_Toc436661519)

[Xampp 2](#_Toc436661520)

[IntelliJ 2](#_Toc436661521)

[Features of IntelliJ for databases 2](#_Toc436661522)

[Connect database 2](#_Toc436661523)

[Integrated SQL Interpreter 4](#_Toc436661524)

[Connect to MySQL database in projects 5](#_Toc436661525)

[via PHP 5](#_Toc436661526)

[via JDBC driver for Java applications 5](#_Toc436661527)

[SQL Syntax 6](#_Toc436661528)

[SELECT 6](#_Toc436661529)

[UPDATE 6](#_Toc436661530)

[DELETE 7](#_Toc436661531)

[Functions 7](#_Toc436661532)

[Procedures 7](#_Toc436661533)

[Task in class 7](#_Toc436661534)

[Set up database with phpMyAdmin 7](#_Toc436661535)

[Insert data with the SQL Interpreter 8](#_Toc436661536)

[with IntelliJ 8](#_Toc436661537)

[with phpMyAdmin 8](#_Toc436661538)

[Set up project 8](#_Toc436661539)

[Index.php 8](#_Toc436661540)

[Info.php 9](#_Toc436661541)

[Functions.php 10](#_Toc436661542)

# Installation

## Xampp

Please install Xampp for this tutorial. You can get it from [here](https://www.apachefriends.org/de/download.html). It contains an Apache Web Server and a MySQL database that are necessary for this.

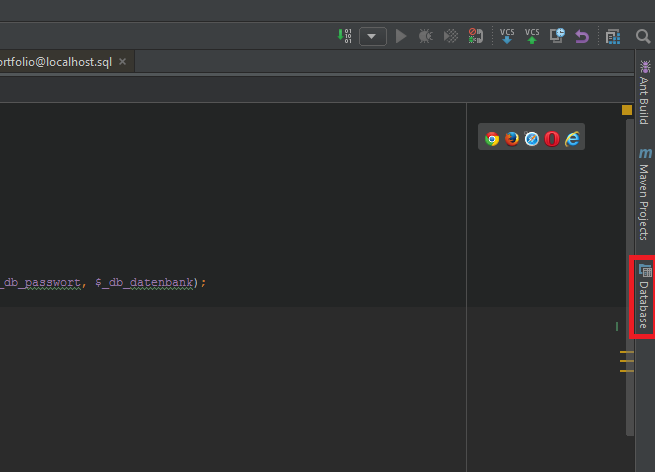
## IntelliJ

You can get IntelliJ from [here](https://www.jetbrains.com/idea/download/)for free (also the Ultimate Edition as student). I recommend you also to install IntelliJ because it has some features that are nice to have. But you can also use another IDE.

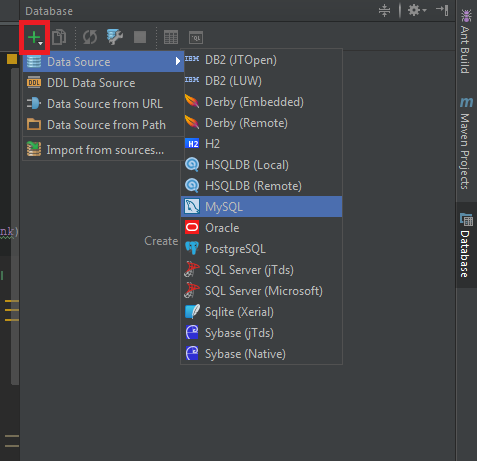
# Features of IntelliJ for databases

## Connect database

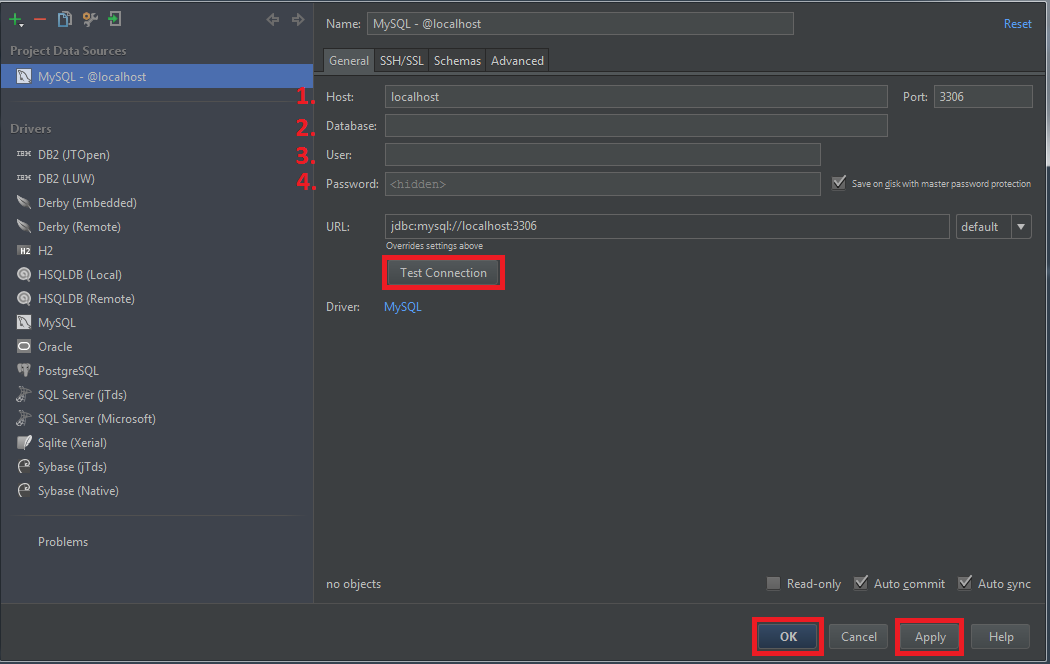
In IntelliJ you can create a connection to a database. For these feature you have to be in a project. Then you must click on the database button on the right side.



After that a window open. In this windows are a green plus symbol, on that you click.



Now you choose data source and MySQL. A new window will open. In this window you have to enter the information of the database.



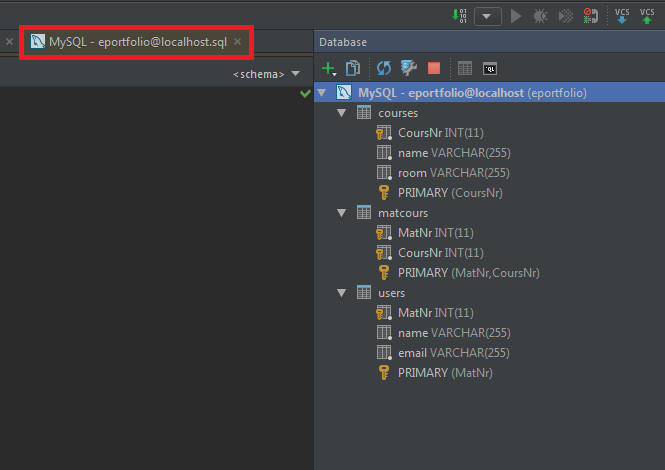
In line one IntelliJ insert as standard localhost as host. If you have another place for your database than enter it here.

In line two you have to enter the name of the database you want to open.

In the third line you have to enter the user, as standard root.

In the last line you have to enter the password for the user, for root let it empty.

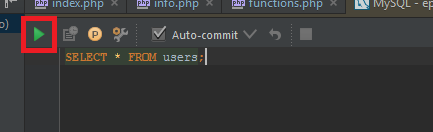
After this click on the test connection button to test the data given. If correct data is given the connection is fine and you can click on ok. Elsewhere you have to enter the correct data.



Now, if you click the database button, a new database is shown. If you click on it, you see all tables and rows of it. Also a new window is open called “MySQL - databasename@databasehost.sql”. This is the integrated SQL Interpreter.

## Integrated SQL Interpreter

In this you can query SQL statements. To query a statement, you have to insert it and click on the green arrow on the left.



To query more than one statement you have to select all with Strg+A and click on the green arrow.

# Connect to MySQL database in projects

It gives many ways to connect a MySQL database to a project. I will show you that on to examples. First on PHP on an easy website project and then in a Java application. In both there are only little differences that I now show up.

## via PHP

I a web project you can use a MySQL database via PHP. PHP provides frameworks for this but I will show it to you with PHP functions.

$db\_link = mysqli\_connect ( MYSQL\_HOST,

MYSQL\_BENUTZER,

MYSQL\_KENNWORT,

MYSQL\_DATENBANK);

At the beginning call the mysqli\_connect() function. To the mysqli\_connect() function you give over the host, a user, a password and the database that you want to use. Now you have created a connection to the database.

Now you can create a variable that contains the SQL statement that you want to fetch in the database.

$sql = "SELECT \* FROM users";

To get a resutset from the database you have to call the mysqli\_query() function.

$db\_erg = mysqli\_query( $db\_link, $sql );

This you give over the connection and the SQL statement.

To see if something went wrong you can create an If-clause in PHP that’s tell you what´s going wrong.

if (mysqli\_connect\_errno() ) { echo 'Ungültige Abfrage: ' . mysqli\_error()); }

After that you can use the data from the resultset. With the mysqli\_fetch\_array() function you create an array with the data. Then you can access the data via the name of the column.

while ($zeile = mysqli\_fetch\_array( $db\_erg, MYSQLI\_ASSOC)) { echo $zeile['SPALTENNAME'] ; }

At the end you have to free the space and close the connection to the database.

mysqli\_free\_result( $db\_erg );

mysqli\_close( $db\_link );

## via JDBC driver for Java applications

The same procedure as in PHP.

First you have to connect to the database (green content). After that you can execute a statement(red).

public class DatenbankZugriff {

private Connection connect = null;

private Statement statement = null;

private PreparedStatement preparedStatement = null;

private ResultSet resultSet = null;

public void readDataBase() throws Exception {

try {

Class.forName("com.mysql.jdbc.Driver");

connect = DriverManager.getConnection("jdbc:mysql://localhost/feedback?“ + “user=sqluser&password=sqluserpw");

statement = connect.createStatement();

resultSet = statement.executeQuery("select \* from users");

//Ausgabe von resultSet System.out.println();

} catch (Exception e) {

throw e;

} finally {

resultSet.close();

statement.close();

connect.close();

}

}

}

# SQL Syntax

Here you can find the syntax of a SELECT, an UPDATE and an DELETE statement. Also you find the syntax of custom functions and procedures here. For more syntaxes look [here](http://dev.mysql.com/doc/refman/5.6/en/sql-syntax.html).

## SELECT

SELECT

[ALL | DISTINCT | DISTINCTROW ]

[HIGH\_PRIORITY]

[STRAIGHT\_JOIN]

[SQL\_SMALL\_RESULT] [SQL\_BIG\_RESULT] [SQL\_BUFFER\_RESULT]

[SQL\_CACHE | SQL\_NO\_CACHE] [SQL\_CALC\_FOUND\_ROWS]

***select\_expr*** [, ***select\_expr*** ...]

[FROM ***table\_references***

[PARTITION ***partition\_list***]

[WHERE ***where\_condition***]

[GROUP BY {***col\_name*** | ***expr*** | ***position***}

[ASC | DESC], ... [WITH ROLLUP]]

[HAVING ***where\_condition***]

[ORDER BY {***col\_name*** | ***expr*** | ***position***}

[ASC | DESC], ...]

[LIMIT {[***offset***,] ***row\_count*** | ***row\_count*** OFFSET ***offset***}]

[PROCEDURE ***procedure\_name***(***argument\_list***)]

[INTO OUTFILE '***file\_name***'

[CHARACTER SET ***charset\_name***]

***export\_options*** | INTO DUMPFILE '***file\_name***' | INTO ***var\_name*** [, ***var\_name***]]

[FOR UPDATE | LOCK IN SHARE MODE]]

## UPDATE

UPDATE [LOW\_PRIORITY] [IGNORE]

***table\_reference***

SET

***col\_name1***={***expr1***|DEFAULT} [, ***col\_name2***={***expr2***|DEFAULT}] ...

[WHERE ***where\_condition***]

[ORDER BY ...]

[LIMIT ***row\_count***]

## DELETE

DELETE [LOW\_PRIORITY] [QUICK] [IGNORE]

FROM

***tbl\_name***

[PARTITION (***partition\_name***,...)]

[WHERE ***where\_condition***]

[ORDER BY ...]

[LIMIT ***row\_count***]

## Functions

CREATE

[DEFINER = { ***user*** | CURRENT\_USER }]

FUNCTION ***sp\_name*** ([***func\_parameter***[,...]])

RETURNS ***type*** [***characteristic*** ...]

Begin

***statement***

RETURN ***parameter***

END;

## Procedures

CREATE

[DEFINER = { ***user*** | CURRENT\_USER }]

PROCEDURE ***sp\_name*** ([***proc\_parameter***[,...]])

[***characteristic*** ...]

Begin

***statement***

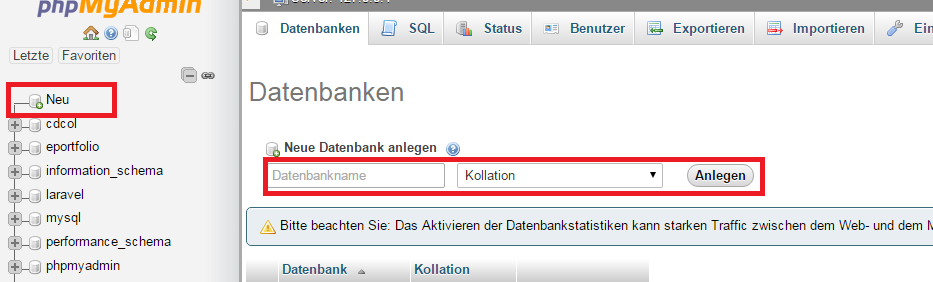
END;

# Task in class

Create a „Studentverzeichnis”. On the homepage you should see the Matrikelnummer, the name and the email of every student that is in the database. If you click on a delete button the entry of the student should be deleted, if you click on a view button you should see additional information to the student like the courses.

## Set up database with phpMyAdmin

After installing Xampp you should go in the browser to the location localhost/phpMyAdmin.



On the left site click on the plus and create a database named “eportfolio”. As collation use utf\_8\_general\_ci

## Insert data with the SQL Interpreter

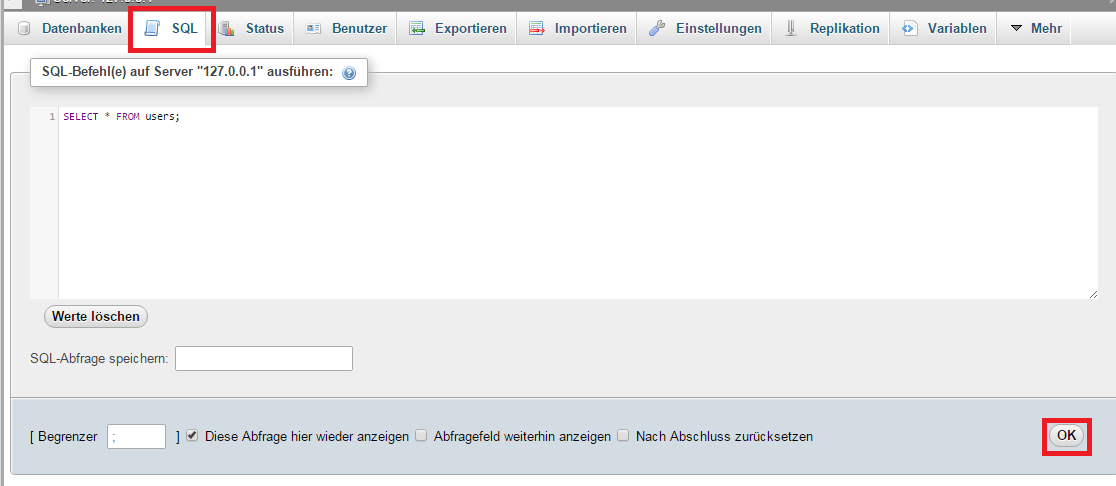
### with IntelliJ

If you have integrated the database into IntelliJ, you can use it to execute SQL statements directly. For this you can go the the tab called “MySQL – eportfolio@localhost”.

In this tab you can run the content of the database.sql from github.

### with phpMyAdmin

Click on SQL and you have open the integrated SQL interpreter.



In this you can run the content of the database.sql from github.

## Set up project

Create a project in the htdocs folder unter C://Program Files/Xampp/htdocs.

In this you create a index.php, info.php and functions.php .

### Index.php

The index.php is the main site which contains the list of all students with name, email and Matrikelnummer.

<?php

include('functions.php');

$\_db\_host = "localhost";

$\_db\_username = "root";

$\_db\_passwort = "";

$\_db\_datenbank = "eportfolio";

if (isset($\_GET['id\_d'])) {

delete($\_GET['id\_d'], $\_db\_host, $\_db\_username, $\_db\_passwort, $\_db\_datenbank);

}

?>  
<html>  
 <head>  
 <title>Studentenverzeichnis</title>  
 </head>  
 <body style="align-content: center">  
 <h1>Studentenverzeichnis</h1>  
 <div id="users">  
 <table border="0.5">  
 <thead>  
 <th>Matrikelnummer</th>  
 <th>Name</th>  
 <th>Email</th>  
 </thead>  
<?php

$db\_erg = onLoad($\_db\_host, $\_db\_username, $\_db\_passwort, $\_db\_datenbank);

while ($zeile = mysqli\_fetch\_array( $db\_erg, MYSQLI\_ASSOC)) {

$\_link = "location.href='index.php?id\_d=" . $zeile['MatNr'] . "'";

$\_link2 = "location.href='info.php?id\_b=" . $zeile['MatNr'] . "'";

echo "<tr>";

echo "<td>" . $zeile['MatNr'] . "</td>";

echo "<td>" . $zeile['name'] . "</td>";

echo "<td>" . $zeile['email'] . "</td>";

echo '<td><button onclick="' . $\_link . '">Delete</button></td>';

echo '<td><button onclick="' . $\_link2 . '">View Curses</button></td>';

echo "</tr>";

}

mysqli\_free\_result( $db\_erg );

?>

</table>  
 </div>  
 <div id="ergebnis">  
 </div>  
 </body>  
</html>

### Info.php

The info.php contains the details of a student and a list of all courses of him.

<?php

$\_db\_host = "localhost";

$\_db\_username = "root";

$\_db\_passwort = "";

$\_db\_datenbank = "eportfolio";

include('functions.php');

if (isset($\_GET['id\_b'])) {

$\_student = getStudent($\_GET['id\_b'], $\_db\_host, $\_db\_username, $\_db\_passwort, $\_db\_datenbank);

$\_courses = getCourses($\_GET['id\_b'], $\_db\_host, $\_db\_username, $\_db\_passwort, $\_db\_datenbank);

$\_daten = mysqli\_fetch\_array( $\_student, MYSQLI\_ASSOC);

}

?>

<html>

<head>

<title>Student <?php echo $\_daten['MatNr']; ?></title>

</head>

<body>

<div id="info">

<h1>Data of Student <?php echo $\_daten['MatNr']; ?></h1>

<p>Matrikelnummer: <?php echo $\_daten['MatNr']; ?></p>

<p>Name: <?php echo $\_daten['name']; ?></p>

<p>Email: <?php echo $\_daten['email']; ?></p>

</div>

<div id="courses">

<h3>Courses</h3>

<table border="0">

<thead>

<th style="text-align: left">Cours</th>

<th>Room</th>

</thead>

<?php

while ($zeile = mysqli\_fetch\_array( $\_courses, MYSQLI\_ASSOC)) {

echo "<tr>";

echo "<td>" . $zeile['name'] . "</td>";

echo "<td>" . $zeile['room'] . "</td>";

echo "</tr>";

}

?>

</table>

</div>

<button onclick="location.href='index.php'">back</button>

</body>

</html>

### Functions.php

In the function.php are all functions defined.

<?php

function onLoad($\_db\_host, $\_db\_username, $\_db\_passwort, $\_db\_datenbank)

{

$db\_link = mysqli\_connect(

$\_db\_host,

$\_db\_username,

$\_db\_passwort,

$\_db\_datenbank

);

$sql = "SELECT \* FROM users ORDER BY MatNR ASC";

$db\_erg = mysqli\_query($db\_link, $sql);

if (!$db\_erg) {

die('Ungültige Abfrage: ' . mysqli\_error());

}

return $db\_erg;

}

function delete($id,$\_db\_host,$\_db\_username,$\_db\_passwort, $\_db\_datenbank) {

$db\_link = mysqli\_connect (

$\_db\_host,

$\_db\_username,

$\_db\_passwort,

$\_db\_datenbank

);

$sql = "DELETE FROM matcours WHERE MatNr = " .$id.";";

$sql1 = "DELETE FROM users WHERE MatNr = " .$id.";";

$db\_erg = mysqli\_query( $db\_link, $sql );

$db\_erg1 = mysqli\_query( $db\_link, $sql1 );

if ( ! $db\_erg || ! $db\_erg1 )

{

die('Ungültige Abfrage: ' . mysqli\_error());

}

}

function getStudent($\_MatNR,$\_db\_host,$\_db\_username,$\_db\_passwort, $\_db\_datenbank)

{

$db\_link = mysqli\_connect(

$\_db\_host,

$\_db\_username,

$\_db\_passwort,

$\_db\_datenbank

);

$sql = "SELECT \* FROM users WHERE MatNr=$\_MatNR;";

$db\_erg = mysqli\_query($db\_link, $sql);

if (!$db\_erg) {

die('Ungültige Abfrage: ' . mysqli\_error());

}

return $db\_erg;

}

function getCourses($\_MatNr, $\_db\_host, $\_db\_username, $\_db\_passwort, $\_db\_datenbank)

{

$db\_link = mysqli\_connect(

$\_db\_host,

$\_db\_username,

$\_db\_passwort,

$\_db\_datenbank

);

$sql = "SELECT courses.name, courses.room FROM courses, matcours WHERE courses.CoursNr = matcours.CoursNr AND matcours.MatNr = $\_MatNr;";

$db\_erg = mysqli\_query($db\_link, $sql);

if (!$db\_erg) {

die('Ungültige Abfrage: ' . mysqli\_error());

}

return $db\_erg;

}

?>