

# LU07a - SQL-DDL: DB Management

## Learning Objectives

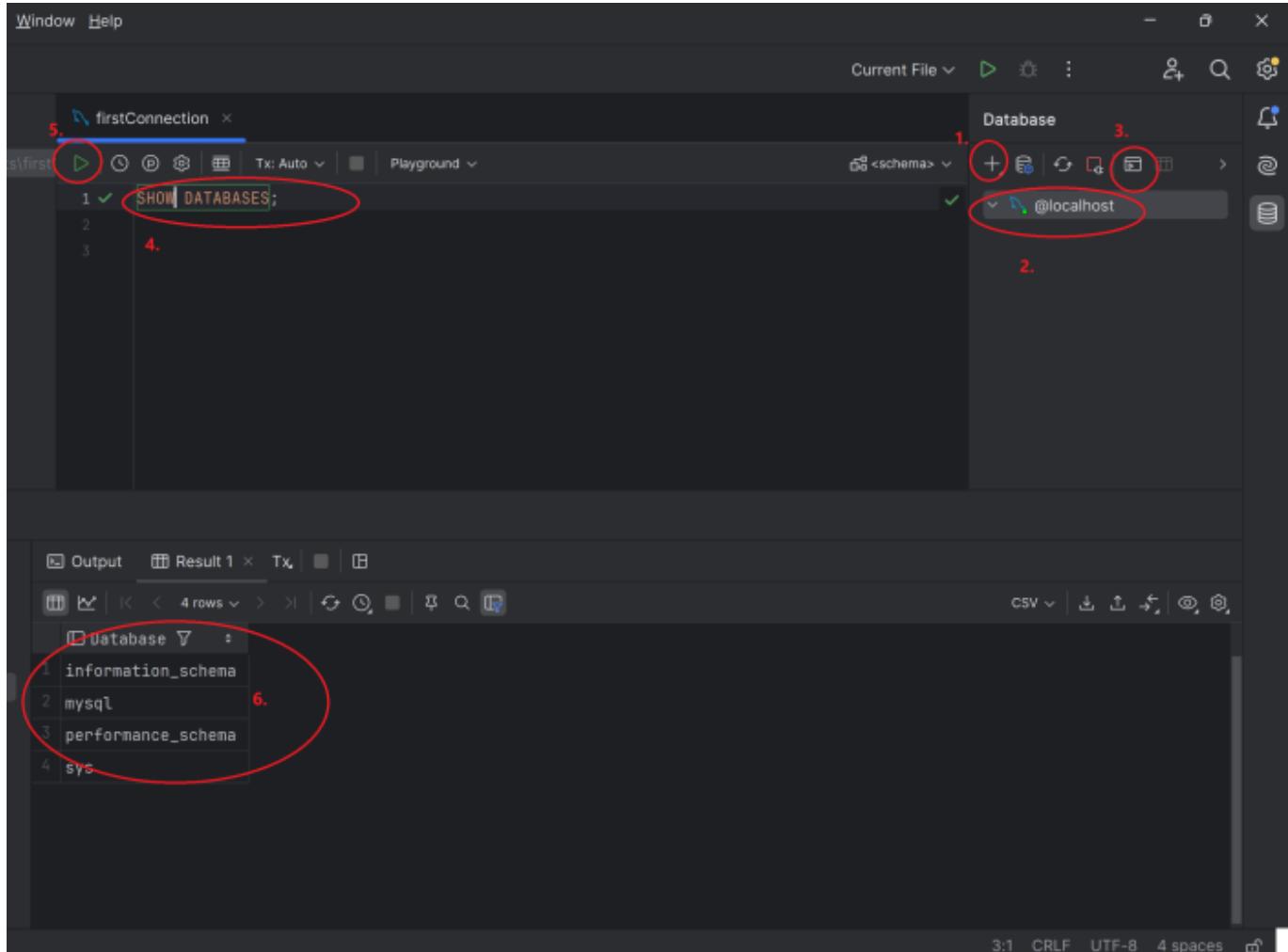
In this session, we will learn some thrilling details about how to properly manage our databases. To achieve this, we need to know more about:

1. What a db schema / db instance is.
2. How to create database instances.
3. How to display all currently available instances.
4. How to connect to one specific db-instance.
5. how you can delete obsolete or no longer required databases.

## Access to the db with Webstorm code editor

There are many ways how we can connect to our database instance. One way is from the terminal or commandline. Instead, during the module 290 we are going to use do the management with the code editor WEBSTORM, which is pictured below.

1. New connection to our MySQL database system
2. The already existing connections to our database instance
3. New terminal window to the database to enter sql commands (leads to 4.)
4. Terminal to the database to enter our SQL commands
5. Execute button that sends the SQL command to the database



The screenshot shows the MySQL Workbench interface. In the top-left, a connection named 'firstConnection' is selected. The top-right shows the 'Database' section with a connection to 'localhost'. The main area contains a SQL editor with the following content:

```
1 ✓ SHOW DATABASES;
```

The result pane below shows the output of the query:

Database
information_schema
mysql
performance_schema
sys

Red circles with numbers 1 through 6 highlight specific elements:

1. The 'Database' button in the top right.
2. The connection dropdown in the top right.
3. The connection status in the top right.
4. The 'Run' button in the top left.
5. The connection name in the top left.
6. The 'Information Schema' entry in the result table.

## DB Schema vs. DB Instance

A database schema and a database instance are related concepts in the context of a database system, but they refer to different aspects:

**1. Database Schema:** The schema is the structure or blueprint that defines how data is organized within the database. It describes the database's logical design, including:

- **Tables:** The individual entities (e.g., Customers, Orders).
- **Columns:** The attributes of each entity (e.g., CustomerID, OrderDate).
- **Constraints:** Rules applied to the data (e.g., primary keys, foreign keys, unique constraints).
- **Relationships:** How tables relate to one another (e.g., one-to-many, many-to-many relationships).
- **Views, indexes, triggers, and stored procedures:** Other elements that define how the data is accessed or managed.

Think of the schema as the blueprint or structure of the database that remains mostly static or changes rarely.

**2. Database Instance:** The instance refers to the actual data stored in the database at a particular point in time. It is the current state of the database, including all of the records or entries inside the tables defined by the schema.

It includes actual rows of data in the tables. The instance can change frequently as data is inserted, updated, or deleted. You can think of the instance as the current snapshot of the database contents, while the schema defines the permanent framework.

Analogy: The schema is like the blueprint of a building (defining rooms, walls, and layout), while the instance is like the current state of the building (furniture, people inside, and their positions, which can change).

## Commands

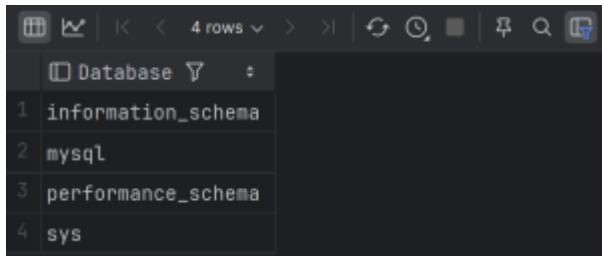
### SHOW DATABASES

Source: [Alle verfügbaren Datenbanken anzeigen lassen](#)

To show all currently available databases we use the SHOW command. The basic syntax is as shown below:

```
SHOW DATABASES;
```

After executing this command, we will receive the list of currently installed database instances, as shown in the following picture.



Database	Engine
information_schema	
mysql	
performance_schema	
sys	

### CREATE DATABASES

Source: [CREATE DB](#)

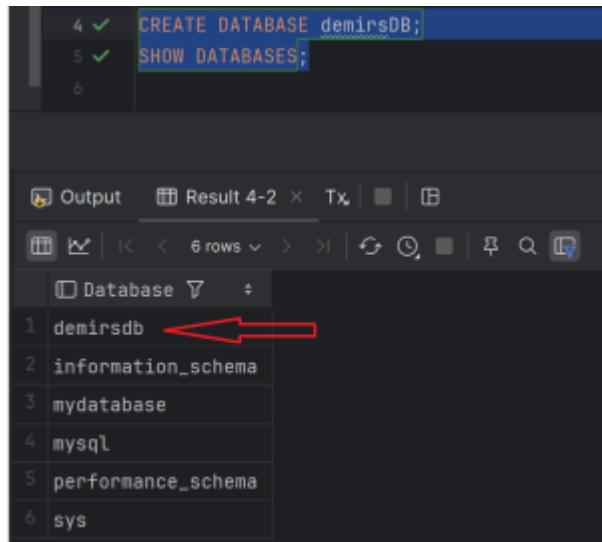
A new application is basically constructed within a new environment. Therefore we need to create a new database instance for us. To achieve this, which is done by the following command syntax.

```
CREATE DATABASE myDatabase;
```

So, if we want to create a new database for a webshop for our customer „Demir“, we need to type in:

```
CREATE DATABASE DemirsDB;  
SHOW DATABASES;
```

After executing both commands, our result-set would look like:



```
4 ✓ CREATE DATABASE demirsDB;
5 ✓ SHOW DATABASES;
6
```

Output Result 4-2 × Tx | | | | |

Database
demirsdb
information_schema
mydatabase
mysql
performance_schema
sys

## DROP DATABASES

Source: [DROP DB](#)

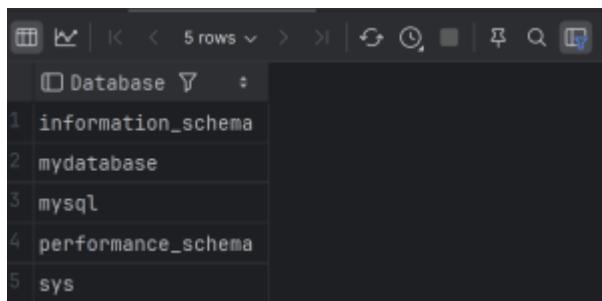
It can happen, that our database is corrupted, obsolete or otherwise not required any more. Thus, we have to delete it, which is done by the command

```
DROP DATABASE myDatabase;
```

So, if we want to delete demirsDB, we need to type in the following SQL commands.

```
DROP DATABASE demirsDB;
SHOW DATABASES;
```

After, executing both commands, the result-set would look like:



Database
information_schema
mydatabase
mysql
performance_schema
sys

## Using databases

Most database systems usually have more than one instance (project areas) running, e.g. DIGITECH.CH and GALAXUS.CH, these two database instances (storage areas for different purposes) are running on the same database systems, but organize their data separately.

As a database administrator, we cannot work on different instances at the same time, so we have to choose which instance we actually want to work on. The SQL statement in question is the USE command.

As a database administrator, we cannot work on different instances at the same time, so we have to choose which instance we actually want to work on. The SQL statement in question is the USE command.

```
7
10  CREATE DATABASE demirsDB1;
11  CREATE DATABASE demirsDB2;
12  CREATE DATABASE demirsDB3;
13  SHOW DATABASES;
14 ✓ USE demirsDB3;
```

## Vocabulary

English	Deutsch
Obsolete	veraltet
result-set	Ergebnismenge



Volkan Demir

From:

<https://wiki.bzz.ch/> - BZZ - Modulwiki

Permanent link:

<https://wiki.bzz.ch/modul/m290/learningunits/lu05/theorie/01?rev=1729766775>

Last update: **2024/10/24 12:46**

