

Introduction to MySQL Workbench

(LAB Lecture)



Department of Information Technology
Database Systems II (IT226)
Spring 2024-2025
Week 1&2 – February 9-16, 2025
Lecturer: Soma Soleimanzadeh



1

Contents

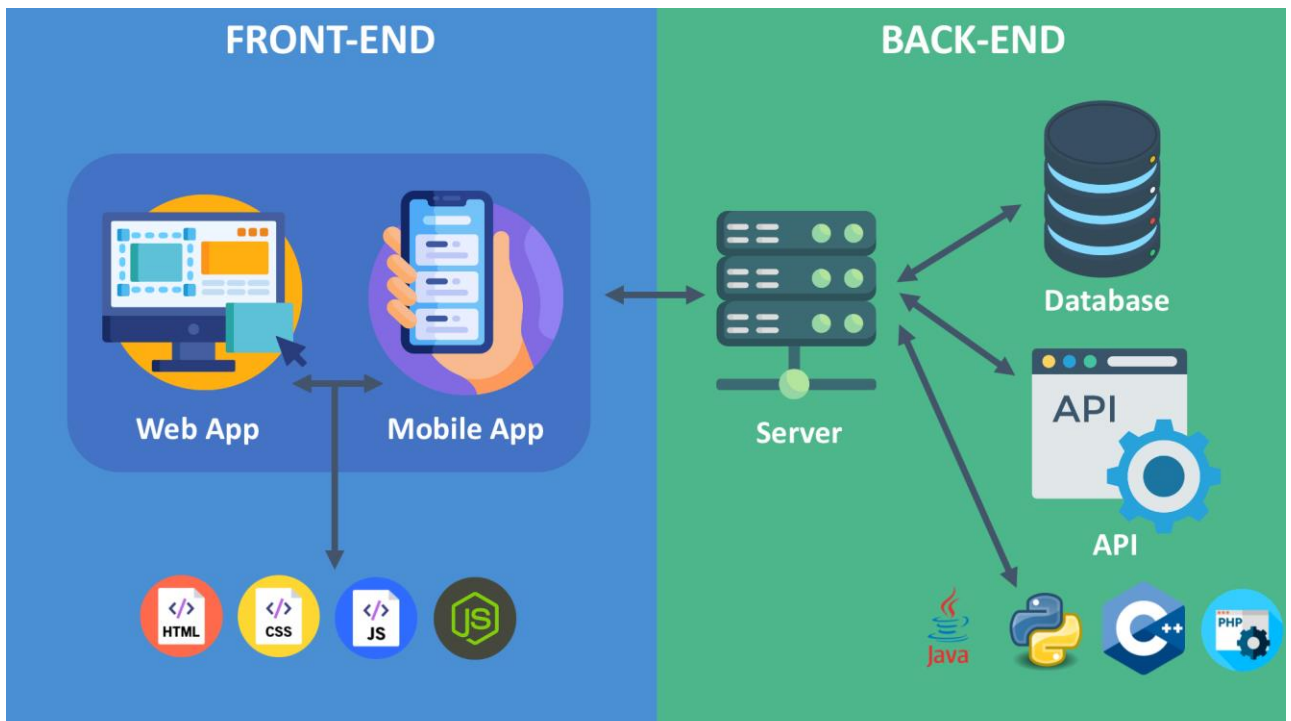
- MySQL and MySQL Workbench
- Create a Database and Create Table
- Modifying the Structure of Table
- Alter Table
 - Add a New Column
 - Drop an Existing Column
 - Modifying the Data Type of a Column
 - Renaming a Column or Table

2

RDBMS Software



3



MySQL

- **MySQL** is the world's most popular open-source database management system.
- **MySQL** is an SQL-based relational database designed to store and manage structured data.
- **MySQL** is a popular database management system used in web development as a web application backend.



SQL Language

- **SQL (Structured Query Language)** is used for
 - managing and manipulating relational databases.
- **SQL** is designed to interact with databases
 - by defining, querying, updating, and managing the data within them.

Microsoft Access vs. MySQL

MS Access	MySQL
Slower, small number of users	Quicker, better performance for many users
Single login and password	Multiple security options
Can handle lower numbers of queries & payment transactions	Better for eCommerce - can support multiple queries and payment transactions
Cost per user licence can become expensive	No licences, more cost effective
Has a maximum storage and user number	Easily scalable for a growing business

7

MySQL Key Benefits

- **Ease of Use**
- **Reliability**
- **Scalability**
- **Performance**
- **High Availability**
- **Security**
- **Flexibility**

8

MySQL Workbench

- **MySQL Workbench** is a graphical tool for working with **MySQL** servers and databases.

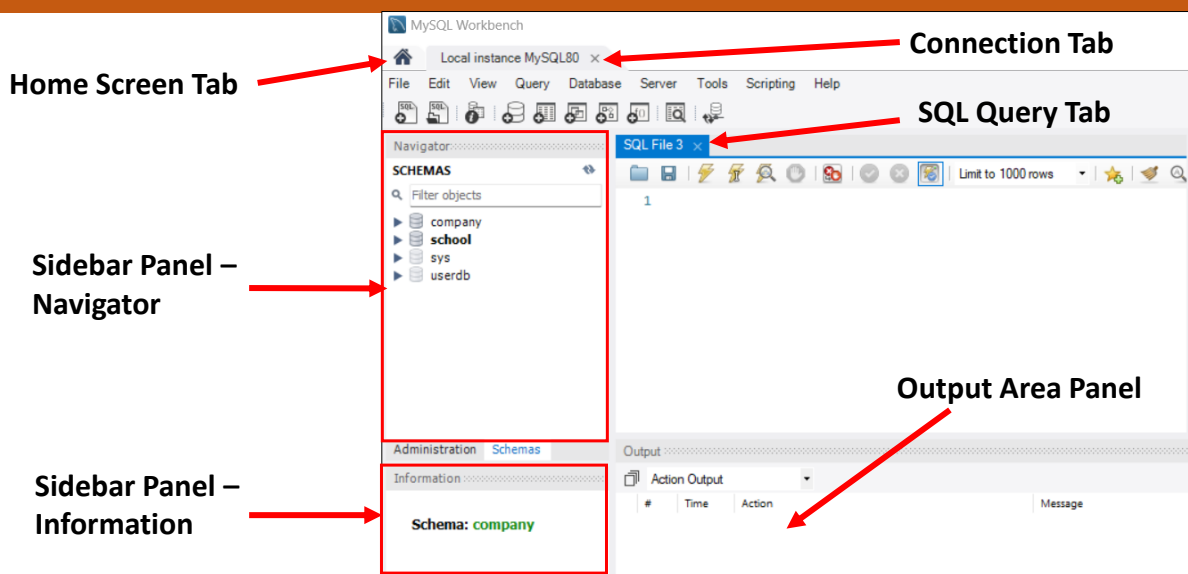
- **MySQL Workbench** is available in two editions:

- **Community** Edition → free of charge
- **Commercial** Edition → provides advanced features, not free.

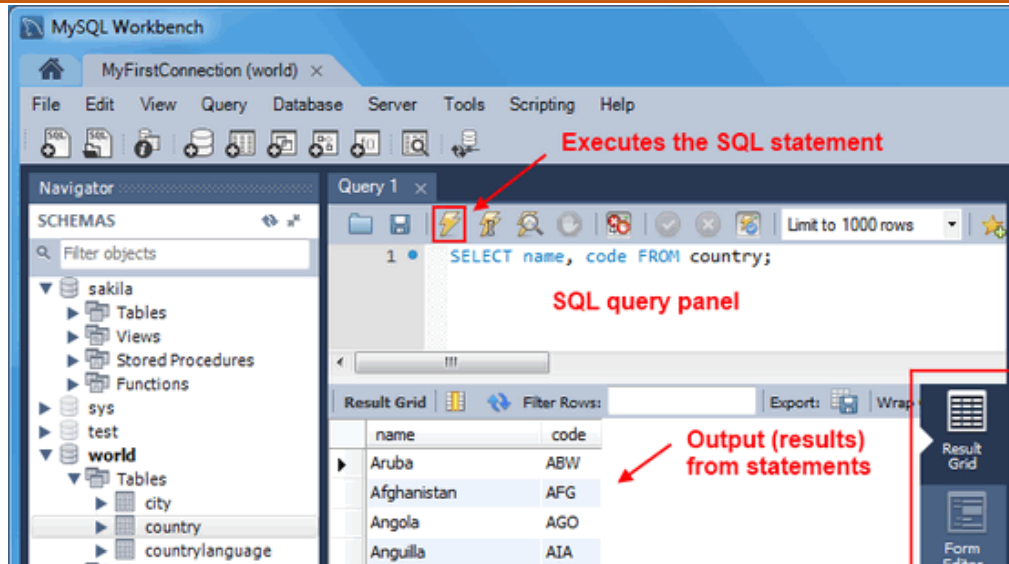


9

MySQL Workbench Environment



SQL Editor –SQL Query Tab



11

Some Data Types in MySQL

Data Type	Description	Storage (Bytes)
INT	A standard integer	4 Bytes
DECIMAL	A fixed-point number	4 Bytes
NUMERIC	A fixed-point number	4 Bytes
FLOAT	A floating-point number	4 Bytes
DOUBLE	A floating-point number that stores larger values than FLOAT	8 bytes

12

Some Data Types in MySQL

Data Type	Description	Number of Characters
CHAR (M)	A fixed-length string	M → 0-255
VARCHAR (M)	A variable-length string	M → 0-255
TEXT (M)	A long string	M → 0-65,535

13

Some Data Types in MySQL

Data Type	Description	Storage
DATE	A date	3 Bytes
DATETIME	A date and time combination	8 Bytes
TIME	A time	3 Bytes
YEAR	A year in four-digit format	1 Byte
TIMESTAMP	A timestamp	4 Bytes

14

Create and Use a Database

- Creating a new database in MySQL:

create database <database_name> ;

- Selecting the Database for Use:

use <database_name> ;

- Show list of all databases:

show databases;

15

First Database – Company Database

Department

deptName	Budget	Floor
HR	11000	3
Accounting	7000	1
Customer Services	5000	4

Employee

ID	name	salary	deptName
1	Mike	4000.5	Accounting
2	Kate	2500.75	Customer Services
3	Sandy	7000	HR

16

Create Table

- Creating a new table in MySQL:

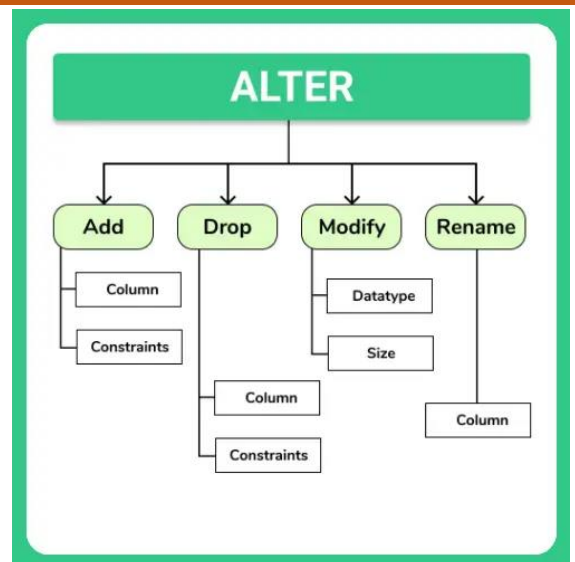
```
create table <table_name>  
  
  (column_names  data_type,  
  
   constraints);
```

17

ALTER TABLE

- After creating a table, you can change the table's structure by using **alter table** command.

```
alter table table_name  
  
[any modification];
```



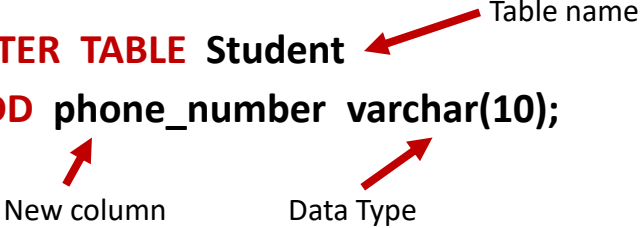
Adding a New Column to the Table

- Syntax:

```
ALTER TABLE table_name  
ADD column_name data_type;
```

- Example:

```
ALTER TABLE Student  
ADD phone_number varchar(10);
```



New column

Data Type

	stuld	lastName	firstName	major	credits	phone_number
1	S1001	Smith	Tom	History	90	NULL
2	S1002	Chin	Ann	Mathematics	36	NULL

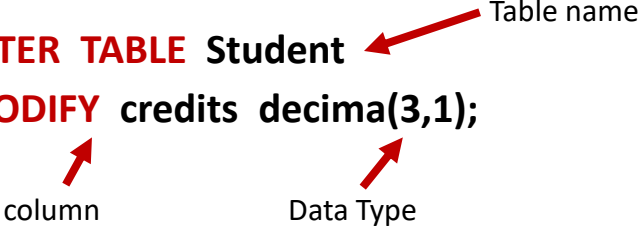
Modifying Data Type of a Column

- Syntax:

```
ALTER TABLE table_name  
MODIFY column_name data_type;
```

- Example:

```
ALTER TABLE Student  
MODIFY credits decima(3,1);
```



column

Data Type

	stuld	lastName	firstName	major	credits
1	S1001	Smith	Tom	History	90.0
2	S1002	Chin	Ann	Mathematics	36.0

Drop an Existing Column from a Table

- Syntax:

```
ALTER TABLE table_name  
DROP column_name;
```

- Example:

```
ALTER TABLE Student  
DROP credits;
```

Table name

column

	stuld	lastName	firstName	major	credits
1	S1001	Smith	Tom	History	90
2	S1002	Chin	Ann	Mathematics	36

Renaming a Column

- Syntax:

```
ALTER TABLE table_name  
RENAME COLUMN old_name TO new_name;
```

- Example:

```
ALTER TABLE Student  
RENAME COLUMN major TO deptName;
```

	stuld	lastName	firstName	deptName	credits
1	S1001	Smith	Tom	History	90
2	S1002	Chin	Ann	Mathematics	36
3	S1004	Smith	Jack	English	75

Renaming a Table

- **Syntax:**

```
ALTER TABLE table_name  
RENAME TO new_table_name;
```

- **Example:**

```
ALTER TABLE Faculty  
RENAME TO Teacher;
```

23

Let's Try it!

- Create a database named **university** and **activate** it.
- Create the following tables in the **university** database.
student (SID , Sname, age)
teacher (TID , Tname, salary)
advisor (StudentId, TeacherId, projectTitle)
- Make **SID** and **TID** auto_increment starting from 1.
- Make sure **Sname** and **Tname** values can not be Null.

24

Create Tables and Entering Data

- Make **SID** and **TID** auto_increment starting from 1.
- Make sure **Sname** and **Tname** values can not be Null.

student	<u>SID</u>	Sname	age
	1	John	20
	2	Peter	21

Teacher	<u>TID</u>	Tname	salary
	1	Lara	30000
	2	James	24000

Advisor	<u>StudentID</u>	<u>TeacherID</u>	projectTitle
	1	Lara	30000
	2	James	24000

25

Let's Try Alter Table Command

- Add a new column to **Teacher** table, showing the rank of teacher.
- Change the data type of **salary** column from int to decimal(7,1).
- Drop **age** column from **student** table.
- Rename **salary** column in **teacher** table to **income**.

26

Let's Try it!

- Create a database named **hospital** and **activate** it.
- Create the following tables in the **hospital** database.
 - doctor** (doctorID , Dname, specialization)
 - patient** (patientID , Pname, age)
 - visit** (patientID, doctorID, visitDate)
- Make **doctorID** and **patientID** auto_increment starting from 1.
- Make sure **specialization** and **Pname** values can not be Null.

27

Create Tables and Entering Data

- Make **doctorID** and **patientID** auto_increment starting from 1.
- Make sure **specialization** and **Pname** values can not be Null.

doctor	<u>doctorID</u>	Dname	specialization	patient	<u>patientID</u>	Pname	age
	1	John	General Surgery		1	Lara	67
	2	Peter	Dermatology		2	James	45

visit	<u>patientID</u>	<u>DoctorID</u>	visitDate
	1	1	2/5/2024
	2	1	6/10/2024

28

Let's Try Alter Table Command

- Add a new column to **Patient** table, showing the job of patient.
- Change the data type of **visitDate** column from **date** to **datetime**.
- Drop **age** column from **patient** table.
- Rename **specialization** column in **doctor** table to **field**.

29