Database Fundamentals

Cybersecurity Department

Course Code: ***CBS 213***

Practical Lecture 2: Creating Databases and Tables

Halal Abdulrahman Ahmed

Lecture Outline

- Overview of MySQL Workbench and MySQL Server
- SQL Editor interface and basic commands
- Creating databases (graphically and with SQL)
- Creating tables and adding data
- Viewing and managing data in tables
- Saving and reopening SQL scripts

Learning Outcomes

By the end of this lab, students will be able to:

- Connect to a **local MySQL server** and use the SQL editor.
- Create databases and tables using both GUI and SQL commands.
- Insert and retrieve data from tables.
- Save and execute SQL files in Workbench.

Introduction to MySQL Workbench

•MySQL Workbench is a **graphical tool (GUI)** that allows you to work easily with MySQL Server.

•Instead of writing all commands in the Terminal, you can:

•Create and design databases visually.

•Write and run SQL queries.

•See your data in tables.

•Import or export data files.

•Manage users and passwords.

Understanding MySQL Server –

- **MySQL Server** is the **main program** that stores and manages all your data. It is the heart of the Database System.
- It is also called the **database engine** because it does all the real work.
- It runs **in the background** and waits for commands from the user through MySQL Workbench or other apps.

MySQL Server vs MySQL Workbench

MySQL Server

The “engine” that stores and manages databases.

Runs in the background on your computer.

Responds to SQL commands.

MySQL Workbench

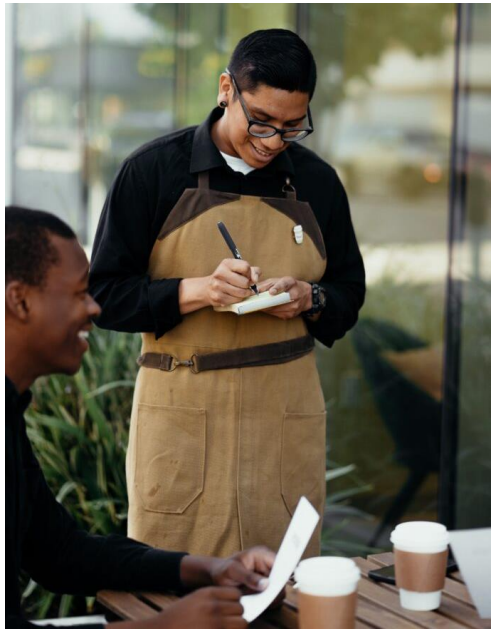
The “tool” used to control and visualize the databases.

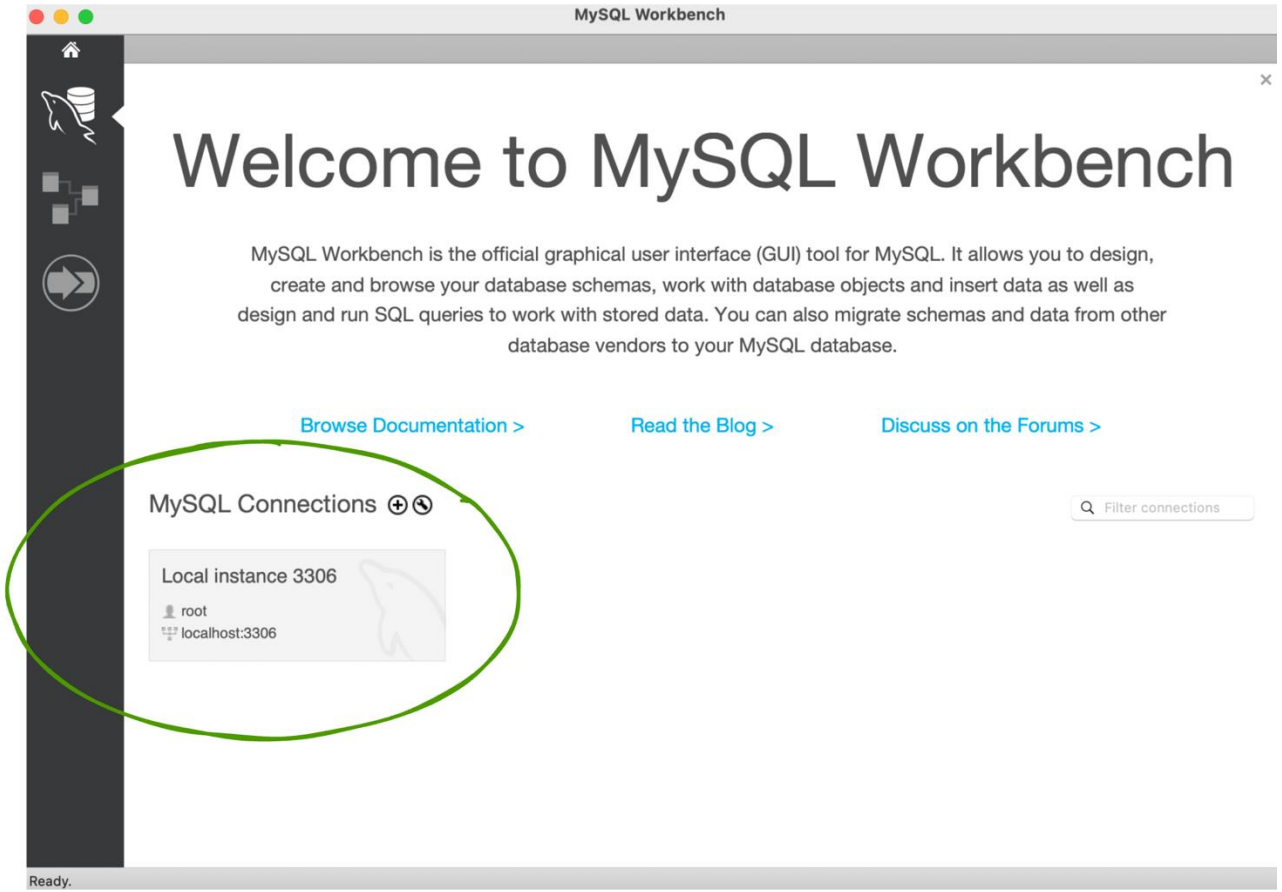
Runs in the foreground (you see the interface).

Sends SQL commands for you and shows results nicely.

MySQL Workbench is the bridge between the user and the MySQL Server. The server is where data lives; Workbench is the visual tool you use to interact with it.

- **MySQL Server = Kitchen** (where food = data is cooked).
- **MySQL Workbench = Waiter** (who takes your order and brings back the food).



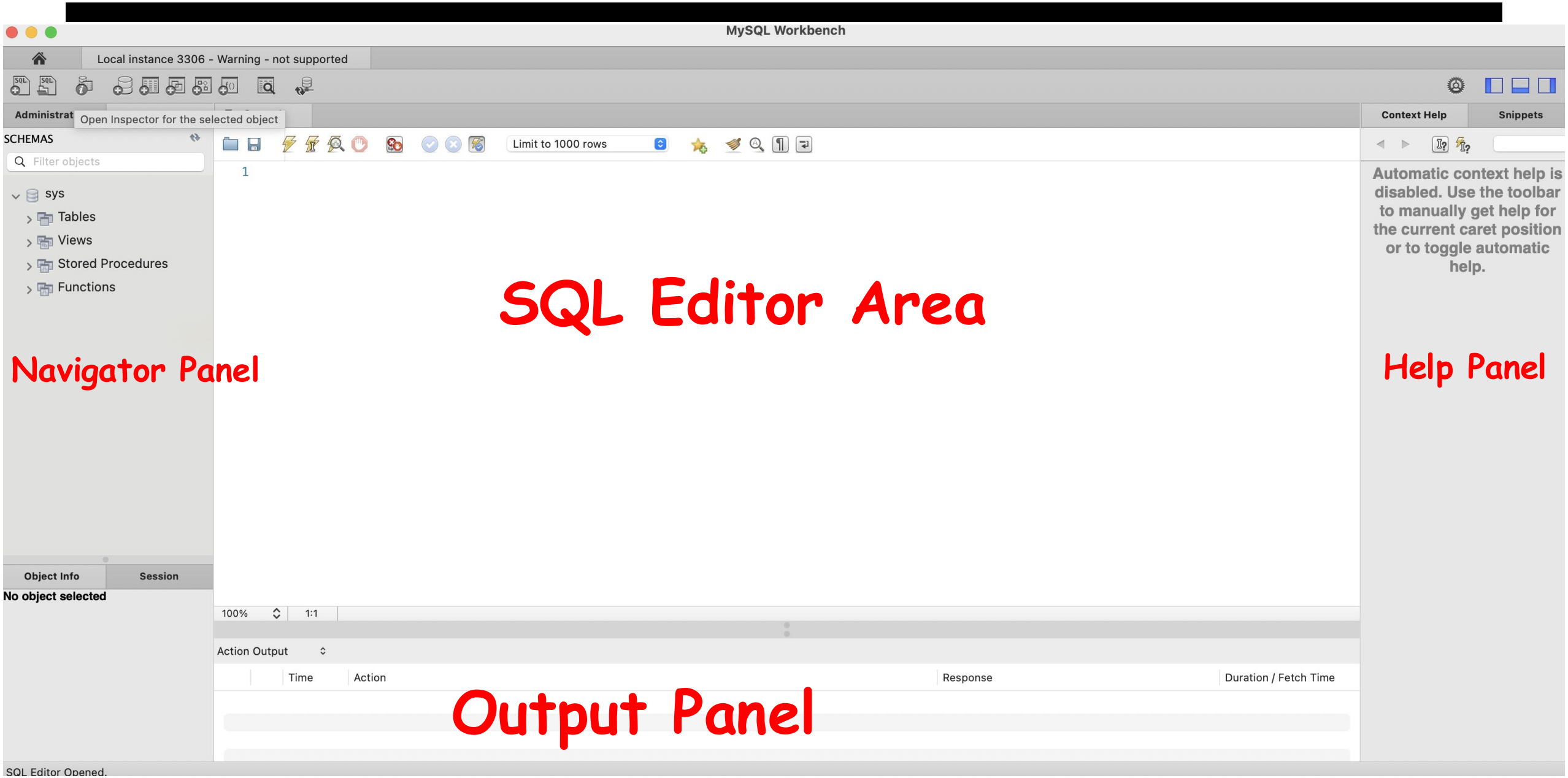


- **Local instance 3306** = your MySQL Server running on your own computer.
- **localhost** → means "this computer."
- **3306** → is the **port number** (like a door that connects Workbench to the Server).
- **root** → is the **administrator user** (full permissions).

The SQL Editor Workspace

- It has several important areas:
- **Toolbar (top)** – run, stop, save, and open SQL scripts.
  - ⚡ Run → executes your SQL commands.
  - 📁 Open / 💾 Save → work with .sql files.

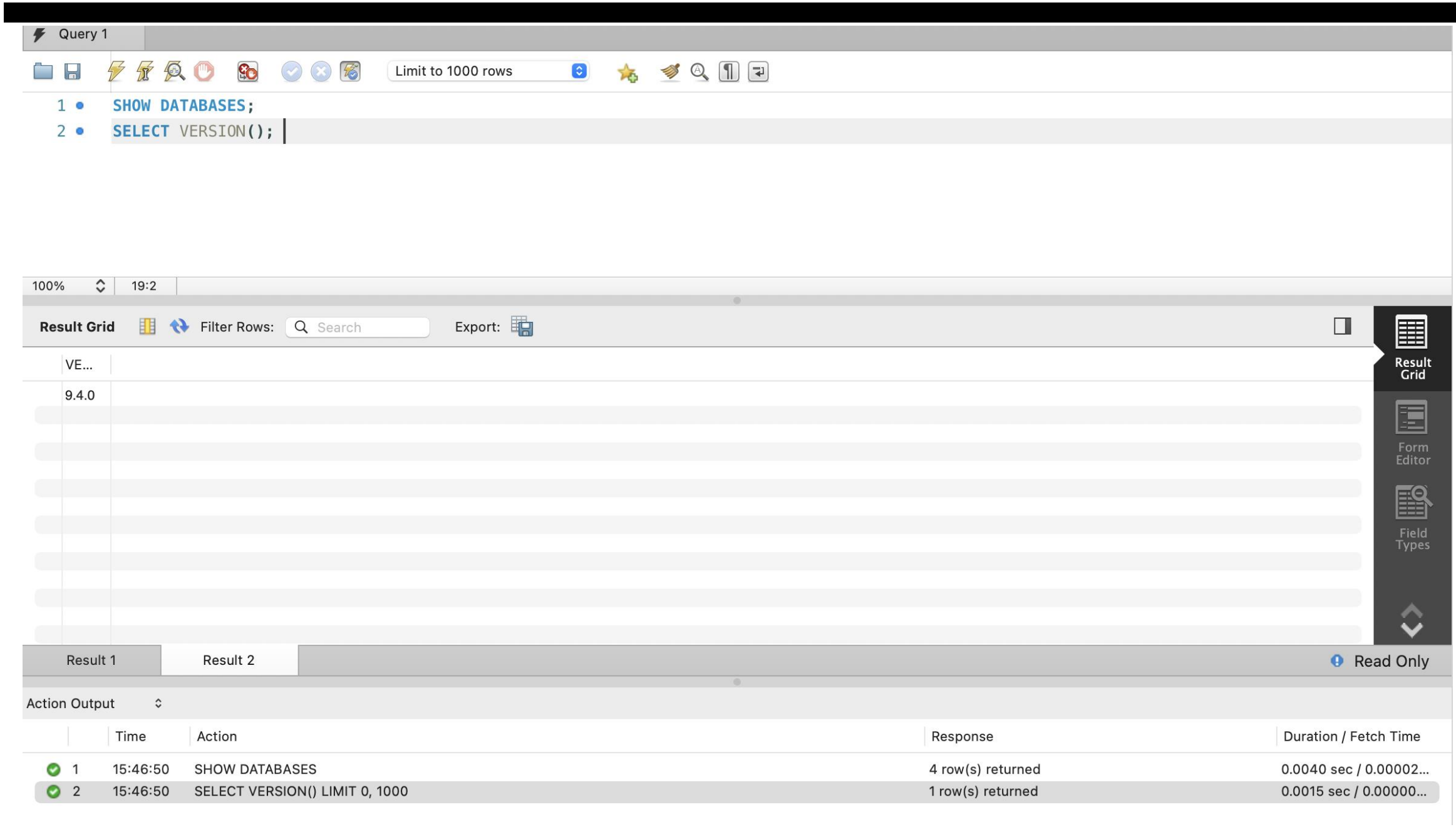
- **SQL Editor Area (middle)**: type your SQL commands here.
- **Output Panel (bottom)**: shows the results or error messages.
- **Navigator Panel (left)**: manage schemas, tables, users, and server status.
- **Help Panel (right)**: shows help or code snippets.


MySQL Workbench
Local instance 3306 - Warning - not supported
Open Inspector for the selected object
SCHEMAS
Filter objects
sys
Tables
Views
Stored Procedures
Functions
Limit to 1000 rows
1
Context Help
Snippets
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
SQL Editor Area
Navigator Panel
Help Panel
Object Info
Session
No object selected
100%
1:1
Action Output
Time
Action
Response
Duration / Fetch Time
Output Panel
SQL Editor Opened.

Your First SQL Commands

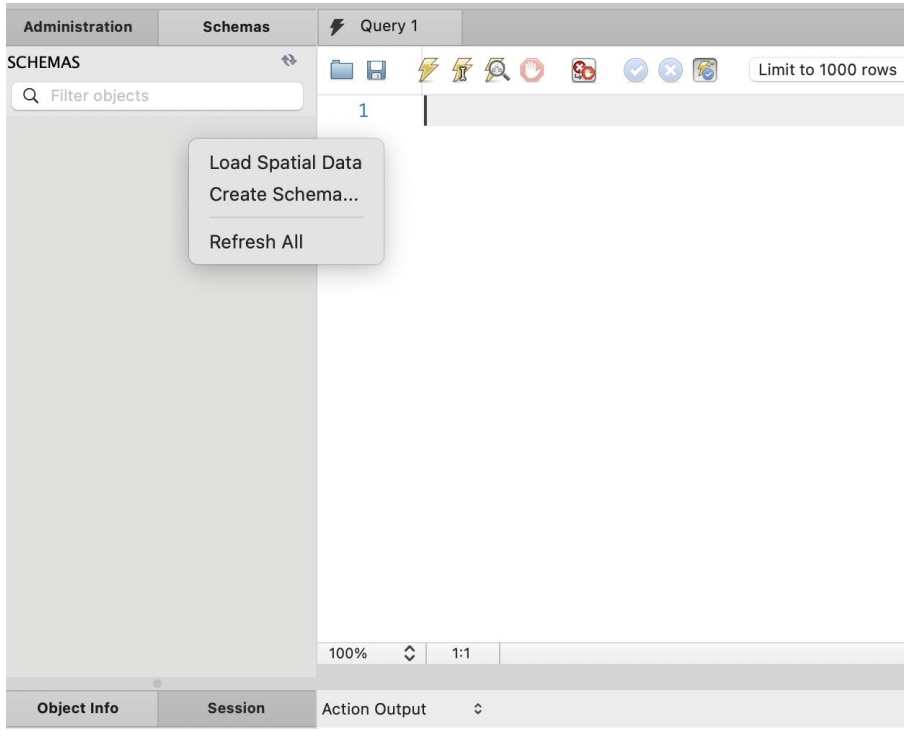
```
SHOW DATABASES;
SELECT VERSION();
```

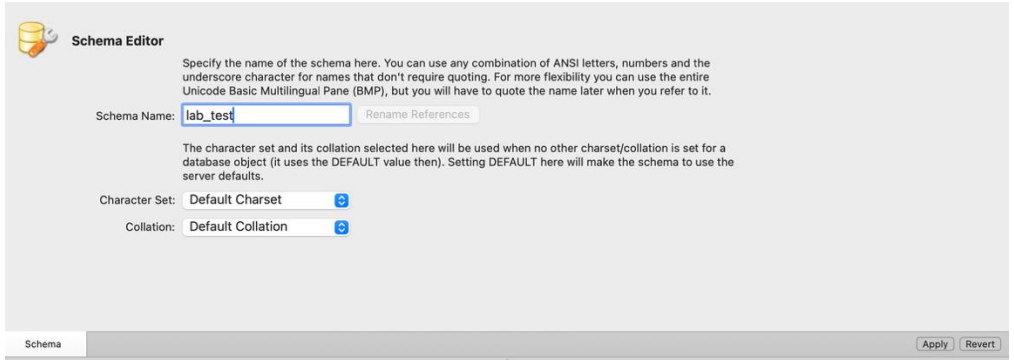
- SHOW DATABASES; → lists all existing databases on the server.
- SELECT VERSION(); → shows the version of MySQL installed.


Query 1
Limit to 1000 rows
1 SHOW DATABASES;
2 SELECT VERSION();
100%
19:2
Result Grid
Filter Rows:
Search
Export:
VE...
9.4.0
Result Grid
Form Editor
Field Types
Result 1
Result 2
Read Only
Action Output
Time
Action
Response
Duration / Fetch Time
1 15:46:50 SHOW DATABASES 4 row(s) returned 0.0040 sec / 0.00002...
2 15:46:50 SELECT VERSION() LIMIT 0, 1000 1 row(s) returned 0.0015 sec / 0.00000...

Creating a New Database (Graphical Way)

- Go to the **Schemas tab** (left).
- Right-click → **Create Schema** → enter name lab_test → Apply.

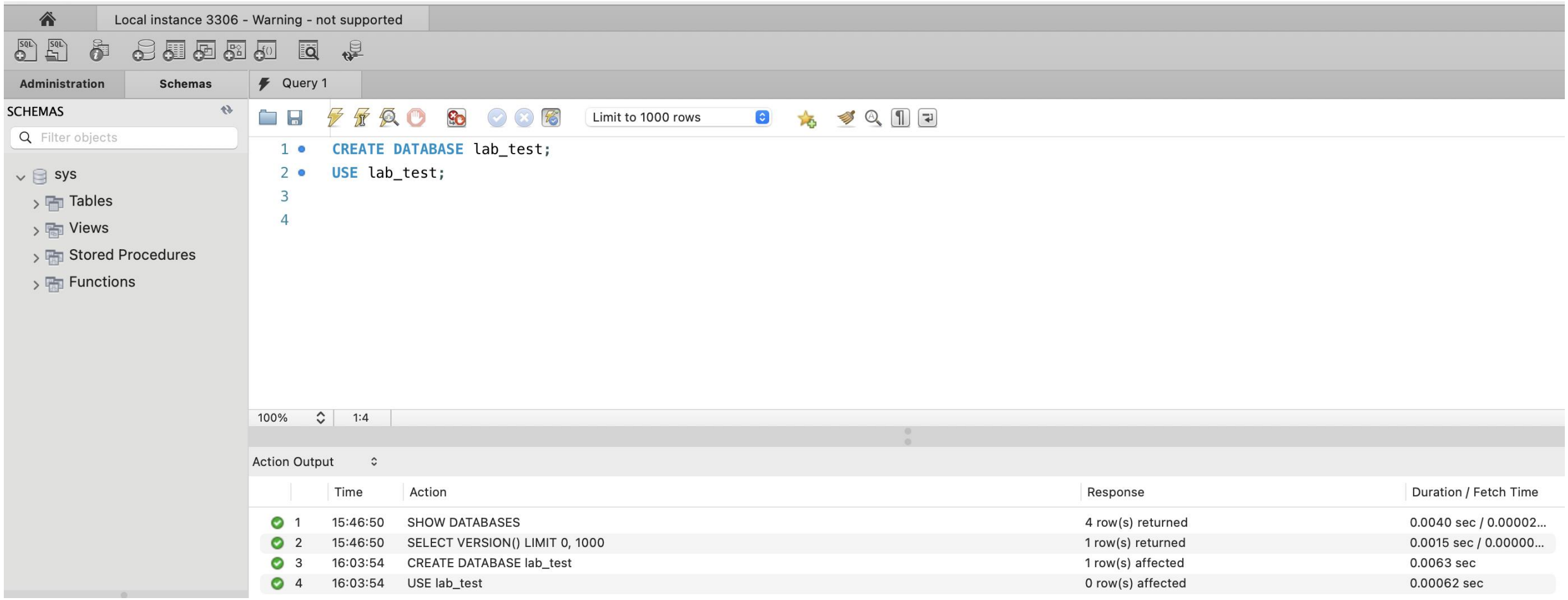




Creating a New Database (SQL Command way)

```
CREATE DATABASE lab_test;
USE lab_test;
```

- CREATE DATABASE lab_test;→ makes a new database.
- USE lab_test; → tells MySQL to start working inside that database.

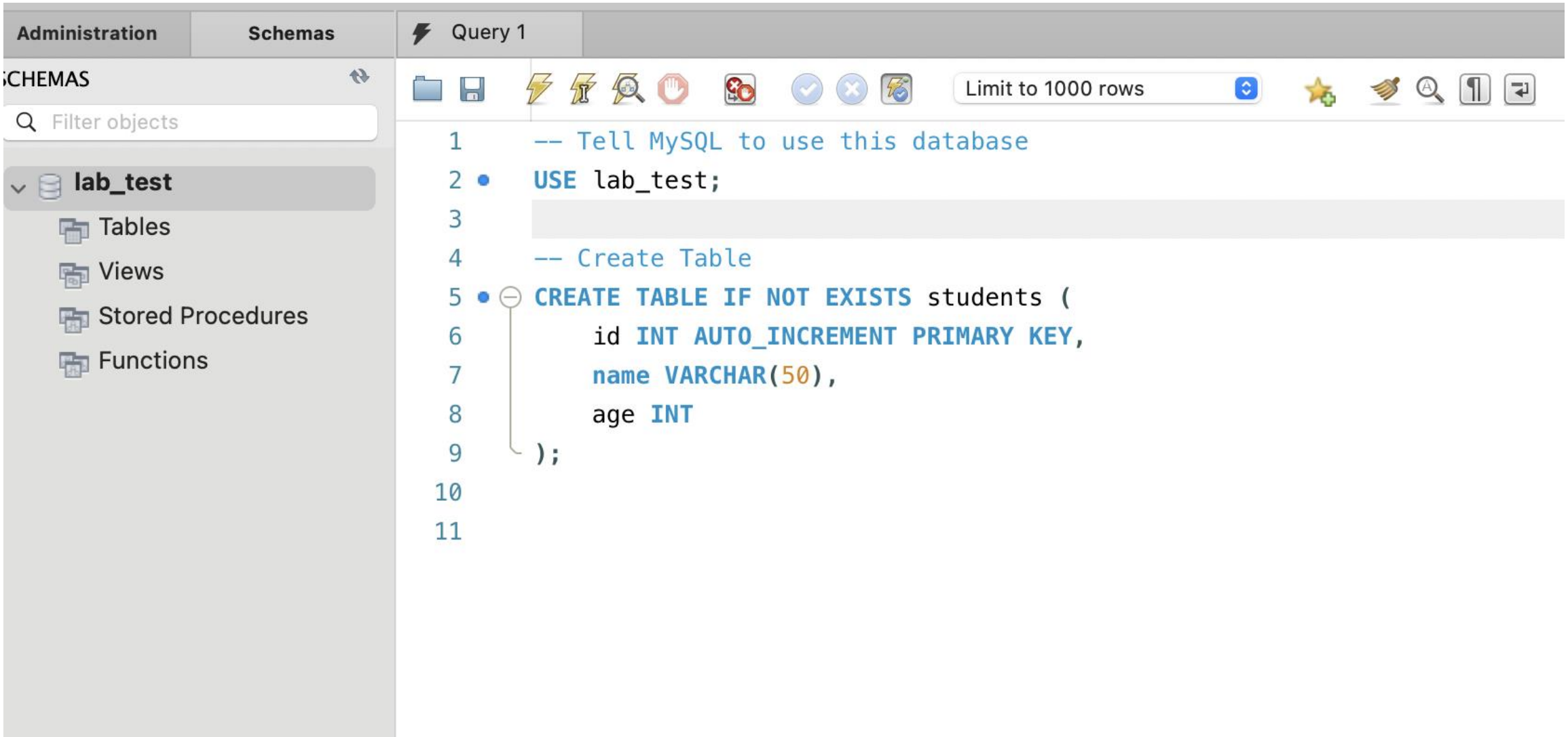

Local instance 3306 - Warning - not supported
Administration
Schemas
Query 1
SCHEMAS
Filter objects
sys
Tables
Views
Stored Procedures
Functions
Limit to 1000 rows
1 CREATE DATABASE lab_test;
2 USE lab_test;
3
4
100%
1:4
Action Output
Time
Action
Response
Duration / Fetch Time
1 15:46:50 SHOW DATABASES 4 row(s) returned 0.0040 sec / 0.00002...
2 15:46:50 SELECT VERSION() LIMIT 0, 1000 1 row(s) returned 0.0015 sec / 0.00000...
3 16:03:54 CREATE DATABASE lab_test 1 row(s) affected 0.0063 sec
4 16:03:54 USE lab_test 0 row(s) affected 0.00062 sec

Creating Tables & Adding Data

Create Table with SQL

```sql
CREATE TABLE IF NOT EXISTS students (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(50),
    age INT
);
```

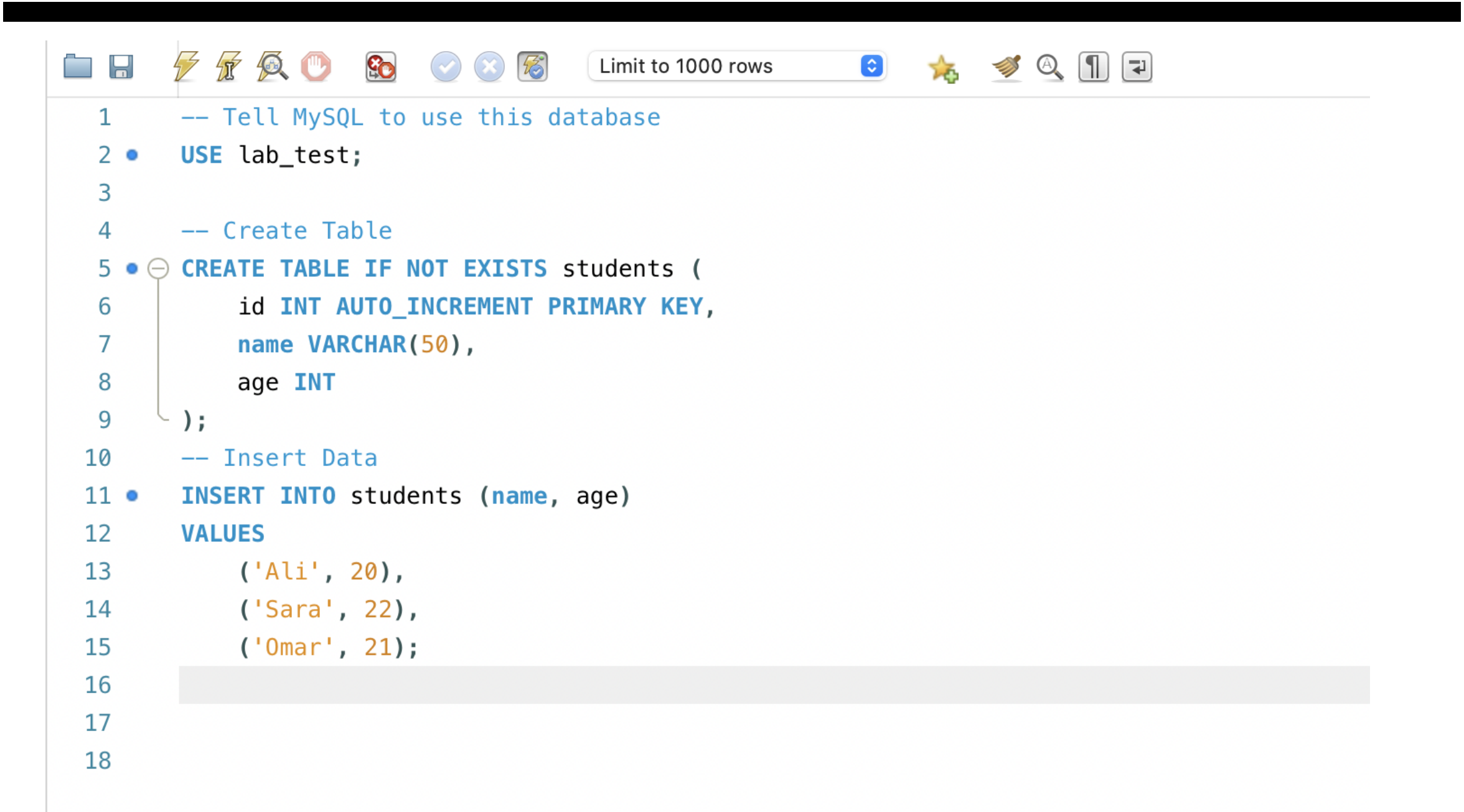
- CREATE TABLE → creates a new table inside your current database.
- IF NOT EXISTS → avoids an error if the table already exists.
- id INT AUTO_INCREMENT PRIMARY KEY → creates a unique ID that counts up automatically (1, 2, 3…).
- name VARCHAR(50) → a text column for names (up to 50 characters).
- age INT → a number column for age.


Administration
Schemas
Query 1
SCHEMAS
Filter objects
lab_test
Tables
Views
Stored Procedures
Functions
Limit to 1000 rows
1 -- Tell MySQL to use this database
2 USE lab_test;
3
4 -- Create Table
5 CREATE TABLE IF NOT EXISTS students (
6 id INT AUTO_INCREMENT PRIMARY KEY,
7 name VARCHAR(50),
8 age INT
9);
10
11

Insert Data into the Table

```sql
INSERT INTO students (name, age)
VALUES
    ('Ali', 20),
    ('Sara', 22),
    ('Omar', 21);
```

- INSERT INTO → adds new records (rows) to your table.
- (name, age) → tells MySQL which columns we're filling.
- VALUES → gives the actual data.
- Each set of parentheses ( ) represents one new row.


Limit to 1000 rows
1 -- Tell MySQL to use this database
2 USE lab_test;
3
4 -- Create Table
5 CREATE TABLE IF NOT EXISTS students (
6 id INT AUTO_INCREMENT PRIMARY KEY,
7 name VARCHAR(50),
8 age INT
9);
10 -- Insert Data
11 INSERT INTO students (name, age)
12 VALUES
13 ('Ali', 20),
14 ('Sara', 22),
15 ('Omar', 21);
16
17
18

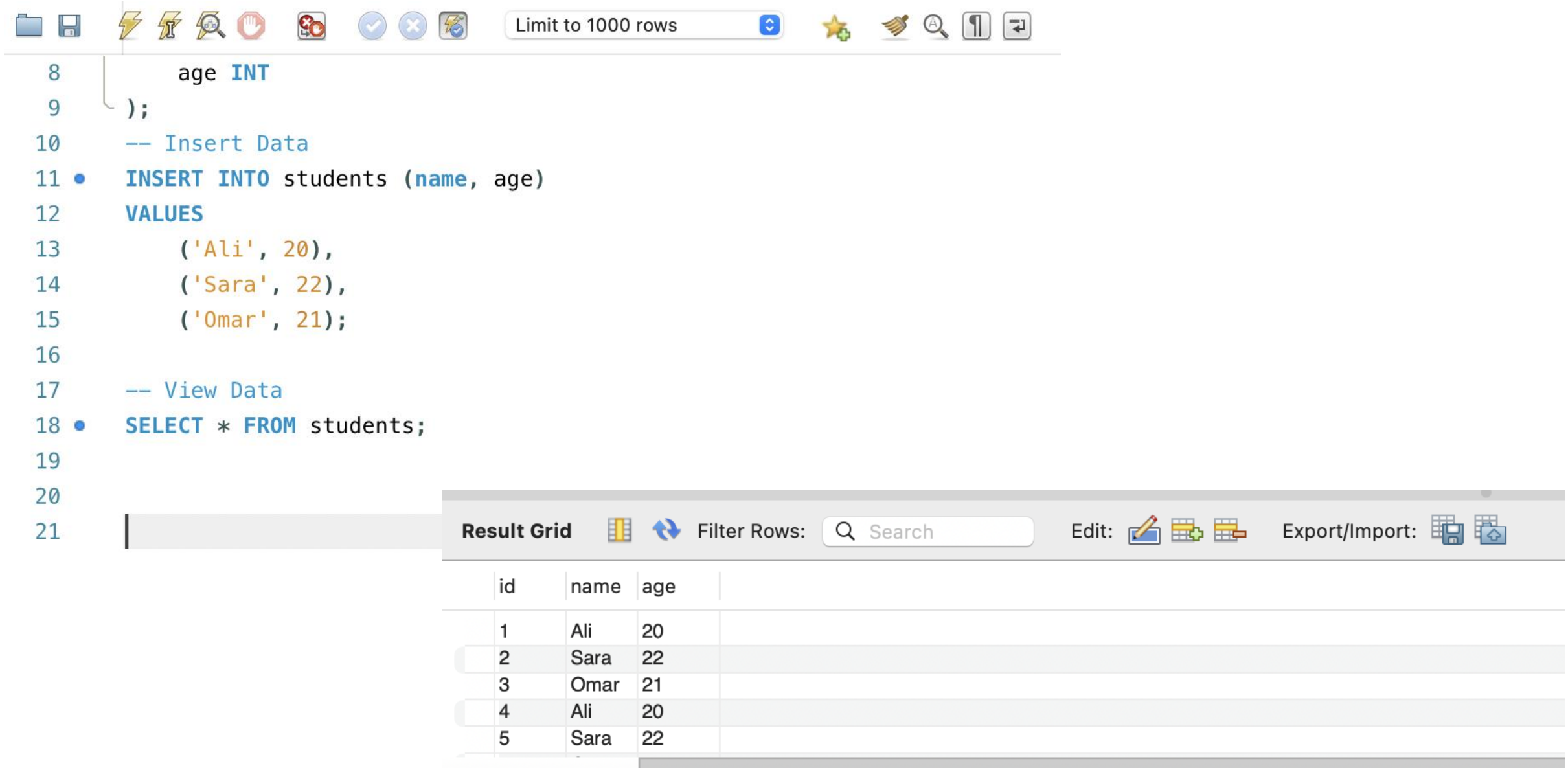
View the Data

```
SELECT * FROM students;
```

SELECT → retrieves data from the table.

* → means “all columns.”

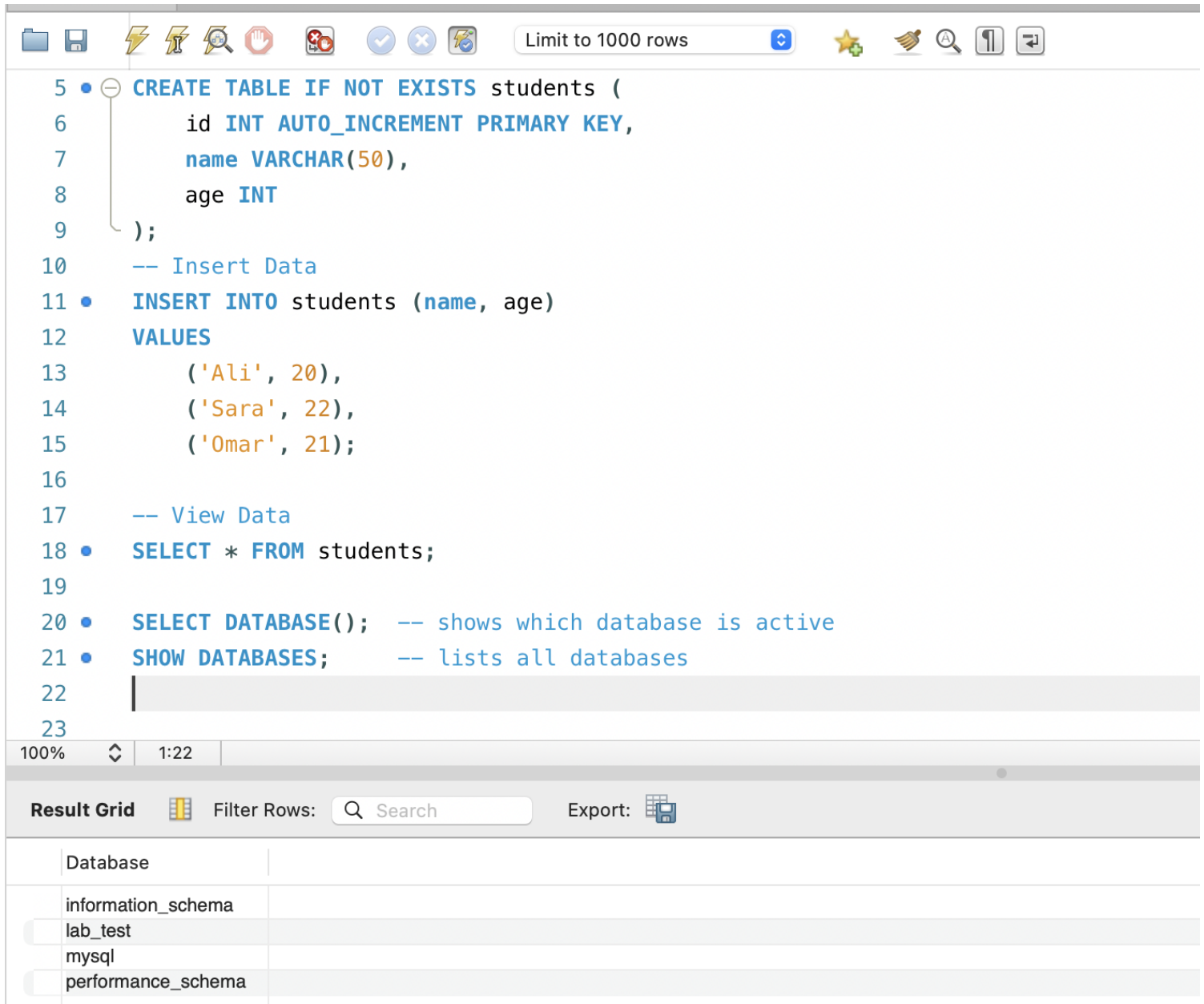
FROM students → tells MySQL which table to read from.


Limit to 1000 rows
8 age INT
9);
10 -- Insert Data
11 INSERT INTO students (name, age)
12 VALUES
13 ('Ali', 20),
14 ('Sara', 22),
15 ('Omar', 21);
16
17 -- View Data
18 SELECT * FROM students;
19
20
21
Result Grid
Filter Rows:
Search
Edit:
Export/Import:
id name age
1 Ali 20
2 Sara 22
3 Omar 21
4 Ali 20
5 Sara 22

Check the Active Database

```sql
SELECT DATABASE();  -- shows which database is active
SHOW DATABASES;     -- lists all databases
```

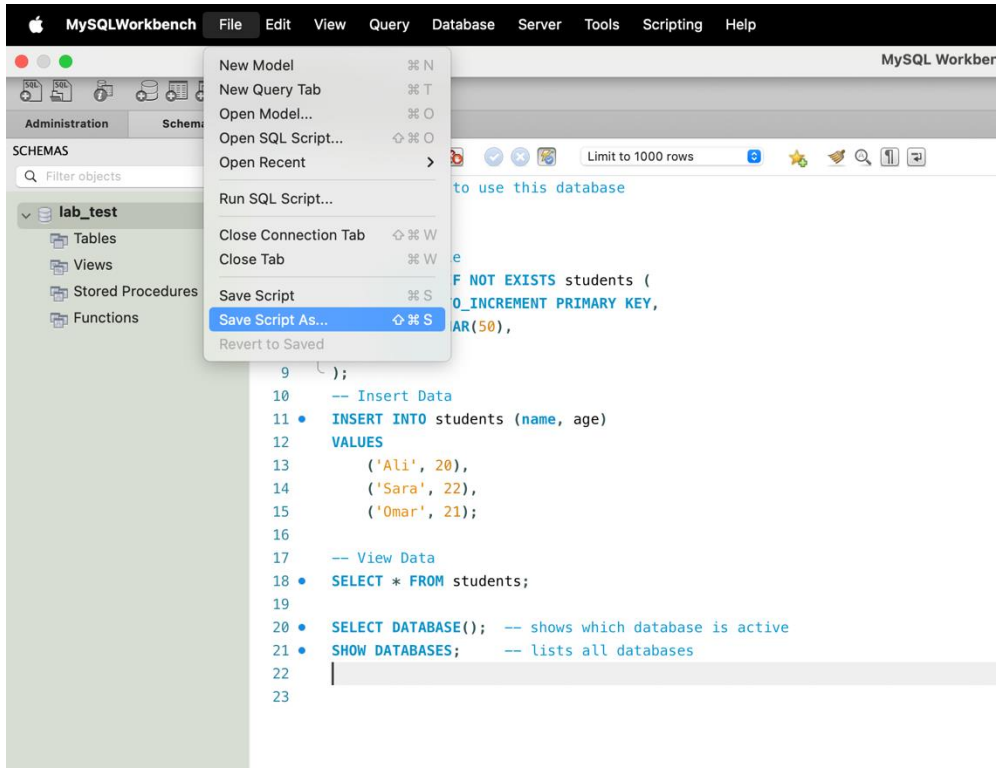
- SELECT DATABASE(); → tells you which database MySQL is currently using.
- SHOW DATABASES; → lists all databases available on your system.

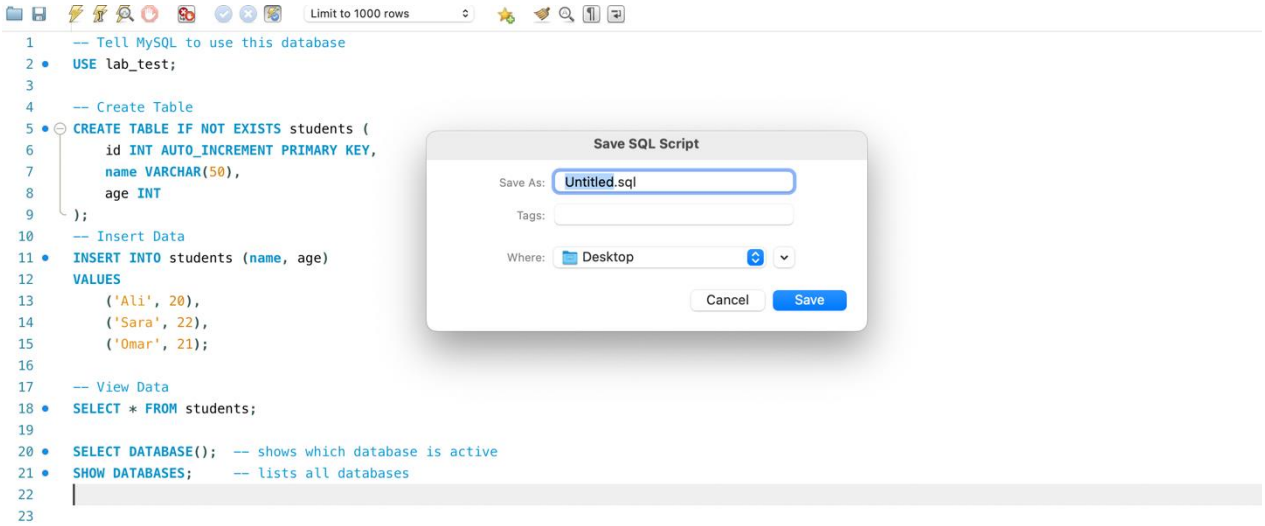

Limit to 1000 rows
5 CREATE TABLE IF NOT EXISTS students (
6 id INT AUTO_INCREMENT PRIMARY KEY,
7 name VARCHAR(50),
8 age INT
9);
10 -- Insert Data
11 INSERT INTO students (name, age)
12 VALUES
13 ('Ali', 20),
14 ('Sara', 22),
15 ('Omar', 21);
16
17 -- View Data
18 SELECT * FROM students;
19
20 SELECT DATABASE(); -- shows which database is active
21 SHOW DATABASES; -- lists all databases
22
23
100%
1:22
Result Grid
Filter Rows:
Search
Export:
Database
information_schema
lab_test
mysql
performance_schema

Saving in MySQL Workbench

Click **File** → **Save Script As…**

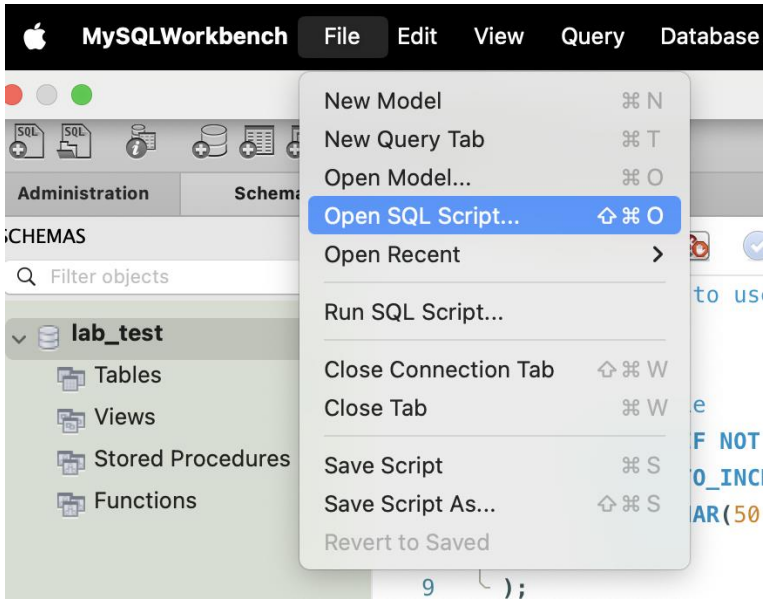
Choose where to save (e.g., *Documents* → *Database Labs*).





How to Open a Saved SQL File

- Open **MySQL Workbench**.
- Click **File → Open SQL Script…**
- Locate your saved file (e.g., Week2_Lab_MySQL.sql).
- Click **Open** → The code will appear in a new tab inside the SQL Editor.
- Click ⚡ **Run** to execute the commands again.



References

- Silva, B. (2021). *MySQL crash course: A hands-on introduction to database development*. No Starch Press.
- Grippa, V. M., & Kuzmichev, S. (2022). *Learning MySQL: Get a handle on your data* (2nd ed.). O'Reilly Media.


Any
Question