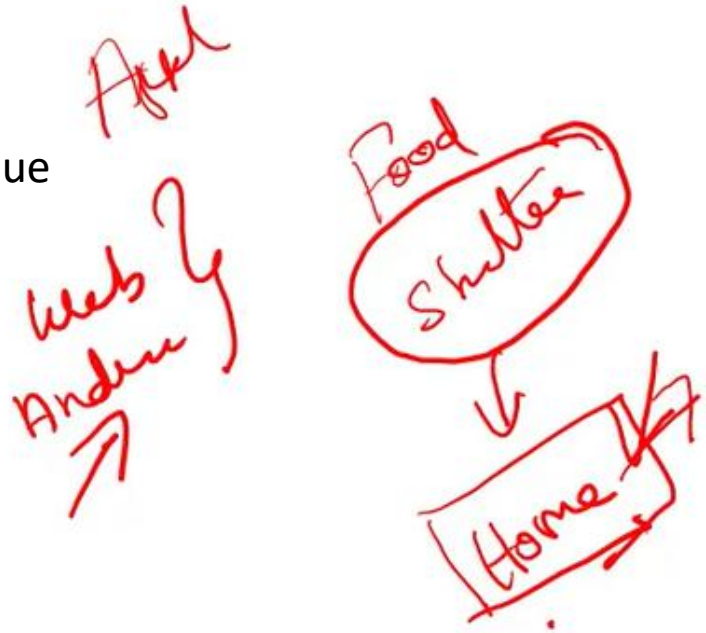
SQL- Structured Query Language

5. Find the total number of products sold and the total revenue generated for each product.

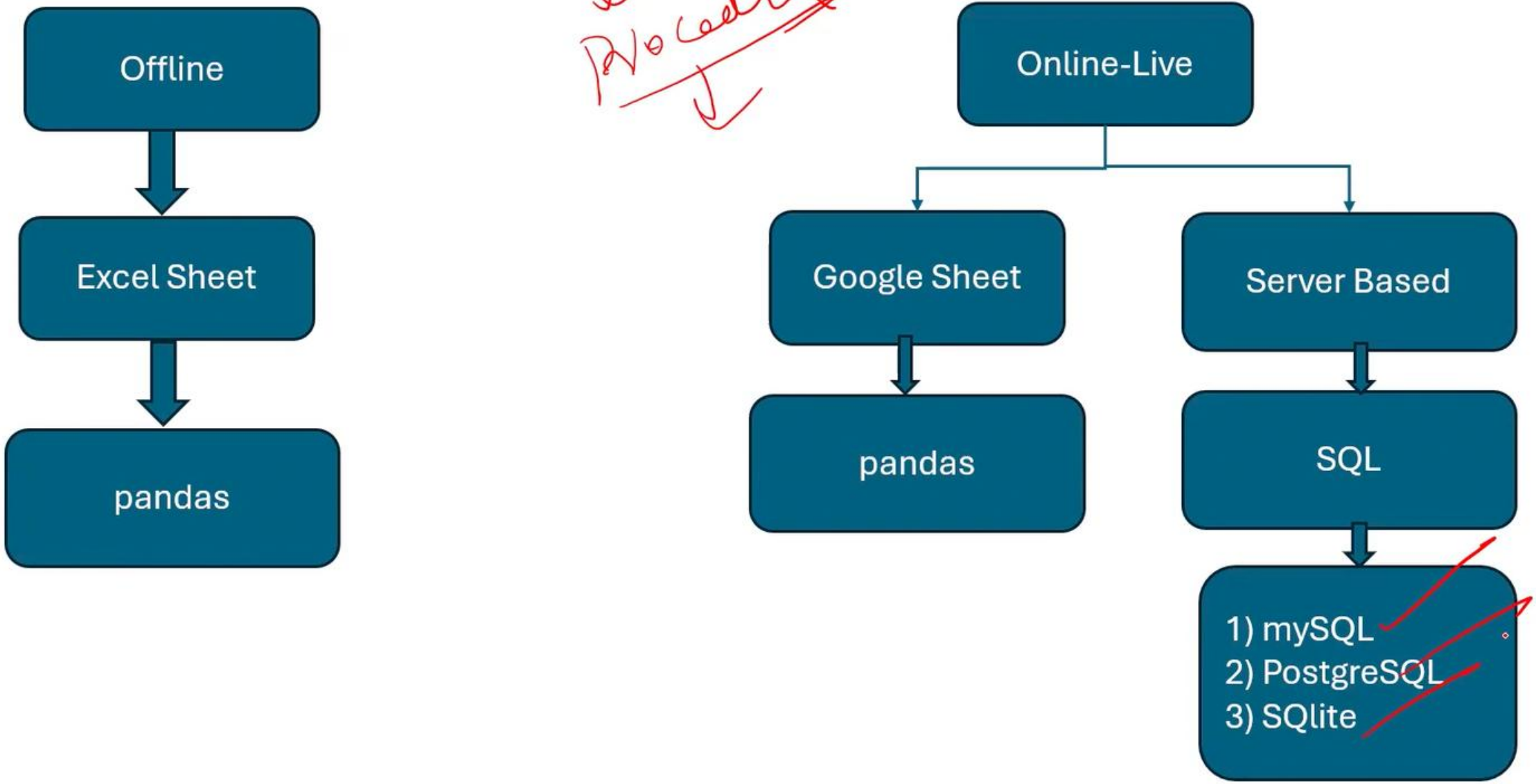
What is SQL?

&

Why SQL?



SQL- Structured Query Language



dev.mysql.com/downloads/

MySQL Community Downloads

- MySQL Yum Repository
- MySQL APT Repository
- MySQL SUSE Repository

- MySQL Community Server
- MySQL NDB Cluster
- MySQL Router
- MySQL Shell
- MySQL Operator
- MySQL NDB Operator
- MySQL Workbench

- C API (libmysqlclient)
- Connector/C++
- Connector/J
- Connector/NET
- Connector/Node.js
- Connector/ODBC
- Connector/Python
- MySQL Native Driver for PHP

- MySQL Benchmark Tool
- Time zone description

Accounts and Roles

Root Account Password

Enter the password for the root account. Please remember to store this password in a secure place.

MySQL Root Password: ••••

Repeat Password: ••••

Password strength: Weak

MySQL User Accounts

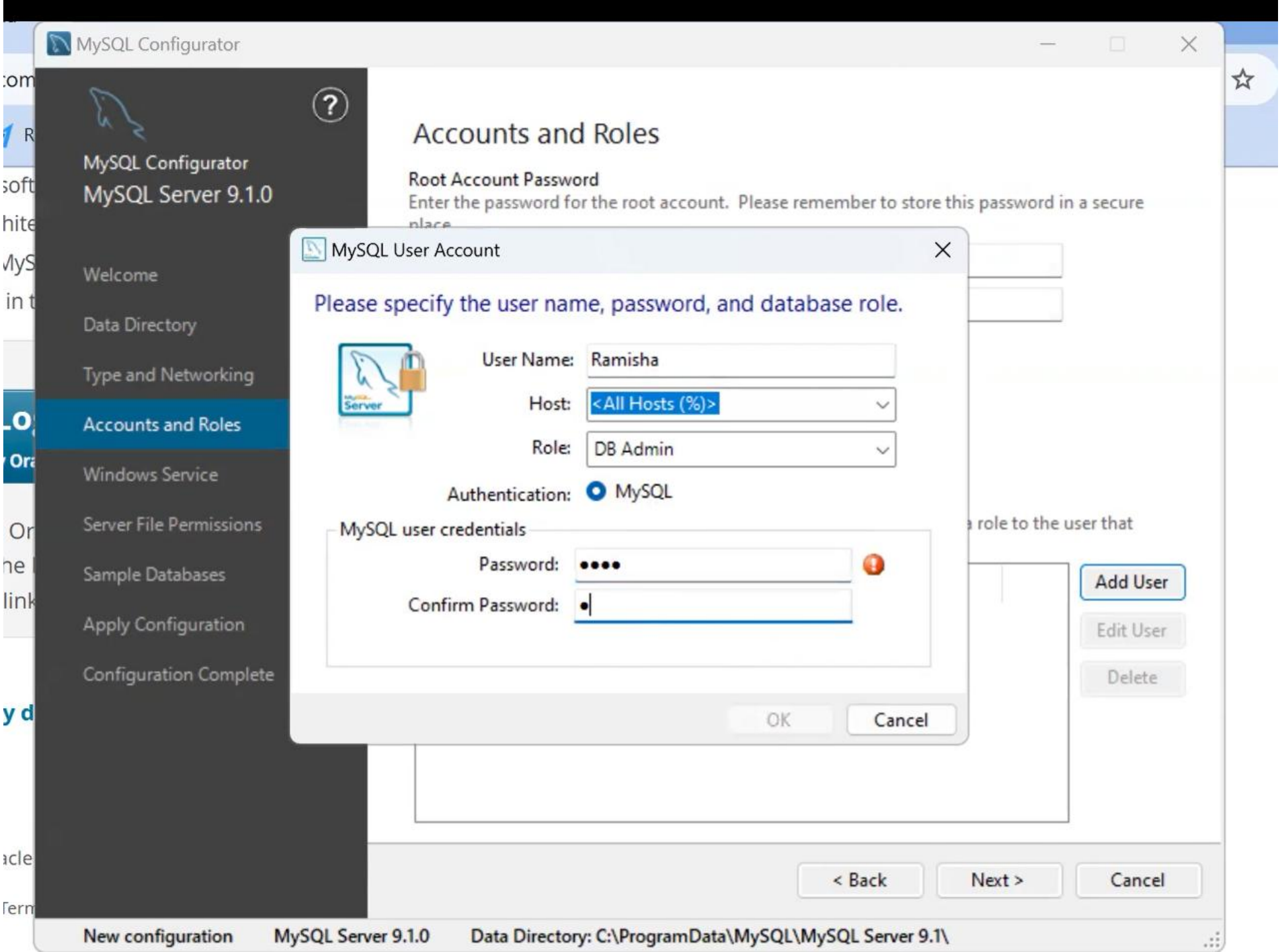
Create MySQL user accounts for your users and applications. Assign a role to the user that consists of a set of privileges.

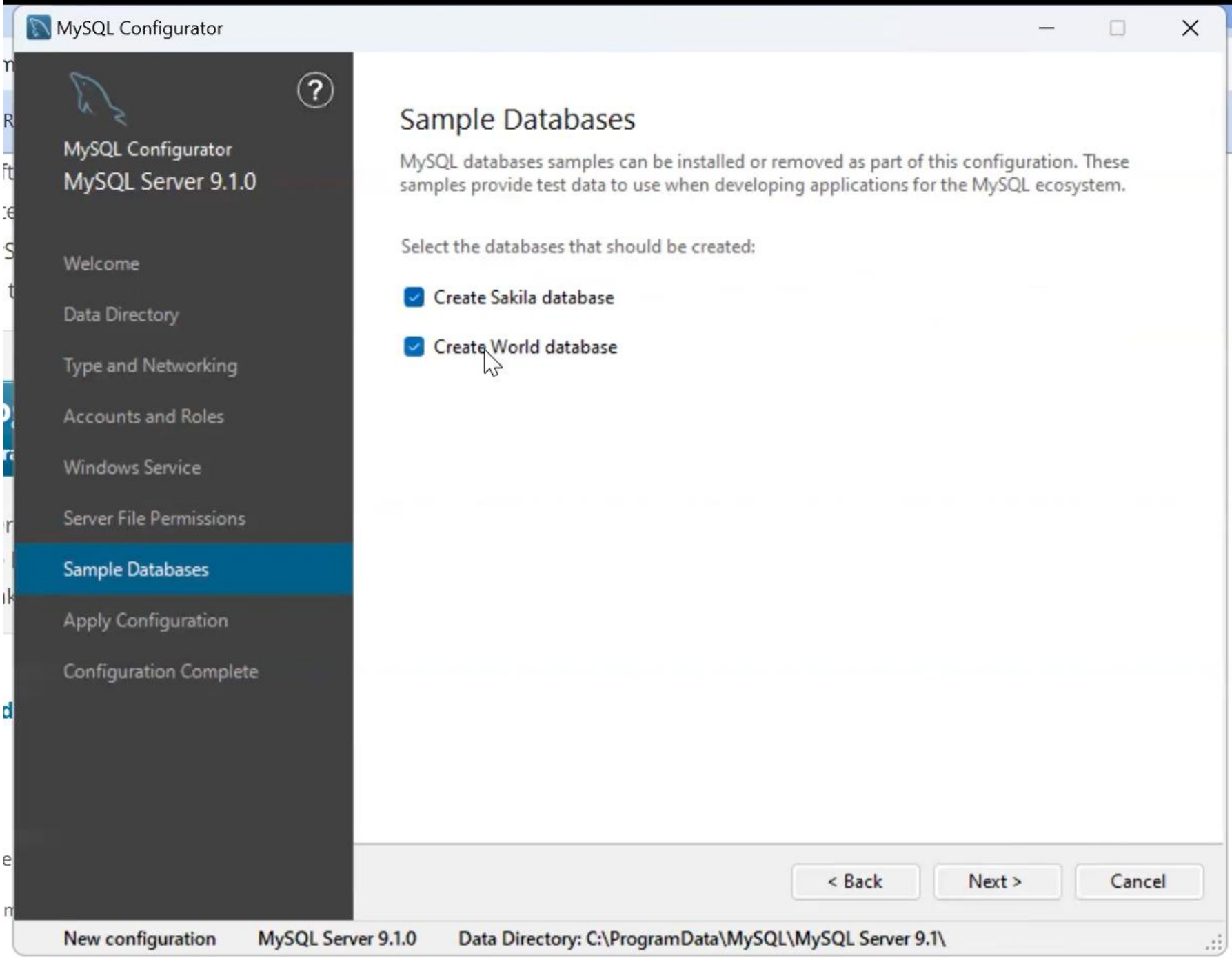
MySQL User Name	Host	User Role

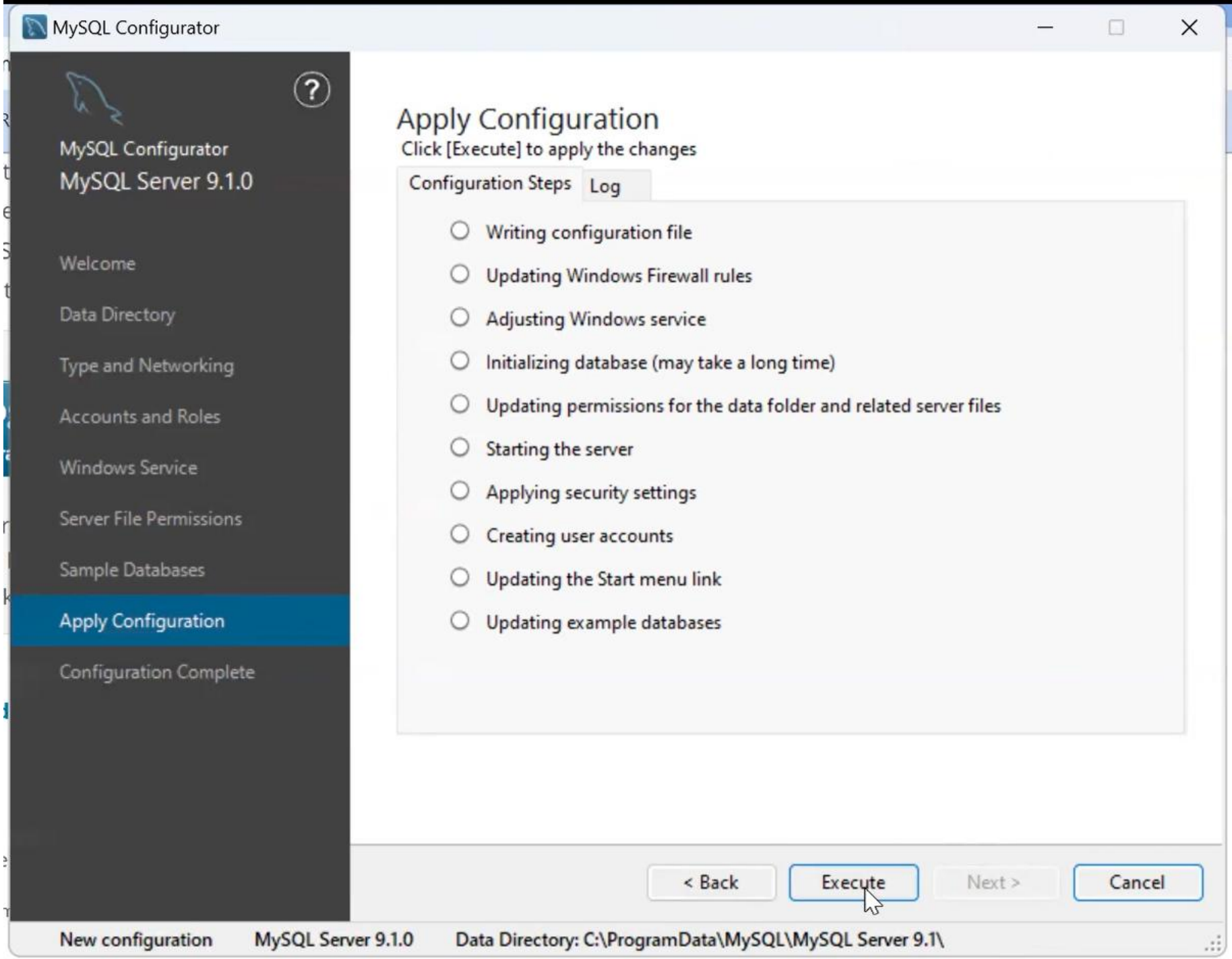
Add User

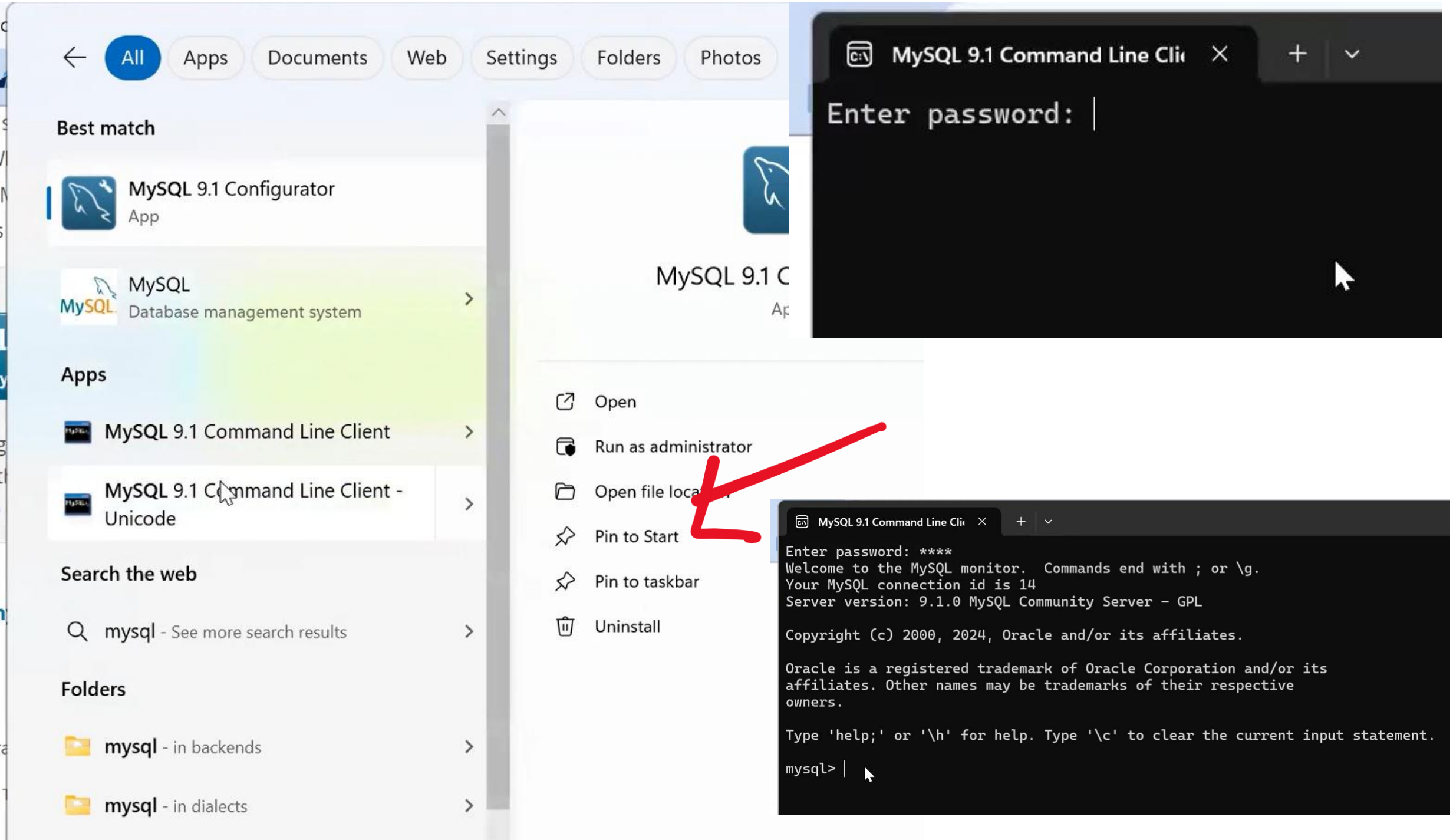
Edit User

Delete


MySQL Configurator
MySQL Configurator
MySQL Server 9.1.0
Welcome
Data Directory
Type and Networking
Accounts and Roles
Windows Service
Server File Permissions
Sample Databases
Apply Configuration
Configuration Complete
Accounts and Roles
Root Account Password
Enter the password for the root account. Please remember to store this password in a secure
MySQL User Account
Please specify the user name, password, and database role.
User Name: Ramisha
Host: <All Hosts (%)>
Role: DB Admin
Authentication: MySQL
MySQL user credentials
Password: ••••
Confirm Password:
OK
Cancel
role to the user that
Add User
Edit User
Delete
< Back
Next >
Cancel
New configuration
MySQL Server 9.1.0
Data Directory: C:\ProgramData\MySQL\MySQL Server 9.1\


MySQL Configurator
MySQL Configurator
MySQL Server 9.1.0
Welcome
Data Directory
Type and Networking
Accounts and Roles
Windows Service
Server File Permissions
Sample Databases
Apply Configuration
Configuration Complete
Sample Databases
MySQL databases samples can be installed or removed as part of this configuration. These samples provide test data to use when developing applications for the MySQL ecosystem.
Select the databases that should be created:
Create Sakila database
Create World database
< Back
Next >
Cancel
New configuration
MySQL Server 9.1.0
Data Directory: C:\ProgramData\MySQL\MySQL Server 9.1\


MySQL Configurator
MySQL Configurator
MySQL Server 9.1.0
Welcome
Data Directory
Type and Networking
Accounts and Roles
Windows Service
Server File Permissions
Sample Databases
Apply Configuration
Configuration Complete
Apply Configuration
Click [Execute] to apply the changes
Configuration Steps
Log
Writing configuration file
Updating Windows Firewall rules
Adjusting Windows service
Initializing database (may take a long time)
Updating permissions for the data folder and related server files
Starting the server
Applying security settings
Creating user accounts
Updating the Start menu link
Updating example databases
< Back
Execute
Next >
Cancel
New configuration
MySQL Server 9.1.0
Data Directory: C:\ProgramData\MySQL\MySQL Server 9.1\


All
Apps
Documents
Web
Settings
Folders
Photos
Best match
MySQL 9.1 Configurator
App
MySQL
Database management system
Apps
MySQL 9.1 Command Line Client
MySQL 9.1 Command Line Client - Unicode
Search the web
mysql - See more search results
Folders
mysql - in backends
mysql - in dialects
MySQL 9.1 C
Ap
Open
Run as administrator
Open file loca
Pin to Start
Pin to taskbar
Uninstall
MySQL 9.1 Command Line Clie
Enter password:
MySQL 9.1 Command Line Clie
Enter password: ****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 9.1.0 MySQL Community Server - GPL
Copyright (c) 2000, 2024, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql>

dev.mysql.com/downloads/

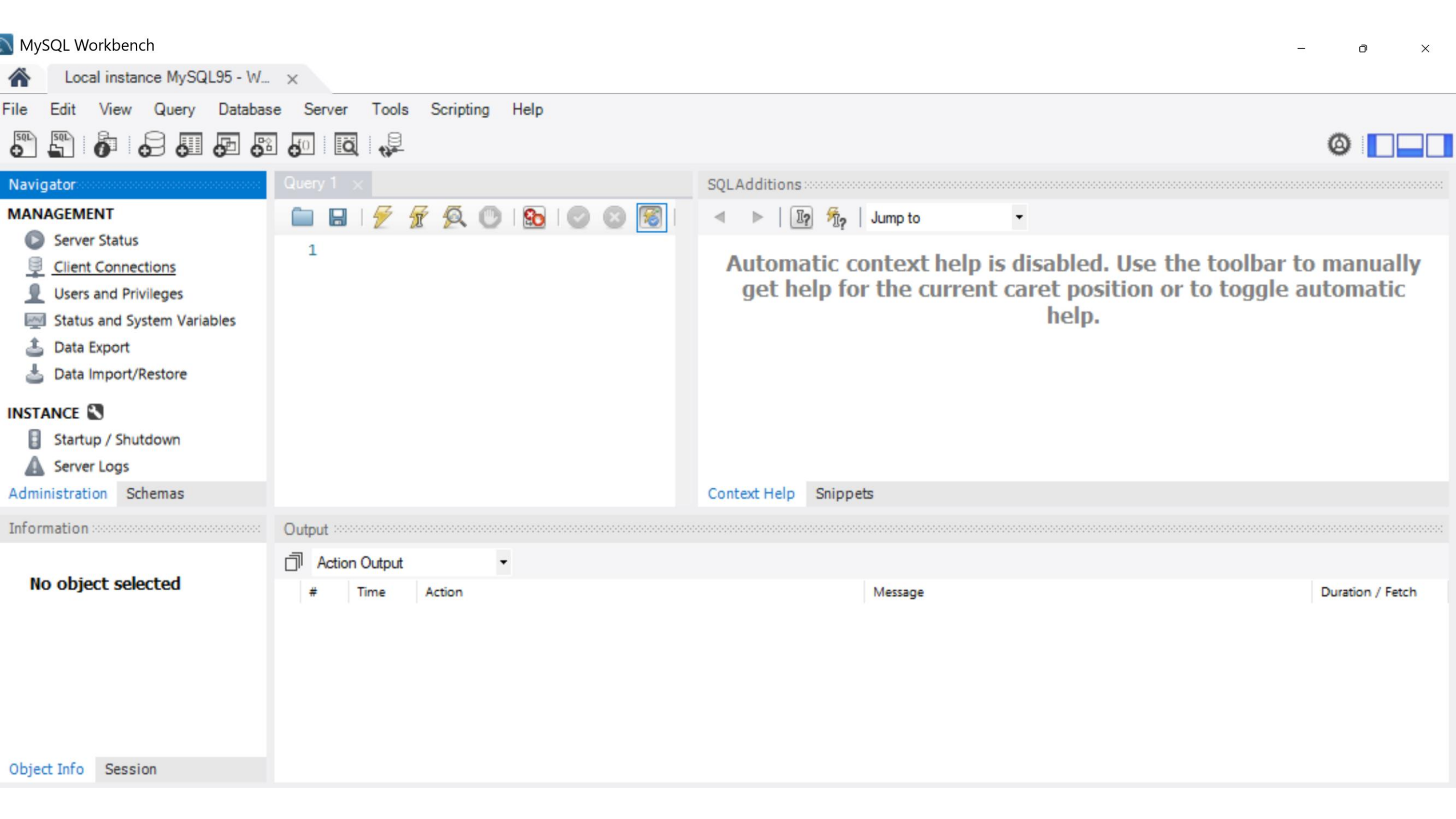
- MySQL Yum Repository
- MySQL APT Repository
- MySQL SUSE Repository

- MySQL Community Server
- MySQL NDB Cluster
- MySQL Router
- MySQL Shell
- MySQL Operator
- MySQL NDB Operator
- MySQL Workbench

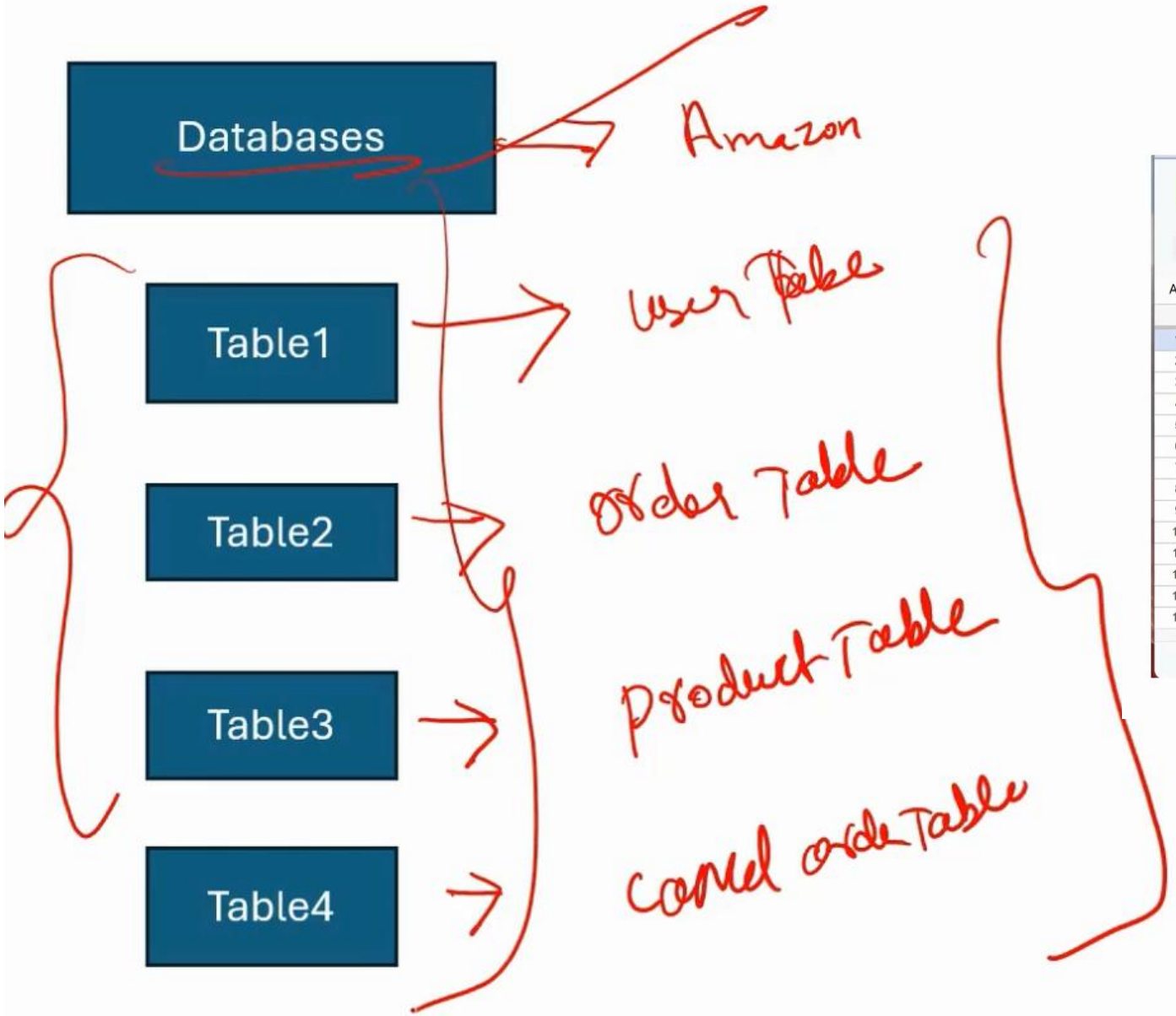
- MySQL Installer for Windows

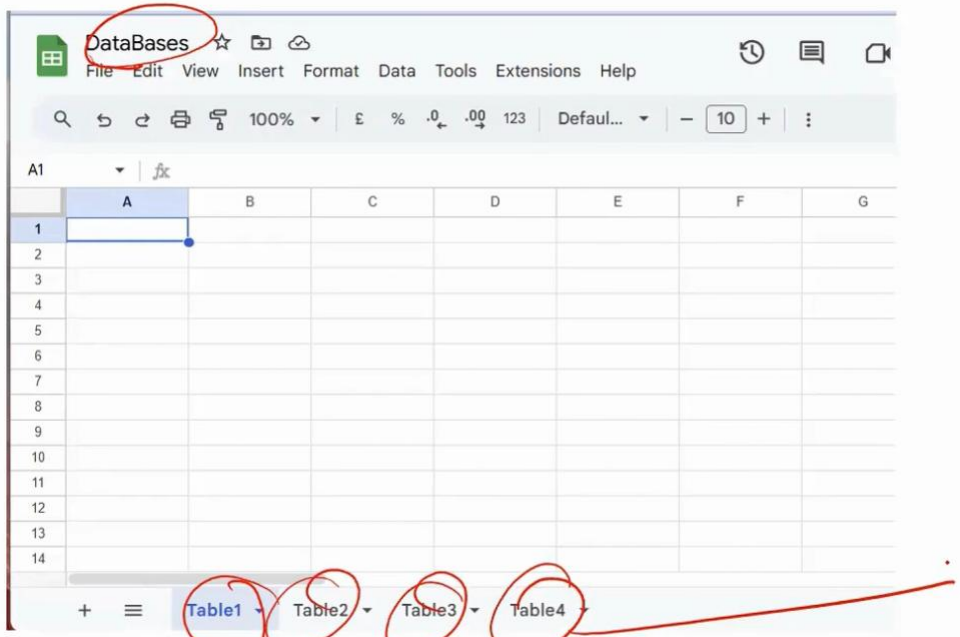
- C API (libmysqlclient)
- Connector/C++
- Connector/J
- Connector/NET
- Connector/Node.js
- Connector/ODBC
- Connector/Python
- MySQL Native Driver for PHP

- MySQL Benchmark Tool
- Time zone description tables
- Download Archives


MySQL Workbench
Local instance MySQL95 - W...
File Edit View Query Database Server Tools Scripting Help
Navigator
MANAGEMENT
Server Status
Client Connections
Users and Privileges
Status and System Variables
Data Export
Data Import/Restore
INSTANCE
Startup / Shutdown
Server Logs
Administration Schemas
Information
No object selected
Object Info Session
Query 1
1
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
Context Help Snippets
Output
Action Output
Time Action Message Duration / Fetch

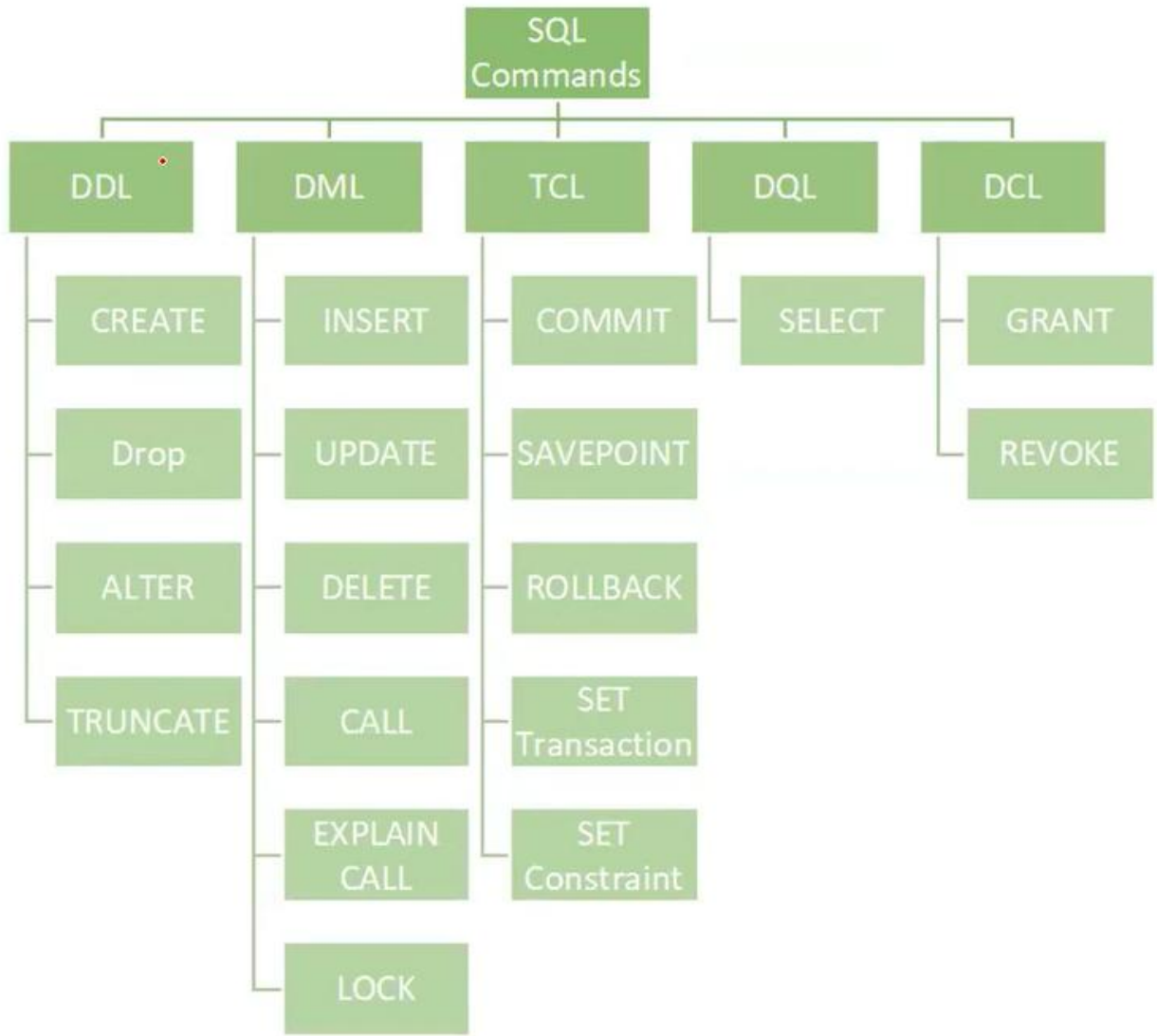
SQL- Structured Query Language






SQL Commands
Data Definition Language
Data Manipulation Language
Transaction Control Language
Data Query Language
Data Control Language

SQL- Structured Query Languag



Primary Key Vs Foreign Key

Table 1: `departments`

department_id	department_name
1	HR
2	IT
3	Marketing

Here, `department_id` is the **primary key** of the `departments` table.

Table 2: `employees`

employee_id	name	department_id
101	Alice	1
102	Bob	2
103	Charlie	3

Here:

- `employee_id` is the **primary key** for the `employees` table.
- `department_id` is a **foreign key** that references `department_id` in the `departments` table.

Table 3: `department_heads`

department_id	head_name
1	Alice
2	Bob
3	Charlie

Here:

- `department_id` is the **primary key** for `department_heads`.
- It also acts as a **foreign key** referencing the `departments` table to ensure that the `department_id` exists in `departments`.

SQL- Structured Query Language

Constraint	Use	Purpose	Duplicates	NULL Values
Primary Key	Uniquely identifies each record in a table.	Ensures that every record has a unique and non-null identifier (e.g., `CustomerID`, `OrderID`).	**Not Allowed**	**Not Allowed**
Foreign Key	Links one table to another by referencing the primary key in another table.	Enforces referential integrity between tables (e.g., `CustomerID` in `Orders` references `Customers`).	**Allowed (in referencing table)**	**Allowed** unless constrained by the referenced primary key.
Unique Key	Ensures that the values in a column (or group of columns) are unique.	Enforces the uniqueness of data without being the primary identifier (e.g., `Email`, `Username`).	**Not Allowed**	**Allowed** (1 NULL value per unique column, depending on the database system).

```sql
-- ----------------------------------
-- Data Definition Language (DDL)
-- Create database
CREATE DATABASE database1;
USE database1;
SHOW DATABASES;

-- Create table
CREATE TABLE table_1(
student_id INT PRIMARY KEY AUTO_INCREMENT,
student_name VARCHAR(20) NOT NULL,
email_id VARCHAR(50) UNIQUE,
join_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

SHOW TABLES;

-- Rename
ALTER TABLE table_1 RENAME TO table1;
ALTER TABLE table1 RENAME COLUMN join_date TO joined_date;

-- Add
ALTER TABLE table1 ADD phone_number VARCHAR(10);

-- Modify
ALTER TABLE table1 MODIFY phone_number INT(10);

-- Create with forign key
CREATE TABLE table2 (
subject_id INT AUTO_INCREMENT PRIMARY KEY,
age INT CHECK(age BETWEEN 1 AND 100),
student_id  INT,
FOREIGN KEY (student_id) REFERENCES table1(student_id )
);

-- Drop
DROP DATABASE database1;
DROP TABLE table1;
ALTER TABLE table1 DROP COLUMN age;

-- Delete all row
TRUNCATE TABLE table1;
```

Table 1: Customers

CustomerID	CustomerName	Country
1	Alice	USA
2	Bob	Canada
3	Charlie	UK
4	Diana	Australia

Table 2: Orders

OrderID	CustomerID	OrderDate	Amount
101	1	2024-11-01	150
102	3	2024-11-10	200
103	1	2024-11-15	300
104	5	2024-11-18	250

Left Join Query

```sql
SELECT
    Customers.CustomerID,
    Customers.CustomerName,
    Orders.OrderID,
    Orders.OrderDate,
    Orders.Amount
FROM
    Customers
LEFT JOIN
    Orders
ON
    Customers.CustomerID = Orders.CustomerID;
```

Resulting Output Table

The **LEFT JOIN** includes all rows from the **Customers** table and the matching rows from the **Orde** table. If there is no match, the columns from **Orders** will show `NULL`.

CustomerID	CustomerName	OrderID	OrderDate	Amount
1	Alice	101	2024-11-01	150
1	Alice	103	2024-11-15	300
2	Bob	NULL	NULL	NULL
3	Charlie	102	2024-11-10	200
4	Diana	NULL	NULL	NULL

Customers Table

c_id	name
1	John
2	Mary
3	Alex

Orders Table

o_id	c_id
101	1
102	3
103	4

✅ 1️⃣ INNER JOIN

```sql
SELECT name, o_id
FROM customers
INNER JOIN orders USING(c_id);
```

👉 Only matching rows

name	o_id
John	101
Alex	102

✅ 2️⃣ LEFT JOIN

```sql
SELECT name, o_id
FROM customers
LEFT JOIN orders USING(c_id);
```

👉 All customers + matching orders

name	o_id
John	101
Mary	NULL
Alex	102

✅ 3️⃣ RIGHT JOIN

```sql
SELECT name, o_id
FROM customers
RIGHT JOIN orders USING(c_id);
```

👉 All orders + matching customers

name	o_id
John	101
Alex	102
NULL	103

✅ 4️⃣ FULL JOIN (Not supported in MySQL directly)

Equivalent using UNION:

```sql
SELECT name, o_id
FROM customers
LEFT JOIN orders USING(c_id)
UNION
SELECT name, o_id
FROM customers
RIGHT JOIN orders USING(c_id);
```

👉 Everything from both tables

name	o_id
John	101
Mary	NULL
Alex	102
NULL	103

✅ 6️⃣ NATURAL JOIN

```sql
SELECT name, o_id
FROM customers NATURAL JOIN orders;
```

👉 Auto join on matching column names (here: c_id)

name	o_id
John	101
Alex	102

⚠️ Same as INNER JOIN but **dangerous** if more columns match by accident!

users

user_id	name	email	registered_date	membership
1	Alice Johnson	alice.j@example.com	2024-01-15	Prime
2	Bob Smith	bob.s@example.com	2024-02-01	Basic
3	Charlie Brown	charlie.b@example.com	2024-03-10	Prime
4	Daisy Ridley	daisy.r@example.com	2024-04-12	Basic

Orders

order_id	user_id	order_date	total_amount
1	1	2024-05-01	79.98
2	2	2024-05-03	129.99
3	1	2024-05-04	49.99
4	3	2024-05-05	24.99

Products

product_id	name	price	category	stock
1	Echo Dot	49.99	Electronics	120
2	Kindle Paperwhite	129.99	Books	50
3	Fire Stick	39.99	Electronics	80
4	Yoga Mat	19.99	Fitness	200
5	Wireless Mouse	24.99	Electronics	150

OrderDetails

order_details_id	order_id	product_id	quantity
1	1	1	2
2	2	2	1
3	3	1	1
4	4	5	1

users

user_id	name	email	registered_date	membership
1	Alice Johnson	alice.j@example.com	2024-01-15	Prime
2	Bob Smith	bob.s@example.com	2024-02-01	Basic
3	Charlie Brown	charlie.b@example.com	2024-03-10	Prime
4	Daisy Ridley	daisy.r@example.com	2024-04-12	Basic

Orders

order_id	user_id	order_date	total_amount
1	1	2024-05-01	79.98
2	2	2024-05-03	129.99
3	1	2024-05-04	49.99
4	3	2024-05-05	24.99

Products

product_id	name	price	category	stock
1	Echo Dot	49.99	Electronics	120
2	Kindle Paperwhite	129.99	Books	50
3	Fire Stick	39.99	Electronics	80
4	Yoga Mat	19.99	Fitness	200
5	Wireless Mouse	24.99	Electronics	150

OrderDetails

order_details_id	order_id	product_id	quantity
1	1	1	2
2	2	2	1
3	3	1	1
4	4	5	1

users

user_id	name	email	registered_date	membership
1	Alice Johnson	alice.j@example.com	2024-01-15	Prime
2	Bob Smith	bob.s@example.com	2024-02-01	Basic
3	Charlie Brown	charlie.b@example.com	2024-03-10	Prime
4	Daisy Ridley	daisy.r@example.com	2024-04-12	Basic

Orders

order_id	user_id	order_date	total_amount
1	1	2024-05-01	79.98
2	2	2024-05-03	129.99
3	1	2024-05-04	49.99
4	3	2024-05-05	24.99

Products

product_id	name	price	category	stock
1	Echo Dot	49.99	Electronics	120
2	Kindle Paperwhite	129.99	Books	50
3	Fire Stick	39.99	Electronics	80
4	Yoga Mat	19.99	Fitness	200
5	Wireless Mouse	24.99	Electronics	150

OrderDetails

order_details_id	order_id	product_id	quantity
1	1	1	2
2	2	2	1
3	3	1	1
4	4	5	1

users

user_id	name	email	registered_date	membership
1	Alice Johnson	alice.j@example.com	2024-01-15	Prime
2	Bob Smith	bob.s@example.com	2024-02-01	Basic
3	Charlie Brown	charlie.b@example.com	2024-03-10	Prime
4	Daisy Ridley	daisy.r@example.com	2024-04-12	Basic

Orders

order_id	user_id	order_date	total_amount
1	1	2024-05-01	79.98
2	2	2024-05-03	129.99
3	1	2024-05-04	49.99
4	3	2024-05-05	24.99

Products

product_id	name	price	category	stock
1	Echo Dot	49.99	Electronics	120
2	Kindle Paperwhite	129.99	Books	50
3	Fire Stick	39.99	Electronics	80
4	Yoga Mat	19.99	Fitness	200
5	Wireless Mouse	24.99	Electronics	150

OrderDetails

order_details_id	order_id	product_id	quantity
1	1	1	2
2	2	2	1
3	3	1	1
4	4	5	1

users

user_id	name	email	registered_date	membership
1	Alice Johnson	alice.j@example.com	2024-01-15	Prime
2	Bob Smith	bob.s@example.com	2024-02-01	Basic
3	Charlie Brown	charlie.b@example.com	2024-03-10	Prime
4	Daisy Ridley	daisy.r@example.com	2024-04-12	Basic

Orders

order_id	user_id	order_date	total_amount
1	1	2024-05-01	79.98
2	2	2024-05-03	129.99
3	1	2024-05-04	49.99
4	3	2024-05-05	24.99

Products

product_id	name	price	category	stock
1	Echo Dot	49.99	Electronics	120
2	Kindle Paperwhite	129.99	Books	50
3	Fire Stick	39.99	Electronics	80
4	Yoga Mat	19.99	Fitness	200
5	Wireless Mouse	24.99	Electronics	150

OrderDetails

order_details_id	order_id	product_id	quantity
1	1	1	2
2	2	2	1
3	3	1	1
4	4	5	1

users

user_id	name	email	registered_date	membership
1	Alice Johnson	alice.j@example.com	2024-01-15	Prime
2	Bob Smith	bob.s@example.com	2024-02-01	Basic
3	Charlie Brown	charlie.b@example.com	2024-03-10	Prime
4	Daisy Ridley	daisy.r@example.com	2024-04-12	Basic

Orders

order_id	user_id	order_date	total_amount
1	1	2024-05-01	79.98
2	2	2024-05-03	129.99
3	1	2024-05-04	49.99
4	3	2024-05-05	24.99

Products

product_id	name	price	category	stock
1	Echo Dot	49.99	Electronics	120
2	Kindle Paperwhite	129.99	Books	50
3	Fire Stick	39.99	Electronics	80
4	Yoga Mat	19.99	Fitness	200
5	Wireless Mouse	24.99	Electronics	150

OrderDetails

order_details_id	order_id	product_id	quantity
1	1	1	2
2	2	2	1
3	3	1	1
4	4	5	1

users

user_id	name	email	registered_date	membership
1	Alice Johnson	alice.j@example.com	2024-01-15	Prime
2	Bob Smith	bob.s@example.com	2024-02-01	Basic
3	Charlie Brown	charlie.b@example.com	2024-03-10	Prime
4	Daisy Ridley	daisy.r@example.com	2024-04-12	Basic

Orders

order_id	user_id	order_date	total_amount
1	1	2024-05-01	79.98
2	2	2024-05-03	129.99
3	1	2024-05-04	49.99
4	3	2024-05-05	24.99

Products

product_id	name	price	category	stock
1	Echo Dot	49.99	Electronics	120
2	Kindle Paperwhite	129.99	Books	50
3	Fire Stick	39.99	Electronics	80
4	Yoga Mat	19.99	Fitness	200
5	Wireless Mouse	24.99	Electronics	150

OrderDetails

order_details_id	order_id	product_id	quantity
1	1	1	2
2	2	2	1
3	3	1	1
4	4	5	1