

SQL in Data Analysis

(Introduction to Database)

Data & Code

Vinh Dinh Nguyen
PhD in Computer Science

Outline



➤ **Introduction to Database**

➤ **How to install MySQL**

➤ **How to create Database**

➤ **How to use SQL for data analysis**

➤ **Kahoot Quiz**

➤ **Summary**



Introduction to Database

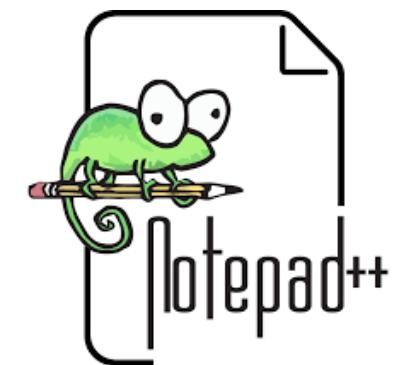
Database Motivation

Data for Online Shopping System

Customer ID	Name	Address	Product Name	Quantity	Date
1	Nguyen Van A	Tay Ninh	Monitor	5	01-01-2024
1	Nguyen Van A	Tay Ninh	Mouse	6	01-01-2024
2	Nguyen Van B	Can Tho	Monitor	4	01-01-2024
2	Nguyen Van B	Can Tho	Keyboard	5	02-01-2024
3	Nguyen Thi C	TP. HCM	Keyboard	5	02-01-2024
3	Nguyen Thi C	TP. HCM	Mouse	4	03-01-2024



What is the problem for a hundred thousand customers



Database Motivation

Customer

C_ID	Name	Address
1	Nguyen Van A	Tay Ninh
2	Nguyen Van B	Can Tho
3	Nguyen Thi C	TP. HCM

Product

P_ID	Name	Quantity
1	Monitor	100
2	Mouse	100
3	Keboard	100

Index

Values

Primary Key

Solution

Order

C_ID	P_ID	Quantity	Date
1	1	5	01-01-2024
1	2	6	01-01-2024
2	1	4	01-01-2024

Index/
Foreign Key

Values

Database Motivation

Customer

C_ID	Name	Address
1	Nguyen Van A	Tay Ninh
2	Nguyen Van B	Can Tho
3	Nguyen Thi C	TP. HCM

Product

P_ID	Name	Quantity
1	Monitor	100
2	Mouse	100
3	Keboard	100



Primary Key

Order

O_ID	C_ID	Date
1	1	01-01-2024
2	2	01-01-2024



Primary Key

Order Detail

OD_ID	O_ID	P_ID	Quantity
1	1	1	5
2	1	2	6
3	2	1	4



Another Solution

Definition

A collection of information/data.

Stored and organized in some format.

Easily accessed, managed and retrieved

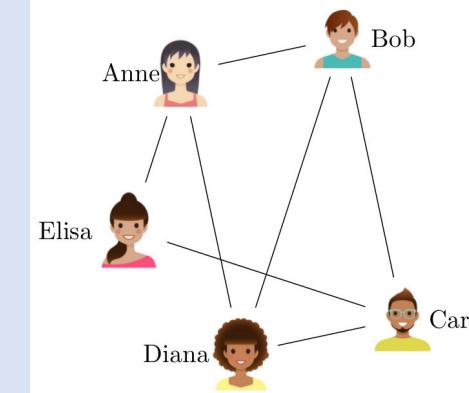
Databases



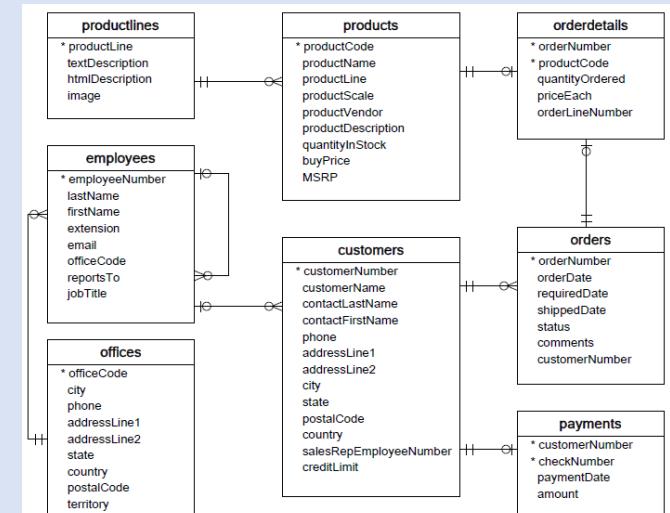
Phone book



Shopping list



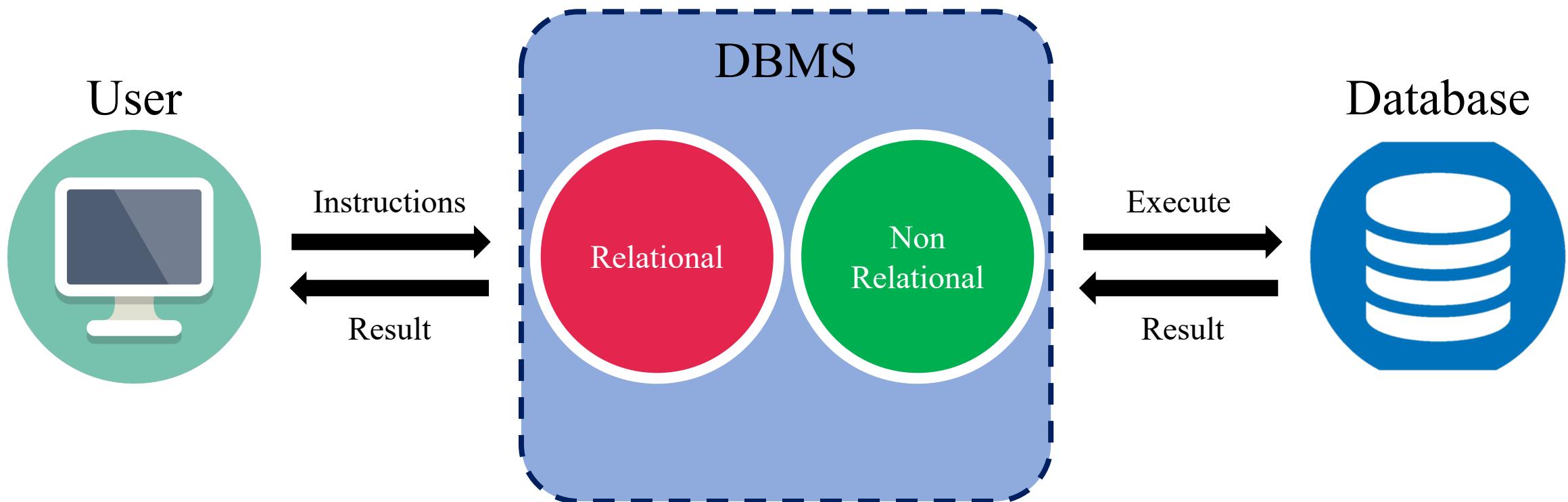
Friend Graph



Relation database

Database Management System

A software or application that allow user to interact with the database.



Relational Databases

RELATIONAL DATABASES

Customers

Products

Orders

STRUCTURED QUERY LANGUAGE (SQL)

```
SELECT *
FROM products
WHERE category = 'food'
ORDER BY price
```

C.R.U.D

Create Read Update Delete

RELATIONAL DBMS



ORACLE
DATABASE

Outline



- **Introduction to Database**
- **How to install MySQL**
- **How to create Database**
- **How to use SQL for data analysis**
- **Kahoot Quiz**
- **Summary**



Installing MySQL

Installing MySQL on MAC

The world's most popular open source database

MYSQL.COM DOWNLOADS DOCUMENTATION DEVELOPER ZONE

Step 2: Click on the DOWNLOADS tab

Step 3: Scroll to the bottom of the pages and click MySQL Community (GPL) Downloads »

MySQL Community Downloads

- MySQL Yum Repository
- MySQL APT Repository
- MySQL SUSE Repository
- MySQL Community Server
- MySQL 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

Step 1: Go to [mysql.com](https://www.mysql.com).

Step 4: Click on MySQL Community Server and Download the first link

General Availability (GA) Releases Archives

MySQL Community Server 8.0.33

Select Operating System: macOS

Select OS Version: All

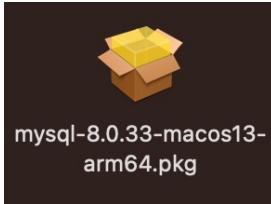
Looking for previous GA versions?

Packages for Ventura (13) are compatible with Monterey (12)

macOS 13 (ARM, 64-bit), DMG Archive	8.0.33	559.1M	Download
(mysql-8.0.33-macos13-arm64.dmg)	MD5: 065662283bf45231ea0b47e4abf1e17 Signature		
macOS 13 (x86, 64-bit), DMG Archive	8.0.33	564.5M	Download
(mysql-8.0.33-macos13-x86_64.dmg)	MD5: ddf70a939f0b64d585393e3bfd4fe36c Signature		

Installing MySQL on MAC

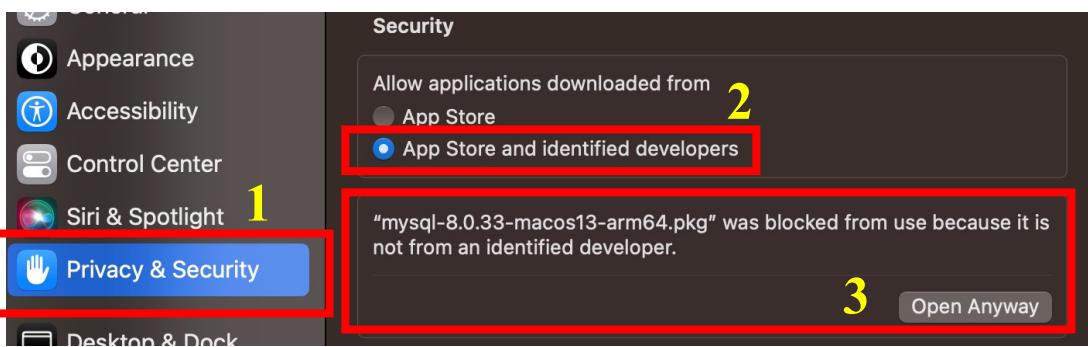
Step 5: Open the download file.



Step 5-1: If mac security prevent opening the file, go to System Settings



Step 5-2: On Privacy & Security tab, check on these boxex



Step 6: Re-open the download file and follow the steps below

To continue installing the software you must agree to the terms of the software license agreement.

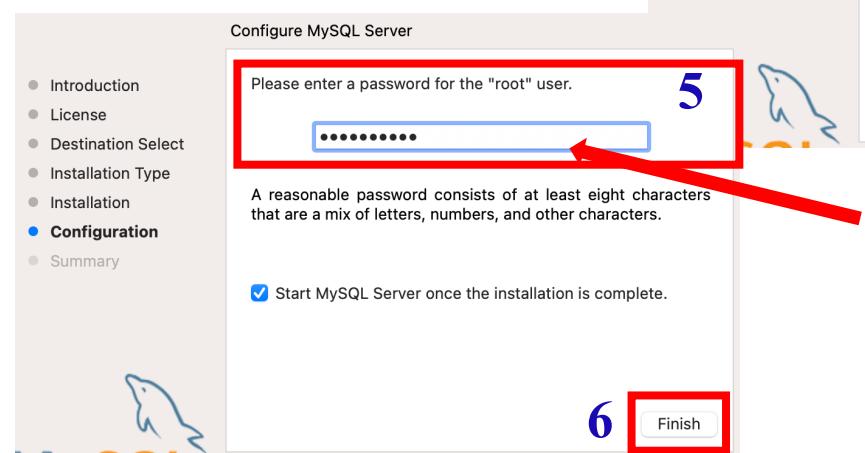
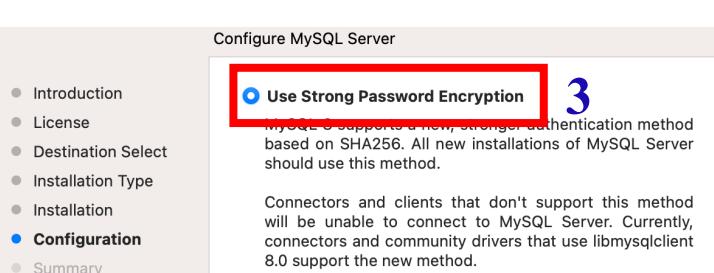
Click Agree to continue or click Disagree to cancel the installation and quit the Installer.

2

Read License

Disagree

Agree



Create your password for the “root” user

Installing MySQL on MAC



Step 7: Go back to the Download Page

**Step 8: Scroll to the bottom of the pages
and click [MySQL Community \(GPL\) Downloads »](#)**

MySQL Community Downloads

- [MySQL Yum Repository](#)
 - [MySQL APT Repository](#)
 - [MySQL SUSE Repository](#)
 - [MySQL Community Server](#)
 - [MySQL 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](#)

**Step 9: Click on MySQL Workbench
and Download the first link**

The screenshot shows the MySQL Workbench download page for macOS. It includes dropdown menus for 'Select Operating System' (set to 'macOS') and 'Select OS Version' (set to 'All'). A note states 'Packages for Ventura (13) are compatible with Monterey (12)'. Two download links are shown: 'macOS (ARM, 64-bit), DMG Archive' and 'macOS (x86, 64-bit), DMG Archive'. The first link is highlighted with a red box.

Version	File Type	Size	Download Link
8.0.33	macOS (ARM, 64-bit), DMG Archive	122.4M	Download
8.0.33	macOS (x86, 64-bit), DMG Archive	124.8M	Download



**Step 10: Open the
downloaded file and
drag to Applications.
Then open the app.**

Installing MySQL on Windows



The world's most popular open source database



Step 1: Go to [mysql.com.](#)

MYSQL.COM

DOWNLOADS

DOCUMENTATION

DEVELOPER ZONE

Step 2: Click on the DOWNLOADS tab

**Step 3: Scroll to the bottom of the pages
and click [MySQL Community \(GPL\) Downloads »](#)**

MySQL Community Downloads

- [MySQL Yum Repository](#)
 - [MySQL APT Repository](#)
 - [MySQL SUSE Repository](#)
 - [**MySQL Community Server**](#)
 - [MySQL 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](#)

**Step 4: Click on MySQL Community Server and
Go to Download Page from the recommended
download section**

Recommended Download:

MySQL Installer for Windows
 All MySQL Products. For All Windows Platforms. In One Package.



Starting with MySQL 5.6 the MySQL Installer package replaces the standalone MSI packages.

Windows (x86, 32 & 64-bit), MySQL Installer MSI

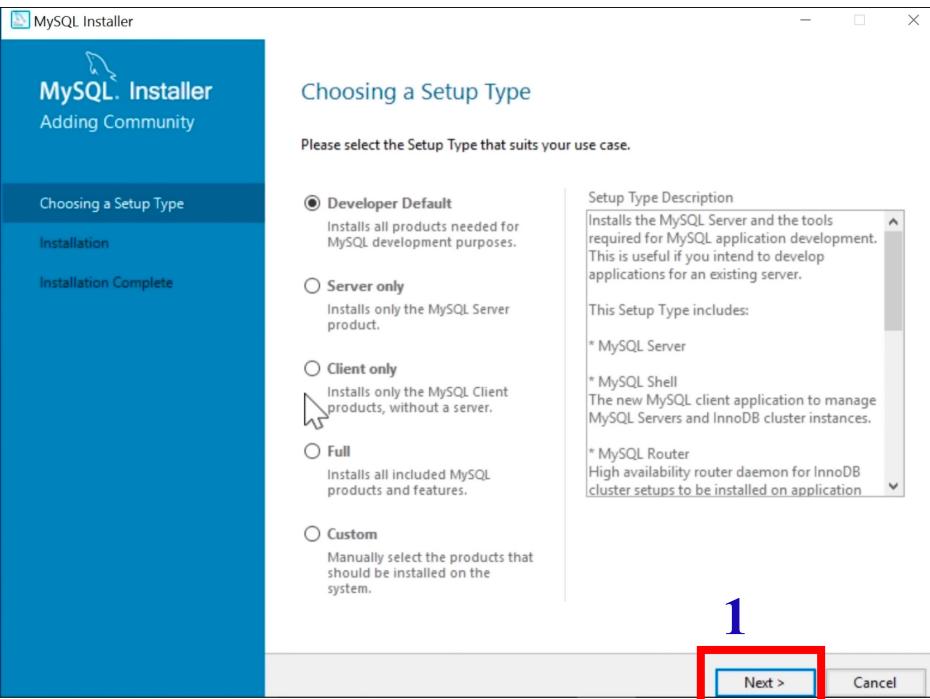
[Go to Download Page >](#)

Step 5: Download the first link

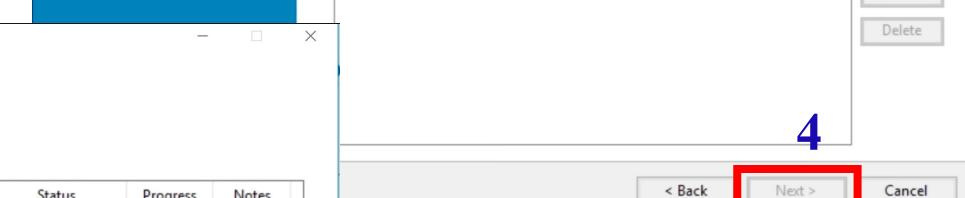
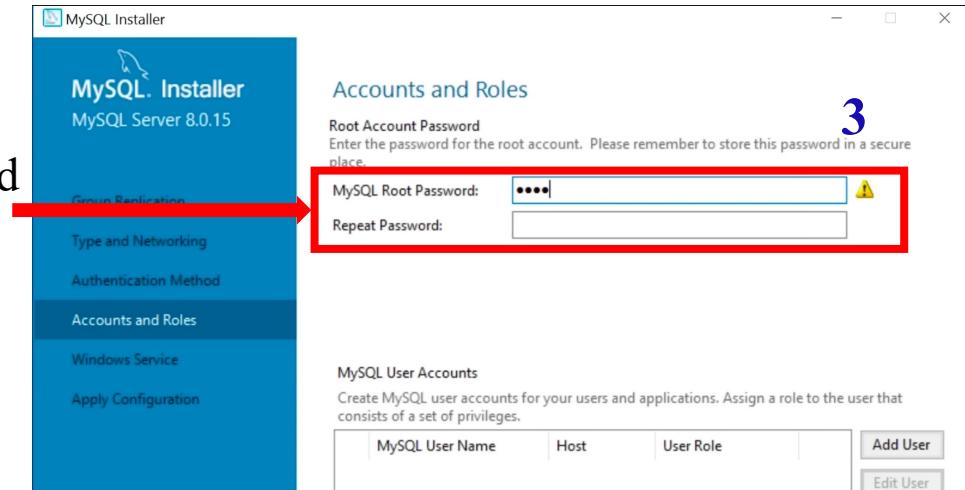
Windows (x86, 32-bit),	8.0.15	16.4M	Download
MSI Installer (mysql-installer-web-community-8.0.15.0.msi)		66363d23d2474113decf683cf84f5820	 MD5: Signature

Installing MySQL on Windows

Step 5: Open the download file and follow the steps bellow.

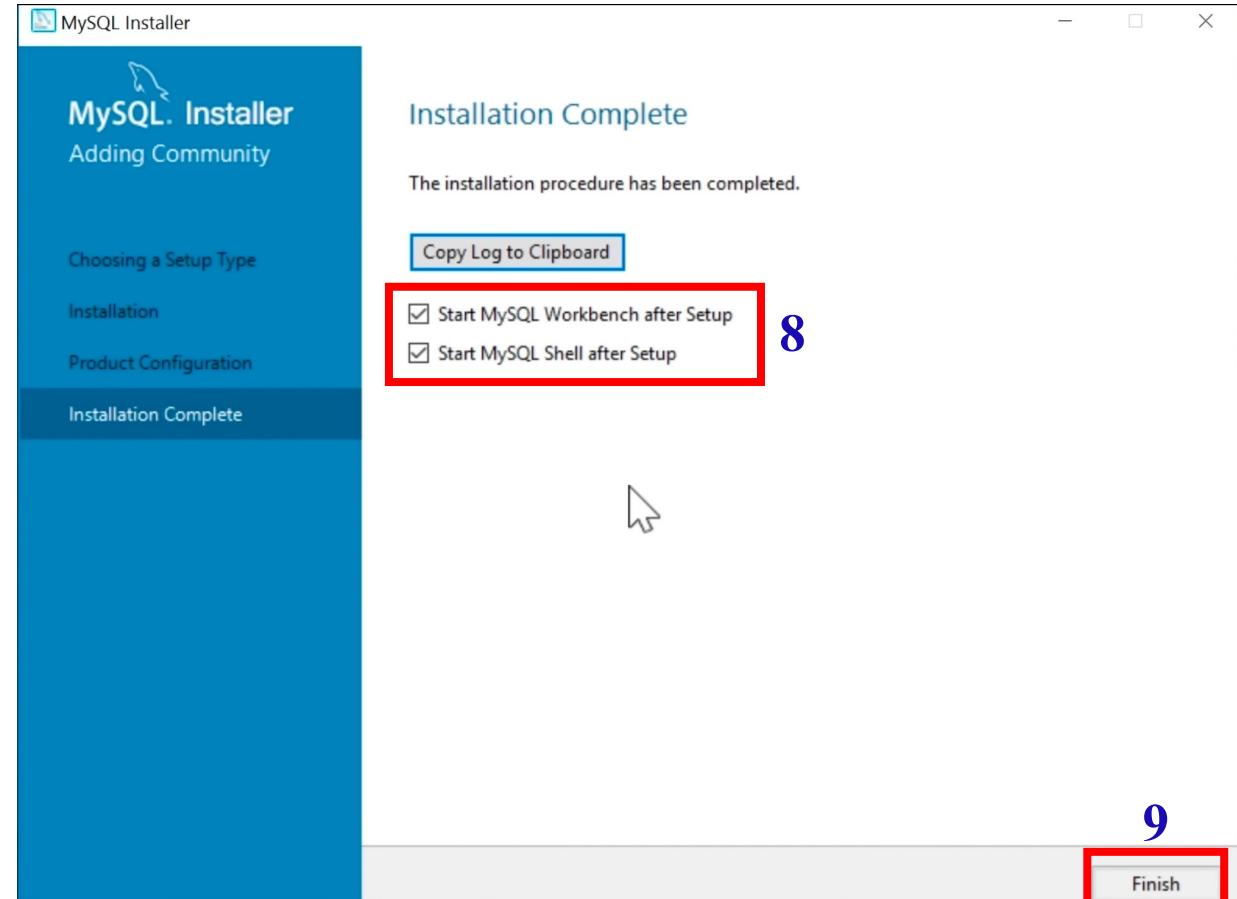
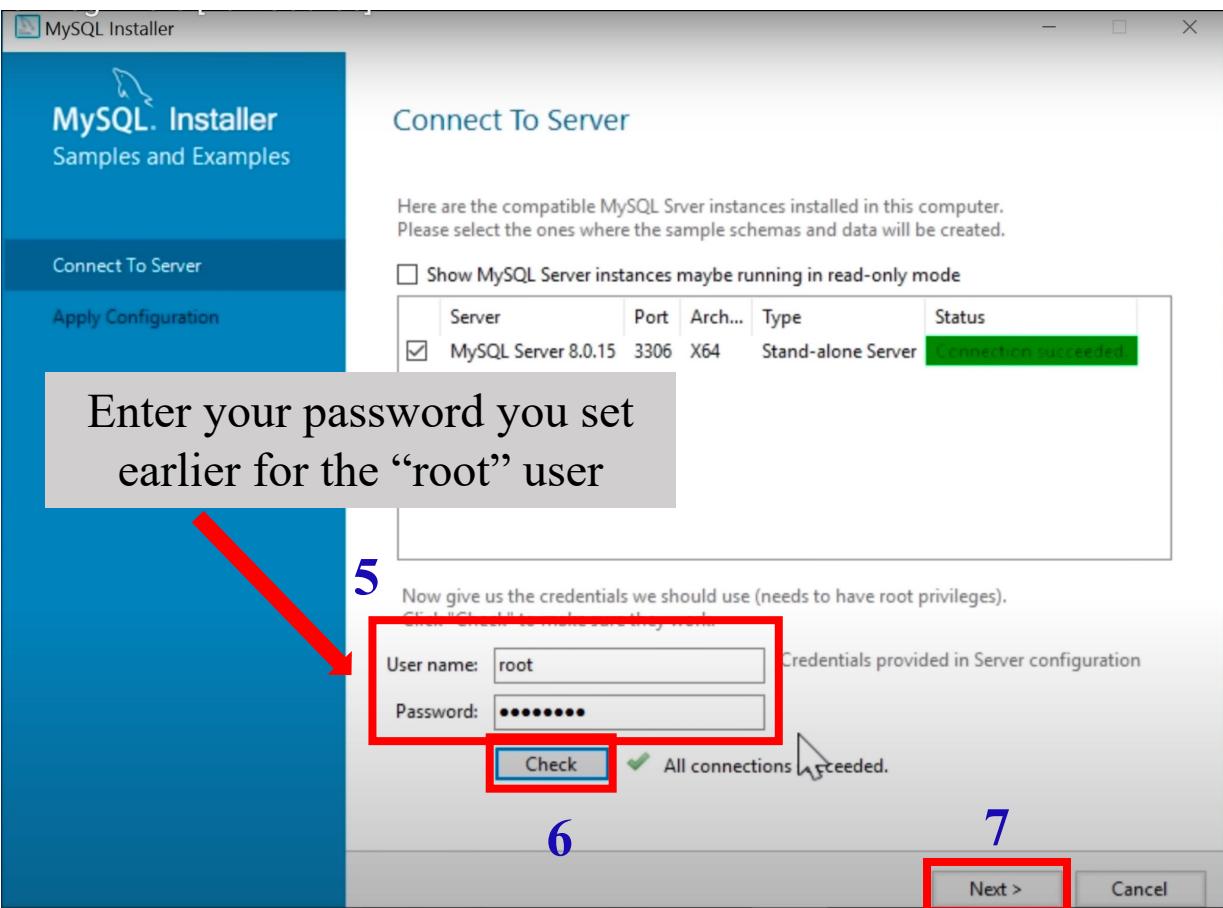


Create your password for the “root” user



Installing MySQL on Windows

Step 6: Continue clicking on
Next/Execute with default setup

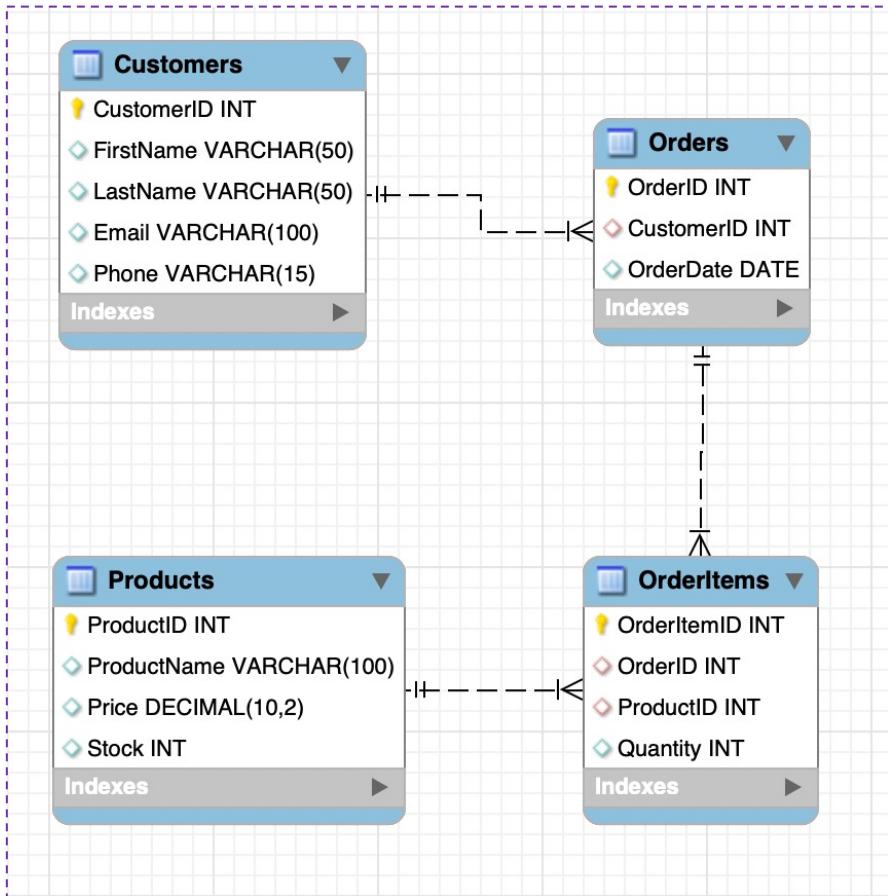


Outline



- **Introduction to Database**
- **How to install MySQL**
- **How to create Database**
- **How to use SQL for data analysis**
- **Kahoot Quiz**
- **Summary**

Create Databases



Storing Data in My SQL Database

Step 1: Design the Database Schema

Step 2: Create Table

Step 3: Insert Data

Step 4: Retrieve Data

Step 5: Update or Delete Data



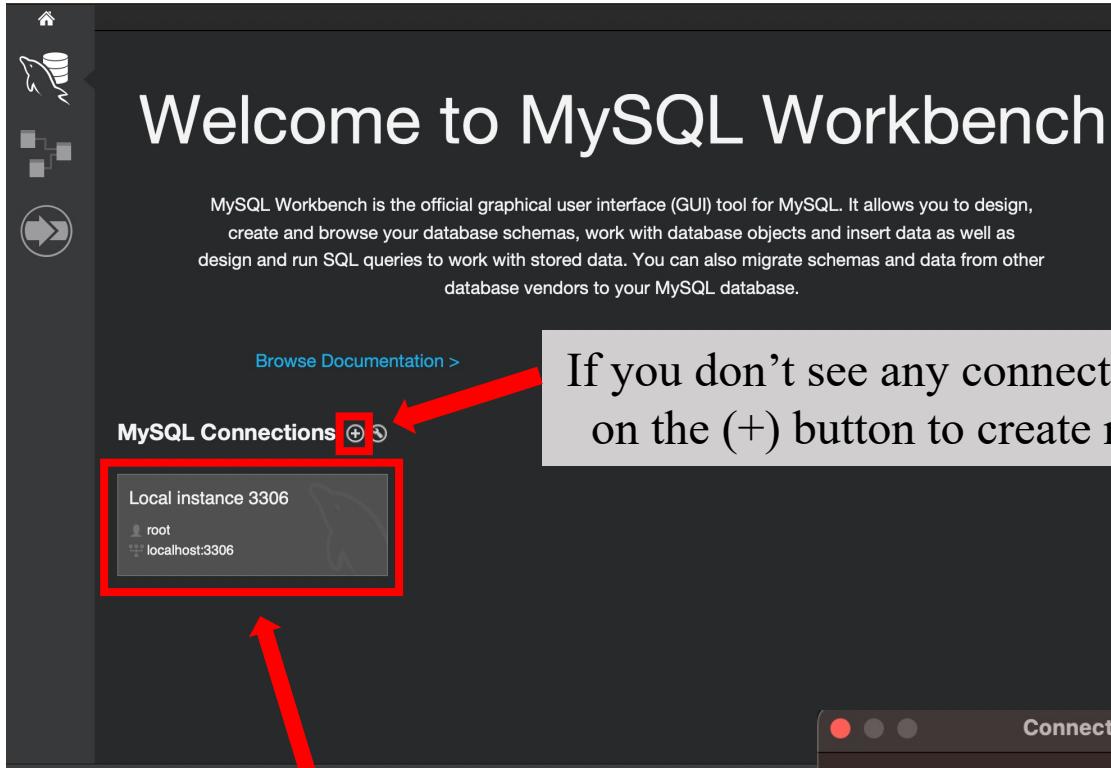
Create Databases

How to store Customer's Information to Database

CustomerID	FirstName	LastName	Email	Phone
1	John	Doe	john.doe@example.com	555-1234
2	Jane	Smith	jane.smith@example.com	555-5678
3	Alice	Johnson	alice.johnson@example.com	555-8765

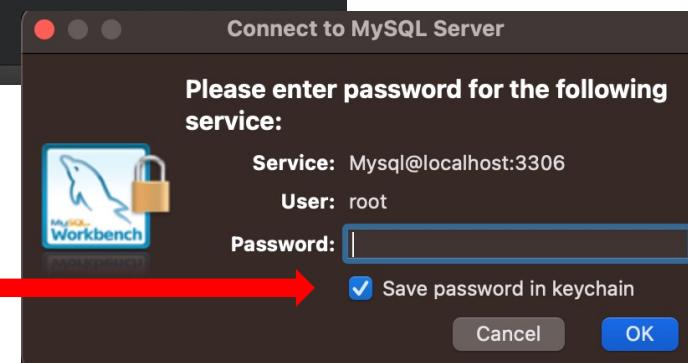
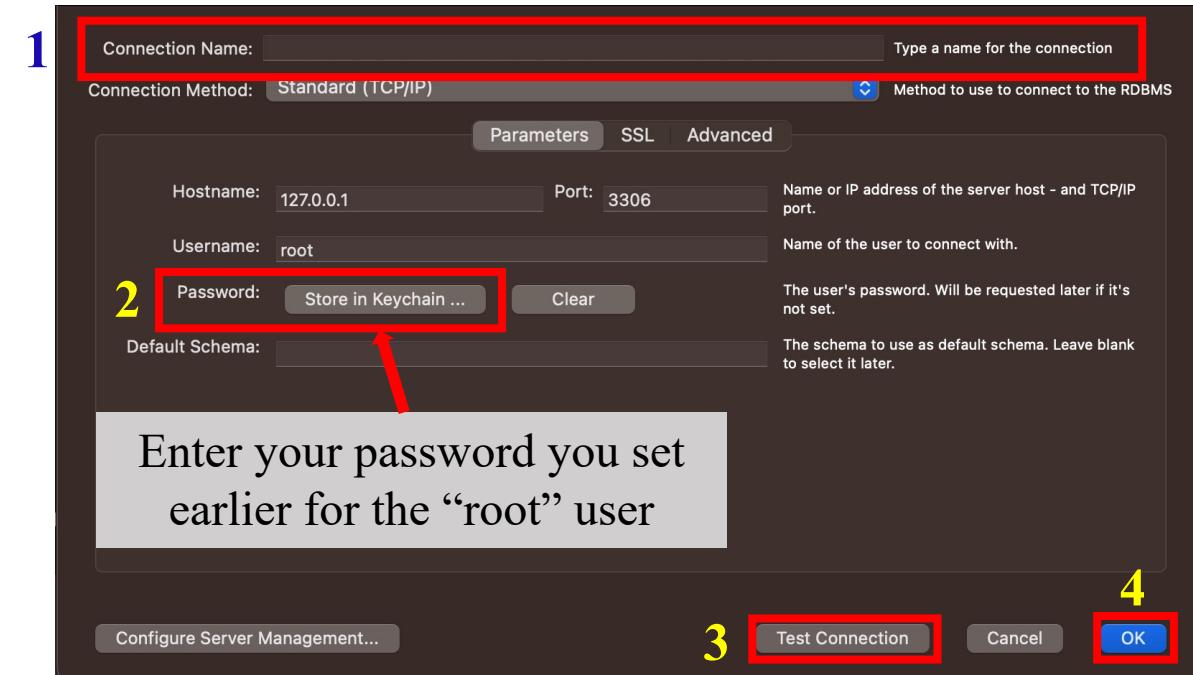
Create Databases

Open MySQL Workbench



You should see the default connection

Click and enter your password to access the server

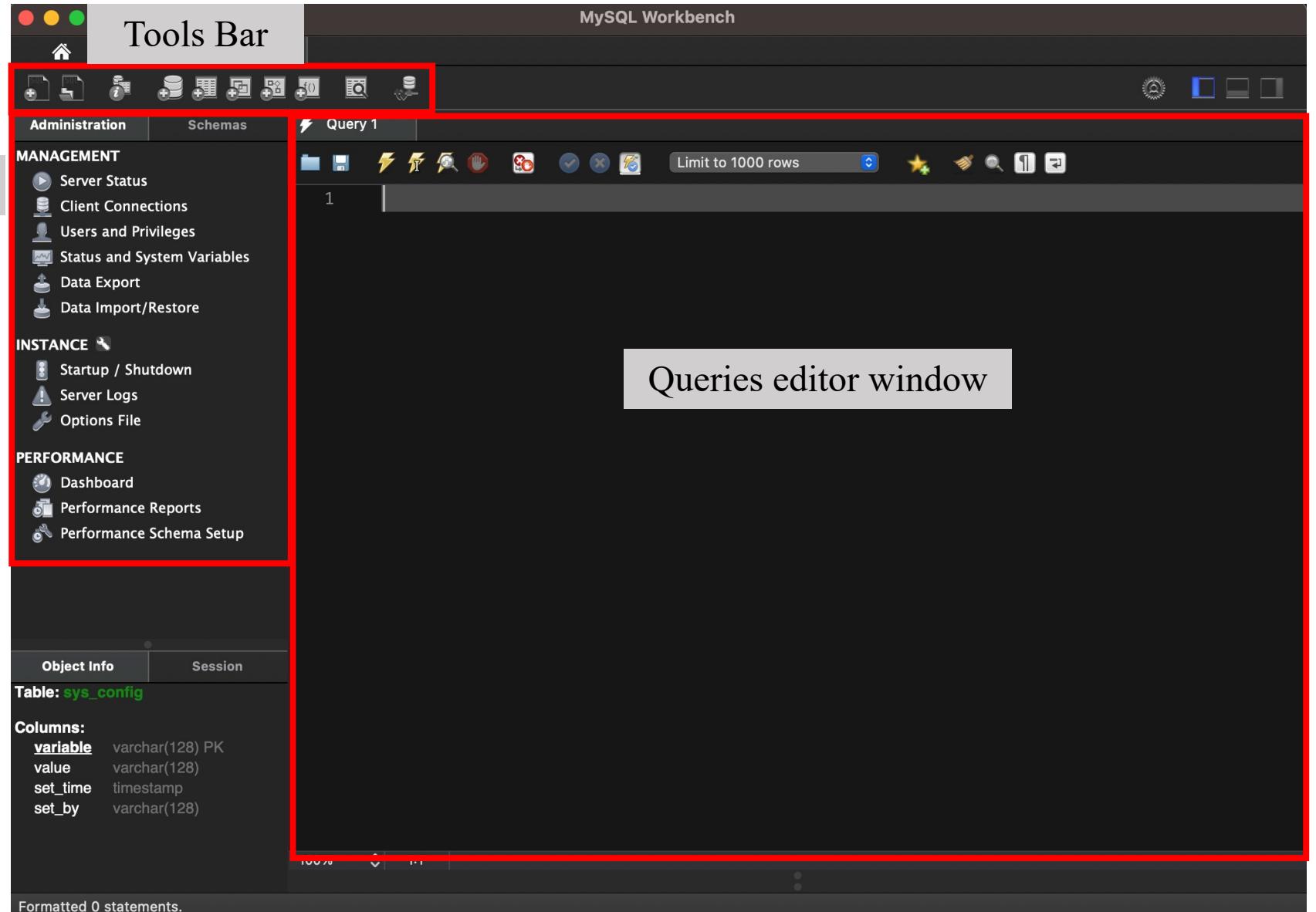


Create Databases

User Interface

Navigation panels

Queries editor window



Create a Database

1 Create Database

```
CREATE DATABASE databasename;
```

The *CREATE DATABASE* statement is used to create a new SQL database.

```
DROP DATABASE databasename;
```

The *DROP DATABASE* statement is used to drop an existing SQL database.

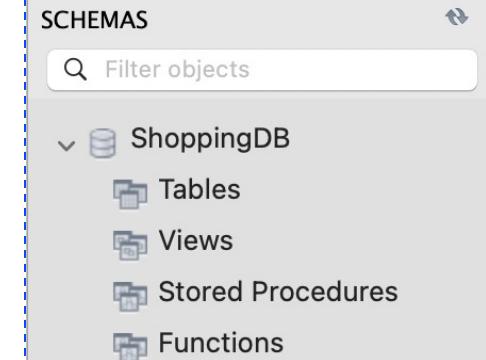
2 Create Table

```
CREATE TABLE table_name (
    column1 datatype,
    column2 datatype,
    column3 datatype,
    ...
);
```

The *CREATE TABLE* statement is used to create a new table in a database.

1

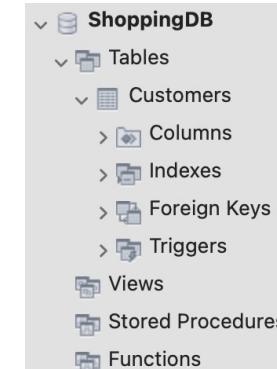
Create ShoppingDB Database



```
1 • DROP DATABASE IF EXISTS ShoppingDB;
2 • CREATE DATABASE ShoppingDB;
3 |
```

2

Create Customers Table in ShoppingDB



```
USE ShoppingDB;
CREATE TABLE Customers (
    CustomerID INT AUTO_INCREMENT PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Email VARCHAR(100),
    Phone VARCHAR(15)
);
```

Create a Database

3

Insert Data

```
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

3

Insert data into the Customers Table

```
INSERT INTO Customers (CustomerID, FirstName, LastName, Email, Phone)
VALUES
(1, 'John', 'Doe', 'john.doe@example.com', '555-1234'),
(2, 'Jane', 'Smith', 'jane.smith@example.com', '555-5678'),
(3, 'Alice', 'Johnson', 'alice.johnson@example.com', '555-8765');
```

4

Query Data

```
SELECT column1, column2, ...
FROM table_name;
```

```
SELECT * FROM table_name;
```

4

Query Data in the Customers Table

CustomerID	FirstName	LastName	Email	Phone
1	John	Doe	john.doe@example.com	555-1234
2	Jane	Smith	jane.smith@example.com	555-5678
3	Alice	Johnson	alice.johnson@example.com	555-8765

The screenshot shows a database interface with a result grid displaying customer data. Below the grid is a toolbar with various icons for editing and exporting. A blue arrow points from the bottom toolbar up to the grid, indicating the execution of the query.

```
1 • SELECT * FROM ShoppingDB.Customers;
```

Create Databases

Customer Information

CustomerID	FirstName	LastName	Email	Phone
1	John	Doe	john.doe@example.com	555-1234
2	Jane	Smith	jane.smith@example.com	555-5678
3	Alice	Johnson	alice.johnson@example.com	555-8765

Product Information



ProductID	ProductName	Price	Stock
1	Laptop	999,99	50
2	Smartphone	499,99	100
3	Tablet	299,99	200

Create a Database

1

Create Products Table in ShoppingDB

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' tree view has 'ShoppingDB' selected. Under 'Tables', 'Products' is highlighted. The main pane displays the SQL code for creating the 'Products' table:

```

1 • USE ShoppingDB;
2 • CREATE TABLE Products (
3     ProductID INT PRIMARY KEY,
4     ProductName VARCHAR(100),
5     Price DECIMAL(10, 2),
6     Stock INT
7 );

```

2

Insert data into Product Table

```

1 INSERT INTO Products (ProductID, ProductName, Price, Stock)
2 VALUES
3 (1, 'Laptop', 999.99, 50),
4 (2, 'Smartphone', 499.99, 100),
5 (3, 'Tablet', 299.99, 200);

```

3

Query Data in the Product Table

The screenshot shows the MySQL Workbench interface with a dashed box around the query editor. The query is:

```

1 • SELECT * FROM ShoppingDB.Products;

```

Result Grid

ProductID	ProductName	Price	Stock
1	Laptop	999.99	50
2	Smartphone	499.99	100
3	Tablet	299.99	200

Create a Database

Customer

CustomerID	FirstName	LastName	Email	Phone
1	John	Doe	john.doe@example.com	555-1234
2	Jane	Smith	jane.smith@example.com	555-5678
3	Alice	Johnson	alice.johnson@example.com	555-8765

Product

ProductID	ProductName	Price	Stock
1	Laptop	999,99	50
2	Smartphone	499,99	100
3	Tablet	299,99	200

Customer might buy many products. How?

OrderID	CustomerName	OrderDate	ProductName	Quantity
1	John Doe	2023-07-01	Laptop	1
1	John Doe	2023-07-01	Tablet	2
2	Jane Smith	2023-07-02	Smartphone	1

Create Databases

Customer

CustomerID	FirstName	LastName	Email	Phone
1	John	Doe	john.doe@example.com	555-1234
2	Jane	Smith	jane.smith@example.com	555-5678
3	Alice	Johnson	alice.johnson@example.com	555-8765

Order

OrderID	CustomerID	OrderDate
1	1	2023-07-01
2	2	2023-07-02

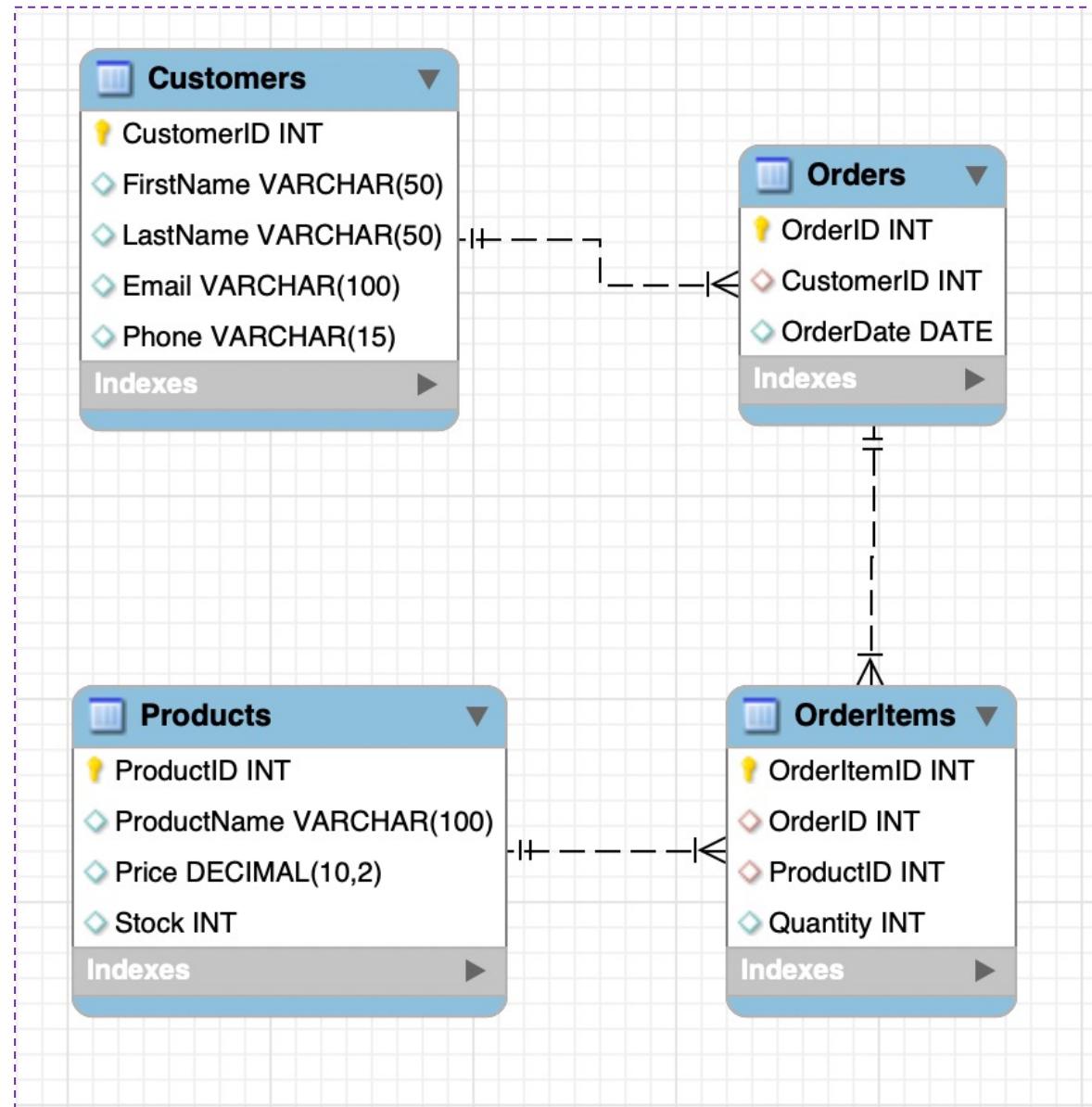
Product

ProductID	ProductName	Price	Stock
1	Laptop	999,99	50
2	Smartphone	499,99	100
3	Tablet	299,99	200

OrderItem

OrderItemID	OrderID	ProductID	Quantity
1	1	1	1
2	1	3	2
3	2	2	1

Create Databases



Create a Database

1

Create Order Table in ShoppingDB

```

1 USE ShoppingDB;
2 • - CREATE TABLE Orders (
3     OrderID INT PRIMARY KEY,
4     CustomerID INT,
5     OrderDate DATE,
6     FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
7 );

```

2

Create OrderItems Table

```

9 • USE ShoppingDB;
10 • - CREATE TABLE OrderItems (
11     OrderItemID INT PRIMARY KEY,
12     OrderID INT,
13     ProductID INT,
14     Quantity INT,
15     FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
16     FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
17 );

```

3

Insert Data

```

1 • INSERT INTO Customers (CustomerID, FirstName, LastName, Email, Phone)
2     VALUES
3         (1, 'John', 'Doe', 'john.doe@example.com', '555-1234'),
4         (2, 'Jane', 'Smith', 'jane.smith@example.com', '555-5678'),
5         (3, 'Alice', 'Johnson', 'alice.johnson@example.com', '555-8765');
6
7 • INSERT INTO Products (ProductID, ProductName, Price, Stock)
8     VALUES
9         (1, 'Laptop', 999.99, 50),
10        (2, 'Smartphone', 499.99, 100),
11        (3, 'Tablet', 299.99, 200);
12
13 • INSERT INTO Orders (OrderID, CustomerID, OrderDate)
14     VALUES
15         (1, 1, '2023-07-01'),
16         (2, 2, '2023-07-02');
17
18 • INSERT INTO OrderItems (OrderItemID, OrderID, ProductID, Quantity)
19     VALUES
20         (1, 1, 1, 1),
21         (2, 1, 3, 2),
22         (3, 2, 2, 1);

```

Entity Relationship Diagram

1

Database Server Tools Scripting

Connect to Database...
Manage Connections...
Reverse Engineer...
Schema Transfer Wizard...
Migration Wizard...
Edit Type Mappings for Generic Migration...
Search Table Data...

2

Reverse Engineer Database

Set Parameters for Connecting to a DBMS

Connection Options
Stored Connection: Local instance 3306
Select from saved connection settings
Connection Method: Standard (TCP/IP)
Method to use to connect to the RDBMS

Parameters SSL Advanced

Hostname: localhost Port: 3306 Name or IP address of the server host - and TCP/IP port.
Username: root Name of the user to connect with.
Password: The user's password. Will be requested later if it's not set.

3

Reverse Engineer Database

Connect to DBMS and Fetch Information

The following tasks will now be executed. Please monitor the execution. Press Show Logs to see the execution logs.

- Connect to DBMS
- Retrieve Schema List from Database
- Check Common Server Configuration Issues

Execution Completed Successfully
Fetch finished.

4

Reverse Engineer Database

Select Schemas to Reverse Engineer

Select the schemas you want to include:

Connection Options
Connect to DBMS
 Select Schemas
Retrieve Objects
Select Objects
Reverse Engineer
Results

5

Reverse Engineer Database

Retrieve and Reverse Engineer Schema Objects

The following tasks will now be executed. Please monitor the execution. Press Show Logs to see the execution logs.

- Retrieve Objects from Selected Schemas
- Check Results

Retrieval Completed Successfully
Finished.

6

Reverse Engineer Database

Select Objects to Reverse Engineer

Import MySQL Table Objects
4 Total Objects, 4 Selected

Connection Options
Connect to DBMS
Select Schemas
Retrieve Objects
 Select Objects
Reverse Engineer
Results

7

Reverse Engineer Database

Reverse Engineering Progress

The following tasks will now be executed. Please monitor the execution. Press Show Logs to see the execution logs.

- Reverse Engineer Selected Objects
- Place Objects on Diagram

Operation Completed Successfully

8

Customers

- CustomerID INT
- FirstName VARCHAR(50)
- LastName VARCHAR(50)
- Email VARCHAR(100)
- Phone VARCHAR(15)

Indexes

Orders

- OrderID INT
- CustomerID INT
- OrderDate DATE

Indexes

Products

- ProductID INT
- ProductName VARCHAR(100)
- Price DECIMAL(10,2)
- Stock INT

Indexes

OrderItems

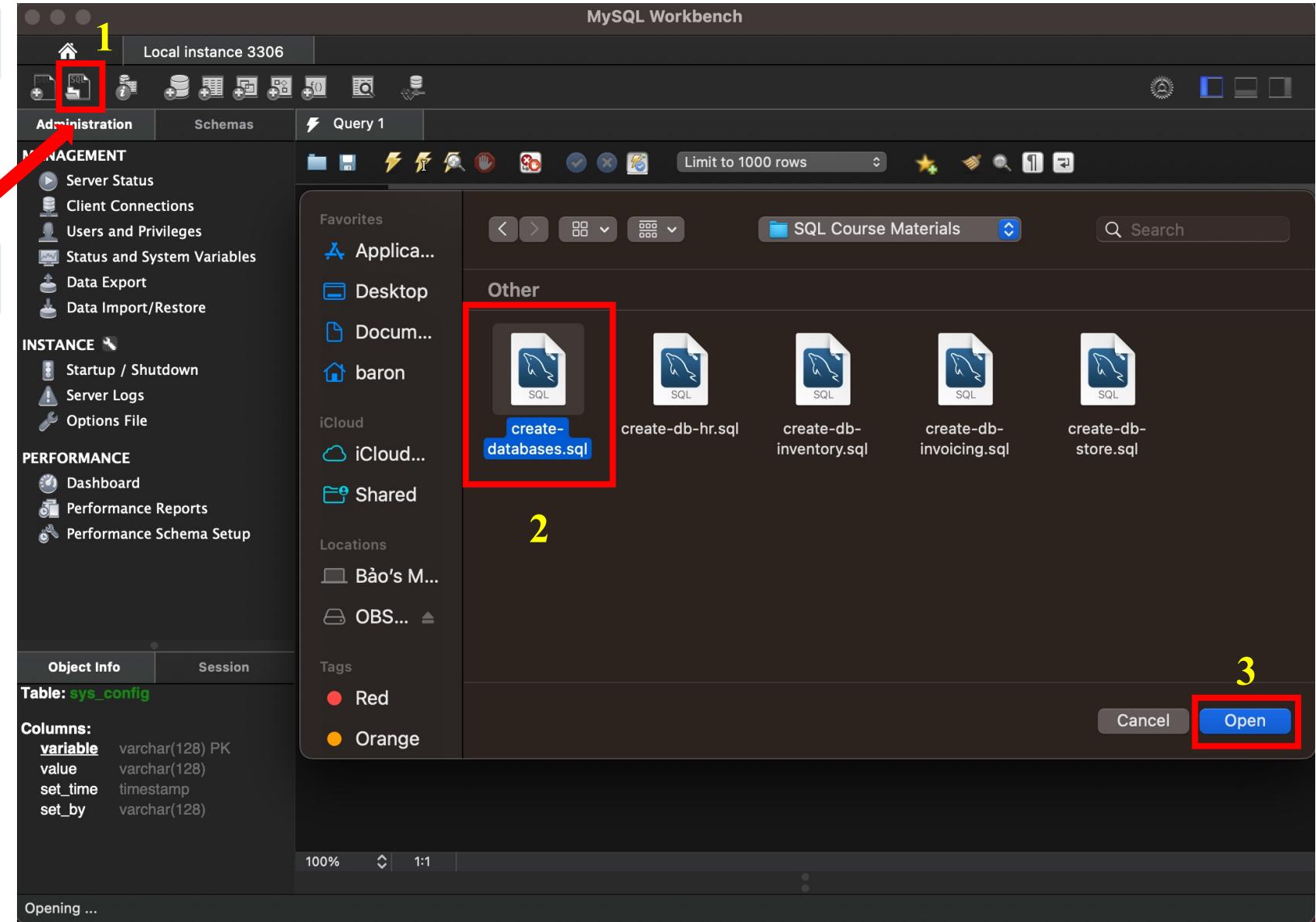
- OrderItemID INT
- OrderID INT
- ProductID INT
- Quantity INT

Indexes

Create Databases From Script

Download the pre-created database script here

Click to open the downloaded script

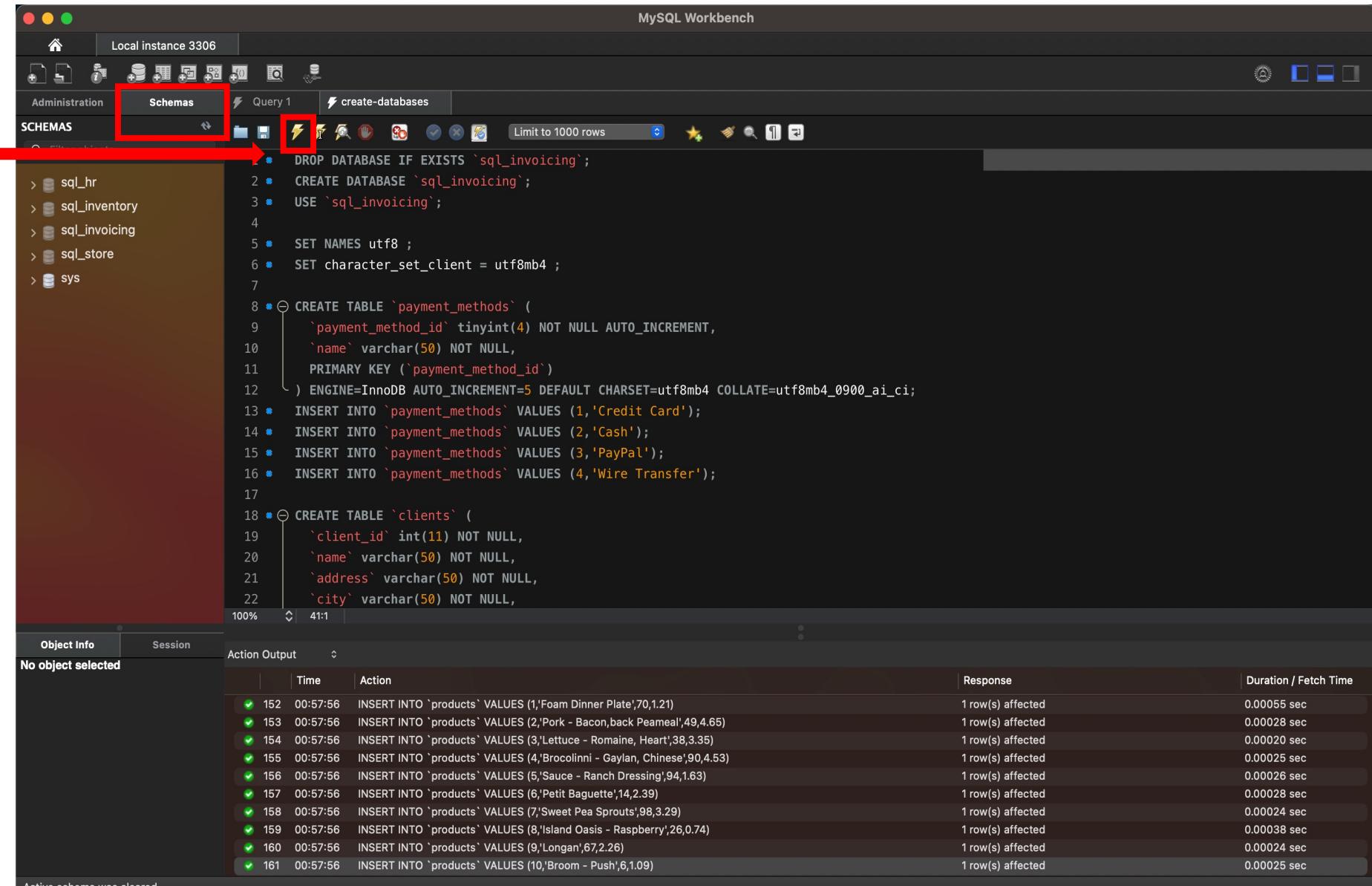


Create Databases

Click the thunder icon to execute and create our databases

NOTE: This button can be used to execute SELECTED SQL code. To execute ALL code, UNSELECT everything

Go to Schemas tab on the Navigation panels and click the refresh icon to see the created schemas



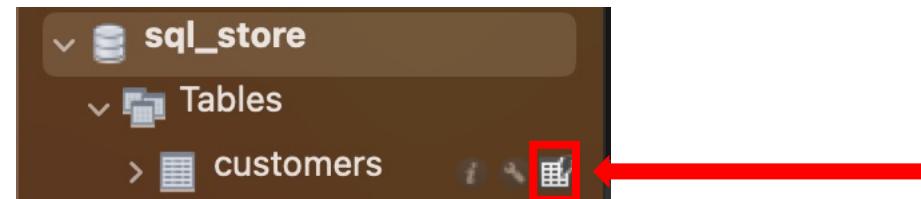
The screenshot shows the MySQL Workbench interface with the following details:

- Schemas Tab:** The "Schemas" tab is selected in the navigation bar.
- Query Editor:** The "Query 1" tab contains the following SQL code:


```

1 DROP DATABASE IF EXISTS `sql_invoicing`;
2 CREATE DATABASE `sql_invoicing`;
3 USE `sql_invoicing`;
4
5 SET NAMES utf8 ;
6 SET character_set_client = utf8mb4 ;
7
8 CREATE TABLE `payment_methods` (
9     `payment_method_id` tinyint(4) NOT NULL AUTO_INCREMENT,
10    `name` varchar(50) NOT NULL,
11    PRIMARY KEY (`payment_method_id`)
12 ) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
13 INSERT INTO `payment_methods` VALUES (1,'Credit Card');
14 INSERT INTO `payment_methods` VALUES (2,'Cash');
15 INSERT INTO `payment_methods` VALUES (3,'PayPal');
16 INSERT INTO `payment_methods` VALUES (4,'Wire Transfer');
17
18 CREATE TABLE `clients` (
19     `client_id` int(11) NOT NULL,
20    `name` varchar(50) NOT NULL,
21    `address` varchar(50) NOT NULL,
22    `city` varchar(50) NOT NULL,
      
```
- Action Output:** The "Action" section shows the results of the executed queries, including the insertion of 16 rows into the "products" table.
- Object Info:** The "Object Info" tab shows the status: "No object selected".
- Session:** The "Session" tab shows the status: "Active schema was cleared".

Explore the data



Go to Schemas tab on the Navigation panel, select any schemas and table and click on this icon to view the data.

SQL-AI Assistant

Prompt

Given the description: '*An online shopping system is a digital platform that allows customers to browse, select, and purchase products over the internet. It involves several key components, including customers, products, and orders,*' identify the corresponding database tables.

Chat GPT's Response

1. Customers		
Stores information about users of the system.		
Field Name	Data Type	Description
CustomerID	INT (Primary Key)	Unique identifier for each customer
Name	VARCHAR	Customer's full name
Email	VARCHAR (UNIQUE)	Customer's email address
PasswordHash	VARCHAR	Encrypted password
PhoneNumber	VARCHAR	Customer's phone number
ShippingAddress	TEXT	Customer's shipping address
BillingAddress	TEXT	Customer's billing address

2. Products		
Contains details of all items available for purchase.		
Field Name	Data Type	Description
ProductID	INT (Primary Key)	Unique identifier for each product
Name	VARCHAR	Name of the product
Description	TEXT	Detailed description
Price	DECIMAL	Product price
StockQuantity	INT	Number of units available
Category	VARCHAR	Product category
ImageURL	VARCHAR	Link to product image

3. Orders		
Captures information about each customer purchase.		
Field Name	Data Type	Description
OrderID	INT (Primary Key)	Unique identifier for each order
CustomerID	INT (Foreign Key)	Refers to the customer placing the order
OrderDate	DATETIME	Date and time of order
TotalAmount	DECIMAL	Total price for the order
PaymentStatus	VARCHAR	e.g., Paid, Pending
ShippingStatus	VARCHAR	e.g., Shipped, Delivered
ShippingAddress	TEXT	Address to which order is delivered

SQL-AI Assistant

Prompt

Write SQL statements to create the following tables: customer, product, order, and order_detail."

Chat GPT's Response

-- Table: customer

```
CREATE TABLE customer (
    customer_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    email VARCHAR(100) NOT NULL UNIQUE,
    password_hash VARCHAR(255) NOT NULL,
    phone_number VARCHAR(20),
    shipping_address TEXT,
    billing_address TEXT
);
```

-- Table: product

```
CREATE TABLE product (
    product_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    description TEXT,
    price DECIMAL(10, 2) NOT NULL,
    stock_quantity INT NOT NULL,
    category VARCHAR(100),
    image_url VARCHAR(255)
);
```

-- Table: `order`

```
CREATE TABLE `order` (
    order_id INT PRIMARY KEY AUTO_INCREMENT,
    customer_id INT NOT NULL,
    order_date DATETIME DEFAULT CURRENT_TIMESTAMP,
    total_amount DECIMAL(10, 2) NOT NULL,
    payment_status VARCHAR(50),
    shipping_status VARCHAR(50),
    shipping_address TEXT,
    FOREIGN KEY (customer_id) REFERENCES customer(customer_id)
);
```

-- Table: order_detail

```
CREATE TABLE order_detail (
    order_detail_id INT PRIMARY KEY AUTO_INCREMENT,
    order_id INT NOT NULL,
    product_id INT NOT NULL,
    quantity INT NOT NULL,
    price_at_purchase DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (order_id) REFERENCES `order`(order_id),
    FOREIGN KEY (product_id) REFERENCES product(product_id)
);
```

Outline



- **Introduction to Database**
- **How to install MySQL**
- **How to create Database**
- **How to use SQL for data analysis**
- **Kahoot Quiz**
- **Summary**



SQL Queries

SELECT

SQL (Structured Query Language) is a powerful tool for data analysis. It allows you to query and manipulate databases efficiently

SELECT

1/ Select all data from a table

```
SELECT * FROM <schema>.<table>;
```

```
SELECT * FROM sql_store.orders;
```

NOTE: You can use the keyword USE to specify the schema you want to query from beforehand

```
USE sql_store;
SELECT * FROM orders;
```

order_id	customer_id	order_date	status	comments	shipped_date	shipper_id
1	6	2019-01-30	1	NULL	NULL	NULL
2	7	2018-08-02	2	NULL	2018-08-03	4
3	8	2017-12-01	1	NULL	NULL	NULL
4	2	2017-01-22	1	NULL	NULL	NULL
5	5	2017-08-25	2		2017-08-26	3
6	10	2018-11-18	1	Aliquam erat volutpat. In congue.	NULL	NULL
7	2	2018-09-22	2	NULL	2018-09-23	4
8	5	2018-06-08	1	Mauris enim leo, rhoncus sed, vestibulum sit am...	NULL	NULL
9	10	2017-07-05	2	Nulla mollis molestie lorem. Quisque ut erat.	2017-07-06	1
10	6	2018-04-22	2	NULL	2018-04-23	2
NULL	NULL	NULL	NULL	NULL	NULL	NULL

2/ Select specific columns

```
SELECT <column>,<column>,<...>
FROM <table>;
```

```
SELECT first_name, last_name
FROM customers;
```

first_name	last_name
Babara	MacCaffrey
Ines	Brushfield
Freddi	Boagey
Ambur	Roseburgh
Clemmie	Betchley
Elka	Twiddell
Ilene	Dowson
Thacher	Naseby
Romola	Rumgay
Levy	Mynett

SELECT

3/ Select with modify values

```
SELECT
    points,
    points + 10,
    points / 10
FROM customers;
```

	points	points + 10	points / 10
	2273	2283	227.3000
	947	957	94.7000
	2967	2977	296.7000
	457	467	45.7000
	3675	3685	367.5000
	3073	3083	307.3000
	1672	1682	167.2000
	205	215	20.5000
	1486	1496	148.6000
	796	806	79.6000

4/ Select with Alias

```
SELECT <column> AS <alias>
FROM <table>;
```

```
SELECT
    points,
    points * 110 / 100,
    points * 110 / 100 AS VAT
FROM customers;
```

	points	points * 110 / 100	VAT
	2273	2500.3000	2500.3000
	947	1041.7000	1041.7000
	2967	3263.7000	3263.7000
	457	502.7000	502.7000
	3675	4042.5000	4042.5000
	3073	3380.3000	3380.3000
	1672	1839.2000	1839.2000
	205	225.5000	225.5000
	1486	1634.6000	1634.6000
	796	875.6000	875.6000

SELECT

5/ Select unique values of a column

```
SELECT state
FROM customers;
```

state
VA
VA
CO
FL
TX
IL
TN
FL
CA
GA

```
SELECT DISTINCT state
FROM customers;
```

state
VA
CO
FL
TX
IL
TN
CA
GA

6/ Select all the products:

- Name
- Unit price
- New price (Unit price * 1.1)

```
SELECT
    name,
    unit_price,
    unit_price * 1.1 AS new_price
FROM products;
```

name	unit_price	new_price
Foam Dinner Plate	1.21	1.331
Pork - Bacon,back Peameal	4.65	5.115
Lettuce - Romaine, Heart	3.35	3.685
Brocolinni - Gaylan, Chinese	4.53	4.983
Sauce - Ranch Dressing	1.63	1.793
Petit Baguette	2.39	2.629
Sweet Pea Sprouts	3.29	3.619
Island Oasis - Raspberry	0.74	0.814
Longan	2.26	2.486
Broom - Push	1.09	1.199



<https://aivietnam.edu.vn/space/selab-sql-animate-7>

SQL Query Visualization

See how data is transformed through each step of SQL query execution

Choose SQL Clause:

SELECT

Available Tables

Your SQL Query:

1 `SELECT name, age FROM users WHERE age > 25 ORDER BY age;`

Execute

Reset

Execution Pipeline

› Initial Data

WHERE Clause Filter

ORDER BY Sorting

SELECT Columns

Initial Data

Initial Data

The original table with all data

1 `SELECT name, age FROM users WHERE age > 25 ORDER BY age;`

id	name	age	city	department	salary
1	Alice	24	New York	Engineering	85000



SQL Queries

WHERE

WHERE

1/ Select data with condition

```
SELECT <column>
FROM <table>
WHERE <condition>
```

```
SELECT *
FROM customers
WHERE points > 3000;
```

2/ Comparison Operators

<, >, <=, >=, =, !=, <>

3/ Select every customer from state ‘VA’

```
SELECT *  
FROM customers  
WHERE state = 'VA';
```

4/ Select every customer not from state 'VA'

```
SELECT *  
FROM customers  
WHERE state != 'VA';
```

WHERE

5/ Get the orders placed after 2018

```
SELECT *
FROM orders
WHERE order_date >= '2018-01-01';
```

order_id	customer_id	order_date	status	comments	shipped_date	shipper_id
1	6	2019-01-30	1	NULL	NULL	NULL
2	7	2018-08-02	2	NULL	2018-08-03	4
6	10	2018-11-18	1	Aliquam erat volutpat. In congue.	NULL	NULL
7	2	2018-09-22	2	NULL	2018-09-23	4
8	5	2018-06-08	1	Mauris enim leo, rhoncus sed, vestibulum sit am...	NULL	NULL
10	6	2018-04-22	2	NULL	2018-04-23	2
NULL	NULL	NULL	NULL	NULL	NULL	NULL

7/ From order_items table, get order_id #6 and total price is less than 30

```
SELECT *, unit_price * quantity AS total_price
FROM order_items
WHERE
    order_id = 6 AND unit_price * quantity < 30;
```

order_id	product_id	quantity	unit_price	total_price
6	2	4	3.28	13.12
6	3	4	7.46	29.84
6	5	1	3.45	3.45

6/ The AND, OR, and NOT Operators

AND Truth Table

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR Truth Table

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

NOT Truth Table

A	B
0	1
1	0

NOT '<' is '>='

NOT '>' is '<='

NOT 'AND' is 'OR'

NOT 'OR' is 'AND'

NOT '=' is '!='

NOT '<>' is '='



SQL Queries

IN - BETWEEN

IN - BETWEEN

**1/ Select all customer from state (use OR):
VA, GA, FL**

```
SELECT *
FROM customers
WHERE
    state = 'VA' OR state = 'GA' OR state = 'FL';
```

**2/ Select all customer from state (use IN):
VA, GA, FL**

```
SELECT *
FROM customers
WHERE
    state IN ('VA', 'GA', 'FL');
```

IN - BETWEEN

**3/ Select all customer from state VA, GA, FL
points greater than 300 but less than 2000**

```
SELECT * FROM customers  
WHERE  
    state IN ('VA', 'GA', 'FL')  
    AND points > 300  
    AND points < 2000;
```

**4/ Select all customer from state VA, GA, FL
points BETWEEN 300 - 2000**

```
SELECT * FROM customers
WHERE
    state IN ('VA', 'GA', 'FL')
    AND points BETWEEN 300 AND 2000;
```

customer_id	first_name	last_name	birth_date	phone	address	city	state	points
2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Commercial Trail	Hampton	VA	947
4	Ambur	Roseburgh	1974-04-14	407-231-8017	30 Arapahoe Terrace	Orlando	FL	457
10	Levy	Mynett	1969-10-13	404-246-3370	68 Lawn Avenue	Atlanta	GA	796

customer_id	first_name	last_name	birth_date	phone	address	city	state	points
2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Commercial Trail	Hampton	VA	947
4	Ambur	Roseburgh	1974-04-14	407-231-8017	30 Arapahoe Terrace	Orlando	FL	457
10	Levy	Mynett	1969-10-13	404-246-3370	68 Lawn Avenue	Atlanta	GA	796



SQL Queries

IS NULL - ORDER BY - LIMIT

IS NULL - ORDER BY - LIMIT

1/ Select all customer don't have a phone number

```
SELECT *  
FROM customers  
WHERE phone IS NULL;
```

2/ Select all customer who have a phone number

```
SELECT *  
FROM customers  
WHERE phone IS NOT NULL;
```

customer_id	first_name	last_name	birth_date	phone	address	city	state	points
1	Babara	MacCaffrey	1986-03-28	781-932-9754	0 Sage Terrace	Waltham	VA	2273
2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Commercial Trail	Hampton	VA	947
3	Freddi	Boagey	1985-02-07	719-724-7869	251 Springs Junction	Colorado Springs	CO	2967
4	Ambur	Roseburgh	1974-04-14	407-231-8017	30 Arapahoe Terrace	Orlando	FL	457
6	Elka	Twiddell	1991-09-04	312-480-8498	7 Manley Drive	Chicago	IL	3073
7	Ilene	Dowson	1964-08-30	615-641-4759	50 Lillian Crossing	Nashville	TN	1672
8	Thacher	Naseby	1993-07-17	941-527-3977	538 Mosinee Center	Sarasota	FL	205
9	Romola	Rumgay	1992-05-23	559-181-3744	3520 Ohio Trail	Visalia	CA	1486
10	Levy	Mynett	1969-10-13	404-246-3370	68 Lawn Avenue	Atlanta	GA	796

IS NULL - ORDER BY - LIMIT

3/ Order the customer by points in ascending order

```
SELECT * FROM customers
ORDER BY points;
```

customer_id	first_name	last_name	birth_date	phone	address	city	state	points
8	Thacher	Naseby	1993-07-17	941-527-3977	538 Mosinee Center	Sarasota	FL	205
4	Ambur	Roseburgh	1974-04-14	407-231-8017	30 Arapahoe Terrace	Orlando	FL	457
10	Levy	Mynett	1969-10-13	404-246-3370	68 Lawn Avenue	Atlanta	GA	796
2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Commercial Trail	Hampton	VA	947
9	Romola	Rumgay	1992-05-23	559-181-3744	3520 Ohio Trail	Visalia	CA	1486
7	Ilene	Dowson	1964-08-30	615-641-4759	50 Lillian Crossing	Nashville	TN	1672
1	Babara	MacCaffrey	1986-03-28	781-932-9754	0 Sage Terrace	Waltham	VA	2273
3	Freddi	Boagey	1985-02-07	719-724-7869	251 Springs Junction	Colorado Springs	CO	2967
6	Elka	Twiddell	1991-09-04	312-480-8498	7 Manley Drive	Chicago	IL	3073
5	Clemmie	Betchley	1973-11-07	NULL	5 Spohn Circle	Arlington	TX	3675

4/ Get the customer have points < 1000 and order them in descending order by points

```
SELECT * FROM customers
WHERE points < 1000
ORDER BY points DESC;
```

customer_id	first_name	last_name	birth_date	phone	address	city	state	points
2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Commercial Trail	Hampton	VA	947
10	Levy	Mynett	1969-10-13	404-246-3370	68 Lawn Avenue	Atlanta	GA	796
4	Ambur	Roseburgh	1974-04-14	407-231-8017	30 Arapahoe Terrace	Orlando	FL	457
8	Thacher	Naseby	1993-07-17	941-527-3977	538 Mosinee Center	Sarasota	FL	205

IS NULL - ORDER BY - LIMIT

5/ Get three customer with the highest points

```
SELECT * FROM customers
ORDER BY points DESC
LIMIT 3;
```

customer_id	first_name	last_name	birth_date	phone	address	city	state	points
5	Clemmie	Betchley	1973-11-07	NULL	5 Spohn Circle	Arlington	TX	3675
6	Elka	Twiddell	1991-09-04	312-480-8498	7 Manley Drive	Chicago	IL	3073
3	Freddi	Boagey	1985-02-07	719-724-7869	251 Springs Junction	Colorado Springs	CO	2967

6/ Get four customer with the highest points (not including the first 3)

```
SELECT * FROM customers
ORDER BY points DESC
LIMIT 3, 4;
```

customer_id	first_name	last_name	birth_date	phone	address	city	state	points
1	Babara	MacCaffrey	1986-03-28	781-932-9754	0 Sage Terrace	Waltham	VA	2273
7	Ilene	Dowson	1964-08-30	615-641-4759	50 Lillian Crossing	Nashville	TN	1672
9	Romola	Rumgay	1992-05-23	559-181-3744	3520 Ohio Trail	Visalia	CA	1486
2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Commercial Trail	Hampton	VA	947

LIMIT offset, count

3 is the **offset** – skip the first 3 rows of the sorted result.
 4 is the **count** – return the next 4 rows after skipping.



SQL Queries

LIKE - REGEXP

LIKE

1/ Select all customer whose last name start with ‘B’

```
SELECT *
FROM customers
WHERE last_name LIKE 'B%';
```

2/ Select all customer who have a ‘b’ in their last name

```
SELECT *
FROM customers
WHERE last_name LIKE '%B%';
```

NOTE: ‘%’ is used to represent any number of characters

LIKE

3/ Select all customers whose last name is 6 letters long and end with ‘y’

```
SELECT *
FROM customers
WHERE last_name LIKE '_____y';
```

customer_id	fir...	last_name	birth_date	phone	address	city	state	points
3	Fr...	Boagey	1985-02-07	719-724-7869	251 Springs Junction	Colorado Springs	CO	2967
8	Th...	Naseby	1993-07-17	941-527-3977	538 Mosinee Center	Sarasota	FL	205
9	R...	Rumgay	1992-05-23	559-181-3744	3520 Ohio Trail	Visalia	CA	1486

NOTE: ‘_’ is used to represent one characters

**4/ Get the customer whose
-- addresses contain ‘TRAIL’ or ‘AVENUE’
-- phone numbers end with 9---**

```
SELECT *
FROM customers
WHERE
    address LIKE '%trail%' OR
    address LIKE '%avenue%';
```

customer_id	fir...	last_name	birth_date	phone	address	city	state	points
2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Commercial Trail	Hampton	VA	947
9	R...	Rumgay	1992-05-23	559-181-3744	3520 Ohio Trail	Visalia	CA	1486
10	Levy	Mynett	1969-10-13	404-246-3370	68 Lawn Avenue	Atlanta	GA	796

```
SELECT *
FROM customers
WHERE phone LIKE '%9____';
```

customer_id	fir...	last_name	birth_date	phone	address	city	state	points
1	Ba...	MacCaffrey	1986-03-28	781-932-9754	0 Sage Terrace	Waltham	VA	2273
2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Commercial Trail	Hampton	VA	947

REGEXP

1/ Get the customer whose first name are ELKA or AMBUR

```
SELECT *
FROM customers
WHERE first_name REGEXP 'ELKA | AMBUR';
```

customer_id	first_name	last_name	birth_date	phone	address	city	state	points
4	Ambur	Roseburgh	1974-04-14	407-231-8017	30 Arapahoe Terrace	Orlando	FL	457
6	Elka	Twiddell	1991-09-04	312-480-8498	7 Manley Drive	Chicago	IL	3073

NOTE: ‘|’ is used to represent alternation.
‘|’ = OR

2/ Get the customer whose last name end with EY or ON

```
SELECT *
FROM customers
WHERE last_name REGEXP 'EY$|ON$';
```

customer_id	first_name	last_name	birth_date	phone	address	city	state	points
1	Babara	MacCaffrey	1986-03-28	781-932-9754	0 Sage Terrace	Waltham	VA	2273
3	Freddi	Boagey	1985-02-07	719-724-7869	251 Springs Junction	Colorado Springs	CO	2967
5	Clemmie	Betchley	1973-11-07	NULL	5 Spohn Circle	Arlington	TX	3675
7	Ilene	Dowson	1964-08-30	615-641-4759	50 Lillian Crossing	Nashville	TN	1672

NOTE: ‘\$’ is used to match the end of a string.

REGEXP

3/ Get the customer whose last name start with MY or contains SE

```
SELECT *
FROM customers
WHERE last_name REGEXP '^MY|SE';
```

customer_id	first_name	last_name	birth_date	phone	address	city	state	points
4	Ambur	Roseburgh	1974-04-14	407-231-8017	30 Arapahoe Terrace	Orlando	FL	457
8	Thacher	Naseby	1993-07-17	941-527-3977	538 Mosinee Center	Sarasota	FL	205
10	Levy	Mynett	1969-10-13	404-246-3370	68 Lawn Avenue	Atlanta	GA	796
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

NOTE: '^' is used to match the beginning of a string.

4/ Get the customer whose last name contain B followed by R or U

```
SELECT *
FROM customers
WHERE last_name REGEXP 'B[RU]';
```

customer_id	first_name	last_name	birth_date	phone	address	city	state	points
2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Commercial Trail	Hampton	VA	947
4	Ambur	Roseburgh	1974-04-14	407-231-8017	30 Arapahoe Terrace	Orlando	FL	457
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

NOTE: '[]' is used to match any characters in it.

REGEX

basic

syntaxs

REGEXP

REGEX SYNTAX	MEANING	EXAMPLE	MATCHES	DOES NOT MATCH
.	Any single character	go.gle	google, goggle	gogle
[abc]	Any of these character	analy[zs]e	analyse, analyze	analyxe
[a-z]	Any character in this range	demo[2-4]	demo2, demo3	demo1, demo5
[^abc]	None of these characters	analy[^zs]e	analyxe	analyse, analyze
[^a-z]	Not a character in this range	demo[^2-4]	demo1, demo5	demo2, demo3
	Or	demo example	demo, demos, example	test
^	Starts with	^demo	demos, demonstration	my demo
\$	Ends with	demo\$	my demo	demonstration
?	Zero or one times (greedy)	demos?123	demo123, demos123	demoA123
??	Zero or one times (lazy)			
*	Zero or more times (greedy)	goo*gle	google, gooooogle	goggle
*?	Zero or more times (lazy)			
+	One or more times (greedy)	goo+gle	google, gooooogle	gogle, goggle
+?	One or more times (lazy)			
{n}	n times exactly	w{3}	www	w, ww
{n,m}	from n to m times	a{4, 7}	aaaa, aaaaa, aaaaaa, aaaaaaa	aaaaaaaa, aaa, a
{n,}	at least n times	go{2,}gle	google, gooole, goooole	gle, gogle
()	Group	^(demolexample)[0-9]+	demo1, example4	demoexample2
(?:)	Passive group (Useful for filters)			
\	Escape	AU\\$10	AU\$10, AU\$100	AU10, 10
\s	White space			
\S	Non-white space			
\d	Digit character			
\D	Non-digit character			
\w	Word			
\W	Non-word (e.g. punctuation, spaces)			

Outline

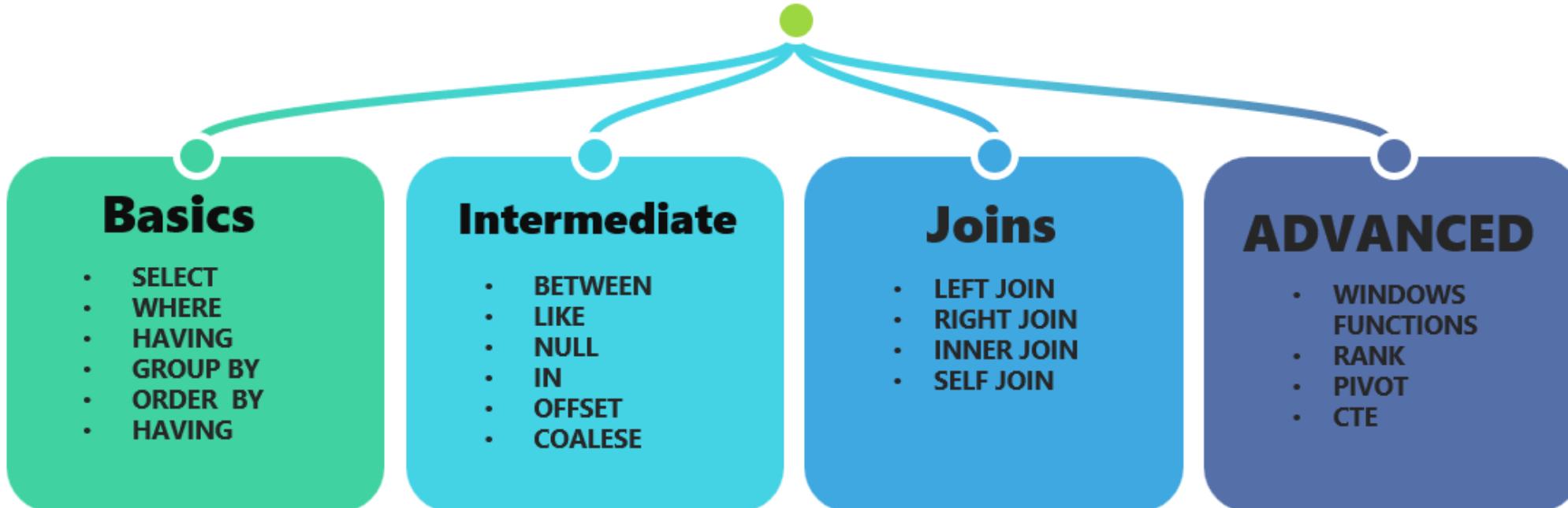
- **Introduction to Database**
- **How to install MySQL**
- **How to create Database**
- **How to use SQL for data analysis**
- **Kahoot Quiz**
- **Summary**



Summary



DATA ANALYST SQL QUERIES





Further study

Retrieve the first name and last name of customers whose points are greater than 1000

customer_id	first_name	last_name	birth_date	phone	address	city	state	points
1	Babara	MacCaffrey	1986-03-28	781-932-9754	0 Sage Terrace	Waltham	MA	2273
2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Commercial Trail	Hampton	VA	947
3	Freddi	Boagey	1985-02-07	719-724-7869	251 Springs Junction	Colorado Springs	CO	2967
4	Ambur	Roseburgh	1974-04-14	407-231-8017	30 Arapahoe Terrace	Orlando	FL	457
5	Clemmie	Betchley	1973-11-07	NULL	5 Spohn Circle	Arlington	TX	3675
6	Elka	Twiddell	1991-09-04	312-480-8498	7 Manley Drive	Chicago	IL	3073
7	Ilene	Dowson	1964-08-30	615-641-4759	50 Lillian Crossing	Nashville	TN	1672
8	Thacher	Naseby	1993-07-17	941-527-3977	538 Mosinee Center	Sarasota	FL	205
9	Romola	Rumgay	1992-05-23	559-181-3744	3520 Ohio Trail	Visalia	CA	1486
10	Levy	Mynett	1969-10-13	404-246-3370	68 Lawn Avenue	Atlanta	GA	796

```
SELECT first_name, last_name  
FROM customers  
WHERE points > 1000;
```

Further study

Retrieve the product name and unit price of products with a quantity in stock between 50 and 100.

product_id	name	quantity_in_stock	unit_price
1	Foam Dinner Plate	70	1.21
2	Pork - Bacon,back Peameal	49	4.65
3	Lettuce - Romaine, Heart	38	3.35
4	Brocolinni - Gaylan, Chinese	90	4.53
5	Sauce - Ranch Dressing	94	1.63
6	Petit Baguette	14	2.39
7	Sweet Pea Sprouts	98	3.29
8	Island Oasis - Raspberry	26	0.74
9	Longan	67	2.26
10	Broom - Push	6	1.09

```
SELECT name, unit_price  
FROM products  
WHERE quantity_in_stock BETWEEN 50 AND 100;
```

Further study

Retrieve the first name and last name of customers whose phone number is null.

customer_id	first_name	last_name	birth_date	phone	address	city	state	points
1	Babara	MacCaffrey	1986-03-28	781-932-9754	0 Sage Terrace	Waltham	MA	2273
2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Commercial Trail	Hampton	VA	947
3	Freddi	Boagey	1985-02-07	719-724-7869	251 Springs Junction	Colorado Springs	CO	2967
4	Ambur	Roseburgh	1974-04-14	407-231-8017	30 Arapahoe Terrace	Orlando	FL	457
5	Clemmie	Betchley	1973-11-07	NULL	5 Spohn Circle	Arlington	TX	3675
6	Elka	Twiddell	1991-09-04	312-480-8498	7 Manley Drive	Chicago	IL	3073
7	Ilene	Dowson	1964-08-30	615-641-4759	50 Lillian Crossing	Nashville	TN	1672
8	Thacher	Naseby	1993-07-17	941-527-3977	538 Mosinee Center	Sarasota	FL	205
9	Romola	Rumgay	1992-05-23	559-181-3744	3520 Ohio Trail	Visalia	CA	1486
10	Levy	Mynett	1969-10-13	404-246-3370	68 Lawn Avenue	Atlanta	GA	796

```
SELECT first_name, last_name
FROM customers
WHERE phone IS NULL;
```

Further study

Retrieve the order ID and order date of the latest 5 orders, ordered by order date in descending order.

order_id	customer_id	order_date	status	comments	shipped_date	shipper_id
1	6	2019-01-30	1	NULL	NULL	NULL
2	7	2018-08-02	2	NULL	2018-08-03	4
3	8	2017-12-01	1	NULL	NULL	NULL
4	2	2017-01-22	1	NULL	NULL	NULL
5	5	2017-08-25	2		2017-08-26	3
6	10	2018-11-18	1	Aliquam erat volutpat. In congue.	NULL	NULL
7	2	2018-09-22	2	NULL	2018-09-23	4
8	5	2018-06-08	1	Mauris enim leo, rhoncus sed, vestibulum sit am...	NULL	NULL
9	10	2017-07-05	2	Nulla mollis molestie lorem. Quisque ut erat.	2017-07-06	1
10	6	2018-04-22	2	NULL	2018-04-23	2
NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
SELECT order_id, order_date
FROM orders
ORDER BY order_date DESC
LIMIT 5;
```

Further study

Retrieve the product name and unit price of products that have a name starting with a vowel (AIUEO).

product_id	name	quantity_in_stock	unit_price
1	Foam Dinner Plate	70	1.21
2	Pork - Bacon,back Peameal	49	4.65
3	Lettuce - Romaine, Heart	38	3.35
4	Brocolinni - Gaylan, Chinese	90	4.53
5	Sauce - Ranch Dressing	94	1.63
6	Petit Baguette	14	2.39
7	Sweet Pea Sprouts	98	3.29
8	Island Oasis - Raspberry	26	0.74
9	Longan	67	2.26
10	Broom - Push	6	1.09

```
SELECT name, unit_price  
FROM products  
WHERE name REGEXP '^[aeiouAEIOU]';
```

Further study

Retrieve the order ID and order date of orders that were placed on weekends (Saturday or Sunday)

order_id	customer_id	order_date	status	comments	shipped_date	shipper_id
1	6	2019-01-30	1	NULL	NULL	NULL
2	7	2018-08-02	2	NULL	2018-08-03	4
3	8	2017-12-01	1	NULL	NULL	NULL
4	2	2017-01-22	1	NULL	NULL	NULL
5	5	2017-08-25	2		2017-08-26	3
6	10	2018-11-18	1	Aliquam erat volutpat. In congue.	NULL	NULL
7	2	2018-09-22	2	NULL	2018-09-23	4
8	5	2018-06-08	1	Mauris enim leo, rhoncus sed, vestibulum sit am...	NULL	NULL
9	10	2017-07-05	2	Nulla mollis molestie lorem. Quisque ut erat.	2017-07-06	1
10	6	2018-04-22	2	NULL	2018-04-23	2
NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
SELECT order_id, order_date
FROM orders
WHERE DAYOFWEEK(order_date) IN (1, 7);
```

