



Toward Data Science Database - SQL

Outline

- What is a Database?
- Installing MySQL
- Create Databases
- SQL Queries – Single table Queries
- Quizzes



What is a Database?

Definition

A collection of information/data.

Stored and organized in some format.

Easily accessed, managed and retrieved

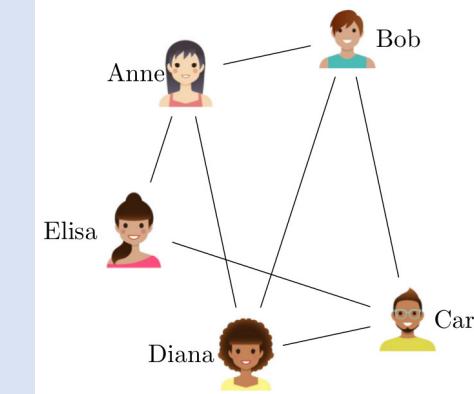
Databases



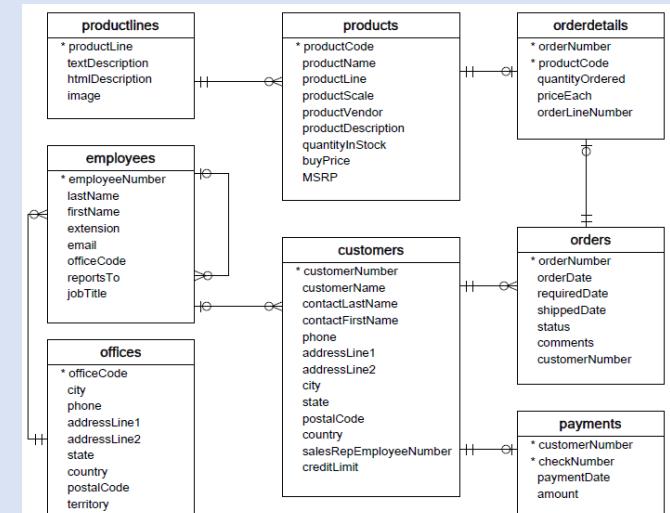
Phone book



Shopping list



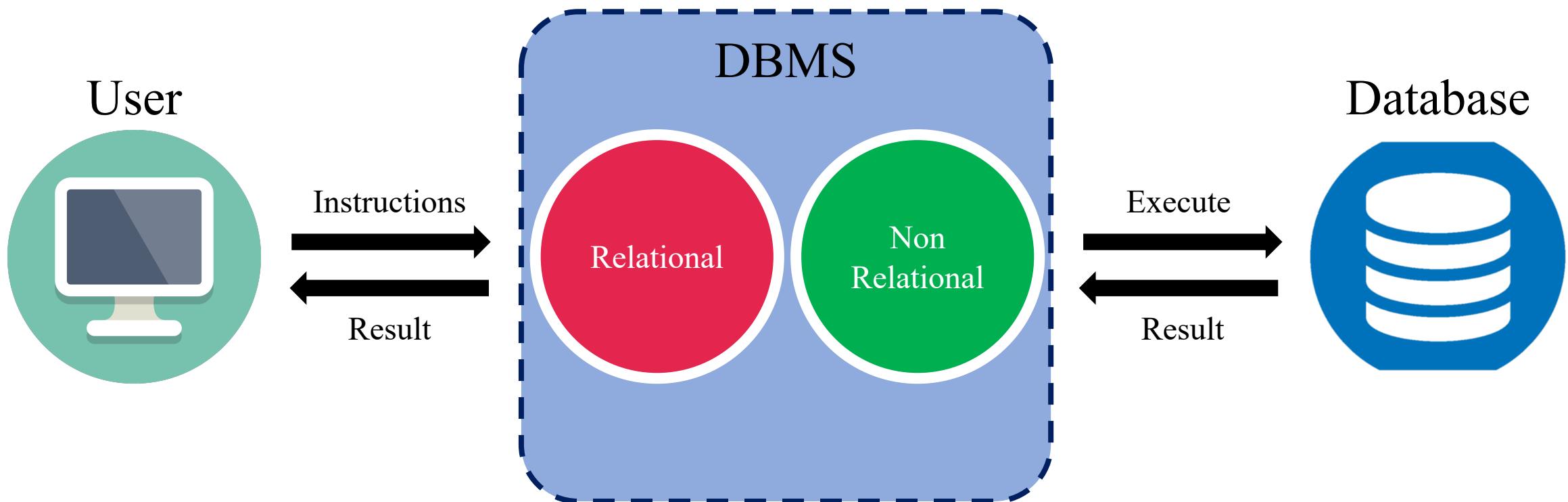
Friend Graph



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



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.](#)

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 Looking for previous GA versions?

Select OS Version: All

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

Version	File Type	Size	Download
macOS 13 (ARM, 64-bit), DMG Archive	8.0.33	559.1M	Download
(mysql-8.0.33-macos13-arm64.dmg)		MD5: 065662283bf45231ea0b47e4abf1e7 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	

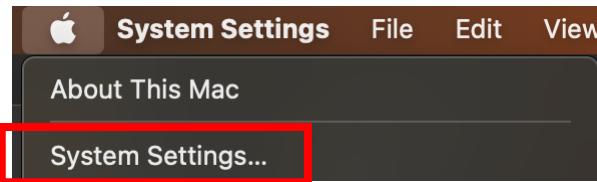


Installing MySQL on MAC

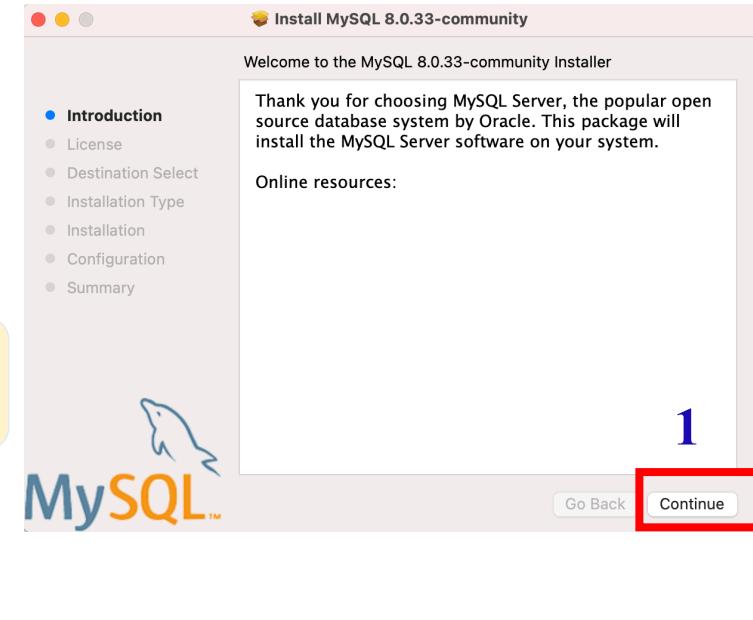
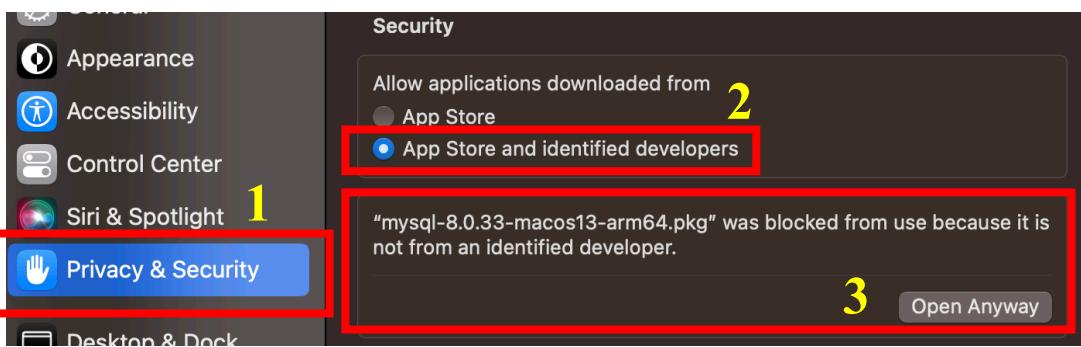
Step 5: Open the download file.



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

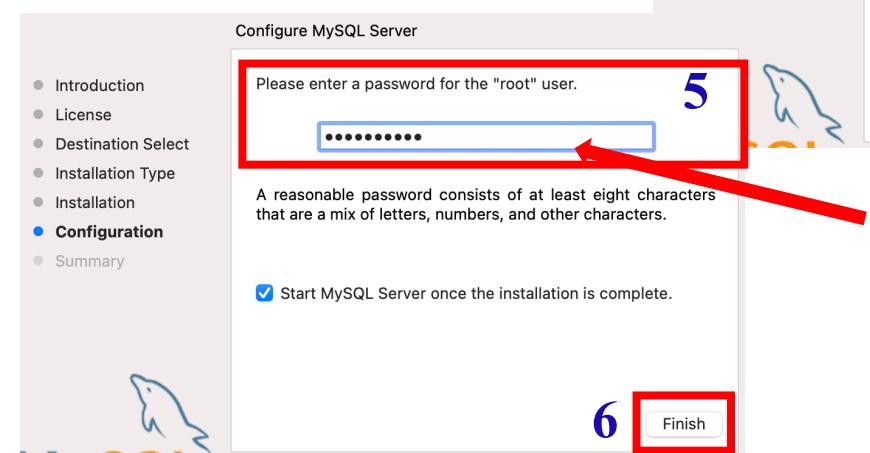
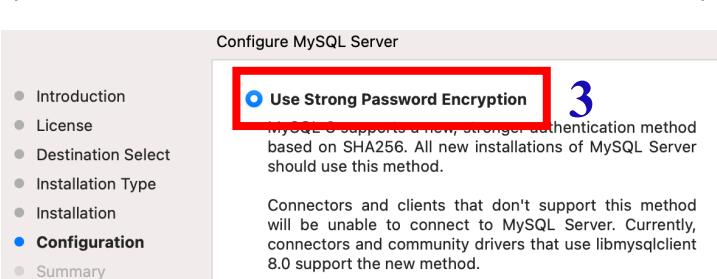
To continue installing the software you must agree to the terms of the software license agreement.

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



The world's most popular open source database

MYSQL.COM DOWNLOADS DOCUMENTATION DEVELOPER ZONE

A red box highlights the "DOWNLOADS" button.

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**

General Availability (GA) Releases Archives

MySQL Workbench 8.0.33

Select Operating System: macOS Select OS Version: All

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

Platform	Version	Size	Download
macOS (ARM, 64-bit), DMG Archive	8.0.33	122.4M	Download
macOS (x86, 64-bit), DMG Archive	8.0.33	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



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.](#)

**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

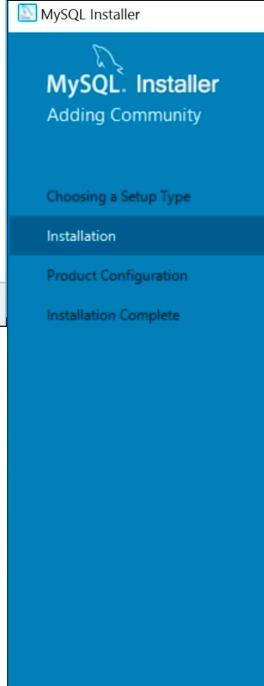
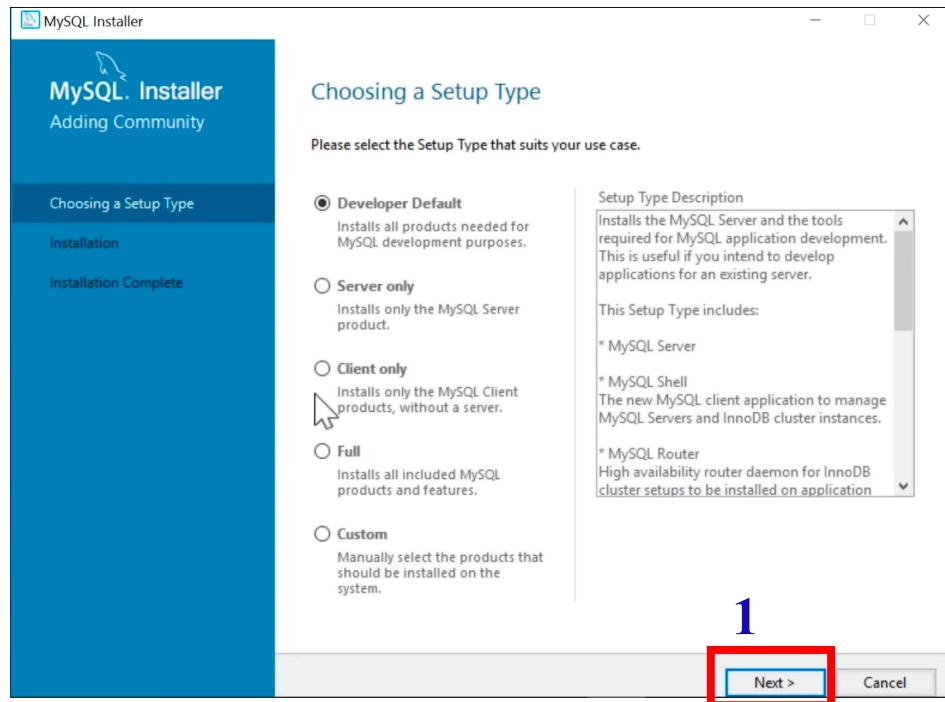
[**Go to Download Page >**](#)

Step 5: Download the first link

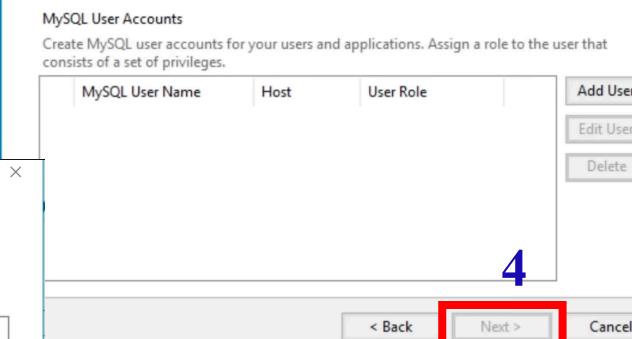
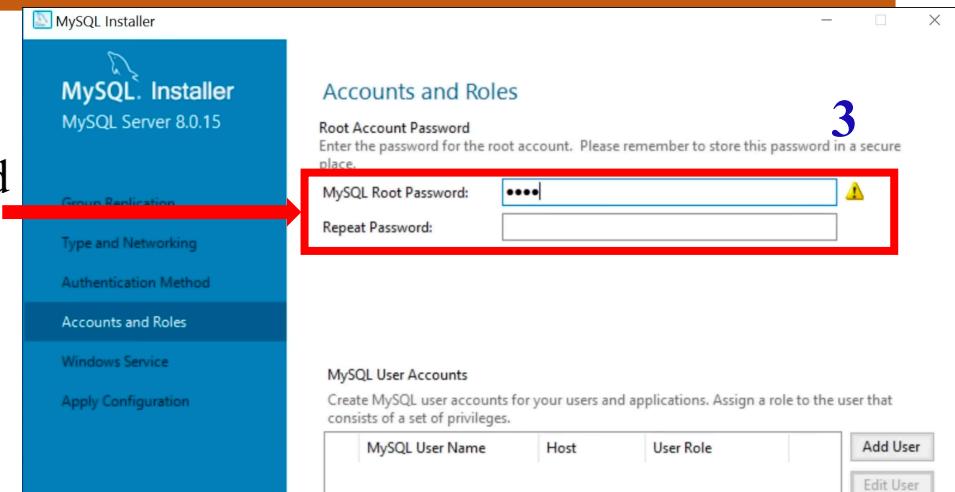
Windows (x86, 32-bit), MSI Installer (mysql-installer-web-community-8.0.15.0.msi)	8.0.15	16.4M	Download
			 MD5: 66363d23d2474113decf683cf84f5820 Signature

Installing MySQL on Windows

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

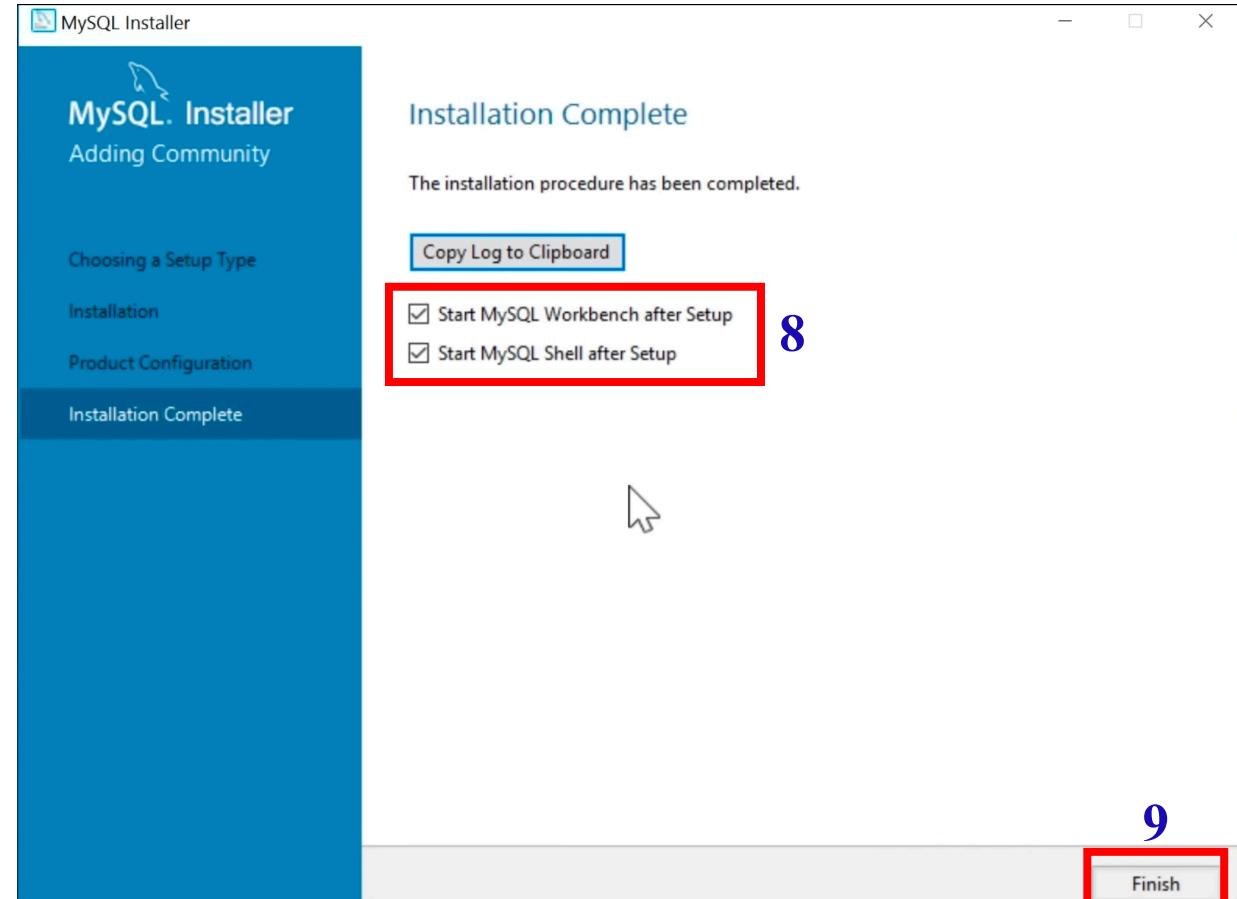
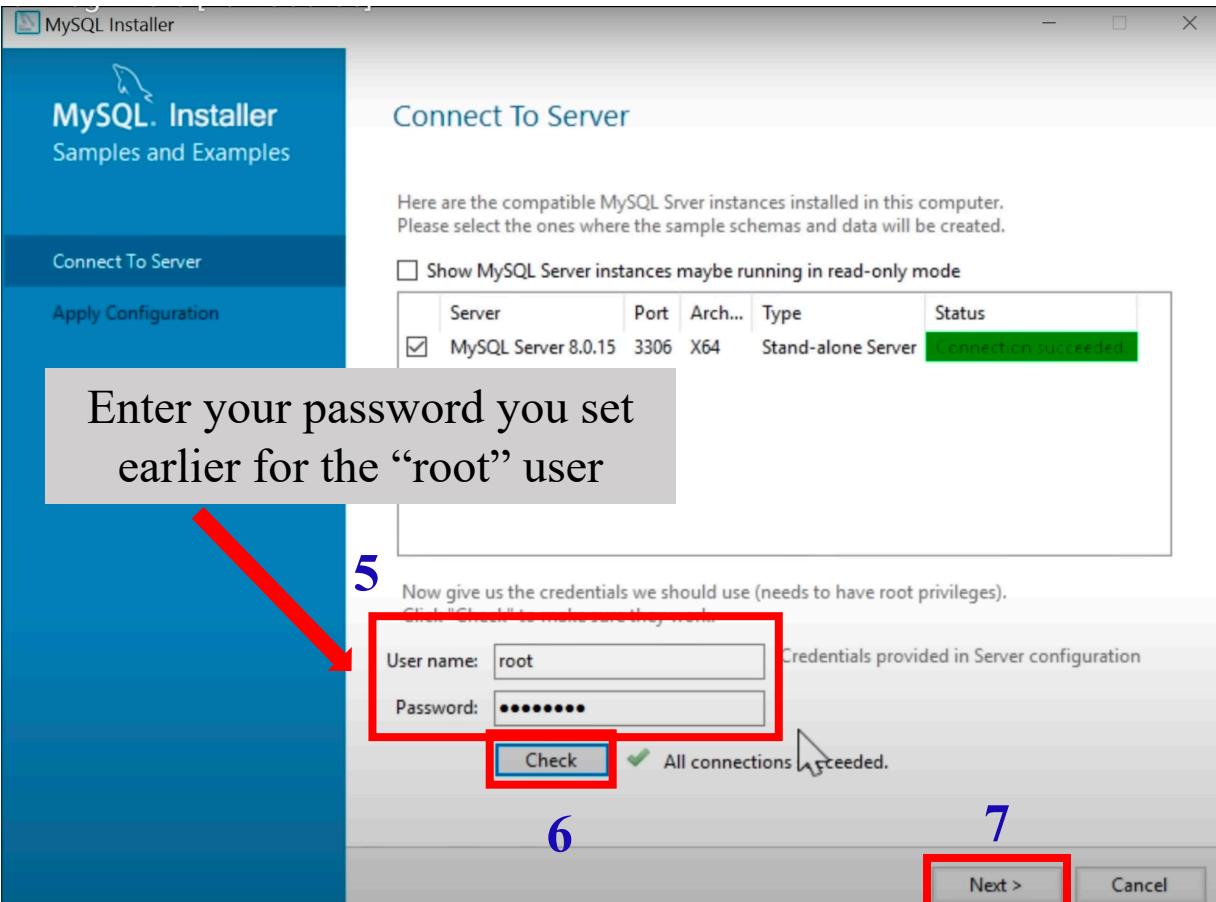


Create your password for the “root” user



Installing MySQL on Windows

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



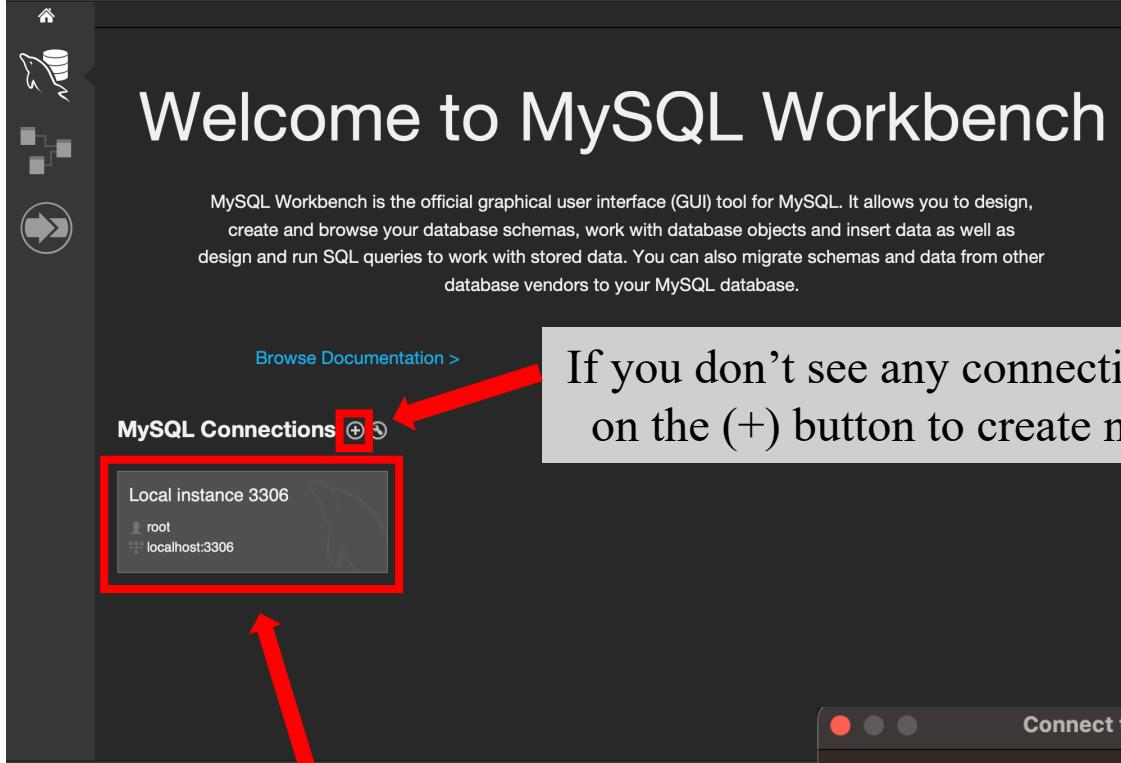


Create Databases



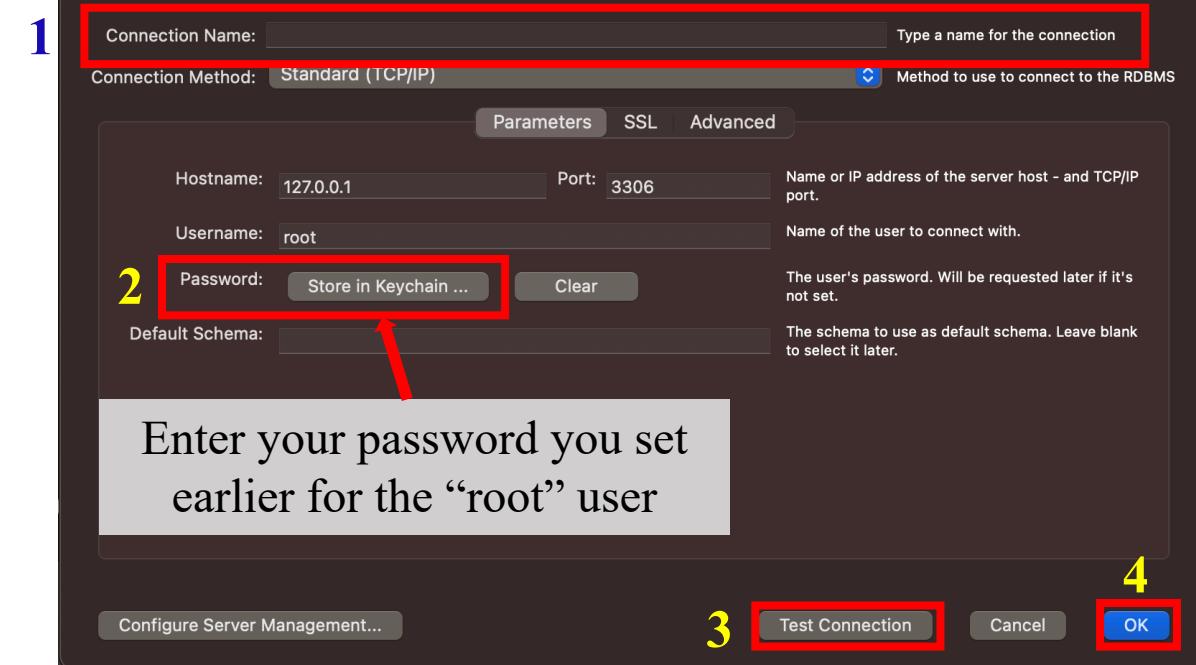
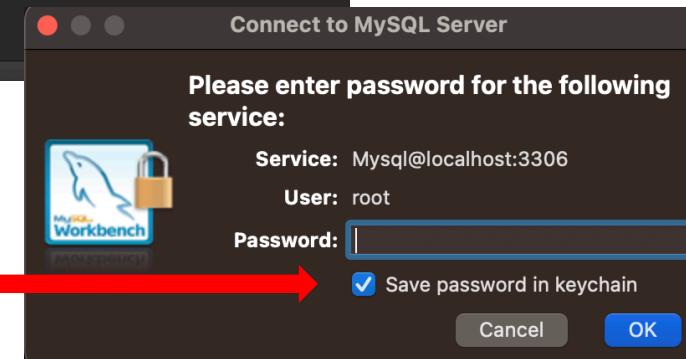
Create Databases

Open MySQL Workbench



You should see the default connection

Click and enter your password to access the server



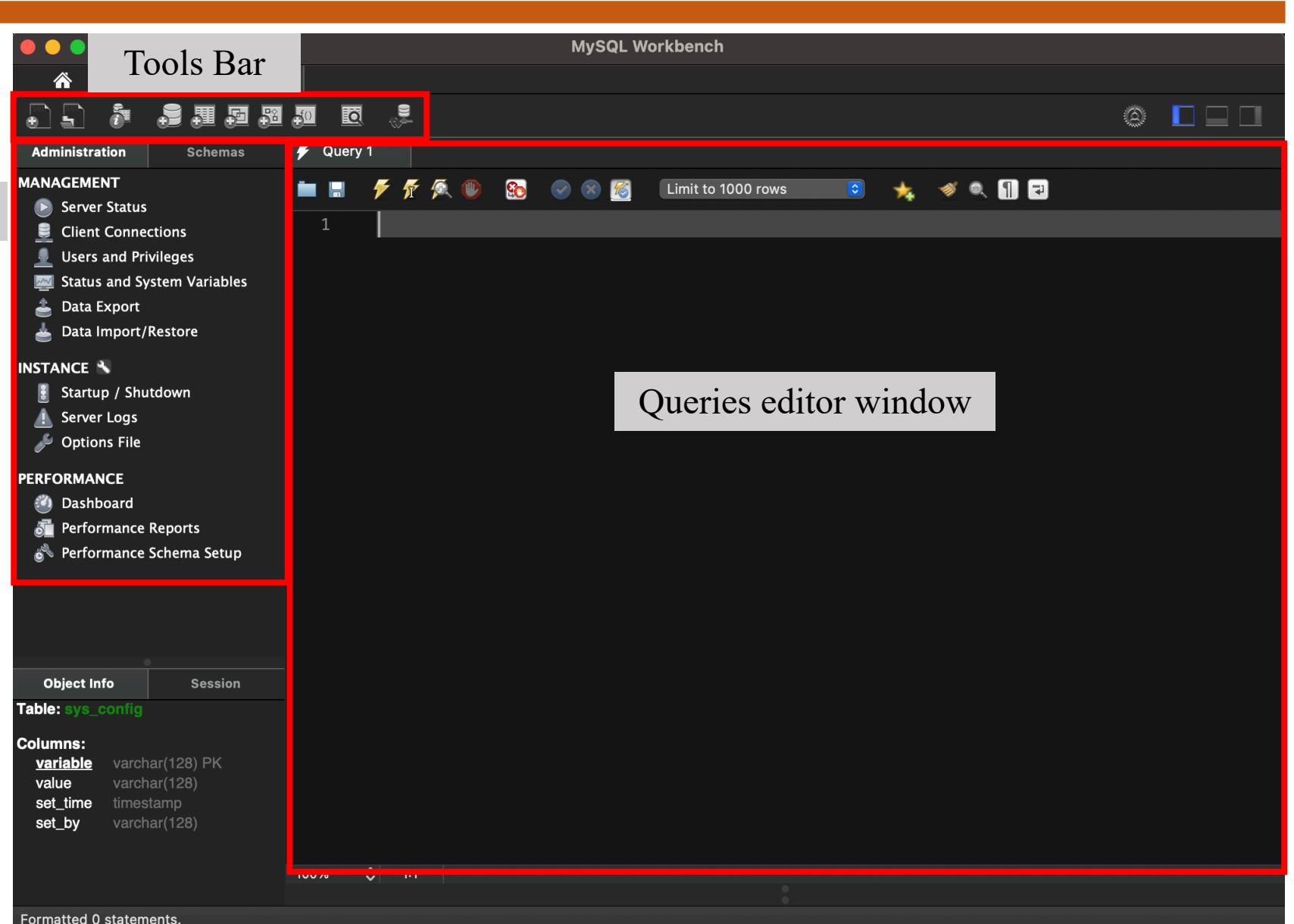


Create Databases

User Interface

Navigation panels

Queries editor window

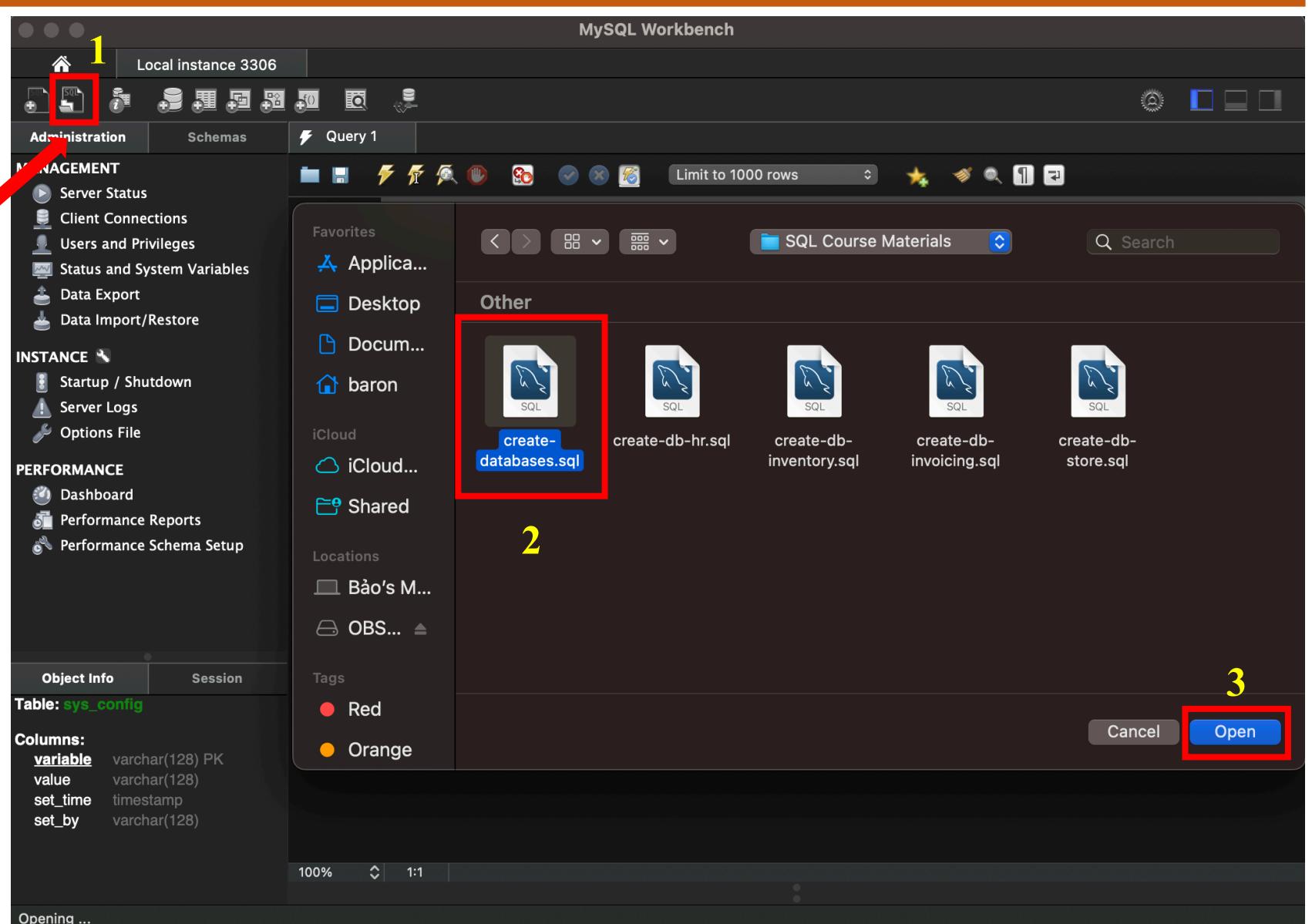




Create Databases

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

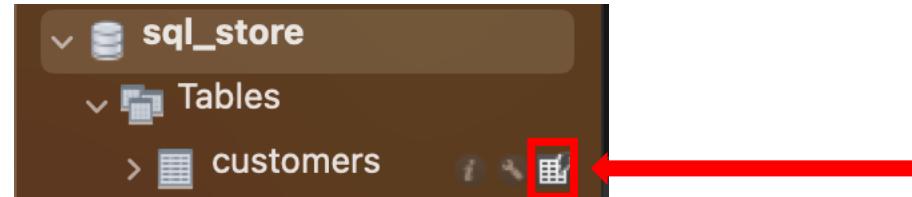
Go to Schemas tab on the Navigation panels and click the refresh icon to see the created schemas

The screenshot shows the MySQL Workbench interface. The top navigation bar has tabs for Administration and Schemas, with Schemas selected. A red box highlights the 'Schemas' tab. Below it, the 'Object Info' and 'Session' panes are visible, with 'No object selected' in Object Info. The main area is the 'Query 1' editor, which contains SQL code for creating a database, switching to it, setting character sets, and creating two tables: 'payment_methods' and 'clients'. The code uses InnoDB storage engine and utf8mb4 charset/collation. The 'Action Output' pane at the bottom shows the execution results for each query, indicating successful insertions into the 'products' table. The status bar at the bottom left says 'Active schema was cleared'.

```
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	Time	Response	Duration / Fetch Time
152 00:57:56	INSERT INTO `products` VALUES (1,'Foam Dinner Plate',70,1.21)	1 row(s) affected	0.00055 sec
153 00:57:56	INSERT INTO `products` VALUES (2,'Pork - Bacon,back Peameal',49,4.65)	1 row(s) affected	0.00028 sec
154 00:57:56	INSERT INTO `products` VALUES (3,'Lettuce - Romaine, Heart',38,3.35)	1 row(s) affected	0.00020 sec
155 00:57:56	INSERT INTO `products` VALUES (4,'Brocolini - Gaylan, Chinese',90,4.53)	1 row(s) affected	0.00025 sec
156 00:57:56	INSERT INTO `products` VALUES (5,'Sauce - Ranch Dressing',94,1.63)	1 row(s) affected	0.00026 sec
157 00:57:56	INSERT INTO `products` VALUES (6,'Petit Baguette',14,2.39)	1 row(s) affected	0.00028 sec
158 00:57:56	INSERT INTO `products` VALUES (7,'Sweet Pea Sprouts',98,3.29)	1 row(s) affected	0.00024 sec
159 00:57:56	INSERT INTO `products` VALUES (8,'Island Oasis - Raspberry',26,0.74)	1 row(s) affected	0.00038 sec
160 00:57:56	INSERT INTO `products` VALUES (9,'Longan',67,2.26)	1 row(s) affected	0.00024 sec
161 00:57:56	INSERT INTO `products` VALUES (10,'Broom - Push',6,1.09)	1 row(s) affected	0.00025 sec

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 Queries

SELECT



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



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;
```

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

**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



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	Barbara	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



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%';
```

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
3	Fr...	Boagey	1985-02-07	719-724-7869	251 Springs Junction	Colorado Springs	CO	2967
4	A...	Roseburgh	1974-04-14	407-231-8017	30 Arapahoe Terrace	Orlando	FL	457
5	Cl...	Betchley	1973-11-07	NULL	5 Spohn Circle	Arlington	TX	3675
8	Th...	Naseby	1993-07-17	941-527-3977	538 Mosinee Center	Sarasota	FL	205

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

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

LIKE

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, goooogle	goggle
*?	Zero or more times (lazy)			
+	One or more times (greedy)	goo+gle	google, goooogle	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, goooogle	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)			



SQL Queries

AGGREGATE FUNCTIONS - OVER

AGGREGATE FUNCTIONS - OVER

1/ Calculate the average points of all customers

```
SELECT AVG(points) AS average_point  
FROM customers;
```

points
2273
947
2967
457
3675
3073
1672
205
1486
796

average_point

1755.1000

NOTE: Aggregate functions take in and apply on all value of the selected column.

2/ Find out the total quantity of every products in stock

```
SELECT SUM(quantity_in_stock) AS total_quantity  
FROM products;
```

quantity_in_sto...
70
49
38
90
94
14
98
26
67
6

total_quantity

552

What if we want to aggregate only on some specifics group??

AGGREGATE FUNCTIONS - OVER

3/ Calculate the average points of all customers by state.

```
SELECT state, AVG(points)
      OVER(PARTITION BY state)
      AS average_state_point
  FROM customers;
```

state	points
VA	2273
VA	947
CO	2967
CO	457
TX	3675
TX	3073
FL	1672
FL	205
GA	1486
GA	796

state	average_state_point
CO	1712.0000
CO	1712.0000
FL	938.5000
FL	938.5000
GA	1141.0000
GA	1141.0000
TX	3374.0000
TX	3374.0000
VA	1610.0000
VA	1610.0000

NOTE: OVER dictate the window which the aggregate function will act on

4/ Calculate the cumulative sum points of all customers by state

```
SELECT points, state, SUM(points)
      OVER(PARTITION BY state
            ORDER BY points)
      AS average_state_point
  FROM customers;
```

state	points
VA	2273
VA	947
CO	2967
CO	457
TX	3675
TX	3073
FL	1672
FL	205
GA	1486
GA	796

points	state	cumulative_state_point
457	CO	457
2967	CO	3424
205	FL	205
1672	FL	1877
796	GA	796
1486	GA	2282
3073	TX	3073
3675	TX	6748
947	VA	947
2273	VA	3220



AGGREGATE FUNCTIONS

AGGREGATE FUNCTIONS

Name	Description	Name	Description
<u>AVG ()</u>	Return the average value of the argument	<u>MAX ()</u>	Return the maximum value
<u>BIT_AND ()</u>	Return bitwise AND	<u>MIN ()</u>	Return the minimum value
<u>BIT_OR ()</u>	Return bitwise OR	<u>STD ()</u>	Return the population standard deviation
<u>BIT_XOR ()</u>	Return bitwise XOR	<u>STDDEV ()</u>	Return the population standard deviation
<u>COUNT ()</u>	Return a count of the number of rows returned	<u>STDDEV_POP ()</u>	Return the population standard deviation
<u>COUNT(DISTINCT)</u>	Return the count of a number of different values	<u>STDDEV_SAMP ()</u>	Return the sample standard deviation
<u>GROUP_CONCAT ()</u>	Return a concatenated string	<u>SUM ()</u>	Return the sum
<u>JSON_ARRAYAGG ()</u>	Return result set as a single JSON array	<u>VAR_POP ()</u>	Return the population standard variance
<u>JSON_OBJECTAGG ()</u>	Return result set as a single JSON object	<u>VAR_SAMP ()</u>	Return the sample variance
		<u>VARIANCE ()</u>	Return the population standard variance



SQL Queries

GROUP BY- HAVING

GROUP BY- HAVING

1/ Calculate the average points of all customers by state.

```
SELECT state, AVG(points) AS average_state_point  
FROM customers  
GROUP BY state;
```

state	points
VA	2273
VA	947
CO	2967
CO	457
TX	3675
TX	3073
FL	1672
FL	205
GA	1486
GA	796

state	average_state_point
VA	1610.0000
CO	1712.0000
TX	3374.0000
FL	938.5000
GA	1141.0000

NOTE: GROUP BY reduce the number of row into one representing that group

2/ Select the state that have total points of all customers in that state > 3000.

```
SELECT state, SUM(points) AS total_state_point  
FROM customers  
GROUP BY state  
HAVING total_state_point > 3000;
```

state	points
VA	2273
VA	947
CO	2967
CO	457
TX	3675
TX	3073
FL	1672
FL	205
GA	1486
GA	796

state	total_state_point
VA	3220
CO	3424
TX	6748

NOTE: HAVING is WHERE but act on the aggregated group values



SQL Queries

QUIZZES



QUIZZES

Use the learned SQL functions to solve these SQL questions below.

- 1/ Retrieve the first name and last name of customers whose points are greater than 1000.
- 2/ Retrieve the product name and unit price of products with a quantity in stock between 50 and 100.
- 3/ Retrieve the order ID and order date of orders placed between '2018-01-01' and '2018-12-31'.
- 4/ Retrieve the first name and last name of customers whose phone number is null.
- 5/ Retrieve the customer ID and points of customers whose first name starts with 'I'.
- 6/ Retrieve the order ID and order date of the latest 5 orders, ordered by order date in descending order.
- 7/ Retrieve the product name and unit price of the top 3 most expensive products.
- 8/ Retrieve the first name and last name of customers whose birth year is a leap year.
- 9/ Retrieve the customer ID and the count of orders placed by each customer, ordered by the count in descending order.
- 10/ Retrieve the product name and quantity in stock of products whose quantity is a multiple of 10.
- 11/ Retrieve the product name and unit price of products that have a name starting with a vowel (AIEUO).
- 12/ Retrieve the customer ID and the total points for customers whose points are above the average points.
- 13/ Retrieve the order ID and order date of the 5th to 10th orders, ordered by order date.



QUIZZES

Use the learned SQL functions to solve these SQL questions below.

- 14/ Retrieve the customer ID and the total points for customers born in the same month, ordered by total points in descending order.
- 15/ Retrieve the product name and unit price of products whose unit price is a whole number.
- 16/ Retrieve the customer ID and the count of orders placed by each customer, where the count is greater than 3.
- 17/ Retrieve the product name and unit price of products whose unit price is greater than the average unit price.
- 18/ Retrieve the order ID and order date of the oldest order for each customer.
- 19/ Retrieve the order ID and order date of orders placed in the last 7 days.
- 20/ Retrieve the customer ID and points of customers whose first name contains the letter 'a' and last name contains the letter 'b'.
- 21/ Retrieve the customer ID and the count of orders placed by each customer, ordered by the count in ascending order.
- 22/ Retrieve the product name and quantity in stock of products whose quantity is a power of 2.
- 23/ Retrieve the order ID and order date of orders that were placed on weekends (Saturday or Sunday).

