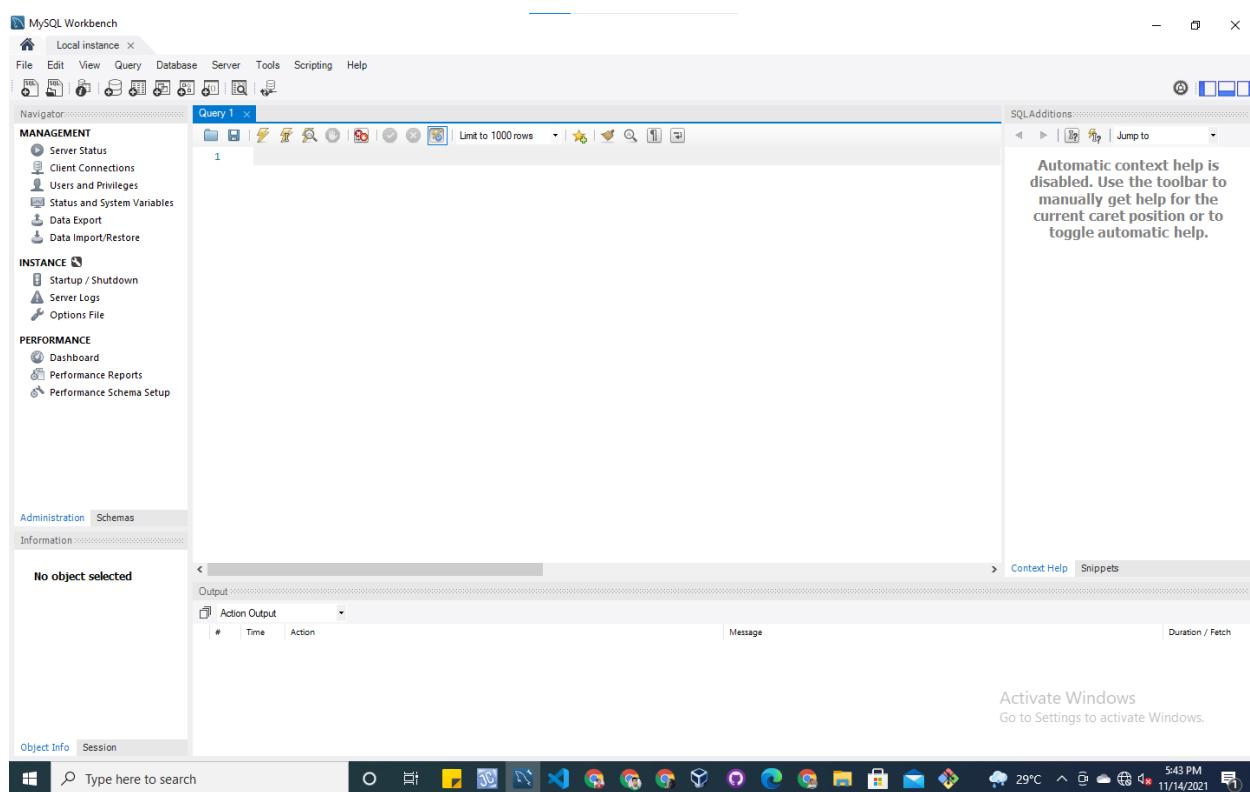
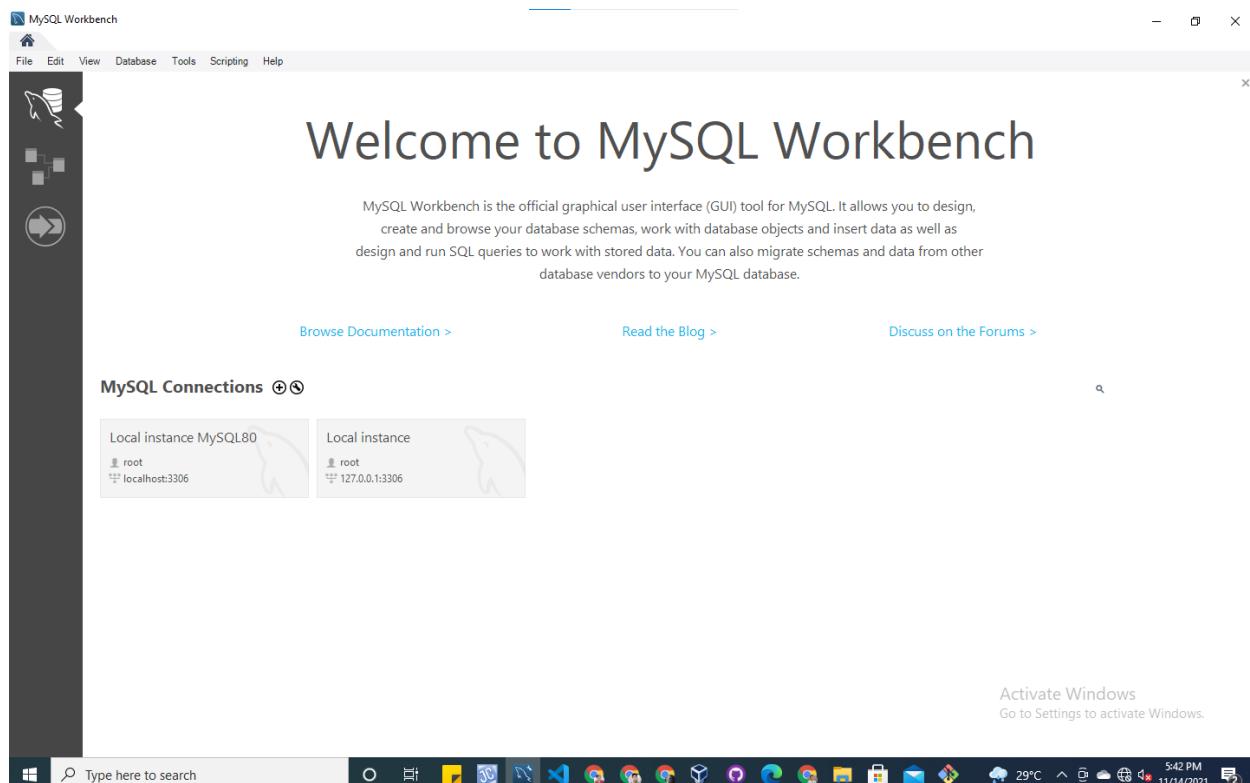
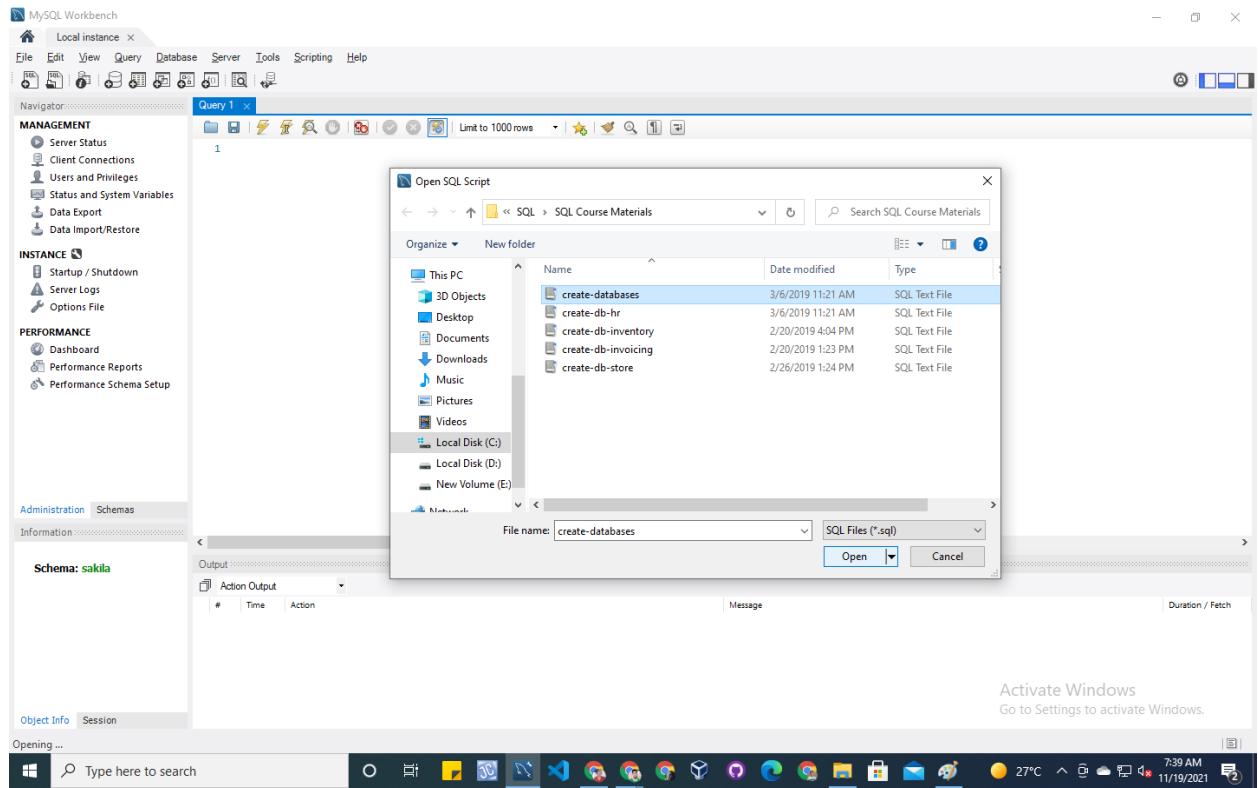


INSTALLING MySQL on Windows

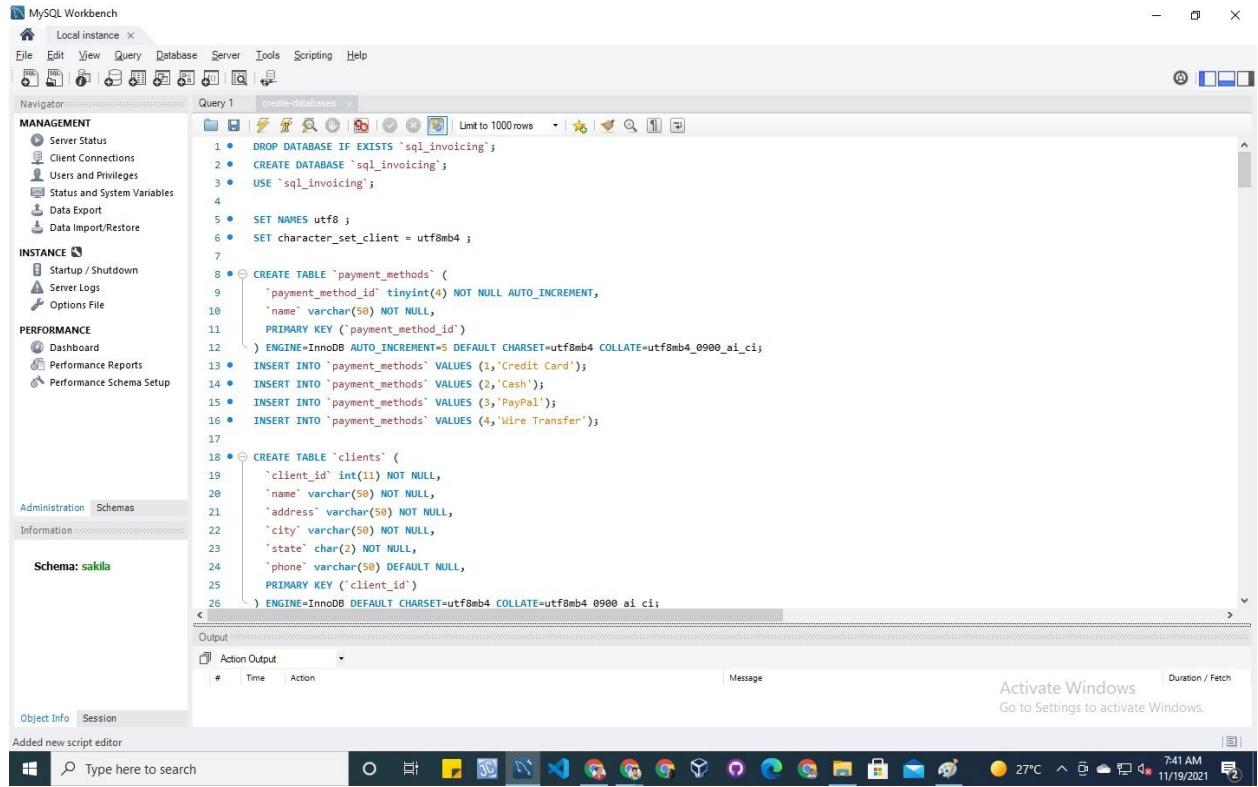


CREATING THE DATABASE

- Let's Open the Main SQL file. That is to create a database.



- So this is an example of SQL Code. Let's Run this SQL code.



- Now, here down on the bottom, we have this panel called the output window that shows all the operations performed in our database server, so we can see all the operations completed successfully, or something went wrong.

MySQL Workbench Local instance

File Edit View Query Database Server Tools Scripting Help

Navigator

Query 1 create databases

```

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

```

Output

#	Time	Action	Message	Duration / Fetch
141	07:45:03	INSERT INTO `employees` VALUES (75900,'Virge','Goodrum','Information Systems Manager',54578,3,1)	1 row(s) affected	0.078 sec
142	07:45:04	INSERT INTO `employees` VALUES (76196,'Mirla','Janowski','Cost Accountant',119241,37270,3)	1 row(s) affected	0.110 sec
143	07:45:04	INSERT INTO `employees` VALUES (80529,'Lynde','Aronson','Junior Executive',77182,37270,4)	1 row(s) affected	0.093 sec
144	07:45:04	INSERT INTO `employees` VALUES (80679,'Mildred','Sokale','Geologist II',67987,37270,4)	1 row(s) affected	0.110 sec
145	07:45:04	INSERT INTO `employees` VALUES (84791,'Hazel','Tabert','General Manager',93760,37270,4)	1 row(s) affected	0.078 sec
146	07:45:04	INSERT INTO `employees` VALUES (95213,'Cole','Kesterton','Pharmacist',86119,37270,4)	1 row(s) affected	0.047 sec
147	07:45:04	INSERT INTO `employees` VALUES (96513,'Theresa','Binney','Food Chemist',47354,37270,5)	1 row(s) affected	0.047 sec
148	07:45:04	INSERT INTO `employees` VALUES (98374,'Estrella','Daleman','Staff Accountant IV',70187,37270,5)	1 row(s) affected	0.156 sec
149	07:45:05	INSERT INTO `employees` VALUES (115357,'Ivy','Feeary','Structural Engineer',92710,37270,5)	1 row(s) affected	0.140 sec
150	07:45:05	DROP DATABASE IF EXISTS `sql_inventory`	0 row(s) affected, 1 warning(s): 1008 Can't drop database `sql_inventory`; database doesn't exist	0.063 sec
151	07:45:05	CREATE DATABASE `sql_inventory`	1 row(s) affected	0.297 sec
152	07:45:05	USE `sql_inventory`	0 row(s) affected	0.000 sec
153	07:45:05	CREATE TABLE `products` (`product_id` int(11) NOT NULL AUTO_INCREMENT, `name` varchar(40) NOT NULL, `unit` varchar(20) NOT NULL, `price` decimal(10,2) NOT NULL, `stock` int(11) NOT NULL, `category` varchar(20) NOT NULL, `description` text NOT NULL, `image` varchar(100) NOT NULL, PRIMARY KEY (`product_id`)) ENGINE=InnoDB DEFAULT CHARSET=utf8;	0 row(s) affected, 2 warning(s): 1681 Integer display width is deprecated and will be removed in a future release. 1681 Integer display width is deprecated and will be removed in a future release.	0.406 sec
154	07:45:06	INSERT INTO `products` VALUES (1,'Foam Dinner Plate','70.1.21')	1 row(s) affected	0.078 sec
155	07:45:06	INSERT INTO `products` VALUES (2,'Pork - Bacon,back Peameal','49.4.65')	1 row(s) affected	0.109 sec
156	07:45:07	INSERT INTO `products` VALUES (3,'Lettuce - Romaine, Heart','38.3.35')	1 row(s) affected	0.047 sec
157	07:45:07	INSERT INTO `products` VALUES (4,'Bucoloni - Gaylor, Chinese','90.4.53')	1 row(s) affected	0.109 sec
158	07:45:07	INSERT INTO `products` VALUES (5,'Sauce - Ranch Dressing','94.1.63')	1 row(s) affected	0.125 sec
159	07:45:07	INSERT INTO `products` VALUES (6,'Pett Baguette','14.2.39')	1 row(s) affected	0.125 sec
160	07:45:07	INSERT INTO `products` VALUES (7,'Sweet Pea Sprouts','98.3.23')	1 row(s) affected	0.125 sec

Activate Windows Go to Settings to activate 0.265 sec

Query Completed

7:45 AM 27°C 11/19/2021

- I explore the sql_store database. I look at all the tables, the data to get an id of the kind of data that is in this database.

MySQL Workbench

Local instance ×

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- sakila
- sql_inventory
- sql_hr
- sql_store
- Tables
- Views
- Stored Procedures
- Functions
- sql_invoicing
- sys
- world

Query 1 customers

```
1 • SELECT * FROM sql_store.customers;
```

	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	MA	2273	
2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Commercial Trail	Hampton	VA	947	
3	Freddi	Boeger	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	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	60 Julian Crossing	Nashville	TN	1672	
8	Thadde	Naseby	1993-07-17	941-527-3977	538 Moines Center	Sarasota	FL	205	
9	Rombola	Rumgay	1992-05-23	559-181-3744	3520 Ohio Trail	Visalia	CA	1466	
10	Levy	Mynett	1969-10-13	404-246-3370	68 Lawn Avenue	Atlanta	GA	795	

Administration Schemas

Information

Table: customers

Columns:

- customer_id int AI PK
- first_name varchar(50)
- last_name varchar(50)
- birth_date date
- phone varchar(50)
- address varchar(50)
- city varchar(50)
- state char(2)
- points int

Object Info Session customers 1 x

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

Activate Windows
Go to Settings to activate Windows.

8:35 AM Apply Revert

MySQL Workbench

Local instance ×

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- sakila
- sql_inventory
- sql_hr
- sql_store
- Tables
- Views
- Stored Procedures
- Functions
- sql_invoicing
- sys
- world

Query 1 customers orders

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

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

Administration Schemas

Information

Table: customers

Columns:

- customer_id int AI PK
- first_name varchar(50)
- last_name varchar(50)
- birth_date date
- phone varchar(50)
- address varchar(50)
- city varchar(50)
- state char(2)
- points int

Object Info Session orders 1 x

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

Activate Windows
Go to Settings to activate Windows.

8:42 AM Apply Revert

- So this was a brief introduction to relational databases. I learned about databases, tables, columns, rows, and relationships.

THE SELECT STATEMENT

- Basics
- SQL is not a case-sensitive language.
- In MySQL, every statement must be terminated with a semicolon.

```
USE sql_store;
SELECT *
FROM customers
WHERE state = 'CA'
ORDER BY first_name
LIMIT 3;
```

- Comments

We use comments to add notes to our code.

— This is a comment and it won't get executed.

SELECT CLAUSE

— Using expressions

```
SELECT (points * 10 + 20) AS discount_factor
FROM customers
```

Order of operations:

- Parenthesis
- Multiplication / division
- Addition / subtraction

— Removing duplicates

```
SELECT DISTINCT state
FROM customers
```

~ EXERCISE 1 ~

-- Return all the products

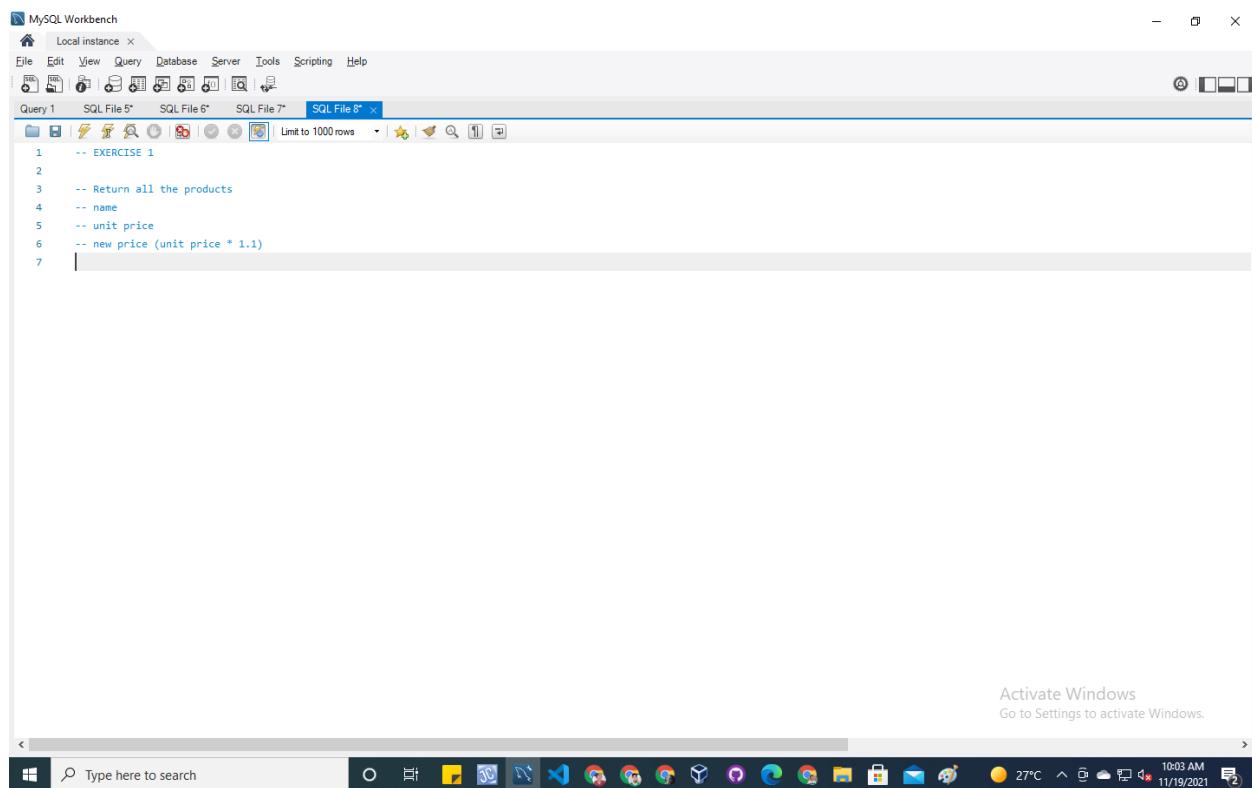
-- name

-- unit price

-- new price (unit price * 1.1)

```
SELECT name, unit_price, unit_price * 1.1 AS new_price
```

```
FROM products
```



The screenshot shows the MySQL Workbench interface. The main window displays the following SQL code:

```
-- EXERCISE 1
-- Return all the products
-- name
-- unit price
-- new price (unit price * 1.1)
```

The code is listed in the Query Editor under Query 1. The interface includes a toolbar with various icons, a menu bar with File, Edit, View, Query, Database, Server, Tools, Scripting, Help, and a tab bar with multiple tabs (Query 1, SQL File 5*, SQL File 6*, SQL File 7*, SQL File 8*). Below the editor is a status bar with the message "Activate Windows Go to Settings to activate Windows." At the bottom of the screen is the Windows taskbar.

The screenshot shows the MySQL Workbench interface. In the top window, a query is being typed:

```

1 -- EXERCISE 1
2
3 -- Return all the products
4 -- name
5 -- unit price
6 -- new price (unit price * 1.1)
7 • SELECT name, unit_price, unit_price * 1.1 AS new_price
8 FROM products

```

The results grid below displays the following data:

	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
	Broccolini - 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

The Windows taskbar at the bottom shows various application icons.

WHERE CLAUSE

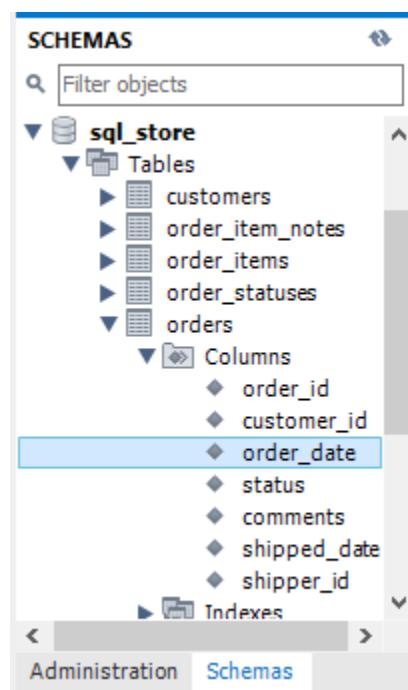
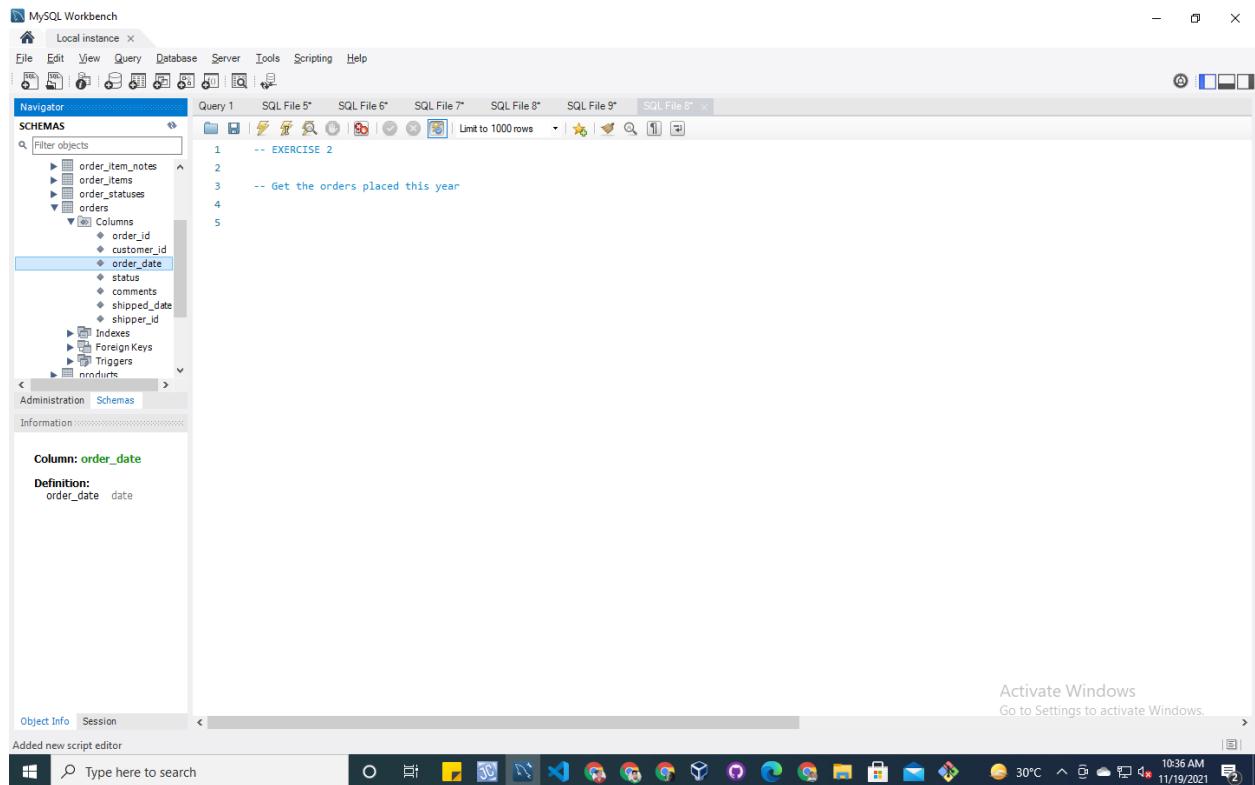
We use the WHERE clause to filter data.

Comparison operators:

- Greater than: >
- Greater than or equal to: >=
- Less than: <
- Less than or equal to: <=
- Equal: =
- Not equal: <>
- Not equal: !=

~ EXERCISE 2 ~

-- Get the orders placed this year



```
SELECT *
FROM orders
WHERE order_date >= '2019-01-01'
```

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
-- EXERCISE 2
-- Get the orders placed this year
SELECT *
FROM orders
WHERE order_date >= '2019-01-01'.
```

The results grid displays one row of data:

order_id	customer_id	order_date	status	comments	shipped_date	shipper_id
1	6	2019-01-30	1	NULL	NULL	NULL

The status bar at the bottom indicates "orders 1 x" and "Query Completed". The taskbar at the bottom right shows the date and time as "10:37 AM 11/9/2021".

LOGICAL OPERATORS

— AND (both conditions must be True)

```
SELECT *
FROM customers
WHERE birthdate > '1990-01-01' AND points > 1000
```

— OR (at least one condition must be True)

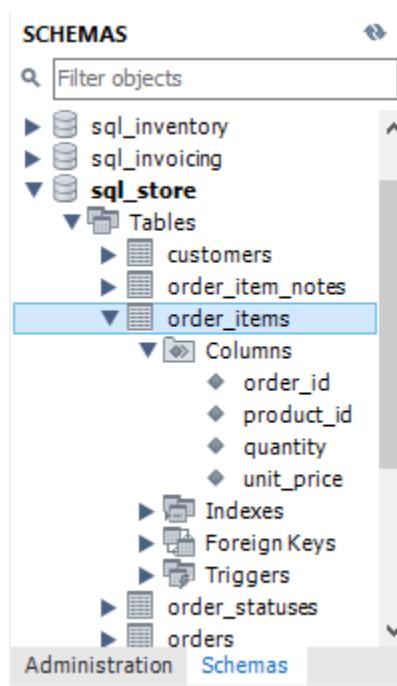
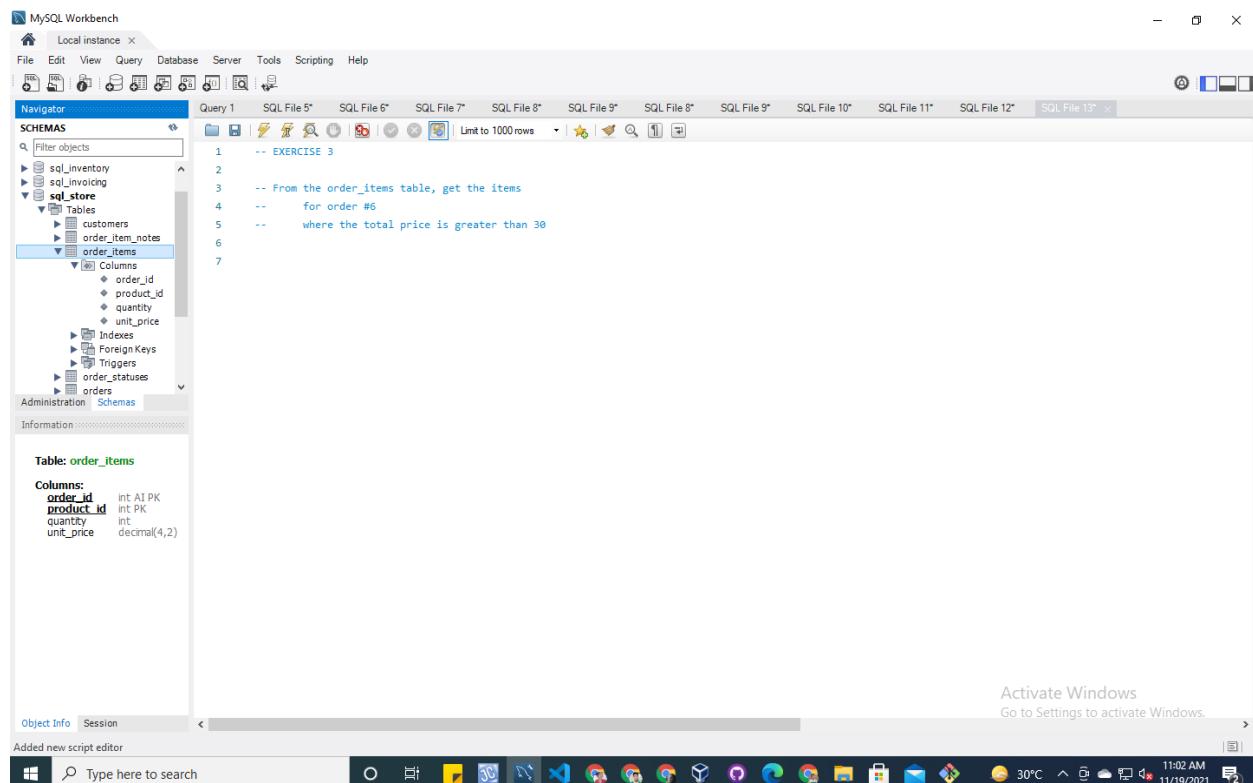
```
SELECT *
FROM customers
WHERE birthdate > '1990-01-01' OR points > 1000
```

— NOT (to negate a condition)

```
SELECT *
FROM customers
WHERE NOT (birthdate > '1990-01-01')
```

~ EXERCISE 3 ~

-- From the order_items table, get the items
-- for order #6
-- where the total price is greater than 30



```

-- EXERCISE 3
-- From the order_items table, get the items
-- for order #6
-- where the total price is greater than 30
SELECT *
FROM order_items
WHERE order_id = 6 AND unit_price * quantity > 30

```

order_id	product_id	quantity	unit_price
6	1	4	8.65

IN OPERATOR

— Returns customers in any of these states: VA, NY, CA

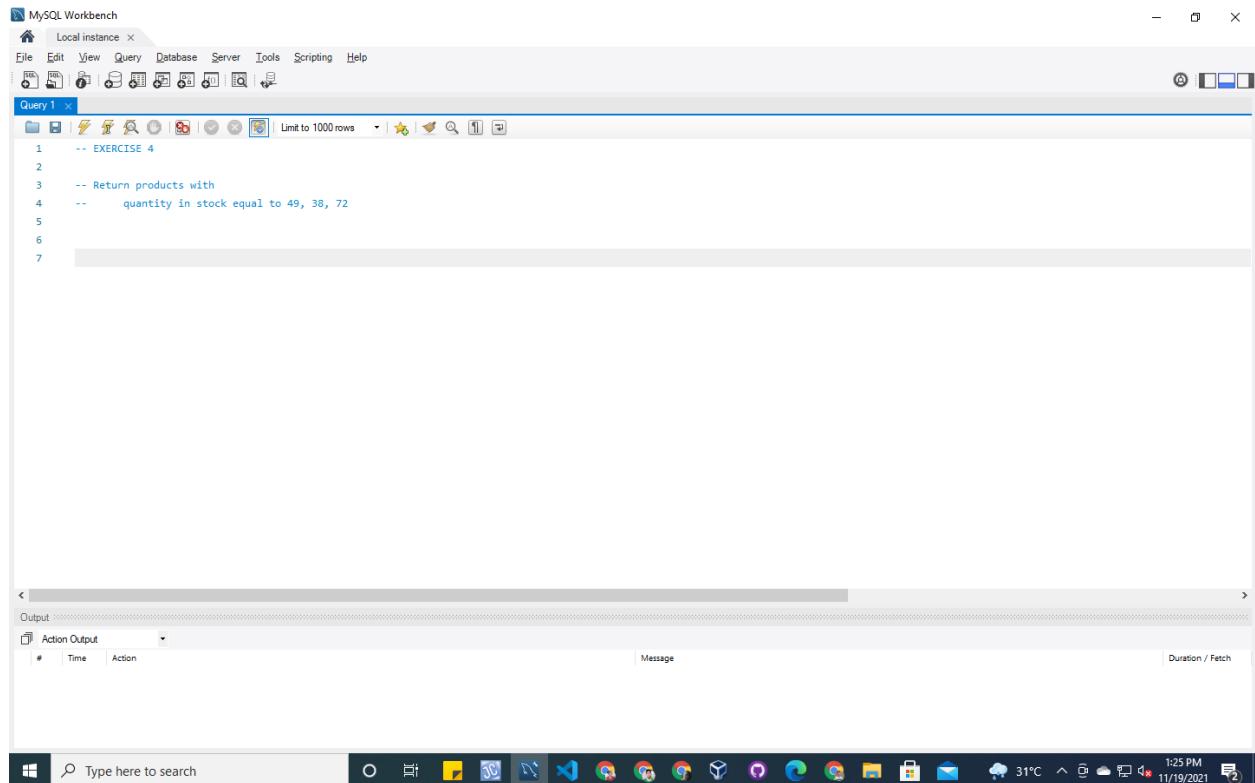
```

SELECT *
FROM customers
WHERE state IN ('VA', 'NY', 'CA')

```

~ EXERCISE 4 ~

-- Return products with
-- quantity in stock equal to 49, 38, 72



The screenshot shows the MySQL Workbench interface. The Query Editor window contains the following SQL code:

```
1 -- EXERCISE 4
2
3 -- Return products with
4 --     quantity in stock equal to 49, 38, 72
5
6
7
```

The Taskbar at the bottom of the screen includes the Start button, a search bar, and various pinned application icons such as File Explorer, Edge, and File History.

```

MySQL Workbench
Local instance X
File Edit View Query Database Server Tools Scripting Help
Query 1
1 -- EXERCISE 4
2
3 -- Return products with
4 --     quantity in stock equal to 49, 38, 72
5
6
7 • SELECT *
8 FROM products
9 WHERE quantity_in_stock IN (49, 38, 72)
10

```

product_id	name	quantity_in_stock	unit_price
2	Pork - Bacon,back Peameal	49	4.65
3	Lettuce - Romaine, Heart	38	3.35
•		NULL	NULL

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor | Field Types | Query Stats | Execution Plan | Apply | Revert | Type here to search | 1:29 PM | 31°C | 11/19/2021 |

BETWEEN OPERATOR

```

SELECT *
FROM customers
WHERE points BETWEEN 1000 AND 3000

```

-- WHERE points >= 1000 AND points <= 3000

The screenshot shows the MySQL Workbench interface. In the top left, the Navigator pane displays available databases: sakila, sql_hr, sql_inventory, sql_invoicing, sql_store, sys, and world. The Query 1 pane contains the following SQL code:

```
1 • SELECT *
2   FROM customers
3 WHERE points BETWEEN 1000 AND 3000
```

The Result Grid pane displays the results of the query, showing customer information for those with points between 1000 and 3000. The columns are customer_id, first_name, last_name, birth_date, phone, address, city, state, and points. The data includes:

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	MA	2273
3	Freddi	Boagey	1985-02-07	719-724-7869	251 Springs Junction	Colorado Springs	CO	2967
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

The bottom status bar shows the system tray with icons for network, battery, and date/time.

~ EXERCISE 5 ~

-- Return customers born

-- between 1/1/1990 and 1/1/2000

MySQL Workbench

Local instance X

File Edit View Query Database Server Tools Scripting Help

SQL File 9* SQL File 8* SQL File 9* SQL File 10* SQL File 11* SQL File 12* SQL File 13* SQL File 14* SQL File 15* SQL File 16* SQL File 17* SQL File 19* SQL File 20* SQL File 21* SQL File 22* X

Limit to 1000 rows

```
1 --> Return customers born
2 -- between 1/1/1990 and 1/1/2000
```

Activate Windows
Go to Settings to activate Windows.

Added new script editor

Type here to search

Windows Taskbar: 11:51 AM 11/19/2021

MySQL Workbench

Local instance X

File Edit View Query Database Server Tools Scripting Help

SQL File 9* SQL File 8* SQL File 9* SQL File 10* SQL File 11* SQL File 12* SQL File 13* SQL File 14* SQL File 15* SQL File 16* SQL File 17* SQL File 19* SQL File 20* SQL File 21* SQL File 22* X

Limit to 1000 rows

```
1 --> Return customers born
2 -- between 1/1/1990 and 1/1/2000
3
4 • SELECT *
5   FROM customers
6 WHERE birth_date BETWEEN '1990-01-01' AND '2000-01-01'
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

customer_id	first_name	last_name	birth_date	phone	address	city	state	points
6	Elka	Twidell	1991-09-04	312-480-8498	7Marley Drive	Chicago	IL	3073
8	Thacher	Naseby	1993-07-17	941-527-3977	538 Mosinee Center	Sarasota	FL	205
9	Romola	Rungay	1992-05-23	559-181-3744	3520 Ohio Trail	Visalia	CA	1486
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result Grid Form Editor Field Types Query Stats Execution Plan

Activate Windows
Go to Settings to activate Windows.

customers 1 X

Query Completed

Type here to search

Windows Taskbar: 11:53 AM 11/19/2021

LIKE OPERATOR

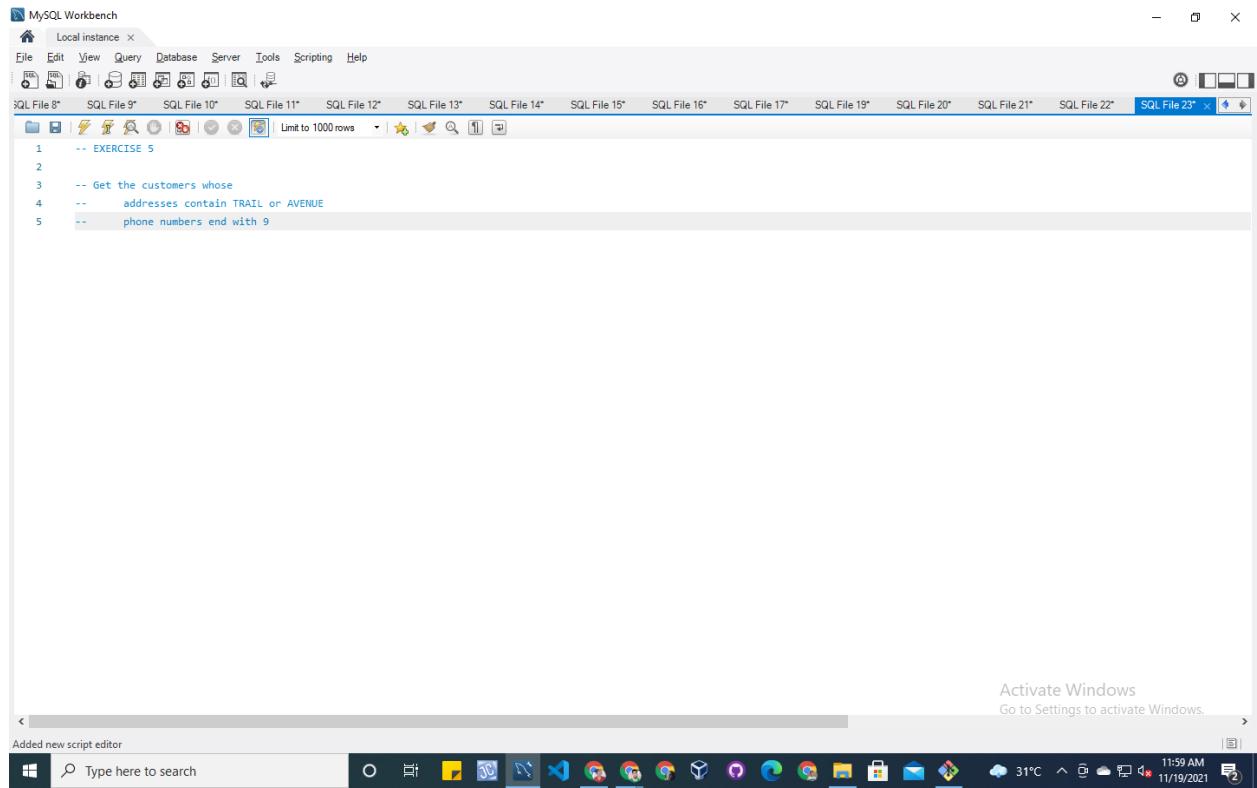
-- Returns customers whose first name starts with b

```
SELECT *
FROM customers
WHERE first_name
LIKE 'b%'
```

- %: any number of characters
- _: exactly one character

~ EXERCISE 6 ~

-- Get the customers whose
-- addresses contain TRAIL or AVENUE
-- phone numbers end with 9



-- addresses contain TRAIL or AVENUE

The screenshot shows the MySQL Workbench interface with a query editor and results grid. The query is:

```
4 -- addresses contain TRAIL or AVENUE
5 -- phone numbers end with 9
6
7 • SELECT *
8 FROM customers
9 WHERE address LIKE '%trail%' OR
10    address LIKE '%avenue%'
```

The results grid shows customer data where the address contains either 'TRAIL' or 'AVENUE'. The highlighted row is for customer ID 9, Romola Rungay, with the address '3520 Ohio Trail'.

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

The Windows taskbar at the bottom shows the date as 11/19/2021 and the time as 12:01 PM.

-- phone numbers end with 9

The screenshot shows the MySQL Workbench interface with a query editor and results grid. The query is:

```
1 -- EXERCISE 5
2
3 -- Get the customers whose
4 -- addresses contain TRAIL or AVENUE
5
6 -- phone numbers end with 9
7
8 • SELECT *
9 FROM customers
10 WHERE phone LIKE '%9'
11
```

The results grid shows customer data where the phone number ends with '9'. The highlighted row is for customer ID 7, Ilene Doyson, with the phone number '615-641-4759'.

customer_id	first_name	last_name	birth_date	phone	address	city	state	points
3	Freddi	Boagey	1985-02-07	719-724-7869	251 Springs Junction	Colorado Springs	CO	2967
7	Ilene	Doyson	1964-08-30	615-641-4759	50 Lillian Crossing	Nashville	TN	1672

The Windows taskbar at the bottom shows the date as 11/19/2021 and the time as 12:03 PM.

-- phone numbers not end with 9

The screenshot shows the MySQL Workbench interface. In the top-left pane, there is a code editor window containing the following SQL query:

```
-- EXERCISE 5
-- Get the customers whose
-- addresses contain TRAIL or AVENUE
-- phone numbers end with 9
SELECT *
FROM customers
WHERE phone NOT LIKE '%9'
```

The results of the query are displayed in a grid below. The columns are labeled: customer_id, first_name, last_name, birth_date, phone, address, city, state, and points. The data includes 10 rows of customer information, with the phone number column showing the value '312-400-8998' for the 6th row.

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
4	Amber	Roseburgh	1974-04-14	407-231-8017	30 Arapahoe Terrace	Orlando	FL	457
6	Elka	Twidell	1991-09-04	312-400-8998	7 Manley Drive	Chicago	IL	3073
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
•	HOLLY	HOLLY	HOLLY	HOLLY	HOLLY	HOLLY	HOLLY	HOLLY

At the bottom of the interface, there is a taskbar with various application icons and system status indicators like temperature and date.

REGEXP OPERATOR

— Returns customers whose first name starts with a

```
SELECT *
FROM customers
WHERE first_name
REGEXP '^a'
```

- ^ : beginning of a string
- \$: end of a string
- | : logical OR
- [abcd] : match any single characters
- [a-d] : any characters from a to d

~ EXERCISE 7 ~

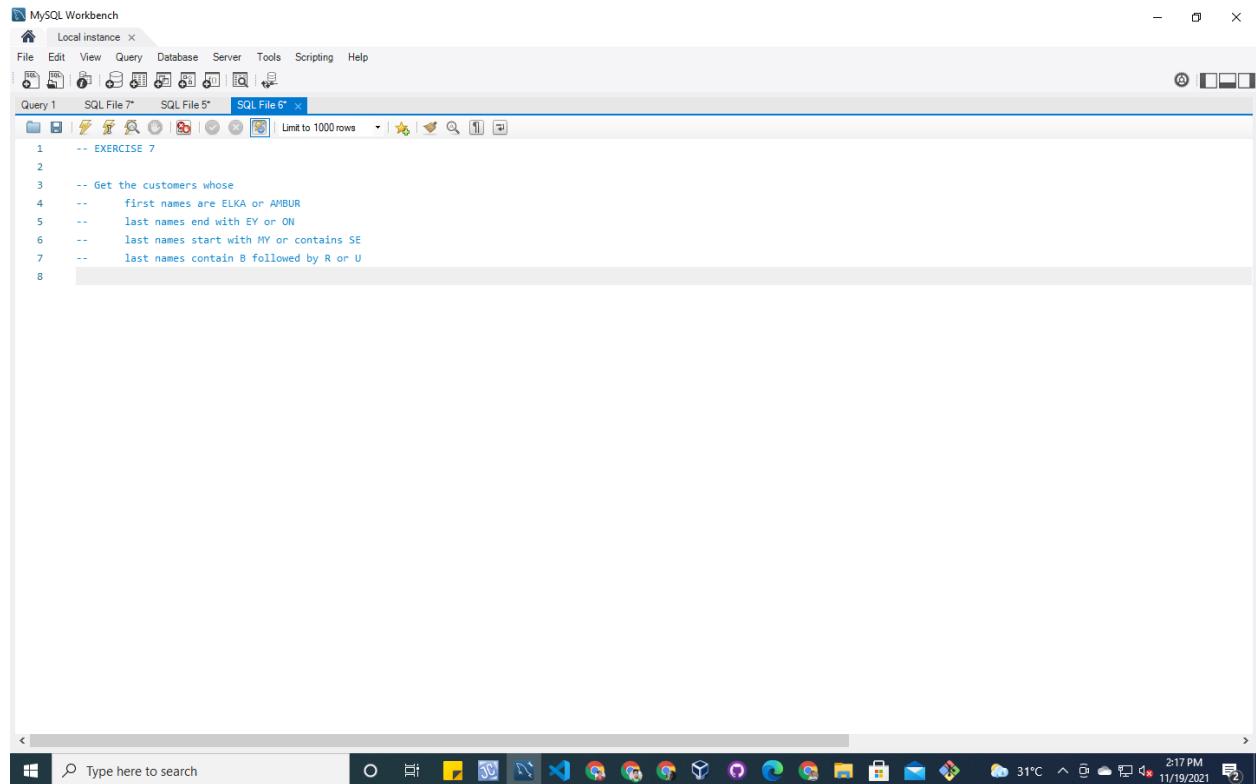
-- Get the customers whose

-- first names are ELKA or AMBUR

-- last names end with EY or ON

-- last names start with MY or contains SE

-- last names contain B followed by R or U



The screenshot shows the MySQL Workbench interface. The title bar says "MySQL Workbench Local instance". The menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, Help. The toolbar has various icons for database management. Below the toolbar, there are three tabs: "Query 1", "SQL File 7", and "SQL File 6" (which is selected). The main area contains the following SQL code:

```
1 -- EXERCISE 7
2
3 -- Get the customers whose
4 --   first names are ELKA or AMBUR
5 --   last names end with EY or ON
6 --   last names start with MY or contains SE
7 --   last names contain B followed by R or U
8
```

-- first names are ELKA or AMBUR

MySQL Workbench - Local instance

Query 1 SQL File 7* SQL File 5* SQL File 6*

```
1 -- EXERCISE 7
2
3 -- Get the customers whose
4 --   first names are ELKA or AMBUR
5
6 • SELECT *
7   FROM customers
8   WHERE first_name REGEXP 'elka|ambur'
9
10 --   last names end with EY or ON
11 --   last names start with MY or contains SE
12 --   last names contain B followed by R or U
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

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

customers 1

Type here to search

2:19 PM 11/19/2021

-- last names end with EY or ON

MySQL Workbench - Local instance

Query 1 SQL File 7* SQL File 5* SQL File 6*

```
1 -- EXERCISE 7
2
3 -- Get the customers whose
4 --   first names are ELKA or AMBUR
5 --   last names end with EY or ON
6
7 • SELECT *
8   FROM customers
9   WHERE last_name REGEXP 'ey$|on$'
10
11 --   last names start with MY or contains SE
12 --   last names contain B followed by R or U
13
14
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

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
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
■	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

customers 6

Type here to search

2:28 PM 11/19/2021

Last names start with MY or contains SE

The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. Below the menu is a toolbar with various icons for file operations like Open, Save, Print, and Database management. The main area has tabs for Query 1, SQL File 7*, SQL File 5*, and SQL File 6*. The SQL File 6* tab is active, displaying the following SQL code:

```
1 -- EXERCISE 7
2
3 -- Get the customers whose
4 --     first names are ELKA or AMBUR
5 --     last names end with EY or ON
6 --     last names start with MY or contains SE
7
8 ● SELECT *
9 FROM customers
10 WHERE last_name REGEXP '^my|se'
11
12 --     last names contain B followed by R or U
13
14
```

Below the code is a Result Grid table with columns: customer_id, first_name, last_name, birth_date, phone, address, city, state, and points. The data is as follows:

	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	Thadher	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

The bottom status bar shows "customers 3" and "Apply". On the right side, there is a vertical panel with tabs for Result Grid, Form Editor, Field Types, Query Stats, Execution Plan, and a search bar.

Last names contain B followed by R or U

IS NULL OPERATOR

-- Returns customers who don't have a phone number

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

~ EXERCISE 8 ~

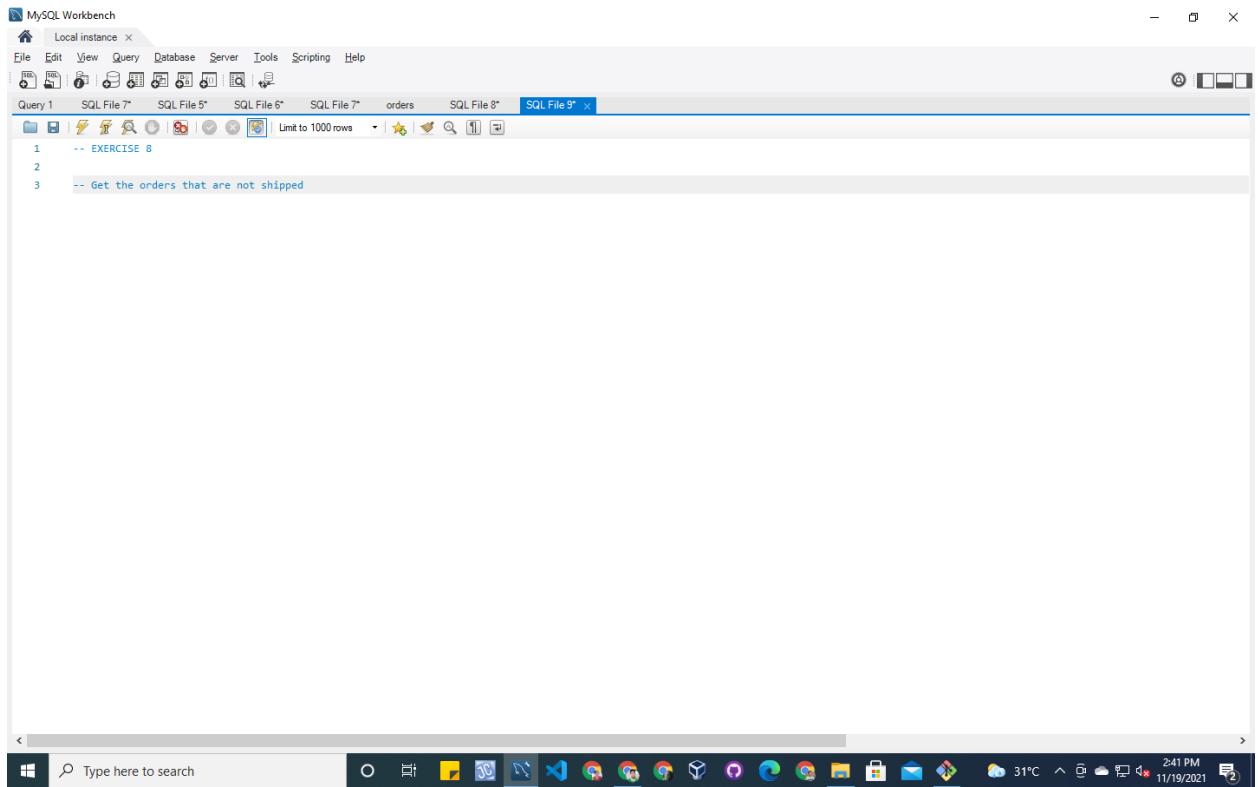
-- Get the orders that are not shipped

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

- Navigator:** Shows the schema structure with the 'orders' table selected. Under 'Columns', it lists: order_id, customer_id, order_date, status, comments, shipped_date, and shipper_id.
- Query Editor:** Displays the SQL query: `SELECT * FROM sql_store.orders;`
- Result Grid:** Shows the results of the query. The table has columns: order_id, customer_id, order_date, status, comments, shipped_date, and shipper_id. The data includes rows 1 through 10, with some comments containing placeholder text like "Aliquam erat volutpat. In congue." and "Nulla mollis molestie lorem. Quisque ut erat."

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	NULL	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

The interface also includes a toolbar, a status bar at the bottom, and various navigation and search tools on the right.



The screenshot shows the MySQL Workbench interface. The title bar says "MySQL Workbench" and "Local instance". The menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, Help. The toolbar has icons for New, Open, Save, Print, Copy, Paste, Find, Replace, and Execute. The query editor has tabs for "Query 1", "SQL File 7*", "SQL File 5*", "SQL File 6*", "SQL File 7*", "orders", and "SQL File 8*". The current tab is "SQL File 8*". The code in the editor is:

```
1 -- EXERCISE 8
2
3 -- Get the orders that are not shipped
4
5 • SELECT *
6 FROM orders
7 WHERE shipper_id IS NULL
```

Below the editor is a "Result Grid" window showing the results of the query. The table has columns: order_id, customer_id, order_date, status, comments, shipped_date, and shipper_id. The data is:

order_id	customer_id	order_date	status	comments	shipped_date	shipper_id
1	6	2019-01-30	1	HULL	NULL	NULL
3	8	2017-12-01	1	HULL	NULL	NULL
4	2	2017-01-22	1	HULL	NULL	NULL
6	10	2018-11-18	1	Aliquam erat volutpat. In congue.	NULL	NULL
8	5	2018-06-08	1	Mauris enim leo, rhoncus sed, vestibulum sit am...	NULL	NULL
•	NULL	NULL	NULL	NULL	NULL	NULL

The "Result Grid" window also has tabs for "Form Editor", "Field Types", "Query Stats", and "Execution Plan". The system tray at the bottom shows the date and time as 241 PM, 11/19/2021, and the temperature as 31°C.

ORDER BY CLAUSE

— Sort customers by state (in ascending order), and then
— by their first name (in descending order)

```
SELECT *
FROM customers
ORDER BY state, first_name DESC
```

~ EXERCISE 9 ~

-- In this database, we have this table
-- called order items, where we find
-- the items for each order.
-- Now, for the items for the order with ID 2, and we have sorted
-- these items based on the total price
-- for each item.
-- So, the total price for each item equals
-- quantity times unit price.
-- In this case the total price of product 1 is just 18 dollars.
-- Write a query to select all the items for order with ID2.

-- And sort them by their total price in descending order.

The screenshot shows the MySQL Workbench interface. The left pane displays the Navigator with the schema 'sql_store' selected, showing tables like customers, order_items, and products. The central pane contains a query editor window titled 'Query 1' with the following SQL code:

```
-- EXERCISE 9
-- In this database, we have this table
-- called order items, where we find
-- the items for each order.
-- Now, for the items for the order with ID 2, and we have sorted
-- these items based on the total price
-- for each item.
-- So, the total price for each item equals
-- quantity times unit price.
-- In this case the total price of product 1 is just 18 dollars.
-- Write a query to select all the items for order with ID2.
-- And sort them by their total price in descending order.
```

The bottom pane shows the results of the query in a 'Result Grid' table:

order_id	product_id	quantity	unit_price
2	1	2	9.10
2	4	4	1.66
2	6	2	2.94
NULL	NULL	NULL	NULL

MySQL Workbench Local instance X

File Edit View Query Database Server Tools Scripting Help

Navigator Query 1 SQL File 12* x order_items

SCHEMAS

- sql_hr
- sql_inventory
- sql_invoicing
- sql_store
 - Tables
 - customers
 - order_item_notes
 - order_items
 - order_statuses
 - orders
 - products
 - shippers
 - Views
 - Stored Procedures
 - Functions
- sys
- world

Administration Schemas Information

Table: order_items

Columns:

order_id	int AI PK		
product_id	int PK		
quantity	int		
unit_price	decimal(4,2)		
2	1	2	9.10
2	4	4	1.66
2	6	2	2.94

Result Grid Filter Rows: Export: Wrap Cell Content: □

```
6
7 -- Now, for the items for the order with ID 2, and we have sorted
8 --      these items based on the total price
9 --      for each item.
10
11 -- So, the total price for each item equals
12 --      quantity times unit price.
13
14 -- In this case the total price of product 1 is just 18 dollars.
15
16 -- Write a query to select all the items for order with ID2.
17
18 -- And sort them by their total price in descending order.
19
20 • SELECT *, quantity * unit_price AS total_price
21   FROM order_items
22   WHERE order_id = 2
23   ORDER BY total_price DESC
```

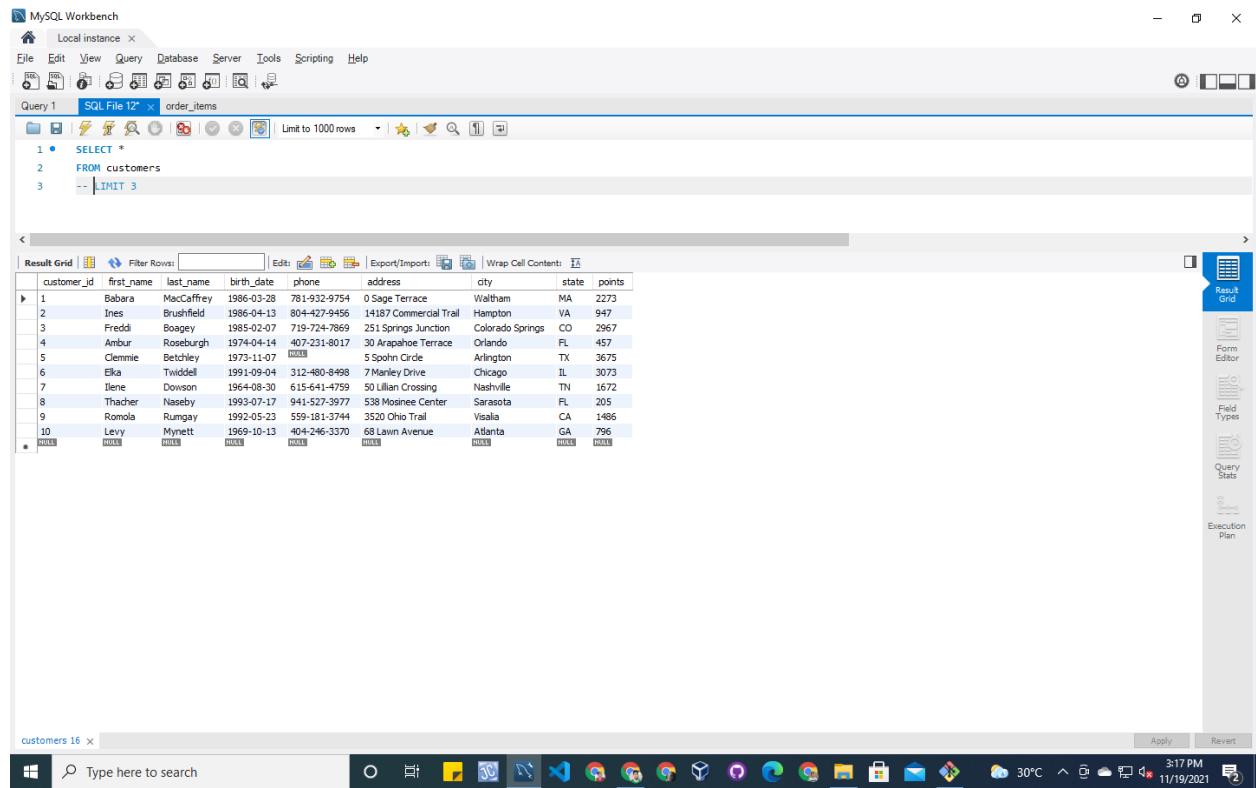
Result 11 x Read Only

Type here to search

The screenshot shows the MySQL Workbench interface. The top menu includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The Navigator pane shows the schema structure with the 'order_items' table selected. The main area contains a query editor with a script and a results grid. The script is a SQL query to select items from the 'order_items' table where 'order_id' is 2, ordered by 'total_price' in descending order. The results grid displays three rows of data: (order_id: 2, product_id: 1, quantity: 2, unit_price: 9.10, total_price: 18.20), (order_id: 2, product_id: 4, quantity: 4, unit_price: 1.66, total_price: 6.64), and (order_id: 2, product_id: 6, quantity: 2, unit_price: 2.94, total_price: 5.88). The status bar at the bottom shows the date and time (11/19/2021, 3:08 PM) and system information (30°C).

LIMIT CLAUSE

SELECT *
FROM customers



The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query editor contains the following SQL code:

```
1 • SELECT *
2 FROM customers
3 -| LIMIT 3
```

The result grid displays the first three rows of the customers table:

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	MA	2273
2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Commercial Trail	Hampton	VA	947
3	Freddi	Boagley	1985-02-07	719-724-7869	251 Springs Junction	Colorado Springs	CO	2967
4	Amber	Roseburgh	1974-04-14	407-231-8017	30 Arapahoe Terrace	Orlando	FL	457
5	Clemmie	Bethley	1973-11-07	NULL	5 Spohn Circle	Arlington	TX	3675
6	Erika	Tiddell	1991-09-04	312-480-8498	7 Manley Drive	Chicago	IL	3073
7	Irene	Dowson	1964-08-30	615-641-4759	50 Lillian Crossing	Nashville	TN	1672
8	Thacher	Naseby	1993-07-17	941-521-3977	538 Mosinee Center	Sarasota	FL	205
9	Romola	Rungay	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
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

— Return only 3 customers

SELECT *
FROM customers
LIMIT 3

The screenshot shows the MySQL Workbench interface. In the top-left corner, there's a title bar with the application name and a 'Local instance' tab. Below it is a menu bar with File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. A toolbar follows with various icons for file operations like Open, Save, Print, and Database management. The main area contains a 'Query 1' tab titled 'SQL File 12*' which displays the following SQL code:

```
1 • SELECT *
2 FROM customers
3 LIMIT 3
```

Below the code is a results grid titled 'Result Grid'. It has columns for customer_id, first_name, last_name, birth_date, phone, address, city, state, and points. The data shows three rows of customer information:

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	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	PO Box 251	WA	2967
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

On the right side of the results grid, there's a vertical toolbar with icons for Result Grid, Form Editor, Field Types, Query Stats, and Execution Plan. At the bottom of the screen, a Windows taskbar is visible with the Start button, a search bar, and icons for various applications like File Explorer, Microsoft Edge, and others. The system tray shows the date and time as 3:22 PM on 11/19/2021.

— Skip 6 customers and return 3

```
SELECT *
FROM customers
LIMIT 6, 3
```

The screenshot shows the MySQL Workbench interface. In the top-left pane, a query editor window titled 'SQL File 12*' displays the following SQL code:

```
1 • SELECT *
2   FROM customers
3   LIMIT 6, 3
4   -- page 1: 1 - 3
5   -- page 2: 4 - 6
6   -- page 3: 7 - 9
```

In the bottom-right pane, a 'Result Grid' window titled 'customers 18 x' shows the results of the query. The grid has columns: customer_id, first_name, last_name, birth_date, phone, address, city, state, and points. The data is as follows:

customer_id	first_name	last_name	birth_date	phone	address	city	state	points
7	Irene	Dowson	1964-08-30	615-641-4759	50 Lillian Crossing	Nashville	TN	1672
8	Thasher	Naseby	1993-07-17	941-527-3977	538 Moine Center	Sarasota	FL	205
9	Ronola	Rungay	1992-05-23	559-181-3744	3520 Ohio Trail	Visalia	CA	1486

The Windows taskbar at the bottom of the screen shows various application icons.

~ EXERCISE 10 ~

-- Get the top three loyal customers

The image shows a screenshot of the MySQL Workbench application. At the top, there's a menu bar with options like File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. Below the menu is a toolbar with various icons for file operations like Open, Save, Print, and Database management. The main area is a query editor with three tabs: "Query 1" (which is currently active), "SQL File 13*", and "SQL File 12*". The "Query 1" tab contains the following SQL code:

```
1 -- EXERCISE 10
2
3 -- Get the top three loyal customers
```

A screenshot of the MySQL Workbench application. The title bar shows 'MySQL Workbench' and 'Local instance'. The menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. Below the menu is a toolbar with icons for file operations like Open, Save, Print, and a search icon. The main area is a query editor with two tabs: 'Query 1' and 'SQL File 12'. The current tab, 'Query 1', contains the following SQL code:

```
1 -- EXERCISE 10
2
3 -- Get the top three loyal customers
4
5 • SELECT *
6   FROM customers
7   ORDER BY points DESC
```

The code is numbered 1 through 7. The fifth line has a blue dot next to it, indicating it is the active line.

The screenshot shows the MySQL Workbench interface. In the top-left pane, there is a code editor window titled 'Query 1 - SQL File 13*' containing the following SQL code:

```

1 -- EXERCISE 10
2
3 -- Get the top three loyal customers
4
5 • SELECT *
6   FROM customers
7   ORDER BY points DESC
8   LIMIT 3

```

In the bottom-right pane, there is a results grid titled 'Result Grid' showing the output of the query. The grid has columns: customer_id, first_name, last_name, birth_date, phone, address, city, state, and points. The data is as follows:

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	Twidell	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
•	NULL	NULL	NULL	NULL	NULL	NULL	CO	NULL

INNER JOINS

```

SELECT *
FROM customers c
JOIN orders o
  ON c.customer_id = o.customer_id

```

~ EXERCISE 11 ~

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- sql_hr
- sql_inventory
- sql_invoicing
- sql_store
 - Tables
 - customers
 - order_item_notes
 - order_items
 - order_statuses
 - orders
 - products
 - shippers
 - Views
 - Stored Procedures
 - Functions
- sys
- world

Administration Schemas Information

Table: **order_items**

Columns:

order_id	int AI PK
product_id	int DK
quantity	int
unit_price	decimal(4,2)

Query 1 SQL File 13* SQL File 12* order_items

```

1 -- EXERCISE 11
2
3 -- Write a query and join this table
4 -- with the products table so for
5 -- each order return both the product
6 -- id as well as this name,
7 -- followed by the quantity,
8 -- and the unit price
9 -- from the order items table.
10
11 -- Use an alias to simplyfy you code

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

order_id	product_id	quantity	unit_price
1	4	4	3.74
2	1	2	9.10
2	4	4	1.66
2	6	2	2.94
3	3	10	9.12
4	3	7	6.99
4	10	7	6.40
5	2	3	9.89
6	1	4	8.65
6	2	4	3.28
6	3	4	7.46
6	5	1	3.45
7	3	7	9.17
8	5	2	6.94
8	8	2	8.59
9	6	5	7.28

Object Info Session order_items 1 x Apply Revert

Type here to search 3:58 PM 11/19/2021

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- sql_hr
- sql_inventory
- sql_invoicing
- sql_store
 - Tables
 - customers
 - order_item_notes
 - order_items
 - order_statuses
 - orders
 - products
 - shippers
 - Views
 - Stored Procedures
 - Functions
- sys
- world

Administration Schemas Information

Table: **order_items**

Columns:

order_id	int AI PK
product_id	int PK
quantity	int
unit_price	decimal(4,2)

Query 1 SQL File 13* SQL File 12* order_items

```

1 • SELECT *
2 FROM order_items oi
3 JOIN products p ON oi.product_id = p.product_id

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

order_id	product_id	quantity	unit_price	product_id	name	quantity_in_stock	unit_price
2	1	2	9.10	1	Foam Dinner Plate	70	1.21
6	1	4	8.65	1	Foam Dinner Plate	70	1.21
10	1	10	6.01	1	Foam Dinner Plate	70	1.21
5	2	3	9.89	2	Pork - Bacon,back Peameal	49	4.65
6	2	4	3.28	2	Pork - Bacon,back Peameal	49	4.65
3	3	10	9.12	3	Lettuce - Romaine, Heart	38	3.35
4	3	7	6.99	3	Lettuce - Romaine, Heart	38	3.35
6	3	4	7.46	3	Lettuce - Romaine, Heart	38	3.35
7	3	7	9.17	3	Lettuce - Romaine, Heart	38	3.35
1	4	4	3.74	4	Brocolini - Gaylan, Chinese	90	4.53
2	4	4	1.66	4	Brocolini - Gaylan, Chinese	90	4.53
6	5	1	3.45	5	Sauce - Ranch Dressing	94	1.63
8	5	2	6.94	5	Sauce - Ranch Dressing	94	1.63
2	6	2	2.94	6	Petit Baguette	14	2.39
9	6	5	7.28	6	Petit Baguette	14	2.39
8	8	2	8.59	8	Island Oasis - Raspberry	26	0.74
10	9	9	4.28	9	Longan	67	2.26
4	10	7	6.40	10	Broom - Push	6	1.09

Object Info Session Result 27 x Read Only

Type here to search 3:58 PM 11/19/2021

MySQL Workbench - Local instance

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- Filter objects
- sql_hr
- sql_inventory
- sql_invoicing
- sql_store**
 - Tables
 - customers
 - order_item_notes
 - order_items
 - order_statuses
 - orders
 - products
 - shippers
 - Views
 - Stored Procedures
- sys
- world

Administration Schemas

Information

Table: order_items

Columns:

order_id	int AI PK		
product_id	int PK		
quantity	int		
unit_price	decimal(4,2)		
2	1	2	9.10
6	1	4	8.65
10	1	10	6.01
5	2	3	9.89
6	2	4	3.28
3	3	10	9.12
4	3	7	6.99
6	3	4	7.46
7	3	7	9.17
1	4	4	3.74
2	4	4	1.66
6	5	1	3.45
8	5	2	6.94
2	6	2	2.94
9	6	5	7.28
8	8	2	8.59
10	9	9	4.28
4	10	7	6.40

Result Grid

Filter Rows:

Export: Wrap Cell Content:

Result 28 x

Object Info Session

Read Only

Type here to search

Windows Taskbar: 403 PM 11/19/2021

JOINING ACROSS DATABASES

USE sql_inventory;

```
SELECT *
FROM sql_store.order_items oi
JOIN sql_inventory.products p
  ON oi.product_id = p.product_id
```

SELF JOINS

USE sql_hr;

```
SELECT
    e.employee_id,
    e.first_name,
    m.first_name AS manager
FROM employees e
JOIN employees m
    ON e.reports_to = m.employee_id
```

JOINING MULTIPLE TABLES

USE sql_store;

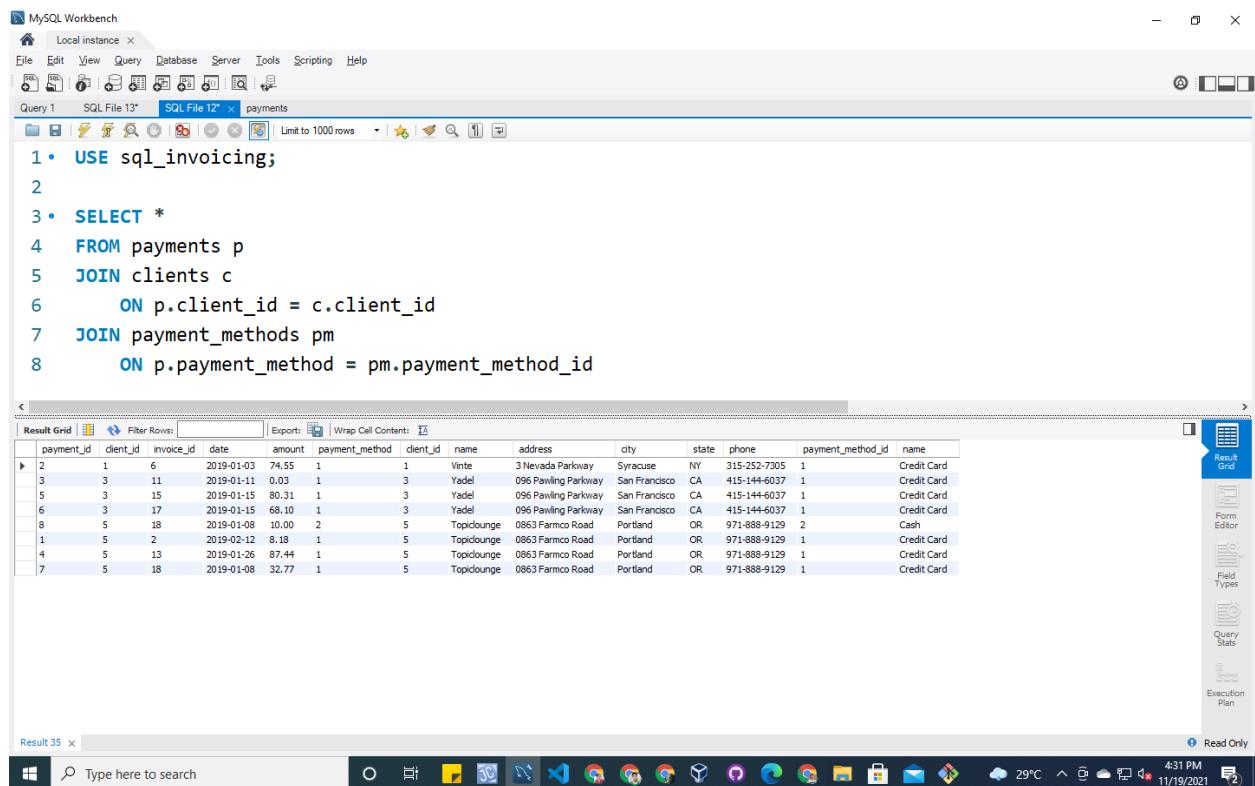
```
SELECT
    o.order_id,
    o.order_date,
    c.first_name,
    c.last_name,
    os.name AS status
FROM orders o
JOIN customers c
    ON o.customer_id = c.customer_id
JOIN order_statuses os
    ON o.status = os.order_status_id
```

~ EXERCISE 12 ~

- Write a query and join this table
 - with the payment method table
 - as well as the client's table.

- Produce a report that shows
 - the payments,
 - with more details,
 - such as the name of the client,
 - and the payment method.

- Use an alias to simplify your code



The screenshot shows the MySQL Workbench interface. In the top-left pane, there is a code editor window containing the following SQL query:

```
1 • USE sql_invoicing;
2
3 • SELECT *
4   FROM payments p
5     JOIN clients c
6       ON p.client_id = c.client_id
7     JOIN payment_methods pm
8       ON p.payment_method = pm.payment_method_id
```

In the bottom-right pane, the results of the query are displayed in a grid format. The columns are: payment_id, client_id, invoice_id, date, amount, payment_method, client_id, name, address, city, state, phone, payment_method_id, name. The data is as follows:

payment_id	client_id	invoice_id	date	amount	payment_method	client_id	name	address	city	state	phone	payment_method_id	name
2	1	6	2019-01-03	74.55	1	1	Vinte	3 Nevada Parkway	Syracuse	NY	315-252-7305	1	Credit Card
3	3	11	2019-01-11	0.03	1	3	Yadel	096 Pawling Parkway	San Francisco	CA	415-144-6037	1	Credit Card
5	3	15	2019-01-15	80.31	1	3	Yadel	096 Pawling Parkway	San Francisco	CA	415-144-6037	1	Credit Card
6	3	17	2019-01-15	68.10	1	3	Yadel	096 Pawling Parkway	San Francisco	CA	415-144-6037	1	Credit Card
8	5	18	2019-01-16	10.00	2	5	Topiclounge	0863 Farmco Road	Portland	OR	971-888-9129	2	Cash
1	5	2	2019-02-12	8.18	1	5	Topiclounge	0863 Farmco Road	Portland	OR	971-888-9129	1	Credit Card
4	5	13	2019-01-26	87.44	1	5	Topiclounge	0863 Farmco Road	Portland	OR	971-888-9129	1	Credit Card
7	5	18	2019-01-08	32.77	1	5	Topiclounge	0863 Farmco Road	Portland	OR	971-888-9129	1	Credit Card

```

1 • USE sql_invoicing;
2
3 • SELECT
4     p.date,
5     p.invoice_id,
6     p.amount,
7     c.name,
8     pm.name
9  FROM payments p
10 JOIN clients c
11      ON p.client_id = c.client_id
12 JOIN payment_methods pm
13      ON p.payment_method = pm.payment_method_id
<

```

	date	invoice_id	amount	name
>	2019-02-12	2	8.18	Topidlounge Credit Card
	2019-01-03	6	74.55	Vinte Credit Card
	2019-01-11	11	0.03	Yadel Credit Card
	2019-01-26	13	87.44	Topidlounge Credit Card
	2019-01-15	15	80.31	Yadel Credit Card
	2019-01-15	17	68.10	Yadel Credit Card
	2019-01-08	18	32.77	Topidlounge Credit Card
	2019-01-08	18	10.00	Topidlounge Cash

COMPOUND JOIN CONDITIONS

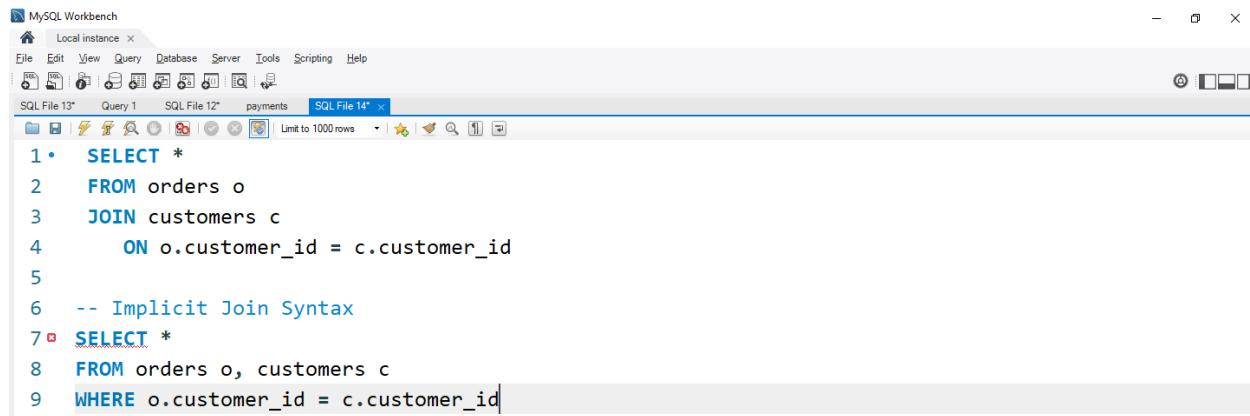
USE sql_store;

```

SELECT *
FROM orders_items oi
JOIN order_item_notes oin
    ON oi.order_id = oin.order_id
    AND oi.product_id = oin.product_id

```

IMPLICIT JOIN SYNTAX

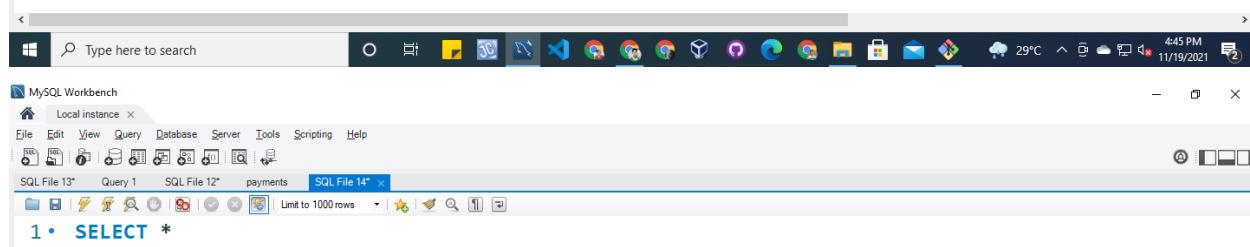


The screenshot shows the MySQL Workbench interface with the title bar "MySQL Workbench" and "Local instance". The menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The toolbar has various icons for file operations like Open, Save, Print, and Database. Below the toolbar, there are tabs for "SQL File 13*", "Query 1", "SQL File 12*", "payments", and "SQL File 14*". The main area contains the following SQL code:

```

1 • SELECT *
2   FROM orders o
3     JOIN customers c
4       ON o.customer_id = c.customer_id
5
6 -- Implicit Join Syntax
7 □ SELECT *
8   FROM orders o, customers c
9 WHERE o.customer_id = c.customer_id

```

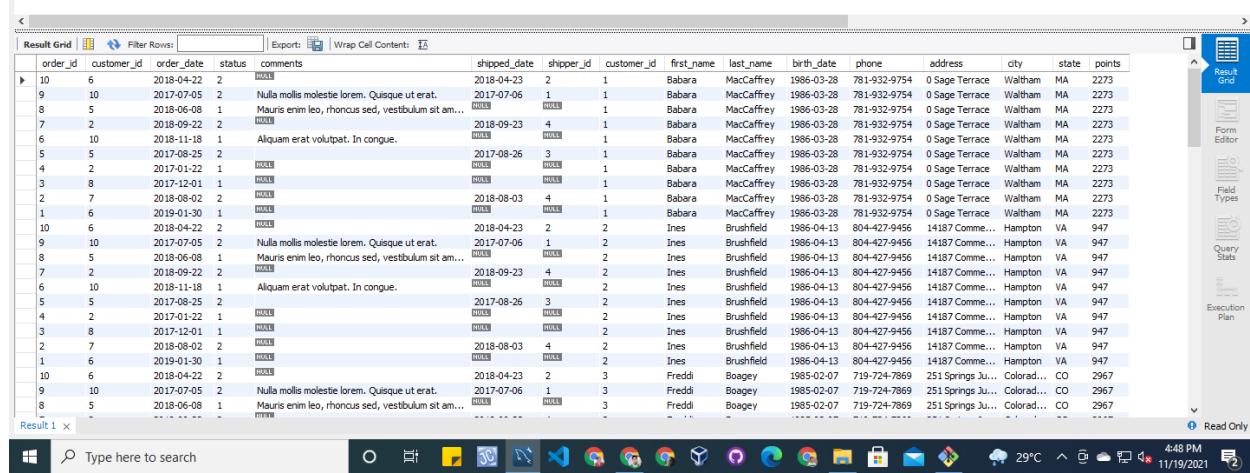


The screenshot shows the MySQL Workbench interface with the title bar "MySQL Workbench" and "Local instance". The menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The toolbar has various icons for file operations like Open, Save, Print, and Database. Below the toolbar, there are tabs for "SQL File 13*", "Query 1", "SQL File 12*", "payments", and "SQL File 14*". The main area contains the following SQL code:

```

1 • SELECT *
2   FROM orders o, customers c
3

```



The screenshot shows the MySQL Workbench interface with the title bar "MySQL Workbench" and "Local instance". The menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The toolbar has various icons for file operations like Open, Save, Print, and Database. Below the toolbar, there are tabs for "SQL File 13*", "Query 1", "SQL File 12*", "payments", and "SQL File 14*". The main area shows the results of the implicit join query:

order_id	customer_id	order_date	status	comments	shipped_date	shipper_id	customer_id	first_name	last_name	birth_date	phone	address	city	state	points
10	6	2018-04-22	2	NULL	2018-04-23	2	1	Babara	MacCaffrey	1986-03-28	781-932-9754	0 Sage Terrace	Waltham	MA	2273
9	10	2017-07-05	2	Null mollis molestie lorem. Quisque ut erat.	2017-07-06	1	1	Babara	MacCaffrey	1986-03-28	781-932-9754	0 Sage Terrace	Waltham	MA	2273
8	5	2018-06-08	1	Mauris enim leo, rhoncus sed, vestibulum sit am...	NULL	NULL	1	Babara	MacCaffrey	1986-03-28	781-932-9754	0 Sage Terrace	Waltham	MA	2273
7	2	2018-09-22	2	NULL	2018-09-23	4	1	Babara	MacCaffrey	1986-03-28	781-932-9754	0 Sage Terrace	Waltham	MA	2273
6	10	2018-11-18	1	Aliquam erat volutpat. In congue.	NULL	NULL	1	Babara	MacCaffrey	1986-03-28	781-932-9754	0 Sage Terrace	Waltham	MA	2273
5	5	2017-08-25	2	NULL	2017-08-26	3	1	Babara	MacCaffrey	1986-03-28	781-932-9754	0 Sage Terrace	Waltham	MA	2273
4	2	2017-01-22	1	NULL	NULL	NULL	1	Babara	MacCaffrey	1986-03-28	781-932-9754	0 Sage Terrace	Waltham	MA	2273
3	8	2017-12-01	1	NULL	NULL	1	Babara	MacCaffrey	1986-03-28	781-932-9754	0 Sage Terrace	Waltham	MA	2273	
2	7	2018-08-02	2	NULL	2018-08-03	4	1	Babara	MacCaffrey	1986-03-28	781-932-9754	0 Sage Terrace	Waltham	MA	2273
1	6	2019-01-30	1	NULL	NULL	1	Babara	MacCaffrey	1986-03-28	781-932-9754	0 Sage Terrace	Waltham	MA	2273	
10	6	2018-04-22	2	NULL	2018-04-23	2	2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Comme...	Hampton	VA	947
9	10	2017-07-05	2	Null mollis molestie lorem. Quisque ut erat.	2017-07-06	1	2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Comme...	Hampton	VA	947
8	5	2018-06-08	1	Mauris enim leo, rhoncus sed, vestibulum sit am...	NULL	NULL	2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Comme...	Hampton	VA	947
7	2	2018-09-22	2	NULL	2018-09-23	4	2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Comme...	Hampton	VA	947
6	10	2018-11-18	1	Aliquam erat volutpat. In congue.	NULL	NULL	2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Comme...	Hampton	VA	947
5	5	2017-08-25	2	NULL	2017-08-26	3	2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Comme...	Hampton	VA	947
4	2	2017-01-22	1	NULL	NULL	2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Comme...	Hampton	VA	947	
3	8	2017-12-01	1	NULL	NULL	2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Comme...	Hampton	VA	947	
2	7	2018-08-02	2	NULL	2018-08-03	4	2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Comme...	Hampton	VA	947
1	6	2019-01-30	1	NULL	NULL	2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Comme...	Hampton	VA	947	
10	6	2018-04-22	2	NULL	2018-04-23	2	3	Freddi	Boagey	1985-02-07	719-724-7869	251 Springs Ju...	Colorad...	CO	2967
9	10	2017-07-05	2	Null mollis molestie lorem. Quisque ut erat.	2017-07-06	1	3	Freddi	Boagey	1985-02-07	719-724-7869	251 Springs Ju...	Colorad...	CO	2967
8	5	2018-06-08	1	Mauris enim leo, rhoncus sed, vestibulum sit am...	NULL	NULL	3	Freddi	Boagey	1985-02-07	719-724-7869	251 Springs Ju...	Colorad...	CO	2967

-- Even though MySQL supports the syntax, it's something that I suggest you not to use,
Because if you accidentally forget to type out the WHERE clause, you will get a
cross join.

-- It's better to use the EXPLICIT join syntax, because this syntax forces you to type out
the joint condition, if you simply order with customers, without typing the joint condition,
you're going to get a syntax error.

-- Be aware of the Implicit Syntax, but write all your joints using the Explicit syntax.

OUTER JOINS

— Return all customers whether they have any orders or not

```
SELECT
    c.customer_id,
    c.first_name,
    o.order_id
FROM customers c
JOIN orders o
    ON c.customer_id = o.customer_id
ORDER BY c.customer_id
```

~ EXERCISE 13 ~

The screenshot shows the MySQL Workbench interface. In the top-left pane, there is a code editor with the following SQL query:

```
1 • SELECT
2     p.product_id,
3     p.name,
4     oi.quantity
5 FROM products p
6 LEFT JOIN order_items oi
7     ON p.product_id = oi.product_id
```

In the bottom-left pane, the results of the query are displayed in a grid:

product_id	name	quantity
2	Pork - Bacon,back Peameal	3
2	Pork - Bacon,back Peameal	4
3	Lettuce - Romaine, Heart	10
3	Lettuce - Romaine, Heart	7
3	Lettuce - Romaine, Heart	4
3	Lettuce - Romaine, Heart	7
4	Broccolini - Gaylan, Chinese	4
4	Broccolini - Gaylan, Chinese	4
5	Sauce - Ranch Dressing	1
5	Sauce - Ranch Dressing	2
6	Petit Baguette	2
6	Petit Baguette	5
7	Sweet Pea Sprouts	10
8	Island Oasis - Raspberry	2
9	Longan	9
10	Broom - Push	7

The MySQL Workbench interface includes various toolbars and panels on the right side, such as 'Result Grid', 'Form Editor', 'Field Types', and 'Query Stats'.

-- You can get the same result in both left or right join, you just swap the orders from the tables.

-- As a best practice, Avoid using Right joins and use Left joins instead.

OUTER JOINING BETWEEN MULTIPLE TABLES

```
SELECT
    c.customer_id,
    c.first_name,
    o.order_id,
    sh.name AS shipper
FROM customers c
LEFT JOIN orders o
    ON c.customer_id = o.customer_id
LEFT JOIN shippers sh
    ON o.shipper_id = sh.shipper_id
ORDER BY c.customer_id
```

~ EXERCISE 14 ~

MySQL Workbench

Local instance

File Edit View Query Database Server Tools Scripting Help

SQL File 13* SQL File 12* payments orders SQL File 14* SQL File 15* SQL File 16*

1 • SELECT
 2 o.order_id,
 3 o.order_date,
 4 c.first_name AS customer,
 5 sh.name AS shipper
 6 FROM orders o
 7 JOIN customers c
 8 ON o.customer_id = c.customer_id
 9 JOIN shippers sh
 10 ON o.shipper_id = sh.shipper_id

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

order_id	order_date	customer	shipper
9	2017-07-05	Levy	Hettinger LLC
10	2018-04-22	Elka	Schinner-Predovic
5	2017-08-25	Clemmie	Satterfield LLC
2	2018-08-02	Irene	Mraz, Renner and Nolan
7	2018-09-22	Ines	Mraz, Renner and Nolan

Activate Windows
Go to Settings to activate Windows.

Result 1 | Read Only

MySQL Workbench

Local instance

File Edit View Query Database Server Tools Scripting Help

SQL File 13* SQL File 12* payments orders SQL File 14* SQL File 15* SQL File 16*

1 • SELECT
 2 o.order_id,
 3 o.order_date,
 4 c.first_name AS customer,
 5 sh.name AS shipper
 6 FROM orders o
 7 JOIN customers c
 8 ON o.customer_id = c.customer_id
 9 LEFT JOIN shippers sh
 10 ON o.shipper_id = sh.shipper_id

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

order_id	order_date	customer	shipper
1	2019-01-30	Elka	NULL
2	2018-08-02	Irene	Mraz, Renner and Nolan
3	2017-12-01	Theodore	NULL
4	2017-01-22	Ines	NULL
5	2017-08-25	Clemmie	Satterfield LLC
6	2018-11-18	Levy	NULL
7	2018-09-22	Ines	Mraz, Renner and Nolan
8	2018-06-08	Clemmie	NULL
9	2017-07-05	Levy	Hettinger LLC
10	2018-04-22	Elka	Schinner-Predovic

Activate Windows
Go to Settings to activate Windows.

Result 2 | Read Only

The screenshot shows the MySQL Workbench interface. In the top-left, there's a title bar for 'MySQL Workbench' with a 'Local instance' connection. Below it is a menu bar with 'File', 'Edit', 'View', 'Query', 'Database', 'Server', 'Tools', 'Scripting', and 'Help'. A toolbar follows with icons for file operations like Open, Save, Print, and Database. The main area contains a query editor with the following SQL code:

```

1 • SELECT
2     o.order_id,
3     o.order_date,
4     c.first_name AS customer,
5     sh.name AS shipper,
6     os.name AS status
7 FROM orders o
8 JOIN customers c
9     ON o.customer_id = c.customer_id
10 LEFT JOIN shippers sh
11     ON o.shipper_id = sh.shipper_id
12 JOIN order_statuses os
13     ON o.status = os.order_status_id

```

Below the query is a results grid titled 'Result Grid' showing the output of the query. The columns are 'order_id', 'order_date', 'customer', 'shipper', and 'status'. The data includes rows for various orders with their respective details.

order_id	order_date	customer	shipper	status
8	2018-06-08	Clemmie	NULL	Processed
6	2018-11-18	Levy	NULL	Processed
4	2017-01-22	Ines	NULL	Processed
3	2017-12-01	Thader	NULL	Processed
1	2019-01-30	Elka	NULL	Processed
10	2018-04-22	Elka	Schninner-Predovic	Shipped
9	2017-07-05	Levy	Hettinger LLC	Shipped
7	2018-09-22	Ines	Mraz, Renner and Nolan	Shipped
5	2017-08-25	Clemmie	Satterfield LLC	Shipped
2	2018-08-02	Ilene	Mraz, Renner and Nolan	Shipped

At the bottom of the screen is a Windows taskbar with various pinned icons and a search bar. The system tray shows the date and time as 5:22 PM, 11/19/2021, and the temperature as 29°C.

SELF OUTER JOINS

USE sql_hr;

```

SELECT
    e.employee_id,
    e.first_name,
    m.first_name AS manager
FROM employees e
LEFT JOIN employees m
    ON e.reports_to = m.employee_id

```

```

1 • USE sql_hr;
2
3 • SELECT
4     e.employee_id,
5     e.first_name,
6     m.first_name AS manager
7 FROM employees e
8 LEFT JOIN employees m
9     ON e.reports_to = m.employee_id

```

employee_id	first_name	manager
33391	Darcy	Yovonna
37270	Yovonna	
37951	Sayer	Yovonna
40448	Mindy	Yovonna
56274	Kerann	Yovonna
63196	Alaster	Yovonna
67009	North	Yovonna
67370	Elladine	Yovonna
68249	Nisse	Yovonna
72540	Guthrey	Yovonna
72913	Kass	Yovonna
75900	Virge	Yovonna
76196	Minlie	Yovonna
80529	Lynde	Yovonna
80679	Mildrid	Yovonna
84791	Hazel	Yovonna
95213	Cole	Yovonna
96513	Theresa	Yovonna

Activate Windows
Go to Settings to activate Windows.

Result 3 X

5:27 PM 11/19/2021

THE USING CLAUSE

If column names are exactly the same, you can simplify the join with the **USING** clause.

```

SELECT *
FROM customers c
JOIN orders o
    USING (customer_id)

```

~ EXERCISE 15 ~

MySQL Workbench

Local instance X

File Edit View Query Database Server Tools Scripting Help

SQL File 13* SQL File 12* SQL File 14* SQL File 15* SQL File 16* SQL File 17* SQL File 18* SQL File 19* X

1 • USE sql_invoicing;

2

3 • SELECT

4 p.date,

5 c.name AS client,

6 p.amount,

7 pm.name AS payment_method

8 FROM payments p

9 JOIN clients c USING (client_id)

10 JOIN payment_methods pm

11 ON p.payment_method = pm.payment_method_id

12

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	date	client	amount	payment_method
▶	2019-02-12	Topidoung	8.18	Credit Card
	2019-01-03	Vint	74.55	Credit Card
	2019-01-11	Yadel	0.03	Credit Card
	2019-01-26	Topidoung	87.44	Credit Card
	2019-01-15	Yadel	80.31	Credit Card
	2019-01-15	Yadel	68.10	Credit Card
	2019-01-08	Topidoung	32.77	Credit Card
	2019-01-08	Topidoung	10.00	Cash

Result 2 x Type here to search

Activate Windows
Go to Settings to activate Windows.

Result Grid Form Editor Field Types Query Stats

Read Only

NATURAL JOINS

```
SELECT
    o.order_id,
    c.first_name
FROM orders o
NATURAL JOIN customers c
```

The screenshot shows the MySQL Workbench interface. In the top-left pane, a query editor window displays the following SQL code:

```
1 SELECT
2     o.order_id,
3     c.first_name
4 FROM orders o
5 NATURAL JOIN customers c
```

In the bottom-left pane, a "Result Grid" shows the output of the query:

order_id	first_name
4	Ines
7	Ines
5	Clemmie
8	Clemmie
1	Elka
10	Elka
2	Ilene
3	Thacher
6	Levy
9	Levy

The right side of the interface features a vertical toolbar with icons for various functions: Result Grid (selected), Form Editor, Field Types, Query Stats, and Execution Plan.

At the bottom of the screen, a Windows taskbar is visible with the message "Activate Windows Go to Settings to activate Windows." and a "Read Only" status indicator.

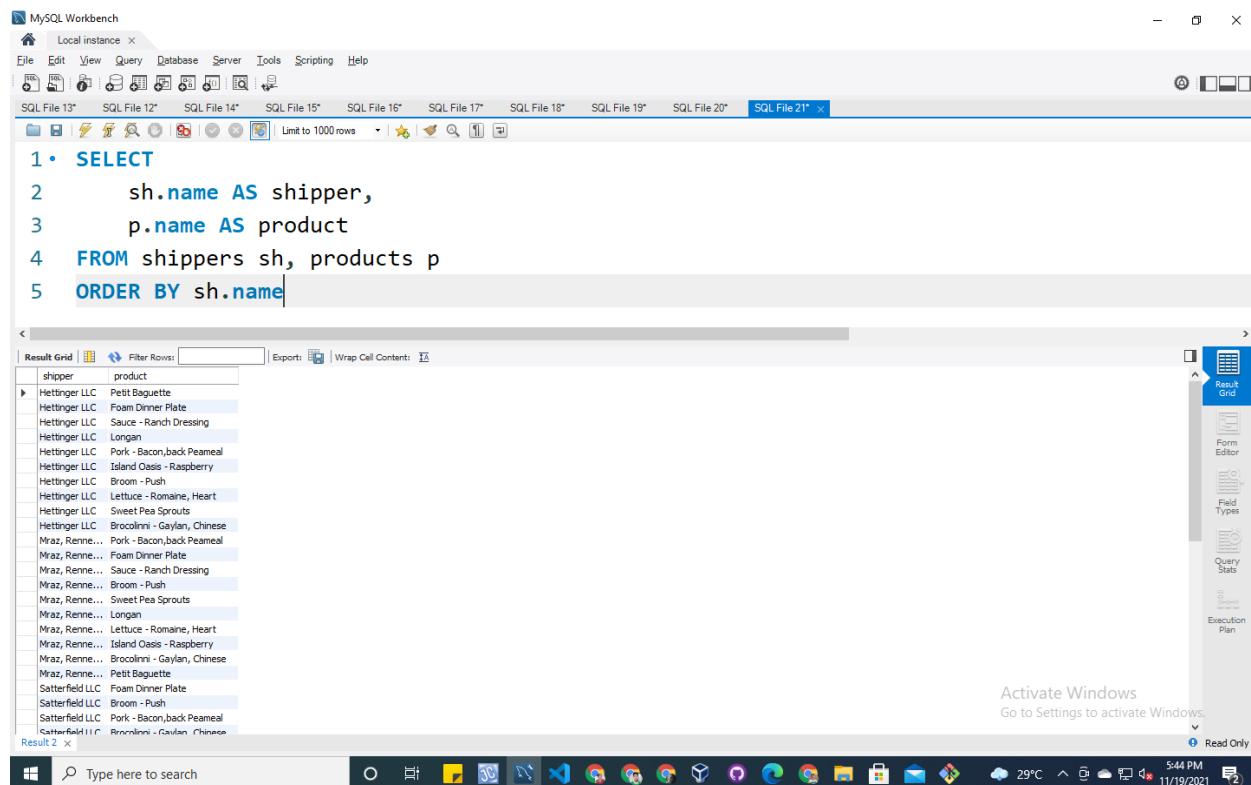
CROSS JOINS

— Combine every color with every size

```
SELECT *
FROM customers
CROSS JOIN sizes
```

~EXERCISE 16~

- Do a cross join between shippers and products
- using the implicit syntax
- and then using the explicit syntax



The screenshot shows the MySQL Workbench interface with a query editor window. The query is:

```
1 • SELECT
2     sh.name AS shipper,
3     p.name AS product
4 FROM shippers sh, products p
5 ORDER BY sh.name
```

The result grid displays the following data:

shipper	product
Hettinger LLC	Pest Baguette
Hettinger LLC	Foam Dinner Plate
Hettinger LLC	Sauce - Ranch Dressing
Hettinger LLC	Longan
Hettinger LLC	Pork - Bacon,back Peameal
Hettinger LLC	Island Oasis - Raspberry
Hettinger LLC	Broom - Push
Hettinger LLC	Lettuce - Romaine, Heart
Hettinger LLC	Sweet Pea Sprouts
Hettinger LLC	Broccolini - Gaylan, Chinese
Mraz, Renne...	Pork - Bacon,back Peameal
Mraz, Renne...	Foam Dinner Plate
Mraz, Renne...	Sauce - Ranch Dressing
Mraz, Renne...	Broom - Push
Mraz, Renne...	Sweet Pea Sprouts
Mraz, Renne...	Longan
Mraz, Renne...	Lettuce - Romaine, Heart
Mraz, Renne...	Island Oasis - Raspberry
Mraz, Renne...	Broccolini - Gaylan, Chinese
Mraz, Renne...	Pest Baguette
Satterfield LLC	Foam Dinner Plate
Satterfield LLC	Broom - Push
Satterfield LLC	Pork - Bacon,back Peameal
Satterfield LLC	Broccolini - Gaylan, Chinese

There are 21 rows in the result set.

The screenshot shows the MySQL Workbench interface with a query editor and a results grid. The query editor contains the following SQL code:

```

1 • SELECT
2     sh.name AS shipper,
3     p.name AS product
4 FROM shippers sh
5 CROSS JOIN products p
6 ORDER BY sh.name

```

The results grid displays the output of the query, showing columns for shipper and product. The data includes various items from different suppliers like Hettinger LLC and Mraz, Renne... along with their respective product names.

shipper	product
Hettinger LLC	Petit Baguette
Hettinger LLC	Foam Dinner Plate
Hettinger LLC	Sauce - Ranch Dressing
Hettinger LLC	Longan
Hettinger LLC	Pork - Bacon,back Peameal
Hettinger LLC	Island Oasis - Raspberry
Hettinger LLC	Broom - Push
Hettinger LLC	Lettuce - Romaine, Heart
Hettinger LLC	Sweet Pea Sprouts
Hettinger LLC	Brocolini - Gaylan, Chinese
Mraz, Renne...	Pork - Bacon,back Peameal
Mraz, Renne...	Foam Dinner Plate
Mraz, Renne...	Sauce - Ranch Dressing
Mraz, Renne...	Broom - Push
Mraz, Renne...	Sweet Pea Sprouts
Mraz, Renne...	Longan
Mraz, Renne...	Lettuce - Romaine, Heart
Mraz, Renne...	Island Oasis - Raspberry
Mraz, Renne...	Brocolini - Gaylan, Chinese
Mraz, Renne...	Petit Baguette
FentressFood LLC	Foam Dinner Plate

Activate Windows
Go to Settings to activate Windows.

UNIONS

— Combine records from multiple result sets

```

SELECT name, address
FROM customers
UNION
SELECT name, address
FROM clients

```

~ EXERCISE 17 ~

The screenshot shows the MySQL Workbench interface. The query editor window contains the following SQL code:

```
7 WHERE points < 2000
8 UNION
9 SELECT
10     customer_id,
11     first_name,
12     points,
13     'Silver' AS type
14 FROM customers
15 WHERE points BETWEEN 2000 AND 3000
16 UNION
17 SELECT
18     customer_id,
19     first_name,
20     points,
21     'Gold' AS type
22 FROM customers
23 WHERE points > 3000
24 ORDER BY first_name
```

The results grid displays the following data:

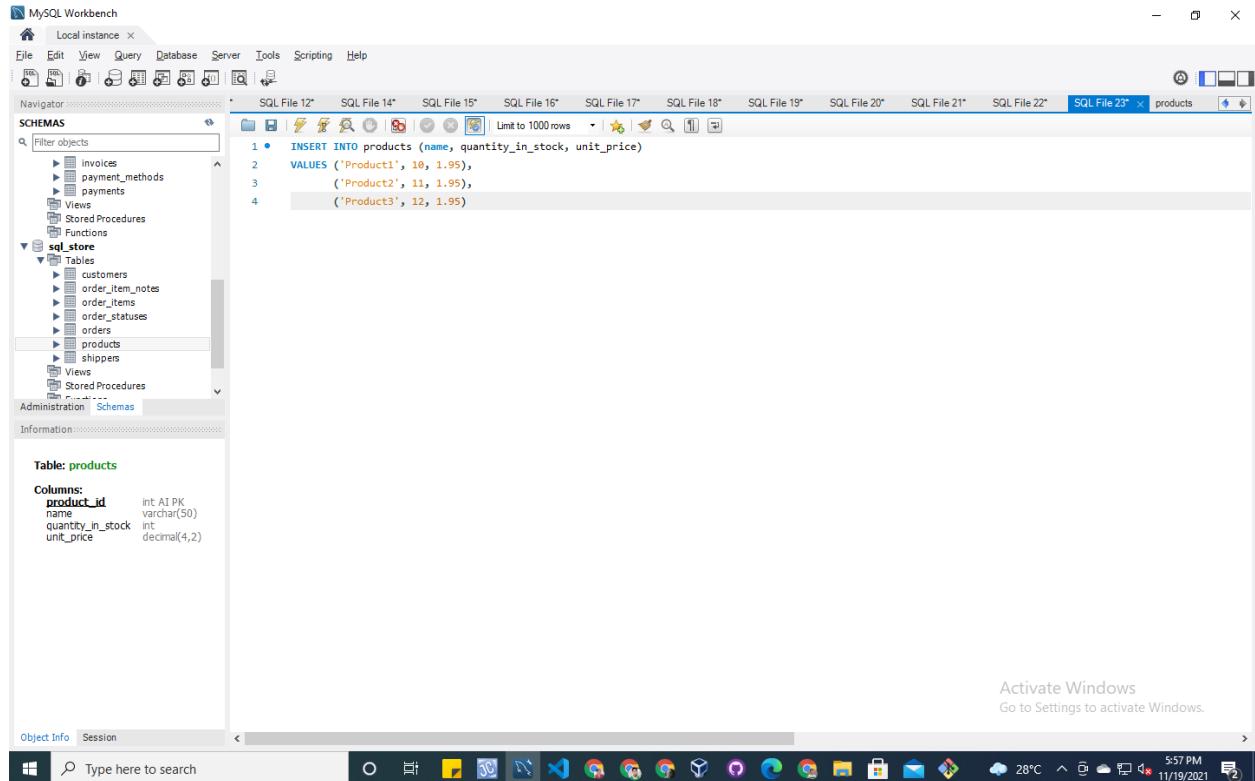
customer_id	first_name	points	type
4	Amber	457	Bronze
1	Babara	2273	Silver
5	Clemmie	3675	Gold
6	Elka	3073	Gold
3	Freddi	2967	Silver
7	Ilene	1672	Bronze
2	Ires	947	Bronze
10	Levy	796	Bronze
9	Ronola	1486	Bronze
8	Thacher	205	Bronze

The status bar at the bottom right shows "Activate Windows Go to Settings to activate Windows".

INSERTING MULTIPLE ROWS

~ EXERCISE 18 ~

-- Insert three rows in the products table



The screenshot shows the MySQL Workbench interface. The SQL editor tab contains the following SQL code:

```
1 • INSERT INTO products (name, quantity_in_stock, unit_price)
2   VALUES ('Product1', 10, 1.95),
3          ('Product2', 11, 1.95),
4          ('Product3', 12, 1.95)
```

The Navigator pane shows the database schema, including the 'products' table under the 'sql_store' schema. The Object Info pane displays the structure of the 'products' table:

Table: products

Column:

product_id	int AI PK
name	varchar(50)
quantity_in_stock	int
unit_price	decimal(4,2)

At the bottom, the Windows taskbar shows various application icons and the system tray.

MySQL Workbench

Local instance X

File Edit View Query Database Server Tools Scripting Help

Navigator

Schemas

Filter objects

sql_store

Tables

customers
order_item_notes
order_items
order_statuses
orders
products
shippers

Views

Stored Procedures

Functions

Administration Schemas

Information

Table: products

Columns:

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	Brocolini - Gaylan, Chinese	90	4.53
5	Sauce + Ranch Dressing	94	1.63
6	Peit 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
11	Product1	10	1.95
12	Product2	11	1.95
13	Product3	12	1.95
*	NULL	NULL	NULL

Result Grid Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

Activate Windows
Go to Settings to activate Windows.

Object Info Session products 1 X

Type here to search

Windows Taskbar: 557 PM 11/19/2021

MySQL Workbench interface showing the results of a query against the 'products' table in the 'sql_store' schema. The results are displayed in a grid format with columns: product_id, name, quantity_in_stock, and unit_price. The grid shows 13 rows of data, including several placeholder entries at the bottom.

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	Brocolini - Gaylan, Chinese	90	4.53
5	Sauce + Ranch Dressing	94	1.63
6	Peit 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
11	Product1	10	1.95
12	Product2	11	1.95
13	Product3	12	1.95
*	NULL	NULL	NULL

CREATING A COPY OF A TABLE

~ EXERCISE 19 ~

The screenshot shows the MySQL Workbench interface. The left pane displays the Navigator with the schema 'sql_invoicing' selected. Under 'Tables', the 'invoices_archived' table is visible. The right pane contains a SQL editor window with the following code:

```
1 • USE sql_invoicing;
2
3 • CREATE TABLE invoices_archived AS
4 SELECT
5     i.invoice_id,
6     i.number,
7     c.name AS client,
8     i.invoice_total,
9     i.payment_total,
10    i.invoice_date,
11    i.payment_date,
12    i.due_date
13 FROM invoices i
14 JOIN clients c
15     USING (client_id)
16 WHERE payment_date IS NOT NULL
```

Below the code, the 'Table: invoices_archived' section shows the columns and their data types:

Columns:	
invoice_id	int
number	varchar(50)
client	varchar(50)
invoice_total	decimal(9,2)
payment_total	decimal(9,2)
invoice_date	date
payment_date	date
due_date	date

The status bar at the bottom right indicates 'Activate Windows Go to Settings to activate Windows.' The taskbar at the bottom shows various application icons.

UPDATING MULTIPLE ROWS

~EXERCISE 20~

- Write a SQL statement to
- give any customers born before 1990
- 50 extra points

The screenshot shows the MySQL Workbench interface. The SQL editor pane contains the following SQL code:

```
1 • USE sql_store;
2
3 • UPDATE customers
4   SET points = points + 50
5 WHERE birth_date < '1990-01-01'
```

The Navigator pane shows the database schema with the following tables listed under the 'invoices_archived' schema:

- clients
- invoices
- payments
- payments
- Views
- Stored Procedures
- Functions

Under the 'sql_store' schema, the following tables are listed:

- customers
- order_item_notes
- order_items
- order_statuses
- orders
- orders archived

The bottom of the screen shows the Windows taskbar with various application icons and system status.

MySQL Workbench Local instance

File Edit View Query Database Server Tools Scripting Help

Navigator

Schemas

Tables: clients, invoices, invoices_archived, payment_methods, payments

Views

Stored Procedures

Functions

sql_store

Tables: customers, order_item_notes, order_items, order_statuses, orders, orders_archived

Administration Schemas

Table: invoices_archived

Columns:

- invoice_id int
- number varchar(50)
- client varchar(50)
- invoice_total decimal(9,2)
- payment_total decimal(9,2)
- invoice_date date
- payment_date date
- due_date date

SQL File 23* products SQL File 24* SQL File 25* SQL File 26* SQL File 28* SQL File 19* SQL File 20* SQL File 21* invoices_archived SQL File 22* customers

1 • SELECT * FROM sql_store.customers;

Result Grid Filter Rows: Edit Export/Import Wrap Cell Content

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	Waltham	2323
2	Ines	Brusfield	1986-04-13	804-427-9456	14187 Commercial Trail	Hampton	VA	997
3	Freddi	Boagey	1985-02-07	719-724-7869	251 Springs Junction	Colorado Springs	CO	3017
4	Amber	Roseburgh	1974-04-14	407-231-8017	30 Arapahoe Terrace	Orlando	FL	507
5	Clemmie	Betshley	1973-11-07	NULL	5 Spohn Circle	Arlington	TX	3725
6	Eika	Iwiddell	1991-09-04	312-480-8998	7 Manley Drive	Chicago	IL	3073
7	Tene	Dowson	1964-08-30	611-641-4759	50 Lillian Crossing	Nashville	TN	1722
8	Thaddei	Naseby	1993-07-17	941-527-3977	538 Mosinee Center	Sarasota	FL	205
9	Ronola	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	846
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result Grid Form Editor Field Types Query Stats Execution Plan

Activate Windows Go to Settings to activate Windows.

Object Info Session customers 1 x

Type here to search

Windows Taskbar: 6:46 PM 29°C 11/19/2021

USING SUBQUERIES IN UPDATES

~ EXERCISE 21 ~

The screenshot shows the MySQL Workbench interface. The left pane displays the Navigator and Schemas (customers) for the sql_store database. The central pane contains a SQL editor with the following query:

```
1 •  SELECT *
2      FROM customers
3     WHERE points > 3000
```

The right pane shows a table editor for the 'clients' table, displaying the following data:

customer_id	first_name	last_name	birth_date	phone	address	city	state	points
3	Freddi	Boagey	1985-02-07	719-724-7869	251 Springs Junction	Colorado Springs	CO	3067
5	Clemmie	Bethley	1973-11-07	NULL	5 Spohn Circle	Arlington	TX	3775
6	Ella	Twiddell	1991-09-04	312-480-8498	7 Manley Drive	Chicago	IL	3073
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

The bottom pane shows an output window with the following message:

```
39 19:01:12 UPDATE orders SET comments = 'Gold customer' WHERE customer_id IN (SELECT customer_id FROM customers WHERE points > 3000)
```

Message: 0 rows affected. Rows matched: 4 Duration: 0.000 sec

MySQL Workbench

Local instance X

File Edit View Query Database Server Tools Scripting Help

Navigator SQL File 24* SQL File 25* SQL File 26* SQL File 28* SQL File 19* SQL File 20* SQL File 21* SQL File 22* SQL File 23* SQL File 25* SQL File 24* customers

SCHEMAS

Filter objects

Stored Procedures
Functions
Tables
 customers
 Columns
 Indexes
 Foreign Keys
 Triggers
 order_item_notes
 order_items
 order_statuses
 orders
 orders_archived
 products
 shippers
Views

Administration Schemas

Information

Table: clients

Columns:

client_id	int PK
name	varchar(50)
address	varchar(50)
city	varchar(50)
state	char(2)
phone	varchar(50)

1 • UPDATE orders

```
2 SET comments = 'Gold customer'
3 WHERE customer_id IN
4   (SELECT customer_id
5    FROM customers
6    WHERE points > 3000)
```

Action Output

Object Info Session

Type here to search

Activate Windows Go to Settings to activate Windows.

Duration / Fetch 0 rows/sec

7:03 PM 11/19/2021

MySQL Workbench

Local instance X

File Edit View Query Database Server Tools Scripting Help

Navigator SQL File 24* SQL File 25* SQL File 26* SQL File 28* SQL File 19* SQL File 20* SQL File 21* SQL File 22* SQL File 23* SQL File 25* SQL File 24* customers

SCHEMAS

Filter objects

Stored Procedures
Functions
Tables
 customers
 Columns
 Indexes
 Foreign Keys
 Triggers
 order_item_notes
 order_items
 order_statuses
 orders
 orders_archived
 products
 shippers
Views

Administration Schemas

Information

Table: clients

Columns:

client_id	int PK
name	varchar(50)
address	varchar(50)
city	varchar(50)
state	char(2)
phone	varchar(50)

1 • SELECT * FROM sql_store.customers;

Result Grid

Filter Rows: []

Edit: []

Export/Import: []

Wrap Cell Content: []

customers 1

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	MA	2373
2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Commercial Trail	Hampton	VA	1047
3	Freddi	Boagey	1985-02-07	719-724-7869	251 Springs Junction	Colorado Springs	CO	3067
4	Ambur	Roseburgh	1974-04-14	407-231-8017	30 Arapahoe Terrace	Orlando	FL	557
5	Clemmie	Bettsley	1973-11-07	5 Spohn Circle	Arlington	TX	3775	
6	Eika	Twiddell	1991-09-04	312-480-8498	7 Marley Drive	Chicago	IL	3073
7	Irene	Dowson	1964-08-30	615-641-4759	50 Lillian Crossing	Nashville	TN	1772
8	Thadde	Naseby	1993-07-17	941-527-3977	538 Mosinee Center	Sarasota	FL	205
9	Ronola	Rumgay	1992-05-23	559-181-3744	3520 Ohio Trail	Visalia	CA	1486

Output

Action Output

Object Info Session

Type here to search

Activate Windows Go to Settings to activate Windows.

Duration / Fetch 0 rows/sec

7:03 PM 11/19/2021