

# *DATABASE*

# *PROJECT*

*2022 W - T1 CSD:2206*

*DATBASE DESIGN & SQL 03*  
*(CSAM GROUP 1)*

# *PROJECT PART – 1*

## *RESTAURANT MANAGEMENT SYSTEM*

The objective of this part of the project is to design the database which is required for the restaurant management along with the implementation of the some basic constraints.

### *Group Information:*

Harpreet Kaur (c0854330)

Puneet Gill (c0856076)

Parminder kaur (c0854624)

# LIST OF THE ENTITIES FOR THE RESTAURANT MANAGEMENT SYSTEM:

## 1. Manager

MANAGER	
PRIMARY KEY	manager_id
	manager_name
	manager_address

### Description of the entity: Manager

manager_id	Unique id of the manager
manager_name	Name of the manager
manager_address	Address of the manager

Primary key: manager\_id

Required/ mandatory attributes: manager\_id, manager\_name, manager\_address

## 2. Cook

Cook	
PRIMARY KEY	cook_id
	cook_name
	address
	mobile
	speciality

### Description of the entity: Cook

cook_id	Unique id of the chef
cook_name	name of the chef
Address	Address of the chef
Mobile	Mobile of the chef
speciality	Special skill of the chef

Primary key: cook\_id

Required/ mandatory attributes: cook\_id, cook\_name, address, mobile, speciality

## 3. Client

CLIENT	
PRIMARY KEY	C_email
	C_name
	C_Address
	C_Mobile

### Description of the entity: Client

C_email	Email of the customer which is unique to everyone
C_name	Name of the customer
C_address	Address of the customer
C_mobile	Mobile of the customer

Primary key: C\_email

Required/ mandatory attributes:C\_email , C\_name , C\_address, C\_mobile

## 4. Menu

Menu	
PRIMARY KEY	item_id
	Item_name
	price
	status
	category

### Description of the entity: Menu

item_id	It is the unique id given to each item or dish
Item_name	Name of the item or dish
Price	Cost of the item/ dish
Status	Is it available or not?
Category	Fastfood or organic?

Primary key: item\_id

Required/ mandatory attributes: item\_id , item\_name, status, price

Optional attribute: category

## 5. Order INFO

Order INFO	
PRIMARY KEY	order_id
	quantity
	Torec
	TOdel

### Description of the entity: Order INFO

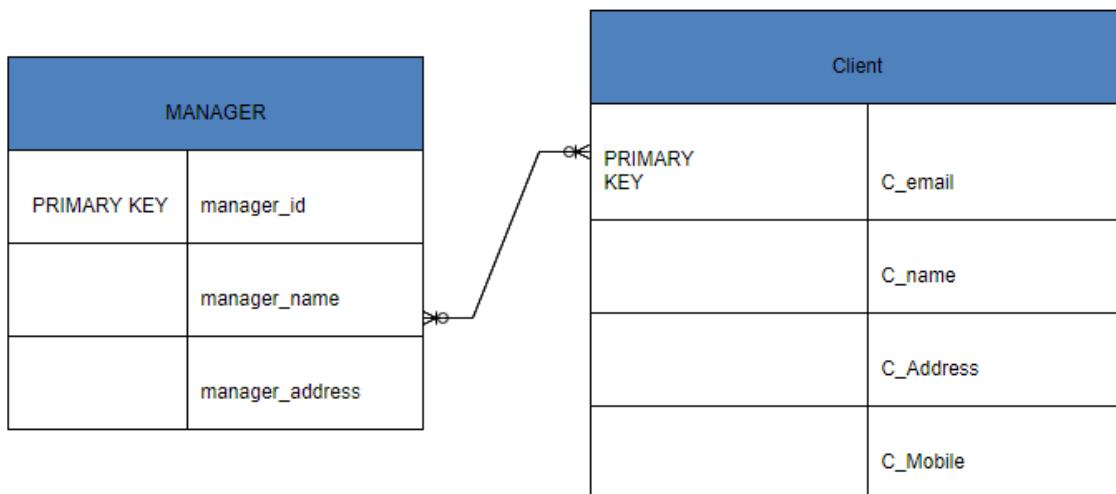
order_id	Unique number of the specific order
Quantity	Number of items ordered
TOrec	Time of receiving
TOdel	Time of delivery

Primary key: order\_id

Required/ mandatory attributes: order\_id, TOdel, TOrec, quantity

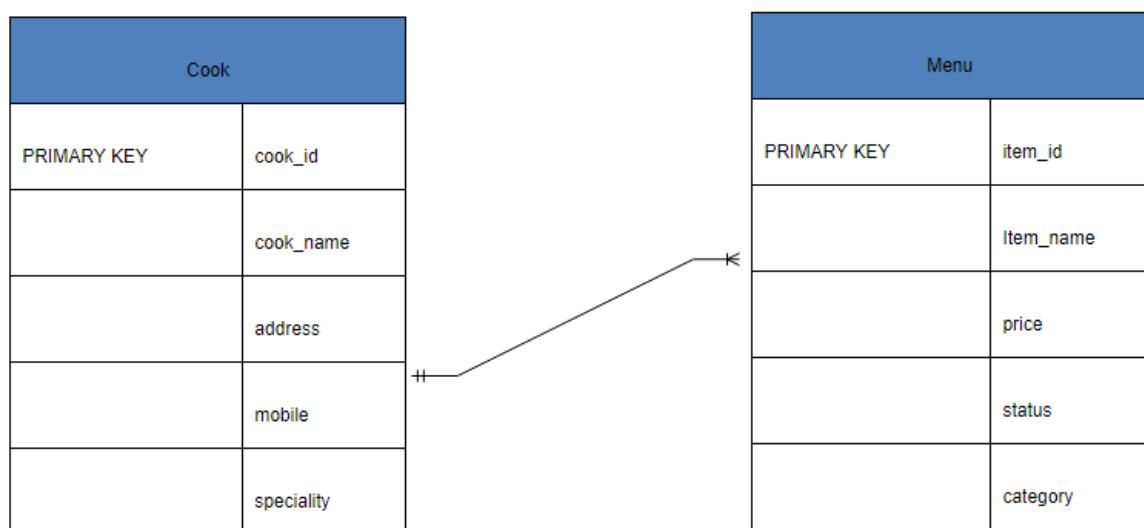
# *RELATIONSHIP BETWEEN THE ENTITIES*

## (I)RELATIONSHIP BETWEEN MANAGER AND COOK



- \*one manager can serve number of clients
- \*one client can be served by number of managers

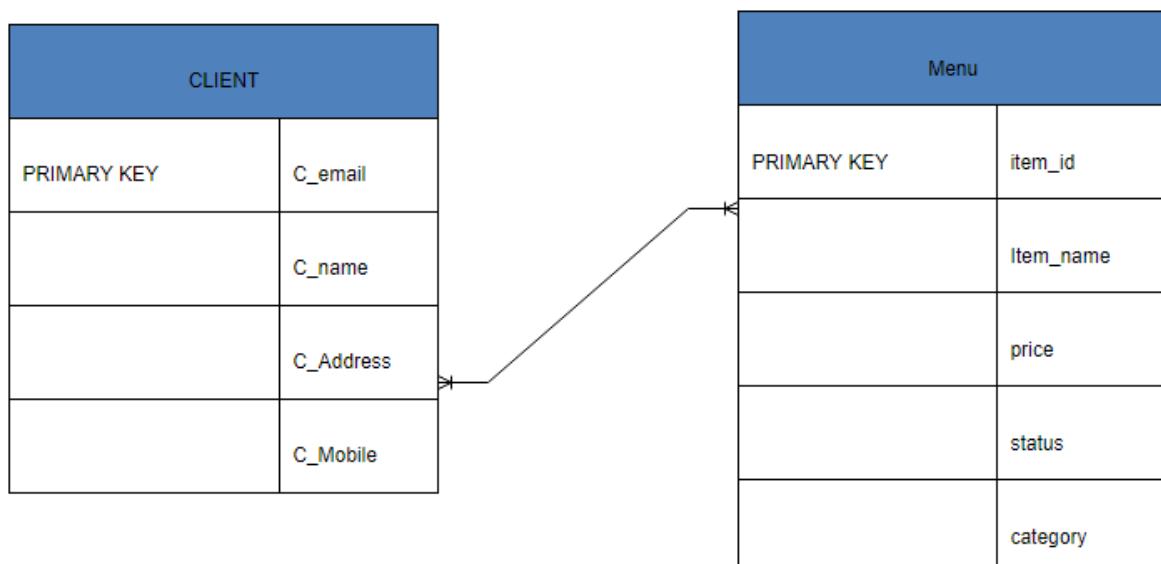
## (ii)RELATIONSHIP BETWEEN COOK AND MENU



\*ONE cook can prepare many menus

\*one menu can be prepared by one cook

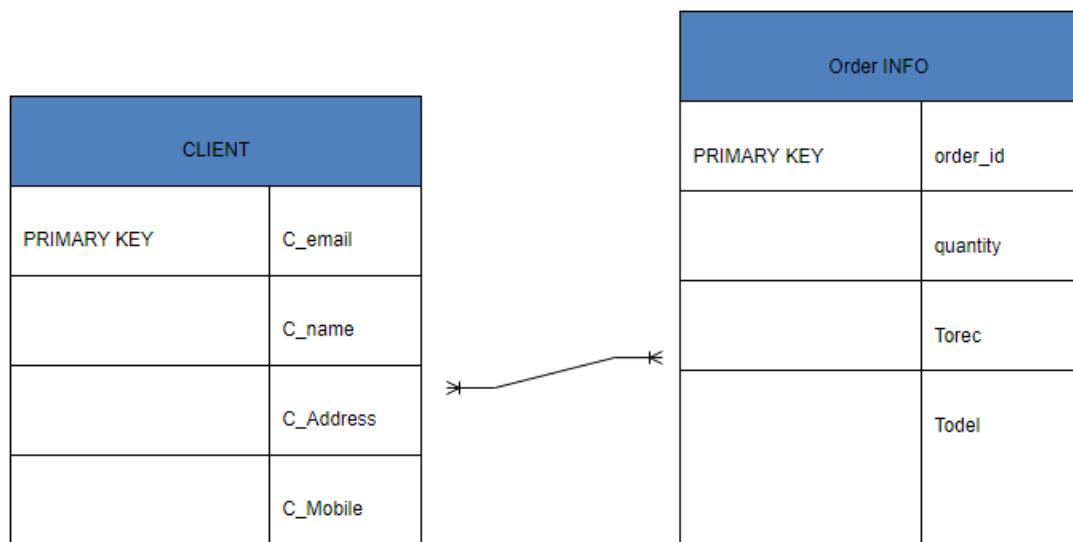
### (iii) RELATIONSHIP BETWEEN CLIENT AND MENU



\*ONE client have many menus

\*One menu have many clients

#### (iv)RELATIONSHIP BETWEEN CLIENT AND ORDER



\*ONE CLIENT CAN PLACE MANY ORDERS

\*ONE ORDER CAN BE PLACED BY MANY CLIENTS

## RELATIONSHIP MATRIX

	MANAGER	COOK	CLIENT	MENU	ORDER INFO
MANAGER			SERVES		
COOK				PREPARE	
CLIENT	SERVED BY			DECIDES	PLACE
MENU		PREPARE BY	DECIDE BY		
ORDER INFO			PLACED BY		

THE description of the entities along with the datatypes is given below:

MANAGER		
PK	manager_id	integer(10)
	manger_name	varchar(20)
	manager_address	varchar(20)

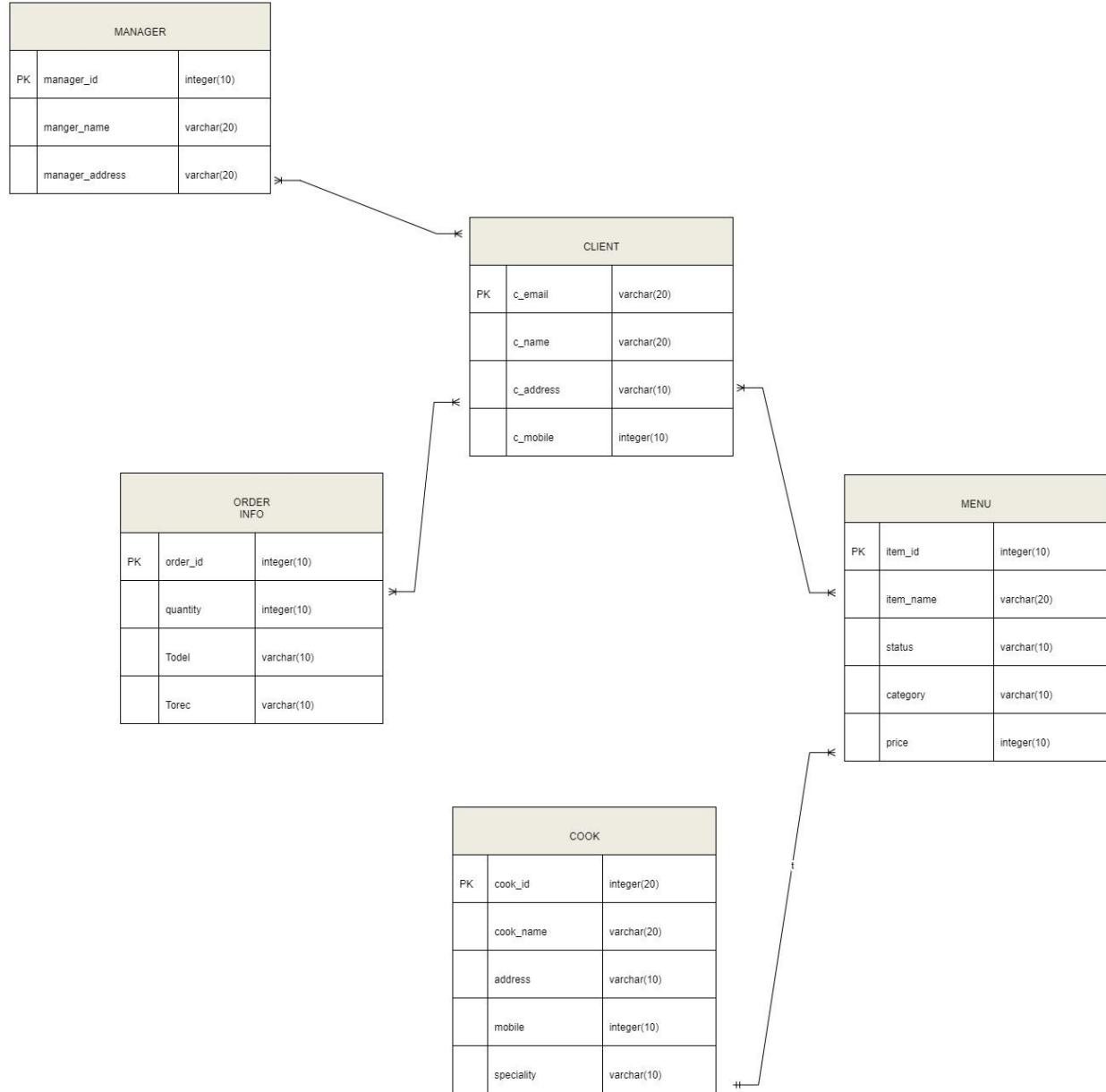
MENU		
PK	item_id	integer(10)
	item_name	varchar(20)
	status	varchar(10)
	category	varchar(10)
	price	integer(10)

COOK		
PK	cook_id	integer(20)
	cook_name	varchar(20)
	address	varchar(10)
	mobile	integer(10)
	speciality	varchar(10)

ORDER INFO		
PK	order_id	integer(10)
	quantity	integer(10)
	Todel	varchar(10)
	Torec	varchar(10)

CLIENT		
PK	c_email	varchar(20)
	c_name	varchar(20)
	c_address	varchar(10)
	c_mobile	integer(10)

## ENTITY RELATIONSHIP DIAGRAM:



- \*One manager can handle many clients
- \*one client can be handled by many managers
- \*one cook can prepare many menu
- \*one menu must need atleast one or many cook
- \*one client can have many menus
- \*one menu can be given to many clients
- \*one client can place many orders
- \*one order can be placed by many clients

## **NORMALISATION:**

NORMALISATION can be defined as the sequential process of transforming the unnormalised relation into relation with the simpler structures. Normalisation works through a series of stages called normal forms.

There is the need for normalization because while working with the relational database theory, unnormalised relation exhibits certain problems whenever updation was made to it. These problems are referred to as anomalies. So, in order to tackle with these issues normalization is done.

Let us discuss the normal forms in detail . the following information provides the conditions which need to be satisfied by the relation to be in that certain normal form.

### **FIRST NORMAL FORM(1NF):**

- \*No repeating rows i.e. primary key is defined.
- \*no multivalued columns
- \*all the values must be atomic i.e. at the intersection of each row and column there should be only one value.

### SECOND NORMAL FORM(2NF):

- \*It should be in first normal form
- \*Every non key attribute(which don't form the primary key) should be fully functionally dependent on the primary key .  
In other words, there should not be any partial dependencies.

### THIRD NORMAL FORM(3NF):

- \*It should be in second normal form.
- \*Non-Key attributes of relation should not be transitively functionally dependent on the primary key. Transitively functionally dependent means that a certain attribute is dependent on the attribute other than the primary key for its value which is not allowed.

Consider THE FOLLOWING EXAMPLE WHICH CONTAINS THE ATTRIBUTES OF VARIOUES ENTITIES:

c_address	item_name	category	quantity	price	item_id	c_email	order_id
21 street willingston,Bk	burger	fastfood	1	10	12	<a href="mailto:aman@gmail">aman@gmail</a>	1
32 street crist,ON	PIZZA	fastfood	2	40	13	<a href="mailto:devin@hotmail">devin@hotmail</a>	2
12 sec ,MN	salad	organic	1	30	11	<a href="mailto:kev11@gmail">kev11@gmail</a>	3
34 sector,VC	pasta	fastfood	2	40	16	<a href="mailto:romil12@yahoo">romil12@yahoo</a>	4

LET US NOW DO THE NORMALISATION:

### 1 NORMAL FORM:

Since the values are atomic . We just need to provide a primary key to make the data in first normal form.We will specify order\_id as the primary key.

c_address	item_name	category	quantity	price	item_id	c_email	order_id
21 street willingston,Bk	burger	fastfood	1	10	12	<a href="mailto:aman@gmail">aman@gmail</a>	1
32 street crist,ON	PIZZA	fastfood	2	40	13	<a href="mailto:devin@hotmail">devin@hotmail</a>	2
12 sec ,MN	salad	organic	1	30	11	<a href="mailto:kev11@gmail">kev11@gmail</a>	3
34 sector,VC	pasta	fastfood	2	40	16	<a href="mailto:romil12@yahoo">romil12@yahoo</a>	4

Thus, the given relation is in first normal form.

### 2 NORMAL FORM:

In this normal form we are going to remove the partial dependencies.

quantity	item_id	c_email	order_id
1	12	aman@gmail	1
2	13	devin@hotmail	2
1	11	kev11@gmail	3
2	16	romil12@yahoo	4

## CLIENT information

c_email	c_address
aman@gmail	21 street willingston,Bk
devin@hotmail	32 street crist,ON
kev11@gmail	12 sec ,MN
romil12@yahoo	34 sector,VC

## ITEM information

item_id	item_name	category	price
12	burger	fastfood	10
13	PIZZA	fastfood	40
11	salad	organic	30
16	pasta	fastfood	40

Here, item\_id and c\_email acts as the primary key in the tables CLIENT information and ITEM information.Denoted by yellow color.

Item\_id and c\_email acts as the foreign key in the main table.Denoted by green color.

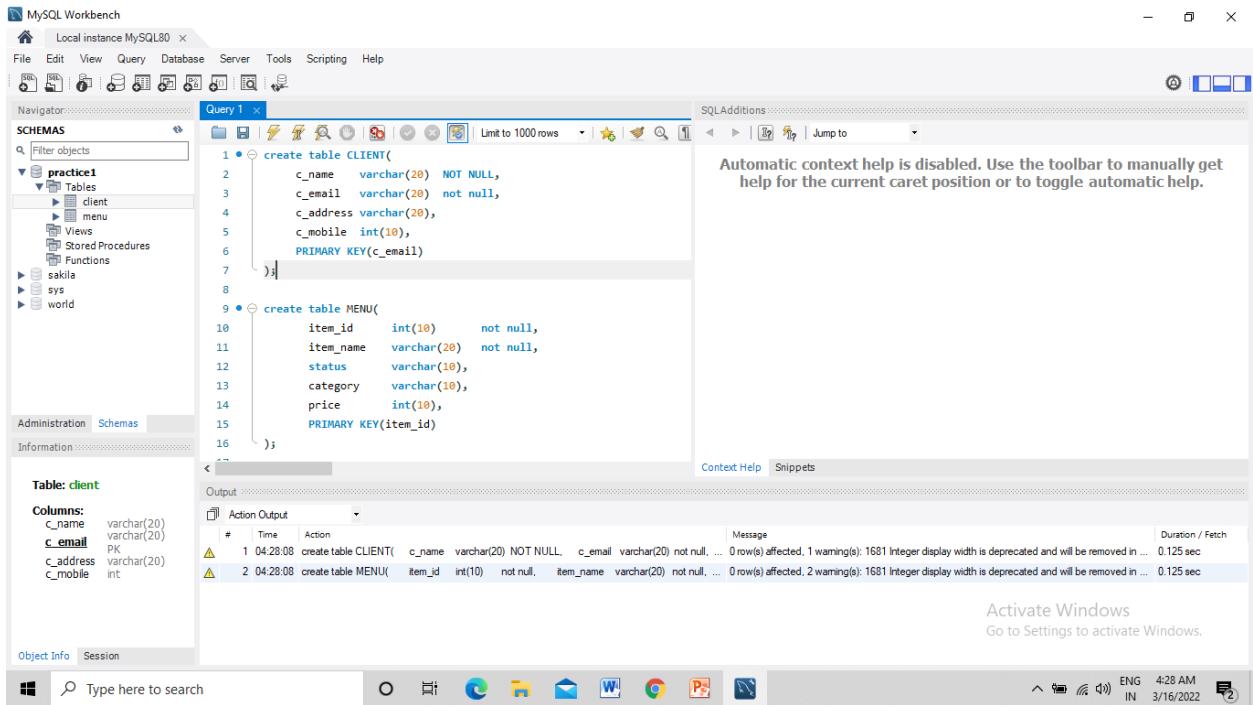
## 3 NORMAL FORM:

IN this normal form, the transitive dependencies is removed. However, there is not such attribute in the given scenario. So, It is in third normal form already.

# QUERIES AND CONSTRAINTS APPLICATION IN SERVER:

Let us first create the table and apply some of the constraints on it.

## PRIMARY KEY CONSTRAINT:



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

- Navigator:** Shows the database schema with the `practice1` database selected, containing the `client` and `menu` tables.
- Query 1:** Displays the SQL code for creating the `CLIENT` and `MENU` tables.

```
1 • 1 create table CLIENT(
2     c_name    varchar(20)  NOT NULL,
3     c_email   varchar(20)  not null,
4     c_address varchar(20),
5     c_mobile  int(10),
6     PRIMARY KEY(c_email)
7 )
8
9 • 2 create table MENU(
10    item_id   int(10)      not null,
11    item_name varchar(20)  not null,
12    status    varchar(10),
13    category  varchar(10),
14    price     int(10),
15    PRIMARY KEY(item_id)
16 );
```
- Output:** Shows the execution results with two warnings:

#	Time	Action	Message	Duration / Fetch
1	04:28:08	create table CLIENT( c_name varchar(20) NOT NULL, c_email varchar(20) not null, ... )	0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed in ...	0.125 sec
2	04:28:08	create table MENU( item_id int(10) not null, item_name varchar(20) not null, ... )	0 row(s) affected, 2 warning(s): 1681 Integer display width is deprecated and will be removed in ...	0.125 sec

# FOREIGN KEY CONSTRAINT:

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

- File Menu:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** Standard MySQL Workbench toolbar icons.
- Navigator:** Shows the schema `practice1` containing tables `client`, `main`, and `menu`.
- Query Editor (Query 1):** Contains the SQL code for creating the `main` table:

```
1 • 1 create table main(
2     order_id int(10) NOT NULL,
3     c_email varchar(20),
4     item_id int(10),
5     quantity int(10),
6     PRIMARY KEY (order_id),
7     FOREIGN KEY (c_email) References CLIENT(c_email),
8     FOREIGN KEY (item_id) References MENU(item_id)
9 )
10
11
12
```
- Output Window:** Shows the execution log with three warning messages:

#	Time	Action	Message	Duration / Fetch
1	04:28:08	create table CLIENT( c_name varchar(20) NOT NULL, c_email varchar(20) not null, ... )	0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed in ...	0.125 sec
2	04:28:08	create table MENU( item_id int(10) not null, item_name varchar(20) not null, ... )	0 row(s) affected, 2 warning(s): 1681 Integer display width is deprecated and will be removed in ...	0.125 sec
3	04:37:47	create table main( order_id int(10) NOT NULL, c_email varchar(20), item_id int(10), quantity int(10), PRIMARY KEY (order_id), FOREIGN KEY (c_email) References CLIENT(c_email), FOREIGN KEY (item_id) References MENU(item_id) )	0 row(s) affected, 3 warning(s): 1681 Integer display width is deprecated and will be removed in ...	0.281 sec
- Status Bar:** Shows "Activate Windows" and "Go to Settings to activate Windows.", the system language as "ENG IN", the date and time as "3/16/2022 4:38 AM", and a session identifier "(2)".

# UNIQUE KEY CONSTRAINT:

The screenshot shows the MySQL Workbench interface. In the Navigator pane, the schema 'practice1' is selected, and the 'client' table is highlighted. The Query 1 editor contains the following SQL code:

```
1 • Alter table CLIENT
2 ADD CONSTRAINT CLIENT_unique
3 UNIQUE (c_name);
4
5
6
```

The Output pane displays the execution log:

#	Time	Action	Message	Duration / Fetch
1	04:28:08	create table CLIENT( c_name varchar(20) NOT NULL, c_email varchar(20) not null, ... )	0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed in ...	0.125 sec
2	04:28:08	create table MENU( item_id int(10) not null, item_name varchar(20) not null, ... )	0 row(s) affected, 2 warning(s): 1681 Integer display width is deprecated and will be removed in ...	0.125 sec
3	04:37:47	create table main( order_id int(10) NOT NULL, c_email varchar(20), item_id int(10), ... )	0 row(s) affected, 3 warning(s): 1681 Integer display width is deprecated and will be removed in ...	0.281 sec
4	04:46:14	Alter table CLIENT ADD CONSTRAINT CLIENT_unique UNIQUE (c_name)	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to ...	0.000 sec
5	04:47:02	Alter table CLIENT ADD CONSTRAINT CLIENT_unique UNIQUE (c_name)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 Go to Settings to activate W	0.110 sec

The status bar at the bottom right shows the date and time as 3/16/2022 4:47 AM.

# NOT NULL CONSTRAINT

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

Schemas

practice1

Tables

client

main

menu

Views

Stored Procedures

Functions

sakila

sys

world

Query 1

```
1 • Alter table CLIENT
2   MODIFY c_mobile int NOT NULL;
3
4
5
```

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Administration Schemas

Information

Table: client

Columns:

c_name	varchar(20)
c_email	varchar(20)
c_address	varchar(20)
c_mobile	int

Action Output

#	Time	Action	Message	Duration / Fetch
1	04:28:08	create table CLIENT( c_name varchar(20) NOT NULL, c_email varchar(20) not null, ... )	0 rows(a) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed in ...	0.125 sec
2	04:28:08	create table MENU( item_id int(10) not null, item_name varchar(20) not null, ... )	0 row(s) affected, 2 warning(s): 1681 Integer display width is deprecated and will be removed in ...	0.125 sec
3	04:37:47	create table main( order_id int(10) NOT NULL, c_email varchar(20), item_id int(10), ... )	0 row(s) affected, 3 warning(s): 1681 Integer display width is deprecated and will be removed in ...	0.281 sec
4	04:47:14	Alter table CLIENT ADD CONSTRAINT CLIENT_unique UNIQUE (c_name)	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to ...	0.000 sec
5	04:47:02	Alter table CLIENT ADD CONSTRAINT CLIENT_unique UNIQUE (c_name)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.110 sec
6	04:50:20	Alter table CLIENT MODIFY c_mobile int NOT NULL	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.323 sec

Object Info Session

Type here to search

ENG IN 4:50 AM 3/16/2022

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

Schemas

practice1

Tables

client

main

menu

Views

Stored Procedures

Functions

sakila

sys

world

Query 1

```
1 • SELECT * FROM client
```

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Administration Schemas

Information

Table: client

Columns:

c_name	c_email	c_address	c_mobile
aman	aman@gmail.com	x	3456
devin	devin@hotmail.com	y	5678
kev	kev11@gmail.com	z	2345
romil	romil12@gmail.com	r	12334
*	HULL	HULL	HULL

Result Grid

client 1

Apply Revert

Context Help Snippets

Action Output

#	Time	Action	Message	Duration / Fetch
1	05:53:01	SELECT * FROM client LIMIT 0, 1000	4 row(s) returned	0.015 sec / 0.000 sec

Activate Windows  
Go to Settings to activate Windows.

# SECTION A

## SELECT STATEMENT:

The screenshot shows the MySQL Workbench interface. In the top-left corner, the title bar reads "MySQL Workbench Local instance MySQL80". The menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The toolbar contains various icons for database management tasks.

The left sidebar is the Navigator, showing the schema "practice1" which contains tables, views, stored procedures, and functions. Other schemas listed are sakila, sys, and world.

The main area is the "Query 1" editor, which displays the following SQL code:

```
1 create table Menu
2   ( Item_ID int(10) primary key,
3     Item_name varchar(20),
4     Category varchar(10),
5     Status varchar(10),
6     price int(10)
7 );
8 • Insert into Menu values (11,'salad','organic','available',30);
9 • Insert into Menu values (12,'burger','fastfood','available',10);
10 • Insert into Menu values (13,'pizza','fastfood','available',40);
11 • Insert into Menu values (14,'fruits','organic','available',50);
12 • Insert into Menu values (15,'sandwich','fastfood','available',15);
13 • Insert into Menu values (16,'pasta','fastfood','available',10);
14 • Select * from Menu
15
```

The right side of the interface has a "SQLAdditions" panel with a note: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help." Below this is the "Output" pane, which shows the result of a previous query:

#	Time	Action
1	21:46:00	DROP TABLE 'practice1'.employee_c0854624'

Message: 0 row(s) affected Duration / Fetch: 0.140 sec

At the bottom right, there is a message: "Activate Windows Go to Settings to activate Windows." and a "Query Completed" status indicator.

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS practice1

Query 1 × practice1.employee\_c0854624

```

9 • Insert into Menu values (12,'burger','fastfood','available',10);
10 • Insert into Menu values (13,'pizza','fastfood','available',40);
11 • Insert into Menu values (14,'fruits','organic','available',50);
12 • Insert into Menu values (15,'sandwich','fastfood','available',15);
13 • Insert into Menu values (16,'pasta','fastfood','available',10);
14 • Select * from Menu
15

```

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

Item_ID	Item_name	Category	Status	price
11	salad	organic	available	30
12	burger	fastfood	available	10
13	pizza	fastfood	available	40
14	fruits	organic	available	50
15	sandwich	fastfood	available	15
16	pasta	fastfood	available	10

Administration Schemas

Information: Schema: practice1

Action Output

#	Time	Action	Message	Duration / Fetch
4	22:06:26	Insert into Menu values (12,'burger','fastfood','available',10)	1 row(s) affected	0.016 sec
5	22:06:26	Insert into Menu values (13,'pizza','fastfood','available',40)	1 row(s) affected	0.016 sec
6	22:06:26	Insert into Menu values (14,'fruits','organic','available',50)	1 row(s) affected	0.015 sec
7	22:06:26	Insert into Menu values (15,'sandwich','fastfood','available',15)	1 row(s) affected	0.016 sec
8	22:06:26	Insert into Menu values (16,'pasta','fastfood','available',10)	1 row(s) affected	0.015 sec
9	22:06:26	Select * from Menu LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Query Completed

select Item\_ID ,Item\_name,Category from menu

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS practice1

Query 1 × practice1.employee\_c0854624

```

1
2 • select Item_ID ,Item_name,Category from menu
3

```

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

Item_ID	Item_name	Category
11	salad	organic
12	burger	fastfood
13	pizza	fastfood
14	fruits	organic
15	sandwich	fastfood
16	pasta	fastfood

Administration Schemas

Information: Schema: practice1

Action Output

#	Time	Action	Message	Duration / Fetch
5	22:06:26	Insert into Menu values (13,'pizza','fastfood','available',40)	1 row(s) affected	0.016 sec
6	22:06:26	Insert into Menu values (14,'fruits','organic','available',50)	1 row(s) affected	0.016 sec
7	22:06:26	Insert into Menu values (15,'sandwich','fastfood','available',15)	1 row(s) affected	0.015 sec
8	22:06:26	Insert into Menu values (16,'pasta','fastfood','available',10)	1 row(s) affected	0.015 sec
9	22:06:26	Select * from Menu LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
10	22:08:42	select Item_ID ,Item_name,Category from menu LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Query Completed

SELECT Item\_ID, Item\_Name, status, Category, Price\*3 from Menu

The screenshot shows the MySQL Workbench interface. The top navigation bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. Below the navigation bar is a toolbar with various icons. The main area is divided into sections: Navigator, Query Editor, and Result Grid.

**Navigator:** Shows the database schema. Under the 'practise1' schema, there is a 'Tables' section containing 'menu'. Other schemas listed are 'sakila', 'sys', and 'world'.

**Query Editor:** The tab is labeled 'Query 1' and contains the SQL query: `SELECT Item_ID, Item_Name, status, Category, Price*3 from Menu`. The results of the query are displayed in the Result Grid.

**Result Grid:** The grid has columns: Item\_ID, Item\_Name, status, Category, and Price\*3. The data is as follows:

	Item_ID	Item_Name	status	Category	Price*3
▶	11	salad	available	organic	90
	12	burger	available	fastfood	30
	13	pizza	available	fastfood	120
	14	fruits	available	organic	150
	15	sandwich	available	fastfood	45
	16	pasta	available	fastfood	30

SELECT Item\_ID+' '+ Item\_Name, status, Category, Price from Menu

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

practice1

Tables

menu

Views

Stored Procedures

Functions

sakila

sys

world

Query 1 x

1

2 • SELECT Item\_ID+' '+ Item\_Name, status, Category, Price from Menu

3

4

5

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

	Item_ID+' '+ Item_Name	status	Category	Price
11		available	organic	30
12		available	fastfood	10
13		available	fastfood	40
14		available	organic	50
15		available	fastfood	15

Administration Schemas

Information Result 4 x

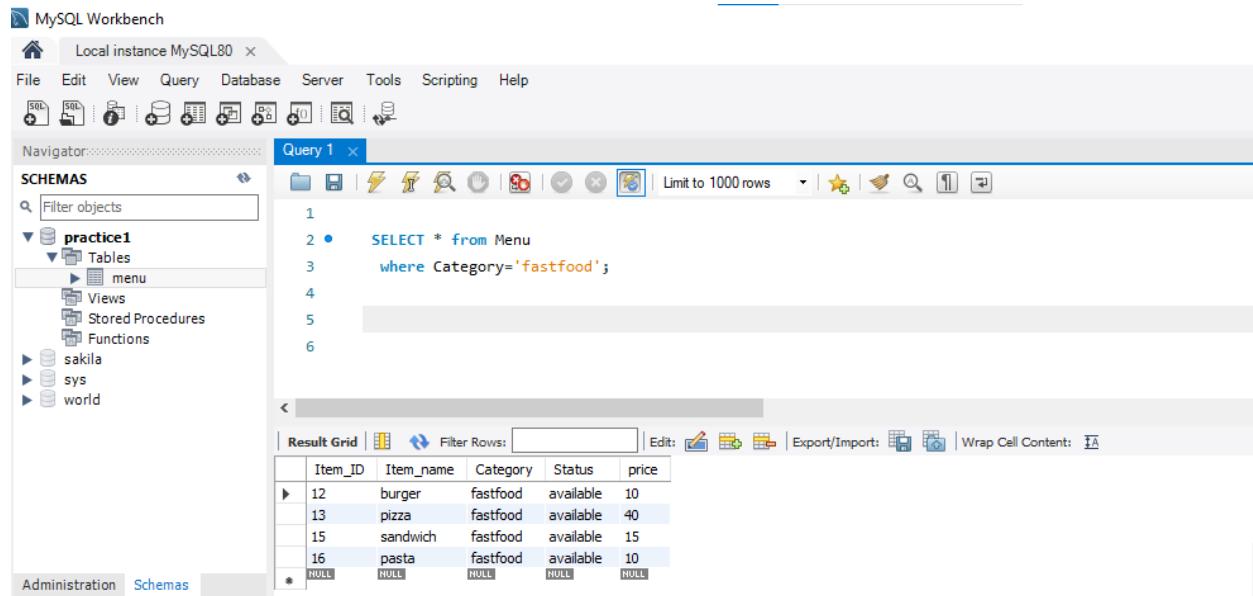
Result Grid | Filter Rows: [ ] Export: [ ] Wrap Cell Content: [ ]

	Item_ID+' '+ Item_Name	status	Category	Price
12		available	fastfood	10
13		available	fastfood	40
14		available	organic	50
15		available	fastfood	15
16		available	fastfood	10

Result 4 x

SELECT \* from Menu

where Category='fastfood';



The screenshot shows the MySQL Workbench interface. In the Navigator pane, under the 'practise1' schema, the 'menu' table is selected. In the Query Editor (Query 1), the following SQL query is run:

```
1
2 •  SELECT * from Menu
3      where Category='fastfood';
4
5
6
```

The Result Grid displays the following data:

Item_ID	Item_name	Category	Status	price
12	burger	fastfood	available	10
13	pizza	fastfood	available	40
15	sandwich	fastfood	available	15
16	pasta	fastfood	available	10
*	NULL	NULL	NULL	NULL

## BETWEEN (AND):

SELECT \* FROM Menu

where Price between '5' AND '50';

The screenshot shows the MySQL Workbench interface. In the Navigator pane, under the 'practice1' schema, the 'menu' table is selected. In the Query Editor (Query 1), the following SQL query is run:

```
1
2 •   SELECT * FROM Menu
3     where Price between '5' AND '50';
4
5
6
```

The Result Grid displays the following data:

	Item_ID	Item_name	Category	Status	price
▶	11	salad	organic	available	30
	12	burger	fastfood	available	10
	13	pizza	fastfood	available	40
	14	fruits	organic	available	50
	15	sandwich	fastfood	available	15
	16	pasta	fastfood	available	10

## IN statement:

SELECT \* FROM Menu

where Item\_name IN ('salad','sandwich','pasta');

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. The left sidebar, titled 'Navigator', shows the database schema with 'practise1' selected, containing 'Tables' (menu), 'Views', 'Stored Procedures', and 'Functions'. Other databases listed are 'sakila', 'sys', and 'world'. The main area is titled 'Query 1' and contains the following SQL code:

```
1
2 •   SELECT * FROM Menu
3     where Item_name IN ('salad','sandwich','pasta');
4
5
6
```

Below the code is a 'Result Grid' table with the following data:

	Item_ID	Item_name	Category	Status	price
▶	11	salad	organic	available	30
▶	15	sandwich	fastfood	available	15
▶	16	pasta	fastfood	available	10
*	NULL	NULL	NULL	NULL	NULL

## LIKE statement:

SELECT \* FROM Menu

where category like 'fastfood' ;

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Schemas Navigator Query 1

**Query 1**

```
1
2 •   SELECT * FROM Menu
3   where category like 'fastfood' ;
4
5
6
```

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

	Item_ID	Item_name	Category	Status	price
▶	12	burger	fastfood	available	10
	13	pizza	fastfood	available	40
	15	sandwich	fastfood	available	15
	16	pasta	fastfood	available	10
*	NULL	NULL	NULL	NULL	NULL

## SECTION B

## LOGICAL OPERATOR:

{AND}

Query 1

```
7      );
8 •  insert into cook values(1,'rahul','ontario',675844,'indian');
9 •  insert into cook values(2,'zen','ontario',575795,'italian');
10 • insert into cook values(3,'devid','winnipeg',648368,'indian');
11 • insert into cook values(4,'george','toronto',463635,'chinese');
12 • insert into cook values(5,'florida','california',674578,'seafood');
13 • select* from cook
```

Result Grid

	cook_id	cook_name	address	mobile	speciality
▶	1	rahul	ontario	675844	indian
	2	zen	ontario	575795	italian
	3	devid	winnipeg	648368	indian
	4	george	toronto	463635	chinese
*	5	florida	california	674578	seafood
	NULL	NULL	NULL	NULL	NULL

cook 11

select\* from cook

where speciality='indian' and address ='ontario';

Query 1

```
1
2 •  select* from cook
3   where speciality='indian' and address = 'ontario';
4
```

Result Grid

	cook_id	cook_name	address	mobile	speciality
▶	1	rahul	ontario	675844	indian
*	NULL	NULL	NULL	NULL	NULL

cook 12

{OR}

`select* from cook`

`where speciality='indian' or address ='ontario';`

The screenshot shows the MySQL Workbench interface. In the 'Query 1' editor, the following SQL code is written:

```
1
2 • select* from cook
3 where speciality='indian' or address ='ontario';
4
```

The results are displayed in the 'Result Grid' tab, showing the following data:

cook_id	cook_name	address	mobile	specialty
1	rahul	ontario	675844	indian
2	zen	ontario	575795	italian
3	devid	winnipeg	648368	indian
*	NULL	NULL	NULL	NULL

In the 'Output' pane, the following log entries are shown:

#	Time	Action	Message	Duration / Fetch
32	23:23:49	insert into cook values(3,devid,'winnipeg',648368,'indian')	1 row(s) affected	0.016 sec
33	23:23:49	insert into cook values(4,'george','toronto',463635,'chinese')	1 row(s) affected	0.016 sec
34	23:23:49	insert into cook values(5,'florida','california',674578,'seafood')	1 row(s) affected	0.000 sec / 0.000 sec
35	23:23:49	select * from cook LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
36	23:29:26	select * from cook where speciality='indian' and address ='ontario' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
37	23:31:12	select * from cook where speciality='indian' and address ='ontario' LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

{NOT}

`select* from cook`

`where address NOT like 'toronto';`

MySQL Workbench - Local instance MySQL80

**Query 1**

```

1
2 • select* from cook
3 where address NOT like 'toronto';
4

```

**Result Grid**

cook_id	cook_name	address	mobile	specialty
1	rahul	ontario	675844	indian
2	zen	ontario	575795	italian
3	devid	winnipeg	648368	indian
5	florida	california	674578	seafood
*	NULL	NULL	NULL	NULL

## Combined statement:

`select* from cook`

`where address ='toronto' or address='ontario' and cook_id=1 or cook_id=2;`

MySQL Workbench - Local instance MySQL80

**Query 1**

```

1
2 • select* from cook
3 where address ='toronto' or address='ontario' and cook_id=1 or cook_id=2;
4

```

**Result Grid**

cook_id	cook_name	address	mobile	specialty
1	rahul	ontario	675844	indian
2	zen	ontario	575795	italian
4	george	toronto	463635	chinese
*	NULL	NULL	NULL	NULL

**Output**

Action Output

#	Time	Action	Message	Duration / Fetch
34	23:23:49	insert into cook values(5,'florida','california',674578,'seafood')	1 row(s) affected	0.000 sec
35	23:23:49	select* from cook LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
36	23:29:26	select* from cook where specialty='indian' and address ='ontario' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
37	23:31:12	select* from cook where specialty='indian' or address ='ontario' LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
38	23:34:15	select* from cook where address NOT like 'toronto' LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
39	23:41:56	select* from cook where address ='toronto' or address='ontario' and cook_id=1 or cook_id=2 ...	3 row(s) returned	0.000 sec / 0.000 sec

{ANY}:

select\* from cook

where cook\_id > any(select cook\_id from cook

where cook\_id > '1');

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

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** Includes icons for SQL, DDL, Scripts, Tables, Schemas, and other database management functions.
- Navigator:** Shows the schema structure under the 'practise1' database, including 'Tables' (cook, menu), 'Views', 'Stored Procedures', and 'Functions'. Other databases listed are 'sakila', 'sys', and 'world'.
- Query Editor:** Titled 'Query 1', contains the following SQL code:

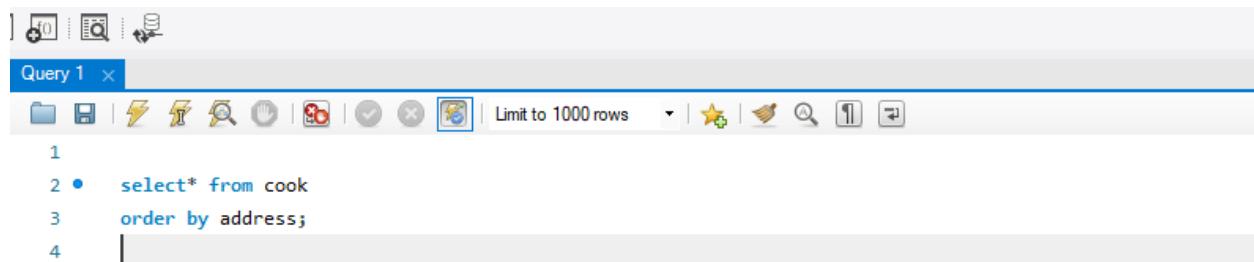
```
1
2 •  select* from cook
3   where cook_id> any(select cook_id from cook
4   where cook_id> '1');
5
```
- Result Grid:** Displays the results of the query in a tabular format with columns: cook\_id, cook\_name, address, mobile, and speciality. The data is as follows:

	cook_id	cook_name	address	mobile	speciality
3	3	devid	winnipeg	648368	indian
4	4	george	toronto	463635	chinese
5	5	florida	california	674578	seafood
*	NULL	NULL	NULL	NULL	NULL

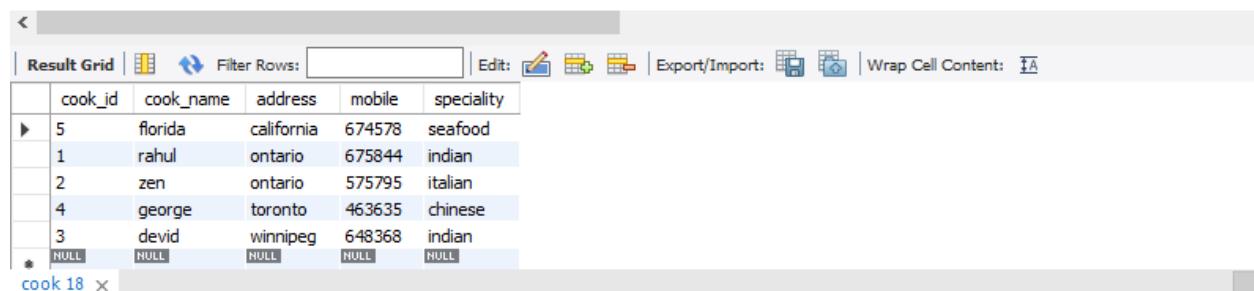
## ORDER BY:

select\* from cook

order by address;



```
1
2 •  select* from cook
3   order by address;
4
```



	cook_id	cook_name	address	mobile	speciality
▶	5	florida	california	674578	seafood
	1	rahul	ontario	675844	indian
	2	zen	ontario	575795	italian
	4	george	toronto	463635	chinese
*	3	devid	winnipeg	648368	indian
*	HULL	HULL	HULL	HULL	HULL

select\* from cook

order by cook\_name;

The screenshot shows the MySQL Workbench interface. In the top navigation bar, 'File', 'Edit', 'View', 'Query', 'Database', 'Server', 'Tools', and 'Help' are visible. Below the menu is a toolbar with various icons. The 'Navigator' pane on the left lists 'SCHEMAS' (practice1, sakila, sys, world) and 'Tables' (cook, menu). The 'Query 1' tab contains the following SQL code:

```
1
2 • select* from cook
3   order by cook_name;
4
```

The 'Result Grid' pane displays the results of the query:

cook_id	cook_name	address	mobile	speciality
3	devid	winnipeg	648368	indian
5	florida	california	674578	seafood
4	george	toronto	463635	chinese
1	rahul	ontario	675844	indian
2	zen	ontario	575795	italian
*	NULL	NULL	NULL	NULL

## STRING FUNCTION(LENGTH):

select LENGTH(address) AS LENGTHOFADDRESS

FROM COOK;

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Schemas

Filter objects

practice1

Tables

cook

menu

Views

Stored Procedures

Functions

sakila

sys

world

Administration Schemas

Query 1

1

2 • select LENGTH(ADDRESS) AS LENGTHOFADDRESS

3 FROM COOK;

4

5

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

LENGTHOFADDRESS
7
7
8
7
10

## STRING FUNCTION(REPLACE):

select REPLACE ('ONTARIO','ONT','M');

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Schemas

Filter objects

practice1

Tables

cook

menu

Views

Stored Procedures

Functions

sakila

sys

world

Administration Schemas

Query 1

1

2 • select REPLACE ('ONTARIO','ONT','M');

3

4

5

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

REPLACE (ONTARIO,'ONT','M')
MARIO

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Schemas

Filter objects

practise1

Tables

cook

menu

Views

Stored Procedures

Functions

sakila

sys

world

Navigator:.....

Query 1 ×

1 • UPDATE COOK

2 SET ADDRESS= REPLACE ('ONTARIO','ONT','M');

3 • SELECT \*FROM COOK

4

5

6

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

	cook_id	cook_name	address	mobile	speciality
▶	1	rahul	MARIO	675844	indian
	2	zen	MARIO	575795	italian
	3	devid	MARIO	648368	indian
	4	george	MARIO	463635	chinese
	5	florida	MARIO	674578	seafood
*	NULL	NULL	NULL	NULL	NULL

Administration Schemas

## SECTION C

### AGGREGATE FUNCTIONS:

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

	Item_ID	Item_name	Category	Status	price
▶	11	salad	organic	available	30
	12	burger	fastfood	available	10
	13	pizza	fastfood	available	40
	14	fruits	organic	available	50
	15	sandwich	fastfood	available	15
	16	pasta	fastfood	available	10

## AGGREGATE FUNCTION(AVG):

SELECT AVG(PRICE)

FROM MENU;

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

practice1

- Tables
  - cook
  - menu
- Views
- Stored Procedures
- Functions

sakila

sys

world

Query 1

```

1 •  SELECT AVG(PRICE)
2   FROM MENU;
3
4
5
6
7
  
```

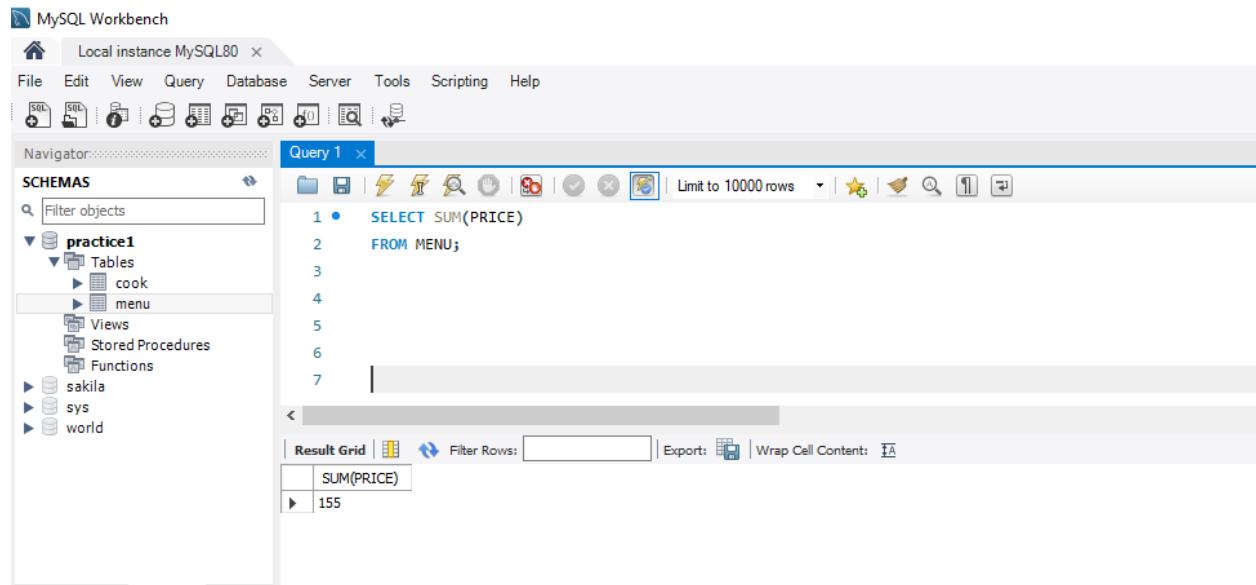
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

AVG(PRICE)
25.8333

## AGGREGATE FUNCTION(SUM):

SELECT SUM(PRICE)

FROM MENU;



The screenshot shows the MySQL Workbench interface. The 'Query 1' tab contains the following SQL code:

```
1 •  SELECT SUM(PRICE)
2   FROM MENU;
3
4
5
6
7
```

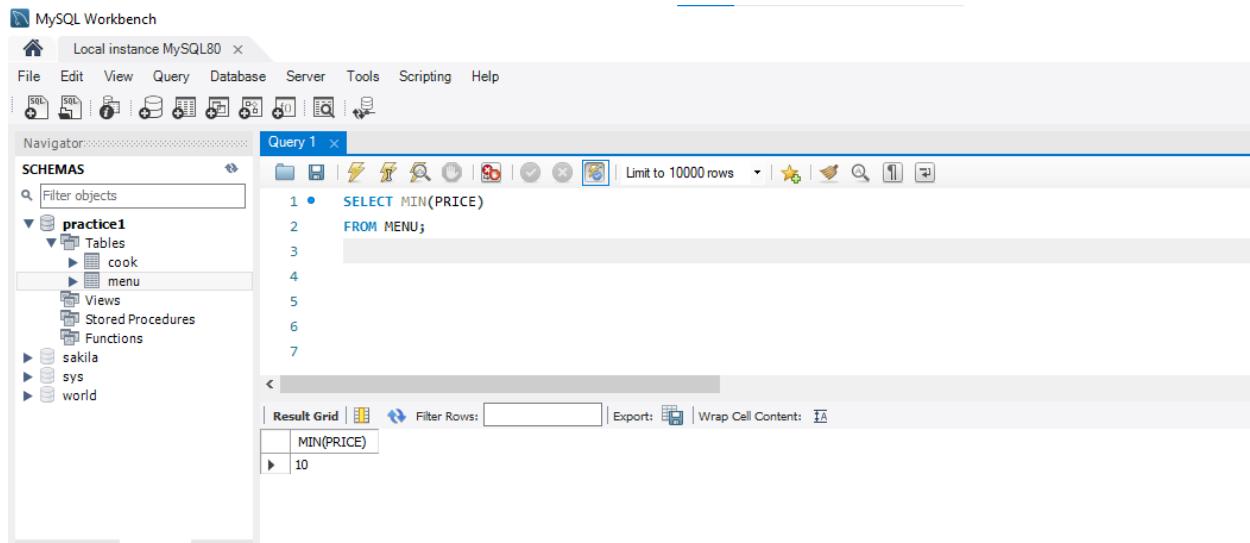
The 'Result Grid' shows the output of the query:

SUM(PRICE)
155

## AGGREGATE FUNCTION(MIN):

SELECT MIN(PRICE)

FROM MENU;



MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

SELECT MIN(PRICE)  
FROM MENU;

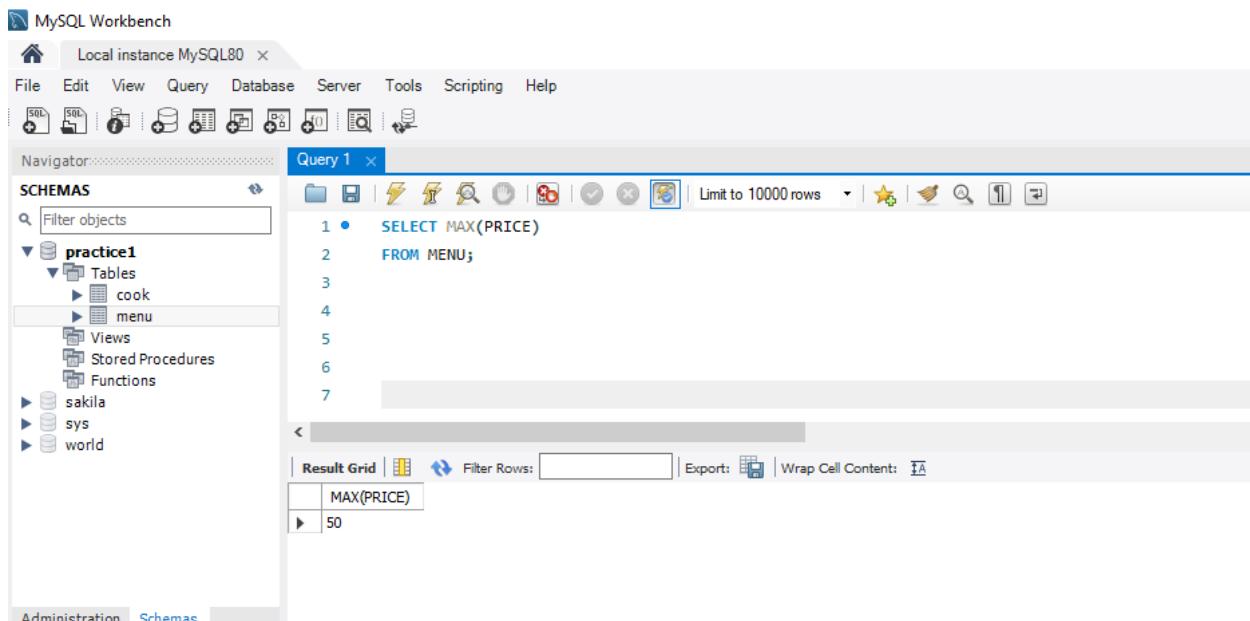
Result Grid | Filter Rows: Export: Wrap Cell Content:

MIN(PRICE)
10

## AGGREGATE FUNCTION(MAX):

SELECT MAX(PRICE)

FROM MENU;



MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

SELECT MAX(PRICE)  
FROM MENU;

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

MAX(PRICE)
50

## AGGREGATE FUNCTION(COUNT):

SELECT COUNT(PRICE)

FROM MENU;

The screenshot shows the MySQL Workbench interface. The top bar has tabs for 'Query 1' and 'SQL'. Below the tabs is a toolbar with various icons. The main area contains the following SQL code:

```
1 •  SELECT COUNT(PRICE)
2   FROM MENU;
3
4
5
6
7
```

Below the code is a results pane titled 'Result Grid' which displays the following data:

COUNT(PRICE)
6

SELECT COUNT(PRICE)

FROM MENU

WHERE PRICE>10;

Query 1

```
1 •  SELECT COUNT(PRICE)
2   FROM MENU
3 WHERE PRICE>10;
4
5
6
7
```

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

	COUNT(PRICE)
▶	4

## SECTION D

### GROUP BY:

```
SELECT COUNT(ITEM_ID),CATEGORY  
FROM MENU  
GROUP BY CATEGORY;
```

The screenshot shows the MySQL Workbench interface. In the top-left, the Navigator pane displays the database schema with a table named 'menu' under the 'practice1' schema. The Query 1 pane contains the following SQL code:

```
1 • SELECT COUNT(ITEM_ID),CATEGORY
2   FROM MENU
3   GROUP BY CATEGORY;
4
5
6
7
```

The Result Grid shows the output of the query:

COUNT(ITEM_ID)	CATEGORY
2	organic
4	fastfood

The Output pane at the bottom shows the execution log:

#	Time	Action	Message	Duration / Fetch
55	00:23:10	SELECT SUM(PRICE) FROM MENU LIMIT 0, 10000	1 row(s) returned	0.016 sec / 0.000 sec
56	00:25:10	SELECT MIN(PRICE) FROM MENU LIMIT 0, 10000	1 row(s) returned	0.000 sec / 0.000 sec
57	00:26:51	SELECT MAX(PRICE) FROM MENU LIMIT 0, 10000	1 row(s) returned	0.000 sec / 0.000 sec
58	00:30:10	SELECT COUNT(PRICE) FROM MENU LIMIT 0, 10000	1 row(s) returned	0.016 sec / 0.000 sec
59	00:31:29	SELECT COUNT(PRICE) FROM MENU WHERE PRICE<10 LIMIT 0, 10000	1 row(s) returned	0.000 sec / 0.000 sec
60	00:36:49	SELECT COUNT(ITEM_ID),CATEGORY FROM MENU GROUP BY CATEGORY LIMIT 0, 1...	2 row(s) returned	0.031 sec / 0.000 sec

The status bar at the bottom right indicates: 38°C Haze, ENG IN 12:37 AM 4/7/2022.

**SELECT COUNT(ITEM\_ID),STATUS**

**FROM MENU**

**GROUP BY STATUS;**

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Schemas Navigator: Query 1 ×

Filter objects

SCHEMAS

practice1

- Tables
- cook
- menu

Views

Stored Procedures

Functions

sakila

sys

world

Query 1

1 • SELECT COUNT(ITEM\_ID), STATUS  
2 FROM MENU  
3 GROUP BY STATUS;  
4  
5  
6  
7

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

COUNT(ITEM_ID)	STATUS
6	available

```
SELECT COUNT(ITEM_ID),PRICE  
FROM MENU  
GROUP BY PRICE;
```

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Schemas Navigator Query 1

**Query 1**

```
1 • SELECT COUNT(ITEM_ID),PRICE
2 FROM MENU
3 GROUP BY PRICE;
4
5
6
7
```

Result Grid Filter Rows: Export: Wrap Cell Content:

	COUNT(ITEM_ID)	PRICE
▶	1	30
	2	10
	1	40
	1	50
	1	15

Administration Schemas

The screenshot shows the MySQL Workbench interface. In the 'Query 1' editor, a SQL query is written to count items by price. The result grid displays five rows of data, each representing a group from the query. The columns are labeled 'COUNT(ITEM\_ID)' and 'PRICE'. The data shows item counts of 1, 2, 1, 1, and 1 for prices of 30, 10, 40, 50, and 15 respectively.

## GROUP BY HAVING:

```
SELECT COUNT(ITEM_ID),PRICE
FROM MENU
GROUP BY PRICE
HAVING PRICE>10;
```

The screenshot shows a MySQL Workbench interface. The query editor window is titled "Query 1" and contains the following SQL code:

```
1 •  SELECT COUNT(ITEM_ID),PRICE
2   FROM MENU
3   GROUP BY PRICE
4   HAVING PRICE>10;
5
6
7
```

The results grid displays the following data:

	COUNT(ITEM_ID)	PRICE
▶	1	30
	1	40
	1	50
	1	15

```
SELECT COUNT(ITEM_ID),PRICE
FROM MENU
GROUP BY PRICE
HAVING SUM(PRICE)>20;
```

The screenshot shows the MySQL Workbench interface. In the top left, it says "MySQL Workbench" and "Local instance MySQL80". The menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. Below the menu is a toolbar with various icons. On the left is the Navigator pane showing "SCHEMAS" with "practice1" selected, containing "Tables" (cook, menu), "Views", "Stored Procedures", and "Functions". Other schemas listed are sakila, sys, and world. The main area is titled "Query 1" and contains the following SQL code:

```
1 •  SELECT COUNT(ITEM_ID),PRICE
2   FROM MENU
3   GROUP BY PRICE
4   HAVING SUM(PRICE)>20;
5
6
7
```

Below the code is a "Result Grid" table with two columns: "COUNT(ITEM\_ID)" and "PRICE". The data shows three rows:

COUNT(ITEM_ID)	PRICE
1	30
1	40
1	50

```
SELECT COUNT(ITEM_ID),PRICE
FROM MENU
GROUP BY PRICE
HAVING PRICE>5
ORDER BY COUNT(ITEM_ID) DESC;
```

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

practice1

Tables

cook

menu

Views

Stored Procedures

Functions

sakila

sys

world

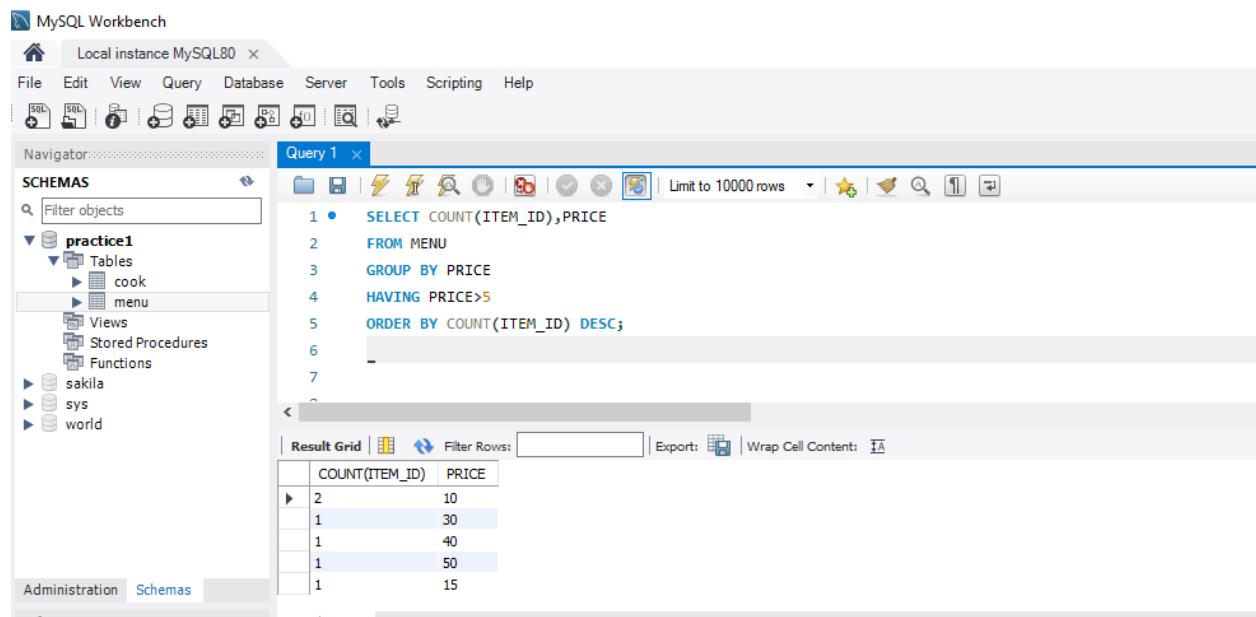
Query 1

1 • SELECT COUNT(ITEM\_ID),PRICE  
2 FROM MENU  
3 GROUP BY PRICE  
4 HAVING PRICE>5  
5 ORDER BY COUNT(ITEM\_ID) DESC;  
6  
7

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

COUNT(ITEM_ID)	PRICE
2	10
1	30
1	40
1	50
1	15

Administration Schemas

A screenshot of 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. The left sidebar shows the Navigator and Schemas sections, with 'practice1' selected, displaying its tables (cook, menu), views, stored procedures, and functions. The main area is titled 'Query 1' and contains a SQL query: 'SELECT COUNT(ITEM\_ID),PRICE FROM MENU GROUP BY PRICE HAVING PRICE>5 ORDER BY COUNT(ITEM\_ID) DESC;'. The results are displayed in a 'Result Grid' table with two columns: 'COUNT(ITEM\_ID)' and 'PRICE'. The data shows five rows: (2, 10), (1, 30), (1, 40), (1, 50), and (1, 15). The bottom navigation bar has tabs for Administration and Schemas.

# SECTION E

## JOINS:

	order_id	quantity	Torec	Todel	c_id
▶	1	2	2021-03-12	2021-03-12	2
	2	3	2021-03-13	2021-03-13	1
	3	1	2021-03-13	2021-03-13	2
	4	2	2021-04-14	2021-04-14	3
	5	3	2021-03-15	2021-03-15	4
	6	4	2021-03-23	2021-04-24	3
	7	1	2021-05-04	2021-05-04	5
*	NULL	NULL	NULL	NULL	NULL

	c_id	c_name	c_email	c_address	c_mobile
▶	1	rakesh	raks@gmail	montreal	345678
	2	aman	aman@gmail	brampton	567848
	3	devin	devin@hotmail	ontario	456546
	4	kevin	kev11@gmail	montreal	686897
	5	romil	romil12@yahoo	vancouver	567866
	6	zen	zen13@hotmail	brunswick	567778
*	NULL	NULL	NULL	NULL	NULL

## INNER JOIN:

```
select * from orders inner join client on orders.c_id=client.c_id;
```

Query 1 ×

38  
39 •     select \* from orders inner join client on orders.c\_id=client.c\_id;  
40  
41  
42

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

	order_id	quantity	Torec	Todel	c_id	c_id	c_name	c_email	c_address	c_mobile
▶	1	2	2021-03-12	2021-03-12	2	2	aman	aman@gmail	brampton	567848
	2	3	2021-03-13	2021-03-13	1	1	rakesh	raks@gmail	montreal	345678
	3	1	2021-03-13	2021-03-13	2	2	aman	aman@gmail	brampton	567848
	4	2	2021-04-14	2021-04-14	3	3	devin	devin@hotmail	ontario	456546
	5	3	2021-03-15	2021-03-15	4	4	kevin	kev11@gmail	montreal	686897
	6	4	2021-03-23	2021-04-24	3	3	devin	devin@hotmail	ontario	456546
	7	1	2021-05-04	2021-05-04	5	5	romil	romil12@yahoo	vancouver	567866

Result 13 ×

## LEFT JOIN:

select \* from orders left join client on orders.c\_id=client.c\_id;

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Schemas | Filter objects

▶ practice1

- Tables
  - client
  - cook
  - dept\_624
  - emp\_624
  - menu
  - orders
- Views
- Stored Procedures
- Functions

sakila

Information

Administration Schemas

Query 1 ×

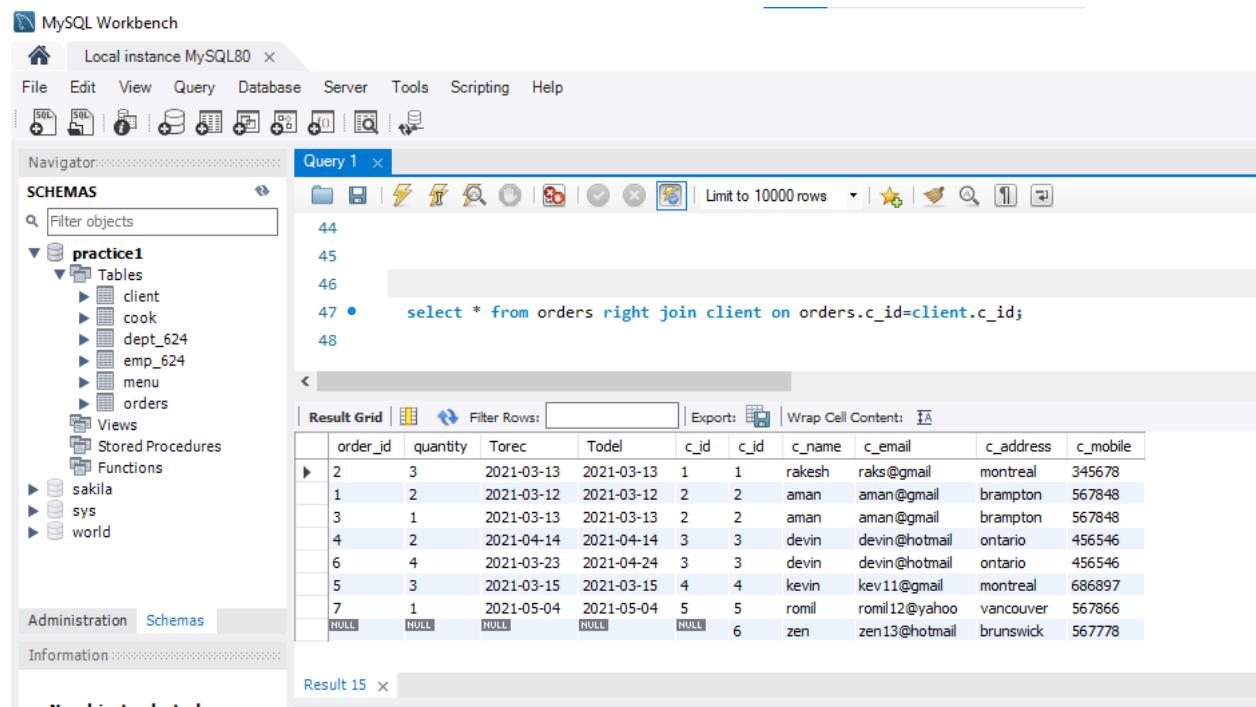
40  
41  
42  
43 •     select \* from orders left join client on orders.c\_id=client.c\_id;  
44

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

	order_id	quantity	Torec	Todel	c_id	c_id	c_name	c_email	c_address	c_mobile
▶	1	2	2021-03-12	2021-03-12	2	2	aman	aman@gmail	brampton	567848
	2	3	2021-03-13	2021-03-13	1	1	rakesh	raks@gmail	montreal	345678
	3	1	2021-03-13	2021-03-13	2	2	aman	aman@gmail	brampton	567848
	4	2	2021-04-14	2021-04-14	3	3	devin	devin@hotmail	ontario	456546
	5	3	2021-03-15	2021-03-15	4	4	kevin	kev11@gmail	montreal	686897
	6	4	2021-03-23	2021-04-24	3	3	devin	devin@hotmail	ontario	456546
	7	1	2021-05-04	2021-05-04	5	5	romil	romil12@yahoo	vancouver	567866

## RIGHT JOIN:

select \* from orders right join client on orders.c\_id=client.c\_id;



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

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Shows the schema **practice1** with tables: client, cook, dept\_624, emp\_624, menu, orders, sakila, sys, world.
- Query Editor:** Contains the following SQL code:

```
44
45
46
47 •    select * from orders right join client on orders.c_id=client.c_id;
48
```
- Result Grid:** Displays the results of the query. The columns are: order\_id, quantity, Torec, Todel, c\_id, c\_id, c\_name, c\_email, c\_address, c\_mobile. The data is as follows:

order_id	quantity	Torec	Todel	c_id	c_id	c_name	c_email	c_address	c_mobile
2	3	2021-03-13	2021-03-13	1	1	rakesh	raks@gmail	montreal	345678
1	2	2021-03-12	2021-03-12	2	2	aman	aman@gmail	brampton	567848
3	1	2021-03-13	2021-03-13	2	2	aman	aman@gmail	brampton	567848
4	2	2021-04-14	2021-04-14	3	3	devin	devin@hotmail	ontario	456546
6	4	2021-03-23	2021-04-24	3	3	devin	devin@hotmail	ontario	456546
5	3	2021-03-15	2021-03-15	4	4	kevin	kev11@gmail	montreal	686897
7	1	2021-05-04	2021-05-04	5	5	romil	romil12@yahoo	vancouver	567866
NULL	NULL	NULL	NULL	NULL	6	zen	zen13@hotmail	brunswick	567778

## FULL JOIN:

select \* from orders full join client ;

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Schemas

Tables

Views

Stored Procedures

Functions

sakila

sys

world

Query 1

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

	order_id	quantity	Torec	Todel	c_id	c_id	c_name	c_email	c_address	c_mobile
▶	1	2	2021-03-12	2021-03-12	2	6	zen	zen13@hotmail	brunswick	567778
	1	2	2021-03-12	2021-03-12	2	5	romil	romil12@yahoo	vancouver	567866
	1	2	2021-03-12	2021-03-12	2	4	kevin	kev11@gmail	montreal	686897
	1	2	2021-03-12	2021-03-12	2	3	devin	devin@hotmail	ontario	456546
	1	2	2021-03-12	2021-03-12	2	2	aman	aman@gmail	brampton	567848
	1	2	2021-03-12	2021-03-12	2	1	rakesh	raks@gmail	montreal	345678
	2	3	2021-03-13	2021-03-13	1	6	zen	zen13@hotmail	brunswick	567778
	2	3	2021-03-13	2021-03-13	1	5	romil	romil12@yahoo	vancouver	567866
	2	3	2021-03-13	2021-03-13	1	4	kevin	kev11@gmail	montreal	686897

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

	order_id	quantity	Torec	Todel	c_id	c_id	c_name	c_email	c_address	c_mobile
	2	3	2021-03-13	2021-03-13	1	5	romil	romil12@yahoo	vancouver	567866
	2	3	2021-03-13	2021-03-13	1	4	kevin	kev11@gmail	montreal	686897
	2	3	2021-03-13	2021-03-13	1	3	devin	devin@hotmail	ontario	456546
	2	3	2021-03-13	2021-03-13	1	2	aman	aman@gmail	brampton	567848
	2	3	2021-03-13	2021-03-13	1	1	rakesh	raks@gmail	montreal	345678
	3	1	2021-03-13	2021-03-13	2	6	zen	zen13@hotmail	brunswick	567778
	3	1	2021-03-13	2021-03-13	2	5	romil	romil12@yahoo	vancouver	567866
	3	1	2021-03-13	2021-03-13	2	4	kevin	kev11@gmail	montreal	686897
	3	1	2021-03-13	2021-03-13	2	3	devin	devin@hotmail	ontario	456546
	3	1	2021-03-13	2021-03-13	2	2	aman	aman@gmail	brampton	567848

Result 16

Query 1

```
49
50
51
52 •      select * from orders full join client ;
53
```

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

	order_id	quantity	Torec	Todel	c_id	c_id	c_name	c_email	c_address	c_mobile
	3	1	2021-03-13	2021-03-13	2	3	devin	devin@hotmail	ontario	456546
	3	1	2021-03-13	2021-03-13	2	2	aman	aman@gmail	brampton	567848
	3	1	2021-03-13	2021-03-13	2	1	rakesh	raks@gmail	montreal	345678
	4	2	2021-04-14	2021-04-14	3	6	zen	zen13@hotmail	brunswick	567778
	4	2	2021-04-14	2021-04-14	3	5	romil	romil12@yahoo	vancouver	567866
	4	2	2021-04-14	2021-04-14	3	4	kevin	kev11@gmail	montreal	686897
	4	2	2021-04-14	2021-04-14	3	3	devin	devin@hotmail	ontario	456546
	4	2	2021-04-14	2021-04-14	3	2	aman	aman@gmail	brampton	567848
	4	2	2021-04-14	2021-04-14	3	1	rakesh	raks@gmail	montreal	345678
	5	3	2021-03-15	2021-03-15	4	6	zen	zen13@hotmail	brunswick	567778

Result 16

Query 1

```
49
50
51
52 •      select * from orders full join client ;
53
```

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

	order_id	quantity	Torec	Todel	c_id	c_id	c_name	c_email	c_address	c_mobile
	5	3	2021-03-15	2021-03-15	4	6	zen	zen13@hotmail	brunswick	567778
	5	3	2021-03-15	2021-03-15	4	5	romil	romil12@yahoo	vancouver	567866
	5	3	2021-03-15	2021-03-15	4	4	kevin	kev11@gmail	montreal	686897
	5	3	2021-03-15	2021-03-15	4	3	devin	devin@hotmail	ontario	456546
	5	3	2021-03-15	2021-03-15	4	2	aman	aman@gmail	brampton	567848
	5	3	2021-03-15	2021-03-15	4	1	rakesh	raks@gmail	montreal	345678
	6	4	2021-03-23	2021-04-24	3	6	zen	zen13@hotmail	brunswick	567778
	6	4	2021-03-23	2021-04-24	3	5	romil	romil12@yahoo	vancouver	567866
	6	4	2021-03-23	2021-04-24	3	4	kevin	kev11@gmail	montreal	686897
	6	4	2021-03-23	2021-04-24	3	3	devin	devin@hotmail	ontario	456546

Result 16

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

```
49  
50  
51  
52 •     select * from orders full join client ;  
53
```

The results grid displays the following data:

	order_id	quantity	Torec	Todel	c_id	c_name	c_email	c_address	c_mobile
6	4	2021-03-23	2021-04-24	3	4	kevin	kev11@gmail	montreal	686897
6	4	2021-03-23	2021-04-24	3	3	devin	devin@hotmail	ontario	456546
6	4	2021-03-23	2021-04-24	3	2	aman	aman@gmail	brampton	567848
6	4	2021-03-23	2021-04-24	3	1	rakesh	raks@gmail	montreal	345678
7	1	2021-05-04	2021-05-04	5	6	zen	zen13@hotmail	brunswick	567778
7	1	2021-05-04	2021-05-04	5	5	romil	romil12@yahoo	vancouver	567866
7	1	2021-05-04	2021-05-04	5	4	kevin	kev11@gmail	montreal	686897
7	1	2021-05-04	2021-05-04	5	3	devin	devin@hotmail	ontario	456546
7	1	2021-05-04	2021-05-04	5	2	aman	aman@gmail	brampton	567848
7	1	2021-05-04	2021-05-04	5	1	rakesh	raks@gmail	montreal	345678

Result 16 ×

## SECTION F

### SUBQUERIES:

Result Grid | Filter Rows: | Edit: | Export

c_id	c_name	c_email	c_address	c_mobile
1	rakesh	raks@gmail	montreal	345678
2	aman	aman@gmail	brampton	567848
3	devin	devin@hotmail	ontario	456546
4	kevin	kev11@gmail	montreal	686897
5	romil	romil12@yahoo	vancouver	567866
6	zen	zen13@hotmail	brunswick	567778
NULL	NULL	NULL	NULL	NULL

CLIENT 1 X

Result Grid | Filter Rows: | Edit: | Export

*	order_id	quantity	Torec	Todel	c_id
1	2	2021-03-12	2021-03-12	2	
2	3	2021-03-13	2021-03-13	1	
3	1	2021-03-13	2021-03-13	2	
4	2	2021-04-14	2021-04-14	3	
5	3	2021-03-15	2021-03-15	4	
6	4	2021-03-23	2021-04-24	3	
7	1	2021-05-04	2021-05-04	5	
*	NULL	NULL	NULL	NULL	NULL

## SINGLE ROW SUBQUERY:

```
select c_id,c_name,c_email,c_address,c_mobile
from client
where c_id =
(select max(c_id) from client );
```

The screenshot shows a MySQL Workbench interface. The top bar has tabs for 'Query 1' and 'SQL'. Below the tabs is a toolbar with various icons for database management. The main area contains a numbered SQL query:

```
5  
6  
7 •  select c_id,c_name,c_email,c_address,c_mobile  
8   from client  
9   where c_id =  
10    (select max(c_id) from client );  
11
```

The result grid below the query shows one row of data:

	c_id	c_name	c_email	c_address	c_mobile
▶	6	zen	zen13@hotmail	brunswick	567778
*	NULL	NULL	NULL	NULL	NULL

## MULTIPLE ROW SUBQUERIES:

```
select c_id,c_name,c_email,c_address,c_mobile  
from client  
where c_address in  
(select c_address from client where c_address='montreal');
```

Query 1 ×

The screenshot shows the MySQL Workbench interface. The top bar has tabs for 'Query 1' and 'client 4'. Below the tabs is a toolbar with various icons. The main area contains a query editor with the following SQL code:

```
1 •  select c_id,c_name,c_email,c_address,c_mobile
2   from client
3   where c_address in
4     (select c_address from client where c_address='montreal');
5
```

Below the query editor is a 'Result Grid' pane. It has a header row with columns: c\_id, c\_name, c\_email, c\_address, c\_mobile. The data grid contains three rows of data:

c_id	c_name	c_email	c_address	c_mobile
1	rakesh	raks@gmail	montreal	345678
4	kevin	kev11@gmail	montreal	686897
*	NULL	NULL	NULL	NULL

client 4 × Apply

**select\* from client**

**where c\_id>**

**(select c\_id from client where c\_id=1);**

The screenshot shows the MySQL Workbench interface. The top menu bar includes Server, Tools, Scripting, and Help. Below the menu is a toolbar with various icons for database management. The main area is titled "Query 1" and contains the following SQL code:

```
11
12 •      select* from client
13      where c_id>
14      (select c_id from client where c_id=1);
15
```

The result grid below the code displays the following data:

	c_id	c_name	c_email	c_address	c_mobile
▶	2	aman	aman@gmail	brampton	567848
	3	devin	devin@hotmail	ontario	456546
	4	kevin	kev11@gmail	montreal	686897
	5	romil	romil12@yahoo	vancouver	567866
	6	zen	zen13@hotmail	brunswick	567778
*	NULL	NULL	NULL	NULL	NULL

((IN))

select \* from orders

where quantity in

(select quantity from orders where quantity>1);

The screenshot shows the MySQL Workbench interface. The top menu bar includes Database, Server, Tools, Scripting, and Help. Below the menu is a toolbar with various icons for database management. The main area is titled "Query 1" and contains the following SQL code:

```
15
16 •      select * from orders
17      where quantity in
18          (select quantity from orders where quantity>1);
19
```

The results are displayed in a "Result Grid" table:

	order_id	quantity	Torec	Todel	c_id
▶	1	2	2021-03-12	2021-03-12	2
	2	3	2021-03-13	2021-03-13	1
	4	2	2021-04-14	2021-04-14	3
	5	3	2021-03-15	2021-03-15	4
	6	4	2021-03-23	2021-04-24	3
*	NULL	NULL	NULL	NULL	NULL

select \* from orders

where Torec in

(select Torec from orders where Torec>'2021/3/12');

The screenshot shows a MySQL Workbench interface. The top menu bar includes 'File', 'Server', 'Tools', 'Scripting', and 'Help'. Below the menu is a toolbar with various icons. A tab labeled 'Query 1' is active. The query window contains the following code:

```
19
20 •      select * from orders
21      where Torec in
22          (select Torec from orders where Torec>'2021/3/12');
23
```

The result grid displays the following data:

	order_id	quantity	Torec	Todel	c_id
▶	2	3	2021-03-13	2021-03-13	1
	3	1	2021-03-13	2021-03-13	2
	4	2	2021-04-14	2021-04-14	3
	5	3	2021-03-15	2021-03-15	4
	6	4	2021-03-23	2021-04-24	3
*	7	1	2021-05-04	2021-05-04	5
	*	NULL	NULL	NULL	NULL

(ANY)

select \* from client

where c\_id > any

(select c\_id from client where c\_address='montreal');

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1

SCHEMAS Filter objects

practice1

Tables

- client
- cook
- dept\_624
- emp\_624
- menu
- orders
- Views
- Stored Procedures
- Functions

sakila

sys

world

23

24 • select \* from client

25 where c\_id > any

26 (select c\_id from client where c\_address='montreal');

27

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

c_id	c_name	c_email	c_address	c_mobile
2	aman	aman@gmail	brampton	567848
3	devin	devin@hotmail	ontario	456546
4	kevin	kev11@gmail	montreal	686897
5	romil	romil12@yahoo	vancouver	567866
6	zen	zen13@hotmail	brunswick	567778
*	NULL	NULL	NULL	NULL

(ALL)

select \*from client

where c\_id < all

(select c\_id from client where c\_address='brunswick');

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

practice1

Tables

- client
- cook
- dept\_624
- emp\_624
- menu
- orders

Views

Stored Procedures

Functions

sakila

sys

world

Query 1

```
27
28
29 •      select *from client
30      where c_id< all
31      (select c_id from client where c_address='brunswick');
```

Result Grid

c_id	c_name	c_email	c_address	c_mobile
1	rakesh	raks@gmail	montreal	345678
2	aman	aman@gmail	brampton	567848
3	devin	devin@hotmail	ontario	456546
4	kevin	kev11@gmail	montreal	686897
5	romil	romil12@yahoo	vancouver	567866
*	NULL	NULL	NULL	NULL

## (NOT EXISTS)

select \* from orders

where not exists

(select c\_id from client

where orders.order\_id=client.c\_id);

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

SCHEMAS

Filter objects

practice1

Tables

- client
- cook
- dept\_624
- emp\_624
- menu
- orders

Views

Stored Procedures

Functions

sakila

sys

world

Query 1

```
32
33 •      select * from orders
34 where not exists
35   (select c_id from client
36 where orders.order_id=client.c_id);
```

Result Grid

order_id	quantity	Torec	Todel	c_id
7	1	2021-05-04	2021-05-04	5
NULL	NULL	NULL	NULL	NULL

# THANK YOU