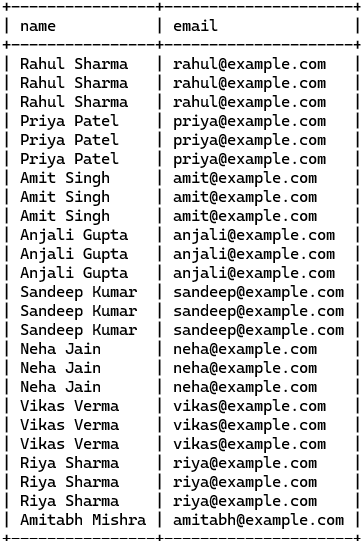
**SQL Query Output**

# 1.To select the name and email of all users who have placed an order:

SELECT Users.name, Users.email

FROM Users

JOIN Orders ON Users.user\_id = Orders.user\_id;



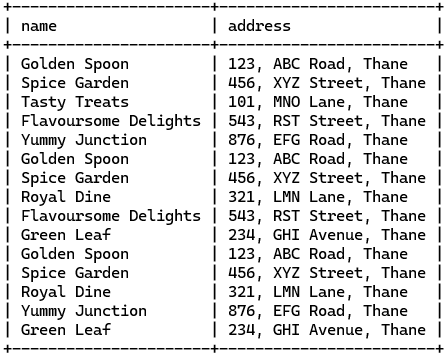
# 2.To select the name and address of all restaurants that have received a rating of 4 or higher:

SELECT Restaurants.name, Restaurants.address

FROM Restaurants

JOIN Rating ON Restaurants.restaurant\_id = Rating.restaurant\_id

WHERE Rating.rating >= 4;



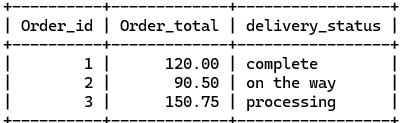
# 3.To select the order id, total, and status of all orders that have been placed by a user with the email ‘rahul@example.com’:

SELECT Order\_id, Order\_total, delivery\_status

FROM Orders

JOIN Users ON Orders.user\_id = Users.user\_id

WHERE Users.email = 'rahul@example.com';



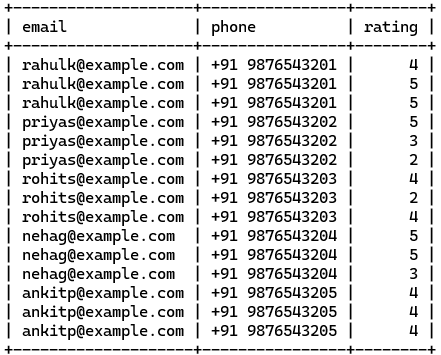
# 4.To select the name, phone, and rating of all drivers who have delivered an order:

SELECT Drivers.email, Drivers.phone, Rating.rating

FROM Drivers

JOIN Orders ON Drivers.driver\_id = Orders.driver\_id

JOIN Rating ON Orders.order\_id = Rating.rating\_id;



# 5.To select the name, price and status of the payment of all menu items on a particular restaurant:

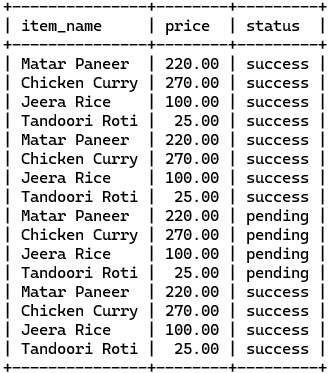
SELECT Menu.item\_name, Menu.price, Payment.status

FROM Menu

JOIN Orders ON Menu.restaurant\_id = Orders.restaurant\_id

JOIN Payment ON Orders.order\_id = Payment.order\_id

WHERE Menu.restaurant\_id = 1;



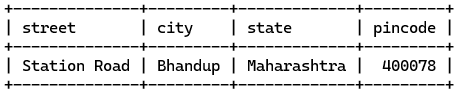
# 6.To select all the addresses of the user who placed a particular order:

SELECT Address.street, Address.city, Address.state, Address.pincode

FROM Address

JOIN Orders ON Address.user\_id = Orders.user\_id

WHERE Orders.order\_id = 4;



# multiple join and groupby & Having clause

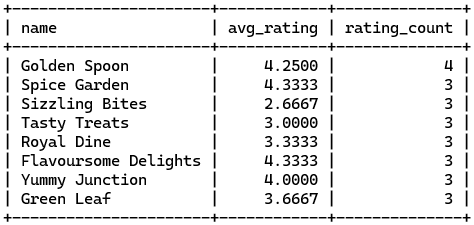
# 1. To select the average rating for each restaurant, along with the number of ratings they have received:

SELECT Restaurants.name, AVG(Rating.rating) as avg\_rating, COUNT(Rating.rating) as rating\_count

FROM Restaurants

JOIN Rating ON Restaurants.restaurant\_id = Rating.restaurant\_id

GROUP BY Restaurants.name;



# Groupby clause

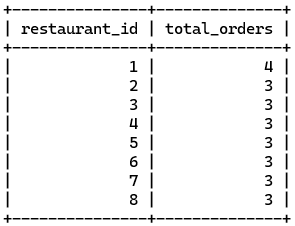
# Get the total number of orders per restaurant with order total greater than 50:

SELECT restaurant\_id, COUNT(\*) AS total\_orders

FROM Orders

GROUP BY restaurant\_id

HAVING SUM(order\_total) > 100;

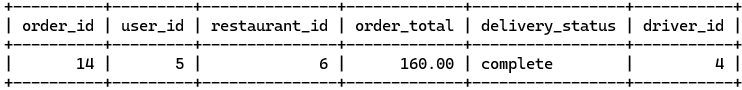


# Where Clause

# Get orders with delivery status 'Delivered' and placed by user with ID 5:

SELECT \* FROM Orders

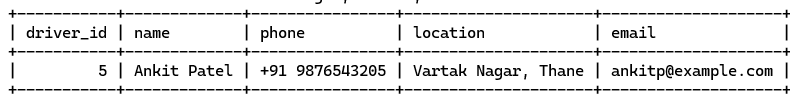
WHERE delivery\_status = 'Complete' AND user\_id = 5;



# Get drivers located in 'Vartak Nagar':

SELECT \* FROM Drivers

WHERE location = 'Vartak Nagar, Thane';



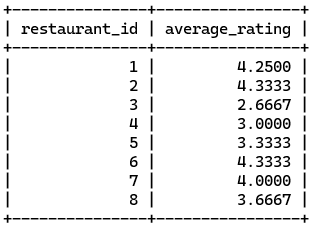
# Entries with GROUP BY Clause:

# Get the average rating for each restaurant:

SELECT restaurant\_id, AVG(rating) AS average\_rating

FROM Rating

GROUP BY restaurant\_id;

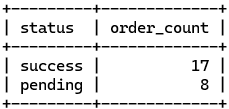


# Get the count of orders for each payment status:

SELECT status, COUNT(\*) AS order\_count

FROM Payment

GROUP BY status;



# Single Sub Query

# Get the total number of orders for users who have placed more than 2 orders:

SELECT COUNT(\*) AS total\_orders

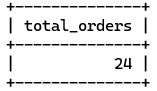
FROM Orders

WHERE user\_id IN (SELECT user\_id

FROM Orders

GROUP BY user\_id

HAVING COUNT(\*) > 2);



# Get the names of users who have placed orders with a total greater than 150:

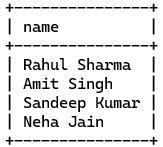
SELECT name FROM Users

WHERE user\_id IN (

SELECT user\_id

FROM Orders

WHERE order\_total > 150);



# Get the average rating of restaurants that have received ratings:

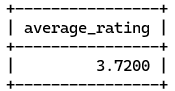
SELECT AVG(rating) AS average\_rating

FROM Rating

WHERE restaurant\_id IN (

SELECT DISTINCT restaurant\_id

FROM Rating);



# Get the names of users who have not placed any orders:

SELECT name FROM Users

WHERE user\_id NOT IN (

SELECT DISTINCT user\_id

FROM Orders);



# Double Subqueries:

# Identify users who have ordered from restaurants that are serviced by drivers operating from 'Kolshet, Thane'.

SELECT name FROM Users

WHERE user\_id IN (

SELECT user\_id

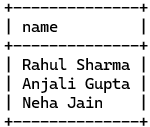
FROM Orders

WHERE restaurant\_id IN (

SELECT driver\_id

FROM Drivers

WHERE location = 'Kolshet, Thane'));



# Get the total number of orders placed at restaurants where the average rating is above 4:

SELECT COUNT(\*) AS total\_orders

FROM Orders

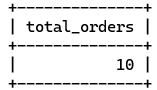
WHERE restaurant\_id IN (

SELECT restaurant\_id

FROM Rating

GROUP BY restaurant\_id

HAVING AVG(rating) > 4);



# Get the names of users who have placed orders with a total greater than the average order total:

SELECT name

FROM Users

WHERE user\_id IN (

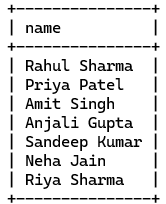
SELECT user\_id

FROM Orders

WHERE order\_total > (

SELECT AVG(order\_total)

FROM Orders));

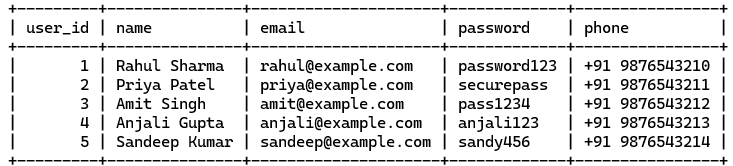


# Like Operator:

# Get users whose email contains 'gmail':

SELECT \* FROM Users WHERE email LIKE '%example%'

limit 5;



# Having Clause:

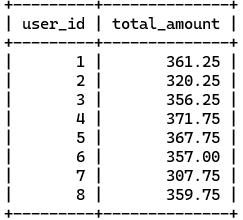
# Get the total order amount for each user who has placed orders more than 200:

SELECT user\_id, SUM(order\_total) AS total\_amount

FROM Orders

GROUP BY user\_id

HAVING total\_amount > 200;



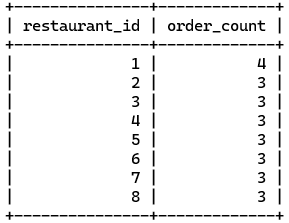
# Group By Clause:

# Get the count of orders for each restaurant:

SELECT restaurant\_id, COUNT(\*) AS order\_count

FROM Orders

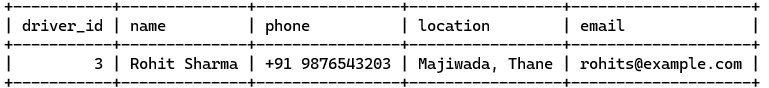
GROUP BY restaurant\_id;



# Where Clause:

# Get the details of the driver with the name 'Rohit Sharma':

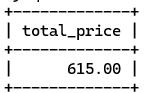
SELECT \* FROM Drivers WHERE name = 'Rohit Sharma';



# Arithmetic Operators:

# Get the total price of all menu items for a restaurant with ID 1:

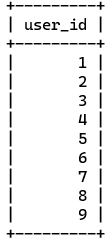
SELECT SUM(price) AS total\_price FROM Menu WHERE restaurant\_id = 1;



# Comparison Operators:

# Get users who have placed orders with a total greater than $50:

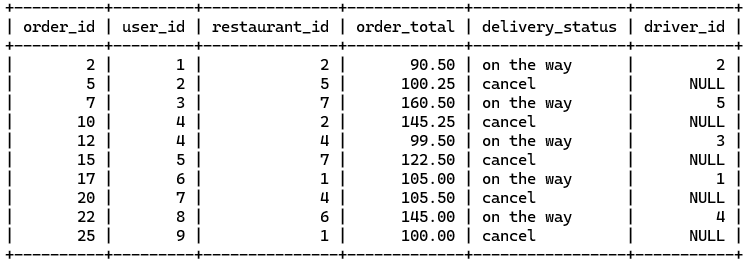
SELECT DISTINCT user\_id FROM Orders WHERE order\_total > 50;



# Logical Operators:

# Get orders with a delivery status of 'Cancel' or 'on the way':

SELECT \* FROM Orders WHERE delivery\_status = 'cancel' OR delivery\_status = 'on the way';

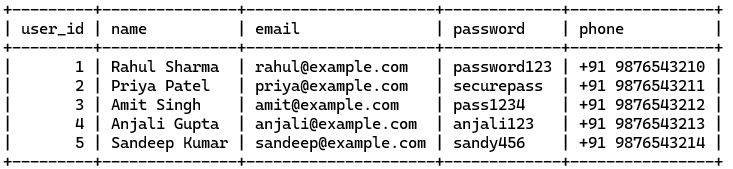


# Null Value:

# Get users who do not have an associated address:

SELECT \* FROM Users WHERE user\_id IN (SELECT DISTINCT user\_id FROM Address)

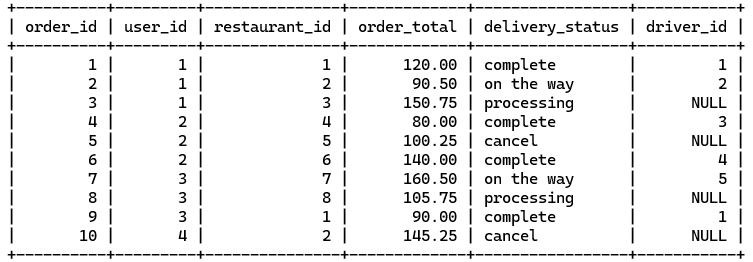
limit 5;



# Limit Command:

# Get the first 10 orders:

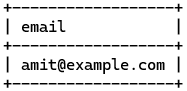
SELECT \* FROM Orders LIMIT 10;



# Single Row Subqueries:

# Get the email of the user who placed the highest order:

SELECT email FROM Users WHERE user\_id = (SELECT user\_id FROM Orders ORDER BY order\_total DESC LIMIT 1);

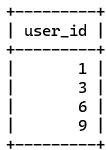


# Multiple Row Subqueries:

# Get users who have placed orders at restaurants in 'Thane':

SELECT DISTINCT user\_id FROM Orders WHERE restaurant\_id IN (SELECT restaurant\_id

FROM Restaurants WHERE address = '123, ABC Road, Thane');



# Subquery with IN:

# Get the names of restaurants where the average order total is greater than $50:

SELECT name FROM Restaurants WHERE restaurant\_id IN (SELECT restaurant\_id

FROM Orders GROUP BY restaurant\_id HAVING AVG(order\_total) > 50);

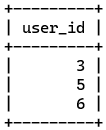


# Subquery with ALL:

# Get users who have placed orders with a total greater than all orders placed by user with ID 1:

SELECT DISTINCT user\_id FROM Orders WHERE order\_total > ALL (SELECT order\_total

FROM Orders WHERE user\_id = 1);

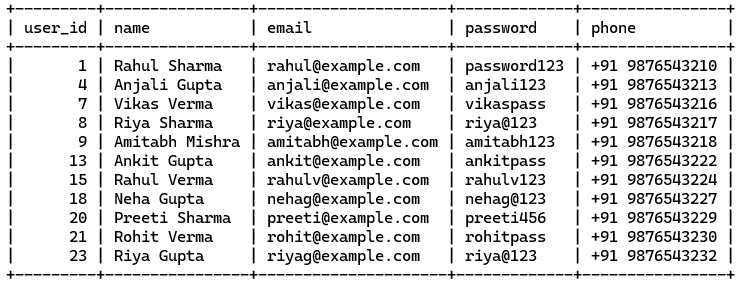


# like Operator

SELECT \* FROM Users WHERE name LIKE 'A%';



SELECT \* FROM Users WHERE name LIKE '%A';



SELECT \* FROM Users WHERE name LIKE '\_R%';



SELECT \* FROM Users WHERE name LIKE 'R\_%';

