

# SWIGGY CASE STUDY



submitted by - Paridhi Bhardwaj

# 1. DISPLAY ALL CUSTOMERS WHO LIVE IN 'DELHI'.

The screenshot shows a MySQL Workbench interface. The left sidebar displays database objects, tables, and schemas. The main area shows a query editor with the following SQL code:

```
1 •   SELECT
2       name, city
3   FROM
4       customers
5 WHERE
6       city = 'delhi';
```

The result grid below shows the output of the query:

	name	city
▶	Rohini Verma	Delhi
▶	Manish Kumar	Delhi
▶	Sonali Mishra	Delhi

## 2. FIND THE AVERAGE RATING OF ALL RESTAURANTS IN 'MUMBAI'

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

- SQL Editor:** The current tab is "swiggy database (1)".
- Query:** The query is:1 • SELECT
2 avg(rating)
3 FROM
4 restaurants
5 WHERE
6 city = "Mumbai";
- Result Grid:** The result of the query is displayed in a grid:
|  | avg(rating) |
| --- | --- |
| ▶ | 4.300000 |

### 3. List all customers who have placed at least one order.



```
order details  SQL File 2*  swiggy database (1)  SQL File 6* 
1 •  SELECT
2      distinct(customers.name)
3  FROM
4      customers
5  JOIN
6      orders ON customers.customer_id = orders.customer_id
```

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

name
Amit Sharma
Rohini Verma
Rajesh Gupta
Sneha Mehta
Manish Kumar
Priya Singh
Vikas Reddy
Anjali Patel
Suresh Nair
Kavita Deshmukh
Vivek Bhatt
Meera Joshi
Pankaj Jain

Result 7 \*

The screenshot shows a SQL query being run against a database named 'swiggy database (1)'. The query selects distinct customer names from the 'customers' table, joining it with the 'orders' table on the 'customer\_id' column. The results are displayed in a grid, listing 15 distinct customer names. The interface includes tabs for 'order details', 'SQL File 2\*', 'SQL File 6\*', and the current result set. A sidebar on the left lists various tables in the database, including 'customers', 'deliverypartners', 'deliveryupdates', 'feedback', 'menuitems', 'orderdelivery', 'orderitems', 'orders', 'payment', 'restaurants', and 'news'. The bottom of the interface shows a toolbar with various icons and a limit of 1000 rows.

## 4. Display the total number of orders placed by each customer.

The screenshot shows a MySQL Workbench interface. The left pane displays the database schema with tables like customers, deliverypartners, deliveryupdates, feedback, menuitems, orderdelivery, orderitems, and orders. The central pane contains a SQL query:

```
1 • SELECT
2     customers.name , count(orders.order_id) total_orders
3     FROM
4     customers
5     JOIN
6     orders ON customers.customer_id = orders.customer_id
7     group by customers.name
```

The right pane shows the results of the query in a grid:

name	total_orders
Amit Sharma	2
Rohini Verma	3
Rajesh Gupta	3
Sneha Mehta	2
Manish Kumar	4
Priya Singh	3
Vikas Reddy	3
Anjali Patel	3
Suresh Nair	1
Kavita Deshmukh	2
Vivek Bhatt	2
Meera Joshi	2

## 5. Find the total revenue generated by each restaurant.

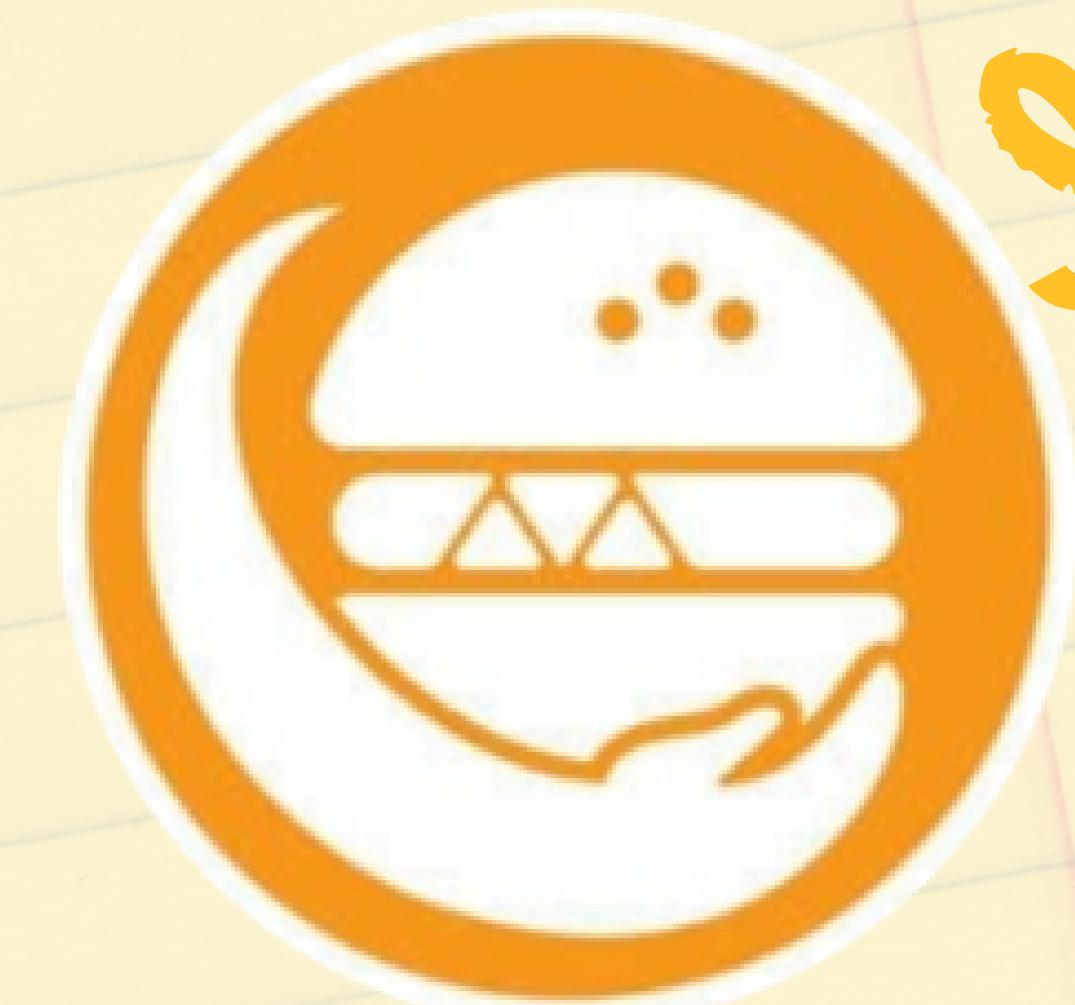
The screenshot shows a MySQL query interface with the following SQL code:

```
1 • SELECT
2     restaurants.restaurant_id , sum(orders.total_amount) total_revenue , restaurants.name
3 FROM
4     restaurants
5     JOIN
6     orders ON restaurants.restaurant_id = orders.restaurant_id
7     group by restaurants.name, restaurants.restaurant_id;
```

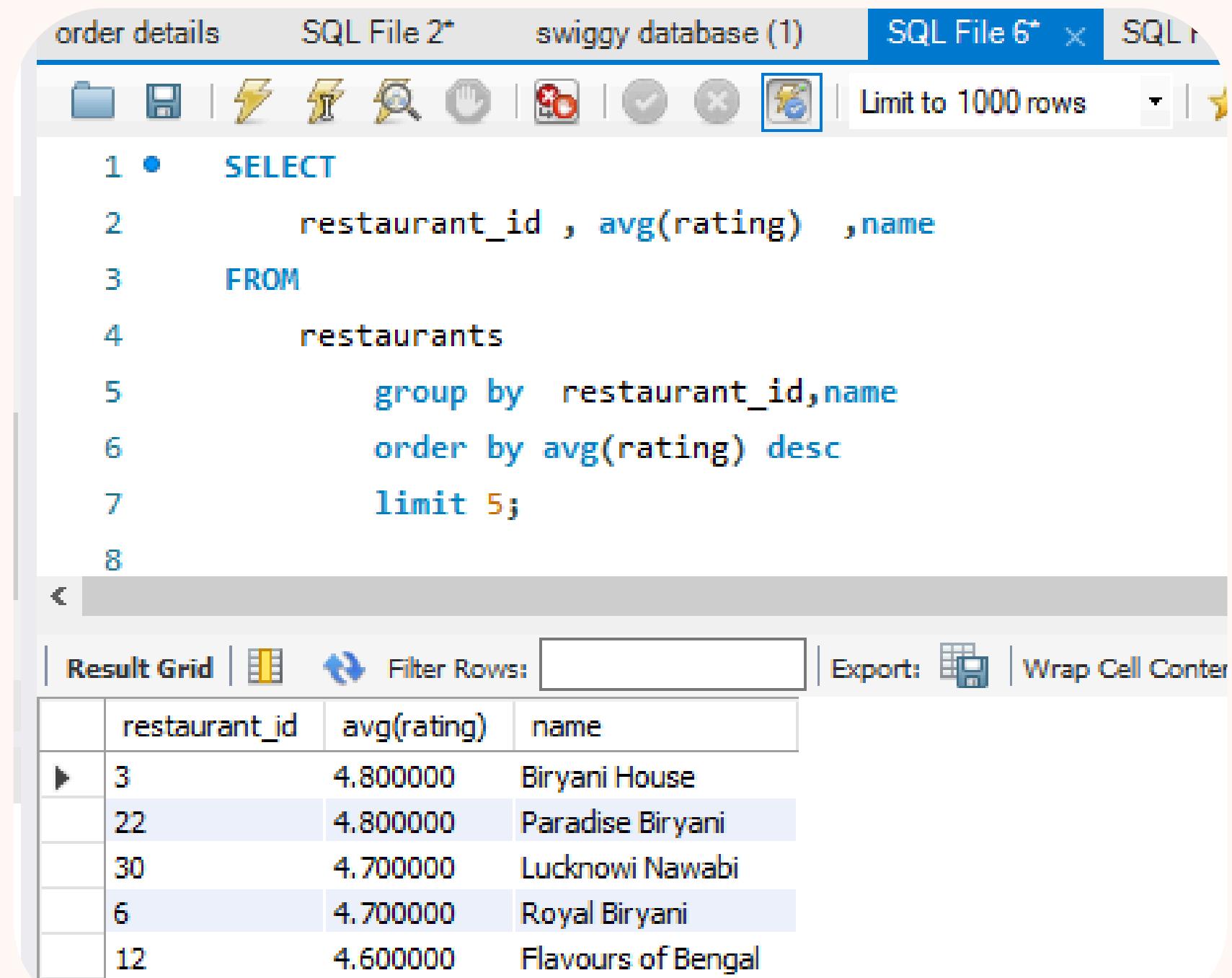
The results grid displays the following data:

restaurant_id	total_revenue	name
3	5300.00	Biryani House
5	600.00	Taste of Punjab
1	1100.00	Spice of India
7	2100.00	Coastal Delight
2	1200.00	Tandoori Flames
4	3200.00	Curry Pot
8	1600.00	Veggie Delight
9	2550.00	Gujarat Express
6	650.00	Royal Biryani
11	900.00	Punjabi Tadka
12	4050.00	Flavours of Ben...
13	7050.00	South Treat

Result 9 x



# 6. FIND THE TOP 5 RESTAURANTS WITH THE HIGHEST AVERAGE RATING



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

- SQL File 6\*** tab is selected.
- Toolbar icons:** folder, table, lightning bolt, wrench, magnifying glass, refresh, checkmark, close, and a gear icon.
- Limit to 1000 rows** dropdown menu.
- SQL Query:**

```
1 •  SELECT
2      restaurant_id ,  avg(rating) ,name
3  FROM
4      restaurants
5      group by  restaurant_id,name
6      order by avg(rating) desc
7      limit 5;
8
```
- Result Grid:** Shows the results of the query in a table format.
- Table Headers:** restaurant\_id, avg(rating), name.
- Table Data:**

	restaurant_id	avg(rating)	name
▶	3	4.800000	Biryani House
	22	4.800000	Paradise Biryani
	30	4.700000	Lucknowi Nawabi
	6	4.700000	Royal Biryani
	12	4.600000	Flavours of Bengal

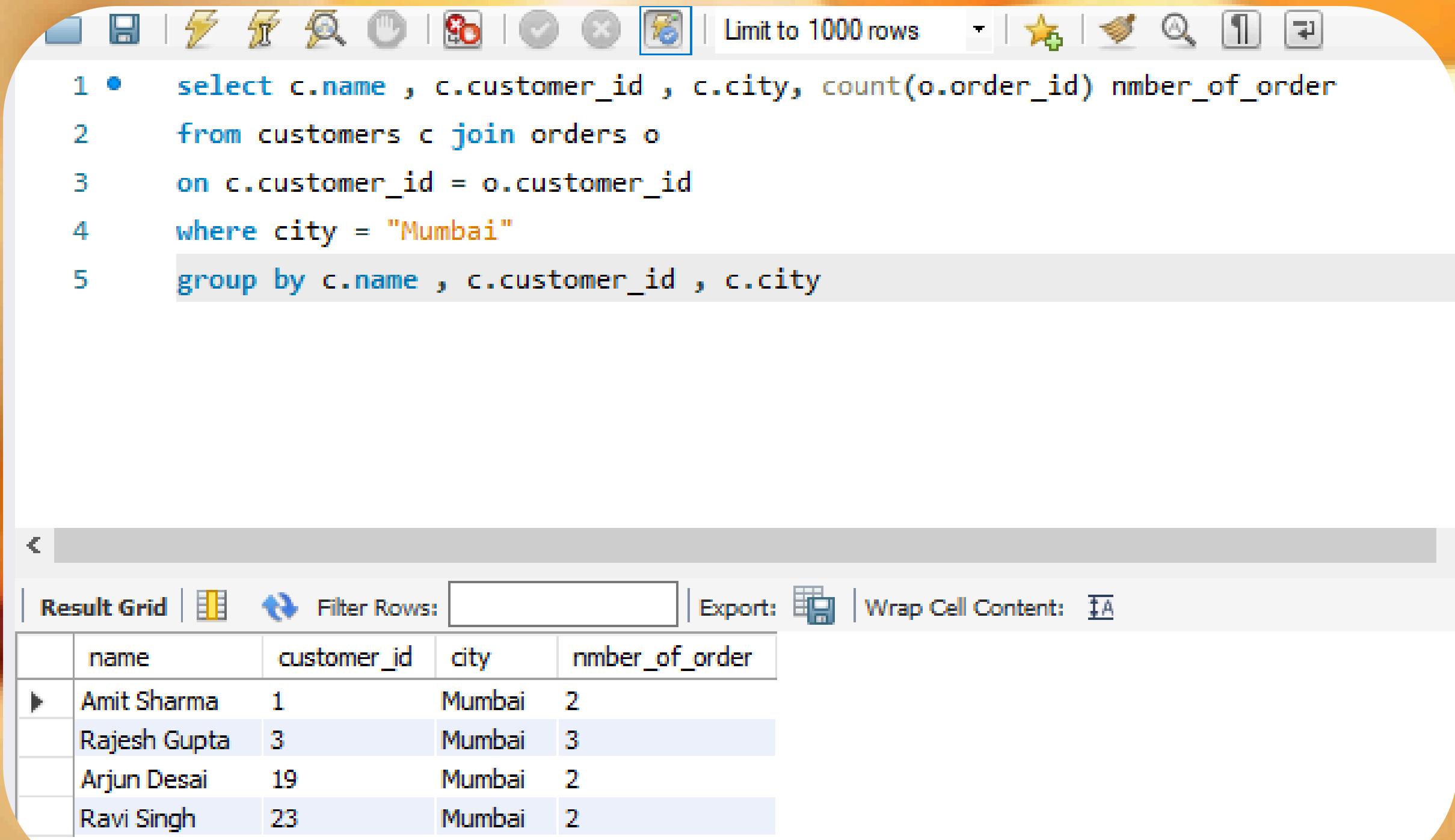
# 7. DISPLAY ALL CUSTOMERS WHO HAVE NEVER PLACED AN ORDER.

```
1  SELECT
2    *
3  FROM
4    customers AS c
5    LEFT JOIN
6    orders AS o ON c.customer_id = o.customer_id
7 WHERE
8    o.order_id IS NULL;
```

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

	customer_id	name	email	phone_number	city	address	order_id	customer_id	restaurant_id	order_date	total_amount	status
▶	24	Sonal Kaur	sonal.kaur@gmail.com	HULL	Amritsar	S-19, Ranjit Avenue	HULL	HULL	HULL	HULL	HULL	HULL
	25	Vivek Malhotra	vivek.malhotra@hotmail.com	9812789012	Thane	HULL	HULL	HULL	HULL	HULL	HULL	HULL
	26	Divya Iyer	HULL	9823890123	Bangalore	T-20, Indiranagar	HULL	HULL	HULL	HULL	HULL	HULL
	27	Rakesh Yadav	rakesh.yadav@gmail.com	9834901234	Varanasi	U-31, Lanka	HULL	HULL	HULL	HULL	HULL	HULL
	28	Mona Sharma	mona.sharma@yahoo.com	9845012345	Ranchi	HULL	HULL	HULL	HULL	HULL	HULL	HULL
	29	Sudha Pillai	sudha.pillai@gmail.com	9856123789	Kozhikode	V-42, Mavoor Road	HULL	HULL	HULL	HULL	HULL	HULL
	30	Gaurav Khanna	HULL	9867238901	Gwalior	W-53, City Centre	HULL	HULL	HULL	HULL	HULL	HULL

## 8. FIND THE NUMBER OF ORDERS PLACED BY EACH CUSTOMER IN 'MUMBAI'.



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

```
1 •  select c.name , c.customer_id , c.city, count(o.order_id) nmber_of_order
2    from customers c join orders o
3      on c.customer_id = o.customer_id
4    where city = "Mumbai"
5    group by c.name , c.customer_id , c.city
```

The results grid displays the following data:

	name	customer_id	city	nmber_of_order
▶	Amit Sharma	1	Mumbai	2
	Rajesh Gupta	3	Mumbai	3
	Arjun Desai	19	Mumbai	2
	Ravi Singh	23	Mumbai	2



# 9. Display all orders placed in the last 30 days.

Indexes  
Foreign Keys  
Triggers  
deliverypartners  
deliveryupdates  
feedback  
menuitems  
orderdelivery  
orderitems  
orders

Navigation Schemas

Table: orders

Columns:

- order\_id int AI PK
- customer\_id int
- restaurant\_id int
- order\_date datetime
- total\_amount decimal(10,2)
- status varchar(20)

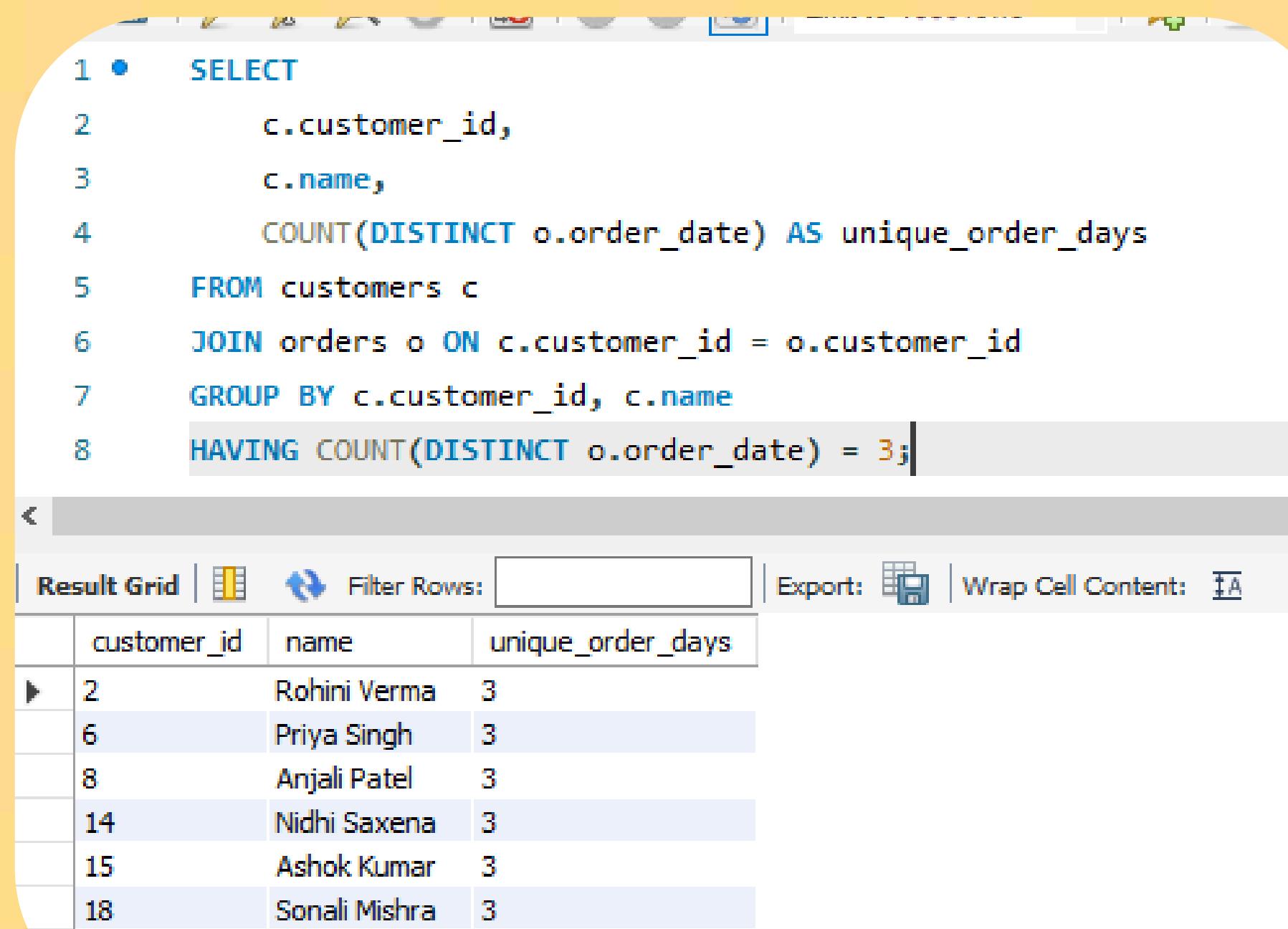
```
3
4 • SELECT *
5   FROM orders
6 WHERE order_date >= (SELECT DATE_SUB(MAX(order_date), INTERVAL 30 DAY)
7   FROM orders);
```

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

	order_id	customer_id	restaurant_id	order_date	total_amount	status
▶	1	1	3	2024-08-01 00:00:00	750.00	Completed
	2	2	5	2024-08-02 00:00:00	600.00	Completed
	3	3	1	2024-08-04 00:00:00	0.00	Cancelled
	4	4	7	2024-08-01 00:00:00	850.00	Completed
	5	5	2	2024-08-03 00:00:00	1200.00	Completed
	6	1	4	2024-08-06 00:00:00	500.00	Processing
	7	6	8	2024-08-03 00:00:00	950.00	Completed
	8	7	9	2024-08-08 00:00:00	700.00	Completed
	9	8	6	2024-08-02 00:00:00	650.00	Completed
	10	9	11	2024-08-09 00:00:00	0.00	Cancelled
..	11	10	12	2024-08-01 00:00:00	800.00	Completed



# 10. FIND THE CUSTOMERS WHO HAVE PLACED ORDERS ON EXACTLY THREE DIFFERENT DAYS.



The screenshot shows a MySQL query editor window with the following details:

- Query Text:**

```
1 • SELECT
2     c.customer_id,
3     c.name,
4     COUNT(DISTINCT o.order_date) AS unique_order_days
5   FROM customers c
6   JOIN orders o ON c.customer_id = o.customer_id
7   GROUP BY c.customer_id, c.name
8   HAVING COUNT(DISTINCT o.order_date) = 3;
```
- Result Grid:**

	customer_id	name	unique_order_days
▶	2	Rohini Verma	3
	6	Priya Singh	3
	8	Anjali Patel	3
	14	Nidhi Saxena	3
	15	Ashok Kumar	3
	18	Sonali Mishra	3

# 11 FIND THE DELIVERY PARTNER WHO HAS WORKED WITH THE MOST DIFFERENT CUSTOMERS.

The screenshot shows a database interface with a sidebar containing objects like deliveryupdates, feedback, menuitems, orderdelivery, orderitems, orders, payment, restaurants, Views, Stored Procedures, and Functions. The 'Orders' table is selected. Below the sidebar, the main area displays a SQL query:

```
1 •   SELECT
2       dp.partner_id,
3       dp.name,
4       COUNT(DISTINCT o.customer_id) AS unique_customers
5   FROM
6       deliverypartners dp
7       JOIN
8           orderdelivery od ON dp.partner_id = od.partner_id
9       JOIN
10          orders o ON od.order_id = o.order_id
11      GROUP BY dp.partner_id , dp.name
12      ORDER BY unique_customers DESC
13      LIMIT 1;
```

The results grid shows one row:

	partner_id	name	unique_customers
▶	4	Suresh Reddy	6

**THANK  
YOU!**

