LAB-7
QUERIES

Query-1: Top 10 places having most expensive dishes

Relational Algebra:

$$\pi_{r.Name,\, m.Dish\_name,\, m.Price}\left(\tau_{m.Price\ \mathrm{DESC}}\left(\sigma_{r.Restaurant\_id=m.Restaurant\_id}(Restaurant \times Menu)\right)\right)_{Top\ 10}$$

SQL:
```
SELECT r.Name AS Restaurant, m.Dish_name, m.Price
FROM Restaurant r
JOIN Menu m ON r.Restaurant_id = m.Restaurant_id
ORDER BY m.Price DESC
LIMIT 10;
```

Output:

```
498    LIMIT 5;
```

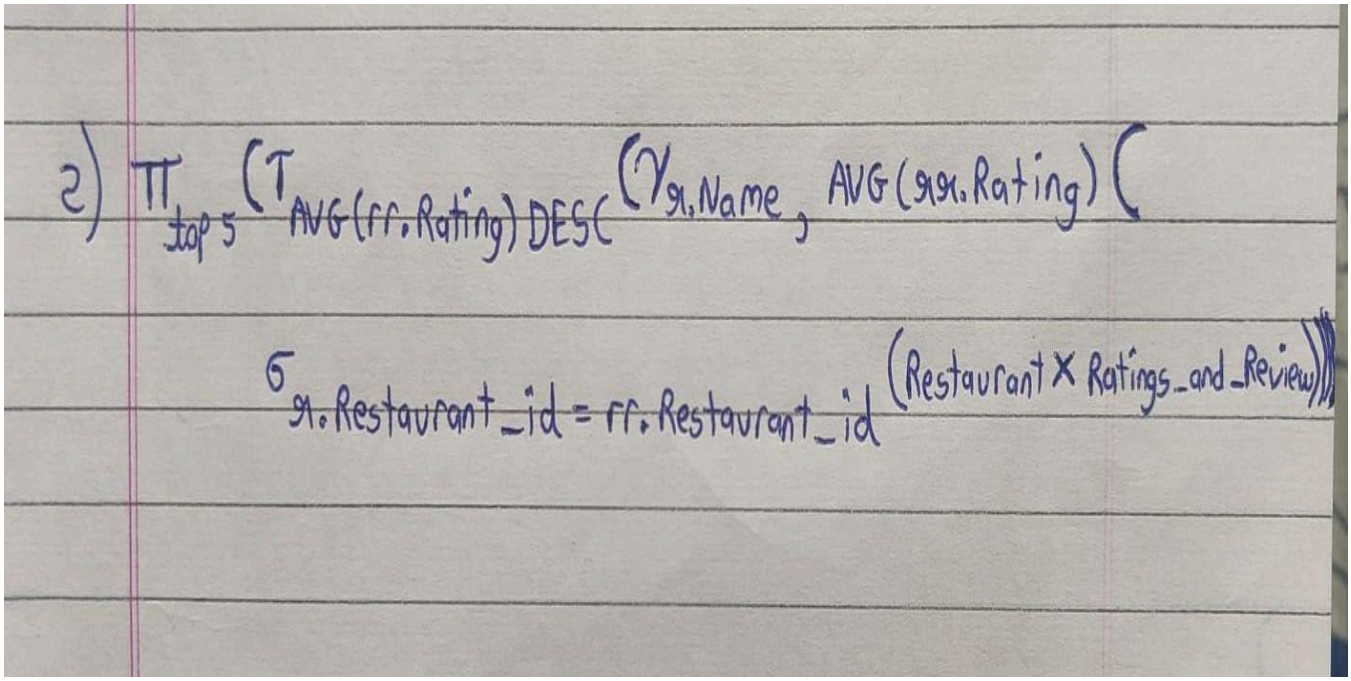
Data Output | Messages | Notifications

	restaurant character varying (100)	dish_name character varying (100)	price integer
1	Leato	Pizza (Pepperoni)	350
2	Rajkot Rasoi	Thali Combo	350
3	Leato	Lasagna	350
4	Tandoori Treats	Butter Chicken	350
5	Leato	Pizza (Vegetable)	320
6	Leato	Pizza (Margherita)	300
7	Rangoli Rasoi	Dal Makhani	300
8	Limbdi Delight	Dal Makhani	300
9	Rajkot Rasoi	Dal Baati Churma	300
10	Tandoori Treats	Chicken Tikka Masala	300

Total rows: 10 of 10 Query complete 00:00:00.070

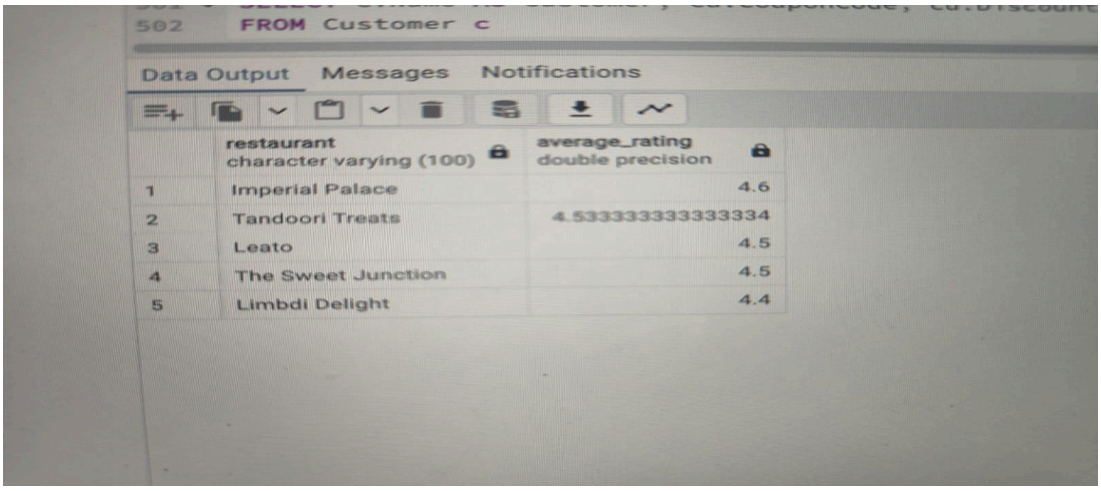
Query-2: Top 5 Restaurants with Highest Average Rating

Relational Algebra:



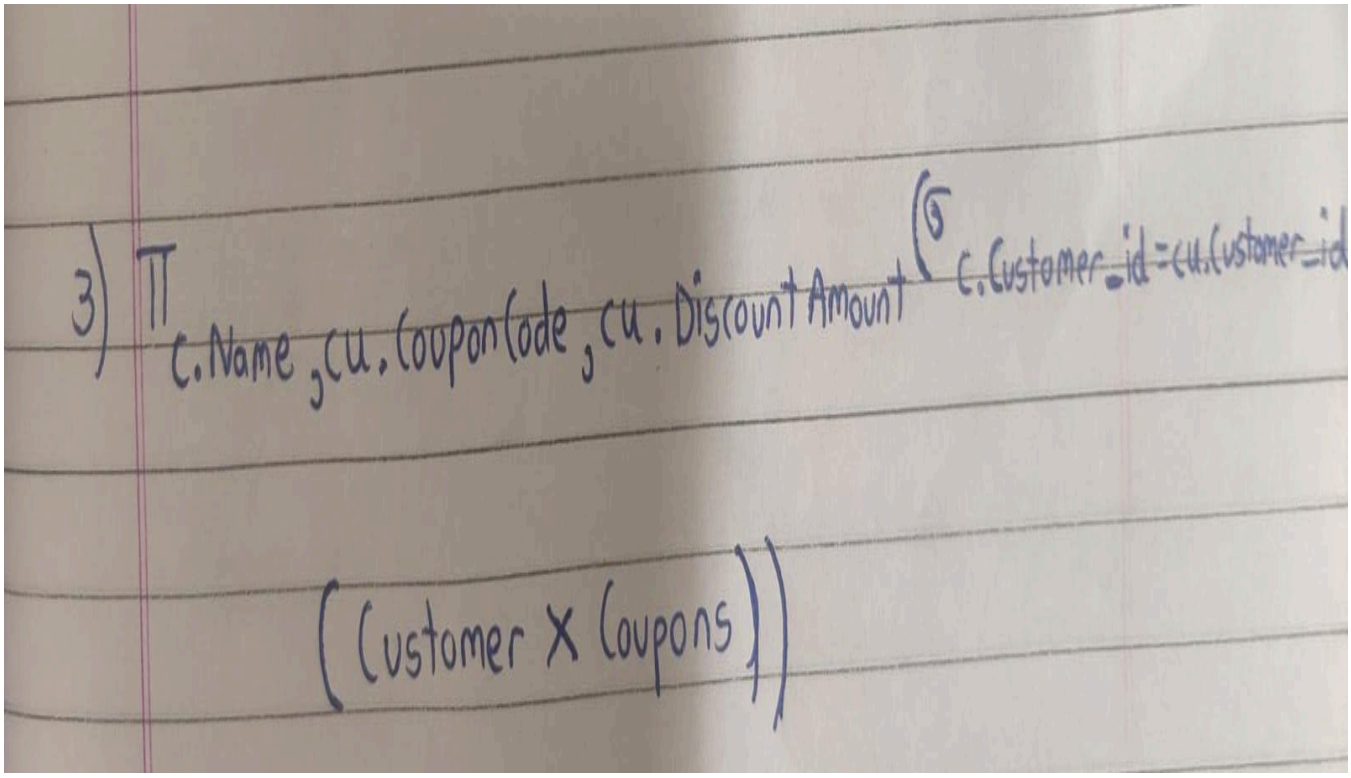
SQL: SELECT r.Name AS Restaurant, AVG(rr.Rating) AS Average_Rating
FROM Restaurant r
JOIN Ratings_and_Review rr ON r.Restaurant_id = rr.Restaurant_id
GROUP BY r.Name
ORDER BY Average_Rating DESC
LIMIT 5;

Output:



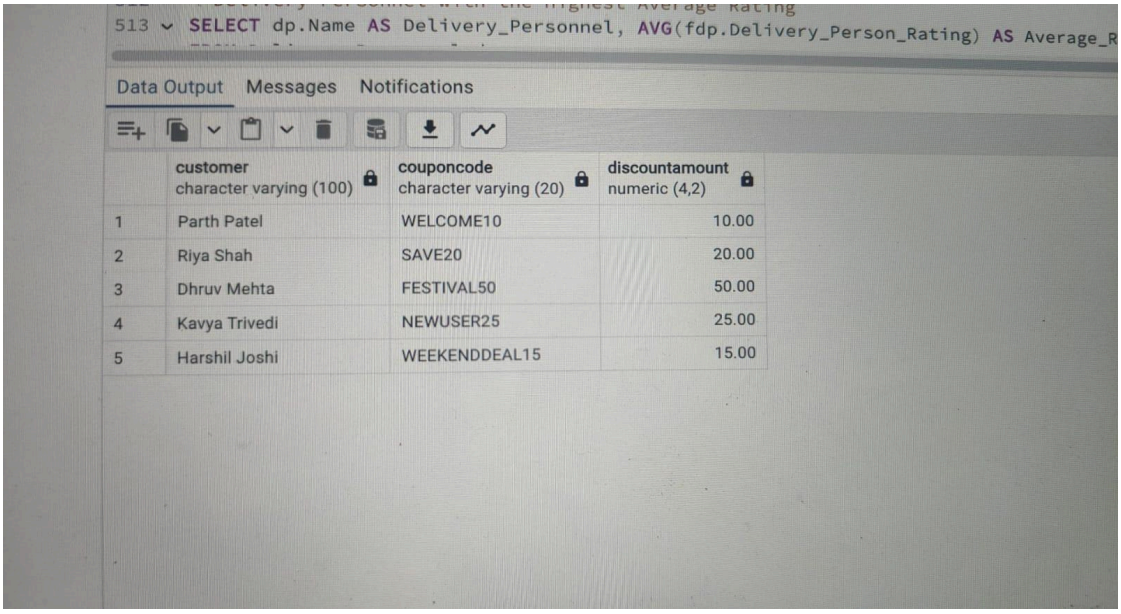
Query-3: Customers Who Have Used Coupons

Relational Algebra:



SQL: SELECT c.Name AS Customer, cu.CouponCode, cu.DiscountAmount
FROM Customer c
JOIN Coupons cu ON c.Customer_id = cu.Customer_id;

Output:



	customer character varying (100)	couponcode character varying (20)	discountamount numeric (4,2)
1	Parth Patel	WELCOME10	10.00
2	Riya Shah	SAVE20	20.00
3	Dhruv Mehta	FESTIVAL50	50.00
4	Kavya Trivedi	NEWUSER25	25.00
5	Harshil Joshi	WEEKENDDEAL15	15.00

Query-4: Orders by Customers in Specific Area

Relational Algebra:

$$\pi_{o.Order\_id,\, c.Name,\, o.Item\_Ordered,\, o.Total\_Amount,\, o.Order\_date\_and\_time}(\sigma_{ca.DeliveryAddress\ \text{LIKE}\ '\%Kalawad\ Road\%'}(\sigma_{c.Customer\_id=ca.CustomerID}(\sigma_{o.Customer\_id=c.Customer\_id}(Orders \times Customer) \times CustomerAddresses)))$$

SQL: SELECT o.Order_id, c.Name AS Customer, o.Item_Ordered, o.Total_Amount, o.Order_date_and_time
FROM Orders o
JOIN Customer c ON o.Customer_id = c.Customer_id
JOIN CustomerAddresses ca ON c.Customer_id = ca.CustomerID
WHERE ca.DeliveryAddress LIKE '%Kalawad Road%';

Output:

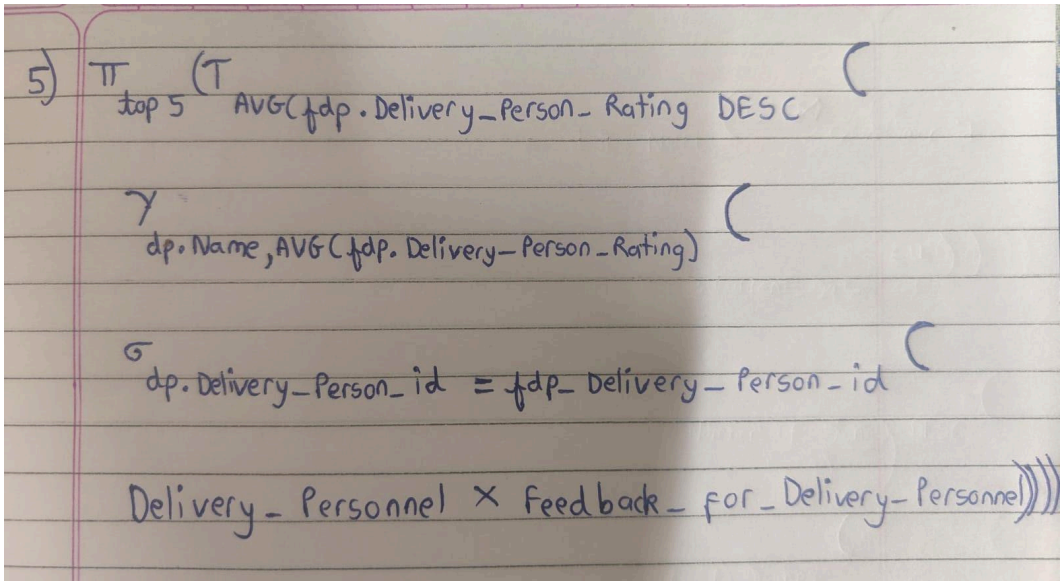
```
512  --Delivery Personnel with the Highest Average Rating
513  SELECT dp.Name AS Delivery_Personnel, AVG(fdp.Delivery_Person_Rating) AS Average_Rating
```

Data Output | Messages | Notifications

	order_id bigint	customer character varying (100)	item_ordered character varying (100)	total_amount integer	order_date_and_time timestamp without time zone
1	20230303	Dhruv Mehta	Vada Pav	30	2024-10-17 20:45:00
2	20230309	Jay Patel	Paneer Tikka	200	2024-10-18 18:45:00
3	202303012	Dhruv Mehta	Chicken Tikka Masala	300	2024-10-16 11:30:00
4	202303018	Jay Patel	Rasmalai	80	2024-10-17 20:00:00
5	202303021	Dhruv Mehta	Khichdi	150	2024-10-15 15:15:00
6	202303027	Jay Patel	Pizza (Vegetable)	320	2024-10-16 09:45:00
7	202303030	Dhruv Mehta	Ice Cream	80	2024-10-19 13:15:00

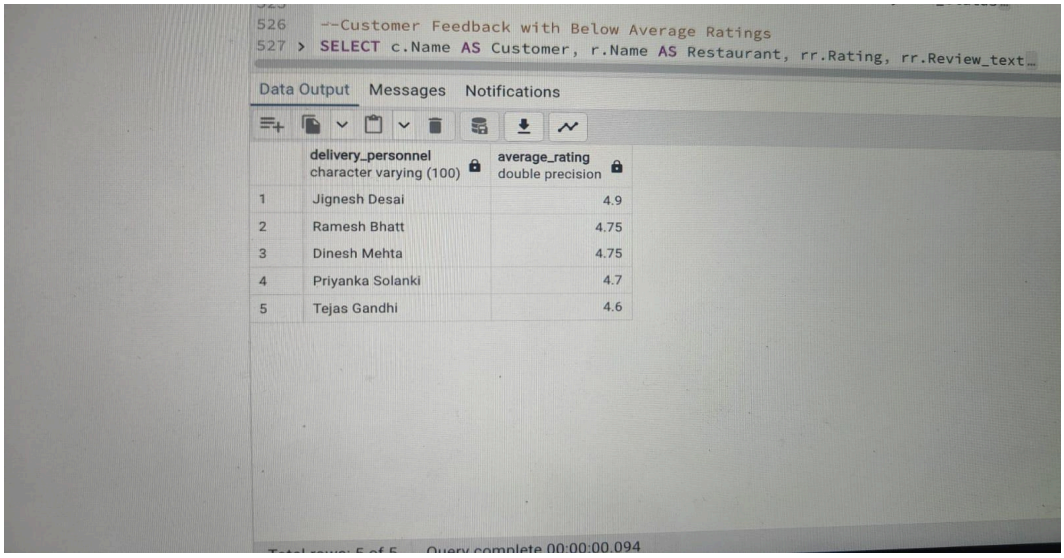
Query-5: Delivery Personnel with the Highest Average Rating

Relational Algebra:



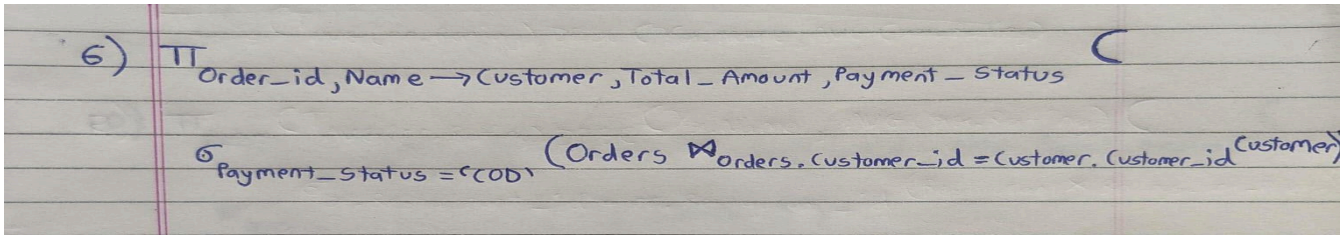
SQL: SELECT dp.Name AS Delivery_Personnel, AVG(fdp.Delivery_Person_Rating) AS Average_Rating
FROM Delivery_Personnel dp
JOIN Feedback_for_Delivery_Personnel fdp ON dp.Delivery_Person_id = fdp.Delivery_Person_id
GROUP BY dp.Name
ORDER BY Average_Rating DESC
LIMIT 5;

Output:



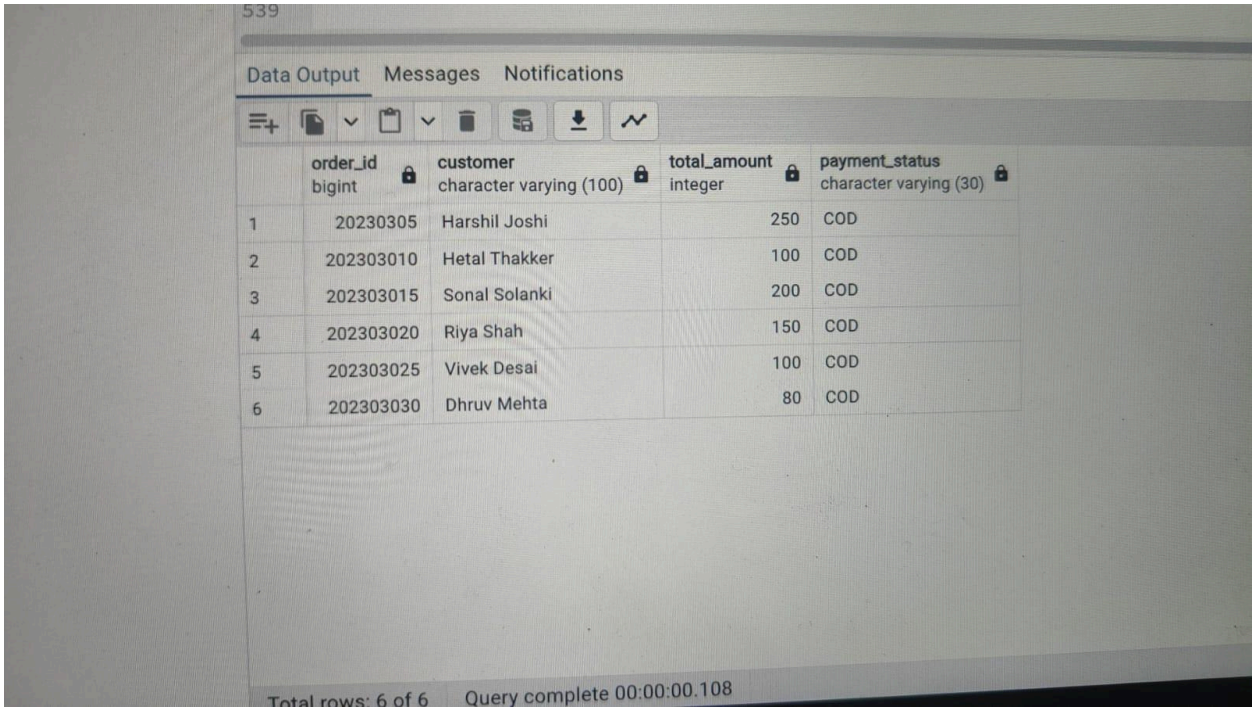
Query-6: List of Orders with Pending Payment Status

Relational Algebra:



SQL: SELECT o.Order_id, c.Name AS Customer, o.Total_Amount, o.Payment_Status
FROM Orders o
JOIN Customer c ON o.Customer_id = c.Customer_id
WHERE o.Payment_Status = 'COD';

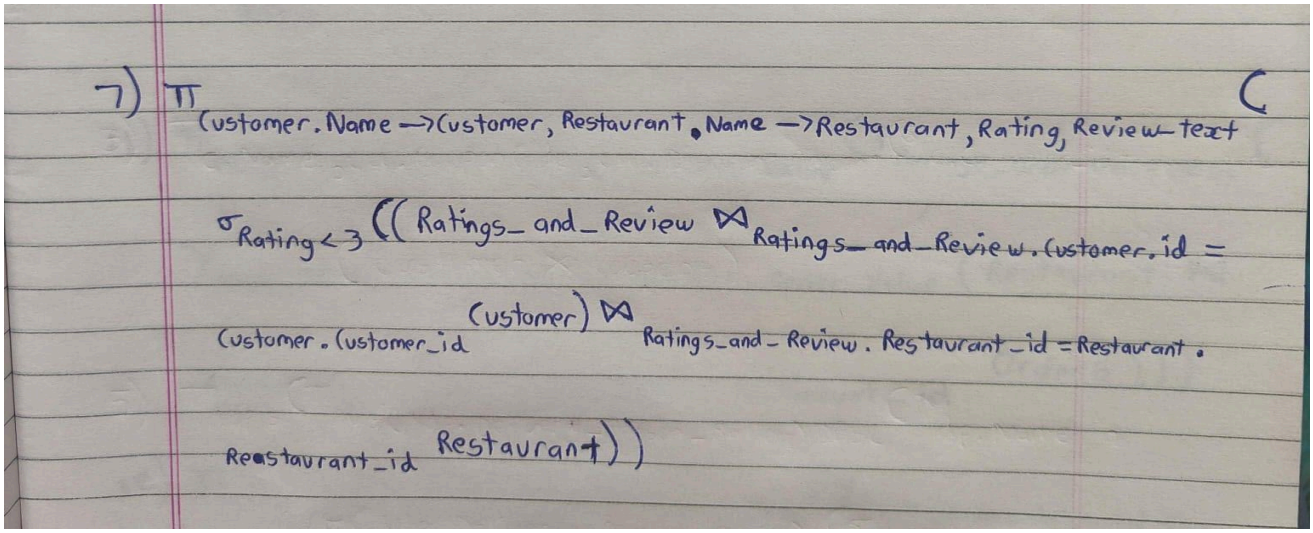
Output:



	order_id bigint	customer character varying (100)	total_amount integer	payment_status character varying (30)
1	20230305	Harshil Joshi	250	COD
2	202303010	Hetal Thakker	100	COD
3	202303015	Sonal Solanki	200	COD
4	202303020	Riya Shah	150	COD
5	202303025	Vivek Desai	100	COD
6	202303030	Dhruv Mehta	80	COD

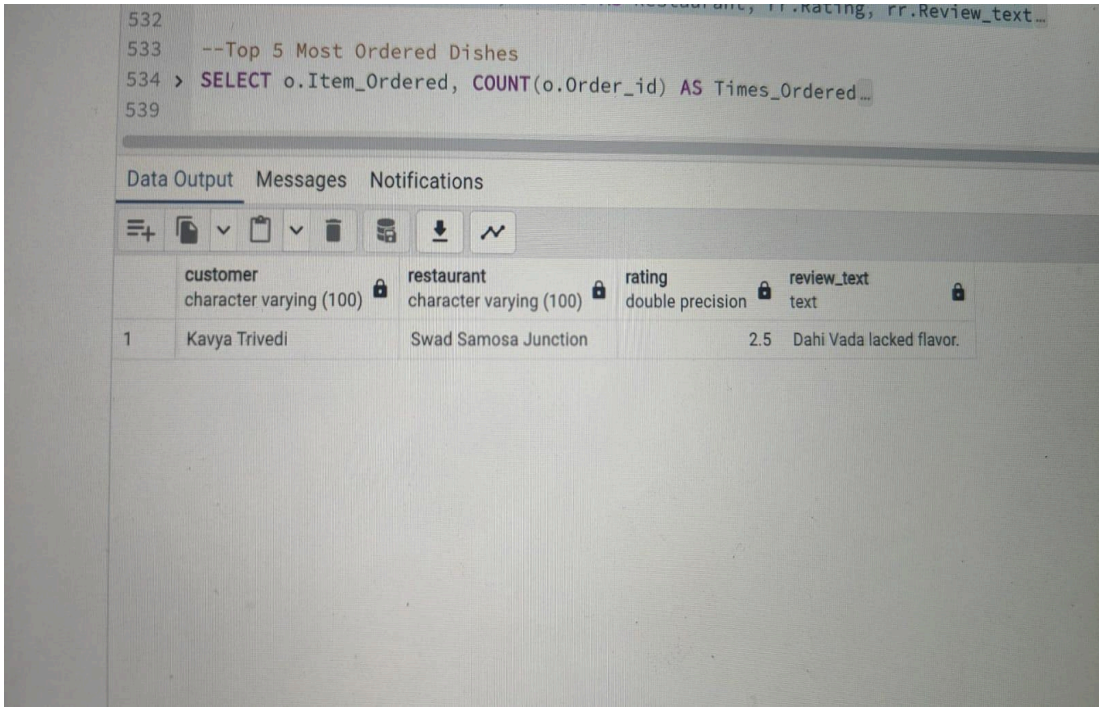
Query-7: Customer Feedback with Below Average Ratings

Relational Algebra:

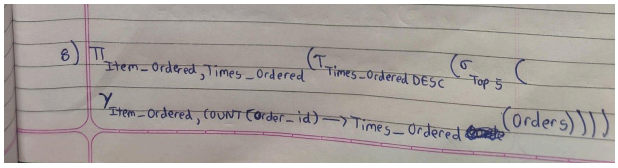


SQL: SELECT c.Name AS Customer, r.Name AS Restaurant, rr.Rating, rr.Review_text
FROM Ratings_and_Review rr
JOIN Customer c ON rr.Customer_id = c.Customer_id
JOIN Restaurant r ON rr.Restaurant_id = r.Restaurant_id
WHERE rr.Rating < 3;

Output:

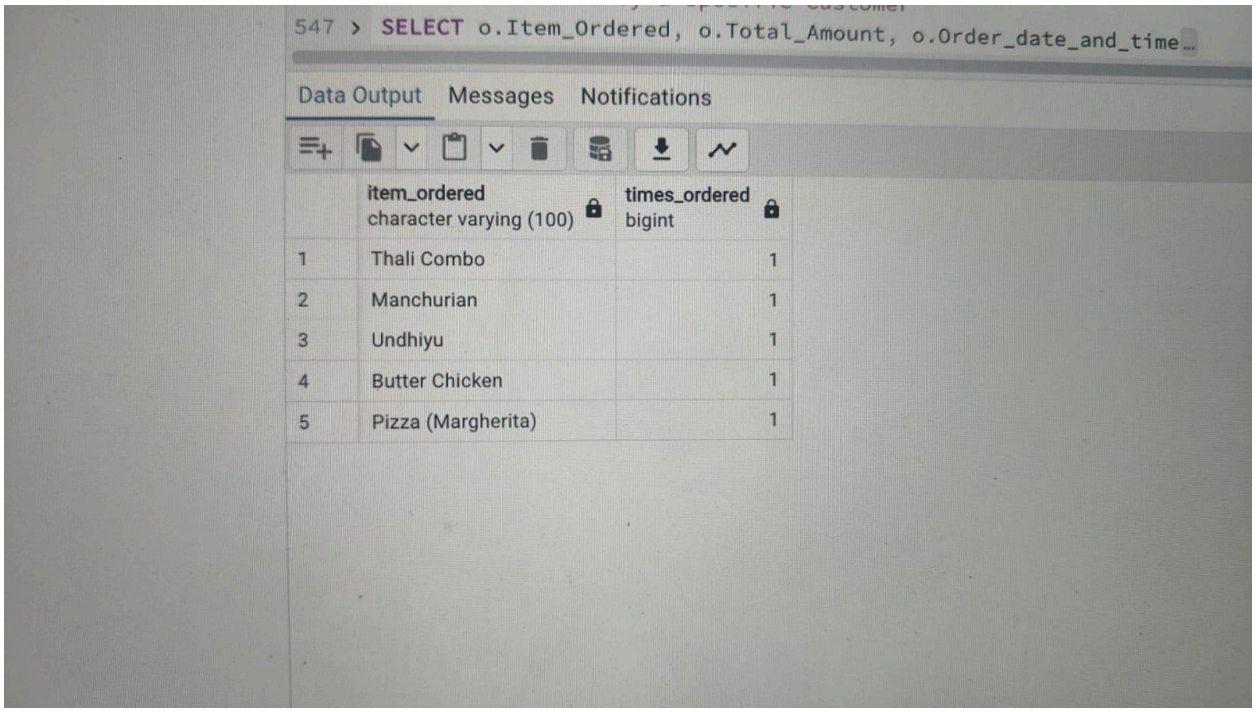


Query-8: Top 5 Most Ordered Dishes

Relational Algebra: 

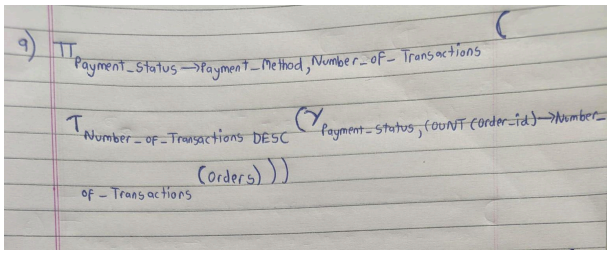
SQL: SELECT o.Item_Ordered, COUNT(o.Order_id) AS Times_Ordered
FROM Orders o
GROUP BY o.Item_Ordered
ORDER BY Times_Ordered DESC
LIMIT 5;

Output:



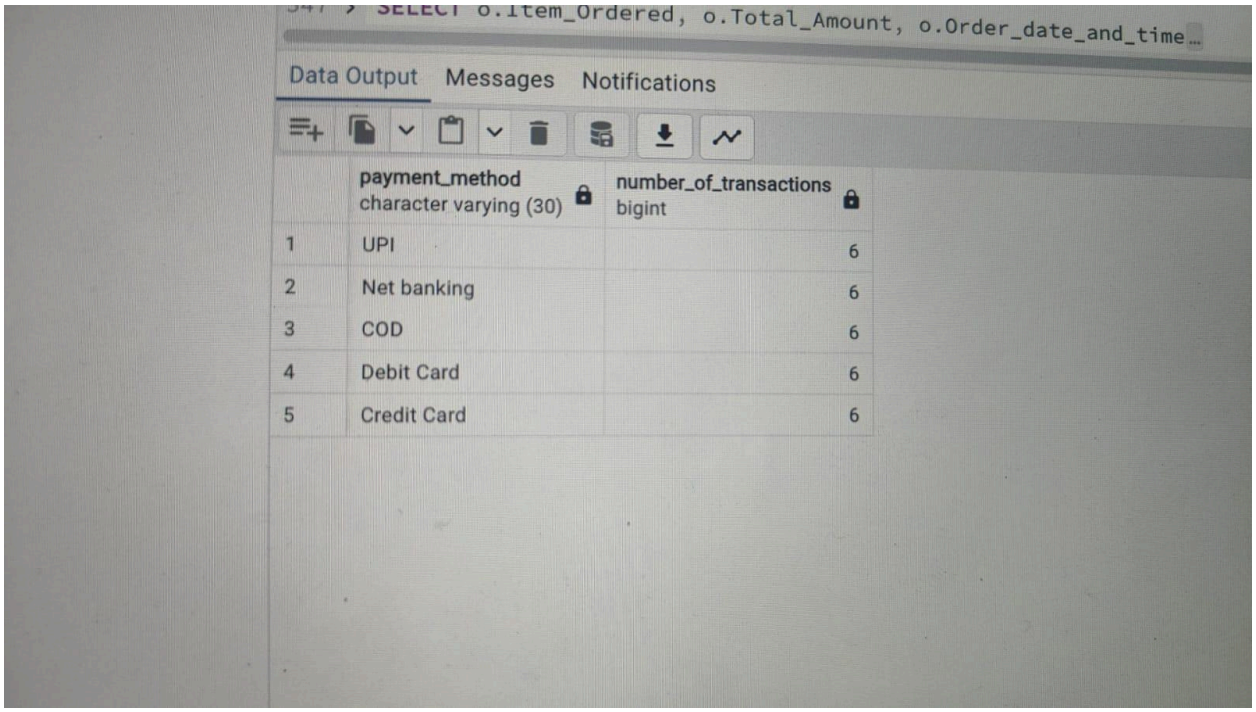
	item_ordered character varying (100)	times_ordered bigint
1	Thali Combo	1
2	Manchurian	1
3	Undhiyu	1
4	Butter Chicken	1
5	Pizza (Margherita)	1

Query-9: Most Popular Payment Method

Relational Algebra: 

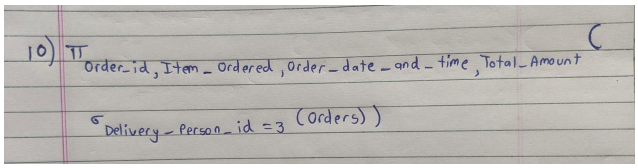
SQL: SELECT o.Payment_Status AS Payment_Method, COUNT(o.Order_id) AS Number_of_Transactions
FROM Orders o
GROUP BY o.Payment_Status
ORDER BY Number_of_Transactions DESC;

Output:



	payment_method character varying (30)	number_of_transactions bigint
1	UPI	6
2	Net banking	6
3	COD	6
4	Debit Card	6
5	Credit Card	6

Query-10: Orders Delivered by a Specific Delivery Personnel

Relational Algebra: 

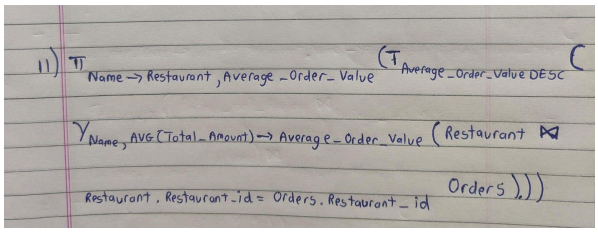
SQL: SELECT o.Order_id, o.Item_Ordered, o.Order_date_and_time, o.Total_Amount
FROM Orders o
WHERE o.Delivery_Person_id = 3;

Output:

	order_id [PK] bigint	item_ordered character varying (100)	order_date_and_time timestamp without time zone	total_amount integer
1	20230303	Vada Pav	2024-10-17 20:45:00	30
2	202303013	Dahi Vada	2024-10-17 19:45:00	50
3	202303023	Pani Puri	2024-10-17 18:15:00	30

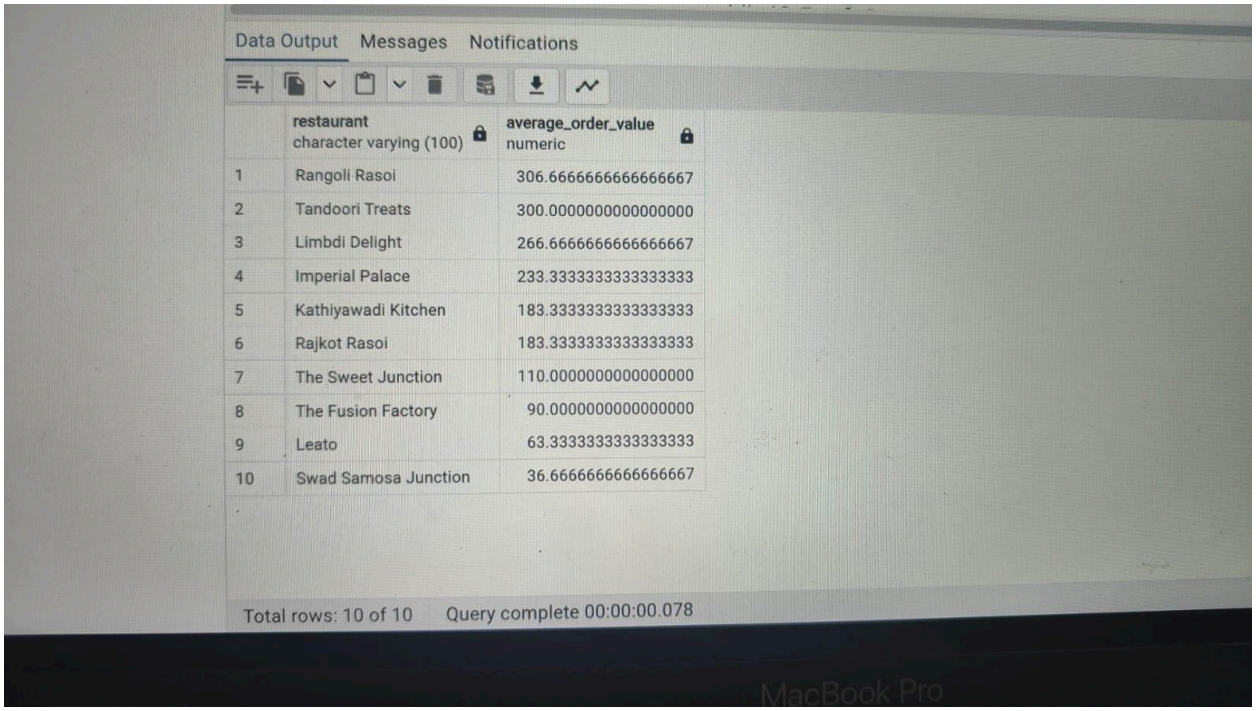
Query-11: Average Order Value by Restaurant

Relational Algebra:



SQL: SELECT r.Name AS Restaurant, AVG(o.Total_Amount) AS Average_Order_Value
FROM Restaurant r
JOIN Orders o ON r.Restaurant_id = o.Restaurant_id
GROUP BY r.Name
ORDER BY Average_Order_Value DESC;

Output:



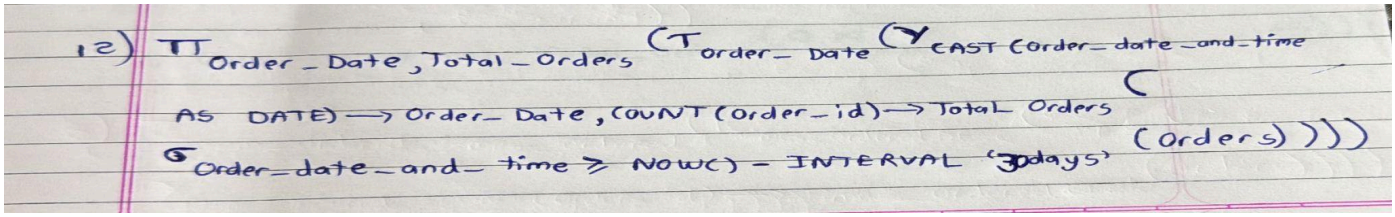
Data Output | Messages | Notifications

	restaurant character varying (100)	average_order_value numeric
1	Rangoli Rasoi	306.6666666666666667
2	Tandoori Treats	300.0000000000000000
3	Limbdi Delight	266.6666666666666667
4	Imperial Palace	233.3333333333333333
5	Kathiyawadi Kitchen	183.3333333333333333
6	Rajkot Rasoi	183.3333333333333333
7	The Sweet Junction	110.0000000000000000
8	The Fusion Factory	90.0000000000000000
9	Leato	63.3333333333333333
10	Swad Samosa Junction	36.6666666666666667

Total rows: 10 of 10 Query complete 00:00:00.078

Query-12: Total Number of Orders for Each Day in the Last Week

Relational Algebra:



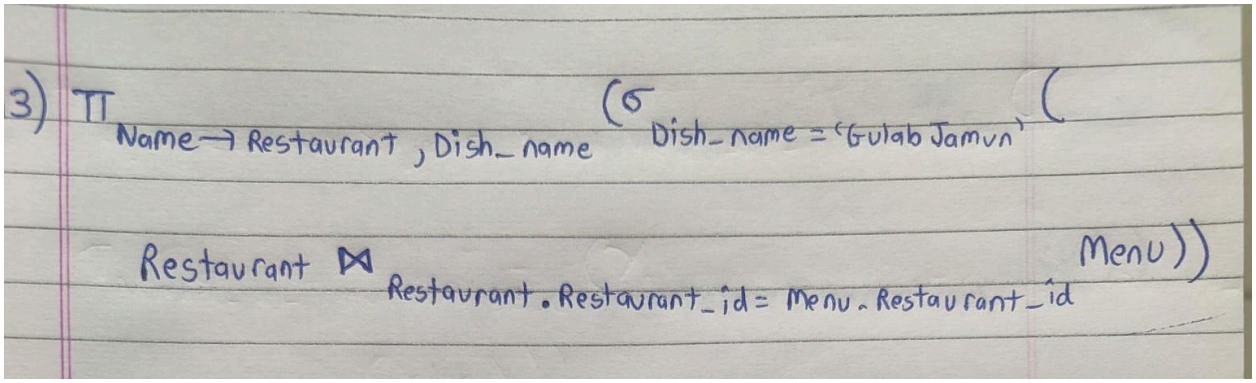
SQL: SELECT CAST(o.Order_date_and_time AS DATE) AS Order_Date,
COUNT(o.Order_id) AS Total_Orders
FROM Orders o
WHERE o.Order_date_and_time >= NOW() - INTERVAL '7 days'
GROUP BY Order_Date
ORDER BY Order_Date;

Output:

	order_date date	total_orders bigint
1	2024-10-15	6
2	2024-10-16	6
3	2024-10-17	6
4	2024-10-18	6
5	2024-10-19	6

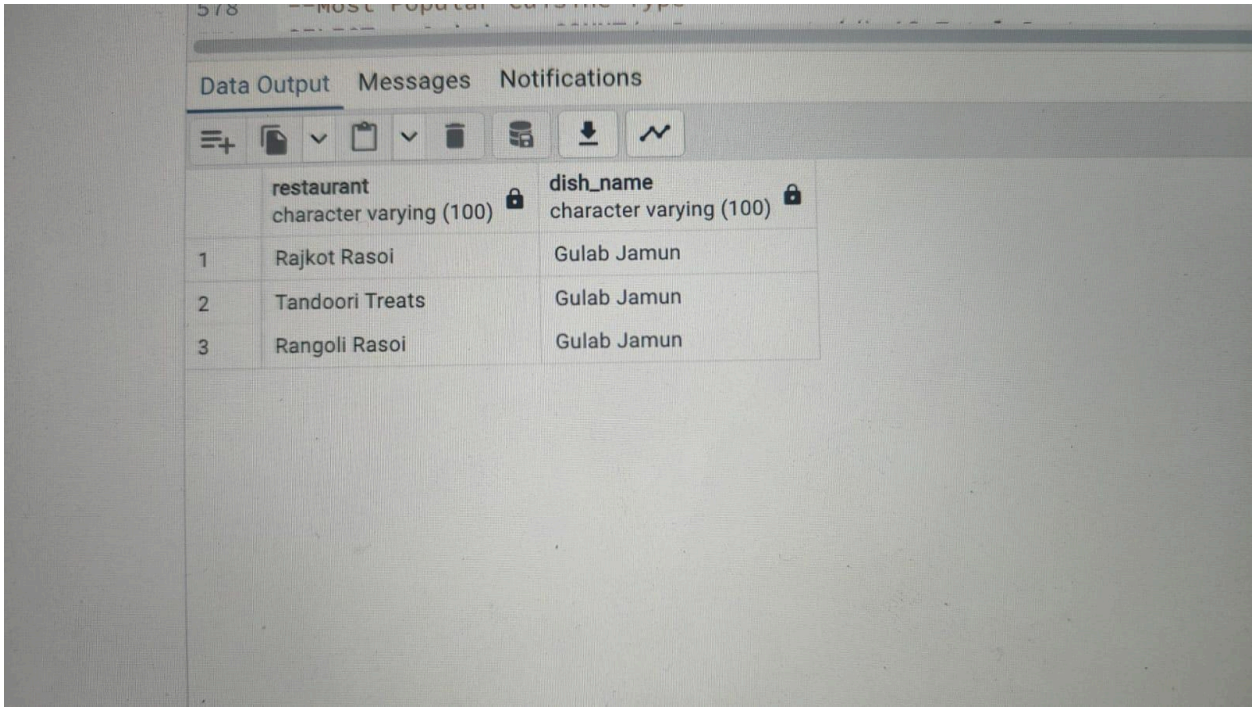
Query-13: Restaurants That Serve the Same Dish

Relational Algebra:



SQL: SELECT r.Name AS Restaurant, m.Dish_name
FROM Restaurant r
JOIN Menu m ON r.Restaurant_id = m.Restaurant_id
WHERE m.Dish_name = 'Gulab Jamun';

Output:

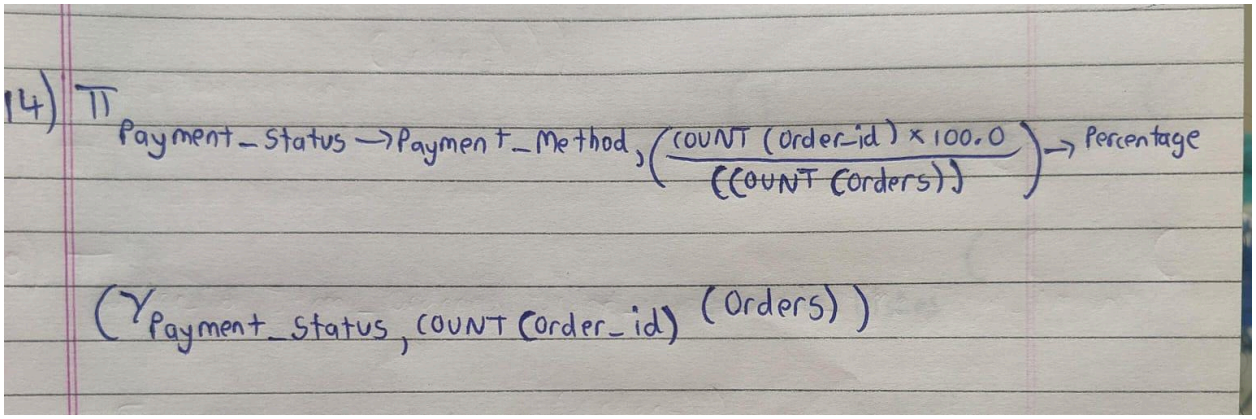


Data Output | Messages | Notifications

	restaurant character varying (100)	dish_name character varying (100)
1	Rajkot Rasoi	Gulab Jamun
2	Tandoori Treats	Gulab Jamun
3	Rangoli Rasoi	Gulab Jamun

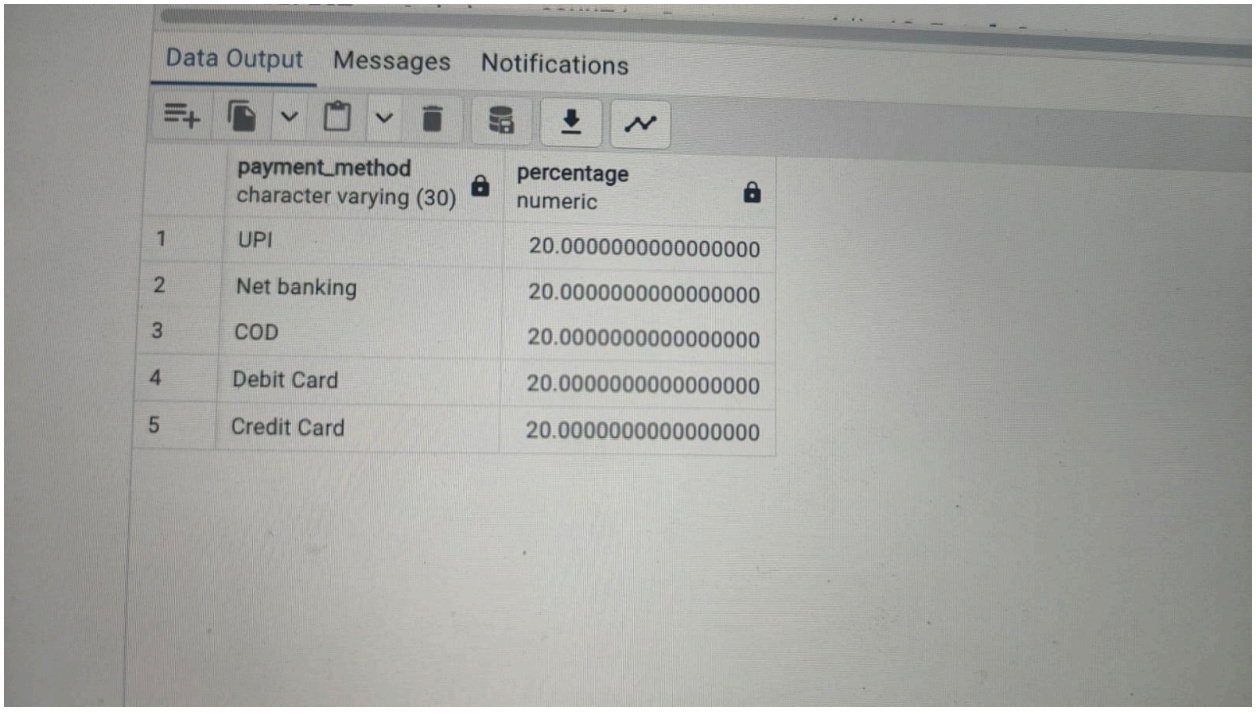
Query-14: Percentage of Orders with Each Payment Method

Relational Algebra:



SQL: SELECT o.Payment_Status AS Payment_Method,
(COUNT(o.Order_id) * 100.0 / (SELECT COUNT(*) FROM Orders)) AS Percentage
FROM Orders o
GROUP BY o.Payment_Status;

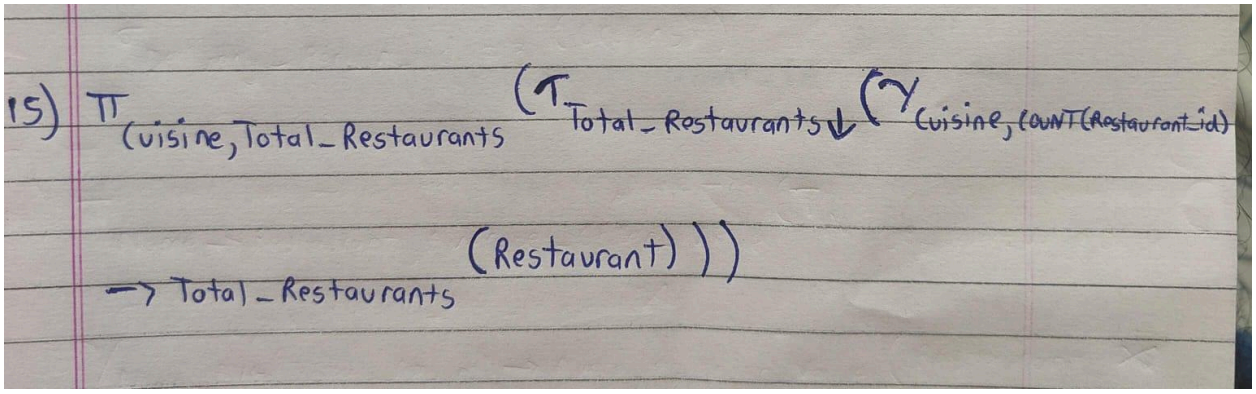
Output:



	payment_method character varying (30)	percentage numeric
1	UPI	20.0000000000000000
2	Net banking	20.0000000000000000
3	COD	20.0000000000000000
4	Debit Card	20.0000000000000000
5	Credit Card	20.0000000000000000

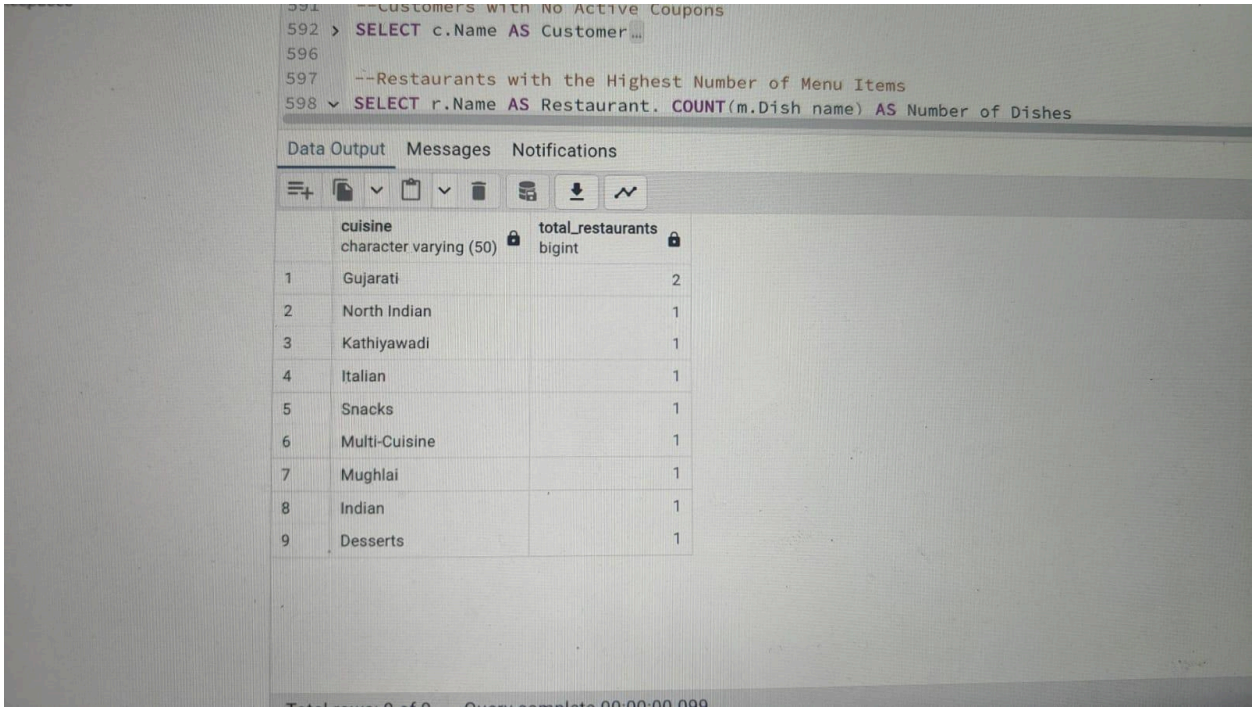
Query-15: Most Popular Cuisine Type

Relational Algebra:



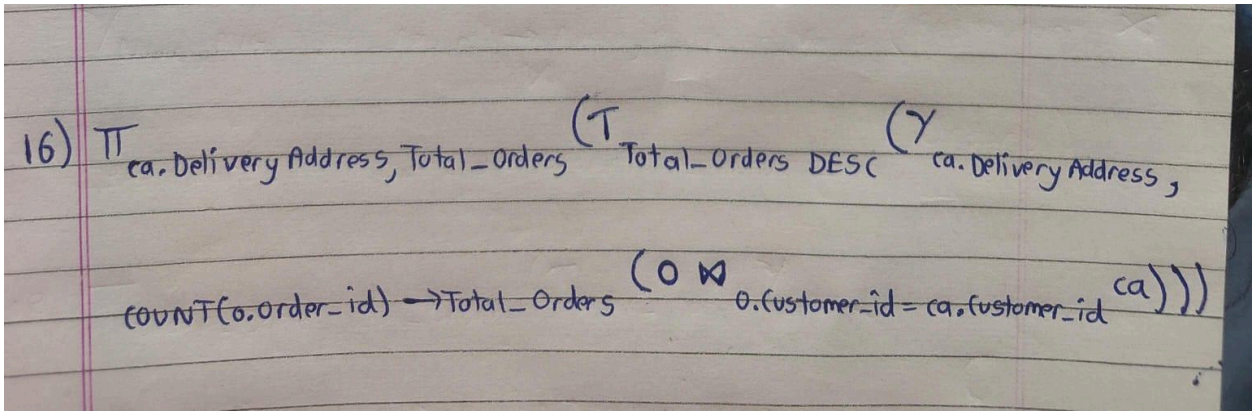
SQL: SELECT r.Cuisine, COUNT(r.Restaurant_id) AS Total_Restaurants
FROM Restaurant r
GROUP BY r.Cuisine
ORDER BY Total_Restaurants DESC;

Output:



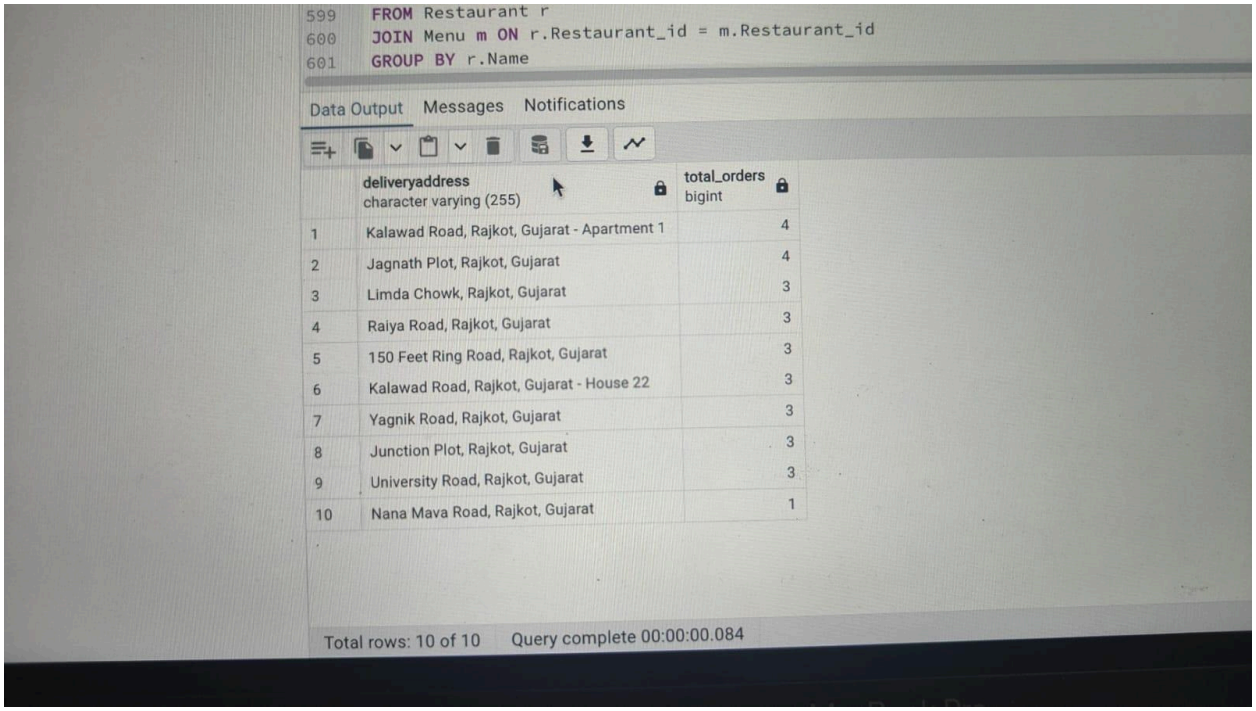
Query-16: Most Common Delivery Locations

Relational Algebra:



SQL: SELECT ca.DeliveryAddress, COUNT(o.Order_id) AS Total_Orders
FROM Orders o
JOIN CustomerAddresses ca ON o.Customer_id = ca.CustomerID
GROUP BY ca.DeliveryAddress
ORDER BY Total_Orders DESC;

Output:



```
599  FROM Restaurant r
600  JOIN Menu m ON r.Restaurant_id = m.Restaurant_id
601  GROUP BY r.Name
```

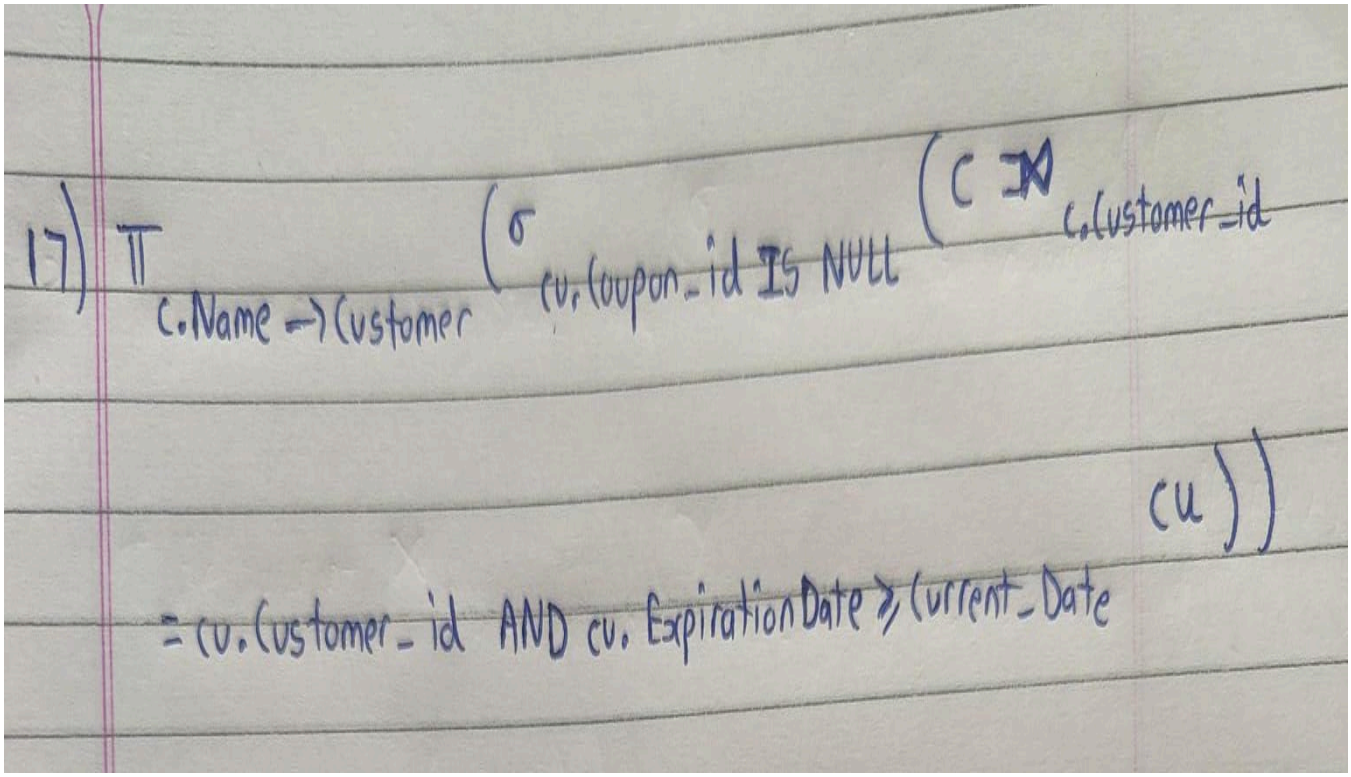
Data Output | Messages | Notifications

	deliveryaddress character varying (255)	total_orders bigint
1	Kalawad Road, Rajkot, Gujarat - Apartment 1	4
2	Jagnath Plot, Rajkot, Gujarat	4
3	Limda Chowk, Rajkot, Gujarat	3
4	Raiya Road, Rajkot, Gujarat	3
5	150 Feet Ring Road, Rajkot, Gujarat	3
6	Kalawad Road, Rajkot, Gujarat - House 22	3
7	Yagnik Road, Rajkot, Gujarat	3
8	Junction Plot, Rajkot, Gujarat	3
9	University Road, Rajkot, Gujarat	3
10	Nana Mava Road, Rajkot, Gujarat	1

Total rows: 10 of 10 Query complete 00:00:00.084

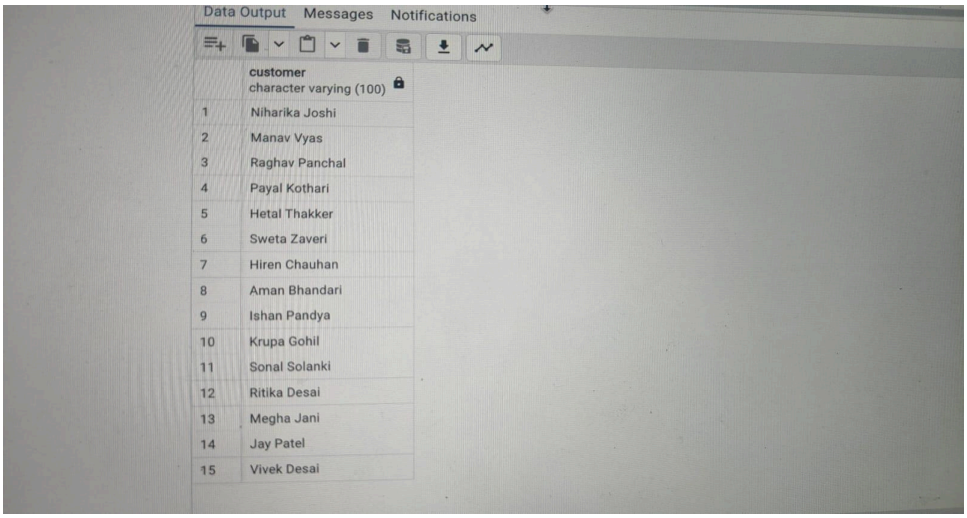
Query-17: Customers with No Active Coupons

Relational Algebra:



SQL: SELECT c.Name AS Customer
FROM Customer c
LEFT JOIN Coupons cu ON c.Customer_id = cu.Customer_id AND cu.ExpirationDate >= CURRENT_DATE
WHERE cu.Coupon_id IS NULL;

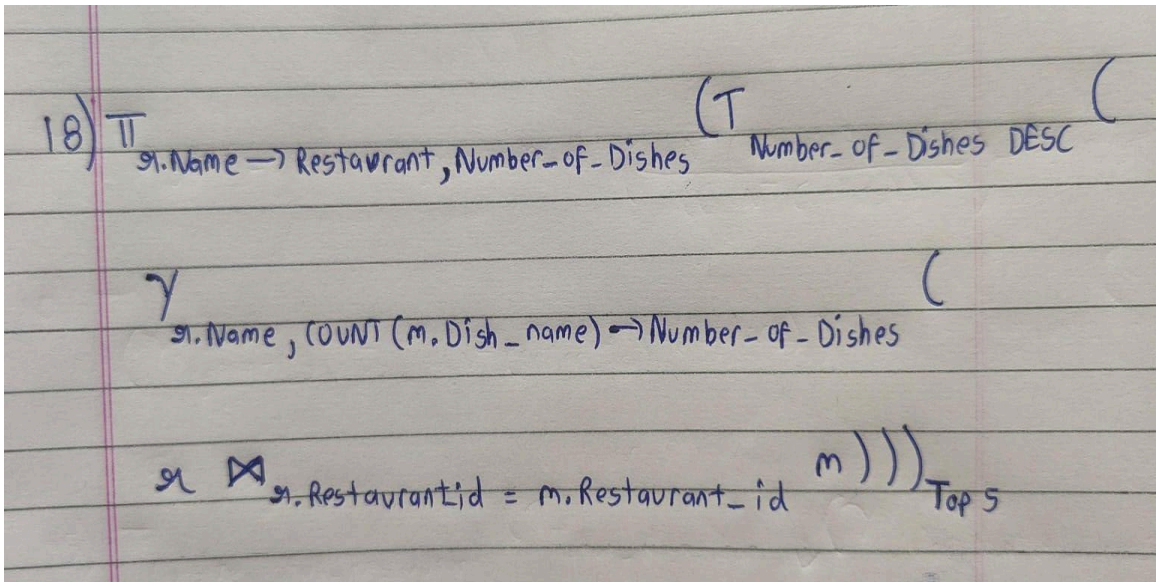
Output:


Data Output Messages Notifications

	customer character varying (100)
1	Niharika Joshi
2	Manav Vyas
3	Raghav Panchal
4	Payal Kothari
5	Hetal Thakker
6	Sweta Zaveri
7	Hiren Chauhan
8	Aman Bhandari
9	Ishan Pandya
10	Krupa Gohil
11	Sonal Solanki
12	Ritika Desai
13	Megha Jani
14	Jay Patel
15	Vivek Desai

Query-18: Restaurants with the Highest Number of Menu Items

Relational Algebra:



SQL: SELECT r.Name AS Restaurant, COUNT(m.Dish_name) AS Number_of_Dishes
FROM Restaurant r
JOIN Menu m ON r.Restaurant_id = m.Restaurant_id
GROUP BY r.Name
ORDER BY Number_of_Dishes DESC
LIMIT 5;

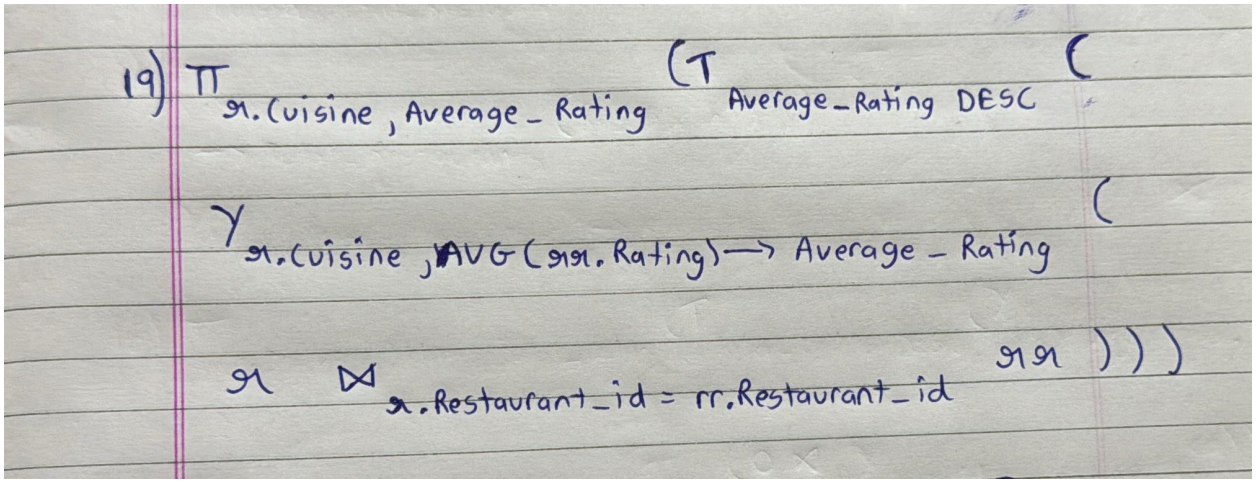
Output:

Data Output | Messages | Notifications

	restaurant character varying (100)	number_of_dishes bigint
1	Rangoli Rasoi	13
2	Leato	13
3	The Fusion Factory	11
4	Imperial Palace	11
5	Rajkot Rasoi	10

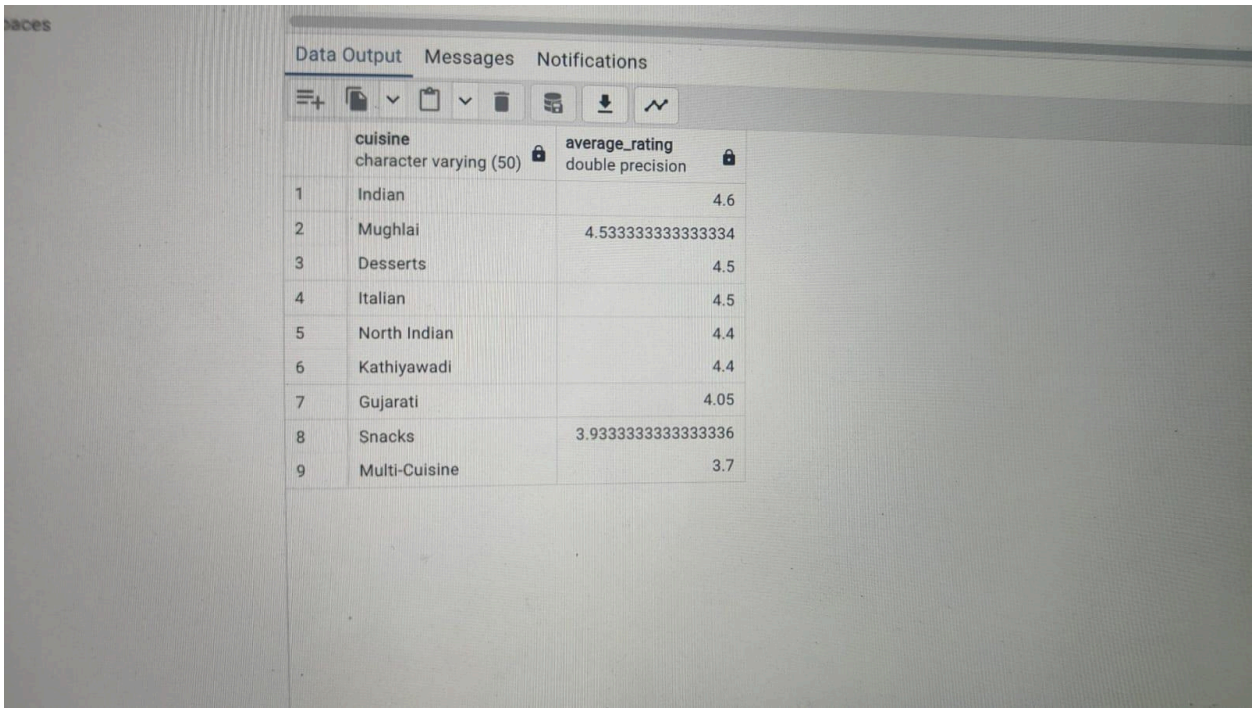
Query-19: Average Rating Per Cuisine Type

Relational Algebra:



SQL:
```
SELECT r.Cuisine, AVG(rr.Rating) AS Average_Rating
FROM Restaurant r
JOIN Ratings_and_Review rr ON r.Restaurant_id = rr.Restaurant_id
GROUP BY r.Cuisine
ORDER BY Average_Rating DESC;
```

Output:

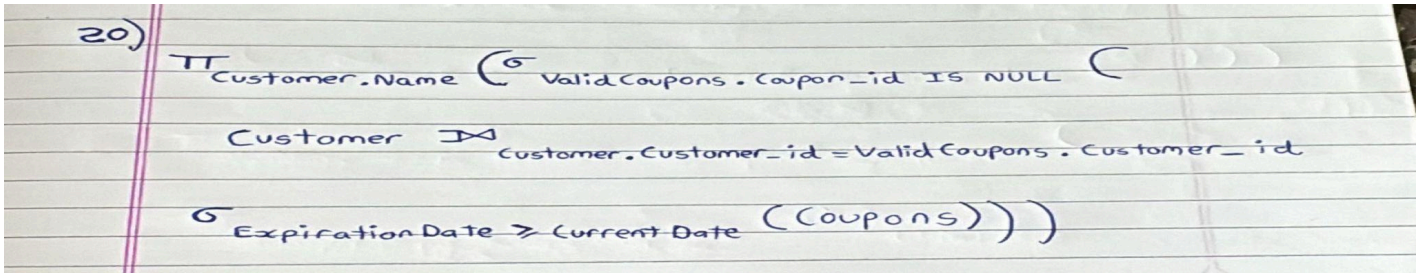


Data Output | Messages | Notifications

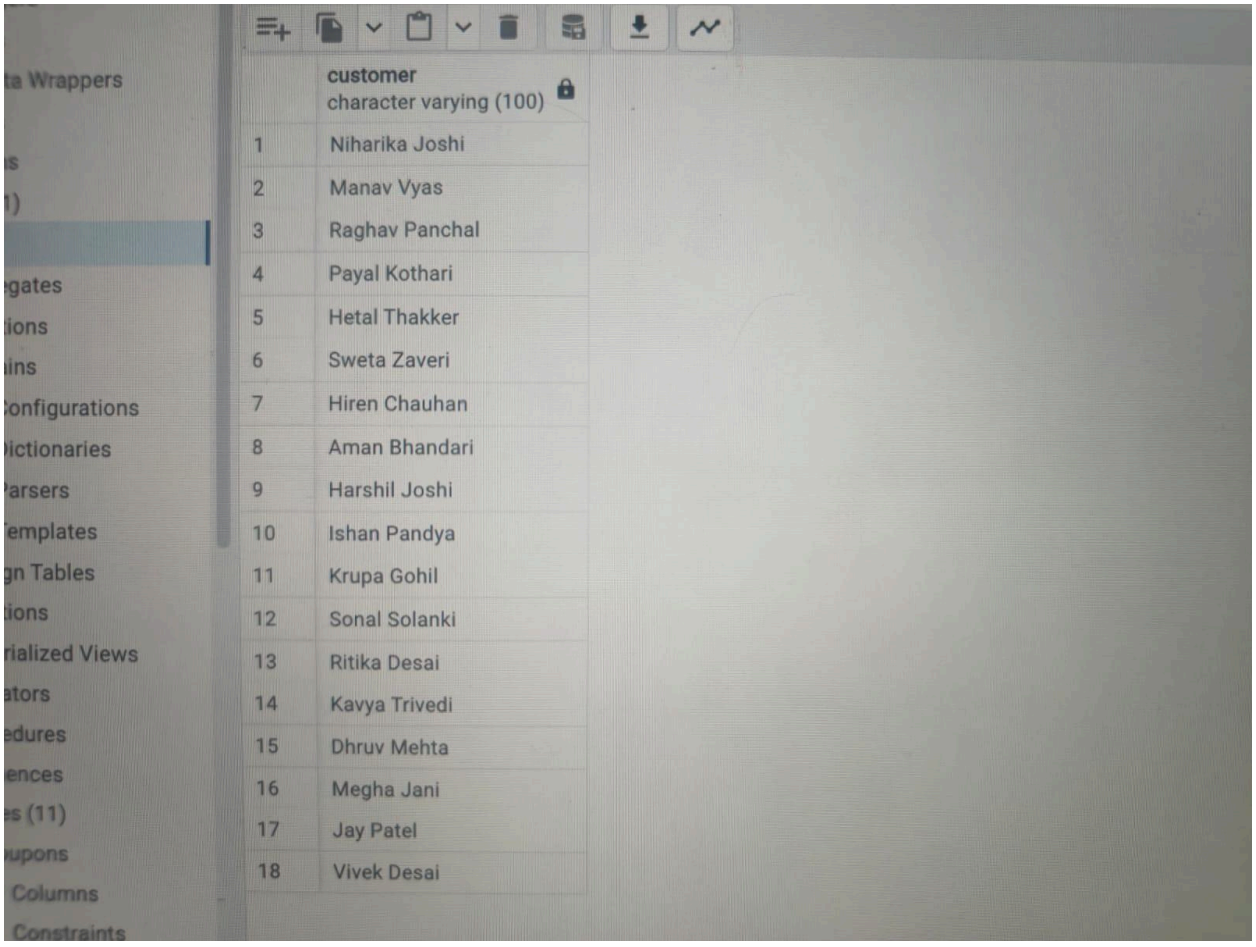
	cuisine character varying (50)	average_rating double precision
1	Indian	4.6
2	Mughlai	4.533333333333334
3	Desserts	4.5
4	Italian	4.5
5	North Indian	4.4
6	Kathiyawadi	4.4
7	Gujarati	4.05
8	Snacks	3.9333333333333336
9	Multi-Cuisine	3.7

Query-20: Identify Customers with Active Coupons

Relational Algebra:



SQL: SELECT c.Name AS Customer
FROM Customer c
LEFT JOIN Coupons cu ON c.Customer_id = cu.Customer_id AND cu.ExpirationDate >= CURRENT_DATE
WHERE cu.Coupon_id IS NULL;
Output:



	customer character varying (100)
1	Niharika Joshi
2	Manav Vyas
3	Raghav Panchal
4	Payal Kothari
5	Hetal Thakker
6	Sweta Zaveri
7	Hiren Chauhan
8	Aman Bhandari
9	Harshil Joshi
10	Ishan Pandya
11	Krupa Gohil
12	Sonal Solanki
13	Ritika Desai
14	Kavya Trivedi
15	Dhruv Mehta
16	Megha Jani
17	Jay Patel
18	Vivek Desai