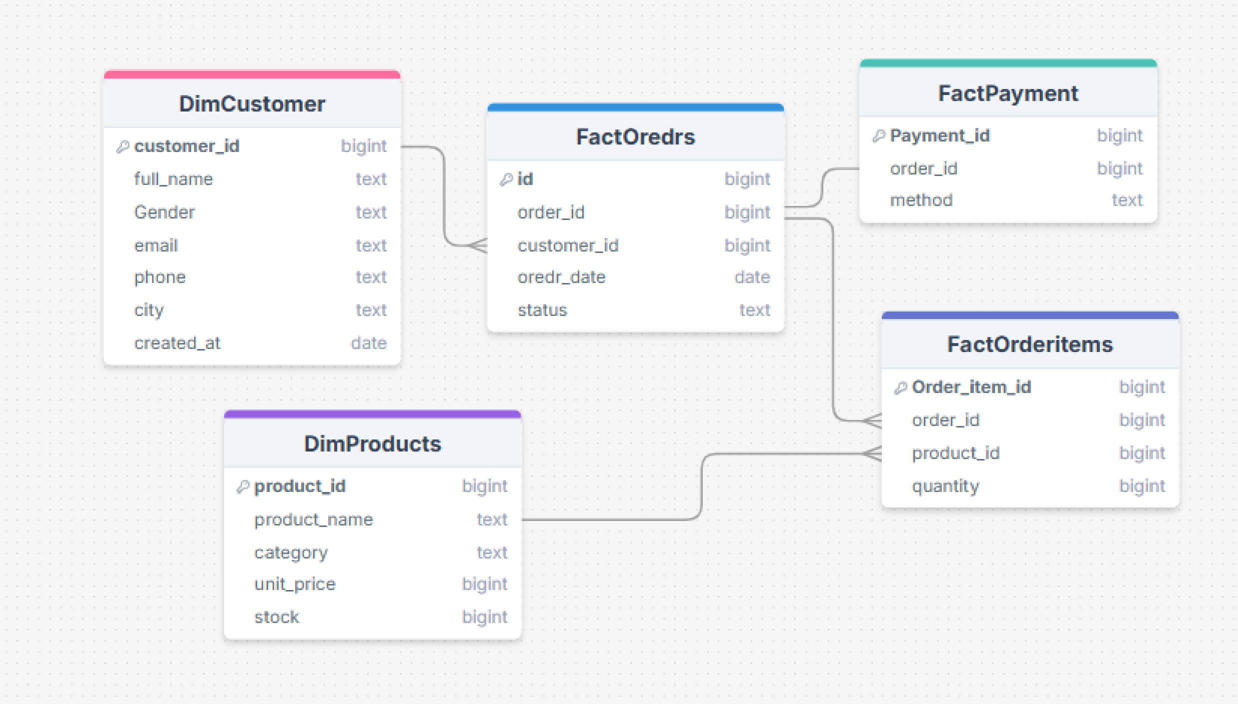
UrbanCart Retail Shop Performance and Customer Behavior Analysis Using PostgreSQL

Overview

This project analyzes UrbanCart's sales and customer data to evaluate business performance and purchasing behavior. The analysis identifies key revenue drivers, customer trends, and product performance. The insights support data-driven decisions to improve sales, inventory planning, and customer engagement.

ER Diagram



DimCustomer	
customer_id	bigint
full_name	text
Gender	text
email	text
phone	text
city	text
created_at	date

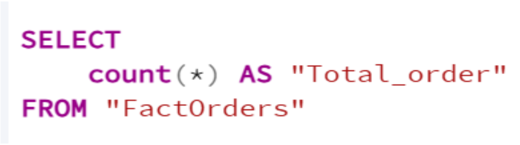
FactOredrs	
id	bigint
order_id	bigint
customer_id	bigint
oredr_date	date
status	text

FactPayment	
Payment_id	bigint
order_id	bigint
method	text

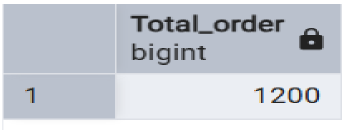
FactOrderitems	
Order_item_id	bigint
order_id	bigint
product_id	bigint
quantity	bigint

DimProducts	
product_id	bigint
product_name	text
category	text
unit_price	bigint
stock	bigint

1. How many total orders has UrbanCart received so far?



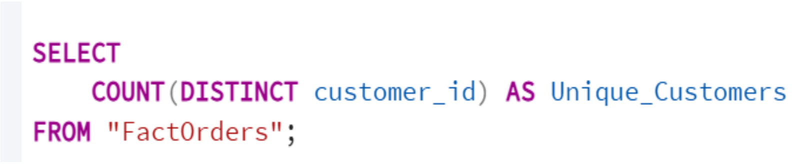
```
SELECT
    count(*) AS "Total_order"
FROM "FactOrders"
```



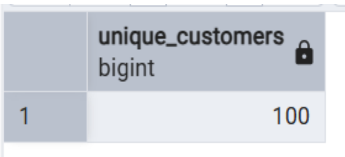
	Total_order bigint
1	1200

- UrbanCart has received a total of 1200 orders

2. How many unique customers have placed at least one order?



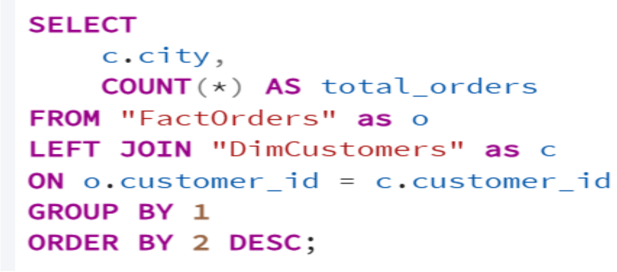
```
SELECT
    COUNT(DISTINCT customer_id) AS Unique_Customers
FROM "FactOrders";
```



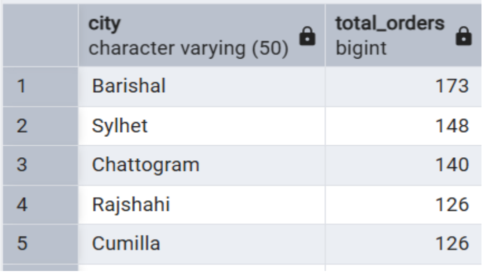
	unique_customers bigint
1	100

- 100 unique customers have placed at least one order

3. Which cities generate the highest number of orders?



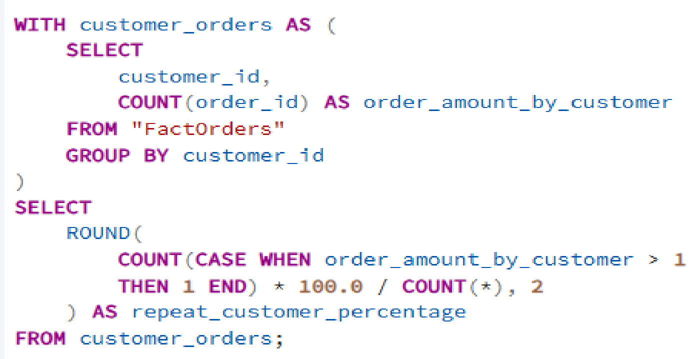
```
SELECT
    c.city,
    COUNT(*) AS total_orders
FROM "FactOrders" as o
LEFT JOIN "DimCustomers" as c
ON o.customer_id = c.customer_id
GROUP BY 1
ORDER BY 2 DESC;
```



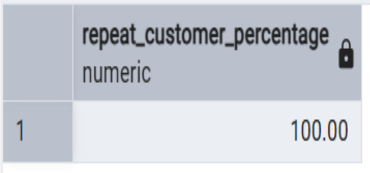
	city character varying (50)	total_orders bigint
1	Barishal	173
2	Sylhet	148
3	Chattogram	140
4	Rajshahi	126
5	Cumilla	126

- Barishal generates the highest number of orders among all cities.

4. What percentage of customers have placed more than one order?



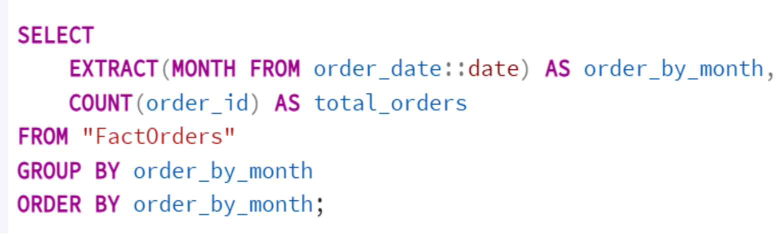
```
WITH customer_orders AS (
    SELECT
        customer_id,
        COUNT(order_id) AS order_amount_by_customer
    FROM "FactOrders"
    GROUP BY customer_id
)
SELECT
    ROUND(
        COUNT(CASE WHEN order_amount_by_customer > 1
        THEN 1 END) * 100.0 / COUNT(*), 2
    ) AS repeat_customer_percentage
FROM customer_orders;
```



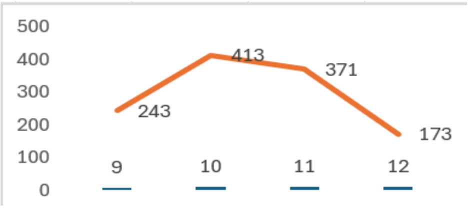
	repeat_customer_percentage numeric
1	100.00

- 100% of customers have placed more than one order, indicating strong repeat purchasing behavior across the customer base.

5. What is the monthly trend for total orders over time?

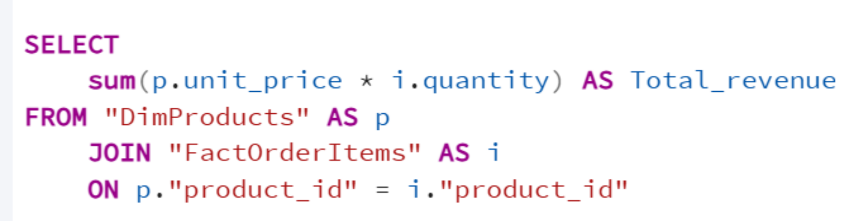


```sql
SELECT
    EXTRACT(MONTH FROM order_date::date) AS order_by_month,
    COUNT(order_id) AS total_orders
FROM "FactOrders"
GROUP BY order_by_month
ORDER BY order_by_month;
```

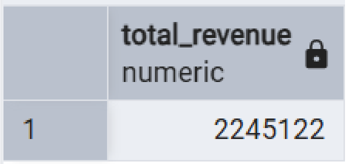


- Order volume peaks in one month, highlighting seasonal demand patterns.

6. What is the total revenue generated by UrbanCart?



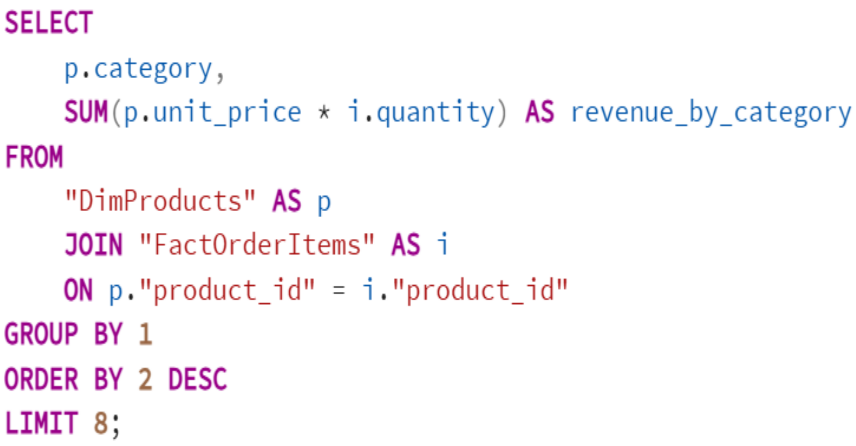
```sql
SELECT
    sum(p.unit_price * i.quantity) AS Total_revenue
FROM "DimProducts" AS p
    JOIN "FactOrderItems" AS i
    ON p."product_id" = i."product_id"
```



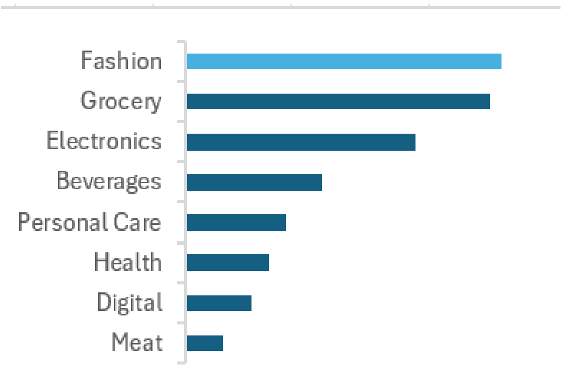
	total_revenue numeric 🔒
1	2245122

- UrbanCart has generated a total revenue of 2,245,122, reflecting strong overall sales performance.

7. Which product categories contribute the most to total revenue?

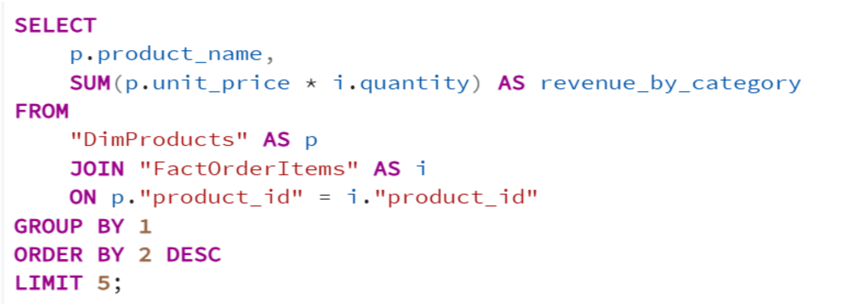


```sql
SELECT
    p.category,
    SUM(p.unit_price * i.quantity) AS revenue_by_category
FROM
    "DimProducts" AS p
    JOIN "FactOrderItems" AS i
    ON p."product_id" = i."product_id"
GROUP BY 1
ORDER BY 2 DESC
LIMIT 8;
```

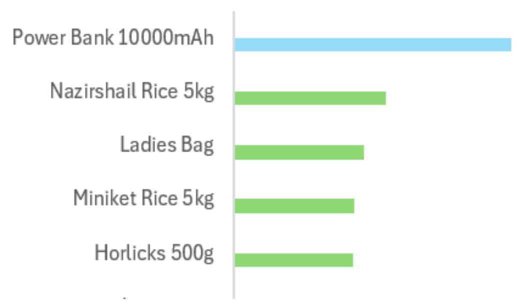


- Fashion and Grocery contribute the highest share of total revenue, making them the top-performing product categories.

8. Which individual products generate the highest revenue?

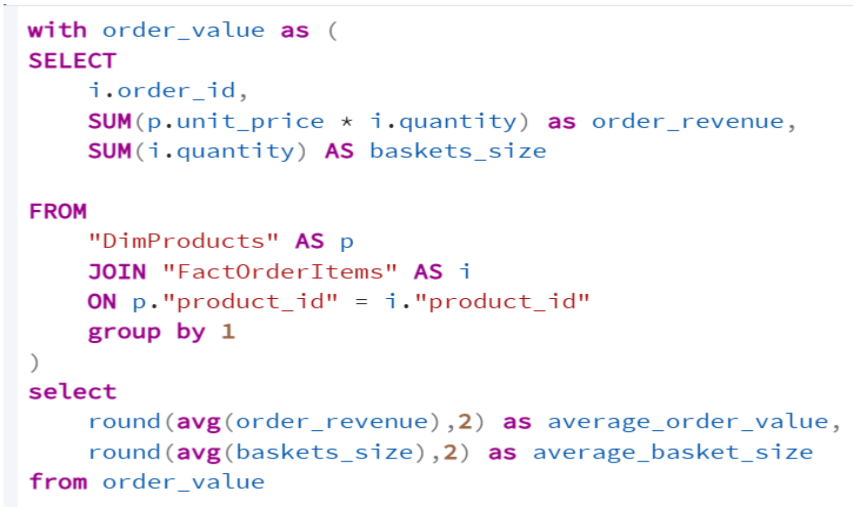


```sql
SELECT
    p.product_name,
    SUM(p.unit_price * i.quantity) AS revenue_by_category
FROM
    "DimProducts" AS p
    JOIN "FactOrderItems" AS i
    ON p."product_id" = i."product_id"
GROUP BY 1
ORDER BY 2 DESC
LIMIT 5;
```

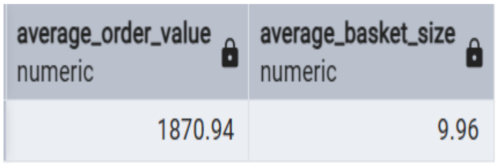


- **The Power Bank 10000mAh generates the highest revenue.**

9. What is the average order value (AOV) and Average Basket Size?



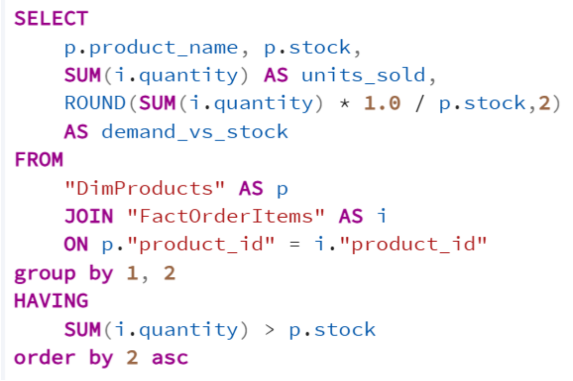
```sql
with order_value as (
SELECT
    i.order_id,
    SUM(p.unit_price * i.quantity) as order_revenue,
    SUM(i.quantity) AS baskets_size

FROM
    "DimProducts" AS p
    JOIN "FactOrderItems" AS i
    ON p."product_id" = i."product_id"
    group by 1
)
select
    round(avg(order_revenue),2) as average_order_value,
    round(avg(baskets_size),2) as average_basket_size
from order_value
```



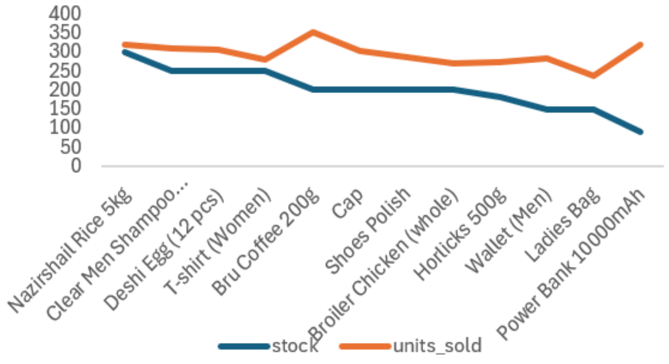
average_order_value numeric	average_basket_size numeric
1870.94	9.96

- **The average order value is 1,870.94, with an average basket size of 9.96 items**

10. Which products are at risk of stock-out due to high sales volume and low inventory?

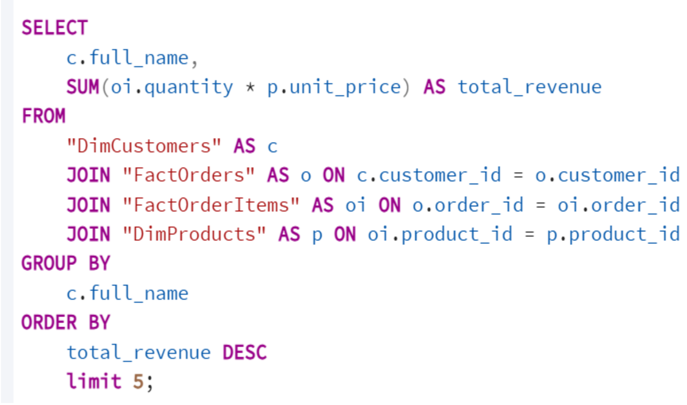


```sql
SELECT
    p.product_name, p.stock,
    SUM(i.quantity) AS units_sold,
    ROUND(SUM(i.quantity) * 1.0 / p.stock,2)
    AS demand_vs_stock
FROM
    "DimProducts" AS p
    JOIN "FactOrderItems" AS i
    ON p."product_id" = i."product_id"
group by 1, 2
HAVING
    SUM(i.quantity) > p.stock
order by 2 asc
```

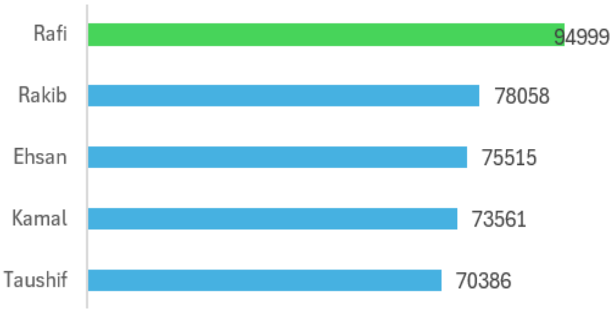


- **Products with sales volumes exceeding current stock levels are at high risk of stock-outs, highlighting the nee for timely inventory replenishment.**

11. Which customers contribute the highest total revenue?

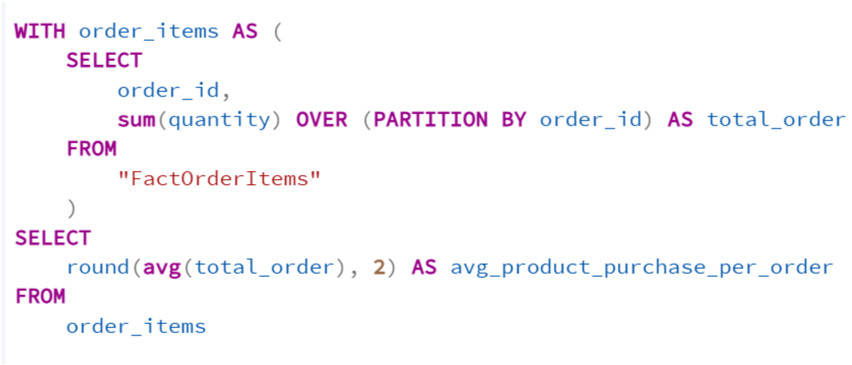


```
SELECT
    c.full_name,
    SUM(oi.quantity * p.unit_price) AS total_revenue
FROM
    "DimCustomers" AS c
    JOIN "FactOrders" AS o ON c.customer_id = o.customer_id
    JOIN "FactOrderItems" AS oi ON o.order_id = oi.order_id
    JOIN "DimProducts" AS p ON oi.product_id = p.product_id
GROUP BY
    c.full_name
ORDER BY
    total_revenue DESC
    limit 5;
```

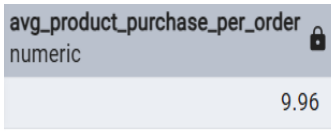


- **Rafi generating the most revenue among all customers.**

12. What is the average number of products purchased per order?



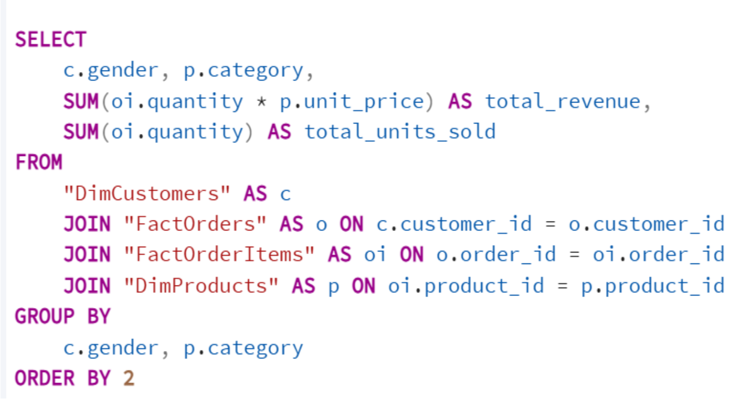
```
WITH order_items AS (
    SELECT
        order_id,
        sum(quantity) OVER (PARTITION BY order_id) AS total_order
    FROM
        "FactOrderItems"
    )
SELECT
    round(avg(total_order), 2) AS avg_product_purchase_per_order
FROM
    order_items
```



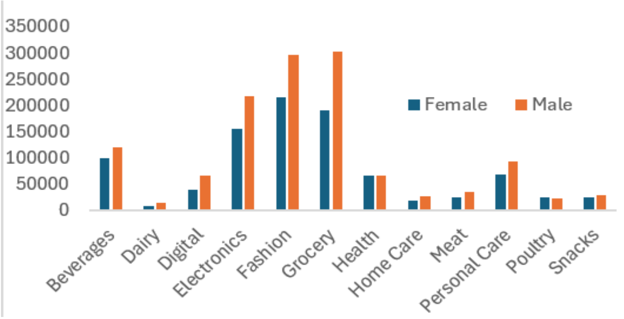
avg_product_purchase_per_order numeric
9.96

- **The average number of products purchased per order is 9.96.**

13. Do male and female customers show different purchasing patterns by category?

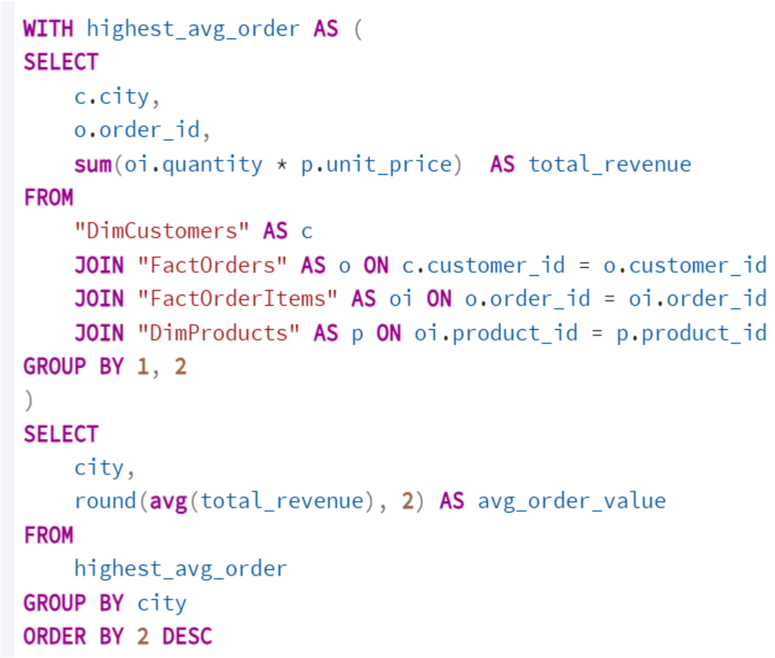


```
SELECT
    c.gender, p.category,
    SUM(oi.quantity * p.unit_price) AS total_revenue,
    SUM(oi.quantity) AS total_units_sold
FROM
    "DimCustomers" AS c
    JOIN "FactOrders" AS o ON c.customer_id = o.customer_id
    JOIN "FactOrderItems" AS oi ON o.order_id = oi.order_id
    JOIN "DimProducts" AS p ON oi.product_id = p.product_id
GROUP BY
    c.gender, p.category
ORDER BY 2
```

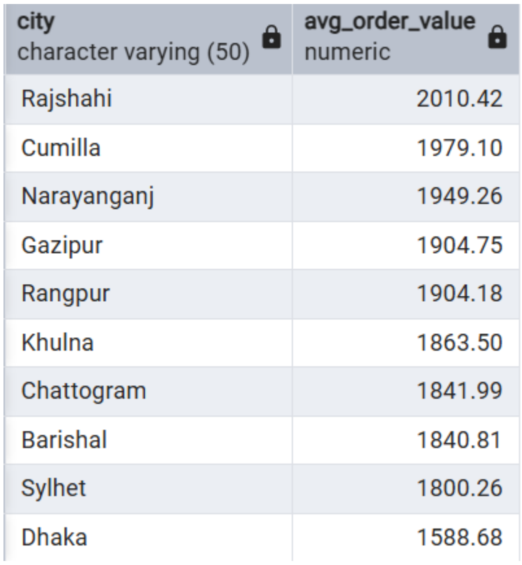


- **Male and female customers show noticeable differences in purchasing behavior across categories, with variation in both sales volume and revenue contribution.**

14. Which cities have the highest average order value?



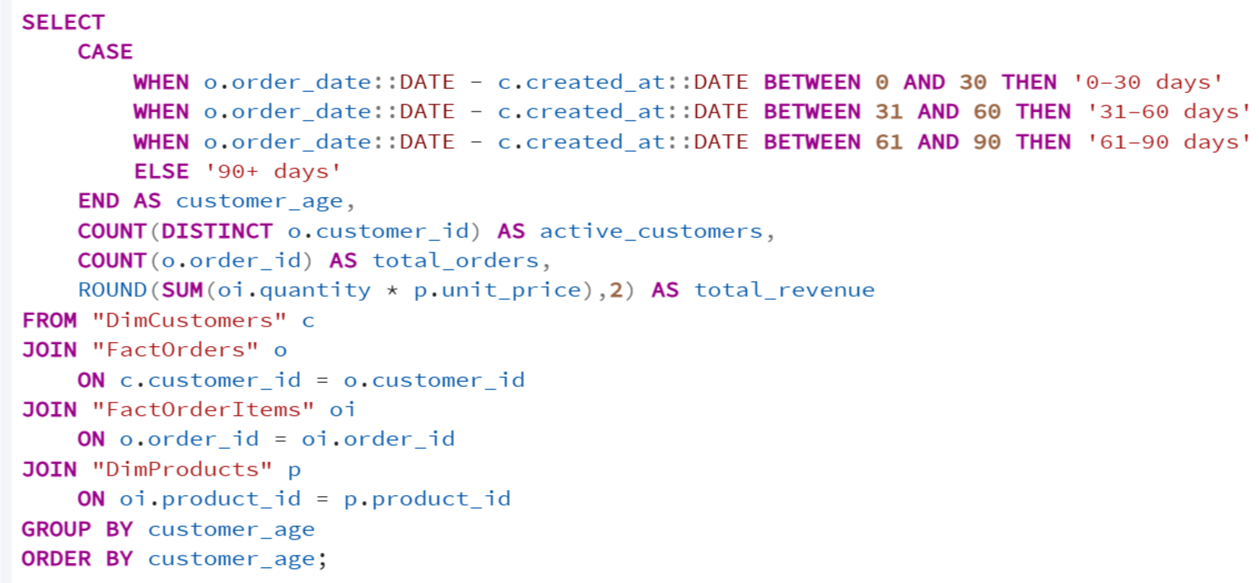
```
WITH highest_avg_order AS (
SELECT
    c.city,
    o.order_id,
    sum(oi.quantity * p.unit_price)  AS total_revenue
FROM
    "DimCustomers" AS c
    JOIN "FactOrders" AS o ON c.customer_id = o.customer_id
    JOIN "FactOrderItems" AS oi ON o.order_id = oi.order_id
    JOIN "DimProducts" AS p ON oi.product_id = p.product_id
GROUP BY 1, 2
)
SELECT
    city,
    round(avg(total_revenue), 2) AS avg_order_value
FROM
    highest_avg_order
GROUP BY city
ORDER BY 2 DESC
```



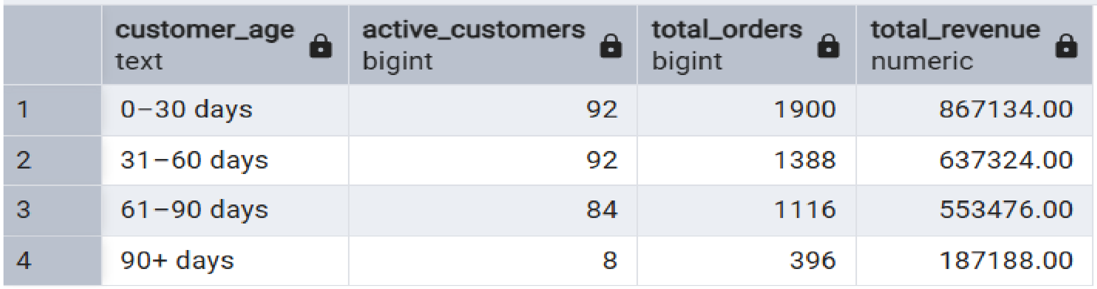
city character varying (50)	avg_order_value numeric
Rajshahi	2010.42
Cumilla	1979.10
Narayanganj	1949.26
Gazipur	1904.75
Rangpur	1904.18
Khulna	1863.50
Chattogram	1841.99
Barishal	1840.81
Sylhet	1800.26
Dhaka	1588.68

- **Rajshahi records the highest average order value, indicating higher per-order spending compared to other cities.**

15. How does customer purchasing behavior changed over time since account creation?



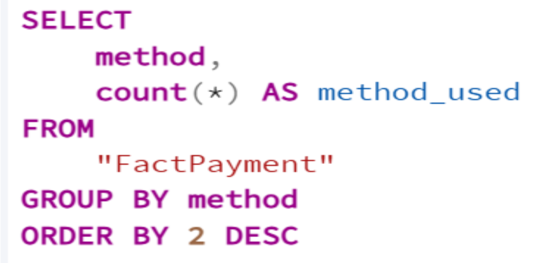
```
SELECT
    CASE
        WHEN o.order_date::DATE - c.created_at::DATE BETWEEN 0 AND 30 THEN '0-30 days'
        WHEN o.order_date::DATE - c.created_at::DATE BETWEEN 31 AND 60 THEN '31-60 days'
        WHEN o.order_date::DATE - c.created_at::DATE BETWEEN 61 AND 90 THEN '61-90 days'
        ELSE '90+ days'
    END AS customer_age,
    COUNT(DISTINCT o.customer_id) AS active_customers,
    COUNT(o.order_id) AS total_orders,
    ROUND(SUM(oi.quantity * p.unit_price),2) AS total_revenue
FROM "DimCustomers" c
JOIN "FactOrders" o
    ON c.customer_id = o.customer_id
JOIN "FactOrderItems" oi
    ON o.order_id = oi.order_id
JOIN "DimProducts" p
    ON oi.product_id = p.product_id
GROUP BY customer_age
ORDER BY customer_age;
```



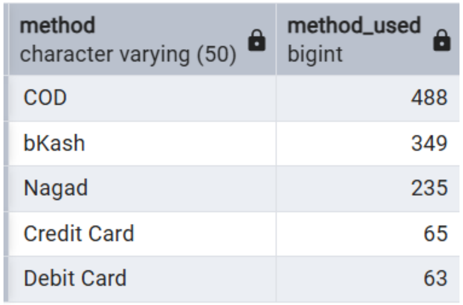
	customer_age text	active_customers bigint	total_orders bigint	total_revenue numeric
1	0–30 days	92	1900	867134.00
2	31–60 days	92	1388	637324.00
3	61–90 days	84	1116	553476.00
4	90+ days	8	396	187188.00

- **Customer engagement and spending are strongest in the early lifecycle (0–60 days) and decline over time, highlighting the importance of early retention efforts.**

16. Which payment methods are used most frequently?



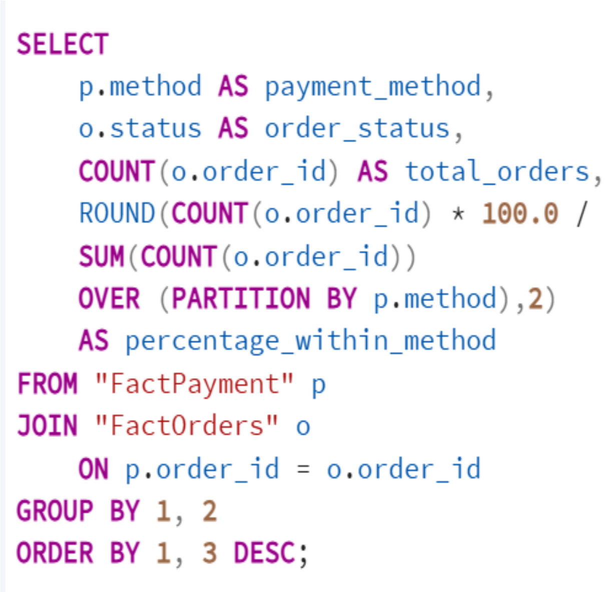
```
SELECT
    method,
    count(*) AS method_used
FROM
    "FactPayment"
GROUP BY method
ORDER BY 2 DESC
```



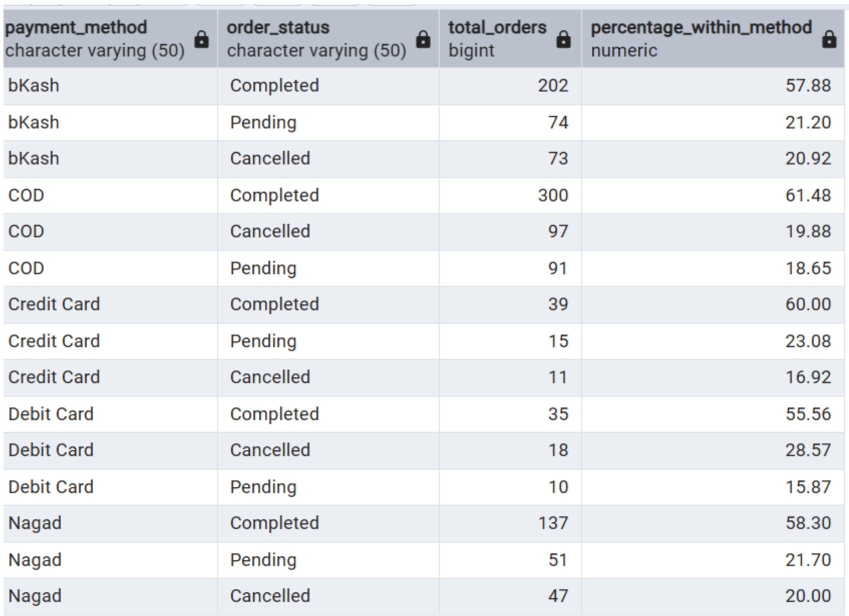
method character varying (50)	method_used bigint
COD	488
bKash	349
Nagad	235
Credit Card	65
Debit Card	63

- **Cash on Delivery (COD) is the most frequently used payment method, followed by bKash and Nagad**

17. Is there any relationship between payment method and order status?



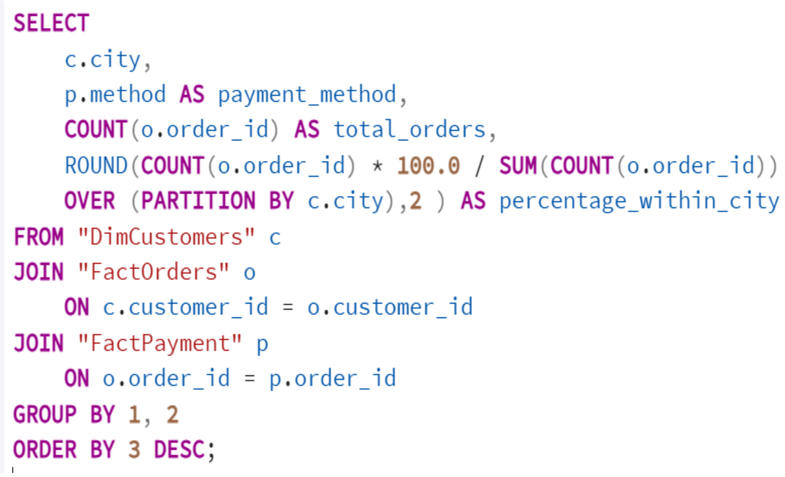
```
SELECT
    p.method AS payment_method,
    o.status AS order_status,
    COUNT(o.order_id) AS total_orders,
    ROUND(COUNT(o.order_id) * 100.0 /
    SUM(COUNT(o.order_id))
    OVER (PARTITION BY p.method),2)
    AS percentage_within_method
FROM "FactPayment" p
JOIN "FactOrders" o
    ON p.order_id = o.order_id
GROUP BY 1, 2
ORDER BY 1, 3 DESC;
```



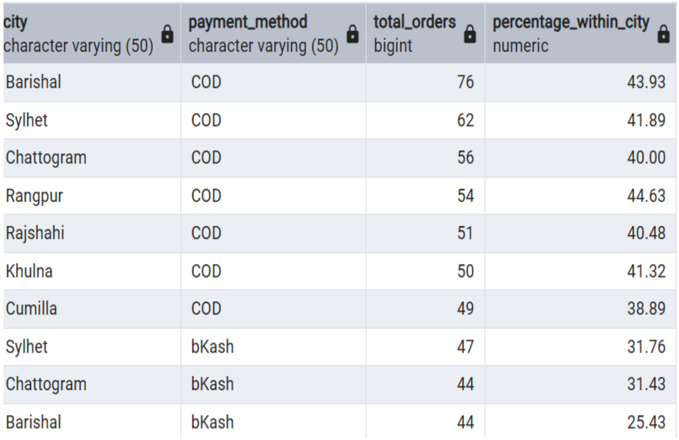
payment_method character varying (50)	order_status character varying (50)	total_orders bigint	percentage_within_method numeric
bKash	Completed	202	57.88
bKash	Pending	74	21.20
bKash	Cancelled	73	20.92
COD	Completed	300	61.48
COD	Cancelled	97	19.88
COD	Pending	91	18.65
Credit Card	Completed	39	60.00
Credit Card	Pending	15	23.08
Credit Card	Cancelled	11	16.92
Debit Card	Completed	35	55.56
Debit Card	Cancelled	18	28.57
Debit Card	Pending	10	15.87
Nagad	Completed	137	58.30
Nagad	Pending	51	21.70
Nagad	Cancelled	47	20.00

- **Order completion rates vary by payment method, with COD and digital payments showing higher completion compared to cancellations and pending orders**

18. Do certain cities prefer specific payment methods?



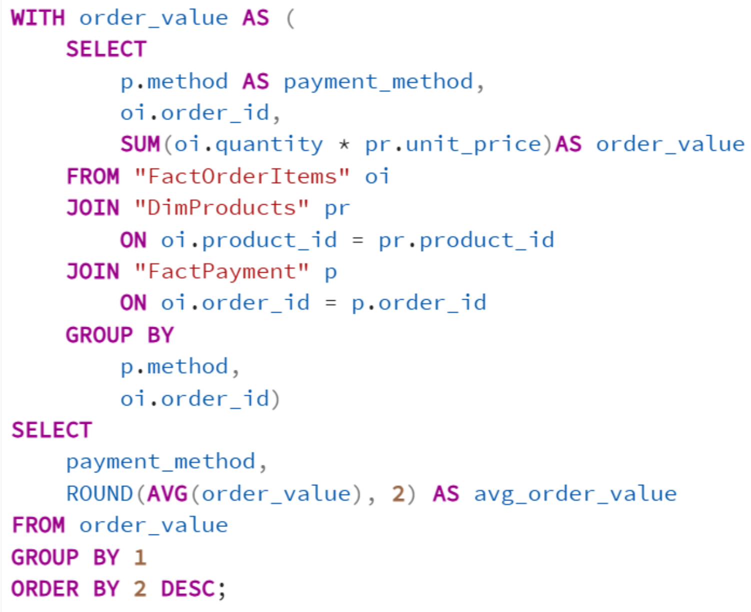
```sql
SELECT
    c.city,
    p.method AS payment_method,
    COUNT(o.order_id) AS total_orders,
    ROUND(COUNT(o.order_id) * 100.0 / SUM(COUNT(o.order_id))
    OVER (PARTITION BY c.city),2 ) AS percentage_within_city
FROM "DimCustomers" c
JOIN "FactOrders" o
    ON c.customer_id = o.customer_id
JOIN "FactPayment" p
    ON o.order_id = p.order_id
GROUP BY 1, 2
ORDER BY 3 DESC;
```



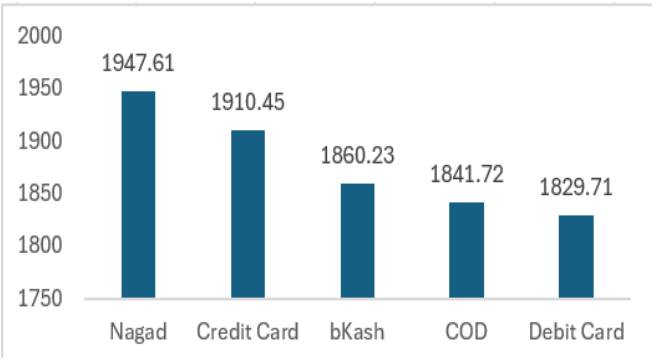
city character varying (50) 🔒	payment_method character varying (50) 🔒	total_orders bigint 🔒	percentage_within_city numeric 🔒
Barishal	COD	76	43.93
Sylhet	COD	62	41.89
Chattogram	COD	56	40.00
Rangpur	COD	54	44.63
Rajshahi	COD	51	40.48
Khulna	COD	50	41.32
Cumilla	COD	49	38.89
Sylhet	bKash	47	31.76
Chattogram	bKash	44	31.43
Barishal	bKash	44	25.43

- **Payment method preferences vary by city, with Cash on Delivery dominating across most cities, while mobile payments show moderate regional adoption.**

19. Are higher-value orders associated with specific payment methods?

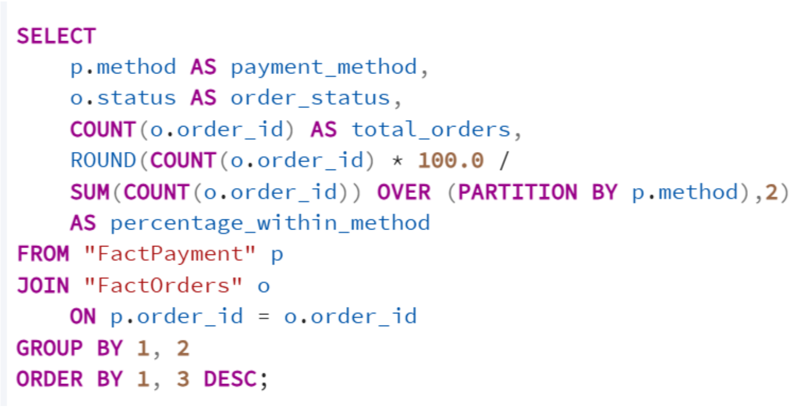


```sql
WITH order_value AS (
    SELECT
        p.method AS payment_method,
        oi.order_id,
        SUM(oi.quantity * pr.unit_price)AS order_value
    FROM "FactOrderItems" oi
    JOIN "DimProducts" pr
        ON oi.product_id = pr.product_id
    JOIN "FactPayment" p
        ON oi.order_id = p.order_id
    GROUP BY
        p.method,
        oi.order_id)
SELECT
    payment_method,
    ROUND(AVG(order_value), 2) AS avg_order_value
FROM order_value
GROUP BY 1
ORDER BY 2 DESC;
```

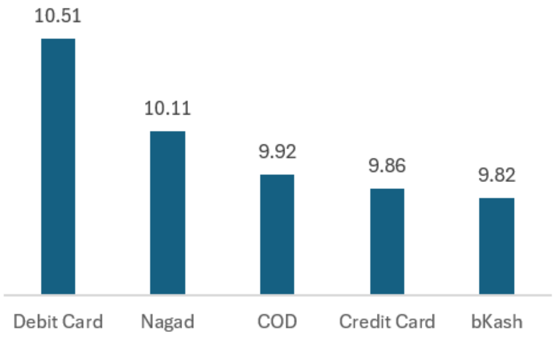


- **Higher average order values are associated with Nagad and Credit Card payments, while COD and Debit Card order tend to have lower order values.**

20. What is the average number of items per order by payment method?



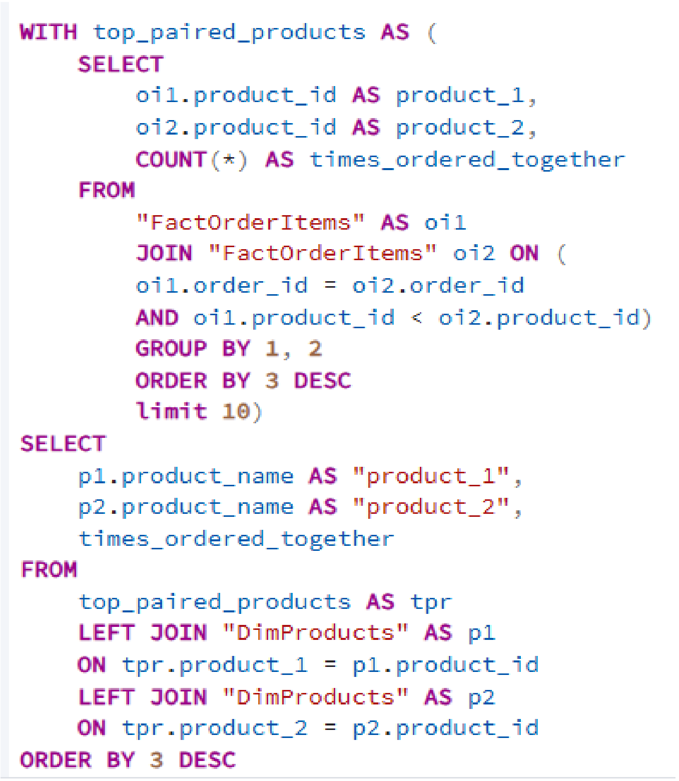
```
SELECT
    p.method AS payment_method,
    o.status AS order_status,
    COUNT(o.order_id) AS total_orders,
    ROUND(COUNT(o.order_id) * 100.0 /
    SUM(COUNT(o.order_id)) OVER (PARTITION BY p.method),2)
    AS percentage_within_method
FROM "FactPayment" p
JOIN "FactOrders" o
    ON p.order_id = o.order_id
GROUP BY 1, 2
ORDER BY 1, 3 DESC;
```



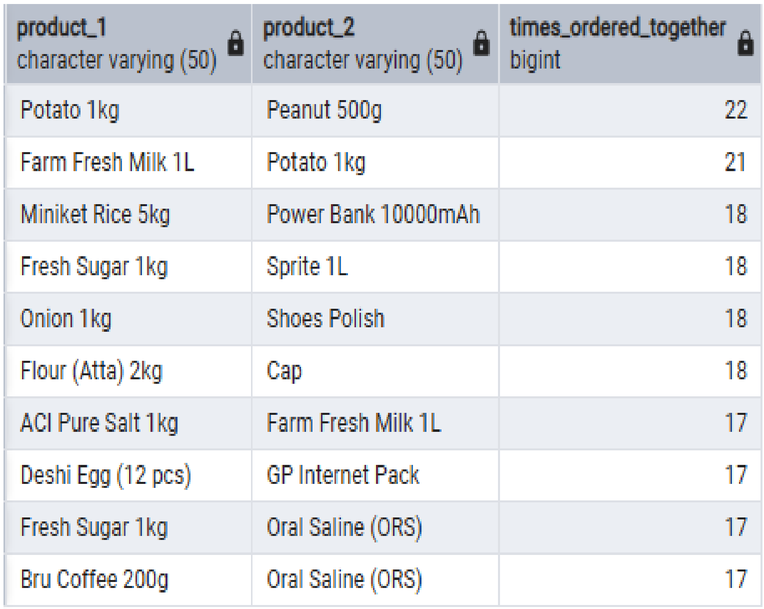
- **Orders paid via Debit Card have the highest average number of items per order, while**

Other payment methods show similar basket sizes.

21. Which products are most frequently ordered together?



```
WITH top_paired_products AS (
    SELECT
        oi1.product_id AS product_1,
        oi2.product_id AS product_2,
        COUNT(*) AS times_ordered_together
    FROM
        "FactOrderItems" AS oi1
        JOIN "FactOrderItems" oi2 ON (
        oi1.order_id = oi2.order_id
        AND oi1.product_id < oi2.product_id)
        GROUP BY 1, 2
        ORDER BY 3 DESC
        limit 10)
SELECT
    p1.product_name AS "product_1",
    p2.product_name AS "product_2",
    times_ordered_together
FROM
    top_paired_products AS tpr
    LEFT JOIN "DimProducts" AS p1
    ON tpr.product_1 = p1.product_id
    LEFT JOIN "DimProducts" AS p2
    ON tpr.product_2 = p2.product_id
ORDER BY 3 DESC
```

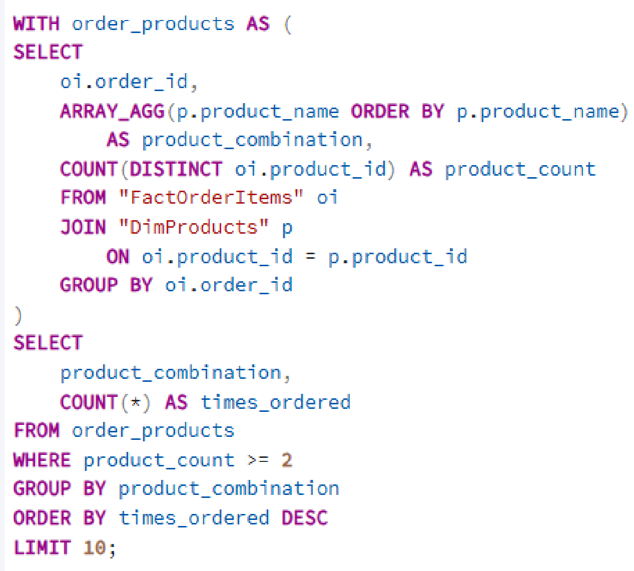


product_1 character varying (50) 🔒	product_2 character varying (50) 🔒	times_ordered_together bigint 🔒
Potato 1kg	Peanut 500g	22
Farm Fresh Milk 1L	Potato 1kg	21
Miniket Rice 5kg	Power Bank 10000mAh	18
Fresh Sugar 1kg	Sprite 1L	18
Onion 1kg	Shoes Polish	18
Flour (Atta) 2kg	Cap	18
ACI Pure Salt 1kg	Farm Fresh Milk 1L	17
Deshi Egg (12 pcs)	GP Internet Pack	17
Fresh Sugar 1kg	Oral Saline (ORS)	17
Bru Coffee 200g	Oral Saline (ORS)	17

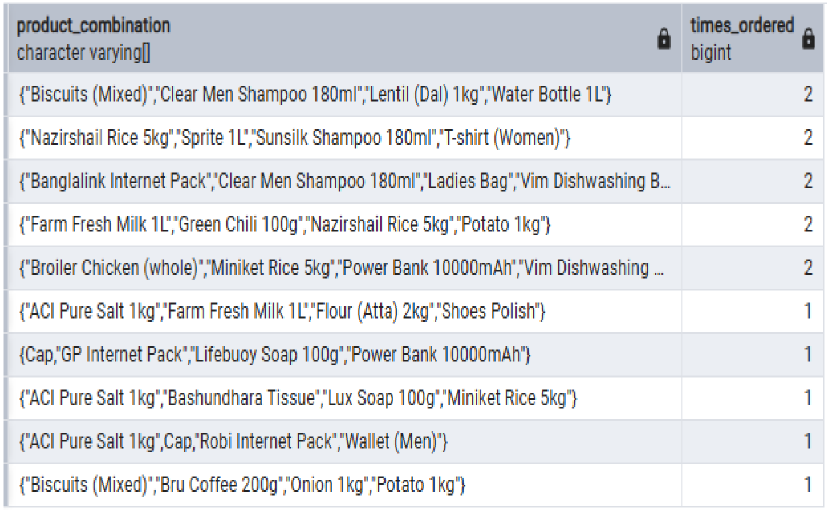
- **Certain product pairs are frequently ordered together, indicating common co-purchase patterns**

that can be leveraged for product bundling and cross-selling.

22. Which products combination are most frequently ordered together?



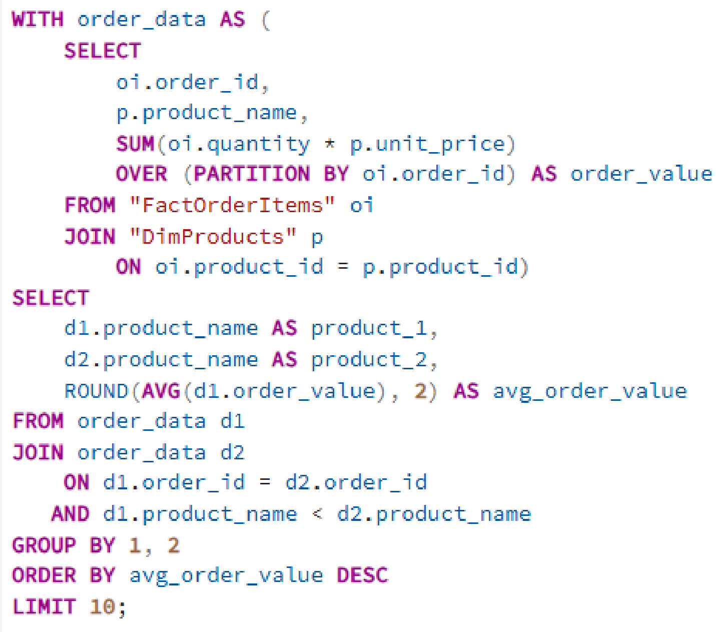
```
WITH order_products AS (
SELECT
    oi.order_id,
    ARRAY_AGG(p.product_name ORDER BY p.product_name)
        AS product_combination,
    COUNT(DISTINCT oi.product_id) AS product_count
    FROM "FactOrderItems" oi
    JOIN "DimProducts" p
        ON oi.product_id = p.product_id
    GROUP BY oi.order_id
)
SELECT
    product_combination,
    COUNT(*) AS times_ordered
FROM order_products
WHERE product_count >= 2
GROUP BY product_combination
ORDER BY times_ordered DESC
LIMIT 10;
```



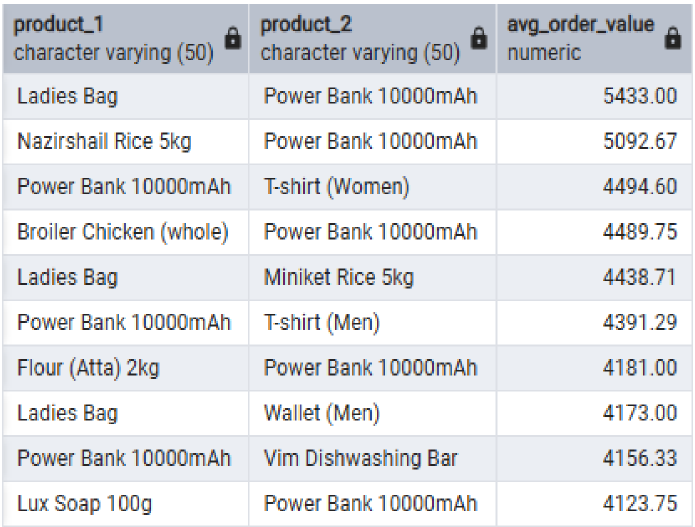
product_combination character varying[]	times_ordered bigint
{"Biscuits (Mixed)","Clear Men Shampoo 180ml","Lentil (Dal) 1kg","Water Bottle 1L"}	2
{"Nazirshail Rice 5kg","Sprite 1L","Sunsilk Shampoo 180ml","T-shirt (Women)"}	2
{"Banglalink Internet Pack","Clear Men Shampoo 180ml","Ladies Bag","Vim Dishwashing B...	2
{"Farm Fresh Milk 1L","Green Chili 100g","Nazirshail Rice 5kg","Potato 1kg"}	2
{"Broiler Chicken (whole)","Miniket Rice 5kg","Power Bank 10000mAh","Vim Dishwashing ...	2
{"ACI Pure Salt 1kg","Farm Fresh Milk 1L","Flour (Atta) 2kg","Shoes Polish"}	1
{Cap,"GP Internet Pack","Lifebuoy Soap 100g","Power Bank 10000mAh"}	1
{"ACI Pure Salt 1kg","Bashundhara Tissue","Lux Soap 100g","Miniket Rice 5kg"}	1
{"ACI Pure Salt 1kg",Cap,"Robi Internet Pack","Wallet (Men)"}	1
{"Biscuits (Mixed)","Bru Coffee 200g","Onion 1kg","Potato 1kg"}	1

- **Repeated product combinations across orders indicate stable multi-item purchasing patterns, which can be leveraged for targeted bundling and promotional offers.**

23. Are there product pairs that consistently drive higher order values?



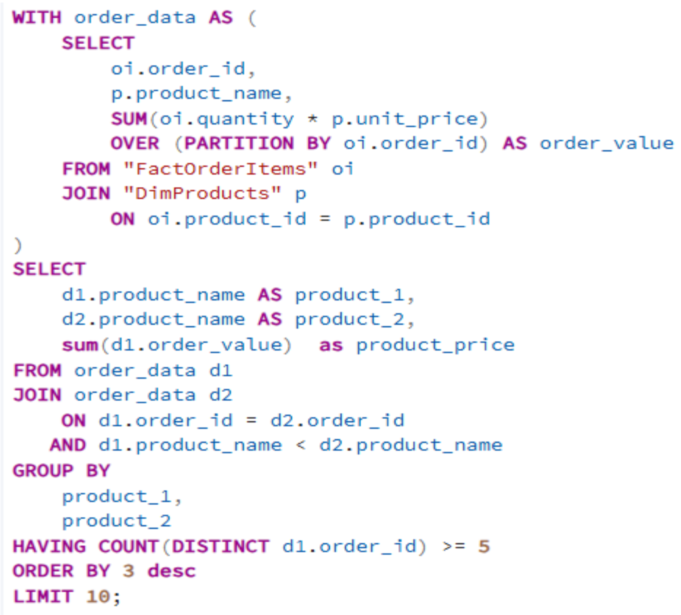
```
WITH order_data AS (
    SELECT
        oi.order_id,
        p.product_name,
        SUM(oi.quantity * p.unit_price)
        OVER (PARTITION BY oi.order_id) AS order_value
    FROM "FactOrderItems" oi
    JOIN "DimProducts" p
        ON oi.product_id = p.product_id)
SELECT
    d1.product_name AS product_1,
    d2.product_name AS product_2,
    ROUND(AVG(d1.order_value), 2) AS avg_order_value
FROM order_data d1
JOIN order_data d2
    ON d1.order_id = d2.order_id
   AND d1.product_name < d2.product_name
GROUP BY 1, 2
ORDER BY avg_order_value DESC
LIMIT 10;
```



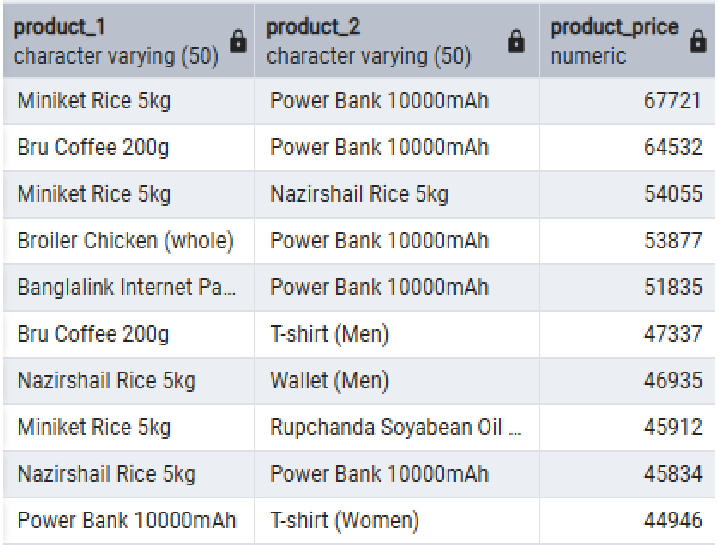
product_1 character varying (50)	product_2 character varying (50)	avg_order_value numeric
Ladies Bag	Power Bank 10000mAh	5433.00
Nazirshail Rice 5kg	Power Bank 10000mAh	5092.67
Power Bank 10000mAh	T-shirt (Women)	4494.60
Broiler Chicken (whole)	Power Bank 10000mAh	4489.75
Ladies Bag	Miniket Rice 5kg	4438.71
Power Bank 10000mAh	T-shirt (Men)	4391.29
Flour (Atta) 2kg	Power Bank 10000mAh	4181.00
Ladies Bag	Wallet (Men)	4173.00
Power Bank 10000mAh	Vim Dishwashing Bar	4156.33
Lux Soap 100g	Power Bank 10000mAh	4123.75

- **Certain product pairs, particularly those involving high-value items, are associated with higher average order values, indicating strong cross-selling potential.**

24. Which product combination could be recommended as bundles to increase revenue?



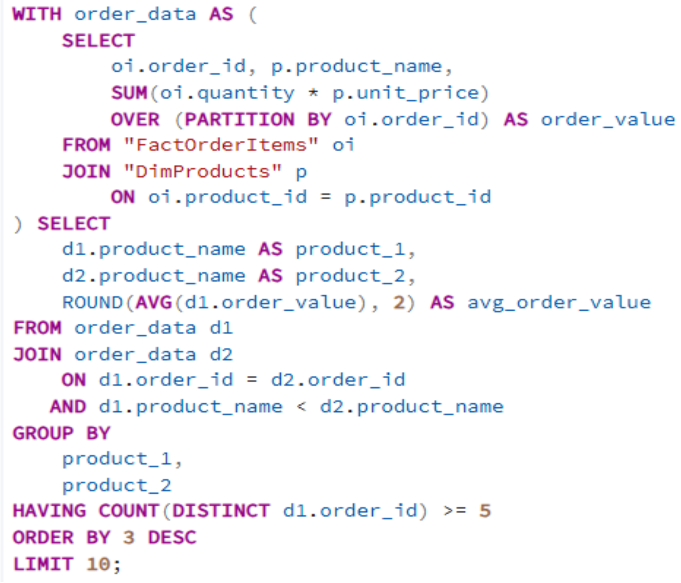
```sql
WITH order_data AS (
    SELECT
        oi.order_id,
        p.product_name,
        SUM(oi.quantity * p.unit_price)
        OVER (PARTITION BY oi.order_id) AS order_value
    FROM "FactOrderItems" oi
    JOIN "DimProducts" p
        ON oi.product_id = p.product_id
)
SELECT
    d1.product_name AS product_1,
    d2.product_name AS product_2,
    sum(d1.order_value)  as product_price
FROM order_data d1
JOIN order_data d2
    ON d1.order_id = d2.order_id
   AND d1.product_name < d2.product_name
GROUP BY
    product_1,
    product_2
HAVING COUNT(DISTINCT d1.order_id) >= 5
ORDER BY 3 desc
LIMIT 10;
```



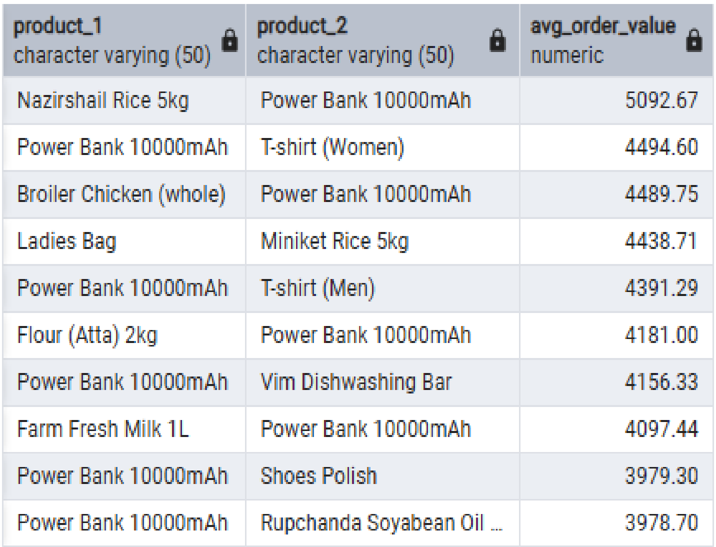
product_1 character varying (50) 🔒	product_2 character varying (50) 🔒	product_price numeric 🔒
Miniket Rice 5kg	Power Bank 10000mAh	67721
Bru Coffee 200g	Power Bank 10000mAh	64532
Miniket Rice 5kg	Nazirshail Rice 5kg	54055
Broiler Chicken (whole)	Power Bank 10000mAh	53877
Banglalink Internet Pa...	Power Bank 10000mAh	51835
Bru Coffee 200g	T-shirt (Men)	47337
Nazirshail Rice 5kg	Wallet (Men)	46935
Miniket Rice 5kg	Rupchanda Soyabean Oil ...	45912
Nazirshail Rice 5kg	Power Bank 10000mAh	45834
Power Bank 10000mAh	T-shirt (Women)	44946

- **Product combinations involving high-value and frequently co-purchased items are strong candidates for bundle offers to drive higher order values and overall revenue.**

25. Based on product co-occurrence and customer behavior, which products should UrbanCart promote together to maximize cross-selling opportunities?



```sql
WITH order_data AS (
    SELECT
        oi.order_id, p.product_name,
        SUM(oi.quantity * p.unit_price)
        OVER (PARTITION BY oi.order_id) AS order_value
    FROM "FactOrderItems" oi
    JOIN "DimProducts" p
        ON oi.product_id = p.product_id
) SELECT
    d1.product_name AS product_1,
    d2.product_name AS product_2,
    ROUND(AVG(d1.order_value), 2) AS avg_order_value
FROM order_data d1
JOIN order_data d2
    ON d1.order_id = d2.order_id
   AND d1.product_name < d2.product_name
GROUP BY
    product_1,
    product_2
HAVING COUNT(DISTINCT d1.order_id) >= 5
ORDER BY 3 DESC
LIMIT 10;
```



product_1 character varying (50) 🔒	product_2 character varying (50) 🔒	avg_order_value numeric 🔒
Nazirshail Rice 5kg	Power Bank 10000mAh	5092.67
Power Bank 10000mAh	T-shirt (Women)	4494.60
Broiler Chicken (whole)	Power Bank 10000mAh	4489.75
Ladies Bag	Miniket Rice 5kg	4438.71
Power Bank 10000mAh	T-shirt (Men)	4391.29
Flour (Atta) 2kg	Power Bank 10000mAh	4181.00
Power Bank 10000mAh	Vim Dishwashing Bar	4156.33
Farm Fresh Milk 1L	Power Bank 10000mAh	4097.44
Power Bank 10000mAh	Shoes Polish	3979.30
Power Bank 10000mAh	Rupchanda Soyabean Oil ...	3978.70

- **High-value product pairs that are frequently purchased together—especially combinations involving Power Bank 10000mAh—should be promoted jointly to maximize cross-selling opportunities**