SQL for Data Analysis

NAME: RAJ RAMJI JAISWAL

Objective: Analyze monthly revenue and order volume using SQL aggregation functions.

Tools: Oracle (Used Oracle Express Edition 11g, as it was available on my laptop).

Deliverables: SQL queries in a SQL file + screenshots of output file

Dataset: This SQL task was performed using Oracle Database.

- Database Name: Raj_task_6
- Tables Used: orders, products *(used for extended optional analysis)*

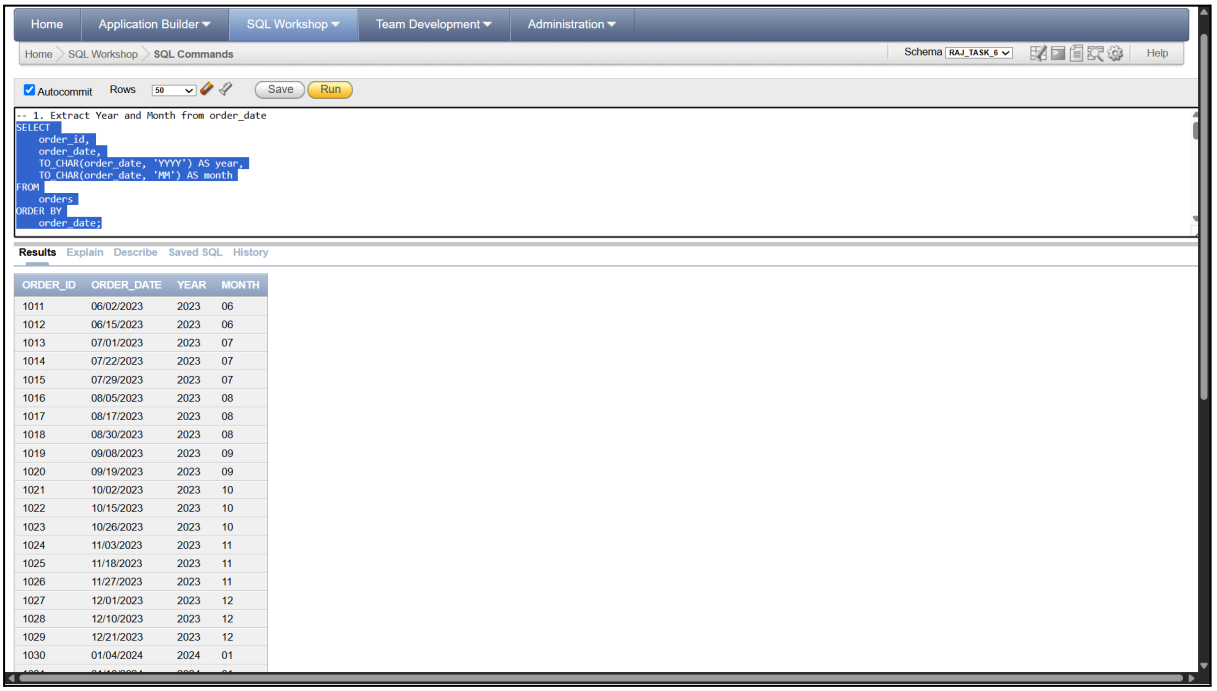
Task Summary:
In this task, I analyzed monthly sales trends using Oracle SQL. I used the `orders` table to extract and group data by year and month to calculate total revenue and order volume. SQL functions like `SUM()`, `COUNT()`, and `TO_CHAR()` were used along with `GROUP BY` and `ORDER BY` for aggregation and sorting.
Optional extended analysis was done using the `products` table to analyze category-wise and product-wise sales trends. This helped identify top-performing products and understand category-wise contributions to monthly revenue.

---------- Perform Task and Screenshots ----------

-- 1. Extract Year and Month from order_date

```
SELECT
    order_id,
    order_date,
    TO_CHAR(order_date, 'YYYY') AS year,
    TO_CHAR(order_date, 'MM') AS month
FROM
    orders
ORDER BY
    order_date;
```

Output:

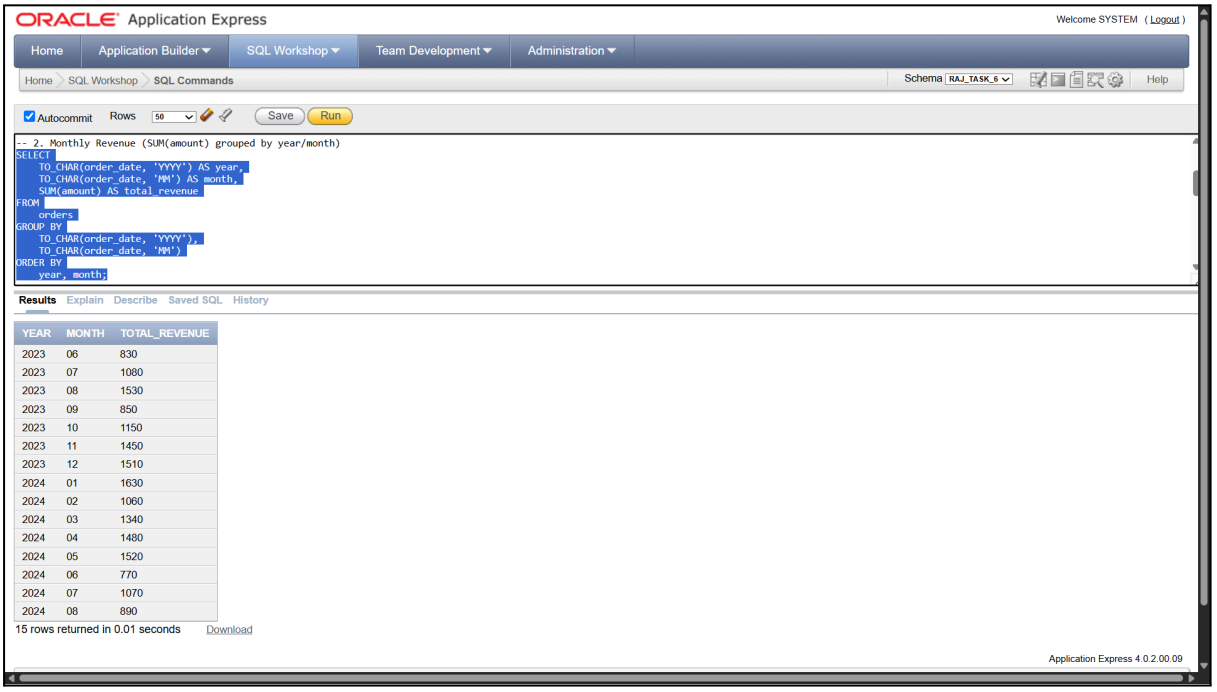


ORDER_ID	ORDER_DATE	YEAR	MONTH
1011	06/02/2023	2023	06
1012	06/15/2023	2023	06
1013	07/01/2023	2023	07
1014	07/22/2023	2023	07
1015	07/29/2023	2023	07
1016	08/05/2023	2023	08
1017	08/17/2023	2023	08
1018	08/30/2023	2023	08
1019	09/08/2023	2023	09
1020	09/19/2023	2023	09
1021	10/02/2023	2023	10
1022	10/15/2023	2023	10
1023	10/26/2023	2023	10
1024	11/03/2023	2023	11
1025	11/18/2023	2023	11
1026	11/27/2023	2023	11
1027	12/01/2023	2023	12
1028	12/10/2023	2023	12
1029	12/21/2023	2023	12
1030	01/04/2024	2024	01

-- 2. Monthly Revenue (SUM(amount) grouped by year/month)

```
SELECT
    TO_CHAR(order_date, 'YYYY') AS year,
    TO_CHAR(order_date, 'MM') AS month,
    SUM(amount) AS total_revenue
FROM
    orders
GROUP BY
    TO_CHAR(order_date, 'YYYY'),
    TO_CHAR(order_date, 'MM')
ORDER BY
    year, month;
```

Output:

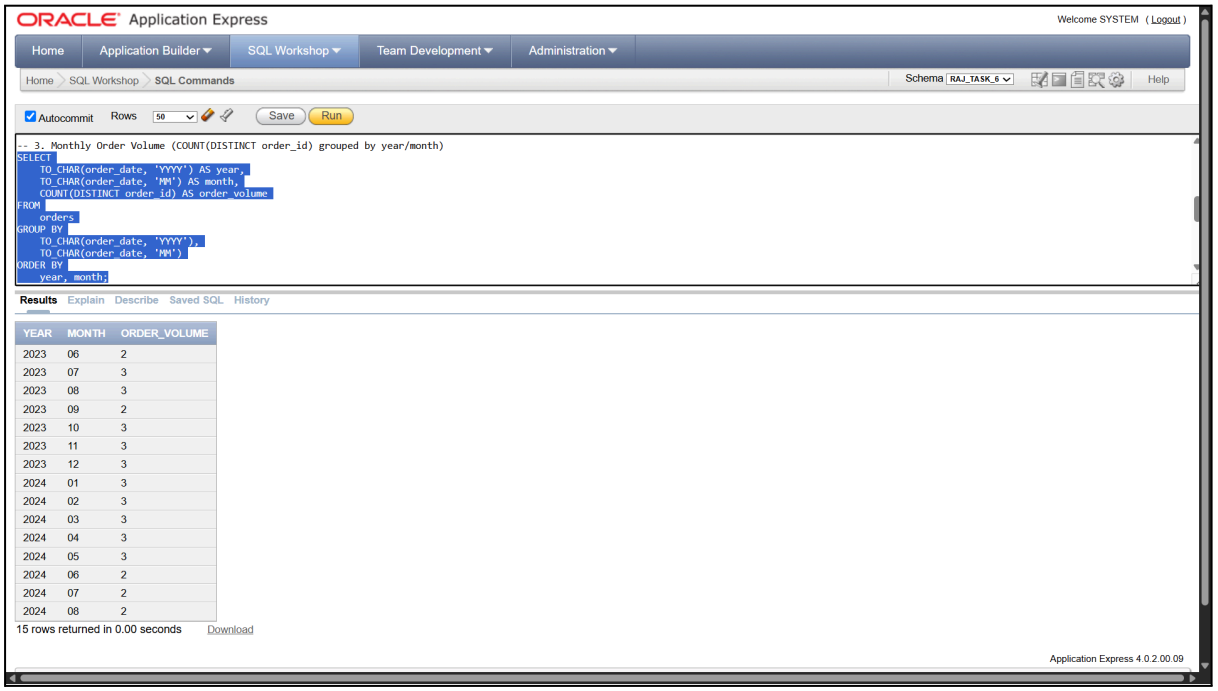


YEAR	MONTH	TOTAL_REVENUE
2023	06	830
2023	07	1080
2023	08	1530
2023	09	850
2023	10	1150
2023	11	1450
2023	12	1510
2024	01	1630
2024	02	1060
2024	03	1340
2024	04	1480
2024	05	1520
2024	06	770
2024	07	1070
2024	08	890

-- 3. Monthly Order Volume (COUNT(DISTINCT order_id) grouped by year/month)

```
SELECT
    TO_CHAR(order_date, 'YYYY') AS year,
    TO_CHAR(order_date, 'MM') AS month,
    COUNT(DISTINCT order_id) AS order_volume
FROM
    orders
GROUP BY
    TO_CHAR(order_date, 'YYYY'),
    TO_CHAR(order_date, 'MM')
ORDER BY
    year, month;
```

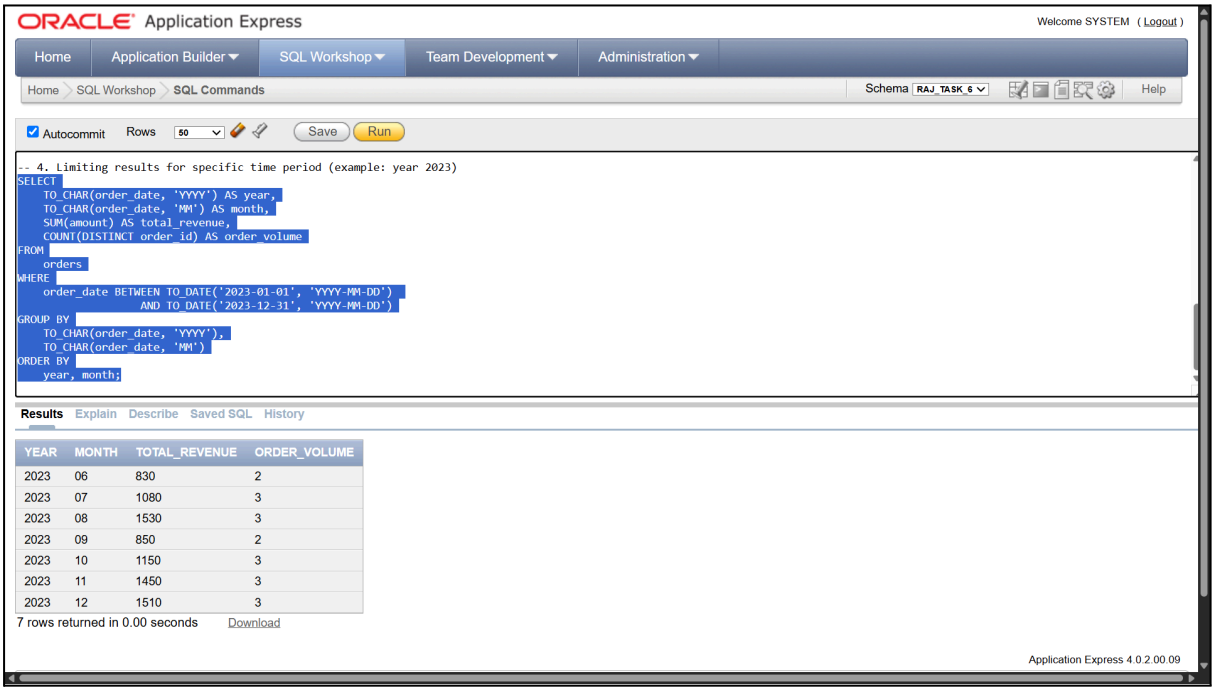
Output:



-- 4. Limiting results for specific time period (example: year 2023)

```
SELECT
    TO_CHAR(order_date, 'YYYY') AS year,
    TO_CHAR(order_date, 'MM') AS month,
    SUM(amount) AS total_revenue,
    COUNT(DISTINCT order_id) AS order_volume
FROM
    orders
WHERE
    order_date BETWEEN TO_DATE('2023-01-01',
'YYYY-MM-DD')
                   AND TO_DATE('2023-12-31',
'YYYY-MM-DD')
GROUP BY
    TO_CHAR(order_date, 'YYYY'),
    TO_CHAR(order_date, 'MM')
ORDER BY
    year, month;
```

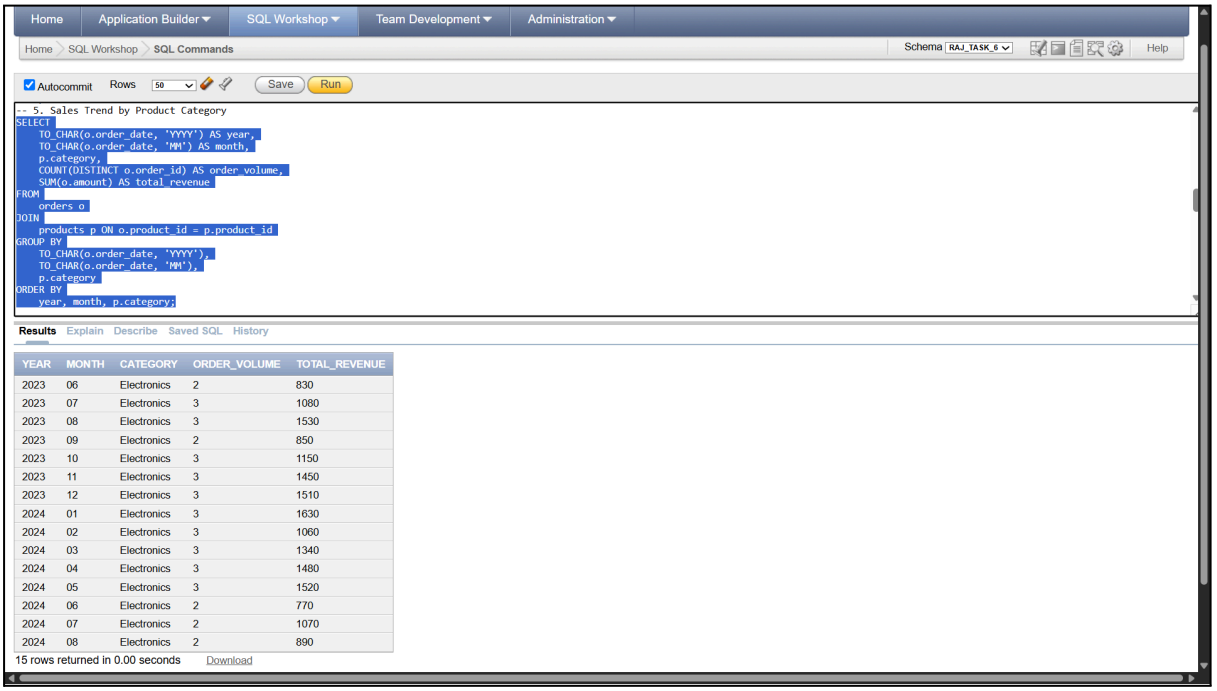
Output:



-- 5. Sales Trend by Product Category

```
SELECT
    TO_CHAR(o.order_date, 'YYYY') AS year,
    TO_CHAR(o.order_date, 'MM') AS month,
    p.category,
    COUNT(DISTINCT o.order_id) AS order_volume,
    SUM(o.amount) AS total_revenue
FROM
    orders o
JOIN
    products p ON o.product_id = p.product_id
GROUP BY
    TO_CHAR(o.order_date, 'YYYY'),
    TO_CHAR(o.order_date, 'MM'),
    p.category
ORDER BY
    year, month, p.category;
```

Output:

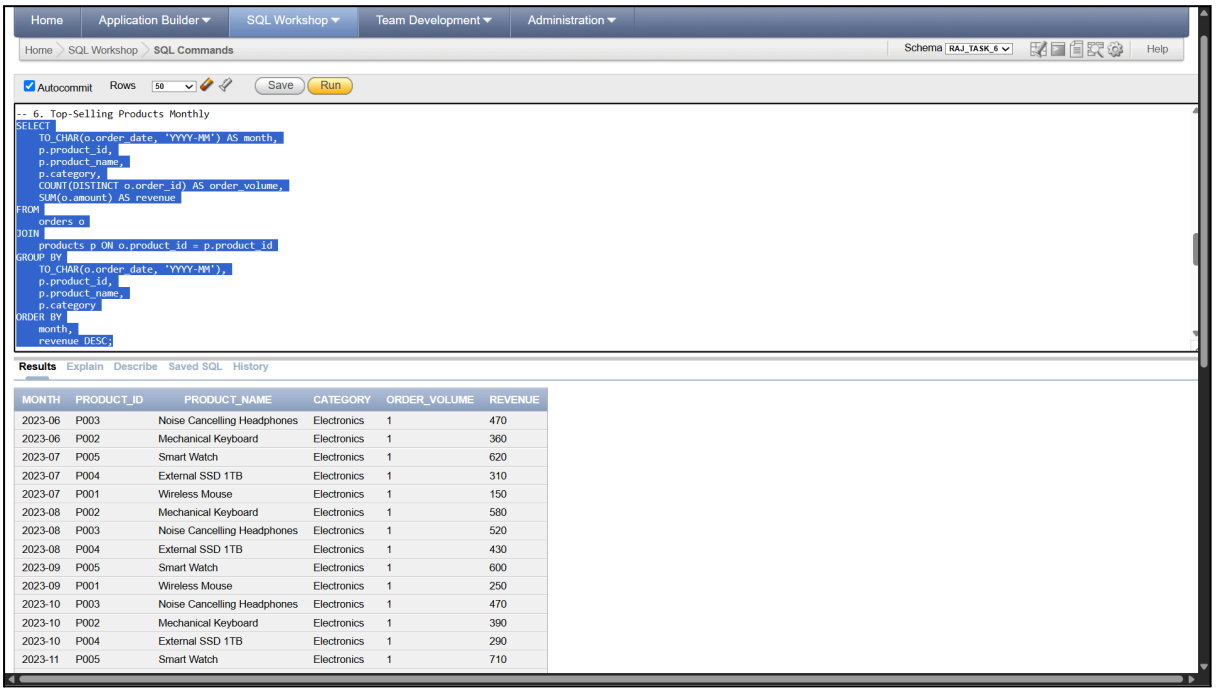


YEAR	MONTH	CATEGORY	ORDER_VOLUME	TOTAL_REVENUE
2023	06	Electronics	2	830
2023	07	Electronics	3	1080
2023	08	Electronics	3	1530
2023	09	Electronics	2	850
2023	10	Electronics	3	1150
2023	11	Electronics	3	1450
2023	12	Electronics	3	1510
2024	01	Electronics	3	1630
2024	02	Electronics	3	1060
2024	03	Electronics	3	1340
2024	04	Electronics	3	1480
2024	05	Electronics	3	1520
2024	06	Electronics	2	770
2024	07	Electronics	2	1070
2024	08	Electronics	2	890

-- 6. Top-Selling Products Monthly

```
SELECT
    TO_CHAR(o.order_date, 'YYYY-MM') AS month,
    p.product_id,
    p.product_name,
    p.category,
    COUNT(DISTINCT o.order_id) AS order_volume,
    SUM(o.amount) AS revenue
FROM
    orders o
JOIN
    products p ON o.product_id = p.product_id
GROUP BY
    TO_CHAR(o.order_date, 'YYYY-MM'),
    p.product_id,
    p.product_name,
    p.category
ORDER BY
    month,
    revenue DESC;
```

Output:

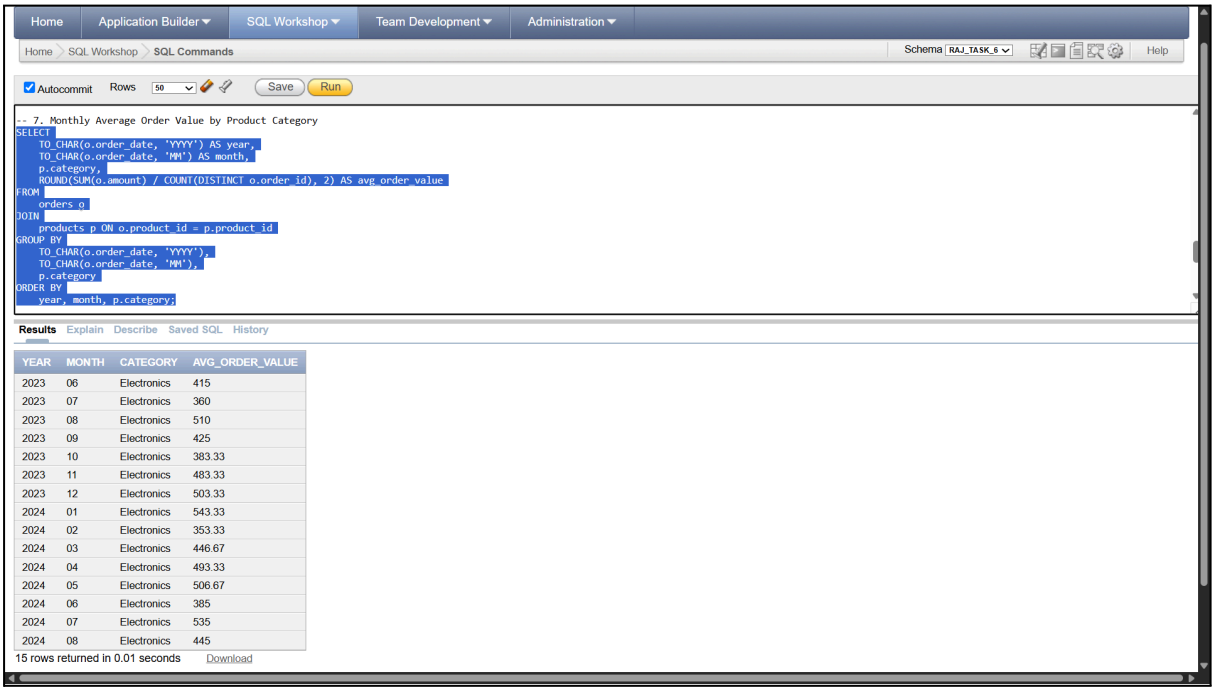


MONTH	PRODUCT_ID	PRODUCT_NAME	CATEGORY	ORDER_VOLUME	REVENUE
2023-06	P003	Noise Cancelling Headphones	Electronics	1	470
2023-06	P002	Mechanical Keyboard	Electronics	1	360
2023-07	P005	Smart Watch	Electronics	1	620
2023-07	P004	External SSD 1TB	Electronics	1	310
2023-07	P001	Wireless Mouse	Electronics	1	150
2023-08	P002	Mechanical Keyboard	Electronics	1	580
2023-08	P003	Noise Cancelling Headphones	Electronics	1	520
2023-08	P004	External SSD 1TB	Electronics	1	430
2023-09	P005	Smart Watch	Electronics	1	600
2023-09	P001	Wireless Mouse	Electronics	1	250
2023-10	P003	Noise Cancelling Headphones	Electronics	1	470
2023-10	P002	Mechanical Keyboard	Electronics	1	390
2023-10	P004	External SSD 1TB	Electronics	1	290
2023-11	P005	Smart Watch	Electronics	1	710

-- 7. Monthly Average Order Value by Product Category

```
SELECT
    TO_CHAR(o.order_date, 'YYYY') AS year,
    TO_CHAR(o.order_date, 'MM') AS month,
    p.category,
    ROUND(SUM(o.amount) / COUNT(DISTINCT o.order_id), 2)
AS avg_order_value
FROM
    orders o
JOIN
    products p ON o.product_id = p.product_id
GROUP BY
    TO_CHAR(o.order_date, 'YYYY'),
    TO_CHAR(o.order_date, 'MM'),
    p.category
ORDER BY
    year, month, p.category;
```

Output:



YEAR	MONTH	CATEGORY	AVG_ORDER_VALUE
2023	06	Electronics	415
2023	07	Electronics	360
2023	08	Electronics	510
2023	09	Electronics	425
2023	10	Electronics	383.33
2023	11	Electronics	483.33
2023	12	Electronics	503.33
2024	01	Electronics	543.33
2024	02	Electronics	353.33
2024	03	Electronics	446.67
2024	04	Electronics	493.33
2024	05	Electronics	506.67
2024	06	Electronics	385
2024	07	Electronics	535
2024	08	Electronics	445

-- 8. Total Revenue and Orders by Product

```
SELECT
    p.product_id,
    p.product_name,
    p.category,
    COUNT(DISTINCT o.order_id) AS total_orders,
    SUM(o.amount) AS total_revenue
FROM
    orders o
JOIN
    products p ON o.product_id = p.product_id
GROUP BY
    p.product_id,
    p.product_name,
    p.category
ORDER BY
    total_revenue DESC;
```

Output: