TAKE HOME ASSIGNMENT 4

1) Pull total number of orders that were completed on 18th March 2023:

SELECT COUNT(DISTINCT order_id) AS total_orders

FROM SALES

WHERE Date = '2023-03-18';

```
sales_db=# SELECT COUNT(DISTINCT order_id) AS total_orders
sales_db-# FROM SALES
sales_db-# WHERE Date = '2023-03-18';
 total_orders
--------------
            4
(1 row)

sales_db=#
```

2) Pull total number of orders that were completed on 18th March 2023 with the first name 'John' and last name Doe'

```
SELECT COUNT(DISTINCT s.order_id) AS total_orders
FROM SALES s
JOIN CUSTOMERS c ON s.customer_id = c.customer_id
WHERE s.Date = '2023-03-18'
AND c.first_name = 'John'
AND c.last_name = 'Doe';
```

```
sales_db=# SELECT COUNT(DISTINCT s.order_id) AS total_orders
sales_db-# FROM SALES s
sales_db-# JOIN CUSTOMERS c ON s.customer_id = c.customer_id
sales_db-# WHERE s.Date = '2023-03-18'
sales_db-# AND c.first_name = 'John'
sales_db-# AND c.last_name = 'Doe';
 total_orders
--------------
            2
(1 row)
```

3) Pull total number of customers that purchased in January 2023 and the average amount spend per customer

```
SELECT COUNT(DISTINCT customer_id) AS total_customers,
      AVG(total_spent) AS avg_spent_per_customer
FROM (
   SELECT s.customer_id, SUM(s.revenue) AS total_spent
   FROM SALES s
   WHERE s.Date BETWEEN '2023-01-01' AND '2023-01-31'
   GROUP BY s.customer_id
) customer_spending;
```

```
sales_db=# SELECT COUNT(DISTINCT customer_id) AS total_customers,
sales_db-#        AVG(total_spent) AS avg_spent_per_customer
sales_db-# FROM (
sales_db(#     SELECT s.customer_id, SUM(s.revenue) AS total_spent
sales_db(#     FROM SALES s
sales_db(#     WHERE s.Date BETWEEN '2023-01-01' AND '2023-01-31'
sales_db(#     GROUP BY s.customer_id
sales_db(# ) customer_spending;
 total_customers | avg_spent_per_customer
-----------------+------------------------
               3 |   450.0000000000000000
(1 row)

sales_db=#
```

4) Pull the departments that generated less than $600 in 2022

```
SELECT i.department, SUM(s.revenue) AS total_revenue
FROM SALES s
JOIN ITEMS i ON s.item_id = i.item_id
WHERE s.Date BETWEEN '2022-01-01' AND '2022-12-31'
GROUP BY i.department
HAVING SUM(s.revenue) < 600;
```

```
sales_db=# SELECT i.department, SUM(s.revenue) AS total_revenue
sales_db-# FROM SALES s
sales_db-# JOIN ITEMS i ON s.item_id = i.item_id
sales_db-# WHERE s.Date BETWEEN '2022-01-01' AND '2022-12-31'
sales_db-# GROUP BY i.department
sales_db-# HAVING SUM(s.revenue) < 600;
 department  | total_revenue
-------------+---------------
 Electronics |        200.00
 Furniture   |        400.00
(2 rows)
```

5) What is the most and least revenue we have generated by an order

```
SELECT MAX(order_revenue) AS max_revenue, MIN(order_revenue) AS min_revenue
FROM (
   SELECT order_id, SUM(revenue) AS order_revenue
   FROM SALES
   GROUP BY order_id
) order_totals;
```

```
sales_db=# SELECT MAX(order_revenue) AS max_revenue, MIN(order_revenue) AS min_revenue
sales_db-# FROM (
sales_db(#     SELECT order_id, SUM(revenue) AS order_revenue
sales_db(#     FROM SALES
sales_db(#     GROUP BY order_id
sales_db(# ) order_totals;
 max_revenue | min_revenue
-------------+-------------
     1200.00 |       50.00
(1 row)
```

6) What were the orders that were purchased in our most lucrative order

```
WITH order_totals AS (
   SELECT order_id, SUM(revenue) AS order_revenue
   FROM SALES
   GROUP BY order_id
)
SELECT s.*
FROM SALES s
WHERE s.order_id = (
   SELECT order_id FROM order_totals
   ORDER BY order_revenue DESC
   LIMIT 1
);
```

```
sales_db=# WITH order_totals AS (
sales_db(#     SELECT order_id, SUM(revenue) AS order_revenue
sales_db(#     FROM SALES
sales_db(#     GROUP BY order_id
sales_db(# )
sales_db-# SELECT s.*
sales_db-# FROM SALES s
sales_db-# WHERE s.order_id = (
sales_db(#     SELECT order_id FROM order_totals
sales_db(#     ORDER BY order_revenue DESC
sales_db(#     LIMIT 1
sales_db(# );
    date    | order_id | item_id | customer_id | quantity | revenue
------------+----------+---------+-------------+----------+---------
 2023-03-18 |     1004 |     101 |           4 |        1 | 1200.00
(1 row)
```