PIZZA SALES REPORT

WELCOME TO MY PIZZA SALES REPORT PROJECT

HELLO !

- MY NAME IS ABHISHEK PARETA AND IN THIS PROJECT, I HAVE UTILIZED SQL QUERIES TO SOLVE QUESTIONS THAT WERE RELATED TO PIZZA SALES.
- I'M THRILLED TO SHARE SOME EXCITING MILESTONES AS VENTURE INTO THIS FIELD.
- WITH A KEEN INTEREST IN EMBRACING THE DATA-DRIVEN WORLD.
- I'M PROUD TO PRESENT MY FORAY INTO THE REALM OF DATA ANALYTICS.
- THIS JOURNEY MARKS JUST THE BEGINNING OF MY EXPLORATION INTO THIS FASCINATING DOMAIN.

TABLES

- order_details
  - Columns
    - order_id
    - order_details_id
    - pizza_id
    - quantity

- pizza_types
  - Columns
    - pizza_type_id
    - name
    - category
    - ingredients

- orders
  - Columns
    - order_id
    - order_date
    - order_time

- pizzas
  - Columns
    - pizza_id
    - pizza_type_id
    - size
    - price

Q.1.Retrieve the total number of orders placed

Query

```
SELECT
    COUNT(order_id) AS Total_Orders
FROM
    orders;
```

Result

Total_Orders
21350

Q.2. Calculate the total revenue generated from pizza sales

Query

```
SELECT
    round(SUM(pizzas.price * order_details.quantity),2) AS total_revenue
FROM
    pizzas
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id;
```

Result

total_revenue
817860.05

Q.3. Identify the highest-priced pizza

Query

```sql
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

Result

name	price
The Greek Pizza	35.95

Q.4. Identify the most common pizza size ordered

Query

```
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS total_order
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY total_order DESC
LIMIT 1;
```

Result

size	total_order
L	18526

Q.5. List the top 5 most ordered pizza types along with their quantities.

Query

```
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;
```

Result

name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

Q.6. Join the necessary tables to find the total quantity of each pizza category ordered

Query

```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS total_quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.category
ORDER BY total_quantity DESC;
```

Result

category	total_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Q.7. Determine the distribution of orders by hour of the day

Query

```sql
-- 7. Determine the distribution of orders by hour of the day
SELECT
    HOUR(orders.order_time) AS hour,
    COUNT(orders.order_id) AS total_orders
FROM
    orders
GROUP BY hour;
```

Result

hour	total_orders
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

Q.8. Join relevant tables to find the category-wise distribution of pizzas

Query

```
-- 8. Join relevant tables to find the category-wise
-- distribution of pizzas.
SELECT
    COUNT(pizza_types.name) as name, pizza_types.category
FROM
    pizza_types
GROUP BY pizza_types.category;
```

Result

category	name
Chicken	6
Classic	8
Supreme	9
Veggie	9

Q.9. Group the orders by date and calculate the average number of pizzas ordered per day

Query

```sql
-- 9. Group the orders by date and calculate the average number of
-- pizzas ordered per day
SELECT
    ROUND(AVG(data.quantity), 0)
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        order_details
    JOIN orders ON order_details.order_id = orders.order_id
    GROUP BY orders.order_date) AS data;
```

Result

ROUND(AVG(data.quantity), 0)
138

Q.10. Determine the top 3 most ordered pizza types based on revenue

Query

```sql
-- 10. Determine the top 3 most ordered pizza types based on revenue
SELECT
    pizza_types.name,
    CEIL(SUM(pizzas.price * order_details.quantity)) AS revenue
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
        JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```

Result

name	revenue
The Thai Chicken Pizza	43435
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41410

Q.11. Determine the top 3 most ordered pizza types based on revenue for each pizza category

Query

```sql
-- 11. Determine the top 3 most ordered pizza types based on revenue for each pizza category
SELECT
    pizza_types.category,
    CEIL(SUM(pizzas.price * order_details.quantity)) AS revenue
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
        JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY pizza_types.category
ORDER BY revenue DESC
LIMIT 3;
```

Result

category	revenue
Classic	220054
Supreme	208197
Chicken	195920

CONCLUSION

As this is my first project so good little excited to work on this project. Based on the SQL analysis conducted on the pizza sales data, several key findings and insights have been revealed. These insights include identifying the best-selling pizza toppings, total revenue generated, peak sales hours, highest-priced pizza, popular pizza sizes and customer spending patterns. By doing so, stack-holders can make informed decisions and foster business growth within the competitive pizza industry.

Thank you