SQL QUERIES FOR PIZZA SALES

INTRODUCTRION

JAYESH SHRIVASTAVA

Hey everyone!
I'm JAYESH SHRIVASTAVA, and in this project, I explored a pizza sales dataset to see how SQL can be used to solve real-world business problems.
I used SQL to analyze sales data from different tables like order_detail, pizzas, and pizza_types, and uncovered key insights about revenue, popular categories, and overall sales performance.
This helped me understand how data can tell a story and guide smarter business moves.

ABOUT DATASET

The pizza sales dataset used in this project consists of four interconnected files: order_details, orders, pizzas, and pizza_types. The order_details file contains information about individual items in each order, including quantity and the specific pizza ordered. The orders file provides order-level details such as order ID and date. The pizzas file includes pricing and size information for each pizza, while pizza_types contains additional attributes like the name, category (e.g., classic, veggie, supreme), and ingredients of each pizza. Together, these files form a relational structure that allows for detailed analysis of sales, customer preferences, and product performance using SQL.

RETRIVE THE TOTAL NUMBER OF ORDERS PLACED

```sql
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

total_orders
21350

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

```sql
SELECT
    round(SUM(order_detail.quantity * pizzas.price) , 2) AS total_sales
FROM
    order_detail
        JOIN
    pizzas ON pizzas.pizza_id = order_detail.Pizza_id;
```

total_sales
817860.05

IDENTIFY THE HIGHEST PRICE PIZZA

```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;
```

name	price
The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDER

```sql
SELECT
    pizzas.size,
    COUNT(order_detail.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    order_detail ON pizzas.pizza_id = order_detail.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```

size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28

LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

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

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

TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED

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

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

DETERMINE THE DISTRIBUTION OF ORDERS BY HOURS OF THE DAY

```sql
SELECT
    HOUR(order_time), COUNT(order_ID)
FROM
    orders
GROUP BY HOUR(order_time);
```

hour(order_time)	count(order_ID)
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

JOIN RELEVANT TABLES TO FIND THE CATEGORY- WISE DISSTRIBUTION OF PIZZAS

```
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```

category	COUNT(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY

```sql
SELECT
    AVG(quantity)
FROM
    (SELECT
        orders.order_date, SUM(order_detail.quantity) AS quantity
    FROM
        orders
    JOIN order_detail ON orders.order_id = order_detail.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

avg(quantity)
138.4749

DETERMINE TOP3 MOST ORDERED PIZZA TYPE BASED ON REVENUE.

```sql
SELECT
    pizza_types.name,
    SUM(order_detail.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_detail ON order_detail.Pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

CALCULATE THE PERCENTAGE OF EACH PIZZA TYPE TO TOTAL REVENUE.

```sql
SELECT
    pizza_types.category,
    (SUM(order_detail.quantity * pizzas.price) / (SELECT
            ROUND(SUM(order_detail.quantity * pizzas.price),
                        2) AS total_sales
        FROM
            order_detail
                JOIN
            pizzas ON pizzas.pizza_id = order_detail.Pizza_id) )* 100 as revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_detail ON pizzas.pizza_id = order_detail.Pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC;
```

category	revenue
Classic	26.9059602556696 7
Supreme	25.4563112600986 2
Chicken	23.9551375684728 7
Veggie	23.6825909273845 77

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```sql
select order_date,
sum(revenue) over(order by order_date) as cum_revenue
from
(select orders.order_date,
sum(order_detail.quantity * pizzas.price) as revenue
from order_detail join pizzas
on order_detail.Pizza_id = pizzas.pizza_id
join orders
on orders.order_ID = order_detail.order_id
group by orders.order_date)  as sales;
```

order_date	cum_revenue
2015-12-24 00:00:00	807553.75
2015-12-26 00:00:00	809196.8
2015-12-27 00:00:00	810615.8
2015-12-28 00:00:00	812253
2015-12-29 00:00:00	813606.25
2015-12-30 00:00:00	814944.05
2015-12-31 00:00:00	817860.05

DETERMINE THE TOP 3 PIZZA TYPE ON THE BASE OF REVENUE IN EACH CATEGORIES

```sql
select name, revenue from
(select category, name, revenue,
rank() over(partition by category order by revenue desc) as rn
from
(select pizza_types.category, pizza_types.name,
sum((order_detail.quantity) * pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_detail
on order_detail.Pizza_id = pizzas.pizza_id
group by pizza_types.category, pizza_types.name) as a) as b
where rn<= 3;
```

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5
The Classic Deluxe Pizza	38180.5
The Hawaiian Pizza	32273.25
The Pepperoni Pizza	30161.75
The Spicy Italian Pizza	34831.25

THANK YOU!