Liceria & Co.

Pizza Sales Analysis using SQL (PizzaHut Data)

SQL PROJECT ON PIZZA SALES

PIZZA

A Delicious Journey

Analyzing Pizza Sales Data to Discover Business Insights

Prakash Kumar Jha

INTRODUCTION

- This project focuses on analyzing PizzaHut's sales data using SQL.
- The goal is to gain insights into order trends, pizza preferences, and revenue performance.
- The database consists of:
- orders
- order_details
- pizzas
- pizza_types

These tables are related through foreign keys for data integration and analysis.

Liceria & Co.

PROJECT OBJECTIVES

- Understand customer purchasing behavior.
- Analyze sales trends and performance.
- Identify the top-performing pizzas and categories.
- Generate revenue insights for decision-making.
- Practice and apply SQL skills for real-world business analysis.

Liceria & Co.

DATABASE STRUCTURE

Tables Used:

- orders → order_id, date, time
- order_details → order_details_id, order_id, pizza_id, quantity
- pizzas → pizza_id, pizza_type_id, size, price
- pizza_types → pizza_type_id, name, category, ingredients

Relationships:

- orders 🔗 order_details via order_id
- order_details 🔗 pizzas via pizza_id
- pizzas 🔗 pizza_types via pizza_type_id

Liceria & Co.

BASIC SQL ANALYSIS

Queries Performed:

1. Total number of orders placed

```
select count(order_id) as total_orders from orders;
```

total_orders
21350

2. Total revenue generated from pizza sales

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

total_sales
69635.8

BASIC SQL ANALYSIS

Queries Performed:

3. Highest-priced pizza

```
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

4. Most common pizza size ordered

```
select quantity, count(order_details_id)
from order_details group by quantity;

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

quantity	count(order_details_id)
1	4073
2	70
3	3

5. Top 5 most ordered pizza types with quantities

```
SELECT
    pizza_types.name,
    SUM(order_details.quantity) AS quantity
FROM
    pizza_types
JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.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;
```

name	quantity
The Pepperoni Pizza	237
The Barbecue Chicken Pizza	211
The California Chicken Pizza	202
The Thai Chicken Pizza	199
The Sicilian Pizza	191

INTERMEDIATE SQL ANALYSIS

Queries Performed:

1. Total quantity of pizzas ordered by category

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

category	quantity
Classic	1252
Supreme	1041
Veggie	1016
Chicken	913

2. Order distribution by hour of the day

```
SELECT HOUR(order_time) AS hour, count(order_id) as order_count from orders
group by hour (order_time);
```

hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468

3. Category-wise pizza distribution

```
select category, count(name) from pizza_types
group by category;
```

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

4. Average number of pizzas ordered per day

```
SELECT
    ROUND(AVG(quantity), 0) as avg_pizza_ordered_per_day
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

avg_pizza_ordered_per_day
136

5. Top 3 most ordered pizza types by revenue

```
select pizza_types.name,
sum(order_details.quantity * pizzas.price) as revenue
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 revenue desc limit 3;
```

name	revenue
The Barbecue Chicken Pizza	3770.25
The Thai Chicken Pizza	3657.25
The California Chicken Pizza	3471.5

ADVANCED SQL ANALYSIS

Queries Performed:

1. Percentage contribution of each pizza type to total revenue

```
select pizza_types.category,
(sum(order_details.quantity * pizzas.price) /
    (SELECT
        ROUND(SUM(order_details.quantity * pizzas.price),
    2) AS total_sales
    FROM
        order_details
    JOIN
        pizzas ON pizzas.pizza_id = order_details.pizza_id
    ) * 100)  as revenue
FROM
    pizza_types
JOIN
    pizzas
    ON pizza_types.pizza_type_id = pizzas.pizza_type_id
JOIN
    order_details
    ON order_details.pizza_id = pizzas.pizza_id
GROUP BY
    pizza_types.category
ORDER BY
    revenue desc
```

category	revenue
Classic	26.630913983899084
Supreme	25.68219220573325
Veggie	24.439153423957254
Chicken	23.247740386410438

2. Cumulative revenue generated over time

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

order_date	cum_revenue
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55

3. Top 3 most ordered pizza types by revenue per category

```
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_details.quantity) * pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.category, pizza_types.name) as a) as b
where rn <= 3;
```

name	revenue
The Barbecue Chicken Pizza	3770.25
The Thai Chicken Pizza	3657.25
The California Chicken Pizza	3471.5
The Pepperoni Pizza	2973.5
The Classic Deluxe Pizza	2941.5

THANK YOU