SQL Projects On

Pizza Sales

By Rahul Mahapatra

Introduction

Hi, I'm Rahul Mahapatra, and I am passionate about data analytics and business intelligence. I recently developed a SQL project focused on analyzing pizza sales data to address various business queries. This project involved designing and executing SQL queries to extract meaningful insights, optimize sales strategies, and improve overall operational efficiency.

Database Overview

The pizza sales database manages and analyzes pizza orders with the following key tables:

Table 1 -Order details

- order_details_id: Unique ID for order detail.
- order_id: ID of the order.
- pizza_id: ID of the pizza.
- quantity: Number of pizzas ordered.

Table 2 - Orders

- order_id: Unique ID for the order.
- date: Order date.
- time: Order time.

Table 3 - Pizza types

- pizza_type_id: Unique ID for the pizza type.
- name: Name of the pizza.
- category: Pizza category
- ingredients: Ingredients of the pizza.

Table 4 - Pizzas

- pizza_id: Unique ID for the pizza.
- pizza_type_id: ID linking to the pizza type.
- size: Size of the pizza
- price: Price of the pizza.

Questions

1. The total number of order place
2. The total revenue generated from pizza sales
3. The highest priced pizza.
4. The most common pizza size ordered.
5. The top 5 most ordered pizza types along their quantities.
6. The quantity of each pizza categories ordered.
7. The distribution of orders by hours of the day.
8. The category-wise distribution of pizzas.
9. The average number of pizzas ordered per day.
10. Top 3 most ordered pizza type base on revenue.
11. The percentage contribution of each pizza type to revenue.
12. The cumulative revenue generated over time.
13. The top 3 most ordered pizza type based on revenue for each pizza category.

Q1. The total number of order place

```
# Q1: The total number of order place

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

Result Grid

	total_orders
▶	21350

Q2. The total revenue generated from pizza sales

```
# The total revenue generated from pizza sales

select round(sum(order_details.quantity * pizzas.price),2) as
total_revenue from order_details
join pizzas
on order_details.pizza_id = pizzas.pizza_id ;
```

Result Grid

	total_revenue
▶	817860.05

Q3. The highest priced pizza.

```
# The 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;
```

Result Grid | Filter Ro

	name	price
▶	The Greek Pizza	35.95

Q4. The most common pizza size ordered.

```sql
# The most common pizza size ordered

select Count(order_details.order_id) as total_orders, pizzas.size
from order_details join pizzas
on order_details.pizza_id = pizzas.pizza_id
group by pizzas.size order by  total_orders desc
```

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

Q5. The top 5 most ordered pizza types along their quantities.

```sql
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 Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

Q6. The quantity of each pizza categories ordered.

```
use pizzasales;
select pizza_types.category ,
 sum(order_details.quantity) as Total_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 Total_Quantity desc
```

category	Total_Quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Q7. The distribution of orders by hours 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
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

Q8. The category-wise distribution of pizzas.

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

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

Q9. The average number of pizzas ordered per day.

```sql
# The average number of pizzas ordered per day

select round(avg (Quantity),0) as Avg_Order_Perday 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_Order_Perday
138

Q10. Top 3 most ordered pizza type base on revenue.

```sql
# Top 3 most ordered pizza type base on revenue.

select pizza_types.name , round(sum(order_details.quantity * pizzas.price),0)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.name order by revenue desc limit 3
```

name	revenue
The Thai Chicken Pizza	43434
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41410

Q11. The percentage contribution of each pizza type to revenue.

```sql
# The percentage contribution of each pizza type to revenue.
select pizza_types.category ,
round((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, 2) 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	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Q12. The cumulative revenue generated over time.

```sql
select order_date,
sum(revenue) over ( order by order_date ) as cumulative_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 order by revenue ) as sales;
```

order_date	cumulative_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
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004
2015-01-15	34343.50000000001
2015-01-16	36937.65000000001
2015-01-17	39001.75000000001

Q13. The top 3 most ordered pizza type based on revenue for each pizza category.

```sql
select name , revenue from
( select category , name ,revenue ,
rank() over ( partition by category order by revenue desc ) as rankk
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 rankk <=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
The Italian Supreme Pizza	33476.75
The Sicilian Pizza	30940.5
The Four Cheese Pizza	32265.70000
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.5

Thank You