



PIZZAS SALES SQL PROJECT

Create BY - Rohit Singh



Project Introduction

In this project, I developed a comprehensive analysis using SQL on four interconnected datasets: `order_details`, `orders`, `pizza_types`, and `pizzas`. By leveraging these datasets, I addressed and solved 13 distinct questions, providing valuable insights into various aspects of the pizza business. This involved extracting meaningful information such as sales trends, customer preferences, and inventory management. Through meticulous query design and optimization, the project highlights the power of SQL in handling complex data relationships and generating actionable business intelligence. This analysis serves as a foundation for data-driven decision-making in the context of a pizza enterprise.

-- 1. Retrieve the total number of orders placed.

```
SELECT *  
FROM orders;
```

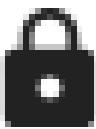
```
SELECT COUNT(order_id) AS total_orders  
FROM orders;
```

	total_orders	bigint	lock
1	21350		

-- 2.Calculate 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 pizzas.pizza_id = order_details.pizza_id;
```

total_revenue	numeric
1	827450.00



-- 3. Identify 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;
```

	name text	price integer
1	The Greek Pizza	36

-- 4.Identify the most common pizza size ordered.

```
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
LIMIT 1;
```

	size text	order_count bigint
1	L	18526

-- 5.List the top 5 most ordered pizza types along with their 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 text	quantity bigint
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418
5	The Thai Chicken Pizza	2371

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

```
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 text	quantity bigint
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

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

```
SELECT EXTRACT(HOUR  
              FROM order_time) AS order_hour,  
       COUNT(order_id) AS order_count  
  FROM orders  
 GROUP BY order_hour;
```

	order_hour numeric	order_count bigint
1	11	1231
2	23	28
3	18	2399
4	19	2009
5	15	1468

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

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

	category text	count bigint
1	Supreme	9
2	Chicken	6
3	Classic	8
4	Veggie	9

```
-- 9. Group the orders by date and calculate the average number of pizzas ordered per day.  
SELECT ROUND(AVG(quantity), 0) AS AVG_pizza_order_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_qantity;
```

avg_pizza_order_per_day	numeric
1	138



1

138

```
-- 10.Determine the top 3 most ordered pizza types based on 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 text	revenue bigint
1	The Thai Chicken Pizza	44027
2	The Barbecue Chicken Pizza	43376
3	The California Chicken Pizza	42002

```
-- 11.Calculate the 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)) AS total_revenue
FROM order_details
JOIN pizzas ON pizzas.pizza_id = order_details.pizza_id)* 100 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.category|
ORDER BY revenue DESC ;
```

	category text	revenue double precision
1	Classic	26.95727838540093
2	Supreme	25.50510604870385
3	Chicken	24.011360203033416
4	Veggie	23.526255362861804

```
-- 12. Analyze the 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 date	cum_revenue numeric
1	2015-01-01	2746
2	2015-01-02	5512
3	2015-01-03	8203
4	2015-01-04	9983
5	2015-01-05	12075
6	2015-01-06	14532
7	2015-01-07	16761
8	2015-01-08	19628
9	2015-01-09	21777
10	2015-01-10	24270

-- 13.Determine the top 3 most ordered pizza types based on revenue for each pizza 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 text	revenue bigint
1	The Thai Chicken Pizza	44027
2	The Barbecue Chicken Pizza	43376
3	The California Chicken Pizza	42002
4	The Classic Deluxe Pizza	38417
5	The Hawaiian Pizza	33122
6	The Pepperoni Pizza	30637
7	The Spicy Italian Pizza	35516
8	The Italian Supreme Pizza	34232
9	The Sicilian Pizza	30456
10	The Four Cheese Pizza	32478