



PIZZA SALES SQL PROJECT



HELLO !

This project focuses on analyzing a pizza sales dataset using SQL to uncover valuable business insights. The objective was to practice SQL querying skills while exploring how data-driven decisions can improve sales, customer preferences, and overall business performance.





OBJECTIVES OF THE PROJECT

- To analyze pizza sales data using SQL queries.
- To calculate revenue, order trends, and product performance.
- To identify the most popular pizzas, categories, and sizes.
- To uncover time-based ordering patterns (daily, monthly, hourly).
- To provide actionable insights for menu optimization and business growth.



DATASET INFORMATION

- Dataset Name: Pizza Sales Dataset (Wscube Tech)

Tables Included:

- orders → Order details with order date & time
- order_details → Quantity of pizzas in each order
- pizzas → Pizza ID, type, size, and price
- pizza_types → Pizza category and name

- Records: ~48,000 (approx)
- Duration: One full year of sales



RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
SELECT  
    COUNT(*)  
FROM  
    orders1;
```

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

SELECT

```
ROUND(SUM(od.quantity * p.price), 2) total_sales
```

FROM

```
order_details od
```

LEFT JOIN

```
pizzas p ON od.pizza_id = p.pizza_id;
```

IDENTIFY THE HIGHEST-PRICED PIZZA.

```
SELECT  
    name, MAX(price)  
FROM  
    pizzas p  
        LEFT JOIN  
    pizza_types pt ON p.pizza_type_id = pt.pizza_type_id  
GROUP BY name  
ORDER BY MAX(price) DESC  
LIMIT 1;
```

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
SELECT  
    COUNT(od.order_details_id) total_orders, p.size  
FROM  
    order_details od  
        LEFT JOIN  
    pizzas p ON od.pizza_id = p.pizza_id  
GROUP BY p.size  
ORDER BY COUNT(od.order_details_id) DESC  
LIMIT 1;
```

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

```
SELECT  
    pizza_types.name, SUM(order_details.quantity) 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;
```

JOIN THE NECESSARY TABLES TO
FIND THE TOTAL QUANTITY OF
EACH PIZZA CATEGORY ORDERED.

SELECT

pt.category, COUNT(order_details_id)

FROM

pizza_types pt

LEFT JOIN

pizzas p **ON** p.pizza_type_id = pt.pizza_type_id

LEFT JOIN

order_details od **ON** od.pizza_id = p.pizza_id

GROUP BY pt.category

ORDER BY COUNT(order_details_id) **DESC**;

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```
SELECT  
    HOUR(o.order_time), COUNT(o.order_id)  
FROM  
    orders1 o  
GROUP BY HOUR(order_time);
```

JOIN RELEVANT TABLES TO FIND
THE CATEGORY-WISE
DISTRIBUTION OF PIZZAS.

SELECT

category, COUNT(name)

FROM

pizza_types

GROUP BY category;

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

```
SELECT
    ROUND(AVG(quantity), 0) AS avg_pizza_ordered_per_day
FROM
    (SELECT
        orders1.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders1
    JOIN order_details ON orders1.order_id = order_details.order_id
    GROUP BY orders1.order_date) AS order_quantity;
```

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

```
SELECT pt.name, SUM(od.quantity * p.price) total_revenue
FROM pizza_types pt
    LEFT JOIN pizzas p ON pt.pizza_type_id = p.pizza_type_id
    LEFT JOIN order_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.name
ORDER BY total_revenue DESC
LIMIT 3;
```

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```
with pizza_revenue as
(
  select pizza_type_id , sum(od.quantity * p.price) revenue
  from pizzas p
  left join order_details od
  on od.pizza_id = p.pizza_id
  group by pizza_type_id
),
total as(
  select sum(revenue) total_revenue from pizza_revenue
)
select pr.pizza_type_id, round(pr.revenue,2),
round((pr.revenue/t.total_revenue)*100,2) percentage_contribution
from pizza_revenue pr, total t
order by pr.revenue desc;
```

ANALYZE THE CUMULATIVE REVENUE GENERATED MONTH ON MONTH.

```
select month(o.order_date) month_name,round(sum(od.quantity * p.price),2) total_revenue,  
round(sum(sum(od.quantity * p.price)) over(order by month(o.order_date))),2) MOM_Revenue  
from orders1 o  
left join order_details od  
on o.order_id = od.order_id  
left join pizzas p  
on p.pizza_id = od.pizza_id  
group by month(o.order_date);
```

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

```
select *  
from  
  (select pt.category,pt.name, sum(od.quantity * p.price) total_revenue,  
        rank() over(partition by pt.category order by sum(od.quantity * p.price) desc) ranking  
     from order_details od  
   left join pizzas p  
     on od.pizza_id = p.pizza_id  
   left join pizza_types pt  
     on pt.pizza_type_id = p.pizza_type_id  
   group by pt.name, pt.category  
   order by total_revenue desc) t  
where ranking <=3;
```



BY: HRITIK MISHRA

THANK YOU!

