

SQL PROJECT ON PIZZA SALES



INTRODUCTION



This project focuses on analyzing pizza sales data through SQL to derive meaningful business insights. The analysis is categorized into three levels: Basic, Intermediate, and Advanced. It covers a range of queries such as calculating total orders and revenue, identifying top-performing pizzas, analyzing customer preferences by size and time, and evaluating category-wise contributions to revenue. This project demonstrates the power of SQL in transforming raw data into actionable insights for data-driven decision-making in the food industry.

```
select * from orders
```

```
select count(order_id) as total_order
```



1 Retrieve the total number of orders placed.

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

2 Calculate the 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
```

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;
```

4

Identify the 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;
```

```
select * from orders
```

```
select count(order_id) as total_order
```

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;
```

```
select * from orders
```

```
select count(order_id) as total_order
```

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;
```

```
select * from orders
```

```
select count(order_id) as total_order
```

7

Determine the distribution of orders by hour of the day.

```
select hour(order_time) as hour, count(order_id) as order_count from orders  
group by hour(order_time);
```

8

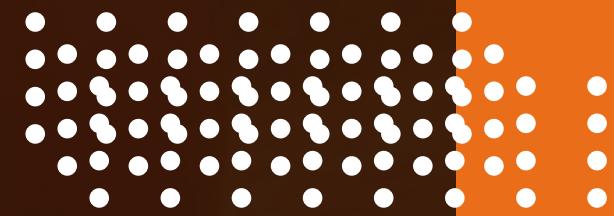
Join relevant tables to find the category-wise distribution of pizzas.

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

9

Group the orders by date and calculate the average number of pizzas ordered per day.

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



```
select * from orders
```

```
select count(order_id) as total_order
```

10 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),
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 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

LIMIT 3;

```
select * from orders
```

```
select count(order_id) as total_order
```

11 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 order_date) as sales;
```

```
select * from orders
```

```
select count(order_id) as total_order
```

12 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;
```