

SQL PROJECT ON PIZZA SALES





HELLO!!

THIS IS HARSHITH, IN THIS
PROJECT I HAVE UTILIZED
SQL QUERIES TO SOLVE
QUESTIONS RELATED TO
PIZZA SALES





I have covered 3 basic, 3 intermediate and 1 advance questions

1. Retrieve the total number of orders placed.

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

| | total_orders |
|---|--------------|
| ▶ | 21350 |

2. Calculate the total revenue generated from pizza sales

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

| | total_revenue |
|---|---------------|
| ▶ | 817860.05 |

3. Identify the highest-priced pizza sales

```
select pizza_types.name, pizzas.price as highest_priced  
from pizzas join pizza_types  
on pizzas.pizza_type_id = pizza_types.pizza_type_id  
order by pizzas.price desc limit 1;
```

| | name | highest_priced |
|---|-----------------|----------------|
| ▶ | The Greek Pizza | 35.95 |

4. Identify the most common pizza size ordered.

```
SELECT
    pizzas.size,
    COUNT(orders_details.order_details_id) AS order_count
FROM
    pizzas
    JOIN
        orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC
LIMIT 1;
```

| | size | order_count |
|---|------|-------------|
| ▶ | L | 18526 |

5. List the top 5 most ordered pizza types along with their quantities.

```
SELECT
```

```
    pizza_types.name, SUM(orders_details.quantity) AS quantity
```

```
FROM
```

```
    pizza_types
```

```
        JOIN
```

```
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
```

```
        JOIN
```

```
    orders_details ON orders_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 |

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

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

| | category | quantity |
|---|----------|----------|
| ▶ | Classic | 14888 |
| | Supreme | 11987 |
| | Veggie | 11649 |
| | Chicken | 11050 |

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);
```

| | 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 |

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

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

| | category | total |
|---|----------|-------|
| ▶ | Chicken | 6 |
| | Classic | 8 |
| | Supreme | 9 |
| | Veggie | 9 |

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

```
SELECT  
    ROUND(AVG(quantity), 2) AS avg_pizzas_ordered_per_day  
FROM  
(SELECT  
    orders.order_date, SUM(orders_details.quantity) AS quantity  
FROM  
    orders  
JOIN orders_details ON orders.order_id = orders_details.order_id  
GROUP BY order_date) AS order_quantity;
```

| | avg_pizzas_ordered_per_day |
|---|----------------------------|
| ▶ | 138.47 |

10. Determine the top 3 most ordered pizza types based on revenue.

```
SELECT  
    pizza_types.name,  
    SUM(orders_details.quantity * pizzas.price) AS revenue  
FROM  
    pizza_types  
        JOIN  
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
        JOIN  
    orders_details ON orders_details.pizza_id = pizzas.pizza_id  
GROUP BY pizza_types.name  
ORDER BY revenue DESC  
LIMIT 3;
```

| | name | revenue |
|---|------------------------------|----------|
| ▶ | The Thai Chicken Pizza | 43434.25 |
| | The Barbecue Chicken Pizza | 42768 |
| | The California Chicken Pizza | 41409.5 |

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(orders_details.quantity * pizzas.price) as revenue  
      from orders_details join pizzas  
        on orders_details.pizza_id = pizzas.pizza_id  
     join orders  
        on orders_details.order_id = orders.order_id  
   group by orders.order_date) as sales;
```

| | order_date | cum_revenue |
|---|------------|-------------------|
| ▶ | 2015-01-01 | 2713.850000000004 |
| | 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 | 23990.35000000002 |
| | 2015-01-11 | 25862.65 |
| | 2015-01-12 | 27781.7 |
| | 2015-01-13 | 29831.30000000003 |
| | 2015-01-14 | 32358.70000000004 |
| | 2015-01-15 | 34343.50000000001 |
| | 2015-01-16 | 36937.65000000001 |
| | 2015-01-17 | 39001.75000000001 |
| | 2015-01-18 | 40978.60000000006 |
| | 2015-01-19 | 43365.75000000001 |
| | 2015-01-20 | 45763.65000000001 |
| | 2015-01-21 | 47804.20000000001 |
| | 2015-01-22 | 50300.90000000001 |
| | 2015-01-23 | 52724.60000000006 |
| | 2015-01-24 | 55013.85000000006 |
| | 2015-01-25 | 56631.40000000001 |



THE END

THANK YOU!

