



About



Contact

SQL PROJECT ON PIZZA SALES





In this project, I utilized SQL queries to analyze and gain insights into pizza sales data. By leveraging the power of SQL, I was able to extract meaningful information from the data and answer various questions related to pizza sales.

HELLO!!

I'M VAIBHAVI DESHMUKH.





BRIEF OF THE PIZZAHUT About

Contact

DATABASE

The PizzaHut database contains key tables like:

Table Name	DESCRIPTION
Pizzas	pizza_id , Pizza_type_id , Size , Price
Pizza_types	Name , Pizza_type_id , Category , Ingredients
Orders	Order_Id , Date , Time , Date (Month)
Order_details	order_details_id , order_id , pizza_id , quantity





1. Identify the highest-priced pizza.

[About](#)[Contact](#)

```
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 Rows:

	name	price
▶	The Greek Pizza	35.95





Contact

2. Retrieve the total number of orders placed.

• **SELECT**

```
COUNT(order_id) AS total_orders
```

FROM

```
orders;
```

Result Grid	
	total_orders
▶	21350





Contact

3. 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;
```

Result Grid	
	total_sales
▶	817860.05



[About](#)[Contact](#)

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;
```

	size	order_count
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28





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

About

Contact

```
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

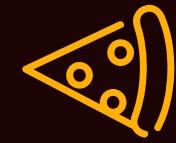


[About](#)[Contact](#)

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

```
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	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);
```

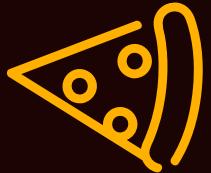
	hour	order_count
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009
	20	1642
	21	1198





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

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



category	pizza_count
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), 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_quantity;
```

	avg_pizza_order_per_day
	1.38



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

[About](#)[Contact](#)

```
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	Revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5



11. Calculate the percentage contribution of each pizza type to total revenue.

[About](#)[Contact](#)

```
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



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	cum_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



13. Determine the top 3 most ordered pizza types based on revenue for each pizza category.

[About](#)[Contact](#)

```
select name,revenue from
  (select category,name,revenue,
rank() over(partition by category order by revenue desc ) as rnk
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 rnk <=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	32265.7000000065
The Mexicana Pizza	26780.75



14. Monthly Sales Trend

[About](#)[Contact](#)

```
SELECT EXTRACT(MONTH FROM o.order_date) AS month,
       round( SUM(p.price * od.quantity), 0)AS total_sales
  FROM order_details od
 JOIN pizzas p ON od.pizza_id = p.pizza_id
 JOIN orders o ON od.order_id = o.order_id
 GROUP BY EXTRACT(MONTH FROM o.order_date)
 ORDER BY month;
```

month	total_sales
1	69793
2	65160
3	70397
4	68737
5	71403
6	68230
7	72558
8	68278
9	64180
10	64028
11	70395



15. Ordering Frequency by Order ID

[About](#)[Contact](#)

```
SELECT order_id,  
       COUNT(od.order_id) AS order_count,  
       SUM(p.price) AS total_spent  
  FROM Order_details od  JOIN pizzas p ON od.pizza_id=p.pizza_id  
 GROUP BY order_id  
 HAVING COUNT(order_id) > 5 -- filter customers with more than 5 orders  
 ORDER BY order_count DESC;
```

order_id	order_count	total_spent
10760	21	349.9
18845	21	338.7
13906	15	239.5
14304	15	256.95
17464	15	261
18280	15	240.75
20163	15	232.5
20710	15	244
21158	15	243.5
330	14	233.45
440	14	242.7

THANK YOU

• PIZZA SALES PRESENTATION