



SQL PROJECT

By

Sruthika

A Project on Pizza House Sales Analysis.



PROJECT SUMMARY

01

OBJECTIVE

Analyze pizza sales data

- provide actionable insights on sales performance
- customer preferences, and
- revenue generation

02

FEATURES OF THE TOPIC

- Sub Query
- Group by
- Cumulative sum
- Visualizations

03

ABOUT THE DATA SET

- Analyzing sales records from a pizza restaurant, spanning one year.
- Dataset has 48,620 sales transactions to analyze

04

TOOLS USED

- MYSQL for data querying and manipulation

PROBLEM STATEMENTS

1. Retrieve the total number of orders placed.
2. Calculate the total revenue generated from pizza sales.
3. Identify the highest-priced pizza.
4. Identify the most common pizza size ordered.
5. List the top 5 most ordered pizza types along with their quantities.
6. Join the necessary tables to find the total quantity of each pizza category ordered.
7. Determine the distribution of orders by hour of the day.
8. Join relevant tables to find the category-wise distribution of pizzas.
9. Group the orders by date and calculate the average number of pizzas ordered per day.
10. Determine the top 3 most ordered pizza types based on revenue.
11. Calculate the percentage contribution of each pizza type to total revenue.
12. Analyze the cumulative revenue generated over time.
13. Determine the top 3 most ordered pizza types based on revenue for each pizza category.

DATA SCHEMA

pizzas	
	Columns
◆	pizza_id
◆	pizza_type_id
◆	size
◆	price

pizza_types	
	Columns
◆	pizza_type_id
◆	name
◆	category
◆	ingredients

orders	
	Columns
◆	order_id
◆	date_date
◆	order_time

order_details	
	Columns
◆	order_details_id
◆	order_id
◆	pizza_id
◆	quantity

CREATING DATABASE AND TABLES

```
1 • create database pizzahut;
2 • ⊖ create table orders (
3   order_id int not null,
4   date_date date not null,
5   order_time time not null,
6   primary key (order_id));
7
8 • ⊖ create table order_details (
9   order_details_id int not null,
10  order_id int not null,
11  pizza_id text not null,
12  quantity int not null,
13  primary key (order_details_id));
```

1. Retrieve the total number of orders placed

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

Result Grid	
	total_orders
▶	21350

2. Calculate the total revenue generated from pizza sales

```
select  
    round(sum(order_details.quantity * pizzas.price),2) as total_sales_revenue  
from order_details join pizzas  
on pizzas.pizza_id = order_details.pizza_id;
```

total_sales_revenue
817860.05



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	price
The Greek Pizza	35.95

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.

```
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

6. find the total quantity of each pizza category ordered.

```
select pizza_types.category, sum(order_details.quantity) as quantity
from order_details join pizzas
on order_details.pizza_id=pizzas.pizza_id
join pizza_types
on pizza_types.pizza_type_id=pizzas.pizza_type_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 by order_count desc;
```

hour	order_count
12	2520
13	2455
18	2399
17	2336
19	2009
16	1920
20	1642
14	1472
15	1468
11	1231
21	1198
22	663
23	28
10	8
9	1

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

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

category	distr_Pizza
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_pz_order_prday  
from  
(select orders.date_date,sum(order_details.quantity) as quantity  
from orders join order_details  
on orders.order_id=order_details.order_id  
group by orders.date_date ) as order_quantity;
```

avg_pz_order_prday
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 pizzas join pizza_types
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.

```
> select pizza_types.category,round(sum(order_details.quantity*pizzas.price) /  
> (select sum(order_details.quantity*pizzas.price) as total_sales  
from order_details  
join pizzas  
on pizzas.pizza_id = order_details.pizza_id)*100 ,2) as revenue_prcnt  
  
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 category  
order by revenue_prcnt desc;
```

category	revenue_prcnt
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

12.Analyze the cumulative revenue generated over time.

```
select date_date,
       sum(revenue) over (order by date_date) as cum_revenue
  from
    (select orders.date_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.date_date) as sales;
```

date_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	23990.35000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.30000000003
2015-01-14	32358.70000000004
2015-01-15	34343.5000000001
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

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

```
select category, 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;
```

category	name	revenue
Chicken	The Thai Chicken Pizza	43434.25
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41409.5
Classic	The Classic Deluxe Pizza	38180.5
Classic	The Hawaiian Pizza	32273.25
Classic	The Pepperoni Pizza	30161.75
Supreme	The Spicy Italian Pizza	34831.25
Supreme	The Italian Supreme Pizza	33476.75
Supreme	The Sicilian Pizza	30940.5
Veggie	The Four Cheese Pizza	32265.70000000065
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5



THANK YOU!



SRUTHIKA BEKKUNUTI



Sruthika0727@gmail.com



www.linkedin.com/in/contact-sruthika