



PIZZAHUT
R E S T A U R A N T
SALES PROJECT
USING
SQL



ABOUT PROJECT

This SQL project is designed to analyze and retrieve various insights from a pizzahut database. The database consists of multiple tables, including orders, order_details, pizzas, and pizza_types. The project involves writing queries to extract and analyze data at different levels of complexity

BY-PRINCE SINGH



PIZZAHUT
R E S T A U R A N T

Retrieve the total number of orders placed.

```
use pizzahut;  
select count(*) as Total_Orders from orders;
```

Total_Orders
21350

@prince_singh



Calculate the total revenue generated from pizza sales

SELECT

```
ROUND(SUM(quantity * price), 2) AS total_revenue
```

FROM

```
order_details
```

JOIN

```
pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

@prince_singh

	total_revenue
▶	817860.05



Identify the highest-priced pizza.

```
SELECT  
    pizza_types.name, pizzas.price  
FROM  
    pizza_types  
        JOIN  
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id  
ORDER BY price DESC  
LIMIT 1;
```

@prince_singh

	name	price
▶	The Greek Pizza	35.95



Identify the most common pizza size ordered.

```
SELECT  
    COUNT(quantity) as total_count, size  
FROM  
    order_details  
    JOIN  
    pizzas ON pizzas.pizza_id = order_details.pizza_id  
GROUP BY size  
order by total_count desc;
```

@prince_singh

total_count	size
18526	L
15385	M
14137	S
544	XL
28	XXL



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

SELECT

```
    pizza_types.name AS name,  
    sum(order_details.quantity) AS total_count  
FROM  
    pizza_types  
        JOIN  
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
```

GROUP BY name

ORDER BY total_count DESC

LIMIT 5;

@prince_singh

name	total_count
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371



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

SELECT

```
    pizza_types.category AS category,  
    sum(order_details.quantity) AS total_quantity
```

@prince_singh

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 category

order by total_quantity desc



category	total_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Determine the distribution of orders by hour of the day

SELECT

```
HOUR((order_time)) AS hours,  
COUNT((order_id)) AS order_count
```

FROM

orders

GROUP BY hours

ORDER BY hours;

@prince_singh

hours	order_count
9	1
10	8
11	1231
12	2520
13	2455



find the category-wise distribution of pizzas.

SELECT

category, COUNT(name) as distribution

FROM

pizza_types

GROUP BY category

@prince_singh

category	distribution
Chicken	6
Classic	8
Supreme	9
Veggie	9



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

```
select round(avg(total_quantity),0) as avg_order_perday from  
(SELECT  
    orders.order_date, sum(order_details.quantity) as total_quantity  
FROM  
    order_details  
    JOIN  
    orders ON orders.order_id = order_details.order_id  
    group by order_date) as total_order;
```

@prince_singh

avg_order_perday
138



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

SELECT

```
pizza_types.name AS name,  
ROUND(SUM(pizzas.price * order_details.quantity),  
0) AS total_price
```

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 name

ORDER BY total_price DESC

LIMIT 3;

@prince_singh

name	total_price
The Thai Chicken Pizza	43434
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41410



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

SELECT

```
    pizza_types.category AS category,  
    ROUND((SUM(pizzas.price * order_details.quantity) / (SELECT  
        SUM(pizzas.price * order_details.quantity)  
    FROM  
        pizzas  
        JOIN  
        order_details ON order_details.pizza_id = pizzas.pizza_id)) * 100,  
    2) AS percentage
```

FROM

```
    pizza_types  
        JOIN  
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id  
        JOIN  
    order_details ON pizzas.pizza_id = order_details.pizza_id
```

GROUP BY category

@prince_singh

category	percentage
Classic	26.91
Veggie	23.68
Supreme	25.46
Chicken	23.96

Analyze the cumulative revenue generated over time.

```
select order_date, round(sum(revenue)
over(order by order_date),1 ) as cmu_revenue from
(SELECT
orders.order_date,
SUM(pizzas.price * order_details.quantity) AS revenue
FROM
pizzas
JOIN
order_details ON order_details.pizza_id = pizzas.pizza_id
JOIN
orders ON orders.order_id = order_details.order_id
GROUP BY order_date) as sales
```

@prince_singh

order_date	cmu_revenue
2015-01-01	2713.9
2015-01-02	5445.8
2015-01-03	8108.2
2015-01-04	9863.6
2015-01-05	11929.6

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

```
select name, category, revenue from
(select category,name,revenue,
rank() over(partition by category order by revenue desc) as rannk from
(SELECT
pizza_types.name AS name,
pizza_types.category as category,
SUM(pizzas.price * order_details.quantity) AS revenue
FROM
pizza_types
join pizzas on pizzas.pizza_type_id=pizza_types.pizza_type_id
join order_details on pizzas.pizza_id= order_details.pizza_id
group by name,category
) as a) as b
where rannk <=3;
```

@prince_singh

name	category	revenue
The Thai Chicken Pizza	Chicken	43434.25
The Barbecue Chicken Pizza	Chicken	42768
The California Chicken Pizza	Chicken	41409.5
The Classic Deluxe Pizza	Classic	38180.5
The Hawaiian Pizza	Classic	32273.25





PIZZAHUT
R E S T A U R A N T

THANK YOU

BY- PRINCE SINGH

