



SQL Project
PIZZA SALES INSIGHTS

An SQL Analysis of
PIZZA SALES

PROJECT



ROHAN SINGH

DATA ANALYTICS ENTHUSIAST

ABOUT ME

I AM AN ASPIRING DATA ANALYST SKILLED IN POWER BI, EXCEL, AND SQL, WITH A STRONG FOCUS ON TRANSFORMING DATA INTO ACTIONABLE INSIGHTS. ALTHOUGH I AM A FRESHER IN THE WORKFORCE, I HAVE COMPLETED SEVERAL IMPACTFUL PROJECTS, INCLUDING IMPROVING SALES FORECASTING AND ENHANCING DATA EFFICIENCY.

CURRENTLY, I AM PURSUING A BACHELOR OF TECHNOLOGY IN ELECTRICAL ENGINEERING AT JAIPUR ENGINEERING COLLEGE (2022-2025).

SKILLS:

- MS EXCEL: DATA ANALYSIS, PIVOT TABLES, ADVANCED FORMULAS
- POWER BI: CREATING INTERACTIVE DASHBOARDS AND VISUALIZATIONS
- SQL: DATABASE QUERYING AND OPTIMIZATION
- PROBLEM SOLVING: ANALYZING DATA FOR ACTIONABLE INSIGHTS
- COMMUNICATION: EFFECTIVELY PRESENTING COMPLEX DATA TO NON-TECHNICAL AUDIENCES

I AM EAGER TO LEVERAGE MY SKILLS AND PROJECT EXPERIENCE TO SUPPORT BUSINESS GROWTH AND DRIVE STRATEGIC DECISIONS.

Pizza Sales Data Schema



Retrieve the total number of orders placed.

CODE

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

RESULT

Result Grid	
	total_orders
▶	21350

Calculate the total revenue generated from pizza sales

CODE

```
SELECT  
    ROUND(SUM(od.quantity * pz.price), 2) AS revenue  
FROM  
    order_detail AS od  
        JOIN  
    pizzas AS pz ON od.pizza_id = pz.pizza_id;
```

RESULT

Result Grid	
	revenue
▶	817860.05

Identify the highest-priced pizza

CODE

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

RESULT

	name	price
▶	The Greek Pizza	35.95

Identify the most common pizza size ordered

CODE

```
SELECT  
    pz.size AS pizza_size , COUNT(od.order_id) AS total_order  
FROM  
    pizzas AS pz  
        JOIN  
    order_detail AS od ON pz.pizza_id = od.pizza_id  
GROUP BY pz.size  
ORDER BY total_order DESC  
LIMIT 1;
```

RESULT

Result Grid		Filter Row
	pizza_size	total_order
L		18526

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

CODE

```
SELECT
    pz.pizza_type_id AS pizza_type,
    COUNT(od.order_id) AS total_order,
    SUM(od.quantity) AS total_quantity
FROM
    pizzas AS pz
        JOIN
    order_detail AS od ON pz.pizza_id = od.pizza_id
GROUP BY pz.pizza_type_id
ORDER BY total_order DESC
LIMIT 5;
```

RESULT

pizza_type	total_order	total_quantity
classic_dlx	2416	2453
bbq_ckn	2372	2432
hawaiian	2370	2422
pepperoni	2369	2418
thai_ckn	2315	2371

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

CODE

```
SELECT  
    pz_ty.category, SUM(od.quantity) AS total_quantity  
FROM  
    pizza_types AS pz_ty  
        JOIN  
    pizzas AS pz ON pz_ty.pizza_type_id = pz.pizza_type_id  
        JOIN  
    order_detail AS od ON od.pizza_id = pz.pizza_id  
GROUP BY pz_ty.category;
```

RESULT

category	total_quantity
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

Determine the distribution of orders by hour of the day

CODE

```
SELECT  
    HOUR(time) AS "hour" , COUNT(order_id) AS "orders"  
FROM  
    orders  
GROUP BY HOUR(time)  
ORDER BY HOUR(time);
```

RESULT

	hour	orders
▶	9	1
	10	8
	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

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

CODE

```
SELECT  
    category, COUNT(pizza_type_id) as num_of_pizza  
FROM  
    pizza_types  
GROUP BY category  
order by num_of_pizza desc;
```

RESULT

category	num_of_pizza
Supreme	9
Veggie	9
Classic	8
Chicken	6

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

CODE

```
SELECT  
    ROUND(AVG(quantity)) AS avg_order_per_day  
FROM  
    (SELECT  
        od.date, SUM(odd.quantity) AS quantity  
    FROM  
        orders AS od  
    JOIN order_detail AS odd ON od.order_id = odd.order_id  
    GROUP BY od.date) AS avgquantity;
```

RESULT

avg_order_per_day
138

Determine the top 3 most ordered pizza types based on revenue

CODE

```
SELECT  
    pz_ty.name, round(SUM(od.quantity * pzz.price),2) AS revenue  
FROM  
    pizza_types AS pz_ty  
        JOIN  
    pizzas AS pzz ON pzz.pizza_type_id = pz_ty.pizza_type_id  
        JOIN  
    order_detail AS od ON od.pizza_id = pzz.pizza_id  
GROUP BY pz_ty.name  
ORDER BY revenue DESC  
LIMIT 3;
```

RESULT

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

Calculate the percentage contribution of each pizza type to total revenue

CODE

```
SELECT
    pz_ty.category,
    concat(round(((SUM(od.quantity * pzz.price)) / (SELECT
        ROUND(SUM(od.quantity * pz.price), 2) AS total_price
    FROM
        order_detail AS od
        JOIN
            pizzas AS pz ON od.pizza_id = pz.pizza_id)*100),2)," ", "%") as 'revenue in %'
FROM
    pizza_types AS pz_ty
    JOIN
        pizzas AS pzz ON pzz.pizza_type_id = pz_ty.pizza_type_id
    JOIN
        order_detail AS od ON od.pizza_id = pzz.pizza_id
GROUP BY pz_ty.category
order by 'revenue in %' desc;
```

RESULT

category	revenue in %
Classic	26.91 %
Veggie	23.68 %
Supreme	25.46 %
Chicken	23.96 %

Analyze the cumulative revenue generated over time

CODE

```
select date,  
round(sum(revenue) over ( order by date),2) as cumulative_revenue  
from  
(select orders.date, round (sum(order_detail.quantity*pizzas.price),2) as revenue  
from order_detail  
join pizzas  
on order_detail.pizza_id=pizzas.pizza_id  
join orders  
on orders.order_id=order_detail.order_id  
group by orders.date) as day_revenue;
```

RESULT

Result Grid		Filter Rows:
	date	cumulative_revenue
▶	2015-01-01	2713.85
	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	23990.35
	2015-01-11	25862.65
	2015-01-12	27781.7
	2015-01-13	29831.3
	2015-01-14	32358.7
	2015-01-15	34343.5
	2015-01-16	36937.65
	2015-01-17	39001.75
	2015-01-18	40978.6
	2015-01-19	43365.75
	2015-01-20	45763.65
	2015-01-21	47804.2

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

CODE

```
select category, name, revenue
from
  (select category, name, revenue,
  rank() over (partition by category order by revenue desc ) as ranking
  from
    (SELECT
      pz_ty.category, pz_ty.name,
      round(SUM(od.quantity * pzz.price),2) as revenue
    FROM
      pizza_types AS pz_ty
      JOIN
      pizzas AS pzz ON pzz.pizza_type_id = pz_ty.pizza_type_id
      JOIN
      order_detail AS od ON od.pizza_id = pzz.pizza_id
    GROUP BY pz_ty.category, pz_ty.name) as a) as b
  where ranking <=3;
```

RESULT

	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	32265.7
	Veggie	The Mexicana Pizza	26780.75
	Veggie	The Five Cheese Pizza	26066.5



SQL Project

PIZZA SALES INSIGHTS

THANK YOU



[WWW.LINKEDIN.COM/IN/ROHAN-SINGH-EE](https://www.linkedin.com/in/rohan-singh-ee)



ROHANSINGHEE03@GMAIL.COM



+91-7877443611

