

SQL PROJECT

PIZZA SALES

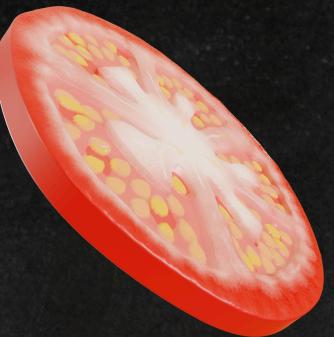
Life is short, eat the pizza!

LET'S BEGIN

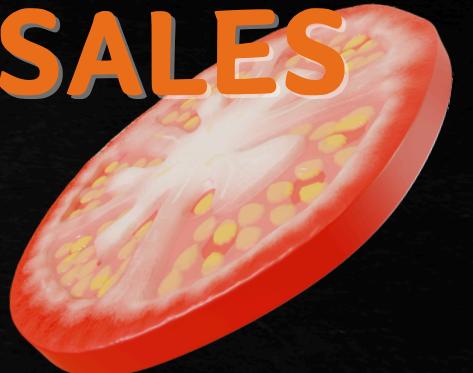
SQL developer
ROHIT TRIPATHY



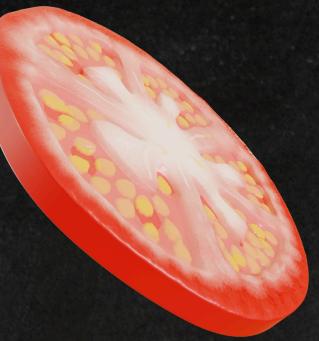
INTRO



INTRODUCING MYSELF AS ROHIT TRIPATHI, A CERTIFIED DATA SCIENTIST WITH A PASSION FOR LEVERAGING SQL TO DISSECT PIZZA SALES DATA. FROM BEGINNER TO ADVANCED QUERIES, I NAVIGATES THROUGH THE INTRICATE LAYERS OF INFORMATION, UNRAVELING INSIGHTS THAT DRIVE DECISION-MAKING IN THE DYNAMIC PIZZA INDUSTRY. WITH A KEEN EYE FOR DETAIL AND A KNACK FOR UNRAVELING COMPLEX DATASETS, I EXPERTISE ILLUMINATES TRENDS, PREFERENCES, AND OPTIMIZATIONS, SHAPING THE FUTURE OF PIZZA SALES STRATEGIES



QUESTIONS RELATED TO DATABASE



BASIC:

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

IDENTIFY THE HIGHEST-PRICED PIZZA.

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

INTERMEDIATE:

JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

SELECT RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

ADVANCED:

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.



1. RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
SELECT  
    COUNT(Order_ID) AS Total_Order  
FROM  
    orders
```

Result Grid	
	Total_Order
▶	21350

2.CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
SELECT  
    ROUND(SUM(O.Quantity * P.price), 2) AS Total_Revenue  
FROM  
    orders_details AS O  
    JOIN  
    pizzas AS P ON O.Pizza_ID = P.pizza_id
```

Result Grid	
	Total_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 Price DESC  
LIMIT 1
```

Result Grid | Filter Row

	name	price
▶	The Greek Pizza	35.95

4. IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
SELECT
    pizzas.size,
    COUNT(orders_details.Order_Detail_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
```

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(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

	category	Quantity
>	Classic	14888
	Veggie	11649
	Supreme	11987
	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
```

	Hour	order_count
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920

8.SELECT RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

```
Select category , count(name) AS Name  
from pizza_types  
group by category
```

	category	Name
▶	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 no. Of Pizza / Day'  
FROM  
(SELECT  
    orders.Order_Date AS Date,  
    SUM(orders_details.Quantity) AS Quantity  
FROM  
    orders  
JOIN orders_details ON orders.Order_ID = orders_details.Order_ID  
GROUP BY Date) AS Order_Quantity
```

	Avg no. Of Pizza / Day
▶	138

10.DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

SELECT

```
pizza_types.name AS 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 Name

ORDER BY revenue DESC

LIMIT 3

Result Grid		Filter Rows:
	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
    Category,
    Revenue,
    ROUND((Revenue / total_revenue) * 100, 2) AS PercentageContribution
FROM
    (SELECT
        pizza_types.category AS Category,
        ROUND(SUM(orders_details.Quantity * pizzas.price), 2) 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 category) AS category_revenue
    JOIN
    (SELECT
        SUM(orders_details.Quantity * pizzas.price) AS total_revenue
    FROM
        pizzas
    JOIN orders_details ON orders_details.Pizza_ID = pizzas.pizza_id) AS total ON 1 =
    ORDER BY Revenue DESC;
```

	Category	Revenue	PercentageContribution
▶	Classic	220053.1	26.91
	Supreme	208197	25.46
	Chicken	195919.5	23.96
	Veggie	193690.45	23.68

12. ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
select order_date,  
Round(sum(Revenue) over (order by order_date),2) as Cumm_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.order_id = orders_details.order_id  
group by orders.Order_Date) As sales
```

	order_date	Cumm_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

13.DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

```
select 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((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.category, pizza_types.name) As A) as B
where rn <= 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