

Pizza Sales Project

Hello, my name is Amrita. The primary goal of this SQL project is to identify the best-performing pizza types and understand their contribution to total revenue, offering insights into consumer preferences and trends. The insights generated can help in optimizing inventory, improving marketing strategies, and guiding future menu decisions for pizza restaurants.





Q.Retrieve the total number of orders

```
SELECT COUNT(order_id) AS Total_Orders  
FROM dbo.orders;
```

133 %

Results Messages

	Total_Orders
1	21350



Q. Calculate the total revenue generated from pizza sales



```
SELECT round(sum(pizzas.price*order_details.quantity),2) AS Total_Sales
FROM order_details
INNER JOIN pizzas ON pizzas.pizza_id = order_details.pizza_id;

ALTER TABLE order_details
ALTER COLUMN pizza_id VARCHAR(MAX);

ALTER TABLE pizzas
ALTER COLUMN pizza_id VARCHAR(MAX);
```

133 %	
Results	Messages
	Total_Sales
1	817860.05



Q. Identify the highest priced pizza

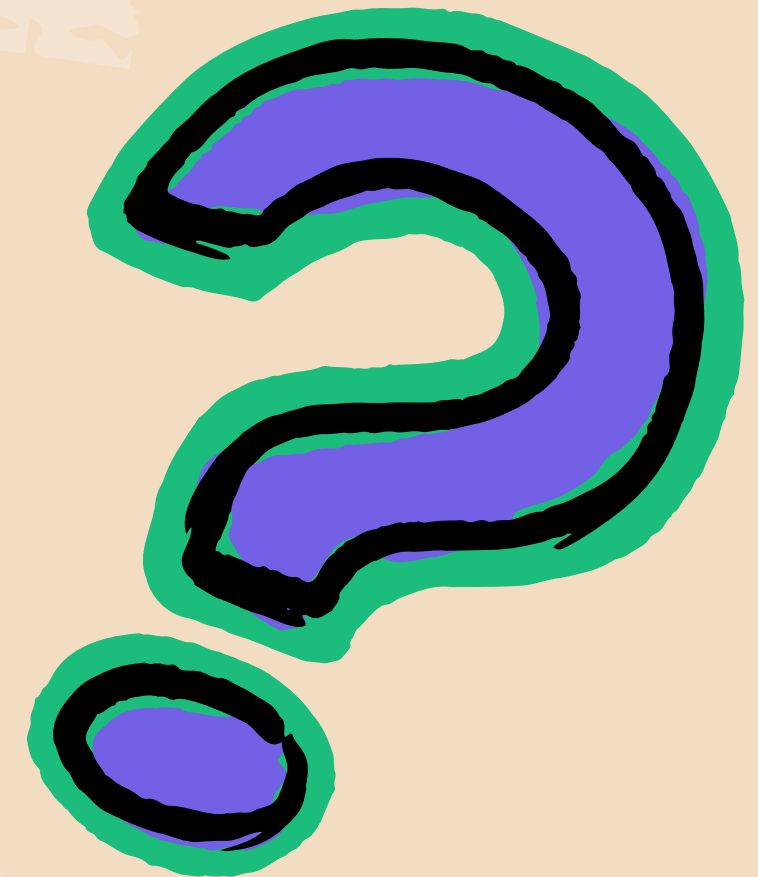
```
SELECT TOP 1 pizza_types.name, round(pizzas.price, 2) AS highest_priced_pizza  
FROM Pizzas INNER JOIN pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id  
ORDER BY price DESC;
```

```
ALTER TABLE pizza_types  
ALTER COLUMN pizza_type_id VARCHAR(MAX);  
ALTER TABLE pizzas  
ALTER COLUMN pizza_type_id VARCHAR(MAX);
```

121 %

Results Messages

	name	highest_priced_pizza
1	The Greek Pizza	35.95



Q. Identify the most common pizza size ordered

```
SELECT TOP 1 pizzas.size, count(order_details.order_details_id) AS most_ordered_pizza
FROM Pizzas INNER JOIN order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY most_ordered_pizza DESC;

ALTER TABLE pizzas
ALTER COLUMN size VARCHAR(MAX);
```

121 %

Results Messages

	size	most_ordered_pizza
1	L	18526

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

```
PIZZA-SQL.sql - LA...IPV\S. AMRITA (61))*  ~vsEF04.sql - LAPT...IPV\S. AMRITA (60))

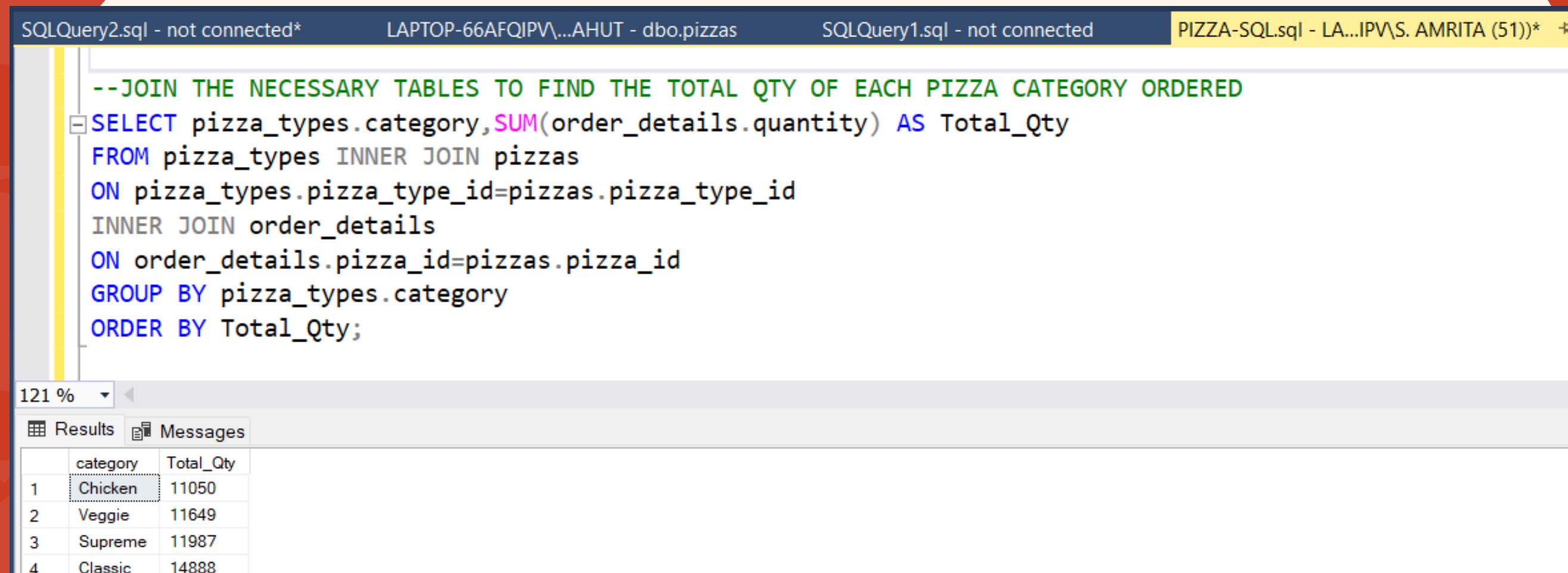
--List the top 5 most order pizza types along with their quantities
SELECT TOP 5 pizza_types.name, SUM(order_details.quantity) AS Total_Qty
FROM pizza_types INNER JOIN pizzas
ON pizza_types.pizza_type_id=pizzas.pizza_type_id
INNER JOIN order_details
ON order_details.pizza_id=pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY Total_Qty DESC;
```

120 %

Results Messages

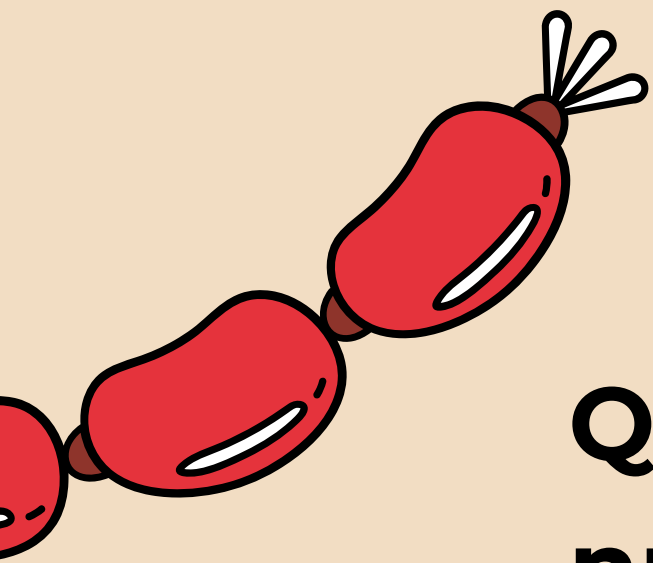
	name	Total_Qty
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418
5	The Thai Chicken Pizza	2371

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



```
--JOIN THE NECESSARY TABLES TO FIND THE TOTAL QTY OF EACH PIZZA CATEGORY ORDERED
SELECT pizza_types.category, SUM(order_details.quantity) AS Total_Qty
FROM pizza_types INNER JOIN pizzas
ON pizza_types.pizza_type_id=pizzas.pizza_type_id
INNER JOIN order_details
ON order_details.pizza_id=pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY Total_Qty;
```

	category	Total_Qty
1	Chicken	11050
2	Veggie	11649
3	Supreme	11987
4	Classic	14888



Q.Determine the distributions by hour by the number of orders

```
--Determine the distributions by hour of the day

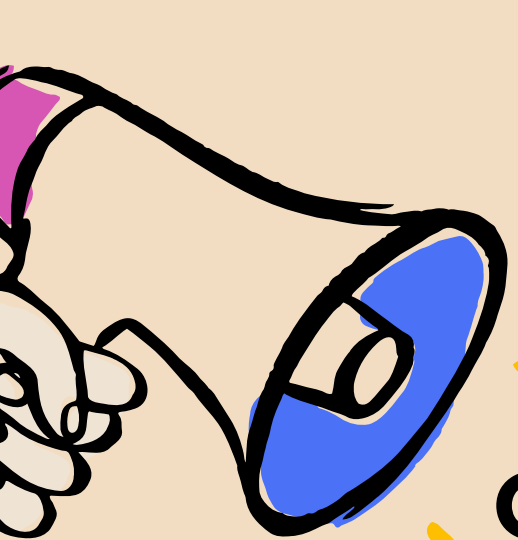
SELECT DATEPART(HOUR,time) AS HourValue,COUNT(order_id) AS Count_Order
FROM orders
GROUP BY DATEPART(HOUR,time)
ORDER BY HourValue;
```

120 %

Results Messages

	HourValue	Count_Order
1	9	1
2	10	8
3	11	1231
4	12	2520
5	13	2455
6	14	1472
7	15	1468
8	16	1920
9	17	2336
10	18	2399
11	19	2009
12	20	1642
13	21	1198
14	22	663
15	23	28





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



```
--Calculate the percentage contribution of each pizza type to total revenue
SELECT
  pizza_types.category,
  ROUND(
    (SUM(pizzas.price * order_details.quantity) * 100.0) /
    (SELECT ROUND(SUM(pizzas.price * order_details.quantity), 2)
     FROM order_details
     INNER JOIN pizzas ON pizzas.pizza_id = order_details.pizza_id),
    2) AS Contribution_Revenue
FROM
  pizza_types
INNER JOIN
  pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
INNER JOIN
  order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY
  pizza_types.category
ORDER BY
  Contribution_Revenue DESC;
```

120 %

Results Messages

	category	Contribution_Revenue
1	Classic	26.91
2	Supreme	25.46
3	Chicken	23.96
4	Veggie	23.68



Q.Analyze the cumulative revenue generated over time

PIZZA-SQL.sql - LA...IPV\S. AMRITA (61))* X ~vsEF04.sql - LAPT...IPV\S. AMRITA (60))

```
--Analyze the cumulative revenue generated over time

SELECT Sales.date, SUM(Sales.Total_Revenue) OVER (ORDER BY Sales.date) AS CUM_Revenue
FROM (SELECT orders.date, ROUND(SUM(pizzas.price * order_details.quantity), 2) AS Total_Revenue
FROM orders INNER JOIN order_details ON orders.order_id = order_details.order_id
INNER JOIN pizzas ON order_details.pizza_id = pizzas.pizza_id
GROUP BY orders.date) AS Sales
ORDER BY Sales.date;

--OR

SELECT orders.date, ROUND(SUM(pizzas.price * order_details.quantity), 2) AS Total_Revenue,
ROUND(SUM(SUM(pizzas.price * order_details.quantity)) OVER (ORDER BY orders.date), 2) AS CUM_Revenue
FROM orders INNER JOIN order_details ON orders.order_id = order_details.order_id
INNER JOIN pizzas ON order_details.pizza_id = pizzas.pizza_id
GROUP BY orders.date
ORDER BY orders.date;
```

120 %

Results Messages

	date	Total_Revenue	CUM_Revenue
1	2015-01-01	2713.85	2713.85
2	2015-01-02	2731.9	5445.75
3	2015-01-03	2662.4	8108.15
4	2015-01-04	1755.45	9863.6
5	2015-01-05	2065.95	11929.55
6	2015-01-06	2428.95	14358.5
7	2015-01-07	2202.2	16560.7
8	2015-01-08	2838.35	19399.05
9	2015-01-09	2127.35	21526.4
10	2015-01-10	2463.95	23990.35
11	2015-01-11	1872.3	25862.65
12	2015-01-12	1919.05	27781.7
13	2015-01-13	2049.6	29831.3
14	2015-01-14	2527.4	32358.7
15	2015-01-15	1984.8	34343.5

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

```
--Determine the top 3 most ordered pizza types based on revenue for each category
WITH RankedPizzas AS (
    SELECT pt.category, pt.name AS pizza_name,
           ROUND(SUM(od.quantity * p.price), 2) AS Total_Revenue,
           DENSE_RANK() OVER (PARTITION BY pt.category ORDER BY SUM(od.quantity * p.price) DESC) AS RevenueRank
    FROM pizza_types pt
    JOIN pizzas p ON pt.pizza_type_id = p.pizza_type_id
    JOIN order_details od ON od.pizza_id = p.pizza_id
    GROUP BY pt.category, pt.name
)
SELECT category, pizza_name, Total_Revenue, RevenueRank
FROM RankedPizzas
WHERE RevenueRank <= 3
ORDER BY category, Total_Revenue DESC;
```

120 %

Results Messages

	category	pizza_name	Total_Revenue	RevenueRank
1	Chicken	The Thai Chicken Pizza	43434.25	1
2	Chicken	The Barbecue Chicken Pizza	42768	2
3	Chicken	The California Chicken Pizza	41409.5	3
4	Classic	The Classic Deluxe Pizza	38180.5	1
5	Classic	The Hawaiian Pizza	32273.25	2
6	Classic	The Pepperoni Pizza	30161.75	3
7	Supreme	The Spicy Italian Pizza	34831.25	1
8	Supreme	The Italian Supreme Pizza	33476.75	2
9	Supreme	The Sicilian Pizza	30940.5	3
10	Veggie	The Four Cheese Pizza	32265.7	1
11	Veggie	The Mexicana Pizza	26780.75	2
12	Veggie	The Five Cheese Pizza	26066.5	3

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

```
PIZZA-SQL.sql - LA...IPV\S. AMRITA (61))* X ~vsEF04.sql - LAPT...IPV\S. AMRITA (60))

--Determine the top3 most ordered pizza types based on revenue


SELECT TOP 3 pizza_types.name, round(sum(pizzas.price*order_details.quantity),2) AS Total_Revenue
FROM pizza_types INNER JOIN pizzas
ON pizza_types.pizza_type_id=pizzas.pizza_type_id
INNER JOIN order_details
ON order_details.pizza_id=pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY Total_Revenue DESC;
```

120 %

Results Messages

	name	Total_Revenue
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5

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



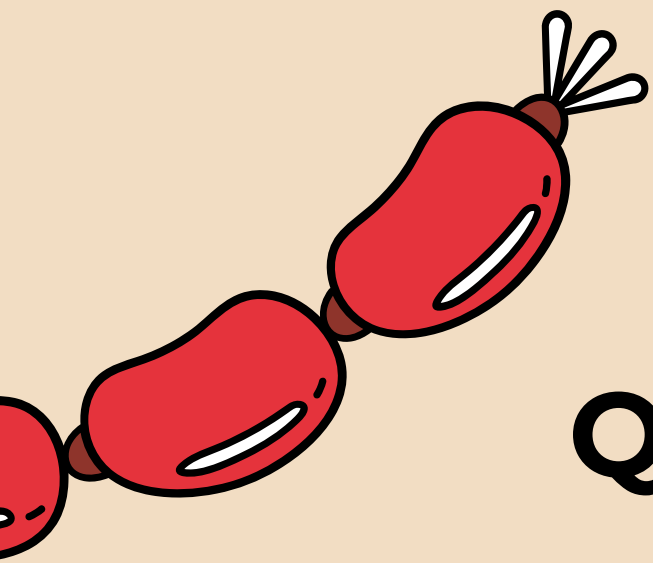
```
PIZZA-SQL.sql - LA...IPV\S. AMRITA (61))* X ~vsEF04.sql - LAPT...IPV\S. AMRITA (60))

--Group the orders by date & calculate the average number of pizzas ordered per day
SELECT ROUND(AVG(DailyTotals.total_quantity), 0) AS avg_pizzas_per_day
FROM (
    SELECT orders.date, SUM(order_details.quantity) AS total_quantity
    FROM orders
    INNER JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.date
) AS DailyTotals;
```

120 %

Results Messages

	avg_pizzas_per_day
1	138



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

PIZZA-SQL.sql - LA...IPV\S. AMRITA (61))* X ~vsEF04.sql - LAPT...IPV\S. AMRITA (60))

```
--Join relevant tables to find the category-wise distribution of pizzas.  
SELECT Category,COUNT(name) AS Pizza_Count  
FROM pizza_types  
GROUP BY category;
```

120 %

Results Messages

	Category	Pizza_Count
1	Chicken	6
2	Classic	8
3	Supreme	9
4	Veggie	9



Thank You!



Email

amritasamuel2@gmail.com



Contact

via mail



Website

www.linkedin.com/in/amrita-samuel234



Phone Number

-

