THE PIZZA POINT

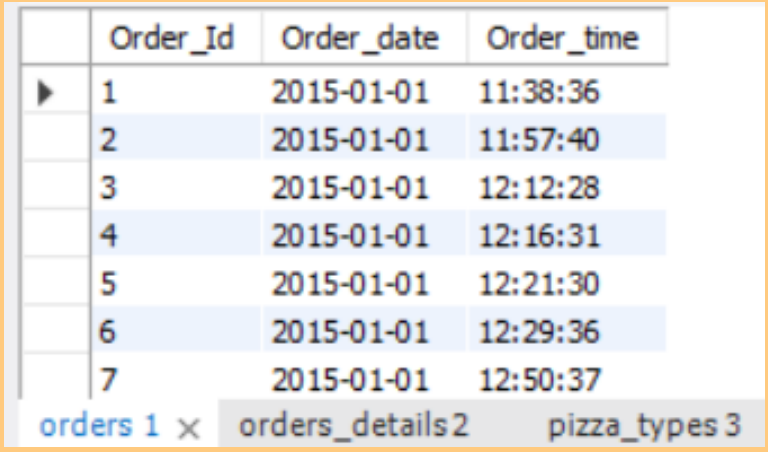
RESTAURANT SALES
SQL PROJECT

PRESENTED BY
BHAWNA GUPTA

CREATE DATABASE PIZZAPOINT AND LOAD TABLES

```
Create Database Pizzapoint;
```

TABLE NAMES
ORDERS, ORDERS_DETAILS, PIZZA_TYPES, PIZZAS)

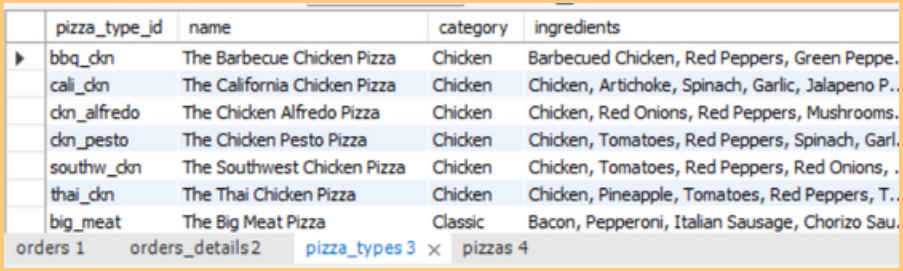


Order_Id	Order_date	Order_time
1	2015-01-01	11:38:36
2	2015-01-01	11:57:40
3	2015-01-01	12:12:28
4	2015-01-01	12:16:31
5	2015-01-01	12:21:30
6	2015-01-01	12:29:36
7	2015-01-01	12:50:37

orders 1 × orders_details 2 pizza_types 3

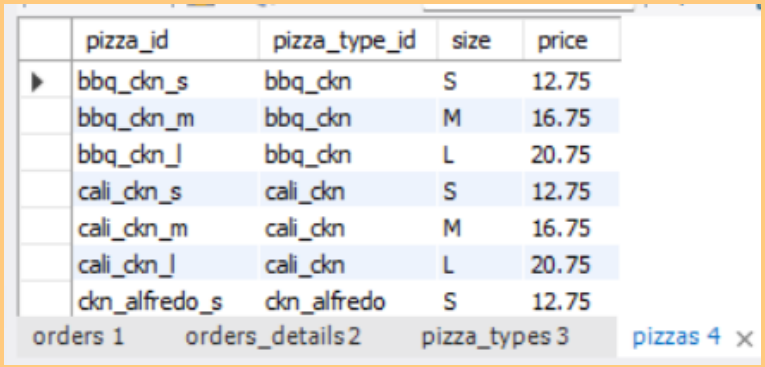
Order_details_Id	Order_Id	Pizza_Id	Quantity
1	1	hawaiian_m	1
2	2	classic_dlx_m	1
3	2	five_cheese_l	1
4	2	ital_supr_l	1
5	2	mexicana_m	1
6	2	thai_ckn_l	1
7	3	ital_supr_m	1

orders 1 orders_details 2 × pizza_types 3 pizzas 4



pizza_type_id	name	category	ingredients
bbq_ckn	The Barbecue Chicken Pizza	Chicken	Barbecued Chicken, Red Peppers, Green Peppe...
cali_ckn	The California Chicken Pizza	Chicken	Chicken, Artichoke, Spinach, Garlic, Jalapeno P...
ckn_alfredo	The Chicken Alfredo Pizza	Chicken	Chicken, Red Onions, Red Peppers, Mushrooms...
ckn_pesto	The Chicken Pesto Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Spinach, Garl...
southw_ckn	The Southwest Chicken Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Red Onions, ...
thai_ckn	The Thai Chicken Pizza	Chicken	Chicken, Pineapple, Tomatoes, Red Peppers, T...
big_meat	The Big Meat Pizza	Classic	Bacon, Pepperoni, Italian Sausage, Chorizo Sau...

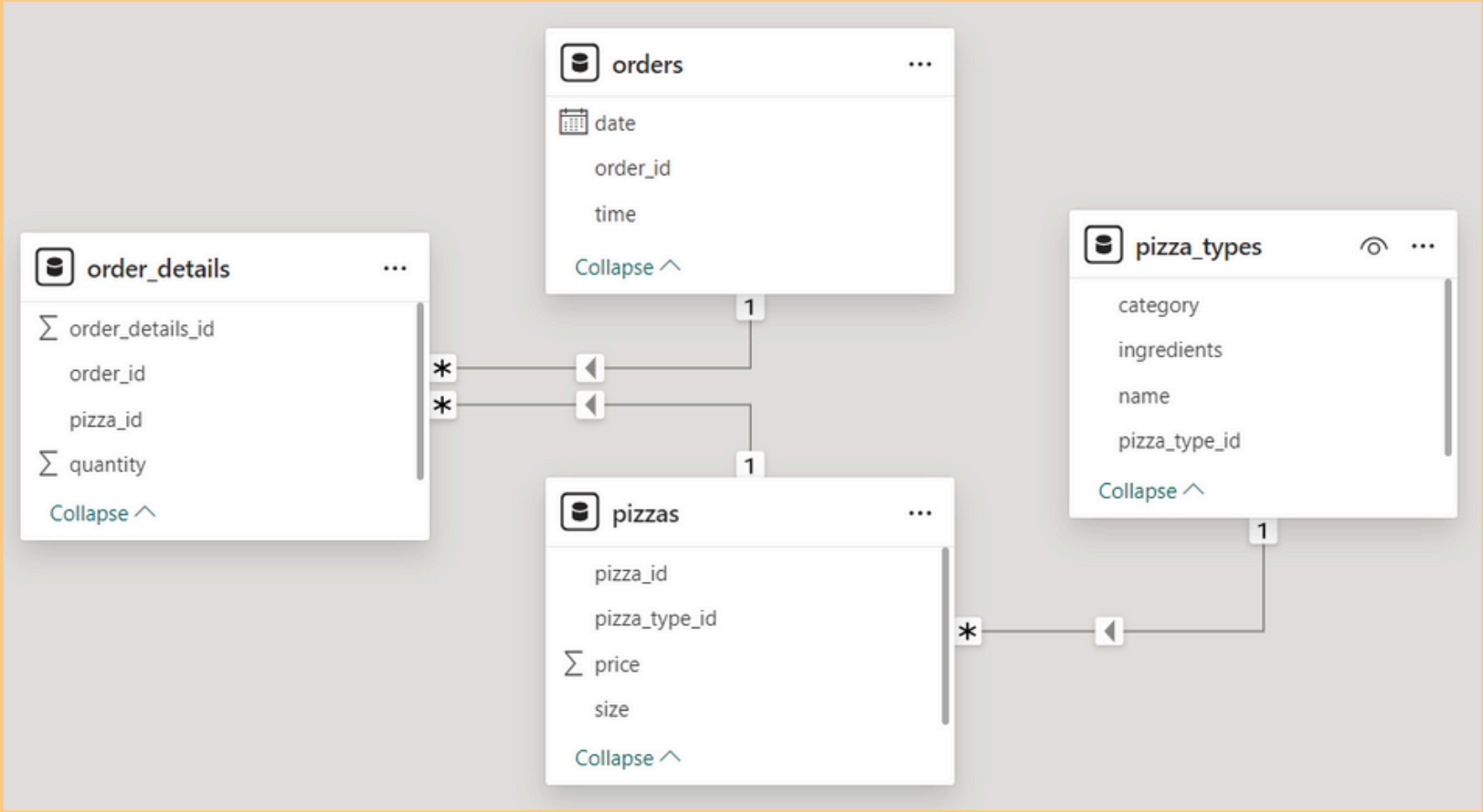
orders 1 orders_details 2 pizza_types 3 × pizzas 4



pizza_id	pizza_type_id	size	price
bbq_ckn_s	bbq_ckn	S	12.75
bbq_ckn_m	bbq_ckn	M	16.75
bbq_ckn_l	bbq_ckn	L	20.75
cali_ckn_s	cali_ckn	S	12.75
cali_ckn_m	cali_ckn	M	16.75
cali_ckn_l	cali_ckn	L	20.75
ckn_alfredo_s	ckn_alfredo	S	12.75

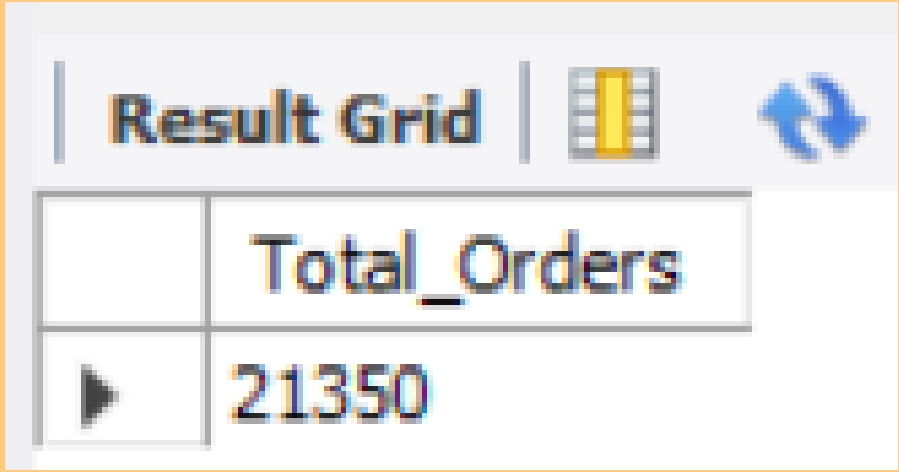
orders 1 orders_details 2 pizza_types 3 pizzas 4 ×

MODEL VIEW


orders
date
order_id
time
Collapse
order_details
order_details_id
order_id
pizza_id
quantity
Collapse
pizza_types
category
ingredients
name
pizza_type_id
Collapse
pizzas
pizza_id
pizza_type_id
price
size
Collapse
1
*
*
1
1
*

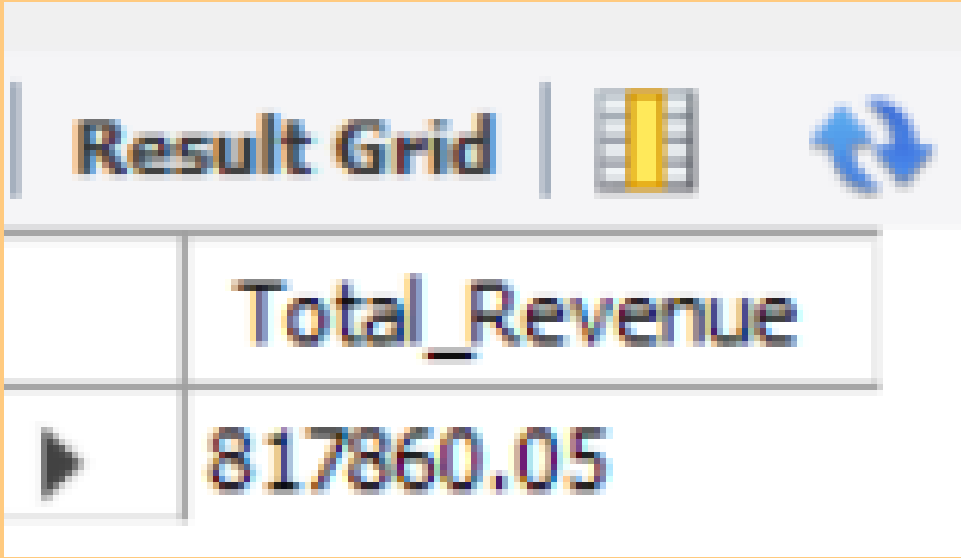
RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED

```sql
SELECT
    COUNT(Order_Id) AS Total_Orders
FROM
    orders;
```



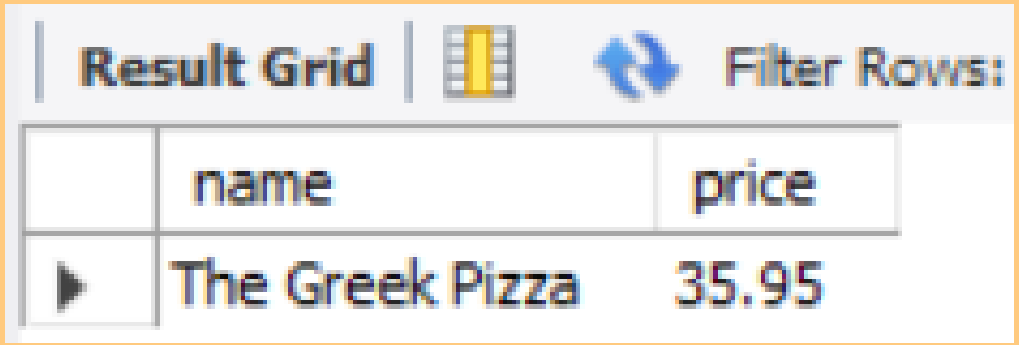
CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

```sql
SELECT
    ROUND(SUM(d.Quantity * p.price), 2) AS Total_Revenue
FROM
    orders_details d
        JOIN
    pizzas p ON d.Pizza_Id = p.Pizza_Id;
```



IDENTIFY THE HIGHEST-PRICED PIZZA

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



IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

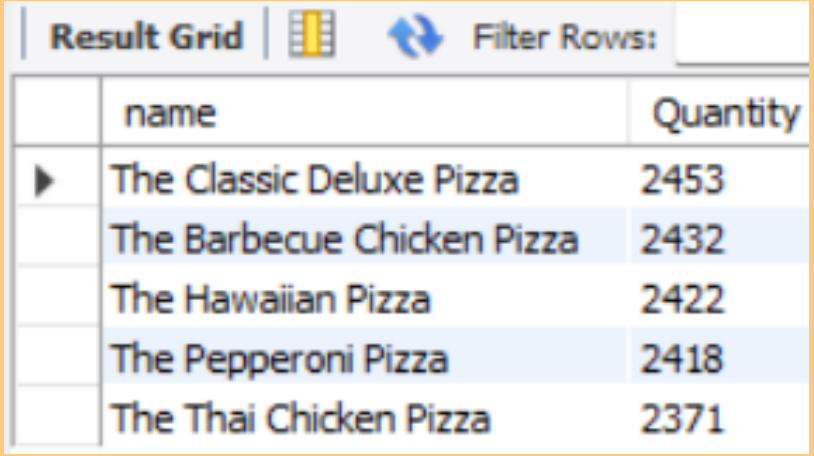
```sql
SELECT
    p.size, COUNT(d.quantity) AS Size_Count
FROM
    orders_details d
        JOIN
    pizzas p ON d.Pizza_Id = p.pizza_id
GROUP BY size
ORDER BY Size_Count DESC;
```

Result Grid

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

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

```sql
SELECT
    t.name, SUM(d.Quantity) AS Quantity
FROM
    pizza_types t
        JOIN
    pizzas p ON t.pizza_type_id = p.pizza_type_id
        JOIN
    orders_details d ON p.Pizza_Id = d.Pizza_Id
GROUP BY name
ORDER BY Quantity DESC
LIMIT 5;
```



Result Grid | Filter Rows:

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

JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS

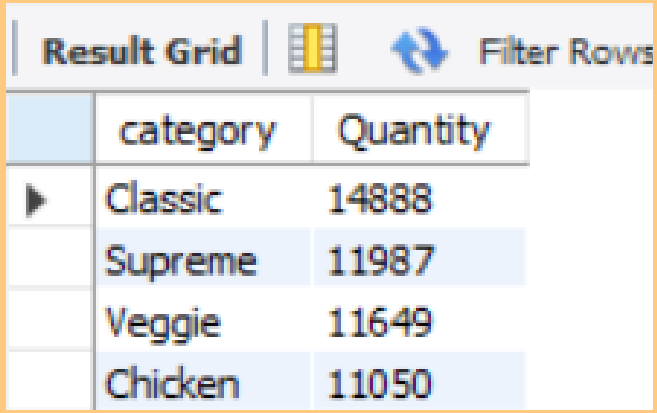
```
Select category, Count(category) As Count From pizza_types
Group by category;
```

Result Grid

	category	Count
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

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

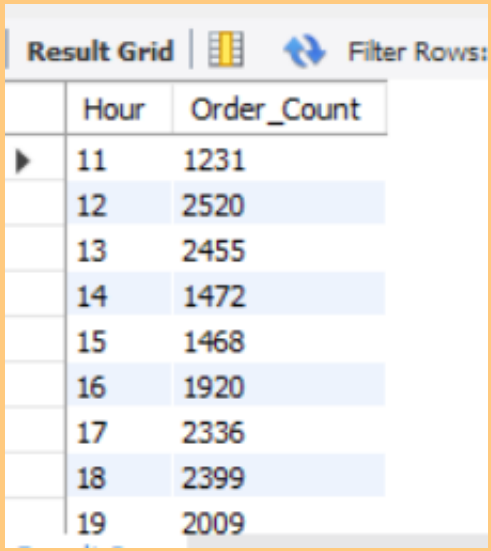
```sql
SELECT
    t.category, SUM(d.Quantity) AS Quantity
FROM
    pizza_types t
        JOIN
    pizzas p ON t.pizza_type_id = p.pizza_type_id
        JOIN
    orders_details d ON p.Pizza_Id = d.Pizza_Id
GROUP BY category
ORDER BY Quantity DESC;
```



category	Quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

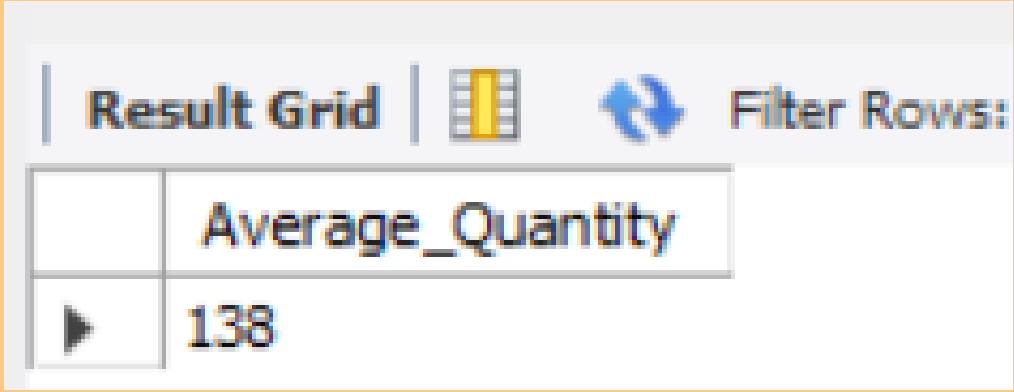
```sql
SELECT
    HOUR(order_time) As Hour, COUNT(order_id) As Order_Count
FROM
    orders
GROUP BY HOUR(order_time);
```



Hour	Order_Count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009

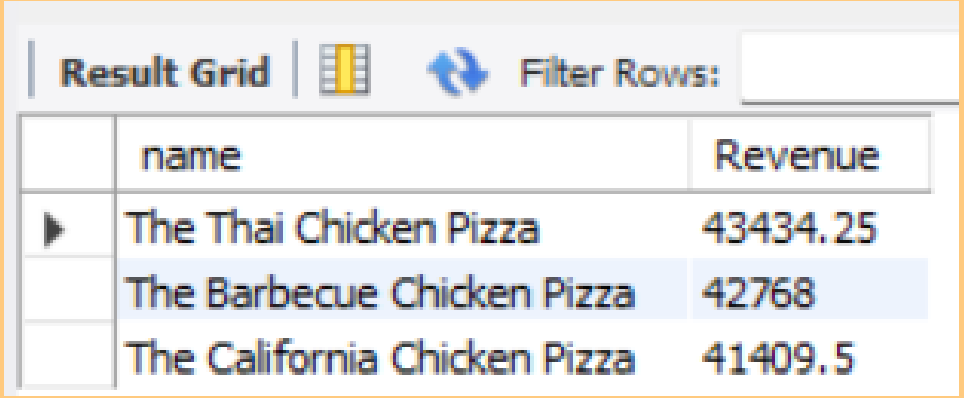
GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY

```sql
SELECT
    ROUND(AVG(Quantity), 0) AS Average_Quantity
FROM
    (SELECT
        o.order_date, SUM(d.Quantity) AS Quantity
    FROM
        orders o
    JOIN orders_details d ON o.Order_Id = d.Order_Id
    GROUP BY order_date) AS Order_Quantity;
```



DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE

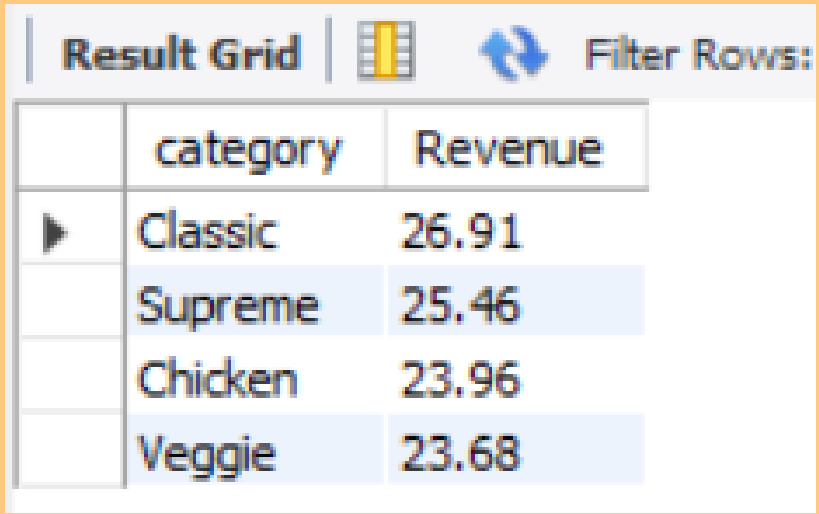
```sql
SELECT
    t.name, SUM(d.Quantity * p.price) AS Revenue
FROM
    orders_details d
        JOIN
    pizzas p ON d.Pizza_Id = p.Pizza_Id
        JOIN
    pizza_types t ON t.pizza_type_id = p.pizza_type_id
GROUP BY name
ORDER BY Revenue DESC
LIMIT 3;
```



name	Revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE

```sql
SELECT
t.category, Round(SUM(d.Quantity * p.price)/(Select Sum(d.Quantity * p.price) As Total_Sales FROM
orders_details d JOIN pizzas p ON d.Pizza_Id = p.Pizza_Id) *100, 2) As Revenue
FROM
    orders_details d
        JOIN
    pizzas p ON d.Pizza_Id = p.Pizza_Id
        JOIN
    pizza_types t ON t.pizza_type_id = p.pizza_type_id
GROUP BY category
ORDER BY Revenue DESC;
```



	category	Revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

```sql
Select order_date, Sum(Revenue) Over (Order By order_date) as Cumulative
From
(SELECT
    o.order_date, Round(SUM(d.Quantity * p.price), 2) AS Revenue
FROM
    orders_details d
        JOIN
    pizzas p ON d.Pizza_Id = p.Pizza_Id
        JOIN
    orders o ON o.order_id = d.order_id
GROUP BY o.order_date) as Sales;
```

Result Grid | Filter Rows:

order_date	Cumulative
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

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

```sql
SELECT category, name, Revenue From
(SELECT category, name, Revenue, rank() Over(
Partition By category Order By Revenue DESC) As Ranking From
(
SELECT
    t.category, t.name, Round(SUM(d.Quantity * p.price), 2) AS Revenue FROM
    pizza_types t Join pizzas p On t.pizza_type_id = p.pizza_type_id
        JOIN
    orders_details d ON d.Pizza_Id = p.Pizza_Id
    GROUP BY t.category, t.name) As a) As b
    WHERE Ranking <=3;
```

Result Grid | Filter Rows:

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

PROJECT INSIGHTS

- **Total Sales Revenue:** Understanding the overall revenue generated over a specific period can help evaluate the financial performance of the restaurant.
- **Popular Menu Items**: Identifying the best-selling dishes or items can help optimize the menu, pricing, and promotional strategies.
- **Customer Preferences:** Analyzing customer orders can reveal preferences for specific category, size, or ingredients, enabling the restaurant to tailor their offerings.
- **Peak Hours:** Identifying peak times of customer orders can help optimize staff scheduling, inventory management, and service efficiency.
- **Average Order Value:** Calculating the average amount spent per customer can provide insights into customer spending habits and help increase profitability.