PIZZA SALES ANALYSIS- SQL PROJECT

FATHIMA S

HELLO, I'M FATHIMA S.,

A PASSIONATE AND DETAIL-ORIENTED GRADUATE WITH AN INTEREST IN DATA ANALYSIS AND SQL-BASED PROJECTS. THIS PRESENTATION SHOWCASES ONE OF MY RECENT HANDS-ON PROJECTS FOCUSED ON ANALYZING PIZZA SALES USING SQL.

PROJECT OVERVIEW

THIS PROJECT AIMS TO ANALYZE PIZZA SALES DATA TO UNCOVER MEANINGFUL INSIGHTS SUCH AS TOP-SELLING PIZZAS, REVENUE TRENDS, CUSTOMER PREFERENCES, AND OVERALL BUSINESS PERFORMANCE.

USING SQL, I QUERIED A STRUCTURED RELATIONAL DATABASE TO GENERATE DATA-DRIVEN ANSWERS FOR BETTER BUSINESS DECISIONS.

DATABASE STRUCTURE

Table: pizzas

Columns:

pizza_id	text
pizza_type_id	text
size	text
price	double

Table: pizza_types

Columns:

pizza_type_id	text
name	text
category	text
ingredients	text

Table: orders

Columns:

order_id	int PK
order_date	date
order_time	time

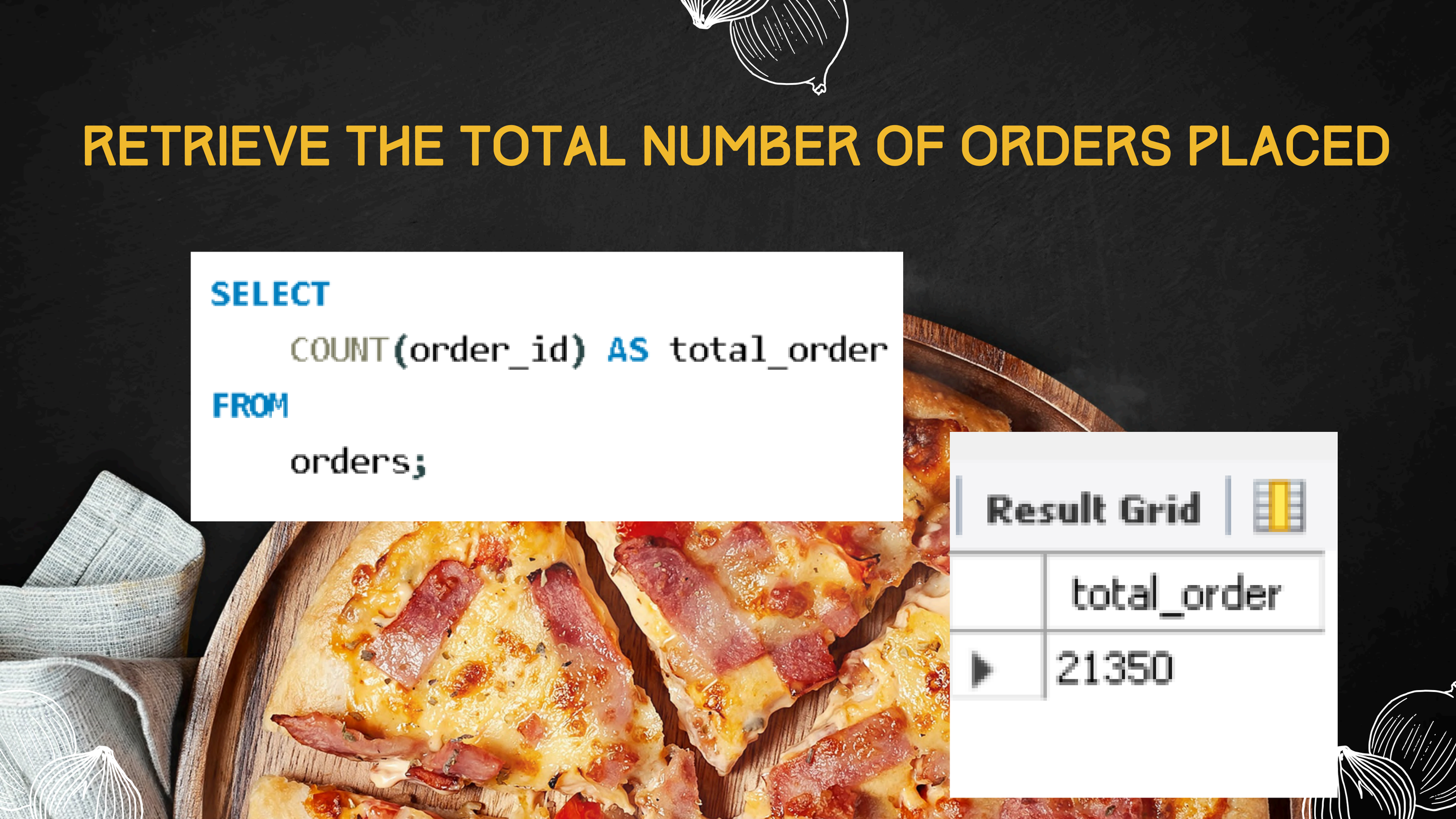
Table: order_details

Columns:

order_details_id	int PK
order_id	int
pizza_id	text
quantity	int

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED

```sql
SELECT
    COUNT(order_id) AS total_order
FROM
    orders;
```



Result Grid

	total_order
▶	21350

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

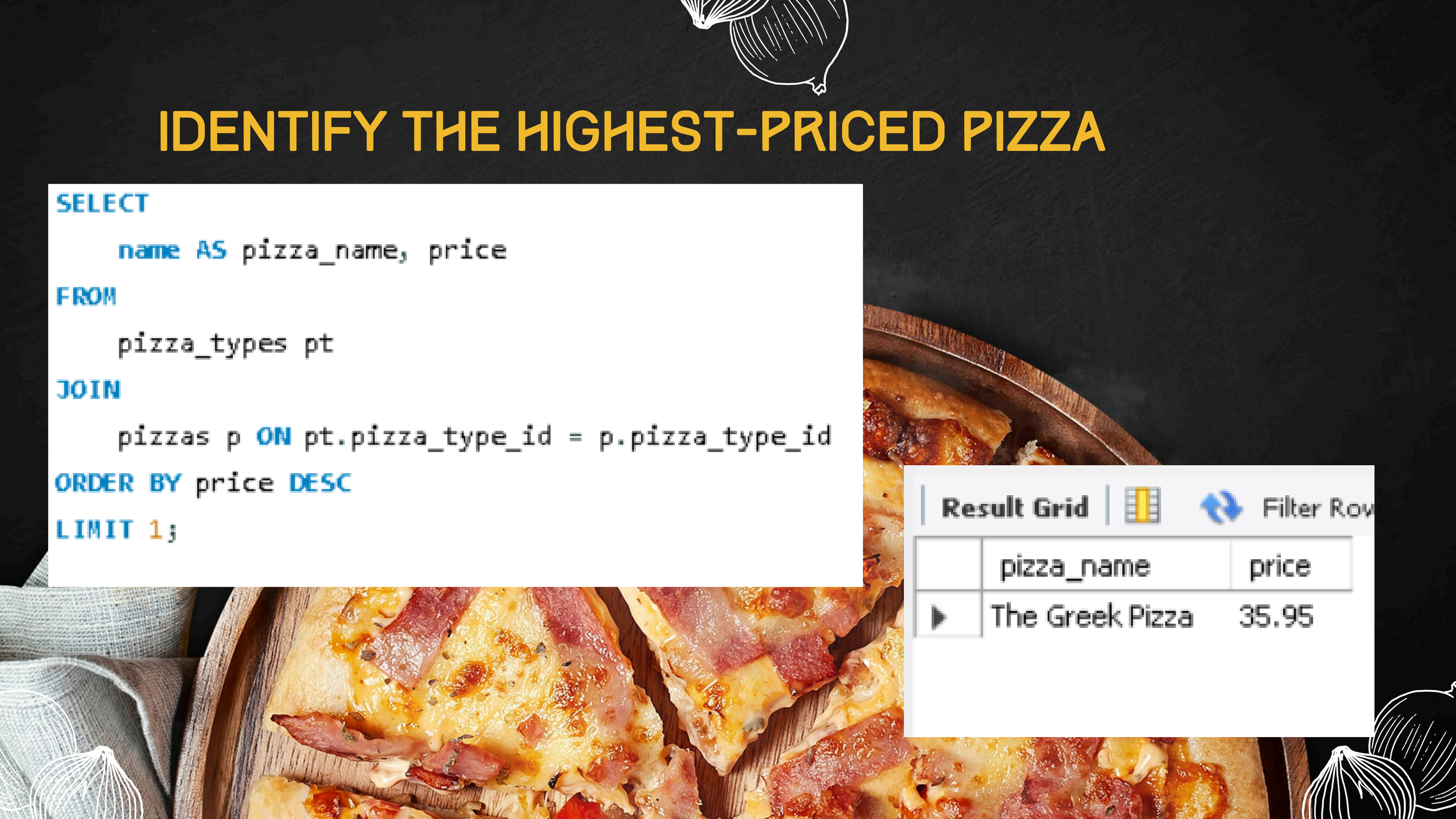
```sql
SELECT
    ROUND(SUM(price * quantity), 2) AS total_revenue
FROM
    pizzas p
        JOIN
    order_details o ON p.pizza_id = o.pizza_id;
```

Result Grid

	total_revenue
▶	817860.05

IDENTIFY THE HIGHEST-PRICED PIZZA

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



Result Grid | Filter Rov

	pizza_name	price
▶	The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

```sql
SELECT
    size, COUNT(order_details_id) AS Count
FROM
    pizzas p
        JOIN
    order_details o ON p.pizza_id = o.pizza_id
GROUP BY size
ORDER BY Count DESC
LIMIT 1;
```

Result Grid

	size	Count
▶	L	18526

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

```sql
SELECT
    name AS Name, SUM(quantity) AS Quantity
FROM
    pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details or1 ON p.pizza_id = or1.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 THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED

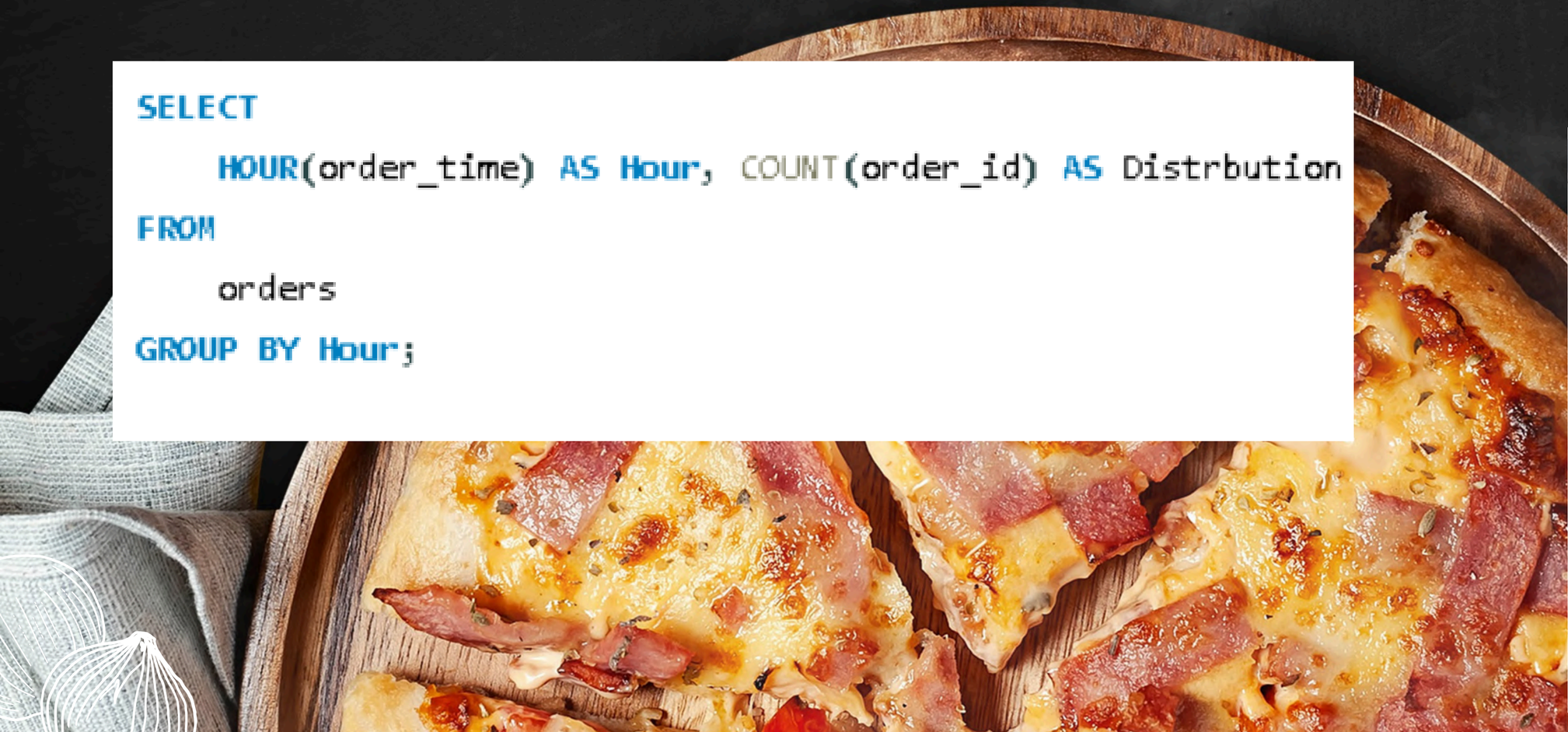
```sql
SELECT
    category, SUM(quantity) AS Total
FROM
    pizzas p
        JOIN
    order_details o ON p.pizza_id = o.pizza_id
        JOIN
    pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
GROUP BY category;
```

Result Grid | Filter Rows

category	Total
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

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



Result Grid

Hour	Distrbution
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

```sql
SELECT
    category, COUNT(name) AS Dstribution
FROM
    pizza_types
GROUP BY category;
```

Result Grid | Filter Rov

	category	Dstribution
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

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

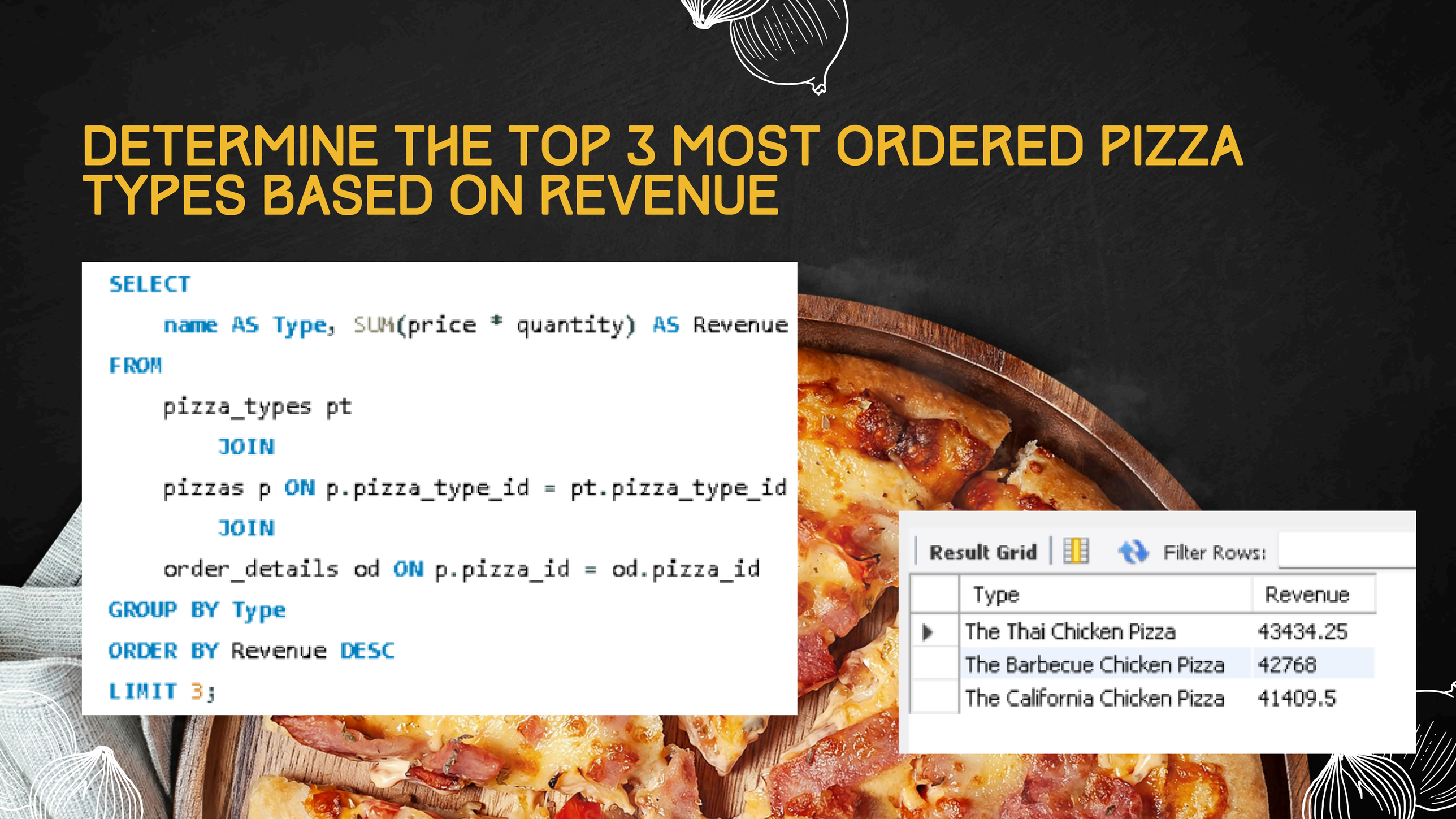
```sql
SELECT ROUND(AVG(Average),0) AS Average_Per_Day
FROM (
    SELECT order_date AS Date, SUM(quantity) AS Average
    FROM orders o
    JOIN order_details od ON o.order_id = od.order_id
    GROUP BY order_date
) daily_totals;
```

Result Grid | Filter

	Average_Per_Day
▶	138

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE

```sql
SELECT
    name AS Type, SUM(price * quantity) AS Revenue
FROM
    pizza_types pt
        JOIN
    pizzas p ON p.pizza_type_id = pt.pizza_type_id
        JOIN
    order_details od ON p.pizza_id = od.pizza_id
GROUP BY Type
ORDER BY Revenue DESC
LIMIT 3;
```

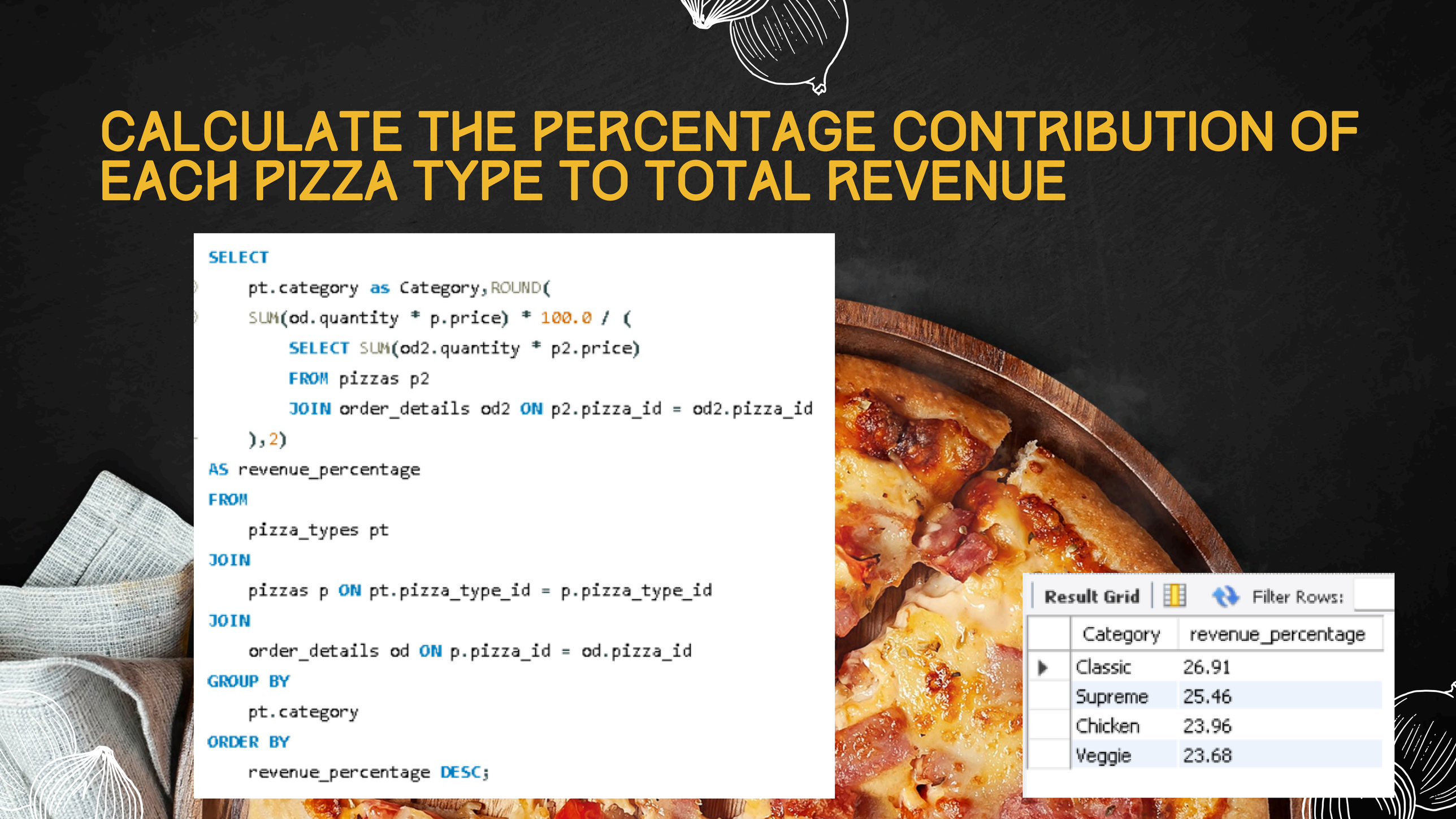


Result Grid | Filter Rows:

Type	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
    pt.category as Category,ROUND(
    SUM(od.quantity * p.price) * 100.0 / (
        SELECT SUM(od2.quantity * p2.price)
        FROM pizzas p2
        JOIN order_details od2 ON p2.pizza_id = od2.pizza_id
    ),2)
AS revenue_percentage
FROM
    pizza_types pt
JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
JOIN
    order_details od ON p.pizza_id = od.pizza_id
GROUP BY
    pt.category
ORDER BY
    revenue_percentage DESC;
```



Result Grid | Filter Rows:

Category	revenue_percentage
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

```
select order_date,sum(revenue) over(order by order_date) as Cum_revenue from
(select order_date,sum(price*quantity) as revenue from
order_details join orders on  orders.order_id=order_details.order_id
join pizzas on
 pizzas.pizza_id=order_details.pizza_id
 group by order_date) as sales;
```

Result Grid | Filter Rows:

order_date	Cum_revenue
2015-01-01	2713.850000000004
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	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004
2015-01-15	34343.50000000001
2015-01-16	36937.65000000001
2015-01-17	39001.75000000001

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 rn from

(select pizza_types.category,pizza_types.name,sum(order_details.quantity*pizzas.price) as Revenue
from order_details join pizzas on
order_details.pizza_id=pizzas.pizza_id
join pizza_types on
pizza_types.pizza_type_id=pizzas.pizza_type_id
group by pizza_types.category,pizza_types.name) as a) as b where rn<=3;
```

Result Grid | Filter Rows: | Export:

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	32265.70000000065
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5

THANK YOU!

My_portfolio

linkedin.com/in/fathima-s-309a00224

FATHIMA S