SQL PROJECT ON PIZZA SALES

PIZZA SALES

TITLE:"PIZZA SALES ANALYSIS USING MYSQL"

INTRODUCTION:

THIS PROJECT FOCUSES ON ANALYZING PIZZA SALES DATA TO DERIVE ACTIONABLE INSIGHTS. BY LEVERAGING MYSQL, WE AIM TO:

- UNDERSTAND SALES TRENDS AND CUSTOMER PREFERENCES.
- OPTIMIZE BUSINESS STRATEGIES THROUGH DATA-DRIVEN DECISIONS.
- IDENTIFY KEY PERFORMANCE METRICS SUCH AS REVENUE, ORDER FREQUENCY, AND TOP-SELLING ITEMS.

KEY OBJECTIVES:

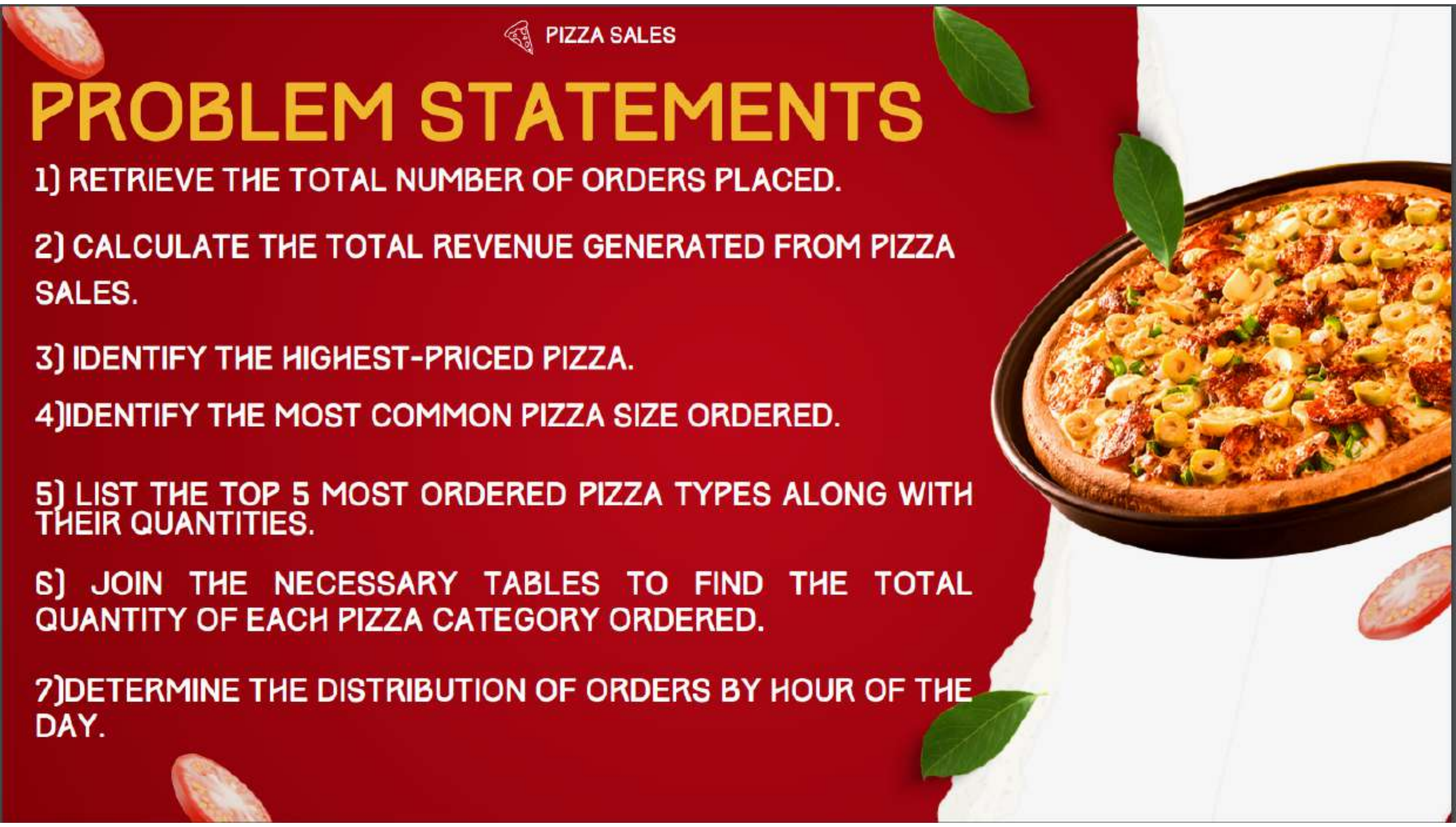
- EFFICIENTLY RETRIEVE AND ANALYZE SALES DATA.
- USE SQL QUERIES TO SOLVE REAL-WORLD BUSINESS PROBLEMS.
- HIGHLIGHT THE MOST POPULAR PIZZA TYPES, REVENUE DISTRIBUTION, AND CUSTOMER ORDERING BEHAVIOR.

PIZZA SALES

HELLO !!!

PIYUSH TIGAONKAR

I AM A PASSIONATE DATA ANALYST WITH EXPERIENCE IN MYSQL, POWER BI, AND ADVANCED EXCEL. I ENJOY SOLVING REAL-WORLD PROBLEMS THROUGH DATA-DRIVEN INSIGHTS AND RECENTLY WORKED ON A PIZZA SALES PROJECT, APPLYING ADVANCED SQL TECHNIQUES TO ANALYZE AND OPTIMIZE SALES PERFORMANCE.



PROBLEM STATEMENTS

1) RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

2) CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

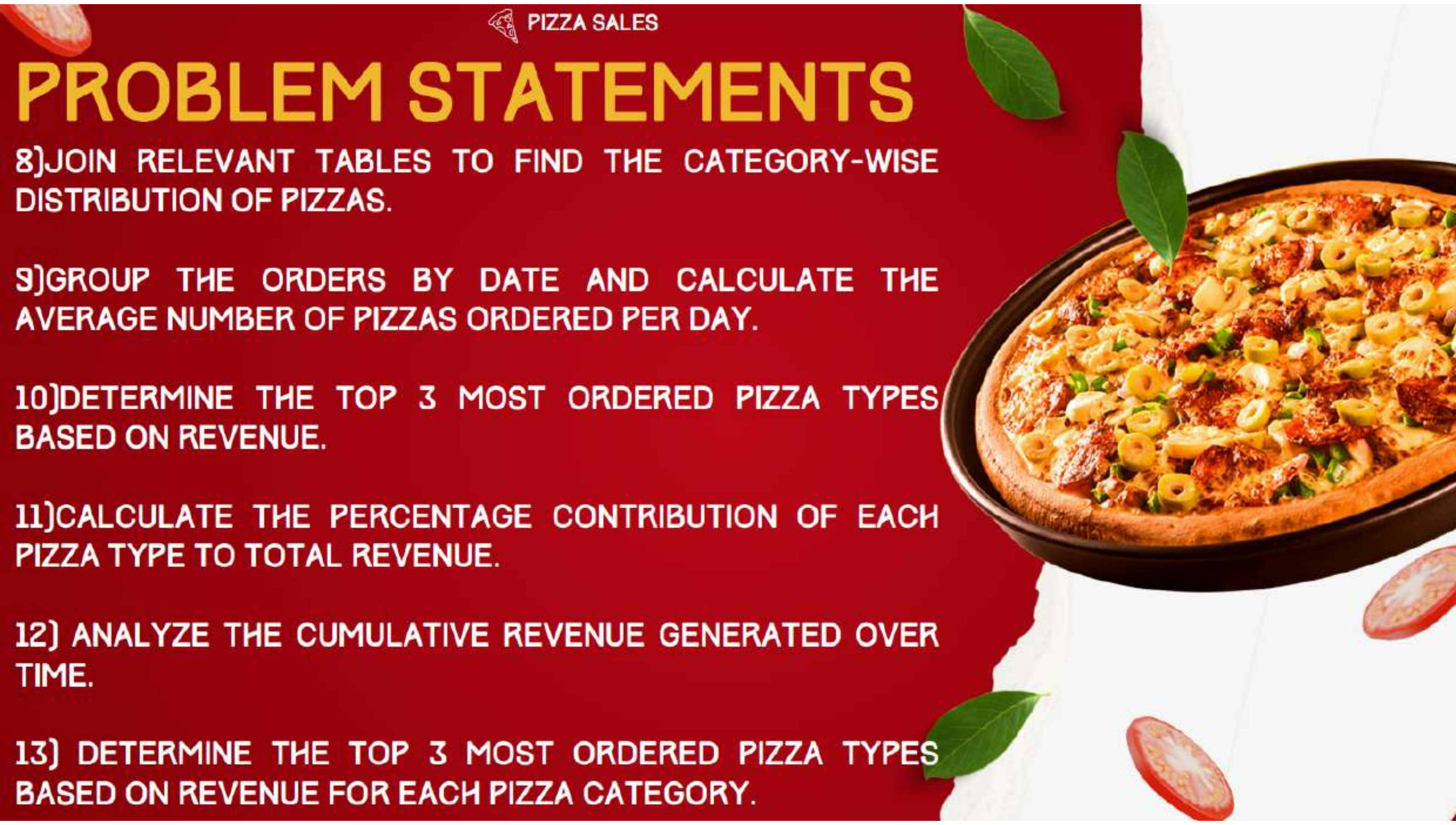
3) IDENTIFY THE HIGHEST-PRICED PIZZA.

4)IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

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

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

7)DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.



PIZZA SALES

PROBLEM STATEMENTS

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

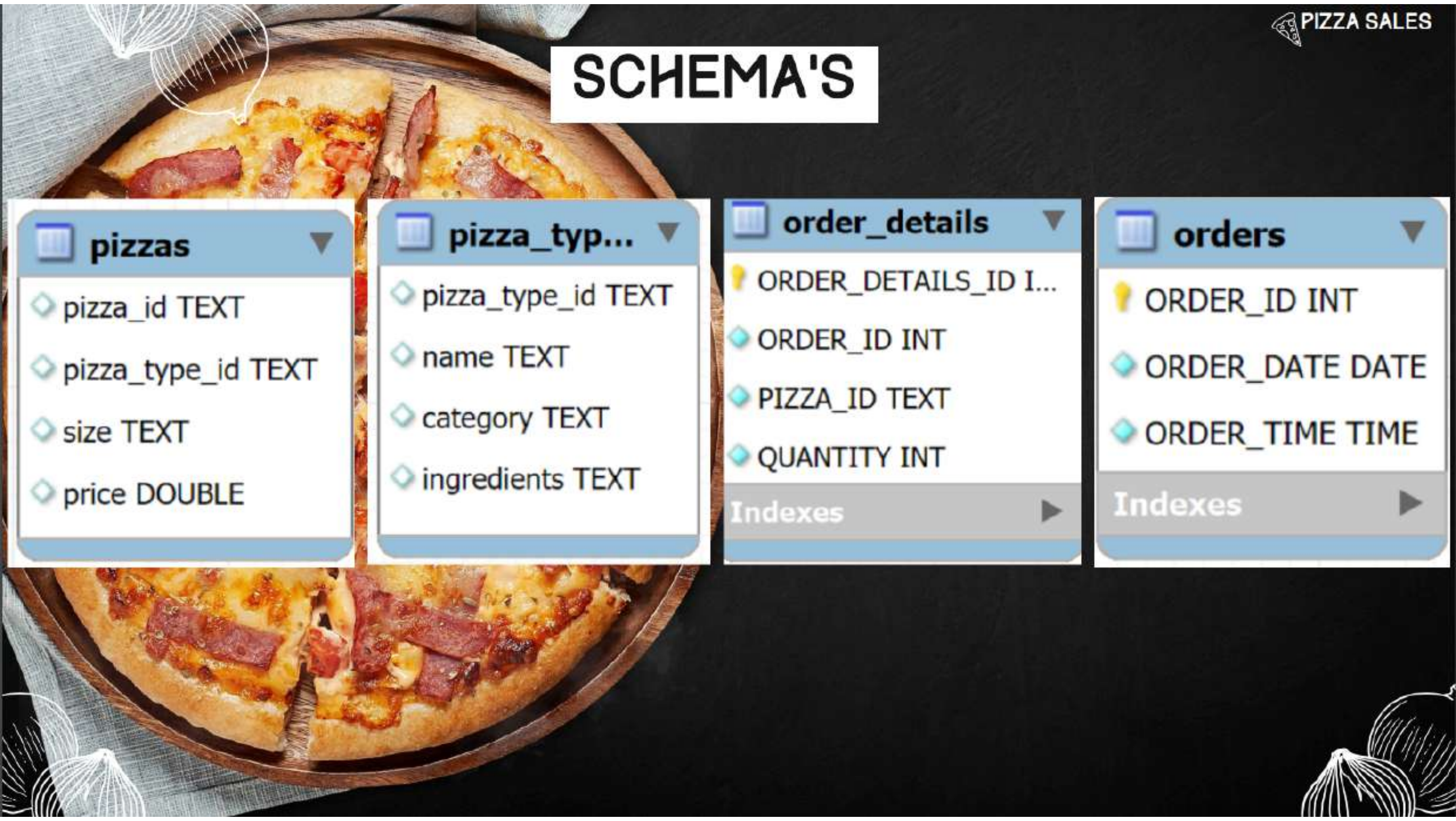
9)GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

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

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

12) ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

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



PIZZA SALES

SCHEMA'S

pizzas
- pizza_id TEXT
- pizza_type_id TEXT
- size TEXT
- price DOUBLE

pizza_typ...
- pizza_type_id TEXT
- name TEXT
- category TEXT
- ingredients TEXT

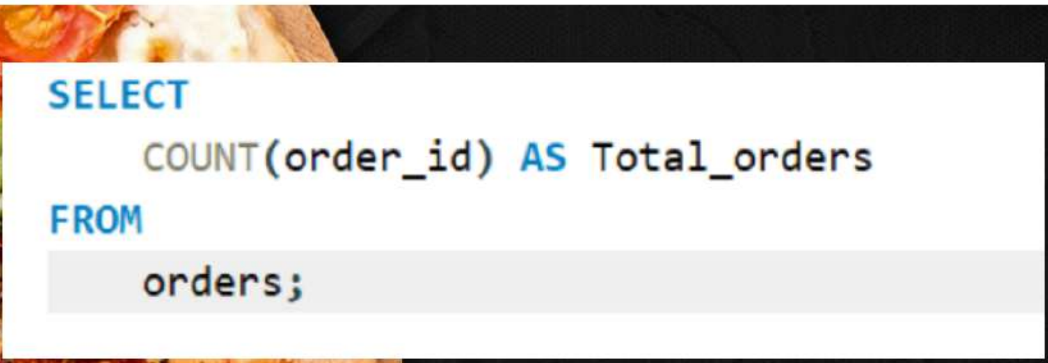
order_details
- ORDER_DETAILS_ID I...
- ORDER_ID INT
- PIZZA_ID TEXT
- QUANTITY INT

Indexes

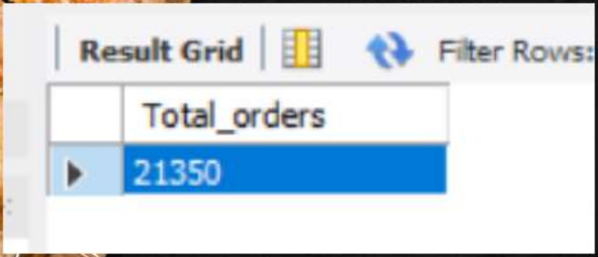
orders
- ORDER_ID INT
- ORDER_DATE DATE
- ORDER_TIME TIME

Indexes

1) RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

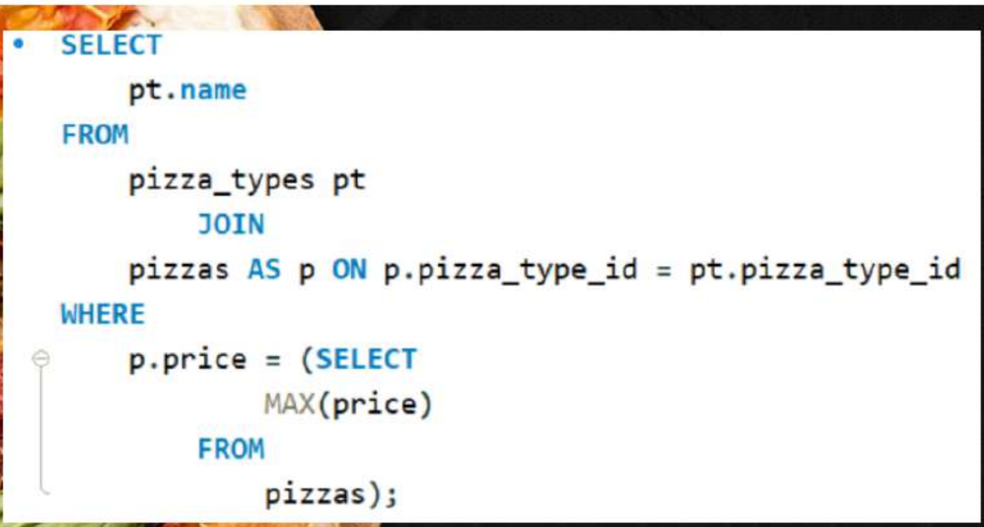


```
SELECT
    COUNT(order_id) AS Total_orders
FROM
    orders;
```

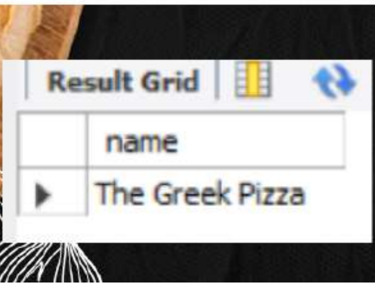


```
Total_orders
21350
```

3)IDENTIFY THE HIGHEST-PRICED PIZZA.

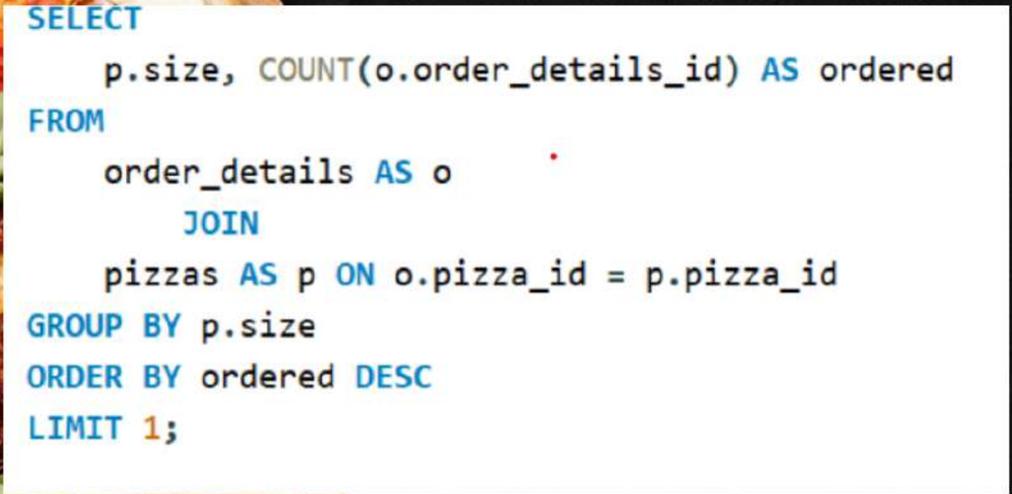


```
SELECT
    pt.name
FROM
    pizza_types pt
        JOIN
    pizzas AS p ON p.pizza_type_id = pt.pizza_type_id
WHERE
    p.price = (SELECT
            MAX(price)
        FROM
            pizzas);
```

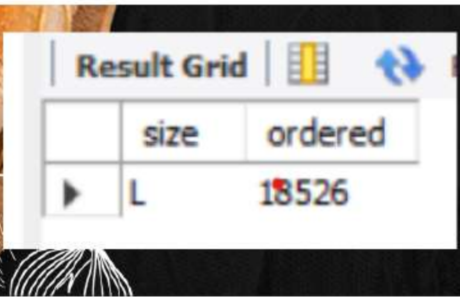


name
The Greek Pizza

4) IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

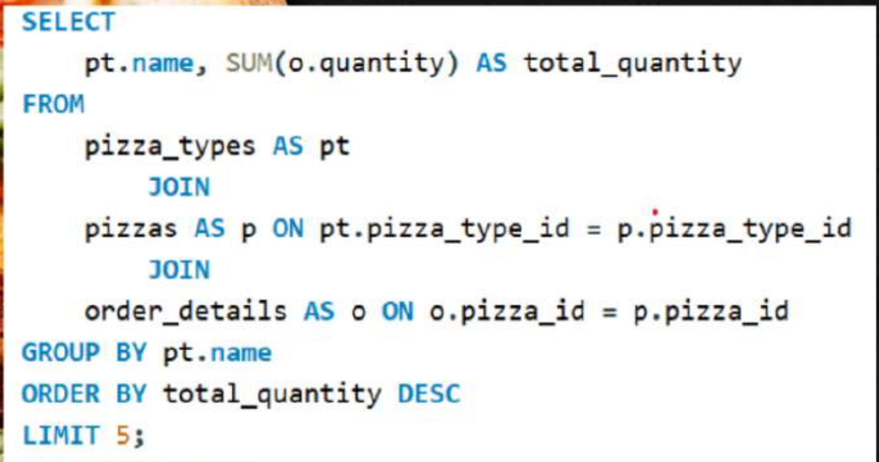


```
SELECT
    p.size, COUNT(o.order_details_id) AS ordered
FROM
    order_details AS o
        JOIN
    pizzas AS p ON o.pizza_id = p.pizza_id
GROUP BY p.size
ORDER BY ordered DESC
LIMIT 1;
```

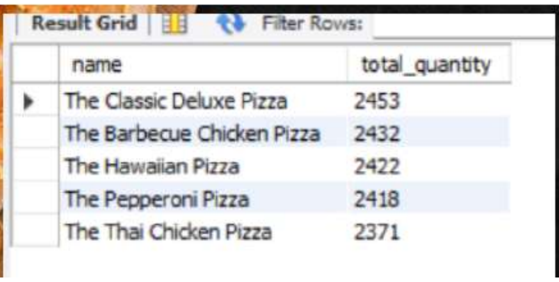


size	ordered
L	18526

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

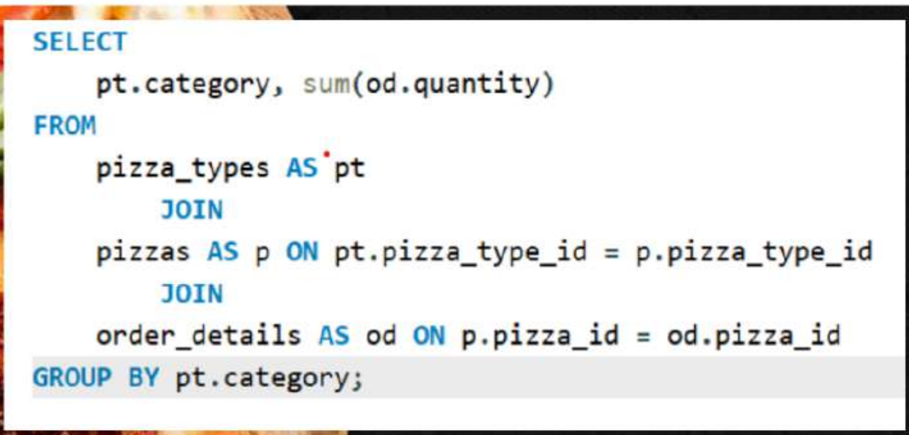


```sql
SELECT
    pt.name, SUM(o.quantity) AS total_quantity
FROM
    pizza_types AS pt
        JOIN
    pizzas AS p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details AS o ON o.pizza_id = p.pizza_id
GROUP BY pt.name
ORDER BY total_quantity DESC
LIMIT 5;
```

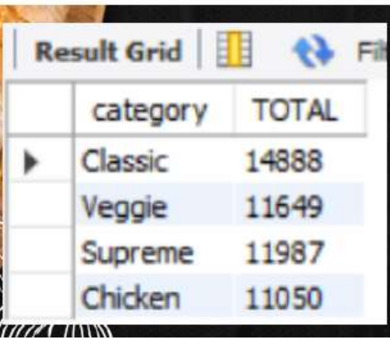


name	total_quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

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

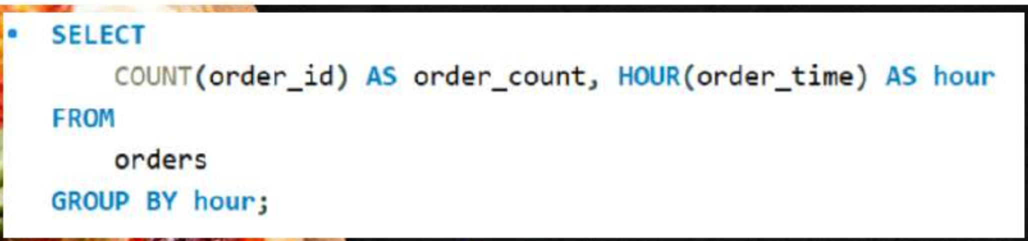


```
SELECT
    pt.category, sum(od.quantity)
FROM
    pizza_types AS pt
        JOIN
    pizzas AS p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details AS od ON p.pizza_id = od.pizza_id
GROUP BY pt.category;
```

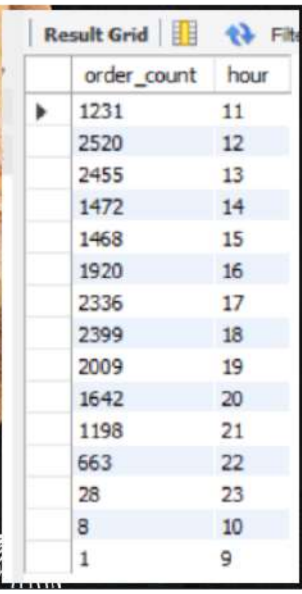


category	TOTAL
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

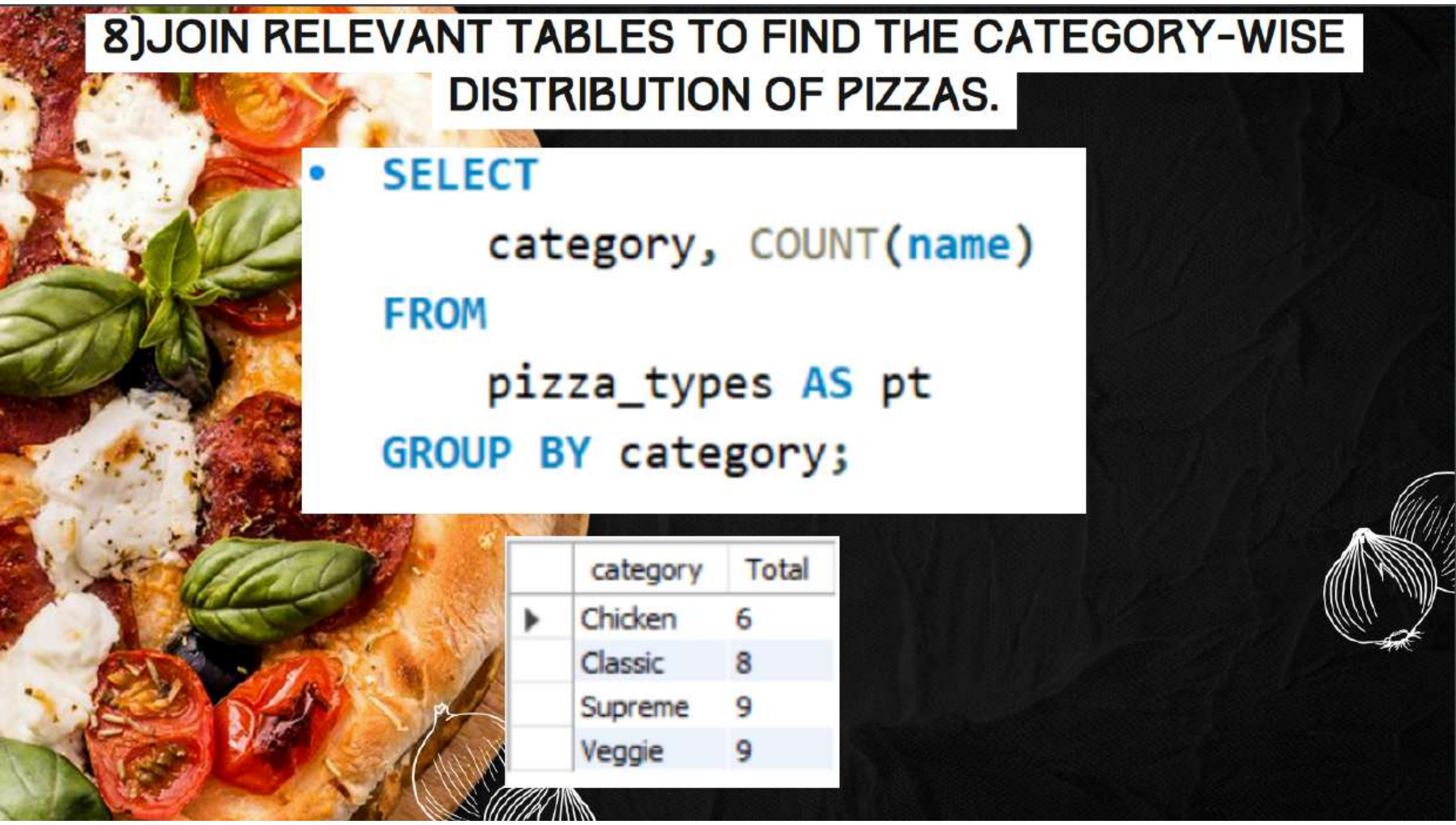
7) DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.



```
SELECT
    COUNT(order_id) AS order_count, HOUR(order_time) AS hour
FROM
    orders
GROUP BY hour;
```



order_count	hour
1231	11
2520	12
2455	13
1472	14
1468	15
1920	16
2336	17
2399	18
2009	19
1642	20
1198	21
663	22
28	23
8	10
1	9

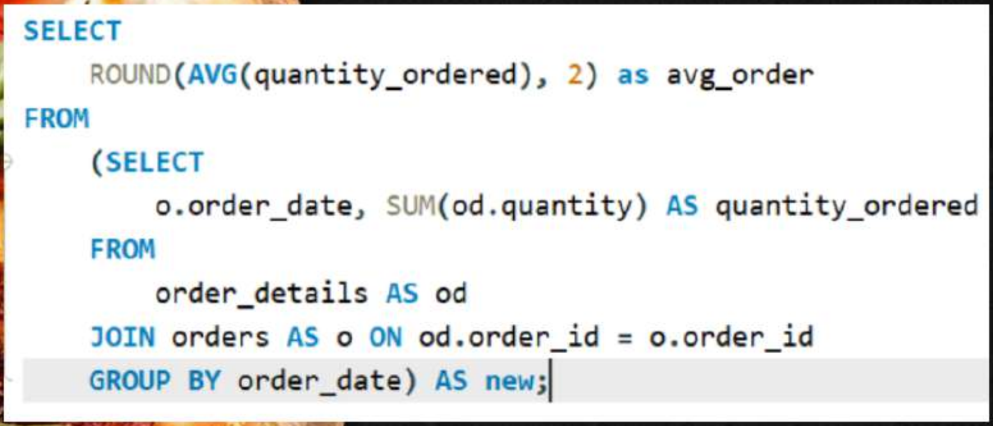


8)JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

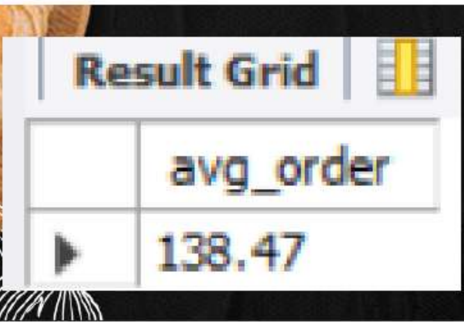
```
• SELECT
      category, COUNT(name)
  FROM
      pizza_types AS pt
  GROUP BY category;
```

	category	Total
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

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

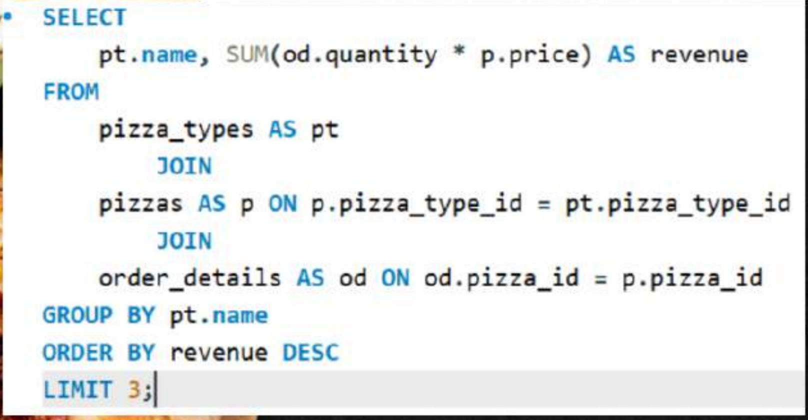


```
SELECT
    ROUND(AVG(quantity_ordered), 2) as avg_order
FROM
    (SELECT
        o.order_date, SUM(od.quantity) AS quantity_ordered
    FROM
        order_details AS od
    JOIN orders AS o ON od.order_id = o.order_id
    GROUP BY order_date) AS new;
```

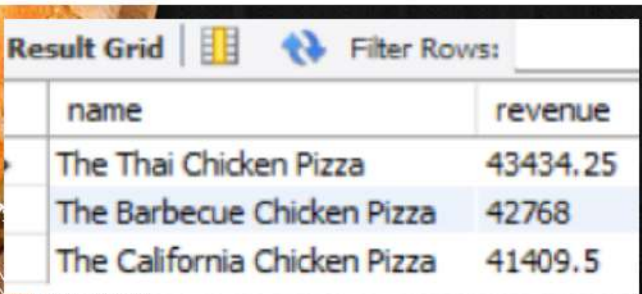


```
avg_order
138.47
```

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

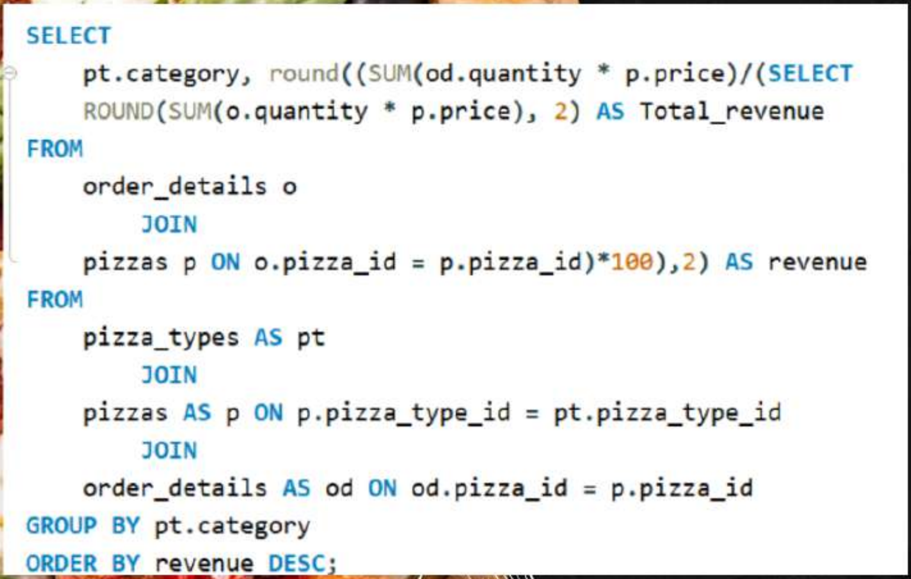


```
SELECT
    pt.name, SUM(od.quantity * p.price) AS revenue
FROM
    pizza_types AS pt
        JOIN
    pizzas AS p ON p.pizza_type_id = pt.pizza_type_id
        JOIN
    order_details AS od ON od.pizza_id = p.pizza_id
GROUP BY pt.name
ORDER BY revenue DESC
LIMIT 3;
```

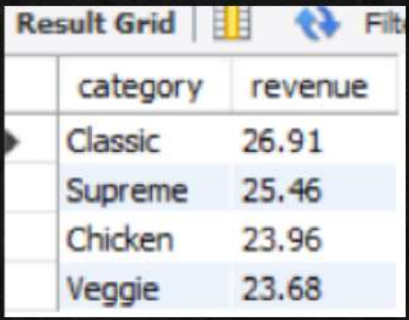


name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

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

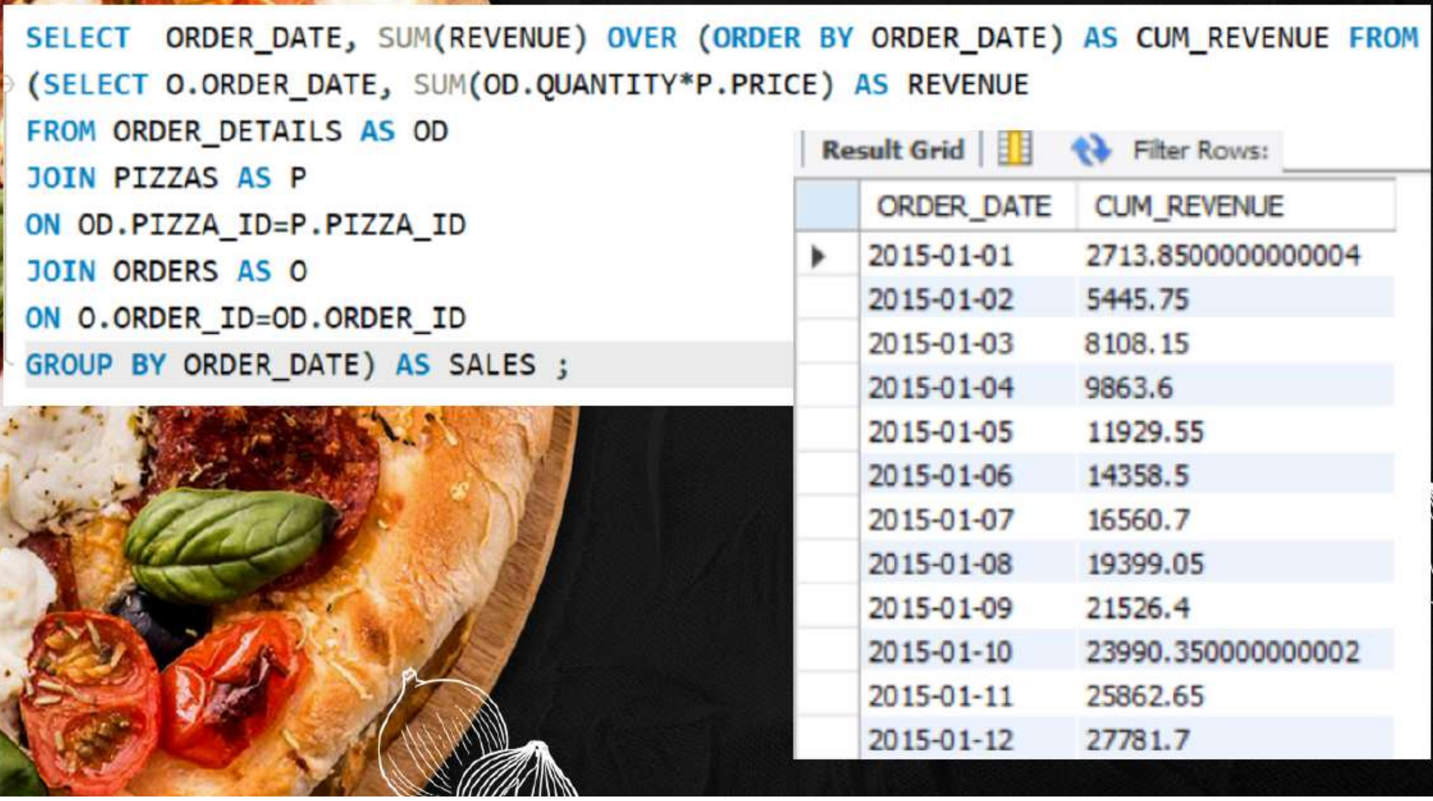


```
SELECT
    pt.category, round((SUM(od.quantity * p.price)/(SELECT
    ROUND(SUM(o.quantity * p.price), 2) AS Total_revenue
FROM
    order_details o
        JOIN
    pizzas p ON o.pizza_id = p.pizza_id)*100),2) AS revenue
FROM
    pizza_types AS pt
        JOIN
    pizzas AS p ON p.pizza_type_id = pt.pizza_type_id
        JOIN
    order_details AS od ON od.pizza_id = p.pizza_id
GROUP BY pt.category
ORDER BY revenue DESC;
```



category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

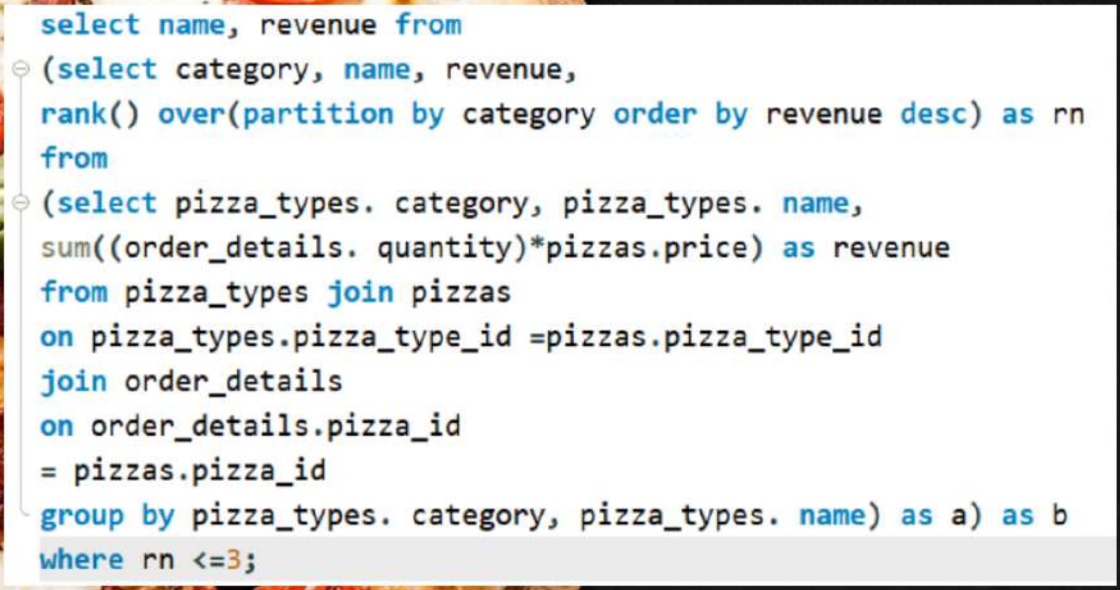
12) ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.



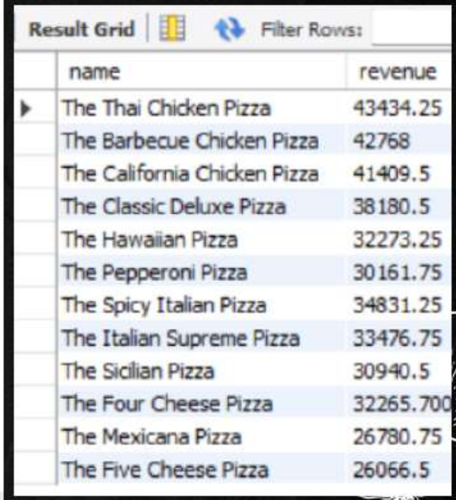
```
SELECT  ORDER_DATE, SUM(REVENUE) OVER (ORDER BY ORDER_DATE) AS CUM_REVENUE FROM
(SELECT O.ORDER_DATE, SUM(OD.QUANTITY*P.PRICE) AS REVENUE
FROM ORDER_DETAILS AS OD
JOIN PIZZAS AS P
ON OD.PIZZA_ID=P.PIZZA_ID
JOIN ORDERS AS O
ON O.ORDER_ID=OD.ORDER_ID
GROUP BY ORDER_DATE) AS SALES ;
```

ORDER_DATE	CUM_REVENUE
2015-01-01	2713.8500000000004
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

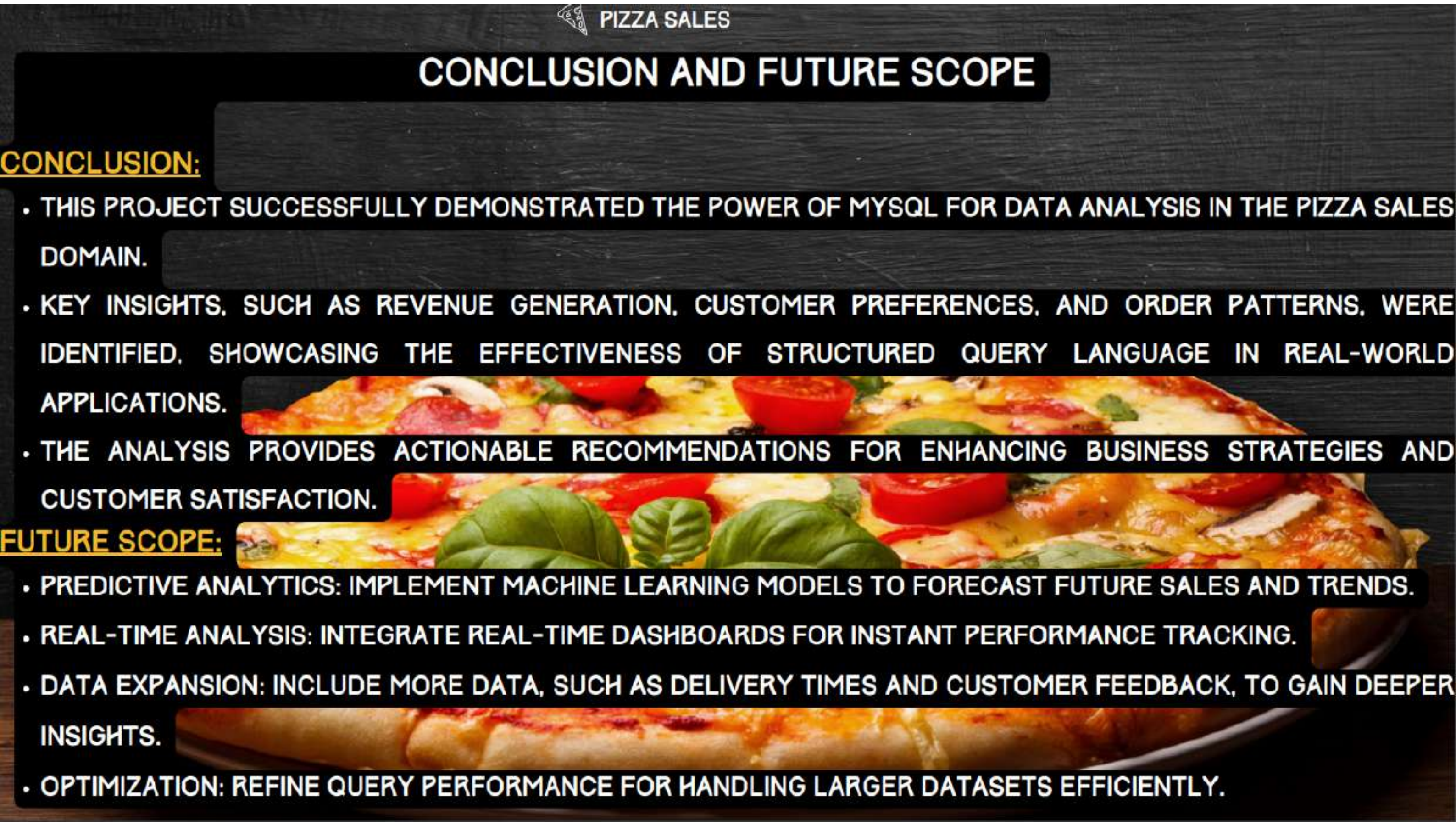
13) DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.



```sql
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((order_details. quantity)*pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id =pizzas.pizza_type_id
join order_details
on order_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
The Spicy Italian Pizza	34831.25
The Italian Supreme Pizza	33476.75
The Sicilian Pizza	30940.5
The Four Cheese Pizza	32265.700
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.5



PIZZA SALES

CONCLUSION AND FUTURE SCOPE

CONCLUSION:

- THIS PROJECT SUCCESSFULLY DEMONSTRATED THE POWER OF MYSQL FOR DATA ANALYSIS IN THE PIZZA SALES DOMAIN.
- KEY INSIGHTS, SUCH AS REVENUE GENERATION, CUSTOMER PREFERENCES, AND ORDER PATTERNS, WERE IDENTIFIED, SHOWCASING THE EFFECTIVENESS OF STRUCTURED QUERY LANGUAGE IN REAL-WORLD APPLICATIONS.
- THE ANALYSIS PROVIDES ACTIONABLE RECOMMENDATIONS FOR ENHANCING BUSINESS STRATEGIES AND CUSTOMER SATISFACTION.

FUTURE SCOPE:

- PREDICTIVE ANALYTICS: IMPLEMENT MACHINE LEARNING MODELS TO FORECAST FUTURE SALES AND TRENDS.
- REAL-TIME ANALYSIS: INTEGRATE REAL-TIME DASHBOARDS FOR INSTANT PERFORMANCE TRACKING.
- DATA EXPANSION: INCLUDE MORE DATA, SUCH AS DELIVERY TIMES AND CUSTOMER FEEDBACK, TO GAIN DEEPER INSIGHTS.
- OPTIMIZATION: REFINE QUERY PERFORMANCE FOR HANDLING LARGER DATASETS EFFICIENTLY.