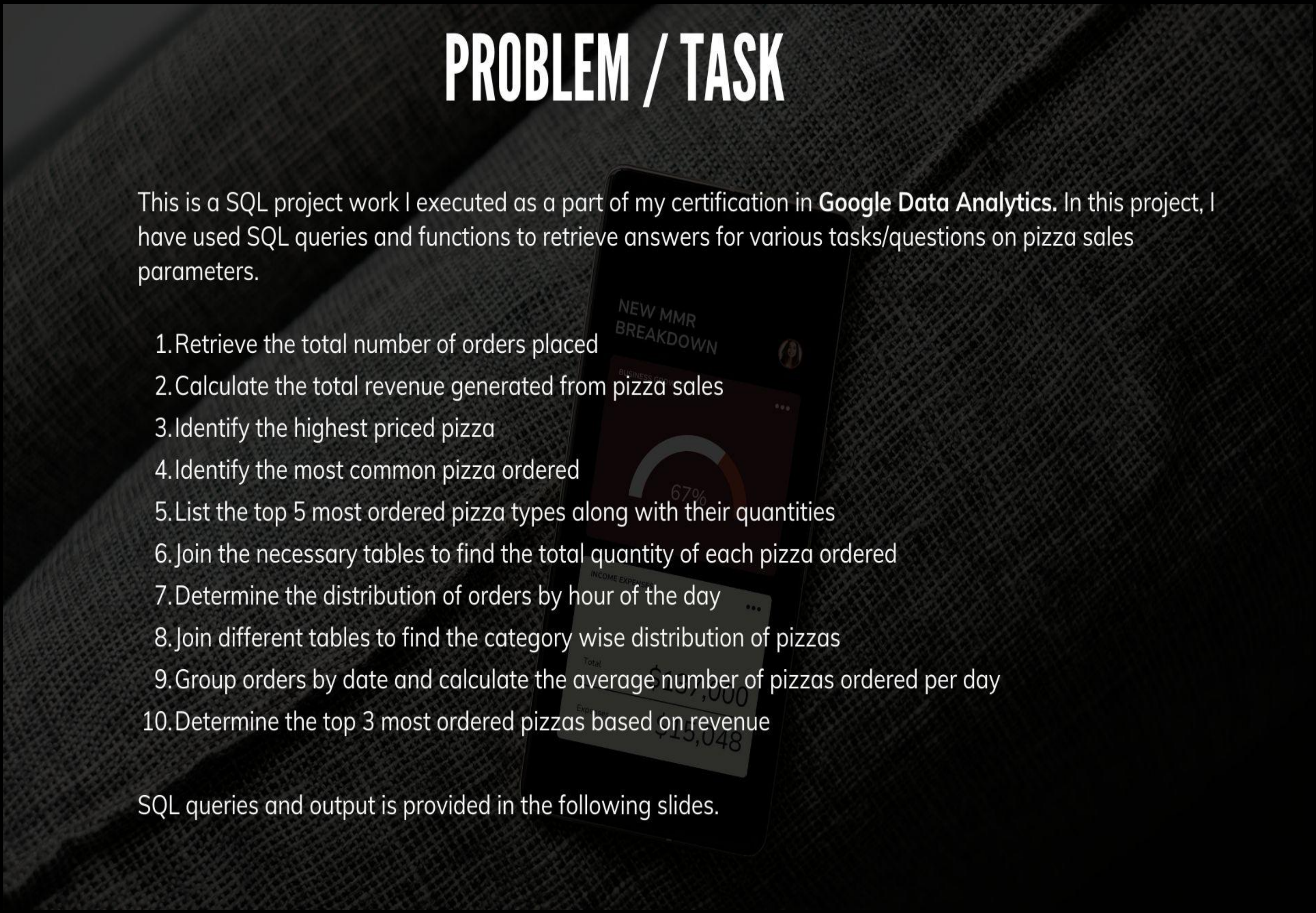
PIZZA SALES

SQL PROJECT

AS A PART OF GOOGLE DATA ANALYTICS CERTIFICATION

SANGEETH CV
28 JANUARY 2025



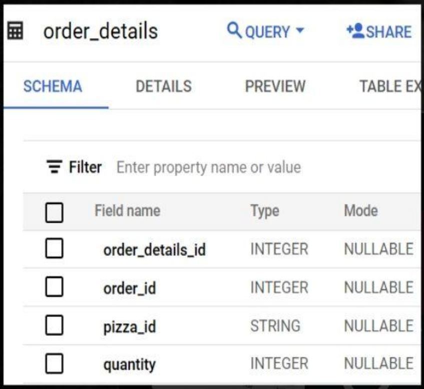
PROBLEM / TASK

This is a SQL project work I executed as a part of my certification in **Google Data Analytics.** In this project, I have used SQL queries and functions to retrieve answers for various tasks/questions on pizza sales parameters.

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 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 ordered
7. Determine the distribution of orders by hour of the day
8. Join different tables to find the category wise distribution of pizzas
9. Group orders by date and calculate the average number of pizzas ordered per day
10. Determine the top 3 most ordered pizzas based on revenue

SQL queries and output is provided in the following slides.

TABLES / SCHEMA

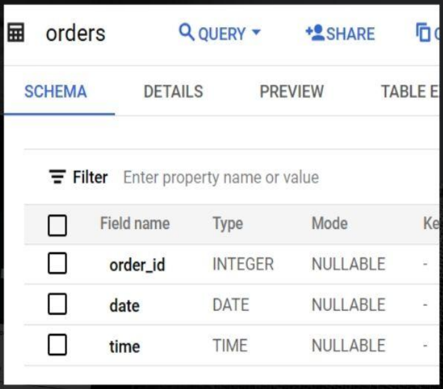


order_details — QUERY ▾ — SHARE

SCHEMA — DETAILS — PREVIEW — TABLE EX

Filter — Enter property name or value

□	Field name	Type	Mode
□	order_details_id	INTEGER	NULLABLE
□	order_id	INTEGER	NULLABLE
□	pizza_id	STRING	NULLABLE
□	quantity	INTEGER	NULLABLE

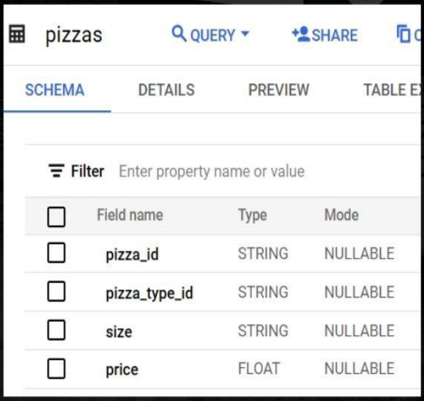


orders — QUERY ▾ — SHARE

SCHEMA — DETAILS — PREVIEW — TABLE E

Filter — Enter property name or value

□	Field name	Type	Mode	Ke
□	order_id	INTEGER	NULLABLE	-
□	date	DATE	NULLABLE	-
□	time	TIME	NULLABLE	-

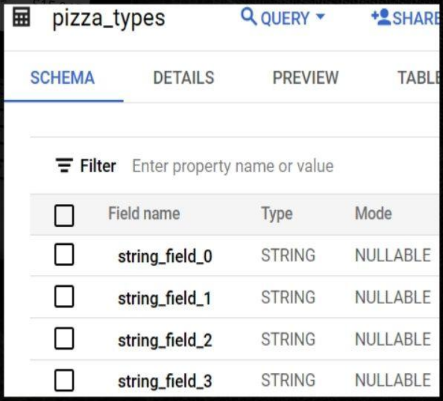


pizzas — QUERY ▾ — SHARE

SCHEMA — DETAILS — PREVIEW — TABLE EX

Filter — Enter property name or value

□	Field name	Type	Mode
□	pizza_id	STRING	NULLABLE
□	pizza_type_id	STRING	NULLABLE
□	size	STRING	NULLABLE
□	price	FLOAT	NULLABLE



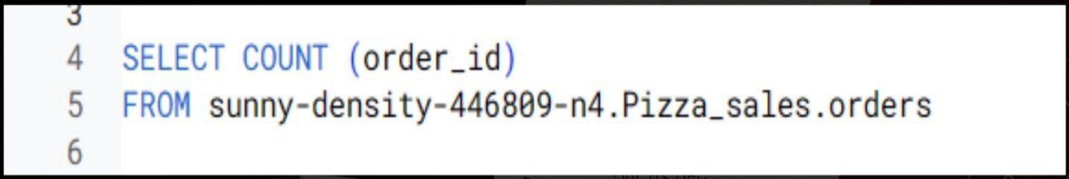
pizza_types — QUERY ▾ — SHARE

SCHEMA — DETAILS — PREVIEW — TABLE

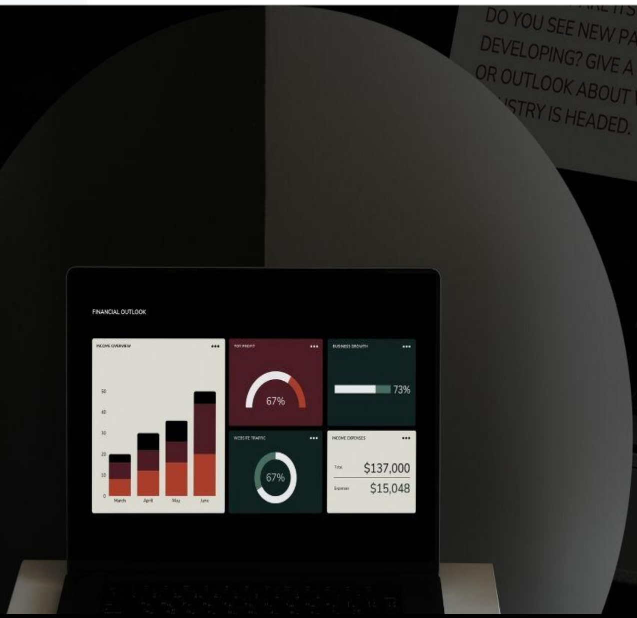
Filter — Enter property name or value

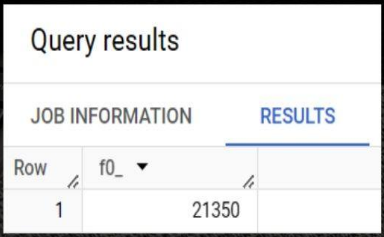
□	Field name	Type	Mode
□	string_field_0	STRING	NULLABLE
□	string_field_1	STRING	NULLABLE
□	string_field_2	STRING	NULLABLE
□	string_field_3	STRING	NULLABLE

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED



```sql
SELECT COUNT (order_id)
FROM sunny-density-446809-n4.Pizza_sales.orders
```



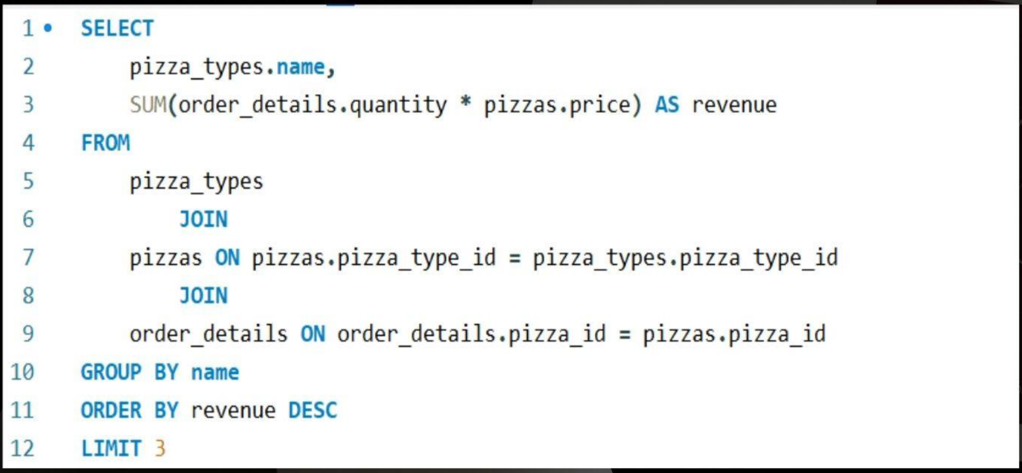


Query results

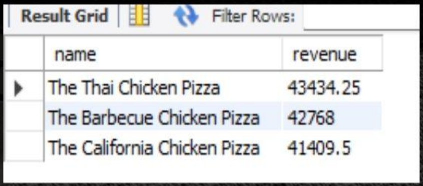
JOB INFORMATION — RESULTS

Row	f0_ ▾
1	21350

DETERMINE THE TOP 3 MOST ORDERED PIZZAS BASED ON REVENUE

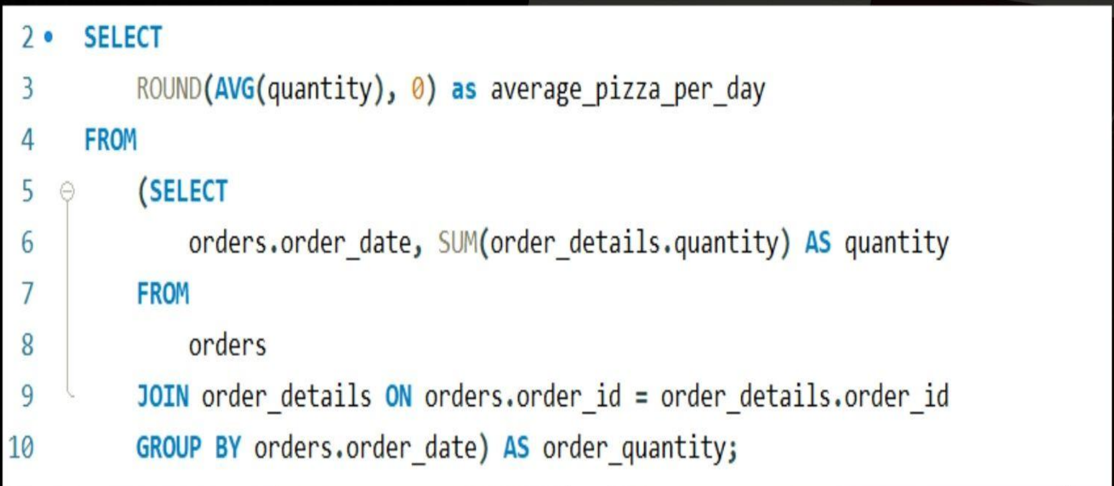


```sql
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY name
ORDER BY revenue DESC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

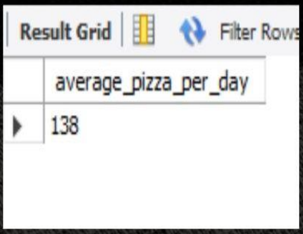


name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

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

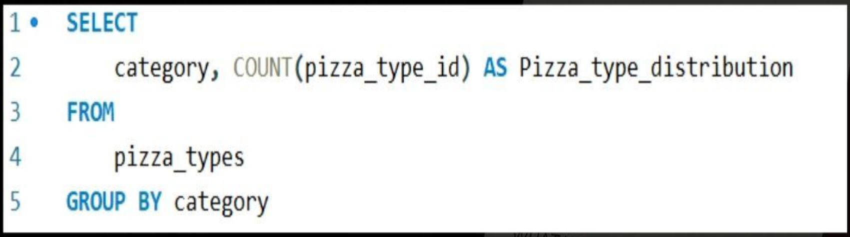


```sql
SELECT
    ROUND(AVG(quantity), 0) as average_pizza_per_day
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

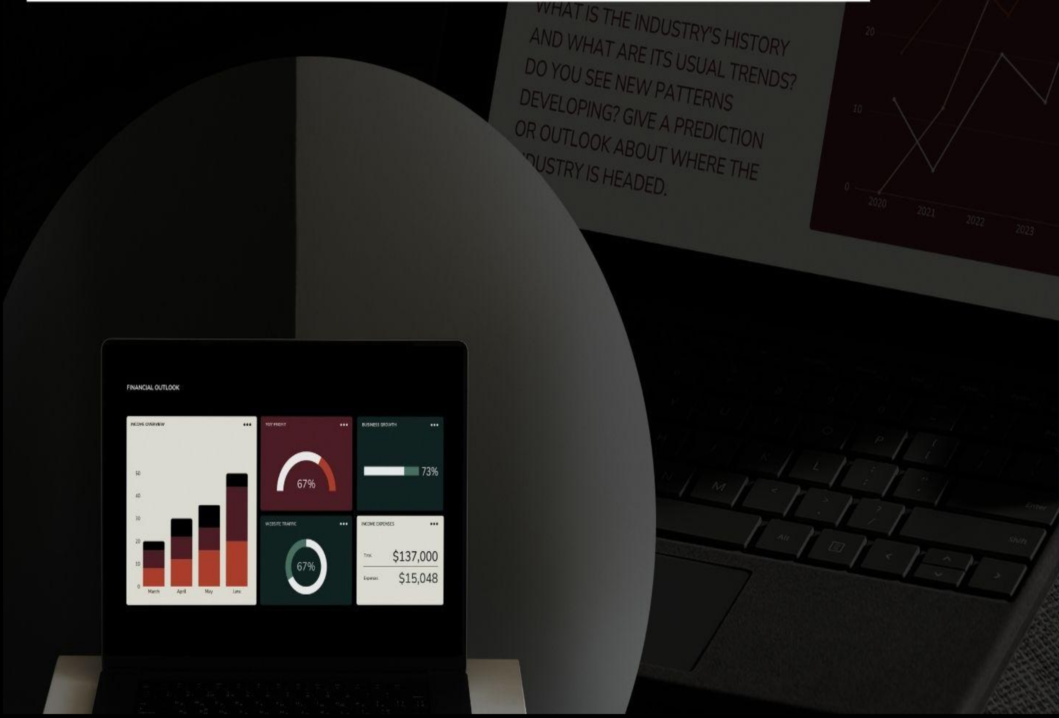


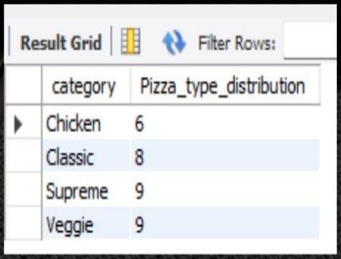
average_pizza_per_day
138

JOIN DIFFERENT TABLES TO FIND THE CATEGORY WISE DISTRIBUTION OF PIZZAS



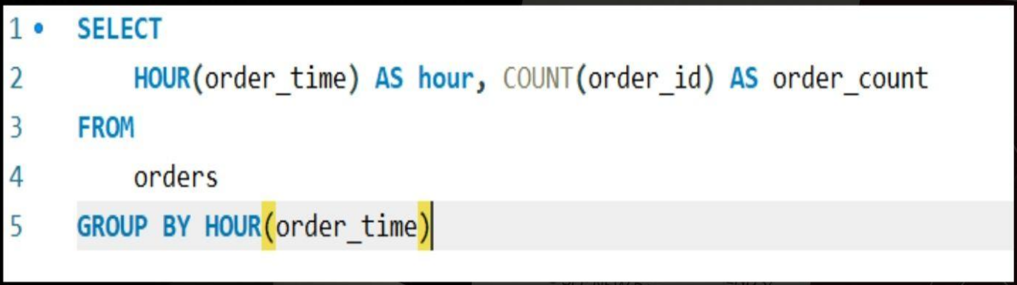
```
SELECT
    category, COUNT(pizza_type_id) AS Pizza_type_distribution
FROM
    pizza_types
GROUP BY category
```


WHAT IS THE INDUSTRY'S HISTORY AND WHAT ARE ITS USUAL TRENDS? DO YOU SEE NEW PATTERNS DEVELOPING? GIVE A PREDICTION OR OUTLOOK ABOUT WHERE THE DUSTRY IS HEADED.
FINANCIAL OUTLOOK
67%
73%
$137,000
$15,048

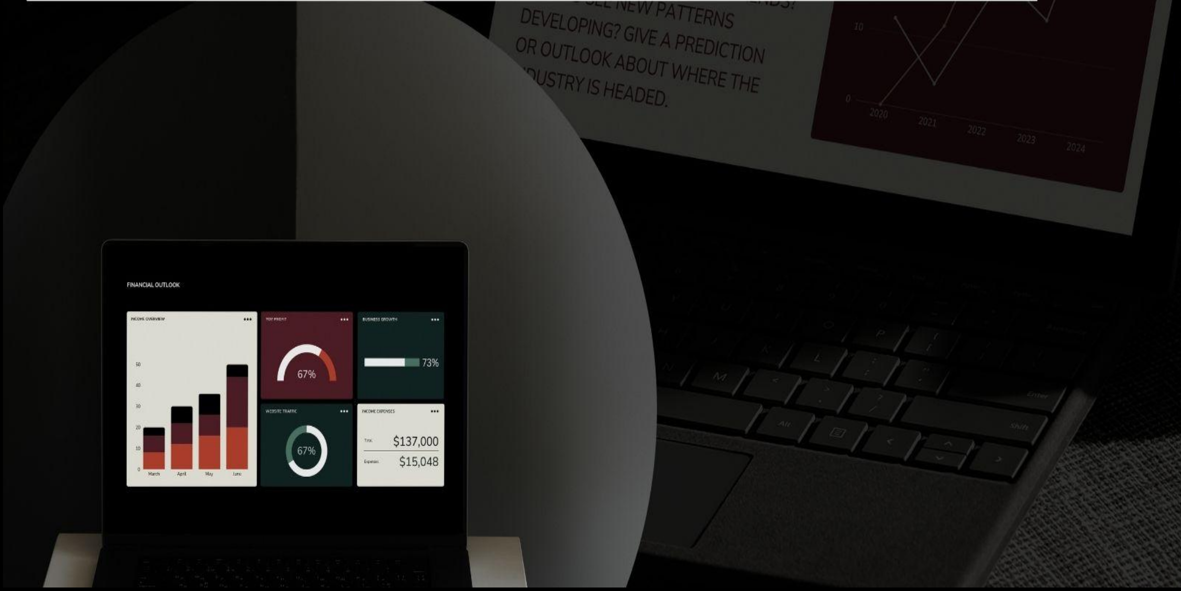


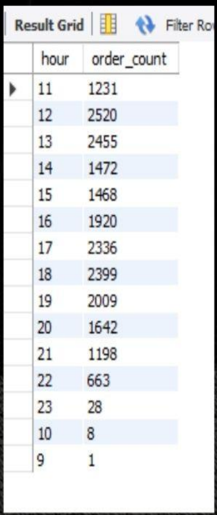
category	Pizza_type_distribution
Chicken	6
Classic	8
Supreme	9
Veggie	9

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY



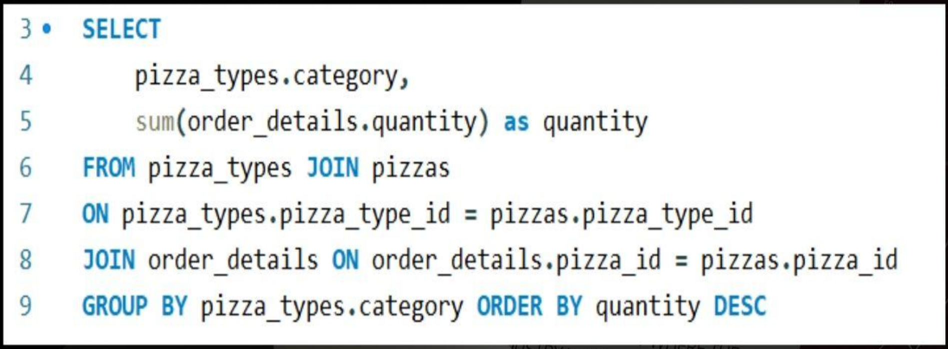
```
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time)
```


DEVELOPING? GIVE A PREDICTION OR OUTLOOK ABOUT WHERE THE DUSTRY IS HEADED.
FINANCIAL OUTLOOK
67%
73%
$137,000
$15,048

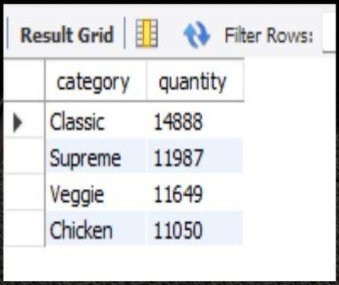


hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

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

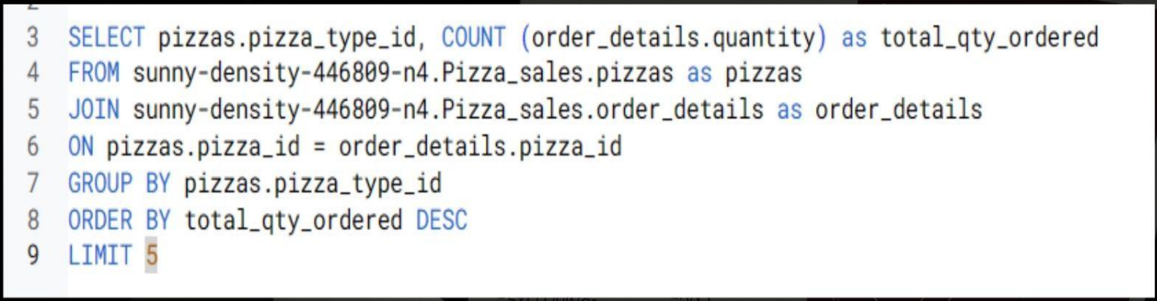


```
SELECT
    pizza_types.category,
    sum(order_details.quantity) as quantity
FROM pizza_types JOIN pizzas
ON pizza_types.pizza_type_id = pizzas.pizza_type_id
JOIN order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category ORDER BY quantity DESC
```

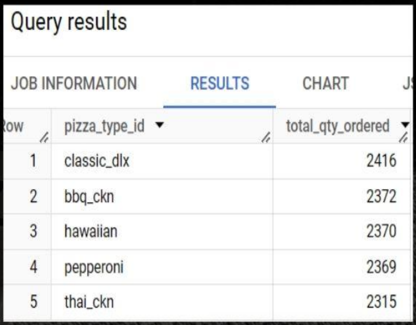


category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

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

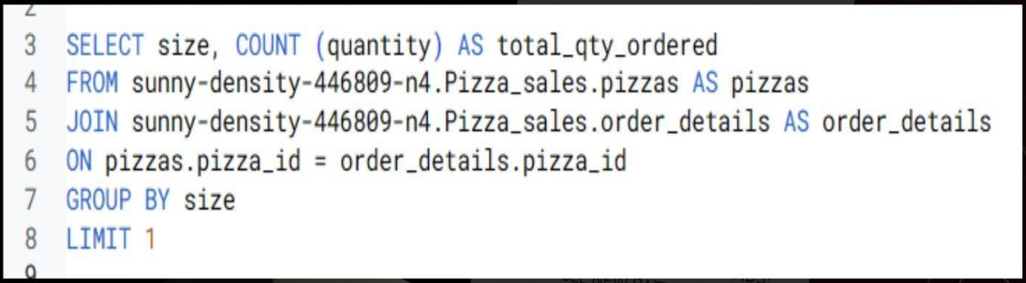


```
SELECT pizzas.pizza_type_id, COUNT (order_details.quantity) as total_qty_ordered
FROM sunny-density-446809-n4.Pizza_sales.pizzas as pizzas
JOIN sunny-density-446809-n4.Pizza_sales.order_details as order_details
ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.pizza_type_id
ORDER BY total_qty_ordered DESC
LIMIT 5
```

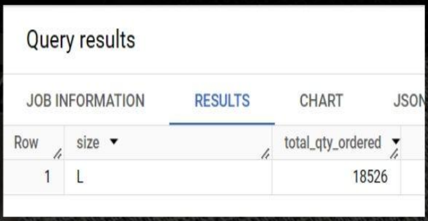


ow	pizza_type_id	total_qty_ordered
1	classic_dlx	2416
2	bbq_ckn	2372
3	hawaiian	2370
4	pepperoni	2369
5	thai_ckn	2315

IDENTIFY THE MOST COMMON PIZZA ORDERED



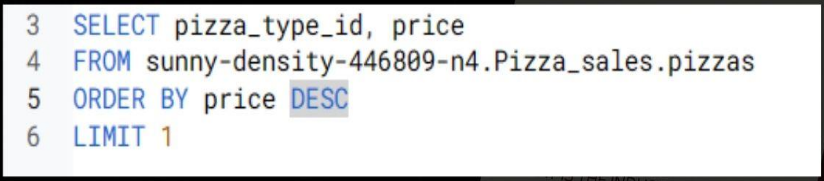
```
SELECT size, COUNT (quantity) AS total_qty_ordered
FROM sunny-density-446809-n4.Pizza_sales.pizzas AS pizzas
JOIN sunny-density-446809-n4.Pizza_sales.order_details AS order_details
ON pizzas.pizza_id = order_details.pizza_id
GROUP BY size
LIMIT 1
```



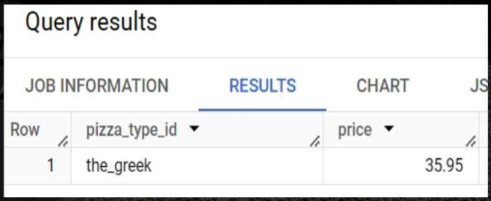
Query results

Row	size	total_qty_ordered
1	L	18526

IDENTIFY THE HIGHEST PRICED PIZZA



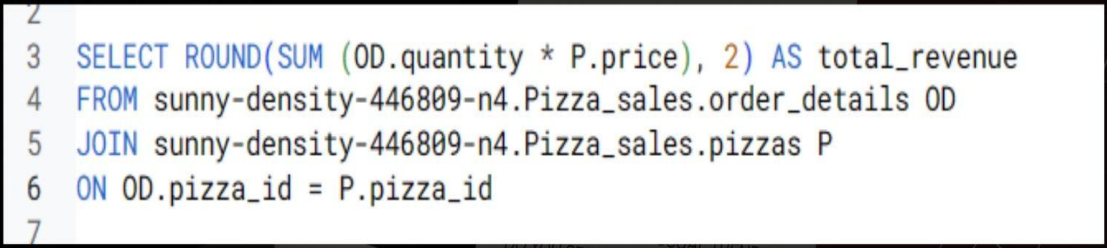
```
SELECT pizza_type_id, price
FROM sunny-density-446809-n4.Pizza_sales.pizzas
ORDER BY price DESC
LIMIT 1
```



Query results

Row	pizza_type_id	price
1	the_greek	35.95

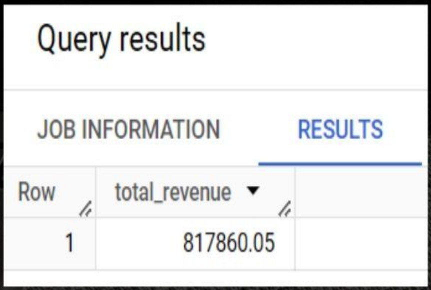
CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES



```sql

SELECT ROUND(SUM (OD.quantity * P.price), 2) AS total_revenue
FROM sunny-density-446809-n4.Pizza_sales.order_details OD
JOIN sunny-density-446809-n4.Pizza_sales.pizzas P
ON OD.pizza_id = P.pizza_id

```



```
Row   total_revenue
1     817860.05
```

THANK YOU