Pizzahut Sales Analysis using SQL.

Project by Mohd Siddiq .

Hello! My Name is Mohd Siddiq.

In this project using MySQL I have solved some questions related to Pizzahut using sql queries.

Objective :

- To understand sales performance and trends.
- To identify the most popular and profitable items.
- To analyze ordering behaviour over different times and dates.

Table Overview :

order_details :

- order_details_id
- order_id
- pizza_id
- quantity

orders :

- order_id int NOT NULL
- order_date date NOT NULL
- order_time time NOT NULL
- PRIMARY KEY(order_id)

pizza_types :

- pizza_type_id text
- name text
- category text
- ingredients text

pizzas :

- pizza_id text
- pizza_type_id text
- size text
- price double DEFAULT NULL

QUESTIONS :

Basic:

- Retrieve the total number of orders placed.
- Calculate the total revenue generated from pizza sales.
- Identify the highest-priced pizza.
- Identify the most common pizza size ordered.
- List the top 5 most ordered pizza types along with their quantities.

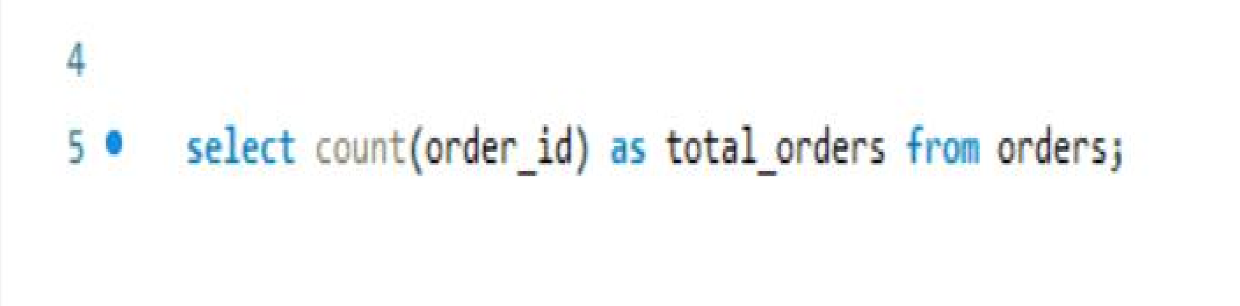
Intermediate:

- Join the necessary tables to find the total quantity of each pizza category ordered.
- Determine the distribution of orders by hour of the day.
- Join relevant tables to find the category-wise distribution of pizzas.
- Group the orders by date and calculate the average number of pizzas ordered per day.
- Determine the top 3 most ordered pizza types based on revenue.

Advanced:

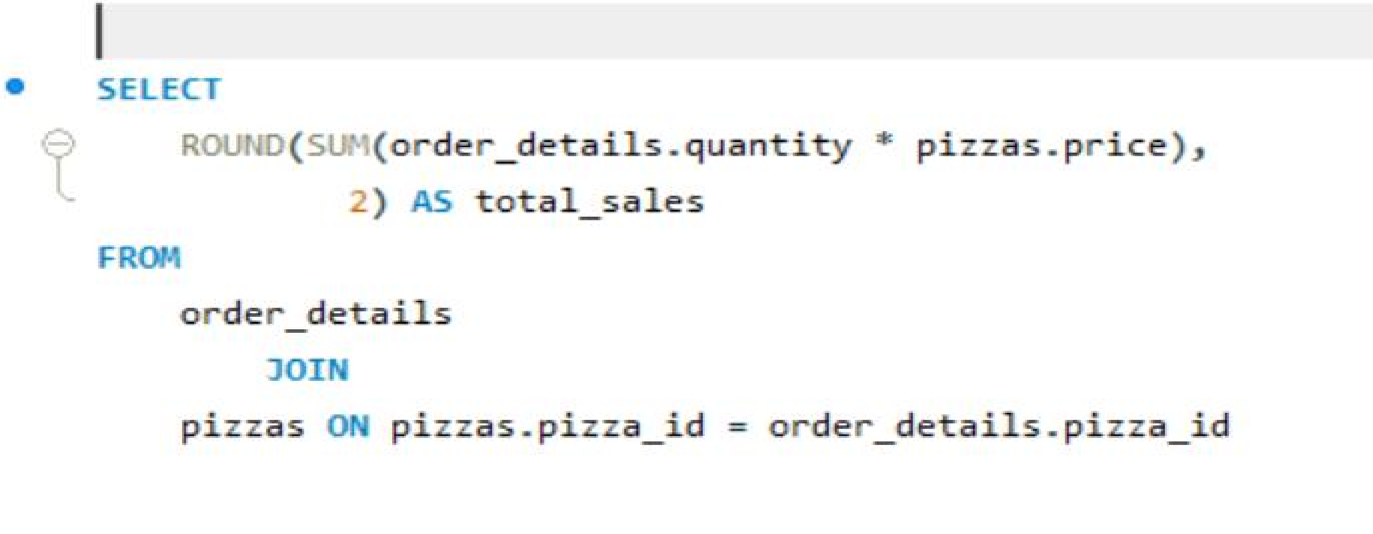
- Calculate the percentage contribution of each pizza type to total revenue.
- Analyze the cumulative revenue generated over time.
- Determine the top 3 most ordered pizza types based on revenue for each pizza category.

Retrieve the total number of orders placed.



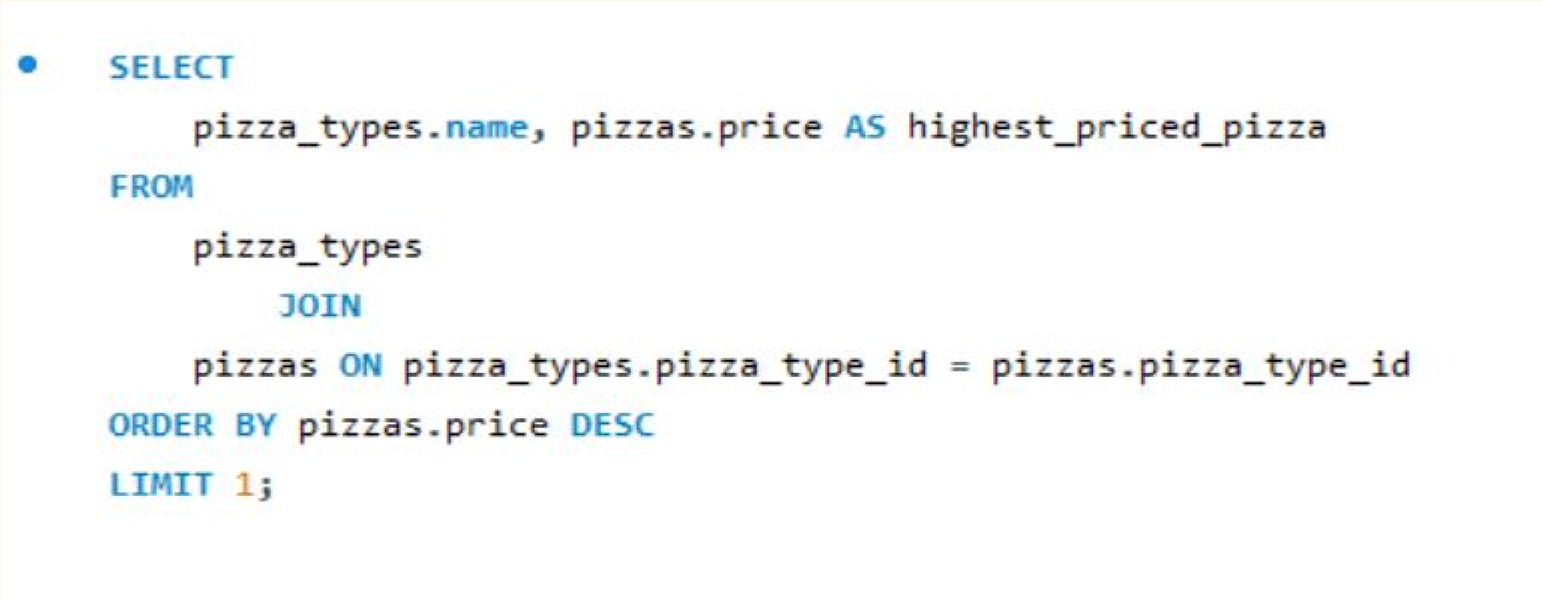
```sql
select count(order_id) as total_orders from orders;
```

Calculate the total revenue generated from pizza sales.



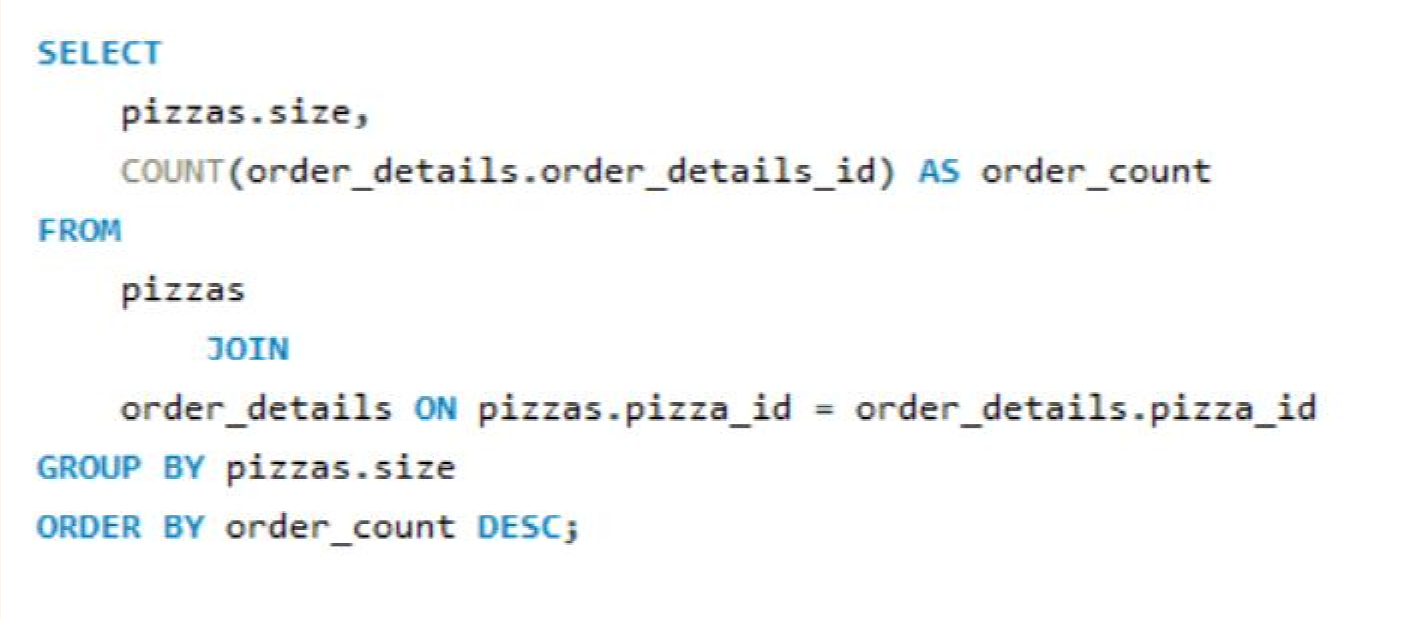
```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
```

identify the highest priced pizzas.



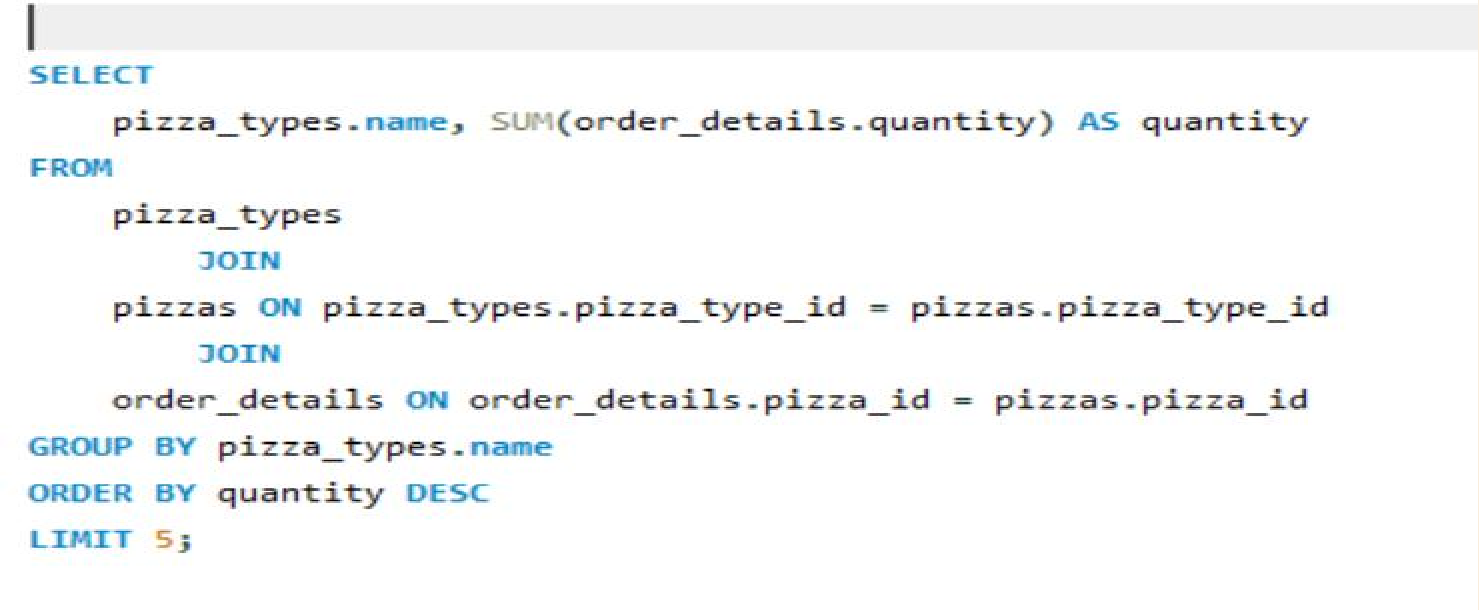
```
SELECT
    pizza_types.name, pizzas.price AS highest_priced_pizza
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

identify the most common pizza side ordered



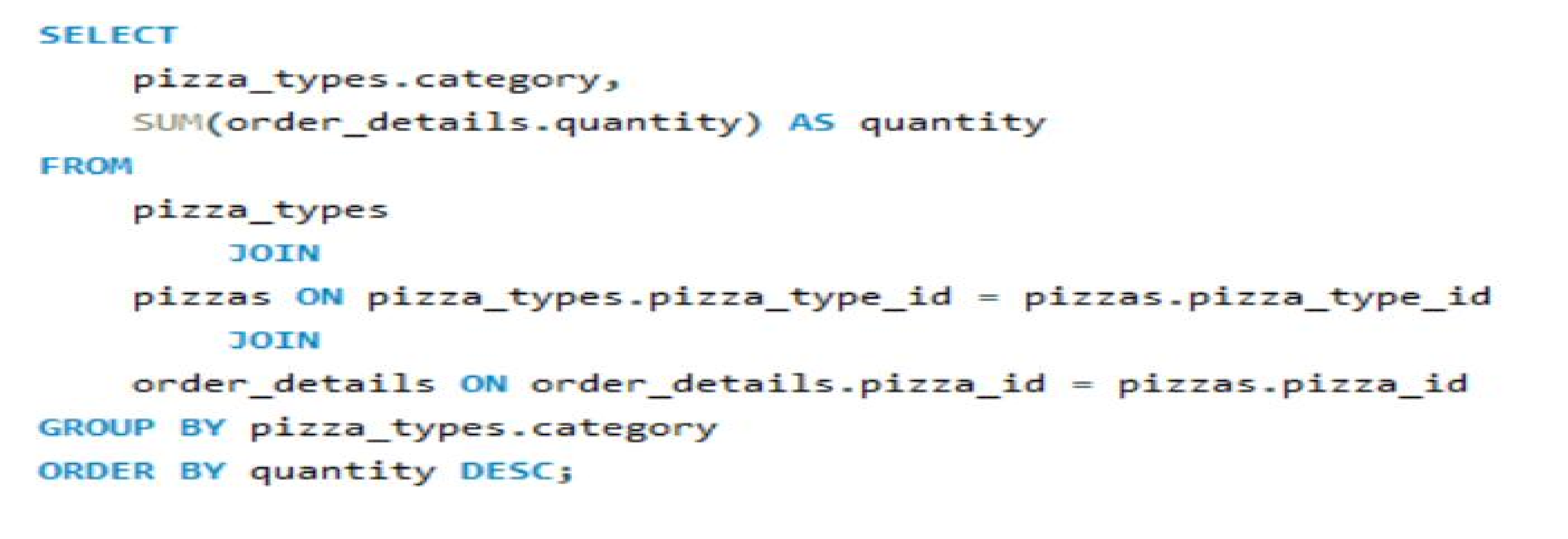
```
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```

list the most ordered types along with their quantities.



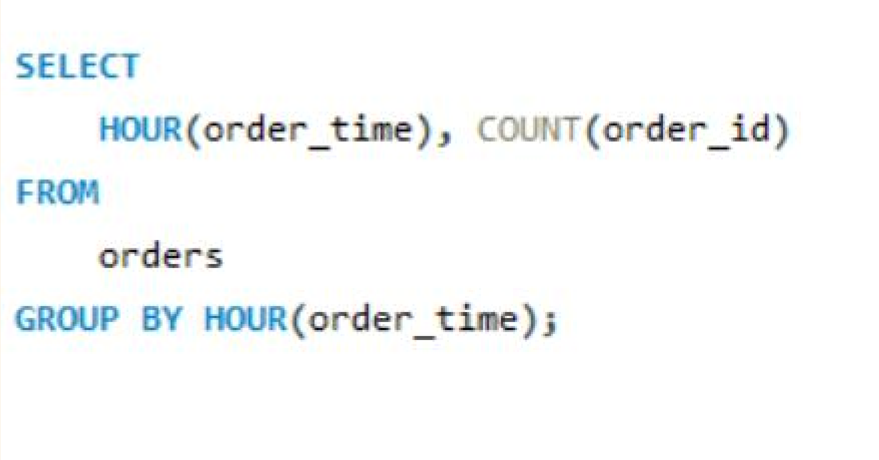
```sql
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT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

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



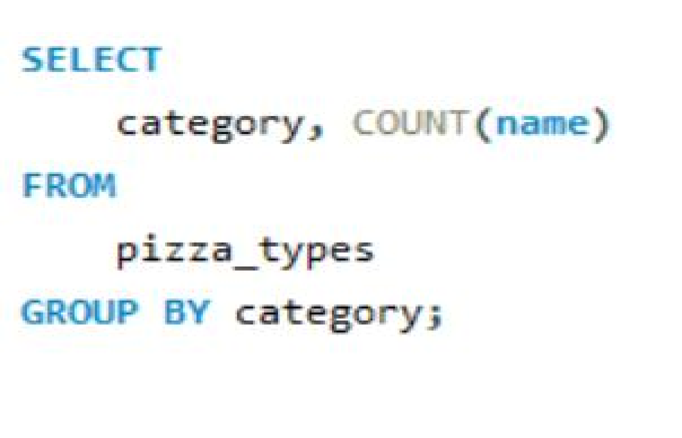
```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity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

determine the distribution of orders by hour of the day.



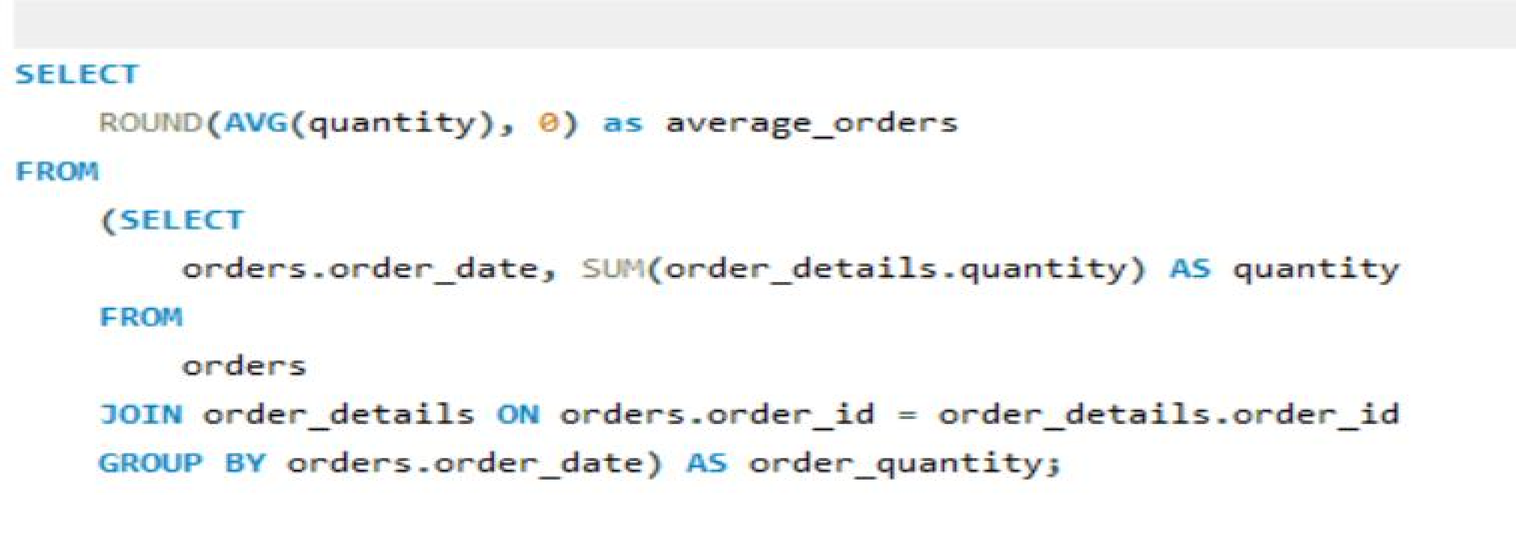
```
SELECT
    HOUR(order_time), COUNT(order_id)
FROM
    orders
GROUP BY HOUR(order_time);
```

join relevant tables ti find the category-wise distribution of pizzas.



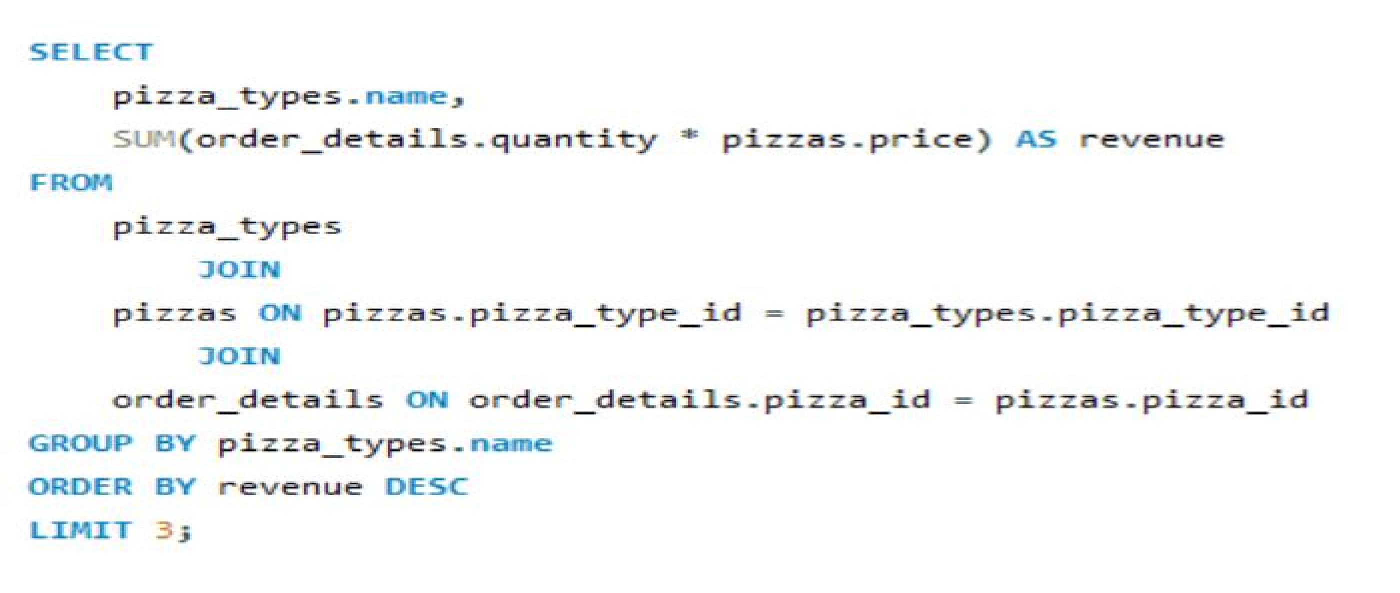
```
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```

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



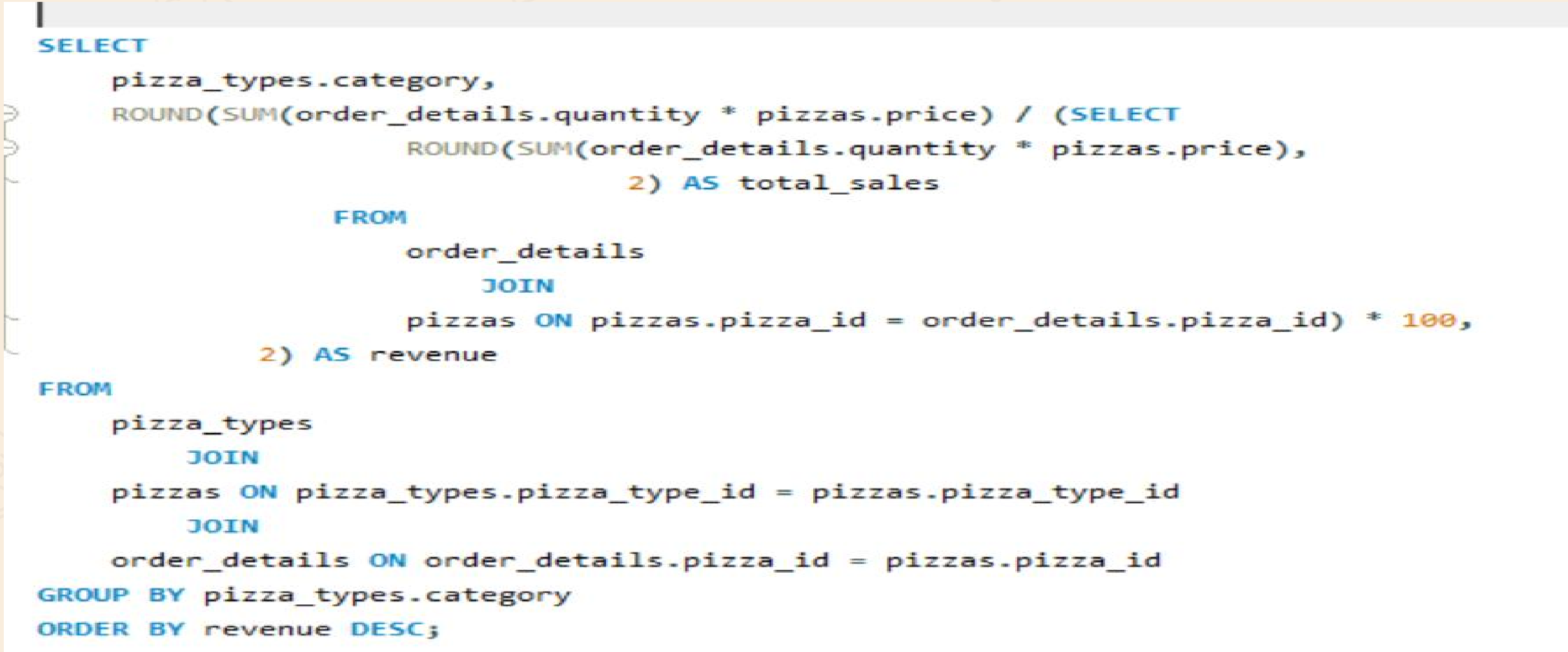
```
SELECT
    ROUND(AVG(quantity), 0) as average_orders
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

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



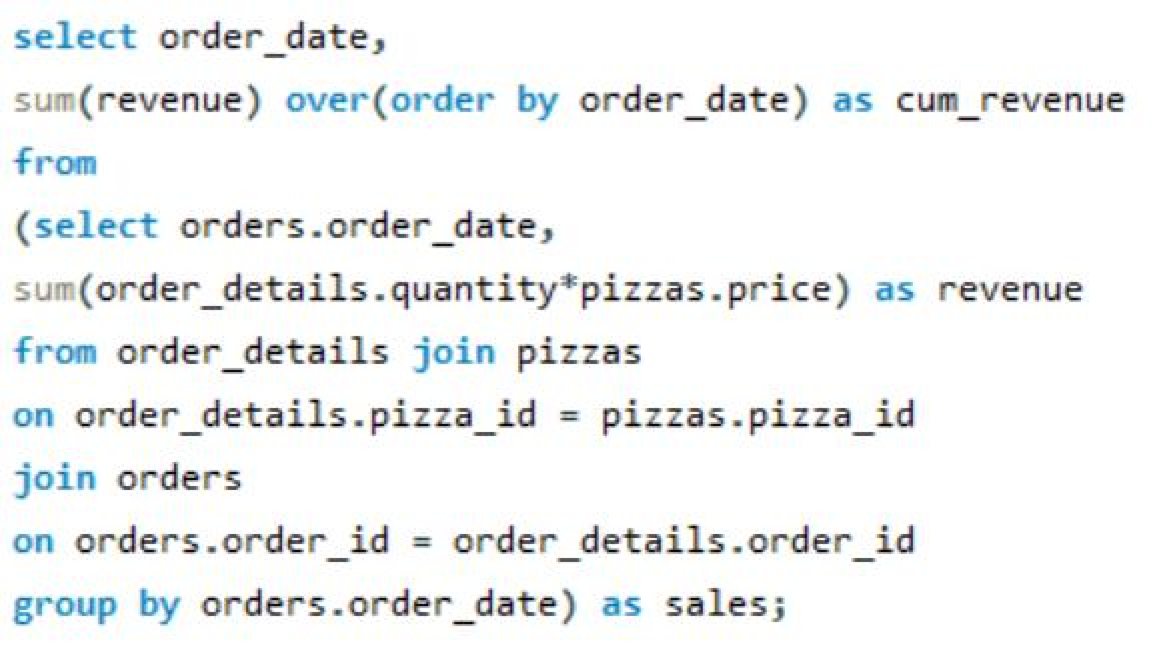
```
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue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

contribution the percentage of each pizza type to total revenue.



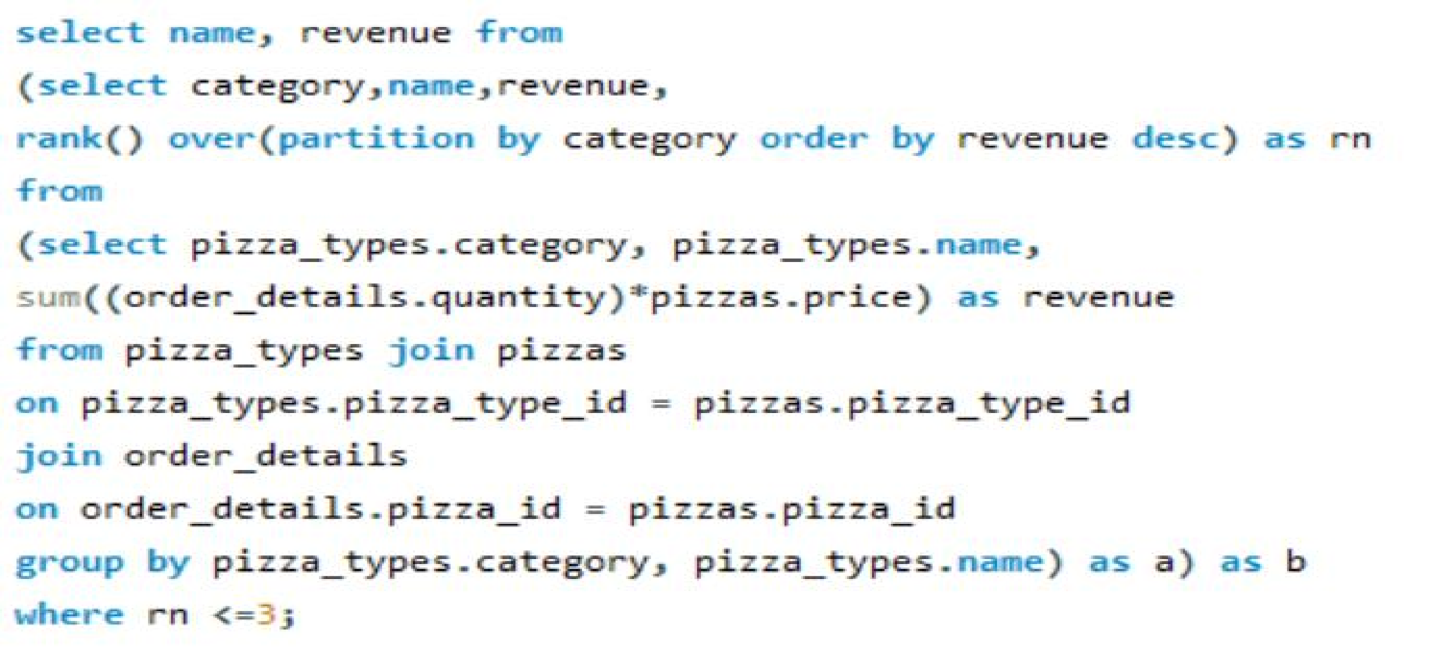
```
SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
                ROUND(SUM(order_details.quantity * pizzas.price),
                            2) AS total_sales
                FROM
                    order_details
                        JOIN
                    pizzas ON pizzas.pizza_id = order_details.pizza_id) * 100,
            2) 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
ORDER BY revenue DESC;
```

Analyze the cumulative revenue generated over time.



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

Determine the top 3 most ordered pizza types base on revneue for each pizza category



```
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;
```

CONCLUSION :

Project Creation:

Approach: This project involved analyzing pizza sales data using SQL to extract meaningful insights and provide business recommendations.

Key Insights:

- **High Demand:** Significant volume of orders and revenue.
- **Popular Items:** Large-sized pizzas and certain types that are frequently ordered.
- **Order Patterns:** Peak ordering times identified, helping in resource planning.
- **Revenue Drivers:** Key pizza types contributing most to the revenue

THANKS!

Do you have any questions?

https://github.com/Siddiqmohd620

www.linkedin.com/in/mohd-siddiq-445044299

✉ siddiqmohd620@gmail.com

☎ +91 7995431724