Pizza Sales Analysis Using SQL

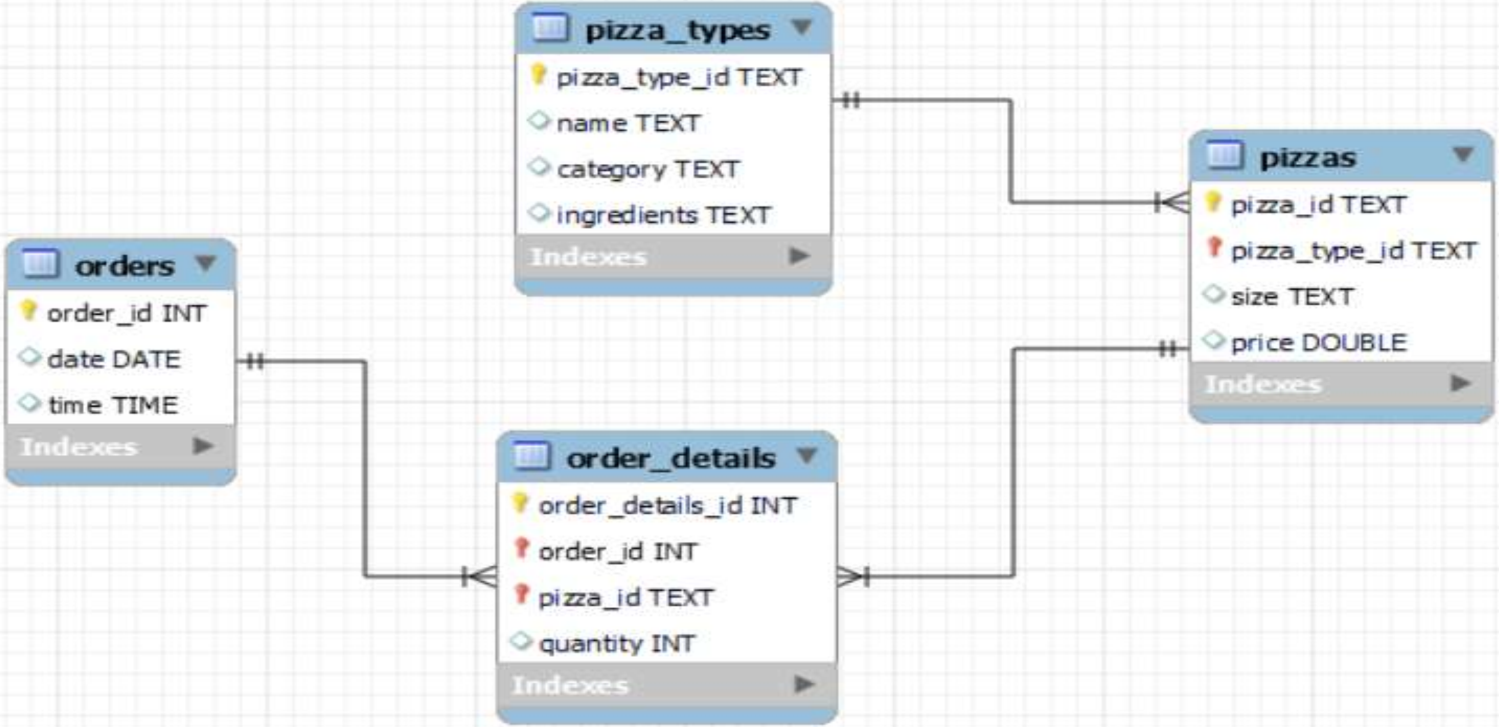
Mohit Yadav

Overview

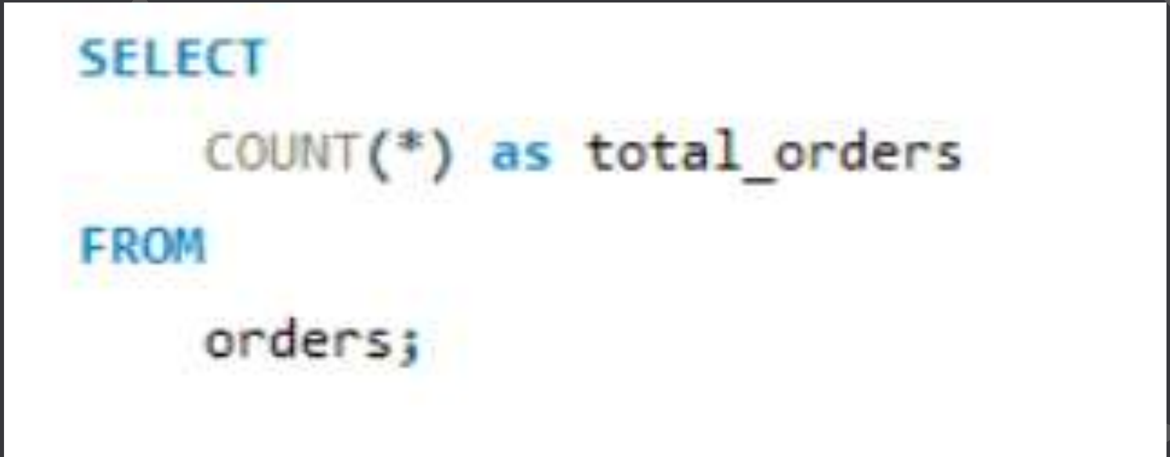
1. Imported CSV data into My SQL RDBMS.
2. Designed ERD Diagram.
3. Performed detailed analysis using advanced SQL Qeries.

Note : Queries separated by a dividing line are two different solution for the same question.

Designed ERD Diagram

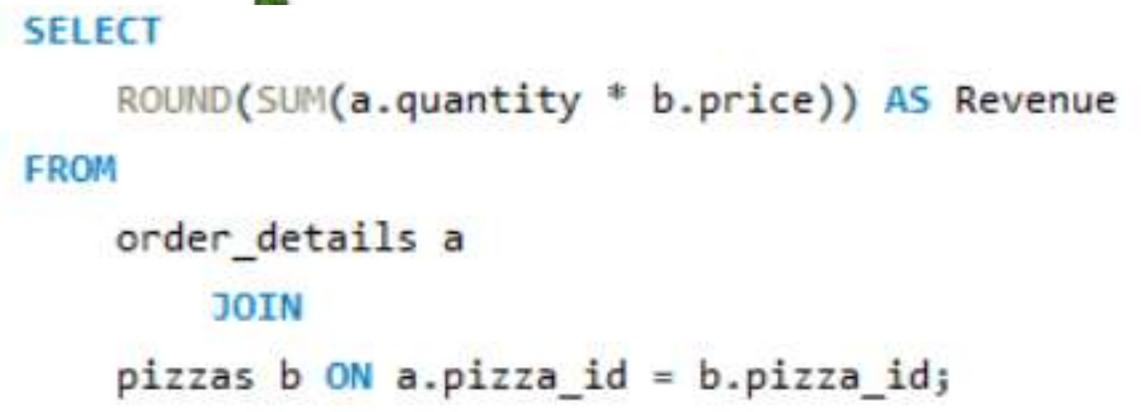


Total number of orders placed.



```
SELECT
    COUNT(*) as total_orders
FROM
    orders;
```

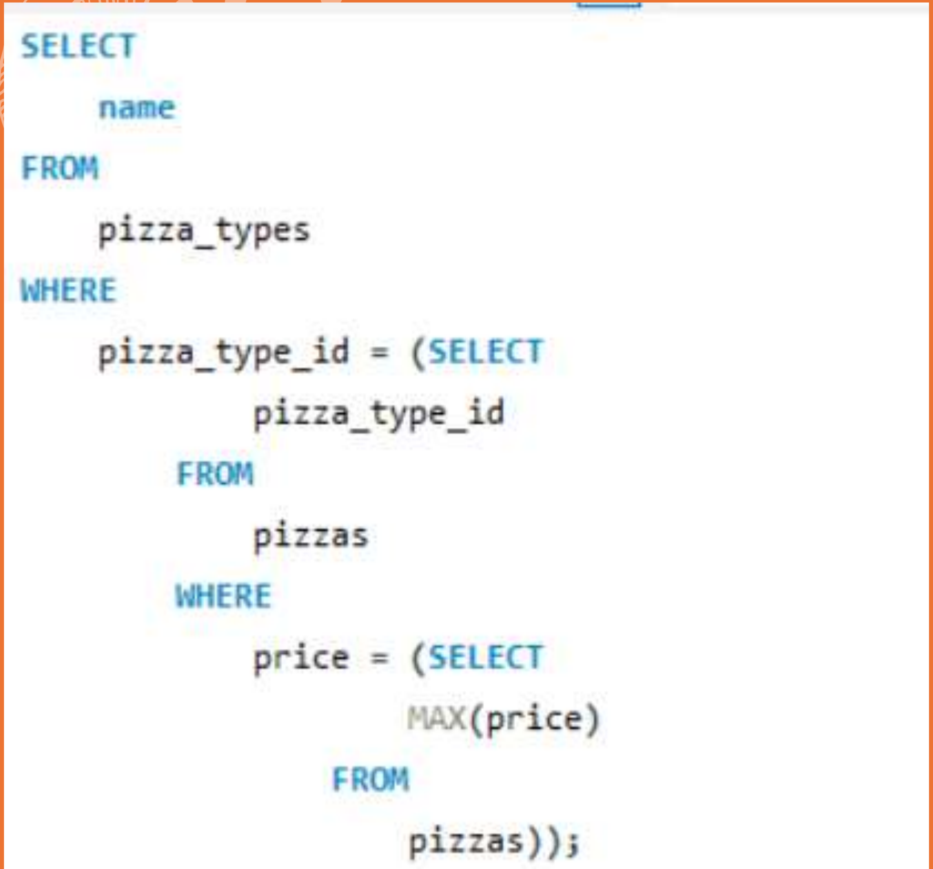
Total revenue generated from pizza sales.



```
SELECT
    ROUND(SUM(a.quantity * b.price)) AS Revenue
FROM
    order_details a
        JOIN
    pizzas b ON a.pizza_id = b.pizza_id;
```

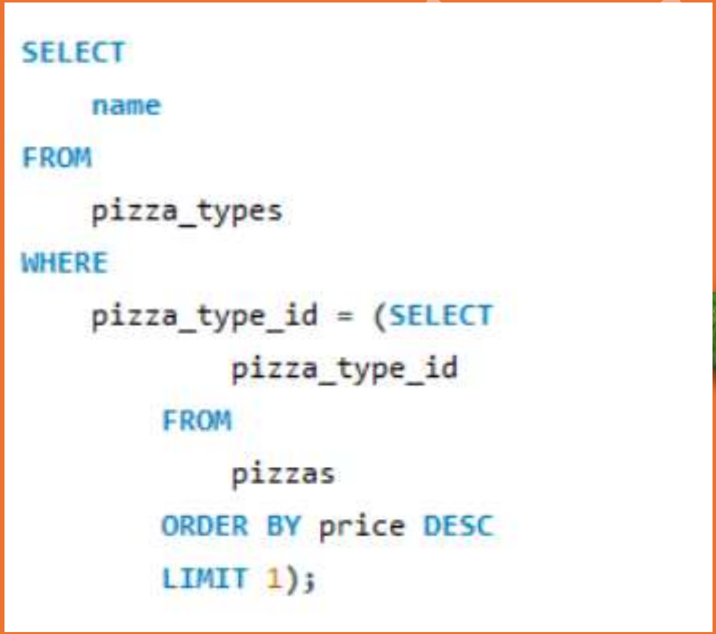
Identify the highest-priced pizza.

Method 1



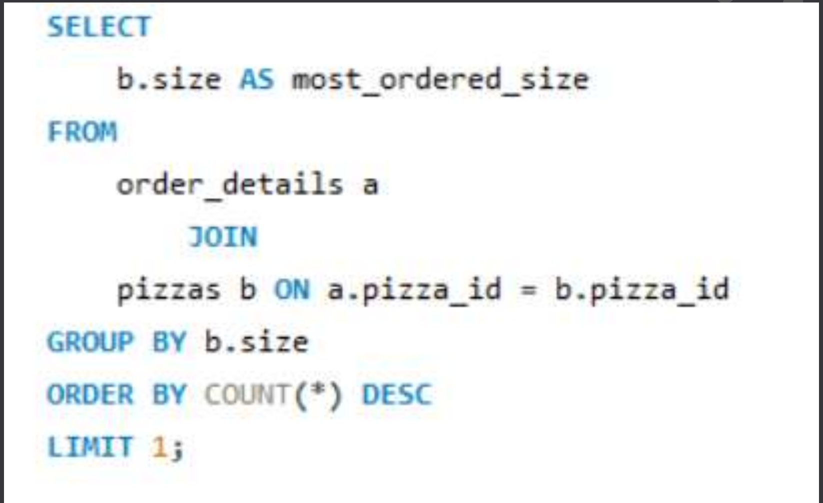
```
SELECT
    name
FROM
    pizza_types
WHERE
    pizza_type_id = (SELECT
            pizza_type_id
        FROM
            pizzas
        WHERE
            price = (SELECT
                    MAX(price)
                FROM
                    pizzas));
```

Method 2

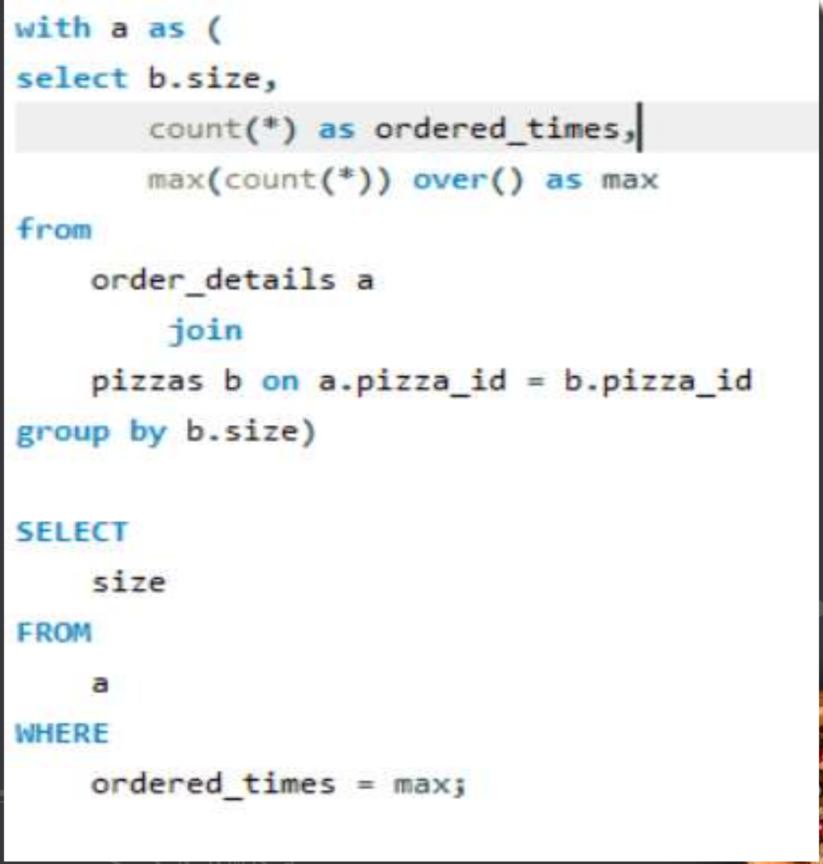


```
SELECT
    name
FROM
    pizza_types
WHERE
    pizza_type_id = (SELECT
            pizza_type_id
        FROM
            pizzas
        ORDER BY price DESC
        LIMIT 1);
```

Identify the most common pizza size ordered.

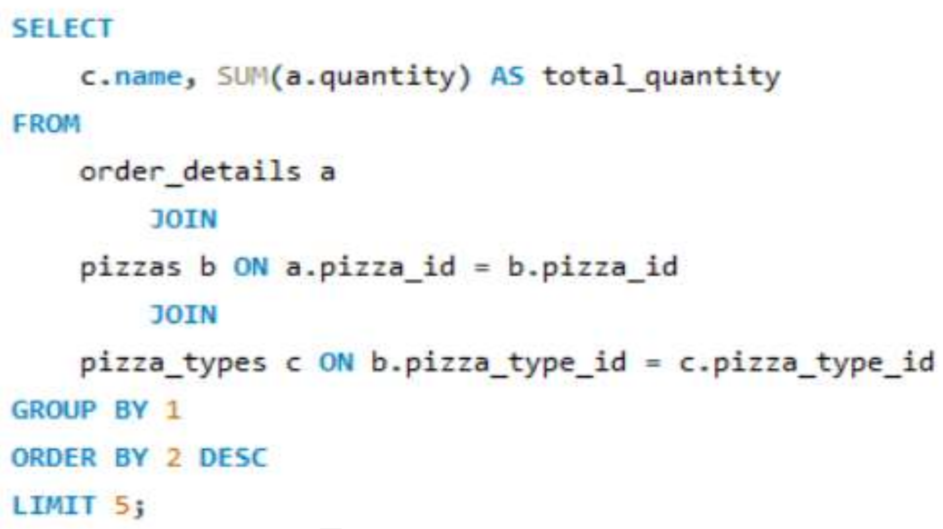


```
SELECT
    b.size AS most_ordered_size
FROM
    order_details a
        JOIN
    pizzas b ON a.pizza_id = b.pizza_id
GROUP BY b.size
ORDER BY COUNT(*) DESC
LIMIT 1;
```

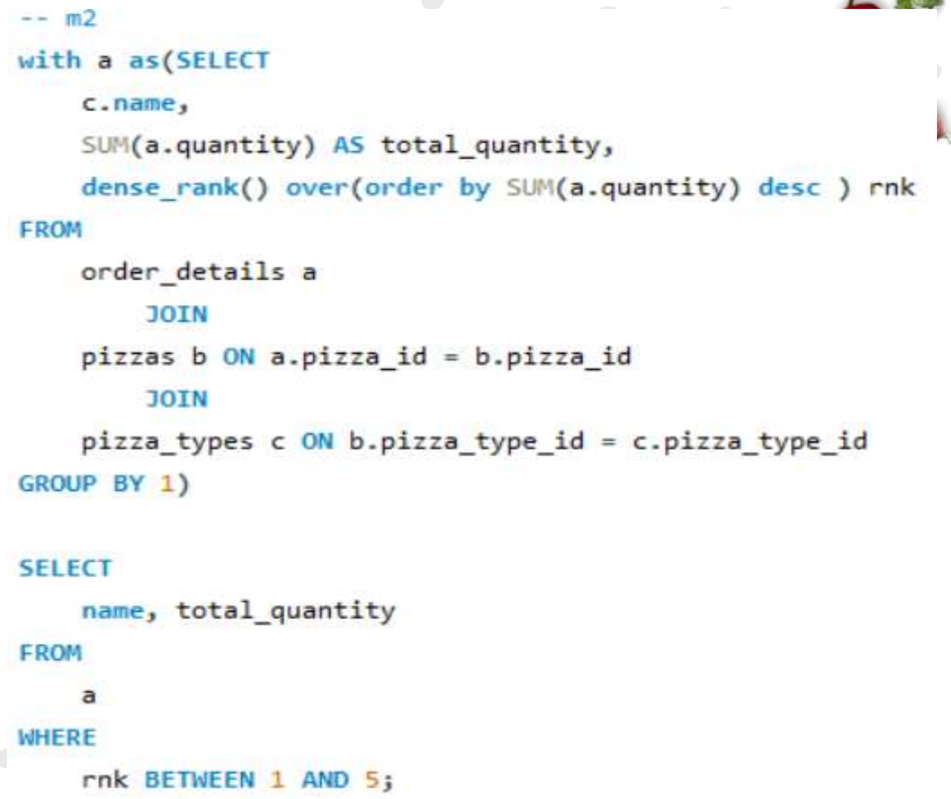


```
with a as (
select b.size,
       count(*) as ordered_times,
       max(count(*)) over() as max
from
    order_details a
        join
    pizzas b on a.pizza_id = b.pizza_id
group by b.size)

SELECT
    size
FROM
    a
WHERE
    ordered_times = max;
```

List the top 5 most ordered pizza types along with their quantities.

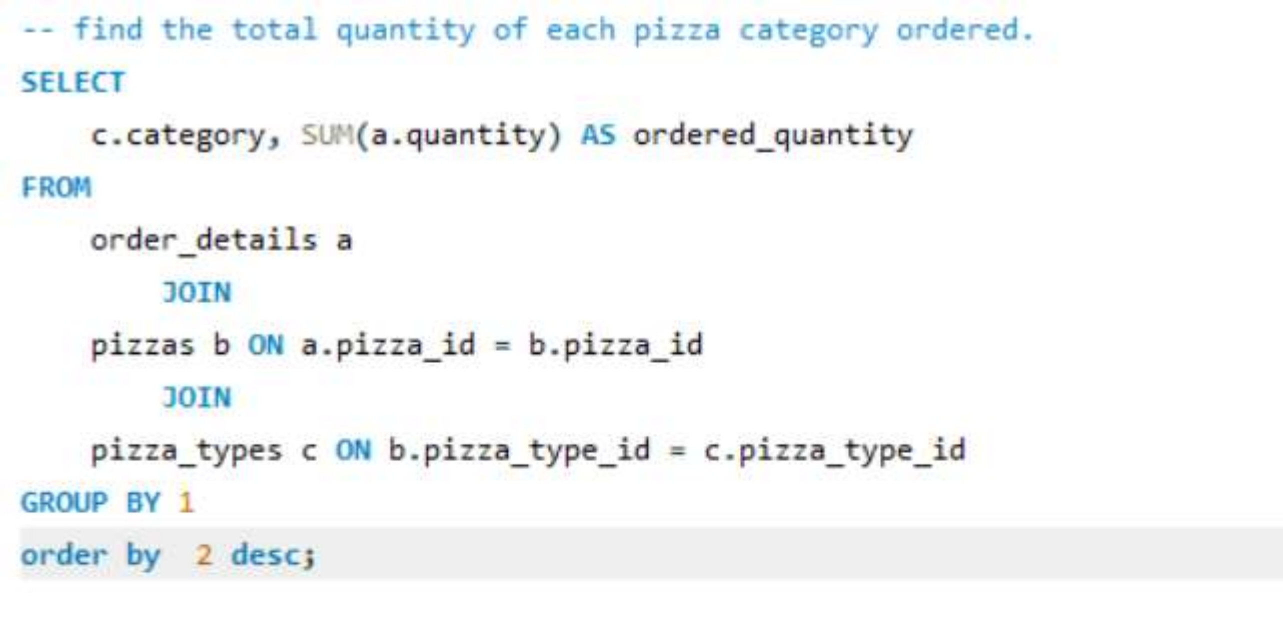


```sql
SELECT
    c.name, SUM(a.quantity) AS total_quantity
FROM
    order_details a
        JOIN
    pizzas b ON a.pizza_id = b.pizza_id
        JOIN
    pizza_types c ON b.pizza_type_id = c.pizza_type_id
GROUP BY 1
ORDER BY 2 DESC
LIMIT 5;
```



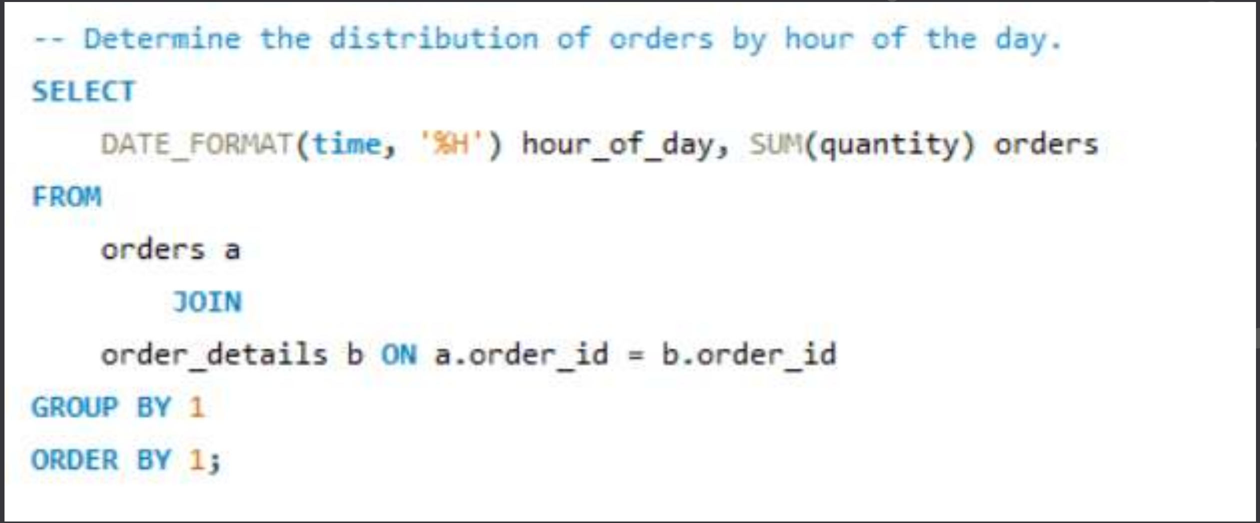
```sql
-- m2
with a as(SELECT
    c.name,
    SUM(a.quantity) AS total_quantity,
    dense_rank() over(order by SUM(a.quantity) desc ) rnk
FROM
    order_details a
        JOIN
    pizzas b ON a.pizza_id = b.pizza_id
        JOIN
    pizza_types c ON b.pizza_type_id = c.pizza_type_id
GROUP BY 1)

SELECT
    name, total_quantity
FROM
    a
WHERE
    rnk BETWEEN 1 AND 5;
```

find the total quantity of each pizza category ordered.



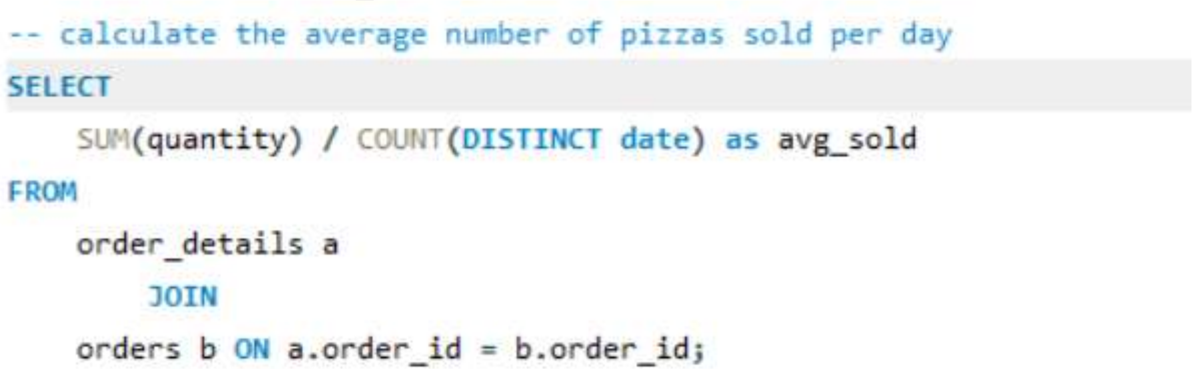
```sql
-- find the total quantity of each pizza category ordered.
SELECT
    c.category, SUM(a.quantity) AS ordered_quantity
FROM
    order_details a
        JOIN
    pizzas b ON a.pizza_id = b.pizza_id
        JOIN
    pizza_types c ON b.pizza_type_id = c.pizza_type_id
GROUP BY 1
order by  2 desc;
```

Determine the distribution of orders by hour of the day.



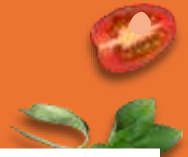
```
-- Determine the distribution of orders by hour of the day.
SELECT
    DATE_FORMAT(time, '%H') hour_of_day, SUM(quantity) orders
FROM
    orders a
        JOIN
    order_details b ON a.order_id = b.order_id
GROUP BY 1
ORDER BY 1;
```

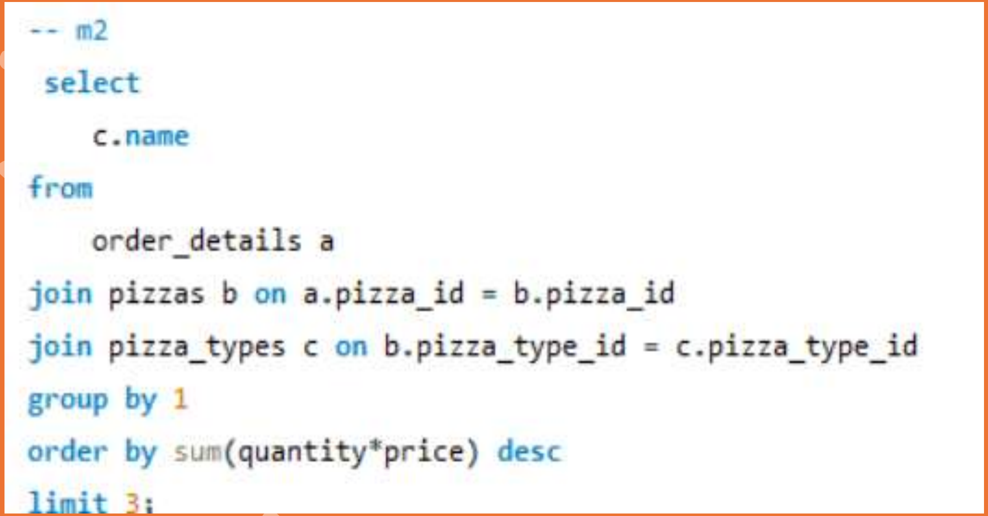
calculate the average number of pizzas sold per day



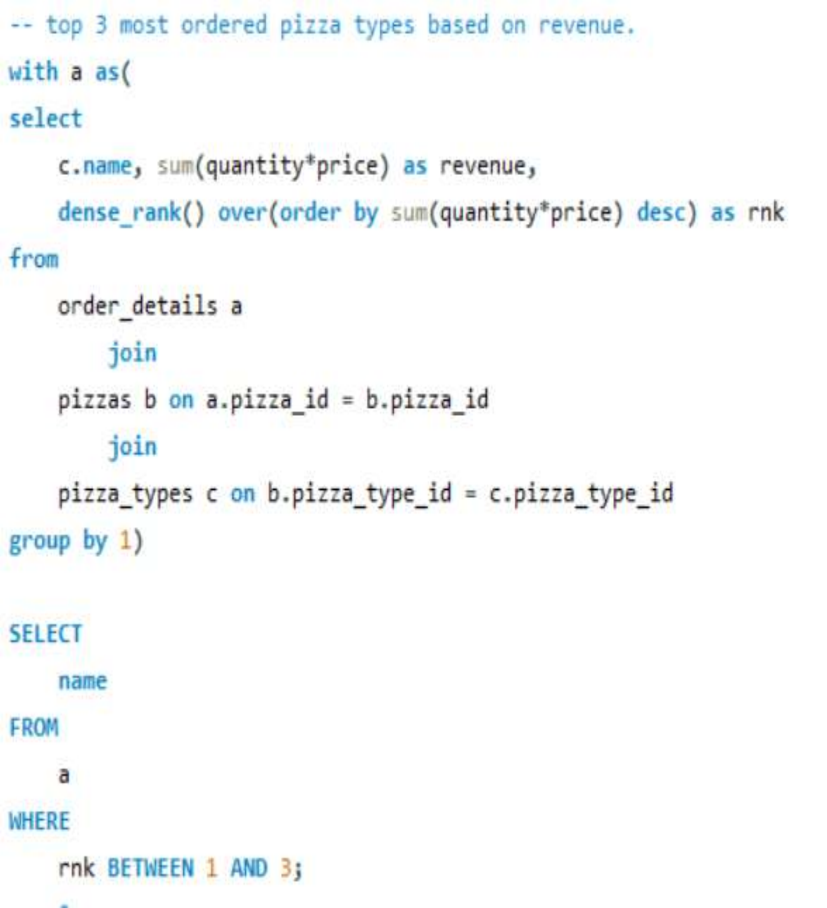
```
-- calculate the average number of pizzas sold per day
SELECT
    SUM(quantity) / COUNT(DISTINCT date) as avg_sold
FROM
    order_details a
        JOIN
    orders b ON a.order_id = b.order_id;
```

Top 3 most ordered pizza types based on revenue.



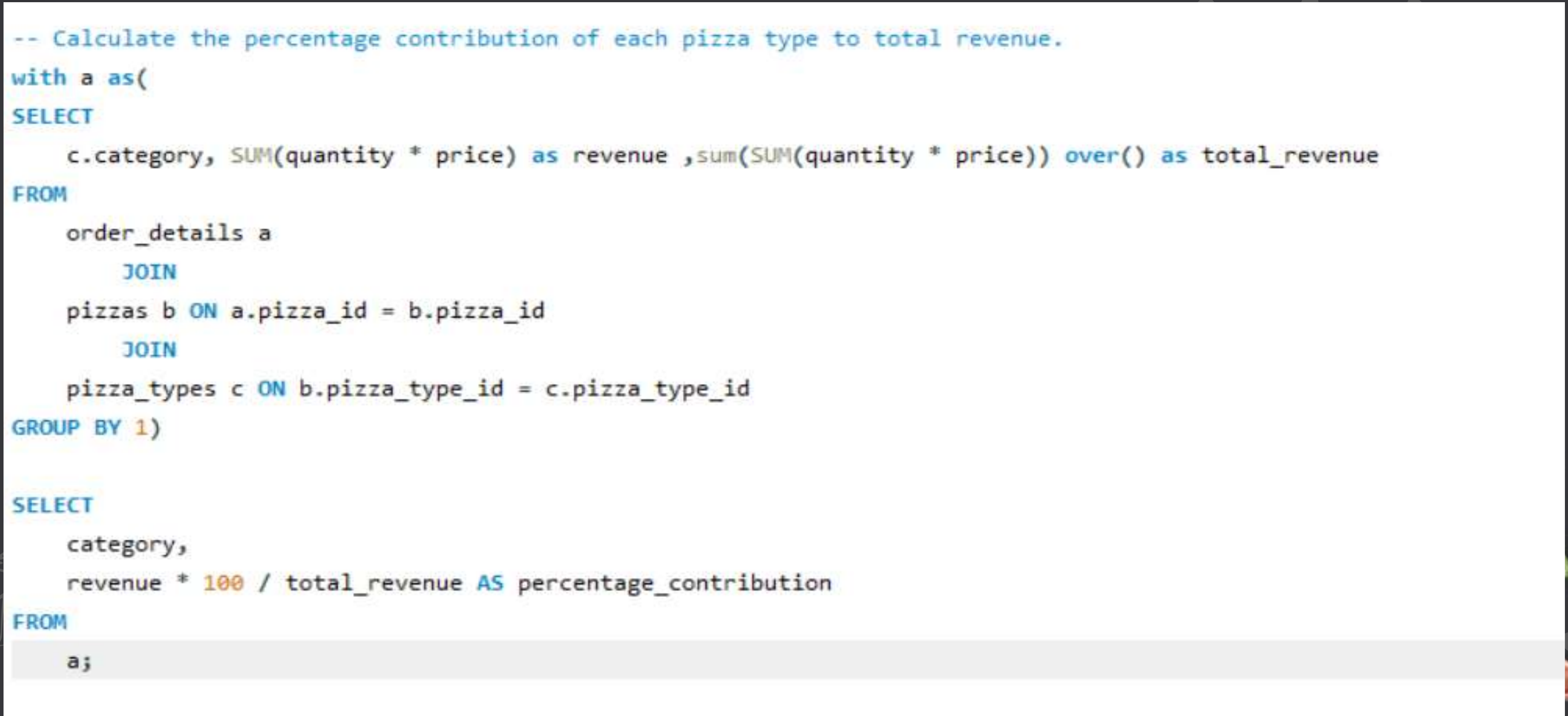


```
-- m2
 select
    c.name
from
    order_details a
join pizzas b on a.pizza_id = b.pizza_id
join pizza_types c on b.pizza_type_id = c.pizza_type_id
group by 1
order by sum(quantity*price) desc
limit 3;
```



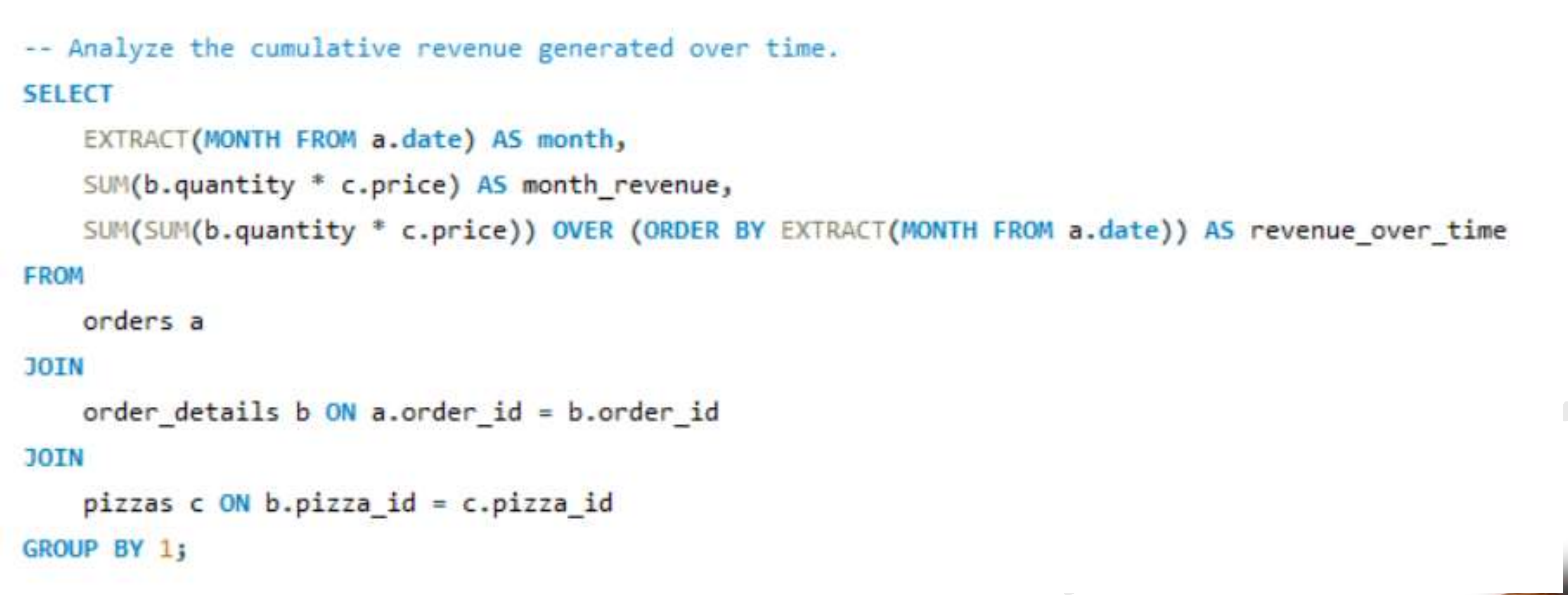
```
-- top 3 most ordered pizza types based on revenue.
with a as(
select
    c.name, sum(quantity*price) as revenue,
    dense_rank() over(order by sum(quantity*price) desc) as rnk
from
    order_details a
        join
    pizzas b on a.pizza_id = b.pizza_id
        join
    pizza_types c on b.pizza_type_id = c.pizza_type_id
group by 1)

SELECT
    name
FROM
    a
WHERE
    rnk BETWEEN 1 AND 3;
```

Calculate the percentage contribution of each pizza type to total revenue.



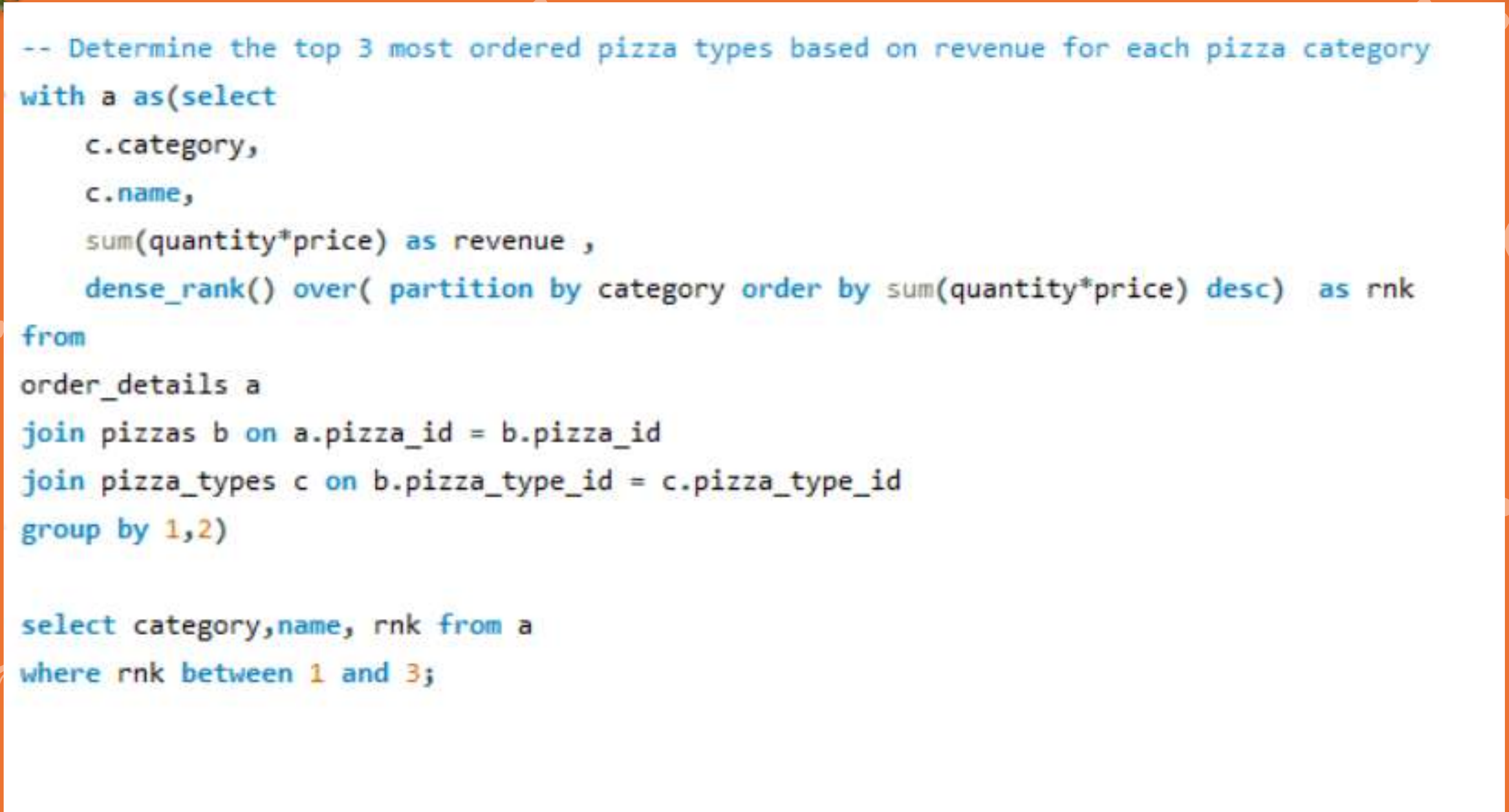
```
-- Calculate the percentage contribution of each pizza type to total revenue.
with a as(
SELECT
    c.category, SUM(quantity * price) as revenue ,sum(SUM(quantity * price)) over() as total_revenue
FROM
    order_details a
        JOIN
    pizzas b ON a.pizza_id = b.pizza_id
        JOIN
    pizza_types c ON b.pizza_type_id = c.pizza_type_id
GROUP BY 1)

SELECT
    category,
    revenue * 100 / total_revenue AS percentage_contribution
FROM
    a;
```

Analyze the cumulative revenue generated over time.



```
-- Analyze the cumulative revenue generated over time.
SELECT
    EXTRACT(MONTH FROM a.date) AS month,
    SUM(b.quantity * c.price) AS month_revenue,
    SUM(SUM(b.quantity * c.price)) OVER (ORDER BY EXTRACT(MONTH FROM a.date)) AS revenue_over_time
FROM
    orders a
JOIN
    order_details b ON a.order_id = b.order_id
JOIN
    pizzas c ON b.pizza_id = c.pizza_id
GROUP BY 1;
```

Determine the top 3 most ordered pizza types based on revenue for each pizza category



```
-- Determine the top 3 most ordered pizza types based on revenue for each pizza category
with a as(select
    c.category,
    c.name,
    sum(quantity*price) as revenue ,
    dense_rank() over( partition by category order by sum(quantity*price) desc)  as rnk
from
order_details a
join pizzas b on a.pizza_id = b.pizza_id
join pizza_types c on b.pizza_type_id = c.pizza_type_id
group by 1,2)

select category,name, rnk from a
where rnk between 1 and 3;
```