Pizza Hut Sales Analysis using SQL

No one outpizzas the hut

CONTENTS

01 Objective

02 Model Schema

03 Questions are Answered

04 Key Insights

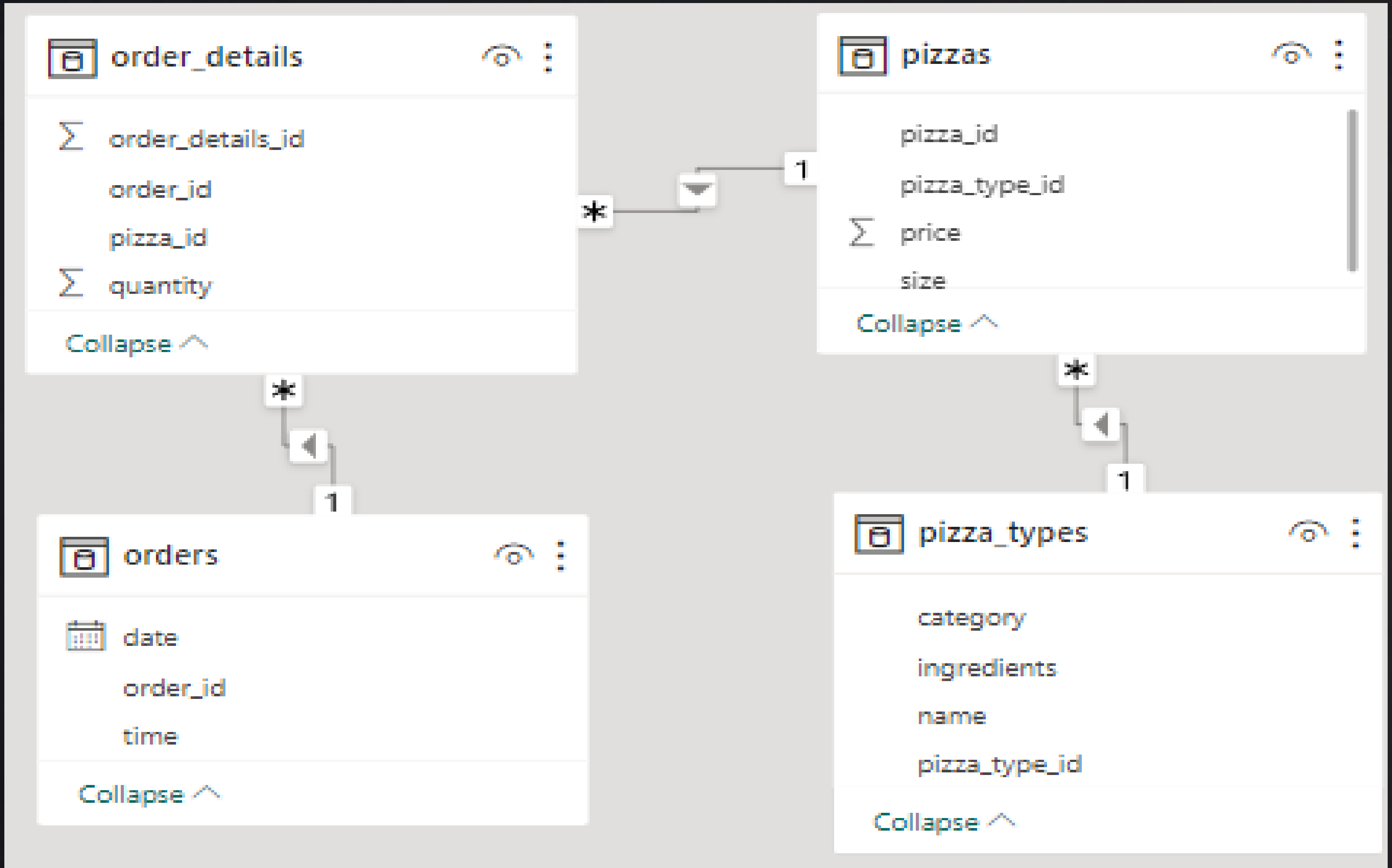
05 Recommandations

Objective

The objective of this project is to efficiently query the Pizza Hut sales data base and analyse the sales data to evaluate customer preferences, under stand customer behavior, and assess customer satisfaction, ultimately facilitating informed decision-making and driving business growth.

Tool Used: *SQL and MySQL database*

Model Schema



order_details

- ∑ order_details_id
- order_id
- pizza_id
- ∑ quantity

pizzas

- pizza_id
- pizza_type_id
- ∑ price
- size

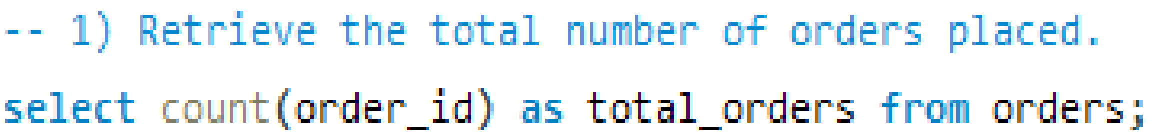
orders

- date
- order_id
- time

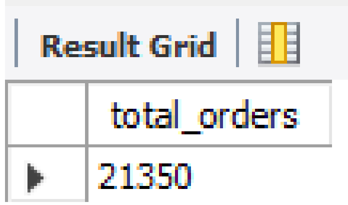
pizza_types

- category
- ingredients
- name
- pizza_type_id

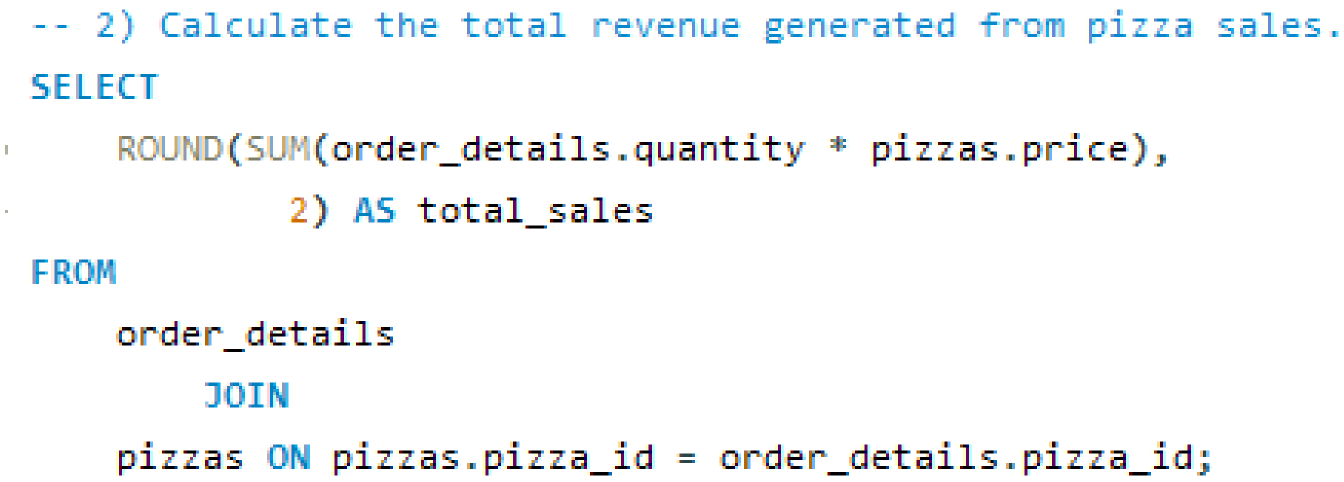
Questions are Answered



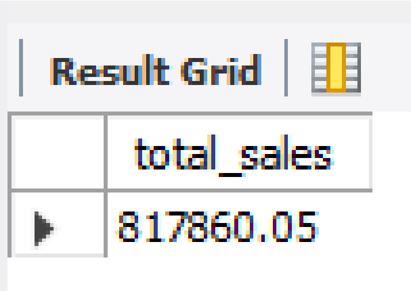
```
-- 1) Retrieve the total number of orders placed.
select count(order_id) as total_orders from orders;
```



total_orders
21350

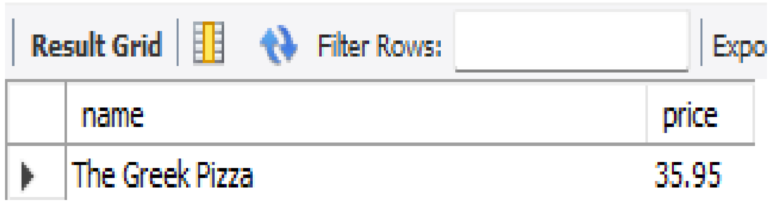


```
-- 2) Calculate the total revenue generated from pizza sales.
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id;
```



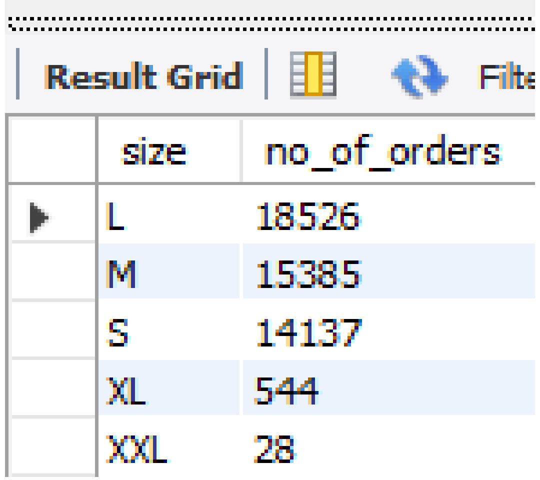
total_sales
817860.05

```
-- 3) Identify the highest-priced pizza.
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

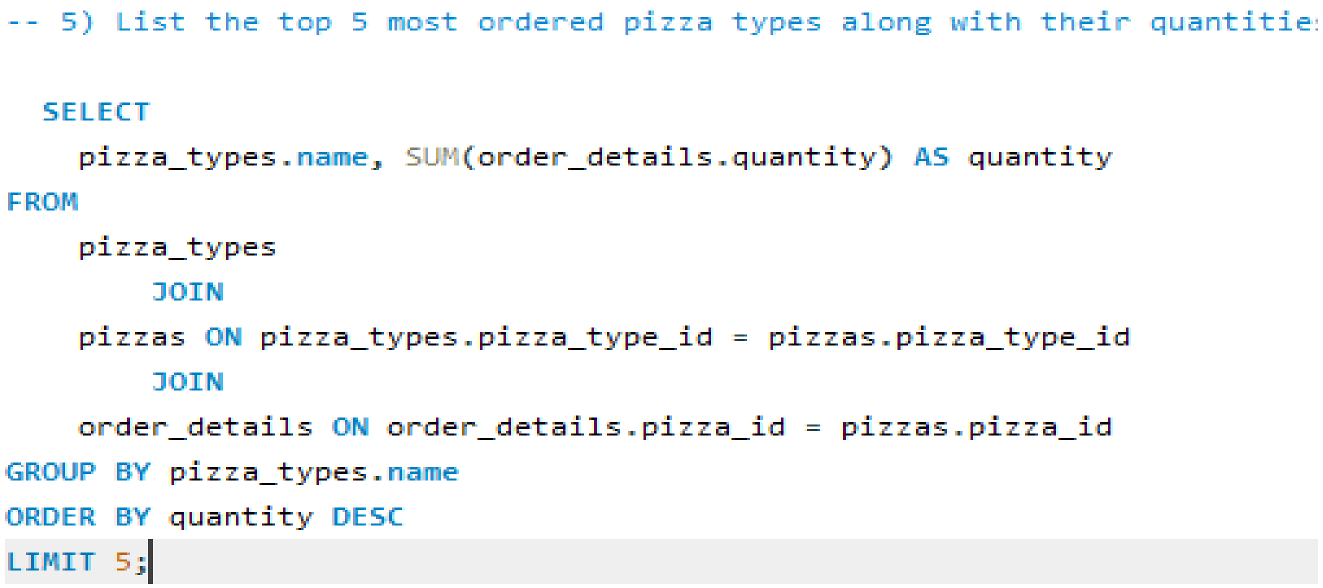


name	price
The Greek Pizza	35.95

```
-- 4) Identify the most common pizza size ordered.
select quantity, count(order_details_id)
from order_details group by  quantity;
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS no_of_orders
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY no_of_orders DESC;
```

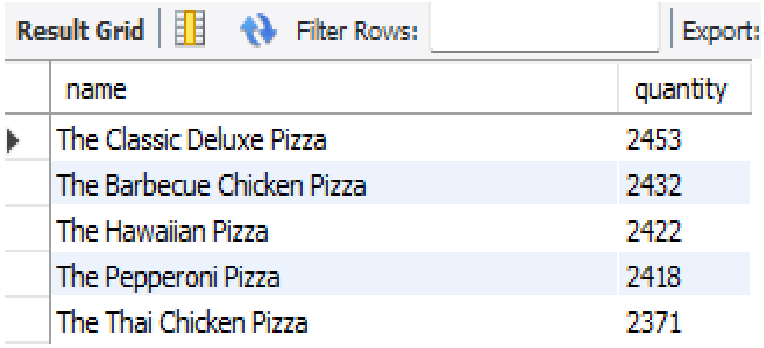


size	no_of_orders
L	18526
M	15385
S	14137
XL	544
XXL	28

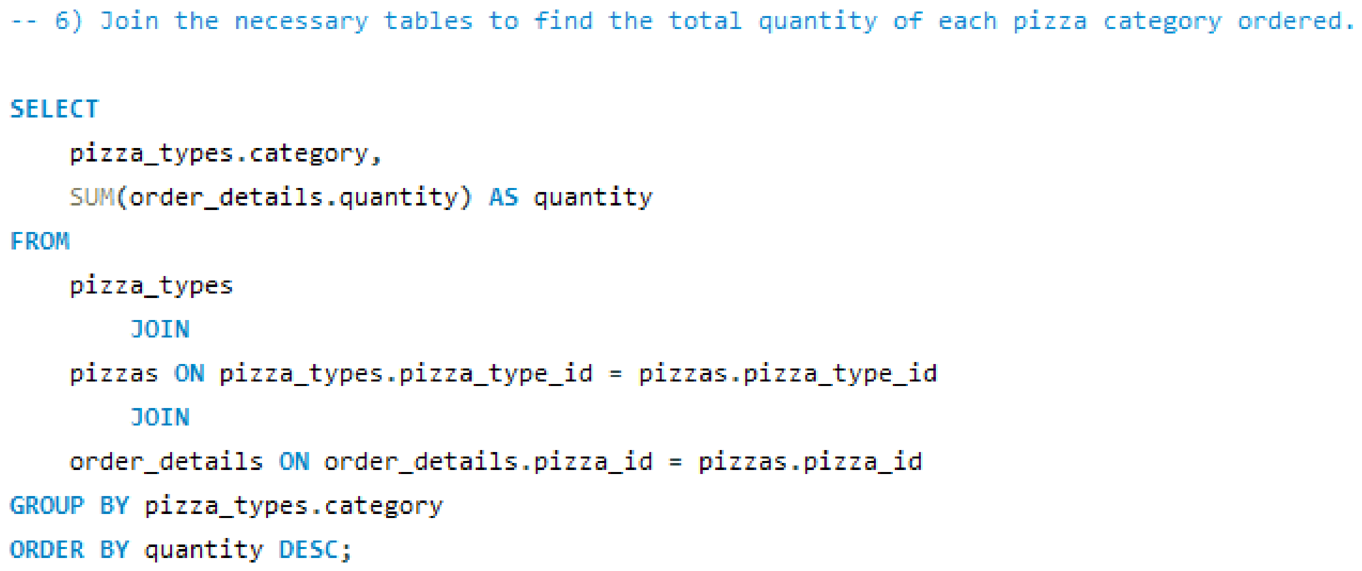


```sql
-- 5) List the top 5 most ordered pizza types along with their quantitie

 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;
```

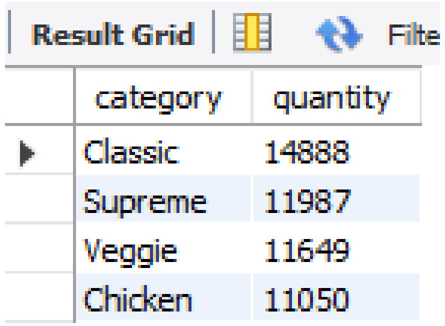


name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371



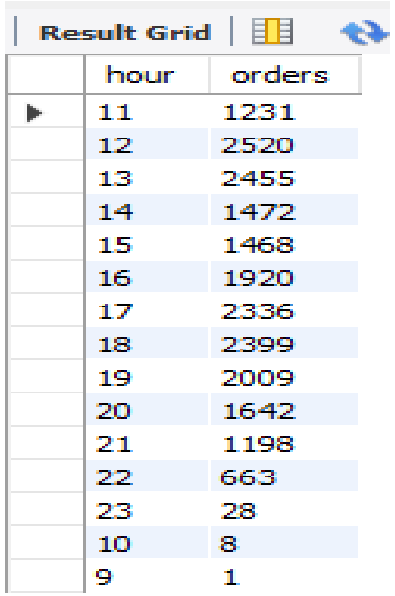
```sql
-- 6) Join the necessary tables to find the total quantity of each pizza category ordered.

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;
```



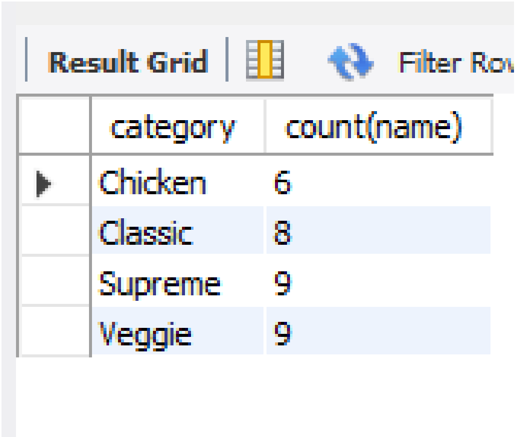
category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

```
-- 7) Determine the distribution of orders by hour of the day.
select hour(order_time)as hour, count(order_id) as orders from orders
group by hour(order_time);
```

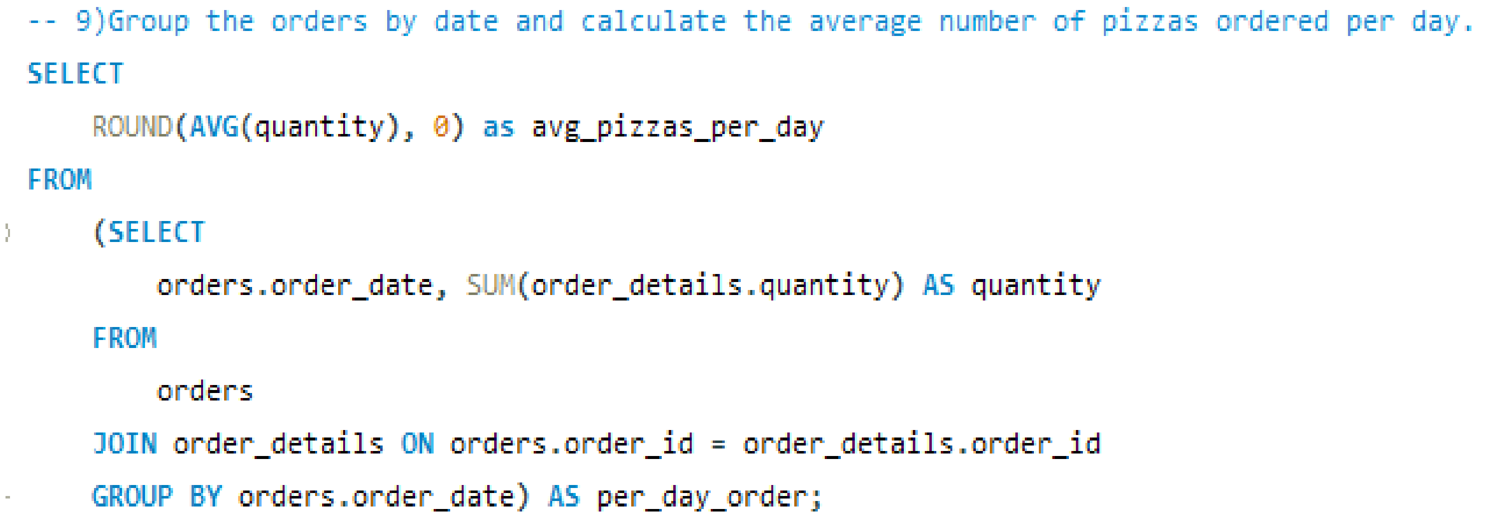


hour	orders
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

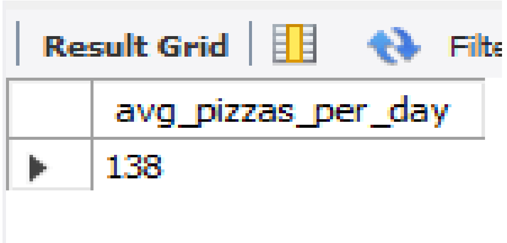
```
-- 8) Join relevant tables to find the category-wise distribution of pizzas.
select category, count(name) from pizza_types
group by category;
```



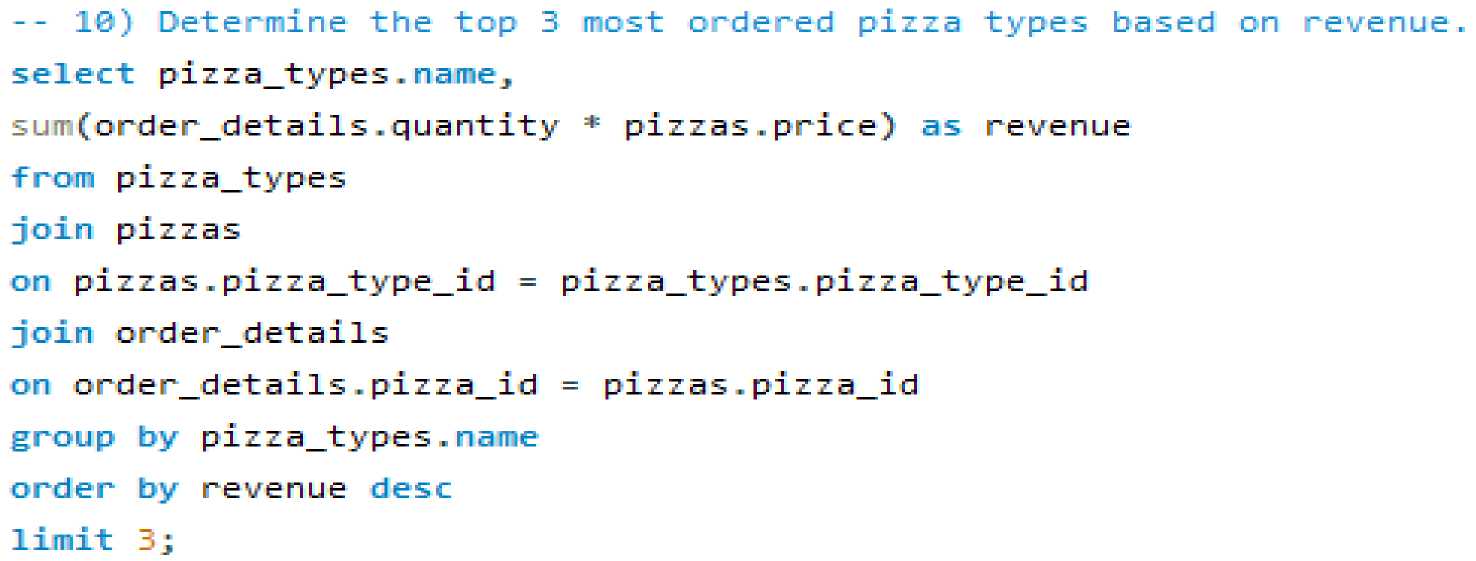
category	count(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9



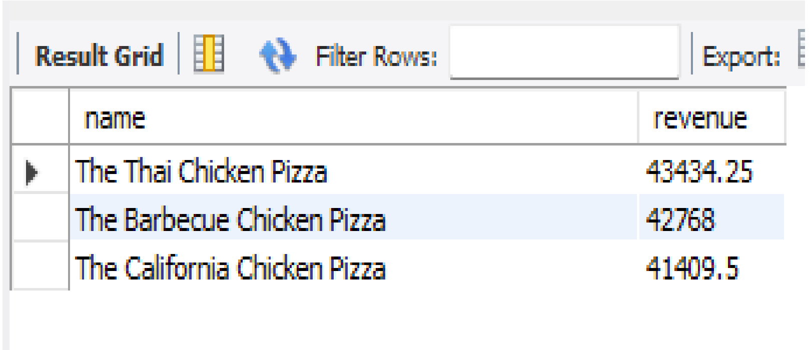
```
-- 9)Group the orders by date and calculate the average number of pizzas ordered per day.
SELECT
    ROUND(AVG(quantity), 0) as avg_pizzas_per_day
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS per_day_order;
```



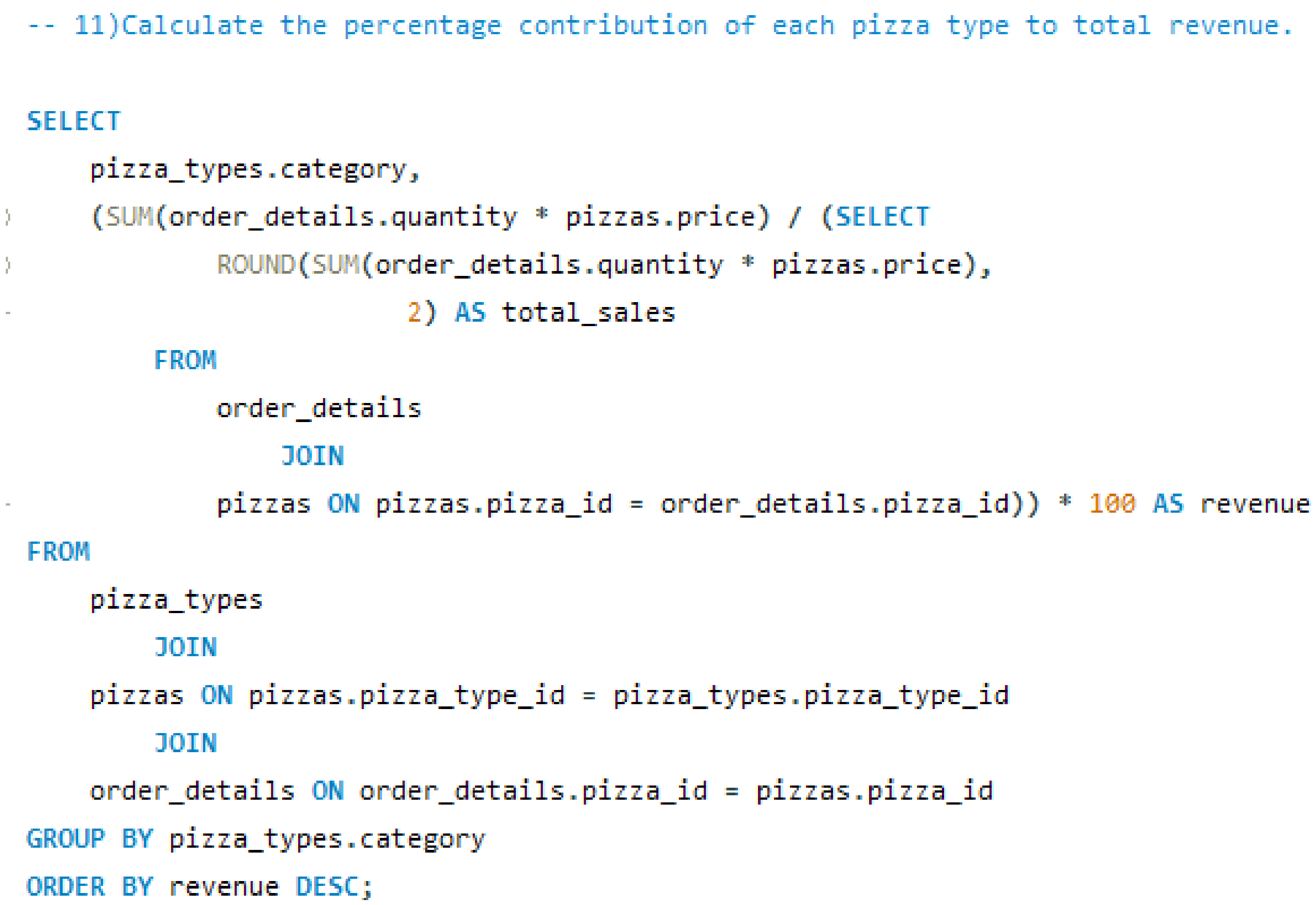
avg_pizzas_per_day
138



```
-- 10) 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;
```

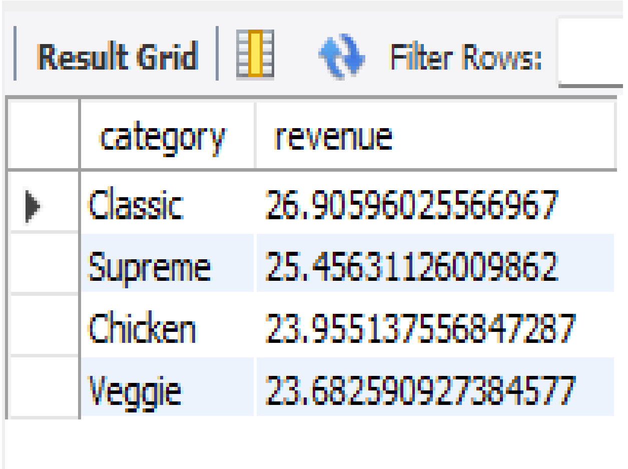


name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5



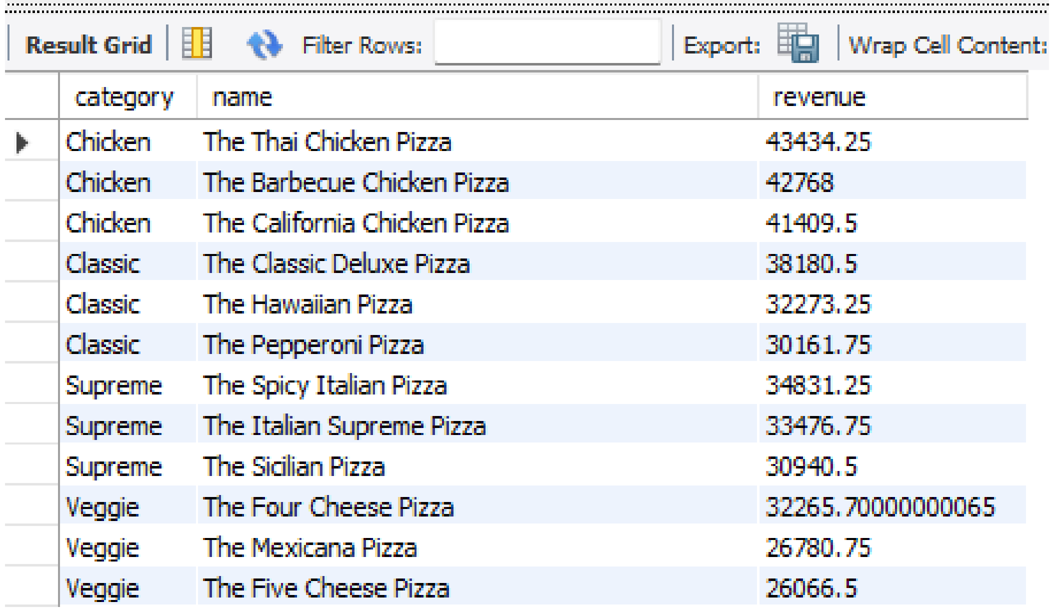
```sql
-- 11)Calculate the percentage contribution of each pizza type to total revenue.

SELECT
    pizza_types.category,
    (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 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.category
ORDER BY revenue DESC;
```



category	revenue
Classic	26.90596025566967
Supreme	25.45631126009862
Chicken	23.955137556847287
Veggie	23.682590927384577

```
-- 12) Determine the top 3 most ordered pizza types based on revenue for each pizza category.
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 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;
```



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

Key Insights

Customer Preferences:

- ***Pizza Size:*** *Large pizzas are the most popular, suggesting customers tend to feed multiple people or prefer leftovers.*
- ***Pizza Flavors:*** *The classics reign supreme! "The Classic Deluxe Pizza" and "The Barbecue Chicken Pizza" are the most frequently ordered, indicating a preference for familiar Flavors.*

Menu and Sales:

- ***High-Priced Item:*** *"The Greek Pizza" might be the most expensive item, but it doesn't necessarily translate to high sales volume.*
- ***Top Seller:*** *Interestingly, "The Thai Chicken Pizza" generates the highest revenue despite not being the most ordered. This suggests it has a higher price point or a larger profit margin compared to other pizzas.*

Busy Times:

- ***Lunch and Dinner Rush:*** *The busiest hours are around lunchtime (12 PM to 1 PM) and dinner time (5 PM to 6 PM), aligning with typical meal times.*

Recommendations

- *Highlight "The Greek Pizza" in marketing campaigns to attract higher-spending customers and create special promotions or bundles that include this premium pizza.*
- *Promote large size pizzas by offering discounts or deals, and emphasize their value in advertisements to appeal to families and groups.*
- *Focus on popular pizzas by ensuring "The Classic Deluxe Pizza" and "The Barbecue Chicken Pizza" are prominently featured on the menu and in promotional materials, and introduce combo deals or limited-time offers featuring these pizzas.*
- *Optimize staffing and resources during peak hours (12 PM to 1 PM and 5 PM to 6 PM) by ensuring adequate staffing, preparing ingredients in advance, and optimizing delivery routes to manage the rush efficiently.*

THANK YOU

V. Karthik Chary

vkc0793@gmail.com