PIZZA STORE ANALYSIS

USING SQL QUERIES
By: Karan Malhotra

CSV DATASET

Pizza_types

- Pizza_type_id
- Name
- Category
- ingredients

Order_details

- Order_detail_id
- Order_id
- Pizza_id
- quantity

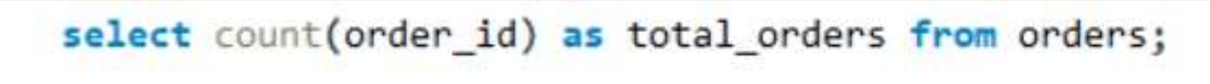
orders

- order_id
- Order_date
- Order_time

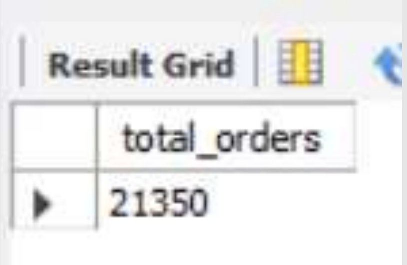
Pizza

- Pizza_type_id
- Pizza_id
- size
- price

1. Retrieve the total number of orders placed.

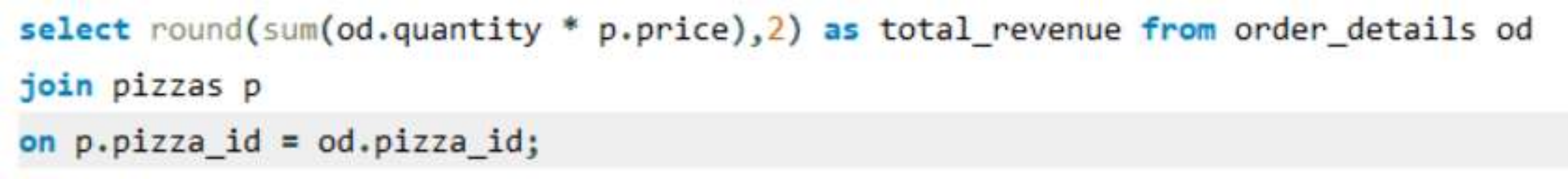


```sql
select count(order_id) as total_orders from orders;
```

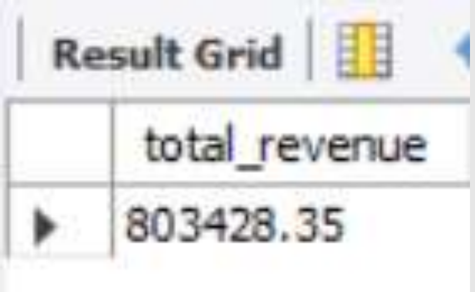


```
total_orders
21350
```

2. Calculate the total revenue generated from pizza sales.

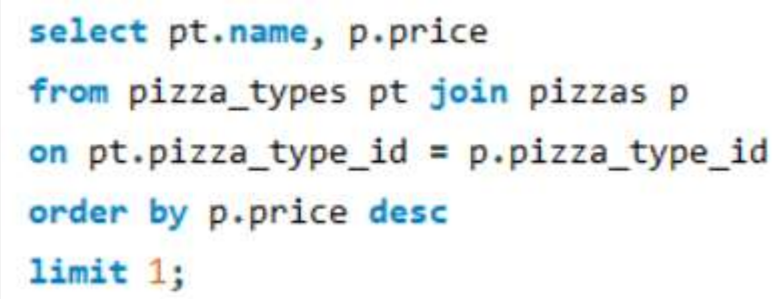


```sql
select round(sum(od.quantity * p.price),2) as total_revenue from order_details od
join pizzas p
on p.pizza_id = od.pizza_id;
```

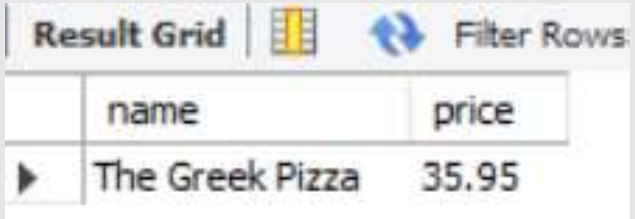


```
total_revenue
803428.35
```

3. Identify the highest-priced pizza.

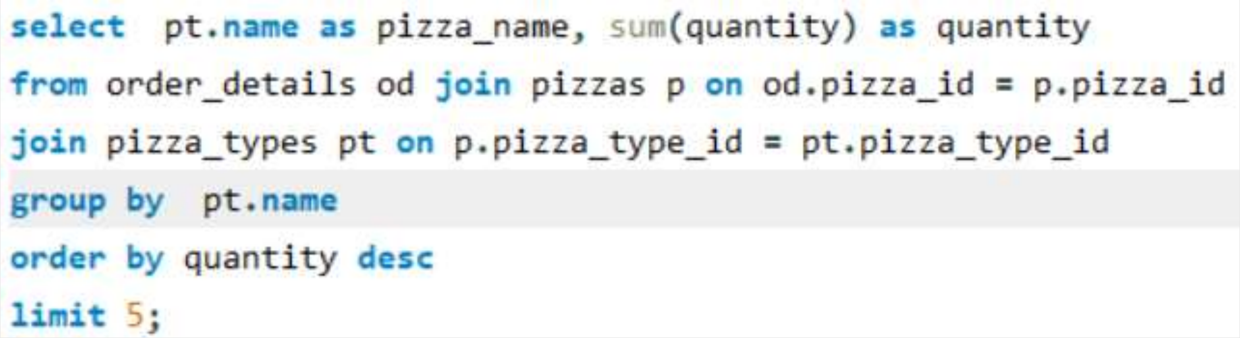


```
select pt.name, p.price
from pizza_types pt join pizzas p
on pt.pizza_type_id = p.pizza_type_id
order by p.price desc
limit 1;
```

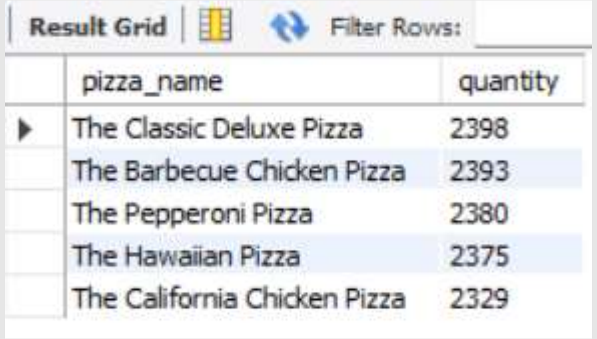


name	price
The Greek Pizza	35.95

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

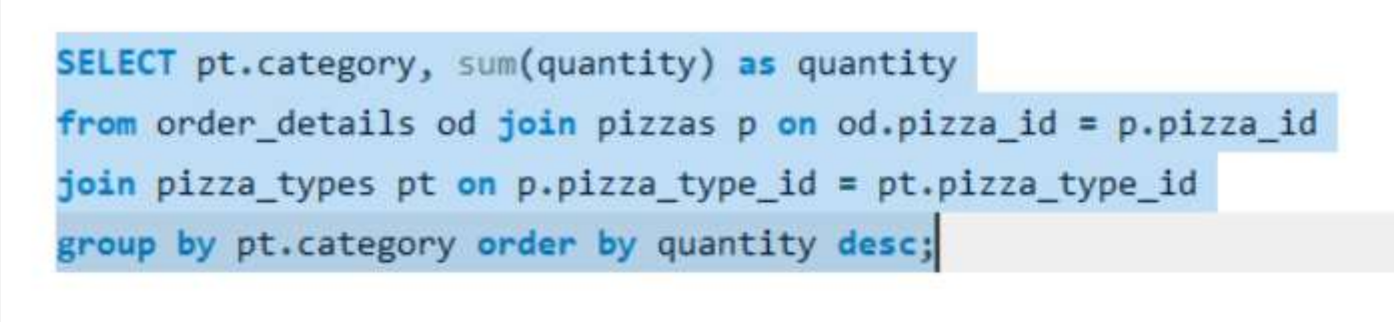


```
select  pt.name as pizza_name, sum(quantity) as quantity
from order_details od join pizzas p on od.pizza_id = p.pizza_id
join pizza_types pt on p.pizza_type_id = pt.pizza_type_id
group by  pt.name
order by quantity desc
limit 5;
```

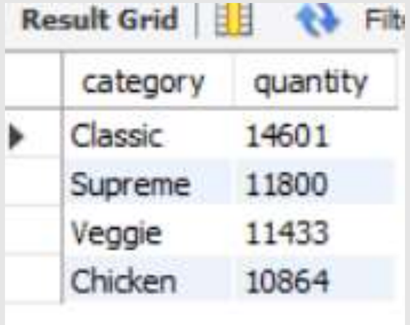


pizza_name	quantity
The Classic Deluxe Pizza	2398
The Barbecue Chicken Pizza	2393
The Pepperoni Pizza	2380
The Hawaiian Pizza	2375
The California Chicken Pizza	2329

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

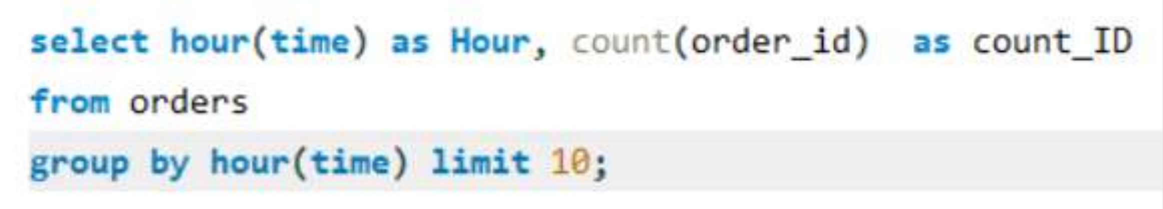


```sql
SELECT pt.category, sum(quantity) as quantity
from order_details od join pizzas p on od.pizza_id = p.pizza_id
join pizza_types pt on p.pizza_type_id = pt.pizza_type_id
group by pt.category order by quantity desc;
```

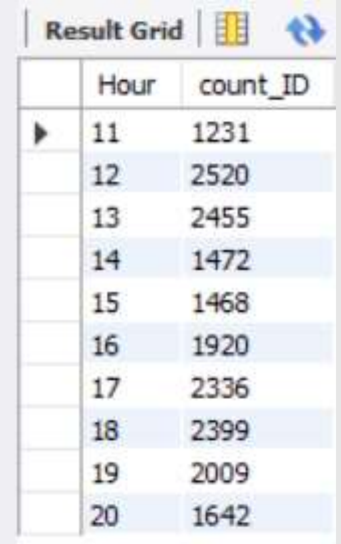


category	quantity
Classic	14601
Supreme	11800
Veggie	11433
Chicken	10864

6. Determine the distribution of orders by hour of the day.

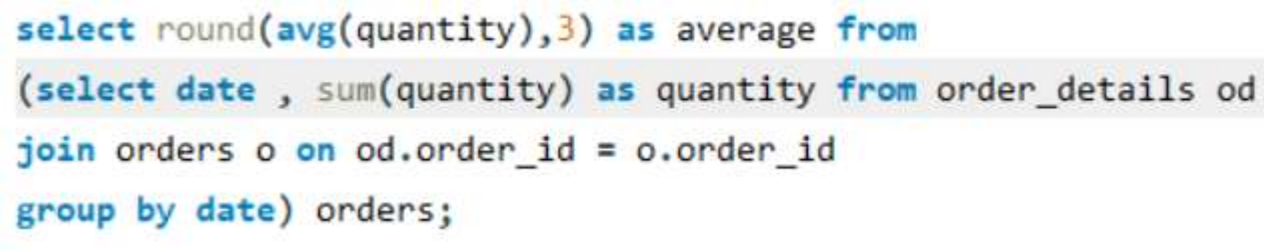


```sql
select hour(time) as Hour, count(order_id)  as count_ID
from orders
group by hour(time) limit 10;
```

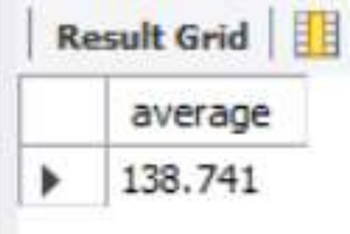


Hour	count_ID
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642

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

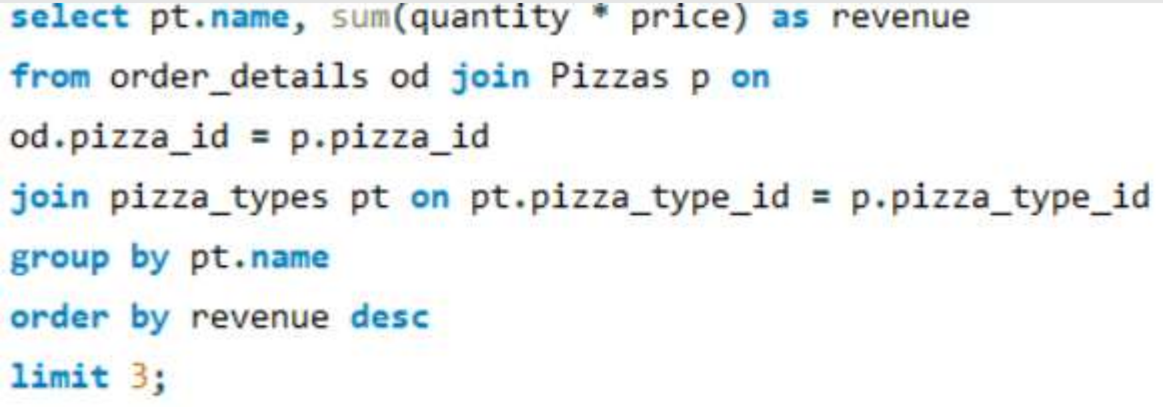


```
select round(avg(quantity),3) as average from
(select date , sum(quantity) as quantity from order_details od
join orders o on od.order_id = o.order_id
group by date) orders;
```

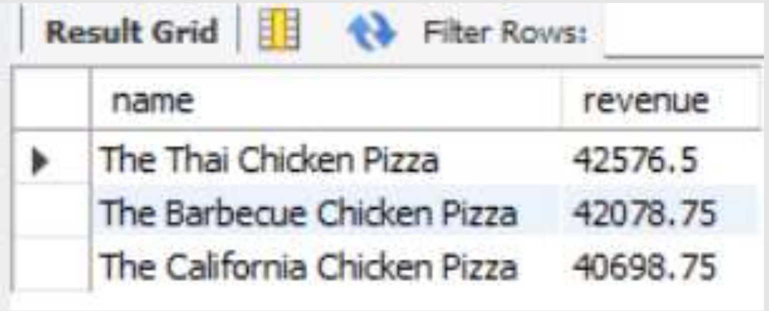


	average
▶	138.741

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

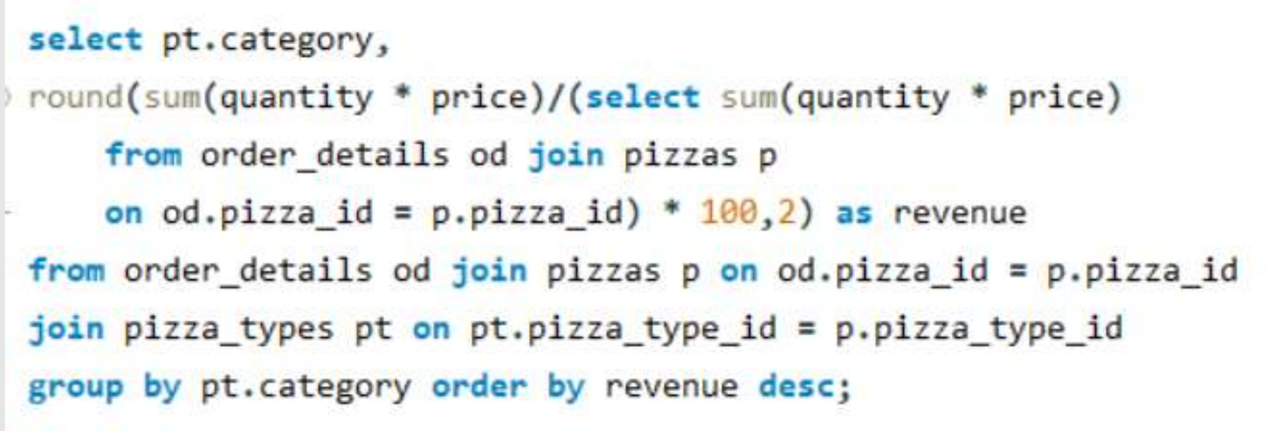


```
select pt.name, sum(quantity * price) as revenue
from order_details od join Pizzas p on
od.pizza_id = p.pizza_id
join pizza_types pt on pt.pizza_type_id = p.pizza_type_id
group by pt.name
order by revenue desc
limit 3;
```

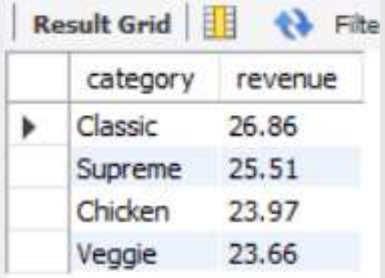


	name	revenue
▶	The Thai Chicken Pizza	42576.5
	The Barbecue Chicken Pizza	42078.75
	The California Chicken Pizza	40698.75

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

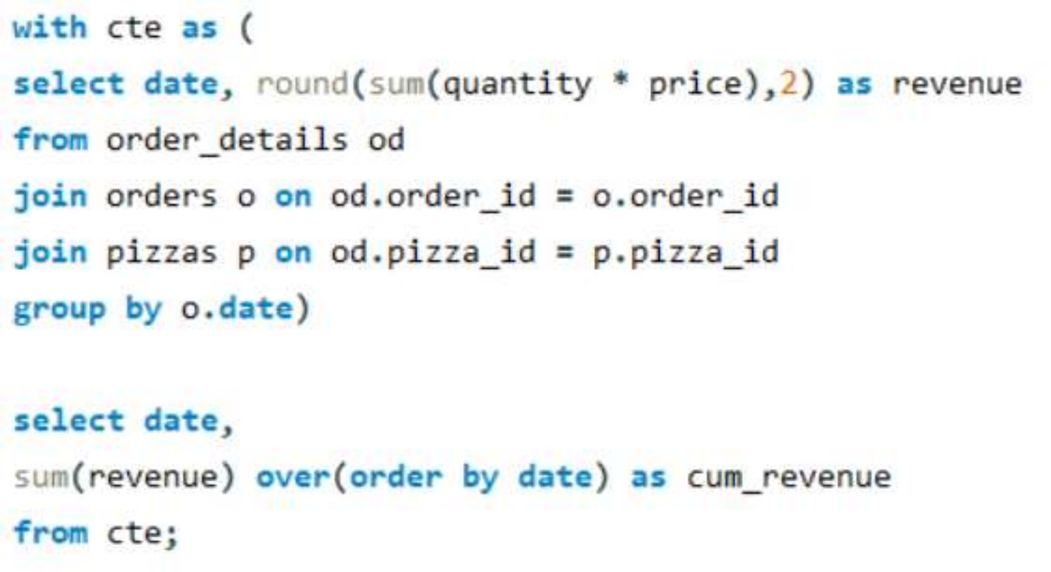


```
select pt.category,
round(sum(quantity * price)/(select sum(quantity * price)
    from order_details od join pizzas p
    on od.pizza_id = p.pizza_id) * 100,2) as revenue
from order_details od join pizzas p on od.pizza_id = p.pizza_id
join pizza_types pt on pt.pizza_type_id = p.pizza_type_id
group by pt.category order by revenue desc;
```



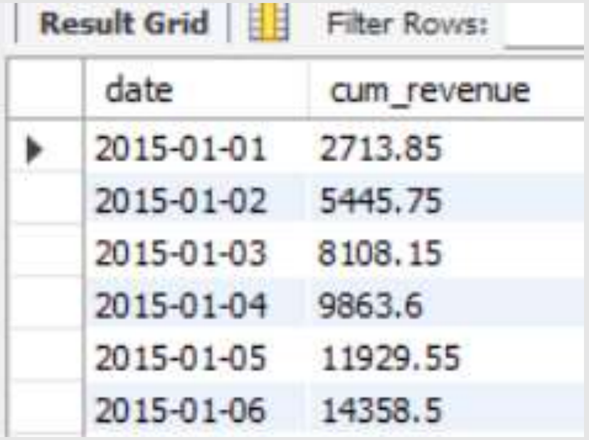
category	revenue
Classic	26.86
Supreme	25.51
Chicken	23.97
Veggie	23.66

10. Analyze the cumulative revenue generated over time.



```
with cte as (
select date, round(sum(quantity * price),2) as revenue
from order_details od
join orders o on od.order_id = o.order_id
join pizzas p on od.pizza_id = p.pizza_id
group by o.date)

select date,
sum(revenue) over(order by date) as cum_revenue
from cte;
```



date	cum_revenue
2015-01-01	2713.85
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