Pizza Sales Analysis: Unlocking Insights with SQL

Hello , I am Abhishek Dhokale.
Welcome to my SQL project focused on analyzing pizza sales data. Let's delve into the data-driven world of pizza sales.

Introduction to the SQL Project

Project Goals

This SQL project aims to analyze pizza sales data from a fictional pizzeria. We'll uncover valuable insights from the pizza sales data using SQL.

Data Source

The project leverages a comprehensive dataset of pizza sales transactions, including Order Details, Orders, Pizza, Pizza Type.

Exploring the Pizza Sales Database

Order Details

Contains details on each sale, including order ID, Pizza ID , Order Detail ID , Quantity.

Orders Table

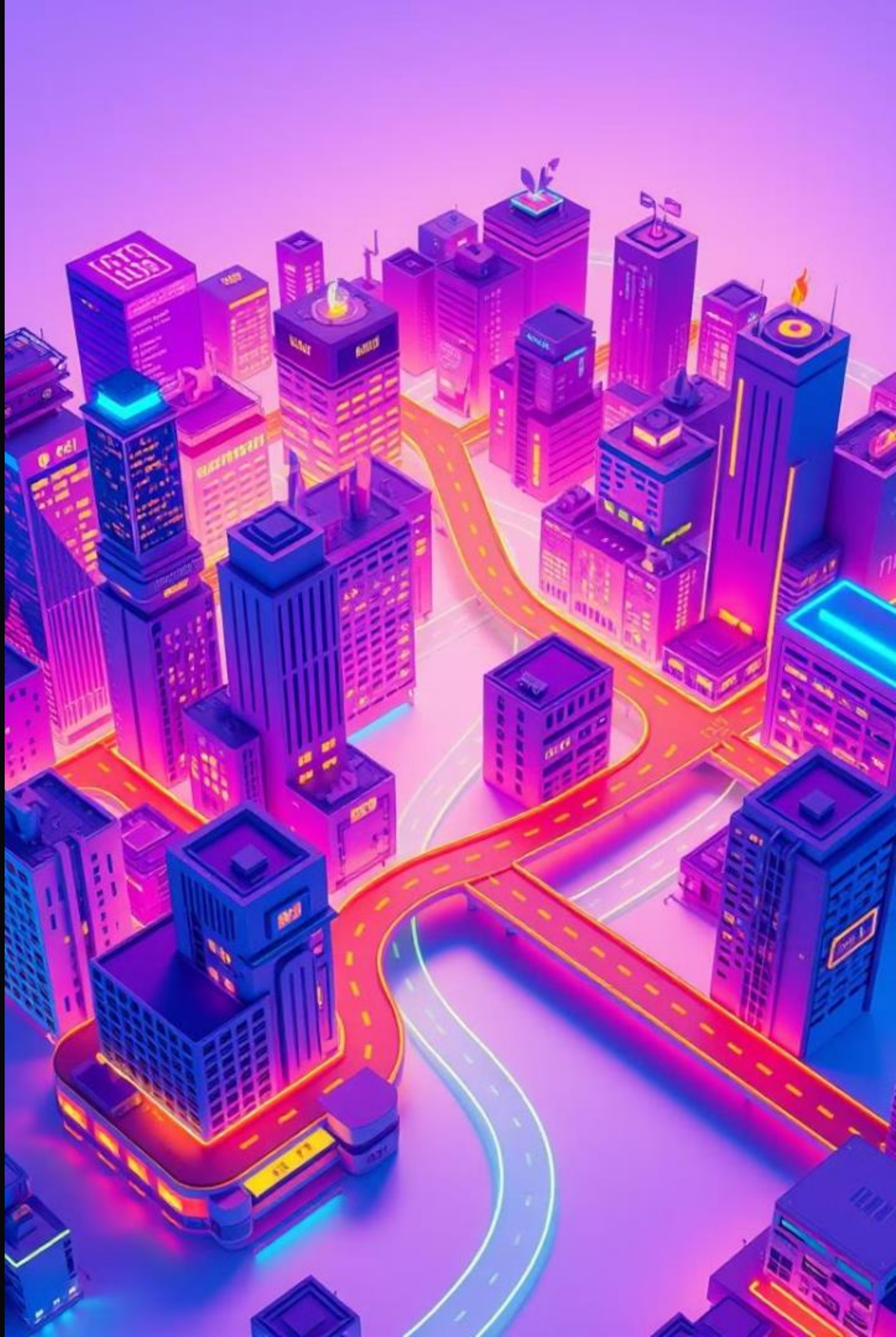
Lists all orders placed, including order ID, Order Date, Order Time.

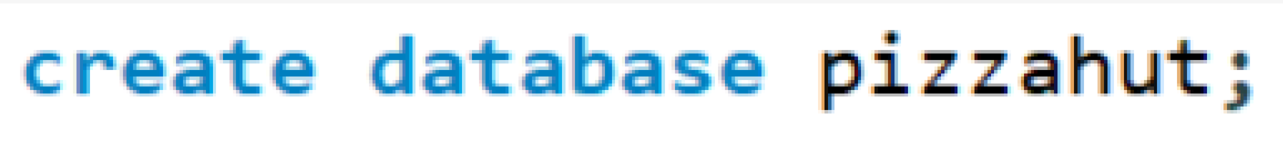
Pizzas

Stores Pizza information like Pizza ID , Pizza Type ID, Pizza Size and Price.

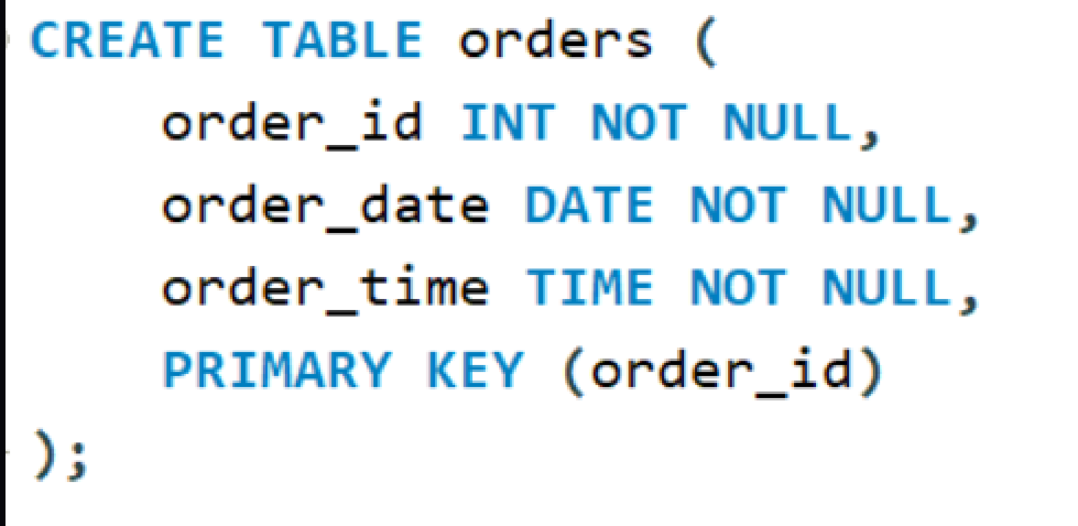
Pizza Details

Lists all pizza including their Names, Pizza Type ID, Category, and Ingredients.

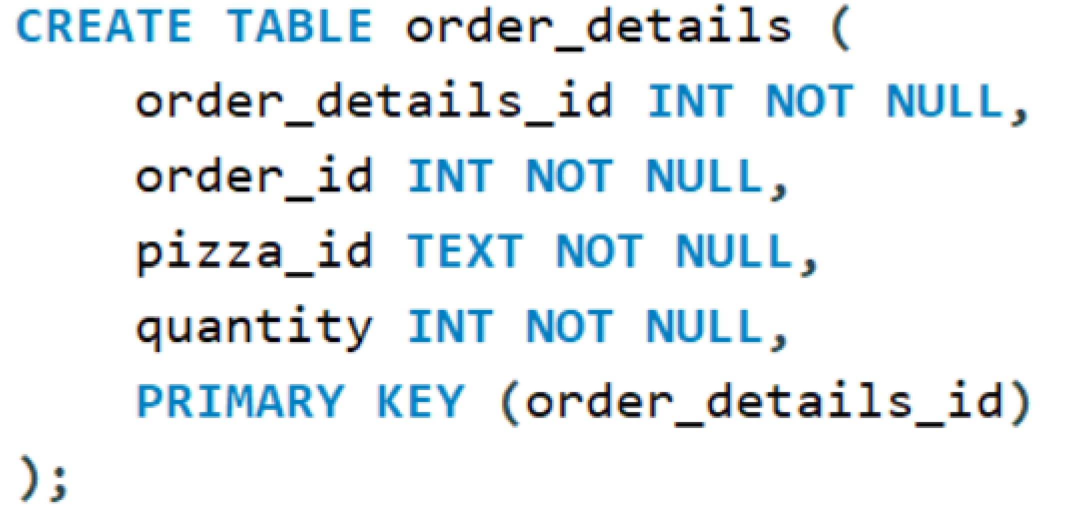




```
create database pizzahut;
```



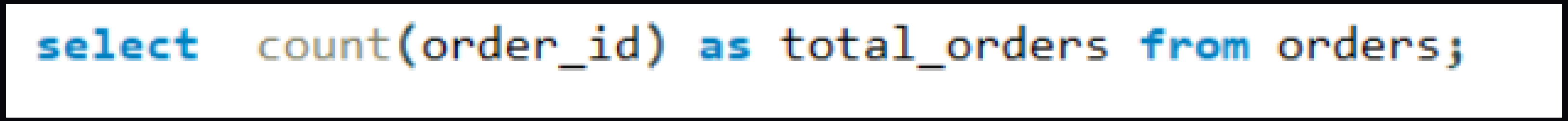
```
CREATE TABLE orders (
    order_id INT NOT NULL,
    order_date DATE NOT NULL,
    order_time TIME NOT NULL,
    PRIMARY KEY (order_id)
);
```



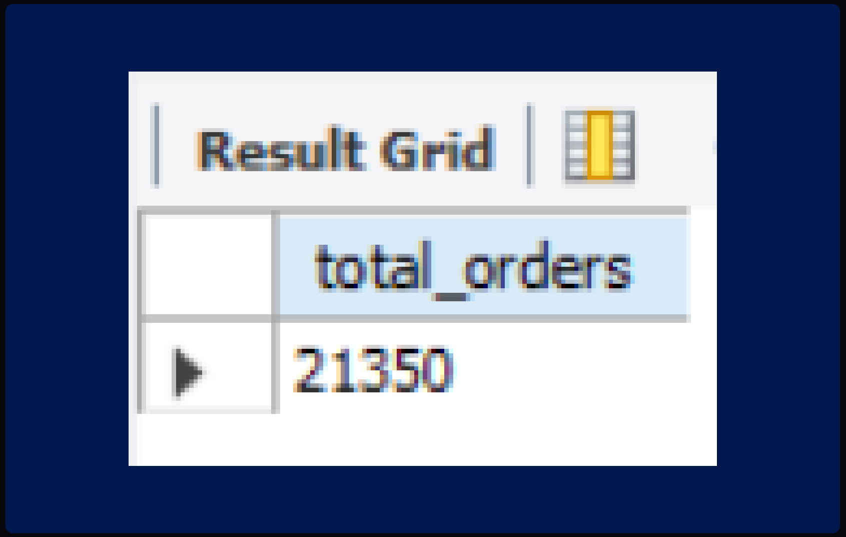
```
CREATE TABLE order_details (
    order_details_id INT NOT NULL,
    order_id INT NOT NULL,
    pizza_id TEXT NOT NULL,
    quantity INT NOT NULL,
    PRIMARY KEY (order_details_id)
);
```

SQL Queries

Retrieve the total number of orders placed.



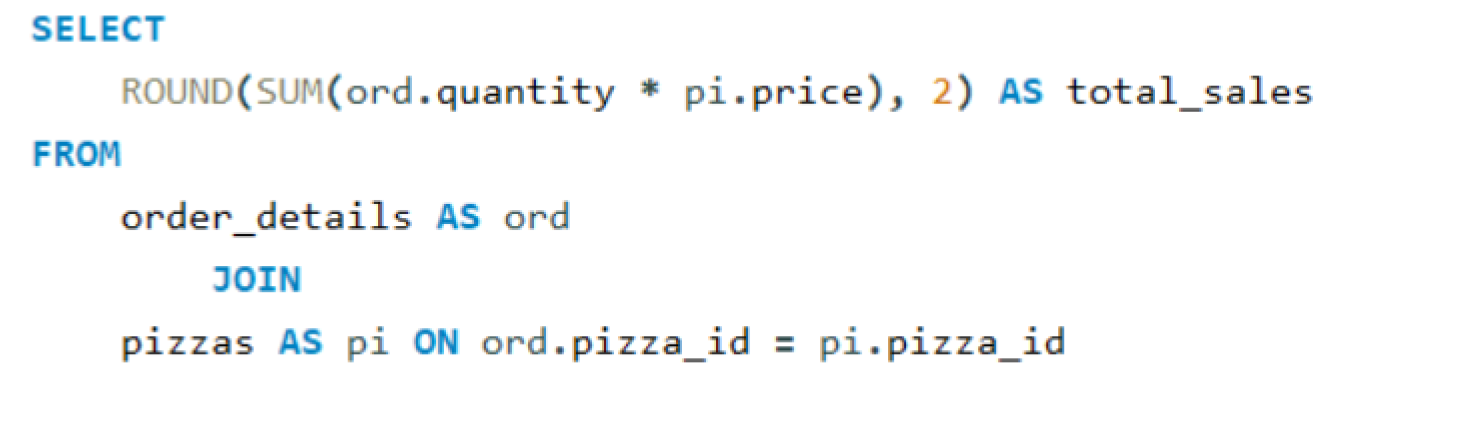
```sql
select  count(order_id) as total_orders from orders;
```



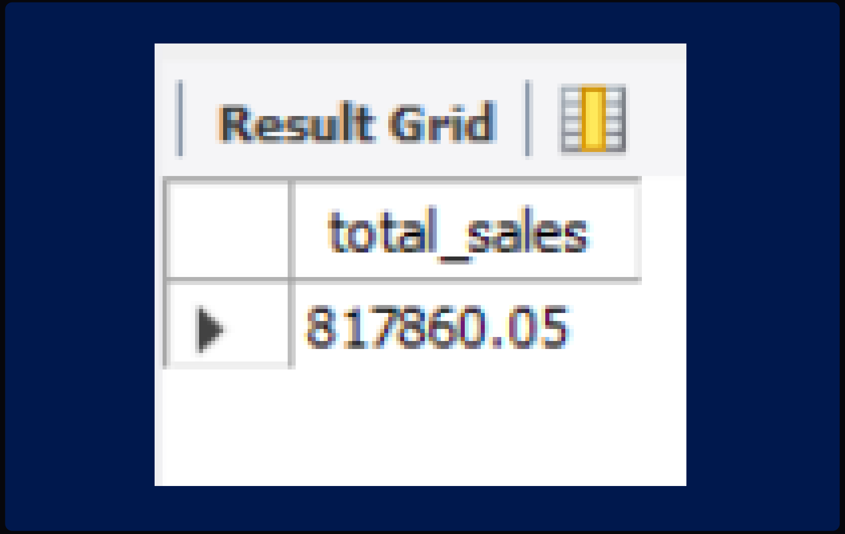
	total_orders
▶	21350

SQL Queries

Calculate the total revenue generated from pizza sales.



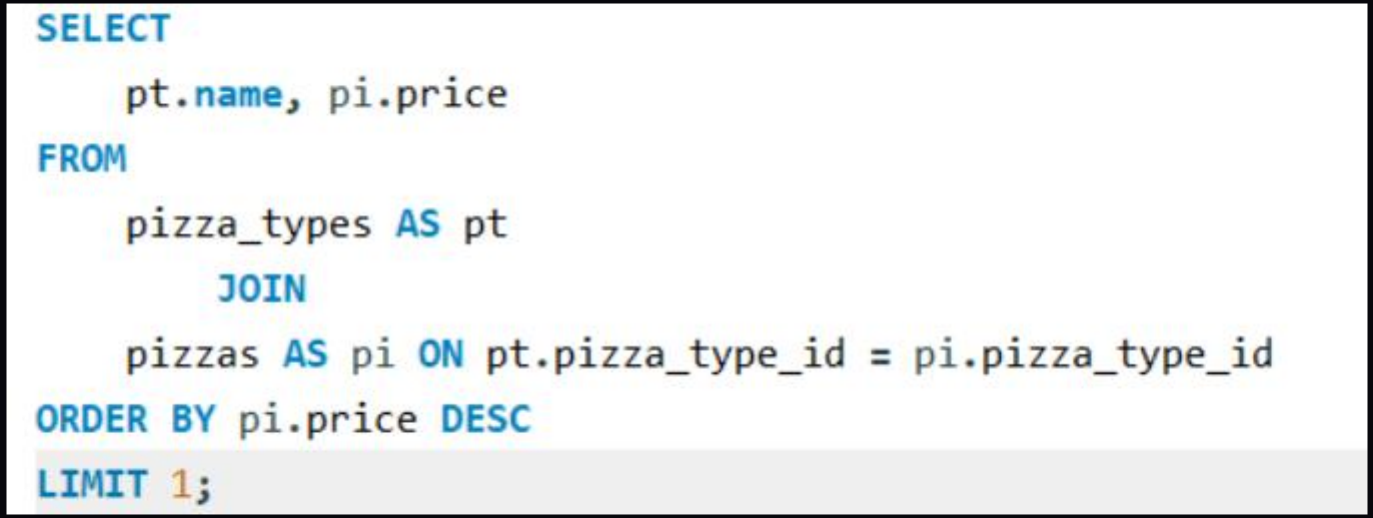
```
SELECT
    ROUND(SUM(ord.quantity * pi.price), 2) AS total_sales
FROM
    order_details AS ord
        JOIN
    pizzas AS pi ON ord.pizza_id = pi.pizza_id
```



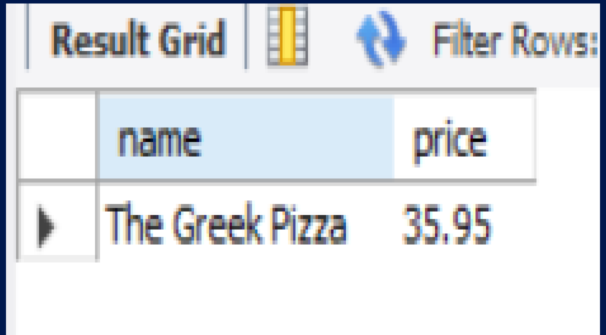
```
total_sales
▶ 817860.05
```

SQL Queries

Identify the highest-priced pizza.



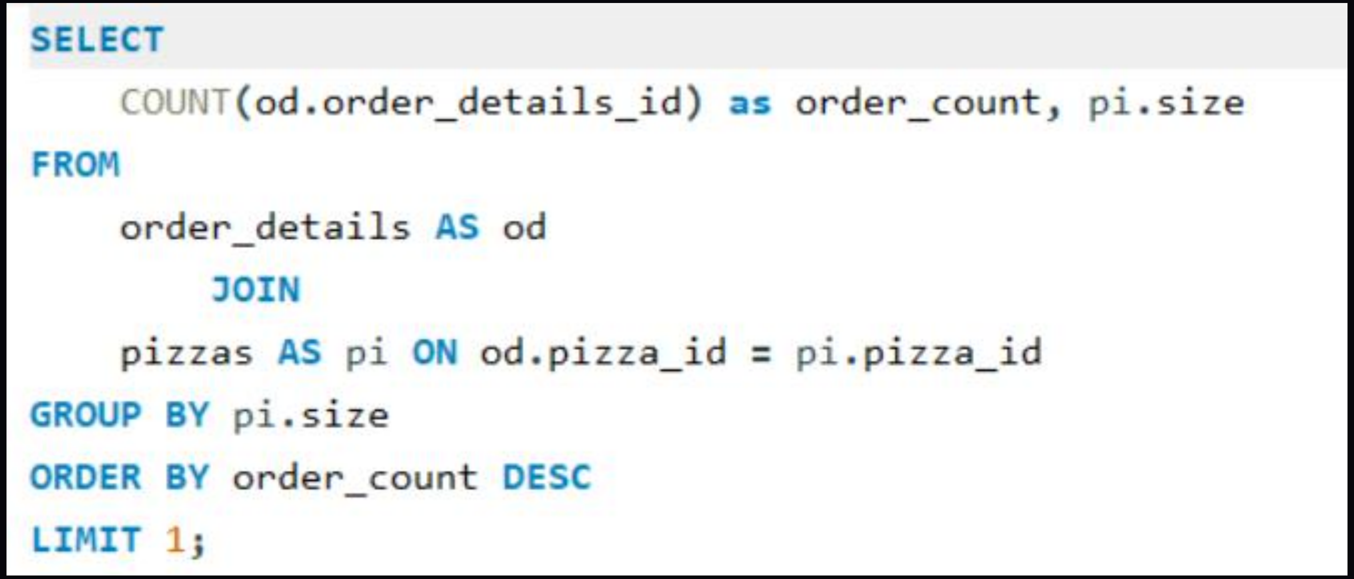
```sql
SELECT
    pt.name, pi.price
FROM
    pizza_types AS pt
        JOIN
    pizzas AS pi ON pt.pizza_type_id = pi.pizza_type_id
ORDER BY pi.price DESC
LIMIT 1;
```



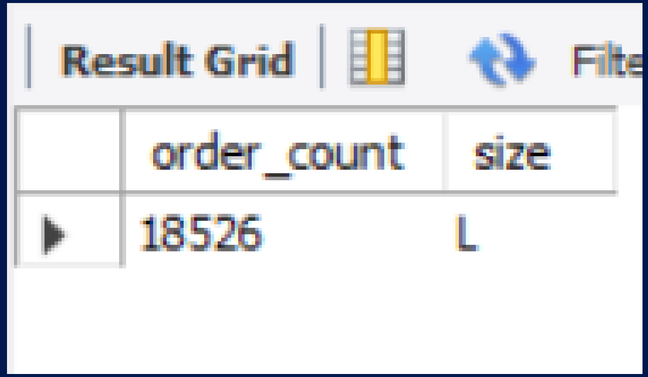
name	price
The Greek Pizza	35.95

SQL Queries

Identify the most common pizza size ordered



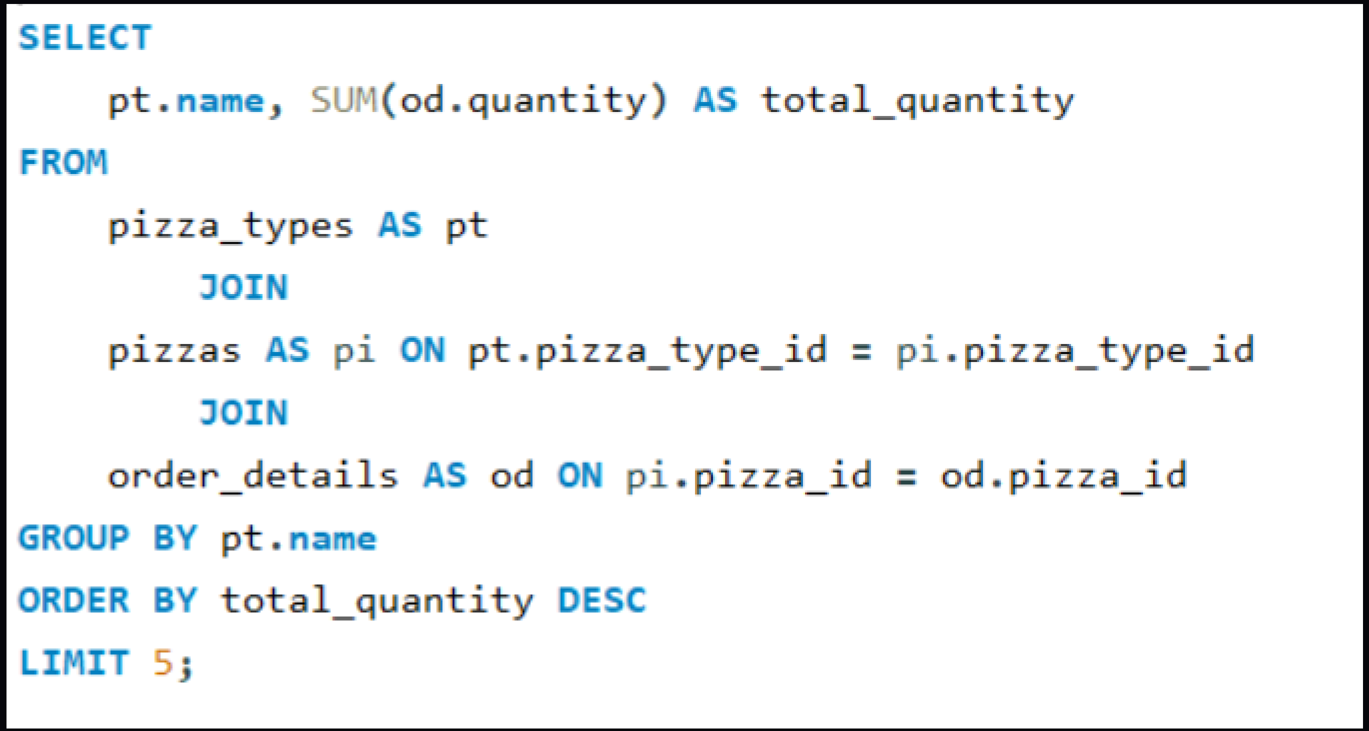
```
SELECT
    COUNT(od.order_details_id) as order_count, pi.size
FROM
    order_details AS od
        JOIN
    pizzas AS pi ON od.pizza_id = pi.pizza_id
GROUP BY pi.size
ORDER BY order_count DESC
LIMIT 1;
```



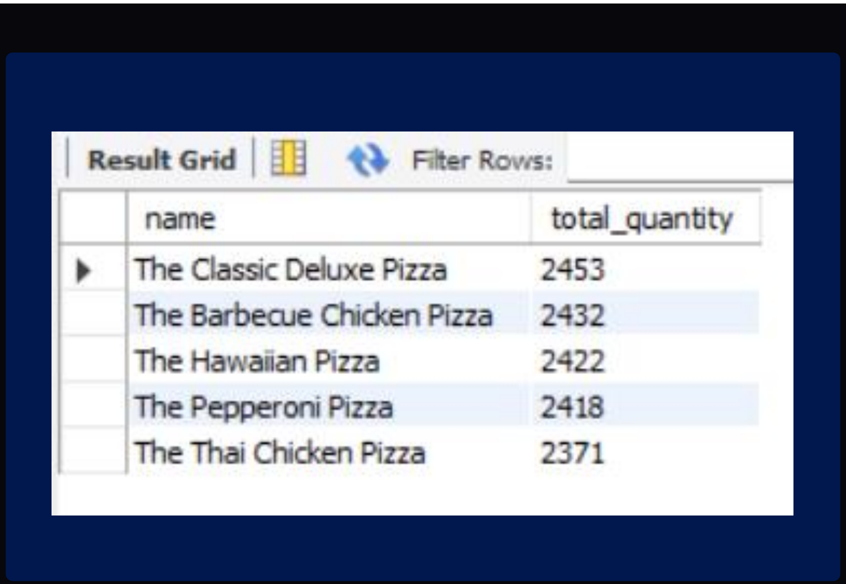
order_count	size
18526	L

SQL Queries

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



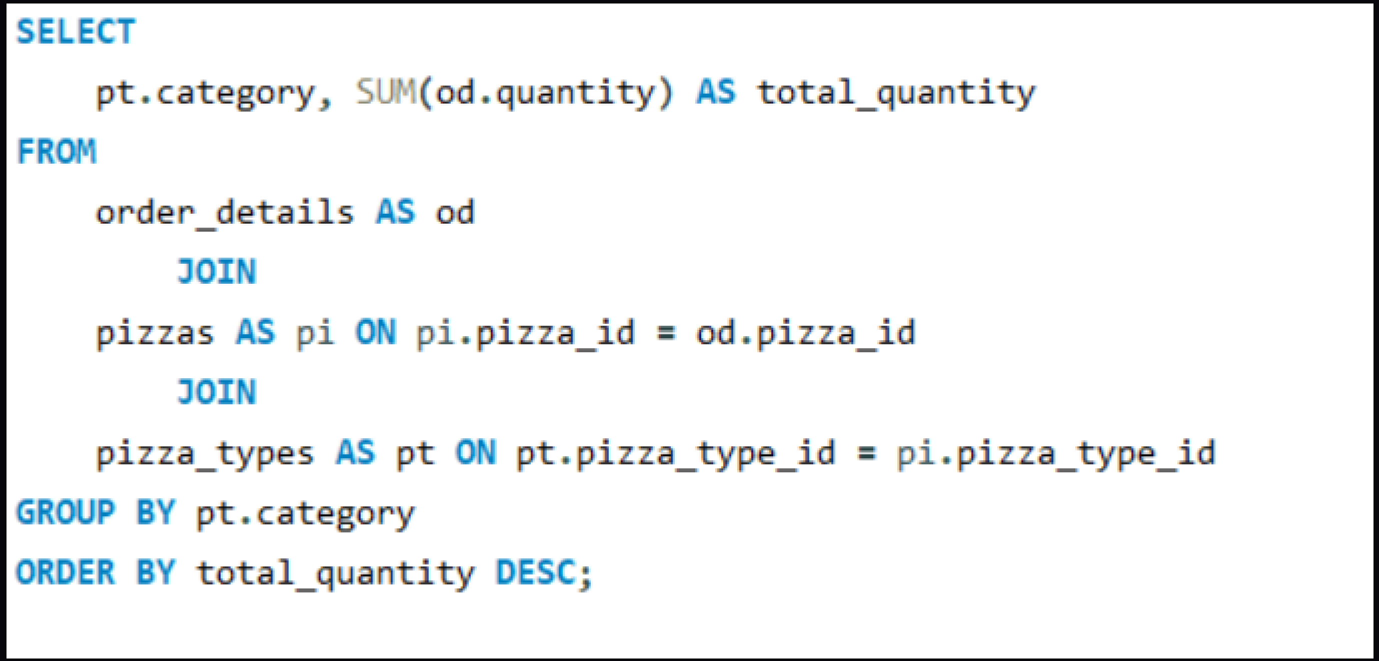
```
SELECT
    pt.name, SUM(od.quantity) AS total_quantity
FROM
    pizza_types AS pt
        JOIN
    pizzas AS pi ON pt.pizza_type_id = pi.pizza_type_id
        JOIN
    order_details AS od ON pi.pizza_id = od.pizza_id
GROUP BY pt.name
ORDER BY total_quantity DESC
LIMIT 5;
```



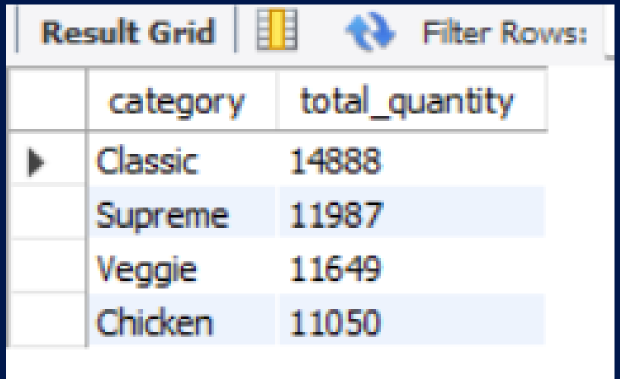
name	total_quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

SQL Queries

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



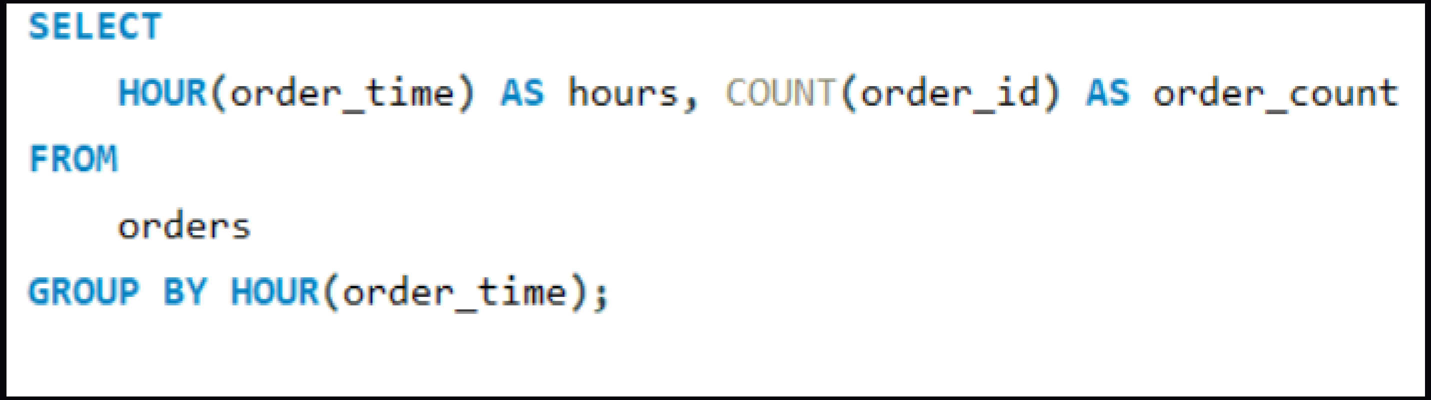
```
SELECT
    pt.category, SUM(od.quantity) AS total_quantity
FROM
    order_details AS od
        JOIN
    pizzas AS pi ON pi.pizza_id = od.pizza_id
        JOIN
    pizza_types AS pt ON pt.pizza_type_id = pi.pizza_type_id
GROUP BY pt.category
ORDER BY total_quantity DESC;
```



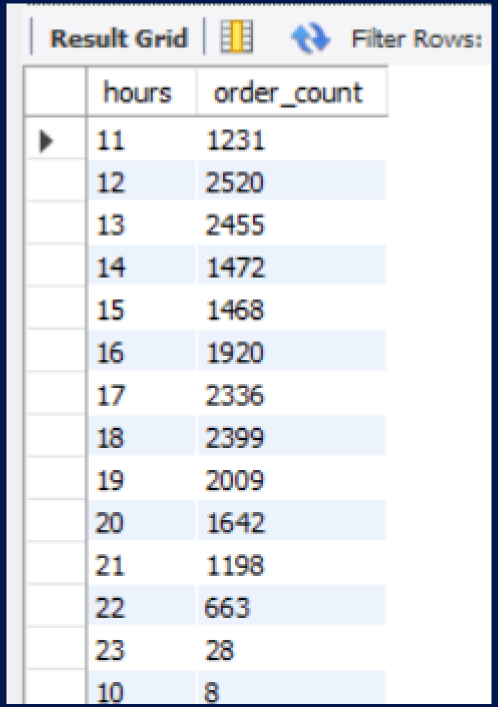
category	total_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

SQL Queries

Determine the distribution of orders by hour of the day.



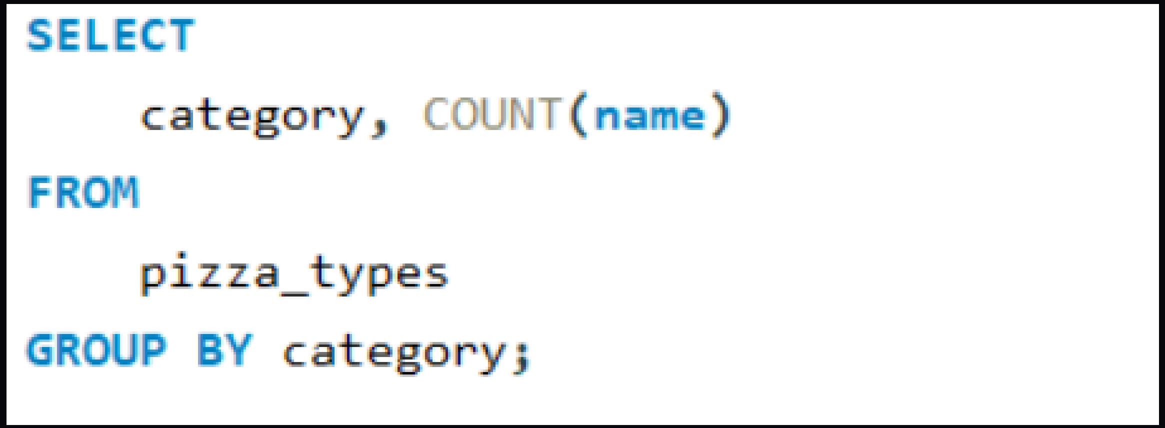
```
SELECT
    HOUR(order_time) AS hours, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time);
```



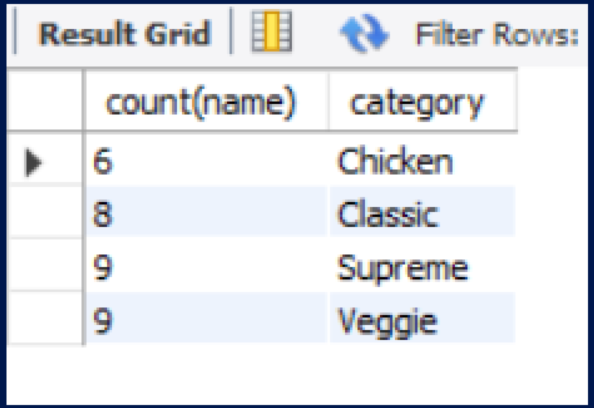
hours	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8

SQL Queries

Join relevant tables to find the category-wise distribution of pizzas.



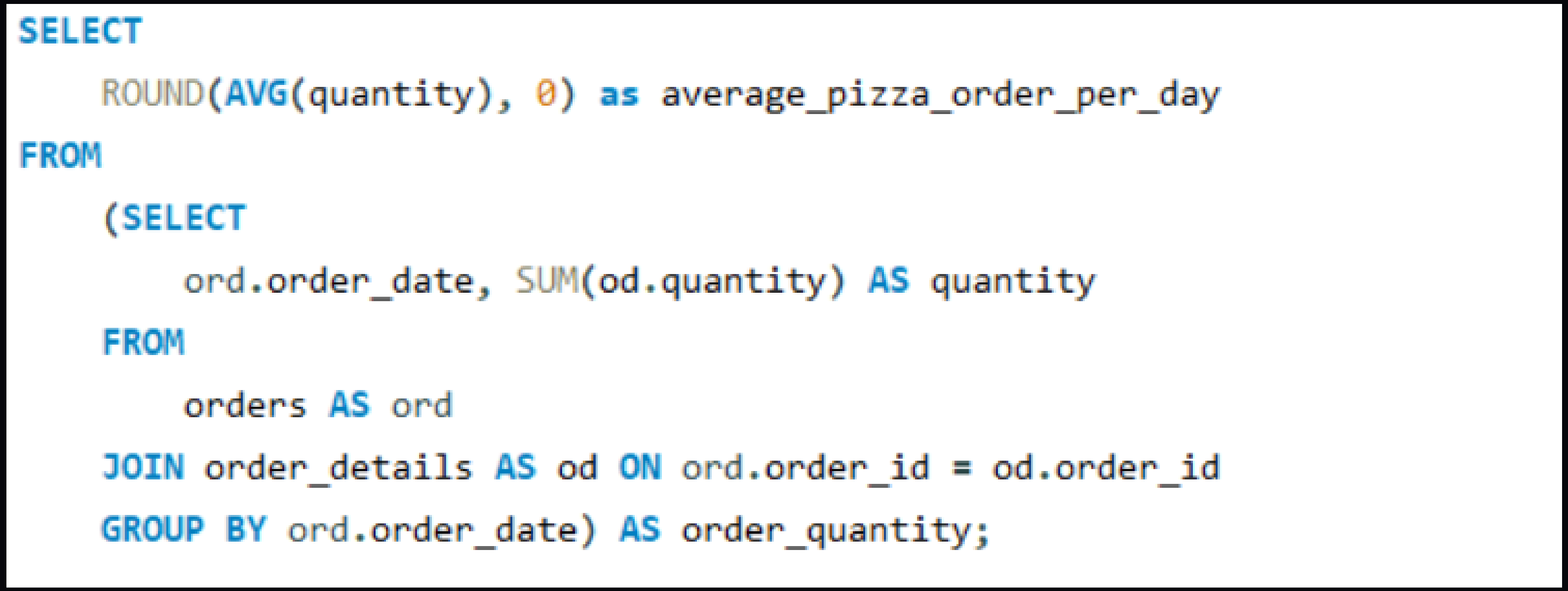
```
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```



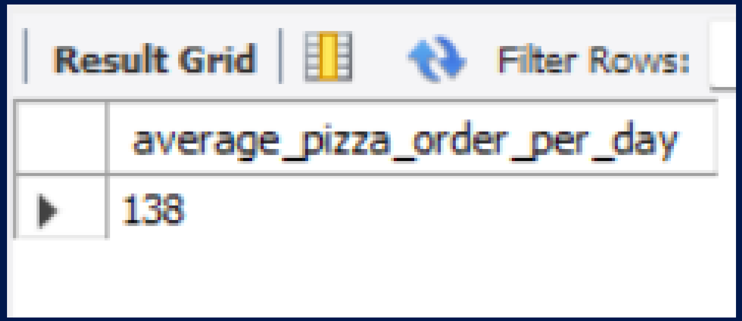
count(name)	category
6	Chicken
8	Classic
9	Supreme
9	Veggie

SQL Queries

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



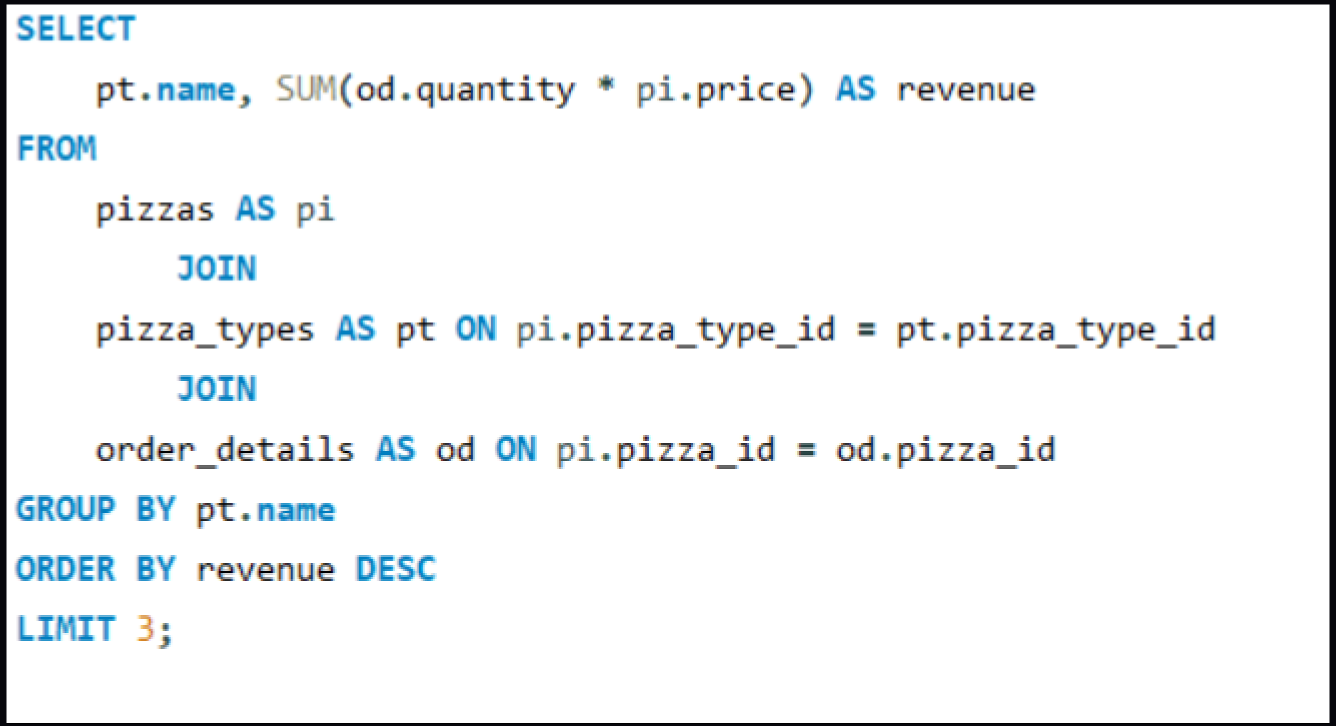
```sql
SELECT
    ROUND(AVG(quantity), 0) as average_pizza_order_per_day
FROM
    (SELECT
        ord.order_date, SUM(od.quantity) AS quantity
    FROM
        orders AS ord
    JOIN order_details AS od ON ord.order_id = od.order_id
    GROUP BY ord.order_date) AS order_quantity;
```



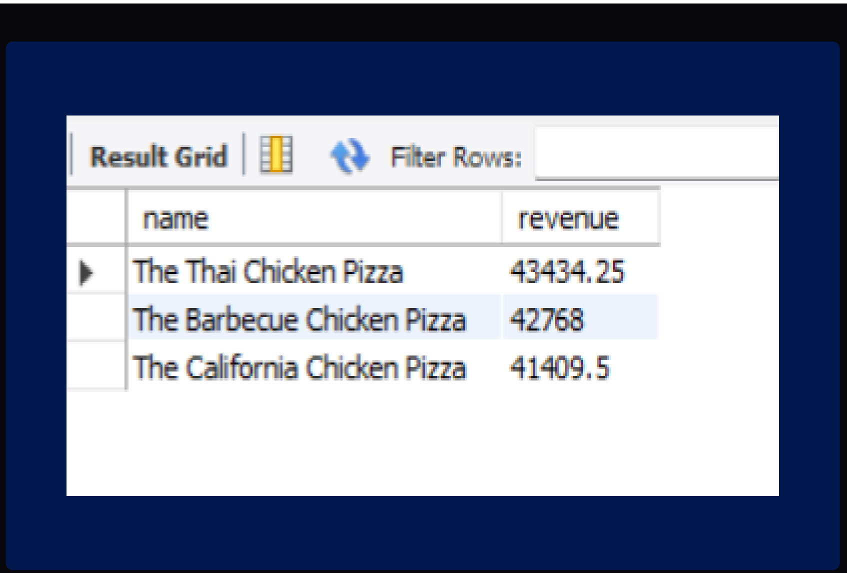
	average_pizza_order_per_day
▶	138

SQL Queries

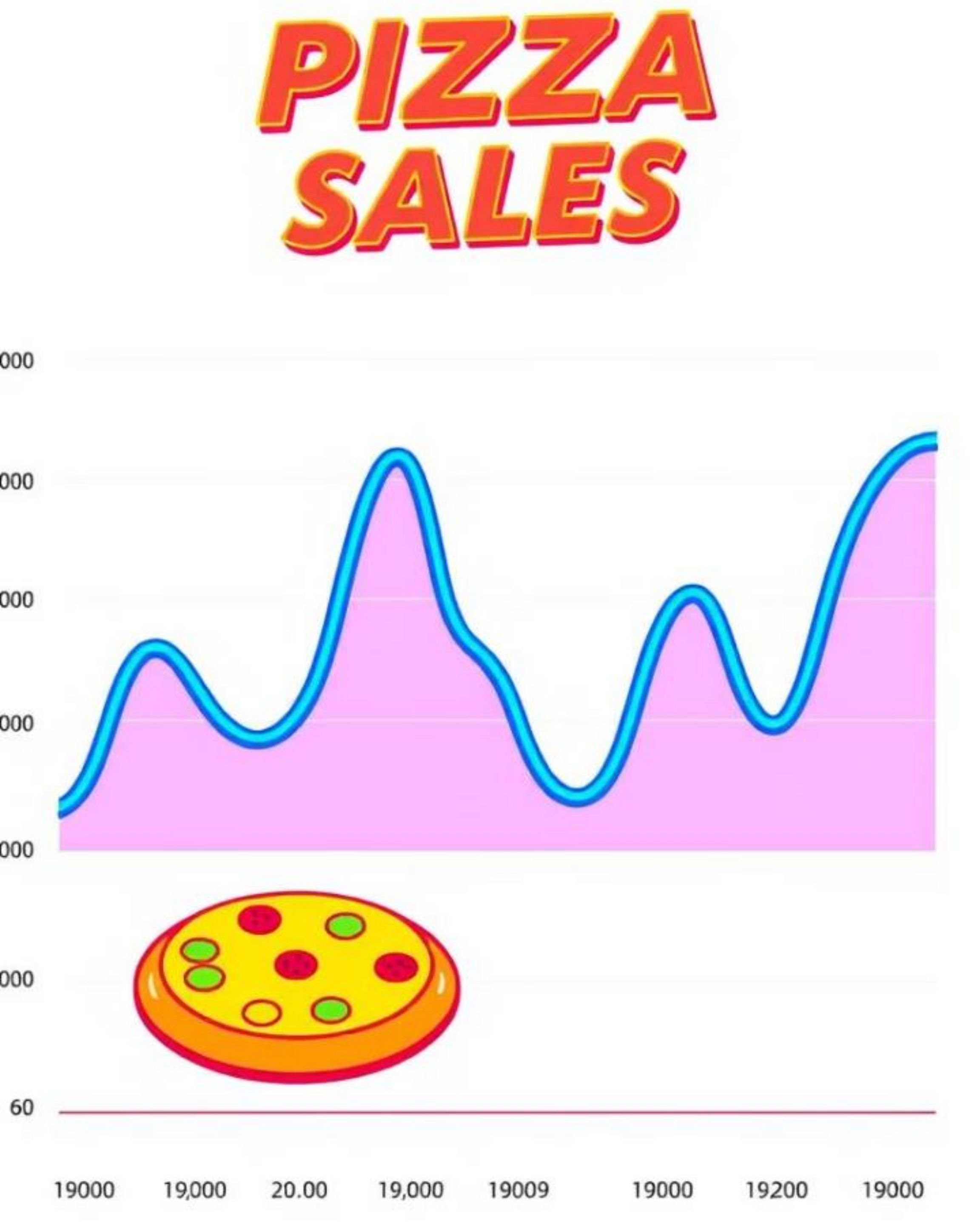
Determine the top 3 most ordered pizza types based on revenue.



```
SELECT
    pt.name, SUM(od.quantity * pi.price) AS revenue
FROM
    pizzas AS pi
        JOIN
    pizza_types AS pt ON pi.pizza_type_id = pt.pizza_type_id
        JOIN
    order_details AS od ON pi.pizza_id = od.pizza_id
GROUP BY pt.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



Analyzing Sales Trends

1

Hourly Trends

Identifying peak Hour for pizza sales in a day.

2

Day to DAY Sales

Analyzing the overall growth of sales over Time.

3

Most Revenue generated Pizza

Determining which Pizza has the highest volume of sales.

Identifying Top-Selling Items

1

Classic

The classic favorite reigns supreme!

2

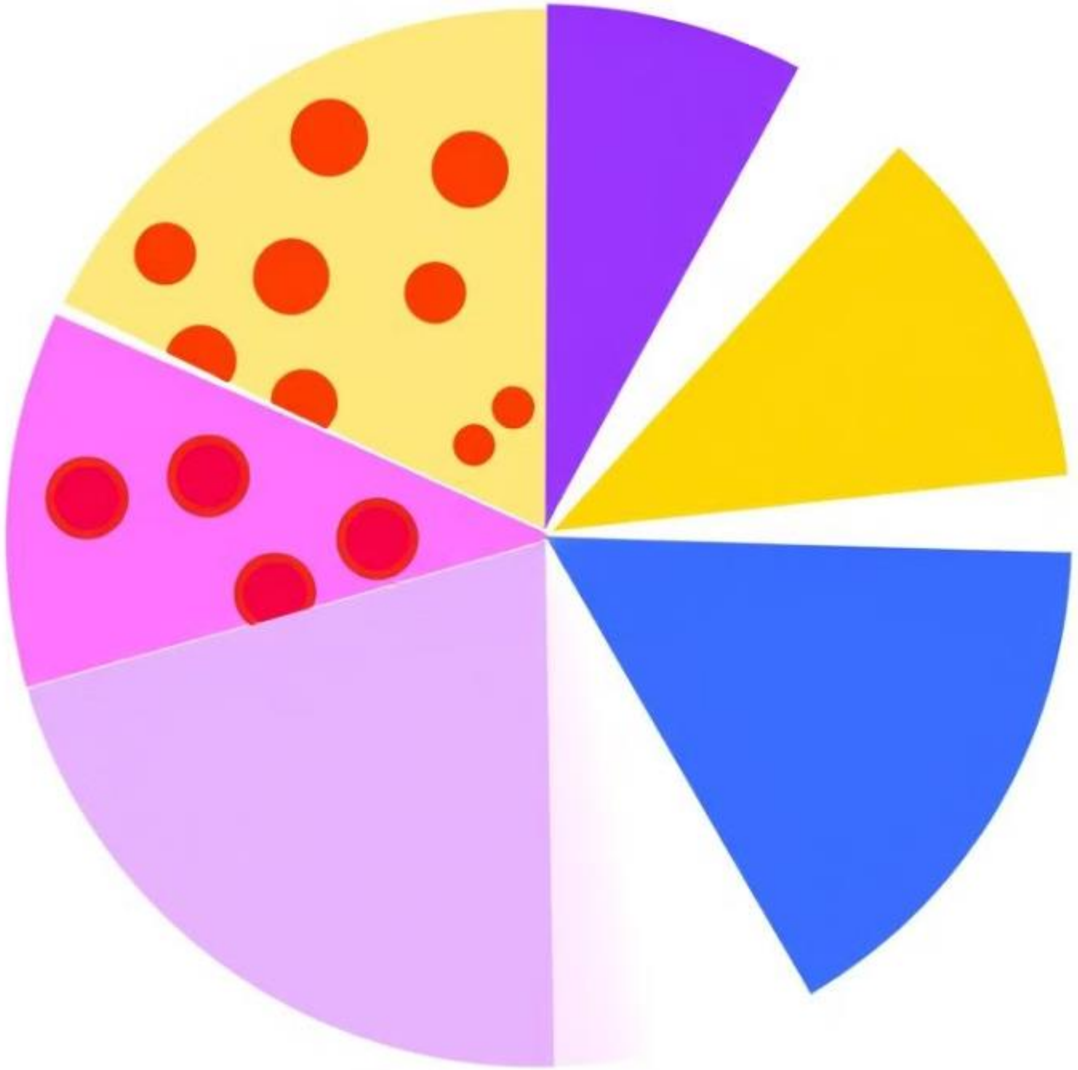
Supreme

A combination of toppings appeals to many.

3

Veggie

Vegetarian options are gaining popularity.



Conclusion :

The SQL project yielded valuable insights into pizza sales trends, customer behavior, and top-selling items. These findings can be used to optimize marketing strategies, inventory management, and overall business operations, leading to increased sales and customer satisfaction.