**Project Name: Pizza Sales Analysis using MySQL**

CREATE DATABASE pizza;

USE pizza;

**TABLES:**

1. **pizza\_types**

CREATE TABLE pizza\_types

(

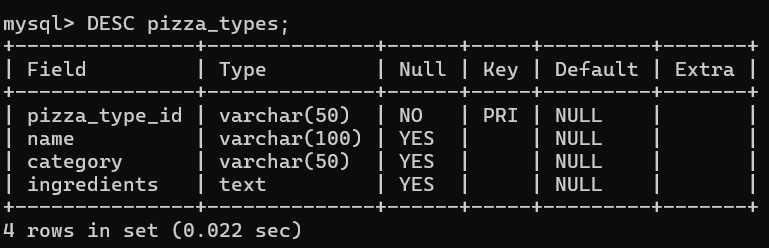
pizza\_type\_id VARCHAR(50) PRIMARY KEY,

name VARCHAR(100),

category VARCHAR(50),

ingredients TEXT

);



1. **pizzas**

CREATE TABLE pizzas

(

pizza\_id VARCHAR(50) PRIMARY KEY,

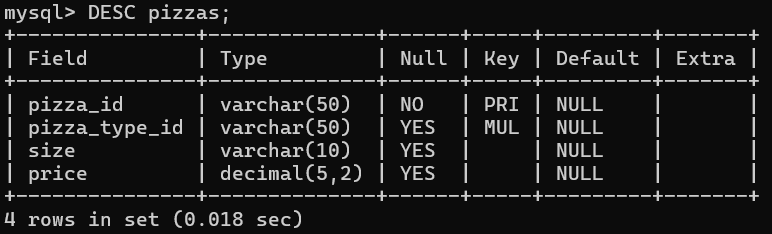
pizza\_type\_id VARCHAR(50),

size VARCHAR(10),

price DECIMAL(5,2),

FOREIGN KEY (pizza\_type\_id) REFERENCES pizza\_types(pizza\_type\_id)

);



1. **orders**

CREATE TABLE orders

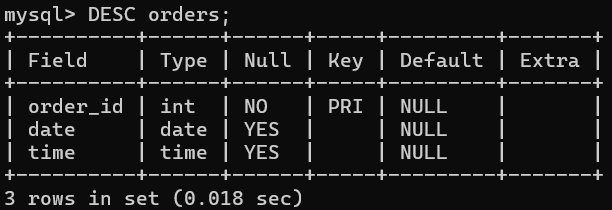
(

order\_id INT PRIMARY KEY,

date DATE,

time TIME

);



1. **order\_details**

CREATE TABLE order\_details

(

order\_details\_id INT PRIMARY KEY,

order\_id INT,

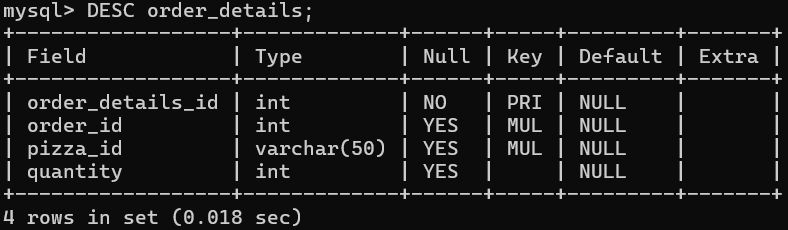
pizza\_id VARCHAR(50),

quantity INT,

FOREIGN KEY (order\_id) REFERENCES orders(order\_id),

FOREIGN KEY (pizza\_id) REFERENCES pizzas(pizza\_id)

);



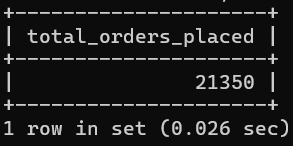
**QUESTIONS:**

**BASIC:**

1. **Retrieve the total number of orders placed.**

SELECT COUNT(order\_id) as total\_orders\_placed

FROM orders;

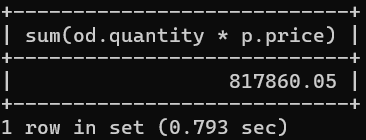


1. **Calculate the total revenue generated from pizza sales.**

SELECT sum(od.quantity \* p.price) from order\_details od

INNER JOIN pizzas p

ON od.pizza\_id=p.pizza\_id;



1. **Identify the highest-priced pizza.**

SELECT pt.name AS pizza\_name ,

max(p.price) AS highest\_price

from pizzas p

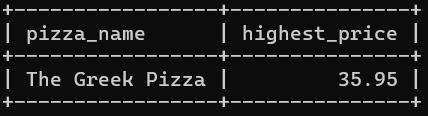
join pizza\_types pt

on p.pizza\_type\_id = pt.pizza\_type\_id

group by pt.name

order by highest\_price desc

LIMIT 1;



1. **Identify the most common pizza size ordered.**

SELECT p.size,

SUM(od.quantity) AS total\_order

FROM order\_details od

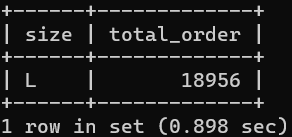
join pizzas p

ON od.pizza\_id = p.pizza\_id

GROUP BY p.size

ORDER BY total\_order DESC

LIMIT 1;



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

SELECT pt.name,

SUM(od.quantity) AS total\_order

FROM order\_details od

join pizzas p

ON od.pizza\_id = p.pizza\_id

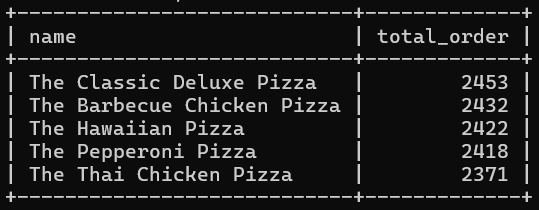
JOIN pizza\_types pt

ON p.pizza\_type\_id = pt.pizza\_type\_id

GROUP BY pt.name

ORDER BY total\_order DESC

LIMIT 5;



**INTERMEDIATE:**

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

SELECT

pt.category AS pizza\_category,

SUM(od.quantity) AS total\_quantity

FROM order\_details od

JOIN pizzas p

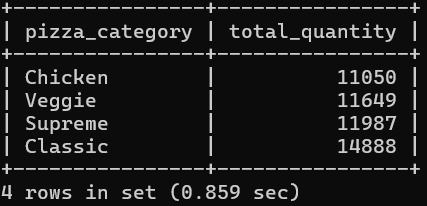
ON od.pizza\_id = p.pizza\_id

JOIN pizza\_types pt

ON p.pizza\_type\_id = pt.pizza\_type\_id

GROUP BY pt.category

ORDER BY total\_quantity;



1. **Determine the distribution of orders by hour of the day.**

SELECT

HOUR(time) AS order\_hour,

COUNT(\*) AS total\_orders

FROM

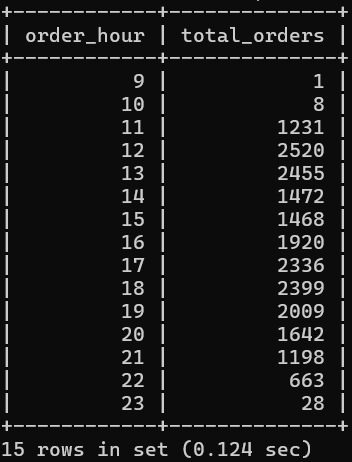
orders

GROUP BY

order\_hour

ORDER BY

order\_hour;



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

SELECT

pt.category,

COUNT(p.pizza\_id) AS total\_pizzas

FROM

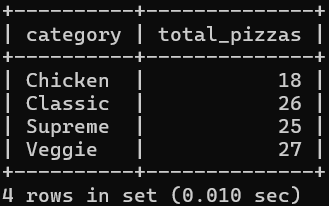
pizzas p

JOIN

pizza\_types pt ON p.pizza\_type\_id = pt.pizza\_type\_id

GROUP BY

pt.category;



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

SELECT

o.date,

SUM(od.quantity) AS total\_pizzas\_ordered

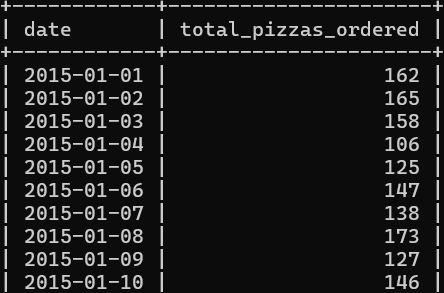
FROM

orders o

JOIN

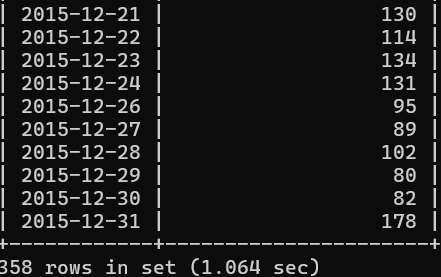
order\_details od ON o.order\_id = od.order\_id

GROUP BY

 o.date

ORDER BY

o.date;



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

SELECT

pt.name AS pizza\_name,

SUM(od.quantity \* p.price) AS total\_revenue

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

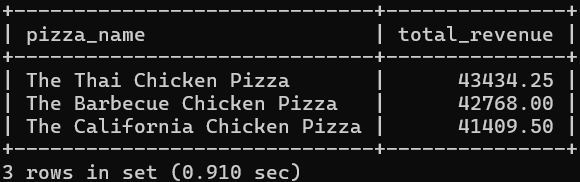
GROUP BY

pt.name

ORDER BY

total\_revenue DESC

LIMIT 3;



**ADVANCED:**

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

SELECT

pt.name AS pizza\_name,

SUM(od.quantity \* p.price) AS revenue,

ROUND(SUM(od.quantity \* p.price) / SUM(SUM(od.quantity \* p.price)) OVER () \* 100, 2) AS percentage\_of\_total\_revenue

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

percentage\_of\_total\_revenue DESC;



1. **Analyze the cumulative revenue generated over time.**

SELECT

o.date,

SUM(od.quantity \* p.price) AS daily\_revenue,

SUM(SUM(od.quantity \* p.price)) OVER (ORDER BY o.date) AS cumulative\_revenue

FROM

orders o

JOIN

order\_details od ON o.order\_id = od.order\_id

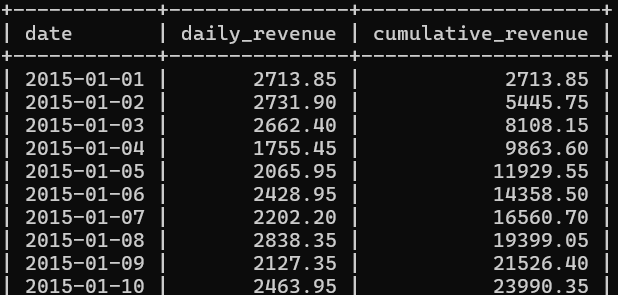
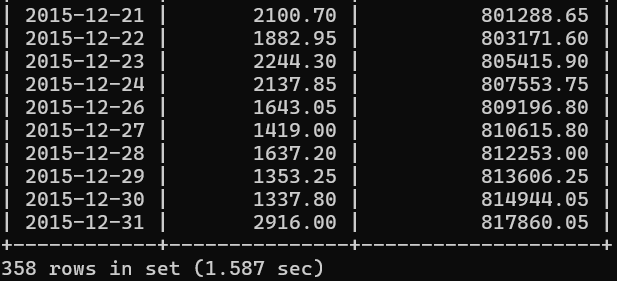
JOIN

pizzas p ON od.pizza\_id = p.pizza\_id

GROUP BY

o.date

ORDER BY

 o.date;

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

WITH revenue\_per\_pizza AS (

SELECT

pt.category,

pt.name AS pizza\_name,

SUM(od.quantity \* p.price) AS revenue

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, pt.name

)

SELECT

category,

pizza\_name,

revenue

FROM (

SELECT

\*,

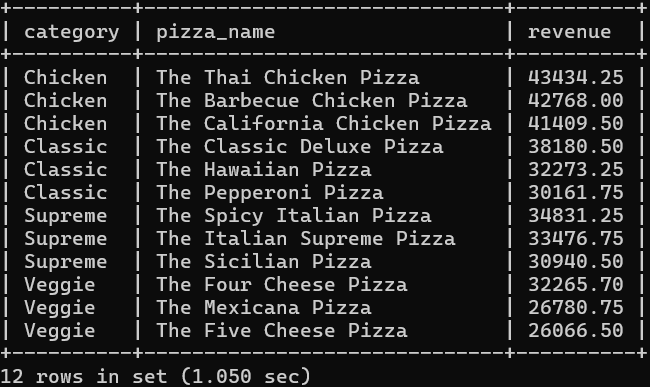
ROW\_NUMBER() OVER (PARTITION BY category ORDER BY revenue DESC) AS rank\_in\_category

FROM

revenue\_per\_pizza

) ranked

WHERE rank\_in\_category <= 3

ORDER BY category, revenue DESC;