





Hello, My name is MONIKA NAIKAL.

In this project i have utilised MYSQL Queries to solve question that where related to PIZZA_SALES

DATABASE SCHEMA

This SQL project revolves around a database schema designed to manage and analyze data for a pizza store. The database consists of four primary tables

- order_details
- pizzas
- orders
- pizza_types.

SCHEMA DETAILS detailed description of each table and its columns

1. order_details:

- order_details_id: A unique identifier for each entry in the order details.

- order_id: References the ID from the orders table, linking the order detail to a specific order.

- pizza_id: References the ID from the pizzas table, identifying which pizza was ordered.

- quantity: The number of pizzas ordered of the specified type.

2. pizzas:

- pizza_id: A unique identifier for each type of pizza available.
- pizza_type_id: Links to the pizza_types table, specifying the type of pizza.
- size: The size of the pizza (e.g., small, medium, large).
- price: The cost of the pizza.

3. orders:

- order_id: A unique identifier for each order placed.
- date: The date on which the order was placed.
- time: The time at which the order was placed.

4. pizza_types:

- pizza_type_id: A unique identifier for each type of pizza.
- name: The name of the pizza type (e.g., Margherita, Pepperoni).
- category: Categorizes the pizza (e.g., Vegetarian, Non-Vegetarian).
- ingredients: Lists the ingredients used in the pizza.



Q1. RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

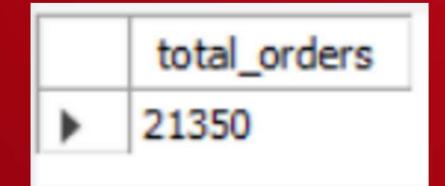
INPUT

SELECT

COUNT(order_ID) AS total_orders

FROM

orders;





Q2. CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

INPUT

```
select
sum(order_details.quantity * pizzas.price) as total_sales
from order_details join pizzas
on pizzas.pizza_id=order_details.pizza_id
;
```

1	I believed	
	total_sales	
•	817860.049999993	



Q3. IDENTIFY THE HIGHEST PRICED PIZZA..

INPUT

```
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



Q4. LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES...

INPUT

```
SELECT pizza_types.name,

SUM(order_details.quantity) AS total_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 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

Q5. IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.



INPUT

```
select pizzas.size, count(order_details.order_details_id) as order_count
from pizzas join order_details
ON pizzas.pizza_id = order_details.pizza_id
group by pizzas.size order by order_count desc;
```

	size	order_count
Þ	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28



QB. JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.



INPUT

```
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 -- Corrected this join condition

JOIN order_details

ON pizzas.pizza_id = order_details.pizza_id -- Added this join for order_details

GROUP BY pizza_types.category

ORDER BY quantity DESC;
```

	category	quantity
•	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



Q7. DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.



INPUT

```
SELECT HOUR(time) AS order_hour,

COUNT(order_id) AS order_count

FROM orders

GROUP BY order_hour

ORDER BY order_hour;
```

	order_hour	order_count
•	9	1
	10	8
	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

Q8. JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS

INPUT

```
SELECT pizza_types.category,

COUNT(order_details.pizza_id) AS pizza_count

FROM pizza_types

JOIN pizzas

ON pizza_types.pizza_type_id = pizzas.pizza_type_id

JOIN order_details

ON pizzas.pizza_id = order_details.pizza_id

GROUP BY pizza_types.category

ORDER BY pizza_count DESC;
```



	category	pizza_count
•	Classic	14579
	Supreme	11777
	Veggie	11449
	Chicken	10815

Q9. GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY...

INPUT

```
SELECT
   ROUND(AVG(quantity), 0) AS avg_pizza_ordered_per_day
FROM
   (SELECT
       orders.date AS order_date,
       SUM(order_details.quantity) AS quantity
    FROM
       orders
    JOIN
       order_details ON orders.order_id = order_details.order_id
    GROUP BY
       orders.date) AS order_quantity;
```

```
avg_pizza_ordered_per_day

138
```

Q10. DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

INPUT

	pizza_type	category	total_revenue
•	The Thai Chicken Pizza	Chicken	43434.25
	The Barbecue Chicken Pizza	Chicken	42768
	The California Chicken Pizza	Chicken	41409.5

#