

Where Every data is important for Perfection

WELCOME TO PIZZA SALES PROJECT

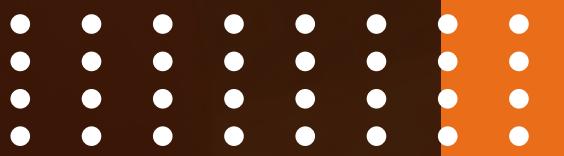




ABOUT OUR PROJECT

- Objective of the Analysis: Understanding sales trends, revenue, order patterns, and top-performing pizzas.
- Dataset Overview: Orders, order details, pizzas, and pizza types tables.
- Key attributes like order ID, pizza size, category, price, and quantity.
- Tech Stack Used: SQL





RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
SELECT  
    COUNT(order_id) AS total_orders  
FROM  
    orders;
```

Result Grid	
	total_orders
▶	21350



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 order_details.pizza_id = pizzas.pizza_id;
```

Result Grid	
	total_sales
▶	817860.05



IDENTIFY THE HIGHEST-PRICED PIZZA.

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

Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

SELECT

```
COUNT(order_details.quantity) AS order_cont, pizzas.size
```

FROM

```
order_details
```

JOIN

```
pizzas ON order_details.pizza_id = pizzas.pizza_id
```

GROUP BY pizzas.size

ORDER BY order_cont **DESC**;

	order_cont	size
▶	18526	L
	15385	M
	14137	S
	544	XL
	28	XXL

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

SELECT

```
    pizza_types.name AS pizza_name,  
    COUNT(order_details.quantity) AS total_orders
```

FROM

```
pizzas
```

JOIN

```
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
```

JOIN

```
    order_details ON pizzas.pizza_id = order_details.pizza_id
```

GROUP BY pizza_name

ORDER BY total_orders **DESC**

LIMIT 5;

	pizza_name	total_orders
▶	The Classic Deluxe Pizza	2416
	The Barbecue Chicken Pizza	2372
	The Hawaiian Pizza	2370
	The Pepperoni Pizza	2369
	The Thai Chicken Pizza	2315

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

SELECT

```
pizza_types.category AS Category,  
SUM(order_details.quantity) AS total_orders
```

FROM

```
pizzas
```

JOIN

```
pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
```

JOIN

```
order_details ON pizzas.pizza_id = order_details.pizza_id
```

GROUP BY Category

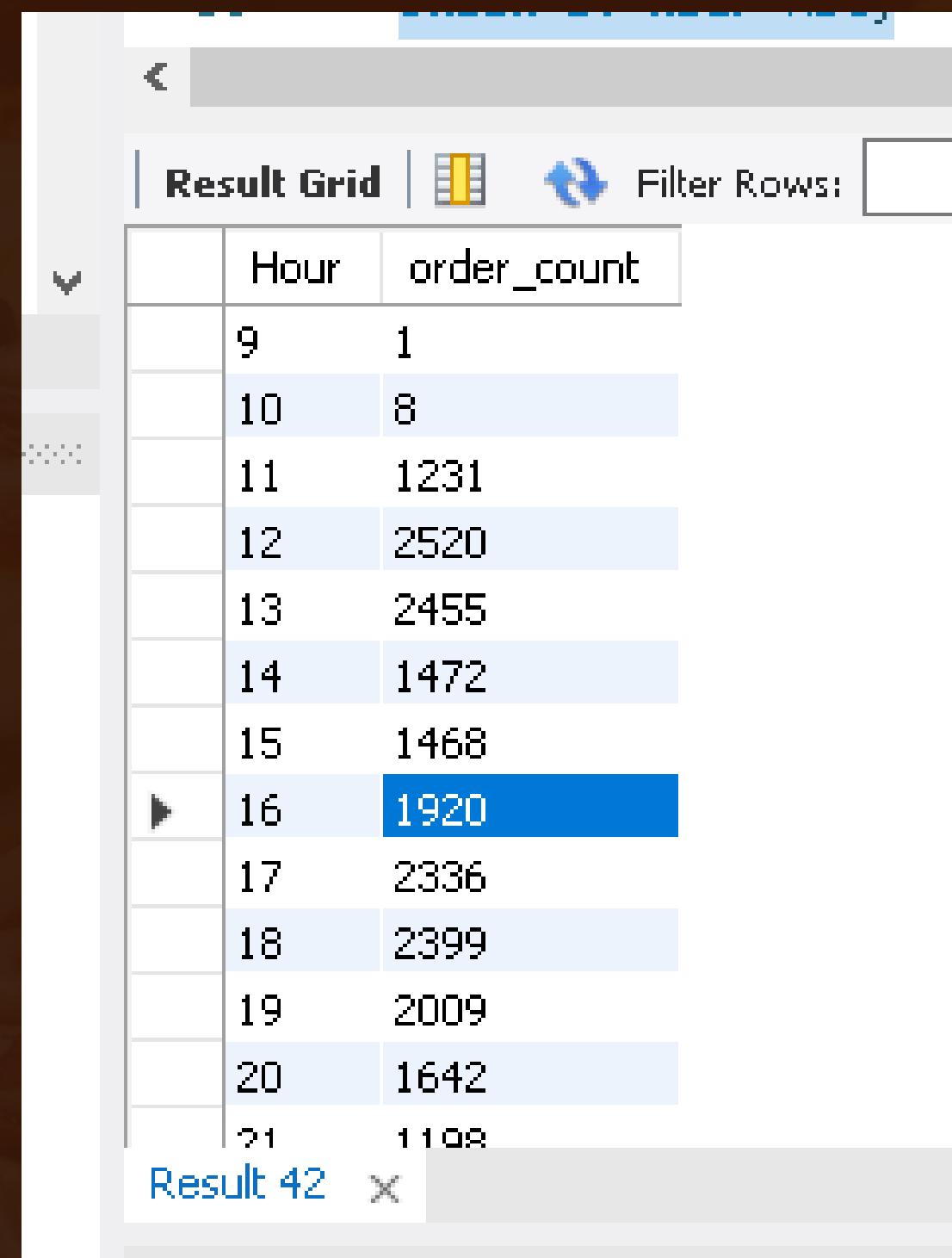
ORDER BY total_orders **DESC**;

Result Grid | Filter Rows:

	Category	total_orders
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```
SELECT  
    HOUR(orders.order_time) AS Hour,  
    COUNT(order_id) AS order_count  
FROM  
    orders  
GROUP BY Hour  
ORDER BY Hour ASC;
```



The screenshot shows a MySQL Workbench result grid titled "Result Grid". The grid displays the distribution of orders by hour of the day. The columns are "Hour" and "order_count". The data shows that the highest number of orders was at hour 12, with 2520 orders. The count decreases as the hour progresses, with the lowest count at hour 9 (1 order). The row for hour 16 is highlighted with a blue background.

	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	1108

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

SELECT

```
    pizza_types.category AS Category,  
    COUNT(pizza_types.name) AS pizzas_count
```

FROM

```
    pizza_types
```

GROUP BY Category;

Result Grid | Filter Rows

	Category	pizzas_count
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

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

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

Result Grid	
	ROUND(AVG(Quantity_count), 0)
▶	138

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

```
SELECT  
    pizza_types.name AS NAmE,  
    SUM(order_details.quantity * pizzas.price) AS pizza_revenue  
FROM  
    pizzas  
        JOIN  
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id  
        JOIN  
    order_details ON pizzas.pizza_id = order_details.pizza_id  
GROUP BY NAmE  
ORDER BY pizza_revenue DESC  
LIMIT 3;
```

Result Grid		
	NAmE	pizza_revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5

PIZZA SALES SQL Presentation

THANK YOU
FOR ATTENTION