

PAPA JOHN'S PIZZA



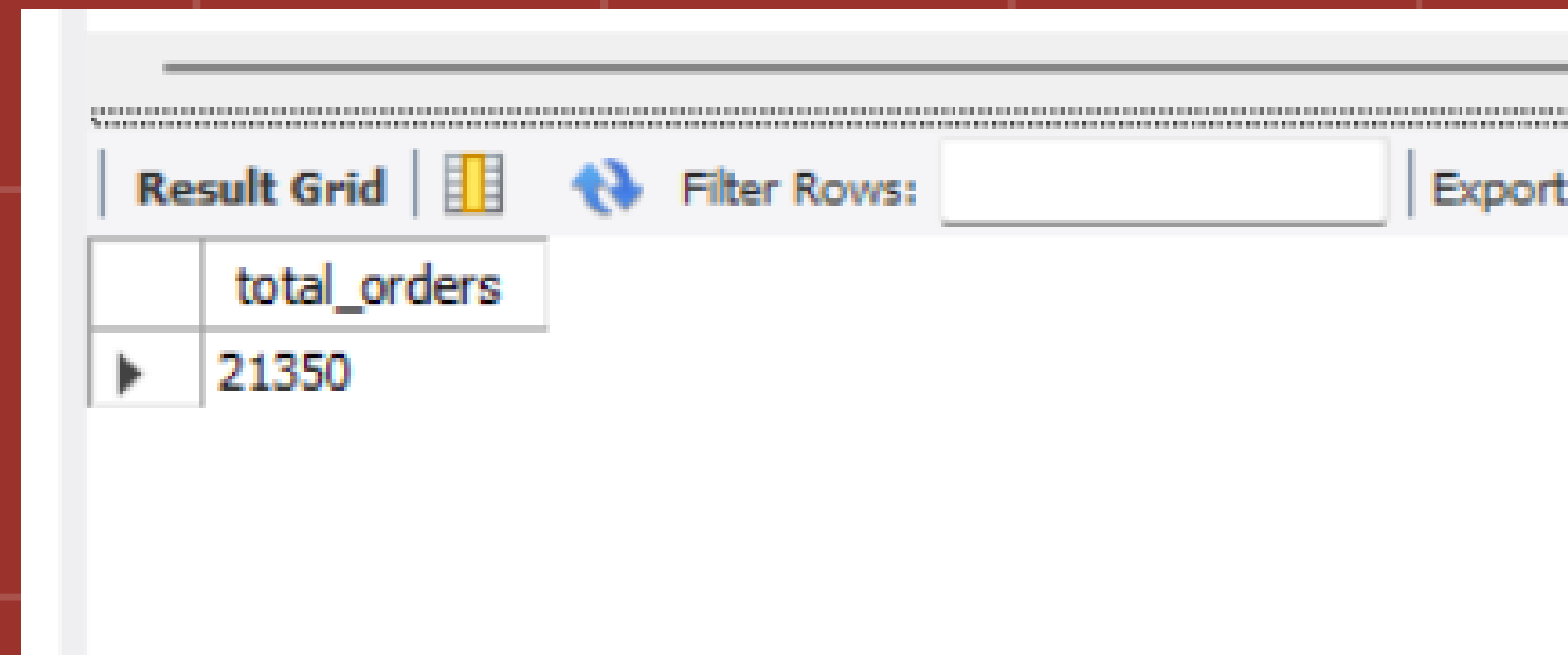


INTRODUCTION

Hi, My name is Shubham Singh. In this project, I have analyzed pizza sales data using MySQL to uncover valuable insights that can drive better business decisions. I worked with extensive datasets, focusing on key metrics like sales trends and customer preferences. By crafting complex queries, I was able to identify patterns in sales performance, determine the most popular pizza varieties, and highlight areas for improvement. This project demonstrates my proficiency in using MySQL to manage and analyze data, ultimately helping to optimize operations and enhance the overall sales strategy.

PROBLEM STATEMENT: RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

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





The screenshot shows a database query result interface. At the top, there's a header bar with 'Result Grid', a table icon, a 'Filter Rows' button with a double arrow icon, and an 'Export' button. Below this is a table with one column named 'total_orders' and one row containing the value '21350'.

	total_orders
▶	21350

PROBLEM STATEMENT: CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES



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

Result Grid |   Filter Rows

	total_revanue
▶	817860.05

PROBLEM STATEMENT: IDENTIFY THE HIGHEST-PRICED PIZZA



```
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;
```

Result Grid |   Filter Rows:

	name	price
▶	The Greek Pizza	35.95

PROBLEM STATEMENT: IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED



```
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
LIMIT 1;
```

Result Grid |   Filter Rows:

	size	order_count
▶	L	18526



PROBLEM STATEMENT: LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES

```
select pizza_types.name,  
sum(order_details.quantity) as 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 quantity desc limit 5;
```

Result Grid   Filter Rows:

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

PROBLEM STATEMENT: JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED



Result Grid |   Filter R

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

```
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  
join order_details  
on order_details.pizza_id=pizzas.pizza_id  
group by pizza_types.category order by quantity desc;
```


PROBLEM STATEMENT: DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

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

Result Grid |   Filter Rows

	hour	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
	9	1

PROBLEM STATEMENT: JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS



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

Result Grid |   Filter Rows:

	category	COUNT(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

PROBLEM STATEMENT: CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE

```
select pizza_types.category,  
round((sum(order_details.quantity*pizzas.price)/(SELECT  
    ROUND(SUM(order_details.quantity * pizzas.price),  
        2) AS total_revenue  
FROM  
    order_details  
    JOIN  
    pizzas ON pizzas.pizza_id = order_details.pizza_id))*100,0) as revenue  
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.category order by revenue desc;
```

Result Grid |   Filter Rows

	category	revenue
▶	Classic	27
	Supreme	25
	Veggie	24
	Chicken	24

PROBLEM STATEMENT: ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

```
select order_date,  
sum(revenue) over (order by order_date) as cum_revenue  
from  
(select orders.order_date,  
sum(order_details.quantity*pizzas.price) as revenue  
from order_details join pizzas  
on order_details.pizza_id = pizzas.pizza_id  
join orders  
on orders.order_id = order_details.order_id  
group by orders.order_date) as sales;
```

Result Grid			Filter Rows:
	order_date	cum_revenue	
▶	2015-01-01	2713.8500000000004	
	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	
	2015-01-07	16560.7	
	2015-01-08	19399.05	
	2015-01-09	21526.4	
	2015-01-10	23990.350000000002	
	2015-01-11	25862.65	
	2015-01-12	27781.7	
	2015-01-13	29831.300000000003	
	2015-01-14	32358.700000000004	
	2015-01-15	34343.500000000001	
	2015-01-16	36937.650000000001	

PROBLEM STATEMENT: DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY

```
select name, revenue from
(select category, name, revenue,
rank() over(partition by category order by revenue desc) as rn
from
(select pizza_types.category, pizza_types.name,
sum((order_details.quantity)
* pizzas. price) as revenue
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.category, pizza_types.name) as a) as b
where rn <= 3;
```

Result Grid | Filter Rows: | Export

	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5
	The Classic Deluxe Pizza	38180.5
	The Hawaiian Pizza	32273.25
	The Pepperoni Pizza	30161.75
	The Spicy Italian Pizza	34831.25
	The Italian Supreme Pizza	33476.75
	The Sicilian Pizza	30940.5
	The Four Cheese Pizza	32265.70000000065
	The Mexicana Pizza	26780.75
	The Five Cheese Pizza	26066.5



THANK YOU