



# PIZZA SALES - DATA ANALYSIS USING MYSQL

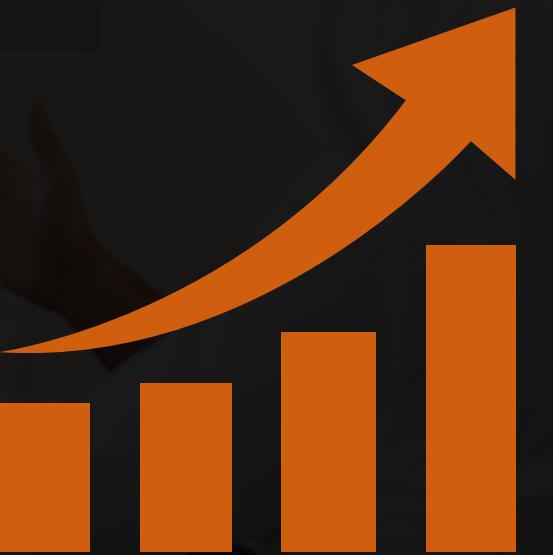
presented by  
**Sharjeel Ali Momin**





# DATA ANALYSIS

Data analysis is the process of systematically applying statistical and logical techniques to describe, illustrate, condense, and evaluate data. It encompasses a variety of methods to extract useful information from raw data, uncover patterns, and draw conclusions that help in decision-making.



**presented by**  
**Sharjeel Ali Momin**



# SUMMARY

We are Analysing the data of PizzaHut sales 2015, In MySQL using different queries and patterns to evaluate the data and bring out the meaningful insights from the data by solving the set of given questions.



# RETRIEVE TOTAL NUMBER OF ORDER 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 pizza_types join pizzas  
on pizza_types.pizza_type_id = pizzas.pizza_type_id  
order by pizzas.price desc limit 1;
```

Result Grid | Filter Row

	name	price
▶	The Greek Pizza	35.95

# JOIN NECESSARY TABLES TO FIND OUT TOTAL QUANTITY OF EACH PIZZA CATEGORY.

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;
```

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

# TOP 5 MOST ORDERED PIZZA TYPE ALONG WITH THEIR QUANTITY

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;



	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

# DETERMINE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

SELECT

HOUR(order\_time) AS hour, COUNT(order\_id) AS order\_count

FROM

orders

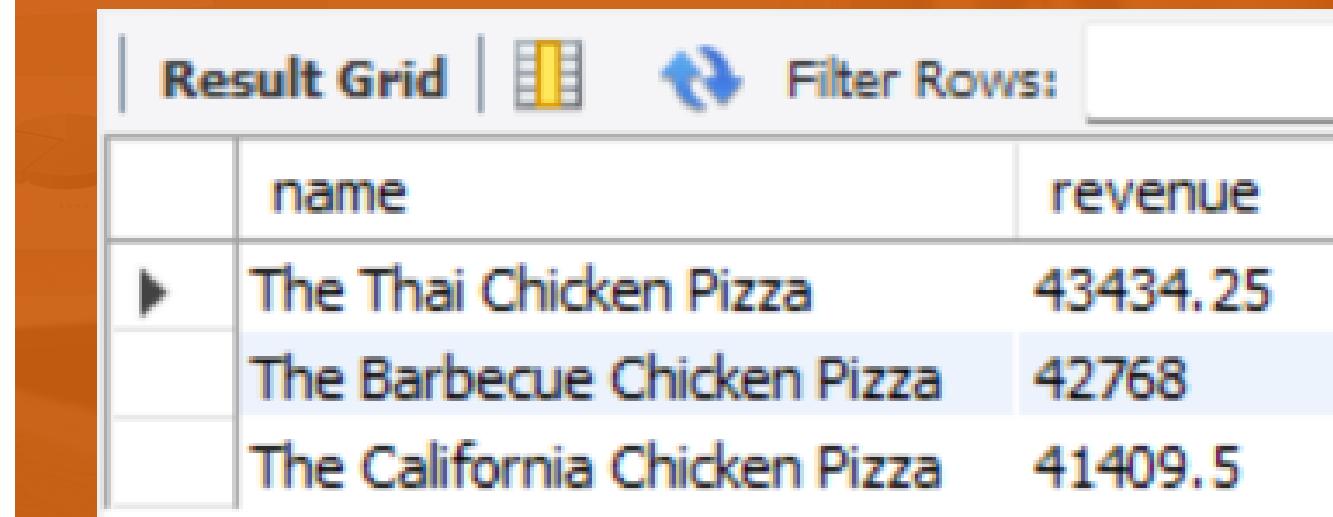
GROUP BY hour;

	hour	order_count
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2000



# DETERMINE TOP 3 PIZZA TYPES BASED ON REVENUE.

```
SELECT  
    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.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

# GROUP THE ORDERS BY DATE AND CALCULATE THE AVERGAE NUMBER OF PIZZAS ORDERED PER DAY

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

	Result Grid			
	avg_pizza_perday			
▶	138			

# CALCULATE PERCENTAGE CONTRIBUTION OF EACH TYPES OF PIZZA TO TOTAL REVENUE GENERATED.

```
select pizza_types.category,  
round((sum(order_details.quantity * pizzas.price) / (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)) * 100, 2) 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;
```

	category	revenue
▶	Classic	26.01
	Supreme	
	Chicken	23.96
	Veggie	23.68



# ANALYZE THE CUMULATIVE GENERATE 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;
```

order_date	cum_revenue
2015-12-23	805415.9
2015-12-24	807553.75
2015-12-26	809196.8
2015-12-27	809196.8
2015-12-28	812253
2015-12-29	813606.25
2015-12-30	814944.05
2015-12-31	817860.05
...+ 2 ...	

# DETERMINE TOP 3 MOST ORDERED PIZZA TYPE 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
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;
```

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

9987252182

sharjeel.3119030.co@mhssce.ac.in

# THANK YOU!



Thank you for your attention to our sales report analysis presentation. If you have any questions or would like to discuss the findings in more detail, please don't hesitate to reach out to us. We appreciate your continued support and partnership.