PIZZA SALES ANALYSIS USING SQL

BILLU'S PIZZA HUB

INTRODUCTION

Hello! I am Karuna Rathore an Aspiring Data Analyst and Economics professional.

- Objective: Used SQL to analyze a comprehensive pizza sales dataset. Used MYSQL for this Project. The dataset includes four tables:

1. Orders 2. Order_Detail 3. Pizzas 4. Pizza_Types

- Key Activities: Executed SQL queries to answer critical business questions, focusing on uncovering patterns in orders, understanding customer preferences, evaluating revenue trends, and assessing product performance.

- Outcome: This project showcases proficiency in working with relational databases and converting unprocessed data into actionable insights and strategic business intelligence.

- Business Impact: The analysis underscores the significance of data-driven decision-making in understanding sales patterns and optimizing inventory management, particularly in the dynamic food industry.

QUESTIONS

Basic:

- Retrieve the total number of orders placed.
- Calculate the total revenue generated from pizza sales.
- Identify the highest-priced pizza.
- Identify the most common pizza size ordered.
- List the top 5 most ordered pizza types along with their quantities.

Intermediate:

- Join the necessary tables to find the total quantity of each pizza category ordered.
- Determine the distribution of orders by hour of the day.
- Join relevant tables to find the category-wise distribution of pizzas.
- Group the orders by date and calculate the average number of pizzas ordered per day.
- Determine the top 3 most ordered pizza types based on revenue.

Advanced:

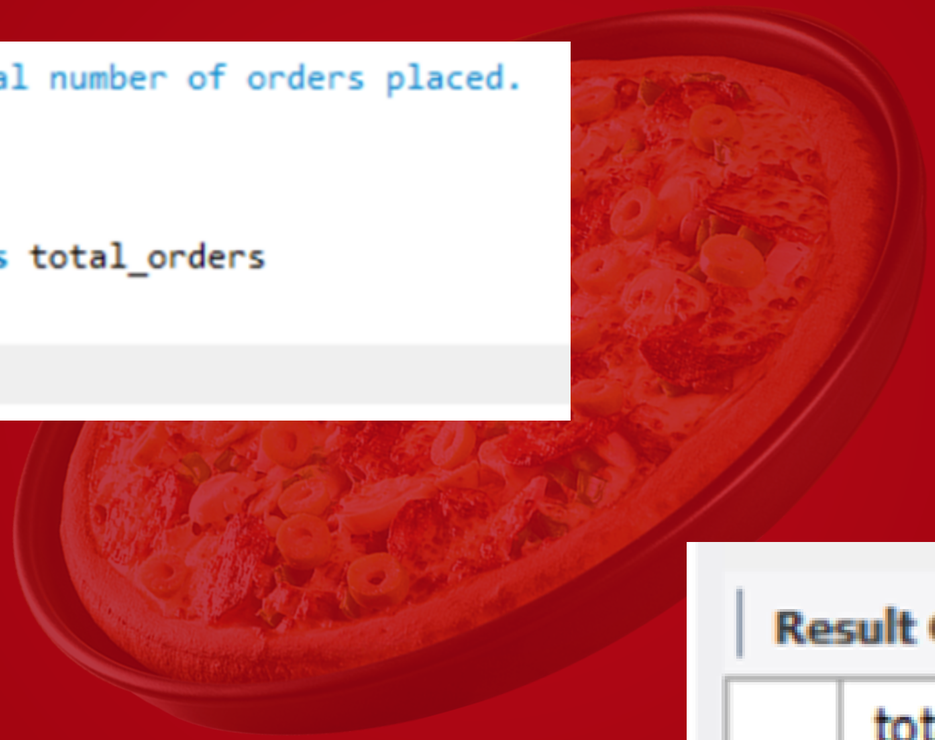
- Calculate the percentage contribution of each pizza type to total revenue.
- Analyze the cumulative revenue generated over time.
- Determine the top 3 most ordered pizza types based on revenue for each pizza category.

BUSINESS INSIGHTS

- Large pizzas were the most commonly ordered size around 18,526 units, indicating a preference for larger portions among customers.
- The classic Deluxe Pizza is the most ordered (Quantity sold) pizza type followed by others.
- Classic Category has highest Total quantity of pizza ordered in comparison to Supreme, Veggie, and Chicken.
- The busiest hours for orders were in the evening (5–7 PM) and afternoon (12–1 PM), reflecting mealtime peaks for the Billu's Pizza Hub.
- in terms of revenue, the Chicken category dominated, with its top three pizza types contributing significantly. Contribute around 23.96% in total revenue.
- Total ordered placed are around 21,350 and total revenue generated is around $8,17,860.
- Average pizza ordered per day is 138.

Q1. RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED

```
20      -- Retrieve the total number of orders placed.
21
22 •    select
23        count(order_id) as total_orders
24      from
25        orders;
```



Result Grid | Filter R

	total_orders
▶	21350

Q2. CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

```sql
-- 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 pizzas.pizza_id = order_details.pizza_id;
```

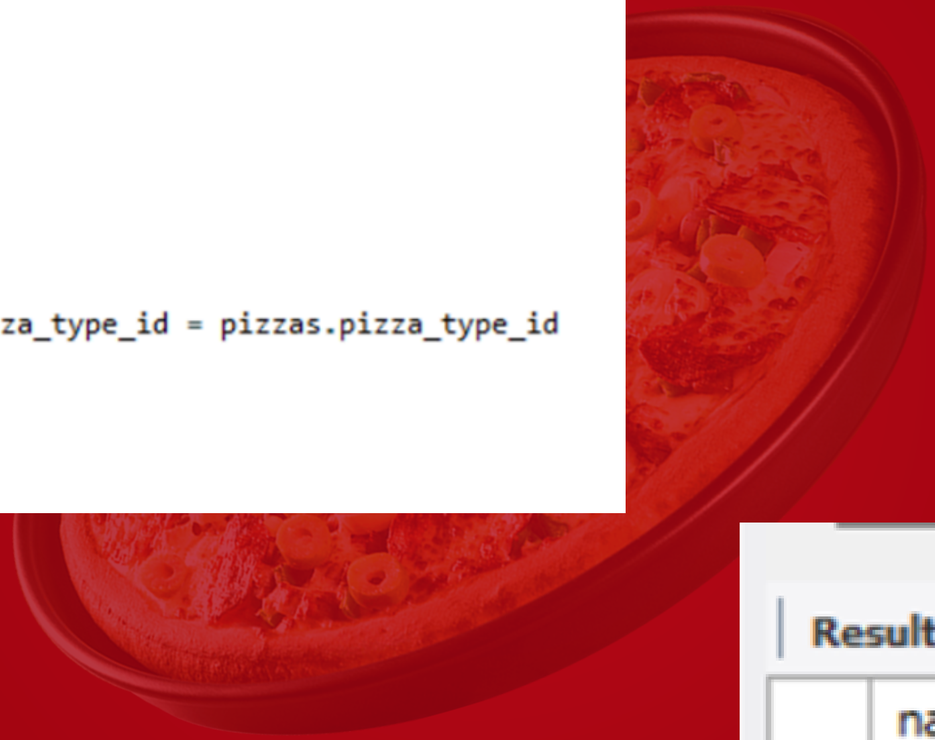
Result Grid | Filter Row

	total_sales
▶	817860.05

Q3. IDENTIFY THE HIGHEST-PRICED PIZZA

```sql
-- 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

Q4. IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

```sql
-- 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;
```

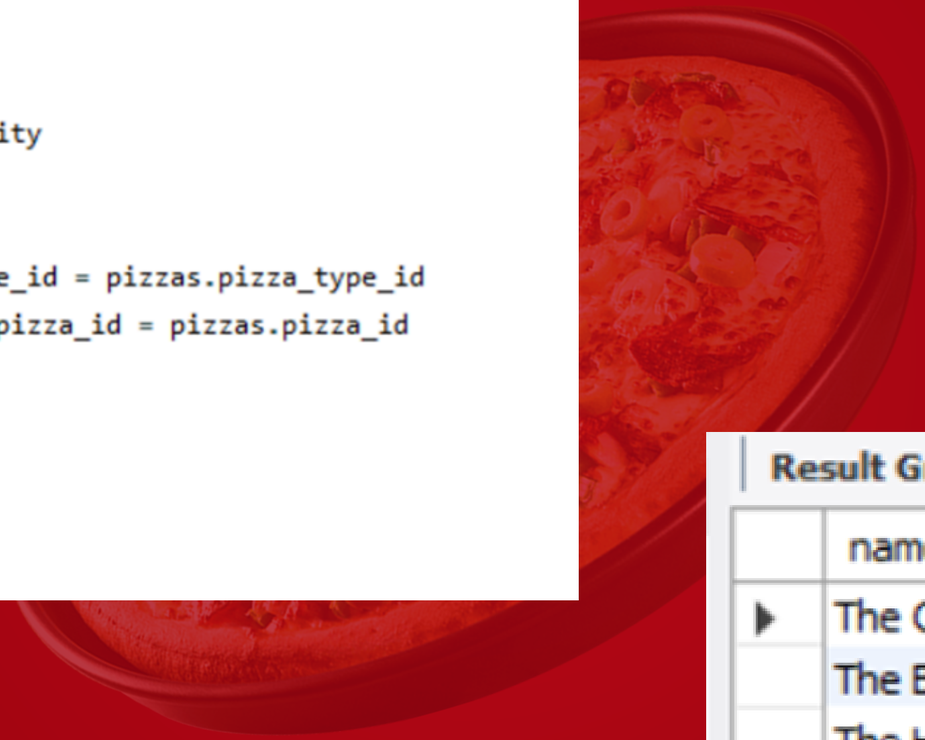
Result Grid | Filter Rows:

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

Q5. LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES

```sql
-- 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

Q8. JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED

```sql
-- Join the necessary tables to find the total quantity of each pizza category ordered.

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;
```

Result Grid | Filter R

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

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

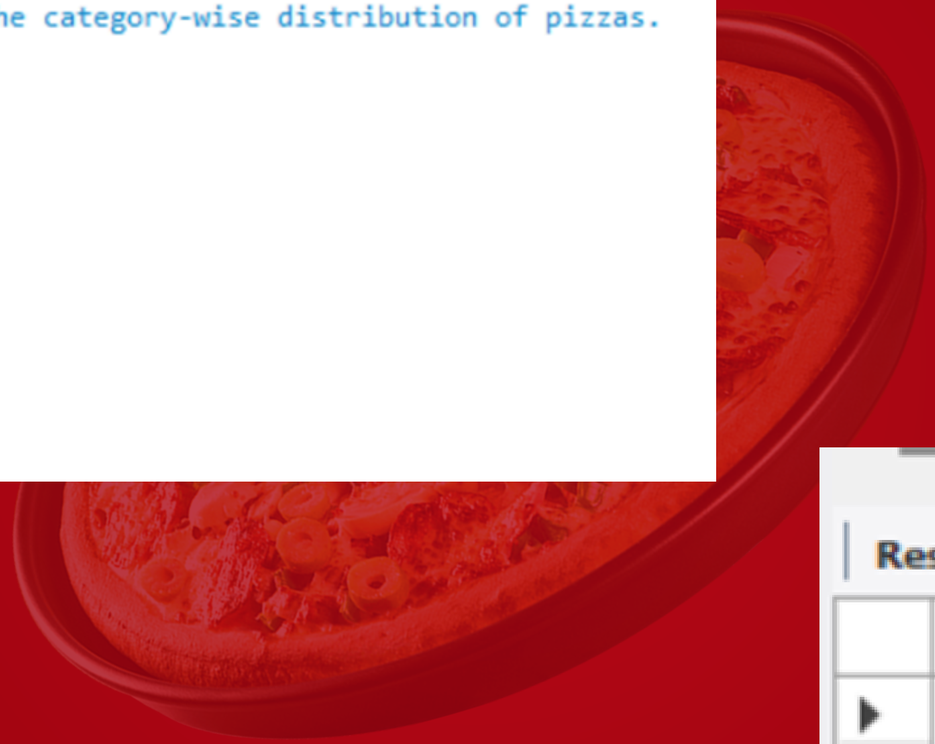
```
95    -- Determine the Distrbution of orders by hour of the Day.
96
97    select
98      hour(order_time) as hour,
99      count(order_id) as order_count
100   from
101     orders
102   group by
103     hour;
104
```

Result Grid | Filter Ro

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

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

```
105    -- Join relevant tables to find the category-wise distribution of pizzas.
106
107 •  select
108      category,
109      count(name)
110    from
111      pizza_types
112    group by
113      category;
114
```



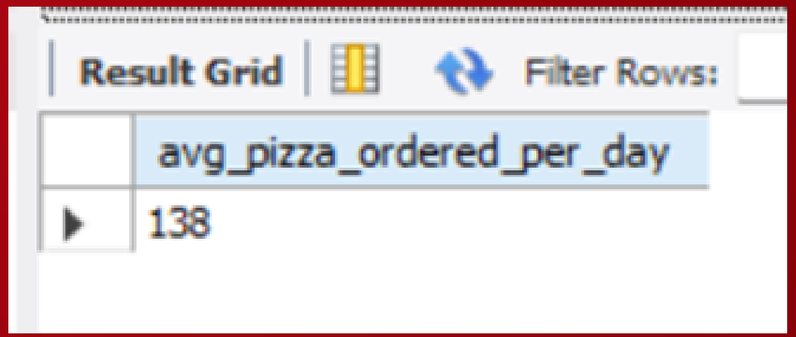
Result Grid | Filter Rows:

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

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

```sql
-- Group the orders by date and calculate the average number of pizzas ordered per day.

select
  round(
    avg(quantity),
    0
  ) as avg_pizza_ordered_per_day
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;
```



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

```sql
-- Determine the top 3 most ordered 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;
```

Result Grid | Filter Rows:

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

Q11. CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE

```sql
-- 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_sales
from
  order_details
  join pizzas on pizzas.pizza_id = order_details.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;
```

Result Grid | Filter Rows

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Q12. ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

```sql
-- 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

Q13. DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY

```sql
-- Determine the top 3 most ordered pizza types based on revenue for each pizza category.

select category, 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
          pizzas
          join pizza_types on pizzas.pizza_type_id = pizza_types.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:

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

BILLU'S PIZZA HUB

THANK YOU!