PIZZA SALES DATA ANALYSIS DONIMOJHUB

A MYSQL-BASED DATA ANALYTICS PROJECT

BY ARYA BHATTACHARYA

ABOUT ME

Hi, I am **Arya Bhattacharya,** currently a 4th year Undergraduate Student in Production Engineering, Jadavpur University with strong interests in data analytics, SQL, and business insights.

This is my portfolio project where i have used MySQL to analyse the pizza sales of a company named **DonimojHub**

PROJECT OVERVIEW

Objective : Analyze sales data to understand order trends, customer preferences, and revenue distribution.

Dataset: Orders, Order Details, Pizza Types, Pizzas

Tools used: MySQL for data analysis, **Canva** for presentation

BUSINESS QUESTIONS

- **Basic Analysis:** Total orders, revenue, highest-priced pizza, most common pizza size, top 5 pizza types by quantity
- **Intermediate Analysis:** Category-wise sales, hourly trends, daily averages, top pizzas by revenue
- **Advanced Analysis:** Revenue contribution percent, cumulative revenue trend, top 3 pizzas by category & revenue

1. Retrieve the total number of orders placed

QUERY

```sql
USE donimojhub;
SELECT
    COUNT(order_id) AS total_orders
FROM
    ORDERS;
```

OUTPUT

Result Grid

total_orders
21350

2. Calculate the total revenue generated from pizza sales

QUERY

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

OUTPUT

Result Grid

total_revenue
817860.05

3. Identify the highest-priced pizza.

QUERY

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

OUTPUT

Result Grid | Filter Rows:

name	price
The Greek Pizza	35.95

4. Identify the most common pizza size ordered

QUERY

```sql
-- IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS countOrders
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY countOrders DESC
LIMIT 1;
```

OUTPUT

Result Grid | Filter Rows:

size	countOrders
L	18526

5. List the top 5 most ordered pizza types along with their quantities

QUERY

```sql
-- IDENTIFY THE 5 MOST COMMON PIZZA types ORDERED along with their quantities

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

OUTPUT

Result Grid | Filter Rows:

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

6. Join the necessary tables to find the total quantity of each pizza category ordered

QUERY

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

SELECT
    pizza_types.category, SUM(order_details.quantity) AS summary
FROM
    pizza_types
        JOIN
    pizzas 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
ORDER BY summary DESC;
```

OUTPUT

Result Grid | Filter Rows:

category	summary
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

7. Determine the distribution of orders by hour of the day

QUERY

```sql
-- Determine the distribution of orders by hour of the day
SELECT
    HOUR(order_time), COUNT(order_id)
FROM
    orders
GROUP BY (HOUR(order_time));
```

OUTPUT

Result Grid | Filter Rows:

HOUR(order_time)	COUNT(order_id)
11	1231
12	2520
13	2455
14	1472
15	1468

8. Join relevant tables to find the category-wise distribution of pizzas.

QUERY

```sql
-- Join relevant tables to find the category-wise distribution of pizzas.
select category, count(name) from pizza_types
group by category;
```

OUTPUT

Result Grid | Filter Rows:

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

9. Group the orders by date and calculate the average number of pizzas ordered per day

QUERY

```sql
-- Group the orders by date and calculate the average number of pizzas ordered per day.
SELECT
    ROUND(AVG(quantity), 0) AS dataset
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        order_details
    JOIN orders ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS avgquantity;
```

OUTPUT

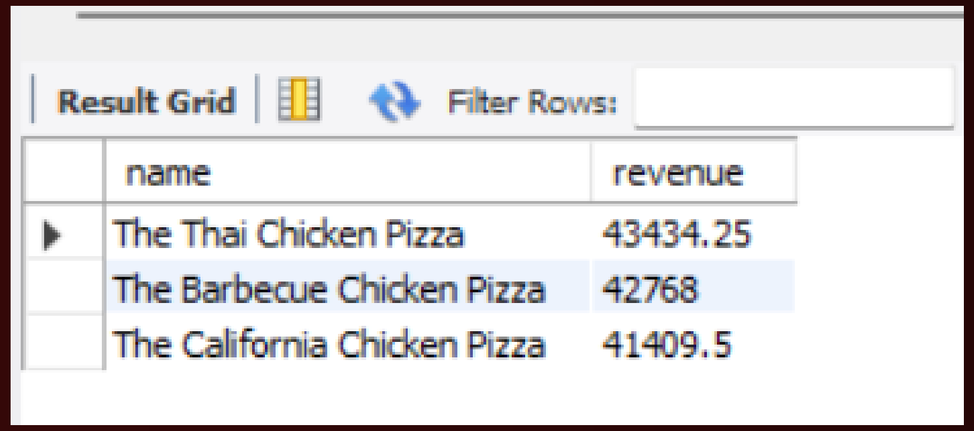
	dataset
▶	138

10. Determine the top 3 most ordered pizza types based on revenue

QUERY

```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;
```

OUTPUT



Result Grid | Filter Rows:

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

11. Calculate the percentage contribution of each pizza type to total revenue

QUERY

```sql
-- Calculate the percentage contribution of each pizza type to total revenue
select pizza_types.category,
(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 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 category
order by revenue desc;
```

OUTPUT

Result Grid | Filter Rows:

category	revenue
Classic	26.90596025566967
Supreme	25.45631126009862
Chicken	23.955137556847287
Veggie	23.682590927384577

12. Analyze the cumulative revenue generated over time

QUERY

```sql
-- Analyse 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 order_date) as sales;
```

OUTPUT

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

13. Determine the top 3 most ordered pizza types based on revenue for each pizza category.

QUERY

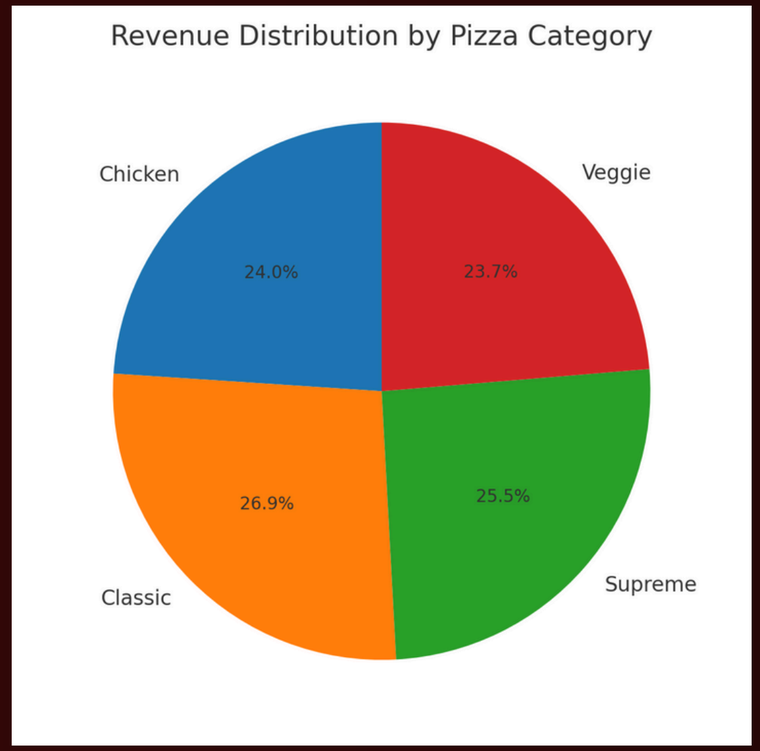
```sql
-- Determine the top 3 most ordered pizza types based on revenue for each pizza category
select name, revenue from
(select category,name, revenue,
rank() over (partition by category order by revenue desc) as rnk from
(select pizza_types.category, pizza_types.name, sum(order_details.quantity * 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 category, name) as a) as b
where rnk <=3;
```

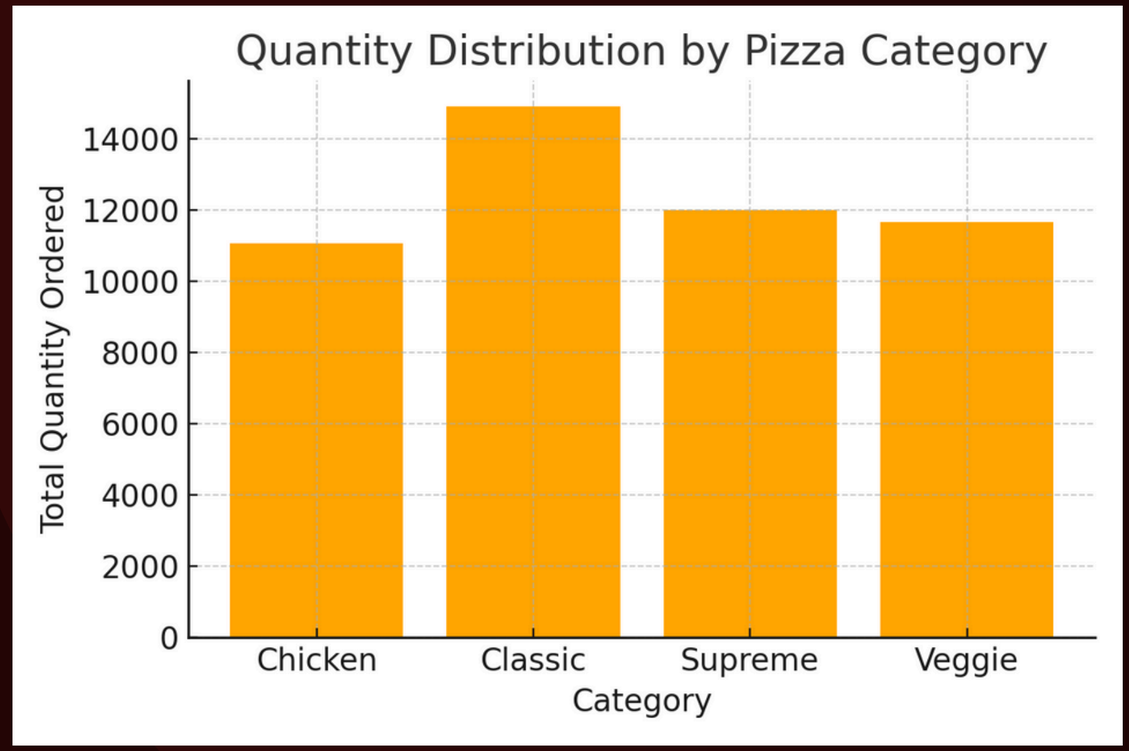
OUTPUT

Result Grid | Filter Rows:

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

SALES REPORT





TOP-SELLING PIZZAS TO FOCUS MARKETING ON (BY REVENUE)

THAI CHICKEN PIZZA– ₹43,434.25

BARBECUE CHICKEN PIZZA – ₹42,768.00

CALIFORNIA CHICKEN PIZZA – ₹41,409.50

Remarks: Can be pushed in social media ads & combo offers to boost repeat orders.

BEST TIME SLOTS FOR OFFERS (BY ORDER COUNT)

- Lunch Rush: 12 PM (2,520 orders), 1 PM (2,455 orders)
- Dinner Peak: 6 PM (2,399 orders), 5 PM (2,336 orders)

Remarks: Consider discounts in off-peak hours (e.g., 3–5 PM) to increase traffic.

CATEGORIES THAT NEED PROMOTION (LOWEST TOTAL REVENUE)

- Veggie – ₹1,93,690.45
- Chicken – ₹1,95,919.50

Remarks: Run seasonal specials or bundle deals for these to boost visibility..

THANK YOU

CONTACT INFORMATION:

https://www.linkedin.com/in/arya-bhattacharya-04a264291/

aryabhattacharya16@gmail.com

https://github.com/AryaBhatta0000