PIZZA HUT

A PROJECT ON PIZZA SALES ANALYSIS

HELLO!

i’m Dintakurthi kavya supriya, have utilized SQL queries to analyze and address various questions related to pizza sales.

This project focuses on analyzing the sales trends of a pizza business and developing strategies to increase its market share.

PROJECT SUMMARY

01 OBJECTIVE

Using SQL to extract, analyze, and interpret data to :

- draw meaningful insights related to pizza sales
- finding customer preference
- Revenue generation

02 ABOUT THE DATASET

- Analyzing sales records from a pizza restaurant.
- This dataset has around 48,000 sales transactions to analyze.

03 OBJECTIVE

- Sun Query
- Group By
- Cumulative sum, Rank
- Visualizations

04 ABOUT THE DATASET

- MySQL for data querying and manipulation.

BASIC QUESTIONS

Calculate the total revenue generated from pizza sales.

```sql
use pizzahut;
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

	total_sales
▶	817860.05

Identify the highest-priced pizza.

```sql
SELECT
    pizza_types.pizza_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:

	pizza_name	price
▶	The Greek Pizza	35.95

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

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

```
SELECT
    pizza_types.pizza_name,
    SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_i
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.pizza_name
ORDER BY quantity DESC
LIMIT 5;
```

Result Grid | Filter Rows:

pizza_name	quantity
The California Chicken Pizza	4802
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418

ADVANCED QUESTIONS

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

```
    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 Rows

category	quantity
Chicken	12509
Classic	12435
Supreme	3311
Classic	2453

Determine the distribution of orders by hour of the day.

```sql
SELECT
    HOUR(order_time), COUNT(order_id) as order_count
FROM
    orders
GROUP BY HOUR(order_time);
```

Result Grid | Filter Rows:

HOUR(order_time)	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

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

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

Result Grid | Filter Rows:

category	COUNT(pizza_name)
Chicken	7
Classic	7
Classic	1
Supreme	3
Supreme	6
Veggie	9

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

```sql
SELECT
    pizza_types.pizza_name,
    SUM(order_details.quantity * pizzas.price) AS revenue
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.pizza_name
ORDER BY revenue DESC
LIMIT 3;
```

Result Grid | Filter Rows:

pizza_name	revenue
The California Chicken Pizza	84177.5
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768

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

```sql
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 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 revenue DESC;
```

Result Grid | Filter Rows:

category	revenue
Chicken	27.142265965919226
Classic	22.237618771084374
Supreme	7.458427636855531
Classic	4.66834148458529

THANK YOU !

DO you have any questions?

- https://www.linkedin.com/in/kavya-supriya-dintakurthi-71b4982b9/
- https://www.hackerrank.com/profile/kavyasupriya_d
- https://github.com/kavya2804