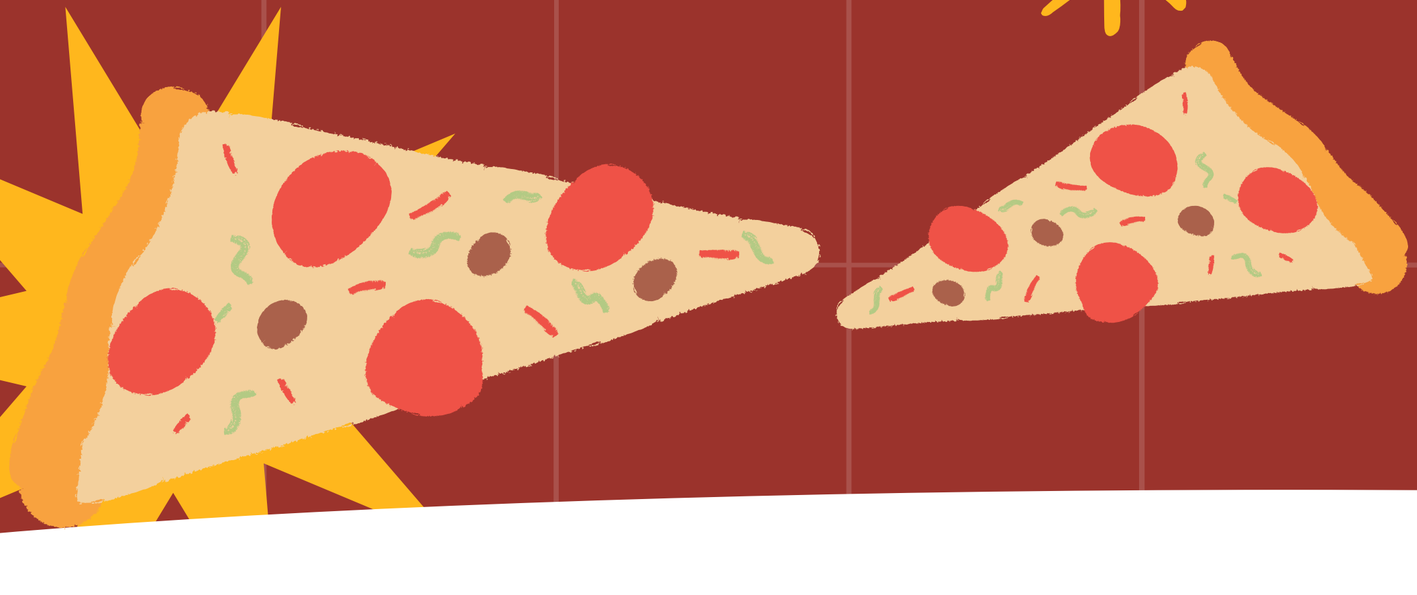
(DATA ANALYTICS)

SQL PROJECT ON PIZZA SALES



BY
ANURAG YESANSURE

LIST OF CONTENTS

INTRODUCTION

This project explores pizza sales data using SQL to uncover key business insights. The database consists of tables capturing essential information about pizzas, orders, and their details. By analyzing sales trends, customer preferences, and revenue patterns, the project aims to provide actionable insights to enhance operational efficiency and maximize revenue.

Objectives:

1. **Basic Analysis**
   - **Calculate total orders and revenue.**
   - **Identify the highest-priced pizza and the most common pizza size.**
   - **Highlight the top 5 most ordered pizza types.**
2. **Intermediate Analysis**
   - **Explore pizza category distributions and order trends by time.**
   - **Analyze daily pizza orders and category-based trends.**
   - **Identify revenue-generating pizza types.**
3. **Advanced Analysis**
   - **Determine revenue contributions by pizza type.**
   - **Analyze cumulative revenue over time.**
   - **Investigate top-performing pizza types by category and revenue.**

DATABASE BRIEF

PIZZA TABLE

- Purpose: Contains details about individual pizzas.
- Key Attributes:
- pizza_id: Unique identifier for each pizza.
- pizza_type_id: Links to the pizza type category.
- size: Indicates the size of the pizza (e.g., Small, Medium, Large).
- price: Cost of the pizza based on its size.

orders Table

- Purpose: Records details of customer orders.
- Key Attributes:
  - order_id: Unique identifier for each order.
  - order_date: Date when the order was placed.
  - order_time: Time when the order was placed.

order_details Table

- Purpose: Captures the details of each order, linking pizzas to specific orders.
- Key Attributes:
  - order_details_id: Unique identifier for each order detail record.
  - order_id: Links to the corresponding order in the orders table.
  - pizza_id: Links to the specific pizza in the pizzas table.
  - quantity: Number of pizzas of this type ordered.

pizza_types Table

- Purpose: Stores information about different types of pizzas and their categories.
- Key Attributes:
  - pizza_type_id: Unique identifier for each pizza type.
  - name: Name of the pizza (e.g., Margherita, Pepperoni).
  - category: Category of the pizza (e.g., Vegetarian, Non-Vegetarian).
  - ingredients: List of ingredients used in the pizza.

Relationships Between Tables

- pizzas.pizza_type_id → pizza_types.pizza_type_id (Pizza type and category mapping).
- order_details.order_id → orders.order_id (Details of each order).
- order_details.pizza_id → pizzas.pizza_id (Details of pizzas in an order).

SCHEMAS

pizzas Table

Column Name	Data Type	Constraints	Description
pizza_id	INT	PRIMARY KEY	Unique identifier for each pizza.
pizza_type_id	INT	FOREIGN KEY (REFERENCES pizza_types.pizza_type_id)	Links to the pizza type category.
size	VARCHAR	NOT NULL	Size of the pizza (e.g., Small, Medium).
price	DECIMAL	NOT NULL	Price of the pizza based on its size.

orders Table

Column Name	Data Type	Constraints	Description
order_id	INT	PRIMARY KEY	Unique identifier for each order.
order_date	DATE	NOT NULL	Date when the order was placed.
order_time	TIME	NOT NULL	Time when the order was placed.

pizza_types Table

Column Name	Data Type	Constraints	Description
pizza_type_id	INT	PRIMARY KEY	Unique identifier for each pizza type.
name	VARCHAR	NOT NULL	Name of the pizza (e.g., Margherita).
category	VARCHAR	NOT NULL	Category of the pizza (e.g., Vegetarian).
ingredients	TEXT		List of ingredients used in the pizza.

order_details Table

Column Name	Data Type	Constraints	Description
order_details_id	INT	PRIMARY KEY	Unique identifier for each record.
order_id	INT	FOREIGN KEY (REFERENCES orders.order_id)	Links to the corresponding order.
pizza_id	INT	FOREIGN KEY (REFERENCES pizzas.pizza_id)	Links to the specific pizza.
quantity	INT	NOT NULL	Quantity of pizzas ordered.

QUESTIONS

01 Retrieve the total number of orders placed.

```sql
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

Result Grid

	total_orders
▶	21350

QUESTIONS

02 Calculate the total revenue generated from pizza sales.

```
1 •   SELECT
2         ROUND(SUM(order_details.quantity * pizzas.price),
3                 2) AS total_sales
4     FROM
5         order_details
6             JOIN
7         pizzas ON pizzas.pizza_id = order_details.pizza_id
```

Result Grid

	total_sales
▶	817860.05

QUESTIONS

03 Identify the highest-priced pizza.

```sql
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

QUESTIONS

04 Identify the most common pizza size ordered.

```sql
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 Rov

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

QUESTIONS

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

```sql
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

QUESTIONS

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

```sql
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 Ro

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

QUESTIONS

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

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

Result Grid | Filter Rows:

HOUR(order_time)	COUNT(order_id)
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

QUESTIONS

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

```
1 •    SELECT
2          category, COUNT(name)
3      FROM
4          pizza_types
5      GROUP BY category;
6
```

Result Grid | Filter Rows:

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

QUESTIONS

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

```sql
SELECT
    ROUND(AVG(quantity), 0) as avg_pizzas_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
```

Result Grid | Filter Rows:

	avg_pizzas_ordered_per_day
▶	138

QUESTIONS

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

```sql
select pizza_types.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.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

QUESTIONS

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

```sql
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 Ro

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

QUESTIONS

12 Analyze the cumulative revenue generated over time.

```sql
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	34343.50000000001

QUESTIONS

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

```sql
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: | Exp

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

REFERENCES




YOUTUBE: LINK(https://www.youtube.com/watch?v=zZpMvAedh_E)
CHANNEL: WsCube TECH


THANK YOU
• “When life gives you lemons, sell them and buy pizza”