HAVE A SLICE!

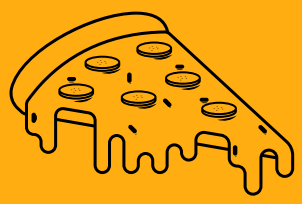
IT'S A PIZZA PARTY!

Pizza is the answer, no matter the question

PIZZA SALES SQL ANALYSIS

SQL queries for insights on pizza order and revenue

- Name: Anshu Kumari
- Course: Data Analysis



PROJECT OVERVIEW

Objective:

- Analyze pizza sales using SQL to gain insights into order patterns, revenue, and pizza preferences.

Tools Used:

- SQL
- Database: Pizza Sales DB

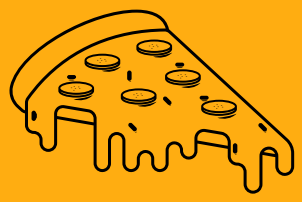
Categories:

- Basic Queries
- Intermediate Queries
- Advanced Queries

Pizza is the answer, no matter the question

BASIC QUERY

- Query 1: Total Number of Orders
- Query 2: Total Revenue Generated
- Query 3: Highest-Priced Pizza
- Query 4: Most Common Pizza Size Ordered
- Query 5: Top 5 Most Ordered Pizza Types



SNAPSHOTS OF BASIC QUERY

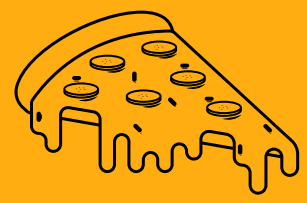
```sql
-- Retrieve the total number of orders placed.
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;

-- 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;

-- 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;
```

```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;

-- List the top 5 most ordered pizza types along with their quantit
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;
```

Pizza is the answer, no matter the question

INTERMEDIATE QUERY

- Query 1: Total Quantity by Pizza Category
- Query 2: Orders Distribution by Hour
- Query 3: Category-wise Pizza Distribution
- Query 4: Average Pizzas Ordered per Day
- Query 5: Top 3 Pizza Types Based on Revenue



SNAPSHOTS OF INTERMIDIATE QUERY

```sql
-- Join the necessary tables to find the total quantity of each
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;

-- Determine the 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(order_time);

-- Join relevant tables to find the category-wise distribution
SELECT
    category, COUNT(name)
FROM
```

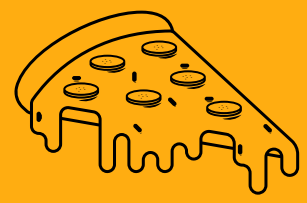
```sql
-- Group the orders by date and calculate the average numbe
SELECT
    ROUND(AVG(quantity), 0) as avg_pizzas_per_day
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS q
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.o
    GROUP BY orders.order_date) AS order_quantity;


-- Determine the top 3 most ordered pizza types based on re
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
        JOIN
```

Pizza is the answer, no matter the question

ADVANCE QUERY

- Query 1: Percentage Contribution of Each Pizza Type to Revenue
- Query 2: Cumulative Revenue Over Time
- Query 3: Top 3 Pizza Types by Revenue per Category

SNAPSHOTS OF ADVANCE QUERY



```sql
-- Calculate the percentage contribution of each pizza type to t
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)
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;

--  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
```

```sql
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;


-- Determine the top 3 most ordered pizza types based on revenu
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
```

Pizza is the answer, no matter the question

CONCLUSION

- Deeper analysis of sales trends.
- Customer behavior segmentation.
- Insights from each level: Basic, Intermediate, Advanced.

FOOD
Diary

every sunday
8:00am

only on our channel