KOL_PIZZAS

PIZZA SALES ANALYSIS

KOL_PIZZA

welcome to

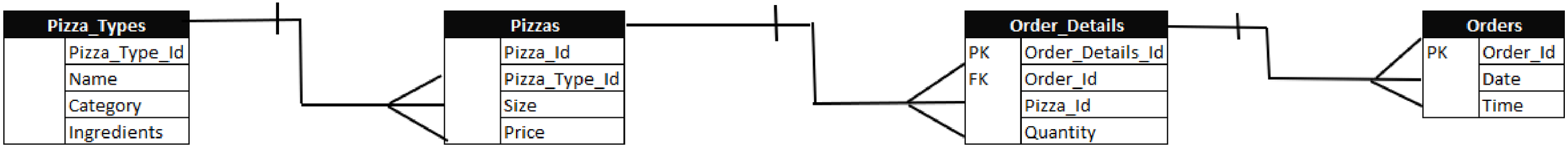
PIZZA SALES ANALYSIS

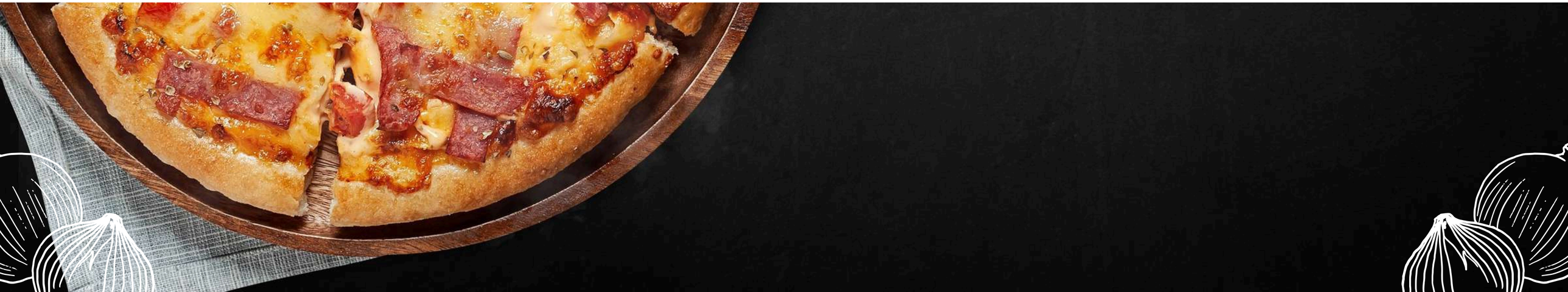
Hi, I am Nisha Choudhary, a passionate data enthusiast, and I am excited to present a detailed SQL-based data analysis project. In this project, I have leveraged basic to advanced SQL functionalities to solve a wide range of insight-driven questions. The analysis focuses on uncovering key business patterns such as revenue trends, top-performing products, customer behavior, and overall sales performance.

By applying complex queries, aggregate functions, joins, subqueries, and CTEs, I was able to transform raw data into meaningful insights. This project not only highlights my technical skills but also demonstrates how data can be used to support decision-making and business growth strategies.

The ultimate goal of this project is to understand revenue patterns, identify best-selling products, and provide actionable insights that can help businesses improve their performance and efficiency.

Entity Relationship Diagram





KOL_PIZZA

QUESTIONS TO ADDRESS FOR EFFECTIVE DATABASE ANALYSIS

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.

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

★★★★★

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

Result Grid | Filter Rows:

	total_orders
▶	21350

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```sql
SELECT
    ROUND(SUM(p.price * od.quantity), 3) AS total_revenue
FROM
    pizzas p
        JOIN
    order_details od ON p.pizza_id = od.pizza_id;
```

Result Grid | Filter Rows:

total_revenue
817860.05

IDENTIFY THE HIGHEST-PRICED PIZZA.

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

Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```sql
SELECT
    p.size, COUNT(od.order_details_id) AS odr
FROM
    pizzas p
        JOIN
    order_details od ON p.pizza_id = od.pizza_i
GROUP BY p.size
ORDER BY odr DESC
LIMIT 1;
```

Result Grid | Filter Rows:

	size	odr
▶	L	18526

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

★★★★★

```sql
SELECT
    p.size, COUNT(od.order_details_id) AS odr
FROM
    pizzas p
        JOIN
    order_details od ON p.pizza_id = od.pizza_id
GROUP BY p.size
ORDER BY odr DESC
LIMIT 1;
```

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

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

```sql
SELECT
    pt.category, sum(od.quantity) AS quantity
FROM
    pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.category;
```

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

Result Grid | Filter Rows:

category	quantity
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```sql
SELECT
    HOUR(o.order_time), COUNT(o.order_id) AS Count_orders
FROM
    orders o
GROUP BY HOUR(o.order_time)
ORDER BY Count_orders DESC;
```

Result Grid | Filter Rows:

HOUR(o.order_time)	Count_orders
12	2520
13	2455
18	2399
17	2336
19	2009
16	1920
20	1642
14	1472
15	1468
11	1231
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_type_id) AS count_of_pizza_types
FROM
    pizza_types
GROUP BY category;
```

Result Grid | Filter Rows:

category	count_of_pizza_types
Chicken	6
Classic	8
Supreme	9
Veggie	9

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

```sql
SELECT
    ROUND(AVG(quantity), 2)
FROM
    (SELECT
        o.order_date, SUM(od.quantity) AS quantity
    FROM
        orders o
    JOIN order_details od ON o.order_id = od.order_id
    GROUP BY o.order_date) AS order_quatity;
```

Result Grid | Filter Rows:

	ROUND(AVG(quantity), 2)
▶	138.47

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

```sql
SELECT
    pt.name, SUM(p.price * od.quantity) AS revenue
FROM
    pizzas p
        JOIN
    order_details od ON p.pizza_id = od.pizza_id
        JOIN
    pizza_types pt ON pt.pizza_type_id = p.pizza_type_id
GROUP BY pt.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

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

```sql
SELECT
    pt.category,
    SUM(od.quantity * p.price) AS revenue_total,
    ((SUM(od.quantity * p.price) / (SELECT
            SUM(od.quantity * p.price)
        FROM
            pizzas p
                JOIN
            order_details od ON p.pizza_id = od.pizza_id)) * 100) AS revenue_percentage
FROM
    pizzas p
        JOIN
    order_details od ON p.pizza_id = od.pizza_id
        JOIN
    pizza_types pt ON pt.pizza_type_id = p.pizza_type_id
GROUP BY pt.category;
```

Result Grid | Filter Rows: | Export:

category	revenue_total	revenue_percentage
Classic	220053.1000000001	26.905960255669903
Veggie	193690.45000000298	23.682590927384783
Supreme	208196.99999999822	25.45631126009884
Chicken	195919.5	23.955137556847493

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```sql
select order_date, sum(revenue)
over(order by order_date) as cum_revenue -- windows function
from(select o.order_date, round(sum(od.quantity * p.price),
1)
as revenue
from order_details  od join
 orders o on
 od.order_id = o.order_id
 join pizzas p on
 p.pizza_id = od.pizza_id
 group by o.order_date) as per_day_sales
 ;
```

Result Grid | Filter Rows:

order_date	cum_revenue
2015-01-01	2713.9
2015-01-02	5445.8
2015-01-03	8108.200000000001
2015-01-04	9863.7
2015-01-05	11929.7

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 pt.category, pt.name, round(sum(od.quantity * p.price),2) as revenue
from pizza_types pt join
pizzas p  on
pt.pizza_type_id = p.pizza_type_id
join
order_details od on
od.pizza_id = p.pizza_id
group by  pt.category,pt.name
) AS A) AS B;
```

Result Grid | Filter Rows:

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5
The Southwest Chicken Pizza	34705.75
The Chicken Alfredo Pizza	16900.25
The Chicken Pesto Pizza	16701.75
The Classic Deluxe Pizza	38180.5
The Hawaiian Pizza	32273.25

KOL_PIZZA

- GRATEFUL FOR YOUR TIME AND ATTENTION
- THIS PROJECT SHOWCASED THE POWER OF SQL IN ANALYZING DATA
- FROM BASIC QUERIES TO ADVANCED TECHNIQUES, THE ANALYSIS HELPED UNCOVER INSIGHTS ON:
- REVENUE PATTERNS
- BEST-PERFORMING PRODUCTS
- CUSTOMER AND SALES BEHAVIOR
- DATA-DRIVEN INSIGHTS CAN SUPPORT BETTER BUSINESS DECISIONS AND GROWTH STRATEGIES

KOL_PIZZA

THANK YOU!