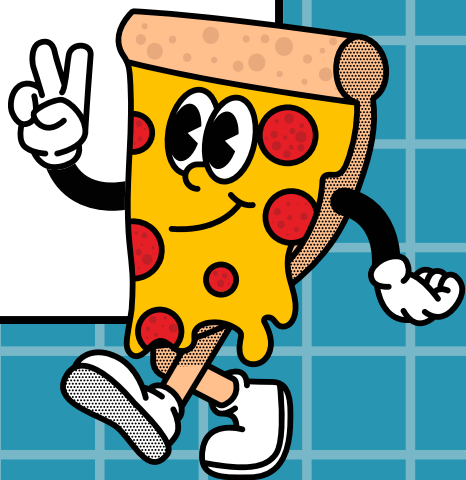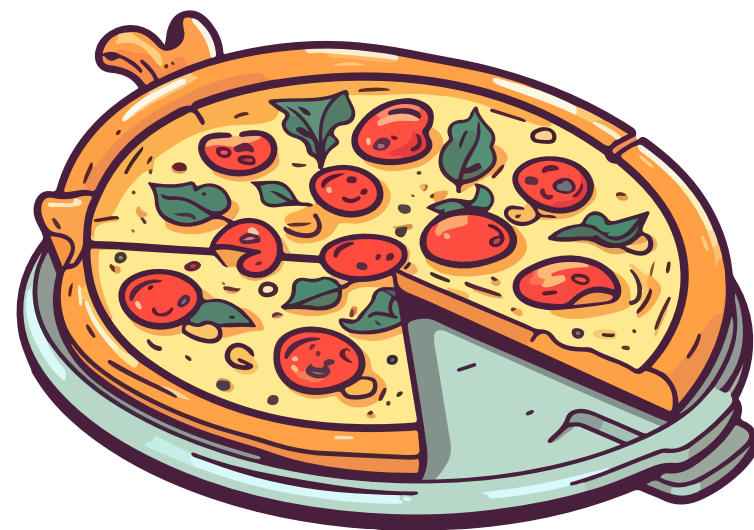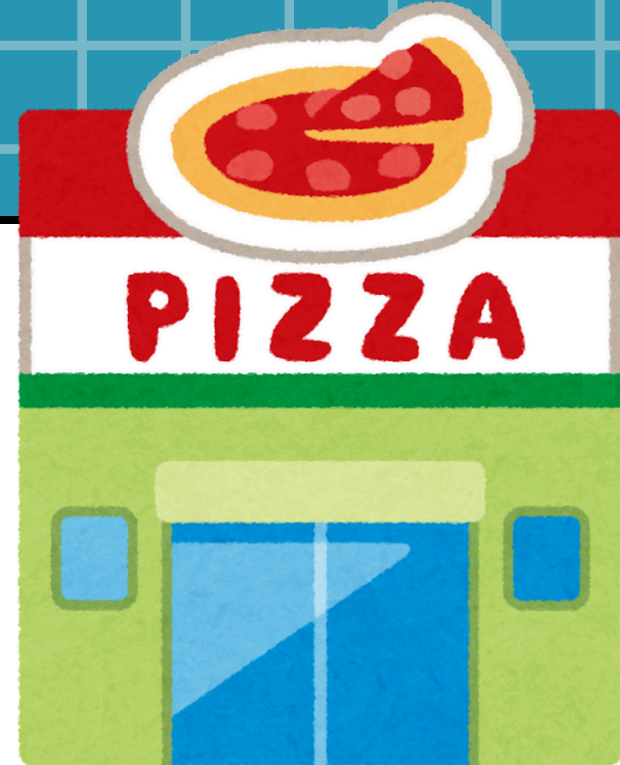# PIZZA SALES ANALYSIS USING SQL

A Data Exploration Project by

Aasif Shaikh

- **This project explores a pizza sales database to analyze order patterns, popular pizzas, and revenue trends.**
- **The goal is to understand customer behavior and identify best-selling pizzas.**

SCHEMA

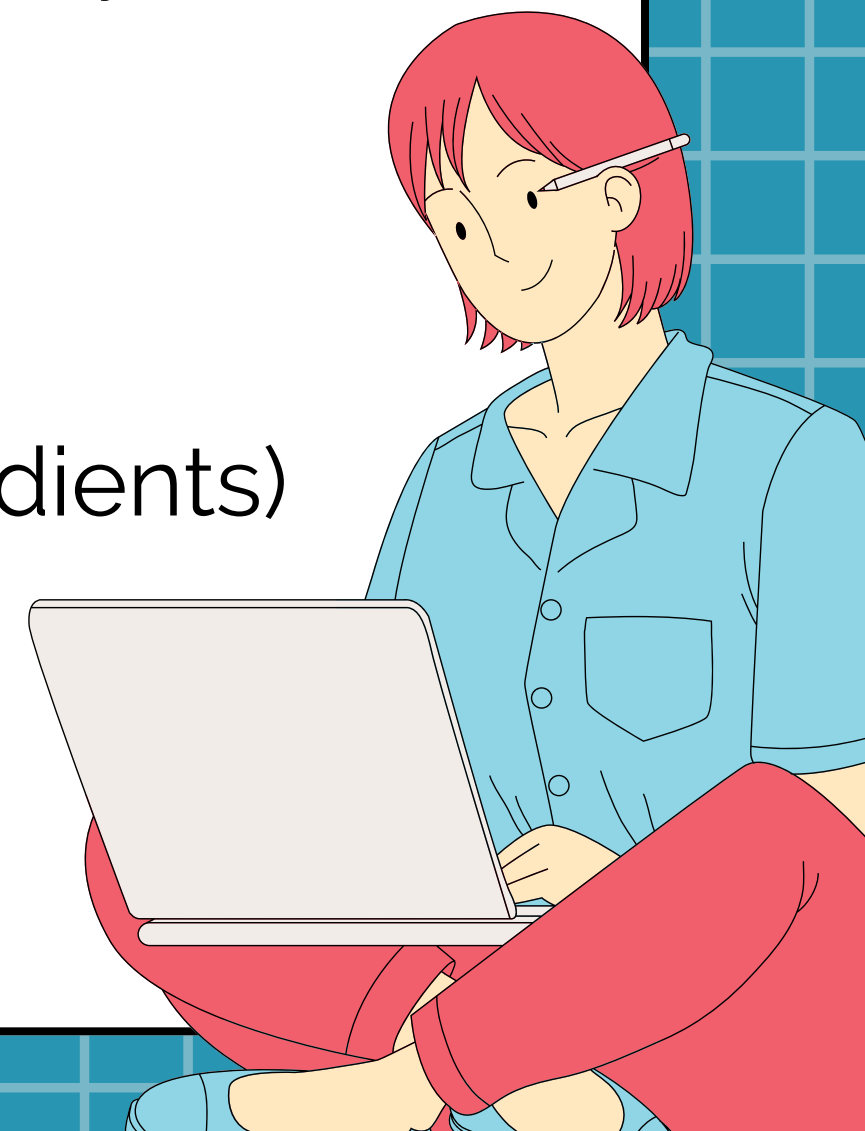**orders** (order_id, date, time)

|

|-- **order_details** (order_details_id, order_id, pizza_id, quantity)

  |

  |-- **pizzas** (pizza_id, pizza_type_id, size, price)

    |

    |-- **pizza_types** (pizza_type_id, name, category, ingredients)

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

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

total_order_placed
21350

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

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

total_revenue
561234.85

IDENTIFY THE HIGHEST-PRICED PIZZA.

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

name	highest_prize_pizza
▶ The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```sql
SELECT
    p.size, COUNT(od.order_details_id) AS most_common_order
FROM

    order_details od
        JOIN

    pizzas p ON od.pizza_id = p.pizza_id
GROUP BY p.size
ORDER BY most_common_order DESC;
```

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

```sql
SELECT
    pt.name, SUM(od.quantity) AS total_quantity
FROM
    pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details od ON p.pizza_id = od.pizza_id
GROUP BY pt.name , od.quantity
ORDER BY total_quantity DESC
LIMIT 5;
```

OUTPUT :

name	total_quantity
The Classic Deluxe Pizza	2382
The Pepperoni Pizza	2321
The Hawaiian Pizza	2319
The Barbecue Chicken Pizza	2316
The Thai Chicken Pizza	2260

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

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

OUTPUT :

category	total_quantity
Classic	14888
Supreme	8440
Chicken	11050

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```sql
SELECT
    HOUR(order_time) AS order_hours,
    COUNT(order_id) AS total_order
FROM
    orders
GROUP BY order_hours
ORDER BY order_hours;
```

OUTPUT :

order_hours	total_order
9	1
10	8
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

JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS

```sql
SELECT
    category, COUNT(pizza_type_id) AS total_pizzas
FROM
    pizza_types
GROUP BY category;
```

OUTPUT :

category	total_pizzas
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(total_quantity), 2) AS avg_order_per_day
FROM
    (SELECT
        o.order_date, SUM(od.quantity) AS total_quantity
    FROM
        orders o
    JOIN order_details od ON o.order_id = od.order_id
    GROUP BY o.order_date) AS order_quantity;
```

OUTPUT :

	avg_order_per_day
▶	138.47

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

```sql
SELECT
    pt.name, SUM(p.price * od.quantity) AS total_revenue
FROM
    pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details od ON p.pizza_id = od.pizza_id
GROUP BY pt.name
ORDER BY total_revenue DESC
LIMIT 3;
```

OUTPUT :

name	total_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,
    ROUND(ROUND(SUM(od.quantity * p.price), 2) / (SELECT
                        ROUND(SUM(od.quantity * p.price), 0)
                FROM
                        order_details od
                            JOIN
                        pizzas p ON od.pizza_id = p.pizza_id) * 100,
            2) AS total_revenue
FROM
    order_details od
        JOIN
    pizzas p ON od.pizza_id = p.pizza_id
        JOIN
    pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
GROUP BY pt.category;
```

OUTPUT :

category	total_revenue
Classic	39.21
Supreme	25.88
Chicken	34.91

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```sql
select order_date,revenue,
sum(revenue) over (order by order_date) as cum_revenue
from(
select order_date,round(sum(od.quantity*p.price),2) as revenue
from order_details od join pizzas p
on od.pizza_id = p.pizza_id
join orders o
on o.order_id = od.order_id
group by order_date) as revenue;
```

OUTPUT :

order_date	revenue	cum_revenue
2015-01-01	1838	1838
2015-01-02	1772	3610
2015-01-03	1897.15	5507.15
2015-01-04	1193.05	6700.2
2015-01-05	1457.05	8157.25
2015-01-06	1658.9	9816.15
2015-01-07	1507.8	11323.949999999999
2015-01-08	1772.25	13096.199999999999
2015-01-09	1378.65	14474.849999999999
2015-01-10	1544.9	16019.749999999998

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

```sql
select * from (
    select category,name,revenue,
    rank() over (partition by category order by revenue desc) as rn
    from (
    select pt.category,pt.name, sum(od.quantity*p.price) 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
    where rn<=3;
```
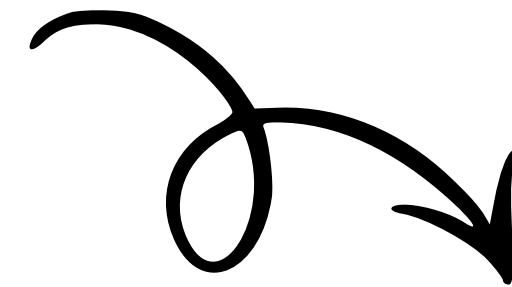
OUTPUT :

category	name	revenue	rn
Chicken	The Thai Chicken Pizza	43434.25	1
Chicken	The Barbecue Chicken Pizza	42768	2
Chicken	The California Chicken Pizza	41409.5	3
Classic	The Classic Deluxe Pizza	38180.5	1
Classic	The Hawaiian Pizza	32273.25	2
Classic	The Pepperoni Pizza	30161.75	3
Supreme	The Italian Supreme Pizza	33476.75	1
Supreme	The Sicilian Pizza	30940.5	2
Supreme	The Pepper Salami Pizza	25529	3