PIZZA SALES ANALYSIS

INTRODUCTION

This project focuses on analysing sales data from pizzahut restaurant to uncover performance trends, customer preferences, and operational insights. Using SQL in MySQL Workbench, I explored key business questions related to revenue, order volume, peak hours, and best-selling items. The goal was to support data-driven decision-making and provide actionable recommendations to optimise sales and improve restaurant operations.

VISSION & MISSION

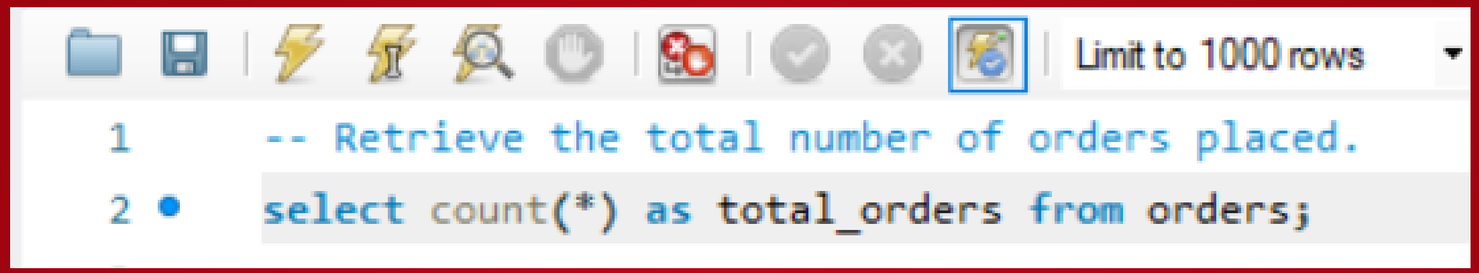
VISSION

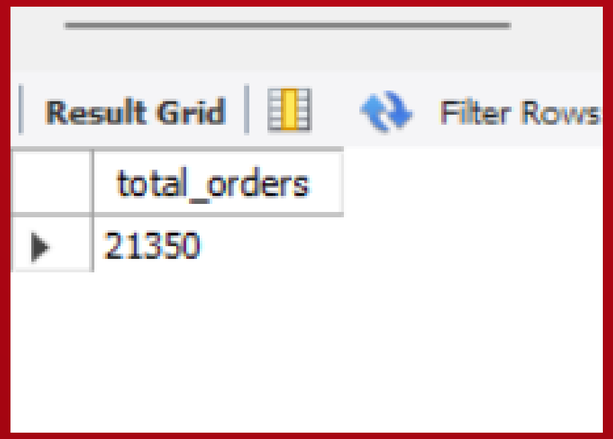
To empower restaurant decision-makers with data-driven insights that enhance sales performance, optimize operations, and elevate the customer dining experience.

MISSION

To analyze pizza sales data using SQL and MySQL Workbench, identify trends in customer preferences and peak performance periods, and deliver clear, actionable insights to improve menu strategies, staffing, and overall profitability.

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED





IDENTIFY THE HIGHEST PRICED PIZZA

```
12
13          -- Identify the highest-priced pizza.
14          SELECT
15          pizza_types.name, pizzas.price
16      FROM
17          pizza_types
18              JOIN
19          pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
20      ORDER BY pizzas.price DESC
21      LIMIT 1;
22
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

name	price
The Greek Pizza	35.95

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

```
3
4   -- Calculate the total revenue generated from pizza sales.
5   SELECT
6       ROUND(SUM(orders_details.quantity * pizzas.price),
7               2) AS total_sales
8   FROM
9       orders_details
10          JOIN
11      pizzas ON pizzas.pizza_id = orders_details.pizza_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

total_sales
817860.05

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

```
22
23      -- Identify the most common pizza size ordered. (quantity)
24      SELECT
25          quantity, COUNT(order_details_id)
26      FROM
27          orders_details
28      GROUP BY quantity;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

quantity	count(order_details_id)
1	47693
2	903
3	21
4	3

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

```sql
-- Identify the most common pizza size ordered.
SELECT
    pizzas.size,
    COUNT(orders_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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

```
34    -- List the top 5 most ordered pizza types along with their quantities.
35    SELECT
36        pizza_types.name,
37        SUM(orders_details.quantity) AS order_quantity
38    FROM
39        pizza_types
40            JOIN
41        pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
42            JOIN
43        orders_details ON pizzas.pizza_id = orders_details.pizza_id
44    GROUP BY pizza_types.name
45    ORDER BY order_quantity DESC
46    LIMIT 5;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

name	order_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
-- Join the necessary tables to find the total quantity of each pizza category ordered.
SELECT
    pizza_types.category,
    SUM(orders_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

```
60
61      -- Determine the distribution of orders by hour of the day.
62  •   SELECT
63          HOUR(order_time), COUNT(order_id) as order_count
64      FROM
65          orders
66      GROUP BY HOUR(order_time);
67
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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

```
68    -- Join relevant tables to find the category-wise distribution of pizzas.
69    SELECT
70        category, COUNT(name)
71    FROM
72        pizza_types
73    GROUP BY category;
74
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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

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

```
74
75      -- Group the orders by date and calculate the average number of pizzas ordered per day.
76      SELECT
77          ROUND(AVG(quantity), 0) as avg_pizza_ordered_per_day
78      FROM
79          (SELECT
80              orders.order_date, SUM(orders_details.quantity) AS quantity
81          FROM
82              orders
83          JOIN orders_details ON orders.order_id = orders_details.order_id
84          GROUP BY orders.order_date) AS order_quantity;
85
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

avg_pizza_ordered_per_day
138

CALCULATE THE % DISTRIBUTION OF EACH PIZZA TYPE TOTOTAL REVENUE

```sql
-- Calculate the percentage contribution of each pizza type to total revenue.
SELECT
    pizza_types.category,
    round(SUM(orders_details.quantity * pizzas.price) / (SELECT
            ROUND(SUM(orders_details.quantity * pizzas.price),
                    2) AS total_sales
        FROM
            orders_details
                JOIN
            pizzas ON pizzas.pizza_id = orders_details.pizza_id) * 100, 2) as revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE

```sql
-- Determine the top 3 most ordered pizza types based on revenue.
SELECT
    pizza_types.name,
    SUM(orders_details.quantity * pizzas.price) AS total_revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY total_revenue DESC
LIMIT 3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

name	total_revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

ANALYSE THE CUMULATIVE REVENUE GENERATED OVER TIME

```
119     -- Analyze the cumulative revenue generated over time.
120     select order_date,
121     sum(revenue) over(order by order_date) as cum_revenue
122     from (select orders.order_date, sum(orders_details.quantity*pizzas.price) as revenue
123     from orders_details
124     join pizzas on pizzas.pizza_id = orders_details.pizza_id
125     join orders on orders.order_id = orders_details.order_id
126     group by orders.order_date) as sales;
127
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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

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

```sql
-- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
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(orders_details.quantity*pizzas.price) as revenue
from pizza_types join pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id
join orders_details on orders_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: | Export: | Wrap Cell Content:

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

THANK YOU!