PIZZA SALES DATA EXPLORATION

Using Mysql

INTRODUCTION

In this project, I explored a pizza sales dataset using MySQL to analyze different aspects of the business, such as total revenue, most popular pizzas, order trends, and sales distribution. The goal was to gain insights into customer preferences and sales performance by writing SQL queries to answer key business questions.

1. RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED

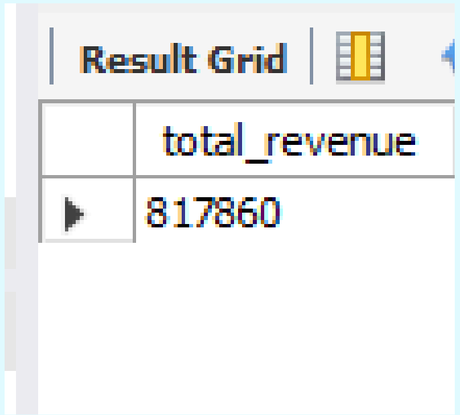
```sql
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

Result Grid

	total_orders
▶	21350

2. CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

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



3. IDENTIFY THE HIGHEST-PRICED PIZZA.

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

	name	price
▶	The Greek Pizza	35.95

4. IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

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

	size	common_size
▸	L	18526

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

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

name	Total_Quantity
The Barbecue Chicken Pizza	49574
The California Chicken Pizza	49574
The Chicken Alfredo Pizza	49574
The Chicken Pesto Pizza	49574
The Southwest Chicken Pizza	49574

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

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

	category	total_quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

7. DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

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

hours	total_orders
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

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

```
SELECT
    category, COUNT(`name`) AS `count`
FROM
    pizza_types
GROUP BY category
ORDER BY `count`;
```

	category	count
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

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

```sql
WITH order_quantity AS
    (SELECT order_date as `Day`, SUM(quantity) AS total_order FROM Orders AS o
    JOIN order_details AS od ON od.order_id = o.order_id
    GROUP by `Day`)
SELECT ROUND(AVG(total_order)) AS avg_order_per_day FROM order_quantity;
```

avg_order_per_day
138

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

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

	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5

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

```sql
SELECT
    pt.category,
    ROUND(SUM(p.price * o.quantity) / (SELECT
                    SUM(p2.price * o2.quantity)
                FROM
                    order_details o2
                        JOIN
                    pizzas p2 ON p2.pizza_id = o2.pizza_id) * 100,
            2) AS percent
FROM
    pizza_types AS pt
        JOIN
    pizzas AS p ON p.pizza_type_id = pt.pizza_type_id
        JOIN
    order_details AS o ON o.pizza_id = p.pizza_id
GROUP BY pt.category
ORDER BY percent DESC;
```

	category	percent
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

12. ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```sql
WITH Sales AS (
  SELECT o.order_date, SUM(od.quantity * p.price) 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 o.order_date
)
SELECT order_date, SUM(revenue) OVER(order by order_date) AS cum_revenue FROM Sales;
```

order_date	cum_revenue
2015-12-20	799187.9500000001
2015-12-21	801288.65
2015-12-22	803171.6
2015-12-23	805415.9
2015-12-24	807553.75
2015-12-26	809196.8
2015-12-27	810615.8
2015-12-28	812253
2015-12-29	813606.25
2015-12-30	814944.05
2015-12-31	817860.05

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

```sql
WITH ranks AS (
  SELECT pt.name, pt.category, SUM(od.quantity * p.price) as revenue,
  RANK() OVER(PARTITION BY pt.category ORDER BY SUM(od.quantity * p.price) DESC) as `rank`
  FROM pizza_types as pt
  JOIN pizzas as p ON p.pizza_type_id = pt.pizza_type_id
  JOIN order_details as od ON p.pizza_id = od.pizza_id
  Group BY pt.category, pt.name
)
SELECT name, category, revenue FROM ranks
WHERE `rank` <= 3
ORDER BY category, revenue DESC;
```

name	category	revenue
The Thai Chicken Pizza	Chicken	43434.25
The Barbecue Chicken Pizza	Chicken	42768
The California Chicken Pizza	Chicken	41409.5
The Classic Deluxe Pizza	Classic	38180.5
The Hawaiian Pizza	Classic	32273.25
The Pepperoni Pizza	Classic	30161.75
The Spicy Italian Pizza	Supreme	34831.25
The Italian Supreme Pizza	Supreme	33476.75
The Sicilian Pizza	Supreme	30940.5
The Four Cheese Pizza	Veggie	32265.70000000065
The Mexicana Pizza	Veggie	26780.75
The Five Cheese Pizza	Veggie	26066.5

THANK YOU