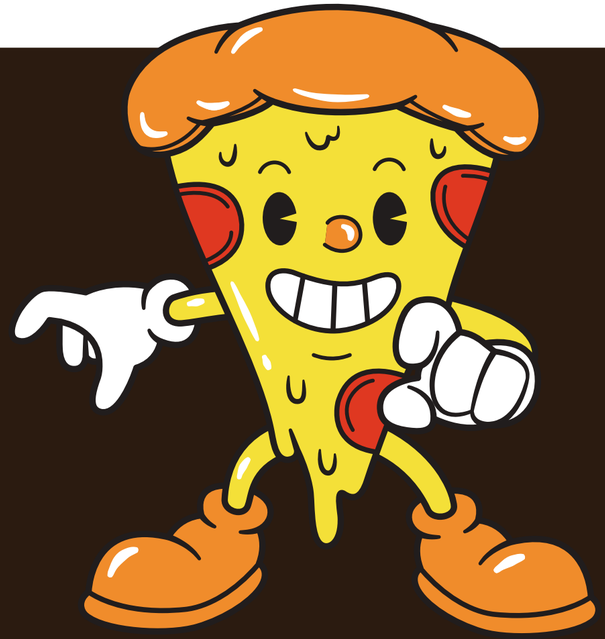
PIZZA SALES ANALYSIS



🍕 Basic Analysis Question :

- Total number of orders placed 📊
- Total revenue generated from pizza sales 💰
- Highest-priced pizza identified $
- Most common pizza size ordered 📏
- Top 5 most ordered pizza types with quantities 📋

🍕 Intermediate Analysis Question:

- Joining tables to find total quantity of each pizza category ordered 🔍
- Distribution of orders by hour of the day ⏰
- Category-wise distribution of pizzas 🍕📊
- Average number of pizzas ordered per day 📅
- Top 3 most ordered pizza types based on revenue 💵

🍕 Advanced Analysis Question:

- Percentage contribution of each pizza type to total revenue 📈
- Cumulative revenue generated over time ⏳
- Top 3 most ordered pizza types based on revenue for each pizza category 🥇🍕

1 : Total number of orders placed

```
20    -- Question 1
21
22 •  SELECT
23        COUNT(order_id)
24    FROM
25        orders;
26
```

Result Grid | Filter Rows:

	COUNT(order_id)
▶	21350

2 : Total revenue generated from pizza sales 💰

```sql
-- Question 2

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;
```

Result Grid

total_sales
817860.05

3 : Highest-priced pizza identified $

```
39    -- Question 3
40
41 •  SELECT
42        pizza_types.name, pizzas.price
43    FROM
44        pizza_types
45            JOIN
46        pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
47    ORDER BY pizzas.price DESC
48    LIMIT 1;
```

Result Grid | Filter Rows

	name	price
▶	The Greek Pizza	35.95

4 : Most common pizza size ordered 📏

```sql
-- Question 4
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

size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28

5 : Top 5 most ordered pizza types with quantities 📋

```sql
-- Question 5
SELECT
    pizza_types.name, 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 orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;
```

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

6 : Joining tables to find total quantity of each pizza category ordered 🔍

```sql
-- Question 6
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 orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC;
```

Result Grid | Filte

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

7 : Distribution of orders by hour of the day ⏰

```
86    -- Question 7
87
88 •  SELECT
89        HOUR(order_time) AS hour, COUNT(order_id) AS order_count
90    FROM
91        orders
92    GROUP BY HOUR(order_time);
```

hour	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

8 : Category-wise distribution of pizzas 🍕📊

```sql
-- Question 8

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

Result Grid | Filter Rows:

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

9 : Average number of pizzas ordered per day

```sql
-- Question 9
SELECT
    ROUND(AVG(quantity), 0) AS Avg_pizza_ordered_per_day
FROM
    (SELECT
        orders.order_date, SUM(orders_details.quantity) AS quantity
    FROM
        orders
    JOIN orders_details ON orders.order_id = orders_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

Avg_pizza_ordered_per_day
138

10 : Top 3 most ordered pizza types based on revenue 💵

```sql
-- Question 10
SELECT
    pizza_types.name,
    SUM(orders_details.quantity * pizzas.price) AS revanue
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 revanue DESC
LIMIT 3;
```

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

11 : Percentage contribution of each pizza type to total revenue 📈

```sql
-- Question 11
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 revanue
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
ORDER BY revanue DESC;
```

category	revanue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

12 : Cumulative revenue generated over time ⏳

```sql
-- Question 12
SELECT
    order_date,
    SUM(revenue) OVER (ORDER BY order_date) AS cum_revenue
FROM
    (SELECT
        orders.order_date,
        SUM(orders_details.quantity * pizzas.price) AS revenue
    FROM
        orders_details
    JOIN
        pizzas ON orders_details.pizza_id = pizzas.pizza_id
    JOIN
        orders ON orders.order_id = orders_details.order_id
    GROUP BY
        orders.order_date) AS sales;
```

order_date	cum_revenue
2015-01-01 00:00:00	2713.8500000000004
2015-01-02 00:00:00	5445.75
2015-01-03 00:00:00	8108.15
2015-01-04 00:00:00	9863.6
2015-01-05 00:00:00	11929.55
2015-01-06 00:00:00	14358.5

13 : Top most ordered pizza types based on revenue for each pizza category 🥇🍕

```sql
-- Question 13
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;
```

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