SQL CASE STUDY

DATA ANALYSIS

Introduction :

- COLLECTED DIFFERENT DATASETS OF PIZZA SALES , TYPE OF PIZZAS, CATEGORY OF PIZZAS, ORDER ID, DATE OF ORDERS, PRICE OF PIZZA ETC, USED VARIOUS SQL QUERIES AND FUNCTIONS FOR THE MANIPULATION OF VARIOUS INFORMATION FROM DATASET. TAKEN VARIOUS QUESTION AND FOUNDED THERE ANSWERS BY USING SQL AS A CASE STUDY.

Q.1: Retrieve the total number of orders placed.

```sql
1 •    create database pizza;
2 •    use pizza;
3 •    select * from orders;
4
5
6      -- Retrieve the total number of orders placed.
7 •    select count(order_id) from orders;
```

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

count(order_id)
5753

Q.2: Calculate the total revenue generated from pizza sales.

```sql
1
2  •    use pizza;
3
4      -- Calculate the total revenue generated from pizza sales.
5  •    SELECT
6          ROUND(SUM(order_details.quantity * pizzas.price),
7                  2) AS Total_sales
8      FROM
9          order_details
10             JOIN
11         pizzas ON pizzas.pizza_id = order_details.pizza_id
12
13  |
```

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

Total_sales
283082.2

Q.3: Identify the highest-priced pizza.

```sql
1 •    use pizza;
2
3      -- Identify the highest-priced pizza.
4 •    SELECT
5          pizza_types.name, pizzas.price
6      FROM
7          pizza_types
8              JOIN
9          pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
10     ORDER BY pizzas.price DESC limit 1;
11
12
```

name	price
The Greek Pizza	35.95

Q.4: Identify the most common pizza size ordered.

```sql
1 •    use pizza;
2
3      -- Identify the most common pizza size ordered.
4 •    SELECT
5          pizzas.size,
6          COUNT(order_details.order_details_id) AS order_count
7      FROM
8          pizzas
9              JOIN
10         order_details ON pizzas.pizza_id = order_details.pizza_id
11     GROUP BY pizzas.size
12     ORDER BY order_count DESC
13     LIMIT 1;
```

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

size	order_count
L	6410

Q.5: List the top 5 most ordered pizza types along with their quantities.

```sql
2
3       -- List the top 5 most ordered pizza types along with their quantities.
4 •     SELECT
5           pizza_types.name, SUM(order_details.quantity) AS quantity
6       FROM
7           pizza_types
8               JOIN
9           pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
10              JOIN
11          order_details ON order_details.pizza_id = pizzas.pizza_id
12      GROUP BY pizza_types.name
13      ORDER BY quantity DESC
14      LIMIT 5;
```

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

name	quantity
The Barbecue Chicken Pizza	874
The Hawaiian Pizza	846
The Pepperoni Pizza	838
The Classic Deluxe Pizza	813
The Thai Chicken Pizza	808

Q.6: Join the necessary tables to find the total quantity of each pizza category ordered.

```sql
1 •   use pizza;
2
3     -- Join the necessary tables to find the total quantity of each pizza category ordered.
4 •   SELECT
5         pizza_types.category,
6         SUM(order_details.quantity) AS quantity
7     FROM
8         pizza_types
9             JOIN
10        pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
11            JOIN
12        order_details ON order_details.pizza_id = pizzas.pizza_id
13    GROUP BY pizza_types.category
14    ORDER BY quantity DESC;
```

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

category	quantity
Classic	5093
Supreme	4133
Veggie	4090
Chicken	3844

Q.7: Join relevant tables to find the category-wise distribution of pizzas.

```sql
1 •    use pizza;
2
3      -- Join relevant tables to find the category-wise distribution of pizzas.
4 •    SELECT
5          category, COUNT(name)
6      FROM
7          pizza_types
8      GROUP BY category;
```

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

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

Q.8: Group the orders by date and calculate the average number of pizzas ordered per day.

```sql
1 •    use pizza;
2
3      -- Group the orders by date and calculate the average number of pizzas ordered per day.
4 •    SELECT
5          AVG(quantity)
6      FROM
7          (SELECT
8              orders.date, SUM(order_details.quantity) AS quantity
9          FROM
10             orders
11         JOIN order_details ON orders.order_id = order_details.order_id
12         GROUP BY orders.date) AS order_quantity;
```

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

avg(quantity)
137.9278

Q.9: Determine the top 3 most ordered pizza types based on revenue.

```sql
1    use pizza;
2
3    -- Determine the top 3 most ordered pizza types based on revenue.
4    SELECT
5        pizza_types.name,
6        SUM(order_details.quantity * pizzas.price) AS revenue
7    FROM
8        pizza_types
9            JOIN
10       pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
11           JOIN
12       order_details ON order_details.pizza_id = pizzas.pizza_id
13   GROUP BY pizza_types.name
14   ORDER BY revenue DESC
15   LIMIT 3;
```

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

name	revenue
The Barbecue Chicken Pizza	15447.5
The Thai Chicken Pizza	14702
The California Chicken Pizza	14114

Q.10: Calculate the percentage contribution of each pizza type to total revenue.

```sql
1 •   use pizza;
2
3     -- Calculate the percentage contribution of each pizza type to total revenue.
4 •   select pizza_types.category,
5 ⊖   round(sum(order_details.quantity * pizzas.price)/ (select round(sum(order_details.quantity * pizzas.price),2) as total_sales
6
7     from order_details
8     join pizzas
9     on pizzas.pizza_id = order_details.pizza_id) * 100,2) as revenue
10
11    from pizza_types join pizzas
12    on pizza_types.pizza_type_id = pizzas.pizza_type_id
13    join order_details
14    on order_details.pizza_id = pizzas.pizza_id
15    group by pizza_types.category order by revenue desc ;
```

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

category	revenue
Classic	26.61
Supreme	25.33
Chicken	24.08
Veggie	23.97

THANKS