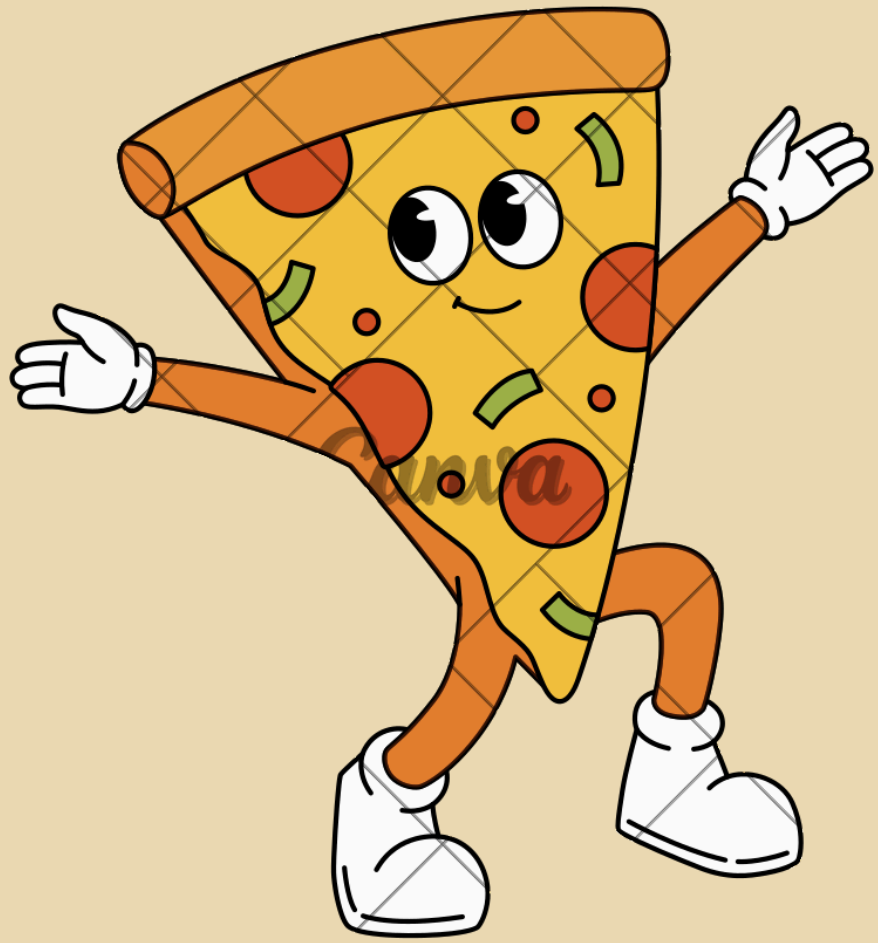
SQL BASED ON PIZZA SALES



Hello !

My self Sayali Mahendra Ghadi. In this project i have utilized SQL queries to solve questions that are related to pizza sales.

Description

I created database pizzahut and i took the dataset from kaggle, extract the data set in desktop. Then under the database i created tables orders, orders_details, pizzas, pizza_types and in each tables i imported data and after that i started the run queries.

Retrieve the total number of orders placed.

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

Result Grid

total_orders
4943

Calculate the total revenue generated from pizza sales.

```
SELECT
    ROUND(SUM(orders_details.quantity * pizzas.price),
            2) AS total_revenue
FROM
    orders_details
        JOIN
pizzas ON pizzas.pizza_id = orders_details.pizza_id
```

Result Grid

total_revenue
21932.35

Identify the highest-priced pizza.

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

Result Grid | Filter R

	name	price
▶	The Greek Pizza	35.95

Identify the most common pizza size ordered.

```sql
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 | Filt

size	order_count
L	491
M	420
S	399
XL	8

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

```sql
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;
```

Result Grid | Filter Rows:

name	quantity
The Barbecue Chicken Pizza	141
The California Chicken Pizza	113
The Big Meat Pizza	98
The Classic Deluxe Pizza	94
The Hawaiian Pizza	66

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

```sql
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

category	quantity
Classic	432
Chicken	362
Veggie	283
Supreme	262

Determine the distribution of orders by hour of the day.

```sql
SELECT
    HOUR(time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(time);
```

Result Grid

hour	order_count
11	271
12	564
13	544
14	435
15	360
16	434
17	529
18	548
19	470

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

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

Result Grid | Filter

category	pizza_count
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(quantity), 0) as avg_pizza_ordered
FROM
    (SELECT
        orders.date, SUM(orders_details.quantity) AS quantity
    FROM
        orders
    JOIN orders_details ON orders.order_id = orders_details.order_id
    GROUP BY orders.date) AS order_quantity;
```

Result Grid

avg_pizza_ordered
158

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

```sql
SELECT
    pizza_types.name,
    SUM(orders_details.quantity * pizzas.price) 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.name
ORDER BY revenue DESC
LIMIT 3;
```

Result Grid | Filter Rows:

name	revenue
The Barbecue Chicken Pizza	2513.75
The California Chicken Pizza	1932.75
The Classic Deluxe Pizza	1478

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

```
select pizza_types.category, (sum(orders_details.quantity * pizzas.price) / (SELECT
    ROUND(SUM(orders_details.quantity * pizzas.price),
            2) AS total_revenue
FROM
    orders_details
        JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizza_id) )* 100 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 order by revenue desc;
```

Result Grid | Filter Rows:

category	revenue
Chicken	28.831839725337232
Classic	28.162736779232507
Veggie	21.985560142893927
Supreme	21.01986335253632 3

Analyze the cumulative revenue generated over time.

```sql
select date, sum(revenue) over(order by date) as cum_revenue
from
(select orders.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.date) as sales;
```

Result Grid | Filter Rows:

date	cum_revenue
2015-01-01	3409.049999999997
2015-01-02	6737.549999999999
2015-01-03	10124.55
2015-01-04	12780.449999999999
2015-01-05	15477.3
2015-01-06	18696.899999999998
2015-01-07	21585.75
2015-01-08	25057.2

Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```sql
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:

name	revenue
The Barbecue Chicken Pizza	2513.75
The California Chicken Pizza	1932.75
The Chicken Alfredo Pizza	532.75
The Classic Deluxe Pizza	1478
The Big Meat Pizza	1176
The Hawaiian Pizza	879.75
The Italian Supreme Pizza	968.25
The Pepper Salami Pizza	642.5

Thank you!