A Pizza_Store Dataset Analysis

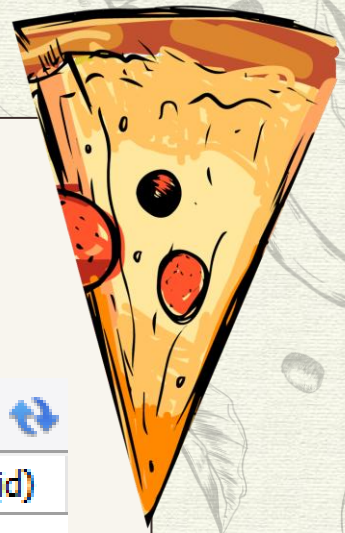
Using MySQL

LinkedIn
Gaurav Gupta | LinkedIn

Tables In dataset

Pizzas	Pizza_types	Orders	Order_details
• pizza_id • pizza_type_id • size(L>M>S) • price	• pizza_type_id • name • category • ingredients	• order_id • order_date • order_time	• order_details_id • order_id • pizza_id • Quantity

Q1-Retrieve the total number of orders placed.



Query

```
SELECT
    COUNT(order_id)
FROM
    orders;
```

Result

Result Grid

	COUNT(order_id)
▶	21350

Q2 Calculate the total revenue generated from pizza sales.

Query

```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            0) AS total_revene
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id;
```

Result

Result Grid

	total_revene
▶	817860

Q3 Determine the distribution of orders by hour of the day.

Query

```
SELECT
    HOUR(order_time) AS Order_hour,
    SUM(order_id) AS total_orders
FROM
    orders
GROUP BY HOUR(order_time);
```

Result

Result Grid | Filter Rows:

Order_hour	total_orders
11	13336362
12	26929470
13	26615205
14	14867592
15	15634879
16	20551671
17	24312547
18	25808745
19	21634044
20	17668990
21	12868673
22	7269872
23	330700
10	73999
9	19176

Q4 Join relevant tables to find the category-wise distribution of pizzas.

Query

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

Result

Result Grid | Filter Rows:

	category	numbers
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

Q5 Identify the highest-priced pizza.

Query

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

Result

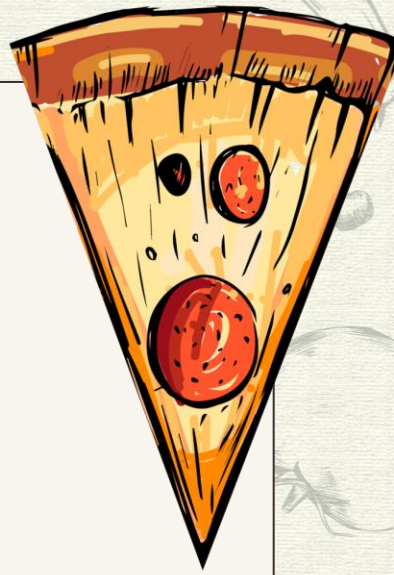
Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95

Q6 Identify the most common pizza size ordered.

Query

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



Result

Result Grid | Filter Rows:

	Viral_pizza	size
▶	18526	L

Q7 List the top 5 most ordered pizza types along with their quantities.

Query

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

Result

Result Grid | Filter Rows:

name	total
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

Q8 Join the necessary tables to find the total quantity of each pizza category ordered.

Query

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

Result

Result Grid | Filter Rows:

category	total_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Q9 Group the orders by date and calculate the average number of pizzas ordered per day.

Query

```
SELECT
    ROUND(AVG(total_sale), 0) as average_pizza
FROM
    (SELECT
        orders.order_date,
            COUNT(order_details.quantity) AS Total_sale
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS DATA;
```

Result

Result Grid | Filter Rows:

	average_pizza
▶	136

Q10 Determine the top 3 most ordered pizza types based on revenue.

Query

```
SELECT
    pizza_types.name AS Pizza_name,
    ROUND(SUM(order_details.quantity * pizzas.price),
            0) AS tr
FROM
    pizzas
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
GROUP BY Pizza_name
ORDER BY tr DESC
LIMIT 3;
```

Result

Result Grid | Filter Rows:

Pizza_name	tr
The Thai Chicken Pizza	43434
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41410

Q11 Calculate the percentage contribution of each pizza type to total revenue.

Query

```
SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
                    ROUND(SUM(order_details.quantity * pizzas.price),
                                0) AS total_sales
                FROM
                    order_details
                        JOIN
                    pizzas ON order_details.pizza_id = pizzas.pizza_id) * 100,
            2) AS Something
FROM
    pizzas
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category;
```

Result

Result Grid | Filter Rows:

category	Something
Classic	26.91
Veggie	23.68
Supreme	25.46
Chicken	23.96

Q12 Analyze the cumulative revenue generated over time.

Query

```
select order_date,
sum(revene_perday) over(order by order_date)as cum_revene
from
(select orders.order_date, sum(order_details.quantity * pizzas.price) as revene_perday
from pizzas
join order_details
on pizzas.pizza_id=order_details.pizza_id
join orders
on orders.order_id=order_details.order_id
group by orders.order_date) AS View_table;
```

Result

Result Grid | Filter Rows:

order_date	cum_revene
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
2015-01-09	21526.4
2015-01-10	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7

And so on…..

Q13 Determine the top 3 most ordered pizza types based on revenue for each pizza category.

Query

```
SELECT category, name , revene_pc
FROM
(SELECT category, name, revene_pc,
RANK() OVER(PARTITION BY category ORDER BY revene_pc DESC)
AS rank_id
FROM
(SELECT
    pizza_types.name,
    pizza_types.category,
    SUM(order_details.quantity * pizzas.price) AS Revene_pc
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name , pizza_types.category) as a) AS B
WHERE rank_id <=3;
```

Result

Result Grid | Filter Rows: | Export:

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

THANK_YOU

LinkedIn
Gaurav Gupta | LinkedIn