PIZZA SALES ANALYSIS

ABOUT PIZZA ANALYSIS

Introduction

Hello I’m Tanmay Kawase .In this project, I conducted a comprehensive analysis of pizza sales data using MySQL to uncover key insights into customer preferences, top-performing categories, and overall revenue trends. By leveraging SQL queries, I explored patterns in order volume, sales distribution by category, and peak sales periods to support data-driven decision-making for menu optimization and marketing strategies.

ABOUT TABLE

order_details_id	order_id	pizza_id	quantity
1	1	hawaiian_m	1
2	2	classic_dlx_m	1
3	2	five_cheese_l	1
4	2	ital_supr_l	1
5	2	mexicana_m	1
6	2	thai_ckn_l	1
7	3	ital_supr_m	1

Order Details

pizza_id	pizza_type_id	size	price
bbq_ckn_s	bbq_ckn	S	12.75
bbq_ckn_m	bbq_ckn	M	16.75
bbq_ckn_l	bbq_ckn	L	20.75
cali_ckn_s	cali_ckn	S	12.75
cali_ckn_m	cali_ckn	M	16.75
cali_ckn_l	cali_ckn	L	20.75

Pizza

order_id	order_date	order_time
1	2015-01-01	11:38:36
2	2015-01-01	11:57:40
3	2015-01-01	12:12:28
4	2015-01-01	12:16:31
5	2015-01-01	12:21:30
6	2015-01-01	12:29:36

Order

bbq_ckn	The Barbecue Chicken Pizza	Chicken	Barbecued Chicken, Red Peppers, Green Peppe..
cali_ckn	The California Chicken Pizza	Chicken	Chicken, Artichoke, Spinach, Garlic, Jalapeno P...
ckn_alfredo	The Chicken Alfredo Pizza	Chicken	Chicken, Red Onions, Red Peppers, Mushrooms..
ckn_pesto	The Chicken Pesto Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Spinach, Garl..
southw_ckn	The Southwest Chicken Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Red Onions, ..
thai_ckn	The Thai Chicken Pizza	Chicken	Chicken, Pineapple, Tomatoes, Red Peppers, T...
big_meat	The Big Meat Pizza	Classic	Bacon, Pepperoni, Italian Sausage, Chorizo Sau..
classic_dlx	The Classic Deluxe Pizza	Classic	Pepperoni, Mushrooms, Red Onions, Red Peppe..
hawaiian	The Hawaiian Pizza	Classic	Sliced Ham, Pineapple, Mozzarella Cheese
ital_cpclo	The Italian Capocollo Pizza	Classic	Capocollo, Red Peppers, Tomatoes, Goat Chee...
napolitana	The Napolitana Pizza	Classic	Tomatoes, Anchovies, Green Olives, Red Onion..
pep_msh_pep	The Pepperoni, Mushroom, ...	Classic	Pepperoni, Mushrooms, Green Peppers
pepperoni	The Pepperoni Pizza	Classic	Mozzarella Cheese, Pepperoni

za types 1 ×

Pizza Type

BASIC

- **1. Retrieve the total number of orders placed.**

```
select count(order_id) as total_orders from orders;
```

	total_orders
▶	21350

- **2. Calculate the total revenue generated from pizza sales.**

```
select
round(sum(order_details.quantity * pizzas.price), 2) as total_revenue
from order_details join pizzas
on pizzas.pizza_id = order_details.pizza_id;
```

total_revenue
817860.05

3. Identify the highest-priced pizza.

```
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;
```

	name	price
▶	The Greek Pizza	35.95

4. Identify the most common pizza size ordered.

```
select pizzas.size, count(order_details.order_details_id) as order_size
from pizzas join order_details
on pizzas.pizza_id = order_details.pizza_id
group by pizzas.size;
```

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

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

```sql
select pizza_types.name,
sum(order_details.quantity) as quantity
from pizza_types join pizzas
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 quantity desc limit 5;
```

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

INTERMEDIATE

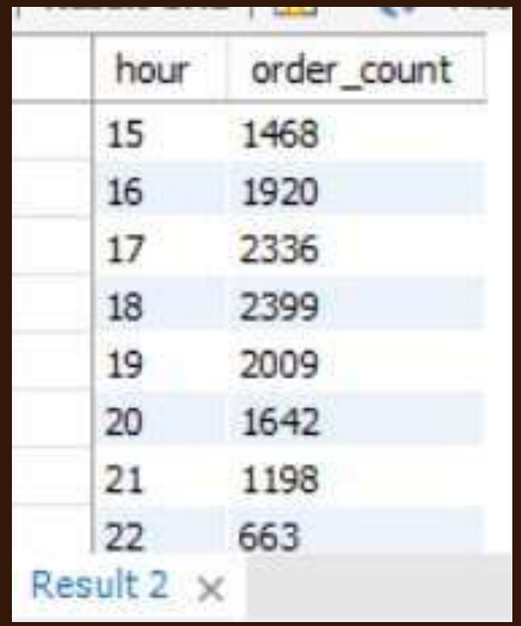
1. Join the necessary tables to find the total quantity of each pizza category ordered.

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

	cat	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

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

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



hour	order_count
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663

Result 2

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

```
select category, count(name) from pizza_types
group by category;
```

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

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

```
elect round(avg(quantity),0) from
select orders.order_date, sum(order_details.quantity) as quanti
rom orders join order_details
n orders.order_id = order_details.order_id
roup by orders.order_date) as order_quantity;
```

	round(avg(quantity),0)
▶	138

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

```sql
select pizza_types.name,
sum(order_details.quantity * pizzas.price) as Revenue
from pizza_types join pizzas
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.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

ADVANCED

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

```sql
SELECT
    pizza_types.category,
    ROUND(
        SUM(order_details.quantity * pizzas.price) / (
            SELECT
                ROUND(SUM(order_details.quantity * pizzas.price), 2)
            FROM
                order_details
            JOIN
                pizzas ON pizzas.pizza_id = order_details.pizza_id) * 100, 2) AS revenue
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.category
ORDER BY
    revenue DESC;
```

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

2. Analyze the cumulative revenue generated over time.

```
select order_date,
sum(revenue) over(order by order_date) as cum_evenue
from
(select orders.order_date,
sum(order_details.quantity * pizzas.price) as revenue
from order_details join pizzas
on order_details.pizza_id = pizzas.pizza_id
join orders
on orders.order_id = order_details.order_id
group by orders.order_date) as Sales;
```

order_date	cum_evenue
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

Result 8

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

```sql
select category, 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((order_details.quantity) * pizzas.price) as revenue
from pizza_types join pizzas
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.category, pizza_types.name) as a) as b
where rn <= 3;
```

category	name	revenue
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 FOR ATTENTION