MySQL

SQL REPORT ON

PIZZA SALES

Ritam Bhowmick

About
The report

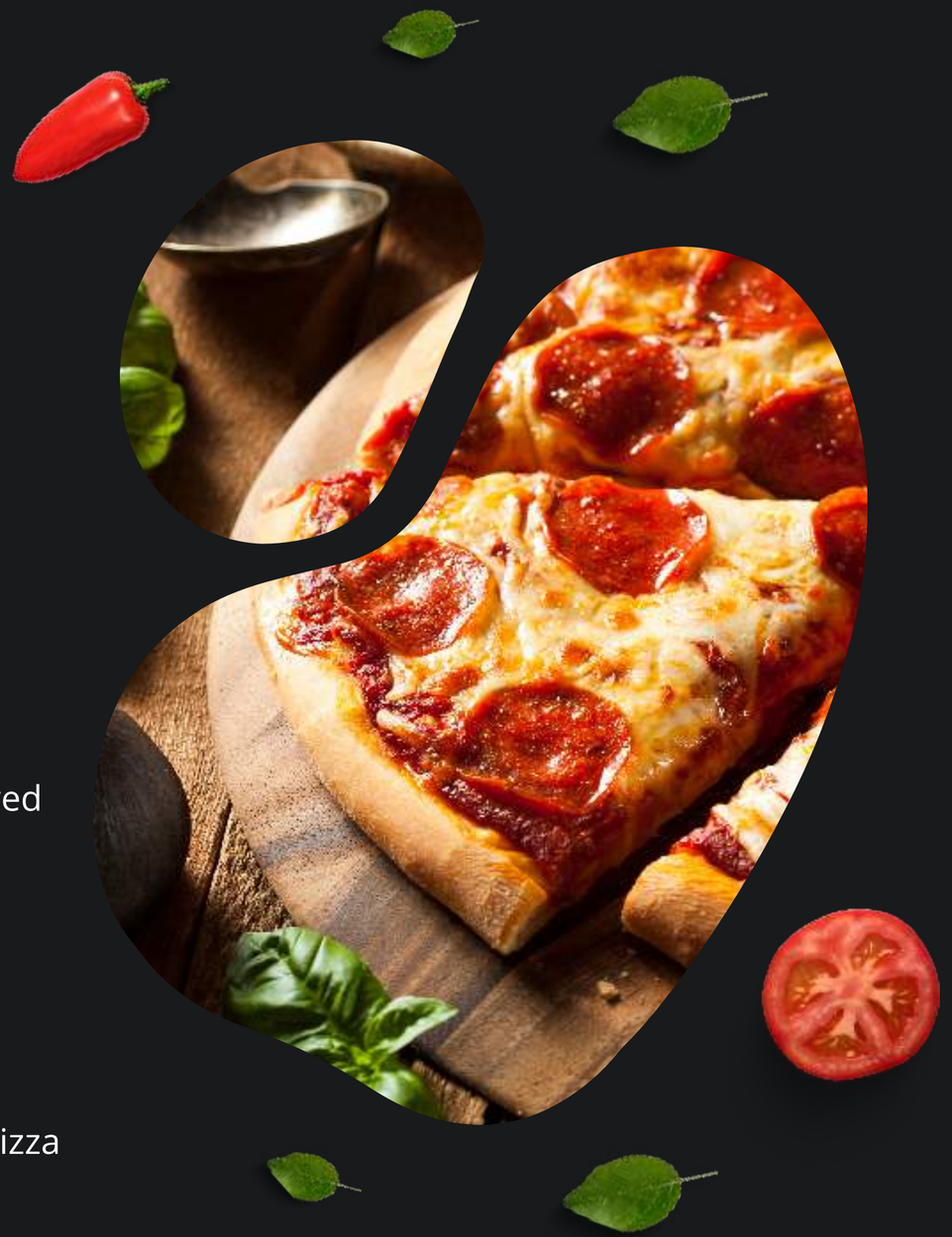
Hi guys My name is Ritam bhowmick and here I have created sample pizza sales report using SQL queries

PROBLEMS MENU

Basic:
Retrieve the total number of orders placed.
Calculate the total revenue generated from pizza sales.
Identify the highest-priced pizza.
Identify the most common pizza size ordered.
List the top 5 most ordered pizza types along with their quantities.

Intermediate:
Join the necessary tables to find the total quantity of each pizza category ordered.
Determine the distribution of orders by hour of the day.
Join relevant tables to find the category-wise distribution of pizzas.
Group the orders by date and calculate the average number of pizzas ordered per day.
Determine the top 3 most ordered pizza types based on revenue.

Advanced:
Calculate the percentage contribution of each pizza type to total revenue.
Analyze the cumulative revenue generated over time.
Determine the top 3 most ordered pizza types based on revenue for each pizza category.



Retrieve the total number of orders placed.

```sql
select count(order_id) from orders;
```

OUTPUT

	count(order_id)
▶	21350

Calculate the total revenue generated from pizza sales.

```sql
SELECT SUM(order_details.quantity * pizzas.price) AS total_price
FROM order_details
JOIN pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

total_price
817860.049999993

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 Rows:

name	price
The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

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

Result Grid | Filter Rows:

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

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

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

```sql
SELECT pizza_types.category, 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.category order by quantity desc;
```

Result Grid | Filter R

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Determine the distribution of orders by hour of the day.

```sql
SELECT DATE_FORMAT(orders.order_time, '%H') AS hour, COUNT(*) AS order_count
FROM orders
GROUP BY DATE_FORMAT(orders.order_time, '%H')
ORDER BY hour;
```

09	1
10	8
11	1231
12	2520
13	2455

	hour	order_count
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920

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

```sql
SELECT
    pizza_types.category,
    COUNT(pizza_types.name) AS order_count
FROM
    pizza_types
GROUP BY
    pizza_types.category
```

category	order_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 AVG(daily_quantity) AS average_pizzas_per_day
FROM (
    SELECT
        orders.order_date,
        SUM(order_details.quantity) AS daily_quantity
    FROM
        orders
    JOIN
        order_details ON orders.order_id = order_details.order_id
    GROUP BY
        orders.order_date
) AS daily_totals;
```

average_pizzas_per_day
138.4749

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

```sql
SELECT pizza_types.name AS pizza_type,
       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;
```

pizza_type	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

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

```sql
SELECT pizza_types.category AS pizza_type,
      round( (SUM(order_details.quantity * pizzas.price) / (SELECT SUM(order_details.quantity * pizzas.price) AS total_price
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 order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC;
```

pizza_type	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Analyze the cumulative revenue generated over time.

```sql
SELECT orders.order_date ,
       SUM(order_details.quantity * pizzas.price) AS daily_revenue,
       SUM(SUM(order_details.quantity * pizzas.price))
       OVER (ORDER BY orders.order_date ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS cumulative_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
ORDER BY orders.order_date;
```

order_date	daily_revenue	cumulative_revenue
2015-01-01	2713.850000000004	2713.850000000004
2015-01-02	2731.8999999999996	5445.75
2015-01-03	2662.3999999999996	8108.15
2015-01-04	1755.4500000000003	9863.6
2015-01-05	2065.95	11929.55

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((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;
```

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

Thank
For You