SALES REPORT SQL

06-092024

INTRODUCTION

In this report, SQL queries were employed to analyze pizza sales data, addressing various complexities. Basic queries covered total orders, revenue, and pizza preferences, while intermediate queries involved joins and distributions by hour and date. Advanced queries provided insights into revenue contributions and trends, highlighting top pizza types across categories.

AGENDA

01 Retrieve the total number of orders placed.

02 Calculate the total revenue generated from pizza sales.

03 Identify the highest-priced pizza.

04 Identify the most common pizza size ordered.

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

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

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

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

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

Ingoude Company

AGENDA

10 Determine the top3 most ordered pizza types based on revenue.

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

12 Analyze the cumulative revenue generated over time.

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

sql quary 1

```sql
-- Retrieve the total number of orders placed?

select * from orders;

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

	total_orders
▶	15682

sql quary 2

```sql
-- Calculate the total revenue generated from pizza sales?

SELECT
    SUM(order_details.quantity * pizzas.price) as revanue
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id;
```

revanue
817860.049999993

sql quary 3

```sql
-- Identify the highest-priced pizza?
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

name	price
The Greek Pizza	35.95

sql quary 4

```sql
-- Identify the most common pizza size ordered?

SELECT

    quantity, COUNT(order_details_id) AS total_orders

FROM

    order_details

GROUP BY quantity;
```

quantity	total_orders
1	47693
2	903
3	21
4	3

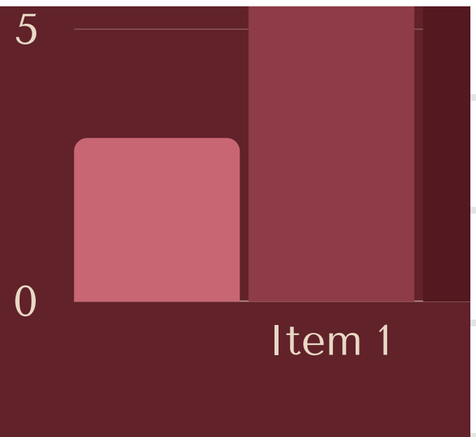
sql quary 5

```sql
-- List the top 5 most ordered pizza types along with their quantities?
SELECT
    pizzas.pizza_id,
    COUNT(order_details.quantity) AS order_count
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_id
ORDER BY order_count DESC
LIMIT 5;
```

pizza_id	order_count
big_meat_s	1811
thai_ckn_l	1365
five_cheese_l	1359
four_cheese_l	1273
classic_dlx_m	1159

sql quary 6

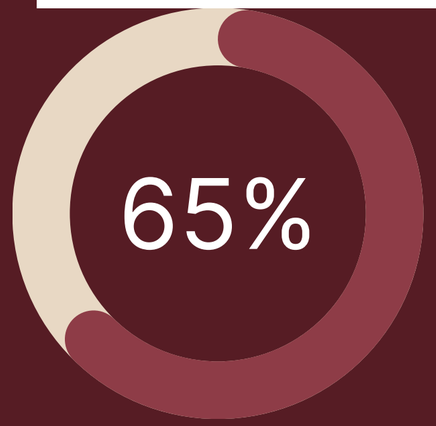
```sql
-- Join the necessary tables to find the total quantity of each pizza category ordered? (Intermediate questions)
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY category
ORDER BY quantity DESC;
```



category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

sql quary 7

```sql
-- 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)
ORDER BY COUNT(order_id) DESC;
```



hour	order_count
12	1831
13	1775
18	1765
17	1744
19	1455
16	1414
20	1178
14	1138
15	1082

sql quary 8

```sql
-- 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)
ORDER BY COUNT(order_id) DESC;
```

hour	order_count
16	1414
20	1178
14	1138
15	1082
11	911
21	886
22	475
23	21
10	7

sql quary 9

```sql
-- Join relevant tables to find the category-wise distribution of pizzas?
SELECT
    category, count(name) as count_name
FROM
    pizza_types
GROUP BY category;
```

category	count_name
Chicken	6
Classic	8
Supreme	9
Veggie	9

sql quary 10

```sql
-- Join relevant tables to find the category-wise distribution of pizzas?
SELECT
    category, count(name) as count_name
FROM
    pizza_types
GROUP BY category;
```

category	count_name
Chicken	6
Classic	8
Supreme	9
Veggie	9

sql quary 11

```sql
-- Group the orders by date and calculate the average number of pizzas ordered per day?
SELECT
    ROUND(AVG(quantity), 0) as avg_ordered_pizza_per_day
FROM
    (SELECT
        DAY(orders.order_date) AS day_wise,
            SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS orders_quantity;
```

	avg_ordered_pizza_per_day
▶	138

sql quary 12

```sql
-- Determine the top 3 most ordered pizza types based on revenue?
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revanue
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 revanue DESC
LIMIT 3;
```

name	revanue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

sql quary 13

```sql
-- Calculate the percentage contribution of each pizza type to total revenue?(Advanced questions)
SELECT
    pizza_types.category,
   (SUM(order_details.quantity * pizzas.price) / (SELECT
    round(SUM(order_details.quantity * pizzas.price))
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id )) *100 as revanue
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 revanue DESC;
```

	category	revanue
▶	Classic	26.905961900569793
	Supreme	25.456312816374222
	Chicken	23.955139021348398
	Veggie	23.682592375223507

sql quary 14

```sql
-- Analyze the cumulative revenue generated over time?
select order_date , sum(revanue) over(order by order_date) as cum_revanue from
(SELECT
    orders.order_date,
    SUM(order_details.quantity * pizzas.price) AS revanue
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_revanue
2015-01-01	2713.85000000
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

sql quary 15

```sql
-- Determine the top 3 most ordered pizza types based on revenue for each pizza category?
select name , revanue from
(select category , name , revanue ,rank() over(partition by category order by revanue desc) as rn from
(SELECT
    pizza_types.name,pizza_types.category,
    SUM((order_details.quantity) * pizzas.price)
     AS revanue
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,pizza_types.category ) as a) as b where rn <= 3 ;
```

name	revanue
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
The Pepperoni Pizza	30161.75
The Spicy Italian Pizza	34831.25
The Italian Supreme Pizza	33476.75
The Sicilian Pizza	30940.5
The Four Cheese Pizza	32265.70000
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.5

Yours faithfully Lokesh chopra

THANK YOU

06-09-2024