SQL PROJECT ON PIZZA SALES

11 July, 2024

INTRODUCTION

Hello, My name is Abhay Pratap Singh. In this Project I had utilized SQL query to solve Question that were related to Pizza Sales.

RETRIEVE THE TOTAL NUMBER OF ORDERS PALCED.

```sql
-- Retrieve the total number of orders palced.



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

total_orders
21350

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

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

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

total_sales
356723.85

IDENTIFY THE HIGHEST-PRICED PIZZA.

```sql
-- 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

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

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

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;
```

size	order_count
L	7893
M	6827
S	6507
XL	115
XXL	8

LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

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

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 Barbecue Chicken Pizza	2329
The California Chicken Pizza	1721
The Big Meat Pizza	1695
The Classic Deluxe Pizza	1506
The Hawaiian Pizza	1166

JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

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

SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS total_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 total_quantity DESC;
```

category	total_quantity
Classic	6975
Chicken	6238
Veggie	4357
Supreme	4134

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```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);
```

hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198

JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

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

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

category	total_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
-- Group the orders by date and calculate the average number of pizzas ordered per day.

SELECT
    ROUND(AVG(quantity), 0) as avg_pizza_ordered_per_day
FROM
    (SELECT
        orders.order_date, 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 order_quantity;
```

avg_pizza_ordered_per_day
61

DETERMINE THETOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

```sql
-- Determine thetop 3 most ordered pizza types based on revenue.

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 Barbecue Chicken Pizza	41230.75
The California Chicken Pizza	30102.75
The Classic Deluxe Pizza	23548

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```sql
--  Calculate the percentage contribution of each pizza type to total revenue

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

category	revenue
Chicken	30.8
Classic	28.06
Supreme	20.63
Veggie	20.51

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```sql
-- Analyze the cumulative revenue generated over time.

select  order_date,
sum(revenue) over (order by order_date) as cum_revenue
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 order by order_date)
 as sales;
```

order_date	cum_revenue
2015-01-01	1136.3500000000001
2015-01-02	2245.8500000000004
2015-01-03	3374.8500000000004
2015-01-04	4260.150000000001
2015-01-05	5159.1
2015-01-06	6232.3
2015-01-07	7195.25
2015-01-08	8352.4
2015-01-09	9386.85
2015-01-10	10501.15
2015-01-11	11341.949999999999

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGROY.

```sql
-- Determine the top 3 most ordered pizza types based on revenue for each pizza categroy.


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 Barbecue Chicken Pizza	41230.75
The California Chicken Pizza	30102.75
The Chicken Alfredo Pizza	11606
The Classic Deluxe Pizza	23548
The Big Meat Pizza	20340
The Hawaiian Pizza	15546.5
The Italian Supreme Pizza	14359.75
The Calabrese Pizza	12463.5
The Brie Carre Pizza	9057.949999999937
The Four Cheese Pizza	17526.550000000232
The Five Cheese Pizza	15299.5
The Mexicana Pizza	9428.5

THANK YOU