SQL PROJECT: ANALYZING PIZZA SALES DATA

26 AUG, 2024

INTRODUCTION

This project focuses on analyzing a large pizza sales dataset using SQL queries to extract valuable business insights. The dataset contains detailed information on orders, pizza types, sizes, and their corresponding revenue, allowing for a comprehensive analysis of customer preferences and sales trends. Through a series of queries ranging from basic to advanced, this project explores various aspects of pizza sales, including the total revenue generated, most ordered pizzas, order patterns by time, and category-specific insights. The goal is to derive actionable data-driven conclusions to enhance decision-making in the food and beverage industry.

CALCULATED TOTAL REVENUE GENERATED FROM PIZZA SALES.

```sql
-- CALCULATED TOTAL REVENUE GENERATED FROM PIZZA SALES.
SELECT
    ROUND(SUM((QUANTITY * PRICE)), 2) AS TOTAL_REVENUE
FROM
    ORDER_DETAILS
        JOIN
    PIZZAS ON ORDER_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID;
```

TOTAL_REVENUE
817860.05

```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;
```

-- IDENTIFY THE HIGHEST PRICE PIZZA

	NAME	PRICE
▶	The Greek Pizza	35.95

list the top 5 ordered pizza along with their types and quantities

Limit to 1000 rows

```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;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

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

```sql
-- IDENTIFY THE MOST COMMON PIZZA ORDERED
SELECT
    PIZZAS.SIZE,
    COUNT(order_details.ORDER_DETAILS_ID) AS ORDER_COUNT
FROM
    ORDER_DETAILS
        JOIN
    PIZZAS ON ORDER_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID
GROUP BY PIZZAS.SIZE
ORDER BY ORDER_COUNT DESC;
```

-- IDENTIFY THE HIGHEST PRICE PIZZA

SIZE	ORDER_COUNT
L	18526
M	15385
S	14137
XL	544
XXL	28

JOIN THE NECCSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEFORY ORDERED.

Limit to 1000 rows

```
-- JOIN THE NECCSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEFORY O
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 Rows: | Export: | Wrap Cell Content:

CATEGORY	QUANTITY
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Determine the distribution of orders by hour of the day

Limit to 1000 rows

```sql
SELECT
    HOUR(ORDER_TIME) AS HOUR,
    COUNT(ORDER_ID) AS ORDERS_COUNT
FROM
    ORDERS
GROUP BY
    HOUR(ORDER_TIME)
ORDER BY
    HOUR;
```

Result Grid | Filter Rows: | Export: | Wrap Cell C

HOUR	ORDERS_COUNT
9	1
10	8
11	1231
12	2520
13	2455

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

```
-- Join relevant tables to find the category-wise distribution of pizzas.
SELECT CATEGORY, COUNT(NAME) FROM PIZZA_TYPES
GROUP BY CATEGORY;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

CATEGORY	COUNT(NAME)
Chicken	6
Classic	8
Supreme	9
Veggie	9

```
SELECT
    ROUND(AVG(QUANTITY),0)
FROM
    (SELECT
        SUM(ORDER_DETAILS.QUANTITY) AS QUANTITY, ORDERS.ORDER_DATE
    FROM
        ORDER_DETAILS
    JOIN ORDERS ON ORDER_DETAILS.ORDER_ID = ORDERS.ORDER_ID
    GROUP BY ORDERS.ORDER_DATE) AS ORDER_QUANTITY;
```

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


```
    ROUND(AVG(QUANTITY),0)
FROM
    (SELECT
        SUM(ORDER_DETAILS.QUANTITY) AS QUANTITY, ORDERS.ORDER_DATE
    FROM
        ORDER_DETAILS
    JOIN ORDERS ON ORDER_DETAILS.ORDER_ID = ORDERS.ORDER_ID
    GROUP BY ORDERS.ORDER_DATE) AS ORDER_QUANTITY;
```


ROUND(AVG(QUANTITY),0)
138

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 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 REVENUE DESC LIMIT 3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Cont

NAME	REVENUE
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

```sql
WITH REVENUE_CTE AS (
    SELECT
        PIZZA_TYPES.CATEGORY,
        SUM(ORDER_DETAILS.QUANTITY * PIZZAS.PRICE) AS CATEGORY_REVENUE
    FROM
        PIZZA_TYPES
    JOIN
  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
TOTAL_REVENUE AS (
    SELECT SUM(CATEGORY_REVENUE) AS TOTAL
    FROM REVENUE_CTE
)
SELECT
    REVENUE_CTE.CATEGORY,
    REVENUE_CTE.CATEGORY_REVENUE,
    ROUND((REVENUE_CTE.CATEGORY_REVENUE / TOTAL_REVENUE.TOTAL) * 100, 2) AS PERCENTAGE_CONTRIBUTION
FROM
    REVENUE_CTE, TOTAL_REVENUE
ORDER BY
    REVENUE_CTE.CATEGORY_REVENUE DESC;
```

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

CATEGORY	CATEGORY_REVENUE	PERCENTAGE_CONTRIBUTION
Classic	220053.1000000001	26.91
Supreme	208196.99999999822	25.46
Chicken	195919.5	23.96
Veggie	193690.45000000298	23.68

```sql
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
        join order_details
        on order_details.pizza_id = pizzas.pizza_id
        group by pizza_types.category, pizza_types.name) as a;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

category	name	revenue	rn
Chicken	The Thai Chicken Pizza	43434.25	1
Chicken	The Barbecue Chicken Pizza	42768	2
Chicken	The California Chicken Pizza	41409.5	3
Chicken	The Southwest Chicken Pizza	34705.75	4
Chicken	The Chicken Alfredo Pizza	16900.25	5

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

```
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) as sales;
```

Result Grid | Filter Rows:

order_date	cum_revenue
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

Analyze the cumulative revenue generated over time.

THANK YOU

26 AUG 2024