SALES REPORT

02 May, 2024

Pizza Sales Analysis

PIZZA IS ONE OF THE FAVORITE FOODS OF ALMOST EVERYONE IN THE WORLD.

INTRODUCTION

Hello, my name is Piyush, and I have worked on a pizza sales project where I leveraged SQL to analyze and manipulate data effectively. The project involved various tasks such as calculating total revenue, identifying the most popular pizza sizes and types, determining order distributions by time, and analyzing the percentage contribution of different pizza types to overall sales. Throughout the project, I utilized SQL queries for data extraction, aggregation, and optimization to provide valuable insights into pizza sales performance.

Pizza Sales Analysis

PIZZA IS ONE OF THE FAVORITE FOODS OF ALMOST EVERYONE IN

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

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

Total Orders = 21,350

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
SELECT
    SUM(orders_details.quantity * pizzas.price) AS total_sales
FROM
    orders_details
        JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizza_id;
```

Total Sales = 817860.05

IDENTIFY THE HIGHEST-PRICED PIZZA.

```
SELECT
   SUM(orders_details.quantity * pizzas.price) AS total_sales
FROM
   orders_details
      JOIN
   pizzas ON pizzas.pizza_id = orders_details.pizza_id;
```

Total Greek pizza = 35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
SELECT
   quantity, COUNT(order_details_id)
FROM
   orders_details
GROUP BY quantity;
```

Size & Orders count =

L 18526

M 15385

S 14137

XL 544

XXL 28

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

```
SELECT
   pizza_types.name, SUM(orders_details.quantity) AS quantity
FROM
   pizza_types
      JOIN
   pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
      JOIN
   orders_details ON orders_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 Pizza2371
```

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

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

```
Quantity & Category =
11050   Chicken
11649   Veggie
11987   Supreme
14888  Classic
```

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

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

```
Hours & orders_count =
11  1231
12  2520
13  2455
14  1472
15  1468
16  1920
17  2336
18  2399
19  2009
20  1642
```

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

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

```
Category & Count

Chicken  6
Classic  8
Supreme  9
Veggie  9
```

GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY. I HAVE USED SUBQUERY HERE.

```
SELECT
  ROUND(AVG(quantity), 0)
FROM
  (SELECT
    orders.order_date, SUM(orders_details.quantity) AS quantity
  FROM
    orders
  JOIN orders_details ON orders.order_id = orders_details.order_id
  GROUP BY orders.order_date) AS order_quantity;
```

Average quantity = 138

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

```
SELECT
    pizza_types.name,
    ROUND(SUM(orders_details.quantity * pizzas.price),
            0) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```

```
Name & Revenue =
The Thai Chicken Pizza 43434
The Barbecue Chicken Pizza 42768
The California Chicken Pizza 41410
```

PIZZA TYPE TO TOTAL REVENUE.

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

Catogery & Revenue =
Classic 26.91
Supreme 25.46
Chicken 23.96
Veggie 23.68

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
select order_date, sum(revenue) over(order by order_date) as
cum_revenue
from
(select orders.orders_date,
sum(orders_details.quantity * pizzas.price) as revenue
from orders_details join pizzas
on orders_details.pizza_id = pizzas.pizza_id
join orders
on orders.order_id = orders_details.order_id
group by orders.order_date) as sales ;
```

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

```
select name, revenue from
(select  category, name, revenue,
rank() over(partition by category order by revenue) as rn
from
(select pizza_types.category, pizza_types.name,
sum((orders_details.quantity) * pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join orders_details
on orders_details.pizza_id = pizzas.pizza_id
group by pizza_types.category, pizza_types.name) as a) as b
where rn <= 3 ;
```

```
Name & Revenue
The Chicken Pesto Pizza 16701.75
The Chicken Alfredo Pizza  16900.25
The Southwest Chicken Pizza 34705.75
```

Ingoude Company

THANK YOU

02 May, 2024