SQL PROJECT

PIZZA SALES ANALYSIS PROJECT

- Rajeev Mishra

```sql
-- Q.1 Retrieve the total number of orders placed.

Select
   COUNT(order_id) AS total_orders
FROM
   orders
```

	total_orders
1	21350

```sql
-- Q.2 Calculate the total revenue generated from pizza sales.

Select
  ROUND(SUM(pizzas.price * order_details.quantity), 2) AS total_revenue
FROM
  pizzas
  JOIN order_details ON pizzas.pizza_id = order_details.pizza_id
```

	total_revenue
1	817860.05

```sql
-- Q.3 Identify the highest-priced pizza.

Select
  TOP 1 pizza_types2.name,
  pizzas.price
FROM
  pizza_types2
  JOIN pizzas ON pizza_types2.pizza_type_id = pizzas.pizza_type_id
ORDER BY
  pizzas.price DESC
```

	name	price
1	The Greek Pizza	35.9500007629395

```sql
-- Q.4 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 order_details.pizza_id = pizzas.pizza_id
GROUP BY
  pizzas.size
ORDER BY
  order_count DESC
```

	size	order_count
1	L	18526
2	M	15385
3	S	14137
4	XL	544
5	XXL	28

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

Select
  TOP 5 pizza_types2.name,
  SUM(order_details.quantity) AS quantity_ordered
FROM
  pizza_types2
  JOIN pizzas ON pizza_types2.pizza_type_id = pizzas.pizza_type_id
  JOIN order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY
  pizza_types2.name
ORDER BY
  quantity_ordered DESC
```

	name	quantity_ordered
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418
5	The Thai Chicken Pizza	2371

```sql
-- Q.6 Join the necessary tables to find the total quantity of each pizza category ordered.

Select
  pizza_types2.category,
  SUM(order_details.quantity) AS total_quantity
FROM
  pizza_types2
  JOIN pizzas ON pizza_types2.pizza_type_id = pizzas.pizza_type_id
  JOIN order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY
  pizza_types2.category
ORDER BY
  total_quantity DESC
```

	category	total_quantity
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

```sql
-- Q.7 Determine the distribution of orders by hour of the day.

Select
  DATEPART(hour, order_time) AS order_hours,
  COUNT(order_id) AS num_of_orders
FROM
  orders
GROUP BY
  DATEPART(hour, order_time)
ORDER BY
  num_of_orders DESC
```

	order_hours	num_of_orders
1	12	2520
2	13	2455
3	18	2399
4	17	2336
5	19	2009

```sql
-- Q.8 Join relevant tables to find the category-wise distribution of pizzas.

Select
  category,
  COUNT(name) AS num_of_pizzas
FROM
  pizza_types2
GROUP BY
  category
ORDER BY
  num_of_pizzas DESC
```

	category	num_of_pizzas
1	Supreme	9
2	Veggie	9
3	Classic	8
4	Chicken	6

```sql
-- Q.9 Group the orders by date and calculate the average number of pizzas ordered per day

Select
  Avg(quantity_sum) AS avg_pizza_ordered
FROM
  (Select orders.date,
     SUM(order_details.quantity) AS quantity_sum
    FROM
      orders
      JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY
      orders.date) AS order_quantity
```

	avg_pizza_ordered
1	138

```sql
-- Q.10 Determine the top 5 most ordered pizza types based on revenue.

Select
  TOP 5 pizza_types2.name,
  Round(SUM(order_details.quantity * pizzas.price), 2) AS total_sales
FROM
  order_details
  JOIN pizzas ON order_details.pizza_id = pizzas.pizza_id
  JOIN pizza_types2 ON pizzas.pizza_type_id = pizza_types2.pizza_type_id
GROUP BY
  pizza_types2.name
ORDER BY
  total_sales DESC
```

	name	total_sales
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5
4	The Classic Deluxe Pizza	38180.5
5	The Spicy Italian Pizza	34831.25

```sql
-- Q.11 Calculate the percentage contribution of each pizza type to total revenue.

Select
  pizza_types2.category,
  ROUND(SUM(order_details.quantity * pizzas.price) * 100 / (Select
        SUM(order_details.quantity * pizzas.price)
      From
        order_details
        JOIN pizzas ON order_details.pizza_id = pizzas.pizza_id), 2) AS revenue
FROM
  order_details
  JOIN pizzas ON order_details.pizza_id = pizzas.pizza_id
  JOIN pizza_types2 ON pizza_types2.pizza_type_id = pizzas.pizza_type_id
Group by
  pizza_types2.category
Order by
  revenue Desc
```

	category	revenue
1	Classic	26.91
2	Supreme	25.46
3	Chicken	23.96
4	Veggie	23.68

```sql
-- Q.12 Analyze the cumulative revenue generated over time.

Select date,
  ROUND(SUM(revenue) Over(ORDER BY date),2) As cum_revenue
FROM
  (Select orders.date,
      SUM(order_details.quantity * pizzas.price) AS revenue
    FROM
      orders
      JOIN order_details ON orders.order_id = order_details.order_id
      JOIN pizzas ON order_details.pizza_id = pizzas.pizza_id
    GROUP BY
      orders.date) AS sales
```

	date	cum_revenue
1	2015-01-01	2713.85
2	2015-01-02	5445.75
3	2015-01-03	8108.15
4	2015-01-04	9863.6
5	2015-01-05	11929.55

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

Select category, name, revenue
FROM
  (Select category, name, revenue,
      Rank() Over(Partition BY category
        ORDER BY revenue DESC) AS rank_num
    FROM
      (Select pizza_types2.category, pizza_types2.name,
          ROUND(SUM(order_details.quantity * pizzas.price),2) AS revenue
        FROM
          pizza_types2
          JOIN pizzas ON pizza_types2.pizza_type_id = pizzas.pizza_type_id
          JOIN order_details ON order_details.pizza_id = pizzas.pizza_id
        GROUP BY pizza_types2.name, pizza_types2.category) AS a) AS b
WHERE
  rank_num <= 3
```

	category	name	revenue
1	Chicken	The Thai Chicken Pizza	43434.25
2	Chicken	The Barbecue Chicken Pizza	42768
3	Chicken	The California Chicken Pizza	41409.5
4	Classic	The Classic Deluxe Pizza	38180.5
5	Classic	The Hawaiian Pizza	32273.25

THANK YOU

Feel free to reach out if you'd like to learn more or collaborate on similar projects! 🤝

rm2171291@gmail.com

https://rajeevmishra.dorik.io