Pizza Sales

Presented by Pranali Abhyankar

ABOUT ME

Hi, I’m Pranali, an aspiring Data Analyst with a passion for turning raw data into meaningful insights.

This Pizza Sales SQL project is a showcase of my skills in querying, retrieving, and analyzing data to derive valuable information.

Through this project, I aim to highlight my ability to work with datasets and my enthusiasm for a career in data analytics.

Dataset Description

The dataset contains information about pizza sales. It is designed to analyze sales performance, customer preferences, and operational trends

1. pizzas: Contains coloums pizza_id, pizza_type_id, size, prize
   Columns: 4,
   Rows: 388

2.pizza_types: Contains columns pizza_type_id, name, category, ingridients
   Columns: 4,
   Rows: 132

3.orders: Contains columns order_id, order_date, order_time
   Columns: 3,
   Rows: 64053

4. order_details: Contains columns order_details_id, order_id, pizza_id, quantity
   Columns: 4,
   Rows:194484

Basic Question

Question 1: Retrieve total number of orders

```
1  SELECT count(order_id) AS total_orders FROM orders
2
```

	total_orders bigint
1	21350

Question 2: Calculate the total revenue generated from pizza sales

```sql
SELECT
SUM(order_details.quantity * pizzas.prize)AS revenue
FROM order_details
INNER JOIN pizzas
ON pizzas.pizza_id=order_details.pizza_id
```

	revenue double precision
1	817860.049999993

Question 3: Identify the highest priced pizza

```sql
SELECT pizza_types.pizza_type_id, pizza_types.name, pizzas.prize
FROM pizza_types
INNER JOIN pizzas
ON pizzas.pizza_type_id=pizza_types.pizza_type_id
ORDER BY prize DESC
LIMIT 1
```

	pizza_type_id text	name text	prize double precision
1	the_greek	The Greek Pizza	35.95

Question 4: Identify the most common pizza size orderedpizza

```
SELECT pizzas.size, COUNT(order_details.order_details_id)
FROM pizzas
JOIN order_details
ON pizzas.pizza_id= order_details.pizza_id
GROUP BY pizzas.size
ORDER BY COUNT(order_details.order_details_id) DESC
```

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

Question 5: List the top 5 most ordered pizza types along with their quantities

```sql
SELECT pizza_types.name, SUM(order_details.quantity)
FROM pizza_types
JOIN pizzas
ON pizzas.pizza_type_id=pizza_types.pizza_type_id
JOIN order_details
ON pizzas.pizza_id=order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY SUM(order_details.quantity) DESC
LIMIT 5
```

	name text	sum numeric
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

Intermediate Questions

Question 1: Join necessary tables to find total quantity of each pizza category ordered

```sql
--Question 1: Join necessary tables to find total quantity of each pizza category ordered
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 pizza_types.category
ORDER BY quantity DESC;
```

	category text	quantity numeric
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

Question 2: Determine the distribution of orders by hours of the day

```sql
--Question 2: Determine the distribution of orders by hours of the day
SELECT EXTRACT(HOUR FROM order_time) AS time, COUNT(order_id) AS order_count FROM orders
GROUP BY time
```

	time numeric	order_count bigint
1	11	1231
2	23	28
3	18	2399
4	19	2009
5	15	1468
6	9	1
7	21	1198
8	17	2336
9	20	1642
10	13	2455
11	10	8
12	16	1920
13	22	663
14	12	2520
15	14	1472

Question 3: Join the relevant tables to find the category wise distribution of pizzas.

```sql
--Question 3: Join the relevant tables to find the category wise distribution of pizzas.
SELECT category, COUNT(name) FROM pizza_types
GROUP BY category
```

	category text	count bigint
1	Supreme	9
2	Chicken	6
3	Classic	8
4	Veggie	9

Question 4: Group the orders by date and calculate the average number of pizzas ordered per day

```sql
--Question 4: Group the orders by date and
--calculate the average number of pizzas ordered per day

SELECT ROUND (AVG(quantity)) 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 numeric
1	138

Question 5: Determine the top most ordered pizza based on the revenue

```sql
--Question 5: Determine the top most ordered pizza based on the revenue
SELECT pizza_types.name, SUM (pizzas.prize*order_details.quantity) 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 text	revenue double precision
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5

Advanced Queries

Question 1: Calculate the percentage of each pizza type to total revenue

```sql
--Question 1: Calculate the percentage of each pizza type to total revenue

SELECT pizza_types.category,
(SUM(pizzas.prize*order_details.quantity)/(SELECT
SUM(order_details.quantity * pizzas.prize)
FROM order_details
INNER JOIN pizzas
ON pizzas.pizza_id=order_details.pizza_id)*100) AS revenue
FROM pizza_types
JOIN pizzas
ON pizzas.pizza_type_id=pizza_types.pizza_type_id
JOIN order_details
ON pizzas.pizza_id=order_details.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC
```

	category text	revenue double precision
1	Classic	26.905960255669903
2	Supreme	25.45631126009884
3	Chicken	23.955137556847493
4	Veggie	23.68259092738478 3

Question 2: Analyze the cumulative revenue generated over time.

```sql
--Question 2: Analyze the cumulative revenue generated over time.
SELECT order_date, SUM(revenue) OVER(ORDER BY order_date) AS cumulative_revenue
FROM
(SELECT orders.order_date, SUM(order_details.quantity* pizzas.prize) AS revenue
FROM orders
JOIN order_details
ON order_details.order_id= orders.order_id
JOIN pizzas
ON pizzas.pizza_id= order_details.pizza_id
GROUP BY orders.order_date) AS total_sales
```

	order_date date	cumulative_revenue double precision
1	2015-01-01	2713.8500000000004
2	2015-01-02	5445.75
3	2015-01-03	8108.15
4	2015-01-04	9863.6
5	2015-01-05	11929.55
6	2015-01-06	14358.5
7	2015-01-07	16560.7
8	2015-01-08	19399.05
9	2015-01-09	21526.4
10	2015-01-10	23990.350000000002
11	2015-01-11	25862.65
12	2015-01-12	27781.7
13	2015-01-13	29831.300000000003
14	2015-01-14	32358.700000000004
15	2015-01-15	34343.50000000001
16	2015-01-16	36937.65000000001
17	2015-01-17	39001.75000000001
18	2015-01-18	40978.600000000006
19	2015-01-19	43365.75000000001
20	2015-01-20	45763.65000000001
21	2015-01-21	47804.20000000001

Total rows: 358 | Query complete 00:0

Question 3:Determine the top 3 most ordered pizza types basedon revenue for each pizza category

```sql
--Question 3: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 rn
FROM
(SELECT pizza_types.category, pizza_types.name,
SUM(order_details.quantity*pizzas.prize) AS revenue
FROM pizza_types
JOIN pizzas
ON pizzas.pizza_type_id=pizza_types.pizza_type_id
JOIN order_details
ON pizzas.pizza_id=order_details.pizza_id
GROUP BY pizza_types.category, pizza_types.name) AS a) AS b
WHERE rn<=3
```

	category text	name text	revenue double precision
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
6	Classic	The Pepperoni Pizza	30161.75
7	Supreme	The Spicy Italian Pizza	34831.25
8	Supreme	The Italian Supreme Pizza	33476.75
9	Supreme	The Sicilian Pizza	30940.5
10	Veggie	The Four Cheese Pizza	32265.70000000065
11	Veggie	The Mexicana Pizza	26780.75
12	Veggie	The Five Cheese Pizza	26066.5