SQL Pizza Sales Project

In this project, I utilized SQL queries to answer some questions based on Pizza sales datasets

Datasets Used for this project

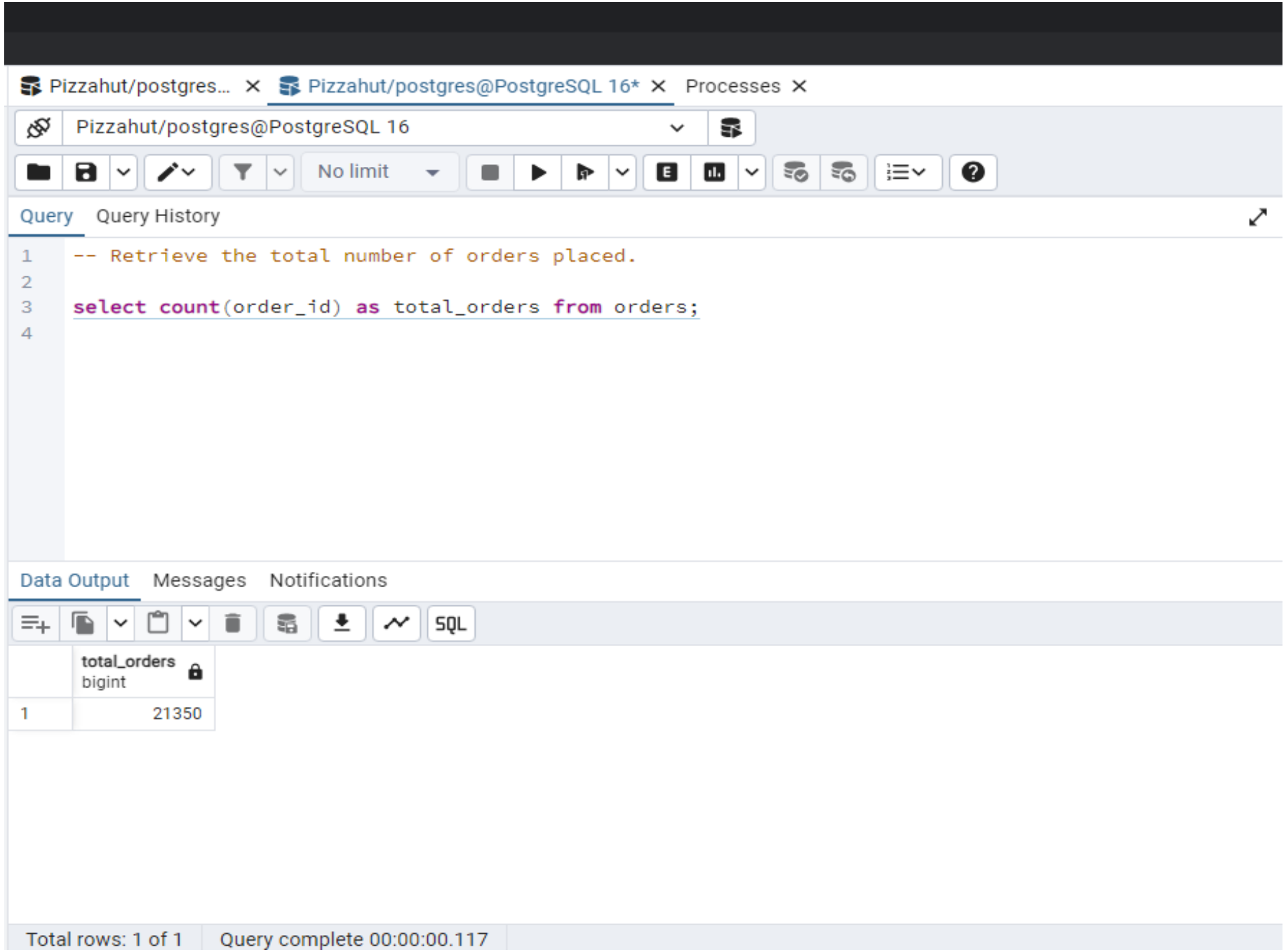
- Order_details
- Orders
- Pizza_types
- Pizzas

Questions Answered

- **Basic:**
- Retrieve the total number of orders placed.
- Calculate the total revenue generated from pizza sales.
- Identify the highest-priced pizza.
- Identify the most common pizza size ordered.
- List the top 5 most ordered pizza types along with their quantities.

- **Intermediate**:
- Join the necessary tables to find the total quantity of each pizza category ordered.
- Determine the distribution of orders by hour of the day.
- Join relevant tables to find the category-wise distribution of pizzas.
- Group the orders by date and calculate the average number of pizzas ordered per day.
- Determine the top 3 most ordered pizza types based on revenue.

- **Advanced**:
- Calculate the percentage contribution of each pizza type to total revenue.
- Analyze the cumulative revenue generated over time.
- Determine the top 3 most ordered pizza types based on revenue for each pizza category.


Pizzahut/postgres...
Pizzahut/postgres@PostgreSQL 16*
Processes
Pizzahut/postgres@PostgreSQL 16
No limit
Query
Query History
1 -- Retrieve the total number of orders placed.
2
3 select count(order_id) as total_orders from orders;
4
Data Output
Messages
Notifications
SQL
total_orders
bigint
1
21350
Total rows: 1 of 1
Query complete 00:00:00.117

Pizzahut/postgres... | Pizzahut/postgres@PostgreSQL 16* | Processes

Pizzahut/postgres@PostgreSQL 16

No limit

Query | Query History

Scratch Pad

```
1  -- Calculate the total revenue generated from pizza sales.
2
3
4
5  select sum(pz.price*o.quantity) as total_revenue
6  from order_details o inner join pizzas pz
7  on o.pizza_id=pz.pizza_id;
8
9
```

Data Output | Messages | Notifications

SQL

	total_revenue double precision
1	817860.049999993

Total rows: 1 of 1 | Query complete 00:00:00.116 | Ln 7, Col 27

Pizzahut/postgres... | Pizzahut/postgres@PostgreSQL 16* | Processes

Pizzahut/postgres@PostgreSQL 16

No limit

Query | Query History

Scratch Pad

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


select pt.pizza_name as highest_priced_pizza, pz.price
from pizza_types pt join pizzas pz
on pt.pizza_type_id=pz.pizza_type_id
order by pz.price desc
limit 1;
```

Data Output | Messages | Notifications

SQL

	highest_priced_pizza text	price double precision
1	The Greek Pizza	35.95

Total rows: 1 of 1 | Query complete 00:00:00.147 | Ln 8, Col 9

Pizzahut/postgres@PostgreSQL 16*

Pizzahut/postgres@PostgreSQL 16

No limit

Query | Query History

Scratch Pad

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



select sum(o.quantity), pz.size as most_common_pizza_size
from order_details o join pizzas pz
on o.pizza_id=pz.pizza_id
group by pz.size
order by sum(o.quantity) desc
limit 1;


```

Data Output | Messages | Notifications

SQL

	sum bigint	most_common_pizza_size text
1	18956	L

Total rows: 1 of 1 | Query complete 00:00:00.138 | Ln 11, Col 1

Pizzahut/postgres@PostgreSQL 16*

Pizzahut/postgres@PostgreSQL 16

No limit

Query | Query History

Scratch Pad

```
1
2  --List the top 5 most ordered pizza types along with their quantities.
3
4  select pt.pizza_name, sum(od.quantity) from pizza_types pt join pizzas pz
5  on pt.pizza_type_id=pz.pizza_type_id
6  join order_details od
7  on od.pizza_id=pz.pizza_id
8  group by pt.pizza_name
9  order by sum(od.quantity) desc
10 limit 5;
11
```

Data Output | Messages | Notifications

SQL

	pizza_name text	sum bigint
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

Total rows: 5 of 5 | Query complete 00:00:00.302 | Ln 10, Col 9

Pizzahut/postgres@PostgreSQL 16*

Pizzahut/postgres@PostgreSQL 16

No limit

Query | Query History

Scratch Pad

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


select pt.category, sum(od.quantity)
from pizza_types pt join pizzas pz
on pt.pizza_type_id=pz.pizza_type_id
join order_details od
on od.pizza_id=pz.pizza_id
group by pt.category;

```

Data Output | Messages | Notifications

SQL

	category text	sum bigint
1	Supreme	11987
2	Chicken	11050
3	Classic	14888
4	Veggie	11649

Total rows: 4 of 4 | Query complete 00:00:00.337 | Ln 10, Col 1

Pizzahut/postgres@PostgreSQL 16*

Pizzahut/postgres@PostgreSQL 16

No limit

Query | Query History

Scratch Pad

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


select extract(hour from order_time) as hour_of_the_day, count(order_id) as number_of_orders
from orders
group by extract(hour from order_time)
order by hour_of_the_day;
```

Data Output | Messages | Notifications

SQL

	hour_of_the_day numeric	number_of_orders bigint
1	9	1
2	10	8
3	11	1231
4	12	2520
5	13	2455
6	14	1472
7	15	1468
8	16	1920
9	17	2336
10	18	2399
11	19	2009

Total rows: 15 of 15 | Query complete 00:00:00.115 | Ln 2, Col 1

Pizzahut/postgres@PostgreSQL 16*

Pizzahut/postgres@PostgreSQL 16

No limit

Query | Query History

Scratch Pad

```sql
-- find the category-wise distribution of pizzas.


select category, count(pizza_name)
from pizza_types
group by category;

```

Data Output | Messages | Notifications

SQL

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

Total rows: 4 of 4 | Query complete 00:00:00.139 | Ln 7, Col 1

Pizzahut/postgres@PostgreSQL 16*

Pizzahut/postgres@PostgreSQL 16

No limit

Query | Query History

Scratch Pad

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

select round(avg(quantity),0) as average_number_of_pizzas_ordered_per_day from
(select o.order_date as date_of_month, sum(od.quantity) as quantity
from order_details od join orders o
on od.order_id=o.order_id
group by o.order_date);
```

Data Output | Messages | Notifications

	average_number_of_pizzas_ordered_per_day numeric
1	138

Total rows: 1 of 1 | Query complete 00:00:00.115 | Ln 7, Col 24

Pizzahut/postgres@PostgreSQL 16*

Pizzahut/postgres@PostgreSQL 16

No limit

Query | Query History

Scratch Pad

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


select pt.pizza_name, sum(od.quantity*pz.price) as revenue from order_details od join pizzas pz
on od.pizza_id=pz.pizza_id
join pizza_types pt
on pt.pizza_type_id=pz.pizza_type_id
group by pt.pizza_name
order by revenue desc
limit 3;
```

Data Output | Messages | Notifications

SQL

	pizza_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

Total rows: 3 of 3 | Query complete 00:00:00.248 | Ln 10, Col 9

Pizzahut/postgres@PostgreSQL 16*

Pizzahut/postgres@PostgreSQL 16

No limit

Query | Query History

Scratch Pad

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

select pt.category,
(sum(od.quantity*pz.price)/(select sum(od.quantity*pz.price) from order_details od join pizzas pz
on od.pizza_id=pz.pizza_id)*100) as revenue_percentage_contribution
from order_details od join pizzas pz
on od.pizza_id=pz.pizza_id
join pizza_types pt
on pt.pizza_type_id=pz.pizza_type_id
group by pt.category
order by revenue_percentage_contribution desc;
```

Data Output | Messages | Notifications

SQL

	category text	revenue_percentage_contribution double precision
1	Classic	26.905960255669903
2	Supreme	25.45631126009884
3	Chicken	23.955137556847493
4	Veggie	23.682590927384783

Total rows: 4 of 4 | Query complete 00:00:00.383 | Ln 12, Col 1

Pizzahut/postgres@PostgreSQL 16*

Pizzahut/postgres@PostgreSQL 16

No limit

Query | Query History

Scratch Pad

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


select order_date, sum(revenue) over(order by order_date) as cum_revenue from
(select o.order_date, sum(od.quantity*pz.price) as revenue
from order_details od join pizzas pz
on od.pizza_id=pz.pizza_id
join orders o
on o.order_id=od.order_id
group by o.order_date);
```

Data Output | Messages | Notifications

SQL

	order_date date	cum_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

Total rows: 358 of 358 | Query complete 00:00:00.306 | Ln 3, Col 1

Pizzahut/postgres@PostgreSQL 16*

Pizzahut/postgres@PostgreSQL 16

No limit

Query | Query History

Scratch Pad

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

select category, pizza_name, revenue from
(select category, pizza_name,
revenue, rank() over(partition by category order by revenue desc) as rn
from
(select pt.category, pt.pizza_name, sum(od.quantity*pz.price) as revenue
from order_details od join pizzas pz
on od.pizza_id=pz.pizza_id
join pizza_types pt
on pt.pizza_type_id=pz.pizza_type_id
group by pt.category, pt.pizza_name))
where rn<=3;

```

Data Output | Messages | Notifications

SQL

	category text	pizza_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

Total rows: 12 of 12 | Query complete 00:00:00.339 | Ln 13, Col 13