SQL PROJECT

Pizza sales Analysis

We would like to offer you a stylish and reasonable presentation that will help you to promote your business

Slide Members

Introduction

Hello, my name is Chaitanya Kohad. In today's competitive food industry, understanding sales patterns and customer preferences is crucial for business success. This project focuses on analyzing pizza sales data to uncover insights that can drive better decision-making and optimize sales strategies. Utilizing SQL, a powerful language for managing and manipulating relational databases, this analysis aims to identify trends, provide actionable insights, and ultimately enhance pizza sales.

Objective

- A comprehensive understanding of sales patterns and customer preferences.
- Identification of top-selling products and key sales periods.
- Recommendations for inventory management to reduce waste and stockouts.
- Marketing strategies to boost sales during off-peak periods

QUESTIONS

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.

BASIC

- **Retrieve the total number of orders placed.**

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

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

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

total_orders
21350

- **Calculate the total revenue generated from pizza sales.**

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

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

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

total_revenue
827450

Result 1 | Read Only

- **Identify the highest-priced pizza.**

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


SELECT
    pizza_types.name AS name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;

```

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

name	price
The Greek Pizza	36

- **Identify the most common pizza size ordered.**

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

select pizzas.size , count(order_details.order_id) as order_count
from pizzas join order_details
on pizzas.pizza_id = order_details.pizza_id
group by pizzas.size
order by order_count desc
limit 1;
```

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

size	order_count
L	18526

Result 1 | Read Only

- **List the top 5 most ordered pizza types along with their quantities.**

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

SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
FROM
    pizzas
        JOIN
    pizza_types 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 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

Result Grid | Form Editor | Field Types

Result 1 | Read Only

Intermediate

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

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

SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS total_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 category
ORDER BY total_quantity DESC;
```

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

category	total_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Result 1 | Read Only

- **Determine the distribution of orders by hour of the day.**

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

SELECT
    HOUR(orders.time) AS hours, COUNT(order_id) AS orders
FROM
    orders
GROUP BY hours;
```

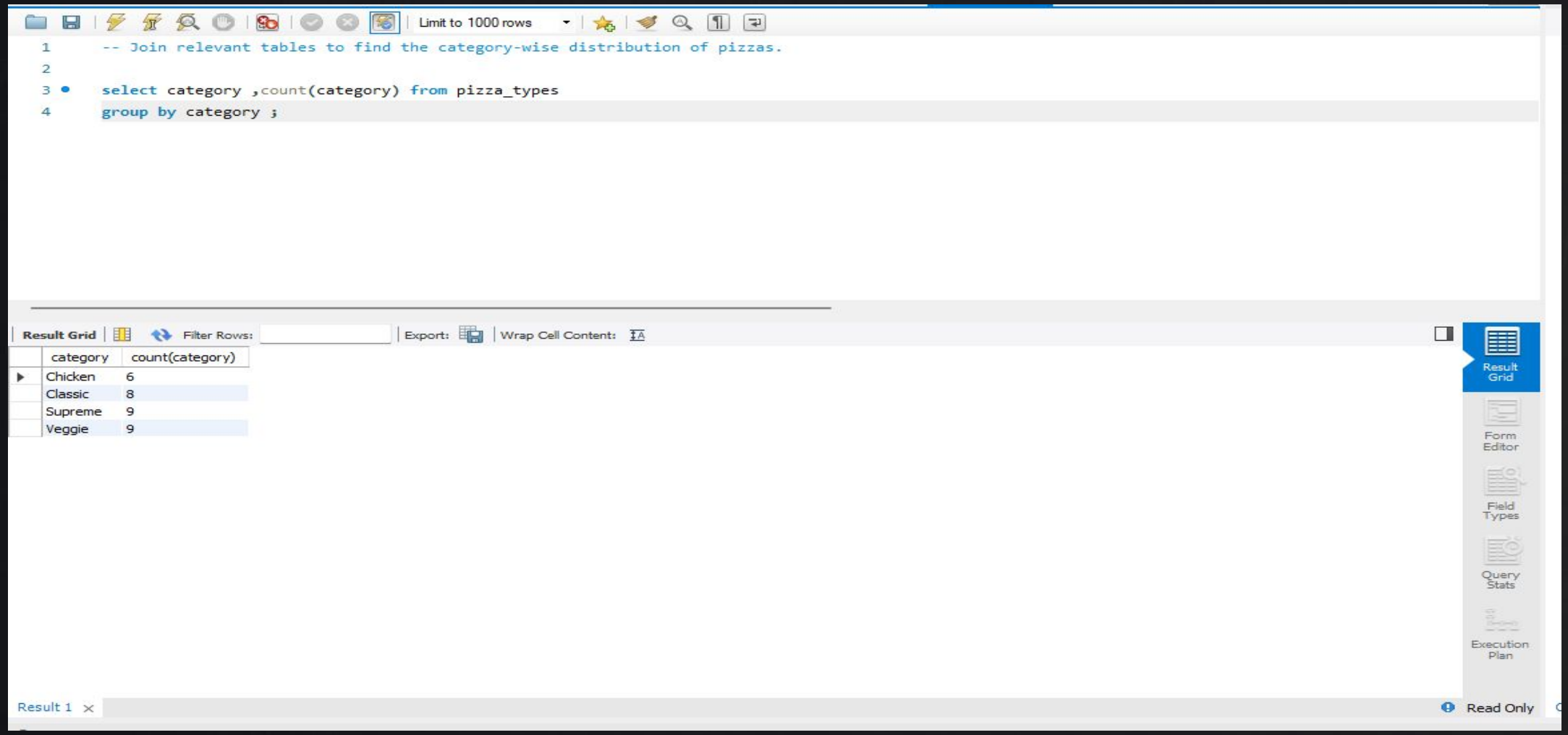
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

hours	orders
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

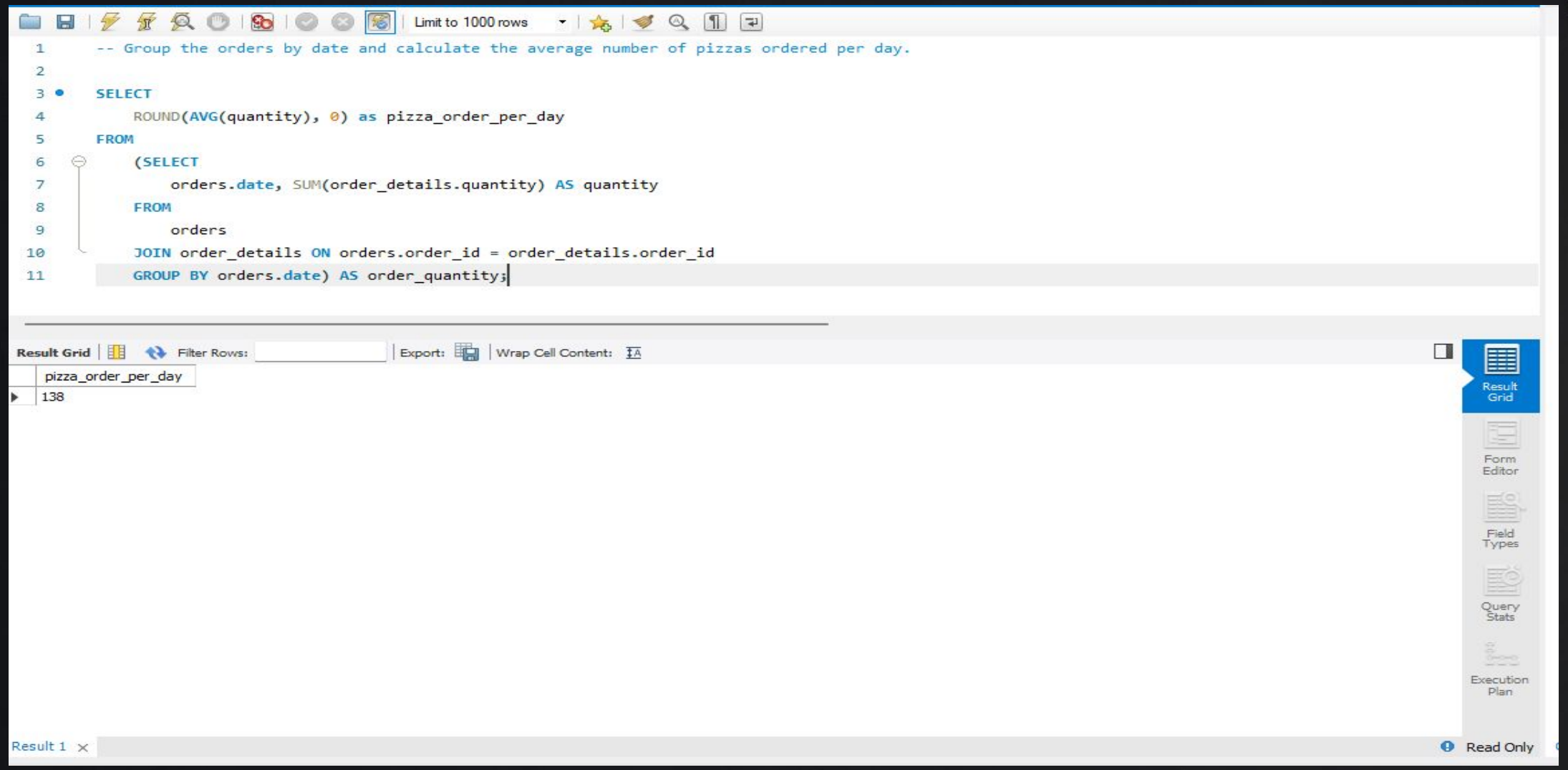
Result Grid | Form Editor | Field Types | Query Stats | Execution Plan

Result 1 | Read Only

- **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.**

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

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 Content: | Fetch rows:

name	revenue
The Thai Chicken Pizza	44027
The Barbecue Chicken Pizza	43376
The California Chicken Pizza	42002

Result 1 | Read Only

Advanced

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

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

SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price) / 827450 * 100,
            2) AS percentage_of_total_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.category;
```

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

category	percentage_of_total_revenue
Classic	26.96
Veggie	23.53
Supreme	25.51
Chicken	24.01

Result 1 | Read Only

- **Analyze the cumulative revenue generated over time.**

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

select date ,
    sum(revenue) over (order by date) as Cummulative_revenue
from
(select orders.date , sum(pizzas.price * order_details.quantity) as revenue
from pizzas join order_details
on pizzas.pizza_id = order_details.pizza_id
join orders
on orders.order_id = order_details.order_id
group by orders.date)as sales  ;
```

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

date	Cummulative_revenue
2015-01-01	2746
2015-01-02	5512
2015-01-03	8203
2015-01-04	9983
2015-01-05	12075
2015-01-06	14532
2015-01-07	16761
2015-01-08	19628
2015-01-09	21777
2015-01-10	24270
2015-01-11	26161
2015-01-12	28105
2015-01-13	30182
2015-01-14	32742
2015-01-15	34752
2015-01-16	37371
2015-01-17	39459
2015-01-18	41458
2015-01-19	43872
2015-01-20	46301
2015-01-21	48363

Result Grid | Form Editor | Field Types | Query Stats | Execution Plan

Result 1 | Read Only

THANK YOU