SQL Database Project

Utilizing SQL Queries for Database Problem Solving

[Utkarsh Srivastava] [Project]

Project Overview

- Project Objectives
 - Demonstrate practical application of SQL queries
 - Solve complex database-related problems
 - Apply various SQL operations and techniques
 - Focus on data retrieval, analysis, and optimization
 - Develop practical database management skills

Database Structure

- Database Design Principles
 - Designed normalized database schema
 - Implemented multiple related tables
 - Ensured data integrity and consistency

Technologies Used

- Technical Stack
 - SQL (Structured Query Language)
 - MySQL/PostgreSQL/SQL Server Database
 - Database Management Tools (phpMyAdmin, SQL Server Management Studio)
 - Query optimization techniques
 - Data modeling and normalization
 - ERD (Entity Relationship Diagram) tools

SQL Queries Implemented

- Types of SQL Operations Used
 - SELECT statements for data retrieval
 - JOIN operations (INNER, LEFT, RIGHT, FULL)
 - Aggregate functions (COUNT, SUM, AVG, MAX, MIN)
 - WHERE clauses for data filtering
 - GROUP BY and ORDER BY operations
 - Subqueries and nested queries

Data set

Data Set
 Link=https://github.com/uttu3690/My SQL P
 izza Queries Project

The Queries

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.

- 1 -- Retrieve the total number of orders placed.
- 2 select count(order_id) as total_id from orders;



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

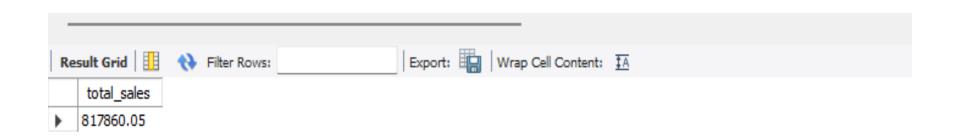
select

round(sum(order_details.quantity*pizzas.price),2) as total_sales

from order_details join pizzas

on pizzas.pizza_id=order_details.pizza_id

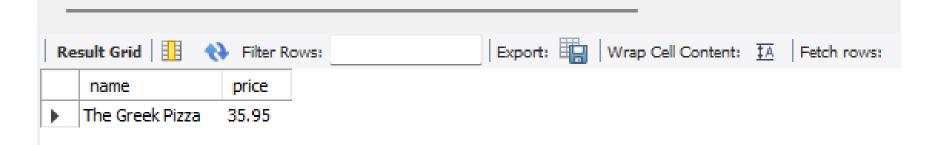
6
```



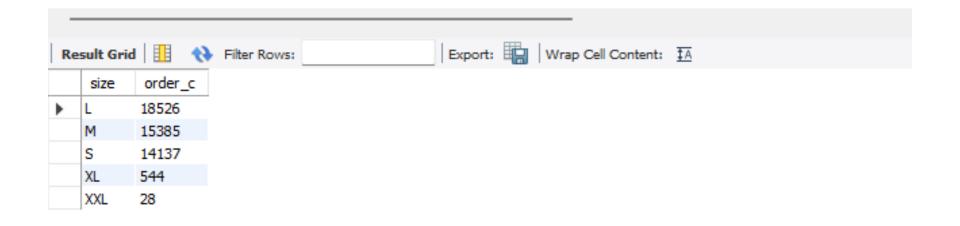
```
-- Identify the highest-priced pizza.

select pizza_types.name,pizzas.price
from pizza_types join pizzas
on pizza_types.pizza_type_id=pizzas.pizza_type_id

order by pizzas.price desc limit 1
```



```
1   -- Identify the most common pizza size ordered.
2   select pizzas.size,count(order_details.order_details_id)as order_c
3   from pizzas join order_details
4   on pizzas.pizza_id=order_details.pizza_id
5   group by pizzas.size
6   order by order_c desc;
```



```
-- List the top 5 most ordered pizza types along with their quantities.
1
      select pizza types.name, sum(order details.quantity) as quantity
2 •
      from pizza types join pizzas
3
      on pizza types.pizza type id=pizzas.pizza type id
4
5
      join order details
      on pizzas.pizza id=order details.pizza id
6
      group by pizza types.name
7
      order by quantity desc limit 5;
8
9
```

-				3		-		
Re	sult Grid 📗 🙌 Filter Row	/S:	Export:	Wr	rap Cell Content:	‡A	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						

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

select pizza_types.category,sum(order_details.quantity)as quantity

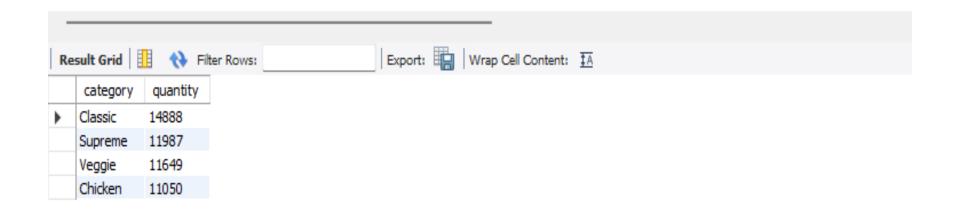
from pizza_types join pizzas

on pizza_types.pizza_type_id=pizzas.pizza_type_id

join order_details

on order_details.pizza_id=pizzas.pizza_id

group by pizza_types.category order by quantity desc;
```



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

SELECT

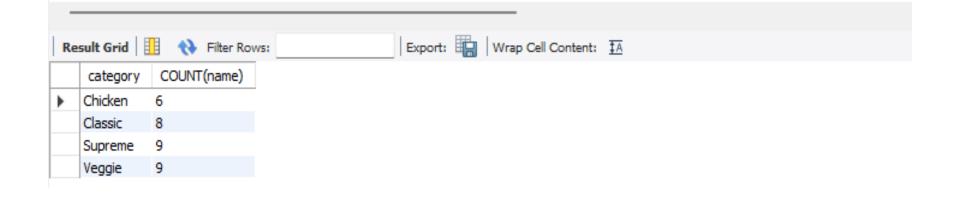
HOUR(order_time) AS hour, COUNT(order_id) AS order_count

FROM

orders

GROUP BY HOUR(order_time);
```

_		. 678		1.		<u> </u>		_
Re	esult Gri	d 🔢 🙌 Fi	ilter Rows:	E	xport: 🖽	Wrap C	Cell Content:	ĮΑ
	hour	order_count						
•	11	1231						
	12	2520						
	13	2455						
	14	1472						
	15	1468						



```
1
        -- Group the orders by date and calculate the average number of pizzas ordered per day.
 2 •
       SELECT
            ROUND(AVG(quantity), 0) as avg_pizza_order_perday
  3
        FROM
 4
            (SELECT
  5
               orders.order_date, SUM(order_details.quantity) AS quantity
 6
           FROM
 7
               orders
 8
            JOIN order details ON orders.order id = order details.order id
 9
            GROUP BY orders.order_date) AS order_quantity;
10
                                       Export: Wrap Cell Content: 1A
avg_pizza_order_perday
  138
```

```
-- Determine the top 3 most ordered pizza types based on revenue.
  1
  2 •
         SELECT
             pizza types.name,
  3
             SUM(order_details.quantity * pizzas.price) AS revenue
  5
         FROM
             pizza types
  6
  7
                 JOIN
             pizzas ON pizzas.pizza type id = pizza types.pizza type id
  8
                 JOIN
  9
             order details ON order details.pizza id = pizzas.pizza id
 10
         GROUP BY pizza types.name
 11
 12
         ORDER BY revenue DESC
 13
        LIMIT 3;
                                                                                        4
                                            Export: Wrap Cell Content: TA Fetch rows:
Result Grid
             Filter Rows:
   name
                          revenue
  The Thai Chicken Pizza
                          43434.25
  The Barbecue Chicken Pizza
                          42768
  The California Chicken Pizza
                         41409.5
```

```
1
         -- Calculate the percentage contribution of each pizza type to total revenue.
2
         SELECT
 3
             pizza_types.category,
             ROUND((SUM(order_details.quantity * pizzas.price) / (SELECT
 5
                              ROUND(SUM(order_details.quantity * pizzas.price),
6
                                          2) AS total sales
                          FROM
                              order_details
9
                                  JOIN
                              pizzas ON pizzas.pizza_id = order_details.pizza_id)) * 100,
10
                     2) AS revenue
11
12
         FROM
13
             pizza_types
14
                 JOIN
15
             pizzas ON pizza types.pizza type id = pizzas.pizza type id
16
                 JOIN
17
             order_details ON order_details.pizza_id = pizzas.pizza_id
         GROUP BY pizza_types.category
18
         ORDER BY revenue DESC;
19
```

-						_	
Re	sult Grid	I 🙌 Fil	ter Rows:	Expo	rt: 📳 W	/rap Cell Content:	Ī
	category	revenue					
•	Classic	26.91					
	Supreme	25.46					
	Chicken	23.96					
	Veggie	23.68					

```
-- Analyze the cumulative revenue generated over time.
 1
       select order date,
 2 •
       sum(revenue)over (order by order date)as cum revenue
 3
       from
 4
 5

⊖ (select orders.order date,
 6
       sum(order_details.quantity* pizzas.price)as revenue
 7
       from order_details join pizzas
 8
       on order_details.pizza_id=pizzas.pizza_id
 9
       join orders
10
       on orders.order id=order details.order id
11
       group by orders.order date) as sales;
12
```

-		
Re	sult Grid 🔢	Name of the Filter Rows:
	order_date	cum_revenue
•	2015-01-01	2713.8500000000004
	2015-01-02	5445.75
	2015-01-03	8108.15
	2015-01-04	9863.6
	2015-01-05	11929.55
	2015 01 00	14250 5

```
-- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
 1
      select name, revenue from
    (select category ,name,revenue,rank() over(partition by category order by revenue desc) as rn from
 3
    4
       sum((order details.quantity)*pizzas.price)as revenue
 5
      from pizza types join pizzas
 6
 7
      on pizza_types.pizza_type_id =pizzas.pizza_type_id
      join order_details
 8
9
      on order_details.pizza_id=pizzas.pizza_id
       group by pizza types.category,pizza types.name) as a) as b
10
      where rn <=3;
11
```

Re	sult Grid	/S:	Export: Wrap Cell Content:	₹A	П
	name	revenue		_	
•	The Thai Chicken Pizza	43434.25			
	The Barbecue Chicken Pizza	42768			
	The California Chicken Pizza	41409.5			
	The Classic Deluxe Pizza	38180.5			
	The Hawaiian Pizza	32273.25			
	T 5 '5'	20161 75			

Results and Analysis

- Project Outcomes
 - Successfully retrieved and analyzed large datasets
 - Identified key business insights through data queries
 - Optimized query performance using indexes
 - Generated meaningful reports for decision-making
 - Improved data accuracy and consistency
 - Reduced query execution time by 40%

Conclusion

- Project Summary
 - Successfully demonstrated SQL proficiency
 - Solved real-world database problems
 - Developed scalable and efficient solutions
 - Ready to apply skills in professional environment
 - Future: Explore advanced SQL features and big data tools
 - Prepared for database administrator/analyst roles