

↓ Pizza Sales report



A large white downward arrow is positioned above the title. To the right of the title is a cartoon illustration of two people at a table. On the left, a person wearing an apron and glasses is looking thoughtful, resting their chin on their hand. On the right, another person is gesturing with their hands while holding a book titled "FINANCE LITERACY". The table has a MySQL logo and two stacks of coins (one pink, one blue) on it. In the background, there are three small boxes containing icons related to finance: a shield with a dollar sign, a bar chart with an upward arrow and a dollar sign, and a plant growing from coins.

Using MySQL



By - Mohsin khan

2024

Introduction

In this project, I have taken a dataset of Pizza Store, using some of the important concept of SQL Queries such as JOINTS, AGG. FUNCTIONS etc. Several Practise Problems of the Business being implemented.
Further Slides will shows the Challenges along with their Solution(Queries) & their Output



Total numbers of Orders placed!



```
4  
5      -- Retrieve the total number of orders placed.  
6  
7  
8  
9 •   select count(order_id) from orders;  
10  
11
```

Result Grid | Filter Row

	count(order_id)
▶	12779

Highest-Priced Pizza!



```
3 -- Identify the highest-priced pizza.  
4  
5  
6 • SELECT  
7     size, price  
8 FROM  
9     pizza  
10 ORDER BY price DESC  
11 LIMIT 3;
```

Result Grid | Filter Rows:

	size	price
▶	XXL	35.95
	XL	25.5
▶	S	23.65

Total revenue generated from Sale



```
2   -- Calculate the total revenue generated from pizza sales.  
3  
4 • SELECT  
5     SUM(price * quantity) AS TOTAL_AMOUNT  
6   FROM  
7     order_details o  
8   JOIN  
9     pizza p ON o.pizza_id = p.pizza_id
```

A screenshot of a database query results interface. At the top, there are tabs for 'Result Grid' (which is selected), 'SQL', 'File', and 'Help'. Below the tabs, there is a table with one row and two columns. The first column is empty, and the second column is labeled 'TOTAL_AMOUNT' with the value '16878.65'.

	TOTAL_AMOUNT
▶	16878.65

Top-5 Most ordered Pizza!



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

SELECT
    name, SUM(quantity) AS Total_Quantity
FROM
    order_details o
    JOIN
    pizza p ON o.pizza_id = p.pizza_id
    JOIN
    pizza_types t ON p.pizza_type_id = t.pizza_type_id
GROUP BY name
ORDER BY Total_Quantity DESC
LIMIT 5;
```

Result Grid | Filter Rows:

	name	Total_Quantity
▶	The Pepperoni Pizza	58
	The Thai Chicken Pizza	55
	The Classic Deluxe Pizza	53
	The Barbecue Chicken Pizza	51
	The Hawaiian Pizza	50

Most common Pizza size ordered!



```
-- Identify the most common pizza size ordered.  
• SELECT  
    size, SUM(quantity) AS TotalQuantity  
FROM  
    pizza p  
    JOIN  
    order_details o ON p.pizza_id = o.pizza_id  
GROUP BY size  
ORDER BY TotalQuantity DESC;
```

Result Grid | Filter

	size	TotalQuantity
▶	L	403
	M	324
	S	284
	XL	10

Category-wise order placed!



```
1  -- Join the necessary tables to find the
2  -- total quantity of each pizza category ordered.
3
4  SELECT
5      category, SUM(quantity) AS Quantity
6  FROM
7      pizza p
8      JOIN
9      order_details o ON p.pizza_id = o.pizza_id
10     JOIN
11     pizza_types t ON p.pizza_type_id = t.pizza_type_id
12  GROUP BY category;
```

Result Grid | Filter

	category	Quantity
▶	Classic	314
▶	Veggie	235
▶	Supreme	241
▶	Chicken	231

Per-Hour Distribution of Sale!



```
1 -- Determine the distribution of orders by hour of the day.  
2 SELECT  
3     HOUR(time) AS Hours, COUNT(order_id) AS orders  
4 FROM  
5     orders  
6 GROUP BY Hours;
```

	Hours	orders
▶	11	721
	12	1500
	13	1448
	14	927
	15	891
	16	1149
	17	1437
	18	1437
	19	1187
	20	976
	21	700
	22	383
	23	17
	10	6

Category-wise Distribution of Sale!



```
-- Join relevant tables to find the category-wise distribution of pizzas.  
select category,count(name) from  
pizza p join pizza_types t  
on p.pizza_type_id=t.pizza_type_id  
group by t.category;
```

Result Grid | Filter Row

	category	count(name)
▶	Chicken	18
	Classic	26
	Supreme	25
	Veggie	27

Per-day Distribution of sale



```
1 -- Group the orders by date and calculate the average  
2 -- number of pizzas ordered per day.  
3  
4  
5 • select date,sum(quantity) as TotalSale from  
6 orders o join order_details od  
7 on o.order_id=od.order_id  
8 group by date;
```

Result Grid | Filter Row

date	TotalSale
2015-01-01	162
2015-01-02	165
2015-01-03	158
2015-01-04	106
2015-01-05	125
2015-01-06	147
2015-01-07	138
2015-01-08	20

Contact



+91-9169450018



mohsinjaisi78692@gmail.com





Thank you!

