SQL PROJECT

Pizza shop sales

Objectives:

- Analysis of sales trends and customer preferences.
- Generation of insightful reports for decision making.
- Efficient data storage and retrieval of the data.

Overview:

This project focuses on managing and analyzing the sales data of a pizza shop using MySQL.

Our database design ensures efficient data storage, quick retrieval, and robust data integrity.

Database Design

Tables:

Orders: Contains order date and time.

Orders_details: Records order details.

Pizzas: Contains pizza menu details.

Pizza_types:Information about pizza type and category

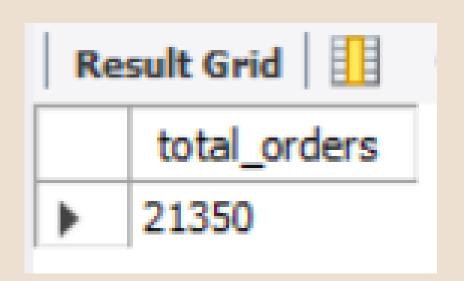
Retrieve the Total number of orders placed.

```
SELECT

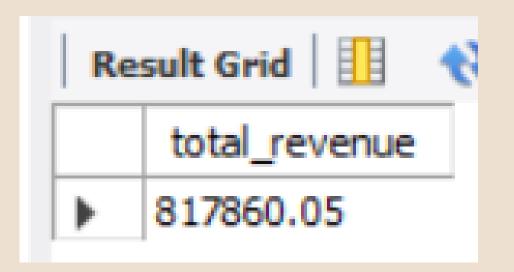
COUNT(order_id) AS total_orders

FROM

orders;
```



Calculate the Total revenue generated from pizza sales.



Identify the highest priced pizza

name		price
•	The Greek Pizza	35.95

List the top 5 ordered pizza types along with their quantities

```
select sum(orders_details.quantity) as quantity,pizza_types.name
from pizzas
join pizza_types on
pizzas.pizza_type_id=pizza_types.pizza_type_id
join orders_details on
orders_details.pizza_id=pizzas.pizza_id
group by pizza_types.name
order by quantity desc
limit 5;
```

	quantity	name
•	2453	The Classic Deluxe Pizza
	2432	The Barbecue Chicken Pizza
	2422	The Hawaiian Pizza
	2418	The Pepperoni Pizza
	2371	The Thai Chicken Pizza

Identify the most common pizza size ordered

```
select pizzas.size,count(orders_details.order_details_id) as size_count
from pizzas
join orders_details on
pizzas.pizza_id = orders_details.pizza_id
group by pizzas.size
order by size_count desc;
```

	size	size_count
•	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28

Join the necessary table to find the total quantity of each pizza ordered.

```
select sum(orders_details.quantity) as total_quantity, pizza_types.category as category
from pizza_types
join pizzas on
pizza_types.pizza_type_id = pizzas.pizza_type_id
join orders_details on
orders_details.pizza_id = pizzas.pizza_id
group by category
order by total_quantity desc;
```

	total_quantity	category
•	14888	Classic
	11987	Supreme
	11649	Veggie
	11050	Chicken

Determine the distribution of orders by hour of the day.

```
SELECT

HOUR(order_time) AS hours, COUNT(order_id)

FROM

orders

GROUP BY hours;
```

	hours	COUNT(order_id)
•	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336

Find the category wise distribution of pizzas.

```
SELECT

category, COUNT(name)

FROM

pizza_types

GROUP BY category;
```

	category	COUNT(name)
•	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

Group the orders by date and calculate the average number of pizza ordered per day.

```
select round(avg(sum_quantity),0) as avg_pizz_ordered_per_day from
  (select orders.order_date as date,sum(orders_details.quantity) as sum_quantity
  from orders
  join orders_details on
  orders.order_id=orders_details.order_id
  group by date) as order_quantity;
```

```
avg_pizz_ordered_per_day

138
```

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

```
select pizza_types.name as names, sum(pizzas.price * orders_details.quantity) as revenue
from pizza_types
join pizzas on
pizza_types.pizza_type_id=pizzas.pizza_type_id
join orders_details on
pizzas.pizza_id=orders_details.pizza_id
group by names
order by revenue desc limit 3;
```

The Thai Chicken Pizza 43434.25 The Barbecue Chicken Pizza 42768 The California Chicken Pizza 41400.5		names	revenue
	•	The Thai Chicken Pizza	43434.25
The California Chicken Dizza 41400 5		The Barbecue Chicken Pizza	42768
THE California Chicker Fizza 41403.3		The California Chicken Pizza	41409.5

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

```
select pizza_types.category as category ,
    round(sum(pizzas.price * orders_details.quantity)/(select
        round(sum(pizzas.price * orders_details.quantity),
            2) as total_revenue
from pizzas
    join orders_details on
        pizzas.pizza_id=orders_details.pizza_id)* 100,2) as revenue
from pizza_types
    join pizzas on
        pizza_types.pizza_type_id=pizzas.pizza_type_id
    join orders_details on
        pizzas.pizza_id=orders_details.pizza_id
 group by category
 order by revenue desc;
```

	category	revenue
•	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

Analyze the cumulative revenue generated over time.

```
select date ,
sum(revenue) over(order by date) as cum_revenue
from
    (select orders.order_date as date, round(sum( orders_details.quantity * pizzas.price),2) as revenue
    from orders_details
    join pizzas on
    pizzas.pizza_id=orders_details.pizza_id
    join orders on
    orders.order_id=orders_details.order_id
    group by date)
as sales;
```

	date	cum_revenue
•	2015-01-01	2713.85
	2015-01-02	5445.75
	2015-01-03	8108.15
	2015-01-04	9863.6
	2015-01-05	11929.55
	2015-01-06	14358.5

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

	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
	The Pepperoni Pizza	30161.75

THANK YOU

Presented By: Vimal Varma