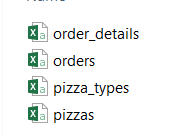
**SQL project**

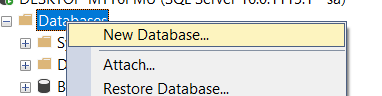
Import the data set into the SQL workbench

The data set name is Pizza\_sale

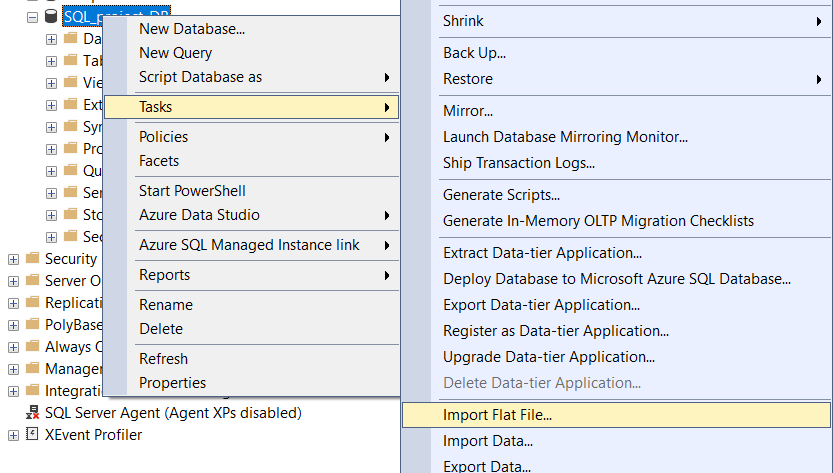
And it contains 4 table as name below



Now open the SQL work bench and create one data base using the step

And I have given name SQL\_project\_DB

Now import this tables in the sql in the following steps

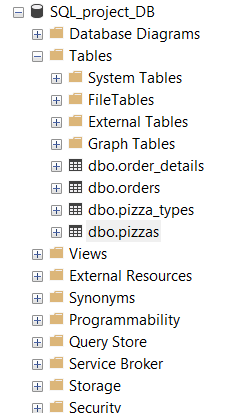


Click on Next and give the path of the table and give the table name

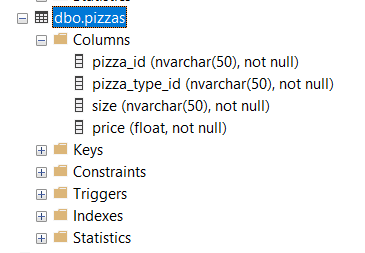
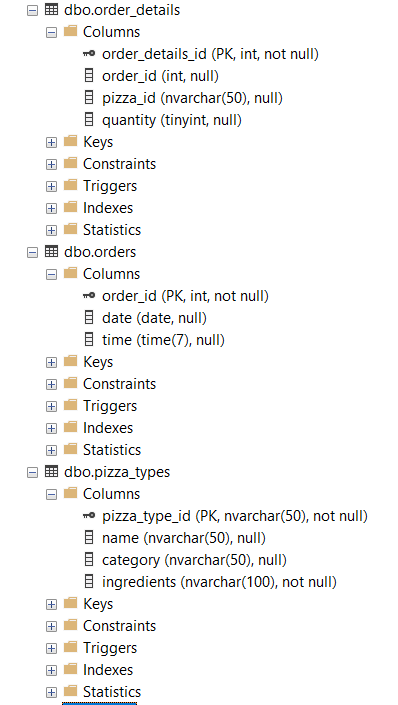
After loading the data you can verify the column and can edit the column’s and can edit the date data type of the column.

And next Finish the data is loaded in the SQL workbench

The all tables are imported successfully as show below



Now let’s understand the data which is present in the table, Expand the table and observer the table columns and which are connected to which table.



Here the data is connected through the primary key and foreign key of the table

Primary key is main key of the table which is same repeated in the other table for inter connection of the data.

Primary key of order\_details is order\_details\_id

Primary key of order is order\_id

Primary key of pizza\_types is pizza\_type\_id

And Pizzas having foreign key which are pizza\_id, pizza\_type\_id—which connect the two table and it will be more use full for us in solving the queries.

First Query we are using in this project will be

Use SQL\_project\_DB;

This query will help us to connect the database which we are going to work/ do project.

To know or to see the data of the each and every table we can use the following query to view the data.

select \* from order\_details;

select \* from orders;

select \* from pizza\_types;

select \* from pizzas;

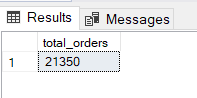
And the following questions will be solving the data retrieving and viewing the data

1) Retrieve the total number of orders placed.

Showing the all the total count of orders

Select count (order\_id) as total\_orders from orders;

Execute will get the results as below



2) Calculate the total revenue generated from pizza sales.

--Sum of all categories in the order\_details and making the sum with round of 2 decimals.

Select ROUND (sum (order\_details.quantity \* pizzas.price), 2)

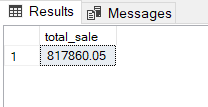
As total\_sale

From order\_details

join

pizzas on pizzas.pizza\_id = order\_details.pizza\_id;

Execute will get the results as below



3) Identify the highest-priced pizza.

--Identifying the top pizza and its price

Select top 1 pizza\_types.name, pizzas.price

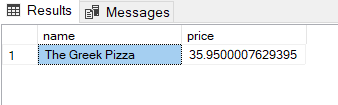
from pizza\_types

join pizzas

on pizza\_types.pizza\_type\_id = pizzas.pizza\_type\_id

Order by pizzas.price desc;

Execute will get the results as below



4) Identify the most common pizza size ordered.

--Which size pizza ordered more identifying

Select pizzas.size,

COUNT (order\_details.order\_details\_id) as order\_count

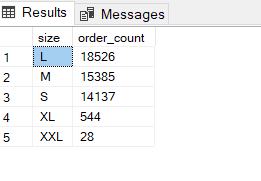
from

pizzas join order\_details

on pizzas.pizza\_id = order\_details.pizza\_id

group by pizzas.size order by order\_count desc;

Execute will get the results as below



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

--Top 5 pizza and quantity

Select top 5 pizza\_types.name, SUM (order\_details.quantity) as quantity

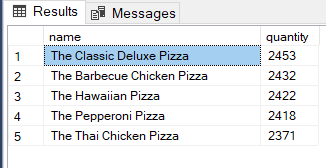
from pizza\_types

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.name order by quantity desc;

Execute will get the results as below



6) Join the necessary tables to find the total quantity of each pizza category ordered.

Select pizza\_types.Category, SUM (order\_details.quantity) as quantity

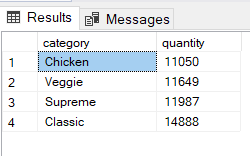
from pizza\_types

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;

Execute will get the results as below



7) Determine the distribution of orders by hour of the day.

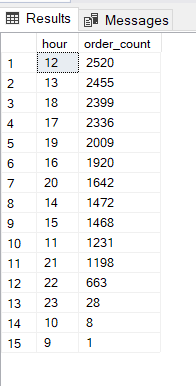
Select DATEPART (HOUR, time) as hour,

Count (order\_id) as order\_count

from orders

Group by DATEPART (HOUR, time) order by order\_count desc;

Execute will get the results as below

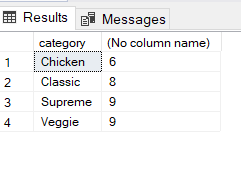


8) Join relevant tables to find the category-wise distribution of pizzas.

Select category, count (name) from pizza\_types

Group by category;

Execute will get the results as below



9) Group the orders by date and calculate the average number of pizzas ordered per day.

Select AVG (quantity) as pizza\_order\_per\_day

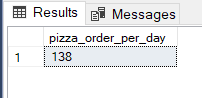
From (select orders. Date, SUM (order\_details.quantity) as quantity

From orders join order\_details

On order\_details. Order\_id = orders. Order\_id

Group by orders.Date) as order quantity;

Execute will get the results as below



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

Select top 3 pizza\_types.name, sum (order\_details.quantity \* pizzas.price) as revenue

From pizza\_types

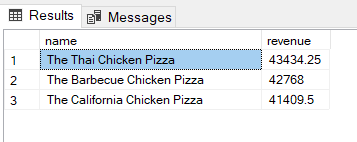
join pizzas on

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;

Execute will get the results as below



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

-- (price of one pizza category / sum of all pizza categories) \* 100

Select pizza\_types. Category,

Round (SUM (order\_details.quantity \* pizzas.price)/ (select ROUND (sum (order\_details.quantity \* pizzas.price), 0) as total\_sale

From order\_details join pizzas on pizzas.pizza\_id = order\_details.pizza\_id)\*100, 2) as revenue

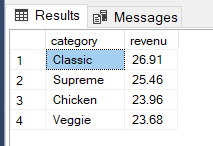
From pizza\_types

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. Category order by revenue desc;

Execute will get the results as below



12) Analyze the cumulative revenue generated over time.

Select date, round (sum (revenue) over (order by date), 2) as cumrevenue

From

(Select orders. date,

SUM (order\_details.quantity \* pizzas.price) as revenue

From order\_details

join pizzas

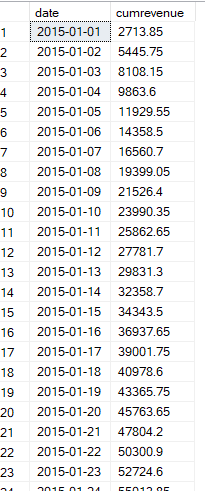
on order\_details.pizza\_id = pizzas.pizza\_id

join orders

On orders. Order\_id = order\_details. Order\_id

Group by orders.date) as sales;

Execute will get the results as below



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

Select name, revenue from

(Select category, name, revenue, rank () over (partition by category order by revenue desc) as rn

From

(Select pizza\_types. Category, 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. Category, pizza\_types.name) as a) as b

Where rn <=3;

Execute will get the results as below

