**8218-Arsh nadeem-Introduction to Databases-Jan 2025**

1. Problem Statement: There can be multiple customers, who can place multiple orders on the site. Now a sales person can handle these orders and will distribute them to multiple sales persons (One order will be assigned to one salesperson only). So a sales person can have multiple orders of multiple customers

2.Design Schema

-> For designing schema we should understand the data provided and how many tables may be created or required

-> there can be multiple customers who can have multiple orders

-> there can be multiple sales person who can have multiple orders of multiple customers

-> we need to find the relationship between : Customers and orders and Sales Person and Orders.

-> According to the Question:

Customer ---> Orders

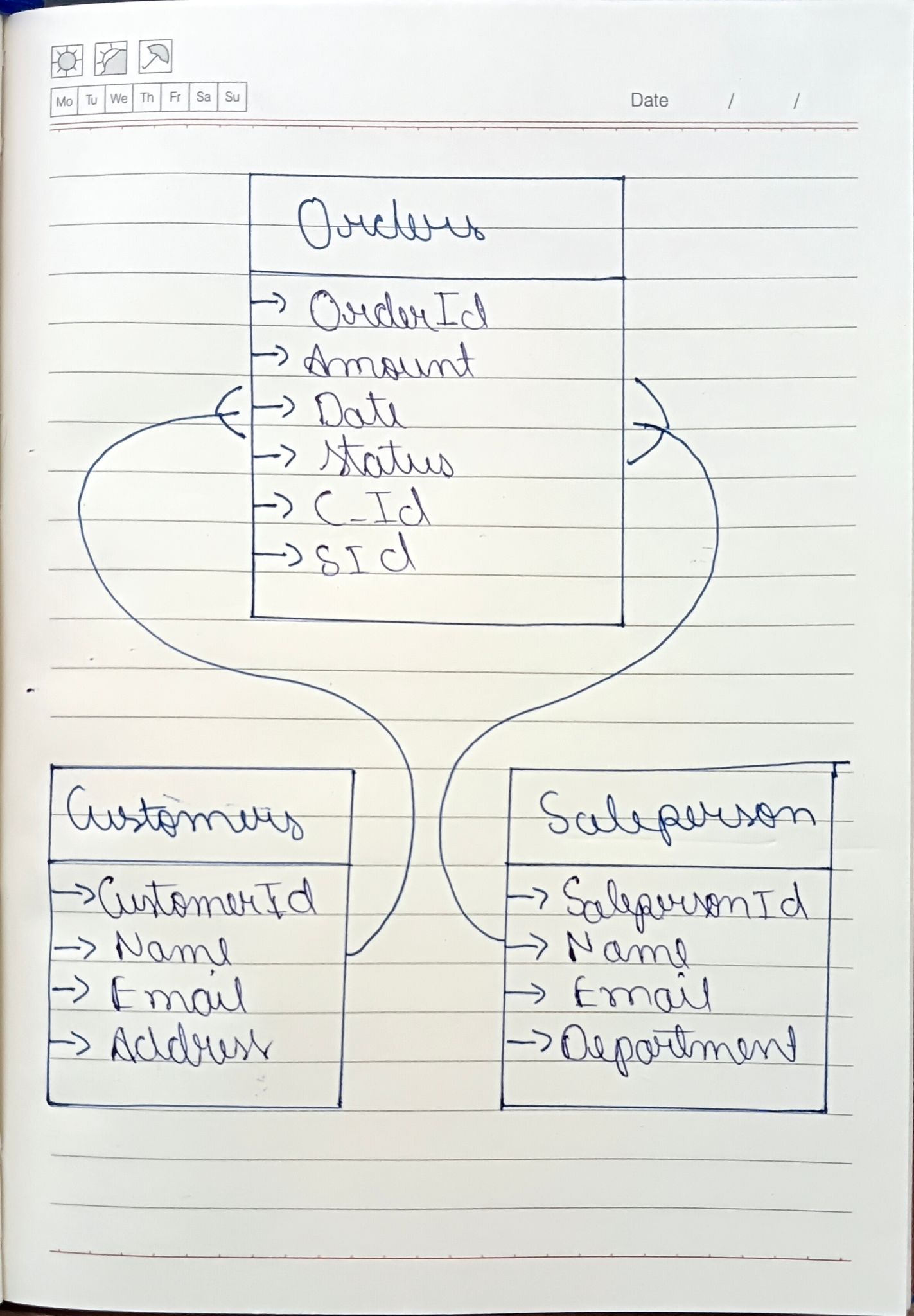
Customer can have multiple orders, but an Order can’t have multiple Customers

This shows 1 to Many relationship between Customer and Order.

SalesPerson ---> Orders

SalePerson can have multiple orders, but an Order can’t have multiple SalePerson

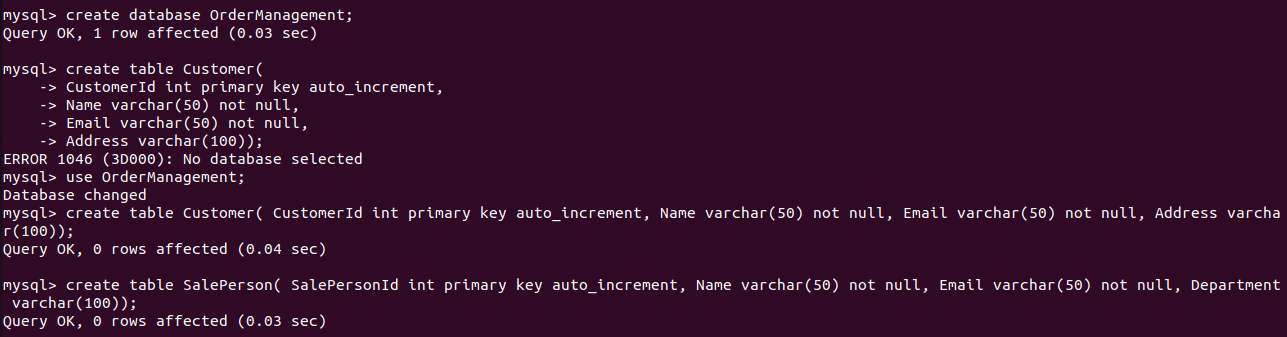
This shows 1 to Many relationship between SalePerson and Order.



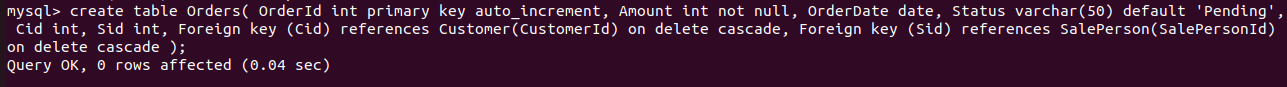
-> Above is the schema that shows three tables each having their attributes.

3. Create tables

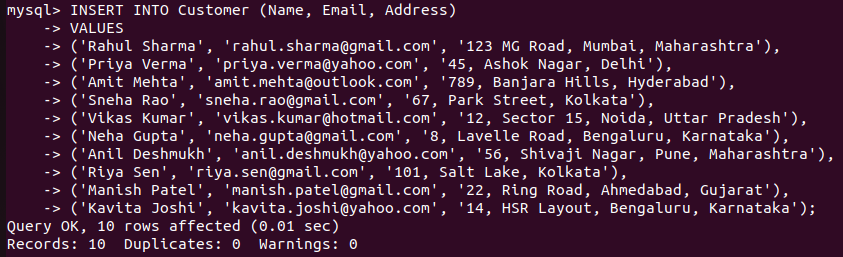
Creating a Customer and SalePerson table.



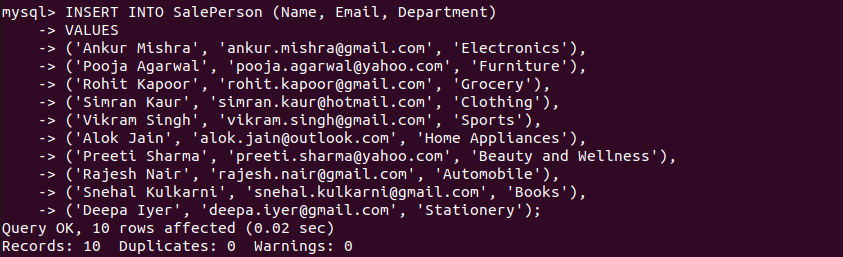
Creating Orders Table:



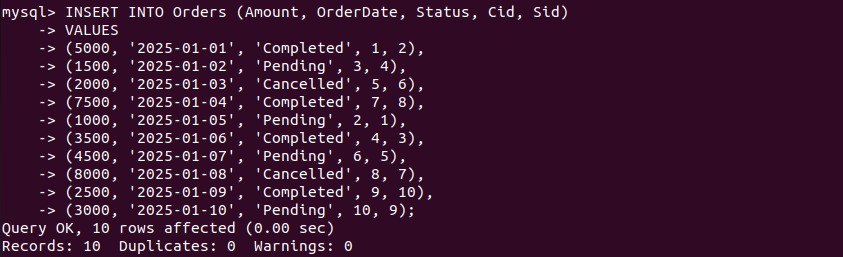
4. Insert sample data

Inserting into customers 

Inserting into SalesPerson



Inserting into Orders



5. Find the sales person who has multiple orders.

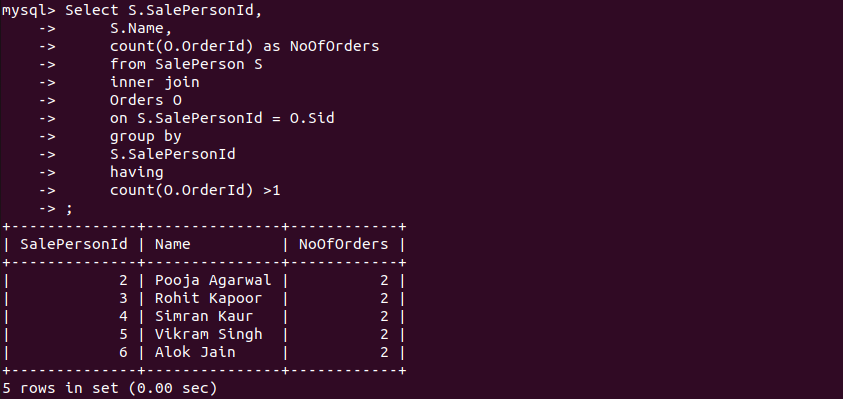
Using the concept of inner join we can find sales person having multiple order



As the set is empty there is no sales person having multiple order

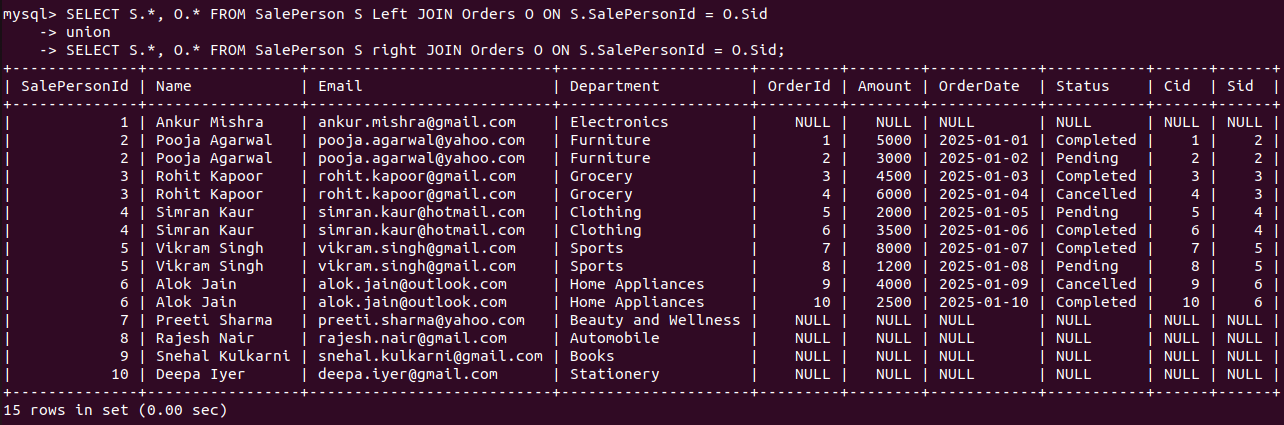
So i truncated the table and updated duplicates in sid

Here is the desired output.



6. Find the all sales person details along with order details

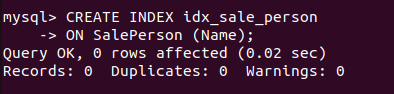
Applying left and right join union will give us all the required data.



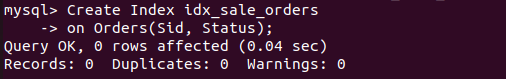
7.Create index

There are various types of index.

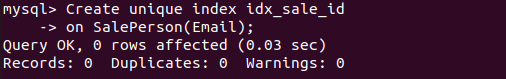
i.Single-Column Index: Index created on a single column.



ii. Composite Index: Index created on multiple columns. Useful when queries filter on more than one column.

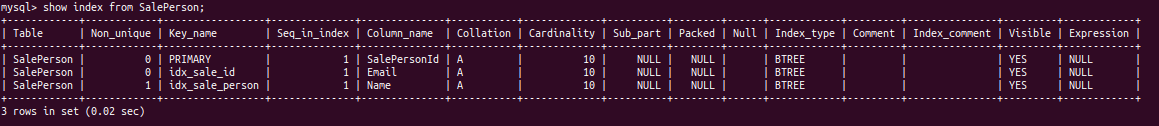


iii. Unique Index: Ensures that all values in the indexed column(s) are unique.



8.How to show index on a table

Using “show index from table\_name;” query



9. Find the order number, sale person name, along with the customer to whom that order belongs to

->This can be achieved by joining multiple tables.

