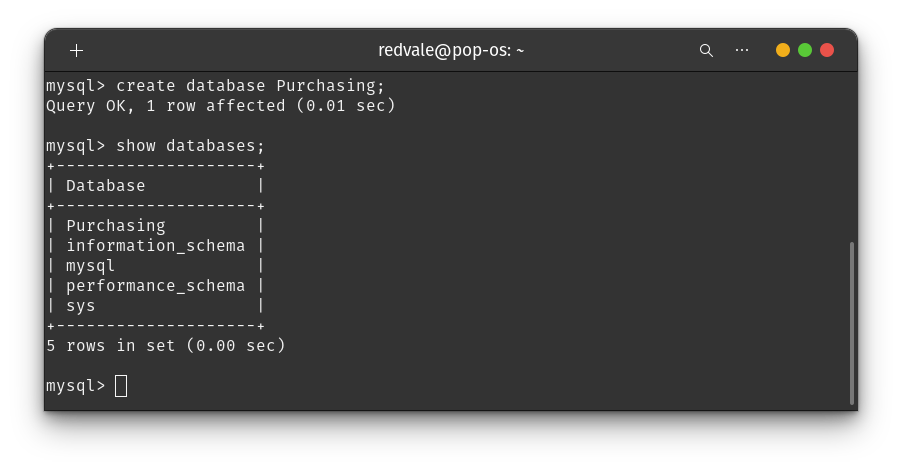
Problem statement : There can be multiple customers, who can place multiple orders on the site. Now these orders will be distributed to multiple sales persons (One order will be assigned to one salesperson only). So a sales person can have multiple orders of multiple customers.

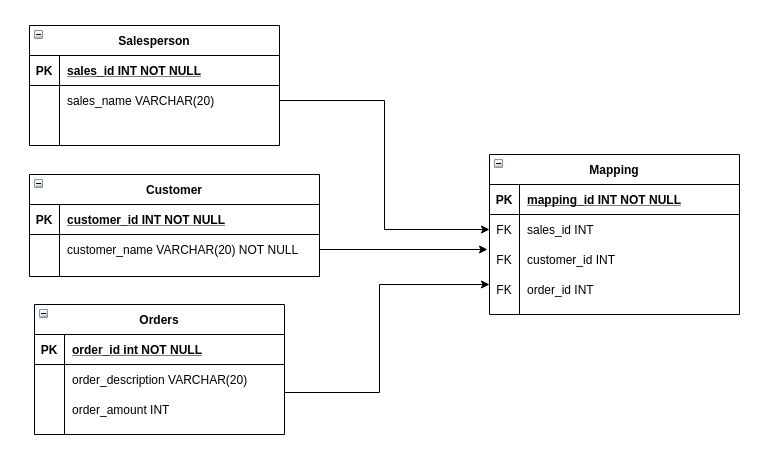
Q1: Create Database

Sol:Database is an organised collection of data. In this we are going to create a database with the name ‘Purchasing’ which will have all the tables related to the problem statement.



Q2: Design Schema

Sol:



We have to use this kind of schema as each entity has many-to-many relationship which will cause redundancy if we try to create FOREIGN KEY inside of tables instead of creating them separately. In this schema we have 3 separate tables of CUSTOMER,SALESPERSON and ORDERS and for linking each of them we have a junction table which has all the FOREIGN KEY needed for linking these tables.

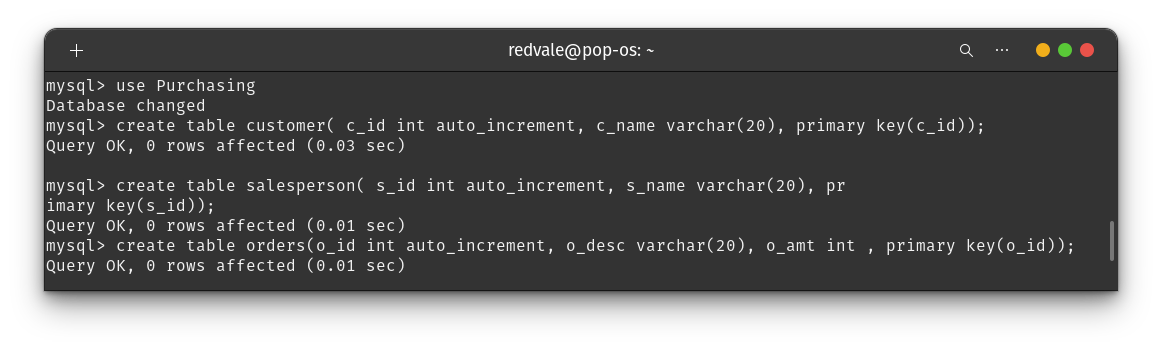
Q3: Create tables

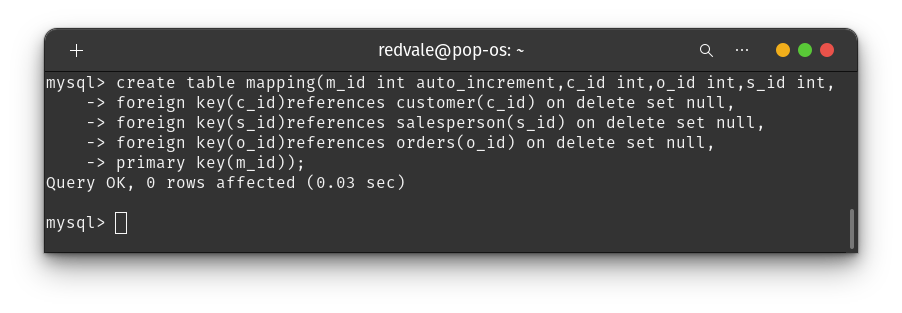
Sol: In this we have created 3 tables :

\*one for customer containing customer\_id and customer\_name

\*second for salesperson containing salesperson\_id and salesperson\_name

\*third for orders containing order\_id, order\_name and order\_amount



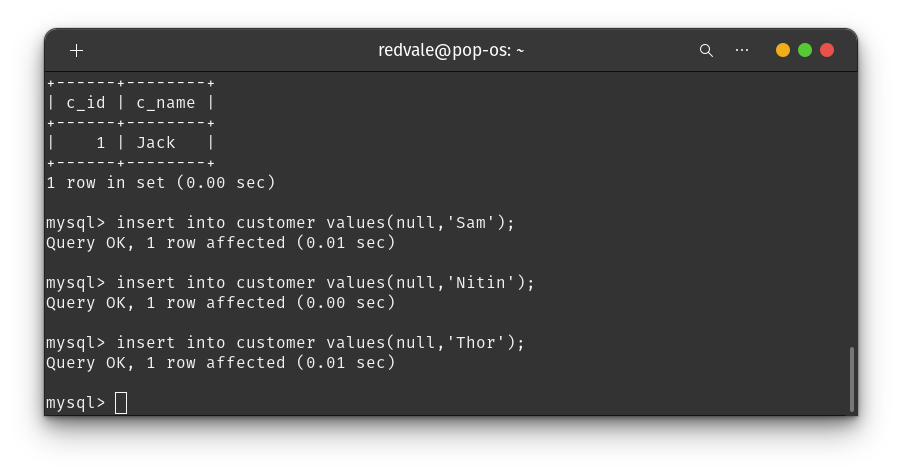




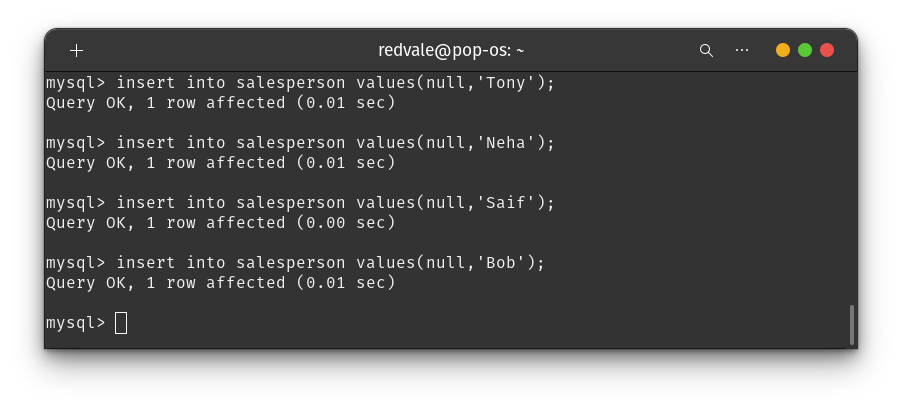
And we have also created a 4th table which will act as a junction table for all the three tables and make us able to solve queries related to the problem statement.

Q4: Insert sample data

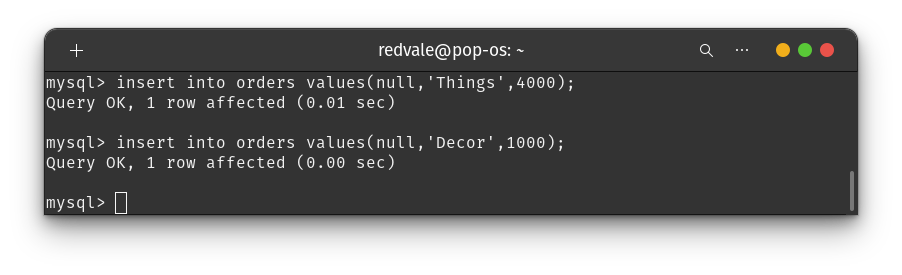
Sol:Inserting data into customer table



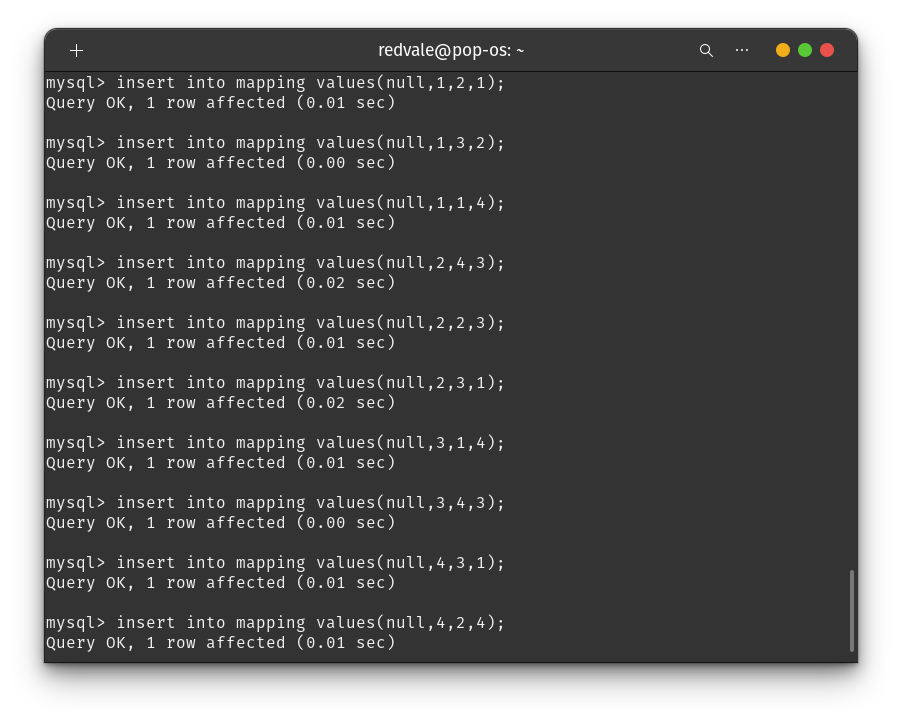
Inserting data into salesperson table



Inserting data into orders table

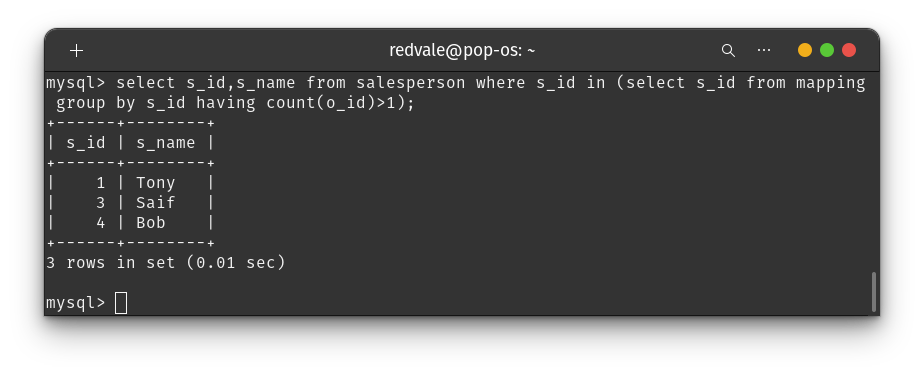


Inserting data into mapping table



Q5:Find the sales person having multiple orders.

Sol: To solve this query we need to get the s\_id of salesperson’s having multiple orders from the mapping table. After that we can use the IN clause and find the salesperson details based on the s\_id fetched from the mapping table.



We can do this using nested queries as done in above screenshot.

Q6:Find the all sales person details along with order details

Sol:To find the details of all the salesperson with their corresponding order details we need to use the mapping table as the mapping table is behaving like a junction between these tables.

And to use mapping table we need to join the tables based on foreign keys which will emulate the connection between them.



On joining the tables we could fetch the data of our need using the select statement and display them as done in above attached screenshot.

Q7:Create index

Sol:Indexes can be created or dropped with no effect on the data. Creating an index involves **the CREATE INDEX statement**, which allows you to name the index, to specify the table and which column or columns to index, and to indicate whether the index is in an ascending or descending order.

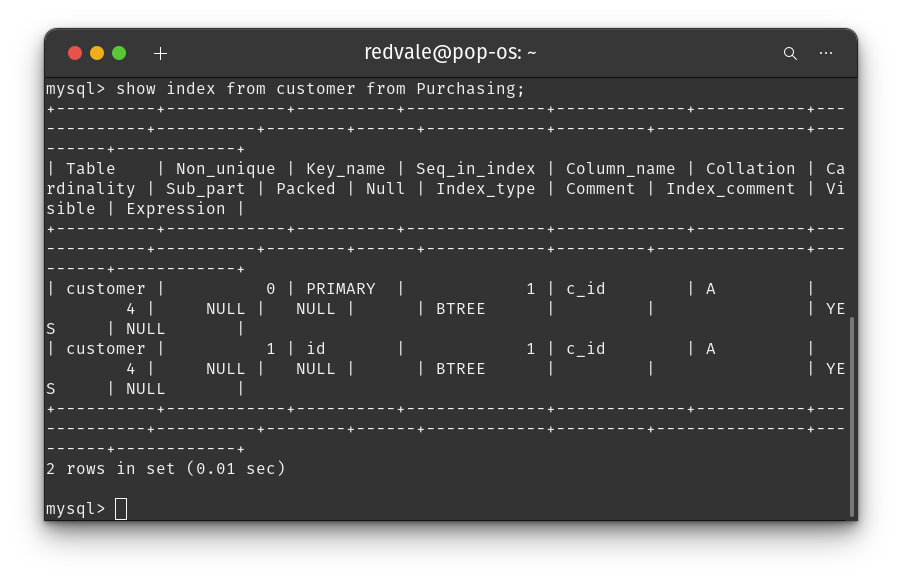


Q8:How to show index on a table

Sol:show index will take two things table name and database name by that we can coordinate and show index on tables.

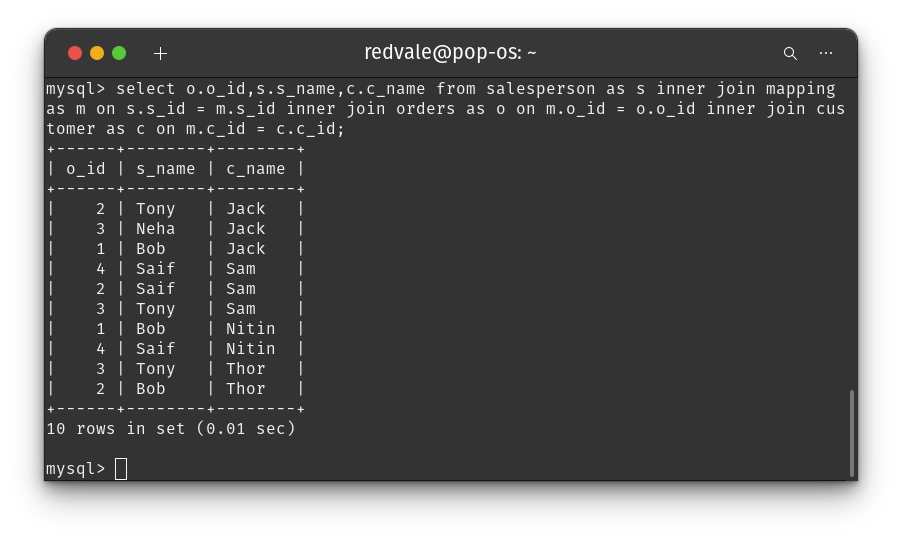
The command for showing index in

SHOW INDEX id FROM <table name> FROM <Database name>



Q9:Find the order number, salesperson name, along with the customer to whom that order belongs to

Sol: For doing these we need to join all the three tables by means of that we can get all three parameters we require for our problem i.e order number, salesperson name and customer name.



After joining the tables we can fetch data of our need using select statement and display the data as done in above attached screenshot.