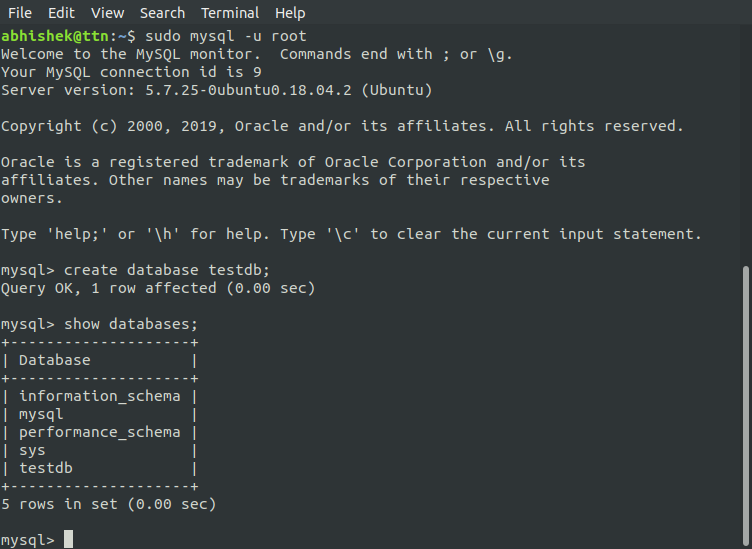
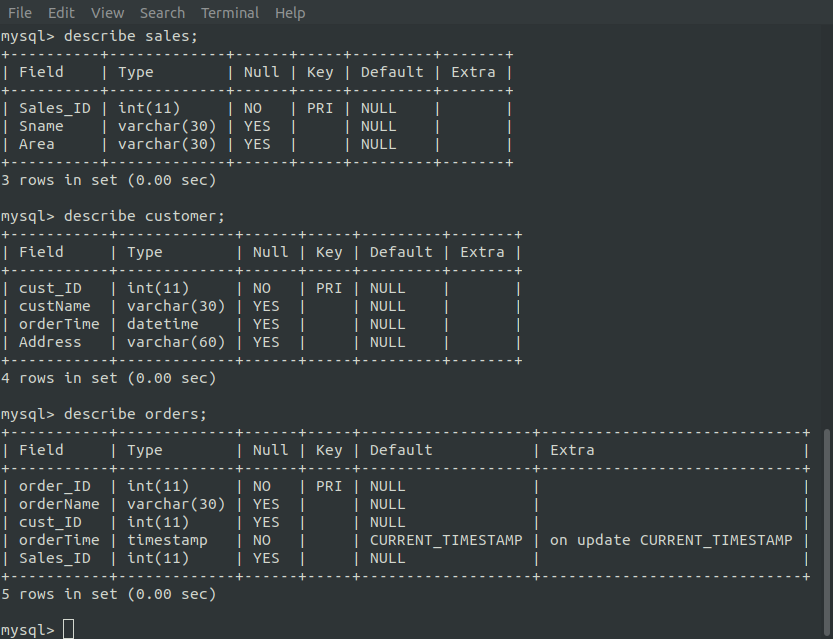
# Introduction to Databases Exercise 3 Abhishek Maurya

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

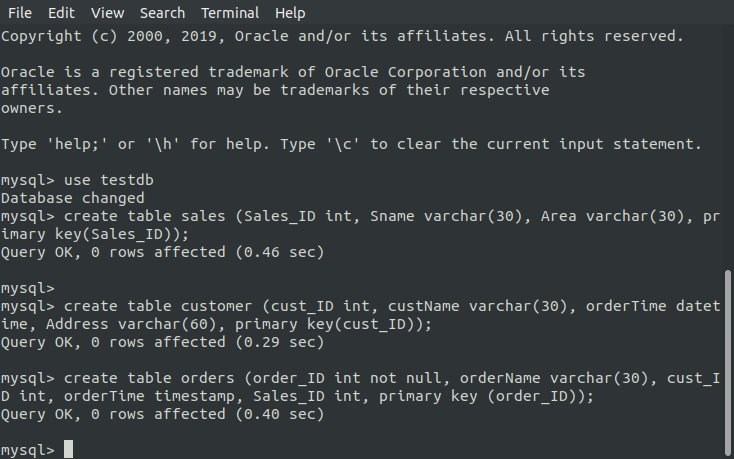
1. **Create Database**
   * create database testdb;



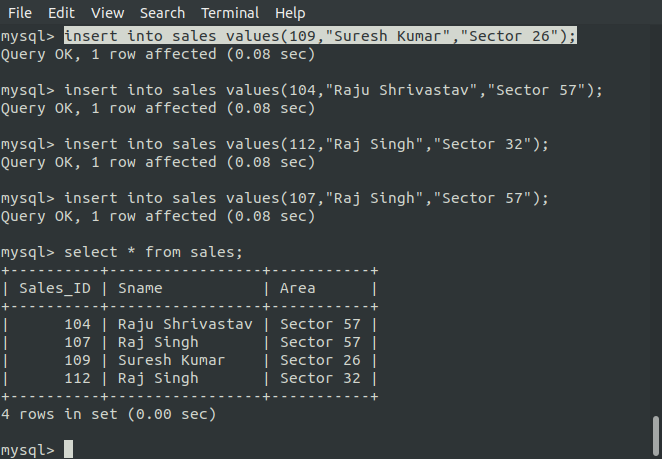
1. **Design Schema**



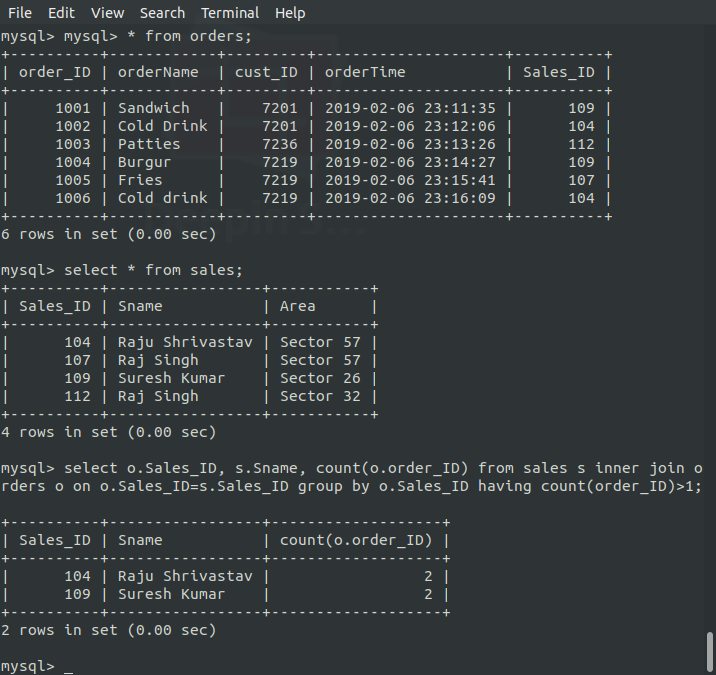
1. **Create tables** 
   * create table sales (Sales\_ID int, Sname varchar(30), Area varchar(30), primary key(Sales\_ID));
   * create table customer (cust\_ID int, custName varchar(30), orderTime datetime, Address varchar(60), primary key(cust\_ID));
   * create table orders (order\_ID int not null, orderName varchar(30), cust\_ID int, orderTime timestamp, Sales\_ID int, primary key (order\_ID));



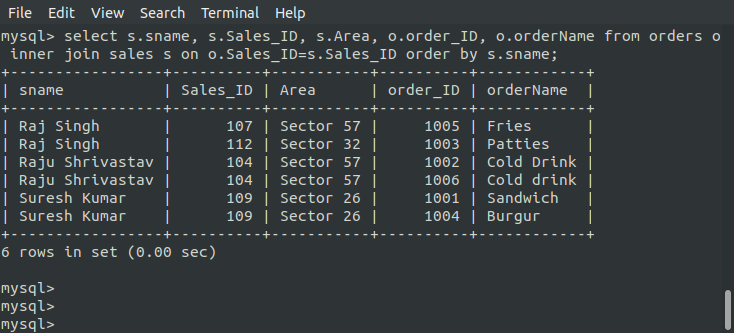
1. **Insert sample data**
   * insert into sales values(109,"Suresh Kumar","Sector 26");



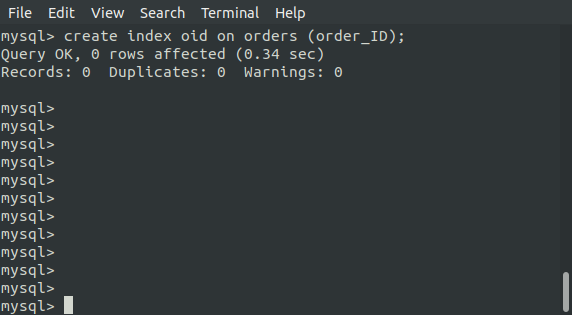
1. **Find the sales person have multiple orders.** 
   * select o.Sales\_ID, s.Sname, count(o.order\_ID) from sales s inner join orders o on o.Sales\_ID=s.Sales\_ID group by o.Sales\_ID having count(order\_ID)>1;



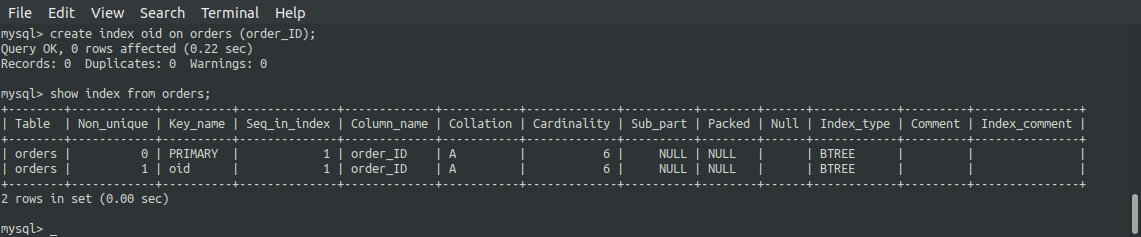
1. **Find the all sales person details along with order details**
   * select s.sname, s.Sales\_ID, s.Area, o.order\_ID, o.orderName from orders o inner join sales s on o.Sales\_ID=s.Sales\_ID order by s.sname;



1. **Create index**
   * create index oid on orders (order\_ID);



1. **How to show index on a table**
   * show index from others;



1. **Find the order number, salesperson name, along with the customer to whom that order belongs to**
   * select o.order\_ID, s.sname, c.custName, o.orderName from orders o inner join sales s inner join customer c on o.Sales\_ID=s.Sales\_ID and o.cust\_ID=c.cust\_ID;

