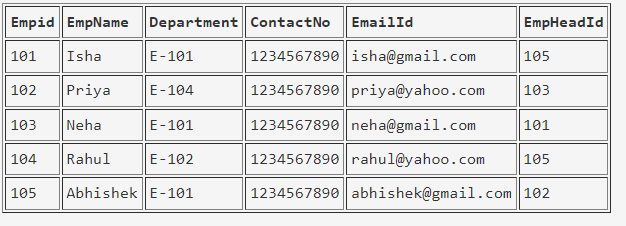
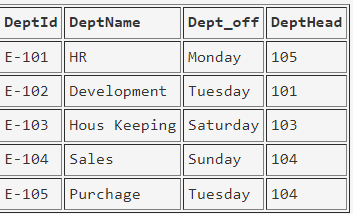
**Employee**



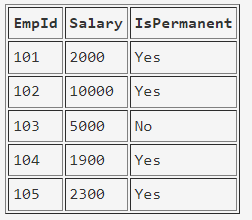
create table employee(empid int primary key,empname varchar(100), department varchar(50),contactno bigint, emaildid varchar(100), empheadid int)

**EmpDept**



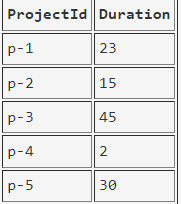
create table empdept(deptid varchar(50) primary key,deptname varchar(100), dept\_off varchar(100), depthead int foreign key references employee(empid))

**EmpSalary**



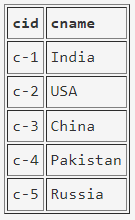
create table empsalary(empid int foreign key references employee(empid), salary bigint, ispermanent varchar(3))

**Project**



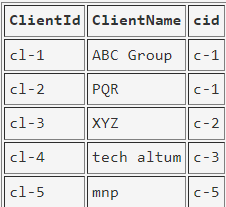
create table project(projectid varchar(50) primary key, duration int)

**Country**



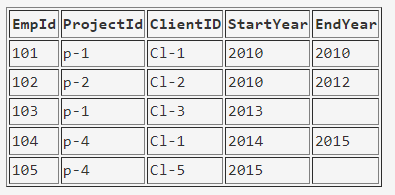
create table country(cid varchar(50) primary key, cname varchar(100))

**ClientTable**



create table clienttable(clientid varchar(50) primary key, clientname varchar(100), cid varchar(50) references country(cid))

**EmpProject**



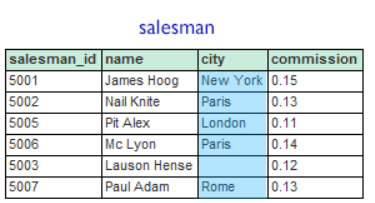
create table empproject(empid int foreign key references employee(empid), projectid varchar(50) foreign key references project(projectid), clientid varchar(50) foreign key references clienttable(clientid),startyear int, endyear int)

**Query:**

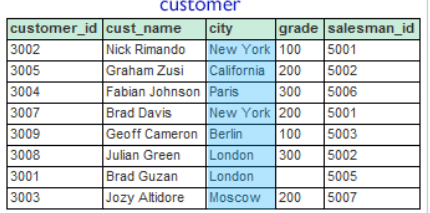
1. Select the detail of the employee whose name start with P.
2. How many permanent candidate take salary more than 5000.
3. Select the detail of employee whose emailId is in gmail.
4. Select the details of the employee who work either for department E-104 or E-102.
5. What is the department name for DeptID E-102?
6. What is total salarythat is paid to permanent employees?
7. List name of all employees whose name ends with a.
8. List the number of department of employees in each project.
9. How many project started in year 2010.
10. How many project started and finished in the same year.
11. select the name of the employee whose name's 3rd charactor is 'h'.

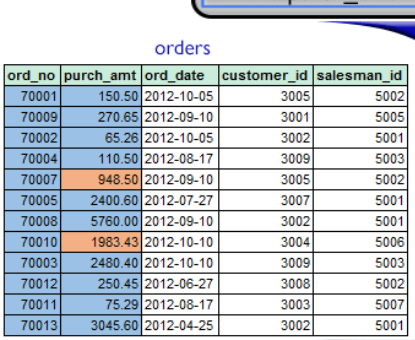
**Assignment 2**

***Sample table*: salesman**

****

***Sample table*: customer**

****

****

1. From the above tables write a SQL query to find the salesperson and customer who reside in the same city. Return Salesman, cust\_name and city.
2. write a SQL query to find those orders where the order amount exists between 500 and 2000. Return ord\_no, purch\_amt, cust\_name, city.
3. write a SQL query to find the salesperson(s) and the customer(s) he represents. Return Customer Name, city, Salesman, commission.
4. write a SQL query to find salespeople who received more than 12 percent commissions from the company. Return Customer Name, customer city, Salesman, commission.
5. write a SQL query to locate those salespeople who do not live in the same city where their customers live and have received a commission of more than 12% from the company. Return Customer Name, customer city, Salesman, salesman city, commission.
6. write a SQL query to find the details of an order. Return ord\_no, ord\_date, purch\_amt, Customer Name, grade, Salesman, commission.
7. Write a SQL statement to join the tables salesman, customer and orders so that the same column of each table appears once and only the relational rows are returned.
8. write a SQL query to display the customer name, customer city, grade, salesman, salesman city. The results should be sorted by ascending customer\_id.
9. write a SQL query to find those customers with a grade less than 300. Return cust\_name, customer city, grade, Salesman, salesmancity. The result should be ordered by ascending customer\_id.
10. Write a SQL statement to make a report with customer name, city, order number, order date, and order amount in ascending order according to the order date to determine whether any existing customers have placed an order.