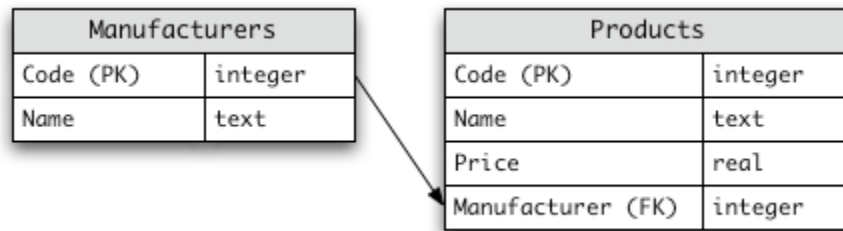


DBMS Assignment-4 on SQL

1. (a) Create the following Tables in your database:



```
CREATE TABLE Manufacturers (  
    Code INTEGER,  
    Name VARCHAR(255) NOT NULL,  
    PRIMARY KEY (Code)  
);
```

```
CREATE TABLE Products (  
    Code INTEGER,  
    Name VARCHAR(255) NOT NULL ,  
    Price DECIMAL NOT NULL ,  
    Manufacturer INTEGER NOT NULL,  
    PRIMARY KEY (Code),  
    FOREIGN KEY (Manufacturer) REFERENCES Manufacturers(Code)  
);
```

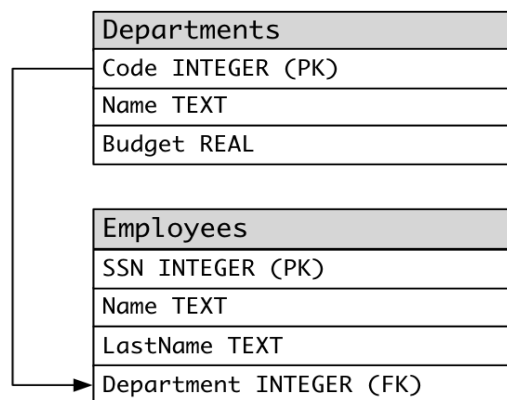
```
INSERT INTO Manufacturers(Code,Name) VALUES(1,'Sony');  
INSERT INTO Manufacturers(Code,Name) VALUES(2,'Creative Labs');  
INSERT INTO Manufacturers(Code,Name) VALUES(3,'Hewlett-Packard');  
INSERT INTO Manufacturers(Code,Name) VALUES(4,'Iomega');  
INSERT INTO Manufacturers(Code,Name) VALUES(5,'Fujitsu');  
INSERT INTO Manufacturers(Code,Name) VALUES(6,'Winchester');  
INSERT INTO Products(Code,Name,Price,Manufacturer) VALUES(1,'Hard drive',240,5);  
INSERT INTO Products(Code,Name,Price,Manufacturer) VALUES(2,'Memory',120,6);  
INSERT INTO Products(Code,Name,Price,Manufacturer) VALUES(3,'ZIP drive',150,4);  
INSERT INTO Products(Code,Name,Price,Manufacturer) VALUES(4,'Floppy disk',5,6);  
INSERT INTO Products(Code,Name,Price,Manufacturer) VALUES(5,'Monitor',240,1);  
INSERT INTO Products(Code,Name,Price,Manufacturer) VALUES(6,'DVD drive',180,2);  
INSERT INTO Products(Code,Name,Price,Manufacturer) VALUES(7,'CD drive',90,2);  
INSERT INTO Products(Code,Name,Price,Manufacturer) VALUES(8,'Printer',270,3);  
INSERT INTO Products(Code,Name,Price,Manufacturer) VALUES(9,'Toner cartridge',66,3);  
INSERT INTO Products(Code,Name,Price,Manufacturer) VALUES(10,'DVD burner',180,2);
```

(b) Implement the following queries in MySQL with the Tables generated in Question 1(a):

- i. Compute the average price of all products with manufacturer code equal to 2.
- ii. Compute the number of products with a price larger than or equal to \$180.

- iii. Select the name and price of all products with a price larger than or equal to \$180, and sort first by price (in descending order), and then by name (in ascending order).
- iv. Select all the data from the products, including all the data for each product's manufacturer.
- v. Select the average price of each manufacturer's products, showing only the manufacturer's code.
- vi. Select the product name, price, and manufacturer name of all the products.
- vii. Select the names of manufacturer whose products have an average price larger than or equal to \$150.
- viii. Select the name and price of the cheapest product.
- ix. Select the name of each manufacturer along with the name and price of its most expensive product. Apply a 10% discount to all products with a price larger than or equal to \$120.
- x. Apply a 10% discount to all products with a price larger than or equal to \$120.

2. (a) Create the following Tables in your database:



CREATE TABLE
Departments (
Code INTEGER

PRIMARY KEY,
Name varchar(255) NOT NULL ,
Budget decimal NOT NULL
);

CREATE TABLE Employees (
SSN INTEGER PRIMARY KEY,
Name varchar(255) NOT NULL ,
LastName varchar(255) NOT NULL ,
Department INTEGER NOT NULL ,
foreign key (department) references Departments(Code)
);

INSERT INTO Departments(Code,Name,Budget) VALUES(14,'IT',65000);
INSERT INTO Departments(Code,Name,Budget) VALUES(37,'Accounting',15000);
INSERT INTO Departments(Code,Name,Budget) VALUES(59,'Human Resources',240000);
INSERT INTO Departments(Code,Name,Budget) VALUES(77,'Research',55000);

INSERT INTO Employees(SSN,Name,LastName,Department)
VALUES('123234877','Michael','Rogers',14);

```

INSERT INTO Employees(SSN,Name,LastName,Department)
VALUES('152934485','Anand','Manikutty',14);
INSERT INTO Employees(SSN,Name,LastName,Department)
VALUES('222364883','Carol','Smith',37);
INSERT INTO Employees(SSN,Name,LastName,Department)
VALUES('326587417','Joe','Stevens',37);
INSERT INTO Employees(SSN,Name,LastName,Department) VALUES('332154719','Mary-
Anne','Foster',14);
INSERT INTO Employees(SSN,Name,LastName,Department)
VALUES('332569843','George','ODonnell',77);
INSERT INTO Employees(SSN,Name,LastName,Department)
VALUES('546523478','John','Doe',59);
INSERT INTO Employees(SSN,Name,LastName,Department)
VALUES('631231482','David','Smith',77);
INSERT INTO Employees(SSN,Name,LastName,Department)
VALUES('654873219','Zacary','Efron',59);
INSERT INTO Employees(SSN,Name,LastName,Department)
VALUES('745685214','Eric','Goldsmith',59);
INSERT INTO Employees(SSN,Name,LastName,Department)
VALUES('845657245','Elizabeth','Doe',14);
INSERT INTO Employees(SSN,Name,LastName,Department)
VALUES('845657246','Kumar','Swamy',14);

```

(b) Implement the following queries in MySQL with the Tables generated in Question 2(a):

- i. Select the number of employees in each department (you only need to show the department code and the number of employees).
- ii. Select the name and last name of each employee, along with the name and budget of the employee's department.
- iii. Select the name and last name of employees working for departments with a budget greater than \$60,000.
- iv. Select the departments with a budget larger than the average budget of all the departments.
- v. Select the names of departments with more than two employees.
- vi. Select the name and last name of employees working for departments with second lowest budget.
- vii. Reduce the budget of all departments by 10%.
- viii. Reassign all employees from the Research department (code 77) to the IT department (code 14).
- ix. Add a new department called "Quality Assurance", with a budget of \$40,000 and departmental code 11. And Add an employee called "Mary Moore" in that department, with SSN 847-21-9811.
- x. Delete from the table all employees who work in departments with a budget greater than or equal to \$60,000.

3. Put all your screenshots(query with output) in a single PDF file and upload(10)