A Solution to Assignment 1

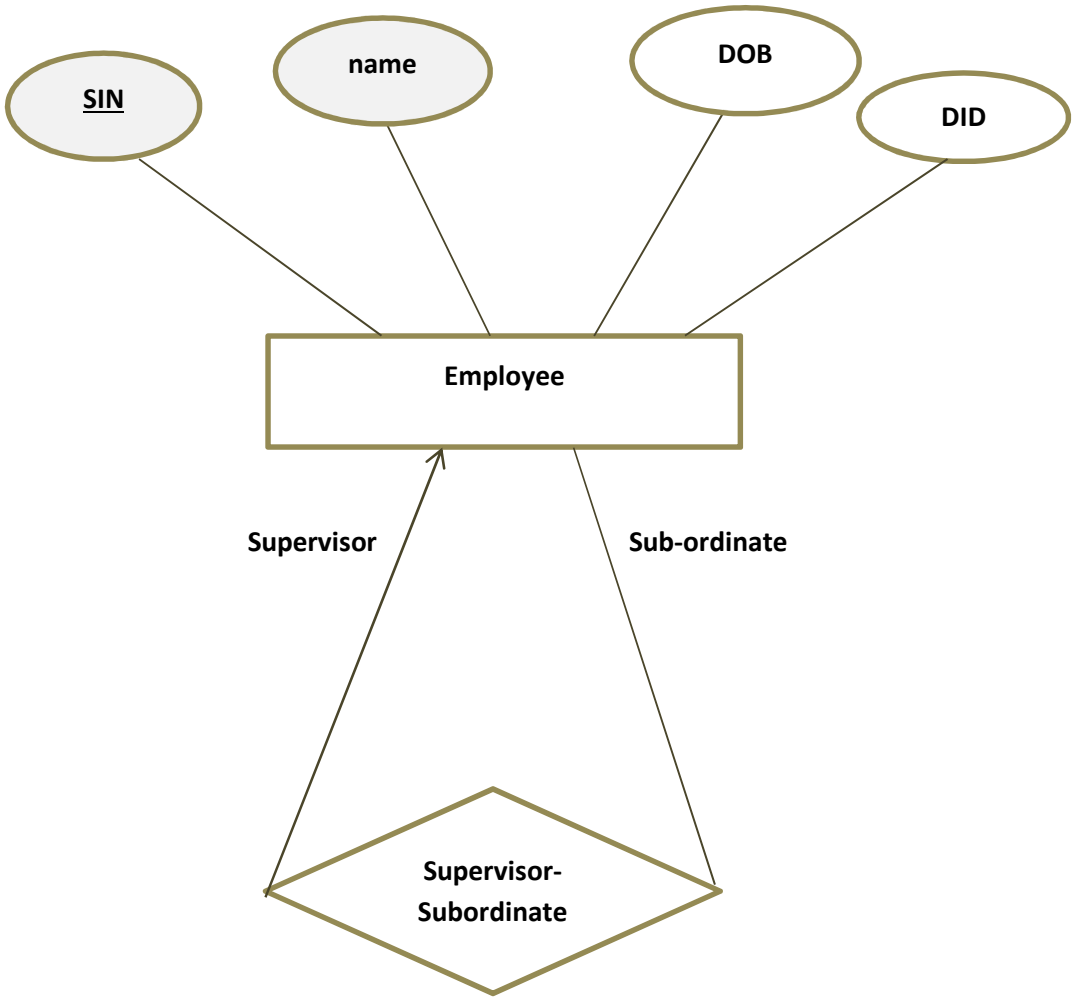
Q2. The E/R diagram for this database includes an entity set *Employee* and two roles from Employee to itself, where the role *Supervisor* is many-to-one and *Subordinate* is one-to-many.

The diagram is converted into the following two relations with the key attributes underline.

Employee (SIN, Name, DOB, DID)
Supervised-by (SIN-Employee, SIN-Supervisor)

The relation Supervised-by records the immediate supervisor of each employee.

Note: Perhaps it is better to add to Employee relation the new attribute EmployeeID and use it as the primary key of the relation. In this case, SIN would just be a unique attribute and not a key.



Q3. Product = {manufacturerID, model, eq-type}
PC = {model, speed, ram, hd, price}
Laptop = {model, speed, ram, hd, screen, price}
Printer = {model, color, type, price}

To be more realistic, *manufacture* and *model* together will form the key for Product table.

Part (1). Creation of tables.

```
CREATE TABLE Product (
        manufacturer CHAR(20),
        model  CHAR(16) PRIMARY KEY,
        Eq-type CHAR(15)
        );

CREATE TABLE PC (
        model CHAR(16) PRIMARY KEY,
        speed DECIMAL(4,2),
        ram INTEGER,
        hd INTEGER,
        price DECIMAL(7,2)
        );

CREATE TABLE Laptop (
        model CHAR(16) PRIMARY KEY,
        speed DECIMAL(4,2),
        ram INTEGER,
        hd INTEGER,
        screen DECIMAL(3,1),
        price DECIMAL(7,2)
       );

CREATE TABLE Printer (
        model CHAR(16) PRIMARY KEY,
        color BOOLEAN,
        type CHAR (10),
        price DECIMAL(7,2)
     );
```

(2) ALTER TABLE Printer DROP color;

(3) ALTER TABLE Laptop ADD odt CHAR (10) DEFAULT 'none';

Q4.

(1) Account Number: INT
AccType: Char (2)
Balance: Decimal (10,2)
OpenDate: Date
CustID: INT
FirstName: CHAR (15)
LastName: CHAR (15)
Address: CHAR (255)
AccNumber: INT

(2) Should be easy!

(3) The proposed design does not support clients to have joint accounts or multiple accounts. The problem can be fixed by removing the attribute AccNumber from Customer, and creating a relationship CustAcc = {CustID, AccNumber} between the entity sets Customer and Account. Another problem is that there is no provision to record transactions! This is fixed by adding another relationship called Transaction between Customer and Account. The schema of Transaction would include the attributes CustID, AccNumber, TransactionType, Date, and Amount, where the type of transaction could be either deposit or withdraw. The key attributes are CustID, AccNumber, and Date. Note that, for each customer, this allows one transaction per day. To fix this, add the attribute the Time to the schema. Note that while this design improves some of the limitation of the original design, it may not be quite desired.