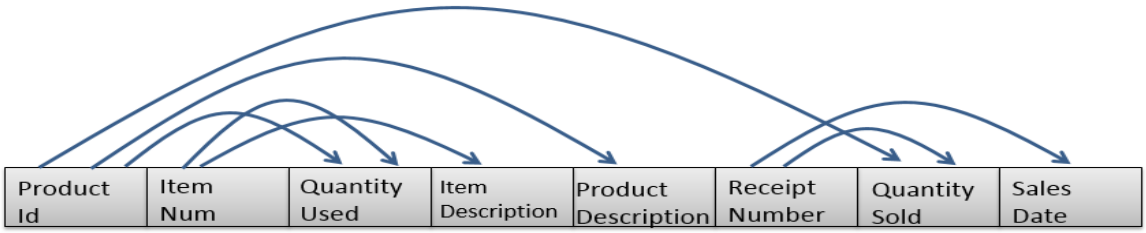
Name: Omid Azodi
Due Date: 03/10/2016
Professor: Dr. Hadaegh
Class: CS443

Assignment # 1

Question #1

Consider the following data. Arrows show the functional dependency.



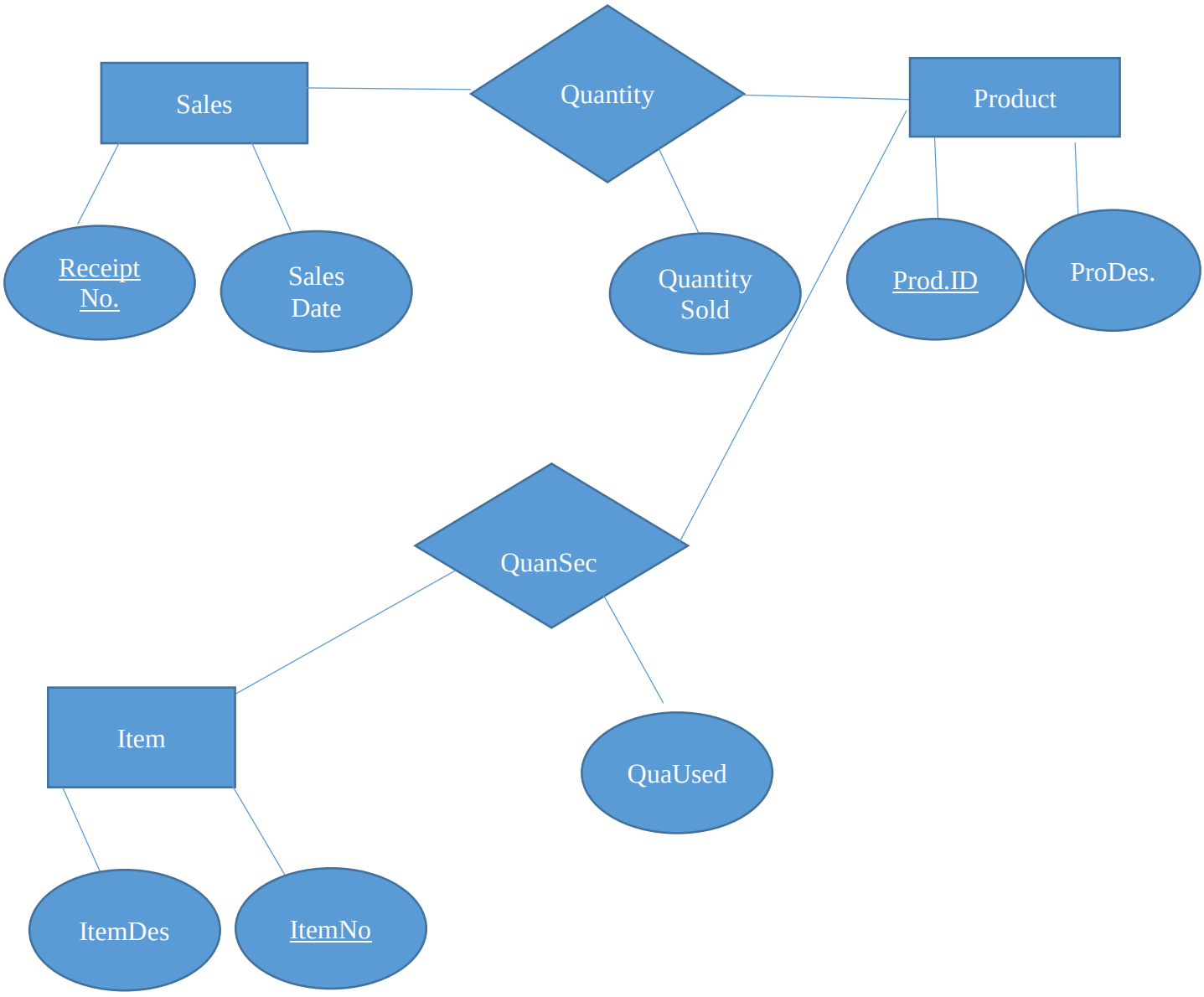
a) Write the tables

Sales(ReceiptNumber, SalesDate)
Quantity(ProductID, ReceiptNumber, QuantitySold)
Product(ProductID, ProductDescription)
Item(ItemNum, ItemDescription)
Quantity2(ProductID, ItemNum, QuantityUsed)

b) Place the tables in 3rd normal form (if necessary)

The tables are already in 3rd normal form. I knew this, because I checked for the ones that only had a composite primary key. However for the tables that had a composite primary key there was no transitive or derived dependency.

c) Create ERD based on the normalized tables

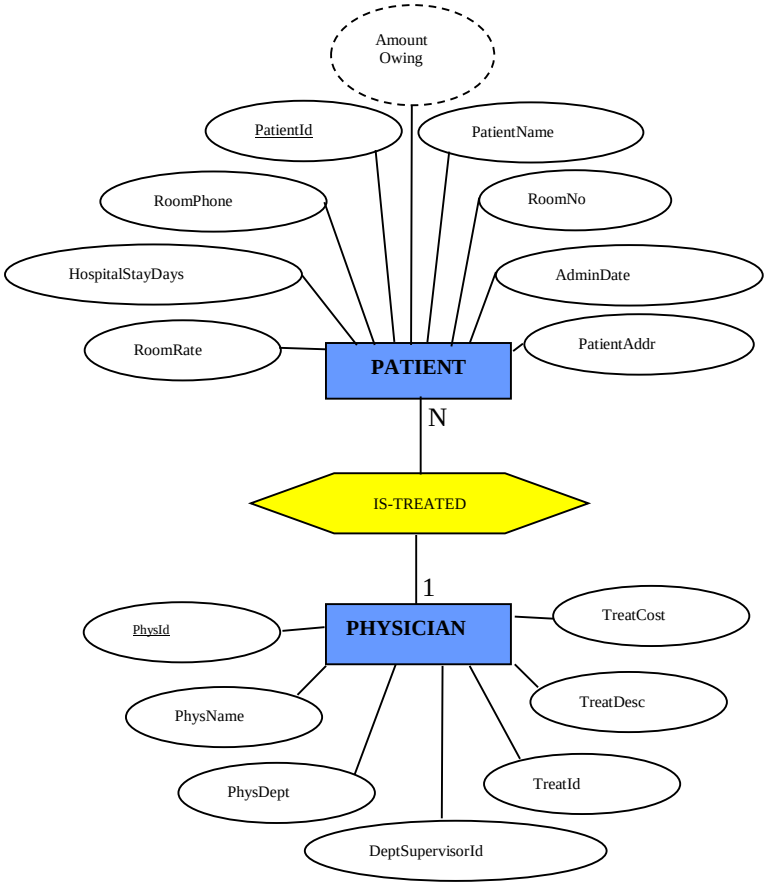


d) Write a script to create a database. Your script should create the tables and ensures that all constraints are set properly.

```
CREATE TABLE Sales
(
ReceiptNumber INTEGER,
SalesDate DATE,
CONSTRAINT pk_SalesTable PRIMARY KEY(ReceiptNumber)
);

CREATE TABLE Quantity
(
ProductID INTEGER,
ReceiptNumber INTEGER,
QuantitySold INTEGER,
CONSTRAINT pk_QuantityTable PRIMARY KEY (ProductID,ReceiptNumber),
CONSTRAINT fk_ProductSecond FOREIGN KEY(ProductID) REFERENCES
Product(ProductID),
CONSTRAINT fk_SalesForeign FOREIGN KEY (ProductID) REFERENCES
Sales(ReceiptNumber),
CONSTRAINT checkQuanSold CHECK(QuantitySold >= 0)
);

CREATE TABLE Product
(
ProductID INTEGER,
ProductDescription VARCHAR(200),
CONSTRAINT pk_ProductPrimary PRIMARY KEY (ProductID)
);

CREATE TABLE Item
(
ItemNum INTEGER,
ItemDescription VARCHAR(200),
CONSTRAINT pk_ItemPrimary PRIMARY KEY (ItemNum)
);

CREATE TABLE QuantitySecond
(
 ProductID INTEGER,
 ItemNum INTEGER,
 QuantityUsed INTEGER,
 CONSTRAINT pk_ProductAndItem PRIMARY KEY (ProductID, ItemNum),
 CONSTRAINT fk_ItemForeign FOREIGN KEY (ItemNum) REFERENCES
Item(ItemNum),
 CONSTRAINT fk_ProductForeign FOREIGN KEY (ProductID) REFERENCES
Product(ProductID),
 CONSTRAINT checkQuanUsed CHECK (QuantityUsed >= 0)
 );
```

Question #2

Consider the following ERD



Note: The arrows are drawn on HARD COPY. The tables are NOT separate they are supposed to be combined into one big long table, however since it would not fit I had to make them separate, however they should be merged from Room Rate and PhysID.

a) Using arrows draw the dependencies that exist in the data.

Patient Id	Patient Name	Patient Addr	AdminDate	AmoutnOwning	RoomNo	RoomPhone	Hosp. Stay Day	Room Rate

PhysID	PhysName	PhysDept	DeptSupervisorId	TreatId	TreatDesc	TreatCost

b) Write the initial tables

Patient(PatientID, PatientName, PatientAddr., OwingAmt., RoomPhone, Hosp.DaySpent, RoomRate, RoomNo, AdminDate)

Physician(PhysID, PhysName, PhysDept., Dept.SupervisorId, TreatId, TreatDesc, TreatCost)

c) Place the tables in 3rd normal form (if necessary)

Patient(PatientID, PatientName, PatientAddr., AdmitDate, Hosp.DayStay, RoomNo, PhysID)

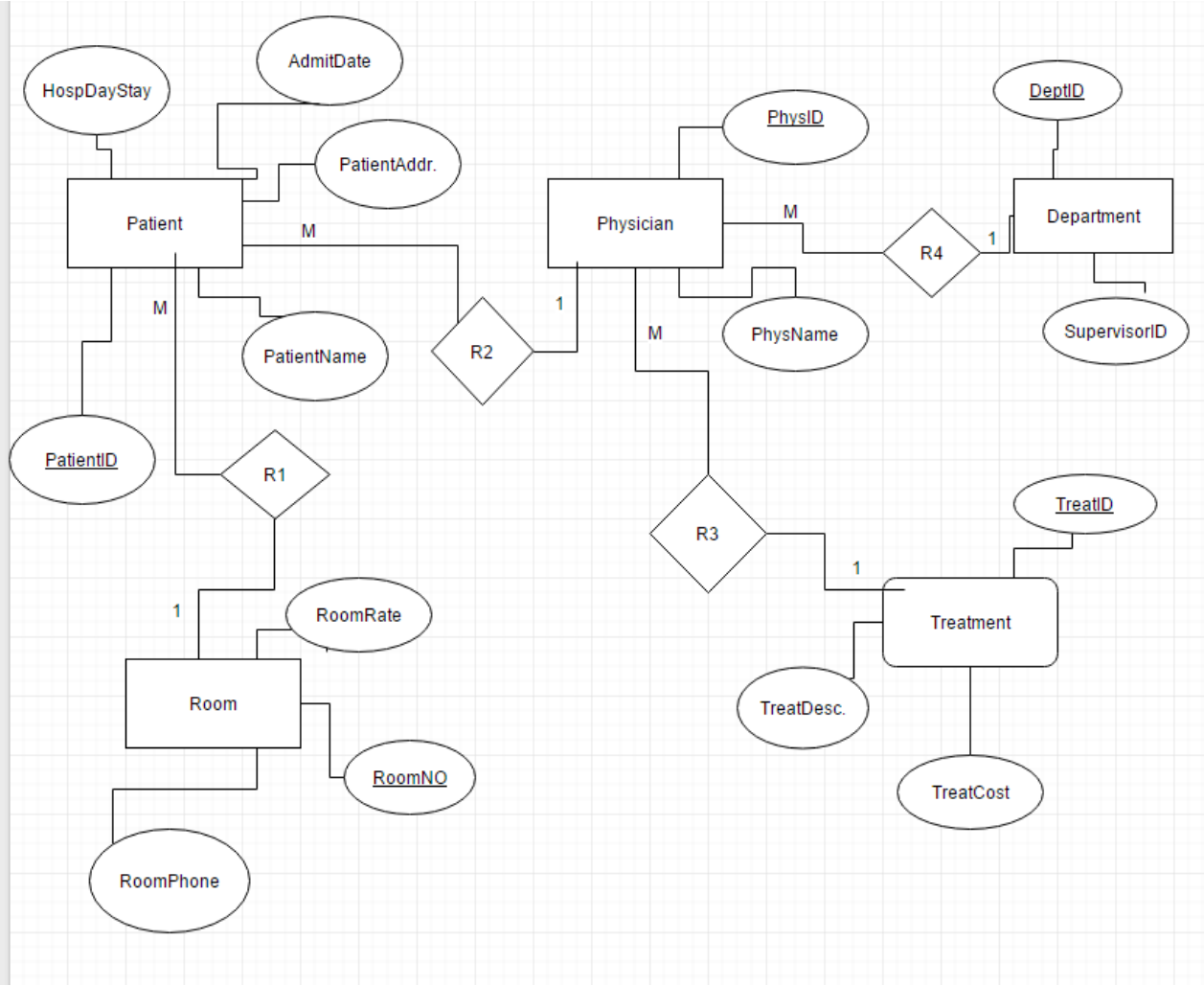
Physician(PhysID, PhysName, PhysDept., TreatId)

Department(PhysDeptID, Dept.SupervisorID)

Room(roomNO., roomPhone, roomRate)

Treatment(treatID, treatDesc., treatCost)

d) Revise the given ERD based on the normalized tables



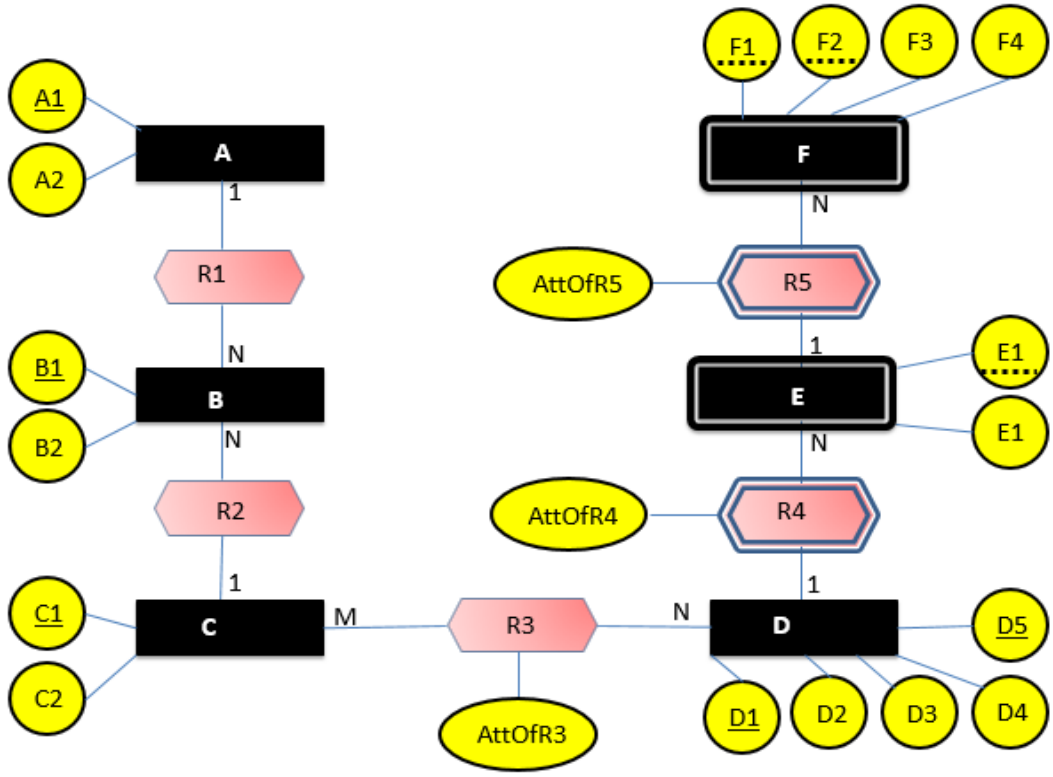
e) Write a script to create a database. Your script should create the tables and ensures that all constraints are set properly. **Note: There is attached SQL File to RUN.**

```
CREATE TABLE Patient
(
PatientID INTEGER,
PatientName VARCHAR2(50)
CONSTRAINT PatientNameNotEmpty NOT NULL,
PatientAddress VARCHAR2(200)
CONSTRAINT PatientAddressNotEmpty NOT NULL,
AdmitDate DATE,
HospitalStayDays INTEGER,
RoomNo INTEGER,
PhysID INTEGER,
CONSTRAINT pk_PrimaryPatientIDs PRIMARY KEY (PatientID),
CONSTRAINT fk_RoomNoForeigns FOREIGN KEY (RoomNo) REFERENCES
Room(RoomNo),
CONSTRAINT fk_PhysIDForeigns FOREIGN KEY (PhysID) REFERENCES
Physician(PhysID),
CONSTRAINT checkRoomNos CHECK ((RoomNo >=100) AND (RoomNo <=999)),
CONSTRAINT checkHospitalStayDays CHECK (HospitalStayDays >=0)
);

CREATE TABLE Physician
(
PhysID INTEGER,
PhysName VARCHAR2(50)
CONSTRAINT PhysNameNotEmpty NOT NULL,
PhysDept INTEGER,
TreatID INTEGER,
CONSTRAINT pk_PhysIDPrimary PRIMARY KEY (PhysID),
CONSTRAINT fk_TreatIDForeign FOREIGN KEY (TreatID) REFERENCES
Treatment(TreatID),
CONSTRAINT fk_PhysIDForeign FOREIGN KEY (PhysDept) REFERENCES
Department(PhysDeptId)
);

CREATE TABLE Department
(
PhysDeptId INTEGER,
DeptSupervisorId INTEGER,
CONSTRAINT pk_PhysDeptIDPrimary PRIMARY KEY (PhysDeptId)
);
```

```
CREATE TABLE Room
(
RoomNo INTEGER,
RoomPhone VARCHAR2(8),
RoomRate NUMBER(10,2),
CONSTRAINT pk_RoomNoPrimarys PRIMARY KEY (RoomNo),
CONSTRAINT checkRoomNo CHECK ((RoomNo >=100) AND (RoomNo <=999)),
CONSTRAINT RoomRateRange CHECK ((RoomRate >=30.00) AND (RoomRate
<=100.00))
);

CREATE TABLE Treatment
(
TreatID INTEGER,
TreatDescription VARCHAR2(200),
TreatCost NUMBER(10,2),
CONSTRAINT pk_TreatIDPrimarys PRIMARY KEY (TreatID),
CONSTRAINT TreatCostChecking CHECK (TreatCost >=50.00)
);


ALTER TABLE DEPARTMENT ADD CONSTRAINT fk_PhysDeptIDForeign
FOREIGN KEY (DeptSupervisorId) REFERENCES Physician(PhysID)
```

Question #3

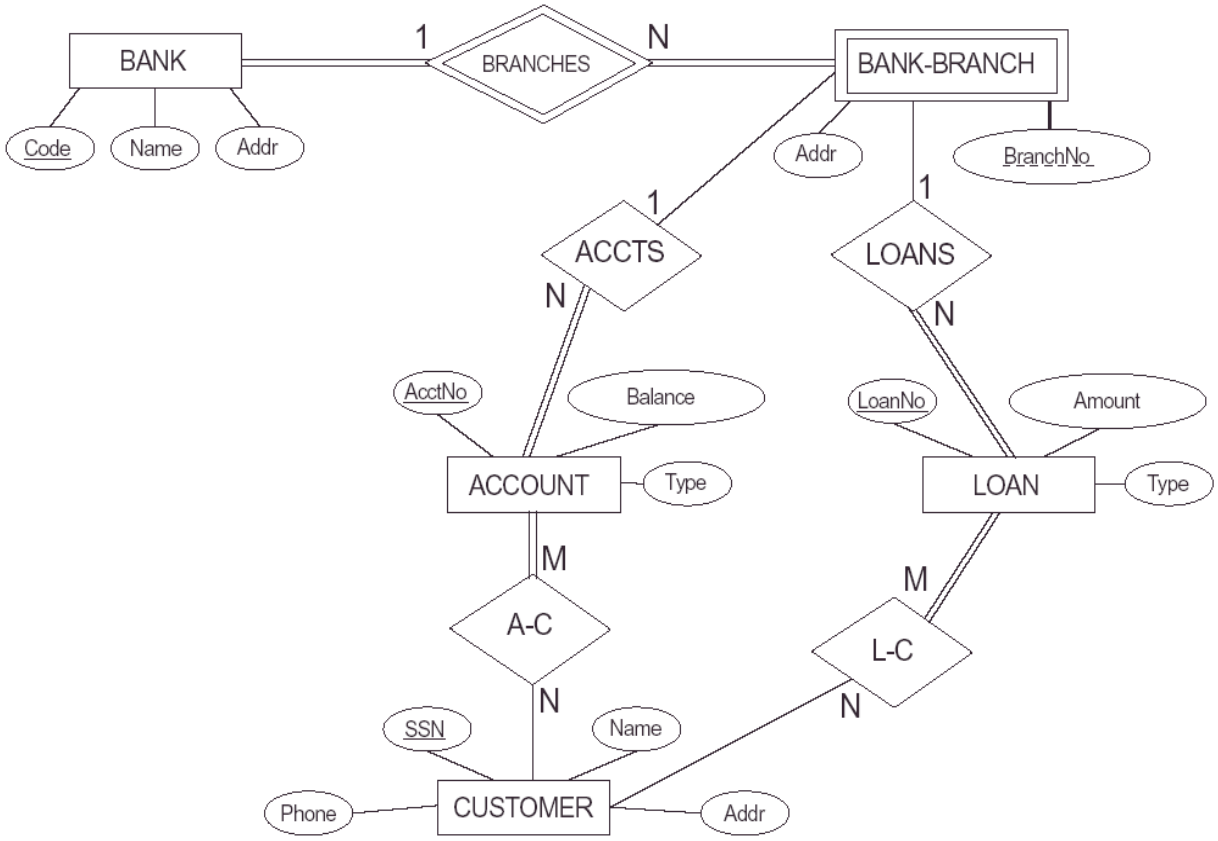
Create the tables related to the following ERD



First Table: A(A1, A2)
Second Table: B(B1, B2, A1, C1)
Third Table: C(C1,C2)
Fourth Table: R3(C1, D1,D5, AttOfR3)
Fifth Table: D(D1,D5,D2,D3,D4)
Sixth Table: E(E1,D1,D5,E2, AttOfR4)
Seventh Table: F(F1,F2,E1,D1,D5,F3,F4,AttOfR5)

Question # 4

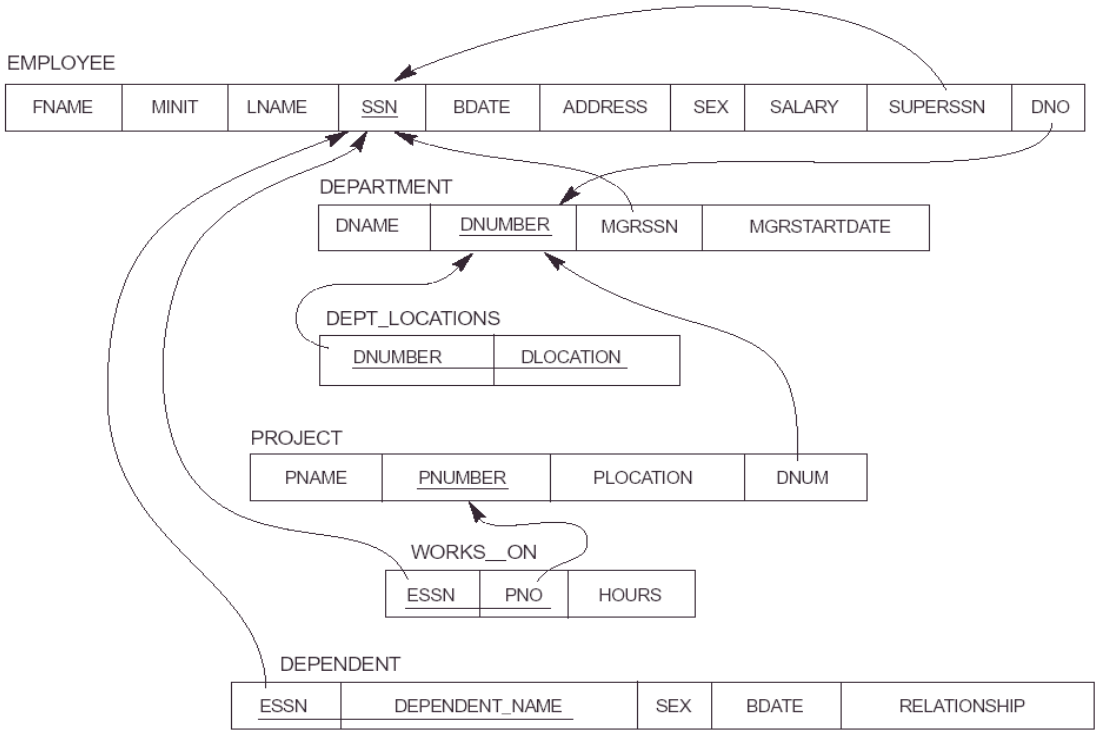
Create the tables related to the following ERD.



First Table: Bank(Code, Name, Addr)
Second Table: Bank-Branch(BranchNo,Code,Addr)
Third Table: Loan(LoanNo, Amount, Type, BranchNo,Code)
Fourth Table: Account(AcctNo, Balance, Type, BranchNo, Code)
Fifth Table: Customer(SSN, Name, Addr, Phone)
Sixth Table: A-C(SSN,AcctNo)
Seventh Table: L-C(SSN, LoanNo)

Question #5

Create ERD based on the following tables. The underlines attributes are primary keys. The links are connection between primary keys and foreign keys.



Note: The Relationships (1-M, M-1, M-N, and R6, and R7) are filled in on the HARD COPY. (I had trouble with this one on Microsoft Word), However I did my other ERD on draw.io. (From now on I will use draw.io, much easier) Sorry if this one is a bit messy!