Assignment 1

CS443 – Database management systems

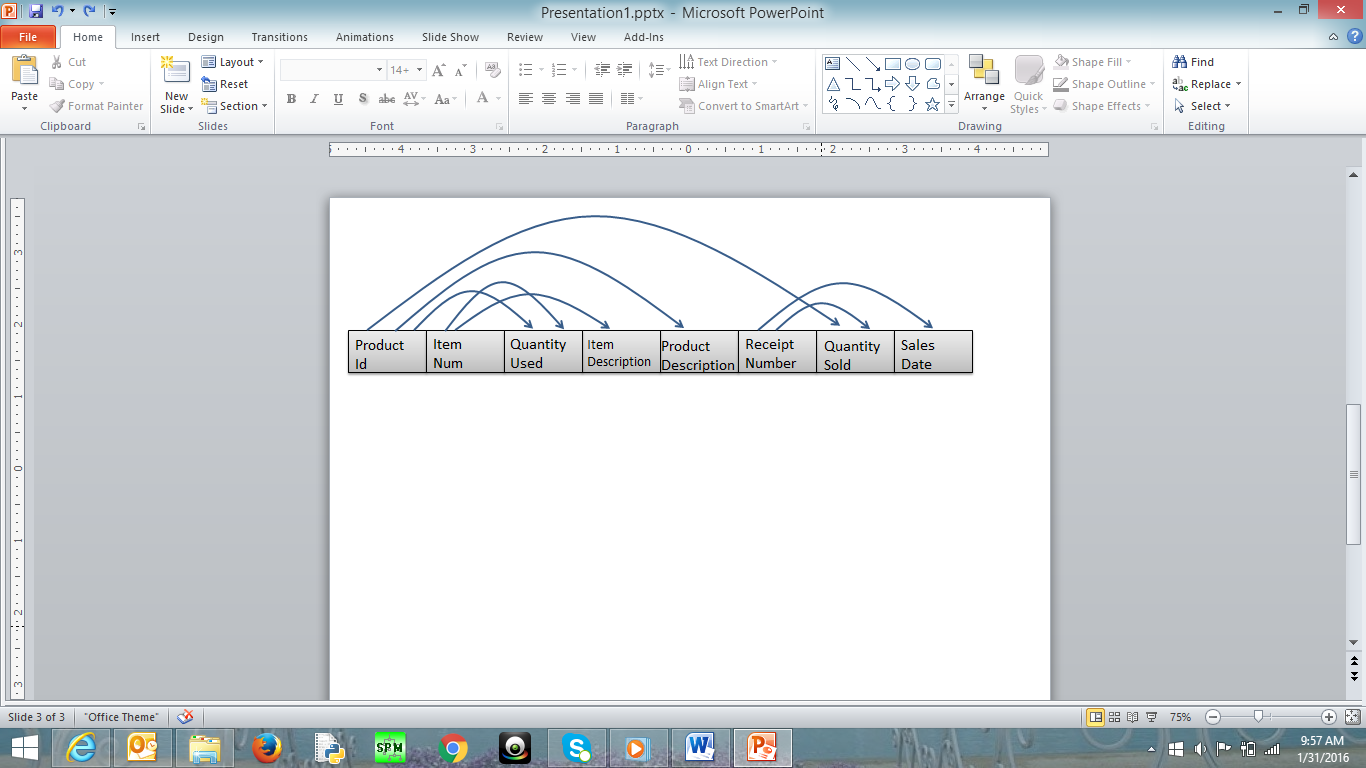
**Parth kapur**

2018

**CS443 - Assignment 1**

**Question #1**

Consider the following data. Arrows show the functional dependency.



The arrows in this question indicated the determination of two attributes. For example, the arrow that goes ProductID to ProductDescription indicates that ProductID determines the ProductDescription. This in turn means that ProductId can be considered as primary key for ProductDescription.

1. Write the tables
2. Place the tables in 3rd normal form (if necessary)
3. Create ERD based on the normalized tables
4. Write a script to create a database. Your script should create the tables and ensures that all constraints are set properly.

Here is some information to create your tables

|  |  |  |
| --- | --- | --- |
| Data Item (Column Name) | Type | Restriction |
| ProductId | Numeric – Integer |  |
| ItemNum | Numeric – Integer | Not null |
| QuantityUsed | Numeric – Integer | >= 0 |
| ItemDescription | Character – Up to 200 |  |
| ProductDescription | Character – Up to 200 |  |
| ReceiptNumber | Numeric – Integer | Not null |
| QuantitySold | Numeric – Integer | > = 0 |
| SalesDate | Date |  |

1a)

Product(ProductID, ProductDescription)

Sold(ProductID,ReceiptNumber, QuantitySold) NOTE: ProductID & ReceiptNumber are foreign keys that combined make up the primary key of Quantity.

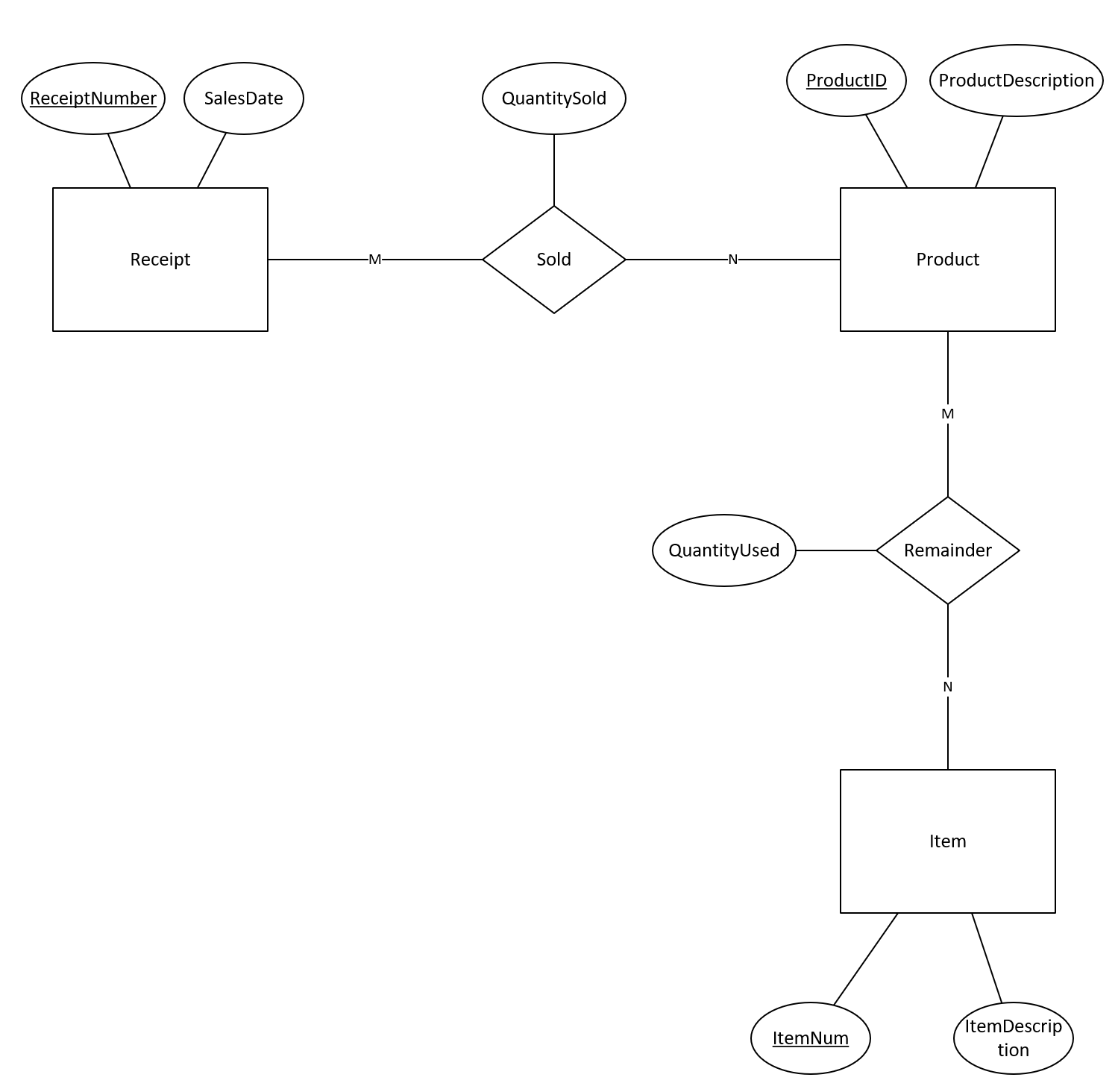
Receipt(ReceiptNumber, SalesDate)

Remainder(ProductID, ItemNum, QuantityUsed) NOTE: ProductID & ItemNum are foreign keys that make up the primary key of RemainingQuantity

Item(ItemNum, ItemDescription)

1b) The tables are already in third normal form as the definition of third normal form is that the table must be in second normal form (which it is) and have no transitive or derived dependencies, which it does not because no non-key attribute can be calculated from other non-key attribute in the same table or in other tables as well as no non-key attribute depends on another non-key attribute in the same table.

1c)



1d)

CREATE TABLE Product

(

ProductID NUMBER,

ProductDescription VARCHAR2(200),

CONSTRAINT Product\_PK PRIMARY KEY(ProductID)

);

CREATE TABLE Sales

(

ReceiptNumber NUMBER,

SalesDate DATE,

CONSTRAINT Sales\_PK PRIMARY KEY(ReceiptNumber)

);

CREATE TABLE Quantity

(

ProductID NUMBER,

ReceiptNumber NUMBER,

QuantitySold NUMBER,

CONSTRAINT Quantity\_PK PRIMARY KEY(ProductID, ReceiptNumber),

CONSTRAINT Quantity\_FK1 FOREIGN KEY(ProductID) REFERENCES Product(ProductID),

CONSTRAINT Quantity\_FK2 FOREIGN KEY(ReceiptNumber) REFERENCES Sales(ReceiptNumber),

CONSTRAINT QuantitySold\_CHECK CHECK(QuantitySold >= 0)

);

CREATE TABLE Item

(

ItemNum NUMBER,

ItemDescription VARCHAR2(200),

CONSTRAINT Item\_PK PRIMARY KEY (ItemNum)

);

CREATE TABLE RemainingQuantity

(

ProductID NUMBER,

ItemNum NUMBER,

QuantityUsed NUMBER,

CONSTRAINT RemainingQuantity\_PK PRIMARY KEY (ProductID, ItemNum),

CONSTRAINT RemainingQuantity\_FK1 FOREIGN KEY (ProductID) REFERENCES Product(ProductID),

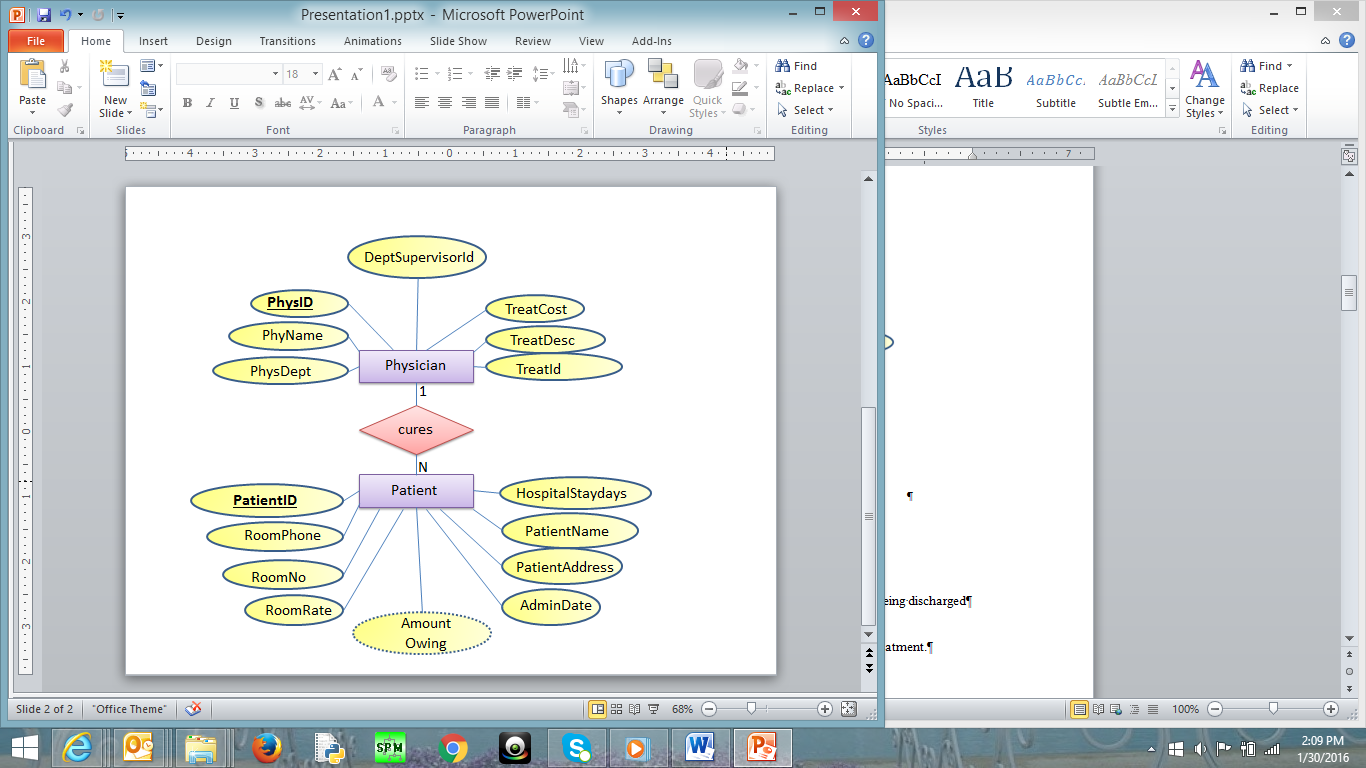
CONSTRAINT RemainingQuantity\_FK2 FOREIGN KEY (ItemNum) REFERENCES Item(ItemNum),

CONSTRAINT QuantityUsed\_CHECK CHECK (QuantityUsed >= 0)

);

**Question #2**

Consider the following ERD



Where

* PatientId: It is the identification number of each patient
* PatientName: It is the name of the patient
* Patientddr: It is the address of the patient
* AdmitDate: It is the date when the patient is admitted to the hospital
* AmounOwing: The amount the patient owes based on his/her sickness after being discharged
* RoomNo: it is the room where the patient is kept in the hospital
* RoomPhone: The phone number in the patient’s room
* HospitalStayDays: Number of days the patient would be in the hospital for treatment.
* RoomRate: The rate charged for every day the patient is in the room

In the second table:

* PhysId: It is the identification number of each physician
* PhyName: It is the name of each physician
* PhysDept: It is the department id where physician works
* DeptSupervisorId: It is the id of the physician who is in change of managing the PhyDept. For example, suppose physician x works in department y. DeptSupervisorId is the id of the physician (not necessarily physician x) who in managing department y.
* TreatId is a number that represents the type of treatment the physician can do
* TreatDesc and TreatCost are Treatment description and treatment cost
  + - Each patient is assigned one doctor, but a doctor can have many patients
    - There may be more than one patient in a room but each patient is kept in one room only
    - Each patient is being treated for one sickness only
    - There is only one phone number in each room in the hospital
    - Each doctor can do only do one treatment, but a treatment can be done by many doctors
    - The treatment cost is fixed for each treatment
    - Each doctor works in only one department, but a department can have many doctors
    - Each department has 1 supervisor. This supervisor is just one of the physicians who works in that department
    - A Patient is charged based on the treatment cost and number of days in hospital

Note that not all the rooms in the hospital has patient at a particular time but all patient must be is some rooms. Further, only some of the physicians are supervising the departments in the hospital; however, all departments must be managed by some physicians.

You may make any other assumption you think is necessary but you have to be very specific and realistic. You can add other assumptions but you are not allowed to change the above assumptions

Do the following

1. Change the ERD to tables
2. Place the tables in 3rd normal form (if necessary)
3. Revise the given ERD based on the normalized tables (if necessary)
4. Write a script to create a database. Your script should create the tables and ensures that all constraints are set properly.

Here is some information to create your tables using SQL. Depending on your normalization process, some of the following fields may not be in your final normalized table.

|  |  |  |
| --- | --- | --- |
| Data Item (Column Name) | Type | restrictions |
| PatientID | Numeric – Integer |  |
| PhysID | Numeric – Integer |  |
| RoomNo | Numeric – Integer | >= 100 and <= 999 |
| AdmitDate | Date |  |
| PatientName | Character -- Up to 50 | Not null |
| PatientAddress | Character -- Up to 200 | Not null |
| RoomPhone | Character -- Up to 8 |  |
| HospitalStayDays | Numeric – Integer | >= 0 |
| RoomRate | Numeric – Decimal  10 with 2 decimals | >= 30.00 and < =100.00 |
| AmountOwing | Numeric – Decimal  10 with 2 decimals |  |
| PhysName | Character -- Up to 50 | Not null |
| PhysDept | Numeric – Integer |  |
| DeptSupervisorId | Numeric – Integer |  |
| TreatId | Numeric – Integer |  |
| TreatDesc | Character -- Up to 200 |  |
| TreatCost | Numeric – Decimal  10 with 2 decimals | >= 50.00 |

2a)

Patient(PatientID, RoomPhone, RoomNo, RoomRate, AmountOwning, AdminDate, PatientAddress, PatientName, HospitalStaydays, PhysID\*)

Physician(PhysID, PhyName, PhysDept, DeptSupervisorID, TreatCost, TreatDesc, TreatID)

2b)

PhysInfo(PhysID, PhyName, PhysDept\*, TreatID\*)

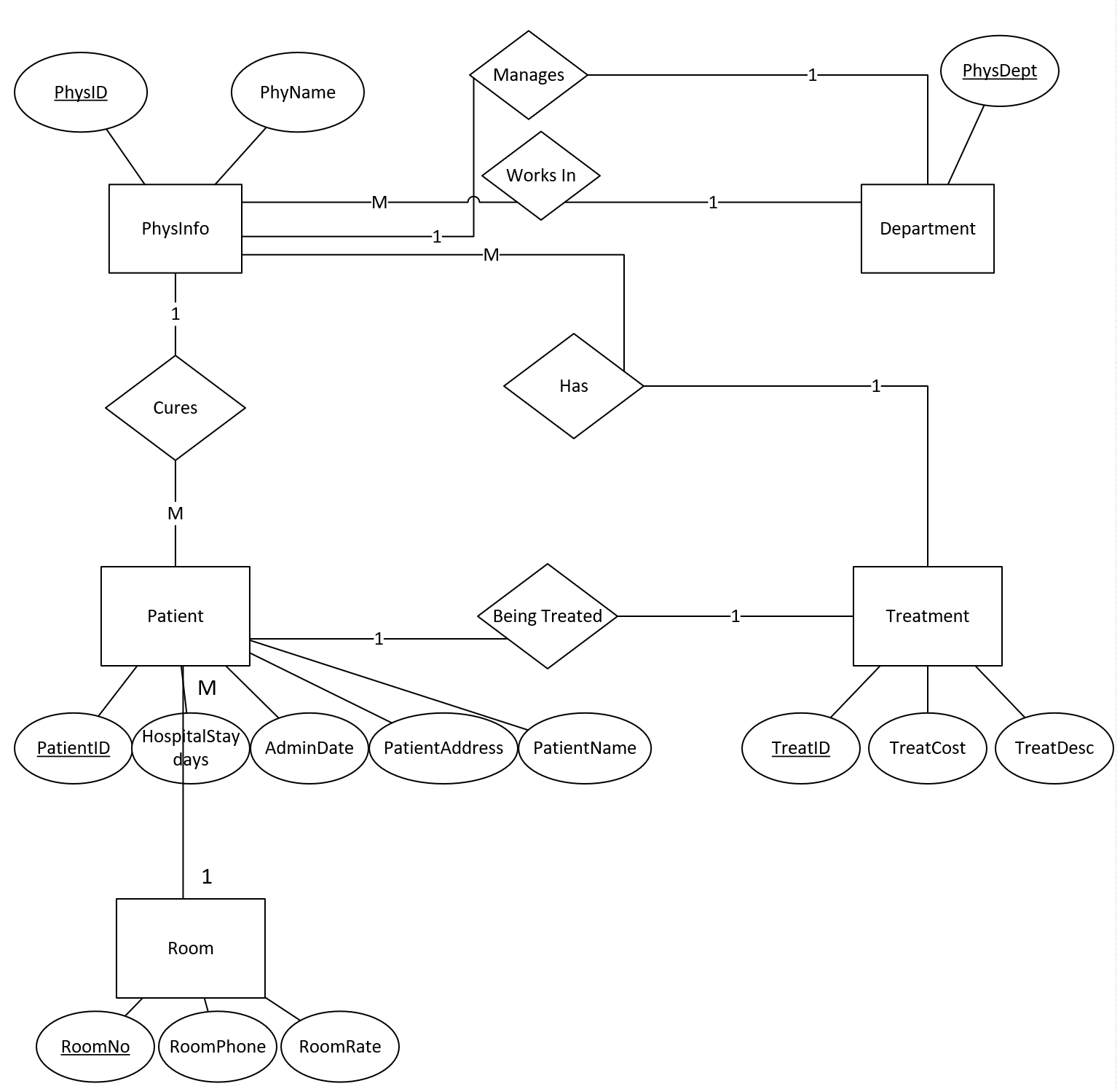
Department(PhysDept, DeptSupervisorID\*)

Treatment(TreatID, TreatCost, TreatDesc, PhysID\*)

Patient(PatientID, AdminDate, PatientAddress, PatientName, HospitalStaydays, PhysID\*, RoomNo\*)

Room(RoomNo, RoomPhone, RoomRate)

2c)



2d)

CREATE TABLE Treatment

(

TreatID NUMBER,

TreatCost NUMBER(9,2),

TreatDesc VARCHAR2(200),

PatientID NUMBER,

CONSTRAINT Treatment\_PK PRIMARY KEY (TreatID),

CONSTRAINT Treatment\_CHECK CHECK (TreatCost >= 50.00)

);

CREATE TABLE Department

(

PhysDept NUMBER,

DeptSupervisorID NUMBER,

CONSTRAINT Department\_PK PRIMARY KEY (PhysDept)

);

CREATE TABLE Room

(

RoomNo NUMBER(3),

RoomPhone NUMBER(9),

RoomRate NUMBER(5,2),

CONSTRAINT Room\_PK PRIMARY KEY (RoomNo),

CONSTRAINT Room\_CHECK CHECK (RoomNo >= 100 AND RoomNo <= 999),

CONSTRAINT Room\_CHECK2 CHECK (RoomRate >= 30.00 AND RoomRate <= 100.00)

);

CREATE TABLE PhysInfo

(

PhysID NUMBER,

PhyName VARCHAR2(25)

CONSTRAINT PhyName\_Null NOT NULL,

PhysDept NUMBER,

TreatID NUMBER,

CONSTRAINT PhysInfo\_PK PRIMARY KEY (PhysID),

CONSTRAINT PhysInfo\_FK1 FOREIGN KEY (PhysDept) REFERENCES Department(PhysDept),

CONSTRAINT PhysInfo\_FK2 FOREIGN KEY (TreatID) REFERENCES Treatment(TreatID)

);

CREATE TABLE Patient

(

PatientID NUMBER,

AdminDate DATE,

PatientAddress VARCHAR2(200)

CONSTRAINT PatientAddress\_Null NOT NULL,

PatientName VARCHAR2(25)

CONSTRAINT PatientName\_Null NOT NULL,

HospitalStayDays NUMBER,

PhysID NUMBER,

RoomNo NUMBER,

CONSTRAINT Patient\_PK PRIMARY KEY (PatientID),

CONSTRAINT Patient\_FK1 FOREIGN KEY (PhysID) REFERENCES PhysInfo(PhysID),

CONSTRAINT Patient\_FK2 FOREIGN KEY (RoomNo) REFERENCES Room(RoomNo)

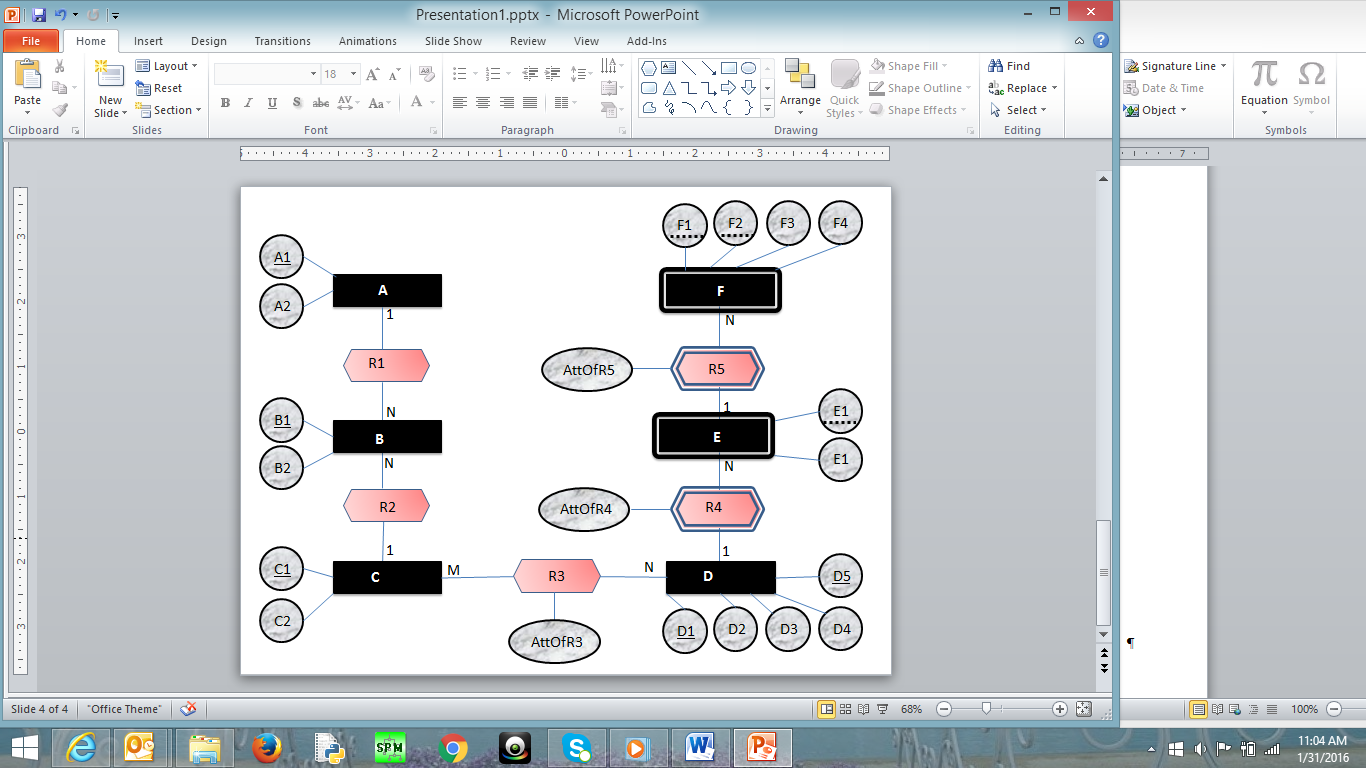
);

ALTER TABLE DEPARTMENT

ADD CONSTRAINT DEPARTMENT\_FK1 FOREIGN KEY(DeptSupervisorID) REFERENCES PhysInfo(PhysID);

**Question #3**

Create the tables related to the following ERD



E2

**The Tables Related to the Following ERD are:**

A = (A1, A2)

B = (B1, B2, A1\*, C1\*)

C = (C1, C2)

D = (D1, D5, D2, D3, D4)

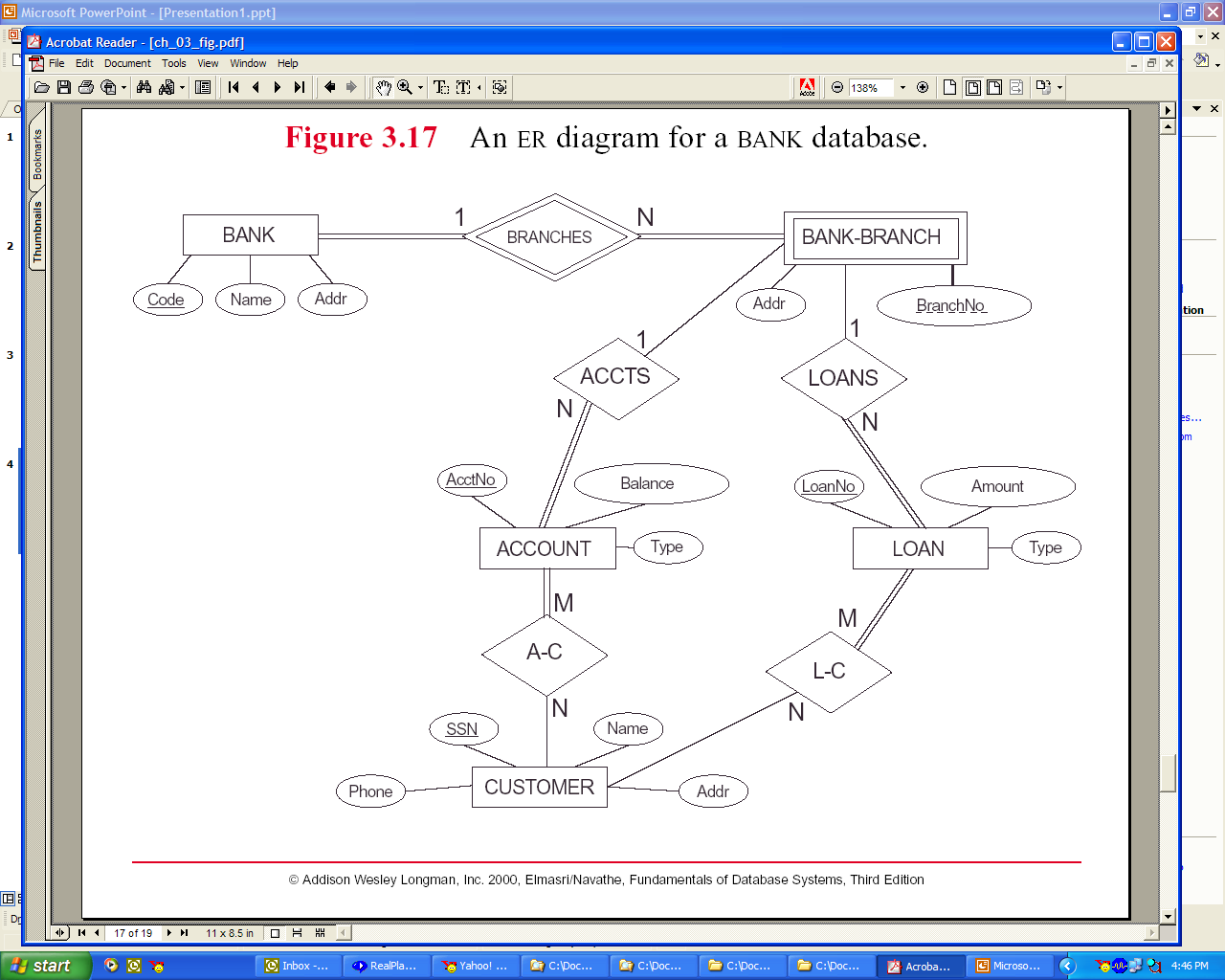
R3 = (C1\*, (D1, D5)\*, Attofr3)

E = (E1, (D1, D5)\*, E2, Attofr4)

f = (F1, f2, (E1, (d1, d5)\*)\*, F3, F4, attofr5)

**Question #4**

Create the tables related to the following ERD



**The tables related to the following ERD are**:

Bank(Code, Name, Addr)

Bank-Branch(BranchNo, Code\*, Addr)

Account(AcctNo, Balance, Type, (BranchNo, Code)\*)

Loan(LoanNo, Amount, Type, (BranchNo, Code)\*)

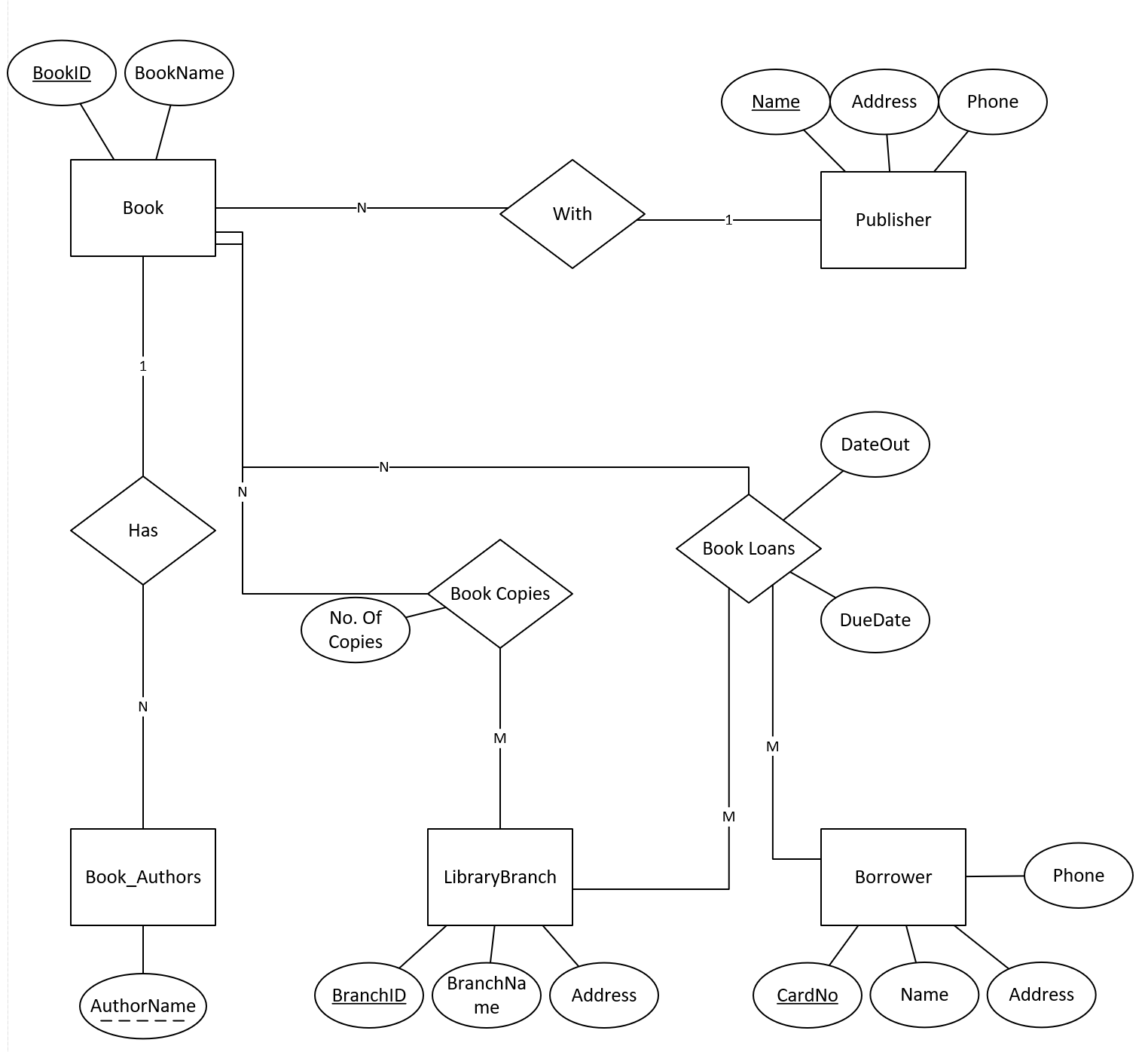
Customer(SSN, Name, Addr, Phone)

A-C(AcctNo\*, SSN\*)

L-C(LoanNo\*, SSN\*)

**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



**Question #6**

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

