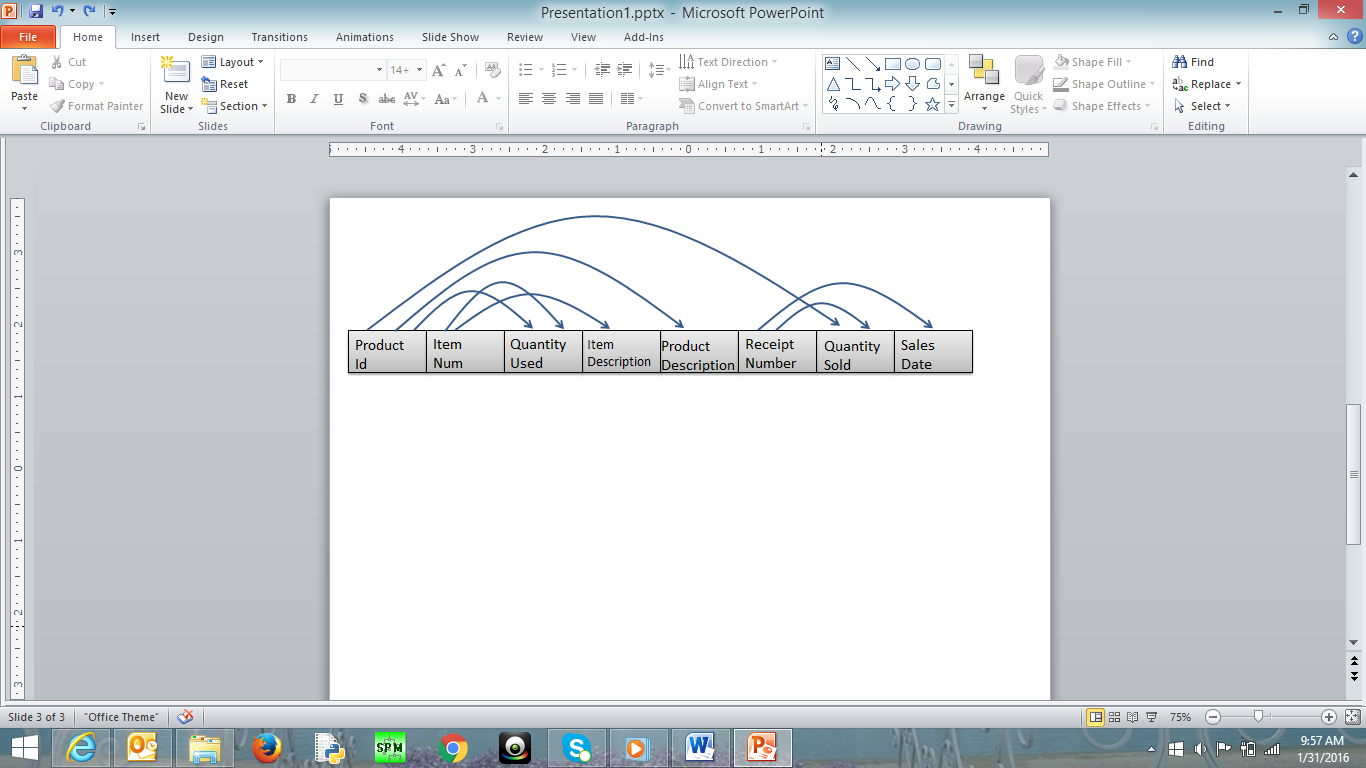
**CS443 - Assignment 1**

**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

Product(ProductID, ProductDescription)

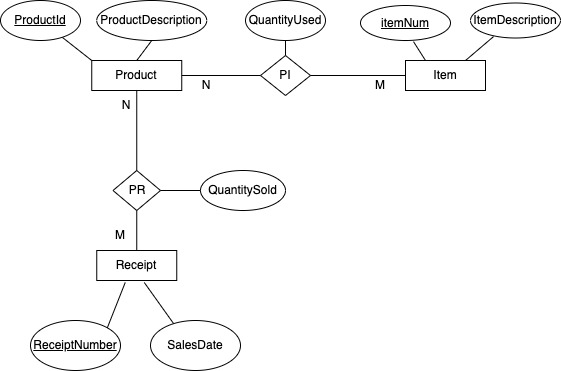
Item(ItemNum, ItemDescription)

PI(ProductID\*, ItemNum\*, QuantityUsed)

Receipt(ReceiptNumber, SalesDate)

PR(ProductID\*, ReceiptNumber\*, QuantitySold)

1. Place the tables in 3rd normal form (if necessary) Not necessary.
2. Create ERD based on the normalized tables



1. 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 |  |

CREATE TABLE Product

( ProductID NUMBER,

  ProductDescription VARCHAR2(200),

  CONSTRAINT ProductPK PRIMARY KEY(ProductID)

);

CREATE TABLE Item

( ItemNum NUMBER,

  ItemDescription VARCHAR(200),

  CONSTRAINT ItemPK PRIMARY KEY(ItemNum)

);

CREATE TABLE PI

( ProductID NUMBER,

  ItemNum NUMBER NOT NULL,

  QuantityUsed NUMBER,

  CONSTRAINT PIPK PRIMARY KEY(ProductID),

  CONSTRAINT PIItemFK FOREIGN KEY(ItemNum) REFERENCES

  Item(ItemNum),

  CONSTRAINT checkQuantityUsed CHECK(QuantityUsed >= 0)

);

CREATE TABLE Receipt

( ReceiptNumber NUMBER,

  SalesDate Date,

  CONSTRAINT ReceiptPK PRIMARY KEY(ReceiptNumber)

);

CREATE TABLE PR

( ProductID NUMBER,

  ReceiptNumber NUMBER NOT NULL,

  QuantitySold NUMBER,

  CONSTRAINT PRPK PRIMARY KEY(ProductID),

  CONSTRAINT ProductIDFK FOREIGN KEY(ProductID) REFERENCES

  Product(ProductID),

  CONSTRAINT ReceiptNumberFK FOREIGN KEY(ReceiptNumber) REFERENCES

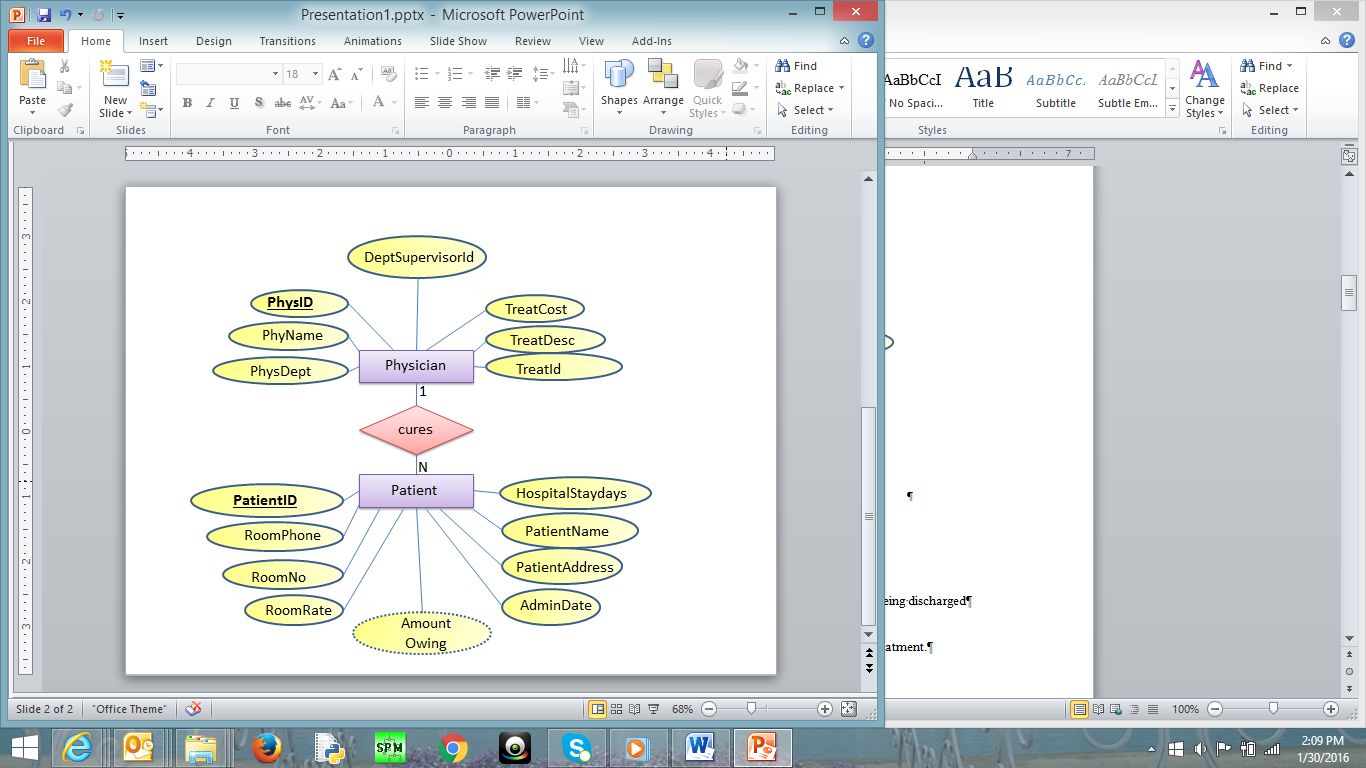
  Receipt(ReceiptNumber),

  CONSTRAINT checkQuantitySold CHECK(QuantitySold >= 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

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

Patient(PatientID, RoomPhone, RoomNo, RoomRate, AmountOwing, AdminRate, PatientAddress, PatientName, HospitalStayDays, PhysID\*)

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

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

Treatment(TreatID, TreatDesc, TreatCost)

Department(PhysDept, DeptSupervisorID\*)

Room(RoomNo, RoomPhone, RoomRate)

Patient(PatientID, PatientName, PatientAddress, AdmitDate, PhysID\*, RoomNo\*)

1. Revise the given ERD based on the normalized tables (if necessary)

Diagram

Description automatically generated

1. 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 |

CREATE TABLE Treatment

( TreatID NUMBER,

TreatDesc VARCHAR2(200),

TreatCost NUMBER(10,2),

CONSTRAINT TreatmentPK PRIMARY KEY(TreatID),

CONSTRAINT checkTreatCost CHECK(TreatCost >= 50.00)

);

CREATE TABLE Room

( RoomNo NUMBER CHECK(RoomNo >= 100 AND RoomNo <= 999),

RoomPhone VARCHAR2(8),

RoomRate NUMBER(10,2),

CONSTRAINT RoomPK PRIMARY KEY(RoomNo),

CONSTRAINT checkRoomRate CHECK(RoomRate >= 30.00 AND RoomRate <= 100.00)

);

CREATE TABLE Physician

( PhysID NUMBER,

PhysName VARCHAR2(50) NOT NULL,

PhysDept NUMBER,

TreatID NUMBER,

CONSTRAINT PhysicianPK PRIMARY KEY(PhysID),

CONSTRAINT PhysicianTreatmentFK FOREIGN KEY(TreatID) REFERENCES Treatment(TreatID)

);

CREATE TABLE Department

( PhysDept NUMBER,

DeptSupervisorID NUMBER,

CONSTRAINT DepartmentSupervisorFK FOREIGN KEY(DeptSupervisorID) REFERENCES Physician(PhysID),

CONSTRAINT DepartmentPK PRIMARY KEY(PhysDept)

);

ALTER TABLE Physician ADD CONSTRAINT PhysicianDeptFK FOREIGN KEY(PhysDept) REFERENCES Department(PhysDept);

CREATE TABLE Patient

( PatientID NUMBER,

PatientName VARCHAR2(50) NOT NULL,

PatientAddress VARCHAR2(200) NOT NULL,

AdmitDate DATE,

PhysID NUMBER,

RoomNo NUMBER,

CONSTRAINT PatientPK PRIMARY KEY(PatientID),

CONSTRAINT PatientPhysIDFK FOREIGN KEY(PhysID) REFERENCES Physician(PhysID),

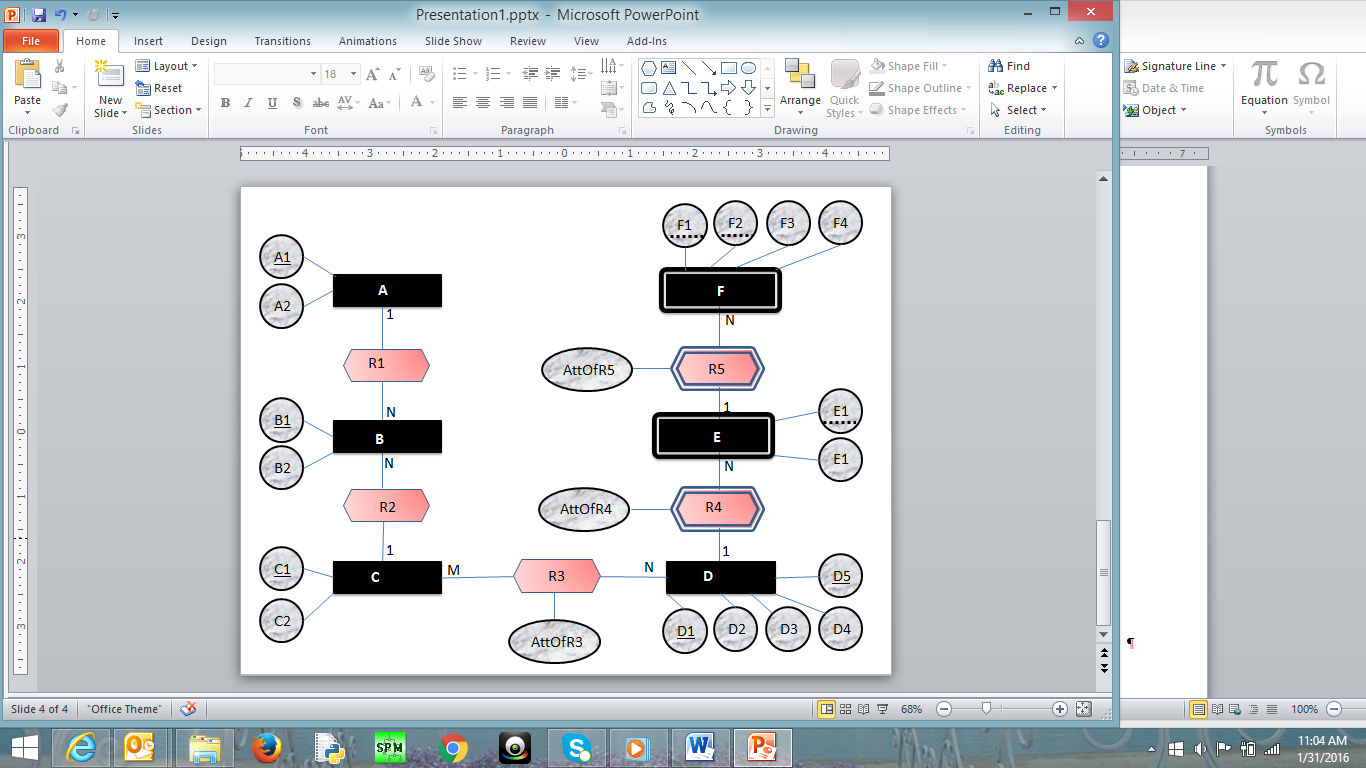
CONSTRAINT PatientRoomNoFK FOREIGN KEY(RoomNo) REFERENCES Room(RoomNo),

CONSTRAINT checkRoomNo CHECK(RoomNo >= 100 AND RoomNo <= 999)

);

**Question #3**

Create the tables related to the following ERD. Determine the primary Keys and the foreign keys of each table.



E2

E2

A(A1, A2)

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

C(C1, C2)

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

E([D1, D5]\*, E1, E2, AttOfR4)

F([D1, D5, E1]\*, F1, F2, F3, F4, AttOfR5)

R3(C1\*, [D1, D5]\*, AttOfR3)

**Question #4**

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

Timeline

Description automatically generated



**Diagram

Description automatically generated**

**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

Diagram

Description automatically generated

Diagram

Description automatically generated