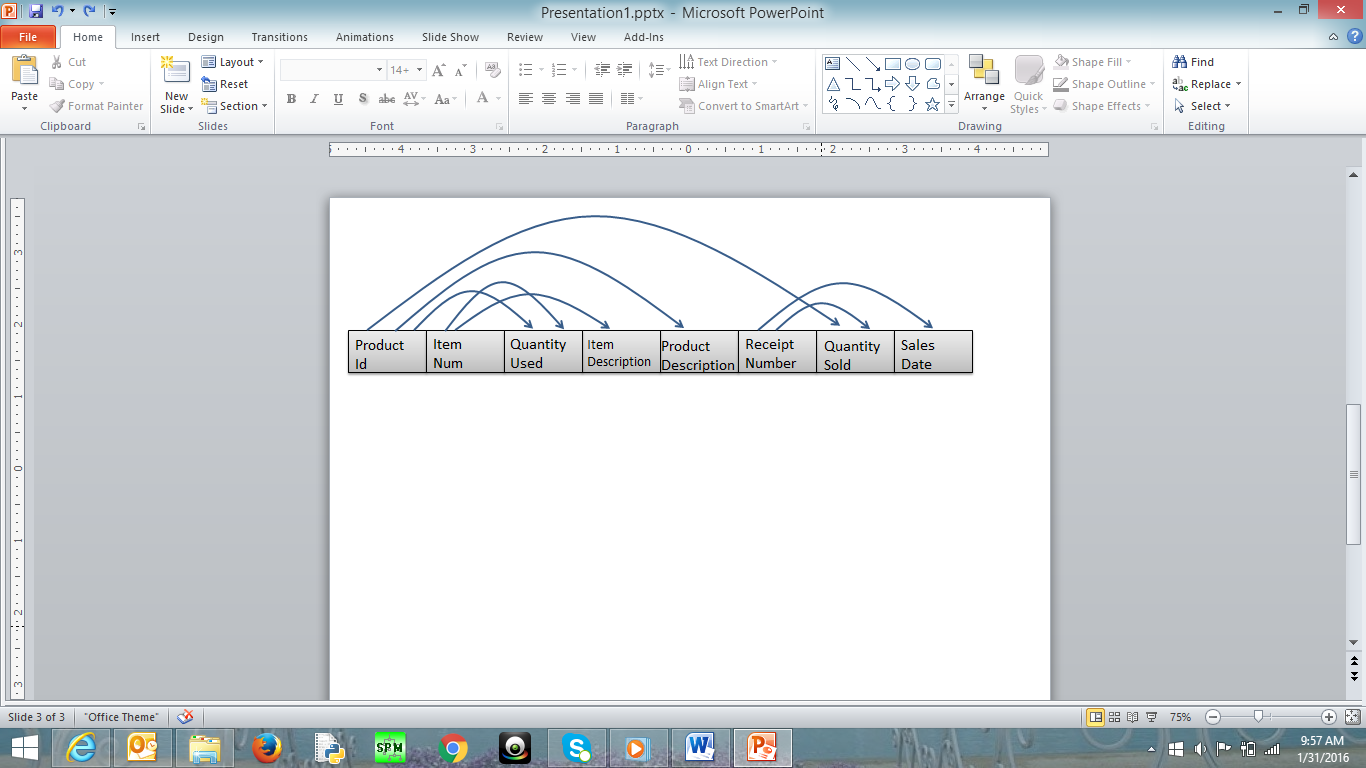
**CS443 - Assignment 1**

**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

(T1) Product: ProductId, ProductDescription

(T2) Item: ItemNum, ItemDescription

(T3) Receipt: ReciptNum, SalesDate

(T4) ProdItem: ProductId, ItemNum, QuantityUsed

(T5) ProdRec: ProductId, ReciptNum, QuantitySold

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

\*\*It was not necessary to set the tables into normal form since there aren’t any transitive nor derived dependencies to remove.

**Product**(ProductId, ProductDescription)

**Item**(ItemNum, ItemDescription)

**Receipt**(ReciptNum, SalesDate)

**ProdItem(**ProductId, ItemNum, QuantityUsed**)**

**ProdRec(**ProductId, ReciptNum, QuantitySold**)**

1. Create ERD based on the normalized tables Diagram

   Description automatically generated
2. 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 Item

(

ItemNum NUMBER CONSTRAINT

ItemNum\_Null NOT NULL,

ItemDescription VARCHAR2(200),

CONSTRAINT Item\_PK PRIMARY KEY(ItemNum)

);

CREATE TABLE Product

(

ProductId NUMBER,

ProductDescription VARCHAR2(200),

CONSTRAINT Product\_PK PRIMARY KEY(ProductId)

);

CREATE TABLE Receipt

(

SalesDate DATE,

ReciptNumber NUMBER CONSTRAINT

ReciptNumber\_Null NOT NULL,

CONSTRAINT Receipt\_PK PRIMARY KEY(ReciptNumber)

);

CREATE TABLE ProdItem

(

ProductId NUMBER,

ItemNum NUMBER CONSTRAINT ItemNum\_Null

NOT NULL,

QuantityUsed NUMBER,

CONSTRAINT ProdItem\_PK PRIMARY KEY(ProductId, ItemNum),

CONSTRAINT ProdItem\_FK1 FOREIGN KEY(ProductId) REFERENCES

Product(ProductId),

CONSTRAINT ProdItem\_FK2 FOREIGN KEY(ItemNum) REFERENCES

Item(ItemNum),

CONSTRAINT QuantityUsed\_CK CHECK(QuantityUsed >= 0)

);

CREATE TABLE ProdRec

(

ProductId NUMBER,

ReciptNumber NUMBER CONSTRAINT

ReciptNumber\_Null NOT NULL,

QuantitySold NUMBER,

CONSTRAINT ProdRec\_PK PRIMARY KEY(ProductId,

ReciptNumber),

CONSTRAINT ProdRec\_FK1 FOREIGN KEY(ProductId) REFERENCES

Product(ProductId),

CONSTRAINT ProdRec\_FK2 FOREIGN KEY(ReciptNumber) REFERENCES Receipt(ReciptNumber),

CONSTRAINT QuantitySold\_CK CHECK (QuantitySold >= 0)

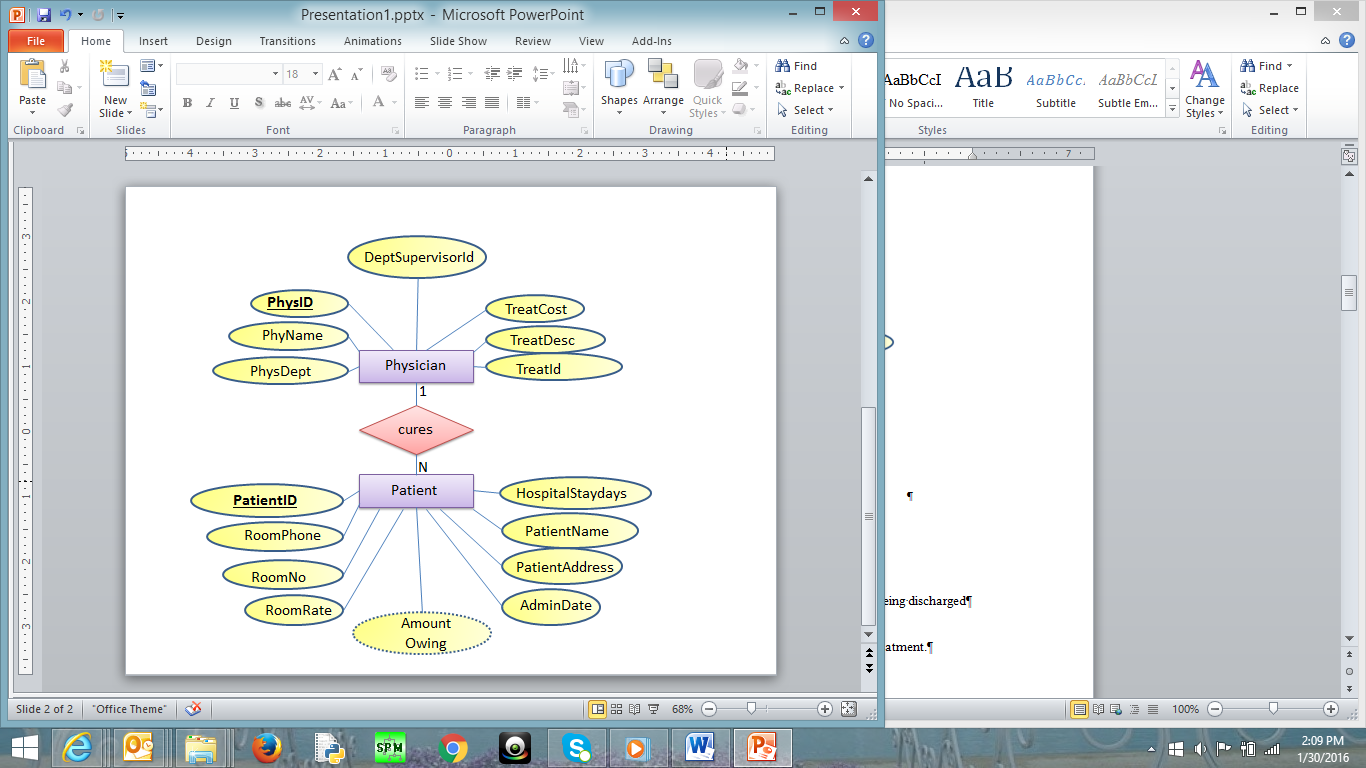
);

Graphical user interface, application

Description automatically generated

**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, PhysName, PhysDept, DeptSupervisorID, TreatCost, TreatDesc, TreatID**)**

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

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

\*\*There are derived and transitive dependencies.

In the Physician table, the following transitive dependencies include the department and treatment, so in contrast, it can be split into 3 tables being Physician, Department, and Treatment.

In the Patient table, the following included both transitive/derived dependencies. With that in mind, it can be split into 2 tables being Patient and Room. AmountOwing would be derived in this case since the following attributes from the Patient table consist of RoomRate, HospitalStaydays, and TreatCost (RoomRate \* HospitalStaydays + TreatCost.)

**Physician (**PhysID, PhysName, PhysDept\*, TreatId\***)**

**Department (**PhysDept, DeptSupervisorID\***)**

**Treatment (**TreatId, TreatDesc, TreatCost**)**

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

**Room (**RoomNo, RoomPhone, RoomRate**)**

\*Note: DeptSupervisorID and PhysID are the same, which is why DeptSupervisorID is a foreign key inside of the Department table.

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

   Description automatically generated
2. Write a script to create a database. Your script should create the tables and ensures that all constraints are set properly.

CREATE TABLE Physician(

PhysID NUMBER,

PhysName VARCHAR2(50) CONSTRAINT

PhysName\_Null NOT NULL,

PhysDept NUMBER,

TreatId NUMBER,

CONSTRAINT Physician\_PK PRIMARY KEY(PhysID),

CONSTRAINT Physician\_FK1 FOREIGN KEY(PhysDept)

REFERENCES

Department(PhysDept),

CONSTRAINT Physician\_FK2 FOREIGN KEY(TreatId)

REFERENCES Treatment(TreatId)

);

CREATE TABLE Department(

PhysDept NUMBER,

DeptSupervisorID NUMBER,

CONSTRAINT Department\_PK PRIMARY KEY(PhysDept)

);

CREATE TABLE Treatment(

TreatId NUMBER,

TreatDesc VARCHAR2(200),

TreatCost NUMBER(10, 2),

CONSTRAINT Treatment\_PK PRIMARY KEY(TreatId),

CONSTRAINT TreatCost\_CK CHECK(TreatCost >= 50.00)

);

CREATE TABLE Patient(

PatientId NUMBER,

PatientName VARCHAR2(50) CONSTRAINT

PatientName\_Null NOT NULL,

PatientAddress VARCHAR2(200) CONSTRAINT

PatientAddress\_Null NOT NULL,

AdminDate DATE,

HospitalStaydays NUMBER,

RoomNo NUMBER,

PhysID NUMBER,

CONSTRAINT Patient\_PK PRIMARY KEY(PatientId),

CONSTRAINT Patient\_FK1 FOREIGN KEY(RoomNo)

REFERENCES Room(RoomNo),

CONSTRAINT Patient\_FK2 FOREIGN KEY(PhysID)

REFERENCES Physician(PhysID),

CONSTRAINT HospitalStaydays\_CK CHECK(HospitalStaydays >= 0)

);

CREATE TABLE Room(

RoomNo NUMBER,

RoomPhone VARCHAR2(8),

RoomRate NUMBER(10, 2),

CONSTRAINT Room\_PK PRIMARY KEY(RoomNo),

CONSTRAINT RoomNo\_CK CHECK(RoomNo >= 100 AND

RoomNo <= 999),

CONSTRAINT RoomRate\_CK CHECK(RoomRate >= 30.00 AND

RoomRate <= 100.00)

);

ALTER TABLE Department

ADD CONSTRAINT Department\_FK1 FOREIGN KEY(DeptSupervisorID) REFERENCES Physician(PhysID);

**SCREENSHOTS:**

Text

Description automatically generated









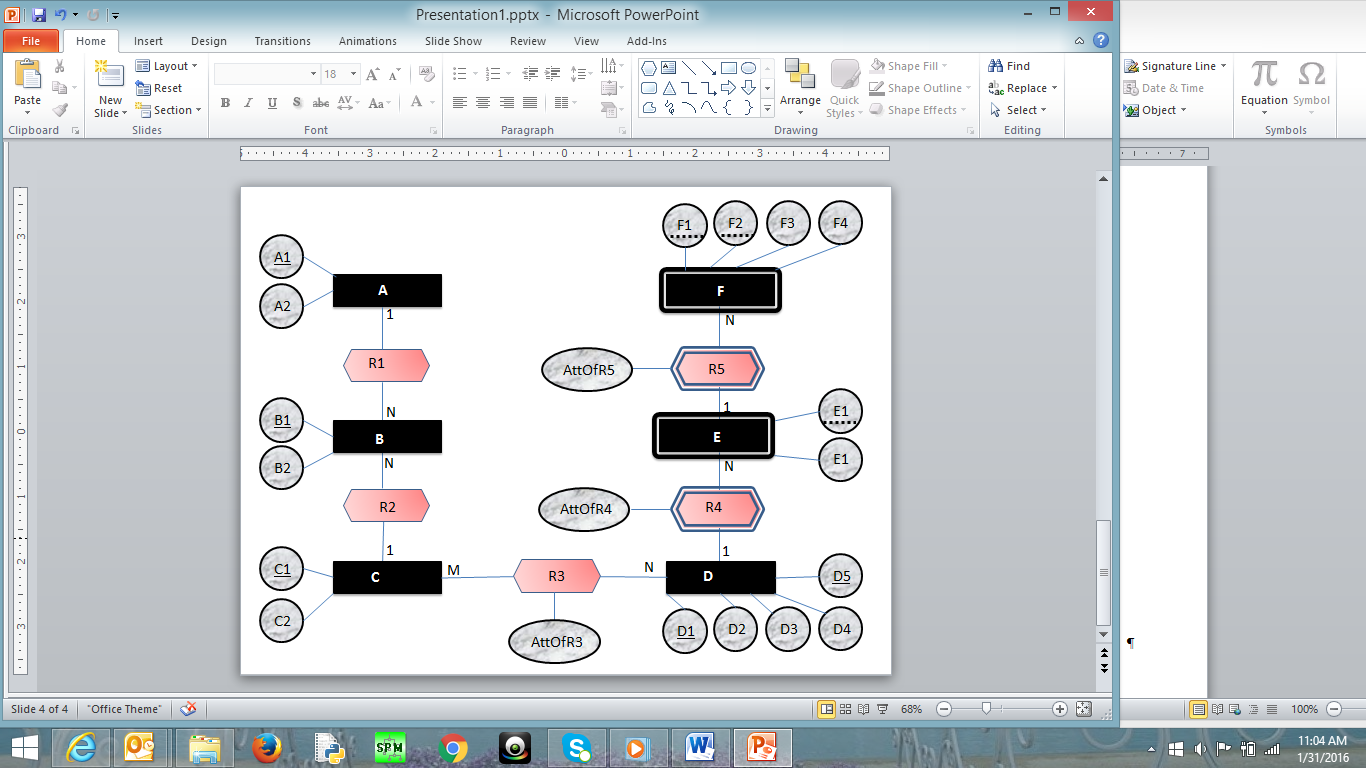


Here is some information to create your tables using SQL. Depending on your normalization process, some of the following fields may not be in your final normalized table.

|  |  |  |
| --- | --- | --- |
| 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 |

**Question #3**

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



E2

A (A1, A2)

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

C (C1, C2)

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

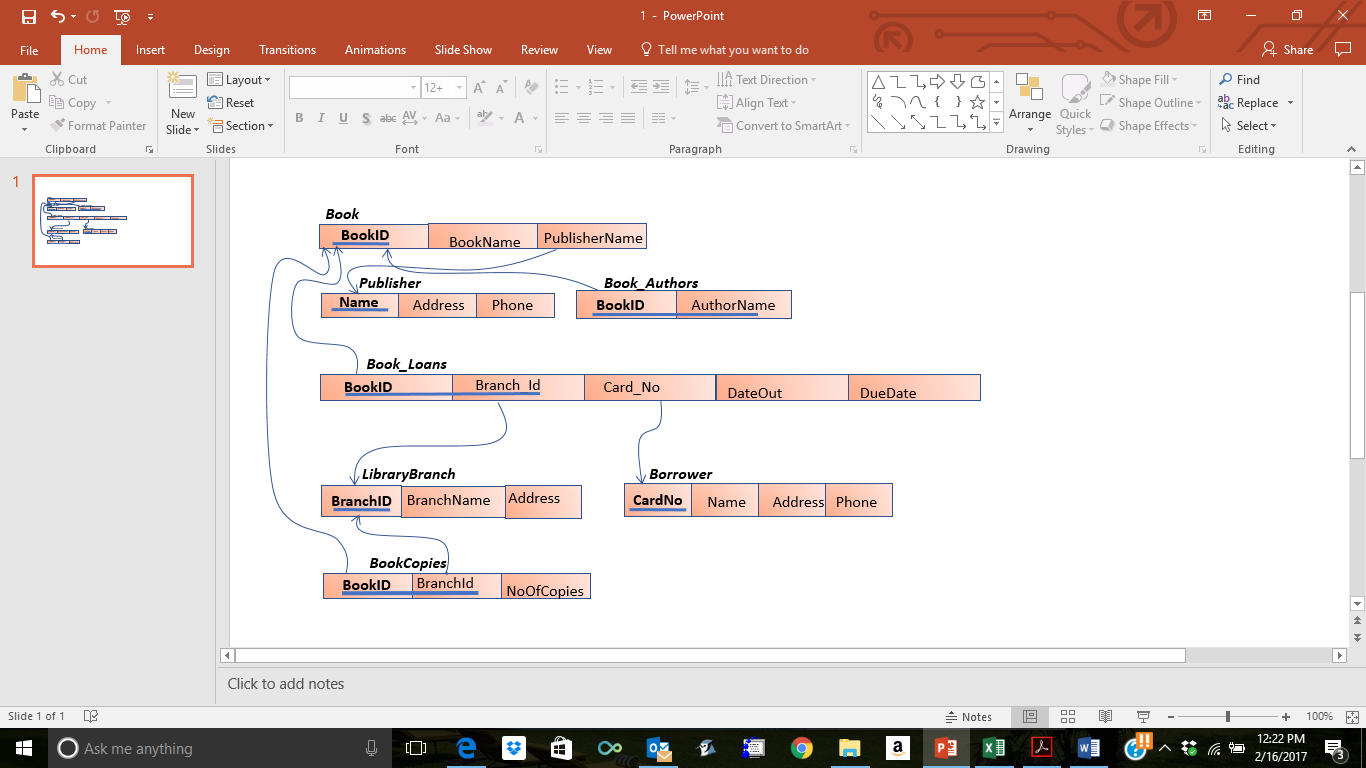
D (D1, D5, D2, D3, D4)

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

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

**Question #4**

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

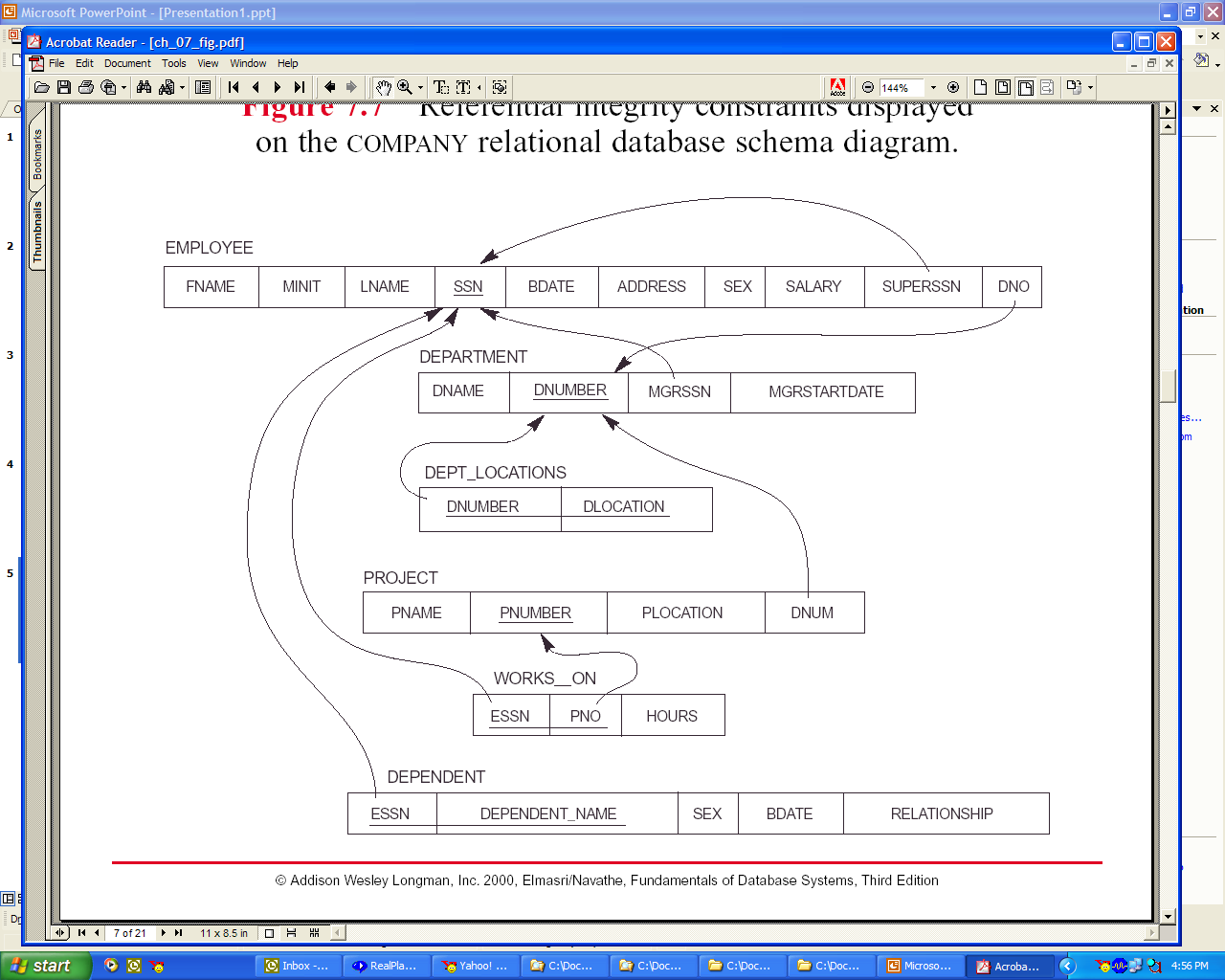


**Diagram

Description automatically generated**

**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