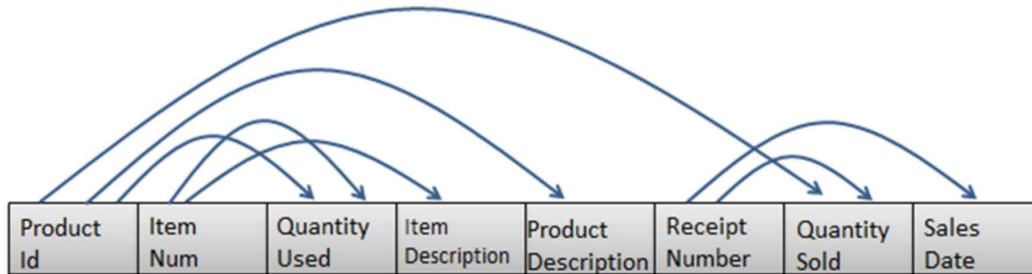


CS443 - Assignment 1

Shane Dyrdaahl
3/7/2024

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.

- Write the tables
- Place the tables in 3rd normal form (if necessary)
- Create ERD based on the normalized tables
- 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	

- a) Write the tables:

Product (**ProductID**, ProductDescription)
 Receipt (**ReceiptNumber**, SalesDate)
 Item (**ItemNum**, ItemDescription)
 Invoice (**ProductID***, **ReceiptNumber***, QuantitySold)
 Goods (**ProductID***, **ItemNum***, QuantityUsed)

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

3rd Normal Form:

- Derived: none
- Transitive: none

No derived or transitive dependencies, therefore the tables are already in 3rd Normal Form.

c) Create ERD based on the normalized tables

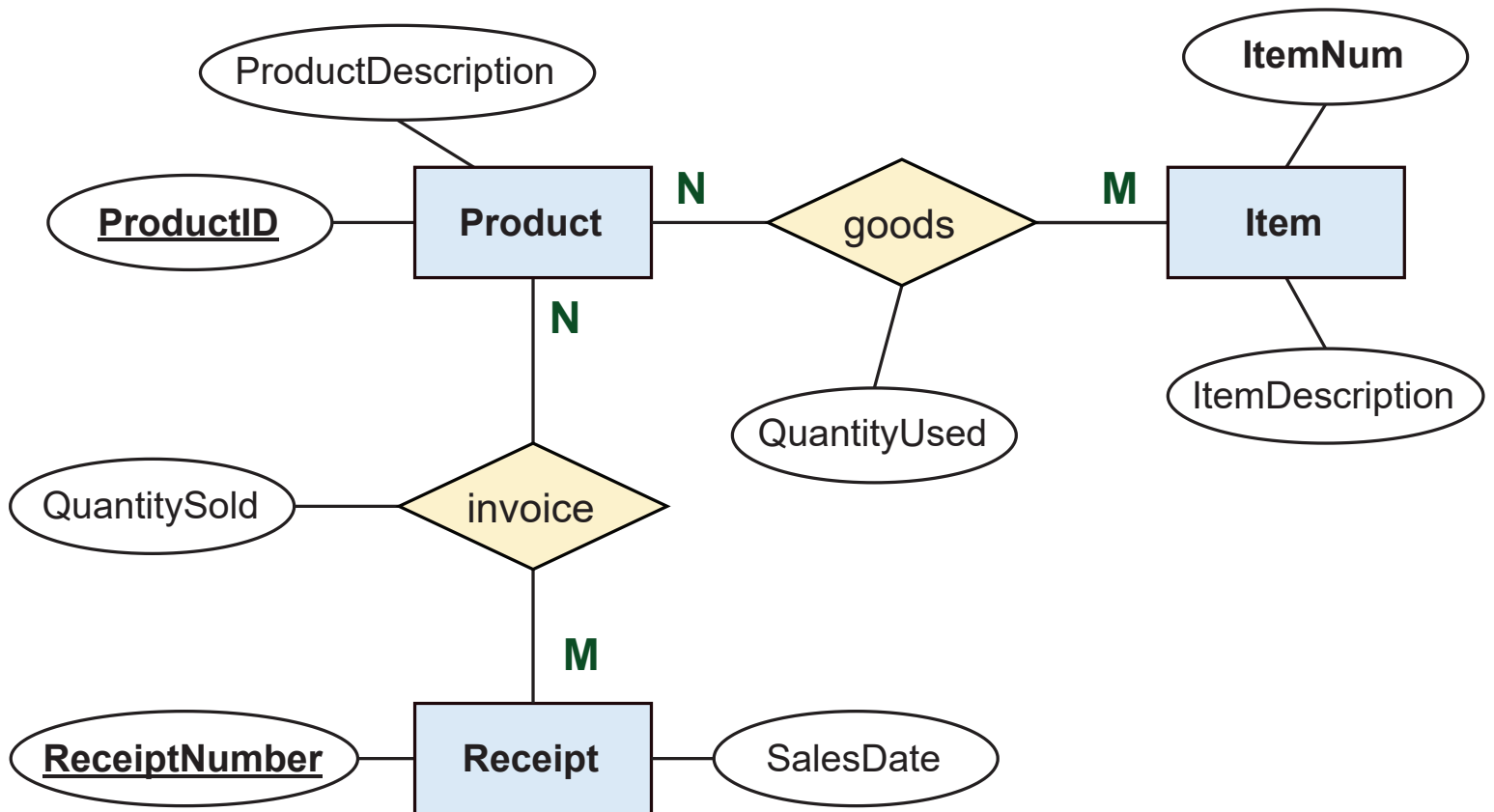
Product (ProductID, ProductDescription)

Receipt (ReceiptNumber, SalesDate)

Item (ItemNum, ItemDescription)

Invoice (ProductID*, ReceiptNumber*, QuantitySold)

Goods (ProductID*, ItemNum*, QuantityUsed)



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

```
CREATE TABLE Product
(
    ProductID          NUMBER,
    ProductDescription  VARCHAR2(200),
    CONSTRAINT ProductPK PRIMARY KEY (ProductID)
);

CREATE TABLE Item
(
    ItemNum            NUMBER NOT NULL,
    ItemDescription    VARCHAR2(200),
    CONSTRAINT ItemPK PRIMARY KEY (ItemNum)
);

CREATE TABLE Receipt
(
    ReceiptNumber      NUMBER NOT NULL,
    SalesDate          DATE,
    CONSTRAINT ReceiptPK PRIMARY KEY (ReceiptNumber)
);

CREATE TABLE Goods
(
    QuantityUsed       NUMBER,
    ProductID          NUMBER,
    ItemNum            NUMBER NOT NULL,

    CONSTRAINT CheckUsed CHECK ( QuantityUsed >= 0 ),
    CONSTRAINT GoodsPK PRIMARY KEY (ProductID, ItemNum),
    CONSTRAINT GoodsFK1 FOREIGN KEY (ProductID)
        REFERENCES Product(ProductID),
    CONSTRAINT GoodsFK2 FOREIGN KEY (ItemNum)
        REFERENCES Item(ItemNum)
);

CREATE TABLE Invoice
(
    QuantitySold       NUMBER,
    ProductID          NUMBER,
    ReceiptNumber      NUMBER NOT NULL,

    CONSTRAINT CheckSold CHECK ( QuantitySold >= 0 ),
    CONSTRAINT InvoicePK PRIMARY KEY (ProductID, ReceiptNumber),
    CONSTRAINT InvoiceFK1 FOREIGN KEY (ProductID)
        REFERENCES Product(ProductID),
    CONSTRAINT InvoiceFK2 FOREIGN KEY (ReceiptNumber)
        REFERENCES Receipt(ReceiptNumber)
);
```

Connections

Oracle Connections

TigerConnection

- Tables (Filtered)
 - GOODS
 - INVOICE
 - ITEM
 - PRODUCT
 - RECEIPT
- Views
- Indexes
- Packages
- Procedures
- Functions
- Operators
- Queues
- Queues Tables
- Triggers
- Types
- Sequences
- Materialized Views
- Materialized View Logs
- Synonyms
- Public Synonyms
- Database Links
- Public Database Links
- Directories
- Editions
- Java
- XML Schemas
- XML DB Repository
- OLAP Option
- Analytic Views
- Scheduler

Reports

All Reports

- Analytic View Reports
- Data Dictionary Reports
- Data Modeler Reports
- OLAP Reports
- TimesTen Reports
- User Defined Reports

Welcome Page | TigerConnection

0.116 seconds

Worksheet | Query Builder

```
SalesDate          DATE,
CONSTRAINT ReceiptPK PRIMARY KEY (ReceiptNumber)
);

CREATE TABLE Goods
(
    QuantityUsed      NUMBER,
    ProductID          NUMBER,
    ItemNum            NUMBER NOT NULL,

    CONSTRAINT CheckUsed CHECK ( QuantityUsed >= 0 ),
    CONSTRAINT GoodsPK PRIMARY KEY (ProductID, ItemNum),
    CONSTRAINT GoodsFK1 FOREIGN KEY (ProductID)
        REFERENCES Product(ProductID),
    CONSTRAINT GoodsFK2 FOREIGN KEY (ItemNum)
        REFERENCES Item(ItemNum)
);

CREATE TABLE Invoice
(
    QuantitySold       NUMBER,
    ProductID          NUMBER,
    ReceiptNumber      NUMBER NOT NULL,

    CONSTRAINT CheckSold CHECK ( QuantitySold >= 0 ),
```

Script Output x

Task completed in 0.116 seconds

Table PRODUCT created.

Table ITEM created.

Table RECEIPT created.

Table GOODS created.

Table INVOICE created.

Question #2

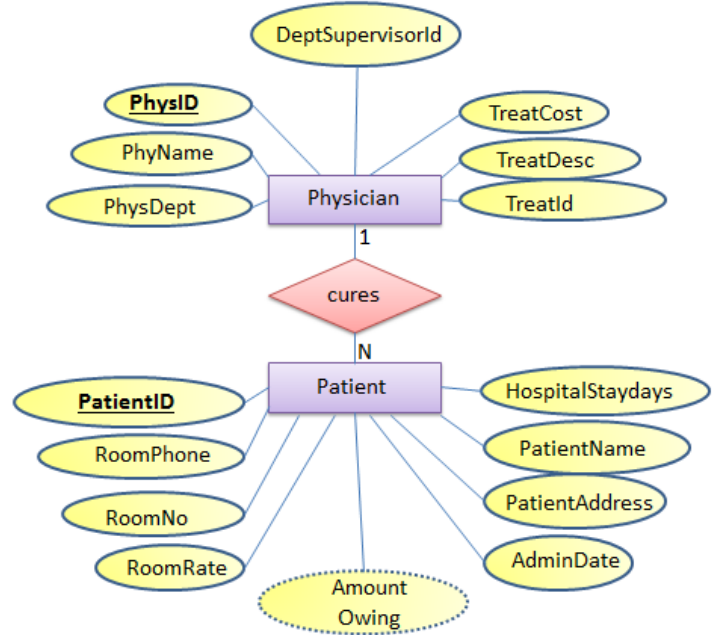
Consider the following ERD

Where

- PatientId: It is the identification number of each patient
- PatientName: It is the name of the patient
- PatientAddr: It is the address of the patient
- AdmitDate: It is the date when the patient is admitted to the hospital
- AmountOwing: 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 charge 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 is 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 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 in 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

- Change the ERD to tables
- Place the tables in 3rd normal form (if necessary)
- Revise the given ERD based on the normalized tables (if necessary)
- 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

2)

a) Write the tables

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

Patient(**PatientID**, RoomPhone, RoomNo, RoomRate, AmountOwing, HospitalStayDays, PatientName, PatientAddress, AdminDate)

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

Dependencies:

- Derived:

- AmountOwing can be calculated using: $(\text{RoomRate} * \text{HospitalStayDays}) + \text{TreatCost}$

- Transitive: (Both tables have transitive dependencies.)

- We can create two new tables from the Physician table.

- By extracting TreatID, TreatCost, and TreatDesc, we can create a Treatment table

- Extracting PhyDept and DeptSupervisorID allows for a Department table.

- For the Patient table, we can create one extra table.

- We can create a Room table by extracting RoomNo, RoomPhone, and RoomRate

Normalized Tables:

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

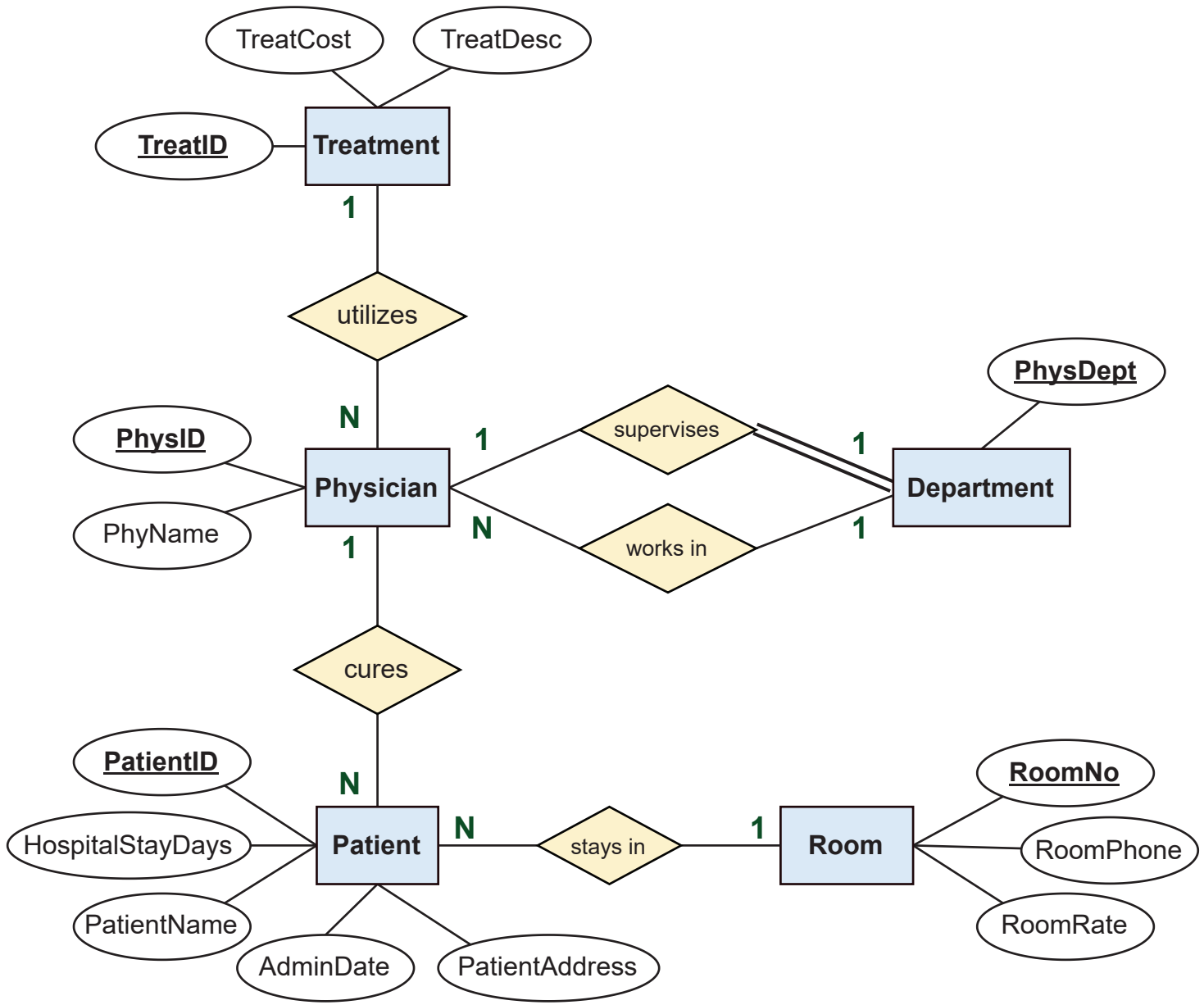
Treatment(**TreatId**, TreatCost, TreatDesc)

Department(**PhysDept**, DeptSupervisorID*)

Patient(**PatientID**, HospitalStayDays, PatientName, AdminDate, PatientAddress, PhysID*, RoomNo*)

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

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 Department
(
    PhysDept          NUMBER,
    DeptSupervisorID  NUMBER,
    CONSTRAINT DepartmentPK PRIMARY KEY (PhysDept)
);
CREATE TABLE Treatment
(
    TreatId           NUMBER,
    TreatCost         NUMBER(10, 2),
    TreatDesc         VARCHAR2(200),
    CONSTRAINT CheckTreatCost CHECK ( TreatCost >= 50.00 ),
    CONSTRAINT TreatmentPK PRIMARY KEY (TreatId)
);
CREATE TABLE Room
(
    RoomNo            NUMBER,
    RoomPhone         VARCHAR2(8),
    RoomRate          NUMBER(10, 2),
    CONSTRAINT CheckRoomNo CHECK ( RoomNo >= 100 AND RoomNo <= 999 ),
    CONSTRAINT CheckRoomRate CHECK ( RoomRate >= 30.00 AND RoomRate <= 100.00 ),
    CONSTRAINT RoomPK PRIMARY KEY (RoomNo)
);
CREATE TABLE Physician
(
    PhysID            NUMBER,
    PhyName           VARCHAR2(50) NOT NULL,
    PhysDept          NUMBER,
    TreatID           NUMBER,
    CONSTRAINT PhysicianPK PRIMARY KEY (PhysID),
    CONSTRAINT PhysicianFK1 FOREIGN KEY (PhysDept)
        REFERENCES Department(PhysDept),
    CONSTRAINT PhysicianFK2 FOREIGN KEY (TreatID)
        REFERENCES Treatment(TreatID)
);
CREATE TABLE Patient
(
    PatientID         NUMBER,
    HospitalStayDays  NUMBER,
    PatientName       VARCHAR2(50) NOT NULL,
    AdmitDate         DATE,
    PatientAddress    VARCHAR2(200) NOT NULL,
    PhysID            NUMBER,
    RoomNo            NUMBER,
    CONSTRAINT CheckStayDays CHECK ( HospitalStayDays >= 0 ),
    CONSTRAINT PatientPK PRIMARY KEY (PatientID),
    CONSTRAINT PatientFK1 FOREIGN KEY (PhysID)
        REFERENCES Physician(PhysID),
    CONSTRAINT PatientFK2 FOREIGN KEY (RoomNo)
        REFERENCES Room(RoomNo)
);
ALTER TABLE Department
    ADD CONSTRAINT DepartmentFK FOREIGN KEY (DeptSupervisorID)
        REFERENCES Physician(PhysID);
```


Connections

Oracle Connections

TigerConnection

- Tables (Filtered)
 - DEPARTMENT
 - PATIENT
 - PHYSICIAN
 - ROOM
 - TREATMENT
- Views
- Indexes
- Packages
- Procedures
- Functions
- Operators
- Queues
- Queues Tables
- Triggers
- Types
- Sequences
- Materialized Views
- Materialized View Logs
- Synonyms
- Public Synonyms
- Database Links
- Public Database Links
- Directories
- Editions
- Java
- XML Schemas
- XML DB Repository
- OLAP Option
- Analytic Views
- Scheduler

Reports

- All Reports
- Analytic View Reports
- Data Dictionary Reports
- Data Modeler Reports
- OLAP Reports
- TimesTen Reports
- User Defined Reports

Welcome Page x TigerConnection x

0.29899999 seconds

Worksheet Query Builder

```
CREATE TABLE Department
(
    PhysDept          NUMBER,
    DeptSupervisorID  NUMBER,

    CONSTRAINT DepartmentPK PRIMARY KEY (PhysDept)
);

CREATE TABLE Treatment
(
    TreatId           NUMBER,
    TreatCost          NUMBER(10, 2),
    TreatDesc          VARCHAR2(200),

    CONSTRAINT CheckTreatCost CHECK ( TreatCost >= 50.00 ),
    CONSTRAINT TreatmentPK PRIMARY KEY (TreatId)
);

CREATE TABLE Room
(
    RoomNo             NUMBER,
    RoomPhone           VARCHAR2(8),
    RoomRate            NUMBER(10, 2),

    CONSTRAINT CheckRoomNo CHECK ( RoomNo >= 100 AND RoomNo <= 999 ),
    CONSTRAINT CheckRoomRate CHECK ( RoomRate >= 30.00 AND RoomRate <= 100.00 ),
    CONSTRAINT RoomPK PRIMARY KEY (RoomNo)
);
```

Script Output x

Task completed in 0.299 seconds

Table TREATMENT created.

Table ROOM created.

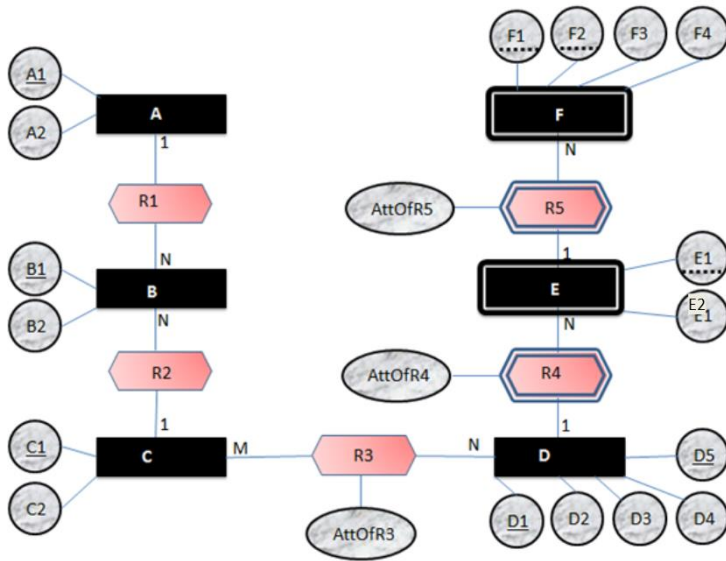
Table PHYSICIAN created.

Table PATIENT created.

Table DEPARTMENT altered.

Question #3

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



A (A1, A2)

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

C (C1, C2)

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

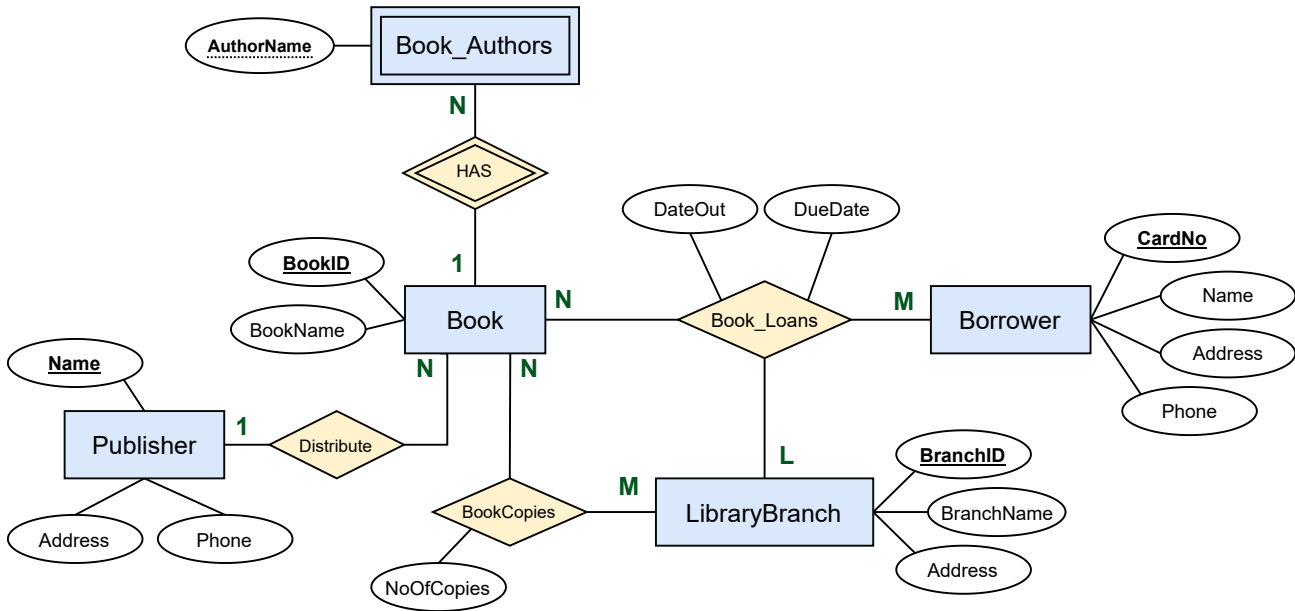
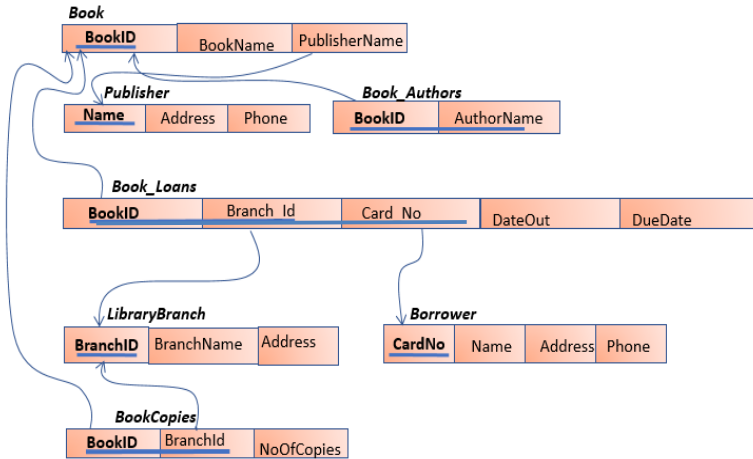
E (E1, (D1, D5)*, E2, AttOfR4)

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

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

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



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

