**Assignment#2 – CREATE TABLE Statement and Table Constraints**

**(Earn 10 points out of 11)**

--7.9/10

--Not sure on the last few questions.

1. 0.5 points – Define EmpID as primary key by column constraint (replace comment with your code).

--.25/.5

--Missing constraint

--CONSTRAINT PRIMARY KEY

CREATE TABLE Employee(

EmpID varchar(5) PRIMARY KEY,

Name varchar(50),

Location varchar(50),

Experience int

)

1. 0.5 points – Define EmpID as primary key by table constraint (replace comment with your code).

--.25/.5

--Missing constraint

--CONSTRAINT PRIMARY KEY

CREATE TABLE Employee(

EmpID varchar(5) PRIMARY KEY,

Name varchar(50),

Location varchar(50),

Experience int,

SAB

)

1. 1.0 points – Define primary key for table Tb\_Purchase as composed of Invoice# and Line# (replace comment with your code).

--1/1

--SAME AS PERSON IN CLASS

CREATE TABLE Tb\_Purchase(

Invoice# DECIMAL(10),

Line# DECIMAL(3),

Prod\_ID INT,

Quantity DECIMAL(8,2),

Price DECIMAL(8,2),

PRIMARY KEY(Invoice#, Line#)

)

1. 0.5 points – Define Title as candidate key by column constraint (replace comment with your code).

--.5/.5

--SAME AS PERSON IN CLASS

CREATE TABLE Album (

ID INT,

Title VARCHAR(30) UNIQUE,

ReleaseYear INT,

PRIMARY KEY (ID)

)

1. 1.0 points – Define candidate key for table Tb\_ Enrollment as composed of Student\_ID and Course\_ID (replace comment with your code).

--.5/1

--Not sure where I went wrong

--UNIQUE (Student\_ID, Course\_ID)

CREATE TABLE Tb\_Enrollment(

Enrollment# CHAR(8) PRIMARY KEY,

Student\_ID CHAR(20),

Course\_ID CHAR(20),

Grade DECIMAL(3,2),

Tb\_Enrollment CHAR(20)(Student\_ID, Course\_ID)

)

1. 1.0 points – Define constraints and defaults such that: courses can have 1 to 4 credits with 3 as default, there can be up to 6 sections of any course and course duration can be 4, 8 or 15 weeks with 15 as default (replace comments with your code).

--.9/1

--Missing default for duration. Have default in credits.

CREATE TABLE Course(

CourseNo CHAR(8) PRIMARY KEY,

Title VARCHAR(30) UNIQUE,

Credits INT DEFAULT 3 CHECK(Credits >=1 AND Credits <=4),

Sections INT CHECK (SECTIONS <= 6),

Duration INT CHECK(DURATION IN(4,8,15))

)

1. 1.0 points – Use a column constraint to define column AlbumID in table Song as foreign key referencing primary key ID in table Album (replace comment with your code).

--1/1

--SAME AS PERSON IN CLASS

CREATE TABLE Album (

ID INT,

Title VARCHAR(30) UNIQUE,

ReleaseYear INT,

PRIMARY KEY (ID)

)

CREATE TABLE Song (

ID INT,

Title VARCHAR(30),

Artist VARCHAR(30),

AlbumID INT FOREIGN KEY REFERENCES Album(ID),

PRIMARY KEY (ID)

)

1. 1.0 points – Use a table constraint to define column AlbumTitle in table Song as foreign key referencing candidate key Title in table Album (replace comment with your code).

--1/1?

--I think that I got the right results. I just commented the line of code out and put it in the bottom to make it work. Other than the first part, it is the same as the person in class.

CREATE TABLE Album (

ID INT,

Title VARCHAR(30) UNIQUE,

ReleaseYear INT,

PRIMARY KEY (ID)

)

CREATE TABLE Song (

ID INT,

--AlbumTitle VARCHAR(30),

Artist VARCHAR(30),

AlbumID INT,

PRIMARY KEY (ID),

AlbumTitle VARCHAR(30) FOREIGN KEY REFERENCES Album(Title)

--It works if you comment out the AlbumTitle since its included

)

1. 2.0 points – Add a foreign key definition to table Laboratory on columns DoctorID, PatientID, and AppointmentDate to reference the columns with the same name in table AppointmentBill. Add missing constraint to table AppointmentBill so that the foreign key definition in table Laboratory works correctly (replace comments with your code).

--1/2

--SAB. Commented some of the code and added it to the front. Same as the person who got it wrong in class. Have many foreign keys rather than one.

CREATE TABLE AppointmentBill(

Appointment\_no INT PRIMARY KEY,

DoctorID INT,

PatientID INT,

AppointmentDate DATETIME,

Charge MONEY,

CONSTRAINT FK\_AppointmentDate FOREIGN KEY (AppointmentDate) REFERENCES Appointment(no),

CONSTRAINT FK\_DoctorID FOREIGN KEY (DoctorID) REFERENCES Doctor(ID),

CONSTRAINT FK\_PatientID FOREIGN KEY (PatientID) REFERENCES Patient(ID)

)

CREATE TABLE Laboratory(

LabTest\_no INT PRIMARY KEY,

--DoctorID INT,

--PatientID INT,

--AppointmentDate DATETIME,

Amount MONEY,

DoctorID INT FOREIGN KEY AppointmentBill(DoctorID),

PatientID INT FOREIGN KEY AppointmentBill(PatientID),

AppointmentDate DATETIME FOREIGN KEY AppointmentBill(AppointmentDate)

CHECK (Amount>0)

)

1. 2.5 points – (a) Add foreign key constraints referencing the persons who are the mother and father of current person. (1.0 points); (b) Add check constraints to make sure every person has two parents, and no one can be her own parent. (1.5 points) ; (replace comments with your code).

--2.5/2.5

--I am not sure if I did this question correctly. I believe I did I mostly correct but I am not sure.

CREATE TABLE Person(

PersonID INT PRIMARY KEY,

Firstname varchar(15),

Lastname varchar(15),

Birthdate DATETIME,

Gender CHAR(1) CHECK (Gender IN ('M', 'F')),

Mother INT NULL FOREIGN KEY REFERENCES Mother(PersonID),

Father INT NULL FOREIGN KEY REFERENCES Father(PersonID),

CHECK (Mother = 1),

CHECK (Father = 1),

CHECK (PersonID <> Mother),

CHECK (PersonID <> Father)

)

1. ***(EXTRA CREDIT – 2%)*** – Add column definitions as needed and all the necessary constraints to guarantee that the references to one’s parents point to persons of the right gender and that no one is registered as her own parent (replace comments with your code).

--0/0

CREATE TABLE Person(

PersonID INT PRIMARY KEY,

Firstname varchar(15),

Lastname varchar(15),

Birthdate DATETIME,

Gender CHAR(1) CHECK (Gender IN ('M', 'F')),

Mother INT NULL,

/\* Your code here \*/,

Father INT NULL,

/\* Your code here \*/,

/\* Your code here \*/,

/\* Your code here \*/,

/\* Your code here \*/,

/\* Your code here \*/,

/\* Your code here \*/)

Robert Dollinger October 2, 2021