Worksheet 6 - creating databases

Jo Hardin

2023-01-16

Your Name: _____

Names of people you worked with: _____

- Don't introduce yourself. Name the people in front of you and behind you. Tell your partner what you did this weekend.
- What ideas do you have for working on the class project?

Consider the following fictional tables which exist in a fictional hospital database.[1]

donor

dName	age	bloodType
Alice	53	A+
Peter	34	AB+
Bob	44	AB-
Gert	23	A-

doctor

docName	insurance	rate
Wilhelm	HMO	15,000
Wilhelm	PPO	20,000
Heinz	HMO	12,000
Pferd	PPO	14,000

organ

donor	organ	available
Alice	Heart	2014
Bob	Lung	2015
Bob	Bladder	2015
Peter	Foot	2011
Gert	Lung	2014

patient

pName	insurance	age	bloodType
Hilde	HMO	13	A-
Fritz	PPO	87	AB+

takeCare

patient	organ	doctor
Hilde	Lung	Wilhelm
Fritz	Heart	Wilhelm

- Variables with black background are the primary keys of a table.
- The variable `donor` of table `organ` is a foreign key to table `donor`.
- The variable `patient` of table `takeCare` is a foreign key to table `patient`.
- The variable `doctor` of table `takeCare` stores doctors. However, it is not a foreign key to table doctor, because the primary key of that table also includes insurance information.

[1] Example taken from http://cs.iit.edu/~cs425/previous/14fall/

Task:

1. Write a **SQL** statement that

- creates a new table `assignedTo`
- which stores the `donor` and the donor's `organ` assigned to a `patient`.
- Furthermore, we want to store a treatment `price` for each such record.
- The combination of donor, organ, and patient uniquely identifies a record * `PRIMARY KEY`(s) and FOREIGN KEY‘(s) should be identified.
- Each record must have a treatment `price` that is bigger than 0 and smaller than 1,000,000 dollar.

2. Write a **SQL** statement that creates a table `worksFor` that records which `doctor` works for which `hospital`. For each such relationship between doctors and hospitals we record a `salary` for the doctor. A doctor may work for several hospitals (and obviously a hospital can employ multiple doctors).

Hint: can the table with only the variables `doctor`, `hospital`, and `salary` have any foreign keys?

Solution:

1. Write a **SQL** statement that creates a new table `assignedTo`, which stores the `donor` and the donor's `organ` assigned to a `patient`. Furthermore, we want to store a treatment `price` for each such record. The combination of donor, organ, and patient uniquely identifies a record. Each record must have a treatment price that is bigger than 0 and smaller than 1,000,000 dollar.

Hint: indicate the primary key(s) as well as the foreign key(s) which link to both the `organ` table and the `patient` table.

```
CREATE TABLE assignedTo (
    donor VARCHAR (256),
    organ VARCHAR (256),
    patient VARCHAR (256),
    price NUMERIC (8,2) NOT NULL CHECK (price BETWEEN 1 AND 999999),
    PRIMARY KEY (donor, organ, patient),
    FOREIGN KEY (donor, organ) REFERENCES organ(donor, organ),
    FOREIGN KEY (patient) REFERENCES patient(pName)
) ;
```

2. Write a **SQL** statement that creates a table `worksFor` that records which `doctor` works for which `hospital`. For each such relationship between doctors and hospitals we record a `salary` for the doctor. A doctor may work for several hospitals (and obviously a hospital can employ multiple doctors).

No foreign keys can be established because none of the variables in `worksFor` match up with primary keys in any of the other tables.

```
CREATE TABLE worksFor (
    doc VARCHAR (256),
    hospital VARCHAR (256),
    salary NUMERIC (8),
    PRIMARY KEY (doc, hospital)
);
```