CS2102 Database Systems

Semester 1 2019/2020

Assignment 03

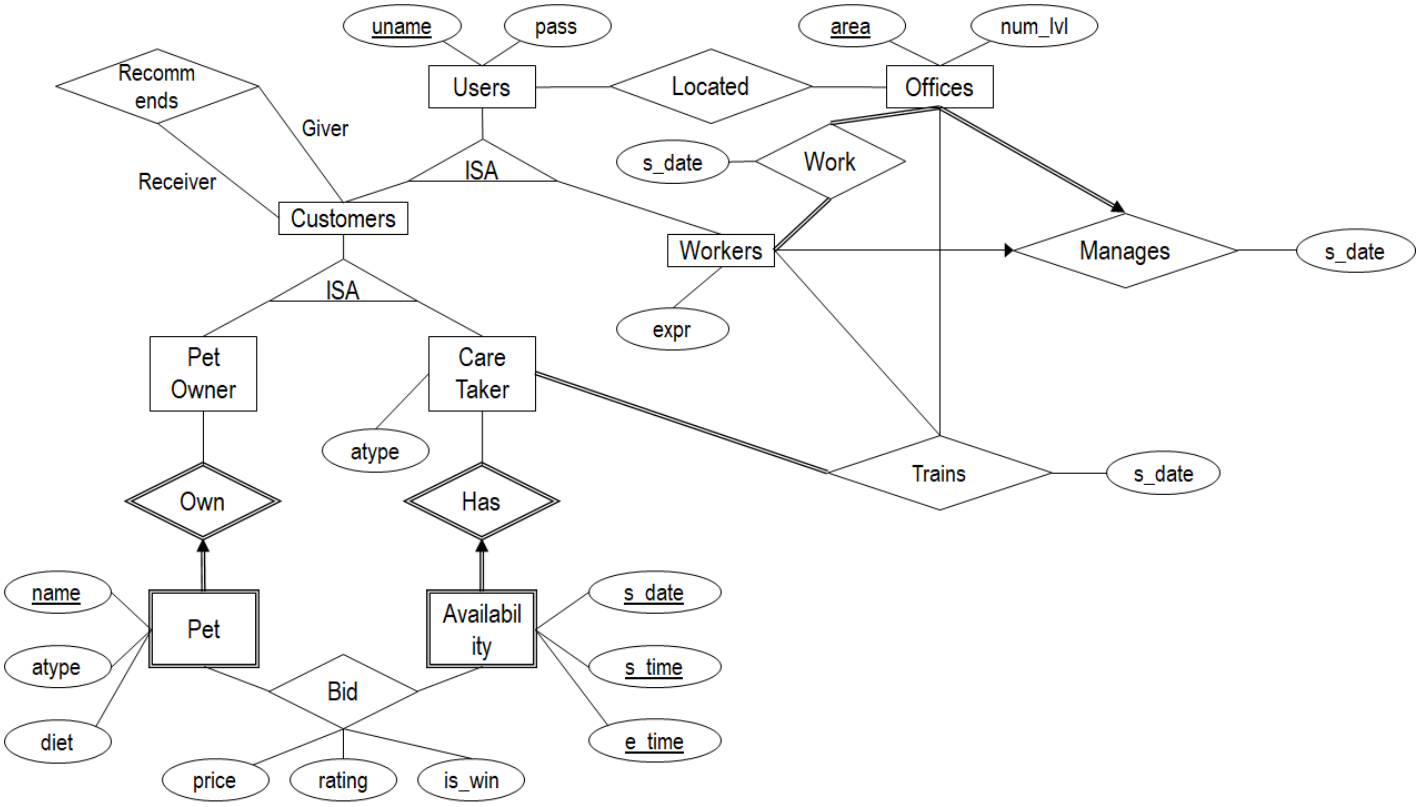
DEADLINE: 06 October 2019 **5 MARK**

1 Database schema

The questions are based on the following application about a Pet Caring business called PetER. This was part of project description from last semester. The topic is included to give you some idea about designing ER data model and its associated constraints.

Pet Caring. *This application allows pet owners to search for care takers for their pets for certain periods of time (e.g., https://dogvacay.com). Both users and pets have a profile. Care takers can advertise their availability (when they can take care of a pet, for how long, the kind of pet they can take care of and other constraints and requirements), and pet owners can browse/search for care takers and bid for their services. The successful bidder could either be chosen by the care taker or automatically selected by the system based on some criteria. Each user has an account.*

Its ER data model is shown below with the following constraints.



A user (identified by uid with name recorded [for simplicity, password is not recorded]) must be only of the following: a customer or a worker of PetER. Each user may or may not be closely located to some offices of PetER (identified by area with number of level of building recorded) but there may be users not closely located to offices and vice versa. Each customer must be one of the following: a pet owner, a care taker, or both. Each customer may give recommendations to other customers (which act as the receiver of said recommendations). Each worker (with previous number of year of work experience recorded) must work for some offices (which they may or may not be closely located at) and each office must have some workers working in the office. Each office also must be managed by exactly one manager and a worker may manage at most one office. The starting date of both work and manages must be recorded.

Each pet owner may own a pet (since new pet owner may not yet put their pet profile in the service) but each pet must be owned by a pet owner. Each pet (partially identified by name with the type of animal [e.g., cat, dog, salamander, etc] recorded as well as the pet's dietary restriction) is a weak entity set with pet owner as the owning entity. Each care taker (with preferred animal type recorded) must be trained at least once by some worker in some office. If further training of a care taker is done by the same worker in the same office, only the latest training need to be recorded. Each care taker can register their availability but some care taker may not have available

time slots. Each availability (partially identified by date, start time, and end time) is a weak entity set with care taker as the owning entity. All availability after current date and time is automatically advertised on the application front page. Each pet owner may bid for an availability slot of a care taker for the pet owner's pet with the bid price recorded but they cannot bid for their own availability. The successful bidder can be selected automatically based on (1) highest price, followed by (2) pet type compatibility, and (3) number of user location compatibility. By user location compatibility we mean the number of offices of PetER that the pet owner and care taker both closely located with. The selection is to be done 24 hours before the start of availability period. If there are any ties after the third comparison, the care taker can choose the winning bid manually. The winning bid is recorded via its is_win attribute and the optional rating ($0 \leq$ rating ≤ 5) may be given by pet owner after the end of availability period.

2 Relational schema

The following is the relational schema for this application.

```
CREATE TABLE Users (
  uname     varchar(50) PRIMARY KEY,
  pass      varchar(256) NOT NULL
);

CREATE TABLE Workers (
  uname     varchar(50) PRIMARY KEY REFERENCES Users (uname),
  expr      integer
);

CREATE TABLE Customers (
  uname     varchar(50) PRIMARY KEY REFERENCES Users (uname)
);

CREATE TABLE CareTaker (
  uname     varchar(50) PRIMARY KEY REFERENCES Customers (uname),
  atype     varchar(20) NOT NULL
);

CREATE TABLE PetOwner (
  uname     varchar(50) PRIMARY KEY REFERENCES Customers (uname)
);

CREATE TABLE Offices (
  area      varchar(20) PRIMARY KEY,
  num_lvl   integer DEFAULT 1,
  uname     varchar(50) NOT NULL UNIQUE REFERENCES Workers (uname), /* Manager */
  s_date    date NOT NULL
);

CREATE TABLE Located (
  uname     varchar(50) REFERENCES Users (uname),
  area      varchar(20) REFERENCES Offices (area),
  PRIMARY KEY (uname, area)
);

CREATE TABLE Work (
  uname     varchar(50) REFERENCES Workers (uname),
  area      varchar(20) REFERENCES Offices (area),
  s_date    date NOT NULL
);
```

```
CREATE TABLE Recommends (
  giver     varchar(50) REFERENCES Users (uname),
  receiver  varchar(50) REFERENCES Users (uname),
  PRIMARY KEY (giver, receiver)
);

CREATE TABLE Trains (
  cuname    varchar(50) REFERENCES CareTaker (uname),
  wuname    varchar(50) REFERENCES Workers (uname),
  area      varchar(20) REFERENCES Offices (area),
  PRIMARY KEY (cuname, wuname, area)
);

CREATE TABLE Pet (
  uname     varchar(50) REFERENCES PetOwner (uname)
                        ON DELETE cascade,
  name      varchar(50),
  atype     varchar(20) NOT NULL,
  diet      varchar(20) NOT NULL,
  PRIMARY KEY (uname, name)
);

CREATE TABLE Availability (
  uname     varchar(50) REFERENCES CareTaker (uname)
                        ON DELETE cascade,
  s_date    date,
  s_time    time,
  e_time    time,
  PRIMARY KEY (uname, s_date, s_time, e_time)
);

CREATE TABLE Bid (
  pouname   varchar(50),
  name      varchar(50),
  ctuname   varchar(50),
  s_date    date,
  s_time    time,
  e_time    time,
  price     numeric NOT NULL,
  rating    integer CHECK ((rating IS NULL) OR (rating >= 0 AND rating <= 5)),
  is_win    boolean DEFAULT FALSE,
  FOREIGN KEY (pouname, name) REFERENCES Pet (uname, name),
  FOREIGN KEY (ctuname, s_date, s_time, e_time)
              REFERENCES Availability (uname, s_date, s_time, e_time),
  PRIMARY KEY (pouname, name, ctuname, s_date, s_time, e_time),
  CHECK (pouname <> ctuname)
);
```

3 Restrictions

The following restrictions are in effect for this assignment.

- You are to answer using only one SQL queries within the view (terminated by a semicolon).
- You are to remove duplicate records from all query results.
- You are only allowed to use the concepts taught in class and limited to at most 2 CTEs in your answer.
- You are NOT to use single line comment (i.e., `-- comment`) but if you need comment, use C-style multi-line comments (i.e., `/* comment */`).
- You are NOT allowed to suspend the assessment.

4 Questions

Question 1. [0.5 marks]

Find all worker of PetER with at least 5 years of previous work experience. Answer the question by creating SQL view with the schema shown below:

```
CREATE VIEW qn1 (uname) AS ;
```

Question 2. [0.5 marks]

Find all worker of PetER who either have between 5 to 10 years of previous work experience (inclusive) or work in an office with at least 3 level. Answer the question by creating SQL view with the schema shown below:

```
CREATE VIEW qn2 (uname, expr) AS ;
```

Question 3. [0.5 marks]

Find all customers that are not pet owners. Answer the question by creating SQL view with the schema shown below:

```
CREATE VIEW qn3 (uname, atype) AS ;
```

where `uname` is the `uname` of the customer and `atype` is the preferred animal if customer is a care taker and `null` otherwise.

Question 4. [0.5 marks]

For each care taker, find all the pet with the same `atype`. Exclude care taker without any associated pet. Answer the question by creating SQL view with the schema shown below:

```
CREATE VIEW qn4 (ctuname, atype, name) AS ;
```

where `ctuname` is the `uname` of the care taker and `name` is the `name` of the pet.

Question 5. [0.5 marks]

For each customers, find the pair (`uname`, `num`) where `uname` is the customers's username and `num` is defined as follows:

- If the customer is a pet owner, then `num` is the number of recommendation the pet owner gives to any customers
- Otherwise, then `num` is the number of recommendation the customer received from any customers

Exclude any customers that the associated `num` is 0. Answer your question by creating SQL view with the schema shown below:

```
CREATE VIEW qn5 (uname, num) AS ;
```

Question 6. [0.5 marks]

We say that a care taker is *preferred* by a pet owner if one of the pet the pet owner owns is of the same `atype` as the care taker preferred animals. Find all pairs of (pet owner, care taker) such that the care taker is preferred by the pet owner and the care taker is not the pet owner themselves. Include pet owner that does not have the associated care taker. Answer your question by creating SQL view with the schema shown below:

```
CREATE VIEW qn6 (pouname, ctuname) AS ;
```

where `pouname` is the `uname` of the pet owner and `ctuname` is the `uname` of the care taker.

Question 7. [1 mark]

We say that a pet owner *likes* a care taker if the two customers satisfy all of the following:

- The two customers are different
- The pet owner has given a rating of at least 4
- The pet owner has won at least one of the care taker availability

Find all care taker that is not liked by any pet owner. We include any care taker that has not specified any availability, has not been bidded, or has no winning bid. Answer your question by creating SQL view with the schema shown below:

```
CREATE VIEW qn7 (ctuname) AS ;
```

where `ctuname` is the `uname` of the care taker.

Question 8. [1 mark]

For each care taker, find all potential successful bids. Recap that the potential successful bids must be based on the three criteria in the description and it may have after the third criteria. Exclude any availability that already has a winning bid. Exclude any care taker that has no availability or has not been bidded. Answer your question by creating SQL view with the schema shown below:

```
CREATE VIEW qn8 (pouname, name, ctuname, s_date, s_time, e_time, price) AS ;
```