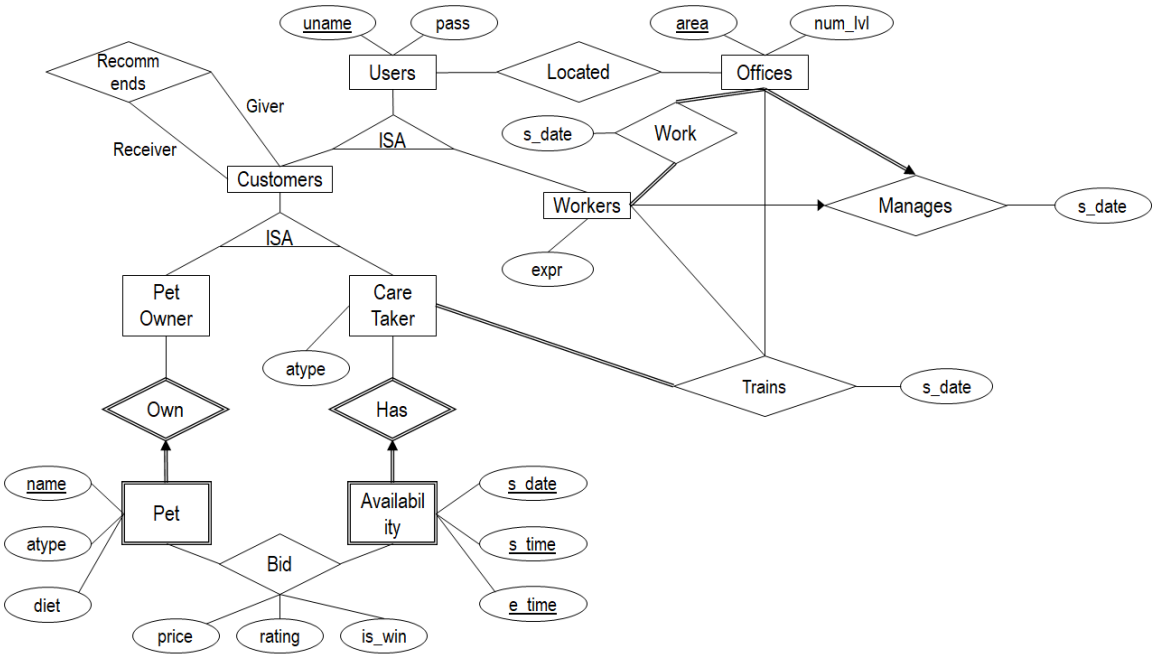
CS2102 Database Systems

Semester 1 2019/2020

Tutorial 06 (*Selected Answers*)

The entire tutorial will use the schema from Assignment 03 and Midterm. The ER diagram is reproduced below.



Quiz

1. We want to enforce the constraint that rating can only be given on a winning bid. Can we do it *without* trigger?

Tutorial Questions *[Discussion: 2, 3, 6]*

All the questions below are to be treated separately. The questions involving triggers are not intended to interact with one another. When answering a question, ignore constraints unrelated to the question.

2. Make a trigger on customers table to check that customers cannot be workers (or symmetrically, make a trigger on workers table to check that workers cannot be customers).

Solution:

```
CREATE OR REPLACE FUNCTION not_worker()
RETURNS TRIGGER AS $$
DECLARE count NUMERIC;
BEGIN
  SELECT COUNT(*) INTO count FROM Workers WHERE NEW.uname = Workers.uname;
  IF count > 0 THEN
    RETURN NULL; -- prevent
  ELSE
    RETURN NEW;  -- allow
  END IF;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER check_customer()
BEFORE INSERT OR UPDATE ON Customers
FOR EACH ROW EXECUTE PROCEDURE not_worker();
```

3. Create a stored procedure `add_customers` that takes in `uname` and `pass` and add the customer in a way that satisfy covering constraint on the ISA (or alternatively, create a stored procedure `add_workers` that takes in `uname`, `pass`, and `expr` and add the customer in a way that satisfy covering constraint on the ISA).

Solution:

```
CREATE OR REPLACE PROCEDURE add_customers(uname varchar(50), pass varchar(256))
AS $$
BEGIN
  INSERT INTO Users VALUES (uname, pass);
  INSERT INTO Customers VALUES (uname);
END;
$$ LANGUAGE plpgsql;
```

4. Consider a care taker specifying their availability with the date and time. When the availability is specified, it may *overlap* with another of the care taker's availability. We only care about overlap on any single day. For instance, on 26th August 2018, Alice specified that she is available on 1300 to 1500 as well as on 1430 to 1600. These two availability overlap.
   Create a trigger on any update and insertion on availability to check for overlap. When an overlap occur with an existing availability, *merge* the two availability such that they become a single availability with longer time. For instance, the overlap on 26th August 2018 for Alice can be resolved by making it a single availability from 1300 to 1600.
5. When an office is created (ignoring the constraint on work), a manager needs to be added. We wish to add a manager automatically via a trigger on insertion to offices. We want to choose the manager with the highest past work experience that has not become a manager. If there are multiple candidate, break ties with lowest `uname`. This will *replace* whoever manager is originally assigned to the office if it is a different manager.
6. We wish to prevent bidding on any availability that already has a winner. This can be enforced via trigger on any insertion or update to bid. Create this trigger.

Solution:

```
CREATE OR REPLACE FUNCTION not_won()
RETURNS TRIGGER AS $$
DECLARE count NUMERIC;
BEGIN
  SELECT COUNT(*) INTO count FROM Bid B
  WHERE  NEW.ctuname = B.ctuname AND NEW.s_date  = B.date
    AND  NEW.s_time   = B.s_time  AND NEW.e_time  = B.e_time
    AND  B.is_win;
  IF count > 0 THEN
    RETURN NULL; -- prevent
  ELSE
    RETURN NEW;  -- allow
  END IF;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER check_bid_insert()
BEFORE INSERT OR UPDATE ON Bid
FOR EACH ROW EXECUTE PROCEDURE not_won();
```

7. We want to ensure the quality of training performed by our worker. We only allow worker to train care takers if they have a combined total past work experience and current working experience at PetER of 5 years. For a worker who works in multiple offices, look for the oldest `s_date`. In other words, considering 1st January 2019, we allow worker with 3 years past work experience but have worked for PetER from 1st January 2017. Create a trigger on insertion on trains to enforce this constraint.