

CS2102 Database Systems

Semester 1 2019/2020

Assignment 03

5 Answers

Disclaimer: there are other correct answers as well, the answer shown here is only one possibility.

Question 1. Single-table query [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 ;
```

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

```
SELECT uname
```

```
FROM Workers W
```

```
WHERE W.expr >= 5
```

```
;
```

Question 2. Disjunctive condition [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 ;
```

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

```
SELECT DISTINCT W.uname, W.expr
```

```
FROM Workers W, Work Wk, Offices O
```

```
WHERE W.uname = Wk.uname AND O.area = Wk.area
```

```
AND ( ( W.expr >= 5 AND W.expr <= 10 ) OR  
( O.num_lvl >= 3 ) )
```

```
;
```

Question 3. Subquery [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.

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

```
SELECT DISTINCT C.uname, C.atype
```

```
FROM CareTaker C
```

```
WHERE C.uname NOT IN (  
    SELECT * FROM PetOwner  
)
```

```
;
```

Question 4. Simple join [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:

Relational Algebra

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

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

```
CREATE VIEW qn4 (ctuname, atype, name) AS
  SELECT DISTINCT C.uname, P.atype, P.name
  FROM   CareTaker C, Pet P
  WHERE  P.atype = C.atype
;
```

Question 5. Case analysis [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 ;
```

```
CREATE VIEW qn5 (uname, num) AS
  WITH otherwise AS (
    SELECT uname FROM CareTaker
  EXCEPT
    SELECT uname FROM PetOwner
  )
  SELECT  R.giver AS uname, COUNT(R.receiver) AS num
  FROM    Recommends R, PetOwner P
  WHERE   R.giver = P.uname
  GROUP BY R.giver
  UNION
  SELECT  R.receiver AS uname, COUNT(R.giver) AS num
  FROM    Recommends R, Customers C, otherwise O
  WHERE   R.receiver = O.uname
         AND R.giver = C.uname
  GROUP BY R.receiver
;
/* Note: this is not done in a case-analysis way using CASE-THEN-END. */
```

Question 6. Dangling tuple [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.

Relational Algebra

```
CREATE VIEW qn6 (pouname, ctuname) AS
SELECT DISTINCT PO.uname AS pouname, C.uname AS ctuname
FROM   PetOwner PO LEFT JOIN
      ( Pet P JOIN CareTaker C
        ON P.atype = C.atype AND P.uname <> C.uname )
      ON PO.uname = P.uname
;
```

Question 7. Business analysis [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 bidden, 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.

```
CREATE VIEW qn7 (ctuname) AS
SELECT uname AS ctuname
FROM   CareTaker
EXCEPT
SELECT CT.uname AS ctuname
FROM   PetOwner PO, CareTaker CT
WHERE  PO.uname <> CT.uname
      AND EXISTS (
        SELECT *
        FROM   Bid B
        WHERE  B.pouname = PO.uname
              AND B.ctuname = CT.uname
              AND B.rating >= 4.0
      )
      AND EXISTS (
        SELECT *
        FROM   Bid B
        WHERE  B.pouname = PO.uname
              AND B.ctuname = CT.uname
              AND B.is_win
      )
;
/* There is actually a rather simple solution here */
-- CREATE VIEW qn7 (ctuname) AS (
--   SELECT uname AS ctuname
--   FROM Caretaker
--   EXCEPT
--   SELECT ctuname
```

Relational Algebra

```
--      FROM Bid
--      WHERE is_win = TRUE OR rating >= 4
-- );
```

Question 8. Business analysis [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 bid. 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 ;
```

```
CREATE VIEW qn8 (pouname, name, ctuname, s_date, s_time, e_time, price) AS
SELECT B1.pouname, B1.name, B1.ctuname, B1.s_date, B1.s_time, B1.e_time,
B1.price
FROM Bid B1
WHERE NOT EXISTS (
    SELECT 1
    FROM Bid B2, Pet P, CareTaker C
    WHERE B2.ctuname = B1.ctuname
        AND B2.ctuname = C.uname
        AND B2.s_date = B1.s_date
        AND B2.s_time = B1.s_time
        AND B2.e_time = B1.e_time
        AND B2.pouname = P.uname
        AND B2.name = P.name
        AND B2.is_win
)
AND NOT EXISTS (
    SELECT 1
    FROM Bid B2, Pet P, CareTaker C
    WHERE B2.ctuname = B1.ctuname
        AND B2.pouname <> B1.pouname
        AND B2.ctuname = C.uname
        AND B2.s_date = B1.s_date
        AND B2.s_time = B1.s_time
        AND B2.e_time = B1.e_time
        AND B2.pouname = P.uname
        AND B2.name = P.name
        AND ( -- selection criteria
            ( B2.price > B1.price )
            OR ( B2.price = B1.price AND
                P.atype = C.atype )
            OR ( B2.price = B1.price AND
                P.atype <> C.atype AND
                ( SELECT COUNT(*)
                  FROM Located L1, Located L2
                  WHERE L1.area = L2.area AND L1.uname = B2.pouname AND L2.uname
= C.uname )
        )
    )
)
```

Relational Algebra

```
>  
( SELECT COUNT(*)  
  FROM   Located L1, Located L2  
  WHERE  L1.area = L2.area AND L1.uname = B1.pouname AND L2.uname  
= C.uname )  
      )  
    )  
  )  
;
```