Lucas Hasting
CS 447
11/15/2024
Homework #3

Contents

1 Question #1

Let $R(a, b, c)$, $S(a, b, c)$ and $T(a, b, c)$ be three relations. Write one or more Datalog rules that define the result of each of the following expressions of relational algebra:

a) $R \cup S$

b) $R \cap S$

c) $R - S$

d) $(R \cup S) - T$

e) $(R - S) \cap (R - T)$

f) $\pi_{a,b}(R)$

g) $\pi_{a,b}(R) \cap \rho_{U(a,b)}(\pi_{b,c}(S))$

Solution:

a)
```
A(a, b, c) <- R(a, b, c)
A(a, b, c) <- S(a, b, c)
```

b)
```
A(a, b, c) <- R(a, b, c) AND S(a, b, c)
```

c)
```
A(a, b, c) <- R(a, b, c) AND NOT S(a, b, c)
```

d)
```
AnsA(a, b, c) <- R(a, b, c)
AnsA(a, b, c) <- S(a, b, c)
AnsB(a, b, c) <- AnsA(a, b, c) AND NOT T(a, b, c)
```

e)
```
AnsA(a, b, c) <- R(a, b, c) AND NOT S(a, b, c)
AnsB(a, b, c) <- R(a, b, c) AND NOT T(a, b, c)
AnsC(a, b, c) <- AnsA(a, b, c) AND AnsB(a, b, c)
```

f)
```
A(a, b) <- R(a, b, c)
```

g)
```
AnsA(a, b) <- R(a, b, c)
U(b, c) <- R(a, b, c)
AnsB(a, b) <- AnsA(a, b) AND U(a, b)
```

2 Question #2

Let $R(x, y, z)$ be a relation. Write one or more Datalog rules that define $\sigma_C(R)$, where C stands for each of the following conditions:

a) $x = y$

b) $x < y$ AND $y < z$

c) $x < y$ OR $y < z$

d) NOT $(x < y$ OR $x > y)$

e) NOT $((x < y$ OR $x > y)$ AND $y < z)$

f) NOT $((x < y$ OR $x < z)$ AND $y < z)$

Solution:

a) A(x , y , z) <– R(x , y , z) AND x = y

b) A(x , y , z) <– R(x , y , z) AND x < y AND y < z

c) A(x , y , z) <– R(x , y , z) AND x < y
   A(x , y , z) <– R(x , y , z) AND y < z

d) A(x , y , z) <– R(x , y , z) AND x = y

e) A(x , y , z) <– R(x , y , z) AND x = y
   A(x , y , z) <– R(x , y , z) AND y >= z

f) A(x , y , z) <– R(x , y , z) AND x >= y AND x >= z
   A(x , y , z) <– R(x , y , z) AND y >= z

3 Question #3

Using the database schema of the Movies relation:

$Movies(title, year, length, genre, studioName, producer\#)$

$StarsIn(movieTitle, movieYear, starName)$

$MovieStar(name, address, gender, birthdate)$

$MovieExec(name, addresscert\#, networth)$

$Studio(name, address, presC\#)$

Write the following queries in SQL:

a) Who were male stars in Titanic?

b) Which stars appeared in movies produced by MGM in 1995?

c) Who is president of MGM studios?

d) Which movies are longer than Gone With the Wind?

e) Which executives are worth more than Merv Griffin?

Solution:

a)
```
SELECT StarsIn.starName
FROM StarsIn, MovieStar
WHERE StarsIn.starName = MovieStar.name
    AND MovieStar.gender = 'male'
    AND StarsIn.movieTitle = 'Titanic';
```

b)
```
SELECT StarsIn.starName
FROM StarsIn, Movies
WHERE Movies.title = StarsIn.movieTitle
    AND StarsIn.movieYear = 1995
    AND Movies.studioName = 'MGM';
```

c)
```
SELECT MovieExec.name
FROM Studio, MovieExec
WHERE Studio.presC# = MovieExec.addresscert#
    AND Studio.name = 'MGM';
```

d)
```
SELECT title
FROM Movies
WHERE length > (
        SELECT length
        FROM Movies
        WHERE title = 'Gone With the Wind'
    );
```

e)
```
SELECT name
FROM MovieExec
WHERE networth > (
          SELECT networth
          FROM MovieExec
          WHERE name = 'Merv Griffin'
      );
```

4 Question #4

Show how to alter your relation schemas for the movie example:

$Movies(title, year, length, genre, studioName, producer\#)$

$StarsIn(movieTitle, movieYeaar, starName)$

$MovieStar(name, address, gender, birthdate)$

$MovieExec(name, addresscert\#, networth)$

$Studio(name, address, presC\#)$

In the following ways:

a) Make *title* and *year* the key for $Movie$.

b) Require the referential integrity constraint that the producer of every movie appear in $MovieExec$.

c) Require that no movie length be less than 60 nor greater than 250.

d) Require that no name appear as both a movie star and movie executive.

e) Require that no two studios have the same address.

Solution:

a)
```
ALTER TABLE Movies
ADD CONSTRAINT keys
PRIMARY KEY (title, year);
```

b)
```
ALTER TABLE Movies
ADD CONSTRAINT producers
producer# INT REFERENCES MovieExec(addresscert#)
CHECK (producer# IN (
          SELECT addresscert#
          FROM MovieExec
        )
       );
```

c)
```
ALTER TABLE Movies
ADD CONSTRAINT movieLength
CHECK (length <= 250 AND length >= 60);
```

d)
```
ALTER TABLE MovieStar
ADD CONSTRAINT MovieStarNames
CHECK (name NOT IN (
          SELECT name
          FROM MovieExec)
       );

ALTER TABLE MovieExec
ADD CONSTRAINT MovieExecNames
CHECK (name NOT IN (
          SELECT name
          FROM MovieStar)
       );
```

e)
```
ALTER TABLE Studio
ADD CONSTRAINT addresses
UNIQUE(address);
```