CSC 370

Quiz:
Relational Data Model

Dr. Sean Chester

Fall 2022

Questions

1. (4 points) You have a relational database for tracking movies and who starred in them with the following relations:

   Movie(<u>title</u>, <u>year</u>, genre)
   MovieStar(<u>name</u>, address, birthdate)
   StarsIn(<u>title</u>, <u>year</u>, <u>starName</u>)

   The options below indicate a constraint and a syntactically correct attempt to write it in relational algebra. Indicate in which options the plain English constraint and the relational algebra statement match.

   - ☐ No star is born before 1900

   $$\sigma_{\text{birthdate}<\text{`01-01-1900'}}(\text{MovieStar}) = \emptyset$$

   - ☐ Since first starring in 'Birdemic: Shock and Terror', 'Alan Bagh' must star in a movie every year that some movie has been released.

   $$\pi_{\text{year}}(\text{Movie}) \bowtie_{\text{y<year}} \rho_{\text{(t,y,s)}}(\sigma_{\text{title=`Birdemic:Shock and Terror'} \wedge \text{starName=`Alan Bagh'}}(\text{StarsIn})) \subseteq \pi_{\text{year}}(\sigma_{\text{starName=`Alan Bagh'}}(\text{StarsIn}))$$

   - ☐ No two movies can have the same title and year

   $$\rho_{\text{M1}}(\text{Movie}) \bowtie_{\text{M1.title=M2.title} \wedge \text{M1.year=M2.year}} \rho_{\text{M2}}(\text{Movie}) = \emptyset$$

   - ☐ Every star in a movie must refer to a star in MovieStar

   $$\text{MovieStar} \bowtie_{\text{starName}\neq\text{name}} \text{StarsIn} = \emptyset$$

2. (4 points) Consider a database set up to track which students have enrolled in which classes. Choose which of the proposed functional dependencies (FD's) below are correct, keeping in mind the formal definition of an FD.

   In this question, a *course* refers to something recurring like CSC 370, whereas a *class* refers to a specific offering of a course, such as CSC 370 - 202201. Moreover, assume that every class has a single instructor and a single textbook.

   - ☐ instructor student → instructor student_name
   - ☐ instructor student class → course semester grade instructor_name
   - ☐ student_name → student_number
   - ☐ student instructor course → class

3. (5 points) You are given a relation R(A, B, C, D, E, F). Each option below indicates a set of functional dependencies (FD's) and an attribute set. Please indicate which of the options show an attribute set that is a key (not a superkey) for R with the corresponding set of FD's.

- ☐ **Attributes** {A, C, D, E}
  Functional dependencies
  $A \rightarrow B$

- ☐ **Attributes** {A, F}
  Functional dependencies
  $AB \rightarrow CD$
  $CD \rightarrow E$
  $F \rightarrow A$

- ☐ **Attributes** {C, D, F}
  Functional dependencies
  $A \rightarrow E$
  $C \rightarrow E$
  $BE \rightarrow F$
  $D \rightarrow AB$
  $EF \rightarrow F$

- ☐ **Attributes** {A}
  Functional dependencies
  $A \rightarrow B$
  $B \rightarrow CD$
  $C \rightarrow EF$
  $D \rightarrow A$

- ☐ **Attributes** {B}
  Functional dependencies
  $AC \rightarrow E$
  $B \rightarrow CF$
  $E \rightarrow AD$
  $F \rightarrow A$

4. (4 points) In each of the questions below, you are given a set of functional dependencies. Indicate which of them could produce a decomposition into three relations by applying the BCNF Decomposition Algorithm. For all questions, the initial relation is R(A, B, C, D, E).

- ☐ $A \rightarrow B$
  $B \rightarrow C$
  $C \rightarrow D$

- ☐ $AB \rightarrow BC$
  $BC \rightarrow DE$
  $C \rightarrow B$

- ☐ $A \rightarrow BC$
  $BC \rightarrow D$
  $C \rightarrow B$

- ☐ $A \rightarrow BCDE$
  $B \rightarrow ACDE$

Answer Key

Question 1

No star is born before 1900

$$\sigma_{\text{birthdate}<\text{`01-01-1900'}}(\text{MovieStar}) = \emptyset$$

Since first starring in 'Birdemic: Shock and Terror', 'Alan Bagh' must star in a movie every year that some movie has been released.

$$\pi_{\text{year}}(\text{Movie}) \bowtie_{\text{y}<\text{year}}$$
$$\rho_{(\text{t,y,s})}(\sigma_{\text{title=`Birdemic:Shock and Terror'} \wedge \text{starName=`Alan Bagh'}}(\text{StarsIn}))$$
$$\subseteq \pi_{\text{year}}(\sigma_{\text{starName=`Alan Bagh'}}(\text{StarsIn}))$$

Feedback
We need to ascertain whether the logic matches.

The relational algebra states:

- consider the (title,year,star) from all tuples in StarsIn, where title is Birdemic and the star is Alan Bagh.
- let's rename the variables as (t,y,s) for convenience.
- consider also all unique year values that appear in the Movie table
- let's use the theta join operator to match every one of those years to (t,y,s) values where y is greater (in effect, this will keep every year after y).
- Finally, let us confirm that all of those years appear in the set of years where Alan Bagh appeared in the StarsIn relation.

Indeed, the logic is equivalent, but this was an unnecessarily difficult question, especially for a quiz, and especially especially because we didn't have time to cover relational algebra constraints in lecture at the depth that I had planned.

Question 2

instructor student → instructor student_name
instructor student class → course semester grade instructor_name

Feedback
This is an example of the trivial dependency rule which can simplify the FD to:

instructor student → student_name

student_name can be functionally determined from student so clearly it can be determined if you have both instructor and student.

This is easier to see if we apply the splitting rule to get three separate FD's:

instructor student class → grade
instructor student class → course semester
instructor student class → instructor_name

We see that this is a more complicated version of the previous question, by visually striking out the attributes on the left-hand side that are distracting.

Question 3

Attributes {A}
Functional dependencies
A → B
B → CD
C → EF
D → A

Attributes {B}
Functional dependencies
AC → E
B → CF
E → AD
F → A

Feedback
A determines B, B determines both C and D, and C determines both E and F. Therefore, the closure of {A} = {ABCDEF}, i.e., all the attributes of R, because all elements of {ABCDEF} \ {A} can be determined transitively from {A}.

- We get {B, C, F} using B → CF
- We then get {A, B, C, F} using F → A (since F is now in the set)
- We can now use both A and C to obtain E
- Finally, we use E → D (using the splitting rule) to obtain the last remaining attribute of R, confirming that this is a superkey.

Moreover, since it is a single attribute, it cannot possibly be a superkey of another key.

Question 4

A → BC
BC → D
C → B

Feedback
This example comes from the worksheet.