Algorithms and datastructures
Exercises

Kristoffer Fischer Klokker
16/08/02
krklo21

2022

Contents

1 From E/R diagram til reltional model

1.1 The relational schemata must be created using the E/R-style translation

Song(title,year,EnteredBy)
Country(name,title)
PopularIn(title,name)
RockBallad(title,guitarSolo,hairLength)
PopSong(title,autotune)

Each set needs it own schema, to ensure an object wil fitt into the model. The entered by relation is joined in the song schema, due to it must be entered by a country in order to be in the database. Whereas a a country's most popular song may not be entered and therefore be null.

1.2 The relational schemata must be created using the object-oriented translation

Song(title,year,enteredBy)
Country(name)
PopularIn(title,name)
RockBallad(title,year,guitarSolo,hairLength)
PopSong(title,year,autotune)
PopRockBallad(title,year,guitarSolo,autotune,enteredBy)

Here the same argument goes for joining the enteredRelation but not the popular relation. No schema are able to be joined otherwise, due none of the attributes matching.

1.3 The relational schemata must be created using the null translation

Song(title,year,guitarSolo,hairLength,autotune,enteredBy)
Country(name,popularSong)

Here it can be joined into two schemas, the relations are joined into the schemas and the ISA's are joined into the Song schema.
The Country and Song schema was not joined due to it would create redundency, when a country had multiple popular songs.

1.4 Write CREATE TABLE statements for the second translation (b), namely for theobject-oriented translation

```
CREATE TABLE Song(
   title VARCHAR(20) PRIMARY KEY,
   year int,
   enteredBy VARCHAR(20)
);
CREATE TABLE Country(
   name VARCHAR(20) PRIMARY KEY
);
CREATE TABLE PopularIn(
   title VARCHAR(20) REFERENCES Song(title),
   name VARCHAR(20)  REFERENCES Country(name)
);
CREATE TABLE RockBallad(
   title VARCHAR(20) PRIMARY KEY,
   year INT,
   guitarSolo BOOLEAN,
   hairLength INT
);
CREATE TABLE PopSong(
   title VARCHAR(20) PRIMARY KEY,
   year INT,
   autotune BOOLEAN
);
CREATE TABLE RockBalladPopSong(
   title VARCHAR(20) PRIMARY KEY,
   year INT,
   guitarSolo BOOLEAN,
   hairLength INT,
   autotune BOOLEAN
);
)
```

1.5 How would the translation (a) change if there was another ISA relationship connecting to the entity Love song with the same attributes, as Rock ballad?

It would depend upon how important it is to know the song genre. If the genre is not important it would not change due the ability of joining it with Rock ballad. If the genre is important we would loose that information and therefore a schema would be added: LoveSong(title,guitarSolo,hairLength)

2 SQL queries

2.1 Write an SQL SELECT query (without subqueries), which lists theEduNames thatgot a timeslot in theafternoon

```
SELECT EduName FROM educations,timeslots WHERE
  Timeslot = 'afternoon' AND educations.EduID =
  timeslots.EduId;
```

2.2 Write an SQL SELECT query (without subqueries), which lists the total numberofGroups that got a timeslot in themorning

```
SELECT SUM(groups) FROM educations,timeslots WHERE
  Timeslot = 'morning' AND educations.EduID =
  timeslots.EduId;
```

2.3 Write an SQL SELECT query with subqueries, which lists allEduIDs that didnotget any timeslot at all

```
SELECT EduID FROM educations WHERE EduID NOT IN(
  SELECT EduID FROM timeslots)
```

2.4 Write an SQL SELECT query, which lists first all morning slots with the corre-sponding educaton IDs and the education names (in alphabetic order) followed byall afternoon slots with the corresponding educations (in alphabetic order).

```sql
SELECT EduId, EduName FROM (timeslots INNER JOIN
  educations USING (EduID)) ORDER BY Timeslot DESC,
  EduName
```

2.5 Write an SQL SELECT query, which lists all the EduNames that have gotten time-slots both in the morning and in the afternoon

```sql
SELECT EduName FROM educations WHERE EduId IN(SELECT
  EduId FROM (timeslots INNER JOIN timeslots AS
  timeslots2 USING (EduID)) WHERE timeslots.Timeslot =
   'morning' AND timeslots2.Timeslot = 'afternoon');
```

3 Relational algebra

A	B	C
1	2	3
9	3	8
4	7	1

R

A	B	D
2	3	1
3	1	3
5	3	4

S

3.1 For each of the following expressions, give the resulting relation as a table similarto the ones for R and S above.

3.1.a $\sigma_{A>D}(S)$

A	B	D
2	3	1
5	3	4

3.1.b $\Pi_{A,B}(R) \cap \Pi_{D,A}(S)$

This will be empty due to the non matching attributes.

3.1.c $S \bowtie R$

Empty

3.1.d $\sigma_{S.B<C}(S \times R)$

S.A	S.B	D	R.A	R.B	C
2	3	1	9	3	8
3	1	3	1	2	3
3	1	3	9	3	8
5	3	4	9	3	8

3.1.e $\gamma_{B,sum(D)}(S)$

B	SUM(D)
3	5
1	3

3.2 For each of the following SQL statements, write an expression of extended relational algebra which is equivalent in its effect to the following SQL statement independentof the actual tuples present in R and S

3.2.a SELECT R.A, S.B, C, D FROM R JOIN S ON C = D AND R.A > S.A

$\pi_{R.A,S.B,C,D}(R \bowtie_{C=D \wedge R.A>S.A} S)$

3.2.b SELECT DISTINCT A,B FROM R WHERE C = 3 OR B = 3

$\delta(\pi_{A,B}(\sigma_{C=3 \vee B=3}(R))$

4 Normalization

Consider the relation

$$R(MealName, Appetizer, MainCourse, Dessert, Guest, Address)$$

with the following functional dependencies:
$MealName \rightarrow Appetizer\ MainCourse\ Dessert$
$MainCourse \rightarrow MealName$
$Guest \rightarrow Address$

4.1 List all keys of R and argue why there can be no other keys

$MainCourse\ Guest$
$Guest, MealName$
These are the two keys from which all attributes can be found. Other combinations will not result in a full row.

4.2 Show that R is not in BCNF, i.e., show that there is at least one BCNF violation. Decompose R until it is in BCNF. Document the steps of the decomposition process and the resulting relations.

$MealName^+ = \{Appetizer\ MainCourse\ Dessert\}$
$R_1 = MealName^+$
$R_2 = \{MealName\ Guest, Address\}$
$R_1^+ = \{\}$
$R_2^+ = \{Appetizer\ MainCourse\ Dessert\}$

Therefore making the final relations:
$R_1 = \{Appetizer\ MainCourse\ Dessert\}$
$R_2 = \{MealName\ Guest, Address\}$

4.3 Analyze whether R is 3NF, if it is, show that there are no 3NF violations. If it is not, show that there is at least one 3NF violation and decompose R until it is in 3NF using the 3NF synthesis algorithm

The firsst FD $MealName \rightarrow Appetizer\ MainCourse\ Dessert$ is a violation due to the left side not being a non minal key and the rightside not being part of a key.
All of the FD's are minimal so that gives us the three relations:
$R_1 = \{MealName,\ Appetizer,\ MainCourse,\ Dessert\}$
$R_2 = \{MainCourse,\ MealName\}$
$R_3 = \{Guest,\ Address\}$
None of these relations are a non minal key so a fourth relation is added.
$R_4 = \{Guest,\ MealName\}$

5 Indices

5.1 Write CREATE statements for indices based on the primary keys.

```
    CREATE INDEX playerKeyIndex ON players(PlayerID);
    CREATE INDEX gameKeyIndex ON games(WhitePlayerID,
      BlackPlayerID);
```

5.2 What indices would you create additionally, if SQL queries of the following type areexecuted much more often than other SQL statements? Explain your reasoning.

In the case of the select being executed more than manipulations on the schema, a index for Name, NameOfWinningPlayer and Handicap can be used. Then when the select is performed every index can be used such only pointers from each index are left. Then pointers which point to the same rows may be selected.