Database Systems, CSCI 4380-01
Homework # 2 Answers

Database Description.

games(gameid, name, year, publisher, min_players, max_players, min_age_rec, playtime_min, playtime_max, iscooperative, description, link)
gametypes(gameid, gametype)
gamecategories(gameid, category)
gamemechanics(gameid, mechanic)
gamedesigners(gameid, designername)
onlinegamesites(siteid, url, price_per_month, notes)
gamesonsite(siteid, gameid, isfree, min_players, max_players)
gameprices(gameid, storename, price)
gamereviews(gameid, userid, review_text, review_date, stars, num_likes)
awardsnominations(gameid, awardname, year, iswinner)

Question 1. Write the following queries using relational algebra. You may use any valid relational algebra expression, break into multiple steps as needed. However, please make sure that your answers are well-formatted and are easily readable. Also, pay attention to the attributes required in the output!

(a) Return the `gameid, name` and `designername` of games that are either in `'Exploration'` or `'Adventure'` categories, but not in both categories.

Answer.

```
T1 = select_{category='Exploration'} gamecategories
T2 = select_{category='Adventure'} gamecategories
T3 = (T1-T2) union (T2-T1)
T4 = project_{gameid, name} (T3*games)

**Note: you could also do: T3 = (T1 union T2) - (T1 intersect T2)
```

(b) Return the `gameid, name` of games that are available in exactly two online sites.

Answer.

```
T1 = project_{siteid, gameid} (gamesonsite)
T2(siteid1, gameid1) = T1
T3(siteid2, gameid2) = T1

T4 = T1 join_{gameid=gameid1 and siteid<>siteid2} T2
     --games in T4 are available in at least two sites

T5 = T4 join_{gameid=gameid2 and siteid<>siteid3 and siteid2<>sideid3} T3
     --games in T5 are available in at least three sites

T6 = project_{gameid} T4 - project_{gameid} T5
     --games in T5 are available in exactly two sites

T7 = project_{gameid, name} (T6*games)
```

Suppose you are given the following set of functional dependencies for relation $R(A, B, C, D, E, F, G, H)$.

$$\mathcal{F} = \{AB \to CD, AC \to DE, EF \to AG\}$$

which is decomposed into $R1(A, C, D, E), R2(A, B, C)$ and $R3(A, B, F, G, H)$.

Use this relation to answer questions 2,3 and 4.

Question 2. Is this decomposition lossless? Show your work using the Chase decomposition algorithm.

Answer.

A	B	C	D	E	F	G	H
a	b1	c	d	e	f1	g1	h1
a	b	c	d2	e2	f2	g2	h2
a	b	c3	d3	e3	f	g	h

Apply $AB \to CD$ to rows 2 and 3.

A	B	C	D	E	F	G	H
a	b1	c	d	e	f1	g1	h1
a	b	c	d2	e2	f2	g2	h2
a	b	c	d2	e3	f	g	h

Apply $AC \to DE$ to all rows

A	B	C	D	E	F	G	H
a	b1	c	d	e	f1	g1	h1
a	b	c	d	e	f2	g2	h2
a	b	c	d	e	f	g	h

Last row has no subscript, hence this decomposition is lossless.

Question 3. Find the projection of $\mathcal{F}$ into each decomposed relation, $R1, R2, R3$, given by F_1, F_2, F_3.

Additionally, check whether the decomposition is dependency preserving or not. Explain your answer. To do this, you must find if F_1 *union* F_2 *union* F_3 is equivalent to the original set, $\mathcal{F}$.

$R1(A, C, D, E)$, $F_1 = \{AC \to DE\}$

$R2(A, B, C)$, $F_2 = \{AB \to C\}$

$R3(A, B, F, G, H)$, $F_3 = \{ABF \to G\}$

$F_{new} = F_1 \, union \, F_2 \, union \, F_3 = \{AC \to DE, AB \to C, ABF \to G\}$

We already know that all functional dependencies in F_{new} are implied by $\mathcal{F}$ (i.e. $F+_{new} \subseteq \mathcal{F}+$) because each fd in F_{new} is in $\mathcal{F}+$ by the definition of projections.

Now, we check if all the functional dependencies in the original set, $\mathcal{F}$ are implied by F_{new}.

$$F_{new} = \{AC \to DE, AB \to C, ABF \to G\}$$

$$\mathcal{F} = \{AB \to CD, AC \to DE, EF \to AG\}$$

Given F_{new}: $AB+ = \{A, B, C, D, E\}$, since $CD \subseteq AB+$, $AB \rightarrow CD \in F+_{new}$.

$AC \rightarrow DE$ is in F_{new}, so it is implied.

Given F_{new}: $EF+ = \{E, F\}$, since $\{A, G\} \nsubseteq EF+$, then $EF \rightarrow AG \in F+_{new}$. Hence, this projection is not dependency preserving.

Answer.

Question 4. Use the BCNF decomposition for the above set of functional dependencies, by decomposing using $AC \rightarrow DE$ first.

For each relation in the decomposition, show the functional dependencies and the key for that relation. Check if they are in BCNF.

If any of the resulting relations are not in BCNF, do any extra steps of BCNF decomposition to find relations that are in BCNF.

Answer. $R(A, B, C, D, E, F, G, H)$

$$\mathcal{F} = \{AB \rightarrow CD, AC \rightarrow DE, EF \rightarrow AG\}$$

Key: ABFH, BEFH

Decompose using $AC \rightarrow DE$:

$AC+ = \{A, C, D, E\}$

$R1(A, C, D, E)$, $F_1 = \{AC \rightarrow DE\}$, key: AC, in BCNF

$R2(A, B, C, F, G, H)$, $\{AB \rightarrow C, ABF \rightarrow G, ACF \rightarrow G\}$

Key: ABFH, not in BCNF

Decompose using $AB \rightarrow C$ (chosen randomly here)

$R21(A, B, C)$ $\{AB \rightarrow C\}$, in BCNF

$R22(A, B, F, G, H)$, $\{ABF \rightarrow G\}$, Key: ABFH, not in BCNF

Decompose using $\{ABF \rightarrow G\}$

$R221(A, B, F, G)$, $\{ABF \rightarrow G\}$, Key: ABF, in BCNF

$R222(A, B, F, H)$, $\{\}$, Key: ABFH, in BCNF

Final result:

(A, C, D, E), $F_1 = \{AC \rightarrow DE\}$, key: AC, in BCNF
(A, B, C) $\{AB \rightarrow C\}$, in BCNF
(A, B, F, G), $\{ABF \rightarrow G\}$, Key: ABF, in BCNF
(A, B, F, H), $\{\}$, Key: ABFH, in BCNF

Alternative solution. Alternatively you can start from $R2$ and follow a different path:

$R2(A, B, C, F, G, H)$, $\{AB \rightarrow C, ABF \rightarrow G, ACF \rightarrow G\}$

Decompose using $ACF \rightarrow G$

$R21(A, C, F, G)$, $\{ACF \rightarrow G\}$, key: ACF, in BCNF

$R22(A, B, C, F, H)$, $\{AB \rightarrow C\}$, key: ABFH, not in BCNF decompose using $AB \rightarrow C$,

$R221(A, B, C)$, $\{AB \rightarrow C\}$, key: AB, in BCNF

$R222(A, B, F, H)$, $\{\}$, key: ABFH, in BCNF

Final result:

(A, C, D, E), $F_1 = \{AC \rightarrow DE\}$, key: AC, in BCNF
(A, B, C), $\{AB \rightarrow C\}$, key: AB, in BCNF
(A, C, F, G), $\{ACF \rightarrow G\}$, key: ACF, in BCNF
(A, B, F, H), $\{\}$, key: ABFH, in BCNF

You can try another path and see if it gives a different result as well.

Question 5. You are given the following relation with the stated rules.

Restaurants(restaurant_name, state, street, city, zip, latitude, longitude, url, review_id, review_text, cuisine_type)

Given a restaurant_name and state, url is fixed, alternative given a URL, the restaurant name and state are fixed.

A given state, street, city and zip address has a specific latitude and longitude.

A latitude and longitude together also determine state, street, city and zip values.

Given a review_id, the review_text is fixed.

Given a URL, latitude and longitude, revivew_id is fixed.

1. List all the rules as functional dependencies. Make sure your final set of functional dependences are minimal.
2. List all keys for this relation.
3. State whether it is in BCNF or 3NF.
4. If it is not in 3NF, use the 3NF decomposition to convert it to relations in 3NF. Show your work by finding the functional dependencies projected into each relation, their keys and whether they are also in BCNF or not.
5. Discuss briefly if all the relations are in 4NF or not, why and why not.

Answer

1. Functional dependencies:

   restaurant_name state $\rightarrow$ url
   url $\rightarrow$ restaurant_name state
   state street city zip $\rightarrow$ latitude longitude
   latitude longitude $\rightarrow$ state street city zip
   review_id $\rightarrow$ review_text
   url latitude longitude $\rightarrow$ review_id

2. keys:

   url, street, city, zip, cuisine_type
   url, latitude, longitude, cuisine_type
   restaurant_name, state, street, city, zip, cuisine_type
   restaurant_name, latitude, longitude, cuisine_type

3. Not in 3NF or BCNF. All fds violate BCNF, review_id $\rightarrow$ review_text violates 3NF.

4. Third normal form decomposition:

   R1(restaurant_name, state, url)
   restaurant_name state $\rightarrow$ url, url $\rightarrow$ restaurant_name state
   Key: url or restaurant_name, state. In 3NF and BCNF.

   R2(state, street, city, zip, latitude, longitude)
   state street city zip $\rightarrow$ latitude longitude, latitude longitude $\rightarrow$ state street city zip
   Key: state, street,city, zip or latitude, longitude, In 3NF and BCNF.

   R3 (review_id, review_text)
   review_id $\rightarrow$ review_text
   Key: review_d, In 3NF and BCNF.

   R4 (url, latitude, longitude, review_id) url latitude longitude $\rightarrow$ review_id
   Key: url, latitude, longitude, In 3NF and BCNF.

   R5(url, latitude, longitude, cuisine_type) Key: all attributes, In 3NF and BCNF.

   We added R5 for one of the keys. You can choose any of the keys above, my choice was arbitrary. I chose one of the smallest keys.

5. R1, R2, R3 and R4 are all in 4NF because a subset of the attributes imply all the rest. Hence, the remaining attributes can have a single value for the unique attributes of the key.

   As for R5, no attribute imply each other. We understand this relation to be about the different locations of a restaurant and their cuisine type. We do not know what multi determines cuisine type.

   (a) Option 1: url, latitude, longitude $\Rightarrow$ cuisine_type
   which tells you there are multiple cuisine types for each specific location of a restaurant. If this is the case, then this relation is in 4NF and we are done.

   (b) Option 2: Use this multi-valued dependency: url $\Rightarrow$ cuisine_type
   which tells you that cuisine_type is multivalued but has nothing to do with location. This is not in 4NF. To convert it to 4NF, we will have this:
   (url, cuisine_type) key: both attributes (url, latitute, longitude) key: all attributes
   Both relations are in 4NF.

   Both solutions are acceptable and there might be other solutions as well.

   Note that in the second case, one can say that the second relation is unnecessary as the relevant information is already contained in R4. Hence, we might remove it without loosing any information. This is just the type of refinement you can go through after processing the relations.