Database Systems, CSCI 4380-01
Homework # 1 Answers
Due Monday September 13, 2021 at 11:59:59 PM

Homework Statement. This homework is worth 5% of your total grade. It has 10 questions with 10 points for each question. You are required to complete at least 4 queries (equivalent of 2 points). Any points that you did not complete will be added to Midterm #1. (For example, if you only solved 4 queries worth 2 points, your Midterm #1 will be worth 3% more).

Database Description. I don't know about you, but I played quite a lot of board games during the last few years. So, we will use a board game database for our homework for this reason. This database is loosely based on the `boardgamegeek` site. It contains the following relations:

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)

We store information about games as well as types, categories, mechanics and designs for each game. There are a number of online sites where a game can be played online (e.g. boardgamearena), for each site we have subscription prices as well as which game can be played on which site. In an online site, min and max player restrictions for the game may be different than a physical board game. Finally, games may be nominated for awards and if they win the award, `iswinner` is set to true.

Additionally, we store where the game is sold and at which price in `gameprices`.

Finally, in our game community, we have a number of user reviews and each review is by a user at the site. The reviews have text, star rating and number of likes by other member of the same community.

Note: All date fields are formatted as `mon-day-year`, e.g. `01-31-2020`. You can assume that you can check if a date value `X` comes after another value `Y` by checking whether `X > Y`.

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 `name`, min and max `playtime` and `link` of all games that can be played with 4 people, came out in 2020 and were published by `'Rio Grande Games'`.

Answer.

```
  T1 = select_{min_players<= 4 and 4 <= max_players
              and year=2020 and published='Rio Grande Games'} Games
  T2 = project_{name, playtime_min, playtime_max,link} T1
```

(b) Return the `name` and `designername` of games that won or were nominated for the `'Golden Geek Most Innovative Board Game'` award in 2019 or 2020.

Answer.

```
T1 = project_{gameid} (select_{awardname='GGMIBG'
                               and (year=2019 or year=2020)} awardsnominations)
T2 = T1 * gamedesigners * games
T3 = project_{name,designername}

Notes:
- ** shortened 'Golden Geek Most Innovative Board Game' to 'GGMIBG'
- If you do not project out year from awardnominations, then you cause
  an additional join condition that the award is in the same year as
  the year the game came out, which the question did not ask for.
```

(c) Return the `userid` of all users who never reviewed a game with the `'Loose a Turn'` mechanic.

Answer.

```
T1 = select_{mechanic = 'Loose a Turn'} gamemechanics
T2 = project_{userid} (T1 * gamereviews)
T3 = project_{userid} (gamereviews)
T4 = T3 - T2

Notes:

- T2 is the is id of all users who reviewed a game with the given mechanic
- We subtract T2 from all users who reviewed a game. Note that it is not
  possible to solve this question without a set subtraction!
```

(d) Return `gameid` of all award winner games in categories both `'Exploration'` and `'Adventure'` that do not involve any `'Dice Rolling'` mechanic.

Answer.

```
T1 = project_{gameid} (select_{iswinner='True'} awardnominations)
T2 = project_{gameid} (select_{category='Adventure'} (gamecategories))
T3 = project_{gameid} (select_{category='Exploration'} (gamecategories))
```

```
T4 = project_{gameid} (select_{mechanic='Dice Rolling'} (gamemechanics))
T5 = (T1 intersect T2 intersect T3) - T4

Notes:

- T1 is all award winner games
- Alternatively, you can compute T5 as follows:

T5 = (T1 * T2 * T3) - T4

- Note that set subtraction is necessary in this question.
```

An alternative way to solve the part for both Exploration and Adventure categories is as follows:

```
T1(gameid1, category1) = gamecategories
T2 = T1 join_{gameid1 = gameid} gamecategories
T3 = project_{gameid}(select_{category1 = 'Exploration' and category = 'Adventure'}(T2))

- Note that the join is necessary here as the condition refers to two
  different tuples, one for each category.
```

(e) Return the `name, publisher` of all names in the `Strategy` category that won an `'SXSW'` award and are either available for less than $40 in a store or can be played online.

Answer.

```
T1 = select_{category='Strategy'} Gamecategories
T2 = project_{gameid} (select_{winner=True and awardname='SXSW'} Awardnominations)
T3 = project_{gameid} (select_{price < 40} gameprices)
T4 = project_{gameid} (gamesonsite)
T5 = games * T1 * T2 * (T3 union T4)
T6 = project_{name, publisher} (T5)
```

(f) Find the `name, publisher` of `cooperative` games in the `'Farming'` category that are either of type `'Strategy'` or have the `'Hidden Victory Points'` mechanic (or both).

Answer.

```
T1 = select_{category='Farming'} (gamecategories)
T2 = project_{gameid} (select_{gametype='Strategy'} (gametypes))
T3 = project_{gameid} (select_{mechanic='Hidden Victory Points'} (gamemechanics))
T4 = (T2 union T3) * T1 * select_{iscooperative=True} (games)
T5 = project_{name, publisher} T4
```

(g) Return the `storename` of stores with the cheapest price for the game named `'Beyond The Sun'` that came out in 2019.

Assume there is a single game with this name. It is possible that multiple stores have the same cheapest price, if so, all such stores must be returned.

Answer.

```
T1 = project_{storename,price} (select_{name='Beyond The Sun'} games) * gameprices)
T2(storename1, price1) = T1
T3 = project_{storename} (T1 join_{price > price1} T2)
T4 = (project_{storename} T1) - T3

Notes:

- T1 is all the stores selling the game in question.
- T3 is all stores that are not the cheapest (for which there exists
  at least one other store with a cheaper price)
- T4 are the stores with the cheapest price (for which  there DOES NOT exist
  a store with a cheaper price)
- This query requires a set subtraction, there is no way to solve it without it!
```

Question 2. For the following relations, (a) find and list the keys, (b) check whether they satisfy BCNF, discuss why or why not, (c) check whether they satisfy 3NF, discuss why or why not.

To show that a relation is not in BCNF or 3NF, you only need to show a violation. To show that they are in BCNF or 3NF, check each functional dependency and discuss why it is ok.

1. $R1(A, B, C, D, E, F, G)$, $\mathcal{F} = \{ABC \rightarrow DG, G \rightarrow AEF\}$

   Answer. Keys: ABC, BCG. Not in BCNF because $G \rightarrow AEF$ where G is not a superkey. Not in 3NF, because E or F are not prime attributes either for this functional dependency.

2. $R2(A, B, C, D, E, F, G)$, $\mathcal{F} = \{ABD \rightarrow CEFG, AE \rightarrow BCDG\}$

   Answer. Keys: ABD, AE. Both in BCNF and 3NF because both functional dependencies have a superkey on the left.

3. $R3(A, B, C, D, E, F, G)$, $\mathcal{F} = \{AB \rightarrow CDE, BE \rightarrow F, F \rightarrow G\}$

   Answer. Keys. AB. Not in BCNF because both $BE \rightarrow F, F \rightarrow G$ do not have a superkey on the left hand side. It is also not in 3NF because F and G are not prime attributes.