Database Systems, CSCI 4380-01
Homework # 1
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).

Remember, practice is extremely important to do well in this class. I recommend that not only you solve this homework in its entirety, but also work on homeworks from past semesters. Link to those is already provided in Teams, which I am repeating here:

`http://cs.rpi.edu/~sibel/DBS_Past_Materials/`

This homework aims to test relational algebra first and foremost, and a bit of normalization theory.

It is recommended that you do the parts in sequence. The questions get harder and build on your knowledge of relational algebra from previous parts. Each question is equal weight.

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'`.

project_{name, playtime_min, playtime_max, link} (select_{min_players $\leq$ 4 and max_players $\geq$ 4 and year $\geq$ 01-01-2020 and year $\leq$ 12-31-2020 and publisher = 'Rio Grande Games'} (games))

(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.

project_{name, designername}((select_{awardname = 'Golden Geek Most Innovative Board Game' and year $\geq$ 01-01-2019 and year $\leq$ 12-31-2020} (awardsnominations)) * games * gamedesigners)

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

project_{userid} (gamereviews) - project_{userid}(gamereviews * (select_{mechanic = 'Loose a Turn'} (gamemechanics)))

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

R1 = project_{gameid} (gamemechanics - (select_{mechanic = 'Dice Rolling'}(gamemechanics)))
R2 = project_{gameid} (select_{iswinner = true}(awardsnominations))
R3 = (project_{gameid}(select_{category = 'Exploration'}(gamecategories))) intersect (project_{gameid}
(select_{category = 'Adventure'}(gamecategories)))
R4 = R1 intersect R2 intersect R3
Return R4

(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.

R1 = project_{gameid} (select_{category = 'Strategy'} (gamecategories))
R2 = project_{gameid} (select_{awardname = 'SXSW' and iswinner = true} (awardsnominations))
R3 = project_{gameid} (select_{price <40} (gameprices))
R4 = project_{gameid}(gamesonsite * onlinegamesites)
R5 = R1 intersect R2 intersect (R3 union R4)
R6 = project_{name, publisher} (R5 * games)
return R6

(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.

R1 = project_{gameid} (select_{category = 'Farming'} (gamecategories))
R2 = project_{gameid} (select_{gametypes = 'Strategy'}(R1 * gametypes))
R3 = project_{gameid} (select_{mechanic = 'Hidden Victory Points'} (R1 * gamemechanics))
R4 = project_{name, publisher} (select_{iscooperative = true} (R1 intersect (R2 union R3) *

games))
return R4

(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.

R1 = project_{gameid} (select_{name = 'Beyond The Sun' and year = 2019} (games))
R2 = R1 * gameprices
R3(gameid1, storename1, price1) = R2
R4 = R3 x R2 - select_{price1 > price} (R3 × R2)
R5 = project_{storename1} R4
return R5

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

   (a) keys:
   ABC+ = {A, B, C, D, G}(ABC → DG)
   ABC+ = {A, B, C, D, E, F, G}(G → AEF)

   G+ = {A, E, F, G}(G → AEF)
   BCG+ = {A, B, C, D, E, F, G}(ABC → DG)

   keys are ABC and BCG

   (b) ABC→ DG: ok, ABC is a super key

   G→AEF: not ok for BCNF: G is not a supre key

   (c) ABC→ DG: ok, ABC is a super key

   G→AEF: not ok for 3NF: G is not a supre key and E, F are not prime attributes

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

   (a) keys:
   ABD+ = {A, B, C, D, E, F, G}(ABD → CEFG)

   AE+ = {A, B, C, D, E, G}(AE → BCDG)
   AE+ = {A, B, C, D, E, F, G}(ABD → CEFG)
   keys are ABD and AE

   (b) ABD → CEFG: ok, ABD is a super key

   AE → BCDG: ok for BCNF: AE is a supre key

(c) ABD $\rightarrow$ CEFG: ok, ABD is a super key

AE $\rightarrow$ BCDG: ok for 3NF: AE is a supre key

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

   (a) keys:
   AB+ = {A, B, C, D, E}(AB $\rightarrow$ CDE)
   AB+ = {A, B, C, D, E, F}(BE $\rightarrow$ F)
   AB+ = {A, B, C, D, E, F, G}(F $\rightarrow$ G)

   BE+ = {B, E, F}(BE $\rightarrow$ F)
   ABE+ = {A, B, E, F, G}
   ABE+ = {A, B, C, D, E, F, G}(AB $\rightarrow$ CDE)

   F+ = {F, G}(F $\rightarrow$ G)
   ABF+ = {A, B, F, G}
   ABF+ = {A, B, C, D, E, F, G}(AB $\rightarrow$ CDE)
   AB is the only key, ABE and ABF are not since they are not minimal.

   (b) AB $\rightarrow$ CDE: ok, AB is a supre key

   BE $\rightarrow$ F: not ok for BCNF, BE is not a super key

   F $\rightarrow$ G: not ok for BCNF, F is not a super key

   (c) AB $\rightarrow$ CDE: ok, AB is a supre key

   BE $\rightarrow$ F: not ok for 3NF, BE is not a super key and F is not a prime attribute.

   F $\rightarrow$ G: not ok for 3NF, F is not a super key and G is not a prime attribute

SUBMISSION INSTRUCTIONS. Submit a single PDF or Text document for this homework using Submitty. No other format and no hand written homeworks please. No late submissions will be allowed.

If the Submitty for homework submissions is not immediately available, we will announce when it becomes available on Submitty.

Help with relational algebra formatting. While in class I have been using a text version of relational algebra, which I have allowed for many years for students who do not want to figure out the Greek symbols. However, many past solutions use the more standard version with Greek symbols. You can use either one in your solutions, but do not mix and match. Use one consistently.

I present you with the full syntax here in both ways (as well as the Latex symbols for it). Note that for the standard version, I simply use the Math mode in Latex.

Operation	Text Version	Standard Version
Set Union	R union S	$R \cup S$
Set Intersection	R intersect S	$R \cap S$
Set Difference	R - S	$R - S$
Rename	T(A,B,C) = R	$\rho_{T(A,B,C)}(R)$
Select	select_{C} (R)	$\sigma_C(R)$
Project	project_{A1,..,An} (R)	$\pi_{A1,..,An}(R)$
Cartesian product	R x S	$R \times S$
Theta-Join	R join_{C} S	$R \bowtie_C S$
Natural Join	R * S	$R * S$

As an additional help, I format one of the queries we did in class in the standard format below for two equivalent solutions. Please format to make sure your queries are readable by a human.

Query: Find the id and title of all movies starring a hero with the power to `'Stop Time'`.

Solution 1 (text format)

T1 = project_{heroname,multiverseid} (select_{power = 'stop time'}(MarvelHeroes))
T2 = project_{heroname,multiverseid} (select_{power = 'stop time'}(DCHeroes))
T3(heroname1,multiverseid1) = T1 union T2
T4 = T3 join_{heroname=heroname1 and multiverseid=multiverseid1} HeroInMovie
T5(heroname1,movieid1, heroname, multiverseid1) = T4
T6 = project_{movieid, title} (T5 join_{movieid = movieid1} Movies)

Solution 2 (standard format)

$$T1 = \pi_{heroname,multiverseid}(\sigma_{power='stoptime'}(MarvelHeroes)$$
$$T2 = \pi_{heroname,multiverseid}(\sigma_{power='stoptime'}(DCHeroes)$$
$$T3(heroname1, multiverseid1) = T1 \cup T2$$
$$T4 = T3 \bowtie_{heroname=heroname1andmultiverseid=multiverseid1} HeroInMovie$$
$$T5(heroname1, movieid1, heroname, multiverseid1) = T4$$
$$T6 = \pi_{movieid,title}(T5 \bowtie_{movieid=movieid1} Movies)$$