CS3431-A22 Wong
Assignment 3: Relational Algebra and More SQL

This assignment is to be typed, named as Assignment3 and saved in Word or PDF format. The following relational algebra symbols are provided for your use in the assignment:

$\sigma, \pi, \gamma, \delta, \bowtie, \leftarrow, \cap, \cup$

Part 1: Art Association (85 points, 17 points for each subpart)
The relations given below are from Assignment 2. The characters before the table name are the table abbreviations to be used in your relational algebra answers below. Do not use the table abbreviations in your SQL code unless you create them in your code.

AR Artwork (year PK FK, artworkID PK, artistID FK, title, materialsID FK, price, chosen, purchased, purchaseDate, gallery FK)

N NewArt (year PK FK, artworkID PK, artistID FK, title, materialsID FK, price, chosen, purchased, purchaseDate, gallery FK)

A Artist (artistID PK, firstName, lastName, city, state, memberLevel FK)

MB Membership (memberLevel PK, yearlyFee, discount)

M Materials (materialsID, category, medium)

G Gallery (gallery PK, floor, building FK)

B Building (buildingName PK, street, city, state, zipcode)

T TicketPrice (year PK, price)

You may assume that there are no duplications in the combination of year and artworkIDs between Artwork and NewArt.

1. (17 points) Based on the primary keys and foreign keys in the relational schema above, specify below the foreign key relationships that exist.
    a. **Using the table abbreviations**, write the constraints using the following format:
    Table1.ID1, Table1.ID2 References Table2.ID1, Table2.ID2 (for a 2-attribute foreign key)
        i. AR.year References T.year
        ii. AR.artistID References A.artistID
        iii. AR.materialsID References M.materialsID
        iv. AR.gallery References Gallery.gallery
        v. N.year References T.year
        vi. N.artistID References A.artistID
        vii. N.materialsID References M.materialsID
        viii. N.gallery References Gallery.gallery
        ix. A.memberLevel References MB.memberLevel
        x. G.building References Building.buildingName

b. Assume the tables above have already been created but without any foreign key references. Create just the SQL constraints for the foreign keys using the ALTER TABLE command and the full table names instead of the abbreviations. Note that if an Artist leaves the association, all of the records in tables that reference the Artist table will have its value for artistID set to null.
   i. ALTER TABLE Artwork ADD CONSTRAINT ARTWORK_YEAR_FK FOREIGN KEY (year) REFERENCES TicketPrice (year);
   ii. ALTER TABLE Artwork ADD CONSTRAINT ARTWORK_ARTISTID_FK FOREIGN KEY (artistID) References Artist (artistID);
   iii. ALTER TABLE Artwork ADD CONSTRAINT ARTWORK_MATERIALSID_FK FOREIGN KEY (materialsID) References Materials (materialsID);
   iv. ALTER TABLE Artwork ADD CONSTRAINT ARTWORK_GALLERY_FK FOREIGN KEY (gallery) References Gallery (gallery);
   v. ALTER TABLE NewArt ADD CONSTRAINT NEWART_YEAR_FK FOREIGN KEY (year) REFERENCES TicketPrice (year);
   vi. ALTER TABLE NewArt ADD CONSTRAINT NEWART_ARTISTID_FK FOREIGN KEY (artistID) References Artist (artistID);
   vii. ALTER TABLE NewArt ADD CONSTRAINT NEWART_MATERIALSID_FK FOREIGN KEY (materialsID) References Materials (materialsID);;
   viii. ALTER TABLE NewArt ADD CONSTRAINT NEWART_GALLERY_FK FOREIGN KEY (gallery) References Gallery (gallery);
   ix. ALTER TABLE Artist ADD CONSTRAINT ARTIST_MEMBERLEVEL_FK FOREIGN KEY (memberLevel) References Membership (memberLevel);
   x. ALTER TABLE Gallery ADD CONSTRAINT GALLERY_BUILDING_FK FOREIGN KEY (building) References Building (buildingName);

NO ‘NEWART’ TABLE BELOW THIS

2. (17 points) Display, just for artists who have a Student level membership, the artist’s full name in a column FullName (first name, a space, and then the last name), and the number of artworks that have titles which include the word ‘Beach’ in it. Use natural joins. Then make your relational algebra more efficient by filtering the records of the tables before joining the tables together. Remember to use the table abbreviations provided at the beginning of this assignment.

   a. Write the relational algebra

$$Result \leftarrow \gamma_{firstName \,||\, ' ' \,||\, lastName \text{ as } FullName,\ \ count(title)}(\sigma_{memberLevel='Student'}(A)) \bowtie (\sigma_{title\ LIKE\ '\%Beach\%'}(AR))$$

b. Translate the relational algebra in 2.a into SQL code. Note you will need to use subqueries in the natural joins in order to filter the records before the joins.

```
SELECT FullName, count(title)
FROM (
        SELECT title, artistID
        FROM Artwork
        WHERE title LIKE '%Beach%')
        NATURAL JOIN (
                SELECT firstName || ' ' || lastName AS FullName, artistID
                FROM Artist
                WHERE memberLevel='Student')
GROUP BY FullName;
```

3. (17 points) Write the relational algebra to display for each city, the number of artworks in each gallery located in that city. Only include artworks from the years 2009, 2013, and 2019. Use theta joins and make your relational algebra more efficient by filtering the records of the tables before joining the tables together.

$$T1 \leftarrow \sigma_{year=2009\ OR\ year=2013\ OR\ year=2019}(AR)$$
$$T2 \leftarrow G \bowtie_{G.building=B.buildingName} (B)$$
$$Result \leftarrow \gamma_{city,\ gallery,\ count(title)}(T1 \bowtie_{T1.gallery=T2.gallery} (T2))$$

4. (17 points) Write the relational algebra to display the following information for the artist who has donated the greatest total value of artworks: the first name, last name, and total value of all artwork donated with a heading of MaxDonatedValue. You will want to use the assignment operator (←) to make the relational algebra easier to do. Use theta joins.

$$T1 \leftarrow \sigma_{firstName,\ lastName,\ artistID}(A)$$
$$T2 \leftarrow \sigma_{artistID,\ price}(AR)$$
$$T3 \leftarrow T1 \bowtie_{T1.artistID=T\ .artistID} (T2)$$
$$Result \leftarrow \sigma_{ROWNUM=1}(\tau_{sum(price)\ DESC}(\gamma_{firstName,\ lastName,\ sum(price)}(T3)))$$

5. (17 points) Write the relational algebra to display for each membership level, the number of members from the state of Rhode Island with the heading NumberOfMembers, and the total of yearly fees with the heading of TotalFees. Only display the results for those membership levels where the total of yearly fees is greater than $5000.

    a. Write the relational algebra

$$T1 \leftarrow \sigma_{state='RI'}(A)$$
$$T2 \leftarrow \pi_{memberLevel,\ yearlyFee}(M)$$
$$T3 \leftarrow T1 \bowtie_{T1.memberLevel= \quad .memberLevel}$$
$$Result \leftarrow \sigma_{sum(yearlyFee)>500\ }(\gamma_{memberLevel,\ count(yearlyFee)\ AS\ NumberOfMembers,\ sum(yearlyFee)\ AS\ Total\ Fees}(T3))$$

    b. Translate the relational algebra in Part 5.a into SQL code

```
SELECT *
FROM (
    SELECT A.memberLevel, COUNT(yearlyFee) AS NumberOfMembers, SUM(yearlyFee) AS
TotalFees
    FROM (
        SELECT memberLevel
        FROM Artist
        WHERE Artist.state_='RI') A
        JOIN Membership ON A.memberLevel = Membership.memberLevel
    GROUP BY A.memberLevel)
WHERE TotalFees > 5000;
```

<u>Part 2</u>: Relational Algebra (15 points, 5 points for each subpart)

Provide the resulting tables with attribute headings for the following problems.

1. $N \bowtie_{B=T} (\pi_{B \text{ as } T} M - \pi_{A \text{ as } T} N)$

M

A	B	Y	Z
3	4	Tom	Jane
6	2	Mary	Susan
10	8	David	Paul
4	3	Mark	Helen
1	1	Lisa	Brian

N

A	B	Q
1	8	David
7	10	Jane
4	7	Paul
4	10	Tom
2	3	Susan

WORK
M
T | 4, 2, 8, 3, 1

N
T| 1, 7, 4, 4, 2

M-N = T | 2, 8, 3

$N \bowtie_{B=T}$ M-N
A | 1, 2
B | 8, 3
Q | David, Susan
T | 8, 3
ANSWER
Result

A	B	Q	T
1	8	David	8
2	3	Susan	3

2. $\pi_{B,Y,Z}(M) \bowtie \pi_{B,Q,\ A+B\ as\ Z}(N)$

M

A	B	Y	Z
3	4	Tom	2
6	2	Mary	9
4	3	Mark	5
1	1	Lisa	4

N

A	B	Q
1	1	David
7	2	Jane
4	7	Paul
4	10	Tom
2	3	Susan

WORK

M

B | 4, 2, 3, 1

Y | Tom, Mary, Mark, Lisa

Z | 2, 9, 5, 4

N

B | 1, 2, 7, 10, 3

Q | David, Jane, Paul, Tom, Susan

Z | 2, 9, 11, 14, 5

M NATURAL JOIN N (B : B & Z : Z)

M⋈N

B | 2, 3

Y | Mary, Mark

Z | 9, 4

Q | Jane, Susan

ANSWER

Result

B	Y	Z	Q
2	Mary	9	Jane
3	Mark	4	Susan

3. $\sigma_{A=1 \text{ or } B=3}\ (\pi_{A,B,Z \text{ as } Q}(M) \cup N)$

M

A	B	Y	Z
3	4	Tom	Jane
6	2	Mary	Susan
10	8	David	Paul
4	3	Mark	Helen
1	1	Lisa	Brian

N

A	B	Q
1	8	David
7	10	Jane
4	7	Paul
4	10	Tom
2	3	Susan

WORK

M

A | 3, 6, 10, 4, 1

B | 4, 2, 8, 3, 1

Q | Jane, Susan, Paul, Helen, Brian

N

A | 1, 7, 4, 4, 2

B | 8, 10, 7, 10, 3

Q | David, Jane, Paul, Tom, Susan

M U N

A | 3, 6, 10, 4, 1, 1, 7, 4, 4, 2

B | 4, 2, 8, 3, 1, 8, 10, 7, 10, 3

Q | Jane, Susan, Paul, Helen, Brian, David, Jane, Paul, Tom, Susan

A=1 or B=3

A | 4, 1, 1, 2

B | 3, 1, 8, 3

Q | Helen, Brian, David, Susan

ANSWER

Results

A	B	Q
4	3	Helen
1	1	Brian
1	8	David
2	3	Susan