**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:

σ, π, γ, **δ,** ⋈, 🡨, ∩, **∪**

**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.
   1. **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)

Table abbreviations: Gallery G, Building B, Artist A, Artwork AR, newArt NA

G.building references B.buildingName

A.memberLevel references M.memberLevel

AR.year references ticketprice.year

AR.artist references artist.artistID

AR.materialsID references materials.materialsID

AR.gallery references gallery.gallery

NA.year references ticketprice.year

NA.artist references artist.artistID

NA.materialsID references materials.materialsID

NA.gallery references gallery.gallery

* 1. 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.

ALTER TABLE gallery

ADD CONSTRAINT gallery\_building\_FK FOREIGN KEY (building) REFERENCES Building(buildingName)

ALTER TABLE artist

ADD CONSTRAINT artist\_FK FOREIGN KEY (memberLevel) REFERENCES membership (memberLevel)

ALTER TABLE artwork

ADD CONSTRAINT artwork\_year\_FK foreign key (year) references TicketPrice (year)

ALTER TABLE artwork

ADD CONSTRAINT artwork\_artistID\_FK foreign key (artistID) references Artist (artistID) ON DELETE SET NULL

ALTER TABLE artwork

ADD CONSTRAINT artwork\_Materials\_FK foreign key (materialsID) references Materials (materialsID)

ALTER TABLE artwork

ADD CONSTRAINT artwork\_gallery\_FK foreign key (gallery) references Gallery (gallery)

ALTER TABLE newArt

ADD CONSTRAINT newArt \_year\_FK foreign key (year) references TicketPrice (year)

ALTER TABLE newArt

ADD CONSTRAINT newArt \_artistID\_FK foreign key (artistID) references Artist (artistID) ON DELETE SET NULL

ALTER TABLE newArt

ADD CONSTRAINT newArt \_Materials\_FK foreign key (materialsID) references Materials (materialsID)

ALTER TABLE newArt

ADD CONSTRAINT newArt \_gallery\_FK foreign key (gallery) references Gallery (gallery)

1. (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.
   1. Write the relational algebra

σ, π, γ, **δ,** ⋈, 🡨, ∩, **∪**

γartist.firstName || ‘ ‘ || artist.lastName as FullName, count(\*)( (σartist.memberLevel = ‘Student’)Artist))⋈ ((σartwork.title like ‘%Beach%’ )(Artwork))

* 1. 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 firstname || ' ' || lastname as FullName, count(\*)

from (select \* from artwork where artwork.title like '%Beach%')

natural join (select \* from artist where memberLevel = 'Student')

group by firstName, lastName;

1. (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.

GB🡨 Gallery ⋈gallery.building=building.buildingName (Building)

γcity, gallery.gallery, count(\*)( (σartwork.year=’2009’ or artwork.year=’2013’ or artwork.year = ‘2019’)(Artwork))⋈artwork.gallery=GB.buildingGB

1. (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.

AA🡨 Artist ⋈artwork.artistID=artist.artistID (Artwork)

(σ(sum(price)=γmax(sum(price))))γfirstName, lastName, sum(price) as “MaxDonatedValue”(AA)

1. (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.
   1. Write the relational algebra

γartist.memberLevel, count(\*) as “NumberOfMembers”, sum(membership.yearlyFee) as “totalFees” (σartist.state = ‘RI’)(artist) ⋈artist.memberLevel=membership.memberlevel (σsum(membership.yearlyFee) > 5000 )(membership)

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

select artist.memberLevel, count(\*) as "NumberOfMemebers", sum(membership.yearlyFee) as "totalFees"

from artist

join membership on artist.memberLevel = membership.memberLevel

where artist.state = 'RI'

group by artist.memberLevel

having sum(membership.yearlyFee) > 5000;

**Part 2: Relational Algebra (15 points, 5 points for each subpart)**

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

1. N ⋈B=T ( π B as T M – π A as T N)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **M** | | | |  | **N** | | |
| **A** | **B** | **Y** | **Z** |  | **A** | **B** | **Q** |
| 3 | 4 | Tom | Jane |  | 1 | 8 | David |
| 6 | 2 | Mary | Susan |  | 7 | 10 | Jane |
| 10 | 8 | David | Paul |  | 4 | 7 | Paul |
| 4 | 3 | Mark | Helen |  | 4 | 10 | Tom |
| 1 | 1 | Lisa | Brian |  | 2 | 3 | Susan |

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

1. πB,Y,Z(M) ⋈ πB,Q, A+B as Z (N)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **M** | | | |  | **N** | | |
| **A** | **B** | **Y** | **Z** |  | **A** | **B** | **Q** |
| 3 | 4 | Tom | 2 |  | 1 | 1 | David |
| 6 | 2 | Mary | 9 |  | 7 | 2 | Jane |
| 4 | 3 | Mark | 5 |  | 4 | 7 | Paul |
| 1 | 1 | Lisa | 4 |  | 4 | 10 | Tom |
|  |  |  |  |  | 2 | 3 | Susan |

|  |  |  |  |
| --- | --- | --- | --- |
| B | Y | Q | Z |
| 2 | Mary | Jane | 9 |
| 3 | Mark | Tom | 5 |

1. σA=1 or B=3 (πA,B,Z as Q (M) **∪** N)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **M** | | | |  | **N** | | |
| **A** | **B** | **Y** | **Z** |  | **A** | **B** | **Q** |
| 3 | 4 | Tom | Jane |  | 1 | 8 | David |
| 6 | 2 | Mary | Susan |  | 7 | 10 | Jane |
| 10 | 8 | David | Paul |  | 4 | 7 | Paul |
| 4 | 3 | Mark | Helen |  | 4 | 10 | Tom |
| 1 | 1 | Lisa | Brian |  | 2 | 3 | Susan |

|  |  |  |
| --- | --- | --- |
| A | B | Z |
| 4 | 3 | Helen |
| 1 | 1 | Brian |
| 1 | 8 | David |
| 2 | 3 | Susan |