CS 3431 Assignment 3

Adam Camilli (aocamilli@wpi.edu)

September 12, 2017

Part 1: Tours

1. List for each level of guide - junior guide, guide or senior guide - how many mismatches there are between the required tours vehicle type and the guide’s license type.

$$RT_\theta \leftarrow \Big(RT \bowtie_{RT.customerID=C.customerID} C\Big) \bowtie_{RT.tourID=T.tourID} T$$

$$M_1 \leftarrow \Big(RT_\theta.vehicleType = \text{’boat’}\Big) \land \Big(RT_\theta.licenseType \neq \text{’sea’} \land RT_\theta.licenseType \neq \text{’both’}\Big)$$

$$\begin{aligned} M_2 \leftarrow &\Big(RT_\theta.vehicleType = \text{’car’} \lor RT_\theta.vehicleType = \text{’bus’}\Big) \\ &\land \\ &\Big(RT_\theta.licenseType \neq \text{’land’} \land RT_\theta.licenseType \neq \text{’both’}\Big) \end{aligned}$$

$$\boxed{\sigma_{M_1 \lor M_2}\Bigg(\gamma_{title,\text{COUNT}(title)\ \text{AS ’Mismatches’}}\Big(RT_\theta\Big)\Bigg)}$$

2. For each customer, list the first name, last name, and total amount being spent for land-based tours.

$$RT_\theta \leftarrow \Big(RT \bowtie_{RT.customerID=C.customerID} C\Big) \bowtie_{RT.tourID=T.tourID} T$$

$$\boxed{\sigma_{vehicleType\neq\text{’boat’}}\Bigg(\gamma_{firstName,\ lastName,\ \text{SUM}(price)\ \text{AS ’TotalLandPrice’}}\Big(RT_\theta\Big)\Bigg)}$$

3. Determine the customer who will make the most number of visits to tour locations. List the firstName, lastName, and the number of location visits (use the heading Visits).

$$\gamma_{firstName,\, lastName,\, \text{MAX}(Visits)\ \text{AS Visits}}\left(\gamma_{firstName,\, lastName,\, \text{COUNT}(locationID)\ \text{AS Visits}}\left(RT_{\theta}\right)\right)$$

Part 2: Science Fiction Books

1. SF stands for science fiction books. Based on the given primary keys, specify below the foreign key relationships that exist between the tables that would make sense.

   (a) Write the constraints using the following format: Foreign Key Table1.ID1 References Table2.ID2.

   Foreign Key SF.publisherName References P.publisherName.
   Foreign Key S.warehouseCode References W.code
   Foreign Key S.ISBN References SFBooks.ISBN
   Foreign Key AB.ISBN References SFBooks.ISBN
   Foreign Key AB.fullName AB.address References A.fullName A.address

   (b) Write named SQL constraints for the foreign keys. Note that if a publisher goes out of business, all of the books published by that publisher should automatically be deleted. Otherwise, tuples in referring tables are not deleted. Assume the tables with the field names already exist but without any foreign key constraints. Use ALTER commands to create the foreign key constraints.

```
ALTER TABLE SFBooks ADD CONSTRAINT SFBooks_publisherName_FK (publisherName)
    REFERENCES Publishers(publisherName) ON DELETE CASCADE;

ALTER TABLE Stocks ADD CONSTRAINT Stocks_warehouseCode_FK (warehouseCode)
    REFERENCES Warehouses(publisherName);

ALTER TABLE SFBooks ADD CONSTRAINT Stocks_ISBN_FK (ISBN)
    REFERENCES SFBooks(ISBN) ON DELETE CASCADE;

ALTER TABLE AuthorBooks ADD CONSTRAINT AuthorBooks_ISBN_FK (ISBN)
    REFERENCES SFBooks(ISBN) ON DELETE CASCADE;

ALTER TABLE AuthorBooks ADD CONSTRAINT AuthorBooks_fullName_FK (fullName)
    REFERENCES SFBooks(fullName) ON DELETE CASCADE;

ALTER TABLE AuthorBooks ADD CONSTRAINT AuthorBooks_ISBN_FK (address)
    REFERENCES SFBooks(address);
```

2. For each author, list the author name and address, and the average price of the authors books written by the author before 2000. For just this question, use natural joins instead of theta joins.

   (a) Write the relational algebra:

   $$\gamma_{fullName,\ address,\ \text{AVG}(price)}\Big(AB \bowtie_{SF.year<2000} SF\Big)$$

   (b) Write the SQL code for the above, but sorted by author names

```
SELECT AuthorBooks.fullName, AuthorBooks.address, AVG(SFBooks.price)
FROM AuthorBooks NATURAL JOIN SFBooks
WHERE SFBooks.year < 2000
GROUP BY AuthorBooks.fullName, AuthorBooks.address
ORDER BY AuthorBooks.fullName;
```

3. Report the warehouse code and city for warehouses that stock fewer than 10 copies of any book published by the publisher Wiley.

   (a) Write **efficient** relational algebra.

   $$\begin{aligned}\Pi_{code,\ city}\Bigg(W \bowtie_{W.code=S.warehouseCode}\\ \sigma_{numberOfBooks<10}\Big(S \bowtie_{S.ISBN=SF.ISBN}\\ \sigma_{publisherName=\text{'Wiley'}}\big(SF\big)\Big)\Bigg)\end{aligned}$$

   (b) Write **efficient** SQL code.

```
SELECT code,city
FROM Warehouses
WHERE code IN
    (SELECT warehouseCode
     FROM Stocks
     WHERE numberOfBooks < 10 AND ISBN IN
         (SELECT ISBN
          FROM SFBooks
          WHERE publisherName=’Wiley’)
    );
```

Part 3: Relational Algebra

1.

A	B	Q	T
6	4	David	6
3	1	Tom	3
3	2	Susan	3

2.

B	Q	Y	Z
3	Lisa	Susan	12

3.

Z	G	A	B	Q	H
α	2	2	3	4	1
α	2	2	5	10	1
β	8	2	3	3	3