CSE 462 Homework #1: Relational Model and Relational Algebra

Name: _____ Date: January 31, 2011

***** Due on February 9th, 2011 at the beginning of class. *****

Problem 1. [100pts] Consider the following schema of an online flight reservation agency.

`Customer(cid, lastName, firstName, dob)`
`Flight(fid, airline, fromCity, toCity, stops, onTime)`
`Reservation(cid, fid, date)`

Attribute `dob` is the customer's date of birth, `stops` is the number of stops (0–??) a flight requires, and `onTime` is a percentage (0-100) indicating how often a flight is on time. Keys are underlined. The set `{lastName, firstName, dob}` *is also a key* for `Customer`. However, the set `{airline, fromCity, toCity, stops, onTime}` *is not a key* for `Flight` since an airline may provide, say, different flights with two stops from Buffalo to Orlando, one stopping in Atlanta and one in New York, both of which are always on time. Attributes `cid` and `fid` in `Reservation` are foreign keys referencing homonymous attributes in `Customer` and `Flight`, respectively.

[10pts] Write a `CREATE TABLE` command for each relation. Include all constraints described above.

[90pts] Write Relational Algebra queries that find:

1. [9pts] `cid`'s of customers who reserved some flight to Boston and some flight to New York.

   RF := Reservation ⋈ Flight

   Answer := $\pi_{\text{cid}}(\sigma_{\text{toCity='Boston'}}(\text{RF})) \cap \pi_{\text{cid}}(\sigma_{\text{toCity='NewYork'}}(\text{RF}))$

2. [9pts] `cid`'s of customers who reserved some flight to Miami or some flight to Houston, but not both.

   RF := Reservation ⋈ Flight

   MIorH := $\pi_{\text{cid}}(\sigma_{\text{toCity='Miami'} \vee \text{toCity='Houston'}}(\text{RF}))$

   MIandH := $\pi_{\text{cid}}(\sigma_{\text{toCity='Miami'}}(\text{RF})) \cap \pi_{\text{cid}}(\sigma_{\text{toCity='Houston'}}(\text{RF}))$

   Answer := MIorH – MIandH

3. [9pts] `cid`'s of customers who reserved some flight to Boston before reserving some flight to New York.

   RF := Reservation ⋈ Flight

   B(cid, toBoston) := $\pi_{\text{cid,date}}(\sigma_{\text{toCity='Boston'}}(\text{RF}))$

   NY(cid, toNewYork) := $\pi_{\text{cid,date}}(\sigma_{\text{toCity='NewYork'}}(\text{RF}))$

   Answer := $\pi_{\text{cid}}(\sigma_{\text{toBoston<toNewYork}}(\text{B} \bowtie \text{NY}))$

4. [9pts] `cid`'s of customers who only reserved flights from Buffalo or to Buffalo.

   RF := Reservation ⋈ Flight

   Buffalo := $\pi_{\text{cid}}(\sigma_{\text{fromCity='Buffalo'} \vee \text{toCity='Buffalo'}}(\text{RF}))$

   NotBuffalo := $\pi_{\text{cid}}(\sigma_{\text{fromCity<>'Buffalo'} \wedge \text{toCity<>'Buffalo'}}(\text{RF}))$

   Answer := Buffalo – NotBuffalo

5. [9pts] `cid`'s of customers who never reserved AirTran flights or flights to Houston.

   RF := Reservation ⋈ Flight

   AirtranOrHouston := $\pi_{\text{cid}}(\sigma_{\text{airline='Airtran'}}(\text{RF})) \cup \pi_{\text{cid}}(\sigma_{\text{toCity='Houston'}}(\text{RF}))$

   Answer := $\pi_{\text{cid}}(\text{Customer})$ – AirtranOrHouston

6. [9pts] `cid`'s of customers who reserved every flight that is on time at least 85% of the time.

   RF := Reservation ⋈ Flight

   Actual(cid, fid) := $\pi_{cid,fid}$(RF)

   // for each cid assign all timely flights

   OnTime(cid, fid) := π_{cid}(Reservation) × π_{fid}($\sigma_{onTime>=85}$(Flight))

   // if some (cid, fid) is not removed from OnTime, the corresponding cid is dropped from the result

   // but this only happens if the cid did not actually reserve some timely flight

   Answer := π_{cid}(Actual) − π_{cid}(OnTime − Actual)

7. [9pts] `cid`'s of customers who reserved every flight that the customer with `cid=5` reserved.

   RF := Reservation ⋈ Flight

   Actual(cid, fid) := $\pi_{cid,fid}$(RF)

   // for each cid assign all flights reserved by the customer with cid=5

   Cid5(cid, fid) := π_{cid}(Reservation) × π_{fid}($\sigma_{cid=5}$(RF))

   // if some (cid, fid) is not removed from Cid5, the corresponding cid is dropped from the result

   // but this only happens if the cid did not actually reserve some flight that cid=5 did

   Answer := π_{cid}(Actual) − π_{cid}(Cid5 − Actual)

8. [9pts] `cid`'s of customers who reserved at least three distinct JetBlue flights to Miami.

   RF := Reservation ⋈ Flight

   JetBlueMI(cid, fid) := $\pi_{cid,fid}$($\sigma_{airline='JetBlue' \wedge toCity='Miami'}$(RF))

   Answer := π_{c}($\sigma_{fid1<fid2 \wedge fid2<fid3}$(JetBlueMI(c, fid1) ⋈ JetBlueMI(c, fid2) ⋈ JetBlueMI(c, fid3)))

9. [9pts] Pairs of `cid`'s of distinct customers who reserved the same flight and the first in the pair reserved before the second.

   RF := Reservation ⋈ Flight

   FCD(fid, cid, date) := $\pi_{fid,cid,date}$(RF)

   Answer := $\pi_{cid1,cid2}$($\sigma_{cid1<>cid2 \wedge date1<date2}$(FCD(fid, cid1, date1) ⋈ FCD(fid, cid2, date2)))

10. [9pts] Pairs of `cid`'s of distinct customers who reserved all the same flights. If (c1, c2) is returned, do not return (c2, c1).

    RF := Reservation ⋈ Flight

    FC := $\pi_{fid,cid}$(RF)

    // for each cid1 assign every pair (cid2, fid2) such that cid1 ≠ cid2 and fid2 was reserved by cid2

    Actual := $\pi_{cid1,cid2,fid2}$(FC(ignore, cid1) $\bowtie_{cid1<>cid2}$ FC(fid2, cid2))

    // for each (cid1,fid2) assign every cid2 such that cid1 ≠ cid2

    Required := $\pi_{cid1,cid2,fid2}$(FC(fid2, cid1) $\bowtie_{cid1<>cid2}$ FC(ignore, cid2))

    // if some (cid1, cid2, fid2) is not removed from Required, the corresponding (cid1, cid2) is dropped from the result

    // but this only happens if cid2 did not reserve flight fid2, one that cid1 actually reserved

    // NOTE: this does not imply cid2 reserved ***only*** all flights that cid1 did!!!

    2ndReservedAllOf1st := $\pi_{cid1,cid2}$(Actual) − $\pi_{cid1,cid2}$(Required(cid1, cid2, fid2) − Actual)

    // invert the cids of all tuples

    1stReservedAllOf2nd := $\pi_{cid1,cid2}$(2ndReservedAllOf1st(cid2, cid1))

    // if the intersection returns tuple (a,b) then a and b reserved all the same flights

    // note that if (a,b) is in the intersection, so is (b,a); the selection returns one of them

    Answer := $\sigma_{cid1<cid2}$(1stReservedAllOf2nd ∩ 2ndReservedAllOf1st)