UNIVERSITY OF TORONTO

Faculty of Arts and Science

MIDTERM TEST Fall 2018

CSC343H – Introduction to Databases
Instructor – Daniela Rosu and Sina Meraji

Duration – 90min

No aids allowed

Please answer all questions in the space provided. You may use the blank pages dispersed throughout the exam for rough work. In your answers try to be concise. **Good luck!**

Last Name	
First Name	
Student Number	

Marks

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Total
2	2	6	6	4	6	6	2	2	2	2	10	50

Consider the following relational schema, representing an airline database:

Airport(apID, name, city, country)

AirportDistance(apID1, apID2, distance)

Airline(cID, callSign, headquarters, mainHub, website)

Aircraft(acID, type, avgSpeed, maxSpeed, range, serialNumber, manufacturer, airline)

Employee(eID, name, airline, primary language, secondary language, startDate, salary, isPilot, rank)

PilotExpertise(eID, aircraftType)

Flight(cID, flightNumber, week_day, origin, destination, departureTime, arrivalTime)

FlightInstance(fID, cID, flightNumber, week_day**,** date, acID, actualDepartureTime, actualArrivalTime)

CrewAssignment(fID, eID, role)

Integrity constraints:
AirportDistance[apID1] □ Airport[apID]
AirportDistance[apID2] □ Airport[apID]
Airline[mainHub] □ Airport[apID]
Aircraft[airline] □ Airline[cID]
Flight[cID] □ Airline[cID]
Flight[origin] □ Airport[apID]
Flight[destination] □ Airport[apID]
FlightInstance[cID, flightNumber, week_day] □ Flight[cID, flightNumber, week_day]
FlightInstance[acID] □ Aircraft[acID]
CrewAssignment[fID] □ Flight**Instance**[fID]

CrewAssignment[eID] □ Employee[eID]

PilotExpertise[eID] □ Employee[eID]

PilotExpertise [aircraftType] □ Aircraft[type]

- Airports: apID is the *International Air Transport Association* (IATA) airport identifier. The estimated distance between two airports is stored (in km) in relation AirportDistance. *'YYZ'* is Pearson Airport's IATA code.
- Aircrafts: acID is an aircraft's international registration number (for example, C-GBIP). Aircrafts are registered with an airline, but can be leased to other airlines and assigned to flight implementations belonging to the leasing airline.
- Airlines: each airline is identified by its IATA code, for example '*AC'* for Air Canada, and the database stores information about the airlines international call sign, (e.g., *'Air Canada'*) as well as the locations of its main hub (e.g., *'YYZ'* for Air Canada) and headquarters (e.g., '*Amsterdam*' for KLM).
- Employees: the database holds information about the employees of all airlines. All pilots, flight attendants and ground crew are employees. For every employee, a primary language is

recorded. In addition, each employee may have a second language in which he or she is fluent. Pilots have a rank and expertise in flying different types of aircrafts.

- Flights: a flight is identified by a company code (for example, 'AC') and a flight number (for example, 456). For every flight, the database stores the origin and the destination airports (Toronto and Ottawa for AC456) and the scheduled departure and arrival times (3:10pm and 4:09pm for AC456). Fight
- Flight instance: contains information about the implementation of a flight on a specific date, for example, flight AC 456 on the 1st of July 2018 was implemented using aircraft C-GBIP, departed at 3:20pm and arrived at 4:15pm.
- Crew assignment: the database stores information about the employees (the crew) assigned to each flight implementation. The role of an employee must be one of 'Captain', 'Navigator', 'First Officer', flight attendant' and 'ground crew'.

Note: the schema is very similar, but not identical to the schema used for assignment 1.

Part 1 [4 marks] Integrity constraints.

Write the following integrity constraints using relational algebra:

Q1 [2 marks] "Every flight instance is assigned a 'Captain' and a 'First Officer'."

HasCaptain := $\pi_{\{fID\}}(\sigma_{\{role =\text{'Captain'}\}}$ (FlightInstance $\bowtie$ CrewAssignment))

HasFirstOficer := $\pi_{\{fID\}}(\sigma_{\{role =\text{'FirstOfficer'}\}}$ (FlightInstance $\bowtie$ CrewAssignment))

HasBoth := HasCaptain $\cap$HasFirstOfficer

Answer: $\pi_{\{fID\}}$(FlightInstance) – HasBoth = ∅

Q2 [2 marks] The 'Captain' assigned to a flight instance has expertise in the aircraft assigned to that instance.

FlightInstances_with_correct_expertise : = $\pi_{\{fID\}}((\sigma_{\{role =\text{'Captain'} \wedge type = aircraftType\}}$ (FlightInstance $\bowtie$ CrewAssignment $\bowtie$ PilotExpertise $\bowtie_{aircraftType\ =\ type}$ Aircraft)

Answer**: π{fID}(FlightInstance) - FlightInstances_with_correct_expertise = ∅**

Part 2. Relational algebra queries. Express the following queries in relational algebra. Use the assignment operator to improve the readability of your solution. Write "cannot be expressed" if you believe that the query cannot be expressed in relational algebra.

Q3 [6 marks] Find the eIDs of the employees who have been assigned as crew to exactly one flight implementation between Toronto ('YYZ') and Calgary ('YYC') and have never been assigned to a flight between Toronto and Ottawa ('YYO').

AssignedToTorontoCalgary := $\pi_{\text{eID, fID}}\ (\sigma_{\{(\text{origin} = \text{‘YYZ’} \wedge \text{destination} = \text{‘YYC’}) \vee (\text{origin} = \text{‘YYC’} \wedge \text{destination} = \text{‘YYZ’})\}}$ $(\text{CrewAssignment} \bowtie \text{FlightInstance} \bowtie \text{Flight}))$

AssignedToTorontoOttawa := $\pi_{\text{eID, fID}}\ (\sigma_{\{(\text{origin} = \text{‘YYZ’} \wedge \text{destination} = \text{‘YYO’}) \vee (\text{origin} = \text{‘YYC’} \wedge \text{destination} = \text{‘YYO’})\}}$ $(\text{CrewAssignment} \bowtie \text{FlightInstance} \bowtie \text{Flight}))$

Temp1(eID, fID) := AssignedToTorontoCalgary
Temp2(eID2, fID2) := AssignedToTorontoCalgary

TwoOrMoreTorontoCalgary := $\pi_{\text{eID, fID}}\ (\text{Temp1} \bowtie_{\{\text{eID} = \text{eID2} \wedge \text{fID} \wedge \text{fID2}\}} \text{Temp2})$

ExactlyOneTorontoCalgary := AssignedToTorontoCalgary - TwoOrMoreToronto_Calgary

NeverAssignedToTorontoOttawa := $\boldsymbol{\pi}_{\text{eID}}(\text{Employe}) - \boldsymbol{\pi}_{\text{eID}}(\text{AssignedToTorontoOttawa})$

Answer := ExactlyOneTorontoCalgary □ NeverAssignedToTorontoOttawa

Q4 [6 marks] Find the eIDs of the employees with the highest salary.

NotTheHighestSalaray := $\pi_{\text{E1.eID}}\ (\ \sigma_{\{\text{E1.salary} < \text{E2.salary}\}}(\ \rho_{\text{E1}}\ (\text{Employee}) \times \rho_{\text{E2}}\ (\text{Employee})))$

Answer: HighestSalary := $\boldsymbol{\pi}_{\textbf{eID}}$(Employee) - NotTheHighestSalaray (-2 marks for not having this statement)

Q5 [4 marks] Find the cID, flightNumber, and week_day for flights that have no implementations.

Flights_with_implementations := $\pi_{\text{cID, flightNumber, week\_day}}$ (Flight $\bowtie$ FlightInstance)

Answer: $\boldsymbol{\pi}_{\textbf{cID, flightNumber, week\_day}}$ **(Flights) - Flights_with_implementations**

Part 3 SQL. Express the following queries in SQL. Write **"cannot be expressed"** if you believe that the query cannot be expressed in SQL.

Q6. [6 marks] Find the airlines that have more than ten flights on 'Saturdays' and 'Sundays'. Return their cID and name.

```
SELECT
FROM Airline
WHERE cID IN  (SELECT cID
                FROM   Flight
                WHERE week_day = 'Saturdays'
                GROUP BY cID
                HAVING count(*) > =10

                INTERSECT

                SELECT cID
                FROM   Flight
                WHERE week_day = 'Sundays'
                GROUP BY cID
                HAVING count(*) > =10 )
```

Alternative interpretations also considered correct:
- airlines whose combined number of flights on Saturdays and Sundays exceeds 10

Q7. [6 marks] Find the airlines that lease at least as many planes as they have registered. A leased aircraft should be counted once towards the number of leased planes for an airline, regardless of the number of flight implementations it is assigned to by the leasing airline.

```
SELECT cID
FROM    Airline
WHERE cID IN   ( SELECT cID
                 FROM    (SELECT      airline as cID, count(acid) as registered_planes
                          FROM        Aircraft
                          GROUP BY airline) as T1,

                          (SELECT cID, count(*) as leased_planes
                    FROM     FlightInstance FI, Aircraft A
                    WHERE  FI.acID = A.acID
                        AND   FI. cID <> A.airline
                    GROUP BY cID) as T2

            WHERE T1.cID = T2.cID
                 AND leased_planes >= registered_planes )

                                       AS A_WITH_MORE_LEASED_PLANES
```

Also considered correct: interpretations that find the airlines that are lending out to other airlines at least as many planes as they have registered.

Part 4 [10 Marks] Database instances
Assume the following database instance that models information about twitter:

Follows:

```
   a    |     b
---------+-----------
 sina    | kanyewest
 sina    | bianca
 gary    | bob
 daniela | sina
 sina    | daniela
 sina    | jaime
 sina    | bruce
 bruce   | jaime
(8 rows)
```

Profile:

```
  id    |   name    | location
---------+-----------+-----------
 daniela | supergirl | Montreal
 sina    | biker     | vancouver
 jaime   | smart     | Montreal
 bruce   | architect | waterloo
 gary    | Daisy     | toronto
(5 rows)
```

Tweets:

```
id  | userid  |        content
-----+---------+------------------------
   1 | sina    | hellow twitter
   2 | daniela | what a day
   3 | jaime   | lets make canada great
   8 | bruce   |
   6 | sina    | hellow twitter
 123 | daniela | bye twitter
(6 rows)
```

Show the result of running each of the following queries. If a table is produced, include the column names. If the query generates an error, explain why.

Q8. [2 marks]

```
SELECT id, count(b) AS followers
FROM Profile JOIN Follows ON a = id;
```

```
ERROR:  column "profile.id" must appear in the GROUP BY clause or be used in an aggregate function
LINE 1: SELECT id, count(b) AS followers
```

Q9. [2 marks]

```
SELECT location
FROM Follows, Profile
WHERE id = a AND b = 'sina';
```

```
location
----------
 Montreal
(1 row)
```

Q10. [2 marks]

```
SELECT count(*)
FROM Profile LEFT JOIN Follows  ON a = id;
```

```
count
-------
     9
(1 row)
```

Q11. [2 marks]

```
SELECT P.id, count(T.content) AS number
FROM Profile P JOIN Tweets t ON T.userid = P.id
AND P.location = 'Montreal'
GROUP BY(p.id);
```

```
   id    | number
---------+--------
 jaime   |      1
 daniela |      2
(2 rows)
```

Part 5 [10 Marks] (True/False Statements)

Q12. For each of the following statements, indicate whether they are *true* or *false*. Please clearly *circle* or underline the answer you think is correct. The marking scheme is as follows:

- **+1**: for correct answer
- **-1**: for incorrect answer
- **0**: for not answering

The minimum mark for this question is 0 (never below).

(*) 1 point by default for "Check is less sensitive to NULLs than Where". All other questions marked according to the marking scheme above.

Answer		**Statement**
TRUE	**FALSE**	We can count values in relational algebra
TRUE	**FALSE**	Check is less sensitive to NULLs than Where (*)
TRUE	**FALSE**	In relational algebra (***bag-semantics***) the following holds: {a, a, b, b, c, c} ∩ {b, b, c, d} = {c}
TRUE	**FALSE**	In relational algebra we can change the execution order of select and projection
TRUE	**FALSE**	In relational algebra (***bag-semantics***) the following holds: {4,5,6,6,7,6,8} - {4,6,7} = {5,8}
TRUE	**FALSE**	The default output of projection does not have duplicate values in SQL

TRUE	**FALSE**	In SQL, it holds that: FALSE or NULL → FALSE
TRUE	FALSE	In SQL, *views* can be used to hide data in the database from third-party developers.
TRUE	**FALSE**	The main problem with indexes is the storage
TRUE	**FALSE**	If **|T|** is the cardinality of an inner join between two relations **R** and **S**, then: |T| <= max(|R|,|S|)