CSC 343H1 Y 2019 Midterm Test
Duration — 75 minutes
Aids allowed: none

Student Number: |_|_|_|_|_|_|_|_|_|_|_|_|_|

UTORid: |_|_|_|_|_|_|_|_|

Last Name: _____ **First Name:** _____

Lecture Section: L5101 (R 6-9)
Instructor: Mark Kazakevich

---

*Do **not** turn this page until you have received the signal to start.*
(Please fill out the identification section above, **write your name on the back of the test**, and read the instructions below.)
Good Luck!

This midterm is double-sided, and consists of 5 questions. *When you receive the signal to start, please make sure that your copy is complete.*

- In questions involving relational algebra, you may use only the basic operators $\Pi, \sigma, \bowtie, \times, \cap, \cup, -, \rho$, and assignment.
- Commentary on what you are doing in your queries is not required except where indicated, although it may help us mark your answers.
- If you use any space for rough work, indicate clearly what you want marked.
- **Do not remove any pages from the test booklet.**

1: _____/ 7
2: _____/ 4
3: _____/ 8
4: _____/ 8
5: _____/ 8

TOTAL: _____/35

Question 1. [7 marks]

Part (a) [5 marks]

Indicate whether each statement is True or False by circling the appropriate answer.

TRUE	FALSE	All integrity constraints represent a foreign key constraint.
TRUE	FALSE	Theta joins are natural joins that include a select operator on a condition.
TRUE	FALSE	The assignment operator is not usually used to update the content of an existing relation.
TRUE	FALSE	A natural join on two tables with no common attributes results in an empty relation.
TRUE	FALSE	If a relational algebra query has a select operator followed by a project operator, you cannot always swap the positions of the operators to get the same resulting relation.

Part (b) [2 marks]

Briefly explain what is meant by 'dangling' tuples. Show an example with some small tables.

Recall this schema, which we have used many times in class.

Relations

Student(<u>sID</u>, surName, firstName, campus, email, cgpa)
Course(<u>dept, cNum</u>, name, breadth)
Offering(<u>oID</u>, dept, cNum, term, instructor)
Took(<u>sID, oID</u>, grade)

Integrity constraints

Offering[dept, cNum] $\subseteq$ Course[dept, cNum]
Took[sID] $\subseteq$ Student[sID]
Took[oID] $\subseteq$ Offering[oID]

Question 2. [4 MARKS]

Part (a) [2 MARKS]

Consider this constraint:

$Proom(cNum1, cNum2, term) :=$

$$\Pi_{O1.cNum, O2.cNum, O1.term} \sigma_{\substack{O1.cNum < O2.cNum \\ \wedge \\ O1.dept = O2.dept = \text{`}CSC\text{'} \\ \wedge \\ O1.instructor = O2.instructor \\ \wedge \\ O1.term = O2.term}} [(\rho_{O1} Offering) \times (\rho_{O2} Offering)]$$

$$\sigma_{\substack{P1.cNum1 = P2.cNum1 \\ \wedge \\ P1.cNum2 = P2.cNum2 \\ \wedge \\ P1.term \neq P2.term}} [(\rho_{P1} Proom) \times (\rho_{P2} Proom)] = \emptyset$$

Define an instance of Offering that violates the constraint.

oID	dept	cNum	term	instructor

Part (b) [2 MARKS]

Write the following constraint using relational algebra: If a student takes a course taught by Horton, they cannot take a course taught by Gries.

Question 3. [8 marks]

Write a query in relational algebra to find the following: Consider the student(s) who received the highest grade over all breadth courses (a course where 'breadth' is true). Of those students, find sIDs of the ones who study on the 'St. George' campus.
You should break up your query into steps using the assignment operator. Adding commentary will help you understand your answer and can help us grade your answer.

Student(sID, surName, firstName, campus, email, cgpa)
Course(dept, cNum, name, breadth)
Offering(oID, dept, cNum, term, instructor)
Took(sID, oID, grade)

Offering[dept, cNum] $\subseteq$ Course[dept, cNum]
Took[sID] $\subseteq$ Student[sID]
Took[oID] $\subseteq$ Offering[oID]

Question 4. [8 marks]

For this question, you will write SQL queries using a version of the Restaurants schema from Assignment 1.

Relations

Restaurant(<u>name</u>, owner, capacity, country)
Patron(<u>PID</u>, name, birthday)
Dish(<u>DID</u>, name, dietary)
Reservation(<u>RID</u>, PID, rname, date)
Order(<u>RID, DID</u>, number)
Rating(<u>PID, rname</u>, rating, comment)

Integrity constraints

Reservation[PID] $\subseteq$ Patron[PID]
Reservation[rname] $\subseteq$ Restaurant[name]
Order[RID] $\subseteq$ Reservation[RID]
Order[DID] $\subseteq$ Dish[DID]
Rating[PID] $\subseteq$ Patron[PID]
Rating[rname] $\subseteq$ Restaurant[name]

Part (a) [3 marks]

In our schema, users can leave **comments** on their restaurant rating, and the comment text can be null. Write a query in SQL to find, for each Patron who has made a rating for a restaurant (where the comment text is not null), their name and the number of restaurants they have made a comment on. Report the Patron's name and the number of restaurants. Organize the output in non-increasing order by the number of restaurants.

Part (b) [2 marks]

Write a query in SQL that finds the names and owners of all restaurants that have a lower capacity than the restaurant named 'Red Lobster'.

The following query is supposed to print the number of pairs of dishes which have the same dietary restriction. It runs but does not always give the correct output.

```
SELECT count(*)

FROM (

    SELECT D1.DID, D2.DID

    FROM Dish D1, Dish D2

    WHERE D1.dietary = D2.dietary

    AND D1.DID <> D2.DID

) as DietaryPairs;
```

Part (c) [1 mark]

Suppose that Dish has these values. What will be the output of the query?

```
DID |      name       |  dietary
----+-----------------+----------
 1  | 'veggie burger' | 'veg'
 2  |     'salad'     | 'veg'
 3  |    'lasagna'    | 'normal'
 4  |    'quinoa'     | 'gf'
 5  |   'pistachio'   | 'gf'
```

Part (d) [1 mark]

Generalizing to any dataset, explain what is wrong with the output of this query.

Part (e) [1 mark]

Fix the query by making the smallest change that you can. Write your corrections directly on the query text above.

Question 5. [8 MARKS]

Suppose we have the following tables from a Twitter database:

```
Follows:
     a     |       b
-----------+----------------
 sina      | kanyewest
 sina      | RonConwayFacts
 diane     | LilaFontes
 diane     | swcarpentry
 diane     | mfeathers
 diane     | sina
 michelle  | sina
 michelle  | diane
 michelle  | Jeff
(9 rows)
```

```
Profile:
    id     |    name     | location
-----------+-------------+-----------
 alan      | catman      | Ottawa
 sina      | superman    |
 diane     | superwoman  | Toronto
 michelle  | rockstar    | Montreal
(4 rows)
```

```
Tweets:
id   | userid |    content
-----+--------+----------------
 123 | alan   | hellow twitter
 125 | alan   | bye twitter
 126 | alan   | hellow twitter
 128 | alan   | bye twitter
 476 | sina   | hellow twitter
 553 | diane  | hellow 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.

```
SELECT a, count(*)
FROM Profile RIGHT JOIN Follows
ON a = id GROUP BY a;
```

```
SELECT P.id, count(Follows.b) AS followers
From Profile P Join Follows
  On P.ID=Follows.b
Group by(P.ID) HAVING count(Follows.b) > 1;
```

Here are the tables again, for easy reference:

```
Follows:
     a     |       b
-----------+----------------
 sina      | kanyewest
 sina      | RonConwayFacts
 diane     | LilaFontes
 diane     | swcarpentry
 diane     | mfeathers
 diane     | sina
 michelle  | sina
 michelle  | diane
 michelle  | Jeff
(9 rows)
```

```
Profile:
    id     |    name    | location
-----------+------------+----------
 alan      | catman     | Ottawa
 sina      | superman   |
 diane     | superwoman | Toronto
 michelle  | rockstar   | Montreal
(4 rows)
```

```
Tweets:
id   | userid |    content
-----+--------+----------------
 123 | alan   | hellow twitter
 125 | alan   | bye twitter
 126 | alan   | hellow twitter
 128 | alan   | bye twitter
 476 | sina   | hellow twitter
 553 | diane  | hellow 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.

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

```
SELECT Tweets.content
FROM Tweets JOIN Profile
    On Tweets.userid = Profile.ID
    AND Profile.location IN
        (SELECT location FROM Profile
        WHERE name='catman');
```

[Use the space below for rough work. This page will not be marked unless you clearly indicate the part of your work that you want us to mark.]

[Use the space below for rough work. This page will not be marked unless you clearly indicate the part of your work that you want us to mark.]