CSC 343H1S 2013 Test 1
Duration — 50 minutes
Aids allowed: none

Student Number: |___|___|___|___|___|___|___|___|___|___|

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

Lecture Section: Day   **Instructor: Horton**

Please fill out the identification section above as well as the one on the back page,
and read the instructions below.

Good Luck!

This quiz consists of 5 questions on 8 pages (including this one). When you receive the signal to start, please make sure that your copy of the quiz is complete.

If you use any space for rough work or need to scratch out an answer, circle the part that you want us to mark.

1: _____/ 7

2: _____/ 6

3: _____/ 4

4: _____/ 4

5: _____/ 5

TOTAL: _____/26

Question 1.   [7 MARKS]

Suppose we have these relations: M($\underline{a}$, b), N($\underline{c, d, e}$), and O($\underline{m}$, g) and these constraints:

1. N[c] $\subseteq$ M[a]

2. N[e] $\subseteq$ O[g]

Part (a)   [3 MARKS]

Draw a valid instance of this schema, with three tuples in each relation. Draw the relations in table form and include column headings.

Part (b)   [1 MARK]

Name one attribute (from any relation) that is a foreign key:

Part (c)   [3 MARKS]

Express the following constraint using the $R = \emptyset$ notation we used on the assignment: No value can occur as a b in relation M and as an e in relation N and as a g in relation O. In other words, no value can be in all three columns.

Question 2.   [6 MARKS]

We used the following schema many times in lecture:

Relations

- Students(<u>SID</u>, surName, campus)

- Courses(<u>CID</u>, cName, WR)

- Offerings(<u>OID</u>, CID, term, instructor)

- Took(<u>SID, OID</u>, grade)

Integrity constraints

- Offerings[CID] $\subseteq$ Courses[CID]

- Took[SID] $\subseteq$ Students[SID]

- Took[OID] $\subseteq$ Offerings[OID]

Part (a)   [6 MARKS]

Write a query in relational algebra to report the instructors who have taught all the writing requirement courses that Pitt has taught.

Use only the basic operators we used in class and on the assignment: $\Pi, \sigma, \bowtie, \times, \cap, \cup, -, \rho$.

Question 3.   [4 marks]

This question uses the Offerings table from the courses schema of question 2.

Suppose we were interested in who were the first people to teach courses. The following query is intended to find the instructor or instructors who have the lowest value for term. (There are very likely ties.) The query is syntactically well-formed, but does not produce the correct answer in all cases.

$$NotMin(instructor) := \Pi_{O1.instructor}\sigma_{O1.term>O2.term}(\rho_{O1} Offerings \times \rho_{O2} Offerings)$$

$$Answer(instructor) := (\Pi_{instructor} Score) - NotMin$$

Below you will show that the query is incorrect by giving a valid instance of the table Offerings, with at most five tuples, on which the query would give the wrong result.

1. Valid instance of the table Offerings (with at most 5 tuples):

2. The correct answer for this instance:

3. The result that the query given above would produce for this instance:

Question 4. [4 marks]

Suppose the tables One and Two have the following contents:

One	x	y
	2	5
	1	6
	1	6
	1	6
	4	8
	4	8

Two	x	y
	2	55
	1	1
	4	4
	1	6
	4	8
	4	8
	9	9
	1	6

For each query below, draw the table that results. If the query would give an error, write "Error" instead, and explain what the error is.

```
(SELECT * FROM One)
    EXCEPT
(SELECT * FROM Two);
```

```
(SELECT * FROM One)
    EXCEPT ALL
(SELECT * FROM Two);
```

```
(SELECT * FROM One)
    INTERSECT
(SELECT * FROM Two);
```

```
(SELECT * FROM One)
    INTERSECT ALL
(SELECT * FROM Two);
```

Question 5.   [5 marks]

Here's a portion of the schema for the IMDb database we have used extensively in class:

```
CREATE TABLE people (
  person_id integer primary key,
  name varchar NOT NULL UNIQUE
) ;

CREATE TABLE movies (
  movie_id integer primary key,
  title varchar NOT NULL,
  year integer NOT NULL,
  rating float NOT NULL,
  UNIQUE (title,year)
) ;

CREATE TABLE roles (
  person_id integer NOT NULL references people,
  movie_id integer NOT NULL references movies,
  role varchar NOT NULL,
  PRIMARY KEY (person_id,movie_id,role)
) ;
```

Complete the WHERE clause in the following query so that you find all people who've had a role in a Star Wars movie. Use `LIKE '%Star Wars%'` to identify the Star Wars movies.

```
SELECT *
FROM people
WHERE .  .  .
```

[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.]

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