UNIVERSITY OF WATERLOO

Final Examination

Term: 1155 **Year:** 2015

Student name: _____ _____ _____
(First name) (Middle name) (Last name)

Waterloo student identification number: _____

Course abbreviation and number: MSCI 346

Course title: Database Systems

Section(s): 001

Sections combined course(s): _____

Section numbers of combined course(s): _____

Name of instructor(s): Gunes Aluc

Date of exam: Monday, August 10th

Exam period start time: 9:00am Exam period end time: 11:30am

Duration of exam: 2.5 hours

Number of exam pages: (includes cover page) _____

Exam type: (select one) ✖ Closed book ☐ Special materials ☐ Open book

Materials allowed: (select one)

✖ No additional materials are allowed

☐ Materials allowed are listed below

_____

_____

_____

Exams are printed double sided on white paper.

☐ Select this box if second side of paper is to be used for rough work calculations.

Marking scheme:

Question	Score	Question	Score
1		6	
2		7	
3		8	
4		9	
5		10	

Student ID:

MSCI 346: Database Systems

Spring 2015
Final Exam

1. There are 19 pages in this booklet, including all cover pages. Please make sure that you have no missing pages.
2. This exam consists of 10 questions.
3. You have 150 minutes to answer the questions in this exam.
4. During the exam, you **cannot** refer to any books or notes. However, an Appendix containing some useful material has been compiled for your convenience. You may find the Appendix on pages 18–19.
5. Print your first name, last name and student id on every page including the cover pages.
6. Good luck!

Student ID:

Question 1 (5 pts): What is *data independence*? What is *physical data independence*? How is physical data independence different from logical data independence?

Question 2 (5 pts): A database management system (DBMS) guarantees that transactions have ACID properties. Briefly describe what each of the following terms means.

Atomicity:

Consistency:

Isolation:

Durability:

Student ID:

Question 3 (5 pts): Contrast *Online Transaction Processing* (OLTP) applications with *Online Analytic Processing* (OLAP) applications. List 5 differences.

Question 4 (5 pts): What is the *three-tier architecture*? Why is it important?

Student ID:

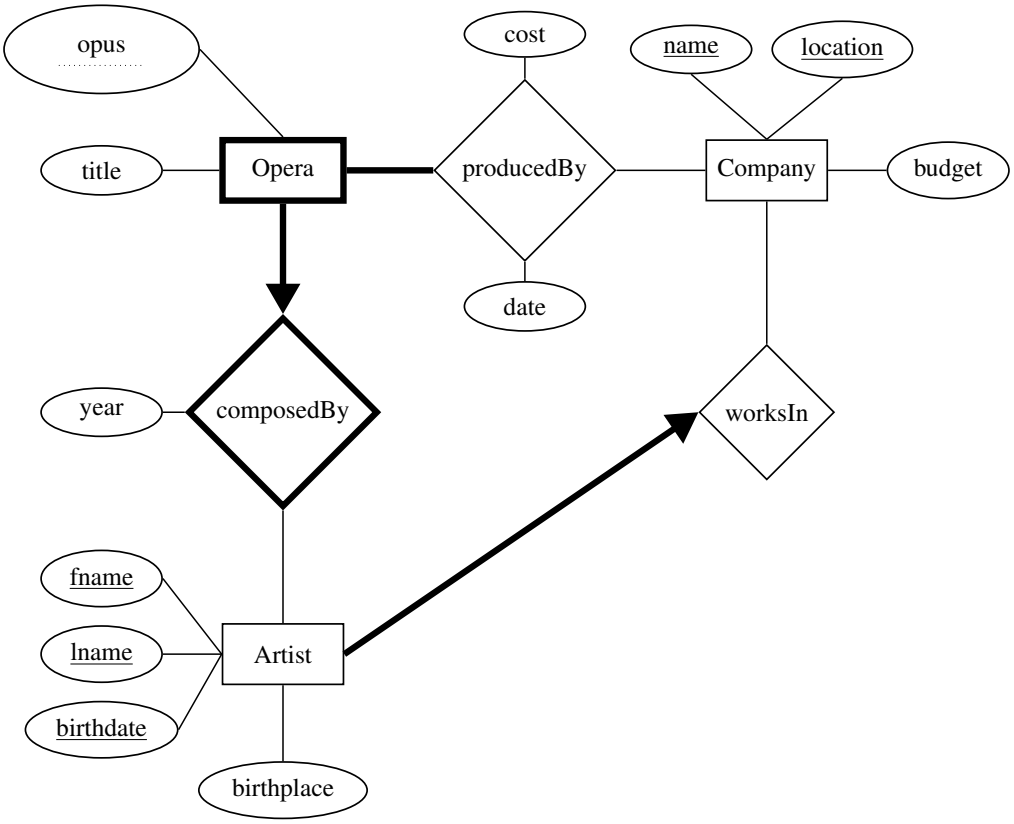

Figure 1: Entity-Relationship (ER) for Question 5

Question 5 (10 pts): Based on the Entity-Relationship (ER) diagram in Figure 1, indicate, for each statement below, whether the statement is true or false. Note that this question continues on the next page.

- Every *Artist* needs to work in a *Company*.
- An *Artist* can work in **at most one** *Company*.
- A minimum of 0 *Artists* can work in a *Company*.
- Every *Opera* is produced by a *Company*.
- A *Company* can produce multiple *Operas*.
- Two different *Operas* can have the same *opus* number.
- Two different *Artists* can have the same first name, last name and birthdate as long as their place of birth is different.

Student ID:

- It is possible to represent the *Artist* entity and the *worksIn* relationship using only a single relation in the relational model.
- An *Opera* can be composed by **more than one** *Artist*.
- A *Company* can produce two different *Operas* on the same date.

Student ID:

Question 6 (10 pts): Consider a database that is created using the following two CREATE TABLE statements.

```
CREATE TABLE Students (
    fname    VARCHAR(25),
    lname    VARCHAR(25),
    dept     VARCHAR(25)
               NOT NULL,
    PRIMARY KEY (fname, lname)
);
```

```
CREATE TABLE Projects (
    pid          INTEGER,
    sup_fname    VARCHAR(25),
    sup_lname    VARCHAR(25),
    PRIMARY KEY (pid),
    FOREIGN KEY (sup_fname, sup_lname)
      REFERENCES Students(fname, lname)
);
```

Then, assume that the following commands are executed to populate the database. What is the content of the two tables *Students* and *Projects* after these commands are executed?

```
BEGIN;
    INSERT INTO Students(fname, lname, dept)
        VALUES ('Alice', 'Shoemaker', 'MSCI');
COMMIT;
BEGIN;
    INSERT INTO Students(fname, lname, dept)
        VALUES ('Alice', 'Brown', 'MSCI');
    INSERT INTO Students(fname, lname, dept)
        VALUES ('Bob', 'Brown', NULL);
COMMIT;
BEGIN;
    INSERT INTO Students(fname, lname, dept)
        VALUES ('Alice', 'Shoemaker', 'CS');
COMMIT;
BEGIN;
    INSERT INTO Projects(pid, sup_fname, sup_lname)
        VALUES (1, 'Alice', 'Shoemaker');
    INSERT INTO Projects(pid, sup_fname, sup_lname)
        VALUES (2, NULL, NULL);
COMMIT;
```

Student ID: _____

Provide your response to Question 6 on this page.

Student ID:

uid	*name*	*location*
1	Alice	Waterloo
2	Bob	Toronto
3	Carol	Toronto
4	Dave	Kitchener

(a) **Users**

follower	*followee*
1	2
1	3
1	4
2	1
3	4

(b) **Follows**

tid	*uid*	*text*
t1	1	"What a boring lecture today! #msci346 #boring"
t2	1	"Thank you #w3schools for saving my day!"
t3	1	"Yes, they are @bob. #msci346 #grades"
t4	2	"Wait, are the grades posted? #msci346"
t5	2	"Lol, questions 1 & 2 are from the midterm. #msci346"

(c) **Tweets**

Figure 3: Database Instance

Question 7 (20 pts): This question consists of 5 parts: 7.a–7e. Consider the hypothetical relational database instance in Figure 3. What will each of the following SQL queries return if executed over the database shown in Figure 3? Explicitly show the contents of the tuples that will be returned.

7.a (4 pts):

SELECT U.name, T.tid
FROM Users U, Tweets T
WHERE U.location = 'Toronto' *AND*
T.tid = 't5'

Student ID: _____

7.b (4 pts):

```
SELECT    U.name, T.tid
FROM      Users U, Tweets T
WHERE     U.location = 'Toronto' AND
          T.tid = 't5' AND
          U.uid = T.uid
```

7.c (4 pts):

```
SELECT    U.uid, U.name, COUNT(*)
FROM      Users U, Follows F
WHERE     U.uid = F.followee
GROUP BY  U.uid, U.name
```

Student ID:

7.d (4 pts):

```
SELECT   U.uid, U.name
FROM     Users U
WHERE    U.uid NOT IN
         ( SELECT    T.uid
         FROM        Tweets T
         GROUP BY    T.uid
         HAVING      COUNT(*) ≥ 3 )
```

7.e (4 pts):

```
SELECT   U.uid, U.name
FROM     Users U
WHERE    EXISTS
         ( SELECT    *
         FROM        Follows F
         WHERE       U.uid = F.follower AND
                     EXISTS
                     ( SELECT    *
                     FROM        Tweets T
                     WHERE       F.followee = T.uid AND
                                 T.text LIKE '%boring%' ) )
```

Student ID:

Question 8 (15 pts): This question consists of 3 parts: 8.a–8.c. Consider the database in Figure 3 (page 9). For each question below, write a SQL query that returns the data that are requested.

8.a (5 pts): Write a SQL query that returns the texts of all tweets made by 'Alice'.

8.b (5 pts): Write a SQL query that returns for each user, the number of followers of that user.

Student ID:

8.c (5 pts): Write a SQL query that returns the *uid* and *name* of every user who does not have any followers *and* who has tweeted at least once. (You are allowed to use EXCEPT, INTERSECT or UNION even though they are not supported by MySQL.)

Student ID:

Question 9 (10 pts): Consider the schedule in Figure 4 where T_1 and T_2 denote two transactions, and X and Y denote two database objects. List all pairs of conflicting actions. Based on the set of conflicting actions, draw the serializability graph for this schedule. Is the serializability graph cyclic or acyclic? Then, is the schedule serializable? Is the schedule recoverable? Why (or why not)? If the orders of commit operations are reversed, is the resulting schedule recoverable?

$\mathbf{T_1}$	$\mathbf{T_2}$
$W_1(X)$	
	$R_2(X)$
$W_1(Y)$	
	$W_2(Y)$
$R_1(X)$	
$R_1(Y)$	
	$W_2(X)$
	$R_2(Y)$
commit	
	commit

Figure 4: Schedule

Student ID:

Question 10 (15 pts): Consider a relation with five attributes: ABCDE. Assume that $AB \rightarrow C$ and $D \rightarrow E$ are the only functional dependencies that hold over this relation. Is this relation in Boyce-Codd Normal Form (BCNF)? If not, decompose the relation(s) until you obtain a database in which all relations are in BCNF. Make sure that the decompositions are lossless-join decompositions; however, you do not need to worry about preserving dependencies. Show all steps of your work.

Student ID: _____

This page is intentionally left blank for your convenience.

Student ID: _____

This page is intentionally left blank for your convenience.

Student ID:

Appendix

Let F denote a set of functional dependencies, and let X denote a set of attributes. Then, the **attribute closure** of X can be computed using the following algorithm.

function $ComputeX^+(X, F)$
begin
  X^+ := X;
  while true do
    if there exists $(Y \rightarrow Z) \in F$ such that
      (1) $Y \subseteq X^+$, and
      (2) $Z \not\subseteq X^+$
    then X^+ := $X^+ \cup Z$
    else exit;
  return X^+;
end

Let R be a relation schema and F a set of functional dependencies. Schema R is in **BCNF** (w.r.t. F) if and only if
whenever $(X \rightarrow Y) \in F^+$ and $XY \subseteq R$, then either

- $(X \rightarrow Y)$ is trivial (i.e., $Y \subseteq X$), or
- X is a superkey of R

A database schema $\{R_1, \ldots, R_n\}$ is in BCNF if each relation schema R_i is in BCNF.

Appendix

Let R be a relation schema and F a set of functional dependencies. The Lossless-Join Decomposition algorithm is given below:

function $DecomposeBCNF(R, F)$
begin
  Result := $\{R\}$;
  while some $R_i \in$ Result and $(X \rightarrow Y) \in F^+$
      violate the BCNF condition **do begin**
    Replace R_i by $R_i - (Y - X)$;
    Add $\{X, Y\}$ to Result;
  end;
  return Result;
end