Carleton UNIVERSITY

FINAL EXAMINATION
Fall 2020

DURATION: 2 hours (120 minutes) **No. of Students: 38**

Department Name & Course Number: **Systems and Computer Engineering**
SYSC 5703 – Integrated Database and Cloud Systems

Course Instructor: **Samuel A. Ajila, PhD., P.Eng.**

AUTHORIZED MEMORADA: Online Exam

Students MUST count the number of pages in this examination question paper before beginning to write, and report any discrepancy to a proctor. This question paper has 8 pages + cover page = 9 pages in all.

This examination question paper may not be taken from the examination room.
In addition to this question paper, students require: Examination booklets yes no ☑
Scantron Sheet yes no ☑

Name: _____

Student Number: _____

Instructions

i. **Total marks = 100**
ii. Answer all questions. This is an online fixed time exam
iii. Use the space provided after each question or use a plain sheet to answer the questions.
iv. **The exam duration is two (2) hours. Upload your exam to cuLearn at the end of the exam duration**.
v. **I WILL NOT accept any submission by email [to me]. ANY submission by email will not be marked.**
vi. **You will have access to the exam paper at most 10 minutes before the start of the exam and 20 minutes maximum will be provided at end of the exam for scanning, combining, and upload to cuLearn.**
vii. **NOTE THAT the 10 minutes before and the 20 minutes after are NOT part of the official exam duration!**
viii. **This final exam should be written by the individual named above. All exam papers will be checked for plagiarism. Any two or more identical (or too similar) submissions will be graded zero (0) and reported to the University authority**.
ix. **This final exam paper is carefully prepared and cross checked. You may ask questions [by email] if you believed you have found a mistake in the exam paper. If there is a mistake, the correction will be announced to the entire class by email. If there is no mistake, this will be confirmed, but no additional explanation will be provided.**

Questions	Marks	Scores
1	18	
2	12	
3	14	
4	12	
5	26	
6	18	
Total	100	

Question 1 [3+2+8+5 = 18 marks]
Consider the following database schema. The attributes are ABCDEGHKLM (10 in total). The FDs are:
1. ABE → CK
2. AB → D
3. C → BE
4. EG → DHK
5. D → L
6. DL → EK
7. KL → DM

i. Compute the attribute closure of EGL with respect to the above set of FDs. Show all steps.

ii. Is EGL a key of the schema? Why?

iii. Assuming after using the 3NF synthesis algorithm to construct a lossless, dependency preserving decomposition, we obtain the five schemas below. **Note that these schemas are 3NF lossless and dependency preserving.**

R1 = (ABCD; {AB → CD})
R2 = (CBE; {C → BE})
R3 = (EGDH; {EG → DH})
R4 = (DLEK; {D → LEK})
R5 = (KLDM; {KL → DM})

a. Are all the schemas in the resulting decomposition (i.e. R1, R2, R3, R4, and R5) in BCNF? If there are schemas that are not in BCNF, identify and decompose them further to achieve BCNF.

b. Is the resulting decomposition dependency-preserving? Explain!

Question 2 [4 + (2 + 3.5 + 2.5) = 12 marks]

a. Let $R(x, y, z)$ be a relation. Write one or more Datalog rules that define $\sigma_c(R)$, where c equals to **x < y OR y < z**

b. For each of the following Datalog rules, write an expression of relational algebra that defines the same relation as the head of the rule:

i. **P**(x, y) ← **Q**(x, z) AND **R**(z, y)

ii. **P(**x, y) ← **Q**(x, z) AND **Q**(z, y)

iii. **P**(x, y) ← **Q**(x, z) AND **R**(z, y) AND x < y

Question 3 [5+5+4 = 14 marks]

Professor (Id: INTEGER, Name: STRING, DeptId: DEPTS)

Given the schema above, our goal is to retrieve the name of all professors in a particular department from the Professor table. The **department** will be supplied at run time. Assume a connection object, *con*, has been created.

i. Write the JDBC statement used to create a prepared statement object *pre* for this query. If the statement involves a host variable, give its value at the time the JDBC statement is executed.

ii. Suppose we want to get the names of all professors in the CS Department using the prepared statement object *pre*. What JDBC statements must be executed to put these names in a results set, *res*.

iii. What JDBC statement must be executed to fetch the first name from *res*?

Question 4 [12 marks]

Email spam filtering models often use a **bag-of-words** representation for emails. The table below lists the bag-of-words representation for the following five emails and a target feature, SPAM, whether they are spam emails or genuine emails:

"money, money, money"
"free money for free gambling fun"
"gambling for fun"
"machine learning for fun, fun, fun"
"free machine learning"

ID	Bag-of-Words							SPAM
	MONEY	FREE	FOR	GAMBLING	FUN	MACHINE	LEARNING	
1	3	0	0	0	0	0	0	true
2	1	2	1	1	1	0	0	true
3	0	0	1	1	1	0	0	true
4	0	0	1	0	3	1	1	false
5	0	1	0	0	0	1	1	false

What target level would a nearest neighbor model using **Manhattan distance** return for the following email: "*machine learning for free*"?

The bag-of-words representation for this query is as follows:

ID	Bag-of-Words							SPAM
	MONEY	FREE	FOR	GAMBLING	FUN	MACHINE	LEARNING	
Query	0	1	1	0	0	1	1	?

Use the table below for the calculation of the **Manhattan distance** between the query instance and each of the instances in the training dataset.

ID	Money	Free	For	Gambling	Fun	Machine	Learning	Manhattan Distance
1	3	0	0	0	0	0	0	
2	1	2	1	1	1	0	0	
3	0	0	1	1	1	0	0	
4	0	0	1	0	3	1	1	
5	0	1	0	0	0	1	1	

What is your conclusion?

Question 5 [3 + 8 + 3 + 8 + 4 = 26 marks]

a. A convicted criminal who reoffends after release is known as a **recidivist**. The table below lists a dataset that describes prisoners released on parole, and whether they reoffended within two years of release.

ID	GOOD BEHAVIOR	AGE < 30	DRUG DEPENDENT	RECIDIVIST
1	false	true	false	true
2	false	false	false	false
3	false	true	false	true
4	true	false	false	false
5	true	false	true	true
6	true	false	false	false

This dataset lists six instances where prisoners were granted parole. Each of these instances are described in terms of three binary descriptive features (GOOD BEHAVIOR, AGE < 30, DRUG DEPENDENT) and a binary target feature, RECIDIVIST. The GOOD BEHAVIOR feature has a value of *true* if the prisoner had not committed any infringements during incarceration, the AGE < 30 has a value of *true* if the prisoner was under 30 years of age when granted parole, and the DRUG DEPENDENT feature is *true* if the prisoner had a drug addiction at the time of parole. The target feature, RECIDIVIST, has a *true* value if the prisoner was arrested within two years of being released; otherwise it has a value of *false*.

i. Calculate the **entropy** (in bits) for the dataset (i.e. RECIDIVIST)

ii. Now, using the **entropy in (i)** and the table below calculate (to 4 decimal places) the partition entropy, remainder, and information gain for GOOD BEHAVIOR, AGE < 30, and DRUG DEPENDENT. **Please, fill the table clearly and legibly.**

Split by Feature	Level	Partition	Instances	Partition Entropy	Rem.	Info. Gain
GOOD BEHAVIOR	true	D1				
	false	D2				
AGE < 30	true	D3				
	false	D4				
DRUG DEPENDENT	true	D5				
	false	D6				

iii. Calculate the **Gini Index** for the dataset (i.e. RECIDIVIST)

iv. Now, using the **Gini Index in (iii)** and the table below calculate (to 4 decimal places) the partition Gini index, remainder, and information gain for GOOD BEHAVIOR, AGE < 30, and DRUG DEPENDENT.

Split by Feature	Level	Partition	Instances	Partition Gini Index	Rem.	Info. Gain
GOOD BEHAVIOR	true	D1				
	false	D2				
AGE < 30	true	D3				
	false	D4				
DRUG DEPENDENT	true	D5				
	false	D6				

v. Compare the Information gain in (ii) and (iv). What is your conclusion? **Two sentences maximum.**

Question 6 [(4 + 5 + 4) + 4 = 18 marks]

Consider the following ODL definitions:

```
class PERSON: OBJECT (extent PERSONEXT): PERSISTENT;
{attribute String Name;
... ... ... }
class STUDENT extends PERSON (extent STUDENTEXT): PERSISTENT;
{attribute Set< TRANSCRIPTRECORD> Transcript;
... ... ... }
struct TRANSCRIPTRECORD {
        String CrsCode;
        float Grade;
        String Semester;}
```

a. Write the following query in OQL (not SQL): "*List all students with their corresponding average grade*."

   i. Using nested query **without** GROUP BY

   ii. **without** the GROUP BY clause and no nesting

iii. **with** the GROUP BY clause but no nesting

b. Given the ODMG ODL (below) that a person [possibly] has a spouse –

```
Class PERSON {
        attribute Integer Id;
        attribute String Name;
        relationship PERSON Spouse;
                        ...         }
```

Write an OQL (not SQL) query that returns the name of a particular person's spouse.