# Computer Science 3005B
# 2019 Fall
# Midterm Examination

DATE: October 29, 2019
TIME: 75 minutes
INSTRUCTOR: Dr. M. Liu

NAME: Khaled

Instructions:

1. This is a closed book exam, with no additional material permitted.
2. Verify that your examination consists of 5 pages including this one.
3. Answer all questions in the spaces provided.
4. Please write as legibly as possible. Anything that cannot be read will be marked 0.
5. Total marks for the exam is **80** and you have **75** minutes. Allocate your time wisely.

**Part 1** (30 marks)
Explain the following terms as complete as possible. Simply give full name is not acceptable. Each question is 3 marks.

1. DBMS: Software to facilitate the creation and maintenance of a computerized database.

2. Key: **minimal** set of attributes that **uniquely identifies** that tuples in the relation

3. End User: person who **uses** the database on a day to day basis for queries, reports and updates, **knows** nothing about the structure of the database

4. Foreign Key: **attributes** that **reference the primary key** of the same or different relation

5. SQL: **standard** database language that combines **data definition language**, **data manipulation language** and **query language**.

**6.** Declarative language: a language that specifies what to do instead of how to do**.**

7. Entity Integrity Constraint: The primary key attributes of each relation cannot have null values in any tuple in the relation

8. Data independence: allow changing data structure without having to change the accessing program.

9. DBA: a person to control the DB **availability**, **performance**, **security**, **backup**, and **recovery**.

10. Relation: consists of **schema** and relational **instance**. The schema consists of the **relation name** and all **its attributes** and the instance is a **set of tuples**.

**Part 2** (50 marks)
Consider the following NHL Fans database with three tables Person, Team, Fans that represent persons, NHL teams, and NHL Fans respectively. The primary keys are underlined and foreign keys are obvious. Use relational algebra (ALG), tuple calculus (TRC), and domain calculus (DRC) to express the following queries. Each query is 10 marks: 3 marks for each method and 1 mark for the results.

**Person**

| P# | Name | Age | City |
|---|---|---|---|
| P1 | Smith | 20 | Ottawa |
| P2 | Jones | 30 | Toronto |
| P3 | Blake | 25 | Calgary |
| P4 | Clark | 20 | Montreal |
| P5 | Adams | 30 | Vancouver |

**Team**

| T# | Name | City |
|---|---|---|
| T1 | Maple Leafs | Toronto |
| T2 | Canucks | Vancouver |
| T3 | Canadiens | Montreal |
| T4 | Flames | Calgary |

**Fan**

| P# | T# |
|---|---|
| P1 | T1 |
| P1 | T2 |
| P1 | T3 |
| P1 | T4 |
| P2 | T1 |
| P2 | T2 |
| P2 | T3 |
| P3 | T1 |
| P3 | T2 |
| P4 | T1 |

1. Get the names of persons who like the NHL team in Toronto.

ALG
T1: = project T# (select city = `Toronto’ (NHL)) ;
T2: = project P#,Name (Person);
T3:= T1 njoin Fans njoin T2;
Project Name (T3);

TRC
{P.name | P in Person and exists (F in Fans, N in NHL)(N.T#=F.T# and F.P#=P.P# and
N.City =’Toronto’}

DRC
{N | (exists T#,P#)(Person(P#,N,_,_) and NHL(T#,_, ‘Toronto’) and Fans(P#,T#,_))}

| Name |
|---|
| Smith |
| Jones |
| Blake |
| Clark |

2. Get the names of persons who are not fans of NHL teams.

   ALG
   T1:= project Name (Person);
   T2:= project Name (Person njoin Fans);
   T1 minus T2;

   TRC

   {P.Name | P in Person and not (exists F in Fans) (P.P# = F.P#)};

   DRC

   {Name | (exists P#)(Person (P#, Name, _, _) and  not (exists T#) ( Fans (P#,T#)))};

   Name
   Adams

3. Get names of persons who are fans of all NHL teams.

   ALG
   T1:= project T# (NHL);
   T2:= Fan divideby T1;
   T3:= Person njoin T2;
   project name (T3);

   TRC
   {P.Name | P in Person and (forall N in NHL) (exists F in Fans)(P.P# =F.P# and F.T# = N.T#)};

   DRC
   {Name | (exists P#)(Person (P#, Name, _, _) and (forall T#)( Team (T#,_,_) and Fans(P#,T#)))};

   Name
   Smith

4. Get the names of persons who are fans of all teams except Flames.

   ALG
   T1:= project T# (select name !=  'Flames' (Team));
   T2:= Fan divideby T1;
   T3:= select name = 'Flames' (Team)
   T4:= project P# (Fan njoin T3) )
   T5:= T2 minus T4;
   project name (Person njoin T5);

   TRC
   {P.Name | P in Person and (forall N in NHL)
   (N.Name = 'Flames' and not (exists F in Fans) (P.P# = F.P# and F.T# = N.T#))
   or
   (N.Name != 'Flames' and (exists F in Fans) (P.P# = F.P# and F.T# = N.T#))};

DRC

```
{Name | (exists P#)(Person (P#, Name, _, _) and (forall T#)(exists Tname)(Team(T#,Tname , _)
    and
    (Tname = ‘Flames’) and not Fans (P#, T#)
    or
    (Tname != ‘Flames’) and Fans (P#, T#)))};
```

| Name |
|---|
| Jones |

5. Get the names of persons who are fans of more than one team.

ALG

```
T1:= Person njoin Fans                              (1)
T2(Name, Count) := aggregate name, count(*)(T1);    (1)
project name (select Count > 1 (T2));               (1)
```

TRC

```
{P.name | P in Person and (exists F1, F2 in Fans)(P.P# = F1.P# and P.P#=F2.P#  and F1 != F2)};
```

```
T(Name, Count) := {P.name , count(*) | P in Person and (exists F in Fans)(P.P#=F.P#)};
{P.name | P in T and P.Count > 1};
```

should use one query instead of two, deduct 2 marks for using this method

DRC

```
{N | (exists P, T1, T2)(Person(P,N,_,_) and Fans(P,T1,_) and Fans(P,T2,_) and T1 !=T2)))};
```

```
T(Name, Count) := {N, Count(*)|(exists P,T)(Person(P, N, _) and Fans(P, T, _))};
{Name | (exists C)(T(Name, C) and C > 1);
```

should use one query instead of two, deduct 2 marks for using this method

| Name |
|---|
| Smith |
| Jones |
| Blake |