UNIVERSITY OF TORONTO SCARBOROUGH

**Final Exam**
**CSCC43**
**Introduction to Database**
**Fall 2017**
**Aid: None**
**Duration: 120 minutes**
**Professor: Marzieh Ahmadzadeh**

**Student Name: ……** **Student ID: ……..**

**UTORID: ……..**

**Mark:**

| | | | |
|---|---|---|---|
| **Part 1** | **Concepts** | | **/15** |
| **Part 2** | **EER -> relation** | | **/10** |
| **Part 3** | **Normalization & Relational Algebra** | | **/20** |
| **Part 4** | **SQL: Output** | | **/25** |
| **Part 5** | **SQL: Query Writing** | | **/30** |
| **Total** | | | **/ 100** |

**Please read the followings before you start writing.**

- **Do not start unless you are told to do so.**
- **This exam consists of 5 parts on 12 pages including this page.**
- **No question will be answered in last 15 minutes of the exam.**
- **Write the answers neatly. If your answer is not readable, no mark will be awarded.**
- **This exam is a closed book exam therefore NO aid including textbook, handout etc. are allowed.**
- **Mobile phone or any other electronic device is not allowed in this exam. Make sure you have turned them off and bring it to the front of the class.**
- **The University of Toronto's Code of Behaviour on Academic Matters prohibits cheating, and the use of unauthorized aids. Students violating the Code may be subject to penalties such as suspension or expulsion from the University.**

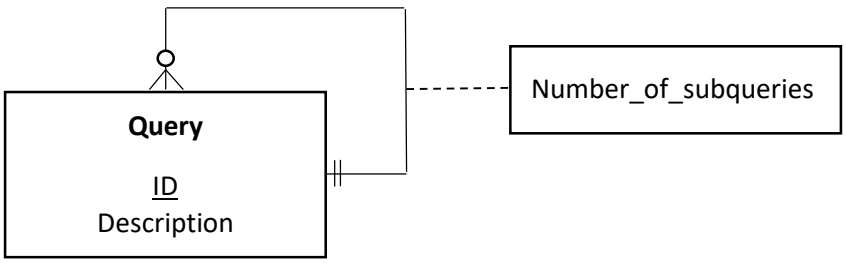
**GOOD LUCK**

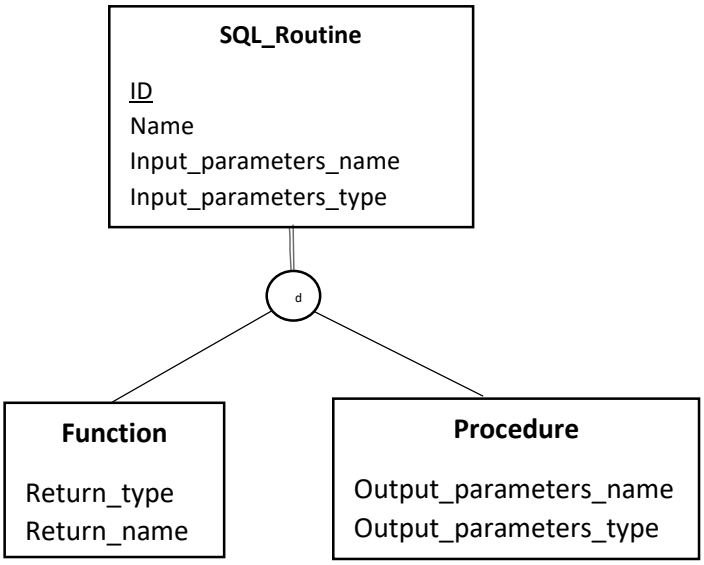
**Part 1**: Answer the following questions briefly: (15 Marks)

| |
|---|
| What is a sub-type discriminator? |
| A relation is in which normal form, if and only if every determinant is a candidate key? |
| We have a table in which we store the information about the shares that are exchanged in a stock market including number of exchanged shares, price, buyer/seller information etc. Every hour, a business analyst hired by a company called A, is responsible to query this table, which contains millions of records of data, to find the exchange price for company A in last 24 hours. Since there are a large number of data in this table, it takes more than a minute for this query to run. How do you help this business analyst to efficiently run this query? |
| Explain how can blind SQL injection attack be useful for an attacker? |
| Using tautology type of attack, when you require to input your debit card number in order to see your balance, what would you input to see the balance of every single client of this bank? As a developer how do you sanitize your input data for this specific type of attack? |

**This page is to be used for your rough work. Nothing would be marked here unless you clearly make a reference to the part that you want it to be marked.**

**Part 2**: Map each EER diagram to corresponding relation(s) using short hand notation. Keys should be specifically indicated. (10 Marks)



**This page is to be used for your rough work. Nothing would be marked here unless you clearly make a reference to the part that you want it to be marked.**

**Part 3**: (20 Marks)

Find the dependencies between attributes and state the type of each dependency. Convert the following table to third normal form and present the result in shorthand notation. Keys should be specifically indicated.

| MovieID | MovieName | Genre | DirectorID | DirectorName | ProducerID | ProducerName | CastID | Cast |
|---|---|---|---|---|---|---|---|---|
| M100 | Friends | SitCome | D100 ,<br><br>D200 | Gary Halvorson, Kevin Bright | P100, P200, P300 | Kevin Bright, David Crane, Marta kaufmman | A100,<br><br>A200, A300, A400, A500, A600 | David Schwimmer, Matt LeBlanc, Lisa Kudrow, Matthew Perry, Jennifer Aniston, Courteney Cox |
| M200 | *Annabelle* | horror | D300 | John Leonetti | P400<br>P500 | Peter Safran, James Wan | A700, A800, A900 | Annabelle Wallis, Ward Horton, Alfre Woodard |

Convert the following SQL statement into its equivalent *relational algebra*.

```
select convention_ID, convention_name, discount_rate
from Convention , discount
where convention.convention_ID = discount.convention_ID
```

**This page is to be used for your rough work. Nothing would be marked here unless you clearly make a reference to the part that you want it to be marked.**

**Part 4**: What would be the output of running the following SQL code snippet? (25 Marks)

```
create table person (
    id char(4) primary key,
    s_name varchar(40),
    post_code char(4));

insert into person values ("1000", "John", "x132");
insert into person values ("2000", "Rose", "m136");
insert into person values ("3000", "Mike", "d565");
insert into person values ("4000", "Tresa", "l987");

create table experience (
    id char(4),
    title varchar (20),
    year_practiced int(2),
    constraint pk primary key (id, title),
    constraint fk foreign key (id) references
    person(id) );

insert into experience values ("1000", "programmer", 3);
insert into experience values ("1000", "database
developer", 3);
insert into experience values ("2000", "programmer", 10);

select s_name from person, experience
where person.id = experience.id and title =
"programmer";



set autocommit = 0;
insert into experience values ("1000", "Software Engineer",
5);
rollback;
insert into experience values ("1000", "Project Manager",
5);
commit;
set autocommit = 1;
insert into experience values ("2000", "Project Manager",
2);
rollback;



select s_name from experience, person
where experience.id = person.id and
      title = "Project Manager";
```

```
insert into experience values ("3000", "database
developer", 5);
insert into experience values ("4000", "database
developer", 4);

Select title, count(title) title_cnt
from experience
where title not like "%manager"
group by title
having title_cnt > 1;

select s_name,title year_practiced
from experience, person
where experience.id = person.id and
      year_practiced =
         (select min(year_practiced)
          from experience
          where title = "database developer");

select p.s_name from person as p
left outer join person on p.id = person.id;

delete from experience;
select * from experience;

insert into experience values ("1000", "programmer", 3);
insert into experience values ("1000", "database
developer", 3);
insert into experience values ("2000", "programmer", 10);
insert into experience values ("1000", "Software Engineer",
5);
insert into experience values ("1000", "Project Manager",
5);
insert into experience values ("2000", "Project Manager",
2);
insert into experience values ("3000", "database
developer", 5);
insert into experience values ("4000", "database
developer", 4);
insert into experience values ("4000", "data analyst", 4);

select sum(year_practiced), title
from experience
group by title;
```

**This page is to be used for your rough work. Nothing would be marked here unless you clearly make a reference to the part that you want it to be marked.**

**Part 7**: Write <u>one</u> SQL command for the requested report. Only a small chunk of the original tables is shown below due to space limitation. (30 Marks)

Person

| ID | Name | PostCode |
|---|---|---|
| 1000 | John | x132 |
| 2000 | Rose | m136 |
| 3000 | Mike | d565 |
| 4000 | Tresa | l987 |

Resume

| ID | Title_code | Year_practiced |
|---|---|---|
| 1000 | A1 | 3 |
| 1000 | A2 | 5 |
| 1000 | A3 | 5 |
| 1000 | C1 | 3 |
| 2000 | A1 | 10 |
| 2000 | A3 | 2 |
| 3000 | C1 | 5 |
| 4000 | C1 | 4 |
| 4000 | C2 | 4 |

Job_title

| Title_code | Title |
|---|---|
| A1 | programmer |
| A2 | Software Engineer |
| A3 | Project Manager |
| C1 | database developer |
| C2 | data analyst |

A) Find the name of all programmers.

B) Find people who have the most experience in each category of job titles.

C) For the given category X (e.g. A), find all the people who have experienced all the jobs in this category (e.g. A1, A2, and A3).

**This page is to be used for your rough work. Nothing would be marked here unless you clearly make a reference to the part that you want it to be marked.**