The University of New South Wales

COMP3311 Database Systems Final Exam 20T1

Thursday 7 May 2020

1. Time Allowed: 3 hours
2. To be completed between:
   9:00am Thu 7 May 2020 - 9:00am Fri 8 May 2020 (AEST)
3. Total number of questions: 10
4. Total marks available: 50
5. Questions are not of equal value.
6. Marks are shown on each question.
7. **Carefully read the notes below before you start the exam.**

By starting this exam as a student of The University of New South Wales, you do solemnly and sincerely declare that you have not seen any part of this specific examination paper for the above course prior to attempting this exam, nor have any details of the exam's contents been communicated to me. In addition, you will not disclose to any University student any information contained in the abovementioned exam for a period of 24 hrs after the exam. Violation of this agreement is considered Academic Misconduct and penalties may apply.

By submitting the exam answers via WebCMS or give, you declare that **all of the work submitted for this exam is your own work, completed without assistance from anyone else**.

Please refer to the Student Conduct website for details.

Notes

1. General Instructions:

- Answer **all** questions.
- Questions are not worth equal marks.
- Questions may be answered in any order.
- You may create additional views to help formulating your queries, if needed, but you are not allowed to create any tables.
- During the 3-hr Exam, you must **not**:
  - access any of your own files
  - access any web pages except the standard documentation in this course.
- During the entire 24-hr period from 9:00am Thu 7 May 2020 (AEST), you must **not**:
  - communicate with other students in *any* way
- Your answers must be submitted using `give` or via WebCMS
- If you have any clarification questions between AEST 9:00am-5:00pm 7th May, 2020 (we may be unavailable outside this period), please email cs3311@cse.unsw.edu.au via an UNSW email account.

Please fill in your answers in the supplied template (`ans.sql`). The comments in the template file indicate where you can fill in the answers. If there are extra files (e.g., drawings) needed to be submitted, the corresponding questions will have instructions specified. SQL queries will be auto-marked by using sqlite3 installed on CSE linux machines, on a database with the same schema with data possibly modified. You can only receive marks for correctly-working queries that loads with no warnings.

2. Submission:

You can submit your exam solution either using:

i) WebCMS: Login to Course Web Site > Exam > Final Exam > Final Exam Paper > Make Submission > upload required files > [Submit]

Or:

ii) The give command: `give cs3311 exam ans.sql`

Required Files: `ans.sql`

Deadline: Friday 8 May 2020 at 9:00am (AEST)

No late submissions will be accepted.

3. Downloads:

Downloads: exam.tgz or exam.zip

Both `.tgz` and `.zip` files contain the same material.
Each archive contains the sqlite3 IMDB database dump `exam.db`, plus the answer template file called `ans.sql`

4. How to Start:

- read these notes carefully and completely
- download one of the archive files above
- unpack the downloaded file
- get familiar with the schema and data by exploring and querying the provided database using the command: `sqlite3 exam.db`
- the schema in exam.db is identical to the database used in Assignment 2 that you should be familiar with
- you can find the dataset description here (extracted from Assignment 2 spec)
- read `ans.sql` and identify where you can fill in your answer for each question
- attempt the questions and fill in the answers in `ans.sql`
- you are allowed to create additional views to help formulate your queries, or use any existing functions provided by sqlite3 on CSE linux machines if needed, but you are not allowed to create any extra tables
- during marking, any SQL query test that does not end in **60 seconds** on `grieg` will be terminated and receives zero marks for the test
- unless specified in the question, the ordering of your SQL results does not matter
- login to `grieg` or a CSE linux machine, and test your `ans.sql`
- submit all the Required Files via WebCMS3 (or give) as described above

End of Notes

Exam Questions

Question 1 (4 marks)

Given the provided exam.db, write an SQL query to list the names of all directors in the database and the number of movies each of them directed (including 0, if any), ranked by the number of movies in descending order, and then by name in ascending order.

For example, if we test your query below, the result will be as follows:

```
sqlite> .headers on
sqlite> .mode column
sqlite>
sqlite> select name, total from Q1 limit 1;
name              total
----------------  ----------
Steven Spielberg  26
sqlite>
```

Instructions:

- Your answer will be expressed as a view with its name defined in `ans.sql`

Question 2 (4 marks)

Given the provided exam.db, write an SQL query to list the best English movie (based on the highest IMDB score) of each year. If there are more than one best movies in a year (i.e., with the same highest IMDB score), output all of them. Your output includes year and movie title, ranked by year in ascending order and then title in ascending order. Ignore the movies with no year or with `num_voted_users` less than 100000, and do not output any year where no such movie exists.

For example, if we test your query below, the result will be as follows:

```
sqlite> .headers on
sqlite> .mode column
sqlite> .width 10 30
sqlite>
sqlite> select * from Q2 limit 2;
year        title
----------  ------------------------------
1936        Modern Times
1937        Snow White and the Seven Dwarf
sqlite>
```

Instructions:

- Your answer will be expressed as a view with its name defined in `ans.sql`

Question 3 (9 marks)

Given the provided exam.db, write an SQL query to find all actors (names) that have only acted in movies that include any of the following 5 genres: Crime, Horror, Mystery, Sci-Fi, Thriller. For example, suppose that actor A has acted in three movies in total. Two of them include at least one of the above genres (plus other genres), and the third movie does not include any of these genres. Then A will not be included in the result (due to the third movie). The names are ranked by the number of movies that they have acted in descending order, and then by name in ascending order. *Reminder:* when we test your query, we may add/remove some genres to/from the database.

For example, if we test your query below, the result will be as follows:

```
... assume headers on and in column mode ...
sqlite>
sqlite> select name from Q3 limit 1;
name
------------
Colin Salmon
sqlite>
```

Instructions:

- Your answer will be expressed as a view with its name defined in `ans.sql`

Question 4 (9 marks)

Given the provided exam.db, write an SQL query to find all actors (names) that have acted in movies covering at least 18 different types of genres in the database. For example, suppose that actor A has acted in four movies in total. If these four movies combined together cover 18 or more different genres (e.g., Adventure, Comedy, ...), then A will be included in the result. The names are ranked by their `facebook_likes` in descending order, and then by name in ascending order. *Reminder:* when we test your query, we may add/remove some genres to/from the database.

For example, if we test your query below, the result will be as follows:

```
... assume headers on and in column mode ...
sqlite>
sqlite> select name from Q4 limit 1;
name
-----------
Johnny Depp
sqlite>
```

Instructions:

- Your answer will be expressed as a view with its name defined in `ans.sql`

Question 5 (4 marks)

Given the provided exam.db, write an SQL query to list any movie (title) where its director also acted in the movie itself (i.e., the director has exactly the same name as one of the actors). The query output is ranked by movie title in ascending order, and then by director name in ascending order.

For example, if we test your query below, the result will be as follows:

```
... assume headers on and in column mode ...
sqlite>
sqlite> select title, name from Q5 limit 1;
title                 name
--------------------  --------------
2016: Obama's America  Dinesh D'Souza
sqlite>
```

Instructions:

- Your answer will be expressed as a view with its name defined in `ans.sql`

Question 6 (6 marks)

Consider a relation R(A,B,C,D,E). For each of the following sets of functional dependencies, assuming that those are the only dependencies that hold for R, list all of the candidate keys (separated by commas) for R.

a. AC -> D, B -> A, CD -> E

b. A -> B, B -> C, C -> DE, E -> B

c. AC -> D, DE -> AB

Instructions:

- Please modify the dummy placeholder REPLACE ME of the corresponding answer in `ans.sql`

Question 7 (6 marks)

Consider a relation R(A,B,C,D,E,F,G). For each of the following sets of functional dependencies, assuming that those are the only dependencies that hold for R, if R is not already in BCNF, decompose it into a set of BCNF relations (separated by commas). If it is already in BCNF, just write ABCDEFG as the final relation (i.e., no need for any BCNF decomposition).

a. A -> BCD, C->DE, E->F

b. A -> C, D -> EF, B -> G

c. ABC -> D, D -> E, F -> ADG

Instructions:

- Please modify the dummy placeholder REPLACE ME of the corresponding answer in `ans.sql`

Question 8 (4 marks)

For each of the following schedules (a) and (b), determine if it is conflict serializable.

a.

```
T1:      R(Y)           R(X) W(X) W(Z)           R(X) R(Z)
T2: R(Y)      W(Y) R(Y)                W(Y) R(X)           R(V) W(V)
```

b.

```
T1: R(X) R(Y) W(X)               W(X)
T2:                    R(Y)                    W(Y)
T3:                         R(Y)
T4:                R(X)                 R(Y) W(X)
```

c. Is schedule (a) view serializable ?

d. Is schedule (b) view serializable ?

Instructions:

- Please modify the dummy placeholder REPLACE ME of the corresponding answer in `ans.sql`

Question 9 (2 marks)

Consider a class enrolment scenario where we have three tables of interest:

```
Students(studentID,name,...)
Classes(classID,classType,day,time,room,capacity,nEnrolments,...)
ClassEnrolments(classID,studentID)
```

A constraint on these tables is that the `nEnrolments` field in the `Classes` table contains a count of the number of students associated with this class in the `ClassEnrolments` table (i.e., it holds a count of all of the `ClassEnrolments` tuples which contain that particular `Classes.classid` value).

Which of the following triggers and trigger functions provides the **best** way to ensure that this constraint is maintained when new enrolments are added, where "best" means only applies trigger actions when necessary and never makes incorrect changes.

A.

```
create function insertTrigger() returns trigger
as $$
begin
    update Classes set nEnrolments = nEnrolments + 1 where  classID = new.classID;
end;
$$ language plpgsql;
create trigger classEnrolmentInsertTrigger
before insert on ClassEnrolments
execute procedure insertTrigger();
```

B.
```
create function insertTrigger() returns trigger
as $$
begin
    update Classes set nEnrolments = nEnrolments + 1 where  classID = new.classID;
end;
$$ language plpgsql;
create trigger classEnrolmentInsertTrigger
after insert on ClassEnrolments
execute procedure insertTrigger();
```

C.
```
create function insertTrigger() returns trigger
as $$
begin
    update Classes set nEnrolments = nEnrolments + 1 where  classID = old.classID;
end;
$$ language plpgsql;
create trigger classEnrolmentInsertTrigger
before insert on ClassEnrolments
execute procedure insertTrigger();
```

D.
```
create function insertTrigger() returns trigger
as $$
begin
    update Classes set nEnrolments = nEnrolments + 1 where  classID = old.classID;
end;
$$ language plpgsql;
create trigger classEnrolmentInsertTrigger
after insert on ClassEnrolments
execute procedure insertTrigger();
```

Instructions:

- Please modify the dummy placeholder REPLACE ME of the corresponding answer in `ans.sql`

Question 10 (2 marks)

Consider the following relational schema: R(a,b) S(c,d,e), where a is the primary key of R and c is the primary key of S and d is a foreign key from S to R.

Which of the following relational algebra expressions (from A to E) is **not** a correct translation of the SQL statement:

```
select *
from   R, S
where  b > 5 and a = d and e < 10;
```

A.
```
(Sel[b>5] R) Join[a=d] (Sel[e<10] S)
```

B.
```
Sel[b>5] (R Join[a=d] (Sel[e<10] S))
```

C.
```
Sel[e<10] (Sel[b>5] (R Join[a=d] S))
```

D.
```
Sel[b>5] (Sel[e<10] (R Join[a=d] S))
```

E.
```
None of the above
```

Instructions:

- Please modify the dummy placeholder REPLACE ME of the corresponding answer in `ans.sql`

End of Exam