

# Assignment 2

## Queries, Functions and Triggers

(worth 15%)

### Aims

The aims of this assignment are to:

- formulate SQL queries;
- populate an RDBMS with a real dataset, and analyse the data;
- design test data for testing SQL queries;
- create SQL views;
- understand the limitations of SQL queries;
- create SQL functions (when needed); and
- create triggers.

### Description

In this assignment, the schema for a simple publication database is provided to you. The file is: [pub-schema.sql](#). Based on the provided schema, you are required to answer the following questions by formulating SQL queries. You may create SQL functions or PLpgSQL to help you, if and only if the standard SQL query language is not expressive and powerful enough to satisfy a particular question. To enable auto-marking, your queries should be formulated as SQL views, using the view names and attribute names provided. Since this assignment is auto-marked, using different view names and/or attribute names from the provided names may result in losing part or all of the marks for the questions involved. If any answer requires you to output a calculated value that has decimals, you should round it to an integer.

1. List all the editors.

```
create or replace view Q1(Name) as ...
```

2. List all the editors that have authored a paper.

```
create or replace view Q2(Name) as ...
```

3. List all the editors that have authored a paper in the proceeding that they have edited.

```
create or replace view Q3(Name) as ...
```

4. For all editors that have authored a paper in a proceeding that they have edited, list the title of those papers.

```
create or replace view Q4(Title) as ...
```

5. Find the title of all papers authored by an author with last name "Clark".

```
create or replace view Q5(Title) as ...
```

6. List the total number of papers published in each year, **ordered** by year in ascending order. Do not include papers with an unknown year of publication. Also do not include years with no publication.

```
create or replace view Q6(Year, Total) as ...
```

7. Find the most common publisher(s) (the name). (i.e., the publisher that has published the maximum total number of papers in the database).

```
create or replace view Q7(Name) as ...
```

8. Find the author(s) that co-authors the most papers (output the name). If there is more than one author with the same maximum number of co-authorships, **output all of them**.

```
create or replace view Q8(Name) as ...
```

9. Find all the author names that never co-author (i.e., **always published a paper as a sole author**).

```
create or replace view Q9(Name) as ...
```

10. For each author, list their total number of co-authors, ordered by the total number of co-authors in descending order, followed by author names in ascending order. For authors that never co-author, their total is 0. For example, assume John has written 2 papers so far: one with Jane, Peter; and one with Jane, David. Then the total number of co-authors for John is 3. In other words, it is the number of people that have written papers with John.

```
create or replace view Q10(Name, Total) as ...
```

11. Find all the author names that have never co-authored with any co-author of Richard (i.e. Richard is the author's first name), nor co-authored with Richard himself.

```
create or replace view Q11(Name) as ...
```

12. Output all the authors that have co-authored with or are indirectly linked to Richard (i.e. Richard is the author's first name). We define that  $a$  is indirectly linked to  $b$  if there exists  $a \text{ C } p_1, p_1 \text{ C } p_2, \dots, p_n \text{ C } b$ , where  $x \text{ C } y$  means  $x$  is co-authored with  $y$ .

```
create or replace view Q12(Name) as ...
```

13. Output the authors name, their total number of publications, the first year they published, and the last year they published. Your output should be ordered by the total number of publications in descending order and then by name in ascending order. If none of their publications have year information, the word "unknown" should be output for both first and last years of their publications.

```
create or replace view Q13(Author, Total, FirstYear, LastYear) as ...
```

14. Suppose that all **papers** that are in the database research area either contain the word or substring "data" (case insensitive) in their title or in a proceeding that contains the word or substring "data". Find the number of authors that are in the database research area. (We only count the number of authors and will not include an editor that has never published a paper in the database research area).

```
create or replace view Q14(Total) as ...
```

15. Output the following information for all proceedings: editor name, title, publisher name, year, total number of papers in the proceeding. Your output should be ordered by the total number of papers in the proceeding in descending order, then by the year in ascending order, then by the title in ascending order.

```
create or replace view Q15(EditorName, Title, PublisherName, Year, Total) as ...
```

16. Output the author names that have never co-authored (i.e., always published a paper as a sole author) nor edited a proceeding.

```
create or replace view Q16(Name) as ...
```

17. Output the author name, and the total number of proceedings in which the author has **at least one** paper published, ordered by the total number of proceedings in descending order, and then by the author name in ascending order.

```
create or replace view Q17(Name, Total) as ...
```

18. Count the number of publications per author and output the minimum, average and maximum count per author for the database. Do not include papers that are not published in any proceedings.

```
create or replace view Q18(MinPub, AvgPub, MaxPub) as ...
```

19. Count the number of publications per proceeding and output the minimum, average and maximum count per proceeding for the database.

```
create or replace view Q19(MinPub, AvgPub, MaxPub) as ...
```

20. Create a **trigger** on RelationPersonInProceeding, to check and disallow any insert or update of a paper in the RelationPersonInProceeding table from an author that is also the editor of the proceeding in which the paper has published.
21. Create a **trigger** on Proceeding to check and disallow any insert or update of a proceeding in the Proceeding table with an editor that is also the author of at least one of the papers in the proceeding.
22. Create a **trigger** on InProceeding to check and disallow any insert or update of a proceeding in the InProceeding table with an editor of the proceeding that is also the author of at least one of the papers in the proceeding.

## Submission

**Submission** : Login to a CSE Linux machine such as `wagner` and use the give command below to submit the assignment (note that the give command does not work on `grieg`):

```
give cs9311 a2 a2.sql
```

**Deadline** : Sunday 7 October 2018 @ 23:59 (Week 10)

**Late Penalty**: 0.075 *marks* of the total mark (i.e., 15 marks) for each hour late (i.e., 1.8 marks per day).

**Notes**: For fairness to all students in the class, no special considerations will be given to those:

- who claim to have submitted their assignment but the assignment has not been received by the `give` system above (use the "classrun" command to check if your assignment has been submitted, e.g., "9311 classrun -check a2");
- who submit their assignment a few minutes late and request to be considered as non-late submissions (please do submit your assignment early in case of your network connection problem, computer breakdown, etc);
- who claim that their assignments worked perfectly on their home computers but somehow did not work on CSE Linux machines (we will only test and mark your assignments on CSE Linux machines, and will not consider the results on your own machines. Hence, please test your assignments on CSE Linux machines before submission).

## Assessment

This assignment is worth a total of **15 marks**.

Your submission (in a file called `a2.sql`) will be auto-marked to check:

- whether it is syntactically correct;
- **if using SQL queries without creating a function or PLpgSQL unless it is necessary**; and of course,
- if each query produces correct results.

Queries are worth equal marks. Also, documentation and coding style (e.g., comments, naming of variable/aliases) will be considered while marking.

## What To Do Now

Make sure you read the above description thoroughly, and review and/or test out the provided schema [pub-schema.sql](#) and sample data [pub-insert.sql](#). The sample data is provided to help you quickly get started. While the same schema will be used to test your submission, a different dataset (that may be larger, smaller, or totally different) will be used for auto-marking. Therefore, **you may need to create your own or modify the provided data file to test your queries before submitting your assignment**. Note that you do not need to submit your

data file as part of the submission.

If any aspect of this assignment specification requires further clarification, ask for it under the section "Assignments/Assignment 2/Assignment 2" on the course website Forums.

**Reminder:** before you submit, ensure that your solution (a2.sql) will load into PostgreSQL without error if used as follows on grieg:

```
% dropdb a2
% createdb a2
% psql a2 -f pub-schema.sql
% psql a2 -f pub-insert.sql
% psql a2 -f a2.sql
... will produce notices, but should have no errors ...
% psql a2
... can start testing your solution ...
```

**Penalty:** If we have to fix errors in your solution before it will load, you will incur 3 (out of a total of 15) marks "penalty". Also, if your view or attribute names are different from the names specified above, you will incur 2 marks "penalty".

## Plagiarism

The work you submit must be your own work. Submission of work partially or completely derived from any other person or jointly written with any other person is not permitted. The penalties for such an offence may include negative marks, automatic failure of the course and possibly other academic discipline. Assignment submissions will be examined both automatically and manually for such submissions.

Do not provide or show your assignment work to any other person - apart from the teaching staff of this subject. If you knowingly provide or show your assignment work to another person for any reason, and work derived from it is submitted you may be penalized, even if the work was submitted without your knowledge or consent. This may apply even if your work is submitted by a third party unknown to you.