The Chinese University of Hong Kong
Department of Systems Engineering and Engineering Management

SEEM5770/ECLT5840 Open Systems/Electronic Commerce

2020-21 Assignment 2: SQL

This assignment is for understanding SQL, which is worth of 20% of the total assessment.[1] The due date for the assignment 2 is 11:59pm, November 18, 2020. The late penalty will be 10% per day. A submission will not be accepted five days after the deadline. You need to put down your student id and your name when you submit your assignment. A soft-copy submission is required to be submitted to the online eLearning system.

The lectures on database part are based on the textbook of "Database System Concepts" (6th Edition). The website for this textbook is at `http://codex.cs.yale.edu/avi/db-book/`. There are solutions to practice exercises in most chapters. We mainly discussed SQL (Chapter 3). It is recommend for you to try the practice exercises and see whether your answers are correct by checking the answers provided at the website. It helps.

There is a well-known free database management system, called MySQL. On the eLearning system, you can find how to install MySQL (`install_xampp_osx.pptx` for Max OS and `install_xampp_windows.pptx` for Windows), how to create relations for the schema given in `database-a.pptx` slide 1.22 by SQL (`create.sql`), and how to insert tuples into a relation (`insert.sql`). Start practicing SQL as soon as possible. If you have any questions, send emails to either `seem5770@se.cuhk.edu.hk` or `eclt5840@se.cuhk.edu.hk`.

You do not need to use MySQL to answer the questions. But it helps, if you try to use MySQL.

[1] Departmental Guideline for Plagiarism (Department of Systems Engineering and Engineering Management): If a student is found plagiarizing, his/her case will be reported to the Department Examination Panel. If the case is proven after deliberation, the student will automatically fail the course in which he/she committed plagiarism. The definition of plagiarism includes copying of the whole or parts of written assignments, programming exercises, reports, quiz papers, mid-term examinations and final examinations. The penalty will apply to both the one who copies the work and the one whose work is being copied, unless the latter can prove his/her work has been copied unwittingly. Furthermore, inclusion of others' works or results without citation in assignments and reports is also regarded as plagiarism with similar penalty to the offender. A student caught plagiarizing during tests or examinations will be reported to the Faculty office and appropriate disciplinary authorities for further action, in addition to failing the course.

1 Question on SQL

A relational database schema for an application is a set of relation schemas. Consider a database for a university. Its relational database schema is highlighted in the slide 1.22 (Introduction to SQL, `database-a.pptx`), which consists of 11 relation schemas (tables). The relation name (table names) are shown in the blue rectangles. The corresponding attribute names (column names) are shown in the white box below the relation names. The underlined attribute names in a relation schema shows that the combination of their values in a relation is unique. For example, in the student relation, every student will have a unique ID. The arrow "$\rightarrow$" from an attribute A in a relation schema R to an attribute B in another relation schema S, $A \rightarrow B$, shows that A's value must appear in B. For example, in the course relation, any value in the attribute dept_name must appear in the department relation. It implies a course must be offered by an existing department at the university. The relation names and attribute names are self-explained.

Formulate the following queries using SQL statements, assuming no null values exist in any attributes.

1. Find the distinct course titles the students in 'SEEM' department take.
2. Find the distinct course titles the students in 'SEEM' department take that are not offered by 'SEEM' department.
3. Show the total number of courses offered in year 2020.
4. Show the names of the departments that offer the largest number of courses in year 2020.
5. Show the department name and the average number of students an instructor supervises in the department, for all departments.
6. List the student names who take the courses his/her advisor teaches.
7. Show student ID, student name, and course_id for those students who have retaken a course at least twice (i.e., the student has taken the course at least three times).
8. Show the ID and name of each instructor who has never given an A grade in any course she/he has taught.
9. Show the names and IDs of those instructors who teach every course taught in his/her department (i.e., every course that appears in the course relation with the instructors department name).
10. Show the names of those departments whose budget is higher than that of Philosophy.

2 Parallel SQL Processing

We have discussed parallel query processing on a parallel database system with n nodes, with a focus on join processing. Suppose that there are two relations, r and s, where the schema of r and s are $R(A, B)$ and $S(A, B)$, respectively. Consider two ways to partiton the two relations on all n.

(a) The two relations are partitioned on all n nodes using round-robin partitioning.

(b) The two relations are partitioned on all n nodes using either range-partitioning or hash-partitioning.

Consider if you can parallelize the following queries when relations are partitioned by (a) and (b), respectively. Hint: You need to give the partitioning attribute name(s) with which the relations are paritioned. You need to consider how to partition relation r and/or s on all n nodes where r_i and s_i are the partition at the i-th node, consider how to process the operation on the i-th node locally, and consider if the union of all the local results at all nodes is the same as the final answer.

1. the difference operation such as $r - s$,

```
(select * from r) except (select * from s)
```

2. aggregation by the **count** operation

```
select count(B) from r group by A)
```

3. aggregation by the **count distinct** operation

```
select count(distinct B) from r group by A)
```