Assignment 1 (10% of total marks)

Due date: Sunday, 26 April 2020

Scope:

The tasks of this assignment cover **functional dependency and indexing**. The assignment covers the topics discussed in lecture 1, 2, and 3.

Assessment criteria:

Marks will be awarded for:

- Correct,
- Comprehensive, and
- Appropriate

application of the materials covered in this subject.

Assignment Specification:

Task 1 (5.0 marks)
Normalization

Consider the specifications of the sample database domains and the respective relational tables given below. For each one of the database domains listed below discover the respective sets of valid functional dependencies in the tables, and **identify its highest normal form**. Next **normalize the relational table to the required normal form specified in the question. Provide justification for each answer. A solution with no comprehensive justification scores no marks**.

a. ABC is an international company that own many buildings in a Science Park. In the current COVID-19 situation, company ABC has increase its cleaning activities. A group of cleaners are engaged to clean the company's buildings several times a day. Groups of 5 cleaners will be assigned to each building. Cleaners are not attached to any specific rooms within a building, that is, cleaners are able to clean any rooms. Once a cleaner has finished cleaning, the cleaner can move on to clean the next room schedule to be cleaned in a cleaning roster for a building. The room number is unique within a building, but not across the company.

   The company stores the information about the cleaning schedule for the building

   CLEANING-SCHEDULE(bldgNum, {roomNum}, sittingCapacity, {cleanerNum})

   into an un-normalized relational table as follow:

   CLEANING-SCHEDULE(bldgNum, roomNum, sittingCapacity, cleanerNum)

   Decompose the relational table CLEANING-SCHEDULE into a minimal number of relational tables in 4NF.

(5.0 marks)

Deliverables

Submit a pdf file consisting of the normalization process and explanation of the processes of the above relational tables. Name your pdf file as **solution1.pdf**.

Task 2 (5.0 marks)

Indexing

a. Consider the TPCHR benchmark database created through processing of CREATE TABLE statements include in the file TPCHR.

An index partIdx(p_name, p_type, p_retailprice) has been created over the relational table PART.

Construct SELECT statements that will use the index partIdx in the following ways:

i. Execution of the first SELECT statement must traverse the index vertically and it **MUST NOT** access a relational table PART.

(0.5 mark)

ii. Execution of the second SELECT statement must traverse the index vertically and then horizontally at the leaf level of the index and it **MUST NOT** access the relational table PART.

(0.5 mark)

iii. Execution of the third SELECT statement must traverse the leaf level of the index horizontally and it **MUST NOT** access the relational table PART.

(0.5 mark)

iv. Execution of the fourth SELECT statement must traverse the index vertically and it **MUST** access the relational table PART.

(0.5 mark)

v. Execution of the fifth SELECT statement must traverse the index vertically and then horizontally and it **MUST** access a relational table PART.

(0.5 mark)

Use ‘Explain plan for …’ to generate the execution plan for each of the SELECT statements that you have proposed and created. Next use the ‘SELECT * FROM TABLE(dbms_xplan.display) to display the generated execution plan to show that your proposed SELECT statements indeed perform according to the specification.

b. Consider the TPCHR benchmark database created through processing of CREATE TABLE statements include in the file TPCHR.

Consider the following SELECT statements:

i.
```
SELECT    distinct o_total, o_orderDate
FROM      ORDERS
ORDER BY O_ORDERDATE;
```

(0.5 mark)

ii.
```
SELECT    *
FROM      PART
WHERE P_BRAND = ‘GOLDEN BOLTS’
AND P_SIZE = 25;
```

(0.5 mark)

iii.
```
SELECT    c_custKey, c_name, c_address
FROM      CUSTOMER;
```

(0.5 mark)

iv.
```
SELECT    l_partKey, count(*)
FROM      LINEITEM
GROUP BY L_PARTKEY
HAVING COUNT(l_tax) > 2;
```

(0.5 mark)

v.
```
SELECT    *
FROM      LINEITEM
WHERE     l_quantity = 100
OR        l_shipMode = ‘FAST’;
```

(0.5 mark)

Find index that speeds up processing of the statements in the best possible way listed above, and create the index. You are only allowed to create **ONE** index per query. The best possible way means the database system will execute a query and uses the index proposed by you.

Use ‘Explain plan for ...’ to generate the execution plan for each of the SELECT statements that you have proposed and created. Next use the ‘SELECT * FROM TABLE(dbms_xplan.display) to display the generated execution plan to show that your proposed SELECT statements indeed perform according to the specification. If you think that no index can help to speed up the query processing, justify and give explanation.

Deliverables

A file pdf file with the SELECT statements created for Part 2(a) above, and the index proposed for each statement in Part 2(b). You may put your solution for Part 2(a) and Part 2(b) together in **one** pdf file named as **solution2.pdf**.

Submissions

This assignment is due by 9:00 pm (21:00 hours) Sunday, 26 April 2020, **Singapore time**.

Submit the files **solution1.pdf** and **solution2.pdf** through Moodle in the following way:

1) Zip all the files (Solution1.pdf and solution2.pdf into one zipped folder.)
2) Access Moodle at **http://moodle.uowplatform.edu.au/**
3) To login use a Login link located in the right upper corner the Web page or in the middle of the bottom of the Web page
4) When successfully logged in, select a site CSCI235 (SP220) Database Systems
5) Scroll down to a section Submissions of Assignments
6) Click at Submit your Assignment 1 here link.
7) Click at a button Add Submission
8) Move the zipped file created in Step 1 above into an area provided in Moodle. You can drag and drop files here to add them. You can also use a link *Add...*
9) Click at a button Save changes,
10) Click at check box to confirm authorship of a submission,
11) When you are satisfied, remember to click at a button Submit assignment.

A policy regarding late submissions is included in the subject outline. Only

one submission per student is accepted.

Assignment 1 is an individual assignment and it is expected that all its tasks will be solved individually without any cooperation with the other students. Plagiarism is treated seriously. Students involved will likely receive zero. If you have any doubts, questions, etc. please consult your lecturer or tutor during lab classes or over e-mail.

End of specification