School of Computing and Information Technology
University of Wollongong

Session: **Spring 2020**
Lecturer: **Janusz R. Getta**

CSCI235/CSCI835 Database Systems
Assignment 2

14 September 2020

Scope

This assignment includes the tasks related to database normalization, indexing of relational tables, and using cursors to search a database.

The outcomes of the laboratory work are due by **Saturday 17 October, 2020, 7.00 pm (sharp).**

Please read very carefully information listed below.

This assignment contributes to 20% of the total evaluation in a subject CSCI235 and it contributes to 17% of the total evaluation in a subject CSCI835.

A submission procedure is explained at the end of specification.

This assignment consists of 4 tasks and specification of each task starts from a new page.

It is recommended to solve the problems before attending the laboratory classes in order to efficiently use supervised laboratory time.

A submission marked by Moodle as "late" is treated as a late submission no matter how many seconds it is late.

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

A submission of compressed files (zipped, gzipped, rared, tared, 7-zipped, lhzed, … etc) is not allowed. The compressed files will not be evaluated.

All files left on Moodle in a state "`Draft(not submitted)`" will not be evaluated.

An implementation that does not compile due to one or more syntactical and/or run time errors scores no marks.

It is expected that all tasks included within **Assignment 2** will be solved **individually without any cooperation** with the other students. If you have any doubts, questions, etc. please consult your lecturer or tutor during lab classes or office hours. Plagiarism will result in a **FAIL** grade being recorded for the assessment task.

Task 1 (6 marks)
Concurrent executions of database transactions

Consider the database transactions listed below.

T1	T2	T3
`read(x)`	`read(y)`	`read(z)`
`write(y,x+1)`	`write(x,y+1)`	`write(x,z+1)`
`commit`	`commit`	`write(y,z+2)`
		`commit`

Assume that the initial values of the persistent data items `x`, `y`, and `z` are the following.
`x = 1, y = 2`, and `z= 3`.

(1) (2 marks)
Show a sample concurrent execution of the transactions `T1`, `T2`, and `T3` that is nonconflict serializable and that is view serializable.

Prove, that the execution is nonconflict serializable and that it is view serializable.

When visualizing the concurrent executions use a technique of two-dimensional diagrams presented to you during the lecture classes, for example, see a presentation 10 Introduction to Transaction Processing (1), slide 9.

(2) (2 marks)
Show a sample concurrent execution of the transactions `T1`, `T2`, and `T3` that is conflict serializable and that is not order-preserving conflict serializable.

Prove, that the execution is conflict serializable and that it is not order-preserving conflict serializable.

When visualizing the concurrent executions use a technique of two-dimensional diagrams presented to you during the lecture classes, for example, see a presentation 10 Introduction to Transaction Processing (1), slide 9.

(3) (2 marks)
Show a sample concurrent execution of the transactions `T1`, `T2`, and `T3` that is recoverable and that is not strict.

Prove, that the execution is recoverable and that it is not strict.

When visualizing the concurrent executions use a technique of two-dimensional diagrams presented to you during the lecture classes, for example, see a presentation 10 Introduction to Transaction Processing (1), slide 9.

Deliverables

A file `solution1.pdf` with:

(1) a visualization of a sample concurrent execution of the transactions `T1`, `T2`, and `T3` that is nonconflict serializable and that is view serializable and a proof that the execution is nonconflict serializable and that it is view serializable.

(2) visualization of a sample concurrent execution of the transactions `T1`, `T2`, and `T3` that is conflict serializable and that is not order-preserving conflict serializable and a proof that the execution is conflict serializable and that it is not order-preserving conflict serializable.

(3) visualization of a sample concurrent execution of the transactions `T1`, `T2`, and `T3` that is recoverable and that is not strict and a proof that the execution is recoverable and that it is not strict.

Task 2 (6 marks)
Serialization graph testing, 2PL, and Timestamp ordering scheduler

Consider a concurrent execution of database transactions `T1`, `T2`, and `T3` such that the execution is not controlled by any scheduler.

```
T1                  T2                  T3
read(x)
write(x,x+1)
                    read(y)
                    write(x,y+1)
                                        read(z)
                                        write(x,z+1)
read(z)
write(z,x+2)
```

(1) (2 marks)
Assume, that the transactions attempt to interleave their operations in the same way as in the execution above. Show a sample concurrent execution of the transactions `T1`, `T2`, and `T3` that is controlled by serialization graph testing scheduler.

Draw a conflict serialization graph.

When visualizing the concurrent executions use a technique of two-dimensional diagrams presented to you during the lecture classes, for example, see a presentation 10 Introduction to Transaction Processing (1), slide 9.

(2) (2 marks)
Assume, that the transactions attempt to interleave their operations in the same way as in the execution above. Show a sample concurrent execution of the transactions `T1`, `T2`, and `T3` that is controlled by 2PL scheduler.

Assume, that to simplify the problem we use only a general concept of a `lock` and we do not distinguish between `shared locks` and `exclusive locks`.

When visualizing the concurrent executions use a technique of two-dimensional diagrams presented to you during the lecture classes, for example, see a presentation 10 Introduction to Transaction Processing (1), slide 9.

(3) (2 marks)
Assume, that the transactions attempt to interleave their operations in the same way as in the execution above. Show a sample concurrent execution of the transactions `T1`, `T2`, and `T3` that is controlled by timestamp ordering scheduler.

When visualizing the concurrent executions use a technique of two-dimensional diagrams presented to you during the lecture classes, for example, see a presentation 10 Introduction to Transaction Processing (1), slide 9.

Show the data items accessed by the transactions together with the timestamps left by the transactions on the data items.

Deliverables

A file `solution2.pdf` with:

(1) visualization of a sample concurrent execution of the transactions `T1`, `T2`, and `T3` that is controlled by <u>serialization graph testing scheduler</u> and a conflict serialization graph,

(2) visualization of a sample concurrent execution of the transactions `T1`, `T2`, and `T3` that is controlled by <u>2PL scheduler</u>,

(3) visualization of a sample concurrent execution of the transactions `T1`, `T2`, and `T3` that is controlled by <u>timestamp ordering scheduler</u>.

Task 3 (6 marks)
Processing transactions at `READ COMMITTED` level by a snapshot isolation scheduler

Consider a stored PL/SQL function `MAX_MIN` given below.

```
CREATE OR REPLACE FUNCTION MAX_MIN ( orderkey IN NUMBER )
RETURN NUMBER IS
 max_value NUMBER(12);
 min_value NUMBER(12);

BEGIN
 SELECT MAX(L_QUANTITY * L_EXTENDEDPRICE)
 INTO max_value
 FROM LINEITEM
 WHERE L_ORDERKEY = orderkey;

 SELECT MIN(L_QUANTITY * L_EXTENDEDPRICE)
 INTO min_value
 FROM LINEITEM
 WHERE L_ORDERKEY = orderkey;

 RETURN max_value-min_value;
END MAX_MIN;
/
```

(1) (4 marks)
Show a sample concurrent execution of the function at `READ COMMITTED` level that interleaves the operations with another transaction and such that a result returned by the function is incorrect. The operations performed by another transaction are up to you.

When visualizing the concurrent executions use a technique of two-dimensional diagrams presented to you during the lecture classes, for example, see a presentation 14 Transaction Processing in Oracle DBMS slide 16.

(2) (2 marks)
Rewrite a stored function such that it can be processed at `READ COMMITTED` level and show how the improved function interleaves the operations with another transaction and such that a result returned by the function is always correct.

When visualizing the concurrent executions use a technique of two-dimensional diagrams presented to you during the lecture classes, for example, see a presentation 14 Transaction Processing in Oracle DBMS slide 16.

Deliverables
A file `solution3.pdf` with:

(1) visualization of a sample concurrent execution of the function at `READ COMMITTED` level that interleaves the operations with another transaction and such that a result returned by the function is incorrect,

(2) visualization of a sample concurrent execution of an improved function at `READ COMMITTED` level that interleaves the operations with another transaction and such that a result returned by the function is always correct.

Task 4 (2 marks)
Deadlocks

Consider a stored PL/SQL function `SWAP` given below.

```
CREATE OR REPLACE PROCEDURE SWAP( order_key1  IN NUMBER,
                                  order_key2  In NUMBER ) IS

 total_price1 ORDERS.O_TOTALPRICE%TYPE;
 total_price2 ORDERS.O_TOTALPRICE%TYPE;

BEGIN
 SELECT O_TOTALPRICE
 INTO total_price1
 FROM ORDERS
 WHERE O_ORDERKEY = order_key1;

 SELECT O_TOTALPRICE
 INTO total_price2
 FROM ORDERS
 WHERE O_ORDERKEY = order_key2;

 UPDATE ORDERS
 SET O_TOTALPRICE = total_price1
 WHERE O_ORDERKEY = order_key2;

 UPDATE ORDERS
 SET O_TOTALPRICE = total_price2
 WHERE O_ORDERKEY = order_key1;

END SWAP;
/
```

Show a sample concurrent execution of two transactions both processing a function `SWAP`, both running at `READ COMMITTED` level, and such that the execution leads to a deadlock.

When visualizing the concurrent executions use a technique of two-dimensional diagrams presented to you during the lecture classes, for example, see a presentation 14 Transaction Processing in Oracle DBMS slide 16.

Deliverables
A file `solution4.pdf` with visualization of a sample concurrent execution of two transactions both processing a function `SWAP`, both running at `READ COMMITTED` level, and such that the execution leads to a deadlock.

Submission

Submit the files **`solution1.pdf`**, **`solution2.pdf`**, **`solution3.pdf`**, and **`solution4.pdf`** through Moodle in the following way:

(1) Access Moodle at **`http://moodle.uowplatform.edu.au/`**
(2) 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
(3) When logged select a site **`CSCI835/CSCI235 (S220) Database Systems`**
(4) Scroll down to a section **SUBMISSIONS**
(5) Click at a link **`In this place you can submit the outcomes of Assignment 2`**
(6) Click at a button **`Add Submission`**
(7) Move a file **`solution1.pdf`** into an area **`You can drag and drop files here to add them`**. You can also use a link **`Add`**...
(8) Repeat a step (7) for the files **`solution2.pdf`**, **`solution3.pdf`**, and **`solution4.pdf`**.
(9) Click at a button **`Save changes`**
(10) Click at a button **`Submit assignment`**
(11) Click at the checkbox with a text attached: **`By checking this box, I confirm that this submission is my own work,`** ... in order to confirm the authorship of your submission.
(12) Click at a button **`Continue`**

End of specification