School of Computing and Information Technology
University of Wollongong

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

CSCI235/CSCI835 Database Systems
Laboratory 4

7 September 2020

<u>Scope</u>

This laboratory includes implementation of database triggers that implement an access control rule.

The outcomes of the laboratory work are due by **Saturday 19 September, 2020, 7.00 pm (sharp).**

Please read very carefully information listed below.

This laboratory contributes to 2% of the total evaluation in the subject.

A submission procedure is explained at the end of specification.

This laboratory consists of 2 tasks and specification of each task starts from a new page.

It is recommended to solve the problems before attending a laboratory class 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 errors scores no marks.

It is expected that all tasks included within **Laboratory 4** 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 **<u>FAIL</u>** grade being recorded for the assessment task.

Prologue

Download the files dbschema.bmp, dbcreate.sql, dbload.sql, and dbdrop.sql included in a section SAMPLE DATABASES on Moodle. To create a sample database, process as script dbcreate.sql. To drop a sample database, process a script dbdrop.sql. To load data into a sample database, process as script dbload.sql. A conceptual schema of a sample database is included in a file dbschema.bmp.

It is possible to use a pseudo-column USER to find a name of user you are connected as. To try it, connect to one of Oracle 19c servers and process the following statement:

```
SELECT USER
FROM DUAL;
```

The system supposed to return a name of a user you are connected as.

No report is expected from the implementations of the actions listed in Prologue.

The objective of this laboratory is use database triggers to implement the following access control rule.

An item can be added to an order only by a user who created the order.

To implement the access control rule listed above we have to extend the structures of a sample database a bit. We add a new relational table to store information about the names of users and the orders created by the users. Then, we implement a database trigger that automatically inserts information about a user and about an order key into the new table whenever a new order is added. If an order is deleted then information about a user who earlier created an order to be deleted must be removed from the table together with an order key.

In the second step, we implement a database trigger that fires when a new item is added to an order. The trigger must check if a user who attempts to insert the item is the same user who inserted an order into which the new item must be inserted. If such access control rule is violated then the trigger aborts a transaction that attempted to insert a new item.

Tasks

Task 1 (1 mark)

Automatic recording of information in a database

In this task we implemented the first part of a system that verifies the following access control rule.

An item can be added to an order only by a user who created the order.

If you skipped the **Prologue** section of specification **Laboratory 4** then it is recommended to read it now.

Implement SQL script `solution1.sql` that performs the following actions.

(1) First, the script creates a relational table to store information about a new order key and a name of user who created a new order.

(2) Next, the script creates a database trigger, that automatically inserts into a relational table created in the previous step, a key of a new order and a name of a user who created the order. If an order is deleted then the trigger must automatically delete information about an order key and a user who earlier created the order. A type of a trigger is up to you.

(3) Next, the script comprehensively tests the trigger. A test must include listing of the contents of a table with order keys and user names, insertion of an order, listing of the contents of the table again, deletion of an order, and listing of the contents of the table again.

When ready, process SQL script `solution1.sql` and create a report from processing of the script in a file `solution1.lst`.

Your report must include a listing of all PL/SQL statements processed. To achieve that put the following SQLcl commands:

```
SPOOL solution1
SET ECHO ON
SET FEEDBACK ON
SET LINESIZE 200
SET PAGESIZE 400
SET SERVEROUTPUT ON
```

at the beginning of SQL script and

```
SPOOL OFF
```

at the end of SQL script.

Deliverables
A file `solution1.lst` with a report from the implementation of a script `solution1.sql` that creates a new relational table, a database trigger, and tests the trigger. A report must have no errors and it must list all SQL statements processed.

Task 2 (1 mark)
Implementation of an access control rule

Task 1 must be implemented before Task 2.

In this task we implement the second part of a system that verifies the following access control rule.

An item can be added to an order only by a user who created the order.

If you skipped the **Prologue** section of specification **Laboratory 4** then it is recommended to read it now.

Implement SQL script `solution2.sql` that performs the following actions.

(1) First, the script creates a database trigger that fires whenever a new item is added to an order. The trigger must verify if a user who creates a new item is the same user who created an order into which a new item is added.

If a user who attempts to insert a new item into an order created by another user the trigger must abort the transaction and must display an error message. Otherwise, the trigger does nothing.

(2) Next, the script tests the trigger.

A test must include creation of two new orders.

Then, insertion of a new item into one of the orders created a moment ago must be successful.

Next, the script updates one of the user names in a relational table created in Task 1.

Next, insertion of another item into an order created by a user whose name has been updated fails the access control rule.

Finally, insertion of an item to an order whose creator is unknown must also fail the access control rule.

In this task a processing error is a good news and we abolish a principle saying that no marks are granted for processing errors. Note, that it applies only to Task 2.

When ready, process SQL script `solution2.sql` and save a report from processing in a file `solution2.lst`.

Your report must include a listing of all PL/SQL statements processed. To achieve that put the following SQLcl commands:

```
SPOOL solution2
SET ECHO ON
SET FEEDBACK ON
SET LINESIZE 100
SET PAGESIZE 200
SET SERVEROUTPUT ON
```

at the beginning of SQL script and

```
SPOOL OFF
```

at the end of SQL script.

Deliverables
A file `solution2.lst` with a report from processing of SQL script `solution2.sql`. A report must list all SQL statements processed and all error messages.

Submission

Submit the files **solution1.lst** and **solution2.lst** 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 Laboratory 4**

(6) Click at a button **Add Submission**

(7) Move a file **solution1.lst** 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 a file **solution2.lst**.

(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