School of Computing and Information Technology
University of Wollongong

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

CSCI235/CSCI835 Database Systems
Laboratory 5
12 October 2020

Scope

This laboratory includes implementation of database links, synonyms, programming of distributed database systems, and transformation of relational tables into BSON documents.

The outcomes of the laboratory work are due by **Saturday 24 October, 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 5**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.

Prologue 1

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`.

Connect to one of `data-pc` Oracle 19c servers. The database server you are connected now will be called as the "host server".

(1) While connected to the "host server" process the scripts `dbcreate.sql` and `dbload`.sql scripts to create a sample database on the "host server" and to load some data.

(2) Connect to another `data-pc` Oracle19c server. This database server will be called as the "remote server".

(3) Recreate the relational tables `LINEITEM` and `ORDERS` on the "remote server". To do so you can update and re-use a script `dbcreate.sql`. Drop all referential integrity constraints that are not needed.

There is no need for a report from the actions performed so far.

Tasks

Task 1 (1 mark)

Creating a database link, synonym, and programming distributed database system.

If you skipped the **Prologue 1** section in a specification of **Laboratory 5** then it is recommended to read it and to perform the actions described there now.

Implement SQL script `solution1.sql` that performs the following actions while connected to the "host server".

(1) Create a database link from the "host server" to the "remote server".

(2) Create a synonym names of the relational tables located at the "remote server".

(3) Copy information about all orders and the contents of all orders submitted before 1 March 1992 from the "host server" to the "remote server".

(4) Delete from the "host server" all orders and the contents of all orders submitted before 1 March 1992.

(5) Implement the following queries as `SELECT` statements.

   (i) Find the total number of orders recorded in both the "host server" and the “remote server”.

   (ii) Find all order keys and all prices of all orders in both “host server” and the “remote server” where a price is greater than `150000`. Sort the results in the ascending order of order keys.

   (iii) Find all order keys of all nonempty orders, i.e. the orders that include at least one item. List the order keys together with the total number of items included in each order. Sort the results in the descending order of the total number of items included in each order. Sort all orders with the same total number of items in the ascending order of order keys. Use both "host server" and "remote server".

   (iv) Find the part keys of all parts included in at least one order located in the "host server" and in at least one order located in the "remote server".

(6) Drop the synonyms and a database link.

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 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
```

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 the database links, synonyms, and processes the distributed databases. A report must have no errors and it must list all SQL statements processed.

Prologue 2

Install VirtualBox on your systems. If you do not remember how you did it in CSIT115 then it is explained in

https://documents.uow.edu.au/~jrg/115/cookbook/e1-1-frame.html

how to do it.

Download from Moodle `ova` image of a virtual machine with Ubuntu and MongoDB. The image is available in a section `OTHER RESOURCES`. You should get a file:

```
Ubuntu18.04-64bits-MongoDB-4.2.2-08-JAN-2020.ova
```

Start VirtualBox and import `ova` image of a virtual machine with Ubuntu and MongoDB. You should get a new virtual machine `Ubuntu18.04-64bits-MongoDB-4.2.2-08-JAN-2020.`

Start a virtual machine `Ubuntu18.04-64bits-MongoDB-4.2.2-08-JAN-2020.`

A password to login as `CSCI235` user is:

```
csci235
```

When logged in, start Terminal program (3rd icon from bottom in a column of icons on the left hand size of a screen).

To start MongoDB server, process the following command in Terminal window.

```
mongod --dbpath DATA --port 4000
```

When MongoDB server is ready then among the other messages you should get a message:
`… waiting for connection on port 4000`

in a large number of messaged displayed by a starting server.

Minimize Terminal window. Do not close the window, from now, it is used as a console window by MongoDB server.

Open another Terminal window and to start MongDB command line interface, process the following command.

```
mongo –port 4000
```

For a good start, process a command `help`.

Task 2 (1 mark)
Transformation of data stored in the relational tables into data stored in BSON collection.

If you skipped the **Prologue 2** section in a specification of **Laboratory 5** then it is recommended to read it and to perform the actions described there now.

An objective of this task is to implement PL/SQL program (anonymous block or stored procedure or stored function), that lists the contents of the relational tables `REGION` and `NATION` as a sequence of invocations of `db.task2.insert(...)` method. Such sequence invocations can be later used to load data into a collection of BSON documents `task2`. Note, that `...` must be replaced with the correctly formatted data such that processing of `db.task2.insert(...)` methods will successfully insert the documents into a collection `task2`.

As an example, download and unzip the files `customer.zip`, `part.zip`, and `supplier.zip` available on Moodle in a section `SAMPLE DATABASES`. You should get the files `customer.js`, `part.js`, and `supplier.js`. The files contains invocations `db.tpchr.insert(...)` methods that insert into a collection `tpchr` the transformed data from the relational tables `CUSTOMER`, `ORDERS`, `PART`, and `SUPPLIER`. Your PL/SQL implementation supposed to generate a sequence of invocations of `db.task2.insert(...)` methods that can be used to load data from the relational tables `REGION` and `NATION` into a collection of BSON documents `task2`.

A PL/SQL implementation technique is up to you. You can implement an anonymous PL/SQL block or stored PL/SQL procedure or stored PL/SQL function. You can reuse the outcomes of Assignment 1, Task 4.

Please note, that the contents of the relational tables `REGION` and `NATION` must be transformed into nested BSON documents. A solution that re-implements relational tables as the separate BSON documents scores no marks.

Save your implementation in a file `solution2.sql`.

When ready, process SQL script `solution2.sql` and save a report from processing in a file `solution2-1.lst`.

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

```
SPOOL solution2-1
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.

To verify the correctness of your transformation copy the generated invocations of `db.task2.insert(...)` methods into a file `solution2.js` and use `load` method to create and load the contents of a collection `task2` on MongDB server.

Next use the methods:

```
db.task2.count();
db.task2.find().pretty();
```

to list the total number of documents in a collection `task2` and to list the contents of collection `task2` in a pretty format.

When ready, copy the contents of Terminal window with the results from counting and listing of the document in `task2` collection and paste it into a file `solution2-2.lst`.

Deliverables

A file `solution2-1.lst` with a report from processing of SQL script `solution2.sql`. A report must list all SQL and PL/SQL statements processed and all error messages. A file `solution2-2.lst` with a report from processing of the methods `db.task2.count()` and `db.task2.find().pretty()`.

Submission

Submit the files **solution1.lst**, **solution2-1.lst** and **solution2-2.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 5**
(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 the files **solution2-1.lst** and **solution2-2.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