School of Information Technology & Computer Science
University of Wollongong

Session: **3, July 2022**
Lecturer: **Janusz R. Getta**

CSCI317 Database Performance Tuning
Singapore 2022-3
Assignment 2

Published on 23 July 2022

Scope

This assignment includes the tasks related to storage defragmentation and appropriate configuration of storage allocation parameters, denormalization of conceptual and relational schemas, and finding the smallest set of indexes that improve performance of processing of a given collection of queries.

This assignment is due by **Saturday, 6 August 2022, 9.00 pm (sharp) Singaporean Time.**

Please read very carefully information listed below.

This assignment contributes to 15% of the total evaluation in the subject.

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 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.

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.

Please read very carefully information included in the Prologue section below about software environment to be used in the subject.

Prologue

In this subject we use Oracle 19c database server running under Oracle Linux 7.4 operating system on a virtual machine hosted by VirtualBox. To start Oracle database server you have to start VirtualBox first. If you have not installed VirtualBox on your system yet then it is explained in Cookbook for CSIT115 Recipe 1.1, Step 1 "How to use VirtualBox ?" (`https://www.uow.edu.au/~jrg/115/cookbook/e1-1-frame.html`) how to install and how to start VirtualBox.

When VirtualBox is started, import an appliance included in a file `OracleLinux7.4-64bits-Oracle19c-22-JAN-2020.ova`. You can download `ova` image of the appliance using the links published on Moodle.

When ready, power on a virtual machine `OracleLinux7.4-64bits-Oracle19c-22-JAN-2020`.

A password to a Linux user `ORACLE` is `oracle` and a password to Oracle users `SYSTEM` and `SYS` (database administrators) is also `oracle`. Generally, whenever you are asked about a password then it is always `oracle`, unless you change it.

When logged as a Linux user, you can access Oracle database server either through a command line interface (CLI) `SQLcl` or through Graphical User Interface (GUI) `SQL Developer`.

You can find in Cookbook for CSCI317, Recipe 1, How to access Oracle 19c database server, how to use SQL Developer, how to use basic SQL and SQLcl, and how to create a sample database ?
(`https://documents.uow.edu.au/~jrg/317sim/cookbook/e1-2-frame.html`) more information on how to use `SQLcl` and `SQL Developer`.

Tasks

Task 1 (3 marks)

An objective of this task is to reduce storage consumption through storage defragmentation and appropriate configuration of storage allocation parameters.

Assume, that we would like to perform a detailed analysis of the orders submitted in 1992 and we expect that the relational tables `ORDERS` and `LINEITEM` will be fully read many times. To speed up reading we would like to create the new relational tables `ORDERS1992` and `LINEITEM1992` to copy into the tables information about all orders submitted in 1992.

We also would like to minimize the total number of data blocks occupied by the new relational tables to reduce the total number of data blocks read during a full scan of both tables. To do so, we assume that only the following vertical fragments of the relational tables `LINEITEM` and `ORDERS` will be needed for data analysis. It means that no other columns will be need in the relational tables `ORDERS1992` and `LINEITEM1992`.

```
LINEITEM(L_ORDERKEY,L_LINENUMBER,L_QUANTITY,L_EXTENDEDPRICE,
         L_DISCOUNT,L_TAX,L_SHIPDATE)
ORDERS(O_ORDERKEY,O_CUSTKEY,O_TOTALRICE,O_ORDERDATE)
```

Assume, that the new tables will be only used in read mode.

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

(1) First, the script finds the total number of data blocks occupied by the relational tables `ORDERS` and `LINEITEM`.

(2) Next the script creates a new tablespace to keep the relational tables `ORDERS1992` and `LINEITEM1992`. The size of a new tablespaces must be carefully adjusted to the size of both relational tables `ORDERS1992` and `LINEITEM1992`.

(3) Next, the script creates the new relational tables `ORDERS1992` and `LINEITEM1992` to store information about orders submitted in 1992. Note, that the new relational tables must have the appropriate consistency constraints enforced. The tables must be stored in the new tablespace created in the previous step.

    Note, that in step (2) and step (3) you must take under the consideration an objective to minimize storage allocation for the new tables and a new tablespace.

(4) Next, copy information about the orders submitted in 1992 to the new relational tables.

(5) Next, whenever it is possible the script should reduce storage allocation for a tablespace used and the relational tables created earlier.

(6) Finally, the script finds the total number of data blocks occupied by the relational tables `ORDERS1992` and `LINEITEM1992`.

Note, that appropriate storage allocation may require few experiments before the final processing of a script `solution1.sql` and generation of a report `solution1.lst` from the processing. It is recommended to implement a script that reverses the persistent storage allocations performed by a script `solution1.sql`.

When ready start `SQLcl` client, connect to Oracle database server, and process SQL script `solution1.sql`. Save a report from processing of the script in a file `solution1.lst`. It is explained in Cookbook, Recipe 1.5, Step 9, "How to create and to save a report" how to save a report from processing of SQL script in a text file.

The script must be processed with `SQLcl` options `ECHO` and `FEEDBACK` set to `ON` such that all SQL statements processed are included in the report !

A good habit is to put `SQLcl` statements

```
SPOOL solution1
SET ECHO ON
SET FEEDBACK ON
SET LINESIZE 300
SET PAGESIZE 300
```

at the beginning of each SQL script implemented and the following statement at the end of the script

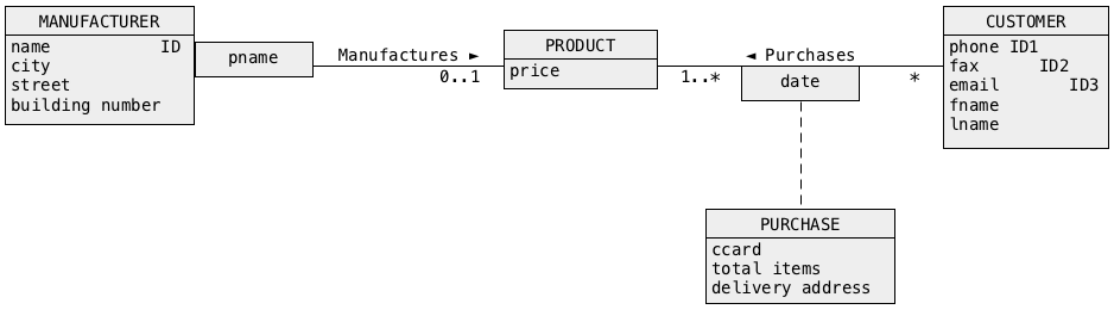
```
SPOOL OFF
```

A report from processing of the script must have NO syntax errors !

Deliverables
A file `solution1.lst` that contains a report from the processing of a script `solution1.sql`.

Task 2 (4 marks)
An objective of this task is to denormalize a conceptual schema to speed up processing of a given class of applications.

Consider the following conceptual schema:



(1) Perform simplification of the conceptual schema above and migration of identifiers and re-draw the simplified conceptual schema.

(2) We would like to improve the performance of the following class of applications:

> *Find the locations of the manufacturers (attributes* `city`, `street`, `building number` *in a class* `MANUFACTURER`*), which manufactured a given product (attribute* `pname` *in a class* `PRODUCT`*) purchased by a given customer (attributes* `fname`, `lname` *in a class* `CUSTOMER`*)*

A sample application that belongs to a class described above is the following.

> *Find the locations of manufacturers that manufacture bolts purchased by James Bond*.

Find the denormalizations of the simplified conceptual schema that improves the performance of the calls of applications described above. When performing the denormalizations apply the following transformations of the simplified conceptual schema: migration of attributes, decomposition of classes of objects. Re-draw the simplified conceptual schema after the denormalizations.

Find the denormalizations of the simplified conceptual schema that improves the performance of the calls of applications described above. Re-draw the simplified conceptual schema after the denormalizations.
You can use UMLet to create a simplified and denormalized conceptual schemas. A link to UMLet `UMLet 14.3 with CSIT115-815Palette (zipped)` is available at CSCI317 site on Moodle in `Resources` section.

The original conceptual schema is provided in a file `task2.uxf`.

Deliverables

A file `solution2.pdf` with a drawing of the simplified conceptual schema and a drawing of the denormalized conceptual schema expressed in a notation of simplified UML object classes. You are allowed to use any line drawing tool to draw the simplified and denormalized schema. A scanned copy of a neat hand drawing is also acceptable.

Task 3 (4 marks)
An objective of this task is to denormalize the relational schemas to speed up processing of a given class of applications.

Implement the following query as `SELECT` statement over TPC-HR benchmark database.

Find the total quantity of all items (attribute `L_QUANTITY` in a table `LINEITEM`) ordered by the customers from a given country (attribute `N_NAME` in a table `NATION`).

Use a denormalization of relational table to speed up processing of `SELECT` statement implemented above. To denormalize a relational table and to test the improvements in the performance create a script file `solution1.sql` that performs the following actions.

(1) First, the script processes the original `SELECT` statement create above. Use `TIMING` option of `SQLcl` client to measure the total processing time. Setting and using `TIMING` option of `SQLcl` client is described in *"How to ..." Cookbook, Recipe 2.1, Step 2*.

(2) Next, the script lists a query processing plan of the original `SELECT` statement.

(3) Next, the script performs denormalization of a relational table that speeds up the processing of a given `SELECT` statement in the best possible way. In this case, there is NO need for indexing and there is no need for creation of materialized views or any additional relational tables.

    It is recommended to denormalize a conceptual schema given in a file `tpchr.pdf` before performing any changes to the relational tables of TPC-HR database. There is no need to provide the outcomes of denormalization of a conceptual schema.

(4) Next, the script processes a new `SELECT` statement that accesses a denormalized relational table and retrieves the same results as the original `SELECT` statement. Use `TIMING` option to measure processing time. Note, that processing time should be shorter than processing time of the statement is processed before denormalization.

(5) Next, the script lists a query processing plan of the new `SELECT` statement.

(6) Next, the script creates an index to speed up processing of the new `SELECT` statement in the best possible way and again uses `TIMING` option to measure processing time. Note, that processing time should be much shorter than processing time in the previous step.

(7) Again, the script lists a query processing plan of the new `SELECT` statement.

When ready start `SQLcl` client, connect to Oracle database server, and process SQL script `solution3.sql`. Save a report from processing of the script in a file

`solution3.lst`. It is explained in Cookbook, Recipe 1.5, Step 9, "How to create and to save a report" how to save a report from processing of SQL script in a text file.

The script must be processed with `SQLcl` options `ECHO` and `FEEDBACK` set to `ON` such that all SQL statements processed are included in the report !

A good habit is to put `SQLcl` statements

```
SPOOL solution3
SET ECHO ON
SET FEEDBACK ON
SET LINESIZE 300
SET PAGESIZE 300
```

at the beginning of each SQL script implemented and the following statement at the end of the script

```
SPOOL OFF
```

A report from processing of the script must have NO syntax errors !

Deliverables
A file `solution3.lst` that contains a report from the processing of a script `solution3.sql`.

Task 4 (4 marks)
An objective of this task is to improve performance of query processing through indexing.

In this task you must operate on the original state of a sample benchmark `TPC-HR` database. It is explained at the end of **Prologue** section how to return to the original state of the database.

Consider the following `SELECT` statements available in a file `task4.sql`:

```
SELECT COUNT(DISTINCT O_TOTALPRICE)
FROM ORDERS;

SELECT *
FROM CUSTOMER JOIN ORDERS
              ON CUSTOMER.C_CUSTKEY = ORDERS.O_CUSTKEY
WHERE C_ACCTBAL = 1000 AND O_TOTALPRICE = 1000;

SELECT *
FROM ORDERS JOIN LINEITEM
            ON ORDERS.O_ORDERKEY = LINEITEM.L_ORDERKEY
WHERE O_ORDERDATE = '09-SEP-2021' OR L_QUANTITY = 500;

SELECT *
FROM LINEITEM JOIN PART
              ON LINEITEM.L_PARTKEY = PART.P_PARTKEY
WHERE L_QUANTITY = 100 AND P_NAME = 'bolt';

SELECT COUNT(PS_AVAILQTY)
FROM PARTSUPP JOIN LINEITEM
              ON PARTSUPP.PS_PARTKEY = LINEITEM.L_PARTKEY
              JOIN ORDERS
              ON LINEITEM.L_ORDERKEY = ORDERS.O_ORDERKEY
WHERE O_ORDERSTATUS = 'F' AND L_QUANTITY > 200 AND
PS_SUPPLYCOST > 100;
```

(1) Find the smallest number of the smallest indexes that improve performance of all queries listed above. Please keep assume, that an objective of this task is to get any possible improvement in performance with the smallest number of the smallest indexes used.

Implement SQL script `solution4.sql` that performs the following actions:

(2) Create the indexes found in the previous step and find the size of the indexes measured in bytes and data blocks.

(3) Next, the script finds a query processing plan for `SELECT` statements give above. You can use a value listed in a column `Cost` of query processing plan as a measure of improvement in query processing with an index.

(4) Next, the script drops all indexes created in a step (3).

When ready, process SQL script file `solution4.sql` and save a report from processing in a file `solution4.lst`.

The script must be processed with `SQLcl` options `ECHO` and `FEEDBACK` set to `ON` such that all SQL statements processed are included in the report !

You must put the following `SQLcl` statements

```
SPOOL solution4
SET ECHO ON
SET FEEDBACK ON
SET LINESIZE 300
SET PAGESIZE 300
```

at the beginning of each SQL script implemented and the following statement at the end of the script

```
SPOOL OFF
```

A report from processing of the script must have NO syntax errors !

Deliverables
A file `solution4.lst` that contains a report from the processing of a script `solution4.sql`.

Submission

Note, that you have only one submission. So, make it absolutely sure that you submit the correct files with the correct contents. No other submission is possible!

Submit the files **`solution1.lst`**, **`solution2.pdf`**, **`solution3.lst`**, and **`solution4.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 **`CSCI317 (SP322) Database Performance Tuning`**
(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.lst`** into an area **`You can drag and drop files here to add them`**. You can also use a link **`Add`**...
(8) Repeat step (7) for the files **`solution2.pdf`**, **`solution3.lst`**, and **`solution4.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