School of Information Technology & Computer Science
University of Wollongong

Session: **Autumn 2021**
Lecturer: **Janusz R. Getta**

CSCI317 Database Performance Tuning
Project

Published on 14 June 2021

Scope

This project includes the tasks in improving performance of database applications.

The outcomes of the project are due by **Friday, 19 June, 2021, 7.00 pm (sharp).**

This project is worth 30% of the total evaluation in the subject.

A submission procedure is explained at the end of project specification.

The outcomes of the project must be submitted through Moodle in the same way as all other coursework tasks in the subject. A submission link is available in a section PROJECT (Available from 14 June, 2021) just below a link to specification of the project.

A submission of the outcomes of the project marked by Moodle as "late" is treated as a late submission no matter how many seconds it is late. If you have poor Internet connection then allocate more time for a submission procedure. Please apply a principle saying that "it is better to submit a project 1hour too early than 1 second too late".

A policy regarding late submissions is same as for late submissions of all coursework tasks in the subject and it is explained 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 **Project** 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 **Prologue** section below about the 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. To start VirtualBox navigate through the following menus: Start->All Programs->Oracle VM VirtualBox->Oracle VM VirtualBox.
It is explained in Cookbook for CSIT115 Recipe 1.1, Step 1 "How to start VirtualBox ?" (`https://www.uow.edu.au/~jrg/115/COOKBOOK/e1-1-frame.html`) how to start VirtualBox.

When VirtualBox is started, import an appliance included in a file `OracleLinux7.4-64bits-Oracle19c-22-JAN-2020.ova` and located on a drive `VMs(E:)` in a folder `Virtual Machines\CSCI317-Janusz`.

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, 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 ? more information on how to use `SQLcl` and `SQL Developer`.

It is strongly recommended to drop the relational tables of `TPC-HR` benchmark database and to drop all tablespaces created in the previous Assignment. Then, recreate `TPC-HR` benchmark database in a way explained in Assignment 1, task 1. The other option is to import a new copy of virtual machine with Oracle 19c and again repeat Assignment 1, task 1.

Preface

A database administrator identified for the workload optimizations the following templates of five frequent data processing tasks:

task1.sql	sample SELECT statement used for power testing in TPC-HR benchmark
task2.sql	sample SELECT statement used for power testing in TPC-HR benchmark
task3.sql	SELECT statement using a stored function that never ends
task4.sql	JDBC application
task5.sql	JDBC application

Note, that a symbol ... included in the implementation of the tasks listed above represents a text or numeric constant that should be used when processing the tasks. Assume, the frequencies of processing are more or less the same for each task.

Objective

An objective of this project is to improve performance of all five data processing task listed above.

Constraints

(1) These days financial situation is pretty hopeless and we have no money to purchase a better hardware and/or to invest into a software license of more powerful database system.

(2) It is possible to invest no more than 100Mbytes of transient memory into the expansion of SGA and data buffer caches. All extension of SGA and data buffer caches must be documented in the final submission.

(3) It is also possible to invest no more than 300Mbytes of persistent storage, i.e. to double the size of TPC-HR benchmark database. All investments into persistent storage like for example additional indexes or materialized views must documented in the final submission.

(4) You are allowed to use the following of optimization of data processing techniques that have been discussed and practiced in the subject:
- data modeling techniques,
- persistent storage techniques,
- SQL programming techniques,
- Advanced/specialized SQL programming techniques,
- SQL+host language programming techniques,
- database server tuning techniques.

You can find an overview of the techniques listed above in a presentation 05 Principles of Database Performance Tuning.

Deliverables

Every implemented improvement of one or more given data processing tasks must be described in a separate document that contains:

(1) a short description of an improvement,
(2) information about the benefits from an improvement,
(3) information about the costs of an improvement, i.e. documented investments into transient and persistent storage and
(4) a report from implementation of an improvement.

For example, if you propose to denormalize a relational table then the benefits from denormalization can be reported as SQL processing plans with the cost estimations performed before and after an improvement, the costs can be measured as the total amount of additional persistent storage required to implement denormalization, and a report from implementation is a report from processing of SQL script that denormalizes a relational table.

The improvements in performance can be demonstrated with SQL processing plans obtained from `EXPLAIN PLAN` statement, application of `time` command for JDBC tasks and application of SQL scripts `utlbstat` and `utlestat`.

It is **not recommended** to use **AWR reports** and **tracing** due to the complexity of the reports. The reports from processing of SQL scripts `utlbstat` and `utlestat` must be limited to information that is only relevant to the improvements achieved. Complete reports will be ignored.

It is expected that each one of the data processing tasks listed above will be improved in at least one way. However, more improvements are mostly welcomed. The complete reports from the implemented improvements must be saved in the `pdf` files `report1.pdf`, `report2.pdf`, `report3.pdf`, `report4.pdf`, and `report5.pdf`.

Evaluations

A significant improvement in performance of each one of data processing tasks listed above is worth 6 marks.

Please remember, that an improvement applied to one data processing task may affect in a positive and/or negative way another data processing tasks. All such cases must be explained in the reports.

The marks will be deducted for the redundant improvements like multiple indexing of the same columns in a relational table, and for the improvements that change nothing like for example modification of SQL code such that a processing plan of a statement after modification is the same as for a statement before a modification.

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 **`report1.pdf`**, **`report2.pdf`**, **`report3.pdf`**, **`report4.pdf`**, and **`report5.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 **`CSCI317 (S121) Database Performance Tuning`**
(4) Scroll down a bit to a section **`PROJECT (Available from 14 June, 2021)`**
(5) Click at a link **`In this place you can submit the outcomes of the Final Project`**
(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 step (7) for the files **`solution2.pdf`**, **`solution3.pdf`**, **`solution4.pdf`**, and **`solution5.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`**

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

End of specification