School of Information Technology & Computer Science
University of Wollongong

Session: **3, July 2021**
Lecturer: **Janusz R. Getta**

CSCI317 Database Performance Tuning
Singapore 2021-3
Assignment 1

Published on 17 July 2021

Scope

This assignment includes the tasks related to denormalization of conceptual and relational schemas, estimation of efficiency of indexing, and finding the smallest set of indexes that improve performance of processing of a given collection of queries.

This assignment is due by **Saturday, 31 July 2020, 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 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 implement and to apply a simple performance measurement tool that uses information in V$ views.

Assume that we would like to speed up processing of a query implemented in a script file `task1.sql`. An idea is to eliminate `UNION` operations in order to minimize the total number of times a relational table `LINEITEM` is accessed while the query is processed.

We would like to compare the total number of read block operations performed before the improvements and after the improvements. To do we shall use a technique described in a presentation 04 Simple Performance Measurement Tools on slides 14 and 16 and also implemented in Cookbook, Recipe 2.2, How to use the dynamic performance views (V$ views) Step 9 and Step 10.

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

(1) First, the script finds the total number of physical and logical read block operations performed by a query implemented in a script file `task1.sql`.

(2) Next the script finds the total number of physical and logical read block operations performed by an improved implementation of the original query.

When ready start `SQLcl` client, connect to Oracle database server, and process SQL script `solutio1.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
```

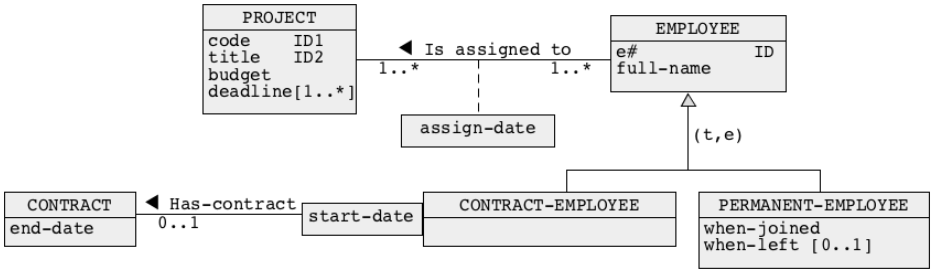
A report from processing of the script must have NO syntax errors !

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.

The following conceptual schema represents a database domain where the contract and permanent employees are assigned to the projects.



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

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

Find the full names of contract employees assigned to a project such that its present deadline is before a given date and such that end date of employees' contract is on or after a give date.

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

Find the full names of contract employees assigned to a project such that its present deadline is before 1 January 2018 and such that end date of employees' contract is on or after 1 January 2018 .

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.

In this task you must operate on the original state of a sample benchmark `TPC-HR` database. It is explained in **Assignment 1**, **Task 1** how to re-create `TPC-HR` database.

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

Find the total number of orders submitted by the customers from each region in a given year(`O_ORDERDATE`). List the names of regions (`R_NAME`) together with the total number of orders submitted by the customers from each region.

Next, apply denormalization of relational tables to speed up processing of the implemented `SELECT` statement. A year when the orders have been submitted is up to you. To test the improvements in performance create a script file `solution1.sql` that performs the following actions.

(1) First, the script finds and lists a query processing plan for the original `SELECT` statement.

(2) Next, the script applies denormalization to speed up the processing of a given `SELECT` statement in the best possible way. Please remember to re-load data after denormalization.

Note, that in this task, there is NO need for indexing, there is no need for creation of derived attributes and there is no need for creation of materialized views or any additional relational tables.

It is recommended to start from denormalization of 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.

(3) Next, the script finds and lists a query processing plan for `SELECT` statement, that implements the same query after denormalization performed in a step (2). Note, that after a denormalization implementation of the query must change because a structure of a database has change.

Of course, the estimated cost of processing in a query processing plan after denormalization must be significantly lower to than the estimated cost of processing before denormalization.

(4) Next, in order to further reduce the estimated costs processing after denormalization, the script creates an index to speed up processing of `SELECT` statement created in

step (3) and again it lists a query processing plan for `SELECT` statement created a step (3).

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 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` statement.

```
SELECT L_ORDERKEY, COUNT(*) TOT
FROM LINEITEM
WHERE L_QUANTITY = <value-1> AND L_DISCOUNT = <value-2>
GROUP BY L_ORDERKEY
HAVING COUNT(*) > 1
ORDER BY L_ORDERKEY
```

The values of placeholders `<value-1>` and `<value-2>` are up to you.

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

(1) First, the script finds and lists a query processing plan for `SELECT` statement given above. It is recommended to record the estimated values of query processing costs listed at the top of `Cost` column of query processing plan.

(2) Next, the script creates an index based on the single column. The index must improve performance of query processing in the best way for any single column key index.

(3) Next, the script finds a query processing plan for `SELECT` statement 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 an index created in a step (2).

(5) Next, the script creates an index based on two columns (composite index key), that improves performance of query processing in the best way for any index created over two columns.

(6) Next, the script finds a query processing plan for `SELECT` statement 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.

(7) Next, the script drops an index created in a step (5).

(8) Next, the script creates two single column indexes, that improve performance of query processing in the best way for any two single column indexes.

(9) Next, the script finds a query processing plan for `SELECT` statement 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 the indexes.

(10) Next, the script drops the indexes created in a step (8).

(11) Now, assume that we have enough persistent storage to create a single index on any number of columns as long as such index improves performance of query processing in the best way. Create an index on any number of columns, that improves performance of query processing in the best way.

(12) Next, the script finds a query processing plan for `SELECT` statement 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.

(13) Next, the script drops the indexes created in a step (11).

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 (SP321) 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