1School of Information Technology & Computer Science
University of Wollongong

Session: **3, July 2023**
Lecturer: **Janusz R. Getta**

CSCI317 Database Performance Tuning
Singapore 2023-3
Assignment 3

Published on 20 August 2023

Scope

This assignment includes the tasks related to estimation of efficiency of indexing, finding optimal clustering of relational tables and transformations of `SELECT` statements.

This assignment is due by **Saturday, 26 August 2023, 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 3 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 3** 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 (5 marks)

An objective of this task is to estimate the efficiency of indexing.

Assume that a relational table ORDERS contains information about the orders submitted by the customers.

```
ORDERS(order#, order_date, product, quantity, price_per_unit)
```

A relational table ORDERS has a primary key (order#).

Assume that:

(i) a relational table ORDERS occupies 1000 data blocks,
(ii) a blocking factor in a relational table ORDERS is 50 rows per block,
(iii) a relational table ORDERS contains information about 200 products,
(iv) a relational table ORDERS contains information about 100 prices per unit,
(v) a primary key is automatically indexed,
(vi) an attribute product is indexed,
(vii) all indexes are implemented as B*-trees with a fanout equal to 10,
(viii) a leaf level of an index on an attribute product consists of 30 data blocks,
(ix) a leaf level of an index on primary key consists of 200 data blocks,

For each one of the following queries briefly describe how the database system processes each query and estimate the total number of read block operations needed to compute each query. There is no need to perform the final computations. A correctly constructed formula filled with the appropriate constants is completely sufficient.

(1) 1 mark

```
SELECT product
FROM ORDERS
WHERE product = 'bolt' OR quantity = 100;
```

(2) 1 mark

```
SELECT count(*)
FROM ORDERS
WHERE product IN ('bolt', 'screw');
```

(3) 1 mark

```
SELECT product, COUNT(*)
FROM ORDERS
GROUP BY product
HAVING count(*) > 5;
```

(4) 1 mark
```
SELECT order#, product, quality
FROM ORDERS
ORDER BY order#, product;
```

(5) 1 mark
```
SELECT *
FROM ORDERS
WHERE order# = 12345 AND product = 'bolt';
```

(6) 1 mark
```
SELECT product
FROM ORDERS;
WHERE product = 'bolt' AND quantity = 100;
```

(7) 1 mark
```
SELECT product
FROM ORDERS;
```

(8) 1 mark
```
SELECT order#, product, quality
FROM ORDERS;
```

(9) 1 mark
```
SELECT order#, product
FROM ORDERS;
```

(10) 1 mark
```
SELECT order#
FROM ORDERS;
```

Deliverables
A file `solution1.pdf` with the comprehensive descriptions of query processing plans for each query and the estimations of the total number of read block operations needed to process each query.

Task 2 (5 marks)
Clustering
Consider a relational database created by the execution of a script `dbcreate.sql`. The database contains information about applicants for the positions advertised by employers, skills, skills possessed by applicants, skills needed for positions and skills required by other skills.

After loading data into the database, the relational tables have the following sizes:

`SKILL`	10 data blocks
`SREQUIRED`	100 data blocks
`APPLICANT`	1000 data blocks
`EMPLOYER`	200 data blocks
`EMPLBY`	2000 data blocks
`POSITION`	400 data blocks
`SPOSSESSED`	500 data blocks
`SNEEDED`	300 data blocks
`APPLIES`	600 data blocks

We would like to use clustering to improve performance of the following types of queries:
(i) Find full information about the applicants who applied for a position offered by a given employer.
(ii) Find full information about the applicants who posses a give skill.
(iii) Find full information about the skills possessed by a given applicant.
(iv) Find full information about the positions applied by a given applicant.
(v) Find full information about employers who advertise more than a given number positions.

Express the queries above as `SELECT` statements.

Assume, that queries *(i)* and *(ii)* are processed 10 times per day. Assume that queries *(iii)* and *(iv)* are processed 20 times per day. Assume that query *(v)* is processed 5 times per day.

Assume that if the relational tables `r` and `s` consist of b_r and b_s blocks then their sequential scan requires b_r and b_s read block operations and their join, i.e. `r JOIN s` requires $3 * (b_r + b_s)$ read block operations.

Use a method of finding suboptimal clustering explained to you during the lecture classes in a presentation 36 Clustering relational tables to find suboptimal clustering of the sample database that improves the performance of the queries listed above.

Deliverables
A file `solution3.pdf` with the following components:
(1) `SELECT` statements implementing the queries *(i), (ii)*, *(iii)*, *(iv)* and *(v)*.
(2) Computations of costs and benefits that lead to construction of clustering graph.

(3) A drawing of a clustering graph.
(4) Suboptimal clustering that improved performance of the queries *(i), (ii)*, *(iii)*, *(iv)* and *(v)*.

Use a method of finding suboptimal clustering explained to you during the lecture classes in a presentation 17 Clustering to find suboptimal clustering of the sample database that improves the performance of the queries listed above.

Task 3 (5 marks)
Transformation of queries

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.

```
(1) SELECT O_ORDERKEY, O_ORDERDATE, C_CUSTKEY
    FROM ORDERS LEFT OUTER JOIN CUSTOMER
                ON ORDERS.O_CUSTKEY = CUSTOMER.C_CUSTKEY;
(2) SELECT *
    FROM PART
    WHERE P_PARTKEY = 101
    INTERSECT
    SELECT *
    FROM PART
    WHERE P_NAME = 'bolt';

(3) SELECT P_NAME
    FROM PART P
    WHERE ( SELECT COUNT(*)
            FROM PART
            WHERE P.P_PARTKEY = PART.P_PARTKEY ) > 5;

(4) SELECT C_NAME, C_ADDRESS
    FROM CUSTOMER
    WHERE C_NAME = (SELECT DISTINCT C_NAME
                    FROM CUSTOMER
                    WHERE C_NAME = 'James');

(5) SELECT DISTINCT C_CUSTKEY, C_NAME
    FROM CUSTOMER CROSS JOIN ORDERS;
```

An objective of this task is to rewrite each one of the statements listed below into an equivalent `SELECT` statement and such that its processing costs are lower than the processing costs of the original `SELECT` statement.

Implement SQL script `solution3.sql` that performs the following actions.

(1) First, the script finds the query processing plans for each one of `SELECT` statements listed above.

    Use SQL script `showplan.sql` to list the processing plans.

(2) Next, the script finds query processing plans the improved `SELECT` statements such that the processing costs of each improved statement are lower than the processing costs of the original one.

Use SQL script `showplan.sql` to list the processing plans.

When processing an improved SQL script `solution3.sql` you must put the following `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 !

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 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`.

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.pdf`**, **`solution2.pdf`**, and **`solution3.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 (SP323) 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 3`**
(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`** and **`solution3.lst`**.
(9) Click at a button **`Submit assignment`** for the bottom of the current web page.
(10) 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.
(11) Click at a button **`Continue`**
(12) Check if **`Submission status`** is **`Submitted for grading`**.

End of specification