CSCI235 – Database Systems
2022 S4
Implementation Task 2

Due date: 8 November 2022

Scope

The Implementation of Task 1 is related to the contents of the topic on Indexing and PL/SQL.

This Implementation is due by **Tuesday, 8 November 2022, 9:00 pm Singapore time**. This task is worth 10% of the total assessment for the subject.

Only electronic submission through Moodle at:
https://moodle.uowplatform.edu.au/

is accepted. All email submission will be deleted and mark 0 (“zero”) will be awarded.

For Task 1, your report or output must include the listing of the execution of your SQL scripts that includes the explain plan statements, the creation of the indexes and the output of the show plan statements.

For Task 2, your report or output must include a listing of all PL/SQL statements processed.

To achieve the above-mentioned requirements, put the following SQL*Plus commands in all your scripts:

```
SPOOL file-name
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.

The submission procedure is explained at the end of this specification.

Specification

Step 1

If you have not done it yet, download the TPCHR sample database and load the sample TPCHR using user account CSCI235. This will make the user account CSCI235 the owner of the TPCHR database.

Step 2

In this step we shall use the relational tables included in a sample TPCHR benchmark database owned by the user CSC235. The conceptual schema of the sample database is included in the file tpchr.pdf.

The objective of this task is to find the **smallest** number of indexes that improve performance of a given collection of SELECT statements. We do not expect the best possible improvement in performance for each SELECT statement, however, processing of each SELECT statement must benefit from the existence of at least one of the indexes. An important objective is to minimize the total number of indexes created.

Task 1 (5.0 marks)
Indexing

Using the relational table LINEITEM of the sample database TPCHR, for each one of the queries listed below:

i. Find all the discount (l_discount) of all the items that are shipped (l_shipdate) most recently. Hint. Most recently mean the latest shipment date.

ii. Find the total number of items shipped by air (l_shipmode) in 1998 (l_shipdate).

iii. Find the order number (l_orderkey) and item number (l_linenumber) that have the highest discount (l_discount).

iv. Find the total number of items per line status (l_linestatus). List the line status and the total items per line status.

v. Find the order key (l_orderkey), line item number (l_linenumber), line status (l_linestatus), shipment date (l_shipdate) and shipment mode (l_shipmode) of all orders with the order number (l_orderkey) 1795718, 1799046, and 1794626.

a) Construct an SQL statement that produces the required output specified in the statement. **(1.5 marks)**

b) Find the **smallest** number of indexes that improve performance of a given collection of SELECT statements of a relational table LINEITEM. The smallest number of indexing means a database system will compute the five queries constructed in (a) using one or more indexes that you have created. Hint, you may create an index that can be used to compute more than one queries. Use the explain plan and show plan statements to justify your solutions. **(2.5 marks)**

Deliverables

A file solution1.pdf with CREATE INDEX statements that improve the performance of the queries listed (i, ii, iii, iv, and v above) and the execution plan generated.

Please remember that you must consider each one of the queries as an individual case! Please remember that all relational tables are large enough to make full table scans more time consuming that accessing the tables through an index! It means that any solution in which an index is not used for query processing is incorrect.

Task 2 (2.5 marks)

Stored PL/SQL Function

Implement a stored PL/SQL function numOfOrders that determines the total number of orders made by a customer. The function is to accept as parameter the name of customer and it must return a string of characters that contains the customer key, customer name, and the total number of orders made by each specified customer. Execute the stored PL/SQL function numOfOrders(custName). A fragment of sample printout is given below:

```
Customer-key Customer-name            Num-of-orders
---------------  -------------------------- ----------------
      38021 Customer#000038021              9
      38249 Customer#000038249             11
      38483 Customer#000038483              8
      41710 Customer#000041710             20
      44188 Customer#000044188             21
      44918 Customer#000044918              8
```

When finish the implementation and testing, execute an SQL script solution2.sql and record the results of processing in a file solution2.lst.

Deliverables

Submit a file solution1.lst with a report from processing of SQL script solution2.sql. The report MUST have no errors the report MUST list all SQL statements processed. The report MUST include ONLY SQL statements and control statements that implement the specifications of Task 2 and NO OTHER statements.

Task 3 (2.5 marks)
Stored PL/SQL procedure

Implement a stored PL/SQL procedure showCustomerOrders to list the customer number, names of customer, the orders number made, the order date, and the total price of order. The names of the customers must be listed in the ascending order, and the total price of order must be in descending order. If a customer did not make any order, list only the customer number and names. No details on order will be listed. Execute the stored PL/SQL procedure showCustomerOrders for the first 10 customers, that is, the customer key less than or equal to 10. A fragment of expected sample printout is given below.

```
1 - Customer#000000001:
           385825,                                      01-Nov-1995,
$254,563.49
          1374019,  05-Apr-1992,    $189,636.00
          1071617,                                      10-Mar-1995,
$156,748.63
           454791,  19-Apr-1992,     $78,172.70
          1590469,  07-Mar-1997,     $59,936.41
           579908,  09-Dec-1996,     $43,874.94
           430243,  24-Dec-1994,     $37,713.17
           1763205,  28-Aug-1994,     $18,112.74
          1755398,  12-Jun-1997,      $1,466.82


2 - Customer#000000002:
           164711,  26-Apr-1992,    $311,344.63
           905633,  05-Jul-1995,    $255,261.98
           135943,  22-Jun-1993,    $249,828.07
          1485505,                                      24-Jul-1998,
$230,389.81
          1192231,  03-Jun-1996,    $100,551.33
           224167,                                      08-May-1996,
$85,477.93
          1226497,  04-Oct-1993,     $81,926.50
           287619,  26-Dec-1996,     $16,946.76


3 - Customer#000000003:


4 - Customer#000000004:
             9154,  23-Jun-1997,    $336,929.37
            36422,  04-Mar-1997,    $266,881.39
           816323,  23-Jan-1996,    $265,441.63
          1603585,  26-Mar-1997,    $243,002.67
           306439,  17-May-1997,    $234,026.80
           895172,  04-Dec-1995,    $229,991.06
           916775,  26-Apr-1996,    $215,744.35
          1406820,  24-Feb-1996,    $195,275.97
           835173,  18-Aug-1993,    $187,151.50
          1490087,  10-Jul-1996,    $177,431.73
          1201223,  13-Jan-1996,    $155,250.48
           491620,  22-May-1998,    $154,443.20
           212870,  30-Oct-1996,    $152,662.65
          1718016,  30-Aug-1994,    $123,028.08
           859108,  20-Feb-1996,    $105,414.33
          1073670,  24-May-1994,     $76,478.32
           869574,  21-Jan-1998,     $58,714.84
           883557,  30-Mar-1998,     $44,043.89
          1774689,  08-Jul-1993,     $15,444.64


5 - Customer#000000005:
           374723,  20-Nov-1996,    $241,348.35
          1572644,  01-Jun-1998,    $201,565.95
          1478917,  06-Oct-1992,    $187,297.10
          1521157,  23-Aug-1997,    $141,934.18
           269922,  19-Mar-1996,    $122,008.56
          1177350,  03-Jul-1997,     $47,596.96
```

Deliverables

Submit a file solution3.lst with a report from processing of SQL script solution3.sql. The report MUST have no errors the report MUST list all SQL statements processed. The report MUST include ONLY SQL statements and control statements that implement the specifications of Task 3 and NO OTHER statements.

Submissions

This assignment is due by 9:00 pm (21:00 hours) **Tuesday, 8 November 2022, 9:00 pm Singapore time.**

Submit the files solution1.lst, solution2.lst, and solution3.lst **(or solution1.pdf, solution2.pdf, and solution3.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 successfully logged in, select a site CSCI235 (SP422) Database Systems
4) Scroll down to a section Submissions of Implementation Tasks
5) Click at Submit your Implementation Task 1 here link.
6) Click at a button Add Submission
7) Move the solution1.lst and solution2.lst **(or solultion1.pdf and**

solution2.pdf) into an area provided in Moodle. You can drag and drop files here to add them. You can also use a link *Add…*

8) Click at a button Save changes,

9) Click at check box to confirm authorship of a submission,

10) When you are satisfied, remember to click at a button Submit assignment.

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

Only one submission per student is accepted.

Implementation Task 1 is an individual assessed task and it is expected that all its tasks will be solved individually without any cooperation with the other students. Plagiarism is treated seriously. Students involved will likely receive zero. If you have any doubts, questions, etc. please consult your lecturer or tutor during lab classes or over e-mail.

End of specification