Assignment 2 (15% of total marks)

Due date: Tuesday, 8 November 2022 by 9:00 pm Singapore time

Scope:

The tasks of this assignment cover topics on **Indexing** and **PLSQL**.

Assessment criteria:

Marks will be awarded for:

- Correct,
- Comprehensive, and
- Appropriate

application of the materials covered in this subject.

Please read carefully information listed below.

This assignment contributes to 15% of the total assessment mark for the subject CSCI235.

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.

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

For all the implemented tasks, your report or output must include a listing of all PL/SQL statements processed. To achieve that 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.

Assignment Specification:

Task 1 (5.0 marks)
Indexing

a) Consider the relational schemas of the TPCHR sample database. Determine what index should be created to improve the performance of the queries listed below in the best possible way. Consider each one of the queries as an **individual** case. If you decide that an index should be created, then list the names of attributes that form an index key and write the **'create index'** statement to create the index. Remember that the order of attributes in an index key is important. If you decide not to create an index, **explain** why the performance will not suffer when a new index is not available. Assume that all relational tables are large enough to make full tables scan more time consuming than accessing the tables through an index.

i.
```
SELECT p_brand, p_type, p_retailprice
FROM part;
```

ii.
```
SELECT count(*)
FROM part
WHERE p_brand = 'Brand#18';
```

iii.
```
SELECT l_orderkey, count(*)
FROM lineitem
WHERE l_orderkey = 1184000
GROUP BY l_orderkey;
```

iv.
```
SELECT max(ps_suppkey)
FROM partsupp;
```

v.
```
SELECT ps_suppkey, ps_partkey, ps_supplycost
FROM partsupp
WHERE ps_suppkey = 3699
AND ps_partkey = 700;
```

Deliverables

Submit your spooled file solution1a.lst (or solution1a.pdf) that contains your SQL script and the output from the execution of the script. The report must have no errors related to the implementation of your task and it must list all PL/SQL and SQL statements processed.

Remember to set ECHO option of SQL*Plus to ON!

b) Consider the relational table **PARTSUPP** of the TPCHR sample database. Find SELECT statements that will use the index in the ways specified in the questions (i) to (iv) below. The values used to create the query such that the queries can meet the specified criteria is up to you, but the values must be sensible.

i. Execution of the first SELECT statement must traverse the index **vertically** then **horizontally** and it **must not** access the relational table PARTSUPP.

ii. Execution of the second SELECT statement must traverse the index **vertically** then **horizontally** and it **must** access the relational table PARTSUPP.

iii. Execution of the third SELECT statement must traverse the leaf level of the index **horizontally** and it **must not** access the relational table PARTSUPP.

iv. Execution of the fourth SELECT statement must traverse the leaf level of the index **horizontally** and it **must** access the relational table PARTSUPP.

Deliverables

Submit your spooled file solution1b.lst (or solution1b.pdf) that contains your SQL script and the output from the execution of the script. The report must have no errors related to the implementation of your task and it must list all PL/SQL and SQL statements processed.

Remember to set ECHO option of SQL*Plus to ON!

Task 2 (5.0 marks)
Stored PL/SQL procedure

a) Implement a **stored** PL/SQL procedure TotalNumOfOrdersMade that lists information about the customer and the details of the orders made by the customers. The procedure takes as parameter a nation name and computes the total number of orders made by all the customers from that nation. The information to be displayed include the customer key, customer name, total number of orders the customer made, and a list of the order key, order date, and order amount. An example of a segment of the output for the list of customers and the customers' order details is as follow:

```
Enter value for nationname: China
old  62:    nationName := upper('&nationName');
new  62:    nationName := upper('China');

Customer key: 6745, Customer#000006745
Total number of orders made: 33
```

Order key	Order date	Order amount
57092	02-05-1993	$115,109.85
127745	03-03-1993	$37,455.13
231142	01-08-1994	$143,399.55
249600	18-12-1994	$95,936.61
251712	11-04-1997	$92,074.43
268322	01-08-1996	$232,737.36
275591	20-01-1998	$127,975.27
303364	05-08-1997	$171,623.70
332932	14-12-1992	$120,153.30
335363	04-10-1992	$119,202.53
402593	27-05-1994	$81,549.75
422881	16-08-1997	$83,703.62
447428	02-08-1997	$154,502.67
450119	14-01-1994	$198,730.51
477027	26-01-1997	$125,974.62
493731	14-09-1997	$230,883.10
518695	27-08-1995	$150,625.98
528000	17-05-1997	$117,289.68
640161	14-07-1994	$153,853.47
650247	06-06-1998	$55,327.56
688134	05-10-1996	$56,012.56
839463	29-11-1993	$267,659.99
905700	10-01-1992	$354,849.22
981699	02-07-1993	$174,903.36
991591	11-04-1993	$66,109.85
1177796	20-07-1997	$244,462.49
1187936	24-03-1995	$2,622.27
1510466	12-04-1994	$150,769.50

1551328	15-12-1993	$105,257.07
1590017	24-01-1996	$96,769.73
1756133	27-08-1994	$290,800.68
1770752	10-12-1993	$329,544.51
1778023	23-07-1993	$137,299.46
	Total order amount:	$4,885,169.38

Customer key: 36343, Customer#000036343
Total number of orders made: 33

Order key	Order date	Order amount
73764	29-07-1998	$251,350.54
142119	06-04-1995	$85,705.43
207687	01-10-1997	$121,832.00
361286	07-07-1993	$162,899.61
455911	27-12-1996	$89,021.33
459968	24-05-1996	$122,759.72
469189	05-04-1994	$68,410.10
543553	19-05-1996	$7,160.87
547431	24-02-1997	$95,629.70
579013	21-04-1993	$171,255.98
755399	30-01-1996	$260,444.95
814246	04-01-1998	$273,316.49
852194	19-07-1993	$210,217.06
867748	23-06-1993	$32,561.87
871808	25-12-1995	$216,665.55
887045	24-01-1995	$105,670.34
888673	09-07-1994	$85,809.02
1014406	15-05-1992	$251,312.53
1054657	24-09-1997	$3,552.68
1074018	07-02-1994	$64,887.99
1089380	10-11-1994	$80,492.99
1175392	19-12-1997	$99,484.16
1242469	30-06-1998	$159,819.75
1287396	12-06-1995	$170,112.85
1341957	07-01-1998	$223,858.72
1345955	20-09-1992	$139,626.48
1380164	15-12-1993	$224,382.21
1527617	01-09-1997	$228,696.56
1633895	18-05-1994	$140,485.51
1669157	10-03-1997	$204,629.57
1680610	06-04-1998	$46,715.12
1716192	30-03-1998	$237,051.15
1778052	19-04-1998	$249,389.92
	Total order amount:	$4,885,208.75

It is up to you to decide if you want to handle exception in your procedure.

Deliverables

Submit your spooled file solution2a.lst (or solution2a.pdf) that contains your SQL script and the output from the execution of the script. The report must have no errors related to the implementation of your task and it must list all PL/SQL and SQL statements processed.

Remember to set ECHO option of SQL*Plus to ON!

b) Implement a **stored** PL/SQL function that finds a supplier who supplies the cheapest (lowest supply cost) or the dearest (highest supply cost) for a part specified by a user. The function should obtain the part key as its formal parameter, and it should return a string of values consisting of the supplier key, supplier name, and the cost (cheapest and dearest). Example of the string output are as follow:

```
Supplier with cheapest cost: 1344, Supplier#000001344, $172.09.
Supplier with dearest cost: 2882, Supplier#000002882, $791.07.

Supplier with cheapest cost: 770, Supplier#000000770, $74.55.
Supplier with dearest cost: 2308, Supplier#000002308, $826.35.
```

Next use the function to list the string output as specified above for part numbers 3753, 43064, 57574 and 60000 using a select statement.

Deliverables

Submit your spooled file solution2b.lst (or solution2b.pdf) that contains your SQL script and the output from the execution of the script. The report must have no errors related to the implementation of your task and it must list all PL/SQL and SQL statements processed.

Remember to set ECHO option of SQL*Plus to ON!

Submissions

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

Submit the files **solution1a.pdf**, **solution1b.pdf, solution2a.pdf**, and **solutions2b.pdf** through Moodle in the following way:

1) Zip all the files (Solution1.pdf, solution2.pdf, and solution3.pdf into one zipped folder.)
2) Access Moodle at **http://moodle.uowplatform.edu.au/**
3) 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
4) When successfully logged in, select a site CSCI235 (SP422) Database Systems
5) Scroll down to a section Submissions of Assignments
6) Click at Submit your Assignment 2 here link.
7) Click at a button Add Submission
8) Move the zipped file created in Step 1 above into an area provided in Moodle. You can drag and drop files here to add them. You can also use a link *Add…*
9) Click at a button Save changes,
10) Click at check box to confirm authorship of a submission,
11) 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.

Assignment 2 is an individual assignment, 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