

Assignment 2 (10% of total marks)

Due date: Sunday, 12 May 2024 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 10% 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:

Prelude

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.

Task 1 (2.5 marks)

Indexing

a) Create the following two indexes.

```
CREATE INDEX A2TASK1IDX1 ON ORDERS(O_CLERK, O_ORDERSTATUS);  
CREATE INDEX A2TASK1IDX2 ON LINEITEM(L_ORDERKEY);
```

For each one of the 10 types of different query or data manipulation statements listed below, provide one SQL statement consistent with the respective type and such that processing of SQL statement benefits from at least one of the indices over the relational tables ORDERS and LINEITEM.

- i. Projection queries (SELECT without WHERE clause).
- ii. Selection queries (SELECT with WHERE clause).
- iii. Join queries (join of two or more relational tables).
- iv. Queries processed by accessing only an index.
- v. Group by query (SELECT with GROUP BY clause and aggregation function).
- vi. Antijoin query (SELECT with NOT IN, NOT EXISTS).
- vii. Simple aggregation queries (SELECT with MAX, MIN, SUM, ... functions).
- viii. Update statement.
- ix. Delete statement.
- x. Query with sorting (SELECT with ORDER BY clause).

(4.0 marks)

Deliverables

Submit a file **solution1Output.lst** with a report from processing of SQL script **solution1.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 1 and NO OTHER statements.

Task 2 (2.5 marks)

PL/SQL Procedure

Preliminary actions

The Assignment 2 folder contains the SQL scripts dbCreate.sql, dbCreateSouth.sql and dbDrop.sql. Execute the script dbCreate.sql to create and to load the sample database.

Implement a stored PL/SQL procedure PROJECTGROUPS to list the project numbers, titles and the names of employees who work on each project.

The names of employees must be listed in the ascending order. Execute the stored PL/SQL procedure PROJECTGROUPS. A fragment of expected sample printout is given below.

```
1001 Computation: Alvin, Peter
1002 Study methods: Bob, Robert
1003 Racing car: Robert
...
```

When ready append all solutions for the questions above into an SQL script file solution2.sql. Next, process the entire script and spool the output to a spool file solution2Output.lst.

Deliverables

Submit your spooled file **solution2Output.lst** that contains your SQL script and the output from the execution of the script **solution2.sql**. 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 3 (2.5 marks)

PL/SQL Function

Implement a stored PL/SQL function `FINDEPENDENTS` that finds the names of dependents of an employee by given employee's number(e#). The function must return a string of characters that contains all dependents' names. All names of dependents must be separated by commas (,).

Execute the stored PL/SQL function `FINDEPENDENTS`. A fragment of sample printout is given below:

```
00100 Albert: Bolt, Edee, Judy
00110 Alvin:
00120 Alice: Blures, Edee, Kadi
...
```

When ready append all solutions for the questions above into an SQL script file `solution3.sql`. Next, process the entire script and spool the output to a spool file `solution3Output.lst`.

Deliverables

Submit your spooled file **`solution3Output.lst`** that contains your SQL script and the output from the execution of the script **`solution3.sql`**. 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 4 (2.5 marks)

Statement trigger

Implement and comprehensively test a **statement trigger** that enforces the following data access constraint.

A user is allowed to insert a row into a relational table Employee only one time per day.

When ready, create an SQL script solution4.sql with CREATE OR REPLACE TRIGGER statement and with SQL statements used for implementation of the constraint and for comprehensive testing of the trigger. A comprehensive test must consist of at least two cases: one when insertion is accepted and one when insertion is rejected.

Hints:

- (1) SYSDATE function returns a value of type DATE that contains the current year, month, day, time, hour, and second.
- (2) Time measurement with a precision of up to a single second is acceptable.
- (3) A value of an expression SYSDATE+1 is tomorrow the same time as now.
- (4) You are allowed to change the structures of a sample database.

Deliverables

Submit a file **solution4Output.lst** with a report from processing of SQL script **solution4.sql**. The report MUST have no errors other than reported by a trigger and the report MUST list all SQL statements processed. The report MUST include ONLY SQL statements and control statements that implement the specifications of Task 4 and NO OTHER statements.

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

Submissions

This assignment is due by 9:00 pm (21:00 hours) 12 May 2024, **Singapore time**.

Submit the files **solution1Output.lst** and **solution1.sql**, **solution2Output.lst** and **solution2.sql**, **solution3Output.lst** and **solution3.sql**, **solution4Output.lst** and **Solution4.sql** 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 (SP224) Database Systems
- 4) Scroll down to a section Submissions of Assignments
- 5) Click at Submit your Assignment 2 here link.
- 6) Click at a button Add Submission
- 7) 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...*
- 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.

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