Homework assignment #1 for CSC 352/452,

Due time/date 8:00 PM, 4/6/2022, Wednesday

Students in CSC352 and CSC452 share the same questions to avoid confusing. When grading the homework/midterm/final/course grade, the instructor will be lenient to students in CSC352 as requested by university policy.

Except the week before midterm/final, each week there will be a homework assignment posted on D2L before Wednesday. Due date/time will be 8:00 pm the next Wednesday. The website will not accept submission after the deadline. Later submission may send to the instructor via email. All submitted programs should have passed the compilation. Unless you explain the reasons, if a program is not runnable, it will not get partial credits. Later submitted homework will lose 20% credits per day delayed.

The main objective of this hw #1 is to make sure each student has finished the preparatory work. You need to set up your work environment, either use the SQL Developer on your own machine or on CDM lab machine, connecting to DePaul teaching Oracle server.

If you have no experience with SQL Developer before, please check the file ACADORADBPRD01Tutorial.docx and/or SQLDeveloper\_Quick\_config.docx posted in the folder Content -> Scripts for creating sample tables.

It is required to build the Oracle HR sample (seven) tables and the simplified Emp-Dept tables. The scripts are provided in the folder “Content -> Scripts for creating tables”.

What do you need to submit for your homework assignments?

One text file (such as hw1.txt) to Submissions section, Assignment1 on D2L site.

Please, use text format .txt file. Better to use Notepad. Optional, you may use WordPad to create a .rtf file. As you are submitting your file through D2L with your own account, the system will assign a special file name with info of the owner.

What should be included in the file ( see sample below):

* Your code for this question, and
* The output of that question (no screenshot); each output right after its code.

Such as below:

REM Q1.

SELECT employee\_id, Last\_name, First\_name, department\_id

FROM Employees

Where upper (last\_name) like 'ST%' ;

[output]

EMPLOYEE\_ID LAST\_NAME FIRST\_NAME DEPARTMENT\_ID

----------- ------------------------- -------------------- -------------

138 Stiles Stephen 50

REM Q2.

code for question #2

OUTPUT of Q2.

. . .

To begin with this homework assignment #1, please build the Oracle sample tables. Consider that some students may have different backgrounds in SQL, some hints are provided at the end of this file.

Questions #1 to #7 require the SQL select statements, based on table **Employees** and **Departments.** (10 points \* 10 questions = 100)

1. List the employee\_id, last name, first name, department ID for those employees that their last name starts with the letters “St”.
2. Display the employees’ employee\_id, full names (the format will be: last\_name, “,” one space, first\_name), salary and department ID for those who report to their immediate supervisor with ID as 145.
3. Display the employees’ employee\_id, full names (the print out format will be: last\_name, “,” one space, first\_name), salary and department ID for those whose immediate supervisor is John Russell (last name Russell, first name John). Sometimes, we do not know how the database stores the literals (all upper, Initial Capital, or all lower cases). Use the function *UPPER*, you can make your query case insensitive.
4. List the department ID, department name and the number (how many) of employees that work in that department. The output will be sorted in order of number of employees in ascending. Exclude those departments that have no employee yet.
5. List the full name (last name, first name), and salary for those employee(s) who is (are) not assigned to a department yet.
6. List the employee ID, last name, first name and department ID for those whose last name is not unique among the employees.
7. List the employee ID, last name, first name, their department ID, their immediate supervisor’s ID (the manager ID in employees table) and their department head’s ID, (the manager ID in departments table) for those employees who work in department (ID) 60 or department (ID) 70, and their immediate supervisor is not their department head/manager.

Questions 8, 9, and 10 are for PL/SQL coding.

1. Write an anonymous PL/SQL block.

In the block, declare a variable called l\_name, data type anchored as the same data type of last\_name in Employees table; also declare a variable named sal, data type as that of salary in Employees table.

In the executable section, the program will assign the value of l\_name as ‘Jackson’, salary as 7777. Then use the DBMS\_OUTPUT.PUT\_LINE (' ' ) command to print out the values of that employee’s name and salary. Please use the function TO\_CHAR to make the output of salary as popular money format like $7,777.00.

1. Each of the following two PL/SQL blocks declares a Boolean variable,

in the 9 (a), there is no error, neither output, please correct it to printout the statement;

When running the 9(b), you will get error, please correct it.

Set serveroutput on

REM (a)

DECLARE

v1 boolean := NULL;

Begin

IF v1 = Null THEN

DBMS\_OUTPUT.PUT\_LINE ('Value of v1 is NULL.');

END IF;

END;

REM 9(b)

DECLARE

v2 boolean := FALSE;

Begin

IF v2 = False THEN

DBMS\_OUTPUT.PUT\_LINE ('Value of v2 is ' || v2 || '.' ');

END IF;

END;

1. Write an anonymous PL/SQL block with a nested (inner) block.

In the out block, declare a variable called counter using integer as its data type with initial value of 101, also declare a variable called v1 using varchar2 (30) as its data type, assigning 'Adams' as its initial value.

In the inner block, declare a variable with the same name as “counter” with data type of integer, and assigning the initial value as 55.

In the executable section of inner block, print out the values of the variable counter declared in the outer block and inner block, also print out the value of variable v1 declared in the outer block.

**Some hints**.

Q1. Use like clause for Pattern match, such as below

Where last\_name *like* 'St%' ; -- this is case sensitive, or

Where UPPER (last\_name) *like* 'ST%' ;

Q2. Use concatenation operation to combine the last name and first name:

last\_name || ', ' || first\_name [AS] fullname, -- [ ] is optional

SELECT ... , TO\_CHAR (salary, '$999,999') salary, ...

-- use TO\_CHAR function to display currency

Q3. A subquery may make it easier in logic, such as

WHERE manager\_id =

( select employee\_id from employees where ...

Use where upper (last\_name) = 'RUSSELL' ...

that can make your where condition case insensitive.

Of course natural join will work too if you prefer: make join the table employees with itself,

such as

FROM employees e, employees m

WHERE m.employee\_id = e.manager\_id AND ...

Q4. Using “Group by” and function COUNT.

Information about which employee works for which department are described in the employee table; department name is listed only in departments table.

Notice that the select list must match the group by column\_list except the aggregation functions.

That means, if you have

“SELECT *d.Department\_ID, department\_name*, count (\*)... ”

as select list, then you will copy the whole select list (excluding the aggregation function)

“ Group by *d.DEPARTMENT\_ID, department\_name*”

Not include “ count(\*)”, as the system knows there is one value returned for each group.

The compiler does not know (or does not want to find out) that each department\_id has a unique department\_name. Group by all the columns in select list (except the aggregation function) makes sure there is only one row returned by each group.

Q5. The department\_id is not assigned, that can be translated into SQL language as

WHERE department\_ID *IS* null; -- “ = null ” is wrong in where condition.

Q6. A subquery may help:

where last\_name in

(select last\_name from employees

group by last\_name

having count(\*) > 1)

This kind subquery is useful to check the duplication (of values) in a table.

Q7. This question reminds you that the two columns with same name of “manager\_id” in

employees and departments tables do not have the same domain.

You need to join the two tables employees and departments;

FROM employees e, departments d

WHERE e.department\_ID = d.department\_ID and

e.department\_id in (60, 70) and

e.manager\_ID != d.manager\_id

Q8. PL/SQL. Simple questions, remember to run command

SET serveroutput on

You only need to run this command once for one session (or say for one log-in).

Q9. Please refer to the sample on page 3, note2\_PLSQL\_Fundamentals.docx.

Q10. Please refer to the Example 1 on page 6 in Note2\_PLSQL\_Fundamentals.docx.