Final (CSC352, CSC452)

June 8, 2022.

|  |  |
| --- | --- |
| Part I. multiple choices  4 points \* 5 = 20 | Part II. Programing  Five question, Total 80 points  15 + 15 + 25 + 15 + 10 |

Notice:

* CSC 352 and 452 students share the same questions. When grading, it will be lenient to students in 352 as requested by university policy.
* Please submit before 11:59 PM
* .txt format is preferred.

No need to repeat the questions. Suggestions formats :

Part I.

1. (a). -- assuming a is your choice.
2. (b).

...

Part II, submit your program codes and outputs as the homework

assignments.

* If you think there are mistakes/typos in the questions, send me email; or write down your comments on your submission. Do whatever you can. If it is my side problem, surely your grade will not be affected.
* If you cannot make your program pass the compilation, please explain your understanding of the possible reasons, thus you may get partial points for that question.

**Part I**. Multiple choices, choose one and only one. Please read carefully,

(4 points for each question, five questions in this part I)

1. Which is the wrong (or the least correct) statement?
2. User defined record type cannot be created in schema level.
3. When we invoke a function defined in a package, we should qualify the function with its package name, such as package\_name.function\_name.
4. If a function is defined in a package specification, then the executable codes of the function will be defined in the executable section of the package body.
5. Trigger may not be defined in a PL/SQL anonymous block
6. In the package body, all the variables declared in the declarative section are private to the package, not to the public.
7. Which is the wrong (or the least correct) statement?
8. The static SQL is sufficient when we know all the information needed at the compilation time.

1. In the dynamic SQL, the placeholders must be declared in the declaration section,
2. In the dynamic SQL program, DDL statements are possible be executed with “execute immediate” statement.
3. In a dynamic SQL statement, the name of placeholder must be represented with “:” before it, such as :place\_holder\_name.
4. In the dynamic SQL, the USING clause is needed if there are placeholders in the statement.
5. Which is the wrong (or the least correct) statement?
6. In current version, two record variables can be compared if they are same with a record level operation, such as “ IF record\_variable1 = record\_ variable2 THEN ... ”.

### Programmer may use Fetch-into clause to populate a user defined record.

1. To reference a field in a record type, the syntax is record\_type\_name.field\_name.
2. Program can assign one record variable values to another variable that has the same record type.
3. A record type can be defined as a partial column list of a database table. Such as only using columns employee\_id and salary from employees table as its record fields.
4. Among these unit types, which is not allowed to be created at schema level then stored in database?
5. Trigger
6. Package
7. Function
8. Procedure
9. Record
10. Which is the wrong (or the least correct) statement for function overloading?

### Function overloading requires the functions have the name, while their parameter number and/or data type are different. Say, both have one parameter, one data type is number, another is vharchar2.

1. Programmer may declare two functions with the same name in one package specification for overloading.
2. Programmer may declare two functions with the same name in one PL/SQL block for overloading.
3. Programmer may create two standalone functions with the same name to achieve the overloading feature.

**Part II**. **Coding** (five questions in this section, total 80 points)

**S**hort programs with similar requirements to your homework assignments,

Q1. (15 points) Create a standalone procedure, then write a PL/SQL block to invoke it.

Create a standalone procedure called Emp\_boss. That procedure will define one IN mode parameter named empno, data type as number. The procedure has three OUT parameters that will pass back the values of the employee’s last name, the immediate supervisor (manager) ID, and the manager’s last name to the invoking program.

Note, it will be easier to use two select-into statements to assign the values to the OUT parameters.

After having successfully compiled the procedure, you will write a PL/SQL block. The program will define a cursor to retrieve the employee ID’s whose salary is equal to or lower than $2,300. Then the program will invoke the procedure Emp\_boss to get these employees’ last names, their manager’s ID’s, and manager’s last names.

Your program will print out/display these info.

Q2. (15 points) Write a PL/SQL block that defines an overloading local (nested) procedure.

In this question, you are required to write a PL/SQL block. In the declarative section of this PL/SQL block, you will define a procedure called Emp\_data two times. The first one declares an IN mode parameter named id\_in with number as its data type. The procedure has two OUT mode parameters: lastname and salary, correspondent to the columns of last\_name and salary in the employees table. The second procedure has the same name, and its IN mode parameter has a name of email\_in with data type of varchar2. That procedure will return the same two out parameters.

In the executable section of the procedures, the first procedure will accept the employee\_id for the IN parameter, then retrieve the employee’s last name and salary; the second procedure will use the email address as input to get these output information.

In the executive section of this PL/SQL block, your program will call this procedure two times: the first time uses employee\_id as 100, in the second time invoking, it will use email address of ‘SKING’. Your program will print out the results.

Q3. (25 points) Create a Package that defines a record type and a functions that will return a record of that type.

This question is based on table employees.

You are required to create a package specification and a package body, that declares/defines a record type and a function that will return a variable of this record type.

The package has a name of Pkg\_EMP\_info. In the specification part, you declare a record type called EMP\_Name\_dname that has three fields as below:

empno, that is correspondent to the column of Employee\_ID in table employees,

full\_name, using first\_name ||' '|| last\_name as employee’s full name,

dept\_name, as the department name, (you can get the name of department via department id

from departments table).

You also declare/define a function called Emp\_Name\_Dept. That function has one IN parameter, named as id\_in, its data type same as the employee\_id, (number). The function returns a variable of the record type of EMP\_Name\_dname.

In the package body, you will have a full function defined. The function should accept the value of employee\_id via the IN parameter id\_in, retrieve the values of the three fields of the record.

In the executable section of the package body, you do not need to write any code for initializing. It can be simply as “Begin null ; END” , or skipped the “begin null; ”, only with “END; ”.

After having successfully compiled the package specification and its body, write a PL/SQL block. It will repeated call that function defined in the package for each employee in the department 60, and print out the record information for these employees in this department.

Q4. (15 points) Create a log table, create a Trigger, then test it and display the query result of the select statement from the log table.

This question is based on employees table. Create a conditional predicates row level trigger called Employee\_trg. Whenever there is an insert or an update on column of manager\_id on the table employees, the trigger will be fired, and for each record changed, one row will be inserted into a table Emp\_log.

You need to create this table Emp\_log with seven columns. They are employee ID, last\_name, the old and new manager\_id (for before and after the update on that special column manager\_id), the user who issued this command, the date that occurred, and a very short description of the command.

Once you have successfully created the table and the trigger, test the trigger with the following two commands, and display the correspondent records in the log table.

UPDATE employees set manager\_id = 120 WHERE department\_id = 90;

INSERT INTO employees VALUES

(300, 'Fred', 'Adams', 'Fadams', '515.123.5555', '11-MAY-2022', 'IT\_PROG', 50000, NULL, NULL, 60);

-- SELECT \* from Emp\_log;

-- rollback;

Q5. (10 points) Dynamic SQL.

Write a PL/SQL block program. In this block, it will run “EXECUTE IMMEDIATE” for a dynamic query string. That query will retrieve the number (how many) of employees that report to a special supervisor (manager\_id).

Your program will declare a string variable, then define the query statement for this string in the executable section.

The string is a select statement that will return the number (how many) whose manager\_id equals to a certain dummy place holder. A basic select statement string is as below:

'select count (\*) from employees where ... ';

The WHERE condition should has a placeholder for manager\_id.

Your code will run this query with a value of 103 to replace this placeholder: only at run time to replace the placeholder.

The program will print out the result, the number (how many) of employees reporting to this supervisor.

\*\* The End of Final. \*\*