**CC-215-L: Database Systems Lab**

*Faculty of Computing & Information Technology*



**BS(CS) Morning - Fall 2021, Semester Spring 2023**

**LAB – 11**

**Course & Lab Instructor:** Dr. Asif Sohail

**Objectives:**

1. PL/SQL Cursors
2. Procedures & Functions

*Allowed time: 120 mins.*

***Instructions:***

1. Gossips are not allowed.
2. Teacher assistants are for your help, so be nice with them. Respect them as they are teaching you. Raise your hands if you have some problem and need help from TA. Avoid calling them by raising your voice and disturbing the environment of Lab.
3. TA may deduct your marks for any kind of ill-discipline or misconduct from your side.
4. Evaluation will be considered final and you cannot debate for the marks. So, focus on performing the tasks when the time is given to you.
5. Paste the query as well as result table screenshot as a result of each task

**Sample:**

**Display All the Employees from emp table**

**Solution:**

**Select \* from emp**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **EMPNO** | **ENAME** | **JOB** | **MGR** | **HIREDATE** | **SAL** | **COMM** | **DEPTNO** |
| 7369 | SMITH | CLERK | 7902 | 12/17/1980 | 800 | - | 20 |
| 7499 | ALLEN | SALESMAN | 7698 | 02/20/1981 | 1600 | 300 | 30 |
| 7521 | WARD | SALESMAN | 7698 | 02/22/1981 | 1250 | 500 | 30 |
| 7566 | JONES | MANAGER | 7839 | 04/02/1981 | 2975 | - | 20 |
| 7654 | MARTIN | SALESMAN | 7698 | 09/28/1981 | 1250 | 1400 | 30 |

**Task 01: [20 Marks]**

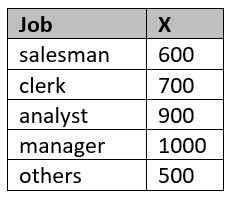
1. Create a PL/SQL block that computes the number of records of emp table using a cursor.
2. Create a Procedure **top\_earners\_per\_department** that identifies the top earners in each department based on their salaries and displays the result.

1. Create a PL/SQL functions find\_dname(deptno number) that receives deptno as a parameter and returns the name of the department call this function to display the dname of employess with their names deptno and sal.

1. Create a PL/SQL function **raise\_sal** that raises the sal of all the employees of emp\_accounts table by 10% of their exisiting sal.
2. Create a PL/SQL function **sal\_status** that receives salary of the employees and returns either "above average salary" or "below average salary" on the basis of the average salary of all the employees.

**Task 02: [70 Marks]**

1. Create a PL/SQL procedure **aggregate** that performs all the built-in aggregate functions on sal column of employee table. The procedure should display count, sum, average, max, and min without using any one of the built-in group functions.
2. Suppose that the maximum years of service of an employee is 30 years. The salary of the employee is raised by x% of the current salary of the year. The number x is passed as parameter to a PL/SQL procedure. The default value of the parameter x is 10. The procedure should display the empno, ename, hiredate, retirement date, salary at the time of hiring, and the salary at the time of retirement of the seniormost employee.
3. Create a PL/SQL function **bonus** (empno) that computes the bonus amount (=sal\*years\_of\_service/100+x). The value of x is determined by the job of the employee as per the following policy:



CREATE OR REPLACE FUNCTION BONUS(EMPNO EMP.EMPNO%TYPE) RETURN EMP.EMPNO%TYPE IS

NUMOFYEARS NUMBER;

SALARY EMP.SAL%TYPE;

BONUS NUMBER;

JOB EMP.JOB%TYPE;

X NUMBER;

BEGIN

SELECT SAL,JOB,MONTHS\_BETWEEN(SYSDATE,HIREDATE)/12 INTO SALARY,JOB,NUMOFYEARS FROM EMP WHERE EMPNO = BONUS.EMPNO;

CASE JOB

WHEN 'SALESMAN' THEN X:=600;

WHEN 'CLERK' THEN X:=700;

WHEN 'ANALYST' THEN X:=900;

WHEN 'MANAGER' THEN X:=1000;

ELSE X:=500;

END CASE;

BONUS:=TRUNC(SALARY\*NUMOFYEARS/100)+X;

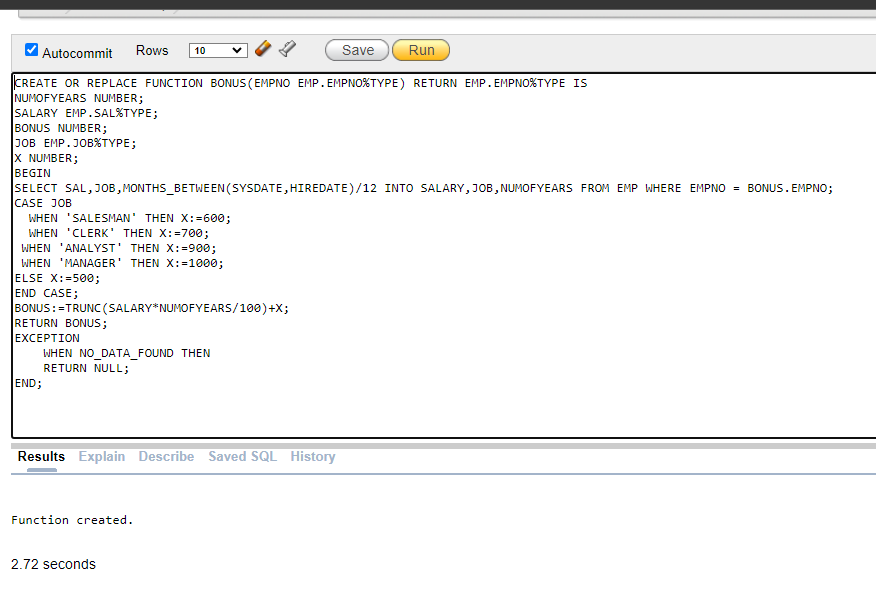
RETURN BONUS;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN NULL;

END;



1. Create a PL/SQL function **salary\_encrypt** that receives salary of the employees and encrypts the sal of an employee as per the following rules:

# is equivalent of thousands

@ is equivalent of hundreds

& is equivalent of tens

The digits 0,1,2,…,9 are replaced by a,b,c,…,j respectively.

Hint: translate('ab12cd34','abcd','wxyz;) = wx12yz34;

It is assumed that the sal can be up to 4 digits, and the sal is multiple of 10.

Examples:

Code for 4530: e#f@d&

Code for 8060: i#a@g&

CREATE OR REPLACE FUNCTION ENCRYPT\_SAL(SAL EMP.SAL%TYPE) RETURN CHAR IS

SAL\_CODE VARCHAR2(20);

SALARY VARCHAR2(10):=TO\_CHAR(SAL);

THOUSANDS CHAR(1);

HUNDREDS CHAR(1);

TENS CHAR(2);

BEGIN

SALARY:=LPAD(TO\_CHAR(SALARY),4,0);

THOUSANDS:=SUBSTR(SALARY,1,1);

SALARY:=MOD(SALARY,1000);

HUNDREDS:=SUBSTR(SALARY,2,1);

SALARY:=MOD(SALARY,100);

TENS:=SUBSTR(SALARY,3,1);

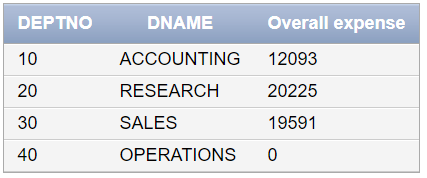
SAL\_CODE:=THOUSANDS||'#'||HUNDREDS||'@'||TENS||'&';

SAL\_CODE:=THOUSANDS(SAL\_CODE,'0123456789','ABCDEFGHIJ');

RETURN SAL\_CODE;

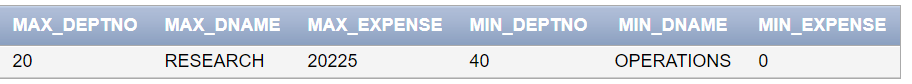
END;

1. Create a PL/SQL function **salary\_decrypt** that receives the encrypted code of the salary that is encrypted using the rules of the last task. The function should decrypt the code. Verify the function using dual table.
2. Write a PL/SQL function **dept\_expense**(deptno) to calculate the total expense incurred by the company for a specified department. Consider the following:
   1. Include the sum of salaries (SAL) and commissions (COMM) for all employees in the department.
   2. Use the previously defined **bonus** function to compute the total bonus for that department.
   3. Assume that the total expense includes the total of salaries, commissions, and bonuses for all employees in the specified department.
   4. Return the calculated expense in function.
   5. Displays the department wise expense of the company



1. Extend the above code to display highest expense dept and lowest one. (5)

**Note:** (You have to do this by expense fun you created.)



**Task 03: Post Lab: [25 Marks]**

1. Use this getUserInput function in a Procedure named as “Find\_Employee” which takes input from user using the above function and find the employee/s with that name. Use Cursors for handling more than one rows. Display all details of the employee/s. Also use appropriate exceptions in case of “No data found”.
2. Create a function named as getUserInput which takes no parameter. It takes input (char / varchar2 type) from user. If the input contains any non-alphabetic character, the function will prompt the user to enter valid input and keeps on looping. The function returns the valid input in the end.