

Assignment 17.04.2023

Name – Saikat Sheet.

University Roll – 18700120024.

The table on which operations are done

EMP_ID	FIRST_NAME	LAST_NAME	DOJ	SALARY
100	John	Doe	04/17/2022	50000
101	ALICE	SARKAR	06/12/2022	45000
102	ROHAN	MONDAL	08/15/2022	65000
103	RAJ	SINHA	09/16/2022	70000
104	ALISHA	SARKAR	04/14/2022	15000

- 1) **Q.1** Write PL/SQL code block to increment the employee's salary by 1000 whose **emp_id is 102** from the given table below.

```
DECLARE
  v_emp_id NUMBER;
  new_sal NUMBER;
BEGIN
  v_emp_id := 102;

  UPDATE employees
  SET salary = salary + 1000
  WHERE emp_id = v_emp_id;
  DBMS_OUTPUT.PUT_LINE('Salary updated');
  salary INTO new_sal from
  employees
  where emp_id = v_emp_id; DBMS_OUTPUT.PUT_LINE('UPDATED
  SALARY: '||new_sal);

END;
```

Output:

```
Salary updated
UPDATED SALARY:66000

1 row(s) updated.
```

- 2) **Q.2** Create a Procedure to accept an Emp_id, and a salary increase amount, if Emp_id is not found or current salary is NULL then raise exceptions otherwise display total salary

```
CREATE OR REPLACE PROCEDURE INCSALARY(  
EMP_ID NUMBER,  
INCREASE_AMT NUMBER) IS  
TOTAL_SALARY NUMBER;  
BEGIN  
SELECT salary INTO TOTAL_SALARY  
FROM employees  
WHERE emp_id = EMP_ID;  
IF TOTAL_SALARY IS NULL THEN  
RAISE_APPLICATION_ERROR(-20001, 'Employee not found or has null salary'); END IF;  
  
UPDATE employees  
SET salary = salary + INCREASE_AMT  
WHERE emp_id = EMP_ID;  
  
salary INTO TOTAL_SALARY  
FROM employees  
WHERE emp_id = EMP_ID;  
  
DBMS_OUTPUT.PUT_LINE('Total salary of employee ' || EMP_ID || ': ' ||  
TOTAL_SALARY);  
END;
```

Output:



Procedure created.

Q.3 Write a Function to find out total annual income for the employee, whose **emp_id** is **102**. .

```
CREATE OR REPLACE FUNCTION get_annual_income(p_emp_id IN NUMBER) RETURN  
NUMBER IS
```

```
    v_salary NUMBER;  
BEGIN  
    SELECT salary*12 INTO v_salary  
    FROM employees  
    WHERE emp_id = p_emp_id;
```

```
    RETURN v_salary;  
END;
```

```
DECLARE
```

```
    v_salary NUMBER;  
BEGIN  
    v_salary := get_annual_income(102); DBMS_OUTPUT.PUT_LINE('Salary of  
employee 102 is ' || v_salary); END;
```

```
OUTPUT
```

```
Salary of employee 102 is 792000
```

```
Statement processed.
```

```
0.00 seconds
```

4) Write a PL/SQL Block to insert one row in employee table with emp_id and other fields. Display appropriate message using exception handling on duplication entry of emp_id.

```
DECLARE
v_emp_id NUMBER := 105;
v_first_name VARCHAR2(50) := 'RAHUL';
v_last_name VARCHAR2(50) := 'SEN';
v_doj DATE := TO_DATE('2022-10-17', 'YYYY-MM-DD');
v_salary NUMBER := 60000;
v_count NUMBER;
BEGIN
    SELECT COUNT(*)
    INTO    v_count
    FROM    employees
    WHERE   emp_id = v_emp_id;

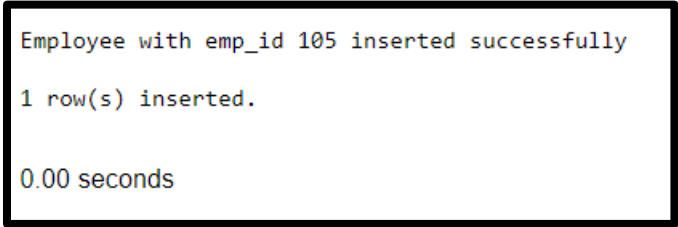
    IF v_count > 0 THEN
        RAISE_APPLICATION_ERROR(-20001, 'Employee with emp_id ' || v_emp_id || ' already
exists');
    ELSE
        INSERT INTO employees (emp_id, first_name, last_name, doj, salary) VALUES
        (v_emp_id, v_first_name, v_last_name, v_doj, v_salary);

        DBMS_OUTPUT.PUT_LINE('Employee with emp_id ' || v_emp_id || ' inserted
successfully');
    END IF;

    COMMIT;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLCODE || ' - ' || SQLERRM);
        ROLLBACK;
END;
```

OUTPUT:

When 105 is Given as INPUT



```
Employee with emp_id 105 inserted successfully

1 row(s) inserted.

0.00 seconds
```

When 102 is Given as Input

```
Error: -20001 - ORA-20001: Employee with emp_id 102 already exists  
1 row(s) inserted.  
  
0.01 seconds
```