Assignment 12

Name – Tanmay Garg
Batch – 3ENC1
Roll No. – 101915001

PL/SQL Lab Assignment-4

EMP(ENO,ENAME, JOB, SALARY, COMMISSION, DEPTNO)

1. Write a PL/SQL code to create a stored procedure fire_employee to delete employee on the basis of employee number.

SQL Worksheet | Clear | Find | Actions | Save | Run

```
1  CREATE TABLE emp(
2  eno INT PRIMARY KEY,
3  ename VARCHAR(15),
4  salary INT ,
5  commission INT,
6  deptno INT
7  );
8
9  INSERT INTO emp VALUES(1,'Amy',40000,35000,2);
10 INSERT INTO emp VALUES(2,'Brian',12500,10000,10);
11 INSERT INTO emp VALUES(3,'Cathy',20000,5000,4);
12 INSERT INTO emp VALUES(4,'Derek',32000,12000,2);
13 INSERT INTO emp VALUES(5,'Emily',15000,5000,12);
14
15 SELECT *FROM emp;
```

ENO	ENAME	SALARY	COMMISSION	DEPTNO
1	Amy	40000	35000	2
2	Brian	12500	10000	10
3	Cathy	20000	5000	4
4	Derek	32000	12000	2
5	Emily	15000	5000	12

Download CSV
5 rows selected.

```
16
17 CREATE OR REPLACE PROCEDURE fireEmployee AS empNo INT:=3;
18 BEGIN
19 DELETE FROM emp WHERE eno = empNo;
20 END;
21
22 EXEC fireEmployee;
23 SELECT *FROM emp;
```

ENO	ENAME	SALARY	COMMISSION	DEPTNO
1	Amy	40000	35000	2
2	Brian	12500	10000	10
4	Derek	32000	12000	2
5	Emily	15000	5000	12

Download CSV
4 rows selected.

2. Write a PL/SQL code of a local procedure raise_salary, which accepts two parameters empid and bonus to be added to the salary. It increases the salary of theemployee and update it to the database.

```
CREATE OR REPLACE PROCEDURE raiseSalary AS
empId INT:= 2;
bonus INT:= 500;
BEGIN
UPDATE emp
SET salary=(SELECT salary FROM emp WHERE eno=empId)+bonus
WHERE eno=empId;
END;

EXEC raiseSalary;
SELECT *FROM emp;
```

ENO	ENAME	SALARY	COMMISSION	DEPTNO
1	Amy	40000	35000	2
2	Brian	13000	10000	10
4	Derek	32000	12000	2
5	Emily	15000	5000	12

Download CSV

4 rows selected.

3. Write a function to add two numbers.

```
CREATE OR REPLACE FUNCTION sumNums(n1 INT,n2 INT) RETURN INT
AS
BEGIN
RETURN n1 + n2;
END;
SELECT sumNums(98,150) FROM dual;
```

SUMNUMS(98,150)
248

Download CSV

4. Write a stored function that accepts department number and return total salary ofthat depart

```
CREATE OR REPLACE FUNCTION ts(deptno INT) RETURN INT AS
dn INT:=deptno;
sums INT;
BEGIN
SELECT SUM(salary) INTO sums FROM emp WHERE deptno=dn;
RETURN sums;
END;
SELECT ts(2) FROM dual;
```

TS(2)
72000

Download CSV

5. Create a trigger on the emp table, which shows the old values and new valueof ename after every updation on ename of emp table.

```
54
55  CREATE OR REPLACE TRIGGER name_change
56  BEFORE UPDATE ON emp
57  FOR each ROW
58  BEGIN
59  dbms_output.put_line ('Updated name :'||:NEW.ename);
60  dbms_output.put_line ('Previous name :'||:OLD.ename);
61  END;
62
63  UPDATE emp SET ename='Debby' WHERE eno=1;
64
65
```

```
1 row(s) updated.
Updated name :Debby
Previous name :Amy
```

6. Write a trigger to ensure that commission of employee cannot be greaterthan his salary.

```
71
72  CREATE OR REPLACE TRIGGER commExcep
73  after INSERT ON emp
74  FOR each ROW
75  BEGIN
76  IF(:NEW.salary<:NEW.commission) THEN
77  raise_application_error(-20000, 'SALARY cant be less than the COMMISSION');
78  END IF; END;
79
80  INSERT INTO emp VALUES(6,'Tim',10000,850000,4);
81
82
83
```

```
ORA-20000: SALARY cant be less than the COMMISSION ORA-06512: at
"SQL_NOUDYSHQNWVUINDBBYRLSHUPV.COMMEXCEP", line 3
ORA-06512: at "SYS.DBMS_SQL", line 1721
```