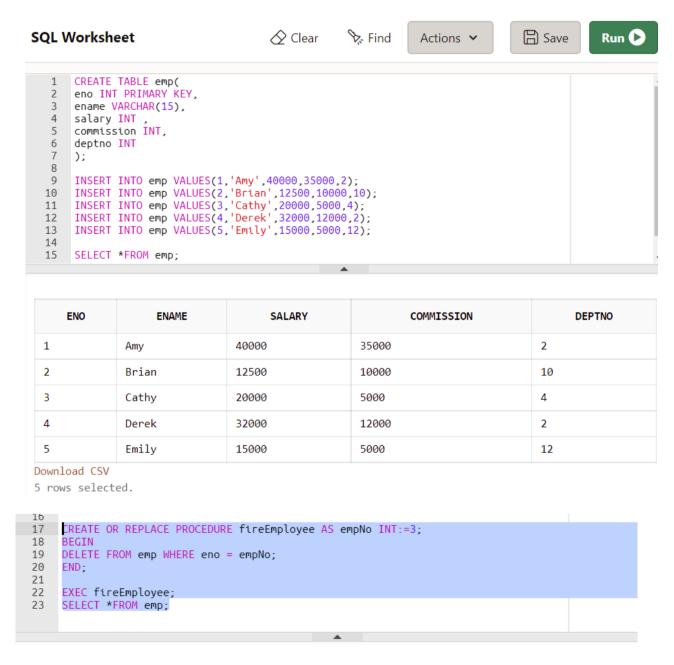
Name – Tanmay Garg Batch – 3ENC1 Roll No. – 101915001

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.



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 the employee and update it to the database.

```
CREATE OR REPLACE PROCEDURE raiseSalary AS
27
     empId INT:= 2:
     bonus INT:= 500;
28
29
     BEGIN
    UPDATE emp
30
31
     SET salary=(SELECT salary FROM emp WHERE eno=empId)+bonus
32
     WHERE eno=empId;
33
     END;
34
35
    EXEC raiseSalary;
     SELECT *FROM emp;
36
37
38
39
40
```

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

40
BEGIN
41
RETURN n1 + n2;
END;
SELECT sumNums(98,150) FROM dual;

44
45
46
```

```
SUMNUMS(98,150)
248
```

Download CSV

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

```
45
    CREATE OR REPLACE FUNCTION ts(deptno INT) RETURN INT AS
46
    dn INT:=deptno;
47
    sums INT;
48
    BEGIN
    SELECT SUM(salary) INTO sums FROM emp WHERE deptno=dn;
49
50
    RETURN sums;
51
    END;
52
    SELECT ts(2) FROM dual;
53
54
```

```
TS(2)

72000

Download CSV
```

5. Create a trigger on the emp table, which shows the old values and new value of 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
      dbms output.put line ('Updated name :'||:NEW.ename);
 59
      dbms output.put line ('Previous name :'||:OLD.ename);
 60
 61
      END:
 62
      UPDATE emp SET ename='Debby' WHERE eno=1;
 63
 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
     IF(:NEW.salary<:NEW.commission) THEN</pre>
76
     raise_application_error(-20000, 'SALARY cant be less than the COMMISSION');
77
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
```