Excersise 2: Data Definition Language(DDL) commands

Query 1:

```
CREATE TABLE emp(
  empno INT,
  ename VARCHAR(10),
  designation VARCHAR(10),
  salary int );
```

Output:

```
Table is Created
```

Query 2:

```
DESC emp;
```

Output:

```
Name                                      Null?    Type
----------------------------------------- -------- ----------------------------
EMPNO                                              NUMBER(38)
ENAME                                              VARCHAR2(10)
DESIGNATION                                        VARCHAR2(10)
SALARY                                             NUMBER(38)
```

Query 3:

```
CREATE TABLE emp1 AS SELECT * FROM emp;
```

Output:

```
Table is Created
```

```
DESC emp1;
```

Output:

```
Name                                      Null?    Type
----------------------------------------- -------- ----------------------------
EMPNO                                              NUMBER(38)
ENAME                                              VARCHAR2(10)
DESIGNATION                                        VARCHAR2(10)
SALARY                                             NUMBER(38)
```

Query 4:

```
CREATE TABLE emp2 AS SELECT empno, ename FROM emp;
```

Output:

```
Table is Created
```

```
DESC emp2;
```

```
Name                                      Null?    Type
----------------------------------------- -------- ----------------------------
EMPNO                                              NUMBER(38)
ENAME                                              VARCHAR2(10)
```

Query 5:

```
ALTER TABLE emp MODIFY empno NUMBER(6);
```

Output:

```
Table is Altered
```

```
DESC emp;
```

Output:

```
Name                                      Null?    Type
----------------------------------------- -------- ----------------------------
EMPNO                                              NUMBER(6)
ENAME                                              VARCHAR2(10)
DESIGNATION                                        VARCHAR2(10)
SALARY                                             NUMBER(38)
```

Query 6:

```
ALTER TABLE emp MODIFY (empno NUMBER(7), ename VARCHAR(12));
```

Output:

```
Table is Altered
```

DESC emp;

Output:

Name	Null?	Type
EMPNO		NUMBER(7)
ENAME		VARCHAR2(12)
DESIGNATION		VARCHAR2(10)
SALARY		NUMBER(38)

Query 7:

ALTER TABLE emp ADD qualification VARCHAR(6);

Output:

Table is Altered

DESC emp;

Output:

Name	Null?	Type
EMPNO		NUMBER(38)
ENAME		VARCHAR2(10)
DESIGNATION		VARCHAR2(10)
SALARY		NUMBER(38)
QUALIFICATION		VARCHAR2(6)

Query 8:

ALTER TABLE emp ADD (DOB DATE, DOJ DATE);

Output:

Table is Altered

DESC emp;

Output:

Name	Null?	Type
EMPNO		NUMBER(38)
ENAME		VARCHAR2(10)
DESIGNATION		VARCHAR2(10)
SALARY		NUMBER(38)
QUALIFICATION		VARCHAR2(6)
DOB		DATE
DOJ		DATE

Query 9:

DROP TABLE student;

Output:

Table is dropped

Query 10:

ALTER TABLE emp DROP COLUMN DOJ;

Output:

Table is altered

DESC emp;

Output:

Name	Null?	Type
EMPNO		NUMBER(38)
ENAME		VARCHAR2(10)
DESIGNATION		VARCHAR2(10)
SALARY		NUMBER(38)
QUALIFICATION		VARCHAR2(6)
DOB		DATE

Query 11:

ALTER TABLE emp DROP(DOB,qualification);

Output:

Table is altered

DESC emp;

Output:

```
Name                                      Null?    Type
----------------------------------------- -------- ----------------------------
EMPNO                                              NUMBER(38)
ENAME                                              VARCHAR2(10)
DESIGNATION                                        VARCHAR2(10)
SALARY                                             NUMBER(38)
```

Query 13:

RENAME emp To employee;

Output:

```
Table is altered
```

DESC employee;

Output:

```
Name                                      Null?    Type
----------------------------------------- -------- ----------------------------
EMPNO                                              NUMBER(38)
ENAME                                              VARCHAR2(10)
DESIGNATION                                        VARCHAR2(10)
SALARY                                             NUMBER(38)
```

Exercise 3: Data Manipulation Language(DML) Commands

Query 1:

```
INSERT INTO emp VALUES(101,'NAGARAJ','LECTURER',15000)
```

Output:

```
1 row created
```

Query 2:

```
SELECT * FROM emp;
```

Output:

```
EMPNO ENAME          DESIGNATION          SALARY
---------- -------------------- -------------------- ----------
     101 NAGARAJ        LECTURER              15000
```

Query 3:

```
INSERT INTO emp VALUES(&empno, '&ename', '&designation', &salary);
```

Output:

```
Enter value of empno: 102
Enter value of ename: SARAVANAN
Enter value of designation: LECTURER
Enter value of salary: 15000

old1: INSERT INTO emp VALUES(&empno,'&ename','&designation',&salary)
new1:INSERT INTO emp VALUES(102,'SARAVANAN','LECTURER',15000)

1 row created
```

```
INSERT INTO emp VALUES(&empno, '&ename', '&designation', &salary);
```

Output:

```
Enter value of empno: 103
Enter value of ename: PANNERSELVAM
Enter value of designation: ASST.PROF
Enter value of salary: 20000

old1: INSERT INTO emp VALUES(&empno,'&ename','&designation',&salary)
new1:INSERT INTO emp VALUES(103,'PANNERSELVAM','ASST.PROF',20000)

1 row created
```

INSERT INTO emp VALUES(&empno, '&ename', '&designation', &salary);

Output:

```
Enter value of empno: 104
Enter value of ename: CHARULATHA
Enter value of designation: HOD, PROF
Enter value of salary: 45000

old1: INSERT INTO emp VALUES(&empno,'&ename','&designation',&salary)
new1:INSERT INTO emp VALUES(104,'CHARULATHA','HOD, PROF',45000)

1 row created
```

SELECT * FROM emp;

Output:

EMPNO	ENAME	DESIGNATION	SALARY
101	NAGARAJ	LECTURER	15000
102	SARAVANAN	LECTURER	15000
103	PANNERSELVAM	ASST.PROF	20000
104	CHARULATHA	HOD, PROF	45000

Query 4:

UPDATE emp SET salary = 16000 WHERE empno = 101;

Output:

```
1 row updated
```

SELECT * FROM emp;

Output:

EMPNO	ENAME	DESIGNATION	SALARY
101	NAGARAJ	LECTURER	16000
102	SARAVANAN	LECTURER	15000
103	PANNERSELVAM	ASST.PROF	20000
104	CHARULATHA	HOD, PROF	45000

Query 5:

UPDATE emp SET salary = 16000, designation = 'ASST.PROF' WHERE empno = 102;

Output:

```
1 row updated
```

SELECT * FORM emp;

Output:

```
    EMPNO ENAME           DESIGNATION          SALARY
---------- --------------------  -------------------- ----------
       101 NAGARAJ            LECTURER               16000
       102 SARAVANAN          ASST.PROF              16000
       103 PANNERSELVAM       ASST.PROF              20000
       104 CHARULATHA         HOD, PROF              45000
```

Query 6:

DELETE emp WHERE empno=103;

Output:

```
1 row deleted
```

SELECT * FROM emp;

Output:

```
    EMPNO ENAME           DESIGNATION          SALARY
---------- --------------------  -------------------- ----------
       101 NAGARAJ            LECTURER               16000
       102 SARAVANAN          ASST.PROF              16000
       104 CHARULATHA         HOD, PROF              45000
```

Data Control Language (DCL) commands

Exercise 4:

Create Table

Query:

```
CREATE TABLE emp(
  empno NUMBER(10),
  ename VARCHAR(20),
  job VARCHAR(20),
  sal NUMBER(6),
  mgrno NUMBER(4),
  deptno NUMBER(3)
  );
```

Output:

Table is created

INSERTING Record

Query:

```
INSERT INTO emp2 VALUES(1001,'MAHESH','PROGRAMMER',15000,1560,200);
```

Output:

1 row created

```
INSERT INTO emp2 VALUES(1002,'MANOJ','TESTER',12000,1560,200);
```

Output:

1 row created

```
INSERT INTO emp2 VALUES(1003,'KARTHIK','PROGRAMMER',13000,1400,201);
```

Output:

1 row created

```
INSERT INTO emp2 VALUES(1004,'NARESH','CLERK',1400,1400,201);
```

Output:

1 row created

```
INSERT INTO emp2 VALUES(1005,'MANI','TESTER',13000,1400,200);
```

Output:

1 row created

SELECT * FROM emp2;

EMPNO	ENAME	JOB	SAL	MGRNO	DEPTNO
1001	MAHESH	PROGRAMMER	15000	1560	200
1002	MANOJ	TESTER	12000	1560	200
1003	KARTHIK	PROGRAMMER	13000	1400	201
1004	NARESH	CLERK	1400	1400	201
1005	MANI	TESTER	13000	1400	200

TABLE 2:

```
CREATE TABLE dept2 (
 Deptno NUMBER(3),
 Deptname VARCHAR(10),
 Location VARCHAR(15) );
```

Output:

Table is created

Insertion:

```
INSERT INTO dept2 VALUES(107,'DEVELOP','ADYAR');
```

Output:

1 row created

```
INSERT INTO dept2 VALUES(201,'DEBUG','UK');
```

Output:

1 row created

```
INSERT INTO dept2 VALUES(200,'TEST','US');
```

Output:

1 row created

```
INSERT INTO dept2 VALUES(201,'TEST','USSR');
```

Output:

1 row created

INSERT INTO dept2 VALUES(108,'DEBUG','ADYAR');

Output:

```
1 row created
```

INSERT INTO dept2 VALUES(109,'BUILD','POTHERI');

Output:

```
1 row created
```

SELECT * FROM dept2;

Output:

```
DEPTNO DEPTNAME   LOCATION
---------- ----------        ---------------
     107  DEVELOP       ADYAR
     201  DEBUG         UK
     200  TEST          US
     201  TEST          USSR
     108  DEBUG        ADYAR
     109  BUILD        POTHERI
```

NESTED SUBQUERY:

Query:

```
SELECT ename FROM emp2
WHERE sal > (SELECT MIN(sal) FROM emp2
WHERE deptno = (SELECT deptno FROM dept2 WHERE Location = 'UK'));
```

Output:

```
ENAME
--------------------
MAHESH
MANOJ
KARTHIK
MANI
```

Exercise 5: CURSOR

Create Table and Insertion:

```
CREATE TABLE EMP (
   EMPNO INT PRIMARY KEY,
   ENAME VARCHAR(50),
   JOB VARCHAR(50),
   MGR INT,
   HIREDATE DATE,
   SAL DECIMAL(10, 2),
   COMM DECIMAL(10, 2),
   DEPTNO INT
);
```

Output:

Table is created

```
INSERT INTO EMP VALUES(7639, 'SMITH', 'CLERK', 7902, TO_DATE('17-DEC-1980', 'DD-MON-YYYY'),
800, NULL, 20);
```

Output:

1 row created

```
INSERT INTO EMP VALUES (7499, 'ALLEN', 'SALESMAN', 7698, TO_DATE('20-FEB-1981', 'DD-MON-
YYYY'), 1600, 300, 30);
```

Output:

1 row created

```
INSERT INTO EMP VALUES (7521, 'WARD', 'SALESMAN', 7698, TO_DATE('22-FEB-1981', 'DD-MON-
YYYY'), 1250, 500, 30);
```

Output:

1 row created

```
INSERT INTO EMP VALUES (7566, 'JONES', 'MANAGER', 7839, TO_DATE('02-APR-1981', 'DD-MON-
YYYY'), 2975, NULL, 20);
```

Output:

1 row created

```
INSERT INTO EMP VALUES (7788, 'SCOTT', 'ANALYST', 7566, TO_DATE('09-DEC-1982', 'DD-MON-
YYYY'),3000, NULL, 20);
```

Output:

1 row created

SELECT * FROM emp;

Output:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7639	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7788	SCOTT	ANALYST	7566	09-DEC-82	3000		20

Implict Cursor:

Query:

```
DECLARE
   CURSOR emp_cursor IS
      SELECT ENAME, SAL FROM EMP WHERE EMPNO = &empno;
   ena EMP.ENAME%TYPE;
   esa EMP.SAL%TYPE;
BEGIN
   OPEN emp_cursor;
   FETCH emp_cursor INTO ena, esa;

   IF emp_cursor%NOTFOUND THEN
      DBMS_OUTPUT.PUT_LINE('Employee does not exist!');
   ELSE
      DBMS_OUTPUT.PUT_LINE('NAME: ' || ena);
      DBMS_OUTPUT.PUT_LINE('SALARY: ' || esa);
   END IF;

   CLOSE emp_cursor;
END;
/
```

Output:

```
Enter value for empno: 7566
NAME: JONES
SALARY: 2975
PL/SQL procedure successfully completed.
```

Explicit cursor:

```
DECLARE
   ena EMP.ENAME%TYPE;
   esa EMP.SAL%TYPE;
   CURSOR c1 IS
      SELECT ENAME, SAL FROM EMP;
BEGIN
   OPEN c1;

   FETCH c1 INTO ena, esa;
   IF c1%FOUND THEN
      DBMS_OUTPUT.PUT_LINE(ena || ' Salary is $' || esa);
   END IF;

   FETCH c1 INTO ena, esa;
   IF c1%FOUND THEN
      DBMS_OUTPUT.PUT_LINE(ena || ' Salary is $' || esa);
   END IF;

   FETCH c1 INTO ena, esa;
   IF c1%FOUND THEN
      DBMS_OUTPUT.PUT_LINE(ena || ' Salary is $' || esa);
   END IF;

   CLOSE c1;
END;
/
```

Output:

```
SMITH Salary is $800
ALLEN Salary is $1600
WARD Salary is $1250
PL/SQL procedure successfully completed.
```

Exercise 6: Trigger

Table Creation:

```
CREATE TABLE emp (
 id NUMBER(3),
 name VARCHAR(50),
 income NUMBER(4),
 expense NUMBER(3),
 savings NUMBER(3) );
```

```
INSERT INTO emp VALUES (2, 'Kumar', 2500, 150, 650);
```

Output:

```
1 row created
```

```
INSERT INTO emp VALUES (3, 'Venky', 5000, 900, 950);
```

Output:

```
1 row created
```

```
INSERT INTO emp VALUES (4, 'Anish', 9999, 999, 999);
```

Output:

```
1 row created
```

```
SELECT * FROM emp;
```

ID	NAME	INCOME	EXPENSE	SAVINGS
2	Kumar	2500	150	650
3	Venky	5000	900	950
4	Anish	9999	999	999

Program – Trigger After Update

Query:

```
CREATE OR REPLACE TRIGGER t_check
AFTER UPDATE OR INSERT OR DELETE
ON emp
FOR EACH ROW
BEGIN
  IF UPDATING THEN
    DBMS_OUTPUT.PUT_LINE('TABLE IS UPDATED');
  ELSIF INSERTING THEN
    DBMS_OUTPUT.PUT_LINE('TABLE IS INSERTED');
  ELSIF DELETING THEN
```

```
    DBMS_OUTPUT.PUT_LINE('TABLE IS DELETED');
  END IF;
END;
/
```

Output:

```
SET SERVEROUTPUT ON;

INSERT INTO emp (id, name, income, expense, savings) VALUES (5, 'John', 4500, 200, 500);

TABLE IS INSERTED
```

Program – Trigger Before Update

```
CREATE OR REPLACE TRIGGER emp1
BEFORE UPDATE OR INSERT OR DELETE
ON employee
FOR EACH ROW
BEGIN
  IF UPDATING THEN
    DBMS_OUTPUT.PUT_LINE('Table is updated');
  ELSIF INSERTING THEN
    DBMS_OUTPUT.PUT_LINE('Table is inserted');
  ELSIF DELETING THEN
    DBMS_OUTPUT.PUT_LINE('Table is deleted');
  END IF;
END;
/
```

Output:

```
SET SERVEROUTPUT ON;

INSERT INTO emp (id, name, income, expense, savings) VALUES (5, 'John', 4500, 200, 500);

TABLE IS INSERTED
```

Exercise 7 : Views

Table Creation:

```
CREATE TABLE EMPLOYEE (
 EMPLOYEE_NAME VARCHAR2(10),
 EMPLOYEE_NO NUMBER(8),
 DEPT_NAME VARCHAR2(10),
 DEPT_NO NUMBER(5),
 DATE_OF_JOIN DATE );
```

DESC employee;

Output:

```
Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPLOYEE_NAME                             VARCHAR2(10)
 EMPLOYEE_NO                               NUMBER(8)
 DEPT_NAME                                 VARCHAR2(10)
 DEPT_NO                                   NUMBER(5)
 DATE_OF_JOIN                              DATE
```

View Creation:

CREATE VIEW EMPVIEW AS SELECT EMPLOYEE_NAME, EMPLOYEE_NO, DEPT_NAME, DEPT_NO, DATE_OF_JOIN FROM EMPLOYEE;

DESC empview;

Output:

```
Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPLOYEE_NAME                             VARCHAR2(10)
 EMPLOYEE_NO                               NUMBER(8)
 DEPT_NAME                                 VARCHAR2(10)
 DEPT_NO                                   NUMBER(5)
 DATE_OF_JOIN                              DATE
```

Display Data from the View

SELECT * FROM EMPVIEW;

Output:

employee_name	employee_no	dept_name	dept_no
RAVI	124	ECE	89
VIJAY	345	CSE	21
RAJ	98	IT	22
GIRI	100	CSE	67

Insert Data into the View

INSERT INTO EMPVIEW VALUES ('SRI', 120, 'CSE', 67, '16-NOV-1981');

Output:

1 row created

Display View

SELECT * FROM EMPVIEW;

Output:

employee_name	employee_no	dept_name	dept_no
RAVI	124	ECE	89
VIJAY	345	CSE	21
RAJ	98	IT	22
GIRI	100	CSE	67
SRI	120	CSE	67

Display Table

SELECT * FROM employee;

employee_name	employee_no	dept_name	dept_no	Date_of_join
RAVI	124	ECE	89	15-JUN-05
VIJAY	345	CSE	21	21-JUN-06
RAJ	98	IT	22	30-SEP-06
GIRI	100	CSE	67	14-NOV-81
SRI	120	CSE	67	16-NOV-81

Delete a row in View :

DELETE FROM EMPVIEW WHERE EMPLOYEE_NAME = 'SRI';

employee_name	employee_no	dept_name	dept_no
RAVI	124	ECE	89
VIJAY	345	CSE	21
RAJ	98	IT	22
GIRI	100	CSE	67

Vo
Update a row in View:

UPDATE EMPVIEW SET EMPLOYEE_NAME = 'KAVI' WHERE EMPLOYEE_NAME = 'RAVI';

employee_name	employee_no	dept_name	dept_no
KAVI	124	ECE	89
VIJAY	345	CSE	21
RAJ	98	IT	22
GIRI	100	CSE	67

Drop a View

DROP VIEW EMPVIEW;

Output:

View Dropped

Exercise 8 : Procedures

Query:

```
CREATE OR REPLACE PROCEDURE award_bonus (
  emp_id IN NUMBER,
  bonus_rate IN NUMBER
)
AS
  emp_sal employees.salary%TYPE;
  emp_comm employees.commission_pct%TYPE;

  salary_missing EXCEPTION;

BEGIN
  SELECT salary, commission_pct
  INTO emp_sal, emp_comm
  FROM employees
  WHERE employee_id = emp_id;

  IF emp_sal IS NULL THEN
    RAISE salary_missing;
  ELSE
    IF emp_comm IS NULL THEN
      UPDATE employees
      SET salary = salary + salary * bonus_rate
      WHERE employee_id = emp_id;

      DBMS_OUTPUT.PUT_LINE('Employee ' || emp_id || ' receives a bonus: ' || TO_CHAR(emp_sal *
bonus_rate) );
    ELSE
      DBMS_OUTPUT.PUT_LINE('Employee ' || emp_id || ' receives a commission. No bonus allowed.');
    END IF;
  END IF;

EXCEPTION
  WHEN salary_missing THEN
    DBMS_OUTPUT.PUT_LINE('Employee ' || emp_id || ' does not have a value for salary. No
update.');
  WHEN OTHERS THEN
    NULL;
END award_bonus;
/
```

```
BEGIN
  award_bonus(123, 0.05);
  award_bonus(179, 0.05);
END;
/
```

Output:

```
Employee 123 received a bonus: 325
Employee 179 receives a commission. No bonus allowed.
```

Exercise 9: Functions

```
CREATE OR REPLACE FUNCTION last_first_name (empid NUMBER)
RETURN VARCHAR2 IS
   lastname employees.last_name%TYPE; -- variable declaration for last name
   firstname employees.first_name%TYPE; -- variable declaration for first name
BEGIN
   -- Fetch last_name and first_name for the provided empid
   SELECT last_name, first_name
   INTO lastname, firstname
   FROM employees
   WHERE employee_id = empid;

   -- Return the formatted employee name
   RETURN ( 'Employee: ' || empid || ' - ' || UPPER(lastname) || ', ' || UPPER(firstname) );

EXCEPTION
   WHEN no_data_found THEN
      RETURN 'Employee not found';
Employee: 123 - KUMAR, DEEPAK/


DECLARE
   empid NUMBER := 163;
BEGIN
   DBMS_OUTPUT.PUT_LINE( last_first_name(empid) );
END;
/
```

Output:

Employee: 123 - KUMAR, DEEPAK

PL/SQL Function to Search Address from a Phonebook:

Table Creation

```
CREATE TABLE phonebook (
   phone_no NUMBER(6) PRIMARY KEY,
   username VARCHAR2(30),
   doorno VARCHAR2(10),
   street VARCHAR2(30),
   place VARCHAR2(30),
   pincode CHAR(6)
);

INSERT INTO phonebook VALUES (20312, 'vijay', '120/5D', 'bharathi street', 'NGO colony', '629002');
INSERT INTO phonebook VALUES (29467, 'vasanth', '39D4', 'RK bhavan', 'sarakkal vilai', '629002');
```

SELECT * FROM phonebook;

Output:

PHONE_NO	USERNAME	DOORNO	STREET	PLACE	PINCODE
20312	vijay	120/5D	bharathi street	NGO colony	629002
29467	vasanth	39D4	RK bhavan	sarakkal vilai	629002

```
CREATE OR REPLACE FUNCTION findAddress (phone IN NUMBER)
RETURN VARCHAR2 AS
   address VARCHAR2(100);
BEGIN
   SELECT username || ',' || doorno || ',' || street || ',' || place || ',' || pincode
   INTO address
   FROM phonebook
   WHERE phone_no = phone;

   RETURN address;

EXCEPTION
   WHEN no_data_found THEN
      RETURN 'Address not found';
END findAddress;
/
```

Example 1:

```
DECLARE
   address VARCHAR2(100);
BEGIN
   address := findAddress(20312);
   DBMS_OUTPUT.PUT_LINE(address);
END;
/
```

Output:

```
Vijay,120/5D,bharathi street,NGO colony,629002
```

Example 2:

```
DECLARE
   address VARCHAR2(100);
BEGIN
   address := findAddress(23556);
   DBMS_OUTPUT.PUT_LINE(address);
END;
/
```

Output:

```
Address not found
```

Exercise 10: Embedded SQL