Lab 3

```
REM   Script: ADBMS_LAB3
REM   PRAC

DECLARE
   TYPE Employee_Record IS RECORD (
       emp_id   NUMBER,
       emp_name VARCHAR2(50),
       salary   NUMBER
   );
   emp Employee_Record;
BEGIN

   emp.emp_id := 1001;
   emp.emp_name := 'Arya Bhatt';
   emp.salary := 50000;
   DBMS_OUTPUT.PUT_LINE('Employee ID: ' || emp.emp_id);
   DBMS_OUTPUT.PUT_LINE('Employee Name: ' || emp.emp_name);
   DBMS_OUTPUT.PUT_LINE('Employee Salary: ' || emp.salary);
END;
/

DECLARE
   total_tables NUMBER;
BEGIN

   SELECT COUNT(*) INTO total_tables
   FROM all_tables;
   DBMS_OUTPUT.PUT_LINE('Total Number of Tables in the System: ' || total_tables);
END;
/

DECLARE
   total_user_tables NUMBER;
BEGIN

   SELECT COUNT(*) INTO total_user_tables
   FROM user_tables;
   DBMS_OUTPUT.PUT_LINE('Total Number of User-Created Tables: ' ||
total_user_tables);
END;
/

CREATE TABLE Student (
   st_id   NUMBER,
```

```
   st_name VARCHAR2(50),
   sub1    NUMBER,
   sub2    NUMBER,
   sub3    NUMBER,
   Total   NUMBER
);

DECLARE

   TYPE Student_Type IS TABLE OF Student%ROWTYPE;
   student_record Student%ROWTYPE;
   v_sub1 Student.sub1%TYPE;
   v_sub2 Student.sub2%TYPE;
   v_sub3 Student.sub3%TYPE;
BEGIN

   student_record.st_id := 101;
   student_record.st_name := 'ABC';
   student_record.sub1 := 85;
   student_record.sub2 := 90;
   student_record.sub3 := 88;
   student_record.Total := student_record.sub1 + student_record.sub2 +
student_record.sub3;
   DBMS_OUTPUT.PUT_LINE('Student ID: ' || student_record.st_id);
   DBMS_OUTPUT.PUT_LINE('Student Name: ' || student_record.st_name);
   DBMS_OUTPUT.PUT_LINE('Total Marks: ' || student_record.Total);
END;
/

CREATE TABLE Employee (
   emp_id    NUMBER,
   emp_name  VARCHAR2(50),
   TA        NUMBER,
   DA        NUMBER,
   HRA       NUMBER,
   Gross     NUMBER,
   bonus     NUMBER
);

DECLARE
    TYPE Employee_Type IS TABLE OF Employee%ROWTYPE;
   employee_record Employee%ROWTYPE;
   v_TA Employee.TA%TYPE;
   v_DA Employee.DA%TYPE;
   v_HRA Employee.HRA%TYPE;
```

```
BEGIN
   employee_record.emp_id := 1001;
   employee_record.emp_name := 'Arya Bhatt';
   employee_record.TA := 5000;
   employee_record.DA := 4000;
   employee_record.HRA := 3000;
   employee_record.Gross := employee_record.TA + employee_record.DA +
employee_record.HRA;
   employee_record.bonus := 0.10 * employee_record.Gross;
   DBMS_OUTPUT.PUT_LINE('Employee ID: ' || employee_record.emp_id);
   DBMS_OUTPUT.PUT_LINE('Employee Name: ' || employee_record.emp_name);
   DBMS_OUTPUT.PUT_LINE('Gross Salary: ' || employee_record.Gross);
   DBMS_OUTPUT.PUT_LINE('Bonus: ' || employee_record.bonus);
END;
/
```

Lab 2

```
REM   Script: ADBMS_LAB2
REM   PRAC

DECLARE
   counter NUMBER := 1;
BEGIN
   FOR counter IN 1..5 LOOP
       DECLARE
           square NUMBER;
       BEGIN
           square := counter * counter;
           DBMS_OUTPUT.PUT_LINE('Square of ' || counter || ' is ' || square);
           IF square > 10 THEN
               DBMS_OUTPUT.PUT_LINE('Square exceeds 10');
           END IF;
       END;
   END LOOP;
END;
/

DECLARE
   first_name VARCHAR2(50) := 'Arya';
   age NUMBER := 21;
   joining_date DATE := TO_DATE('2025-01-27', 'YYYY-MM-DD');
```

```
BEGIN
   DBMS_OUTPUT.PUT_LINE('First Name: ' || first_name);
   DBMS_OUTPUT.PUT_LINE('Age: ' || age);
   DBMS_OUTPUT.PUT_LINE('Joining Date: ' || joining_date);
END;
/

DECLARE
   counter NUMBER;
BEGIN
   FOR counter IN 1..10 LOOP
       DBMS_OUTPUT.PUT_LINE(counter * 5);
   END LOOP;
END;
/

DECLARE
   num NUMBER := 5;
   factorial NUMBER := 1;
BEGIN
   WHILE num > 0 LOOP
       factorial := factorial * num;
       num := num - 1;
   END LOOP;
   DBMS_OUTPUT.PUT_LINE('Factorial is: ' || factorial);
END;
/

DECLARE
   num NUMBER := 7;
BEGIN
   IF num > 0 THEN
       DBMS_OUTPUT.PUT_LINE('The number is positive');
       IF MOD(num, 2) = 0 THEN
           DBMS_OUTPUT.PUT_LINE('The number is even');
       ELSE
           DBMS_OUTPUT.PUT_LINE('The number is odd');
       END IF;
   ELSIF num < 0 THEN
       DBMS_OUTPUT.PUT_LINE('The number is negative');
   ELSE
       DBMS_OUTPUT.PUT_LINE('The number is zero');
   END IF;
END;
/
```

```
DECLARE
   first_name VARCHAR2(50) := 'Arya';
   last_name VARCHAR2(50) := 'Bhatt';
   full_name VARCHAR2(100);
BEGIN
   full_name := first_name || ' ' || last_name;
   DBMS_OUTPUT.PUT_LINE('Full Name: ' || full_name);
END;
/

DECLARE
   num NUMBER := 123;
   sum_of_digits NUMBER := 0;
BEGIN
   WHILE num > 0 LOOP
       sum_of_digits := sum_of_digits + MOD(num, 10);
       num := FLOOR(num / 10);
   END LOOP;
   DBMS_OUTPUT.PUT_LINE('Sum of digits: ' || sum_of_digits);
END;
/
```

Pl sql

```
DECLARE
   CURSOR emp_cur IS
   SELECT empno,ename,job
   FROM EMP
   WHERE job = 'SALESMAN';
   v_eid emp.empno%TYPE;
   v_ename emp.ename%TYPE;
   v_ejob emp.job%TYPE;

BEGIN
   OPEN emp_cur;
   LOOP
       FETCH emp_cur INTO v_eid, v_ename, v_ejob;
       EXIT WHEN emp_cur%NOTFOUND;
```

```
        DBMS_OUTPUT.PUT_LINE ('EMP_ID: ' ||v_eid|| ' EMP_NAME: '|| v_ename || '
EMP_JOB: '|| v_ejob);
   END LOOP;
   CLOSE emp_cur;
END;
/

DECLARE
   CURSOR emp_cur IS
   SELECT *
   FROM EMP
   WHERE deptno = '10';
   emp_dt emp_cur%ROWTYPE;

BEGIN
   OPEN emp_cur;
   LOOP
       FETCH emp_cur INTO emp_dt;
       EXIT WHEN emp_cur%NOTFOUND;
       DBMS_OUTPUT.PUT_LINE('EMP_ID: ' || emp_dt.empno || ' EMP_Name: ' ||
emp_dt.ename || ' EMP_JOB: ' || emp_dt.job || ' EMP_MGR: ' || emp_dt.mgr || '
EMP_HIREDATE: ' || emp_dt.hiredate || ' EMP_SAL: ' || emp_dt.sal || ' EMP_COMM: '
|| emp_dt.comm || ' EMP_DEPTNO: ' || emp_dt.deptno);

   END LOOP;
   CLOSE emp_cur;
END;
/

CREATE TABLE Employee (
   emp_id    NUMBER,
   emp_name  VARCHAR2(50),
   TA        NUMBER,
   DA        NUMBER,
   HRA       NUMBER,
   Gross     NUMBER,
   bonus     NUMBER
)


SELECT * FROM EMPLOYEE;

--DROP TABLE EMPLOYEE;
```

```sql
INSERT INTO Employee (emp_id, emp_name, TA, DA, HRA) VALUES (1001, 'Arya Bhatt',
5000, 4000, 3000);
INSERT INTO Employee (emp_id, emp_name, TA, DA, HRA) VALUES (1002, 'Ravi Kumar',
4500, 3500, 2800);
INSERT INTO Employee (emp_id, emp_name, TA, DA, HRA) VALUES (1003, 'Sita Sharma',
5500, 4200, 3300);
INSERT INTO Employee (emp_id, emp_name, TA, DA, HRA) VALUES (1004, 'Amit Verma',
6000, 4700, 3500);
INSERT INTO Employee (emp_id, emp_name, TA, DA, HRA) VALUES (1005, 'Nina Patel',
5200, 4000, 3100);



DECLARE
   CURSOR employee_cursor IS
       SELECT emp_id, emp_name, TA, DA, HRA
       FROM Employee;

   employee_record employee_cursor%ROWTYPE;
   v_Gross NUMBER;
   v_bonus NUMBER;
BEGIN

   OPEN employee_cursor;

   LOOP
       FETCH employee_cursor INTO employee_record;
       EXIT WHEN employee_cursor%NOTFOUND;

       v_Gross := employee_record.TA + employee_record.DA + employee_record.HRA;
       v_bonus := 0.10 * v_Gross;

       DBMS_OUTPUT.PUT_LINE('Employee ID: ' || employee_record.emp_id);
       DBMS_OUTPUT.PUT_LINE('Employee Name: ' || employee_record.emp_name);
       DBMS_OUTPUT.PUT_LINE('Gross Salary: ' || v_Gross);
       DBMS_OUTPUT.PUT_LINE('Bonus: ' || v_bonus);

       UPDATE Employee
       SET Gross = v_Gross,
           bonus = v_bonus
       WHERE emp_id = employee_record.emp_id;
   END LOOP;

   CLOSE employee_cursor;
```

```
   COMMIT;
END;



BEGIN
UPDATE Employee
   SET GROSS = 70000
   WHERE EMP_ID = 1006;

   IF SQL%NOTFOUND THEN
   INSERT INTO Employee (EMP_ID, GROSS) VALUES (1006,70000);
   END IF;
END;
/



BEGIN
UPDATE Employee
   SET GROSS = 70000
   WHERE EMP_ID = 1007;

   IF SQL%ROWCOUNT = 0 THEN
   INSERT INTO Employee (EMP_ID, GROSS) VALUES (1007,70000);
   END IF;
END;
/


SELECT * FROM EMPLOYEE;


-- CODE EXECUTES LATER FIRST EXCEPTION WILL BE EXECUTED FIRST

DECLARE
   emp_data Employee%ROWTYPE;

BEGIN
   SELECT *
   INTO emp_data
   FROM Employee
   WHERE emp_id = -1;

   DBMS_OUTPUT.PUT_LINE('Employee found: ' || emp_data.emp_id);
```

```
   EXCEPTION
       WHEN NO_DATA_FOUND THEN
           DBMS_OUTPUT.PUT_LINE('NO_DATA_FOUND raised!');
END;
/


CREATE TABLE Sailor(
   sid    NUMBER PRIMARY KEY,
   sname  VARCHAR2(50),
   rating NUMBER,
   age    NUMBER
);

INSERT INTO Sailor (sid, sname, rating, age) VALUES (1, 'John Doe', 3, 25);
INSERT INTO Sailor (sid, sname, rating, age) VALUES (2, 'Jane Smith', 5, 30);
INSERT INTO Sailor (sid, sname, rating, age) VALUES (3, 'Alice Brown', 4, 22);
INSERT INTO Sailor (sid, sname, rating, age) VALUES (4, 'Bob White', 2, 35);


CREATE TABLE Boat(
   bid    NUMBER PRIMARY KEY,
   name   VARCHAR2(50),
   color  VARCHAR2(50)
);

INSERT INTO Boat (bid, name, color) VALUES (1, 'Seagull', 'Blue');
INSERT INTO Boat (bid, name, color) VALUES (2, 'Wave Rider', 'Red');
INSERT INTO Boat (bid, name, color) VALUES (3, 'Shark Fin', 'Black');
INSERT INTO Boat (bid, name, color) VALUES (4, 'Sunset', 'Yellow');


CREATE TABLE Reservation(
   sid    NUMBER,
   bid    NUMBER,
   day    VARCHAR2(50),
   CONSTRAINT fk_sailor FOREIGN KEY (sid) REFERENCES Sailor(sid),
   CONSTRAINT fk_boat FOREIGN KEY (bid) REFERENCES Boat(bid)
);

INSERT INTO Reservation (sid, bid, day) VALUES (1, 1, '2025-01-28');
INSERT INTO Reservation (sid, bid, day) VALUES (2, 2, '2025-01-29');
INSERT INTO Reservation (sid, bid, day) VALUES (3, 3, '2025-01-30');
INSERT INTO Reservation (sid, bid, day) VALUES (4, 4, '2025-02-01');
```

```
DECLARE
   CURSOR red_boat_cursor IS
       SELECT s.sname, r.day, b.bid
       FROM Sailor s
       JOIN Reservation r ON s.sid = r.sid
       JOIN Boat b ON r.bid = b.bid
       WHERE b.color = 'Red';

   v_sname VARCHAR2(50);
   v_day VARCHAR2(50);
   v_bid NUMBER;
BEGIN

   OPEN red_boat_cursor;

   LOOP
       FETCH red_boat_cursor INTO v_sname, v_day, v_bid;

       EXIT WHEN red_boat_cursor%NOTFOUND;

       DBMS_OUTPUT.PUT_LINE('Sailor Name: ' || v_sname || ', Reservation Date: ' ||
v_day || ', Boat ID: ' || v_bid);
   END LOOP;

   CLOSE red_boat_cursor;
END;
/


SELECT s.sname AS sailor_name, r.day AS reservation_date, b.bid AS boat_id
FROM Sailor s
JOIN Reservation r ON s.sid = r.sid
JOIN Boat b ON r.bid = b.bid
WHERE b.color = 'Red';
```