DAY 14

```
PRN : 200243020003
```

Anonymous Block

1. Create and execute a simple anonymous block that outputs ?Hello World.?

```
BEGIN
    dbms_output.put_line ('hello world');
END;
```

2. Create and execute a simple anonymous block to display highest salary among-st employees

```
DECLARE
    SALARY NUMBER;
BEGIN
    SELECT
        max(SALARY) INTO SALARY
    FROM
        EMPLOYEES;
    dbms_output.put_line (SALARY);
END;
```

3. Create PL/SQL block to display details of the employee who earns highest salary

```
DECLARE
    FIRST_NAME VARCHAR2 (20);
    LAST_NAME VARCHAR2 (30);
    SALARY NUMBER;
BEGIN
    SELECT
        FIRST_NAME,
        LAST_NAME,
        SALARY INTO FIRST_NAME,
        LAST_NAME,
        SALARY
    FROM
        EMPLOYEES
    WHERE
        SALARY = (
            SELECT
```

```
                MAX(SALARY)
            FROM
                EMPLOYEES);
    dbms_output.put_line (FIRST_NAME || ' ' || LAST_NAME || ' ' ||
SALARY);
END;
```

4. Create PL/SQL block to display the details for user entered employee_id

```
DECLARE
    FIRST_NAME VARCHAR2 (30);
BEGIN
    SELECT
        FIRST_NAME INTO FIRST_NAME
    FROM
        EMPLOYEES
    WHERE
        EMPLOYEE_ID = :id;
    dbms_output.put_line (FIRST_NAME);
END;
```

5. Create PL/SQL block to display details nos. 1 to 10

```
DECLARE
    n number;
BEGIN
    n: = 1;
    WHILE n <= 10 LOOP
        DBMS_OUTPUT.PUT_LINE (' ' || n);
        n := n + 1;
    END LOOP;
END;
```

6. Create block to accept no. from user and check if it is Prime or Not and display message accordingly

```
DECLARE
    n NUMBER;
    i NUMBER: = 2;
    m NUMBER: = 0;
    flag NUMBER: = 0;
BEGIN
    n: = & n;
    m = n / 2;
    FOR i IN 2..m LOOP
        IF n % i == 0 THEN
            DBMS_OUTPUT.PUT_LINE ('Number is not Prime ');
```

```
            flag = 1;
            EXIT;
            i: = i + 1;
            IF flag == 0 THEN
                DBMS_OUTPUT.PUT_LINE ('Number is  Prime ');
                END;

END;
```

7. Create PL/SQL block to accept no. from user and check if it is armstrong no. or not

```
DECLARE
    n NUMBER: = 153;
    rem NUMBER;
    sum NUMBER: = 0;
    T NUMBER: = n;
BEGIN
    while n > 0 LOOP
        rem: = mod(n, 10);
        sum: = sum + power(rem, 3);
        n: = trunc(n / 10);
    END LOOP;
    IF T = SUM THEN
        DBMS_OUTPUT.PUT_LINE ('Number is  ARMSTRONG ');
    ELSE
        DBMS_OUTPUT.PUT_LINE ('Number is NOT ARMSTRONG ');
    END IF;
END;
```

8. Create PL/SQL block to accept radius and calculate and display area and circumference of circle

```
DECLARE
    r NUMBER: = 5;
    a number;
    c number;
BEGIN
    a: = 3.14 * r * r;
    c: = 2 * 3.14 * r;
    DBMS_OUTPUT.PUT_LINE ('area ' || a || ' Circumference ' || c);
END;
```

9. Create PL/SQL block to accept no. and check if it is even or odd

```
DECLARE
    n NUMBER: = 9;
BEGIN
```

```
    IF mod(n, 2) = 0 THEN
        DBMS_OUTPUT.PUT_LINE ('Even ');
    ELSE
        DBMS_OUTPUT.PUT_LINE ('ODD ');
    END IF;
END;
```

10. Create PL/SQL block to display details of the employee who do not have manager.

```
DECLARE
    ename varchar2 (30);
    emp_id number;
BEGIN
    SELECT
        last_name,
        employee_id INTO ename,
        emp_id
    FROM
        employees
    WHERE
        manager_id IS NULL;
    DBMS_OUTPUT.PUT_LINE ('Name: ' || ename);
    DBMS_OUTPUT.PUT_LINE ('Employee ID:  ' || emp_id);
END;
```

Cursors

1. Create PL/SQL block to display details of employees King

```
DECLARE
    eid number;
    ename varchar2 (20);
    CURSOR emp_king IS
    SELECT
        employee_id,
        last_name
    FROM
        employees
    WHERE
        last_name = 'King';
BEGIN
    OPEN emp_king;
    LOOP
        FETCH emp_king INTO eid,
        ename;
        EXIT
        WHEN emp_king % NOTFOUND;
        DBMS_OUTPUT.PUT_LINE ('ID :: ' || eid || ' name:: ' || ename);
```

```
    END LOOP;
    CLOSE emp_king;
END;
```

2. Create PL/SQL block to display all duplicate names from employees

```
DECLARE
    fname varchar2 (30);
    lname varchar2 (30);
    emp_id number;
    CURSOR emp_did IS
    SELECT
        first_name,
        last_name,
        employee_id
    FROM
        employees e1
    WHERE
        2 <= (
            SELECT
                count(last_name)
            FROM
                employees e2
            WHERE
                e1.last_name = e2.last_name);
BEGIN
    OPEN emp_did;
    DBMS_OUTPUT.PUT_LINE (rpad('First Name', 10, ' ') || rpad('  Last
Name', 11, ' ') || ' Employee ID');
    LOOP
        FETCH emp_did INTO fname,
        lname,
        emp_id;
        EXIT
        WHEN emp_did % NOTFOUND;
        DBMS_OUTPUT.PUT_LINE (rpad(fname, 10, ' ') || ' ' || rpad(lname,
10, ' ') || ' ' || emp_id);
    END LOOP;
    CLOSE emp_did;
END;
```

3. Create PL/SQL block to accept department_id from user and display detail of employees working under this department.

```
DECLARE
    ename varchar2 (30);
    emp_id number;
    dept_id employees.department_id % TYPE;
```

```
    CURSOR emp_did IS
    SELECT
        last_name,
        employee_id,
        department_id
    FROM
        employees
    WHERE
        department_id = & did;
BEGIN
    OPEN emp_did;
    LOOP
        FETCH emp_did INTO ename,
        emp_id,
        dept_id;
        EXIT
        WHEN emp_did % NOTFOUND;
        DBMS_OUTPUT.PUT_LINE ('  Name: ' || ename || '  Employee ID:  ' ||
emp_id || '   Department ID: ' || dept_id);
    END LOOP;
    CLOSE emp_did;
END;
```

4. Create PL/SQL block to display highest salary from each department

```
DECLARE
    emp_id number;
    sal number;
    dept_id employees.department_id % TYPE;
    CURSOR emp_did IS
    SELECT
        employee_id,
        department_id,
        salary
    FROM
        employees e1
    GROUP BY
        department_id,
        employee_id,
        salary
    HAVING
        salary IN(
            SELECT
                max(salary)
                FROM employees e2
            WHERE
                e1.department_id = e2.department_id);
BEGIN
    OPEN emp_did;
    DBMS_OUTPUT.PUT_LINE (rpad('Emp_id', 8, ' ') || ' ' || rpad('Dept_id',
8, ' ') || ' ' || 'Salary');
```

```
    LOOP
        FETCH emp_did INTO emp_id,
        dept_id,
        sal;
        EXIT
        WHEN emp_did % NOTFOUND;
        DBMS_OUTPUT.PUT_LINE (rpad(emp_id, 8, ' ') || ' ' || rpad(dept_id,
8, ' ') || ' ' || sal);
    END LOOP;
    CLOSE emp_did;
END;
```

5. Create PL/SQL block to display details of employees earning highest salary in their department

```
DECLARE
    ename varchar2 (30);
    emp_id number;
    sal number;
    dept_id employees.department_id % TYPE;
    CURSOR emp_did IS
    SELECT
        employee_id,
        last_name,
        department_id,
        salary
    FROM
        employees e1
    GROUP BY
        department_id,
        employee_id,
        salary,
        last_name
    HAVING
        salary IN(
            SELECT
                max(salary)
                FROM employees e2
            WHERE
                e1.department_id = e2.department_id);
BEGIN
    OPEN emp_did;
    DBMS_OUTPUT.PUT_LINE (rpad('Emp_id', 8, ' ') || ' ' || rpad('Name',
10, ' ') || ' ' || rpad('Dept_id', 8, ' ') || ' ' || 'Salary');
    LOOP
        FETCH emp_did INTO emp_id,
        ename,
        dept_id,
        sal;
        EXIT
        WHEN emp_did % NOTFOUND;
        DBMS_OUTPUT.PUT_LINE (rpad(emp_id, 8, ' ') || ' ' || rpad(ename,
```

```
10, ' ') || ' ' || rpad(dept_id, 8, ' ') || ' ' || sal);
    END LOOP;
    CLOSE emp_did;
END;
```

6. Create PL/SQL block to display all employees working in Toronto

```
DECLARE
    ename varchar2 (30);
    emp_id number;
    ct varchar2 (30);
    dept_id employees.department_id % TYPE;
    CURSOR emp_did IS
    SELECT
        employee_id,
        last_name,
        department_id,
        city
    FROM
        employees e1
        JOIN departments USING (department_id)
        JOIN locations USING (location_id)
    WHERE
        city = 'Toronto';
BEGIN
    OPEN emp_did;
    DBMS_OUTPUT.PUT_LINE (rpad('Emp_id', 8, ' ') || ' ' || rpad('Name',
10, ' ') || ' ' || rpad('Dept_id', 8, ' ') || ' ' || 'City');
    LOOP
        FETCH emp_did INTO emp_id,
        ename,
        dept_id,
        ct;
        EXIT
        WHEN emp_did % NOTFOUND;
        DBMS_OUTPUT.PUT_LINE (rpad(emp_id, 8, ' ') || ' ' || rpad(ename,
10, ' ') || ' ' || rpad(dept_id, 8, ' ') || ' ' || ct);
    END LOOP;
    CLOSE emp_did;
END;
```

7. Create PL/SQL block to display employees hired in month of Aug

```
DECLARE
    ename varchar2 (30);
    emp_id number;
    hd date;
    dept_id employees.department_id % TYPE;
```

```
    CURSOR emp_did IS
    SELECT
        employee_id,
        last_name,
        department_id,
        hire_date
    FROM
        employees e1
    WHERE
        hire_date LIKE '%AUG%';
BEGIN
    OPEN emp_did;
    DBMS_OUTPUT.PUT_LINE (rpad('Emp_id', 8, ' ') || ' ' || rpad('Name',
10, ' ') || ' ' || rpad('Dept_id', 8, ' ') || ' ' || 'Date');
    LOOP
        FETCH emp_did INTO emp_id,
        ename,
        dept_id,
        hd;
        EXIT
        WHEN emp_did % NOTFOUND;
        DBMS_OUTPUT.PUT_LINE (rpad(emp_id, 8, ' ') || ' ' || rpad(ename,
10, ' ') || ' ' || rpad(dept_id, 8, ' ') || ' ' || hd);
    END LOOP;
    CLOSE emp_did;
END;
```

8. Display eid, full name, salary, job_id, dept_name, city of employees for user entered city.

```
DECLARE
    ename varchar2 (30);
    emp_id number;
    job_id employees.job_id % TYPE;
    dep_nm varchar2 (30);
    ct varchar2 (30);
    CURSOR emp_did IS
    SELECT
        employee_id,
        last_name,
        job_id,
        department_name,
        city
    FROM
        employees e1
        JOIN departments USING (department_id)
        JOIN locations USING (location_id)
    WHERE
        city = initcap('&nm');
BEGIN
    OPEN emp_did;
    DBMS_OUTPUT.PUT_LINE (rpad('Emp_id', 8, ' ') || ' ' || rpad('Name',
```

```
10, ' ') || ' ' || rpad('Job_id', 8, ' ') || ' ' || rpad('Dept_Name', 10,
' ') || ' ' || 'City');
    LOOP
        FETCH emp_did INTO emp_id,
        ename,
        job_id,
        dep_nm,
        ct;
        EXIT
        WHEN emp_did % NOTFOUND;
        DBMS_OUTPUT.PUT_LINE (rpad(emp_id, 8, ' ') || ' ' || rpad(ename,
10, ' ') || ' ' || rpad(job_id, 8, ' ') || ' ' || rpad(dep_nm, 10, ' ') ||
' ' || ct);
    END LOOP;
    CLOSE emp_did;
END;
```

9. Create PL/SQL block to display employees who have changed their job atleast once

```
DECLARE
    ename varchar2 (30);
    emp_id number;
    jd employees.job_id % TYPE;
    CURSOR emp_did IS
    SELECT
        e1.employee_id,
        last_name,
        j.job_id
    FROM
        employees e1
        JOIN job_history j ON (e1.employee_id = j.employee_id)
    WHERE
        2 <= (
            SELECT
                count(job_id)
            FROM
                employees
            WHERE
                e1.employee_id = j.employee_id)
            AND e1.job_id != j.job_id;
BEGIN
    OPEN emp_did;
    DBMS_OUTPUT.PUT_LINE (rpad('Emp_id', 8, ' ') || ' ' || rpad('Name',
10, ' ') || ' ' || rpad('Job_id', 8, ' '));
    LOOP
        FETCH emp_did INTO emp_id,
        ename,
        jd;
        EXIT
        WHEN emp_did % NOTFOUND;
        DBMS_OUTPUT.PUT_LINE (rpad(emp_id, 8, ' ') || ' ' || rpad(ename,
```

```
10, ' ') || ' ' || rpad(jd, 8, ' '));
    END LOOP;
    CLOSE emp_did;
END;
```

10. Create PL/SQL block to details of employees earning salary in range 10000 to 20000

```
DECLARE
    ename varchar2 (30);
    emp_id number;
    sal employees.salary % TYPE;
    CURSOR emp_did IS
    SELECT
        e1.employee_id,
        last_name,
        salary
    FROM
        employees e1
    WHERE
        salary BETWEEN 10000 AND 20000;
BEGIN
    OPEN emp_did;
    DBMS_OUTPUT.PUT_LINE (rpad('Emp_id', 8, ' ') || ' ' || rpad('Name',
10, ' ') || ' ' || rpad('Salary', 8, ' '));
    LOOP
        FETCH emp_did INTO emp_id,
        ename,
        sal;
        EXIT
        WHEN emp_did % NOTFOUND;
        DBMS_OUTPUT.PUT_LINE (rpad(emp_id, 8, ' ') || ' ' || rpad(ename,
10, ' ') || ' ' || rpad(sal, 8, ' '));
    END LOOP;
    CLOSE emp_did;
END;
```