CURSORS (can be asked as explicit or implicit)

1. Write a cursor to find name, id and age of employees whose name starts with letter ‘P’.

```
DECLARE
  CURSOR emp_cursor IS
    SELECT emp_name, emp_id, age
    FROM employees
    WHERE emp_name LIKE 'P%';  -- Names starting with 'P'
  emp_row emp_cursor%ROWTYPE;
BEGIN
  OPEN emp_cursor;
  LOOP
    FETCH emp_cursor INTO emp_row;
    EXIT WHEN emp_cursor%NOTFOUND;  -- Exit when no more rows are found
    DBMS_OUTPUT.PUT_LINE('Name: ' || emp_row.emp_name || ', ID: ' ||
emp_row.emp_id || ', Age: ' || emp_row.age);
  END LOOP;
  CLOSE emp_cursor;
END;
```

```
DECLARE
  emp_name employees.emp_name%TYPE;
  emp_id employees.emp_id%TYPE;
  age employees.age%TYPE;
BEGIN
  -- Implicit cursor for selecting one employee whose name starts with 'P'
  SELECT emp_name, emp_id, age
  INTO emp_name, emp_id, age
  FROM employees
  WHERE emp_name LIKE 'P%' AND ROWNUM = 1;  -- Fetch one row only

  DBMS_OUTPUT.PUT_LINE('Name: ' || emp_name || ', ID: ' || emp_id || ', Age: ' || age);
END;
```

2. Write a cursor to find names of passengers who travel on RED BUS

```
DECLARE
  CURSOR red_bus_cursor IS
    SELECT passenger_name
    FROM passengers
    WHERE bus_name = 'RED BUS';
  pass_row red_bus_cursor%ROWTYPE;
BEGIN
  OPEN red_bus_cursor;
  LOOP
    FETCH red_bus_cursor INTO pass_row;
    EXIT WHEN red_bus_cursor%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE('Passenger: ' || pass_row.passenger_name);
  END LOOP;
  CLOSE red_bus_cursor;
```

```
END;


DECLARE
  passenger_name passengers.passenger_name%TYPE;
BEGIN
  -- Implicit cursor for selecting one passenger traveling on 'RED BUS'
  SELECT passenger_name INTO passenger_name
  FROM passengers
  WHERE bus_name = 'RED BUS' AND ROWNUM = 1;

  DBMS_OUTPUT.PUT_LINE('Passenger: ' || passenger_name);
END;
```

3. Write a cursor to display names of faculty who teach “Java Programming”.

```
DECLARE
  CURSOR java_faculty_cursor IS
    SELECT faculty_name
    FROM faculty
    WHERE subject = 'Java Programming';
  fac_row java_faculty_cursor%ROWTYPE;
BEGIN
  OPEN java_faculty_cursor;
  LOOP
    FETCH java_faculty_cursor INTO fac_row;
    EXIT WHEN java_faculty_cursor%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE('Faculty: ' || fac_row.faculty_name);
  END LOOP;
  CLOSE java_faculty_cursor;
END;


DECLARE
  faculty_name faculty.faculty_name%TYPE;
BEGIN
  -- Implicit cursor for selecting one faculty teaching 'Java Programming'
  SELECT faculty_name INTO faculty_name
  FROM faculty
  WHERE subject = 'Java Programming' AND ROWNUM = 1;

  DBMS_OUTPUT.PUT_LINE('Faculty: ' || faculty_name);
END;
```

4. Write a cursor to List the employees along with their Experience and Daily Salary.

```
DECLARE
  CURSOR emp_exp_salary_cursor IS
    SELECT emp_name, experience, salary/30 AS daily_salary
    FROM employees;
  emp_row emp_exp_salary_cursor%ROWTYPE;
BEGIN
```

```
  OPEN emp_exp_salary_cursor;
  LOOP
    FETCH emp_exp_salary_cursor INTO emp_row;
    EXIT WHEN emp_exp_salary_cursor%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE('Name: ' || emp_row.emp_name || ', Experience: ' ||
emp_row.experience || ', Daily Salary: ' || emp_row.daily_salary);
  END LOOP;
  CLOSE emp_exp_salary_cursor;
END;


DECLARE
  emp_name employees.emp_name%TYPE;
  experience employees.experience%TYPE;
  daily_salary employees.salary%TYPE;
BEGIN
  -- Implicit cursor for selecting one employee's experience and daily salary
  SELECT emp_name, experience, salary/30 INTO emp_name, experience, daily_salary
  FROM employees
  WHERE ROWNUM = 1;

  DBMS_OUTPUT.PUT_LINE('Name: ' || emp_name || ', Experience: ' || experience || ',
Daily Salary: ' || daily_salary);
END;
```

5. Write a cursor to list names of doctors whose salary is greater than doctor “Jhon”.

```
DECLARE
  CURSOR doctor_cursor IS
    SELECT doctor_name
    FROM doctors
    WHERE salary > (SELECT salary FROM doctors WHERE doctor_name = 'Jhon');
  doc_row doctor_cursor%ROWTYPE;
BEGIN
  OPEN doctor_cursor;
  LOOP
    FETCH doctor_cursor INTO doc_row;
    EXIT WHEN doctor_cursor%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE('Doctor: ' || doc_row.doctor_name);
  END LOOP;
  CLOSE doctor_cursor;
END;


DECLARE
  doctor_name doctors.doctor_name%TYPE;
BEGIN
  -- Implicit cursor for selecting one doctor whose salary is greater than Jhon's salary
  SELECT doctor_name INTO doctor_name
  FROM doctors
  WHERE salary > (SELECT salary FROM doctors WHERE doctor_name = 'Jhon') AND
ROWNUM = 1;
```

```
  DBMS_OUTPUT.PUT_LINE('Doctor: ' || doctor_name);
END;
```

Triggers

1. Write a trigger to check age validity of a customer using row level triggers.
   (Age should not be less than 20)

   ```
   CREATE OR REPLACE TRIGGER age_check_trigger

   BEFORE INSERT OR UPDATE ON customer

   FOR EACH ROW

   BEGIN

     IF :NEW.age < 20 THEN

       RAISE_APPLICATION_ERROR(-20001, 'Age must be at least 20.');

     END IF;

   END;
   ```

2. Create a Trigger for one instance of student table it will update another table while inserting values.

   ```
   CREATE OR REPLACE TRIGGER student_update_trigger

   AFTER INSERT ON student

   FOR EACH ROW

   BEGIN

     UPDATE other_table

     SET related_column = :NEW.student_column

     WHERE condition_column = :NEW.student_id;

   END;
   ```

3. Create a row level after trigger on customer table.

   ```
   CREATE OR REPLACE TRIGGER log_customer_insert

   AFTER INSERT ON customer

   FOR EACH ROW

   BEGIN
   ```