# Exercise 1 (2%)

Create an anonymous block that displays a course list. Declare a cursor and use the OPEN,FETCH, and

CLOSE cursor statements to access the cursor. Use the %ROWTYPE attribute for the cursor.

**Code:**

Drop TABLE CourseList;

CREATE TABLE CourseList (

CourseCode VARCHAR(4) PRIMARY KEY,

CourseTitle VARCHAR(100)

);

-- Altering the CourseCode column to increase its width

ALTER TABLE CourseList MODIFY (CourseCode VARCHAR(10));

INSERT INTO CourseList VALUES ('ACC103', 'Accounting Theory');

INSERT INTO CourseList VALUES ('ACC104', 'Microeconomics');

INSERT INTO CourseList VALUES ('ACC205', 'Financial Accounting');

INSERT INTO CourseList VALUES ('ANT100', 'Anthropology');

INSERT INTO CourseList VALUES ('BUS100', 'Introduction to Business');

INSERT INTO CourseList VALUES ('BUS230', 'Business Planning');

INSERT INTO CourseList VALUES ('CIS100', 'Web Technologies I');

INSERT INTO CourseList VALUES ('CIS105', 'Programming Logic');

INSERT INTO CourseList VALUES ('CIS200', 'Web Technologies II');

INSERT INTO CourseList VALUES ('CIS225', 'Python Programming');

INSERT INTO CourseList VALUES ('CIS300', 'Web Technologies III');

INSERT INTO CourseList VALUES ('CIS400', 'Database Design & SQL');

INSERT INTO CourseList VALUES ('ENG101', 'Communications I');

INSERT INTO CourseList VALUES ('ENG201', 'Communications II');

INSERT INTO CourseList VALUES ('GE0101', 'The Physical Environment');

INSERT INTO CourseList VALUES ('MGT410', 'Human Resources Management');

INSERT INTO CourseList VALUES ('MGT415', 'Project Management');

INSERT INTO CourseList VALUES ('MTH120', 'Algebra');

INSERT INTO CourseList VALUES ('MTH400', 'Geometry');

INSERT INTO CourseList VALUES ('NSG130', 'Nursing Theory I');

INSERT INTO CourseList VALUES ('NSG230', 'Nursing Theory II');

DECLARE

CURSOR Course\_cursor IS

SELECT CourseCode, CourseTitle

FROM CourseList

ORDER BY CourseCode;

Course\_Record Course\_cursor%ROWTYPE;

BEGIN

DBMS\_OUTPUT.PUT\_LINE(LPAD('CourseCode', 10) || LPAD('CourseTitle', 30));

DBMS\_OUTPUT.PUT\_LINE(LPAD('-', 13, '-') || ' ' || RPAD('-', 13, '-'));

OPEN Course\_cursor;

LOOP

FETCH Course\_cursor INTO Course\_Record; -- Fetch data into the record variable

EXIT WHEN Course\_cursor%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(LPAD(Course\_Record.CourseCode, 10) || ' ' || RPAD(Course\_Record.CourseTitle, 100));

END LOOP;

CLOSE Course\_cursor;

END;

# A computer screen shot of a blue screen Description automatically generated

# Exercise 2 (2%)

Write a PL/SQL block that declares and uses two cursors—one without a parameter and one with a parameter. The first cursor retrieves the department number and the department name from the departments table for all departments whose ID number is less than 100. The second cursor receives the department number as a parameter, and retrieves employee details for those who work in that department and whose employee\_id is less than 120.

**CODE:**

DROP TABLE departments;

DROP TABLE employee;

CREATE TABLE departments (

department\_id NUMBER PRIMARY KEY,

department\_name VARCHAR2(100)

);

-- Inserting values into the departments table with unique department names

INSERT INTO departments (department\_id, department\_name) VALUES (10, 'Finance');

INSERT INTO departments (department\_id, department\_name) VALUES (20, 'HR Management');

INSERT INTO departments (department\_id, department\_name) VALUES (30, 'Marketing and Sales');

INSERT INTO departments (department\_id, department\_name) VALUES (45, 'Operations');

INSERT INTO departments (department\_id, department\_name) VALUES (100, 'Customer Support');

INSERT INTO departments (department\_id, department\_name) VALUES (120, 'Product Development');

INSERT INTO departments (department\_id, department\_name) VALUES (110, 'Quality Assurance');

INSERT INTO departments (department\_id, department\_name) VALUES (130, 'Research and Development');

INSERT INTO departments (department\_id, department\_name) VALUES (15, 'Accounting');

INSERT INTO departments (department\_id, department\_name) VALUES (4, 'IT Services');

INSERT INTO departments (department\_id, department\_name) VALUES (5, 'Legal Affairs');

INSERT INTO departments (department\_id, department\_name) VALUES (6, 'Public Relations');

INSERT INTO departments (department\_id, department\_name) VALUES (65, 'Training and Development');

INSERT INTO departments (department\_id, department\_name) VALUES (45, 'Logistics');

INSERT INTO departments (department\_id, department\_name) VALUES (34, 'Procurement');

INSERT INTO departments (department\_id, department\_name) VALUES (98, 'Business Intelligence');

INSERT INTO departments (department\_id, department\_name) VALUES (32, 'Facilities Management');

INSERT INTO departments (department\_id, department\_name) VALUES (12, 'Strategic Planning');

INSERT INTO departments (department\_id, department\_name) VALUES (2, 'Corporate Communications');

CREATE TABLE employee(

employee\_id NUMBER PRIMARY KEY,

first\_name VARCHAR2(100),

last\_name VARCHAR2(100),

department\_id NUMBER,

CONSTRAINT fk\_department\_id FOREIGN KEY (department\_id)

REFERENCES departments (department\_id)

);

-- Inserting values into the employees table

INSERT INTO employee (employee\_id, first\_name, last\_name, department\_id)

VALUES (101, 'John', 'Doe', 10);

INSERT INTO employee (employee\_id, first\_name, last\_name, department\_id)

VALUES (102, 'Jane', 'Smith', 20);

INSERT INTO employee(employee\_id, first\_name, last\_name, department\_id)

VALUES (103, 'Michael', 'Johnson', 30);

INSERT INTO employee (employee\_id, first\_name, last\_name, department\_id)

VALUES (104, 'Emily', 'Brown', 45);

INSERT INTO employee (employee\_id, first\_name, last\_name, department\_id)

VALUES (105, 'David', 'Williams', 100);

INSERT INTO employee (employee\_id, first\_name, last\_name, department\_id)

VALUES (16, 'Jessica', 'Miller', 120);

INSERT INTO employee (employee\_id, first\_name, last\_name, department\_id)

VALUES (107, 'Daniel', 'Martinez', 110);

INSERT INTO employee(employee\_id, first\_name, last\_name, department\_id)

VALUES (108, 'Sarah', 'Taylor', 130);

INSERT INTO employee (employee\_id, first\_name, last\_name, department\_id)

VALUES (109, 'Ryan', 'Anderson', 15);

INSERT INTO employee (employee\_id, first\_name, last\_name, department\_id)

VALUES (110, 'Rachel', 'Garcia', 4);

INSERT INTO employee (employee\_id, first\_name, last\_name, department\_id)

VALUES (123, 'Andrew', 'Rodriguez', 5);

INSERT INTO employee (employee\_id, first\_name, last\_name, department\_id)

VALUES (112, 'Olivia', 'Hernandez', 6);

INSERT INTO employee(employee\_id, first\_name, last\_name, department\_id)

VALUES (113, 'Matthew', 'Lopez', 65);

INSERT INTO employee (employee\_id, first\_name, last\_name, department\_id)

VALUES (14, 'Elizabeth', 'Gonzalez', 45);

INSERT INTO employee (employee\_id, first\_name, last\_name, department\_id)

VALUES (129, 'Christopher', 'Perez', 34);

INSERT INTO employee (employee\_id, first\_name, last\_name, department\_id)

VALUES (136, 'Amanda', 'Wilson', 98);

INSERT INTO employee (employee\_id, first\_name, last\_name, department\_id)

VALUES (117, 'Justin', 'Moore', 32);

INSERT INTO employee (employee\_id, first\_name, last\_name, department\_id)

VALUES (128, 'Lauren', 'Jackson', 12);

INSERT INTO employee (employee\_id, first\_name, last\_name, department\_id)

VALUES (119, 'Kevin', 'White', 2);

DECLARE

-- Declare cursor to fetch departments with IDs less than 100

CURSOR Dept\_Cursor IS

SELECT department\_id, department\_name

FROM departments

WHERE department\_id < 100;

-- Declare cursor to fetch employees for a specific department

CURSOR Emp\_Cursor (dept\_id\_param IN departments.department\_id%TYPE) IS

SELECT employee\_id, first\_name, last\_name

FROM employee

WHERE department\_id = dept\_id\_param AND employee\_id < 120;

-- Declare variables to store department and employee details

dept\_id departments.department\_id%TYPE;

dept\_name departments.department\_name%TYPE;

emp\_id employee.employee\_id%TYPE;

emp\_first\_name employee.first\_name%TYPE;

emp\_last\_name employee.last\_name%TYPE;

BEGIN

-- Open department cursor

OPEN Dept\_Cursor;

LOOP

-- Fetch department details

FETCH Dept\_Cursor INTO dept\_id, dept\_name;

EXIT WHEN Dept\_Cursor%NOTFOUND;

-- Output department details

DBMS\_OUTPUT.PUT\_LINE('Department ID: ' || dept\_id || ', Department Name: ' || dept\_name);

-- Open employee cursor for the current department

OPEN Emp\_Cursor(dept\_id);

LOOP

-- Fetch employee details

FETCH Emp\_Cursor INTO emp\_id, emp\_first\_name, emp\_last\_name;

EXIT WHEN Emp\_Cursor%NOTFOUND;

-- Output employee details

DBMS\_OUTPUT.PUT\_LINE('Employee ID: ' || emp\_id || ', Name: ' || emp\_first\_name || ' ' || emp\_last\_name);

END LOOP;

CLOSE Emp\_Cursor; -- Close employee cursor

END LOOP;

CLOSE Dept\_Cursor; -- Close department cursor

END;

A screenshot of a computer

Description automatically generated