select \* from employees;

select \* from departments;

CREATE TABLE departments (

dep\_id NUMBER,

dep\_name VARCHAR(50),

dep\_location VARCHAR(50)

);

INSERT INTO departments (dep\_id, dep\_name, dep\_location) VALUES (1001, 'FINANCE', 'SYDNEY');

INSERT INTO departments (dep\_id, dep\_name, dep\_location) VALUES (2001, 'AUDIT', 'MELBOURNE');

INSERT INTO departments (dep\_id, dep\_name, dep\_location) VALUES (3001, 'MARKETING', 'PERTH');

INSERT INTO departments (dep\_id, dep\_name, dep\_location) VALUES (4001, 'PRODUCTION', 'BRISBANE');

-- 1 Display a cursor based detail information of employees from employees table

DECLARE

CURSOR emp\_cursor IS

SELECT \* FROM employees;

emp\_record employees%ROWTYPE;

BEGIN

OPEN emp\_cursor;

LOOP

FETCH emp\_cursor INTO emp\_record;

EXIT WHEN emp\_cursor%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('Employee ID: ' || emp\_record.emp\_id ||

', Name: ' || emp\_record.emp\_name ||

', Job: ' || emp\_record.job\_name ||

', Hire Date: ' || TO\_CHAR(emp\_record.hire\_date, 'YYYY-MM-DD') ||

', Salary: ' || emp\_record.salary ||

', Commission: ' || NVL(emp\_record.commission, 0) ||

', Department ID: ' || emp\_record.dep\_id);

END LOOP;

CLOSE emp\_cursor;

END;

/

-- 2 Display the list of managers and the name of the departments.

DECLARE

CURSOR mgr\_dept\_cursor IS

SELECT e.emp\_name AS manager\_name, d.dep\_name AS department\_name

FROM employees e

JOIN departments d ON e.dep\_id = d.dep\_id;

mgr\_dept\_record mgr\_dept\_cursor%ROWTYPE;

BEGIN

OPEN mgr\_dept\_cursor;

LOOP

FETCH mgr\_dept\_cursor INTO mgr\_dept\_record;

EXIT WHEN mgr\_dept\_cursor%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('Manager Name: ' || mgr\_dept\_record.manager\_name || ', Department: ' || mgr\_dept\_record.department\_name);

END LOOP;

CLOSE mgr\_dept\_cursor;

END;

/

-- 3 Write a PL/SQL block to help user to insert salary of an employee.

Display the name of the employee (from the employees table) to know that, the entered salary is present in the table 'employees'.

Display the appropriate messages by using exceptions such as 'no\_data\_found'.

DECLARE

v\_employee\_id employees.emp\_id%TYPE;

v\_new\_salary employees.salary%TYPE;

v\_emp\_name employees.emp\_name%TYPE;

no\_employee\_found EXCEPTION;

insufficient\_salary EXCEPTION;

BEGIN

-- Prompt user for employee ID and new salary

v\_employee\_id := 68319;

v\_new\_salary := 30000;

-- Check if the employee exists

SELECT emp\_name

INTO v\_emp\_name

FROM employees

WHERE emp\_id = v\_employee\_id;

-- Update the employee's salary

UPDATE employees

SET salary = v\_new\_salary

WHERE emp\_id = v\_employee\_id;

IF SQL%ROWCOUNT = 0 THEN

RAISE no\_employee\_found;

END IF;

-- Display confirmation message

DBMS\_OUTPUT.PUT\_LINE('Salary updated for employee: ' || v\_emp\_name);

-- Check if the salary is present in the table

DECLARE

CURSOR emp\_salary\_cursor IS

SELECT emp\_name

FROM employees

WHERE salary = v\_new\_salary;

emp\_record emp\_salary\_cursor%ROWTYPE;

BEGIN

OPEN emp\_salary\_cursor;

FETCH emp\_salary\_cursor INTO emp\_record;

IF emp\_salary\_cursor%FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Employee with salary ' || v\_new\_salary || ': ' || emp\_record.emp\_name);

ELSE

RAISE insufficient\_salary;

END IF;

CLOSE emp\_salary\_cursor;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('No employee found with salary ' || v\_new\_salary);

END;

EXCEPTION

WHEN no\_employee\_found THEN

DBMS\_OUTPUT.PUT\_LINE('No employee found with ID ' || v\_employee\_id);

WHEN insufficient\_salary THEN

DBMS\_OUTPUT.PUT\_LINE('No employee found with salary ' || v\_new\_salary);

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Error occurred: ' || SQLERRM);

END;

/

-- 4 Write a PL/SQL block that declares a user-defined exception, defines a cursor to fetch employee data, and checks if the employees have at least 5 years of experience from a specific reference date '01-01-2000'. If they do not, raise the exception and handle it appropriately.

DECLARE

CURSOR emp\_cursor IS

SELECT emp\_id, emp\_name, hire\_date

FROM employees;

emp\_record emp\_cursor%ROWTYPE;

insufficient\_experience EXCEPTION;

v\_reference\_date DATE := TO\_DATE('01-01-2000', 'DD-MM-YYYY');

v\_experience\_threshold NUMBER := 5 \* 12; -- 5 years in months

BEGIN

OPEN emp\_cursor;

LOOP

FETCH emp\_cursor INTO emp\_record;

EXIT WHEN emp\_cursor%NOTFOUND;

-- Calculate the months of experience

IF MONTHS\_BETWEEN(v\_reference\_date, emp\_record.hire\_date) < v\_experience\_threshold THEN

RAISE insufficient\_experience;

ELSE

DBMS\_OUTPUT.PUT\_LINE('Employee ' || emp\_record.emp\_name || ' has sufficient experience.');

END IF;

END LOOP;

CLOSE emp\_cursor;

EXCEPTION

WHEN insufficient\_experience THEN

DBMS\_OUTPUT.PUT\_LINE('Employee ' || emp\_record.emp\_name || ' has less than 5 years of experience.');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('An unexpected error occurred: ' || SQLERRM);

END;

/