EXPERIMENT 15

EXPERIMENT-15

Title: To understand the concepts of implicit and explicit cursor.

Objective: Students will be able to implement the concept of implicit and explicit cursor.

1. Using implicit cursor update the salary by an increase of 10% for all the records in EMPLOYEES table, and finally display how many records have been updated. If no records exist display the message "**No Change**".
2. Using explicit cursor fetch the employee name, employee_id and salary of all the records from EMPLOYEES table.
3. Using explicit cursor Insert the records from EMPLOYEES table for the columns employee_id, Last_Name and salary for those records whose salary exceeds 2500 into a new table TEMP_EMP

CREATING TABLE:

```sql
CREATE TABLE EMPLOYEES (
    employee_id SERIAL PRIMARY KEY,
    last_name VARCHAR(50),
    salary NUMERIC(10, 2)
);

INSERT INTO EMPLOYEES (last_name, salary) VALUES
('Smith', 2000),
('Johnson', 3000),
('Williams', 2500),
('Brown', 4000),
('Jones', 1500);

SELECT * FROM EMPLOYEES
```

	employee_id [PK] integer	last_name character varying (50)	salary numeric (10,2)
1	1	Smith	2000.00
2	2	Johnson	3000.00
3	3	Williams	2500.00
4	4	Brown	4000.00
5	5	Jones	1500.00

TASK1:

CODE:

```
DECLARE
    row_count INT;
BEGIN
    -- Update salaries by 10%
    UPDATE EMPLOYEES
    SET salary = salary * 1.10;

    -- Get the number of rows affected
    GET DIAGNOSTICS row_count = ROW_COUNT;

    -- Display the result
    IF row_count > 0 THEN
        RAISE NOTICE '% rows have been updated.', row_count;
    ELSE
        RAISE NOTICE 'No Change';
    END IF;
END $$;
```

RESULT:

	employee_id [PK] integer	last_name character varying (50)	salary numeric (10,2)
1	1	Smith	2200.00
2	2	Johnson	3300.00
3	3	Williams	2750.00
4	4	Brown	4400.00
5	5	Jones	1650.00

TASK2:

CODE:

```
DECLARE
    emp_cursor REFCURSOR;
    emp_id INT;
    emp_name VARCHAR(50);
    emp_salary NUMERIC(10, 2);
BEGIN
    -- Open the cursor
    OPEN emp_cursor FOR SELECT employee_id, last_name, salary FROM EMPLOYEES;

    -- Loop through each row in the cursor
    LOOP
        FETCH emp_cursor INTO emp_id, emp_name, emp_salary;
        EXIT WHEN NOT FOUND;

        -- Display each employee's details
        RAISE NOTICE 'Employee ID: %, Name: %, Salary: %', emp_id, emp_name, emp_salary;
    END LOOP;

    -- Close the cursor
    CLOSE emp_cursor;
```

RESULT:

```
NOTICE:  Employee ID: 1, Name: Smith, Salary: 2200.00
NOTICE:  Employee ID: 2, Name: Johnson, Salary: 3300.00
NOTICE:  Employee ID: 3, Name: Williams, Salary: 2750.00
NOTICE:  Employee ID: 4, Name: Brown, Salary: 4400.00
NOTICE:  Employee ID: 5, Name: Jones, Salary: 1650.00
```

TASK3:

CODE:

```
DECLARE
    emp_cursor REFCURSOR;
    emp_id INT;
    emp_name VARCHAR(50);
    emp_salary NUMERIC(10, 2);
BEGIN
    -- Open the cursor to select employees with salary > 2500
    OPEN emp_cursor FOR SELECT employee_id, last_name, salary FROM EMPLOYEES WHERE salary > 2500

    -- Loop through each row in the cursor
    LOOP
        FETCH emp_cursor INTO emp_id, emp_name, emp_salary;
        EXIT WHEN NOT FOUND;

        -- Insert into TEMP_EMP table
        BEGIN
            INSERT INTO TEMP_EMP (employee_id, last_name, salary)
            VALUES (emp_id, emp_name, emp_salary);
        EXCEPTION
            WHEN unique_violation THEN
                -- Ignore duplicate entries
                CONTINUE;
        END;
    END LOOP;

    -- Close the cursor
    CLOSE emp_cursor;
```

RESULT:

	employee_id [PK] integer	last_name character varying (50)	salary numeric (10,2)
1	2	Johnson	3300.00
2	3	Williams	2750.00
3	4	Brown	4400.00