

Assignment No. 7A

Assignment Title: Cursors: (All types: Implicit, Explicit, Cursor FOR Loop, Parameterized Cursor)

Cursor Code:

```
CREATE TABLE TE (roll NUMBER, div CHAR(1), name VARCHAR2(50), marks NUMBER);
```

```
INSERT INTO TE (roll, div, name, marks) VALUES (1, 'B', 'Shraddha', 95);
```

```
INSERT INTO TE (roll, div, name, marks) VALUES (2, 'B', 'Saish', 85);
```

```
INSERT INTO TE (roll, div, name, marks) VALUES (3, 'A', 'Yash', 90);
```

```
INSERT INTO TE (roll, div, name, marks) VALUES (1, 'A', 'Shreya', 88);
```

```
INSERT INTO TE (roll, div, name, marks) VALUES (4, 'A', 'Shraddha', 92);
```

```
SELECT * FROM TE;
```

```
DECLARE
```

```
    CURSOR division_cursor IS  
    SELECT div, name, marks  
    FROM TE  
    ORDER BY div, marks DESC;
```

```
    v_prev_div TE.div%TYPE := NULL;  
    v_topper_name TE.name%TYPE;  
    v_topper_marks TE.marks%TYPE;
```

```
    v_div TE.div%TYPE;  
    v_name TE.name%TYPE;  
    v_marks TE.marks%TYPE;
```

```
BEGIN
```

```
    OPEN division_cursor;
```

```
    LOOP
```

```
        FETCH division_cursor INTO v_div, v_name, v_marks;  
        EXIT WHEN division_cursor%NOTFOUND;
```

```
        IF v_prev_div IS NULL OR v_prev_div != v_div THEN  
            v_prev_div := v_div;  
            v_topper_name := v_name;  
            v_topper_marks := v_marks;
```

```

        DBMS_OUTPUT.PUT_LINE('Division ' || v_prev_div || ':');
    ELSE
        IF v_marks > v_topper_marks THEN
            v_topper_name := v_name;
            v_topper_marks := v_marks;
        END IF;
    END IF;

    DBMS_OUTPUT.PUT_LINE(v_name || ', Marks: ' || v_marks);
END LOOP;

-- Display the topper of the last division
IF v_prev_div IS NOT NULL THEN
    DBMS_OUTPUT.PUT_LINE('Topper of division ' || v_prev_div || ': ' || v_topper_name
|| ', Marks: ' || v_topper_marks);
END IF;

CLOSE division_cursor;
END;
/

```

Assignment 7B

```

CREATE TABLE N_RollCall(id NUMBER, name VARCHAR(30));
CREATE TABLE O_RollCall(id NUMBER, name VARCHAR(30));

```

```

INSERT INTO N_RollCall VALUES(1,'Saish');
INSERT INTO N_RollCall VALUES(2,'Shraddha');
INSERT INTO N_RollCall VALUES(3,'Yash');
INSERT INTO N_RollCall VALUES(4,'Bhumi');

```

```

INSERT INTO O_RollCall VALUES(1,'Saish');
INSERT INTO O_RollCall VALUES(5,'Anushka');
INSERT INTO O_RollCall VALUES(6,'Viraj');
INSERT INTO O_RollCall VALUES(7,'Siddhesh');

```

```

DECLARE
    CURSOR c_new_rollcall IS
        SELECT *
        FROM N_RollCall
        ORDER BY id; -- Assuming "id" is the roll number column

    id1 N_RollCall.id%TYPE;
    name1 N_RollCall.name%TYPE;

    CURSOR c_existing_rollcall IS

```

```

SELECT *
FROM O_RollCall
ORDER BY id; -- Assuming "id" is the roll number column

id1_existing O_RollCall.id%TYPE;
name1_existing O_RollCall.name%TYPE;

BEGIN
    OPEN c_new_rollcall;
    OPEN c_existing_rollcall;

    FETCH c_new_rollcall INTO id1, name1;
    FETCH c_existing_rollcall INTO id1_existing, name1_existing;

    WHILE c_new_rollcall%FOUND OR c_existing_rollcall%FOUND LOOP
        -- Check if we need to insert the new record
        IF NOT c_existing_rollcall%FOUND OR (c_new_rollcall%FOUND AND id1 <
id1_existing) THEN
            -- Insert the new record into O_RollCall
            INSERT INTO O_RollCall (id, name)
            VALUES (id1, name1);

            FETCH c_new_rollcall INTO id1, name1;
        ELSE
            -- Skip the existing record
            FETCH c_existing_rollcall INTO id1_existing, name1_existing;
        END IF;
    END LOOP;

    CLOSE c_new_rollcall;
    CLOSE c_existing_rollcall;

    COMMIT;
    DBMS_OUTPUT.PUT_LINE('Data merged and ordered successfully.');
```

EXCEPTION

```

    WHEN OTHERS THEN
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
END;
/
```

Output:

```
SELECT * FROM N_RollCall;
```

| Results | Explain | Describe | Saved SQL | History |
|---------|----------|----------|-----------|---------|
| | | | | |
| ID | NAME | | | |
| 3 | Yash | | | |
| 4 | Bhumi | | | |
| 1 | Saish | | | |
| 2 | Shraddha | | | |

4 rows returned in 0.00 seconds [CSV Export](#)

SELECT * FROM O_RollCall;

| Results | Explain | Describe | Saved SQL | History |
|---------|----------|----------|-----------|---------|
| | | | | |
| ID | NAME | | | |
| 1 | Saish | | | |
| 5 | Anushka | | | |
| 6 | Viraj | | | |
| 7 | Siddhesh | | | |

4 rows returned in 0.00 seconds [CSV Export](#)

Data merged and ordered successfully.

1 row(s) inserted.

SELECT DISTINCT id,name FROM O_RollCall ORDER BY id;

| Results | Explain | Describe | Saved SQL | History |
|---------|----------|----------|-----------|---------|
| | | | | |
| ID | NAME | | | |
| 1 | Saish | | | |
| 2 | Shraddha | | | |
| 3 | Yash | | | |
| 4 | Bhumi | | | |
| 5 | Anushka | | | |
| 6 | Viraj | | | |
| 7 | Siddhesh | | | |

7 rows returned in 0.01 seconds [CSV Export](#)

Output:

| Results | Explain | Describe | Saved SQL | History |
|---------|---------|----------|-----------|---------|
| ROLL | DIV | NAME | MARKS | |
| 1 | B | Shraddha | 95 | |
| 2 | B | Saish | 85 | |
| 3 | A | Yash | 90 | |
| 1 | A | Shreya | 88 | |
| 4 | A | Shraddha | 92 | |
| 5 | B | Ritu | 90 | |

6 rows returned in 0.00 seconds

[CSV Export](#)

| Results | Explain | Describe | Saved SQL | History |
|---------|---------|----------|-----------|---------|
|---------|---------|----------|-----------|---------|

Division A:

Shraddha, Marks: 92

Yash, Marks: 90

Shreya, Marks: 88

Division B:

Shraddha, Marks: 95

Ritu, Marks: 90

Saish, Marks: 85

Topper of division B: Shraddha, Marks: 95

Statement processed.

0.11 seconds