

FETCHING VALUES FROM CURSER

```
Fetching values from a cursor
DECLARE
CURSOR c_emp_cursor IS
| SELECT employee_id, last_name FROM employees
WHERE department_id = 30;
v_empno employees.employee_id%TYPE;
v_lname employees.last_name%TYPE;
BEGIN
OPEN c_emp_cursor;
LOOP
FETCH c_emp_cursor INTO v_empno, v_lname;
EXIT WHEN c_emp_cursor%NOTFOUND;
DBMS_OUTPUT.PUT_LINE (v_empno || ' ' || v_lname);
END LOOP
| CLOSE c_emp_cursor;
END;
```

```
EXIT WHEN c_emp_cursor%ROWCOUNT > 1
```

```
90 mahesh
```

```
Statement processed. 0.01 seconds
```

FETCHING VALUES FROM THE ROW:

```
Fetching values into a record

DECLARE
CURSOR c_emp_cursor IS
SELECT employee_id, last_name FROM employees
WHERE department_id = 30;
v_emp_record c_emp_cursor%ROWTYPE;
BEGIN
OPEN c_emp_cursor;
LOOP
FETCH c_emp_cursor INTO v_emp_record;
EXIT WHEN c_emp_cursor%NOTFOUND;
DBMS_OUTPUT.PUT_LINE (v_emp_record.employee_id || ' ' || v_emp_record.last_name);
END LOOP;
CLOSE c_emp_cursor;
| END;
```

```
EXIT WHEN c_emp_cursor%ROWCOUNT > 1
```

```
90 mahesh
```

```
Statement processed. 0.01 seconds
```

Using %ROWCOUNT and %NOTFOUND attributes:

```
DECLARE
CURSOR c_emp_cursor IS
SELECT MANAGERID, NAME FROM EMP
WHERE DEPARTMENTID = 90;
v_emp_record c_emp_cursor%ROWTYPE;
BEGIN
OPEN c_emp_cursor;
LOOP
FETCH c_emp_cursor INTO v_emp_record;
EXIT WHEN c_emp_cursor%ROWCOUNT > 10 OR c_emp_cursor%NOTFOUND;
DBMS_OUTPUT.PUT_LINE (v_emp_record.MANAGERID || ' ' || v_emp_record.NAME);
END LOOP;
CLOSE c_emp_cursor;
END;
```

```
EXIT WHEN c_emp_cursor%ROWCOUNT > 10 OR
89 tej
```

Using cursor for loop:

```
DECLARE
CURSOR c_emp_cursor IS
SELECT MANAGERID, NAME FROM EMP
WHERE DEPARTMENTID = 89;
BEGIN
FOR emp_record IN c_emp_cursor
LOOP
DBMS_OUTPUT.PUT_LINE (emp_record.MANAGERID || ' ' || emp_record.NAME);
END LOOP;
END;
```

```
DECLARE CURSOR c_emp_cursor IS SELECT  
emp_record.NAME); END LOOP; END;
```

89 tej

Statement processed. 0.00 seconds

ASCENDING OSER USING CURSER:

```
DECLARE  
CURSOR person_cursor IS  
SELECT * FROM EMP ORDER BY AGE ASC;  
person_record EMP%ROWTYPE;  
BEGIN  
OPEN person_cursor;  
  
LOOP  
FETCH person_cursor INTO person_record;  
EXIT WHEN person_cursor%NOTFOUND;  
  
DBMS_OUTPUT.PUT_LINE('ID: ' || person_record.DEPARTMENTID || ' , Name: ' || person_record.NAME || ' , MANAGER ID ' || person_record.MANAGERID || ' , Age: ' || person_record.AGE);  
END LOOP;  
  
CLOSE person_cursor;  
END;
```

```
person_record; EXIT WHEN person_cursor%NOTFOUND;  
Name: ' || person_record.NAME || ' , MANAGER ID ' || person_record.MANAGERID  
CLOSE person_cursor; END;
```

```
ID: 89 , Name: tej , MANAGER ID 89 , Age: 17  
ID: 90 , Name: mahesh , MANAGER ID 90 , Age: 18  
ID: 82 , Name: mike , MANAGER ID 8182 , Age: 19  
ID: 88 , Name: goutham , MANAGER ID 8888 , Age: 19  
ID: 5068 , Name: sai , MANAGER ID 17 , Age: 20  
ID: 9989 , Name: kumar , MANAGER ID 19 , Age: 25  
ID: 80 , Name: surya , MANAGER ID 80 , Age: 32  
ID: 80 , Name: rohit , MANAGER ID 70 , Age: 37
```

Goto statement:

```

1 DECLARE
2   a number(2) := 30;
3 BEGIN
4   <<loopstart>>
5   -- while loop execution
6   WHILE a < 50 LOOP
7     dbms_output.put_line ('value of a: ' || a);
8     a := a + 1;
9     IF a = 35 THEN
10      a:=a+1;
11     END IF;
12   END LOOP;
13 END;

```

```

value of a: 30
value of a: 31
value of a: 32
value of a: 33
value of a: 34
value of a: 36
value of a: 37
value of a: 38
value of a: 39
value of a: 40
value of a: 41
value of a: 42
value of a: 43
value of a: 44
value of a: 45
value of a: 46
value of a: 47
value of a: 48
value of a: 49

```

Statement processed. 0.00 s

Exception:

```

1  DECLARE
2  dividend    NUMBER := 10;
3  divisor     NUMBER := 0;
4  result      NUMBER;
5  BEGIN
6  BEGIN
7      result := dividend / divisor;
8      DBMS_OUTPUT.PUT_LINE('Result: ' || result);
9  EXCEPTION
10     WHEN ZERO_DIVIDE THEN
11         DBMS_OUTPUT.PUT_LINE('Error: Division by zero');
12 END;
13 END;

```

```

DECLARE dividend    NUMBER := 10; divisor     NUMBER := 0; result
DBMS_OUTPUT.PUT_LINE('Error: Division by zero'); END; END;

Error: Division by zero

Statement processed. 0.01 seconds

```

Handle the NO_DATA_FOUND exception when retrieving a row from a table and no matching record is found.

```

1  DECLARE
2      emp_name    VARCHAR2(100);
3      emp_id      NUMBER := 250;
4  BEGIN
5      BEGIN
6          SELECT NAME INTO emp_name FROM EMPO WHERE MANAGERID = emp_id;
7          DBMS_OUTPUT.PUT_LINE('Employee Name: ' || emp_name);
8      EXCEPTION
9          WHEN NO_DATA_FOUND THEN
10             DBMS_OUTPUT.PUT_LINE('Error: No matching record found');
11     END;
12 END;

```

```

DECLARE    emp_name    VARCHAR2(100);    emp_
emp_name);    EXCEPTION    WHEN NO_DAT

Error: No matching record found

Statement processed. 0.01 seconds

```

RETRIVING THE DATA:

```
1 DECLARE
2 v_sum_sal NUMBER(10,2);
3 v_deptno NUMBER NOT NULL := 89;
4 BEGIN
5 SELECT SUM(MARKS)
6 INTO v_sum_sal FROM STU
7 WHERE STUDENTID = v_deptno;
8 DBMS_OUTPUT.PUT_LINE('Dep #60 Salary Total: ' || v_sum_sal);
9 END;
```

```
DECLARE v_sum_sal NUMBER(10,2); v_d
v_sum_sal FROM STU WHERE STUDENTID
v_sum_sal); END;
```

Dep #60 Salary Total: 210

Statement processed. 0.00 seconds

Commit and rollback:

```
DECLARE
v_emp_id STU.STUDENTID%TYPE := 89; -- Example employee ID
some_condition BOOLEAN := TRUE;
BEGIN
    -- Start a transaction
    UPDATE STU
    SET MARKS = MARKS * 1.1
    WHERE STUDENTID = v_emp_id;

    -- Simulate some condition where we might want to rollback
    IF some_condition THEN
        ROLLBACK; -- Undo the changes if the condition is met
        DBMS_OUTPUT.PUT_LINE('Transaction rolled back.');
```

ELSE

```
        COMMIT; -- If the condition is not met, save the changes
        DBMS_OUTPUT.PUT_LINE('Transaction committed.');
```

END IF;

```
END;
```

```
DBMS_OUTPUT.PUT_LINE('Transaction committed.
Transaction rolled back.
1 row(s) updated. 0.00 seconds
```

ROLLBACK:

```












BEGIN
INSERT INTO EMPO VALUES (3,'MIKE' ,4,89);
ROLLBACK;
INSERT INTO EMPO VALUES (5,'TEJ' ,6,99);
COMMIT;
END;
```

EDIT	DEPARTMENTID	NAME	MANAGERID	AGE
	5	TEJ	6	99
	90	mahesh	90	18
	89	tej	89	17
	82	mike	8182	19
	88	goutham	8888	19
	9989	kumar	19	25
	5068	sai	17	20
	80	rohit	70	37
	80	surya	80	32

SAVEPOINT:

```

BEGIN
INSERT INTO EMPO VALUES(6,'OG',12,88);
SAVEPOINT BEFORE_U;
INSERT INTO EMPO VALUES (3,'MIKE' ,4,89);
ROLLBACK TO BEFORE_U;
INSERT INTO EMPO VALUES (5,'TEJ' ,6,99);
COMMIT;
END;
```

EDIT	DEPARTMENTID	NAME	MANAGERID	AGE
	5	TEJ	6	99
	6	OG	12	88
	5	TEJ	6	99
	90	maresh	90	18
	89	tej	89	17
	82	mike	8182	19
	88	goutham	8888	19
	9989	kumar	19	25
	5068	sai	17	20
	80	rohit	70	37
	80	surya	80	32