

Name: Vinayak Kumar Singh
Register No: 23MCA1030

Subject: DBMS Lab
Date: 21-09-2023

Exercise 6: PL SQL PROGRAMS

Cursors

```
CREATE TABLE employees (  
    employee_id NUMBER,  
    first_name VARCHAR2(50),  
    last_name VARCHAR2(50),  
    salary NUMBER  
);
```

```
INSERT INTO employees (employee_id, first_name, last_name,  
salary)
```

```
VALUES (1, 'Aman', 'Sharma', 50000);
```

```
INSERT INTO employees (employee_id, first_name, last_name,  
salary)
```

```
VALUES (2, 'Chirag', 'Setpal', 60000);
```

```
INSERT INTO employees (employee_id, first_name, last_name,  
salary)
```

```
VALUES (3, 'Umesh', 'Sahu', 30000);
```

```
INSERT INTO employees (employee_id, first_name, last_name,  
salary)
```

```
VALUES (4, 'Prakhar', 'Shukla', 70000);
```

```
INSERT INTO employees (employee_id, first_name, last_name,  
salary)
```

```
VALUES (5, 'Vinayak', 'Singh', 100000);
```

```
DECLARE
```

```
CURSOR emp_cursor IS
    SELECT employee_id, first_name, last_name, salary FROM
employees;
    emp_rec employees%ROWTYPE;
BEGIN
    OPEN emp_cursor;
    LOOP
        FETCH emp_cursor INTO emp_rec;
        EXIT WHEN emp_cursor%NOTFOUND;

        DBMS_OUTPUT.PUT_LINE('Employee ID: ' ||
emp_rec.employee_id);
        DBMS_OUTPUT.PUT_LINE('First Name: ' ||
emp_rec.first_name);
        DBMS_OUTPUT.PUT_LINE('Last Name: ' ||
emp_rec.last_name);
        DBMS_OUTPUT.PUT_LINE('Salary: ' || emp_rec.salary);
    END LOOP;
    CLOSE emp_cursor;
END;
/
```

Output:

SQL Worksheet

Clear

Find

Actions

Save

Run

```
1 CREATE TABLE employees (  
2     employee_id NUMBER,  
3     first_name VARCHAR2(50),  
4     last_name VARCHAR2(50),  
5     salary NUMBER  
6 );  
7  
8 INSERT INTO employees (employee_id, first_name, last_name, salary)  
9 VALUES (1, 'Aman', 'Sharma', 50000);  
10 INSERT INTO employees (employee_id, first_name, last_name, salary)  
11 VALUES (2, 'Chirag', 'Setpal', 60000);  
12 INSERT INTO employees (employee_id, first_name, last_name, salary)  
13 VALUES (3, 'Umesh', 'Sahu', 30000);  
14 INSERT INTO employees (employee_id, first_name, last_name, salary)  
15 VALUES (4, 'Prakhar', 'Shukla', 70000);  
16 INSERT INTO employees (employee_id, first_name, last_name, salary)  
17 VALUES (5, 'Vinayak', 'Singh', 100000);  
18  
19 DECLARE  
20     CURSOR emp_cursor IS  
  
21     SELECT employee_id, first_name, last_name, salary FROM employees;  
22     emp_rec employees%ROWTYPE;  
23 BEGIN  
24     OPEN emp_cursor;  
25     LOOP  
26         FETCH emp_cursor INTO emp_rec;  
27         EXIT WHEN emp_cursor%NOTFOUND;  
28  
29         DBMS_OUTPUT.PUT_LINE('Employee ID: ' || emp_rec.employee_id);  
30         DBMS_OUTPUT.PUT_LINE('First Name: ' || emp_rec.first_name);  
31         DBMS_OUTPUT.PUT_LINE('Last Name: ' || emp_rec.last_name);  
32         DBMS_OUTPUT.PUT_LINE('Salary: ' || emp_rec.salary);  
33     END LOOP;  
34     CLOSE emp_cursor;  
35 END;  
36 /
```

Table created.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

Statement processed.
Employee ID: 1
First Name: Aman
Last Name: Sharma
Salary: 50000
Employee ID: 2
First Name: Chirag
Last Name: Setpal
Salary: 60000
Employee ID: 3
First Name: Umesh
Last Name: Sahu
Salary: 30000
Employee ID: 4
First Name: Prakhar
Last Name: Shukla
Salary: 70000
Employee ID: 5
First Name: Vinayak
Last Name: Singh
Salary: 100000

Triggers

```
CREATE TABLE new_employees (  
    employee_id NUMBER,  
    first_name VARCHAR2(50),  
    last_name VARCHAR2(50),  
    salary NUMBER  
);
```

```
CREATE OR REPLACE TRIGGER after_new_employee_insert  
AFTER INSERT ON new_employees  
FOR EACH ROW  
BEGIN  
    IF :NEW.salary >= 60000 THEN  
        DBMS_OUTPUT.PUT_LINE('Employee ' || :NEW.first_name ||  
' ' || :NEW.last_name || ' is eligible for a bonus.');    END IF;  
END after_new_employee_insert;  
/
```

```
INSERT INTO new_employees (employee_id, first_name,  
last_name, salary)  
VALUES (1, 'Aman', 'Sharma', 50000);  
  
INSERT INTO new_employees (employee_id, first_name,  
last_name, salary)  
VALUES (2, 'Chirag', 'Setpal', 40000);
```

```
INSERT INTO new_employees (employee_id, first_name,  
last_name, salary)
```

```
VALUES (3, 'Umesh', 'Sahu', 30000);
```

```
INSERT INTO new_employees (employee_id, first_name,  
last_name, salary)
```

```
VALUES (4, 'Prakhar', 'Shukla', 60000);
```

```
INSERT INTO new_employees (employee_id, first_name,  
last_name, salary)
```

```
VALUES (5, 'Vinayak', 'Singh', 100000);
```

```
COMMIT;
```

Output:

SQL Worksheet

ClearFindActionsSaveRun

```
1 CREATE TABLE new_employees (  
2     employee_id NUMBER,  
3     first_name VARCHAR2(50),  
4     last_name VARCHAR2(50),  
5     salary NUMBER  
6 );  
7  
8 CREATE OR REPLACE TRIGGER after_new_employee_insert  
9 AFTER INSERT ON new_employees  
10 FOR EACH ROW  
11 BEGIN  
12     IF :NEW.salary >= 60000 THEN  
13         DBMS_OUTPUT.PUT_LINE('Employee ' || :NEW.first_name || ' ' || :NEW.last_name || ' is eligible for a bonus.');14     END IF;  
15 END after_new_employee_insert;  
16 /  
17  
18 INSERT INTO new_employees (employee_id, first_name, last_name, salary)  
19 VALUES (1, 'Aman', 'Sharma', 50000);  
20 INSERT INTO new_employees (employee_id, first_name, last_name, salary)  
21 VALUES (2, 'Chirag', 'Setpal', 40000);  
22 INSERT INTO new_employees (employee_id, first_name, last_name, salary)  
23 VALUES (3, 'Umesh', 'Sahu', 30000);  
24  
25 INSERT INTO new_employees (employee_id, first_name, last_name, salary)  
26 VALUES (4, 'Prakhar', 'Shukla', 60000);  
27 INSERT INTO new_employees (employee_id, first_name, last_name, salary)  
28 VALUES (5, 'Vinayak', 'Singh', 100000);  
29 COMMIT;
```

Table created.

Trigger created.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
Employee Prakhar Shukla is eligible for a bonus.

1 row(s) inserted.
Employee Vinayak Singh is eligible for a bonus.

Statement processed.

END