Ex.No.: 12 WORKING WITH CURSOR, PROCEDURES AND FUNCTIONS
Date:

AIM:
Create PL/SQL Blocks to perform the Item Transaction Operations using CURSOR, FUNCTION and PROCEDUERE.
ALGORITHM:
STEP-1: Start.
STEP-2: Create two tables Item Master and Item Trans.
itemmaster(itemid , itemname, stockonhand)
itemtrans(itemid ,itemname ,dateofpurchase ,quantity)
STEP-3: Create a PROCEDURE with id, name and quantity as parameters which make a call to the FUNCTION by passing id, name, dop, and quantity as parameters dop is set as sysdate.
STEP-4: Using FUNCTION fetch each record from the table Item Master using CURSOR inside a Loop statement,
If Item Master‘s ItemId is equal to the entered ID value then exit the loop otherwise fetch the next record.

```
loop
fetch master into masterrec
exit when master%notfound
if masterrec.itemid=id then
exit;
end if;
end loop;
```

STEP-5: If Itemmaster's itemid = id then,
Add the Itemmaster's stockonhand with the given quantity and update the ItemMaster table and insert the Item information into the ItemTrans table.
STEP-6: Else, if the inputed item is not present in the ItemMaster table then insert the

Ex.No.: 12

WORKING WITH CURSOR, PROCEDURES AND

FUNCTIONS Date:

new Item in both the tables.
STEP-7: Call the Procedure by passing the Item informations which calls the Function.
STEP-8: Exit.

Factorial of a Number Using Function

1. Program to calculate the factorial of a number using a PL/SQL function.
```sql
CREATE OR REPLACE FUNCTION factorial(n NUMBER) RETURN NUMBER IS
   result NUMBER := 1;
BEGIN
   FOR i IN 1..n LOOP
      result := result * i;
   END LOOP;
   RETURN result;
END factorial;
/

-- Testing the function
DECLARE
   num NUMBER := 5;
   fact NUMBER;
BEGIN
   fact := factorial(num);
   DBMS_OUTPUT.PUT_LINE('Factorial of ' || num || ' is ' || fact);
END;
/
```

Program 2: Retrieve Book Information in Library Using IN, INOUT, and OUT Parameters

PL/SQL procedure to retrieve book information using `IN`, `INOUT`, and `OUT` parameters.
```sql
CREATE OR REPLACE PROCEDURE get_book_info(
   p_book_id IN NUMBER,
   p_book_name OUT VARCHAR2,
   p_author INOUT VARCHAR2
) IS
BEGIN
   SELECT book_name, author INTO p_book_name, p_author
   FROM library_books
   WHERE book_id = p_book_id;

   DBMS_OUTPUT.PUT_LINE('Book Name: ' || p_book_name);
   DBMS_OUTPUT.PUT_LINE('Author: ' || p_author);
END get_book_info;
```

```
/

-- Testing the procedure
DECLARE
    v_book_id NUMBER := 101;
    v_book_name VARCHAR2(100);
    v_author VARCHAR2(100) := 'Unknown Author';
BEGIN
    get_book_info(v_book_id, v_book_name, v_author);
    DBMS_OUTPUT.PUT_LINE('Retrieved Book Info: ' || v_book_name || ' by ' || v_author);
END;
/
```

Program to Display Employee ID and Name Using Explicit Cursor

PL/SQL block to display the employee ID and employee name where department number is 11 using an explicit cursor.

```sql
DECLARE
    CURSOR cenl IS
        SELECT eid, ename FROM ssempp WHERE dno = 11;

    ecode ssempp.eid%TYPE;
    ename ssempp.ename%TYPE;
BEGIN
    OPEN cenl;
    LOOP
        FETCH cenl INTO ecode, ename;
        EXIT WHEN cenl%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('Employee ID: ' || ecode || ' and Employee Name: ' || ename);
    END LOOP;
    CLOSE cenl;
END;
/
```

Program to Update Salary Using Implicit Cursor

PL/SQL block to update the salary by 5000 where the job is 'Lecturer' and check updates using implicit cursors.

```sql
DECLARE
    county NUMBER;
BEGIN
    UPDATE ssempp SET sal = sal + 5000 WHERE job = 'lecturer';
    county := SQL%ROWCOUNT;

    IF county > 0 THEN
        DBMS_OUTPUT.PUT_LINE('The number of rows updated is ' || county);
    END IF;

    IF SQL%FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Employee record modification successful');
    ELSIF SQL%NOTFOUND THEN
        DBMS_OUTPUT.PUT_LINE('No Employee record found with job Lecturer');
    END IF;
END;
/
```