**Ex No: 07**

**Date:**

## FUNCTIONS AND PROCEDURES

**AIM:**

To create and execute PL/SQL functions and procedures to manage the payroll process, including calculating gross salaries, displaying payroll details, incrementing salaries, and generating payroll reports.

Table creation:

**1.CREATE TABLE employee** (

```
    emp_id NUMBER(5) PRIMARY KEY,
    emp_name VARCHAR2(50),
    emp_designation VARCHAR2(50),
    email VARCHAR2(100),
    contactno VARCHAR2(15),
    basic_salary NUMBER(10, 2),      --
    hra NUMBER(10, 2),
    allowances NUMBER(10, 2)
);
```

**Table created.**

**2.Insert Values for above created table**

INSERT INTO employee VALUES (**101, 'Siva', 'AM', 'sivak.24mca@kongu.edu', 9345292781, 40000.00, 10000, 5000**);
INSERT INTO employee VALUES (**102, 'Sivakumar', 'DBT', 'sivakumarp.24mca@kongu.edu', 8072363074, 45000.00, 12000, 7000**);
INSERT INTO employee VALUES (**103, 'Sachin', 'C', 'sachins.24mca@kongu.edu', 8754681258, 38000.00, 9000, 4000**);
INSERT INTO employee VALUES (**104, 'Nisanth', 'IOT', 'nisanthg.24mca@kongu.edu', 6380603146, 42000.00, 11000, 6000**);

COMMIT;

**3.select * from employee;**

| EMP_ID | EMP_NAME | EMP_DES | EMAIL | CONTACTNO |
|---|---|---|---|---|
| 101 | Siva | AM | sivak.24mca@kongu.edu | 9345292781 |
| 102 | Sivakumar | DBT | sivakumarp.24mca@kongu.edu | 8072363074 |
| 103 | Sachin | C | sachins.24mca@kongu.edu | 8754681258 |
| 104 | Nisanth | IOT | nisanthg.24mca@kongu.edu | 6380603146 |

| BASIC_SALARY | HRA | ALLOWANCES |
|---|---|---|
| 40000.00 | 10000 | 5000 |
| 45000.00 | 12000 | 7000 |
| 38000.00 | 9000 | 4000 |
| 42000.00 | 11000 | 6000 |

**4. Stored Function for Calculating Gross Salary**

| **Gross Salary = Basic Salary + HRA + Allowances** |
|---|

```
CREATE OR REPLACE FUNCTION calculate_gross_salary(emp_id IN NUMBER)
RETURN NUMBER
IS
   gross_salary NUMBER;
BEGIN
   SELECT basic_salary + hra + allowances
   INTO gross_salary
   FROM employee
   WHERE emp_id = emp_id;

   RETURN gross_salary;
END;
```

**Test the Function**

**SELECT calculate_gross_salary(101) AS gross_salary FROM DUAL;**

**O/P**

| GROSS_SALARY |
| --- |
| 55000.00 |

**5. Stored Procedure for Displaying Employee Payroll**

This procedure fetches and displays the payroll details for an employee based on their ID

```
CREATE OR REPLACE PROCEDURE display_payroll(emp_id IN NUMBER)
IS
   emp_name VARCHAR2(50);
   gross_salary NUMBER;
BEGIN
   -- Calculate Gross Salary
   gross_salary := calculate_gross_salary(emp_id);
   -- Fetch Employee Name
   SELECT emp_name INTO emp_name FROM employee WHERE emp_id = emp_id;
   -- Display Payroll Details
   DBMS_OUTPUT.PUT_LINE('Payroll Details:');
   DBMS_OUTPUT.PUT_LINE('Employee Name: ' || emp_name);
   DBMS_OUTPUT.PUT_LINE('Gross Salary: ' || gross_salary);
END;
```

**Test the Procedure**

```
BEGIN
   display_payroll(101);
END;
```

**O/P**

| **Payroll Details:** |
| --- |
| **Employee Name: Siva** |
| **Gross Salary: 55000** |

**6. Stored Procedure for Incrementing Salaries**

This procedure increments the basic salary of employees based on their designation.

```
CREATE OR REPLACE PROCEDURE increment_salary(designation IN VARCHAR2,
increment_amount IN NUMBER)
IS
BEGIN
   UPDATE employee
   SET basic_salary = basic_salary + increment_amount
   WHERE emp_designation = designation;

   DBMS_OUTPUT.PUT_LINE('Salary incremented for designation: ' || designation);
END;
```

**Test the Procedure**

```
BEGIN
   increment_salary('AM', 5000); -- Increment for Assistant Manager
END;
```

**Verify the Updated Salary**:

```
SELECT emp_id, emp_name, basic_salary FROM employee WHERE emp_designation =
'AM';
```

**O/P**

| **EMP_ID** | **EMP_NAME** | **EMP_SALARY** |
| --- | --- | --- |
| **101** | **Siva** | **45000.00** |

**7. Generate a Payroll Report**

This report displays all employees’ gross salaries.

```
SELECT emp_id, emp_name, calculate_gross_salary(emp_id) AS gross_salary
FROM employee;
```

**O/P**

| EMP_ID | EMP_NAME | GROSS_SALARY |
|---|---|---|
| 101 | Siva | 60000.00 |
| 102 | Sivakumar | 64000.00 |
| 103 | Sachin | 51000.00 |
| 104 | Nisanth | 59000.00 |

| | |
|---|---|
| **COE (30)** | |
| **RECORD (20)** | |
| **VIVA (10)** | |
| **TOTAL (60)** | |

**RESULT:**

PL/SQL functions and procedures were effectively implemented to manage the payroll process.

**Ex No: 08**

**Date:**

## PL/SQL: CURSOR OPERATIONS

**AIM:**

To perform implicit and explicit cursor operations on an employee table to iterate through and display employee details.

**1.Table Creation**

```
CREATE TABLE employee (
    emp_id NUMBER(5) PRIMARY KEY,
    emp_name VARCHAR2(50),
    emp_designation VARCHAR2(50),
    email VARCHAR2(100),
    contactno VARCHAR2(15),
    basic_salary NUMBER(10, 2),
    hra NUMBER(10, 2),
    allowances NUMBER(10, 2)
);
```

**Table created**

**2.Insert Values**

INSERT INTO employee VALUES (**101, 'Siva', 'AM', 'sivak.24mca@kongu.edu', '9345292781', 40000.00, 10000, 5000**);

INSERT INTO employee VALUES (**102, 'Sivakumar', 'DBT', 'sivakumarp.24mca@kongu.edu', '8072363074', 45000.00, 12000, 7000**);

INSERT INTO employee VALUES (**103, 'Sachin', 'C', 'sachins.24mca@kongu.edu', '8754681258', 38000.00, 9000, 4000**);

INSERT INTO employee VALUES (**104, 'Nisanth', 'IOT', 'nisanthg.24mca@kongu.edu', '6380603146', 42000.00, 11000, 6000**);

COMMIT;

**3.select * from employee;**

| EMP_ID | EMP_NAME | EMP_DES | EMAIL | CONTACTNO |
|---|---|---|---|---|
| 101 | Siva | AM | sivak.24mca@kongu.edu | 9345292781 |
| 102 | Sivakumar | DBT | sivakumarp.24mca@kongu.edu | 8072363074 |
| 103 | Sachin | C | sachins.24mca@kongu.edu | 8754681258 |
| 104 | Siva | IOT | nisanthg.24mca@kongu.edu | 6380603146 |

| BASIC_SALARY | HRA | ALLOWANCES |
|---|---|---|
| 40000.00 | 10000 | 5000 |
| 45000.00 | 12000 | 7000 |
| 38000.00 | 9000 | 4000 |
| 42000.00 | 11000 | 6000 |

**4. Implicit Cursor**

To calculate and display the total number of employees in the table.

```
DECLARE
   total_employees NUMBER;
BEGIN
   -- Implicit Cursor Usage
   SELECT COUNT(*) INTO total_employees FROM employee;
   DBMS_OUTPUT.PUT_LINE('Total Number of Employees: ' || total_employees);
END;
```

**O/P**

| Total Number of Employees: **4** |
|---|

**5. Explicit Cursor**

To iterate through all employees and display their details.

```
DECLARE
   -- Declare a cursor
   CURSOR employee_cursor IS
      SELECT emp_id, emp_name, emp_designation, basic_salary FROM employee;

   -- Variables to hold fetched data
   v_emp_id         employee.emp_id%TYPE;
   v_emp_name       employee.emp_name%TYPE;
   v_emp_designation employee.emp_designation%TYPE;
   v_basic_salary   employee.basic_salary%TYPE;
BEGIN
   -- Open the cursor
   OPEN employee_cursor;
```

```
    DBMS_OUTPUT.PUT_LINE('Employee Details:');
    DBMS_OUTPUT.PUT_LINE('----------------------------');
    -- Fetch each row
    LOOP
        FETCH employee_cursor INTO v_emp_id, v_emp_name, v_emp_designation,
v_basic_salary
        -- Exit loop when no more rows
        EXIT WHEN employee_cursor%NOTFOUND;
        -- Display employee details
        DBMS_OUTPUT.PUT_LINE('Emp_ID: ' || v_emp_id || ', Name: ' || v_emp_name ||
                             ', Designation: ' || v_emp_designation ||
                             ', Basic Salary: ' || v_basic_salary);
    END LOOP;

    -- Close the cursor
    CLOSE employee_cursor;
END;
```

**O/P**

**Employee Details:**

---

Emp_ID: 101, Name: Nisanth, Designation: AM, Basic Salary: 40000

Emp_ID: 102, Name: Sivakumar, Designation: DBT, Basic Salary: 45000

Emp_ID: 103, Name: Sachin, Designation: C, Basic Salary: 38000

Emp_ID: 104, Name: Siva, Designation: IOT, Basic Salary: 42000

**6. Explicit Cursor with Parameters**

To filter and display employees based on their designation.

```
DECLARE
    -- Declare a parameterized cursor
    CURSOREmployee_designation_cursor(p_designation
employee.emp_designation%TYPE) IS
```

```
    SELECT emp_id, emp_name, basic_salary FROM employee WHERE emp_designation
= p_designation;

  -- Variables to hold fetched data
  v_emp_id       employee.emp_id%TYPE;
  v_emp_name     employee.emp_name%TYPE;
  v_basic_salary employee.basic_salary%TYPE;
BEGIN
  -- Open the cursor for a specific designation
  OPEN employee_designation_cursor('AM');

  DBMS_OUTPUT.PUT_LINE('Employees with Designation: AM');
  DBMS_OUTPUT.PUT_LINE('--------------------------------');

  -- Fetch each row
  LOOP
    FETCH employee_designation_cursor INTO v_emp_id, v_emp_name, v_basic_salary;

    -- Exit loop when no more rows
    EXIT WHEN employee_designation_cursor%NOTFOUND;

    -- Display employee details
    DBMS_OUTPUT.PUT_LINE('Emp_ID: ' || v_emp_id || ', Name: ' || v_emp_name ||
                         ', Basic Salary: ' || v_basic_salary);
  END LOOP;

  -- Close the cursor
  CLOSE employee_designation_cursor;
END;
```

**O/P**

**Employees with Designation: AM**

**--------------------------------**

Emp_ID: **101**, Name: **Siva,** Basic Salary: **40000**

| | |
|---|---|
| **COE (30)** | |
| **RECORD (20)** | |
| **VIVA (10)** | |
| **TOTAL (60)** | |

**Result**

The PL/SQL programs to perform implicit and explicit cursor operations on the employee table were executed successfully, and the employee details were displayed as expected.