**PRACTICAL NO 7**

**AIM: To study of Cursor and Trigger.**

**Theory:**

**CURSOR:** A cursor is a pointer to this context area. PL/SQL controls the context area through a cursor. A cursor holds the rows (one or more) returned by a SQL statement. The set of rows the cursor holds is referred to as the active set. You can name a cursor so that it could be referred to in a program to fetch and process the rows returned by the SQL statement, one at a time. There are two types of cursors Implicit cursors and Explicit cursors

Implicit cursors are automatically created by Oracle whenever an SQL statement is executed, when there is no explicit cursor for the statement. Programmers cannot control the implicit cursors and the information in it. It has following attributes:

* %FOUND
* %NOTFOUND
* %ISOPEN
* %ROWCOUNT

Explicit cursors are programmer-defined cursors for gaining more control over the context area. An explicit cursor should be defined in the declaration section of the PL/SQL Block. It is created on a SELECT Statement which returns more than one row.

The syntax for creating an explicit cursor is

**CURSOR cursor\_name IS select\_statement;**

**Declaring the Cursor**

Declaring the cursor defines the cursor with a name and the associated SELECT statement. For example:

**CURSOR c\_customers IS**

**SELECT id, name, address FROM customers;**

**Opening the Cursor**

Opening the cursor allocates the memory for the cursor and makes it ready for fetching the rows returned by the SQL statement into it. For example, we will open the above defined cursor as follows: **OPEN c\_customers;**

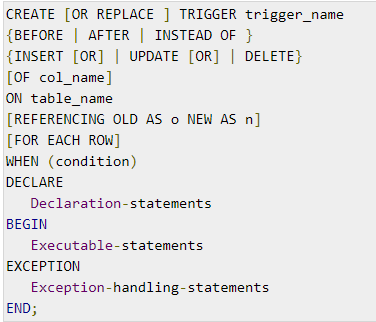
**Fetching the Cursor**

Fetching the cursor involves accessing one row at a time. For example, we will fetch rows from the above-opened cursor as follows: **FETCH c\_customers INTO c\_id, c\_name, c\_addr;**

**Closing the Cursor**

Closing the cursor means releasing the allocated memory. For example, we will close the above-opened cursor as follows: **CLOSE c\_customers;**

**TRIGGER:** Triggers can be defined on the table, view, schema, or database with which the event is associated. The syntax for creating a trigger is:

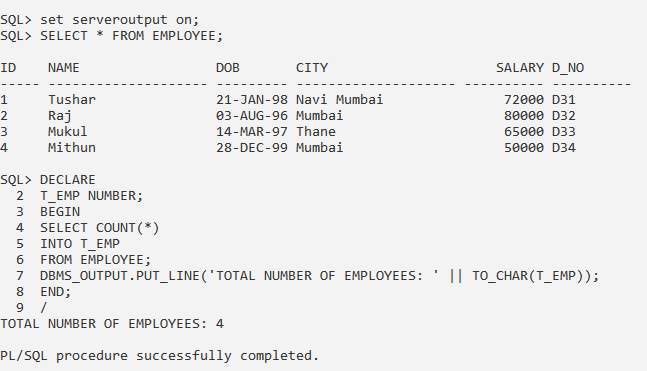


Where,

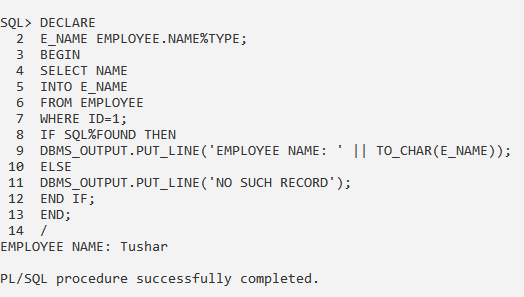
* CREATE [OR REPLACE] TRIGGER trigger\_name: Creates or replaces an existing trigger with the *trigger\_name*.
* {BEFORE | AFTER | INSTEAD OF}: This specifies when the trigger will be executed. The INSTEAD OF clause is used for creating trigger on a view.
* {INSERT [OR] | UPDATE [OR] | DELETE}: This specifies the DML operation.
* [OF col\_name]: This specifies the column name that will be updated.
* [ON table\_name]: This specifies the name of the table associated with the trigger.
* [REFERENCING OLD AS o NEW AS n]: This allows you to refer new and old values for various DML statements, such as INSERT, UPDATE, and DELETE.
* [FOR EACH ROW]: This specifies a row-level trigger, i.e., the trigger will be executed for each row being affected. Otherwise the trigger will execute just once when the SQL statement is executed, which is called a table level trigger.
* WHEN (condition): This provides a condition for rows for which the trigger would fire. This clause is valid only for row-level triggers.

**A) Dealing with table records**

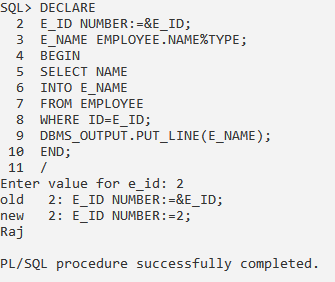
**1. Write a PL/SQL block to display total number of employees from Emp table.**

****

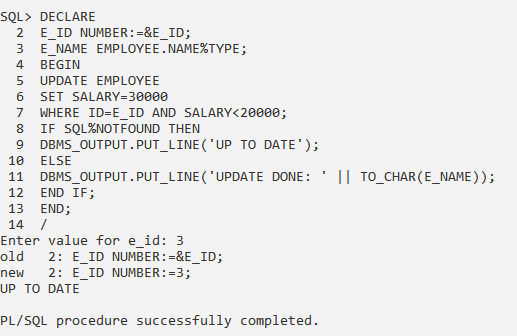
**2. Write a PL/SQL block to display employees Name whose Id is 1 from Emp table and if no record is found display message stating 'No such record.'.**

****

**3. Write a PL/SQL block to display employees Name from Emp table whose Id is accepted from user.**

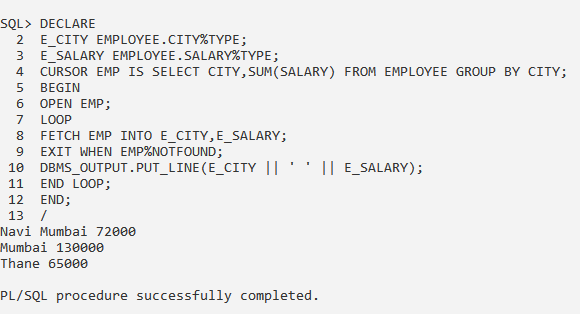
****

**4. Write a PL/SQL block to display employees Name and Basic from Emp table whose Id is accepted from user and if Basic is less than 20000 then update it to 30000 else display appropriate message.**

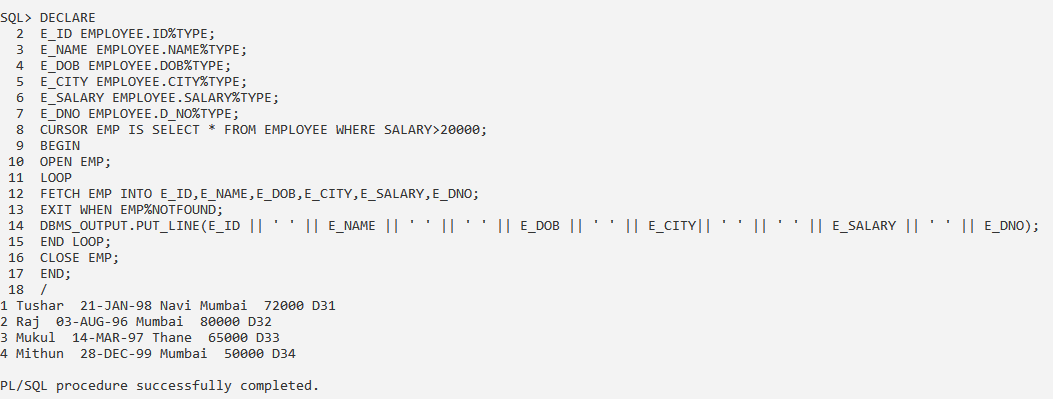
****

**B) Working with Cursor**

**1. Write a PL/SQL block containing cursor to display sum of salary city wise from “Employee” table.**

****

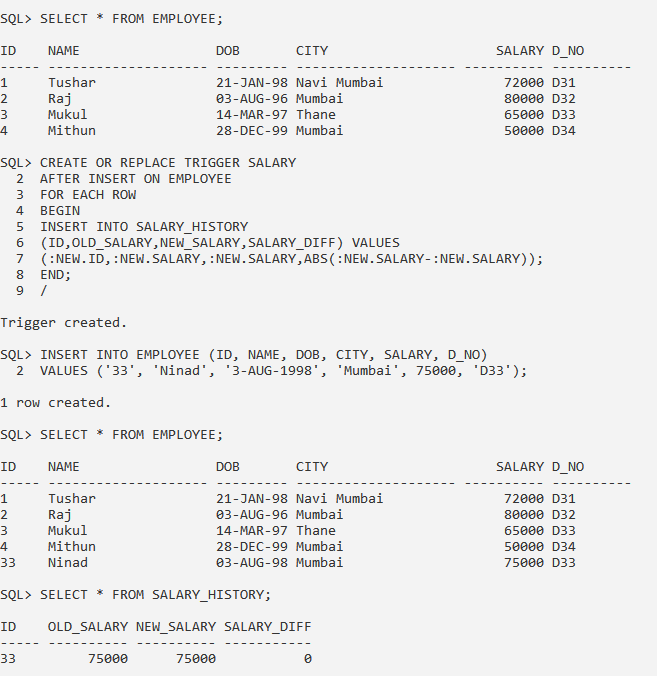
**2. Write a PL/SQL block containing cursor to fetch employee details from “Employee” table whose salary is greater than 25000.**

****

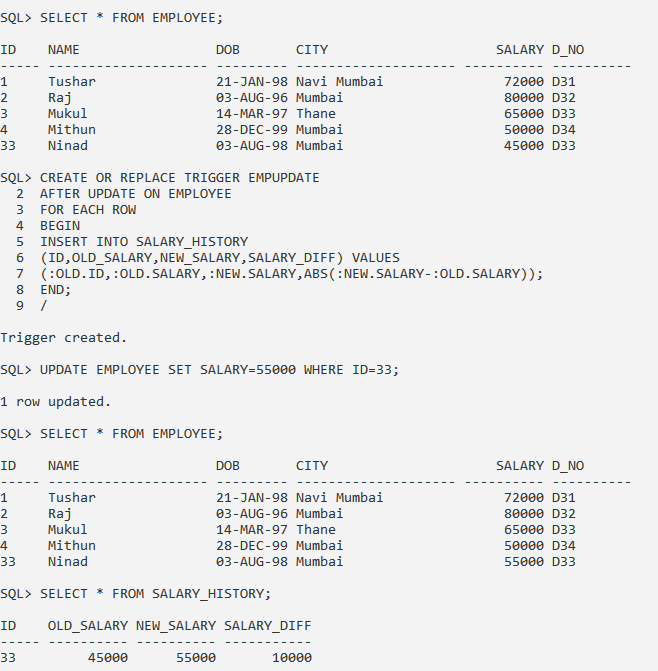
**C) Working with triggers**

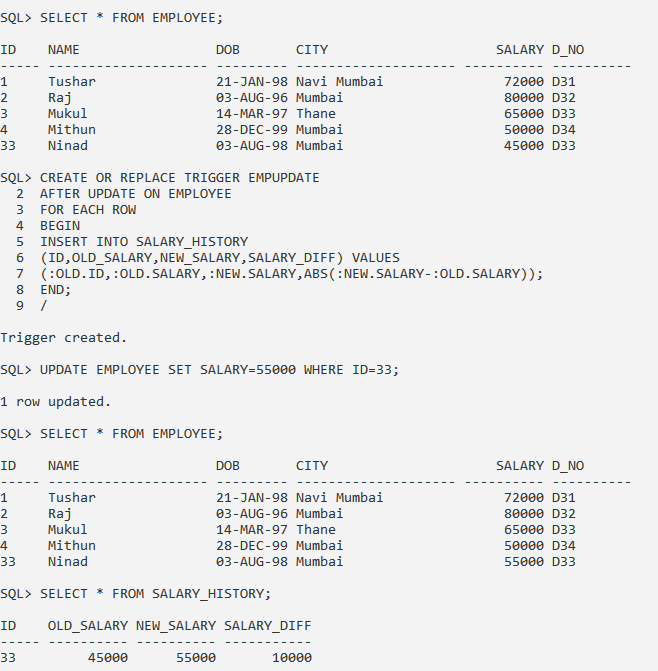
**1. Write a PL/SQL block containing trigger to display old salary, new salary and their difference from “Employee” table whenever an updating or insertion is performed on that table.**

**INSERTING:**

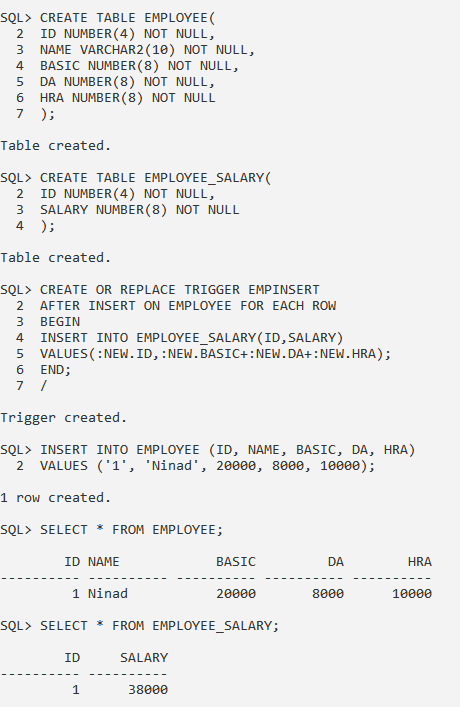
****

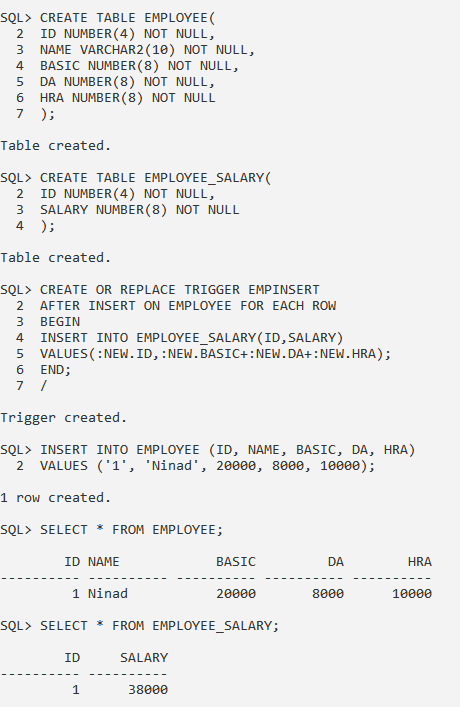
**UPDATING:**

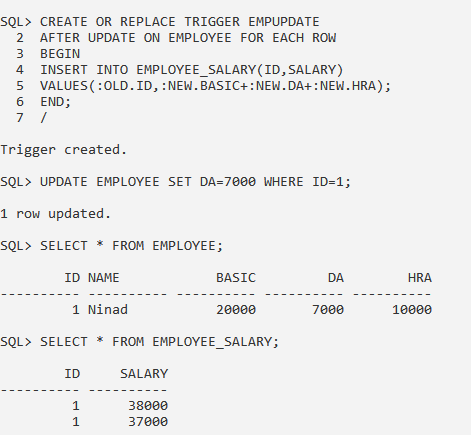
****

****

**2. Write a PL/SQL block containing trigger to calculate and display gross salary from “Emp” table whenever an updating or insertion is performed on that table.**

****

****

****

**CONCLUSION:** We have studied the Cursor and Trigger in details.