EXP. NO:- 13

Date:- 18/3/22.

PL/SQL Exception Handling

Aim :- To write a PL/SQL block to handle all types of exceptions.

PL/SQL Exceptions:-

An error condition during a program execution is called as exception in PL/SQL. It supports programmers to catch such condition using exception block in the program and an appriate action is taken against the error condition their are two types of exceptions.

* system - defined exceptions.

* User - defined exceptions.

Syntax for exception Handling:-

The general syntax for exception handling as follows. Here, you can list down as many as exception you want to handle.

```
DECLARE
<declaration section>
BEGIN
<executable command>
EXCEPTION
<exception handling goes here>
```

```
WHEN EXCEPTION1 THEN
exception1_handling_statements.
WHEN EXCEPTION2 THEN
exception2_handling_statements.
WHEN EXCEPTION3 THEN
exception3_handling_statements.
- - - -
WHEN others THEN
exception3_handling_statements.
END;
```

User-Defined Exceptions:-

PL/SQL allows you to define your own exceptions according to the need of your program. A user-defined exception must be declared and then raised explicity, using either a RAISE statement. (or) the procedure DBMS_STANDARD.RAISE_APPLICATION_ERROR.

syntax:-

```
DECLARE
my_exception EXCEPTION;
```

Result :- Thus, the PL/SQL exceptional handles is created and executed successfully.

EXP. No: 14 creation of Database using Triggers

Date: 30/3/22

Aim :- TO study and execute PL/SQL triggers in database.

Trigger:- A trigger is a stored procedure that defines an action that the database automatically take when some database-related event such as insert, update (or) delete occur.

Types of trigger:-

* Before :- It fires the trigger before executing the trigger statement.
* After :- It fires the trigger after executing the trigger statement.
* For each row:- It specifies that the trigger fires once per row.
* For each statement:- This is the default trigger that is invoked. It specifies that the trigger fires once per statement.

Variables used in Triggers:-

* New
* old

These two variables retain the new and old values of the column updated in database. The values in these variables can be used in the database triggers for data manipulation.

SYNTAX:-

```
create(or) Replace trigger <trg_name> Before/After_Insert/update/Delete
[of column_name, column_name ---]
on <table_name>
[for each row]
[when condition]
begin
--- statement
End.
```

Develop a query to drop the created Trigger:-

```
SQL>drop trigger ittrigg;
Trigger dropped.
```

Result :- Thus, the creation of PL/SQL triggers in database has been executed successfully.