Assignment 4

Advantages of Writing Stored Procedures And Functions

A Stored Procedure is a type of code in SQL that can be used for later use & can be used many times. So whenever you need to execute the querry, instead of calling it you can just use the stored procedure. You can also pass parameters to the stored procedure so that the stored procedure can act based on the parameter values that it has been passed.
Stored function is a program like a stored procedure but it must return a value.

Advantages of Stored Procedures And Stored Functions -

1. Better Performance - The procedure calls are quick and efficient as stored procedures are compiled once and are stored in the executable form. Hence the response is quick. The executable code is automatically cached and hence lowers the memory requirements.
2. Higher Productivity - Productivity is high as the same piece of code is used again and again.
3. Ease of Use - To create a stored procedure, one can use any IDE. Then they can be deployed on any ties of network architecture.
4. Scalability - Stored procedures increase the scalability by isolating the application processing on the server.

5. Maintainability - Maintaining a procedure on a server is much easier than maintaining copies on various machines as the scripts are in one location.

6. Security - Access to oracle data can be restricted by allowing users to manipulate the data only through stored procedures that execute with their definer's previleges

Explain IN, OUT, INOUT type of parameters -

IN parameter - An IN parameter is used to take a parameter as input such as an attribute. When we define an IN parameter in a procedure, the calling program has to pass an argument to the stored procedure.
The value of an IN parameter is protected. It means that even if the value of IN parameter is changed inside the procedure, its original value is retained after the procedure ends (like pass by value). In other words, the procedure only works on the copy of the IN parameter.

OUT parameter - An OUT parameter is used to pass a parameter as output or display like the select operator. The value of an OUT parameter can be changed inside the procedure & its new value is passed back to the calling program.

INOUT parameter - An INOUT parameter is a combination of IN & OUT parameters. It means that the calling program may pass the argument, and the

stored procedure can modify the INOUT parameter & pass the new value back to the calling program.

Difference between Stored Procedures And Functions -

	Stored Procedures	Functions
1.	A procedure does not have a return type but it returns values using the OUT parameters.	A function has a return type and returns a value.
2.	You can use DML querries such as insert, update, select, etc with procedures	You cannot use a function with Data manipulation querries. Only select querries are allowed in functions.
3.	A procedure allows both input & output parameters.	A function does not allow output parameters.
4.	You can manage transactions inside a procedure.	You cannot manage transactions inside a function

	Stored Procedures	Functions
5.	You can call a function from a stored procedure.	You cannot call stored procedures from a function.
6.	You cannot call a stored procedure using a select statement	You can call a function using a select statement.

What are Database Cursors?

When processing a querry, all information pertaining to it is stored in a context area. A cursor is a pointer to this context area. A cursor allows handling the result set of a querry. Cursor is used to store database tables.

Explain Declare, open fetch and close operations on a DB cursor.

1. Declare Cursor Object.

For using a cursor inside a stored procedure, function or trigger it must first be declared.

Syntax → DECLARE cursor_name CURSOR FOR
SELECT * FROM table;

2. Opening the cursor – Opening the cursor initializes the result set for the cursor allowing access to the data.

Syntax -

```
OPEN cursor_name;
```

3. Fetch operations - The rows of the result set can be accessed one at a time by moving its contents into variables using a fetch statement.

Syntax -

```
FETCH cursor_name INTO var1, var2, ... varn;
```

4. Close the cursor connection - The cursor resources are deallocated.

Syntax -

```
CLOSE cursor_name;
```

Explain Exception handling in MySQL.

An exception is an error which disrupts the normal flow of program instructions.

PL/SQL provides us the exception block which raises the exception thus helping the programmer to find the fault & resolve it.

MySQL provides an easy way to define handlers that handles general conditions such as warnings or exceptions to specific conditions → eg: specific error codes. If a condition whose value matches condition_value, MySQL will execute the statement & continue/exit the current code block based on the action.

The action accepts one of the following values -

CONTINUE - the execution of the enclosing code block

(BEGIN ... END) continues.

EXIT → the execution of the enclosing code block where the handler is declared terminates.

Syntax –

DECLARE action HANDLER condition_value statement;

↑ code/name of condition

↑ executed if condition occurs