Day-6

Agenda

- Functions and Cursor
- packages and subprograms
- exception & performance tuning

Packages and subprograms

Exception & Performance tuning

Functions

- MySQL functions is the fact that it can help reducing network traffic in client/server applications.

- A stored function is a set of SQL statements that perform some operation and return a single value. Just like Mysql in-built function, it can be called from within a Mysql statement.

- Functions are simply pieces of code that perform some operations and then return a result.

- A stored function in MySQL is a set of SQL statements that perform some task/operation and return a single value. It is one of the types of stored programs in MySQL.

- MySQL Group By Clause − The MySQL GROUP BY statement is used along with the SQL aggregate functions like SUM to provide means of grouping the result dataset by certain database table column(s).

- MySQL IN Clause − This is a clause, which can be used along with any MySQL query to specify a condition.

- MySQL BETWEEN Clause − This is a clause, which can be used along with any MySQL query to specify a condition.

- MySQL UNION Keyword − Use a UNION operation to combine multiple result sets into one.

- MySQL String Functions − Complete list of MySQL functions required to manipulate strings in MySQL.

- MySQL COUNT Function − The MySQL COUNT aggregate function is used to count the number of rows in a database table.

- MySQL MAX Function − The MySQL MAX aggregate function allows us to select the highest (maximum) value for a certain column.

- MySQL MIN Function − The MySQL MIN aggregate function allows us to select the lowest (minimum) value for a certain column.

- MySQL AVG Function − The MySQL AVG aggregate function selects the average value for certain table column.

- MySQL SUM Function − The MySQL SUM aggregate function allows selecting the total for a numeric column.

- MySQL SQRT Functions − This is used to generate a square root of a given number.
- MySQL RAND Function − This is used to generate a random number using MySQL command.
- MySQL CONCAT Function − This is used to concatenate any string inside any MySQL command.
- MySQL DATE and Time Functions − Complete list of MySQL Date and Time related functions.
- MySQL Numeric Functions − Complete list of MySQL functions required to manipulate numbers in MySQL.

PL/SQL - Cursors

- A **cursor** is a pointer to 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
  - Explicit cursors

- When an SQL statement is processed, Oracle creates a memory area known as context area.
- A cursor is a pointer to this context area.
- It contains all information needed for processing the statement.
- In PL/SQL, the context area is controlled by Cursor.
- A cursor contains information on a select statement and the rows of data accessed by it.

PL/SQL Implicit Cursors

- If the Oracle engine opened a cursor for its internal processing it is known as an Implicit Cursor. It is created “automatically” for the user by Oracle when a query is executed and is simpler to code.

- 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.

- The cursor attributes available are %FOUND, %NOTFOUND, %ROWCOUNT, and %ISOPEN.

- The implicit cursors are automatically generated by Oracle while an SQL statement is executed, if you don't use an explicit cursor for the statement.

Attribute	Description
%FOUND	Its return value is TRUE if DML statements like INSERT, DELETE and UPDATE affect at least one row or more rows or a SELECT INTO statement returned one or more rows. Otherwise it returns FALSE.
%NOTFOUND	Its return value is TRUE if DML statements like INSERT, DELETE and UPDATE affect no row, or a SELECT INTO statement return no rows. Otherwise it returns FALSE. It is a just opposite of %FOUND.
%ISOPEN	It always returns FALSE for implicit cursors, because the SQL cursor is automatically closed after executing its associated SQL statements.
%ROWCOUNT	It returns the number of rows affected by DML statements like INSERT, DELETE, and UPDATE or returned by a SELECT INTO statement.

PL/SQL Explicit Cursors

- The Explicit cursors are defined by the programmers to gain more control over the context area.

- These cursors 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.

- Following is the syntax to create an explicit cursor:

  CURSOR cursor_name **IS** select_statement;;

- Steps:
  - Declare the cursor
  - Open the cursor
  - Fetch the cursor
  - Close the cursor