19B21A0435

PALAGIRI ANJIREDDY

KAKINADA INSTITUTE OF ENGINEERING AND TECHNOLOGY

SQL Assignment 5

1 .Explain how SQL Query Keyword Statement are executed in order.

(Ans)-

Step 1: Getting Data (From, Join) FROM citizen.

Step 2: Row Filter (Where) After getting qualified rows, it is passed on to the Where clause.

Step 3: Grouping (Group by)

Step 4: Group Filter (Having)

Step 5: Return Expressions (Select)

Step 6: Order (Order by) and Paging (Limit / Offset)

2.Explain the advantages of stored producers and their syntax in relation to recompiling stored procedures.

(Ans)-

Since stored procedures are compiled and stored, whenever you call a procedure the response is quick.

you can group all the required SQL statements in a procedure and execute them at once.

Since procedures are stored on the database server which is faster than client. You can execute all the complicated quires using it, which will be faster.

Using procedures, you can avoid repetition of code moreover with these you can use additional SQL functionalities like calling stored functions.

Once you compile a stored procedure you can use it in any number of applications. If any changes are needed you can just change the procedures without touching the application code.

You can call PL/SQL stored procedures from Java and Java Stored procedures from PL/SQL.

3.Give an example of the derived tables.

Ans)-

It is a simple example to demonstrate the SQL Server derived table. The following SQL Query will display all the columns present in the Employees table whose sales amount is greater than 500.

SELECT \* FROM

(

SELECT [EmpID]

,[FirstName]

,[LastName]

,[Education]

,[YearlyIncome]

,[Sales]

,[DeptID]

FROM [EmployeeDetails]

) AS [Derived Employee Details]

WHERE [Sales] > 500

First SELECT \* statement is deriving

4. What is the databases trigger? Explain the difference forms of triggers that can be found in the database.

Ans)---Trigger: A trigger is a stored procedure in database which automatically invokes whenever a special event in the database occurs. For example, a trigger can be invoked when a row is inserted into a specified table or when certain table columns are being updated.

5. What are the benefits and drawbacks of triggers?

Benifits of SQL Server Triggers

Triggers are easy to code. The fact that they are coded like stored procedures which makes getting started with triggers easy.

Triggers allow you to create basic auditing. By using the deleted table inside a trigger you can build a decent audit solution that inserts the contents of the deleted table data into an audit table which holds the data that is either being removed by a DELETE statement or being changed by an UPDATE statement.

You can call stored procedures and functions from inside a trigger.

Drawbacks on SQL Server Triggers

When you use BULK INSERT to insert data into a table, triggers are not fired unless you include the FIRE\_TRIGGERS option in your bulk insert statement. This is very important to keep in mind, because if you have triggers on a table as part of the business logic, and you make a bulk insert without including the FIRE\_TRIGGERS option you will lose data consistency.

Triggers are difficult to locate unless you have proper documentation because they are invisible to the client. For instance, sometimes you execute a DML statement without errors or warnings, say an insert, and you don't see it reflected in the table's data. In such case you have to check the table for triggers that may be disallowing you to run the insert you wanted.

6. Create a stored procedure of call other stored procedures?

Ans)—

Step 1: Create two simple stored procedure to insert some data into two different tables.

usp\_insert\_into\_Log1 to insert data into tbl\_log1

usp\_insert\_into\_Log2 to insert data into tbl\_log2

both accept four parameters to insert the data.

Step 2: Here is a requirement that if less then 50000 rows filled in tbl\_log1 then insert data into tbl\_log1, otherwise another table tbl\_log2.

Here is an example of how to call a stored procedure within another stored procedure.

CREATE PROCEDURE [dbo].[usp\_insert]

(

@a varchar(50),

@b varchar(15),

@c varchar(6),

@d varchar(50)

)

AS

BEGIN

if ((select count(\*) from tbl\_Log1) <50000)

exec [dbo].[usp\_insert\_into\_Log1] @a,@b,@c,@d

else

exec [dbo].[usp\_insert\_into\_Log2] @a,@b,@c,@d

END