**SQL Assignment 4**

1. Explain different types of views. Demonstrate with suitable examples.

Answer:

There are 2 types of Views in SQL: Simple View and Complex View. Simple views can only contain a single base table. Complex views can be constructed on more than one base table. In particular, complex views can contain: join conditions, a group by clause, a order by clause.

1.Simple View:

When user wants data or some columns from same table then simple view is used. For an example if there is employee table which has Employee\_num,Employee\_name,salary columnns and we just need to see Employee\_num,Employee\_name then user can create a simple view.

|  |  |  |  |
| --- | --- | --- | --- |
| Employee\_num | Employee\_name | Department | Salary |
| 1 | Amit | OBIEE | 680000 |
| 2 | Rohan | OBIEE | 550000 |
| 3 | Rohit | OBIEE | 430000 |

We want to restrict data from the above table and show only Employee\_Num and Employee\_name then we will use the view.

Syntax:

Create view Viewname

as Select column\_name1,Coumn\_name2 from tablename.

Example:

Create view V\_Employee

as Select Employee\_num,Employee\_name from Employee;

You can access the view using Select statement of SQL:

select \* from V\_Employee;

Output:

|  |  |
| --- | --- |
| Employee\_num | Employee\_name |
| 1 | Amit |
| 2 | Rohan |
| 3 | Rohit |

2. Complex View:

Complex view is created on using more than one tables. When user wants to retrieve data from more than 1 table then we have to use complex views. To create complex view there should be relation between 2 tables else cartesian product will come by joining 2 tables. There should be some joining conditions, some filters needs to be considered while creating complex views.  
http://tracking.vcommission.com/aff_i?offer_id=22&file_id=124708&aff_id=60738  
Suppose there are 2 tables:

1.Employee tables:-Employee\_name,Employee\_num,Department\_code columns

2.Department:-Department\_code,Deptartment\_name columns

We need to create view where we want to show the associated department to employee.Here We need to use complex join.

Create view V\_EMP\_DEPT

as Select e.Employee\_name,d.Department\_name

from Employee e,Department d

where e.Department\_code=d.Department\_code;

1. What is the difference between function and stored procedure? Write syntax for creating functions and stored procedures.

Answer:

Stored Procedures:

Stored Procedures are pre-compiled objects which are compiled for the first time and its compiled format is saved, which executes (compiled code) whenever it is called.

Syntax:

CREATE PROCEDURE procedure\_name

AS

sql\_statement

GO;

Functions:

A function is compiled and executed every time whenever it is called. A function must return a value and cannot modify the data received as parameters.

Syntax:

CREATE [OR REPLACE] FUNCTION function\_name

[(parameter\_name [IN | OUT | IN OUT] type [, ...])]

RETURN return\_datatype

{IS | AS}

BEGIN

< function\_body >

END [function\_name];

Differences between Stored Procedure and Function:

* The procedure allows SELECT as well as DML(INSERT/UPDATE/DELETE) statement in it whereas Function allows only SELECT statement in it.
* Procedures cannot be utilized in a SELECT statement whereas Function can be embedded in a SELECT statement.
* Stored Procedures cannot be used in the SQL statements anywhere in the WHERE/HAVING/SELECT section whereas Function can be.
* Functions that return tables can be treated as another rowset. This can be used in JOINs with other tables.
* Inline Function can be though of as views that take parameters and can be used in JOINs and other Rowset operations.
* An exception can be handled by try-catch block in a Procedure whereas try-catch block cannot be used in a Function.
* We can use Transactions in Procedure whereas we can't use Transactions in Function.

1. What is an index in SQL? What are the different types of indexes in SQL?

Answer:

Indexes are used to retrieve data from the database more quickly than otherwise. The users cannot see the indexes, they are just used to speed up searches/queries.

Clustered Index:

A clustered index defines the order in which data is physically stored in a table. Table data can be sorted in only way, therefore, there can be only one clustered index per table. In SQL Server, the primary key constraint automatically creates a clustered index on that particular column.

Non-Clustered Indexes:

A non-clustered index doesn’t sort the physical data inside the table. In fact, a non-clustered index is stored at one place and table data is stored in another place. This is similar to a textbook where the book content is located in one place and the index is located in another. This allows for more than one non-clustered index per table.

1. Showcase an example of exception handling in SQL stored procedure.

Answer:

An error condition during a program execution is called an exception and the mechanism for resolving such an exception is known as exception handling.

Declare @val1 int;

Declare @val2 int;

BEGIN TRY

Set @val1=8;

Set @val2=@val1/0; /\* Error Occur Here \*/

END TRY

BEGIN CATCH

Print 'Error Occur that is:'

Print Error\_Message()

END CATCH

1. Create a SQL function to split strings into rows on a given character?

Input String: Stephen;peter;berry;Olivier;caroline;

|  |
| --- |
| Stephen |
| Peter |
| Berry |
| Oliver |
| Caroline |

Answer:

DECLARE @string\_value VARCHAR(MAX) ;

SET @string\_value=' Stephen;peter;berry;Olivier;caroline;'

SELECT \* FROM STRING\_SPLIT (@string\_value, ';')

1. What is a temporary and a variable table? Write suitable syntax to create temporary tables and variable tables.

Answer:

Temporary:

Temporary Tables are physically created in the tempdb database. These tables act as the normal table and also can have constraints, index like normal tables.

Syntax:

CREATE TABLE #EmpDetails (id INT, name VARCHAR(25))

Table Variable:

Table Variable acts like a variable and exists for a particular batch of query execution. It gets dropped once it comes out of batch.

Syntax:

Declare @TempTable TABLE(id int,Name varchar(20))