Designing and Implementing Views

Before you create a view, consider the following guidelines:

- You can create views only in the current database. However, the tables and views referenced by the new view can exist in other databases or even other servers if the view is defined using distributed queries.
- View names must follow the rules for identifiers and must be unique for each schema. Additionally, the name must not be the same as any tables contained by that schema.
- You can build views on other views. Microsoft SQL Server allows views to be nested. Nesting may not exceed 32 levels. The actual limit on nesting of views may be less depending on the complexity of the view and the available memory.
- You cannot associate rules or DEFAULT definitions with views.
- You cannot associate AFTER triggers with views, only INSTEAD OF triggers.
- The query defining the view cannot include the COMPUTE or COMPUTE BY clauses, or the INTO keyword.
- The query defining the view cannot include the ORDER BY clause, unless there is also a TOP clause in the select list of the SELECT statement.
- The query defining the view cannot contain the OPTION clause specifying a query hint.
- The query defining the view cannot contain the TABLESAMPLE clause.
- You cannot define full-text index definitions on views.
- You cannot create temporary views, and you cannot create views on temporary tables.
- Views, tables, or functions participating in a view created with the SCHEMABINDING clause cannot be dropped, unless the view is dropped or changed so that it no longer has schema binding. In addition, ALTER TABLE statements on tables that participate in views having schema binding will fail if these statements affect the view definition.
- If a view is not created with the SCHEMABINDING clause, sp_refreshview should be run when changes are made to the objects underlying the view that affect the definition of the view. Otherwise, the view might produce unexpected results when it is queried.
- You cannot issue full-text queries against a view, although a view definition can include a full-text query if the query references a table that has been configured for full-text indexing.
- You must specify the name of every column in the view if:
    - Any of the columns in the view are derived from an arithmetic expression, a built-in function, or a constant.
    - Two or more of the columns in the view would otherwise have the same name (usually because the view definition includes a join and the columns from two or more different tables have the same name).
    - You want to give any column in the view a name different from the column from which it is derived. (You can also rename columns in the view.) A view column inherits the data type of the column from which it is derived, whether or not you rename it.

Note

This rule does not apply when a view is based on a query containing an outer join, because columns may change from not allowing null values to allowing them.

Otherwise, you do not need to specify column names when creating the view. SQL Server gives the columns of the view the same names and data types as the columns to which the query defining the view refers. The select list can be a full or partial list of the column names in the base tables.

To create a view you must be granted permission to do so by the database owner and, if the view is created with the SCHEMABINDING clause, you must have appropriate permissions on any tables or views referenced in the view definition.

By default, as rows are added or updated through a view, they disappear from the scope of the view when they no longer fall into the criteria of the query defining the view. For example, a query can be created, defining a view that retrieves all rows from a table where the employee's salary is less than $30,000. If the employee's salary is increased to $32,000, then querying the view no longer displays that particular employee because his or her salary does not conform to the criteria set by the view. However, the WITH CHECK OPTION clause forces all data modification statements executed against the view to adhere to the criteria set within the SELECT statement defining the view. If you use this clause, rows cannot be modified in a way that causes them to disappear from the view. Any modification that would cause this to happen is canceled and an error is displayed.

To create a view

- CREATE VIEW (Transact-SQL)
- How to: Create Views (Visual Database Tools)

See Also

Concepts

Designing Indexed Views
Creating Partitioned Views
Modifying Data Through a View
View Resolution

Community Additions