Morgan Tucker
2/17/21
IT FDN 130A
Assignment 06
GitHub

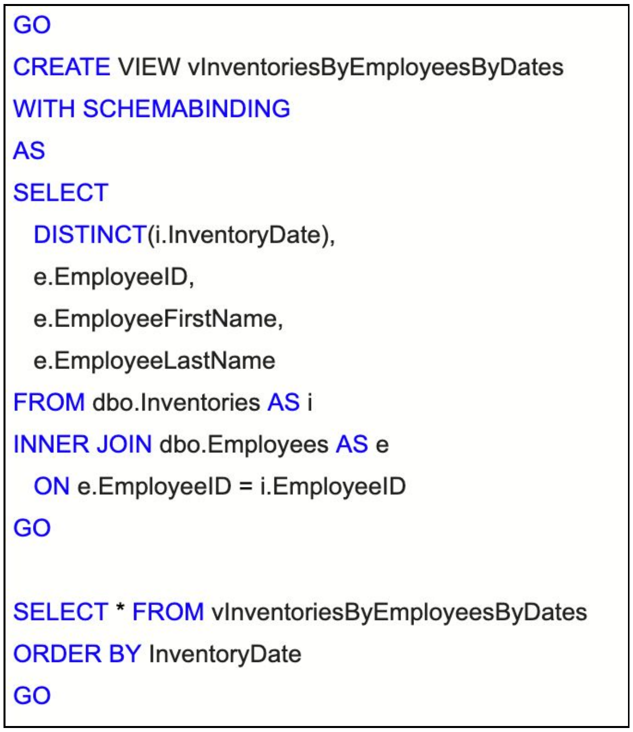
SQL Views & Stored Procedures

Introduction

SQL Views and stored procedures are types of database abstraction layers. These can be used as methods of maintaining data integrity, and can also help with security, access, and keeping downstream applications running smoothly.

When to Use Views

A view table is a representation of the original table and allows users to work with that data indirectly. Views can be used to simplify the use of a complex select statement. Rather than retyping it later to access the same data, you can select all from the view to retrieve the same results. This allows others to use the code that was first written, making reporting easier. Selecting all from a view, and then using the order by clause, is often preferred to using an order by clause directly in the code that creates the view. Figure 1 shows how selecting all, represented by an asterisk, from the view table can be done rather than retyping the select statement used to create the view.

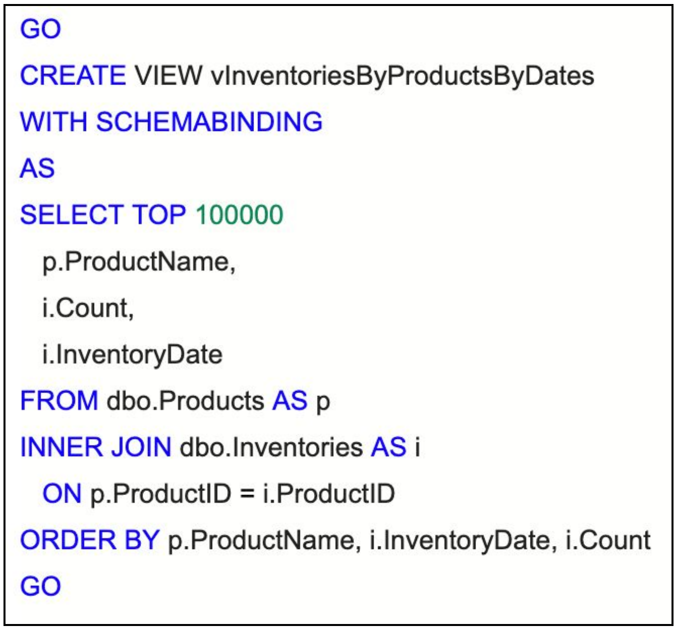


```
GO
CREATE VIEW vInventoriesByEmployeesByDates
WITH SCHEMABINDING
AS
SELECT
  DISTINCT(i.InventoryDate),
  e.EmployeeID,
  e.EmployeeFirstName,
  e.EmployeeLastName
FROM dbo.Inventories AS i
INNER JOIN dbo.Employees AS e
  ON e.EmployeeID = i.EmployeeID
GO

SELECT * FROM vInventoriesByEmployeesByDates
ORDER BY InventoryDate
GO
```

[Fig.1]

Views may also be implemented in order to prevent changes to the database that may be made by mistake. This can be done with SchemaBinding, which shows an error message when a field would be dropped in the table that would result in an orphan field in the view. Figure 2 shows the use of SchemaBinding in creating a view. Note that in this select statement we see the Top function used. This is a way of tricking the system to enable the order by clause, which would not otherwise work.



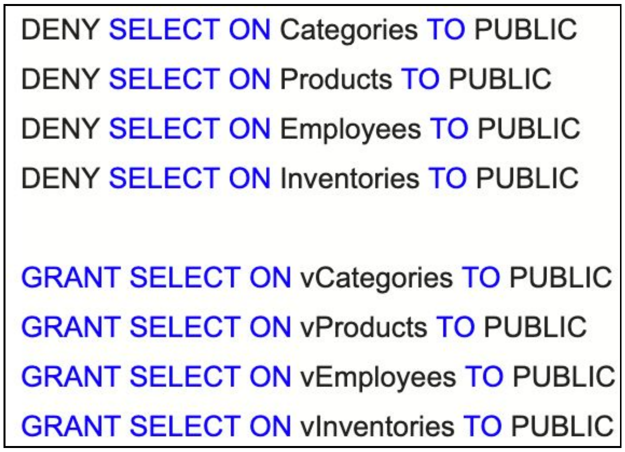
```
GO
CREATE VIEW vInventoriesByProductsByDates
WITH SCHEMABINDING
AS
SELECT TOP 100000
  p.ProductName,
  i.Count,
  i.InventoryDate
FROM dbo.Products AS p
INNER JOIN dbo.Inventories AS i
  ON p.ProductID = i.ProductID
ORDER BY p.ProductName, i.InventoryDate, i.Count
GO
```

[Fig.2]

Views, Functions, and Stored Procedures

Views, functions, and stored procedures are three types of abstraction layers that act as a buffer between applications (or users) and original databases. They can be used to restrict data allowed to be added into a field.

Views are representations of a table, and can be treated as tables themselves, allowing users to select data from views rather than calling the table directly. Views make it simple to restrict access to specific users or groups of users. Figure 3 shows an example of this. We see that the public user group is denied access to Categories, Products, Employees, and Inventories tables. The view tables for each, represented by aliases vCategories, vProducts, vEmployees, and vInventories, grant access to public users. By limiting what fields are selected in these view tables, we limit what the public group may see.



```
DENY SELECT ON Categories TO PUBLIC
DENY SELECT ON Products TO PUBLIC
DENY SELECT ON Employees TO PUBLIC
DENY SELECT ON Inventories TO PUBLIC

GRANT SELECT ON vCategories TO PUBLIC
GRANT SELECT ON vProducts TO PUBLIC
GRANT SELECT ON vEmployees TO PUBLIC
GRANT SELECT ON vInventories TO PUBLIC
```

[Fig.3]

Unlike views, functions can be used in the select clause of a select statement as an expression. This allows adding scalar function, taking one or more parameters and returning a single value. These can also be used as a check constraint.

Stored procedures are used when complexity compels us to avoid writing simple views. In cases where having or group by statements are used, the where clause uses a subquery, or select clause uses distinct function or functions, views are not an option.

Conclusion

Using view tables for abstraction allows users to interact with tables without directly touching them. Views are sometimes easier than writing a function or stored procedure when there is less complexity in the data being retrieved. For more complex work, views are not always the best option, and functions or stored procedures work better. All three options are essentially saved versions of select statements.