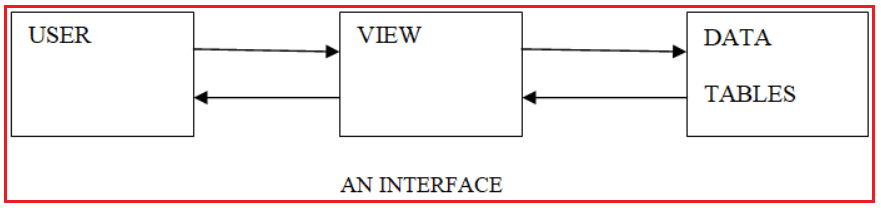
**Views in SQL Server with Examples**

##### **What is a View in SQL Server?**

The views in SQL Server are nothing more than a compiled SQL query. We can also consider the Views as virtual tables. As a virtual table, the Views do not store any data physically by default. So when we query a view it actually gets the data from the underlying database tables as shown in the below image.



Simply we can say that the views in SQL Server act as an interface between the Table(s) and the user.

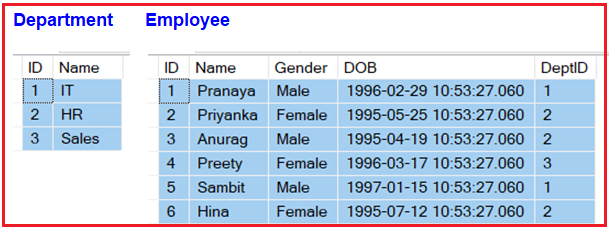
##### **What are the differences between a table and a view in SQL Server?**

When we compared a view with a table we have the following differences.

1. The table is physical whereas the view is logical or virtual.
2. A Table is an independent object whereas a view is a dependent object that is a view depends on the table or tables from which it is loading the data.
3. When a new table is created from an existing table the new and old tables are independent themselves that is the changes of one table will not be reflected into the other table whereas if a view is created based on a table any changes that are performed on the table reflects into the view and any changes performed on the view reflected in the table also.

##### **Examples to understand Views in SQL Server**

We are going to use the following Department and Employee table to understand the Views in SQL Server.



**Please use the below SQL Script to create and populate the Department and Employee table with sample data.**

-- Create Department Table

**CREATE** **TABLE** Department

(

**ID** **INT** **PRIMARY KEY**,

Name **VARCHAR**(50)

)

**GO**

-- Populate the Department Table with test data

**INSERT** **INTO** Department **VALUES**(1, 'IT')

**INSERT** **INTO** Department **VALUES**(2, 'HR')

**INSERT** **INTO** Department **VALUES**(3, 'Sales')

-- Create Employee Table

**CREATE** **TABLE** Employee

(

**ID** **INT** **PRIMARY KEY**,

Name **VARCHAR**(50),

Gender **VARCHAR**(50),

**DOB** **DATETIME**,

DeptID **INT**

)

**GO**

-- Populate the Employee Table with test data

**INSERT** **INTO** Employee **VALUES**(1, 'Pranaya', 'Male','1996-02-29 10:53:27.060', 1)

**INSERT** **INTO** Employee **VALUES**(2, 'Priyanka', 'Female','1995-05-25 10:53:27.060', 2)

**INSERT** **INTO** Employee **VALUES**(3, 'Anurag', 'Male','1995-04-19 10:53:27.060', 2)

**INSERT** **INTO** Employee **VALUES**(4, 'Preety', 'Female','1996-03-17 10:53:27.060', 3)

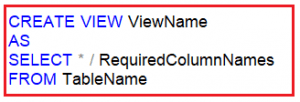
**INSERT** **INTO** Employee **VALUES**(5, 'Sambit', 'Male','1997-01-15 10:53:27.060', 1)

**INSERT** **INTO** Employee **VALUES**(6, 'Hina', 'Female','1995-07-12 10:53:27.060', 2)

**GO**

##### **The syntax for creating a View in SQL Server:**

The View will be created by using the CREATE View ViewName statement followed by the select statement as shown in the below image.



##### **How many types of views are there in SQL Server?**

There are two types of views in SQL Server, they are

1. **Simple Views**
2. **Complex Views**

##### **What is a simple view or an Updatable view in SQL Server?**

The view that is created based on the columns of a single table, then it is known as a simple view. We can perform all the DML operations on a simple view. This is the reason why a simple view can also be called an updatable view or dynamic view.

Let us understand the simple view in SQL with an example. Please have a look at the below SQL statements which creates two simple views. Further, if you notice, both the Views are created based on a single table i.e. Employee. In the first view i.e. vwAllEmployees1, we are retrieving all the columns of the Employee table by using \* and in the second view i.e. vwAllEmployees2, we are specifying the column names explicitly.

-- View with all columns

CREATE VIEW vwAllEmployees1

AS

SELECT \* FROM Employee

-- View with specific columns

CREATE VIEW vwAllEmployees2

AS

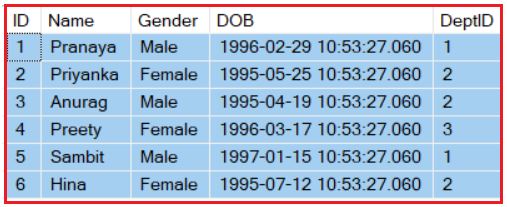
SELECT ID, Name, Gender, DOB, DeptID

FROM Employee

The above two views are examples of Simple or Updatable Views in SQL Server. And, we can perform all the DML operations (INSERT, UPDATE, and DELETE) on the above views including the SELECT operations.

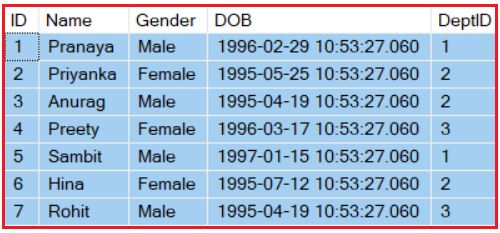
##### **DML Operations on the Simple Views in SQL Server:**

Let us see how to perform the SELECT, INSERT, UPDATE, and DELETE operations on the above-created simple views i.e. either vwAllEmployees1 or vwAllEmployees2. To select data from a view, we need to use the SELECT statement, in the same way, we used to select the data from a table as shown below.  
**SELECT \* from vwAllEmployees1**  
Once you execute the above query, you will get the following output.



##### **Insert Operation on Simple Views in SQL Server**

On a simple view in SQL Server, we can perform the INSERT operation. Let us see this with an example. Please execute the following INSERT statement to Insert an employee into the Employee table using the vwAllEmployees1view.  
**INSERT INTO vwAllEmployees1 (ID, Name, Gender, DOB, DeptID) VALUES(7, ‘Rohit’, ‘Male’, ‘1995-04-19 10:53:27.060’, 3)**  
When you execute the above insert statement, it will successfully insert the record into the Employee table. To check issue a select query against the Employee table as shown below/  
**SELECT \* FROM Employee**  
Once you execute the above query, you will get the following output.



As you can see in the above image, the new record is inserted into the Employee table.

##### **Update Operation on View:**

On a simple view, we can also perform the UPDATE operation. Let us understand this with an example. Let’s try to update the details of the employee whose id is 7 using the vwAllEmployees1 view. Please execute the following UPDATE statement which will UPDATE employee details using the view.

UPDATE vwAllEmployees1 SET

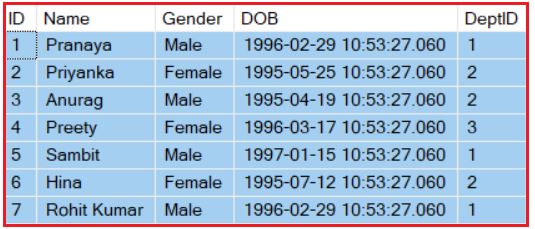
Name = 'Rohit Kumar',

DOB = '1996-02-29 10:53:27.060',

DeptID = 1

WHERE ID = 7

Once you execute the above UPDATE SQL Query, it will update the data in the Employee table. To check whether the data is updated successfully or not make a select query against the Employee table as shown below.  
**SELECT \* FROM Employee**  
Once you execute the above SELECT statement, you will get the following output.



As you can see the Employee whose ID is 7 is successfully updated as expected.

##### **Delete Operation on View:**

On a simple view, it is also possible to perform the UPDATE DML operation. Let us understand this with an example. Let’s try to Delete the Employee whose ID is 7 using the vwAllEmployees1 view. Please execute the following DELETE statement which will DELETE the employee record from the table Employee.    
**DELETE FROM vwAllEmployees1 where ID = 7**  
Once we execute the above delete operation then check the Employee table and you will see that the Employee whose ID is 7 is deleted as expected from the Employee table. So this proofs that we can perform ALL the DML operations on a Simple View which is also called Updatable View in SQL Server.

##### **What is a complex view in SQL Server?**

When we create a view based on more than 1 table then it is known as a complex view and on a complex view, we may or may not perform the DML (INSERT, UPDATE, and DELETE) operations and more ever the complex view may not update the data correctly on the underlying database tables.

##### **Creating Complex Views**

Please have a look at the following code. Here, we created the view by using joining the Employee and Department tables, and hence it is a complex view. Further, if you notice, here we are selecting the columns from both tables.

CREATE VIEW vwAllEmployees3

AS

SELECT emp.ID, emp.Name, emp.Gender, emp.DOB,

dep.Name **as** DepartmentName

FROM Employee emp

INNER JOIN Department dep

**on** emp.DeptID = dep.ID

A view that is created based on a single table will also be considered as a complex view provided if the query contains Distinct. Aggregate Function, Group By Clause, having Clause, calculated columns, and set operations. For better understanding, please have a look at the following query. Here, we are using a single table as well as we are also using the Count aggregate function which makes the view a complex view and also restricts us to perform any DML operations.

**CREATE** **VIEW** vwAllEmployees4

**AS**

**SELECT** Gender, Count(\*) **as** TotalEmployee

**FROM** Employee Group **BY** Gender

In our upcoming article, we will discuss [**Complex Views in SQL Server**](https://dotnettutorials.net/lesson/complex-views-in-sql-server/) in detail with Real-time Examples.

##### **Can we drop a table that has dependent views on it?**

Yes, you can drop a table even if any dependent views are associated with it, but the views that are associated with it will not be dropped. They will still execute in the database only with the status as inactive object and all those views become active and start functioning provided the table is recreated.

##### **Can we create a view based on other views?**

Yes, It is possible in SQL Server to create a view based on other views.

##### **Can we update the views in SQL Server?**

Yes, in SQL server views can be updated. However, updating a view that is based on multiple tables, may not update the base tables correctly. To correctly update a view that is based on multiple tables we can make use of [**INSTEAD OF triggers in SQL Server**](https://dotnettutorials.net/lesson/instead-of-insert-trigger-in-sql-server/)**.**