18. Views and Triggers

The view is a virtual table. It is not created physically in a database. The view contains only query definitions. It doesn't contain data. It takes only a small space to create a view. The view will return a table. We can write queries by considering the view as a table.

Table: Appointments

Id	Date	DoctorId	PatientId
1	2020-06-12	1	2
2	2020-06-13	3	2
3	2020-06-14	3	1
4	2020-06-13	1	4
5	2020-06-13	3	6
6	2020-06-14	2	3
7	2020-06-15	2	2
8	2020-06-15	2	2

Table: Doctors

Id	Name
1	Williams
2	Smith
3	Clark
4	Johnson

Table: Patients

Id	Name
1	Robert
2	James
3	David
4	Michael
5	Oliver
6	Mary

Data Script:

```
CREATE TABLE Doctors(
    Id int NOT NULL,
    Name varchar(50) NOT NULL,
    PRIMARY KEY (Id)
);

CREATE TABLE Patients(
    Id int NOT NULL,
    Name varchar(50) NOT NULL,
    PRIMARY KEY (Id)
);

CREATE TABLE Appointments(
    Id int PRIMARY KEY,
    Date date NOT NULL,
    DoctorId int NOT NULL FOREIGN KEY REFERENCES Doctors(Id),
    PatientId int NOT NULL FOREIGN KEY REFERENCES Patients(Id),
    CreatedDate DATETIME,
    ModifiedDate DATETIME
);

INSERT INTO Doctors(Id, Name)
VALUES (1, 'Williams'), (2, 'Smith'), (3, 'Clark'), (4, 'Johnson');

INSERT INTO Patients(Id, Name)
VALUES (1, 'Robert'), (2, 'James'), (3, 'David'), (4, 'Michael'), (5, 'Oliver'),
(6, 'Mary');

INSERT INTO Appointments(Id, Date, DoctorId, PatientId)
VALUES (1, '06/12/2020', 1, 2), (2, '06/13/2020', 3, 2),(3, '06/14/2020', 3, 1),
(4, '06/13/2020', 1, 4), (5, '06/13/2020', 3, 6), (6, '06/14/2020', 2, 3);
```

Syntax:

```
CREATE VIEW ViewAppointments
AS SelectStatement1
```

Below query will create a view to display appointment details with patient details.

Script:

```
CREATE VIEW ViewAppointmentsWithPatientDetails

AS
SELECT Appointments.Id, Appointments.Date, Patients.Name AS PatientName
, Appointments.DoctorId
FROM Appointments
LEFT JOIN Patients ON Appointments.PatientId = Patients.Id;
```

ViewAppointmentsWithPatientDetails will work as a table. We can write select statements as shown below.

```
SELECT * FROM ViewAppointmentsWithPatientDetails;
```

SQL Join with a View

A view can be joined with another table or with another view. Join queries are the same as normal table joins. The below query shows how to join the above view with the Doctors table.

Script:

```
SELECT ViewAppointmentsWithPatientDetails.Id, Date, PatientName
, Name AS DoctorName
FROM ViewAppointmentsWithPatientDetails INNER JOIN Doctors
ON ViewAppointmentsWithPatientDetails.DoctorId = Doctors.Id;
```

Below query will create a view to display appointment details with patient and doctor details.

Script:

```
CREATE VIEW ViewAppointmentsWithPatientAndDoctorDetails

AS
SELECT Appointments.Id, Appointments.Date, Patients.Name AS PatientName
, Doctors.Name AS DoctorName
FROM Appointments
LEFT JOIN Patients ON Appointments.PatientId = Patients.Id
LEFT JOIN Doctors ON Appointments.DoctorId = Doctors.Id;
```

Triggers in SQL

The trigger is a special type of stored procedure. The trigger is attached to a table or view. Triggers will function for INSERT, DELETE or UPDATE events of a table or a view. You can see available triggers in the Trigger folder attached to the table shown below.
Use the below script to create new fields in Appointments table to test using the trigger

The following trigger will calculate and update ModifiedDate and CreatedDate fields for inserted records in the Appointments table.

```
CREATE TRIGGER TR_UpdateAppointmentsCreatedDate ON Appointments
FOR INSERT
AS

BEGIN
      DECLARE @Date DATETIME = GETDATE();
      UPDATE Appointments SET CreatedDate = @Date, ModifiedDate = @Date
      WHERE Id IN (SELECT DISTINCT Id FROM Inserted);
END
```

The following trigger will calculate and update the ModifiedDate filed for updated records in the Appointments table.

```
CREATE TRIGGER TR_UpdateAppointmentsModifiedDate ON Appointments
FOR UPDATE
AS

BEGIN
  UPDATE Appointments SET ModifiedDate = GETDATE()
  WHERE Id IN (SELECT DISTINCT Id FROM Inserted);
END
```

Once you add a new record to the Appointments table you notice that CreatedDate and Modfied Date will be updated. Also, ModifiedDate will be updated for record updations.

A trigger attached with IA trigger can be used to execute a query in the same table or in a different table