Views

View

- A virtual table that represents the data in one or more tables in an alternative way

- NO physical existence in database

- It contains data at run time only

- It can be used to
  - Select
  - Insert
  - Update
  - Delete data from the tables

Views Continued….

- By using View we can join multiple tables and present the data as coming from a single table

- It is used for security purposes i.e. when you want to hide the actual tables or column of tables participating in the query

- View can be used in other Views and Stored Procedures

- View does not accept any parameter

- View cannot use Stored Procedure in it

Operations

▶ Create

Can Only select query in body

```
CREATE VIEW ViewName
AS
SELECT * FROM TableName
```

▶ Drop

```
DROP VIEW ViewName
```

▶ Call

```
SELECT * FROM ViewName
```

▶ Alter

```
ALTER view ViewName
as
select Column1 from
Table1 join table1
table1.c1=table2.C1
GO
```

Insert Data Through View

- The data will be inserted in tables from which view is created
- Data will only be inserted if not constraints on underlying tables in violated

```
create table customer2
(customerID int not null
,
customername varchar(10) null
)
go

create view Cust2
as
select customerID from customer2

go

insert into Cust2 values(2)
--This will run successfully
--A new row will be inserted in Customer2
--with values (2,nul)

go
```

```
create table customer1
(customerID int not null
,
customername varchar(10) not null
)
go

create view Cust1
as
select customerID from customer1
go

insert into Cust1 values(2)
--This inseration will give error
--As customer name cannot be null in Customer1
go
```

Delete and Update through View

- Delete and update work similarly a Insert
- If Deletion and Update does not violate and constraints on underlying table

Views WITH CHECK Option

- To restrict the changes that users can make

- Use the CREATE VIEW command's WITH CHECK OPTION clause when defining the view

- Note that any data can be added directly in the tables, but not through view

Views WITH CHECK Option Example

- Create a view withCheck Option

```
CREATE   VIEW v_EmployeeDetailWithCheckOption
AS
  SELECT E.*
  FROM Employee E
  WHERE E.EmployeeID < 15
WITH CHECK OPTION
```

- Adding any data via this view that does not fulfill where clause will not be inserted

```
INSERT INTO v_EmployeeDetailWithCheckOption (EmployeeID, NAME)
VALUES( 17, 'Added with View')
```

-->>This will not be inserted

```
INSERT INTO v_EmployeeDetailWithCheckOption (EmployeeID, NAME)
VALUES( 12, 'Added with View')
```

-->>This will be inserted Correctly