# Problems with our database

select top 10 percent \* from tblEmployee

select \* from tblEmployee where EmployeeNumber = 2001

select T.EmployeeNumber as TEmployeeNumber,

E.EmployeeNumber as EEmployeeNumber,

sum(Amount) as SumAmount

from tblTransaction AS T

LEFT JOIN tblEmployee AS E

ON T.EmployeeNumber = E.EmployeeNumber

group by T.EmployeeNumber, E.EmployeeNumber

order by EEmployeeNumber

BEGIN TRAN

UPDATE tblEmployee

SET DateOfBirth = '2101-01-01'

WHERE EmployeeNumber = 537

select \* from tblEmployee ORDER BY DateOfBirth DESC

ROLLBACK TRAN

BEGIN TRAN

UPDATE tblEmployee

SET EmployeeGovernmentID = 'aaaa'

WHERE EmployeeNumber BETWEEN 530 AND 539

select \* from tblEmployee ORDER BY EmployeeGovernmentID ASC

ROLLBACK TRAN

insert into tblEmployee

select NULL, EmployeeFirstName, EmployeeMiddleName, EmployeeLastName, EmployeeGovernmentID, DateOfBirth, Department

from tblEmployee

# What are constraints?

INSERT INTO tblEmployee

values (2001, 'FirstName', 'M', 'LastName', 'AB123456C', '1994-01-01', 'Commerical')

INSERT INTO tblEmployee

values (null, 'AnotherFirstName', 'N', 'AnotherLastName', 'AB123457C', '1994-01-02', 'Finance')

delete from tblEmployee

Where EmployeeNumber > 2000

# Unique constraints in action

alter table tblEmployee

ADD CONSTRAINT unqGovernmentID UNIQUE (EmployeeGovernmentID);

select EmployeeGovernmentID, count(EmployeeGovernmentID) as MyCount from tblEmployee

group by EmployeeGovernmentID

having count(EmployeeGovernmentID)>1

select \* from tblEmployee where EmployeeGovernmentID IN ('HN513777D', 'TX593671R')

begin tran

delete from tblEmployee

where EmployeeNumber < 3

delete top(2) from tblEmployee

where EmployeeNumber in (131, 132)

select \* from tblEmployee where EmployeeGovernmentID IN ('HN513777D', 'TX593671R')

commit tran

alter table tblTransaction

add constraint unqTransaction UNIQUE (Amount, DateOfTransaction, EmployeeNumber)

delete from tblTransaction

where EmployeeNumber = 131

insert into tblTransaction

VALUES (1,'2015-01-01', 131)

insert into tblTransaction

VALUES (1,'2015-01-01', 131)

alter table tblTransaction

Drop constraint unqTransaction

create table tblTransaction2

(Amount smallmoney not null,

DateOfTransaction smalldatetime not null,

EmployeeNumber int not null,

CONSTRAINT unqTransaction2 UNIQUE (Amount,DateOfTransaction,EmployeeNumber))

drop table tblTransaction2

That's right. The values have to be UNIQUE, and if there was more than one NULL, it wouldn't be unique.

# Default constraints in action

alter table tblTransaction

add DateOfEntry datetime

alter table tblTransaction

add constraint defDateOfEntry DEFAULT GETDATE() for DateOfEntry;

delete from tblTransaction where EmployeeNumber < 3

insert into tblTransaction(Amount, DateOfTransaction, EmployeeNumber)

values (1, '2014-01-01', 1)

insert into tblTransaction(Amount, DateOfTransaction, EmployeeNumber, DateOfEntry)

values (2, '2014-01-02', 1, '2013-01-01')

select \* from tblTransaction where EmployeeNumber < 3

create table tblTransaction2

(Amount smallmoney not null,

DateOfTransaction smalldatetime not null,

EmployeeNumber int not null,

DateOfEntry datetime null CONSTRAINT tblTransaction2\_defDateOfEntry DEFAULT GETDATE())

insert into tblTransaction2(Amount, DateOfTransaction, EmployeeNumber)

values (1, '2014-01-01', 1)

insert into tblTransaction2(Amount, DateOfTransaction, EmployeeNumber, DateOfEntry)

values (2, '2014-01-02', 1, '2013-01-01')

select \* from tblTransaction2 where EmployeeNumber < 3

drop table tblTransaction2

alter table tblTransaction

drop column DateOfEntry

alter table tblTransaction

drop constraint defDateOfEntry

BEGIN TRAN

ALTER TABLE tblTransaction

ADD DateOfEntry datetime

DEFAULT GETDATE() WITH VALUES

SELECT \* FROM tblTransaction

ROLLBACK TRAN

# If you don't want the CHECK to affect existing rows, what and where do you add?

# Check constraints in practice

alter table tblTransaction

add constraint chkAmount check (Amount>-1000 and Amount < 1000)

insert into tblTransaction

values (1010, '2014-01-01', 1)

alter table tblEmployee with nocheck

add constraint chkMiddleName check

(REPLACE(EmployeeMiddleName,'.','') = EmployeeMiddleName or EmployeeMiddleName is null)

alter table tblEmployee

drop constraint chkMiddleName

begin tran

insert into tblEmployee

values (2003, 'A', 'B.', 'C', 'D', '2014-01-01', 'Accounts')

select \* from tblEmployee where EmployeeNumber = 2003

rollback tran

alter table tblEmployee with nocheck

add constraint chkDateOfBirth check (DateOfBirth between '1900-01-01' and getdate())

begin tran

insert into tblEmployee

values (2003, 'A', 'B', 'C', 'D', '2115-01-01', 'Accounts')

select \* from tblEmployee where EmployeeNumber = 2003

rollback tran

create table tblEmployee2

(EmployeeMiddleName varchar(50) null, constraint CK\_EmployeeMiddleName check

(REPLACE(EmployeeMiddleName,'.','') = EmployeeMiddleName or EmployeeMiddleName is null))

drop table tblEmployee2

alter table tblEmployee

drop chkDateOfBirth

alter table tblEmployee

drop chkMiddleName

alter table tblTransaction

drop chkAmount

# Primary Key – In Practice

alter table tblEmployee

add constraint PK\_tblEmployee PRIMARY KEY (EmployeeNumber)

insert into tblEmployee(EmployeeNumber, EmployeeFirstName, EmployeeMiddleName, EmployeeLastName,

EmployeeGovernmentID, DateOfBirth, Department)

values (2004, 'FirstName', 'MiddleName', 'LastName', 'AB12345FI', '2014-01-01', 'Accounts')

delete from tblEmployee

where EmployeeNumber = 2004

alter table tblEmployee

drop constraint PK\_tblEmployee

create table tblEmployee2

(EmployeeNumber int CONSTRAINT PK\_tblEmployee2 PRIMARY KEY IDENTITY(1,1),

EmployeeName nvarchar(20))

insert into tblEmployee2

values ('My Name'),

('My Name')

select \* from tblEmployee2

delete from tblEmployee2

truncate table tblEmployee2

insert into tblEmployee2(EmployeeNumber, EmployeeName)

values (3, 'My Name'), (4, 'My Name')

SET IDENTITY\_INSERT tblEmployee2 ON

insert into tblEmployee2(EmployeeNumber, EmployeeName)

values (38, 'My Name'), (39, 'My Name')

SET IDENTITY\_INSERT tblEmployee2 OFF

drop table tblEmployee2

select @@IDENTITY -- last identity used – if one table is 4 then it will be 4

select SCOPE\_IDENTITY() – last identity in scoped

select IDENT\_CURRENT('dbo.tblEmployee2') -- this for specific table

create table tblEmployee3

(EmployeeNumber int CONSTRAINT PK\_tblEmployee3 PRIMARY KEY IDENTITY(1,1),

EmployeeName nvarchar(20))

insert into tblEmployee3

values ('My Name'),

('My Name')

# Foreign key – in practice

BEGIN TRAN

ALTER TABLE tblTransaction ALTER COLUMN EmployeeNumber INT NULL

ALTER TABLE tblTransaction ADD CONSTRAINT DF\_tblTransaction DEFAULT 124 FOR EmployeeNumber

ALTER TABLE tblTransaction WITH NOCHECK

ADD CONSTRAINT FK\_tblTransaction\_EmployeeNumber FOREIGN KEY (EmployeeNumber)

REFERENCES tblEmployee(EmployeeNumber)

ON UPDATE CASCADE

ON DELETE set default

--UPDATE tblEmployee SET EmployeeNumber = 9123 Where EmployeeNumber = 123

DELETE tblEmployee Where EmployeeNumber = 123

SELECT E.EmployeeNumber, T.\*

FROM tblEmployee as E

RIGHT JOIN tblTransaction as T

on E.EmployeeNumber = T.EmployeeNumber

where T.Amount IN (-179.47, 786.22, -967.36, 957.03)

ROLLBACK TRAN

# Creating views

select 1

go

create view ViewByDepartment as

select D.Department, T.EmployeeNumber, T.DateOfTransaction, T.Amount as TotalAmount

from tblDepartment as D

left join tblEmployee as E

on D.Department = E.Department

left join tblTransaction as T

on E.EmployeeNumber = T.EmployeeNumber

where T.EmployeeNumber between 120 and 139

--order by D.Department, T.EmployeeNumber

go

create view ViewSummary as

select D.Department, T.EmployeeNumber as EmpNum, sum(T.Amount) as TotalAmount

from tblDepartment as D

left join tblEmployee as E

on D.Department = E.Department

left join tblTransaction as T

on E.EmployeeNumber = T.EmployeeNumber

group by D.Department, T.EmployeeNumber

--order by D.Department, T.EmployeeNumber

go

select \* from ViewByDepartment

select \* from ViewSummary

# Altering and dropping views

Alter view as

--if exists(select \* from sys.views where name = 'ViewByDepartment')

if exists(select \* from INFORMATION\_SCHEMA.VIEWS

where [TABLE\_NAME] = 'ViewByDepartment' and [TABLE\_SCHEMA] = 'dbo')

drop view dbo.ViewByDepartment

go

CREATE view [dbo].[ViewByDepartment] as

select D.Department, T.EmployeeNumber, T.DateOfTransaction, T.Amount as TotalAmount

from tblDepartment as D

left join tblEmployee as E

on D.Department = E.Department

left join tblTransaction as T

on E.EmployeeNumber = T.EmployeeNumber

where T.EmployeeNumber between 120 and 139

--order by D.Department, T.EmployeeNumber

GO

From SQL Server 2016 Service Pack 1, there is an alternative.

You can use the phrase "CREATE OR ALTER VIEW".

If the view doesn't exist, it is CREATEd.

If the view does exist, it is ALTERed.

However, if you have an earlier version, then you don't have this option.

Create view \*\*\*\*\*\* with encryption as Select ( view will be secure)

# Adding new rows to views

begin tran

insert into ViewByDepartment(EmployeeNumber,DateOfTransaction,TotalAmount)

values (132,'2015-07-07', 999.99)

select \* from ViewByDepartment order by Department, EmployeeNumber

rollback tran

begin tran

select \* from ViewByDepartment order by EmployeeNumber, DateOfTransaction

--Select \* from tblTransaction where EmployeeNumber in (132,142)

update ViewByDepartment

set EmployeeNumber = 142

where EmployeeNumber = 132

select \* from ViewByDepartment order by EmployeeNumber, DateOfTransaction

--Select \* from tblTransaction where EmployeeNumber in (132,142)

rollback tran

USE [70-461]

GO

--if exists(select \* from sys.views where name = 'ViewByDepartment')

if exists(select \* from INFORMATION\_SCHEMA.VIEWS

where [TABLE\_NAME] = 'ViewByDepartment' and [TABLE\_SCHEMA] = 'dbo')

drop view dbo.ViewByDepartment

go

CREATE view [dbo].[ViewByDepartment] as

select D.Department, T.EmployeeNumber, T.DateOfTransaction, T.Amount as TotalAmount

from tblDepartment as D

left join tblEmployee as E

on D.Department = E.Department

left join tblTransaction as T

on E.EmployeeNumber = T.EmployeeNumber

where T.EmployeeNumber between 120 and 139

WITH CHECK OPTION

--order by D.Department, T.EmployeeNumber

GO

# Deleting rows in views

SELECT \* FROM ViewByDepartment

delete from ViewByDepartment

where TotalAmount = 999.99 and EmployeeNumber = 132

GO

CREATE VIEW ViewSimple

as

SELECT \* FROM tblTransaction

GO

BEGIN TRAN

delete from ViewSimple

where EmployeeNumber = 132

select \* from ViewSimple

ROLLBACK TRAN

Because you are using JOINs, you will get the error message "View or function 'dbo.ViewByEmployee' is not updatable because the modification affects multiple base tables". This would go for INSERT statements as well.

# Creating an indexed view

USE [70-461]

GO

--if exists(select \* from sys.views where name = 'ViewByDepartment')

if exists(select \* from INFORMATION\_SCHEMA.VIEWS

where [TABLE\_NAME] = 'ViewByDepartment' and [TABLE\_SCHEMA] = 'dbo')

drop view dbo.ViewByDepartment

go

CREATE view [dbo].[ViewByDepartment] with schemabinding as

select D.Department, T.EmployeeNumber, T.DateOfTransaction, T.Amount as TotalAmount

from dbo.tblDepartment as D

inner join dbo.tblEmployee as E

on D.Department = E.Department

inner join dbo.tblTransaction as T

on E.EmployeeNumber = T.EmployeeNumber

where T.EmployeeNumber between 120 and 139

GO

CREATE UNIQUE CLUSTERED INDEX inx\_ViewByDepartment on dbo.ViewByDepartment(EmployeeNumber, Department)

begin tran

drop table tblEmployee

rollback tran

## SEEK and SCAN is how SQL Server uses an indexed view

Insert into

Insert => after trigger is called (for is same

Not inserted => instead of trigger is called

Create trigger

On table

and then you use AFTER/INSTEAD OF, and then INSERT/DELETE/UPDATE.

# Creating an AFTER trigger

CREATE TRIGGER TR\_tblTransaction

ON tblTransaction

AFTER DELETE, INSERT, UPDATE

AS

BEGIN

--insert into tblTransaction2

select \*, 'Inserted' from Inserted

--insert into tblTransaction2

select \*, 'Deleted' from Deleted

END

GO

BEGIN TRAN

insert into tblTransaction(Amount, DateOfTransaction, EmployeeNumber)

VALUES (123,'2015-07-10', 123)

--delete tblTransaction

--where EmployeeNumber = 123 and DateOfTransaction = '2015-07-10'

ROLLBACK TRAN

GO

DISABLE TRIGGER TR\_tblTransaction ON tblTransaction;

GO

ENABLE TRIGGER TR\_tblTransaction ON tblTransaction;

GO

DROP TRIGGER TR\_tblTransaction;

GO

# Creating an INSTEAD OF trigger

alter TRIGGER tr\_ViewByDepartment

ON dbo.ViewByDepartment

INSTEAD OF DELETE

AS

BEGIN

declare @EmployeeNumber as int

declare @DateOfTransaction as smalldatetime

declare @Amount as smallmoney

select @EmployeeNumber = EmployeeNumber, @DateOfTransaction = DateOfTransaction, @Amount = TotalAmount

from deleted

--SELECT \* FROM deleted

delete tblTransaction

from tblTransaction as T

where T.EmployeeNumber = @EmployeeNumber

and T.DateOfTransaction = @DateOfTransaction

and T.Amount = @Amount

END

begin tran

--SELECT \* FROM ViewByDepartment where TotalAmount = -2.77 and EmployeeNumber = 132

delete from ViewByDepartment

where TotalAmount = -2.77 and EmployeeNumber = 132

SELECT \* FROM ViewByDepartment where TotalAmount = -2.77 and EmployeeNumber = 132

rollback tran

# Nested triggers

ALTER TRIGGER TR\_tblTransaction

ON tblTransaction

AFTER DELETE, INSERT, UPDATE

AS

BEGIN

if @@NESTLEVEL = 1

begin

select \*,'TABLEINSERT' from Inserted

select \*, 'TABLEDELETE' from Deleted

end

END

GO

BEGIN TRAN

insert into tblTransaction(Amount, DateOfTransaction, EmployeeNumber)

VALUES (123,'2015-07-10', 123)

ROLLBACK TRAN

begin tran

--SELECT \* FROM ViewByDepartment where TotalAmount = -2.77 and EmployeeNumber = 132

delete from ViewByDepartment

where TotalAmount = -2.77 and EmployeeNumber = 132

--SELECT \* FROM ViewByDepartment where TotalAmount = -2.77 and EmployeeNumber = 132

rollback tran

EXEC sp\_configure 'nested triggers';

EXEC sp\_configure 'nested triggers',0;

RECONFIGURE

GO

# Update functions

ALTER TRIGGER TR\_tblTransaction

ON tblTransaction

AFTER DELETE, INSERT, UPDATE

AS

BEGIN

IF @@ROWCOUNT > 0

BEGIN

select \* from Inserted

select \* from Deleted

END

END

GO

insert into tblTransaction(Amount, DateOfTransaction, EmployeeNumber)

VALUES (123,'2015-07-11', 123)

SELECT \* FROM ViewByDepartment where TotalAmount = -2.77 and EmployeeNumber = 132

begin tran

delete from ViewByDepartment

where TotalAmount = -2.77 and EmployeeNumber = 132

rollback tran

ALTER TRIGGER TR\_tblTransaction

ON tblTransaction

AFTER DELETE, INSERT, UPDATE

AS

BEGIN

--SELECT COLUMNS\_UPDATED()

IF UPDATE(Amount) -- if (COLUMNS\_UPDATED() & POWER(2,1-1)) > 0

BEGIN

select \* from Inserted

select \* from Deleted

END

END

go

begin tran

--SELECT \* FROM ViewByDepartment where TotalAmount = -2.77 and EmployeeNumber = 132

update ViewByDepartment

set TotalAmount = +2.77

where TotalAmount = -2.77 and EmployeeNumber = 132

rollback tran

Look at this code:

SELECT \* FROM [dbo].[tblEmployee] Where EmployeeFirstName like 'Pe%'

This returns 9 rows, of which 4 are 'Peter'.

Now look at this code:

1. UPDATE [dbo].[tblEmployee]
2. SET EmployeeFirstName = 'Peter' Where EmployeeFirstName like 'Pe%'
3. SELECT @@ROWCOUNT

What will the value of @@ROWCOUNT be?

That's right. The UPDATE statement updates all 9 rows, including those where the EmployeeFirstName is already 'Peter'. To reduce it to 5, then you need to add AND (EmployeeFirstName <> 'Peter' OR SELECT \* FROM [dbo].[tblEmployee] Where EmployeeFirstName like 'Pe%' SELECT @@ROWCOUNT UPDATE [dbo].[tblEmployee] SET EmployeeFirstName = 'Peter' Where EmployeeFirstName like 'Pe%' AND (EmployeeFirstName <> 'Peter' or EmployeeFirstName IS NULL)

**Maxmimum @@NestLevel=32**

# Handling multiple rows in a session

-- Bad code - only allows 1 row to be deleted

alter TRIGGER tr\_ViewByDepartment

ON dbo.ViewByDepartment

INSTEAD OF DELETE

AS

BEGIN

declare @EmployeeNumber as int

declare @DateOfTransaction as smalldatetime

declare @Amount as smallmoney

select @EmployeeNumber = EmployeeNumber, @DateOfTransaction = DateOfTransaction, @Amount = TotalAmount

from deleted

--SELECT \* FROM deleted

delete tblTransaction

from tblTransaction as T

where T.EmployeeNumber = @EmployeeNumber

and T.DateOfTransaction = @DateOfTransaction

and T.Amount = @Amount

END

begin tran

SELECT \* FROM ViewByDepartment where EmployeeNumber = 132

delete from ViewByDepartment

where EmployeeNumber = 132

SELECT \* FROM ViewByDepartment where EmployeeNumber = 132

rollback tran

-- Good code - allows multiple rows to be deleted

alter TRIGGER tr\_ViewByDepartment

ON dbo.ViewByDepartment

INSTEAD OF DELETE

AS

BEGIN

SELECT \*, 'To Be Deleted' FROM deleted

delete tblTransaction

from tblTransaction as T

join deleted as D

on T.EmployeeNumber = D.EmployeeNumber

and T.DateOfTransaction = D.DateOfTransaction

and T.Amount = D.TotalAmount

END

GO

begin tran

SELECT \*, 'Before Delete' FROM ViewByDepartment where EmployeeNumber = 132

delete from ViewByDepartment

where EmployeeNumber = 132 --and TotalAmount = 861.16

SELECT \*, 'After Delete' FROM ViewByDepartment where EmployeeNumber = 132

rollback tran