We creëren een vereenvoudigde tabel waarin we rekeningsaldo's voor personen gaan bijhouden.

□ We voegen een constraint toe: het saldo mag niet kleiner dan 0 zijn.

ALTER TABLE Saldos ADD CHECK (Saldo>=0)

- □ We voegen 2 rijen in de tabel toe:
  - Kevin met een startsaldo van 800€

```
insert into saldos(voornaam, saldo)
values ('Kevin', 800)
```

■ Mia met een startsaldo van 1300€

```
insert into saldos(voornaam, saldo)
values ('Mia', 1300)
```

#### □ Saldostatus na INSERT:

	Na Insert		
Kevin	800€		
Mia	1300€		

We doen een 1<sup>STE</sup> overschrijving: 300€ van Kevin naar Mia

```
update saldos
set saldo = saldo - 300
where Voornaam = 'Kevin'
  (1 row affected)
update saldos
set saldo = saldo + 300
where Voornaam = 'Mia'
  (1 row affected)
```

## □ Saldostatus:

	Na Insert	Na 1 <sup>ste</sup> overschrijving	
Kevin	800€	500€	
Mia	1300€	1600€	

We doen een 2<sup>de</sup> overschrijving: 300€ van Kevin naar Mia

```
update saldos
set saldo = saldo - 300
where Voornaam = 'Kevin'
   (1 row affected)

update saldos
set saldo = saldo + 300
where Voornaam = 'Mia'
   (1 row affected)
```

#### □ Saldostatus:

	Na Insert	Na 1 <sup>ste</sup> overschrijving	Na 2 <sup>de</sup> overschrijving	
Kevin	800€	500€	200€	
Mia	1300€	1600€	1900€	

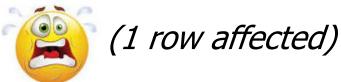
We doen een 3de overschrijving: 300€ van Kevin naar Mia

```
update saldos
set saldo = saldo - 300
where Voornaam = 'Kevin'
```



The UPDATE statement conflicted with the CHECK constraint The statement has been terminated.

```
update saldos
set saldo = saldo + 300
where Voornaam = 'Mia'
```



#### □ Saldostatus:

	Na Insert	Na 1 <sup>ste</sup> overschrijving	Na 2 <sup>de</sup> overschrijving	Na 3 <sup>de</sup> overschrijving
Kevin	800€	500€	200€	200€
Mia	1300€	1600€	1900€	2200€

#### Gebruik van transactie

```
SET XACT_ABORT ON
-- the batch is aborted at the first error, the whole unit of work
is rolled back

BEGIN TRANSACTION
   update saldos set Saldo = saldo - 300 where Voornaam = 'Kevin'
   update saldos set Saldo = saldo + 300 where Voornaam = 'Mia'
COMMIT TRANSACTION
```

#### Gebruik van stored procedure

```
CREATE OR ALTER PROCEDURE MakeBankTransfer
    @FromId int,
    @ToId int,
    @Amount int
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;
    SET XACT ABORT ON -- the batch is aborted at the first error, the whole
    unit of work is rolled back
    BEGIN TRANSACTION
      update saldos set Saldo = saldo - @Amount where id = @FromId
      update saldos set Saldo = saldo + @Amount where id = @ToId
    COMMIT TRANSACTION
END
```

Oproepen van stored procedure

exec dbo.MakeBankTransfer 1,2, 100

■ Wat gebeurt er bij deze?

exec dbo.MakeBankTransfer 10,2, 100

Deze id bestaat niet

#### Verbeterde stored procedure

```
CREATE OR ALTER PROCEDURE MakeBankTransfer
    @FromId int,
    @ToId int,
    @Amount int
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;
SET XACT ABORT ON -- the batch is aborted at the first error, the whole unit of work is rolled bac
BEGIN TRANSACTION
 declare @fromIdRowsFound int
 declare @toIdRowsFound int
  select @fromIdRowsFound = count(1) from saldos where id = @FromId
  select @toIdRowsFound = count(1) from saldos where id = @ToId
 if (@fromIdRowsFound = 1 and @toIdRowsFound = 1)
 begin
      update saldos set Saldo = saldo - @Amount where id = @FromId
      update saldos set Saldo = saldo + @Amount where id = @ToId
  end
COMMIT TRANSACTION
END
```