

Transactions

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

```
CREATE TABLE Saldos
(
  Id          int IDENTITY(1,1)      PRIMARY KEY,
  Voornaam    varchar(200)           NULL,
  Saldo       int                    NULL
)
```

Transactions



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

```
ALTER TABLE Saldos ADD CHECK (Saldo >= 0)
```

Transactions



□ 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)
```

Transactions

- Saldostatus na INSERT:

	Na Insert			
Kevin	800€			
Mia	1300€			

Transactions

- We doen een 1^{STE} 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)

Transactions

□ Saldostatus:

	Na Insert	Na 1 ^{ste} overschrijving		
Kevin	800€	500€		
Mia	1300€	1600€		

Transactions

- We doen een 2^{de} 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)

Transactions

□ Saldostatus:

	Na Insert	Na 1 ^{ste} overschrijving	Na 2 ^{de} overschrijving	
Kevin	800€	500€	200€	
Mia	1300€	1600€	1900€	

Transactions

- We doen een 3^{de} 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'
```



(1 row affected)

Transactions

□ Saldostatus:

	Na Insert	Na 1 ^{ste} overschrijving	Na 2 ^{de} overschrijving	Na 3 ^{de} overschrijving
Kevin	800€	500€	200€	200€
Mia	1300€	1600€	1900€	2200€

Transactions

□ 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
```

Transactions

□ 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
```

Transactions

- 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

Transactions

□ 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 back

    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
```