



Syllabus

Databases

Mileto Di Marco
Kristien Roels
Bart Soete

Inhoudsopgave

H11 - Triggers.....	3
1 Inleiding.....	3
2 De basis van DML-triggers.....	3
3 Wat kan gebruikt worden in de DML-trigger?	5

H11 - Triggers

1 Inleiding

SQL Server kent DML-triggers en DDL-triggers.

(DML = Data Manipulation Language; DDL = Data Definition Language)

- DML-triggers worden automatisch uitgevoerd als reactie op een DML-statement INSERT, UPDATE of DELETE.
- DDL-triggers worden automatisch uitgevoerd als reactie op een DDL-statement CREATE, ALTER of DROP.

In deze cursus beperken we ons tot de DML-triggers (of kortweg triggers).

Relevante Help-pagina's:

<https://docs.microsoft.com/nl-nl/sql/relational-databases/triggers/dml-triggers>

<https://docs.microsoft.com/en-us/sql/t-sql/statements/create-trigger-transact-sql>

De voorbeelddatabank die in dit hoofdstuk gebruikt wordt is de GRADSchool-databank.

2 De basis van DML-triggers

DML-triggers (of kortweg triggers) worden geactiveerd door een DML-statement INSERT, UPDATE of DELETE en kunnen niet rechtstreeks uitgevoerd worden door een gebruiker.

Triggers worden vooral gebruikt voor het afdwingen van data integriteit. Data integriteit betekent dat de data correct is, voldoet aan onze voorwaarden.

Om data integriteit af te dwingen kent SQL Server 2 mechanismen: constraints en triggers.

- Constraints zijn reeds gekend als middel om de data integriteit af te dwingen: (NOT) NULL, Primary Key, Foreign Key, Unique en Check.

De Check-constraint wordt gebruikt om een eenvoudige voorwaarde (business rule) af te dwingen. Een eenvoudige voorwaarde is een voorwaarde waarbij enkel de data uit de huidige rij betrokken is.

Voorbeeld (databank GRADSchool): de kolom Betaald moet een waarde hebben tussen 0 en 1000.

- Triggers worden gebruikt om complexere voorwaarden (business rules) af te dwingen. Een complexe voorwaarde is een voorwaarde waarbij andere data dan de huidige rij betrokken wordt.

Voorbeelden (databank GRADSchool):

- er mogen geen nieuwe cursussen meer toegevoegd worden in de databank
- er mogen zich enkel nog vrouwelijke studenten inschrijven

SQL Server kent 2 soorten triggers:

- after triggers
- instead of triggers

Een after trigger wordt uitgevoerd ná dat het DML-statement, dat de trigger heeft geactiveerd, is afgehandeld, maar vóór dat het statement committed is. Een after trigger kan het DML-statement ongedaan maken (ROLLBACK).

```
CREATE [OR ALTER] TRIGGER triggernaam
ON tabelnaam
AFTER { [ INSERT ] [ , ] [ UPDATE ] [ , ] [ DELETE ] }
AS
SQL-statement [...] – triggercode
```

Een instead of trigger wordt uitgevoerd in de plaats van het DML-statement dat de trigger heeft geactiveerd.

```
CREATE [OR ALTER] TRIGGER triggernaam
ON tabelnaam
INSTEAD OF { [ INSERT ] [ , ] [ UPDATE ] [ , ] [ DELETE ] }
AS
SQL-statement [...] – triggercode
```

De triggers zijn terug te vinden onder de tabelnode, Triggers. Rechtsklik op de gewenste trigger en kies Modify.

Maak een trigger demo1. Telkens als een update in de tabel studenten wordt uitgevoerd, moet een melding getoond worden..
create or alter trigger demo1 on studenten after update as print 'In de after trigger demo1'
Testen of de trigger werkt.
update studenten set boete = boete + 5 where studnr = 1
Verwijder de trigger.
drop trigger if exists demo1

Maak een trigger geenNieuweCursussen. Deze trigger belet dat er nieuwe cursussen worden ingegeven.
create or alter trigger geenNieuweCursussen on cursussen instead of insert as print 'Er worden geen nieuwe cursussen aanvaard'
Testen of de trigger werkt.
insert into cursussen(cursusnaam) values ('.NET Technologie')
Verwijder de trigger.
drop trigger if exists geenNieuweCursussen

3 Wat kan gebruikt worden in de DML-trigger?

Het is belangrijk dat de trigger zicht heeft op de wijzigingen aangebracht door het DML-statement, zodat de trigger de wijzigingen kan evalueren en eventueel actie kan ondernemen.

Een goed begrip van de transactiestroom is essentieel: de trigger en het statement dat de trigger lanceerde, worden behandeld als één transactie.

De gedetailleerde transactiestroom van een I/U/D:

1. Identity insert check
2. Nullability constraint
3. Data-type check
4. **Instead of trigger** wordt uitgevoerd. De uitvoer van het DML-statement stopt hier.
5. Primary key constraint
6. Check constraints
7. Foreign key constraints
8. DML-statement wordt uitgevoerd
9. **After trigger** wordt uitgevoerd
10. Commit transaction

De after trigger:

- wordt uitgevoerd ná de I/U/D, maar vóór dat de I/U/D bevestigd (committed) is
- heeft zicht op de wijzigingen die aangebracht zijn door de I/U/D,
- kan de wijzigingen evalueren a.d.h.v.
 - o functie update(kolom)
 - o speciale tabellen INSERTED en DELETED
- en kan eventueel actie ondernemen: bijvoorbeeld de I/U/D ongedaan maken (rollback)

De UPDATE(kolom)-functie geeft aan of de vermelde kolom beïnvloed is door de DML-transactie.

Maak een trigger demo3 die een boodschap toont als de kolom Boete wordt gewijzigd.
<pre>create or alter trigger demo3 on studenten after update, insert as if update(boete) print 'De boete wordt gewijzigd' else print 'De boete blijft ongewijzigd'</pre>
Testen of de trigger werkt.
<pre>update studenten set boete = 0 where studnr = 1 update studenten set familienaam = 'Dekorte' where studnr = 1 insert into studenten (familienaam, geslacht)</pre>

values ('Soenens', 'V') --insert beïnvloedt alle kolommen!
Verwijder de trigger.
drop trigger if exists demo3

De tabellen INSERTED en DELETED zijn 2 speciale tabellen die kunnen gebruikt worden in DML-triggers. SQL Server maakt en onderhoudt deze tijdelijke tabellen automatisch.

- De tabel DELETED bewaart een kopie van de rijen die betrokken zijn bij een DELETE en een UPDATE. Tijdens de uitvoer van een DELETE of een UPDATE, worden rijen verwijderd uit de triggertabel en geplaatst in de tabel DELETED.
- De tabel INSERTED bewaart een kopie van de rijen die betrokken zijn bij een INSERT en een UPDATE. Tijdens de uitvoer van een INSERT of een UPDATE, worden de nieuwe rijen zowel in de triggertabel als in de tabel INSERTED toegevoegd.
- Een UPDATE-transactie is een delete, gevolgd door een insert.
 - o De oude rijen worden eerst gekopieerd naar de tabel DELETED
 - o en daarna worden de nieuwe rijen gekopieerd in de triggertabel en in de tabel INSERTED.

Maak een trigger demo4 die bij een update van studenten de oude data toont en daarnaast de nieuwe data.
<pre>create or alter trigger demo4 on studenten after update as select deleted.*, inserted.* from deleted join inserted on deleted.studnr = inserted.studnr</pre>
<pre>update studenten set familienaam = 'Van Male' where familienaam = 'Van Maele'</pre> <pre>update studenten set voornaam = 'Piet' where studnr = 1</pre>
drop trigger if exists demo4

Maak een trigger alleenVrouwen die ervoor zorgt dat enkel vrouwelijke studenten zich kunnen inschrijven.
<pre>create or alter trigger alleenVrouwen on studenten_cursussen after insert as if exists (select * from inserted join studenten on inserted.studnr = studenten.studnr where geslacht = 'M') begin print 'Toevoegen gaat niet door: alleen vrouwen' rollback transaction end</pre>

insert into studenten_cursussen(studnr, cursusnr) values (1, 5) insert into studenten_cursussen(studnr, cursusnr) values (2, 5)
drop trigger if exists alleenVrouwen