



Syllabus

Databases

Mileto Di Marco
Kristien Roels
Bart Soete

Inhoudsopgave

H5 - Het toevoegen, wijzigen en verwijderen van data m.b.v. SQL.....	3
1 Inleiding.....	3
2 Data toevoegen met het INSERT-statement.....	3
3 Data wijzigen met het UPDATE-statement	5
4 Data verwijderen met het DELETE-statement	6

H5 - Het toevoegen, wijzigen en verwijderen van data m.b.v. SQL

1 Inleiding

In het vorige hoofdstuk werd het raadplegen van data uit een relationele databank m.b.v. het SELECT-statement behandeld. In dit hoofdstuk wordt het toevoegen, wijzigen en verwijderen van data behandeld. Daartoe worden de SQL-statements INSERT (toevoegen), UPDATE (wijzigen) en DELETE (verwijderen) besproken.

Deze 4 SQL-statements SELECT, INSERT, UPDATE en DELETE vormen samen de zgn. SQL DML.

DML staat voor Data Manipulation Language en bevat de SQL-statements voor het opvragen en up-to-date houden van de data.

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

Vergeet dus niet om van de bibliotheek-databank de actieve databank te maken vooraleer SQL-statements uit te voeren: USE bibliotheek.

Voor een volledige bespreking van het INSERT-, UPDATE en DELETE-statement, zie respectievelijk:

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

<https://docs.microsoft.com/en-us/sql/t-sql/queries/update-transact-sql>

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

2 Data toevoegen met het INSERT-statement

Het INSERT-statement kan op verschillende manieren gebruikt worden om gegevens toe te voegen aan een tabel.

Het INSERT-statement met de VALUES-clausule is de meest voorkomende vorm. Hierbij worden 1 of meerdere rijen toegevoegd aan de tabel. Deze vorm wordt gebruikt om data toe te voegen die afkomstig is van een userinterface. De syntax:

INSERT [INTO] tabel [(kolommenlijst)]

VALUES (waardenlijst) [, ...]

Dit INSERT-statement voegt 1 of meerdere rijen toe aan de tabel die vermeld wordt. Het keyword INTO is facultatief maar wordt meestal wel geschreven. Er wordt 1 rij toegevoegd als er 1 waardenlijst vermeld wordt; er worden meerdere rijen toegevoegd als er meerdere waardenlijsten vermeld worden.

De kolommenlijst bevat een opsomming van de kolommen (gescheiden door een komma) die een waarde zullen krijgen in de VALUES-clausule. Deze kolommenlijst is facultatief, maar het wordt zeer sterk aangeraden steeds de kolommenlijst expliciet te schrijven. Als de kolommenlijst niet vermeld wordt, dan moeten alle kolommen een waarde krijgen en de volgorde van de waarden moet gelijk zijn aan de volgorde van de kolommen in de tabel.

De waardenlijst bevat een opsomming van de waarden (gescheiden door een komma) die als een rij zullen toegevoegd worden aan de tabel. Zorg ervoor dat het type van deze waarden overeenstemt met het type van de corresponderende kolommen.

Het INSERT-statement met de SELECT-clausule kan ook gebruikt worden. Hierbij wordt de resultaatset van het SELECT-statement toegevoegd aan de tabel. De syntax:

INSERT [INTO] tabel [(kolommenlijst)]
SELECT-statement

Dit INSERT-statement voegt een resultaatset toe aan de tabel die vermeld wordt. Het keyword INTO is facultatief maar wordt meestal wel geschreven.

De kolommenlijst bevat een opsomming van de kolommen (gescheiden door een komma) die een waarde zullen krijgen. Deze kolommenlijst is facultatief, maar het wordt zeer sterk aangeraden steeds de kolommenlijst expliciet te schrijven.

Het SELECT-statement levert een resultaatset. De rijen uit deze resultaatset worden toegevoegd aan de tabel. Zorg ervoor dat het type van deze waarden uit de SELECT-clausule overeenstemmen met het type van de corresponderende kolommen.

Het is belangrijk om op te merken dat bij elke INSERT de constraints (integriteitsvoorwaarden) geverifieerd worden! Indien aan de voorwaarden voldaan is, dan wordt de INSERT uitgevoerd. In het andere geval komt er een foutmelding en gebeurt de INSERT niet!

Voeg 2 uitgevers toe: 20 Mc Graw Hill 21 Van Duuren Media
insert into uitgevers(uitgeverid, uitgever) values (20, 'Mc Graw Hill') insert into uitgevers(uitgeverid, uitgever) values (21, 'Van Duuren Media')
insert into uitgevers(uitgeverid, uitgever) values (20, 'Mc Graw Hill'), (21, 'Van Duuren Media')
De onderstaande toevoeging is FOUT omdat er reeds een uitgever is met uitgeverid 1. Uitgeverid is de primaire sleutel in de tabel uitgevers en moet dus unieke waarden bevatten. insert into uitgevers(uitgeverid, uitgever) values (1, 'Mc Graw Hill')
Opmerking: In hoofdstuk 7 zien we hoe de maker van de tabellen ervoor kan zorgen dat de id's (uitgeverid, boekid, auteurid, ...) automatisch opgevuld worden met opeenvolgende waarden. In de bibliotheek-databank moeten de id's echter expliciet ingegeven worden.

Voeg een uitlening toe: 50 - boekid 2 - datum 20/08/2017 - klantid 1
insert into uitleningen(uitleenid, boekid, datum, klantid) values (50, 2, '2017-08-20', 1)
De onderstaande toevoeging is FOUT omdat er geen klant is met klantid 16. Klantid is een vreemde sleutel in de uitleningen-tabel die refereert naar klantid in de klanten-tabel. Dit betekent dat klantid in de uitleningen-tabel een waarde moet hebben die voorkomt in de klanten-tabel (of NULL is). insert into uitleningen(uitleenid, boekid, datum, klantid) values (51, 2, '2017-08-20', 16)

Voeg een klant toe: 30 De Lange Alexander
insert into klanten(klantid, familienaam, voornaam) values (30, 'De Lange', 'Alexander')
insert into klanten(klantid, familienaam, voornaam, woonplaats) values (30, 'De Lange', 'Alexander', NULL)

Voeg een auteur toe: 30 Petkovic Dusan
Voeg een boek toe: 80 SQL A Beginners Guide - auteur 30 (Petkovic Dusan) – categorie 3 (wetenschappelijk) – uitgever 20 (Mc Graw Hill)
insert into auteurs(auteurid, familienaam, voornaam) values (30, 'Petkovic', 'Dusan')
insert into boeken(boekid, auteurid, titel, categorieid, uitgeverid) values(80, 30, 'SQL A Beginners Guide', 3, 20)

Voeg de boeken uit het jaar 1999 en 2000 toe aan de tabel Boeken3 (zie H4 – 12 SELECT INTO).
insert into boeken3(boekid, auteurid, jaar, titel, categorieid, uitgeverid) select boekid, auteurid, jaar, titel, categorieid, uitgeverid from boeken where jaar IN (1999, 2000)

3 Data wijzigen met het UPDATE-statement

Het UPDATE-statement wordt gebruikt voor het wijzigen van data. De syntax:

UPDATE tabel

SET kolomnaam = { expressie | NULL } [, ...]

[FROM-clausule]

[WHERE-clausule]

Dit UPDATE-statement brengt wijzigingen aan, aan de data van de tabel die vermeld wordt.

De SET-clausule geeft aan welke kolom(men) welke nieuwe waarde moeten krijgen.

De WHERE-clausule geeft aan in welke rijen de wijzigingen moeten gebeuren. Als er geen WHERE-clausule wordt opgenomen, dan worden de wijzigingen in alle rijen doorgevoerd.

De FROM-clausule wordt slechts opgenomen als er andere tabellen nodig zijn om de criteria op te stellen of om de nieuwe waarde te bepalen.

Het is belangrijk om op te merken dat bij elke UPDATE de constraints (integriteitsvoorwaarden) geverifieerd worden! Indien aan de voorwaarden voldaan is, dan wordt de UPDATE uitgevoerd. In het andere geval komt er een foutmelding en gebeurt de UPDATE niet!

Wijzig de woonplaats van de klant 10 naar De Haan.
<pre>update klanten set woonplaats = 'De Haan' where klantid = 10</pre>

Verhoog in de uitleningen-tabel de datum met 5 dagen voor alle uitleningen van klant 3.
<pre>update uitleningen set datum = DATEADD(day, 5, datum) where klantid = 3</pre>

Wijzig de familienaam van klant 10 in Arnoldus en de woonplaats in Nieuwpoort.
<pre>update klanten set familienaam = 'Arnoldus', woonplaats = 'Nieuwpoort' where klantid = 10</pre>

Verhoog in de boeken-tabel het jaar met 1 voor de boeken uit de categorie wetenschappelijk.
<pre>update boeken set jaar = jaar + 1 where categorieid = (select categorieid from categorieen where categorie = 'wetenschappelijk')</pre>
<pre>update boeken set jaar = jaar + 1 from boeken join categorieen on boeken.categorieid = categorieen.categorieid where categorieen.categorie = 'wetenschappelijk'</pre>

4 Data verwijderen met het DELETE-statement

Het DELETE-statement wordt gebruikt voor het verwijderen van data. De syntax:

```
DELETE [FROM] tabel
[ FROM-clausule ]
[ WHERE-clausule ]
```

Dit DELETE-statement verwijdert rijen uit de tabel die vermeld wordt. Het keyword FROM is facultatief.

De WHERE-clausule geeft aan welke rijen moeten verwijderd worden. Als er geen WHERE-clausule wordt opgenomen, dan worden alle rijen verwijderd! Er wordt geen bevestiging gevraagd en de rijen worden fysiek verwijderd.

De FROM-clausule wordt slechts opgenomen als er andere tabellen nodig zijn om de criteria op te stellen.

Het is belangrijk om op te merken dat bij elke DELETE de constraints (integriteitsvoorwaarden) geverifieerd worden! Indien aan de voorwaarden voldaan is, dan wordt de DELETE uitgevoerd. In het andere geval komt er een foutmelding en gebeurt de DELETE niet!

Verwijder het boek met boekid 10.

Zal dit steeds lukken?

```
delete from boeken
```

```
where boekid = 10
```

Dit zal maar lukken als er geen uitleningen zijn van dit boek. Vanuit de uitleningen-tabel wordt er namelijk gerefereerd naar de boeken-tabel.

Verwijder de auteurs die geen boeken schreven.

```
delete from auteurs
```

```
where auteurid not in (select auteurid  
                        from boeken)
```

Verwijder de auteurs die een boek schreven.

De onderstaande verwijdering geeft een FOUT, kan niet, omdat er vanuit de boeken-tabel gerefereerd wordt naar de auteurs-tabel.

```
delete from auteurs
```

```
where auteurid in (select auteurid  
                   from boeken)
```

Moeten deze auteurs toch verwijderd worden, dan moeten eerst hun boeken verwijderd worden.

Verwijder alle data uit alle tabellen.

```
delete from uitleningen
```

```
delete from klanten
```

```
delete from boeken
```

```
delete from categorieen
```

```
delete from auteurs
```

```
delete from uitgevers
```

Merk op dat de volgorde van de DELETE-statements belangrijk is.