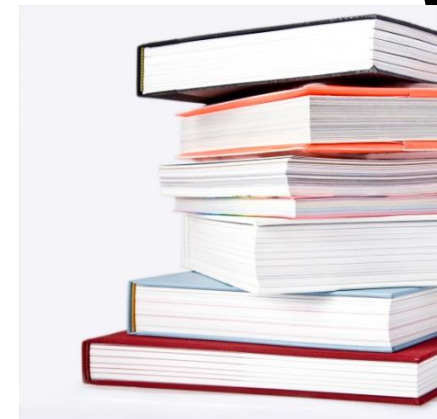


DML

Data manipulation language

cursusmateriaal

- Cursus 'Databanken 1' blz. 51-52
- Deze powerpoint
- Extra's
- Hoofdstuk 10 handboek 'SQL Fundamentals I Exam Guide' blz. 403-437
- <http://docs.oracle.com/database/121/SQLRF/queries002.htm#SQLRF52331>
- http://docs.oracle.com/database/121/SQLRF/statements_9014.htm#SQLRF01604



Overzicht instructies

- Data Definition Language (DDL)
 - Objecten in databank creëren, verwijderen wijzigen
 - instructies: CREATE, DROP, ALTER, TRUNCATE,...
- Data manipulation language (DML)
 - Gegevens in databank manipuleren
 - instructies: select, insert, delete, update
- Data control language (DCL)
 - instructies: commit, rollback, grant, connect

DML

Dit zijn de SQL instructies die gebruikt worden om

- nieuwe rijen toe te voegen aan een tabel → **INSERT**
- de inhoud van bestaande tabellen te wijzigen → **UPDATE**
- rijen uit de tabel te verwijderen → **DELETE**

Bij het geven van DML instructies horen ook 2 transaction management instructies:

- **COMMIT** om een wijziging definitief te maken
- **ROLLBACK** om wijzigingen tot aan de laatst gegeven commit opnieuw ongedaan te maken

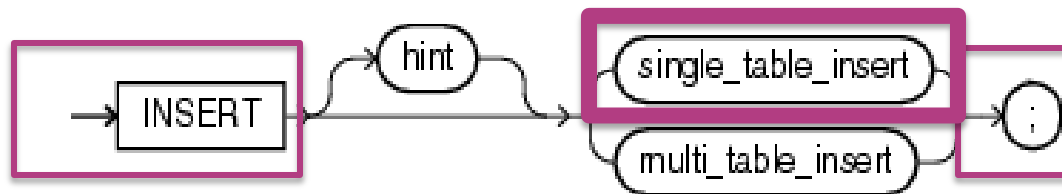
Om alle rijen definitief uit een tabel verwijderen
→ **TRUNCATE**

Dit is een DDL instructie. Ze kan niet gerolbackt worden.

SQL Reference

SQL
Reference
p. 152

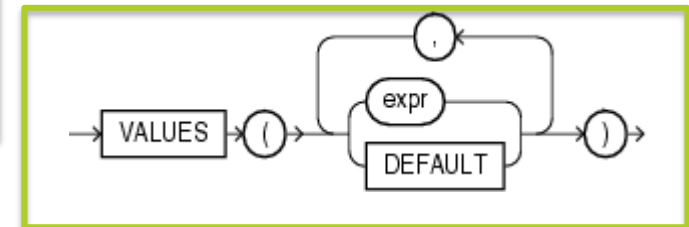
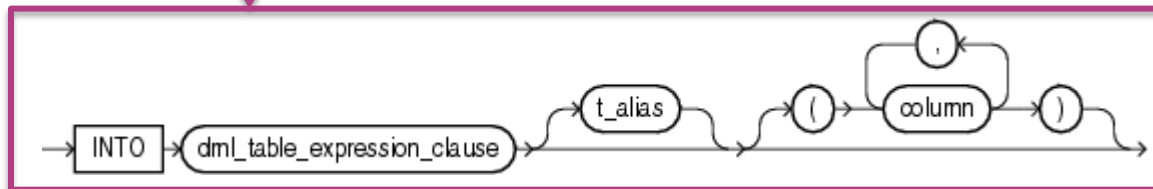
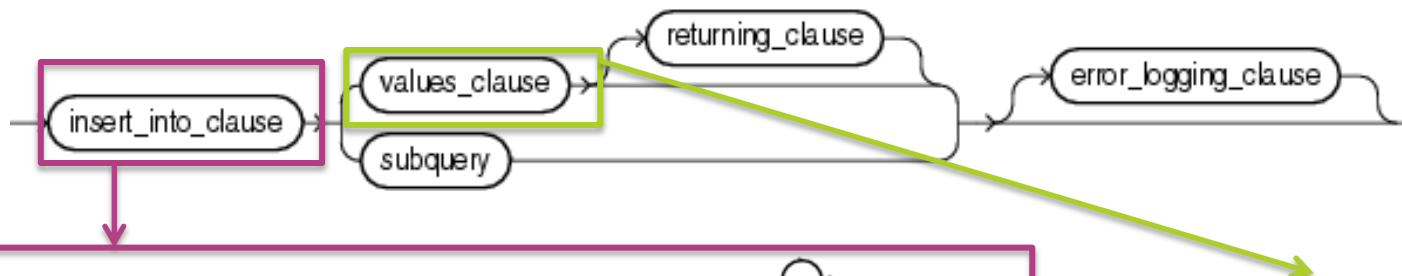
Niet vergeten om telkens met fluo aan te duiden!!



SQL Reference

single_table_insert

- Een 'Clause' verwijst naar een ander deel in de SQL Reference
 - Values_Clause OF subquery NA Insert_into_clause



Nieuwe rijen toevoegen aan de tabel:

INSERT



a. elke insert instructie voegt exact 1 rij toe aan de tabel, alle attributen krijgen een waarde

Syntax:

```
INSERT INTO tabelnaam(attr1,attr2,...)
```

```
VALUES (waarde1, waarde2,...);
```

Nieuwe rijen toevoegen aan de tabel:

INSERT

COLUMN_NAME	DATA_TYPE
AFD_NR	NUMBER(2,0)
AFD_NAAM	VARCHAR2(20 BYTE)
MGR_SOFI_NR	CHAR(9 BYTE)
MGR_START_DATUM	DATE



Voorbeeld

INSERT INTO afdelingen

VALUES (5, 'Verkoop', '999444444', '22-MEI-1998');

- numerieke waarden → zonder quotes
- alfanumerieke en datumwaarden → tussen enkele quotes
- de volgorde van de waarden komen overeen met de volgorde van de attributen in de tabel
- Als je de volgorde van de attributen in de tabel niet kent kan je het volgende doen:
 - Ofwel zoek je de beschrijving van de tabel op → geeft je de volgorde van de attributen binnen de tabel aan
 - Ofwel geef je achter de tabelnaam je eigen volgorde aan en vul je de waarden in dezelfde volgorde in:

INSERT INTO

***afdelingen(afd_naam,afd_nr,mgr_sofi_nr,
mgr_start_datum)***

VALUES('Verkoop',5, '999444444','22-MEI-1998');

Nieuwe rijen toevoegen aan de tabel:

INSERT

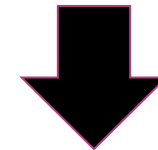


Voorbeeld

b. elke insert voegt exact 1 rij toe aan de tabel maar **niet** alle attributen krijgen een waarde toegewezen:

AFD_NR	AFD_NAAM	MGR_SOFI_NR	MGR_START_DATUM
7	Productie	999444444	22/05/08
3	Administratie	999555555	01/01/11
1	Hoofdvestiging	999666666	19/06/01
5	Verkoop	999444444	22/05/98

INSERT INTO afdelingen(afd_nr,afd_naam)
VALUES(6,'Naverkoop');



COLUMN_NAME	DATA_TYPE	NULLABLE
AFD_NR	NUMBER(2,0)	No
AFD_NAAM	VARCHAR2(20 BYTE)	No
MGR_SOFI_NR	CHAR(9 BYTE)	Yes
MGR_START_DATUM	DATE	Yes

AFD_NR	AFD_NAAM	MGR_SOFI_NR	MGR_START_DATUM
7	Productie	999444444	22/05/08
3	Administratie	999555555	01/01/11
1	Hoofdvestiging	999666666	19/06/01
5	Verkoop	999444444	22/05/98
6	Naverkoop	(null)	(null)

Opgelet: zorg er in dit geval voor dat alle NOT NULL attributen opgenomen werden in het INSERT statement!

Nieuwe rijen toevoegen aan de tabel:

INSERT



Voorbeeld

c. gebruik van de substitutievevariabele &

```
INSERT INTO afdelingen(afd_naam,afd_nr)  
VALUES('&afd_naam',&afd_nr);
```

Enter Substitution Variable

AFD_NAAM:

OK Annuleren

Enter Substitution Variable

AFD_NR:

OK Annuleren

Nieuwe rijen toevoegen aan de tabel:

INSERT

c. gebruik van de substitutievariabele &

```
9 | INSERT INTO afdelingen(afd_naam,afd_nr)
10 | VALUES ('&afd_naam',&afd_nr);
11 |
```

Script Output x

Task completed in 0 seconds

old:INSERT INTO afdelingen(afd_naam,afd_nr)
VALUES('&afd_naam',&afd_nr)
new:INSERT INTO afdelingen(afd_naam,afd_nr)
VALUES('Boekhouding',8)
1 rows inserted.

AFD_NR	AFD_NAAM	MGR_SOFI_NR	MGR_START_DATUM
7	Productie	999444444	22/05/08
3	Administratie	999555555	01/01/11
1	Hoofdvestiging	999666666	19/06/01
5	Verkoop	999444444	22/05/98
6	Naverkoop	(null)	(null)
8	Boekhouding	(null)	(null)



In bestaande rijen wijzigingen aanbrengen:

UPDATE



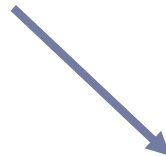
UPDATE tabelnaam

SET attribuutnaam=nieuwe waarde,...

WHERE conditie



enkel de rijen die voldoen
aan de voorwaarde worden
gewijzigd



- constante
- rekenkundige bewerking
- select statement (later)

In bestaande rijen wijzigingen aanbrengen:

UPDATE



Voorbeeld

Er komt in afdeling 1 vanaf 1 april 2017 een nieuwe manager (met sofi_nr 999111111)

UPDATE afdelingen

SET mgr_sofi_nr='999111111';

mgr_start_datum='1-apr-2017'

WHERE afd_nr=1;

constante

In bestaande rijen wijzigingen aanbrengen:

UPDATE



Voorbeeld

Medewerker 999111111 krijgt een loonsverhoging van 10%.

UPDATE medewerkers

SET salaris= salaris*1.1

WHERE sofi_nr='999111111';



rekenkundige bewerking

Bestaande rijen verwijderen: **DELETE**



DELETE FROM tabelnaam
WHERE condition;



enkel de rijen die aan de voorwaarde
voldoen worden verwijderd

Bestaande rijen verwijderen: DELETE



Voorbeeld

DELETE FROM afdeling;

→ alle rijen uit afdeling worden verwijderd

DELETE FROM afdeling

WHERE afd_nr=7;

→ de rij betreffende afdeling 7 wordt verwijderd

opgelet: FK's waken over de afhankelijke rijen!

Wijzigingen definitief maken of opnieuw ongedaan maken



Definitie

COMMIT om een wijziging definitief te maken

ROLLBACK om een wijziging opnieuw ongedaan te maken (tot aan de laatst gegeven commit)

Rollback segmenten

Voor de uitvoering van een dml instructie (update, delete, insert) wordt de oude inhoud van de te wijzigen rijen in **rollback segmenten** gestockeerd.

Daardoor kunnen DML instructies terug ongedaan gemaakt worden met een ROLLBACK statement

Alle rijen onherroepelijk verwijderen:

TRUNCATE



TRUNCATE TABLE *tabelnaam*;

Verwijdert alle rijen uit een tabel.

Deze instructie is onherroepelijk. Ze kan niet ongedaan gemaakt worden door een ROLLBACK statement.

Waarom kan het niet ongedaan worden gemaakt?

Bij het geven van een TRUNCATE wordt de oude situatie niet in rollback segmenten gestockeerd!

DDL instructie : TRUNCATE

Wanneer gebruiken?

Om een zeer grote tabel snel definitief leeg te maken

Waarom gaat dit sneller?

Omdat die grote hoeveelheid informatie niet eerst naar rollback segmenten wordt gekopieerd.

TRUNCATE TABLE medewerkers;



Voorbeeld

DML instructies en constraints

Bij creatie van tabellen worden er op die tabellen constraints aangemaakt.

Het DBMS zal een DML instructie weigeren uit te voeren wanneer gezondigd wordt tegen een constraint

Welke constraints kan je op een tabel zetten?

PRIMARY KEY constraint

NOT NULL constraint

FOREIGN KEY constraint

CHECK constraint

UNIQUE constraint

Bij het uitvoeren van een DML instructie kan je zondigen tegen deze constraints.



DML instructies en constraints

Voorbeeld fout op

PRIMARY KEY constraint

```
9  INSERT INTO afdelingen(afd_nr,afd_naam)
10 VALUES(&afd_nr,&afd_naam');
11
```

Script Output x

Task completed in 0,031 seconds

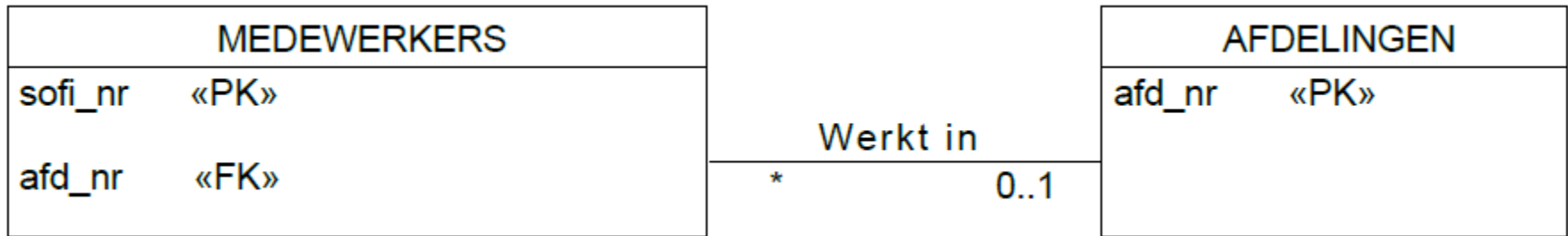
Error starting at line : 9 in command -
INSERT INTO afdelingen(afd_nr,afd_naam)
VALUES(&afd_nr,&afd_naam')

Error report -
SQL Error: ORA-00001: unique constraint (THEORIE.PK_AFDELINGEN) violated
00001. 00000 - "unique constraint (%s.%s) violated"

*Cause: An UPDATE or INSERT statement attempted to insert a duplicate key.
For Trusted Oracle configured in DBMS MAC mode, you may see
this message if a duplicate entry exists at a different level.

*Action: Either remove the unique restriction or do not insert the key.

DML instructies en constraints



CHILD TABLE

PARENT TABLE

Wat gebeurt er bij het verwijderen van een afdeling?

Wat gebeurt bij het wijzigen van de afdeling van een medewerker?

Wat gebeurt er bij het verwijderen van een medewerker?

DML instructies en constraints

DML Statement	Bij de Parent Tabel	Bij de Child Tabel
INSERT	Altijd OK als de parent sleutelwaarde uniek is	OK enkel indien de the foreign key waarde bestaat in parent key of is partieel geheel gelijk aan NULL
UPDATE geen actie	Toegelaten als het statement geen rijen in de child tabel achterlaat zonder parent key waarde	Toegelaten als de nieuwe foreign key waarde nog refereert naar een geldige sleutelwaarde
DELETE geen Actie	Toegelaten als er geen rijen refereren naar de parent key waarde	Altijd OK
DELETE Cascade	Altijd OK	Altijd OK
DELETE Set Null	Altijd OK	Altijd OK