

# Data Analysis & SQL Data Manipulation Language

Koen Bloemen



**DE HOGESCHOOL  
MET HET NETWERK**

Elfde-Liniestraat 24, 3500 Hasselt, [www.pxl.be](http://www.pxl.be)





- 4.1 Gegevens invoeren
- 4.2 Gegevens wijzigen
- 4.3 Gegevens verwijderen
- 4.4 Transactieverwerking
- 4.5 Oefeningen

# DML

- DML = Data Manipulation Language
  - INSERT
  - UPDATE
  - DELETE
- Gegevens aanpassen, niet de structuur
- In productieomgeving gebeurt dit d.m.v. applicaties
  - Oracle Forms
  - .NET applicaties
  - ...
- Waarom DML?
  - Globale aanpassingen
  - Grote updates
  - Correcties



## 4.1 Gegevens invoeren

- Gegevens op 2 manieren toevoegen
  - Via VALUES component -> verzameling kolomwaarden opgeven
    - Rijen 1 per 1 toevoegen
  - Via SUBQUERY -> gebruik maken van bestaande gegevens
    - Mogelijk om meerdere rijen ineens toe te voegen
- INSERT INTO tabelnaam [(kolomnaam,...)]  
VALUES (expressie,...)
- INSERT INTO tabelnaam [(kolomnaam,...)]  
SELECT [kolomnaam,...]  
FROM tabelnaam

## 4.1 Gegevens invoeren

- Fysieke kolomvolgorde

```
INSERT INTO medewerkers  
VALUES (7955, 'NIJS', 'P', 'TRAINER', 7566, date '1967-02-16', 2660, NULL, 20);
```

- Willekeurige volgorde (niet alle kolommen zijn nodig)

```
INSERT INTO medewerkers (naam,voorn,mnr,maandsal,gbdatum)  
VALUES ('VOS', 'G', 7956, 2660, date '1963-05-22');
```

## 4.1 Gegevens invoeren

- Geen hardcoded values? => substitutievariabelen

INSERT INTO medewerkers

VALUES (&mnr,&naam,&voorn,&functie,&chef,to\_date('&gebdatum','dd-mm-yyyy'),&maansal,&comm,&afd);

```
SQL> INSERT INTO medewerkers^KVALUES (&mnr,&naam,&voorn,&functie,&chef,to_date('&gebdatum','dd-mm-yyyy'),&maansal,&comm,&afd);
Enter value for mnr: 7677
Enter value for naam: DENIJS
Enter value for voorn: A
Enter value for functie: TRAINER
Enter value for chef: 7566
Enter value for gebdatum: 24-09-1997
Enter value for maansal: 2960
Enter value for comm: null
Enter value for afd: 20
old 1: INSERT INTO medewerkers^VALUES (&mnr,&naam,&voorn,&functie,&chef,to_date('&gebdatum','dd-mm-yyyy'),&maansal,&comm,&afd)
new 1: INSERT INTO medewerkers^VALUES (7677,'DENIJS','A','TRAINER',7566,to_date('24-09-1997','dd-mm-yyyy'),2960,null,20)

1 row created.
```

MERK OP: alfanumerieke waarden en datums worden tussen " geplaatst

# 4.1 Gegevens invoeren

- Gegevens toevoegen o.b.v. een query

```
INSERT INTO schalen
```

```
SELECT snr+5, ondergrens + 2300, bovengrens + 2300, 500
```

```
FROM schalen
```

```
WHERE snr <= 4;
```

SNR	ONDERGRENS	BOVENGRENS	TOELAGE
1	1700	2200	0
2	2201	2400	50
3	2401	4000	100
4	4001	5000	200
5	5001	9999	500
6	4000	4500	500
7	4501	4700	500
8	4701	6300	500
9	6301	7300	500

Rijen 6 t.e.m. 9 zijn toegevoegd

Voordeel = meerdere rijen ineens toevoegen

## 4.2 Gegevens wijzigen

- UPDATE tabelnaam  
SET kolomnaam = waarde, ....  
WHERE voorwaarde
- Opgelet: Indien de WHERE-component wordt weggelaten worden de wijzigingen op alle rijen uitgevoerd



## 4.2 Gegevens wijzigen

- Meerdere kolommen van rij 7666 in de tabel wijzigen

```
UPDATE medewerkers  
SET functie = 'TRAINER'  
  , maandsal = 5030  
  , afd = 20  
WHERE mnr = 7666;
```

- **ALLE** medewerkers krijgen een loonsverhoging van 10%

```
UPDATE medewerkers  
SET maandsal = maandsal * 1.1;
```

- De werknemers van het hoofdkantoor krijgen een loonsverhoging van 10%

```
UPDATE medewerkers  
SET maandsal = maandsal * 1.1  
WHERE afd = (SELECT anr FROM afdelingen WHERE naam = 'HOOFDKANTOOR');
```

## 4.3 Gegevens verwijderen

- DELETE FROM tabelnaam  
WHERE voorwaarde

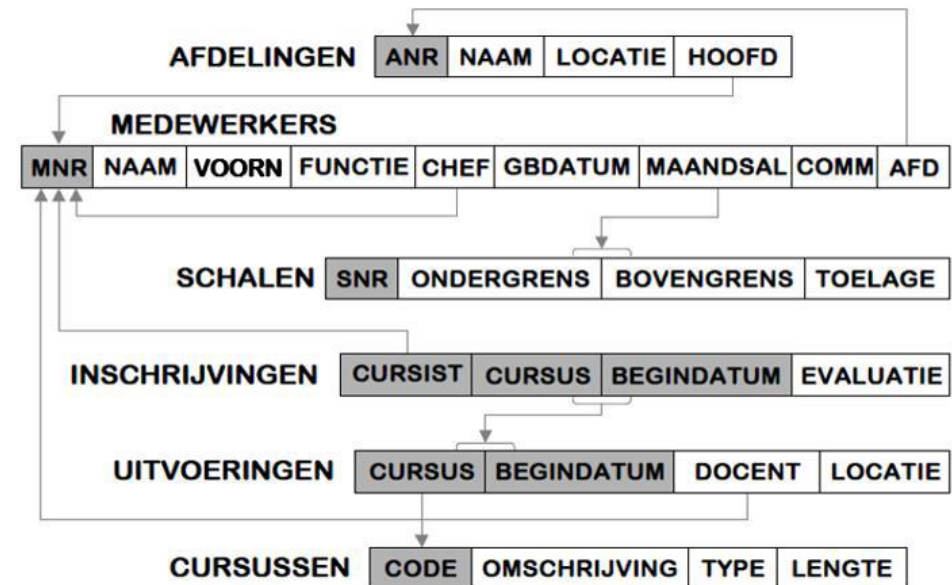
```
DELETE FROM medewerkers  
WHERE mnr = 7499;
```

- Opgelet: Indien de WHERE-component wordt weggelaten worden alle rijen verwijderd met als resultaat een lege tabel!
- DELETE <-> DROP TABLE
  - DROP: DDL-commando  
Dit verwijdert niet alleen inhoud, maar ook de gegevens en de daarmee samenhangende **structuren** (indexen, privileges, vreemde sleutels,...)
  - DELETE: DML-commando  
Dit wijzigt de NIET de structuur, maar verwijdert enkel de **inhoud** van de tabel. Beter TRUNCATE gebruiken (vlugger).

## 4.3 Gegevens verwijderen

- Constrains niet vergeten bij toevoegen, verwijderen of wijzigen. Er kunnen geen gegevens toegevoegd worden die niet voldoen aan voorwaarden. Gegevens kunnen ook niet verwijderd worden die verwijzen naar een andere tabel.
  - ALTER TABLE medewerkers  
DISABLE constraint med\_afd\_FK
- 
- ```

graph TD
    subgraph AFDELINGEN
        ANR[ANR]
        NAAM[NAAM]
        LOCATIE[LOCATIE]
        HOOFD[HOOFD]
    end
    subgraph MEDEWERKERS
        MNR[MNR]
        NAAM2[NAAM]
        VOORN[VOORN]
        FUNCTIE[FUNCTIE]
        CHEF[CHEF]
        GBDATUM[GBDATUM]
        MAANDSAL[MAANDSAL]
        COMM[COMM]
        AFD[AFD]
    end
    AFD --> ANR
  
```



## 4.4 Transacties

- Wijzigingen in de inhoud krijgen in eerste instantie een voorlopig karakter
- COMMIT  
Wijzigingen worden hier pas definitief gemaakt
- ROLLBACK [(TO SAVEPOINT savepointnaam)]  
Wijzigingen worden ongedaan gemaakt
- SAVEPOINT naam  
Markeringspunt plaatsen (tussen verschillende mutaties)



# Transacties

DML-commando 1

→ **Commit**

*mutatie 1 wordt definitief*

DML-commando 2

DML-commando 3

**Savepoint EEN**

DML-commando 4

DML-commando 5

→ **Savepoint TWEE**

DML-commando 6

DML-commando 7

→ **Rollback to savepoint TWEE**

*mutaties 6 en 7 worden ongedaan gemaakt*

→ **Commit**

*mutaties 2, 3, 4 en 5 worden definitief*

# Transacties

- Mutaties vergeten te 'committen' -> risico bij systeemstoring om werk te verliezen. Mutaties zijn nog niet zichtbaar voor andere gebruikers.
- Oracle hanteert een impliciete commit bij het beëindigen van SQL\*Plus.
- Alle DDL-Commando's en DCL-commando's hanteren een impliciete commit. Dit betekent dat transacties onmiddellijk definitief zijn!
- Oracle kent ook een autocommit om DML-commando's onmiddellijk te laten 'committen' (SET AUTOCOMMIT ON ). LET OP, hierbij is rollback niet meer mogelijk!

# Oefeningen p. 65