

# Wijzigen van de tabelstructuur

## Verwijderen van een tabel

Alter table

Drop table



# cursusmateriaal

- Cursus 'Databanken 1' blz. 51-53
- SQL Reference blz. 52 ev.
- Deze powerpoint
- Extra's:
- Hoofdstuk 11 handboek 'SQL fundamentals I Exam Guide' blz.465-466
- [http://docs.oracle.com/database/121/SQLRF/statements\\_3001.htm#SQLRF01001](http://docs.oracle.com/database/121/SQLRF/statements_3001.htm#SQLRF01001)



# ALTER TABLE



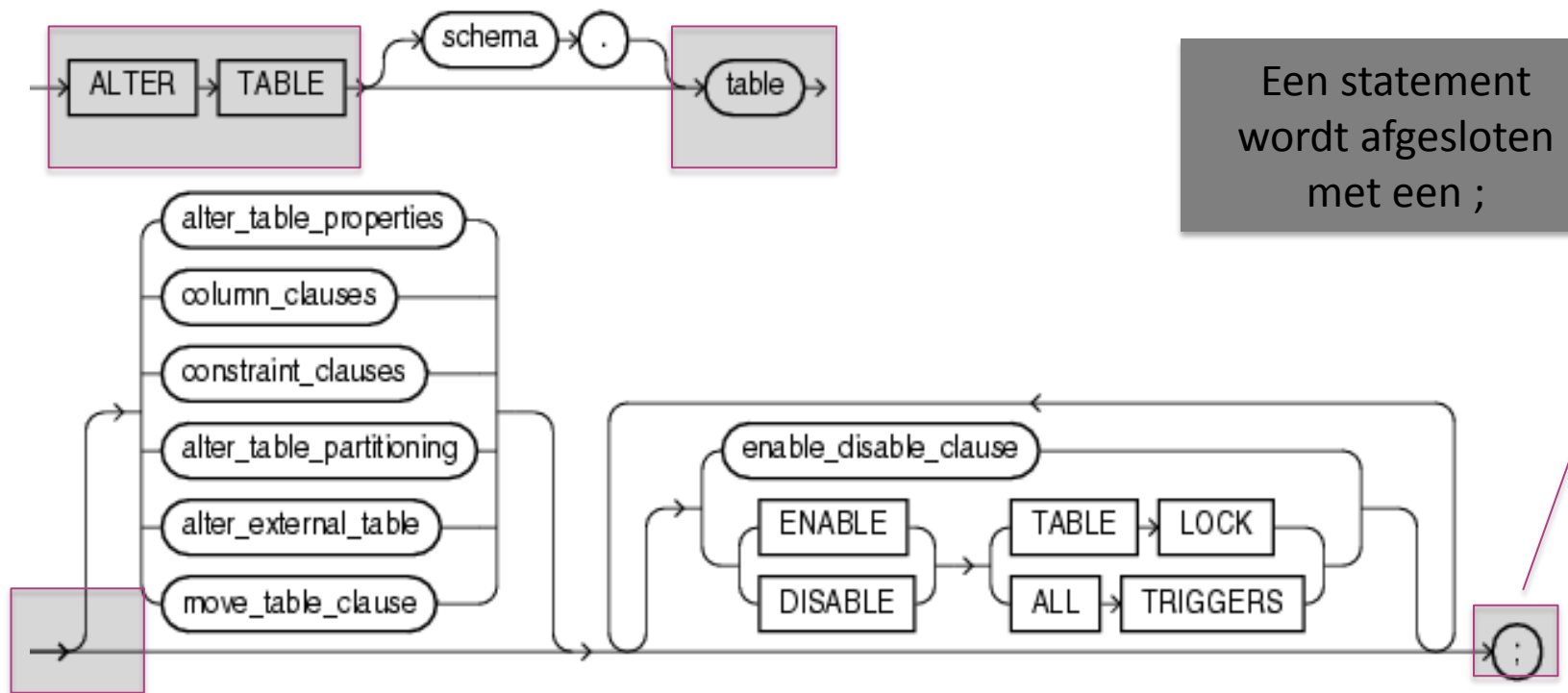
Daarvoor gebruiken we de instructie:

**ALTER TABLE tabelnaam**  
**+ één van de onderstaande opties**  
**(of een combinatie ervan)**

- ☐ **ADD**
- ☐ **MODIFY**
- ☐ **DROP**
- ☐ **SET UNUSED COLUMN**
- ☐ **RENAME COLUMN**
- ☐ **READ ONLY**

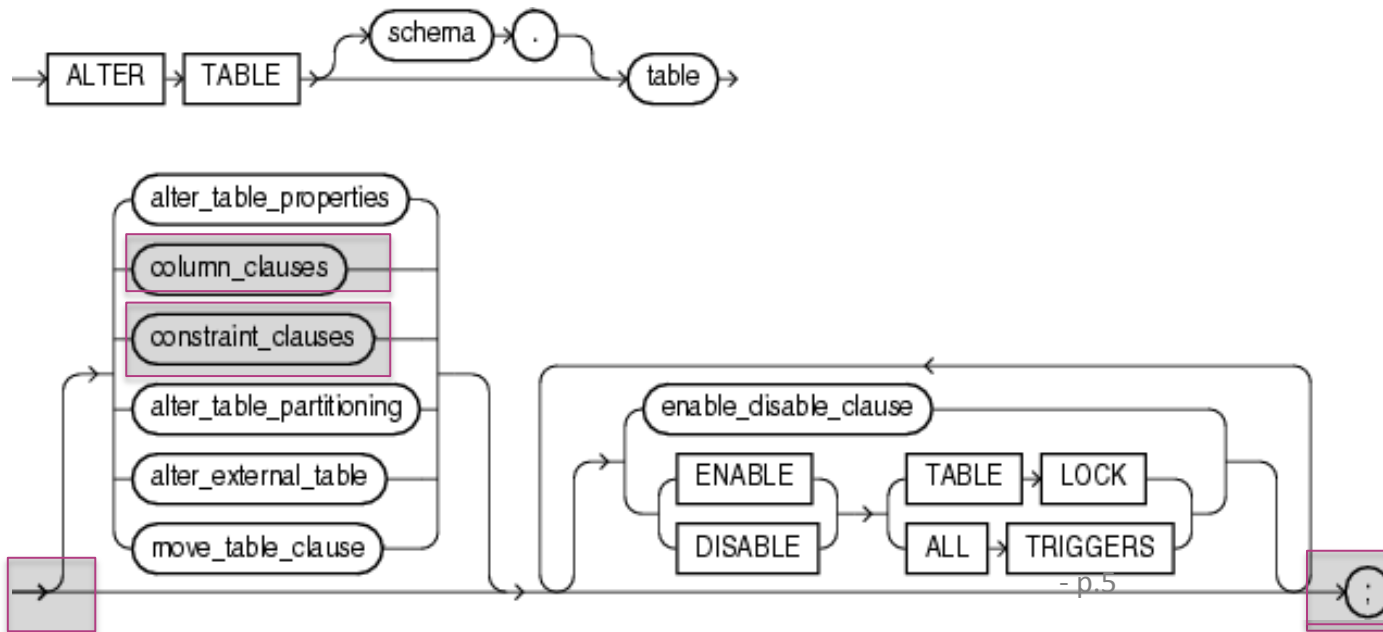
# SQL Reference

**SQL**  
Reference  
p. 52



## SQL Reference

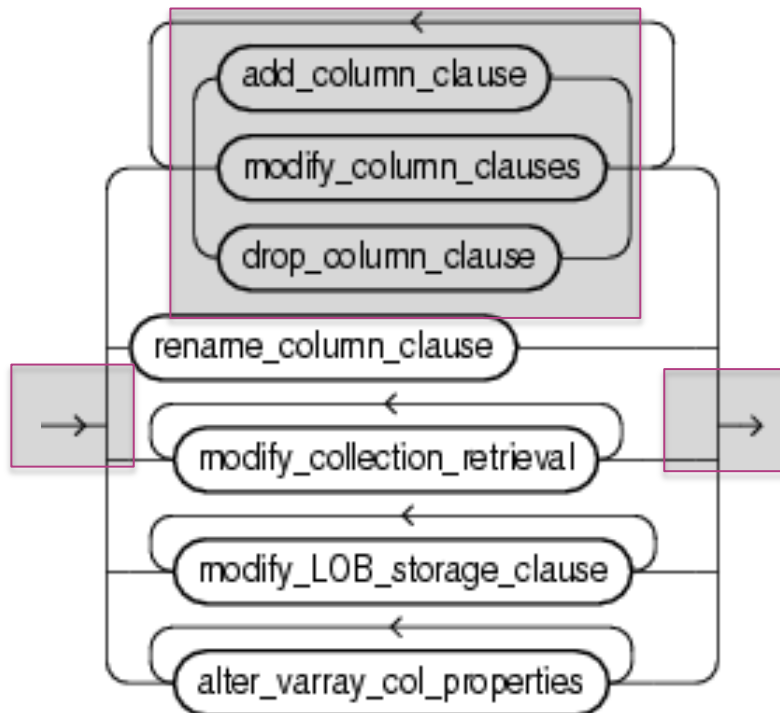
- Een 'Clause' verwijst naar een ander deel in de reference
  - is een deel van een instructie



# SQL Reference

## *column\_clauses*

zie SQL Reference verder onder  
ALTER TABLE p. 55



# ALTER TABLE



Definitie

### ADD-optie

- ☐ attributen toevoegen
- ☐ om table constraints toe te voegen

### MODIFY-optie

- ☐ om in bestaande attributen wijzigingen aan te brengen (datatype wijzigen, default waarde toevoegen of verwijderen, column constraints toevoegen)

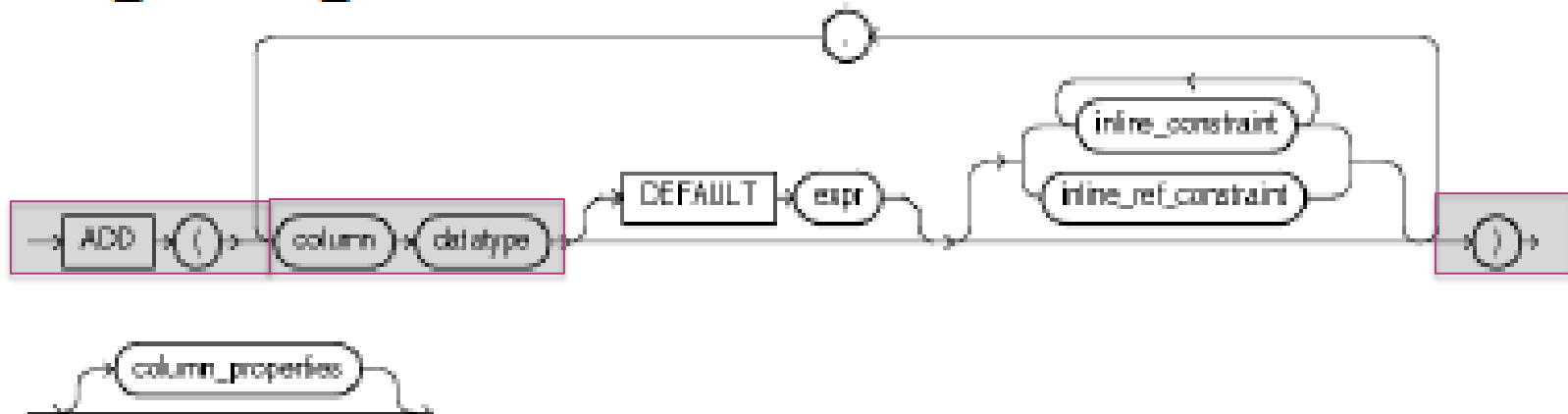
### DROP-optie

- ☐ attribuut verwijderen
- ☐ constraint verwijderen

# SQL Reference

**SQL**  
Reference  
p. 55

## *add\_column\_clause*





## Tabelstructuur wijzigen

---

### ALTER TABLE

#### SET UNUSED COLUMN-optie

zet de weergegeven kolom op 'unused'.

het is nu of de kolom niet meer bestaat

→ SQL referentie p55 (drop\_column\_clause)

#### RENAME COLUMN-optie

om een kolom een nieuwe naam te geven

→ SQL referentie p55 (rename\_column\_clause)

#### READ ONLY-optie

er kan enkel informatie uit de tabel gehaald worden, de inhoud kan niet gewijzigd worden.

→ niet in SQL referentie (nieuw in Oracle 11g)

# ALTER TABLE - ADD optie



(toevoegen van een attribuut aan een tabel)

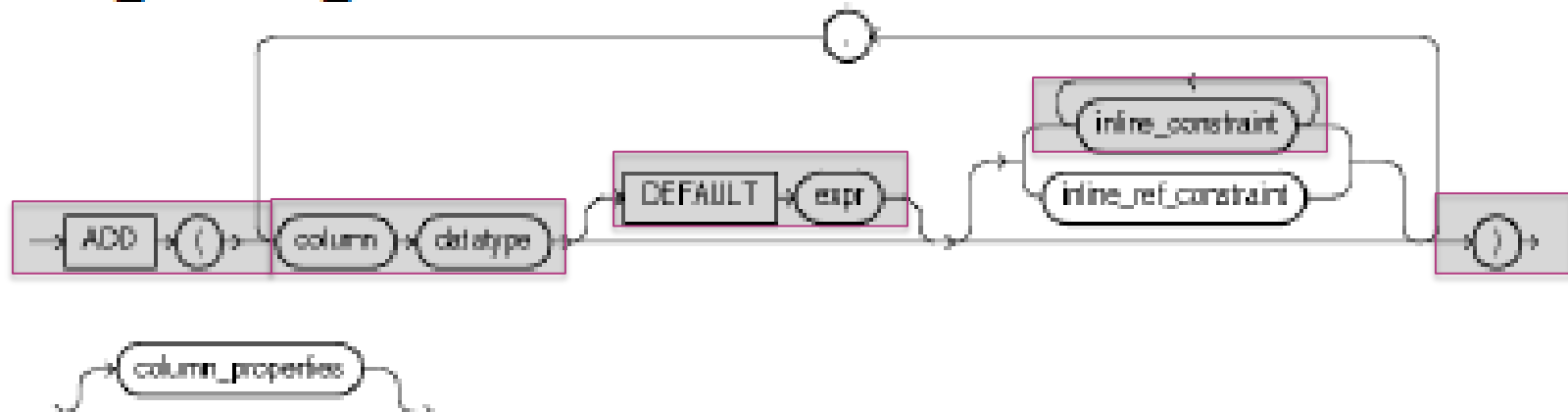
```
ALTER TABLE tabelnaam  
ADD ( attribuutnaam gegevenstype  
    [default waarde]  
    [column constraint-clausule] ,...);
```

- ❑ Om een attribuut (eventueel met default waarde en constraints) toe te voegen aan de tabel.
- ❑ Het attribuut wordt **achteraan de tabel toegevoegd**.
- ❑ Je kan in 1 instructie meerdere attributen toevoegen.

# SQL Reference

**SQL**  
Reference  
p. 55

## *add\_column\_clause*



# ALTER TABLE - ADD-optie



Voorbeeld

Mag ook zonder haakjes behalve als je meerdere attributen in 1 keer wenst toe te voegen!

```
ALTER TABLE medewerkers  
  ADD geslacht char(1);
```

```
❑ ALTER TABLE medewerkers  
  ADD adres VARCHAR2(50) CONSTRAINT  
    c_adres CHECK(adres=UPPER(adres));
```

```
❑ ALTER TABLE medewerkers  
  ADD(plaats VARCHAR2(25),  
    provincie CHAR(2));
```

# ALTER TABLE - ADD optie



(om table constraints toe te voegen aan een tabel)

**ALTER TABLE tabelnaam**

**ADD ( table constraint-clausule,...);**

Om een tabel te voorzien van één of meerdere **table** constraints.

### ALTER TABLE - ADD-optie



Voorbeeld

```
ALTER TABLE opdrachten  
    ADD CONSTRAINT pk_opdrachten  
    PRIMARY KEY(sofi_nr,proj_nr);
```

- Je kan in één instructie meerdere table constraints toevoegen:



```
ALTER TABLE tabelnaam  
    ADD (table constraint1,  
        table constraint2);
```

# ALTER TABLE - **MODIFY**-optie



```
ALTER TABLE tabelnaam  
    MODIFY (attribuutnaam [datatype]  
            [default waarde]  
            [constraint-clausule] ,...);
```

- Om in een **bestaand attribuut** een **wijziging** aan te brengen:
  - ☐ een attribuut verbreden of versmallen,
  - ☐ een default waarde toevoegen of weghalen,
  - ☐ een column constraint toevoegen,
  - ☐ het gegevenstype van het attribuut wijzigen\*

# ALTER TABLE - MODIFY-optie



Voorbeeld

❑ *ALTER TABLE medewerkers*

*MODIFY parkeerplaats*

*CONSTRAINT nn\_parkeerplaats NOT NULL;*

❑ *ALTER TABLE medewerkers*

*MODIFY achternaam CONSTRAINT c\_achternaam*

*CHECK(achternaam=UPPER(achternaam));*

❑ *ALTER TABLE medewerkers*

*MODIFY parkeerplaats NUMBER(5);*

(kolom wordt verbreed van 4 naar 5)

❑ *ALTER TABLE medewerkers*

*MODIFY salaris DEFAULT 0;*



# ALTER TABLE - MODIFY-optie



Voorbeeld

❑ *ALTER TABLE medewerkers*

*MODIFY(*

*achternaam constraint c\_achternaam*

*CHECK(achternaam=UPPER(achternaam)),*

*voornaam constraint c\_voornaam*

*CHECK(voornaam=UPPER(voornaam)));*

**Je kan in 1 instructie meerdere attributen uit de tabel wijzigen**

### ALTER TABLE - DROP-optie



```
ALTER TABLE tabelnaam  
DROP COLUMN attribuutnaam;
```



- Om een **attribuut** te **verwijderen** uit de tabel.

```
ALTER TABLE tabelnaam  
DROP CONSTRAINT constraint_name;
```



- Om een **constraint** te **verwijderen**. Je hebt daartoe de naam van de constraint nodig. Deze vind je in de dictionary tabel USER\_CONSTRAINTS of via connections in SQL Developer

## Tabelstructuur wijzigen

---

# ALTER TABLE - DROP-optie

⚡ CONSTRAINT_NAME	⚡ CONSTRAINT_TYPE	SEARCH_CONDITION
CK_SALARIS	Check	salaris <= 85000
FK_MED_AFD	Foreign_Key	(null)
FK_MED_MED	Foreign_Key	(null)
NN_ACHTERNAAM	Check	"ACHTERNAAM" IS NOT NULL
NN_VOORNAAM	Check	"VOORNAAM" IS NOT NULL
PK_MEDEWERKERS	Primary_Key	(null)
UN_PARKEERPLAATS	Unique	(null)

# ALTER TABLE - DROP-optie



Voorbeeld

*ALTER TABLE medewerkers*  
*DROP COLUMN salaris;*

☐ *ALTER TABLE medewerkers*  
*DROP CONSTRAINT nn\_achternaam;*

## Tabelstructuur wijzigen

---

### ALTER TABLE - DROP-optie

#### Bemerking 1:

Wat als je een PRIMARY KEY constraint wil verwijderen? Je kan maar verwijzen naar een attribuut (of combinatie van attributen) wanneer dat attribuut of die combinatie van attributen een PK constraint heeft! Hoe oplossen?

*ALTER TABLE projecten*

*DROP CONSTRAINT pk\_project **CASCADE**;*

Of

*ALTER TABLE projecten*

*DROP PRIMARY KEY **CASCADE**;*

Gevolg: de FK constraints die verwijzen naar die PK worden mee verwijderd.

## Tabelstructuur wijzigen

---

### **ALTER TABLE - DROP-optie**

#### Bemerking 2:

Wanneer je een bestaande constraint wil wijzigen, dan kan dat niet met de MODIFY optie! Je moet de constraint in dat geval verwijderen (DROP) en opnieuw creëren (MODIFY of ADD).



voor een column  
constraint

voor een  
table constraint

## Tabelstructuur wijzigen

---

### ALTER TABLE - DROP-optie



Voorbeeld

Op het attribuut salaris uit de tabel MEDEWERKERS staat volgende check constraint:

```
CHECK(salaris<=85000);
```

Die bovengrens moet verhoogd worden naar 95000.

Hoe tewerk gaan?

Eerst de constraint droppen:

```
ALTER TABLE medewerkers  
  DROP CONSTRAINT ck_salaris ;
```

*Daarna de aangepaste constraint aanmaken:*

```
ALTER TABLE medewerkers  
  MODIFY salaris  
  CONSTRAINT ck_salaris CHECK(salaris<=95000);
```

## Tabelstructuur wijzigen

# ALTER TABLE – SET UNUSED COLUMN

**ALTER TABLE** tabelnaam  
**SET UNUSED COLUMN** kolomnaam



Een beschrijving van de tabel AFDELINGEN geeft:

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

*ALTER TABLE afdelingen*

*SET UNUSED COLUMN mgr\_start\_datum;*

Een beschrijving van de tabel AFDELINGEN geeft nu:

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

Er wordt met de kolom geen rekening meer gehouden.

**Opgelet! Deze actie is onomkeerbaar.**



# ALTER TABLE – SET UNUSED COLUMN

### Bedenking:

Waarom een kolom die je niet meer gebruikt  
UNUSED zetten?

Waarom ze niet meteen droppen?

- ☐ voor grote tabellen is het droppen van een kolom tijdrovend + het vraagt veel resources.
- ☐ daarom zet men de betreffende kolom voorlopig unused om ze dan op een ander geschikt moment te droppen.

## Tabelstructuur wijzigen

---

# ALTER TABLE – RENAME COLUMN

**ALTER TABLE** tabelnaam

**RENAME COLUMN** oude\_kolomnaam **TO**  
nieuwe\_kolomnaam



*ALTER TABLE afdelingen*

*RENAME COLUMN afd\_naam TO naam\_afd;*



Voorbeeld

Een beschrijving van de tabel AFDELINGEN geeft nu:

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

## Tabelstructuur wijzigen

---

### ALTER TABLE – READ ONLY

**ALTER TABLE tabelnaam READ ONLY**

*ALTER TABLE afdelingen*

*READ ONLY;*

De tabel kan enkel geraadpleegd worden.

Inhoudelijk kan je niets veranderen.

*UPDATE afdelingen*

*SET afd\_naam=UPPER(afd\_naam);*

Error report:

SQL Error: ORA-12081: update operation not allowed on table "THEORIE1"."AFDELING

12081. 00000 - "update operation not allowed on table \"%s\".\"%s\""

\*Cause: An attempt was made to update a read-only materialized view.

\*Action: No action required. Only Oracle is allowed to update a read-only materialized view.



Voorbeeld

Opnieuw inhoudelijk aanpasbaar maken kan met

**ALTER TABLE tabelnaam  
READ WRITE**

# DROP TABLE



**DROP TABLE** tabelnaam  
**[CASCADE CONSTRAINTS];**

Je kan een tabel maar verwijderen wanneer er geen foreign keys verwijzen naar de betreffende tabel, **zelfs wanneer je bij de vreemde sleutel ON DELETE CASCADE/SET NULL definieerde!!**

## Tabel verwijderen

# DROP TABLE

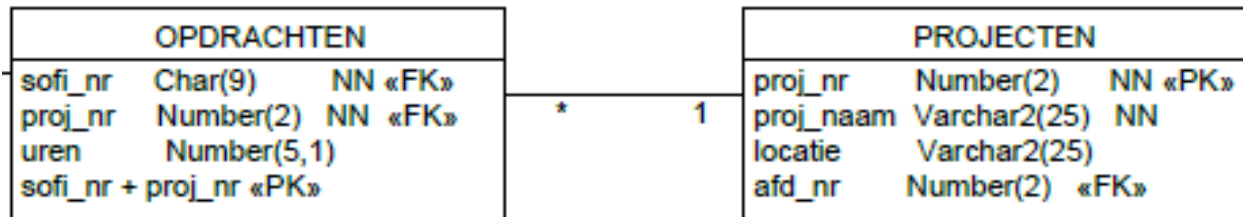


Voorbeeld

*DROP TABLE projecten;*

→ zal een ERROR geven omdat in de tabel OPDRACHTEN het attribuut proj\_nr nog verwijst naar de tabel PROJECTEN.

(Zelfs een foreign key constraint met ON DELETE CASCADE zal een ERROR geven!)



# DROP TABLE

Men kan de verwijdering van een tabel waar nog vreemde sleutels naar verwijzen enkel afdwingen met de

**CASCADE CONSTRAINTS-optie.**

*DROP TABLE projecten CASCADE CONSTRAINTS;*



Voorbeeld

**→ zorgt ervoor dat alle foreign key constraints die nog naar PROJECTEN verwijzen, verwijderd worden.**

## Tabel verwijderen

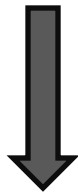
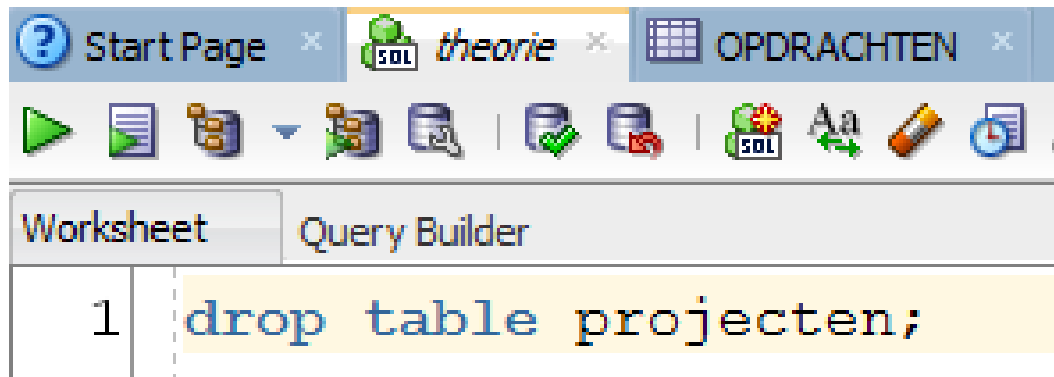
# DROP TABLE

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Connections' pane displays a tree structure for the 'theorie' database, with tables including AFDELINGEN, GEZINSLEDEN, LOCATIES, MEDEWERKERS, OPDRACHTEN (highlighted), and PROJECTEN. The main window shows the 'Constraints' tab for the 'OPDRACHTEN' table. The table lists three constraints: 1 FK\_OPD\_MED, 2 FK\_OPD\_PROJ (circled in red), and 3 PK\_OPDRACHTEN. The columns are CONSTRAINT\_NAME, CONSTRAINT\_TYPE, and SEARCH\_CONDITION.

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION
1	FK_OPD_MED	Foreign_Key	(null)
2	FK_OPD_PROJ	Foreign_Key	(null)
3	PK_OPDRACHTEN	Primary_Key	(null)

## Tabel verwijderen

# DROP TABLE



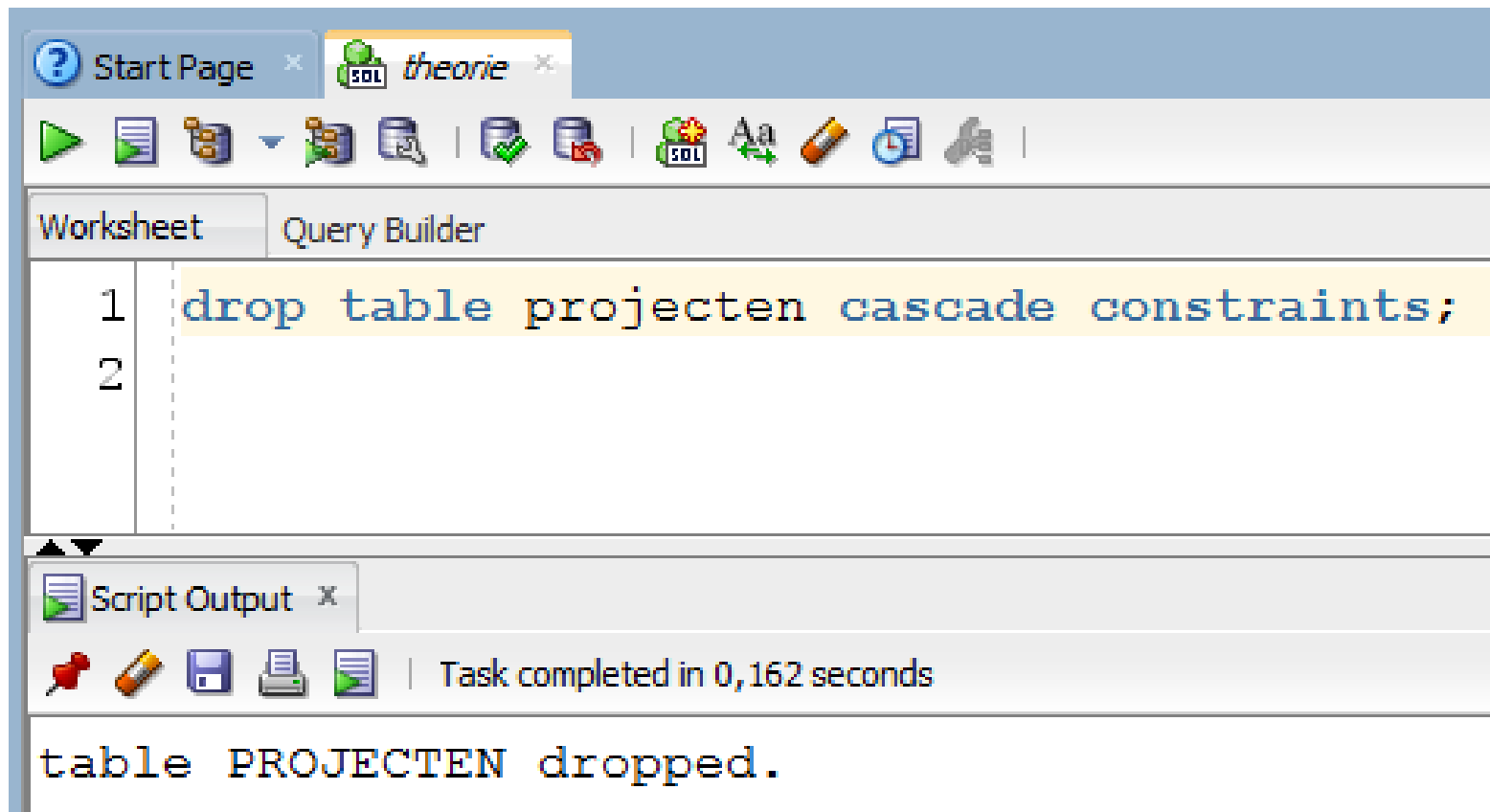
### Error report:

SQL Error: ORA-02449: unique/primary keys in table referenced by foreign keys  
02449. 00000 - "unique/primary keys in table referenced by foreign keys"  
\*Cause: An attempt was made to drop a table with unique or  
primary keys referenced by foreign keys in another table.



## Tabel verwijderen

# DROP TABLE



## Tabel verwijderen

# DROP TABLE

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Connections' pane displays the 'theorie' database connection. Under 'Tables (Filtered)', the 'OPDRACHTEN' table is highlighted. On the right, the 'Constraints' tab for the 'OPDRACHTEN' table is active, showing a list of constraints.

	CONSTRAINT_NAME	CONSTRAINT_TYPE
1	FK_OPD_MED	Foreign_Key
2	PK_OPDRACHTEN	Primary_Key

Log aan in SQL Developer.



## Oefeningen

---

# Wijzigen tabelstructuur – verwijderen tabellen

Neem de oefeningen op Alter en Drop van blackboard en het script *create\_ondernemingsdatabank* (zie *databank scripts op bb*)

Databanken 1 week 3 > Theorie:

⌘ SQL & Oracle  
Databases 1

Mededelingen

Contactinfo

Cursuswijzer

Mijn cijfers

Tools

Software

Databank scripts



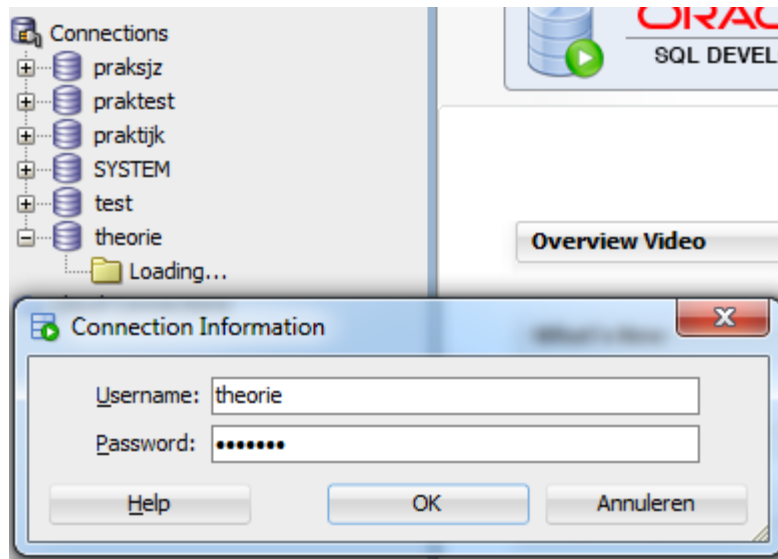
## Oefeningen

---

# Wijzigen tabelstructuur – verwijderen tabellen

Log aan als gebruiker **THEORIE**.

Kopieer het script create\_ondernemingsdatabank

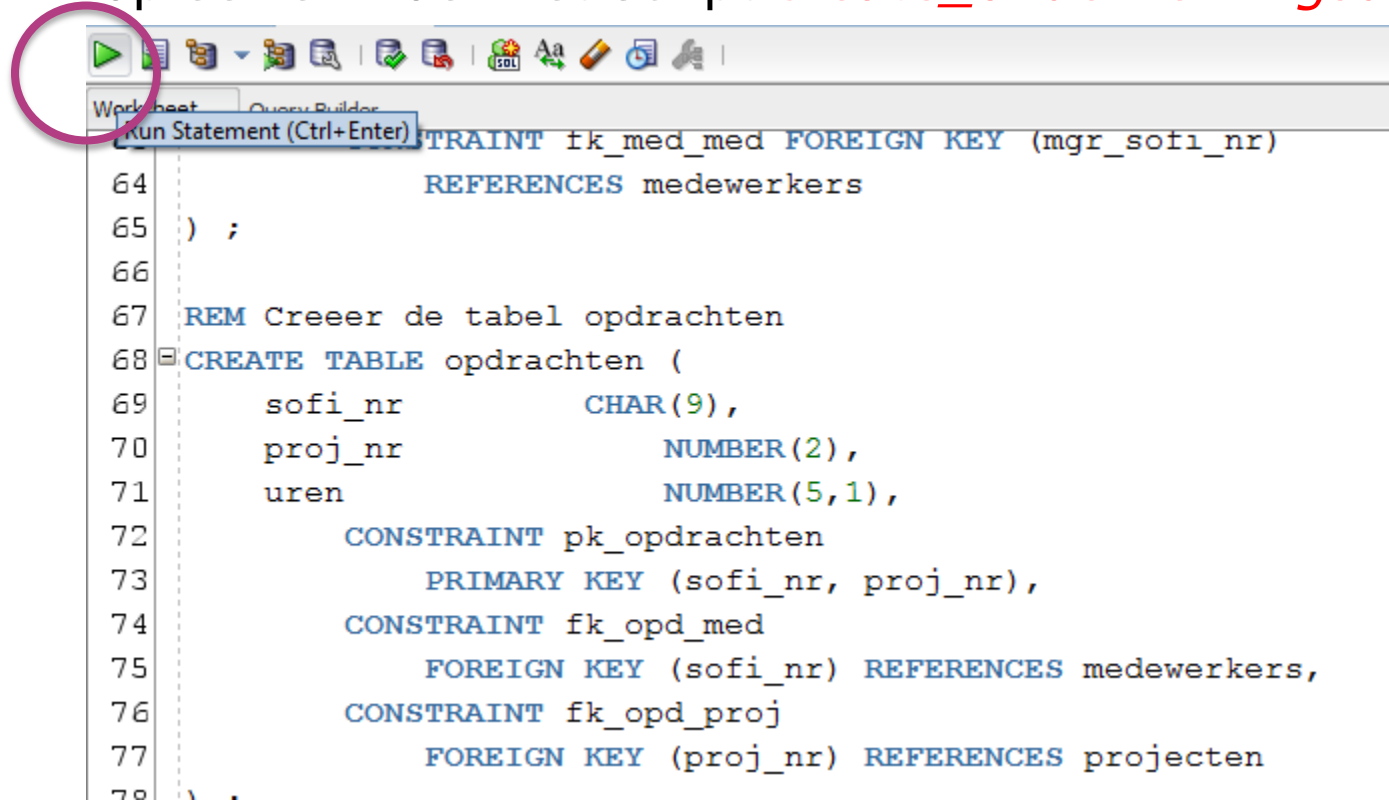


## Oefeningen

# Wijzigen tabelstructuur – verwijderen tabellen

Log aan als gebruiker THEORIE.

Kopieer en voer het script *create\_ondernemingsdatabank* uit



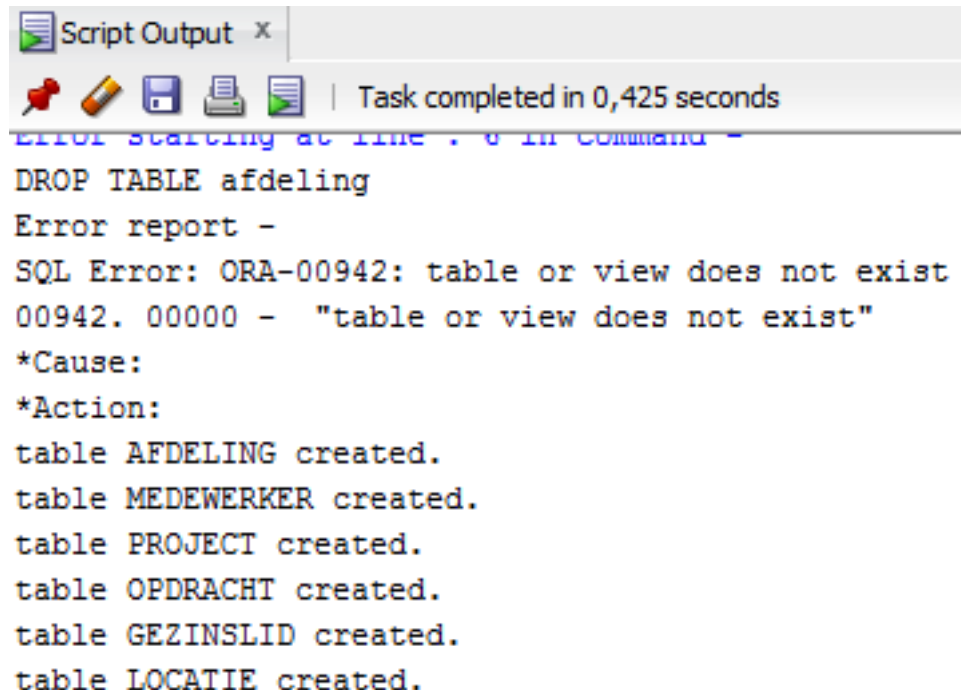
```
64          CONSTRAINT fk_med_med FOREIGN KEY (mgr_sofi_nr)
65          REFERENCES medewerkers
66
67 REM Creeer de tabel opdrachten
68 CREATE TABLE opdrachten (
69     sofi_nr          CHAR(9),
70     proj_nr          NUMBER(2),
71     uren             NUMBER(5,1),
72     CONSTRAINT pk_opdrachten
73         PRIMARY KEY (sofi_nr, proj_nr),
74     CONSTRAINT fk_opd_med
75         FOREIGN KEY (sofi_nr) REFERENCES medewerkers,
76     CONSTRAINT fk_opd_proj
77         FOREIGN KEY (proj_nr) REFERENCES projecten
78 ) ;
```

## Oefeningen

---

# Wijzigen tabelstructuur – verwijderen tabellen

Errors zijn normaal bij DROP (de eerste lijnen uit het txt-bestand) indien de tabellen nog niet bestaan (dan kan je ze ook niet verwijderen ☺ )



```
Script Output x
Task completed in 0,425 seconds
Error starting at line . 6 in Command -
DROP TABLE afdeling
Error report -
SQL Error: ORA-00942: table or view does not exist
00942. 00000 - "table or view does not exist"
*Cause:
*Action:
table AFDELING created.
table MEDEWERKER created.
table PROJECT created.
table OPDRACHT created.
table GEZINSLID created.
table LOCATIE created.
```

## Oefeningen

---

# Wijzigen tabelstructuur – verwijderen tabellen

Dit zou het resultaat moeten zijn (misschien even refreshen)



Bekijk ook het relationeel model vd Ondernemingsdatabase (zie map **Databank scripts** op bb)

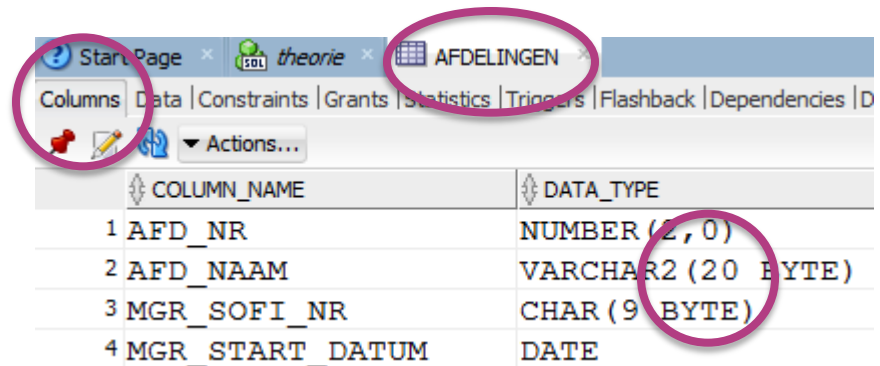
☺ Tip: altijd handig bij de oefeningen om die erbij te nemen.



## Oefeningen

### Wijzigen tabelstructuur – verwijderen tabellen

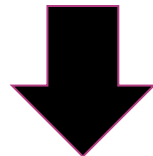
In de tabel AFDELINGEN moet het attribuut AFD\_NAAM verbreed worden naar 25 karakterposities



COLUMN_NAME		DATA_TYPE
1	AFD_NR	NUMBER (2, 0)
2	AFD_NAAM	VARCHAR2 (20 BYTE)
3	MGR_SOFT_NR	CHAR (9 BYTE)
4	MGR_START_DATUM	DATE

*ALTER TABLE afdelingen*

*MODIFY ( afd\_naam VARCHAR2(25) );*



2	AFD_NAAM	VARCHAR2 (25 BYTE)	No
---	----------	--------------------	----

## Oefeningen

---

# Wijzigen tabelstructuur – verwijderen tabellen

In de tabel PROJECTEN moet afgedwongen worden dat de projectnaam in hoofdletters wordt ingegeven

*ALTER TABLE projecten*

*MODIFY proj\_naam CONSTRAINT c\_proj\_naam  
CHECK(proj\_naam=UPPER(proj\_naam));*

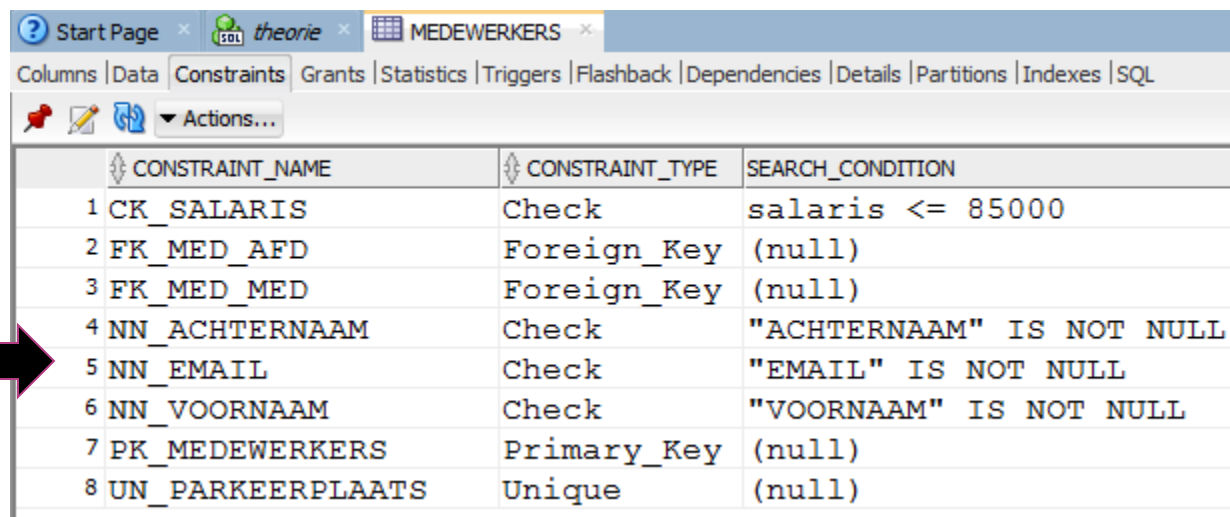
## Oefeningen

# Wijzigen tabelstructuur – verwijderen tabellen

In de tabel MEDEWERKERS moet het attribuut email toegevoegd worden (20 karakters). Het attribuut moet verplicht ingevuld worden.

*ALTER TABLE medewerkers*

*ADD email VARCHAR2(20) CONSTRAINT nn\_email NOT NULL;*



	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION
1	CK_SALARIS	Check	salaris <= 85000
2	FK_MED_AFD	Foreign_Key	(null)
3	FK_MED_MED	Foreign_Key	(null)
4	NN_ACHTERNAAM	Check	"ACHTERNAAM" IS NOT NULL
5	NN_EMAIL	Check	"EMAIL" IS NOT NULL
6	NN_VOORNAAM	Check	"VOORNAAM" IS NOT NULL
7	PK_MEDEWERKERS	Primary_Key	(null)
8	UN_PARKEERPLAATS	Unique	(null)

## Oefeningen

---

# Wijzigen tabelstructuur – verwijderen tabellen

In de tabel AFDELINGEN moet de NOT NULL constraint op de afdelingnaam verwijderd worden

```
ALTER TABLE afdelingen
```

```
DROP CONSTRAINT nn_afd_naam;
```

## Oefeningen

---

# Wijzigen tabelstructuur – verwijderen tabellen

In de tabel MEDEWERKERS moet het attribuut email opnieuw verwijderd worden.

*ALTER TABLE medewerkers*

*DROP COLUMN email;*

## Oefeningen

---

# Wijzigen tabelstructuur – verwijderen tabellen

In de tabel GEZINSLID moet men voor het attribuut GEZ\_GESLACHT BIJKOMEND de waarden v en m toelaten (kleine letters)

*ALTER TABLE gezinsleden*

*DROP CONSTRAINT c\_geslacht;*

*ALTER TABLE gezinsleden*

*MODIFY (geslacht CONSTRAINT c\_gez\_geslacht CHECK (geslacht IN ('M','V','m','v' )));*

C_GEZ_GESLACHT	Check	geslacht IN ('M','V','m','v' )
FK_GEZ_MED	Foreign_Key	(null)
PK_GEZINSLEDEN	Primary_Key	(null)

## Oefeningen

---

# Wijzigen tabelstructuur – verwijderen tabellen

Plaats op de tabel AFDELINGEN een vreemde sleutel naar de tabel MEDEWERKERS;

*ALTER TABLE afdelingen*

*MODIFY mgr\_sofi\_nr CONSTRAINT fk\_afd\_med REFERENCES  
medewerkers(sofi\_nr);*

## Oefeningen

---

# Wijzigen tabelstructuur – verwijderen tabellen

Plaats op de tabel AFDELINGEN een vreemde sleutel naar de tabel MEDEWERKERS;

*Alternatieve oplossing als table constraint:*

*ALTER TABLE afdelingen*

*ADD CONSTRAINT fk\_afd\_med FOREIGN KEY(mgr\_sofi\_nr)  
REFERENCES medewerkers(sofi\_nr);*



## Oefeningen

---

# Wijzigen tabelstructuur – verwijderen tabellen

Verwijder de tabellen die je onder user theorie maakte. Maak **geen** gebruik van CASCADE CONSTRAINTS dus denk aan de volgorde!

## Oefeningen

---

# Wijzigen tabelstructuur – verwijderen tabellen

*DROP TABLE gezinsleden;*

*DROP TABLE locaties;*

*DROP TABLE opdrachten;*

*DROP TABLE projecten;*

*ALTER TABLE medewerkers*

*DROP CONSTRAINT fk\_med\_afd;*

*DROP TABLE afdelingen;*

*DROP TABLE medewerkers;*

## Oefeningen

---

# Wijzigen tabelstructuur – verwijderen tabellen

Alternatief:

```
ALTER TABLE afdelingen
```

```
DROP CONSTRAINT fk_afd_med;
```

```
DROP TABLE gezinsleden;
```

```
DROP TABLE opdrachten;
```

```
DROP TABLE medewerkers;
```

```
DROP TABLE projecten;
```

```
DROP TABLE locaties;
```

```
DROP TABLE afdelingen;
```