

# Data Analysis & SQL

# DDL

# CREATE &

# ALTER table

Koen Bloemen



**DE HOGESCHOOL  
MET HET NETWERK**

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





## Data Definition Language

## 3.1 Schema's en gebruikers

- Schema: logische verzameling databaseobjecten waarmee een gebruiker een connectie kan leggen
  - Tabellen onderdeel van een schema
  - Schema heeft over het algemeen een gebruiker als eigenaar
  - Een-op-een relatie tussen gebruiker en gelijknamige schema's
  - Bij aanmelden → automatisch verbonden met schema

SQL> Create schema authorization STUDENT

Create table medewerkers(...)

Create table afdelingen (...)

....



## 3.2 Tabellen maken

- Creatie van de tabel medewerkers

CREATE TABLE medewerkers			
(mnr	NUMBER(4)	constraint M_PK	primary key
		constraint M_MNR_CHK	check (mnr < 7000)
,naam	VARCHAR2(15)	constraint M_NAAM_NN	not null
,voorn	VARCHAR2(12)	constraint M_VOORN_NN	not null
,functie	VARCHAR2(10)		
,gbdatum	DATE	constraint M_GETDAT_NN	not null
,maandsal	NUMBER(6,2)	constraint M_MNDSAL_NN	not null
,comm	NUMBER(6,2)		
,afd	NUMBER(2)	default 10	
		constraint P_AFD_FK	references afdelingen

kolomnaam

datatype

kolomconstraint (validatieregels)

## 3.2 Tabellen maken

- Creatie van tabel personeel d.m.v. query.  
OPGELET: **constraints worden niet mee gekopieerd!**  
CREATE TABLE personeel  
AS SELECT \*  
FROM medewerkers;
- Creatie van tabel personeel d.m.v. query (een deel wordt overgenomen)  
CREATE TABLE personeelAfdeling30 (pid, pnaam, pvoornaam)  
AS SELECT mnr, lower(naam), voorn FROM medewerkers  
WHERE afd = 30;  
  
Ook hier: **(geen constraints gekopieerd!)**

## 3.2 Tabellen maken

- CREATE TABLE tabelnaam  
  ( kolomnaam        datatype        [kolomconstraint]  
    , kolomnaam        datatype        [kolomconstraint]  
    , ...  
    , [tabelconstraint]  
  ) [AS query]
- Twee componenten
  - Kolomspecificaties
  - Tabelspecificaties
- Mogelijkheid om tabellen te creëren m.b.v. subquery in AS-component (tabel aangemaakt en inhoud gekopieerd)

## 3.2 Tabellen maken

- Naamgeving van tabellen en kolommen
  - Steeds met een letter beginnen
  - Max 30 karakters lang
  - Bevat letters, cijfers en \_
  - Geen verschil tussen kleine letters en hoofdletters
  - Geen gereserveerde woorden

## 3.3 Datatypes

CHAR(n)	alfanumeriek met vaste lengte n
VARCHAR[2](n)	alfanumeriek met variabele lengte van max. n karakters
DATE	datum (4712 BC t.e.m. 4712 AD)
TIMESTAMP	tijdstip (met of zonder tijdzone-informatie)
INTERVAL	tijdsinterval
RAW(n)	binaire gegevens met max. n bytes
NUMBER	geheel getal met een maximale precisie van 38 cijfers
NUMBER(n)	geheel getal van max. n cijfers
NUMBER(n,m)	precisie van n cijfers, waarvan m achter de decimale punt
BINARY_FLOAT	32-bits drijvende komma-getal
BINARY_DOUBLE	64-bits drijvende komma-getal
BLOB	ongestructureerde (binaire) gegevens van max. 4 GB
CLOB	grootte stukken tekst van max. 4 GB



## 3.3 Datatypes

- Opslag van LOB's (Large Objects)
  - BLOB (Binary Large Object)
  - CLOB (Character Large object)
  - BFILE (Binary File)
- Ieder datatype heeft zijn eigen maximale breedte of precisie
  - NUMBER            38 cijfers
  - CHAR                2000 karakters
  - VARCHAR2        4000 karakters

## 3.3 Datatypes

- Voorbeelden:

• Voorletter	char(3)	Ch., P., ...
• Naam	varchar2(25)	Briers , Vos, Slechten, ...
• Getal	number( <b>5</b> )	23856, 99999, ...
• Getal	number( <b>6</b> , <b>2</b> )	6585.50, 9999.99
• Getal	number( <b>6</b> , <b>-1</b> )	748630 (tienvoud)
• Getal	number( <b>6</b> , <b>-3</b> )	749000 (duizendvoud)
• Getal	number	idem number(38,0)
• Getal	number( <b>*</b> , <b>5</b> )	idem number(38,5)

**■** = *total aantal cijfers*

**■** = *aantal cijfers na de komma*

# 3.4 Constraints

- Kolomconstraints
  - Betrekking op één kolom
- Tabelconstraints
  - Constraints betrekking op meer dan één kolom
  - Definiëren van een samengestelde primaire sleutel

	functie = VERKOPER	COMM is NULL		functie = VERKOPER	COMM is NULL	SOM	EINDRESULTAAT
case 1	JA	JA		0	0	0	ONWAAR
case 2	NEE	JA		1	0	1	WAAR
case 3	JA	NEE		0	1	1	WAAR
case 4	NEE	NEE		1	1	2	ONWAAR

```
check(decode(functie,'VERKOPER',0,1)
+ decode(comm,NULL,0,1)) = 1;
```

- Aangeraden een constraint te voorzien van een naam
  - Constraints raadplegen of manipuleren (aanzetten, uitzetten, verwijderen)
  - Namen komen ook voor in foutmeldingen indien ze geschonden worden

## 3.4 Constraints

- Kolomconstraint: (kolomspecificaties)

CONSTRAINT [constraintnaam]

- [NOT] NULL moet (geen) waarde bevatten
- UNIQUE moet uniek, maar niet ingevuld zijn
- PRIMARY KEY primaire sleutel definiëren
- CHECK (voorwaarde) conditie definiëren
- REFERENCES tabelnaam [(kolomnaam)] vreemde sleutel
- [ON DELETE CASCADE] (ouder- en kindrijen verwijderd)
- /[ON DELETE SET NULL] (kindrijen op NULL, indien NULL toegestaan)

constraint M_PK	primary key
constraint M_MNR_CHK	check (mnr > 7000)
constraint M_NAAM_NN	not null
constraint M_VOORN_NN	not null
constraint M_CHEF_FK	references medewerkers
constraint M_GEBDAT_NN	not null
constraint M_MNDSAL_NN	not null



## 3.4 Constraints

- kolomconstraint voorbeeld

```
create table zzz(  
  name varchar2(15) constraint z_name_uniquenotnull unique not null  
);
```

- insert into zzz(name) values('PXL'); *1 row created.*
- insert into zzz(name) values('PXL'); *unique constraint (SYSTEM.Z\_NAME\_UNIQUENOTNULL) violated*
- *insert into zzz(name) values(null); cannot insert NULL into ("SYSTEM"."ZZZ"."NAME")*
- *insert into zzz(name) values(''); cannot insert NULL into ("SYSTEM"."ZZZ"."NAME")*
- *insert into zzz(name) values(' '); 1 row created.*

## 3.4 Constraints

- kolomconstraint voorbeeld

```
alter table medewerkers  
add constraint M_NAAM_UPPER check(naam = upper(naam))
```

!!! Op moment dat de constraint aangemaakt wordt,  
controleert Oracle of de bestaande records voldoen aan die constraint.  
Indien niet, kan je de constraint niet aanmaken!!!

- Update medewerkers set naam = 'caspers' where naam = 'CASPERS';  
*check constraint (SYSTEM.M\_NAAM\_UPPER) violated*
- alter table medewerkers **disable** constraint M\_NAAM\_UPPER;  
*Table altered*
- Update medewerkers set naam = 'caspers' where naam = 'CASPERS';  
*1 row updated*
- alter table medewerkers **enable** constraint M\_NAAM\_UPPER;  
*cannot validate (SYSTEM.M\_NAAM\_UPPER) - check constraint violated*
- alter table medewerkers **drop** constraint M\_NAAM\_UPPER;  
*Table altered*

## 3.4 Constraints

- Tabelconstraint

CONSTRAINT [constraintnaam]

- UNIQUE moet uniek, maar niet ingevuld zijn
- PRIMARY KEY samengestelde primaire sleutel definiëren
- CHECK (voorwaarde) conditie definiëren
- FOREIGN KEY (kolomnaam) REFERENCES ... (idem kolomconstraint)  
vreemde sleutel

```
,constraint U_PK primary key (cursus,begindatum)
```

```
,constraint S_OND_BOV check (ondergrens <= bovengrens )
```

## 3.4 Constraints

- tabelconstraint voorbeeld

```
create table personeel(
    functie varchar2(10)
    , comm number(6,2)
    , constraint m_verk_chk check (decode(functie,'VERKOPER',0,1)
                                   + decode(comm,NULL,0,1) = 1))
```

- insert into personeel(functie, comm) values('VERKOPER', null)  
*check constraint (SYSTEM.M\_VERK\_CHK) violated*
- insert into personeel(functie, comm) values('BOEKHOUDER', null)  
*1 row created.*



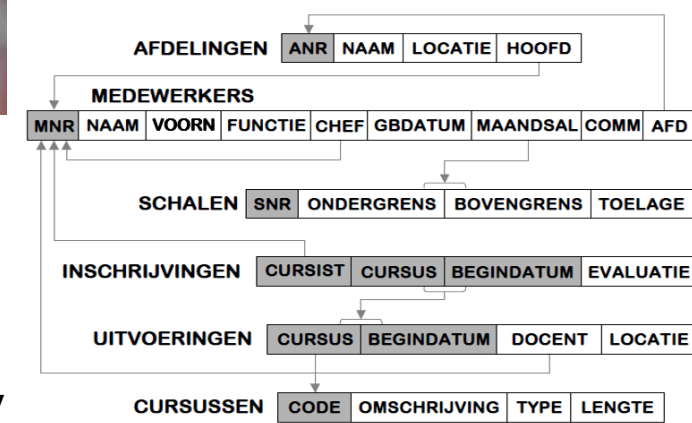
## 3.4 Constraints

- Het is aangeraden constraint-namen betekenisvol te definiëren
  - *Afkorting tabelnaam\_kolomnaam\_soort constraint*
  - Veel gebruikte afkortingen voor *soort constraint*
    - NOT NULL (N)N
    - PRIMARY KEY PK
    - REFERENCES FK
    - UNIQUE UN
    - CHECK CHK

## 3.5 Casustabellen

CREATE TABLE medewerkers

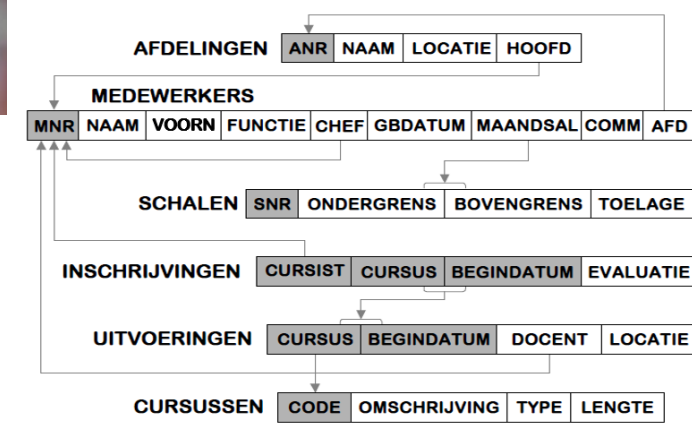
(mnr	NUMBER(4)	constraint M_PK	primary key
		constraint M_MNR_CHK	check (mnr > 7000)
,naam	VARCHAR2(15)	constraint M_NAAM_NN	not null
,voorn	VARCHAR2(12)	constraint M_VOORN_NN	not null
,functie	VARCHAR2(10)		
,chef	NUMBER(4)	constraint M_CHEF_FK	references medewerkers
,gbdatum	DATE	constraint M_GEBDAT_NN	not null
,maandsal	NUMBER(6,2)	constraint M_MNDSAL_NN	not null
,comm	NUMBER(6,2)		
,afd	NUMBER(2)	default 10	
,constraint	M_VERK_CHK	check (decode(functie,'VERKOPER',0,1) + decode(comm , NULL ,0,1) = 1) ;	



## 3.5 Casustabellen

CREATE TABLE afdelingen

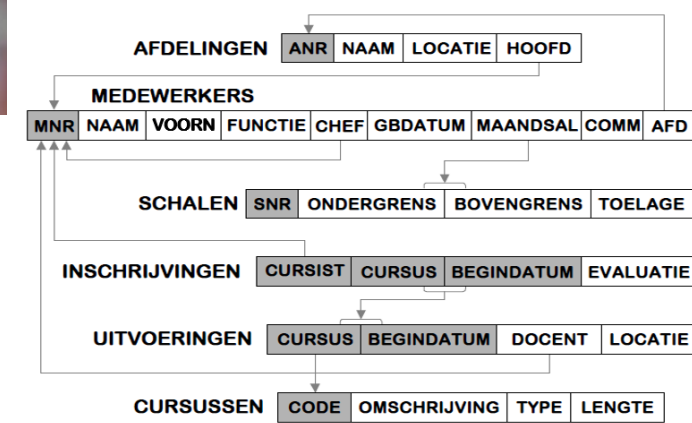
(anr	NUMBER(2)	constraint A_PK	primary key
		constraint A_ANR_CHK	check ( mod(anr,10) = 0 )
,naam	VARCHAR2(20)	constraint A_NAAM_NN	not null
		constraint A_NAAM_UN	unique
		constraint A_NAAM_CHK	check (naam = upper(naam))
,locatie	VARCHAR2(20)	constraint A_LOC_NN	not null
		constraint A_LOC_CHK	check (locatie = upper(locatie))
,hoofd	NUMBER(4)	constraint A_HOOFD_FK	references medewerkers
);			



## 3.5 Casustabellen

ALTER TABLE medewerkers

ADD (constraint M\_AFD\_FK foreign key (afd) references afdelingen);

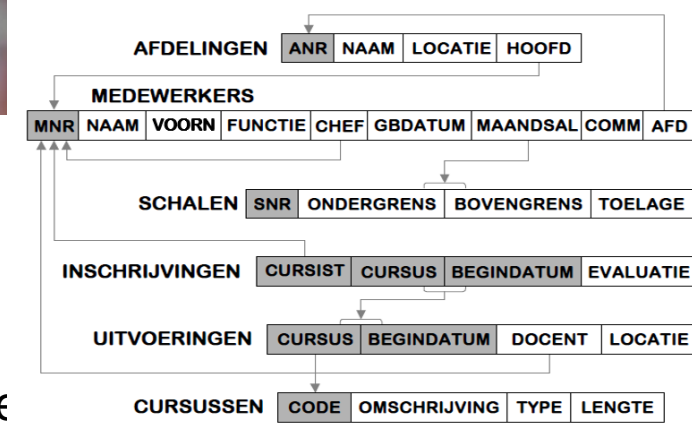




## 3.5 Casustabellen

CREATE TABLE schalen

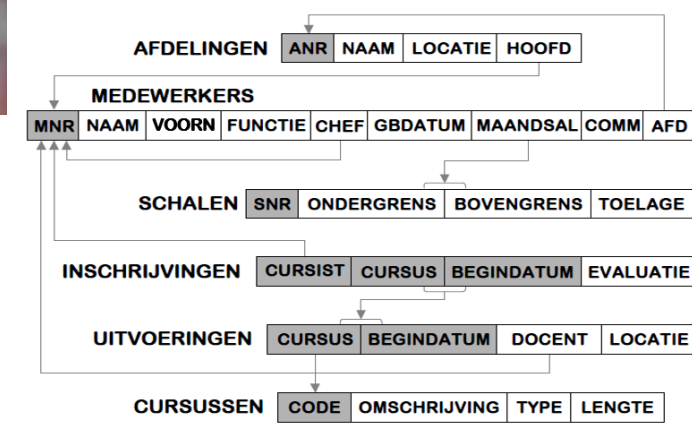
(snr	NUMBER(2)	constraint S_PK	primary key
,ondergrens	NUMBER(6,2)	constraint S_ONDER_NN	not null
		constraint S_ONDER_CHK	check (ondergrens >= 0)
,bovengrens	NUMBER(6,2)	constraint S_BOVEN_NN	not null
,toelage	NUMBER(6,2)	constraint S_TOELG_NN	not null
,constraint S_OND_BOV	check (ondergrens <= bovengrens )		
);			



### 3.5 Casustabellen

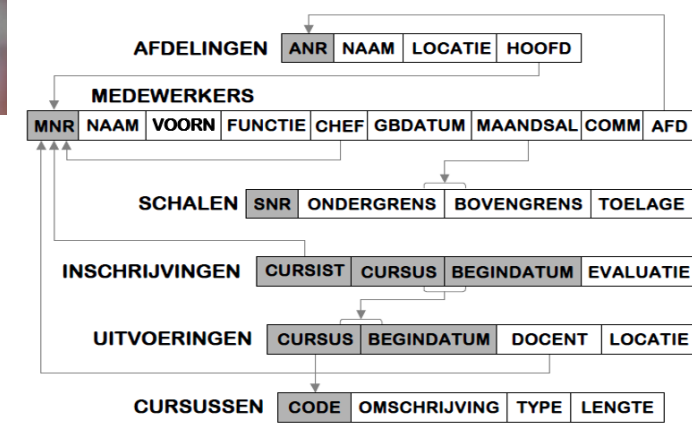
```
CREATE TABLE cursussen
```

```
(code          VARCHAR2(4)    constraint C_PK primary key
,omschrijving  VARCHAR2(50)   constraint C_OMSCHR_NN not null
,type          CHAR(3)        constraint C_TYPE_NN not null
,lengthe       NUMBER(2)      constraint C LENGTE_NN not null
,constraint    C_CODE_CHK     check (code = upper(code))
,constraint    C_TYPE_CHK     check (type in ('ALG','BLD','DSG'))
);
```



## 3.5 Casustabellen

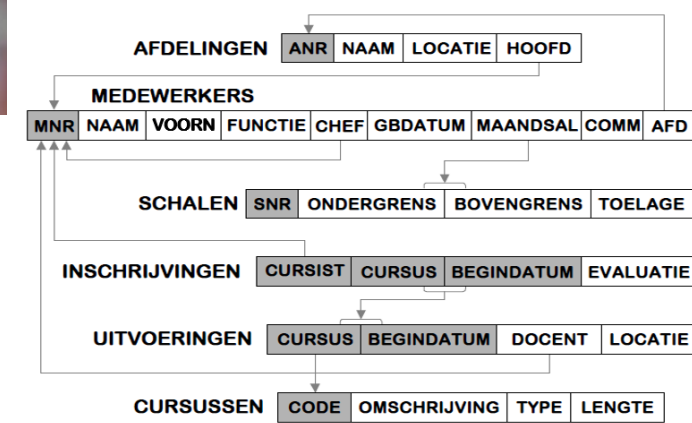
```
CREATE TABLE uitvoeringen
(cursus          VARCHAR2(4)    constraint U_CURSUS_NN not null
                                     constraint U_CURSUS_FK references cursussen
,begindatum     DATE            constraint U_BEGIN_NN  not null
,docent         NUMBER(4)       constraint U_DOCENT_FK references medewerkers
,locatie        VARCHAR2(20)
,constraint U_PK primary key (cursus,begindatum)
);
```



## 3.5 Casustabellen

CREATE TABLE inschrijvingen

```
(cursist          NUMBER(4)          constraint I_CURSIST_NN not null
                                     constraint I_CURSIST_FK references medewerkers
,cursus           VARCHAR2(4)        constraint I_CURSUS_NN not null
,begindatum       DATE               constraint I_BEGIN_NN not null
,evaluatie        NUMBER(1)         constraint I_EVAL_CHK check (evaluatie in (1,2,3,4,5) )
,constraint I_PK  primary key (cursist,cursus,begindatum)
,constraint I_UITV_FK foreign key (cursus,begindatum) references uitvoeringen
);
```





## 5.6 Datadictionary

- Informatie
  - Welke tabellen in de database aanwezig zijn.
  - Welke kolommen ze hebben.
  - Welke privileges ze hebben.
  - Welke sleutels ...
- DATADictionary RAADPLEGEN
- Interne administratie van Oracle
- Gegevens bijgehouden in tabellen → raadplegen met SQL

## 5.6 Datadictionary

- Tabellen in datadictionary opgedeeld in 3 groepen
  - USER\_... informatie over eigen objecten
  - ALL\_... informatie toegankelijk voor de gebruiker
  - DBA\_... alleen toegankelijk voor de databasebeheerder
- De namen geven over het algemeen een duidelijke indicatie van de inhoud van de tabellen
- Voor veel voorkomende tabellen (met lange namen) → synoniemen

## 5.6 Datadictionary

- Synoniemen
  - DICT                      overzicht van alle dictionary tabellen;
  - CAT                      overzicht van onze eigen tabellen, indexen;...
  - TABS                    overzicht van eigen tabellen/synoniemen;
  - COL                    overzicht van de kolommen van onze eigen tabellen
  - USER\_CONSTRAINTS   overzicht van onze eigen constraints
  - USER\_TAB\_PRIVS      overzicht toegekende en verleende privileges
- SQL> select table\_name from tabs;

## 5.6 Datadictionary

- Voorbeeld:

```
SELECT colno, cname, coltype, precision, scale, defaultval, nulls
FROM col
WHERE tname = 'MEDEWERKERS';
```

COLNO	CNAME	COLTYPE	PRECISION	SCALE	DEFAULTVAL	NULLS
1	MNR	NUMBER	4	0		NOT NULL
2	NAAM	VARCHAR2				NOT NULL
3	VOORN	VARCHAR2				NOT NULL
4	FUNCTIE	VARCHAR2				NULL
5	CHEF	NUMBER	4	0		NULL
6	GBDATUM	DATE				NOT NULL
7	MAANDSAL	NUMBER	6	2		NOT NULL
8	COMM	NUMBER	6	2		NULL
9	AFD	NUMBER	2	0 10		NULL

## 5.6 Datadictionary

- Voorbeeld:

```
SELECT constraint_name, constraint_type, search_condition
FROM user_constraints
WHERE table_name = 'MEDEWERKERS'
```

CONSTRAINT_NAME	C	SEARCH_CONDITION
-----	-	-----
M_NAAM_NN	C	"NAAM" IS NOT NULL
M_VOORN_NN	C	"VOORN" IS NOT NULL
M_GEBDAT_NN	C	"GBDATUM" IS NOT NULL
M_MNDSAL_NN	C	"MAANDSAL" IS NOT NULL
M_MNR_CHK	C	mnr > 7000
M_VERK_CHK	C	decode(funcitie,'VERKOPER',0,1) + decode(comm , NULL ,0,1) = 1
M_PK	P	
M_CHEF_FK	R	
M_AFD_FK	R	



## 5.7 Alter table

ALTER TABLE tabelnaam

[ADD (kolomspecificatie|tabelspecificatie)]

[MODIFY(kolomspecificatie)

[RENAME COLUMN oudnaam TO nieuwnaam]

[DROP COLUMN kolomnaam]

ALTER TABLE tabelnaam

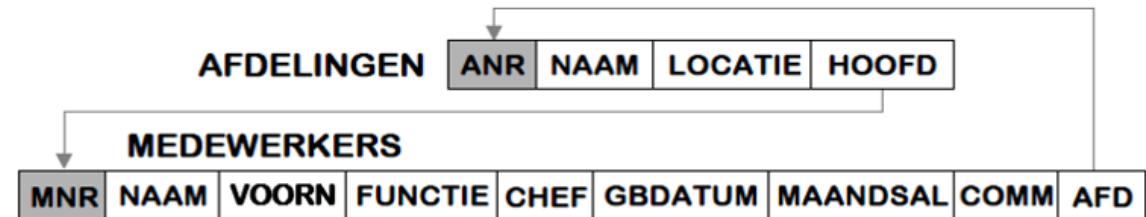
[DROP CONSTRAINT constraintnaam]

[DISABLE CONSTRAINT constraintnaam]

[ENABLE CONSTRAINT constraintnaam]

## 5.7 Alter table

- Toevoegen van een tabelconstraint  
ALTER TABLE medewerkers  
**add** (constraint p\_afd\_fk foreign key (afd) references afdelingen);
- De size van de kolom Afd wordt vergroot en Not Null wordt toegevoegd  
ALTER TABLE medewerkers  
**modify** afd number(5) not null;
- De constraint m\_afd\_fk wordt uitgezet (bij invullen van gegevens)  
ALTER TABLE medewerkers  
**disable** constraint p\_afd\_fk;



## 5.7 Alter table

- **Opmerkingen:**

- Constraints kan je *gedeeltelijk wijzigen*, wel altijd verwijderen en vervolgens toevoegen.
- Voor opties moet je een onderscheid maken tussen een lege tabel of gevulde tabel.
  - Lege tabel
    - Elke alter is toegestaan
  - Gevulde tabel
    - Enkel een NULL-kolom toevoegen (ADD)
    - Enkel de size vergroten en niet verkleinen (MODIFY)
    - Enkel het datatype wijzigen van NOT NULL naar NULL (MODIFY)  
Omgekeerd kan ook, op voorwaarde dat ALLE records een waarde hebben voor die kolom

## 5.7 Alter table

- Voorbeeld:**

1. **SELECT**  
constraint\_name, constraint\_type, search\_condition  
FROM user\_constraints  
WHERE table\_name = 'MEDEWERKERS'
2. **alter table medewerkers disable constraint M2\_AFD\_FK;**  
*Table altered.*
3. **update medewerkers set afd=99 where mnr= 7369;**  
*1 row updated.*
4. **alter table medewerkers enable constraint M2\_AFD\_FK;**  
*cannot validate (SYSTEM.M2\_AFD\_FK) - parent keys not found*

CONSTRAINT_NAME	C SEARCH_CONDITION
M2_CHEF_FK	R
M2_AFD_FK	R
M2_NAAM_NN	C "NAAM" IS NOT NULL
M2_VOORN_NN	C "VOORN" IS NOT NULL
M2_GBDAT_NN	C "GBDATUM" IS NOT NULL
M2_MAANDSAL_NN	C "MAANDSAL" IS NOT NULL
M2_MNR_CHK	C mnr > 7000
M2_VERK_CHK	C decode(funcitie,'VERKOPER',0,1) + decode(comm, NULL, 0,1) = 1
M2_PK	P

