

5 Data Definition Language



5.1 Schema's en gebruikers

5.2 Tabellen maken

5.3 Datatypes

5.4 Constraints

5.5 De casustabellen

5.6 De datadictionary

5.7 Alter table

5.8 Indexen

5.9 Sequences

5.10 Drop table

5.11 Overige commando's

5.12 Oefeningen

5.1 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 (...)
```

```
    ....
```

5.2 Tabellen maken

□ Creatie van 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_VOORL_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 P_AFD_FK	references afdelingen
);			

kolomnaam

datatype

kolomconstraint (validatieregels)

5.2 Tabellen maken

- Creatie van tabel personeel d.m.v. query *(geen constraints 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!)*
→ *update personeel set pid = 9999 werkt perfect*

5.2 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)

Benaming tabellen en kolommen



□ **Naamgeving van tabellen en kolommen (p. 80)**

- Steeds met een letter beginnen;
- Max 30 karakters lang;
- Bevat letters, cijfers en _ ;
- Geen verschil tussen kleine letters en hoofdletters;
- Geen gereserveerde woorden.

5.3 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

5.3 Datatypes

- ❑ BINARY_FLOAT 32-bits drijvende komma-getal
- ❑ BINARY_DOUBLE 64-bits drijvende komma-getal
- ❑ BLOB ongestructureerde (binaire) gegevens van max. 4 GB
- ❑ CLOB grote stukken tekst van max. 4 GB

- ❑ Opmerking: BLOB (Binary Large Object), CLOB (Character large object) en BFILE (Binary File) → opslag van LOB's (Large Objects)

- ❑ Iedere datatype heeft zijn eigen maximale breedte of precisie
 - ❑ NUMBER 38 cijfers
 - ❑ CHAR 2000 karakters
 - ❑ VARCHAR2 4000 karakters


5.3 Datatypes












- Ondersteuning van veel datatypes
- Sommige lijken op elkaar of zijn synoniemen → worden ondersteund vanwege de compatibiliteit van Oracle met andere DBMS
- Voornaamste: NUMBER, VARCHAR2 en DATE

5.3 Datatypes

 = total number of digits

 = number of digits to the right of the decimal point in a number

□ Voorbeelden:

□ Voorletter	char(3)	Ch., P., ...
□ Naam	varchar2(25)	Briers , Vos, Slechten, ...
□ Getal	number()	23856, 99999, ...
□ Getal	number( , )	6585.50, 9999.99
□ Getal	number( , )	748630 (tienvoud)
□ Getal	number( , )	749000 (duizendvoud)
□ Getal	number	idem number(38,0)
□ Getal	number( , )	idem number(38,5)

create table zzz(mnr number(6,-3))

insert into zzz(mnr) values(1256); ➔ 1000

insert into zzz(mnr) values(1556); ➔ 2000

5.4 Constraints

Kolomconstraint

	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

Tabelconstraint

```

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

default 10

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

□ Tabelconstraints

- Constraints betrekking op **meer dan één** kolom
- Definiëren van een **samengestelde** primaire sleutel

□ Kolomconstraints

- Betrekking op **één** kolom

□ 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

5.4 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_CHEF_FK	references medewerkers

5.4 Constraints

□ kolomconstraint voorbeeld

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

■ insert into zzz(name) values('jo');

1 row created.

■ insert into zzz(name) values('jo');

*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.

→ *Oracle treats zero length string(' ' when assigned to a varchar2 column or variable) as NULL*

5.4 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 = 'CASPER';
check constraint (SYSTEM.M_NAAM_UPPER) violated
- ▣ alter table medewerkers **disable** constraint M_NAAM_UPPER;
Table altered
- ▣ Update medewerkers set naam = 'caspers' where naam = 'CASPER';
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

5.4 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 )
```

5.4 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.*

5.4 Constraints

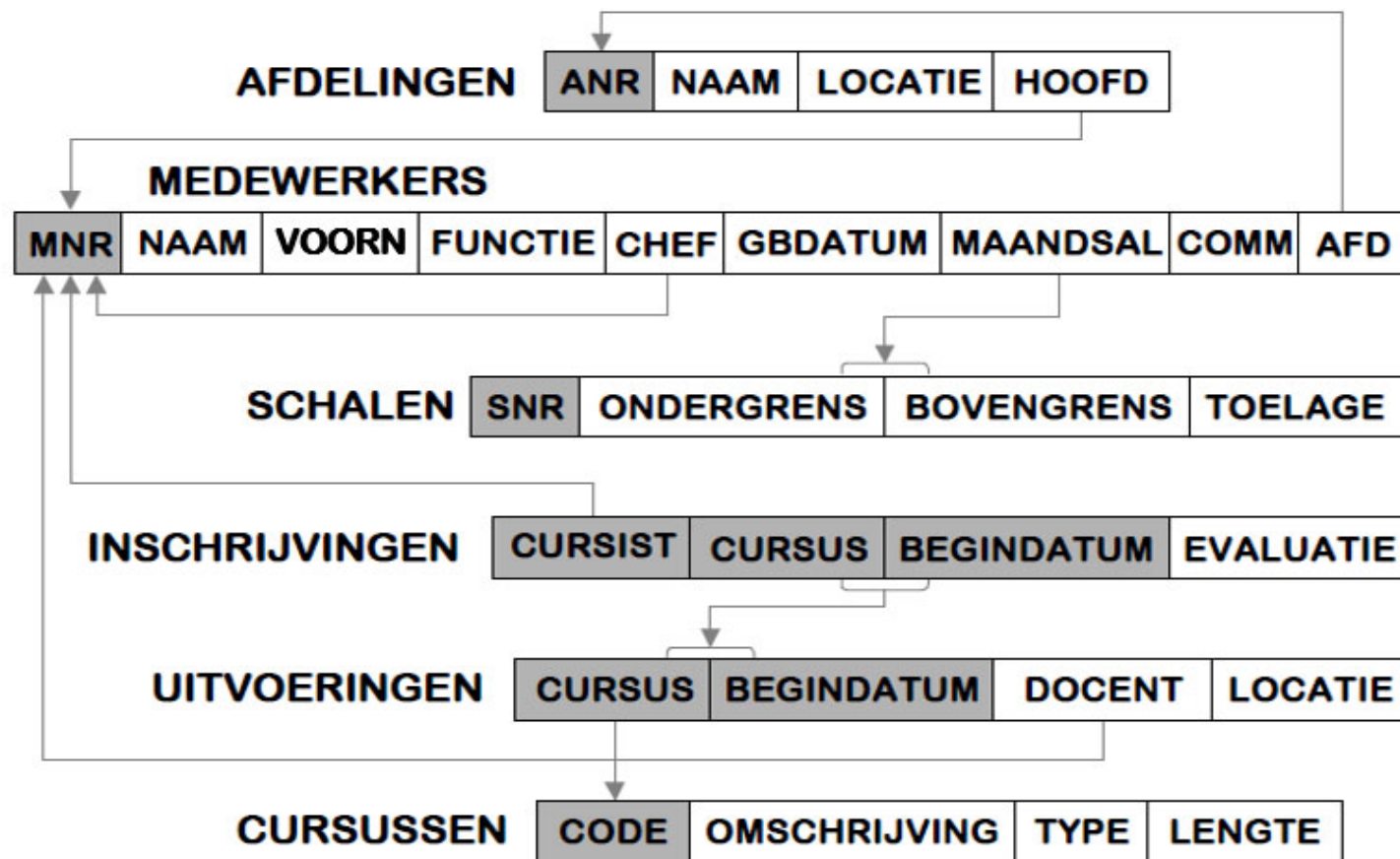
- Het is wenselijk constraint-namen betekenisvol te definiëren

- ▣ *Afkorting tabelnaam_kolomnaam_soort constraint*

- ▣ Veel gebruikte afkortingen voor **soort constraint**

NOT NULL	(N)N
PRIMARY KEY	PK
REFERENCES	FK
UNIQUE	UN
CHECK	CHK

5.5 De casustabellen



5.5 De 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) ;	

5.5 De 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
);			

5.5 De casustabellen

```
ALTER TABLE medewerkers
```

```
ADD (constraint M_AFD_FK foreign key (afd) references afdelingen);
```

5.5 De 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 )
```

```
);
```

5.5 De 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
,lengte       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'))
);
```

5.5 De 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)
);
```


5.5 De 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 De datadictionary

□ Informatie

- ▣ Welke tabellen in de database aanwezig zijn.
- ▣ Welke kolommen ze hebben.
- ▣ Welke privileges ze hebben.
- ▣ Welke sleutels ...

→ DATADictionary RAADPLEGEN

- Datadictionary: interne administratie van Oracle
- Gegevens bijgehouden in tabellen → raadplegen met SQL

5.6 De 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 De datadictionary

□ Synoniemen

- DICT overzicht van alle dictionary tabellen;
- CAT overzicht van onze eigen tabellen, indexen;...
- TAB overzicht van eigen tabellen/synoniemen;
- COL overzicht van de kolommen van onze eigen tabellen
- USER_CONSTRAINTS overzicht van onze constraints
- USER_TAB_PRIVS overzicht toegekende en verleende privileges

□ SQL> select table_name from tabs;

5.6 De datadictionary

□ Bijvoorbeeld:

```
SQL> SELECT colno, cname, coltype, precision, scale, defaultval, nulls
2   FROM col
3   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 De datadictionary

□ Bijvoorbeeld:

```
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(funcnie,'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

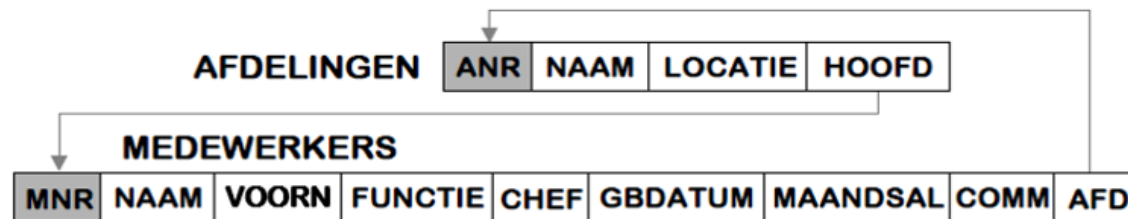
[DROP CONSTRAINT constraintnaam]

[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.

Je kan elk soort Alter doorvoeren bij een lege tabel terwijl bij een gevulde tabel:

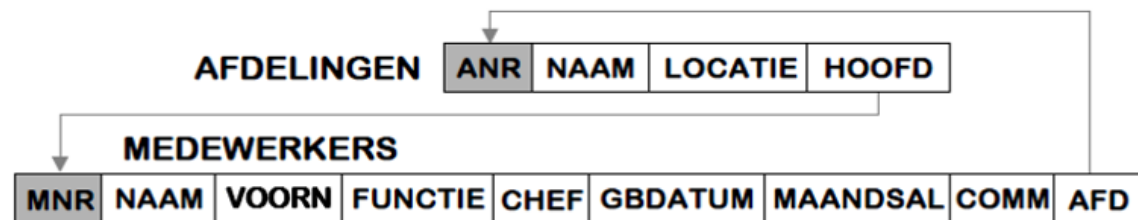
- Enkel een NULL-kolom toevoegen (ADD)
- Enkel de size vergroten en niet verkleinen (MODIFY)
- Enkel het datatype wijzigen van NOT NULL naar NULL (MODIFY)
van NULL naar NOT NULL kan ook, op voorwaarde dat ALLE records een waarde hebben voor die kolom

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_NAAM_NN	C "NAAM" IS NOT NULL
M2_VOORL_NN	C "VOORN" IS NOT NULL
M2_GEBDAT_NN	C "GBDATUM" IS NOT NULL
M2_MAANDSAL_NN	C "MAANDSAL" IS NOT NULL
M2_MNR_CHK	C mn^r > 7000
M2_VERK_CHK	C decode(func^{tie},'VERKOPER',0,1) + decode(comm , NULL ,4
M2_PK	P
M2_CHEF_FK	R
M2_AFD_FK	R



5.8 Indexen

- Index maakt een gesorteerde lijst waarop er snel kan opgezocht wordt.

Vb Medewerkers wiens naam begint met een 'B' kan snel bij een index op naam omdat het gesorteerd kan worden. Het wordt in tegenstelling tot willekeurige volgorde.

Voordelen

- Snel want responstijd korter (geen full table scan nodig)

Nadelen

- Extra opslagruimte
- Datamanipulatie trager omdat wijziging verwerkt moet worden in index.

Index

Symbolen		acties voor afbrengen van normalisatierisultaten	205
* (asterisk)		voor eerste child	209
voor functie COUNT	49	voor eerste parent	209
(in de aanduidingsteken)		active repository's	354
in WHERE-clausule		Active Data Objects (ADO)	414
voor IN en NOT IN	44	objectmodel	414
+ (plus-teken)		ADO (Active Data Objects)	410
voor manipuleren van strings (SQL)	51	objectmodel	414
(pluskomma)		afbrengen van rapporten	512
voor afsluiten van SQL-statements	53	after image	46
(in de context) iederheden		alerts... zie waarschuwingen	358
... (een open-ee relatie)		alleen-lezen databases	
niet identificeerbare verbindingsrelaties		denormaliseren	126
(IDEF/X)	547	normaliseren van	125
wijzingen in 1:N	513	toepassingen voor	125
sterke entiteitsrelaties		normaliseren van	125
relatiemaken	190, 191	ALTERNATIEF	246
1:N (een-op-veel) relaties	547	alternatief... zie. 3e alternatieve sleutels	
niet identificeerbare		alternatieve sleutels	286
verbindingsrelaties (IDEF/X)	547	omzette voor (E/Rwin)	286
recursieve relaties maken 199		ALTER-statement	
sterke entiteitsrelaties maken 191		1:1 relaties wijzingen in 1:N	315
wijzingen in 1:1	312	N-relaties wijzingen in 1:1	317
wijzingen in N:M	318	N-relaties wijzingen in N:M	318
3NF (derde normaalvorm)	95	randvoorwaarden toevoegen/	
2NF (tweede normaalvorm)	95	verwijderen	318
3NF (derde normaalvorm)	95	tabelten/relaties toevoegen	318
4NF (vierde normaalvorm)	96, 110	tabelontbrengen	310
5NF (vijfde normaalvorm)	96	ALTER-statement (SQL)	
		groepsopzet van kolommen wijzingen	314
		kolommen toevoegen/verwijderen	311
A		randvoorwaarden van kolommen	
abstracties (OLE DB)	447	wijzingen	312
Access... zie Microsoft Access			
query's	261	American National Standards	

5.8 Indexen

CREATE [UNIQUE] INDEX indexnaam
ON tabelnaam (kolom1 [ASC | DESC] , [kolom2...])
[TABLESPACE tablespacenaam]

Tip: Indexen worden enkel op bepaalde kolommen ingesteld.

- indexen op vreemde sleutels;
- indexen op kolommen die vaak in de WHERE-restrictie voorkomen;
- indexen op kolommen waarop vaak gesorteerd wordt.

5.8 Indexen

Voorbeeld 1

```
SQL> CREATE INDEX idx_geboortedatum  
2   ON medewerkers (gbdatum);
```

Voorbeeld 2

```
SQL> CREATE INDEX idx_volnaam  
2   ON medewerkers ( naam, voorn)
```

! Volgorde belangrijk want er wordt dadelijk gezocht op eerst naam en dan voornaam (ook wanneer enkel naam wordt opgevraagd).

5.8 Indexen



Voorbeeld 3

```
SQL> CREATE INDEX idx_uppernaam  
2 ON medewerkers (UPPER(naam))
```

Voorbeeld 4

```
SQL> CREATE INDEX idx_jaarsalaris  
2* ON medewerkers (12*maandsal +  
coalesce(comm,0))
```

5.8 Indexen

□ Index opvragen

```
SQL> select index_name, uniqueness,  
table_name  
2 from user_indexes
```

INDEX_NAME	UNIQUENES	TABLE_NAME
U2_PK	UNIQUE	UITVOERINGEN
OP_PK	UNIQUE	OPLEIDING
M2_PK	UNIQUE	MEDEWERKERS
IDX_FAMILIENAAM	NONUNIQUE	MEDEWERKERS
IDX_VOLNAAM	UNIQUE	MEDEWERKERS
IDX_GEBOORTEDATUM	NONUNIQUE	MEDEWERKERS
IDX_JAARSALARIS	NONUNIQUE	MEDEWERKERS
I2_PK	UNIQUE	INSCHRIJVINGEN
C2_PK	UNIQUE	CURSUSSEN
A2_PK	UNIQUE	AFDELINGEN
A2_NAAM_UN	UNIQUE	AFDELINGEN

5.8 Indexen



□ Index verwijderen

We kunnen indexen verwijderen met behulp van het commando **DROP INDEX**.

DROP INDEX indexnaam

5.9 Sequences



Sequences kan je gebruiken om unieke volgnummers te creëren.

Bv. voor werkbonden of factuurnummers die elkaar strikt opvolgen.

- ❑ **CREATE SEQUENCE** sequencenaam
- ❑ **ALTER SEQUENCE** sequencenaam
- ❑ **DROP SEQUENCE** sequencenaam

5.9 Sequences

CREATE SEQUENCE schema_name.sequence_name

[INCREMENT BY interval] *stapgrootte*

[START WITH first_number] *startnummer*

[MAXVALUE max_value | NOMAXVALUE] *maximum*

[MINVALUE min_value | NOMINVALUE] *minimum*

[CYCLE | NOCYCLE] *stapgrootte bij overschrijding max./min.*

[CACHE cache_size | NOCACHE] *geheugenruimte*

[ORDER | NOORDER] *niet van toepassing in deze cursus*

5.9 Sequences

Voorbeeld 1

```
SQL> CREATE SEQUENCE id_seq  
2    INCREMENT BY 10  
3    START WITH 10  
4    MINVALUE 10  
5    MAXVALUE 100  
6    CYCLE  
7    CACHE 5;
```

Sequence created.

CACHE

Specify how many values of the sequence the database preallocates and keeps in memory for faster access. This integer value can have 28 or fewer digits. The minimum value for this parameter is 2. For sequences that cycle, this value must be less than the number of values in the cycle. You cannot cache more values than will fit in a given cycle of sequence numbers. Therefore, the maximum value allowed for **CACHE** must be less than the value determined by the following formula:

5.9 Sequences

Geef de volgende en huidige waarde in een sequence

```
SQL> SELECT id_seq.NEXTVAL, id_seq.CURRVAL  
2      FROM dual
```

Maak de volgende 9 vervolgnummers aan.

```
SQL> SELECT  
2      id_seq.NEXTVAL  
3      FROM dual  
4      CONNECT BY level <= 9;
```

```
      NEXTVAL  
-----  
      30  
      40  
      50  
      60  
      70  
      80  
      90  
     100  
     10 -- Vermits max = 100 + CYCLE geactiveerd, wordt er terug 10 gegenereerd.
```

5.9 Sequences

Voorbeeld 2

```
SQL> CREATE SEQUENCE seq_afd
```

```
2    INCREMENT BY 10
```

```
3    START WITH 50;
```

```
SQL> INSERT INTO afdelingen
```

```
2    VALUES (seq_afd.Nextval,'ADMINISTRATIE', 'ZONHOVEN', 7698)
```

```
SQL> UPDATE medewerkers
```

```
2    SET afd= seq_afd.currval
```

```
3    WHERE mno = 7698
```

5.10 Drop table

❑ DROP TABLE tabelnaam [CASCADE CONSTRAINTS]

- ▣ Tabel wordt uit database verwijderd.
- ▣ Vaak kan een tabel niet verwijderd worden → refererende sleutels vanuit andere tabel.

```
SQL> drop table medewerkers;  
drop table medewerkers  
      *  
ERROR at line 1:  
ORA-02449: unique/primary keys in table referenced by foreign keys
```

- ▣ Oplossing: Cascade Constraints! (alle constraints die van de tabel afhankelijk zijn, worden verwijderd)

```
SQL> drop table medewerkers cascade constraints;  
  
Table dropped.
```

5.1 1 Overige commando's



- TRUNCATE TABLE table

- ▣ Grote tabellen worden op een efficiënte wijze leeggemaakt zonder de tabel te verwijderen.
- ▣ Privileges gaan niet verloren
- ▣ Commando sneller dan het DELETE-commando (zie DML)

5.1 1 Overige commando's

- Tabel een nieuwe naam geven:

`RENAME oude_tabelnaam TO nieuwe_tabelnaam`

- `SQL> RENAME medewerkers TO werknemers;`

5.1 1 Overige commando's



□ CREATE SYNONYM

- ▣ Synoniemen definiëren voor tabelnamen.
- ▣ Toegepast indien tabelnamen erg lang zijn.
- ▣ Handig indien men regelmatig tabellen van andere gebruikers raadpleegt. Anders moet men steeds de naam van de eigenaar voor de tabelnaam specificeren (bv: scott.medewerkers)

5.11 Overige commando's

□ CREATE SYNONYM

- ▣ Creatie van een synoniem med voor de tabelnaam medewerkers
- ▣ CREATE SYNONYM med
FOR medewerkers;

■ SQL> DROP synonym med;

■ SQL> select table_name, synonym_name
from USER_SYNONYMS;

TABLE_NAME	SYNONYM_NAME
MEDEWERKERS	MED
CURSUSSEN	CURS
UITVOERINGEN	UITV
INSCHRIJVINGEN	INS

5.1 1 Overige commando's

□ COMMENT

Verklarende tekst bij tabellen en/of kolommen opslaan in de datadictionary

▣ Commentaar bij de tabel schalen

- COMMENT ON TABLE schalen IS 'salarisschalen en netto toelages';

▣ Commentaar op de kolom COMM van de tabel medewerkers

- COMMENT ON COLUMN medewerkers.comm IS 'alleen voor verkopers';

COMMENT ON

[TABLE tabelnaam | COLUMN kolomnaam] IS 'commentaar'

5.1 1 Overige commando's

- Commentaar opvragen bij de tabellen

- User_tab_comments

- User_col_comments

- Voorbeeld:

- select comments from user_tab_comments
where table_name = 'SCHALEN';

- salarisschalen en netto toelages

- select comments from user_col_comments
where table_name = 'MEDEWERKERS'
and column_name = 'COMM';

- alleen voor verkopers

5.1 2 Script

- Bestand dat zelfstandig uitgevoerd kan worden.

- Vb.

CRECASE2015 en VULCASE2015

- Oefeningen

Script voor TEAMS en KINEPOLIS

NB. In script mag je geen / laten staan van de editor.

@ " pad\crecase2015.sql"