**Verzamelpatronen:**

Select A from C  
UNION/MINUS/INTERSECT/UNION ALL  
0Select B from C

**Union:**Neemt alles van A en B.

**Minus:**Neemt alles van A dat niet in B voor komt

**Intersect:**Neemt alleen wat zowel in A als B voor komt

**Union all:**Plaatst alles uit A & B dus als er 2 de zelfde komen worden er 2 de zelfde geplaatst.

**Views:**

Create [replace]  
View <viewnaam> [kolomalias]  
As <query> [With read only] [with check option]

Select view\_name, tekst  
from user\_views  
Geeft een lijst van al de views in user\_views met query

**de data van de verschillende gegevens in de gegeven view:**Desc <viewnaam>

**Aanroepen view info**  
Select \*  
from <viewnaam>;

**Aanpassingen aan visueel output**

Set long 999  
lengte van kolommen op 999 karakters zetten

Col tekst format a40 word\_wrapped  
tekst terugloop

**View verwijderen**DROP VIEW <viewnaam>

**DML**

**Aanmaken van een table**Create table <naam>

**Verschillende datatypes**

|  |  |
| --- | --- |
| Char(n) | Alfanumeriek met vaste lengte van ‘n’ char (2000 char) |
| Varchar2(n) | Alfanumeriek met niet vaste lengte met max ‘n’ char  (4000 char) |
| Date | datum |
| Timestamp | Een tijdstip zonder tijdzone info |
| interval | tijdsinterval |
| RAW(n) | Binair gegeven van max ‘n’ cijfers |
| Number | Geheel getal van max 38 cijfers |
| Number(n) | Geheel getal van max ‘n’ cijfers |
| Number(n, m) | Geheel getal met ‘n’ cijfers lang met ‘m’ na de komma |
| Binary\_float | 32-bit drijvende komma getal |
| Binary\_double | 64-bits drijvende komma getal |
| BLOB | Ongestructureerde binaire gegevens van max 4GB |
| CLOB | Grootte stukken tekst van max 4GB |
|  |  |
|  |  |

**Handige voorbeelden:**

Getal: number(6,2) = 9999,99

Getal: number(6,-1) = 999990

Getal: number (6, -3) = 999000

Getal: number = idem number(38,0) max mogelijke nummer

Getal: number(\*,5) = idem number (38,5) max mogelijke nummer met 5 getallen na de komma

**Kolom Constraint:**[not] null = moet (geen) waarde bevatten = \_(N)N

Unique = moet uniek maar niet ingevuld zijn = \_UN

Primary key = primaire sleutel = \_PK

Check = conditie definiëren = \_CHK

References = <tabelnaam> [(kolomnaam)] = vreemde sleutel = \_FK

**Tabel Constraint:**Unique = moet uniek maar niet ingevuld zijn = \_UN

Primary key = primaire sleutel = \_PK

Check = conditie definiëren = \_CHK

References = <tabelnaam> [(kolomnaam)] = vreemde sleutel = \_FK

**DataDictionary:**

Synoniemen voor veel voorkomende tabellen

|  |  |
| --- | --- |
| DICT | Overzicht van alle dictionary tabellen |
| CAT | Overzicht van onze eigen tabellen, indexen, … |
| TAB | Overzicht van eigen tabellen/synoniemen |
| col | Overzicht van der kolommen van onze eigen tabellen |
| User\_constraints | Overzicht van onze constraints |
| User\_tab\_privis | Overzicht toegekende en verleende privileges |
|  |  |
|  |  |

**Enkele voorbeelden**

SELECT colno, cname, coltype, precision, scale, defaultval, nulls  
From col  
where tname = ‘MEDEWERKERS’;

Select constraint\_name, constraint\_type, search\_condition  
from user\_constraints  
where table\_name = MEDEWERKERS

**Alter table**

Alter table medewerkers  
Add (constraint p\_afd\_fk foreign key (afd) references afdelingen);

Alter table medewerkers  
Modify afd number(5) not null;

Alter table medewerkers  
Disable constraint p\_afd\_fk;

**Indexen:**

Index aanmaken voorbeelden

Create index idx\_geboortedatum  
on medewerkers (gbdatum);

Create index idx\_volnaam  
On medewerkers (naam, voorn);

Create index idx\_uppernaam  
On medewerkers (upper(naam));

Create index idx\_jaarsalaris  
on medewerkers (12\* maandsal + coalesce(comm,0));

Index verwijderen voorbeelden

Drop index <naam>;

**Sequences:**Increment by <interval> = hoeveel bij tellen  
Start with <eerste nr> = start nummer   
maxvalue <nummer> = maximum nummer  
minValue <nummer> = minimum nr  
Cycle/nocycle = stapgrootte bij overschrijding min/max  
Cache <cache grootte> / nocache = geheugenruimte  
order/ noorder = niet gebruikt in cursus

Voorbeeld:

CREATE SEQUENCE id\_seq   
INCREMENT BY 10   
START WITH 10   
MINVALUE 10   
MAXVALUE 100   
CYCLE   
CACHE 5;

De sequence aanroepen, value vegroten & de current value:  
Select id\_seq.nextval, id\_seq.currval  
from dual;

Aanmaken van meerdere volgnummers:

**Drop table:**Drop table <table naam> [cascade constraints];

Cascade constraints = verwijderd ook de tables afhankelijk van de table die je wilt droppen

**Overige commando’s:**prullenmand bekijken  
select \* from user\_recyclebin;

Table uit prullenmand halen:  
flashback table <naam> to before drom [rename to <name>];

Tabel definitief weg yeeten:  
drop table <naam> purge;

Tabel nieuwe naam geven:  
Rename <oude naam> to <nieuwe naam>;

**Synonym:**aanmaken synonym:  
create synonym med  
for medewerkers;

Verwijderen synonym:  
drop synonym med;

Naam van table en synonym van table opvragen  
select table\_name, synonym\_name  
from user\_synonyms;

**Comment:**commentaar plaatsen op tabel schalen:  
Comment on table <naam> is ‘<comment>’;

Commentaar plaatsen om comm in tabel medewerkers:  
comment on column medewerkers.comm is ‘<comment>’;

Commentaar opvragen in tabel  
select comments from user\_tab\_comments  
where table\_name = ‘schalen’;

Commentaar opvragen kolom:  
select comments from user\_col\_comments  
where table\_name = ‘MEDEWERKERS’  
and column\_name = ‘COMM’;

**DML:**

**Gegevens invoegen:**Insert into <tabelnaam> [(Kolombaam, …)]  
Values (<expressie, …>)

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

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

**Gegevens invoeren dmv substitutievariable:**INSERT INTO personeel VALUES( &mnr,'&naam','&voorn',&functie,&chef, ,to\_date('&gebdatum','dd-mm-yyyy'),&maandsal,&comm,&afd);

**Op basis van een query 4 rijen toevoegen**

Insert into schalen  
Select snr+5, ondergrens +2300, bovengrens +2300, 500  
From schalen  
Where snr <= 4;

**Meerdere kolommen van rij 7666 in tabel wijzigen**update medewerkers  
set functie = ‘TRAINER’,  
maandsal = 5030,  
afd = 20  
where mnr = 766

**Alle maandsallen van medewerkers verhogen:**update medewerkers  
set maandsal = maandsal \* 1.1;

**Loonsverhoging voor iedereen in hoofdkantoor:**update medewerkers  
set maandsal = maandsal \* 1.1  
where afd = (select anr  
 from afdelingen  
 where naam = ‘HOOFDKANTOOR’);

**Gegevens verwijderen:**Delete from medewerkers  
where mnr = 7499;

**ALS JE WHERE VERGEET YEET JE HEEL DE TABEL LEEG!!!!**

**Alles weg yeeten dat met de tabel te maken heeft en de tabel zelf:**DROP TABLE <-> DELETE

**Tabel uit zetten:**Alter table <naam>  
disable constraint med\_afd\_fk

SQL SERVER

Oracle SQL server

|  |  |
| --- | --- |
| Date | Datetime, datetime(2) |
| Interval year(p) to month | Varchar(30) |
| Interval day(p) to seconds | Varchar(30) |
| timestamp | Datetime2(p) |
| Number(p,e) | int |
| number | Float |
| Nvarchar2(n) | Nvarchar(n) |
|  |  |

**Database aanmaken**  
Create datavase <naam>

**Refereren naar een database**  
use <naam>

**Table creeren voorbeelden:**  
create table PROJECT ( projectnr int constraint pr\_pk primary key , projectomschrijving nvarchar(30) not null ,   
budget numeric (7,2) constraint proj\_budg\_df default 5000.00 not null )

create table AFDELING (   
afdeling nchar(5) constraint afdel\_pk primary key ,   
chef nvarchar(20) not null )

create table MEDEWERKER (   
mednr int constraint medw\_pk primary key ,   
naam nvarchar(20) not null constraint medw\_nm\_chk check(naam = upper(naam)) ,   
afdeling nchar(5) references afdeling

create table PROJECTLIJN (   
projectnr int constraint projlijn\_nr\_FK references project ,   
mednr int constraint projlijn\_mednr\_FK references medewerker ,   
uren numeric(5,2) not null , constraint projlijn\_pk primary key(projectnr,mednr) )

**Autonummering direct in de create**  
CREATE TABLE Persoon (   
-- start met 1 en stapgrootte 1   
PersId int IDENTITY(1,1) NOT NULL primary key,   
Naam nvarchar(25) NOT NULL,   
Voornaam nvarchar(15), Leeftijd smallint  
)

**Kolom verwijderen.**   
ALTER TABLE afdeling   
DROP COLUMN chef

**Kolom toevoegen**.   
ALTER TABLE afdeling   
ADD chef nvarchar(20) not null

**Kolom wijzigen.**   
ALTER TABLE afdeling   
ALTER COLUMN chef nvarchar(15)

**Constraint uitschakelen**.   
ALTER TABLE medewerkers   
NOCHECK CONSTRAINT MED\_AFD\_FK

**Constraint inschakelen**.   
ALTER TABLE medewerkers   
CHECK CONSTRAINT MED\_AFD\_FK

**Veilige transactie**SET XACT\_ABORT ON   
BEGIN TRANSACTION   
update saldos set Saldo = saldo - 300 where Voornaam = 'Kevin'   
update saldos set Saldo = saldo + 300 where Voornaam = 'Mia'   
COMMIT TRANSACTION

**SQL SERVER DML**

**Invoeren van gegevens voorbeelden**INSERT INTO afdelingen VALUES (50,'DEPARTEMENT COVID' ,'HASSELT',7782)

INSERT INTO boetes VALUES (4,10045,GETDATE(),2500)

**Voorbeeld gegevens toe voegen in not null kolommen of willekeurige volgorde**INSERT INTO inschrijvingen VALUES (7788,'WBA','2017-02- 24',NULL)  
INSERT INTO inschrijvingen (cursist,cursus,begindatum) VALUES (7934,'WBA','2017-02-24')

**Meerdere gegevens toe voegen**INSERT INTO boetes   
VALUES(5, 90020,getdate(),760.50),   
(6, 10030,'2020-04-01',1500),   
(7, 90020,getdat(),280.30)

**Bulk insert extern bestand**BULK INSERT boetes   
FROM 'C:\SQL Server\bulkinsertboetes.CSV'   
WITH (firstrow = 1, --- Vb. firstrow =2 start op 2de rij (1ste rij = kolomnamen)   
fieldterminator = ';', -- standaard vanuit Excel is het scheidingsteken ;   
rowterminator= '\n' -- einde van lijn )

**Voorbeelden gegevens wijzigen**UPDATE boetes   
SET bedrag = 750.50   
WHERE boetenr = 5

UPDATE spelers   
SET geslacht = default   
WHERE naam = 'Slingers’

**Voorbeelden Gegevens verwijderen**DELETE boetes   
WHERE spelersnr =10020 -- 3 rijen van 10020 verwijderd

DELETE top(3) boetes -- eerste 3 rijen worden verwijderd.

**Transactie verwerking**BEGIN   
transaction DELETE top(3) boetes -- SQL-Commando wordt ongedaan gemaakt.   
ROLLBACK

BEGIN transaction   
DELETE top(3) boetes -- SQL-Commando wordt doorgevoerd.   
COMMIT

**Voorwaardelijke transactieverwerking (bij error word alles terug gebracht**SET XACT\_ABORT ON

BEGIN TRANSACTION   
update medewerkers set mnr = 3000 where naam = 'MARTENS’   
update medewerkers set maandsal = 5000 where naam = 'WOUTERS'   
COMMIT TRANSACTION

**Raadpleging**

**Geef het jaarsalaris (exclusief commissie) van iedere medewerker.**   
select voorn, naam, 12\*maandsal as jaarsalaris   
from medewerkers

**Geef de huidige datum en tijd.**   
select getdate()

**Geef de werknemers geboren na 1 mei 1985.**   
select voorn, naam, gbdatum   
from medewerkers   
where gbdatum >= '1985-05-01'

**Geef de medewerkers van afdeling 10 of afdeling 30.**  
select voorn, naam, afd   
from medewerkers   
where afd in (30,10)

**Geef de medewerkers die werken in afdeling 20 t.e.m afdeling 30**   
select mnr, voorn, naam   
from medewerkers   
where mnr between 7000 and 7700;

**Geef alle medewerkers in wiens naam een 'A' voorkomt. WILDCHARTS idem**   
select voorn, naam   
from medewerkers   
where naam like '\_A%';

**Geef de medewerkers die niet in de afdeling 30 werken.**   
select voorn, naam, afd   
from medewerkers   
where NOT afd = 30;

**Geef de medewerkers van afdeling 10 of afdeling 30.**   
select voorn, naam, afd   
from medewerkers   
where afd= 30 OR afd=10;

**Geef de naam en jaarsalaris (zonder comm) in dalende volgorde van hun jaarsalaris.**   
select voorn + ' '+ naam as naam, maandsal \*12 as salaris   
from medewerkers   
order by salaris desc;  
 -- je kan altijd op een kolomalias sorteren

**Volgorde in functie van kolomnummer 2 (kolom 2 in select-component).**   
select \*   
from medewerkers   
order by 2;

**Volgorde in stijgende volgorde van commissie en dalend voor naam.**   
select \*   
from medewerkers   
order by comm asc ,naam desc;

**CASE**   
select cursist, begindatum ,   
CASE evaluatie   
WHEN 1 THEN 'slecht'   
WHEN 2 THEN 'matig'   
WHEN 3 THEN 'ok'   
WHEN 4 THEN 'goed'   
WHEN 5 THEN 'zeer goed'   
ELSE 'niet ingevuld'   
END as beoordeling   
from inschrijvingen   
where cursus = 'SQL';

SELECT cursist, begindatum ,   
CASE   
WHEN evaluatie = 0 THEN 'Slecht'   
WHEN evaluatie < 2 THEN 'Matig'   
WHEN evaluatie < 5 THEN 'Ok'   
WHEN evaluatie < 7 THEN 'Goed'   
WHEN evaluatie < 8 THEN 'Zeer goed'   
ELSE 'Uitstekend'   
END as beoordeling   
FROM inschrijvingen;

**Geef de medewerkers waarvan de commissie niet van toepassing is.**   
select voorn, naam, comm   
from medewerkers   
where comm is NULL;

**Geeft de gegevens van de eerste 10 inschrijvingen in de tabel.**   
select top(10) \*   
from inschrijvingen

**Geeft de laatste 5 inschrijvingen.**   
select top(5) cursist, cursus as opleiding   
from INSCHRIJVINGEN   
order by cursist desc

**Geeft de eerste 2 inschrijvingen en geeft de overeenkomende**Select top 2 with ties cursist, cursus  
from inschrijvingen  
order by cursist

**Join**  
Select m.naam as MEDEWERKER\_, mm.naam as CHEF\_  
from medewerkers m\_  
inner join medewerkers mm\_  
on m.chef = mm.mnr\_  
where m.gbdatum > ’1985-01-01’;

Algemene SQL server functies

**Geeft NULL als a=b, zo niet geeft het a**SELECT NULLIF(‘SQL’, ‘sql’)  
null

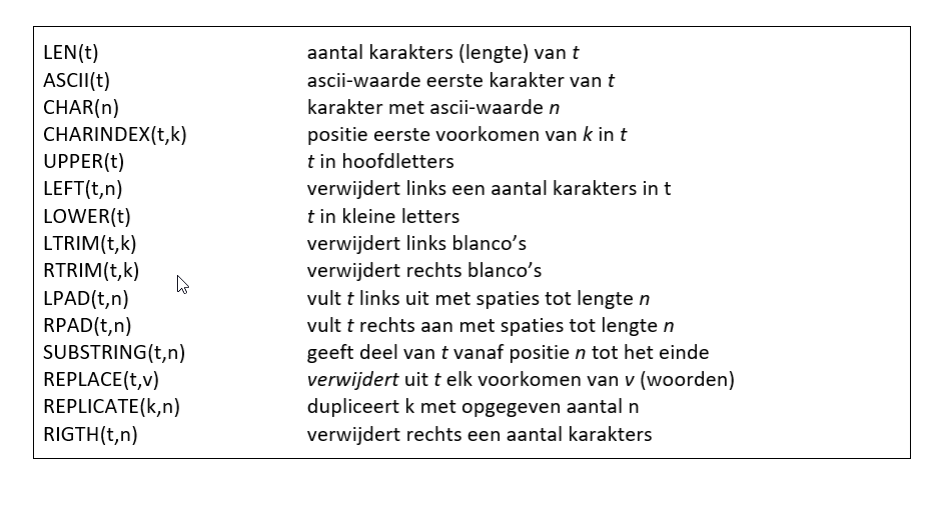
**Retourneert het eerste argument dat niet NULL is**Select naam, comm, (maandsal\*12) + coalesce(comm, 0) as salaris  
from medewerkers  
CASPER NULL 21600

**Retourneert 2de parameter als null is**Select naam, comm, (maandsal\*12) + isnull(comm, 0) as salaris  
from medewerkers  
CASPER NULL 21600

**Je gebruikt % voor modulus in SQL server**

**Rekenfuncties**

Select round(345.178,0) = 346  
select ceiling(345.678) = 346  
select floor(345.678) = 345  
select round(345.678, 2) = 345.68  
select round(345.678, -1) = 350,000  
select ABS(-123), ABS(0), ABS(456) = 123/0/456



Voor functies zie hoofdstuk 7 werken met SQL Server functies