Samenvatting Database Foundations:

# PostgreSQL:

## Datatypes:

* Char(x): Random volgorde van letters/cijfers met lengte x
* Varchar(x): Random volgorde van letters/cijfers met maximum lengte x
* Integer: 4 bytes => getal van -2147483647 t.e.m 2147483647
* Smallint: 2 bytes => getal van -32767 t.e.m 32767
* Date: yyyy-mm-dd
* Numeric(x, y): Getal met lengte x afgerond op y letters na de komma

## Edit CSV:

* Verwijder onnodige kolommen/rijen
* Geef accenten aan letters (weg door character coding, wij altijd UTF-8)
* Check voor header?
* Check voor incomplete data/verwijder
* Getallen met komma? Verander via write options de delimiter naar; => verander , naar . in hele file.
* Verwijder ”” rond getallen

## Wiskunde:

* Modulo: x%y
* Wortel: |/x
* 3de machtswortel: ||/x
* Naar boven afronden: ceiling(x)
* Naar beneden afronden: floor(x)
* Data:
	+ x-y (dagen tussen datum x en datum y)
	+ age(now(), x) (leeftijd voor person met geboortedatum x)
* x < y (x is kleiner dan y)
* x <= y (x is kleiner dan of gelijk aan y)
* x != y (x is niet gelijk aan y)
* Lettervolgorde klein -> groot: aAbBcC…
* Count(x) (telt rijen van kolom x die niet null zijn)
* Sum(x) (telt waardes uit kolom x op die niet null zijn)
* Min(x) (geeft kleinste waarde van kolom x die niet null is)
* Max(x) (geeft grootste waarde van kolom x die niet null is)

## Volgorde (via haakjes fiksen):

1. Vergelijkingsoperatoren =, <, >, …
2. NOT
3. AND
4. OR

## Querry’s:

* Create schema x (naam schema x)
* Create table if not exists a (x y z?, …, constraint I primary key j); (tabelnaam a, kolomnaam x, datatype y en ”not null” z, pk naam I, pk j)
* Select:
* from x
* x as y (berekening x, kolomnaam y (tussen “” voor spaties))
* where:
* x = ’y’/z (y = str, x = int)
* x is/is not null
* like x (output = rijen die x bevatten, % stelt 0 of meerdere letters voor (‘%b‘ is alles wat eindigt met b), \_ is een letter(‘\_b\_‘ is alle 3 letter woorden met b in het midden)
* between x and y (waarde tussen en met x en y)
* x in (y) (waar waarde van kolom x gelijk is aan een van de waardes uit y)
* order by x (automatisch ASC (1,2,3/a,b,c), kan ook nummer van kolom i.p.v naam)
* not x (voorwaarde x)
* x || y (voegt kolom x en y samen, kan ook ‘karakter/tekst’ toegevoegd worden)
* case when x then y

…

else z

end as i (conditie x, output y, output z, kolomnaam i)

* distinct on (x) (zorgt voor geen herhaling van rijen in kolom(men) x)
* extract x from y (year/month/day x, date y)
* to\_char(x, y) (verandert x naar string volgens format y)
* Delete from x (where y)
* Truncate x (verwijdert alle rijen van tabel x)
* Update x set y=z where i (tabel x, kolom y’s waarde veranderdt in z, primary key i)
* Drop x (verwijdert x (x = table y, x = schema y)
* Insert into x values (y),(z),..; (table x en rijen y en z)
* Round(x, y) (getal x aantal kommagetallen y (optioneel))
* Cast(x as y) (data x en datatype y) (cast float as numeric om round te gebruiken)
* Count(x) (kolomnaam x)
* Group by x (kolomnaam x)
	+ having x (soort van where maar voor count(\*) > x enzo)
* Lower(x) (maakt alle letters in x lower case)
* Upper(x)(maakt alle letters in x upper case)
* Substring(x, y (,z)) (string x, beginletter nummer y, lengte z)
* Limit x (aantal rijen van resultaat x)
* Limit x offset y (aantal rijen om over te slaan y)

## Joins:

* x inner join y on x.a = y.b (tabellen x en y met pk a en pk of fk b)
* x left outer join y on x.a = y.b (tabellen x en y met pk a en pk of fk b, de tabel wordt aanguvuld met waarden uit de linker tabel (left) en kolommen uit de rechtertabel krijgen waarde null)
* vb. Alle lectoren die geen coordinator zijn van vak met 6sp :

SELECT l.naam, l.voornaam
FROM opleidingsonderdeel o RIGHT OUTER JOIN lector l
ON o.coordinator = l.lector\_id AND o.studiepunten = 6
WHERE o.studiepunten IS NULL
ORDER BY 1, 2

* x full outer join y on x.a = y.b (tabellen x en y worden indien nodig beiden aangevuld met null waarden waar kolom a van tabel x gelijk is aan kolom b van tabel y)

## Privileges:

* Grant x on y to z (recht x (SELECT/INSERT/UPDATE/DELETE/TRUNCATE/REFERENCES/TRIGGER/ALL PRIVILEGES) (meerdere met x|y|..), tabel/schema y (TABLE tabel/SCHEMA schema), persoon z)
* ALL: alle rechten (behalve DROP).
* SELECT: kolom(men) in een tabel selecteren. Zo goed als altijd ook nodig voor UPDATE en DELETE privileges.
* UPDATE: recht om kolommen in een tabel te updaten.
* DELETE: rijen verwijderen in een tabel.
* CREATE: recht op in een database een nieuw schema te maken, of in een bestaand schema een nieuwe tabel aan te maken.
* USAGE: recht om in een schema toegang te krijgen
* DROP rechten kunnen niet verleent worden op schema
* Voor DELETE rechten moet men ook SELECT rechten hebben
* Revoke x on y from z (neem recht x af van persoon z op tabel/schema y)

## Subquerry’s:

* Voor te werken met resultaat van andere querry x, x tussen haakjes zetten
* Where y > all x (geeft alle waarden terug waar kolomwaarde y groter is dan alle resultaten van subquerry x)
* Where y > any x (geeft alle waarden terug waar kolomwaarde y groter is dan minstens 1 van de resultaten van subquerry x)

# Datamodelleren:

## Conceptueel datamodel:

* Entity relationship diagram
* Entiteiten, relaties en attributen
* Stappenplan:
	+ Entiteittypes verzamelen
	+ Attributen (eigenschappen) bij entiteittypes zetten
	+ Relaties leggen
* DrawIO
* Redundantie: vermijder herhaling
* Verschillende soorten relaties:
	+ Unaire: relatie van 1 tabel met zichzelf
	+ Binaire: relatie tussen 2 tabellen
	+ Ternaire: relatie tussen 3 tabellen
	+ …
* Kardinalitei van een relatie:
	+ Notatie: (min, max)
	+ Minimumkardinaliteit (0 of 1, beslist of de relatie verplicht is of niet)
	+ Maximumkardinaliteit (1 of N, max 1 of meerdere relaties met dit entiteitstype)
* Types relaties:
	+ 1-1 relatie: maximumkardinaliteit is bij beide 1
	+ 1-N relatie: maximumkardinaliteiten zijn 1 en N
	+ N-M relatie: maximumkardinaliteiten zijn N en M
* Verschillende soorten attributen:
	+ Sleutelatrributen: unieke waarden (goed om joins te leggen, wordt onderlijnd)
	+ Afgeleide attributen: attributen die we uit andere attributen kunnen afleiden (ovaal met stippenlijn)
	+ Enkelvoudige attributen: kunnen niet opgesplitst worden in verschillende attributen
	+ Samengestelde attributen: kunnen wel opgesplitst worden in verschillende attributen
	+ Enkelwaardige attributen: kan voor een entiteit max 1 waarde hebben
	+ Meerwaardige attributen: kan meerdere waarden hebben per entiteit (ovaal met dubbele lijn)
	+ Attribuut van een relatie: een attribuut bij een relatie, kann meestal niet bij 1 van de entiteittypes geplaatst worden (enkel bij N-M relaties)
* Verschillende soorten entiteittypes:
	+ Zwakke entiteittypes: kunnen niet bestaan zonder het bestaan van een ander entiteittype (dubbel omlijnd)
	+ Sterke entiteittypes: kunnen wel bestaan zonder het bestaan van een ander entiteittype
* Subtype: entiteittype dat overerft van ander entiteittype en hier aan toevoegt
* Supertype: entiteittype met subtype entiteittype(s)
* Supertype met subtype in ERD: O als subtypes overlappend zijn, D als ze disjunct zijn

## Logisch datamodel:

* DrawIO
* Tabellen met:
* Een kolom per attribuut
* Een rij per entiteit
* Sleutels: unieke waarde, minimaal (in geval van combinatie van meerdere kolommen, als 1 weg valt moet de sleutel niet meer uniek zijn), not null
* Primary key: wordt gekozen uit kanidaatsleutels, 1 per tabel (natuurlijk: uit de tabel genomen, technisch: ordt toegevoegd aan de tabel)
* Foreign key: om relaties te leggen tussen verschillende tabellen
* Conceptueel => logisch:
	+ Entiteittypes worden tabellen (afgeleide en meerwaardige attributen worden niet toegevoegd
	+ Elke tabel krijgt primary key
	+ Voor 1-1 en 1-N relaties foreign keys aanmaken (voor 1-N: primaire key van 1 wordt foreign key bij N, voor 1-1: kies welke primary key wordt overgezet(kijk minimumkardinaliteit))
	+ Voor elke N-M relatie een tussentabel maken

## Fysiek datamodel:

* DBSchema
* Logisch => fysiek:
	+ Tabellen overzetten
	+ Bepaal datatypes voor de kolommen
	+ Geef aan NULL of NOT NULL
	+ Leg constraints op
* Naamgevingsconventies:
	+ Tabel: enkelvoud, lower case, \_ ipv spatie
	+ Kolom: lowercase, \_ ipv spatie, zo kort mogelijk
	+ Primary key: liefst naam id
	+ Foreign key: tabelnaam\_primary key