

Data Management

(CC BY-NC-SA 4.0)

Wim Bertels
Serhat Erdogan

Lesmateriaal

- Cursus (alphacopy):
[Inleiding tot data management](#)
- Elektronisch materiaal beschikbaar via Toledo
- Optioneel (gedetailleerd stappenplan voor SQL): Het SQL Leerboek, R. Van der Lans

Evaluatie

- Examen: 20
 - Schriftelijk (gesloten boek): 10
 - PC (met examenbeperkingen) : 10
- Optioneel Opdracht > gebruik papieren cursus bij het PC examen
- Database Foundations: voorkennis die opnieuw (verdiepend) kan gevraagd worden.
 - Veronderstelde voorkennis: Programming 1 en Computer systems

Extra Aandachtspunten Examen

- Niet kunnen normaliseren, technisch (pk,un,fk) of conceptueel (SDG)
- Fouten tegen de conceptuele verwerking van een select-instructie
- Vergeten te joinen wanneer dit moet
- Niet kunnen werken met transakties
- Data niet kunnen beveiligen (DCL, sec)
- De query cost niet kunnen vergelijken (perf)

Indien u 1 van deze fouten maakt op een examen =
Onvoldoende

Praktisch

- Lessen structuur
 - 2*2u/week (theorie, practica en oefeningen)
(onder voorbehoud)
 - 2u grote groep
 - 2u kleinere groep
- Software
 - SQL: RDBMS : (<https://www.postgresql.org>)
 - SQL: PGAdmin (<http://www.pgadmin.org/>)
 - SQL: <https://dbeaver.io/>
 - SQL: psql, ..
 - GIT
 - ERM: speeltijd is voorbij
 - Bestandsformaten: open ISO standaarden

Kort Onvolledig Overzicht

- Verschillende DBMS paradigma's
- Modellering en implementatie: verbreding
- Basisinstallatie
- Verband programmeren
- SQL verbreding en verdieping
- Beveiliging
- Performantie
- ..
- Zie planning ..

Studie belasting

- 6 ects credits: 150-180 uren
- Contacturen: $10 \times 4 = 40$ uren
- Examen: 4u + 48u voorbereiding
- Zelf studeren: 62 – 92 uren
 - Bv wekelijks 6 uur

Vragen

- Die stel je tijdens de les
- Dit wil niet zeggen dat je geen epost mag sturen bij problemen; maar het antwoord op een inhoudelijke vraag krijg je tijdens de les.
- Als de lessen gedaan zijn, wordt er niet meer op (inhoudelijke) vragen geantwoord.

Een Inleding tot CLA

Wim.bertels@ucll.be
(CC BY-NC-SA 4.0)

CA

Contributor Agreement

- CLA
- CAA

CAA

- Contributor Assignment Agreement

De CAA vereist een toewijzing en daarbij een overdracht van de rechten tot de project eigenaar.

CLA

- Contributor License Agreement

waarbij de CLA een onherroepbare licentie toekent aan de project eigenaar om de bijdrage te gebruiken

CLA in het kort (bijdragen studenten)

- Eigenaarshap (studenten en lectoren)
- Relatie (gezond)
- Juridisch (problemen vermijden)

Concreet

- Impliciet tussen studenten en lectoren (=project eigenaars)
- Zie eventueel het document
[A1_CLA_contributor_license_agreement.pdf](#)
impliciet.

Referenties

- <http://www.contributoragreements.org>

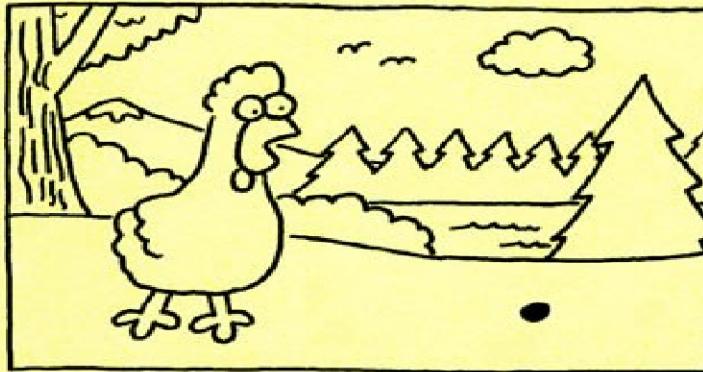
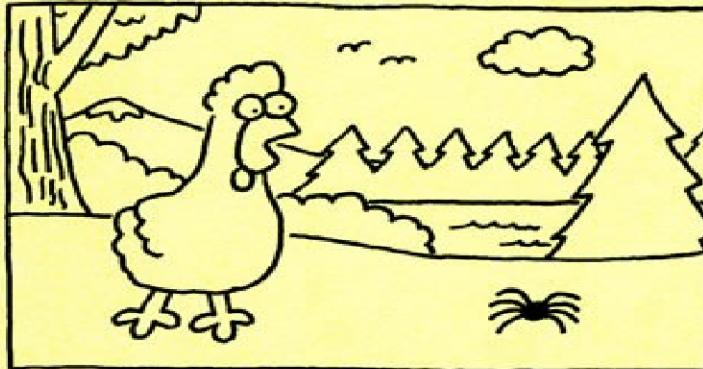
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International Public License

Herhaling ERM

wim.bertels@ucll.be

Naamsvermelding-NietCommercieel-GelijkDelen 4.0 Unported
Licentie

SPOT THE 8 DIFFERENCES BETWEEN THESE TWO PICTURES



© 2011 BY DOUG SAVAGE

www.savagechickens.com

<http://www.savagechickens.com/2011/06/spot-the-differences.html>



	A	B	E	F	I
1	"carrosserienr"	"status"	"merk"	"model"	"beschrijving"
2	"KBE25AZ21	"klaar voor verkoop "	"Renault "	"Trafic "	"nieuwe banden
3	"KBE25AZ21	"klaar voor verkoop "	"Renault "	"Trafic "	"opnieuw gespoten
4	"KBE25AZ21	"klaar voor verkoop "	"Renault "	"Trafic "	"remschijven vervangen
5	"LJB20FK30	"klaar voor verkoop "	"Rover "	"Mini "	"Wielen uitgelijnd
6	"LJB20FK30	"klaar voor verkoop "	"Rover "	"Mini "	"koppelingsplaat vervang
7	"LJB20FK30	"klaar voor verkoop "	"Rover "	"Mini "	"koplampen vervangen
8	"UHDL982HE	"verkocht "	"Mercedes "	"G350 "	"uitlaat veranderd
9	"UHDL982HE	"verkocht "	"Mercedes "	"G350 "	"nieuwe banden
10	"ZZFD73870	"in bewerking "	"Austin "	"Moke "	""
11	"KBE25AZ21	"niet voor verkoop "	"Renault "	"Trafic "	"nieuwe banden
12	"KBE25AZ21	"klaar voor verkoop "	"Renault "	"Trafic "	"opnieuw gespoten
13	"KBE25AZ21	"klaar voor verkoop "	"Renault "	"Trafic "	"remschijven vervangen
14	"KBE25AZ21	"niet voor verkoop "	"Renault "	"Trafic "	"nieuwe banden
15	"KBE25AZ21	"klaar voor verkoop "	"Renault "	"Trafic "	"opnieuw gespoten
16	"KBE25AZ21	"klaar voor verkoop "	"Renault "	"Trafic "	"remschijven vervangen
17	"KBE25AZ21	"niet voor verkoop "	"Renault "	"Trafic "	"nieuwe banden
18	"KBE25AZ21	"klaar voor verkoop "	"Renault "	"Trafic "	"opnieuw gespoten
19	"KBE25AZ21	"klaar voor verkoop "	"Renault "	"Trafic "	"remschijven vervangen

Uitnormaliseren

Redundantie Vermijden
Consistentie Bewaken

Kandidaatsleutels

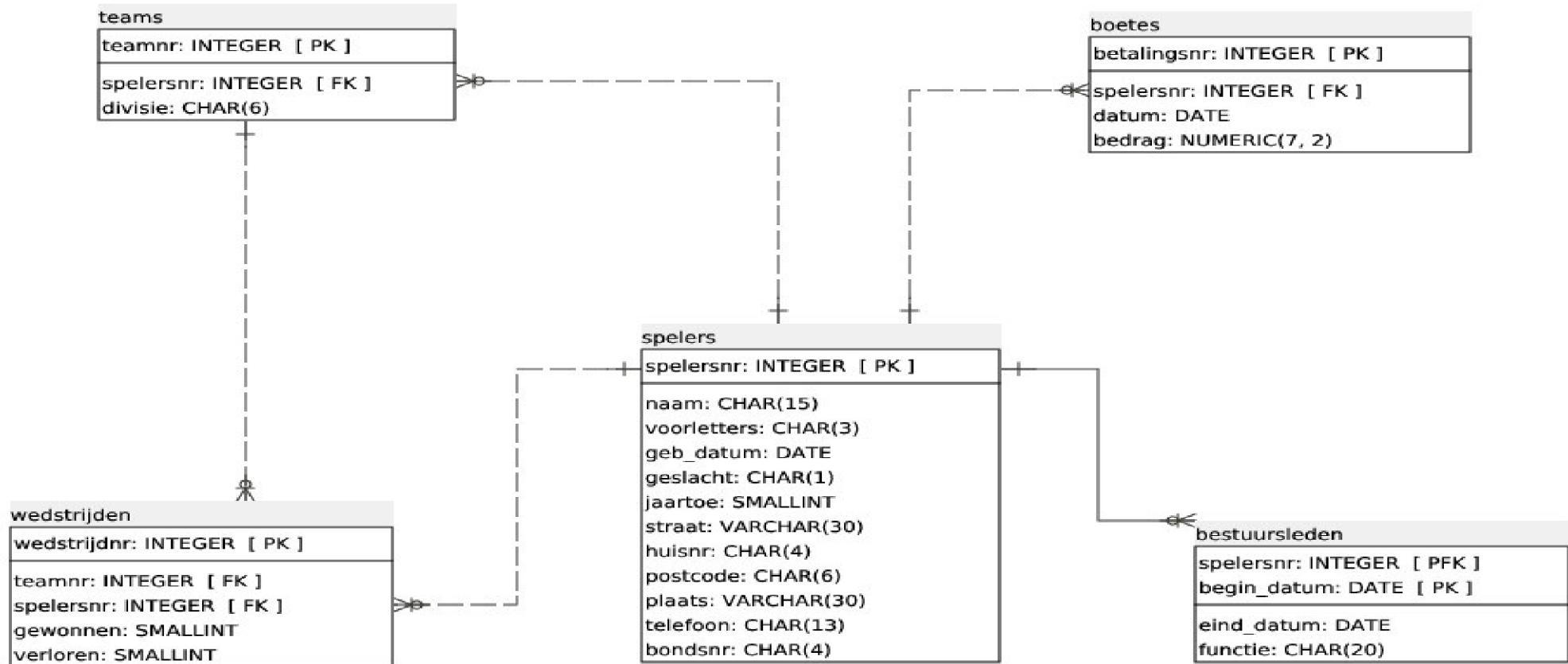
- Minimaal! Anders..?
- Uniek! Anders..?
- Constraints
- Integriteit
- Primary Key is verplicht (NOT NULL)
- Unique is (niet) verplicht

ToID or not to told

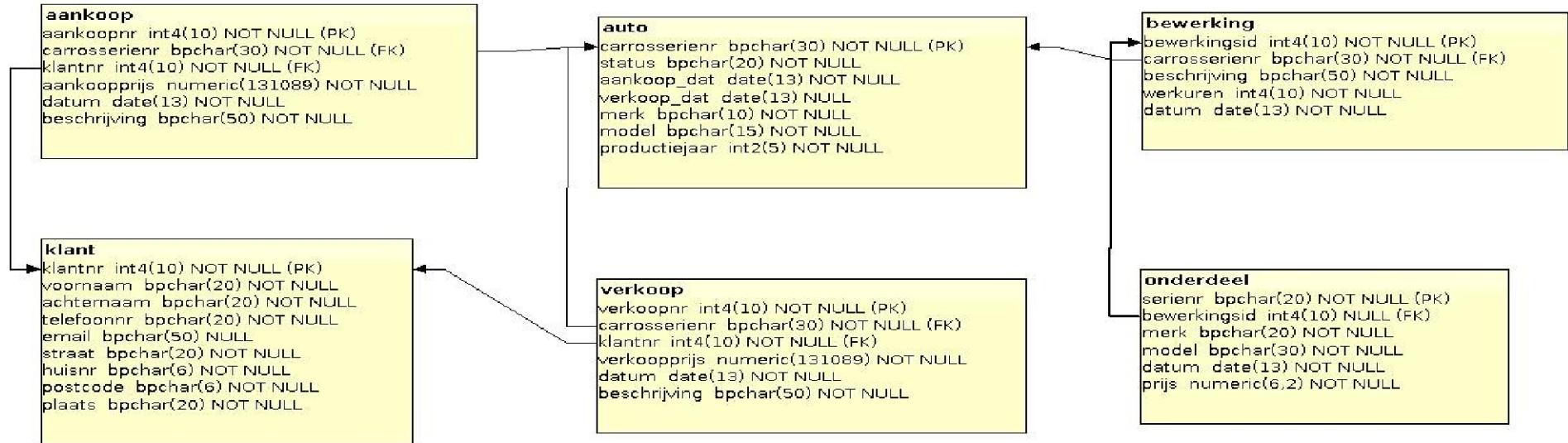
- Reële wereld, virtuele wereld
- Heeft het zin?
- Krijgt deze ID betekenis?
 - Verwijzingen in reële wereld?
 - Als gerefereerde sleutel?
- Of is het een automatisme?
 - Later ORDBMS en frameworks

Vreemde sleutels

- Noodzakelijke, (minimale) vorm van redundantie.
 - Constraints
 - Referentiële integriteit
-
- Foreign Key mag NULL zijn
 - Foreign Key kan verwijzen naar een PK of UN



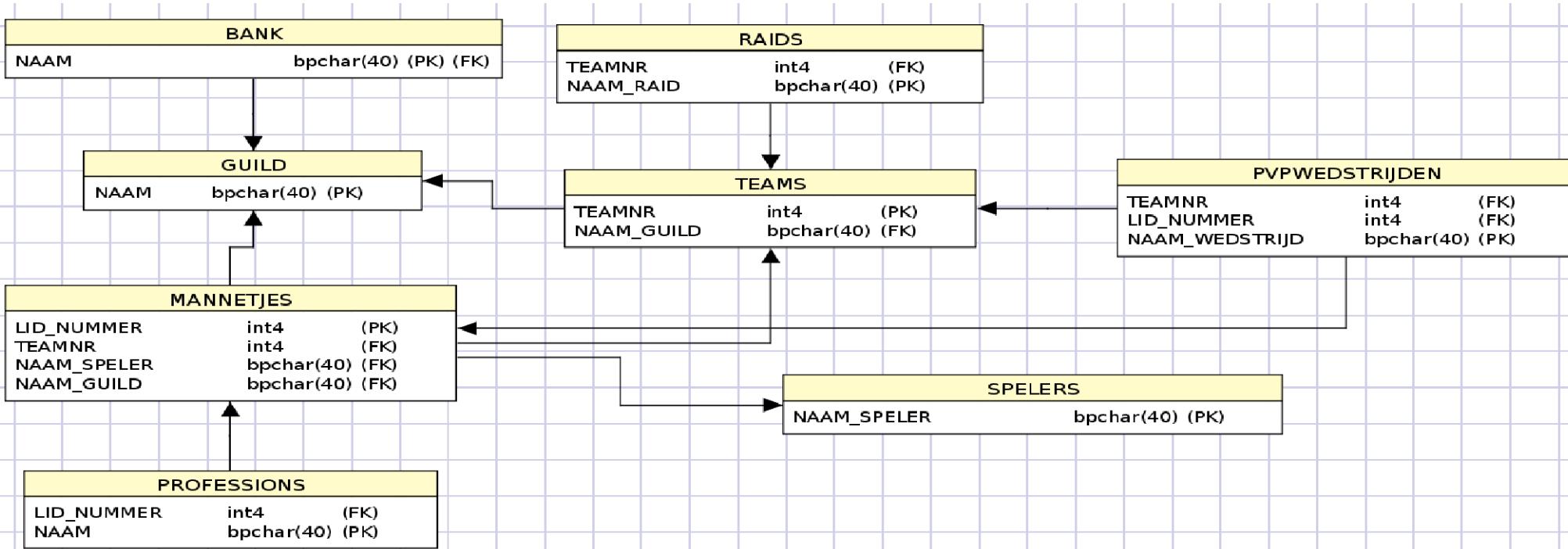
<http://code.google.com/p/power-architect/>



<http://www.squirrelsql.org/>

jdbc:postgresql://databanken.ucll.be:portnr/databasename?ssl=true&sslfactory=org.postgresql.ssl.NonValidatingFactory

<http://jdbc.postgresql.org>



<http://executequery.org/>

ERM

- Conceptueel?
- Logisch?
- Fysisch?
- Praktisch twee stappen om mee te beginnen:
 - analyse
 - Implementatie
- Aantal oplossingen?

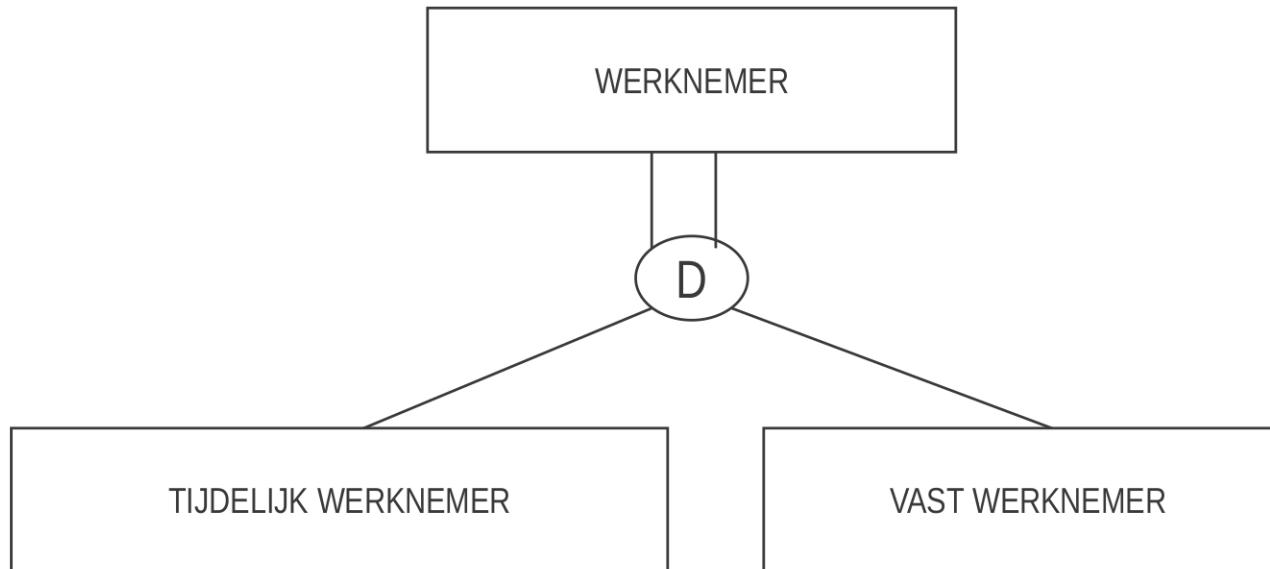
Opgave

Een eenvoudige studentenadministratie. Een student volgt een vak in een reeks.

Probleem: een student mag voor 1 vak maar in 1 reeks ingeschreven zijn.

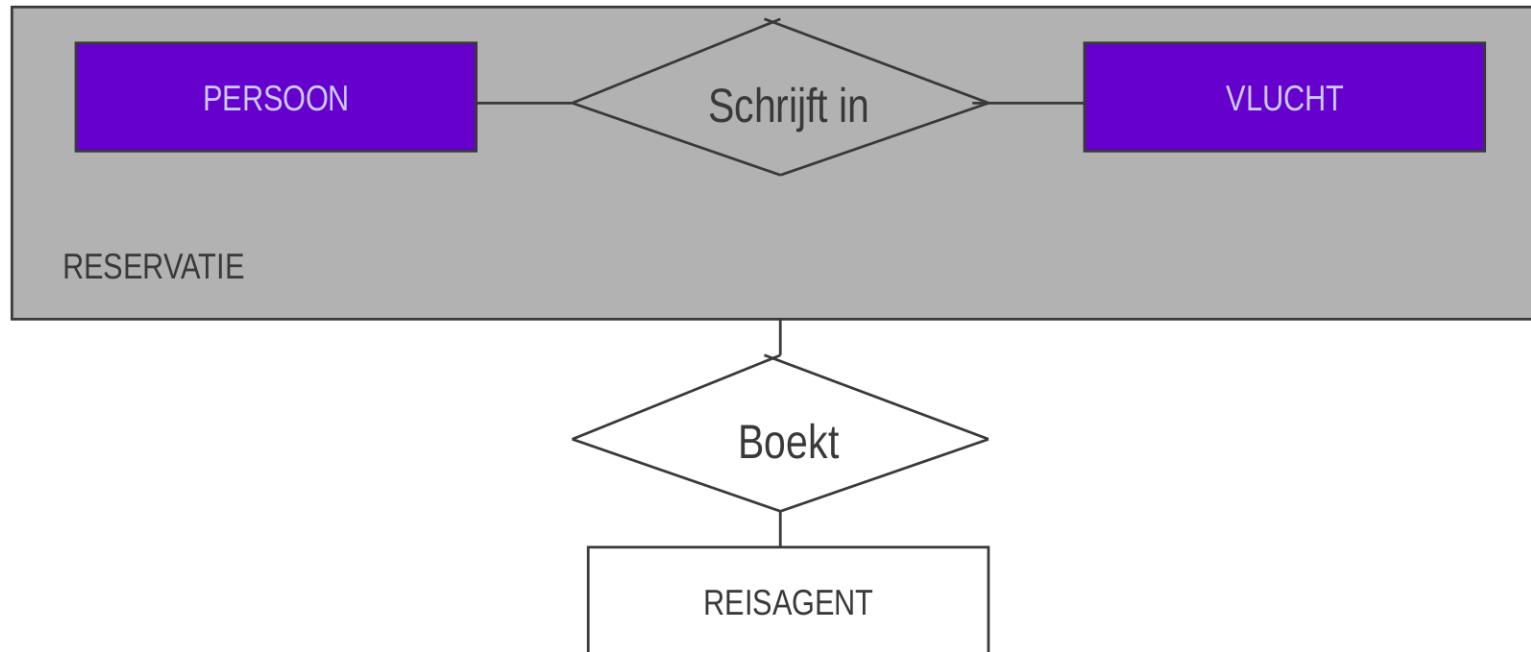
(E)ERM

- 3 mogelijke omzettingen?



(E)ERM

- 1 mogelijk beoogde omzetting?



Herhaling SQL

wim.bertels@ucll.be

Naamsvermelding-NietCommercieel-GelijkDelen 4.0
Unported Licentie

SQL

SQL=Structured Query Language

ISO, no dialects

Lezen van gegevens

SELECT

Muteren van gegevens

INSERT INTO

UPDATE

DELETE FROM

Bepalen structuur voor gegevens

CREATE TABLE

GRANT

Volgorde?

SELECT

FROM

WHERE

GROUP BY

HAVING

ORDER BY

Voorbeeld

```
SELECT      klantrn, MAX(reisnr) AS  
                      laatste_reis  
FROM        deelnames  
WHERE       reisnr > 10  
GROUP BY   klantrn  
HAVING     COUNT(reisnr) > 1  
ORDER BY   klantrn DESC;
```

Functies

EXTRACT ()

CAST ()

TO_CHAR ()

ROUND ()

...

productspecifieke SQL?!

<http://www.postgresql.org/docs/current/interactive/functions.html>

Alias

```
SELECT voornaam AS naam
```

```
SELECT *
FROM     spelers AS s
```

```
SELECT *
FROM     spelers s
```

CASE

CASE

```
WHEN satellietvan IS NULL THEN 'Zon'  
WHEN satellietvan = 'Zon' THEN  
          'Planeet'  
ELSE 'Satelliet'  
END
```

CASE

```
SELECT
CASE
WHEN satellietvan IS NULL THEN 'Zon'
WHEN satellietvan = 'Zon' THEN 'Planeet'
ELSE 'Satelliet'
END AS naamplaneet
FROM hemellichaam
WHERE
CASE
WHEN satellietvan IS NULL THEN 'Zon'
WHEN satellietvan = 'Zon' THEN 'Planeet'
ELSE 'Satelliet'
END
LIKE '%et'
```

Joins

```
SELECT *
FROM    reizen INNER JOIN deelnames
ON reizen.reisnr = deelnames.reisnr
```

LEFT OUTER JOIN

```
SELECT hemelobjecten.objectnaam,  
       bezoeken.reisnr  
  
FROM     hemelobjecten LEFT OUTER JOIN  
        bezoeken  
  
USING (objectnaam)
```

LEFT OUTER JOIN

```
SELECT      s.spelersnr, w.wedstrijdnr
FROM        spelers s LEFT OUTER JOIN
            wedstrijden w
          ON s.spelersnr = w.spelersnr
          AND w.gewonnen > w.verloren
/* condition in de join */
```

LEFT OUTER JOIN

/* versus: */

```
SELECT s.spelersnr, w.wedstrijdnr
FROM spelers s LEFT OUTER JOIN wedstrijden w
    ON s.spelersnr = w.spelersnr
WHERE w.gewonnen > w.verloren
```

/* condition in the where */

LEFT OUTER JOIN

```
SELECT      s.spelersnr  
FROM        spelers s LEFT OUTER JOIN  
            wedstrijden w  
ON          s.spelersnr = w.spelersnr  
AND         w.gewonnen > w.verloren  
GROUP BY    s.spelersnr
```

Selecties op groeperingen

GROUP BY

HAVING

Aggregatiefuncties

COUNT()

MAX()

MIN()

SUM()

AVG()

STDDEV()

<http://www.postgresql.org/docs/current/interactive/functions-aggregate.html>

Oefening 1

Toon alle spelers en het aantal wedstrijden dat ze hebben gespeeld.

Manier 1?

```
SELECT      s.spelersnr, COUNT(*) AS
            aantal

FROM        spelers s INNER JOIN
            wedstrijden w
                    ON s.spelersnr = w.spelersnr

GROUP BY    s.spelersnr;
```

Manier 2?

```
SELECT      s.spelersnr, COUNT(*) AS
            aantal

FROM        spelers s LEFT OUTER JOIN
            wedstrijden w
            ON s.spelersnr = w.spelersnr

GROUP BY    s.spelersnr;
```

Manier 3?

```
SELECT    s.spelersnr, COUNT(w.wedstrijd.nr)
          AS aantal
FROM      spelers s LEFT OUTER JOIN
          wedstrijden w
          ON s.spelersnr = w.spelersnr
GROUP BY s.spelersnr;
```

Toevoegen

```
INSERT INTO bezoeken (
    reisnr,
    volgnr,
    objectnaam,
    verblijfsduur
) VALUES (
    34,
    4,
    'maan',
    2
)
```

Oefening 2

Hoeveel spelers hebben een wedstrijd gespeeld, geef het totaal.

Manier 1

```
SELECT COUNT(DISTINCT s.spelersnr)
      AS aantal
FROM   spelers s INNER JOIN
       wedstrijden w
      ON s.spelersnr = w.spelersnr;
```

Bijwerken

```
UPDATE reizen  
SET vertrekdatum = '2030-12-31',  
    reisduur = 30,  
    prijs = 1.23  
WHERE reisnr = 33;
```

Verwijderen

```
DELETE  
FROM hemelobjecten  
WHERE objectnaam = 'Pluto';  
  
/* zonder where clause?; */
```

Aanmaken

```
CREATE TABLE bezoeken (
    reisnr      numeric(4,0)          NOT NULL,
    objectnaam   character varying(10) NOT NULL,
    volgnr      numeric(2,0)          NOT NULL,
    verblijfsduur numeric(4,0)          NOT NULL,
    CONSTRAINT bezoeken_pkey PRIMARY KEY (reisnr, volgnr),
    CONSTRAINT bezoeken_objectnaam_fkey
        FOREIGN KEY (objectnaam) REFERENCES hemelobjecten (objectnaam),
    CONSTRAINT bezoeken_reisnr_fkey
        FOREIGN KEY (reisnr) REFERENCES reizen (reisnr)
);

-- datatypes zijn productspecifieke SQL?
-- http://www.postgresql.org/docs/current/interactive/datatype.html
```

Rechten geven

```
GRANT      SELECT  
ON         TABLE bezoeken  
TO         student;
```

-- Voldoende?

Rechten geven

```
GRANT    USAGE  
ON        SCHEMA ruimtereizen  
TO        public;
```

Referenties:

- * Slides herhaling sql 2012-13, K. Beheydt
- * SQL Leerboek, R. Van der Lans