

# Inleiding tot databanken

## 4. Normalisatie – Deel 2

*Prof. dr. Paolo Pilozzi*



# Overzicht

Normalisatie is een theoretische en methodologische aanpak ter ontwerp van databanken.

Hoofdstuk 10 & 11 – Oefenzitting 4

HC5 (Deel 1):

- \* Inleiding & Informele Richtlijnen
- \* Functionele Afhankelijkheden en Sleutels
- \* Normaalvormen: NF1, NF2 en NF3

HC6 (Deel 2):

- \* **Herhaling HC5**
- \* Normaalvormen: BCNF (, 4NF en 5NF)
- \* Normalisatie door decompositie

# Functionele Afhankelijkheden

## Definitie functionele afhankelijkheid:

Een **functionele afhankelijkheid (FD)**, genoteerd als  $X \rightarrow Y$ , tussen twee verzamelingen attributen,  $X$  en  $Y$ , beide deelverzamelingen van  $R$ , beschrijft een restrictie op mogelijke tupels dewelke een instantie of databanktoestand  $r$  van  $R$  kunnen vormen, zodat

$$\forall r \text{ of } R \text{ and } \forall t_1, t_2 \in r: t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y].$$

=> Als  $X$  een kandidaatsleutel voor  $R$  is  $\Rightarrow \forall Y \subseteq R: X \rightarrow Y$  in  $R$

=>  $X \rightarrow Y$  in  $R$  zegt niets over  $Y \rightarrow X$  in  $R$

=> Semantische eigenschap v. relatieschema's die beperkt op instanties!  
= Legale toestanden/instanties.

# FDs – Afleidingsregels

$F \models X \rightarrow Y$  noteert dat  $X \rightarrow Y$  kan afgeleid worden uit  $F$ .

Het samenvoegen van attributen  $X$  en  $Y$  wordt genoteerd als  $XY$ .

Basis afleidingsregels (Armstrong's axioms):

AR1: <i>Reflexiviteit</i>	$Y \subseteq X \Rightarrow X \rightarrow Y$	(triviale interferentie)
AR2: <i>Augmentatie</i>	$\{X \rightarrow Y\} \models XZ \rightarrow YZ$	
AR3: <i>Transitiviteit</i>	$\{X \rightarrow Y, Y \rightarrow Z\} \models X \rightarrow Z$	

Bijkomende afleidingsregels afleidbaar uit AR1-3:

AR4: <i>Decompositie</i>	$\{X \rightarrow YZ\} \models X \rightarrow Y$
AR5: <i>Vereniging</i>	$\{X \rightarrow Y, X \rightarrow Z\} \models X \rightarrow YZ$
AR6: <i>Pseudo-transitief</i>	$\{X \rightarrow Y, WY \rightarrow Z\} \models WX \rightarrow Z$

AR1+AR2+AR3 zijn betrouwbaar (sound) en volledig (complete)

=> Exhaustief toepassen v. AR1-3 op  $F$  geeft  $F^+$ , de closure/sluiting van  $F$ .

# FDs – Algoritmisch

Hoe alle FDs (sluiting,  $F^+$ ) voor  $R$  bekomen, gegeven verzameling afhankelijkheden,  $F$ , komende uit de betekenis van relaties en attributen ( $\Rightarrow$  dat wat voor een ontwerper semantisch duidelijk is op basis van de attributen van  $R$ ).

Optie 1: Voor alle FDs in  $F$ : Pas AR1 tot AR3 toe om  $F^+$  te bepalen.

Meer systematisch:

Optie 2: Voor elke FD,  $X \rightarrow Y$ , in  $F$ : Gebruik afleidingsregels om  $X_F^+$  te bepalen, d.w.z. de sluiting van  $X$  t.o.v.  $F$ , of, meer specifiek, de verzameling van alle attributen die functioneel bepaald zijn door  $X$ .

# FDs – Algoritmisch

Hoe  $X_F^+$ , de sluiting van  $X$  t.o.v.  $F$ , berekenen?

$X_F^+ := X$

Herhaal

$OldX_F^+ := X_F^+$

  Voor elke func. afh.  $Y \rightarrow Z$  in  $F$  doe

    Als  $Y \subseteq X_F^+$  dan  $X_F^+ := X_F^+ \cup Z$

Tot ( $OldX_F^+ = X_F^+$ )

*// Initialisatie: Ken aan  $X_F^+$ ,  $X$  toe*

*// Loop (zolang  $OldX_F^+ \neq X_F^+$ )*

*// Hou de oude  $X_F^+$  bij*

*// Zoek FDs om  $X_F^+$  uit te breiden*

*// Als gevonden breidt uit*

*// Als geen uitbreiding, stop*

# FDs – Algoritmisch (Voorbeeld)

$$F = \left\{ \begin{array}{l} \text{Ssn} \rightarrow \{\text{Ename}, \text{Bdate}, \text{Address}, \text{Dnumber}\}, \\ \text{Dnumber} \rightarrow \{\text{Dname}, \text{Dmgr\_ssn}\} \end{array} \right\}$$

Voor  $\text{Ssn} \rightarrow \{\text{Ename}, \text{Bdate}, \text{Address}, \text{Dnumber}\}$ :

- 1:  $\{\text{Ssn}\}_F^+ = \{\text{Ssn}\}$
- 2:  $\{\text{Ssn}\}_F^+ = \{\text{Ssn}, \text{Ename}, \text{Bdate}, \text{Address}, \text{Dnumber}\}$
- 3:  $\{\text{Ssn}\}_F^+ = \{\text{Ssn}, \text{Ename}, \text{Bdate}, \text{Address}, \text{Dnumber}, \text{Dname}, \text{Dmgr\_ssn}\}$

Voor  $\text{Dnumber} \rightarrow \{\text{Dname}, \text{Dmgr\_ssn}\}$ :

- 1:  $\{\text{Dnumber}\}_F^+ = \{\text{Dnumber}\}$
- 2:  $\{\text{Dnumber}\}_F^+ = \{\text{Dnumber}, \text{Dname}, \text{Dmgr\_ssn}\}$

# FDs – Equivalentie

Gegeven verzamelingen  $E$  en  $F$  van FDs van  $R$ , dan

- \*  $E$  wordt overdekt door  $F$  a.s.a.  $E \subseteq F^+$   
d.w.z. elke FD in  $E$  afleidbaar is uit  $F$ . Bijgevolg is  $E^+ \subseteq F^+$ .
- \*  $E$  en  $F$  zijn equivalent a.s.a.  $E^+ = F^+$   
d.w.z.  $E$  overdekt  $F$  ( $E^+ \subseteq F^+$ ) en  $F$  overdekt  $E$  ( $F^+ \subseteq E^+$ ).

Algorithmisch:  $F$  overdekt  $E$  als  $\forall X \rightarrow Y \in E: Y \subseteq X_F^+$



# FDs – Minimaal

Een verzameling FDs,  $F$ , is **minimaal** a.s.a  $F$  voldoet aan:

1. Rechterlid van elke FD in  $F$  bestaat uit slechts 1 attribuut (canonical).
2. Voor geen  $X \rightarrow A$  en  $Z \subset X$  is  $(F \setminus \{X \rightarrow A\}) \cup \{Z \rightarrow A\}$  equivalent met  $F$ .  
=> Geen redundante attributen in afhankelijkheden.
3. Voor geen  $X \rightarrow A$  van  $F$  is  $F \setminus \{X \rightarrow A\}$  equivalent met  $F$ .  
=> Geen redundante afhankelijkheden.

$F$  is een **minimale overdekking** v.  $E$  a.s.a.  $F$  overdekt  $E$  en is minimaal.

# FDs – Sleutels

Gegeven een relatie  $R$  met relatieschema  $S_R = \langle U, F \rangle$ , waarbij  $U$  de set attributen van  $R$  zijn en  $F$  een (minimale) set functionele afhankelijkheden voor  $R$  is. Dan is

$K$  een supersleutel voor  $R$  a.s.a.  $K \subseteq U$  en  $K \rightarrow U \in F^+$ .

=> Een supersleutel determineert alle attributen in  $R$ , of nog  $K_F^+ = U$ .

$K$  een kandidaatsleutel/sleutel voor  $R$  a.s.a.  $K$  is een supersleutel voor  $R$  en er bestaat geen  $K'$  dat een supersleutel is voor  $R$  zodat  $K' \subset K$ .

$A$  is een sleutelattribuut voor  $R$  a.s.a. er een kandidaatsleutel  $K$  voor  $R$  bestaat zodat  $A \in K$ .

# Normaalvormen

Een **normaalvorm** legt eisen op aan relaties

- \* Beperkingen op toegelaten verbanden tussen attributen
- \* Op basis van *functionele afhankelijkheden* en *sleutels*

Opeenvolgend (specialere gevallen of sterkere eisen):

- \* 1NF: eerste normaalvorm
- \* 2NF: tweede normaalvorm
- \* 3NF: derde normaalvorm
- \* BCNF: Boyce-Codd normaalvorm
- \* 4NF: vierde normaalvorm
- \* 5NF: vijfde normaalvorm

# Normaalvormen

**Normalisatie** = relatie in bepaalde normaalvorm brengen via decompositie => relaties die niet voldoen aan de eisen van een normaalvorm opdelen in relaties die er wel aan voldoen.

*Een **goede decompositie** van een relatieschema bevat dezelfde informatie, geen onnodige relaties en is efficiënt te onderhouden.*

Voor een relatie  $R$  noteren we het relatieschema als  $S_R = \langle U_R, F_R \rangle$  met attributenverzameling  $U_R$  en functionele afhankelijkheden  $F_R$ .

**Definitie:** Een verzameling relatieschema's  $\delta(S_R) = \{S_{R1}, S_{R2}, \dots, S_{Rk}\}$  met  $k > 1$  is een decompositie van  $S_R$  a.s.a.  $U_R = U_{R1} \cup U_{R2} \cup \dots \cup U_{Rk}$ .

# 1<sup>e</sup> Normaalvorm (1NF)

Een relatieschema  $S_R = \langle U, F \rangle$  is in de eerste normaalvorm a.s.a. het domein van elk attribuut van U enkelvoudig is. Bij:

- \* Samengestelde attributen:
  - In meerdere attributen onderbrengen.
- \* Meerwaardige attributen:
  - Meerdere tupels + primaire sleutel uitbreiden met attribuut.
  - Nieuwe relatie met verwijssleutel naar (primaire) sleutel orig. rel.

=> Het nieuwe databankschema bevat dezelfde informatie.

=> Er wordt geen enkele eis gesteld aan de functionele afhankelijkheden.

# 2<sup>e</sup> Normaalvorm (2NF)

*Merk op: Voornamelijk v. belang in aanloop naar 3e normaalvorm (3NF).*

\* Een functionele afhankelijkheid  $X \rightarrow Y$  is een triviale functionele afhankelijkheid a.s.a.  $Y \subseteq X$ .

\* Als  $X \rightarrow Y$ , dan zeggen we dat  $Y$  partiëel functioneel afhankelijk is van  $X$  indien er een  $Z \subset X$  bestaat zodat  $Z \rightarrow Y$ ; anders noemen we  $Y$  volledig functioneel afhankelijk van  $X$ .

# 2<sup>e</sup> Normaalvorm (2NF)

*Merk op: Voornamelijk v. belang in aanloop naar 3e normaalvorm (3NF).*

Een 1NF-relatieschema  $S_R = \langle U, F \rangle$  is in de tweede normaalvorm (2NF) a.s.a. voor elke niet-triviale functionele afhankelijkheid  $X \rightarrow A$  in  $F^+$  geldt:  $A$  is een sleutelattribuut of  $X$  is geen strikte deelverzameling van een sleutel van  $R$ .

Of ook: Elk niet-sleutelattribuut in  $R$  is niet partiëel functioneel afhankelijk van een sleutel van  $R$ .

# 2NF – 1NF naar 2NF (Voorbeeld)

<u>Stud_nr</u>	<u>Curs_code</u>	Curs_naam	Inschr_dat	Sc_code	Sc_naam
----------------	------------------	-----------	------------	---------	---------



De sleutel in bovenstaand relatieschema is  $K = \{\text{Stud\_nr}, \text{Curs\_code}\}$ . Volgens de FDs is Curs\_naam partieel afhankelijk van  $K$ , en volledig afhankelijk van  $\{\text{Curs\_code}\} \subset K$ .



<u>Stud_nr</u>	<u>Curs_code</u>	Inschr_dat	Sc_code	Sc_naam
----------------	------------------	------------	---------	---------

<u>Curs_code</u>	Curs_naam
------------------	-----------





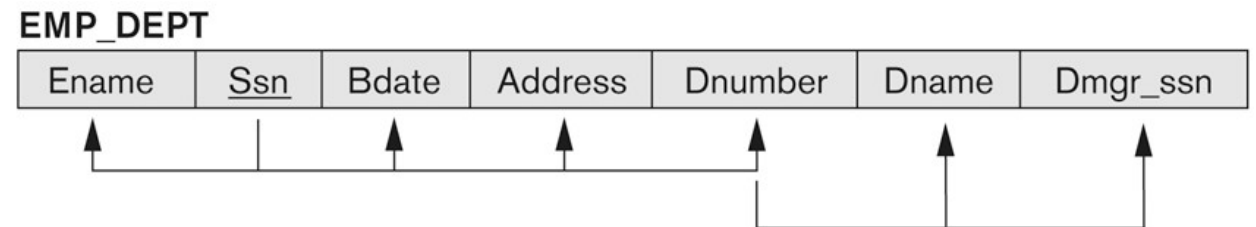
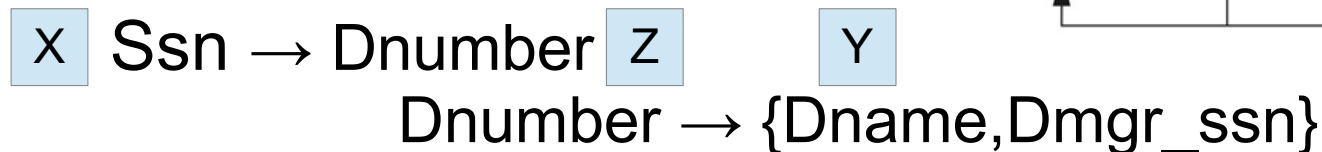
# 3<sup>e</sup> Normaalvorm (3NF)

Een functionele afhankelijkheid  $X \rightarrow Y$  is een *transitieve functionele afhankelijkheid* a.s.a. er een  $Z$  bestaat zodat volgende 3 voorwaarden voldaan zijn:

1.  $Z$  is volledig en niet-triviaal functioneel afhankelijk van  $X$
2.  $Z$  is geen deelverzameling van een kandidaatsleutel
3.  $Y$  is niet-triviaal functioneel afhankelijk van  $Z$

We zeggen dat  $Y$  transitief functioneel afhankelijk is van  $X$  via  $Z$ .

Voorbeeld:

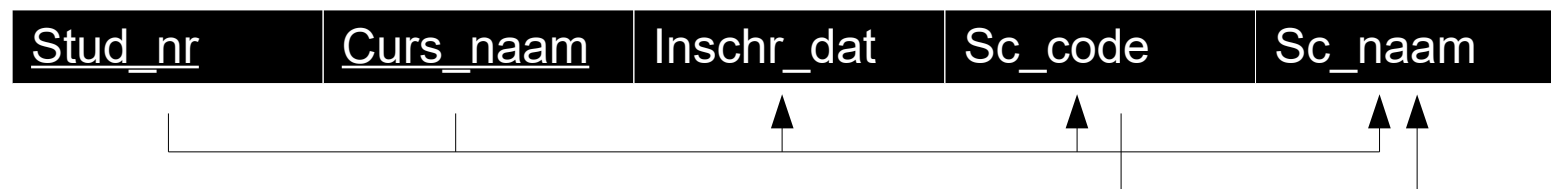


# 3<sup>e</sup> Normaalvorm (3NF)

Een 1NF-relatieschema  $S_R = \langle U, F \rangle$  is in de derde normaalvorm (3NF) a.s.a. voor elke niet-triviale functionele afhankelijkheid  $X \rightarrow A$  in  $F^+$  geldt:  $A$  is een sleutelattribuut of  $X$  is een supersleutel voor  $R$ .

Of ook: Elk niet-sleutelattribuut in  $R$  is niet partieel en ook niet transitief functioneel afhankelijk van een sleutel van  $R$ .

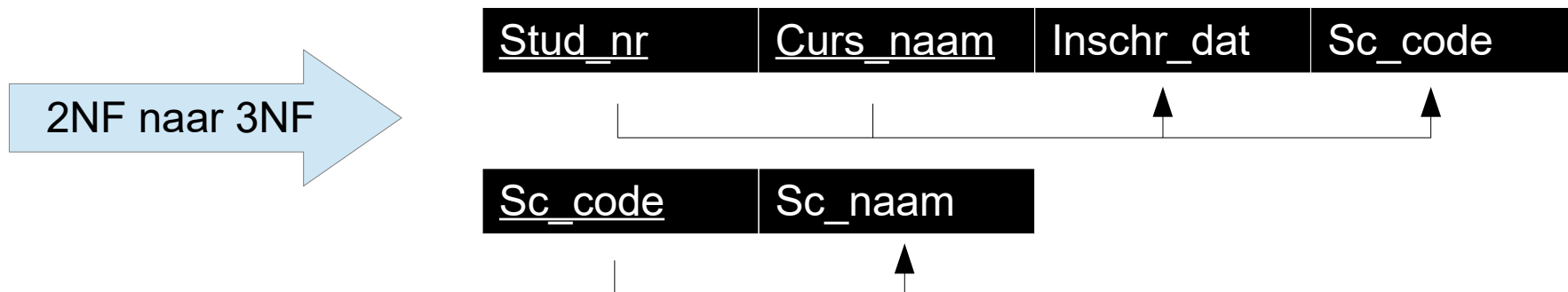
# 3NF – 2NF naar 3NF (Voorbeeld)



Sc\_naam is volledig bepaald door Sc\_code.

Sc\_code bepaald door kandidaatsleutel.

Sc\_naam transitief afhankelijk van kandidaatsleutel.



# Overzicht

Normalisatie is een theoretische en methodologische aanpak ter ontwerp van databanken.

Hoofdstuk 10 & 11 – Oefenzitting 4

HC5 (Deel 1):

- \* Inleiding & Informele Richtlijnen
- \* Functionele Afhankelijkheden en Sleutels
- \* Normaalvormen: NF1, NF2 en NF3

HC6 (Deel 2):

- \* Herhaling HC5
- \* **Normaalvormen: BCNF** (, 4NF en 5NF)
- \* Normalisatie door decompositie

# Boyce-Codd Normaalvorm (BCNF)

Een 1NF-relatieschema  $S_R = \langle U, F \rangle$  is in de Boyce-Codd normaalvorm (BCNF) a.s.a. voor elke niet-triviale functionele afhankelijkheid  $X \rightarrow A$  in  $F^+$  geldt: ~~A is een sleutelattribuut~~ of  $X$  is een supersleutel voor  $R$ .

Of ook: Elk ~~niet-sleutelattribuut~~ in  $R$  is niet partieel en ook niet transitief functioneel afhankelijk van een sleutel van  $R$ .

=> 3NF maar ook op sleutelattributen

# BCNF – 3NF naar BCNF

Gegeven  $S_{R1} = \langle U_{R1}, F_{R1} \rangle$  in 3NF:

\* Voor elk sleutelattribuut A

dat transitief func. afh. is v/e kandidaatsleutel K via Z:

1. Elimineer A uit  $U_{R1}$  alsook de FDs uit  $F_{R1}$  aangaande A:  $F_A$ .

2. Maak een nieuw relatieschema  $S_{R2}$

met attributenverzameling  $U_{R2} = Z \cup A$

en  $F_{R2} \subseteq F_A$  die betrekking hebben op  $U_{R2}$ .

=> Zoals NF2-naar-NF3 maar dan voor sleutelattributen.

# BCNF – Ideaal?

## Pro's:

- \* Partiele & transitieve FDs zijn weggewerkt.

## Contra's:

- \* Opgepast voor informatieverlies!
- \* Sommige FDs worden moeilijk te controleren (DBMS).
  - FDs over relaties heen.
  - Na decompositie, JOIN nodig.

# BCNF – Ideaal? (Voorbeeld)

Stel: Inschr\_dat bepaalt eenduidig Curs\_code

*Verskillende inschrijvingsdata voor elke cursus.*

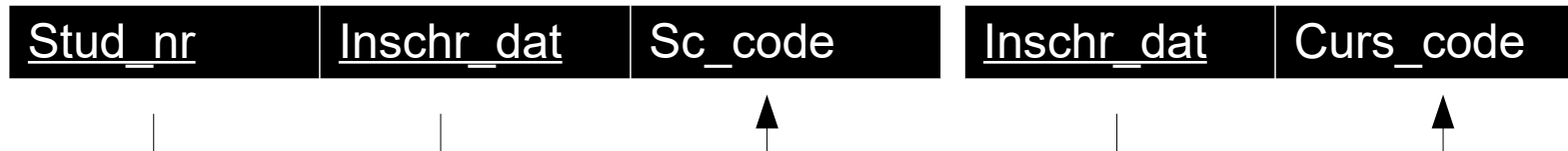
**Kandidaatsleutels:**

{Stud\_nr, Curs\_code}

{Stud\_nr, Inschr\_dat}



Schema is in 3NF maar niet in BCNF! => Decompositie/Opsplitsing:



Redundantie verholpen maar de functionele afhankelijkheid

{Stud\_nr, Curs\_code} → {Inschr\_dat, Sc\_code} moeilijker te controleren.

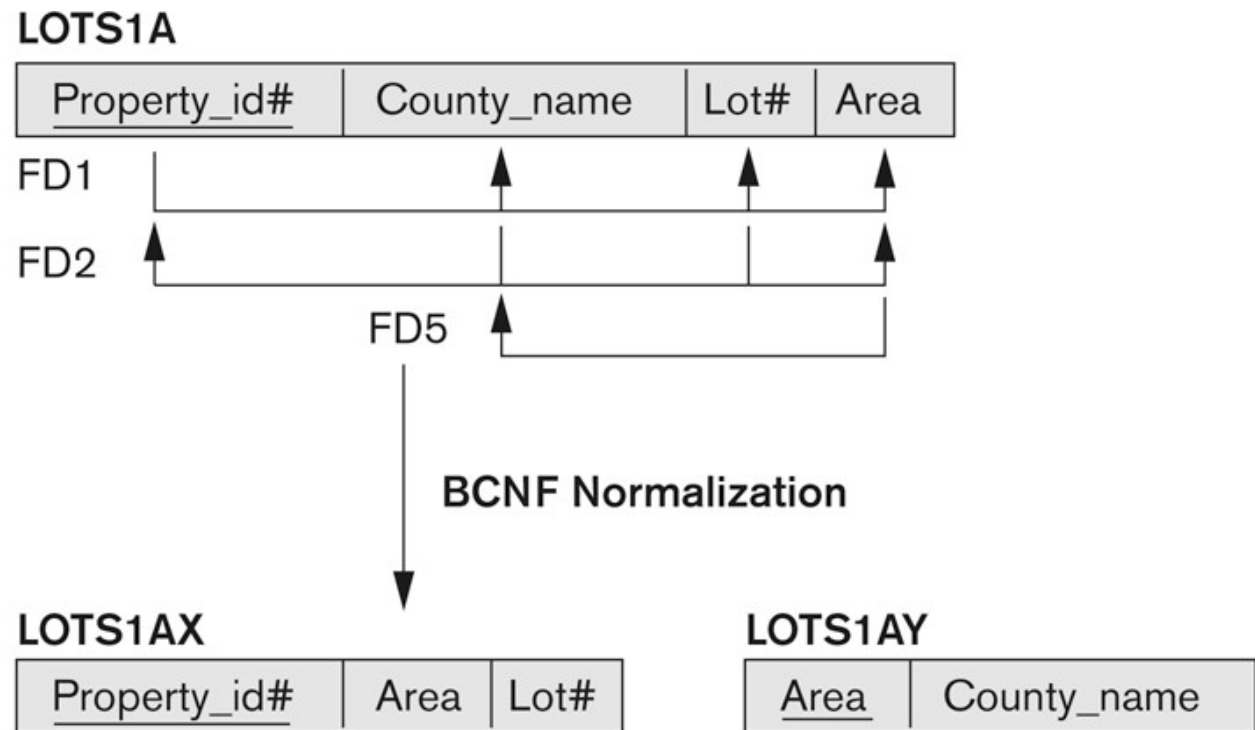


# BCNF – Ideaal? (Voorbeeld)

Kandidaatsleutel: {County\_name, Lot#}

**FD5**: Oppervlakte  
Bepaalt County

FD2 moeilijker te  
Controleren in BCNF  
normaalvorm



# Overzicht

Normalisatie is een theoretische en methodologische aanpak ter ontwerp van databanken.

Hoofdstuk 10 & 11 – Oefenzitting 4

HC5 (Deel 1):

- \* Inleiding & Informele Richtlijnen
- \* Functionele Afhankelijkheden en Sleutels
- \* Normaalvormen: NF1, NF2 en NF3

HC6 (Deel 2):

- \* Herhaling HC5
- \* Normaalvormen: BCNF (, 4NF en 5NF)
- \* **Normalisatie door decompositie**

# Normalisatie door decompositie

Twee benaderingen:

## 1. Via hoog-niveau modelering

- \* ER-schema opstellen en omzetten naar databankschema.
- \* Partiele en transitieve afhankelijkheden wegwerken.

## 2. Meteen relationeel databankschema bouwen

- \* Universeel schema opstellen (met alle attributen).
- \* Technieken, algoritmes voor decompositie toepassen.

# Normalisatie door decompositie

Directe methode  $\Rightarrow$  Vertrek van  $S_R = \langle U_R, F_R \rangle$

Bekom:

$\delta(S_R) = \{S_{R_1}, S_{R_2}, \dots, S_{R_k}\}$  ( $k > 1$ ) is een decompositie zodat:

- \* Elke  $R_i$  is in BCNF of 3NF,
- \* alle informatie behouden blijft, en
- \* alle functionele afhankelijkheden bewaard blijven.

$\Rightarrow$  Laatste twee voorwaarden zijn niet zo evident!

# Normalisatie door decompositie

Ideale decompositie?

1. Alle relaties in BCNF
2. Informatiebewarend (Verliesloos)
3. Afhankelijkhedenbewarend

=> Slechts twee van de drie altijd mogelijk.

# Normalisatie door decompositie

We gaan uit van een decompositie die afhankelijkheden moet bewaren.

=> Functionele afhankelijkheden zijn intra-relationaleel.

=> Na decompositie intra -en inter-relationaleel.

\* Intra-relationaleel:  $\pi_{U_{R_i}}(F_R)$ , projectie  $F_R$  op  $U_{R_i}$

Dus alle  $X \rightarrow Y$  in  $F_R^+$  waarvoor  $(X \cup Y) \subseteq U_{R_i}$

\* Inter-relationaleel is moeilijker te controleren door DBMS (JOIN)

# Normalisatie door decompositie

We gaan uit van een decompositie die afhankelijkheden moet bewaren.

Een decompositie  $\delta(S_R) = \{S_{R_1}, S_{R_2}, \dots, S_{R_k}\}$  van  $R$  bewaart de afhankelijkheden van  $R$  indien

$$(\pi_{UR_1}(F_R) \cup \pi_{UR_2}(F_R) \cup \dots \cup \pi_{UR_k}(F_R))^+ = F_R^+$$

# Normalisatie door decompositie

Ideale decompositie?

*Afhankelijkhedenbewarend.*

1. Alle relaties in ~~BCNF~~ 3NF
2. ~~Informatiebewarend (Verliesloos)~~
3. Afhankelijkhedenbewarend



# Normalisatie door decompositie

## Algoritme Afhankelijkhedenbewarende Decompositie

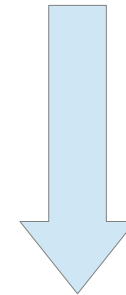
1. Zoek minimale overdekking  $E$  van  $F$
2. Voor elke  $X$  in  $E$  aan de linkerzijde van een FD:  
Maak schema  $\{X \cup A_1 \cup A_2 \cup \dots \cup A_m\}$  (prim. sleutel  $X$ )  
waarbij  $X \rightarrow A_1, X \rightarrow A_2, \dots, X \rightarrow A_m$  alle afhankelijkheden  
in  $E$  met  $X$  als linkerlid
3. Plaats overblijvende attributen in één enkel relatieschema

# Normalisatie door decompositie

## Algoritme *Afhankelijkhedenbewarende Decompositie*

<u>Emp_ssn</u>	<u>Pno</u>	Esal	Ephone	Dno	Pname	Plocation
----------------	------------	------	--------	-----	-------	-----------

### Afhankelijkheden F

 $\{\text{Emp\_ssn}\} \rightarrow \{\text{Esal}, \text{Ephone}, \text{Dno}\}$ 
 $\{\text{Pno}\} \rightarrow \{\text{Pname}, \text{Plocation}\}$ 
 $\{\text{Emp\_ssn}, \text{Pno}\} \rightarrow \{\text{Esal}, \text{Ephone}, \text{Dno}, \text{Pname}, \text{Plocation}\}$ 


### Minimale overdekking E

 $\{\text{Emp\_ssn}\} \rightarrow \{\text{Esal}\}$ 
 $\{\text{Emp\_ssn}\} \rightarrow \{\text{Ephone}\}$ 
 $\{\text{Emp\_ssn}\} \rightarrow \{\text{Dno}\}$ 
 $\{\text{Pno}\} \rightarrow \{\text{Pname}\}$ 
 $\{\text{Pno}\} \rightarrow \{\text{Plocation}\}$ 


<u>Emp_ssn</u>	Esal	Ephone	Dno
----------------	------	--------	-----

<u>Pno</u>	Pname	Plocation
------------	-------	-----------

# Normalisatie door decompositie

## Algoritme *Afhankelijkhedenbewarende Decompositie*

### Afhankelijkheden F

$\{M,S\} \rightarrow \{P,C\}$      $\{S\} \rightarrow \{Y\}$      $\{M\} \rightarrow \{N\}$      $\{N,Y\} \rightarrow \{P\}$

### Minimale overdekking E

$\{M,S\} \rightarrow \{C\}$      $\{S\} \rightarrow \{Y\}$      $\{M\} \rightarrow \{N\}$      $\{N,Y\} \rightarrow \{P\}$

<u>Model#</u>	<u>Serial#</u>	Color
<u>Serial#</u>	Year	
<u>Model#</u>	Name	
<u>Name</u>	<u>Year</u>	Price

# Normalisatie door decompositie

We gaan uit van een decompositie die informatie moet bewaren, die verliesloos is.

=> Oorspronkelijke relatie moet reconstrueerbaar zijn.

=> Herinner: EMP\_LOCS en EMP\_PROJ1

\* Oorspronkelijke relatie niet reconstrueerbaar.

\* Dus, geen behoud van informatie!

- D.w.z. Ook geen onechte tupels introduceren!

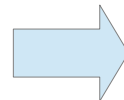
# Normalisatie door decompositie

Vermijden v. (opgedeelde) relaties waarbij een equi-join op gerelateerde attributen onechte/valse tupels oplevert. Vb.:

EMP\_PROJ

Ssn	Pnumber	Hours	Ename	Pname	Plocation
-----	---------	-------	-------	-------	-----------

Ssn	Pnumber	Hours	Ename	Pname	Plocation
123456789	1	32.5	Smith, John B.	ProductX	Bellaire
123456789	2	7.5	Smith, John B.	ProductY	Sugarland
666884444	3	40.0	Narayan, Ramesh K.	ProductZ	Houston
453453453	1	20.0	English, Joyce A.	ProductX	Bellaire
453453453	2	20.0	English, Joyce A.	ProductY	Sugarland
333445555	2	10.0	Wong, Franklin T.	ProductY	Sugarland
333445555	3	10.0	Wong, Franklin T.	ProductZ	Houston
333445555	10	10.0	Wong, Franklin T.	Computerization	Stafford
333445555	20	10.0	Wong, Franklin T.	Reorganization	Houston
999887777	30	30.0	Zelaya, Alicia J.	Newbenefits	Stafford
999887777	10	10.0	Zelaya, Alicia J.	Computerization	Stafford
987987987	10	35.0	Jabbar, Ahmad V.	Computerization	Stafford
987987987	30	5.0	Jabbar, Ahmad V.	Newbenefits	Stafford
987654321	30	20.0	Wallace, Jennifer S.	Newbenefits	Stafford
987654321	20	15.0	Wallace, Jennifer S.	Reorganization	Houston
888665555	20	Null	Borg, James E.	Reorganization	Houston



EMP\_LOCS

Ename	Plocation
-------	-----------

EMP\_LOCS

Ename	Plocation
Smith, John B.	Bellaire
Smith, John B.	Sugarland
Narayan, Ramesh K.	Houston
English, Joyce A.	Bellaire
English, Joyce A.	Sugarland
Wong, Franklin T.	Sugarland
Wong, Franklin T.	Houston
Wong, Franklin T.	Stafford
Zelaya, Alicia J.	Stafford
Jabbar, Ahmad V.	Stafford
Wallace, Jennifer S.	Stafford
Wallace, Jennifer S.	Houston
Borg, James E.	Houston

EMP\_PROJ1

Ssn	Pnumber	Hours	Pname	Plocation
-----	---------	-------	-------	-----------

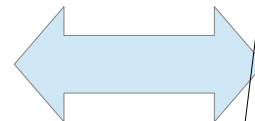
EMP\_PROJ1

Ssn	Pnumber	Hours	Pname	Plocation
123456789	1	32.5	ProductX	Bellaire
123456789	2	7.5	ProductY	Sugarland
666884444	3	40.0	ProductZ	Houston
453453453	1	20.0	ProductX	Bellaire
453453453	2	20.0	ProductY	Sugarland
333445555	2	10.0	ProductY	Sugarland
333445555	3	10.0	ProductZ	Houston
333445555	10	10.0	Computerization	Stafford
333445555	20	10.0	Reorganization	Houston
999887777	30	30.0	Newbenefits	Stafford
999887777	10	10.0	Computerization	Stafford
987987987	10	35.0	Computerization	Stafford
987987987	30	5.0	Newbenefits	Stafford
987654321	30	20.0	Newbenefits	Stafford
987654321	20	15.0	Reorganization	Houston
888665555	20	NULL	Reorganization	Houston

# Normalisatie door decompositie

Vermijden v. (opgedeelde) relaties waarbij een equi-join op gerelateerde attributen onechte/valse tupels oplevert. Vb.:

Ssn	Pnumber	Hours	Ename	Pname	Plocation
123456789	1	32.5	Smith, John B.	ProductX	Bellaire
123456789	2	7.5	Smith, John B.	ProductY	Sugarland
666884444	3	40.0	Narayan, Ramesh K.	ProductZ	Houston
453453453	1	20.0	English, Joyce A.	ProductX	Bellaire
453453453	2	20.0	English, Joyce A.	ProductY	Sugarland
333445555	2	10.0	Wong, Franklin T.	ProductY	Sugarland
333445555	3	10.0	Wong, Franklin T.	ProductZ	Houston
333445555	10	10.0	Wong, Franklin T.	Computerization	Stafford
333445555	20	10.0	Wong, Franklin T.	Reorganization	Houston
999887777	30	30.0	Zelaya, Alicia J.	Newbenefits	Stafford
999887777	10	10.0	Zelaya, Alicia J.	Computerization	Stafford
987987987	10	35.0	Jabbar, Ahmad V.	Computerization	Stafford
987987987	30	5.0	Jabbar, Ahmad V.	Newbenefits	Stafford
987654321	30	20.0	Wallace, Jennifer S.	Newbenefits	Stafford
987654321	20	15.0	Wallace, Jennifer S.	Reorganization	Houston
888665555	20	Null	Borg, James E.	Reorganization	Houston



Ssn	Pnumber	Hours	Pname	Plocation	Ename
123456789	1	32.5	ProductX	Bellaire	Smith, John B.
123456789	1	32.5	ProductX	Bellaire	English, Joyce A.
123456789	2	7.5	ProductY	Sugarland	Smith, John B.
123456789	2	7.5	ProductY	Sugarland	English, Joyce A.
123456789	2	7.5	ProductY	Sugarland	Wong, Franklin T.
666884444	3	40.0	ProductZ	Houston	Narayan, Ramesh K.
666884444	3	40.0	ProductZ	Houston	Wong, Franklin T.
453453453	1	20.0	ProductX	Bellaire	Smith, John B.
453453453	1	20.0	ProductX	Bellaire	English, Joyce A.
453453453	2	20.0	ProductY	Sugarland	Smith, John B.
453453453	2	20.0	ProductY	Sugarland	English, Joyce A.
453453453	2	20.0	ProductY	Sugarland	Wong, Franklin T.
333445555	2	10.0	ProductY	Sugarland	Smith, John B.
333445555	2	10.0	ProductY	Sugarland	English, Joyce A.
333445555	2	10.0	ProductY	Sugarland	Wong, Franklin T.
333445555	3	10.0	ProductZ	Houston	Narayan, Ramesh K.
333445555	3	10.0	ProductZ	Houston	Wong, Franklin T.
333445555	10	10.0	Computerization	Stafford	Wong, Franklin T.
333445555	20	10.0	Reorganization	Houston	Narayan, Ramesh K.
333445555	20	10.0	Reorganization	Houston	Wong, Franklin T.

onecht

# Normalisatie door decompositie

We gaan uit van een decompositie die informatie moet bewaren => die verliesloos is.

Een decompositie  $\delta(S_R) = \{S_{R_1}, S_{R_2}, \dots, S_{R_k}\}$  van  $R$  is verliesloos a.s.a. voor elke instantie  $r$  van  $R$  die aan  $F_R$  voldoet, geldt:  $\pi_{UR_1}(r) * \pi_{UR_2}(r) * \dots * \pi_{UR_k}(r) = r$

=> Natural join van relaties in decompositie levert geen onechte tupels op.

=> Dit kan getest worden met Chase-algoritme.

# Normalisatie door decompositie

## Chase-algoritme

- \* Op basis van Matrix:
  - Kolom per attribuut in R.
  - Rij per relatie  $R_i$  in de decompositie.
- \* Voor elke rij  $i$  vul  $a_j$  in op positie  $A_j$  als in  $S_{R_i}$ , anders  $b_{ij}$ .
- \* Redenering: Als  $X \rightarrow Y$  en de waarden  $X$  gelijk in 2 rijen dan ook de  $Y$
- \* Doorgaan tot onveranderd of tot rij met enkel a's.

Rij met enkel a's  $\Leftrightarrow$  verliesloze decompositie.



# Normalisatie door decompositie

## Chase-algoritme

Test of decompositie  $D = \{R_1, R_2, \dots, R_n\}$  van een relatie  $R$  met afhankelijkheden  $F$  verliesloos is:

### 1. Initialisatie:

Maak matrix  $S$  met 1 rij  $i$  voor elke relatie  $R_i$  in de decompositie  $D$ , en 1 kolom  $j$  voor elk attribuut  $A_j$  in  $R$ .

**voor elke** rij  $i$ :

**voor elke** kolom  $j$ :

**als**  $A_j$  als attribuut voorkomt in  $R_i$  **dan**  $S_{ij} := a_j$  **anders**  $S_{ij} := b_{ij}$

# Normalisatie door decompositie

## Chase-algoritme

Test of decompositie  $D = \{R_1, R_2, \dots, R_n\}$  van een relatie  $R$  met afhankelijkheden  $F$  verliesloos is:

### 2. Procedure

**herhaal tot**  $S$  niet meer verandert:

**voor elke** functionele afhankelijkheid  $X \rightarrow Y$  in  $F$ :

**voor elke** verzameling rijen  $V$  in  $S$  waarvoor geldt dat de symbolen in de kolommen overeenkomend met  $X$  dezelfde zijn:

maak de symbolen in de kolommen van  $Y$  ook gelijk, als volgt:

**als** één van de rijen in  $V$  een  $a_j$  bevat **dan voor elke**  $k$  in  $V$ :  $S_{kj} := a_j$

**anders** kies een  $i \in V$  **voor elke**  $k$  in  $V$ :  $S_{kj} := b_{ij}$

{  $D$  is verliesloos a.s.a. er is een rij enkel met  $a$ 's }

# Normalisatie door decompositie

## Chase-algoritme (Voorbeeld)

$R = \{Ssn, Ename, Pnumber, Pname, Plocation, Hours\}$

$R_1 = EMP\_LOCS = \{Ename, Plocation\}$

$R_2 = EMP\_PROJ1 = \{Ssn, Pnumber, Hours, Pname, Plocation\}$

$D = \{R_1, R_2\}$

$F = \{Ssn \twoheadrightarrow Ename; Pnumber \twoheadrightarrow \{Pname, Plocation\}; \{Ssn, Pnumber\} \twoheadrightarrow Hours\}$

	Ssn	Ename	Pnumber	Pname	Plocation	Hours
$R_1$	$b_{11}$	$a_2$	$b_{13}$	$b_{14}$	$a_5$	$b_{16}$
$R_2$	$a_1$	$b_{22}$	$a_3$	$a_4$	$a_5$	$a_6$

Ename	Plocation
$a_2$	$a_5$
$b_{22}$	$a_5$

Natural JOIN  
levert 4 tupels  
i.p.v. 2



Ssn	Pnumber	Pname	Plocation	Hours
$b_{11}$	$b_{13}$	$b_{14}$	$a_5$	$b_{16}$
$a_1$	$a_3$	$a_4$	$a_5$	$a_6$

Geen verliesloze decompositie!

# Normalisatie door decompositie

## Chase-algoritme (Voorbeeld)

$R = \{Ssn, Ename, Pnumber, Pname, Plocation, Hours\}$

$D = \{R_1, R_2, R_3\}$

$R_1 = EMP = \{Ssn, Ename\}$

$R_2 = PROJ = \{Pnumber, Pname, Plocation\}$

$R_3 = WORKS_ON = \{Ssn, Pnumber, Hours\}$

$F = \{Ssn \twoheadrightarrow Ename; Pnumber \twoheadrightarrow \{Pname, Plocation\}; \{Ssn, Pnumber\} \twoheadrightarrow Hours\}$

	Ssn	Ename	Pnumber	Pname	Plocation	Hours
$R_1$	$a_1$	$a_2$	$b_{13}$	$b_{14}$	$b_{15}$	$b_{16}$
$R_2$	$b_{21}$	$b_{22}$	$a_3$	$a_4$	$a_5$	$b_{26}$
$R_3$	$a_1$	$b_{32}$	$a_3$	$b_{34}$	$b_{35}$	$a_6$



	Ssn	Ename	Pnumber	Pname	Plocation	Hours
$R_1$	$a_1$	$a_2$	$b_{13}$	$b_{14}$	$b_{15}$	$b_{16}$
$R_2$	$b_{21}$	$b_{22}$	$a_3$	$a_4$	$a_5$	$b_{26}$
$R_3$	$a_1$	<del><math>b_{32}</math></del> $a_2$	$a_3$	<del><math>b_{34}</math></del> $a_4$	<del><math>b_{35}</math></del> $a_5$	$a_6$

Wel verliesloze decompositie!

# Normalisatie door decompositie

Niet-additieve JOIN test voor binaire decompositie:

$\delta(S_R) = \{S_{R1}, S_{R2}\}$  is verliesloos t.o.v.  $F_R$  a.s.a.

$(U_{R1} \cap U_{R2}) \rightarrow (U_{R1} \setminus U_{R2}) \in F_R^+$  of

$(U_{R1} \cap U_{R2}) \rightarrow (U_{R2} \setminus U_{R1}) \in F_R^+$

- \* Eenvoudiger dan "Chase" algoritme.
- \* Enkel voor splitsing in 2 schema's.
- \* Kan recursief!

# Normalisatie door decompositie

Niet-additieve JOIN test voor bin. decomp. (Voorbeeld):

$R = \{Ssn, Ename, Pnumber, Pname, Plocation, Hours\}$

$D = \{R_1, R_2\}$

$R_1 = EMP\_LOCS = \{Ename, Plocation\}$

$R_2 = EMP\_PROJ1 = \{Ssn, Pnumber, Hours, Pname, Plocation\}$

$F = \{Ssn \twoheadrightarrow Ename; Pnumber \twoheadrightarrow \{Pname, Plocation\}; \{Ssn, Pnumber\} \twoheadrightarrow Hours\}$

$$\begin{aligned} & \{Ename\} \\ & \boxed{(U_{R_1} \cap U_{R_2})} \rightarrow (U_{R_1} \setminus U_{R_2}) \in F_R^+ \text{ of} \\ & \boxed{(U_{R_1} \cap U_{R_2})} \rightarrow (U_{R_2} \setminus U_{R_1}) \in F_R^+ \\ & \{Ssn, Pnumber, Hours, Pname\} \end{aligned}$$

$\{Plocation\}_{FR}^+ = \{Plocation\} \Rightarrow$  Niet verliesloos

# Normalisatie door decompositie

Niet-additieve JOIN test voor bin. decomp. (Voorbeeld):

$R = \{Ssn, Ename, Pnumber, Pname, Plocation, Hours\}$

$D = \{R_1, R_2, R_3\}$

$R_1 = EMP = \{Ssn, Ename\}$

$R_2 = PROJ = \{Pnumber, Pname, Plocation\}$

$R_3 = WORKS_ON = \{Ssn, Pnumber, Hours\}$

$F = \{Ssn \twoheadrightarrow Ename; Pnumber \twoheadrightarrow \{Pname, Plocation\}; \{Ssn, Pnumber\} \twoheadrightarrow Hours\}$

Stap 1:  $U_{R_1} = \{Ssn, Ename\}$  en  $U_{R_2} = \{Ssn, Pnumber, Pname, Plocation, Hours\}$

$(U_{R_1} \cap U_{R_2}) \rightarrow (U_{R_1} \setminus U_{R_2}) \in F_R^+$  of

$(U_{R_1} \cap U_{R_2}) \rightarrow (U_{R_2} \setminus U_{R_1}) \in F_R^+$

$\{Ssn\}_{FR}^+ = \{Ssn, Ename\} \Rightarrow$  Wel verliesloos

# Normalisatie door decompositie

Niet-additieve JOIN test voor bin. decomp. (Voorbeeld):

$$R = \{\text{Ssn, Ename, Pnumber, Pname, Plocation, Hours}\}$$

$$D = \{R_1, R_2, R_3\}$$

$$R_1 = \text{EMP} = \{\text{Ssn, Ename}\}$$

$$R_2 = \text{PROJ} = \{\text{Pnumber, Pname, Plocation}\}$$

$$R_3 = \text{WORKS\_ON} = \{\text{Ssn, Pnumber, Hours}\}$$

$$F = \{\text{Ssn} \twoheadrightarrow \text{Ename}; \text{Pnumber} \twoheadrightarrow \{\text{Pname, Plocation}\}; \{\text{Ssn, Pnumber}\} \twoheadrightarrow \text{Hours}\}$$

Stap 2:  $U_{R_1} = \{\text{Pnumber, Pname, Plocation}\}$  en  $U_{R_2} = \{\text{Ssn, Pnumber, Hours}\}$

$$(U_{R_1} \cap U_{R_2}) \rightarrow (U_{R_1} \setminus U_{R_2}) \in F_R^+ \text{ of}$$

$$(U_{R_1} \cap U_{R_2}) \rightarrow (U_{R_2} \setminus U_{R_1}) \in F_R^+$$



# Normalisatie door decompositie

We hebben nu volgende methodes:

- \* *Afhankelijkhedenbewarend naar 3NF, niet verliesloos.*
- \* *Chase-algoritme voor verificatie verliesloos*
- \* *Verificatie verliesloos via binaire decompositie*

Alternatieven?

# Normalisatie door decompositie

Ideale decompositie?  
*Informatiebewarend.*

1. Alle relaties in BCNF
2. Informatiebewarend (Verliesloos)
3. ~~Afhankelijkhedenbewarend~~

# Normalisatie door decompositie

Verliesloze decompositie van R naar BCNF normaalvorm:

$$\delta := \{S_R\}$$

**Zolang** er een relatieschema in  $\delta$  niet in BCNF is:

1. Zoek een FD  $X \rightarrow Y$  in  $S_{R_i}$  die BCNF-niet-toegelaten is  
 $\Rightarrow X \rightarrow Y$  in  $F^+$ ,  $X \rightarrow Y$  niet-triviaal,  $X$  geen supersleutel
2. Vervang  $S_{R_i}$  in  $\delta$  door twee schema's  $S_{R_i} \setminus Y$  en  $X \cup Y$ :  
$$\delta := \delta \setminus \{S_{R_i}\} \cup \{S_{R_i} \setminus Y, X \cup Y\}$$

$\Rightarrow$  FDs kunnen verloren gaan!

# Normalisatie door decompositie

## Verliesloze decompositie naar BCNF (Voorbeeld):

- \* Gegevensbank met info over huizen
- \* Universele relatie:  $R(O,S,H,P,W,B,D)$ 
  - O: Identificatienummer huis
  - S: Straat
  - H: Huisnummer
  - P: Postcode
  - W: Wijk
  - B: Bouwbedrijf
  - D: Directeur bouwbedrijf

# Normalisatie door decompositie

## Verliesloze decompositie naar BCNF (Voorbeeld):

- \* Gegevensbank met info over huizen
- \* Universele relatie:  $R(O,S,H,P,W,B,D)$
- \* Functionele afhankelijkheden:  
 $F_R = \{ O \rightarrow BSH, B \rightarrow D, D \rightarrow B, SH \rightarrow OP, P \rightarrow S, S \rightarrow W \}$ 
  - Identificatienummer bepaalt bouwbedrijf, straat en huisnummer
  - Bouwbedrijf en directeur bepalen elkaar
  - Straat en huisnummer bepalen identificatienummer en postcode
  - Postcode bepaalt straat en straat bepaalt wijk
- \* Kandidaatsleutels: O, SH, PH

# Normalisatie door decompositie

## Verliesloze decompositie naar BCNF (Voorbeeld):

\* Universele relatie:  $R(O, S, H, P, W, B, D)$

\* FDs:  $F_R = \{ O \rightarrow BSH, B \rightarrow D, D \rightarrow B, SH \rightarrow OP, P \rightarrow S, S \rightarrow W \}$

\* Kandidaatsleutels: O, SH, PH

\* Voorbeeldinstantie:

O	S	H	P	W	B	D
15351	Perenstraat	<b>23</b>	2745 DD	Fruitwijk	Bouwlust	P. Akker
15352	Perenstraat	<b>25</b>	2745 DD	Fruitwijk	Bouwlust	P. Akker
15788	Appelstraat	<b>8</b>	2745 EG	Fruitwijk	Solide	Z. Glas
20371	Beukenstraat	<b>94</b>	2751 BE	Bomenwijk	Solide	Z. Glas
20372	Beukenstraat	<b>96</b>	2751 BE	Bomenwijk	Solide	Z. Glas
20167	Beukenstraat	<b>1</b>	2751 BB	Bomenwijk	Bouwlust	P. Akker

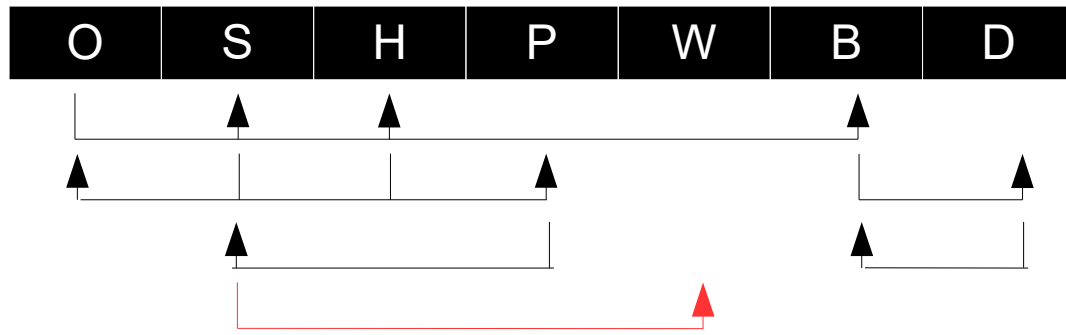
\* Redundantie in Straat, Postcode, Wijk, Bouwbedrijf, Directie

# Normalisatie door decompositie

Verliesloze decompositie naar BCNF (Voorbeeld):

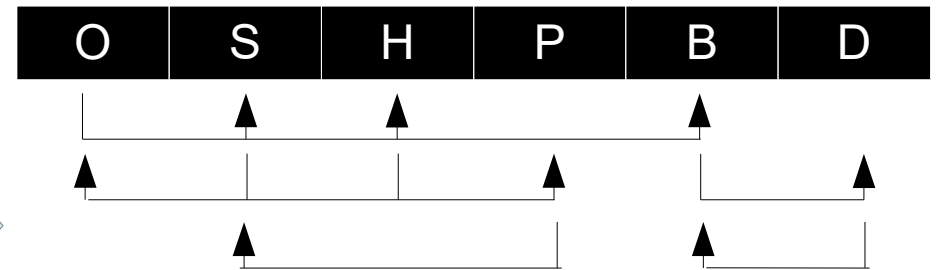
\*  $F_R = \{ O \rightarrow BSH, B \rightarrow D, D \rightarrow B, SH \rightarrow OP, P \rightarrow S, S \rightarrow W \}$

$K_R = \{O, SH, PH\}$



=> S is geen supersleutel!

$K_{R_1} = \{O, SH, PH\}$



$K_{R_2} = \{S\}$



# Normalisatie door decompositie

## Verliesloze decompositie naar BCNF (Voorbeeld):

\* Voorbeeldinstantie wordt nu:

<b>O</b>	<b>S</b>	<b>H</b>	<b>P</b>	<b>B</b>	<b>D</b>
15351	Perenstraat	<b>23</b>	2745 DD	Bouwlust	P. Akker
15352	Perenstraat	<b>25</b>	2745 DD	Bouwlust	P. Akker
15788	Appelstraat	<b>8</b>	2745 EG	Solide	Z. Glas
20371	Beukenstraat	<b>94</b>	2751 BE	Solide	Z. Glas
20372	Beukenstraat	<b>96</b>	2751 BE	Solide	Z. Glas
20167	Beukenstraat	<b>1</b>	2751 BB	Bouwlust	P. Akker

<b>S</b>	<b>W</b>
Perenstraat	Fruitwijk
Appelstraat	Fruitwijk
Beukenstraat	Bomenwijk

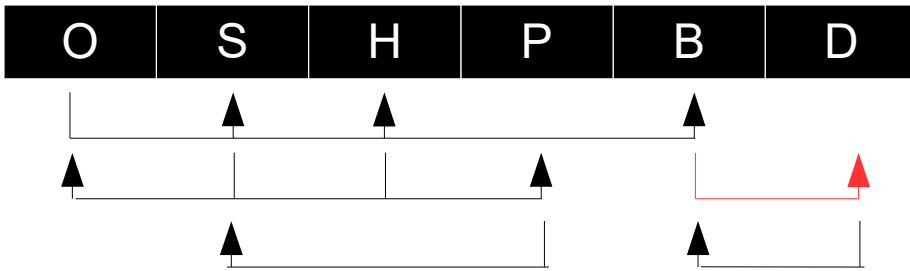


# Normalisatie door decompositie

Verliesloze decompositie naar BCNF (Voorbeeld):

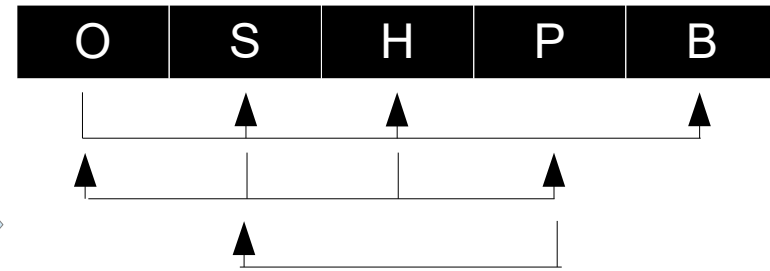
\*  $F_R = \{ O \rightarrow BSH, B \rightarrow D, D \rightarrow B, SH \rightarrow OP, P \rightarrow S, S \rightarrow W \}$

$K_{R1} = \{O, SH, PH\}$

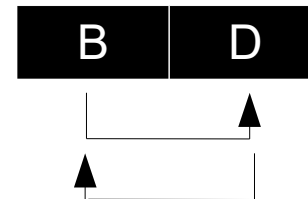


=> B is geen supersleutel!

$K_{R3} = \{O, SH, PH\}$



$K_{R4} = \{B, D\}$



# Normalisatie door decompositie

## Verliesloze decompositie naar BCNF (Voorbeeld):

\* Voorbeeldinstantie wordt nu:

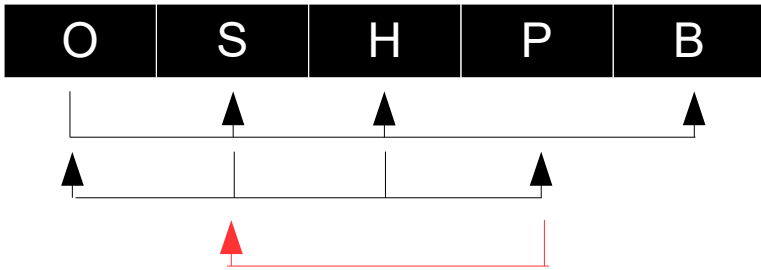
<b>O</b>	<b>S</b>	<b>H</b>	<b>P</b>	<b>B</b>	
15351	Perenstraat	<b>23</b>	2745 DD	Bouwlust	
15352	Perenstraat	<b>25</b>	2745 DD	Bouwlust	
15788	Appelstraat	<b>8</b>	2745 EG	Solide	
20371	Beukenstraat	<b>94</b>	2751 BE	Solide	
20372	Beukenstraat	<b>96</b>	2751 BE	Solide	
20167	Beukenstraat	<b>1</b>	2751 BB	Bouwlust	
<b>S</b>	<b>W</b>	<b>B</b>	<b>D</b>		
Perenstraat	Fruitwijk	Bouwlust	P. Akker		
Appelstraat	Fruitwijk	Solide	Z. Glas		
Beukenstraat	Bomenwijk				

# Normalisatie door decompositie

Verliesloze decompositie naar BCNF (Voorbeeld):

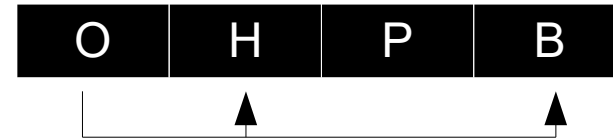
\*  $F_R = \{ O \rightarrow BSH, B \rightarrow D, D \rightarrow B, SH \rightarrow OP, P \rightarrow S, S \rightarrow W \}$

$K_{R3} = \{O, SH, PH\}$



=> P is geen supersleutel!

$K_{R5} = \{O, PH\}$



$K_{R6} = \{P\}$



# Normalisatie door decompositie

Verliesloze decompositie naar BCNF (Voorbeeld):

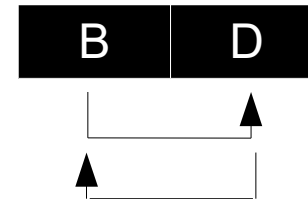
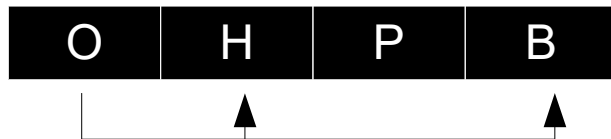
\*  $F_R = \{ O \rightarrow BSH, B \rightarrow D, D \rightarrow B, SH \rightarrow OP, P \rightarrow S, S \rightarrow W \}$

$K_{R5} = \{O, PH\}$

$K_{R2} = \{S\}$

$K_{R4} = \{B, D\}$

$K_{R6} = \{P\}$



\* In BCNF!

\* Maar er zijn FDs "verloren" gegaan (d.w.z. Moeilijker te controleren)  
=> Namelijk:  $O \rightarrow BSH, SH \rightarrow OP$

# Normalisatie door decompositie

## Verliesloze decompositie naar BCNF (Voorbeeld):

\* Uiteindelijke voorbeeldinstantie:

<b>O</b>	<b>H</b>	<b>P</b>	<b>B</b>	<b>P</b>	<b>S</b>
15351	<b>23</b>	2745 DD	Bouwlust	2745 DD	Perenstraat
15352	<b>25</b>	2745 DD	Bouwlust	2745 EG	Appelstraat
15788	<b>8</b>	2745 EG	Solide	2751 BE	Beukenstraat
20371	<b>94</b>	2751 BE	Solide	2751 BB	Beukenstraat
20372	<b>96</b>	2751 BE	Solide		
20167	<b>1</b>	2751 BB	Bouwlust		

<b>S</b>	<b>W</b>	<b>B</b>	<b>D</b>
Perenstraat	Fruitwijk	Bouwlust	P. Akker
Appelstraat	Fruitwijk	Solide	Z. Glas
Beukenstraat	Bomenwijk		

# Normalisatie door decompositie

## Ideale decompositie?

We hebben nu:

- \* Afhankelijkhedenbewarend naar 3NF.
- \* Informatiebewarend naar BCNF.

## Combinatie mogelijk?

- \* Ja, tot 3NF - BCNF niet altijd bereikbaar.

# Normalisatie door decompositie

Ideale decompositie?

*Afhankelijkhedenbewarend & Informatiebewarend.*

1. Alle relaties in ~~BCNF~~ 3NF
2. Informatiebewarend (Verliesloos)
3. Afhankelijkhedenbewarend

# Normalisatie door decompositie

## Algoritme Afh. -& infobewarende Decompositie

*Zoals afhankelijkhedenbewarend met verschil van derde stap!*

1. Zoek minimale overdekking  $E$  van  $F$
2. Voor elke  $X$  in  $E$  aan de linkerzijde van een FD:  
Maak schema  $\{X \cup A_1 \cup A_2 \cup \dots \cup A_m\}$  (prim. sleutel  $X$ )  
waarbij  $X \rightarrow A_1, X \rightarrow A_2, \dots, X \rightarrow A_m$  alle afhankelijkheden in  $E$  met  $X$  als linkerlid
- ~~3. Plaats overblijvende attributen in één enkel relatieschema~~
3. Als geen van de relatieschema's een sleutel van  $R$  bevat dan maak een relatieschema dat attributen bevat die een sleutel voor  $R$  vormen



# Normalisatie door decompositie

## Algoritme *Afhankelijkhedenbewarende Decompositie*

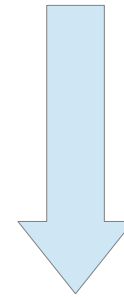
### Afhankelijkheden F

$\{Emp\_ssn\} \rightarrow \{Esal, Ephone, Dno\}$

$\{Pno\} \rightarrow \{Pname, Plocation\}$

$\{Emp\_ssn, Pno\} \rightarrow \{Esal, Ephone, Dno, Pname, Plocation\}$

<u>Emp_ssn</u>	<u>Pno</u>	Esal	Ephone	Dno	Pname	Plocation
----------------	------------	------	--------	-----	-------	-----------



### Minimale overdekking E

$\{Emp\_ssn\} \rightarrow \{Esal\}$

$\{Emp\_ssn\} \rightarrow \{Ephone\}$

$\{Emp\_ssn\} \rightarrow \{Dno\}$

$\{Pno\} \rightarrow \{Pname\}$

$\{Pno\} \rightarrow \{Plocation\}$



2

<u>Emp_ssn</u>	Esal	Ephone	Dno
----------------	------	--------	-----

<u>Pno</u>	Pname	Plocation
------------	-------	-----------

3

<u>Emp_ssn</u>	<u>Pno</u>
----------------	------------

# Normalisatie door decompositie

## Ideale decompositie?

We hebben nu:

- \* Afhankelijkhedenbewaarend naar 3NF.
- \* Informatiebewaarend naar BCNF.
- \* Afhankelijkhedenbewaarend & Informatiebewaarend naar 3NF.

Problemen? Cfr. Informele richtlijnen:

- Semantiek (=> Bewaard na decompositie, soms verdoken)
- Redundantie & anomalieën (=> Vermeden na decompositie)
- Vermijdt NULL Kan nog steeds problemen geven!
- Geen onechte tupels (=> Vermeden als informatiebewaarend)

# Normalisatie door decompositie

## Ideale decompositie?

Waarom geeft NULL problemen?

Omwille van:

1. Effect op aggregaatfuncties
2. Reconstructie originele relatie niet altijd mogelijk

NULL in verwijssleutel => Vb. Werknemer zonder departement

- EMPLOYEE \* DEPARTMENT zal werknemer niet olijsten
- Dit zijn "dangling tuples", niet beschouwd bij INNER JOIN
- Merk op: Een OUTER JOIN lost dit op

# Normalisatie door decompositie

Ideale decompositie – Waarom geeft NULL problemen?  
Dangling tuples!

EMPLOYEE

Ename	<u>Ssn</u>	Bdate	Address	Dnum
Smith, John B.	123456789	1965-01-09	731 Fondren, Houston, TX	5
Wong, Franklin T.	333445555	1955-12-08	638 Voss, Houston, TX	5
Zelaya, Alicia J.	999887777	1968-07-19	3321 Castle, Spring, TX	4
Wallace, Jennifer S.	987654321	1941-06-20	291 Berry, Bellaire, TX	4
Narayan, Ramesh K.	666884444	1962-09-15	975 Fire Oak, Humble, TX	5
English, Joyce A.	453453453	1972-07-31	5631 Rice, Houston, TX	5
Jabbar, Ahmad V.	987987987	1969-03-29	980 Dallas, Houston, TX	4
Borg, James E.	888665555	1937-11-10	450 Stone, Houston, TX	1
Berger, Anders C.	999775555	1965-04-26	6530 Braes, Bellaire, TX	NULL
Benitez, Carlos M.	888664444	1963-01-09	7654 Beech, Houston, TX	NULL

DEPARTMENT

Dname	<u>Dnum</u>	Dmgr_ssn
Research	5	333445555
Administration	4	987654321
Headquarters	1	888665555

# Normalisatie door decompositie

"Nadelen" decompositiemethode:

- \* Functionele afhankelijkheden dienen bekend te zijn
  - Moeilijk bij grote gegevensbanken met veel attributen
- \* De algoritmen zijn niet deterministisch
  - Minimale overdekking => meerdere oplossingen (Volgorde FDs)
- \* Na decompositie, JOINS nodig voor originele informatie
  - Computationeel duur => Kan performantie sterk beïnvloeden

# Normalisatie door decompositie

Tot slot decompositiemethode:

- \* Methode niet blindelings toepassen
- \* Combinatie ER-model en normalisatie is best:
  1. Begin met ER-model
  2. Beeldt af op relationeel model
  3. Zoek functionele afhankelijkheden
  4. Controleer of verdere decompositie mogelijk/wenselijk is