

Inleiding tot databanken

4. Normalisatie – Deel 1

Prof. dr. Paolo Pilozzi



Overzicht

4.1 Inleiding

4.2 Informele richtlijnen

4.3 Functionele Afhankelijkheden

4.4 Normaalvormen (Deel 1 & 2)

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
- * Normalvormen: NF1, NF2 en NF3

HC6 (Deel 2):

- * Herhaling HC5
- * Normalvormen: BCNF, 4NF en 5NF

Inleiding

Ontwerp v. databank gedefinieerd via

=> Relationeel databankschema (RDS)

=> Bestaande uit relatieschema's

=> Bestaande uit attributen

Tot nu toe gebeurde groepering van attributen op basis van:

1. Gezond verstand (bij opstellen RDS)

2. ER model (gemapt naar RDS)

=> Identificeren van (logische) entiteits -en relatietypes

Inleiding

Het is nuttig om via formele maatstaven na te gaan waarom één groepering van attributen beter is dan een ander!

- * **Immers**: We willen de kwaliteit/goedheid/geschiktheid meten!
- * Met als **eis**: informatiebewarend; met als **doel**:
 - redundantie minimaliseren, performantie maximaliseren, e.d.

=> Ontwikkelen van methodologie voor theoretisch databankontwerp
= **Normalisatie**

Inleiding

Twee niveaus om goedheid v. relatieschema's te bepalen:

1. Logisch/Conceptueel niveau
 - => Ter interpretatie van relatieschema's en hun attributen
 - => Maatstaf: duidelijkheid/betekenisvol
2. Implementatie/Opslag niveau
 - => Hoe tupels in relaties opgeslagen/aangepast worden

Normalisatie heeft (voornamelijk) betrekking op het implementatieniveau.

Inleiding

Databank ontwerp kan via twee methodes:

1. Bottom-up (synthese)

=> Vertrekt van individuele attributen en hun basis relaties

=> Toewerken naar vereisten (niet zo praktisch, wat omslachtig)

2. Top-down (analyse)

=> Vertrekt van groeperingen van attributen en natuurlijke relaties

=> Gegeven door toepassingsvereisten (aangeraden)

* Tip: Houdt bottom-up in gedachten.

Normalisatie is hier onafhankelijk van => in beide gevallen toepasbaar.

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

Informele Richtlijnen

Voornamelijk van toepassing op het conceptuele niveau met als richtlijnen:

1. Semantisch verantwoorde relaties en attributen (betekenisvol)
2. Reduceren van redundantie in tupels
3. Reduceren van mogelijke NULL waarden in tupels
4. Het niet toestaan van valse/onechte tupels

=> Informeel => subjectief (op basis van ervaring en gezond verstand)

=> Formeel => objectief ("valt niet over te discussieren")

Informeel - Semantiek

Duidelijke betekenis van relaties en hun attributen:

- * Betekenis makkelijk te verklaren
- * Relatieschema's zijn duidelijk/leesbaar/begrijpbaar

Daarom moet:

- * Een relatie feitelijk opgebouwd zijn
- * Het conceptueel ontwerp verzorgd en doordacht zijn
- * De omzetting naar schema's systematisch geschieden

Met als voornaamste richtlijn (Richtlijn 1):

- * Combineer geen attributen van entiteitstypes en relatietypes.
=> Vermijdt vermenging van aparte "real-world" entiteiten!

Informeel - Semantiek

Voorbeeld

EMP_DEPT

| | | | | | | |
|-------|------------|-------|---------|---------|-------|----------|
| Ename | <u>Ssn</u> | Bdata | Address | Dnumber | Dname | Dmgr_ssn |
|-------|------------|-------|---------|---------|-------|----------|

EMP_PROJ

| | | | | | |
|------------|----------------|-------|-------|-------|-----------|
| <u>Ssn</u> | <u>Pnumber</u> | Hours | Ename | Pname | Plocation |
|------------|----------------|-------|-------|-------|-----------|

VERMENGING



EMPLOYEE

| | | | | |
|-------|------------|-------|---------|---------|
| Ename | <u>Ssn</u> | Bdata | Address | Dnumber |
|-------|------------|-------|---------|---------|

PROJECT

| | | |
|----------------|-------|-----------|
| <u>Pnumber</u> | Pname | Plocation |
|----------------|-------|-----------|

DEPARTMENT

| | | |
|----------------|-------|----------|
| <u>Dnumber</u> | Dname | Dmgr_ssn |
|----------------|-------|----------|

WORKS_ON

| | | |
|------------|----------------|-------|
| <u>Ssn</u> | <u>Pnumber</u> | Hours |
|------------|----------------|-------|

Informeel - Redundantie

Vermijden van overbodigheden om zo

- * opslag te optimaliseren.
- * anomalieën te voorkomen bij aanpassingen.

Voorbeeld opslagoptimalisatie:

EMP_DEPT

| | | | | | | | |
|-------|------------|-------|---------|---------|-------|----------|------------------------|
| Ename | <u>Ssn</u> | Bdata | Address | Dnumber | Dname | Dmgr_ssn | } REDUNDANTE DEPT.DATA |
|-------|------------|-------|---------|---------|-------|----------|------------------------|



| | | | | | | | | |
|---|----------|------------|-------|---------|---------|----------------|-------|-----------|
| { | EMPLOYEE | | | | | PROJECT | | |
| | Ename | <u>Ssn</u> | Bdata | Address | Dnumber | <u>Pnumber</u> | Pname | Plocation |

Informeel - Redundantie

Vermijden van overbodigheden om zo

- * opslag te optimaliseren.
- * anomalieën te voorkomen bij aanpassingen

Voorbeeld toevoeg-anomalieën:

EMP_DEPT

| | | | | | | |
|-------|------------|-------|---------|---------|-------|----------|
| Ename | <u>Ssn</u> | Bdata | Address | Dnumber | Dname | Dmgr_ssn |
|-------|------------|-------|---------|---------|-------|----------|

- * Nieuwe werknemer toevoegen?
 - Departementsgegevens moeten consistent blijven
 - Meerdere NULL waarden als werknemer zonder departement
- * Departement zonder werknemers toevoegen?

Informeel - Redundantie

Vermijden van overbodigheden om zo

- * opslag te optimaliseren.
- * anomalieën te voorkomen bij aanpassingen

Voorbeeld verwijder-anomalieën:

EMP_DEPT

| | | | | | | |
|-------|------------|-------|---------|---------|-------|----------|
| Ename | <u>Ssn</u> | Bdata | Address | Dnumber | Dname | Dmgr_ssn |
|-------|------------|-------|---------|---------|-------|----------|

- * Laatste werknemer uit departement verwijderen?
 - Departementsgegevens worden ook verwijderd

Informeel - Redundantie

Vermijden van overbodigheden om zo

- * opslag te optimaliseren.
- * anomalieën te voorkomen bij aanpassingen

Voorbeeld wijzig-anomalieën:

EMP_DEPT

| | | | | | | |
|-------|------------|-------|---------|---------|-------|----------|
| Ename | <u>Ssn</u> | Bdata | Address | Dnumber | Dname | Dmgr_ssn |
|-------|------------|-------|---------|---------|-------|----------|

- * Departementsgegevens worden gewijzigd?
 - Bij alle werknemers van departement wijzigen (consistentie)
 - Vb. Manager of departementsnaam wijzigen

Informeel - Redundantie

Vermijden van overbodigheden om zo

- * opslag te optimaliseren.
- * anomalieën te voorkomen bij aanpassingen

Met als voornaamste richtlijn (Richtlijn 2):

- Optimaliseer het geheugengebruik.
- Voorkom aanpassingsanomalieën in relaties.
- Indien aanpassingsanomalieën niet vermeden kunnen worden:
 - * Geef dit aan, opdat toepassingen ermee rekening houden.
- Naar performantie toe kan tegen richtlijn 2.B ingegaan worden.
 - * Vb. EMPT_DEP kan nuttig zijn om JOINS te vermijden.
 - * Dikwijls is het gebruik van VIEWS een goede tussenoplossing.

Informeel – NULL

Vermijden van NULL waarden in tupels

* Waarden die ongekend, niet van toepassing, afwezig, e.d. zijn
=> Opslag-onefficiënt, onvoorspelbaar bij JOINS en aggregaten
(cfr. COUNT), meerdere betekenissen (cfr. R1 – Semantiek)

Vb. Een relatie **Dieren** met attributen enkel van toepassing op vissen en vogels, zoals `zee_brak_zoetwater` en `kan_vliegen`, respectievelijk.

=> Steeds een aantal attributen NULL

=> Beter aparte tabel voor vissen, vogels, enz.

Met als voornaamste richtlijn (Richtlijn 3):

* Als NULL onvermijdbaar is, eis dan dat het een speciaal geval is.

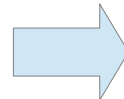
Informeel – Onechte tupels

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 |

Informeel – Onechte tupels

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

Informeel – Onechte tupels

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

Met als voornaamste richtlijn (Richtlijn 4):

- * Ontwerp relatieschema's zodat
 - na gewenste JOINS onechte tupels vermeden worden.
 - gelijknamige attributen (cfr. Natuurlijke JOIN) over relatieschema's heen na een JOIN geen onechte tupels genereren.
- * Splits relatieschema's op via primaire/kandidaat-verwijssleutel paren.

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

Functionele Afhankelijkheden

Definitie functionele afhankelijkheid:

Informeel is een **functionele afhankelijkheid** een beperking tussen twee verzamelingen attributen van een databank, zodat de waarden in tupels overeenkomstig met de afhankelijke attributen uniek bepaald worden door de waarden van de afgehangen attributen.

Beschouw daartoe een relationeel databankschema met n attributen A_1, A_2, \dots, A_n , waarbij we (formeel en dus niet in praktijk) de gehele databank beschrijven als een universeel relatieschema $R = \{A_1, A_2, \dots, A_n\}$.

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.

Functionele Afhankelijkheden

Doel: Expliciet beschrijven van uniciteitsrestricties waaraan attributen in relaties moeten voldoen.

Voorbeeld EMP_PROJ:

- * Het Ssn nummer bepaalt de naam op unieke wijze:

$$\text{Ssn} \rightarrow \text{Ename} ()$$

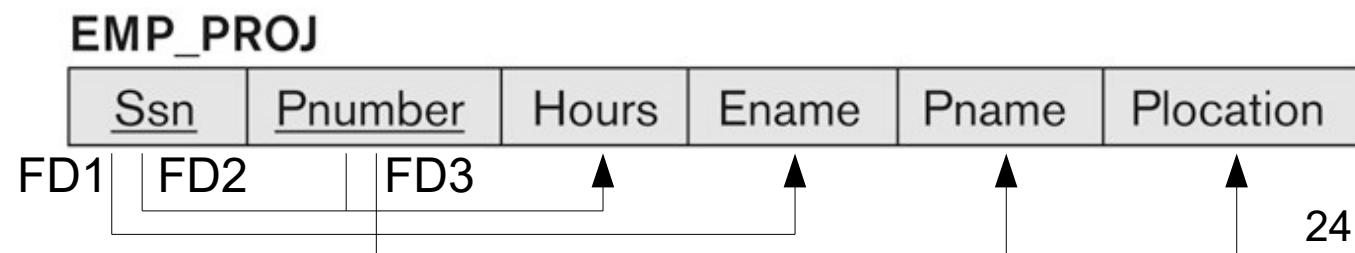
- * Een projectnummer bepaalt de projectnaam -en locatie op unieke wijze:

$$\text{Pnumber} \rightarrow \{\text{Pname}, \text{Plocation}\}$$

- * De combinatie ssn en projectnummer bepaalt gepresteerde tijd op unieke wijze:

$$\{\text{Ssn}, \text{Pnumber}\} \rightarrow \text{Hours}$$

=> FD pijlen in relatieschema:



Functionele Afhankelijkheden

Tupels in een instantie van een relatieschema kunnen enkel tegenvoorbeelden geven, maar bewijzen verder niets:

- * TEXT \rightarrow COURSE is plausibel. (*Maar ook voor alle legale toestanden?*)
- * COURSE \rightarrow TEXT onmogelijk want tegenvoorbeeld.
- * TEACHER \rightarrow COURSE onmogelijk want tegenvoorbeeld.
- * COURSE \rightarrow TEACHER is plausibel.

=> FDs zijn eigenschappen van een relatieschema R en niet van een legale toestand r van R!

=> Wordt bepaald bij ontwerp!

TEACH

| Teacher | Course | Text |
|---------|-----------------|----------|
| Smith | Data Structures | Bartram |
| Smith | Data Management | Martin |
| Hall | Compilers | Hoffman |
| Brown | Data Structures | Horowitz |

FDs – Afleidingsregels

F noteert een verzameling FDs v/e relatieschema R.

Een ontwerper van schema's zal de FDs in F oplist en die duidelijk/direct volgen uit de semantiek van schema's.

Er zullen andere FDs bestaan die niet opgelijst worden omdat ze niet vanzelfsprekend zijn ("vergezocht").

* Deze kunnen in vele gevallen afgeleid/geïnterfereerd worden uit F.

Vb. $Dno \rightarrow Mgr_ssn$ en $Mgr_ssn \rightarrow Mgr_phone \Rightarrow Dno \rightarrow Mgr_phone$

* Het is vrijwel onmogelijk in de praktijk alle FDs op te lijsten.

=> Formeel is het nuttig over sluiting te praten!

FDs – Afleidingsregels

Een verzameling, F , van FDs van R kan nieuwe FDs impliceren: $A \rightarrow B$ en $B \rightarrow C$ impliceert $A \rightarrow C$; wat genoteerd wordt als: $\{A \rightarrow B, B \rightarrow C\} \models A \rightarrow C$

Afleidingsregels laten toe om FDs van R te vinden die geïmpliceerd worden door F van R .

De sluiting van F , genoteerd als F^+ , is de verzameling van alle FDs die door F geïmpliceerd worden (door exhaustief afleidingsregels toe te passen).

FDs – Afleidingsregels

Voorbeeld:

Gegeven dat $F = \{$
 $Ssn \rightarrow \{Ename, Bdate, Address, Dnumber\},$
 $Dnumber \rightarrow \{Dname, Mgr_ssn\}$
}

Volgt er dat:

- * $Ssn \rightarrow \{Dname, Mgr_ssn\}$
- * $Ssn \rightarrow Ssn$
- * $Dnumber \rightarrow Dname$
- * ...

FDs – Afleidingsregels

Het symbool voor interferentie is ' \models ' zodat $F \models X \rightarrow Y$ noteert dat de FD, $X \rightarrow Y$, kan afgeleid worden uit de verzameling FDs, 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> | $\{X \rightarrow Y, Y \rightarrow Z\} \models X \rightarrow Z$ | |

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

* Sound: Alles dat afgeleid kan worden is geïmpliceerd.

* Complete: Alles dat geïmpliceerd is kan afgeleid worden.

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

FDs – Afleidingsregels

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

AR1: *Reflexiviteit* $Y \subseteq X \Rightarrow X \rightarrow Y$

Gegeven $Y \subseteq X$, en tupels $t_1, t_2 \in r$ van R zodat $t_1[X] = t_2[X]$ dan volgt dat $t_1[Y] = t_2[Y]$ omdat $Y \subseteq X$ en dus bij definitie dat $X \rightarrow Y$ in alle r van R .

FDs – Afleidingsregels

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

AR2: *Augmentatie* $\{X \rightarrow Y\} \models XZ \rightarrow YZ$

Stel dat er een r van R bestaat waarin $X \rightarrow Y$ geldt, maar niet $XZ \rightarrow YZ$.
Dan moeten er twee tupels t_1 en t_2 in r bestaan zodat:

* $t_1[X] = t_2[X]$ (1) en $t_1[Y] = t_2[Y]$ (2)

* $t_1[XZ] = t_2[XZ]$ (3) en $t_1[YZ] \neq t_2[YZ]$ (4)

Dit kan niet, want uit (1 en) 3 volgt $t_1[Z] = t_2[Z]$ (5); en uit 2 en 5 volgt dat $t_1[YZ] = t_2[YZ]$ (6), wat 4 tegenspreekt.

FDs – Afleidingsregels

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

AR3: *Transitiviteit* $\{X \rightarrow Y, Y \rightarrow Z\} \models X \rightarrow Z$

Veronderstel dat aan beide $X \rightarrow Y$ (1) en $Y \rightarrow Z$ (2) in alle r van R voldaan is. Dan voor alle tupels t_1 en t_2 in r zodat $t_1[X] = t_2[X]$, volgt er bij definitie dat $t_1[Y] = t_2[Y]$ (3) gegeven 1, waaruit volgt dat $t_1[Z] = t_2[Z]$ gegeven 2 en 3, en zodoende dat $X \rightarrow Z$ geldt voor alle tupels in r .

FDs – Afleidingsregels

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> | $\{X \rightarrow Y, WY \rightarrow Z\} \models WX \rightarrow Z$ |

De bewijzen (kan ook rechtstreeks uit definitie FD):

AR4: Gegeven $X \rightarrow YZ$ (1) en $Y \subseteq YZ$ (2) dan volgt dat $YZ \rightarrow Y$ (3) op basis van AR1 en 3 en dus $X \rightarrow Y$ door AR3 toe te passen op 1 en 3.

AR5: Gegeven $X \rightarrow Y$ (1) en $X \rightarrow Z$ (2) dan met AR2 op 1 krijgen we $X \rightarrow XY$ (3) door te augmenteren met X en met AR2 op 2 krijgen we $XY \rightarrow YZ$ (4) door te augmenteren met Y , en zodoende $X \rightarrow YZ$ door AR3 toe te passen op 3 en 4.

AR6: Gegeven $X \rightarrow Y$ (1) en $WY \rightarrow Z$ (2) dan met AR2 op 1 krijgen we $WX \rightarrow WY$ (3) door te augmenteren met W en vervolgens $WX \rightarrow Z$ door AR3 toe te passen op 2 en 3.

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)

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

Voor $\text{Ssn} \rightarrow \text{Ename}$:

$$1: \{\text{Ssn}\}_F^+ = \{\text{Ssn}\}$$

$$2: \{\text{Ssn}\}_F^+ = \{\text{Ssn}, \text{Ename}\} \quad \text{met } \text{Ssn} \rightarrow \text{Ename}$$

Voor $\text{Pnumber} \rightarrow \{\text{Pname}, \text{Plocation}\}$:

$$1: \{\text{Pnumber}\}_F^+ = \{\text{Pnumber}\}$$

$$2: \{\text{Pnumber}\}_F^+ = \{\text{Pnumber}, \text{Pname}, \text{Plocation}\}$$

met $\text{Pnumber} \rightarrow \{\text{Pname}, \text{Plocation}\}$

FDs – Algoritmisch (Voorbeeld)

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

Voor $\{\text{Ssn}, \text{Pnumber}\} \rightarrow \text{Hours}$:

$$1: \{\text{Ssn}, \text{Pnumber}\}_F^+ = \{\text{Ssn}, \text{Pnumber}\}$$

$$2: \{\text{Ssn}, \text{Pnumber}\}_F^+ = \{\text{Ssn}, \text{Pnumber}, \text{Ename}\} \quad \text{met } \text{Ssn} \rightarrow \text{Ename}$$

$$3: \{\text{Ssn}, \text{Pnumber}\}_F^+ = \{\text{Ssn}, \text{Pnumber}, \text{Ename}, \text{Pname}, \text{Plocation}\}$$

met $\text{Pnumber} \rightarrow \{\text{Pname}, \text{Plocation}\}$

$$4: \{\text{Ssn}, \text{Pnumber}\}_F^+ = \{\text{Ssn}, \text{Pnumber}, \text{Ename}, \text{Pname}, \text{Plocation}, \text{Hours}\}$$

met $\{\text{Ssn}, \text{Pnumber}\} \rightarrow \text{Hours}$

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 – Equivalentie (Voorbeeld)

$$E = \{A \rightarrow BC, D \rightarrow AE\}$$

$$F = \{A \rightarrow B, AB \rightarrow C, D \rightarrow AC, D \rightarrow E\}$$

E overdekt F? Ga na dat $\forall X \rightarrow Y \in F: Y \subseteq X_E^+$

$$A \rightarrow B: \quad A_E^+ = ABC \text{ en } B \subseteq ABC \quad \Rightarrow \text{OK}$$

$$AB \rightarrow C: \quad AB_E^+ = ABC \text{ en } C \subseteq ABC \quad \Rightarrow \text{OK}$$

$$D \rightarrow AC: \quad D_E^+ = DAEBC \text{ en } AC \subseteq DAEBC \quad \Rightarrow \text{OK}$$

$$D \rightarrow E: \quad D_E^+ = DAEBC \text{ en } E \subseteq DAEBC \quad \Rightarrow \text{OK}$$

FDs – Equivalentie (Voorbeeld)

$$E = \{A \rightarrow BC, D \rightarrow AE\}$$

$$F = \{A \rightarrow B, AB \rightarrow C, D \rightarrow AC, D \rightarrow E\}$$

F overdekt E? Ga na dat $\forall X \rightarrow Y \in E: Y \subseteq X_F^+$

$$A \rightarrow BC: \quad A_F^+ = ABC \text{ en } BC \subseteq ABC \quad \Rightarrow \text{OK}$$

$$D \rightarrow AE: \quad D_F^+ = DACBE \text{ en } AE \subseteq DACBE \quad \Rightarrow \text{OK}$$

\Rightarrow E overdekt F en F overdekt E

\Rightarrow **E en F zijn equivalent.**

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 – Minimaal (Algoritme)

Vindt een minimale overdekking F van E :

1. Ken E aan F toe $\Rightarrow F := E$
2. Vervang elke $X \rightarrow \{A_1, A_2, \dots, A_n\}$ in F door $X \rightarrow A_1, X \rightarrow A_2, \dots, X \rightarrow A_n$.
 \Rightarrow Voldoe aan de eerste eigenschap.
3. Voor elke $X \rightarrow A$ in F en voor elk attribuut B in X als $(F \setminus \{X \rightarrow A\}) \cup ((X \setminus \{B\}) \rightarrow A)$ is equivalent aan F vervang $X \rightarrow A$ door $(X \setminus \{B\}) \rightarrow A$.
 \Rightarrow Voldoe aan de tweede eigenschap.
4. Voor elke $X \rightarrow A$ in F als $(F \setminus \{X \rightarrow A\})$ is equivalent aan F verwijder $X \rightarrow A$
 \Rightarrow Voldoe aan de derde eigenschap.

FDs – Minimaal (Algoritme)

Vindt een minimale overdekking F van E :

$F := E$

voor elke $X \rightarrow Y$ in F met $Y = \{A_1, A_2, \dots, A_n\}$ (en $n > 1$):

$F := (F \setminus \{X \rightarrow Y\}) \cup \{X \rightarrow \{A_1\}, \dots, X \rightarrow \{A_n\}\}$

voor elke $X \rightarrow A$ in F :

voor elke $B \in X$:

$X' := X \setminus \{B\}$

$F' := (F \setminus \{X \rightarrow A\}) \cup \{X' \rightarrow A\}$

als $B \in X_{F'}^+$ dan $F := F'$

voor elke $X \rightarrow A$ in F :

$F' := F \setminus \{X \rightarrow A\}$

als $A \in X_{F'}^+$ dan $F := F'$

FDs – Minimaal Alg. (Voorbeeld)

Vindt een minimale overdekking F van $E = \{B \rightarrow A, D \rightarrow A, AB \rightarrow D\}$:

Stap 1: Reeds in canonical vorm.

Stap 2: Enkel van toepassing op $AB \rightarrow D$

* $F = \{B \rightarrow A, D \rightarrow A, A \rightarrow D\} \Rightarrow B \in A_F^+ = \{A, D\}$? NOK

* $F = \{B \rightarrow A, D \rightarrow A, B \rightarrow D\} \Rightarrow A \in B_F^+ = \{B, A, D\}$? OK

$\Rightarrow F = \{B \rightarrow A, D \rightarrow A, B \rightarrow D\}$

Stap 3:

* $F = \{D \rightarrow A, B \rightarrow D\} \Rightarrow A \in B_F^+ = \{B, A, D\}$? OK

* $F = \{B \rightarrow D\} \Rightarrow A \in D_F^+ = \{D\}$? NOK

* $F = \{D \rightarrow A\} \Rightarrow D \in B_F^+ = \{B\}$? NOK

$\Rightarrow F = \{D \rightarrow A, B \rightarrow D\}$

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

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 supersleutel K voor R bestaat zodat $A \in K$.

FDs – Sleutels (Voorbeeld)

Gegeven:

EMPLOYEE

| | | | | | | | | | |
|-------|-------|-------|------------|-------|---------|-----|--------|-----------|-----|
| Fname | Minit | Lname | <u>Ssn</u> | Bdate | Address | Sex | Salary | Super_ssn | Dno |
|-------|-------|-------|------------|-------|---------|-----|--------|-----------|-----|

Supersleutels EMPLOYEE:

{Ssn}, {Ssn,Lname}, {Ssn,Fname}, ..., {Ssn,Lname,Fname}, ...

=> Elke verzameling attributen die Ssn bevat.

Kandidaatsleutels (of sleutels):

{Ssn}

FDs – Sleutels Algoritme

Gegeven relatie R met relatieschema $S_R = \langle U, F \rangle$:

$KS := \{\}$ en $K := U$

(Initialisatie)

func(K):

(Functie def.: Recursie)

Voor elk attribuut $A \in K$ doe:

(Loop over alle attributen)

OldK := K

(Hou oude K bij voor test)

Als $U \subseteq (K \setminus \{A\})_F^+$ dan $K := (K \setminus \{A\})$

(Redundant attribuut?)

Als OldK = K dan $KS := KS \cup \{K\}$ anders func(K)

(basisgeval + recurs. aanroep)

* **Optimalisatie**: Attributen die niet rechts voorkomen in F kunnen nooit teruggekregen worden bij sluiting! => Geen zin deze te beschouwen.

FDs – Sleutels Alg. (Voorbeeld)

Bereken kandidaatsleutels gegeven relatie R met relatieschema $S_R = \langle U, F \rangle$ waarbij $U = \{A, B, C, D, E\}$ en $F = \{AB \rightarrow C, CD \rightarrow E, DE \rightarrow B\}$

Merk op: A en D kunnen nooit teruggekregen worden => het weglaten ervan negeren

1. $K = \{A, B, C, D, E\}$ voor B: $(K \setminus \{B\})_F^+ = \{A, C, D, E, B\} \Rightarrow$ B mag weg

1.1. $K = \{A, C, D, E\}$ voor C: $(K \setminus \{C\})_F^+ = \{A, D, E, B, C\} \Rightarrow$ C mag weg

1.1.1. $K = \{A, D, E\}$ voor E: $(K \setminus \{E\})_F^+ = \{A, D\} \Rightarrow$ E mag niet weg

$\Rightarrow \{A, D, E\}$ is een kandidaatsleutel

1.2. $K = \{A, C, D, E\}$ voor E: $(K \setminus \{E\})_F^+ = \{A, C, D, E, B\} \Rightarrow$ E mag weg

1.2.1. $K = \{A, C, D\}$ voor C: $(K \setminus \{C\})_F^+ = \{A, D\} \Rightarrow$ C mag niet weg

$\Rightarrow \{A, C, D\}$ is een kandidaatsleutel

FDs – Sleutels Alg. (Voorbeeld)

Bereken kandidaatsleutels gegeven relatie R met relatieschema $S_R = \langle U, F \rangle$ waarbij $U = \{A, B, C, D, E\}$ en $F = \{AB \rightarrow C, CD \rightarrow E, DE \rightarrow B\}$

...

$\Rightarrow \{A, D, E\}$ is een kandidaatsleutel

...

$\Rightarrow \{A, C, D\}$ is een kandidaatsleutel

2. $K = \{A, B, C, D, E\}$ voor C: $(K \setminus \{C\})_F^+ = \{A, B, D, E, C\} \Rightarrow C$ mag weg

2.1. $K = \{A, B, D, E\}$ voor B: $(K \setminus \{B\})_F^+ = \{A, D, E, B, C\} \Rightarrow B$ mag weg: $K = \{A, D, E\}$ is KS

2.2. $K = \{A, B, D, E\}$ voor E: $(K \setminus \{E\})_F^+ = \{A, B, D, C, E\} \Rightarrow E$ mag weg

2.2.2. $K = \{A, B, D\}$ voor C: $(K \setminus \{B\})_F^+ = \{A, D\} \Rightarrow B$ mag niet weg

$\Rightarrow \{A, B, D\}$ is een kandidaatsleutel

FDs – Sleutels Alg. (Voorbeeld)

Bereken kandidaatsleutels gegeven relatie R met relatieschema $S_R = \langle U, F \rangle$ waarbij $U = \{A, B, C, D, E\}$ en $F = \{AB \rightarrow C, CD \rightarrow E, DE \rightarrow B\}$

...

$\Rightarrow \{A, D, E\}$ is een kandidaatsleutel

...

$\Rightarrow \{A, C, D\}$ is een kandidaatsleutel

...

$\Rightarrow \{A, B, D\}$ is een kandidaatsleutel

3. $K = \{A, B, C, D, E\}$ voor E: $(K \setminus \{E\})_F^+ = \{A, B, C, D, E\} \Rightarrow E$ mag weg

3.1. $K = \{A, B, C, D\}$ voor B: $(K \setminus \{B\})_F^+ = \{A, C, D, E, B\} \Rightarrow B$ mag weg: $K = \{A, C, D\}$ is KS

3.2. $K = \{A, B, C, D\}$ voor C: $(K \setminus \{C\})_F^+ = \{A, B, D, C, E\} \Rightarrow C$ mag weg: $K = \{A, B, D\}$ is KS

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

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 $\delta(S_R) = \{S_{R_1}, S_{R_2}, \dots, S_{R_k}\}$ met $k > 1$ is een decompositie van S_R a.s.a. $U_R = U_{R_1} \cup U_{R_2} \cup \dots \cup U_{R_k}$.

Normaalvormen

Voorbeelden decompositie: S_R met $U_R = ABCD$

$$\delta_1(S_R) = \{S_{R11}, S_{R12}\}$$

$$\text{met } U_{R11} = AB \text{ en } U_{R12} = CD$$

$$\delta_2(S_R) = \{S_{R21}, S_{R22}, S_{R23}\}$$

$$\text{met } U_{R21} = AB \text{ en } U_{R22} = BCD \text{ en } U_{R23} = ABC$$

$$\delta_3(S_R) = \{S_{R31}, S_{R32}\}$$

$$\text{met } U_{R31} = ABC \text{ en } U_{R32} = BCD$$

$$\delta_4(S_R) = \{S_{R41}, S_{R42}, S_{R43}\}$$

$$\text{met } U_{R41} = AB \text{ en } U_{R42} = BC \text{ en } U_{R43} = AD$$

1^e 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.

1NF (Voorbeeld)

Meerwaardig attribuut Dlocations (zie b).
=> niet in 1NF

Naar 1NF, optie 1:

=> Meerdere tupels (zie c):

- * Primaire sleutel uitbreiden
- * Redunantie vergroot als gevolg!

Naar 1NF, (betere) optie 2:

=> Refererende tabel (zie eerder)

- * DEPT_LOCATIONS
- * Verwijssleutel naar Dnumber
- * Prim. Sleutel {Dnumber, Dlocation}

(a)

DEPARTMENT

| Dname | <u>Dnumber</u> | Dmgr_ssn | Dlocations |
|-------|----------------|----------|------------|
| | | | |

(b)

DEPARTMENT

| Dname | <u>Dnumber</u> | Dmgr_ssn | Dlocations |
|----------------|----------------|-----------|--------------------------------|
| Research | 5 | 333445555 | {Bellaire, Sugarland, Houston} |
| Administration | 4 | 987654321 | {Stafford} |
| Headquarters | 1 | 888665555 | {Houston} |

(c)

DEPARTMENT

| Dname | <u>Dnumber</u> | Dmgr_ssn | <u>Dlocation</u> |
|----------------|----------------|-----------|------------------|
| Research | 5 | 333445555 | Bellaire |
| Research | 5 | 333445555 | Sugarland |
| Research | 5 | 333445555 | Houston |
| Administration | 4 | 987654321 | Stafford |
| Headquarters | 1 | 888665555 | Houston |

2^e 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^e 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 (Alg.)

Gegeven $S_{R_1} = \langle U_{R_1}, F_{R_1} \rangle$ in 1NF:

* Voor elk attribuut A

dat partiëel functioneel afhankelijk is v/e kandidaatsleutel K:

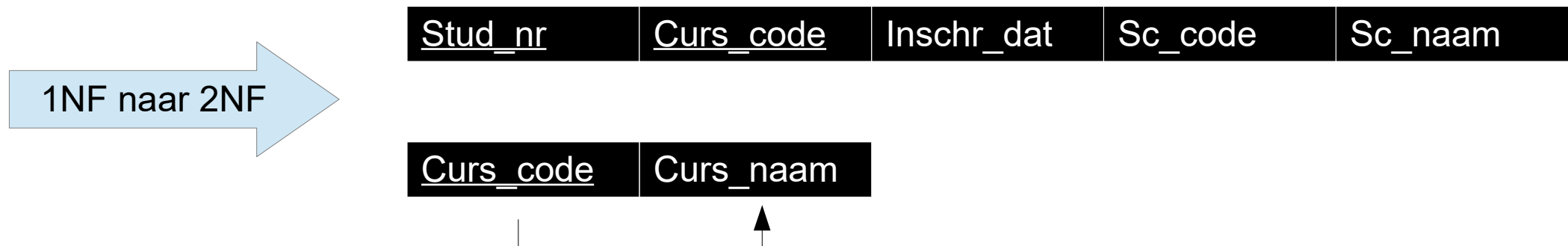
1. Zoek een deelverzameling K' van K waarvan A volledig functioneel afhankelijk is.
2. Elimineer A uit U_{R_1} alsook de FDs uit F_{R_1} aangaande A: F_A .
3. Maak een nieuw relatieschema S_{R_2} met attributenverzameling $U_{R_2} = K' \cup A$ en $F_{R_2} \subseteq F_A$ die betrekking hebben op U_{R_2} .

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 $\{\text{Curs_code}\} \subset K$.

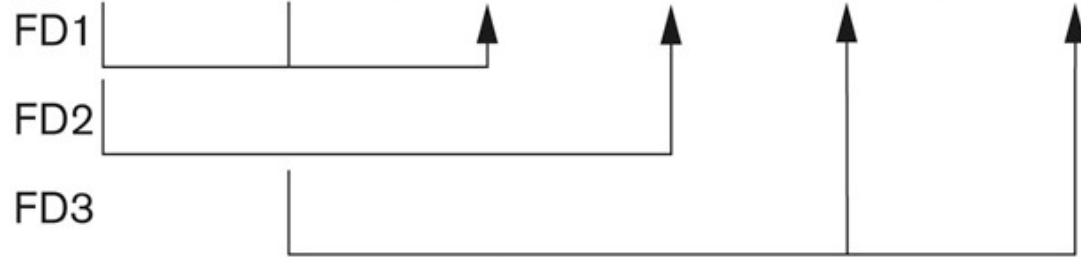


2NF – 1NF naar 2NF (Voorbeeld)

(a)

EMP_PROJ

| | | | | | |
|------------|----------------|-------|-------|-------|-----------|
| <u>Ssn</u> | <u>Pnumber</u> | Hours | Ename | Pname | Plocation |
|------------|----------------|-------|-------|-------|-----------|



Sleutel K = {Ssn, Pnumber}

- * {Hours} volledig afhankelijk v. K => OK
- * {Ename} partieel afhankelijk v. K => NOK
 - Volledig afhankelijk v. {Ssn} ⊂ K
- * {Pname, Plocation} part. afh => NOK
 - Volledig afhankelijk v. {Pnumber} ⊂ K

2NF Normalization

EP1

| | | |
|------------|----------------|-------|
| <u>Ssn</u> | <u>Pnumber</u> | Hours |
|------------|----------------|-------|



EP2

| | |
|------------|-------|
| <u>Ssn</u> | Ename |
|------------|-------|



EP3

| | | |
|----------------|-------|-----------|
| <u>Pnumber</u> | Pname | Plocation |
|----------------|-------|-----------|



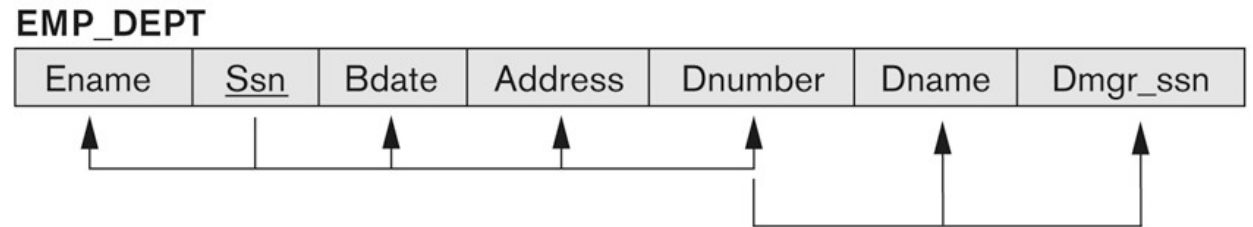
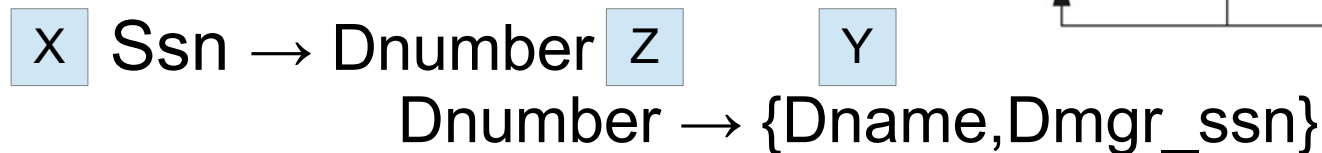
3^e Normaalvorm (3NF)

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^e 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 .

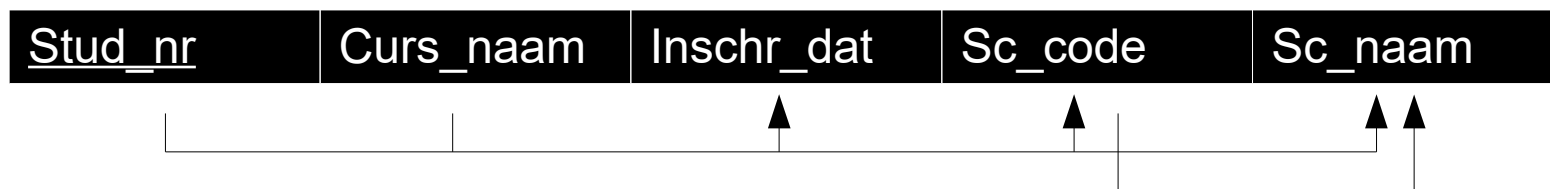
Of ook: Elk niet-sleutelattribuut in R is niet partieel en ook niet transitief functioneel afhankelijk van een sleutel van R .

3NF – 2NF naar 3NF

Gegeven $S_{R_1} = \langle U_{R_1}, F_{R_1} \rangle$ in 2NF:

- * Voor elk niet-sleutel attribuut A dat transitief func. afh. is v/e kandidaatsleutel K via Z:
 1. Elimineer A uit U_{R_1} alsook de FDs uit F_{R_1} aangaande A: F_A .
 2. Maak een nieuw relatieschema S_{R_2} met attributenverzameling $U_{R_2} = Z \cup A$ en $F_{R_2} \subseteq F_A$ die betrekking hebben op U_{R_2} .

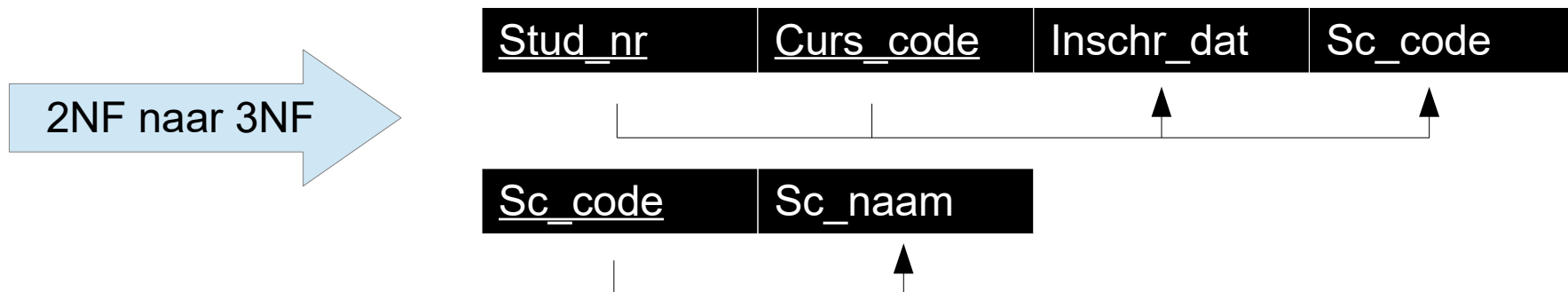
3NF – 2NF naar 3NF (Voorbeeld)



Sc_naam is volledig bepaald door Sc_code.

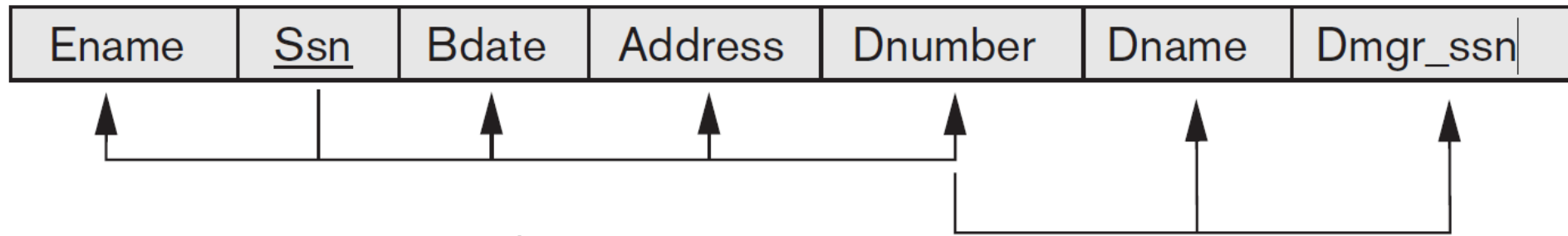
Sc_code bepaald door kandidaatsleutel.

Sc_naam transitief afhankelijk van kandidaatsleutel.



3NF – 2NF naar 3NF (Voorbeeld)

EMP_DEPT

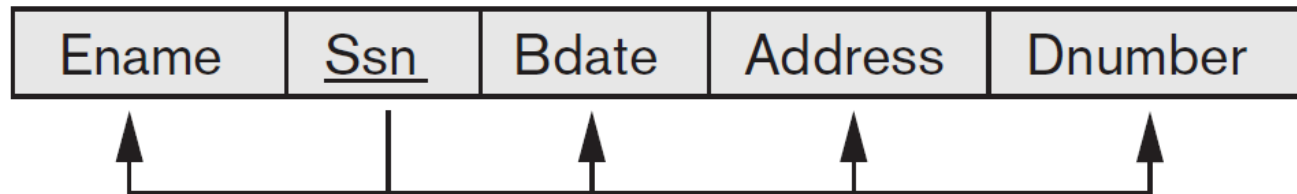


3NF Normalization

Sleutel $K = \{Ssn\}$

* $\{Dname, Dmgr_ssn\}$ trans. func. Afh. v. K via $\{Dnumber\}$

ED1



ED2

