9. Logischer DB-Entwurf

- **Vorgehensweisen beim Entwurf eines relationalen Schemas**
  - Normalisierung
  - Synthese

- **Definitionen und Begriffe**
  - Funktionale Abhängigkeiten, Schlüssel
  - Bestimmung funktionaler Abhängigkeiten
  - Axiomensystem nach Armstrong

- **Normalformenlehre**
  - Erste Normalform (1NF), 2NF, 3NF
  - BCNF, 4NF, 5NF und weitere (nicht behandelt)

Entwurf eines relationalen DB-Schemas

- **ZIEL:**

  Theoretische Grundlage für den Entwurf eines „guten“ relationalen DB-Schemas (→ Entwurfstheorie, Normalisierungslehre)

- **GÜTE:**
  - leichte Handhabbarkeit, Verständlichkeit, Natürlichkeit, Übersichtlichkeit, ...
  - Entwurfstheorie präzisiert/formalisiert „Güte“ z. T.

- **Beispiele**

  TABELLE1 (<u>A1</u>, A2, A3, ..., A300)

  ABTMGR (<u>ANR</u>, ANAME, BUDGET, MNR, PNAME, TITEL, SEIT_JAHR)

- **Was macht einen schlechten DB-Schema-Entwurf aus?**
  - implizite Darstellung von Informationen
  - Redundanzen, potentielle Inkonsistenz (Änderungsanomalien)
  - Einfügeanomalien, Löschanomalien
  - ...

    → *oft hervorgerufen durch „Vermischung“ von Entities, Zerlegung und wiederholte Speicherung von Entities, ...*

- **Normalisierung von Relationen**
  hilft einen gegebenen Entwurf zu verbessern

- **Synthese von Relationen**
  zielt auf die Konstruktion eines „optimalen“ DB-Schemas ab

Normalisierung von Relationen

PNR → ANR
ANR → ANAME

funktionale Abhängigkeiten

R (PNR, ANR, ANAME)

Anfängliche Relationen-Schemata

Normalisierungs-prozeß

R1 (PNR, ANR)
R2 (ANR, ANAME)

Relationen-Schemata in 3NF

Synthese von Relationen

PNR → ANR
ANR → ANAME

funktionale Abhängigkeiten

PNR, ANR, ANAME

Attribute

Synthese-prozeß

R1 (PNR, ANR)
R2 (ANR, ANAME)

Relationen-Schemata in 3NF

Funktionale Abhängigkeit

- **Konventionen**

R, S	Relationenschemata (Relationenname, Attribute)
R, S	Relationen der Relationenschemata **R, S**
A, B, C,...	einfache Attribute
A = $\{A_1,...,A_n\}$	Attributmenge des Relationenschemas
W, X, Y, Z,...	Mengen von Attributen
a, b, c	Werte einfacher Attribute
x, y, z	Werte von X, Y, Z
$XY \equiv X \cup Y$	Mengen brauchen nicht disjunkt zu sein

- **Def.: Funktionale Abhängigkeit (FA)**
  (engl. functional dependency)

  Die FA $X \rightarrow Y$ gilt (X bestimmt Y funktional), wenn für alle R von **R** gilt: Zwei Tupel, deren Komponenten in X übereinstimmen, stimmen auch in Y überein.

  $\forall u \in R \ \forall v \in R \ (u[X] = v[X]) \Rightarrow (u[Y] = v[Y])$

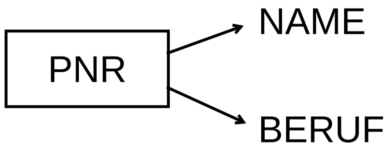
  alternativ:

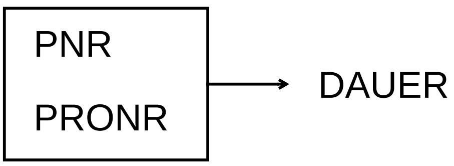
  Die Relation R erfüllt die FA $X \rightarrow Y$, wenn für jeden X-Wert x $\pi_Y(\sigma_{X=x}(R))$ höchstens ein Tupel hat.

- **Notation**

  $\{PNR\} \rightarrow \{NAME, BERUF\}$: verkürzt $PNR \rightarrow NAME, BERUF$

  $\{PNR, PRONR\} \rightarrow \{DAUER\}$: verkürzt $PNR, PRONR \rightarrow DAUER$





Funktionale Abhängigkeit (2)

- **Beispiel**

  Gegeben sei die Relation R mit dem Schema **R** = {A, B, C, D} und der FA A $\rightarrow$ B.

R

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_1	c_1	d_2
a_2	b_2	c_3	d_2
a_3	b_2	c_4	d_3
a_4	b_2	c_4	d_3

  Welche weiteren FA‘s erfüllt die gezeigte Relation R?

- **Triviale funktionale Abhängigkeit**

  Funktionale Abhängigkeiten, die von jeder Relationenausprägung automatisch immer erfüllt sind, nennt man triviale FA‘s.
  Nur FA‘s der Art X $\rightarrow$ Y mit Y $\subseteq$ X sind trivial.

  Es gilt also **R** $\rightarrow$ **R**

- **Achtung:**

  - FA‘s lassen sich nicht durch Analyse einer Relation R gewinnen.
    Sie sind vom Entwerfer festzulegen.

  - FA‘s beschreiben semantische Integritätsbedingungen bezüglich der Attribute eines Relationen*schemas*, die jederzeit erfüllt sein müssen

Schlüssel

- **Superschlüssel**

  - Im Relationenschema **R** ist $X \subseteq \mathbf{R}$ ein Superschlüssel, falls gilt:

    $$X \rightarrow \mathbf{R}$$

  - Falls X Schlüsselkandidat von **R**, dann gilt für alle Y aus **R**:

    $$X \rightarrow Y$$

  ➥ Wir benötigen das Konzept der vollen funktionalen Abhängigkeit, um Schlüssel (-kandidaten) von Superschlüsseln abzugrenzen.

- **Volle funktionale Abhängigkeit**

  Y ist voll funktional abhängig von X, wenn gilt

  1. $X \rightarrow Y$,
  2. X ist „minimal", d. h.
     $\forall A_i \in X : X - \{A_i\}$ -/-> Y

  ➥ Y ist funktional abhängig von X, aber nicht funktional abhängig von einer echten Teilmenge von X

- **Beispiel**

  Eine Stadt werde beschrieben durch Name, BLand (Bundesland), EW (Einwohnerzahl) und VW (Vorwahl)

  Stadt

Name	BLand	EW	VW
K'lautern	Rlp	100 000	0631
Mainz	Rlp	250 000	06131
Frankfurt	Bdg	90 000	0335
Frankfurt	Hes	700 000	069
. . .			

  - Superschlüssel

  - Schlüsselkandidaten

Bestimmung funktionaler Abhängigkeiten

- **Informationsbedarfsanalyse liefert:**
  - Menge aller Attribute
  - Menge F der funktionalen Abhängigkeiten zwischen Attributen

    ➥ Achtung: F kann Redundanz enthalten!

- **Beispiel**
  - Attribute: PNR, SVNR, BERUF, ANR
  - Menge F der FA's:
    1. SVNR $\rightarrow$ BERUF
    2. PNR $\rightarrow$ SVNR, ANR
    3. SVNR, BERUF $\rightarrow$ PNR

  Gilt SVNR $\rightarrow$ ANR?

- **Definition:** ***Logische Implikation***

  Sei F eine Menge von FA für **R** und sei X $\rightarrow$ Y eine FA.
  Dann impliziert F X $\rightarrow$ Y logisch (F $\models$ X $\rightarrow$ Y), wenn jedes R aus **R**,
  das die FA in F erfüllt, auch X $\rightarrow$ Y erfüllt.

Bestimmung funktionaler Abhängigkeiten (2)

- **Axiome für funktionale Abhängigkeiten**

  ➥ Inferenzregeln zum Ableiten von FAs aus einer Menge gegebener FAs

  - vollständig: aus F lassen sich alle FAs in F^+ ableiten
  - korrekt (*sound*): es wird aus F keine FA abgeleitet, die nicht in F^+ ist

- **Axiomensystem nach Armstrong**

A1: (Reflexivität):
Wenn $Y \subseteq X \subseteq \mathbf{A}$, dann $X \rightarrow Y$

A2: (Verstärkung):
$X \rightarrow Y \models XW \rightarrow YZ \quad (Z \subseteq W \subseteq \mathbf{A})$

A3: (Pseudotransitivität):
$X \rightarrow Y,\ YW \rightarrow Z \models XW \rightarrow Z$

R4: (Vereinigung, Additivität):
$X \rightarrow Y,\ X \rightarrow Z \models X \rightarrow YZ$

R5: (Zerlegung):
$X \rightarrow YZ \models X \rightarrow Y$

A1 – A3 sind vollständig und korrekt

Bestimmung funktionaler Abhängigkeiten (3)

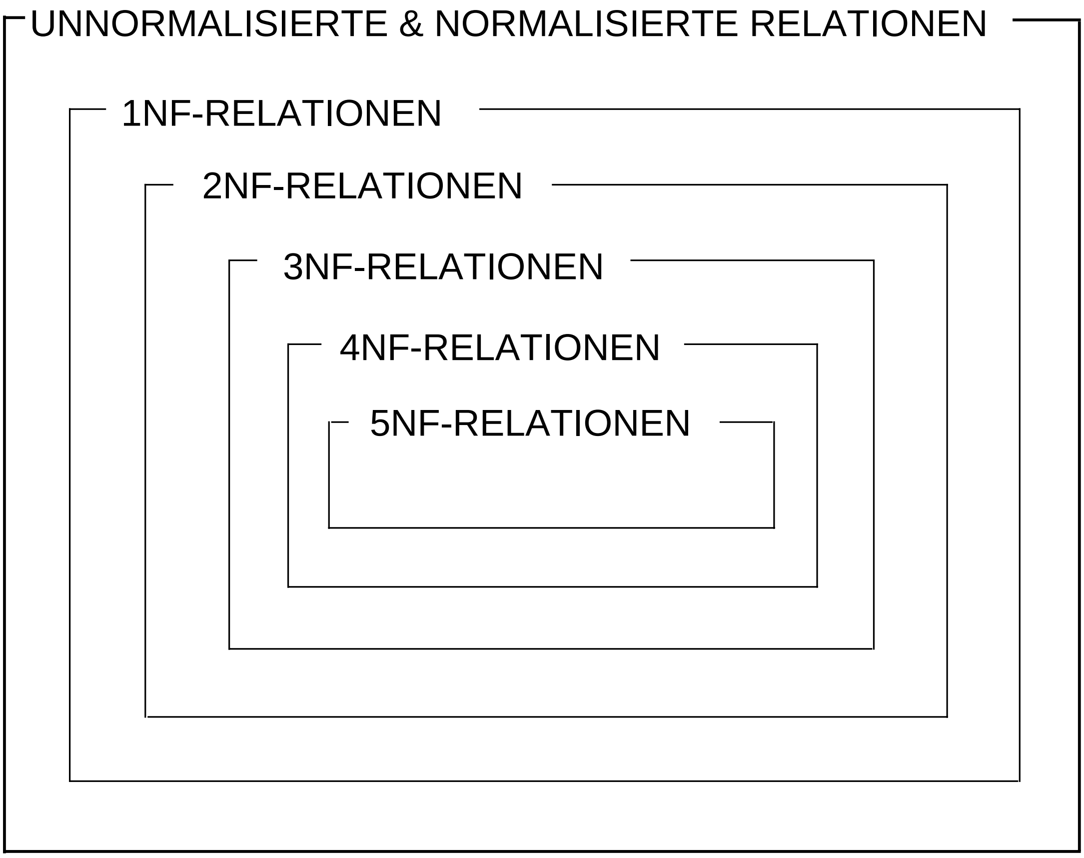
- **Beispiel zur Miniwelt „Universität“**
  - Attribute: PNR, PNAME, FACH, NOTE, PDAT
    MATNR, NAME, GEB, ADR, FBNR, FBNAME, DEKAN
  - Menge F der FA‘s:

    1. PNR $\rightarrow$ PNAME, FACH
    2. MATNR $\rightarrow$ NAME, GEB, ADR, FBNR
    3. NAME, GEB, ADR $\rightarrow$ MATNR
    4. PNR, MATNR, FBNR $\rightarrow$ NOTE, PDAT
    5. FBNR $\rightarrow$ FBNAME
    6. DEKAN $\rightarrow$ FBNR
    7. FBNAME $\rightarrow$ DEKAN

    Ist MATNR $\rightarrow$ DEKAN ableitbar?

    Ist PNR, MATNR, FBNR Schlüsselkandidat?

Normalisierung von Relationen



- **Zerlegung eines Relationenschemas R in höhere Normalformen**
  - Beseitigung von Anomalien bei Änderungsoperationen
  - Erhaltung aller nicht-redundanter Funktionalabhängigkeiten von **R**
    (→ sie bestimmen den Informationsgehalt von **R**)
  - fortgesetzte Anwendung der Projektion im Zerlegungsprozeß
  - Gewährleistung der Rekonstruktion von R durch verlustfreie Verbunde
  - bessere „Lesbarkeit" der aus R gewonnenen Relationen

Normalisierung von Relationen

- **Unnormalisierte Relation: Non-First Normal-Form (NF^2)**

PRÜFUNGSGESCHEHEN

(PNR,	PNAME,	FACH,	STUDENT	(MATNR,	NAME, ...))
1	HÄRDER	DBS		1234	MÜLLER
				5678	Maier
				9000	Schmitt
2	SCHOCK	FA		5678	Maier
				007	Coy

Relation enthält „Attribute“, die wiederum Relationen sind

➥ Darstellung von komplexen Objekten (Hierarchische Sichten)

- VORTEILE: Clusterbildung,
  Effiziente Verarbeitung in einem hierarchisch
  strukturierten Objekt längs der Vorzugsrichtung
- NACHTEILE: Unsymmetrie (nur eine Richtung der Beziehung),
  implizite Darstellung von Information,
  Redundanzen bei (n:m)-Beziehungen,
  Anomalien bei Aktualisierung,
  Definiertheit des Vaters

- **Normalisierung:**
  - „Herunterkopieren“ von Werten führt hohen Grad an Redundanz ein

    → Zerlegung von Relationen
  - aber: Erhaltung ihres Informationsgehaltes

Unnormalisierte Relation:

PRÜFUNGSGESCHEHEN (PNR, PNAME, FACH, STUDENT)

(MATNR, NAME, GEB, ADR, FBNR, FBNAME, DEKAN, PDAT, NOTE)

STUDENT = Wiederholungsgruppe mit 9 einfachen
Attributen (untergeordnete Relation)

- **Normalisierung (Überführung in 1NF):**

1. Starte mit der übergeordneten Relation (Vaterrelation)
2. Nimm ihren Primärschlüssel und erweitere jede unmittelbar untergeordnete Relation damit zu einer selbständigen Relation
3. Streiche alle nicht-einfachen Attribute (untergeordnete Relationen) aus der Vaterrelation
4. Wiederhole diesen Prozeß ggf. rekursiv

- **REGELN:**
  - Nicht-einfache Attribute bilden neue Relationen
  - Primärschlüssel der übergeordneten wird an untergeordnete Relation angehängt ('copy down the key')

- **Relationenschema in 1NF:**

PRÜFER (PNR, PNAME, FACH)

PRÜFUNG (PNR, MATNR, NAME, GEB, ADR, FBNR, FBNAME, DEKAN, PDAT, NOTE)

Überführung in 2NF

- **Beobachtung**
  - 1NF verursacht immer noch viele Änderungsanomalien, da verschiedene Entity-Mengen in einer Relation gespeichert werden können bzw. aufgrund von Redundanz innerhalb einer Relation (Bsp.: PRÜFUNG)
  - 2NF vermeidet einige der Anomalien dadurch, indem nicht voll funktional (partiell) abhängige Attribute eliminiert werden

  ➥ **Separierung verschiedener Entity-Mengen in eigene Relationen**

- **Definition:**

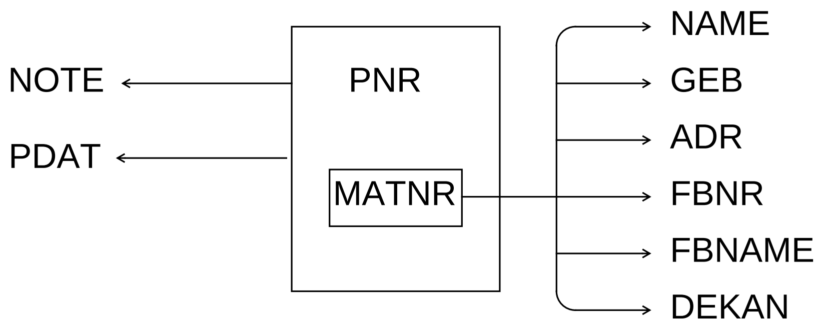
  Ein **Primärattribut** (Schlüsselattribut) eines Relationenschemas ist ein Attribut, das zu mindestens einem Schlüsselkandidaten des Schemas gehört.

  Ein Relationenschema **R** ist in ***2NF***, wenn es in 1NF ist und jedes Nicht-Primär-attribut von **R** voll funktional von jedem Schlüsselkandidaten in **R** abhängt.

- **Überführung in 2NF:**

1. Bestimme funktionale Abhängigkeiten zwischen Nicht-Primärattributen und Schlüsselkandidaten
2. Eliminiere partiell abhängige Attribute und fasse sie in eigener Relation zusammen (unter Hinzunahme der zugehörigen Primärattribute)

Voll funktionale Abhängigkeiten in PRÜFUNG



Relationenschema in 2NF

PRÜFUNG’

PNR	MATNR	PDAT	NOTE
1234	123 766	221000	4
1234	654 711	140299	3
3678	196 481	210999	2
3678	123 766	020399	4
8223	226 302	120799	1

PRÜFER

PNR	PNAME	FACH
1234	Schock	FA
3678	Härder	DBS
8223	Neunzert	NM

STUDENT'

MATNR	NAME	GEB	ADR	FBNR	FBNAME	DEKAN
123 766	Coy	050576	XX	FB1	Mathematik	Franke
654 711	Abel	211175	XY	FB9	Informatik	Richter
196 481	Maier	010177	YX	FB9	Informatik	Richter
226 302	Schulz	310776	YY	FB1	Mathematik	Franke

Überführung in 3NF

- **Beobachtung**

  Änderungsanomalien in 2NF sind immer noch möglich aufgrund von transitiven Abhängigkeiten.

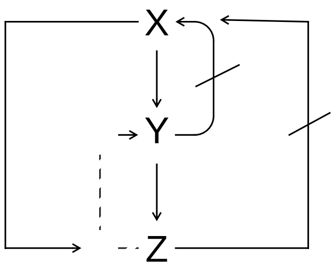
  Beispiel:
  Vermischung von Fachbereichs- und Studentendaten in Student'

- **Definition:**

  Eine Attributmenge Z von Relationenschema **R** ist transitiv abhängig von einer Attributmenge X in **R**, wenn gilt:

  - X und Z sind disjunkt
  - es existiert eine Attributmenge Y in **R**, so daß gilt:

$$X \longrightarrow Y, \; Y \longrightarrow Z, \; Y \not\longrightarrow X, \; Z \not\subseteq Y$$

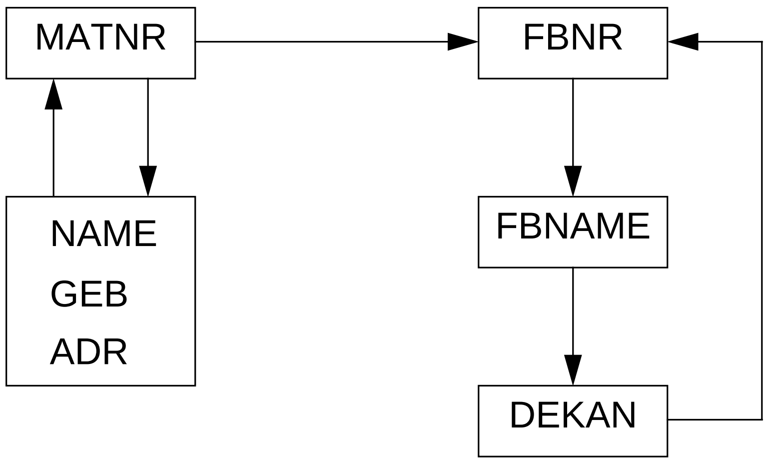


$Z \longrightarrow Y$ zulässig

strikte Transitivität: $Z \not\longrightarrow Y$

Ein Relationenschema **R** befindet sich in ***3NF***, wenn es sich in 2NF befindet und jedes Nicht-Primärattribut von **R** von keinem Schlüsselkandidaten von **R** transitiv abhängig ist.

Funktionale Abhängigkeiten in STUDENT'



Relationenschema in 3NF

PRÜFUNG'

PNR	MATNR	PDAT	NOTE
1234	123 766	221000	4
1234	654 711	140299	3
3678	196 481	210999	2
3678	123 766	020399	4
8223	226 302	120799	1

PRÜFER

PNR	PNAME	FACH
1234	Schock	FA
3678	Härder	DBS
8223	Neunzert	NM

STUDENT''

MATNR	NAME	GEB	ADR	FBNR
123 766	Coy	050576	XX	FB1
654 711	Abel	211175	XY	FB9
196 481	Maier	010177	YX	FB9
226 302	Schulz	310776	YY	FB1

FACHBEREICH

FBNR	FBNAME	DEKAN
FB1	Mathematik	Franke
FB9	Informatik	Richter
FB2	Physik	Jodl

Zusammenfassung

- **Festlegung aller funktionalen Abhängigkeiten**
  - unterstützt präzises Denken beim Entwurf
  - erlaubt Integritätskontrollen durch das DBS

- **<u>ZIEL:</u> klare und natürliche Zuordnung von Objekt und Datenstruktur**
  - ➥ „wachsender Informationsgehalt“ mit zunehmender Normalisierung
  - ➥ durch einen Satztyp (Relation) wird <u>nur ein</u> Objekttyp beschrieben

- **Normalisierung von Relationen**
  - lokales Verfahren auf existierenden Datenstrukturen
  - schrittweise Eliminierung von Änderungsanomalien
  - übergreifende Maßnahmen zur DB-Schema-Integration

1NF: Ein Relationenschema **R** ist in 1NF genau dann, wenn alle seine Wertebereiche nur atomare Werte besitzen.

2NF: Ein Relationenschema **R** ist in 2NF, wenn es in 1NF ist und jedes Nicht-Primärattribut von **R** voll funktional von jedem Schlüsselkandidaten von **R** abhängt.

3NF: Ein Relationenschema **R** ist in 3NF, wenn es in 2NF ist und jedes Nicht-Primärattribut von keinem Schlüsselkandidaten von **R** transitiv abhängig ist.