2. 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 und BCNF, 4NF
  - 5NF und weitere (nicht behandelt)

- **Entwurfstheorie für relationale DB**
  - Membership-Problem
  - Minimale Überdeckungen

- **Synthese von Relationen**
  - Synthese-Algorithmus von Beeri/Bernstein
  - Beispiele
  - Berücksichtigung von Anwendungsaspekten

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 (A1, A2, A3, ..., A300)

  ABTMGR (ANR, 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

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} → {NAME, BERUF}: verkürzt PNR → NAME, BERUF

  {PNR, PRONR} → {DAUER}: verkürzt PNR, PRONR → 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 $\subseteq$ **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 FA's aus einer Menge gegebener FA's

  - vollständig: aus F lassen sich alle FA's 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

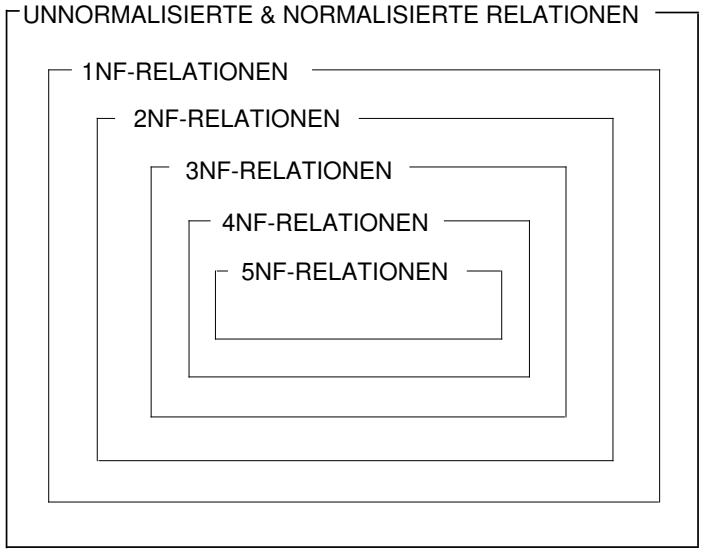
  - Menge F der FA‘s:

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

    Ist MATNR → 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

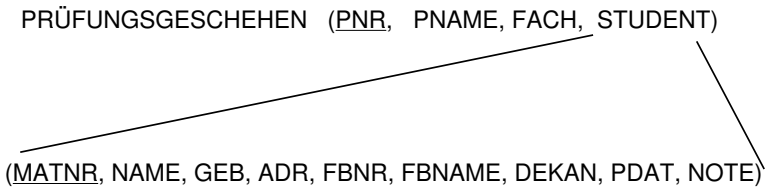
➥ Darstellung von komplexen Objekten (Hierarchische Sichten)

- VORTEILE: Clusterbildung,
  Effiziente Verarbeitung in einem hierarchischen,
  Objekt
- 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:



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:**

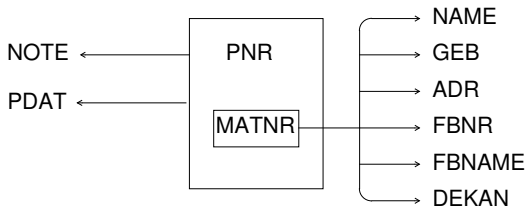
  Ein **Primärattribut** (Schlüsselattribut) eines Relationenschemas ist ein Attribut, das zu mindestens einem Schlüsselkandidaten des Schemas gehört.

  Ein Relationenschema **R** ist in ***2NF***, wenn es in 1NF ist und jedes Nicht-Primärattribut 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.

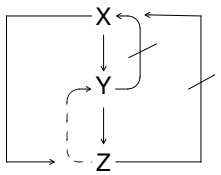
  Beispiel:
  Vermischung von Fachbereichs- und Studentendaten in Student'

- **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:

$$X \longrightarrow Y,\ Y \longrightarrow Z,\quad Y \not\longrightarrow X,\quad Z \notin Y$$

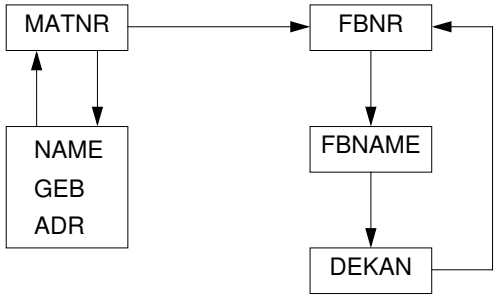


$Z \longrightarrow Y$ zulässig

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

Boyce/Codd-Normalform (BCNF)

- Definition der 3NF hat gewisse Schwächen bei **Relationen mit mehreren Schlüsselkandidaten**, wenn die Schlüsselkandidaten
  - zusammengesetzt sind und
  - sich überlappen
- **Beispiel**:

PRÜFUNG (PNR, MATNR, FACH, NOTE)
PRIMARY KEY (PNR, MATNR)
UNIQUE (FACH, MATNR)

  - es bestehe eine (1:1)-Beziehung zwischen PNR und FACH
  - einziges Nicht-Primärattribut: NOTE ➥ PRÜFUNG ist in 3NF

PRÜFUNG	(PNR ,	MATNR ,	FACH ,	NOTE)
	4	4711	BS	1
	4	1007	BS	2
	4	1234	BS	2
	5	4711	RO	3

➥ Änderungsanomalien z. B. bei FACH

- **ZIEL:**
  Ausschluß/Beseitigung der Anomalien in den Primärattributen

BCNF(2)

- **Definition:**

  Ein Attribut (oder eine Gruppe von Attributen), von dem andere voll funktional abhängen, heißt Determinant.

  Welches sind die Determinanten in PRÜFUNG?

- **Definition:**

  Ein Relationenschema **R** ist in ***BCNF***, wenn jeder Determinant ein Schlüsselkandidat von **R** ist.

- **Formale Definition:**

  Ein Relationenschema ist in ***BCNF***, falls gilt: Wenn eine Sammlung von Attributen Y (voll funktional) abhängt von einer disjunkten Sammlung von Attributen X, dann hängt jede andere Sammlung von Attributen Z auch von X (voll funktional) ab.

  D. h. für alle X, Y, Z mit X und Y disjunkt gilt:

  $$X \rightarrow Y \text{ impliziert } X \rightarrow Z$$

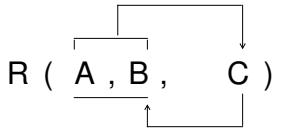
  - Zerlegung von Prüfung

PRÜF1	(PNR, MATNR, NOTE)	oder	PRÜF2	(MATNR, FACH, NOTE)
FBEZ	(PNR, FACH)		FBEZ	(PNR, FACH)

  - Beide Zerlegungen führen auf BCNF-Relationen
    - Änderungsanomalie ist verschwunden
    - alle funktionalen Abhängigkeiten sind erhalten
  - Wann ergeben sich nach BCNF- und 3NF-Definition gleiche Zerlegungen?

BCNF(3)

- **Sind BCNF-Zerlegungen immer sinnvoll?**



ist in 3NF, weil B ja Primäratttribut ist !

Beispiel: STUDENT, FACH → PRÜFER
PRÜFER → FACH

SFP (STUDENT FACH PRÜFER)

Sloppy	DBS	Härder
Hazy	DBS	Ritter
Sloppy	BS	Nehmer

- Jeder Prüfer prüft nur ein Fach
  (aber ein Fach kann von mehreren geprüft werden)
- Jeder Student legt in einem bestimmten Fach nur eine Prüfung ab

- Wie sieht die BCNF-Zerlegung aus?

- **Neue Probleme**
  - STUDENT, FACH **->** PRÜFER ist nun „extern“

    ➥ Konsistenzprüfung?
  - BCNF hier zu streng, um bei der Zerlegung alle funktionalen Abhängigkeiten zu bewahren (*key breaking dependency*)

Mehrwertige Abhängigkeiten (MWA)

- Eine FA bestimmt jeweils (höchstens) ein Wert des abhängigen Attributes
- MWA sind Generalisierungen von FA; sie bestimmen jeweils eine Menge von Werten des abhängigen Attributes
- MWA entstehen durch zwei (oder mehr) unabhängige Attribute im Schlüssel einer Relation (all-key relation): z.B. Fähigkeiten : Kinder
- Bsp.:

PNR	**FÄHIGK.**	**KIND**
123	Englisch	Nadine
123	Englisch	Philip
123	Englisch	Tobias
123	Progr.	Nadine
123	Progr.	Philip
123	Progr.	Tobias

➥ ***Änderungsanomalien*** (obwohl in BCNF)

- **Definition:**

  X, Y, Z seien Attributmengen des Relationenschemas **R**.

  Die ***mehrwertige Abhängigkeit (MWA)***

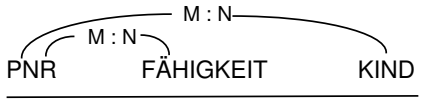
  $X \longrightarrow\!\!\!\!\!\rightarrow Y$

  gilt in **R** genau dann, wenn die Menge der Y-Werte, die zu einem (X-Wert, Z-Wert)-Paar gehören, nur vom X-Wert bestimmt sind (d. h. unabhängig vom Z-Wert sind).

  - MWA im Bsp.: PNR $\longrightarrow\!\!\!\!\!\rightarrow$ FÄHIGK., PNR $\longrightarrow\!\!\!\!\!\rightarrow$ KIND
  - X $\longrightarrow\!\!\!\!\!\rightarrow$ Y impliziert X $\longrightarrow\!\!\!\!\!\rightarrow$ Z
    Schreibweise: X $\longrightarrow\!\!\!\!\!\rightarrow$ Y | Z , z.B. PNR $\longrightarrow\!\!\!\!\!\rightarrow$ FÄHIGK. | KIND
  - Jede FA ist auch eine MWA.

4NF

- behandelt Probleme mit mehrwertigen Abhängigkeiten
- Schlüssel darf nicht 2 oder mehr unabhängige mehrwertige Fakten enthalten
- **Beispiel**



ZERLEGUNG:

andere Attribute sind erlaubt:

PNR	FÄHIGK.
123	Englisch
123	Prog.

PNR	KIND
123	Nadine
123	Philip
123	Tobias

- **Definition:**

  Ein Relationenschema **R** ist in ***4NF***, wenn es in BCNF ist und jede MWA in **R** eine FA ist.

- **Überführung in 4NF:**

  Zerlege Relationenschema mit MWA $X \twoheadrightarrow Y \mid Z$
  in zwei neue Relationenschemata mit den Attributen X, Y bzw. X, Z.

Abhängigkeit bei mehrwertigen Fakten

- Wenn Abhängigkeit besteht, muß sie durch die Wertekombinationen ausgedrückt werden

- **Beispiel:**
  - (M:N)-Beziehung zwischen : PNR - PROJEKT, PNR - FÄHIGKEIT
  - zusätzliche (M:N)-Beziehung zwischen PROJEKT und FÄHIGKEIT, d. h., Projektmitarbeit erfordert bestimmte Fähigkeiten

- **Gültige 4NF: R (PNR, PROJEKT, FÄHIGKEIT)**

PNR	**PROJEKT**	**FÄHIGKEIT**
123	P1	Progr.
123	P2	Progr.
123	P2	Englisch

➥ Zerlegung von R in zwei Projektionen R1 (PNR, PROJEKT) und R2 (PNR, FÄHIGKEIT) führt zu 'Verlust' von Information, da Join-Bildung auf den Projektionen vorher nicht existente Tupel generieren kann (*connection trap*).

- **Gibt es andere verlustfreie Zerlegungen?**
  - Es gibt Relationen, die nicht verlustfrei in zwei, aber unter bestimmten Bedingungen verlustfrei in n (n>2) Projektionen (n-zerlegbar) zerlegt werden können
  - Zerlegung der Beispielrelation R in drei Projektionen R1 (PNR, PROJEKT), R2 (PNR, FÄHIGKEIT) sowie R3 (PROJEKT, FÄHIGKEIT) ist verlustfrei, d.h. Join zwischen diesen drei Projektionen erzeugt genau die Ausgangsrelation !
  - Ist eine solche Zerlegung in drei Projektionen sinnvoll ?

Normalformenlehre nach E.F. Codd

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.

3NF (BCNF): Ein Relationenschema **R** ist in BCNF, falls gilt:
Wenn eine Sammlung von Attributen Y (voll funktional) abhängt von einer disjunkten Sammlung von Attributen X, dann hängt jede andere Sammlung von Attributen Z auch von X (voll funktional) ab.

D.h. für alle X, Y, Z mit X und Y disjunkt gilt:

$X \rightarrow Y$ impliziert $X \rightarrow Z$

- **Alternative Definition der BCNF:**

  Ein normalisiertes Relationenschema **R** ist in 3NF (BCNF), wenn jeder Determinant in **R** ein Schlüsselkandidat von **R** ist.

4NF: Ein Relationenschema **R** ist in 4NF, wenn es in BCNF ist und jede MWA auf **R** eine FA ist.

Entwurfstheorie für relationale Datenbanken

- **Erhebung der Miniwelt (Informationsbedarfsanalyse) liefert:**
  - Menge aller Attribute ***(universelles Relationenschema)***
  - Menge F der funktionalen Abhängigkeiten zwischen Attributen

  ➥ Synthese-Verfahren erzeugt daraus relationales DB-Schema in 3NF

- **Synthese erfordert u. a.**
  - Inferenzregeln, um aus F weitere FA abzuleiten *(Armstrong-Axiome)*
  - Ableitungsregeln zur Bestimmung der Schlüssel und zum Verstehen der logischen Implikationen (Eleimination von Redundanz)

- **Definition:** ***Hülle von F***

  F^+ ist die Menge der FA, die logisch durch F impliziert werden:
  $F^+ = \{X \rightarrow Y \mid F \models X \rightarrow Y\}$

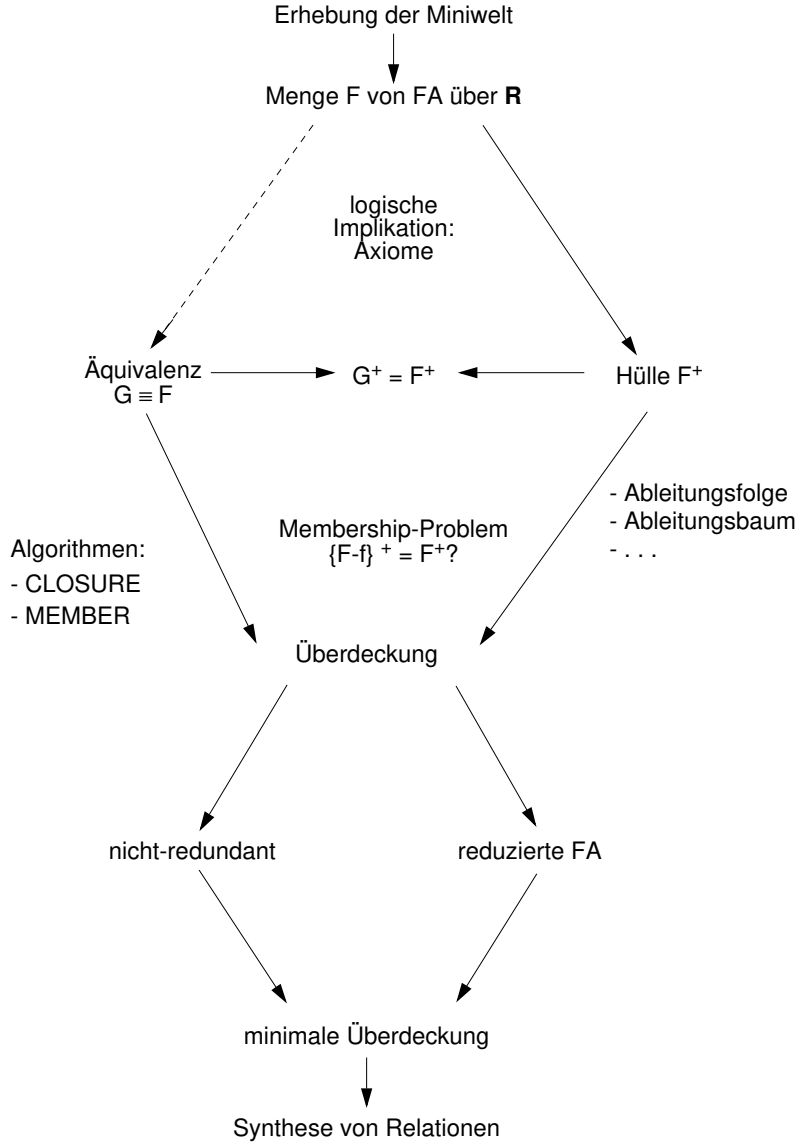
- **Bem.:**

  $F^+ = (F^+)^+$

- **Zwei Aufgaben:**

  1. Berechnung von F^+ aus F
  2. Gegeben F, $X \rightarrow Y$: Ist $X \rightarrow Y$ in F^+ ?
     Ist $X \rightarrow Y$ in $(F - \{X \rightarrow Y\})^+$?

Entwurfstheorie für relationale Datenbanken – Überblick



Entwurfstheorie für relationale Datenbanken (2)

- **Äquivalenz von Familien funktionaler Abhängigkeiten**

Def.: **Äquivalenz von FA-Mengen**

Seien F und G zwei Mengen von FA über **A**. F und G sind äquivalent,

wenn $F^+ = G^+$,

d. h., wenn ihre Hüllen gleich sind.

F heißt auch Überdeckung von G und umgekehrt.

- **Zerlegung**

Für eine gegebene Menge von Funktionalabhängigkeiten läßt sich eine äquivalente Menge von Funktionalrelationen mit jeweils einem Attribut auf der rechten Seite finden:

- Zerlegung mit Regel R5
- Wiedergewinnung der ursprünglichen Menge mit Regel R4
  (➥ Äquivalenz)

- **Minimale Überdeckungen**

Def.: Eine Menge von Funktionalabhängigkeiten ***F ist minimal***, wenn gilt:

1) Jede rechte Seite von einer FA in F besteht aus einem Attribut.
2) Es gibt kein $X \rightarrow A$ in F, so daß die Menge $F - \{X \rightarrow A\}$ äquivalent zu F ist.
3) Es gibt kein $X \rightarrow A$ in F und keine echte Untermenge Z von X, so daß $F - \{X \rightarrow A\} \cup \{Z \rightarrow A\}$ äquivalent zu F ist.

Membership-Problem für funktionale Abhängigkeiten

- **Aufgabe**
  - Test, ob eine Funktionalabhängigkeit f in F^+ enthalten ist
  - Ableitungen von FA's, z. B. zur Überprüfung von Redundanzen

1. Ableitungsfolge (Anwendung der Axiome)

Def.: Eine Folge P von FA's über **R** ist eine **Ableitungsfolge** auf F, wenn jede FA in P

1) entweder aus F ist oder
2) aus vorangehenden FA's in P durch die Axiome A1 – A3 folgt.

- **Haupteigenschaft einer Ableitungsfolge:**

Wenn $X \rightarrow Y$ durch P abgeleitet werden kann, dann ist $X \rightarrow Y \in F^+$.

2. Ableitungsbäume (graphische Methode)

Def.: **Ableitungsbaum**

1) Ein Knoten mit der Marke A ist ein Ableitungsbaum für F.
2) Wenn T ein Ableitungsbaum mit dem Knoten A als Blatt ist und $B_1, B_2, \ldots, B_m \rightarrow A \in F$ gilt, dann ist T' durch Anhängen von $B_1, B_2, \ldots, B_m$ auch ein Ableitungsbaum für F.

- **Haupteigenschaft eines Ableitungsbaumes:**

Y sei nicht-leere Menge von Knoten eines Ableitungsbaumes AB.
X sei die Menge aller Blätter von AB. Dann ist $X \rightarrow Y \in F^+$!

Entwurfstheorie – Syntheseverfahren

- **Gegeben:**
  - **A**; F (erhoben in der Miniwelt)
  - Modellannahme: universelles Relationenschema U enthält alle Attribute

- **Gesucht:**

Relationales DB-Schema RS mit folgenden Eigenschaften bezüglich U:

1) Informationsgleichheit *(lossless join decomposition)*
2) Abhängigkeitsbewahrung *(dependency preservation)*
3) Redundanzminimierung

zu 1: Jedes Attribut von U ist in mindestens einer Relation von RS enthalten. Die Zerlegung in mehrere Relationen ist verlustfrei

zu 2: Alle FA's der minimalen Überdeckung von F sind durch Schlüsselkandidaten in den Relationen von RS verkörpert

zu 3: Alle Relationen sind in 3NF; die Anzahl der Relationen ist minimal.

Syntheseverfahren – Voraussetzungen

1. Eindeutigkeitsannahme

Wenn $f : X \to Y$ und $g : X \to Y$, dann $f \equiv g$

Beispiel: f_1 : PNR $\to$ TELNR (Angest. benutzt Telefon)
f_2 : TELNR $\to$ ANR (Tel. wird abgerechnet über Abt.)

➥ abgeleitete FA

f_{12} : PNR $\to$ ANR („benutzt Telefon, das abgerechnet wird über“)

➥ i. allg. verschieden von der wohl **erhobenen FA**

f_3 : PNR $\to$ ANR (Angest. gehört zu Abt.)

➥ Problem der *Bedeutungstransitivität* !

2. Darstellung nicht-funktionaler Beziehungen

$X — Y$ n : m, d. h. $X \not\mapsto Y$ und $Y \not\mapsto X$

➥ $XY \to \Theta$ mit Θ „leeres Attribut“

Synthese-Algorithmus

- **Eingabe:** **A**; F
- **Ausgabe:** RS in 3NF mit minimaler Anzahl von Relationen

Schritt 1: Ermittle eine minimale Überdeckung H für F
($\to$ MINCOVER(F))

Schritt 2: Teile H in Partitionen mit gleichen linken Seiten auf

Schritt 3: Mische äquivalente Schlüssel
($\to$ Schlüsselkandidaten sollen derselben Relation zugeordnet werden)

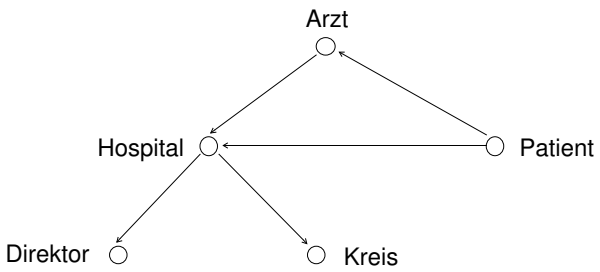
Schritt 4: Eliminiere transitive Abhängigkeiten ($\to$ H'), die durch Schritt 3 eingeführt wurden
($\to$ innerhalb von Schlüsselattributen)

Schritt 5: Konstruiere für jede Partition von H' eine Relation
($\to$ jede Attributmenge auf der linken Seite einer FA ist ein Schlüsselkandidat)

Anwendung des Syntheseverfahrens

Beispiel 1:

F:



Wie sieht F^+ aus?

H:

Schritt 1: H =

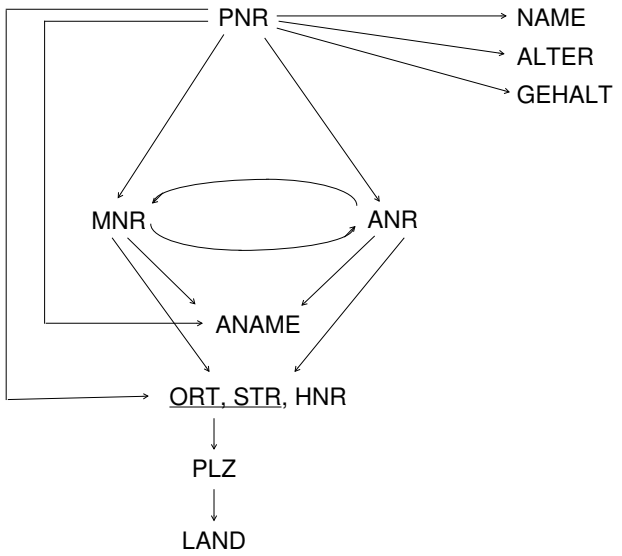
Schritt 2: g_1
g_2
g_3

Schritt 5: R1
R2
R3

Anwendung des Syntheseverfahrens (2)

Beispiel 2:

F:



f1: PNR → NAME
f2: PNR → ALTER
f3: PNR → GEHALT
f4: PNR → ANR
f5: PNR → MNR
f6: PNR → ORT, STR, HNR
f7: PNR → ANAME
f8: MNR → ANR
f9: MNR → ORT,STR,HNR
f10: MNR → ANAME
f11: ANR → MNR
f12: ANR → ANAME
f13: ANR → ORT, STR, HNR
f14: ORT, STR → PLZ
f15: PLZ → LAND

- Wieviele minimale Überdeckungen H existieren?
- Auswahl einer geeigneten Überdeckung: semantische Kriterien!

Anwendung des Syntheseverfahrens (3)

Eine Lösung:

R1 (PNR , NAME, ALTER, GEHALT, MNR)

R2 (MNR , ANR, ANAME, ORT, STR, HNR)

R3 (ORT, STR, PLZ)

R4 (PLZ, LAND)

Fragen:

1. **Wie häufig treten in**

   ORT, STR

   Wiederholungen auf?

2. **Ist die Zerlegung von**

   ORT, STR $\rightarrow$ PLZ $\rightarrow$ LAND

   in R3 und R4 **sinnvoll?**

   - Änderungshäufigkeit?
   - Aufsuchen der Adresse (Verbundoperation) !
   - Ist ORT, STR oder PLZ in diesem Kontext ein Entity?
     (als Kandidat für eigene Relation in 3NF)

     ➥ besser R2 in 2NF !

3. **Stabilität von MNR ?**

   - Änderungshäufigkeit von ANR und MNR!

     ➥ R1 (PNR , NAME, ALTER, GEHALT, ANR)
     R2 (ANR , MNR, ANAME, ORT, STR, HNR, PLZ, LAND)

Entwurfstheorie – Zusammenfassung

- **Festlegung aller funktionalen Abhängigkeiten**
  - unterstützt präzises Denken beim Entwurf
  - erlaubt Integritätskontrollen durch das DBS
- **ZIEL: klare und natürliche Zuordnung von Objekt und Datenstruktur**
  - ➥ „wachsender Informationsgehalt“ mit zunehmender Normalisierung
  - ➥ durch einen Satztyp (Relation) wird nur ein Objekttyp beschrieben
- **Normalisierung von Relationen**
  - lokales Verfahren auf existierenden Datenstrukturen
  - schrittweise Eliminierung von Änderungsanomalien
  - übergreifende Maßnahmen zur DB-Schema-Integration
- **Synthese von Relationen**
  - globales Verfahren liefert 3NF-Relationen
  - ggf. Überprüfung von überlappenden Schlüsselkandidaten, mehrwertigen Abhängigkeiten und Join-Abhängigkeiten
    - ➥ BCNF-, 4NF- bzw. 5NF-Zerlegung
- **Weitere Probleme**
  - Definition aller relevanten FA bei sehr vielen Attributen schwierig
  - Entwurfs-Algorithmen liefern i. allg. mehrere minimale Überdeckungen
  - Bei Überführung von 3NF in BCNF können FA’s verlorengehen
- **Überarbeitung des DB-Schemas**
  - Stabilitätsgesichtspunkte/Änderungshäufigkeiten können schwächere Normalformen erzwingen
  - Berücksichtigung von Abstraktionskonzepten
    - ➥ Der Entwerfer, und nicht die Methode, bestimmt den Entwurf

Einfacher Membership-Test

Def.: **Hülle einer Attributmenge**

X^+ ist ***Hülle einer Attributmenge*** X bzgl. F.

Sie umfaßt die Menge aller Attribute A_i, für die $X \rightarrow A_i$ in F^+ ist.

Satz: Test, ob FA in F^+

$X \rightarrow Y$ ist in F^+ genau dann, wenn $Y \subseteq X^+$

Bem.: Satz liefert einfache Möglichkeit zu entscheiden, ob eine vorgegebene FA in F^+ ist oder nicht

Einfacher Alg. zur Bestimmung von X^+; (F^+ schwierig)

Algorithmus CLOSURE

Eingabe: Menge von Attributen X und Menge von FA F

Ausgabe: X^+ bzgl. F.

```
CLOSURE(X,F)
    begin
        OLDDEP := ∅; NEWDEP := X;
        while NEWDEP ≠ OLDDEP do
        begin
            OLDDEP := NEWDEP;
            for jede FA Y → Z in F do
            if   Y ⊆ NEWDEP then
                 NEWDEP := NEWDEP ∪ Z
        end
        return (NEWDEP)
    end
```


Testen der Mitgliedschaft

Algorithmus MEMBER:

Eingabe: $X \rightarrow Y$, F

Ausgabe: TRUE, wenn $F \models X \rightarrow Y$, sonst FALSE

```
MEMBER (F, X → Y)
    begin
        if   Y ⊆ CLOSURE(X,F) then
             return (TRUE)
        else return (FALSE)
    end
```

Minimale Überdeckungen

Für eine gegebene Menge von Funktionalabhängigkeiten läßt sich eine äquivalente Menge von Funktionalrelationen mit jeweils einem Attribut auf der rechten Seite finden

Algorithmus MINCOVER:

Eingabe: Menge G von FA's mit jeweils minimaler linker Seite und einfacher rechter Seite

Ausgabe: minimale Überdeckung für G

```
MINCOVER (G)
begin
    F := G;
    for  jede FA X → Y in G do
        if   MEMBER (F - {X → Y}, X → Y) then
             F := F - {X → Y};
    return {F};
end
```