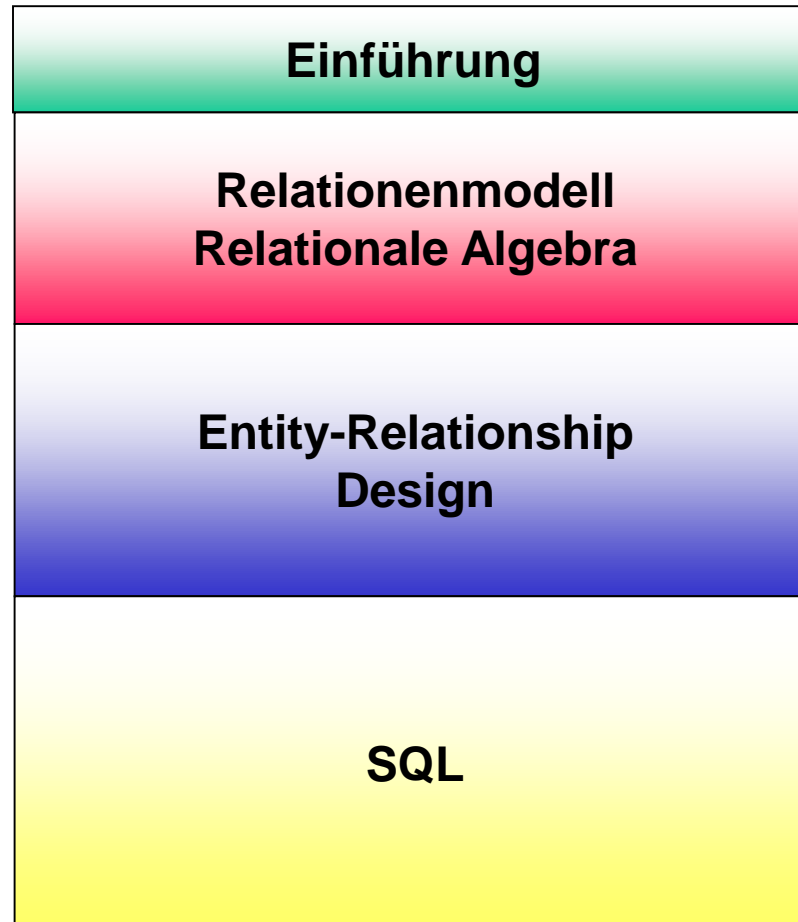


DAB1 – Datenbanken 1

Dr. Daniel Aebi (aebd@zhaw.ch)

Lektion 8: Korrektes ER-Diagramm, Normalisierung

Wo stehen wir?



← "You are here"

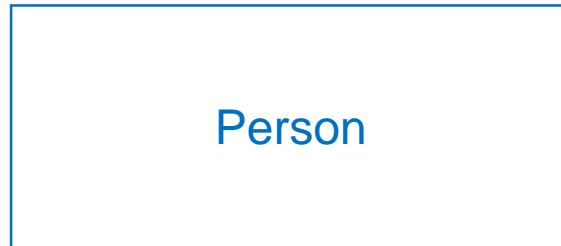
Rückblick

- ER-Modell, zusammenhängendes Beispiel: CD-Shop

Bemerkungen zur Notation

- **Entitätstyp**

- Symbol:

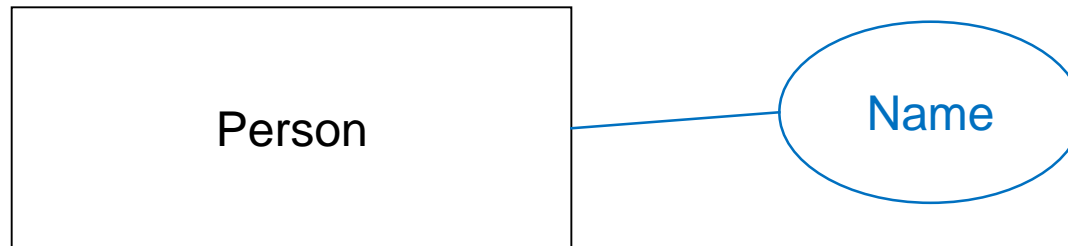


- Zusammenfassung gleichartiger **Entitäten** (z.B. werden Personen anhand gleicher Eigenschaften zusammengefasst). Es muss klar entscheidbar sein, was eine Entität eines Entitätstyps ist.
- Vereinbarung: Bezeichnung mit Substantiv singular.

Bemerkungen zur Notation

- **Attribut**: Eigenschaft eines **Entitäts-** oder **Beziehungstypen**
- **Attributwert**: Eigenschaft einer **Entität** oder **Beziehung**

- Symbol:

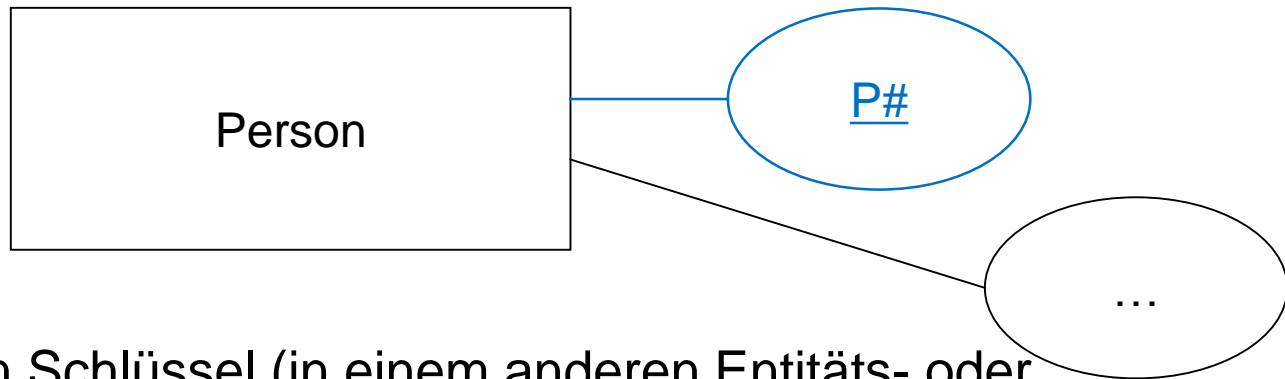


- Es gibt nur einfache Attribute (zusammengesetzte und mehrwertige Attribute müssen **transformiert** werden).
- (Schlüssel)attribute dürfen nicht NULL sein.

Bemerkungen zur Notation

- **Schlüssel**: Minimale Attributkombination, die jede Entität **eindeutig** identifiziert

- Symbol:

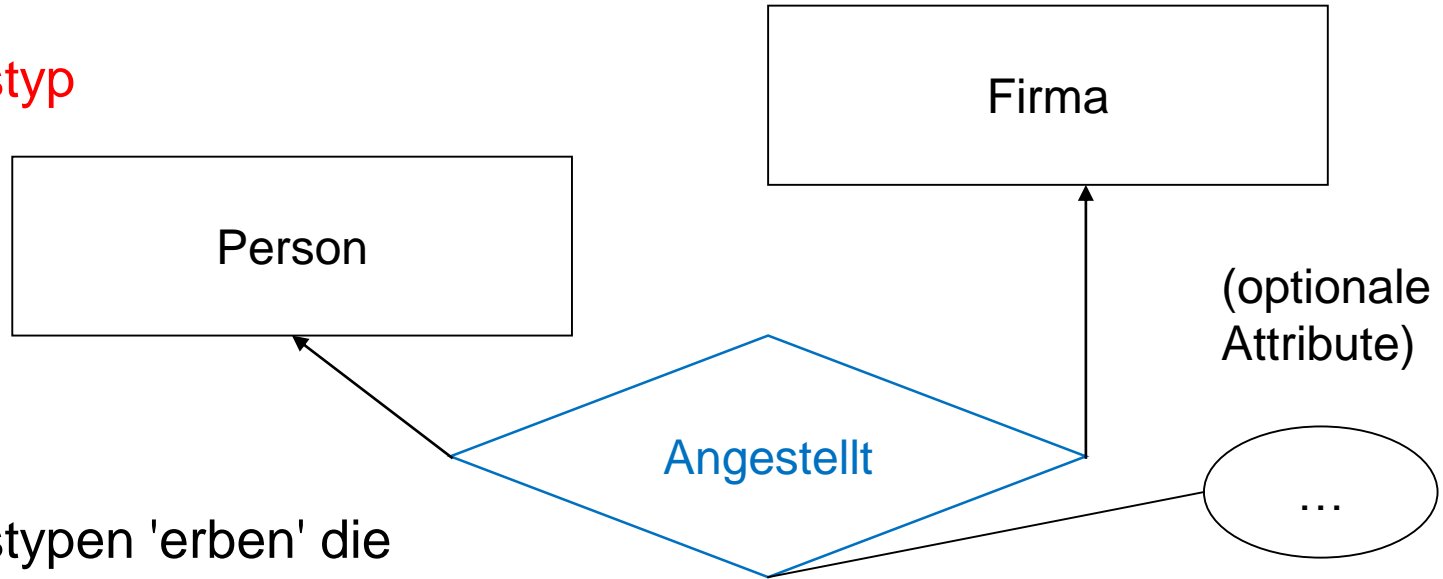


- Wenn auf einen Schlüssel (in einem anderen Entitäts- oder Beziehungstypen) durch einen Fremdschlüssel Bezug genommen wird, so wird er – als Primärschlüssel – **unterstrichen**.
- Sonstige Schlüssel werden **nicht unterstrichen** (aber ggf. als Text notiert)!
- JEDER Entitäts-/Beziehungstyp MUSS **mindestens einen Schlüssel** haben.

Bemerkungen zur Notation

- **Beziehungstyp**

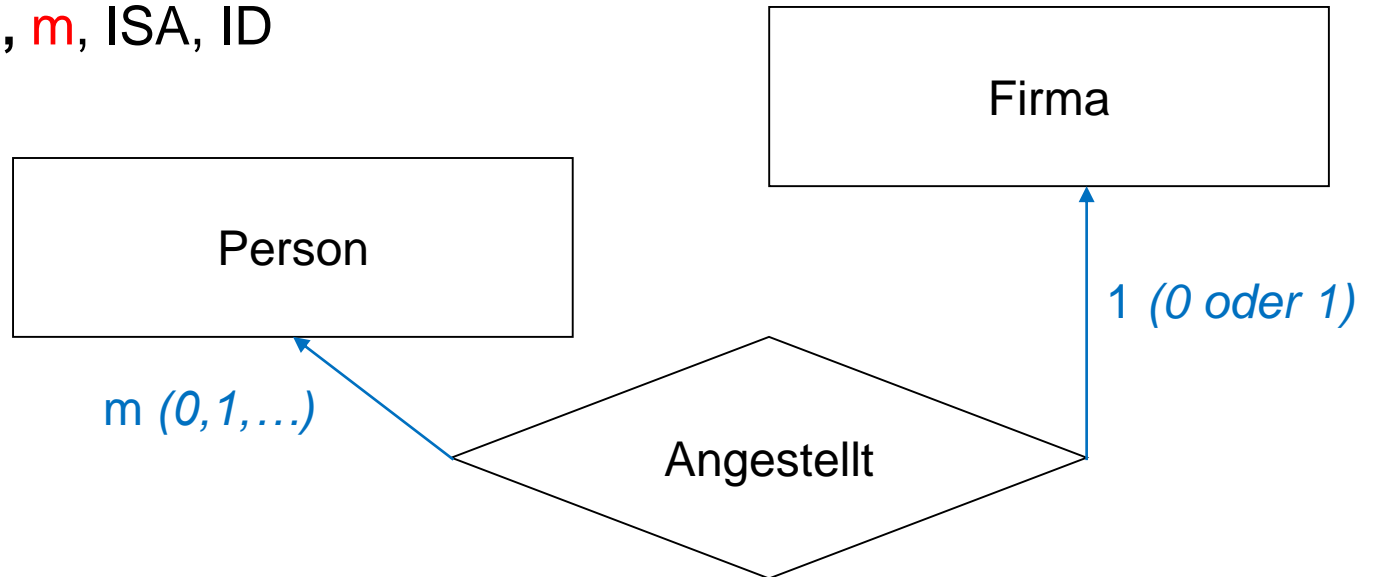
- Symbol:



- Beziehungstypen 'erben' die Schlüsselattribute der verbundenen Entitätstypen. Sie können optional weitere Attribute haben.
- Zusammenfassung gleichartiger Beziehungen. Vom Beziehungstypen führen Pfeile zu den betroffenen Entitätstypen. **Pfeile bezeichnen existenzielle Abhängigkeiten**. Kästchen mit eingehenden Pfeilen haben Primärschlüssel.

Bemerkungen zur Notation

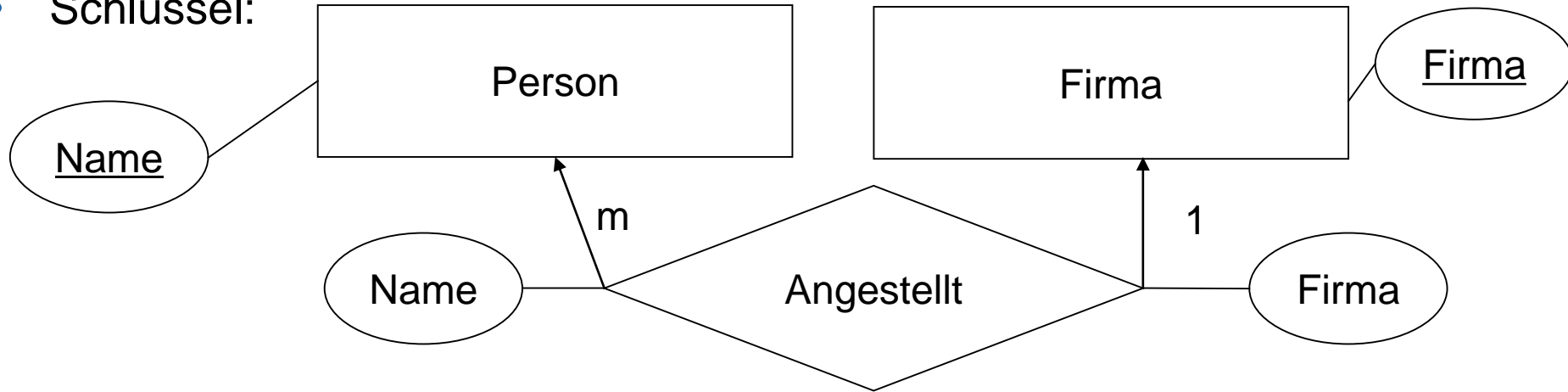
- Varianten: "Was steht auf dem Pfeil"?
- Möglichkeiten: **1**, **m**, ISA, ID
- Symbol:



- 1 bzw. m drücken aus, dass pro Person höchstens eine Firma als Arbeitgeber existiert, während eine Firma beliebig viele (auch 0!) Angestellte haben kann.

Bemerkungen zur Notation

- Schlüssel:



Person:

Meier

Müller

Huber

...

Angestellt:

{Meier,SAP}

{Müller,IBM}

{Meier,IBM}

...

Firma:

SAP

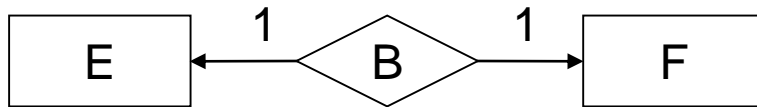
IBM

Microsoft

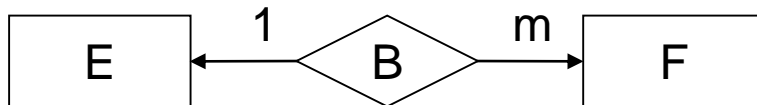
...

Bemerkungen zur Notation

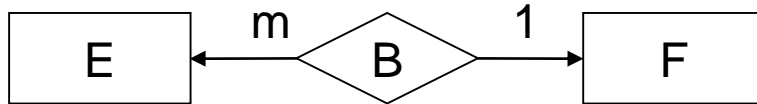
- Beziehungstypen haben Schlüssel, keine Primärschlüssel (Es sei denn, sie werden referenziert; siehe hierzu weiter unten)
- Mögliche Kombinationen: (inkl. passendem Schlüssel in B)



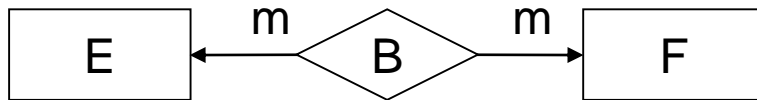
2 Schlüssel (von E **und** F)



1 Schlüssel (von F)



1 Schlüssel (von E)

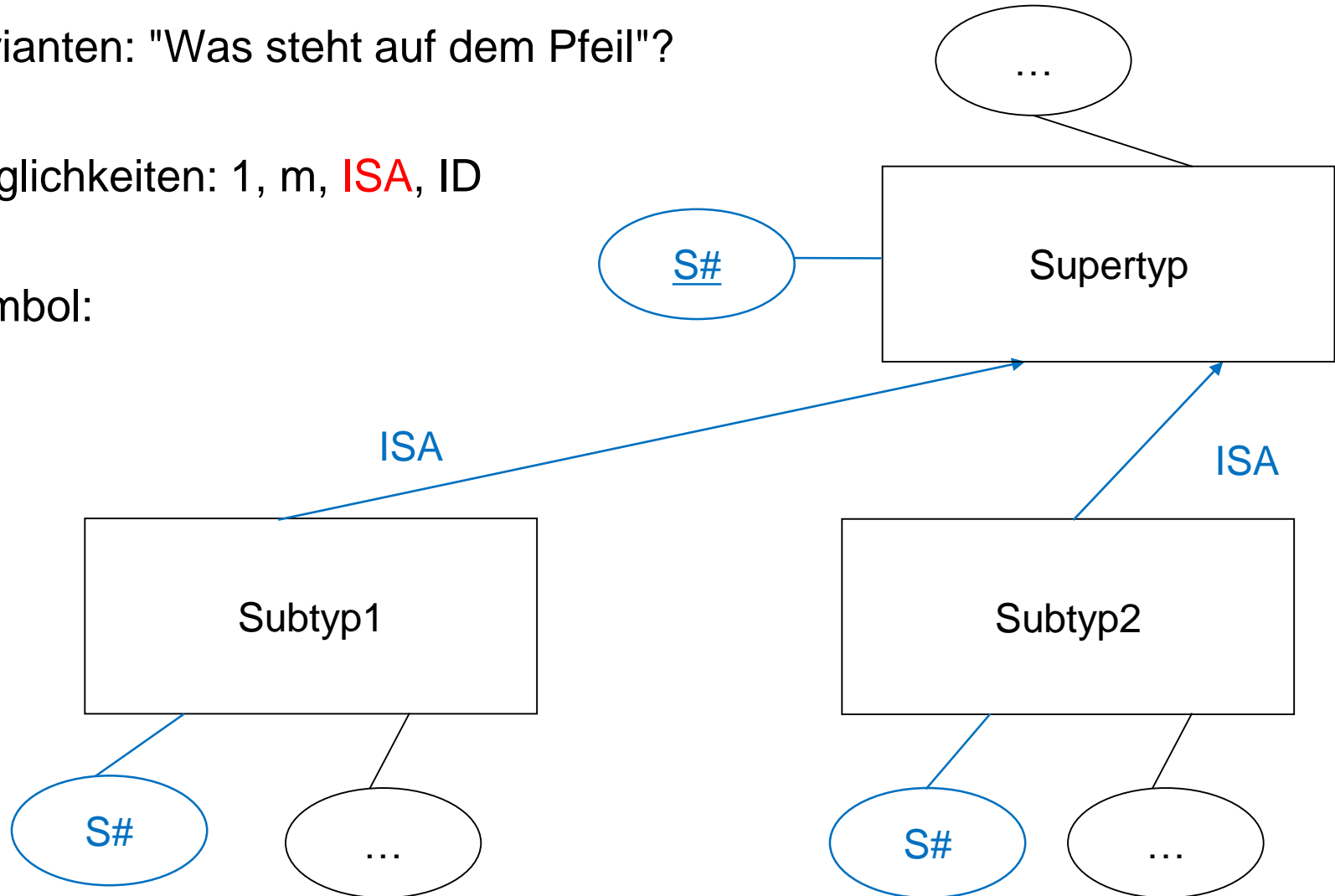


1 Schlüssel (von E **komb.** F)

- Der Beziehungstyp ist **existentiell abhängig** von den Entitätstypen, welche er referenziert

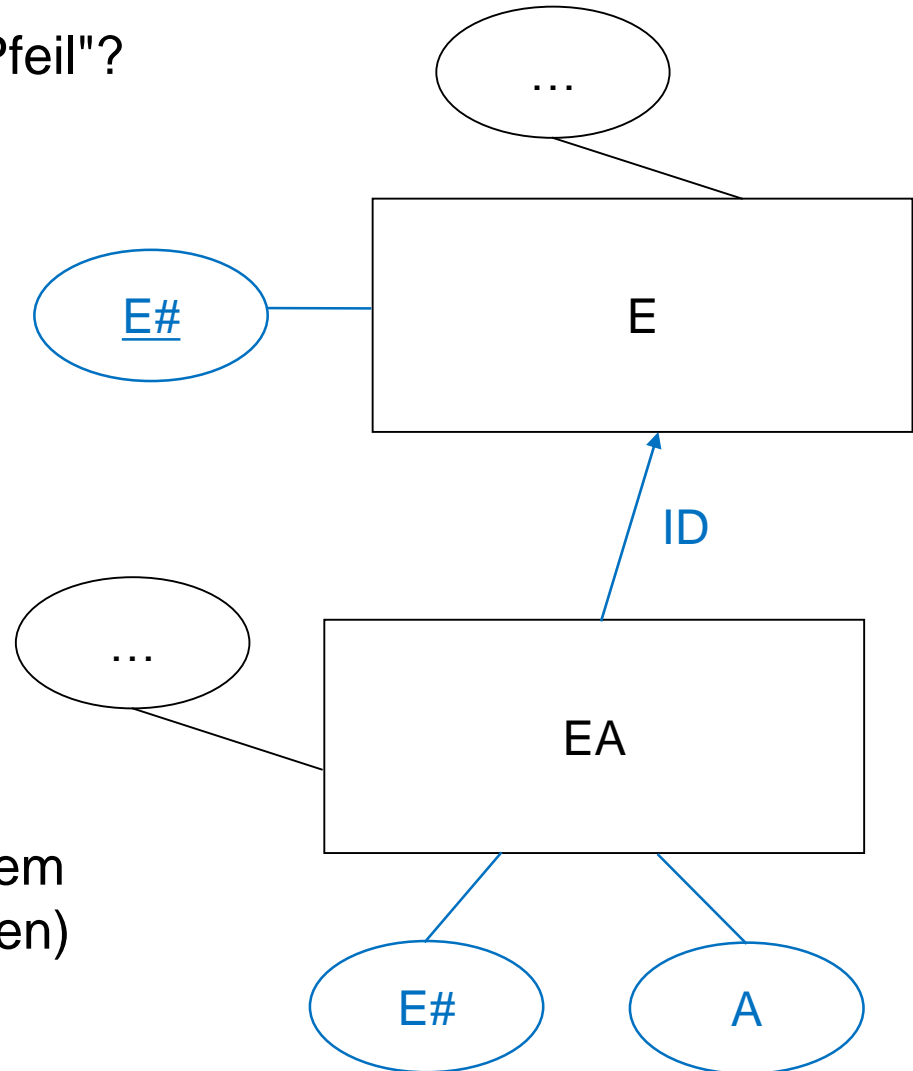
Bemerkungen zur Notation

- Varianten: "Was steht auf dem Pfeil"?
- Möglichkeiten: 1, m, **ISA**, ID
- Symbol:



Bemerkungen zur Notation

- Varianten: "Was steht auf dem Pfeil"?
- Möglichkeiten: 1, m, ISA, **ID**
- Symbol:
- Achtung: ID-Abhängigkeiten können auch zu Beziehungstypen hergestellt werden (der Beziehungstyp wird dann zu einem zusammengesetzten Entitätstypen)



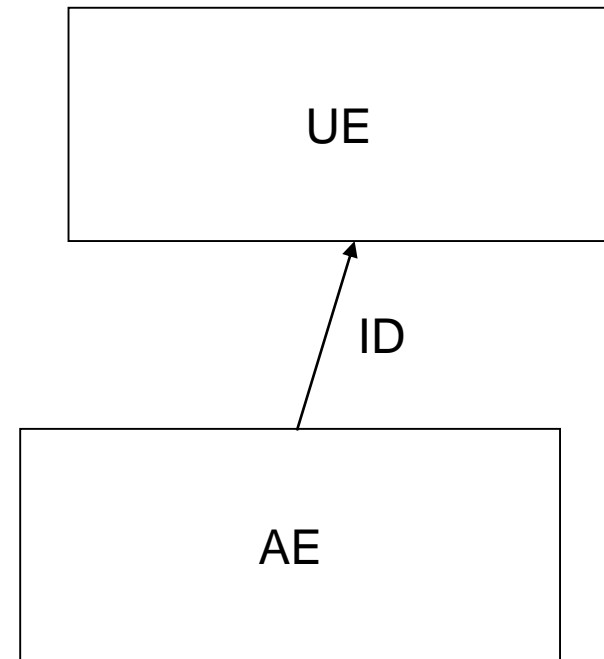
- Zusammengesetzter Entitätstyp



Bemerkungen zur Notation

- **Unabhängiger Entitätstyp**: nur «eingehende» (oder keine) Pfeile
- **Abhängiger Entitäts-/Beziehungstyp**: wenn «ausgehende» Pfeile
- Abhängig heisst: Jedes Objekt vom Typ AE ist **existentiell abhängig** von einem Objekt des Typ UE.

UE hat Primärschlüssel, AE hat Fremdschlüssel zum Primärschlüssel von UE.



Lernziele Lektion 8

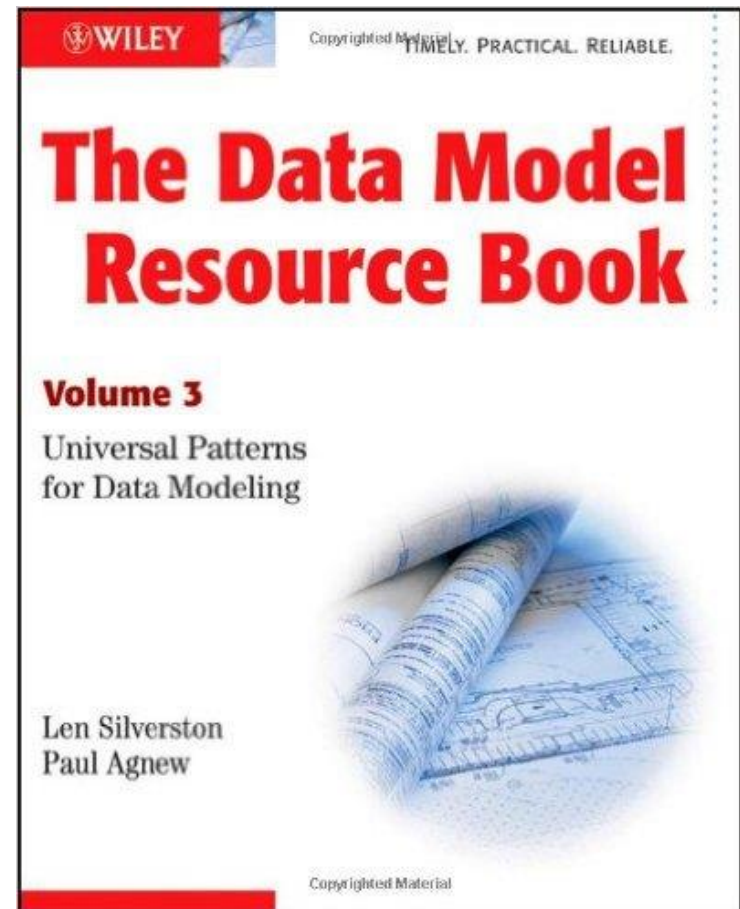
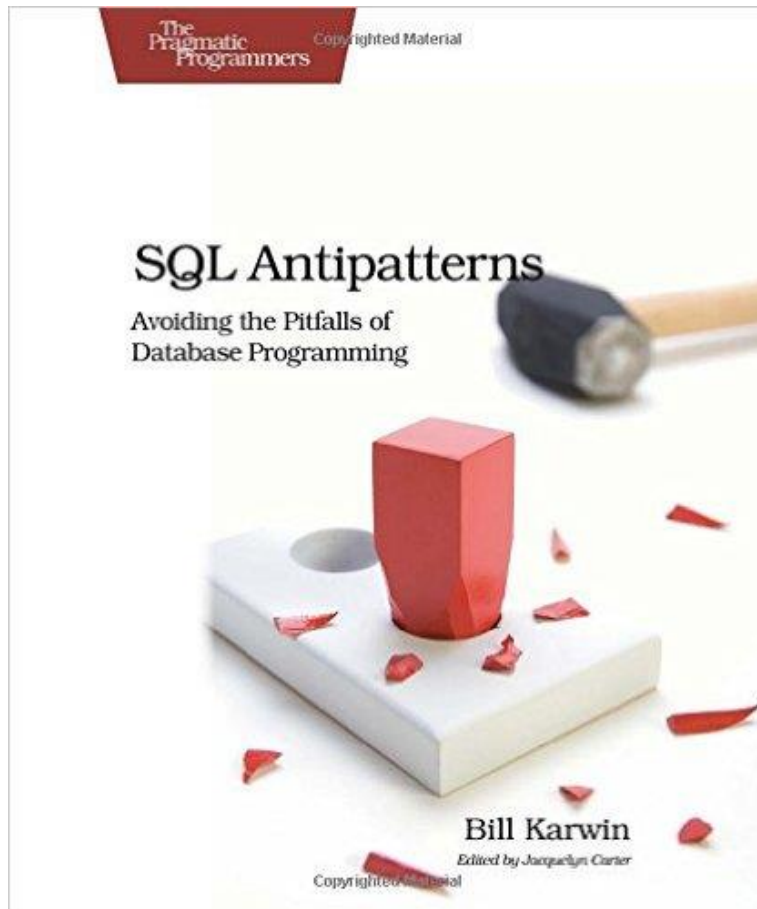
- Beispiel eines «Entwurfsmusters» kennen.
- Regeln kennen für ein «korrektes» ER-Diagramm.
- Probleme kennen, die sich bei «schlechten» Entwürfen ergeben.
- Im Grundsatz verstehen, wozu Normalisierung dient und wie sie funktioniert.

Entwurfsmuster

- Bewährte Idee in der Softwareentwicklung:
 - Nutzung von Komponenten
 - Verwenden von "Mustern"
 - "Proven parts" als "black box" nutzen
(aber Vorsicht: eine solche black box sollte nicht ganz black sein...)
- Analogie im Datenbankentwurf:
 - Für sich wiederholende Problemstellungen bewährte Entwurfsmuster anwenden
 - Ziele: Zeit sparen, weniger Fehler machen
 - Achtung: Wahl & Einsatz von Mustern muss gut überlegt sein

Entwurfsmuster

- Literaturempfehlungen:



Entwurfsmuster: EAV

- EAV: Entity-Attribute-Value
- Manchmal ist in der Designphase noch unklar, **welche und wie viele** Attribute ein Entitätstyp haben soll.
- Vielleicht soll der Benutzer später Attribute hinzufügen können (d.h. eine Tabelle sollte ohne ALTER TABLE «erweiterbar» sein).
- Oder jede Entität hat andere Attribute (mit entsprechenden Attributswerten).
- ACHTUNG: Im Modul (DAB1) wird dieses Muster (nach H.W. Buff) oft als «**Handorgel**» bezeichnet.

Entwurfsmuster: EAV

Diskutiert im Unterricht. Machen Sie Ihre eigenen Notizen.

Entwurfsmuster: EAV

- Statt die Attribute direkt an die Entität zu hängen (d.h. die Attributwerte wären in einer Spalte der zugehörigen Tabelle), haben wir nun in EATTRWERTE Beziehungen der Form:
- $\langle E\#, \text{AttrName}, \text{AttrWert} \rangle$
- Wird «Handorgel» genannt, weil die Attribute der Entität statt in einer Zeile einer Tabelle dargestellt, nun in mehreren Zeilen, d.h. vertikal, «aufgezogen» werden.
- Nachteile:
 - Wir brauchen **pro Datentyp** eine eigene Handorgel.
 - Konsistenzbedingungen für einzelne Attribute sind schwierig zu formulieren.

Korrektes ER-Diagramm

- Welche formal **syntaktischen Eigenschaften** benötigt ein ER-Diagramm, um gewisse erwünschte Eigenschaften bei Abbildung in Relationen zu garantieren?
- Erfahrung zeigt, dass alle in der Praxis vorkommenden Anforderungen an Datenstrukturen durch ein korrektes Diagramm abgedeckt werden können.
- Ein korrektes ER Diagramm ist ein ER Diagramm, welches aus dem leeren Diagramm hergestellt werden kann durch eine Folge von sog. «Metaoperationen» (oder Regeln).
- **ACHTUNG: Ein korrektes Diagramm ist nicht zwingend auch ein richtiges/gutes Diagramm!**

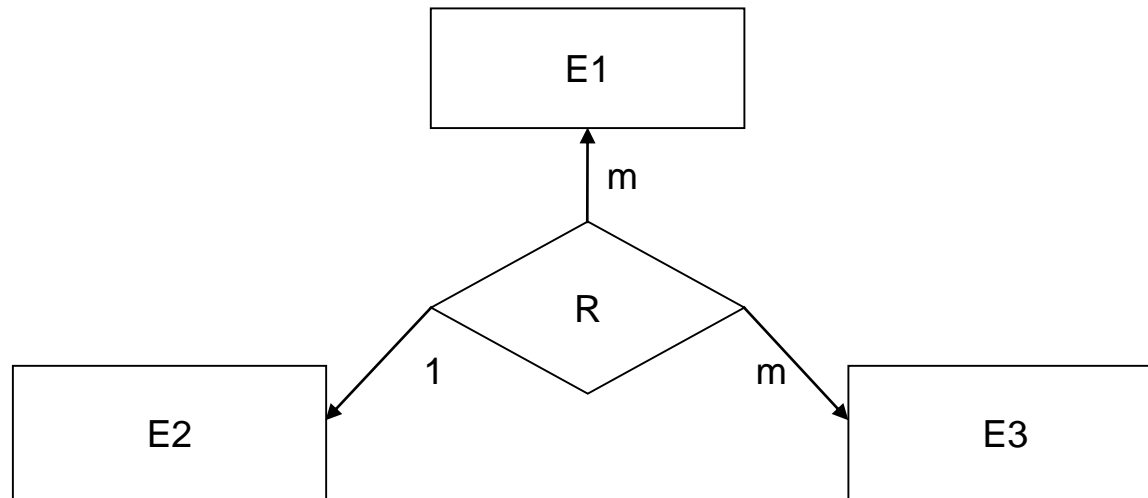
Korrekte ER-Diagramme: Regel 1

- **Definiere unabhängigen Entitätstyp**
- Voraussetzungen: Keine
- Ergebnis: Ein neues Rechteck (mit eindeutiger Bezeichnung)



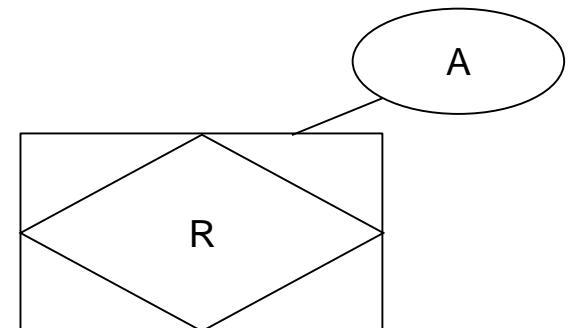
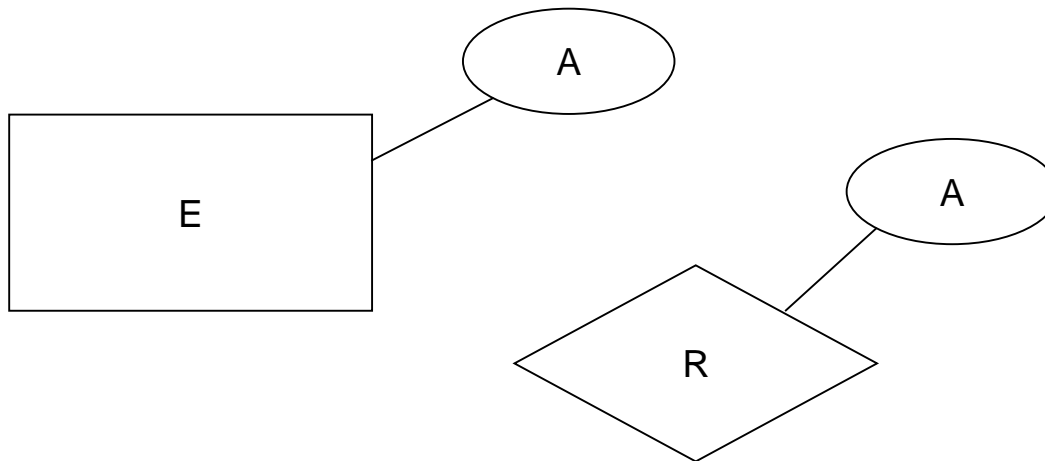
Korrekte ER-Diagramme: Regel 2

- **Definiere Beziehungstyp**
- Voraussetzungen: Mindestens zwei Entitätstypen (Rechtecke oder rechteckumschlossene Rhomben)
- Ergebnis: Ein neuer Rhombus R mit '1' oder 'm' markierten Pfeilen zu den Entitätstypen



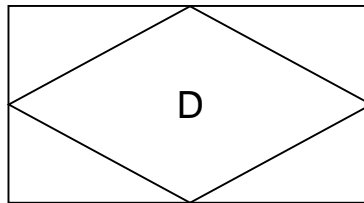
Korrekte ER-Diagramme: Regel 3

- **Definiere Attribut**
- Voraussetzung: Entitäts- oder Beziehungstyp (ein Rechteck, ein Rhombus oder ein rechteckumschlossener Rhombus)
- Ergebnis: Neues Oval strichverbunden mit E oder R und mit (innerhalb E bzw. R eindeutigem Namen)



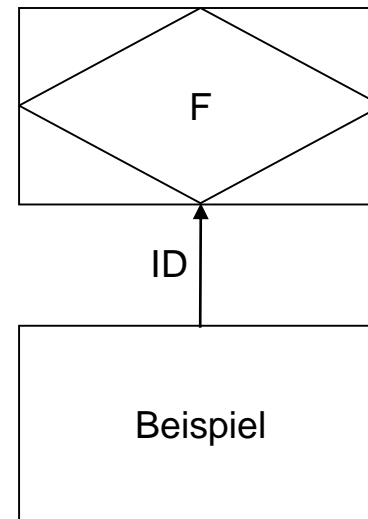
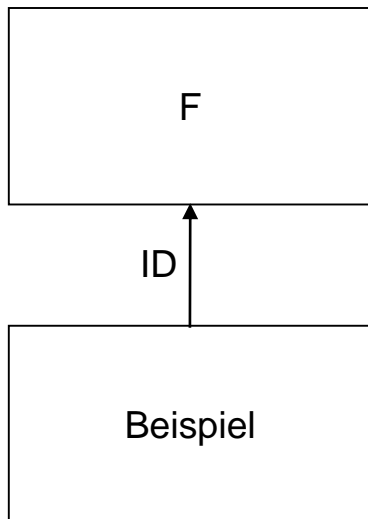
Korrekte ER-Diagramme: Regel 4

- Wandle Beziehungstyp in zusammengesetzten Entitätstyp um
- Voraussetzungen: Ein Rhombus D
- Ergebnis: Rhombus D durch ein Rechteck umschlossen



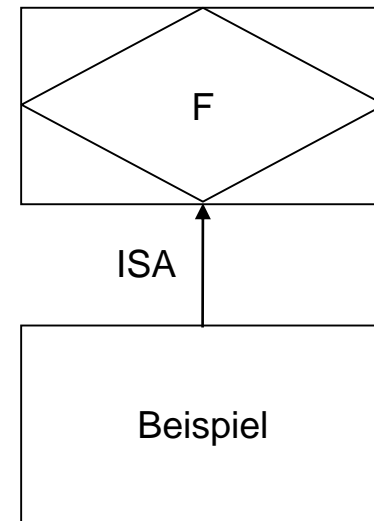
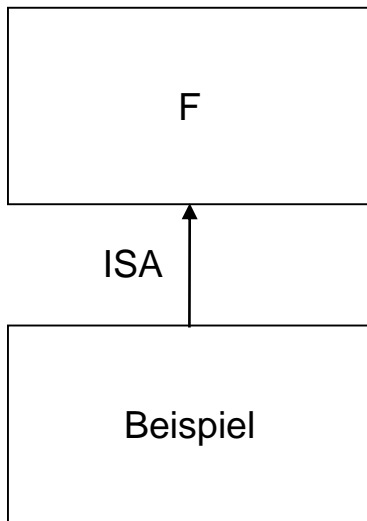
Korrekte ER-Diagramme: Regel 5

- **Definiere ID-abhängigen Entitätstyp**
- Voraussetzungen: Ein Rechteck oder rechteckumschlossener Rhombus F
- Ergebnis: Neues Rechteck mit 'ID' markiertem Pfeil zu F



Korrekte ER-Diagramme: Regel 6

- **Definiere ISA-abhängigen Entitätstyp**
- Voraussetzungen: Ein Rechteck oder rechteckumschlossener Rhombus F
- Ergebnis: Neues Rechteck mit 'ISA' markiertem Pfeil zu F



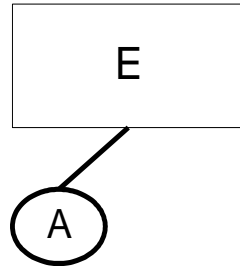
Korrekte ER-Diagramme: Beispiel

- Definiere unabhängigen Entitätstyp (Regel 1)



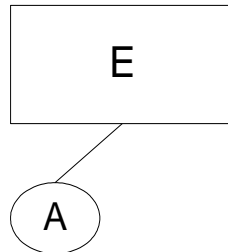
Korrekte ER-Diagramme: Beispiel

- Definiere Attribut (Regel 3)



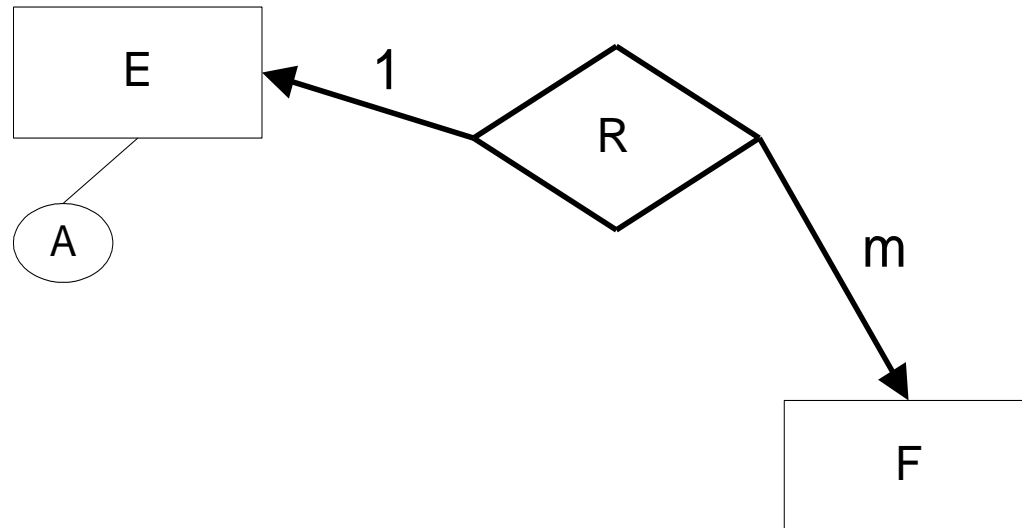
Korrekte ER-Diagramme: Beispiel

- Definiere unabhängigen Entitätstyp (Regel 1)



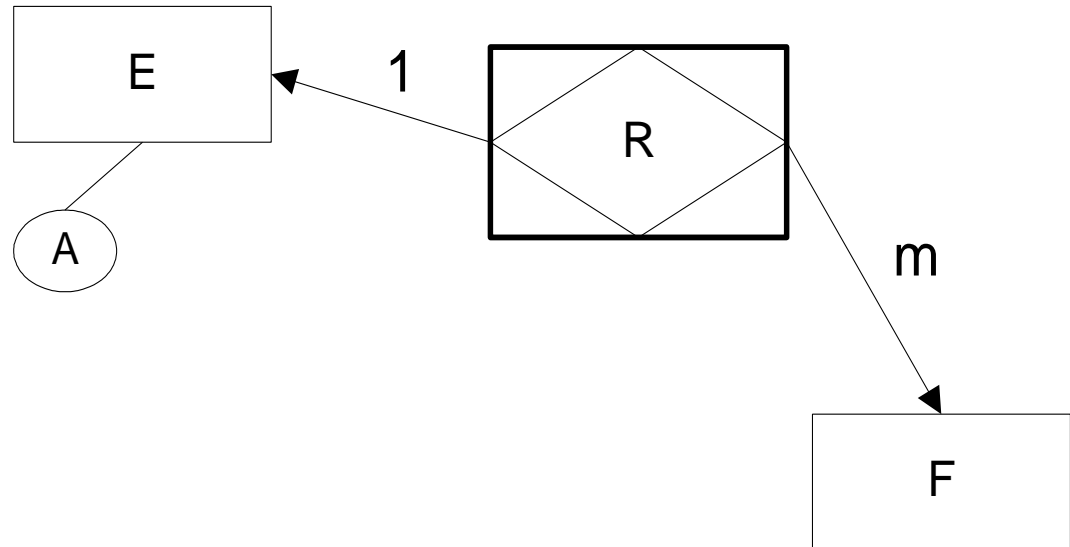
Korrekte ER-Diagramme: Beispiel

- Definiere Beziehungstyp (Regel 2)



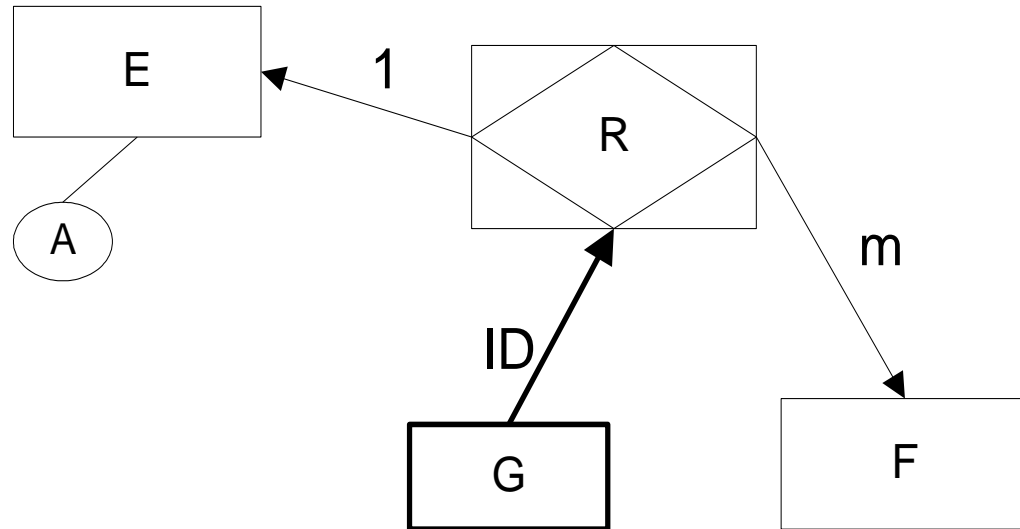
Korrekte ER-Diagramme: Beispiel

- Wandle Beziehungstyp in zusammengesetzten Entitätstypen um (Regel 4)



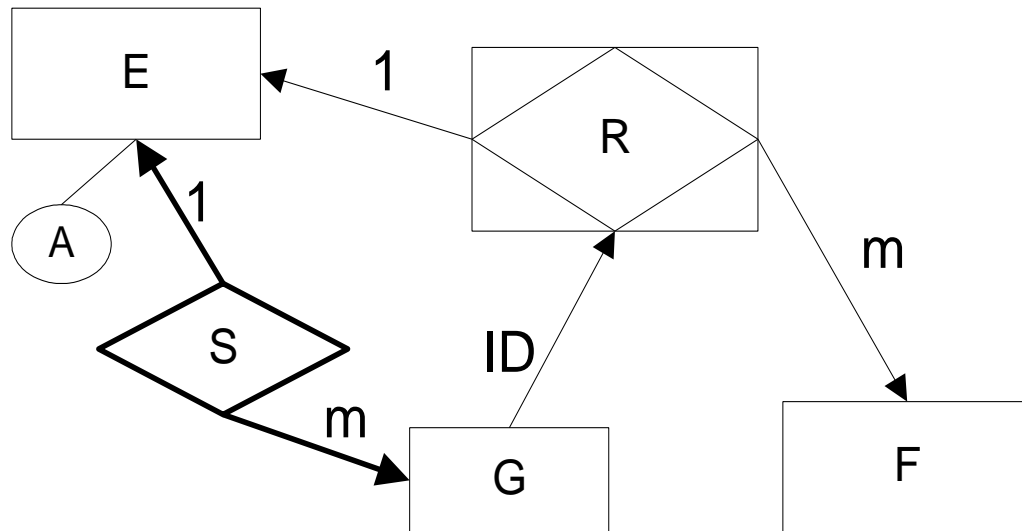
Korrekte ER-Diagramme: Beispiel

- Definiere ID-abhängigen Entitätstypen (Regel 5)



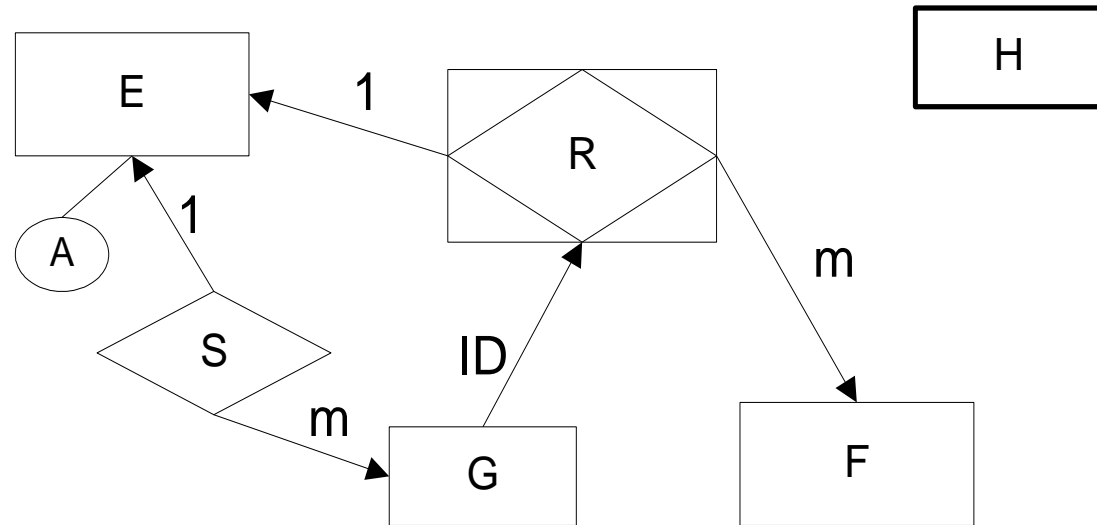
Korrekte ER-Diagramme: Beispiel

- Definiere Beziehungstyp (Regel 2)



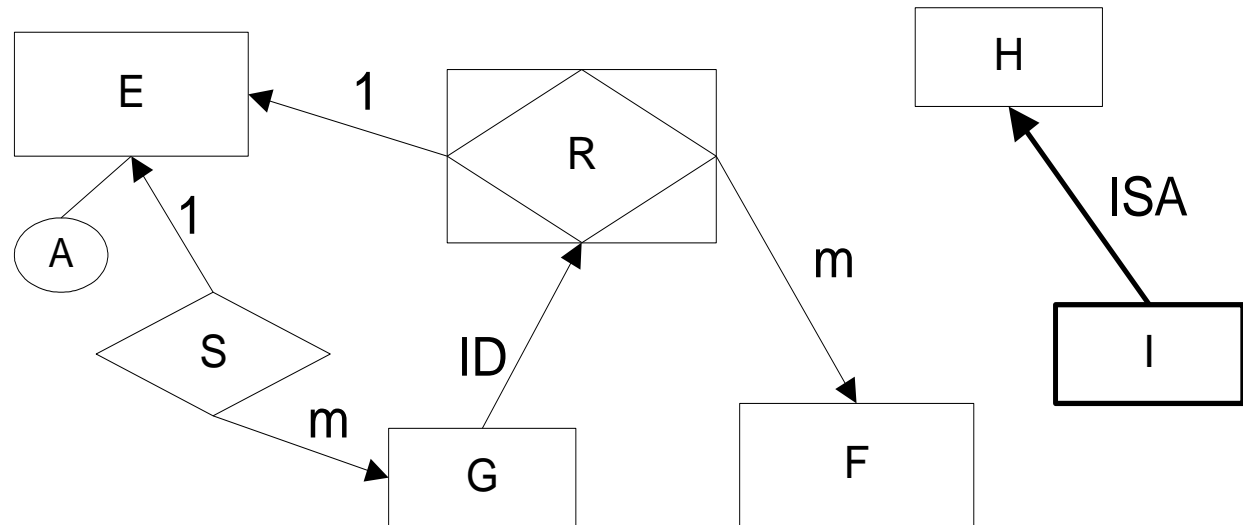
Korrekte ER-Diagramme: Beispiel

- Definiere unabhängigen Entitätstyp (Regel 1)



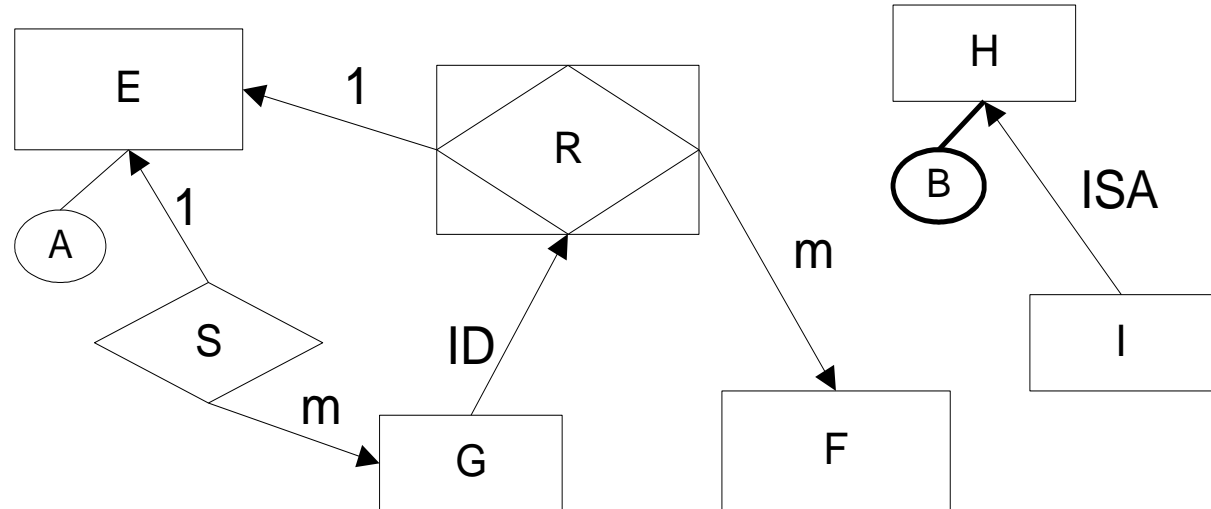
Korrekte ER-Diagramme: Beispiel

- Definiere ISA-abhängigen Entitätstyp (Regel 6)



Korrekte ER-Diagramme: Beispiel

- Definiere Attribut (Regel 3)



- Und so weiter...

Korrekte ER-Diagramme: Schlüssel

- Bevor ein Diagramm «weiterverarbeitet» werden kann, müssen die **Schlüssel** definiert sein. Ausgangslage: Korrektes Diagramm.
- Jeder unabhängige Entitätstyp erhält **einen oder mehrere** Schlüssel.
- Falls der Entitätstyp eingehende Pfeile hat, wählen wir einen **Primärschlüssel** (unterstreichen).
- Für Beziehungstypen: Wahl von **Fremdschlüssel** und **Schlüssel** (gemäss Kardinalitäten).
- Für Umwandlung in zusammengesetzte Entitätstypen: Primärschlüssel wählen (unterstreichen).
- Entitätstyp E ist ID- oder ISA-abhängig von F: **Primärschlüssel in F** wählen, **Fremdschlüssel und Schlüssel in E** wählen.
- → «**Angereichertes**» korrektes ER-Diagramm.

Umwandlung in relationales Modell

- «Rekursiv dem Aufbau des Diagramms entlang».
- Jedes **Kästchen** (Rechteck, Rhombus, rechteckumschlossener Rhombus) geht in eine **Relation** über, mit entsprechenden Attributen (**Domänen sind geeignet zu wählen**).
- Schlüssel, Fremdschlüssel, Primärschlüssel werden als solche übernommen.
- Dies nennt man die «kanonische» Abbildung eines (angereicherten) korrekten ER-Diagramms.

Vorteile des «korrekten» Diagrammes

- Einfach!
- Es gibt keine «Zyklen»
- Wenn Entitätstypen und Attribute gut gewählt («Unabhängiges wird unabhängig modelliert») direkte Abbildung (ohne «Nacharbeiten») in Relationen möglich.
- **ER-Modellstruktur passt zum logischen Entwurf!**

Einschub: Warum ER-Diagramm?

- Es stellt sich die Frage, ob man bei einfachen Problemstellungen den Umweg via **ER-Diagramm** überhaupt gehen muss. Man könnte doch **direkt Relationen** definieren. Ja, könnte man, aber ...
 - Beispiel: Wir wollen folgenden Sachverhalt festhalten:
 - Studierende haben eine Studierendenummer, einen Namen und eine Adresse.
 - Sie studieren an genau einem Departement, das eine Nummer hat und einen Namen.
 - Sie belegen Kurse (die haben eine Bezeichnung) legen dort eine Prüfung ab, die mit einer Note bewertet wird.
- Student(SNo, SName, Adresse, DNo, DName, Kurs, Note)

Einschub: Warum ER-Diagramm?

- Das führt – beispielsweise – zu folgender Relation:

<u>SNo</u>	SName	Address	DNo	DName	<u>Course</u>	Grade
87-604-I	Meier	Basel	IIIC	Informatik	Informatik	6
87-604-I	Meier	Basel	IIIC	Informatik	Analysis	5
87-604-I	Meier	Basel	IIIC	Informatik	Physik	4
91-872-I	Schmid	Bern	IIIC	Informatik	Informatik	5
91-872-I	Schmid	Bern	IIIC	Informatik	Analysis	3
91-109-I	Anderegg	Zürich	IIIC	Informatik	Informatik	4
94-555-P	Imboden	Luzern	IX	Mathematik	Algebra	3

- Was ist daran schlecht?

Einschub: Warum ER-Diagramm?

- Wir haben sogenannte Anomalien:
 - Update-Anomalie: „Meier“ zieht nach Zürich um
→ Änderungen in 3 Tupeln nötig
 - Delete-Anomalie: „Imboden“ verlässt die Schule
→ Fakt, dass Dept. IX = Mathematik, geht verloren
 - Insert-Anomalie: „Kunz“ hat noch keine Prüfung abgelegt
→ Fakt, dass er in Aarau wohnt kann nicht eingefügt werden

<u>SNo</u>	SName	Address	DNo	DName	<u>Course</u>	Grade
87-604-I	Meier	Basel	IIIC	Informatik	Informatik	6
87-604-I	Meier	Basel	IIIC	Informatik	Analysis	5
87-604-I	Meier	Basel	IIIC	Informatik	Physik	4
91-872-I	Schmid	Bern	IIIC	Informatik	Informatik	5
91-872-I	Schmid	Bern	IIIC	Informatik	Analysis	3
91-109-I	Anderegg	Zürich	IIIC	Informatik	Informatik	4
94-555-P	Imboden	Luzern	IX	Mathematik	Algebra	3

Einschub: Warum ER-Diagramm?

- Es gibt eine präzise mathematische Vorgehensweise um diese Probleme zu «flicken»: **Normalisierung**
- Aber: Was wäre passiert beim top-down-Entwurf?
 - Studierende haben eine Studierendenummer, einen Namen und eine Adresse.
 - Sie studieren an genau einem Departement, das eine Nummer hat und einen Namen.
 - Sie belegen Kurse (die haben eine Bezeichnung) legen dort eine Prüfung ab, die mit einer Note bewertet wird.
- Zeichnen Sie das resultierende ER-Diagramm

Einschub: Warum ER-Diagramm?

Diskutiert im Unterricht. Machen Sie Ihre eigenen Notizen.

Lösung: Keine Anomalien!

Diskutiert im Unterricht. Machen Sie Ihre eigenen Notizen.

Normalisierung

- Wir wollen Redundanzen und Anomalien in unserer Datenbank vermeiden.
- Wenn wir einfach «drauflos» Tabellen definieren, treten einige Probleme auf. Wir müssen so entstehende relationale Modelle noch **normalisieren**.
- Man unterscheidet verschiedene 'Normalformen':
 - 1NF -> 2NF -> 3NF -> **BCNF** -> 4 NF -> **5NF** (Aussagen über eine Rel.)
 - IDNF (Aussagen über zwei Rel.)
- Salopp: Normalisieren bedeutet Zerlegen in «schmalere» Relationen.
- ACHTUNG: Normalisierung (wenn korrekt durchgeführt) ist ein mathematisch anspruchsvolles Vorgehen!

Normalisierung

- Grundlegendes Verfahren bei relationalen Datenbanken:
 - Vermeidung von Redundanz
 - Vermeidung von Mutationsanomalien
 - Für das relationale Modell ist Normalisierung nicht erforderlich!
- Zwei Ansätze:
 - Formal:
 - Basis: Mathematisch saubere Theorie (u.a. funktionale Abhängigkeiten)
 - Vorteil: Sehr präzise, kann automatisiert werden
 - Nachteil: Nicht sehr intuitiv, oft etwas praxisfern
 - Informal:
 - Basis: Erfahrung
 - Vorteil: Praxisnah
 - Nachteil: Führt nicht immer zum ‚richtigen‘ Ergebnis

Problem 1: Redundanzen

- Redundanzen: dieselbe Information wird **mehrfach** wiederholt

Name	<u>M#</u>	GebDat	Abt#	AName	AMgr
Meier	17	1985-03-03	3	Verkauf	Berger
Müller	18	1982-07-13	4	Personal	Gerber
Huber	25	1961-08-01	1	Produktion	Baumann
Bühler	28	1982-02-09	3	Verkauf	Berger
Schneider	33	1971-06-22	4	Personal	Gerber

- Verschwendet bei grossen Tabellen **signifikant** Speicherplatz!

Problem 2: Insert-Anomalien

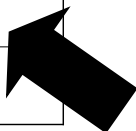
- Was kann passieren, wenn wir einen **neuen** Eintrag vornehmen?

Name	<u>M#</u>	GebDat	Abt#	AName	AMgr
Meier	17	1985-03-03	3	Verkauf	Berger
Müller	18	1982-07-13	4	Personal	Gerber
Huber	25	1961-08-01	1	Produktion	Baumann
Bühler	28	1982-02-09	3	Verkauf	Berger
Schneider	33	1971-06-22	4	Personal	Gerber

Problem 2: Insert-Anomalien

- Es treten **Insert-Anomalien** auf:

Name	<u>M#</u>	GebDat	Abt#	AName	AMgr
Meier	17	1985-03-03	3	Verkauf	Berger
Müller	18	1982-07-13	4	Personal	Gerber
Huber	25	1961-08-01	1	Produktion	Baumann
Bühler	28	1982-02-09	3	Verkauf	Berger
Schneider	33	1971-06-22	4	Personal	Gerber
Fischer	37	1977-01-16	-	-	-
Schmidt	41	1973-05-29	3	Verkauf	Burger

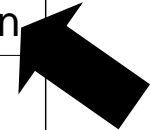


- Was passiert, wenn wir zu einem Mitarbeiter die Abteilung noch nicht wissen? Was passiert, wenn wir eine neue Abteilung mit (vorerst) keinen Mitarbeitern gründen?

Problem 3: Delete-Anomalien

- Was kann passieren, wenn wir einen Eintrag **löschen**?

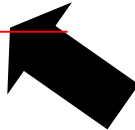
Name	<u>M#</u>	GebDat	Abt#	AName	AMgr
Meier	17	1985-03-03	3	Verkauf	Berger
Müller	18	1982-07-13	4	Personal	Gerber
Huber	25	1961-08-01	1	Produktion	Baumann
Bühler	28	1982-02-09	3	Verkauf	Berger
Schneider	33	1971-06-22	4	Personal	Gerber



Problem 3: Delete-Anomalien

- Es treten **Delete-Anomalien** auf:

Name	<u>M#</u>	GebDat	Abt#	AName	AMgr
Meier	17	1985-03-03	3	Verkauf	Berger
Müller	18	1982-07-13	4	Personal	Gerber
Huber	25	1961-08-01	1	Produktion	Baumann
Bühler	28	1982-02-09	3	Verkauf	Berger
Schneider	33	1971-06-22	4	Personal	Gerber

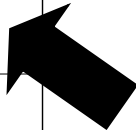


- Was passiert mit der Information zu Abteilung "Produktion"?

Problem 4: Update-Anomalien

- Was kann passieren, wenn wir einen Eintrag **verändern**?

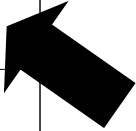
Name	<u>M#</u>	GebDat	Abt#	AName	AMgr
Meier	17	1985-03-03	3	Verkauf	Berger
Müller	18	1982-07-13	4	Personal	Gerber
Huber	25	1961-08-01	1	Produktion	Baumann
Bühler	28	1982-02-09	3	Verkauf	Berger
Schneider	33	1971-06-22	4	Personal	Gerber



Problem 4: Update-Anomalien

- Es treten **Update-Anomalien** auf:

Name	<u>M#</u>	GebDat	Abt#	AName	AMgr
Meier	17	1985-03-03	3	Verkauf	Berger
Müller	18	1982-07-13	4	Personal	Gerber
Huber	25	1961-08-01	1	Produktion	Baumann
Bühler	28	1982-02-09	3	Verkauf	Burger
Schneider	33	1971-06-22	4	Personal	Gerber



- Wir müssen ggf. andere Einträge ebenfalls **anpassen**, um die Konsistenz zu wahren

Problem 5: Join-Anomalien

Was passiert, wenn wir die folgende Tabelle in zwei Tabellen **zerlegen**?

Filiale	<u>Kredit#</u>	Kunde	Betrag
ZH	17	Meier	10000
BE	23	Müller	15000
ZH	15	Huber	10000
BS	11	Bühler	20000
LU	25	Schneider	10000
BS	18	Klein	15000
ZH	16	Bauer	20000

Problem 5: Join-Anomalien

- Wir erhalten (z.B.) folgende zwei Tabellen:

Filiale	<u>Kredit#</u>	Betrag
ZH	17	10000
BE	23	15000
ZH	15	10000
BS	11	20000
LU	25	10000
BS	18	15000
ZH	16	20000

Betrag	<u>Kunde</u>
10000	Meier
15000	Müller
10000	Huber
20000	Bühler
10000	Schneider
15000	Klein
20000	Bauer

- Wir haben Information über die Filialen von Information über den Kunden getrennt.
- Was, wenn wir doch Kunden Filialen zuordnen müssen?

Problem 5: Join-Anomalien

- Wir berechnen den Join:
- Wir erhalten **zusätzliche, unerwünschte** Tupel!
→ **Join-Anomalie!**
- Bem: man nennt dies einen «**lossy join**» d.h. einen verlustbehafteten Join. Warum?

Filiale	Kredit#	Betrag	Kunde
ZH	17	10000	Meier
ZH	17	10000	Huber
ZH	17	10000	Schneider
BE	23	15000	Müller
BE	23	15000	Klein
ZH	15	10000	Meier
ZH	15	10000	Huber
ZH	15	10000	Schneider
BS	11	20000	Bühler
BS	11	20000	Bauer
LU	25	10000	Meier
LU	25	10000	Huber
LU	25	10000	Schneider
BS	18	15000	Müller
BS	18	15000	Klein
ZH	16	20000	Bühler
ZH	16	20000	Bauer

Zusammenfassung der Probleme

- Wir wollen **alle** diese Probleme **vermeiden**:
 - Redundanzen
 - Insert-, Delete-, Update-Anomalien
 - Lossy Joins
- Was tun? → Korrekte Relationen definieren (Entwurfsprozess)!
- Achtung: Normalisierung ist ein «Reparaturvorgang». Wenn ein guter konzeptioneller Entwurf gemacht wurde muss dieser nicht nachträglich wieder «geflickt» werden!

Zur Erinnerung: Schlüssel

- Schlüssel:
 - Sei K eine Teilmenge des Formats von R . Dann ist K **genau dann ein Schlüssel** von R falls gilt:
 - 1. **Eindeutigkeit**: Es gibt in r zu R keine zwei Tupel mit denselben Werten von K
 - 2. **Minimalität**: Keine echte Teilmenge von K hat die Eindeutigkeitseigenschaft
 - Falls nur 1. gilt spricht man von **Superschlüssel** (d.h. jeder Schlüssel ist ein Superschlüssel aber nicht umgekehrt).
 - Ein **Subschlüssel** von R ist eine Teilmenge eines Schlüssels von R .
 - Eine Relation kann **mehrere Schlüssel** haben. Dann wählen wir einen davon als sog. Primärschlüssel.
 - Zur Erinnerung: Das ganze Format von R ist immer ein Superschlüssel.

Begriff: Funktionale Abhängigkeit

- **Funktionale Abhängigkeit** (FD, functional dependency):
 - Seien X und Y Teilmengen des Formats von R (also Attributmengen).
 - Dann gilt die funktionale Abhängigkeit $X \rightarrow Y$ genau dann wenn gilt: stimmen zwei Tupel in X überein, dann stimmen sie auch in Y überein.
 - Ein Attributmenge A ist **voll funktional abhängig** von einem aus S1 und S2 zusammengesetzten Schlüssel, wenn A funktional abhängig vom Gesamtschlüssel, nicht aber von seinen Teilschlüsseln ist.
Schreibweise: $(S1, S2) \rightarrow A$, $S1 \nrightarrow A$ und $S2 \nrightarrow A$
 - Funktionale Abhängigkeiten sind **Bedingungen** die in der Datenbank gelten (und vom RDBMS durchgesetzt werden) sollen!

Begriff: Funktionale Abhängigkeit

- Beispiel: Mitarbeiter(M#, Abt, Div)

Mögliche funktionale Abhängigkeiten sind:

- $\{Abt, Div\} \rightarrow \{Abt\}$
 - $\{M\# \} \rightarrow \{Abt, Div\}$
 - $\{Div\} \rightarrow \{Div\}$
 - ...
-
- Kurzschreibweise: $AB \rightarrow C$ entspricht $\{A,B\} \rightarrow \{C\}$

Begriff: Funktionale Abhängigkeit

- Funktionale Abhängigkeit, Bemerkungen:
 - $K \rightarrow Y$ gilt für **alle Schlüssel** K und alle Teilmengen Y des Heading.
 - Falls $X \rightarrow Y$ gilt, dann gilt auch $X^+ \rightarrow Y^-$ für alle "Übermengen" X^+ von X und alle Teilmengen Y^- von Y
 - Wenn SK ein **Superschlüssel** in R ist dann gilt für alle Teilmengen Y des Formats von R : $SK \rightarrow Y$
 - Die FD $X \rightarrow Y$ heisst **trivial** genau dann, wenn sie durch alle möglichen Relationen deren Format X und Y umfassen, erfüllt wird und wenn gilt $Y \subseteq X$

Normalisierung

- Daten liegen in der Praxis in der Regel nicht als Relationen vor:
 - Formulare
 - Spreadsheets
 - Geschäftsdokumente aller Art (Rechnungen, Bestellungen, ...)
 - ...
- Beispiele:

Employee Number	Employee Name	Dept. Number	Dept. Name	Dept. Location	Qualification 1	
					Description	Year
01267	Clark	05	Auditing	HO	Bachelor of Arts	1970
70964	Smith	12	Legal	MS	Bachelor of Arts	1969
22617	Walsh	05	Auditing	HO	Bachelor of Arts	1972
50607	Black	05	Auditing	HO		

Qualification 2		Qualification 3		Qualification 4	
Description	Year	Description	Year	Description	Year
Master of Arts	1973	Doctor of Philosophy	1976		
Master of Arts	1977				

Figure 2.2 Employee qualifications table.

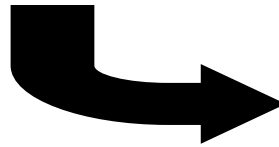
Hospital Number: H17	Hospital Name: St VincentÖs	Operation Number: 48			
Hospital Category: P	Contact at Hospital: Fred Fleming				
Operation Name: Heart Transplant	Operation Code: 7A	Procedure Group: Transplant			
Surgeon Number: S15	Surgeon Specialty: Cardiology	Total Drug Cost: \$75.50			
Drug Code	Full Name of Drug	Manufacturer	Method of Admin.	Cost of Dose (\$)	Number of Doses
MAX 150mg	Maxicillin	ABC Pharmaceuticals	ORAL	\$3.50	15
MIN 500mg	Minicillin	Silver Bullet Drug Co.	IV	\$1.00	20
MIN 250mg	Minicillin	Silver Bullet Drug Co.	ORAL	\$0.30	10

Normalisierung

- Notwendige Informationen (Anwendungswissen) für Überführung in Relationen:
 - Repetitionsgruppen („was kommt wie oft vor?“)
 - Schlüssel
 - (funktionale) Abhängigkeiten
 - Codierungen
 - ...

1NF (Erste Normalform)

Person		Hobbies
Name	Vorname	
Meier	Hans	Fussball, Lesen, Jogging
Müller	Peter	Briefmarken, Fussball
Gerber	Susanne	Volleyball, Ikebana



Name	Vorname	Hobby
Meier	Hans	Fussball
Meier	Hans	Lesen
Meier	Hans	Jogging
Müller	Peter	Briefmarken
Müller	Peter	Fussball
Gerber	Susanne	Volleyball
Gerber	Susanne	Ikebana

- Jedes Attribut muss **atomare** Wertebereiche haben.
- **Zusammengesetzte** oder **mengenwertige Attribute** sind nicht erlaubt.
- Achtung: Nach eingeführter Definition ist **JEDE** Relation in 1NF (sonst ist es keine Relation).

2NF (Zweite Normalform)

Verein	Stadt	Spieler	Salär
Juventus	Basel	Meier	70000
Juventus	Basel	Müller	72000
Juventus	Basel	Gerber	66000
Concordia	Zürich	Müller	67000
Concordia	Zürich	Gerber	87000
Concordia	Zürich	Baumann	55000
Concordia	Zürich	Berger	61000

- Jedes **Nichtschlüsselattribut** muss von jedem Schlüsselkandidaten voll funktional abhängig sein, d.h. alle Attribute, die nicht Teil des Schlüssels sind, sind vom ganzen Schlüssel abhängig, nicht nur von Teilen.

2NF (Zweite Normalform)

- Es sei {Verein, Spieler} Schlüssel. Dann ist aber {Stadt} nur von {Verein} abhängig.

Verein	Stadt
Juventus	Basel
Concordia	Zürich

Verein	Spieler	Salär
Juventus	Meier	70000
Juventus	Müller	72000
Juventus	Gerber	66000
Concordia	Müller	67000
Concordia	Gerber	87000
Concordia	Baumann	55000
Concordia	Berger	61000

- Die linke Tabelle enthält alle Attribute, welche nur von {Verein} abhängig sind.
- Die rechte Tabelle enthält alle Attribute, welche nur von {Verein, Spieler} abhängig sind.

3NF (Dritte Normalform)

Spieler_ID	Spieler	Verein	Gründungsjahr
3207	Meier	Juventus	1992
3209	Müller	Juventus	1992
3216	Gerber	Juventus	1992
7002	Müller	Concordia	1994
7003	Gerber	Concordia	1994
7007	Baumann	Concordia	1994
7011	Berger	Concordia	1994

- Es sind keine **transitiven** Abhängigkeiten erlaubt, d.h. Abhängigkeiten sind verboten, in denen ein Attribut über ein anderes Attribut vom Schlüssel abhängig ist.

3NF (Dritte Normalform)

- {Spieler} und {Verein} lassen sich aus {Spieler_ID} bestimmen
- {Gründungsjahr} hängt dagegen eigentlich von {Verein} ab, und damit nur indirekt von {Spieler_ID}

Spieler_ID	Spieler	Verein
3207	Meier	Juventus
3209	Müller	Juventus
3216	Gerber	Juventus
7002	Müller	Concordia
7003	Gerber	Concordia
7007	Baumann	Concordia
7011	Berger	Concordia

Verein	Gründungsjahr
Juventus	1992
Concordia	1994

BCNF (Boyce-Codd Normalform)

- «Alle Attribute hängen von einem Schlüssel, einem ganzen Schlüssel und nichts als einem Schlüssel ab (a key, a whole key, and nothing but a key)»
→ oft gehört aber **unpräzise!**
(Relationen können mehrere Schlüssel haben)
- Wird zum Beispiel verletzt, wenn ein Teil eines Schlüssels von anderen Attributen abhängt.

Student	Fach	Dozent
Meier	Mathematik	Baumann
Müller	Mathematik	Gerber
Meier	Informatik	Berger
Meier	Physik	Huber
Pfister	Mathematik	Baumann
Pfister	Informatik	Berger

BCNF (Boyce-Codd Normalform)

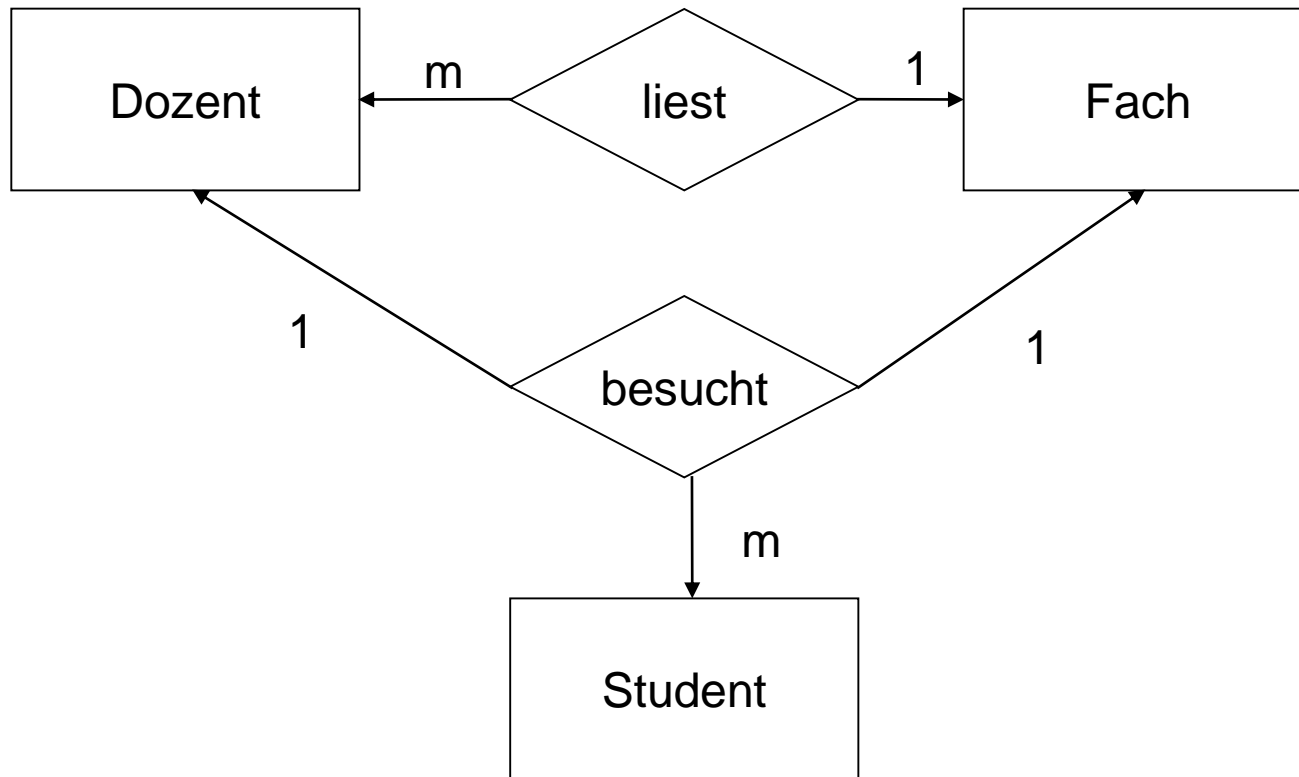
- Z.B. Schlüssel sei {Student, Fach}
- Gilt nun Dozent \rightarrow Fach, d.h. jeder Dozent liest nur 1 Fach, so ist die Tabelle nicht in BCNF
- Die Zerlegung ist nicht offensichtlich, da jede Zerlegung die Abhängigkeit {Student, Fach} \rightarrow {Dozent} zerstört.

Student	Dozent
Meier	Baumann
Müller	Gerber
Meier	Berger
Meier	Huber
Pfister	Baumann
Pfister	Berger
Meier	Gerber

Dozent	Fach
Baumann	Mathematik
Gerber	Mathematik
Berger	Informatik
Huber	Physik

"Korrekte" Lösung

- Wie sieht ein entsprechendes korrektes ER-Diagramm aus?
→ Unabhängiges wird unabhängig modelliert
- Die zwei Abhängigkeiten führen zu zwei unterschiedlichen Beziehungen



Normalformen – Zusammenfassung

- 2 NF – Für jede nichttriviale funktionale Abhängigkeit $X \rightarrow Y$ in R gilt:
 1. X ist ein Superschlüssel, ODER
 2. Y ist ein Subschlüssel, ODER
 3. X ist kein Subschlüssel
- 3 NF – Für jede nichttriviale funktionale Abhängigkeit $X \rightarrow Y$ in R gilt:
 1. X ist ein Superschlüssel, ODER
 2. Y ist ein Subschlüssel
- BCNF – Für jede nichttriviale funktionale Abhängigkeit $X \rightarrow Y$ in R gilt:
 1. X ist ein Superschlüssel

Und zum Schluss noch dies:

- Erstellen Sie ein ERM für folgenden Sachverhalt. Verwaltet werden:
 - Klassen (z.B. IT13t): Diese haben eine eindeutige Klassen#, eine Bezeichnung (Bez) und einen Jahrgang.
 - Vorlesungen (z.B. DAB1): Diese haben eine eindeutige Vorlesungs#, eine Kurzbezeichnung (KurzBez) und eine Bezeichnung (Bez)
 - Dozenten (z.B. Müller Hans): Diese haben ein eindeutiges Kürzel, einen Namen und einen Vornamen
 - Arbeitstage (z.B. 11.12.2015): Diese haben ein Datum
 - Studiengänge (z.B. Bachelorstudiengang WI): Studiengänge haben eine eindeutige Studiengangs# und eine Bezeichnung (Bez)

Und zum Schluss noch dies:

- Zudem müssen folgende Beziehungen gelten:
 - Veranstaltungen. Legen für die Durchführung von Vorlesungen fest:
 - Wann sie durchgeführt wird (Arbeitstage),
 - Für welche Klasse sie durchgeführt wird (Klassen),
 - Für welche Vorlesung sie durchgeführt wird (Vorlesungen).
 - Durch welchen Dozenten sie durchgeführt wird (Dozenten).
 - Veranstaltungen verknüpfen Klassen mit Vorlesungen, Dozenten und Arbeitstagen. Es gilt:
 - Eine Klasse und eine Vorlesung und ein Arbeitstag haben höchstens einen Dozenten.
 - Eine Klasse und ein Dozent und ein Arbeitstag können mehrere Vorlesungen haben.
 - Eine Vorlesung und ein Dozent und ein Arbeitstag können mehrere Klassen haben.
 - Eine Vorlesung und ein Dozent und eine Klasse können mehrere Arbeitstage haben.
 - Vorlesungszugehörigkeit: Zu welchen Studiengängen gehören welche Vorlesungen (eine Vorlesung kann in mehreren Studiengängen gehalten werden und ein Studiengang hat mehrere Vorlesungen).

Lösung:

Diskutiert im Unterricht. Machen Sie Ihre eigenen Notizen.

Und weiter...

Das nächste Mal: SQL

