

Inhaltsverzeichnis

1	Vorwort	4
1.1	Copyright	4
1.2	Persönliches	4
2	Allgemein	5
2.1	Zweck von Modellen	5
2.2	Intension	5
2.3	Extension	5
2.4	Daten - Arten	5
2.4.1	Nutzdaten	5
2.4.2	Metadaten	5
3	Schema Architektur (ANSI/SPARC)	6
3.1	Konzeptionelles Schema	6
3.2	Externe Schemata	6
3.3	Interne Schemata	6
3.4	Vorteile	7
4	ER bzw. EER	8
4.1	Attribute	8
4.1.1	Allgemein	8
4.1.2	Schlüsselattribute	8
4.2	Grundlagen	8
4.3	Klassenhierarchien	10
4.4	Spezialisierung	11
4.5	Generalisierung	12
5	Datenbankbegriffe	13
5.1	Datenbank	13
5.1.1	Definition	13
5.1.2	Vorteile/Nachteile	13
5.2	Datenbank Management System (DBMS)	13
5.3	Datenbankssystem (DBS)	13
5.4	Datenbankschema	14
5.5	Daten (bank) modell	14
5.6	Tupel	14
5.7	Relation	14
5.8	Attribut	14
5.9	Relationenschema [Intension]	14
5.10	Entity-Begriffe	14
5.10.1	Entity	14
5.10.2	Entity-Set [Extension]	14

5.10.3 Entity-Typ [Intension]	14
5.11 Attribute	14
5.11.1 Schlüsselkandidat	14
5.11.2 Primärschlüssel	15
5.11.3 Fremdschlüssel	15
5.12 Referenzielle Integrität	15
6 Mapping	16
7 Anomalien	17
8 Normalformen	18
8.1 Ziel	18
8.2 1.NF	18
8.3 2.NF	18
8.4 3.NF	19
8.5 BCNF	20
8.6 4.NF	21
9 ORACLE SQL	22
9.1 Create	22
9.2 Queries	22
9.3 Weiterführendes	22
9.3.1 Constraints	22
9.3.2 Trigger	23
9.3.3 Top-N-Queries	23
9.3.4 Assertions	24
9.4 Join-Arten	24
10 XML	25
10.1 Prolog	25
10.2 Element-/Attributdefinition	25
10.3 Beispiel	26
10.4 XPath	26
10.4.1 Allgemein	27
10.4.2 Prädikate	28
10.5 XML vs DTD	29
11 Data Warehouse	31
11.1 Definition	31
11.2 OLAP - OLTP	31
11.3 Kennzahlen	31
11.4 Relationenmodell	31
11.5 Multidimensionales Datenmodell	32
11.5.1 Daten	32

11.6 Operationen	32
11.6.1 Slicing und Dicing	32
11.6.2 Drill Down	33
11.6.3 Drill-Up/Roll-Up	33
11.6.4 Drill-Across	33
11.6.5 Drill-Through	33
11.6.6 Pivotierung	33
11.6.7 Aggregationen	34
11.7 Darstellungsmöglichkeiten	34
11.7.1 Multidimensional	34
11.7.2 Relational	34
11.7.3 Hybrid	34
11.8 ROLAP - Relationale Abbildung	35
11.8.1 Star Schema & Snowflake Schema	35
12 UML	36
12.1 Anwendungsfalldiagramm	36
12.2 Aktivitätsdiagramm	37
12.3 Sequenzdiagramm	41
12.3.1 Grundlagen	41
12.3.2 Operatoren	43
12.4 Zustandsdiagramm	47
12.5 Klassendiagramm	49
12.6 Modelle	52
13 Ontologie	53

1 Vorwort

1.1 Copyright

Dieses Dokument ist als Zusammenfassung der Vorlesungen Konzeptionelle Modellierung im WS 13/14 zu sehen. Das Dokument erhebt keinen Anspruch auf Vollständigkeit. Sofern die Texte/Abbildungen nicht von Prof. Dr. Richard Lenz stammen, wurden die Quellen in runden Klammern dahinter angegeben. Das heißt automatisch, dass bei jedem Text und jeder Abbildung, Prof. Dr. Richard Lenz als Urheber anzusehen ist. Ich habe in diesem Dokument lediglich die Lehrinhalte auf Basis der Vorlesungen gegliedert und gesammelt. Das Copyright an den Texte/Abbildungen besitzen demnach die Autoren selbst. Sollte der Lehrstuhl etwas gegen die Veröffentlichung haben, werde ich das Dokument sofort wieder aus dem Internet nehmen.

1.2 Persönliches

Da die Folien teilweise sehr redundant sind und nicht optimal zum gezielten Lernen geeignet sind und ich eh LaTeX etwas vertiefen wollte, habe ich mir meine eigene kleine Zusammenfassung zusammen geschrieben. Verzeiht mir daher bitte auch unsaubere Formatierungen o.ä. dieses Dokument ist mein erstes Größeres. Die Inhalte sind demnach auch nur soweit gefüllt wie ich es für mich für nötig halte und es kann durchaus passieren, dass man als Leser mit den Zusammenfassungen recht wenig anfangen kann. Ich hoffe es hilft euch trotzdem und ich wollte es einfach einmal teilen. Kann ja nicht Schaden. Vielleicht hilft es dem ein oder anderen ja auch. :)

2 Allgemein

2.1 Zweck von Modellen

- Spezifizieren
- Konstruieren
- Visualisieren
- Dokumentieren

2.2 Intension

- Beschreibung (Schema) einer Menge von Entities mit den gleichen Attributen.

2.3 Extension

- Eine Extension (engl. Erweiterung) ist in der Informatik, eine Menge von Beispielen. (Wikipeda)
- Die Menge der Entities, die zu einem Entity-Typ gehören ist die Extension dieses Typs.

2.4 Daten - Arten

2.4.1 Nutzdaten

- Eigentliche Datenbank
- Nutzdaten sind die Daten, die Benutzer in der Datenbank anlegen und aus denen Informationen gewonnen werden. (Geisler 2008)

2.4.2 Metadaten

- Beschreibungsdaten: Struktur der Datenbank Information über Speicherungsstrukturen
- Metadaten (Daten über Daten) helfen, die Nutzdaten der Datenbank zu strukturieren. (Geisler 2008)

3 Schema Architektur (ANSI/SPARC)

3.1 Konzeptionelles Schema

- Der Kern dieser Architektur ist die konzeptionelle Ebene. Hier wird der Teil der Realität, den das Datenbanksystem (DBS) nachbilden soll, in seiner logischen Gesamtheit beschrieben. In dieser Ebene werden alle Daten und ihre Beziehungen zueinander modelliert. Darüber hinaus ist sie unabhängig von den hardwaremäßigen Gegebenheiten und den Anforderungen einzelner Benutzer. (info-wsf.de)
- Beispiel: Auf der konzeptionellen Ebene ist für ein Gehaltsabrechnungsprogramm z.B. festzulegen, dass es die Datensätze "Mitarbeiter" gibt, die die Personalnummer, den Namen, das Gehalt usw. enthalten. Ferner ist festzuhalten, in welcher Beziehung einzelne Mitarbeiter zueinander stehen (z.B. Abteilungsleiter) oder in welchem Bereich der Firma ein Mitarbeiter beschäftigt ist (z.B. Materiallager). (info-wsf.de)

3.2 Externe Schemata

- Sichten für verschiedene Anwendungen
- Die externe Ebene legt fest, welche Daten bestimmte Benutzer bzw. Programme sehen und bearbeiten können. Dabei kann es zu jedem Anwendungsprogramm ein eigenes externes Schema geben. (info-wsf.de)
- Beispiel: Auf der externen Ebene ist für das Gehaltsabrechnungsbeispiel z.B. zu definieren, dass für die Statistiker von den Personaldaten nur das Gehalt (die Statistik soll anonym bleiben) und die Abteilung benötigt werden. (info-wsf.de)

3.3 Interne Schemata

- Datenverwaltung auf Platte
- In der internen Ebene wird die Organisation der Daten und ihrer Zugriffspfade auf den physischen Speicher festgelegt. In dieser Ebene werden Fragen geklärt, wie: Welche Daten werden zu welchen Einheiten (Datensätzen) zusammengefasst und wie schnell wird darauf zugegriffen? (info-wsf.de)
- Beispiel: Auf der internen Ebene ist für das Gehaltsabrechnungsbeispiel zu definieren, in welcher Reihenfolge die Felder "Personalnummer", "Name", "Gehalt" usw. des Datensatzes "Mitarbeiter" gespeichert werden, welche Länge sie haben und wie die Daten codiert werden. Des Weiteren sind hier Angaben über die Dateiorganisation und die Zugriffsmechanismen zu den Daten (z.B. Hash-Verfahren, Binärbaum) notwendig. (info-wsf.de)

3.4 Vorteile

Die Vorteile des Drei-Ebenen-Modells liegen in der

- physischen Datenunabhängigkeit, da die interne von der konzeptionellen und externen Ebene getrennt ist. Physische Änderungen, z.B. des Speichermediums oder des Datenbankprodukts, wirken sich nicht auf die konzeptionelle oder externe Ebene aus.
- logischen Datenunabhängigkeit, da die konzeptionelle und die externe Ebene getrennt sind. Dies bedeutet, dass Änderungen an der Datenbankstruktur (konzeptionelle Ebene) keine Auswirkungen auf die externe Ebene, also die Masken-Layouts, Listen und Schnittstellen haben.

(Wikipedia)

4 ER bzw. EER

4.1 Attribute

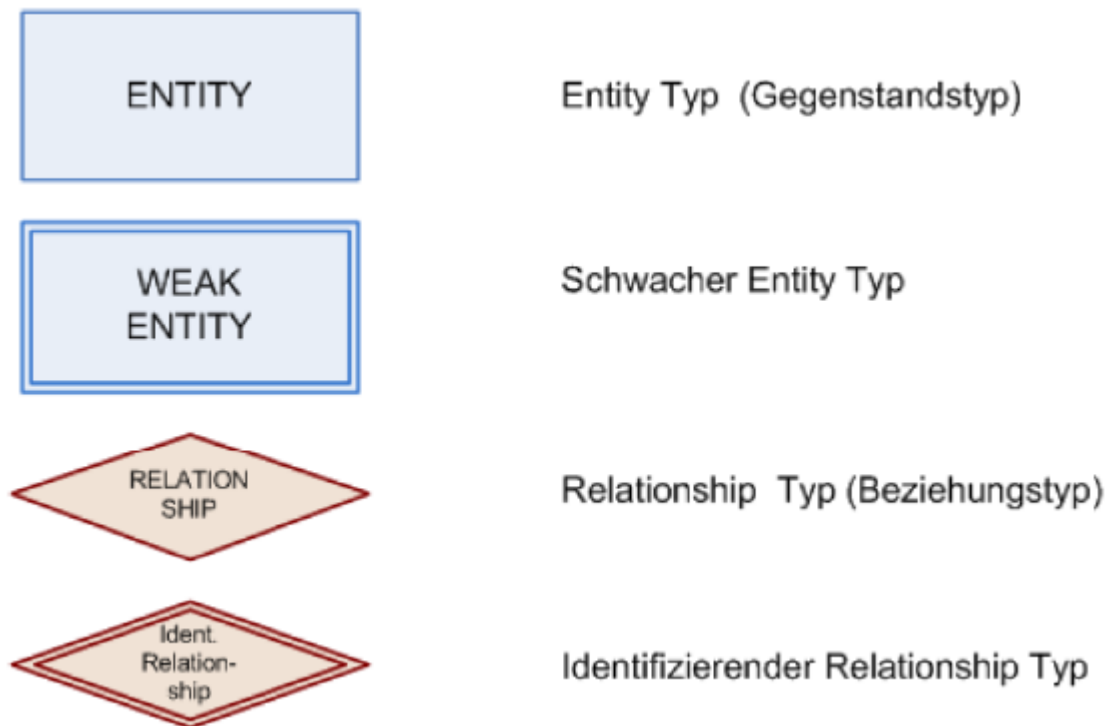
4.1.1 Allgemein

- Eigenschaften/Merkmale von Objekten eines Entity-Typs
- Typisierung gleichartiger Eigenschaften, z. B. Nachname, Vorname (Wikipedia)

4.1.2 Schlüsselattribute

- Attribut oder Attributkombination, die genau ein Entity identifiziert (d.h. es gibt keine Duplikate)

4.2 Grundlagen



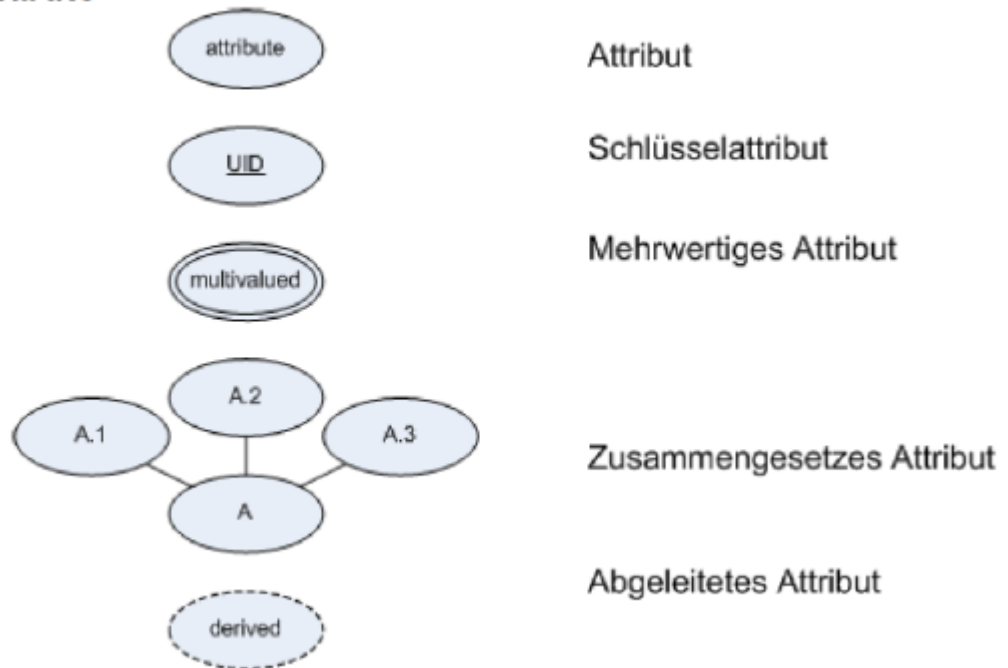
Eigenschaften schwacher Entity-Typen:

- Keine Schlüsselattribute
- Entities werden identifiziert über Entities anderer beteiligter Entity-Typen und partiellen Schlüssel

Eigenschaften Identifizierender Relationship-Typ:

- Doppellinie bei der Raute

Attribute



Funktionalitäten der Beziehungen



Totale Teilnahme von E2 in R

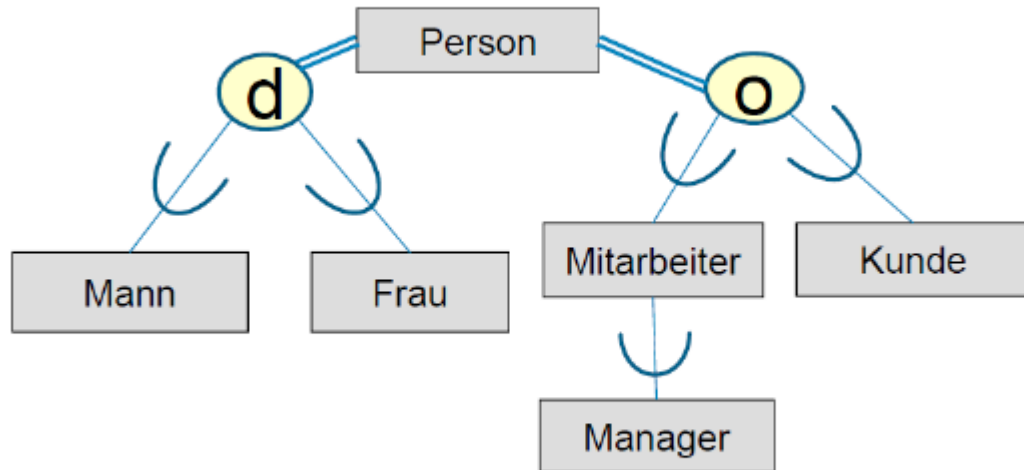


Kardinalität des Beziehungstyps R



4.3 Klassenhierarchien

⊗ EER-Notation nach Elmasri/Navathe („enhanced ER-Model“) Mehrere Klassenhierarchien

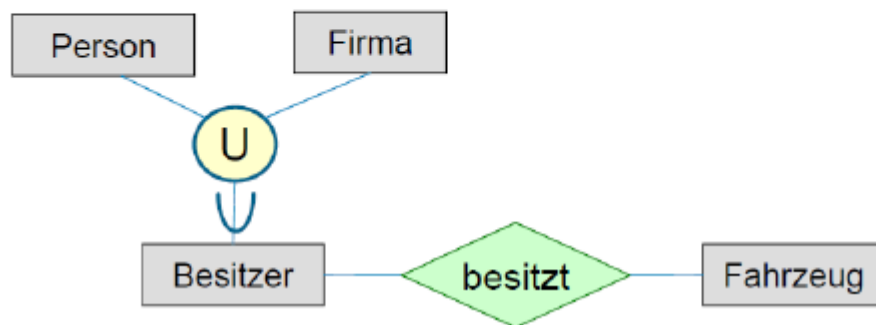


- Jede Person ist Mann oder Frau aber *nicht beides*
- Jede Person ist Mitarbeiter oder Kunde *oder beides*
- Ein Mitarbeiter *kann* Manager sein

Kategorie (Definition)

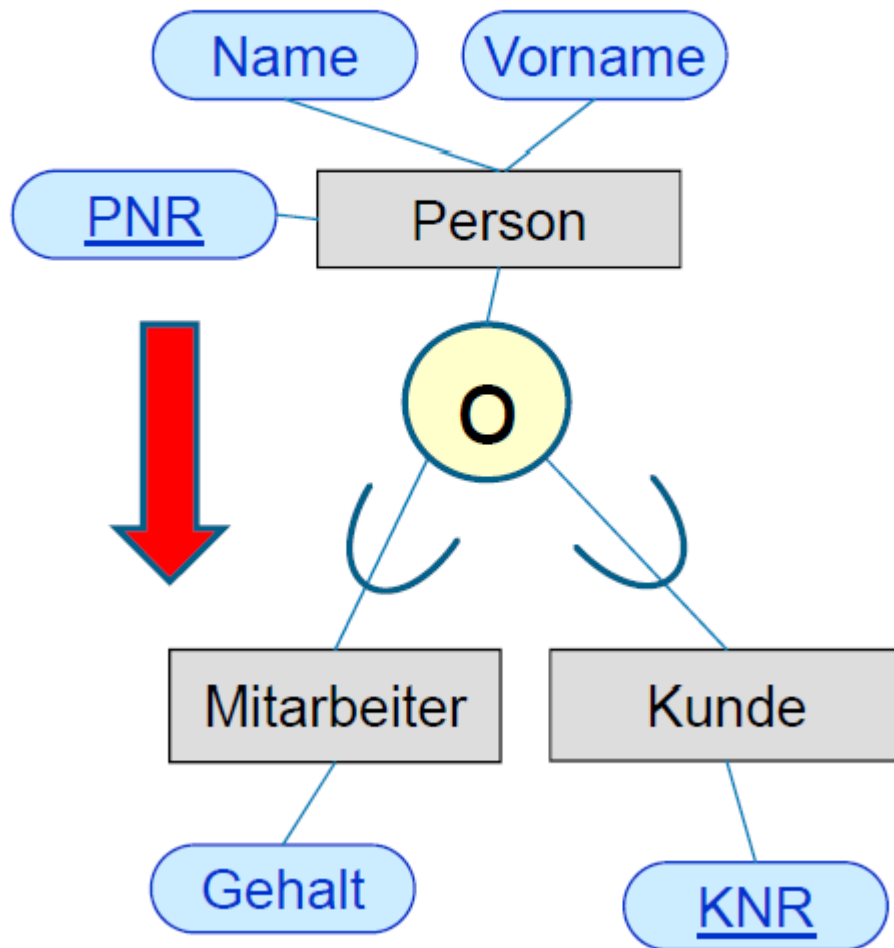
Eine Kategorie ist eine **Teilmenge der Vereinigungsmenge** von mehreren Entitätsmengen

MERKE !



4.4 Spezialisierung

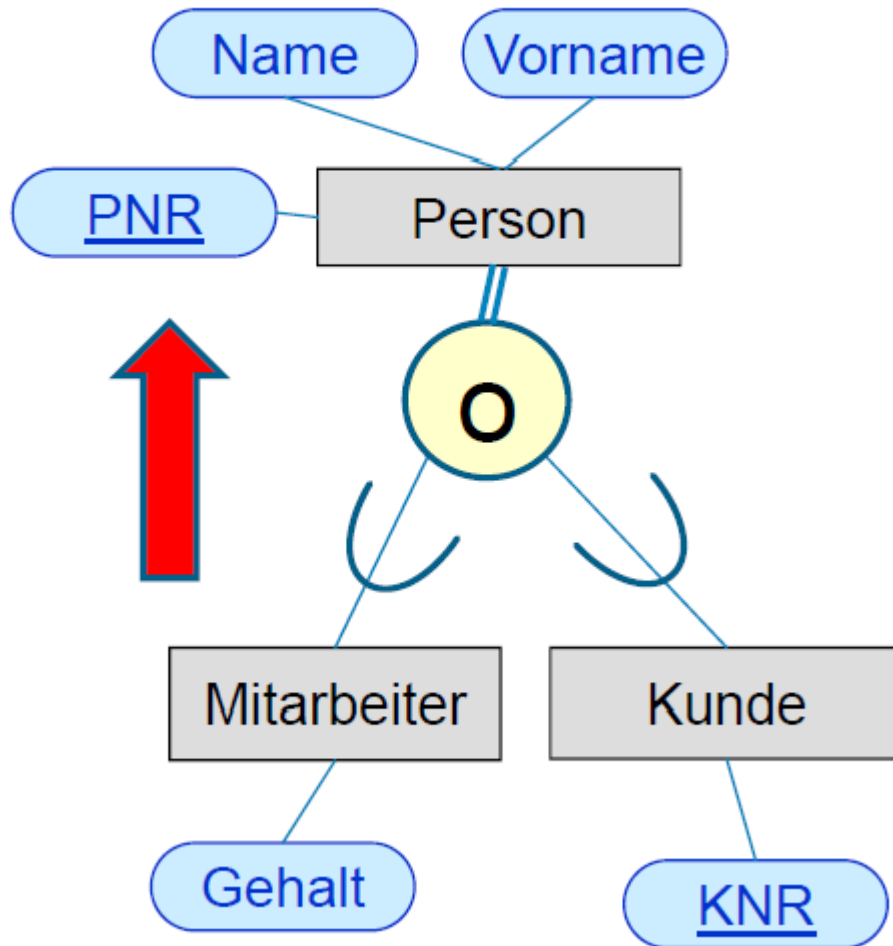
Spezialisierung



- Spezialisierung ist die Definition einer Menge von Subklassen eines Entity-Typs. (dbis.informatik.hu-berlin.de)

4.5 Generalisierung

Generalisierung



- Gleichartige Attribute werden in einer Superklasse zusammengefasst. (dbis.informatik.hu-berlin.de)

5 Datenbankbegriffe

5.1 Datenbank

5.1.1 Definition

- Sammlung zusammenhängender Daten
- Eine Datenbank repräsentiert einen Ausschnitt der realen Welt
- Gegenstand der Betrachtung = „Miniwelt“
- Logisch kohärente Sammlung von Daten
- Eine Datenbank hat einen definierten Zweck

5.1.2 Vorteile/Nachteile

Vorteile:

- Performanz
- Erweiterbarkeit, Flexibilität, Skalierbarkeit
- Verkürzte Entwicklungszeiten für Anwendungen
- Umsetzung von Standards

Nachteile:

- DBMS verursacht hohe initiale Kosten
- DBMS sind keine Spezialanwendungen
- DBMS verursacht hohen Overhead
- Erfordert entsprechende Rechenleistung

5.2 Datenbank Management System (DBMS)

- Sammlung von Programmen zur Verwaltung einer Datenbank
- Erzeugung von DB
- Wartung von DB
- Konsistenter Zugriff auf DB

5.3 Datenbankssystem (DBS)

DB + DBMS

5.4 Datenbankschema

Beschreibung einer konkreten Datenbank

5.5 Daten (bank) modell

Strukturierungsvorschrift für Daten (z.B. Tabellenform)

5.6 Tupel

Zeile einer Tabelle

5.7 Relation

Menge von gleichartig aufgebauten Tupeln

5.8 Attribut

Spalte einer Tabelle

5.9 Relationenschema [Intension]

Beschreibung einer Relation (Intension)

5.10 Entity-Begriffe

5.10.1 Entity

- Eigenständige Existenz
- Identifizierbar (= unterscheidbar)
- Beschreibbar (Merkmale, Attribute)
- Relevant (für die Anwendungswelt)

5.10.2 Entity-Set [Extension]

Die Menge der Entities, die zu einem Entity-Typ gehören ist die Extension dieses Typs.

5.10.3 Entity-Typ [Intension]

Beschreibung (Schema) einer Menge von Entities mit den gleichen Attributen

5.11 Attribute

5.11.1 Schlüsselkandidat

Attribut oder Attributkombination, die ein Tupel eindeutig identifiziert (hiervon können mehrere pro Relation existieren)

5.11.2 Primärschlüssel

Genau ein ausgewählter, kürzester Schlüsselkandidat

5.11.3 Fremdschlüssel

Attribut, das auf einen Schlüsselkandidaten verweist

5.12 Referenzielle Integrität

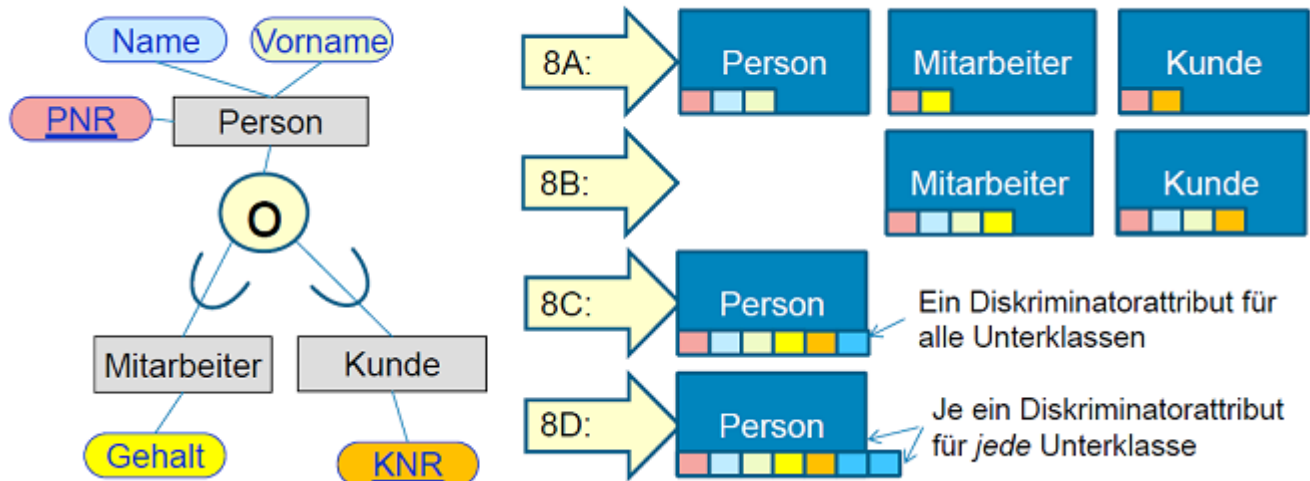
Jeder Fremdschlüsselwert muss auch tatsächlich als Primärschlüsselwert vorkommen. Kein Verweis darf ins Leere gehen! Es muss dem DBS explizit bekannt gemacht werden, so dass die Existenz des Primärschlüsselwerts bei Änderungen überprüft werden kann.

6 Mapping

Mapping allgemein:

ER-Modell	Relationenmodell
Entity-Typ	„Entity“-Tabelle
1:1 oder 1:N Beziehungstyp	Fremdschlüssel (FS) - oder
M:N Beziehungstyp	Beziehungstabelle mit 2 FS
N-ärer Beziehungstyp	Beziehungstabelle mit N FS
Einfaches Attribut	Attribut
Zusammengesetztes Attribut	Menge von Attributen
Mehrwertiges Attribut	„Attribut“-Tabelle mit FS
Wertebereich	Wertebereich
Schlüsselattribut	Schlüsselkandidat → Primärschlüssel

Mapping von Klassenhierarchien:



7 Anomalien

Typen von Anomalien:

- Einfüge-Anomalie: Liegt ein fehlerhaftes Datenbankdesign vor, kann es bei der Einfüge-Anomalie passieren das Daten gar nicht gespeichert/angenommen werden, wenn beispielsweise für den Primärschlüssel kein Wert eingegeben wird, oder es führt bei einer nicht vollständigen Eingabe von Daten zu Inkonsistenz.
- Lösch-Anomalie: Bei der Löschanomalie kann es passieren, dass man durch das bewusste Löschen eines Datensatzes, unbewusst Informationen verliert, die man später wieder gebraucht hätte.
- Änderungs-Anomalie: Bei der Änderungs-Anomalie oder auch Aktualisierungs-Anomalie genannt, werden gleiche Attribute eines Datensatzes nicht automatisch geändert. So entsteht eine Inkonsistenz der Daten. Man muss per Hand alle Einträge mühsam aktualisieren, es darf kein Fehler unterlaufen ansonsten führt es zur Inkonsistenz.

8 Normalformen

8.1 Ziel

Unter Normalisierung eines relationalen Datenschemas (Tabellenstruktur) versteht man die Aufteilung von Attributen (Tabellenspalten) in mehrere Relationen (Tabellen) gemäß den Normalisierungsregeln (s. u.), sodass eine Form entsteht, die keine vermeidbaren Redundanzen mehr enthält.

Ein konzeptionelles Schema, das Datenredundanzen enthält, kann dazu führen, dass bei Änderungen der damit realisierten Datenbank die mehrfach enthaltenen Daten nicht konsistent, sondern nur teilweise und unvollständig geändert werden, womit sie obsolet oder widersprüchlich werden können. Man sagt auch, dass Anomalien auftreten können. Zudem belegt mehrfache Speicherung derselben Daten unnötig Speicherplatz. (Wikipedia)

8.2 1.NF

Eine Tabelle ist in erster Normalform (1NF) wenn sie nur atomare Attributwerte besitzt, d.h. pro Datenfeld darf nur maximal ein Wert enthalten sein. Es gibt zwei Arten zusammengesetzte Attribute/Mehrfachattribute zu beseitigen:

- Das Mehrfachattribut wird innerhalb des Datensatzes in mehrere Einfachattribute zerlegt, d.h. der Datensatz erhält mehr Attribute. (info-wsf.de)

Beispiel:

Adresse wird zu Postleitzahl, Stadt, Straße und Hausnummer zerlegt.

- Wenn das Mehrfachattribut eine Liste von typgleichen Daten enthält, wird jedem Wert der Liste eine eigener Datensatz zugeordnet. (info-wsf.de)

Beispiel:

<u>MNr</u>	<u>Name</u>	<u>PrüfFachNr</u>
005	Meier	10,12,16

wird zu

<u>MNr</u>	<u>Name</u>	<u>PrüfFachNr</u>
005	Meier	10
005	Meier	12
005	Meier	16

(info-wsf.de)

8.3 2.NF

Eine Tabelle ist in zweiter Normalform (2NF) wenn sie in erster Normalform ist und alle Nicht-Schlüsselattribute voll funktional von jedem Schlüsselkandidaten abhängen.

In anderen Worten ausgedrückt:

Eine Relation ist in der zweiten Normalform (2.NF), wenn sie sich in der 1.NF befindet und jedes nicht zum Primärschlüssel gehörige Attribut voll von diesem abhängig ist. (info-wsf.de)

Beispiel:

<u>MNr</u>	<u>Name</u>	<u>PrüfFachNr</u>	<u>PrüfFachBezeichnung</u>
005	Meier	10	Analysis
005	Meier	12	Algebra
005	Meier	16	Datenbanken
006	Müller	16	Datenbanken
006	Müller	17	Techn. Informatik

wird zu

<u>MNr</u>	<u>PrüfFachNr</u>	<u>MNr</u>	<u>Name</u>	<u>PrüfFachNr</u>	<u>PrüfFachBezeichnung</u>
005	10	005	Meier	10	Analysis
005	12	006	Müller	12	Algebra
005	16			16	Datenbanken
006	16			17	Techn. Informatik
006	17				

Der Primärschlüssel der Relation ist MNr und PrüfFachNr. Da weder der Name des Studenten noch die Bezeichnung des Prüfungsfachs voll vom Primärschlüssel abhängig sind, müssen diese in andere Relationen überführt werden. (info-wsf.de)

8.4 3.NF

Eine Tabelle ist in dritter Normalform (3NF) wenn kein Nicht-Schlüsselattribut transitiv abhängig von einem Schlüsselkandidaten ist

In anderen Worten ausgedrückt:

Eine Relation befindet sich dann in der dritten Normalform (3.NF), wenn sie sich in der 1.NF und in der 2.NF befindet und wenn alle Nichtschlüssel-Attribute ausschließlich vom Primärschlüssel funktional abhängig sind, und nicht transitiv über ein Nichtschlüssel-Attribut. (info-wsf.de)

Beispiel:

<u>PrüfFachNr</u>	<u>PrüfFachBezeichnung</u>	<u>ProfNr</u>	<u>ProfName</u>
10	Analysis	90	A
12	Algebra	90	A
16	Datenbanken	65	D
17	Techn. Informatik	68	F

wird zu

PrüfFachNr	PrüfFachBezeichnung	ProfNr	ProfNr	ProfName
10	Analysis	90	90	A
12	Algebra	90	65	D
16	Datenbanken	65	68	F
17	Techn. Informatik	68		

Es fällt auf, dass das Attribut ProfName mehrmals vorkommt, obwohl mit ProfNr der Name des Professors schon gegeben wäre. Diese Redundanzen von Attributen, die nicht zum Primärschlüssel gehören, beseitigt die Überführung in die 3. Normalform. (info-wsf.de)

8.5 BCNF

Eine Tabelle ist in Boyce-Codd Normalform (BCNF), wenn jede Determinante einer FA ein Superschlüssel ist

bzw.

Eine Relation ist in Boyce-Codd-Normalform (BCNF), wenn jedes determinierende Attribut zugleich Schlüsselkandidat ist. (info-wsf.de)

Beispiel:

Belegung (KursNr, KursBez, SNr, Note)

In der Relation Belegung liegen folgende funktionale Abhängigkeiten vor:

$\{KursNr\} \rightarrow \{KursBez\}$

$\{KursBez\} \rightarrow \{KursNr\}$

$\{KursNr, SNr\} \rightarrow \{Note\}$

$\{KursBez, SNr\} \rightarrow \{Note\}$

Schlüsselkandidaten sind nur die Kombinationen KursNr, SNr und KursBez, SNr. Ein Schlüsselkandidat KursNr, KursBez, SNr ist nicht möglich, da damit die 2.NF nicht erfüllt wäre.

Damit ist aber die BCNF nicht erfüllt, da es Determinanten gibt, die keine Schlüsselkandidaten sind, nämlich KursNr bzw. KursBez. Um auch hier BCNF zu erreichen, muss wiederum eine Zerlegung durchgeführt werden, so dass eine derjenigen Determinanten, die nicht Schlüsselkandidaten sind, in eine neue Relation ausgelagert wird. Belegung (KursNr, SNr, Note) Kurs (KursNr, KursBez)

Die Untersuchung, ob sich eine Relation in BCNF befindet, macht nur Sinn, wenn die Relation zusammengesetzte Primärschlüssel besitzen, ansonsten ist die BCNF gleichzusetzen mit der 3.NF. Daher kann die Relation Belegung aus dem Beispiel auch durch die Einführung eines Attributes BNr als Primärschlüssel in die BCNF umgewandelt werden. (info-wsf.de)

8.6 4.NF

Eine Tabelle R ist in vierter Normalform (4NF), wenn für jede nicht-triviale Mehrwertige Abhängigkeit $X \twoheadrightarrow A$ in R gilt: X ist Superschlüssel von R

Superschlüssel: Jede Obermenge eines Schlüsselkandidaten ist ein Superschlüssel

Wann ist eine Relation nicht in 4NF ?

4 Kriterien:

- Die Tabelle muß mindestens drei Attribute haben
- A bestimmt mehrere Werte von B
- A bestimmt mehrere Werte von C
- B und C sind unabhängig voneinander

Auflösung der Redundanz:

- Jedes mehrwertig abhängige Attribut
- in eine eigene Tabelle extrahieren

9 ORACLE SQL

9.1 Create

```
CREATE TABLE inventory
(
id INT PRIMARY KEY,
product VARCHAR(50) UNIQUE,
quantity INT,
price DECIMAL(18,2)
);
```

9.2 Queries

– Inserts data into a SQL Database/Table

```
INSERT INTO orders (customer,day_of_order,product, quantity)
VALUES('Tizag','8/1/08','Pen',4);
```

– Selects data from a SQL Database/Table

```
SELECT * FROM orders;
```

– Updates data in a Database/Table

```
UPDATE orders SET quantity = '6'
WHERE id = '1'
```

9.3 Weiterführendes

9.3.1 Constraints

Definition von Integritätsbedingungen zur Begrenzung des Wertebereiches einzelner oder mehrerer Attribute. Constraints sind Tabellen und Spalten zuzuordnen. Typen von Constraints:

- NOT NULL
Spalte muß stets gefüllt sein.
- UNIQUE
Spalte oder Spaltenkombination ist eindeutig.
- PRIMARY KEY
Spalte oder Spaltenkombination ist Primärschlüssel.

- **FOREIGN KEY**
Spalte oder Spaltenkombination muß in einer separaten Tabelle als Schlüssel vorhanden sein.
- **ON DELETE CASCADE**
Löschen eines Datensatzes führt zum kaskadierenden Löschen der, über foreign key constraints verbundenen, Datensätze.
- **CHECK**
Boolscher Ausdruck ist wahr

Beispiel:

```
CREATE TABLE Persons(
P_Id int NOT NULL,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Address varchar(255),
City varchar(255),
CONSTRAINT chk_Person CHECK (P_Id>0 AND City='Sandnes')
);
```

(w3schools.com)

9.3.2 Trigger

http://de.wikibooks.org/wiki/Einführung_in_SQL:_Trigger

9.3.3 Top-N-Queries

 **Ermittle die TOP(10) der Spitzenverdiener für jede Abteilung und gib den Rang und den Namen aus**

(NB: Alle Angestellten haben unterschiedliches Gehalt!)

```
select      P. ANr, count(*) as Rang, P.Name
from        Pers P, Pers A
where       P.ANr = A.ANr and P.Gehalt <= A.Gehalt
group by    P.PNr, P.Name, P.ANr
having      count(*) <= 10
```

9.3.4 Assertions

```
create assertion totaleilnahme check(not exists
(
  select MuseumsID from Museum WHERE
  MuseumsID NOT IN (select befindet_sich_in from Kunstwerk)
)
```

9.4 Join-Arten

Grundsätzliche Syntax: SELECT <spaltenliste> FROM <linke tabelle> [<join-typ>]
JOIN <rechte tabelle> ON <bedingung>

- Mit einem LEFT JOIN erhält man alle Datensätze der linken Tabelle, ergänzt durch passende Angaben aus der rechten Tabelle.
- Mit einem RIGHT JOIN erhält man alle Datensätze der rechten Tabelle, ergänzt durch passende Angaben aus der linken Tabelle. übernommen, die nach der Verknüpfungsbedingung passen.
- Mit einem FULL JOIN erhält man alle Datensätze beider Tabellen, wenn möglich ergänzt durch passende Angaben aus der jeweils anderen Tabelle.

10 XML

10.1 Prolog

Der Prolog beginnt mit

```
<?xml version="1.0"
```

und endet mit

```
?>
```

Erweiterter Prolog:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
```

10.2 Element-/Attributdefinition

Beispiele zur Elementdefinition:

```
<!ELEMENT Personalien (Vorname+, Nachname)>
```

```
<!ELEMENT Anschrift (((Strasse, Hausnr?) | Postfach), PLZ, Ort)>
```

```
<!ELEMENT Vorname (#PCDATA)>
```

```
<!ELEMENT Nachname (#PCDATA)>
```

Inhaltsmodell kann sein:

- EMPTY
- ANY
- (#PCDATA)
- Kindelemente
- gemischt

```
<!ATTLIST Elementname Attributname Typ Attributbedingungen Attributname Typ  
Attributbedingungen ... >
```

Attributbedingungen:

- # REQUIRED
- # IMPLIED
- # FIXED "Vorgebung"

Attributtypen können sein:

- CDATA Zeichenkette
- ID Identifikationstyp (Werte müssen im Dokument eindeutig sein)
- IDREF Referenztyp (Verweis auf IDs)
- IDREFS Menge von Verweisen

- NMTOKEN Name Token (Zeichenkette in der nur bestimmte Zeichen zugelassen sind)
- NMTOKENS Menge von Token

10.3 Beispiel

```
<?xml version='1.0' encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE adresse SYSTEM "adresse.dtd">

<adresse>
  <einzeladresse>
    <name>Müller</name>
    <vorn>Lieschen</vorn>
    <strasse>Nirgendwostr. 10</strasse>
    <plz>10000</plz>
    <ort>Nirgendwo</ort>
    <tel art="priv">00000/12345</tel>
    <tel art="off">00000/54321</tel>
  </einzeladresse>
  <einzeladresse>
    <name>Maier</name>
    <vorn>Lieschen</vorn>
    <strasse>Brunnenstr. 12</strasse>
    <plz>34500</plz>
    <ort>Nirgendwo</ort>
    <tel art="priv">03030/11145</tel>
    <tel art="off">03330/55521</tel>
  </einzeladresse>
  <einzeladresse>
    <name>Maurer</name>
    <vorn>Hendrik</vorn>
    <strasse>Nirgendwostr. 14</strasse>
    <plz>10000</plz>
    <ort>Nirgendwo</ort>
    <tel art="priv">1234/12445</tel>
    <tel art="off">1234/44441</tel>
  </einzeladresse>
</adresse>
```

10.4 XPath

Die XML Path Language (XPath) ist eine vom W3-Konsortium entwickelte Abfragesprache, um Teile eines XML-Dokumentes zu adressieren und auszuwerten. XPath dient als Grundlage einer Reihe weiterer Standards wie XSLT, XPointer und XQuery. (Wikipedia)

10.4.1 Allgemein



(Siehe Beispieldaten oben) XPath-Ausdruck: `/adresse/einzeladresse/name`

Ergebnis:

`<name>Müller</name>` , `<name>Maier</name>` , `<name>Maurer</name>`

10.4.2 Prädikate

Prädikate

-  Ausdrücke in eckigen Klammern zur genaueren Spezifikation
-  Beispiel:

/AAA/BBB/

/AAA/BBB[1]/

```
<AAA>
  <BBB/>
  <BBB/>
  <BBB/>
  <BBB/>
</AAA>
```

```
<AAA>
  <BBB/>
  <BBB/>
  <BBB/>
  <BBB/>
</AAA>
```

Beispiel:

/AAA/BBB [last()]/

/AAA/*

```
<AAA>
  <BBB/>
  <BBB/>
  <BBB/>
  <BBB/>
</AAA>
```

```
<AAA>
  <BBB/>
  <CCC/>
  <DDD/>
  <EEE/>
</AAA>
```

④ Hierarchieunabhängige Knoten-Lokalisierung

④ Beispiel: **//BBB**

```
<AAA>
  <BBB/>
  <CCC/>
  <BBB/>
  <DDD>
    <BBB/>
  </DDD>
  <CCC>
    <DDD>
      <BBB/>
      <BBB/>
    </DDD>
  </CCC>
</AAA>
```

10.5 DTD

Eine Dokumenttypdefinition (englisch Document Type Definition, DTD, auch Schema-Definition oder DOCTYPE) ist ein Satz an Regeln, der benutzt wird, um Dokumente eines bestimmten Typs zu deklarieren. Ein Dokumenttyp ist dabei eine Klasse ähnlicher Dokumente, wie beispielsweise Telefonbücher oder Inventurdatensätze.

10.6 XML vs DTD

Vorteile von XML-Schema gegenüber DTD DTD:

- haben einen stark eingegrenzten Typvorrat
- unterstützen keine Namensräume (PCDATA)

XML-Schema:

- verfügt über objektorientierte Konzepte
- ist leicht erweiterbar
- unterstützt die Verwendung von Namensräumen
- umfangreiche Datenprüfungen
- hat den vordefinierten Datentyp „date“

11 Data Warehouse

11.1 Definition

unter dem Datawarehouse versteht man eine Datenbank, in der aus unterschiedlichen Quellen Daten geladen werden, die die Quelle für die Berichte und Analysen darstellen. Den Vorgang des Befüllens der Datawarehouse Datenbank nennt man den ETL-Prozess. Bei diesem Prozess wird zusätzlich auch noch sichergestellt, dass die Daten in sich konsistent sind und performant abgefragt werden können. (instantolap.de)

11.2 OLAP - OLTP

OLAP (Online Analytical Processing)

OLAP-Datenbanken (Online Analytical Processing) ermöglichen Business Intelligence-Abfragen. OLAP ist eine Datenbanktechnologie, die nicht für die Verarbeitung von Transaktionen, sondern für die Abfrage und Berichtserstellung optimiert wurde. Die Quelle für OLAP stellen OLTP-Datenbanken (Online Transactional Processing) dar, die meist in Data Warehouses gespeichert sind. OLAP-Daten sind von diesen herkömmlichen Daten abgeleitet und werden in Strukturen aggregiert, die eine anspruchsvolle Analyse zulassen. OLAP-Daten sind zudem hierarchisch organisiert und nicht in Tabellen, sondern in Cubes gespeichert. Es handelt sich um eine hoch entwickelte Technologie, die mithilfe von mehrdimensionalen Strukturen schnellen Zugriff auf Daten für die Analyse ermöglicht. (office.microsoft.com)

OLTP (Online-Transaction-Processing)

Mit OLTP (Online-Transaction-Processing, auf Deutsch manchmal mit Online-Transaktionsverarbeitung übersetzt) bezeichnet man in IT Systemen eine unmittelbare, direkte Verarbeitung von Benutzeraktivitäten. Der Benutzer/Client ist direkt (online) mit dem System verbunden, und seine Eingaben werden sofort verarbeitet. OLTP Anwendungen verarbeiten die Transaktionen aus den operativen Geschäftsfällen und stellen daher hohe Anforderungen bezüglich Verfügbarkeit und Reaktionszeiten an das dem System zugrundeliegende Datenbanksystem. Hingegen sind die einzelnen Verarbeitungsschritte typischerweise kurz und von geringer Komplexität. Ein Großteil der Web-Applikationen, aber auch viele Back-Office Anwendungen (z.B. ERP Systeme) sind OLTP Systeme oder haben zumindest wesentliche OLTP Komponenten. (db-engines.com)

11.3 Kennzahlen

Hierbei handelt es sich um die Fakten, die analysiert werden sollen, wie z.B. Umsatz, verkaufte Menge, Kosten, Gewinn, etc. (instantolap.de)

11.4 Relationenmodell

- Nützlich in beliebigen Domänen
- Manchmal etwas kompliziert in der Anwendung

11.5 Multidimensionales Datenmodell

- Nur nützlich für analytische Anwendungen, für diese aber sehr mächtig

11.5.1 Daten

Mikro-Daten

- Einzelne Observationen / beschreiben Elementarereignisse
- Ergebnis der Ladephase
- Basisdaten

Makro-Daten

- Aggregierte Daten für die Datenanalyse
- Ergebnis der Auswertungsphase data warehouse, data mart

Meta-Daten

- Beschreibungsdaten
- Beschreiben die Eigenschaften von Mikro-Daten und Makro-Daten
- Beschreiben auch den Entstehungsprozess

11.6 Operationen

11.6.1 Slicing und Dicing

Slicing:

- Bei dem Slicen wird aus dem OLAP-Würfel eine Scheibe “herausgeschnitten”. Z.B. das Produkt A über den gesamten Zeitraum und über alle Länder. Die Scheibe ist eine Teilmenge des OLAP-Würfels in einer Dimension auf einen festen Wert eingeschränkt.

Dicing:

- Dicing: Gleichzeitige Slicing-Vorgänge in unterschiedlichen Dimensionen. Hierbei wird ein kleinerer Würfel erzeugt, der einen Teilbereich des Gesamtwürfels enthält.
-> Herausschneiden eines kleineren Würfels.

(instantolap.de)

11.6.2 Drill Down

Drill-Down:

- Die Drill-Down Funktionalität wird genutzt um Analysen schrittweise noch zu verfeinern. Im oberen Beispiel könnte von dem Land noch auf die Region bis hin zur Stadt “Hineingezoomt” werden.

(instantolap.de)

11.6.3 Drill-Up/Roll-Up

Drill-Up/Roll-Up:

- Gegenoperation zu Drill-Down; Hierbei handelt es sich um das verdichten der Kennzahlen von der untersten Hierarchieebene schrittweise bis hin zur obersten Ebene zu aggregieren.

(instantolap.de)

11.6.4 Drill-Across

- Verknüpfung mehrerer Datenwürfel mit gemeinsamen Dimensionen
- Binäre Operationen zwischen mehreren Datenwürfeln: “Cubic Join”

Beispiel:

Verkäufe * Preis = Umsatz

11.6.5 Drill-Through

- Beim Drill-down ist bei der feinsten Granularität einer Dimension Schluss
- Drill-through: Wechsel zu den Originaldaten

11.6.6 Pivotierung

- Wechsel der Darstellung in einer Pivottabelle
- Entspricht Drehen des Würfels
- Operation – Bestandteil der Präsentationsschicht nicht Datenmodells

11.6.7 Aggregationen

Zusammenfassen mehrerer Zellen

Operationen zur Aggregation

- Standard: SUM, AVG, MIN, MAX, COUNT
- Ordnungsbasierte Aggregation: cumulating, ranking (Top(N))

Problem

- Nicht alle Funktionen sind summierbar (Median, Standardabweichung)
- Unter Umständen dürfen selbst einfache Aggregationsfunktionen (SUM, AVG, MIN, MAX, COUNT, ...) nicht angewendet werden -> Anwendungssemantik!

11.7 Darstellungsmöglichkeiten

11.7.1 Multidimensional

- + Straightforward
- - Dünnbesetztheit: In typischen Datenwürfeln sind weniger als 5% der Zellen besetzt!
- - Skalierbarkeit: Multidimensionale Arrays mit mehreren hundert GB Daten sind nicht einfach abzuspeichern!

11.7.2 Relational

- + Skalierbare, ausgereifte Technologie
- - mangelnde Unterstützung an Operatoren
- - Relationale Datenbanken sind für OLTP und nicht für OLAP konzipiert
- - Teilweise schlechtere Performance

11.7.3 Hybrid

- + Vorteile beider Varianten
- - Komplexität

11.8 ROLAP - Relationale Abbildung

11.8.1 Star Schema & Snowflake Schema

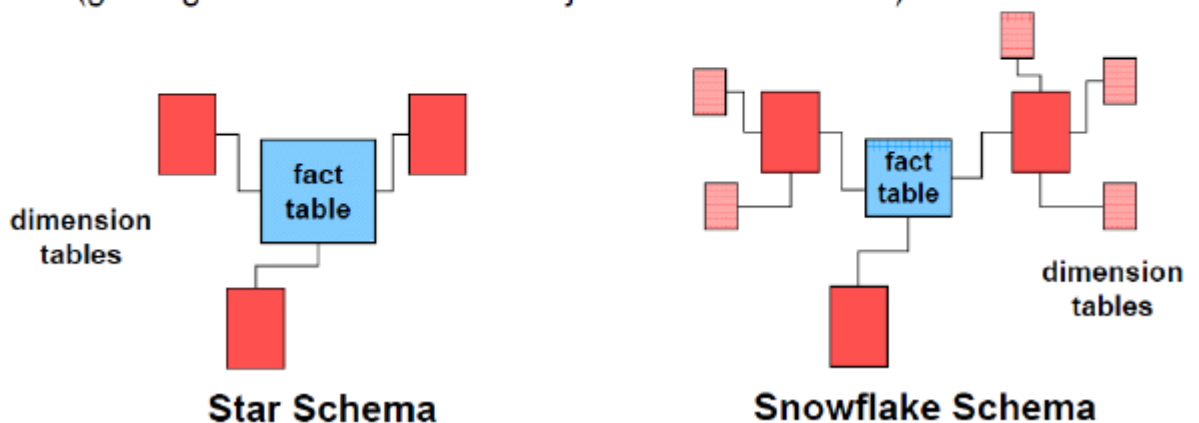
Das Schneeflockenschema ist eine Weiterführung des Sternschemas, das beim OLAP und Data-Warehousing eingesetzt wird. Beim Sternschema liegen die Dimensionstabellen denormalisiert vor, was eine bessere Verarbeitungsgeschwindigkeit zu Lasten der Datenintegrität und des Speicherplatzes mit sich bringt. Im Gegensatz dazu werden beim Schneeflockenschema die einzelnen Dimensionstabellen verfeinert, indem sie klassifiziert oder normalisiert werden. Durch diese Weiterverzweigung des Datenmodells entsteht die Form einer Schneeflocke, woher die Bezeichnung dieses Entwurfsmusters stammt.

- **Star Schema**

- Eine Tabelle für jede Dimension

- **Snowflake Schema**

- Normalisierung der Dimensionstabellen
 - Viele Tabellen je Dimension
(genaugenommen: eine Tabelle je Klassifikationsstufe)



12 UML

12.1 Anwendungsfalldiagramm

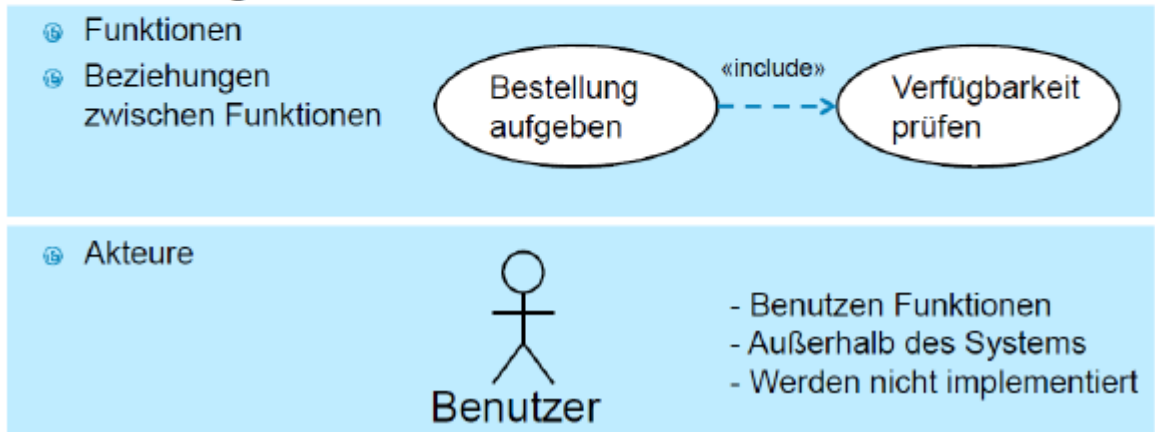
Anwendungsfalldiagramm (Use Case-Diagramm)

10

🔗 Anwendungsfall / Use Case

- 🔗 Ziel: Funktionale Beschreibung eines Systems
- 🔗 Use Cases beschreiben Benutzeranforderungen

🔗 Modellierungselemente



« include »

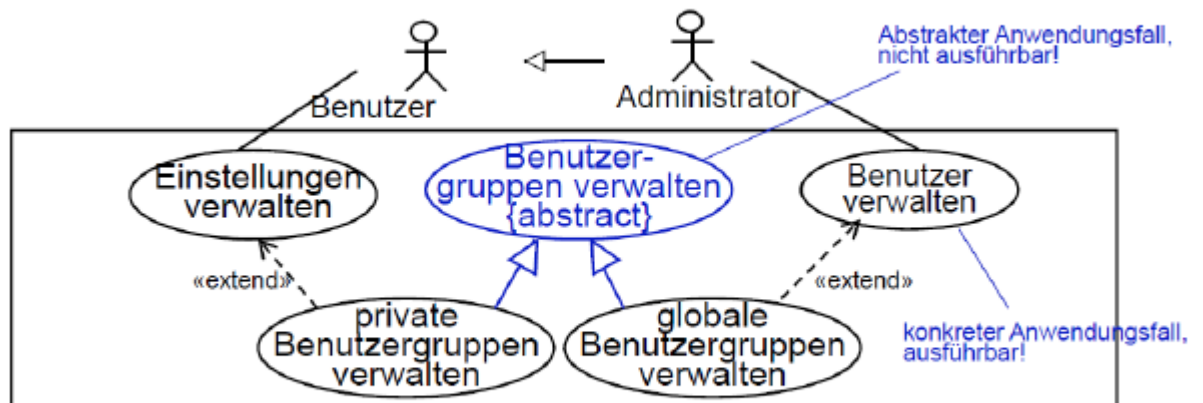
- auch: «benutzt»
- Eine Teilfunktion (Use Case) wird in mehreren anderen Funktionen (Use Cases) wiederverwendet
- Funktionale Dekomposition

« extend »

Vorgehen für «extend»-Beziehungen:

- Auch «erweitert»
- Variation des normalen Verhaltens
- Behandlung von Ausnahmefällen

Abstrakter Anwendungsfall



Ziel: Wiederverwendbarkeit eines gemeinsamen Verhaltens

12.2 Aktivitätsdiagramm

Aktionsknoten:

- Repräsentieren vordefinierte UML-Aktionen
- Empfangen Eingaben
- Produzieren Aktion
- Ausgaben für andere Knoten



Kontrollknoten:

- Steuern Aktivitätsabläufe
- Festlegung von Start und Ende einer gesamten Aktivität

Kontrollknoten

Initialknoten:

- kennzeichnet Beginn eines Aktivitätsablaufs ●

Aktivitätssendknoten:

- beendet *alle* Abläufe einer Aktivität ● («Fisheye»)

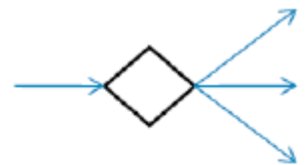
Ablaufendknoten:

- beendet nur *einen* bestimmten Ablauf einer Aktivität ⊗

Kontrollknoten ...

Entscheidungsknoten:

- Abhängig von einer Bedingung wird ein Ausgangspfad gewählt



Vereinigungsknoten:

- Führt alternative Pfade zusammen



Parallelisierungsknoten:

- Parallele Verzweigung



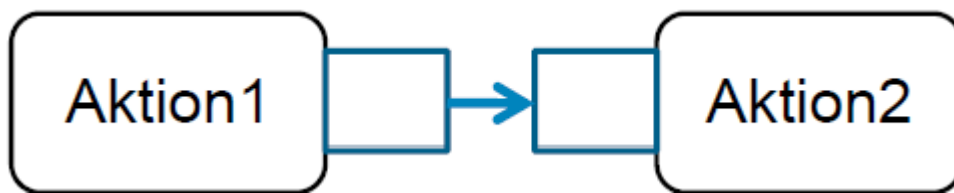
Synchronisierungsknoten:

- Zusammenführung paralleler Pfade



Objektknoten:

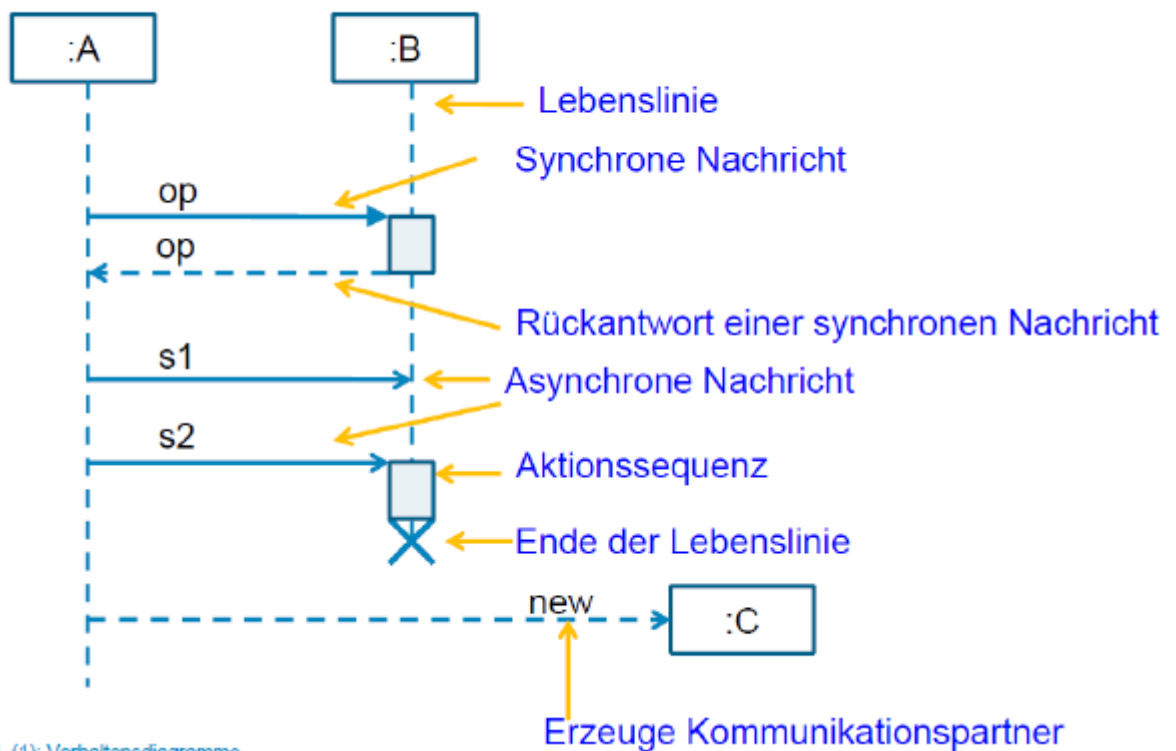
- Ein- und Ausgabeparameter (d.h. Bindeglied zwischen Verhaltensmodellierung Aktivitätsdiagramm) und Strukturmodellierung



12.3 Sequenzdiagramm

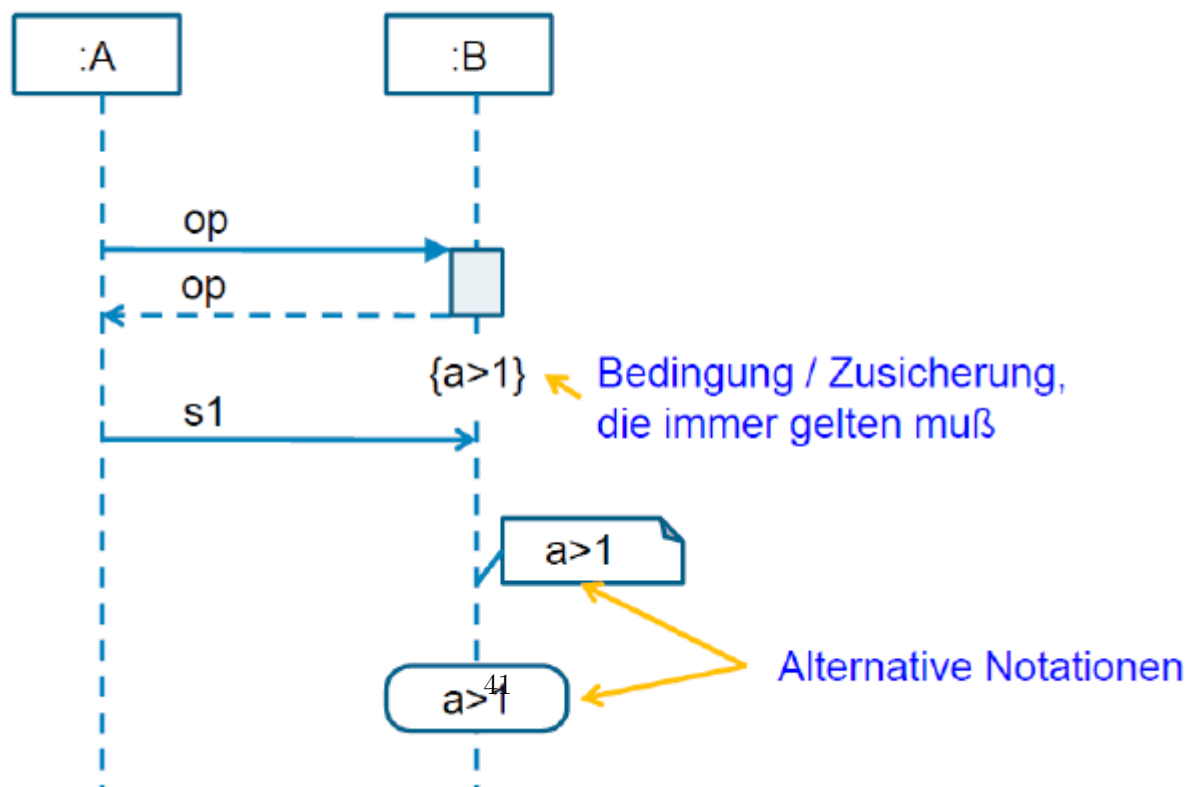
12.3.1 Grundlagen

Beispiel



UML / 13: Verhaltensdiagramme

Zustandsinvarianten

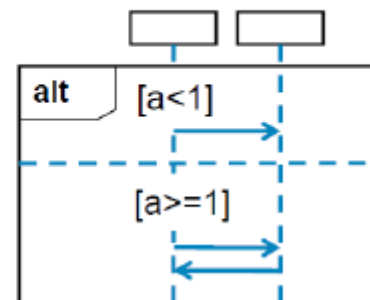


12.3.2 Operatoren

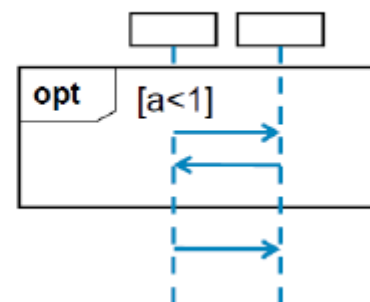
Operatoren für kombinierte Fragmente

⊕ Verzweigungen und Schleifen

- ⊕ alt Alternative Interaktionen
2 oder mehr Operanden



- ⊕ opt Optionale Interaktionen

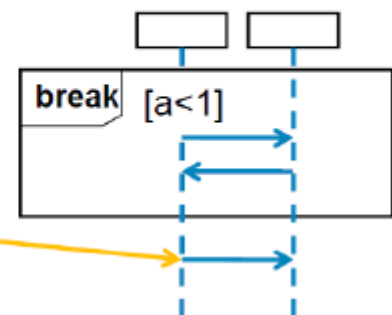


Operatoren für kombinierte Fragmente

⊕ Verzweigungen und Schleifen

- ⊕ break Ausnahme Interaktionen

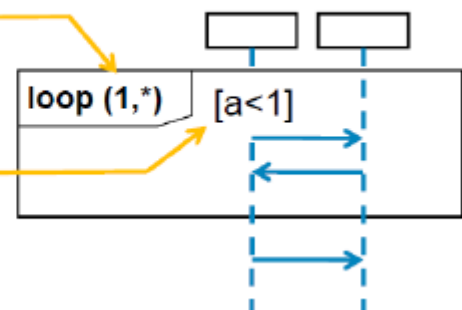
Nachfolgende Interaktionen werden
im Falle des break ($a < 1$)
nicht mehr ausgeführt



- ⊕ loop Iterative Interaktionen

Zählergrenzen

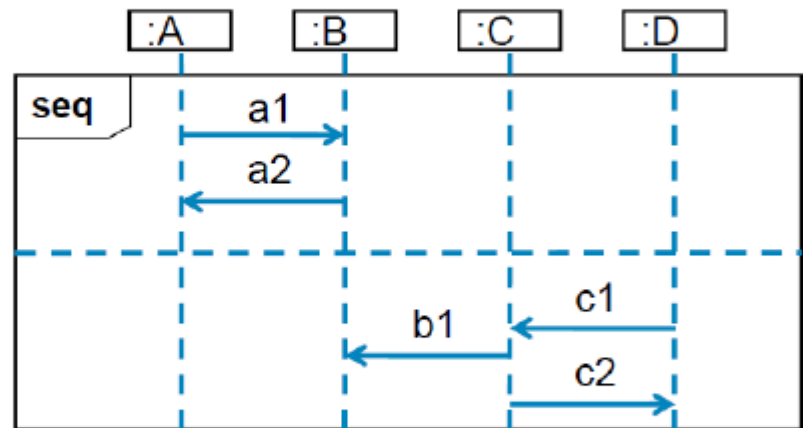
Abbruchkriterium



Ⓢ Operatoren für kombinierte Fragmente

Ⓢ Nebenläufigkeit und Ordnung

Ⓢ seq Sequentielle Interaktionen mit schwacher Ordnung



Ⓢ Mögliche Traces:

$a1 \rightarrow a2 \rightarrow c1 \rightarrow b1 \rightarrow c2$

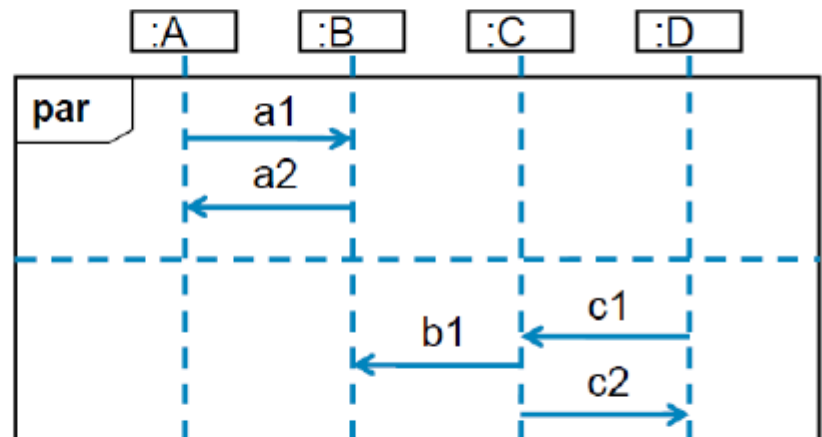
$a1 \rightarrow c1 \rightarrow a2 \rightarrow b1 \rightarrow c2$

$c1 \rightarrow a1 \rightarrow a2 \rightarrow b1 \rightarrow c2$

⌚ Operatoren für kombinierte Fragmente

⌚ Nebenläufigkeit und Ordnung

⌚ par Nebenläufige Interaktionen



⌚ Zulässige Traces z.B:

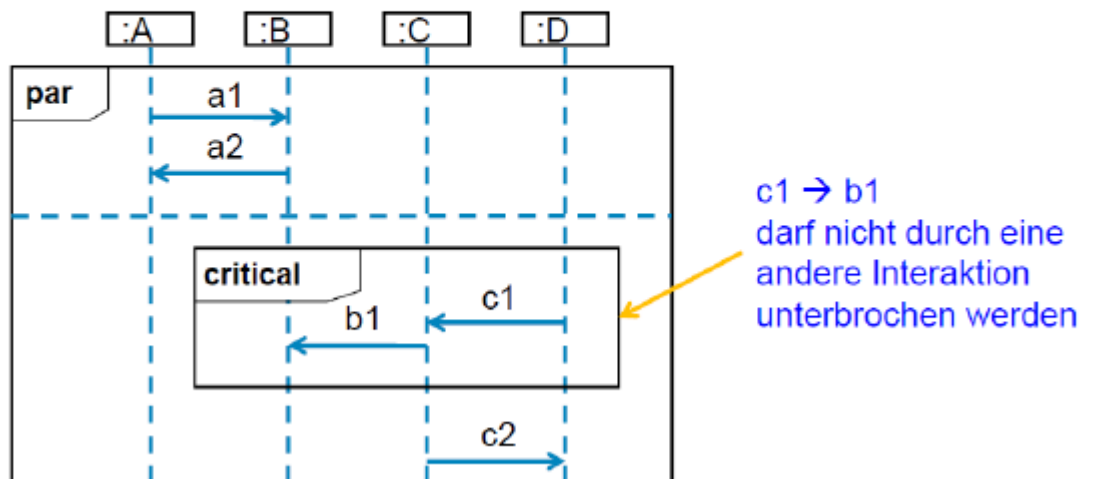
- $c1 \rightarrow b1 \rightarrow c2 \rightarrow a1 \rightarrow a2$
- $c1 \rightarrow a1 \rightarrow b1 \rightarrow a2 \rightarrow c2$
- $a1 \rightarrow c1 \rightarrow b1 \rightarrow c2 \rightarrow a2$

④ Operatoren für kombinierte Fragmente

④ Nebenläufigkeit und Ordnung

④ critical

Atomare Interaktionen



Operatoren für kombinierte Fragmente
ignore

- unwichtige Interaktionen
- Interaktionen spielen für die darzustellende Problematik keine Rolle

consider

- wichtige Interaktionen können aber vorkommen Interaktionen für sind die darzustellende Problematik besonders wichtig

neg











- ungültige Interaktionen
- Explizite Modellierung eines ungültigen Interaktionsablaufs
- Vorsicht! – Nicht vollständig!
- Eignet sich um vor häufigen Fehlern zu warnen

assert

- Zusicherung
- Einzig gültiges Verhalten
- Nicht modellierte Ereignisse dürfen auch nicht auftreten

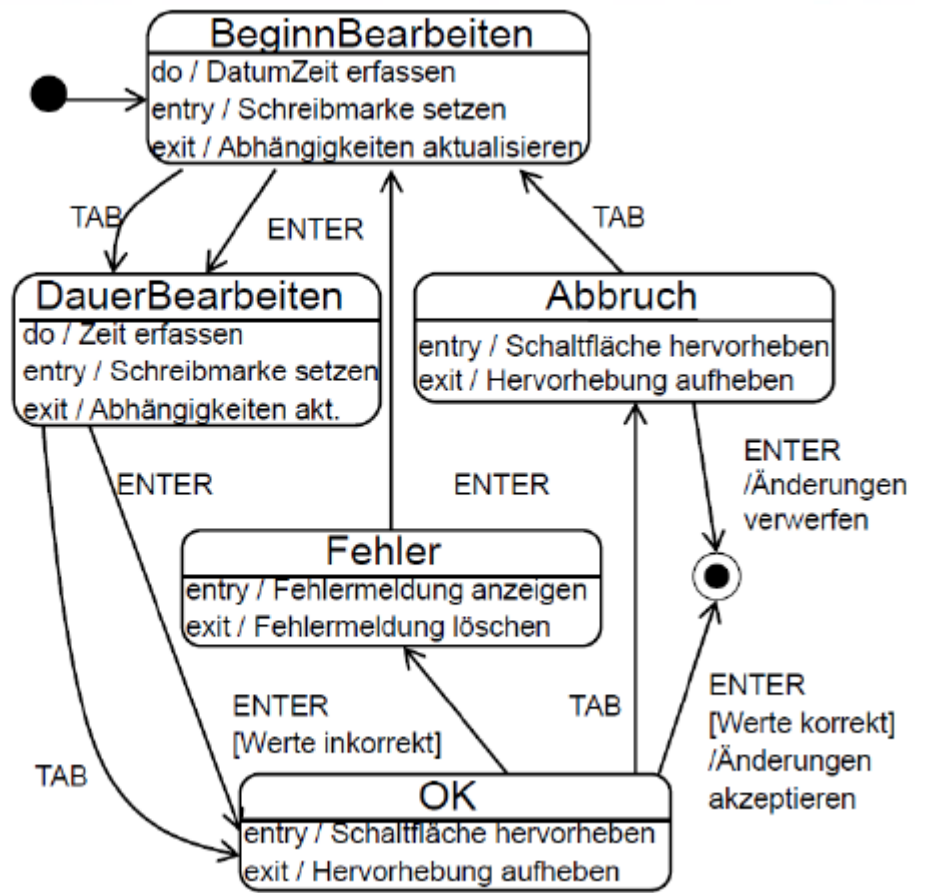
12.4 Zustandsdiagramm

Zustände und Zustandsübergänge

-  Startzustand ●
-  History-Zustand 
= Rücksprungadresse
-  Verbindungsstellen ●
-  Entscheidungsknoten ◇
-  Nebenläufigkeitsbalken
-  Einstiegs- O /Ausstiegspunkt ⊗
= dienen der Verwendung von Unterautomaten
-  Terminierungsknoten X
 -  gibt an, daß das Objekt, dessen Verhalten modelliert wird, aufhört zu existieren
-  Endzustand ●

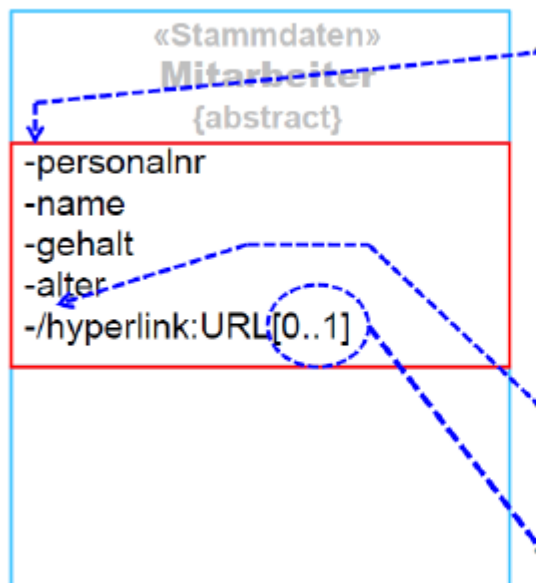
- Zustand = Objekt in bestimmter Situation kann auf äußere Ereignisse reagieren
Notation: Zustandsname + Aktivitäten
- do/ = Aktivitäten während Objekt im Zustand
- entry/ = wenn Objekt in Zustand eintritt
- exit/ = sobald Zustand verlassen wird

⑤ Beispiel



12.5 Klassendiagramm

④ Basisnotation für Klassen: Attribute



Sichtbarkeitsvermerk

- + öffentlich
- ~ innerhalb eines Pakets
- nur für diese Klasse
- # für diese Klasse und ihre Unterklassen

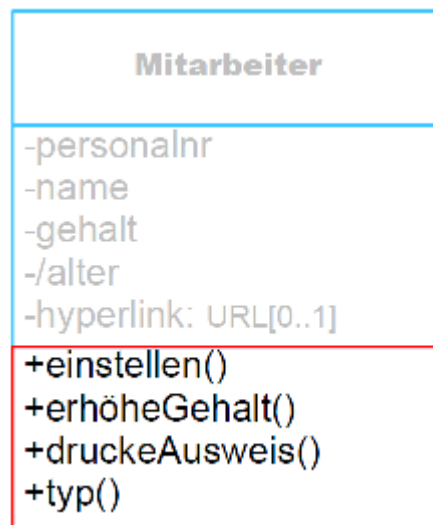
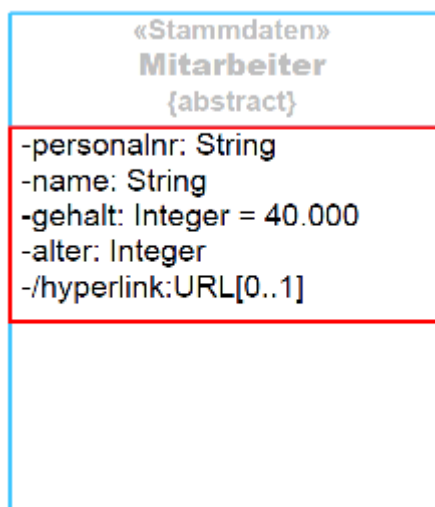
Abgeleitete Attribute:

/ vorangestellter Schrägstrich

Multiplizitätsangabe:

Anzahl der Werte, die das Attribut annehmen kann

UML-Klassendiagramm 2013 / 2014



Aggregation



- ☞ Teil – Ganzes – Beziehung

- ☞ Hier: Ein Eintrag kann Bestandteil einer Liste sein
kann aber auch unabhängig davon existieren
und kann auch in mehreren Listen stehen

Komposition

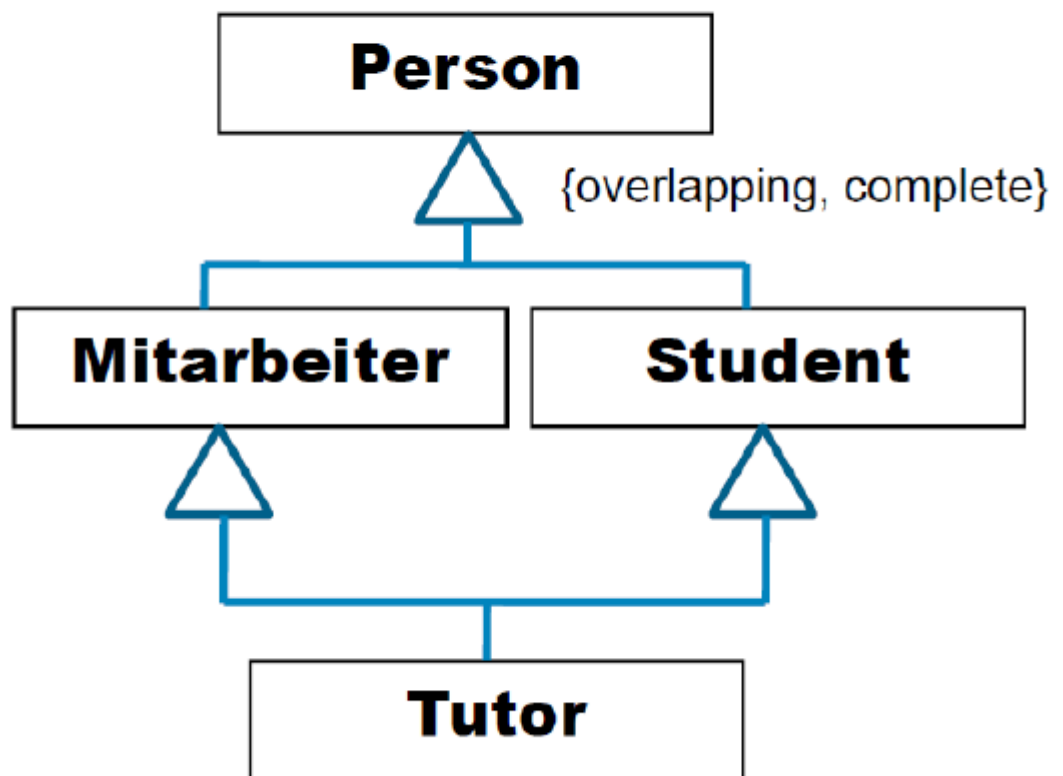
- ☞ Strenge Teil – Ganzes – Beziehung



- ☞ Hier: Ein Kompositionsteil darf höchstens zu einem Ganzen gehören
- ☞ Schwache Aggregationsbeziehungen zu diesem Teil sind nicht erlaubt
- ☞ Mögliche Multiplizität: 1..1 oder 0..1
 - ☞ 1..1 : Teil ist vom Ganzen existenzabhängig
 - ☞ 0..1 : Teil kann auch ohne Ganzes existieren

Mehrfachvererbung

- Eine Klasse ist Unterklasse mehrerer Oberklassen



12.6 Modelle

Ebene	Beschreibung	Beispiel
Meta-Metamodell	Sprache zur Beschreibung von Metamodellen (→ „kernel“)	Metaklasse, Metaoperation, Namespaces, Multiplicities,...
Metamodell	Ausprägung eines Meta-Metamodells Sprache zur Beschreib. von Modellen	Actor, UseCase, ExtensionPoint, ... LifeLine, CombinedFragment, ... Activity, Gate, Join-/ForkNode, ...
Modell	Eine Ausprägung eines Metamodells Beschreib. eines Anwendungsbereichs	Bestellung, pruefe_Zahlung, ...
Anwendungs- objekte	Ausprägung eines Modells	Bestellung4712, pruefe_Zahlung(objekt9311)

13 Ontologie

In einer globalisierten Welt wird ein organisations- und systemübergreifender Datenaustausch immer wichtiger.

Problem:

- Heterogene Systeme
- IT Systeme werden unabhängig voneinander entworfen
- Basieren auf unterschiedlichen Modellen
- Unterschiedliche Begriffe und Konzepte

→ Datenaustausch wird erschwert

→ Datenqualität sinkt!

Idee:

Referenzmodell zur Festlegung von Konzepten und Begriffen, die typisch sind für eine bestimmte Anwendungsdomäne

= Ontologie

Terminologische Kontrolle:

Alle Maßnahmen, die direkt oder indirekt der Definition und Abgrenzung der Begriffe und der Zuordnung von Benennungen und Begriffen dienen werden als terminologische Kontrolle bezeichnet.

Definition nach Tom Gruber (1993):

An ontology is a formal explicit specification of a shared conceptualization

Für uns – “Arbeitsdefinition”: Spezifikation von gemeinsam zu verwendenden Begriffen und deren Zusammenhängen

Zweck: Gemeinsames Verständnis

- Ressource (Resource): Ding, das durch einen RDF-Ausdruck beschreibbar ist, z.B. eine Webseite <http://www.w3.org/Overview.html>, ein Teil oder eine Menge von Seiten, eine eigene Domäne oder ein nicht direkt adressierbares Ding wie z.B. ein Buch. Resources werden durch URI's (Universal Resource Identifiers) identifiziert
- Eigenschaft (Property): Aspekt, Charakteristik, Attribut oder Beziehung, die benutzt wird, um eine Ressource zu beschreiben. Jede Eigenschaft hat eine spezifische Bedeutung, definiert ihre zugelassenen Werte, die Typen der beschriebenen Ressourcen und ihre Beziehung zu anderen Eigenschaften.
- Objekt: kann ein String, ein in XML definierter primitiver Datentyp oder eine weitere Ressource sein.

- Anweisung (Statement): Eine spezifische Ressource zusammen mit einer bezeichneten Eigenschaft und einem zugehörigen Wert.
Analogie: Subjekt - Prädikat - Objekt