

<Object-Relational Mapping:>

Möglichkeiten & Vorgehensweisen

Referent:
Michael Voss
Leiter Training/Coaching

member
of **< ssd > f**
strategic software development
factory

Das innovative
Systemhaus



Agenda

- Die MID GmbH
- Motivation für eine Integration
- Anforderungen an eine Integration
- Lösungsansätze
- Vorgehensweisen
- Diskussion

member
of **< ssd > f**
strategic software development
factory

Das innovative
Systemhaus



Die MID GmbH

→ ein Unternehmen mit Erfahrung

- ⇒ erfolgreich seit 1980
- ⇒ mit über 80 Mitarbeitern
- ⇒ mehr als 20.000 Installationen
- ⇒ bei mehr als 1.600 Kunden aus allen Branchen

von **innovator**® BPE- und CASE-Workbenches

→ ein starker Partner

- ⇒ für große und kleine IT-Projekte
- ⇒ für System-Integratoren und Promotionpartner
- ⇒ als Founding Member der <ssd>f

member
of **< ssd > f**
strategic software development
factory

Das innovative
Systemhaus



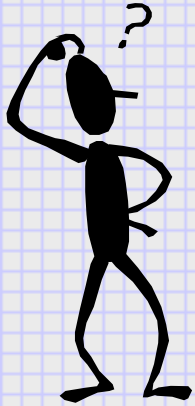
Motivation für eine Integration

member
of **< ssd > f**
strategic software development
factory

Das innovative
Systemhaus



Anforderungen an heutige Softwareprojekte



- ➔ mehrschichtige Client-/Server-Architekturen
- ➔ kurze Entwicklungszyklen
- ➔ Einsatz zeitgemäßer Programmiersprachen
- ➔ WEB, EJB, XML, EAI, E-XXX, ...
- ➔ moderne Modellierungstechniken
- ➔ ?



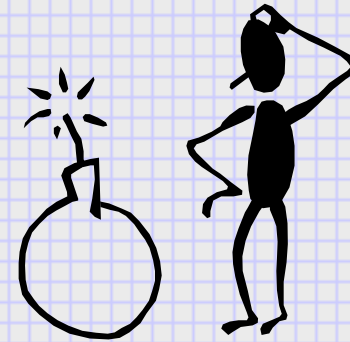
member
of **< ssd > f**
strategic software development
factory

Das innovative
Systemhaus



Realität in laufenden Projekten

- ➔ Kosten- und Termindruck
- ➔ beschränkte Ressourcen
- ➔ fehlende Erfahrung mit neuen Technologien
- ➔ Dominanz relationaler Datenbanken
- ➔ Investitionen
- ➔ Massendatenbestände



member
of **< ssd > f**
strategic software development
factory

Das innovative
Systemhaus



Die semantische Lücke

Objektorientierte Modellierung und Programmierung

?

Zwei Aspekte

- Modellierung
- Laufzeitunterstützung

Physisches relationales Datenbankschema

member
of

< ssd > f
strategic software development
factory

Das innovative
Systemhaus



Anforderungen an eine Integration

member
of **< ssd > f**
strategic software development
factory

Das innovative
Systemhaus



Begriffe der Datenbankwelt

Relation (Tabelle) → KUNDE

Attribut/Spalte (Merkmal, Eigenschaft) → Nummer, Name, Firma, Adresse

Attributwert (Ausprägung) → Badstr. 5, 54321 Hausen

Tupel (Satz, Record) → 4813, Huber, Neue Str. 3, 67890 Neustadt

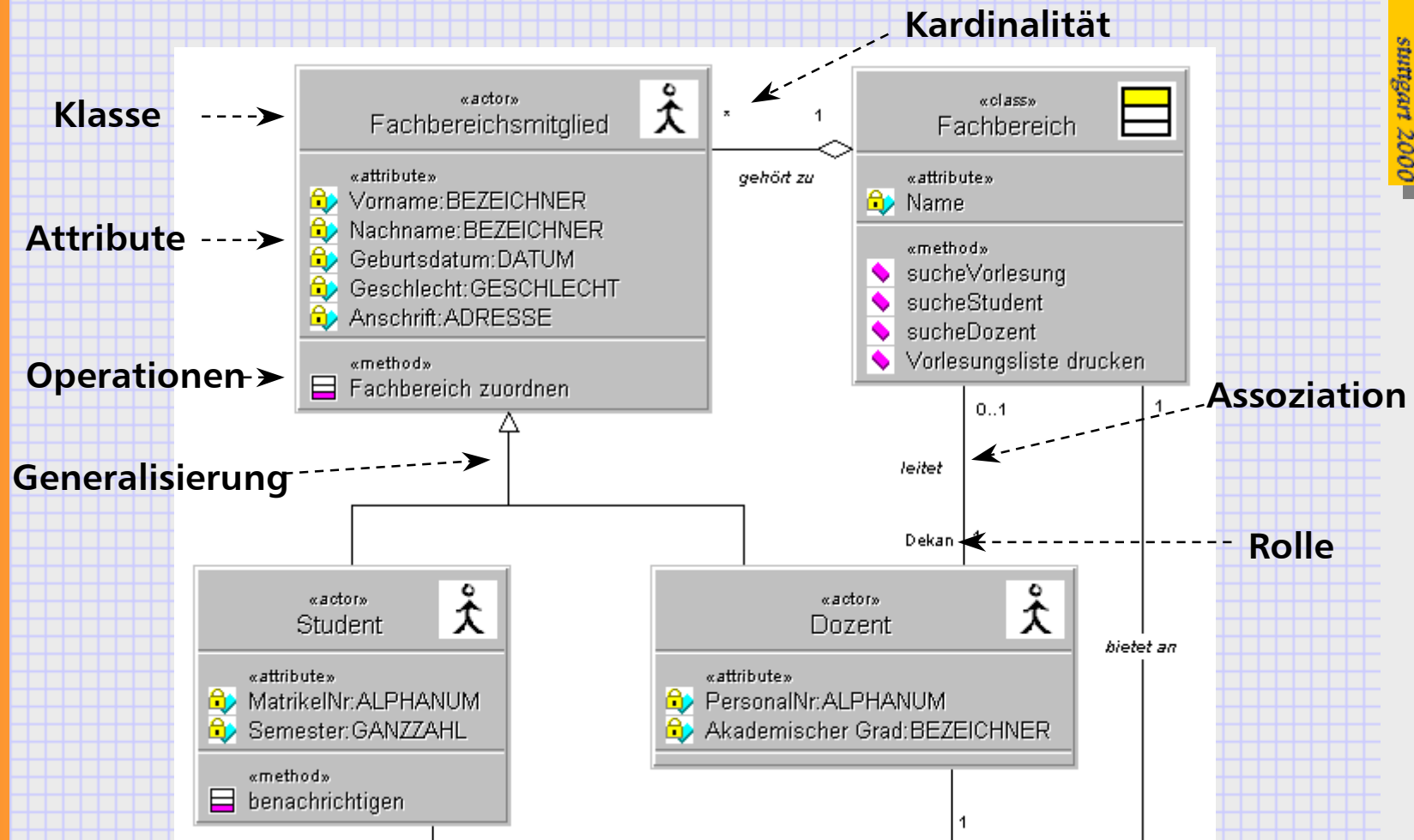
KUNDE	Nummer	Name	Firma	Adresse
	4712	Meier	Schmidt KG	Badstr. 5 54321 Hausen
	4813	Huber		Neue Str. 3 67890 Neustadt
	4783	Maier	Hinz OHG	Goethe Platz 17 13579 Schillern
	⋮	⋮	⋮	⋮

member of **< ssd > f**
strategic software development
factory

Das innovative
Systemhaus



Begriffe der Objektorientierung



member of **< ssd > f**
strategic software development
factory

Das innovative
Systemhaus



Vergleich von Objekt- und Datenorientierung

Eigenschaft	Objektorientierung	Datenorientierung
Struktur	Instanzen von Klassen (= Objekte) besitzen Methoden und Attribute	Speicherung in zweidimensionalen Tabellen (=Relationen)
Datentypen	komplexe Strukturen von Datentypen	nur einfache Datentypen erlaubt
Vererbung	grundlegendes Konstrukt	kein grundsätzliches Konstrukt
Beziehungen	gerichtete Beziehungen	Bildung von Beziehungen über Fremd-/ Primärschlüsselpaar; keine direkte Unterstützung von n:m-Beziehungen
Navigation	Navigation entlang von Beziehungen	direkt keine Navigation möglich
Identität	Objekt besitzt implizit Identität	Identität muß über Primärschlüssel künstlich erzeugt werden
Datenmanipulation	gezielte Manipulation von Instanzdaten	Selektion-Suchen-Ändern-Schreiben-Methodik
Datenkapselung	Datenkapselung über Sichtbarkeit	Daten liegen global vor und können jederzeit manipuliert werden
Operationen	Operationen arbeiten immer auf den Attributen des eigenen Objekts	Datenzugriffe gelten global und können alle Elemente des Datenmodells manipulieren

Grundsätzliche Anforderungen an eine Integration

- hohe Flexibilität des Abbildungsverfahrens
- Verwaltung der Abbildungsverweise im Objektmodell
- zu SQL äquivalente Funktionalität für Datenmanipulation und -navigation
- „native“ Einbettung in objektorientierte Programmiersprachen
- Kompatibilität zu Standards, Middleware-Produkten oder Generatoren
 - ODMG, CORBA, EJB, ...
 - Avantis Unisuite, GFT GRIT Connect, SunSoft JavaBlend, ...
 - TopLink, RogueWave ObjectFactory, ...

member
of **< ssd > f**
strategic software development
factory

Das innovative
Systemhaus



Abbildungsverfahren

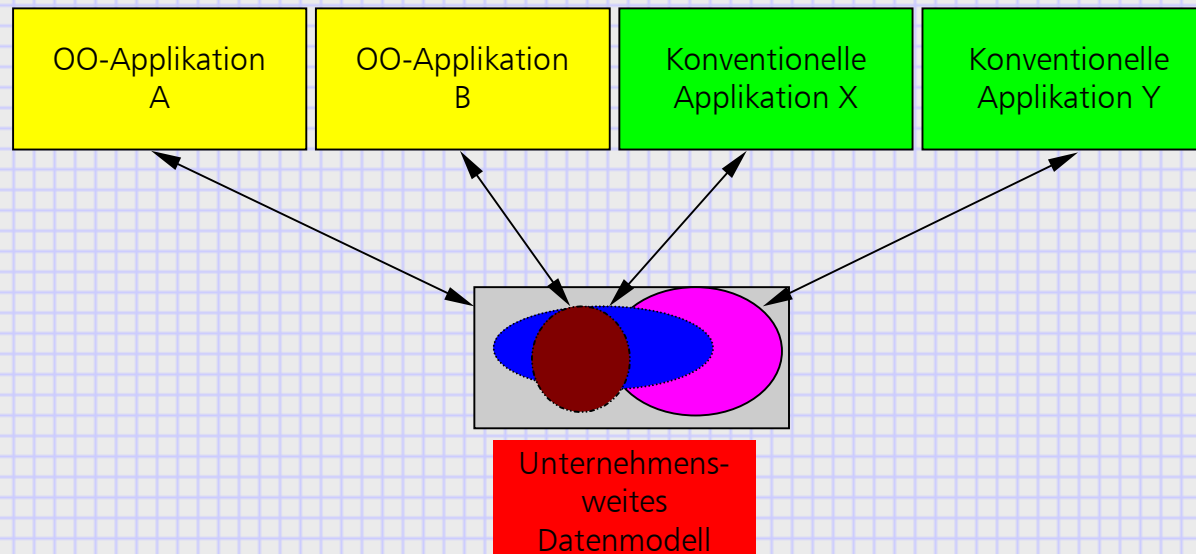
- möglichst hoher Freiraum des Abbildungsverfahrens zwischen den jeweiligen Modellierungskonstrukten
- prinzipiell Verteilung der persistenten Attribute einer Klassen auf Spalten einer oder mehrerer Tabellen
- Erfordernis der Praxis: mehrere Klassen auf eine Tabelle abbilden (z.B. bei existierenden Datenbanken)
- Berücksichtigung des Konzepts der Vererbung
- transparente Abbildung von Assoziationen auf Primär-/ Fremdschlüsselbeziehungen bzw. eigene Beziehungsentitäten

member
of **< ssd > f**
strategic software development
factory

Das innovative
Systemhaus



Abbildungsverweise



- verschiedenste Applikationen arbeiten jeweils auf einem Ausschnitt eines unternehmensweiten Datenmodells
- Verwaltung der Verweise ist in den jeweiligen Applikationsmodellen sinnvoll

member of **< ssd > f**
strategic software development
factory

Das innovative
Systemhaus



Datenmanipulation und -navigation

Funktionalitäten zur Datenmanipulation und -navigation:

- **Create:** INSERT INTO table VALUES ...
- **Read:** SELECT * FROM table WHERE ...
- **Update:** UPDATE table SET row=value WHERE ...
- **Delete:** DELETE FROM table WHERE ...
- **Join:** ... WHERE table1.id==table2.id ...

➡ Anforderungen an die Laufzeitunterstützung

member
of **< ssd > f**
strategic software development
factory

Das innovative
Systemhaus



Einbettung in die Programmiersprache

- embedded SQL
 - standardisierte und ausgereifte Sprache
 - rascher und einfacher Ansatz
 - ⇒ „Fremdkörper“ innerhalb von objektorientierten Sprachen
- ODBC, JDBC
 - weitgehend standardisiert
 - ⇒ zu kompliziert in der Benutzung
 - ⇒ nur für bestimmte Betriebssysteme und/oder Sprachen
- ODMG: OQL, OML, ODL
 - weitgehend standardisiert
 - ⇒ kaum reine Umsetzungen
- produktspezifische Einbettungen
 - meistens basierend auf ODMG-Spezifikation
 - ⇒ produktabhängige Erweiterungen

member
of **< ssd > f**
strategic software development
factory

Das innovative
Systemhaus



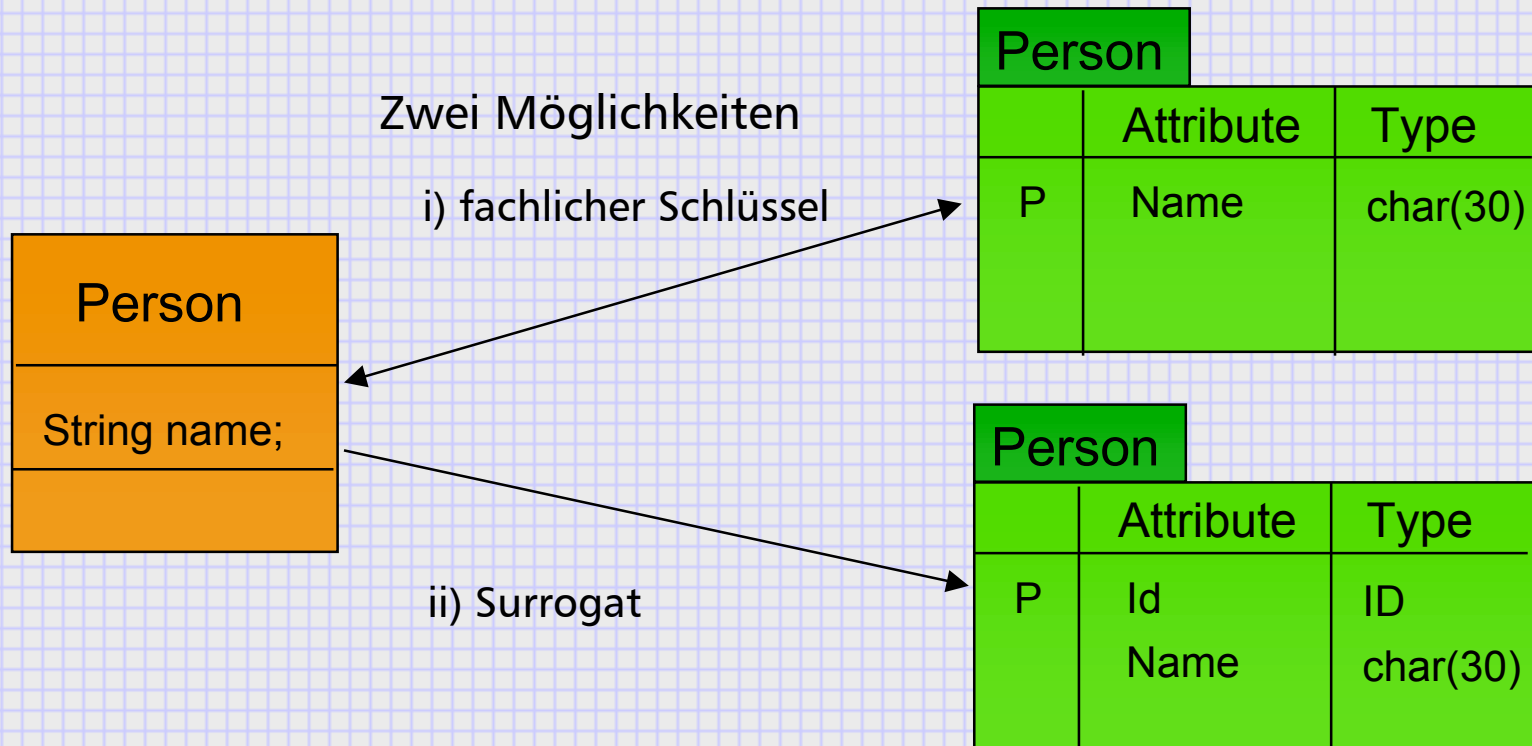
Lösungsansätze

member
of **< ssd > f**
strategic software development
factory

Das innovative
Systemhaus

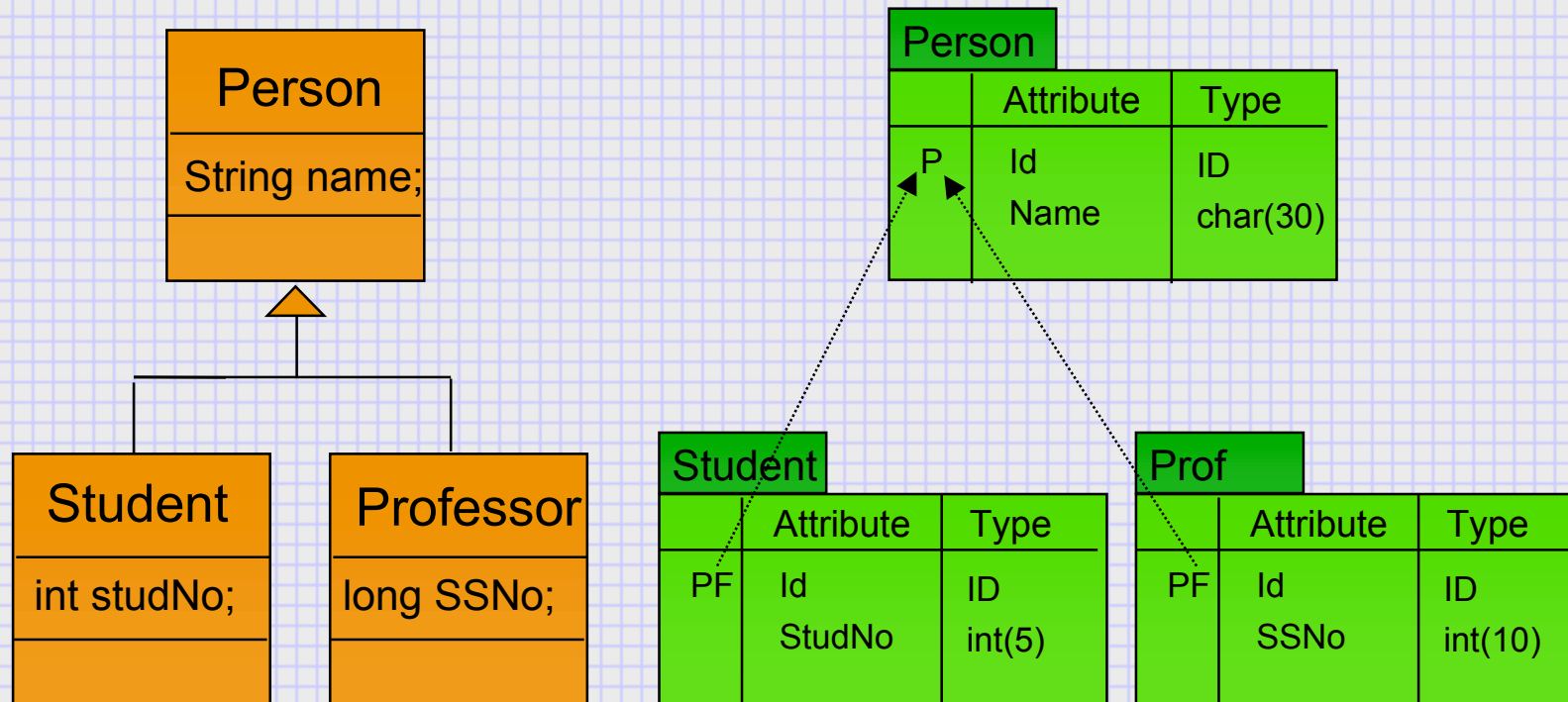


Objektidentität



- ✓ Instanzen besitzen Identität
- ✓ fachliche Primärschlüssel haben im Objektmodell eine Entsprechung
- ✓ Surrogate haben im Objektmodell keine Entsprechung

Abbildung von Generalisierung (1)



Vertikale Partitionierung

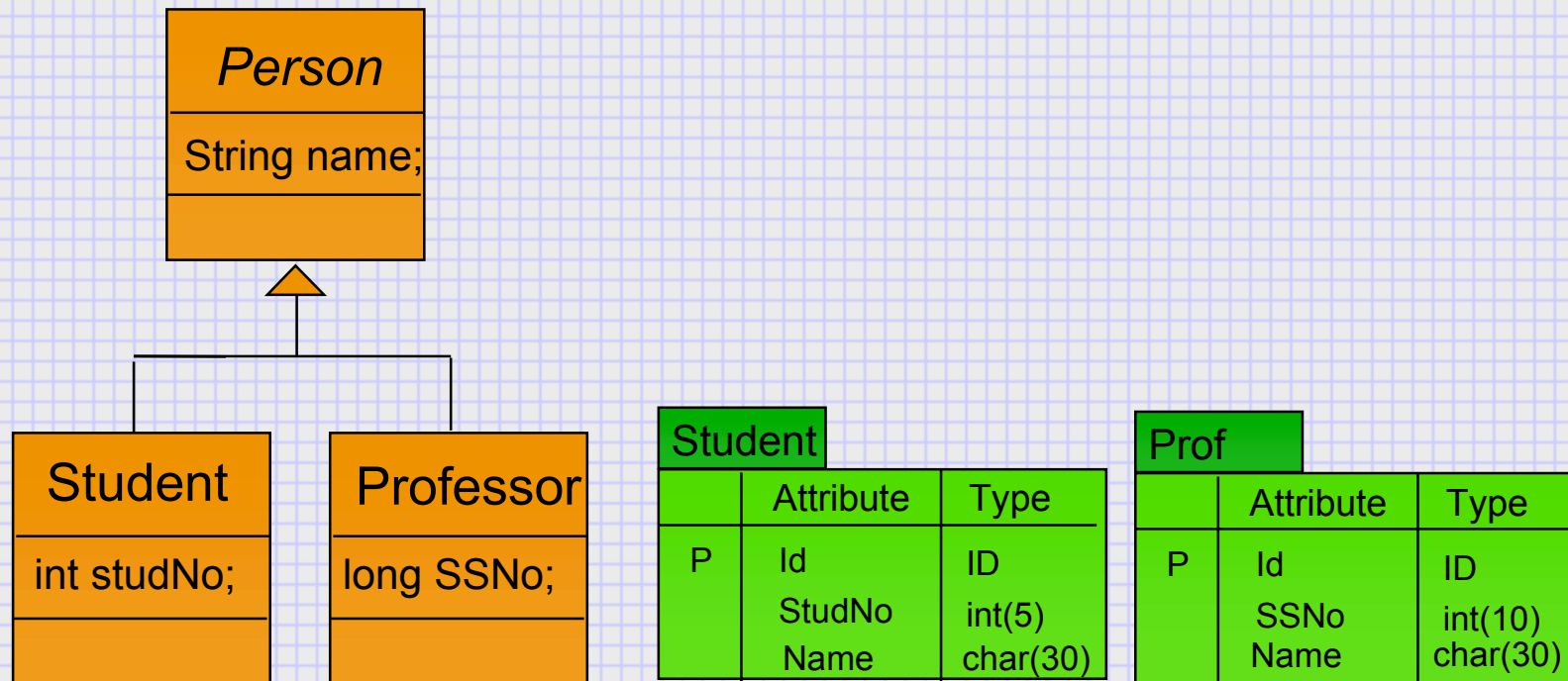
- eine Klasse - eine Tabelle
- durch Fremdschlüssel verknüpft

member of **< ssd > f**
strategic software development
factory

Das innovative
Systemhaus



Abbildung von Generalisierung (2)

*Horizontale Partitionierung*

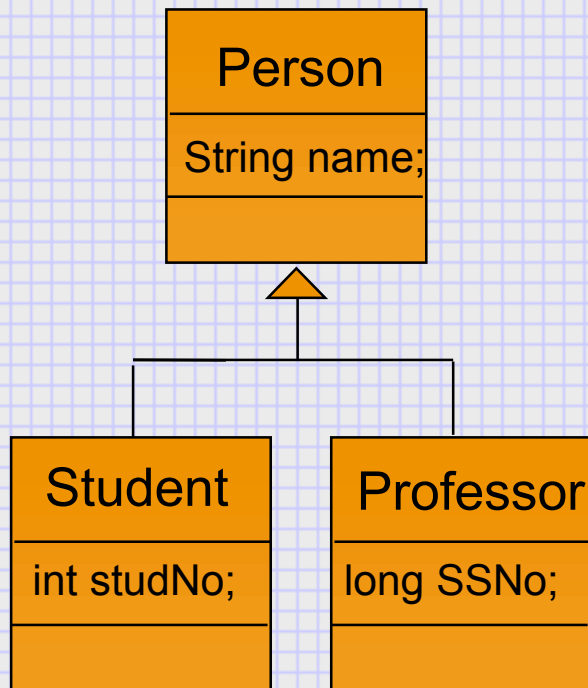
- eine konkrete Klasse - eine Tabelle
- Attribute der Vaterklasse in jede Spezialisierung aufnehmen

member of **< ssd > f**
 strategic software development
 factory

Das innovative
Systemhaus



Abbildung von Generalisierung (3)



Person		
	Attribute	Type
P	Id	ID
	Name	char(30)
	studNo	int(5)
	SSNo	int(10)

Partitionierung nach Typ

- ein Generalisierungsbaum - eine Tabelle
- Attribute jeder Spezialisierung (und evtl. Diskriminator) in die Tabelle der Vaterklasse aufnehmen

member of **< ssd > f**
 strategic software development
 factory

Das innovative
Systemhaus



Übersicht zur Abbildung von Generalisierung

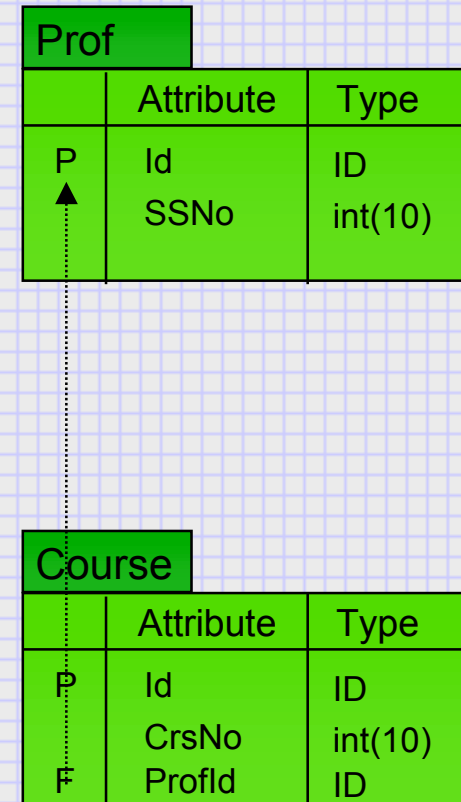
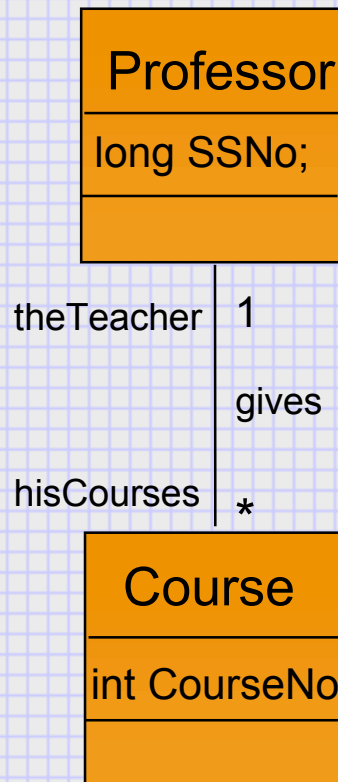
- vertikale Partitionierung
 - guter polymorpher Zugriff
 - ⇒ Schreiben und Lesen erfordert mehrere Zugriffe bzw. Join
 - horizontale Partitionierung
 - einfaches Lesen und Schreiben
 - ⇒ kein unmittelbarer polymorpher Zugriff
 - Partitionierung nach Typ
 - guter polymorpher Zugriff sowie Lesen und Schreiben
 - ⇒ Speicherverschwendung
 - ⇒ evtl. zusätzlicher Diskriminator erforderlich
- ⇒ Art der Partitionierung muß auf jeder Generalisierung individuell entschieden werden
- ⇒ trotz Entscheidung sollte der Bezug zum ursprünglichen Modell erhalten bleiben

member
of **< ssd > f**
strategic software development
factory

Das innovative
Systemhaus



Abbildung von Assoziationen (1)



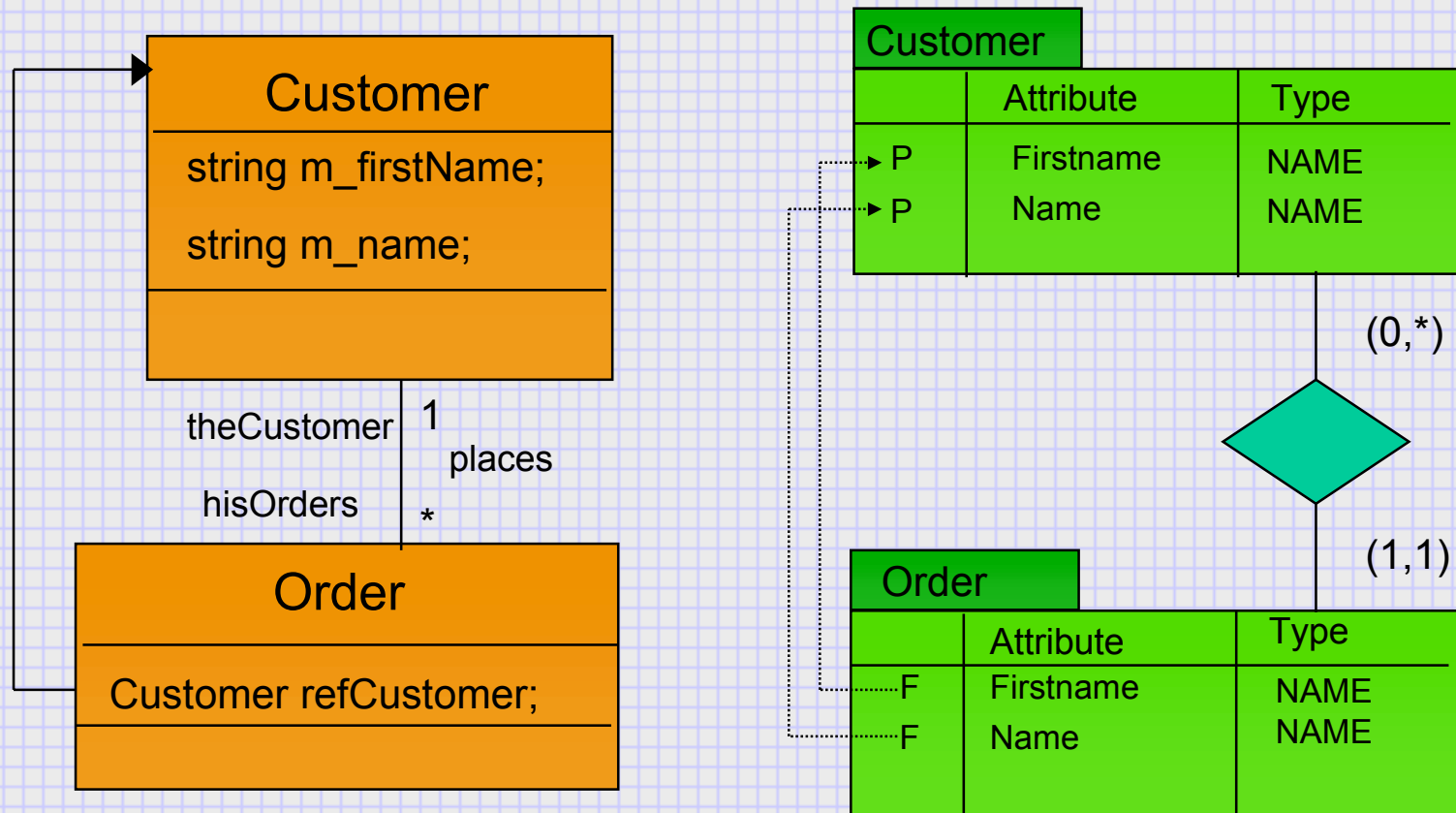
- Implementierung von Fremdschlüsseln in einem Objektmodell ?

member of **< ssd > f**
 strategic software development
 factory

Das innovative
Systemhaus



Abbildung von Assoziationen (2)



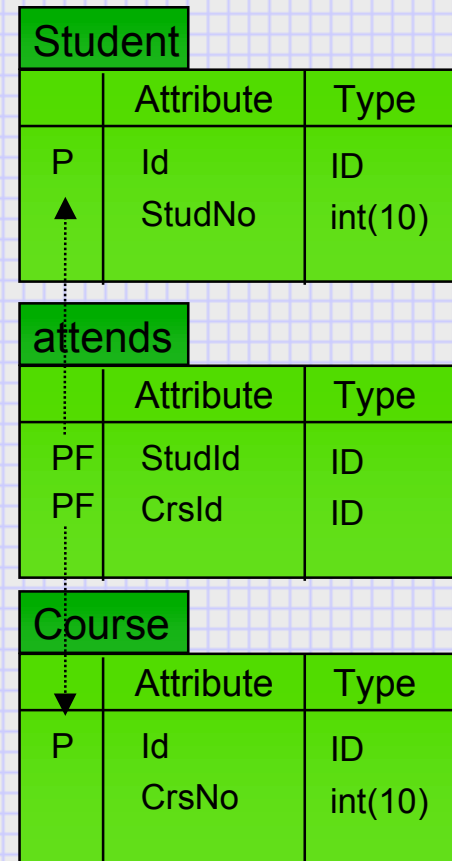
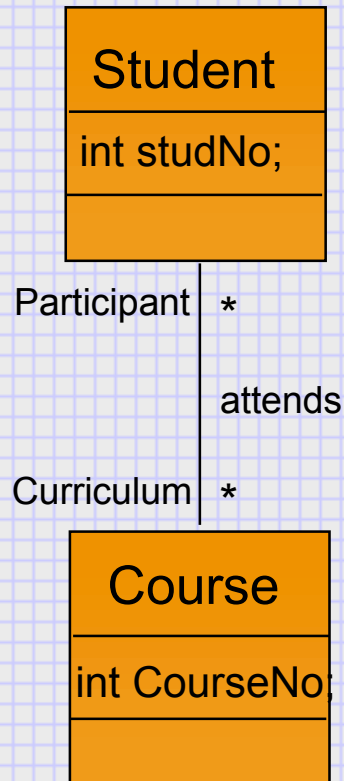
- Objektmodell: Referenzattribut implementiert Assoziation
- Datenmodell: Fremdschlüssel implementiert Beziehung

member of **< ssd > f**
strategic software development
factory

Das innovative
Systemhaus

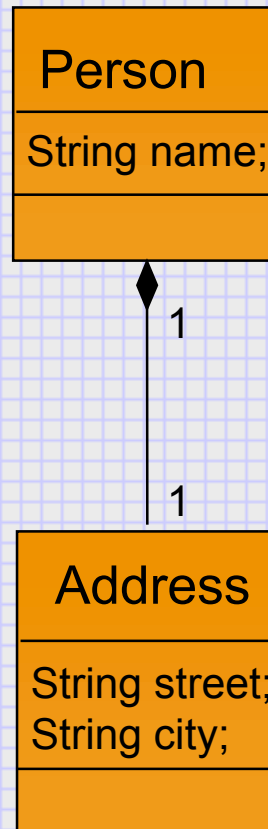


Abbildung von Assoziationen (3)



- (*,*)-Assoziationen müssen als eigene Tabelle implementiert werden
- ansonsten ist die Tabelle optional

Abbildung von Assoziationen (4)



Person		
	Attribute	Type
P	Id	ID
	Name	char(20)
	Street	char(20)
	City	char(20)

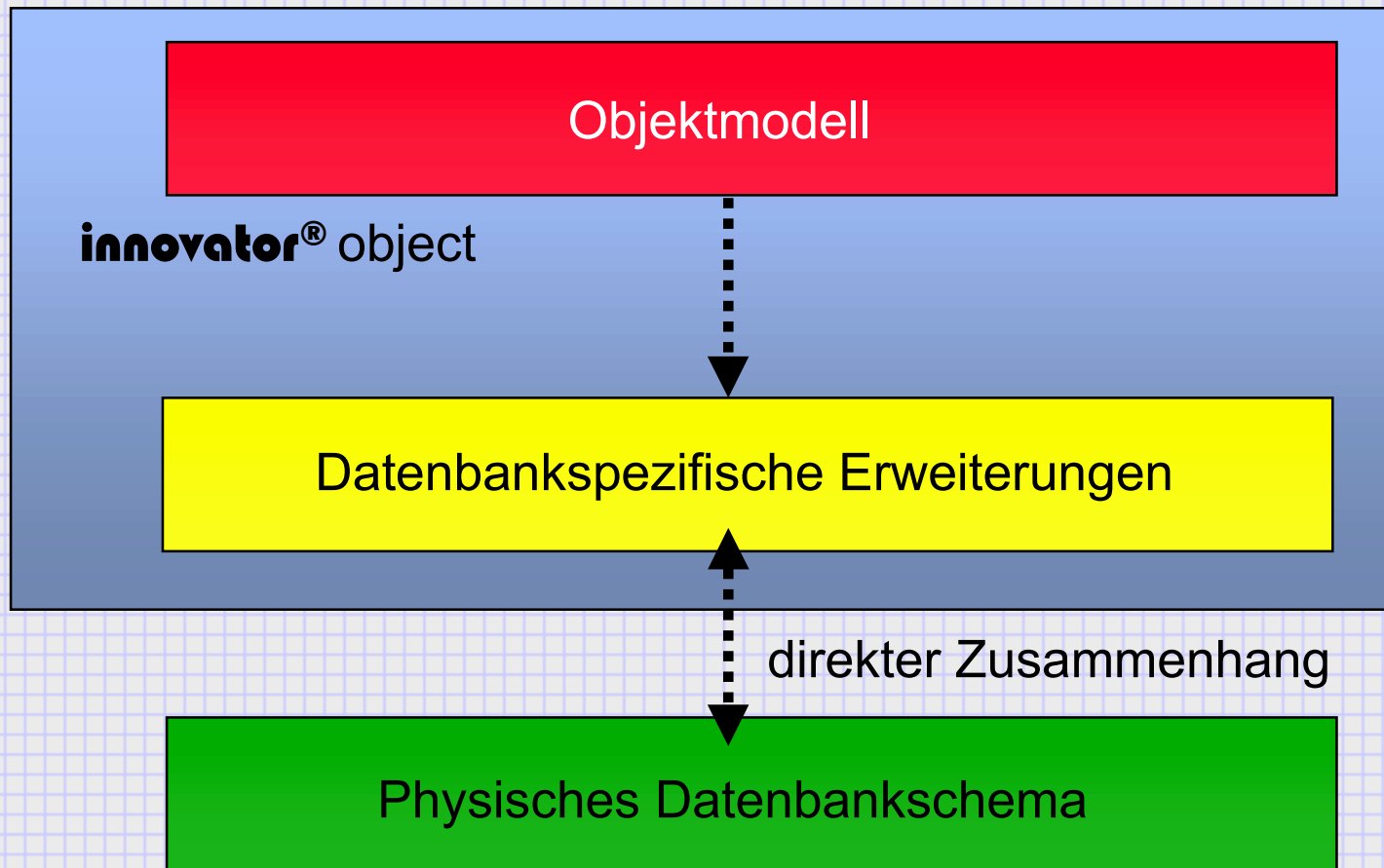
- Komposition: Abbildung in gemeinsame Tabelle

member of **< ssd > f**
 strategic software development
 factory

Das innovative
Systemhaus



Erster Ansatz: Erweiterung des Objektmodells



member of **< ssd > f**
strategic software development
factory

Das innovative
Systemhaus



Notwendige Informationen im Objektmodell

✓ Klassen

- ☆ persistent/transient
- ☆ Schema & Tabellename
- ☆ Darstellung im Objektmodell (Assoziationsklassen, (*,*)-Assoziationen)
- ☆ Partitionierungsart bei Generalisierungen

✓ Attribute

- ☆ Darstellung im Objektmodell (technische/fachliche Primärattribute)
- ☆ Tabellen und Spaltennamen
- ☆ Schlüsseleigenschaft, Datenbanktypen

✓ Assoziationen (Fremdschlüssel)

- ☆ Darstellung im Objektmodell
- ☆ Tabellen- und Spaltenname
- ☆ Schlüsseleigenschaft
- ☆ referenziertes Primärattribut

✓ datenbankspezifische Annotationen

member of **< ssd > f**
strategic software development
factory

Das innovative
Systemhaus



Bewertung des ersten Lösungsansatzes

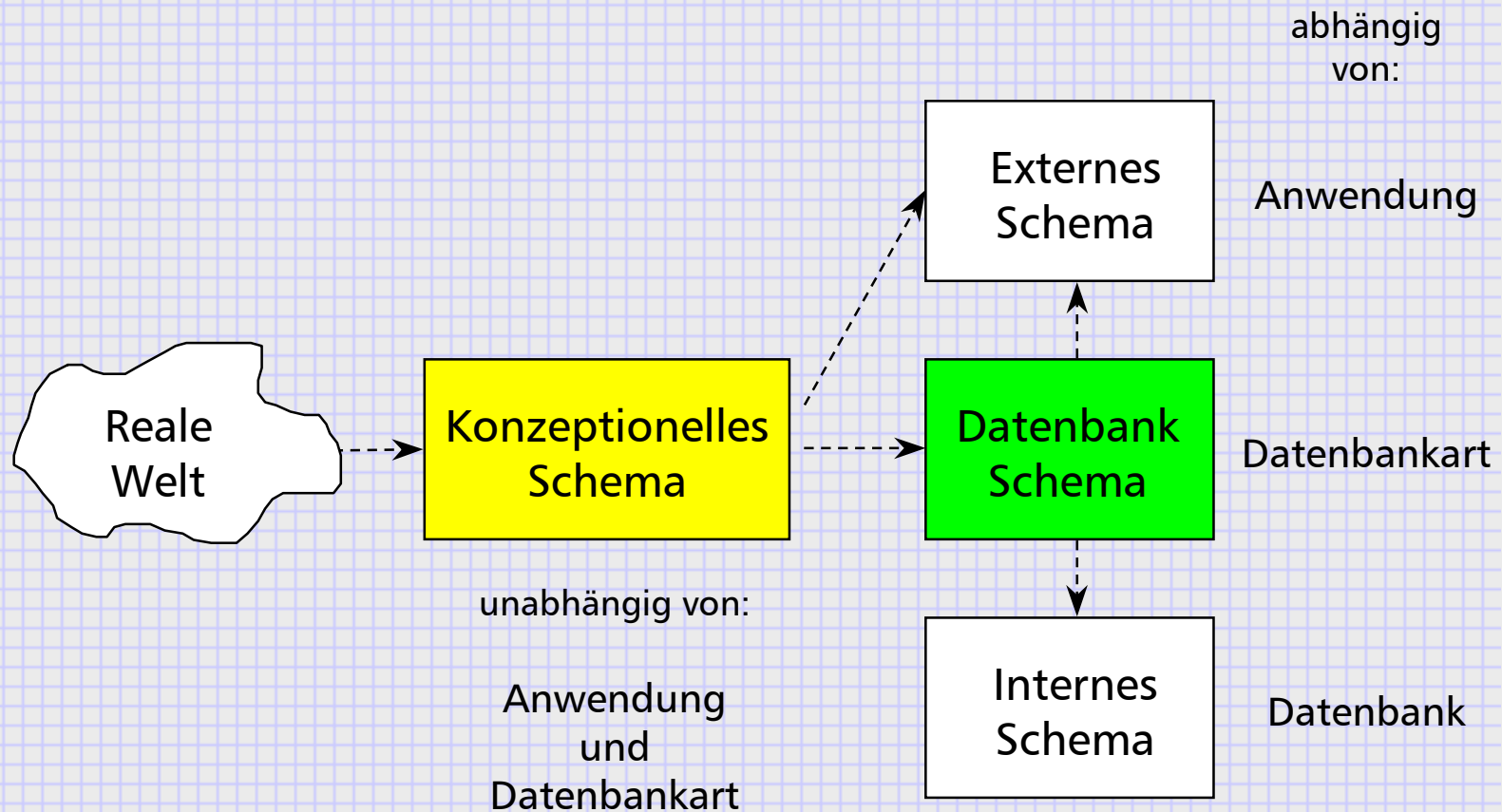
- 😊 Abbildungsinformation kann im Prinzip erfaßt werden
- 😊 Lösung ist schnell implementierbar
- 😞 methodischer Mix von Objektmodellierung, Datenmodellierung und Datenbankdesign im Objektmodell
- 😞 manuelle Pflege erforderlich
- ➔ neue Idee: Nutzung der Konzepte der Datenmodellierung
- ➔ saubere Trennung von Objekt- und Datenmodellierung
- ➔ Drei-Ebenen-Architektur trennt zwischen konzeptionellem Schema und physischem Datenbank-Schema

member
of **< ssd > f**
strategic software development
factory

Das innovative
Systemhaus



Drei-Ebenen-Architektur (nach ANSI/SPARC)



Denormalisierung und physisches Datenbank-Design

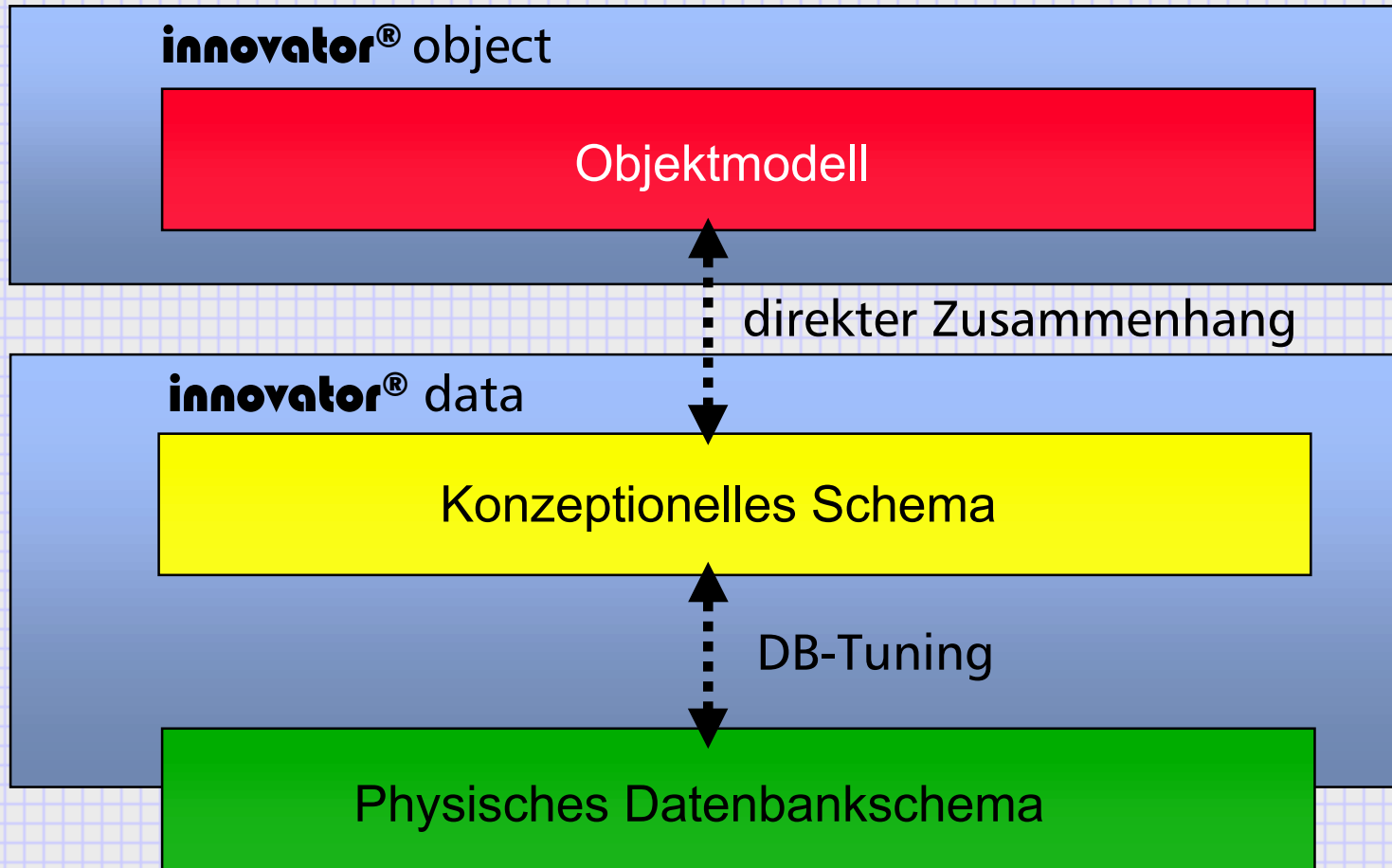
- ✓ physisches DB-Schema basiert auf konzeptionellem Schema
- ✓ Entkopplung von DB-Schema und konzeptionellem Schema
 - ☆ Entitäten/Tabellen
 - ✱ Zusammenfassung über (0,1)-Beziehungen
 - ✱ Zusammenfassung über (0,*)-Beziehungen (Wiederholungsgruppen)
 - ✱ Zusammenfassung über Generalisierung und Spezialisierungen
 - ☆ Attribute/Spalten
 - ✱ technische Spalten (Redundanz)
 - ✱ Abbildung von mehreren Attributen auf dieselbe DB-Spalte
 - ✱ Vergabe von Indizes
- ✓ automatische Konsistenzpflege
- ✓ Generierung von DDL-Skripten

member
of **< ssd > f**
strategic software development
factory

Das innovative
Systemhaus



Zweiter Ansatz: innovator® data object-Integration

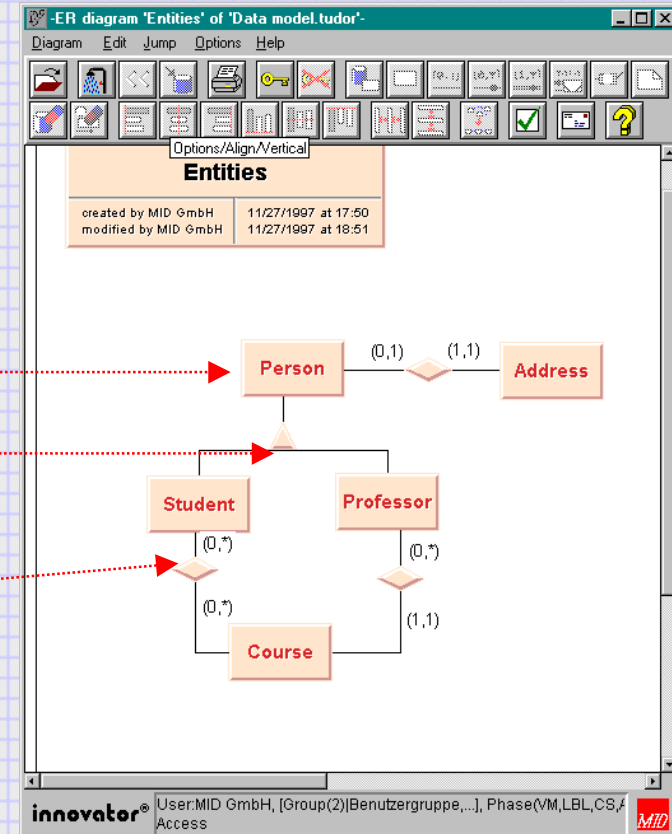
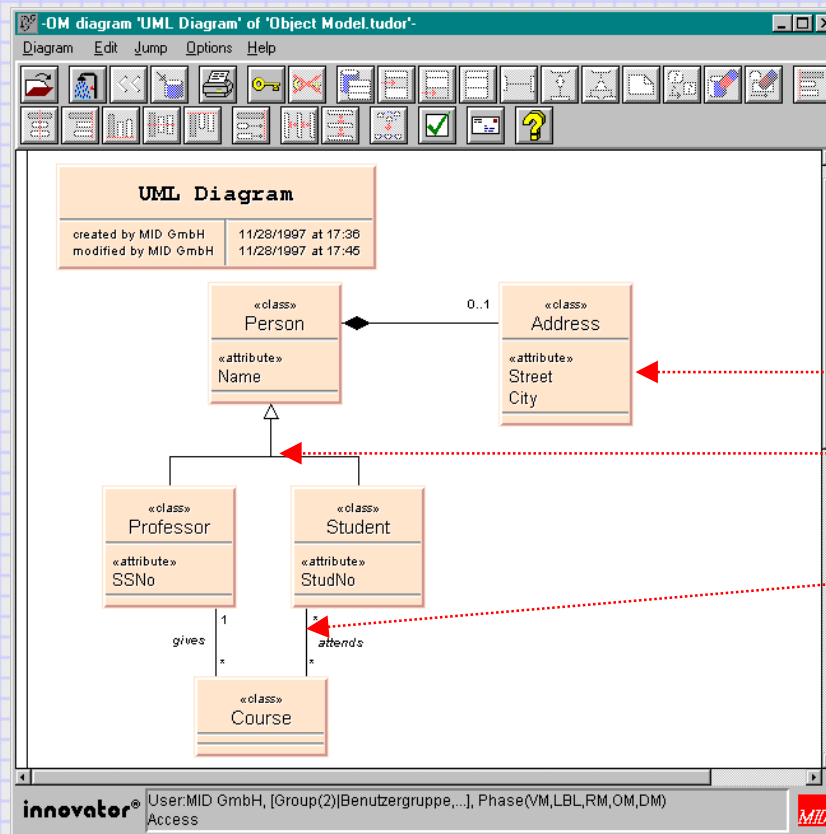


member of **< ssd > f**
strategic software development
factory

Das innovative
Systemhaus



Objektmodell und konzeptionelles Datenmodell



member of **< ssd > f**
strategic software development
factory

Das innovative
Systemhaus



Methodenspezifische Trennung

- ✓ DB-Details wie Schlüssel, DB-Typen, Indizes etc. nur in der Datenmodellierung
- ✓ keine DB-spezifischen Attribute im Objektmodell
- ✓ Abbildung auf konzeptioneller Ebene sehr einfach:
 - ✦ persistente Klasse \Leftrightarrow Entität
 - ✦ persistentes Attribut \Leftrightarrow Attribut
 - ✦ Assoziation \Leftrightarrow Beziehung
 - ✦ Generalisierung \Leftrightarrow Kategorie
- ✓ DB-Tuning nur in der Datenmodellierung

member
of **< ssd > f**
strategic software development
factory

Das innovative
Systemhaus



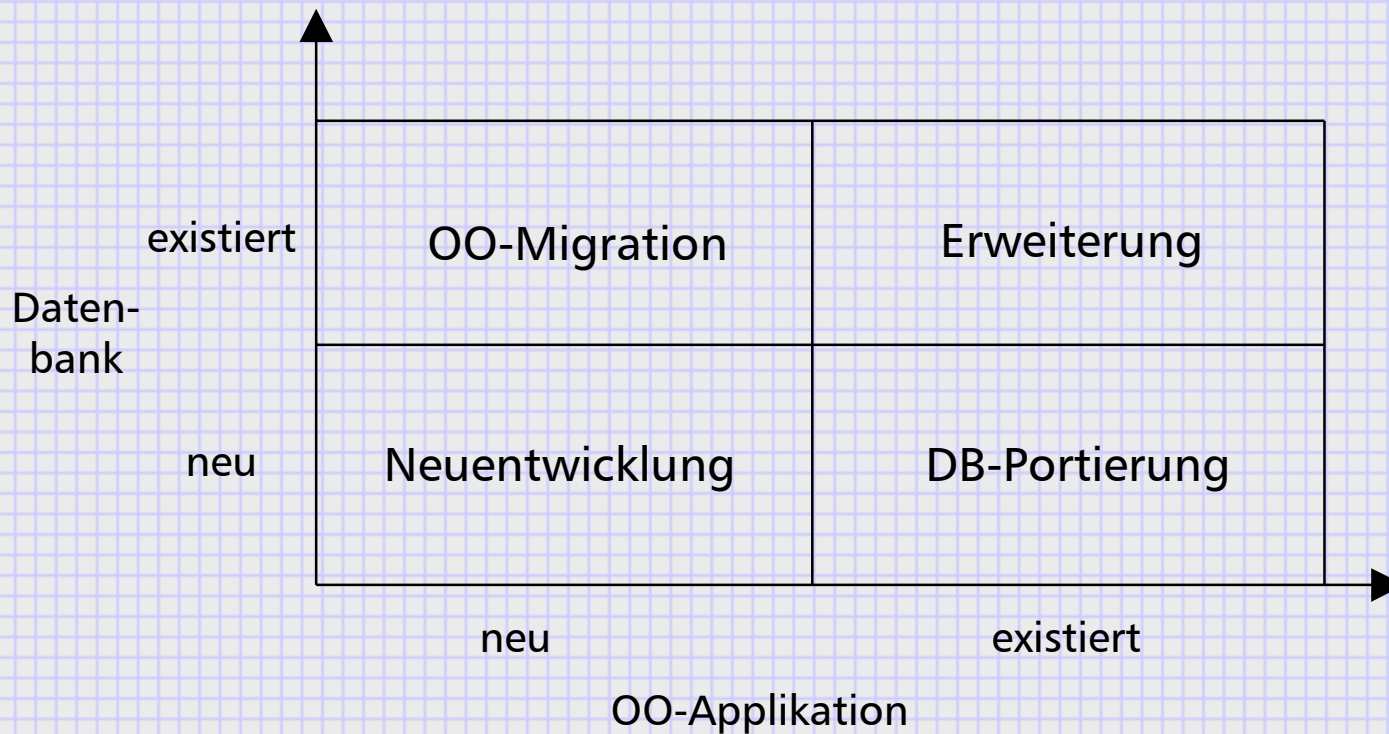
Vorgehensweisen

member
of **< ssd > f**
strategic software development
factory

Das innovative
Systemhaus



Szenarien

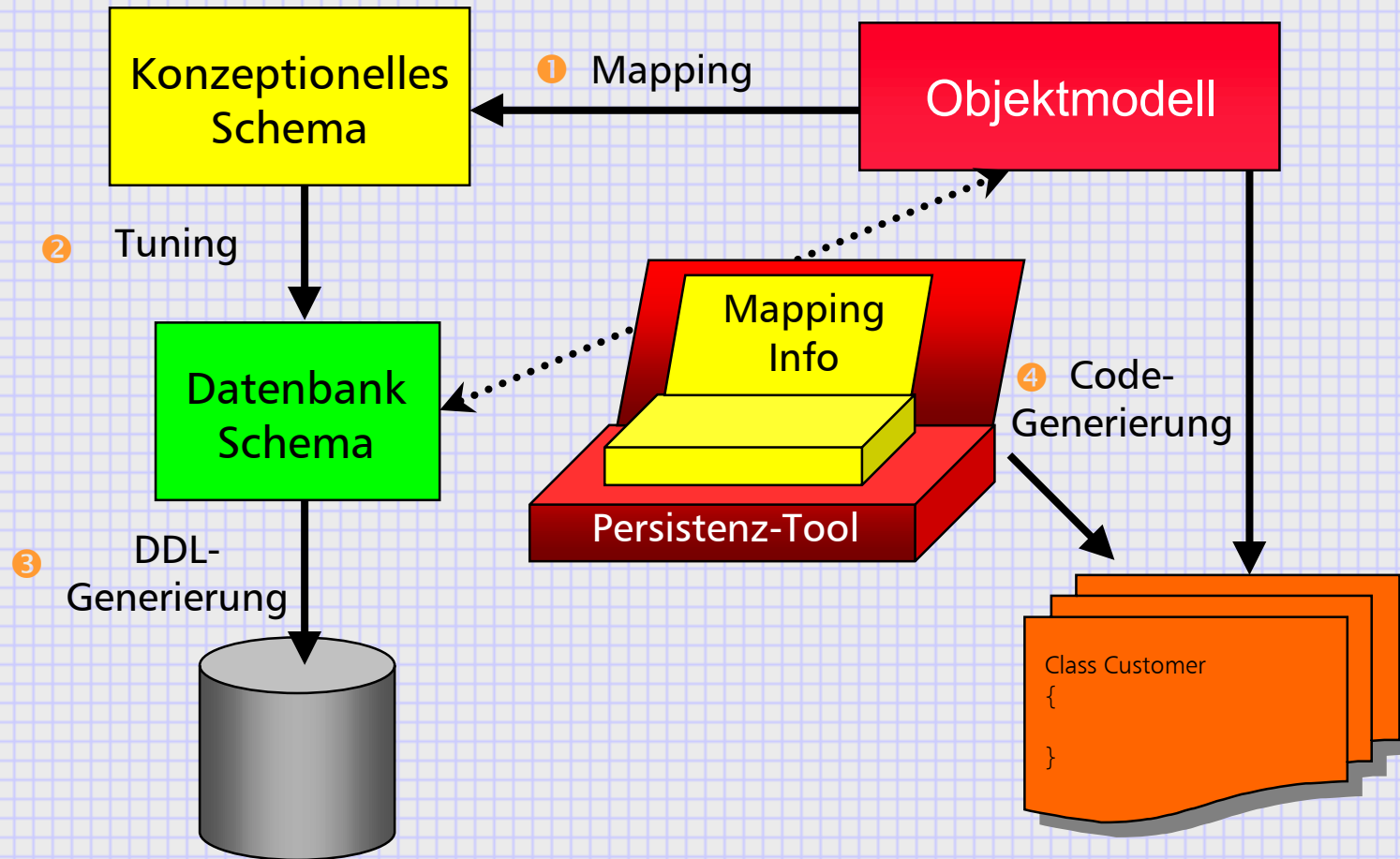


member of **< ssd > f**
of **strategic software development**
factory

Das innovative
Systemhaus



Neuentwicklungsszenario

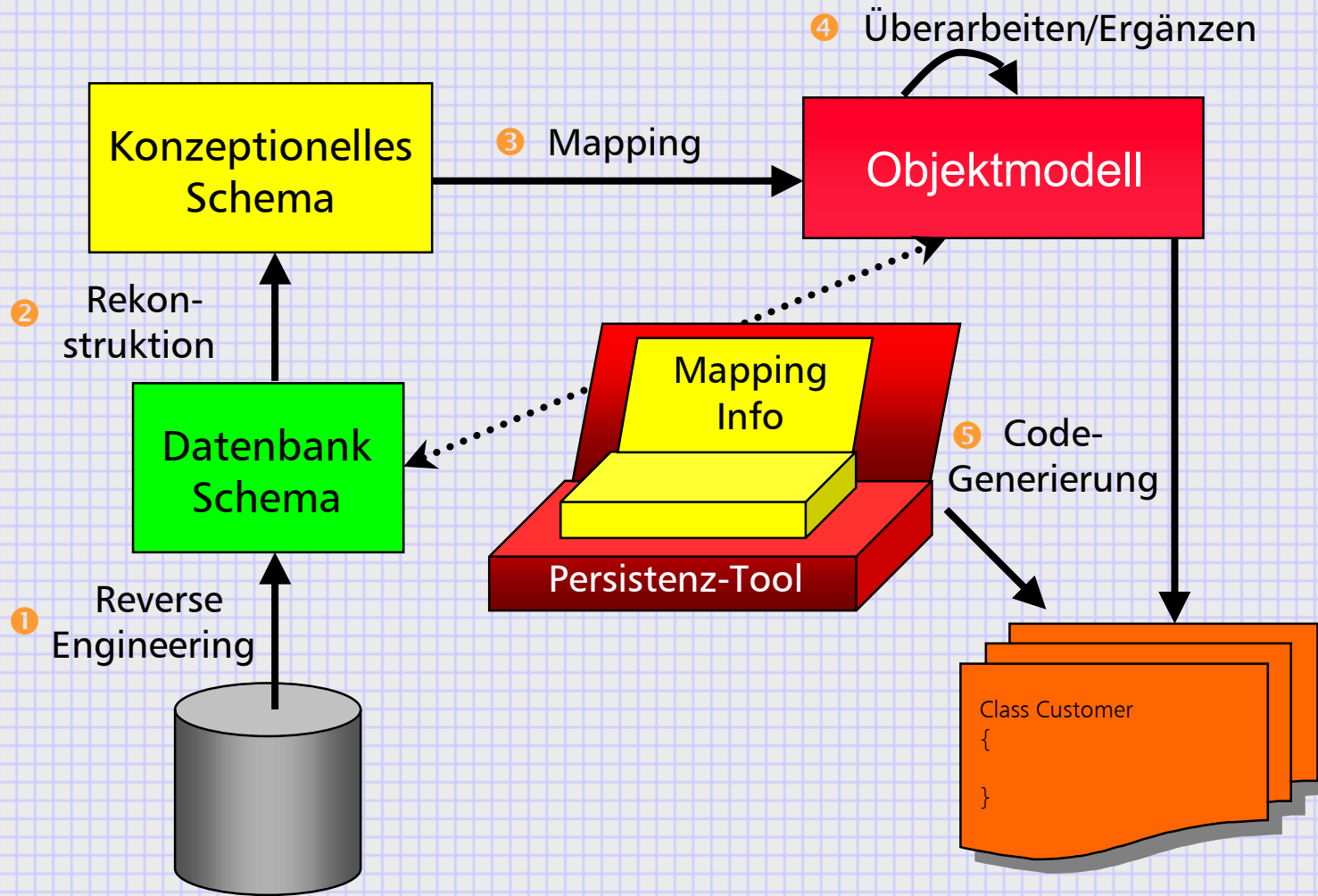


member of **< ssd > f**
strategic software development
factory

Das innovative
Systemhaus



OO-Migrationsszenario

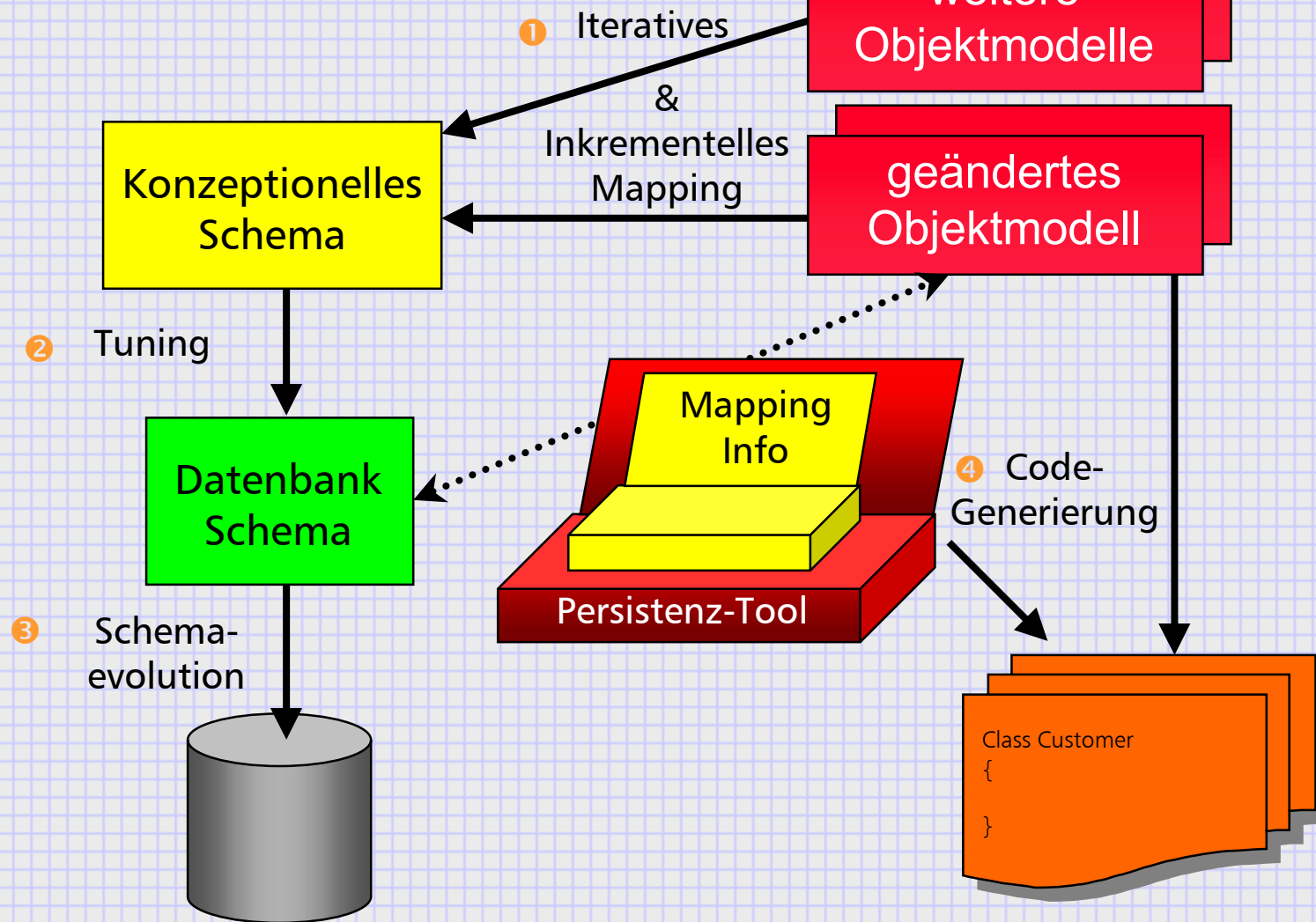


member of **< ssd > f**
strategic software development
factory

Das innovative
Systemhaus



Erweiterungsszenario

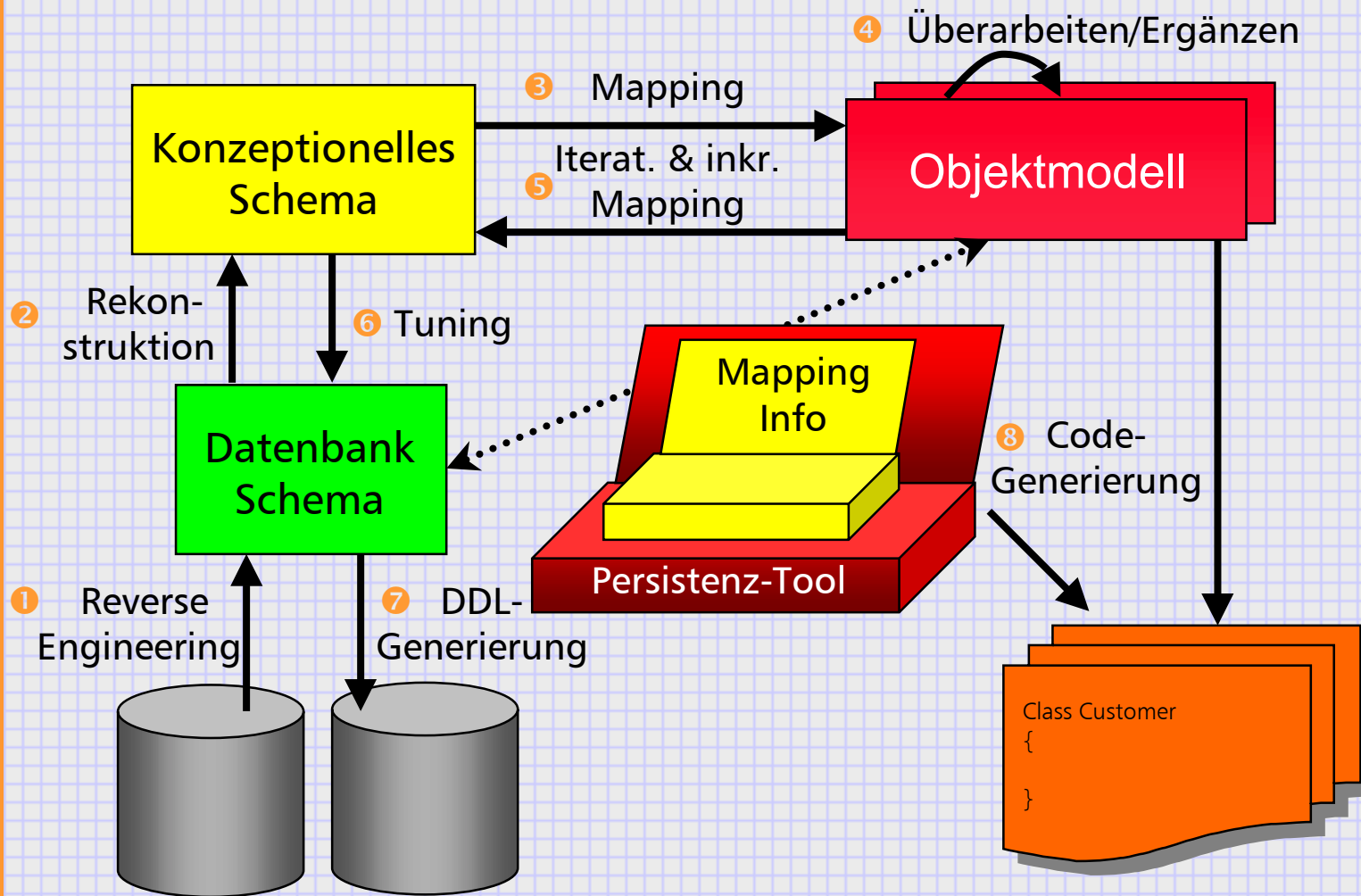


member of **< ssd > f**
strategic software development
factory

Das innovative
Systemhaus



DB-Portierungsszenario



member of **< ssd > f**
strategic software development
factory

Das innovative
Systemhaus



Leistungsmerkmale für eine Integration

- ✓ Bidirektionales Mapping
 - ☆ Erzeugen eines Objektmodells aus einem konzeptionellen Schema und umgekehrt
- ✓ Iteratives & inkrementelles Mapping
 - ☆ Festhalten der Zusammenhänge unabhängig von Bezeichnern
- ✓ Aktualisieren der Abhängigkeiten
 - ☆ Umbenennen, Erzeugen und Löschen von Entitäten/Klassen und Attributen
 - ☆ Umbenennen, Erzeugen und Löschen von Beziehungen/Assoziationen
- ✓ Konsolidierungsunterstützung
- ✓ Abbildungsinformationen für Persistenz-Tools

member
of **< ssd > f**
strategic software development
factory

Das innovative
Systemhaus



</Object-Relational Mapping:>

Fragen?

member
of **< ssd > f**
strategic software development
factory

Das innovative
Systemhaus

