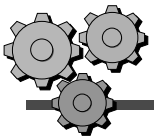


Persistenz

Workplace Solutions

- ➔ RDBMS und OO
- ➔ Strukturkonflikt
- ➔ Object-RDBMS-Mapping

APCON Workplace Solutions

Abbildung Objekte auf RDBMS

Workplace Solutions

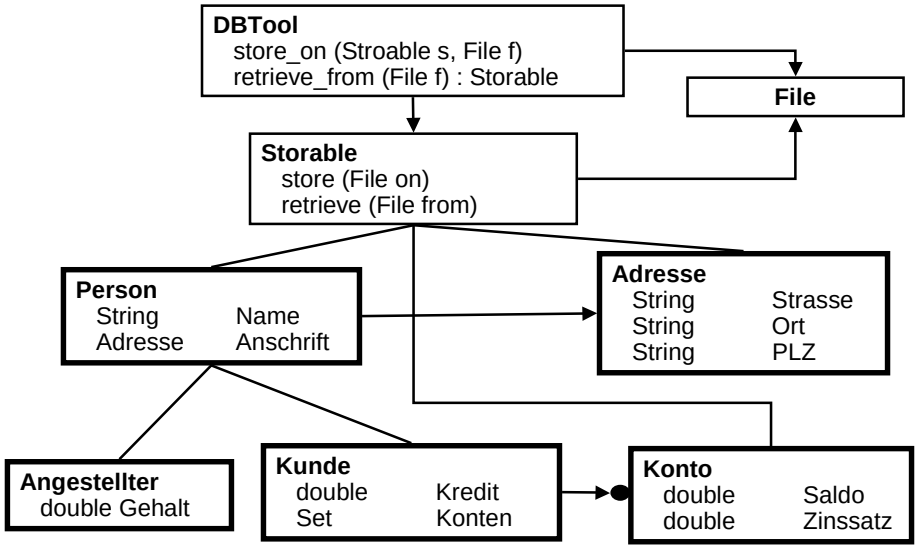
Der Strukturkonflikt
- Basisklassen und Domänen

Klassen zur Kapselung der relationalen Datenbank Abbildung
- der benutzt Beziehung
- der erbt-Beziehung

Datenbanken als Dienstleistung

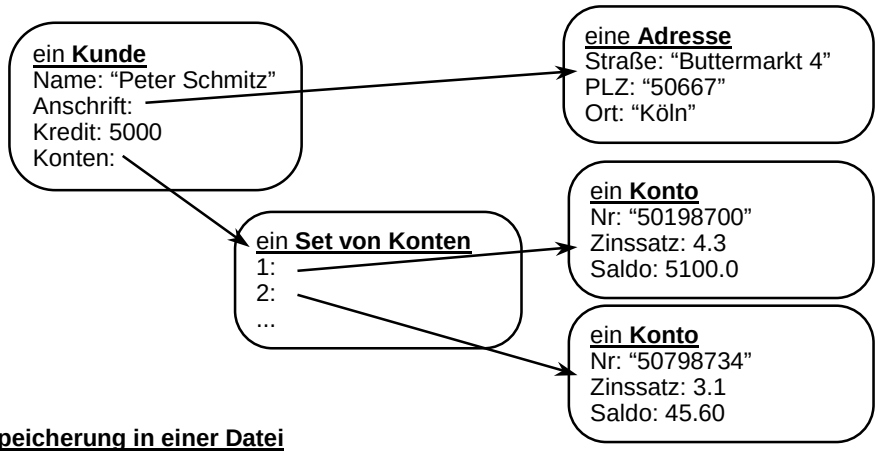
Speicherbare Objekte

Workplace Solutions



... als Objektdiagramm

Workplace Solutions



Speicherung in einer Datei

```
Kunde #1( “Peter Schmitz”, Adresse #2( “Buttermarkt 4”, “Köln”, “50667” ), 5000,
          Set #3( Konto #4( “50198700”, 5100.0),
                  Konto #5(“50798734”, 45.60)    ) )
```

... als Datenbanktabellen

Workplace Solutions

Kunde

Nr	Name	Anschrift	Kredit
1396	Peter Schmitz	1	5000
3415	Maria Otten	2	8000

Adresse

Nr	Straße	PLZ	Ort
1	Buttermarkt 4	50667	Köln
2	Lichhof	50676	Köln

Konto

Nr	Zinssatz	Saldo	Inhaber
50198700	4.3	5100.0	1396
50798734	3.1	45.60	1396
50787802	3.1	3210.4	3415

SQL-Abfrage:

```
select
  ko.Saldo, ku.Name, a.Strasse, a.PLZ, a.Ort
from
  Kunde ku, Konto ko, Adresse a
where
  ku.Nr = ko.Inhaber and
  ku.Anschrift = a.Nr and
  ko.Saldo >= 2000
```

Strukturkonflikt

Workplace Solutions

Objekt	Tupel / Record / Zeile
Attribut	Attribut / Feld / Spalte
Klasse	Relation / Tabelle
Basisdatentyp / Fachwert	Domäne / Attributtyp
Systemweite Identität	Primärschlüssel
Objekt enthält Objekte	Tupel enthält Attribute
Funktionen kapseln Daten	globaler Zugriff
Schnittstelle	Repräsentation

Ansätze zur Verbindung einer objektorientierten Anwendung mit einer relationalen Datenbank

- Objektorientierte Rekonstruktion der Datenbank-Elemente
- Transformation / Anpassung von Objektstrukturen

Basisklassen und Domänen

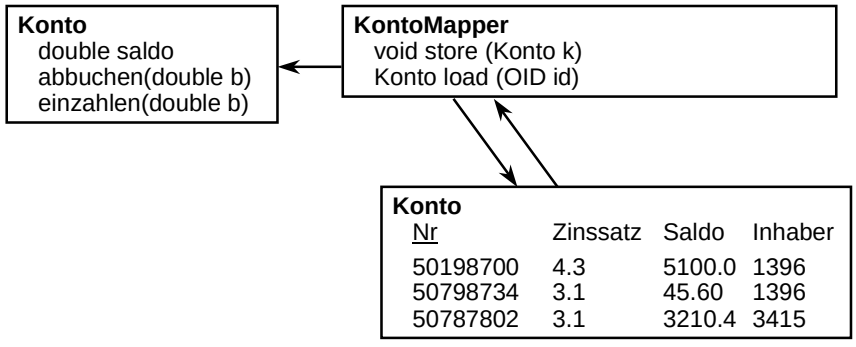
Workplace Solutions

- Vom Datenbanksystem sind einige Domänen vorgegeben, die Wertepräsentation und Wertemengen für Felder definieren , z.B.
  ➔ INTEGER, NUMERIC, MONEY, FLOAT, DATE, CHAR
- In Java sind einige Basisdatentypen vorgegeben, die Wertepräsentation definieren, z.B.
  ➔ Integer, Float, Double, Character, Boolean, Byte
- Unter Verwendung von Basisdatentypen kann man weitere Klassen definieren, die Wertepräsentationen festlegen.
- Durch die Klasse wird sichergestellt, daß nur erlaubte Werte existieren.
- Ein Objekt enthält Attribute mit Werten und Attribute mit Referenzen auf andere Objekte.
- Einige der Attribute werden in korrespondieren Feldern des Tupels dargestellt, welches das Objekt in der Datenbank repräsentiert.
- Die Domänen-Integrität wird durch das Objekt selbst aufrecht erhalten und bei der Konvertierung geprüft.

Konvertierung und Speicherung

Workplace Solutions

Zu jeder Klasse wird eine weitere Mapper-Klasse definiert, die die Konversion von Java-Werten in Domänen-Werte definiert und auch die Zuordnung zu Feldnamen vornimmt.



Objektidentität

Workplace Solutions

Identität der Objekte ist ihre Referenz, in der Datenbank aber eine eigene Objektidentität nötig (Primärschlüssel)

- Identität durch eigenes Identitätsattribut
- Klasse OID stellt eindeutige Identitäten dar

Konto
double saldo
OID id
abbuchen(double b)
einzahlen(double b)

Tabellen mit OIDs

Workplace Solutions

OID ist Primärschlüssel in den Datenbanktabellen

Adresse

OID	Strasse	PLZ	Ort
10	Buttermarkt 4	50667	Köln
11	Lichhof 2	50676	Köln
12	Ahrstraße 45	53175	Bonn
13	Broich 137	53937	Schleiden

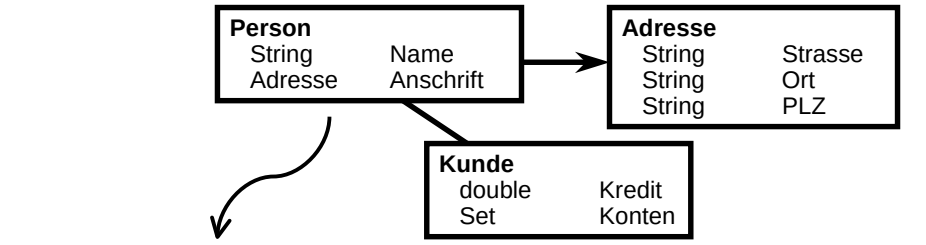
Angestellter

OID	Name	Gehalt
3	Willi Wacker	4300
4	Rafael Raab	3400

Bemerkung: Alle nicht-Schlüssel-Attribute sind nur von der OID abhängig (eine Bedingung für die 2. Normalform)

1:1 Beziehungen: Fremdschlüssel

Workplace Solutions



Darstellung mit Fremdschlüssel

Kunde

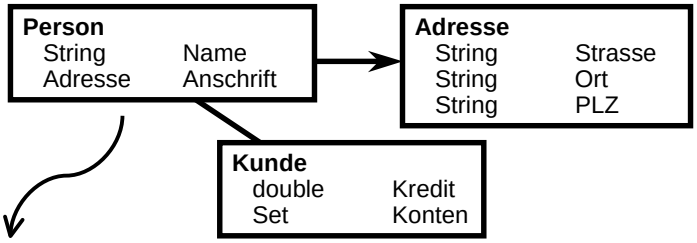
OID	Name	Anschrift	Kredit
1	Peter Schmitz	10	5000
2	Maria Otten	11	8000

Adresse

OID	Straße	PLZ	Ort
10	Buttermarkt 4	50667	Köln
11	Lichhof 2	50676	Köln

1:1 Beziehungen: Expandierter Klasse

Workplace Solutions



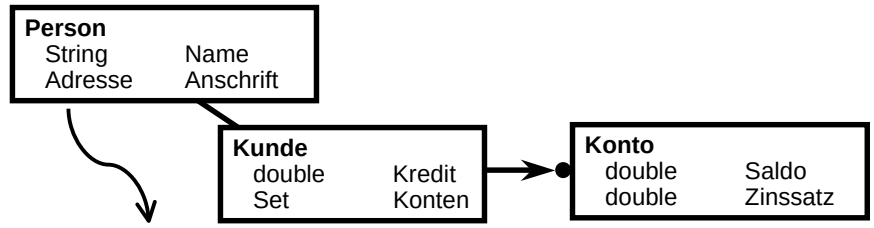
Kunde

OID	Name	Straße	PLZ	Ort	Kredit
1	Peter Schmitz	Buttermarkt 4	50667	Köln	5000
2	Maria Otten	Lichhof 2	50676	Köln	8000

Der Zugriff Objekts wird effizienter, aber
- der Begriff “Adresse” geht verloren.
- Normalisierung wird zurückgenommen.

1:N-Beziehungen: Beziehungstabelle

Workplace Solutions



Darstellung mit Beziehungstabelle

Kunde

OID	Name	Anschrift	Kredit
1	Peter Schmitz	10	5000
2	Maria Otten	11	8000

Konto

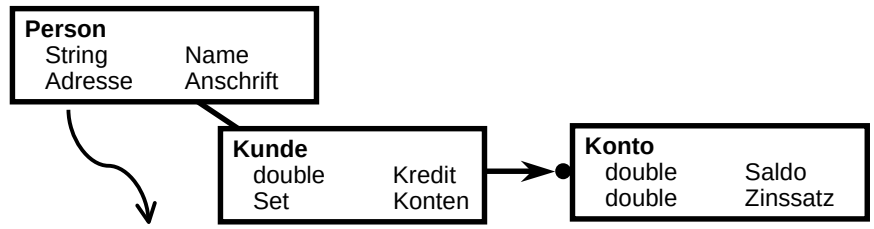
OID	Zinssatz	Saldo
21	4.3	5100.0
22	3.1	45.60
23	3.1	3210.4

Kunde_Konto

Kunde	Konto
1	21
1	22
2	23

1:N-Beziehungen: Fremdschlüssel

Workplace Solutions



Darstellung mit Fremdschlüssel

Kunde

OID	Name	Anschrift	Kredit
1	Peter Schmitz	10	5000
2	Maria Otten	11	8000

Konto

OID	Zinssatz	Saldo	Inhaber
21	4.3	5100.0	1
22	3.1	45.60	1
23	3.1	3210.4	2

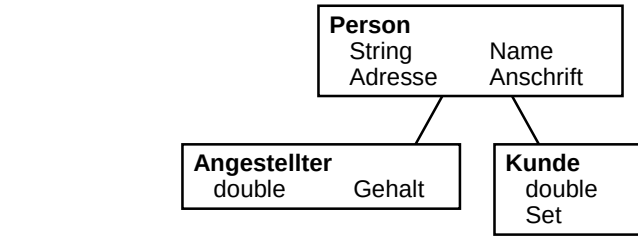
Objekt-Beziehungen

Workplace Solutions

Kardinalität \ Darstellung	expandierte Klasse	Fremd-schlüssel	separate Tabelle
Kunde —1:1— Adresse	**effizient** 2.. Normalform	**flexibel**	**ineffizient**
Kunde —1:n— Konto	effizient nicht normalisiert	**effizient**	**flexibel**
Angestellter —m:n— Projekt	nicht anwendbar	nicht anwendbar	**effizient**

Vererbung: Horizontal abgebildet

Workplace Solutions



Angestellter

OID	Name	Anschrift	Gehalt
3	Willi Wacker	12	4300
4	Rafael Raab	13	34(

Kunde

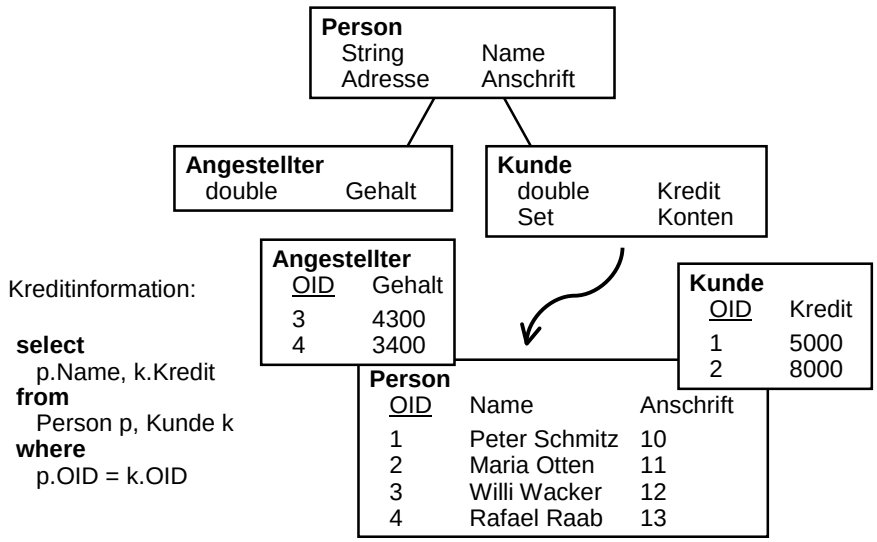
OID	Name	Anschrift	Kredit
1	Peter Schmitz	10	5000
2	Maria Otten	11	8000

Kreditinformation:

select Name, Kredit
from Kunde

Vererbung: Vertikal abgebildet

Workplace Solutions

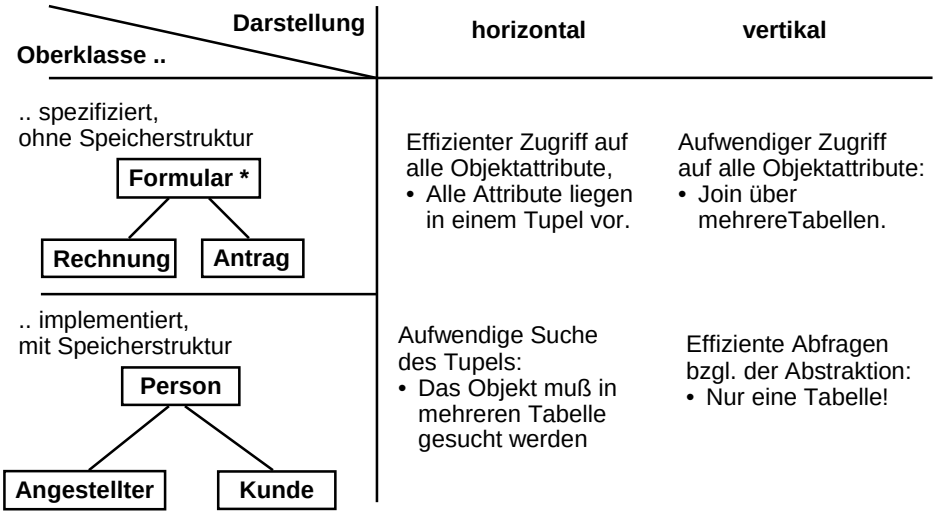


Kreditinformation:

```
select
  p.Name, k.Kredit
from
  Person p, Kunde k
where
  p.OID = k.OID
```

Vertikal vs. Horizontal

Workplace Solutions



Zusammenfassung

Workplace Solutions

- Objektorientierte Systeme i.a. kompatibel zur 2. oder 3. Normalform
- Integritätsbedingungen im Objekt-Modell berücksichtigt und gekapselt. Vergabe von Objektidentitäten “OID” vereinfacht das Modell und die Handhabung.
- Effizient konvertier- und zugreifbar sind einfache, flache Objekte, die als ein Tupel mit atomaren Feldwerten in einer Tabelle repräsentiert werden können.
- Die relationale Datenbank wird zur Dienstleistungskomponente für das objektorientierte System.
- Flexibilität und Tiefe der Objektstrukturen müssen mit Effizienz abgewogen werden. Der Durchsatz kann mit gezieltem Verzicht auf Normalisierung und mit cache-Techniken optimiert werden.

Mapping-Probleme

Workplace Solutions

Objektorientiert: Person enthält Konten

- In Person ist ein Behälter definiert, der Konten enthält

Mapping 1:N

- In der RDBMS umgekehrter Verweis: In Tabelle Konten Fremdschlüssel zur Tabelle Person

Deshalb:

- Der Konto-Mapper muß um ein Attribut erweitert werden (OID für Fremdschlüssel Person)
- Auslesen von Konten nach Person-OID

Alternative: Zusätzliche Mapping-Tabelle

Modellierungsprobleme

Workplace Solutions

In einer Tabelle werden Objekte in ihre Attribute zerlegt.

- Laden/Speichern von Objekten nur durch Zusammensetzen/ Auseinandernehmen der Attribute
- Jedes enthaltene Objekt muß extra modelliert werden (eventuell Tabelle anlegen)

Aber:

- Zugriff auf Objekte über OID, oder
- über einige Attribute
- Nicht für jedes enthaltene Objekt Modellierung sinnvoll

BLOBs

Workplace Solutions

Lösung:

- Objekte werden als serialisierte Datenpakete in die Datenbank geschrieben (Binary Large Objects)
- Dazu Binary-Datenbanktyp verwenden
- Nur ausgewählte Attribute bleiben sichtbar, mindestens die OIDs