

# Konzepte der Datenbanktechnologie

Prof. Dr. U. Hoffmann  
FH Wedel

Data Base for Objects (DB4O)

**Data Base for  
Objects (DB4O)**

Identitätskonzept

Integrität

Integrität

Persistenz

Transaktionen

Zusammenfassung

- ▶ Datenbanken
- ▶ Objekt–Orientierte Programmierung

Offene Frage:

Wie können wir objektorientiert modellieren und dennoch eine Datenbank benutzen?

## Data Base for Objects (DB4O)

Identitätskonzept

Integrität

Integrität

Persistenz

Transaktionen

Zusammenfassung

## Identitätskonzept

## Integrität

## Integrität

## Persistenz

## Transaktionen

### Data Base for Objects (DB4O)

Identitätskonzept

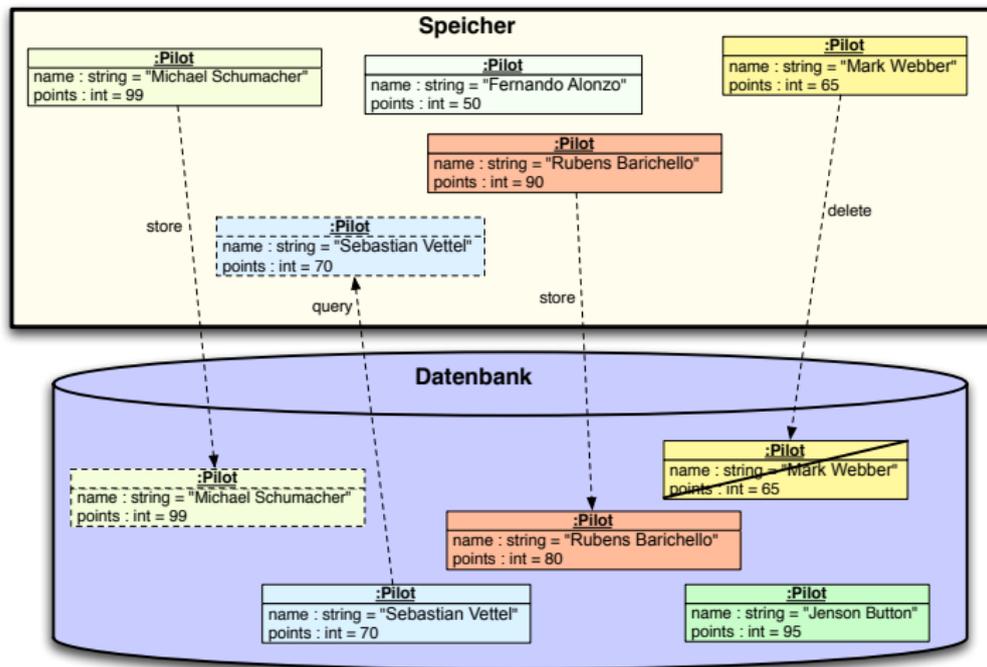
Integrität

Integrität

Persistenz

Transaktionen

Zusammenfassung



Objekte stehen im Speicher und in der Datenbank

Saison2000MS:Pilot
name : string = "Michael Schumacher"
points : int = 99

≠

Saison2001MS:Pilot
name : string = "Michael Schumacher"
points : int = 99

- ▶ Objekte sind unterschiedlich, auch wenn sie gleiche Attribut-Werte haben.
- ▶ In Deutsch: das *selbe* und das *gleiche*
- ▶ In Java: `==` und `equals`
- ▶ Im Speicher: Identität durch mehrfache Repräsentation
- ▶ muss bei persistenten Objekten erhalten bleiben
- ▶ verteilte Objekte: Identität explizit machen
- ▶ DB4O: Objekt-Identifizierer wird beim Anlegen vergeben  
`db.ext.getID()`

## Data Base for Objects (DB4O)

### Identitätskonzept

Integrität

Integrität

Persistenz

Transaktionen

Zusammenfassung

Pilot
name : string
points : string
increasePoints(int)

## Beispiel (Pilot)

```
public class Pilot {  
    private String m_name;  
    private int m_points;  
  
    ...  
  
    public void increasePoints(final int points) {  
        m_points += points;  
    }  
  
    ...  
}
```

- ▶ Die Datenmodellierung erfolgt objekt-orientiert
- ▶ Die Datendefinition erfolgt in Java als POJOs

Pilot
name : string
points : string
increasePoints(int)

## Beispiel (Pilot)

```
Public Class Pilot
    Private m_name As String
    Private m_points As Integer

    ...
    Public Sub IncreasePoints(value As Integer)
        m_points += value
    End Sub

    ...
End Class
```

- ▶ Die Datenmodellierung erfolgt objekt-orientiert
- ▶ Die Datendefinition erfolgt in Visual Basic als Objekte

### Data Base for Objects (DB4O)

#### Identitätskonzept

Integrität

Integrität

Persistenz

Transaktionen

Zusammenfassung

1. Beispiel-Anfragen (*Query by Example*, QBE)  
Anfragen werden durch Beispiel-Objekte gestellt.
2. native Anfragen (*Native Queries*)  
Anfragen werden typischer durch ein Stück Java-Programm gestellt.
3. SODA-Anfragen (*Simple Object Database Access*)  
Anfragen werden durch explizites Konstruieren eines Anfragebaums erzeugt.

## Beispiel (alle Fahrer mit 100 Punkten abfragen)

```
List <Pilot> pilots = db.query(new Predicate<Pilot>() {  
    public boolean match(Pilot pilot) {  
        return pilot.getPoints() == 100;  
    }  
});
```

- ▶ Es werden nicht alle Daten aktiviert.
- ▶ Es erfolgt eine Bytecode-Analyse des Prädikats.
- ▶ Ist es hinreichend einfach, wird eine SODA-Anfrage konstruiert und die Datenbank liefert nur die gewünschten Objekte.

Quelle: DB4O-Tutorial

## Beispiel (alle Fahrer mit 100 Punkten abfragen)

```
Dim pilots As IList(Of Pilot) =  
    db.query(Of Pilot) (  
        Function (p As Pilot) pilot.points = 100  
    )
```

- ▶ Es werden nicht alle Daten aktiviert.
- ▶ Es erfolgt eine Bytecode-Analyse des Prädikats.
- ▶ Ist es hinreichend einfach, wird eine SODA-Anfrage konstruiert und die Datenbank liefert nur die gewünschten Objekte.

## Beispiel (alle Fahrer abfragen, SODA-Anfrage)

```
Query query=db.query();  
query.constrain(Pilot.class);  
ObjectSet result=query.execute();  
listResult(result);
```



- ▶ Durch `constrain`-Aufrufe wird der Abfragebaum konstruiert.

Quelle: DB4O-Tutorial

## Beispiel (alle Fahrer abfragen, SODA-Anfrage)

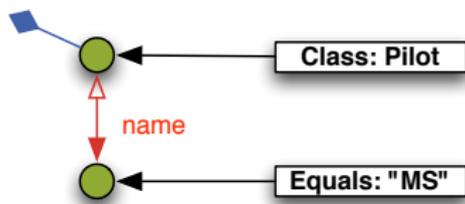
```
Dim query As Query.IQuery = db.Query()  
query.Constrain(GetType(Pilot))  
Dim result As IObjectSet = query.Execute  
ListResult(result)
```



- ▶ Durch `constrain`-Aufrufe wird der Abfragebaum konstruiert.

## Beispiel (Fahrer durch Namen abfragen, SODA-Anfrage)

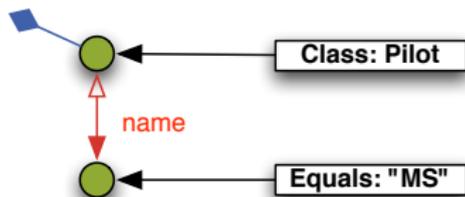
```
Query query=db.query();  
query.constrain(Pilot.class);  
query.descend("name").constrain("Michael Schumacher");  
ObjectSet result=query.execute();  
listResult(result);
```



- ▶ Mit `descend`-Aufrufen in Unterstrukturen absteigen.

## Beispiel (Fahrer durch Namen abfragen, SODA-Anfrage)

```
Dim query As Query.IQuery = db.Query()  
query.Constrain(GetType(Pilot))  
query.Descend("name").Constrain("Michael Schumacher")  
Dim result As IObjectSet = query.Execute  
ListResult(result)
```



- ▶ Mit `descend`-Aufrufen in Unterstrukturen absteigen.

## Beispiel (Disjunktionen, SODA-Anfrage)

```
Query query=db.query();
query.constrain(Pilot.class);
Constraint constr=query.descend("name")
    .constrain("Michael Schumacher");
query.descend("points")
    .constrain(new Integer(99)).or(constr);
ObjectSet result=query.execute();
listResult(result);
```

- Constraint-Objekte können vielfältig verknüpft und ergänzt werden.

not and or greater smaller equal like  
startswith endswith siehe DB4O-Referenz

Quelle: DB4O-Tutorial

## Beispiel (Disjunktionen, SODA-Anfrage)

```
Dim query As Query.IQuery = db.Query()  
query.Constrain(GetType(Pilot))  
Dim constr As Query.IConstraint = query.Descend("name")  
    .Constrain("Michael Schumacher")  
query.Descend("points")  
    .Constrain(99).Or(constr)  
Dim result As IObjectSet = query.Execute  
ListResult(result)
```

- ▶ Constraint-Objekte können vielfältig verknüpft und ergänzt werden.

not and or greater smaller equal like

startswith endswith

siehe DB4O-Referenz

## Beispiel (sortierte Antwort, SODA-Anfrage)

```
Query query=db.query();  
query.constrain(Pilot.class);  
query.descend("name").orderAscending();  
ObjectSet result=query.execute();  
listResult(result);  
query.descend("name").orderDescending();  
result=query.execute();  
listResult(result);
```

- ▶ Sortierte Ausgabe durch ergänzen des `Constraint`-Objekts des passenden Attributs

Quelle: DB4O-Tutorial

## Beispiel (sortierte Antwort, SODA-Anfrage)

```
Dim query As Query.IQuery = db.Query()  
query.Constrain(GetType(Pilot))  
query.Descend("name").OrderAscending()  
Dim result As IObjectSet = query.Execute  
ListResult(result)  
query.Descend("name").OrderDescending()  
result = query.Execute  
ListResult(result)
```

- ▶ Sortierte Ausgabe durch ergänzen des `Constraint`-Objekts des passenden Attributs

## Beispiel (Integritätsbedingungen)

```
public void increasePoints(final int points)
    throws RegulationException {

    if (points > 10) {
        throw new RegulationException("Too many new points");
    } else {
        m_points += m_points;
    }
}
```

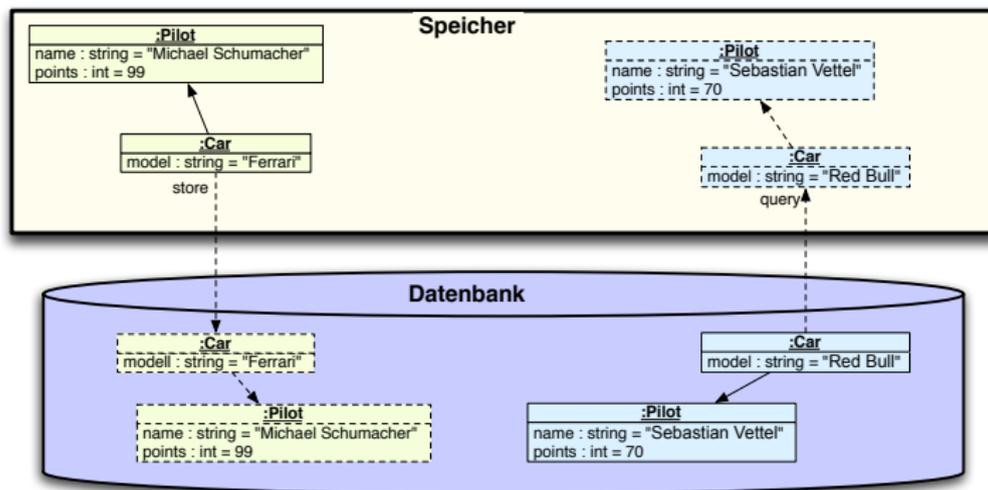
- ▶ Integritäten werden algorithmisch im Objekt-Modell beschrieben.

## Beispiel (Integritätsbedingungen)

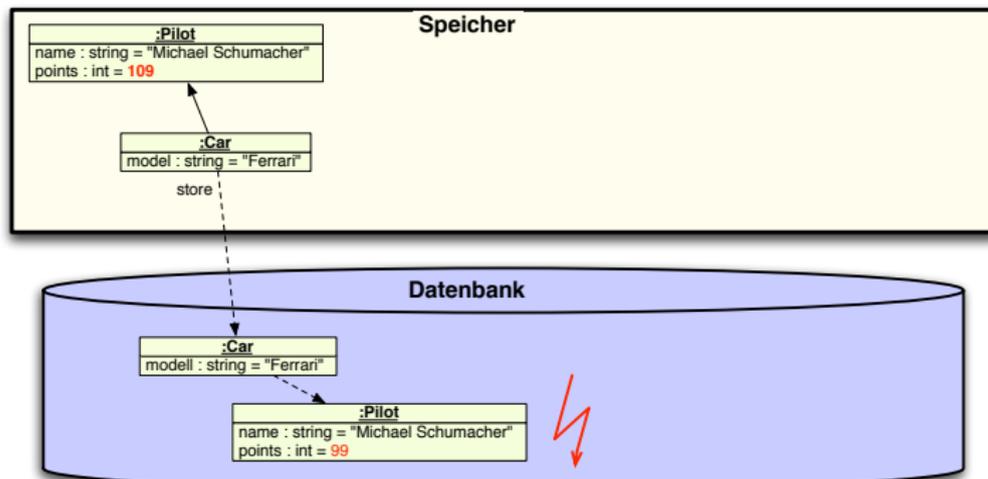
```
Public Sub IncreasePoints(value As Integer)
    If m_points > 10 Then
        Throw New System.Exception("Too many new points")
    Else
        m_points += value;
    End If
End Sub
```

- ▶ Integritäten werden algorithmisch im Objekt-Modell beschrieben.

- ▶ Wie entscheidet man, welche Objekte persistent werden?
- ▶ In DB4O durch verwenden von `store` oder `query` auf ganz normalen Java-Objekten (*Plain Old Java Object, POJO*)
- ▶ Wie werden Objektgeflechte persistent gemacht?



- ▶ Aktualisieren von Objektgeflechten



- ▶ Aktualisierungstiefe (*update depth*)
- ▶ Standardwert: 1 (schreibt nur primitive und String-Attribute)
- ▶ 

```
conf = Db4oEmbedded.newConfiguration();  
conf.common().objectClass(Klasse)  
                .cascadeOnUpdate(true)
```

## Data Base for Objects (DB4O)

Identitätskonzept

Integrität

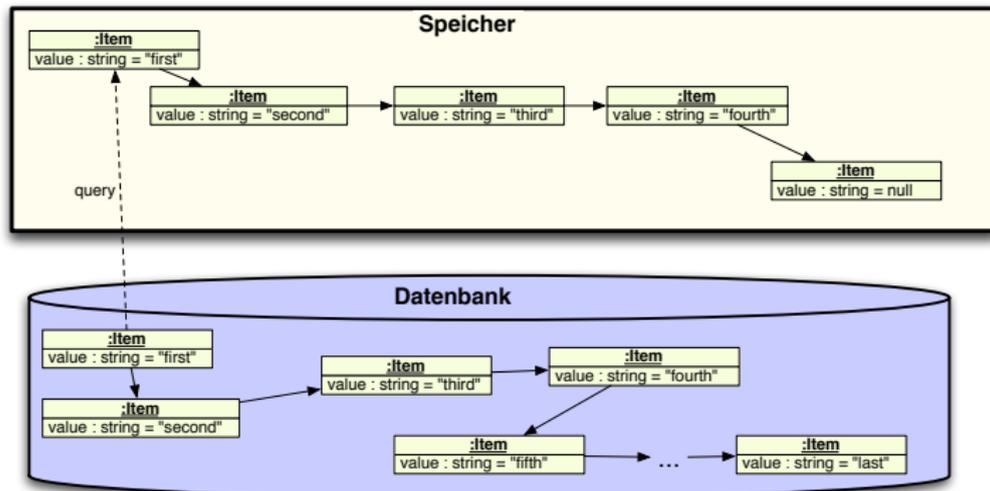
Integrität

Persistenz

Transaktionen

Zusammenfassung

- ▶ Welche Objekte werden beim Anfragen aktiviert?



- ▶ Aktivierungstiefe (*activation depth*)
- ▶ Standardwert: 5
- ▶ Objekte auf Ebene 5 werden nicht mehr (implizit) aktiviert (Attribute sind `null`)
- ▶ Explizite Aktivierung: `ObjectContainer.activate(0, n)`
- ▶ automatische Aktivierung: *Transparent Activation mode*

Data Base for  
Objects (DB4O)

Identitätskonzept

Integrität

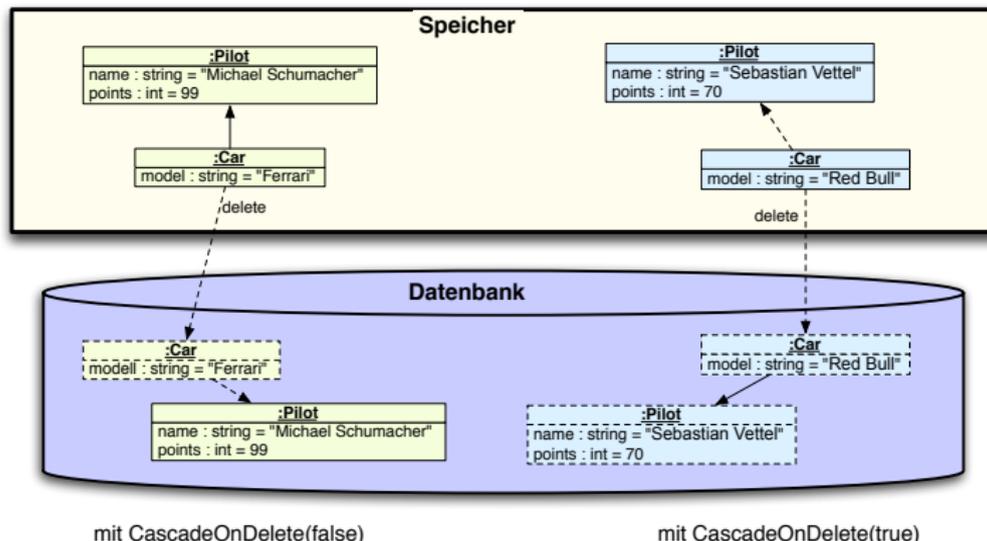
Integrität

Persistenz

Transaktionen

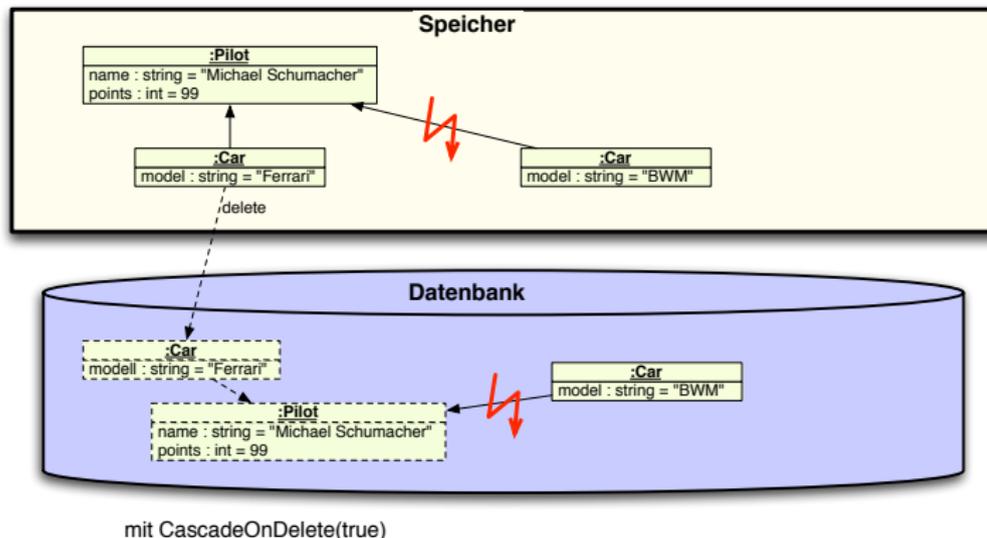
Zusammenfassung

- ▶ Welche Objekte werden beim Löschen entfernt?



- ▶ nicht-kaskadierendes Löschen (Standard): nur gelöscht Objekt wird entfernt
- ▶ kaskadierendes Löschen: gelöscht Objekt und referenzierte Objekte werden entfernt

- ▶ Welche Objekte werden beim kaskadierenden Löschen entfernt?



- ▶ kaskadierendes Löschen: Objekte werden entfernt, auch wenn sie noch von anderen Objekten referenziert werden.

Data Base for  
Objects (DB4O)

Identitätskonzept

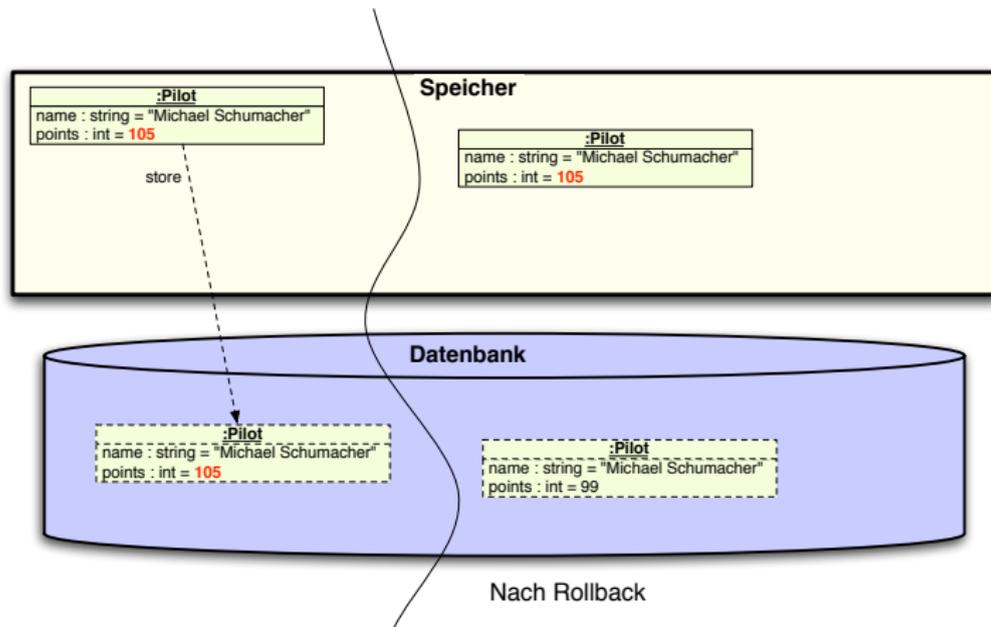
Integrität

Integrität

Persistenz

Transaktionen

Zusammenfassung



- ▶ Jede Operation findet im Kontext einer Transaktion statt.
- ▶ Wird implizit durch Öffnen der Datenbank (`openFile`) erzeugt.
- ▶ `db.commit()`, `db.rollback()`
- ▶ **Achtung!**  
Transiente Objekte werden nicht automatisch aktualisiert.  
Sie müssen explizit neu geladen werden (`refresh`)

Identitätskonzept

Integrität

Integrität

Persistenz

Transaktionen

► Fragen?

**Data Base for  
Objects (DB4O)**

Identitätskonzept

Integrität

Integrität

Persistenz

Transaktionen

Zusammenfassung