Agenda

Assoziationen

Cascading

n-1

1-n

1-1

m-n



Propagieren von CRUDE O

Assoziationen

Cascading

n-1

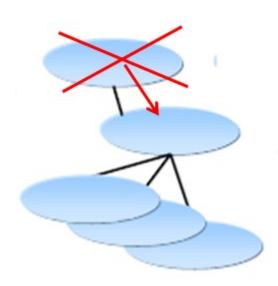
1-n

1-1

m-n

Komponenten

Soll ein persist(), delete() etc. auch auf abhängige Objekte angewandt werden?





Kaskadierungsverhalten mit

Cascado Sul

- Folgende Cascade Style sind vorhanden:
 - PERSIST, MERGE, REMOVE, REFRESH, DETACH
 - ALL
 - Default: Kaskadierung deaktiviert
- Delete-Orphan seit JPA 2.0



Cascading

n-1

1-n

1-1

m-r



Cascading

n-1

1-n

1-1

Komponenten

Cascade Styles (Hibernate)

- Folgende Cascade Style sind vorhanden:
 - save-update, delete, lock, evict, replicate, all, deleteorphan

```
<one-to-one name= "adresse" cascade="save-update"/>
<one-to-one name="adresse" cascade="save-update, delete"/>
```



Agenda

Assoziationen

Cascading

<u>n-1</u>

1-n

1-1

m-n



Relationen vs. Assoziationen

Assoziationen

Cascading

<u>n-1</u>

1-n

1-1

m-n

Relationen	Assoziationen
Verweis auf andere Tabellenzeile	uni- und bidirektional
Beziehung über Fremdschlüssel	Referenzen zu anderen Entitäten
gerichtet	Collection zu vielen anderen Entitäten (Set, List, Map)



Assoziationen

Cascading

<u>n-1</u>

1-n

1-1

m-n

Komponenten

B00K

BOOK_ID BIGINT(19) NOT NULL (PK)
ISBN VARCHAR(50) NOT NULL
BOOK_NAME VARCHAR(100) NOT NULL
PUBLISH_DATE DATE(10) NULL
PRICE INTEGER(10) NULL
PUBLISHER_ID BIGINT(19) NULL (FK)

PUBLISHER

PUBLISHER_ID BIGINT(19) NOT NULL (PK)
CODE VARCHAR(4) NOT NULL
PUBLISHER_NAME VARCHAR(100) NOT NULL
ADDRESS VARCHAR(200) NULL





package com.hibernaterecipes.chapter5;

```
Cascading
n-1
```

1-n

1-1

m-n

Komponenten

Assoziationen

```
//imports
@Entity
//@Table(name = "PUBLISHER", schema = "BOOK5
public class Publisher implements Serializab.
  @Id
  @Column(name = "PUBLISHER_ID")
  @GeneratedValue(strategy = GenerationType.AUTO)
  private Long publisher_id;
  @Column(name = "CODE")
  private String code;
  @Column(name = "PUBLISHER_NAME")
  private String name;
  @Column(name = "ADDRESS")
  private String address;
  // getters and setters
```

JPΔ

```
package com.hibernaterecipes.chapter5;
//imports
@Entity
//@Table(name = "BOOK", schema = "BOOK5")
public class Book 5 1 implements Serializable {
  @Id
  @Column(name = "BOOK ID")
  @GeneratedValue(strategy = GenerationType.AUTO)
  private Long book_id;
  @Column(name = "isbn")
  private String isbn;
  @Column(name = "BOOK_NAME")
  private String name;
  @ManyToOne
  @Cascade(value = CascadeType.SAVE UPDATE)
  @JoinColumn(name = "PUBLISHER ID")
  private Publisher publisher;
  @Column(name = "price")
  private Integer price;
  // getters and setters
```

Assoziationen

Cascading

<u>n-1</u>

1-n

1-1

m-n



```
package com.hibernaterecipes.chapter5;
                                                    Komponenten
import java.io.Serializable;
public class Publisher implements Serializable {
  private Long publisher_id;
  private String code;
  private String name;
   private String address;
                                            Publisher
  // getters and setters
```



Assoziationen

Cascading

n-1 1-n 1-1 m-n

```
m-n
<!DOCTYPE hibernate-mapping PUBLIC</pre>
                                                                        Komponenten
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="com.hibernaterecipes.chapter5">
   <class name="Publisher" table="Publisher" schema="BOOK5">
    <id name="Publisher id" type="long" column="PUBLISHER ID">
             <generator class="native">
             </generator>
    </id>
    cproperty name="code" type="string">
             <column name="CODE" length="4" not-null="true" unique="true" />
    </property>
    cproperty name="name" type="string">
             <column name="PUBLISHER NAME" length="100" not-null="true" />
    </property>
    cproperty name="address" type="string">
             <column name="ADDRESS" length="200" />
    </property>
   </class>
                                                             Publisher
```



</hibernate-mapping>

Assoziationen

Cascading

n-1 1-n 1-1

```
package com.hibernaterecipes.chapter5;
import java.io.Serializable;
import java.util.Date;
                                             Publisher
public class Book 5 1 implements Serializable {
  private Long book_id;
  private String isbn;
  private String name;
  private Publisher publisher;
  private Date publishDate;
  private Integer price;
  // getters and setters
```



Cascading

n-1

1-n

1-1

m-n



hbm yml tit

```
<!DOCTYPE hibernate-mapping PUBLIC</pre>
                                                                        m-n
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
                                                                        Komponenten
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="com.hibernaterecipes.chapter5">
   <class name="Book 5 1" table="Book" schema="BOOK5">
    <id name="book id" type="long" column="BOOK ID">
                                                             Publisher
             <generator class="native">
             </generator>
    </id>
    cproperty name="isbn" type="string">
             <column name="ISBN" length="50" not-null="true" unique="true" />
    </property>
    cproperty name="name" type="string">
             <column name="BOOK NAME" length="100" not-null="true" />
    </property>
    cproperty name="publishDate" type="date" column="PUBLISH DATE" />
    cproperty name="price" type="int" column="PRICE" />
    <many-to-one name="publisher" class="Publisher" column="PUBLISHER ID"</pre>
             cascade="save-update" />
   </class>
</hibernate-mapping>
```



Assoziationen

Cascading

n-1 1-n 1-1

Assoziationen

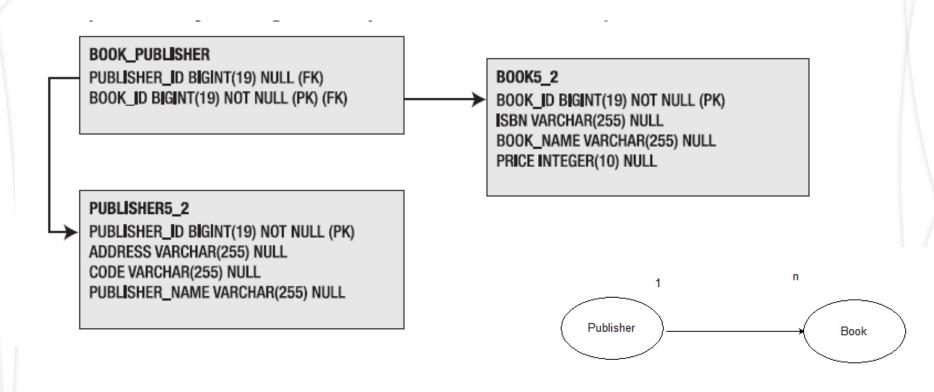
Cascading

<u>n-1</u>

1-n

1-1

m-n





```
m-n
package com.hibernaterecipes.chapter5;
                                                    Komponenten
// imports
@Entity
//@Table(name = "BOOK5 2")
public class Book_5_2 implements Serializable {
  // ......
  @ManyToOne
  @Cascade(value = CascadeType.SAVE UPDATE)
  @JoinTable( name = "BOOK PUBLISHER", joinColumns =
  @JoinColumn(name = "BOOK_ID"), inverseJoinColumns =
  @JoinColumn(name = "PUBLISHER ID"))
  private Publisher5 2 publisher
   // getters and setters
                                    Publisher
                                                       Book
```



Assoziationen

Cascading

n-1 1-n 1-1

Join Table 111

</hibernate-mapping>

```
1-1
<!DOCTYPE hibernate-mapping PUBLIC</pre>
                                                                         m-n
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
                                                                         Komponenten
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="com.hibernaterecipes.chapter5">
<class name="Book 5 2" table="Book5 2" >
   <id name="book id" type="long" column="BOOK ID"</pre>
    <generator class="native">
                                                       Publisher
                                                                              Book
    </generator>
   </id>
   cproperty name="isbn" type="string">
    <column name="ISBN" length="50" not-null="true" unique="true" />
   </property>
   cproperty name="name" type="string">
    <column name="BOOK NAME" length="100" not-null="true" />
   </property>
   cproperty name="publishDate" type="date" column="PUBLISH DATE" />
   cproperty name="price" type="int" column="PRICE" />
   <join table="BOOK_PUBLISHER" optional="true" >
    <key column="BOOK ID" unique="true" />
    <many-to-one name="publisher" class="Publisher5 2" column="PUBLISHER ID"</pre>
              not-null="true" cascade="save-update" lazy="false" />
   </join>
</class>
```



Assoziationen

Cascading

<u>n-1</u> 1-n

Agenda

Assoziationen

Cascading

n-1

1-n

1-1

m-n



1:n Assoziation

Beispiel:

- Book Chapter
- 1 : n

Assoziationen

Cascading

n-1

<u>1-n</u>

1-1

m-n



1:n Assoziation unidirektional

package com.hibernaterecipes.chapter7;

Book

```
Assoziationen
```

```
Cascading
```

n-1

<u>1-n</u> 1-1

m

Komponenten

Chapter

```
import java.util.Date;
import java.util.Set;
public class Book {
 ..//
 private Set chapters;
 // getters and setters
```

1:n Assoziation unidirektional

Assoziationen

Cascading

Komponenten

n-1

1-1

package com.hibernaterecipes.chapter7;

```
public class Chapter {
 private Long id;
 private String title;
 private int noOfPages
```



```
// getters and setters
```



Cascading

n-1

<u>1-n</u> 1-1

m-

Vampananta

```
1:n Assoziation under the Book Book Chapter
```

```
<!DOCTYPE hibernate-mapping PUBLIC</pre>
                                                                   Komponenten
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="com.hibernaterecipes.chapter7">
  <class name="Book" table="BOOK" >
   <id name="book id" column="BOOK ID" type="long">
            <generator class="native">
            </generator>
   </id>
   cproperty name="isbn" type="string" column="ISBN" />
   cproperty name="bookName" type="string" column="BOOK NAME" />
   cproperty name="publishDate" type="date" column="PUBLISH DATE" />
   cproperty name="price" type="long" column="PRICE" />
   <set name="chapters">
            <key column="BOOK ID" />
            <one-to-many class="Chapter" />
   </set>
  </class>
</hibernate-mapping>
```



Cascading

n-1

Chapter

1-1

```
1:n Assoziation u
                        Book
```

```
<!DOCTYPE hibernate-mapping PUBLIC</pre>
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="com.hibernaterecipes.chapter7">
<class name="Chapter" table="CHAPTER" dynamic-insert="true"</pre>
  dynamic-update="true">
     <id name="id" column="id" type="long">
        <generator class="native">
        </generator>
     </id>
     cproperty name="title" type="string" column="title" />
     cproperty name="noOfPages" type="int"
   column="NUM OF PAGES" />
</class>
</hibernate-mapping>
```

Cascading

n-1

<u>1-n</u> 1-1

m-

Chapter

Komponenten

```
1:n Assoziation u
```

```
package com.hibernaterecipes.chapter7;
// imports
@Entity(name = "bkch2")
@Table(name = "BOOK7_1" )
public class Book {
  //..
  @OneToMany(targetEntity = Chapter.class)
  @JoinColumn(name = "book id")
  @Cascade(value = { CascadeType.SAVE UPDATE,
  CascadeType.DELETE_ORPHAN })
  Set chapters=new HashSet();
  // setters and gettes
```

Book



Cascading

n-1

1-n 1-1

```
1:n Assoziation b
                             Book
                                             Chapter
```

```
<!DOCTYPE hibernate-mapping PUBLIC</pre>
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="com.hibernaterecipes.chapter7">
  <class name="Book" table="BOOK">
   <id name="book id" column="BOOK ID" type="long">
            <generator class="native">
            </generator>
   </id>
   cproperty name="isbn" type="string" column="ISBN" />
   cproperty name="bookName" type="string" column="BOOK NAME" />
   cproperty name="publishDate" type="date" column="PUBLISH DATE" />
   cproperty name="price" type="long" column="PRICE" />
   <set name="chapters" inverse="true">
            <key column="BOOK ID" />
            <one-to-many class="Chapter" />
   </set>
  </class>
</hibernate-mapping>
```

Cascading

n-1

<u>1-n</u> 1-1

Chapter

mn

```
1:n Assoziation biling
```

```
<!DOCTYPE hibernate-mapping PUBLIC</pre>
                                                                Komponenten
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="com.hibernaterecipes.chapter7">
  <class name="Chapter" table="CHAPTER" dynamic-insert="true"</pre>
   dynamic-update="true" >
   <id name="id" column="id" type="long">
            <generator class="native">
            </generator>
   </id>
   cproperty name="title" type="string" column="title" />
   cproperty name="noOfPages" type="int" column="NUM OF PAGES" />
   <many-to-one name="book" column="book id" class="Book"></many-to-</pre>
  one>
  </class>
</hibernate-mapping>
```



1:n Assoziation b



```
package com.hibernaterecipes.chapter7;
//..
@Entity(name = "chapter")
@Table(name = "Chapter7 1")
public class Chapter {
  @ManyToOne
  //@JoinColumn(name = "book_id")
  private Book book;
  // getters and setters
```

Assoziationen

Cascading

n-1

<u>1-n</u> 1-1

m-

1:n Assoziation b



```
package com.hibernaterecipes.chapter7;
//imports
@Entity
@Table(name = "BOOK7 1")
public class Book {
  //..
  @OneToMany(targetEntity = Chapter.class)
  @JoinColumn(name = "BOOK ID")
  @Cascade(value = { CascadeType.SAVE UPDATE,
  CascadeType.DELETE ORPHAN })
  Set chapters=new HashSet();
  // getters and setters
```

Assoziationen

Cascading n-1 <u>1-n</u> 1-1

1:n Assoziation unidirektional

```
package com.hibernaterecipes.chapter7;
```

```
public class Chapter {
  private Long id;
  private String title;
  private int noOfPages;
  // getters and setters
}
```

Assoziationen

Cascading

n-1

<u>1-r</u>

1-1

m-



1:n Assoziation unidirektional

```
package com.hibernaterecipes.chapter7;
import java.util.Date;
import java.util.Set;
public class Book {
  private Long book id;
  private String isbn;
  private String bookName;
  private Date publishDate;
  private Long price;
  private Set chapters;
  // getters and setters
```

Assoziationen

Cascading n-1 <u>1-n</u> 1-1

Komponenten

Chapter

Book



Cascading

n-1

1-n 1-1

```
1:n Assoziation bidirektional
                                       Chapter
```

```
<!DOCTYPE hibernate-mapping PUBLIC</pre>
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="com.hibernaterecipes.chapter7">
  <class name="Chapter" table="CHAPTER" dynamic-insert="true"</pre>
   dynamic-update="true">
   <id name="id" column="id" type="long">
          <generator class="native">
          </generator>
   </id>
   cproperty name="title" type="string" column="title" />
   cproperty name="noOfPages" type="int"
  column="NUM_OF_PAGES" />
  </class>
</hibernate-mapping>
```

Cascading

n-1

1-n 1-1

Chapter

```
<!DOCTYPE hibernate-mapping PUBLIC</pre>
                                                                Komponenten
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="com.hibernaterecipes.chapter7">
  <class name="Book" table="BOOK">
   <id name="book_id" column="BOOK_ID" type="long">
            <generator class="native">
            </generator>
   </id>
   cproperty name="isbn" type="string" column="ISBN" />
   cproperty name="bookName" type="string" column="BOOK_NAME" />
   cproperty name="publishDate" type="date" column="PUBLISH_DATE" />
   cproperty name="price" type="long" column="PRICE" />
   <set name="chapters" table="BOOK CHAPTER" cascade="save-update,
delete-orphan">
            <key column="BOOK ID" />
            <many-to-many column="CHAPTER_ID" class="Chapter"</pre>
                    unique="true" />
   </set>
  </class>
```

1:n Assoziation bidirektional



1:n Assoziation bidirektional

Hibernat

```
Book Chapter
```

```
Cascading
n-1
1-n
1-1
m-n
Komponenten
```

- Anstatt one-to-many nutzt man <many-to-many>, da one-to-many nichts über Join Tables weiss.
- Mit unique attribute auf true sagt man dass ein Book ein Chapter nur einmal besitzen kann (durch hashCode() und equals()) und man implementiert dadurch eine one-to-many Association.



1:n Assoziation unidirektional

package com.hibernaterecipes.chapter7;

```
Assoziationen
Cascading
n-1
1-n
```

1-1

Komponenten

Chapter

```
// imports
@Entity
                                     Book
@Table(name = "BOOK7 3")
public class Book {
  //..
  @OneToMany(targetEntity = Chapter.class)
  @JoinTable(name = "Book Chapter", joinColumns =
  { @JoinColumn(name = "book id") })
  @Cascade(value = { CascadeType.SAVE_UPDATE,
  CascadeType.DELETE ORPHAN })
  private Set chapters=new HashSet();
  // getters and setters
```



1:n Assoziation bidirektional

```
<!DOCTYPE hibernate-mapping PUBLIC</pre>
                                                                                        Komponenten
           "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
           "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
           <hibernate-mapping package="com.hibernaterecipes.chapter7">
                   <class name="Book" table="BOOK">
                              <id name="book id" column="BOOK ID" type="long">
                                         <generator class="native">
                                          </generator>
                              </id>
Book
                     Chapter
                              cproperty name="isbn" type="string" column="ISBN" />
                              cproperty name="bookName" type="string" column="BOOK NAME" />
                              cproperty name="publishDate" type="date"
        column="PUBLISH DATE" />
                              cproperty name="price" type="long" column="PRICE" />
                              <set name="chapters" table="BOOK CHAPTER" cascade="save-update,</pre>
                                                     delete-orphan">
                                          <key column="BOOK ID" />
                                          <many-to-many column="CHAPTER ID" class="Chapter"</pre>
                                                     unique="true" />
                              </set>
                   </class>
           </hibernate-mapping>
```



Assoziationen

Cascading

n-1 1-n 1-1 m-n

1:n Assoziation bidirektional

```
<!DOCTYPE hibernate-mapping PUBLIC</pre>
                                                                                              Komponenten
                "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
                "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
                <hibernate-mapping package="com.hibernaterecipes.chapter7">
                        <class name="Chapter" table="CHAPTER" dynamic-insert="true"</pre>
                                    dynamic-update="
                                               true" schema="BookShop7">
                                     id name="id" column="id" type="long">
                      n
                                               <generator class="native">
                                               </generator>
                           Chapter
Book
                                     /id>
                                    cproperty name="title" type="string" column="title" />
                                    cproperty name="noOfPages" type="int" column="NUM OF PAGES" />
                                    <join table="BOOK CHAPTER" optional="true" inverse="true" >
                                               <key column="CHAPTER ID" unique="true" />
                                               <many-to-one name="book" class="Book" column="BOOK ID"</pre>
                                                          not-null="true" />
                                    </join>
                        </class>
                </hibernate-mapping>
```



Cascading n-1

1-n

1-1

m-n



1:n Assoziation bidirektional

```
package com.hibernaterecipes.chapter7;
                                                    Komponenten
// imports
@Entity(name = "bkch73")
@Table(name = "BOOK7_39")
                                      Book
                                                       Chapter
public class Book {
  //..
  @OneToMany(targetEntity = Chapter.class)
  @JoinTable(name = "Book_Chapter", joinColumns =
  { @JoinColumn(name = "book_id") })
  @Cascade(value = { CascadeType.SAVE UPDATE,
  CascadeType.DELETE ORPHAN })
  private Set chapters = new HashSet();
// getter & setter
```



Assoziationen

Cascading

n-1

1-1

Aufgabe



- Aufgabe 4:
- Beziehungen (n-1 und 1-n)

Assoziationen

Cascading

n-1

<u>1-n</u>

1-1

m-n



@JoinColumn(name = "id"))

// getters and setters

private Book book;

@JoinColumn(name = "book id")

```
1-1
                                                         Komponenten
package com.hibernaterecipes.chapter7;
//imports
@Entity
                                          Book
                                                             Chapter
@Table(name = "Chapter7 30")
public class Chapter {
  //...
  @ManyToOne
```

@JoinTable(name = "Book_Chapter5", joinColumns =

Assoziationen

Cascading

n-1

Agenda

Assoziationen

Cascading

n-1

1-n

<u>1-1</u>

m-n



Beispiel Sharing Primary Key

Hibernat



Assoziationen

Cascading
n-1
1-n
1-1
m-n

```
package com.hibernaterecipes.chapter5;
// imports
public class Customer5_1 implements Serializable {
  private static final long serialVersionUID =
  -3534434932962734600L;
  private Long id;
  private String countryCode;
  private String idCardNo;
  private String firstName;
  private String lastName;
  private Address5 1 address;
  private String email;
  // getters and setters
```



Cascading

n-1

1-n

1-1 m-n

```
Beispiel Sharing Primary
                                             Address
                                Customer
```

```
<!DOCTYPE hibernate-mapping PUBLIC</pre>
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="com.hibernaterecipes.chapter5">
  <class name="Customer5 1" table="CUSTOMER" >
   <id name="id" type="long" column="ID">
            <generator class="native"></generator>
   </id>
   cproperty name="firstName" type="string" column="FIRST_NAME" />
   cproperty name="lastName" type="string" column="LAST NAME" />
   cproperty name="idCardNo" type="string" column="ID_CARD_NO" />
   cproperty name="countryCode" type="string"
  column="COUNTRY_CODE" />
   cproperty name="email" type="string" column="EMAIL" />
   <one-to-one name="address"</pre>
  class="com.hibernaterecipes.chapter5.Address5_1"
            cascade="save-update"></one-to-one>
  </class>
</hibernate-mapping>
```

Beispiel Sharing Primary Key

Hibernat



```
Cascading
n-1
1-n
1-1
```

m-n

Assoziationen

```
package com.hibernaterecipes.chapter5;
import java.io.Serializable;
public class Address5 1 implements Serializable {
  private static final long serialVersionUID =
  -605474766287314591L;
  private Long id;
  private String city;
  private String street;
  private String doorplate;
  private Customer5 1 customer;
  // getters and setters
```

Cascading

n-1

1-n

<u>1-1</u> m-n

```
Beispiel Sharing Primary K
```

```
<!DOCTYPE hibernate-mapping PUBLIC</pre>
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="com.hibernaterecipes.chapter5">
  <class name="Address5 1" table="ADDRESS" >
   <id name="id" type="long" column="ID">
            <generator class="foreign">
                    <param name="property">customer</param>
            </generator>
   </id>
   cproperty name="city" type="string" column="CITY" />
   cproperty name="street" type="string" column="STREET" />
   cproperty name="doorplate" type="string" column="DOOR_PLATE" />
   <one-to-one name="customer" class="Customer5_1"</pre>
            constrained="true">
   </one-to-one>
  </class>
</hibernate-mapping>
```

Beispiel Sharing Primary Key



Cascading n-1 1-n 1-1

m-n

Assoziationen

```
package com.hibernaterecipes.chapter5;
// imports
@Entity
@Table(name = "ADDRESS")
public class Address5 1 implements Serializable {
  private static final long serialVersionUID = -605474766287314591L;
  @Id
  @Column(name = "ADDRESS_ID")
  private Long id;
  @Column(name = "CITY")
  private String city;
  @Column(name = "STREET")
  private String street;
  @Column(name = "DOOR PLATE")
  private String doorplate;
  // getters and setters
```



Cascading

n-1

1-n

<u>1-1</u>

m-n

Komponenten

```
package com.hibernaterecipes.chapter5;
//imports
@Entity
@Table(name = "CUSTOMER")
public class Customer5_1 implements Serializable {
  @OneToOne
  @PrimaryKeyJoinColumn(name = "ID")
  private Address5 1 address;
  // getters and setters
```

Customer

Beispiel Sharing Primary Key

Cascading

n-1

1-n

<u>1-1</u> m-n

Komponenten

Beispiel Sharing Primary Key

- Achtung
- Werden Address-Objekte persistiert
 - muss manuell hierzu die Customer-Id gesetzt werden.

Customer

Cascading

```
<hibernate-mapping package="com.hibernaterecipes.chapter5">
  <class name="Customer5_2" table="CUSTOMER" schema="BOOK5">
   <id name="id" type="long" column="ID">
            <generator class="native"></generator>
   </id>
   cproperty name="firstName" type="string" column="FIRST_NAME" />
   cproperty name="lastName" type="string" column="LAST NAME" />
   cproperty name="idCardNo" type="string" column="ID CARD NO" />
   cproperty name="countryCode" type="string" column="COUNTRY CODE" />
   cproperty name="email" type="string" column="EMAIL" />
   <many-to-one name="address"</pre>
            class="com.hibernaterecipes.chapter5.Address5_2"
  column="ADDRESS ID"
            cascade="save-update" unique="true">
   </many-to-one>
                                          Macht es zu einer 1:1
  </class>
</hibernate-mapping>
                                          Beziehung
```



Assoziationen Cascading n-1

1-n

1-1 m-n

```
<!DOCTYPE hibernate-mapping PUBLIC</pre>
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="com.hibernaterecipes.chapter5">
<class name="Address5_2" table="ADDRESS" schema="BOOK5">
   <id name="id" type="long" column="ADDRESS_ID" >
                   <generator class="native">
                   </generator>
   </id>
   cproperty name="city" type="string" column="CITY" />
   cproperty name="street" type="string" column="STREET" />
   cproperty name="doorplate" type="string" column="DOOR PLATE" />
   <one-to-many name="customer" class="Customer5 2"</pre>
  propertyref="address">
   </one-to-many>
</class>
</hibernate-mapping>
```



JPΔ



```
package com.hibernaterecipes.chapter5;
// imports
@Entity
@Table(name = "ADDRESS")
public class Address5_2 implements Serializable {
  //..
  @OneToOne(mappedBy = "address")
  private Customer5_2 customer;
  // getters and setters
```

Assoziationen

Cascading

n-1

1-n

<u>1-1</u>

m-n

JPA



Assoziationen

Cascading
n-1
1-n
1-1
m-n

```
package com.hibernaterecipes.chapter5;
// imports
@Entity
@Table(name = "CUSTOMER")
public class Customer5_2 implements Serializable {
  //...
  @OneToOne(cascade = CascadeType.ALL)
  @JoinColumn(name = "ADDRESS_ID")
  private Address5_2 address;
  // getters and setters
```



Beispiel 1-1 bidirektional Hibernate Join Table

```
Assoziationen
```

```
Cascading
n-1
1-n
1-1
m-n
```

```
<!DOCTYPE hibernate-mapping PUBLIC</pre>
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="com.hibernaterecipes.chapter5">
  <class name="Address5_3" table="Address" >
   <id name="addressId" type="long" column="ADDRESS_ID">
            <generator class="native">
            </generator>
                                                                         Address
                                                   Customer
   </id>
   cproperty name="city" type="string" column="CITY" />
   cproperty name="street" type="string" column="STREET" />
   cproperty name="doorplate" type="string" column="DOOR PLATE" />
   <join table="CustomerAddress" optional="true" inverse="true">
            <key column="ADDRESS ID" unique="true" />
             <many-to-one name="customer"</pre>
                      class="com.hibernaterecipes.chapter5.Customer5 3"
  column="ID"
                      unique="true" not-null="true">
            </many-to-one>
   </join>
  </class>
</hihernate-manning>
```

Hibernat

```
m-n
<!DOCTYPE hibernate-mapping PUBLIC</pre>
                                                                        Komponenten
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="com.hibernaterecipes.chapter5">
  <class name="Customer5 3" table="Customer" >
    <id name="id" type="long" column="ID">
                                                                              Address
                                                      Customer
             <generator class="native"></generato</pre>
    </id>
    cproperty name="firstName" type="string" column="FIRST NAME" />
    cproperty name="lastName" type="string" column="LAST NAME" />
    cproperty name="idCardNo" type="string" column="ID CARD NO" />
    countryCode type="string column="COUNTRY_CODE" />
    cproperty name="email" type="string" column="EMAIL" />
    <join table="CustomerAddress" optional="true" >
             <key column="ID" unique="true">
             </key>
             <many-to-one name="address" column="ADDRESS ID" not-null="true"</pre>
                       cascade="save-update" unique="true">
             </many-to-one>
    </join>
   </class>
</hibernate-mapping>
```



Assoziationen

Cascading

n-1 1-n **1-1**

Assoziationen

Cascading n-1 1-n 1-1 m-n

Komponenten

Address

```
package com.hibernaterecipes.chapter5:
//imports
                                      Customer
@Entity
@Table(name = "ADDRESS")
public class Address5_3 implements Serializable {
  //..
  @OneToOne(mappedBy = "address")
  private Customer5 3 customer;
// getters and setters
```



```
Assoziationen
```

```
Cascading
n-1
1-n
1-1
m-n
```

Komponenten

Address

```
package com.hibernaterecipes.chapter5;
// imports
                                          Customer
@Entity
@Table(name = "CUSTOMER")
public class Customer5_3 implements Serializable {
  //..
  @OneToOne(cascade = CascadeType.ALL)
  @JoinTable(name = "CustomerAddress", joinColumns =
  @JoinColumn(name = "ID"), inverseJoinColumns =
  @JoinColumn(name = "ADDRESS_ID"))
  private Address5_3 address;
  // getters and setters
```



Aufgabe



Komponenten

Assoziationen

Cascading

n-1 1-n <u>1-1</u> m-n

Aufgabe 5: Beziehungen (1-1)



Agenda

Assoziationen

Cascading

n-1

1-n

1-1

m-n





```
Cascading
n-1
1-n
1-1
m-n
Komponenten
```

Assoziationen

```
<!DOCTYPE hibernate-mapping PUBLIC</pre>
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="com.hibernaterecipes.chapter7">
  <class name="Book" table="BOOK" >
    <id name="book id" column="BOOK ID" type="long">
             <generator class="native">
             </generator>
    </id>
    cproperty name="isbn" type="string" column="ISBN" />
    cproperty name="bookName" type="string" column="BOOK_NAME" />
    cproperty name="publishDate" type="date" column="PUBLISH DATE" />
    cproperty name="price" type="long" column="PRICE" />
    <set name="chapters" table="BOOK CHAPTER"</pre>
             cascade="save-update,delete-orphan">
             <key column="BOOK ID" />
             <many-to-many column="ID" class="Chapter" unique="true" />
    </set>
   </class>
</hibernate-mapping>
```

Cascading

n-1

1-n

1-1

<u>m-n</u>

n:m Assoziation unidirektional
Hibernat

```
<!DOCTYPE hibernate-mapping PUBLIC</pre>
                                                         Komponenten
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="com.hibernaterecipes.chapter7">
  <class name="Chapter" table="CHAPTER7 4" dynamic-
  insert="true"
   dynamic-update="true" >
   <id name="id" column="id" type="long">
          <generator class="native">
          </generator>
   </id>
   cproperty name="title" type="string" column="title" />
   cproperty name="noOfPages" type="int"
  column="NUM_OF_PAGES" />
  </class>
</hibernate-mapping>
```



package com.hibernaterecipes.chapter7;

private Set chapters=new HashSet();

// setter & getter

JPΔ



// imports @Entity(name = "bkch74") @Table(name = "BOOK77") public class Book { //... @ManyToMany(targetEntity = Chapter.class) @JoinTable(name = "Book_Chapter22", joinColumns = { @JoinColumn(name = "book_id") }, inverseJoinColumns = { @JoinColumn(name = "chapter_id") }) @Cascade(value = { CascadeType.SAVE UPDATE, CascadeType.DELETE ORPHAN })



Assoziationen

Cascading

Komponenten

n-1 1-n

1-1 m-n

```
m-n
<!DOCTYPE hibernate-mapping PUBLIC</pre>
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="com.hibernaterecipes.chapter7">
  <class name="Book" table="BOOK7 41" >
    <id name="book id" column="BOOK ID" type="long">
             <generator class="native">
             </generator>
    </id>
    cproperty name="isbn" type="string" column="ISBN" />
    cproperty name="bookName" type="string" column="BOOK NAME" />
    cproperty name="publishDate" type="date" column="PUBLISH DATE" />
    cproperty name="price" type="long" column="PRICE" />
    <set name="chapters" table="BOOK CHAPTER"</pre>
             cascade="save-update,delete-orphan">
             <key column="BOOK ID" />
             <many-to-many column="ID" class="Chapter" unique="true" />
    </set>
   </class>
</hibernate-mapping>
```

Book



60

Chapter

Cascading

n-1

1-n

1-1

```
Assoziationen
```

```
Cascading
n-1
1-n
1-1
```

m-n

```
<!DOCTYPE hibernate-mapping PUBLIC</pre>
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="com.hibernaterecipes.chapter7">
  <class name="Chapter" table="CHAPTER" dynamic-insert="true"</pre>
   dynamic-update="true" >
   <id name="id" column="id" type="long">
            <generator class="native">
            </generator>
   </id>
   cproperty name="title" type="string" column="title" />
   cproperty name="noOfPages" type="int" column="NUM OF PAGES" />
   <set name="book" table="BOOK CHAPTER" inverse="true"</pre>
  cascade="save-update">
            <key column="id"></key>
            <many-to-many class="Book" column="BOOK ID" />
   </set>
  </class>
</hibernate-mapping>
                                      Book
                                                            Chapter
```

```
package com.hibernaterecipes.chapter7;
                                                   Komponenten
@Entity(name = "bkch741")
@Table(name = "BOOK7 41666")
public class Book {
  @ManyToMany
  @JoinTable(name = "Book Chapter", joinColumns =
  { @JoinColumn(name = "book_id") },
  inverseJoinColumns = { @JoinColumn(name = "id") })
  @Cascade(value = { CascadeType.SAVE_UPDATE })
  private Set<Chapter> chapters = new
  HashSet<Chapter>();
  // getters and setters
                                                Chapter
                              Book
```



Assoziationen

Cascading

n-1 1-n 1-1

```
package com.hibernaterecipes.chapter7;
//imports
@Entity(name = "chapter741")
@Table(name = "Chapter7_417777")
public class Chapter {
  //..
  @ManyToMany(mappedBy = "chapters")
  private Set<Book> book = new HashSet<Book>();
  // getters and setters
```

Assoziationen

Cascading

n-1

1-n 1-1

m-n



Aufgabe



Assoziationen

Cascading

n-1

1-n

1-1

m-n

- Aufgabe 6 (optional):
- Beziehungen (m-n)

Customer	1 0*	Order	0* 0*	Article
 id: long firstname: string lastname: string creationDate: date changeDate: date type: UserType readDate: date 		- orderTime: date		- number: string - name: string - description: string - price: number



Agenda

Assoziationen

Cascading

n-1

1-n

1-1

m-n



Cascading

1-1

1-n

1-1

Komponenten

Component Mapping

- Entity-Komponente
 - haben einen Primärschlüssel
 - haben einen Lebenszyklus
- Value-Komponente
 - haben keine Datenbankidentität
 - haben keinen Primärschlüssel
 - gehören zu einer Entity und ihr Zustand wird innerhalb der Tabelle der dazugehörigen Entity gesichert
 - Ausnahme sind Collections von Value-Typen
 - typisch sind einfache Objekte vom Typ String
 - Lebensdauer eines Value-Typ ist ist immer an den Lebenszyklus der entsprechenden Entity gebunden
- Komponenten ermöglichen die Abbildung mehrerer Klassen auf eine Tabelle



Komponenten

Component Mapping

- Annotation @Embeddable definiert Komponente auf Klassenebene
- Kennzeichnung @Embedded beim entsprechenden Attribut bzw. bei der Getter-Methode

```
@Entity
public class Person {
    @Embedded
    private Address address;
    ...
    // getter/setter
}
@Embeddable
```

public class Address implements Serializable{



Component Mapping (2)

Spalten können umbenannt werden

```
@Entity
public class Person {
  @Embedded
  @AttributeOverrides({
            @AttributeOverride(name="street",
                    column=@Column (name="strasse"))
            @AttributeOverride(name="city",
                     column=@Column(name="stadt"))
            @AttributeOverride(name="country",
                     column=@Column(name="land"))
})
private Address address;
```

Assoziationen

Cascading

n-1

1-n

1-1

m-n



Collection von Components

Assoziationen

Cascading

n-1

1-n

1-1 m-n

Komponenten

<<Tabelle>>

Address

PERSON ID

STREET

CITY

eigene Tabelle ohne PK (JPA2)

```
<<Tabelle>>
@Entity
                                      Person
                                  PERSON_ID << PK>>
public class Person {
                                  NAME
  @ElementCollection
  private List<Address> addresses;
  // getter/setter
@Embeddable
public class Address implements Serializable{
```

Collection von Basistypen

Assoziationen

Cascading

n-1

1-n

1-1 m-n

Komponenten

eigene Tabelle ohne PK

```
<Tabelle>>
Person

PERSON_ID <<PK>>
NAME
...

<Tabelle>>
PHONENUMBERS

PERSON_ID
NUMBER
```

```
@Entity
public class Person {
    @ElementCollection
    private List<String> phoneNumbers;
    ...
// getter/setter
```



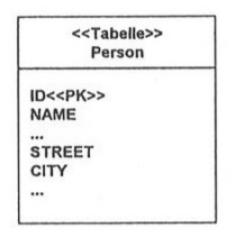
Component Mapping (Hibernate)

```
Assoziationen
```

Cascading n-1 1-n

1-1 m-n

```
<class name=" Person ">
  <id name="id">
      <generator .. />
  </id>
  cproperty name="name" />
  <component name="address"
  class="Address">
      roperty name="street" />
      cproperty name="city" />
      <parent name="person"/>
  </component>
```



- </class>
- Subkomponenten und Assoziationen zu anderen Komponenten
- keine geteilten Referenzen!
- Mapping erlaubt Rückreferenz mit <parent />



Collections von Komponenten

(Hibernet

```
<class name="Person">
  <id name="person_id">
       <generator .. />
  </id>
  cproperty name="name" />
  <set name="addresses" table="addresses" lazy="true">
       <key column="person id"/>
       <composite-element class="Address">
              cproperty name="street"/>
              cproperty name="city"/>
       </composite-element>
  </set>
                                               <<Tabelle>>
                        <<Tabelle>>
                                                Address
</class>
                          Person
                                            PERSON ID
                     PERSON_ID << PK>>
                                            STREET
                     NAME
                                            CITY
```

Assoziationen

Cascading

n-1

1-n

1-1

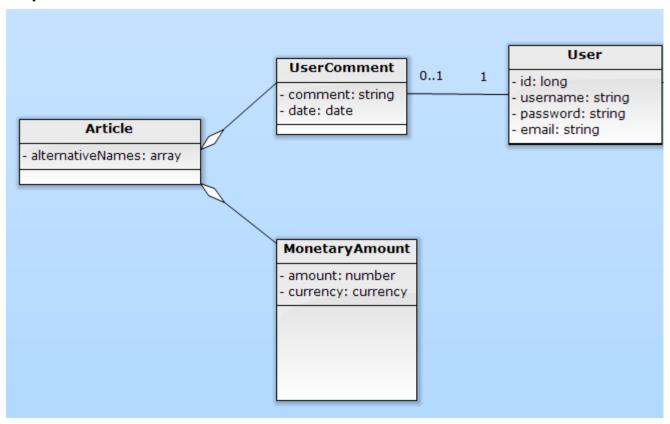
m-n

Aufgabe



- Assoziationen
- Cascading
- n-1
- 1-n
- 1-1 m-n

- Aufgabe 7 (optional):
- Komponenten





@SecondaryTable (1)

 Partitionierung der Attribute einer Klasse in zwei (oder mehr) Tabellen

```
Assoziationen
```

Cascading

n-1

1-n

1-1

```
@Entity
@Table(name = "Person")
@SecondaryTable(name =
    "EMBEDDED_ADDRESS", ...)
public class Person implements
    Serializable {
...
```

@SecondaryTable (2)

Ausgewählte Spalten können nun in die zweite Tabelle gelegt werden

```
@Embeddable
public class Address implements Serializable{
...
@Column(name = "STREET", table = "EMBEDDED_ADDRESS")
private String street
}
```

