

XML ile Entity İlişkileri



M:1 İlişki (XML)



```
<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.Pet">
        <many-to-one name="type"</pre>
class="com.javaegitimleri.petclinic.model.PetType" fetch="join">
            <column name="TYPE ID" />
        </many-to-one>
    </class>
</hibernate-mapping>
<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.PetType">
    </class>
</hibernate-mapping>
```

1:M – Unidirectional Set (XML)



```
<hibernate-mapping>
   <class name="com.javaegitimleri.petclinic.model.Pet">
        <set name="visits">
           <key column="PET ID"/>
           <one-to-many class="com.javaegitimleri.petclinic.model.Visit"/>
        </set>
   </class>
</hibernate-mapping>
<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.Visit">
    </class>
</hibernate-mapping>
```

1:M – Unidirectional List (XML)



```
<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.Owner">
       <list name="pets" inverse="false" table="T PET" lazy="true">
            <key column="OWNER ID"/>
            <list-index base="0" column="PET POSITION"/>
            <one-to-many class="com.javaegitimleri.petclinic.model.Pet" />
        </list>
    </class>
</hibernate-mapping>
<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.Pet">
    </class>
</hibernate-mapping>
```

1:M – Unidirectional Bag (XML)



```
<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.Owner">
        <bag name="pets" table="T_PET" inverse="false" lazy="true"</pre>
access="field">
            <key column="OWNER ID"/>
            <one-to-many class="com.javaegitimleri.petclinic.model.Pet" />
        </bag>
    </class>
</hibernate-mapping>
<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.Pet">
    </class>
</hibernate-mapping>
```

1:M – Unidirectional Map (XML)



```
<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.Owner">
        <map name="pets" table="T_PET" lazy="true" access="field">
            <key column="OWNER ID"/>
            <map-key type="string" formula="NAME"/>
            <one-to-many class="com.javaegitimleri.petclinic.model.Pet" />
        </map>
    </class>
</hibernate-mapping>
<hibernate-mapping>
   <class name="com.javaegitiml@ri.petclinic.model.Pet">
       cproperty name="name" type="string">
           <column name="NAME" />
       </class>
</hibernate-mapping>
```

1:M – Bidirectional Set (XML)



```
<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.Owner">
        <set name="pets" table="T_PET" inverse="true" lazy="true"</pre>
access="field">
             <key column="OWNER ID"/>
             <one-to-many class="com.javaegitimleri.petclinic.model.Pet" />
        </set>
    </class>
                                                mappedBy'ın XML'deki
</hibernate-mapping>
                                                karşılığıdır
<hibernate-mapping>
   <class name="com.javaegitimleri.petclinic.model.Pet">
        <many-to-one name="owner"</pre>
class="com.javaegitimleri.petclinic.model.Owner" fetch="join">
            <column name="OWNER ID" />
        </many-to-one>
   </class>
</hibernate-mapping>
```

1:M – Bidirectional List (XML)



```
<hibernate-mapping>
   <class name="com.javaegitimleri.petclinic.model.Owner">
      t-index base="0" column="PET_POSITION"/>
           <one-to-many class="com.javaegitimleri.petclinic.model.Pet" />
       </list>
   </class>
</hibernate-mapping>
<hibernate-mapping>
   <class name="com.javaegitimleri.petclinic.model.Pet">
       <many-to-one name="owner"</pre>
class="com javaegitimleri petclinic.model.Owner" fetch="join"
insert="false" update="false"
           <column name="OWNER ID" />
       </many-to-one>
   </class>
</hibernate-mapping>
```

1:M – Bidirectional Bag (XML)



```
<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.Owner">
        <bag name="pets" table="T PET" inverse="true" lazy="true"</pre>
access="field">
             <key column="OWNER ID"/>
             <one-to-many class="com.javaegitimleri.petclinic.model.Pet" />
        </bag>
    </class>
</hibernate-mapping>
<hibernate-mapping>
   <class name="com.javaegitimleri.petclinic.model.Pet">
        <many-to-one name="owner"</pre>
class="com.javaegitimleri.petclinic.model.Owner" fetch="join">
            <column name="OWNER ID" />
        </many-to-one>
   </class>
</hibernate-mapping>
```

1:M – Bidirectional Map (XML)



```
<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.Owner">
        <map name="pets" table="T_PET" inverse="true" lazy="true"</pre>
access="field">
             <key column="OWNER ID"/>
             <map-key type="string" formula="NAME"/>
             <one-to-many class="com.javaegitimleri.petclinic.model.Pet" />
         </map>
                                                mappedBy'ın XML'deki
    </class>
                                                karşılığıdır
</hibernate-mapping>
<hibernate-mapping>
   <class name="com.javaegitimleri.petclinic.model.Pet">
        <many-to-one name="owner"</pre>
class="com.javaegitimleri.petclinic.model.Owner" fetch="join">
            <column name="OWNER ID" />
        </many-to-one>
   </class>
</hibernate-mapping>
```

Join Tablo ile 1:M İlişkiler (XML)



```
<hibernate-mapping>
     <class name="com.javaegitimleri.petclinic.model.Owner">
         <set name="pets" table="T_OWNER_PET" inverse="false" lazy="true"</pre>
access="field">
             <key column="OWNER ID"/>
             <many-to-many class="com.javaegitimleri.petclinic.model.Pet"</pre>
unique="true" column="PET_ID"/>
         </set>
     </class>
                                            M:N ilişkiyi 1:M yapar
</hibernate-mapping>
<hibernate-mapping>
     <class name="com.javaegitimleri.petclinic.model.Pet">
     </class>
</hibernate-mapping>
```

Join Tablo ile 1:M İlişkiler (XML)



```
<hibernate-mapping>
     <class name="com.javaegitimleri.petclinic.model.Owner">
         <set name="pets" table="T_OWNER_PET" inverse="true" lazy="true"</pre>
access="field">
             <key column="OWNER ID"/>
             <many-to-many class="com.javaegitimleri.petclinic.model.Pet"</pre>
unique="true" column="PET_ID"/>
         </set>
     </class>
                                            M:N ilişkiyi 1:M yapar
</hibernate-mapping>
<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.Pet">
         <join table="T_OWNER_PET" inverse="false" optional="true">
            <key column="PET_ID"/>
            <many-to-one name="owner"</pre>
class="com.javaegitimleri.petclinic.model.Owner" column="OWNER ID"/>
        </ioin>
     </class>
</hibernate-mapping>
```

Tek Yönlü N:M İlişkiler (XML)



```
<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.Vet">
        <set name="specialties" table="T_VET_SPECIALTY" inverse="false"</pre>
lazv="true">
            <key column="VET ID"/>
            <many-to-many class="com.javaegitimleri.petclinic.model.Specialty"</pre>
column="SPECIALTY ID"/>
        </set>
    </class>
</hibernate-mapping>
<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.Specialty">
    </class>
</hibernate-mapping>
```

Çift Yönlü M:N İlişkiler (XML)



```
<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.Vet">
        <set name="specialties" table="T VET SPECIALTY" inverse="false"</pre>
lazv="true">
            <key column="VET_ID"/>
            <many-to-many class="com.javaegitimleri.petclinic.model.Specialty"</pre>
column="SPECIALTY ID"/>
        </set>
    </class>
</hibernate-mapping>
<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.Specialty">
        <set name="vets" table="T VET SPECIALTY" inverse="true" lazy="true">
            <key column="SPECIALTY ID"/>
            <many-to-many class="com.javaegitimleri.petclinic.model.Vet"</pre>
column="VET ID"/>
        </set>
    </class>
</hibernate-mapping>
```

Join Tablosunun Ara Entity Sınıf İle İfade Edilmesi (XML)



```
<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.Vet">
        <set name="vetSpecialties" table="T VET SPECIALTY" inverse="true" lazy="true">
            <key column="VET ID"/>
            <one-to-many class="com.javaegitimleri.petclinic.model.VetSpecialty"/>
        </set>
    </class>
</hibernate-mapping>
<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.Specialty">
        <set name="vetSpecialties" table="T VET SPECIALTY" inverse="true" lazy="true">
            <key column="SPECIALTY ID"/>
            <one-to-many class="com.javaegitimleri.petclinic.model.VetSpecialty"/>
        </set>
    </class>
</hibernate-mapping>
<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.VetSpecialty" table="T VET SPECIALTY">
        <many-to-one name="vet" class="com.javaegitimleri.petclinic.model.Vet" access="field"</pre>
fetch="ioin">
            <column name="VET ID" />
        </many-to-one>
        <many-to-one name="specialty" class="com.javaegitimleri.petclinic.model.Specialty"</pre>
access="field" fetch="join">
            <column name="SPECIALTY ID" />
        </many-to-one>
    </class>
</hibernate-mapping>
```

Foreign Key Üzerinden 1:1 Unidirectional İlişki (XML)



M:1 ilişkiyi 1:1 yapar

Foreign Key Üzerinden 1:1 Bidirectional İlişki (XML)



M:1 ilişkiyi 1:1 yapar

Primary Key Üzerinden 1:1 Unidirectional İlişki (XML)



```
<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.Owner">
       <id name="id" column="ID">
           <generator class="sequence"/>
       </id>
    </class>
</hibernate-mapping>
<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.Address">
       <id name="id" column="ID">
           <generator class="foreign">
               <param name="property">owner</param>
           </generator>
       </id>
       <one-to-one name="owner" constrained="true"/>
    </class>
</hibernate-mapping>
```

Owner olmadan Address'in olamayacağını anlatır

Primary Key Üzerinden 1:1 Bidirectional İlişki (XML)



```
<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.Owner">
       <id name="id" column="ID">
           <generator class="sequence"/>
       </id>
        <one-to-one name="address" property-ref="owner")>
    </class>
</hibernate-mapping>
<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.Address">
       <id name="id" column="ID">
           <generator class="foreign">
               <param name="property">owner</param>
           </generator>
       </id>
       <one-to-one name="owner" constrained="true"/>
    </class>
</hibernate-mapping>
```

Join Tablo ile 1:1 İlişkiler (XML)



```
<hibernate-mapping>
   <class name="com.javaegitimleri.petclinic.model.Owner">
       <id name="id" column="ID">
           <generator class="sequence" />
       </id>
   </class>
</hibernate-mapping>
<hibernate-mapping>
   <class name="com.javaegitimleri.petclinic.model.Address">
       <id name="id" column="ID">
           <qenerator class="sequence" />
       </id>
       <join table="T_OWNER_ADDRESS">
           <key column="ADDRESS_ID" unique="true" />
           <many-to-one name="owner" column="OWNER_ID" unique="true"</pre>
               not-null="true" />
       </join>
   </class>
</hibernate-mapping>
```

Parent - Child İlişkiler (XML)



```
<hibernate-mapping>
   <class name="com.javaegitimleri.petclinic.model.Pet">
        <set name="visits" table="T_VISIT" cascade="delete-orphan">
           <key column="PET ID"/>
           <one-to-many class="com.javaegitimleri.petclinic.model.Visit"/>
        </set>
   </class>
</hibernate-mapping>
<hibernate-mapping>
   <class name="com.javaegitimleri.petclinic.model.Visit">
   </class>
</hibernate-mapping>
```



Bileşen İçeren Set (XML)

```
<hibernate-mapping>
   <class name="com.javaegitimleri.petclinic.model.Pet">
      <id name="id" column="ID">
          <generator class="sequence"/>
      </id>
       <set name="images" table="T_PET_IMAGES">
          <key column="PET ID"/>
          <composite-element</pre>
class="com.javaegitimleri.petclinic.model.Image">
             column="FILE NAME"
type="string"/>
             column="WIDTH" type="integer"/>
             column="HEIGHT" type="integer"/>
          </composite-element>
       </set>
   </class>
</hibernate-mapping>
```





```
<hibernate-mapping>
   <class name="com.javaegitimleri.petclinic.model.Pet">
      <id name="id" column="ID">
          <generator class="sequence"/>
      </id>
       <list name="images" table="T PET IMAGES">
          <key column="PET ID"/>
          <list-index base="0" column="POSITION"/>
          <composite-element</pre>
                class="com.javaegitimleri.petclinic.model.Image">
             column="FILE_NAME" type="string"/>
             column="WIDTH" type="integer"/>
             column="HEIGHT" type="integer"/>
          </composite-element>
       </list>
   </class>
</hibernate-mapping>
```



Bag (Collection) (XML)

```
<hibernate-mapping>
   <class name="com.javaegitimleri.petclinic.model.Pet">
       <id name="id" column="ID">
          <generator class="sequence" />
       </id>
       <bag name="images" table="T_PET_IMAGES">
          <key column="PET ID" />
           <composite-element</pre>
              class="com.javaegitimleri.petclinic.model.Image">
              column="FILE NAME" type="string" />
              cproperty name="width" column="WIDTH" type="integer" />
              property name="height" column="HEIGHT" type="integer" />
          </composite-element>
       </bag>
   </class>
</hibernate-mapping>
```





```
<hibernate-mapping>
   <class name="com.javaegitimleri.petclinic.model.Pet">
       <id name="id" column="ID">
          <generator class="sequence" />
       </id>
       <map name="imagesByName" table="T PET IMAGES">
          <key column="PET ID"/>
          <map-key type="string" column="IMAGE_NAME"/>
          <composite-element</pre>
                 class="com.javaegitimleri.petclinic.model.Image">
              column="FILE_NAME" type="string" />
              column="WIDTH" type="integer" />
              operty name="height" column="HEIGHT" type="integer" />
          </composite-element>
       </map>
   </class>
</hibernate-mapping>
```

Sınıf Hiyerarşisi İçin Tek Bir Tablo (XML)



```
<hibernate-mapping>
    <class name="com.javaeqitimleri.petclinic.model.Person" table="T PERSON" abstract="true">
        <id name="id" type="long" access="field">
            <column name="ID" />
            <qenerator class="sequence" />
        </id>
       <discriminator type="string" column="P TYPE"/>
    </class>
</hibernate-mapping>
<hibernate-mapping>
    <subclass name="com.javaegitimleri.petclinic.model.Owner"</pre>
              extends="com.javaegitimleri.petclinic.model.Person"
             discriminator-value="0" lazy="false">
         property name="id" type="long">
              <column name="ID" />
         </property>
    </subclass>
</hibernate-mapping>
<hibernate-mapping>
    <subclass name="com.javaegitimleri.petclinic.model.Vet"</pre>
         extends="com.javaegitimleri.petclinic.model.Person"
         discriminator-value="V" lazy="false">
        property name="id" type="long">
            <column name="ID" />
        </subclass>
</hibernate-mapping>
```

Her Sınıf İçin Ayrı Bir Tablo (XML)



```
<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.Person" abstract="true"</pre>
table="T PERSON">
        <id name="id" type="long" access="field">
            <column name="ID" />
            <qenerator class="sequence" />
        </id>
    </class>
</hibernate-mapping>
<hibernate-mapping>
    <joined-subclass name="com.javaegitimleri.petclinic.model.Owner"</pre>
         extends="com.javaegitimleri.petclinic.model.Person" table="T OWNER" lazy="false">
         <key column="ID"/>
    </ioined-subclass>
</hibernate-mapping>
<hibernate-mapping>
    <joined-subclass name="com.javaegitimleri.petclinic.model.Vet"</pre>
         extends="com.javaegitimleri.petclinic.model.Person" table="T VET" lazy="false">
         <kev column="ID"/>
    </joined-subclass>
</hibernate-mapping>
```

Her Concrete Sınıf İçin Ayrı Bir Tablo: Implicit Yöntem (XML)

```
JAVA
Eğitimleri
```

```
<hibernate-mapping>
   <class name="com.javaegitimleri.petclinic.model.Vet" table="T VET">
      <id name="id" type="long" access="field">
        <column name="ID" />
        <qenerator class="sequence" />
     </id>
      column="FIRST NAME" type="string"/>
      column="LAST_NAME" type="string"/>
      column="GRADUATION YEAR" type="int"/>
   </class>
</hibernate-mapping>
<hibernate-mapping>
   <class name="com.javaegitimleri.petclinic.model.Owner" table="T_OWNER">
      <id name="id" type="long" access="field">
        <column name="ID" />
        <generator class="sequence" />
     </id>
      column="FIRST_NAME" type="string"/>
      column="LAST NAME" type="string"/>
      column="EMAIL" type="string"/>
   </class>
</hibernate-mapping>
```





```
<hibernate-mapping>
   <class name="com.javaegitimleri.petclinic.model.Pet">
       <id name="id" column="ID">
           <generator class="sequence" />
       </id>
       <many-to-one name="type"</pre>
       class="com.javaegitimleri.petclinic.model.PetType" not-null="true"
       lazy="proxy">
           <column name="TYPE ID" not-null="true"/>
       </many-to-one>
       <set name="visits" lazy="false">
           <key column="PET ID"/>
           <one-to-many class="com.javaegitimleri.petclinic.model.Visit"/>
       </set>
   </class>
</hibernate-mapping>
```

M:N İlişkide Join Tablosunun Bileşen İle İfade Edilmesi (XML)

```
<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.Vet" lazy="false">
        <set name="specialties" table="VETS SPECIALTIES" inverse="false"</pre>
lazy="true">
           <key column="VET ID" />
            <composite-element</pre>
class="com.javaegitimleri.petclinic.model.VetSpecialty">
               <parent name="vet" />
               <many-to-one name="specialty"</pre>
class="com.javaegitimleri.petclinic.model.Specialty"
column="SPECIALTY_ID" />
           </composite-element>
        </set>
    </class>
</hibernate-mapping>
```



Java Tipi İçeren Set (XML)

Sınıf Hiyerarşisi İçin Tek Bir Tablo: Formula (XML)



Tablo Düzeyinde Constraintler (XML)



```
column="END DATE"
index="IDX END DATE"/>
<class name="com.javaegitimleri.petclinic.model.Item" table="ITEM">
   <id name="id" column="ID"/>
   cproperty name="initialPrice" type="big_decimal"
         column="INITIAL PRICE" index="IDX INITIAL PRICE"/>
   continue = "initialPriceCurrency" type="currency"
         column="INITIAL_PRICE_CURRENCY" index="IDX_INITIAL_PRICE"/>
</class>
```

Veritabanı Düzeyinde Constraintler



İletişim



- Harezmi Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- http://www.java-egitimleri.com
- info@java-egitimleri.com



