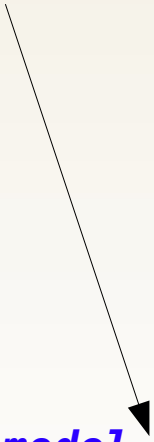
XML ile Entity İlişkileri

harezmi bilişim çözümleri

M:1 İlişki (XML)

```
<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.Pet">
    ...
        <many-to-one name="type"
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"
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.javaegitimleri.petclinic.model.Pet">
        ...
        <property name="name" type="string">
            <column name="NAME" />
        </property>
        ...
    </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"
access="field">
            <key column="OWNER_ID"/>
            <one-to-many class="com.javaegitimleri.petclinic.model.Pet" />
        </set>
    </class>
</hibernate-mapping>
```

mappedBy'ın XML'deki karşılığıdır

```
<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.Pet">
    ...
        <many-to-one name="owner"
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">
        ...
        <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">
        ...
        <many-to-one name="owner"
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"
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"
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"
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>
```

mappedBy'ın XML'deki karşılığıdır

```
<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.Pet">
    ...
        <many-to-one name="owner"
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"
access="field">
            <key column="OWNER_ID"/>
            <many-to-many class="com.javaegitimleri.petclinic.model.Pet"
unique="true" column="PET_ID"/>
        </set>
    </class>
</hibernate-mapping>
```

M:N ilişkiyi 1:M yapar

```
<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"
access="field">
            <key column="OWNER_ID"/>
            <many-to-many class="com.javaegitimleri.petclinic.model.Pet"
unique="true" column="PET_ID"/>
        </set>
    </class>
</hibernate-mapping>
```

M:N ilişkiyi 1:M yapar

```
<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"
class="com.javaegitimleri.petclinic.model.Owner" column="OWNER_ID"/>
        </join>
    </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"
lazy="true">
            <key column="VET_ID"/>
            <many-to-many class="com.javaegitimleri.petclinic.model.Specialty"
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"
lazy="true">
            <key column="VET_ID"/>
            <many-to-many class="com.javaegitimleri.petclinic.model.Specialty"
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"
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"
fetch="join">
            <column name="VET_ID" />
        </many-to-one>
        <many-to-one name="specialty" class="com.javaegitimleri.petclinic.model.Specialty"
access="field" fetch="join">
            <column name="SPECIALTY_ID" />
        </many-to-one>
    </class>
</hibernate-mapping>
```

Foreign Key Üzerinden 1:1 Unidirectional İlişki (XML)

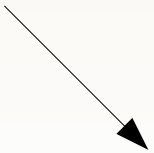
```
<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.Owner">
        ...

    </class>
</hibernate-mapping>


<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.Address">
        ...
        <many-to-one name="owner" unique="true" column="OWNER_ID"/>
    </class>
</hibernate-mapping>
```

M:1 ilişkiyi 1:1 yapar

Foreign Key Üzerinden 1:1 Bidirectional İlişki (XML)

```
<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.Owner">
        ...
        <one-to-one name="address" property-ref="owner"/>
    </class>
</hibernate-mapping>
```

```
<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.Address">
        ...
        <many-to-one name="owner" unique="true" column="OWNER_ID"/>
    </class>
</hibernate-mapping>
```



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">
            <generator class="sequence" />
        </id>
        <join table="T_OWNER_ADDRESS">
            <key column="ADDRESS_ID" unique="true" />
            <many-to-one name="owner" column="OWNER_ID" unique="true"
                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
class="com.javaegitimleri.petclinic.model.Image">
                <property name="filename" column="FILE_NAME"
type="string"/>
                <property name="width" column="WIDTH" type="integer"/>
                <property name="height" column="HEIGHT" type="integer"/>
            </composite-element>
        </set>
    </class>
</hibernate-mapping>
```

List (XML)

```
<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
                    class="com.javaegitimleri.petclinic.model.Image">
                <property name="filename" column="FILE_NAME" type="string"/>
                <property name="width" column="WIDTH" type="integer"/>
                <property name="height" 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
                class="com.javaegitimleri.petclinic.model.Image">
                <property name="filename" column="FILE_NAME" type="string" />
                <property name="width" column="WIDTH" type="integer" />
                <property name="height" column="HEIGHT" type="integer" />
            </composite-element>
        </bag>
    </class>
</hibernate-mapping>
```

Map (XML)

```
<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
                    class="com.javaegitimleri.petclinic.model.Image">
                <property name="filename" column="FILE_NAME" type="string" />
                <property name="width" column="WIDTH" type="integer" />
                <property name="height" column="HEIGHT" type="integer" />
            </composite-element>
        </map>
    </class>
</hibernate-mapping>
```

Sınıf Hiyerarşisi İçin Tek Bir Tablo (XML)

```
<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.Person" table="T_PERSON" abstract="true">
        <id name="id" type="long" access="field">
            <column name="ID" />
            <generator class="sequence" />
        </id>
        <discriminator type="string" column="P_TYPE"/>
    </class>
</hibernate-mapping>


<hibernate-mapping>
    <subclass name="com.javaegitimleri.petclinic.model.Owner"
              extends="com.javaegitimleri.petclinic.model.Person"
              discriminator-value="O" lazy="false">
        <property name="id" type="long">
            <column name="ID" />
        </property>
    </subclass>
</hibernate-mapping>


<hibernate-mapping>
    <subclass name="com.javaegitimleri.petclinic.model.Vet"
              extends="com.javaegitimleri.petclinic.model.Person"
              discriminator-value="V" lazy="false">
        <property name="id" type="long">
            <column name="ID" />
        </property>
    </subclass>
</hibernate-mapping>
```

Her Sınıf İçin Ayrı Bir Tablo (XML)

```
<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.Person" abstract="true"
table="T_PERSON">
        <id name="id" type="long" access="field">
            <column name="ID" />
            <generator class="sequence" />
        </id>
        ...
    </class>
</hibernate-mapping>

<hibernate-mapping>
    <joined-subclass name="com.javaegitimleri.petclinic.model.Owner"
        extends="com.javaegitimleri.petclinic.model.Person" table="T_OWNER" lazy="false">
        <key column="ID"/>
        ...
    </joined-subclass>
</hibernate-mapping>


<hibernate-mapping>
    <joined-subclass name="com.javaegitimleri.petclinic.model.Vet"
        extends="com.javaegitimleri.petclinic.model.Person" table="T_VET" lazy="false">
        <key column="ID"/>
        ...
    </joined-subclass>
</hibernate-mapping>
```

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

```
<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.Vet" table="T_VET">
        <id name="id" type="long" access="field">
            <column name="ID" />
            <generator class="sequence" />
        </id>
        <property name="firstName" column="FIRST_NAME" type="string"/>
        <property name="lastName" column="LAST_NAME" type="string"/>
        <property name="graduationYear" 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>
        <property name="firstName" column="FIRST_NAME" type="string"/>
        <property name="lastName" column="LAST_NAME" type="string"/>
        <property name="email" column="EMAIL" type="string"/>
    </class>
</hibernate-mapping>
```

Fetch Planı (XML)

```
<hibernate-mapping>
    <class name="com.javaegitimleri.petclinic.model.Pet">
        <id name="id" column="ID">
            <generator class="sequence" />
        </id>

        <many-to-one name="type"
        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"
lazy="true">
            <key column="VET_ID" />
            <composite-element
class="com.javaegitimleri.petclinic.model.VetSpecialty">
                <parent name="vet" />

                <many-to-one name="specialty"
class="com.javaegitimleri.petclinic.model.Specialty"
column="SPECIALTY_ID" />
                ...
            </composite-element>
        </set>
    </class>
</hibernate-mapping>
```

Java Tipi 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="PET_IMAGES">
            <key column="PET_ID"/>
            <element column="FILE_NAME" type="string" not-null="true"/>
        </set>
    </class>
</hibernate-mapping>
```

Sınıf Hiyerarşisi İçin Tek Bir Tablo: Formula (XML)

```
<hibernate-mapping>
    <class
name="com.javaegitimleri.petclinic.model.Person"
table="T_PERSON" abstract="true">
        <id name="id" type="long" access="field">
            <column name="ID" />
            <generator class="sequence" />
        </id>
        <discriminator type="string" formula="case when
GRADUATION_YEAR is not null then 'V' else 'O' end" />
    </class>
</hibernate-mapping>
```

Tablo Düzeyinde Constraintler (XML)

```
<property name="endDate" column="END_DATE"
index="IDX_END_DATE"/>
```

```
<class name="com.javaegitimleri.petclinic.model.Item" table="ITEM">
    <id name="id" column="ID"/>

    <property name="initialPrice" type="big_decimal"
          column="INITIAL_PRICE" index="IDX_INITIAL_PRICE"/>

    <property name="initialPriceCurrency" type="currency"
          column="INITIAL_PRICE_CURRENCY" index="IDX_INITIAL_PRICE"/>

    ...
</class>
```

Veritabanı Düzeyinde Constraintler

```
<class name="com.javaegitimleri.petclinic.model.Pet" table="PET">
    <id name="id" column="ID"/>
    <many-to-one name="owner" column="OWNER_ID" foreign-key="FK_PET_OWNER_ID"/>
    ...
</class>
```

```
<class name="com.javaegitimleri.petclinic.model.Vet" table="VET">
    <id name="id" column="ID"/>
    <set name="specialties" table="VETS_SPECIALTIES">
        <key column="VET_ID" foreign-key="FK_VET_ID"/>
        <many-to-many class="com.javaegitimleri.petclinic.model.Specialty"
                column="SPECIALTY_ID" foreign-key="FK_SPECIALTY_ID"/>
    </set>
</class>
```

İletişim

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