

Aufgabe 3 :

```
DROP TYPE Adresstyp FORCE;
DROP TYPE Personentyp FORCE;
DROP TYPE Vertragtyp FORCE;
DROP TYPE Spielertyp FORCE;
DROP TYPE Schiedsrichtertyp FORCE;
DROP TYPE Stadiontyp FORCE;
DROP TYPE trainiertListenTyp FORCE;
DROP TYPE Vereintyp FORCE;
DROP TYPE spieltListenTyp FORCE;
DROP TYPE leitetListenTyp FORCE;
DROP TYPE verkauftyp FORCE;
DROP TYPE Spieltyp FORCE;
DROP TYPE TV_Kanal_Typ FORCE;
DROP TYPE Mediengruppentyp FORCE;
DROP TYPE verkaufTyp FORCE;
```

```
CREATE TYPE Adresstyp AS OBJECT(
  Strasse VARCHAR(50),
  Ort VARCHAR(50),
  PLZ INTEGER
)
```

--

```
CREATE TYPE Personentyp AS OBJECT(  
  Pid    INTEGER,  
  Nachname VARCHAR(50),  
  Vorname VARCHAR(50),  
  Adresse Adresstyp  
) NOT FINAL
```

--

```
CREATE TYPE Vertragtyp AS OBJECT(  
  Vid INTEGER,  
  Startdatum VARCHAR(50),  
  enddatum  VARCHAR(50),  
  Betrag  VARCHAR(50)  
)
```

--

```
CREATE TYPE Spielertyp UNDER Personentyp(  
  Position  VARCHAR(50),  
  Geburtsdatum VARCHAR(50),  
  Größe  VARCHAR(50),  
  vertrag REF Vertragtyp  
)
```

--

```
CREATE TYPE Schiedsrichtertyp UNDER Personentyp(  
  lizenstufe VARCHAR(50)
```

```
)
```

```
--
```

```
CREATE TYPE Stadiontyp AS OBJECT(
```

```
  Stid INTEGER,
```

```
  Name VARCHAR(10),
```

```
  Kapazität VARCHAR(10),
```

```
  Adresse Adresstyp
```

```
)
```

```
--
```

```
CREATE TYPE trainiertListenTyp AS TABLE OF REF Stadiontyp
```

```
--
```

```
CREATE TYPE Vereintyp AS OBJECT(
```

```
  Vereinid INTEGER,
```

```
  Name VARCHAR(50),
```

```
  Gründungsdatum VARCHAR(50),
```

```
  Adresse Adresstyp,
```

```
  trainiert trainiertListenTyp,
```

```
  vertrag REF Vertragtyp
```

```
)
```

```
--
```

```
CREATE TYPE spieltListenTyp AS TABLE OF REF Vereintyp
```

--

```
CREATE TYPE leitetListenTyp AS TABLE OF REF Schiedsrichtertyp
```

--

```
CREATE TYPE VerkaufTyp AS OBJECT(  
    verkaufid INTEGER,  
    Datum VARCHAR(50),  
    betrag VARCHAR(10)  
)
```

--

```
CREATE TYPE SpielTyp AS OBJECT(  
    Sid INTEGER,  
    Datum VARCHAR(50),  
    Ergebnis VARCHAR(10),  
    Stadion REF Stadiontyp,  
    verkauf REF verkaufid,  
    spielt spieltListenTyp,  
    leitet leitetListenTyp  
)
```

--

```
CREATE TYPE TV_Kanal_Typ AS VARRAY(3)  
OF VARCHAR2(15);
```

--

```
CREATE TYPE Mediengruppentyp AS OBJECT(  
  Medienid INTEGER,  
  Nmae VARCHAR(50),  
  TVKanal TV_Kanal_Typ,  
  Hauptsitz VARCHAR(10),  
  verkauf REF verkaufid  
)
```

--

```
CREATE TYPE verkaufTyp AS OBJECT(  
  Medien REF Mediengruppentyp,  
  Spiel REF Spieltyp,  
  Datum ARCHAR(20),  
  Betrag VARCHAR(20)  
)
```

Aufgabe 2:

Die Klasse Person ist eine Oberklasse, diese wurde mit der Annotation `@MappedSuperclass` an dem relationalen Schema, was ermöglicht, dass es keine Entität ist und somit wird kein eigene Tabelle erstellt, sondern werden die Attribute mit der Annotation `@Column` in Spalten eingeordnet und mit der Eigenschaft „name“ gibt diese Spalten einen Namen. Die Annotation `@Id` ordnet eine Spalte als Primärschlüssel, diese wird für die Beziehungen mit anderem Entity sowie mehrwertige Attribute wichtig sein.

```
18 @MappedSuperclass
19 public class Person {
20
21     @Id
22     @GeneratedValue(strategy = GenerationType.IDENTITY)
23     @Column(name = "pid")
24     private int pid;
25
26     @Column(name = "vorname")
27     private String vorname;
28
29     @Column(name = "nachname")
30     private String nachname;
31 }
```

Die Annotation `@GeneratedValue` ermöglicht eine automatische Wertserzeugung, diese wurde für die Attribute `Id` eingesetzt, um einen eindeutigen Primärschlüssel automatisch zu erzeugen, mit der Eigenschaft `IDENTITY` wird die Generierung durch einen inkrementellen Mechanismus erfolgt.

Der Entity Spieler und Schiedsrichter sind Entity, dies wurde mit den Annotationen `@Entity` und `@Table` an dem relationalen Schema ermittelt und eine Tabelle erzeugt. Der Eigenschaft „name“ gibt diese Tabelle einen Namen.

```
12
13 @Entity
14 @Table(name = "Schiedsrichter")
15 public class Schiedsrichter extends Person {
16
17     @Column(name = "lizenzstufe")
18     private String lizenzstufe;
19
20 }
```

```
8 @Entity
9 @Table(name = "spieler")
10
11 public class Spieler extends Person {
12
13     @Column(name = "position")
14     private String position;
15
16     @Column(name = "gdatum")
17     private String gdatum;
18
19     @Column(name = "grosse")
20     private double grosse;
21 }
```

Das Attribut Adresse ist ein strukturiertes Attribut, diese wurde in einer eigenen Klasse implementiert und mit der Annotation `@Embeddable` ermöglicht die Adresse als eingebettetes Objekt kennengezeichnet und diese wird als Bestandteil einer anderen Entitäten betrachtet.

```
7 @Embeddable
8 @Table(name = "adresse")
9
10 public class Adresse {
11
12     @Column(name = "strasse")
13     private String strasse;
14
15     @Column(name = "plz")
16     private int plz;
17
18     @Column(name = "ort")
19     private String ort;
20
21 }
```

Die Annotation `@ElementCollection` wird für die mehrwertigen Attribute eingesetzt und wird die Attribute in einer Sammlung gespeichert. `@CollectionTable` wurde für die Konfiguration eingesetzt, die Eigenschaft „name“ wird die Tabelle genannt, „joinColumns“ wird die Verknüpfung zwischen der Sammlung und der übergeordneten Entität definiert. Hier wurde die Spalte der Verknüpfung

```
8 @Entity
9 @Table(name = "Rundfunkanstalten")
10 public class Rundfunkanstalten {
11
12     @Id
13     @GeneratedValue(strategy = GenerationType.IDENTITY)
14     @Column(name = "MediaId")
15     private int mediaId;
16
17     @Column(name = "Hauptsitz")
18     private String hauptsitz;
19
20     @Column(name = "Name")
21     private String name;
22
23     // die Beziehung zwischen Rund und TV
24     @ElementCollection
25     @CollectionTable(name = "Rund_TV", joinColumns = @JoinColumn(name = "Media_ID"))
26     @Column(name = "tv")
27     private Set<TV_Kanal> tv = new HashSet<TV_Kanal>();
28 }
```

angegeben, Media_ID wird als Fremdschlüssel definiert. Und @Colum mit der „name“ wird der Tabelle genannt.

