河北师范大学软件学院
Software College of Hebei Normal University

HIBERNATE

第三讲
Hibernate继承关系映射

BIG DATA Java与大数据分析

Java与移动智能设备开发

知识回顾

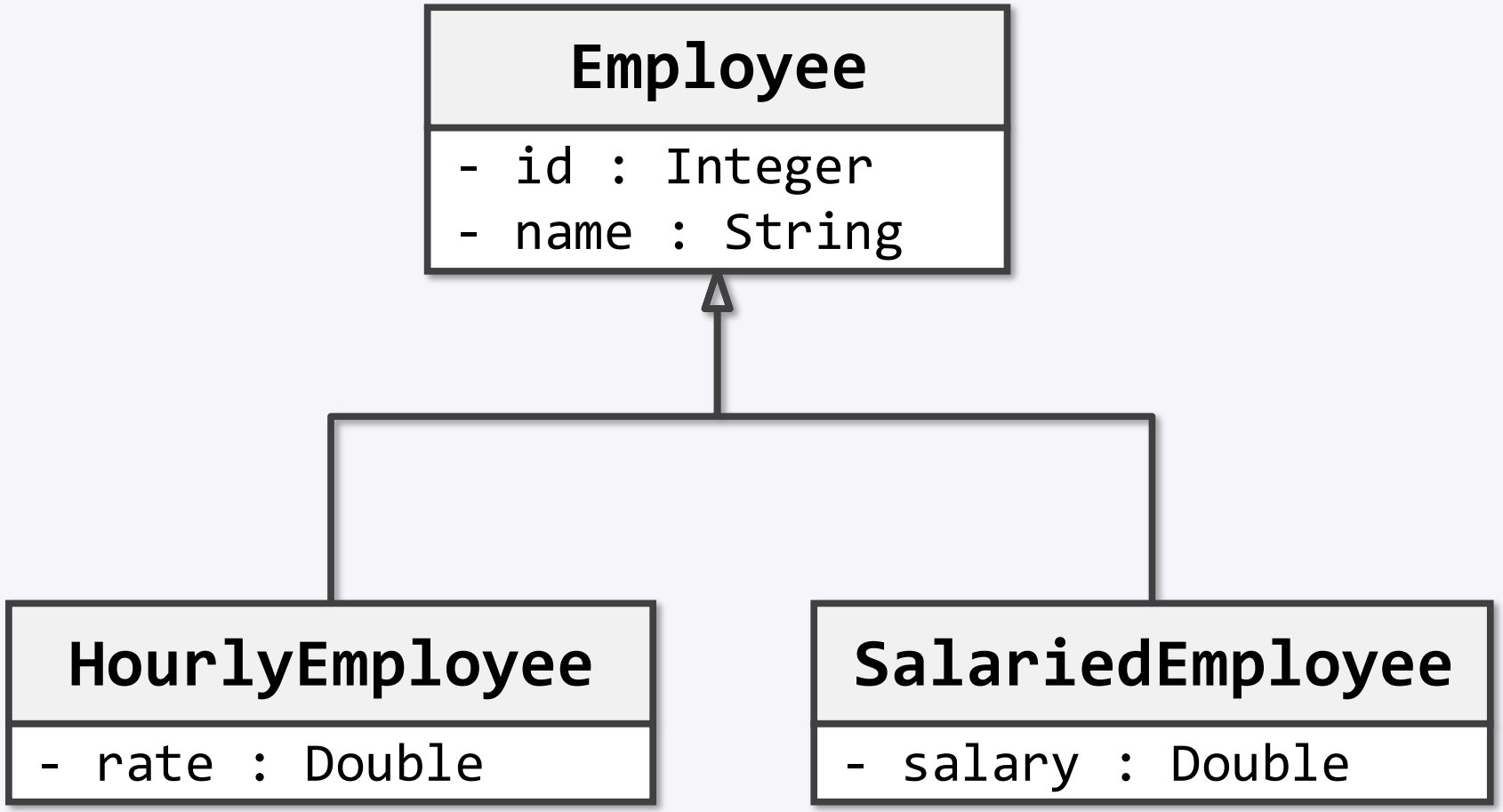
- Hibernate实体映射基础
- 属性映射
- 对象标识符映射

引例

现要为某公司开发一个员工信息管理系统，已经了解到该公司的员工中有按小时计薪和按月计薪两种方式，这种情况下系统中该如何维护员工的基本信息呢？

引例

- 包含继承关系的域模型。



目录

每个具体类对应一个表

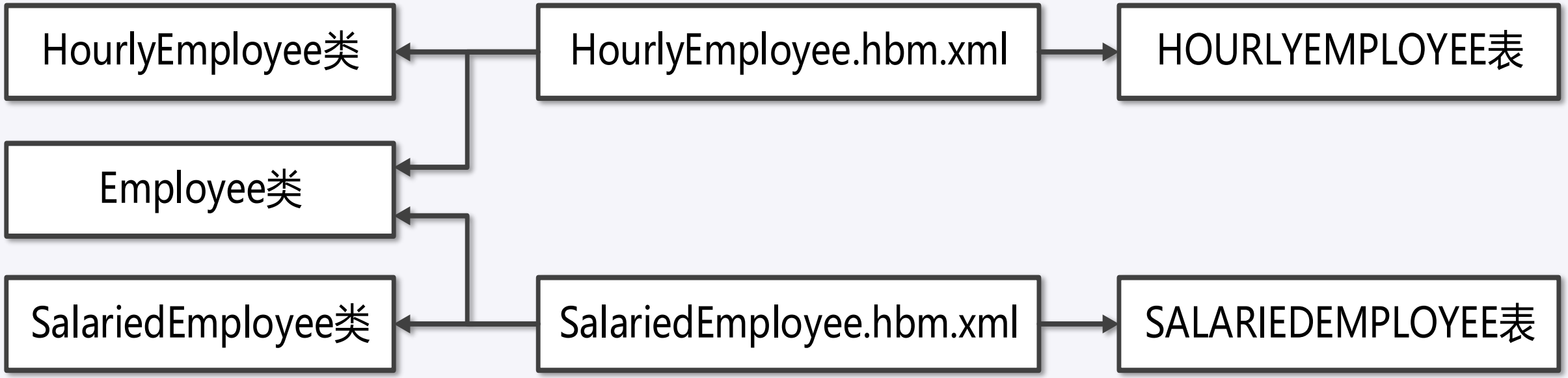
- 关系数据模型不支持域模型中的继承关系和多态。
- 每个子类所对应的表中同时存在从父类继承的属性和自己特有的属性。
- 如果父类不是抽象类并且也需要被持久化，还需要为父类创建对应的表。

数据库表结构

HOURLYEMPLOYEE表	
PK	ID
	NAME
	RATE

SALARIEDEMPLOYEE表	
PK	ID
	NAME
	SALARY

持久化类、映射文件和数据库表之间的对应关系



映射文件

```
<class name="HourlyEmployee" table="HOURLYEMPLOYEE" >
    <id name="id">
        <generator class="increment"/>
    </id>
    <property name="name"/>
    <property name="rate"/>
</class>
```

```
<class name="SalariedEmployee" table="SALARIEDEMPLOYEE" >
    <id name="id">
        <generator class="increment"/>
    </id>
    <property name="name"/>
    <property name="salary"/>
</class>
```

使用注解配置

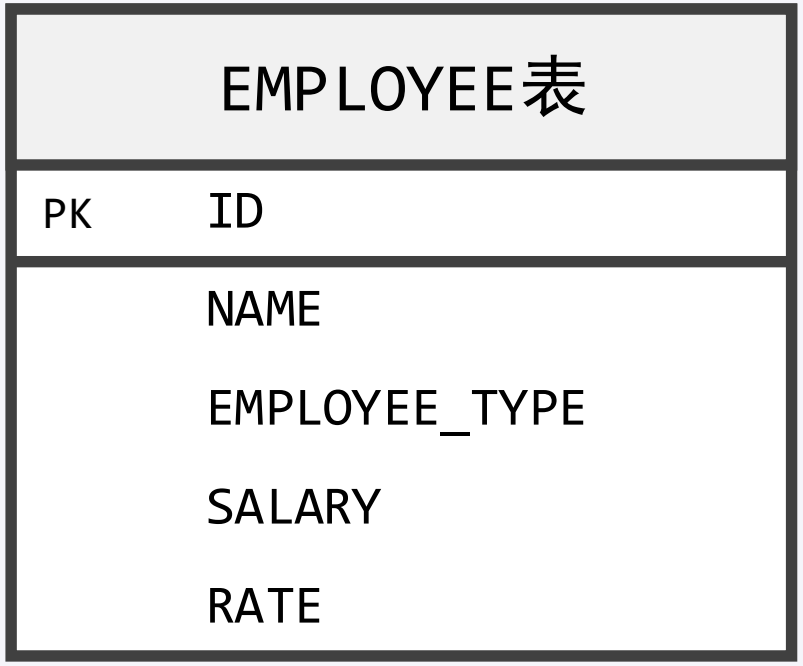
- 继承父类的一些属性，但不用父类作为映射实体时使用注解：@MappedSuperclass。

目录

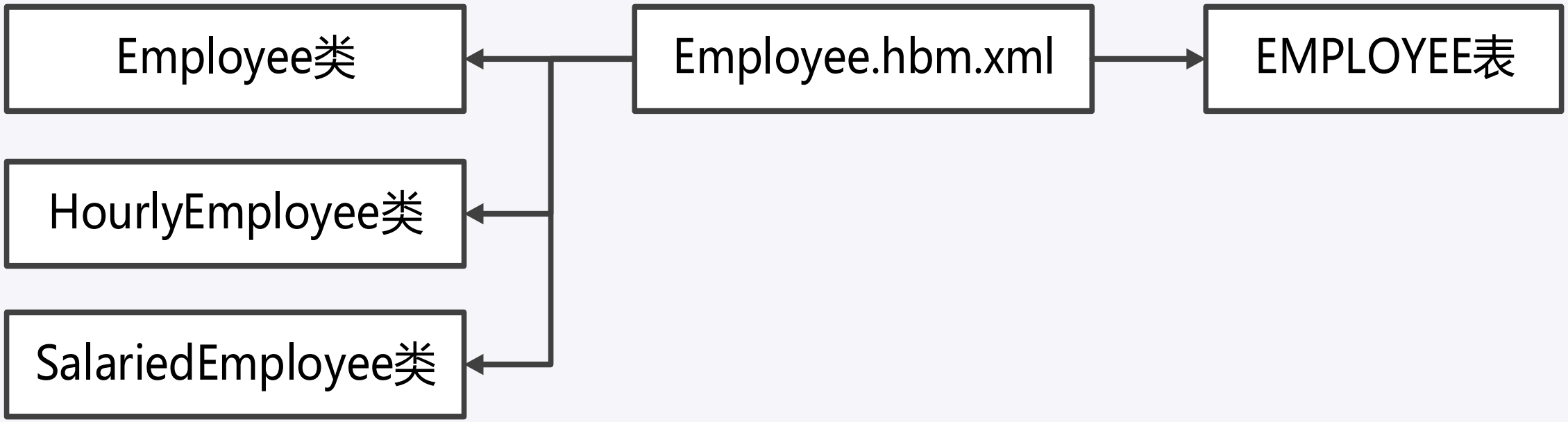
父类对应一个表

- 关系数据模型支持继承关系和多态。
- 在表中加入额外的字段区分子类的类型，表中包含父类和所有子类的属性对应的字段。
- 支持多态查询，就是从数据库中检索父类对象时，同时包含所有子类的对象。

数据库表结构



持久化类、映射文件和数据库表之间的关系



映射文件

```
<class name="Employee" table="EMPLOYEE">
    <id name="id">
        <generator class="increment"/>
    </id>
    <discriminator column="EMPLOYEETYPE" />
    <property name="name"/>
    <subclass name="HourlyEmployees" discriminator-value="HE">
        <property name="rate"/>
    </subclass>
    <subclass name="SalariedEmployees" discriminator-value="SE">
        <property name="salary"/>
    </subclass>
</class>
```

必须紧跟id元素

映射文件

- <discriminator> 元素。
  - column 属性：用于指定表中区分子类类型的字段。
- <subclass> 元素。
  - name 属性：子类类名；
  - discriminator-value 属性：子类中区分类型字段的取值；
  - <property> 子元素：映射子类属性。

映射文件

- 如果 Employees 本身也需要被持久化，可以在<class>元素中设置 discriminator-value 属性的值。

```
<class name="Employees" table="EMPLOYEES"
       discriminator-value="EE">
    ......
</class>
```

使用注解

- 父类使用注解：
  - 指定继承关系的生成策略；

```
@Inheritance(strategy=InheritanceType.SINGLE_TABLE)
```

  - 指定区分子类类型的字段。

```
@DiscriminatorColumn(name="EMPLOYEETYPE")
```

使用注解

- 各个子类使用注解：
  - 指定各个子类区分字段的值。

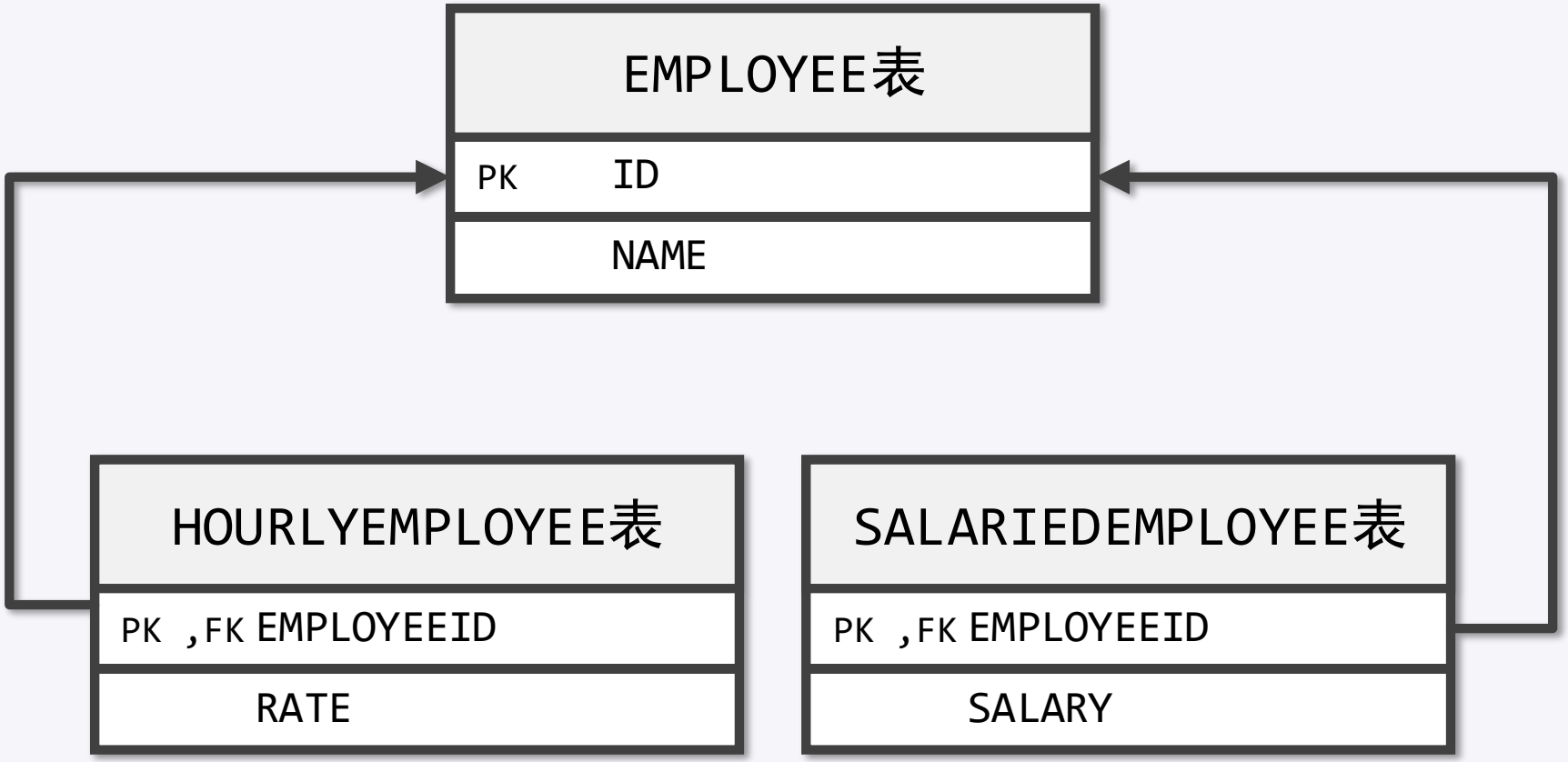
```
@DiscriminatorValue(value = "HE")
@DiscriminatorValue(value = "SE")
```

目录

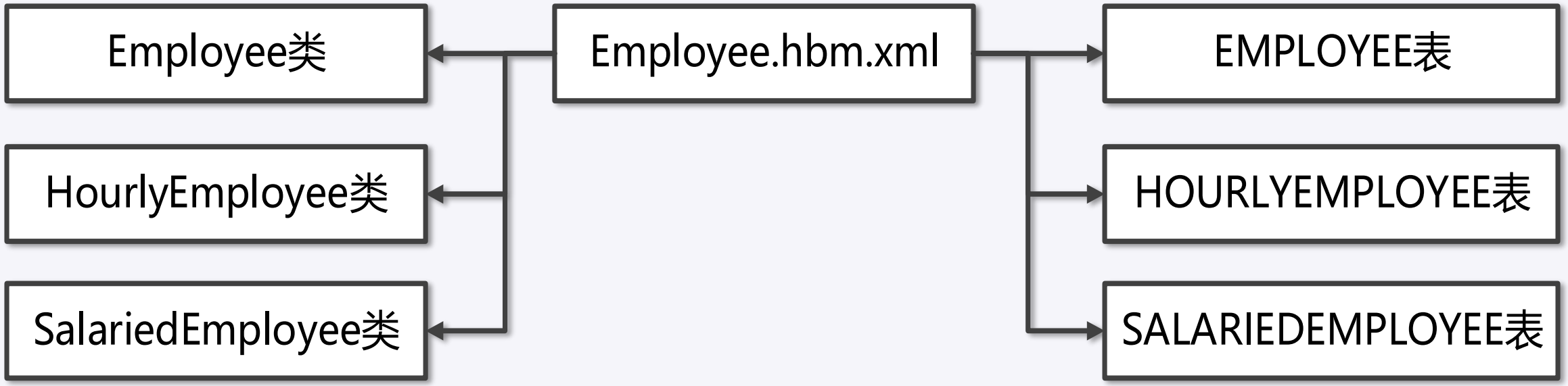
每个类对应一个表

- 在关系数据模型中，用外键参照关系来表示继承关系，子类对应的表中存在外键参照父类对应表的主键。
- 继承关系中的每个类及接口都对应一个表。
- 支持多态查询。

数据库表结构


EMPLOYEE表
PK ID
NAME
HOURLYEMPLOYEE表
PK ,FK EMPLOYEEID
RATE
SALARIEDEMPLOYEE表
PK ,FK EMPLOYEEID
SALARY

持久化类、映射文件和数据库表之间的对应关系



映射文件

```
<class name="Employee" table="EMPLOYEE">
    <id name="id">
        <generator class="increment"/>
    </id>
    <property name="name"/>
    <joined-subclass name="HourlyEmployee" table="HOURLYEMPLOYEE">
        <key column="EMPLOYEEID"/>
        <property name="rate"/>
    </joined-subclass>
    <joined-subclass name="SalariedEmployee"
            table="SALARIEDEMPLOYEE">
        <key column="EMPLOYEEID"/>
        <property name="salary"/>
    </joined-subclass>
</class>
```

映射文件

- <joined-subclass> 元素。
  - name 属性：子类类名；
  - table 属性：子类对应的数据库表；
  - <key> 子元素：指定子类对应表的主键列；
  - <property> 子元素：映射子类的属性。

使用注解

- 父类使用注解：
  - 指定继承关系的生成策略。

    @Inheritance(strategy=InheritanceType.JOINED)
- 子类使用注解：
  - 指定子类对应表的主键列。

    @PrimaryKeyJoinColumn(name="EMPLOYEEID")

目录

三种映射方式对比

- 关系数据模型的复杂度。
  - 每个具体类对应一个表：
    [缺点]：每个具体类对应的表中包含重复字段。
  - 父类对应一个表：
    [优点]：只需创建一个表；
    [缺点]：表中引入额外区分子类类型的字段。
  - 每个类对应一个表：
    [缺点]：表的数目最多，且表之间存在外键参照关系。

三种映射方式对比

- 查询性能。
  - 每个具体类对应一个表：
    [缺点]：缺点：如果要查询父类的对象，必须查询所有具体的子类对应的表。
  - 父类对应一个表：
    [优点]：有很好的查询性能，无需进行表的连接。
  - 每个类对应一个表：
    [缺点]：需要进行表的连接查询。

三种映射方式对比

- 数据库的可维护性。
  - 每个具体类对应一个表：
    [缺点]：如果父类的属性发生变化，必须修改所以子类对应的表。
  - 父类对应一个表：
    [优点]：有很好的查询性能，无需进行表的连接。
  - 每个类对应一个表：
    [缺点]：需要进行表的连接查询。

三种映射方式对比

- 是否支持多态查询。
  - 每个具体类对应一个表：不支持；
  - 父类对应一个表：支持；
  - 每个类对应一个表：支持。

本章小结

- Hibernate中实现继承映射的三种方法：
  - 每个具体类对应一个表
  - 父类对应一个表
  - 每个类对应一个表

练习

THANK YOU