# ORM解释

ORM：对象/关系映射

1. 避免在程序中编写大量SQL语句
2. 减少数据库差异对同一功能的实现
3. 避免数据库方言的不同

业务层->持久化层->数据库层

使用在持久化层

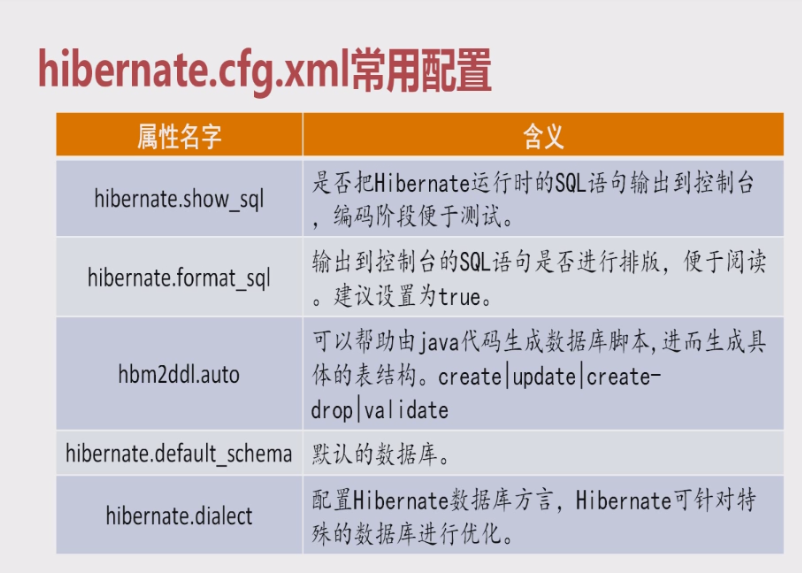
# 创建Hibernate的步骤

创建配置文件->创建持久化类->创建对象关系映射->Hibernate API编写访问数据库代码

# 持久化类的设计原则

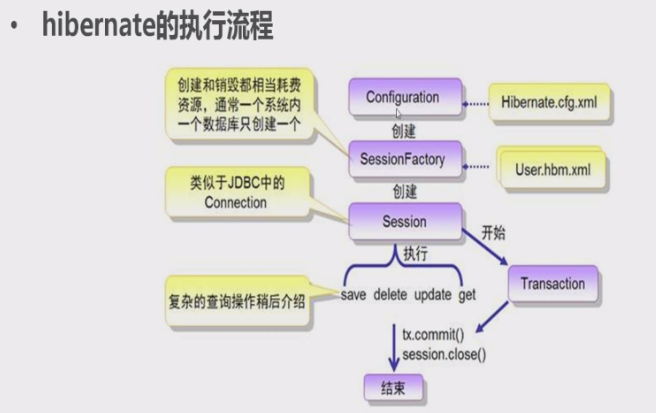
1. 共有类
2. 不带参数的构造方法
3. 属性私有
4. 提供get/set方
5. 实现系列化接口

# Hibernate.cfg.xml常用配置



作为创建表的数据库前缀

# Hibernate的执行流程



# Session简介

Session是操作数据库的对象

每个session有一个对应的connection,而一个connect不同时刻有多个session使用.

# Transaction简介

Hibernate对数据库的操作都是封装在事务当中,并且默认为非自动提交.用session保存对象时,如果不开启事务,并且手动提交事务,对象不会保存在数据库中.

除非重写Dowork(),改为自动提交事务,并强制输出SQL语句..

Session详解

如何获取session对象

opeSession()

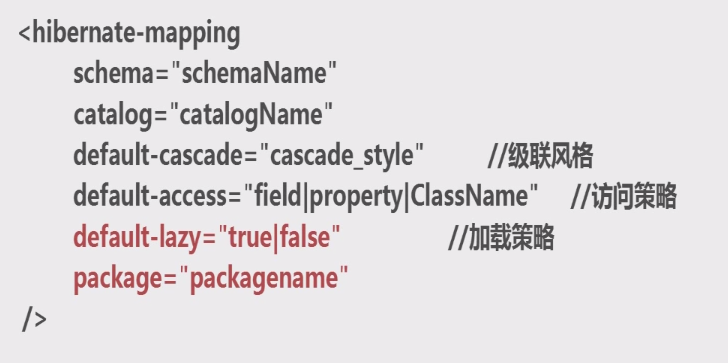
getCurrentSession()



区别:

1. getCurrentSession需要在事务提交或者回滚后session自动关闭,而openSession需要手动关闭session.如果使用openSession而没有手动关闭session,多次之后会导致连接池溢出.
2. OpenSession每次创建新的session对象,getCurrentSession使用现有的session对象.

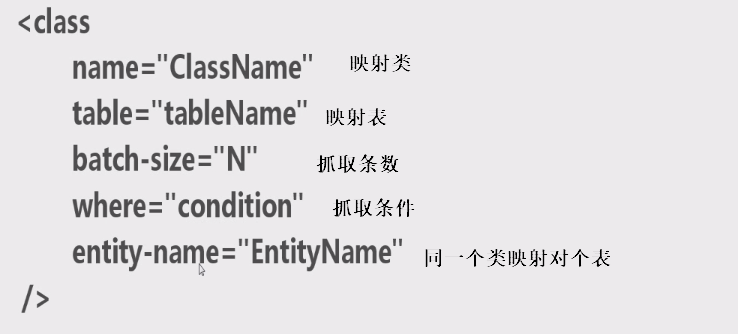
# Hbm配置文件常用设置

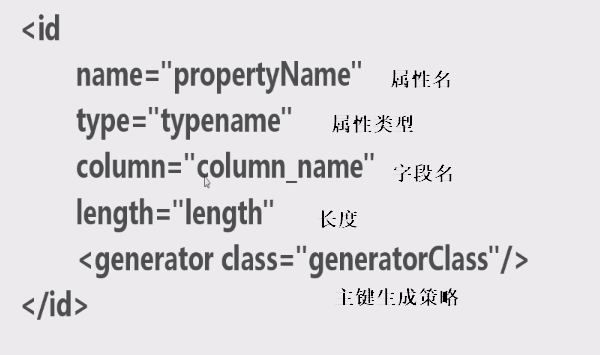


模式名称

目录名称

包名





## 常见的主键生成策略

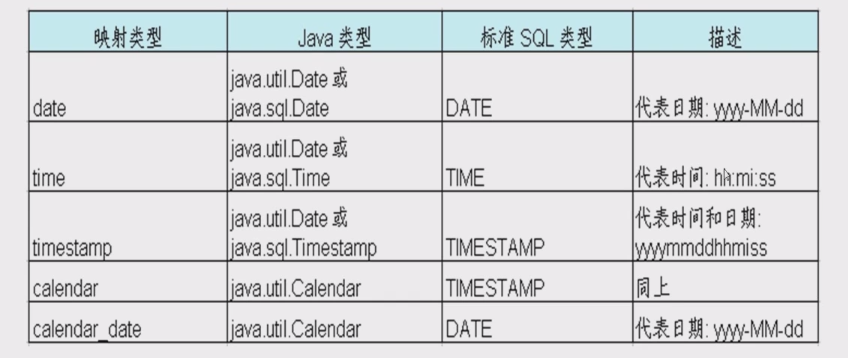


## 映射类型

可以选择Hibernate的类型或者java的类型



## 时间映射类型



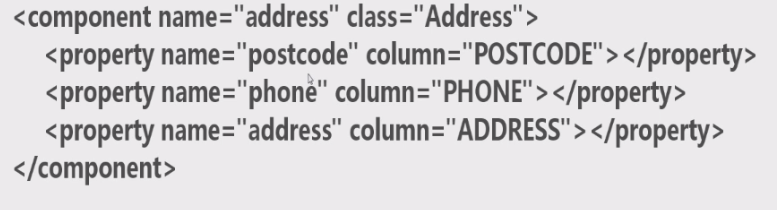
## 对象映射类型



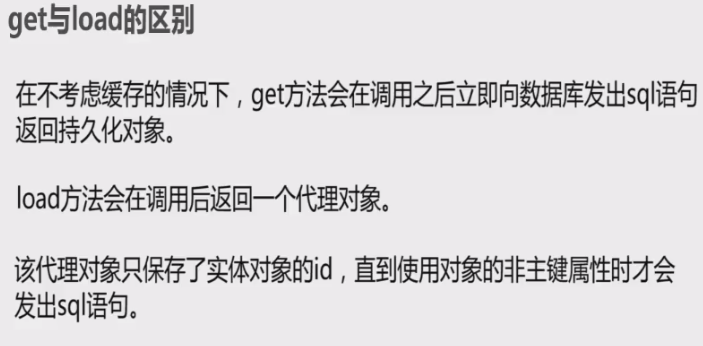
# 组件属性

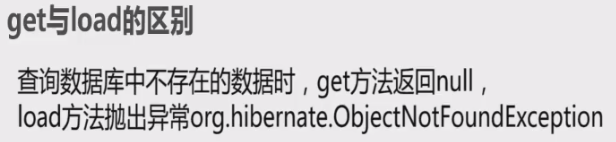
实体类中的某个属性属于用户自定义的类的对象

1. 创建组件对象
2. 引用组件对象
3. 修改映射文件



# Get与Load的区别



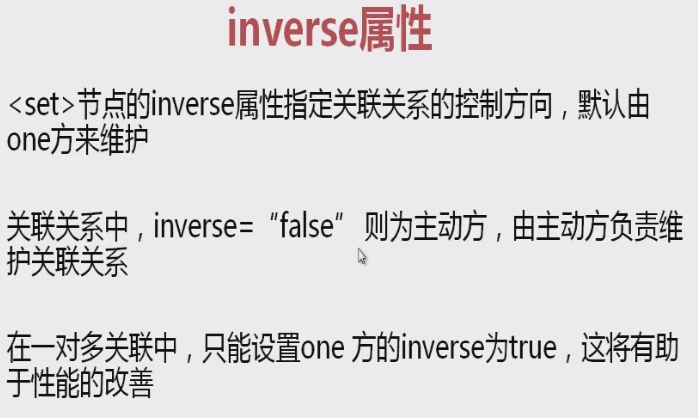


# 一对多

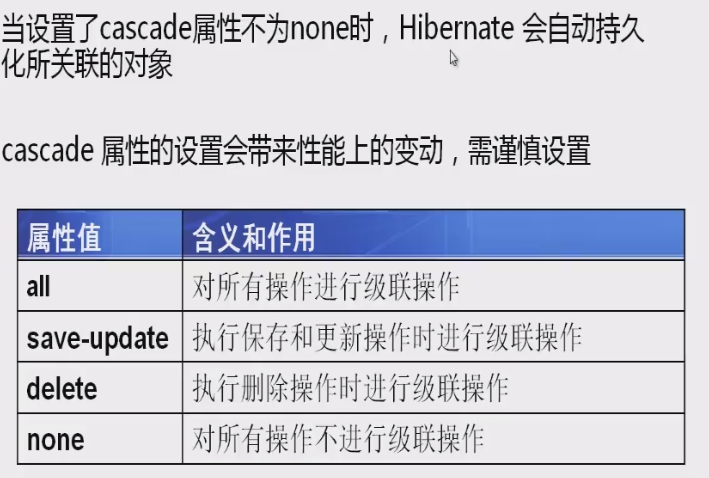
在数据库中,通过添加主外键关联,表现一对多的关系

通过在一方持有多方的集合实现.即在”一”的一端使用<set>元素表示持有多的一端的对象.





级联操作



# 多对一

多对一的关系,在自己的表的一个外键参考另一个表的主键.

通过在多方持有一方的引用实现,需要在’多’的一方使用<many-to-one>配置

1. 在多方创建引用
2. 修改映射文件
3. 保存时注意主外键关系

## 单/多双向关系

1. 在单方设置一对多的<set>和<one-to-many>
2. 在多方设置<many-to-one>
3. 在保存时设置相互引用

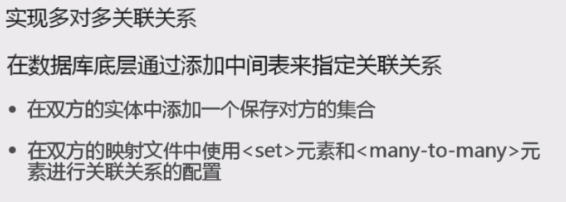
# 多对多

项目和员工之间的关系.

多对多关系一般采用中间表的形式进行实现,增加一张包含双方主键的关联表

多对多配置可以使用<set>元素和<many-to-many>元素进行配置

双方进行双向的配置



# HQL语句查询

## HQL:Hibernate的查询语言

HQL是面型对象的查询语言,查询映射配置的持久化类及其属性

SQL是面向数据库的查询语句,查询主题是数据库的表

## 形式:

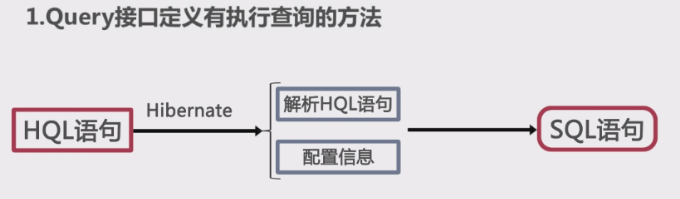
Select子句 from子句 where子句 group by子句 having子句 order by子句

注意:

1. 对类名和属性大小写敏感
2. 对关键字大小写敏感.习惯小写

## Query接口

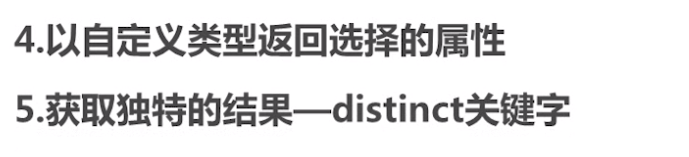
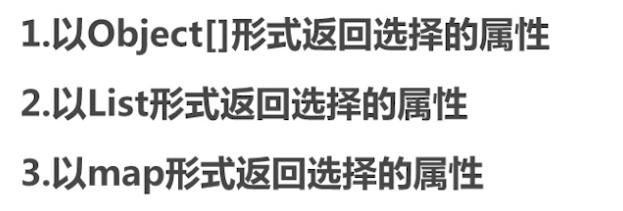
1. 定义执行查询的方法
2. 支持方法链编程风格,代码形式简单



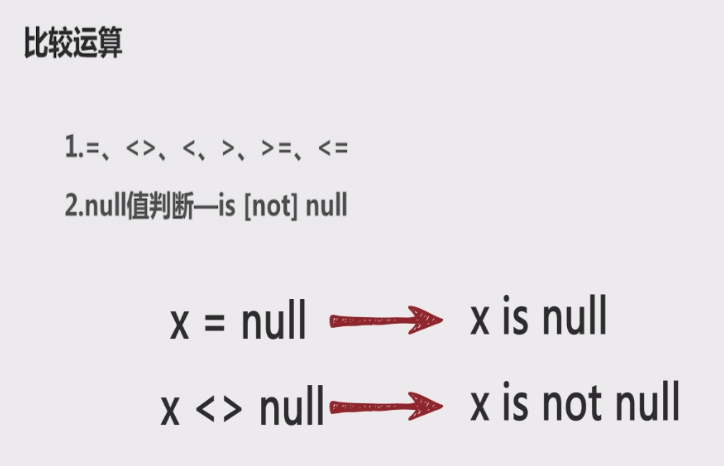
## From子句

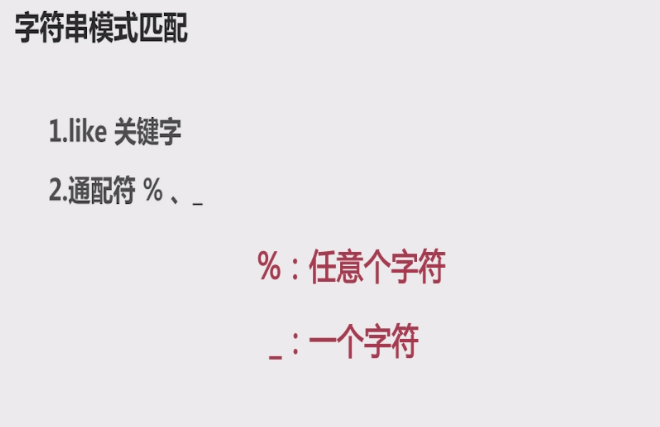


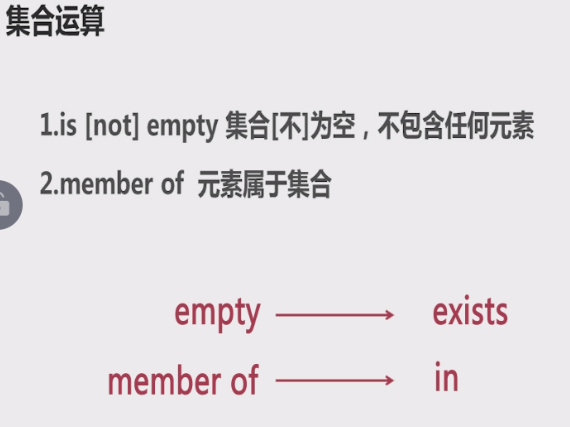
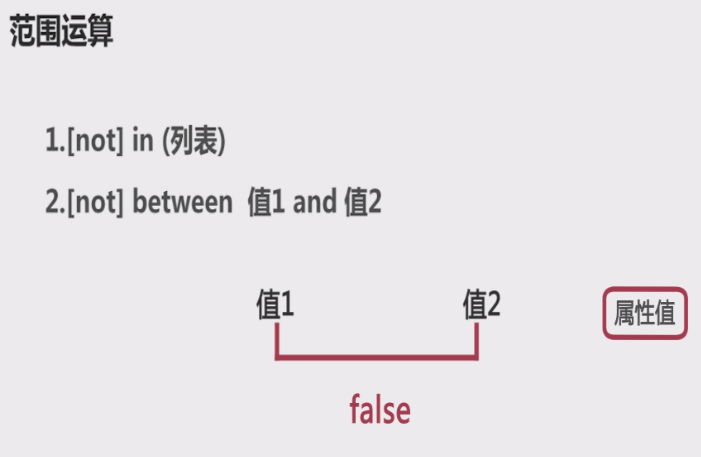
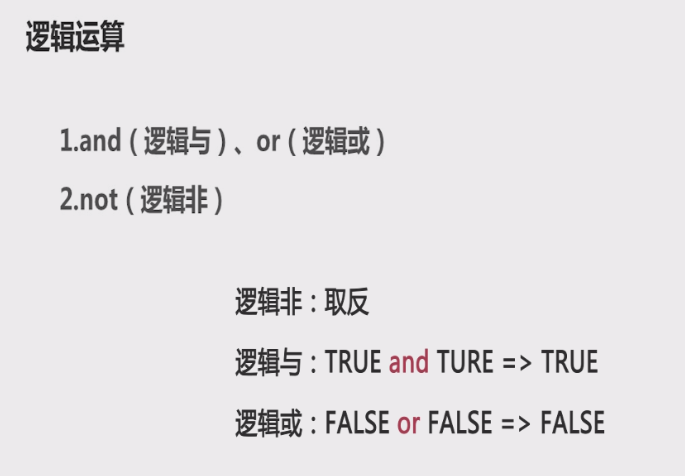
## Select子句

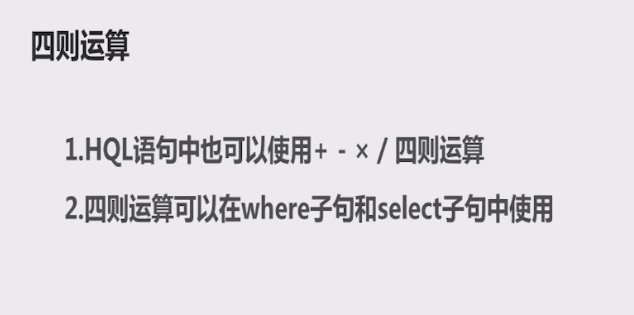


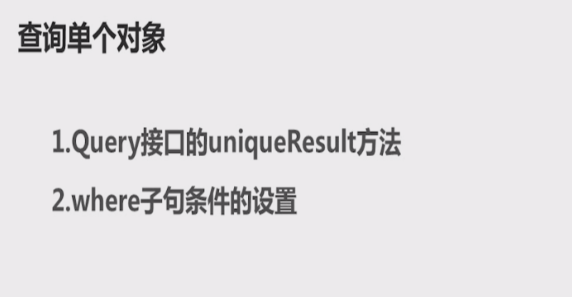
## Where子句



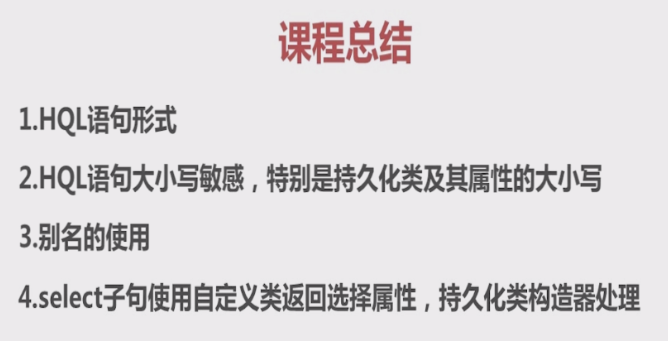






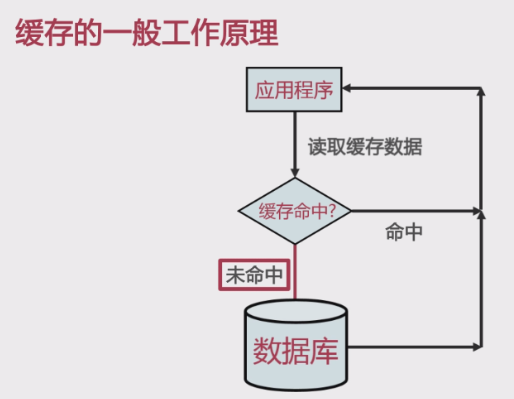




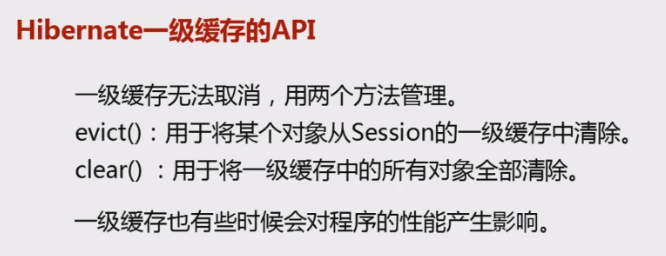
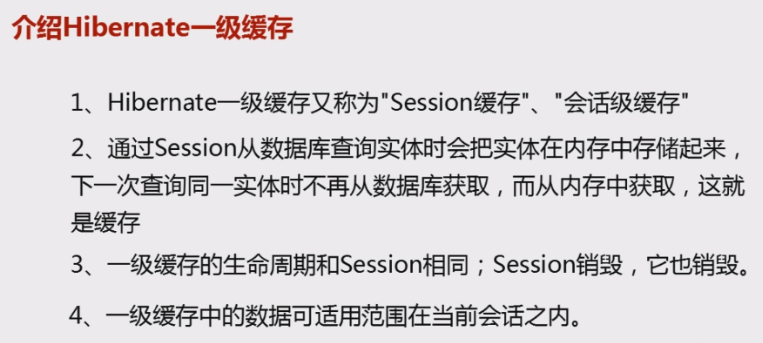


# Hibernate的缓存

为了降低应用程序对物理数据源的访问的频次.从而提高应用程序的运行性能的一种策略.



## 一级缓存

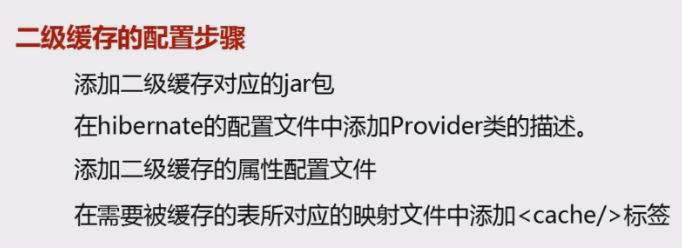


注意:一级缓存也会增删改查,增加程序负担

## 二级缓存

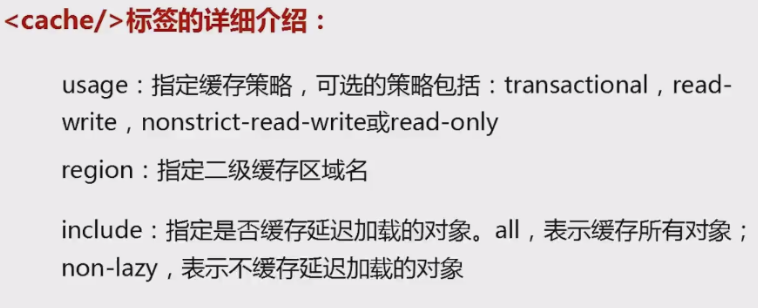
每个session共用的缓存,默认不开启,需要配置开启

二级缓存是可插拔的,默认EHCache,还有其他二级缓存组件,如Hashtable,JBoss等

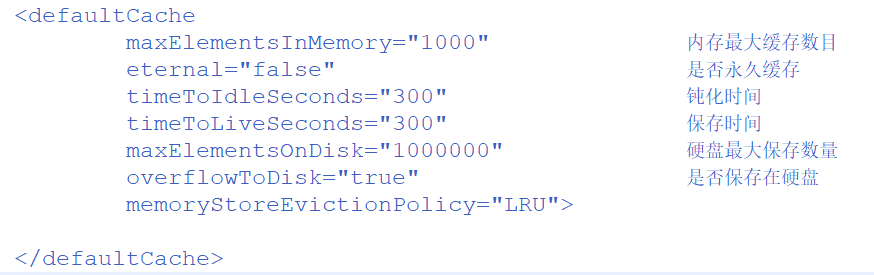


注意:二级缓存不能对二级制属性进行缓存操作

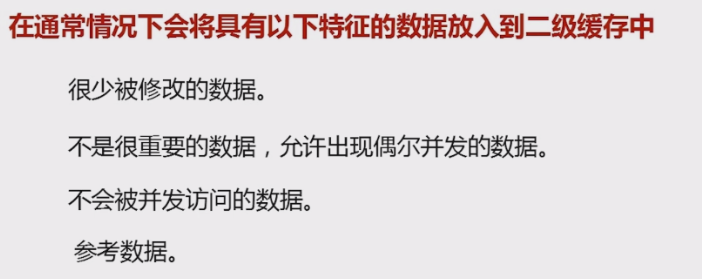
cache标签的配置



ehcache配置



适合放在二级缓存的



## 一二级缓存对比



注意:缓存策略为cache usage=*"read-only"*