



1.Hibernate查询策略

1) : session 2) : HQL

1.1 session.save(),update(),delete(),get(),load() : 此方式仅操作一条记录，不需要任何配置，默认支持二级缓存，read-only 对session是生效的，即session中如果二级缓存配置了read-only，则session.update,delete操作失败，但是save, get, load 是生效的

1.2 HQL : 此方式默认是用来操作多条记录的，list(),executeUpdate() 二级缓存的配置包括read-only 默认是无效的

注意：HQL是没有save方法的，如果需要插入数据，只能使用session.save()

2.Hibernate 缓存策略

2.1.Hibernate 一级缓存（默认存在），即session 缓存，不提升性能，用来处理事务的

2.2.Hibernate 二级缓存为sessionFactory缓存，对所有的session都有效，生命周期与sessionFactory相同 (sessionFactory 是单例的，而且项目启动的时候创建)

3.Hibernate方法行走路线

save : 直接到数据库更新，所以二级缓存配置read-only，对save无效,直接执行5

get,load : 1 --> 4 ,如果二级缓存中不存在，则 1 --> 2 --> 3 --> 4

delete,update : 首先更新二级缓存中数据，如果配置read-only,则更新和删除失败

list():查询多个，5 --> 3 --> 4 (只存不取，交给二级缓存，给get, load使用)

executeUpdate : 也是不支持二级缓存的，直接到数据库更新，Hibernate会保证数据库和缓存同步

```

<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-core</artifactId>
  <version>${hibernate.version}</version>
</dependency>

```

```

<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-ehcache</artifactId>
  <version>${hibernate.version}</version>
</dependency>

```

hibernate.cfg.xml

```

<!-- 开启二级缓存 -->
<property name="hibernate.cache.use_second_level_cache">true</property>
<!-- 指定使用哪一个缓存框架 -->
<property name="hibernate.cache.region.factory_class">org.hibernate.cache.ehcache.EhCacheRegionFactory</property>
<!-- 开启查询缓存 -->
<property name="hibernate.cache.use_query_cache">true</property>

```

<!--

maxElementsInMemory:在内存中缓存最大个数

maxElementsOnDisk:在磁盘内缓存最大个数

eternal : 是否永久保存，建议设置成false

timeToLiveSeconds : 缓存对象的生命周期，当对象存在n秒后销毁

timeToIdleSeconds:当对象闲置n秒后销毁

overflowToDisk:内存溢出，是否支持缓存到磁盘

memoryStoreEvictionPolicy : 缓存策略

FIFO:先进先出，此算法已淘汰

LRU:最近最少访问的，时间优先，会忽略访问次数的多少

LFU:最近最末使用算法，频率优先，会忽略访问的先后顺序

-->

<defaultCache

maxElementsInMemory="2"

eternal="true"

timeToIdleSeconds="60"

timeToLiveSeconds="120"

overflowToDisk="false"

memoryStoreEvictionPolicy="FIFO">

</defaultCache>