## 一级缓存（本地缓存）

sqlSession级别的缓存。一级缓存是一直开启的；SqlSession级别的一个Map

与数据库同一次会话期间查询到的数据会放在本地缓存中。

以后如果需要获取相同的数据，直接从缓存中拿，没必要再去查询数据库；

**一级缓存失效情况（没有使用到当前一级缓存的情况，效果就是，还需要再向数据库发出查询）：**

1. sqlSession不同。
2. sqlSession相同，查询条件不同.(当前一级缓存中还没有这个数据)
3. sqlSession相同，两次查询之间执行了增删改操作(这次增删改可能对当前数据有影响)
4. sqlSession相同，手动清除了一级缓存（缓存清空）

## 二级缓存（全局缓存）

基于namespace级别的缓存：一个namespace对应一个二级缓存

**工作机制：**

1. 一个会话，查询一条数据，这个数据就会被放在当前会话的一级缓存中；
2. 如果会话关闭；一级缓存中的数据会被保存到二级缓存中；新的会话查询信息，就可以参照二级缓存中的内容；
3. sqlSession===EmployeeMapper==>Employee

DepartmentMapper===>Department

不同namespace查出的数据会放在自己对应的缓存中（map）

效果：数据会从二级缓存中获取

**查出的数据都会被默认先放在一级缓存中。**

**只有会话提交或者关闭以后，一级缓存中的数据才会转移到二级缓存中**

**使用：**

1. 开启全局二级缓存配置：

**<setting name="cacheEnabled" value="true"/>**

1. 去mapper.xml中配置使用二级缓存：

**<cache></cache>**

1. **我们的POJO（数据模型）需要实现序列化接口**

**public** **class** Employee **implements** Serializable

## 和缓存有关的设置/属性

1. cacheEnabled=true：false：关闭缓存（**二级缓存关闭**）(**一级缓存一直可用的**)
2. 每个select标签都有useCache="true"：

false：不使用缓存（**一级缓存依然使用，二级缓存不使用**）

1. 每个增删改标签的：flushCache="true"：（**一级二级都会清除**）】

增删改执行完成后就会清楚缓存；

测试：flushCache="true"：一级缓存就清空了；二级也会被清除；

查询标签：flushCache="false"：

如果flushCache=true;每次查询之后都会清空缓存；缓存是没有被使用的；

1. **sqlSession.clearCache();只是清除当前session的一级缓存**；
2. localCacheScope：本地缓存作用域：（一级缓存SESSION）；**当前会话的所有数据保存在会话缓存中；**

STATEMENT：可以禁用一级缓存；

## < cache >标签详解

**<cache eviction="FIFO" flushInterval="60000" readOnly="false" size="1024">**

**</cache>**

**eviction:缓存的回收策略：**

* **LRU –** 最近最少使用的：移除最长时间不被使用的对象。
* **FIFO –** 先进先出：按对象进入缓存的顺序来移除它们。
* **SOFT –** 软引用：移除基于垃圾回收器状态和软引用规则的对象。
* **WEAK –** 弱引用：更积极地移除基于垃圾收集器状态和弱引用规则的对象。
* **默认的是 LRU。**

**flushInterval：缓存刷新间隔**

缓存多长时间清空一次，默认不清空，设置一个毫秒值

**readOnly:是否只读：**

**true**：只读；mybatis认为所有从缓存中获取数据的操作都是只读操作，不会修改数据。

mybatis为了加快获取速度，直接就会将数据在缓存中的引用交给用户。不安全，速度快

**false**：非只读：mybatis觉得获取的数据可能会被修改。

mybatis会利用序列化&反序列的技术克隆一份新的数据给你。安全，速度慢

**size：缓存存放多少元素；**

**type=""：指定自定义缓存的全类名；**

实现Cache接口即可；

## 第三方缓存整合

* + - 1. 导入第三方缓存包即可（或者使用Maven方式）；

**ehcache-core缓存技术的核心包**

<dependency>

<groupId>net.sf.ehcache</groupId>

<artifactId>ehcache-core</artifactId>

<version>2.6.11</version>

</dependency>

**日志包，可以不导**

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-log4j12</artifactId>

<version>1.7.25</version>

<scope>test</scope>

</dependency>

* + - 1. 导入与第三方缓存整合的适配包；

**MyBatis整合ehcache-core和适配器包**

<dependency>

<groupId>org.mybatis.caches</groupId>

<artifactId>mybatis-ehcache</artifactId>

<version>1.1.0</version>

</dependency>

* + - 1. mapper.xml中使用自定义缓存

**<cache type="org.mybatis.caches.ehcache.EhcacheCache"></cache>**

* + - 1. ehcache需要在类路径下又一个xml配置文件，笔记中有，复制即可
      2. **可以引用其他命名空间文件的缓存：**

<cache-ref namespace=*"需要引用的命名空间"*/>