**基础知识**

SqlSessionFactoryBuilder则可以从XML配置文件或一个预先定制的Configuration的实例构建出SqlSessionFactory的实例。每一个MyBatis的应用程序都以一个SqlSessionFactory对象的实例为核心。同时SqlSessionFactory也是线程安全的，SqlSessionFactory一旦被创建，应该在应用执行期间都存在。在应用运行期间不要重复创建多次,建议使用单例模式。SqlSessionFactory是创建SqlSession的工厂，SqlSession是MyBatis的关键对象，是执行持久化操作的独享。

**一级缓存**

MyBatis 默认开启了一级缓存，一级缓存是在SqlSession 层面进行缓存的，基于 HashMap 的本地缓存。同一个SqlSession ，多次调用同一个mapper和同一个方法的同一个参数，只会进行一次数据库查询，然后把数据缓存到缓冲中，以后直接先从缓存中取出数据，不会直接去查数据库。

**二级缓存**

二级缓存是 mapper 级别的缓存，同样是基于 HashMap 进行存储，多个 SqlSession 可以共用二级缓存，其作用域是 mapper 的同一个 namespace。不同的 SqlSession 两次执行相同的 namespace 下的 sql 语句，会执行相同的 sql，第二次查询只会查询第一次查询时读取数据库后写到缓存的数据，不会再去数据库查询。

**启用二级缓存**

1、<!—mybatis.xml中启用二级缓存 -->

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

2、<!—mapper.xml中启用二级缓存 -->

<cache eviction="FIFO" flushInterval="60000"readOnly="true" size="512"/>

type：cache使用的类型，默认是PerpetualCache。

eviction：定义回收的策略，常见的有FIFO，LRU。

flushInterval：配置一定时间自动刷新缓存，单位是毫秒

size：最多缓存对象的个数

readOnly：是否只读，若配置可读写，则需要对应的实体类能够序列化。

Blocking：若缓存中找不到对应的key，是否会一直blocking，直到有对应的数据进入缓存。

3、手动测试类测试时需要

sqlSession.commit();

注：也可以在mapper上添加注解@CacheNamespace开启二级缓存，但是要注意：配置文件和接口注释是不能够配合使用的。只能通过全注解的方式或者全部通过xml配置文件的方式使用

**改用外置二级缓存组件**

1、实体类必须实现可序列化接口

2、更改cache标签

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

3、让二级缓存失效情况（默认useCache是true）

<selectid="getStu" resultType="Student" useCache="false">

**二级缓存脏读数据**

mybatis的二级缓存不适应用于映射文件中存在多表查询的情况。如果存在涉及多个表的查询的话，由于Mybatis的二级缓存是基于namespace的，多表查询语句所在的namspace无法感应到其他namespace中的语句对多表查询中涉及的表进行了修改，引发脏数据问题。

**springboot基于redis实现mybatis二级缓存**

1、pom中添加依赖，spring-boot-starter-data-redis

2、配置文件配置redis基础信息、开启mybatis二级缓存

mybatis-plus.configuration.cache-enabled=true

3、编写redis配置类，主要向spring容器中注入RedisTemplate

4、编写缓存类实现mybatis的cache接口，在该类中手动注入RedisTemplate