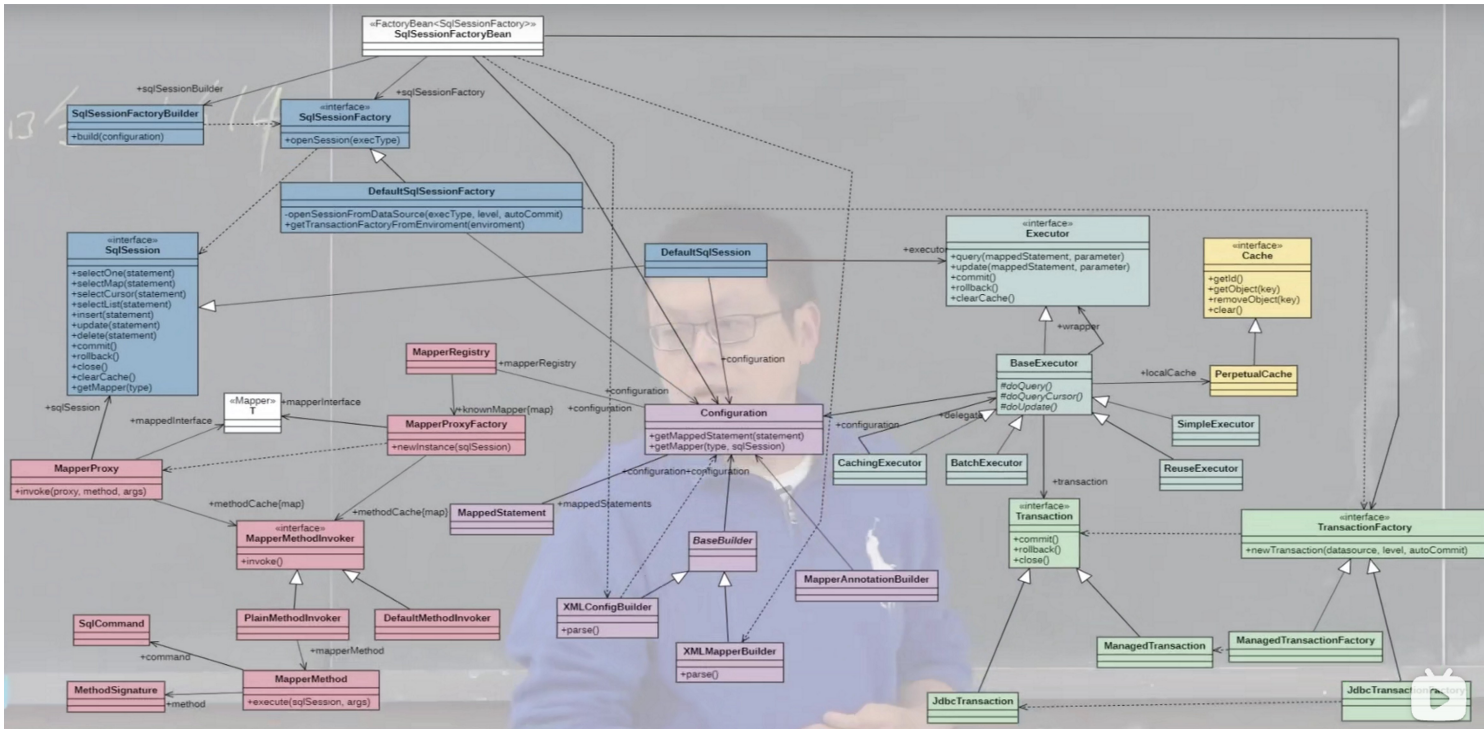


Executor接口

2022年1月1日 23:56



右侧浅蓝色的部分是Executor部分

我们在执行一个SqlSession的命令的时候，比如一个insert命令，它的过程并不是在SqlSession中进行的，而是靠放进去的Executor，Executor是SqlSession的一个属性

不自己执行的原因是，在执行时它有多种选择，所以我们可以看到Executor有多个子类，是它的各种执行方式：

最简单的是SimpleExecutor，就是给它什么就执行什么。SimpleExecutor是实现了本地一级缓存的，这里的一级缓存做在BaseExecutor上，可以看到BaseExecutor有一个LocalCache。所以不管是什么执行方式，它都有一级缓存。缓存还有很多其他类，这里为了方便并没有全部都画出来。所以说，一级缓存是关不掉的，因为一级缓存是做在BaseExecutor上的，他都会试图把同一次执行过程中查过的数据放到localCache里，详情见[一级缓存具体实现](#)

另一个是ReuseExecutor，它可以做到会话级（Session）的缓存，就是在一个Session中间，如果有查过某个数据，就会把这个数据缓存下来。注意，这里并不是把数据的结果缓存下来，而是把预编译过的语句缓存下来（因为所有的执行语句都是需要预编译之后才能使用的，而这个预编译是需要时间的）。这样之后就可以直接拿编译好的语句去执行。

BatchExecutor是MyBatis中用来做批量处理的，因为我们在一个SQL语句中是可以执行多条SQL命令的，这个时候就用BatchExecutor来执行

CachingExecutor是用来实现二级缓存的，可以看到它有一个属性是delegate。我们可以把SimpleExecutor、ReuseExecutor、BatchExecutor这些东西放到CachingExecutor里，这样就可以在这些基础的查询上再加上二级缓存。二级缓存具体见[二级缓存具体实现](#)

CachingExecutor的设计是一种代理设计模式，也就是说，我们并不是直接使用Executor，而是在三种基础的Executor上做了一个代理的CachingExecutor，因此它会有一个属性叫delegate。它在它代理的Executor基础上做了一些增强，即实现了二级缓存

这里使用GRASP的原则来分析，可以知道CachingExecutor这样设计的目的是不想让代码的其他部分知道这里有使用了一个代理，即是否使用二级缓存、是否使用CachingExecutor，对代码的其他部分都是没有影响的。要做到这一点最主要的手段就是降低它与其他部分的耦合，降低耦合的一个重要手段就是使用多态。所以可以看到SqlSession中放的是一个Executor对象，它可能是Simple、可能是Batch、可能是Reuse、可能是Caching，但SqlSession不需要关心具体是什么对象。CachingExecutor和他代理的对象都是一致的，因为他们都实现了同样的接口，所以对于SqlSession来说，代理或不代理都是一样的。