

Hibernate的检索策略小结 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/252/2021\\_2022\\_Hibernate\\_E7\\_c104\\_252307.htm](https://www.100test.com/kao_ti2020/252/2021_2022_Hibernate_E7_c104_252307.htm) Hibernate的检索策略包括类级别检索策略和

关联级别检索策略。类级别检索策略有立即检索和延迟检索，默认的检索策略是立即检索。在Hibernate映射文件中，通过在上配置lazy属性来确定检索策略。对于Session的检索方式，类级别检索策略仅适用于load方法；也就是说，对于get、query检索，持久化对象都会被立即加载而不管lazy是false还是true。一般来说，我们检索对象就是要访问它，因此立即检索是通常的选择。由于load方法在检索不到对象时会抛出异常（立即检索的情况下），因此我个人并不建议使用load检索；而由于中的lazy属性还影响到多对一及一对一的检索策略，因此使用load方法就更没必要了。关联级别检索策略有立即检索、延迟检索和迫切左外连接检索。对于关联级别检索，又可分为一对多和多对多、多对一和一对一两种情况讨论。一对多和多对多关联关系一般使用配置。有lazy和outer-join属性，它们的不同取值绝对了检索策略。1) 立即检索：这是一对多默认的检索策略，此时lazy=false，outer-join=false。尽管这是默认的检索策略，但如果关联的集合是无用的，那么就不要再使用这种检索方式。2) 延迟检索：此时lazy=true，outer-join=false（outer-join=true是无意义的），这是优先考虑的检索方式。3) 迫切左外连接检索：此时lazy=false，outer-join=true，这种检索策略只适用于依靠id检索方式（load、get），而不适用于query的集合检索（它会采用立即检索策略）。相比于立即检索，这种检索策略减少

了一条sql语句，但在Hibernate中，只能有一个配置成 `outer-join=true`. 多对一和一对一检索策略一般使用、配置。中需要配置的属性是 `outer-join`，同时还需要配置one端关联的 `lazy` 属性（配置的可不是中的 `lazy` 哦），它们的组合后的检索策略如下：

- 1) `outer-join=auto`：这是默认值，如果 `lazy=true` 为延迟检索，如果 `lazy=false` 为迫切左外连接检索。
- 2) `outer-join=true`，无关于 `lazy`，都为迫切左外连接检索。
- 3) `outer-join=false`，如果 `lazy=true` 为延迟检索，否则为立即检索。

可以看到，在默认的情况下（`outer-join=auto`，`lazy=false`），对关联的one端对象Hibernate采用的迫切左外连接检索。依我看，很多情况下，我们并不需要加载one端关联的对象（很可能我们需要的仅仅是关联对象的id）；另外，如果关联对象也采用了迫切左外连接检索，就会出现 `0select` 语句中有多个外连接表，如果个数多的话会影响检索性能，这也是为什么Hibernate通过 `hibernate.max_fetch_depth` 属性来控制外连接的深度。对于迫切左外连接检索，`query` 的集合检索并不适用，它会采用立即检索策略。对于检索策略，需要根据实际情况进行选择。对于立即检索和延迟检索，它们的优点在于 `0select` 语句简单（每张表一条语句）、查询速度快，缺点在于关联表时需要多条 `0select` 语句，增加了访问数据库的频率。因此在选择即检索和延迟检索时，可以考虑使用批量检索策略来减少 `0select` 语句的数量（配置 `batch-size` 属性）。对于切左外连接检索，优点在于 `0select` 较少，但缺点是 `0select` 语句的复杂度提高，多表之间的关联会是很耗时的操作。另外，配置文件是死的，但程序是活的，可以根据需要在程序里显示的指定检索策略（可能经常需要在程序中显示指定迫切左外连

接检索)。为了清楚检索策略的配置效果如何，可以配置show\_sql属性查看程序运行时Hibernate执行的sql语句。

100Test 下载频道开通，各类考试题目直接下载。详细请访问  
[www.100test.com](http://www.100test.com)