# Hibernate 和 Mabtis 的区别

## 简单简介

### Hibernate 框架

**Hibernate**是一个开放源代码的对象关系映射框架,它对JDBC进行了非常轻量级的对象封装,建立对象与数据库表的映射。是一个全自动的、完全面向对象的持久层框架。

### Mybatis框架

**Mybatis**是一个开源对象关系映射框架，原名：ibatis,2010年由谷歌接管以后更名。是一个半自动化的持久层框架。

## 两者区别

### 1.开发方面

在项目开发过程当中，就速度而言：

            hibernate开发中，sql语句已经被封装，直接可以使用，加快系统开发；

            Mybatis 属于半自动化，sql需要手工完成，稍微繁琐；

但是，凡事都不是绝对的，如果对于庞大复杂的系统项目来说，发杂语句较多，选择hibernate 就不是一个好方案。

### 2.sql优化方面

Hibernate 自动生成sql,有些语句较为繁琐，会多消耗一些性能；

        Mybatis 手动编写sql，可以避免不需要的查询，提高系统性能；

### 3.对象管理比对

        Hibernate 是完整的对象-关系映射的框架，开发工程中，无需过多关注底层实现，只要去管理对象即可；

        Mybatis 需要自行管理 映射关系；

### 4.缓存方面

|  |
| --- |
| 相同点： Hibernate和Mybatis的二级缓存除了采用系统默认的缓存机制外，都可以通过实现你自己的缓存或为其他第三方缓    存方案，创建适配器来完全覆盖缓存行为。  **不同点：**  **Hibernate的二级缓存配置在SessionFactory生成的配置文件中进行详细配置，然后再在具体的表-对象映射中配置是那种缓存。**  MyBatis的二级缓存配置都是在每个具体的表-对象映射中进行详细配置，这样针对不同的表可以自定义不同的缓存机制。并且Mybatis可以在命名空间中共享相同的缓存配置和实例，通过Cache-ref来实现。 |

**比较：**

**Hibernate**具有良好的管理机制，用户不需要关注SQL，如果二级缓存出现脏数据，系统会保存，；

           Mybatis 在使用的时候要谨慎，避免缓存CAche 的使用。

## Hibernate优势

1. Hibernate的DAO层开发比MyBatis简单，Mybatis需要维护SQL和结果映射。
2. Hibernate对对象的维护和缓存要比MyBatis好，对增删改查的对象的维护要方便。
3. Hibernate数据库移植性很好，MyBatis的数据库移植性不好，不同的数据库需要写不同SQL。
4. Hibernate有更好的二级缓存机制，可以使用第三方缓存。MyBatis本身提供的缓存机制不佳。

## Mybatis优势

1. MyBatis可以进行更为细致的SQL优化，可以减少查询字段。
2. MyBatis容易掌握，而Hibernate门槛较高。

## 一句话总结

**Mybatis**：小巧、方便、高效、简单、直接、半自动化

**Hibernate**：强大、方便、高效、复杂、间接、全自动化

# 数据库的索引

# 内连接和外连接

1、内联接（典型的联接运算，使用像 =  或 <> 之类的比较运算符）。包括相等联接和自然联接。       
内联接使用比较运算符根据每个表共有的列的值匹配两个表中的行。例如，检索 students和courses表中学生标识号相同的所有行。     
**2、外联接。**外联接可以是左向外联接、右向外联接或完整外部联接。       
在 FROM子句中指定外联接时，可以由下列几组关键字中的一组指定：       
1）LEFT  JOIN或LEFT OUTER JOIN       
左向外联接的结果集包括  LEFT OUTER子句中指定的左表的所有行，而不仅仅是联接列所匹配的行。如果左表的某行在右表中没有匹配行，则在相关联的结果集行中右表的所有选择列表列均为空值。         
2）RIGHT  JOIN 或 RIGHT  OUTER  JOIN       
右向外联接是左向外联接的反向联接。将返回右表的所有行。如果右表的某行在左表中没有匹配行，则将为左表返回空值。         
3）FULL  JOIN 或 FULL OUTER JOIN  
完整外部联接返回左表和右表中的所有行。当某行在另一个表中没有匹配行时，则另一个表的选择列表列包含空值。如果表之间有匹配行，则整个结果集行包含基表的数据值。

# MySQL分页 Oracle分页

## 1.Mysql使用limit分页

[?](https://www.jb51.net/article/123002.htm)

|  |  |
| --- | --- |
| 1 | select \* from stu limit m, n; //m = (startPage-1)\*pageSize,n = pageSize |

PS:

（1）第一个参数值m表示起始行，第二个参数表示取多少行（页面大小）

（2）m= (2-1)\*10+1,n=10 ，表示 limit 11,10从11行开始，取10行，即第2页数据。

（3）m、n参数值不能在语句当中写计算表达式，写到语句之前必须计算好值。

## 2.Oracle使用rownum分页

[?](https://www.jb51.net/article/123002.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5 | select \* from (  select rownum rn,a.\* from table\_name a where rownum <= x  //结束行，x = startPage\*pageSize  )  where rn >= y; //起始行，y = (startPage-1)\*pageSize+1 |

PS:

（1）>= y，<= x表示从第y行（起始行）~x行（结束行） 。

（2）rownum只能比较小于，不能比较大于，因为rownum是先查询后排序的，例如你的条件为rownum>1，当查询到第一条数据，rownum为1，则不符合条件。第2、3...类似，一直不符合条件，所以一直没有返回结果。所以查询的时候需要设置别名，然后查询完成之后再通过调用别名进行大于的判断。

# 数据库分组 group by

group by语法可以根据给定数据列的每个成员对查询结果进行**分组统计**，最终得到一个分组汇总表。

SELECT子句中的列名必须为分组列或列函数。列函数对于GROUP BY子句定义的**每个组各返回一个结果**。

Group by 一般和聚合函数一起使用才有意义,比如 count sum avg等,使用group by的两个要素:   
**(1) 出现在select后面的字段 要么是聚合函数中的,要么是group by 中的.   
 (2) 要筛选结果 可以先使用where 再用group by 或者先用group by 再用having**

# cookie和session的区别和用法

  1、Cookie和Session都是会话技术，Cookie是运行在客户端，Session是运行在服务器端。  
     
   2、Cookie有大小限制以及浏览器在存cookie的个数也有限制，Session是没有大小限制和服务器的内存大小有关。

   3、Cookie有安全隐患，通过拦截或本地文件找得到你的cookie后可以进行攻击。

   4、Session是保存在服务器端上会存在一段时间才会消失，如果session过多会增加服务器的压力。