

## 缓存笔记

笔记本： mybatis  
创建时间： 2019/1/14 20:10 更新时间： 2019/4/8 20:37  
作者： qmys2018@163.com  
URL： <https://www.cnblogs.com/xdp-gacl/p/4270403.html>

一级缓存：

1. **一级缓存**：基于PerpetualCache 的 HashMap本地缓存，其**存储作用域为 Session**，当 **Session flush 或 close** 之后，该**Session中的所有 Cache 就将清空**。

```
List<User> list=user.findAll();
for(User u:list){
    System.out.println("用户名: "+u.getUsername()+"部门名:"+u.getDepartment().getDname());
}

List<User> list1=user.findAll();
for(User u:list1){
    System.out.println("用户名: "+u.getUsername()+"部门名:"+u.getDepartment().getDname());
}
```

只执行一次sql

```
List<User> list=user.findAll();
for(User u:list){
    System.out.println("用户名: "+u.getUsername()+"部门名:"+u.getDepartment().getDname());
}
System.out.println("-----");
session.clearCache();//清空缓存
List<User> list1=user.findAll();
for(User u:list1){
    System.out.println("用户名: "+u.getUsername()+"部门名:"+u.getDepartment().getDname());
}
```

执行两次sql

二级缓存：

**二级缓存**与一级缓存其机制相同，默认也是采用 PerpetualCache，HashMap存储，不同在于其**存储作用域为 Mapper(Namespace)**，并且**可自定义存储源**，如 Ehcache。

mapper.xml

mybatis.config.xml

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

测试：

```
user.find();
user.delete(22);
System.out.println("-----");
session.clearCache();
session.commit();
user.find();
```

执行三次sql

对于缓存数据更新机制，当某一个作用域(一级缓存Session/二级缓存Namespaces)的进行了 C/U/D 操作后，默认该作用域下所有 **select** 中的缓存将被**clear**。

