## 一、引言

Mybatis可以直接传入一个集合，通过标签将集合中的数据遍历出来实现批量新增

不同的**[数据库](http://lib.csdn.net/base/mysql" \o "MySQL知识库" \t "/home/caojx/文档\\x/_blank)**批量新增有区别，如下实现两种方式的批量新增

## 二、实现

### 2.1映射文件

**[html]** [view plain](http://blog.csdn.net/u014292162/article/details/52092495" \o "view plain) [copy](http://blog.csdn.net/u014292162/article/details/52092495" \o "copy)

1. <!--单条插入  -->
2. **<insert** id="insertOne" parameterType="com.imooc.bean.CommandContent"**>**
3. insert command\_content(content,command\_id) values(#{content},#{commandId))
4. **</insert>**
6. <!-- mysql 批量添加
7. 批量插入的时候，只要在values后边拼接多个需要插入的值便可
8. insert command\_content(content,command\_id) values(?,?),(?,?),(?,?)
9. --**>**
10. **<insert** id="insertBatchForMysql" parameterType="java.util.List"**>**
11. insert command\_content(content,command\_id) values
12. **<foreach** collection="list" item="item" separator=","**>**
13. (#{content},#{commandId})
14. **</foreach>**
15. **</insert>**
17. <!-- oracle批量增加实现
18. oracle批量插入的时候，可以使用子查询的方式插入
19. insert command\_content(content,commandId) values
20. (
21. select ?,? from dual union all
22. select ?,? from dual union all
23. select ?,? from dual union all
24. )
26. select ?,? from dual union all 执行结果就是"？"的值
27. --**>**
28. **<insert** id="insertBatchForOracle" parameterType="java.util.List"**>**
29. insert command\_content(content,commandId) values
30. **<foreach** collection="list" item="item" separator="union all"**>**
31. select
32. #{item.id,jdbcType=VARCHAR},
33. #{item.commandId,jdbcType=VARCHAR}
34. from dual
35. **</foreach>**
36. **</insert>**

### 2.2接口

**[java]** [view plain](http://blog.csdn.net/u014292162/article/details/52092495" \o "view plain) [copy](http://blog.csdn.net/u014292162/article/details/52092495" \o "copy)

1. **package** com.imooc.dao;
3. **import** java.util.List;
5. **import** com.imooc.bean.CommandContent;
7. **public** **interface** ICommandContentDao {
8. /\*
9. \* 单条插入
10. \* \*/
11. **public** **void** insertOne(CommandContent commandContent);
13. /\*
14. \* oracle批量插入
15. \* \*/
16. **public** **void** insertBatchForOracle(List<CommandContent> commandContentList);
18. /\*
19. \* mysql批量插入
20. \* \*/
21. **public** **void** insertBatchForMysql(List<CommandContent> commandContentList);
22. }

### 3.3批量添加实现

**[html]** [view plain](http://blog.csdn.net/u014292162/article/details/52092495" \o "view plain) [copy](http://blog.csdn.net/u014292162/article/details/52092495" \o "copy)

1. package com.imooc.dao;
3. import java.util.List;
5. import org.apache.ibatis.session.SqlSession;
7. import com.imooc.bean.CommandContent;
8. import com.imooc.bean.Message;
9. import com.imooc.db.DBAccess;
11. public class CommandContentDao {
13. DBAccess dbAccess = new DBAccess();
15. //批量新增
16. public void insertBatch(List**<CommandContent>** contentList) throws Exception{
17. SqlSession session = null;
18. try {
19. session = dbAccess.getSqlSession();
20. /\*for(CommandContent commandContent:contentList){ //效率比较低，在jdbc实现的时候是使用batch
21. ICommandContentDao iCommandContentDao = session.getMapper(ICommandContentDao.class);
22. iCommandContentDao.insertOne(commandContent);
23. }\*/
24. //高效率的，直接传入list
25. ICommandContentDao iCommandContentDao = session.getMapper(ICommandContentDao.class);
26. iCommandContentDao.insertBatchForOracle(contentList);
27. session.commit();
28. } catch (Exception e) {
29. throw new Exception(e.getMessage());
30. } finally {
31. if (session != null) {
32. session.close();
33. }
34. }
35. }
36. }