badqiu

永久域名 http://badqiu.iteye.com



badqiu

浏览: 134293 次

性别: 💣

来自: 珠海



🏠 详细资料 📄 留言簿

搜索本博客

最近访客 >>更多访



corleonelu



<u>liquidthinker</u>

yygybing

博客分类

- 全部博客 (55)
- 默认类别 (9)
- Rapid-framework (22)
- AJAX (3)
- Java (7)
- agile (1)
- flex (2)

>>更多留言 我的留言簿

■ 你好 昨天的问题搞定了,是我 没注意设置数据库

的schema及catalog, 所...

- -- by eric.zhang
- 你好我的邮

箱zhanghouyin@gmail.com 谢

◆ 大规模网站架构ppt | 与Spring BlazeDS Integration相比, 更简单 ... ▶ △15顶



iBatis3基于方言(Dialect)的分页

关键字: ibatis3 dialect

2009-10-19

(注:以下代码是基于ibatis3 beta4的扩展,ibatis3正式版如果实现改变,将会继续跟进修改)

iBatis3默认使用的分页是基于游标的分页,而这种分页在不同的数据库上性能差异不一致,最好的办法当然是使用类似hibernate的基于 方言(Dialect)的物理分页功能。

iBatis3现在提供插件功能,通过插件我们可以编写自己的拦截器来拦截iBatis3的主要执行方法来完成相关功能的扩展。

能够拦截的的类如下:

Java代码

- Executor
- (update,query,flushStatements,commit,rollback,getTransaction,close,isClosed)
- 3. ParameterHandler

23

- (getParameterObject, setParameters)
- 5. ResultSetHandler
- 6. (handleResultSets, handleOutputParameters)
- 7. StatementHandler
- (prepare,parameterize,batch,update,query)

但此次我们感兴趣的是Executor.query()方法,查询方法最终都是执行该方法。 该方法完整是:

Java代码

Executor.query(MappedStatement ms, Object parameter, RowBounds rowBounds, ResultHandler resultHa ndler) throws SQLException;

分页方言的基本实现:

以Mysql数据库示例,即有一个方法 query(sql,offset,limit);

Java代码

query("select * from user",5,10);

经过我们的拦截器拦截处理,变成

Java代码

query("select * from user limit 5,10",0,0);

1. 拦截类实现:

Java代码

@Intercepts({@Signature(

```
-- by eric.zhang
■ 求助老大: 您好, 我最近在学
 习rapid framework框架 (3.0版
 本),遇...
 -- by eric.zhang
其他分类
■ 我的收藏 (28)
 我的代码(0)
■ 我的新闻 (13)
■ <u>我的论坛主题帖</u> (19)
 我的所有论坛帖 (214)
■ 我的精华良好帖(1)

    我解决的问题 (2)

最近加入群组
rapid-framework
■ 珠海之夜
存档
2011-05 (1)
 <u>2011-01</u> (1)
2010-10 (1)
■ 更多存档...
评论排行榜
 rapid-framework v3.9新版本发
■ 接口滥用问题
■ <u>Holder模式</u>
■ log4idbc日志框架介绍
■ 分布式应用上下文(Distributed
 ThreadLocal ...
  3 RSS
  Add to Google
```

```
2.
                       type= Executor.class,
 3.
                        method = "query",
 4.
                        args = {MappedStatement.class, Object.class, RowBounds.class, ResultHandler.cla
      ss})})
 5.
      public class OffsetLimitInterceptor implements Interceptor{
               static int MAPPED STATEMENT INDEX = 0;
 6.
               static int PARAMETER_INDEX = 1;
 7.
 8.
               static int ROWBOUNDS_INDEX = 2;
               static int RESULT_HANDLER_INDEX = 3;
 9.
10.
               Dialect dialect;
11.
12.
13.
               public Object intercept(Invocation invocation) throws Throwable {
                        processIntercept(invocation.getArgs());
14.
                        return invocation.proceed();
15.
16.
               }
17.
18.
               void processIntercept(final Object[] queryArgs) {
19.
                        //queryArgs = query(MappedStatement ms, Object parameter, RowBounds rowBounds,
      ResultHandler resultHandler)
20.
                       MappedStatement ms = (MappedStatement)queryArgs[MAPPED_STATEMENT_INDEX];
21.
                       Object parameter = queryArgs[PARAMETER_INDEX];
22.
                        final RowBounds rowBounds = (RowBounds)queryArgs[ROWBOUNDS_INDEX];
23.
                        int offset = rowBounds.getOffset();
24.
                        int limit = rowBounds.getLimit();
25.
                        if(dialect.supportsLimit() && (offset != RowBounds.NO_ROW_OFFSET || limit != Ro
26.
      wBounds.NO_ROW_LIMIT)) {
27.
                                BoundSql boundSql = ms.getBoundSql(parameter);
28.
                                String sql = boundSql.getSql().trim();
29.
                                if (dialect.supportsLimitOffset()) {
30.
                                         sql = dialect.getLimitString(sql, offset, limit);
                                         offset = RowBounds.NO_ROW_OFFSET;
31.
32.
                                } else {
                                         sql = dialect.getLimitString(sql, 0, limit);
33.
34.
                                limit = RowBounds.NO_ROW_LIMIT;
35.
36.
                                queryArgs[ROWBOUNDS_INDEX] = new RowBounds(offset,limit);
37.
38.
                                BoundSql newBoundSql = new BoundSql(sql, boundSql.getParameterMappings(
      ), boundSql.getParameterObject());
                                MappedStatement newMs = copyFromMappedStatement(ms, new BoundSqlSqlSour
39.
      ce(newBoundSql));
40.
                                queryArgs[MAPPED STATEMENT INDEX] = newMs;
41.
                       }
42.
               }
43.
44.
               private MappedStatement copyFromMappedStatement(MappedStatement ms,SqlSource newSqlSourc
      e) {
45.
                       Builder builder = new MappedStatement.Builder(ms.getConfiguration(),ms.getId(),
      newSqlSource,ms.getSqlCommandType());
46.
                       builder.resource(ms.getResource());
                       builder.fetchSize(ms.getFetchSize());
47.
48.
                       builder.statementType(ms.getStatementType());
49.
                       builder.keyGenerator(ms.getKeyGenerator());
50.
                       builder.keyProperty(ms.getKeyProperty());
51.
                        builder.timeout(ms.getTimeout());
52.
                        builder.parameterMap(ms.getParameterMap());
53.
                        builder.resultMaps(ms.getResultMaps());
54.
                        builder.cache(ms.getCache());
                       MappedStatement newMs = builder.build();
55.
56.
                        return newMs;
57.
               }
58.
59.
               public Object plugin(Object target) {
```

```
return Plugin.wrap(target, this);
61.
               }
62.
63.
               public void setProperties(Properties properties) {
64.
                        String dialectClass = new PropertiesHelper(properties).getRequiredString("diale
      ctClass");
65.
                        try {
66.
                                dialect = (Dialect)Class.forName(dialectClass).newInstance();
67.
                        } catch (Exception e) {
68.
                                throw new RuntimeException("cannot create dialect instance by dialectC
      lass:"+dialectClass,e);
69.
                        }
70.
                        System.out.println(OffsetLimitInterceptor.class.getSimpleName()+".dialect="+dial
      ectClass);
71.
               }
72.
73.
               public static class BoundSqlSqlSource implements SqlSource {
74.
                        BoundSql boundSql;
                        public BoundSqlSqlSource(BoundSql boundSql) {
75.
76.
                                this.boundSql = boundSql;
77.
78.
                        public BoundSql getBoundSql(Object parameterObject) {
79.
                                return boundSql;
80.
81.
               }
82.
83.
```

2.ibatis3配置文件内容:

Xml代码 ☆

```
<configuration>
 1.
 2.
               <plugins>
                       <!-- 指定数据库分页方言Dialect, 其它方言:OracleDialect,SQLServerDialect,SybaseDial
 3.
      ect,DB2Dialect,PostgreSQLDialect,MySQLDialect,DerbyDialect-->
                       <plugin interceptor="cn.org.rapid_framework.ibatis3.plugin.OffsetLimitIntercepto"</pre>
 4.
      r">
                                roperty name="dialectClass" value="cn.org.rapid framework.jdbc.dialec
 5.
      t.MySQLDialect"/>
 6.
                       </plugin>
 7.
               </plugins>
 8.
 9.
               <environments default="development">
                       <environment id="development">
10.
11.
                                <transactionManager type="JDBC" />
12.
                                <dataSource type="POOLED">
13.
                                        cproperty name="driver" value="com.mysql.jdbc.Driver" />
                                        roperty name="url" value="jdbc:mysql://localhost:3306/test"
14.
                                        roperty name="username" value="root" />
15.
                                        roperty name="password" value="123456" />
16.
17.
                                </dataSource>
18.
                       </environment>
19.
               </environments>
20.
21.
               <mappers>
22.
                       <mapper resource="com/company/project/model/mapper/BlogMapper.xml" />
23.
               </mappers>
24.
      </configuration>
```

3.MysqlDialect实现

```
23
Java代码
       public class MySQLDialect extends Dialect{
 1.
 2.
 3.
               public boolean supportsLimitOffset(){
 4.
                        return true;
               }
 5.
 6.
 7.
               public boolean supportsLimit() {
 8.
                        return true;
 9.
               }
10.
               public String getLimitString(String sql, int offset, int limit) {
11.
                        return getLimitString(sql,offset,String.valueOf(offset),limit,String.valueOf(lim
12.
       it));
13.
               }
14.
15.
               public String getLimitString(String sql, int offset,String offsetPlaceholder, int limit,
        String limitPlaceholder) {
16.
                        if (offset > 0) {
                                 return sql + " limit "+offsetPlaceholder+","+limitPlaceholder;
17.
18.
                        } else {
19
                            return sql + " limit "+limitPlaceholder;
20.
21.
               }
22.
23.
```

其它数据库的Dialect实现请查看:

http://rapid_framework.googlecode.com/svn/trunk/rapid_framework/src/rapid_framework common/cn/org/rapid_framework/jdbc/dialect/

4.分页使用

直接调用SqlSession.selectList(statement, parameter, new RowBounds(offset,limit))即可使用物理分页

5.存在的问题

现不是使用占位符的方式(即不是"limit?,?"而是使用"limit 8,20")使用分页

完整代码可以查看即将发布的rapid-framework 3.0的ibatis3插件

浜 度鍛厶勾6.18搴楀簡 绉椒潃

姣忓お1娆惧崐浠风 蠘€,鏈夋湪鏈◆ 80000浠跺晢鍝◆鐏 垎绾◆鏈◆鐤 媯鎶 笉鍋◆/div> www.360buy.com



◆ 大规模网站架构ppt 与Spring BlazeDS Integration相比, 更简单.

09:29 浏览 (5527) <u>评论</u> (6) 分类: <u>Rapid-framework</u> <u>相关推荐</u> <u>MORE</u>

评论

6 楼 <u>liuwei blog</u> 2011-03-02 <u>引用</u>



5 楼 badqiu 2010-04-30 引用

不一定吧, 你可以写两个ibatis configation.xml吧,因为如果是不同数据库的,肯定也存在两个数据源的.

4 楼 <u>cameron8</u> 2010-04-26 <u>引用</u> 并且重新写个xml文件也没有用,这种做法只能用单一的数据库. 3 楼 cameron8 2010-04-26 引用 Java代码 23 1. <plugin interceptor="cn.org.rapid_framework.ibatis3.plugin.OffsetLimitInterceptor"> cproperty name="dialectClass" value="cn.org.rapid_framework.jdbc.dialect.MySQL 2. Dialect"/> 不能同时支持两种数据库Dialect, 因为dialectClass属性加载的时候会复盖 2 楼 <u>pochonlee</u> 2009-10-20 <u>引用</u> 3.0顶一个!期待更多。 1楼超级潜水员 2009-10-19 引用 支持lz的3.0喔!!! 发表评论 表情图标 字体颜色: □□ 字体大小: □□ 对齐: □□ 提示:选择您需要装饰的文字,按上列按钮即可添加上相应的标签 您还没有登录,请登录后发表评论(快捷键 Alt+S / Ctrl+Enter)

声明: ITeye文章版权属于作者,受法律保护。没有作者书面许可不得转载。若作者同意转载,必须以超链接形式标明文章原始出处和作者。 © 2003-2011 ITeye.com. All rights reserved. [京ICP证110151号]