Atitit 多结果集的处理jdbc mybatis jpa

# 多结果集jdbc

   在执行存储过程，或者在使用允许在单个查询中提交多个SELECT语句的数据库时，一个查询有可能会返回多个结果集。下面是获取所有结果集的步骤：

1. 使用execute方法来执行SQL语句。

2. 获取第一个结果集或更新计数。

3. 重复调用getMoreResults方法以移动到下一个结果集（这个调用会自动关闭前一个结

果集）。

4. 当不存在更多的结果集或更新计数时，完成操作。

 每个Connection对象都可以创建一个或一个以上的Statement对象。同一个Statement对象可以用于多个不相关的命令和查询。但是，一个Statement对象最多只能打开一个结果集。如果需要执行多个查询操作，且需要同时分析查询结果，那么必须创建多个Statement对象。

这看上去似乎很有局限性。但实际上，我们通常并不需要同时处理多个结果集。如果结果集相互关联，我们就可以使用组合查询，这样就只需要分析一个结果。对数据库进行组合查询比使用Java程序遍历多个结果集要高效得多。

当使用完ResultSet、 Statement或Connection对象时，应立即调用close方法。这些对象都使用了规模较大的数据结构，所以我们不应该等待垃圾回收器来处理它们。

   如果Statement对象上有一个打开的结果集，那么调用close方法将自动关闭该结果集。同样地，调用Connection类的close方法将关闭该连接上的所有语句。

————————————————

多结果集

   在执行存储过程，或者在使用允许在单个查询中提交多个SELECT语句的数据库时，一个查询有可能会返回多个结果集。下面是获取所有结果集的步骤：

1. 使用execute方法来执行SQL语句。

2. 获取第一个结果集或更新计数。

3. 重复调用getMoreResults方法以移动到下一个结果集（这个调用会自动关闭前一个结

果集）。

4. 当不存在更多的结果集或更新计数时，完成操作。

## 可滚动和更新的结果集

默认情况下，结果集是不可滚动和不可更新的。为了从查询中获取可滚动的结果集，必须使用以下方法得到一个不同的Statement对象

Statement sta = connect.createStatement(type,concurrenry)

PreparedStatement s = connect.preparedStament(xxx,type,concurrenry)

type

TYPE\_FORWARD\_ONLY 结果集不能滚动

TYPE\_SCROLL\_INSENSITIVE 结果集可以滚动，但对数据库变化不敏感

TYPE\_SCROLL\_SENSITIVE 结果集可以滚动，且对数据库变化敏感

concurrenry

CONCUR\_READ\_ONLY 结果集不能用于更新数据库（默认值）

CONCUR\_UPDATABLE 结果集可以用于更新数据库

              生成了结果集之后就可以使用一些方法了set.next(),set.previous(),set.relative(n),set.absolute(n),first、 last、 beforeFirst和afterLast这些简便方法用于将光标移动到第一行、最后一行、第一行之前或最后一行之后。最后， isFirst、 isLast、 isBeforeFirst和isAfterLast用于测试光标是否位于这些特殊位置上.

   并非所有的查询都会返回可更新的结果集。如果查询涉及多个表格的连接操作，那么它所产生的结果集将是不可更新的。如果查询只涉及一个表格，或者在查询时是使用主键连接多个表格的，那么它所产生的结果集将是可更新的结果集。可以调用ResultSet类中的getConcurrency方法来确定结果集是否是可更新的。

   所有对应于SQL类型的数据类型都配有updateXxx方法，比如updateDouble,updateString等。与getXxx方法相同，在使用updateXxx方法时必须指定列的名称或序号。然后，你可以给该字段设置新的值。updateXxx方法改变的只是结果集中的行值，而非数据库中的值。当更新完行中的字段值后，必须调用updateRow方法，这个方法将当前行中的所有更新信息发送给数据库。调用updateRow方法就将光标移动到其他行上，那么所有的更新信息都将被行集丢弃，而且永远也不会被传递给数据库。还可以调用cancelRowUpdates方法来取消对当前行的更新。

   如果想在数据库中添加一条新的记录，首先需要使用moveToInsertRow方法将光标移动到特定的位置，我们称之为插入行（ insert row）。然后，调用updateXxx方法在插入行的位置上创建一个新的行。在上述操作全部完成之后，还需要调用insertRow方法将新建的行发送给数据库。完成插入操作后，再调用moveToCurrentRow方法将光标移回到调用moveToInsertRow方法之前的位置。

ResultSet类中的updateRow、 insertRow和deleteRow方法的执行效果等同于SQL命令中的UPDATE、 INSERT和DELETE。不过，习惯于Java编程语言的程序员通常会觉得使用结果集来操控数据库要比使用SQL语句自然得多。

## 行集 。 RowSet接口

   可滚动的结果集虽然功能强大，却有一个重要的缺陷：在与用户的整个交互过程中，必须始终与数据库保持连接。用户也许会离开电脑旁很长一段时间，而在此期间却始终占有着数据库连接。这种方式存在很大的问题，因为数据库连接属于稀有资源。在这种情况下，我们可以使用行集。 RowSet接口继承了ResultSet接口，却无需始终保持与数据库的连接。

如下所示为javax.sql.rowset包提供的接口，它们都扩展了RowSet接口：

CachedRowSet允许在断开连接的状态下执行相关操作。

WebRowSet对象代表了一个被缓存的行集，该行集可以保存为XML文件。该文件可以移动到Web应用的其他层中，只要在该层中使用WebRowSet重新打开该文件即可。

FilteredRowSet和JoinRowSet接口支持对行集的轻量级操作，它们等同于SQL中的SELECT和JOIN操作。上述两个接口的操作对象是存储在行集中的数据，因此运行时无需建立数据库连接。

JdbcRowSet是ResultSet接口的一个瘦包装器。它从RowSet中继承了get方法和set方法，从而将一个结果集转换成一个bean。

一个被缓存的行集包含了一个结果集中所有的数据。 CachedRowSet是ResultSet接口的子接口，所以你完全可以像使用结果集一样来使用被缓存的行集。被缓存的行集有一个非常重要的优点：断开数据库连接后仍然可以使用行集。

   甚至可以修改被缓存的行集中的数据。当然，这些修改不会立即反馈到数据库中。相反，必须发起一个显式的请求，以便让数据库真正接受所有修改。此时CachedRowSet类会重新连接到数据库，并通过执行SQL命令向数据库中写入所有修改后的数据。

在填充了行集之后，数据库中的数据发生了改变，这显然容易造成数据不一致性。为了解决这个问题，参考实现会首先检查行集中的原始值（即修改前的值）是否与数据库中的当前值一致。如果一致，那么修改后的值将覆盖数据库中的当前值。否则，将抛出SyncProviderException异常，且不向数据库写回任何值。

RowSet

String getURL()

void setURL(String url)获取或设置数据库的URL。

String getUsername()

void setUsername(String username)获取或设置连接数据库所需的用户名。

String getPassword()

void setPassword(String password)获取或设置连接数据库所需的密码。

String getCommand()

void setCommand(String command)获取或设置向行集中填充数据时需要执行的命令。

void execute()

通过执行使用setCommand方法设置的命令集来填充行集。为了使驱动管理器可以获得连接，必须事先设定URL、用户名和密码。

CachedRowSet

void execute(Connection conn)

通过执行使用setCommand方法设置的命令集来填充行集。该方法使用给定的连接，并负责关闭它。

void populate(ResultSet result)将指定的结果集中的数据填充到被缓存的行集中。

String getTableName()

void setTableName(String tableName)

获取或设置数据库表名称，填充被缓存的行集时所需的数据来自于该表。

int getPageSize()

void setPageSize(int size)获取和设置页的尺寸。

boolean nextPage()

boolean previousPage()

加载下一页或上一页，如果要加载的页存在，则返回ture。

void acceptChanges()

void acceptChanges(Connection conn)

重新连接数据库，并写回行集中修改过的数据。如果因为数据库中的数据已经被修改而导致无法写回行集

————————————————

# Mybatis 多结果集

# Jpa多结果集