(1)加载数据库的驱动

(2)主要的接口使用介绍

(2-1)DriverManager类：(1)getConnection() (2)setLoginTimeout()

(2-2)Connection接口：

(1)getMetaData()返回数据库的元数据

(2)CreatStatement()

(3)prepareStatement()----这个会进行预编译 可以防止sql注入 用？作为占位符

(2-3)statement接口：用于查询

(1)execute()所有的sql语句都可以执行返回成功或者失败

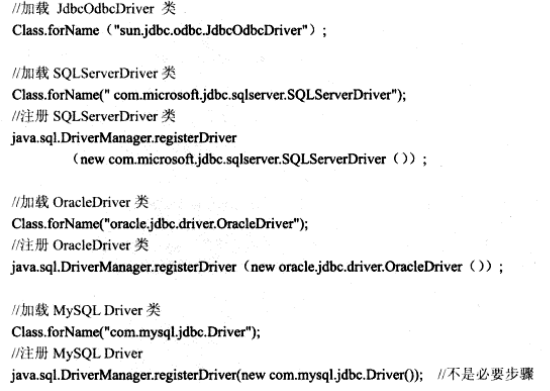
(2)executeUpdate()执行除了select的sql语句

(3)executeQuery这行select的sql语句，并返回ResultSet对象，存放的是结果集

(2-4)ResultSet类提供了getString int char等方法从其中取值

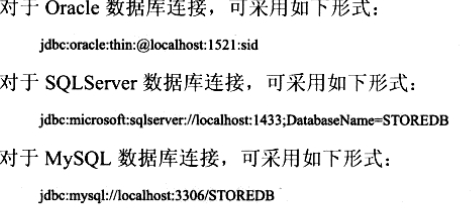
While(rst.nest()){rst.getxxx();}

主要方法：

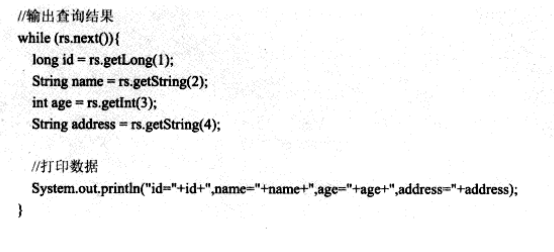


Resgiter已经不是必要步骤了

常见的数据库URL

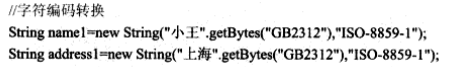


遍历ResultSet

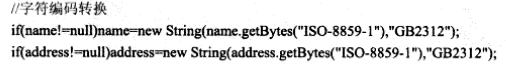


字符转码问题

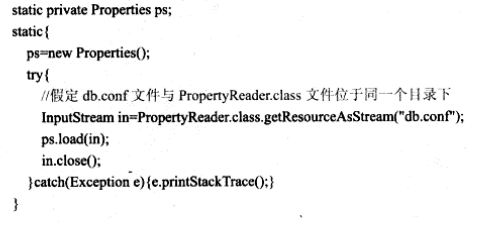
1. 向数据库村数据



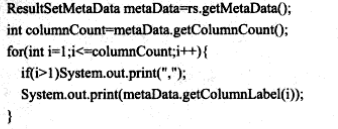
1. 向数据库取数据



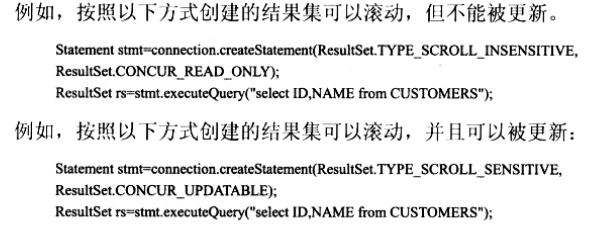
Properties类的用法



ResultSetMetadata类可以遍历任意个ResultSet

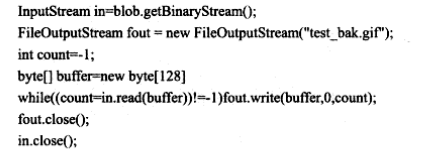


创建可以滚动的结果集

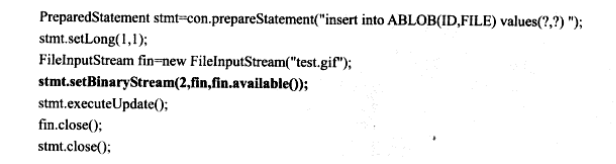


调用存储过程(存储比较大的数据)

从数据库中读取，写到一个文件中



从文件中读取，写到数据库中



数据库事务

(1)connection.setAutoCommit(false)

执行的sql

(2)connection.commit();

(3)捕获到异常：connection.Rollback()-------🡪>>取消在这次事务中的所有的对数据库的操作，如果只想回滚一部分，connection.setSavePoint()方法

Connection.rollBack(上面创建的sqvePoint对象)

批量操作

Statement.addBeatch()

一般addBeatch()都会有几条的

Statement.executeBeatch();

事务隔离级别：用于处理对各事务同时处理同一个数据的时候

