# day13 JDBC

## 批处理

### 1.简介:

在SQL中存在批量处理SQL语句的使用场景.

可以批量的添加/删除/修改/查询...

### 2.批处理的实现方式

#### 2.1 stmt.addBatch()

优点:

可以使用预编译的Statement,效率高;

缺点:

只能用于SQL语句的结构相同的情况下,如果SQL语句的结构不同,这种方式就不合适.

#### 2.2 stmt.addBatch(sql)

优点:

可以同时执行多条不同的SQL 语句.

缺点:

不能使用预编译的Statement对象.

### 3.批处理相关的API方法

stmt.addBatch()/addBatch(sql);

stmt.executeBatch();

stmt.clearBatch();

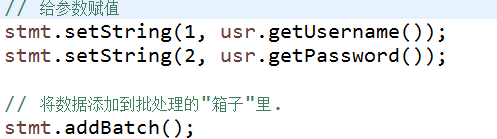
### 4.具体实现

第一种方式,addBatch():

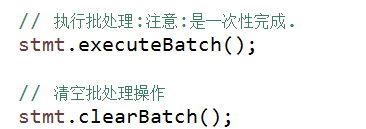
#### 4.1 定义SQL语句



#### 4.2 将待执行的语句添加到批处理中.



#### 4.3 执行批处理和清空批处理

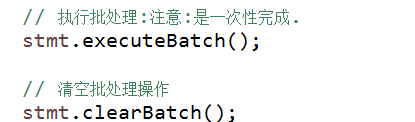


第二种方式,addBatch(sql):

该方法使用较少,主要处理同时执行不同的sql语句的情况.

而且该方法不能使用预编译.



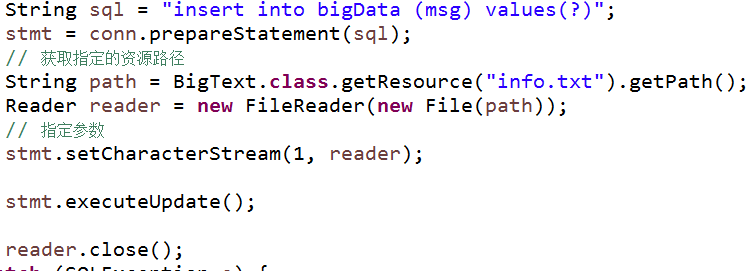


## 添加和读取大文本数据

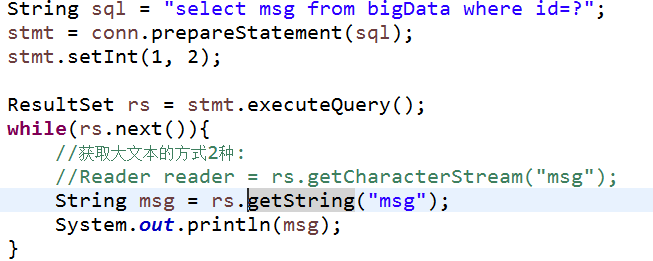
### 添加大文本数据

表中大文本字段的类型应该是text,clob这样的类型.

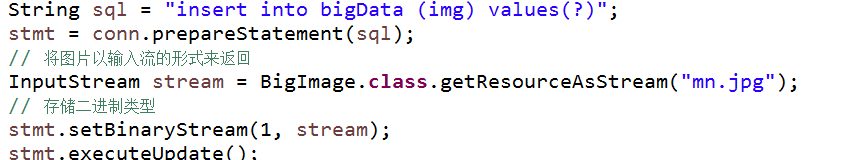
具体代码:



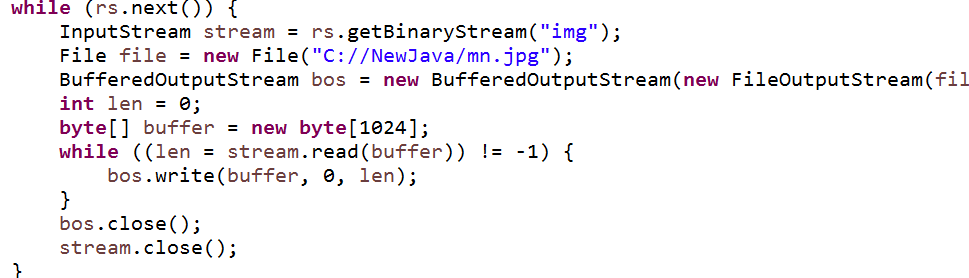
### 读取大文本数据



### 存储图片等多媒体资源



### 读取图片等多媒体资源



## 多表关联操作

案例:实现级联添加

员工表---Emploee,

部门表---->Dept.

Dept:部门

deptId,dname

Emploee:员工

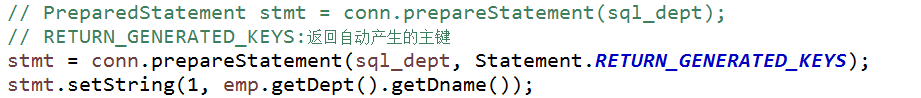
id,ename,deptId;

实现过程:

#### 1.添加到部门表的SQL语句.



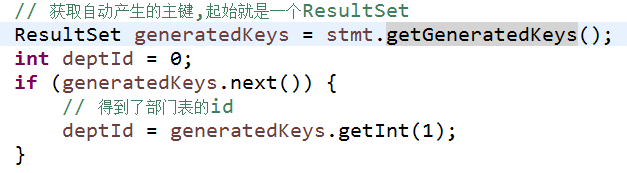
#### 2.预编译该SQL语句



#### 3.执行该预编译的SQL语句

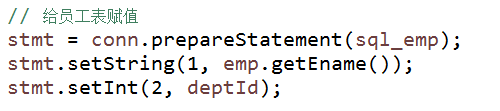


#### 4.获取自动自增长的列值----🡪deptId,主键.



#### 5.将第4步中获取的deptId值,插入到员工表中.





#### 6.执行员工表的插入操作.



## 数据库的事务

### 1.概念:

事务(Transaction),是一种并发控制的单位,是用户自己定义的一种对数据库操纵的序列.

### 2.使用场景

经常用于涉及到钱的操作,比如银行转账,支付等操作时.

### 3.事务的特点(ACID)

**3.1原子性(Atomicity):**

**事务是数据库操作时的一个逻辑单位,事务中的所有的操作要么都做,要么都不做;**

**3.2一致性(Consitency):**

**事务的执行结果,要从一个一致性状态变为另一个一致性状态,一致性与原子性是密不可分的.**

**3.3 隔离性(Isolation):**

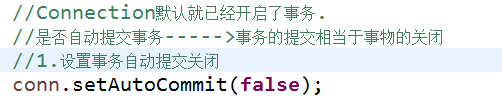
**一个事务的执行不能被别的事务所干扰,只能同时执行一个事务!**

**3.4 持久性(Durability):**

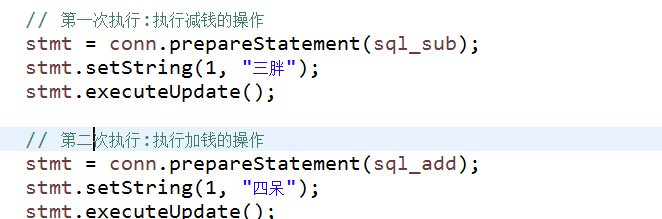
**事务一旦被提交,对数据库中数据的修改是永久性的.**

### 4.具体实现

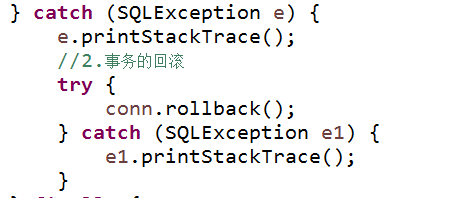
#### 4.1 关闭自动提交功能



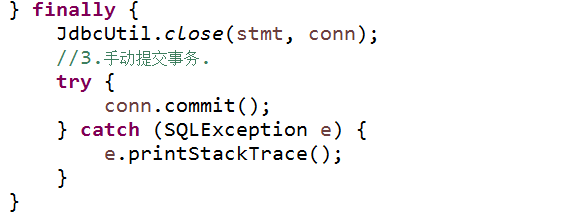
#### 4.2 执行具体的数据库业务逻辑



#### 4.3 设置数据库发生异常时的回滚功能

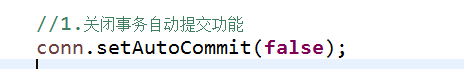


#### 4.4 手动提交事务



### 回滚到保存点

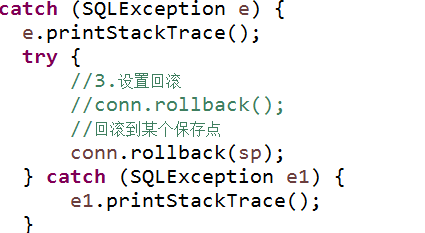
#### 5.1 设置事务自动提交功能关闭



#### 5.2 执行具体的业务逻辑,设置保存点



#### 5.3 回滚到某个保存点



#### 5.4 提交事务

