

《数据库应用程序开发》实验报告

学号:

姓名:

Unit 1 JDBC Programming I

任务 1 将 P171 页程序片段补充完整

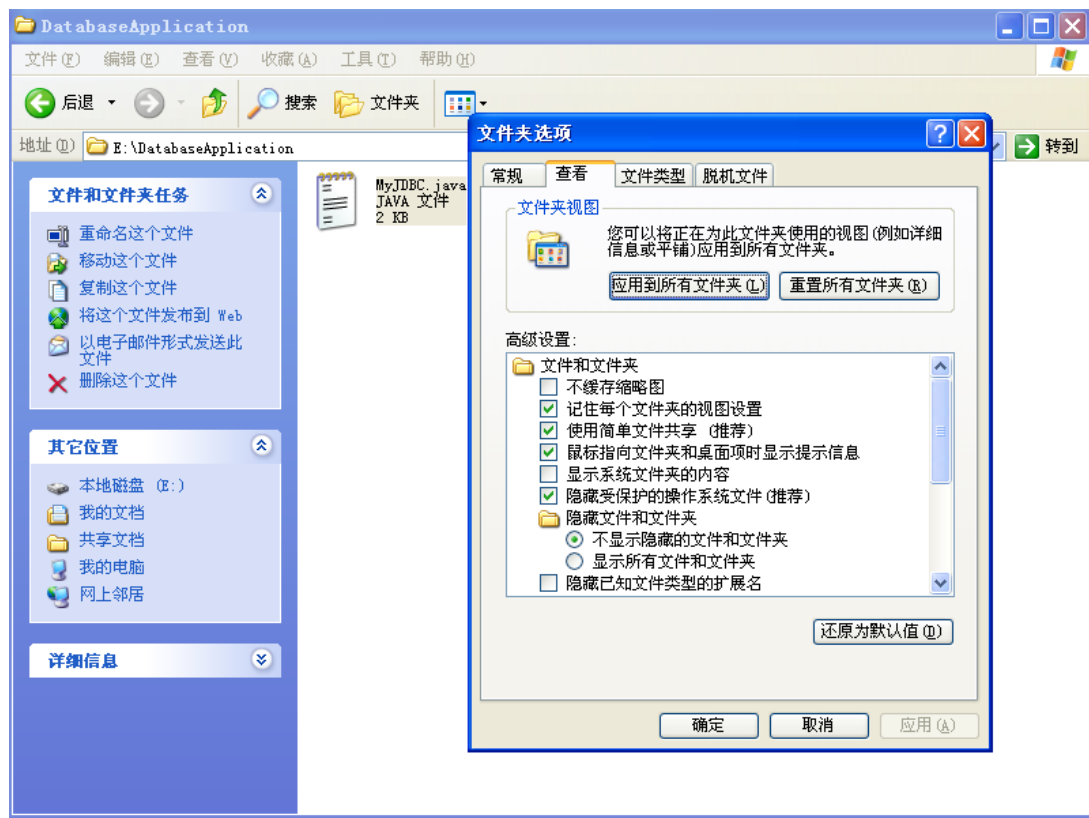
● 调试过程

1) 文件类型

得益于长期与命令行窗口打交道,我没有在文件类型方面犯错,但我认为这是值得一提的。

要运行 java 程序显然要先将保存有 java 程序代码的 txt 文件改为 java 文件。初始情况下,win7 的后缀不可见,我们应先进入“我的电脑”界面,点击菜单栏中的“工具”,然后选择“文件夹选项”,进入“查看”页面,取消勾选“隐藏已知文件类型的扩展名”,最后点击应用即可完成设置。

完成以上设置后就可通过重命名修改后缀,将.txt 修改为.java。

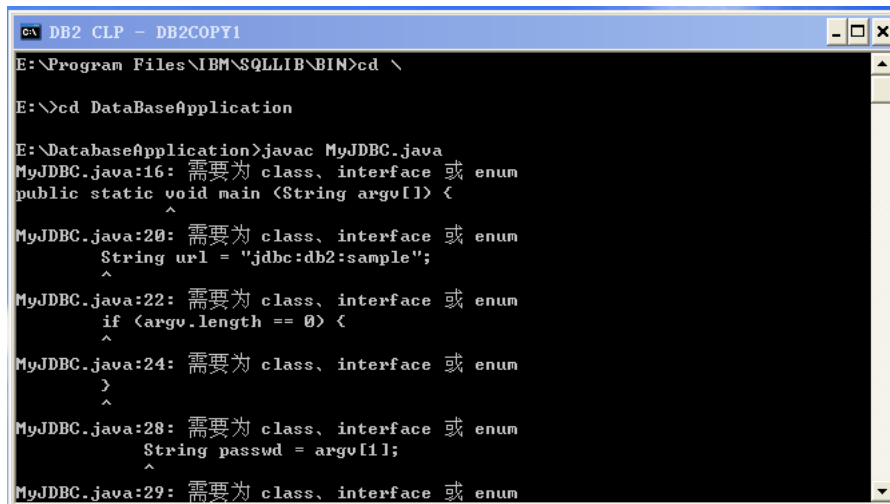


2) 编译错误

如下图所示，在编译中出现错误：需要为 class、interface 或 enum。

原因是 main()函数在 class MyJDBC 之外，也就是老师视频中所说的“有个缺大括号的地方”。因为老师说的是“缺大括号”，我在打代码时就着重注意了括号是否完整，但没想到不是缺大括号，而是 main()函数没在 class MyJDBC 的大括号内。

修改大括号位置后即可正常编译。（犯这种低级错误实在不该，打代码时确实太不认真了）

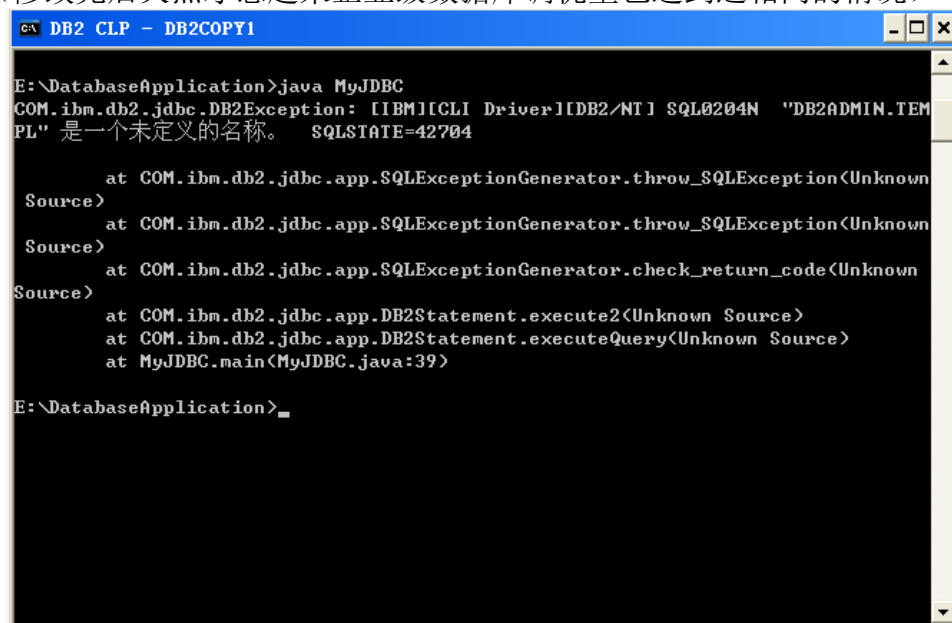


```
DB2 CLP - DB2COPY1
E:\Program Files\IBM\SQLLIB\BIN>cd \
E:\>cd DataBaseApplication
E:\DatabaseApplication>javac MyJDBC.java
MyJDBC.java:16: 需要为 class、interface 或 enum
public static void main (String argv[]) {
^
MyJDBC.java:20: 需要为 class、interface 或 enum
    String url = "jdbc:db2:sample";
^
MyJDBC.java:22: 需要为 class、interface 或 enum
    if (argv.length == 0) {
^
MyJDBC.java:24: 需要为 class、interface 或 enum
    }
^
MyJDBC.java:28: 需要为 class、interface 或 enum
    String passwd = argv[1];
^
MyJDBC.java:29: 需要为 class、interface 或 enum
```

3) 权限（模式）问题

如下图所示，在运行中出现错误：“DB2ADMIN.TEMPL”是一个未定义的名称。

一开始我以为是授权有问题，但经过反复确认，我确实已经将权限授予了 DB2ADMIN 用户。最终我猜测是由于 TEMPL 的模式是 JLU 导致的，于是我将 "FROM TEMPL" 改为了 "FROM JLU.TEMPL"，重新编译后程序便可正确运行。（修改完后突然才想起来企业级数据库调优里也遇到过相同的情况）

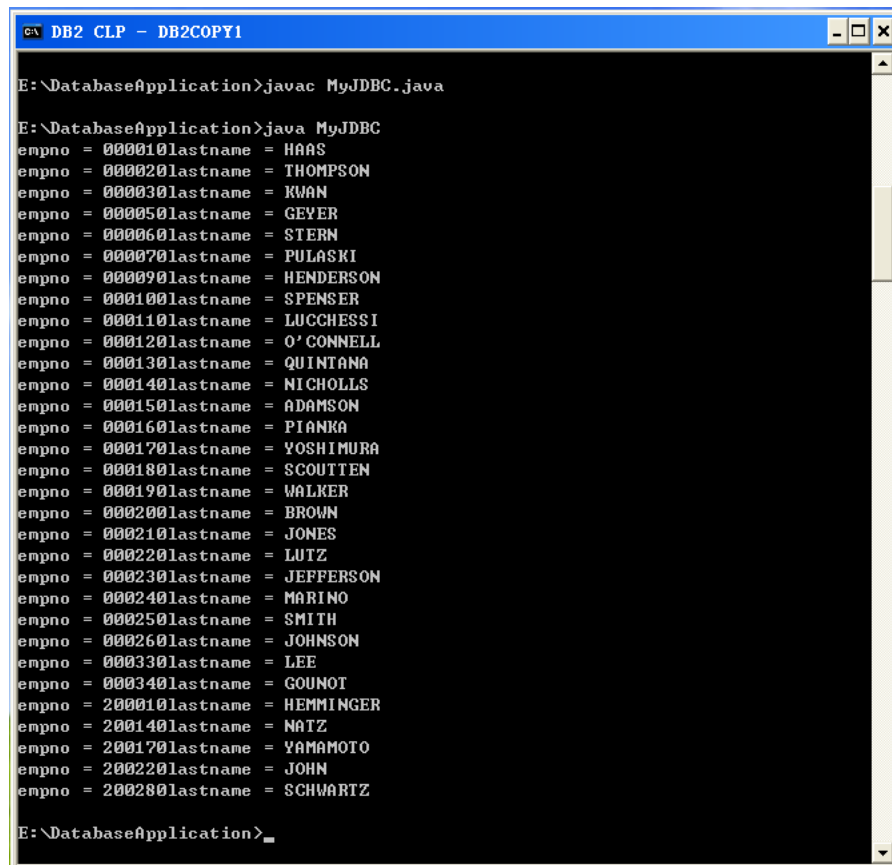


```
DB2 CLP - DB2COPY1
E:\DatabaseApplication>java MyJDBC
COM.ibm.db2.jdbc.DB2Exception: [IBM][CLI Driver][DB2/NT] SQL0204N "DB2ADMIN.TEMPL" 是一个未定义的名称。  SQLSTATE=42704

    at COM.ibm.db2.jdbc.app.SQLExceptionGenerator.throwSQLException(Unknown Source)
    at COM.ibm.db2.jdbc.app.SQLExceptionGenerator.throwSQLException(Unknown Source)
    at COM.ibm.db2.jdbc.app.SQLExceptionGenerator.check_return_code(Unknown Source)
    at COM.ibm.db2.jdbc.app.DB2Statement.execute2(Unknown Source)
    at COM.ibm.db2.jdbc.app.DB2Statement.executeQuery(Unknown Source)
    at MyJDBC.main(MyJDBC.java:39)

E:\DatabaseApplication>
```

- 运行结果



```
E:\DatabaseApplication>javac MyJDBC.java

E:\DatabaseApplication>java MyJDBC
empno = 000010lastname = HAAS
empno = 000020lastname = THOMPSON
empno = 000030lastname = KWAN
empno = 000050lastname = GEYER
empno = 000060lastname = STERN
empno = 000070lastname = PULASKI
empno = 000090lastname = HENDERSON
empno = 000100lastname = SPENSER
empno = 000110lastname = LUCCHESSI
empno = 000120lastname = O'CONNELL
empno = 000130lastname = QUINTANA
empno = 000140lastname = NICHOLLS
empno = 000150lastname = ADAMSON
empno = 000160lastname = PIANKA
empno = 000170lastname = YOSHIMURA
empno = 000180lastname = SCOUTTEN
empno = 000190lastname = WALKER
empno = 000200lastname = BROWN
empno = 000210lastname = JONES
empno = 000220lastname = LUTZ
empno = 000230lastname = JEFFERSON
empno = 000240lastname = MARINO
empno = 000250lastname = SMITH
empno = 000260lastname = JOHNSON
empno = 000330lastname = LEE
empno = 000340lastname = GOUNOT
empno = 200010lastname = HEMMINGER
empno = 200140lastname = NATZ
empno = 200170lastname = YAMAMOTO
empno = 200220lastname = JOHN
empno = 200280lastname = SCHWARTZ

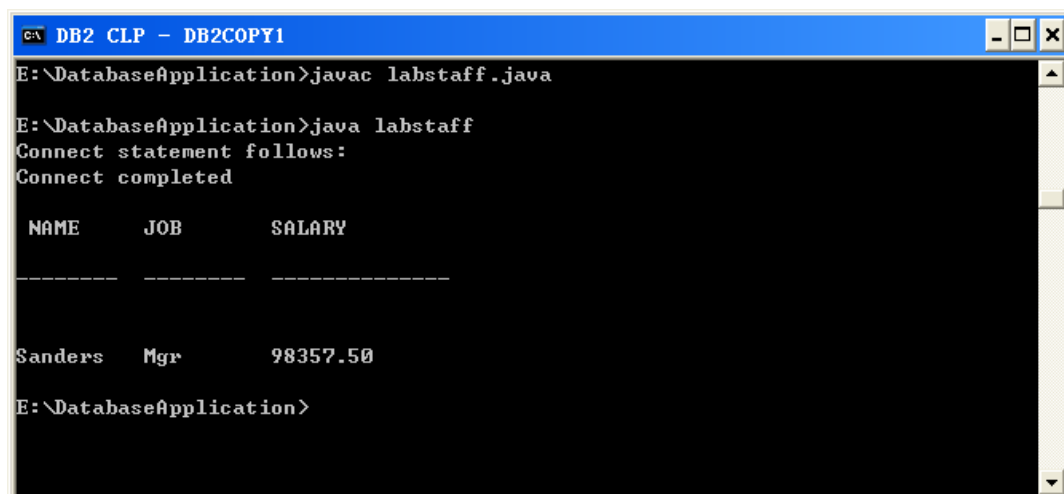
E:\DatabaseApplication>
```

任务 2 Modify the program labstaff.java

- 调试过程

此次任务没有错误

- 运行结果



```
E:\DatabaseApplication>javac labstaff.java

E:\DatabaseApplication>java labstaff
Connect statement follows:
Connect completed

  NAME      JOB      SALARY
-----
Sanders    Mgr      98357.50

E:\DatabaseApplication>
```

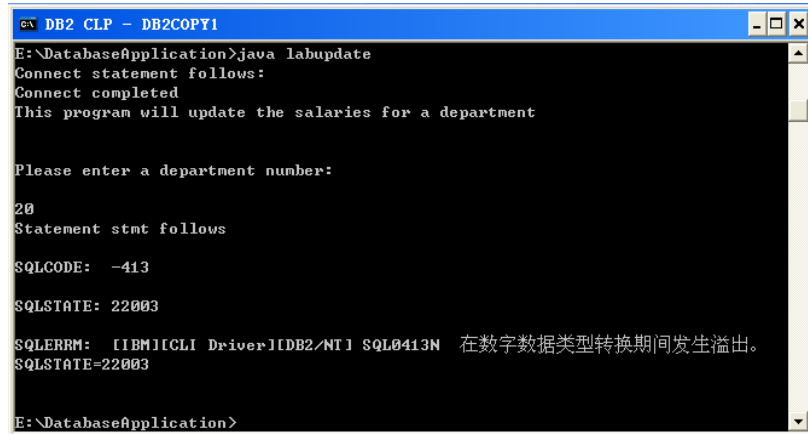
任务 3 Modify the program labupdate.java

● 调试过程

1) 溢出错误

在修改某些部门时出现了错误：**在数字数据类型转换期间发生溢出。**

在给编号为 20 的部门增薪时，发生了溢出；而给其他部门增薪时，没有溢出。造成这一结果最大的可能就是编号为 20 的部门中有成员的原本薪水就高，在增薪后超过了存储的上界。



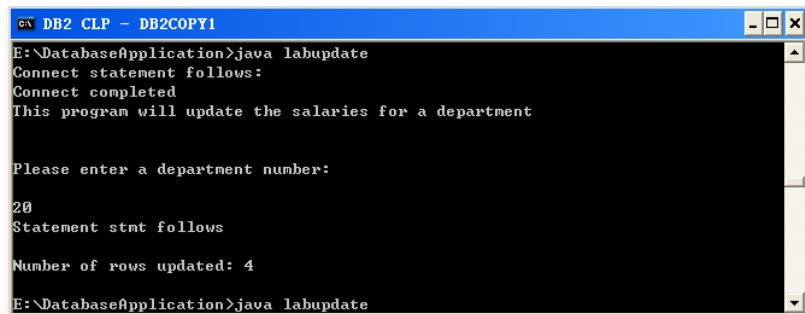
```
DB2 CLP - DB2COPY1
E:\DatabaseApplication>java labupdate
Connect statement follows:
Connect completed
This program will update the salaries for a department

Please enter a department number:
20
Statement stmt follows
SQLCODE: -413
SQLSTATE: 22003
SQLERRM: [IBM] [CLI Driver] [DB2/NT] SQL0413N 在数字数据类型转换期间发生溢出。
SQLSTATE=22003
E:\DatabaseApplication>
```

通过查看 STAFF 表中的数据，可以发现编号为 20 的部门有个成员原薪为 98357.50，而 SALARY 的数据类型为 DECIMAL，长度为 7，因此在增薪后其 SALARY 超出上界，发生溢出。

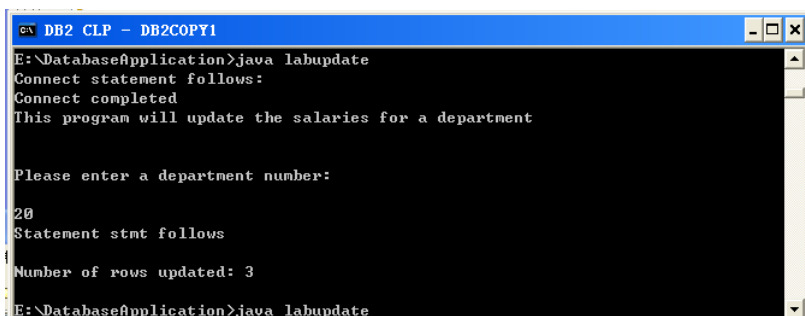
ID	NAME	DEPT	JOB	YEARS	SALARY
10	Sanders	20	Mgr	7	98,357.50

而在删除此员工或降低其 SALARY 后，再运行程序便没再发生这个问题，这也说明了是因为这个员工增薪后的 SALARY 过大导致的数据溢出。



```
DB2 CLP - DB2COPY1
E:\DatabaseApplication>java labupdate
Connect statement follows:
Connect completed
This program will update the salaries for a department

Please enter a department number:
20
Statement stmt follows
Number of rows updated: 4
E:\DatabaseApplication>java labupdate
```

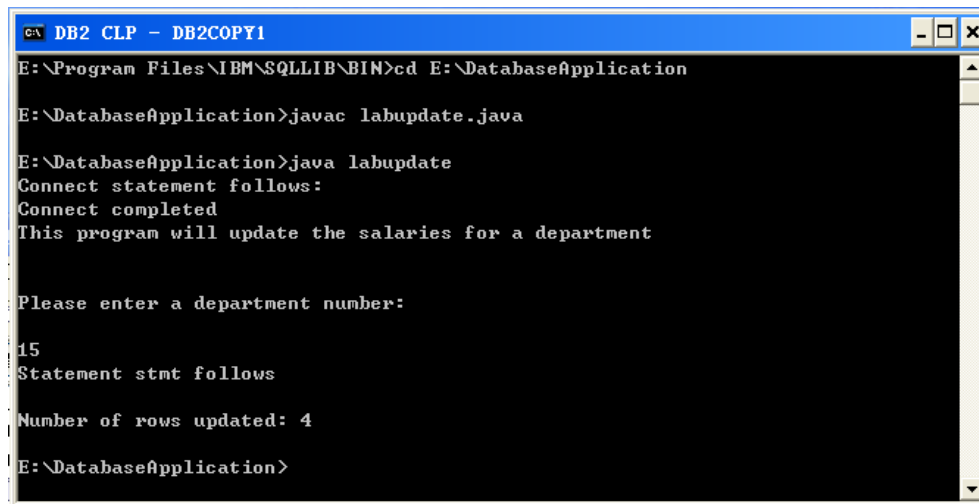


```
DB2 CLP - DB2COPY1
E:\DatabaseApplication>java labupdate
Connect statement follows:
Connect completed
This program will update the salaries for a department

Please enter a department number:
20
Statement stmt follows
Number of rows updated: 3
E:\DatabaseApplication>java labupdate
```

● 运行结果

1) 部门存在，且成功修改



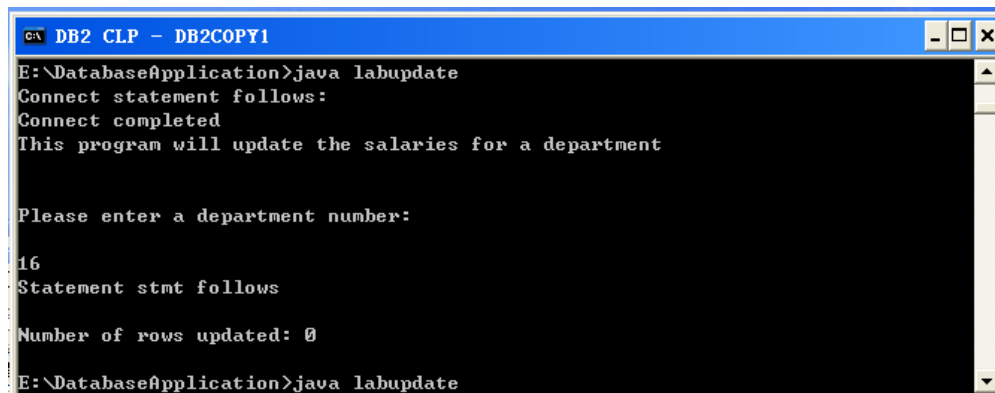
```
DB2 CLP - DB2COPY1
E:\Program Files\IBM\SQLLIB\BIN>cd E:\DatabaseApplication

E:\DatabaseApplication>javac labupdate.java

E:\DatabaseApplication>java labupdate
Connect statement follows:
Connect completed
This program will update the salaries for a department

Please enter a department number:
15
Statement stmt follows
Number of rows updated: 4
E:\DatabaseApplication>
```

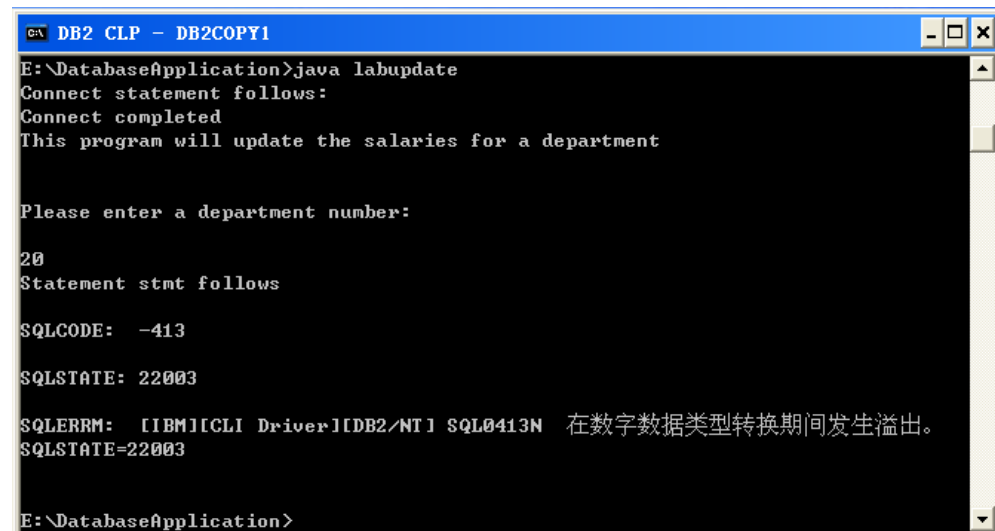
2) 部门不存在



```
DB2 CLP - DB2COPY1
E:\DatabaseApplication>java labupdate
Connect statement follows:
Connect completed
This program will update the salaries for a department

Please enter a department number:
16
Statement stmt follows
Number of rows updated: 0
E:\DatabaseApplication>java labupdate
```

3) 部门存在，但修改失败



```
DB2 CLP - DB2COPY1
E:\DatabaseApplication>java labupdate
Connect statement follows:
Connect completed
This program will update the salaries for a department

Please enter a department number:
20
Statement stmt follows
SQLCODE:  -413
SQLSTATE: 22003
SQLERRM:  [IBM][CLI Driver][DB2/NT] SQL0413N  在数字数据类型转换期间发生溢出。
SQLSTATE=22003
E:\DatabaseApplication>
```

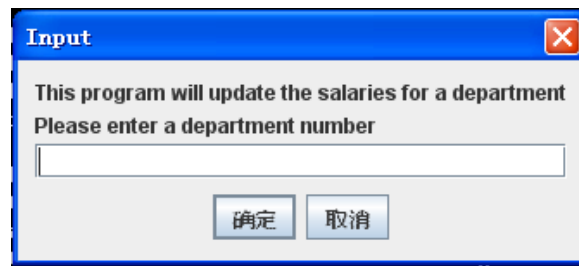
任务 4 将 labupdate.java 修改为 GUI 的形式，用户输入和修改结果的输出均通过调用 JOptionPane 类的适当方法实现。

- 调试过程

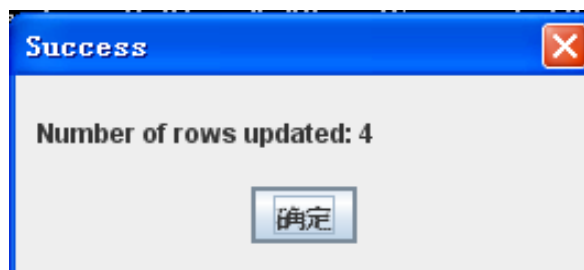
此次任务没有错误

- 运行结果

- 1) 输入界面



- 2) 输出界面（部门存在，且成功修改）



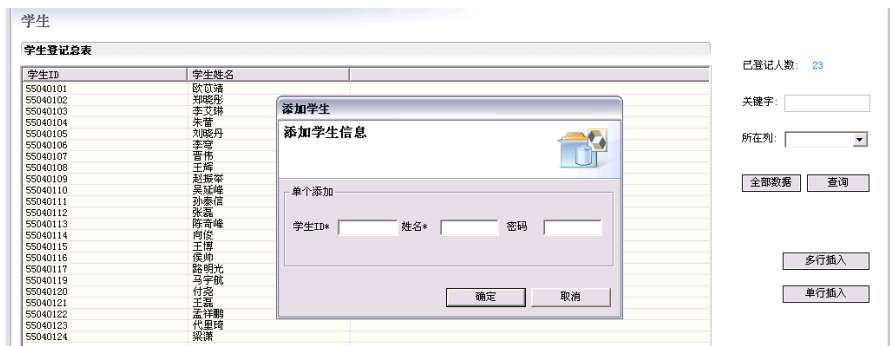
- 3) 输出界面（部门不存在）



- 4) 输出界面（部门存在，但修改失败）



任务 5 模仿下面截图设计并实现向表 **TEMPL** 插入行的操作，
GUI 界面要求实现单行插入、多行插入、通过子查询插入的功能。



● 调试过程

1) JDBC 语句错误

如下图所示，当使用 `PreparedStatement` 类型时，调用 `executeUpdate()` 方法不需要在里面加 `sql` 语句，以为在 `prepareStatement()` 中以加上了 `sql`。这是刚接触 JDBC 不熟悉语法而导致的错误。



2) Java 相关错误

因为对 Java 掌握得还不是很熟练，所以也犯了不少的错误。
其中一个印象比较深刻的是，在对 `String` 类型的语句进行初始化时，使用

null 和使用""不是等效的。

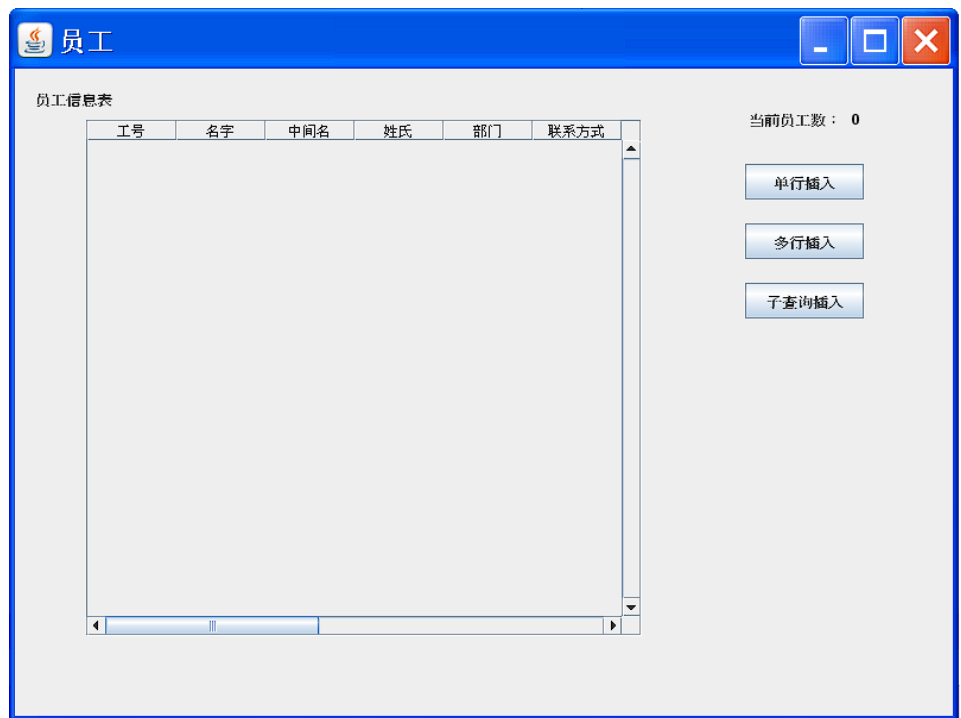
此外在对 String 类型的变量进行比较的时候，不论比较对象是另一个 String 类型变量，还是"xxxx"形式的字符串，都尽量使用 equals()进行比较较，我有个判断就是对""直接使用==进行比较，导致错误，浪费了大量时间。

● 运行结果

1) 公共部分

程序运行后会进入初始界面。

左侧区域对 TEMPL 表信息进行查看，右侧区域对 TEMPL 表进行插入操作，并现实当前 TEMPL 表内的员工数。



此外，为了增强程序的鲁棒型，我根据 TEMPL 表中的信息，对用户的输入进行了限制:

- 工号：主键，不可空，且必须为六位整数。
- 名字：不可空，且必须为字母。
- 中间名：可空，若是填写必须为一个字母。
- 姓氏：不可空，且必须为字母。
- 部门：可空，若是填写必须为一个大写字母加两个数字的形式，如 A00。
- 联系方式：可空，若是填写必须为四位整数。
- 入职日期：可空，若是填写必须为 xxxx-xx-xx 的形式，如 2023-12-31。
- 职务：可空，若是填写必须为字母，且不超过八位。
- 等级：不可空，且必须为两位整数。
- 性别：可空，选择下拉框内选项即可。
- 出生日期：可空，若是填写必须为 xxxx-xx-xx 的形式，如 2024-01-01。
- 薪资、奖金、佣金：可空，若是填写必须为不超过七位的整数（支持 xxxxxxxx.xx 形式，如 8888.00）

用户必须要满足以上条件才可成功插入数据，否则会弹窗提醒。

2) 单行插入

点击“单行插入”按钮便会进入“单行插入”界面。
在页面内输入正确的信息便能成功添加新员工信息。

The screenshot shows the 'Single Row Insert' dialog box overlaid on the 'Employee Information Table' application. The dialog box contains the following fields:

工号	名字	中间名	姓氏	部门	联系方式
工号	名字	中间名	姓氏	工作部门	联系方式
入职日期	工作职务	等级	性别	出生日期	薪资
奖金	佣金				

Buttons: 确定 (OK), 取消 (Cancel).

若输入的信息错误，则会通过弹窗告知错误的具体位置。

(这里用的是我姓名的拼音 **Chengjun Xiao**，工号因为只能 6 位所以是减去入学年份后的学号 **15240619**)。

The screenshot shows the 'Single Row Insert' dialog box with the following input data:

工号	名字	中间名	姓氏	部门	联系方式
150619	Chengjun	C	Xiao		qew

Buttons: 确定 (OK), 取消 (Cancel).

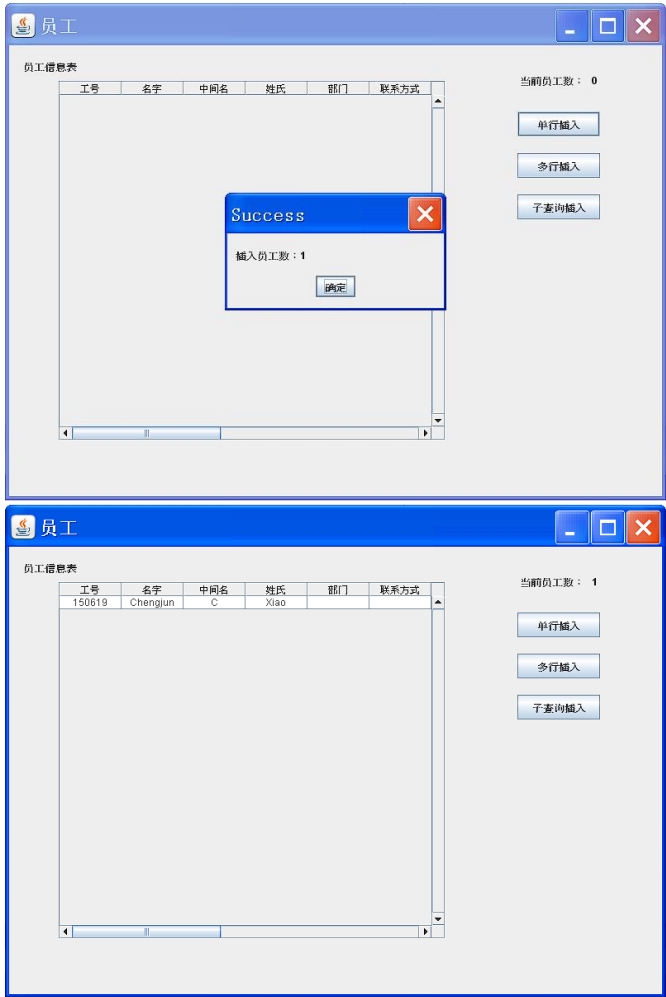
The screenshot shows the 'Error' dialog box with the following text:

Error

员工联系方式非法输入

确定 (OK)

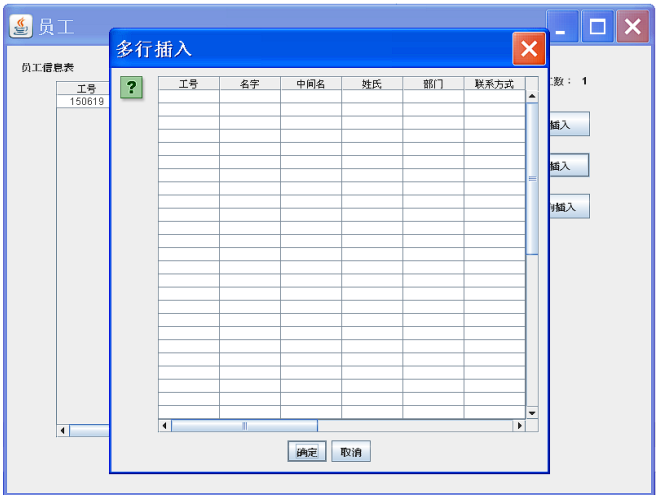
在成功添加员工信息后会弹窗提醒并自动返回初始界面，同时更新初始界面。



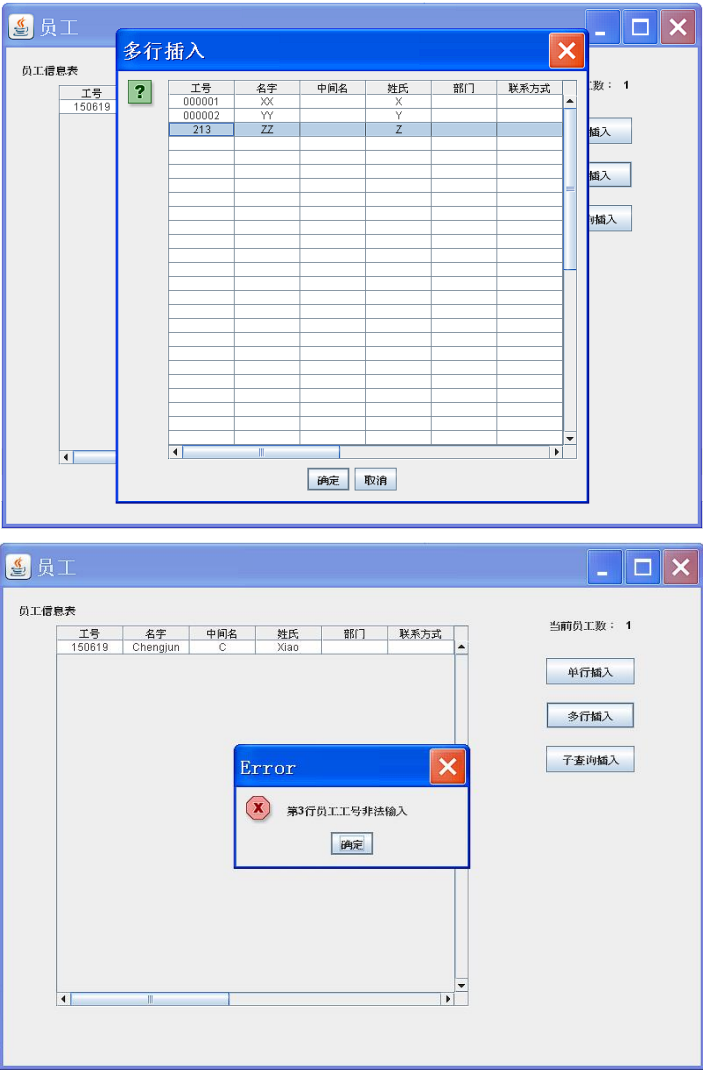
3) 多行插入

点击“多行插入”按钮便会进入“多行插入”界面。

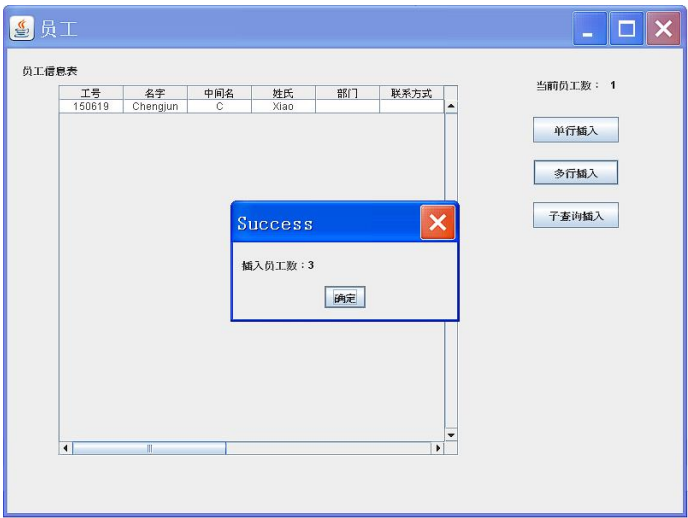
在页面内输入正确的信息便能成功添加新员工信息。同时，多行插入也支持通过复制粘贴来导入要添加的员工信息，比如直接从 word 文档中复制已有的员工信息表，将其粘贴至此页面。

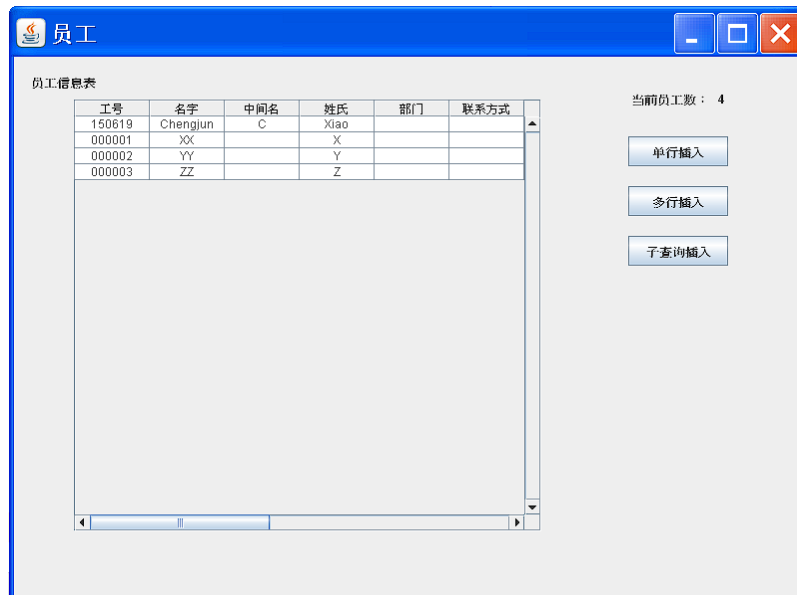


此外，若输入的信息有误，会通过弹窗具体说明错误的具体位置。同时，在弹窗提醒后并不会清空之前已输入的数据，用户根据弹窗提醒对已输入的数据进行修改即可。



在成功添加员工信息后便会弹窗提醒并自动返回初始界面，同时更新初始界面。





4) 子查询插入

点击“子查询插入”按钮便会进入“输入搜索条件”界面，通过输入的条件在 JLU.EMPLOYEE 表中查找，若没有限制条件则可直接关闭。



输入搜索条件后则进入“子查询界面”（这里以“部门为 A00”为查询条件）。

左侧为 JLU.EMPLOYEE 表中符合条件的信息，用户只需在左侧表格中选定需要的数据点击“添加”按钮就可将其加入到右侧表格，同时这一行会在左侧表格中删除。

若用户选择错误，可在右侧表选择不要的数据，点击“删除”按钮就可将其加入回左侧表格。

此外，在这种模式下是不允许对表格内的数据进行修改的，即使在表格内修改，最后插入到数据库表中的还是原来的数据。

子查询插入

?

工号	名字	中间名	姓氏	部门	联系方式
000110	VINCENZO	O	LUCCHESI	A00	3490
200010	DIAN	J	HEMMING	A00	3978
200120	GREG		ORLANDO	A00	2167

工号	名字	中间名	姓氏	部门	联系方式
000010	CHRISTINE	I	HAAS	A00	3978
000120	SEAN		O'CONNELL	A00	2167

添加

删除

确定 取消

员工

员工信息表

工号	名字	中间名	姓氏	部门	联系方式
150619	Chengjun	C	Xiao		
000001	XX		X		
000002	YY		Y		
000003	ZZ		Z		

当前员工数： 4

单行插入

多行插入

子查询插入

Success

插入员工数：2

确定

员工

员工信息表

工号	名字	中间名	姓氏	部门	联系方式
150619	Chengjun	C	Xiao		
000001	XX		X		
000002	YY		Y		
000003	ZZ		Z		
000010	CHRISTINE	I	HAAS	A00	3978
000120	SEAN		O'CONNELL	A00	2167

当前员工数： 6

单行插入

多行插入

子查询插入

任务 6 修改“任务 5”的程序，将结果集（增、改、查）中所有为空的列值均以“空”的形式显示。

● 调试过程

一开始我只用了 `ResultSet` 的 `wasNull()`，但结果就是如下图这样只有一部分为空的列值以“空”的形式显示。

这是因为 `wasNull()` 只能识别 `null`，对于全是空格的数据不会判为 `null`，只要使用 `trim()` 去掉字符串中的空格再进行判断即可解决此问题。



● 运行结果

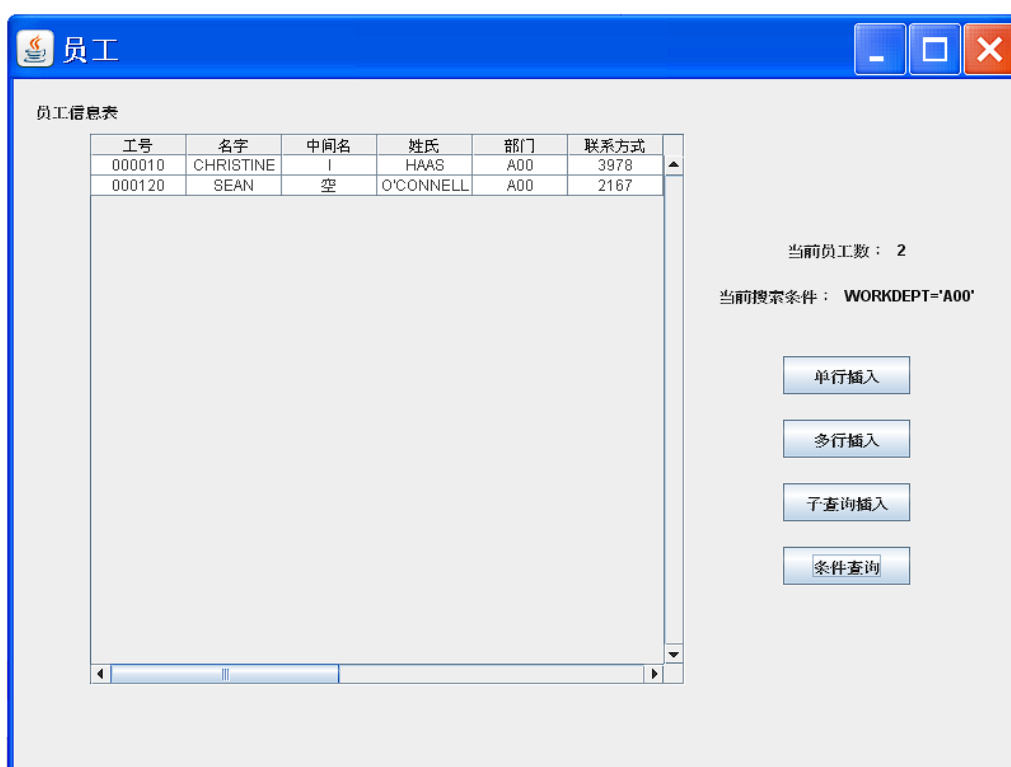
1) 所有为空的列值均以“空”的形式显示



2) 增加搜索功能

能根据输入的条件显示出符合条件的信息，这里以“部门为 A00”为搜索条件。

增加这个功能是因为看到题目中有个“结果集（增、改、查）”，以为除了“增”以外，这题还要顺带实现“改”和“查”的功能。但在实现完“查”这个功能后才发现任务八便是要求实现“改”，所以我可能对题目理解有点偏差，但既然已经实现了也就保留了这个功能。



任务 7 修改“任务 5”的程序，进行异常处理，要求以对话框的形式提示所有的违反约束、类型不匹配、溢出等操纵类错误。

● 调试过程

此次任务总体上没有什么错误。但值得一提的是，我一开始其实是使用 `SQLSTATE` 捕获 SQL 的异常，但在我寻找每种异常的过程中，我发现有些异常的 `ERRORSTATE` 为 `null`，所以我后来统一改用了 `ERRORCODE`。

另外，在任务五中，因为对 Java 的异常处理机制不太了解，我为了尽可能增强程序的鲁棒性，通过 `if - elseif - else` 的方式对输入的数据加以限制。这个方法的好处是在应用层（输入阶段）就确保能得到正确的数据，不会引起异常，并且能更灵活地设置输入限制，缺点就是代码编写较为繁琐（详见任务五或任务六代码）。

做到此次任务后才知道可以使用 Java 自带的异常处理机制，于是我将相关部分代码进行了重构（详见此次任务代码），不过任务五和任务六中还是保留了原先的方式，没有进行更改。

但我觉得单纯依靠 Java 的异常捕获，虽然在编程上更加简便，但也有一些局限性。比如对于 `String` 类型的数据，很难对输入加以限制。以 `EMPNO` 属性为例，观察数据表中已有的数据，不难看出真正符合标准的输入应该是六位整数，但因为其使用的是 `String` 类型，所以只要输入是非空且不超过六位的字符都是符合标准的，个人认为这在实际的使用中是不太合理的，这也是每个 `String` 类型的通病（不过或许也有对应的解决方式，但由于时间问题，我暂时无法进行进一步的探索）。

总之，由于时间关系，在提交报告时我仍有个别疑问尚未解决（不过对于完成实验报告没有什么影响），只能假期再深入研究了。

● 运行结果

1) 违反约束类

违反约束异常一：唯一性约束异常（主键重复）

在数据库中主键必须是唯一的，而工号（`EMPNO`）就是 `TEMPL` 中定义的主键。

从下图中可以看出，工号（`EMPNO`）为“000010”的员工信息已在表 `TEMPL` 中，此时若再次插入这条数据，则会造成主键重复，触发主键唯一性约束的异常。

员工

员工信息表

工号	名字	中间名	姓氏	部门	联系方式
150619	Chengjun	C	Xiao	空	空
000001	XX	空	X	空	空
000002	YY	空	Y	空	空
000003	ZZ	空	Z	空	空
000010	CHRISTINE	I	HAAS	A00	3978
000120	SEAN	空	O'CONNELL	A00	2167

当前员工数：6

子查询插入

工号	名字	中间名	姓氏	部门	联系方式
000020	MICHAEL	L	THOMPSON	B01	3476
000030	SALLY	A	KYAN	C01	4738
000050	JOHN	B	GEYER	E01	6789
000060	IRVING	F	STERN	D11	6423
000070	EVA	D	PULASKI	D21	7831
000080	EILEEN	W	HENDERS	E11	5498
000100	THEODORE	Q	SPENSER	E21	0972
000110	VINCENZO	O	LUCCHESSI	A00	3490
000120	SEAN	空	O'CONNELL	A00	2167
000130	DELORES	M	QUINTANA	C01	4578
000140	HEATHER	A	NICHOLLS	C01	1793
000150	BRUCE	空	ADAMSON	D11	4510
000160	ELIZABETH	R	PIANKA	D11	3782
000170	MASATOSHI	J	YOSHIMURA	D11	2680
000180	MARILYN	S	SCOUTTEN	D11	1682
000190	JAMES	H	WALKER	D11	2986
000200	DAVID	空	BROWN	D11	4501
000210	WILLIAM	T	JONES	D11	0942
000220	JENNIFER	K	LUTZ	D11	0672
000230	JAMES	J	JEFFERSON	D21	2094
000240	SALVATORE	M	MARINO	D21	3780
000250	DANIEL	S	SMITH	D21	0961
000260	SYBIL	P	JOHNSON	D21	8953
000270	MARIA	L	PEREZ	D21	9001
000280	ETHEL	R	SCHNEIDER	E11	8997

添加 删除 确定 取消

员工

员工信息表

工号	名字	中间名	姓氏	部门	联系方式
150619	Chengjun	C	Xiao	空	空
000001	XX	空	X	空	空
000002	YY	空	Y	空	空
000003	ZZ	空	Z	空	空
000010	CHRISTINE	I	HAAS	A00	3978
000120	SEAN	空	O'CONNELL	A00	2167

当前员工数：6

当前搜索条件：

单行插入 多行插入 子查询插入 条件查询

Error

工号不可重复

确定

违反约束异常二：不可空的属性输入为空

从下图中可以看出，等级（EDLEVEL）处为空，而 TEMPL 表中对其规定为不可为空，若插入则会引发此异常。

员工

员工信息表

工号	名字	中间名	姓氏	部门	联系方式
150619	Chengjun	C	Xiao	空	空
000001	XX	空	X	空	空
000002	YY	空	Y	空	空
000003	ZZ	空	Z	空	空
000010	CHRISTINE	I	HAAS	A00	3978
000120	SEAN	空	O'CONNELL	A00	2167

当前员工数：6

当前搜索条件：

单行插入 多行插入 子查询插入 条件查询

单行插入

工号 000001 名字 QQ 中间名

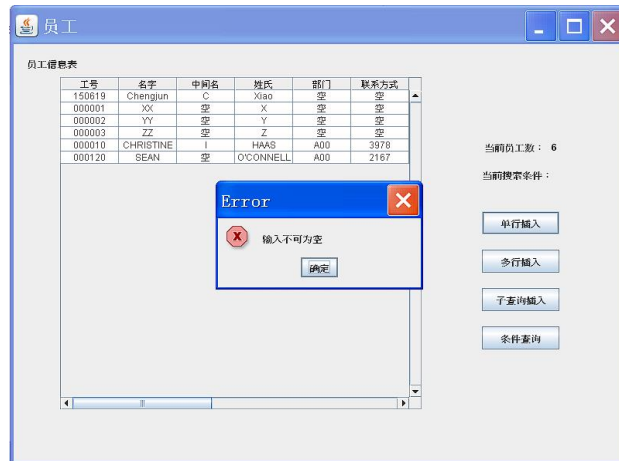
姓氏 Q 工作部门 联系方式

入职日期 工作职务 等级

性别 出生日期 薪资

奖金 佣金

确定 取消

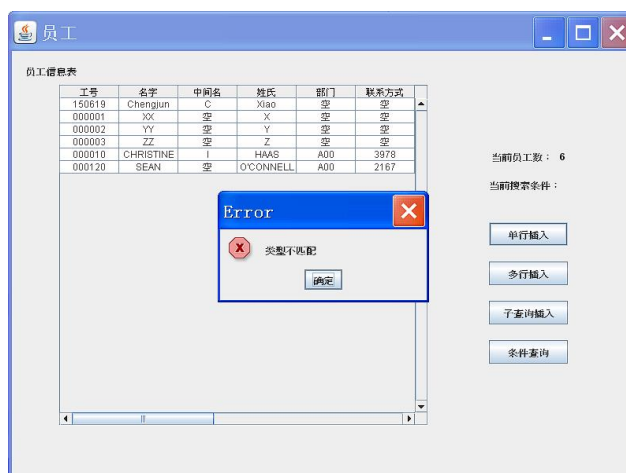
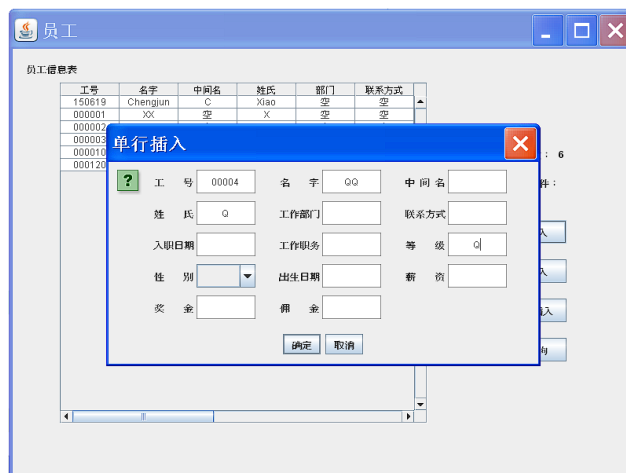


违反约束异常三：外键约束异常

虽然我在代码中实现了对外键约束异常的捕获，但在 `TEMPL` 表中应该是不存在与外键有关的问题，因此在对 `TEMPL` 表进行操作的过程中正常来讲是不会触发外键约束异常的。

2) 类型不匹配类

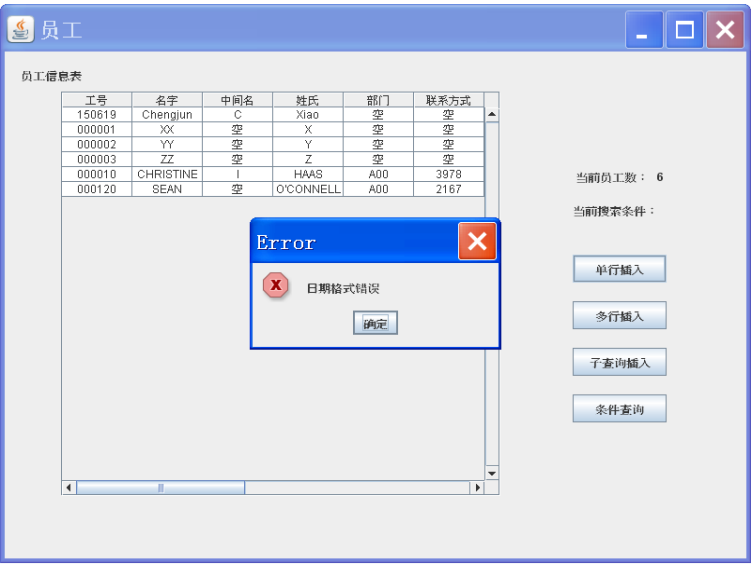
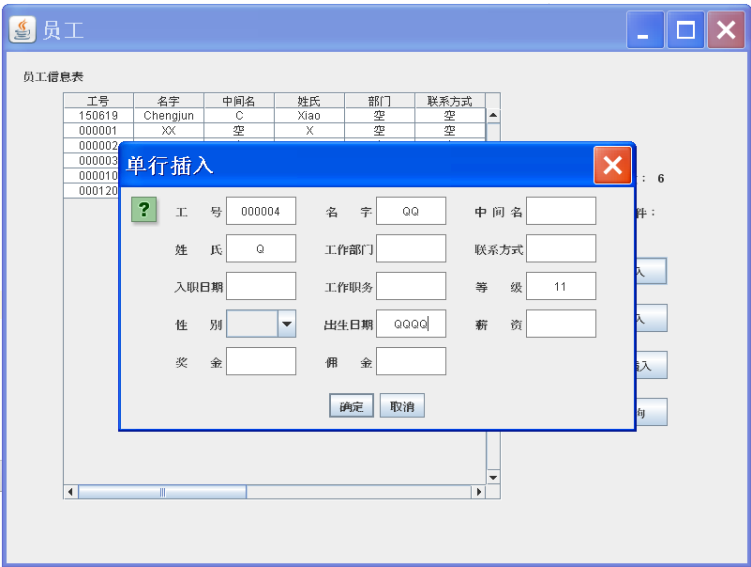
如等级 (`EDLEVEL`) 在数据库中为 `SMALLINT` 型，因此当输入为字符时便会引发类型不匹配的异常。



必须要说的是，虽然我在 `SQLException` 中有试图捕获这一异常，但在这里真正靠的是 `NumberFormatException` 捕获异常。因为在程序编写的过程中，我都统一将输入存为 `String` 类型，然后再在向数据库中插入时，进行数据类型的转换。因此即使数据类型不匹配也只是因为在 `Java` 中转换时失败而发出异常，而不会是在向数据库插入信息时发出类型不匹配的异常。

但按照我对题目的理解，应该是要在插入时捕获这一异常，因此我还在思考如何解决这一问题，这似乎意味着我要彻底改变当前的用来存储数据的数据类型。这也是我在上文提到的未解决的疑问之一。

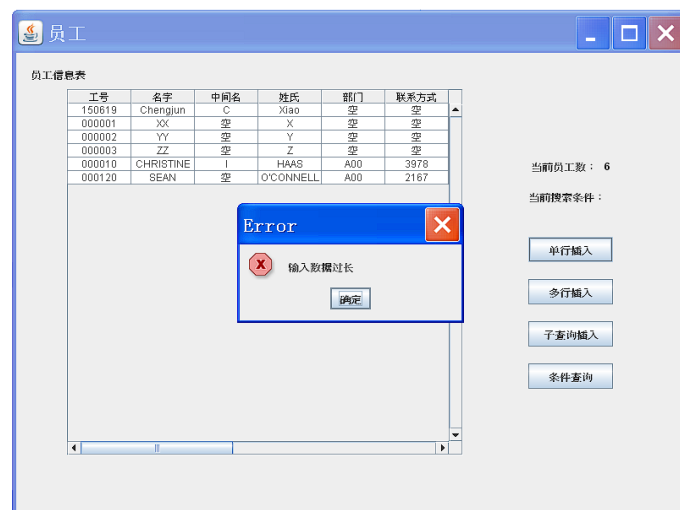
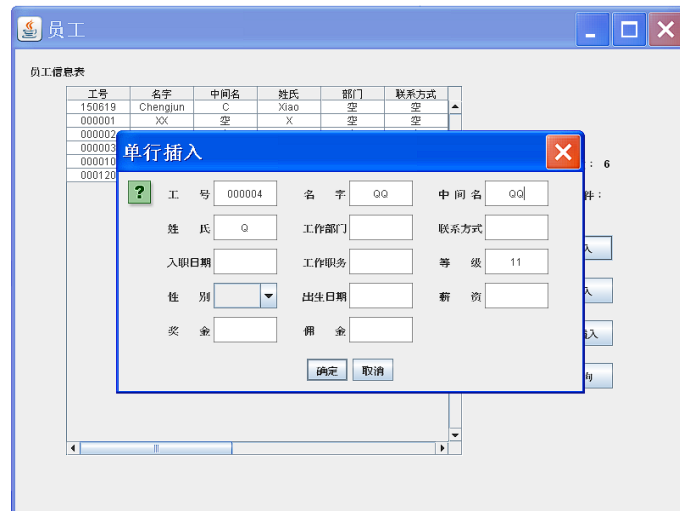
同理，两个与日期有关的信息我也是在 `Java` 中进行转换，若输入字符等不符合条件的信息，则会转换失败，引起 `IllegalArgumentException` 异常。



3) 溢出类

溢出异常一：String 类型的数据输入过长

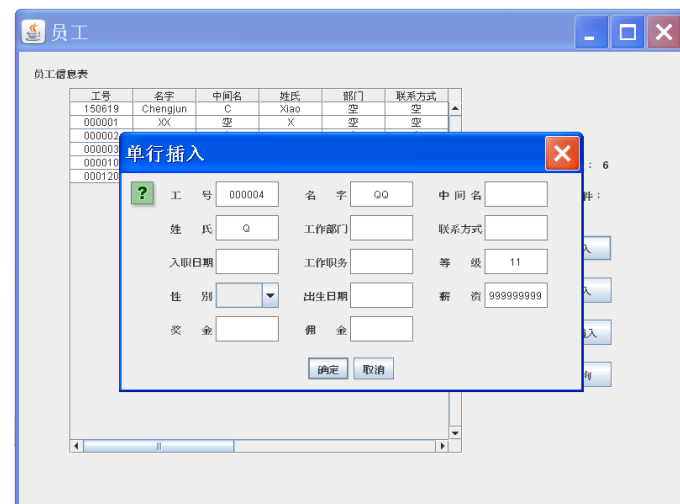
如下图所示，中间名（MIDINIT）长度在数据库中设定为一个字符，因此当输入为两个字符时便会产生溢出异常。

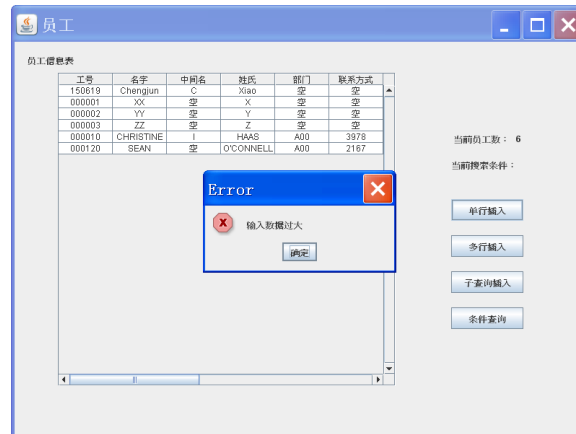


溢出异常二：数字类型的数据输入过大

数字类型的溢出和字符串类型的溢出在 `SQLException` 中是不同的，它们有着不同的 `SQLCODE`。

如下图所示，薪资（`SALARY`）在数据库中是 7 位 `DECIMAL`，因此在输入过大的情况下会产生溢出，触发异常。





任务 8

1. 完成实验文档 P67 页中的“Section 1 - Modify the Program labstaff1.java”。
2. 修改“任务 5”的程序，实现对结果集中任意行、任意列的修改操作。

● 调试过程

此次任务没有错误

不过需要说的是，关于任务八的第二个任务，我一开始先是使用“FOR UPDATE”和“WHERE CURRENT OF”的方法实现了一遍。但在使用 and 调试的过程中，我觉得使用这种方法在设计和使用上不太符合我个人的想法，于是换了一种方法实现这个修改功能。而提交的代码中也只包含当前方法的代码，并不含有使用“FOR UPDATE”和“WHERE CURRENT OF”实现的源代码。

● 运行结果

1) 任务一

```

DB2 CLP - DB2COPY1

E:\DatabaseApplication>java labstaff1

Set AutoCommit off

Autocommit off

Enter the Department number

20

ID      NAME      SALARY
-----
10      Sanders  98357.50
20      Pernal   78171.25
80      James    43504.60
190     Sneider  34252.75

E:\DatabaseApplication>

```

2) 任务二

可直接在初始界面上进行更改，并且能同时更改多个任意行与任意列，只要在更改完成后点击“确认修改”按钮即可提交修改。（我这里同时修改了第二行的员工名字和第三行的员工姓氏）

工号	名字	中间名	姓氏	部门	联系方式
150619	Chengjun	C	Xiao	空	空
000001	XX	空	X	空	空
000002	YY	空	Y	空	空
000003	ZZ	空	Z	空	空
000010	CHRISTINE	I	HAAS	A00	3978
000120	SEAN	空	O'CONNELL	A00	2167

Success

修改员工数: 2

确定

同时，修改后的数据也要符合数据库的要求，否则会引起异常。顺带一提，任务八中的异常处理机制是继承任务七的，与任务五、任务六中的不同。

Error

输入数据过长

确定

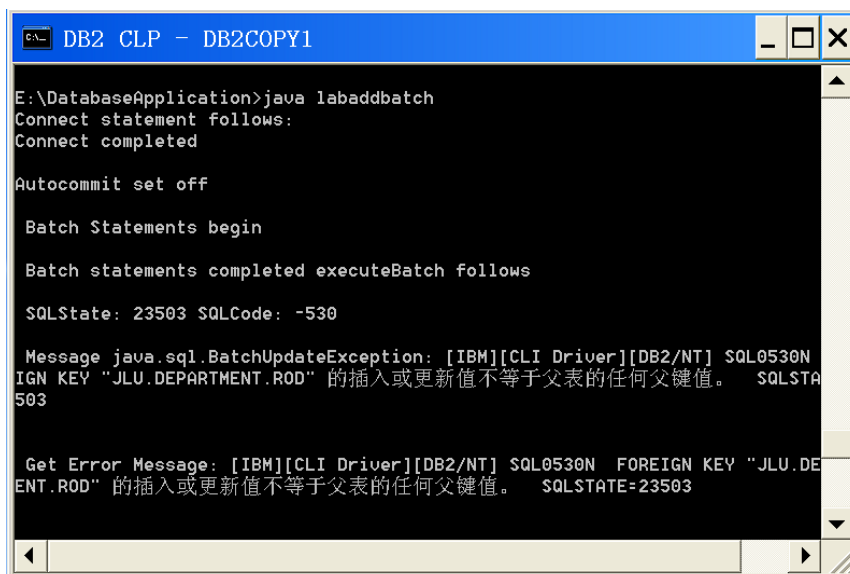
任务 9 完成实验文档中的“Section 3.2、3.3 和 3.4”的实验内容。

● 调试过程

在完成 Section 3.3 的时候，发生了如下错误：“JLU.DEPARTMENT.ROD”的插入或更新值不等于父表的任何父键值。

这是因为 JLU 创建的 DEPARTMENT 表中包含一个外键 "ROD" 列，但该列的值不在父表的主键列中，这样直接插入违反了外键约束。经过查看可以发现“ROD”绑定的是 DEPARTMENT 中的 ADMRDEPT，除此之外还有一个外键“ROE”。

对于这种情况可以把提供的数据改为符合外键约束的数据。不过为了不改动提供的数据，在再三确认过不会对其他任务造成影响后，我这里选择直接删去了这两个外键，当然，在实际操作的时候这样的做法显然是不可取的。



```
DB2 CLP - DB2COPY1

E:\DatabaseApplication>java labadddbatch
Connect statement follows:
Connect completed

Autocommit set off

Batch Statements begin

Batch statements completed executeBatch follows

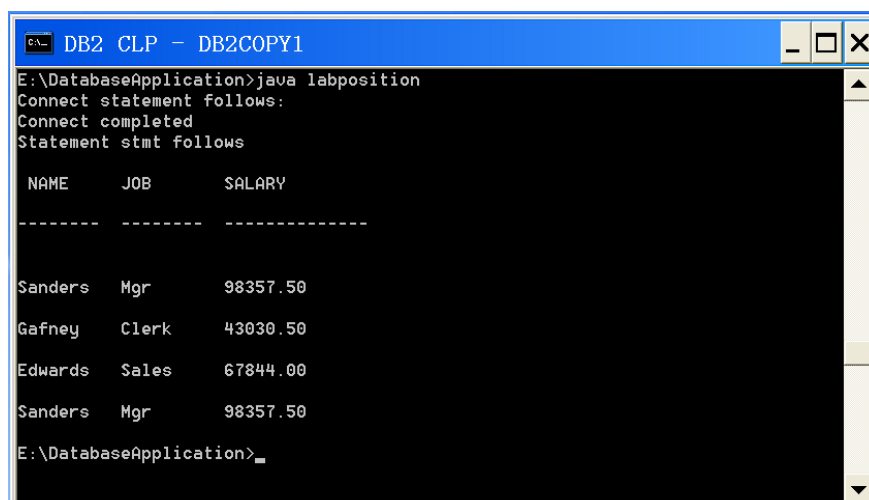
SQLState: 23503 SQLCode: -530

Message java.sql.BatchUpdateException: [IBM][CLI Driver][DB2/NT] SQL0530N
IGN KEY "JLU.DEPARTMENT.ROD" 的插入或更新值不等于父表的任何父键值。 SQLSTA
503

Get Error Message: [IBM][CLI Driver][DB2/NT] SQL0530N FOREIGN KEY "JLU.DE
ENT.ROD" 的插入或更新值不等于父表的任何父键值。 SQLSTATE=23503
```

● 运行结果

1) Section 3.2



```
DB2 CLP - DB2COPY1

E:\DatabaseApplication>java labposition
Connect statement follows:
Connect completed
Statement stmt follows

  NAME      JOB      SALARY
-----
Sanders    Mgr      98357.50
Gafney     Clerk     43030.50
Edwards    Sales     67844.00
Sanders    Mgr      98357.50

E:\DatabaseApplication>
```

2) Section 3.3

打开表 — DEPARTMENT

JLU-29E0FDF8B31 - DB2 - SAMPLE - JLU.DEPARTMENT

采用定位 UPDATE 和定位 DELETE 编辑这些结果。使用“工具设置”笔记本来更改编辑格式。

DEPTNO	DEPTNAME	MGRNO
A00	SPIFFY COMPU...	000010
B01	PLANNING	000020
C01	INFORMATION ...	000030
D01	DEVELOPMEN...	
D11	MANUFACTURI...	000060
D21	ADMINISTRATI...	000070
E01	SUPPORT SER...	000050
E11	OPERATIONS	000090
E21	SOFTWARE SU...	000100
F22	BRANCH OFFIC...	
G22	BRANCH OFFIC...	
H22	BRANCH OFFIC...	
I22	BRANCH OFFIC...	
J22	BRANCH OFFIC...	

添加行(A)

删除行(D)

落实(M) 回滚(R) 过滤(E) 访存更多行(S)

☐ 自动落实更新(I) 内存中的 14 行

关闭 帮助

```
DB2 CLP - DB2COPY1

E:\DatabaseApplication>java labadddbatch
Connect statement follows:
Connect completed

Autocommit set off

Batch Statements begin

Batch statements completed executeBatch follows

Update row count 1
Update row count 1
Update row count 1
Update row count 1
Update row count 1

E:\DatabaseApplication>
```


打开表 — DEPARTMENT

JLU-29E0FDF8B31 - DB2 - SAMPLE - JLU.DEPARTMENT

采用定位 UPDATE 和定位 DELETE 编辑这些结果。使用“工具设置”笔记本来更改编辑格式。

DEPTNO	DEPTNAME	MGRNO
D21	ADMINISTRATI...	000070
E01	SUPPORT SER...	000050
E11	OPERATIONS	000090
E21	SOFTWARE SU...	000100
F22	BRANCH OFFIC...	
G22	BRANCH OFFIC...	
H22	BRANCH OFFIC...	
I22	BRANCH OFFIC...	
J22	BRANCH OFFIC...	
BT6	BATCH6 NEWY...	BBBBB1
BT7	BATCH7 NEWY...	BBBBB2
BT8	BATCH8 NEWY...	BBBBB3
BT9	BATCH9 NEWY...	BBBBB4
BTA	BATCH10 NEW...	BBBBB5

添加行(A) 删除行(D)

落实(M) 回滚(R) 过滤(E) 访存更多行(S)

☐ 自动落实更新(I) 内存中的 19 行

关闭 帮助

3) Section 3.4

DB2 CLP - DB2COPY1

```

E:\DatabaseApplication>java labtables

Catalog Name: null Schema Name: JLU Table Name: ACT
Catalog Name: null Schema Name: JLU Table Name: CATALOG
Catalog Name: null Schema Name: JLU Table Name: CL_SCHED
Catalog Name: null Schema Name: JLU Table Name: CUSTOMER
Catalog Name: null Schema Name: JLU Table Name: DEPARTMENT
Catalog Name: null Schema Name: JLU Table Name: EMPLOYEE
Catalog Name: null Schema Name: JLU Table Name: EMPMDC
Catalog Name: null Schema Name: JLU Table Name: EMPPROJACT
Catalog Name: null Schema Name: JLU Table Name: EMP_PHOTO
Catalog Name: null Schema Name: JLU Table Name: EMP_RESUME
Catalog Name: null Schema Name: JLU Table Name: INVENTORY
Catalog Name: null Schema Name: JLU Table Name: IN_TRAY
Catalog Name: null Schema Name: JLU Table Name: ORG

```

任务 10 以 GUI 的形式实现 sample 数据库中，表 emp_photo 的 picture 列的查询和插入。

● 调试过程

1) 外键约束问题

在进行图片插入的过程中，发生的问题与任务九中一样，是由外键约束导致的异常，不过只要合理选择 EMPNO 即可解决此问题。

另外，由于 EMPNO 为外键，且主键是 EMPNO+PHOTO_FORMAT，因此主键较为有限，经过几次操作后就可能导致主键重复，要进行一些删除操作。

2) 路径问题

在进行图片查询的过程中，我通过文件选择器选择文件保存的位置，但忘记在后面加上图片的名字，导致一直出现“拒绝访问”的异常。

我一开始主观地认为是文件夹的权限存在问题，在这一方向上耗费了大量时间精力，但毫无成效。在我改变了方向，使用其他关键词进行搜索后，一下发现了问题所在。

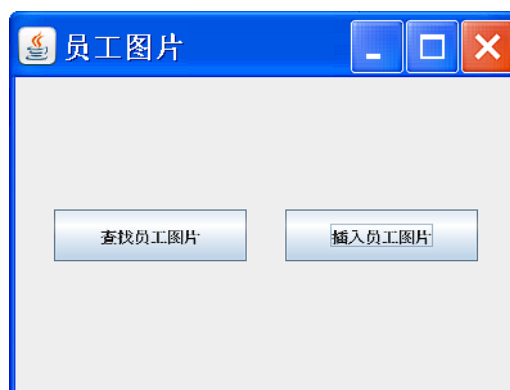
经历此次毒打后，我深刻认识到：在一个方向一直努力却毫无收获时，该认真思考一下是不是努力的方向错了，而不是一直埋头死磕。



● 运行结果

1) 初始界面

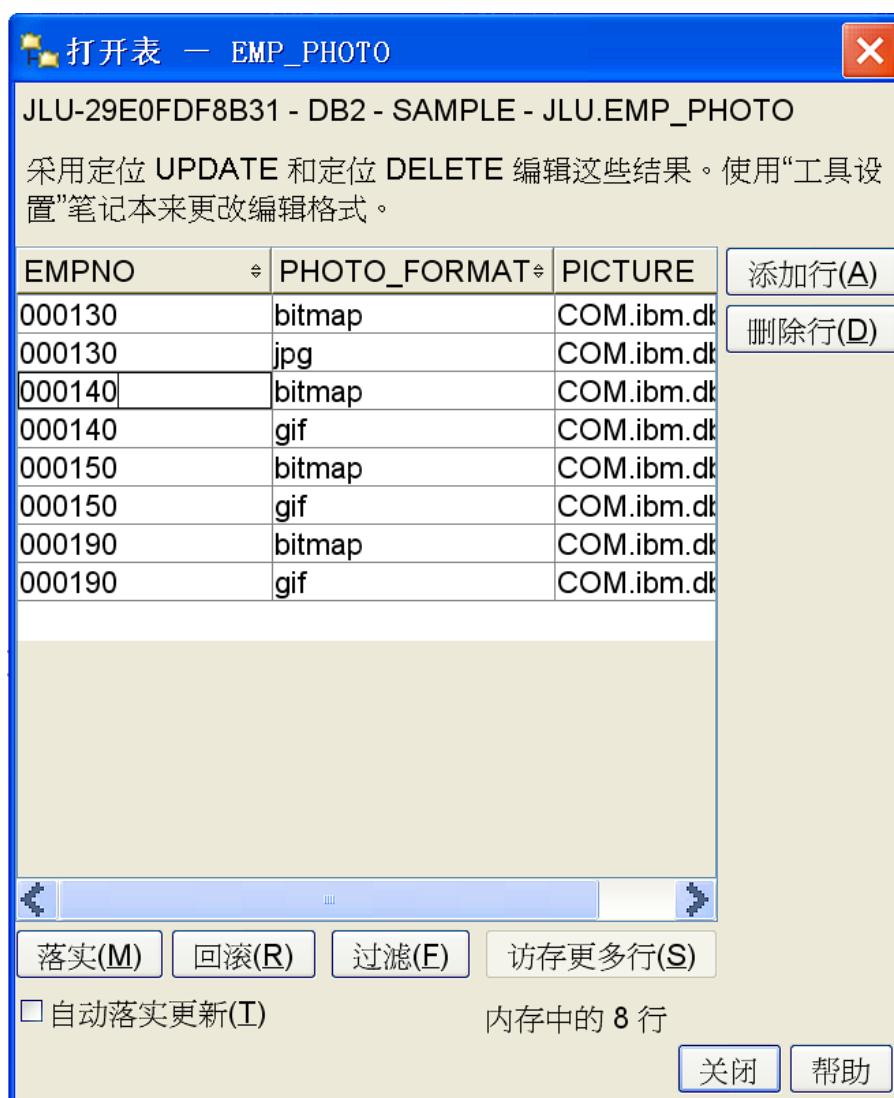
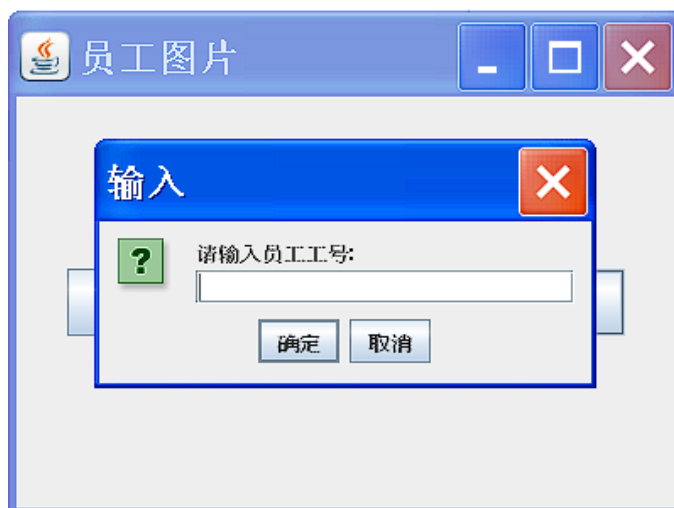
初始界面较为简单，只有两个按钮，分别是“查找员工图片”和“插入员工图片”。



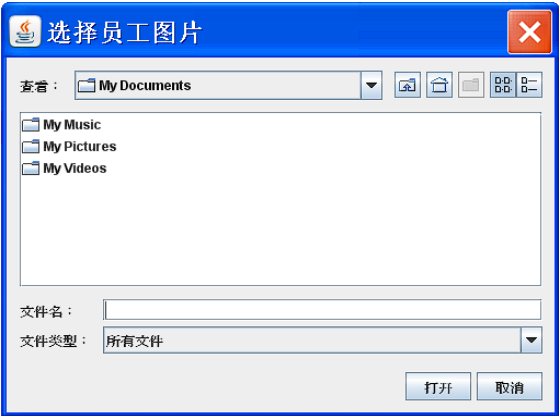
2) 插入图片界面

点击“插入员工图片”按钮后即可进入“插入图片界面”。

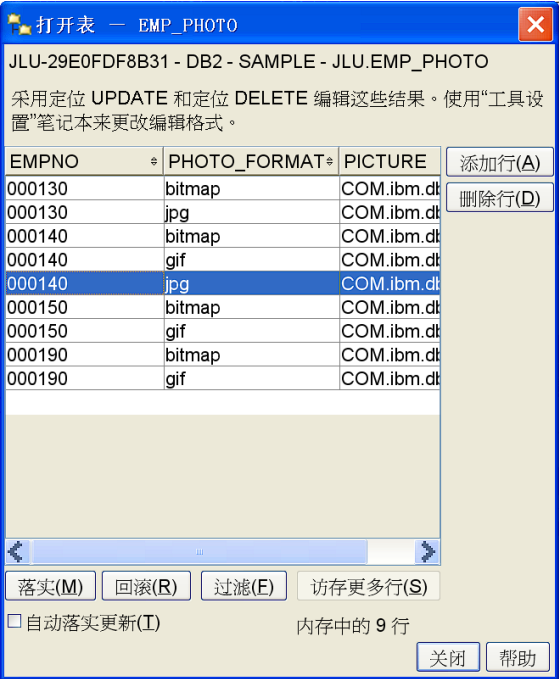
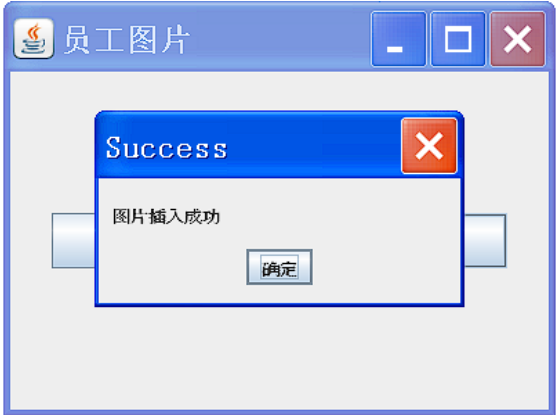
首先要输入插入图片的员工工号。需要注意的是，这里就是会引起“外键约束”异常的地方，要选择表中已有 EMPNO 才行（这里选择 000140）。



输入员工工号后就是进入图片选择界面，寻找到要插入的图片即可（这里使用了系统中自带的图片）。



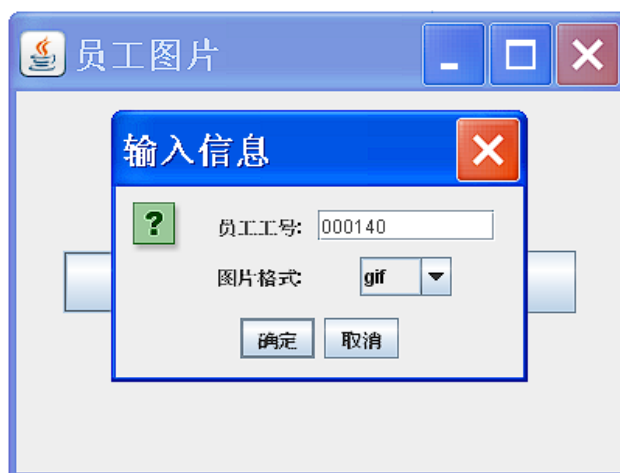
点击“打开”按钮后便插入成功。另外，即使在前一步输入“输入员工工号”时，所输入的工号不满足外键约束，程序也不会报错，因为我没有在这设置检测步骤，但是会在选择完图片进行插入的时候报错。



3) 查询图片界面

点击“查询员工图片”按钮后即可进入“查询图片界面”。

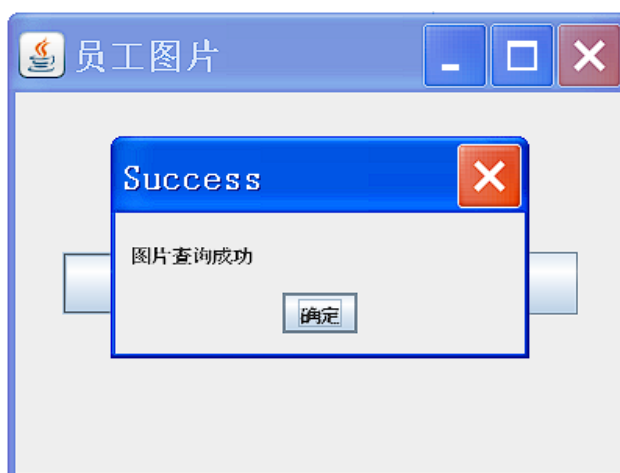
首先是输入查询图片的员工工号，和选择所需要的图片格式。（这里选择“000140”工号和“.gif”格式）。

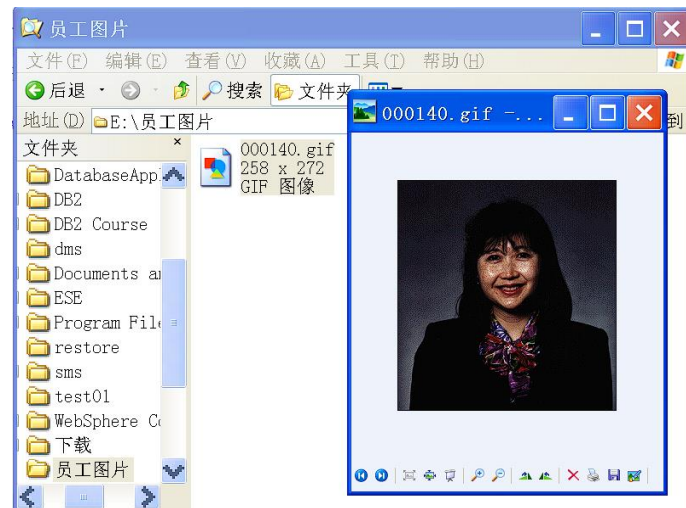


在输入员工工号和图片格式后就进行保存位置的选择。



点击“打开”按钮后，若图片存在，则提示查找成功，并可在选择的保存位置查看图片。





若图片不存在，则提示“未查找到图片”。（还有一种，就是如“调试过程”中展现的发生异常导致程序无法正确运行的报错，这里就不单独展示了）



任务 11 将文档 cg113inst.pdf 中 P461 页的代码补充完整，实现 CLOB 的检索。

- 调试过程

此次任务没有错误

- 运行结果



```
DB2 CLP - DB2COPY1
E:\DatabaseApplication>javac MyClob.java
E:\DatabaseApplication>java MyClob
Employee Number: 000130

Resume Format: ascii

Resume:

Resume: Delores M. Quintana

Department Information

Employee Number:    000130
Dept Number:       C01
Manager:           Sally Kwan
Position:          Analyst
Phone:             (208) 385-4578
Hire Date:         1971-07-28

Education

1965                Math and English, B.A.
                   Adelphi University

1960                Dental Technician
                   Florida Institute of Technology

Work History

10/91 - present     Advisory Systems Analyst
                   Producing documentation tools for engineering
                   department.

12/85 - 9/91        Technical Writer
                   Writer, text programmer, and planner.

1/79 - 11/85        COBOL Payroll Programmer
                   Writing payroll programs for a diesel fuel
                   company.

Interests

o   Cooking
o   Reading
o   Sewing
o   Remodeling

E:\DatabaseApplication>
```