CSCE 4523 Database Management Systems

Homework 4

Due: Tuesday, March 31 at 11:59pm

By: David Russell

Objectives

The objective of this homework was to develop a small database interface using Java that could apply several different specified operations to the underlying mysql database. The database schema and the tuples in the tables came from Homework 3. The goal of this assignment is to understand and practice using database connectors in a high level programming language like C++ or Java. As mentioned, my approach used Java.

I used an AUTO_INCREMENT tag for all of my Primary keys, and set the AUTO_INCREMENT for the respective tables to start at a particular integer corresponding to the conventions that were in the existing data. For example, Clients start at 101, Agents at 201, etc. This makes sense until there are more than 100 tuples. Even then, it will still work, it just won't be possibly to identify an ID's corresponding table by the leading digit only.

Approach

I used Java to implement the program. In addition to the classes and methods I wrote myself, I used most of the methods in Dr. Susan Gauche's java example. I used 4 class files, JdbcApp contains the "Main" program, the menu and the logic for closing and opening connections, running and printing general queries and also initializing the database for testing. Agents contains methods specific to the AGENTS table in the database, such as finding agents and adding agents and helper methods. Likewise the class files Policies and Clients have their respective methods.

The menu contains the operations, and those are ENUM apply() methods which call the respective methods in the other classes. I tried to keep my methods relatively short where possible and split them up into seperate functions where it made sense.

Furthermore, I tried to catch obvious errors like alphabet and punctuation characters being inputted where a floating point number or integer should be, printing an error, and halting the selected database operation and returning to the menu. I did this through a combination of input checking with if statements and try/catch blocks. Some of the methods will return certain inputs like empty strings if they encounter such an error, which is used as a signal for a calling method to terminate.

Results

The entire program basically had to throw IOExceptions because nearly everything was touching RandomAccessFiles. The efficiency of my program is similar to what was discussed in class—O(log(n)) for search, O(n) for overflow search, O(1) for addRecord and O(n) for merging the database.

I found that the biggest challenge was keeping track of whether or not I have hit the end of a particular file, whether there were deleted records I would come across, and handling them.

In particular, I had some hiccups with connecting different methods that were designed to ignore deleted records and having them basically go into an endless loop because one method would hand the other methods a reference to a deleted file and then the other method would skip to the end of a series of deleted record and hand back the original reference, which was not where the end of the deleted records was. I fixed this by making sure the deleted record handling was concentrated in one place.

Testing

For testing, I took advantage of the approach suggested in the Java example of filling out an initialize database method and filled it with the data from Homework 3. I made a "test run" in the main method that contained several calls to each of the operations so I could test valid and invalid input without having to enter each operation. When I wasn't testing, I commented this block out. I will include the commented out test run in the main method for review.

I tested my program by mapping out everywhere a table was read or written and seeing how that affected the database. That meant initializing the database, listing agents, adding an agent, listing the agents again, purchasing a policy and listing clients and purchases policies, deleting a policy and listing the policies and so on. As I mentioned in the approach section, I checked for proper input everywhere improper input could throw a SQL exception, but also where it made sense.

So for instance, if a purchase_ID was not found when canceling a policy, an appropriate message would be displayed.

If you enter an improper zipcode (less than 00001 or greater than 99999 or not a number) then the operation is canceled, same for amount when purchasing a policy. When records are not found, an appropriate message is printed.

I've included a typescript which demonstrates that.

```
Script started, file is typescript
darusse@turing:~/hw4/ideProj/src$ javac *.java
```

```
darusse@turing:~/hw4/ideProj/src$ java -cp .:mysql-connector-java-5.1.40-bin.jar JdbcApp
```

ADD AGENT FOR CITY SUCCESS

```
Please enter the name of a city:

SCHENECTADY

Please enter the name of the new Agent:

PERRY

Please enter the city of the new Agent:

SCHENECTADY

Please enter the 5 digit zip code of the new Agent:

12345

INSERT into AGENTS (A_NAME, A_CITY, A_ZIP) values ('PERRY','SCHENECTADY',12345)

A_ID,  A_NAME,  A_CITY,  A_ZIP
211,  PERRY,  SCHENECTADY,  12345
```

ADD AGENT FOR CITY FAILURE (INVALID ZIPCODE)

```
Please enter the name of a city:

SCHENECTADY

Please enter the name of the new Agent:

TERRY

Please enter the city of the new Agent:

SCHENECTADY

Please enter the 5 digit zip code of the new Agent:

93939393939
The zipcode you entered is not valid
A_ID,  A_NAME,  A_CITY,  A_ZIP
211,  PERRY,  SCHENECTADY,  12345
```

ADD AGENT FOR CITY SUCCESS

```
Please enter the name of a city:

SCHENECTADY
```

Please enter the name of the new Agent:

BERRY

Please enter the city of the new Agent:

SCHENECTADY

Please enter the 5 digit zip code of the new Agent:

12345

INSERT into AGENTS (A_NAME, A_CITY, A_ZIP) values ('BERRY','SCHENECTADY',12345)

A_ID, A_NAME, A_CITY, A_ZIP
211, PERRY, SCHENECTADY, 12345
212, BERRY, SCHENECTADY, 12345

LIST POLICIES BY AGENT SUCCESS

Please enter the name of the agent whose policies sold you'd like to see:

BRYAN

Please enter the city of the agent whose policies sold you'd like to see:

ROGERS

Query:
SELECT A_NAME, TYPE, COMMISSION_PERCENTAGE from POLICIES_SOLD, AGENTS, POLICY WHERE POLICY.POLICY_ID = POLICIES_SOLD.POLICY_ID AND AGENT_ID=A_ID AND A_NAME="BRYAN" AND A_CITY="ROGERS"

Result:
A_NAME, TYPE, COMMISSION_PERCENTAGE
BRYAN, HOME, 10
BRYAN, HEALTH, 7

LIST POLICIES BY AGENT FAILURE (NO RESULTS TO SHOW YET)

Please enter the name of the agent whose policies sold you'd like to see:

PERRY

Please enter the city of the agent whose policies sold you'd like to see:

SCHENECTADY
No results found for that agent name and city combo.

PURCHASE POLICY SUCCESS (NEW AGENT)

Please enter your name:

JUDGE

Please enter your city:

SCHENECTADY

Please enter your 5-digit zip code:

12345

INSERT into CLIENTS (C_NAME, C_CITY, C_ZIP) values ('JUDGE','SCHENECTADY',12345)

A_ID, A_NAME, A_CITY, A_ZIP
211, PERRY, SCHENECTADY, 12345
212, BERRY, SCHENECTADY, 12345

Please enter the TYPE of policy you'd like to purchase:

VEHICLE

Query:
SELECT * from POLICY WHERE TYPE ="VEHICLE"

Result:
POLICY_ID, NAME, TYPE, COMMISSION_PERCENTAGE
305, UNITEDCAR, VEHICLE, 9

Please enter the dollar amount of the policy you'd like to purchase:

3000.00

INSERT into POLICIES_SOLD (AGENT_ID, CLIENT_ID, POLICY_ID, DATE_PURCHASED, AMOUNT) values ('211','120','305', DATE '2020-03-29','3000.0')

LIST POLICIES BY AGENT SUCCESS (NEW AGENT, NEW POLICY)

Please enter the name of the agent whose policies sold you'd like to see:

PERRY

Please enter the city of the agent whose policies sold you'd like to see:

SCHENECTADY

Query:
SELECT A_NAME, TYPE, COMMISSION_PERCENTAGE from POLICIES_SOLD, AGENTS, POLICY WHERE POLICY.POLICY_ID = POLICIES_SOLD.POLICY_ID AND AGENT_ID=A_ID AND A_NAME="PERRY" AND A_CITY="SCHENECTADY"

Result:
A_NAME, TYPE, COMMISSION_PERCENTAGE
PERRY, VEHICLE, 9

Query:
SELECT * FROM POLICIES_SOLD

CANCEL POLICY SUCCESS

Result:
PURCHASE_ID, AGENT_ID, CLIENT_ID, POLICY_ID, DATE_PURCHASED, AMOUNT
401, 204, 106, 303, 2020-01-02, 2000.00
402, 201, 105, 305, 2019-08-11, 1500.00
403, 203, 106, 301, 2019-09-08, 3000.00
404, 207, 101, 305, 2019-06-21, 1500.00
405, 203, 104, 302, 2019-11-14, 4500.00
406, 207, 105, 305, 2019-12-25, 1500.00
407, 205, 103, 304, 2020-10-15, 5000.00
408, 204, 103, 304, 2020-02-15, 5000.00
409, 203, 103, 304, 2020-01-10, 5000.00
410, 202, 103, 303, 2020-01-30, 2000.00
418, 211, 120, 305, 2020-03-29, 3000.00
Please enter the Purchase ID of the policy you'd like to cancel:
418
Policy with purchase ID 418 canceled.

PURCHASE POLICY SUCCESS (EXISTING AGENT)

Please enter your name:

JEFF

Please enter your city:

ROGERS

Please enter your 5-digit zip code:

72762

INSERT into CLIENTS (C_NAME, C_CITY, C_ZIP) values ('JEFF','ROGERS',72762)

A_ID, A_NAME, A_CITY, A_ZIP
204, BRYAN, ROGERS, 78291
207, SMITH, ROGERS, 78291

Please enter the TYPE of policy you'd like to purchase:

HOME

Query:

```
SELECT * from POLICY WHERE TYPE ="HOME"

Result:
POLICY_ID,  NAME,  TYPE,  COMMISSION_PERCENTAGE
303,  WELLCARE,  HOME,  10

Please enter the dollar amount of the policy you'd like to purchase:

3.00

INSERT into POLICIES_SOLD (AGENT_ID, CLIENT_ID, POLICY_ID, DATE_PURCHASED, AMOUNT) values
('204','121','303', DATE '2020-03-29','3.0')
```

CANCEL POLICY FAILURE (NO MATCHING PURCHASE_ID), BUT SHOWS SUCCESS CANCELING PREVIOUS POLICY AND CREATING NEW POLICY

```
--------------------------------
Query:
SELECT * FROM POLICIES_SOLD

Result:
PURCHASE_ID,  AGENT_ID,  CLIENT_ID,  POLICY_ID,  DATE_PURCHASED,  AMOUNT
401,  204,  106,  303,  2020-01-02,  2000.00
402,  201,  105,  305,  2019-08-11,  1500.00
403,  203,  106,  301,  2019-09-08,  3000.00
404,  207,  101,  305,  2019-06-21,  1500.00
405,  203,  104,  302,  2019-11-14,  4500.00
406,  207,  105,  305,  2019-12-25,  1500.00
407,  205,  103,  304,  2020-10-15,  5000.00
408,  204,  103,  304,  2020-02-15,  5000.00
409,  203,  103,  304,  2020-01-10,  5000.00
410,  202,  103,  303,  2020-01-30,  2000.00
419,  204,  121,  303,  2020-03-29,  3.00
Please enter the Purchase ID of the policy you'd like to cancel:
420
No Policy found with purchase ID 420
```

PURCHASE POLICY FAILURE (INVALID ZIP)

```
Please enter your name:

KHANG

Please enter your city:

ROGERS

Please enter your 5-digit zip code:

999999
The zipcode you entered is not valid
```

CANCEL POLICY SUCCESS

```
---------------------------------
Query:
SELECT * FROM POLICIES_SOLD

Result:
PURCHASE_ID,  AGENT_ID,  CLIENT_ID,  POLICY_ID,  DATE_PURCHASED,  AMOUNT
401,  204,  106,  303,  2020-01-02,  2000.00
402,  201,  105,  305,  2019-08-11,  1500.00
403,  203,  106,  301,  2019-09-08,  3000.00
404,  207,  101,  305,  2019-06-21,  1500.00
405,  203,  104,  302,  2019-11-14,  4500.00
406,  207,  105,  305,  2019-12-25,  1500.00
407,  205,  103,  304,  2020-10-15,  5000.00
408,  204,  103,  304,  2020-02-15,  5000.00
409,  203,  103,  304,  2020-01-10,  5000.00
410,  202,  103,  303,  2020-01-30,  2000.00
419,  204,  121,  303,  2020-03-29,  3.00
Please enter the Purchase ID of the policy you'd like to cancel:
401
Policy with purchase ID 401 canceled.
```

PURCHASE POLICY FAILURE (INVALID POLICY TYPE)

```
Please enter your name:

JEFFPHEEREE

Please enter your city:

ROGERS

Please enter your 5-digit zip code:

72762

INSERT into CLIENTS (C_NAME, C_CITY, C_ZIP) values ('JEFFPHEEREE','ROGERS',72762)

A_ID,  A_NAME,  A_CITY,  A_ZIP
204,  BRYAN,  ROGERS,  78291
207,  SMITH,  ROGERS,  78291

Please enter the TYPE of policy you'd like to purchase:

CAR
No results found for that policy type.
```

CANCEL POLICY FAILURE (NO MATCHING PURCHASE ID, PREVIOUSLY DELETED PURCHASE ID)

```
---------------------------------
Query:
```

```
SELECT * FROM POLICIES_SOLD

Result:
PURCHASE_ID,  AGENT_ID,  CLIENT_ID,  POLICY_ID,  DATE_PURCHASED,  AMOUNT
402,  201,  105,  305,  2019-08-11,  1500.00
403,  203,  106,  301,  2019-09-08,  3000.00
404,  207,  101,  305,  2019-06-21,  1500.00
405,  203,  104,  302,  2019-11-14,  4500.00
406,  207,  105,  305,  2019-12-25,  1500.00
407,  205,  103,  304,  2020-10-15,  5000.00
408,  204,  103,  304,  2020-02-15,  5000.00
409,  203,  103,  304,  2020-01-10,  5000.00
410,  202,  103,  303,  2020-01-30,  2000.00
419,  204,  121,  303,  2020-03-29,  3.00
Please enter the Purchase ID of the policy you'd like to cancel:
401
No Policy found with purchase ID 401
```

LIST AGENTS, CLIENTS BY CITY SUCCESS (CLIENT OF CANCELLED POLICY NOT LISTED, NEW AGENTS)

```
Please enter the name of a city:

SCHENECTADY
A_ID,  A_NAME,  A_CITY,  A_ZIP
211,  PERRY,  SCHENECTADY,  12345
212,  BERRY,  SCHENECTADY,  12345
C_ID,  C_NAME,  C_CITY,  C_ZIP
```

LIST AGENTS, CLIENTS BY CITY SUCCESS, PREVIOUSLY EXISTING AND NEW

```
Please enter the name of a city:

ROGERS
A_ID,  A_NAME,  A_CITY,  A_ZIP
204,  BRYAN,  ROGERS,  78291
207,  SMITH,  ROGERS,  78291
C_ID,  C_NAME,  C_CITY,  C_ZIP
121,  JEFF,  ROGERS,  72762
122,  JEFFPHEEREE,  ROGERS,  72762
darusse@turing:~/hw4/ideProj/src$ exit
```