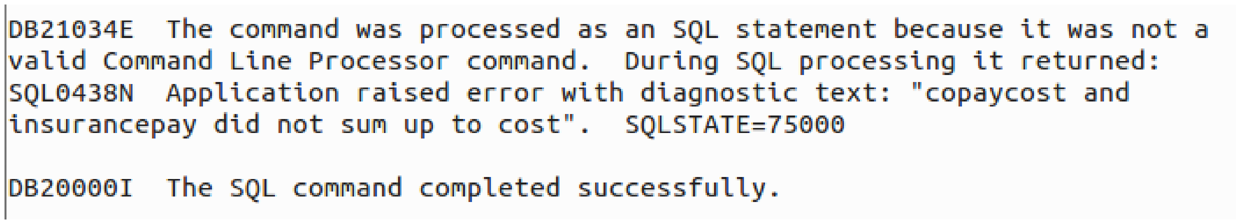
Question 1

```
DROP TRIGGER checkclaims

CREATE TRIGGER checkclaims \
before insert on claims \
referencing new as nclaim \
for each row \
when (nclaim.cost != nclaim.coPayCost + nclaim.insurancePay) \
signal sqlstate '75000' \
set message_text = 'copaycost and insurancepay did not sum up to cost'


INSERT INTO claims
(claimID,memberID,providerID,planID,procedureCode,cost,coPayCost,insurancePay,date) VALUES
(61,17,6,5,'M120',306,28,7,'2004-12-09')

INSERT INTO claims
(claimID,memberID,providerID,planID,procedureCode,cost,coPayCost,insurancePay,date) VALUES
(61,17,6,5,'M120',35,28,7,'2004-12-09')
```


```
DB21034E  The command was processed as an SQL statement because it was not a
valid Command Line Processor command.  During SQL processing it returned:
SQL0438N  Application raised error with diagnostic text: "copaycost and
insurancepay did not sum up to cost".  SQLSTATE=75000

DB20000I  The SQL command completed successfully.
```

Question 2 Procedure

```
CREATE PROCEDURE UpdateRating(IN pID int, IN avgCost int)
LANGUAGE SQL
BEGIN
DECLARE v_averageMemberCost int;
DECLARE v_providerID int;
DECLARE at_end INT DEFAULT 0;
DECLARE not_found CONDITION FOR SQLSTATE'02000';

DECLARE C1 CURSOR FOR
SELECT providerID, averageMemberCost FROM provider WHERE providerID = pID;
DECLARE CONTINUE HANDLER FOR not_found SET at_end = 1;
OPEN C1;
FETCH C1 INTO v_providerID, v_averageMemberCost;
WHILE at_end = 0 DO
        IF (v_averageMemberCost < avgCost)
                THEN UPDATE providerRating set rating = rating + 1 WHERE providerID = v_providerID;
        END IF;
        FETCH C1 INTO v_providerID, v_averageMemberCost;
END WHILE;
```

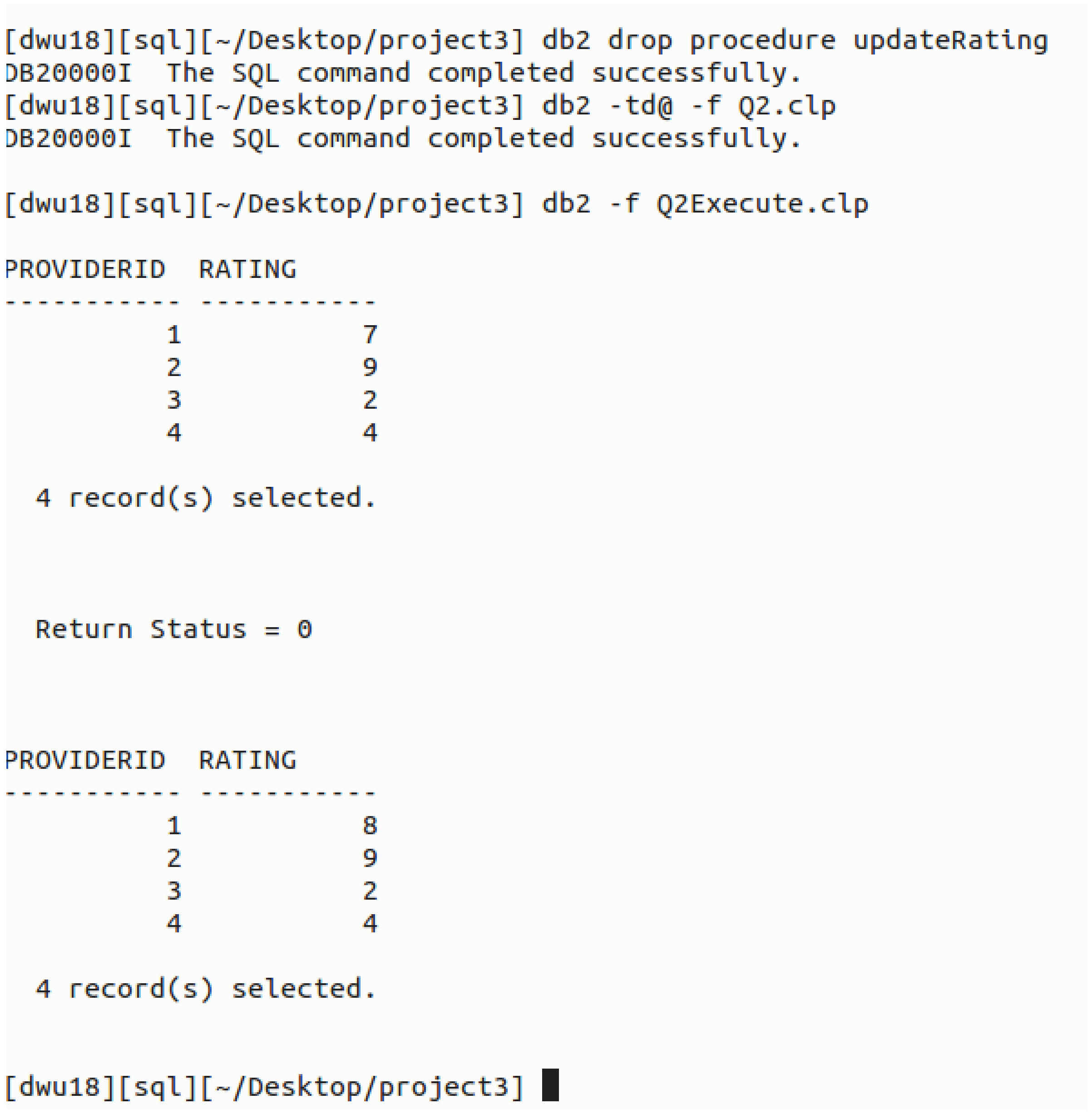
CLOSE C1;
END
@

Executing Q2

select * from providerRating

CALL UpdateRating(1,9999)

select * from providerRating



```
[dwu18][sql][~/Desktop/project3] db2 drop procedure updateRating
DB20000I  The SQL command completed successfully.
[dwu18][sql][~/Desktop/project3] db2 -td@ -f Q2.clp
DB20000I  The SQL command completed successfully.

[dwu18][sql][~/Desktop/project3] db2 -f Q2Execute.clp

PROVIDERID  RATING
----------- -----------
          1           7
          2           9
          3           2
          4           4

  4 record(s) selected.



  Return Status = 0



PROVIDERID  RATING
----------- -----------
          1           8
          2           9
          3           2
          4           4

  4 record(s) selected.


[dwu18][sql][~/Desktop/project3]
```

Question 3

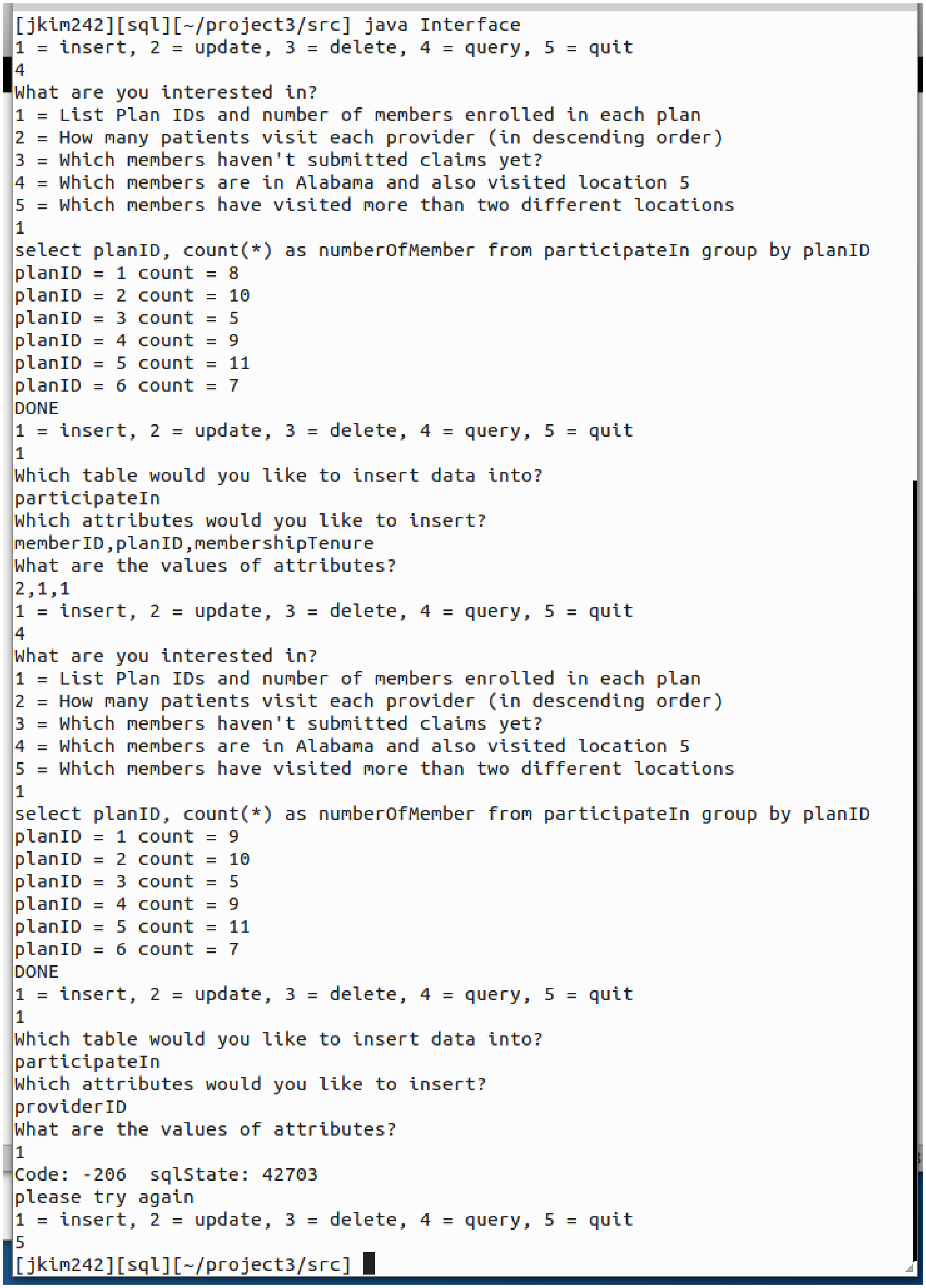
```
import java.util.Scanner;
import java.sql.*;

public class Interface {

	public static void main(String[] args) throws SQLException {
		// load DB2 JDBC driver
		try {
			DriverManager.registerDriver(new com.ibm.db2.jcc.DB2Driver());
		} catch (Exception cnfe) {
			System.out.println("Cannot find class!!!");
		}


		// connect to database

		Connection conn = DriverManager.getConnection("jdbc:db2://db2:50000/cs421", "dwu18",
"");
		Statement statement = conn.createStatement ();

		try {
			Runtime rt = Runtime.getRuntime();
			Process pr = rt.exec("db2 -f project3C.clp");
		} catch (Exception e) {
		}


		int sqlCode=0;      // Variable to hold SQLCODE
		String sqlState="00000";  // Variable to hold SQLSTATE
		boolean loop = true;
		Scanner input = new Scanner(System.in);
		input.useDelimiter("\n");
		while (loop) {
			System.out.println("1 = insert, 2 = update, 3 = delete, 4 = query, 5 = quit");

			int choice = input.nextInt();
			String tablename, attributes, values, attribute, newvalue, constraint;
			switch(choice) {
				case 1: //sql
					try {
					System.out.println("Which table would you like to insert data
into?");
					tablename = input.next();
					System.out.println("Which attributes would you like to insert?");
					attributes = input.next();
					System.out.println("What are the values of attributes?");
```

```
						values = input.next();
						String insert = "INSERT INTO " + tablename + " (" + attributes +
") values (" + values + ")";
						statement.executeUpdate(insert);
						} catch (SQLException e) {
							sqlCode = e.getErrorCode(); // Get SQLCODE
						sqlState = e.getSQLState(); // Get SQLSTATE
							System.out.println("Code: " + sqlCode + "  sqlState: " +
sqlState);

							System.out.println("please try again");
						}
						break;
					case 2: //sql update
						try {
						System.out.println("Which table would you like to update?");
						tablename = input.next();
						System.out.println("Which attribute would you like to update?");
						attribute = input.next();
						System.out.println("What is the new value?");
						newvalue = input.next();
						System.out.println("What is the constraint?");
						constraint = input.next();
						String update = "UPDATE " + tablename + " SET " + attribute + "
= " + newvalue + " WHERE " + constraint;
						System.out.println(update);

						statement.executeUpdate(update);
						} catch (SQLException e) {
							sqlCode = e.getErrorCode(); // Get SQLCODE
						sqlState = e.getSQLState(); // Get SQLSTATE
							System.out.println("Code: " + sqlCode + "  sqlState: " +
sqlState);

							System.out.println("please try again");
						}
						break;
					case 3: //sql delete
						try {
						System.out.println("Which table would you like to delete from?");
						tablename = input.next();
						System.out.println("What is the constraint?");
						constraint = input.next();
						String delete = "DELETE FROM " + tablename + " WHERE " +
constraint;
						System.out.println(delete);
						statement.executeUpdate(delete);
						} catch (SQLException e) {
							sqlCode = e.getErrorCode(); // Get SQLCODE
						sqlState = e.getSQLState(); // Get SQLSTATE
							System.out.println("Code: " + sqlCode + "  sqlState: " +
sqlState);
```

```
                                        System.out.println("please try again");
                                    }
                                    break;
                        case 4: //sql query
                                    try {
                                    String query;
                                    System.out.println("What are you interested in?");
                                    System.out.println("1 = List Plan IDs and number of members
enrolled in each plan");
                                    System.out.println("2 = How many patients visit each provider (in
descending order)");
                                    System.out.println("3 = Which members haven't submitted claims
yet?");
                                    System.out.println("4 = Which members are in Alabama and also
visited location 5");
                                    System.out.println("5 = Which members have visited more than
two different locations");
                                    int option = input.nextInt();
                                    switch(option) {
                                        case 1: query = "select planID, count(*) as
numberOfMember from participateIn group by planID";
                                                System.out.println(query);
                                                java.sql.ResultSet rs1 =
statement.executeQuery(query);
                                                while (rs1.next()) {
                                                    int planID = rs1.getInt(1);
                                                    int numberOfMember = rs1.getInt(2);
                                                    System.out.println("planID = " + planID
+ " count = " + numberOfMember);
                                                }
                                        System.out.println ("DONE");
                                        break;
                                        case 2: query = "select p.providerID, p.firstName,
p.lastName, p.specialty, p.experience, p.averageMemberCost, count(*) as count from provider p, services s
where p.providerID = s.providerID group by p.providerID, p.firstName, p.lastName, p.specialty, p.experience,
p.averageMemberCost order by count desc";
                                                System.out.println(query);
                                                java.sql.ResultSet rs2 =
statement.executeQuery(query);
                                                while (rs2.next()) {
                                                    int providerID = rs2.getInt(1);
                                                    int averageMemberCost = rs2.getInt(6);
                                                    int count = rs2.getInt(7);
                                                    System.out.println("providerID = " +
providerID + " averageMemberCost = " + averageMemberCost + " count = " + count);
                                                }
                                        System.out.println ("DONE");
                                        break;
                                        case 3: query = "select m.memberID, m.firstName,
m.LastName from members m where m.memberID not in ( select s.memberID from submits s)";
```

```
                                                System.out.println (query) ;
                                                java.sql.ResultSet rs3 = statement.executeQuery
(query) ;
                                                while (rs3.next()) {
                                                        int id = rs3.getInt (1) ;
                                                        String firstname = rs3.getString (2);
                                                        String lastname = rs3.getString (3);
                                                        System.out.println ("member id: " + id);
                                                        System.out.println ("first name: " +
firstname);
                                                        System.out.println ("last name: " +
lastname);
                                                }
                                                        System.out.println ("DONE");
                                                break;
                                        case 4: query = "select memberID from belongsTo b
where b.state = 'AL' intersect select memberID from visits v where v.locationID = 5";
                                                System.out.println (query) ;
                                                java.sql.ResultSet rs4 = statement.executeQuery
(query) ;
                                                while (rs4.next()) {
                                                        int id = rs4.getInt (1) ;
                                                        System.out.println ("member id: " + id);
                                                }
                                                System.out.println ("DONE");
                                                break;
                                        case 5: query = "select m.memberID, m.firstName,
m.lastName from members m where m.memberID in (select v.memberID from visits v group by v.memberId
having count(*) > 2)";
                                                System.out.println (query) ;
                                                java.sql.ResultSet rs5 = statement.executeQuery
(query) ;
                                                while (rs5.next()) {
                                                        int id = rs5.getInt (1) ;
                                                        String firstname = rs5.getString (2);

                                                        String lastname = rs5.getString (3);
                                                        System.out.println ("member id: " + id);
                                                        System.out.println ("first name: " +
firstname);
                                                        System.out.println ("last name: " +
lastname);
                                                }
                                                System.out.println ("DONE");
                                                break;
                                }
                                } catch (SQLException e) {
                                        sqlCode = e.getErrorCode(); // Get SQLCODE
                                sqlState = e.getSQLState(); // Get SQLSTATE
```

```
                                                            System.out.println("Code: " + sqlCode + "  sqlState: " + 
sqlState);
                                                            System.out.println("please try again");
                                                  }
                                                  break;
                                        case 5: loop = false;
                                                  break;
                                        default: System.out.println("Please choose between 1 and 5");
                                                  break;
                              }
                    }

                    input.close();
                    try {
                              Runtime rt = Runtime.getRuntime();
                              Process pr = rt.exec("db2 -f project3D.clp");
                    } catch (Exception e) {
                    }

          }

}
```

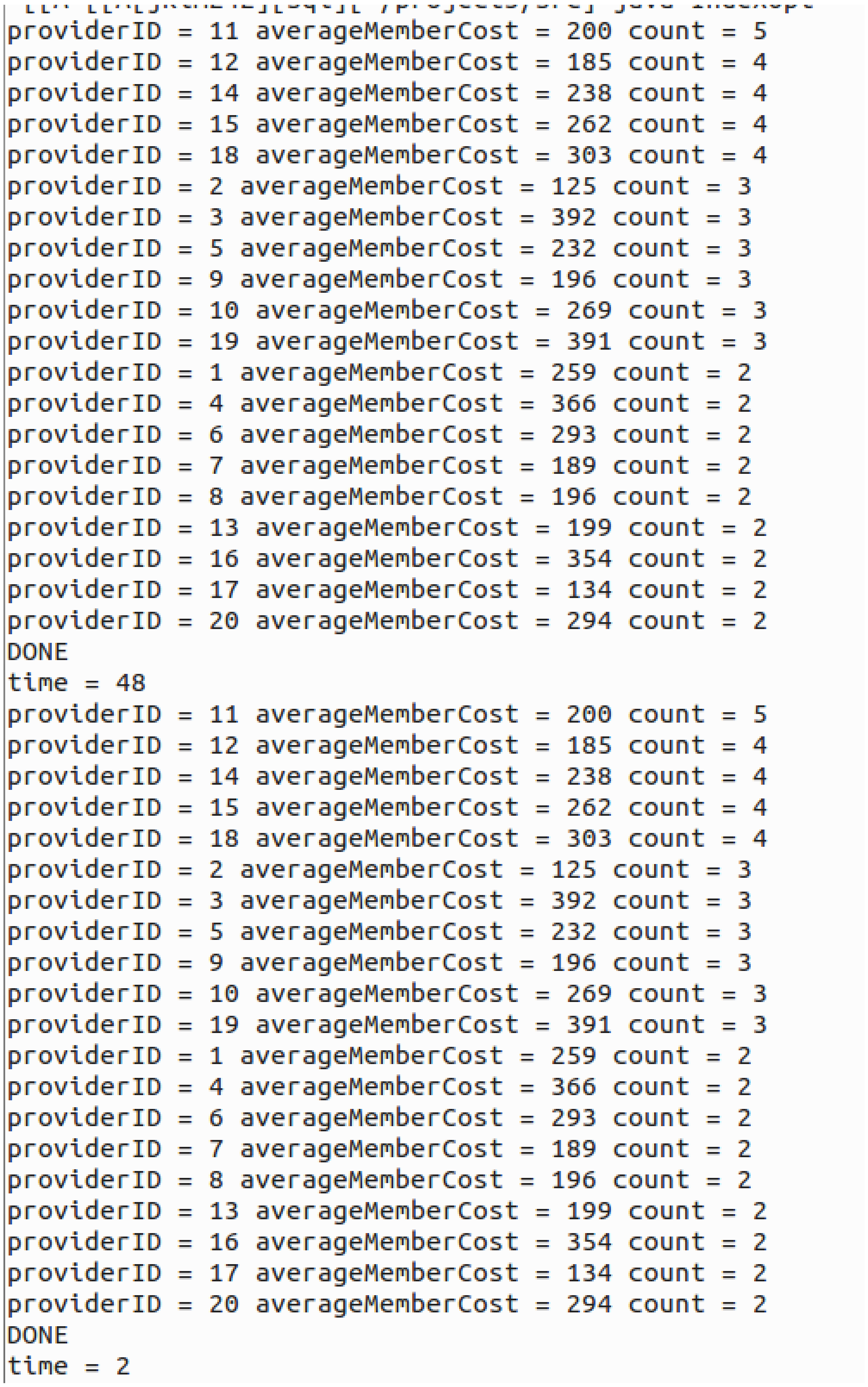


```
[jkim242][sql][~/project3/src] java Interface
1 = insert, 2 = update, 3 = delete, 4 = query, 5 = quit
4
What are you interested in?
1 = List Plan IDs and number of members enrolled in each plan
2 = How many patients visit each provider (in descending order)
3 = Which members haven't submitted claims yet?
4 = Which members are in Alabama and also visited location 5
5 = Which members have visited more than two different locations
1
select planID, count(*) as numberOfMember from participateIn group by planID
planID = 1 count = 8
planID = 2 count = 10
planID = 3 count = 5
planID = 4 count = 9
planID = 5 count = 11
planID = 6 count = 7
DONE
1 = insert, 2 = update, 3 = delete, 4 = query, 5 = quit
1
Which table would you like to insert data into?
participateIn
Which attributes would you like to insert?
memberID,planID,membershipTenure
What are the values of attributes?
2,1,1
1 = insert, 2 = update, 3 = delete, 4 = query, 5 = quit
4
What are you interested in?
1 = List Plan IDs and number of members enrolled in each plan
2 = How many patients visit each provider (in descending order)
3 = Which members haven't submitted claims yet?
4 = Which members are in Alabama and also visited location 5
5 = Which members have visited more than two different locations
1
select planID, count(*) as numberOfMember from participateIn group by planID
planID = 1 count = 9
planID = 2 count = 10
planID = 3 count = 5
planID = 4 count = 9
planID = 5 count = 11
planID = 6 count = 7
DONE
1 = insert, 2 = update, 3 = delete, 4 = query, 5 = quit
1
Which table would you like to insert data into?
participateIn
Which attributes would you like to insert?
providerID
What are the values of attributes?
1
Code: -206  sqlState: 42703
please try again
1 = insert, 2 = update, 3 = delete, 4 = query, 5 = quit
5
[jkim242][sql][~/project3/src]
```

Question 4

```
import java.sql.*;

public class Indexopt {

	public static void main(String[] args) throws SQLException {
		// load DB2 JDBC driver
		try {
			DriverManager.registerDriver(new com.ibm.db2.jcc.DB2Driver());
		} catch (Exception cnfe) {
			System.out.println("Cannot find class!!!");
		}


		// connect to database

		Connection conn = DriverManager.getConnection("jdbc:db2://db2:50000/cs421", "dwu18",
"");
		Statement statement = conn.createStatement ();

		try {
			Runtime rt = Runtime.getRuntime();
			Process pr = rt.exec("db2 -f project3C.clp");
		} catch (Exception e) {
		}


		int sqlCode=0;      // Variable to hold SQLCODE
		String sqlState="00000";  // Variable to hold SQLSTATE


		try {
			String query = "select p.providerID, p.firstName, p.lastName, p.specialty,
p.experience, p.averageMemberCost, count(*) as count from provider p, services s where p.providerID =
s.providerID group by p.providerID, p.firstName, p.lastName, p.specialty, p.experience,
p.averageMemberCost order by count desc";
			//without indexing
			long startTime = System.currentTimeMillis();

			java.sql.ResultSet rs1 = statement.executeQuery(query);
			while (rs1.next()) {
				int providerID = rs1.getInt(1);
				int averageMemberCost = rs1.getInt(6);
				int count = rs1.getInt(7);
				System.out.println("providerID = " + providerID + " averageMemberCost
= " + averageMemberCost + " count = " + count);
			}
```

```
					System.out.println ("DONE");
					long endTime = System.currentTimeMillis();
					long time = (endTime - startTime);
					System.out.println("time = " + time);



					String indexing = "create index ind1 on Provider(providerID)";

					statement.executeUpdate(indexing);


					//with indexing
					startTime = System.currentTimeMillis();

					java.sql.ResultSet rs2 = statement.executeQuery(query);
					while (rs2.next()) {
						int providerID = rs2.getInt(1);
						int averageMemberCost = rs2.getInt(6);
						int count = rs2.getInt(7);
						System.out.println("providerID = " + providerID + " averageMemberCost
= " + averageMemberCost + " count = " + count);
					}
					System.out.println ("DONE");

					endTime = System.currentTimeMillis();
					time = (endTime - startTime);
					System.out.println("time = " + time);



				} catch (SQLException e) {
								sqlCode = e.getErrorCode(); // Get SQLCODE
							sqlState = e.getSQLState(); // Get SQLSTATE
								System.out.println("Code: " + sqlCode + "  sqlState: " +
sqlState);
								System.out.println("please try again");
				}


				try {
					Runtime rt = Runtime.getRuntime();
					Process pr = rt.exec("db2 -f project3D.clp");
				} catch (Exception e) {
				}
		}
}
```



```
[illegible]
providerID = 11 averageMemberCost = 200 count = 5
providerID = 12 averageMemberCost = 185 count = 4
providerID = 14 averageMemberCost = 238 count = 4
providerID = 15 averageMemberCost = 262 count = 4
providerID = 18 averageMemberCost = 303 count = 4
providerID = 2 averageMemberCost = 125 count = 3
providerID = 3 averageMemberCost = 392 count = 3
providerID = 5 averageMemberCost = 232 count = 3
providerID = 9 averageMemberCost = 196 count = 3
providerID = 10 averageMemberCost = 269 count = 3
providerID = 19 averageMemberCost = 391 count = 3
providerID = 1 averageMemberCost = 259 count = 2
providerID = 4 averageMemberCost = 366 count = 2
providerID = 6 averageMemberCost = 293 count = 2
providerID = 7 averageMemberCost = 189 count = 2
providerID = 8 averageMemberCost = 196 count = 2
providerID = 13 averageMemberCost = 199 count = 2
providerID = 16 averageMemberCost = 354 count = 2
providerID = 17 averageMemberCost = 134 count = 2
providerID = 20 averageMemberCost = 294 count = 2
DONE
time = 48
providerID = 11 averageMemberCost = 200 count = 5
providerID = 12 averageMemberCost = 185 count = 4
providerID = 14 averageMemberCost = 238 count = 4
providerID = 15 averageMemberCost = 262 count = 4
providerID = 18 averageMemberCost = 303 count = 4
providerID = 2 averageMemberCost = 125 count = 3
providerID = 3 averageMemberCost = 392 count = 3
providerID = 5 averageMemberCost = 232 count = 3
providerID = 9 averageMemberCost = 196 count = 3
providerID = 10 averageMemberCost = 269 count = 3
providerID = 19 averageMemberCost = 391 count = 3
providerID = 1 averageMemberCost = 259 count = 2
providerID = 4 averageMemberCost = 366 count = 2
providerID = 6 averageMemberCost = 293 count = 2
providerID = 7 averageMemberCost = 189 count = 2
providerID = 8 averageMemberCost = 196 count = 2
providerID = 13 averageMemberCost = 199 count = 2
providerID = 16 averageMemberCost = 354 count = 2
providerID = 17 averageMemberCost = 134 count = 2
providerID = 20 averageMemberCost = 294 count = 2
DONE
time = 2
```