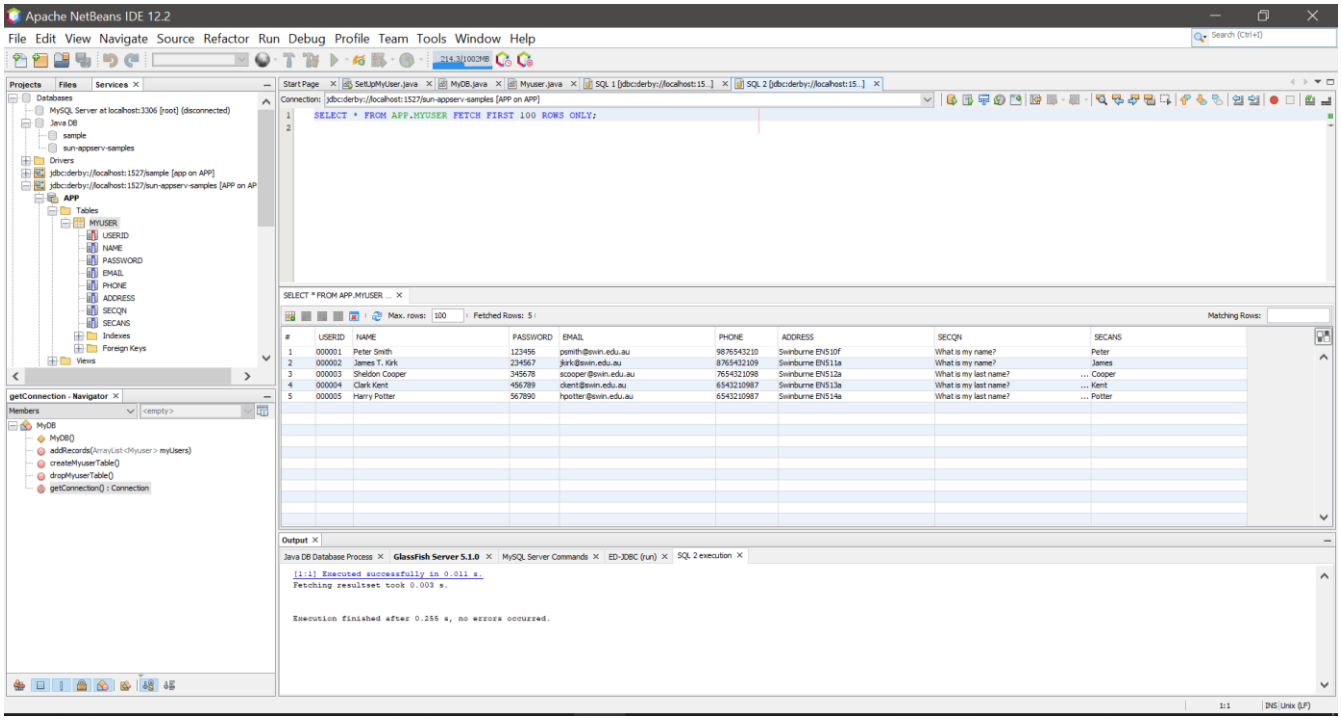
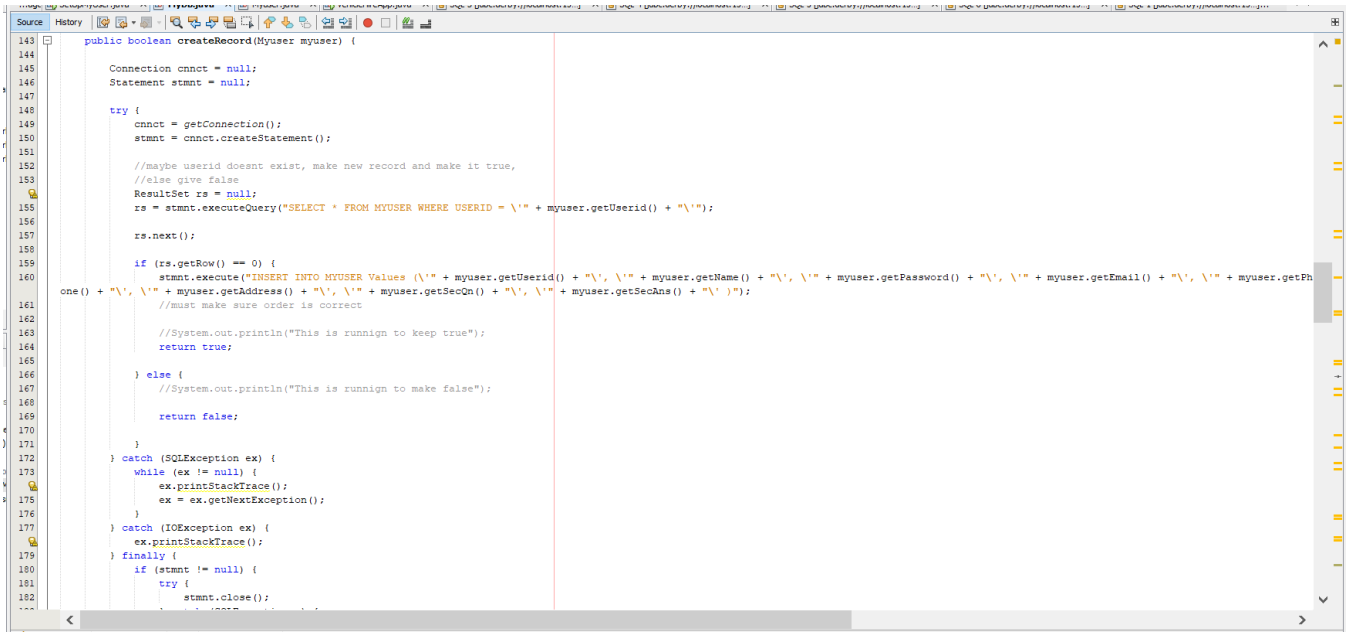
Name: S M Ragib Rezwan
ID: 103172423
Tutor name: Wei Lai Task 2.1P correction, COS30041

After step 1 was done:

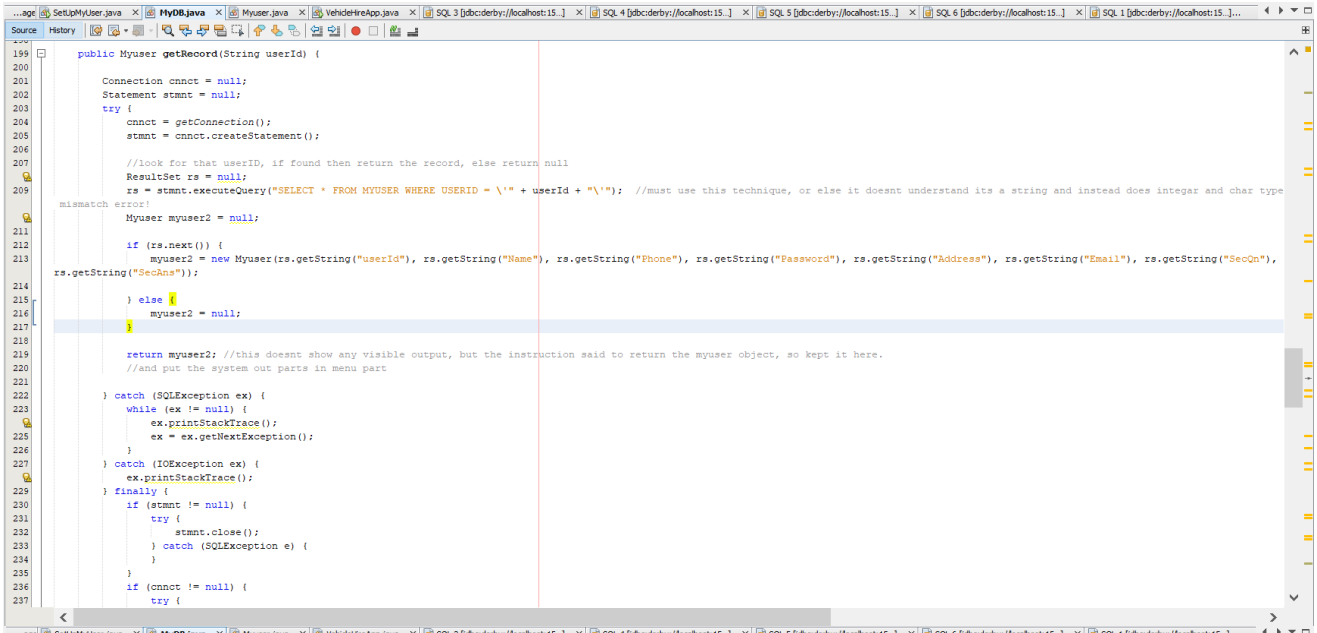


The picture for codes for step2:

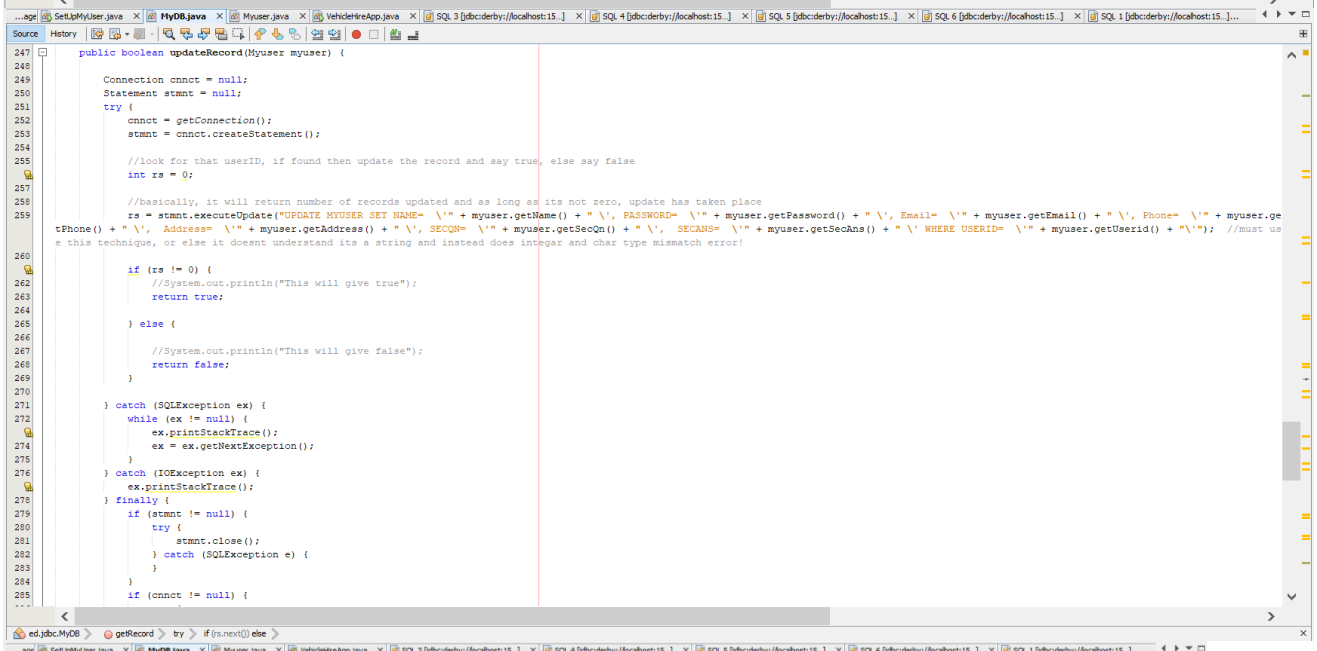


```
    public boolean createRecord(Myuser myuser) {

        Connection cnnct = null;
        Statement stmnt = null;

        try {
            cnnct = getConnection();
            stmnt = cnnct.createStatement();

            //maybe userid doesnt exist, make new record and make it true,
            //else give false
            ResultSet rs = null;
            rs = stmnt.executeQuery("SELECT * FROM MYUSER WHERE USERID = \'" + myuser.getUserid() + "\'");

            rs.next();

            if (rs.getRow() == 0) {
                stmnt.execute("INSERT INTO MYUSER Values (\'" + myuser.getUserid() + "\', \'" + myuser.getName() + "\', \'" + myuser.getPassword() + "\', \'" + myuser.getEmail() + "\', \'" + myuser.getPh
one() + "\', \'" + myuser.getAddress() + "\', \'" + myuser.getSecQn() + "\', \'" + myuser.getSecAns() + "\' )");
                //must make sure order is correct

                //System.out.println("This is runnign to keep true");
                return true;

            } else {
                //System.out.println("This is runnign to make false");

                return false;

            }
        } catch (SQLException ex) {
            while (ex != null) {
                ex.printStackTrace();
                ex = ex.getNextException();
            }
        } catch (IOException ex) {
            ex.printStackTrace();
        } finally {
            if (stmnt != null) {
                try {
                    stmnt.close();
```

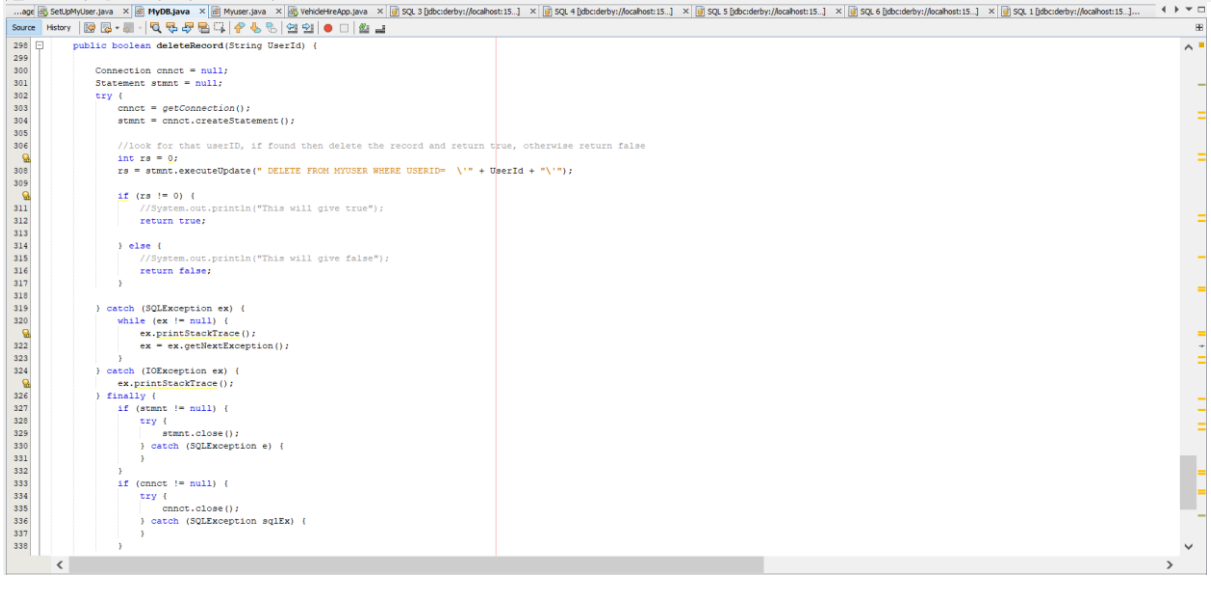
Name: S M Ragib Rezwan
ID: 103172423
Tutor name: Wei Lai Task 2.1P correction, COS30041



```
    public Myuser getRecord(String userId) {

        Connection cnnct = null;
        Statement stmnt = null;
        try {
            cnnct = getConnection();
            stmnt = cnnct.createStatement();

            //look for that userID, if found then return the record, else return null
            ResultSet rs = null;
            rs = stmnt.executeQuery("SELECT * FROM MYUSER WHERE USERID = \'" + userId + "\'");  //must use this technique, or else it doesnt understand its a string and instead does integar and char type
 mismatch error!
            Myuser myuser2 = null;

            if (rs.next()) {
                myuser2 = new Myuser(rs.getString("userId"), rs.getString("Name"), rs.getString("Phone"), rs.getString("Password"), rs.getString("Address"), rs.getString("Email"), rs.getString("SecQn"),
rs.getString("SecAns"));

            } else {
                myuser2 = null;
            }

            return myuser2; //this doesnt show any visible output, but the instruction said to return the myuser object, so kept it here.
            //and put the system out parts in menu part

        } catch (SQLException ex) {
            while (ex != null) {
                ex.printStackTrace();
                ex = ex.getNextException();
            }
        } catch (IOException ex) {
            ex.printStackTrace();
        } finally {
            if (stmnt != null) {
                try {
                    stmnt.close();
                } catch (SQLException e) {
                }
            }
            if (cnnct != null) {
                try {
```



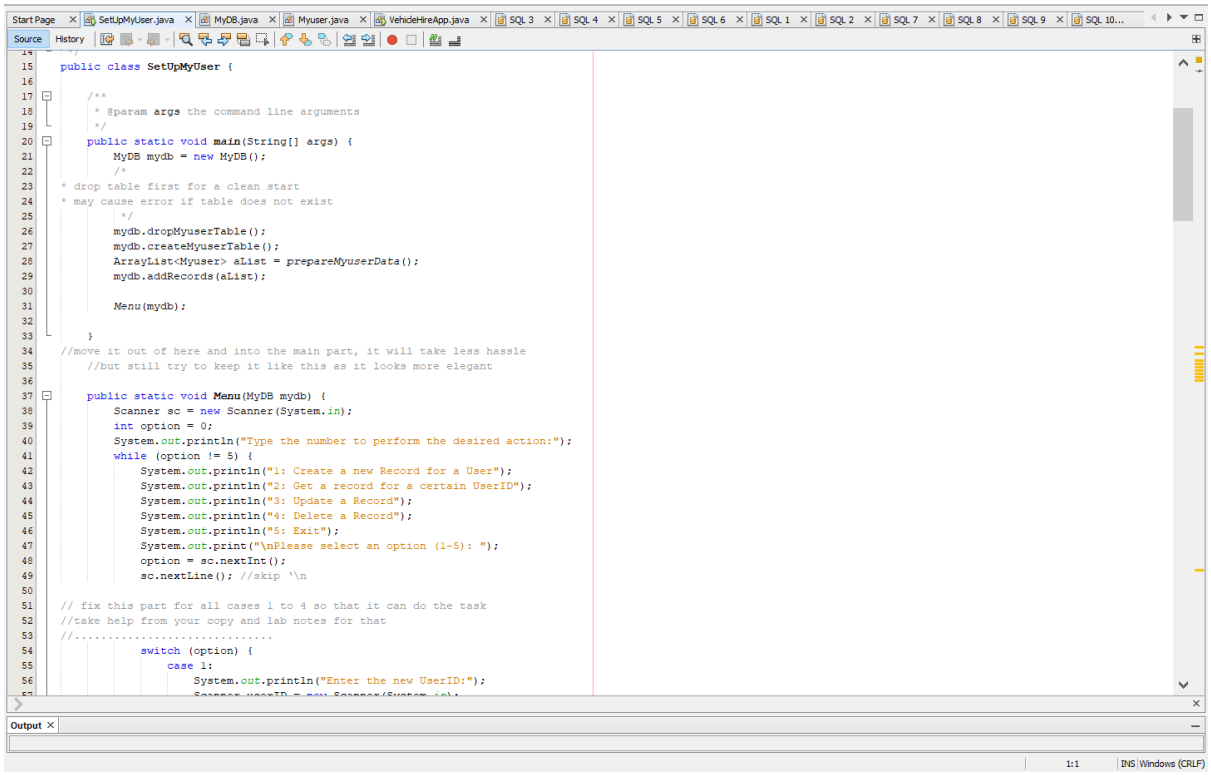
```
    public boolean updateRecord(Myuser myuser) {

        Connection cnnct = null;
        Statement stmnt = null;
        try {
            cnnct = getConnection();
            stmnt = cnnct.createStatement();

            //look for that userID, if found then update the record and say true, else say false
            int rs = 0;

            //basically, it will return number of records updated and as long as its not zero, update has taken place
            rs = stmnt.executeUpdate("UPDATE MYUSER SET NAME=  \'" + myuser.getName() + " \', PASSWORD=  \'" + myuser.getPassword() + " \', Email=  \'" + myuser.getEmail() + " \', Phone=  \'" + myuser.ge
tPhone() + " \',  Address=  \'" + myuser.getAddress() + " \', SECQN=  \'" + myuser.getSecQn() + " \',  SECANS=  \'" + myuser.getSecAns() + " \' WHERE USERID=  \'" + myuser.getUserid() + "\'");  //must us
e this technique, or else it doesnt understand its a string and instead does integar and char type mismatch error!

            if (rs != 0) {
                //System.out.println("This will give true");
                return true;

            } else {

                //System.out.println("This will give false");
                return false;
            }

        } catch (SQLException ex) {
            while (ex != null) {
                ex.printStackTrace();
                ex = ex.getNextException();
            }
        } catch (IOException ex) {
            ex.printStackTrace();
        } finally {
            if (stmnt != null) {
                try {
                    stmnt.close();
                } catch (SQLException e) {
                }
            }
            if (cnnct != null) {
```



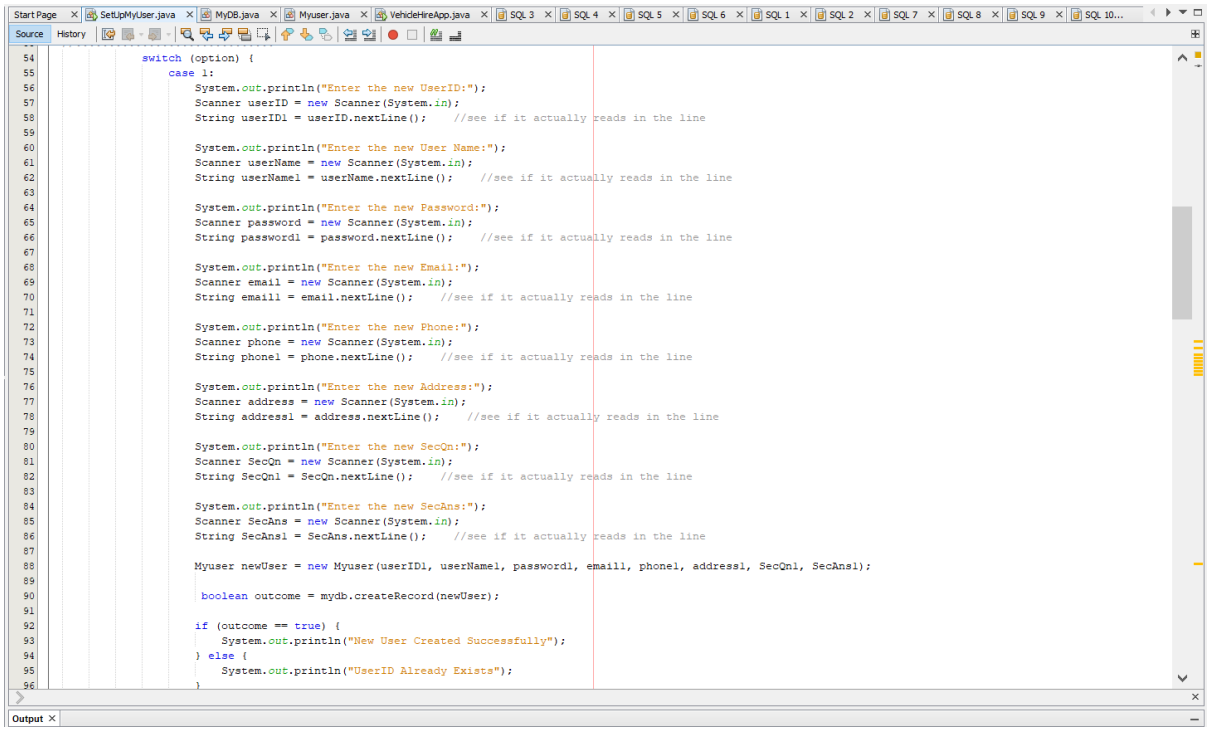
```
    public boolean deleteRecord(String UserId) {

        Connection cnnct = null;
        Statement stmnt = null;
        try {
            cnnct = getConnection();
            stmnt = cnnct.createStatement();

            //look for that userID, if found then delete the record and return true, otherwise return false
            int rs = 0;
            rs = stmnt.executeUpdate(" DELETE FROM MYUSER WHERE USERID=  \'" + UserId + "\'");

            if (rs != 0) {
                //System.out.println("This will give true");
                return true;

            } else {
                //System.out.println("This will give false");
                return false;
            }

        } catch (SQLException ex) {
            while (ex != null) {
                ex.printStackTrace();
                ex = ex.getNextException();
            }
        } catch (IOException ex) {
            ex.printStackTrace();
        } finally {
            if (stmnt != null) {
                try {
                    stmnt.close();
                } catch (SQLException e) {
                }
            }
            if (cnnct != null) {
                try {
                    cnnct.close();
                } catch (SQLException sqlEx) {
                }
            }
```

Name: S M Ragib Rezwan
ID: 103172423
Tutor name: Wei Lai Task 2.1P correction, COS30041
Step 3 pictures:

Code pictures: (part of the correction, 5 pictures to fit in all codes)



```
public class SetUpMyUser {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        MyDB mydb = new MyDB();
        /*
* drop table first for a clean start
* may cause error if table does not exist
        */
        mydb.dropMyuserTable();
        mydb.createMyuserTable();
        ArrayList<Myuser> aList = prepareMyuserData();
        mydb.addRecords(aList);

        Menu(mydb);

    }
//move it out of here and into the main part, it will take less hassle
    //but still try to keep it like this as it looks more elegant

    public static void Menu(MyDB mydb) {
        Scanner sc = new Scanner(System.in);
        int option = 0;
        System.out.println("Type the number to perform the desired action:");
        while (option != 5) {
            System.out.println("1: Create a new Record for a User");
            System.out.println("2: Get a record for a certain UserID");
            System.out.println("3: Update a Record");
            System.out.println("4: Delete a Record");
            System.out.println("5: Exit");
            System.out.print("\nPlease select an option (1-5): ");
            option = sc.nextInt();
            sc.nextLine(); //skip '\n

// fix this part for all cases 1 to 4 so that it can do the task
//take help from your copy and lab notes for that
//.............................
            switch (option) {
                case 1:
                    System.out.println("Enter the new UserID:");
```



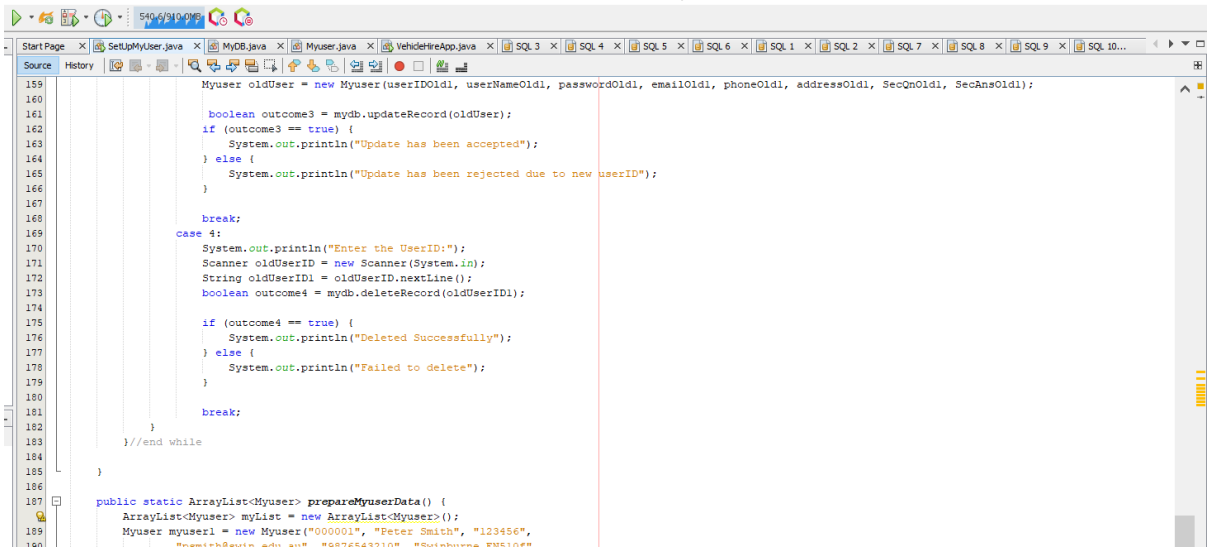
```
            switch (option) {
                case 1:
                    System.out.println("Enter the new UserID:");
                    Scanner userID = new Scanner(System.in);
                    String userID1 = userID.nextLine();     //see if it actually reads in the line

                    System.out.println("Enter the new User Name:");
                    Scanner userName = new Scanner(System.in);
                    String userName1 = userName.nextLine();     //see if it actually reads in the line

                    System.out.println("Enter the new Password:");
                    Scanner password = new Scanner(System.in);
                    String password1 = password.nextLine();     //see if it actually reads in the line

                    System.out.println("Enter the new Email:");
                    Scanner email = new Scanner(System.in);
                    String email1 = email.nextLine();     //see if it actually reads in the line

                    System.out.println("Enter the new Phone:");
                    Scanner phone = new Scanner(System.in);
                    String phone1 = phone.nextLine();     //see if it actually reads in the line

                    System.out.println("Enter the new Address:");
                    Scanner address = new Scanner(System.in);
                    String address1 = address.nextLine();     //see if it actually reads in the line

                    System.out.println("Enter the new SecQn:");
                    Scanner SecQn = new Scanner(System.in);
                    String SecQn1 = SecQn.nextLine();     //see if it actually reads in the line

                    System.out.println("Enter the new SecAns:");
                    Scanner SecAns = new Scanner(System.in);
                    String SecAns1 = SecAns.nextLine();     //see if it actually reads in the line

                    Myuser newUser = new Myuser(userID1, userName1, password1, email1, phone1, address1, SecQn1, SecAns1);

                     boolean outcome = mydb.createRecord(newUser);

                    if (outcome == true) {
                        System.out.println("New User Created Successfully");
                    } else {
                        System.out.println("UserID Already Exists");
                    }
```

Name: S M Ragib Rezwan
ID: 103172423
Tutor name: Wei Lai Task 2.1P correction, COS30041

```
                    } else {
                        System.out.println("UserID Already Exists");
                    }

                    break;
                case 2:
                    System.out.println("Enter the UserID:");
                    Scanner newUserID = new Scanner(System.in);
                    String newUserID1 = newUserID.nextLine();
                    //mydb.getRecord(newUserID1.toString()); //may cause it to run twice

                    if (mydb.getRecord(newUserID1.toString()) != null) {
                        //try to see if you can call it to print the lines of userid and stuff from here by calling that record itself hereas it returns a myuser type
object
                        Myuser myuser2 = mydb.getRecord(newUserID1.toString());
                        //not sure if this is ok to do like this, ask it to him during lab

                        System.out.println(myuser2.getUserid().toString());
                        System.out.println(myuser2.getName().toString());
                        System.out.println(myuser2.getPhone().toString());
                        System.out.println(myuser2.getPassword().toString());
                        System.out.println(myuser2.getAddress().toString());
                        System.out.println(myuser2.getEmail().toString());
                        System.out.println(myuser2.getSecQn().toString());
                        System.out.println(myuser2.getSecAns().toString());

                    }
                    else{
                        System.out.println("UserID is not found");

                    }

                    break;
                case 3:
                    System.out.println("Enter the UserID:");
                    Scanner userIDOld = new Scanner(System.in);
                    String userIDOld1 = userIDOld.nextLine();    //see if it actually reads in the line

                    System.out.println("Enter the User Name:");
                    Scanner userNameOld = new Scanner(System.in);
                    String userNameOld1 = userNameOld.nextLine();    //see if it actually reads in the line
```

```
                    String userNameOld1 = userNameOld.nextLine();    //see if it actually reads in the line

                    System.out.println("Enter the Password:");
                    Scanner passwordOld = new Scanner(System.in);
                    String passwordOld1 = passwordOld.nextLine();    //see if it actually reads in the line

                    System.out.println("Enter the Email:");
                    Scanner emailOld = new Scanner(System.in);
                    String emailOld1 = emailOld.nextLine();    //see if it actually reads in the line

                    System.out.println("Enter the Phone:");
                    Scanner phoneOld = new Scanner(System.in);
                    String phoneOld1 = phoneOld.nextLine();    //see if it actually reads in the line

                    System.out.println("Enter the Address:");
                    Scanner addressOld = new Scanner(System.in);
                    String addressOld1 = addressOld.nextLine();    //see if it actually reads in the line

                    System.out.println("Enter the SecQn:");
                    Scanner SecQnOld = new Scanner(System.in);
                    String SecQnOld1 = SecQnOld.nextLine();    //see if it actually reads in the line

                    System.out.println("Enter the SecAns:");
                    Scanner SecAnsOld = new Scanner(System.in);
                    String SecAnsOld1 = SecAnsOld.nextLine();    //see if it actually reads in the line

                    Myuser oldUser = new Myuser(userIDOld1, userNameOld1, passwordOld1, emailOld1, phoneOld1, addressOld1, SecQnOld1, SecAnsOld1);

                     boolean outcome3 = mydb.updateRecord(oldUser);
                    if (outcome3 == true) {
                        System.out.println("Update has been accepted");
                    } else {
                        System.out.println("Update has been rejected due to new userID");
                    }

                    break;
                case 4:
                    System.out.println("Enter the UserID:");
                    Scanner oldUserID = new Scanner(System.in);
                    String oldUserID1 = oldUserID.nextLine();
                    boolean outcome4 = mydb.deleteRecord(oldUserID1);

                    if (outcome4 == true) {
```

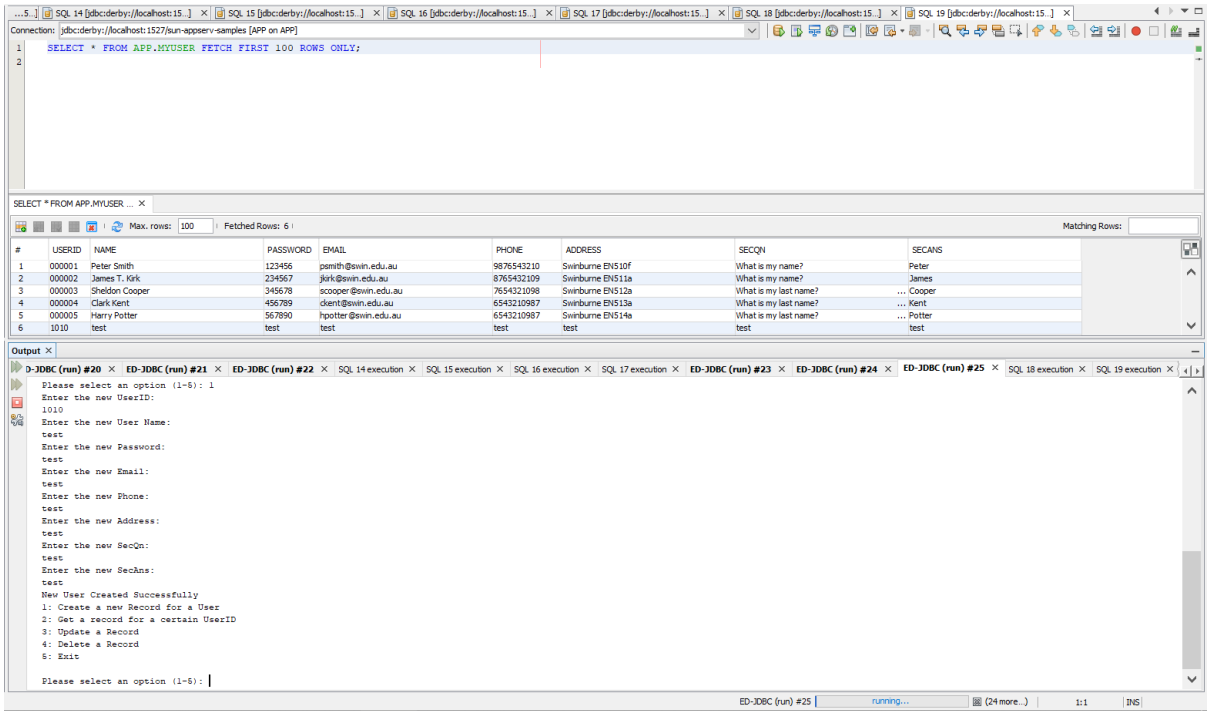
Name: S M Ragib Rezwan
ID: 103172423
Tutor name: Wei Lai Task 2.1P correction, COS30041



```
                    Myuser oldUser = new Myuser(userIDOld1, userNameOld1, passwordOld1, emailOld1, phoneOld1, addressOld1, SecQnOld1, SecAnsOld1);

                     boolean outcome3 = mydb.updateRecord(oldUser);
                    if (outcome3 == true) {
                        System.out.println("Update has been accepted");
                    } else {
                        System.out.println("Update has been rejected due to new userID");
                    }

                    break;
                case 4:
                    System.out.println("Enter the UserID:");
                    Scanner oldUserID = new Scanner(System.in);
                    String oldUserID1 = oldUserID.nextLine();
                    boolean outcome4 = mydb.deleteRecord(oldUserID1);

                    if (outcome4 == true) {
                        System.out.println("Deleted Successfully");
                    } else {
                        System.out.println("Failed to delete");
                    }

                    break;
            }
        }//end while

    }

    public static ArrayList<Myuser> prepareMyuserData() {
        ArrayList<Myuser> myList = new ArrayList<Myuser>();
        Myuser myuser1 = new Myuser("000001", "Peter Smith", "123456",
```

Outcome for each test:

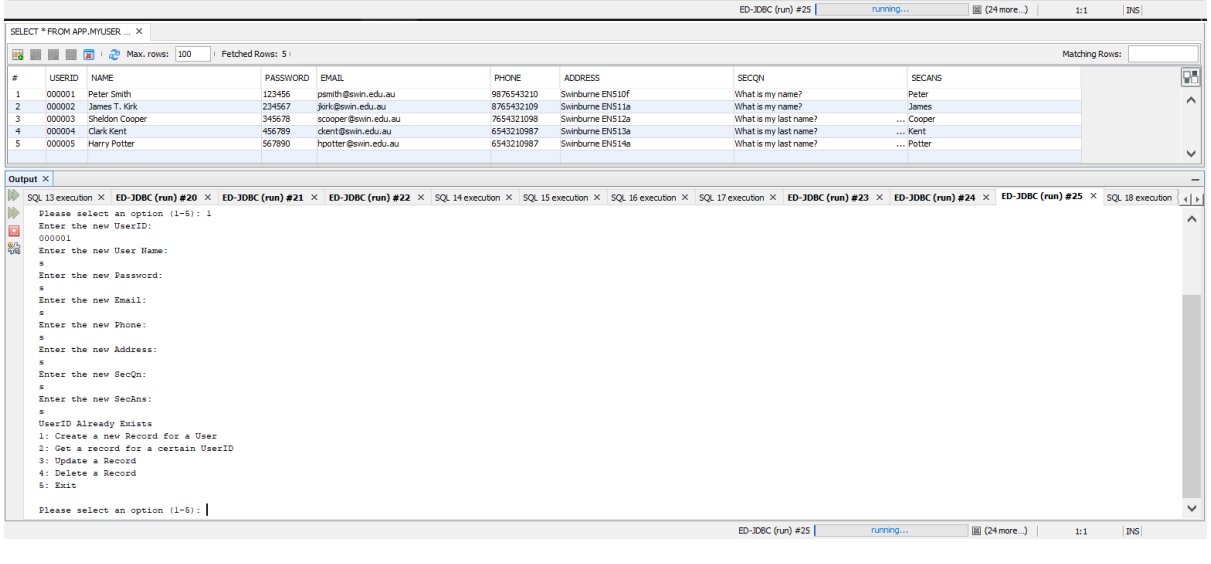
New record entry both cases:



```
SELECT * FROM APP.MYUSER FETCH FIRST 100 ROWS ONLY;
```

#	USERID	NAME	PASSWORD	EMAIL	PHONE	ADDRESS	SECQN	SECANS
1	000001	Peter Smith	123456	psmith@swin.edu.au	9876543210	Swinburne EN510f	What is my name?	Peter
2	000002	James T. Kirk	234567	jkirk@swin.edu.au	8765432109	Swinburne EN511a	What is my name?	James
3	000003	Sheldon Cooper	345678	scooper@swin.edu.au	7654321098	Swinburne EN512a	What is my last name?	... Cooper
4	000004	Clark Kent	456789	ckent@swin.edu.au	6543210987	Swinburne EN513a	What is my last name?	... Kent
5	000005	Harry Potter	567890	hpotter@swin.edu.au	6543210987	Swinburne EN514a	What is my last name?	... Potter
6	1010	test	test	test	test	test	test	test

```
Please select an option (1-5): 1
Enter the new UserID:
1010
Enter the new User Name:
test
Enter the new Password:
test
Enter the new Email:
test
Enter the new Phone:
test
Enter the new Address:
test
Enter the new SecQn:
test
Enter the new SecAns:
test
New User Created Successfully
1: Create a new Record for a User
2: Get a record for a certain UserID
3: Update a Record
4: Delete a Record
5: Exit

Please select an option (1-5):
```



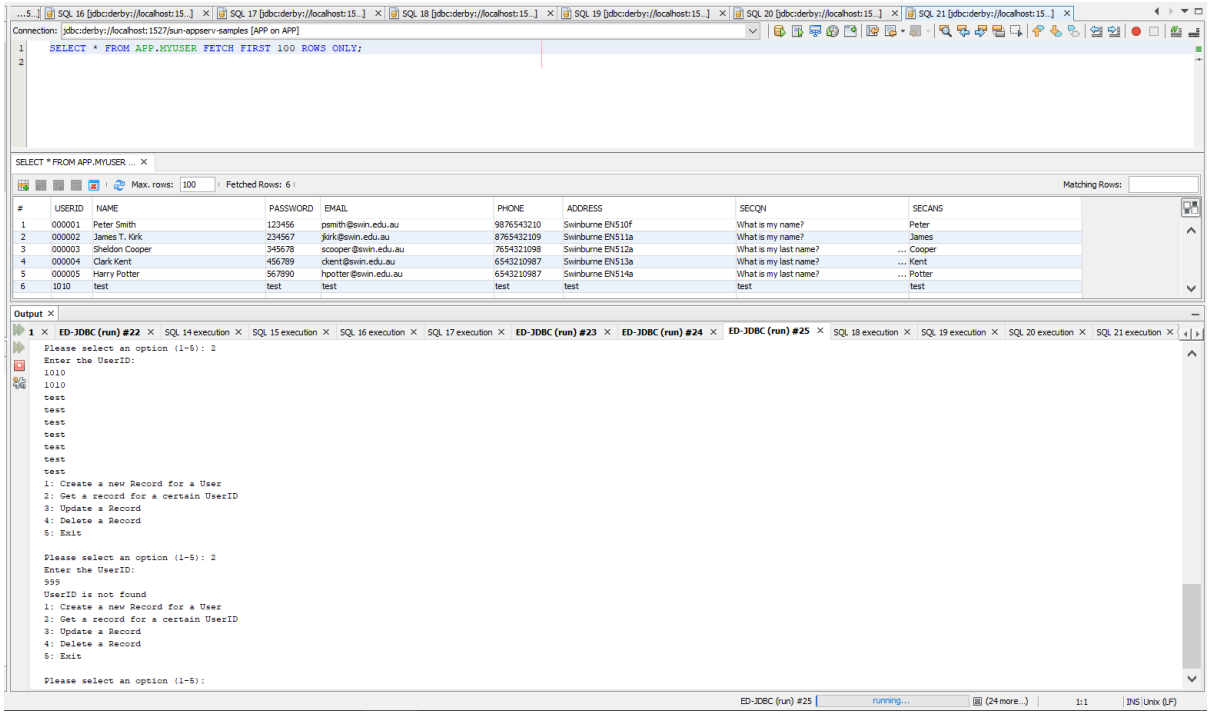
```
SELECT * FROM APP.MYUSER FETCH FIRST 100 ROWS ONLY;
```

#	USERID	NAME	PASSWORD	EMAIL	PHONE	ADDRESS	SECQN	SECANS
1	000001	Peter Smith	123456	psmith@swin.edu.au	9876543210	Swinburne EN510f	What is my name?	Peter
2	000002	James T. Kirk	234567	jkirk@swin.edu.au	8765432109	Swinburne EN511a	What is my name?	James
3	000003	Sheldon Cooper	345678	scooper@swin.edu.au	7654321098	Swinburne EN512a	What is my last name?	... Cooper
4	000004	Clark Kent	456789	ckent@swin.edu.au	6543210987	Swinburne EN513a	What is my last name?	... Kent
5	000005	Harry Potter	567890	hpotter@swin.edu.au	6543210987	Swinburne EN514a	What is my last name?	... Potter

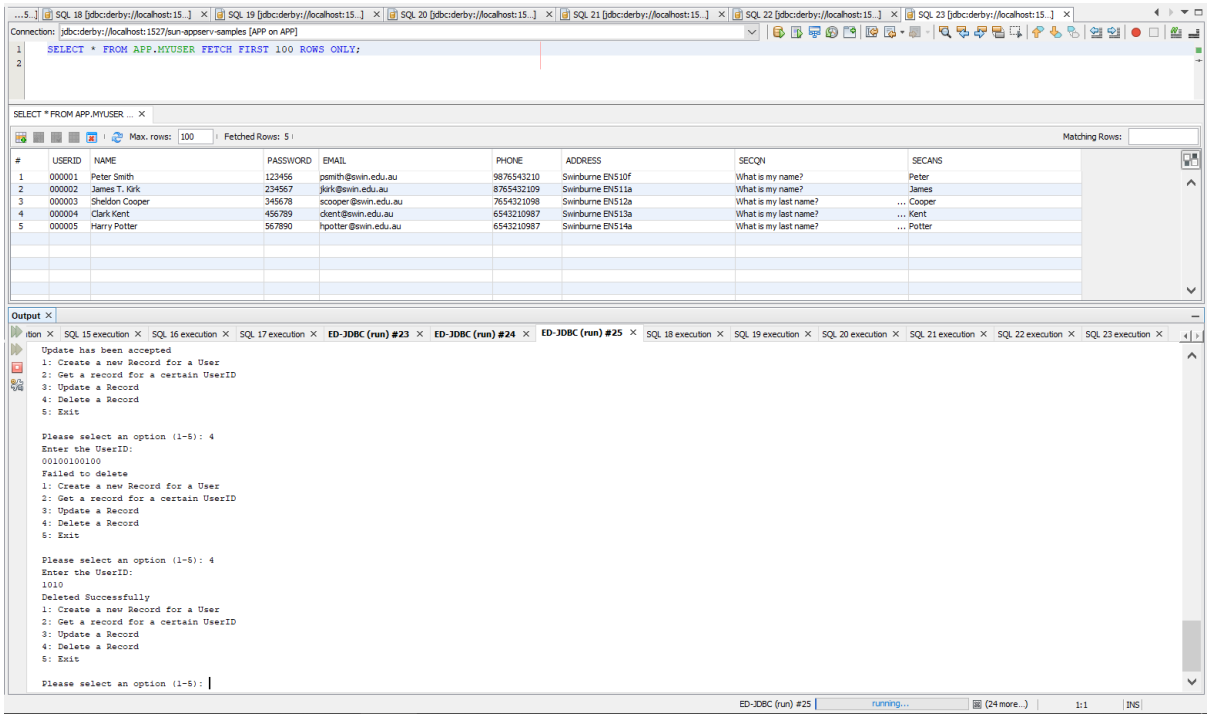
```
Please select an option (1-5): 1
Enter the new UserID:
000001
Enter the new User Name:
s
Enter the new Password:
s
Enter the new Email:
s
Enter the new Phone:
s
Enter the new Address:
s
Enter the new SecQn:
s
Enter the new SecAns:
s
UserID Already Exists
1: Create a new Record for a User
2: Get a record for a certain UserID
3: Update a Record
4: Delete a Record
5: Exit

Please select an option (1-5):
```

Name: S M Ragib Rezwan
ID: 103172423
Tutor name: Wei Lai Task 2.1P correction, COS30041

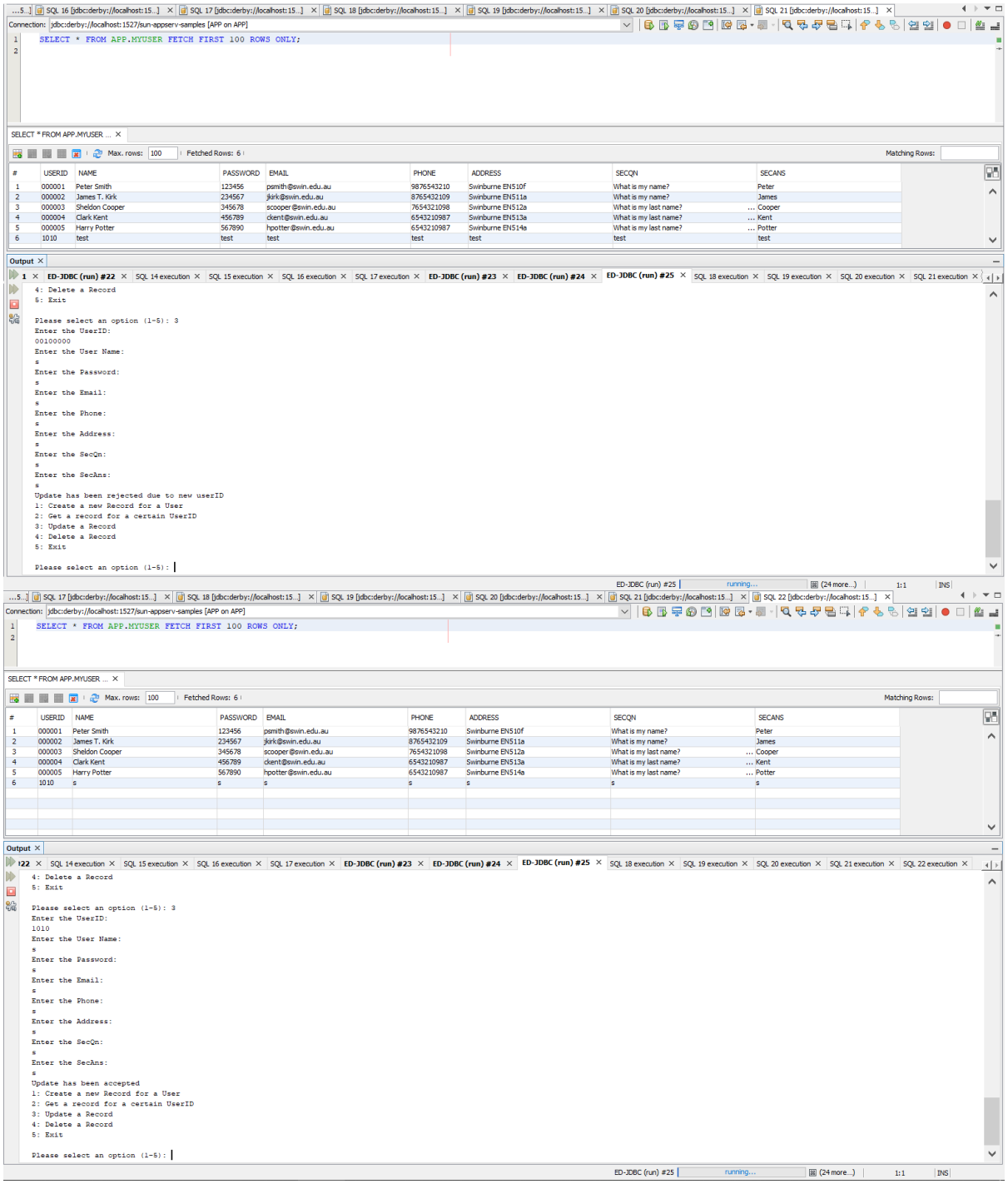
Get record both outcomes:



Delete record both outcomes:



Name: S M Ragib Rezwan
ID: 103172423
Tutor name: Wei Lai Task 2.1P correction, COS30041
Update record both outcomes:



Step -4:

4.1: DTO means data transfer object and DAO means data access object. Myuser class is a DTO as it holds the pattern to create individual user, instead of making them separately by calling userid, name,etc.

4.1.1: DAO basically does some certain, specific data operations without exposing details of the database itself. And since myuser is not a DAO, MyDB must be the DAO.

Name: S M Ragib Rezwan
ID: 103172423
Tutor name: Wei Lai Task 2.1P correction, COS30041

4.2: No. That's because myuser.setname is a method of the myuser class which is not a dao. Thus it can't access the database when called in the program.

4.3: No. That's because even though myDB is a DAO, it calling method of myuser class which is not a dao. Thus it can't change value in the database.