**­­­­­Project 1.B**

1. Project Description

Figure 1 shows the ER diagram for University database we implemented in project 1.A.

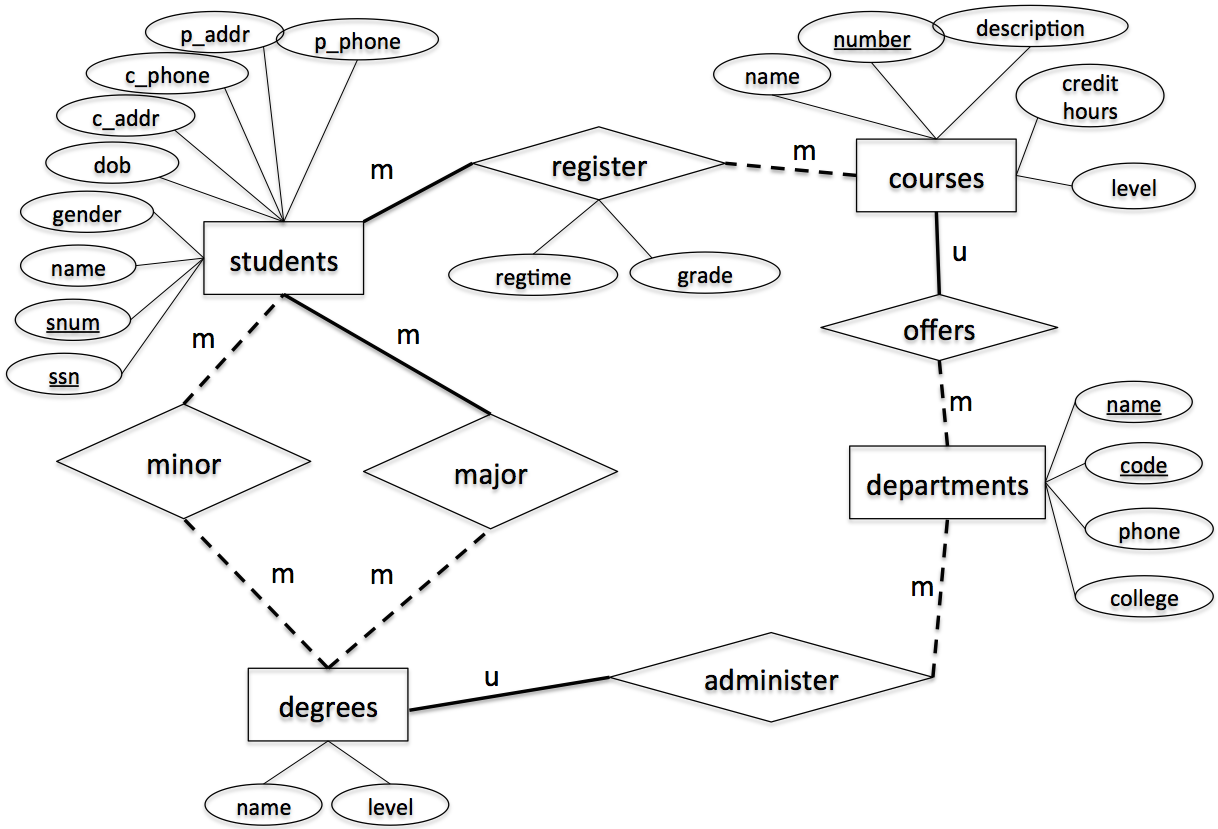


Figure 1. ER-diagram

This project is to implement the above design using a relational data model. Specifically, you are asked to write the following Java programs using JDBC connection to connect MySQL database to Java programs.

1. **CreateTables.java [Points: 15]**

After execution, your program must create tables in MySQL, which are the same with project 1A. You can find your MySQL code from Gradescope.

1. **InsertRecords.java [Points: 15]**

After execution, your program must insert the records to the appropriate tables created by CreateTables.java. The records are the same with project 1A.

1. **Query.java [Points: 40]**

After execution, your program must print out the following information (note that these are new queries different from project 1A)

* 1. **The numbers and names of courses and their corresponding average grades from students registered in the past semesters.**
  2. **The count of female students who major or minor in a degree managed by LAS departments**
  3. **The names of degrees that have more male students than female students (major or minor)**
  4. **The major degree that the youngest student is taken**

1. **ModifyRecords.java [25]**

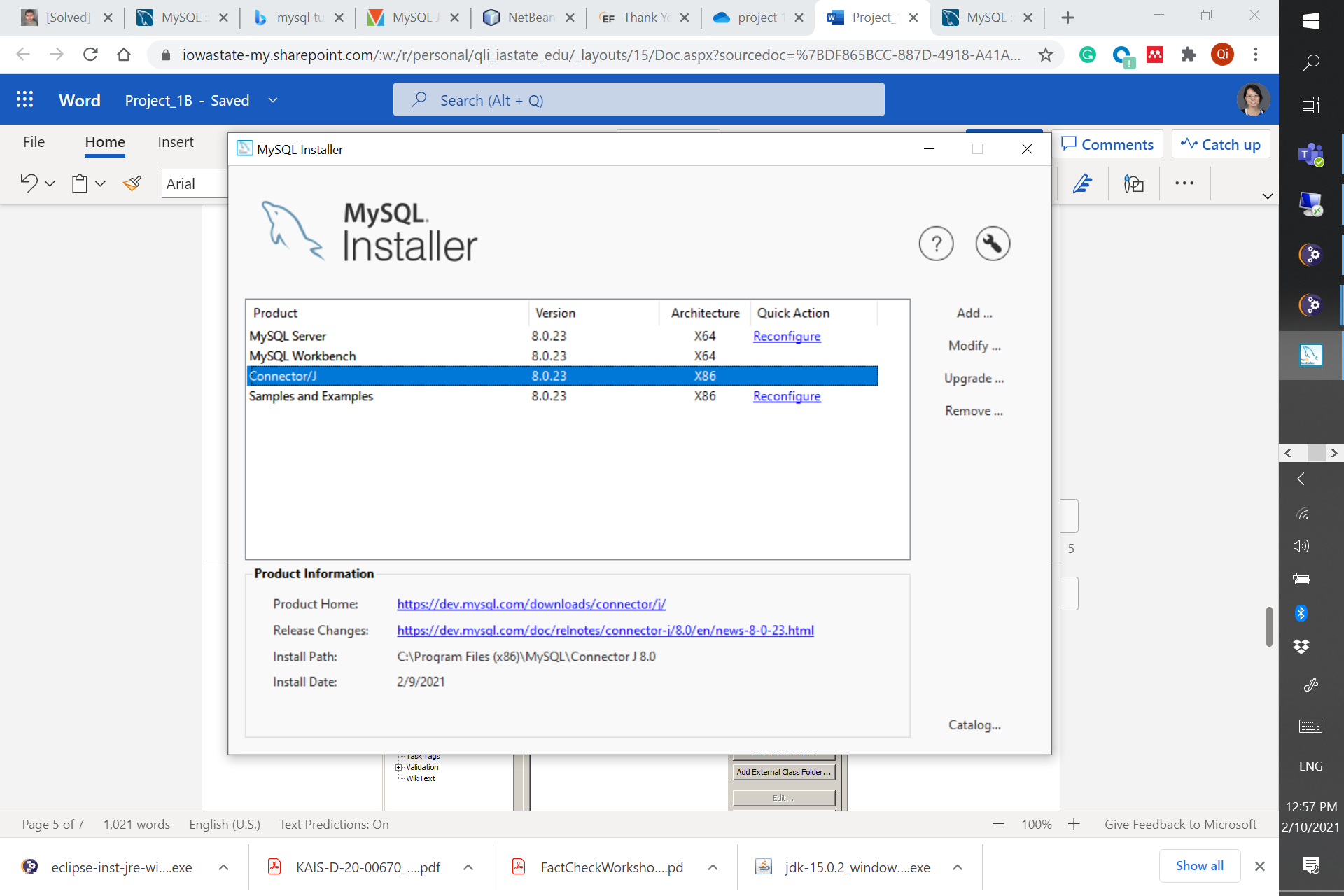
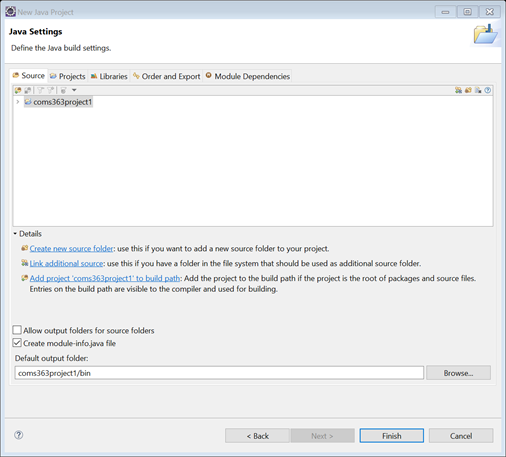
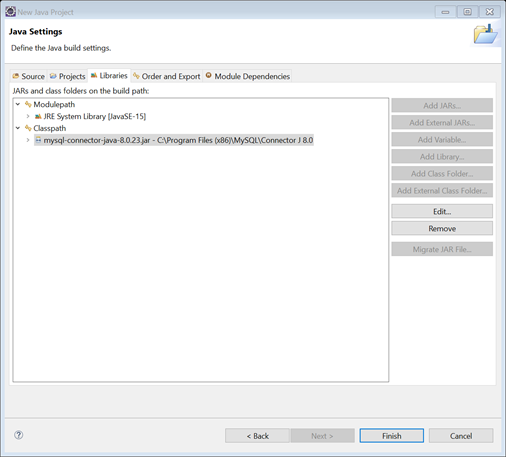
After execution, your program must modify the following information and print out the new tables where the modifications are conducted

1. Change the name of the student with ssn = 746897816 to Scott
2. Change the major of the student with ssn = 746897816 to Computer Science, Master.
3. Delete all registration records that were in “Spring2021”
4. **Remove duplicated records. If a group of courses have the same level and department\_code, only keep the one with the smallest course number and delete the rest. If a course is deleted, the corresponding record in register relation should also be deleted. (Hint: You cannot delete from a table and select from the same table in a subquery in MySQL. Check** [**this link**](https://dev.mysql.com/doc/refman/8.0/en/delete.html) **for multi-table deletes)**
5. **DropTables.java [5]**

After execution, your program must delete all tables.

**Submission Instruction**

*Submit all your java programs (\*.java) to Canvas. Be sure to*

1. *Name your files as required, i.e., CreateTables.java, InsertRecords.java, Query.java, ModifyRecords.java, DropTables.java;*
2. *Make each of these java files independent executable, i.e., each having its main() method;*
3. *Set user name to be “coms363” and password to be “password” in database authentication.*
4. Set up working environment using Eclipse (This instruction is based on Windows)
   1. Make sure that you have Java JDK installed in your computer, if not, you can get Java JDK at <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
   2. Download and install Eclipse IDE at <https://eclipse.org/downloads/> (choose “Eclipse IDE for Java Enterprise Java Developers”)
   3. Download and install Connector J at <https://dev.mysql.com/downloads/connector/j/>If you use mac, choose “platform independent”.  
      If you use Windows, search for MySQL installer – community app on your computer. If you see the following window, you have Connector J installed. You can also find the install path. If you don’t see Connector/J on the list, then click “Add” on the right, choose “MySQL connectors”, find Connector/J and install it.  
      
   4. Open Eclipse
   5. Create new Java project by go to “File > New > Other…” then select “Java Project”
   6. Give your Java project a name, then click “next”.   
      
   7. Next, add Connector J JAR file to your project build path. Click on “libraries->classpath” and choose “Add External JARs…”
   8. Go to directory that you installed Connector J and select Connector J JAR file (this can be vary based on directory you have installed. For Windows, you can find the path in installers). Then click Finish.  
      
5. Examples of Java codes

This link <https://docs.oracle.com/javase/tutorial/jdbc/basics/processingsqlstatements.html> gives you a tutorial on coding JDBC. For your convenience, we give you some sample code below.

* 1. Establishing a connection

import java.sql.\*;

private static Connection connect = null;

try

{

Class.forName("com.mysql.jdbc.Driver");

//Set up connection parameters

String userName = "*[username]*";

String password = "*[password]*";

String dbServer = "jdbc:mysql://mysql.cs.iastate.edu/*[schema]*";

//Set up connection

connect = DriverManager.getConnection(dbServer,userName,password);

}

Catch(Exception e)

{

}

* 1. Executing DML & DDL

Statement stmt = null;

stmt = connect.createStatement();

String sql = "INSERT INTO Registration " + "VALUES (100, 'Zara', 'Ali', 18)";

stmt.executeUpdate(sql);

* 1. Executing SQL query

ResultSet resultSet = null;

String sqlQuery = "";

String outputString = "";

sqlQuery = "SELECT \* FROM student";

stmt = connect.createStatement();

resultSet = statement.executeQuery(sqlQuery);

while(resultSet.next())

{

outputString += resultSet.getInt("sid") + "....";

outputString += resultSet.getString("student\_name") + "....";

outputString += resultSet.getString("student\_email") + "\n";

}

Note:

For testing your code, we will use username= ‘coms363’ and password= ‘password’. To set up this user account, run the following code as root user.

Run the following code in MySQL to setup proper username and password.

CREATE USER 'coms363'@'localhost' IDENTIFIED BY 'password';

GRANT ALL PRIVILEGES ON \*.\* TO 'coms363'@'localhost';

Once you run your Java code, you should see updates on MySQL (remember to refresh SCHEMAS)