Project 1.2

I. Project Description

Figure 1 shows an ER diagram for University database. This is the same in Project 1.1.

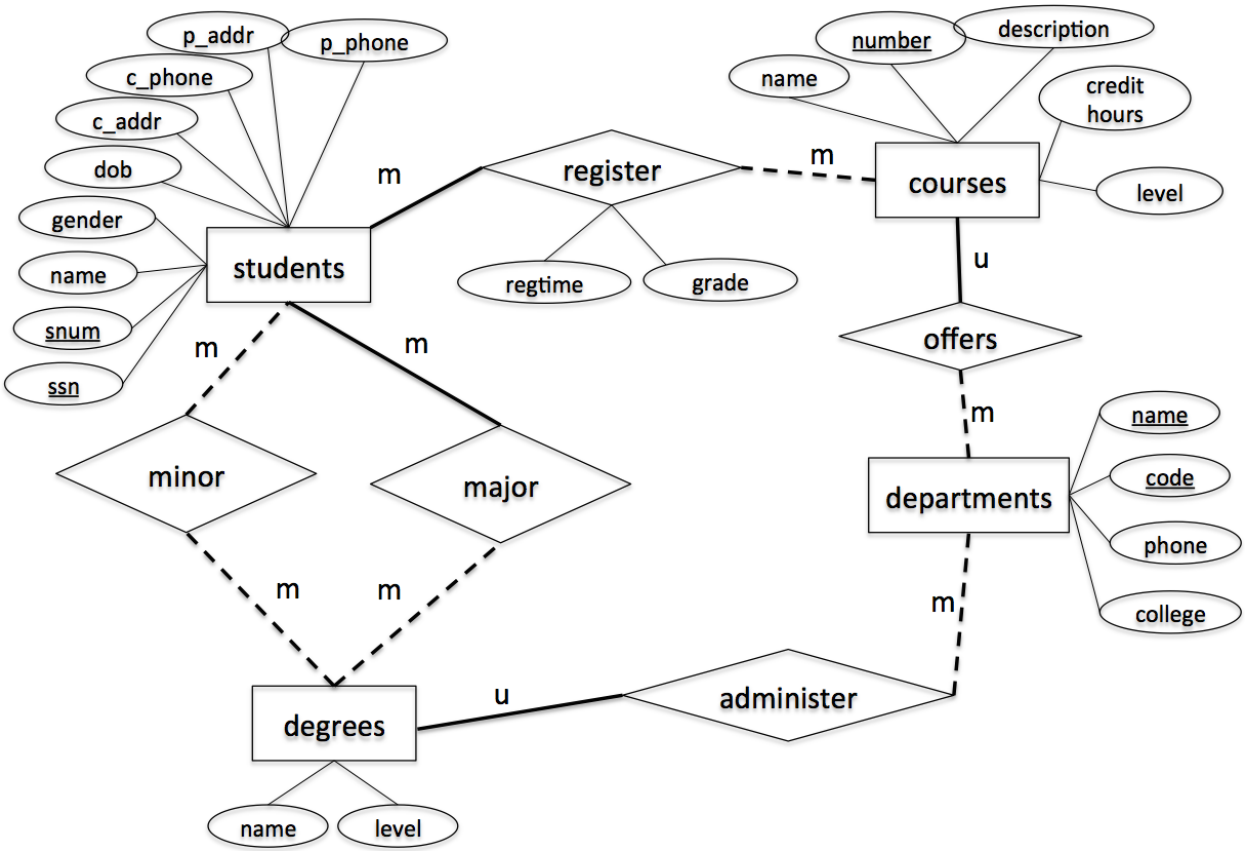


Figure 1. ER-diagram

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 the following tables. Each table must be created with the table name, attribute names and corresponding types and length as specified. Also, make sure to specify primary key, candidate key and foreign key (if any), accordingly.

- students
  a) Attribute, type and length: *snum: integer, ssn: integer, name: varchar(10), gender: varchar(1), dob: datetime, c_addr: varchar(20), c_phone: varchar(10), p_addr: varchar(20), p_phone: varchar(10)*
  b) Primary key: *ssn*

c) Candidate key: *snum*
d) Foreign key: *N/A*

- departments
  a) Attribute, type and length: *code: integer, name: varchar(50), phone: varchar(10), college: varchar(20)*
  b) Primary key: *code*
  c) Candidate key: *name*
  d) Foreign key: *N/A*
- degrees
  a) Attribute, type and length: *name: varchar(50), level: varchar(5),* department_code: integer
  b) Primary key: *name, level*
  c) Candidate key: *N/A*
  d) Foreign key: *department_code refers to code in table departments*
- courses
  a) Attribute, type and length: *number: integer, name: varchar(50), description: varchar(50), credithours: integer, level: varchar(20), department_code: integer*
  b) Primary key: *number*
  c) Candidate key: *name*
  d) Foreign key: *department_code refers to code in table departments*
- register
  a) Attribute, type and length: *snum: integer, course_number: integer, regtime: varchar(20), grade: integer*
  b) Primary key: *snum, course_number*
  c) Candidate key: *N/A*
  d) Foreign key: *snum refers to snum in table students, course_number refers to number in table courses*
- major
  a) Attribute, type and length: *snum: integer, name: varchar(50), level: varchar(5)*
  b) Primary key: *snum, name, level*
  c) Candidate key: *N/A*
  d) Foreign key: *snum refers to snum in table students, name & level refer to name & level in table degrees*
- minor
  a) Attribute, type and length: *snum: integer, name: varchar(50), level: varchar(5)*
  b) Primary key: *snum, name, level*
  c) Candidate key: *N/A*
  d) Foreign key: *snum refers to snum in table students, name & level refer to name & level in table degrees*

2. InsertRecords.java [Points: 15]

After execution, your program must insert the following records to the appropriate tables created by CreateTables.java.

- **students**

snum	ssn	name	gender	dob	c_addr	c_phone	p_addr	p_phone
1001	654651234	Randy	M	2000/12/01	301 E Hall	5152700988	121 Main	7083066321
1002	123097834	Victor	M	2000/05/06	270 W Hall	5151234578	702 Walnut	7080366333
1003	978012431	John	M	1999/07/08	201 W Hall	5154132805	888 University	5152012011
1004	746897816	Seth	M	1998/12/30	199 N Hall	5158891504	21 Green	5154132907
1005	186032894	Nicole	F	2001/04/01	178 S Hall	5158891155	13 Gray	5157162071
1006	534218752	Becky	F	2001/05/16	12 N Hall	5157083698	189 Clark	2034367632
1007	432609519	Kevin	M	2000/08/12	75 E Hall	5157082497	89 National	7182340772

- **departments**

code	name	phone	college
401	Computer Science	5152982801	LAS
402	Mathematics	5152982802	LAS
403	Chemical Engineering	5152982803	Engineering
404	Landscape Architect	5152982804	Design

- **degrees**

name	level	department_code
Computer Science	BS	401
Software Engineering	BS	401
Computer Science	MS	401
Computer Science	PhD	401
Applied Mathematics	MS	402
Chemical Engineering	BS	403
Landscape Architect	BS	404

- **major**

snum	name	level
1001	Computer Science	BS
1002	Software Engineering	BS
1003	Chemical Engineering	BS
1004	Landscape Architect	BS
1005	Computer Science	MS
1006	Applied Mathematics	MS

1007	Computer Science	PhD

- **minor**

snum	**name**	**level**
1007	Applied Mathematics	MS
1005	Applied Mathematics	MS
1001	Software Engineering	BS

- **courses**

number	**name**	**description**	**credithours**	**level**	**department_code**
113	Spreadsheet	Microsoft Excel and Access	3	Undergraduate	401
311	Algorithm	Design and Analysis	3	Undergraduate	401
531	Theory of Computation	Theorem and Probability	3	Graduate	401
363	Database	Design Principle	3	Undergraduate	401
412	Water Management	Water Management	3	Undergraduate	404
228	Special Topics	Interesting Topics about CE	3	Undergraduate	403
101	Calculus	Limit and Derivative	4	Undergraduate	402

- **register**

snum	**course_number**	**when**	**grade**
1001	363	Fall2020	3
1002	311	Fall2020	4
1003	228	Fall2020	4
1004	363	Spring2021	3
1005	531	Spring2021	4
1006	363	Fall2020	3
1007	531	Spring2021	4

3. **Query.java [Points: 55]**

After execution, your program must print out the following information

1) The student number and ssn of the student whose name is "Becky"
2) The major name and major level of the student whose ssn is 123097834
3) The names of all courses offered by the department of Computer Science
4) All degree names and levels offered by the department Computer Science
5) The names of all students who have a minor

4. **ModifyRecords.java [10]**

After execution, your program must modify the following information

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”,

5. **DropTables.java [5]**

After execution, your program must delete all tables.

Submission Instruction

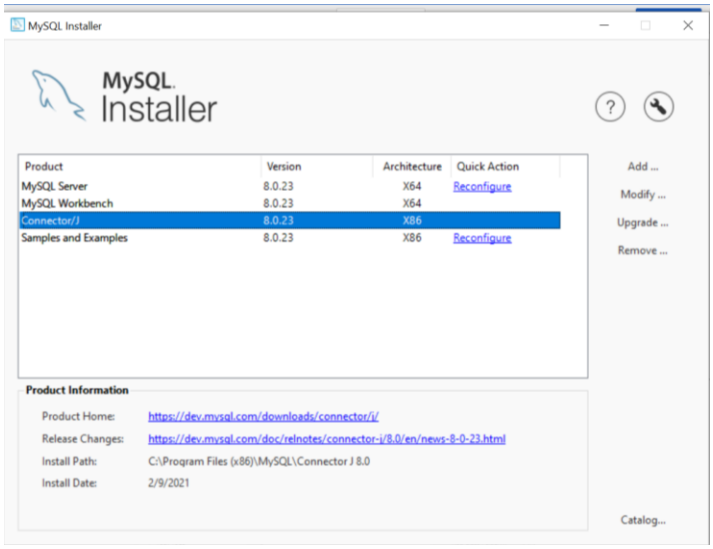
Submit all your java programs (.java) to Canvas. Be sure to*

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.*

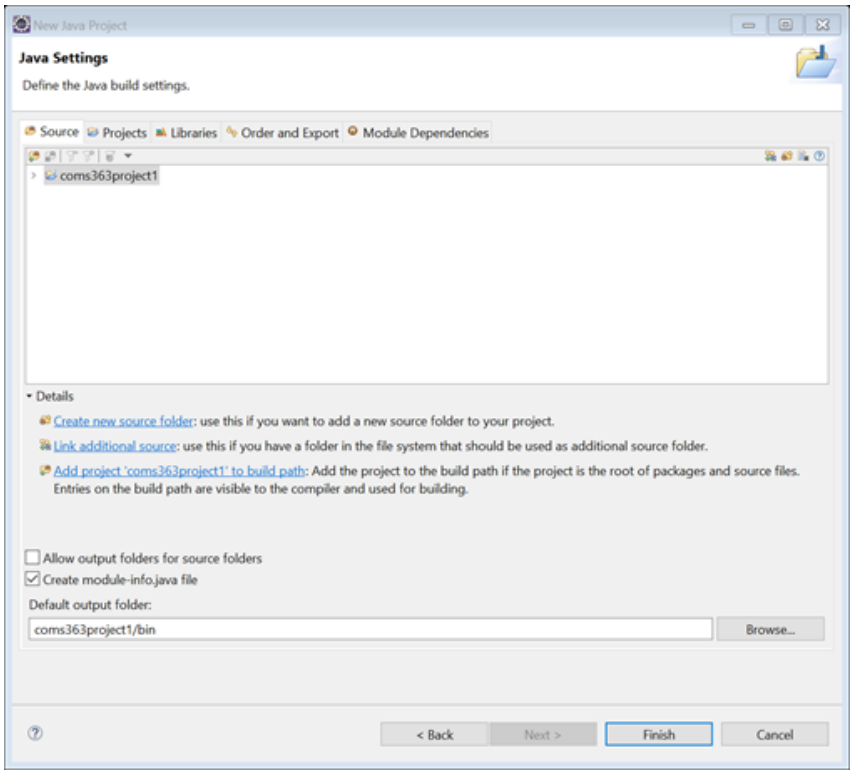
II. Set up working environment using Eclipse (This instruction is based on Windows)

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/

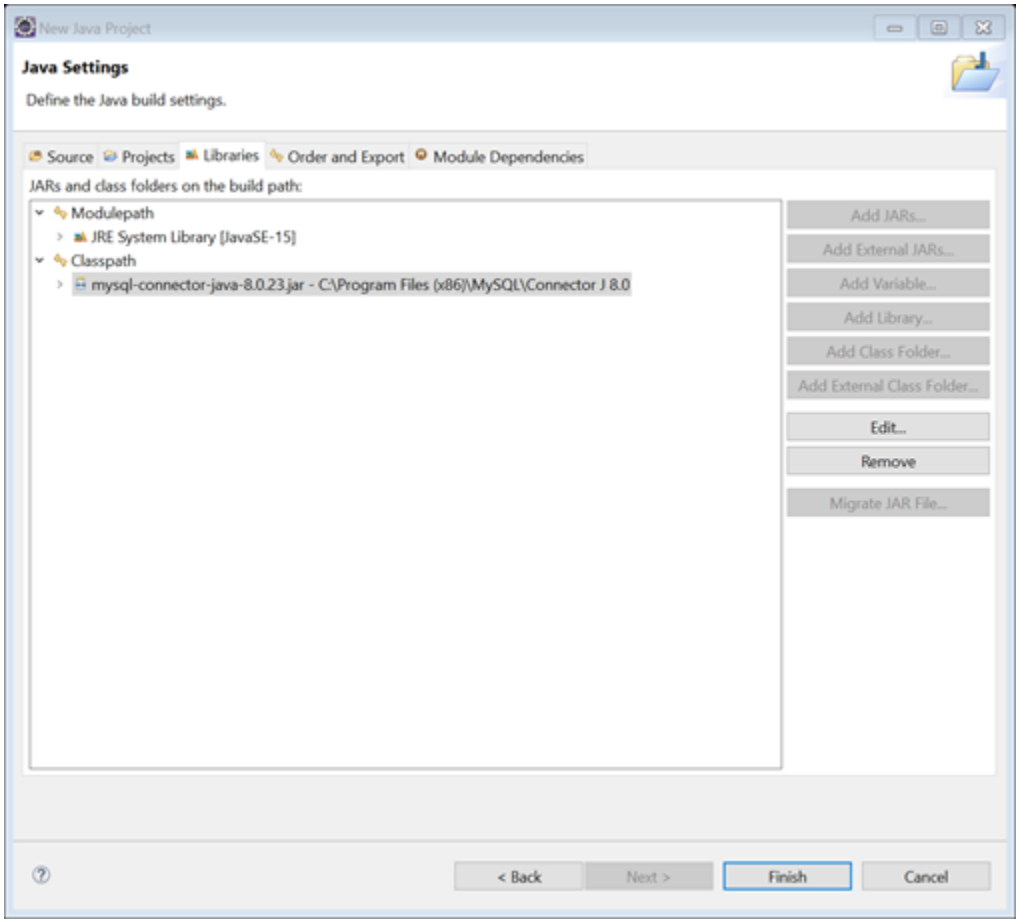
If you use Windows, search for MySQL installer – community app on your computer. If you see the following window, you have Connector J installed. You can also find the install path. If you don't see Connector/J on the list, then click "Add" on the right, choose "MySQL connectors", find Connector/J and 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.



III. 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)
{
}
```

2. Executing DML & DDL

```
Statement stmt = null;
stmt = connect.createStatement();
String sql = "INSERT INTO Registration " + "VALUES (100, 'Zara', 'Ali', 18)";
stmt.executeUpdate(sql);
```

3. 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";
}
```

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.

```
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)