**Practical: Model Driven Engineering - Exercise 2 (Optional)**

# Introduction

The practical exercises are not tied to a specific programming language, although an object-oriented (OO) one makes it easier to implement the examples presented in the practical. It is hoped that you will be able to use the programming languages with which you are most familiar or which you deem to be the most appropriate to complete each exercise. The key is to adopt a Model Driven Engineering (MDE) tool which targets your programming language of preference. Also, MySQL will be used as our database technology for MDE of database schemas. The practical will use:

* MySQL Workbench – a desktop application for modelling database schemas (tables, their relationships, data types and sample data), and

In the following sections we will first model a database schema (using entity relationship model), and build a MySQL Server Schema.

**Terms:**

Forward engineering in MDE is the ability to generate a software artefact from your model e.g. database and tables from entity relationship model. When changes are made to the model, the software artefact has to be re-generated.

Round trip generation in MDE is the feature that allows real-time reflection of changes: i.e. allows changes in software artefact to reflect in model and vice versa at real-time.

# Model Driven Schema: ER Model to Schema

This first section covers download and installation process of MySQL Workbench, design of Entity Relationship Model (ER Model), MySQL Server installation (using WAMP) and Schema generation from ER Model (Model driven).

Note: Workbench uses EER Model; which is enhanced-entity relationship model.

For this section we will develop a MySQL schema application for student records. The student record application is a schema with three entities:

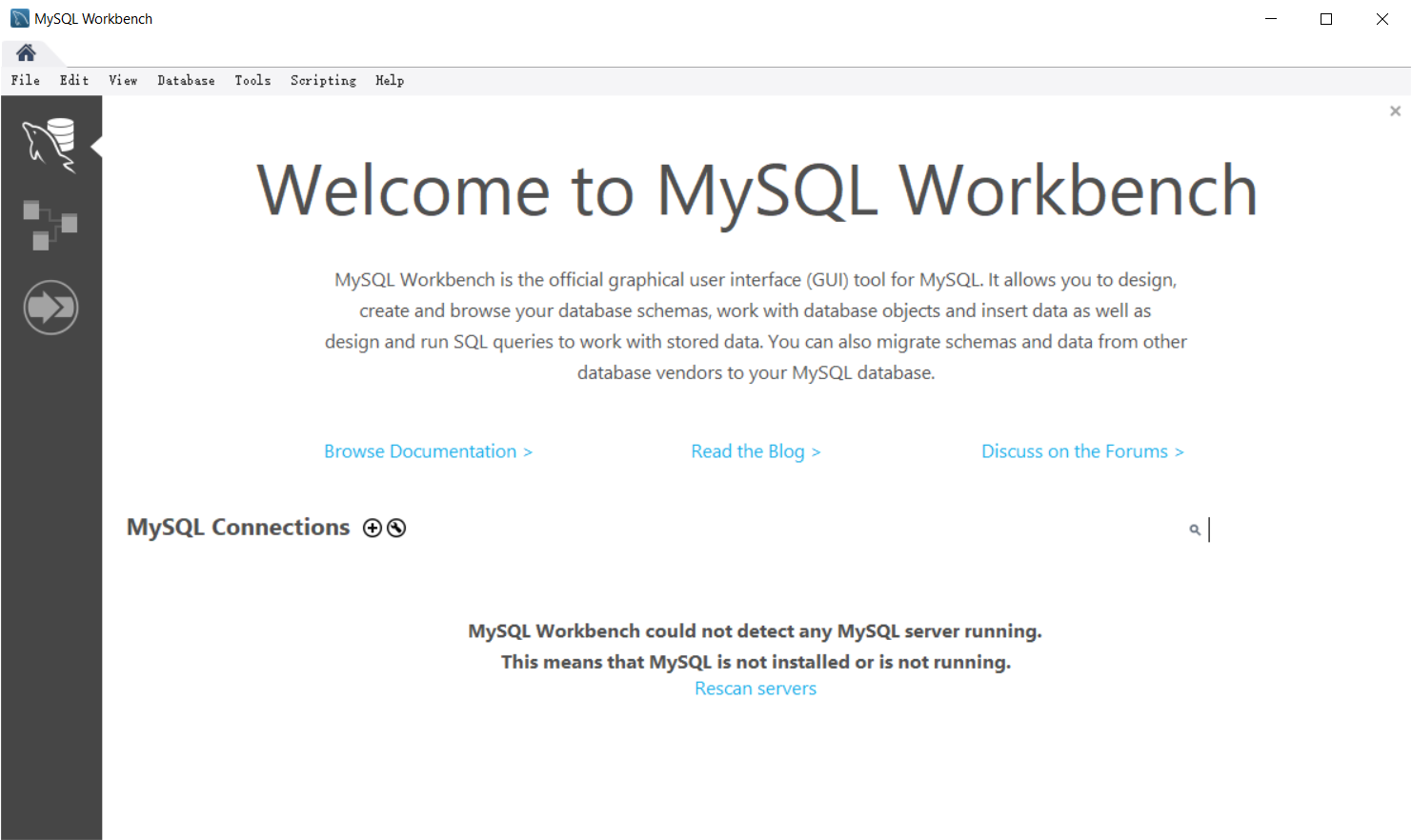
* Student
* Module
* Student\_has\_Module (a linking table)

From the schema description above, we can deduce that it would be appropriate for a student to be able to register for one or more modules. And also valid to say a module can have one or more students. We call this many to many relationship mapping; *Student\_has\_Module*. We also further extend our mapping link to include a *score* information for students on a module. Our relationship mapping would look like the figure below:



## Workbench installation

* Go to <https://dev.mysql.com/downloads/workbench/>
* Select Microsoft Windows as Platform (if using PC)
* Download the Windows (x86, 64-bit), MSI Installer
* Run the downloaded file
* Click Next and use Complete Install option
* Click Finish to Install
* When prompted, click Yes to accept User Account Control
* When Installation is done, Click finish to launch MySQL Workbench now.



## Designing an EER Model

* In the menu bar go to; File > New Model
* Click Model > Add Diagram (an EER Diagram window opens, displaying an empty work area with a left panel of icons.)

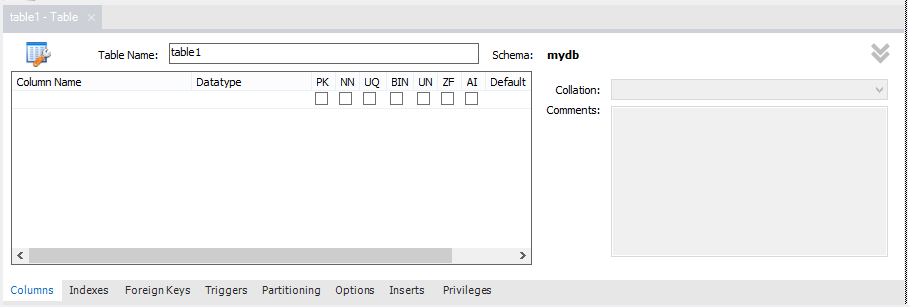


* Click on the table icon from the left panel
* Then click on the work area to drop the table. This displays table1 in the work area.
* Repeat step 3 and 4 to reproduce a second table (table2) in the work area.

The screen at the work area should look as below:



* Double click on table1 to display its properties window at the bottom of the work area, as below:



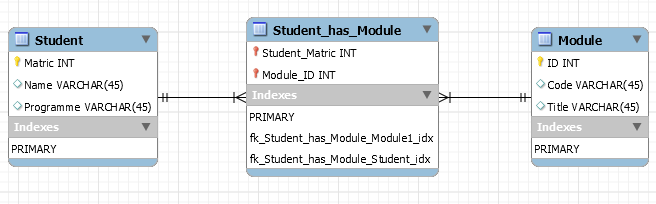
* Change the Table Name to **Student**
* By double-clicking under the Column Name, add the following columns: **Matric**, **Name** and **Programme**.
* Ensure the first column (i.e. Matric) is your identity column by checking PK and NN, which means the column is primary key and not null respectively.
* Also ensure that Matric is set to INT, while other fields are set to VARCHAR (this should be done by default).
* Double click table 2, to display its properties window
* Change the Table Name to **Module**
* Add the following columns: **ID**, **Code**, **Title**

(Set the ID to auto increment, by checking AI column. The other columns can be VARCHAR data type)

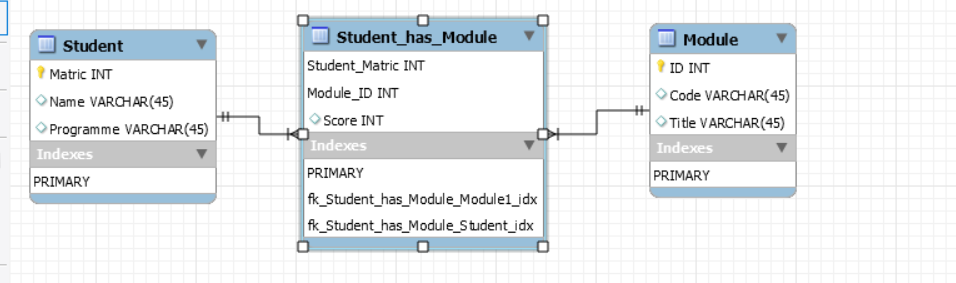


* Click the many-to-many icon
* Click on student table and then click on Module table respectively

This creates a third table (Student\_has\_Module) giving the view below;



* Double click **Student\_has\_Module** table to display its properties window
* Add a new column called **Score**, with an INT data type.
* The final work EER looks like the figure below:



* Click File > Save Model

Save as **StudentRecord.mwb**

* In the Workbench, click on MySQL Model top menu beside the home icon.



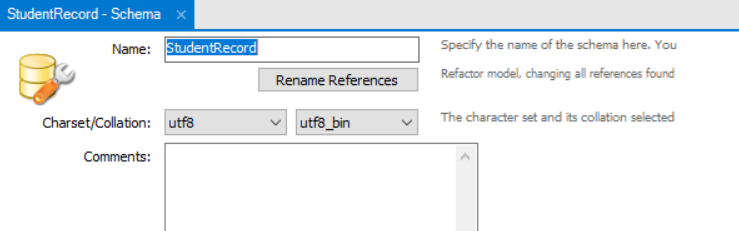
This shows you our schema so far, however it is called mydb

* To rename it right-click mydb, select Edit schema

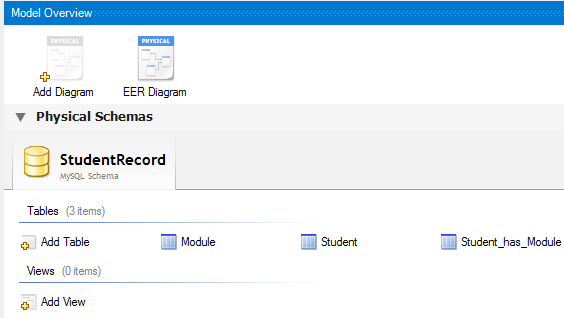
(This opens schema property window)

* Change the schema name to **StudentRecord**

(This will match to the name of our database in RDBMS – in later steps)



* Click File > Save Model
* Our final view will look like the figure below;



## MySQL as Target of EER

### Download and Installation of WAMP

Next we need a MySQL Server running, so we will use WAMP to install MySQL server to expedite the process.

* Go to <https://sourceforge.net/projects/wampserver/files/latest/download>
* Download WAMP Server (we will use 64bit version)
* Run the installer, and follow the prompt to install WAMP Server
* After installation, start WAMP Server. The WAMP Server icon appears on the right of your taskbar – a green icon shows that all WAMP Services (including MySQL Server) is running successfully. 
* Click the WAMP Server icon on the taskbar, then click phpMyAdmin, to open a web-based DB management interface.

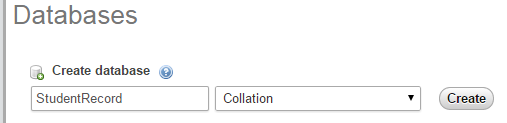
(Or simply browse to <http://localhost/phpmyadmin>)

### Create Database in MySQL

* In the phpMyAdmin web interface at <http://localhost/phpmyadmin>

Click Databases menu

* Name your database **StudentRecord**, and click Create.



* Click on the studentrecord database and then click Privileges
* Click Add User
* Supply authentication details for the database, we will use **developer** as our username and password, and **localhost** as our host (selecting Local, sets the host to localhost).
* Ensure that: Grant all privileges on database "studentrecord", is checked.
* Click Go to finish.

### Forward Engineering to MySQL Server

* Open your MySQL Workbench project (**StudentRecord.mwb**)
* In the menu bar go to; Database > Forward Engineer…

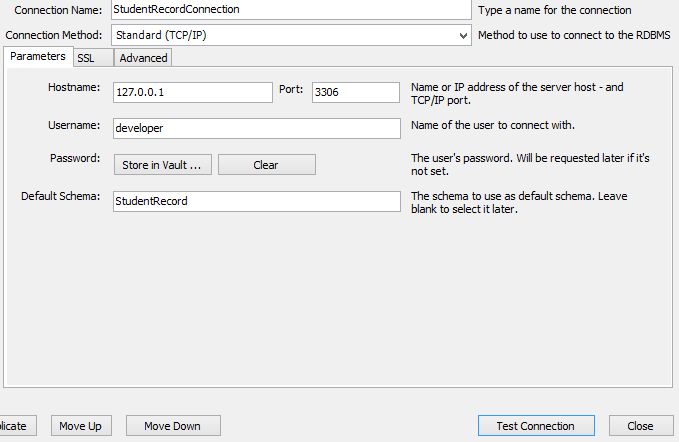
(This opens Forward Engineer to Database window)

* For Stored Connections dropdown, select Manage Stored Connections…

(This opens Manage DB Connections window)

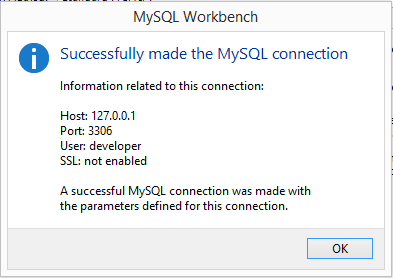
* Click New
* Give your new connection the name; **StudentRecordConnection**
* For Username, type **developer**
* For Password, Click Store in Vault, and type **developer**
* For Default Schema, type **StudentRecord**
* Hostname and Port should be the default; **127.0.0.1** and **3306**, respectively.

(Your settings will look like the window below)



* Click Test Connection

(If you have sequentially followed the steps so far, you should see a success message as below)



* Click OK, then close your Manage DB Connections window.

(Now your Stored Connection is set to StudentRecordConnection, and the parameters fields automatically filled)

* Click Next

(…to Set Options for Database to be Created)

* Ensure the following is checked:
  + Generate INSERT statements for tables
  + Include model attached scripts
* Click Next

(…to Select Objects to Forward Engineer)

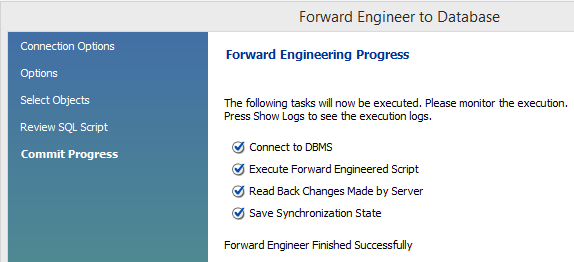
* Ensure Export MySQL Table Objects is checked
* Click Next

(…to Review the SQL Script to be Executed)

* Click Next

(…to Forward Engineer the EER Model to the MySQL database. If you encounter an error, review your connection, ensure WAMP Server is running, and refer to the previous steps.)

* If your process was successful. You will get a success message as shown below:



* Click Close to finish.

**Verify database tables in MySQL Server**

* Go to <http://localhost/phpmyadmin> in your browser
* Click StudentRecord database
* You can see that three tables (module, student, student\_has\_module) have been created in our empty schema – which executes/implements the entities (Module, Student, Student\_has\_Module) in our model.
* Since in our database options set up of forward engineering phase, we checked *Generate INSERT statements for tables*, we can insert tables from our model and drive it to our MySQL server.

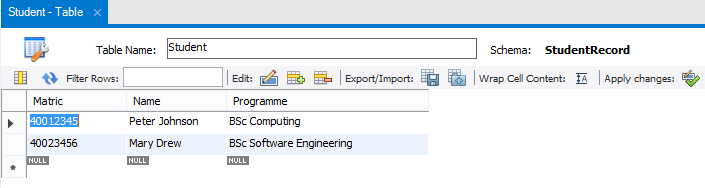
**Generating data from EER**

* Go back to your Workbench and open the EER work area
* Double Click on the Student entity

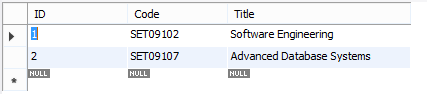
(This opens the Student - table property tab at the bottom of the work area)

* Click Inserts tab
* Populate the table with sample data and click Apply changes to data icon 

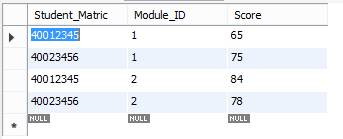
(Your final table should look as follows)



* Also populate the table for Module and apply changes.



* Also populate the table for Student\_has\_Module and apply changes.



* From the menu bar, click File > Save Model
* Click Database > Forward Engineer…

(Our StudentRecordConnection connection is already set and configured for this project, so we need not make any changes)

* Click Next > Next > Next > Next
* (We should receive a success message: Forward Engineer Finished Successfully)
* Click Close.

**Verify data in MySQL Server**

* Go to <http://localhost/phpmyadmin> in your browser

(ensure to refresh the page if it is already open)

* Click StudentRecord database
* Click on each table to see the data populated.
* You have successfully driven your data from model to schema in MySQL server

# Further Work

* Investigate ways to ensure data integrity of your model, and implement it in the EER model at your Workbench project (e.g. data type checking)
* Attempt to drive your model into a Microsoft SQL database project. You can accomplish this with Visual Studio 2019 IDE to manage local database file. See the link <https://docs.microsoft.com/en-us/visualstudio/data-tools/create-a-sql-database-by-using-a-designer?view=vs-2019>
* Attempt to create a class model for the postgraduate student type, and make the courseDuration fields fixed to 1 year. How would you present this field in OOP?