

# Medical Informatics: Lab 1

## Introduction to MySQL & MySQL Workbench

### Introduction

In this lab you will learn how to use MySQL and MySQL Workbench to navigate, query and modify relational databases. It is assumed that you have successfully installed MySQL and MySQL Workbench on your personal machine. To make the most of the lab, please download the following files from the course website: i) lab1\_db.sql and ii) lab1\_queries.sql.

### Overview of the “lab1” database

The database that we’re going to be working with in this lab helps keep track of patients, their GPs and drugs prescribed by GPs to patients. In Figure 1 you can see a visual representation of the database, which displays the different tables, their columns and the links between them.

### MySQL jargon

**Schema:** this is the term used in MySQL to refer to a database. So a schema is a set of interrelated database objects, such as tables, table columns, data types of the columns, foreign keys, and so on.

**Server:** the machine where the schema is hosted. To access it, you need to make a connection to the server. Note that the server can be your local machine or a remote machine (e.g. in the local hospital or the University), and you typically need to have the right credentials to access it.

**Model:** MySQL Workbench provides modelling functionality, which simplifies database design and maintenance. Models are useful for making changes and experimenting with the structure of your tables, without having to enforce them on the schema on the server (which may be shared between several people). When desired, models can be translated into physical databases.



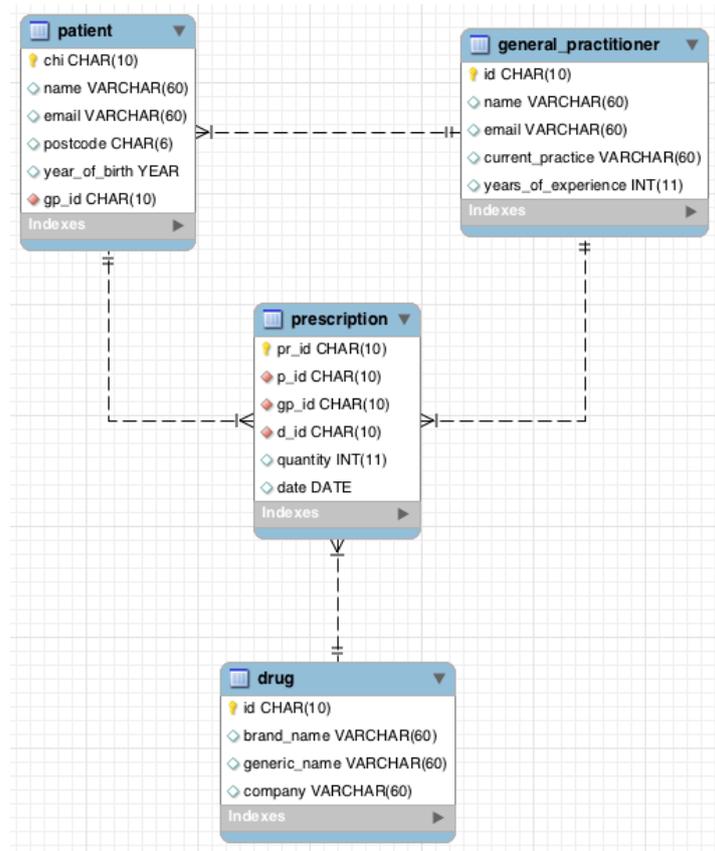


Figure 1: Visual representation of the database to be used in Lab 1.

## Part 1: Creating a new MySQL connection

Further details can be found at the corresponding page of the MySQL Workbench Manual: <http://dev.mysql.com/doc/workbench/en/wb-getting-started-tutorial-create-connection.html>

1. Launch MySQL Workbench. You will be presented with the Home window.
2. From the MySQL Workbench Home window, click the [+] icon near the MySQL Connections label to open the Setup New Connection wizard.
3. Give any name you like to the Connection Name value. For this lab, we will call it "Lab1Connection".
4. Fill out the connection details.
  - a. For this lab, we'll keep the default connection values.
  - b. If you've been given access to a particular database, then provide the hostname, username, password and Default Schema values.
5. If you wish, you can hit the Test Connection button to check the connection parameters.
6. Optionally, you can choose to Configure Server Management by clicking the corresponding button.
7. Press the OK button to complete the process.

### Connecting to the server

From the MySQL Workbench Home window, click on the connection you are interested in. You might be asked for a password if it is not already saved.

The Navigator panel has a Management tab, which allows you to monitor and configure your selected MySQL connection (e.g. manage the server status, check client connections, add users and privileges, etc.).

The Navigator panel has a Schemas tab, enlisting the schemata you have access to, along with their tables, columns, etc.

### [optional] Viewing connection information

In order to view connection information from the home page:

- If on Windows or Linux: hover over the right side of a connection title and click the title
- If on OS X: hover over a connection title and click the little (i) that appears in the bottom right corner

### [optional] Managing connections

You can view and change the parameters of your connections through the Manage Server Connections dialog. To launch this from the main menu, select Database, Manage Connections.

## Part 2: Importing schemata and data to the server

Further details can be found at the corresponding page of the MySQL Workbench Manual: <http://dev.mysql.com/doc/workbench/en/wb-admin-export-import-management.html>

1. After connecting to the server, select Server and then Data Import.
2. Choose the project folder or self-contained SQL file to be imported.
  - a. If you're importing a self-contained SQL file (i.e. a .sql file), you can choose the schema that the data will be imported to, or choose New to define a new schema.
    - i. In this lab, we'll import the self-contained SQL file `lab1_db.sql` without choosing or creating a new schema.
  - b. If you're importing a project folder, you can select the database objects to import.
3. Click Start Import to begin the import process.
4. Once the process completes, you should be able to see the new database (in our case *lab1*) in the list of schemata in the Navigator panel. If not, simply hit the refresh button in the schemata panel.

## Part 3: Navigating tables and data in the live database

Further details can be found at the corresponding page of the MySQL Workbench Manual: <http://dev.mysql.com/doc/workbench/en/wb-admin-export-import-management.html>

1. Within the Navigator, you can view the schemata you have access to (including the imported “lab1” schema).
2. To get an overview of a schema, do one of the following:
  - a. Right-click on the schema within the Navigator panel and select Schema Inspector.
  - b. From the Navigator panel, click on the arrow next to a schema to view the tables. You can view the columns of a table in a similar way.
3. To have a quick look at the first 1,000 rows within a table, right-click on a table of interest in the Navigator panel, and choose Select Rows - Limit 1000. This displays the query and the results grid.
4. To view the definition of a table (e.g. columns and their data types, primary and foreign key constraints, etc.), right-click on a table name in the Object Viewer and choose ALTER TABLE. (Note that, even though this is meant to be for modifying a table, we’ll just view its definition here.) In our case, we’ll inspect the “patient” table.
  - a. In the Columns tab you can see the names of the columns, the order in which they appear in the table, their data types and their properties.
    - i. For the complete list of data types in MySQL, please check <http://dev.mysql.com/doc/refman/5.7/en/data-types.html>.
    - ii. Regarding properties, PK refers to "PRIMARY KEY", NN refers to "NOT NULL" and AI refers to "AUTO\_INCREMENT". For the complete list of column properties, please check <http://dev.mysql.com/doc/workbench/en/wb-table-editor-columns-tab.html>.
  - b. In the Foreign Keys tab you can see information about any foreign key constraints.
    - i. In the case of the “patient” table, we can see that there is a foreign key constraint on the gp\_id column, referencing the id column in the “general\_practitioner” table. We can also see what is supposed to happen “on delete” or “on update”.

*Important note:* The best way to navigate a database and get a sense of its structure (but not the data) is by reverse engineering the database, so as to get the model and corresponding EER diagram. This is covered in Part 6 of the lab.

## Part 4: SQL querying

Further details can be found at the corresponding page of the MySQL Workbench Manual: <http://dev.mysql.com/doc/workbench/en/wb-sql-editor.html>

1. Type, in the SQL query panel, the queries that you would like to execute.
  - a. Some tips:
    - i. After connecting to the server, remember to select the database that you're going to be querying. You can do this by double-clicking the name of the database in the Navigator panel.
    - ii. Remember to include a ';' at the end of each statement.
    - iii. You may find the auto-complete option useful when writing your queries.
    - iv. Within the Format submenu of the Edit menu, you can find the following formatting options: Beautify Query, UPPERCASE Keywords, lowercase Keywords, and Un/Comment.
  - b. For this lab, we'll try the following queries:
    - i. `SELECT * FROM patient;`
    - ii. `select * from patient where year_of_birth < 1970;`
    - iii. `SELECT name FROM general_practitioner WHERE years_of_experience >= 15;`
    - iv. `SELECT P.name FROM patient AS P, general_practitioner AS GP WHERE P.gp_id = GP.id AND GP.name = 'David Taylor';`
2. To run a query, select it and click on the Execute button from the SQL Query menu. Alternatively, to run the query under the cursor, click on the corresponding button from the SQL Query menu.
3. You should be able to see the output on your screen, as well as the log of executed statements (at the bottom of the screen).
4. You can export the results to an external file (e.g. in CSV format) by clicking on the Export button within the Results Window.
5. You can save your SQL script to a file for future use by clicking on the Save button from the SQL Query menu. Highly recommended! In this lab, you can save your current script as `myQueries.sql`, anywhere you like on your computer.
6. You can also open an existing SQL script file to edit and execute the queries it contains.
  - a. In this lab, we'll be using the file `lab1_queries.sql`. Open it and try running a few queries.

## Part 5: Modifying tables in the live database and adding data

### Modifying tables in the live database

Further details can be found at the corresponding page of the MySQL Workbench Manual: <http://dev.mysql.com/doc/workbench/en/wb-table-editor.html>

1. To modify a table in the live database, right-click on a table name in the Object Viewer and choose ALTER TABLE.
2. Within the new tab, you can add or modify columns, indexes, foreign keys, etc.
  - a. For this lab, we will modify the “patient” table, in particular we will add a new column right after the “gp\_id” column. The new column will be called “gender” and it will be of type VARCHAR(45).
3. Click Apply to apply these changes to the live MySQL server.

### Adding data

Further details can be found at the corresponding page of the MySQL Workbench Manual: <http://dev.mysql.com/doc/workbench/en/wb-sql-editor.html>

1. From the Navigator panel, right-click on a table of interest and choose Select Rows - Limit 1000. This displays the query and the results grid.
2. To add data manually, simply input data into the results grid and click Apply to apply these changes to the live MySQL server.
  - a. For this lab, we will manually add gender information into the newly created “gender” column in the patient table. (Note: Alastair, Andrew, Hamish and Iain are “male”, while the rest of the patients are “female”).
3. To add data by importing an SQL script, select Server and then Data Import. Choose self-contained SQL file and choose the schema that the data will be imported to or choose New to define a new schema. Finally, click Start Import.
4. To add data to a table by importing a CSV file, right-click on that table and choose Table Data Import Wizard. Select the file to import, specify the destination and import settings, and click Finish to complete the process.

## Part 6: Reverse engineer the live database and get the EER diagram

Further details can be found at the corresponding page of the MySQL Workbench Manual: <https://dev.mysql.com/doc/workbench/en/wb-reverse-engineer-live.html>

MySQL Workbench allows you to reverse engineer your live database, so as to get a so-called “model”. Models are useful for making changes and experimenting with the structure of your tables, without having to enforce them on the live database (which may be shared between several people).

1. To reverse engineer the live database, choose Database and then Reverse Engineer from the main menu. This will launch the Reverse Engineer Database wizard.
2. Within the wizard, first choose the connection from which you can access the database (in our case it is Lab1Connection), and hit Next.
3. Then select the schema that you want to reverse-engineer, and hit Next. In our case, we will choose the lab1 schema.
4. In the next screen you can choose which tables to include. All tables are selected by default. We'll accept this for the purpose of this lab, and hit Execute.
5. The next screen in the wizard informs us whether the operation was successful. Hit Next, and in the next (and final) screen, hit Finish.
6. You are then presented with the Model and the EER Diagram, which is a visual representation of the database. To save the model, click File and then Save from the menu. In our case, we'll call it lab1\_model.mwb.

Note: You can navigate and modify the model, if you wish. You can also push these changes to the live database (this is called forward engineering or synchronising). We won't learn how to work with models in this lab, but if you're curious, you can have a look at Chapter 9 of the MySQL Workbench Manual: <https://dev.mysql.com/doc/workbench/en/wb-data-modeling.html>