LAB SESSION 1

BT2102 Data Management and Visualisation · Semester 2 · AY2017/18

ABOUT THIS SESSION

In this first lab session, we will complete a few introductory tasks, making sure your set up of MySQL and Tableau are complete. We will also review some key concepts on normalisation and ER modelling. You should use this lab document to guide you through completing tasks, but activities will be injected into the session by your TA, as well. At various points, we will also pause to discuss particular concepts and practices.

TASK 1: GETTING STARTED WITH MYSQL

Let's make sure your installation of MySQL is correct and complete.

INSTALLATION

1. If you haven't already done so, make sure you are logged into an administrator account on your computer, visit https://dev.mysql.com/downloads/ , and download the latest version of the MySQL Community Server for your operating system. The main focus is on MYSQL COMMUNITY SERVER (the current version is 5.7). Some other tools listed (MySQL Workbench, Notifier, etc) can be useful, too. As far as possible, use the installation packages (*.msi* downloads for Windows, *.dmg* downloads for MacOS) provided by MySQL, and not the *.zip* archives as these require additional steps for setting up.

   For help on installations, refer to these guides:
   Windows: https://dev.mysql.com/doc/refman/5.7/en/windows-installation.html
   MacOS: https://dev.mysql.com/doc/refman/5.7/en/osx-installation.html

STARTING MYSQL SERVER (MYSQLD)

2. After installation is complete, we need to start the MySQL server – *mysqld* (ie. the mysql daemon). You can start the server using MySQL Notifier (in Windows' system tray, or check Start > MySQL > MySQL Notifier), or the MySQL Preference Pane (in MacOS' preferences).

   Windows: https://dev.mysql.com/doc/mysql-windows-excerpt/5.7/en/windows-notifier-usage.html
   Mac: https://dev.mysql.com/doc/refman/5.7/en/osx-installation-prefpane.html

   If necessary, you could also use the command line interface to start mysqld:

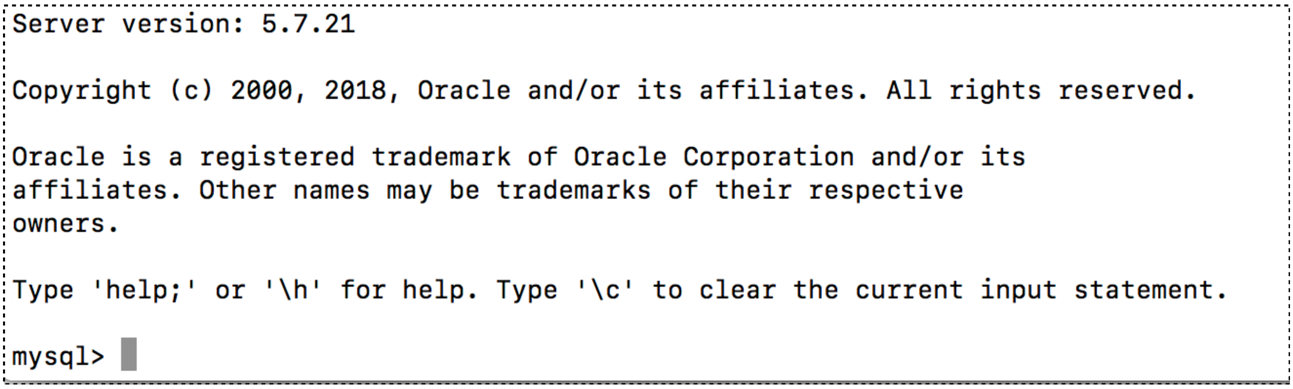
   Windows: https://dev.mysql.com/doc/refman/5.7/en/windows-start-command-line.html

CONNECTING A CLIENT TO THE MYSQL SERVER

3. With the server successfully running, we can now connect to our DBMS: https://dev.mysql.com/doc/mysql-getting-started/en/#mysql-getting-started-connecting

   For Mac OS, this can be done from the Terminal application. For Windows, this can be done from the MySQL Command Line Client or from the Windows command prompt. You may be prompted for the MySQL temporary root password at this point. If you received one during the installation process, this is where you will enter it.

4. Once you have connected your client (Terminal / Command Prompt), you will see the MySQL shell (similar to the image below), indicating that MySQL is successfully running.



```
Server version: 5.7.21

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> █
```

INSTRUCT THE DBMS: CHANGE THE DEFAULT ROOT PASSWORD

5. If this is your first time connecting to the DBMS, change your MySQL root password using (you can change *newpassword* to any password you like):

```
mysql> ALTER USER ‘root’@‘localhost’ IDENTIFIED BY ‘newpassword’;
```

INSTRUCT THE DBMS: SHOW AND DESCRIBE EXISTING DATABASES

6. Use the following command at the MySQL shell to ask the DBMS to show you the existing databases:

```
mysql> SHOW DATABASES;
```

   Notice all MySQL instructions end with a “;”. The instruction will display a list of existing databases in the DBMS (these are likely to be administrative databases the DBMS uses itself). One of these databases should be named *mysql*. Let’s examine that database:

```
mysql> USE mysql;
mysql> SHOW TABLES;
```

The USE statement above tells the DBMS we want to refer/use a specific database. The SHOW TABLES statement list the many tables in that specific database. One of these tables is named *user*. Let's see what fields that table consists of:

```
mysql> DESC user;
```

DESC describes a table by listing its fields, along with the type of information that field contains.

INSTRUCT THE DBMS: CREATE A NEW DATABASE

7. Recall the database we designed in our Thursday class to capture the *Better Books and Bargains* invoice information. It has 4 tables:

   INVOICE (InvoiceID, OrderDate, InvoiceDate, CustomerID)
   CUSTOMER (CustomerID, BillingAddr, ShippingAddr)
   ITEM (InvoiceID, ProductCode, Qty)
   STOCK (ProductCode, Description, Price, VAT)

   Let's create this new database in MySQL:

   ```
   mysql> CREATE DATABASE betterbooks;
   mysql> SHOW DATABASES;
   mysql> USE betterbooks;
   ```

   The list of databases should include a new one called *betterbooks*. Let's create one of its tables - *Stock*.

   ```
   mysql> CREATE TABLE Stock (ProductCode CHAR(15), Description CHAR(50), Price
   DECIMAL(5,2), VAT DECIMAL(4,2), PRIMARY KEY (ProductCode));
   ```

8. The terms *CHAR(15)*, *Char(50), Decimal(5,2)* and *Decimal(4,2)* are data types for each field. MySQL data fields can store many different types of data. Read about them here: https://dev.mysql.com/doc/refman/5.7/en/data-types.html. The PRIMARY KEY part of the CREATE TABLE statement is a *constraint* we're putting on the table. It sets one of the fields up as the PRIMARY KEY, which means that all values in that field must be unique and must not be blank.

9. **ON YOUR OWN:** Create the remaining tables in the *betterbooks* database. Note some constraints:
   CustomerID in **INVOICE** must be linked to CustomerID in **CUSTOMER**.
   InvoiceID and ProductCode in ITEM must be linked to the fields in INVOICE and STOCK, too.
   These constraints are called FOREIGN KEYS. How do you add foreign key constraints into your CREATE statement? Finally, what are the best data types of each field?

10. Insert some data into the *Stock* tables of the database, using the INSERT statement:

```
mysql> INSERT INTO Stock VALUES(‘9781906040130’, ‘Memory of Flames’, ’7.50’,
’20’);
```

11. ON YOUR OWN: Insert data into the other tables, as well.

Add another Stock product:
Product Code - 9781906040376
Description - Strangled in Paris
Price - 7.50
VAT - 20

Add a Customer:
Customer ID - 5417678U
Shipping Address - Loxley University, Loxley LO1 5XC
Billing Address - 8 Mien Pl, Sheffield S6 9JH

Add an Invoice:
InvoiceID - 21345454398
Order Date - 11/12/12
Invoice Date - 11/12/12
CustomerID - 5417678U

Add two Item Qty:
InvoiceID - 21345454398
ProductCode - 9781906040376
Qty - 1

InvoiceID - 21345454398
ProductCode - 9781906040130
Qty - 1

12. When you have finished, let you lab tutor know so he can verify.

TASK 2: PROBLEM > ER DIAGRAMS > DATA DECISIONS > MYSQL

Consider the following problem:

A database is needed to keep track of student enrolments in classes and students' final grades.

The university is organised into faculties, and each faculty has a unique name, a main office and phone number, and a particular faculty member who is dean. Each faculty administers a number of academic departments. Each department has a unique name, a unique code number, a main office and phone, and a particular faculty member who heads the department. We need to keep track of the start date when that faculty member began heading the department.

A department offers a number of courses, each of which has a unique course name, a unique course code number, a course level (this is coded as 1 for freshman level, 2 for sophomore, 3 for junior, 4 for senior), a course credit hours, and a course description.

The database also keeps track of instructors; and each instructor has a unique identifier, a name, office, phone, and rank; in addition, each instructor works for one primary academic department.

The database will keep student data and stores each student's name (composed of first name, middle name, last name), student ID (unique for every student), address, phone, major code, and date of birth. A student is assigned to one primary academic department. It is required to keep track of the student's grades in each section the student has completed.

Courses are offered as sections. Each section is related to a single course and a single instructor and has a unique section identifier. A section also has a section number (this is coded as 1, 2, 3, . . . for multiple sections offered during the same semester), semester, year, classroom (this is coded as a combination of building code and room number within the building), and days/times (for example, 'MWF 9am-9.50am' or 'TR 3.30pm-5.20pm'— restricted to only days/time values).

The database needs to keep track of all the sections offered for the past several years, in addition to the current offerings. The section identifier is unique for all sections, not just the sections for a particular semester. The database keeps track of the students in each section, and the grade is recorded when available (there is a many-to-many relationship between students and sections). A section must have at least five students.

Draw an ER diagram to represent the information needs of this problem.
Convert the diagram into a set of relations.
For each field, indicate the data type you would use to represent that data.
Finally, create the database in MySQL.