

Lab and Homework session 1

The tasks of this lab can be done on your local XAMPP installation, or on the university's mysql server: csse-mysql.xjtlu.edu.cn

Username:

Your user account without the dot (.) For example, if my email account is:

Jianjun.Chen19@xjtlu.edu.cn

My username will then be:

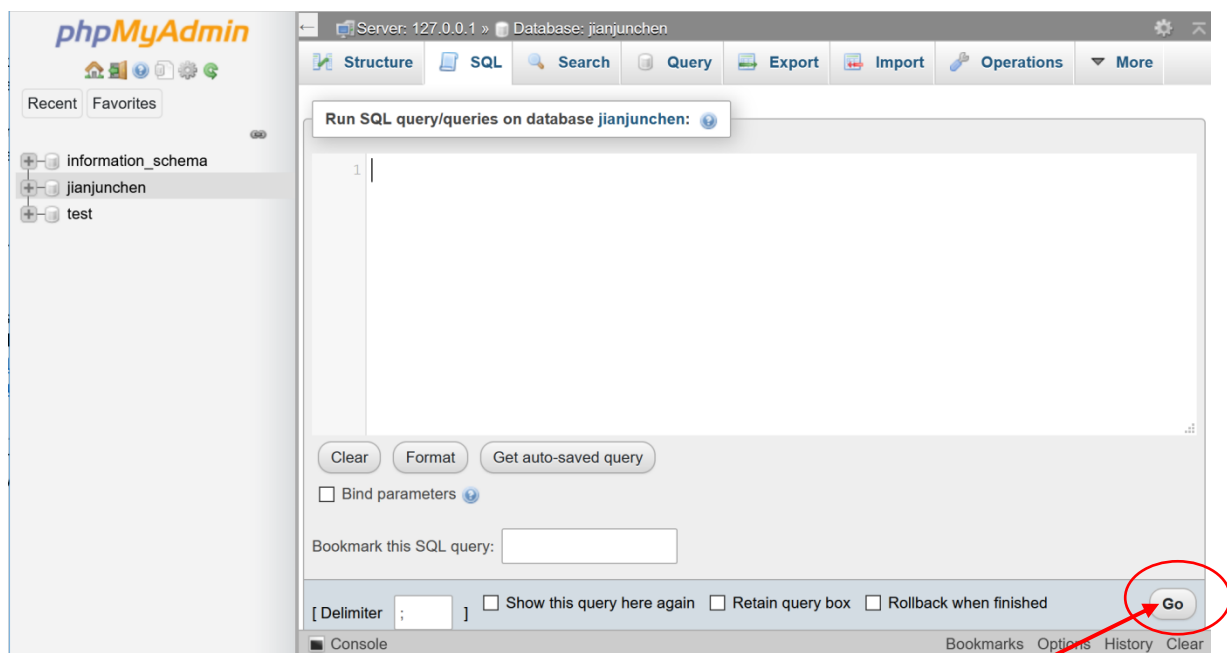
JianjunChen19

Password:

123

To access this server from outside of the campus, you need to connect to XJTLU VPN first.

NOTE: XAMPP and this server is almost the same. Your lab/coursework tasks can be done on any of these. Don't reset the database password to be null (""), because this database does not allow the account to login with the password null.



SQL queries need to be executed through this text field. Remember to click "Go" to start executing your SQL queries.

Task 1: Creating tables

Guided step 1: Write the SQL necessary to represent the following data requirements. A precise data type has not been specified for the fields in these tables. You should use your judgement and real world knowledge to help decide the most appropriate data type for each field. There's **no need to specify primary keys and foreign keys in your CREATE statement for now.**

Table Name: teachers		
Field Name	Data Description	Notes
ID	A unique ID number.	No precise format of ID
Name	A teacher's full name	All teachers have a name.
Telephone Number	An multi-format telephone number	Might contain country codes e.g. +44 123456789 Not all teachers have a tel.
Office	The room where the teacher's office resides.	Not unique. All teachers have an office. Teachers might share an office.

Table Name: modules		
Field Name	Data Description	Notes
Code	A unique module code	Up to 10 characters
Title	The module title	All modules must have a name.
Teacher	A reference to a teacher	All modules are taught by a single teacher.

Table Name: student		
Field Name	Data Description	Notes
ID	A unique ID number.	6 digit number
Name	A Student's full name	All students have a name.
Email	A standard email address	All students must have an email address.
Enrolment Modules	A list of modules that the student is studying.	A student may be studying zero, one or many subjects.

Possible issues at this step:

1. **“MySQL complains that my table already exists when running my CREATE TABLE statements”:** You tried to create these tables multiple times. As a result, some of the tables already exist in the database. You should delete existing tables before you create them again. Use DROP TABLE properly.
2. **The last item in CREATE TABLE does not end with comma.**
3. **Turn off your input method and use English input only.** (关掉输入法以防止中文标点符号误入代码)
4. **Table names are not case sensitive.** To add spaces to table names, enclose the table name with two ` symbols (E.g. `some table`). However, table names with spaces are not recommended, use some alternatives like `some_table`.
5. **Field names are only suggestions for the attribute names,** they don't need to be the same.

Guided step 2: Now add primary keys to these three tables using the ALTER keyword.

Task 2: Inserting some data into tables.

Guided step 3: Similar to step 2, use ALTER to add a foreign key to the `modules` table. So that its column `teacher` correctly references the `teacher` table. No need to consider `student` right now.

Guided step 4: Write the SQL necessary to insert the following data into your newly created tables.

Teachers and Modules

1. John Drake teaches Human Computer Interaction (HCI-101). His office is located in SD-766 and his telephone number is 12022017202020.
2. Felicia Gomez teaches Haskell for Beginners (HSB). Her office is located in BES-207 and her telephone number is 1024.
3. John Cartwright teaches Mathematics (MC1). His office is located in BES-201 and his number is: 12345 ext 1212.
4. Dave Moe teaches advanced Mathematics (MC2). His office is located in BES-205 and his phone number is: 65432 ext 2121.

Also remember to answer the following questions related to foreign key in this task:

1. Which table should you insert data into first, modules or teachers? Why?
2. What is the difference between having foreign keys and not having foreign keys?
3. What kind of problems can occur when foreign keys are not defined in the `modules` table?

Task 3: Redesigning the database

Guided step 5: Make sure you have added a primary key to the `student` table (on column ID). Now, add a foreign key to the attribute called `Enrolment modules`, so that it references the modules in the `modules` table.

Guided step 6: In the real life, a student can be enrolled in two modules. Can you assign modules “MC1” and “HSB” to a same student?

What errors will you get when you try to assign a second module?

What causes this error?

Once you have finished this part, scroll to the next page. (The next page shows the answer!)

Answer for step 6: The error in the previous question is caused by the primary key. If you remove the primary key, the error will disappear. However, this new table design with no primary key is **not a good design!**

Why? Consider the following table:

Table Name: Activity		
Field Name	Data Description	Notes
Activity name	The activity's name.	Some string values.
Student ID	A reference to a valid student ID	Must exist in the students table
Activity description	Descriptions	Some string values

Guided step 7: Create the table above, and try to add a foreign key for `student_id` so that it references the IDs in the `students` table. **With the primary key in the `students` table removed**, are you allowed to create this foreign key?

Once you have answered this part, scroll to the next page.

Answer for step 7: The referenced columns by foreign keys must be UNIQUE (or a primary key).

Without primary keys or unique keys, an attribute can have duplicated items. Database thus won't be able to tell which row to reference to. (Imagine two rows in the student table, all having the same ID 6501295)

The idea of having foreign keys is to tell databases to ensure correct data referencing between tables. Adding primary (Unique) keys to the referenced table's column ensures that this column can be referenced properly.

Open question 8: (Take this as homework) Can you redesign the student table (Not necessarily using one table, you can split it into two tables) so that:

1. Student and module enrolment information is correctly referenced using foreign keys. In a correct design, you can no longer have a student registered in a non-existing module and a module registered with a non-existing student.
2. Student ID can be referenced properly by the `activity` table.

The formal techniques to design/redesign databases will be taught in the lectures related to "Entity-relationship modelling" and "Normalisation". But you can try without these knowledge and see if you can work it out.

Task 4: (Homework) Reviewing the lecture content.

In the slides, we created the staff table and the branch table and linked them using foreign key. Now, create a new table called "StaffBranch" that is capable of storing the information of both tables.

- a) What will this combined table look like?
- b) Now, put the information of the staff table and the branch table into the new table you created.
- c) Once all information is recorded, carry out the following operations to both the original solution (staff table + branch table) and the new solution (staffbranch table):
 - a. The company want to relocate staffs in branch B005 to B002. Update the tables.
 - b. Change the branch office of "Mary Howe" to B002.
 - c. The branch office B007 is no longer needed, remove B007.
- d) Once you have done the three tasks above, answer:
 - a. What's the difference between the original solution and the new solution?
 - b. Which one do you think is a better design of the database? Why?

Good luck to all of you!