Database Homework and Lab Activity

Design Database Tables, Add Foreign Keys, Add Data, SQL Select Statements

CONTENTS

1. Overview

For the Lab Activity,

- You will use MySQL Workbench to create two database tables (web_user and user_role), add a foreign key, add data, then extract data. If you are unfamiliar with databases and/or MySQL workbench, there are resources for you (database overview pdf, MySQL Workbench Tutorial pdf, and video).
  - The design for tables web_user and user_role will be exactly the same for all students (so that sample code will work for you) but the data will be different for each student.
- There is nothing to publish. You will submit a word document with screen captures (that shows your work) to Canvas.

For the Homework,

- Using MySQL Workbench, you'll create your own individual database table (which I call your "other" table), add a foreign key, add data, then extract data.
- You'll add a database blog to your website and link from there to a word document (that shows all of your database work, all three tables). As normal, you'll submit a zip file of your projec to Canvas and you'll publish/test.

2. Document Your Work

Create a word document with the following sections to show your database work. Please capitalize these headings so we can find our way around your document. You'll understand what to put in these sections after you have gone through this document and the MySQL Workbench Tutorial.

- ***For the Lab Activity, you'll submit ONLY the items related to web_user and user_role tables (indicated with gold font).***
- For the ***Homework***, you'll submit ***everything*** below – so the Lab Activity document is a subset of the Homework document.

Here are the headings for you to add into your document – and explanation of what goes into each heading.

1. **NAME:** Your name
2. **WEB SITE:** Your Web Site title, URL to your home page, copy/paste what you currently have in your home content, where you describe the functionality of your web site (and entice users to visit your web site).
3. **DATABASE:** In the next sections, you will be asked to copy/paste screens (into this word document) from MySQL Workbench – to show that you have done the prescribed database work.

   TABLE DESIGNS

   - **USER_ROLE DESIGN:** user_role table design (right click in MySQL WB and select Alter Table)
   - **WEB_USER DESIGN:** web_user table_design plus screen capture of "foreign keys" tab (showing FK key to user_role)
   - **"OTHER" DESIGN:** "other" table design plus screen capture of "foreign keys" tab (showing FK to web_user)

   DATA ENTRY ERRORS

   - Duplicate user_role PK in user_role record
   - Invalid user_role FK in web_user record
   - error trying to delete a user_role record that's referenced by a web _user record
   - Invalid web_user FK in "other" record
   - error trying to delete a web_user record that's referenced by an "other" record

   SELECT STATEMENTS

   - **USER_ROLE:** all rows, all columns, sorted by user_role_id.
   - **WEB_USER:** all rows, all columns, sorted by web_user_id.
   - **WEB_USER JOINED WITH USER_ROLE:** all rows, certain columns, sorted.
   - **"OTHER":** all rows, all columns, sorted by the id of your "other" table.
   - **"OTHER" JOINED WITH WEB_USER** all rows, certain columns, sorted.
   - **"OTHER" JOINED WITH WEB_USER** some rows, certain columns, sorted.

To get a screen capture, click on Alt-PrtSc (copies active window into the clipboard), then paste into an image editor like MSPaint, then copy out just the part you want to show and paste that into the word document. If you paste from Alt-PrtSc directly into the word document, the screen capture shows too much, making things too small for us to read.

3. web_user and user_role Tables

Following the MySQL Workbench Tutorial,

- ***design tables web_user and user_role exactly as shown below.*** If it your tables are not designed exactly as shown, the sample code provided in this class will not work against your database.

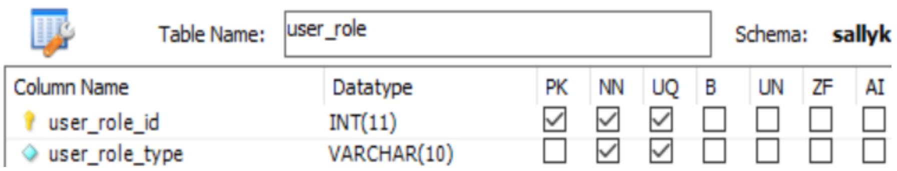


Table Name: user_role Schema: **sallyk**

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI
user_role_id	INT(11)	☑	☑	☑	□	□	□	□
user_role_type	VARCHAR(10)	□	☑	☑	□	□	□	□

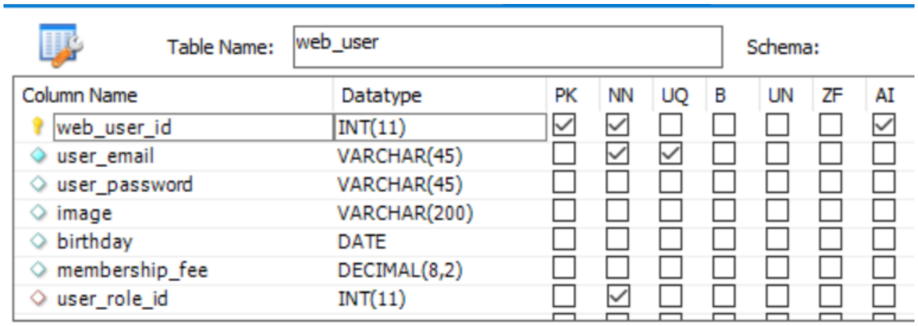
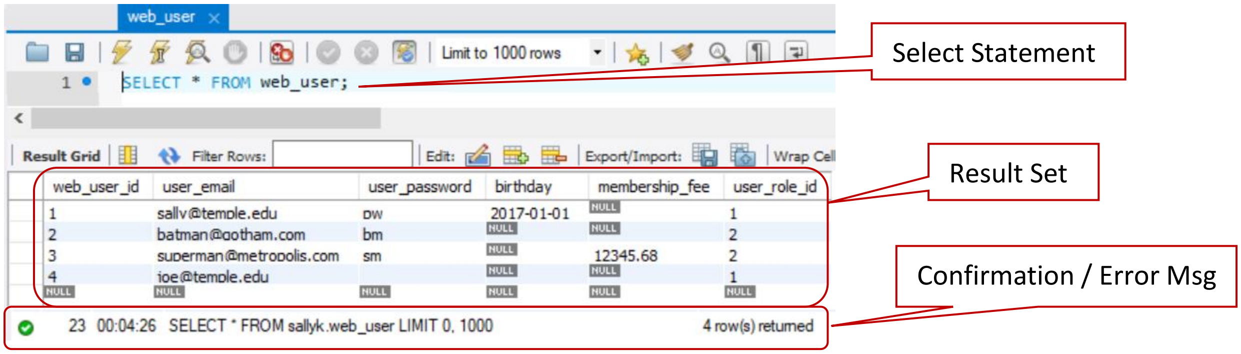


Table Name: web_user Schema:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI
web_user_id	INT(11)	☑	☑	□	□	□	□	☑
user_email	VARCHAR(45)	□	☑	☑	□	□	□	□
user_password	VARCHAR(45)	□	□	□	□	□	□	□
image	VARCHAR(200)	□	□	□	□	□	□	□
birthday	DATE	□	□	□	□	□	□	□
membership_fee	DECIMAL(8,2)	□	□	□	□	□	□	□
user_role_id	INT(11)	□	☑	□	□	□	□	□

- add a foreign key from *web_user.user_role_id* to *user_role.user_role.id.*
- Add 2-3 records into your user_role table.
  - Since web_user references user_role, you must enter records into user_role first.
  - Try to add a record with a duplicate primary key and notice that the database management system will not let that record be inserted. Screen capture the Duplicate user_role PK error.
- Add 5-7 records into your web_user table.
  - At least one of these records shall have null for all nullable non character fields. At least one of these records shall have all of its fields populated.
  - Since this table has an auto-increment primary key, you do not provide web_user_id when you insert – let the database management system do that for you.
  - Try to add a record that has an invalid (non-existent) user_role_id and notice that the database management system will not let that record be inserted. Screen capture the Invalid FK from web_user to user_role.
  - Then try to delete a user_role record that has been referenced by a web_user record. Notice the DBMS will not allow you to do that either. It is the "job" of the DBMS to maintain the integrity of the DB at all times, allowing no invalid foreign key references. Screen capture the error – cannot delete a referenced user_role record.
- A select statement showing your user_role data (all rows, all columns, SELECT * is OK), sorted by user_role_id.
- A select statement showing your web_user data (all rows, all columns, SELECT * is OK), sorted by web_user_id.
- A select statement of the web_user joined with user_role table (all rows), showing just the user role type and email address columns (sorted by user role type, then email address). Note: there should be as many rows in this result set as there were in the above result set.

When you get a screen capture for a SELECT statement, capture the following areas of the screen:



```
SELECT * FROM web_user;
```

web_user_id	user_email	user_password	birthday	membership_fee	user_role_id
1	sally@temple.edu	pw	2017-01-01	NULL	1
2	batman@gotham.com	bm	NULL	NULL	2
3	superman@metropolis.com	sm	NULL	12345.68	2
4	joe@temple.edu		NULL	NULL	1
NULL	NULL	NULL	NULL	NULL	NULL

```
23  00:04:26  SELECT * FROM sallyk.web_user LIMIT 0, 1000          4 row(s) returned
```

- **Note: stop here for your Lab Activity. Save the word document (or rtf) as a PDF (easier to read, no spelling errors) then submit both documents (doc/rtf & pdf) to Canvas. There is nothing to publish.**
- **To complete the Homework, continue on… If you need help, refer to the database resources (database overview pdf, MySQL Workbench Tutorial pdf, video).**

4. Design Your “other” Table

Then **create your own unique table** (which I will refer to as your “other” table but name this table according to what data it will hold). In a previous homework, you already have indicated what you expect to have in your “other” database table. You can keep with what you already said, or change it now, but make sure the web site functionality you describe in your home page matches your database design. The following information was listed in the Home Page Homework, suggesting ideas for database design that would work given the database restrictions:

- Users are chefs, other table is recipe (one chef can contribute many recipes)
- Users are travelers, other table is places visited (one traveler can visit many places).
- Users are salesmen, other table is products (one salesman can sell many products).
- Users (or their secretaries) are football players, the other table is highlights of their career.
- Users (or their secretaries) are musicians, the other table is songs or albums they have created.
- PLUS MANY MORE IDEAS … Perhaps you are starting to see a pattern of what can be done with the “shape” of database design that you are being asked to implement.

For your “other” table, do not choose any type of “people” (customers, players, etc) because you can use the web_user table for that purpose. Try to fit your idea to the database restrictions, not the other way around. Try to keep things realistic but simple.

Here are requirements for the fields in your “other” database table:

- Id: primary key, auto-increment, name this field with table name followed by “_id”)
- Name or some kind of character identifier (require it to be unique, click on “UQ”)
- image URL – a long varchar - perhaps varchar(200) since fully qualified URLs can be pretty long
- at least two nullable (user optional) non-character fields, e.g., integer, decimal (for money), date.
    - Note: having a “date_entered” field does not meet the requirement of “user optional” non-character field because we should not ask the user to tell us what date they entered the data (the system should know and provide it – if we wanted that). So, come up with something else.
- any other fields you like (but not too many because it will make extra work for you).
- “web_user_id”, a foreign key that points to the user who contributed this record (required “NN”).

Don’t use any SQL keywords as table names or field names. Otherwise, your java code will have errors trying to run various SQL commands against your table. (Google to see list of SQL keywords, but you here are some examples that you cannot use: ***user, role, password, state, date***.)

a) Add Foreign Key from Your “Other” Database Table to web_user

Add a foreign key from your “other” database table to table web_user. If you are unsure how to do this, read the database introduction PDF, listen to the database video, re-read the MySQL Workbench tutorial.

b) Add “other” Data

- Add about 10 ***realistic looking*** records into your “other” table.
    - At least one of these records shall have null for all nullable non character fields.
    - At least one of these records shall have all of its fields populated.
    - Try to add an “other” record with an invalid FK to web_user. Screen capture the Invalid FK from “other” to web_user.
    - Try to delete a web_user record that’s being referenced by an “other” record. Screen capture the error cannot delete a referenced web_user record.

c) Run SQL Select Statements

Execute each of the following SELECT statements (in this order) and paste the following screen captures. ***Each screen capture should show the Select Statement and the result set and the error/confirmation message like this:***

a. A select statement showing the data from your “other” table (all rows, all columns, SELECT * is OK), sorted by the id of your “other” table.

b. A select statement of your “other” table joined with web_user (all rows), showing just the descriptive field from your “other” table and then the user email address from web_user. To have a primary then secondary sort, you just specify two columns after the ORDER BY (e.g., ORDER BY col1, col2). Note: there should be as many rows in this result set as there were in the above result set.

c. A select statement of your “other” table joined with web_user (but just some rows). It would be the same select statement as above, but filtering out some of the rows by adding additional criteria to your WHERE clause.

5. Submission Requirements

At this point, for your homework, you should have a **word document (or other rich text editor)** with all of its sections completed. Save your document (naming it like yourLastName_database.docx) either into your Web Pages folder or into your pics folder. (Save the original format in case you need to edit it.) Save your document also as PDF for publishing.

BLOG: Add a blog entry to your blog content (with "Database" as blog heading"). Include the following:

- A paragraph talking about your database experience.
- A paragraph that includes a link to your database PDF and says what parts of this homework you found hard/easy/valuable. Here's how you'd link to that (if you stored the PDF directly under the Web Pages folder). Target="blank" will open the pdf in a new tab.

  Click <a target="_blank" href='smith_database.pdf'>here</a> to see my database document.

Finally,

- Test your web site locally (e.g., the link from your blog to the database PDF). **Publish your web site** and test the link again.
- Then, submit a zip file of your whole project into the Canvas assignment for this week.
- Into the text area of your Canvas submission, **perform a self assessment.** This consists of you grading your own homework, using the list of deductions from the end of this document. Enter one line per deduction (numeric deduction first, followed by deduction explanation). At the top, type in the total grade you expect to receive for the HW. If you wish, you and a friend/classmate can grade each other's HW. And, of course, you can fix whatever's broken and remove the related deduction as long as it's before the due date.

6. Example Deductions for Lab Activity

- up to -5: missing/incorrect table designs for user_role and web_user
  - -0.5: user_role PK is auto-increment (and it is not supposed to be)
  - -0.5: web_user PK is not auto-increment (and it is supposed to be)
  - -0.5 each: birthday and/or membership fee not optional (should be optional, NN clicked)
  - -1: web_user FK to user_role - tab not shown and/or incorrect
- Data Entry Errors:
  - -0.5 for each missing screen capture (Duplicate user_role PK, Invalid FK from web_user to user_role, Cannot delete a user_role record that's referenced by a web_user)
- up to -0.5: user_role data insufficient and/or unrealistic.
- up to -1: web_user data insufficient and/or unrealistic.
- -1: for not having at least one web_user row with null for membership fee.
- -1: for not having at least one web_user row with null for birthday.
- -1: not having at least one web_user record with all fields populated.
- up to -3: SQL join statement (joining web_user with user_role) is not correct.

7. Example Deductions For Homework

- **-9 for Not Publishing:** We perform functional testing each week based on what you have published. You need to test locally, publish, then test what you published.
- **-9 for Lack of Canvas Submission:** If there is ever any question about a grade, we go by the code that was uploaded into Canvas at the time the assignment was due.
- **Up to -2 for Missing/Insufficient Blog Entry** (your db experience, link to the DB document/pdf, what was hard/easy/valuable about this assignment).
- **Up to -2 for Missing/Insufficient Self Assessment** (where you list the deductions you think you'll get).
- **Up to -3: web_user and user_role table design.** The web_user and user_role table designs must exactly match what was prescribed (or sample code will not work for you all semester).
  - **-1: missing FK tab** showing web_user FK to user_role.
  - **Up to -1.5 for not providing Data Entry error messages** (0.5 each: Duplicate user_role PK, Invalid FK from web_user to user_role, Cannot delete a user_role record that's referenced by a web_user). If you cannot produce these error messages, you probably did not design the tables properly.
- **-5: if your "other" database table holds people (like web_user does) – choose something else.**
- **Up to -5 "other" table design.**
  - **Up to -3 if your "other" table does not have two nullable (user optional) non character fields**. Why? Because "real data" often has null in it and your code needs to be able to deal with null data. A field like insert/update date does not meet this requirement (not user optional) since you would never ask a user to enter today's date when the system already knows that information.
  - **Up to -2:** poor naming of table and/or field names (must be realistic and representative).
  - **Up to -2 if you have used SQL keywords** as table or field names (examples not to use: password, role, state – google "SQL keywords").
  - **Up to -2 for not providing Data Entry error messages** (1 each: Invalid FK from "other" to web_user, Cannot delete a web_user referenced by an "other" record). If you cannot produce these error messages, you probably did not design the tables properly.
- **Up to -4 if your database design does not align with the functionality you say your web app will provide.** Modify one or the other, but make sure your database can deliver the functionality that you are promising in your home page content. (Your final web app functionality will be very simple – users will be able to log on, insert/edit/delete records from two database tables: web_user and your "other" database table).
- **Up to -2 for non compliant data.** You were asked to include at least one record (in web_user, in "other") that has all fields populated andat least one record (in web_user, in "other") that has null for all nullable fields.
- **Up to -2 for Non-Realistic Data:** You were asked to enter realistic data so that your web application looks good when it begins to display data on its pages.
- **Up to -3 for Problems with SELECT Statements** as prescribed in the database document.
- **Up to -2 for Unprofessional Home Page Content:** The "marketing material" from your home page should be of a quality that acceptable by a "real company" (small or large) that might be paying you to create their web site.
- **Up to -8 for Lack of Originality of Project and/or Data Model and/or Data:** As in most homework assignments, points will be deducted if your submission is too similar to the sample(s) provided or to another student.