Contents

1. Overview .............................................................................................................................................. 2
2. Before You Start ..................................................................................................................................... 2
3. Database Requirements ......................................................................................................................... 3
   a) Create Tables .................................................................................................................................... 3
   b) Add Data .......................................................................................................................................... 4
   c) Run SQL Select Statements .............................................................................................................. 5
4. Submission Requirements ...................................................................................................................... 6
5. Sample Deductions (Grading) ................................................................................................................ 7
1. Overview

In this homework, using MySQL Workbench, you will:

1. Create database tables:
   - web_user and user_role (exactly as prescribed) and
   - an “other” table named and designed as you propose (meeting the requirements of this document).
2. Populate all three tables with test data.
3. Generate several SQL select statements to extract data from the database.
4. Create a word document that shows your database work.

2. Before You Start

Before starting on this Homework, make sure that you have installed MySQL Workbench and tested that you can connect to your database (if you added this course late, you may have to email your instructor to request a database be created for you ASAP). From the course web page (for this assignment) you’ll find links to the following documents:

- How to install MySQL Workbench
- How to connect to your Temple database (shows you how to find your database credentials, etc)
- MySQL Workbench Tutorial: How to create database tables and relationships (primary keys, foreign keys), enter data, and write SELECT statements

Create a word document to hold this week’s homework submission. Into this document put the following sections (please make the capitalized words into headings so we can find our way around your document):

1. NAME: Your name
2. WEB SITE: Your Web Site title and URL to your home page.
3. FUNCTIONALITY: Copy/paste from 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).
   - It is OK to change your mind (from last week’s homework) about the functionality that your web application will provide – as long as your database table meets the design requirements of this homework AND the web functionality you describe on your home page can be supported by your database design.
4. 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. Look for red font.
   - 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. Database Requirements

a) Create Tables

*Design tables web_user and user_role exactly as shown below* (otherwise sample code will not work against your database). Then add a foreign key from `web_user.user_role_id` to `user_role.user_role.id`. The MySQL Workbench tutorial (linked from this week’s homework) tells you how to do this.

Then *create your own unique table* (which I will refer to as your “other” table but you must name this table according to what data it will hold). Name this table and design its fields according to what you proposed in your last homework. Make sure that what you propose for your database table aligns with what you said your website will allow users to do. Also be sure to meet these additional requirements:

- 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`.)

Add heading “TABLE DESIGNS” into your word doc, then paste the following screen captures (in this order):

- `user_role table design` (right click the table, select “alter table”, click on the “columns” tab).
- `web_user table design` (showing foreign key to user_role)
- “other” table design plus screen capture of “foreign keys” tab (showing reference to web_user).
b) Add Data

Add realistic looking data into your three database tables.

- Add 3-4 records into your user_role table.
  - Since your web_user table references data from 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 – you 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.

- Add about 10 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 your user_role data (all rows, all columns, SELECT * is OK), sorted by user_role_id.

b. A select statement showing your web_user data (all rows, all columns, SELECT * is OK), sorted by web_user_id.

c. A select statement of the web_user joined with user_role table (all rows), showing just the user role name and email address columns (sorted by user role name, then email address). Note: there should be as many rows in this result set as there were in the above result set.

d. 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.

e. A select statement of your “other” table joined with web_user (all rows), showing just the descriptive field from your “other” table and the user email address from web_user (sort the records by the columns show, compound sort order). Note: there should be as many rows in this result set as there were in the above result set.

f. 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.

If you need help figuring out how to generate any of the above SELECT statements, refer to the MySQL tutorials mentioned at the start of this document.
4. Submission Requirements

Your word document should now contain the following headings (capitalized words). Then add in the rest (screen captures, etc).

NAME: Your name

WEB SITE: Your Web Site title and URL to your home page.

FUNCTIONALITY: Copy/paste from 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).

DATABASE:

TABLE DESIGNS

• USER_ROLE DESIGN: user_role table design
• 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, Invalid FK (web_user to user_role), error trying to delete a referenced user_role record
• Invalid FK (“other” to web_user), error trying to delete a referenced web_user 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.

Save your document (naming it like your LastName_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.

### 5. Sample Deductions (Grading)

- **-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 and at 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.