Assignment3

Part 1

Description: Frequently, not only will you be asked to design and develop Web applications, but you will also be asked to create (design) database solutions that interact with the Web application—and, in fact, the data repository is the *core* of all Web applications. Hence, the following business requirements.

A pet store owner, who owns a number of pet stores, requests that you develop a Web application whereby he and his team can record, track, and maintain relevant company data, based upon the following business rules:

1. A customer can buy many pets, but each pet, if purchased, is purchased by only one customer.
2. A store has many pets, but each pet is sold by only one store.

Remember: an organization's business rules are the key to a well-designed database.

For the Pet's R-Us business, it's important to ask the following questions to get a better idea of how the database and Web application should work together:

- Can a customer exist without a pet? Seems reasonable. Yes. (optional)
- Can a pet exist without a customer? Again, yes. (optional)
- Can a pet store not have any pets? It wouldn't be a pet store. (mandatory)
- Can a pet exist without a pet store? Not in this design. (mandatory)

NB: Unless otherwise noted, *ALL* ERDs must be created and submitted in **MySQL Workbench format** (.mwb), and include the following criteria:

- entities (include color)
- connectivities
- cardinalities
- relationship strengths (use appropriate drawings of lines between entities)
- relationship participations (optional/mandatory)
- PKs, FKs, PFs (if used)
- Attributes (define *suitable* attributes):

Mandatory attributes for entities such as people, places, or organizations:

- first name
- last name
- street
- city
- state
- zip (**char 9 – only numbers**)
- phone (**bigint unsigned**)
- email
- url

NB: *Be sure* to change default FK options to On Update Cascade, On Delete Cascade —*and* understand the ramifications of these changes!

Notes:

1. ***All*** tables must have notes attribute.
2. Include *suitable* data in the MWB file, at least **10** "unique" records per table, ****must forward-engineer locally****!
3. *Be sure* to match data types.

Additional attributes:

store:
ytd_sales
url

customer:
balance, total_sales

pet:
type, sex, cost, price, age, color, sale_date, vaccine, neuter

Data:
Each table ***must*** include at least **10** records, ***and* be able to forward-engineer locally**.

Note: after attempting to forward-engineer the ERD data design with data, double-click on the connection from the main page in MWB, then issue the following commands:

```
use database_name;
```
(i.e., your fsuid)
```
show tables;
```
```
select * from table_name;
```
(for each table)

When grading, the tables are checked for:
1) being created, and
2) having data (min. **10** records per table).

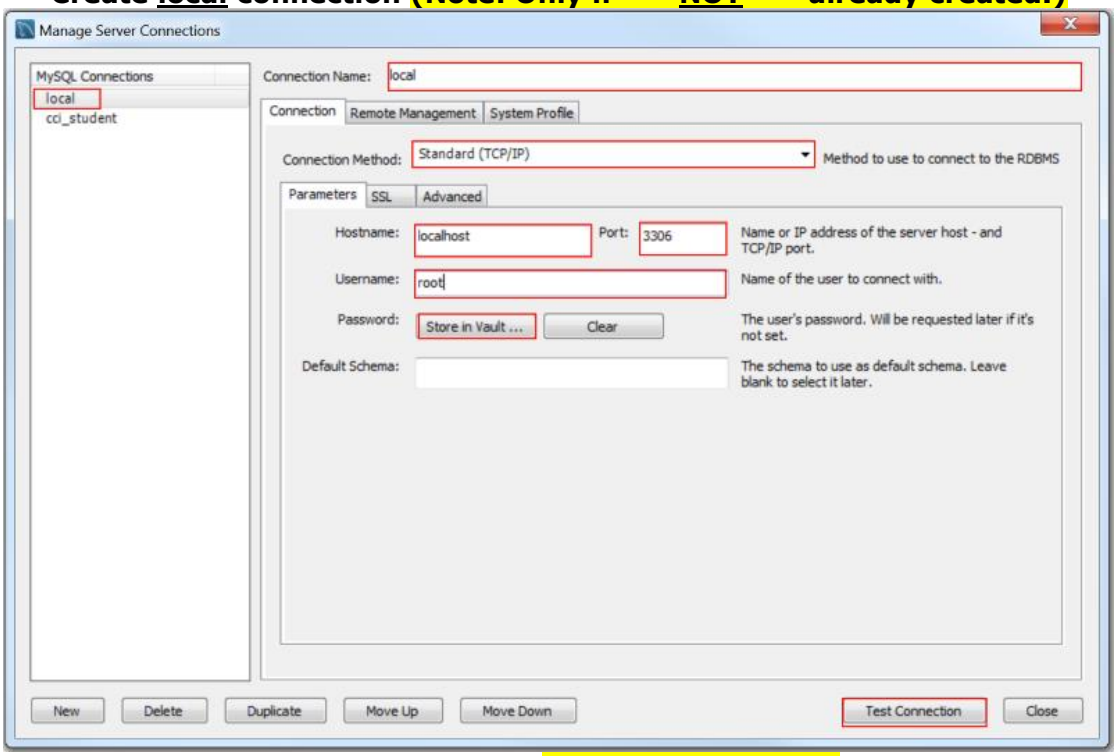
Note (download MWB, ***only*** if not currently installed)**:**

1. Download latest **MySQL Workbench** case tool: http://dev.mysql.com/downloads/workbench/
2. Save **a3.sql** file:
    a. Create tables and include data
    b. Export file: File > Export > Forward Engineer SQL CREATE SCRIPT

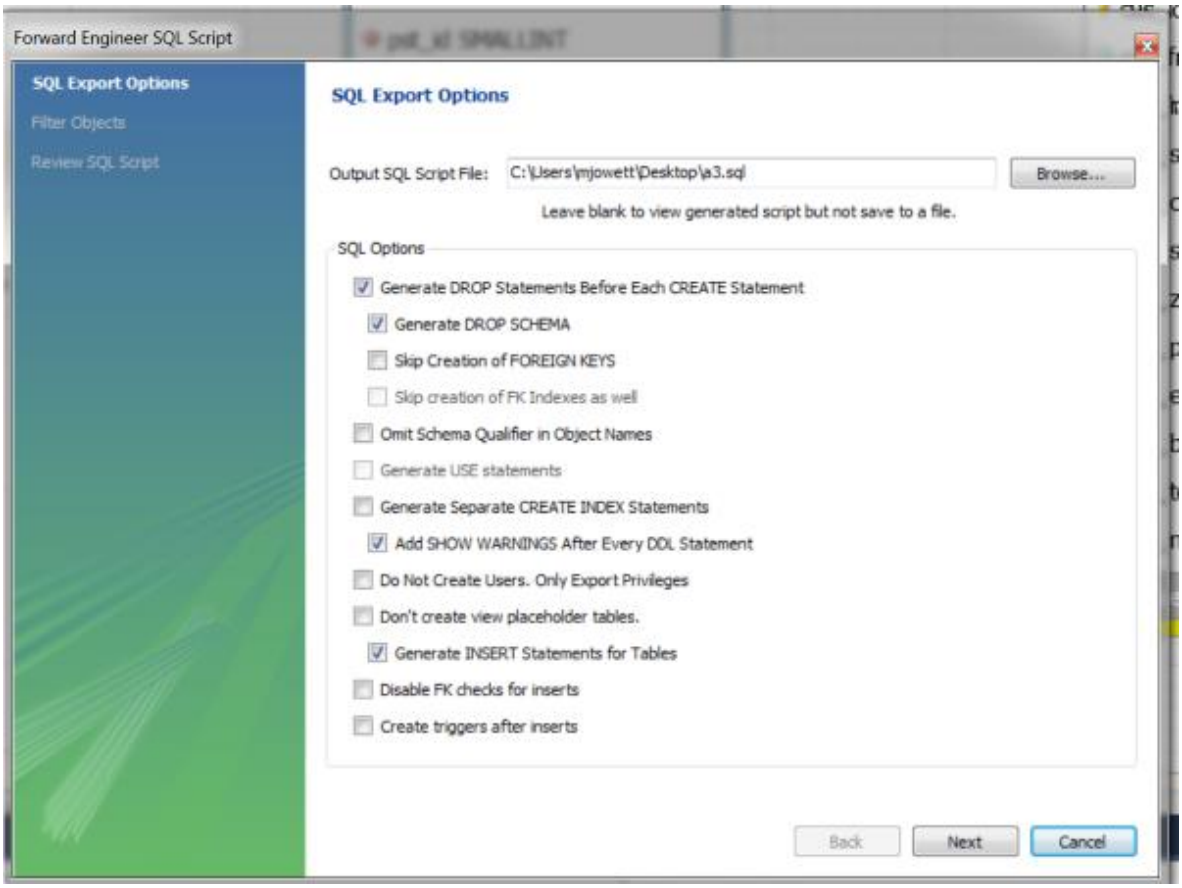
1. Database Requirement: http://www.qcitr.com/vids/LIS4381_A3_Database_Requirements.mp4
MySQL Workbench login below: Username: **root** and Password: **mysql**
Notes:

a) The video serves both courses: LIS4381 and LIS4368.
b) The tables ***must*** forward-engineer ***locally***--**not** to cci_student (as in the video). (See below.)
c) These are older videos. Do ***not*** create repos for individual subdirectories—that is, do ***not*** create an a3 repo. Just push ***all*** your files from your lis4368 local repo, as was done for a1 and a2 (i.e., a3 subdirectory should reside inside lis4368 as well).

Create local connection (Note: Only if ***NOT* already created!)**



Note: .mwb file **and** SQL script should be placed in **lis4368 > a3 > docs** subdirectory.
Export file: File > Export > Forward Engineer SQL CREATE SCRIPT



Part 2

README.md file should include the following items:

1. **Course title, your name, assignment requirements, as per A1;**
2. Screenshot of ERD;
3. Screenshots of 10 records for **each** table—use *select * from each table*;
4. Screenshot of a3/index.jsp
5. Links to the following files:
    a. a3.mwb
    b. a3.sql

Deliverables:

1. Provide **Bitbucket** read-only access to **lis4368** repo, include links to the repos you created in the above tutorials in **README.md**, using Markdown syntax (**README.md** must also include links and screenshot(s) as per above.)
2. **Blackboard Links**: lis4368 **Bitbucket** repo

Markdown syntax for screenshot and links:

```
#### Assignment Screenshot and Links:

*Screenshot A3 ERD*:

![A3 ERD](img/a3.png "ERD based upon A3 Requirements")

*A3 docs: a3.mwb and a3.sql*:

[A3 MWB File](docs/a3.mwb "A3 ERD in .mwb format")

[A3 SQL File](docs/a3.sql "A3 SQL Script")
```

lis4368 Bitbucket repo link to a3 subdirectory screenshot:

> **NOTE:** This README.md file should be placed at the **root of each of your repos directories.**
>
> Also, this file **must** use Markdown syntax, and provide project documentation as per below--otherwise, points **will** be deducted.

LIS4368 - Advanced Web Applications Development

Mark K. Jowett, Ph.D.

Assignment 3 Requirements:

Deliverables:

1. Entity Relationship Diagram (ERD)
2. Include data (at least 10 records each table)
3. Provide Bitbucket read-only access to [redacted] repo (Language SQL), *must* include README.md, using Markdown syntax, and include links to *all* of the following files (from README.md):
    - docs folder: a3.mwb, and a3.sql
    - img folder: a3.png (export a3.mwb file as a3.png)
    - README.md (*MUST* display a3.png ERD)
4. Blackboard Links: [redacted] Bitbucket rep

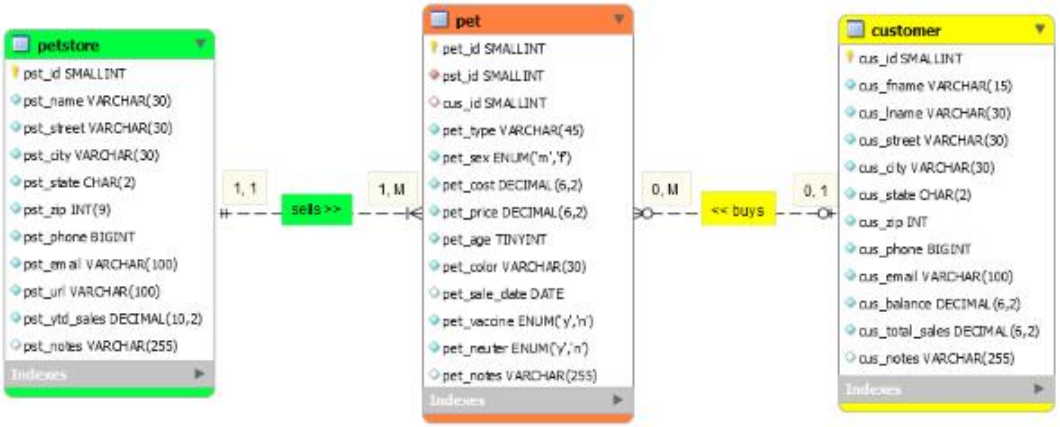
README.md file should include the following items:

- Screenshot of [redacted] ERD that links to the image:

> This is a blockquote.
>
> This is the second paragraph in the blockquote.

Assignment Screenshot and Links:

Screenshot A3 ERD:



Business Rules:
a. A customer can buy many pets, but each pet is purchased by only one
b. A store has many pets, but each pet is sold by only one store

Remember: the business rules are the key to a well-designed database.
It's important to ask yourself questions like, in the Pet's R-Us DB...
* Can a customer exist without a pet? Seems reasonable. Yes. (optional)
* Can a pet exist without a customer? Again, yes. (optional)
* Can a pet store not have any pets? Possible, but highly unlikely. (mandatory)
* Can a pet exist without a pet store? No. (mandatory)

The following criteria are required for the submitted ERD: Write all appropriate
- connectivities (1:M, 1:1 use Crow's Foot diagram)
- cardinalities
- relationship strengths (use appropriate drawings of lines between
- relationship participations (optional/mandatory)
- PKs, FKs, PFs (if used)
- Attributes (define suitable attributes)
- relationships between entities (use of verbs) in the ERD

A3 docs: a3.mwb and a3.sql:

A3 MWB File

A3 SQL File

Part 3

Answer the following questions (Chs. 7, 8):

1. A servlet can retrieve the current session object by using the getSession method of the

a. HttpSession class
b. response object
c. the request object
d. the ServletContext object

2. A session ends when

a. when a specified amount of time elapses without another request
b. when the user exits the browser
c. both a and b
d. none of the above

3. A session object is created

a. every time a browser makes a request for a web page
b. every time a browser requests a servlet
c. every time a browser requests a jsp
d. the first time a browser makes a request to a site

4. Cookies are stored

a. in a database
b. on the web server that sent the cookie
c. on the web server that created the cookie
d. on the client

5. If you use the MVC pattern with EL,

a. the servlets create the JavaBeans and set the bean properties
b. the servlets create the JavaBeans and the JSPs set the bean properties
c. the JSPs create the JavaBeans and set the bean properties
d. the JSPs create the JavaBeans and the servlets set the bean properties

6. Persistent cookies

a. are removed after the session ends
b. exist after the session ends
c. can't be deleted
d. are often called hidden fields

7. The JSP Expression Language (EL) provides a compact syntax that lets you get data from

a. JavaBeans
b. maps and arrays
c. lists that have been stored as attributes of a web application
d. all of the above

8. The advantages of using hidden fields over URL rewriting to pass parameters from a browser to the server are

a. a hidden field can contain spaces and other special characters that are difficult to work with when using URL rewriting
b. the user can see the parameters by using the browser to look at the page's source code
c. no limit to the number of characters that can be stored in a hidden field
d. both a and b
e. both a and c

9. The cookie named mediaCookie in the following statement can be accessed by

```
mediaCookie.setPath("/");
```

a. only the current directory
b. the entire web application
c. the entire web server
d. anyone

10. The current session object is available

a. only to JSPs
b. only to servlets
c. to both servlets and JSPs
d. to neither servlets nor JSPs, as they are not part of the API.

The following link demonstrates how to use

```
<a href="schedule?user_id=${user.id}" >
   View your schedule
</a>
```

a. URL rewriting
b. URL encoding
c. hidden fields
d. cookies

11. The initParam implicit object lets you use EL

a. to get the value of a context initialization parameter
b. to get the value of a servlet initialization parameter
c. to set the starting value of a context initialization parameter
d. to set the starting value of a servlet initialization parameter

12. The paramValues implicit object lets you use EL

a. to get the values of request parameters
b. to get the values of response parameters
c. to set the values of request parameters
d. to set the values of response parameters

13. The types of scope that can be used with EL are

a. request, response, session, and application
b. page, request, session, and application
c. page, response, session, and application
d. request, response, and session

14. To create a persistent cookie, set the cookie's maximum age to

a. 0
b. -1
c. a positive number
d. none of the above

15. To delete a persistent cookie, set the age of a cookie to

a. -1
b. 1
c. 0
d. none of the above

16. Typical uses for cookies are

a. to allow users to skip login and registration forms
b. to customize pages that display info like weather reports, sports scores, etc.
c. to focus advertising like banner ads that target the user's interests
d. all of the above

17. When you set an attribute in the request object, the attributes are _____.

a. available until the user closes the browser
b. removed when the request has been completed
c. available until the session times out
d. none of the above

18. When you set an attribute in the session object, the attributes are

a. available until the user closes the browser
b. available until the session times out
c. available until you use the removeAttribute method to remove an attribute from the session object
d. all of the above

19. Which of the following is the correct EL code for getting the value of a cookie named userPreference?

a. ${cookie.userPreference}
b. ${cookie.userPreference.value}
c. ${cookie.value.userPreference}
d. ${cookie.userPreference[value]}

20. Which statement is true?

a. HTTP doesn't maintain state
b. HTTP maintains state
c. FTP doesn't maintain state
d. all of the above

21. You can use the implicit EL objects to work with

a. request parameters and HTTP headers
b. cookies and context initialization parameters
c. the implicit pageContext object that's available to all JSPs
d. all of the above

SQL Statements

1. List only the pet store IDs, full address, and phone number for all of the pet stores.
2. Display the pet store name, along with the number of pets each pet store has.
3. List each pet store ID, along with all of the customer first, last names and balances associated with each pet store.
4. Update the customer last name to 'Valens' for Customer #2.
5. Remove Pet #4.
6. Add two more customers.