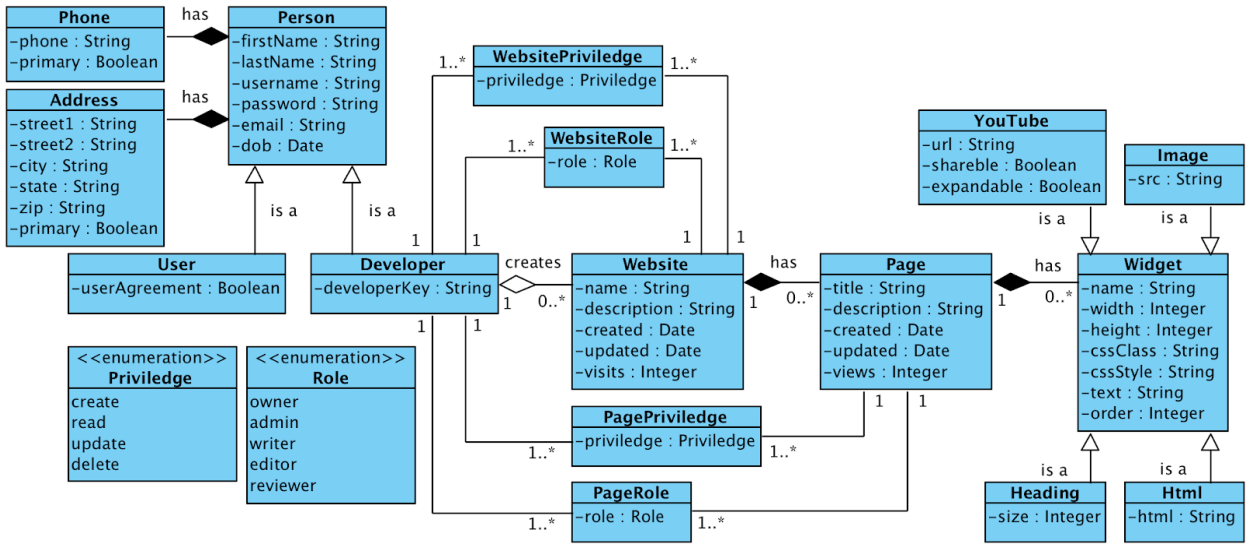
Relational Model Assignment

Introduction to Relational Databases

Consider the UML class diagram shown below. Create the corresponding SQL that implements the equivalent relational model, fulfills the use cases, implements the relations, and enforces the constraints. A link to this assignment can be found here.

UML Class Diagram



Choice of Remote Databases:

In this assignment you can choose to go with (AWS) **or** (Heroku and Local Database) depending on your choice.

Heroku

If you are going with heroku,

1. Create a local database for triggers as triggers are not allowed in heroku.
2. The schema name cannot be changed due to restricted user privileges.

   How to set up a local database

AWS:

1. You might run into issues with triggers, while trying them on AWS. If so, please try the following:

a. https://techtavern.wordpress.com/2013/06/17/mysql-triggers-and-amazon-rds/
b. https://www.youtube.com/watch?v=MJ8SUSqcwqQ

Creating a remote database on Heroku:

At this point you should have a working local and remote development environment. If not, make sure to follow the instructions for setting up a remote account, remote application, and a development environment as described in Setting up a Development Environment lecture. Once the environment is setup, create and deploy a Spring Boot application on Heroku as described in Deploying Spring Boot Web Applications to Heroku. Finally, add a remote MySQL database to the remote development environment as described in Adding a Remote MySQL Database to a Spring Boot Web Application on Heroku. The general steps for setting up the environment on Heroku are listed below. Refer to the original documents for more details.

1. Install JDK 8 or later
2. Install Apache Maven
3. Install the Spring Boot framework
4. Install MySQL Workbench or some other MySQL client
5. Create an account on Heroku
6. Install the Heroku CLI
7. Create a simple Spring Boot Web application, e.g.,
   spring init --dependencies=web myapp

8. Deploy the Spring Boot application to Heroku, e.g.,
   heroku create

9. Add a MySQL remote database to the remote Spring Boot application on Heroku
10. Connect your local MySQL Workbench to the remote MySQL on Heroku

Create remote database on AWS:

If you chose to use AWS for your assignment, this section describes creating a remote MySQL database instance running on AWS.

- Login to the Amazon AWS console and expand *All Services*.
- Under the *Database* section, select *RDS*.
- In the *Amazon RDS* landing page, select *Launch* or *Get Started Now* to add a new RDS instance. If you are on your dashboard, you can choose ***Create Database*** instead.
- In the *Select engine* screen, select *MySQL* and then click *Next*.
- In the *Choose use case* screen, select *Free Tier - MySQL* and then click *Next*.
- In the *Specify DB details* screen, keep the default settings, and choose the following configuration and then click *Next*. Use your own identifier, username, and password. The

usernames, names, and identifiers shown in this document are based on a particular course, e.g., `cs1234`, semester, e.g., `fall2345`, and your last name, e.g., `<your last name>`. Please use your particular values where applicable.

- DB instance class: `db.t2.micro`
- DB instance identifier: `cs1234-fall2345-<your last name>`
- Master username: `<choose a username easier to remember>`
- Master password: `<Password of your choice.>`
- Confirm password: `<Password of your choice.>`

Make note of the actual values used above since they will be used in later steps.

- In the *Configure advanced settings* screen, select *Yes* for the *Public accessibility*. Also, keep the default settings, but choose the name of the database, e.g., `cs1234_fall2345_<your last name>`. **Note the use of underscores**. This will be the name of the schema where tables and their records will be stored. Click on *Launch DB Instance* to continue. The database will take a few moments to be created after which you can navigate to it by clicking on *View DB Instance Details* or clicking *Instances* on the left.
- The details screen will be titled with the DB instance identifier chosen earlier, e.g., `cs1234-fall2345-<your last name>`. Scroll down to the *Connect* section. Make a note of the *Endpoint* since it will be used later to connect remotely to the database. The endpoint is the name of the server machine where the database is running, e.g.,
  - `cs1234-fall2345-annunziato.cne500ro4imj.us-west-2.rds.amazonaws.com`
- Also note the *Port* where the server is listening for incoming connections, e.g., `3306`. If the *Endpoint* is not yet available, wait a few more minutes while the database service is setup.

- Note that the connection might not be publicly available by default as denoted by the configuration *Publicly accessible: No*. To make the connection publicly available, under the *Security group*, click on the *Inbound* security group. In the security group screen, click the *Inbound* tab, and then the *Edit* button. Under the *Source* column, select *Anywhere* from the dropdown, and click *Save*. Verify that the *Publicly accessible* setting is set to *Yes*.
- You can use MySQL workbench to connect to the RDS server using the
  - Hostname: endpoint of the DB instance.
  - Username: username chosen in the above steps.
  - Password: password chosen in the above steps.
- Once the AWS DB has been setup and connection has been verified successfully, update the following details in *application.properties* file:
  - spring.datasource.url=jdbc:mysql://<endpoint>:<port>/
  - spring.datasource.username=<username_of_mysql_db>

- spring.datasource.password=<password>
- spring.jpa.hibernate.ddl-auto=update
- spring.jpa.show-sql=true
- spring.jpa.hibernate.naming-strategy=org.hibernate.cfg.ImprovedNamingStrategy
- spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect

A Small tutorial on using MySQL workbench can be found here.
https://docs.google.com/document/d/1hv9-wJv1Y2rTdaQmw4HVACNWsH5D5OfaN-BeVgghrBM/edit?usp=sharing

Create a Schema (15pts.)

Use SQL to create the following schemas with the properties, data types, and relations as described in the UML class diagram. All tables should define a primary key called "id" configured to auto increment if no value is provided. Foreign keys should have the same name as the table they refer to. Enforce required fields and cardinality. Do not use "enum" to implement the <<enumeration>>. Implement associations between the tables either as primary key/foreign key and/or additional mapping tables, e.g., roles.

1. (0pts) Create a brand new schema in a remote database hosted on Heroku or AWS. Do all your work in that remote database. As a deliverable, provide the connection string to connect to the remote database which should include the host URL (or IP address), the username and password to login to the database.
2. (5pts) Create tables **person**, **developer**, and **user**. Implement generalization using separate tables for each class, e.g., the ***normalized strategy***. Name the constraint on the foreign keys using the following pattern: subclass_superclass_generalization, where subclass and superclass are the subclass and superclass in the diagram. For instance if a person is a superclass and faculty is a subclass of person, then the foreign key constraint name would be faculty_person_generalization. The names of the tables and fields should use the following naming convention
    a. all table and field names must be lower case
    b. all compound words must have an underscore between words, e.g., WebsitePrivilege would become website_privilege
    c. All primary keys must be called “id” and use auto increment where possible
3. (5pts) Create tables website, page, widget, heading, html, youtube, image. Implement generalization using a single table, e.g., the ***denormalized strategy***. Use a new field called DTYPE to discriminate for the type, e.g., WIDGET, HEADING, HTML, YOUTUBE, IMAGE. Use the class name as the values of the field. Default heading size should be 2
4. (5pts) Create tables address, phones, website and page roles, and enumerations. If your database supports "enum", do not use it. Instead implement enumerations using the portable enumeration strategy discussed in class.

Implement Triggers (15pts.)

(7pts.) Write a trigger to create, update or delete ***website*** privileges when roles are created, updated or deleted for a website, such that all the privileges that apply for a particular role are given to the corresponding user and website.

(8pts.) Write a trigger to create, update or delete ***page*** privileges when roles are created, updated or deleted for a page, such that all the privileges that apply for that particular role are given to the corresponding user and page.

Use the following roles for the privileges:

1. owner - create, read, update, delete
2. admin - create, read, update, delete
3. writer - create, read, update
4. editor - read, update
5. reviewer - read

Implement Inserts (15pts.)

Provide SQL queries that insert the data shown below. Note that some will require inserting into more than one table. Use the IDs where provided, otherwise the ID fields should be configured to auto increment. Auto generated IDs can be used where not specified. Nested queries may be used to inquire about previously inserted data. Later inserts can assume data exists from earlier inserts.

1. (3pts.) Create the following developers and users. Insert into the correct tables depending on the type

id	Username	Password	First	Last	Type	Email	Key	User Agreement
12	alice	alice	Alice	Wonder	Developer	alice@wonder.com	4321rewq	N/A
23	bob	bob	Bob	Marley	Developer	bob@marley.com	5432trew	N/A
34	charlie	charlie	Charles	Garcia	Developer	chuch@garcia.com	6543ytre	N/A
45	dan	dan	Dan	Martin	User	dan@martin.com	N/A	true
56	ed	ed	Ed	Karaz	User	ed@kar.com	N/A	false

2. (3pts.) Create the following web sites for the developers above. For the created field use CURRENT_TIMESTAMP as the default value and use ON UPDATE CURRENT_TIMESTAMP for the updated field. This will allow to keep a history of when

the records were created and updated last.

id	Name	Description	Owner	Editor	Admin	Visits
123	Facebook	an online social media and social networking service	alice	bob	charlie	1234234
234	Twitter	an online news and social networking service	bob	charlie	alice	4321543
345	Wikipedia	a free online encyclopedia	charlie	alice	bob	3456654
456	CNN	an American basic cable and satellite television news channel	alice	bob	charlie	6543345
567	CNET	an American media website that publishes reviews, news, articles, blogs, podcasts and videos on technology and consumer electronics	bob	charlie	alice	5433455
678	Gizmodo	a design, technology, science and science fiction website that also writes articles on politics	charlie	alice	bob	4322345

3. (3pts.) Create the following pages for the web sites above. For the created field use CURRENT_TIMESTAMP as the default value and use ON UPDATE CURRENT_TIMESTAMP for the updated field. This will allow to keep a history of when the records were created and updated last.

id	Title	Description	Website	Editor	Reviewer	Writer	Views
123	Home	Landing page	CNET	alice	bob	charlie	123434
234	About	Website description	Gizmodo	bob	charlie	alice	234545
345	Contact	Addresses, phones, and contact info	Wikipedia	charlie	alice	bob	345656
456	Preferences	Where users can configure their preferences	CNN	alice	bob	charlie	456776
567	Profile	Users can configure their personal information	CNET	bob	charlie	alice	567878

4. (3pts.) Create the following widgets for the pages shown.

id	Name	Type	Text	Order	Width/Height	Url	Page
123	head123	heading	Welcome	0	null	null	Home
234	post234	html	Lorem	0	null	null	About
345	head345	heading	Hi	1	null	null	Contact
456	intro456	html	Hi	2	null	null	Contact
567	image345	image	null	3	50x100	/img/567.png	Contact
678	video456	youtube	null	0	400x300	https://youtu.be/h67VX51QXiQ	Preferences

5. (3pts.) Create the following phones and addresses for the users or developers shown

Username	Phones	Addresses
alice	**123-234-3456** 234-345-4566	**123 Adam St., Alton, 01234**, 234 Birch St. Boston, 02345
bob	**345-456-5677**	**345 Charles St., Chelms, 03455**, 456 Down St., Dalton, 04566, 543 East St., Everett, 01112
charlie	**321-432-5435** 432-432-5433 543-543-6544	**654 Frank St., Foulton, 04322**

Implement View (10pts.)

Create a view called "deleveloper_roles_and_privileges" that captures a developer's privileges across all websites and pages. The view should join various tables to capture the developer's personal information as well as the websites, pages, and their associated roles and privileges. The view should provide the following information:

1. Developer's first name
2. Developer's last name
3. Developer's username
4. Developer's email
5. Website's name
6. Website's visits
7. Website's last updated date

8. Developer's role in that website
9. Developer's privileges in that website
10. Page's title
11. Page's views
12. Page's last updated date
13. Developer's role in that page
14. Developer's privileges in that page

Implement Queries (15pts.)

Write SQL to implement the following queries. Assume the data inserted in the prior problem set. Nested loops may be used.

1. (3pts.) Retrieve developers
    a. Retrieve all developers
    b. Retrieve a developer with id equal to 34
    c. Retrieve all developers who have a role in Twitter other than owner
    d. Retrieve all developers who are page reviewers of pages with less than 300000 visits
    e. Retrieve the writer developer who added a heading widget to CNET's home page
2. (3pts.) Retrieve websites
    a. Retrieve the website with the least number of visits
    b. Retrieve the name of a website whose id is 678
    c. Retrieve all websites with videos reviewed by bob
    d. Retrieve all websites where alice is an owner
    e. Retrieve all websites where charlie is an admin and get more than 6000000 visits
3. (3pts.) Retrieve pages
    a. Retrieve the page with the most number of views
    b. Retrieve the title of a page whose id is 234
    c. Retrieve all pages where alice is an editor
    d. Retrieve the total number of pageviews in CNET
    e. Retrieve the average number of page views in the Website Wikipedia
4. (3pts.) Retrieve widgets
    a. Retrieve all widgets in CNET's Home page
    b. Retrieve all youtube widgets in CNN
    c. Retrieve all image widgets on pages reviewed by Alice
    d. Retrieve how many widgets are in Wikipedia
5. (3pts.) To verify the page and website triggers written earlier function properly:
    a. Retrieve the names of all the websites where Bob has DELETE privileges.
    b. Retrieve the names of all the pages where Charlie has CREATE privileges.

Implement Updates (15pts.)

1. (3pts.) Update developer - Update Charlie's primary phone number to 333-444-5555
2. (3pts.) Update widget - Update the relative order of widget head345 on the page so that it's new order is 3. Note that the other widget's order needs to update as well
3. (4pts.) Update page - Append 'CNET - ' to the beginning of all CNET's page titles
4. (5pts.) Update roles - Swap Charlie's and Bob's role in CNET's Home page

Implement Deletes (15pts.)

1. (3pts.) Delete developer - Delete Alice's primary address
2. (3pts.) Delete widget - Remove the last widget in the Contact page. The last widget is the one with the highest value in the order field
3. (4pts.) Delete page - Remove the last updated page in Wikipedia
4. (5pts.) Delete website - Remove the CNET web site, as well as all related roles and privileges relating developers to the Website and Pages

Deliverables

As a deliverable, Please fill the details in the “Submission.txt”. Failing to do so will not allow us to review your submission. Key items to submit are:

- the hostname,
- username,
- password,
- port

Submit your respective queries in the following files:

1. Queries for Create Schema: create.sql
2. Queries for Triggers: triggers.sql
3. Queries for Inserts: inserts.sql
4. Queries for Views: view.sql
5. Queries for Retrieve: retrieve.sql
6. Queries for Update: update.sql
7. Queries for delete: delete.sql

- Please create the above files and submisson.txt file and submit them in your github Assignment-2 folder of your repository.
- Submit your github link on the blackboard.

On Heroku, the connection string can be found in the Settings section of your dashboard. Click on Reveal Config Vars. Copy the environment variable labeled CLEARDB_DATABASE_URL and submit it on Blackboard

On AWS, click on your database identifier and in the *Connect* tab, you would find the endpoint(hostname) and port of your DB. Submit them along with your username and password.

Deliverables format
https://piazza.com/class/k08462fzewg2m0?cid=147