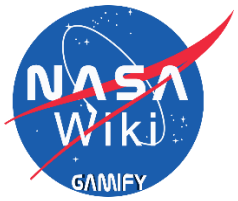
NASA EVA
Gamification

DATABASE DESIGN

PHASE II



Prepared By
Laura Addiego
Samia Alam
Kelli Corey
Charles Milk
Adeola Odusola
Hung Pham

Table of Contents

Version Number	Description of Change	Author	Date
1.0	Initial Creation of Document	Laura Addiego Prospero	06/25/2018
1.1	Review Document	Laura Addiego Prospero	07/15/2018

Overview

The database described in this document will augment the default MediaWiki database to allow for the gamification of the NASA EVA MediaWiki. The database will store the values necessary to track point earned by users, badge reward and leaderboard.

Specific Terms and Acronyms

Term/Acronym	Description
SQL	Structured Query Language
DDL	Data Definition Language
DML	Data Manipulation Language
ERD	Entity Relationship Diagram

Hardware and Software

Diagram Tool

SQL Developer Data Modeler tool for diagramming.

Database

MariaDB 10.1.30

MariaDB is platform independent and will run on Windows, Mac, or Linux.
HeidiSQL was used to interface with the database (HeidiSQL Version: 9.4.0.5125).

SQL Editor

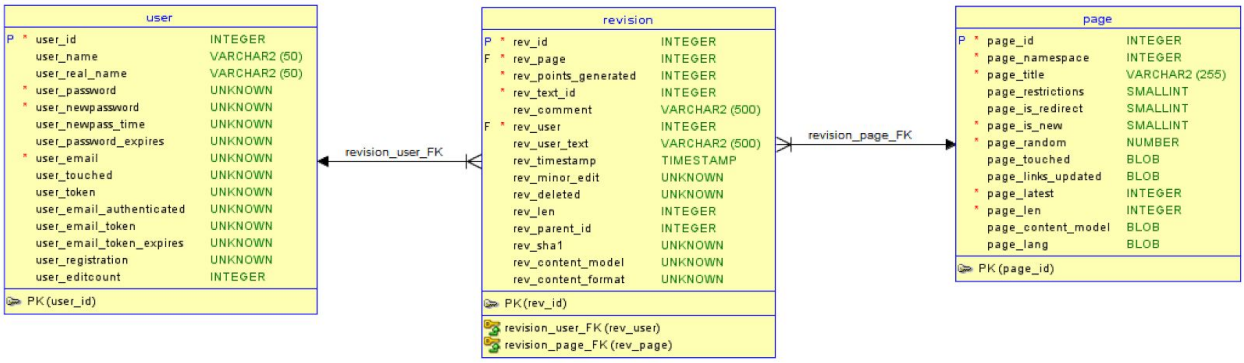
NotePad++.

DDL and DML

SQL will be used for the DDL and DML. The information captured in the ER diagram will be used to generate SQL scripts to create the database tables. In addition, SQL will also be used to create DML scripts to insert data to be used for testing.

ERD

The following ERD diagram contains three tables that are part of the default MediaWiki database.



From the default MediaWiki database we will be using the **user, revision, and page** tables because users will be earning points and badges according to their contributions (edits/revisions to pages).

Relationships

a) **user** and **page**
There is a **M:N relationship** between **user** table and **page** table because a user may revise many pages and a page may be revised by many users. To overcome a many-to-many relationship the application has created **revision** table which will hold the rev_page as a Foreign Key **(FK)** from the page table, and rev_user as a Foreign Key (**FK**) from the user table. By adding this bridge table (revision) it creates a **1:M** relationship between **user** and **revision tables** (a user may have 1 or many revisions, but a revision belongs to 1 and only 1 user) and a **1:M** relationship between **page** and **revision tables** (a page may have 1 or many revisions, but a revision belongs to 1 and only 1 page)

Cardinalities/Business Rules

a) **user** and **page**

- A user may contribute to zero, one, or many pages.
- A page may be contributed by one, or many contributors (users).

b) **user** and **revision**
- A user may revise zero, one, or many pages
- A revision must belong to one and only one user

c) **page** and **revision**
- A page may have one or many revisions
- A revision must belong to one and only one page.

Entities

There will not be an addition of custom entities.

Badges' information: name (which will also be used to pull its correspondent image) and their corresponding point-value will be stored in an array called "$wgBadges"on the "LocalSettings.php" file.

1. Entity Name: **user (this table is generated by the MediaWiki application)**

Entity Description: registered users in the MediaWiki database

Attribute Name	Description	Type	Other Info
user_id	to identify users	INT	Primary Key auto incremented
user_name	user's screen name	VARCHAR	Unique Mandatory
user_real_name	user's real name	VARCHAR	Mandatory
user_password	user's password	TINYBLOB	Mandatory
user_newpassword	user's new password	TINYBLOB	Mandatory
user_new_pass_time	datetime user's new password was generated	BINARY	
user_email	User's email address	TINYTEXT	Mandatory
user_touched	last time user logged in	BINARY	Mandatory
user_token	user's token (if applicable)	BINARY	Mandatory
user_email_authenticated	datetime user's email was authenticated	BINARY	
user_email_token	token for user's email (if applicable)	BINARY	
user_email_token_expires	datetime user's email token expires (if applicable)	BINARY	
user_registration	datetime user registered on site	BINARY	
user_editcount	number of times user has edited a page	INT	
user_password_expires	expiration date of user's password (if applicable)	VARBINARY	

2. Entity Name: **revision (this table is generated by the MediaWiki application)**

Entity Description: edits and revisions users make to the page

Attribute Name	Description	Type	Other Info
rev_id	identifies the revision	INT	Primary Key auto incremented
rev_page	identifies the page	INT	Foreign Key (user) Mandatory
rev_points_generated	helps track the number of points generated by the revision	INT	Mandatory
rev_text_id	assigns an id for the generated text	INT	Mandatory
rev_comment	high level comment of what the page revision was about	VARBINARY	
rev_user	identifies the user by the ID	INT	Mandatory
rev_user_text	identifies the user by the name	VARCHAR	Mandatory
rev_timestamp	time when the revision took place; it specifies year, month, day, hour, and minutes	BINARY	Mandatory
rev_minor_edit	identifies if users have marked the revision as “minor”	TINYINT	Mandatory
rev_deleted	identifies if the revision has been deleted	TINYINT	Mandatory
rev_len	counts the total number of characters –including spaces, special characters, etc. that are in the page	INT	
rev_parent_id	Identifies the previous revision to the current one.	INT	
rev_sha1	information set up by the database	VARBINARY	Mandatory
rev_content_model	information set up by the database	VARBINARY	
rev_content_format	information set up by the database	VARBINARY	

3. Entity Name: **page (this table is generated by the MediaWiki application)**

Entity Description: stores page information

Attribute Name	Description	Type	Other Info
page_id	identifies the page	INT	Primary Key auto incremented
page_namespace	this field has internal meaning for the application	INT	Mandatory unique
page_title	stores page's title	VARCHAR	Mandatory unique
page_restrictions	stores page's restrictions	TINIBLOB	Mandatory
page_is_redirect	identifies if the page redirects to another page	TINYINT	Mandatory
page_is_new	identifies if it is a new page	TINYINT	Mandatory
page_random	stores random number assigned to the page	DOUBLE	Mandatory
page_touched	identifies the date (year, month, day, hour, minute) when the page was modified	BINARY	Mandatory
page_links_updated	stores the date when links were updated	VARBINARY	
page_latest	identifies the latest revision ID	INT	Mandatory
page_len	counts the total number of characters in the page	INT	Mandatory
page_content_model	this field has internal meaning for the application	VARBINARY	
page_lang	stores the page's language	VARBINARY	

Assumptions and Special Considerations

1. Players may have multiple badges of the same type until they reach a higher level which they will be exchange for the higher level badge.

DDL Statements

Add Column to the Revision Table

```
/* = = = = = = = = = = = =

= =  ADD COLUMN TO revision TABLE

=  = = = = = = = = = = = = = */

ALTER TABLE revision

        ADD rev_points_generated INT(4) unsigned NOT NULL

        AFTER rev_page;
```