**Database Test Scripts**

**NASA EVA Gamification**

*Group 3*

**Prepared By**

Okechukwu Ogudebe

Victoria Guadagno

Jacqueline Macfadyen

Michael Salgo

Kevin Fortier

Table of Content

[Test Script < Database-02 > 3](#_Toc509837300)

[Identification 3](#_Toc509837301)

[Preparation of the environment 3](#_Toc509837302)

[Test Script 4](#_Toc509837303)

|  |  |  |  |
| --- | --- | --- | --- |
| **Version Number** | **Description of Change** | **Author** | **Date** |
| 1.0 | Initial Creation of Document | Okechukwu Ogudebe | 03/03/2018 |
| 1.1 | Revised Document for Formatting | Michael Salgo | 03/15/2018 |

# Test Script < Database-02 >

## Identification

Unique Identifier of Script: Database-02

Test Script Version: Version 02

Author of Test Script:

Test Object: Database

## Preparation of the environment

Before this script is loaded the following actions should be taken to meet the desired environment conditions.

* MariaDB / MySQL should be correctly configured
* Build the game’s database with real “functions”
* Load the database
* Configure the database to work correctly on the browser and testing tools

The following SQL query was run to create the gamification\_badges table.

CREATE TABLE IF NOT EXISTS /\*\_\*/ gamification\_badges (  
  user\_id int(10) UNSIGNED NOT NULL,  
  badge\_tag varchar(255) NOT NULL DEFAULT '',  
  badge\_rank varchar(255) NOT NULL DEFAULT '',  
  date\_badge\_earned binary(14) DEFAULT NULL,  
  PRIMARY KEY(user\_id, badge\_tag, badge\_rank)  
) ENGINE=InnoDB DEFAULT CHARSET=binary;

An unverified user profile was used in testing the database-02 script.

## Test Script

Tester: Okechukwu Ogudebe

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | **Logging** | | |
| **Step#** | **Action to take** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** |
| 1 | Connect game with the database | Game should connect to the database | Game connects with the database | Pass | The game will be storing user data in the gamification\_badges table. It is important to test if it successfully connects with the database before trying other tests. |
| 2 | Check if table gamification\_badges exists | gamification\_badges table should exist after running the SQL script mentioned above | gamification\_badges table exists | Pass | gamification\_badges table was successfully added to the database after running our SQL script. We will be storing information in this table |
| 3 | Check that the field user\_id is spelled correctly, is all lowercase, is of type int(10), and does not allow nulls | user\_id field should be spelled correctly and is in lowercase | user\_id field is spelled correctly, is all lowercase, is of type int(10), and does not allow nulls | Pass | user\_id is spelled correctly and meets our requirements. user\_id field could contribute to game errors if it does not match its requirements. |
| 4 | Check that the field badge\_tag is spelled correctly, is all lowercase, is of type varchar(255), and does not allow nulls | badge\_tag field should be spelled correctly and is in lowercase | badge\_tag field is spelled correctly, is all lowercase, is of type varchar(255), and does not allow nulls | Pass | badge\_tag will be storing user badge tag data. In this instance it has met our database requirement threshold |
| 5 | Check that the field badge\_rank is spelled correctly, is all lowercase, is of type varchar(255) | badge\_rank field should be spelled correctly and is in lowercase | badge\_rank field is spelled correctly, is all lowercase, is of type varchar(255), and does not allow nulls | Pass | badge\_rank field conforms to our requirements |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 6 | Check that the field date\_badge\_earned is spelled correctly, is all lowercase, is of type varbinary(14) and allows nulls | date\_badge\_earned should be spelled correctly, is all lowercase, is of type varbinary(14) and allows nulls | date\_badge\_earned is spelled correctly, is all lowercase, is of type varbinary(14) and allows nulls | Pass | Date\_badge\_earned will storing user badge rank and meets our database field requirements |
| 7 | Check that the primary key is comprised of user\_id, badge\_tag, and badge\_rank | user\_id, badge\_tag, and badge\_rank should be compromised as the table’s primary key | user\_id, badge\_tag, and badge\_rank act as gamification\_badges table primary key | Pass | user\_id, badge\_tag, and badge\_rank will act as our primary key. This helps in preventing duplicate user detail in our table. |
|  |  |  |  |  |  |