**Database Test Summary Report**

**NASA EVA Gamification**

*Group 3*

**Prepared By**

Michael Salgo

Okechukwu Ogudebe

Victoria Guadagno

Jacqueline Macfadyen

Kevin Fortier

Montrell Nuble

Table of Contents

[Introduction 3](#_Toc510341991)

[Test Summary 3](#_Toc510341992)

[Test Script Create Gamification Badges Table New 3](#_Toc510341993)

[Test Script 3](#_Toc510341994)

[Test Script Create Gamification Badges Table Preexisting 4](#_Toc510341995)

[Test Script 5](#_Toc510341996)

[Test Script Insert Gamification Badges New 6](#_Toc510341997)

[Test Script 7](#_Toc510341998)

[Test Script Insert Gamification Badges Preexisting 8](#_Toc510341999)

[Test Script 8](#_Toc510342000)

[Test Assessment 9](#_Toc510342001)

[Test Results 9](#_Toc510342002)

[Suggested Actions 9](#_Toc510342003)

|  |  |  |  |
| --- | --- | --- | --- |
| **Version Number** | **Description of Change** | **Author** | **Date** |
| 1.0 | Initial Creation of Document | Michael Salgo | 03/27/2018 |
| 1.1 | Population of Summary Report | Montrell Nuble | 03/28/2018 |

# Introduction

The purpose of this document is to test all of the scripts in the Database Test Suite to ensure that the database scripts function as desired and that there are no unexpected issues. Results of these tests, along with additional comments will be listed. Lastly, the test itself will be assessed as will be the results of the test itself with suggested actions thereafter.

# Test Summary

These four tests were put in place to ensure that the one database table required for the NASA EVA Gamification project, which keeps track of badges earned and is used to interact with the User Gamification Profile page, is fully functional and created as specified. Additionally, the statement that inserts data into this one database table is tested to ensure that records get created as specified. In both cases, we are testing the create and insert functionalities in situations representing new data created after the NASA\_EVA\_Gamification extension would be installed and existing data created before the extension would be installed. All four test cases worked as intended with all results passing.

## Test Script Create Gamification Badges Table New

Test Script Version: Version 01

Test Object: Database

Tester: Montrell Nuble

Test Results:

### Test Script

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | **Logging** | | |
| **Step#** | **Action to take** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** |
| 1 | Check that the table gamification\_badges does not exist in the database in use for MediaWiki | The gamification\_badges table does not exist in the MediaWiki database | The gamification badges table does not exist within the database | Pass |  |
| 2 | Copy the CREATE TABLE script from the add\_objects.sql file and paste it into the query editor of the database in use for MediaWiki | Code pasted into query editor matches the origin source | Code matches | Pass |  |
| 3 | Execute the query | Query is executed without error | Query executes without error | Pass |  |
| 4 | Check that the gamification\_badges table has been created by the SQL script; this may require a refresh of the tables listed | The gamification\_badges table now exists in the MediaWiki database | Table exist | Pass |  |
| 5 | Check that the field user\_id is spelled correctly, is all lowercase, is of type int(10), and does not allow nulls | The field user\_id is spelled correctly, is all lowercase, is of type int(10), and does not allow nulls | The field user\_id is spelled correctly, is all lowercase, is of type int(10), and does not allow nulls | Pass |  |
| 6 | Check that the field badge\_tag is spelled correctly, is all lowercase, is of type varchar(255), and does not allow nulls | The field badge\_tag is spelled correctly, is all lowercase, is of type varchar(255), and does not allow nulls | The field badge\_tag is spelled correctly, is all lowercase, is of type varchar(255), and does not allow nulls | Pass |  |
| 7 | Check that the field badge\_rank is spelled correctly, is all lowercase, is of type varchar(255), and does not allow nulls | The field badge\_rank is spelled correctly, is all lowercase, is of type varchar(255), and does not allow nulls | The field badge\_rank is spelled correctly, is all lowercase, is of type varchar(255), and does not allow nulls | Pass |  |
| 8 | Check that the field date\_badge\_earned is spelled correctly, is all lowercase, is of type binary(14), has a default of NULL, and allows nulls | The field date\_badge\_earned is spelled correctly, is all lowercase, is of type binary(14), has a default of NULL, and allows nulls | The field date\_badge\_earned is spelled correctly, is all lowercase, is of type binary(14), has a default of NULL, and allows nulls | Pass |  |
| 9 | Check that the primary key is comprised of user\_id, badge\_tag, and badge\_rank | The gamification\_badges table primary key is comprised of user\_id, badge\_tag, and badge\_rank | The gamification\_badges table primary key is comprised of user\_id, badge\_tag, and badge\_rank | Pass |  |

Additional Comments:

None.

## Test Script Create Gamification Badges Table Preexisting

Test Script Version: Version 01

Test Object: Database

Tester: Montrell Nuble

Test Results:

### Test Script

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | **Logging** | | |
| **Step#** | **Action to take** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** |
| 1 | Check that the table gamification\_badges already exists in the database in use for MediaWiki | The gamification\_badges table exists in the MediaWiki database | Table exist within database | Pass |  |
| 2 | Manually create two records in the gamification\_badges table with the following values (user\_id, badge\_tag, badge\_rank, date\_badge\_earned, respectively):   * 1, tag, rank, (NULL) * 2, tag, rank, (NULL) | The gamification\_badges table now contains at least two records, with user\_id, badge\_tag, badge\_rank, and date\_badge\_earned values of (respectively):   * 1, tag, rank, (NULL) * 2, tag, rank, (NULL) | The records were able to be created in the table | Pass |  |
| 3 | Copy the CREATE TABLE script from the add\_objects.sql file and paste it into the query editor of the database in use for MediaWiki | Code pasted into query editor matches the origin source | Code pasted into query editor matches the origin source | Pass |  |
| 4 | Execute the query | Query is executed without error | Query is executed without error | Pass |  |
| 5 | Check that the gamification\_badges table still exists | The gamification\_badges table still exists in the MediaWiki database | The gamification\_badges table still exists in the MediaWiki database | Pass |  |
| 6 | Check that the field user\_id is still spelled correctly, is all lowercase, is of type int(10), and does not allow nulls | The field user\_id is spelled correctly, is all lowercase, is of type int(10), and does not allow nulls | The field user\_id is spelled correctly, is all lowercase, is of type int(10), and does not allow nulls | Pass |  |
| 7 | Check that the field badge\_tag is still spelled correctly, is all lowercase, is of type varchar(255), and does not allow nulls | The field badge\_tag is spelled correctly, is all lowercase, is of type varchar(255), and does not allow nulls | The field badge\_tag is spelled correctly, is all lowercase, is of type varchar(255), and does not allow nulls | Pass |  |
| 8 | Check that the field badge\_rank is still spelled correctly, is all lowercase, is of type varchar(255), and does not allow nulls | The field badge\_rank is spelled correctly, is all lowercase, is of type varchar(255), and does not allow nulls | The field badge\_rank is spelled correctly, is all lowercase, is of type varchar(255), and does not allow nulls | Pass |  |
| 9 | Check that the field date\_badge\_earned is still spelled correctly, is all lowercase, is of type binary(14), has a default of NULL, and allows nulls | The field date\_badge\_earned is spelled correctly, is all lowercase, is of type binary(14), has a default of NULL, and allows nulls | The field date\_badge\_earned is spelled correctly, is all lowercase, is of type binary(14), has a default of NULL, and allows nulls | Pass |  |
| 10 | Check that the primary key is comprised of user\_id, badge\_tag, and badge\_rank | The gamification\_badges table primary key is comprised of user\_id, badge\_tag, and badge\_rank | The gamification\_badges table primary key is comprised of user\_id, badge\_tag, and badge\_rank | Pass |  |
| 11 | Check that the manually created two records in the gamification\_badges table still exist with the following values (user\_id, badge\_tag, badge\_rank, date\_badge\_earned, respectively):   * 1, tag, rank, (NULL) * 2, tag, rank, (NULL) | The gamification\_badges table still contains at least two records, with user\_id, badge\_tag, badge\_rank, and date\_badge\_earned values of (respectively):   * 1, tag, rank, (NULL) * 2, tag, rank, (NULL) | The table contains two records | Pass |  |

Additional Comments:

None.

## Test Script Insert Gamification Badges New

Test Script Version: Version 01

Test Object: Database

Tester: Montrell Nuble

Test Results:

### Test Script

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | **Logging** | | |
| **Step#** | **Action to take** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** |
| 1 | Manually create two records in the user table with the following values (user\_id, user\_name, user\_password, user\_newpassword, user\_email, user\_email\_authenticated, respectively):   * 10, 123, 123, 123, ‘’, (NULL) * 11, 456, 456, 456, ‘’, 4500000000000   000000000000000 | The user table now contains at least two records, with user\_id, user\_name, user\_password, user\_newpassword, user\_email, user\_email\_authenticated values of (respectively):   * 10, 123, 123, 123, ‘’, (NULL) * 11, 456, 456, 456, ‘’, 0x45000000000000   00000000000000 | The user table now contains at least two records, with user\_id, user\_name, user\_password, user\_newpassword, user\_email, user\_email\_authenticated values of (respectively):   * 10, 123, 123, 123, ‘’, (NULL) * 11, 456, 456, 456, ‘’, 0x45000000000000   00000000000000 | Pass |  |
| 2 | Copy the INSERT script from the add\_objects.sql file and paste it into the query editor of the database in use for MediaWiki | Code pasted into query editor matches the origin source | Code pasted into query editor matches the origin source | Pass |  |
| 3 | Execute the query | Query is executed without error | Query is executed without error | Pass |  |
| 4 | Check that one record was created in the gamification\_badges table for user\_id 11 with the following values (badge\_tag, badge\_rank, date\_badge\_earned, respectively):   * gamification-badge-emailverification * gamification-rank-1 * (NULL) | One record was created in the gamification\_badges table for user\_id 11 with badge\_tag, badge\_rank, date\_badge\_earned values of (respectively):   * gamification-badge-emailverification * gamification-rank-1 * (NULL) | One record was created in the gamification\_badges table for user\_id 11 with badge\_tag, badge\_rank, date\_badge\_earned values of (respectively):   * gamification-badge-emailverification * gamification-rank-1   (NULL) | Pass |  |
| 5 | Check that no record was created in the gamification\_badges table for user\_id 10 | No record was created in the gamification\_badges table for user\_id 10 | No record was created in the gamification\_badges table for user\_id 10 | Pass |  |

Additional Comments:

None.

## Test Script Insert Gamification Badges Preexisting

Test Script Version: Version 01

Test Object: Database

Tester: Montrell Nuble

Test Results:

### Test Script

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | **Logging** | | |
| **Step#** | **Action to take** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Remarks** |
| 1 | Check that the manually created two records in the user table still exist with the following values (user\_id, user\_name, user\_password, user\_newpassword, user\_email, user\_email\_authenticated, respectively):   * 10, 123, 123, 123, ‘’, (NULL) * 11, 456, 456, 456, ‘’, 4500000000000   000000000000000 | The user table still contains at least two records, with user\_id, user\_name, user\_password, user\_newpassword, user\_email, user\_email\_authenticated values of (respectively):   * 10, 123, 123, 123, ‘’, (NULL) * 11, 456, 456, 456, ‘’, 0x45000000000000   00000000000000 | The user table still contains at least two records, with user\_id, user\_name, user\_password, user\_newpassword, user\_email, user\_email\_authenticated values of (respectively):   * 10, 123, 123, 123, ‘’, (NULL) * 11, 456, 456, 456, ‘’, 0x45000000000000   00000000000000 | Pass |  |
| 2 | Copy the INSERT script from the add\_objects.sql file and paste it into the query editor of the database in use for MediaWiki | Code pasted into query editor matches the origin source | Code pasted into query editor matches the origin source | Pass |  |
| 3 | Execute the query | Query is executed without error | Query is executed without error | Pass |  |
| 4 | Check that one record still exists in the gamification\_badges table for user\_id 11 with the following values (badge\_tag, badge\_rank, date\_badge\_earned, respectively):   * gamification-badge-emailverification * gamification-rank-1 * (NULL) | One record still exists in the gamification\_badges table for user\_id 11 with badge\_tag, badge\_rank, date\_badge\_earned values of (respectively):   * gamification-badge-emailverification * gamification-rank-1 * (NULL) | One record still exists in the gamification\_badges table for user\_id 11 with badge\_tag, badge\_rank, date\_badge\_earned values of (respectively):   * gamification-badge-emailverification * gamification-rank-1   (NULL) | Pass |  |
| 5 | Check that no record was created in the gamification\_badges table for user\_id 10 | No record was created in the gamification\_badges table for user\_id 10 | No record was created in the gamification\_badges table for user\_id 10 | Pass |  |

Additional Comments:

None.

# Test Assessment

I believe the test script was adequate in testing all major functionality.

# Test Results

No deviations made, test cases ran and passed as expected.

# Suggested Actions

No suggestions.