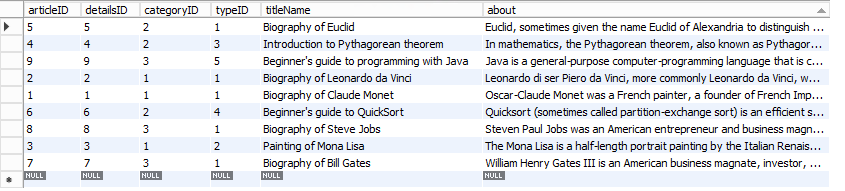
**Testing Document**

IT6037 Data Access and Management

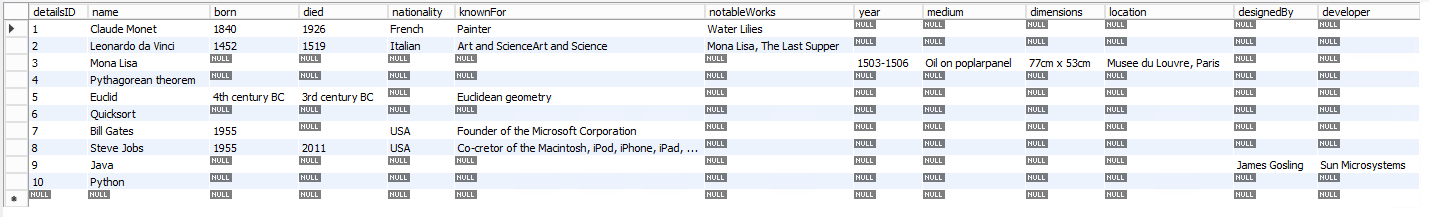
Team: SWESLBS

**Testing the view of the database**

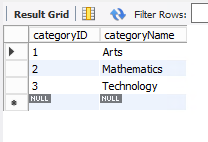
**Article Table**



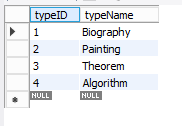
**ArticleDetails Table**



**Category Table**



**Type Table**



**Test 1**

Test Case

|  |  |
| --- | --- |
| Test Condition | When a student search's the database for the category Arts |
| Preconditions | MySQL workbench is opened, and the project database is available |
| Test | Checking that searchArticleByCategory(‘...’) works when searching Arts |
| Input Data | CALL searchArticleByCategory(‘Arts’); |
| Expected Result | Three articles should be displaying with Category: Arts, Title: “the title”, and About: “the article” |
| Post-Condition | Showing a small table with three areas filled in with three different articles |

Test Script

|  |  |
| --- | --- |
| Requirement ID | DB 1.1 |
| Purpose | To see that a student can search for one of the category's |
| Precondition | MySQL workbench is opened, and the project database is available |
| Test Data | CALL searchArticleByCategory(‘Arts’); |
| Process Steps | 1. Double click the project schema if not highlighted 2. Enter test data into the query: CALL searchArticleByCategory(‘Arts’); 3. Press Ctrl + Enter |
| Post Test | You should see a result grid with three different articles all with the categoryName Arts |

**Results:** It showed the results grid with all three articles with the category Arts

Incident Report

|  |  |
| --- | --- |
| Incident Date/Time |  |
| Tester |  |
| Contact Details |  |
| Logon |  |
| Test Script ID |  |
| Incident Severity |  |
| Repeatable? |  |
| Description |  |

**Test 2**

Test Case

|  |  |
| --- | --- |
| Test Condition | When a student search's the database for the category Mathematics |
| Preconditions | MySQL workbench is opened, and the project database is available |
| Test | Checking that searchArticleByCategory(‘...’) works when searching Mathematics |
| Input Data | CALL searchArticleByCategory(‘Mathematics’); |
| Expected Result | Three articles should be displaying with Category: Mathematics, Title: “the title”, and About: “the article” |
| Post-Condition | Showing a small table with three areas filled in with three different articles |

Test Script

|  |  |
| --- | --- |
| Requirement ID | DB 1.2 |
| Purpose | To see that a student can search for one of the category's |
| Precondition | MySQL workbench is opened, and the project database is available |
| Test Data | CALL searchArticleByCategory(‘Mathematics’); |
| Process Steps | 1. Double click the project schema if not highlighted 2. Enter test data into the query: CALL searchArticleByCategory(‘Mathematics’); 3. Press Ctrl + Enter |
| Post Test | You should see a result grid with three different articles all with the categoryName Mathematics |

**Results:** It showed the results grid with all three articles with the category Mathematics

Incident Report

|  |  |
| --- | --- |
| Incident Date/Time |  |
| Tester |  |
| Contact Details |  |
| Logon |  |
| Test Script ID |  |
| Incident Severity |  |
| Repeatable? |  |
| Description |  |

**Test 3**

Test Case

|  |  |
| --- | --- |
| Test Condition | When a student search's the database for the category Technology |
| Preconditions | MySQL workbench is opened, and the project database is available |
| Test | Checking that searchArticleByCategory(‘...’) works when searching Technology |
| Input Data | CALL searchArticleByCategory(‘Technology’); |
| Expected Result | Three articles should be displaying with Category: Technology, Title: “the title”, and About: “the article” |
| Post-Condition | Showing a small table with three areas filled in with three different articles |

Test Script

|  |  |
| --- | --- |
| Requirement ID | DB 1.3 |
| Purpose | To see that a student can search for one of the category's |
| Precondition | MySQL workbench is opened, and the project database is available |
| Test Data | CALL searchArticleByCategory(‘Technology’); |
| Process Steps | 1. Double click the project schema if not highlighted 2. Enter test data into the query: CALL searchArticleByCategory(‘Technology’); 3. Press Ctrl + Enter |
| Post Test | You should see a result grid with three different articles all with the categoryName Technology |

**Results:** It showed the results grid with all three articles with the category Technology

Incident Report

|  |  |
| --- | --- |
| Incident Date/Time |  |
| Tester |  |
| Contact Details |  |
| Logon |  |
| Test Script ID |  |
| Incident Severity |  |
| Repeatable? |  |
| Description |  |

**Test 4**

Test Case

|  |  |
| --- | --- |
| Test Condition | When a student search's the database for part of the title of the article |
| Preconditions | MySQL workbench is opened, and the project database is available |
| Test | Checking that if your search for a word in the title it will find all articles with that word in the title |
| Input Data | CALL searchArticleByKeyword(‘%Biography%’); |
| Expected Result | It will show five different articles with the keyword biography in the title |
| Post-Condition | Showing a small table with two areas filled in with five different articles |

Test Script

|  |  |
| --- | --- |
| Requirement ID | DB 1.4 |
| Purpose | To see if you search for part of a title it will display all the articles with that keyword in it. |
| Precondition | MySQL workbench is opened, and the project database is available |
| Test Data | CALL searchArticleByKeyword(‘%Biography%’); |
| Process Steps | 1. Double click the project schema if not highlighted 2. Enter the test data into the query: CALL searchArticleByKeyword(‘%Biography%’); 3. Press Ctrl + Enter |
| Post Test | You should see a result grid with five different articles all with the word Biography in the title |

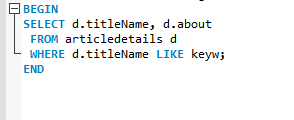
**Results:**

Incident Report

|  |  |
| --- | --- |
| Incident Date/Time | 24/09/2019 17:04 |
| Tester |  |
| Contact Details |  |
| Logon |  |
| Test Script ID | DB.1.4 |
| Incident Severity | Major |
| Repeatable? | Yes |
| Description | Would keep coming up with an error as shown below |

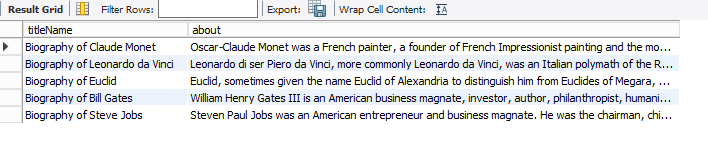


**Debugging**



The code was wrong, was looking in the wrong table

**Retesting:** worked fine, shows five articles



**Test 5**

Test Case

|  |  |
| --- | --- |
| Test Condition | When a student search's the database for the full title of the article |
| Preconditions | MySQL workbench is opened, and the project database is available |
| Test | Checking that if your search for the full title it will find only the article with that title |
| Input Data | CALL searchArticleByKeyword(‘Biography of Bill Gates’); |
| Expected Result | It will only show one article with the title Biography of Bill Gates |
| Post-Condition | Showing a small table with two areas filled in with correct article |

Test Script

|  |  |
| --- | --- |
| Requirement ID | DB 1.5 |
| Purpose | Testing that if the full correct title is given it will only show the article with that title |
| Precondition | MySQL workbench is opened, and the project database is available |
| Test Data | CALL searchArticleByKeyword(‘Biography of Bill Gates’); |
| Process Steps | 1. Double click the project schema if not highlighted 2. Enter the test data into the query: CALL searchArticleByKeyword(‘Biography of Bill Gates’); 3. Press Ctrl + Enter |
| Post Test | You should see a result grid with the correct articles with the title Biography of Bill Gates |

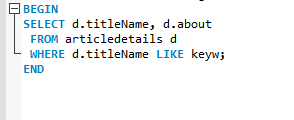
**Results:**

Incident Report

|  |  |
| --- | --- |
| Incident Date/Time | 24/09/2019 17:06 |
| Tester |  |
| Contact Details |  |
| Logon |  |
| Test Script ID | DB.1.5 |
| Incident Severity | Major |
| Repeatable? | Yes |
| Description | Would keep coming up with an error as shown below |

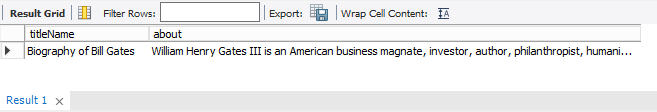


**Debugging**



The code was wrong, was looking in the wrong table

**Retesting:** worked fine, shows one result



**Test 6**

Test Case

|  |  |
| --- | --- |
| Test Condition | When a student search's the database for the incorrect title no articles will show up |
| Preconditions | MySQL workbench is opened, and the project database is available |
| Test | Checking that if your search for the incorrect title it will find no articles with that title |
| Input Data | CALL searchArticleByKeyword(‘Super Troopers’); |
| Expected Result | No articles will be found |
| Post-Condition | You should see a result grid with no data in the grid |

Test Script

|  |  |
| --- | --- |
| Requirement ID | DB 1.6 |
| Purpose | Testing that if an incorrect title is given it will show no articles with that title |
| Precondition | MySQL workbench is opened, and the project database is available |
| Test Data | CALL searchArticleByKeyword(‘Super Troopers’); |
| Process Steps | 1. Double click the project schema if not highlighted 2. Enter the test data into the query: CALL searchArticleByKeyword(‘Super Troopers’); 3. Press Ctrl + Enter |
| Post Test | You should see a result grid with no articles with the title Super Troopers |

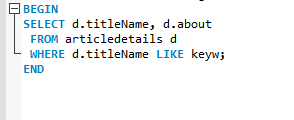
**Results:**

Incident Report

|  |  |
| --- | --- |
| Incident Date/Time | 24/09/2019 17:12 |
| Tester | B Smith |
| Contact Details | 02040971765 |
| Logon | 91030609 |
| Test Script ID | DB.1.6 |
| Incident Severity | Major |
| Repeatable? | Yes |
| Description | Would keep coming up with an error as shown below |

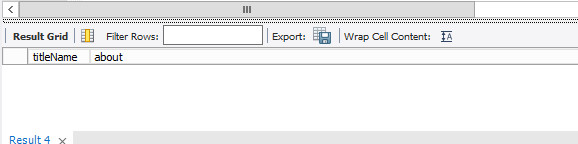


**Debugging**



The code was wrong, was looking in the wrong table

**Retesting:** worked fine, shows empty table



**Test 7**

Test Case

|  |  |
| --- | --- |
| Test Condition | Can add an article to the database |
| Preconditions | MySQL workbench is opened, and the project database is available |
| Test | Can add an article with set fields to the database |
| Input Data | INSERT INTO ‘project’.’articledetails’ (‘detailsID’, ‘name’) VALUES (‘10’, ‘Python’);  INSERT INTO ‘project’.’article’ (`articleID`, `detailsID`, `categoryID`, `typeID`, `titleName`, `about`) VALUES(‘10’, ‘10’, ‘3’, ‘5’, ‘Beginner\’s guide to programming with Python’, ‘pie’); |
| Expected Result | You will get a green tick and the data will be entered into the tables |
| Post-Condition | A new article will be made into the database |

Test Script

|  |  |
| --- | --- |
| Requirement ID | DB 1.7 |
| Purpose | To ensure that an article can be added into the database |
| Precondition | MySQL workbench is opened, and the project database is available |
| Test Data | INSERT INTO ‘project’.’articledetails’ (‘detailsID’, ‘name’) VALUES (‘10’, ‘Python’);  INSERT INTO ‘project’.’article’ (`articleID`, `detailsID`, `categoryID`, `typeID`, `titleName`, `about`) VALUES(‘10’, ‘10’, ‘3’, ‘5’,, ‘Beginner\’s guide to programming with Python’, ‘pie’); |
| Process Steps | 1. Double click the project schema if not highlighted 2. Enter the test data into the query: INSERT INTO ‘project’.’articledetails’ (‘detailsID’, ‘name’) VALUES (‘10’, ‘Python’); INSERT INTO ‘project’.’article’ (`articleID`, `detailsID`, `categoryID`, `typeID`, `titleName`, `about`) VALUES(‘10’, ‘10’, ‘3’, ‘5’, ‘Beginner\’s guide to programming with Python’, ‘pie’); 3. Press Ctrl + Enter |
| Post Test | If you look at the Article and ArticleDetails table you will see a new article |

**Results:** in both the tables had the new data

Incident Report

|  |  |
| --- | --- |
| Incident Date/Time |  |
| Tester |  |
| Contact Details |  |
| Logon |  |
| Test Script ID |  |
| Incident Severity |  |
| Repeatable? |  |
| Description |  |

**Test 8**

Test Case

|  |  |
| --- | --- |
| Test Condition | Able to modify an article |
| Preconditions | MySQL workbench is opened, and the project database is available |
| Test | Modify the about section of the article |
| Input Data | UPDATE 'project`.`article`  SET ‘about’ = ‘python can be used to make some games’ WHERE (‘articleID’ = ‘10’); |
| Expected Result | You will get a green tick and the article about section will change from “pie” to “python can be used to make some games” |
| Post-Condition | The article with articleID 10 will be modified |

Test Script

|  |  |
| --- | --- |
| Requirement ID | DB 1.8 |
| Purpose | Testing that we can modify an article |
| Precondition | MySQL workbench is opened, and the project database is available |
| Test Data | UPDATE ‘project`.`article` SET ‘about’ = ‘python can be used to make some games’ WHERE (‘articleID’ = ‘10’); |
| Process Steps | 1. Double click the project schema if not highlighted 2. Enter the test data into the query:  UPDATE ‘project`.`article` SET ‘about’ = ‘python can be used to make some games’ WHERE {‘articleID’ = ‘10’); 3. Press Ctrl + Enter |
| Post Test | If you look at the Article table you will see article 10 about section will say “python can be used to make some games” |

**Results:** The about section in the database now shows the new data

Incident Report

|  |  |
| --- | --- |
| Incident Date/Time |  |
| Tester |  |
| Contact Details |  |
| Logon |  |
| Test Script ID |  |
| Incident Severity |  |
| Repeatable? |  |
| Description |  |

**Test 9**

Test Case

|  |  |
| --- | --- |
| Test Condition | Able to delete an article |
| Preconditions | MySQL workbench is opened, and the project database is available |
| Test | Deleting an article from the database |
| Input Data | DELETE FROM `project`.`article` WHERE (‘articleID’ = ‘10’); |
| Expected Result | You will get a green tick and the article with articleID 10 will be deleted |
| Post-Condition | The article about python will no longer be in the database |

Test Script

|  |  |
| --- | --- |
| Requirement ID | DB 1.9 |
| Purpose | Testing that an article can be deleted |
| Precondition | MySQL workbench is opened, and the project database is available |
| Test Data | DELETE FROM `project`.`article` WHERE (‘articleID’ = ‘10’); |
| Process Steps | 1. Double click the project schema if not highlighted 2. Enter the test data into the query: DELETE FROM `project`.`article` WHERE (‘articleID’ = ‘10’); 3. Press Ctrl + Enter |
| Post Test | If you look at the Article table you will see that there is no longer an article 10 in the database |

**Results:** The article about python has been removed

Incident Report

|  |  |
| --- | --- |
| Incident Date/Time |  |
| Tester |  |
| Contact Details |  |
| Logon |  |
| Test Script ID |  |
| Incident Severity |  |
| Repeatable? |  |
| Description |  |