Student Results

Functions and techniques used to test that a solution is working correctly.

**Testing**

Testing involves checking that a software solution is working as expected.

By testing, any bugs (or errors) in the solution can be identified, and then these bugs can be removed.

When creating a database management system (DBMS) there are three areas that can be tested:

* Input masks and validation rules
* Queries
* Calculations in queries

**Student Results DBMS**

Below is a database of student results. The DBMS is a flat-file database consisting of one table.

A sample of the data is shown below:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **StudentID** | **FirstName** | **Gender** | **Age** | **Class** | **Test1** | **Test2** |
| S125 | Dexter | Male | 20 | 12A | 98 | 100 |
| S126 | Cassandra | Female | 19 | 12C | 80 | 80 |
| S127 | Bruce | Male | 18 | 12B | 60 | 70 |
| S128 | Susan | Female | 20 | 12A | 50 | 90 |
| S129 | Zhen | Male | 19 | 12C | 66 | 50 |
| S130 | Hang | Female | 18 | 12C | 98 | 92 |
| S131 | Jack | Male | 20 | 12A | 45 | 78 |
| S132 | Tien | Female | 19 | 12C | 56 | 87 |
| S133 | Tim | Male | 18 | 12B | 48 | 50 |
| S134 | Rose | Female | 19 | 12C | 67 | 45 |

The following data dictionary was created during the design stage that outlines all of the fields used in the results table including any properties.

tblResults

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Data type** | **Field Size**  **(if short text)** | **Other properties** |
| StudentID | Short Text | 5 | Primary key |
| Name | Short Text | 50 |  |
| Age | Number | 50 | >=18 |
| Class | Short Text | 3 | Input mask = 00L |
| Test1 | Number |  | Validation rule >= 0 and <=100 |
| Test2 | Number |  | Validation rule > =0 and <=100 |

The validation rule for Test 1 and 2 consists of two conditions >= 0 and <=100. Both these conditions need to be tested using test data values that are within the range and values that are outside the range.

The best test data values would be -1, 0, 100 and 101.

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Test data | Expected result | Actual result |
| Validation rule for Test1 | -1 | Not accepted |  |
| Validation rule for Test1 | 0 | Accepted |  |
| Validation rule for Test1 | 100 | Accepted |  |
| Validation rule for Test1 | 101 | Not accepted |  |

**Required**

1. **Open** the Student Results database, then open the results table.
2. **Edit** the data in the table by inputting the test data identified above.
3. **Record** what actually happens in the Actual Result column of the test table.

**Queries**

When testing queries the test data may include all of the data within the tables of the DBMS.

Often, the number of records that will be returned will be identified in the expected result.

Based on the query below:

**qryA**SELECT StudentID, Name, Gender, Age  
FROM tblResults  
WHERE Gender= ‘Male’;

The test table would look like this:

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Test data | Expected result | Actual result |
| qryA is working correctly | Records in tblResults | 5 records returned |  |

**Required**

Open the DBMS, run the query and record what actually happens.

Complete the first three columns for qryB.

**qryB**SELECT StudentID, FirstName, Gender, Age, Class  
FROM tblResults  
WHERE Class= ‘12A’;

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Test data | Expected result | Actual result |
|  |  |  |  |

**Required**

Open the DBMS, run the query and record what actually happens.

**qryC**

SELECT FirstName, Gender, Age, Test1  
FROM tblResults  
WHERE Test1 > 50;

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Test data | Expected result | Actual result |
|  |  |  |  |

**Required**

Open the DBMS, run the query and record what actually happens.

**qryD**SELECT FirstName, Gender, Age, Test2  
FROM tblResults  
WHERE Test2 >= 50;

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Test data | Expected result | Actual result |
|  |  |  |  |

**Required**

Open the DBMS, run the query and record what actually happens.

**Calculations**

Any calculations performed in a query also require testing. Often one or two records can be selected so an expected result can be calculated.

The query below is calculating the average mark for each student:

**qryE**SELECT FirstName, Gender, Age, ((Test1+Test2)/2) as AverageMark  
FROM tblResults;

The test table will appear as below:

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Test data | Expected result | Actual result |
| Calculation of AverageMark | Dexter Mark1 = 98 and Mark2 = 100 | 99 |  |
| Calculation of AverageMark | Rose  Mark1 = 67  Mark2 = 48 | 57.5 |  |

**Required**

Open the DBMS and run the query and record what actually happens.