

Automated Grading of Access® Databases Using the Matlab® Database Toolbox

Dr. Curtis Cohenour Ph.D., P.E. P.E., Ohio University

Dr. Cohenour is an Assistant Professor in the Ohio University Engineering Technology and Management Department, in Athens, Ohio. He received a Bachelor of Science degree from West Virginia Institute of Technology in 1980, a Master of Science degree from Ohio University in 1988, and a Ph. D. in Electrical Engineering from Ohio University in 2009. He is a registered professional engineer in West Virginia, and Ohio.

Dr. Cohenour has worked in Industry as an electrical engineer and project manager. He joined Ohio University in 2002 as a research engineer working for the Ohio University Avionics Engineering Center. He has worked on projects covering a wide variety of avionics and navigation systems such as, the Instrument Landing System (ILS), Microwave Landing System (MLS), Distance Measuring Equipment (DME), LAAS, WAAS, and GPS.

His recent work has included research with the Air Force Research Laboratory in Dayton, Ohio, aimed at understanding and correcting image geo-registration errors from a number of airborne platforms.

Ms. Audra Lynn Hilterbran, Ohio University

Audra Hilterbran is an instructional technologist in the Russ College of Engineering and Technology at Ohio University, Athens, Ohio. She works with faculty to design and redesign courses through best practices in technology integration. She is currently pursuing a PhD in Instructional Technology. Her background is in Applied Linguistics and English as a Second Language (ESL) instruction. She also teaches courses in Computer Assisted Language Learning. Her research interests include learning aptitudes, usability testing, and the facilitation of classroom communication.

Automated Grading of ACCESS® Databases Using the MATLAB® Database Toolbox

Abstract

This evidence based paper describes an automated method of grading Microsoft Access® databases using the MATLAB® Database Toolbox. A total of 70 students completed an average of 50 individual database assignments per week over a five-week portion of a semester. The assignments were from the Benchmark Series Microsoft® Access® 2013 (BM) text. The course schedule covered two chapters every week. Students were required to import and manipulate either raw text data or a partially completed database to complete each assignment. Students were given unique starter data to discourage academic dishonesty. The automated grader compared each student database submission with the starter data and a truth database. In addition, the database relations, keys, and data types were compared to the truth and graded. Graded assignments were returned to students within one day of submission, thus eliminating a potential bottleneaking of instructor feedback. The use of the grader allows for increased formative assessment opportunities rather than a reliance on traditional, summative examinations. Automated grading offers a flexible and reliable tool for instructors to provide objective, detailed, and timely feedback to students. The features of the automated grader, current outcomes, and future directions are discussed.

Introduction

An automated grader was developed using the MATLAB® Database Tool box for grading the Access® portion of a freshman-level Enterprise Computing Course (ECC) in Engineering Technology and Management (ETM) at Ohio University. The ECC is a freshmen level class that all students entering the ETM program must take and pass as a requisite for subsequent ETM courses. The class is taught online and is approximately 50% Microsoft Excel, 40% Access, and 10% other Microsoft Office products, including PowerPoint and Visio. Database skills are necessary for subsequent courses taught in detail using Microsoft Structured Query Language (SQL). Access® introduces the students to tables, keys, relationships, and queries which are used in the subsequent classes.

The main motivation for developing the grader was to provide detailed, timely, and objective feedback on a high volume of gradable material to students in a large online class. The promptness of the feedback was crucial, as it allowed students to reflect on their errors, take corrective actions, and fill knowledge gaps well in advance of summative assessments. The weekly output of gradable assignments for each of the 70 enrolled students typically covered two chapters and included ten Access® databases with approximately 60 tables, queries, reports, and datagrams (relationships). Students submitted these databases and printouts (in PDF format) using a Box file sharing service (Box Inc., n.d.). The automated grader allowed the instructor to grade and return all the assignments within a day of submission. Additionally, the automated grader maintains a record of individual performances and provides indicators to identify top performers in the course who may do well in the advanced courses in the ETM program.

Before the development of the automated grader, managing the grading was challenging for ECC instructors, as students were unable to receive quality feedback in a prompt manner. Because of the sheer volume of items to be graded, instructors were forced to find alternate methods to full

manual grading. Methods tried for this course include: a) grading one submission at random per week, b) using highly specific quizzes to elicit correct/incorrect values, c) using the BM publisher's assessment tool, SNAP, and d) relying on summative assessments only (no grading of homework assignments).

In the random selection method, a graduate student or teaching assistant reviewed a single database submitted for each student per week. The submission was considered complete only if all the components were included, such as relationships, tables, and queries, reports, and forms. Although the method was designed to decrease grading time, it is not amenable for use in a large class, as manual grading still demands the focus and availability of the human graders, the evaluation of individual components of each assignment, and additional checks for completeness and academic dishonesty, across several students. In addition to a slow turnaround time, human graders may not be able to provide consistent, objective feedback and specific suggestions for improved learning outcomes for each students.

Developing quizzes which elicit specific information from completed assignments assumes that students have completed the required work to perform well in the quizzes. These quizzes are multiple choice and are graded automatically. Given that information related to the database assignments is freely available online (for example, cramster.com), there is no objective way to uphold student accountability unless the quizzes are proctored. Proctoring requires extra time and resources from the ECC teaching team.

Using the publisher's assessment tool SNAP provides options for automated grading but presents a unique set of disadvantages. First, it requires students to purchase an electronic pass, which increases the financial burden. Second, SNAP covers only a part of the topic, meaning students are not exposed to all the learning activities outlined in the course objectives. Third, the SNAP tool is designed in a way that allows students to skip directly to the assessment portion without visiting the learning content. Students can consult online resources in real time to fare the exams without the need to master the learning material or activities. Finally, the SNAP system requires the student to complete the exercise in small steps. The student is not allowed to make mistakes, discover his/her error, and make corrections.

The final possibility is to simply not grade the assignments and rely solely on summative assessments. The problem with this approach is that the students learn to cram for an exam, not to how use Access®.

The method described in this paper addresses all of the concerns of the alternate methods. The use of automated grading relieves the instructor of a heavy grading load. The automated grader evaluates all student submissions and provides specific and timely feedback. Students don't feel cheated if they do 90% of the work correctly and make mistakes on the one item that is graded. Students are free to make mistakes in one step and then backtrack and correct the errors as they are discovered. Students loose (at least partially) the incentive to game the system by using plagiarized data, or cramming for exams.

Background

The grader described here is a continuation of the Automated Excel® Grader described in (Cohenour & Hilterbran, 2016). In this previous work, an Excel® spreadsheet is opened and graded using the MATLAB® ActiveX Com controls. This grader is used for the Excel® portion of the ECC described above. Students graded with the Excel® grader are in a good position to benefit from use of the Access® grader. The grader output is similar, and the requirements including filenames, are understood by the time the students start to use Access®. There are other grading systems available for grading Access® databases. These include MEAGER (Hill, 2003), and the proprietary system SNAP from Paradigm Publishing. In addition, McCart (McCart & Jarman, 2008) describes a system to detect potential plagiarism. In Hill's (Hill, 2003) system the grading is performed in MS Access® and compares the student submission with a correct version of the database and checks tables, and queries with respect to, column names, data types, and size. The system also checks the keys and relationships. The system does not check the data. A similar system is presented in (N., Širanović, & Kavran, 2012) and is implemented using Visual Basic for Applications (VBA).

Automated Grading of Access® databases using the MATLAB® Database Toolbox

The MATLAB® Database Toolbox provides an interface to Access® (and other) databases. The data, formats, and relationships can all be extracted and used within MATLAB®. The user may read, write, and alter the database. For the automated grader, only reading is required. The feasibility of this grading technique is based on the Paradigm Education Solutions *Benchmark Series Microsoft® Access®2013 (BM)* Level I (Rutkosky, Roggenkamp, & Rutkosky, 2014) textbook. The student opens a root database from the student files for the chapter, renames the database, modifies the database, and saves the database. The student also prints tables, reports and other items as requested in the text. If the student follows the instructions (including sorts), every database will be the same. Grading is then carried out by comparing the student submission with the truth database. A more free-form text where column names and formats were not specified would be more problematic for the grader.

The BM text is divided into eight chapters with a unit review after Chapters Four and Eight, for a total of ten units. Each unit contains an explanation of the material followed by a project. There may be several projects within a unit. The projects include detailed step-by-step instructions. A typical project may require four pages of text for the instructions. Assessments are located at the end of each unit. These are similar in scope to the projects but the instructions are less detailed and may only require about one half of a page.

A typical sequence of instructions for a project might as follows:

1. Open **AL1-C2-Griffin.accdb** and enable the contents.
2. Click the DATABASE TOOLS tab.
3. Click the Relationships button in the Relationships group.
4. ...

In an assessment this might be reduced to

1. Open **AL1-C2-Griffin.accdb**.
2. Create a one-to-many relationship between EmpId in the Employees table and the EmpId in the Expenses table.
3. ...

In both cases, there is sufficient information in the instructions to ensure that a given student database will match the truth database. This includes additional data entered by the student and sorted order of the students completed database. Students are instructed to print at various points in the process. Printing in Access® allows for a variety of page formatting operations. These operations are described in the text and are a useful part of the course. Students are instructed to print to PDF and use a default name for printing. To address the problem of duplicate print names, students are instructed to add a sequential number to the PDF file (for example, Renters.pdf, Renters2.pdf, and Renters3.pdf). Currently, the prints are checked to see if they exist and are not graded otherwise. In the future, they could be graded by converting to an image and using correlation to evaluate correctness.

For each database, there are three files to consider: the root file, the truth file, and the student's submission file. The root file is supplied by BM and contains some of the data and settings used for the exercise. The student file refers to the submitted database containing operations performed by the student as required by the text. The truth file contains the correct solution for comparing the student submission file. Gradable items are determined by comparing the truth file to the root file, and the student's submission is evaluated against the truth file.

The truth file is generated by doing the exercise as described in the BM text. This file was then compared to the submitted student files, using the grader, to find any errors in the truth file. Once the truth file was verified it was used to grade the student files. This only needs to be done once.

Grading is performed for each table, query, and relationship. The truth table is summarized, followed by a summary of the student's table and a list of errors by table. If the error list is too long (which may occur if a lengthy table is not sorted correctly), it will be truncated. The grader is programmed to apply the following rules to individual parts of the submission files. If there is a new table in the submission file, the entire table is graded. If there is an existing table with a minimum of two new records, then only the new records are graded. If there is a table with a new column, only the new column is graded. If a relationship in the submission file already exists in the root database, it is excluded from grading, while a new relationship is always graded. This process is followed by a summarized score for the database. An example of the grader output is shown below:

John,

Your grade for Module 10 is attached

51 SmithJohn, js123456

AL1-C5-Griffin.accdb Sheet:

Truth DataBase

Employees (30 x 7) (Rows x Columns):

Key EmpID: Short Text 255

LastName: Short Text 255

FirstName: Short Text 255
BirthDate: Date/Time
AnnualSalary: Currency
HireDate: Date/Time
DeptID: Short Text 2
DeptID: Short Text 2

Your DataBase

Employees (30 x 7) (Rows x Columns):

Key EmpID: Short Text 255
LastName: Short Text 255
FirstName: Short Text 255
BirthDate: Date/Time
AnnualSalary: Currency
HireDate: Date/Time
DeptID: Short Text 2
DeptID: Short Text 2

Table Employees:

Column EmpID:
Column LastName:
Row 30: "Williamson" should be "Williamson"
Column FirstName:
Column BirthDate:
Column AnnualSalary:
Column HireDate:
Column DeptID:

AL1-C5-Griffin.accdb

Score is 5 out of 6, 83.3%

In the above example, the database contains five tables (one table is shown here for purposes of brevity). Within the Employees table, The EmpID is a key and inserted by Access. There is one new line with six entries, one of which contains an error. The student has submitted the file with a minor error; there is one "l" in "Williamson". This pattern of grading is repeated for each table and query across all databases. The total possible and the total correct scores are tallied. The PDFs and other files are checked and assigned a point value per Equation 1. If the PDF exists, the student is given a score. If it is missing, the results are noted.

$$PntsPerPdf = \text{floor} \left(\frac{MaxAccessPnts}{4 \times NumOfPdfs} \right) \quad \text{Equation 1}$$

Where: *MaxAccessPnts* is the maximum number of points for the Access Database
NumOfPdfs is the number of PDF files for the assignment
floor(*) returns the largest integer less than the *

PntsPerPdf is the integer number of points per PDF

The results appear as follows:

...

Total score for Access files: 1369 out of 1542 = 88.8%.

File found : PlanCodeCQuery.pdf

File found : Plans.pdf

File found : Plans2.pdf

File found : Prices.pdf

File found : Projects.pdf

File missing: ProjsManagedByEmp1947Query.pdf

7 points per PDF or other file

Total score for Pdf and other files : 364 out of 378 = 96.3%.

Weekly Score Summary:

51 SmithJohn, js123456

Total score: 1733 out of 1920 = 90.3%.

Weekly points: 9.0 out of 10.0

In this example, the student has received 1369 out of 1542 points for the Access® files, and 364 out of 378 points for the PDF, to a total of 1733 out of 1920 points, which is 90.3% of the points for the assignment. Each student is assigned a number. In the above example, the number for the student named John Smith is 51, with a student ID js123456. The student's folder in Box is named SmithJohn and the data is saved in a MATLAB® *.mat file. This feedback can be updated en masse, or one student at a time (in cases of late student submissions).

A MATLAB® script converts this save file to MATLAB®. The correct column for the grades is located and the student ID is used to locate the row and insert the grades. This process is performed for each student from the Access® grader. The modified CSV file is saved to the instructor's hard drive, which can be uploaded to Blackboard using the work off line/upload files option.

Improvements

The current tool can be enhanced for efficiency and accuracy with regards to grading already exiting items. In the future, these items could be graded by converting the relevant component within a submission to an image and using a correlation to evaluate for accuracy.

A plagiarism prevention program has been implemented using the technique described in Kline and (Kline & Janicki, 2003). Each student is provided with a unique set of student files to discourage students from resorting to shortcuts or plagiarism in submitting their assignments. Further refining of this process is needed. Given that the tool requires students to trace a learning curve and overcome some complex instructions, instructors considering the tool are suggested to provide clear instructions regarding the tool and to implement rounds of learner training before administering and discuss related queries, so that expectations regarding assignments and submission are clear.

A current initiative is to use automated grading techniques to assist with the evaluation of applied activities. To ensure students are evaluated on their Excel and Access® skills, part of the ECC summative assessments include a practical where the students are given general instructions and are required to produce a spreadsheet or database to meet certain requirements. Case Studies from the BM text are used to meet this directive. A rubric provided with the case studies assigns points to various items. These practicals must be graded manually because they are free-form, but the grading can be assisted with MATLAB®. The rubric is converted to a spreadsheet with the values entered for a perfect score. MATLAB® copies the rubric, names it with the student name, and opens it. MATLAB® then opens the database or spreadsheet file. The instructor examines the student's submission and scores any deficient items using the rubric. The files are closed and MATLAB® opens files for the next student. When this is completed, a MATLAB® script reads all of the rubrics and generates a score file which is used to update Blackboard gradebook, as described above. This procedure takes about 3 minutes per student. The rubrics used are detailed and provide good feedback to the student.

Summary

In summary, the automated Access® grader reduces the manual grading load for the ECC class. The grader provides timely, detailed, and objective formative feedback to the learner. Instructors can use the time otherwise consumed by grading to focus on intervening with struggling students, generating creative and engaging course content, or mentoring advanced students. In addition to improved class efficiency, the grader forces students to perform the assigned work. The combination of plagiarism detection and grading of all assignments reduces the ways in which a student can avoid the work and, by extension, the learning that is required by the course.

References

- Box Inc. (n.d.). *Box Homepage*. Retrieved Jan. 2, 2016, from <https://www.box.com/home>
- Cohenour, C., & Hilterbran, A. (2016). Automated Grading of Excel Workbooks Using Matlab. *Proceedings of the 2016 ASEE Annual Conference & Exposition*. New Orleans.
- Hill, T. (2003). MEAGER: Microsoft Excel automated grader. *The Journal of Computing Sciences in Colleges*, 18(6), 151-164.
- Kline, D., & Janicki, T. (2003). Enhancing economic and finance learning through automated grading of spreadsheet exercises. *Journal of Economics and Financial Education*, 2(2), 23-29.
- McCart, J., & Jarman, J. (2008). A Technological Tool to Detect Plagiarized Projects in Microsoft Access. *IEEE Transactions on Education*, 51(2), 166-173.
- N., M., Širanović, Ž., & Kavran, K. (2012). Tools for Grading Students' Exercises for Microsoft Access Applications. *MOPRO 2012: Proceedings, 35th International Convention of Information and Communication*, (pp. 1290-1292). Opatija, Croatia.
- Rutkosky, N., Roggenkamp, A., & Rutkosky, I. (2014). *Benchmark Series Microsoft® Access 2013 Level 1*. St. Paul: Paradigm Publishing, Inc. .