The need for change in the way we teach accounting information systems

Roger K. Doost
School of Accountancy and Legal Studies, Clemson University, Clemson, South Carolina, USA

Keywords
Accounting,
Information,
Systems development,
Technology, Databases

Abstract
The one area in accounting that shows the sign of volatility, unsettlement, and continuous change is accounting information systems. While textbooks have been revised and updated in line with the change in technology, the auxiliary teaching materials do not seem to have kept pace. Professors teaching the course may be handicapped by their own shortcomings as well as the inadequacy of the supplementary material. The emphasis on database and database management should be supplemented with relevant software and teaching material to make the course more meaningful and the efforts more fruitful. One easily available tool is Access. Incorporation of Access into the course provides an excellent demonstration into application and utility of database management systems.

Managerial Auditing Journal
17/5 [2002] 277–282

[ISSN 0268-6902]
[DOI 10.1108/02686900210429713]

Introduction

This paper is based on the author's over two decades of struggle in teaching the course, Accounting Information Systems (AIS), and his success and failure in this regard. This observation is made in conjunction with the variety of resource material and textbooks that he has used plus informal discussions with various colleagues who are involved with this subject matter.

Until recently, most accounting schools considered the AIS course as any other accounting course that needs be taught by the department's accounting professors. The prerequisite for teaching the course was the instructor's interest and/or experience in the subject matter. The quick pace in technology has changed this perception. The ideal instructor is one who is well-versed in technology-related issues as well as the accountant's and the auditor's needs on this matter.

It appears that most professors who have taught AIS as well as other accounting courses receive a lower student evaluation rating for the former as compared to any other course that they may have taught. Is this because of the quality of textbooks, professors, the subject matter itself, a combination of these factors, or some other phenomena?

The typical AIS textbooks cover the topics of data management, computer-based transaction processing, database modeling and applications, risk and internal control structure, general control and application controls, security measures, auditing of information systems, decision support and expert systems, transaction cycles, and the development of information systems. Few textbooks provide software supplements to enhance learning of the topics listed. Supplementary materials use basic accounting packages such as, Peachtree or Quicken to demonstrate computerized accounting software.

Instructors who have little training in AIS and have limited system and computer background are further limited in teaching this subject matter effectively with the use of textbooks available in the market. What is missing is a linkage between introduction of the ideas and the subject matter and some practical hands-on experience as well as hiring of individuals who are qualified and trained in the area.

The ideal professor to teach this subject matter is one who has adequate graduate work in the areas of both accounting and computer science and has had years of training and on-the-job experience. Such a professor should also keep up current with the ever-changing state of technology. Unfortunately, such talent comes with a price tag that most schools cannot afford to pay.

Course coverage

The linkage between the subject matter and the actual work is not difficult in some areas and is a huge task and barrier in others. For example, the students can be assigned and learn more about computers, printers, and other auxiliary equipment by taking a tour of a computer store, understanding the machines involved and writing a report about it. Mastering the subject of security is feasible through taking a comprehensive tour of a modern computer center and understanding the issues dealing with measures taken for protection of facilities, hardware, data, files, programs, and Web-based sources. It is one thing to read about such security measures, and it is another to virtually see the operation of the system. Computer security involves the security guards, receptionists, electronic scanning devices, motion detection devices, locked doors and restricted entry policies, television

monitors, locked terminals, inaccessible terminals, card-activated locks, employee badges, log in/log out of visitors, photographing the visitors, and escorting visitors to sensitive areas. It is also helpful to see protection measures from natural environment or disasters through the use of sprinkler systems, fire proof vaults, air conditioning and humidity controls, waterproof floors, walls, and ceilings, proper water drainage facilities, under floor water detectors, water pumps, sump pumps, fireproof construction, halon gas containers, auxiliary power supplies, accessible power switches, prudent location, disaster recovery and contingency plans, and isolation of the computer site.

Another visit to a modern computer center and a chat with the operator and system manager can also be enlightening when we find out about preventive maintenance procedures, backup hardware systems, and insurance coverage. We can also become better informed about access logs, control logs, access control software, system and program change logs, library logs, transaction logs, snap shots, rollback, and rollforward procedures. Demonstrating the features of tape-file protection rings, write-protect floppies, hard disks, and CDs, external file labels, internal file labels, read-only memory, batch control logs, and lockout procedures help us understand the means available to protect data from loss or alterations. Such a visit would also help us understand the nature of backup procedures, off-premise storage, dual servers, multiprogramming, multiprocessing, disk mirroring, reconstruction procedures, rollback and recovery procedures, and checkpoints for reconstruction of lost files. A dialogue with the internal security group who monitor such security measures can be enlightening as well[1].

Gaps in effectiveness

Four gaps still remain in making an AIS course more effective. First, *auditing through the computer* techniques such as, test data, integrated test facility, embedded audit module, program code checking, parallel processing, parallel simulation, and controlled processing are often covered in detail in textbooks without adequate practical examples. Such a coverage should be supplemented with useful demonstrative software to make some sense for the students. Second, *auditing with the computer* techniques such as, extracting data from files, calculating with data, performing comparisons with data, summarizing data, analyzing data, reorganizing data, selecting sample data, gathering statistical data, and printing confirmation requests should also be demonstrated through the use of applicable software (Wilkinson *et al.*, 2000, Ch. 10, pp. 341-72). Third, the discussion and extensive coverage of hierarchical, network, relational, and object-oriented database models without some sample software for demonstration and learning make little sense (Wilkinson *et al.*, 2000, Ch. 6, pp. 190-230). Finally, an AIS course may be strengthened by introducing other software packages such as a typical accounting package, decision support systems, artificial intelligence and expert systems.

The third deficiency stated above is somewhat addressed by a supplementary textbook for building accounting systems through the use of Access (James and Schneider, 2000). The rest of this paper is used in demonstrating how Access is used to teach the concept of relational database. It appears that incorporation of the above would require the AIS course to be taught in at least a two-semester sequence capped with a graduate course which would deal primarily with relevant cases and demonstration of various techniques mastered during the undergraduate program.

Access

Access appears to be one of the misunderstood and least used software in the Microsoft Office family of software. While millions use the Word and Excel packages, far fewer use the Access package. There could be a variety of reasons for this gap aside from the fact that other packages may be more appealing to a wider audience. There is a longer lead-time in learning Access and its applications. Access is counterintuitive to accountants; i.e. it defies the notion of double-entry bookkeeping. Each item needs to be recorded once and only once. As such, Access or any database management system introduces the notion of single entry bookkeeping. Due to versatility of Access, virtually all company needs can be satisfied through its use. Access tables can hold financial and non-financial data and extract columns and rows instructed according to management's needs – something that is impossible to do quickly under the traditional, non-versatile accounting systems.

Access is a relational database management system. Relational database systems are the most intuitively appealing

and easier to use as compared to other database systems such as hierarchical or network models. Access resources are *table, query, form, report, macro, and modules.* Tables are used for storing data. Tables must be normalized; i.e. they cannot contain repeating groups, and there should not be functional and transient dependendencies. What this basically means is that data such as invoice detail will appear in a separate table from invoice where we store general information such as invoice number and date, and we create another table for the customer where general information about the customer is stored. This particular format of storage is referred to as *Third Normal Form* that is often used in RDBMS. This method of data storage allows retrieval of data in virtually innumerable formats, depending on management specific needs.

Queries are used for selecting rows and columns from tables in response to specific questions. Forms are extraction from tables in a simple and intuitive way. Reports are used as outputs from tables such as sales orders, purchase orders, invoices, receiving reports, vouchers, and checks. Macros are commands that are built to carry out certain tasks. Modules are programs written to support macros. The conventionally accepted suffix for a table is *tbl*, for a query is *qry*, for a form is *frm*, for a report is *rpt*. Special tables are sometimes created for the mere purpose of looking up values, they are called "look up tables" and are designated with the prefix, *tlkp*. Another convention is to use field names in a concatenated (joined) fashion. For example, we show customer number as CustomerNumber. Each module provides at least two views: *design view* is used for construction/design of a table, query, form, or report. *Datasheet view* is used for entering of data (tables, forms) or seeing the results (query, report).

Tables

The basic storage entity for a relational database system is a two-dimensional object with rows and columns called a *table*. Each row corresponds to one instance of data like a record. Columns correspond to different characteristics called attributes. For example, consider a customer file that holds information such as, customer number, customer name, customer address, customer phone number, and customer credit line on various customers. A collection of tables that are related is known as a database. For example, we can have a customer table, a salesperson table, an invoice table, and an inventory table in a database. Several types of data can be stored in a table: text, memo, number, date/time, currency, hyperlink, auto-number, yes/no, and OLE. The data in a row remains with the row, even if the rows are stored or displayed in a different order. The order of table columns is not important except if we choose a *primary key* (the unique identifier of a record) that should appear in the first column. Tables are first *designed* and then viewed in *data sheet view*. The normalization rules must be followed in constructing a table; i.e. repeating groups must be avoided and functional dependencies must be avoided. In practical terms, we do not mix invoice information with customer information because each customer could have several invoices, and each invoice would have several lines. In design view, we identify *field names* (CustomerNumber, CustomerName, etc.), *field type* (text, number, currency, etc.), *field attributes* (field size, input mask, caption, validity rule, indexed, etc.). Input masks and validity rules are used to ascertain proper and valid entry of data. For most fields, we use *text* for *field attributes. Currency* is used for financial numbers and *Number* is used for numbers such as quantity that is used in calculations. The *caption* field allows designating what field title we wish to appear for the field when we go to datasheet view. Access provides a detailed help menu to guide the user in using proper input masks and validity rules. While input mask works on the field one letter/digit at a time, validity rule works on the whole field and either accepts or rejects it in its entirety. After a table is designed, we can start entering data (populating the table) while in datasheet view. Figure 1 is an example of a Customer table in design view. Figure 2 is an example of a Customer table in data sheet view.

Joining related tables

Related tables maintain their relationship through a *primary key/foreign key* association. For example, a customer table has as its primary key, the customer number. The sales invoice table has as its primary key the invoice number and as its foreign key the customer number. By going to relationships options in the Tools menu, we can bring to the surface all the related tables and create necessary relationships through dragging primary keys to their foreign key positions in related tables when needed. Such relationship needs to be established to be able to work with related tables. If the foreign key is not worded or spelled exactly as the

primary key, the association needs to be manually established through dragging the subject primary key to its related foreign key in another table. Establishing such relationships is defined as *referential integrity*. Referential integrity rules prevent us from adding a record to a related table if there is not an associated record in the primary table. We can easily right click and add or remove tables while in the relationship mode.

Figure 1
Customer table in design view. The mouse is at the CustomerNumber field

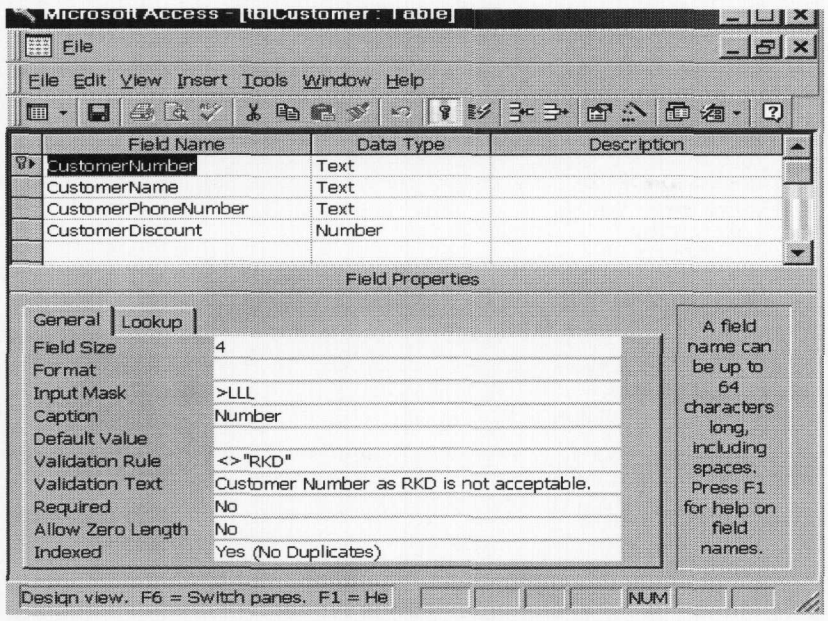
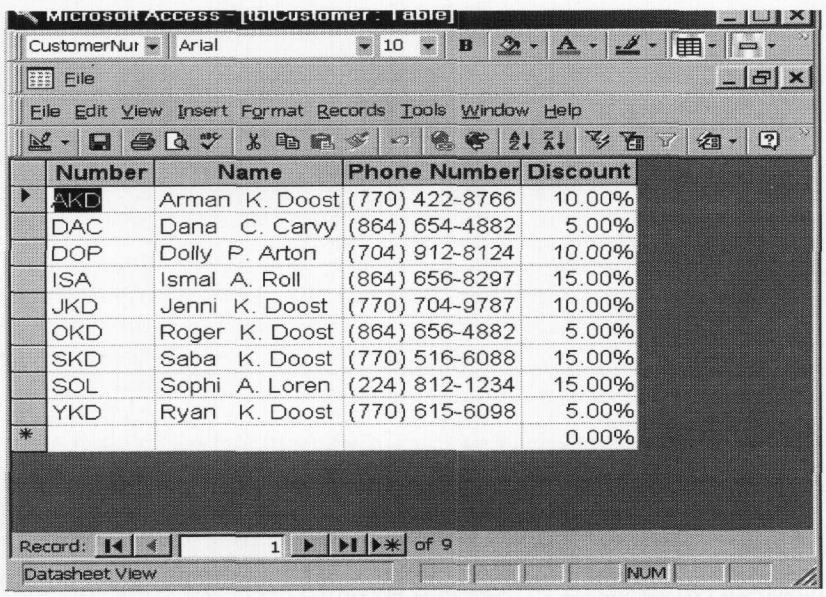


Figure 2
Customer table in datasheet view



Queries

Tables are where data is stored. Queries, forms, and reports access data in tables but they do not store them separately. They just extract and display data. Most queries are called *select queries*, because they retrieve rows from tables. Another form of query extracts columns from a table and that is called a *project query*. A *Join query* extracts desired rows and columns from a table. Queries can extract rows and columns from one or many tables. We select rows from a table by specifying under *Criteria* the items of interest. We select columns by choosing the desired columns from a table and displaying those columns in design view of a query before running it in data sheet view. For example, we may be interested to retrieve all inventory items that have fallen below 25 units. When there are more than one criteria for a column, it should be so specified by using an *And* in the subject column under criteria. If there are multiple criteria that affect different columns, those restrictions should be identified under criteria of respective columns. Conditional queries dealing with "*and, or*" situations should be so specified under *Criteria* and *Or* rows of indicated columns. For example, a query may ask for a list of all purchase orders that were more than one thousand dollars *and* issued within the last one month *or* were ordered from a particular vendor. Fields in a query can be sorted in ascending or descending order. Comparison operators such as > for greater than, < for less than, or = for equal to can be used to specify a range of values to be extracted from a column in a query.

It is a violation of normalization rules to include extensions in tables. Access does not perform calculations on tables. If some value is changed in the tables, the subject extensions would not be changed. The solution is to include any needed calculation in a new column in a query. Changing the format of certain columns as well as combining of columns can also be achieved through a query. A new column such as, *Total: [Quantity]*[Price]* would get the relevant data from quantity and price columns and create a new column called total to provide the necessary extension. Three columns for first name, middle initial, and last name may be combined into one field in a similar manner. An option available while in the query menu is called *Action Queries*. Action queries include *Update, Delete, Append, Crosstab*, and *Make table*. These specialized queries change the nature of the underlying tables.

Roger K. Doost
The need for change in the way we teach accounting information systems
Managerial Auditing Journal
17/5 [2002] 277–282

Forms

While in table mode, we can use the menu to convert a table to a form using Autoform. For more elaborate form design, we can go to forms and click New and choose to create a form from the list in Design view, Form Wizard, AutoForm: Columnar, AutoForm: Tabular, AutoForm: Datasheet, Chart Wizard, or PivotTable Wizard. A combobox allows us to choose the tables and queries to be used in creating a form. With the form wizard, we can elect to choose only the fields that we need in the final form. While in design view, we can modify the form by deleting, modifying or adding Labels and Text boxes as well as other buttons such as a Label, Text Box, Combo Box, Command Button, Rectangle, etc., using the *toolbox* on the toolbar. Data entry via a form is easier and more intuitive than entering data directly into a table. When selecting Design view for creating forms, we choose to make a form from scratch with the help of toolbox. The distinction made for Label and Text box in Access is important. With Label, we choose the headings or titles. Real data reside in Text boxes. Creating some forms such as, an invoice, purchase order, or receiving report requires accessing several tables such as, customer table, invoice table, invoice line-item, invoice-inventory table, and inventory table. The first two tables are needed to create the upper part of the form (customer number, name, address, etc.) while the latter three tables are needed to create the lower part of the form called sub-form. The structure will then be saved as *form* and *subform*. The complete form can be viewed while in datasheet view. The Records menu gives the user the option of filtering. *Filtering* allows us to put restrictions on what we view. After choosing filter, we must choose *Apply Filter* to be able to view the desired outcome. *Filter by form* selects from a form while *filter by selection* selects data from a table. We can physically and manually perform cosmetic changes such as, changing the width, length, position, color, background, etc., of a field or the entire form, or do the same by right clicking Properties, and performing the needed changes under the Format option.

Reports

We can create reports from scratch or by using the report wizard that guides us through selecting the fields that we want from one or a combination of tables. The report wizard allows us to view the format of the report and to go back and make changes to it if necessary. As you click next for the next screen, the wizard also guides you to group and sort several columns. The summary option gives you the opportunity to get totals and averages for the desired columns. In the final screen, you are given the option of viewing the report or going to its design view to provide additional necessary changes. Like forms, we can add labels and text box controls to a report. A report can come with report header and footer and page header and footer. We can use View from the menu bar to add headers and footers or to close unneeded headers and footers. We can also add other features by using View and sorting and grouping option. You can delete unwanted objects and controls from a report or add objects and controls to it. You will find the page break in the toolbox, by placing it in the report, we can then provide the page breaks as needed.

Properties

Each form, report, query or table has a set of *Properties*. We can get a form or a report Properties by going to Edit menu and select form or report while in design view. Then, we can go to View menu and select property. Alternatively, we can go to the corner of the form or report in design view, right click, and then choose property. Each field would in turn have properties. While you get the property, you can leave it open, and as you click from one field to another field, the property screen would display the properties of the selected fields. Properties differ from tables to queries to forms to reports. Properties for Forms are the most detailed. Within properties for Forms, there are four major options: *Format, Data, Event, Other*. Format property allows us to do cosmetic changes such as *Caption, Font, Font size, Width, Height, Back Style, Back Color, Border Style*, etc. Data and Event properties have a list of items for bound controls (text boxes) but none for relevant labels. While in the data section of a field property, we can do substantive changes. For example, we can go to the control source property and insert a certain formula so that the field would provide a needed calculation such as the sum total of a column. Other items under Data are *Input Mask, Default Value, Validation Rule, Validation Text, Enabled, Locked*, and *Filter Lookup*. The Enabled and Locked features allow us to lock certain fields to avoid access and alteration. Under Event property, we can attach macros to particular fields to perform the task *Before Update, After Update, On Change, On Enter, On Exit, On Click*, etc. The

Other property shows *Name* (that is, the internal name of the field), *Tab Stop* (indicating whether the tab should stop on that particular field or not), *Short Menu Bar* (in case we decide to create a special menu bar in conjunction with a form), etc. *All* – under which the detail from the other four properties are displayed. Properties for the entire form, as opposed to individual labels or text boxes, would vary as you click the mouse while leaving the Properties open.

Macros

Macros help automate and simplify the use of tables, queries, forms, and reports. Macros can be attached to forms, etc., and perform certain functions such as, go to first, previous, next, last, or new record. Other macros can be used to save, delete, and cancel records. The above *action macros* will be attached to *On Click Event* property of the relevant buttons. Macro tools are taken from the Toolbox on the toolbar. You can use the macro's format property to use pictures that explain the designed macro better. Access allows you to create your own action menus in conjunction with the current menu or completely replace the menu with that of your own creation. Many macros are *conditional macros*. These macros are used extensively in accounting for data verification such as field check, validity check, reasonableness check, range check, limit check, etc. These macros may be attached to the Before Update Event property. These macros usually include an expression such as "inventory on hand", a domain, such as "inventory table", and a criteria which refers to where we should look for such data.

Why database management systems?

Accounting has traditionally been financially oriented with only secondary attention to non-financial data. Today's accounting task is much broader than providing summary financial statements. A database management system (DBMS) such as, Access, allows responding to a host of questions and provision of details on financial as well as non-financial numbers. Although DBMS has been around for over three decades, many companies, particularly, the smaller ones do not take full advantage of its various possibilities. The whole business cycle including the revenue cycle (sales orders, packing slips, sales, invoicing, and receivables), purchasing cycle (purchase order, receiving reports, inventory, vouchers, payments, journals), payroll cycle (personnel actions, payroll calculations, payroll register, payroll checks, earning statements, journals), production cycle (job order, processing, hybrid, activity-based costing work in process, finished goods, cost of sales, journals) can be successfully transferred to a workable, flexible, and responsive business system using an effective DBMS. Such a system will be responsive to managers' needs by providing *ad-hoc* reports inclusive of desired financial and non-financial information. This process requires diligence and perseverance on the part of business and academic community, but it is well worth it. Teaching students a bunch of definitions in accounting information systems without effective, hands-on experience is an exercise in futility and we are not helping them in running the cutting edge and technology-based businesses of tomorrow. Neither are they being equipped to deal effectively with the technology-based audit requirements.

Note

1 For a detailed coverage of computer security measures refer to *Accounting Information Systems* by Wilkinson *et al.* (2000), Ch.r 9, pp. 307-40.

References

Perry, J.T. and Schneider, G.P. (2000), *Building Accounting Systems*, Southwestern University, Cincinnati, OH.

Wilkinson, J.W., Cerullo, M.J., Raval, V. and Wong-On-Wing, B. (2000), *Accounting Information Systems*, 4th ed., Wiley, New York, NY.