SeQueL: THE OTHER AUDIT TOOL—

Using software to execute audit procedures improves audit effectiveness and efficiency.

PART I: BASIC, COMPLETENESS, AND UNIQUENESS PROCEDURES

PASCAL A. BIZARRO AND ANDY GARCIA

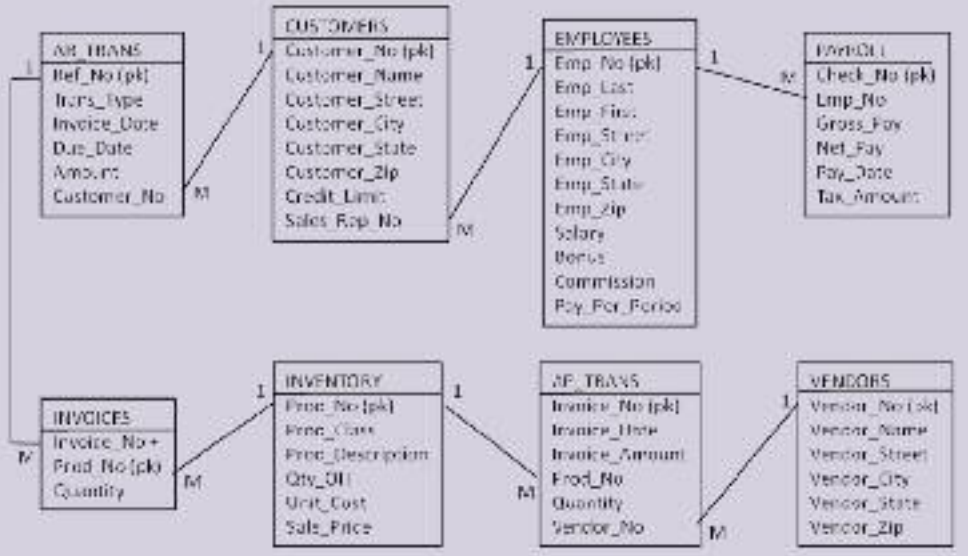
The purpose of this article is to assist auditors and IT professionals with the automation of common audit procedures through the use of Structured Query Language (SQL) in Microsoft Access. Auditors who are able to read and create basic queries are better suited to address the challenges encountered in information technology (IT) environments. Additionally, knowledge of SQL queries may allow auditors to switch to IT audit and helps auditors better understand audit software such as ACL and IDEA.

Automating audit procedures through the use of software improves both audit effectiveness and audit efficiency. Audit effectiveness is improved through the proper use of the software, which results in procedures being carried out in an error-free manner. Audit effectiveness is also improved because automation frees valuable audit resources for more complex analysis and other value-added procedures, which can only be done by an auditor, not software. Audit efficiency is improved because fewer audit resources (the number of auditors and/or audit time) are required to carry out the same procedures.

DR. PASCAL A. *BIZARRO is an assistant professor of accounting in the department of accounting and management information systems at Bowling Green State University, Ohio. He is a certified information systems auditor. He has received his master and doctorate degrees in Accounting Information Systems from The University of Alabama. He has published a number of papers in leading journals such as* The CPA Journal and Compendium of Classroom Cases and Tools for AIS Applications. *He can be contacted at* pbizarr@bgsu.edu.

DR. ANDY *GARCIA is an associate professor of accounting in the department of accounting and management information systems at Bowling Green State University, Ohio. He is a Colorado CPA. He holds Master of International Management degree from the American Graduate School of International Management (Thunderbird) and a doctorate degree from Texas Tech University. He has published papers in* Internal Auditing *and* The CPA Journal.

EXHIBIT 1 Sample data model



Comments:
(pk) and "+" next to the field(s) that play the role of the primary key.
1..M represents one-to-many relationship between two tables.

Data extraction and analysis software are effective and efficient at importing and investigating large amounts of data for internal, external, and forensic audits.[1] The results of a recent Institute of Internal Auditors (IIA) survey were used to select Microsoft Access, which is one of the most frequently used data extraction and analysis software packages.[2] Although Microsoft Excel is a popular software package among accountants and auditors, it is not optimized for database functions, therefore, is not appropriate for the procedures in this article. Most audit departments and firms should already have Microsoft Access. For those that don't, it is easily available and considerably less expensive than other specialized audit software packages such as ACL and IDEA.

This article first discusses the software interface environment and the data import process, then moves to recurring data manipulation procedures (e.g., filters, computed fields, and sorting). This is followed by commands to verify data completeness (batch controls using count, total, and statistics commands). Based on prior surveys, one of the main obstacles to using software in audit procedures is getting clean (error-free) data in computer-readable form, so the importance of data extraction and data verification should not be overlooked. The article ends with procedures to identify duplicate transactions and missing items along with some comments regarding software selection.

Software interface environment and the data import process

The first step is to load the data into the selected software.[3] The data needed for the audit is usually specified by the audit manager based on a review of planned audit procedures. Before performing any procedure, it is critical to ensure that the downloaded data are an accurate copy of real accounting data. In addition, the auditor should request a file/table description to form a clear understanding of the data structure. Exhibit 1 presents a sample data model using the entity-relationship diagram annotation, listing the table names along with the field names, primary keys, and the relationships between the entities (e.g. one-to-many). This infor-

mation will help the auditor with upload and table definition. The next sections describe the software's interface and the data import procedures.

SQL queries in Microsoft Access 2007. SQL is the primary language used by numerous software packages to interact with relational databases. Although a standard definition of SQL exists, individual database management systems (DBMS) developers implement SQL in different ways, resulting in some incompatibilities. This article uses the version of SQL implemented in Microsoft Access 2007. By default, Microsoft Access 2007 displays a Query-By-Example (QBE) interface to create queries. QBE can be used to implement simple queries but is inadequate for creating more elaborate queries. To run SQL queries, the following steps are required:

1 Select the *Create* tab. In the *Other* group, select the *Query Design* icon on the ribbon. Do not add any tables to the query. Click the *Close* button in the *Show Table* dialog box.
2 In the *Results* group, select *SQL* to display the *SQL View*.
3 Type the SQL statement in the *SQL View*. Click the *Run* icon (red exclamation mark) to execute the query. The record set resulting from the query is displayed in a *Datasheet View*.
4 In the *Results* group, select *SQL* to return to the *SQL View* to make any necessary changes.

Importing external data in Microsoft Access. Microsoft Access permits importing data sets from a wide variety of sources: other Microsoft Access databases, Microsoft Excel spreadsheets, text files, XML files, ODBC databases (e.g., Microsoft SQL Server), HTML documents, or other files (e.g., dBASE, Paradox, Lotus 1-2-3, etc.). The following steps explain how to import data from a text file (similar steps can be applied to the other source types):

1 Select the *External Data* tab. In the *Import* group, select the *Text File* option to start the *Get External Data* wizard.
2 Locate and select the data file to import.
3 Click the *OK* button to accept the default option to import the data into a new table in the current database.
4 Select either the delimited or the fixed-width option based on how the data is organized and click the *Next* button.
5 Check the *First Row contains Field Names* box if the first row of the data file contains field names and click the *Next* button.
6 For each column, make any necessary changes (e.g., field name, data type) and click the *Next* button.
7 Specify the field(s) to play the role of primary key for the table. Microsoft Access has the option to add a new field with AutoNumber as data type for the primary key field. The authors advise to use this option especially if the objective later is to identify fictitious transactions (e.g., foreign keys with no corresponding primary key values). Click the *Next* button.
8 The last step is to specify a name for the new table. Click the *Finish* button to import the data.

The import procedure can be saved by creating a Microsoft Outlook 2007 task. This allows running the import specification at a future date or at fixed intervals such weekly or monthly. This feature helps automate the audit process and may help an organization with the implementation of continuous auditing. Before running any audit procedure, the tables need to be opened in Datasheet view and compared with the source file to make sure the data import process appears to be successful.

MICROSOFT ACCESS PERMITS IMPORTING DATA SETS FROM A WIDE VARIETY OF SOURCES.

Filters, computed fields, and sorting

Filters, additional field computations, computation of existing numerical fields for accuracy checking, and record sorting (in ascending or descending order) are procedures often used in addition to or as part of other audit tests. Examples include sorting prior to judgmental sample selection or manual calculation of sub-totals to compare and reconcile with an account total. The following sec-

tions describe these techniques starting with filters.

Filters. Applying filters to a data set will only display the rows that meet certain specified criteria/conditions and hides rows and/or columns that do not. Criteria are conditions to limit the number of records as output of a filter or a query. Filtered records can be copied, edited, formatted, and/or printed.

Database users employ Data Query Language (DQL) commands to retrieve data from a database and manipulate data to produce useful information. The primary DQL command is SELECT. A query is a request for information from the database implemented using the SELECT command. A query retrieves records from one or more tables and lists the records in a recordset. A recordset is similar to a table, except a recordset may consist of data from multiple tables, and the DBMS does not allocate space in the database for the recordset. The recordset only exists while the query is open. The basic syntax of the SELECT command is as follows: SELECT field1, field2, . . . FROM table;. The WHERE clause selects records based on specific characteristics defined in an expression. The following SQL statement creates a recordset of Accounts Receivable transactions with an amount over 1,000 dollars:

SORTING RECORDS IS AN INTEGRAL PART OF ANY DATA ANALYSIS.

```
1 SELECT *
2 FROM AR_TRANS
3 WHERE Amount > 1000;
```

The SELECT clause in line 1 lists the different fields to be included in the recordset. The asterisk (*) is used instead of the field list to list all fields in the table. Line 2 after the FROM clause specifies which table is used to retrieve the desired records. Line 3 states the criteria each record needs to satisfy to be part of the result. To create more elaborate conditions using multiple fields, the AND and OR operators can be used. The AND operator has precedence over the OR operator unless overridden using parentheses.

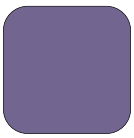
Computed Fields. To check for mechanical accuracy, it is necessary to create a new field (column) and assign an expression to define the formula using other fields and compare the new computed field values with the existing field. Any difference(s) will need to be reconciled. SQL provides the capability to perform arithmetic operations as part of queries. SQL supports typical arithmetic operators, including addition (+), subtraction (-), multiplication (*), and division (/). SQL performs mathematical operations using the standard order of operations (multiplication and division first, followed by addition and subtraction) unless the user overrides this precedence by grouping operations using parentheses. Following is the SQL statement to compute the net pay based on gross pay and tax along with any differences between the computed net pay and the one provided by the client:

```
1 SELECT (Gross_Pay-
  Tax_Amount) AS Com-
  puted_NetPay, Net_Pay
2 FROM PAYROLL
3 WHERE (Gross_Pay-Tax
  _Amount)<>Net_Pay;
```

Notice that we cannot use the new computed field label ("Computed_NetPay") in the WHERE clause where the condition is specified (return only the records with incorrect computation for the net pay).

Sorting. Sorting records is an integral part of any data analysis. An auditor might need to get a list of customer names in alphabetical order or obtain a list of products from highest to lowest sales price. Sorting records in ascending or descending order helps the auditor to look at the data in different ways.

The ORDER BY clause specifies the sort order for records in the recordset. The default sort order is ascending (ASC). The DESC parameter changes the sort order to descending. If multiple fields appear in an ORDER BY clause, the software performs the sort from left to right. The following shows the SQL statement to list all customers with a credit limit over



SOME BATCH CONTROL TOTALS MAY BE "MEANINGFUL".

1,000 dollars sorting numerically these customers:

```
1     SELECT *
2 FROM CUSTOMERS
3 WHERE Credit_Limit > 1000
4 ORDER BY Customer_No.;
```

If you choose to sort on more than one key field, you must select the key fields in the order in which you want the sorting to occur (for example, enter the primary key field first, the secondary key field second, and so on). The Sort command sorts on any number of key fields as long as the combined length of the fields being sorted does not exceed 255 characters. The ascending and descending sort order specifications are specific only to a particular field and can be mixed in one Sort command (for example, account number ascending, transaction amount descending).

Batch controls

Batch controls are completeness checks built into electronic data processing systems and applied to batches of transaction data during the input stage. Batch control totals can involve one or both of the following: 1) record counts, and 2) totals of the values in selected fields within each record. Some batch control totals may be "meaningful".[4] In addition to their control purpose, those totals have another meaning for an accountant or auditor (e.g., account totals may appear on the financial statements). Others are meaningless totals referred to as hash totals (e.g., total customer number). To verify that the entire data set has been uploaded into the data analysis software, the system can compute a series of control totals using one of the following procedures: record count, financial/hash totals, and statistics. The computed totals should be compared with those provided by the client(s). Any differences should be investigated and reconciled.

SQL provides a set of aggregate functions that operate on multiple records. Common aggregate functions include average (AVG), count (COUNT), maximum (MAX), minimum (MIN), summation (SUM), and Standard Deviation (StDev or StDevp). The Standard Deviation aggregate function returns estimates of the standard deviation for a population (StDevp) or a population sample (StDev). The following presents queries to compute summary statistics. The first query returns the invoice amount total, average, standard deviation, and the number of invoices for the AP_TRANS table. Aliases (AS) are used to name the expressions. Brackets are used only when multiple words are used for the naming scheme. The second query displays the 5 highest amounts followed by the third query to display the 5 lowest invoices. The TOP *n* keyword is used to return the first *n* records in the table.

```
1 SELECT
  SUM(Invoice_Amount) AS
  Total,
  AVG(Invoice_Amount) AS
  Average,
2 COUNT(Invoice_Amount) AS
  [Count],
  StDev(Invoice_Amount) AS
  [Standard Deviation]
3 FROM AP_TRANS;
```

```
1 SELECT TOP 5
  Invoice_Amount
2 FROM AP_TRANS
3 ORDER BY Invoice_Amount
  DESC;
```

```
1 SELECT TOP 5
  Invoice_Amount
2 FROM AP_TRANS
3 ORDER BY Invoice_Amount
  ASC;
```

To apply the statistics to only positive or negative amounts, you can add a condition to filter the records (e.g., WHERE Invoice_Amount > 0; < 0; = 0).

Missing and duplicate items

When an auditor has a data set of items that contain pre-numbering for internal control purposes, such as checks or invoices, a search for missing and duplicate items is a logical first step. As with earlier procedures, searching for missing

and/or duplicate transactions helps the auditor establish or reject the completeness and accuracy (including uniqueness test) assertions. Duplicate or missing items may be evidence of weak internal controls, over- or under-stated account balances, or even fraud.

Rather than attempting to exhaustively examine missing and duplicate items in every account balance, a very brief example using duplicates in accounts payable can be used. Duplicates in accounts payable records may indicate duplicate payments, invoices or vendors. Although duplicate payments are typically less frequent than pricing or compliance errors, duplicate payments typically get more attention, perhaps because this is an "obvious" error that should be detected by internal controls before payment is made. Duplicate payments, invoices, or vendors may indicate a need for changes in the system of internal control, or other changes in the system used to process payments.

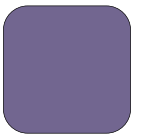
Although the tests to find missing and duplicate items may appear highly automated, auditor judgment is still required to determine what, exactly, constitutes a "duplicate" and how to deal with them. For example, a transposed invoice number may be entered once (incorrectly) then entered again (correctly). Additionally, the output of the software must be carefully examined in conjunction with the system of internal control to identify the underlying cause of the errors, so that appropriate corrective action can be taken. The rest of the article details procedures to find gaps and is followed by the procedures to find duplicates.

Missing items (gaps). In SQL, missing items can be identified by using a correlated query and the EXISTS operator combined with the NOT operator. The template to determine the missing item ranges is as follows.

```
1 SELECT A.Field_Name AS
  [Gap Start (Exclusive)],
  MIN(B.Field_Name) AS
  [Gap End
2 (Exclusive)],
  (MIN(B.Field_Name)-1)-
  A.Field_Name AS [Number
  of Missing Items]
3 FROM Table_Name AS A,
  Table_Name AS B
4 WHERE NOT EXISTS (SELECT
  * FROM Table_Name AS C
  WHERE A.Field_Name+1 =
5 C.Field_Name)
6 AND B.Field_Name>A.Field
  _Name
7 GROUP BY A.Field_Name
```

Line 1 and 2 specify the SELECT clause to display lower gap range and the upper gap range (exclusive) along with the number of missing items. In line 4, the NOT EXISTS condition returns all records from a copy of the original table where there are no records in another copy of the original table for the given record ID. The auditor can replace *Field_Name* with the name of the field to apply the gap search on and *Table_Name* with the name of the table containing the field of interest. The example used to illustrate the coding for finding gaps uses data from an INVOICES table.

```
1 SELECT A.Invoice_No AS
  [Gap Start (Exclusive)],
  MIN(B.Invoice_No) AS [Gap
  End
2 (Exclusive)],
  (MIN(B.Invoice_No)-1)-
  A.Invoice_No AS [Number
  of Missing Items]
3 FROM INVOICES AS A,
  INVOICES AS B
4 WHERE NOT EXISTS (SELECT
  * FROM INVOICES AS C
  WHERE A.Invoice_No+1 =
5 C.Invoice_No.)
6 AND B.Invoice_No.>A.
  Invoice_No.
7 GROUP BY A.Invoice_No.
```



AUDITORS CAN USE THE CODE TO FIND DUPLICATES IN A SEQUENCE.

Duplicates. The duplicates audit procedure is used to find duplicate records with the same values, typically document or record numbers. Auditors can use the code to find duplicates in a sequence of numbers or duplicated transaction dollar amounts. Additionally, duplicate vendor numbers in a vendor master file table could be detected. The following SQL statement presents a template to identify duplicate records:

```
1 SELECT Field1_Name,
  Field2_Name, . . . ,
  COUNT(Field1_Name) AS
  Count
2 FROM Table_Name
3 GROUP BY Field1_Name,
  Field2_Name, . . .
4 HAVING
  COUNT(Field1_Name)>1 AND
  COUNT(Field2_Name)>1 AND
  . . .
```

The auditor can replace *Field1_Name, Field2_Name, . . .* with the field(s) name to apply the duplicate test on and *Table_Name* with the name of the table containing the field(s) of interest. The COUNT function is an aggregate function which returns the number of records within a subset. The GROUP BY clause and the aggregate function work together where the GROUP BY clause defines the subset. To specify a condition clause on the aggregate function, HAVING has to be used after the GROUP BY clause. Aliases cannot be used to reference the function within the HAVING clause. The following SQL statement shows how to generate a list of all the employee numbers that are recurring more than once inside the table PAYROLL indicating duplicate paychecks:

```
1 SELECT Emp_No. AS
  [Duplicate Employee],
  COUNT(Emp_No.) AS Count
2 FROM PAYROLL
3 GROUP BY Emp_No.
4 HAVING COUNT(Emp_No.)>1
```

Selecting the appropriate data analysis tool

Nowadays, it is crucial for both auditors and IT professionals to have a thorough understanding of computer-assisted audit techniques and tools (CAATTs). It is very important to know where and when to apply them. Professional organizations such as ISACA provide guidelines to help auditors (e.g., ISACA G3: Guideline on Computer Assisted Audit Techniques). An auditor should weigh the costs and benefits of any data analysis tool before going through the process of purchasing or developing them in house. Issues to consider include: ease of use, training, maintenance, flexibility of uses, installation, processing efficiencies, data import, integrity of imported data, obtaining permission to install the software on the client servers, software reliability, confidentiality of the data, and reporting capability such as logs.[5]

The purpose of this article is to help auditors and IT professionals use commonly available software, Microsoft Access, to automate various audit procedures including data extraction, summary analysis, and the detection of missing and duplicate items in an audit sample. Auditors and IT professionals may find the Pratt (2004) guide to SQL[6] useful for improving their technical skills. ■

NOTES

1 Gupta, Mudit. 2007. Data Analysis: Tools and Techniques. The Trusted Professional 10(1). http://www.nysscpa.org/trustedprof/107/tp10.htm (accessed September 11, 2009).

2 Gray, Glen L. 2006. An Array of Technology Tools. Internal Auditor 63(4): 56-62.

3 Sayana, S. Anantha. 2003. Using CAATs to Support IS Audit. Information Systems Control Journal 1. http://www.isaca.org/ (accessed January 24, 2010).

4 Daintith, John. 2004. Batch control. A Dictionary of Computing. Encyclopedia.com. http://www.encyclopedia.com/doc/1O11-batchcontrol.html (accessed December 14, 2009).

5 ISACA. 2009. CISA Review Manual 2010. Rolling Meadows, IL: ISACA.org.

6 Pratt, Philip J. 2004. A Guide to SQL. Boston, MA: Course Technology.