

# Database Updates in Access

Laurence R. Paquette (E-mail: lpaquette1@aol.com), Westfield State College

## Abstract

*Many of the Accounting Information Systems (AIS) courses now include database technology as an integral part of the course. In most cases this technology is implemented using a relational database, and in most cases Microsoft Access is used as a vehicle for instruction on this topic. After a quick overview/review of Access, this paper focuses on the different meanings of the term update, and more importantly, looks at how one can update a field in a record such as items on hand or a customers' balance that most if not all current texts fail to describe. An Access Basic Event handling procedure that is contained in a form will be used to accomplish the update.*

## Introduction

Upon entering Access one is presented with a dialog box called the database container that has six tabs. Four of these tabs, Tables, Queries, Forms, and Reports will be examined here with the focus being on the Forms tab and how a form can be used to either update a table that is bound to the form, by adding records to it, or by writing Access Basic code that is associated with an event that will trigger the update of a field in another table. The setting that I would like to use for this example is the Cash Receipts cycle.

*Readers with comments or questions are encouraged to contact the author via e-mail.*

Assume that two tables are to be created. The first table Customer has four fields and contains the data as shown in Figure 1.

The first field is the customer number, the second is customer name, the third is debit dollars and represents the amount of dollar sales made to that customer, and the last field is credit dollars which represents the total cash received from that customer. Note that the balance due by the customer can be calculated using the fields debit dollars and credit dollars. Good database design specifies that such fields be omitted from the table reducing the chances of inconsistencies from occurring in the future. In order to produce the balance for each customer one can ei-

Figure 1

Cust. No.	Cust. Name	Debit	Credit
1000	Paquette	\$200.00	\$100.00
2000	Rienzo	\$900.00	\$500.00
3000	Ferris	\$600.00	\$600.00
4000	Knipes	\$100.00	\$125.00
5000	Rathay	\$2,000.00	\$1,500.00
6000	Merlo	\$700.00	\$0.00
7000	Trump	\$800.00	\$600.00

ther produce a report that includes a calculated field called balance or write a query that includes the calculation and print the resulting dynaset.

**Performing Calculations**

*Using Reports*

A report can be built by clicking on the Report tab of the database container and specifying that a new report is to be generated and that data for the report is to be supplied by the table customer and that an Auto report: Tabular is desired.

While in the design mode use the Label button in the Toolbox to add the label Balance to the Page Header and the Text Box button in the Toolbox to add the calculation  $=[\text{DrDollars}] - [\text{CrDollars}]$  in the Detail Area of the form. Next change the format of the calculated field by right clicking on the text box, choosing Properties and change the format to currency. Finally add the word Balances to the Report Header. The resulting report is shown in Figure 2.

by using a variation of Query By Example, called graphical QBE, or by entering SQL statements directly. A query can be defined by clicking on the Query tab of the database container and specifying that a new query is to be generated using the design view and that data for the query is to be supplied by the table Customer. To include a field in a query, the field name is simply dragged from the table box and is dropped on a column of the query grid in the bottom part of the form. This query is a single-table query in that the table Customer will provide all the data needed except for the calculated field that will be called balance. After dragging all the table fields in the Customer table to the first four columns of the query grid write the following expression in the field row cell of the fifth column:

Balance:  $[\text{DrDollars}] - [\text{CrDollars}]$ .  
Switching to the view mode will result in the dynaset shown in Figure 3.

When one constructs a query using graphical QBE as was done here, Access trans-

**Figure 2**  
**Customer Balances**

<b>Cust. No.</b>	<b>Cust. Name</b>	<b>Debit</b>	<b>Credit</b>	<b>Balance</b>
1000	Paquette	\$200.00	\$100.00	\$100.00
2000	Rienzo	\$900.00	\$500.00	\$400.00
3000	Ferris	\$600.00	\$600.00	\$0.00
4000	Knipes	\$100.00	\$125.00	(\$25.00)
5000	Rathay	\$2,000.00	\$1,500.00	\$500.00
6000	Merlo	\$700.00	\$0.00	\$700.00
7000	Trump	\$800.00	\$600.00	\$200.00

This report can be saved as Customer-Balances as it will be used later to verify that the field CreditDollars has been properly updated when the CashReceipts table that will be created in the next section is updated.

*Using Queries*

Queries can be defined in Access either

lates the QBE expressions that are in the Query Design grid into a series of statements in Structured Query Language (SQL). Access then carries out these instructions on any tables that contain fields matching those specified in the query. This implementation is referred to as Embedded SQL. The SQL statements that are generated can be viewed by clicking on the SQL view option found in the View pull down menu.

Figure 3

Cust. No.	Cust. Name	Debit	Credit	Balance
1000	Paquette	\$200.00	\$100.00	\$100.00
2000	Rienzo	\$900.00	\$500.00	\$400.00
3000	Ferris	\$600.00	\$600.00	\$0.00
4000	Knipes	\$100.00	\$125.00	(\$25.00)
5000	Rathay	\$2,000.00	\$1,500.00	\$500.00
6000	Merlo	\$700.00	\$0.00	\$700.00
7000	Trump	\$800.00	\$600.00	\$200.00

For someone familiar with SQL, an alternative approach to generate the same dataset that is shown above is to click on the Query tab of the database container and then specify that one desires to generate a new query using the design view. Next, one would specify that data for the query is to be supplied by the table customer as was done before. Now rather than using QBE one can immediately switch to the SQL view and enter the following SQL statement:

```
SELECT Customer.CustNo, Customer.Cust-
Name, Customer.DrDollars, Customer.CrDol-
lars, [DrDollars]-[CrDollars] AS Balance
FROM Customer;
```

The SELECT statement is at the heart of SQL and is usually the first word in an SQL statement that returns records to a query result set. The expressions that follow specify the fields involved in the query. Fields are identified by the name of the table followed by a period and the name of a field. SQL is introduced here because an Update SQL statement or command will be used later to accomplish the objective of updating a field in a record by including an SQL command in an event handling procedure contained in a form.

### Performing Updates

Much confusion can occur over the use of the term update when used in conjunction with databases. Conventionally, one thinks of an update as updating a field in a record such as a

customer's balance or the number of units on hand. However, in a database context this term is also used to refer to the adding of a row to a table. In a file management setting this process was referred to as appending or adding a record to a file and not updating. Both procedures, adding records to a table, and updating fields within a record, are very common and the student should be familiar with how to accomplish both procedures. Unfortunately most if not all of the AIS texts and many of the database texts only illustrate the first procedure. Before looking at how both types of updates are accomplished a second table called Receipts will be added to our database. This table also has four fields and does not contain any data at this point in time. The four fields are as follows: RemitNo, Date, CustNo, and Amount. Whenever a cash payment is made two actions should occur. First a record should be added to the Receipts table and second the CreditDollars field in the Customer table should be updated to reflect the current cash payment. The first action is quite easy to accomplish and is widely discussed, the second action can prove to be more difficult and is rarely treated.

### Adding a Record to a Table

A form can be used to display information from one or more tables. A form can also be used to add records to a table. Note that up to this point we have made use of the Table, Query, and Report tabs of the database container. We will now make use of the Forms tab.

Figure 4

RemitNo	Date	CustNo	Amount
1	1/2/98	1000	\$50.00
2	1/2/98	5000	\$200.00
3	1/2/98	2000	\$100.00
			\$0.00

In this section we will begin by creating a form using the table Receipts. Press the Forms tab in the database container, and click on the New button. Select AutoForm:Tabular and specify that the table Receipts is to be used as the data source. This associates or binds the form with the Receipts table. Display the form in Design View and right-click the label box in the form header and select Properties from the shortcut menu that appears, click the Data Tab and select or enter Yes in the Data Entry box. This indicates that the form is to be used for data entry rather than for the display of information. Return to the Form View and enter the three records shown in Figure 4.

in most AIS texts. What is desired to occur is that after the Receipts table is updated (i.e. a record is added to the table) one would like to update the CrDollars field for that customer in the Customer table. One can cause this to occur by returning to the design view of the form just created and right clicking on the Text Box Amount in the Detail section of the form. Select Properties from the shortcut menu that appears, click the Event Tab, place the cursor in the After Update box, and select an Event procedure from the pull down list. To define the Event procedure click on the build button. This will place you on a screen where you can define the procedures you want executed on the Event After Update. The screen will appear as illustrated below.

```
Option Compare Database
Option Explicit

Private Sub Amount_AfterUpdate()

End Sub
```

Microsoft Access automatically posts any newly entered data to the database table Receipts, when one moves from the current record or closes the form. This can be confirmed by closing the form and viewing the table called Receipts.

#### *Updating a Field in a Record*

This is the procedure that is not covered

The following commands must be entered where the cursor is positioned (i.e. before the End Sub command).

Of the three commands entered the first and last commands simply turn off and on the warning message that appears telling you that you are about to update a record and that the Undo command cannot be used to reverse the changes. The middle command will update the

```
DoCmd.SetWarnings (False)
DoCmd.RunSQL "Update Customer
SET Customer.CrDollars = Customer.CrDollars + Forms!Receipts!Amount
WHERE Customer.CustNo = Forms!Receipts!CustNo ;"
DoCmd.SetWarnings (True)
```

CrDollars field in the Customer table. Note that this command must be entered as a single long continuous string, and not as shown here. Once these lines of code have been added it is necessary to click on the compile button on the Access menu. The code should successfully compile and you are ready to see if the form will work properly by performing the updates to the CreditDollars field of Customer.

However before entering any data one should note that the field CustNo on the Receipts form determines which record is updated in the table Customer. A control to help minimize an incorrect entry for this field that would result in no matching value being found in the Customer table is to employ a combo box, which is a combination text box and list box for that field. To create a combo box for the CustNo field go to the design view for the Receipts form and right click on the box and select Change to from the pull down menu that appears, and then change the text box to a combo box. You will note a change in the box's appearance that now includes a pull down arrow. Right click on the combo box and select Properties from the pull down menu. Click the All tab and make the following changes: (a) In the box labeled Row Source Type select the Table/Query option; (b) In the box labeled Row Source enter the following: Select Customer.CustNo from Customer; and (c) In the Limit to List box select Yes. This will result in data entry for this field to be limited to valid customer numbers that are found in the table

Customer.

Enter the same three transactions as shown in Figure 4. Note before doing this one should go back to the Receipts table and delete the three records that were just entered given that we are about to reenter them. To verify that the CreditDollars field has been updated in the Customer table we can use the report Customer-Balances that was developed and saved earlier. Using this report one can see by referencing Figure 5 that the fields have been updated for the entries made and that the customer balances reflect this change.

**Summary**

This paper provides a quick review of some procedures that can be accomplished using Microsoft Access in the context of it being applied in an AIS course. Four of the six tabs presented in the database container dialog box are discussed namely; Tables, Queries, Forms, and Reports. The process of using reports or queries to provide values for calculated fields that have been omitted from tables as a result of good database design is discussed first. Queries in Access can be defined using either graphical QBE or by entering SQL statements directly. Both methods are introduced. This introduction to SQL lays the groundwork for the main focus of the paper and that is the distinction between updating a table and updating fields contained in select records within a table. Forms are intro-

**Figure 5  
Customer Balances**

<b>Cust. No.</b>	<b>Cust. Name</b>	<b>Debit</b>	<b>Credit</b>	<b>Balance</b>
1000	Paquette	\$200.00	\$150.00	\$50.00
2000	Rienzo	\$900.00	\$600.00	\$300.00
3000	Ferris	\$600.00	\$600.00	\$0.00
4000	Knipes	\$100.00	\$125.00	(\$25.00)
5000	Rathay	\$2,000.00	\$1,700.00	\$300.00
6000	Merlo	\$700.00	\$0.00	\$700.00
7000	Trump	\$800.00	\$600.00	\$200.00

duced next and they can be used to either display information contained in a table or for data entry. By binding a form to a table, each time a record is entered it is added to or updates a table. Finally Access Basic Code, which is triggered by an event, is written and associated with a text box on the data entry form. This code or event handling procedure includes an Update SQL action command that allows one to update a field in another table. This last topic is an omission in most AIS texts that include database technology as part of the course and is illustrated in this paper. 