JOURNAL OF EMERGING TECHNOLOGIES IN ACCOUNTING
Vol. 13, No. 2
Fall 2016
pp. 195–213

American Accounting Association
DOI: 10.2308/jeta-51597

International AC: An Education Case on Continuous Monitoring SQL Server Data with ODBC-Linked Tables in Microsoft Access

Sherwood Lane Lambert
Jon Holladay
University of West Florida

Dawna M. Drum
Western Washington University

ABSTRACT: This education case teaches students Continuous Auditing/Continuous Monitoring (CA/CM) using Open Database Connectivity (ODBC) connections from students' Microsoft Access (Access) to tables in Microsoft SQL Server (SQL Server). First, the institution's database administrator (DBA) imports the student tables provided with the Teaching Notes into the institution's SQL Server with read-only access. In Access, students create ODBC connections to the SQL Server student tables. Then, students create Access queries using the ODBC-linked tables to monitor for anomalies. During a scheduled week, the DBA, in coordination with the faculty member, enters anomalies in the SQL Server student tables at random times, and students run their Access queries to identify the anomalies.

Keywords: continuous auditing; continuous monitoring; Microsoft SQL Server; Microsoft Access Database; Open Database Connectivity (ODBC); principle of least privilege; Relational Database Management System (RDBMS).

THE CASE

International AC is a corporation operating in Houston, Texas that specializes in selling industrial air conditioners to businesses within the United States and throughout the world. You recently joined International AC as the Director of Marketing. You have just learned that write-offs of accounts receivables have been increasing and you are concerned that your sales managers may be authorizing sales on account that substantially exceed customers' credit limits. Even more troubling, one of your employees has reported an incident of a salesperson having sold an air conditioner for far less than the standard sales price established for the unit. You worry that one or more employees may have perpetrated fraud by selling air conditioners to their friends or relatives at a cut-rate price. Therefore, you want to monitor sales data on an ongoing basis, searching for air conditioners sold at substantially less than standard unit prices or at sale amounts that substantially exceed customers' credit limits. You have heard about continuous auditing and continuous monitoring (CA/CM), and you think this may be a good option for you, but you have much research to do first.

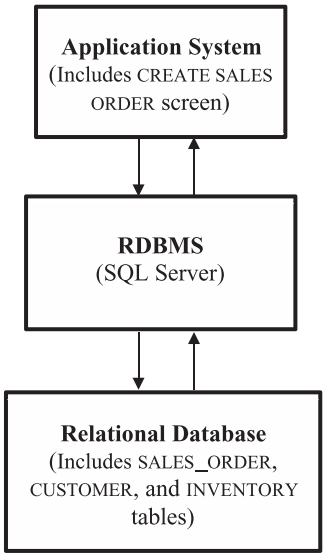
Database Basics

You begin by discussing your requirements with the Director of Information Technology (IT) at International AC and receiving her approval to proceed. You then schedule a meeting with Jon Hampton, the Database Administrator (DBA) for International AC. Jon begins by explaining that International AC's enterprise business system is a client-server system that uses Microsoft SQL Server (hereafter, SQL Server) as its relational database management system (RDBMS). SQL Server can store

Editor's Note: Accepted by Hui Du.

Submitted: June 2015
Accepted: September 2016
Published Online: September 2016

FIGURE 1
Three-Tier Client-Server System



virtually unlimited amounts of data and support any number of users. Many business and government entities use SQL Server as the RDBMS for their Enterprise Resource Planning (ERP) system. Figure 1 depicts International AC's three-tier, client-server system. Marketing employees use the CREATE SALES ORDER screen that is part of the company's application system to access and update data stored in the SALES_ORDER, CUSTOMER, and INVENTORY tables that exist along with other tables in the relational database.

Since you are familiar with using queries in Microsoft Access (hereafter, Access) for auditing and analyzing data, you want to use Access for your CA/CM. Jon explains that Access can connect to SQL Server database tables directly through Open Database Connectivity (ODBC) links. Without ODBC, you normally would need to obtain a download of the data from IT and then import it into tables in your Access database. If your SQL Server database contains many large tables, you may not be able to import them all into a single Access database because the maximum size of an Access database is two gigabytes. One advantage of using ODBC is that the two-gigabyte size limit does not apply to ODBC-linked tables in SQL Server. Another advantage of ODBC is that you are accessing the table as it currently exists in SQL Server. These two facts about ODBC are fundamentally important when considering using Access with ODBC for continuously auditing or monitoring data in SQL Server tables. Jon mentions that most client-server systems use either SQL Server or Oracle Database as their RDBMS, and it is also possible to establish ODBC connections from Access to tables in Oracle Database.

Jon says it is possible to grant you direct access to specified tables in SQL Server, but your access must be authorized and read-only. You question Jon about the read-only restriction, since, as Director of Marketing, you already have access to screens in the production application system that add, change, and delete sales orders. Jon explains that each system tier has its set of IT controls for ensuring the security and integrity of the production system. Directly accessing the relational database tables bypasses many of the client-server system's IT controls. For example, directly accessing tables bypasses the application system's logical segregation of function control that limits each user's access to only those screens and transactions that the user needs to perform his/her job. Also, with direct update access to tables in the relational database, a user could bypass online data entry edits in the application system's screens that might otherwise prevent the entry of erroneous data in table fields. Referential integrity controls in the database relationships will not prevent users who have direct update access to tables from changing the values in primary key and foreign key fields that break referential integrity. [Note: A subsequent section of the case discusses referential integrity, primary keys, and foreign keys.] Also, system transaction logs that normally record information about each update transaction would not record changes made by directly accessing the tables and would, therefore, bypass this important detective and corrective IT control.

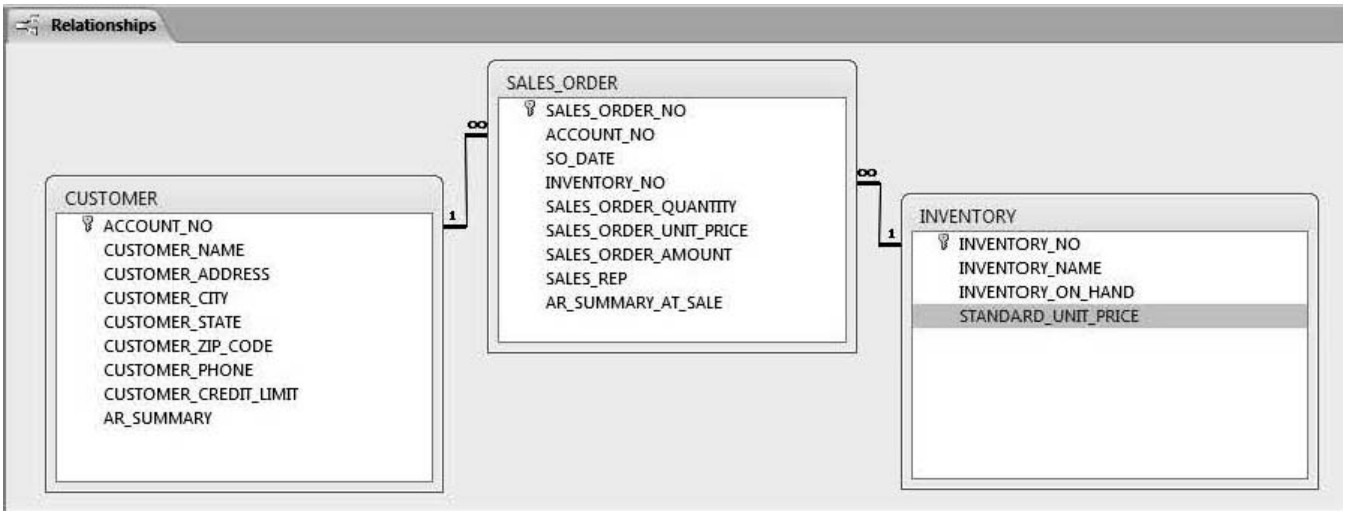
Jon will set up your direct access to SQL Server using the Principle of Least Privilege. Limiting your direct access to read-only on only the tables you require will compensate for bypassing the system controls he mentioned, such as logical segregation of function control in the application system, and will prevent you from entering erroneous data that online screen

edits and referential integrity controls would prevent. Also, limiting your access to read-only will prevent you from entering data in primary and foreign key fields that would break the referential integrity in database relationships.

Marketing Production System Basics

The next day, you meet with Sally Downing, Director of Application Systems, and ask her to provide an overview of the IT that supports sales order processing. Specifically, you ask her to explain the processing that occurs when sales personnel enter sales orders with unit prices that are less than the standard unit prices. Also, you want to understand how the system processes sales orders when the sales amount exceeds the customer's credit limit.

Sally offers to guide you as you enter transactions in a test system in Access. [Note to students: At this point you should open the INTERNATIONAL AC SALES ORDER SYSTEM Access database provided by your faculty member. Using this database, enter the transactions that Sally indicates as she guides you through a demonstration of International AC's sales order processing.] You open the INTERNATIONAL AC SALES ORDER SYSTEM Access database and then click on the tab in the menu named **Database Tools**. When this tab opens, you click the icon labeled **Relationships**. As shown below, your screen displays the portion of International AC's relational database that supports sales order processing.



Primary Keys and Referential Integrity

As Sally explains, the symbol resembling a key to the left of ACCOUNT_NO shows that it is a primary key in the CUSTOMER table. The ACCOUNT_NO field is also the foreign key in the SALES_ORDER table. The line connecting these two fields is the database relationship connecting these two tables. There can be many SALES_ORDER entries, but each row in SALES_ORDER records a sale to only one customer. Hence, the relationship between the CUSTOMER and SALES_ORDER tables is a one-to-many relationship (the ∞ symbol represents *many*). Sally instructs you to double-click anywhere on this line, and the **Edit Relationships** window opens. Sally points out the check mark by "Enforce Referential Integrity" that indicates the relationship enforces referential integrity. In this relationship, enforcing referential integrity means any account number that you enter in a new sales order record in the SALES_ORDER table must first exist in the CUSTOMER table. You close the **Edit Relationships** window. Similarly, the INVENTORY_NO field is the foreign key in the SALES_ORDER table and the primary key in the INVENTORY table, enabling a one-to-many relationship between the INVENTORY table and the SALES_ORDER table. You close the **Relationships** window.

Marketing System Tables

Next, Sally asks you to open the CUSTOMER table in **Datasheet View** by double-clicking on the table's name. Your screen appears as follows:

CUSTOMER

ACCOUNT_NO	CUSTOMER_NAME	CUSTOMER_ADDRESS	CUSTOMER_CITY	CUSTOMER_STATE	CUSTOMER_ZIP_CODE	CUSTOMER_PHONI	CUSTOMER_CREDIT_LIMIT	AR_SUMMARY
1	Johnson's Retail	153 Main St.	Pensacola	FL	34543	850-333-2321	20000	1000
2	Goodman Industries	334 Broadway	Fort Worth	TX	76123	817-412-2345	30000	3500
3	Mason's Outlets	3876 Sunset Blvd	Los Angles	CA	90007	213-333-2222	40000	0
4	Wilson Restruants	2312 Creighton Dr.	Cleveland	OH	24222	276-434-3443	11000	6000

Because Sally is showing you a test version of the system, there are only a few customers, sales orders, and inventory items contained in the tables. This limited number of records will be sufficient to illustrate the system's functionality.

Next, you change to **Design View** by selecting **Home** from the menu and clicking the **Design View** icon in **View** in the far left of the menu. Then, you click ACCOUNT_NO to select this field. The top half of your screen appears as follows:[1]

CUSTOMER

Field Name	Data Type
ACCOUNT_NO	Number
CUSTOMER_NAME	Text
CUSTOMER_ADDRESS	Text
CUSTOMER_CITY	Text
CUSTOMER_STATE	Text
CUSTOMER_ZIP_CODE	Text
CUSTOMER_PHONE	Text
CUSTOMER_CREDIT_LIMIT	Number
AR_SUMMARY	Number

In the screen above, the key symbol to the left of ACCOUNT_NO specifies that this field is the primary key. The other fields are customer attributes. You close the CUSTOMER table and then open the INVENTORY table in **Datasheet View** as shown below:

INVENTORY

INVENTORY_NO	INVENTORY_NAME	INVENTORY_ON_HAND	STANDARD_UNIT_PRICE
1	AC 2000	96	10000
2	Deluxe Air Conditioner	4	13000
3	Supreme AC Unit	14	15000
4	AC-5000	10	21000

[1] Depending on your version of Access, **Short Text** rather than **Text** may appear as the **Data Type**.

Change to **Design View** so that the top of your screen appears as shown below. The key symbol shows INVENTORY_NO is the primary key. The other fields are attributes of inventory items.

INVENTORY

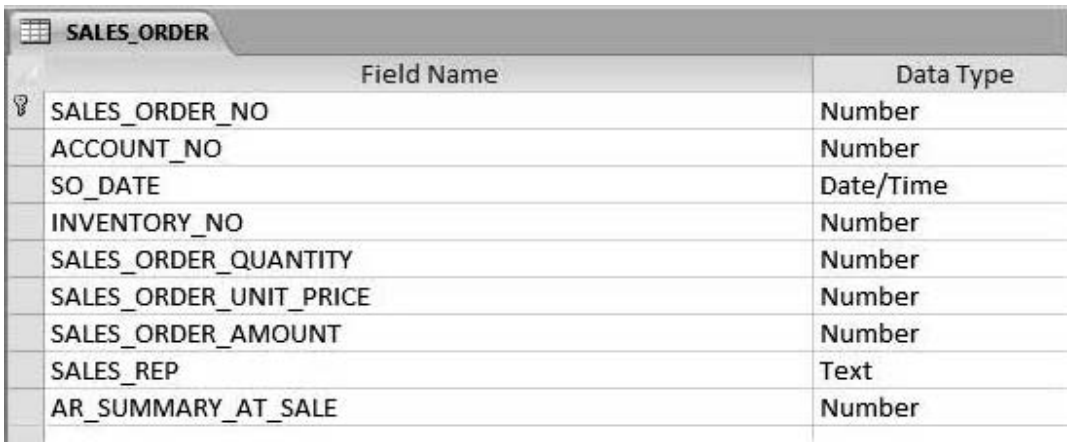
Field Name	Data Type
INVENTORY_NO	Number
INVENTORY_NAME	Text
INVENTORY_ON_HAND	Number
STANDARD_UNIT_PRICE	Number

Close INVENTORY and open the SALES_ORDER table in **Datasheet View** as shown below:

SALES_ORDER

SALES_ORDER_NO	ACCOUNT_NO	SO_DATE	INVENTORY_NO	SALES_ORDER_QUANTITY	SALES_ORDER_UNIT_PRICE	SALES_ORDER_AMOUNT	SALES_REP	AR_SUMMARY_AT_SALE
1	1	9/20/2016	1	2	9000	18000	Stan Johnson	20000
2	3	9/28/2016	3	1	10000	10000	Mark Weaver	40000
3	3	9/29/2016	2	1	16000	16000	Joe Stanley	30000
4	1	9/30/2016	2	1	5000	5000	Mary Boston	11000
5	1	12/17/2016	2	1	2500	2500	Mary Boston	6000

Next, change to **Design View** so that the top of your screen displays as shown below. SALES_ORDER_NO is the primary key, and ACCOUNT_NO and INVENTORY_NO are foreign keys. The other fields are attributes of sales orders. Close the SALES_ORDER table.



SALES_ORDER

Field Name	Data Type
SALES_ORDER_NO	Number
ACCOUNT_NO	Number
SO_DATE	Date/Time
INVENTORY_NO	Number
SALES_ORDER_QUANTITY	Number
SALES_ORDER_UNIT_PRICE	Number
SALES_ORDER_AMOUNT	Number
SALES_REP	Text
AR_SUMMARY_AT_SALE	Number

Forms and Screens

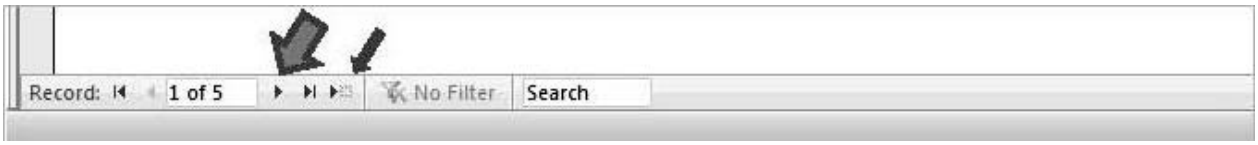
Next, you open the CREATE SALES ORDER form as shown below:

SALES_ORDER

CREATE SALES ORDER

SALES_ORDER_NO	1	SO_DATE	9/20/2016	SALES_REP	Stan Johnson
ACCOUNT_NO	1			INVENTORY_NO	1
CUSTOMER_NAME	Johnson's Retail			INVENTORY_NAME	AC 2000
CUSTOMER_ADDRESS	153 Main St.			INVENTORY_ON_HAND	96
CUSTOMER_CITY	Pensacola			STANDARD_UNIT_PRICE	10000
				SALES_ORDER_QUANTITY	2
CUSTOMER_STATE	FL			SALES_ORDER_UNIT_PRICE	9000
CUSTOMER_ZIP_CODE	34543			SALES_ORDER_AMOUNT	18000
CUSTOMER_PHONE	850-333-2321			AR_SUMMARY	1000
CUSTOMER_CREDIT_LIMIT	20000				

Salespersons use this form (screen) to enter new sales orders. At the bottom of the form shown below, you click on the **Next record** right-pointing arrowhead indicated by the larger arrow to scroll through each of the five sales orders. The smaller arrow indicates the **New (blank) record** arrowhead used when creating new records.



CREATE SALES ORDER is *bound* to the SALES_ORDER table. A form can display and update the data in the fields of its bound table. Also, a form can display and update the data in the fields of the tables related to the bound table in the relational database.

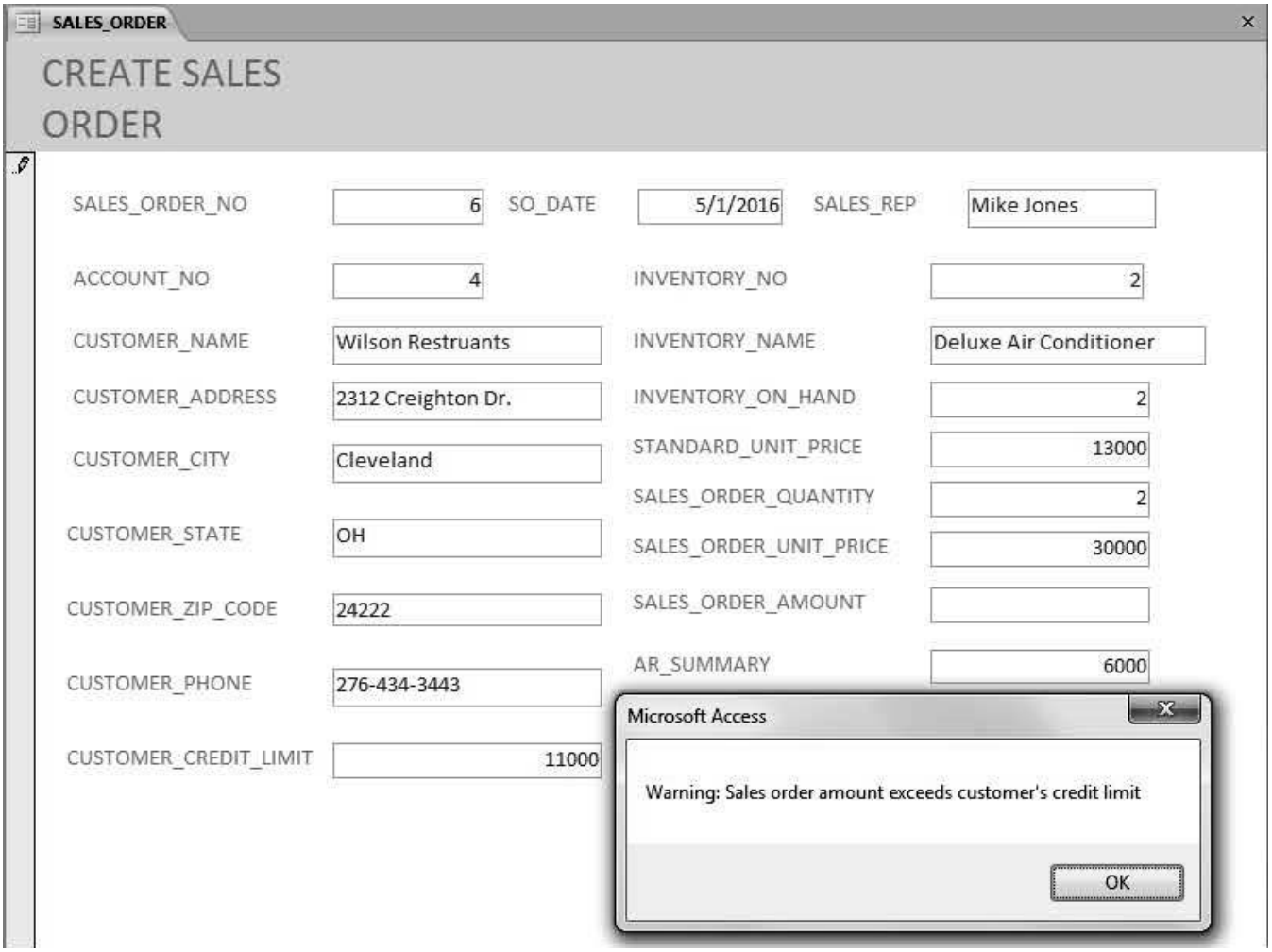
To demonstrate how CREATE SALES ORDER functions, Sally asks you to click the **New (blank) record** arrowhead so that "6 of 6" is showing, and the screen fields are blank. She instructs you to enter the information shown below, pressing the Enter key after each entry.[2]

[2] Rather than pressing the Enter key, you can make the entry by clicking on any other blank field in the form.

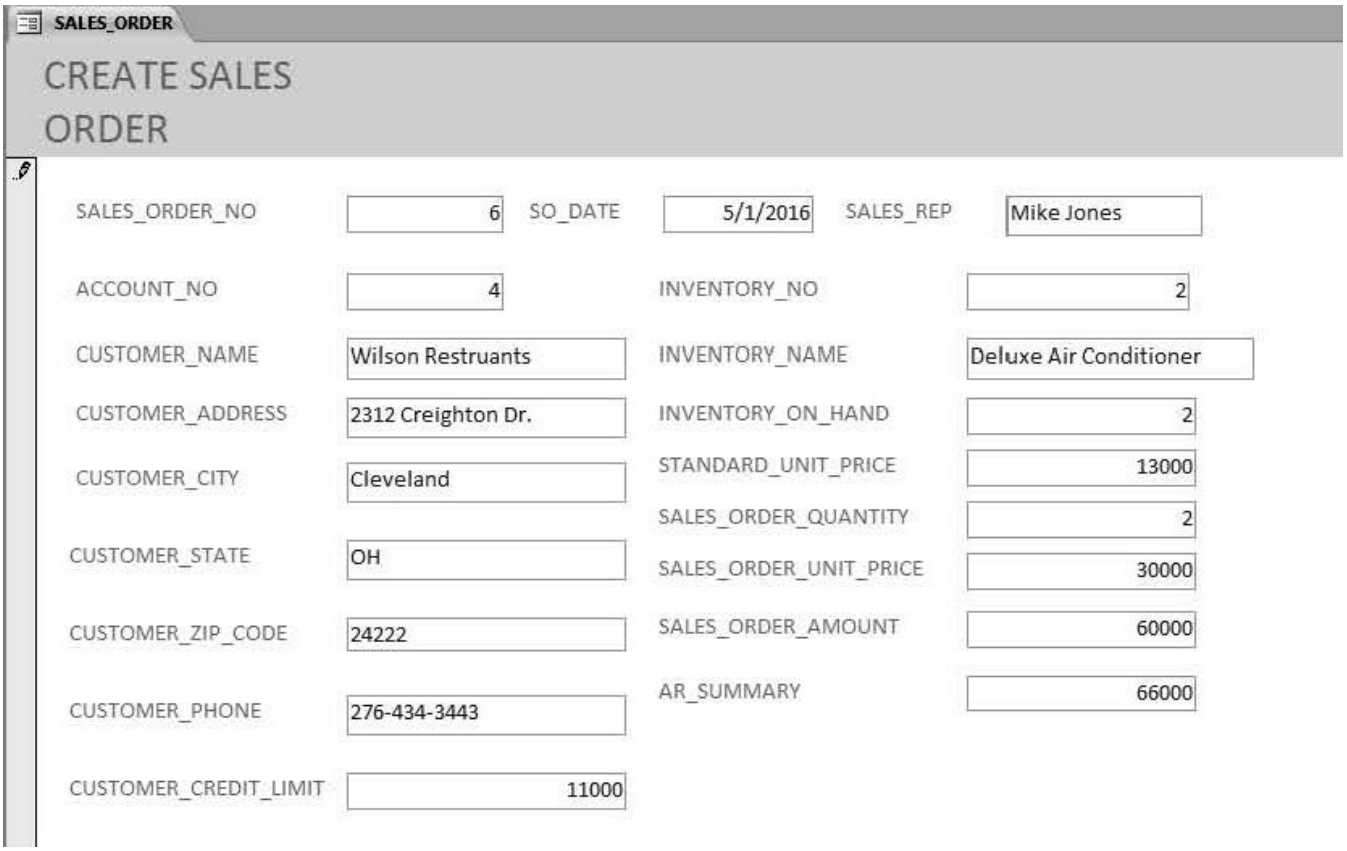
Field Name	Data (press Enter key after entering each value)
SALES_ORDER_NO	6
SO_DATE	5/1/2016
SALES_REP	Mike Jones
ACCOUNT_NO	4

Entering 4 in ACCOUNT_NO and pressing the Enter key causes the form to transmit account number 4 to the RDBMS that randomly accesses 4 in the index on the ACCOUNT_NO primary key. Once found, the RDBMS transmits back to the form the data stored in customer 4's row in the CUSTOMER table. Just by entering account number 4, the system automatically populates the fields for customer 4's name, address, city, state, zip code, phone number, credit limit, and accounts receivables summary.

Next, Sally asks you to enter **2** in INVENTORY_NO and press Enter so that the system automatically inputs the other inventory information for this item into the screen. The system obtains this information for inventory item 2 from the INVENTORY table in the same way the system obtained the information for account number 4 from the CUSTOMER table, as previously described. Next, Sally asks you to enter **2** in SALES_ORDER_QUANTITY and **30000** in SALES_ORDER_UNIT_PRICE and then to press Enter. The system displays a warning message that you have created a sales order that exceeds the customer's credit limit. After repositioning the message box by selecting it and then moving it, your screen appears as follows:



When you click OK on the warning message, the transaction completes. The screen automatically computes SALES_ORDER_AMOUNT shown as 60000 in the screen. At the same time, the screen reduces INVENTORY_ON_HAND from 4 to 2 for the sale of 2 units and increases AR_SUMMARY from 6000 to 66000 to reflect the new accounts receivables balance. Your screen now appears as follows:



Database and Table Processing

AR_SUMMARY displays the total amount of the customer's outstanding accounts receivables due in the accounts receivables subsidiary ledger table. Sally simplified her demonstration by not showing the accounts receivables table in her Access database, but this table exists in the production system. When checking to determine whether a sale will exceed the customer's credit limit, the CREATE SALES ORDER form adds the new sale amount to AR_SUMMARY and compares this sum to CUSTOMER_CREDIT_LIMIT. When this sum exceeds the customer's credit limit, the form displays the error message previously demonstrated. Then, the form moves the value of AR_SUMMARY to AR_SUMMARY_AT_SALE in the new SALES_ORDER record created for the sale. AR_SUMMARY_AT_SALE is the value of AR_SUMMARY just before processing a sale. Then, the form adds the current sale amount to AR_SUMMARY and displays the new value of AR_SUMMARY. In the sections that follow, Sally will explain how CREATE SALES ORDER performs this processing.

Close the CREATE SALES ORDER form and open the SALES_ORDER table. The data that you entered and the amounts computed in the CREATE SALES ORDER form for sales order number 6 now appear in a new row in the SALES_ORDER table as shown below.

SALES_ORDER

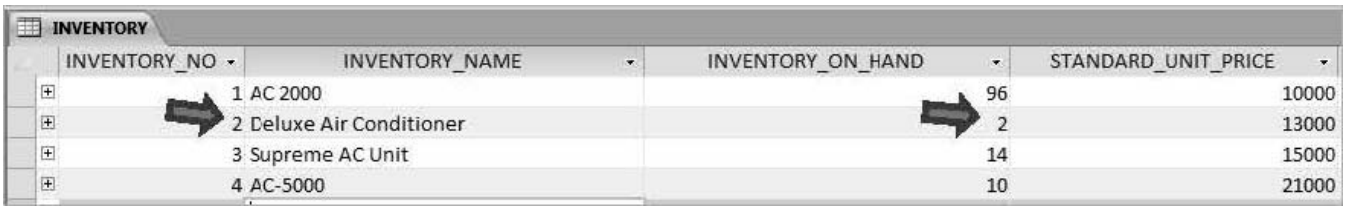
SALES_ORDER_NO	ACCOUNT_NO	SO_DATE	INVENTORY_NO	SALES_ORDER_QUANTITY	SALES_ORDER_UNIT_PRICE	SALES_ORDER_AMOUNT	SALES_REP	AR_SUMMARY_AT_SALE
1	1	9/20/2016	1	2	9000	18000	Stan Johnson	20000
2	3	9/28/2016	3	1	10000	10000	Mark Weaver	40000
3	3	9/29/2016	2	1	16000	16000	Joe Stanley	30000
4	1	9/30/2016	2	1	5000	5000	Mary Boston	11000
5	1	12/17/2016	2	1	2500	2500	Mary Boston	6000
6	4	5/1/2016	2	2	30000	60000	Mike Jones	6000

Close the SALES_ORDER table and open the CUSTOMER table as shown below. The form updated AR_SUMMARY for account number 4 in the CUSTOMER table.

CUSTOMER

ACCOUNT_NO	CUSTOMER_NAME	CUSTOMER_ADDRESS	CUSTOMER_CITY	CUSTOMER_STATE	CUSTOMER_ZIP_CODE	CUSTOMER_PHONE	CUSTOMER_CREDIT_LIMIT	AR_SUMMARY
1	Johnson's Retail	153 Main St.	Pensacola	FL	34543	850-333-2321	20000	1000
2	Goodman Industries	334 Broadway	Fort Worth	TX	76123	817-412-2345	30000	3500
3	Mason's Outlets	3876 Sunset Blvd	Los Angles	CA	90007	213-333-2222	40000	0
4	Wilson Restruants	2312 Creighton Dr.	Clevelard	OH	24222	276-434-3443	11000	66000

Close the CUSTOMER table and open the INVENTORY table as shown below. The form changed INVENTORY_ON_HAND from 4 to 2 for inventory number 2 in the INVENTORY table. Close the INVENTORY table.



INVENTORY

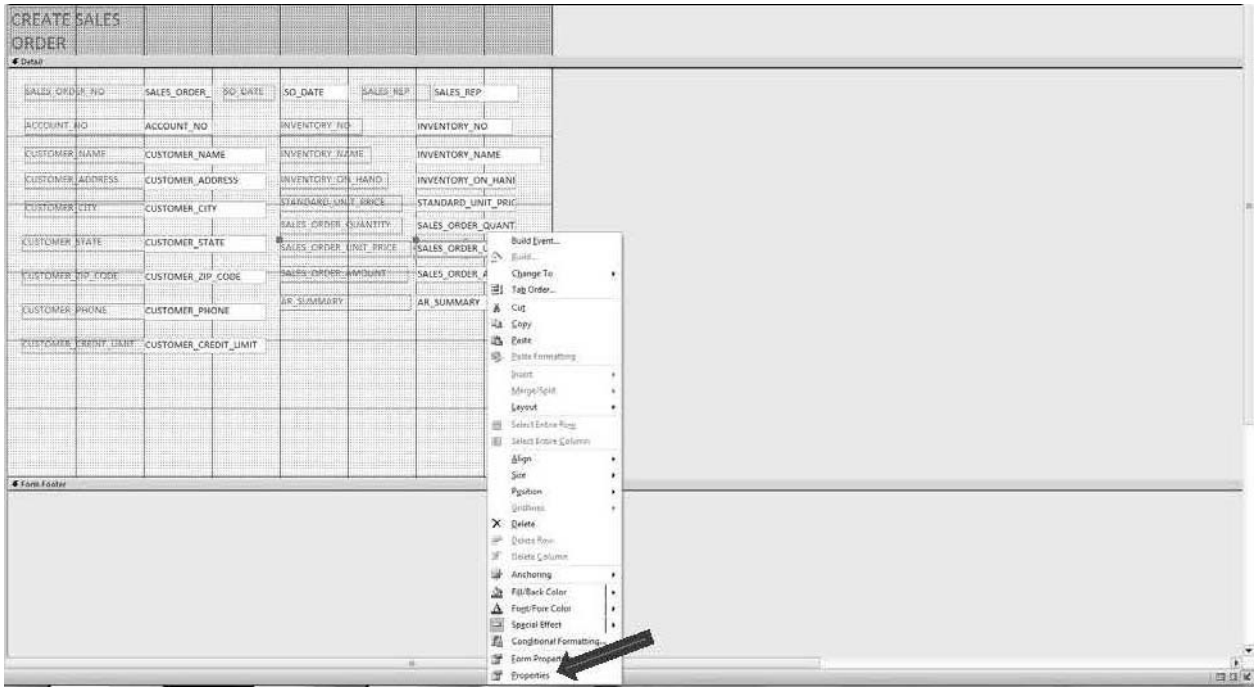
INVENTORY_NO	INVENTORY_NAME	INVENTORY_ON_HAND	STANDARD_UNIT_PRICE
1	AC 2000	96	10000
2	Deluxe Air Conditioner	2	13000
3	Supreme AC Unit	14	15000
4	AC-5000	10	21000

Error Message Processing

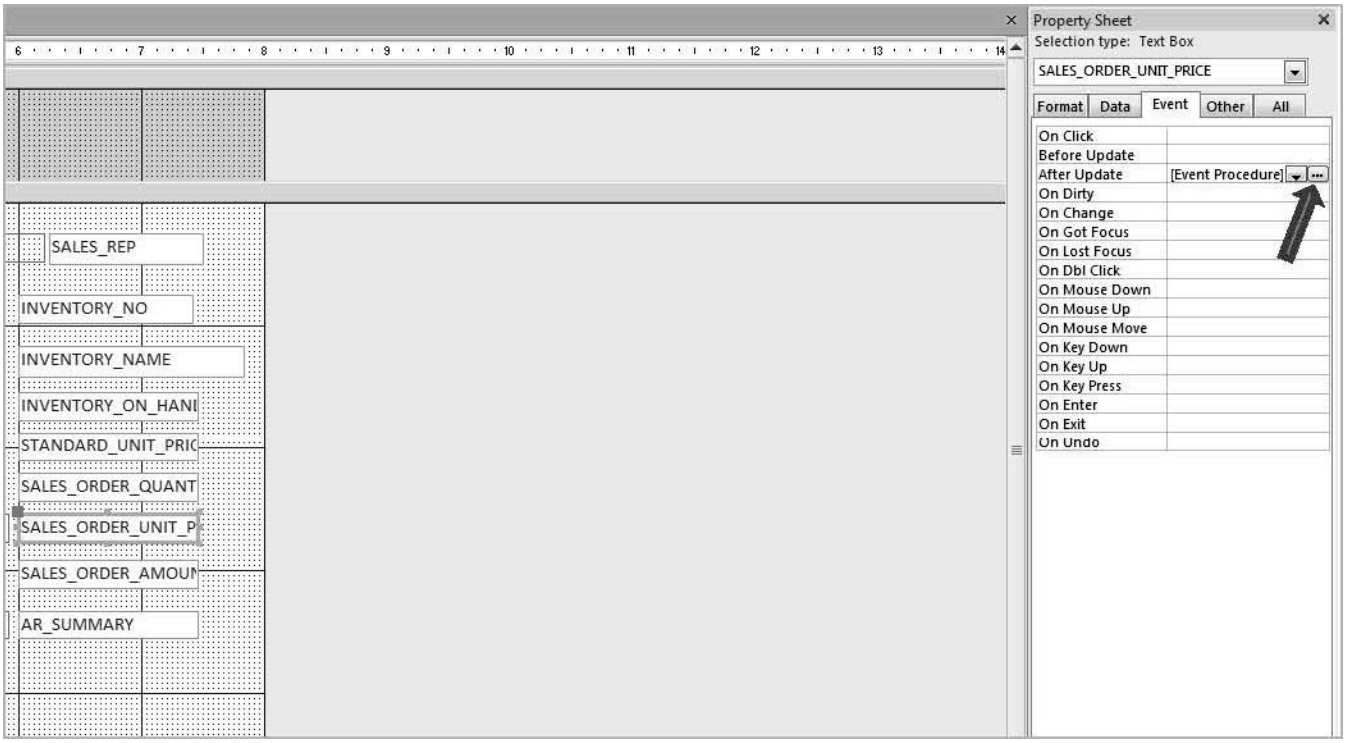
Before processing sales order 6, account number 4 had available credit of $5,000 based on a total credit limit of $11,000 and AR_SUMMARY of $6,000. Hence, the customer had available credit of $5,000 ($11,000 – $6,000). You were able to process a sale for $60,000 when the customer's available credit was $5,000 because the edit is a warning-only edit. The warning message tells the salesperson that the sale, if completed, will exceed the customer's credit limit. However, the edit does not prevent the transaction from processing. To see this edit in the form, perform the following:

- Right-click the CREATE SALES ORDER form's name and select **Design View**
- Select (highlight) the SALES_ORDER_UNIT_PRICE (do not select this field's label)
- Right-click so that a window opens showing **Properties** at the bottom of the list.

At this point, your screen appears as follows:



Select **Properties** and, after the **Property Sheet** opens, select the **Event** tab if not already selected. Click [Event Procedure] directly to the right of **After Update** so that the button with three periods (see the arrow) is to the right of [Event Procedure]. Your screen should appear as shown below.



Now, click (click twice if necessary) on the button with three periods. This action displays an Event Procedure that Sally developed with the Visual Basic for Applications (VBA) programming language. [Note: If your screen does not display [Event

Procedure], click in the cell to the right of **After Update** and then click the button with three periods. In the **Code Builder** window that opens, select **Code Builder** and then click **OK**.] Your screen appears as follows:

```
Option Compare Database

Private Sub SALES_ORDER_QUANTITY_AfterUpdate()

    If [SALES_ORDER_QUANTITY] > [INVENTORY_ON_HAND] Then
         MsgBox "Sales quantity exceeds quantity on-hand."
         [SALES_ORDER_QUANTITY] = 0
    Else
         [INVENTORY_ON_HAND] = [INVENTORY_ON_HAND] - [SALES_ORDER_QUANTITY]
    End If


End Sub


Private Sub SALES_ORDER_UNIT_PRICE_AfterUpdate()


    If [SALES_ORDER_UNIT_PRICE] < [STANDARD_UNIT_PRICE] Then
         MsgBox "Warning: SALES_ORDER_UNIT_PRICE is less than the STANDARD_UNIT_PRICE for this item."
    End If


    If ([SALES_ORDER_QUANTITY] * [SALES_ORDER_UNIT_PRICE]) + [AR_SUMMARY] > [CUSTOMER_CREDIT_LIMIT] Then
         MsgBox "Warning: Sales order amount exceeds customer's credit limit"
    End If


    [AR_SUMMARY_AT_SALE] = [AR_SUMMARY]

    [SALES_ORDER_AMOUNT] = [SALES_ORDER_QUANTITY] * [SALES_ORDER_UNIT_PRICE]

    [AR_SUMMARY] = [AR_SUMMARY] + [SALES_ORDER_AMOUNT]

End Sub
```

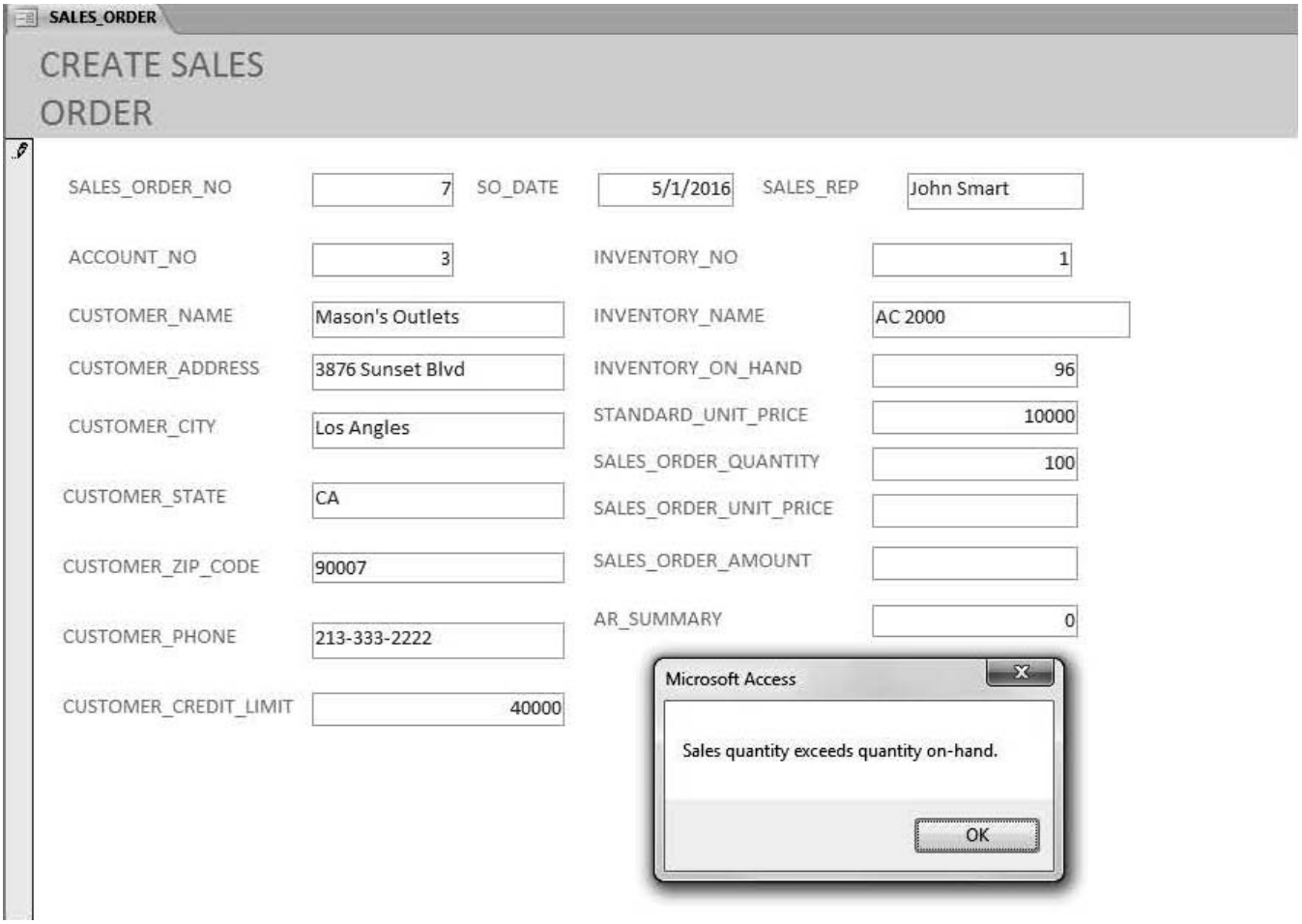
First, look in the bottom half of the code at the procedure that begins with **Private Sub SALES_ORDER_UNIT_PRICE_AfterUpdate()** and ends with **End Sub**. Access processes this procedure each time an online user updates the SALES_ORDER_UNIT_PRICE field in the CREATE SALES ORDER form. This procedure has two IF statements that end with "End If." The first IF statement checks whether the SALES_ORDER_UNIT_PRICE is less than the STANDARD_UNIT_PRICE for the sales item. If true, Access displays the message, "Warning: SALES_ORDER_UNIT_PRICE is less than the STANDARD_UNIT_PRICE for this item." This edit is a warning only because the logic in the IF statement allows the transaction to process even if the sales unit price is less than the standard unit price. The second IF statement checks whether the sale amount plus the customer's total accounts receivables due exceeds the customer's credit limit. If true, Access displays "Warning: Sales order amount exceeds customer's credit limit" but does not prevent processing the sale. Both edits produce a warning message, but neither prevents sale transactions from processing. The procedure moves the value in AR_SUMMARY to the AR_SUMMARY_AT_SALE field in the new record created in the SALES_ORDER table for the sale. Then, the procedure calculates SALES_ORDER_AMOUNT by multiplying the SALES_ORDER_QUANTITY times the SALES_ORDER_UNIT_PRICE. Finally, the procedure adds the SALES_ORDER_AMOUNT to AR_SUMMARY.

Now, look at the procedure in the top half of the code that begins with **Private Sub SALES_ORDER_QUANTITY_AfterUpdate()** and ends with **End Sub**. This procedure checks whether you are attempting to sell more units than you have on-hand. If so, the procedure generates a message, but it is not a warning. After displaying, "Sales quantity exceeds quantity on-hand," the procedure replaces the quantity entered with 0 so that the user must reenter another sales quantity. Only when you enter a quantity less than or equal to the on-hand quantity will the procedure allow the transaction to process and calculate the new on-hand quantity by subtracting the sale quantity from the on-hand quantity. The system moves the new on-hand quantity to the INVENTORY record, and the form displays the new on-hand quantity.

Close the Event Procedure and close the CREATE SALES ORDER screen (select "NO" if asked whether you want to save changes since you made no changes). Reopen the CREATE SALES ORDER screen in **Form View**, click the **New (blank) record** arrowhead previously illustrated to reach the "7 of 7" blank record, and enter the following data:

Field Name	Data (press Enter key after entering each value)
SALES_ORDER_NO	7
SO_DATE	5/1/2016
SALES_REP	John Smart
ACCOUNT_NO	3
INVENTORY_NO	1
SALES_ORDER_QUANTITY	100

After pressing Enter and repositioning the message box, your screen should appear as follows:



Access displays the error message because the sales quantity of 100 exceeds the on-hand quantity of 96. Sally asks you to watch SALES_ORDER_QUANTITY change from 100 to 0 when you click OK in the error message window. You click OK and see the change occur. The edit prevents salespersons from selling a quantity of an item that exceeds the item's on-hand quantity. Sally instructs you to change 0 to **1** in SALES_ORDER_QUANTITY, enter **50000** in SALES_ORDER_UNIT_PRICE, and then press Enter.

Your screen displays the message, "Warning: Sales order amount exceeds customer's credit limit." You click OK and close the CREATE SALES ORDER form, and the system processes the transaction because the credit limit edit is a warning-only edit. The system calculates and displays the new AR_SUMMARY of 50000 and updates this amount in the CUSTOMER table record for account number 3. Also, the system creates a new SALES_ORDER record for SALES_ORDER_NO 7 and updates INVENTORY_ON_HAND to 95 in the INVENTORY table for inventory number 1. You close the CREATE SALES ORDER form.

Production System Summary

At the conclusion of Sally's demonstration, you thank her for taking the time to show you how the current production system processes sales orders. You have accomplished the following:

- Defined your CA/CM's objective: to monitor sales for unit prices that are substantially less than standard prices and for sale amounts that are more than customers' credit limits;
- Obtained an in-depth understanding of how the production business system processes sales orders, including the tables and fields that contain the data that your CA/CM will access;
- Learned that the system displays a warning message when the unit sales price is less than the standard unit price or when a sale amount exceeds a customer's credit limits; however, it does not prevent users from completing sales orders;
- Learned that restricting your access to only the tables you will monitor and limiting your access to read-only will compensate for bypassing other IT controls; and
- Coordinated with the DBA regarding your CA/CM project.

Authorizations and Test Setup

Now, you are ready to obtain the data owner's and custodian's authorizations to have direct, read-only access to the three tables required for your CA/CM. Normally, data owners are the business managers who are the primary users of the data and authorize users' access to the data. You are the owner of the SALES_ORDER and CUSTOMER tables, so you obtain authorization to access these tables from your supervisor, the Vice President of Marketing. At your request, the Director of Operations, who is the owner of the INVENTORY table, grants you read-only access to this table. Also, you obtain authorization from the Vice President of IT, who is the ultimate custodian of the three tables.

After receiving authorization, you send Jon an email requesting that he set up your user ID with read-only access to the SALES_ORDER, INVENTORY, and CUSTOMER tables in SQL Server. The next day, Jon replies saying that he has completed setting up the system to allow your requested access. Jon includes in his email an attachment with instructions on how to set up ODBC links in Access to the requested tables in SQL Server. [Note to students: Your faculty member will provide you with these instructions that have been tailored by your institution's DBA to suit your institution's SQL Server.]

Before setting up your ODBC connections, you will first develop and test your two CA/CM queries in the Access database that you and Sally used in her demonstration. [Note to students: This database is the **INTERNATIONAL AC SALES ORDER SYSTEM** Access database provided by your faculty member that you have already used in performing the case.] Your test queries will use the Access tables in this database in the same way that your CA/CM queries will use ODBC-linked SQL Server tables. Use the following definitions when building these two queries.

Create a new query with the fields below using **Query Design** that identifies all sales orders with sales amounts greater than the customer's credit limit.

SALES_ORDER_NO
SO_DATE
INVENTORY_NO
SALES_ORDER_QUANTITY
SALES_ORDER_UNIT_PRICE
SALES_ORDER_AMOUNT
REMAINING_CREDIT (Build this new field as the customer's remaining credit just before the sale.)
AMOUNT_OVER_CREDIT_LIMIT (Build this new field as the amount by which the SALES_ORDER_AMOUNT exceeds the customer's remaining credit just before the sale.) Sort this field in descending order.
CUSTOMER_CREDIT_LIMIT
AR_SUMMARY_AT_SALE
SALES_REP
ACCOUNT_NO
CUSTOMER_NAME

Create a second new query using **Query Design** that identifies all sales orders for which the sales unit price on the sales order is less than the standard unit price for the air conditioner(s) sold.

SALES_ORDER_NO
ACCOUNT_NO
SO_DATE
INVENTORY_NO
INVENTORY_NAME
SALES_ORDER_QUANTITY
SALES_ORDER_UNIT_PRICE
SALES_ORDER_AMOUNT
STANDARD_UNIT_PRICE
LOSS (Build this new field to calculate the amount by which the actual sales amount is less than what would have been the sales amount had the sale used standard unit price.) Sort this field in descending order.
SALES_REP

Run these two queries, copy and paste screenshots of the results in the Word document, and submit the Word document to your faculty member after completing the case. Also, submit to your faculty member your copy of **INTERNATIONAL AC SALES ORDER SYSTEM** Access database that includes your two queries.

To develop your CA/CM application, you create a new Access database with ODBC-linked tables by performing the following steps:

1. Create a new Access database on your personal computer. Then use the instructions in Jon's email (provided by your faculty member) to set up three ODBC-linked tables in the Access database, one each for the CUSTOMER, INVENTORY, and SALES_ORDER tables.
2. In the new database, recreate the two queries that you developed and tested in the **INTERNATIONAL AC SALES ORDER SYSTEM** Access database, except use the ODBC-linked SQL Server tables instead of Access tables.
3. During the dates and times specified by your faculty member, run your two queries as often as you can. During this period, the DBA at your institution will be entering data at random times into the SQL Server student tables that contain one or both of the anomalies your queries are designed to identify. Each time you find a sales order that has one or both of these anomalies, copy the screen showing the query results along with the date and time that you ran the query and paste it into the Word document that you will submit to your faculty member after completing the case.
4. Submit to your faculty member your new Access database that contains your two CA/CM queries using the ODBC-links to the SQL Server student tables.

Discussion Questions for the Case

Include in your Word document your answers to the questions listed below. When applicable, you should answer each question by referring to the **INTERNATIONAL AC SALES ORDER SYSTEM** Access database that you worked through with Sally, the Director of Application Systems.

1. The incident of a sale substantially below standard price is especially troubling to the organization. Why might this happen and what are the implications?
2. Why won't the DBA allow you "write authorization" directly to the ODBC-linked database when you already have the authorization to create transactions in the production system?
3. Discuss what Referential Integrity means. How do you see it at work in the International AC database?
4. As Director of Marketing, you are the owner of the data in this case. What implications do you think that might have? [Hint: Think about audits, financial reporting, etc.]
5. You saw that a sales order would be processed even though the total sales amount exceeded the customer's credit limit.
   a. Explain Event Procedures and describe their purpose.
   b. Why might this type of sale transaction be allowed?
   c. Should it be allowed to continue? If yes, why? If not, how should it be controlled?
6. What pairs of foreign and primary keys do you see in the data?
7. What happened when you entered "4" into the ACCOUNT_NO field of the CREATE SALES ORDER form? [In other words, what did you see on the screen and how did that happen?] What are the benefits of this database approach to AIS?
8. Describe a CA/CM that would identify anomalies where International AC Purchasing Agents bought air conditioners from vendors at unit costs that exceeded standard unit costs. Include in your description the names of the two tables that the CA/CM would access in SQL Server, the primary key/foreign key fields that relate the two tables, and the names of the CA/CM output fields. Also, indicate the title of the manager who might run this CA/CM and describe a fraud that the CA/CM might reveal.
9. In this case, you developed and performed CA/CM using ODBC links to SQL Server tables. What are the advantages and disadvantages of accessing monitored data in this way as compared to obtaining data extracts from SQL Server tables and importing the data into Access tables?
10. Generally describe the results that you obtained from continuously monitoring the student tables during the continuous monitoring week. Also, describe any indications of fraud that may exist in your results.

CASE LEARNING OBJECTIVES AND IMPLEMENTATION GUIDANCE

Motivation for Development of the Case

There is an increasing need for managers, internal auditors, external auditors, consultants, accountants, and others to monitor and audit entities' computerized data to effectively evaluate risk management, controls, and governance processes. This increasing need is illustrated by the Public Company Accounting Oversight Board's Staff Audit Practice Alert No. 11, which requires external auditors to audit the IT controls upon which IT-dependent controls rely (PCAOB 2013). CA/CM is emerging as one of the primary methods for monitoring and auditing data and controls in computerized systems (Lombardi, Vasarhelyi, and Verver 2014). Many users of CA/CM do not have one of the commercially available Generalized Audit Software (GAS) packages such as Audit Command Language (ACL). However, most do have a personal computer with Microsoft Windows operating system and Access installed. This case teaches students how to implement CA/CM using Access with ODBC links to data stored on a Microsoft SQL Server. In the case, students learn how to coordinate with information systems staff in setting up ODBC links from Access on their personal computer to an actual SQL Server database and use this access to continuously monitor production system data stored in SQL Server for anomalies that are symptomatic of internal control weaknesses.

Learning Objectives

The overarching objectives of this case are to guide students through the analysis, design, and implementation of a CA/CM application and to teach them the challenges, such as ensuring the security and integrity of the production system database, as well as advantages of CA/CM as compared to using GAS to audit data downloaded from a production system. The student learning objectives of this case that support these overarching objectives are as follows:

- Explain how managers and auditors use CA/CM to oversee business processes;
- Explain the advantages and risks of CA/CM;
- Coordinate with IT personnel, including the DBA and the Director of Application Systems, to obtain an understanding of the business system that is to be monitored and to obtain assistance in setting up the ODBC connections;
- Explain why direct access to tables in the relational database bypasses many IT controls in the business system;
- Explain why restricting direct access to read-only on only the tables that will be continuously monitored will compensate for bypassing other IT controls;
- Obtain the data owner's and IT Department's authorizations to directly access the specified tables in the production RDBMS that the CA/CM will monitor;
- Create ODBC links in Access to tables residing in the institution's SQL Server; and
- Create queries in Access using ODBC connections to tables in the production SQL Server and use these queries as CA/CM to identify anomalies.

Prerequisite Knowledge

Students will need the basic foundation in information systems technology that is typically covered in undergraduate AIS courses and a basic understanding of accounting that is normally provided in principles of accounting courses. The case further assumes students are familiar with Access tables, queries, forms, and database relationships. Before assigning the case, we have already taught students how to import data in Excel worksheets into Access tables and how to analyze and audit the imported data using Access queries. We caution against using the case before students have learned these basics, as the case contains no instructions on how to create Access queries in **Design View**.

Course Use

We have used the case in undergraduate and graduate AIS courses, but we believe faculty members can adapt the case for use in auditing and management information systems (MIS) courses. In the case, we consistently refer to CA/CM, as opposed to only continuous monitoring, to indicate the case's relevance to auditing as well as monitoring. Also, we believe that the case's use of information technology such as SQL Server RDBMS and ODBC makes the case adaptable for use in MIS courses. By performing the case, students can better understand what CA/CM is, how managers and auditors use CA/CM to oversee business processes, the challenges and advantages of using CA/CM, and how to create and apply CA/CM in an actual organization.

We recommend assigning the case individually to students so that each student benefits from the hands-on process of developing CA/CM in Access and continuously monitoring for anomalies dynamically recorded in an SQL Server database.

Optionally, the faculty member can assign the ten discussion questions listed at the end of the case to groups of students who present and discuss their responses in class.

When first assigning the case, we provide an electronic copy of the INTERNATIONAL AC SALES ORDER SYSTEM Access database that is included in the Teaching Notes. We spend about 30 minutes in class using the Access database to review general concepts about Access tables, relational databases, forms, and queries. Then, we spend another 30 minutes discussing CA/CM and ODBC. We demonstrate how to set up ODBC connections in Access to the SALES_ORDER, INVENTORY, and CUSTOMER student tables that our DBA previously uploaded in our institution's SQL Server with security established so that each student has only read-access to these tables. We explain that the three Access tables in INTERNATIONAL AC SALES ORDER SYSTEM have the same tables, database relationships, table names, field names, and data types as the three student tables in SQL Server. The SQL Server tables contain considerably more records than the Access tables to provide a more realistic CA/CM experience.

Normally, students require two weeks to complete the case. During the first week, students develop their CA/CM queries, establish ODBC links from their Access database to the student tables in their institution's SQL Server, and identify "test" anomalies in the data stored in the SQL Server student tables. During the second week, students use their CA/CM queries with ODBC connections to student tables in SQL Server to identify "live" anomalies as soon as possible after the DBA enters the anomalies at random times that are known only to the DBA and the faculty member.

At the start of the second week, we discuss in class any problems that students may have encountered in creating their two CA/CM queries with ODBC connections or in using their queries to identify the test anomalies in the SQL Server student tables. Then, the DBA removes the test anomalies and begins inputting new anomalies at random times in the SQL Server student tables. At the same time, the students begin running their CA/CM queries as often as possible to identify each new anomaly as soon as possible after the DBA has entered it. Students copy the screens showing the output from each query run and paste these copies in the Word document that they submit after having completed the case. For each copied screen, a student records the date and time when he/she ran the query.

At the end of the second week, each student submits to the faculty member the Word document that contains their CA/CM outputs and, when assigned individually to students, their answers to the case's ten discussion questions. Also, each student submits electronically two Access databases: (1) INTERNATIONAL AC SALES ORDER SYSTEM provided at the start of the case that contains the student's two queries without ODBC and (2) the Access database created by the student that contains the student's two CA/CM queries with ODBC.

After students have completed the case, we spend about 30 minutes in class leading a discussion during which students describe their experience and key takeaways from performing the case. Also, as an added incentive for timely identification of anomalies, we normally award either extra points or coffee store gift cards to the students who first identified each anomaly after it was entered by the DBA.

Beginning in Fall 2014, we have offered the case in a fully online, asynchronous graduate AIS course. The online graduate AIS classes comprised 34 students in Fall 2014 and 17 students in Fall 2015. We find that online graduate AIS students are able to understand the case materials and perform the case with no difficulties. Instead of in-class discussions, we hold one collaborative session to introduce the case. In addition, we ask students to share their thoughts and experiences performing the case using an online discussion board. Students submit their Word document with screenshots of their CA/CM results and answers to the ten discussion questions along with their two Access databases to an electronic dropbox associated with the online course.

Case Efficacy

A total of 122 students completed the case and efficacy survey questionnaire. These 122 comprised 65 graduate accounting students in a graduate AIS course (14 in Fall 2013, 34 in Fall 2014, and 17 in Fall 2015) and 57 undergraduate accounting students in an undergraduate AIS course (16 in Summer 2014 and 41 in Fall 2014). One of the coauthors used the case in teaching the 65 graduate students and the 16 undergraduate students at one university while another coauthor independently tested the case by using it in teaching the 41 undergraduate students at another university.

We obtained student feedback both orally in a discussion after completion of the case, as well as by distributing a post-case questionnaire, which we administered in accordance with IRB guidelines, with survey questions as shown in Table 1. Overall student feedback was positive. For most of the students, this case was their first learning experience with continuous monitoring. The student feedback indicates that most of the students felt that they had a better understanding of the subject after completing the case. Students said that they especially liked the experience of continuously monitoring data stored in the university's SQL Server database via ODBC communication links set up in Access on their personal computer. Typical student feedback was: "This was a great way to learn hands-on and I thoroughly enjoyed it" and, "I found this case very interesting, but the ODBC was the best part."

TABLE 1

Summary of Results from Students Who Completed the Questionnaire after Completing the Case

Question	University 1 (n = 81)	University 2 (n = 41)	Mean	Std. Dev.	Variance	Median
1. This case helped me to understand better and apply Open Database Connectivity (ODBC) links established between a manager's or auditor's Access database on his/her PC and data residing in a large-scale relational database in a client-server system that supports a midsize or large business. (n = 122)	4.75	4.61	4.70	0.69	0.47	5.00
2. This case helped me to understand better how managers and auditors use continuous monitoring to oversee business processes. (n = 122)	4.74	4.51	4.66	0.66	0.44	5.00
3. Overall, performing this case increased my software skills in using Microsoft Access as a tool for continuously monitoring and auditing computerized data. (n = 122)	4.73	4.41	4.62	0.70	0.48	5.00
4. This case was interesting to me, and I enjoyed it more than a typical AIS homework assignment. (n = 122)	4.81	4.54	4.72	0.55	0.30	5.00
5. This assignment helped me in applying the concepts and software skills I have learned in this class. (n = 122)	4.58	4.46	4.54	0.72	0.51	5.00
6. Before starting this case I had not learned about continuous monitoring, but after completing this case I understand what continuous monitoring is and how to use Access with ODBC to implement continuous monitoring. (n = 122)	4.67	4.56	4.63	0.69	0.48	5.00

For each question shown in Table 1, students were asked to select a response on a five-point Likert scale: 5 ("Strongly Agree") to 1 ("Strongly Disagree"). The survey results were, on average, very close to the same for both the 65 graduate students as compared to the 57 undergraduate students and for the 81 students at one university as compared to the 41 students at the other university. The survey results suggest that students generally found the case to be an enjoyable learning experience that increased their understanding of CA/CM.

Contributions to Classroom Practices

This case is the first we know of that teaches students how to create and use Access queries with ODBC connections to continuously monitor for anomalies dynamically recorded in an SQL Server RDBMS. Students learn how to monitor and audit data stored in SQL Server tables of virtually any size in real-time. By enabling real-time access to the data as it currently exists in SQL Server tables, ODBC enables Access queries to function as nearly real-time CA/CM applications. The only lag is the time from when the data changes until the time when the user runs his/her Access query. Also, the case teaches students how to work with IT and the DBA to set up ODBC links that do not risk corrupting the entity's enterprise data or impacting the performance of the enterprise business system.

TEACHING NOTES AND STUDENT VERSION OF THE CASE

Teaching Notes and the Student Version of the Case are available only to non-student-member subscribers to *Journal of Emerging Technologies in Accounting* through the American Accounting Association's electronic publications system at http://www.aaapubs.org/. Non-student-member subscribers should use their usernames and passwords for entry into the system where the Teaching Notes can be reviewed and printed. The "Student Version of the Case" is available as a supplemental file that is posted with the Teaching Notes. Please do not make the Teaching Notes available to students or post them on websites.

If you are a non-student-member of AAA with a subscription to *Journal of Emerging Technologies in Accounting* and have any trouble accessing this material, please contact the AAA headquarters office at info@aaahq.org or (941) 921-7747.

REFERENCES

Lombardi, D. R., M. A. Vasarhelyi, and J. Verver. 2014. Continuous controls monitoring: A case study. *Journal of Emerging Technologies in Accounting* 11 (1): 83–98.

Public Company Accounting Oversight Board (PCAOB). 2013. *Considerations for Audits of Internal Control Over Financial Reporting*. Staff Audit Practice Alert No. 11. Available at: https://pcaobus.org/Standards/QandA/10-24-2013_SAPA_11.pdf

Copyright of Journal of Emerging Technologies in Accounting is the property of American Accounting Association and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.