Spring 4-1-2002

Coca Cola Bottling of Central South Dakota: Sales Center Warehouse

Kristian C. D. Palmer
Dakota State University

MSIS
PROJECT APPROVAL FORM

Student Name: Kristian C. D. Palmer

Expected Graduation Date: 1/15/03

Master's Project Title: Coca Cola Bottling of Central South Dakota: ~~South~~ Sales Server Warehouse Inventory Database

Date Project Plan Approved: 8/1/02

Date Project Coordinator Notified and Grade Submitted: _____

Approvals/Signatures:

Student: K Pal Date: 2/23/03

Faculty supervisor: Kathe Wang Date: 2/27/03

Committee member: Rudy Tupper Date: 2-20-03

Committee member: Ishai Shaw Date: 2/24/03

To Ensure Certification of Completion:
Student must bring or send the original to the Graduate Programs Office.
Copies on acid free paper go to the library with the reports for binding.

Original to Graduate Programs Office
Acid-free copies with written reports to library
Copies to: Project supervisor and committee and to MSIS Coordinator

Revised 1/28/03

Coca Cola Bottling of Central South Dakota: Sales Center Warehouse Inventory Database

BY: Kristian C. D. Palmer

A project submitted in partial fulfillment of the requirements for the Master of Science in Information Systems

Dakota State University

2002

Abstract

Coca-Cola Bottling of Central South Dakota (CCBCSD), which is comprised of four separate sales centers and warehouses, and is owned by Chesterman Company of Sioux City Iowa, has experienced radical (month to month) discrepancies in inventory tracking and charges over the past 12 to 32 months. To gain an understanding of individual sales center inventory trends involving shipments both into and out of sales centers, as well as sales and damaged/destroyed products, CCBCSD has determined the need for an onsite inventory tracking database. Three of the four sales centers maintain Microsoft Windows 98 personal computers with Microsoft Office 2000 Professional Edition business software. It was determined that MS Access 2000 database software should be used as the basis for the inventory tracking database. The project to design the custom database for the CCBCSD sales centers was initiated September 1, 2002, with the expressed goal of full implementation on December 1, 2002. The project was conceived, designed, tested and implemented within the allotted three month period. A business web site for CCBCSD sales centers was created as a demonstration of possible future expansion for the database.

Acknowledgements

I would like to thank the following for their greatly appreciated aid:

- The Coca-Cola Bottling Company of Central South Dakota Managers and Secretaries
- Dr. Wang (Project Advisor)
- Dr. Zhou (Project Committee)
- Randy Tupper (Project Committee, Watertown Sales Center Manager)
- The MSIS Faculty of Dakota State University
- Laurie Dennis and Dawn Hamilton DSU Graduate Office
- Maki Shinohara (Graphic Design)
- Pamela and David Palmer (Parents)

Table of Contents

List of Figures & Tables

Introduction

Inventory tracking is important for any business. Businesses may track product inventory regarding shipments and sales. They may also track office supplies and equipment. Product inventory tracking and monitoring allows businesses to determine many aspects of a product as it enters a warehouse and is stored, sold and delivered to the customer. Inventory turnover rates, sales volumes, and deliveries provide valuable information about products. For example, understanding how many days' sales are stored as inventory at warehouses aids managers in determinations of warehouse efficiency rates. Determining sales volumes for individual products can provide the basis for cost efficiency of that product. It may then be determined whether the product price needs to be adjusted or if the product should be discontinued. Inventory management may also provide hints of theft by employees.

Tracking deliveries allows a business to determine the efficiency of wholesale providers and its own retail deliveries. Tracking deliveries allows a business to determine the efficiencies of not only its retail, but also those of wholesale providers. Businesses with ten, or as many as ten thousand products, benefit from inventory tracking and monitoring.

Inventory management, tracking and monitoring requires and generates vast amounts of data. Managing incoming inventory with stored and outgoing inventory, determining what is being sold, when it is sold and who is purchasing that inventory are all aspects contributing to the data pool. There are literally hundreds of software companies and products devoted to inventory

management. There are inventory software suites, stand-alone software solutions, handheld devices, barcode readers, enterprise software solutions based on all types of software. There are Oracle databases, Microsoft Access databases, Sun Microsystems databases, and on and on. The diversity of the programs is only exceeded by that of the inventory which, is managed, controlled and monitored. All of these programs, software solutions, suites, and hardware devices track and monitor point of sale, inventory control, order entry, returns, shipping/receiving, invoices, reporting and various other inventory specific data. The cost for these programs and devices may run into the tens and hundreds of thousands of dollars and track millions, billions and trillions of dollars in inventory.

Statement of Problem

Coca-Cola Bottling of Central South Dakota, with sales centers in Huron, Pierre, Watertown and Mitchell, was purchased by Chesterman Company of Sioux City, IA, in February 1999. Chesterman Company maintains a bottling and canning plant in Sioux City, Iowa. In October 1999 the current Coca-Cola Bottling of Central South Dakota (CCBCSD) stopped all bottling production and sold off its bottling equipment, thus becoming a group of branch sales centers for Chesterman Company. CCBCSD then began receiving the majority of its product supply from the bottling and canning plant in Sioux City. See figure 1 for a breakdown, by percentages, of product supply companies.

Sioux City Coca-Cola Brand Products	87.2
Coca-Cola USA	6.6
Premium Waters Incorporated	2.4
Others	4.8
	100.0

Figure 1 CCBCSD Product Providers Breakdown (by percentage)

The CCBCSD sales centers division of Chesterman Company provides Coca-Cola brand soft drinks as well as various brands of water and non-Coca-Cola brand flavored drinks to approximately 25 percent of the state of South Dakota's population (205,000 people) in a region covering approximately 35 percent of the geography of the state. See figures 2a, 2b for a region map.

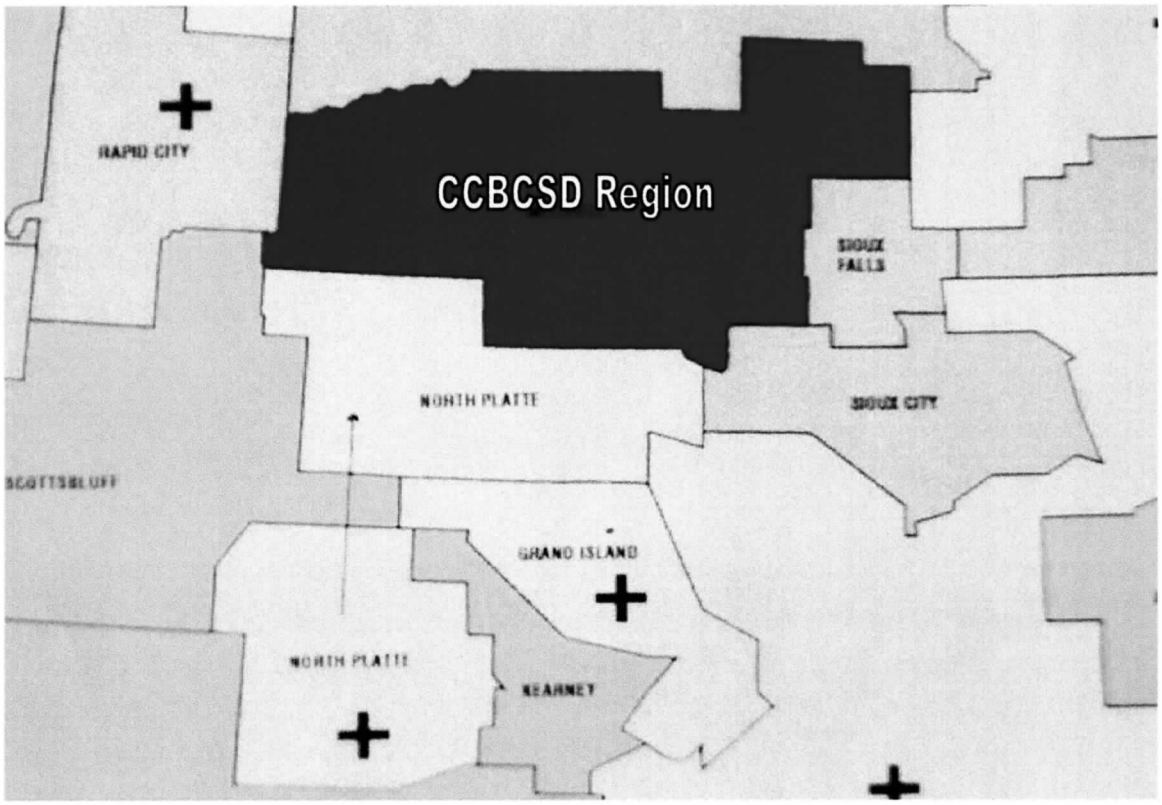


Figure 2a CCBCSD Sales Region Map

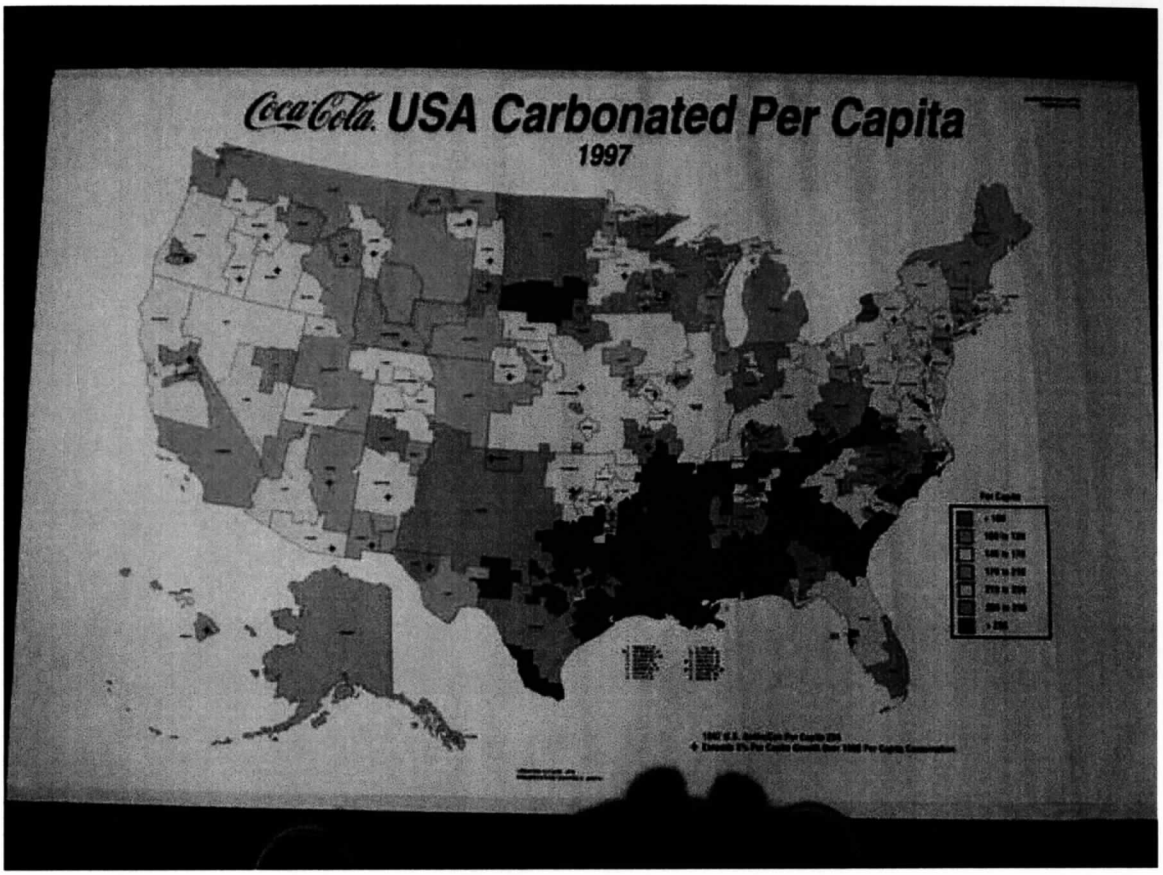


Figure 2b Coca-Cola Sales Market Regions, United States Map

CCBCSD case sale volumes for the combined four sales centers average one hundred twenty-five thousand per month. See table 1 for the FY 2002 case sales report.

Currently the Pierre and Watertown sales centers order their own products from the Sioux City plant, and the Mitchell sales center orders products for both the Mitchell and Huron sales centers. Orders for products are sent via e-mail using an MS Excel Order loadsheet, appendix (A-1). The Order loadsheet contains twenty-four unique package or product name types i.e. (Six Pack Cans, Two Liter Bottles) and one hundred seventy-nine flavor names i.e. (Coca-Cola,

Sprite) constituting the same number of unique individual products. The products on the Order loadsheet do not represent the entire product list that any sales center may store. These are only the most commonly ordered products. Between thirty and sixty other products not listed in the Order loadsheet may be ordered and stored at any of the sales centers, depending on the respective market of each sales center. See appendix (A-2) for a complete product list covering all sales centers.

Products, and their packages, stored and sold by the sales centers, are often changed, added or discontinued. All of these products are packed into cases of various individual package counts, and these cases are stacked onto pallets. The pallets of cases may then be transferred by forklift to and from semi-tractor trailers, from bottling and canning plants and, between sales centers.

Currently the Sioux City bottling and canning plant tracks all products, which are sold and purchased by the CCBCSD sales centers, with what is known as a NORAND code. The NORAND code is a six digit identifying number for each individual product. See appendix (A-2) for existing NORAND code numbers. The NORAND code numbers accompany products in all sales and inventory reports handled by the Sioux City plant and sales centers. NORAND numbers are also used by pre-salesmen during daily uploads of sales information from their handheld sales devices. The devices are provided by NORAND Data Systems, and upload sales information to the database server at the Sioux City plant. UPC codes are not used to track or order any products sold or purchased by the sales centers. Only NORAND code numbers are used when referring to

various products. Therefore, sales centers do not possess or use UPC barcode readers at any time.

For the week of November 4-8 of 2002 the Sioux City plant maintained a product list of 461 four hundred sixty-one, unique products. In comparison, for the same week the Mitchell sales center maintained a product list of 229, two hundred twenty-nine unique products. These numbers represent the total number of unique products that the Sioux City plant or individual sales centers may sell or store in inventory. Referring to table 1, one also gains a perspective on average inventory values for each sales center, as well as sales numbers, which keep inventory values fluctuating daily.

Table 1
Inventory and Sales Report for CCBCSD Sales Centers Fiscal Year 2002 Jan-Oct.

Values represent cases of product. Inv. Total represents beginning inventory for that month. Mo. Sales represents Monthly Sales. Sales YTD represents Case Sales Year to Date.

January	Mitchell	Huron	Pierre	Watertown
Inv. Total	21348	4600	12902	18334
Mo. Sales	36158	11442	26532	45819

February	Mitchell	Huron	Pierre	Watertown
Inv. Total	22991	3905	15587	25508
Mo. Sales	38765	14456	23010	33958

March	Mitchell	Huron	Pierre	Watertown
Inv. Total	29832		16248	27094
Mo. Sales	52964		26288	35046

April	Mitchell	Huron	Pierre	Watertown
Inv. Total	30254		15803	20784
Mo. Sales				

May	Mitchell	Huron	Pierre	Watertown
Inv. Total	27441		14628	23463
Mo. Sales	85296		36841	54819

June	Mitchell	Huron	Pierre	Watertown
Inv. Total	42973		18057	28869
Mo. Sales	59071		32486	39269

July	Mitchell	Huron	Pierre	Watertown
Inv. Total	43722		18640	30987
Mo. Sales	77105		43252	50137

August	Mitchell	Huron	Pierre	Watertown
Inv. Total	33037		14472	24952
Mo. Sales	85085		34330	42580

September	Mitchell	Huron	Pierre	Watertown
Inv. Total	34014		16961	24409
Mo. Sales	47679		27479	44707

October	Mitchell	Huron	Pierre	Watertown
Inv. Total	32566		16143	24366
Mo. Sales	68700		28782	42985

Totals				
Sales YTD	550,823	25,898	279,000	389,320

When the CCBCSD sales centers began purchasing all products from the Sioux City plant and other third party wholesalers, the sales centers became dependent on the Sioux City sales and transfers database for tracking and monitoring their inventories. For example, the Mitchell sales center would order products from Sioux City using an Order loadsheet. Sioux City would then load semi-trailers with product destined for the sales center. Case inventory values represented in a Shipping loadsheet would be entered into the Sioux City inventory database as sales to the sales center. The sales center was, and is

currently, charged for those products at the end of each month. Also, at the end of each month every sales center is required to take inventory of the warehouse and return the inventory values to the Sioux City office. Sales center purchases and sales are totaled and this value is referenced with actual inventory values provided by the sales center. This allows the Sioux City office to determine charges accrued and owed by the sales center. See table 2 for book minus actual inventory discrepancies during FY 2002.

Table 2
Sioux City office Book Inventory minus CCBCSD Sales Center Actual Inventory for FY 2002

Table values represent case value discrepancies between actual number of cases in inventory at a sales center and number of cases stored in book inventory at the Sioux City office.
Values are in cases. Negative (red) values are charged to the Sales Centers.
Positive black values represent inventory longs, and costs are not re-imbursed to the Sales Centers.

	Mitchell	YTD	Huron	YTD	Pierre	YTD	Watertown	YTD
January	(511)	(511)	(568)	(568)	(352)	(352)	(145)	(145)
February	5256	4745	(2016)	(2584)	(585)	(937)	(484)	(629)
March	(3557)	1188			(204)	(1141)	(868)	(1497)
April	933	2121			(255)	(1396)	(106)	(1603)
May	53	2174			(1331)	(2727)	366	(1237)
June	(3601)	(1427)			(1)	(2728)	(377)	(1614)
July	(1189)	(2616)			(2184)	(4912)	797	(817)
August	14	(2602)			(519)	(5431)	(211)	(1028)
September	(204)	(2806)			(801)	(6232)	(1312)	(2340)
October	(58)	(2864)			(832)	(7064)	(555)	(2895)
November								
December								
Totals Case Discrepancies YTD								(15407)

This method of maintaining inventory, sales and purchase values at the Sioux City office often results in discrepancies. There are many possibilities for case counts to be mistakenly maintained by the Sioux City office. First of all,

products that are ordered by one sales center via the Order loadsheet may actually be charged to another sales center. Other problems may also occur. Inventory number discrepancies between Sioux City and a sales center may also be due to occurrences at the sales center. Products may be stolen from a warehouse or a sales delivery truck, sales delivery drivers may deliver too many or too few products to customers, they may deliver product that was not purchased by a customer, they might pick up product from a customer and return it to the warehouse without reporting the pick up or they may not turn in sales reports to the sales center office. This latter problem typically occurs when deliveries are made during non-business days or hours.

With case product costs ranging from $3.50 to $28.00 any discrepancies in book and actual inventory values become quite costly. With average monthly sales of 124,500 cases and yearly sales of 1.5 million cases, charges for an additional .0085 percent of yearly sales may look at first glance to be rather insignificant. However, consider a low average product cost per case of $4.00 and multiply that by current year to date discrepancies (table 2) of 15,407 through October 2002 and one begins to realize the great importance in determining where errors are occurring. Certainly a fiscal error costing $61,628.00 would not be acceptable, over ten months, for most companies or corporations. Yet, these costs still do not elucidate the full, negative effect discrepancies can make on net sales income figures. With a case product cost of $4.00 the wholesale cost is even larger. The product cost for one case purchased by CCBCSD may be $4.00 yet, the wholesale cost which CCBCSD

charges its customers may be twenty to one hundred percent higher than the product cost. With this consideration the $4.00 case would actually be worth $5.00 to $8.00 and the $61,628.00 cost in errors could be valued up to $123,256.00. Hence, these errors become even more unacceptable when considering most of this cost is borne through data input errors and, in some circumstances, incorrect shipments.

It is understandable how and why these problems may occur in this system. It is also understandable why the sales centers would want to find a better method for tracking their inventories, and in the end saving themselves money, by not being charged for product that was either never delivered, or was mis-reported at some point during the month.

In the past twelve months the CCBCSD sales centers have each received at least one PC and printer. The PC's are equipped with Windows 98 second edition operating systems and Windows 2000 Office Professional edition. These computers are Internet enabled by either broadband cable provided by Midcontinent Communications Company or by Dial-up. The computers also have access to the Chesterman Company VPN (Virtual Private Network).

Objectives

The managers of the CCBCSD sales centers wanted a method to maintain their own inventory databases, in order to determine for themselves where, why and how discrepancies are occurring. They would like to be able to

maintain as close to possible a running inventory, so that it will be known how many cases of any product are actually in the warehouse and on sales delivery trucks at any given time during the month.

The managers also wanted to use only that software which was currently installed and available to them on their sales center office computers. There would also not be any need, as seen, for remote access to the database as hard copy sales and inventory reports are kept at each sales center office and are not to be removed. Remote access was also not seen as a good choice for the database as most office secretaries and managers have limited PC abilities and skills. It was suggested that perhaps, at most, the ability to e-mail data would be desirable.

In order to maintain a running inventory at a sales center warehouse, several values must be tracked and verified. These include: a beginning inventory, received products, shipped products, sales and damaged products. As mentioned, the product list for a sales center is large and variable, with new products coming in, and other products being discontinued. With this in mind any inventory database would need an adjustable product list, which would allow for adding new products, as well as marking products as discontinued.

It was determined that there were eight goals a sales center warehouse inventory database should achieve, maintain and deliver. These goals were:

1. The ability to view expected and actual inventory for individual products.

2. The ability to view shipment inventory values of both incoming and outgoing shipments.
3. The ability to maintain a daily inventory for individual products.
4. The ability to input sales.
5. The ability to input received shipments.
6. The ability to input outgoing shipments.
7. The ability to input damaged products.
8. The ability to add or discontinue products from the product list.

These were the original objectives for this project: **The Coca Cola Bottling of Central South Dakota Sales Center Warehouse Inventory Database.**

Given these objectives, the current computing and software availabilities and constrictions on the use of software, it was determined that MS Access 2000 would serve as the database software to base this project upon. It was determined that the database program would be replicated and copies installed at each of three sales center offices in the cities of Mitchell, Pierre and Watertown, SD. The program would be as simple to use as possible due to a perceived high learning curve for employees. The program would provide printable reports on inventory and product list data. The program would not require large amounts of time for data input, as this would be considered an extra task for office employees. Individual training for sales center managers and secretaries would be provided. Individual copies of the required program files and a hard copy manual would be provided to explain the program. The program

files and manuals would be distributed and required office managers and secretaries would be trained in order that full deployment and use of the database could begin on December 1, 2002.

Scope of Project

This project not only presented a steep learning curve for employees of CCBCSD, but for me as well. See appendix (A-3) for a Gantt Chart related to the overall progress of process concerning this project. The first step taken in this project was familiarization with MS Access 2000 software by myself the project designer and implementer. Two methods for this familiarization included; firstly, the purchase of an MS Access self-study kit and secondly, registering with an MS Access developer forum (Microsoft Press, 1999; Utter Access Discussion Forums, 2002). On September 1, 2002, I purchased the study kit, registered with the Utteraccess forums, and began learning how to use MS Access 2000 database software. By September 5, 2002, I determined that I would need approximately the remaining days of the month to feel fully secure in beginning actual work towards creating the Warehouse Inventory Database for CCBCSD. The MS Access 2000 study kit was very helpful in explaining how forms and form objects performed as well as how to set up queries, which are quite a different beast from standard SQL queries. The Utteraccess forums were abundantly helpful as they were highly searchable and extremely thorough. Real-world application of MS Access applications and tenets are replete throughout the

Utteraccess discussion forums. With the aid of the study kit and forums the month of September was filled by trials and tests and exploration of MS Access.

By the end of September I felt confident that MS Access would suit the project well. On October 2nd I met with the CCBCSD manager to discuss where I was with the project and for the purpose of retaining a current product list. We discussed the NORAND system of identification of products and I explained that I felt it might not be a necessity for this particular application. It was agreed that leaving the NORAND code numbers out would be acceptable; however, they would be used if during the first trials employees and managers felt that they were needed. This turned out to be a mistake, and will be explained when discussing version 2 of the project. We also discussed the goals of the project again and it was stressed that "actual" inventory numbers and reports concerning these numbers were very important. At this time it was determined that the "actual" inventory data should be maintained in the database month to month. This meant stock takes in a warehouse must be maintained for a report of actual beginning and ending inventory. It was determined that there would be an individual report for actual inventory data only, with a separate book inventory report covering sales and shipments. By using this method I explained that neither report would then represent an expected inventory for the warehouse. In order to determine the expected actual inventory, a running inventory, one would need to add the actual inventory value to the book inventory value. This also turned out to be a mistake and will also be explained when discussing version 2 of the project.

On October 3rd, I received the requested product list via e-mail and was prepared to begin work on the database. The product table was then created and populated with the products. Products as explained earlier, are two-part entities with a package name and flavor name. For example, a 24 pack of Coca-Cola constitutes one unique product with 24 pack as the package name and Coca-Cola as the flavor name. A 6 pack of Coca-Cola constitutes another unique product. The product table was named tblProducts and included four unique fields: Product_ID an auto number field and primary key, Package_Name a text field, Flavor_Name a text field, and Discontinued a yes/no field. The original product list, which populated tblProducts, contained 219 products. The Stock Take table named tblStock_Take was created with five unique fields: Stock_Take_ID an auto number field and primary key, Stock_Take_Date a date field, Product_ID foreign key, Unit_Count a number field, and Damaged_Product a number field. The Stock_Take_ID would serve as an indexed non-duplicate identifier for a data input of each unique product and stock take of actual inventory. Unit_Count would serve as the input field for case/unit values as would the Damaged_Product field. The tblInventory_Transactions was created with seven unique fields: Transaction_ID an auto number, indexed, non-duplicate primary key field was created to index separate transactions, Transaction_Date, Product_ID foreign key, Transaction_Note a text field, Units_Received, Units_Sold and Units_Shipped. The transactions table would hold book inventory values and the Stock Take table would hold actual inventory values.

The tables were then placed into relationship with each other. The relational database was now created, normalized and ready for use (Figure 3).

CCBCSD Sales Center Warehouse Inventory Database Relationship Diagram

Table Relationship Diagram for Sales Center Warehouse Database version 1.0

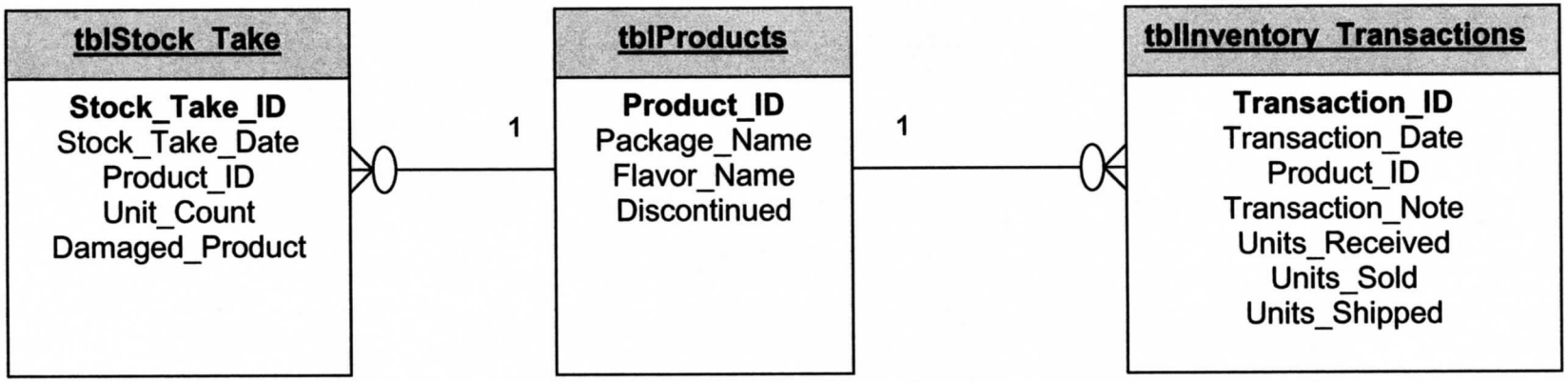


Figure 3 Table Relationships version 1

It was now possible to create forms for a better graphical interface with the database. It was also possible to create working reports and delete queries. Separate forms for inputting actual (figure 4) and book inventory (figure 5) were created along with a form for adding or discontinuing products (figure 6).

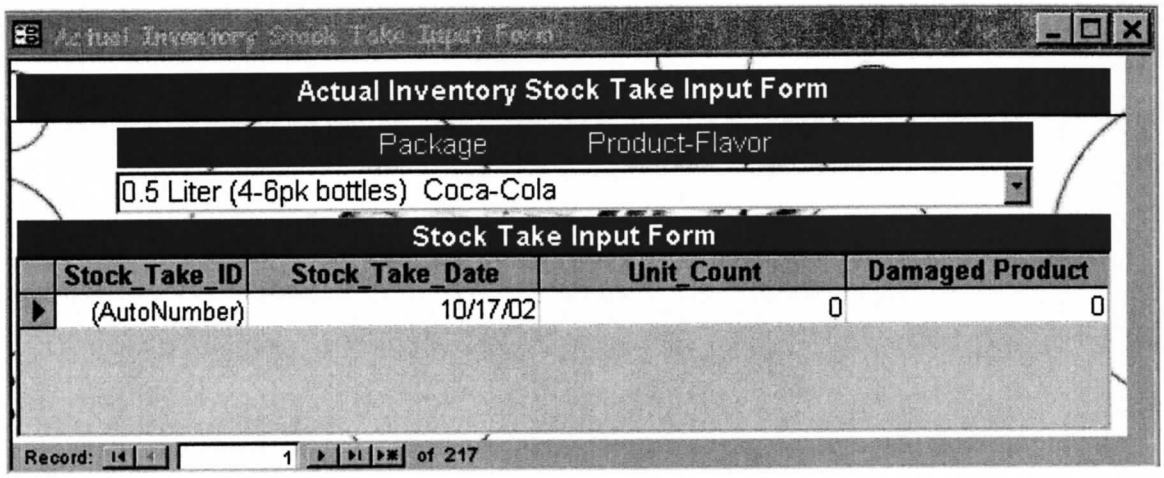


Figure 4 Actual Inventory Stock Take Input Form version 1

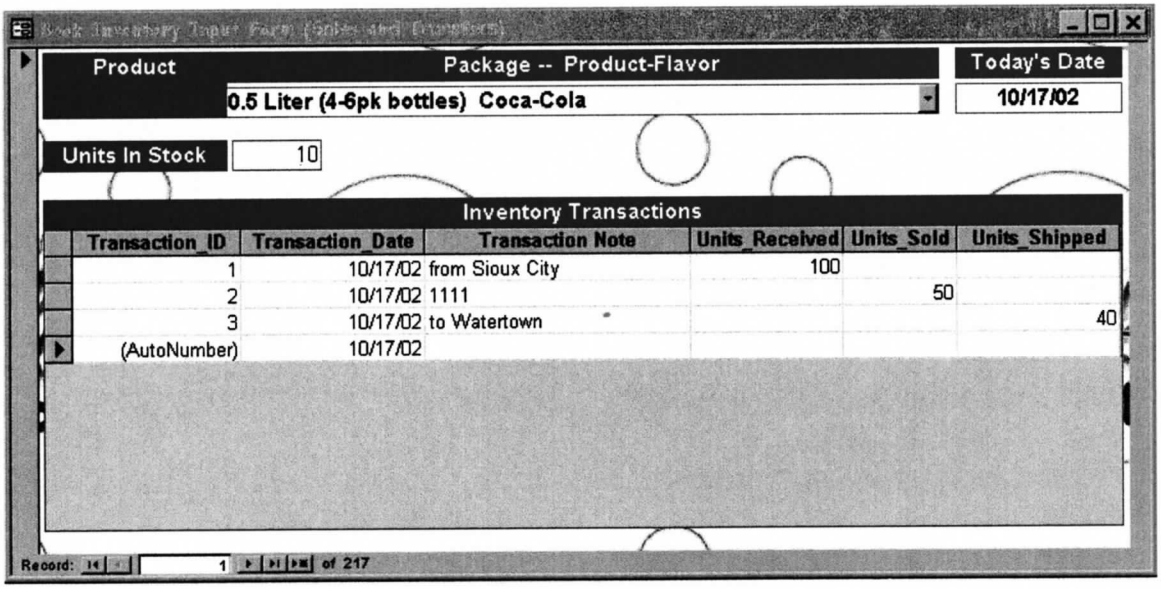


Figure 5 Book Inventory Input Form version 1

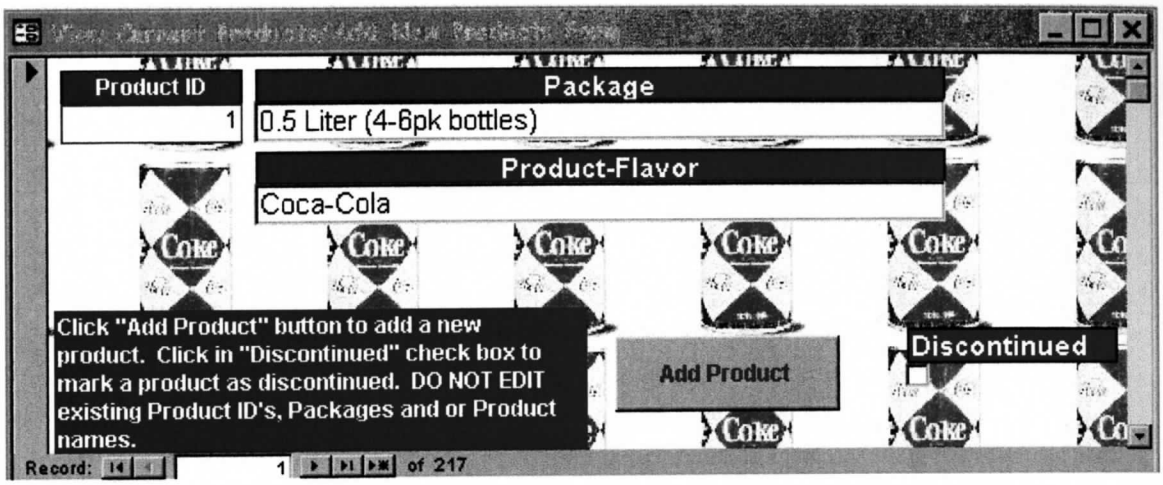


Figure 6 View Add Discontinue Products Form version 1

Delete queries, which act to delete table data specified by user-defined time periods (figure 7) were also created in order to clean up the database, by removing old data. Book inventory data would be deleted monthly and actual inventory data would be deleted when three or more months' data had accumulated.

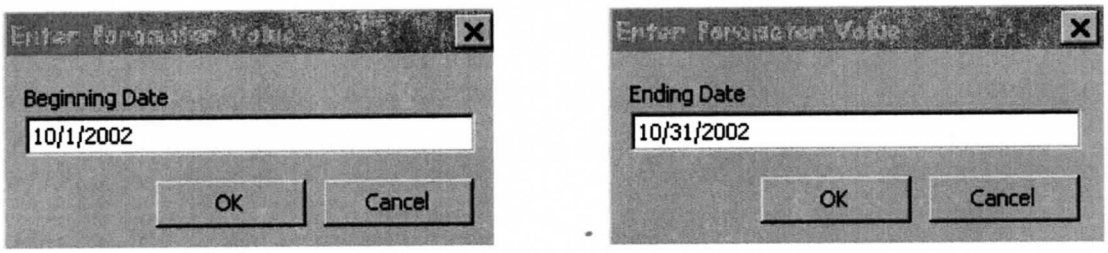


Figure 7 Prompt windows requesting user to define time period

On October 14th I obtained actual company generated inventory and sales reports from the CCBCSD manager for use in testing the program. The forms and delete queries were then tested for compatibility with the tables and data

input. Next, the MS Access reports were designed. See reports (1,2) for sample Actual and Book inventory reports from version 1.

Report 1, Version 1
Actual Inventory Report by Month and Product (total units)

Actual Inventory Report

by Date and Product

For Stock Take on Date: October 26, 2002

Package	Product-Flavor	Total Unit Count	Damaged
0.5 Liter (4-6pk bottles)	Coca-Cola	100	0
0.5 Liter (4-6pk bottles)	Sprite	100	0
0.5 Liter (4-6pk bottles)	CF Diet Coke	100	0
0.5 Liter (4-6pk bottles)	Mr. Pibb	50	0
0.5 Liter (4-6pk bottles)	Mello Yello	50	0
0.5 Liter (4-6pk bottles)	Dr. Pepper	50	0
0.5 Liter (4-6pk bottles)	Yoohoo	25	0
1 Liter (12 bottles)	Coca-Cola	25	0
1 Liter (12 bottles)	Diet Coke	50	5
1 Liter (12 bottles)	Cherry Coke	100	5
1 Liter (12 bottles)	Sprite	100	5
1 Liter (12 bottles)	Pibb Xtra	450	0
1 Liter (12 bottles)	Mello Yello	40	0
1 Liter (12 bottles)	Dr. Pepper	13	0
1 Liter (12 bottles)	Diet Dr. Pepper	78	0
12 Pack (2-12pk cans)	Coca-Cola	56	0
12 Pack (2-12pk cans)	CF Coca-Cola	59	0
12 Pack (2-12pk cans)	Diet Coke	45	0

Report Created on: 11/26/2002 10:40:03 PM

Page 1 of 1

Report 2, Version 1:
Estimated Inventory Report by Time Period and Product

Book Inventory Report

by Product and Month

For Transactions in the Month: October 2002

Package	Product-Flavor	Total Units Received	Total Units Sold	Total Units Shipped	Estimated Inventory
0.5 Liter (4-6pk bottles)	Coca-Cola	50	10	0	40
0.5 Liter (4-6pk bottles)	Diet Coke	56	25	0	31
0.5 Liter (4-6pk bottles)	Sprite	100	56	0	44
0.5 Liter (4-6pk bottles)	CF Diet Coke	40	5	20	15
0.5 Liter (4-6pk bottles)	Mr. Pibb	100	50	0	50
0.5 Liter (4-6pk bottles)	Mello Yello	10	1	0	9
0.5 Liter (4-6pk bottles)	Dr. Pepper	400	40	0	360
0.5 Liter (4-6pk bottles)	Yoohoo	36	15	0	21
1 Liter (12 bottles)	Coca-Cola	50	10	0	40
1 Liter (12 bottles)	Diet Coke	96	60	0	36
1 Liter (12 bottles)	Cherry Coke	96	62	0	34
1 Liter (12 bottles)	Sprite	800	69	0	731

Report Created on: 11/26/2002 10:40:03 PM

Creating the initial report layouts and calculations proved to be fairly difficult tasks. To create understandable reports required building rather complex SQL statements where the tblProducts table is inner-joined with either tblInventory_Transactions or tblStock_Take table via the Product_ID primary key of each table.

SQL Statements for Reports: Version 1

```
SELECT tblProducts.Package_Name, tblProducts.Flavor_Name,
tblStock_Take.Stock_Take_Date, tblStock_Take.Product_ID, tblStock_Take.Unit_Count,
tblStock_Take.Damaged_Product
FROM tblProducts
INNER JOIN tblStock_Take
ON tblProducts.Product_ID = tblStock_Take.Product_ID
WHERE (((tblStock_Take.Stock_Take_Date)=[Date of Inventory Stock Take]));

SELECT tblProducts.Package_Name, tblProducts.Flavor_Name,
tblInventory_Transactions.Product_ID, tblInventory_Transactions.Units_Received,
tblInventory_Transactions.Units_Sold, tblInventory_Transactions.Units_Shipped,
tblInventory_Transactions.Transaction_Date
FROM tblProducts
INNER JOIN tblInventory_Transactions
ON tblProducts.Product_ID = tblInventory_Transactions.Product_ID
WHERE (((tblInventory_Transactions.Transaction_Date)>=[Beginning Date] And
(tblInventory_Transactions.Transaction_Date)<=[Ending Date]));
```

These SQL statements allowed the reports to generate the unique product names from the tblProducts table, which could then be mated with inventory values from the tblInventory_Transactions and tblStock_Take tables. The calculations upon the inventory values stored in the transactions and stock take tables required creating aggregate functions. To sum the values from a field normally would only require a Sum([This_Field]) calculation. The Sum function calculates populated and zero length strings when creating a total. This means fields must either contain a value or "" two quotes representing a zero length string. If all field values contain values or zero length strings then the Sum

function will calculate a total. However, when a field contains a NULL value the Sum function will not calculate a total. Understanding how the Sum function performs makes it evident, that for any calculations to occur in the Inventory reports an aggregate function is needed. Consider the Book Inventory report. The report contains calls for values from fields: Units_Received, Units_Sold and Units_Shipped. Any individual product in the Inventory Transactions table may hold values in all, one, two or none of these fields. When summing the fields individually the Sum function could be used to produce a total for that field. However, the report also calls for an "Estimated Inventory" calculation, which subtracts the Units_Sold and Units_Shipped fields from the Units_Received field for each product. NULL values present in any one of these fields would produce a NULL value total without the nz (non-zero) function. The nz function takes the form nz(variant,[valueifnull]). Therefore zero, zero length strings or other string values may be substituted in the place of a NULL value. Consider table three in regard to this information.

Table 3 tblInventory_Transactions Version 1, datasheet view

tblInventory_Transactions : Table

Transaction ID	Transaction Date	Units Received	Units Sold	Units Shipped
16	10/1/2002	50		
17	10/1/2002		10	
18	10/1/2002	56		
19	10/1/2002		25	
20	10/1/2002	100		
21	10/1/2002		56	
22	10/1/2002	40		
23	10/1/2002		5	
24	10/1/2002			20

As seen in table three a single transaction may hold up to three inventory related values, however most commonly the transaction holds only one value leaving other fields NULL. The reality of the nature of stored field values in these tables ultimately requires use of aggregate functions in the form of =Sum(nz([Field_Name])), which places a value of zero in NULL fields, allowing calculations to produce a non-NULL value as total. The aggregate function calculations for the version 1 Book Inventory report:

1. Total Units Received for an individual product
   =Sum(nz([Units_Received]))
2. Total Units Sold for an individual product =Sum(nz([Units_Sold]))
3. Total Units Shipped for an individual product
   =Sum(nz([Units_Shipped]))
4. Total Estimated Inventory for an individual product
   =Sum(nz([Units_Received])-nz([Units_Sold])-nz([Units_Shipped]))

With the forms, delete queries and tables finalized and the reports working properly the next step was to create a menu board for navigating through the program. MS Access 2000 makes creating a menu board fairly simple via the Switchboard Manager. Using the Switchboard Manager is not a highly esteemed method for creating menus according to discussions within the Utteraccess forums, however I chose to try the method as I found it simple and functional on my MS Windows 2000 PC. By October 20th the entire program fulfilled the stated

agreed upon objectives at this point. Yet, I was not satisfied with the graphical nature of the forms or switchboard menus. I chose to obtain a few publicly available .gif files for the purpose of adding some flair and color. Figure 8 contains the initial opening switchboard menu form.

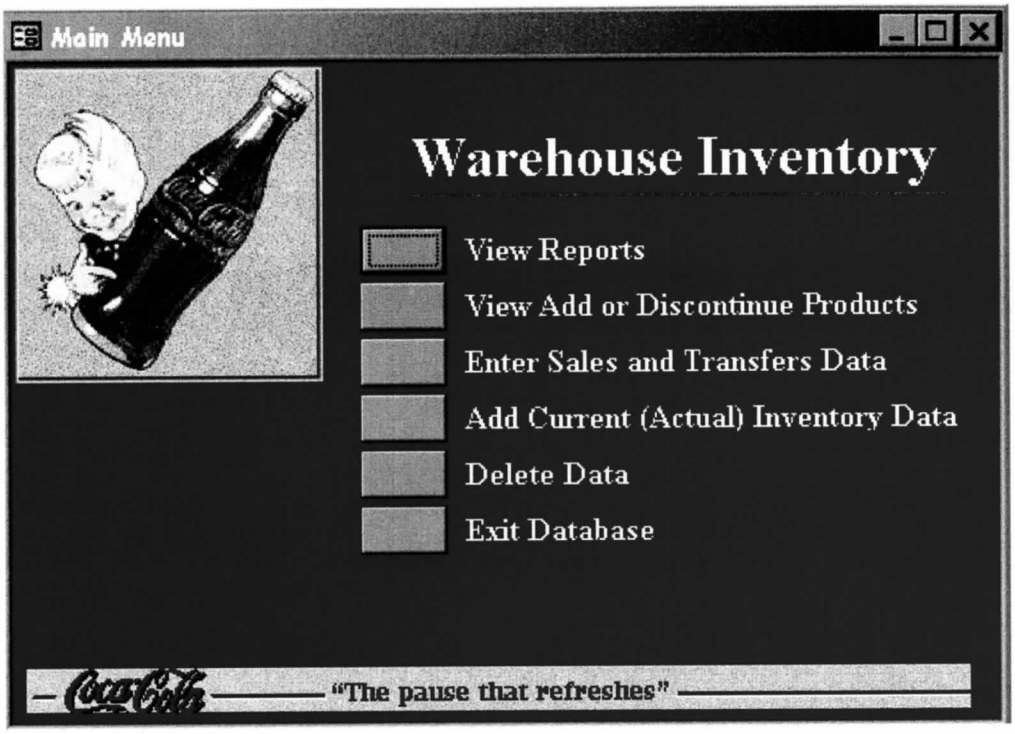


Figure 8 Main Menu Switchboard: Version 1

I also chose to hire a graphics designer to create one graphic file for use as a background image for some forms, as well as a report header graphic. Freelance graphics designer Maki Shinohara produced the graphic (figure 9) for a fee of twenty-five dollars.

Figure 9 Commissioned Graphic

With the program finalized as "Warehouse Inventory.mdb" version 1, the next step was to create the required manual (appendix A-4) and package the program files on compact disc for distribution. The Warehouse Inventory Manual.doc and Warehouse Inventory.mdb files were packaged with WinZip software and the resulting .zip file was built into an executable file with the directory path of c:\Warehouse Inventory. The resulting Warehouse Inventory.exe file and an explanatory Readme.doc were burned to compact disc using ahead Nero5 burning rom version 5.5.7.1 software. The Readme.doc file (appendices A-8, A-10) contains file definitions, software requirements, directory path information and creator details.

On October 29th I informed the CCBCSD manager that the project was complete and that we should meet to perform a review of the program and discuss when training sessions could begin. A meeting date of October 31st was planned and the program was deemed acceptable.

The meeting was held in Mitchell and I was informed that I could install the program on the Mitchell office computers, set up any networking tasks to share the database between the two computers and begin training the secretaries immediately. The database was installed on the least commonly used computer

and the c:\Warehouse Inventory directory was set to share with a required password using MS Windows 98 Network Neighborhood. Shortcut icons for the database were set up on each computer under each of the two secretaries personal desktops and the CCBCSD manger's personal desktop. The training session lasted forty-five minutes. During the training session I presented the manual in a clear sheet insert plastic folder. I gave a brief overview of the manual to the secretaries, and then proceeded to present the procedures for opening and accessing the database on both office PC's. Next, we reviewed how the switchboard menu worked, and I discussed navigation of the application through the switchboard forms. Each secretary was given a chance to enter data on the Products and Actual Inventory forms, shown how to add or discontinue products through the View/Discontinue/Add Products form, shown how to define time periods in creating reports and created and printed reports were reviewed and explained.

On November 1st I traveled to Pierre, South Dakota for setup and training. I found the Pierre PC to have an older version of the Windows 98 OS. Immediately, I learned that this version and the database switchboard were not working together. The program would open and all of the input and delete query forms worked. However, the switchboard would not open in any configuration. I proceeded with the training session, which lasted two hours. The manager at the Pierre office had only been in the position for "just over a month" and was not familiar with MS Access or 10-key. The training went well however, and I informed the manager that I would need to build a new switchboard for the Pierre

office. I informed the CCBCSD manager of the problem and that I would need to return to Pierre or have a new program disc mailed or delivered there once I determined the resolution to the problem. I was informed that this was satisfactory as there was "plenty of time" for me to resolve the issue as December was the agreed upon date for finalizing the project.

On November 5th I traveled to the Watertown sales center and again found the same older version of Windows 98 and experienced the same problem with the switchboard. I went through the training session with the Watertown manger as I had with the Pierre manager. The session lasted one and a half hours. The Watertown manager had several questions about the reports and also made some suggestions for any revisions to the program. I thanked him for the suggestions.

Due to the helpful suggestions I received in Watertown, I called the manager on November 6th to perform an impromptu interview. This was one of the more helpful experiences for myself thus far. The manager described his difficulty with the product list ordering and expressed that using the NORAND code numbers would be very helpful in his opinion. He explained that he had spent nine hours inputting one day's sales report numbers. This was his first attempted use of the program; however this seemed to be an extremely extensive amount of time. I asked if he might fax his reports to the Mitchell office so I could review how much data was inputted, and determine how much of the program I might need to change to remedy the problem we perceived.

I had scheduled an interview and review meeting with the secretaries in the Mitchell sales center for the 6th of November. When I arrived the faxed reports had arrived and it was fairly obvious that nine hours was indeed much too long for the data inputted. The secretaries expressed the same difficulties and also suggested changing to the NORAND code numbers as they have, through continued use, memorized most of the codes. Other suggestions included problems understanding where new products, which were added via the View/Add/Discontinue Products form were located in the record list. I was informed that it was too confusing to learn an entire new product numbering scheme. It was also explained that at least one of the secretaries assumed MS Access would be similar to MS Excel. Besides these problems they expressed great difficulty in understanding the reports and how to determine what the expected actual inventory in the warehouse should be from the two reports. There were also issues with the automatic date setting in the date fields on both the Actual Inventory Stock Take form and Book Inventory Input form. Having the current date populating the date fields was not acceptable for them and it was requested that the date fields be left blank. There were a few other minor details and requests discussed at the end of the meeting such as background images on forms. When we concluded the meeting I asked to meet with the CCBCSD manager to discuss the suggestions, requested changes and the non-functioning switchboard problem.

During the meeting I explained the problems with the CCBCSD manager. I explained that I most of the suggestions I had received could be easily

accomplished. I would create a new switchboard or menu board system, adjust background images and remove the automatic date code from date fields. We then discussed the problems involving the inventory reports. The manager described that the most important function of the database and the generated reports was presenting an expected or estimated inventory of each product in the warehouse. We determined that we had misunderstood each other on the main purpose for the database. The actual inventory counts that occur at the end of a month are reported and treated as beginning inventory for the new month. Upon understanding this it was now clear that the current version of the database was improper for the desired purposes. The database program would need some overhauling by reducing the number of tables and combining elements of the Stock Take table with the Inventory Transactions table.

By the next day November 7th the Stock_Take and Inventory Transactions tables had been combined with relevant fields and the new relationships specified. See figure ten for the version 2 table relationship diagram.

Table Relationship Diagram for Sales Center Warehouse Inventory version 2.0

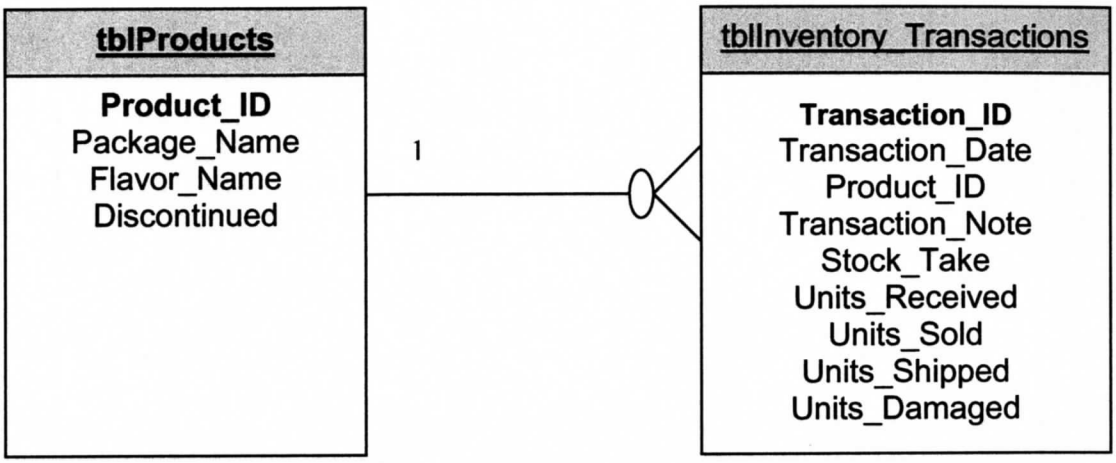


Figure 10 Relationship diagram version 2.0

The troublesome switchboard was removed and the database now consisted of only two tables: tblProducts and tblInventory_Transactions. Irrelevant forms were removed as was the delete query dealing with the non-existent Stock_Take table. Of considerable importance was how to make the new database version easier and more familiar for the employees. The most obvious solution would be instating the NORAND code system as the Product_ID in tblProducts. I requested a product list from the Mitchell sales center office secretaries, which was to include a complete list of the CCBCSD products, and associated NORAND codes. November 8th I received the new product list (appendix A-2) and began re-populating the tblProducts table. With the

NORAND identifier code associated with products I was sure no user would ever spend nine hours inputting a day's sales reports into the database.

The Book Inventory Input Form (figure 11) needed to be modified to reflect the new relationships and added fields of its associated table tblInventory_Transactions. A NORAND identifying object was placed on the form, providing a visual reference for the user confirming the code along with the product's Package name and Flavor name. A check box object denoting a product's current status was placed on the form as well with the discontinued label. A product may not be marked as discontinued in the Products table through this object; it serves only as a reference for its status. A "Quick NORAND Search" combo box object was also placed on the form. This object allows the user to quickly scroll through available codes and select the desired product by selecting its corresponding code. The "Quick NORAND Search" object allows the user to easily navigate large ranges of products quickly, thus reducing dependency on the record selector buttons, which become cumbersome and tedious when dealing with hundreds of records.

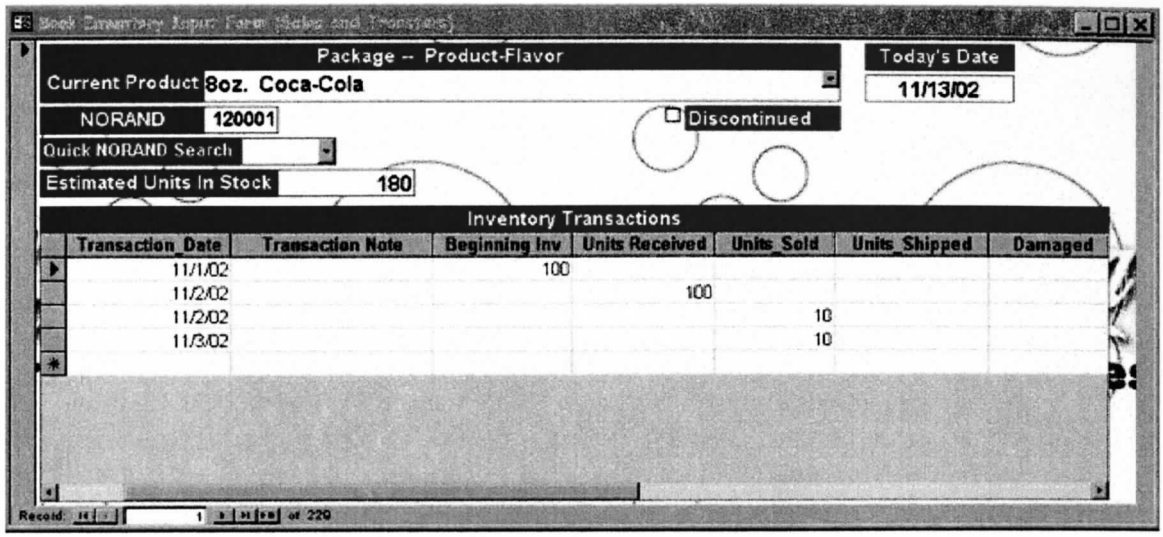


Figure 11 Book Inventory Form version 2.0

The Book Inventory Input subform also required changes. The Transaction_ID auto number field was hidden from view. The Unit_Count and Damaged_Product fields migrated version 1 tblStock_Take now demanded representation in the subform. Unit_Count became Stock_Take in the Inventory Transactions table and is represented as Beginning Inv on the subform. Damaged_Product became Units_Damaged in the Inventory Transactions table and is represented as Damaged on the subform.

The View/Add/Discontinue Products form remained virtually the same however; inputting a new product now requires the input of a non-duplicate NORAND code. The other change to this form involved replacing the previous background with a more subtle and less eye-straining image (figure 12).

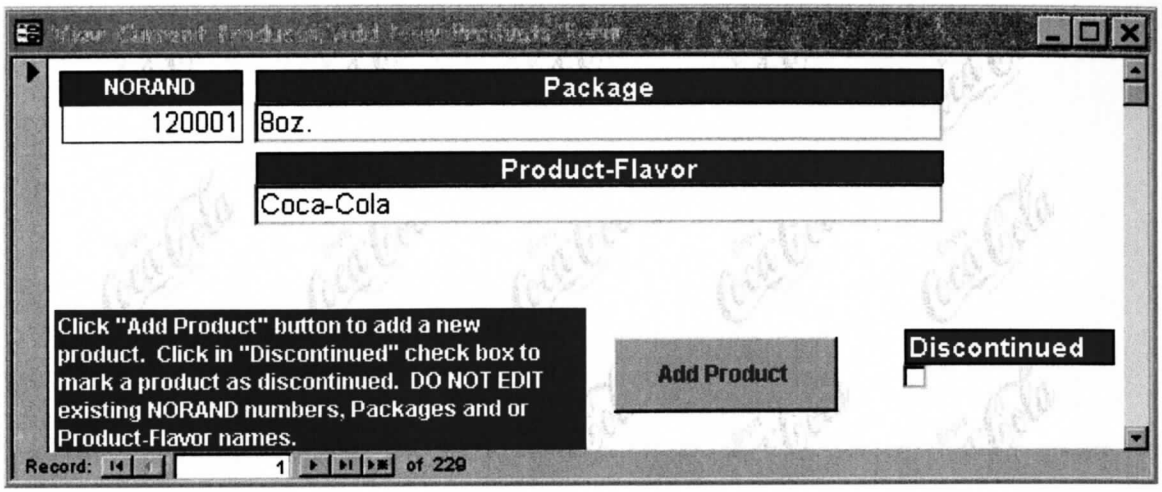


Figure 12 View Current Products/ Add New Products Form version 2.0

The final concerns now remaining involved creating a new menu system and rebuilding and removing reports. The switchboard system was replaced with individually created forms (figure 13), foregoing the aid of the switchboard manager and accompanying switchboard table.

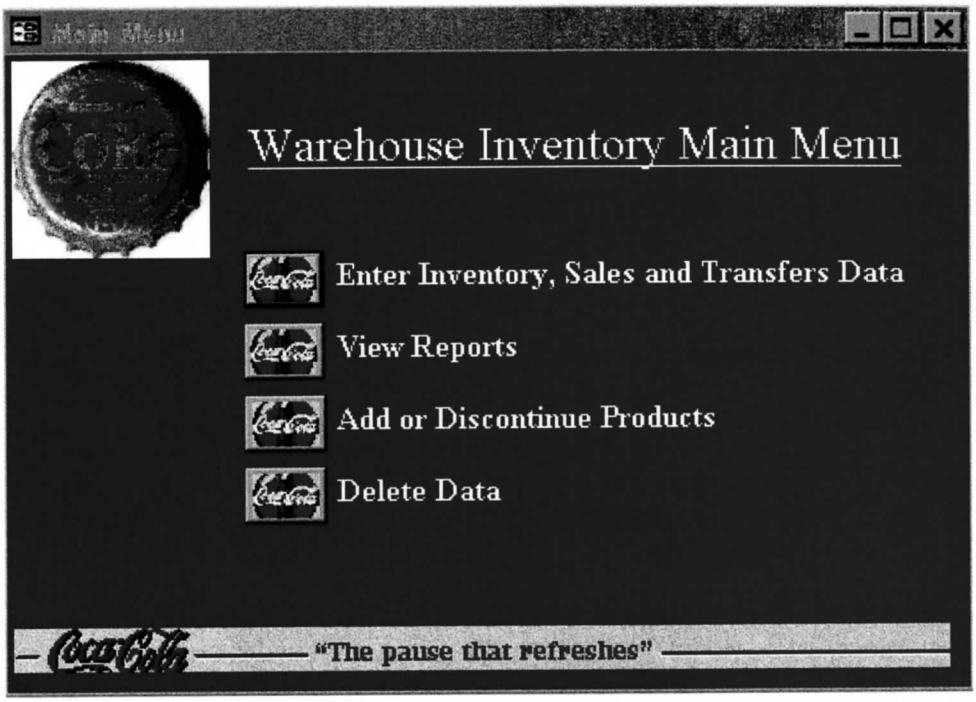


Figure 13 Main Menu Form version 2.0

The Actual Inventory reports were removed and the "Estimated Inventory report by Time Period and Product Report" (figure 14) was revised. "Beginning Inventory" and "Total Units Damaged" columns were added to the report. All report calculation objects were redesigned to reflect the new formulas. Extra report footer calculations reflecting totals for all fields were also added to this report. Appropriate formatting was also implemented.

Book Inventory Report

by Product and Month

For Transactions in the Month: November 2002

Package	Product-Flavor	Beginning Inventory	Total Units Received	Total Units Sold	Total Units Shipped	Total Units Damaged	Estimated Inventory
8oz.	Coca-Cola	100	50	10	0	0	140
KMX (24 8.4oz cans)	Orange	25	50	60	0	0	15
1 Liter (12 bottles)	Chippewa	500	0	300	0	0	200
1 Liter (12 bottles)	Dasani	100	30	60	0	0	70
2 Liter (8 bottles)	Vanilla Coke	96	60	65	0	0	91
Premix (4.75 gallons)	Fanta Tonic	65	25	75	0	0	15
Premix (4.75 gallons)	Schweppe's Club Soda	25	10	0	0	0	35
Postmix 2.5 gallon BIB	Fanta Sour	0	60	0	0	10	50
Postmix 2.5 gallon BIB	Fanta Tonic	0	100	0	-5	0	105
Postmix 2.5 gallon BIB	POWERade Mountain Blast	36	0	25	8	1	2
Postmix 5 gallon BIB	Nestea	0	6	4	0	3	-1
Postmix 5 gallon BIB	Barq's	25	36	10	5	1	45
	Totals:	**972**	**427**	**609**	**8**	**15**	**767**

Report Created On: 11/27/2002 6:02:57 AM

Page 1 of 1

Figure 14 Book Inventory Report by Time and Product version 2

The final designs for all components of the new version of the database were tested with sales and inventory report data. Sample reports with data were produced and printed and I met with the CCBCSD manager once more for approval of the new version. The changes made were accepted and I was provided with an appointment date of November 15th for the Mitchell sales center at which the secretaries and both the Pierre and Watertown managers were to be present. I was instructed to have all distribution materials for the three sales centers prepared and in hand at the meeting.

On November 12th work to prepare the distribution materials began. The new database required a revision of the manual, which was created and revised with new screenshots and instructions. See appendix (A-6) for the Sales Center Warehouse Inventory version 2 manual. The new manuals were printed and clear view pocket folders were purchased to serve as binders. Binders were purchased at 3 x $3.29 plus tax for a total of $10.66. With the new manual and database program finished a new Readme.doc was required and created for packaging with the new compact discs. With the extra time it was decided the compact discs and jewel cases should come with some graphics. A *Fellowes* Neato brand CD/DVD Labeling System kit was purchased for 19.99 plus tax for a total of $21.49. The coca-cola logo purchased from Maki Shinohara was used for both the CD label as well as the booklet cover. Each sales center would receive a custom labeled CD (figure 15) and jewel case with a custom booklet cover (figure 16). See appendix (A-9) for itemized costs.

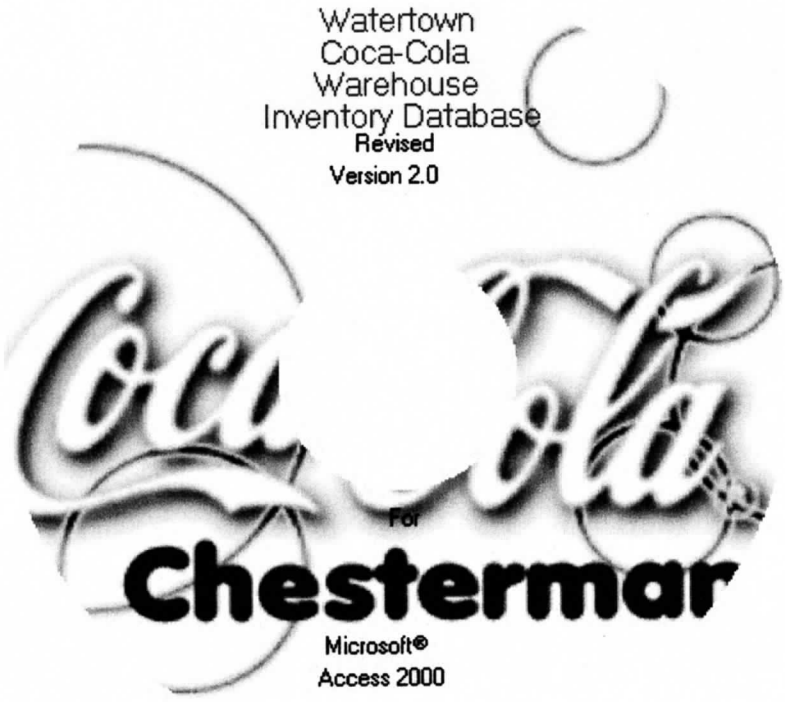


Figure 15 CD Label Graphic version 2.0

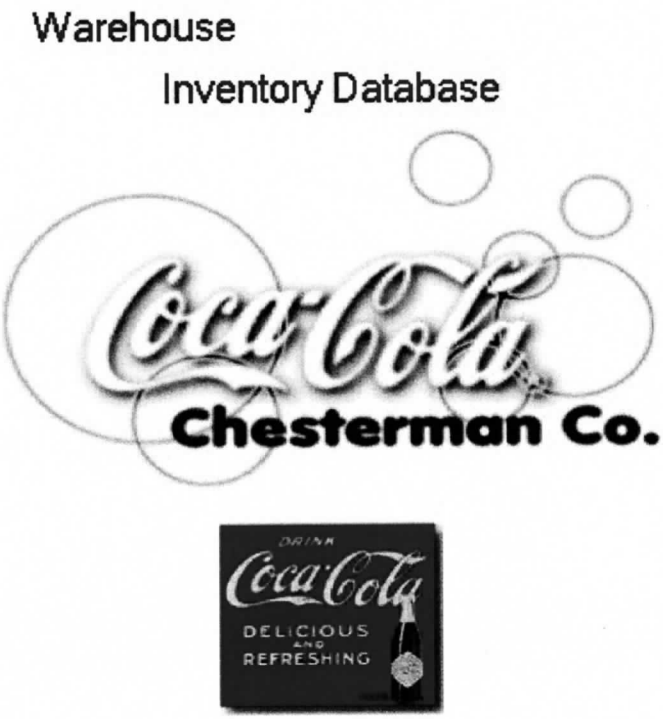


Figure 16 Booklet Cover Graphic Version 2

The final meeting and training sessions on November 15th were well received. I met with the Mitchell office secretaries first and presented them with the new materials. The new version of the database was installed and we spent twenty minutes discussing the changes and reviewing the new manual, forms, procedures and sample reports. Next, I met with the Pierre and Watertown managers. I presented them with the new materials, went through the process they would follow for installing the software and files at their individual sales centers, and we reviewed the changes and new manual. We discussed the reports and clarified what each column on the new Inventory report represented. The meeting with the managers lasted approximately thirty minutes. Finally, I asked the CCBCSD manager for one quick interview on the following Tuesday to

discuss issues the managers and secretaries may express or have questions on regarding the new version of the database program.

On Tuesday, November 19, 2002, I received a telephone call from the CCBCSD manager. He informed me that no issues or problems had occurred or been brought to his attention. The program and menu system was working well at all three sales centers and he had heard no complaints regarding design issues. He also informed me that the Watertown manager informed him that he could now input the sales report data for one day in the span of thirty minutes. The Pierre manager was able to install the database files and had spent Saturday entering the entire month to date sales and inventory reports data. That was thirteen days of sales reports and one day beginning inventory numbers. The manager revealed he had finished the tasks in less than nine hours.

Future Expansions and Possibilities

The explanation of options and general discussion of future possibilities is best understood, and perhaps realized, through actual demonstration. In other words, the best way to sell a product is to demonstrate a sample product. The databases, web formatted interfaces, table additions, and other expanded web specific applications can be demonstrated by creating a sample business web presence. With this in mind a working sample web site for the CCBCSD sales centers was created.

Considerations when designing a business web-site include many things. Initial considerations may include:

1. What content should be available?
2. Who will use the site?
3. How much will it cost in time/money?

These considerations are typically defined by going through the initial steps of:

1. Organization
2. Content
3. Appearance

Once the initial questions are answered details about setting up the actual site must be addressed. These details may include:

1. Determining the Server software and hardware.
2. Determining the Web site software.
3. Determining the Internet connection.

Here, steps one and two are closely related as web pages must function seamlessly with the server software. As well, the web pages must function with associated software, databases, applications etc. Internet connection details would include what type of connection the web server computer will use to connect to the Internet, as well as how users will locate the web site with domain naming and URL's.

To create a demonstration web site for the CCBCSD sales centers it was determined the theme based upon the use of Microsoft related software would continue. This decision was based upon the previous understanding of the

user's knowledge base, and also to reduce the amount of extra software which would need to be installed on sales center PC's. The following list details the hardware, software and Internet connection used to edit, run and access the demonstration web site:

- Microsoft Windows 2000 Professional Operating System (NTFS) permissions running on Intel Pentium III 733Mhz, 256 MB SDRAM, 22GB NTFS Partition Hard Drive
- Microsoft Internet Information Services 5.0
- Microsoft Internet Information Services Lockdown Tool
- Microsoft Office XP (Access 2002, Excel 2002, Frontpage 2002)
- Microsoft Jet 4.0 OLE DB Provider
- Microsoft Office Web Components 10.0
- Microsoft Data Access Components 2.7 (MDAC)
- Microsoft Internet Explorer version 5.0 or higher
- Microsoft Data Access Pages
- Active Server Pages (.asp)
- ASPProtect (lite version)
- Snitz Forums 2000 v. 3.4.03
- Midcontinent Communications residential Cable Internet Service
- DynDNS.org Members Network Information Center (Dynamic DNS service)
- .NET Framework version 1.1 (ASP.NET)
- MSDE Microsoft Desktop Engine a SQL type Database

- IBuySpy Portal
- Web Matrix (.NET editor from www.asp.net)

The home page address for the CCBCSD web site was registered with DynDNS.org as http://ccb.gotdns.com figure (17). DynDNS.org allows any public IP address to be stored with its database service, and lets the user supply the domain name for static or dynamic IP addresses. Custom domain names (i.e. yourname.com) may be purchased for a one time fee of thirty dollars on either static or dynamic IP addresses. Semi-custom domain names such as, yourname.gotdns.com, yourname.dyndns.org etc. are maintained free of charge by DynDNS.org and only require users to update configurations once every thirty five days to retain their unique domain name service. The home address and page are both freely accessible to the public over the Internet.

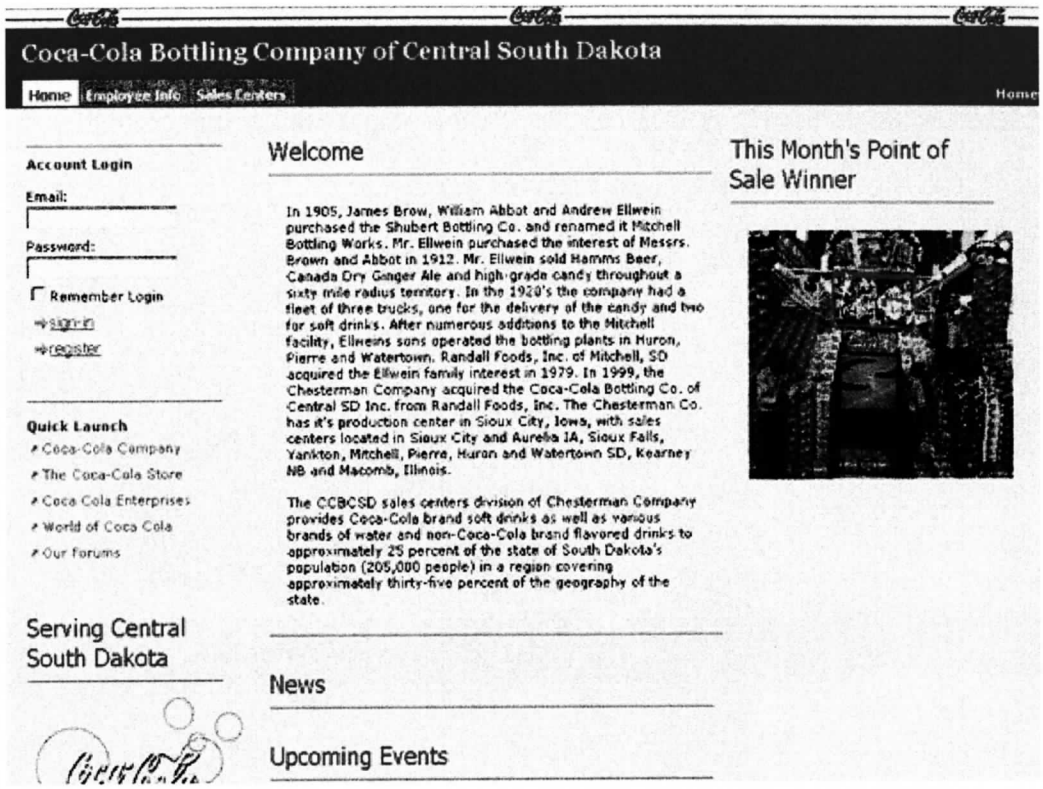


Figure 17 Home Page (http://ccb.gotdns.com/CCPortal)

The entire site is served via Microsoft Internet Information Server 5.0. Users access ASP, ASP.NET and HTML pages, such as the home page via the IUSR_Servername or ASPNET_account public Internet access accounts. The IUSR_Servername account is provided through IIS 5.0 anonymous access, and is setup using the Internet Services Manager snap-in. This account is given read and execute permissions. Similarly the ASPNET_account is created upon installation of the .NET Framework and is also given read and execute permissions on the database and active server pages.

Microsoft Access allows database designers the opportunity to create what are referred to as Data Access Pages (DAP). Data Access Pages are MS Access built HTML pages which allow database users the opportunity to interact

with an MS Access database through a web browser. DAP were initially designed by Microsoft to interact with Intranet based MS Access or MS-SQL databases which would be served from a network server or a locally stored client database. This configuration is referred to as two-tier architecture. Two-tier architecture is not however suited for serving DAP over the Internet. According to Mark Roberts in the MSDN article Deploying Data Access Pages on the Internet or Your Intranet,

"There are two reasons for this:

- You can't directly access either an Access database or a SQL Server database through Web protocols such as HTTP. And in most cases, these standard protocols are the only way for people outside of your company to access your site—security firewalls and proxy servers won't allow any other traffic.
- Even if physical access were possible, you probably wouldn't want it. If the database were accessible directly from your data access page, it would also be accessible from other pages or applications. This means that anybody who wanted to could access the data directly and view or manipulate your data in ways that you might not want."

To serve DAP over the Internet a different, three-tier architecture must be employed (figure 18).

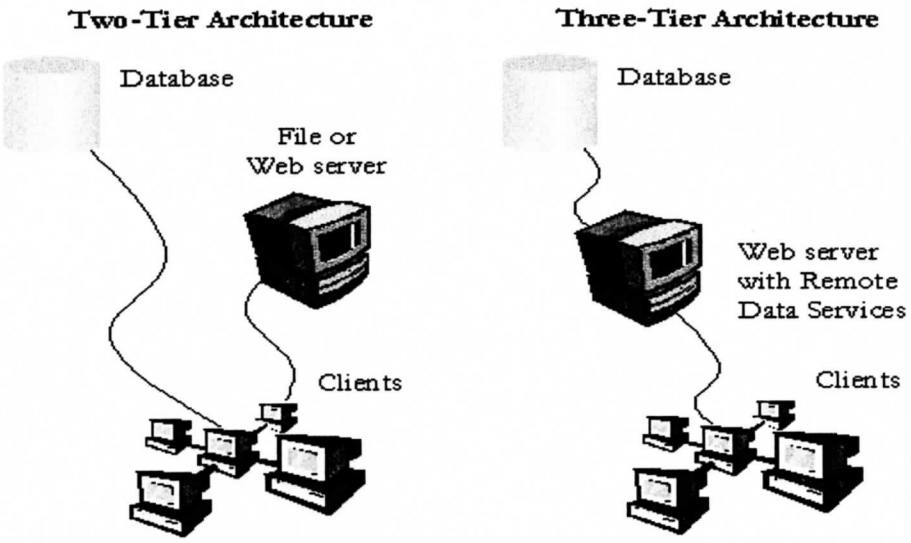


Figure 18 Database Architecture Models

To implement the DAP for the CCBCSD web site the three-tier architecture was adopted. By following the Microsoft Knowledge Base Article – 291783: ACC2002: How to Deploy Data Access Pages over the Internet, special disk and file configurations were performed. Required setup and configurations to the NTFS permissions, Windows 2000, IIS 5.0 server and files included:

1. Create a DAPInternetAccount via Computer Management snap-in.
2. Create NTFS read permissions for DAPInternetAccount on the Remote Data Services (RDS) folder (MSADC folder), and database (.mdb) files/folders.
3. Modify the Msdfmap.ini file to allow read/write access and sql statements against data sources on the server.
4. Set the page properties for every DAP to **UseRemoteProvider=True**
5. Give DAPInternetAccount full control permissions over all DAP.

To create the DAPInternetAccount, or any new user account, the Computer Management snap-in which is a Windows 2000 Administrative tool, is opened and the new user is added. When the new user is created passwords and password properties may also be defined. NTFS permissions may be assigned to any folder or file on an NTFS formatted partition. The permissions are set by right-clicking the folder or file, selecting properties and the security tab. Using these steps the DAPInternetAccount can be given read permissions on the MSADC folder in the web site. This is done to give the account access to the Remote Data Service components located within the MSADC folder. Full control permissions for the DAPInternetAccount are also needed for interacting with the MS Access database and its locking (.ldb) file when the database is opened. Modifying the Msdfmap.ini file only requires editing the file within a text editor with adjustments to two lines. In the **[connect default]** section **Access=NoAccess** is changed **to Access=ReadWrite**. This allows read and write connections to all data connections on the server. In the **[sql default]** section **sql=" "** is changed to **;sql=" "**. This allows any SQL statement against any data source on the server (ACC2002: How To Deploy Data Access Pages over the Internet). An important new setting for DAP in Office XP is **UseRemoteProvider**. By setting this attribute to the non-default setting of True the DAP connection string will search the server for the database and not the client. This means a connection string of D:/Database/Database.mdb will search for the connection on the server machine, and not on the accessing client machine. Finally, by giving Full control permissions to the DAPInternetAccount

on each DAP the accessing client/user will be able to read, write and modify data within the database through the DAP page controls. With these configurations to the server and DAP files, direct connections to the databases are possible.

With the three-tier architecture configured the web directories could be organized (figures 19, 20). A web folder named Mitchell was created to store the respective DAP's that would be used for inventory management and database manipulation. Publicly accessible DAP were placed in the wwwroot directory. Databases for the sales centers are stored outside the wwwroot directory ensuring outside users may not download or upload these files with any malicious intent.

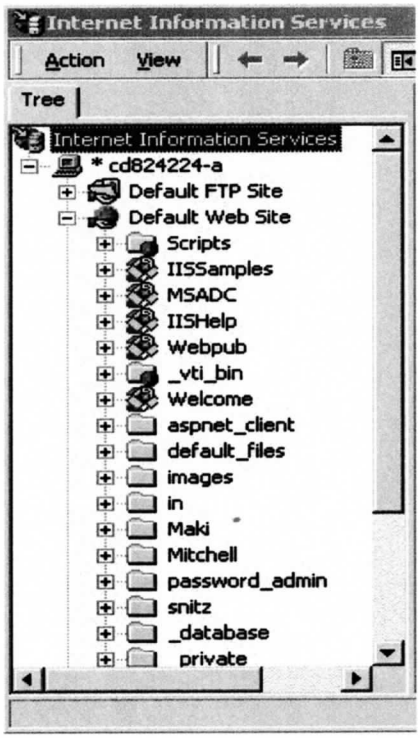


Figure 19 IIS Snap-In showing directory configuration of Server

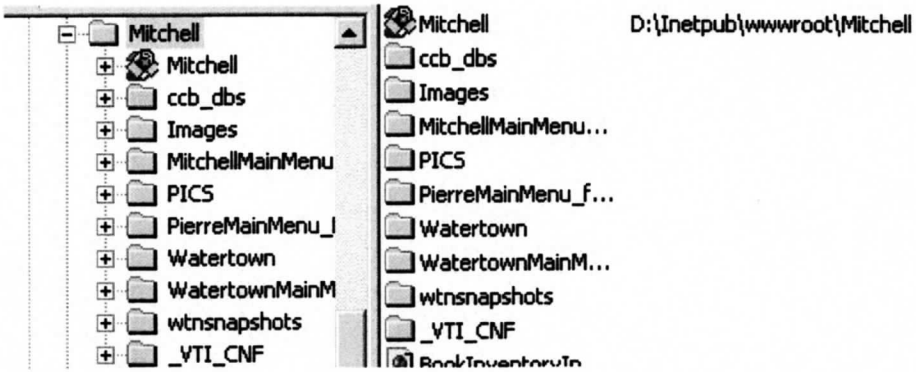


Figure 20 IIS Snap-In showing configuration of Inventory Management virtual directory (Mitchell)

Data Access Pages were designed within the database for each respective sales center (Pierre, Watertown, Mitchell). Each sales center is provided with its own main menu page containing links to pertinent DAP pages for the individual sales center (figure 21).

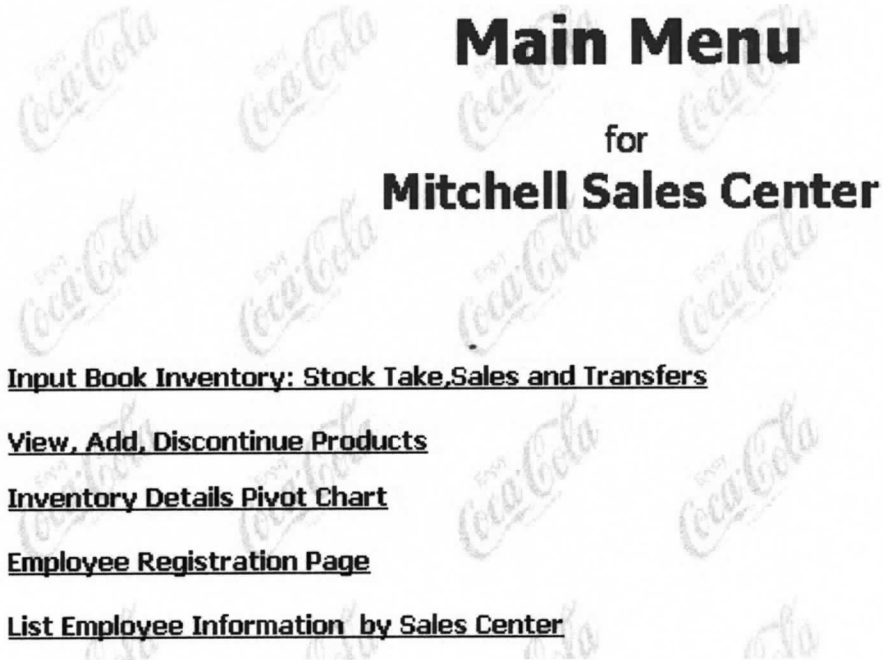


Figure 21 Inventory Management Users Main Menu DAP

The Input Book Inventory link sends the user to a DAP which performs similarly to the Book Inventory Input form in the version 2 database, and users may input beginning inventory, sales, shipments, and damaged products for any product into the database table tblInventory_Transactions (figure 22).

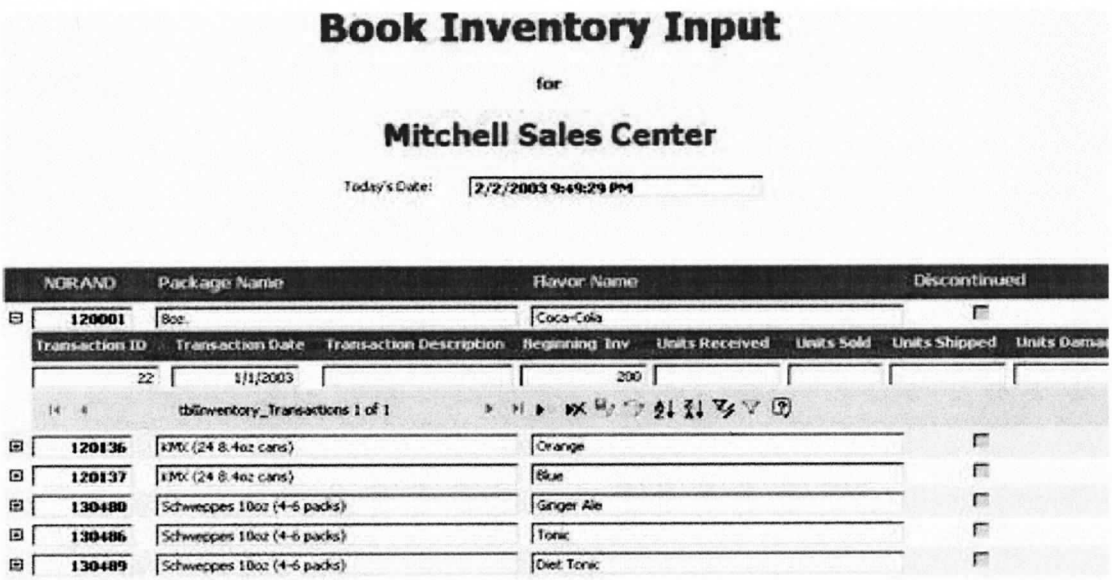
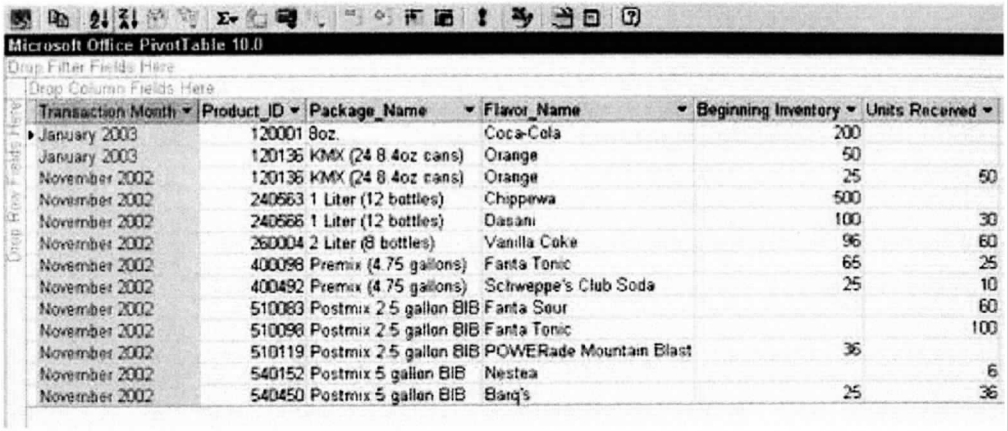


Figure 22 Inventory Transactions Input DAP

The View, Add, Discontinue Products link sends the user to a DAP similar to Book Inventory Input DAP, however no transaction data may be inputted, and the current products may be edited, viewed, marked as discontinued or new products may be added to the product list in the database (tblProducts).

The Inventory Details Pivot Chart link, sends the user to a DAP based on a query to the tblInventory_Transactions table and allows the user to view and also manipulate inventory transaction data (figure 23). The pivot table allows a user various calculation choices, data or field arrangement choices, and the choice to export portions or the entire table to MS Excel for further analysis.



Transaction Month	Product_ID	Package_Name	Flavor_Name	Beginning Inventory	Units Received
January 2003	120001	8oz.	Coca-Cola	200	
January 2003	120136	KMX (24 8.4oz cans)	Orange	50	
November 2002	120136	KMX (24 8.4oz cans)	Orange	25	50
November 2002	240563	1 Liter (12 bottles)	Chippewa	500	
November 2002	240586	1 Liter (12 bottles)	Dasani	100	30
November 2002	260004	2 Liter (8 bottles)	Vanilla Coke	96	60
November 2002	400098	Premix (4.75 gallons)	Fanta Tonic	65	25
November 2002	400492	Premix (4.75 gallons)	Schweppe's Club Soda	25	10
November 2002	510083	Postmix 2.5 gallon BIB	Fanta Sour		60
November 2002	510098	Postmix 2.5 gallon BIB	Fanta Tonic		100
November 2002	510119	Postmix 2.5 gallon BIB	POWERade Mountain Blast	36	
November 2002	540152	Postmix 5 gallon BIB	Nestea		6
November 2002	540450	Postmix 5 gallon BIB	Barq's	25	36

Figure 23 Office Web Components Pivot Table DAP for Inventory Transactions Data

As this web site is for demonstration of further extension and uses for the CCBCSD Inventory Database some new applications were created and or installed. A table to hold employee information was created (tblEmployees). A registration DAP for employees, as well as a pivot table DAP to display the employee data were designed to input and access the new table. The Employee Registration Page and List Employee Information by Sales Center links send the user to DAP for interacting with the new table (figures 24, 25).

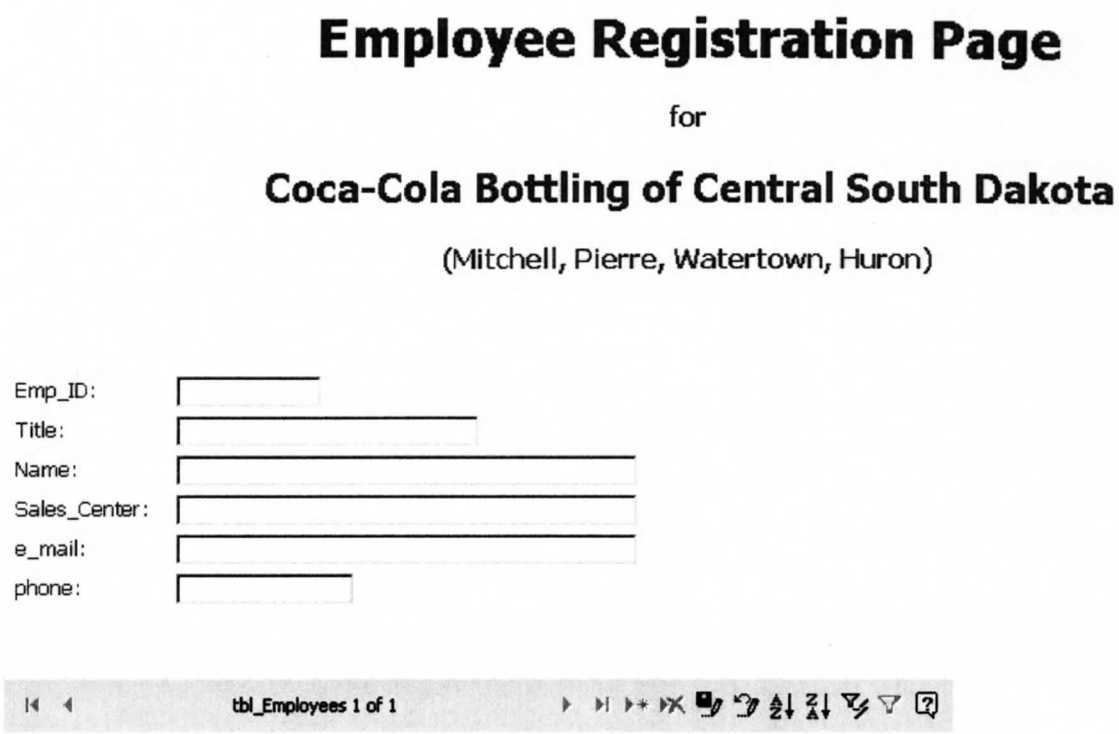


Figure 24 Employee Registration DAP

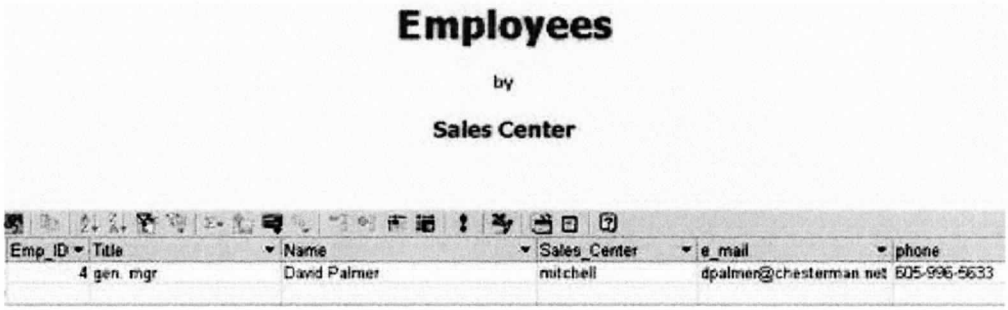


Figure 25 Office Web Components Pivot Table lists employee data per Sales Center

The second expansion chosen was a forum board. The Snitz Forums 2000 forum was installed (http://forum.snitz.com). The Snitz Forums store and access data via ASP pages and an MS Access Database using the Microsoft Jet 4.0. OLE DB Provider. The Snitz Forums 2000 is freeware and source code is available through a General Public License. Administration and customization of

the forum is simple and fairly straightforward through an administration menu. The forum was chosen because it required minimal setup configuration and uses the current Jet provider on the server. The addition of a forum seemed logical for the CCBCSD sales centers considering the several remote sales center locations. The forum was configured with one publicly accessible category for forums and one private category and forum for manager-only use and view. Internet forums provide users with a virtual bulletin board where they may carry out extended conversations, give quick notices and also allow access from anywhere in the world. The public URL for the CCBCSD forum is http://ccb.gotdns.com/snitz. Refer to figure 26 for a screenshot of the CCBCSD forum main page.

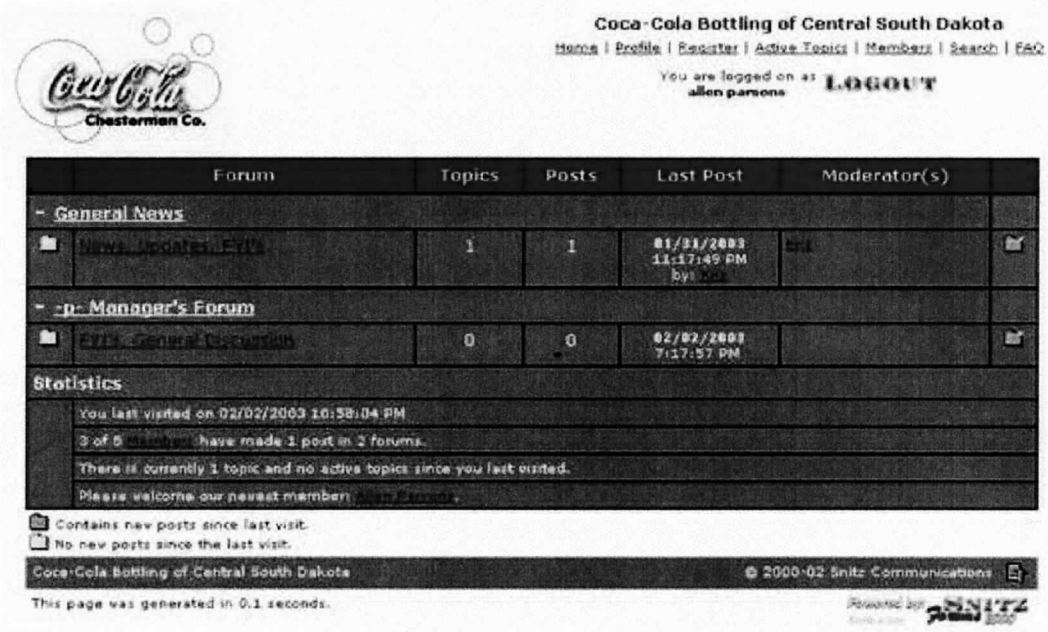


Figure 26 Snitz Forums 2000 for CCBCSD (view for a logged on manager)

Because this is a business web site and there exists business critical data and means to access and manipulate critical data, security must be a consideration. Good web site security is created and maintained through several tools and configurations. First, all server software security patches should be installed. Second, access to sensitive data files should be minimized as much as possible. Third, sample admin scripts and extraneous unused files should be removed. Finally, server logging should be enabled.

As this demonstration site resides in a Windows 2000 environment with NTFS permissions all folders in the server's root directory were checked to remove the Everyone user, and the minimum number of accounts with write, modify and control privileges were assigned. Directory browsing was also disabled. Files which provide access to the database were given long, complex names to reduce URL guessing. With this completed the IIS lockdown tool was used to remove unseen files and directories, turn on server logging, implement URLscan (which disallows users to scan for directory structures and should block most webbots), and shutdown the SMTP and NNTP services.

Because both public and private access is provided by the web site an ASP password system was implemented to gain access to the Inventory Management Welcome page. An ASP connected password database controls access to this page. The ASPProtect (Lite Version) database and ASP scripts were used to protect this page in the web site. The database holds usernames and passwords for users and the majority of the ASPProtect script code is encrypted. To further protect this database a password to open the database file

was created and a new admin account and password were setup. By performing these steps the only method to open the database file would require access to the system.mdw security file which is located on the server and nowhere near the web directory. This removes the possibility of anyone gaining usernames and passwords by being lucky enough to download the password database. Using the ASPProtect password method is by no means the only method for protecting pages. The DAPInternetAccount user account could be configured with a password on the standard NTFS access control list (ACL). A user would then be required to enter DAPInternetAccount as username and the password to gain access to the DAP. Alternatively individual user accounts might also be created and added to the permissions list for access to the DAP. Another alternative for restricting access to critical pages might also involve using ASP.NET login methods. This method requires installation and configuration of MS-SQL server and or MSDE, and the .NET Framework. The ASP pages which are deemed available to the public, registered users or power users can be marked as such through the admin portal (figure 27).

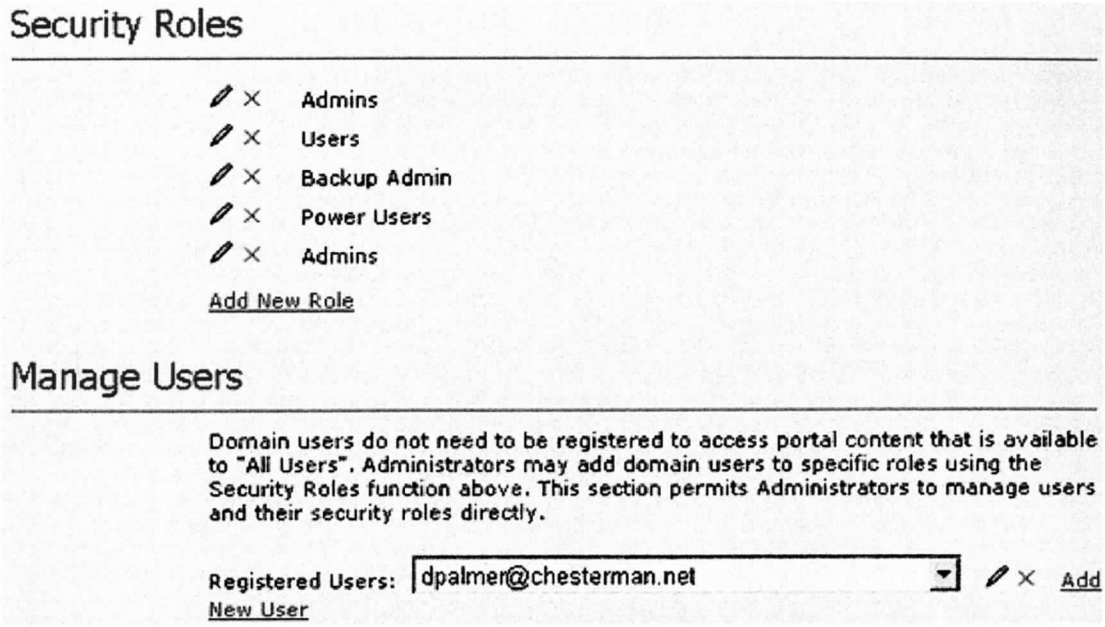


Figure 27 User Management tools within ASP.NET Admin tools

By implementing the Security Roles feature the administrator is able to assign rights and privileges to any number of user groups. Through the Manage Users feature the administrator is able to assign registered users to these groups, thus allowing unique users different rights and privileges. Anyone reaching the home page has the right to register as a user; however, registering as a user does not allow that user to gain any extra privileges without authorization from the site administrator. Once a user is registered the administrator may assign that user to a privileged user group providing greater access within the site to that user. The following list details a user's progression to a privileged group and details the additional rights granted to the user.

1. User registers with the site with an email of tom@chesterman.com and desired password.

2. Administrator recognizes tom@chesterman.com as a user that will need access to most all features and pages within the site, adds the account to the Power Users group, and emails the user an acknowledgement message confirming privileges granted.
3. The user now gains access to the Product News, Inventory, and Uploads tabs/pages, as well as, privileges to edit most content within the site.

This user is now granted access to nearly all site features barring administration. However, the user does not have access to the main inventory management welcome page, or any inventory DAP. If the user requires access to the DAP sections of the site he must request a password for the welcome page which is protected with the previously mentioned ASPProtect database login.

With a working web site which requires certain software applications and utilities for proper access, an FAQ or frequently asked question page was designed. The FAQ includes details along with information aimed at the CCBCSD employees who would be using the DAP for data entry and manipulation. The basic requirements for viewing and working with the CCBCSD web site include:

1. Microsoft Internet Explorer version 5.0 or higher
2. Microsoft Office Web Components 10
4. Microsoft Data Access Components MDAC version 2.5 or higher

Further explanation and links to utility and software installers are provided in the FAQ pages.

I believe this web site configuration could work as an alternative to the straight MS Access database and may provide some helpful ideas or spark other ideas with the CCBCSD managers as to what future expansion of the database might include. However, I don't believe that I would want to host the web site from my own server and limited bandwidth Internet connection. With the files and directory structure a company server might very well be configured to host the web site, and I would be more than happy to work with any server administrator to ensure setup.

Results and Conclusions

This project to create and implement the **Coca-Cola Bottling of Central South Dakota Sales Center Warehouse Inventory Database** using MS Access 2000 software on Microsoft Windows 98 Operating system PC's was completed, and satisfied all goals ten business days ahead of December 1, 2002. I have offered to maintain and rebuild certain customizations for this project. I have also offered to provide advice and or offer expansion of the project if new uses or methods of use for the database program are desired and deemed applicable.

Customizations for the project would involve minor user specific details such as font sizes and colors, date formats, and form sizing. Other customizations may involve creating new custom reports or adding/removing fields from the current tables. Rebuilding or updating the database tables and adding new tables creating new relationships could also be foreseen. Individual

users normally have individual tastes and requirements. Font sizes or colors in the final version of this project may not be agreeable with some users and could be customized individually at the sales centers. Date formatting and whether particular users would benefit from the presence or absence of a date in a required date field are both foreseeable customizations, which might be implemented. New tables or table fields may also be desired in the future.

I have explained these options for future expansion and customization with the CCBCSD manager, and some interest has been shown. Most interest has been received regarding the sharing of the database among the sales centers. Expressed interest in remotely accessing report data from any of the sales centers has been discussed briefly, however at this time implementation of the database program, getting employees and managers to "speed" with the program remained the greatest concern. The decision for future expansions and customizations ultimately resides with the CCBCSD managers, and as explained this database was requested simply as a means of recourse for sales centers to better understand the radical inventory discrepancies over the past twelve to thirty two months.

Personally this project challenged me in several ways and I learned many things. I learned how difficult undertaking a project with unfamiliar software can be. I also learned that the customer does not always know what they need, and helping a customer realize their needs can be challenging, yet helpful. I had little to no experience with most of the software used in this project when I began and now feel much more comfortable with a great deal more applications and

software suites because of it. I also learned some software and scripting, especially ASP, which were not used in the overall project but only tested as possible alternative solutions. I am grateful for the instruction I received during my master's studies and found myself using several methods learned throughout the course of those studies. Project guidelines, flow charting and work scheduling were among some of the methods I found most worthy during the project. But two very big lessons, which I took into the project at the beginning, were the recognition of project scope and scope creep. I worked hard to make sure not to promise things which were off-line from the project, and which could quite easily have set back project completion dates. Sometimes I had no choice, and unexpected extras were thrown into the project, and as the project progressed I realized that these were very good considerations to take into the project. A simple project can turn into a large complex maze rather quickly if attention is not paid to the basic fundamental details and deliverables, which are determined at the beginning.

There are certainly other methods and software solutions which could be used when considering a web presence for the CCBCSD database. Different server software and database connections or database software could be used. Certainly, entirely ASP or PHP based scripts and pages could be used to access and manipulated the data. However, I tried to stay as close as possible to the original guidelines by using Microsoft and Microsoft software solutions. I do believe these databases, software files, and the directory configurations could be

implemented on a CCBCSD or Chesterman Company server to serve as an alternative to the MS Access databases.

References

Microsoft Press. Microsoft Access 2000 Step by Step. Redmond, WA: Catapult Inc., 1999.

Utter Access Discussion Forums. Utteraccess.com. Copyright 2002 < http://www.utteraccess.com>.

August 8, 2002. ACC2002: How to Deploy Data Access Pages over the Internet. Accessed January 15, 2003. <http://support.microsoft.com/default.aspx?scid=kb;EN-US;291783>.

Roberts, Mark. September, 2001. Deploying Data Access Pages on the Internet or Your Intranet. Accessed January 14, 2003. <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnacc2k2/html/deploydap.asp>.

Non-Microsoft Software

Williams, Chris. ASPProtect(Lite version). Accessed January 20, 2003. <http://www.ASPProtect.com>.

Anderson, Michael. Gorissen, Pierre et. al. Snitz™ Forums 2000 Version 3.4.03. Accessed January 25, 2003. <http://forum.snitz.com>

Appendices

A-1 (Order loadsheet)

Delivery Date: **City:** **Driver:** **Load#:**

16oz.		20 oz.		6 Packs		12 Packs	
Sweet Tea no/lemon		Classic		Classic		Classic	
Unsweet Tea w/lemon		Vanilla Coke		Diet Coke		CF Classic	
Sweet Tea w/lemon		Diet Coke		CFDC		Vanilla Coke	
Sweet Tea w/raspberry		Diet Coke w/Lemon		Cherry Coke		Diet Coke	
100% Orange Juice		CFDC		Pibb Xtra		Diet Coke w/Lemon	
100% Apple Juice		Cherry Coke		Sprite		CFDC	
Ruby Red Grapefruit		Pibb Xtra		Diet Sprite		Cherry Coke	
Cran/Apple/Rasp.		Sprite		Mello Yello		Diet Cherry Coke	
Cran/Grape		Diet Sprite		Fanta Orange		Pibb Xtra	
12 oz. Powerade		Mello Yello		Fanta Grape		Diet Mr. Pibb	
Lemon Lime		Diet Mello Yello		Barq's		Sprite	
Mtn. Blast		Cherry Mello Yello		Dr. Pepper		Diet Sprite	
Fruit Punch		Melon Mello Yello		Diet Dr. Pepper		Mello Yello	
20 oz. Powerade		Fresca		**24 Packs**		Diet Mello Yello	
Lemon Lime		Fanta Orange		Classic		Fanta Orange	
Fruit Punch		Fanta Grape		Diet Coke		MM Diet Orange	
Mtn. Blast		Fanta Strawberry		CFDC		Fanta Grape	
Artic Shatter		MM Lemonade		Cherry Coke		Fanta Strawberry	
Green Squall		MM Lemonade Light		Sprite		MM Lemonade	
Jagged Ice		MM Pink Lemonade		Diet Sprite		MM Lemonade Light	
Infrared Freeze		Barq's		Mr. Pibb		MM Pink Lemonade	
24 oz. Powerade		Nestea Cool		Mello Yello		Citra	
Lemon Lime		Dr. Pepper		Barq's		Fresca	
Fruit Punch		Diet Dr. Pepper		Dr. Pepper		Barq's	
Mtn. Blast		**1/2 Liters**		Diet Dr. Pepper		Diet Barq's	
Artic Shatter		Classic		**Chippewa**		Nestea Cool	
Green Squall		Diet Coke		300 ml		Dr. Pepper	
Jagged Ice		CFDC		1/2 Liter 12 pack		Diet Dr. Pepper	
Infrared Freeze		Sprite		20 oz		**Premix**	
32 oz Powerade		Mr. Pibb		24 oz		Classic	
Lemon Lime		Mello Yello		1 Liter		Diet Coke	
Fruit Punch		**1 Liters**		**Dasani**		Sprite	
Mtn. Blast		Classic		1/2 Liter 6 pack		Mr. Pibb	
Artic Shatter		Diet Coke		12 oz 12 pack		Mello Yello	
Green Squall		Cherry Coke		20 oz		Barq's	

Jagged Ice		Sprite		1 Liter		Tonic	
Infrared Freeze		Mr. Pibb		**Evian**		Club Soda	
Powdered Powerade		Mello Yello		1/2 Liter loose		Sour	
Fruit Punch 2 GAL		Dr. Pepper		1 Liter		**BIB 5 Gal.**	
Mtn. Blast 2 GAL		**2 Liters**		**KMX**		Classic	
Lemon Lime 2GAL		Classic		Orange		Diet Coke	
Schweppes 10 oz		Diet Coke		Blue		CFDC	
Tonic		CFDC		**BIB 2.5**		Cherry Coke	
Club Soda		Cherry Coke		Tonic		Sprite	
Ginger Ale		Sprite		Sour		Mr. Pibb	
Fruitopia		Diet Sprite		HIC Fruit Punch		Mello Yello	
Strawberry Passion		Mr. Pibb		Raspberry Tea		MM Lemonade	
Cherry Vanilla		Mello Yello		Powerade Ft. Punch		MM Orange	
Kiwiberry		MM Orange		Powerade Mtn. Blast		Nestea	
Planet Java		Barq's		**FCB 2.5**		HIC Pink Lemonade	
Tremble		Dr. Pepper		Classic		Barq's	
Javadelic				MM Cherry		Dr. Pepper	
Caramocha				Blue/Raspberry			
Miscellaneous:							

A-2 (Product List/Products by ID Report)

All Products

by ID

NORAND	Package	Product-Flavor
120001	8oz.	Coca-Cola
120136	KMX (24 8.4oz cans)	Orange
120137	KMX (24 8.4oz cans)	Blue
130480	Schweppes 10oz (4-6 packs)	Ginger Ale
130486	Schweppes 10oz (4-6 packs)	Tonic
130489	Schweppes 10oz (4-6 packs)	Diet Tonic
130492	Schweppes 10oz (4-6 packs)	Club Soda
140553	1/3 Liter	Evian
150563	300 ml	Chippewa
160563	12 oz (2-12packs)	Chippewa
160566	12 oz (2-12packs)	Dasani

161055	Minute Maid (4-6pk cans)	Apple Juice
161061	Minute Maid (4-6pk cans)	Orange Juice
161119	POWERade 12oz. (4-6pk bottles)	Mountain Blast
161121	POWERade 12oz. (4-6pk bottles)	Lemon Lime
161122	POWERade 12oz. (4-6pk bottles)	Arctic Shatter
161124	POWERade 12oz. (4-6pk bottles)	Fruit Punch
180167	Planet Java 9.5oz (12 bottles)	Tremble
180168	Planet Java 9.5oz (12 bottles)	Mojovation
180169	Planet Java 9.5oz (12 bottles)	Carameal
210055	Minute Maid Juices 16oz (24	Apple Juice
210061	Minute Maid Juices 16oz (24	100% Orange Juice
210065	Minute Maid Juices 16oz (24	Apple\Cran\Raspberry
210075	Minute Maid Juices 16oz (24	Cran\Grape
210077	Minute Maid Juices 16oz (24	Ruby Red Grapefruit
210148	Nestea 16oz (24 bottles)	Raspberry
210154	Nestea 16oz (24 bottles)	Sweetened
210156	Nestea 16oz (24 bottles)	Sweetened w/ Lemon
210162	Nestea 16oz (24 bottles)	Unsweetened
210553	Loose 16oz	Evian
210625	16oz (24 bottles)	Yoohoo
212563	16oz 12pk bottles (2 packs)	Chippewa
213001	0.5 Liter (4-6pk bottles)	Coca-Cola
213002	0.5 Liter (4-6pk bottles)	Diet Coke
213005	0.5 Liter (4-6pk bottles)	CF Diet Coke
213017	0.5 Liter (4-6pk bottles)	Mr. Pibb
213018	0.5 Liter (4-6pk bottles)	Sprite
213025	0.5 Liter (4-6pk bottles)	Mello Yello
213566	0.5 Liter (4-6pk bottles)	Dasani
220001	20oz (24 bottles)	Coca-Cola
220002	20oz (24 bottles)	Diet Coke
220003	20oz (24 bottles)	Diet Coke w/ Lemon
220004	20oz (24 bottles)	Vanilla Coke
220005	20oz (24 bottles)	CF Diet Coke
220007	20oz (24 bottles)	Diet Vanilla Coke
220010	20oz (24 bottles)	Cherry Coke
220011	20oz (24 bottles)	Diet Cherry Coke
220018	20oz (24 bottles)	Sprite
220019	20oz (24 bottles)	Pibb Xtra

220021	20oz (24 bottles)	Diet Sprite
220025	20oz (24 bottles)	Mello Yello
220026	20oz (24 bottles)	Diet Mello Yello
220027	20oz (24 bottles)	Mello Yello Cherry
220028	20oz (24 bottles)	Mello Yello Melon
220030	20oz (24 bottles)	Fresca
220059	20oz (24 bottles)	MM Pink Lemonade
220070	20oz (24 bottles)	MM Lemonade
220071	20oz (24 bottles)	MM Lemonade Lite
220089	20oz (24 bottles)	Fanta Orange
220095	20oz (24 bottles)	Fanta Grape
220101	20oz (24 bottles)	Fanta Strawberry
220119	POWERade 20oz (24 bottles)	Mountain Blast
220121	POWERade 20oz (24 bottles)	Lemon Lime
220122	POWERade 20oz (24 bottles)	Arctic Shatter
220124	POWERade 20oz (24 bottles)	Fruit Punch
220130	POWERade 20oz (24 bottles)	Jagged Ice
220131	POWERade 20oz (24 bottles)	Green Squall
220133	POWERade 20oz (24 bottles)	Infrared Freeze
220164	20oz (24 bottles)	Nestea Cool
220185	20oz (24 bottles)	Fruitopia Straw. Passion
220187	20oz (24 bottles)	CHY Vanilla Groove
220194	20oz (24 bottles)	Kiwi Berry
220450	20oz (24 bottles)	Barq's
220460	20oz (24 bottles)	Dr. Pepper
220461	20oz (24 bottles)	Diet Dr. Pepper
220465	20oz (24 bottles)	Red Fusion
220563	20oz (24 bottles)	Chippewa
220566	20oz (24 bottles)	Dasani
223563	20oz (24 bottles)	Jack's Chippewa Water
240001	1 Liter (12 bottles)	Coca-Cola
240002	1 Liter (12 bottles)	Diet Coke
240010	1 Liter (12 bottles)	Cherry Coke
240018	1 Liter (12 bottles)	Sprite
240019	1 Liter (12 bottles)	Pibb Xtra
240021	1 Liter (12 bottles)	Diet Sprite
240025	1 Liter (12 bottles)	Mello Yello
240460	1 Liter (12 bottles)	Dr. Pepper

240461	1 Liter (12 bottles)	Diet Dr. Pepper
240553	1 Liter (12 bottles)	Evian
240563	1 Liter (12 bottles)	Chippewa
240566	1 Liter (12 bottles)	Dasani
260001	2 Liter (8 bottles)	Coca-Cola
260002	2 Liter (8 bottles)	Diet Coke
260004	2 Liter (8 bottles)	Vanilla Coke
260005	2 Liter (8 bottles)	CF Diet Coke
260010	2 Liter (8 bottles)	Cherry Coke
260018	2 Liter (8 bottles)	Sprite
260019	2 Liter (8 bottles)	Pibb Xtra
260021	2 Liter (8 bottles)	Diet Sprite
260025	2 Liter (8 bottles)	Mello Yello
260089	2 Liter (8 bottles)	Fanta Orange
260450	2 Liter (8 bottles)	Barq's
260460	2 Liter (8 bottles)	Dr. Pepper
260461	2 Liter (8 bottles)	Diet Dr. Pepper
270119	POWERade 32oz (12 bottles)	Mountain Blast
270121	POWERade 32oz (12 bottles)	Lemon Lime
270122	POWERade 32oz (12 bottles)	Arctic Shatter
270124	POWERade 32oz (12 bottles)	Fruit Punch
270130	POWERade 32oz (12 bottles)	Jagged Ice
270131	POWERade 32oz (12 bottles)	Green Squall
270133	POWERade 32oz (12 bottles)	Infrared Freeze
270134	POWERade 32oz (12 bottles)	Andean Chill
280119	POWERade 24oz	Mountain Blast
280121	POWERade 24oz	Lemon Lime
280122	POWERade 24oz	Arctic Shatter
280124	POWERade 24oz	Fruit Punch
280130	POWERade 24oz	Jagged Ice
280131	POWERade 24oz	Green Squall
280133	POWERade 24oz	Infrared Freeze
280134	POWERade 24oz	Andean Chill
283563	24oz	Chippewa
330001	6 Pack (4-6pk cans)	Coca-Cola
330002	6 Pack (4-6pk cans)	Diet Coke
330003	6 Pack (4-6pk cans)	Diet Coke w/ Lemon
330004	6 Pack (4-6pk cans)	Vanilla Coke

330005	6 Pack (4-6pk cans)	CF Diet Coke
330006	6 Pack (4-6pk cans)	CF Coca-Cola
330010	6 Pack (4-6pk cans)	Cherry Coke
330011	6 Pack (4-6pk cans)	Diet Cherry Coke
330015	6 Pack (4-6pk cans)	Tab
330018	6 Pack (4-6pk cans)	Sprite
330019	6 Pack (4-6pk cans)	Pibb Xtra
330021	6 Pack (4-6pk cans)	Diet Sprite
330025	6 Pack (4-6pk cans)	Mello Yello
330089	6 Pack (4-6pk cans)	Fanta Orange
330450	6 Pack (4-6pk cans)	Barq's
330460	6 Pack (4-6pk cans)	Dr. Pepper
360001	12 Pack (2-12pk cans)	Coca-Cola
360002	12 Pack (2-12pk cans)	Diet Coke
360003	12 Pack (2-12pk cans)	Diet Coke w/ Lemon
360004	12 Pack (2-12pk cans)	Vanilla Coke
360005	12 Pack (2-12pk cans)	CF Diet Coke
360006	12 Pack (2-12pk cans)	CF Coca-Cola
360007	12 Pack (2-12pk cans)	Diet Vanilla Coke
360010	12 Pack (2-12pk cans)	Cherry Coke
360011	12 Pack (2-12pk cans)	Diet Cherry Coke
360018	12 Pack (2-12pk cans)	Sprite
360019	12 Pack (2-12pk cans)	Pibb Xtra
360020	12 Pack (2-12pk cans)	Diet Mr. Pibb
360021	12 Pack (2-12pk cans)	Diet Sprite
360025	12 Pack (2-12pk cans)	Mello Yello
360026	12 Pack (2-12pk cans)	Diet Mello Yello
360030	12 Pack (2-12pk cans)	Fresca
360039	12 Pack (2-12pk cans)	Citra
360059	12 Pack (2-12pk cans)	MM Pink Lemonade
360070	12 Pack (2-12pk cans)	MM Lemonade
360071	12 Pack (2-12pk cans)	MM Lemonade Lite
360089	12 Pack (2-12pk cans)	Fanta Orange
360095	12 Pack (2-12pk cans)	Fanta Grape
360101	12 Pack (2-12pk cans)	Fanta Strawberry
360164	12 Pack (2-12pk cans)	Nestea Cool
360450	12 Pack (2-12pk cans)	Barq's
360451	12 Pack (2-12pk cans)	Diet Barq's

360460	12 Pack (2-12pk cans)	Dr. Pepper
360461	12 Pack (2-12pk cans)	Diet Dr. Pepper
360465	12 Pack (2-12pk cans)	Red Fusion
380001	18 Pack (cans)	Coca-Cola
380002	18 Pack (cans)	Diet Coke
380018	18 Pack (cans)	Sprite
390001	24 Pack (cans)	Coca-Cola
390002	24 Pack (cans)	Diet Coke
390005	24 Pack (cans)	CF Diet Coke
390017	24 Pack (cans)	Mr. Pibb
390018	24 Pack (cans)	Sprite
390019	24 Pack (cans)	Pibb Xtra
390021	24 Pack (cans)	Diet Sprite
390025	24 Pack (cans)	Mello Yello
390450	24 Pack (cans)	Barq's
390460	24 Pack (cans)	Dr. Pepper
390461	24 Pack (cans)	Diet Dr. Pepper
400001	Premix (4.75 gallons)	Coca-Cola
400002	Premix (4.75 gallons)	Diet Coke
400017	Premix (4.75 gallons)	Mr. Pibb
400018	Premix (4.75 gallons)	Sprite
400019	Premix (4.75 gallons)	Pibb Xtra
400025	Premix (4.75 gallons)	Mello Yello
400083	Premix (4.75 gallons)	Schweppe's Sour
400098	Premix (4.75 gallons)	Fanta Tonic
400450	Premix (4.75 gallons)	Barq's
400460	Premix (4.75 gallons)	Dr. Pepper
400492	Premix (4.75 gallons)	Schweppe's Club Soda
510080	Postmix 2.5 gallon BIB	Blue Raspberry
510083	Postmix 2.5 gallon BIB	Fanta Sour
510098	Postmix 2.5 gallon BIB	Fanta Tonic
510119	Postmix 2.5 gallon BIB	POWERade Mountain Blast
510124	Postmix 2.5 gallon BIB	POWERade Fruit Punch
510150	Postmix 2.5 gallon BIB	Nestea Raspberry\Peach
510200	Postmix 2.5 gallon BIB	HI-C Fruit Punch
510203	Postmix 2.5 gallon BIB	HI-C Orange
511001	FCB 2.5 gallon	Coca-Cola
511010	FCB 2.5 gallon	Cherry Coke

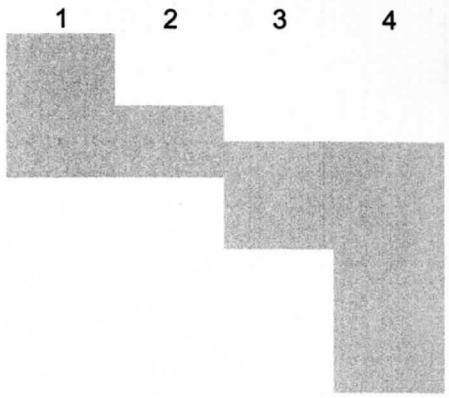
511025	FCB 2.5 gallon	Mello Yello
511052	Postmix 2.5 gallon BIB	MM Cherry
511119	FCB 2.5 gallon	POWERade Mountain Blast
540001	Postmix 5 gallon BIB	Coca-Cola
540002	Postmix 5 gallon BIB	Diet Coke
540004	Postmix 5 gallon BIB	Vanilla Coke
540005	Postmix 5 gallon BIB	CF Diet Coke
540010	Postmix 5 gallon BIB	Cherry Coke
540018	Postmix 5 gallon BIB	Sprite
540019	Postmix 5 gallon BIB	Pibb Xtra
540025	Postmix 5 gallon BIB	Mello Yello
540072	Postmix 5 gallon BIB	MM Lemonade
540083	Postmix 5 gallon BIB	Sour
540089	Postmix 5 gallon BIB	Fanta Orange
540098	Postmix 5 gallon BIB	Fanta Tonic
540152	Postmix 5 gallon BIB	Nestea
540154	Postmix 5 gallon BIB	Sweet Nestea
540218	Postmix 5 gallon BIB	HI-C Pink Lemonade
540450	Postmix 5 gallon BIB	Barq's
540460	Postmix 5 gallon BIB	Dr. Pepper
540461	Postmix 5 gallon BIB	Diet Dr. Pepper
550550	5 gallon	Kandiyohi
600119	Powdered Packs	POWERade Mountain Blast
600121	Powdered Packs	POWERade Lemon Lime
600124	Powdered Packs	POWERade Fruit Punch
601119	Powder (canister)	POWERade Mountain Blast
601121	Powder (canister)	POWERade Lemon Lime
601124	Powder (canister)	POWERade Fruit Punch

A-3 CCBCSD Sales Center Warehouse Inventory Database Project Gannt Chart

Work schedule

Kris

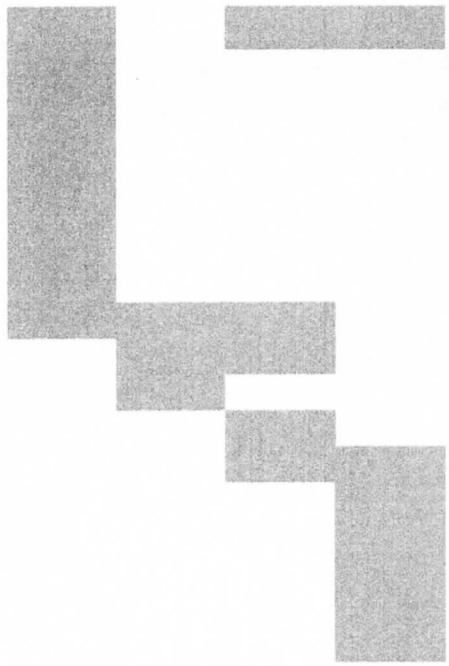
Week



Purchase MS Access textbook
Register with MS Access Developer Forum
Study MS Access textbook
Review Forum Posts and ask questions
Practice creating tables and relationships in MS ACCESS
Practice creating forms in MS ACCESS
Practice creating queries in MS ACCESS
Practice with SQL statements in MS ACCESS
Practice with function formulas in MS ACCESS
Practice with expression builder in MS ACCESS

Work schedule

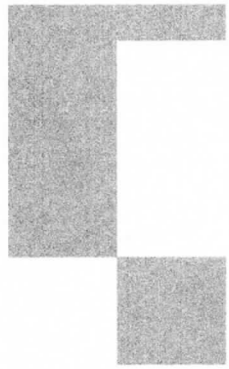
For October 2002



Meet with CCBCSD Manager
Obtain Current Product list from CCBCSD
Begin Building Database version 1.0
Design Product table
Populate Product table
Design Stock Take table
Design Inventory Transactions table
Create Database Relationships
Create Working Forms
Create Working Delete Queries
Obtain Actual Inventory and Sales Reports
Test and Finalize Forms, Delete Queries and Tables
Design and Create Database Specific Reports
Create Working Switchboard/Menu
Obtain Graphics for Forms and Reports
Purchase and Retain Coca-Cola/Chesterman Logo
Create Database version 1 Manual
Package Database version 1 for Distribution

Work schedule

for November 2002



Meet with CCBCSD Manager
Finalize version 1 Database Report Formats
Training session with Mitchell office employees
Training session with Pierre office employee
Training session with Watertown office employee
Distribute version 1 Database to three CCBCSD offices
Distribute version 1 Manual to three CCBCSD offices
Review meeting with Mitchell employees
Telephone interview with Watertown manager
Begin Design on Database version 2

Obtain NORAND Product List from CCBCSD
Design Tables
Populate New Product Table
Design Relationships
Design Forms
Design Reports
Create Database version 2 Manual
Package Database version 2 for Distribution
Meet with Mitchell, Pierre Watertown employees
Training session with CCBCSD employees
Distribute Database and Manual Materials
Interview CCBCSD Manager regarding version 2 Database
Create Project Report Draft 1
Distribute Project Report Draft 1

Work Schedule
For December 2002

Revise and Distribute Project Paper Draft 2
Plan Future Expansions for Project
Determine Web site software solutions

Work Schedule
For January 2003

Design Data Access Pages
Install and configure IIS 5.0 server
Configure IIS 5.0 for three tier RDS
Design Public Data Access Pages
Design Public HTML Pages
Implement ASP password security page
Install and configure Snitz 2000 forums

A-4 CCBCSD Sales Center Warehouse Inventory Database version 1 Manual

Warehouse Inventory Database for Mitchell, Watertown, Pierre SD

Software: Microsoft Access 2000, Microsoft Office 2000 (all rights reserved).

Purpose: This database is to be used on location at individual warehouses to maintain and track "Book" Sales and Transfers into and out of each particular warehouse. Book Sales and Transfers data can then be compared with "Actual" Stock Take data from the warehouse. The Warehouse Inventory Database is to be used by warehouse managers for on-site inventory tracking.

The Database:

Screen 1: Main Menu

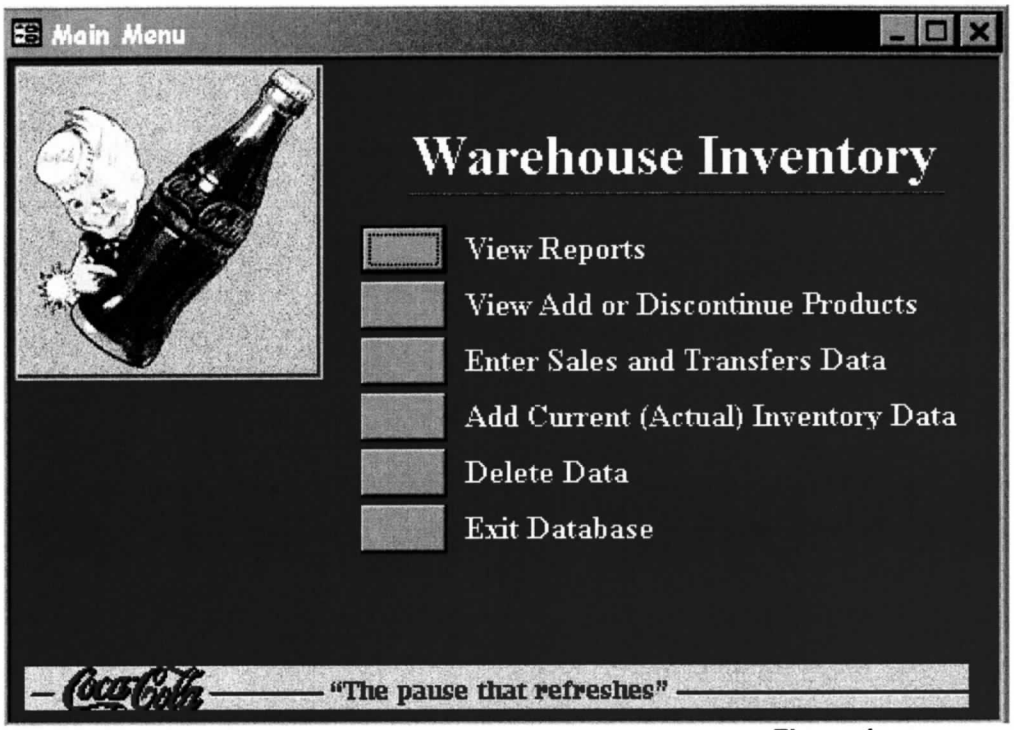


Figure 1.

Upon opening the Warehouse Inventory Database you will be presented with the Main Menu Screen (figure 1). The Main Menu provides the user with captioned command buttons (small rectangular boxes) which will guide the user to various database objects and controls.

Main Menu Command Button Breakdown:

1. Button 1 "View Reports". Clicking this button will take the user to the Inventory and Product Reports Menu. All database reports can be created, previewed and printed by entering this menu. (Figure 2)

2. Button 2 "View Add or Discontinue Products". Clicking this button will open the View, Add or Discontinue Products Menu. (Screen 3) Users can discontinue or add new products in the database by entering this menu.

3. Button 3 "Enter Sales and Transfers Data". Clicking this button will open the Book Inventory Input Form (Sales and Transfers) (Figure 5). Users can enter Sales and Transfers data by entering this form.

4. Button 4 "Add Current (Actual) Inventory Data" Clicking this button will open the Actual Inventory Stock Take Input Form (figure 6). Users can enter Actual Stock Take Inventory Data by entering this form.

5. Button 5 "Delete Data" Clicking this button will open the Delete Data Menu (figure 7). Users can Delete either Actual or Book inventory data by entering this menu.

6. Button 6 "Exit Database" Clicking this button will save and close the Warehouse Inventory Database. Microsoft Access 2000 may then be closed.

Screen 2: View Reports

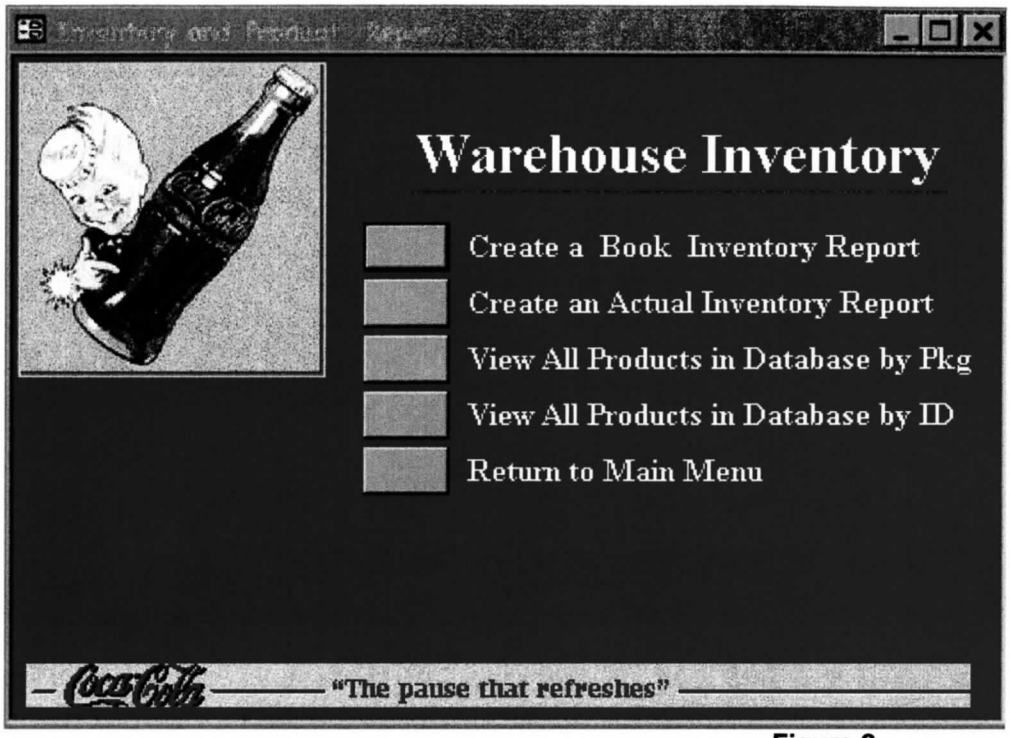
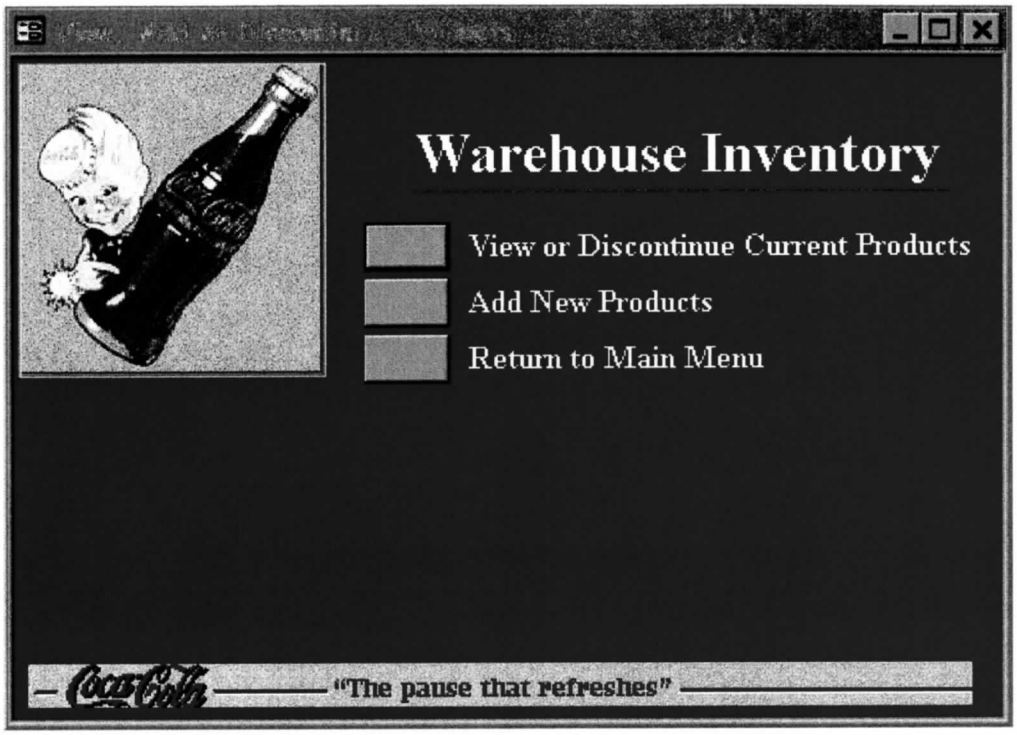


Figure 2.

This menu contains five choices and respective command buttons.

1. Button 1 "Create a Book Inventory Report". Clicking this button will open two consecutive prompt windows (Figure 9) Entering beginning and ending dates specifies the time period for data that will populate the Book Inventory Report.
2. Button 2 "Create an Actual Inventory Report". Clicking this button will open a prompt window (Figure 9) Entering the Stock Take Date in the prompt window will specify what Stock Take data will populate the Actual Inventory Report. **NOTE** Actual Inventory Reports are created using only one specific date.
3. Button 3 "View All Products in Database by Pkg". Clicking this button will open a report preview which lists all products in package specific categories. This report can be used to provide quick reference to the user as to what products exist in the database, and what products in the database are marked as discontinued.
4. Button 4 "View All Products in Database by ID". Clicking this button will open a report preview, which lists all products in the database by product ID. This report will be most useful to new database users, by providing a quick reference to where specific products are located in the database.
5. Button 5 "Return to Main Menu". Clicking this button will return the user to the Main Menu.

Screen 3: View, Add or Discontinue Products



This menu presents three command buttons.

1. Button 1 "View or Discontinue Current Products". Clicking this button will open the View, Add or Discontinue Products form (figure 4) in edit mode. The user may use the record selector buttons (figure 3) to locate an individual product and then click inside the "Discontinued" check box to mark the product as discontinued and remove that product from the list of current products in all other database forms.

2. Button 2 "Add New Products". Clicking this button will open the View, Add or Discontinue Products form (figure 4) in data entry mode. The user may enter a Package name in the Package field and a Product-Flavor name in the Product-Flavor field. ***NOTE*** Only while this original screen is open may the Package or Product Flavor names be edited. Once this window is closed the product is committed to the database. If an error is made in spelling, or for any other reason the Product Package and Flavor names are incorrect the user must mark the product as Discontinued and create a new Product entry.

3. Button 3 "Return to Main Menu". Clicking this button will return the user to the Main Menu Screen (figure 1).

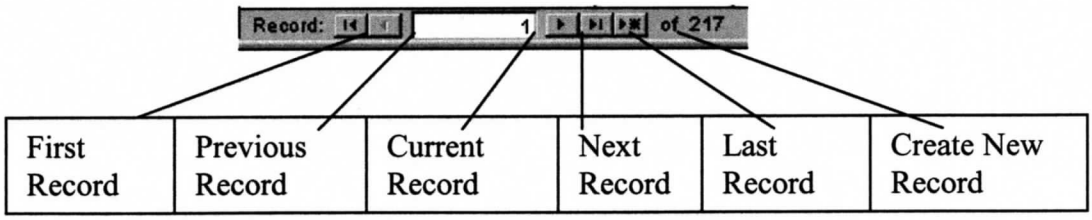


Figure 3.

Form 1: View Current Products/Add New Products Form

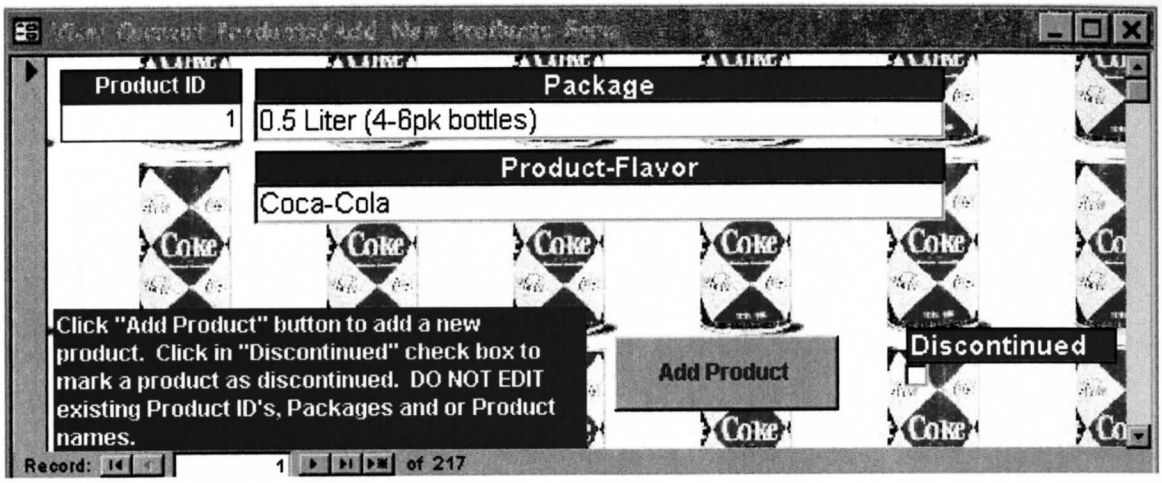


Figure 4.

Screen 4: Book Inventory Input Form (Sales and Transfers)

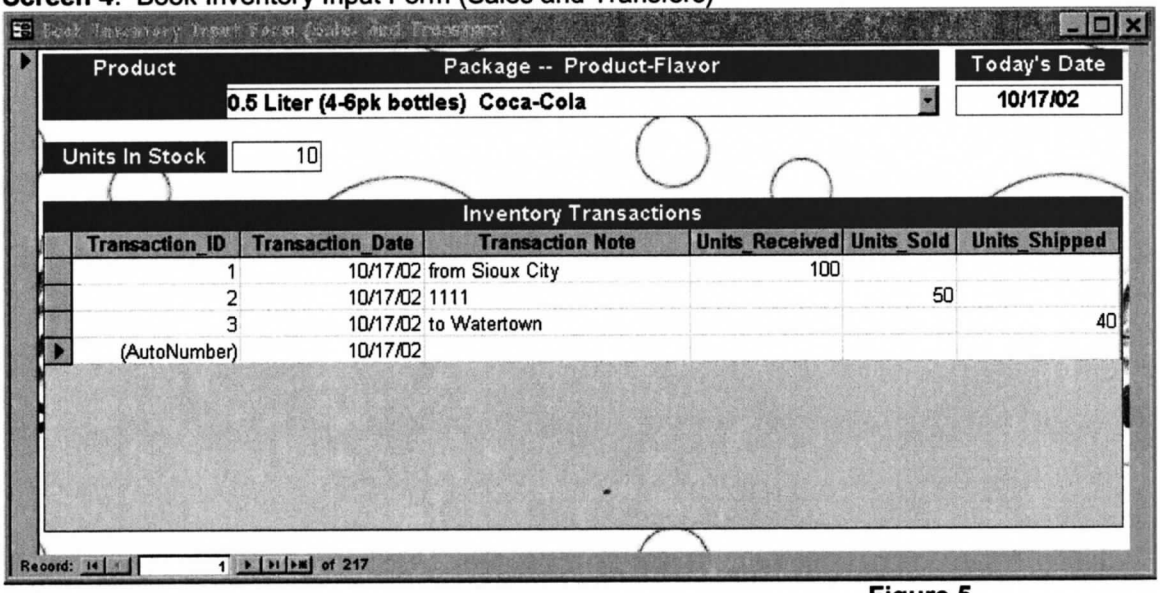


Figure 5.

The Book Inventory Input Form allows the user to enter and or edit Transfers In, Sales and Transfers Out data for any product in the database that has not been discontinued. The Product (Package – Product-Flavor) Field shows the current product for which data is being displayed. By using the Record Selectors at the bottom of the form the user may cycle to the desired product and then begin data input. See Figure 3 for a review of record selector navigation instructions. The Today's Date field will show the current date in short date format. The "Units in Stock" field will calculate and show current units in stock for the current product being displayed.

NOTE "Units in Stock" calculates a value from all transaction data shown in the Inventory Transactions portion (sub form) of the Book Inventory Input Form. Therefore, when or if data

from previous months are left in the transactions table "Units in Stock" values for products will reflect these data in the calculation. (See Delete Data Section for instructions on removing previous month's data values from Book Inventory Transaction tables.)

Below the "Inventory Transactions" heading users may input data. The Transaction_ID field is an autonumber field and requires nor will it allow any editing. The Transaction_Date field is where the user should begin inputting data. The current date will appear in this field. If inputted data relates to a different date correct the date as required using the same short date format. The Transaction_Note field will hold text or numbers "as text". The maximum length for entry to this field is 50 characters. Users may want to include Route Driver Numbers, Truck or Shipment Numbers etc. in the Transaction_Note field. The Units_Received field requires numerical values. Units_Received values pertain to Transfers In or Received shipments to the warehouse. The Units_Sold field requires numerical values (positive for sold units, negative for picked-up units). The Units_Shipped field values pertain to Transfers Out or Shipments out of the warehouse. **NOTE** Data in the "Inventory Transactions Datasheet Table" is editable or changeable for all fields except Transaction ID, which should always be ignored.

Screen 5: Add Current (Actual) Inventory Data

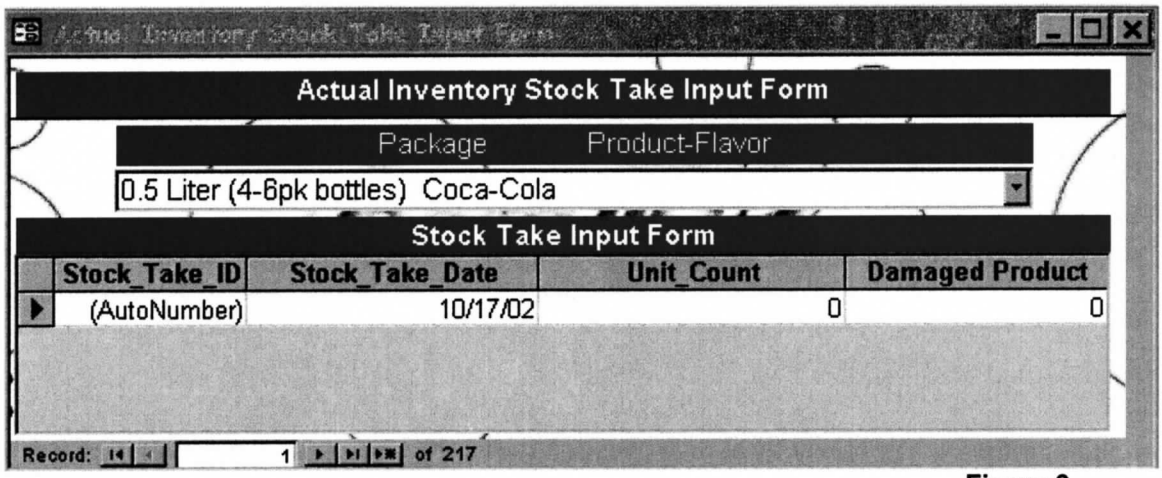


Figure 6.

The Actual Inventory Stock Take Input Form allows the user to input Stock Take or Actual Inventory data. Actual Inventory or Stock Take data comes from an actual physical count of product units in the warehouse. This form is similar to the Book Inventory Input Form with a Product field showing the current Product and a datasheet table for user data input below the Stock Take Input Form heading. The first field Stock Take ID is another autonumber field and should never be edited. The user should begin data input in the Stock Take Date field. This field contains the current date in short date format. If the Stock Take Date is different from this current shown date change as required. The Unit Count Field should hold data for actual units on hand during the Stock Take. The Damaged Product field holds data for actual units damaged at the time of the Stock Take.

NOTE: Warehouse managers will want a Beginning Inventory Stock Take Report i.e. 10/1/02 and an Ending Inventory Stock Take Report ie. 10/31/02. Therefore, all Stock Take data for a particular Stock Take must correspond to one single date. Stock Take inventory data is also editable as in the Book Inventory Input Form. For details on removing old or unneeded Stock Take data see the Delete Data section.

Screen 6: Delete Data

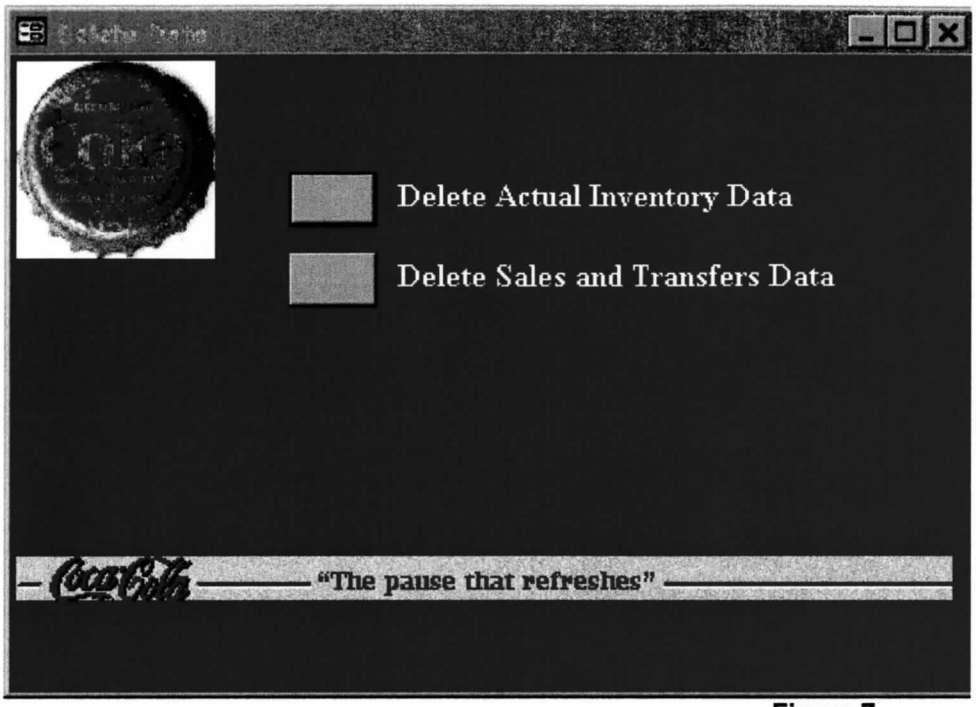
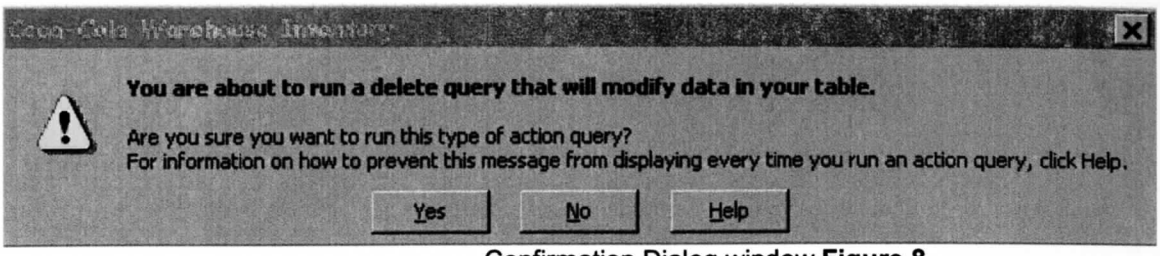


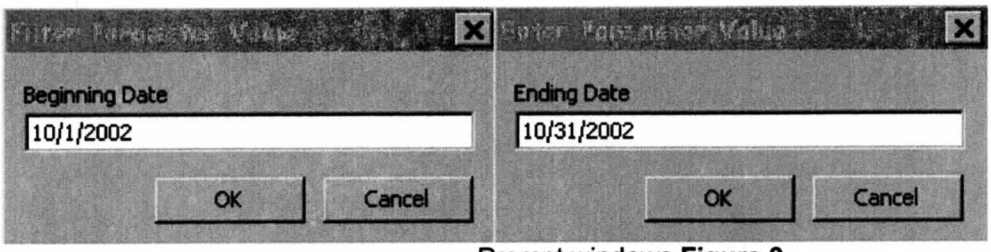
Figure 7.

The Delete Data menu contains two command buttons.

1. Button 1 "Delete Actual Inventory Data". Clicking this button will open a confirmation dialog window (figure 8) confirming whether the user wants to delete data. Upon selecting "YES" from this dialog window another prompt window will open (figure 9). . This prompt window will ask for a date in short date format for "Delete from Beginning Date" i.e. 10/1/2002 Upon clicking "YES" on this prompt window another prompt window will open asking "Delete from Ending Date" ie. 10/31/2002. By filling in these two dates the user is creating a time period or range from which Actual Stock Take Inventory data will be deleted or permanently removed from the Warehouse Inventory Database. Entering and invalid date in either of these date prompt windows will result in no data being selected for removal or deletion. Upon completion of these tasks a final confirmation dialog window will appear which lists the number of rows of data that are being considered for deletion. Clicking "YES" on this window will delete and permanently remove that data, clicking "NO" will abort the delete process.
2. Button 2 "Delete Sales and Transfers Data". Clicking this button will open up the confirmation dialog window once again (figure 8). Upon Selecting "YES" from this dialog window another prompt window will open. This prompt window (figure 9) will ask for a date in short date format for "Delete from Beginning Date" i.e. 10/1/2002 upon clicking "YES" on this prompt window another prompt window will open asking "Delete from Ending Date" ie 10/31/2002. By filling in these two dates the user is creating a time period or range from which Book Inventory data will be deleted or permanently removed from the Warehouse Inventory Database. Entering and invalid date in either of these date prompt windows will result in no data being selected for removal or deletion. Upon completion of these tasks, a final confirmation dialog window will appear which lists the number of rows of data that are being considered for deletion. Clicking "YES" on this window will delete and permanently remove that data, clicking "NO" will abort the delete process.



Confirmation Dialog window **Figure 8.**



Prompt windows **Figure 9.**

Upon completing a Delete query, deleting any database data, it is a good practice to compact the database. This reduces the size of the database and reduces memory usage. To compact the database use the Access menu bar and choose "Tools". In the "Tools" menu choose "Database Utilities" and then choose "Compact and Repair Database...". This process should take between five and thirty seconds and then any new activities can proceed.

A-5 Actual Inventory Report by Time Period and Product

Actual Inventory by Date

Stock Take Date	10/26/2002
Product ID	1
Package	0.5 Liter (4-6pk bottles)
Product-Flavor	Coca-Cola
Unit_Count	100
Stock Take Date	11/26/2002
Product ID	2
Package	0.5 Liter (4-6pk bottles)
Product-Flavor	Diet Coke
Unit_Count	15
Stock Take Date	10/26/2002
Product ID	3
Package	0.5 Liter (4-6pk bottles)
Product-Flavor	Sprite

Unit_Count 100

A-6 CCBCSD Sales Center Warehouse Inventory Database version 2 Manual

Warehouse Inventory Database for Mitchell, Watertown, Pierre SD

Software: Microsoft Access 2000, Microsoft Office 2000 (all rights reserved).

Purpose: This database is to be used on location at individual warehouses to maintain and track "Book" Sales and Transfers into and out of each particular warehouse. Book Sales and Transfers data can then be compared with "Actual" Stock Take data from the warehouse. The Warehouse Inventory Database is to be used by warehouse managers for on-site inventory tracking.

The Database:

Screen 1: Main Menu

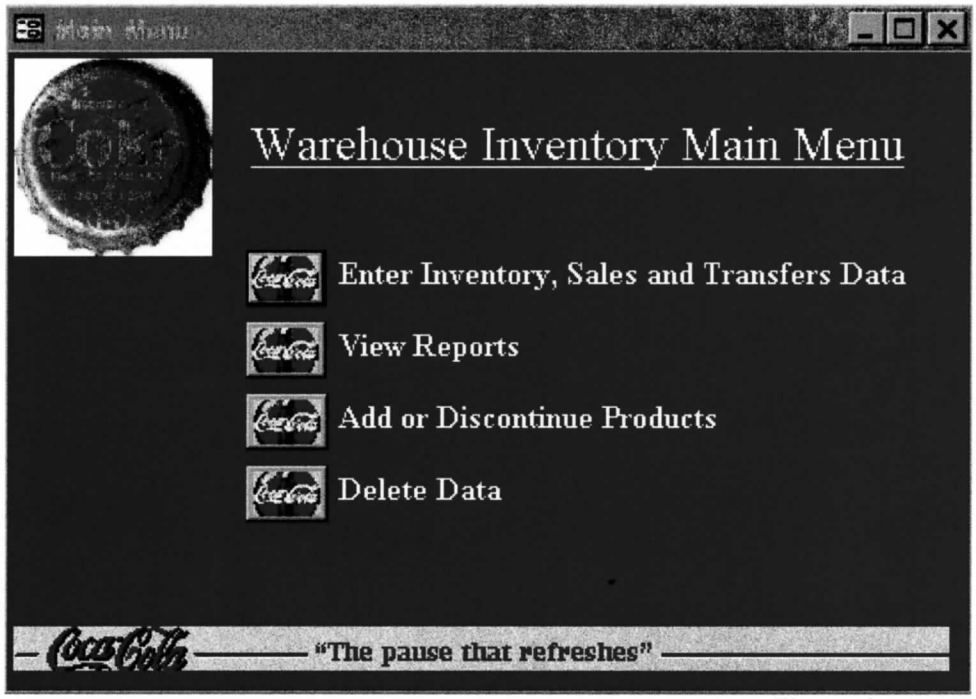


Figure 1.

Upon opening the Warehouse Inventory Database you will be presented with the Main Menu Screen (figure 1). The Main Menu provides the user with captioned command buttons (small rectangular boxes), which will guide the user to various database objects and controls.

Main Menu Command Button Breakdown:

1. **Button 1** "Enter Inventory, Sales and Transfers Data". Clicking this button will take the user to the Book Inventory Input Form. Users may enter all Sales, Transfers, Beginning Inventory and Damaged Product data into the database by entering this form. (Form 2; Figure 5)
2. **Button 2** "View Reports". Clicking this button will open the Create Reports menu. (Screen 2; Figure 2)

3. **Button 3** "Add or Discontinue Products". Clicking this button will open the "View Current Products/Add New Products Form" (Form 1; Figure 4) in edit mode. Users can discontinue or add new products to the database by entering this form. **Do Not Edit (type inside, overwrite) existing NORAND numbers or Package and Product-Flavor's.** If any of these fields are accidentally edited for an existing product the user will need to replace the edited values with correct values. You may do this by referring to a previously printed Product Report. The user may use the record selector buttons (figure 3) to locate an individual product and then click inside the "Discontinued" check box to mark the product as discontinued and remove that product from the list of current products in all other database forms. The form also includes an "Add Product" button. Clicking this button will reset the form to blank values for "NORAND", "Package" and "Product-Flavor". The new product's NORAND number, Package type and Product-Flavor type may then be entered into the form creating a new product listing in the database.
4. **Button 4** "Delete Data". Clicking this button will open the Delete Data Menu (figure 6). Users can delete Book inventory data by entering this menu.

Screen 2: View Reports

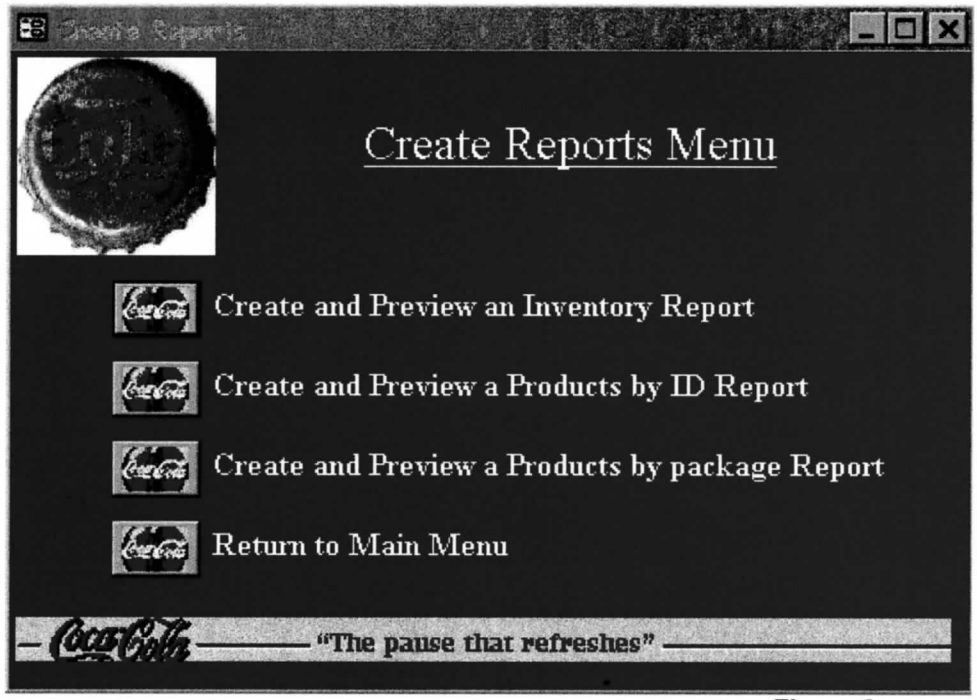


Figure 2.

The Create Reports menu contains four choices and the respective command buttons.

1. **Button 1** "Create and Preview an Inventory Report". Clicking this button will open two consecutive prompt windows (Figure 8) Entering beginning and ending dates specifies the time period for data that will populate the Book Inventory Report.
2. **Button 2** "Create and Preview a Products by ID Report". Clicking this button will open a report preview, which lists all products in the database by product ID. This report will be most useful to new database users, by providing a quick reference to where specific products are located in the database.

3. **Button 3** "Create and Preview a Products by package Report". Clicking this button will open a report preview, which lists all products in package specific categories. This report can be used to provide quick reference to the user as to what products exist in the database, and what products in the database are marked as discontinued.

4. **Button 4** "Return to Main Menu". Clicking this button will return focus to the Main Menu.

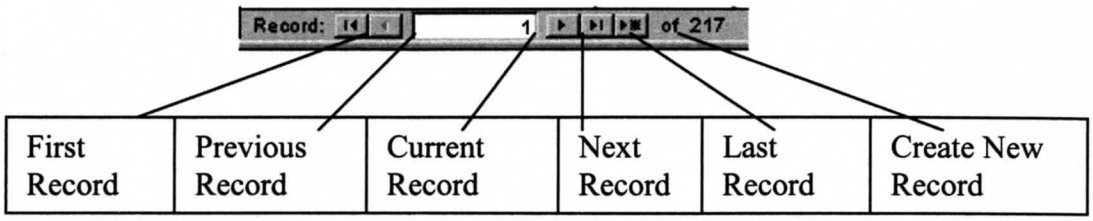


Figure 3.

Form 1: View Current Products/Add New Products Form

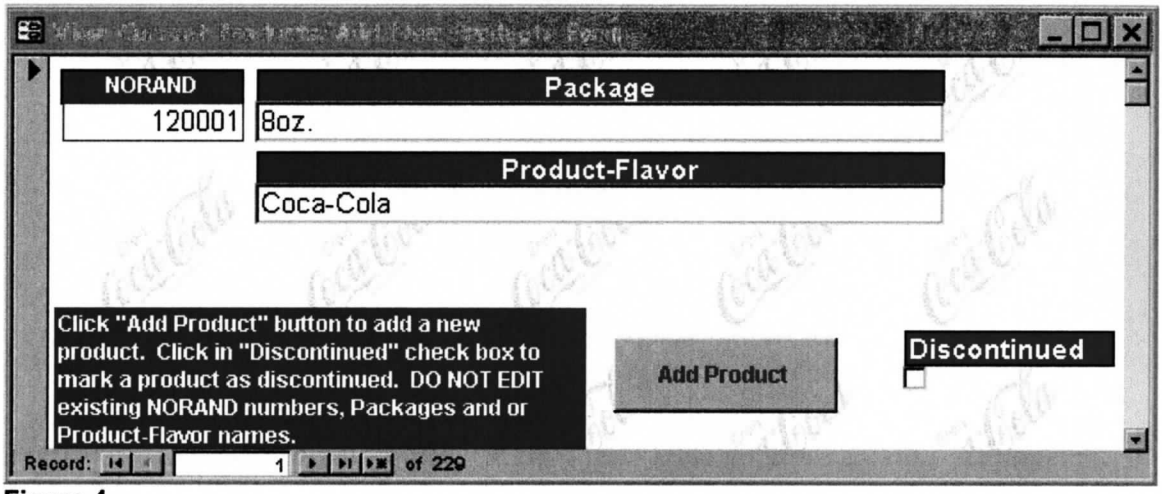


Figure 4.

T

Form 2: Book Inventory Input Form (Sales and Transfers)

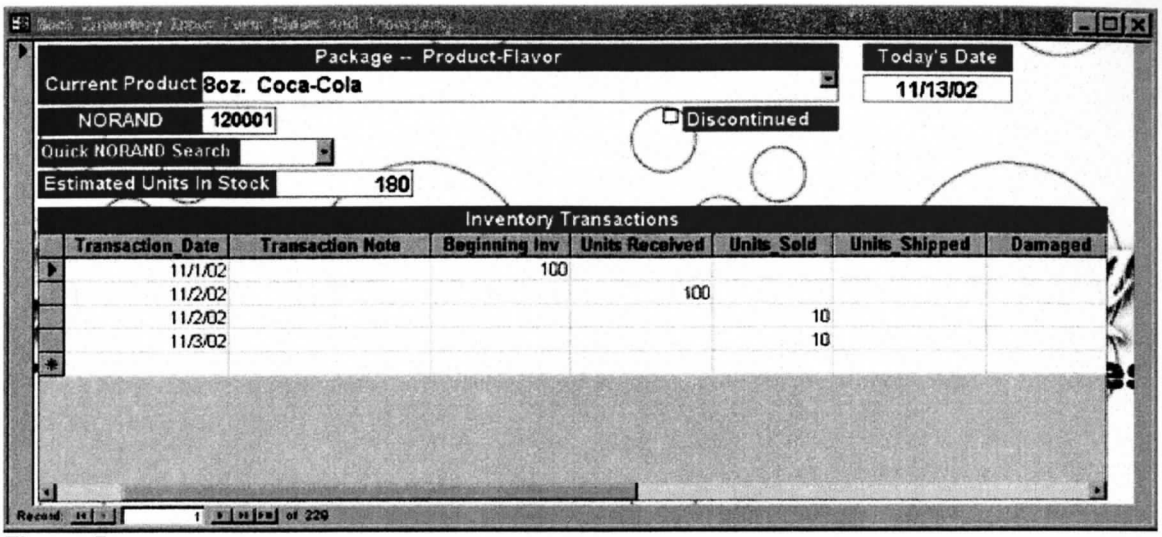


Figure 5.

The Book Inventory Input Form allows the user to enter and or edit Beginning Inventory (Beginning Inv), Transfers In(Units Received), Sales(Units_Sold), Transfers Out(Units_Shipped), and Damaged Product(Damaged) data for any product in the database that has not been discontinued. The Product (Package – Product-Flavor) Field shows the current product for which data is being displayed. By using the Record Selectors at the bottom of the form the user may cycle to the desired product and then begin data input. See Figure 3 for a review of record selector navigation instructions. If the user wants to find a specific product by NORAND code they may use the "Quick NORAND Search" combo box. Selecting a specific NORAND code from the quick search list will bring up that product's data input form. The Today's Date field will show the current date in short date format. The "Estimated Units in Stock" field will calculate and show current units in stock for the current product being displayed.

NOTE "Estimated Units in Stock" calculates a value from all transaction data shown in the Inventory Transactions portion (sub form) of the Book Inventory Input Form. Therefore, when or if data from previous months are left in the transactions table "Estimated Units in Stock" values for products will reflect these data in the calculation. (See Delete Data Section for instructions on removing previous month's data values from Book Inventory Transaction tables.)

Below the "Inventory Transactions" heading users may input data. The Transaction_Date field is where the user should begin inputting data. The Transaction_Date is required and should be inputted in the short date format MM/DD/YY. For example (01/01/02) for the date January 1, 2002. The Transaction_Note field will hold text or numbers "as text". The maximum length for entry to this field is 50 characters. Users may want to include Route Driver Numbers, Truck or Shipment Numbers etc. in the Transaction_Note field. The Beginning Inv field requires numerical values. Actual Stock Take numbers from the warehouse should be entered here once per month on the first of the month for each product. The Units_Received field requires numerical values. Units_Received values pertain to Transfers In or Received shipments to the warehouse. The Units_Sold field requires numerical values (positive for sold units, negative for picked-up units). The Units_Shipped field values pertain to Transfers Out or Shipments out of the warehouse.

Screen 3: Delete Data

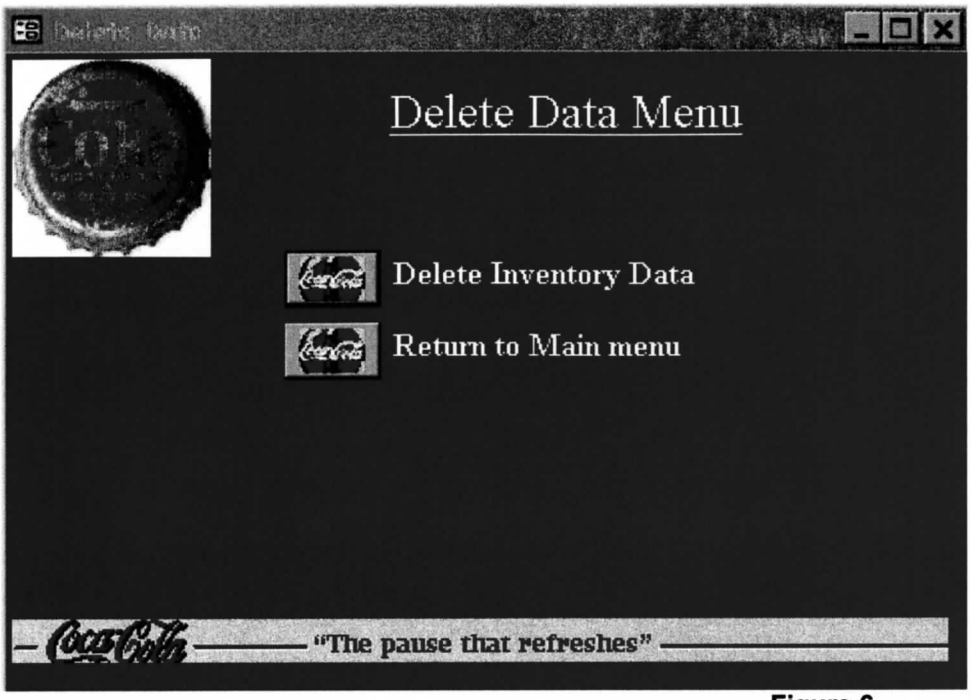
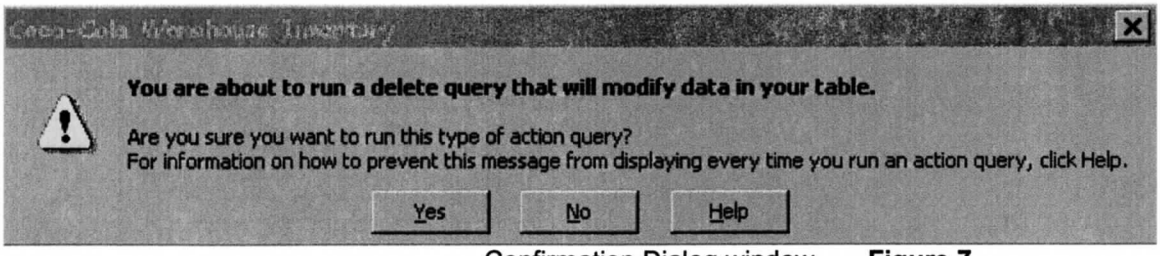


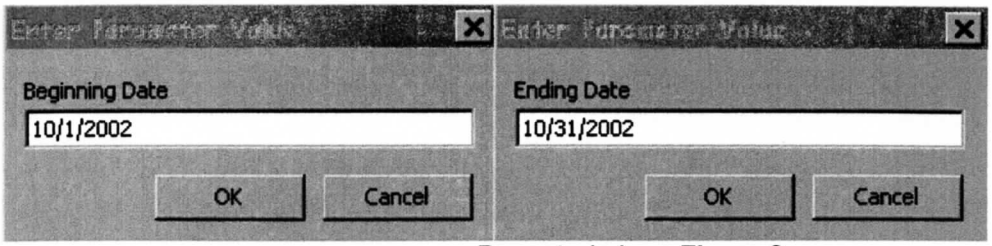
Figure 6.

The Delete Data menu contains two command buttons.

1. **Button 1** "Delete Inventory Data". Clicking this button will open up the confirmation dialog window (figure 7). Upon Selecting "YES" from this dialog window another prompt window will open. This prompt window (figure 8) will ask for a date in short date format for "Delete from Beginning Date" i.e. 10/1/2002 Upon clicking "OK" on this prompt window another prompt window will open asking "Delete from Ending Date" ie. 10/31/2002. By filling in these two dates the user is creating a time period or range from which Book Inventory data will be deleted or permanently removed from the Warehouse Inventory Database. Entering and invalid date in either of these date prompt windows will result in no data being selected for removal or deletion. Upon completion of these tasks, a final confirmation dialog window will appear which lists the number of rows of data that are being considered for deletion. Clicking "YES" on this window will delete and permanently remove that data, clicking "NO" will abort the delete process.



Confirmation Dialog window **Figure 7.**



Prompt windows **Figure 8.**

Important Note: Delete queries are the housekeeping tasks for this database. Performing monthly Delete queries will remove old data, ensure that the "Estimated Units in Stock" (see Form 2; Figure 5) values are correct for the month per product and will speed up overall computing tasks when using this database. Upon completing a Delete query, it is a good practice to compact the database. This reduces the size of the database and reduces memory usage. To compact the database use the Access menu bar (Top of MS Access window, File Edit View etc.) and choose "Tools". In the "Tools" menu choose "Database Utilities" and then choose "Compact and Repair Database…". This process should take between five and thirty seconds and then any new activities can proceed.

A-7 CCBCSD Sales Center Warehouse Inventory Database (Form and Delete Query Related SQL Statements) versions 1 and 2

Sales Center Warehouse Inventory Database version 1.0

Delete Data Query SQL Statement "Delete Inventory Transactions (By Date):

```
DELETE tblInventory_Transactions.*, tblInventory_Transactions.Transaction_Date
FROM tblInventory_Transactions
WHERE (((tblInventory_Transactions.Transaction_Date)>=[Delete from Beginning Date] And
(tblInventory_Transactions.Transaction_Date)<=[Delete from Ending Date]));
```

Delete Data Query SQL Statement "Delete Actual Stock Take Inventory Records (By Date):

```
DELETE tblStock_Take.Stock_Take_ID, tblStock_Take.Stock_Take_Date,
tblStock_Take.Product_ID, tblStock_Take.Unit_Count
FROM tblStock_Take
WHERE (((tblStock_Take.Stock_Take_Date)>=[Enter beginning Date For Deleting] And
(tblStock_Take.Stock_Take_Date)<=[Enter ending Date for Deleting]));
```

Sales Center Warehouse Inventory Database version 2.0

Form_frm Products SQL Statements for objects of consequence

Current Product combo box: SELECT tblProducts.Product_ID, ([Package_Name] & " " & [Flavor_Name]) AS Expr1
FROM tblProducts
WHERE (((tblProducts.Discontinued)=False))
ORDER BY ([Package_Name] & "" & [Flavor_Name]);

Quick NORAND Search combo box: SELECT tblProducts.Product_ID
FROM tblProducts;

Delete Data Query SQL Statement:

DELETE tblInventory_Transactions.*, tblInventory_Transactions.Transaction_Date
FROM tblInventory_Transactions
WHERE (((tblInventory_Transactions.Transaction_Date)>=[Delete from Beginning Date] And (tblInventory_Transactions.Transaction_Date)<=[Delete from Ending Date]));

A-8 Readme.doc included with version 1 distributed compact disc

---

Filename: Warehouse Inventory.exe
Created by: Kristian C.D. Palmer
Date: October 28, 2002

Included Files: **Warehouse Inventory.mdb** – MS Access 2000 Database file
Invicon.ico – Graphic file
Manual for Warehouse Inventory Database.doc – MS Word Document file

---

Required Software: To open Warehouse Inventory.exe you will need at least an evaluation version of **WinZip**. A free evaluation version of **WinZip** may be obtained from www.winzip.com. MS Access version 2000 or higher is required for running the Warehouse Inventory.mdb Database.

What to do with Warehouse Inventory.exe: Open (double click) this file. Choose "Unzip" from the right hand buttons on the Prompt dialog box. The three included files will be extracted to the directory **C:\Warehouse Inventory**.

Purpose: **Warehouse Inventory.mdb** is the MS Access 2000 Database file, which may be used to track and report on Individual Warehouse Inventories. Please refer to the **Manual for Warehouse Inventory Database.doc** to familiarize yourself with this database.

A-9 Itemized Costs

$25.00	1 computer graphic image from Maki Designs
$10.66	3 Clear view Pocket Folders
$21.49	1 Fellowes Neato Brand CD/DVD Labeling System
$36.65	1 HP color ink cartridge

$125.00 5 tanks gasoline for traveling
$218.80

A-10 Readme.doc included with version 2 distributed compact disc

---

Filename: Warehouse Inventory.exe
Created by: Kristian C.D. Palmer
Date: November 14, 2002

Included Files: Directory: C:\Warehouse Inventory
4 Files
1. **Warehouse Inventory.mdb** – MS Access 2000 Database file
2. **Invicon.ico** – Graphic file
3. **Warehouse Inventory Database Mitchell, Pierre anWatertown.doc** – Manual for the Database. MS Word Document file
4. **Warehouse Inventory Manual Cover.doc –** Cover page for Manual.

Directory: C:\Warehouse Inventory\pics
13 Graphics files

Required Software: To open Warehouse Inventory.exe you will need at least an evaluation version of **WinZip**. A free evaluation version of **WinZip** may be obtained from www.winzip.com. MS Access version 2000 or higher is required for running the Warehouse Inventory.mdb Database.

* **What to do with Warehouse Inventory.exe: Open (double click) this file. Choose "Unzip" from the right hand buttons on the Prompt dialog box. Choose "Close". The seventeen included files will be extracted to the directory C:\Warehouse Inventory and C:\Warehouse Inventory\pics.**

Purpose: **Warehouse Inventory.mdb** is the MS Access 2000 Database file, which may be used to track and report on Individual Warehouse Inventories. Please refer to the **Manual for Warehouse Inventory Database.doc** to familiarize yourself with this database.

A-11 Graphics files

graphics commissioned and purchased.

cocacolalogo.tiff (commissioned graphic provided by freelance graphics designer, cost $25.00)

Mitchell CD cover.gif (cd booklet image version 1)