KaDo: An Advanced Enterprise Modeling, Database Design, Database Implementation, and Information Retrieval Case for the Accounting Information Systems Class

Guido L. Geerts
University of Delaware

Kinsun Tam
State University of New York at Albany

ABSTRACT: The KaDo case, designed for either an advanced undergraduate AIS class or a graduate AIS class, has four parts: enterprise modeling, relational database design, integration of legacy systems, and information retrieval. The first part exposes students to advanced enterprise modeling concepts such as subtypes and advanced REA concepts such as policy specifications. The second part asks students to translate advanced modeling concepts such as subtypes into relational database structures. The objective of the third part is for students to learn how to use database software to integrate legacy systems. Their task involves two steps: importing data and integrating data. The fourth part involves information retrieval where students map information requirements into queries that can be executed by the computer. The advanced query techniques that students apply include multi-column join definitions, union, and view hierarchies. KaDo is a dynamic case that allows instructors to create a version of the case adapted to their own learning objectives. KaDo’s Web Environment (KWE) further enables them to create and deliver case materials online.

Keywords: database design; dynamic web application; enterprise modeling; information retrieval; integration of legacy systems; REA.

I. INTRODUCTION

While there are numerous database-oriented cases available for the introductory accounting information systems (AIS) class (Geerts and Waddington 2000; McCarthy 2003; Dunn et al. 2005), instructional materials for an advanced AIS class are rare. Therefore, we created the KaDo case with the following objectives in mind:

- **Enterprise Modeling.** Students learn (1) advanced enterprise modeling constructs such as subtypes, recursive relationships, ternary relationships, weak entities, and
type specifications, and (2) advanced REA constructs such as commitments, alliance associations, and policy specifications.

- **Relational Database Design.** Students learn how to map advanced modeling constructs such as subtypes and recursive relationships into relational database structures. Students are also exposed to design concepts such as load and referential integrity.

- **Integration of Legacy Systems (Database Implementation).** Students learn how to use database software to integrate data stored in legacy systems. Students first learn how to use database software such as Microsoft Access to import data from databases, spreadsheets, and text files. Following that, students are asked to integrate the imported data into a coherent description of KaDo’s economic activities.

- **Information Retrieval.** Students learn how to translate information requirements into executable queries using either QBE or SQL. They are being exposed to advanced query techniques such as inner joins, outer joins, multi-column joins, union, and view hierarchies.

Reusing cases provides the following advantages to instructors: fewer errors, shorter preparation time, and an in-depth understanding of the case and its issues. The major disadvantage to reusing cases is that students have access to case solutions, from either other students or the instructor. However, we have designed case materials and strategies for KaDo that make it easier for instructors to reconfigure their assignments across semesters. The dynamic nature of the KaDo case further enables instructors to adapt the case materials to their specific learning objectives. Additionally, we have built a web environment that facilitates the online creation of assignments and their delivery through the Internet.

II. CASE DESCRIPTION

Your assignment consists of four parts. First, you build an enterprise model for KaDo’s acquisition business process. Second, you define a relational database structure for your enterprise model. Third, you design and implement an integrated database—a portion of which will come from several legacy systems—that depicts KaDo’s acquisition and revenue activities. Fourth, you use KaDo’s enterprise system (the integrated database that you created) to generate information to solve a series of problems.

**Part I: Enterprise Modeling—KaDo’s Acquisition Business Process**

Ashley Clark and Mark King decide to open a Kachina Dolls store in downtown Newark, Delaware. They name their store KaDo and buy their Kachina dolls from artists (vendors) all over the nation (most of them in Arizona). They decide they need an Enterprise Information System that records all transactions related to the acquisition of Kachina dolls and they ask you to help them.

Kachina dolls are pieces of art made by artists known as Kachina carvers. A key to success for KaDo is to find the best artists that carve the types of dolls wanted by East Coast customers. Ashley has selected more than 200 artists she would like to do business with, and she wants to record them into the enterprise system. Ashley has never done business with 50 of the 200 artists, i.e., she has not previously ordered anything from them. She wants to know the following for each artist: code, name, phone, fax, email, street, city,

---

1 Figures 1 and 2 are available in the teaching notes (as Figures 6 and 9) since instructors might opt to create assignments that require students to define those figures as outcome.
state, and zip code. Unfortunately, not all artists have an email address, nor do all artists have a fax number.

Hundreds of different doll types ("doll type" and "item type" are used interchangeably) exist. Ashley and Mark have selected 50 doll types they would like to sell in their store, e.g., eagles, bears, unicorns, and wolves. They have information for all 50 types; i.e., all 50 item types need to be recorded in the enterprise system. They record the following information for each item type: code, name, and description.

Artists always specialize in a limited number of doll types (usually not more than ten). Ashley has contacted all artists, and for each she records the artist’s available doll types. Stated differently, she records the different types of dolls each artist makes. She has more than 20 artists that specialize in wolves, but she has no artist yet that specializes in unicorns.

When Ashley decides to buy dolls, she contacts an artist and asks the artist to carve a number of dolls (i.e., she places an order). She records a number, the current date, and an anticipated delivery date for each order. She specifies the doll types and the quantity per doll type that she wants for each order. For example, she can ask an artist to carve two bears and two eagles. There is at least one doll type per order and exactly one artist per order. Ashley can order many times from the same artist, and she can order the same item type from more than one artist. She has not ordered a unicorn yet.

Ashley usually asks an artist to complete (fill) an order within a month (depending on the size of the order). She accepts partial shipments because artists often ship the dolls one by one. There is exactly one order for each purchase (the terms "purchase," "shipment," and "delivery" are used interchangeably). Ashley records a number and the date received for each delivery. At least one doll (the terms "doll" and "item" are used interchangeably) is involved in each purchase. Ashley assigns an individual tag to each doll. Each doll is purchased, and she puts a short description in the enterprise system for each doll. She does not record the artist involved in a purchase because that information can be derived from the purchase-order relationship (stated differently, do not draw a relationship between the artist and purchase entities). Ashley records exactly one item type (doll type) for each item (doll). This morning she received a shipment with two items—items 2001 and 2002 (individual tags). Item 2001 is of type Bear and item 2002 is of type Eagle. She received another item of type Eagle last week (item 1985).

Ashley’s payment policies are fairly simple. At the end of the month, she pays an artist with one check for all dolls delivered during that month. A check (cash disbursement) always pays for a purchase. An artist can receive many payments (cash disbursements) and exactly one artist needs to be recorded for each payment. There is exactly one account for each cash disbursement. Ashley uses the same account for all payments. An account can be opened before actually making any payment from that account. Ashley and Mark would like to record the following information for each cash disbursement (payment): check number, date, and amount. They would like to record a code and a name for each of the accounts (instances of cash).

One of the most difficult tasks for Ashley is to determine how much she pays each artist. The price of a doll is determined by (1) its item type, and (2) the quality of the art. Customers are usually willing to pay more for a bear than for an eagle. Also, a well-carved doll is worth more than a poorly carved doll. To determine the quality of a doll, Ashley has created five quality categories: A, B, C, D, and E, with A being superior quality and E being junk. Ashley records both the code (e.g., A) and the label (e.g., superior) for each quality category. Using these five categories, Ashley has created a list that determines how much she is willing to pay per item type per quality category. For example, she will pay
$50 for a doll of type Wolf with quality A; however, she will pay only $20 dollars for a doll of type Wolf with quality E. She would like to record this information (price) for each possible combination of item type and quality category.

To determine the price of an item (and thus the price she is actually willing to pay), Ashley asks the artist to rate her/his work. For each item, she records the category claimed by the artist. For example, for the Bear (item 2001) that she received this morning, the artist considers the quality (quality category) B. Therefore, Ashley records category B as the claimed category for item 2001. She records that information at the time the Bear is delivered. Therefore, exactly one claimed category is recorded for each item. Actually, the same artist rated item 2002 as being of category B as well.

Next, Ashley and Mark evaluate the item themselves and assign a quality category to the item. It sometimes takes them a week to decide the quality category of an item (in order to assign a category). Only one category can be assigned to an item. The assigned quality category for an item might be different from its claimed quality category. For example, they rated item 2001 as B, but item 2002 as C. If the claimed and the assigned categories are different, Ashley has to argue with the artist. This often results in a compromise, and the agreed-upon price needs to be explicitly recorded. The same quality category can be assigned to more than one item. Although unlikely, it is possible that none of the artists claim to have an item of quality category E, and it is possible that Ashley and Mark do not assign category E to any item.

Ashley and Mark further group Kachina dolls in eight major categories. These Kachina categories are animal, bird, chief, clown, guard, insect and reptile, plant, and woman. They record a code and a description for each Kachina (or item) category. Exactly one Kachina category is recorded for each item type. For example, buffalo and bear Kachinas are both animal Kachinas. There is at least one item type for each Kachina category.

KaDo further records information for the prestigious Kachina doll exhibitions that take place annually. KaDo records a code and a name for each exhibition. Currently, there are three such exhibitions: (1) the National Kachina Doll Exhibition (NKD), (2) the Phoenix Native Art Exhibition (PNA), and (3) the Southwest Native Art Exhibition (SWNA). For each year, KaDo records when (year) and where (location) the exhibition takes place. For example, the 2004 National Kachina Doll Exhibition (NKD) took place in Santa Fe. In 2005, the exhibition took place in Philadelphia. A unique code is assigned to the actual exhibitions that take place; e.g., code 1 was assigned to the 2004 NKD exhibition in Santa Fe. Artists that are invited to one of these exhibitions are considered as renowned artists by KaDo. Each renowned artist participates in at least one exhibition. For example, renowned artist Lori Fuller participated in (was invited to) the 2004 NKD in Santa Fe and Greg Wilber participated in (was invited to) the 2004 NKD in Santa Fe and the 2005 NKD in Philadelphia. Each of these exhibitions presents awards to the three best artists. Each award has some prize money (e.g., for the NKD exhibition, first place receives $25,000, second place receives $10,000, and third place receives $5,000). If a renowned artist wins an award, KaDo records the prize (place) and the amount (e.g., Greg Wilber won second place (2) for the 2005 NKD and received $10,000). None of KaDo’s artists was invited to the 2003 NKD.

KaDo also records mentor relationships between artists. An artist can have at most one mentor. Many artists do not have a mentor. Most artists never mentor other artists. Some artists mentor many other artists.

**Assignment:**

Draw an enterprise model (E-R diagram) for KaDo’s acquisition business process.
Part II: Relational Database Design

Assignment:
Design a relational database structure for KaDo’s acquisition enterprise model. Make sure to include possible null values (bubbles) and referential integrity definitions (subsets and equalities).

Part III: Integration of Legacy Systems

The database that supports KaDo’s acquisition activities has been implemented for you—master.mdb. However, you have to extend the database to support KaDo’s sales (revenue) activities. Make sure you have a copy of the original master.mdb database (the one that was given to you) at all times. This way you can always start over if you make a mistake.

The Enterprise Model and Relational Database Structure

To complete this part, you are given the enterprise model and the relational database structure for KaDo’s revenue business process, as shown in Figures 1 and 2, respectively. Use both of these as your starting point. Here are a few comments about the enterprise model and the relational database structure:

- The shaded entities and relationships (enterprise model) are already implemented as part of the acquisition cycle and are not shown as part of the revenue cycle’s relational database structure (Figure 2).
- Primary keys are underlined.
- The primary key for SOrderLine has three attributes: SOrder, ItemType, and QualityCategory.
- The “referenced by” recursive relationship for customer is implemented as a foreign key in the customer table (REFERENCEDBY).

Integrating the Legacy System

KaDo currently uses what we will refer to as a legacy system for its revenue cycle activities. This system consists of a mixture of Access database files, Excel® spreadsheet files, and text files. Your task is to create an integrated enterprise system (database) starting from KaDo’s legacy system; i.e., transfer the data in the legacy system to the master.mdb database, eliminating any redundancies, inconsistencies, etc. that exist (and there are many). The three parts of KaDo’s legacy system—(a) database system, (b) spreadsheet files, and (c) text files—are discussed in more detail below.

A. Database System—KaDoLegacy.mdb

KaDo’s legacy database system is named KaDoLegacy.mdb and contains three tables: CustomerData, OrderData, and ShipmentData. Your task is to integrate these tables into the master.mdb database. Next, the structures of the three tables and your task are explained in more detail.

I. CustomerData Table

Description (Data Structure)

<table>
<thead>
<tr>
<th>code</th>
<th>name</th>
<th>phone</th>
<th>fax</th>
<th>email</th>
<th>street</th>
<th>city</th>
<th>state</th>
<th>zip</th>
<th>awardtotal</th>
<th>REFERENCEDBY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Customer
Tasks
• Import the CustomerData table into the master.mdb database.
• Rename the CustomerData table as Customer.
• The zip attribute is currently implemented as a Number field (data type) but should be implemented as a Text field (data type). Change the data type of the zip attribute from number to text.

2. OrderData Table
Description (Data Structure)

<table>
<thead>
<tr>
<th>number</th>
<th>date</th>
<th>CUSTOMER</th>
<th>ITEMTYPE</th>
<th>QUALITYCATEGORY</th>
<th>quantity</th>
</tr>
</thead>
</table>
| OrderData

- No primary key is defined for the OrderData table.
- There is a lot of redundancy in the table—e.g., the customer code (ANKEH) for the first order is repeated six times.
- Some of the item type codes used in the legacy system are incorrect. Here is a list of the changes to be made:

<table>
<thead>
<tr>
<th>Current Value</th>
<th>Needs to be</th>
<th>New Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(legacy system)</td>
<td>Changed to</td>
<td></td>
</tr>
<tr>
<td>EOTO</td>
<td>→ EOT</td>
<td>EOT</td>
</tr>
<tr>
<td>MOSA</td>
<td>→ MOS</td>
<td>MOS</td>
</tr>
<tr>
<td>WAKA</td>
<td>→ WAK</td>
<td>WAK</td>
</tr>
</tbody>
</table>

Tasks
• Import the OrderData table into the master.mdb database.
• Write action queries that correct the ItemType codes (EOTO → EOT, MOSA → MOS, and WAKA → WAK).
• Write action queries that transfer the data in the OrderData table into the tables shown in Figure 2 (SOrder and SOrderLine). Make sure you use the same attribute names and table names as those illustrated in Figure 2—to make sure that your queries (Part IV of the project) work.
• Order (sort) the records in the SOrder and SOrderLine tables by order number (ascending).
• Define primary keys for your tables (SOrder and SOrderLine).
• There should be 25 instances (records) in your SOrder table. There should be 80 instances (records) in your SOrderLine table.

3. ShipmentData Table
Description (Data Structure)

<table>
<thead>
<tr>
<th>number</th>
<th>date</th>
<th>daystopay</th>
<th>ITEM</th>
<th>SORDER</th>
<th>CUSTOMER</th>
</tr>
</thead>
</table>
| ShipmentData

- No primary key is defined for the ShipmentData table.
- There is a lot of redundancy in the table—e.g., the customer code (ANKEH) for the first shipment is repeated 11 times.
Tasks
- Import the ShipmentData table into the master.mdb database.
- Write an action query that transfers the data in the ShipmentData table into the Shipment table shown in Figure 2. Make sure to use the same attribute names as the ones shown in Figure 2.
- There should be 36 instances (records) in your Shipment table.
- Define a primary key for the Shipment table (number).

B. Spreadsheet Files
KaDo’s legacy system contains two Excel files: cashreceiptdata.xls and discountdata.xls. Your next task is to integrate the data in both files into the master.mdb database.

1. CashReceiptData.xls
Description (Data Structure) All data related to payments (cash receipts) are currently stored in a worksheet that has the following format (data structure):

<table>
<thead>
<tr>
<th>rano</th>
<th>date</th>
<th>checkno</th>
<th>CUSTOMER</th>
<th>CASH</th>
<th>SHIPMENT</th>
<th>amount</th>
</tr>
</thead>
</table>

CashReceiptData
- The data in the Excel worksheet record attributes of cash receipt (rano, date, checkno), who paid (customer), the account for the payment (cash), the shipment, and the amount per cash receipt per shipment (amount).
- Again, there is some redundancy—e.g., the customer code (ANKEH) for payment 13 is repeated two times.

Tasks
- Convert (import) the Excel file (cashreceiptdata.xls) into an Access table. Select “no primary key,” and name the new table CashReceiptData.
- Write action queries that transfer the data in the CashReceiptData table into the tables shown in Figure 2 (CashReceipt and ShipmentCashReceipt). Make sure to use the same attribute names and table names as the ones illustrated in Figure 2.
- There should be 15 instances (records) in your CashReceipt table. There should be 16 instances (records) in your ShipmentCashReceipt table.
- Define primary keys for your tables.
- Order the records in the ShipmentCashReceipt table by shipment number (ascending).

2. DiscountData.xls
Description (Data Structure) All data related to discounts are currently stored in a worksheet that has the following format (data structure):

<table>
<thead>
<tr>
<th>number (SHIPMENT)</th>
<th>date</th>
<th>CUSTOMER</th>
<th>type</th>
<th>amount</th>
</tr>
</thead>
</table>

DiscountData
- The data in the Excel worksheet record attributes of shipment (number and date), the customer (name), and the discount type and amount.
- Again, there is some redundancy—e.g., the shipment date for shipment 17 is repeated 2 times.

2 Remittance advice number.
Tasks
- Convert (import) the Excel file (discountdata.xls) into an Access table. Select “no primary key,” and name the new table DiscountData.
- Write an action query that transfers the data in the DiscountData table into the Discount table shown in Figure 2. Make sure to use the same attribute names and table name as the ones illustrated in Figure 2.
- Define a primary key for the Discount table (shipment + type).
- There should be 21 instances (records) in your Discount table.

C. Text Files
1. PreferredCustomerData.txt
Description (Data Structure) KaDo keeps a text file that stores data about preferred customers: preferredcustomerdata.txt. The text file has the following format (data structure):

<table>
<thead>
<tr>
<th>code (preferred customer)</th>
<th>date</th>
<th>code (status)</th>
<th>discount</th>
</tr>
</thead>
</table>

PreferredCustomerData

Tasks
- Convert (import) the text file (preferredcustomerdata) into an Access table. Select “no primary key,” and name the new table PreferredCustomerData.
- Write action queries that transfer the data from the PreferredCustomerData table into the PreferredCustomer, StatusHistory, and Status tables shown in Figure 2. Make sure to use the same attribute names as the ones shown in Figure 2.
- There should be seven instances (records) in your PreferredCustomer table, ten instances (records) in your StatusHistory table, and three instances (records) in your Status table.
- Define primary keys for your tables.
- Think about the following issue: What should we do to define Preferred Customer as a subtype of Customer?

Some Final Comments
- The structure of your database (for the revenue business process) should be exactly the same as the relational database structure in Figure 2.
- Update the Relationship View screen in Microsoft Access!

Part IV: Information Retrieval
Your final task is to formulate queries to retrieve information from the integrated KaDo database you constructed in Part III. The KaDo query assignments are individualized and accessible via the Internet at www.kadocase.com/query.html. After entering an assignment code given to you by your instructor, you will see a screen similar to the one in Figure 3.

As can be seen from Figure 3, you will be given a list of queries. Each query will have an ID, a description, and an “execute” button. As is shown for query 401 in Figure 4, the expected results for a query will be displayed when you click on the “execute” button. The results are shown to assist you in getting the correct answer.
TEACHING NOTES

Teaching Notes are available only to full-member subscribers to the Journal of Information Systems through the American Accounting Association’s electronic publications system at http://www.atypon-link.com/action/showPublisherJournals?code=AAA. Full member subscribers should use their personalized usernames and passwords for entry into the system where the Teaching Notes can be reviewed and printed.

If you are a full member of AAA with a subscription to the Journal of Information Systems and have any trouble accessing this material, please contact the AAA headquarters office at office@aaahq.org or (941) 921–7747.
FIGURE 4
Results for Query 401

REFERENCES