CASE STUDY – INTERNET SHOP

DBDudes Inc., a well-known database-consulting firm, has been called in to help Barns and Nobble (B & N) with its database design and implementation. B & N is a large bookstore specializing in books on horseracing, and it has decided to go online. DBDudes first verifies that B & N is willing and able to pay its steep fees and then schedules a lunch meeting --- to do requirement analysis.

Requirement Analysis

The owner of B & N, explains the requirements

“ I would like my customers to be able to browse my catalog of books and place orders over the Internet. Currently, I take orders over the phone. I have mostly corporate customers who call me and give me the ISBN number of books and a quantity; they often pay by credit card. I then prepare a shipment that contains the books they ordered. If I don’t have enough copies in stock, I order additional copies and delay the shipment until the new copies arrive; I want to ship a customer’s entire order together. My catalog includes all the books I sell. For each book, the catalog contains its ISBN number, title, author, purchase price, sales price, and the year the book was published. Most of my customers are regulars, and I have records with their names and addresses.

New customers have to call me first and establish an account before they can use my website.

On my new website, customers should first identify themselves by their unique customer identification number. Then they should be able to browse my catalog and to place orders online.”

CONCEPTUAL DESIGN

DBDudes develops a high level description of the data in terms of the ER model. The initial design is given, where books and customers are modeled as entities and related through orders that customers place.

Orders is a relationship set connecting the Books and Customer entity sets.

.

Books

Customers

orders

For each order, the attributes quantity, order date and ship date are filled. As soon as order is shipped, the ship date is set; until then the ship date is set to null, indicating that this order has not been shipped yet.

DBDudes has an internal design review at this point, and several questions are raised.

The name of the designer leader name is Dude 1 and the design reviewer name is Dude 2.

Dude 2 What if a customer places two orders for the same book in one day?

Dude 1 The first order is handled by creating a new orders relationship and the second order is handled by updating the value of the quantity attributes in this relationship.

Dude 2 What if a customer places two orders for different books in one day?

Dude 1 No problem, each instance of Orders relationship set relates the customer to a different book.

Dude 2 What happens if a customer places two orders for the same book on different days?

Dude 1 We can use the attribute order date of the order relationship to distinguish the two orders.

Dude 2 : No we can’t. The attributes of customers and Books must jointly contain a key for orders. So this design does not allow a customer to place orders for the same book on different days.

Dude 1 : yes, you’re right. Well, B & N probably won’t care; we’ll see.

DBDudes decides to proceed with the next phase, logical database design.

Books

**Isbn**  title author qty\_in\_stock price year published

Customers

**Cid** cname address

orders

**isbn cid**  cardnum qty **order\_date** ship\_date

Isbn reference to books

Cid references to Customers

The Owner of B & N now brings up some additional requirements he did not mention during initial discussions.

## Customer can order several different books in a single order

Again redundancy comes, and it is not possible to maintain a key with the existing structure in orders table.

They added a new column called ordernum and makes

Ordernum and isbn as a key.

Orders

**Ordernum isbn** cid cardnum qty order\_date ship\_date

Schema Refinement

Orders

Orders (**ordernum**, cid, order\_date, cardnum)

Orderlists (**ordernum,isbn**,qty,ship\_date)