

# CS 4400 Introduction to Database Systems

## 2002 Spring Term Project (Section A)

*Last updated 1/15/2002, 9:30pm*

In this term project, you are asked to design a small database system, create and populate this database by using ORACLE, and write a number of application programs to access the database. The topic of the project is to design the information system that **everest.com** can use. In the following, you are given the requirements for the conceptual design and the functional analysis (Section 1), the requirements for the database design (Section 2), and the requirements for application design (Section 3).

### Important Dates

- **2/05** Send your group members and the email of your group contact person to the instructor
- **2/21** Phase I Report Due
- **3/14** Phase II Report Due
- **4/11** Send your choice of Demo Date and Hour to the instructor
- **4/25** Final Report Due

## 1. GENERAL GUIDELINES

The following guidelines apply for the project:

1. This project is a team project. The group size should not exceed 4 members. Each group should do the project independently, though "discussion" among groups is encouraged. All material submitted that represent work copied from elsewhere **MUST** be explicitly acknowledged.
2. The project should be conducted in three consecutive phases:

**Phase 1: Requirement Analysis and Specification** (1/17 -2/21).

The main task of this phase is to perform conceptual modeling using ER model and to conduct functional analysis using Data Flow Diagram approach.

**Phase 2: Database Design** (2/22-3/14).

The main task of this phase is to map the ER model to the relational data model and to normalize the relational schemas into the third normal form.

**Phase 3: Application Interface Implementation and Testing** (3/14-4/25).

One of the main tasks in this phase is to write a set of application programs that implement all the functions/processes specified in your data flow diagram (DFD). In order to test and run your application interface programs (APIs), you will need

to create the relational schemas for the database, design the data sets that you use for the testing and demo, and populate the database with the data sets.

3. The application design and the implementation of your **Everest.com** system must be implemented either on Oracle with a host programming language such as C, Java, or MS SQL Server or MS ACCESS database management system. If you would like to use other DBMS product instead of Oracle, SQL Server or ACCESS, you must discuss your alternative choice with your instructor and obtain the permission in advance. **If you choose to use non-Oracle DBMS for your project, you will be responsible for setting up your system in the lab or your own laptop for project demo at the end of the semester.**
4. The following description is rather generic on purpose. YOU ARE ENCOURAGED TO EXPAND THE DESCRIPTION OR PROPOSE A PROJECT TO REPLACE THIS DEFAULT ONE. This could involve making a real database for someone running a business. However, you have to use ORACLE or Microsoft SQL server as the underlying DBMS. Permission **must** be obtained before doing this. Work done in the past is not admissible unless a new component is added. You may be asked to present your project to the class if your project exhibits interesting design challenges or demonstrates the use of other types of DBMSs. We will do this if time permits.
5. You are required to **demonstrate** your programs to the TA and/or the instructor. The demonstration will be scheduled in two weekdays: April 22 (10:00am - 5:30pm), and April 24 (10:00 - 5:30pm). You should treat these demonstrations as if you were giving them to your customer. So, prepare them professionally.
6. You are required to submit a **typed** project report for grading at the end of each phase. For each of the three reports, you should identify the major design decisions that you faced and the design decisions that you made with justifications for those decisions. The first and second project design reports are due in class at the end of the Phase I (**2/21**) and the end of the Phase II (**3/14**). The final project report is due in class on **4/25**. The first two reports are primarily to help you to make sure that your project design is on the right track. The marked design report will be returned to you no later than a week from the due date. The grade of the projects will be returned to you by email on 4/28. The closing date for appeals for the project is May 4, 2000 at 5 PM.
7. Grading policy is as follows:
  - Phase 1 design report: 30%
  - Phase 2 design report: 30%
  - Phase 3 application programs: 40%

## **2. Phase 1 -- CONCEPTUAL DESIGN REQUIREMENTS**

**Everest.com** is a new bookstore chain that is being set up with the objective of providing the best services to its customers by keeping their services at the leading edge of technology. Thus, they would not only like to provide the traditional bookstore services, but they also keep detailed records on their customers and their preferences in order to customize their services to the customers' needs and preferences. You have been hired to set-up an information system to assist **Everest.com** achieve its goals.

A requirement analysis that was conducted has identified a number of things about the operations and goals of **Everest.com**. You, as the systems analyst/designer, should feel free to add to these requirements in order to achieve a richer design.

The bookstore keeps records about a number of entities. First, it keeps track of the books in the system, which are uniquely identified by their ISBN number. Other essential information includes the title and authors of the book, its publisher, the year of which it was or will be published, the price of the book, and the shipping cost. **Everest.com** is thinking about linking their books with the book reviews conducted by a number of independent organizations such as New York Times.

Another essential piece of the data that **Everest.com** wants to maintain is the data about its customers (so-called customers' profiles). This customer information includes the name, delivery address, telephone number, email, the membership that the customer has (if any), the preferred payment method (e.g., credit cards and their numbers) and a list of customer-specific preferences (hard/soft cover, preferred authors, preferred subjects, preferred shipping days, etc.).

The third entity that **Everest.com** keeps is the Writers' information, including writer's name, highest national/international award if any, years of writer's experience, birth place, and the city in which the writer lives. This entity is important to support customers who wish to search for books by authors' birthplace, the type of national/international recognition, etc.

The fourth and the fifth entities are the reviews' information and the book suppliers' information. A review is described by the title and the authors of the book being reviewed, the writer(s) of the review, and the publisher (such as NY times) of the review. To ensure a timely delivery of book orders from customers, **Everest.com** often needs to make several orders to various book suppliers for a book that is out of order in the stock. Example suppliers are publishers and bookstores. The supplier's name, the supplier's address that orders can be placed and the contact person or telephone (optional).

Several complications are noticed during requirement analysis. First, different types of books may be ordered at once by the same customer at sale price or regular price, and the sale amount differs on different weekdays. Therefore, the number of books in stock for each title changes. Second, the book price seems to change on different days (weekends are cheaper than weekdays, for example) or for different seasons. Therefore, there may be a need to keep track of the discount price of those book "instances" that are on sale or the books whose shipping charges are at special discount, in addition to their routine price

and routine shipping cost. A book may be assigned to several sale-book instances (of course as long as their times do not conflict, but we are not concerned with this scheduling problem; that is the responsibility of the bookstore chain). Last but not least, **Everest.com** classifies its customers space with different status of preferred membership. For instance, the gold level Everest.com customers receive 10% discount for every book he/she buys; and the silver level Everest.com customers receive 5% discount, and so forth.

When a customer makes a purchase order of books, this should, of course, be kept in the company's database. The order record should identify the customer, the order identifier which is assigned when the order is placed successfully, the preferred sales that are part of the same order (e.g., a book by Orfali Harkey with 50% discount or a free shipping), the payment mode for this order, and the status of the order (paid order, paid pending order, unpaid pending order, etc.). Typically, an order is either OK'ed which means it is confirmed and a book is reserved for the customer, or the order is on a waiting list. Note that it is possible for one book in an order to be OK'ed and another book in the same order to be on the waiting list. For each order, the customer's preferences (if any, new or updated) are also recorded.

Similarly, when a book is out of stock, the clerk of **Everest.com** will need to place an order to one or more suppliers to obtain the best bet for books to be ordered from suppliers. For simplicity, this function is not required for implementation in the Phase III of the project.

This is the end of the requirement report. You are now expected to read this report carefully and to come up with your (data and functional) requirement specification, and writes the first report -- the Phase I report. This report must include a description of the purpose of this project, and the purpose of this phase of the project. It must describe the problems encountered in this phase, and justify the solution. It must contain all the documentation produced in this phase, including the development of (a) an entity relationship model for the conceptual design, and (b) a data flow diagram.

### **3. Phase 2 -- DATABASE DESIGN REQUIREMENTS**

In Phase II, you will need to design the relational schemas for the database, and design the data sets that you use to populate the database for the testing and the demo. You are expected to populate this database by the sets of data either downloaded from the Web or created by your self so that you can demonstrate the functionality developed in Phase III to your customer.

The Phase II report must contain a description of the purpose of this project. It must attach with the Phase I report and must describe any revisions made to the specification described in the Phase I report. It must describe the problems encountered in this phase, and justify the solution. It must contain all documentation produced in this phase,

including the development of (a) a relational schema from the entity relationship model produced in Phase I, (b) the set of functional dependencies used to evaluate if the resulting schema is a BCNF or a 3NF; (c) the normalization process (if necessary) that transforms the relational schema to a relational schema of the BCNF or 3NF.

## 4. Phase 3 -- APPLICATION DEVELOPMENT REQUIREMENTS

**Everest.com** is interested in an application program interface that is easy to use so that they do not have to spend unnecessary amount of time to train their employees. To meet this requirement, you are expected to write at least a menu-based main program that consists of the following four application programs for their daily transactions and is open for the inclusion of future programs.

### 4.1 Everest.com Query Program

The Query program is used by at least two types of users: (a) the customers who need to browse and find the books of interest, and the price and shipping quotes before making an order; and (b) the **Everest.com** clerks or managers who are interested in conducting a survey of sales for a given period of time, a given geographical region, or a given customer age range. The query system should at least support the following two types of queries:

1. Conditional Retrieval of Books, Customer Profilers, Writer Profiles, and On-Sale Books -- This program should support queries on single relation with various atomic or logic (AND/OR/NOT) conditions. Let's take the relation **Book** as an example, this program should allow users to browse the collection of books in **Everest.com** system with simple or sophisticated search conditions. Examples include the Boolean conditions on title, authors, publishers, publication year of books, range queries such as "find all the books by James Peterson, which were published between 1997 to 1999?", or point queries such as "find all the books by James Peterson on the subject of on Operating Systems or Networks, which were published by Addison Wesley?".
2. Join Queries -- This program should allow users to perform various join queries. For instance, users should be able to retrieve books using join conditions such as searching books by a particular sale offer, a particular review, or a particular national or international award recognition of authors. Similar scenarios apply to Customers' profiles, Writers' profiles, and On-Sale Books.

The query program, invoked by a one-word command or a click on the main menu, should first issue prompts to allow users to choose the type of queries. Once a query type is chosen, it should issue prompts to let users enter their queries. You are encouraged to design an API that is user-friendly.

Note that you are expected to list all functions that you think are important and choose the subset of the functions in the above categories in your implementation. However, the minimal requirement is to support queries on all relations that you have created and at least join queries over one pair of relations.

## 4.2 Purchase Order Program

This program is used to make the purchase, by which a **Everest.com** agent should be able to insert the following data items into the database: the order number, customer id, name, telephone number, delivery address, email, membership status (gold/silver/beginner), payment mode, a list of book instances and the customer preferences. A book can be put on the delivery list only if the payment method is entered. The status of each order should be calculated automatically according to the policy given below. You design the interface for insertion of these data into the database. The database should be updated accordingly after each purchase order, including the number of available books on a title and the customer's profile.

**The order policy:** **Everest.com** allows customers to order books that are currently not available. However, all books on the waiting list should not be charged until they are ready for delivery.

## 4.3 Customer Profile Update Program

This application will go into the database and given a customer name or an order number, prints a summarization report of the customer or the order. Since a customer may have a number of outstanding orders, if the access to the database is via the customer name, the system should display the set of active orders for that customer. An order report should provide the status of the ordering process and the status and method of the payment.

A receipt can only be printed if the order is paid. An order is closed when the books ordered have been delivered. The ticketing program must also keep track of which receipts are printed. Each receipt should only be printed once.

## 4.4 Activity Summarization Program

The main purpose of daily activity report for all book orders is to monitor the in-stock level of books. The report should be generated at the end of each business day, and should contain the following data items for each order that was made that day: the order number, the book instances, the price for each book and the sale tax (if any). The report content should be grouped in terms of the customer. The total number of purchase orders (by count), the grand total of dollars, and the total dollar amount of save from on-sale books and discounts by membership should also be included in the report for each customer with sale tax separated.

This summarization program may also provide richer summarization queries in addition to the daily activity reporting, such as how many sale transactions per week, what is the

total sale in dollars in the past 3 days? What is the percentage of books sold which were on-sale books? How many customers are female customers or senior customers? You are encouraged to provide more of such summarization capabilities in your query system, though they are not required.

Implement the above four programs and produce sample output. Your application programs should contain a menu from which a user can select one of the four applications. Each application may again use a pop-up menu for a user to further select the service she wants. You may use SQL\*FORMS, SQL\*MENU for creating the user interface if you are experienced with these 4GL tools.

For privacy reason, the activity summarization program can only be used by restricted set of users, such as the **Everest.com** employees who are authorized to use the program.

The final report, due at end of the third phase, should be an integrated report. For the list of relational instances you use to populate your database, you should take data from a realistic situation, and make sure that your data show some variety.

The report for Phase III must include the following five components:

1. a description of the purpose of this phase of the project and the list of tasks, a summary of the system requirements and any additions or any revisions made to the specification and the design in the previous two phases,
2. the description of the list of problems encountered and your solutions.
3. the system architecture, and the design and description of all functions, including the list of required ones and the list of functions that your group add into the Everest.com system.
4. the conclusion that evaluates the system you have implemented, the current limitations and the potential for improvements.
5. each team member will provide a half page to one page of description on your contributions to the group project and what you learned from this project and the team work.
6. The following attachments
  - an attachment of the Phase I report,
  - an attachment of the Phase II report,
  - an attachment of the print-outs of source code, including (1) the relational schema in Oracle SQL DDL, (2) the source code of the four application programs, and (3) the relational instances you use to populate the database, (4) a sample output of the Everest.com query and update programs, including the summarization outputs, and
  - an attachment of a user manual for the system.

A hard copy of all the above components is expected at the demo time or on 4/25 in class. Together with the hard copy of your final project report, you need to turn in the report and the source code in electronic form using a gzipped or winzipped file.

## 5. Grading Policy

The following aspects are important for the grading:

- clarity of description and diagrams,
- completeness and sophistication of the ER model, the database design, and the data flow diagram,
- coverage and quality of application program design and implementation, including correctness of all functions, friendliness of the API interfaces, and the clarity of user-manual.

A general acceptable design report or implementation will get a score of 80. A design report with more complicated database design or application design will get more. An implementation with more complicated database or application or more user-friendly interface will get more. Please be aware that the database functions are more important and will take the larger portion of the grade.

---

*January 15, 2002, updated on January 16, 2002*

*lingliu@cc.gatech.edu*