Cases in Database Design

LUDWIG SLUSKY

Cases in Database Design

LUDWIG SLUSKY

A division of the C.V. Mosby Company
11830 Westline Industrial Drive, St. Louis, MO 63146

Printed in the United States of America

PREFACE

THE PURPOSE OF THIS CASEBOOK

Database design is an applied discipline. An understanding of the concepts and methodologies presented by most database texts requires exposure to practical, business-oriented situations. To meet this need, **Cases in Database Design** presents realistic case studies supported by a sample case study with solutions, practical explanations, forms, helpful notes on database design, and suggested project assignments. Students working on these case studies should need minimal assistance from the instructor.

In compliance with the author's objectives in writing this text, the cases should:

* **be flexible** enough to allow for a variety of approaches, techniques, and assignments over a period of time
* **accommodate different levels** of database design skills,
* **motivate** a reader and maintain his/her interest,
* **maximize productivity** of a reader by incorporating helpful forms, tables of data, and naming conventions,
* **minimize demands** on the instructor's time,
* **help** the instructor to monitor the design process and to control its results.

HOW TO USE THIS TEXT

Because of its pragmatic instructional approach, **Cases in Database Design** can be used easily in conjunction with a core text or as a stand-alone text.

Used with a core text, this book provides graduate or undergraduate students with the opportunity to work with database design fundamentals, either alone or as part of a team, reinforcing and clarifying the concepts presented in the main text.

As a stand-alone text, **Cases in Database Design** can be adopted as practical training case material for the professional in practice-oriented specialized short courses and seminars.

The book was developed to address several concerns that an instructor may have:

1. **The instructor may need to relate a database design to a system design.**
   The book provides for each case study a description of functions and data flows supported by a data flow diagram and a data dictionary.

2. **A project team often does not have enough time to accomplish the wide range of steps involved in database design and implementation.**
   The book provides tables for each case study summarizing intermediate results of data analysis: definitions of data, required outputs, and data relationships. To minimize the time necessary to complete a project, a team may accept these intermediate results (modified or not) and proceed with further database design and implementation steps. The scope of the project can also be narrowed by reducing the number of data elements involved in the database, modifying or reducing the number of required outputs, and simplifying data relationships.

3. **The instructor may need to compare the results of different teams working on the same case study.**
   The book greatly simplifies this comparison if teams use the same naming convention for data elements specified in the data dictionaries, and if essential data relationships are explicitly specified in the table format.

4. **The instructor may need to vary a database design approach to the same case study by using either the entity-driven or output-driven approach.**
   The book supports both approaches. If the entity-driven approach is chosen, the project team may use only a table of data definitions and may disregard the remaining tables of required outputs and data relationships.

5. **The instructor may need to use the same case study over a period of time.**
   The book allows for applying modifications to a scope of data elements, output requirements, and data relationships. For the

output-driven approach, these modifications are likely to result in different database designs.

6. **The instructor may not have time to discuss application environment details and functions with a project team.**
If the output-driven approach is used, and requirements for database outputs and data relationships are specified in the tables, no additional application explanations are necessary to complete a project.

FLEXIBILITY

This book has been organized to help the student develop solutions to case studies. The first two chapters outline a case study, solutions to the case study, and helpful database design notes. Solutions for selected assignments in Chapter 2 can be used as an example for the five case studies that follow in Chapters 4 through 8. Suggested student assignments for the case studies are presented in Chapter 3 in conjunction with guidelines for planning the case study project and evaluating team performance on projects.

The six case studies are realistic, although abbreviated, and encompass a variety of situations. Each has the same basic organization and requirements as the Eagle Corporation sample case study in Chapter 1: introduction, problems of the current information system, database planning phase, conceptualization phase, and user requirements. In addition, each case study includes a data flow diagram, data dictionary of data elements, a summary of user-requested database outputs, and a table of data relationships assumptions. This consistent structure helps the student, and it also makes it easy for the instructor to assign projects and to evaluate student performance.

The case studies are presented in the order of increasing difficulty and complexity, from the least complex DuSchah College Scholarship Case Study in Chapter 4 to the Sergeant Video Case Study in Chapter 8 with a consideration of a distributed database environment.

The reader is encouraged to compare a chosen case study with a similar application in the reader's community. Such a comparison will provide a valuable perspective on database design and implementation.

The cases support two accepted approaches to database design:

* **entity-driven database design** with an open applications scope allowing a reader to define his/her understanding of application and to analyze data entities (if this approach is chosen, the reader does not need to use the tables provided in each case study),

* **output-driven database design** with a specified, rigid applications scope and therefore with more "preprogrammed," more easily verifiable results (using this approach, the reader should specify and use the accepted assumptions in the provided tables).

Although the entity-driven (problem oriented) approach to database design is emphasized in contemporary research, the output-driven approach is well suited to training. The instructor can select an appropriate approach for the assignments.

By using the forms in the appendix, the student can modify basic assumptions in each case study. For example, the student can change data relationships, expand the data dictionary, or develop additional user-requested database outputs. Also, these forms encourage a variety of solutions so the same cases can be assigned to different classes or teams and used over several academic terms without redundancy in student solutions. In addition, the instructor can adapt assignments or develop new ones to meet specific requirements, to suit different course levels, or to comply with various time constraints.

To adapt assignments, the instructor can narrow or broaden the scope of case studies, as follows:

* use a subset of selected data elements rather than all data elements listed in the data dictionary, or add new data elements,

* eliminate some of the database outputs discussed in the case study, modify some outputs, or add new outputs,

* delete, modify, or add data elements relationships, and revise sorting sequences,

* reduce or increase the number of many-to-many and many-to-one data relationships to simplify the database models or make it more complex.

Over a period of time, an instructor may accumulate for each case study a library of solutions reflecting different data analysis assumptions.

The book does *not* provide technical material on physical database design. Physical database design depends on concepts, facilities, and technical characteristics and constraints of the implemented database management system (DBMS). The methodology of physical design is not universal and varies from one DBMS to another. Therefore, the assignments for physical database design and the methodological material should be defined by the instructor to fit the scope of the course. The cases, however, provide basic material to extend logical database design into physical database design and prototyping.

The enterprises and characters presented in the case studies are wholly fictional and spring from the imagination of the author. Any resemblance to real persons or organizations is purely unintentional and coincidental.

TEXTS OF RELATED INTEREST

Database Systems for Management
Written by Jim Courtney and Dave Paradice, this core database text is intended to support a management-oriented first course in database-processing concepts and features:

* managerial orientation, emphasizing decision making and the impact of the database environment on the decision-making process,
* case examples, providing relevant business scenarios,
* the user's view, addressing the special managerial issues specific to database users,
* Municipal Authority of Lincoln, a comprehensive, ongoing case study.

A Primer on SQL
SQL continues to gain acceptance as the database language of choice. Yet to date almost all of the text material on SQL can be found only within chapters of database management textbooks. **A Primer on SQL** by Roy Ageloff is dedicated to a first course on SQL. It is flexible enough to be used in a variety of instructional settings. Furthermore, it provides simple explanations of SQL commands and numerous, easy to understand examples suitable for first-time SQL users. This 250-page text features:

* business user orientation, covering the basics of SQL,
* running, realistic case scenarios,
* margin notes and boxed topics, giving additional details about programs,
* follow-up exercises after each section; end-of-chapter exercises based on new case scenarios.

ACKNOWLEDGMENTS

I would like to express my gratitude to the reviewers who contributed greatly to the development of this text:

John Boggess
Purdue University

Susan Lenker
Central Michigan University

Jim Courtney
Texas A & M University

Doug May
Appalachian State University

Jan Harrington
Bentley College

Dave Paradice
Texas A & M University

Herman Hoplin
Syracuse University

I also want to express my heartfelt appreciation to Durward Jackson of California State University at Los Angeles for his help and support during the writing of this text. My thanks also go to my editor Susan Solomon for her professional expertise and many kindnesses, and to Becky Reece for her continued support. Finally, my colleague and friend, Rawlene Briar Watters, contributed immeasurably to this text.

Everyone involved with this casebook tried to find and correct errors, but as author I must take responsibility for any errors that managed to get into the finished product. Any comments, criticisms, suggestions, or improvements are welcome and appreciated. Write to me in care of Times Mirror/Mosby College Publishing.

Ludwig Slusky

TABLE OF CONTENTS

CHAPTER 1

A SAMPLE CASE STUDY: EAGLE CORPORATION

OBJECTIVE OF THE CASE STUDY

The objective of this case study is to introduce the problems and challenges of designing a flexible database as a major component of a system development effort to meet the identified needs of a corporation. By providing a realistic business environment, the case study approach is intended to reinforce concepts and provide the opportunity to apply database design techniques. This chapter introduces the Eagle Corporation case study, while Chapter 2 presents helpful notes on database design and suggested case study assignments and solutions. This sample case study is more complex than the case studies that follow in Chapters 4 through 8.

Although some solutions have been developed and presented in Chapter 2, this case may be used as the basis for additional student assignments, primarily in the areas of physical design and implementation. This case also can be used for logical database design using the entity-driven database design approach, so the results can be compared with those presented in Chapter 2 for the output-driven database design approach.

INTRODUCTION TO EAGLE CORPORATION

Eagle Corporation began twelve years ago as a small manufacturer of custom astronomy instruments and equipment. As it established a reputation for quality, sales and the number of products increased rapidly. Net sales were approximately $16 million last year and are expected to reach nearly $19 million this year. With the anticipated growth and continued diversification, the number of employees is expected to increase from 65 to nearly 75 by the end of this year.

Currently, Eagle Corporation has four product lines: custom telescopic equipment (accounting for nearly 40% of sales), photographic accessories, observatory equipment, and precision measurement instruments. Because of its expertise in precision optical equipment, the company is considering expanding into the competitive medical optics field.

PROBLEMS

As a consequence of Eagle Corporation's rapid growth, the partially manual information-processing systems which support inventory control, scheduling of work orders, and material purchases are no longer adequate to meet the company's needs for improved information on a timely basis. Some information processes were computerized three years ago with the assistance of Timeshare, Inc., the computer timesharing corporation that continues to maintain corporate data and to provide information processing. However, too much of the information is generated and maintained manually. Improved manual record keeping and several programming enhancements to a few computer files were only temporary solutions to control rapidly expanding corporate data. The problems and deficiencies of the system as a whole had not been addressed from the data-planning and data-modeling approach.

B.J. Kaplan, the controller, wanted a solution to the current system deficiencies and problems. As a result of her discussions with users and the manager of the data processing department, she was convinced that Eagle Corporation had outgrown the current transaction processing system to the extent that this was now the major obstacle to entering the medical optics market. She realized that development and implementation of a new system would require a major cooperative effort on the part of computer professionals and users that would require time and substantial financing. She was having difficulty, however, in justifying this expense to the executive committee, which would have to approve such a major expenditure.

Searching for persuasive documentation for the committee, B.J. reviewed a log of information-related problems and another log of user requests to the manager of the data processing department for new information reports and new system functions.

Some reported information-related deficiencies in the current system were:

* scheduling of work orders is a manual process that does not optimize usage of available resources,

* labor is allocated for each work order based on an estimate, which does not optimize usage of employees skills or consider an employee's hourly wage,

* customer information is not integrated with customer order and work order data,

* because of partial deliveries and back orders, it is extremely difficult to monitor outstanding purchase order quantities.

User requests indicated a need for ad hoc reports and online queries such as:

* comparison of actual and planned hours spent for a specific work order or specific customer order,

* comparison of actual and planned finish dates for a specific work order or specific customer order,

* analysis of daily labor costs showing labor costs and labor hours in total,

* status of any open purchase order (i.e. quantity on order and quantity received),

* listing of all customer orders (including order description, order number, order type, work orders, start and completion dates) for a specific customer during the past three years.

These documents presented isolated facts without any analysis of causes and needs, however, and she felt that the necessary comprehensive analysis would require several weeks. B.J. asked Chris Denning, the manager of the data processing department, to analyze current information-related problems and to develop a data flow diagram (DFD) that would serve as a model of the existing system.

At this time, Eagle was undergoing its annual audit. From her previous experience in public accounting, B.J. knew that auditors were very adept at identifying problem areas. She asked Jim Wilson, the partner on the audit, whether they had found any problems.

Jim noted that there were numerous problems with the system. "As part of the audit, we'll issue a management letter if we find problem areas or weaknesses. This way, you can address any potential problems. We already have a number of comments regarding the way Eagle keeps and maintains data. Inventory information isn't available on a timely basis, so it's difficult to plan parts stock levels and purchases. Parts purchases aren't efficient, stockouts are common, and customer orders are late because of parts shortages.

"And these problems have had a negative impact on employee morale. We've heard a number of complaints, mainly from people who are frustrated because they don't have information they need to do their jobs well. Or, they don't have information when they need it.

"But, I'm sure that you're aware of these problems. Have you considered getting a new system or modifying this one?"

B.J. chuckled. The auditors' objective comments regarding deficiencies of the current system would strengthen her proposal for a new system. "When do you plan to present the management letter?" she asked.
Jim thought for a moment. "Well, we're nearly finished with the fieldwork, so we'll try to get it to you by the end of the next week."

On Friday the CPA firm presented the management letter summarizing their findings. All materials supported B.J.'s conclusion that immediate changes were needed with the way corporate data was maintained and made available for management use.

At the next executive committee meeting, B.J. presented a detailed proposal, supported by a study of deficiencies in the current system and a cost/benefit analysis, to develop an information system tailored to Eagle's specific needs. Additional data would support new business activities and would allow for better modeling and planning.

The proposal would require a consultant to manage the project, two full-time systems analysts from the data processing department, and one database analyst. Chris Denning, the manager of the data processing department, would coordinate the development effort between data processing personnel and users and also assume responsibility for system implementation. Additional data would support new business activities, and would allow for better modeling and planning.

PLANNING PHASE

B.J. obtained the names of three available consultants who were recommended by a reputable consulting firm. She offered a contract to the second consultant she interviewed, Gregory Weiss. Gregory then interviewed and hired the database analyst, Henry Schmid.

This was not the first major project for Gregory as project leader. He was a database designer who had managed several projects and had worked on a broad range of data modeling assignments. He realized that before there could be any detailed planning, phases and activities would need to be identified. The major phases of the project were clear: planning, conceptualization, logical database design, physical database design, and implementation (that could also include prototyping on a smaller scale of database). Each phase required its own resources, methods, and tools for development.

During the planning phase, Gregory needed to apply the top-down methodology of data-driven system design, on one hand, and select software tools, on the other, in order to develop and control the project schedule. He visited a local computer store where, among several packages for project management, he chose an inexpensive one. Using this package, Gregory constructed a forecasted schedule of project development for each phase, activity, and task. (Refer to Figure 1-1.)

This project package also helped him to assign members of the team to various tasks, to calculate the time and cost of various resources involved in each task, both human (members of the project team, users) and computer, and the total time estimated for each task and for each member of the team.

Gregory realized that development should be rapid. He analyzed the need for resources required during other phases to be sure that resources would be available when needed.

The conceptualization phase, he thought, would require a software graphics tool capable of producing data and system models with some data dictionary capabilities. He expected to select such a tool among several offered on the microcomputer database. A logical system design phase would require more extensive use of a microcomputer-based data dictionary with graphics data models.

The physical design phase would require knowledge of a database management system (DBMS) to be selected soon and installed on Eagle's recently acquired Hewlett-Packard (HP) minicomputer. This phase would depend upon the capabilities and constraints of the DBMS in the areas of data organization, data navigation, and data security. Gregory intended to save time by converting data defined in the microcomputer-based data dictionary of logical data models into the data dictionary of the selected DBMS that would be used in the production environment. Therefore, he planned to completely identify logical and physical characteristics of the data in the same microcomputer data dictionary used for logical modeling.

Gregory had learned from experience that the process of data normalization typically is very labor intensive and time consuming. He looked for a tool that would automate this activity, but the few on the market were prohibitively expensive. He decided to use the microcomputer-based data dictionary to manipulate data during the data normalization process.

The implementation phase, in Gregory's view, consisted of two activities: database prototyping and full-scale database implementation. Prototyping would be limited to key and major elements on the database. He intended to use prototyping in three ways: to verify access paths to the data, to facilitate ad hoc reporting using the data query DBMS facility, and to train users in data query and data-reporting facilities.

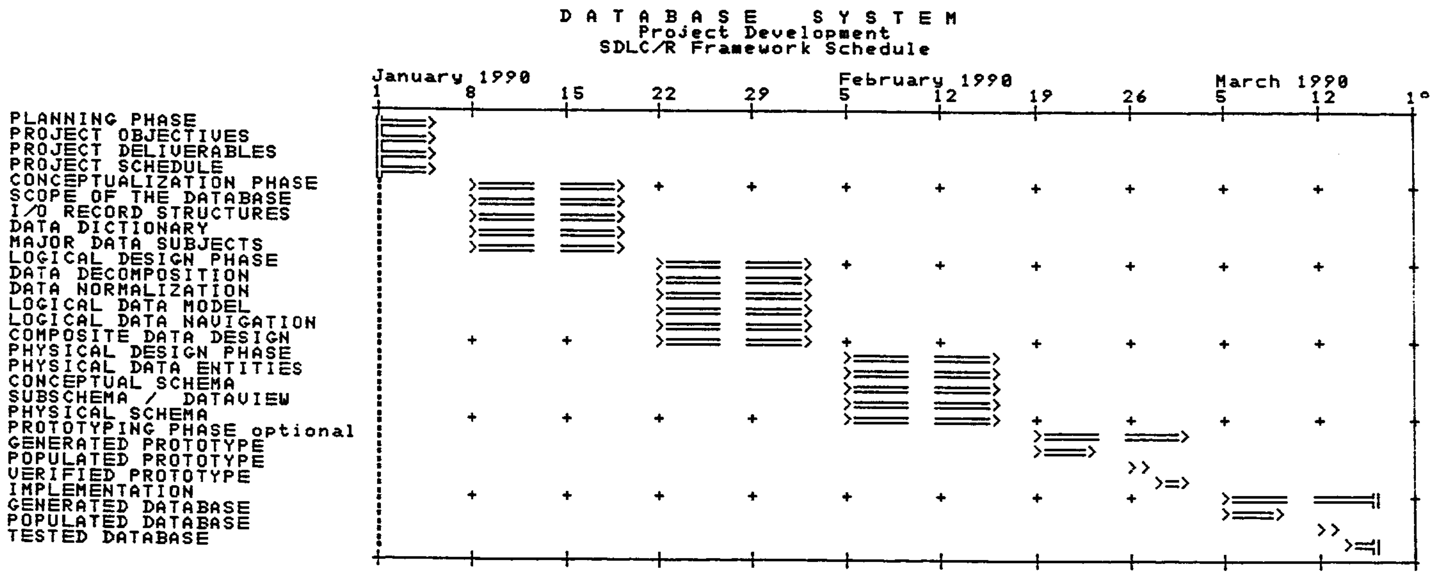

Figure 1-1. Database Project Development Schedule

Gregory was not sure during the planning phase whether prototyping should be done on the same host computer used for production database or on a stand-alone microcomputer utilizing a relational database. His colleague from the university almost convinced him that prototyping for a relational database on a DBMS such as ORACLE in a microcomputer environment is often a matter not of weeks but of days.
The final implementation of the database, of course, would be on the new HP minicomputer. Because of the large scope of the project and limited resources available for its development, Gregory stressed the importance of avoiding any redundancy of activities during the development cycle. The team could not afford to duplicate its efforts.

Data definitions on the element or record level would be needed to support database development from conceptual and logical design through implementation. Although this information would originate with (and initially be kept on) the microcomputer, he intended during the implementation phase to migrate (transfer as a file) all data definitions from the microcomputer into the production DBMS data dictionary. He knew from experience that this could be done easily using one of the DBMS utilities.

Finally, Gregory felt that he had identified all resources (human, computer, and methods) for all phases of development and specified their use in an addendum to the database development schedule. He considered this an important facet of planning because he intended to continue using the project planning software to control the project by comparing the actual schedule of development against the plan.

Because the project was new and he did not know members of the team, Gregory had the difficult task of identifying which person would be the best candidate for a particular task. Careful planning of work assignments and an emphasis on deliverables would be essential. Otherwise, members of the team would probably compete for preferred work assignments and would endlessly argue the merits of different approaches. He also had the formidable task of creating a sense of teamwork and instilling a spirit of cooperation.

Although the schedule of activities accounted for time spent on different tasks, Gregory emphasized that to be successful, the team had to be deliverables oriented, rather than activities oriented. To reinforce this point, in addition to the schedule of activities, he produced a schedule of deliverables that represented a team commitment to the project. This schedule listed deliverables only; details of activities were eliminated. Planning ahead, he decided that, to correct potential problems with work assignments, some slack time for tasks in the critical path was needed.