DBMS

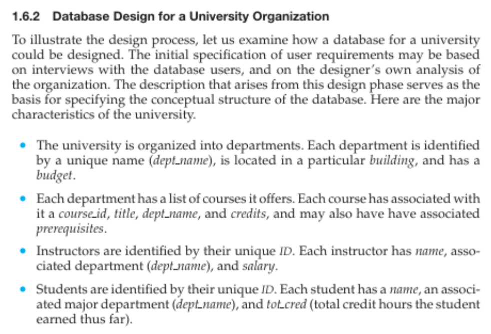
CHAPTER-1

1.1This chapter has described several major advantages of a database system. What are two disadvantages?

1.2 List five ways in which the type declaration system of a language such as Java or C++ differs from the data definition language used in a database.

1.3 List six major steps that you would take in setting up a database for a particular enterprise.

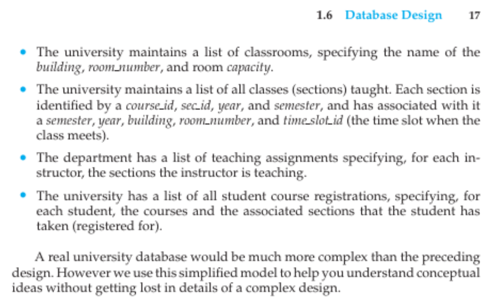
1.4 List at least 3 different types of information that a university would main tain, beyond those listed in Section 1.6.2.



1.6.2 Database Design for a University Organization

To illustrate the design process, let us examine how a database for a university could be designed. The initial specification of user requirements may be based on interviews with the database users, and on the designer's own analysis of the organization. The description that arises from this design phase serves as the basis for specifying the conceptual structure of the database. Here are the major characteristics of the university.

- The university is organized into departments. Each department is identified by a unique name (*dept_name*), is located in a particular *building*, and has a *budget*.
- Each department has a list of courses it offers. Each course has associated with it a *course_id*, *title*, *dept_name*, and *credits*, and may also have have associated *prerequisites*.
- Instructors are identified by their unique *ID*. Each instructor has *name*, associated department (*dept_name*), and *salary*.
- Students are identified by their unique *ID*. Each student has a *name*, an associated major department (*dept_name*), and *tot_cred* (total credit hours the student earned thus far).



- The university maintains a list of classrooms, specifying the name of the *building*, *room_number*, and room *capacity*.
- The university maintains a list of all classes (sections) taught. Each section is identified by a *course_id*, *sec_id*, *year*, and *semester*, and has associated with it a *semester*, *year*, *building*, *room_number*, and *time_slot_id* (the time slot when the class meets).
- The department has a list of teaching assignments specifying, for each instructor, the sections the instructor is teaching.
- The university has a list of all student course registrations, specifying, for each student, the courses and the associated sections that the student has taken (registered for).

A real university database would be much more complex than the preceding design. However we use this simplified model to help you understand conceptual ideas without getting lost in details of a complex design.

SECTION

1.5 Suppose you want to build a video site similar to YouTube. Consider each of the points listed in Section 1.2, as disadvantages of keeping data in a f ile-processing system. Discuss the relevance of each of these points to the storage of actual video data, and to metadata about the video, such as title, the user who uploaded it, tags, and which users viewed it.

1.6 Keyword queries used in Web search are quite different from database queries. List key differences between the two, in terms of the way the queries are specified, and in terms of what is the result of a query

1.7 Listfour applications you have used that most likely employed a database system to store persistent data.

1.8 List four significant differences between a file-processing system and a DBMS.

1.9 Explain the concept of physical data independence, and its importance in database systems.

1.10 List five responsibilities of a database-management system. For each re sponsibility, explain the problems that would arise if the responsibility were not discharged.

1.11 Listat least two reasons why database systems support data manipulation using a declarative query language such as SQL, instead of just providing a alibrary of C or C++ functions to carry out data manipulation.

1.12 Explain what problems are caused by the design of the table in Figure 1.4.

1.13 What are five main functions of a database administrator?

1.14 Explain thedifferencebetweentwo-tierandthree-tierarchitectures. Which is better suited for Web applications? Why?

1.15 Describe at least 3 tables that might be used to store information in a social-networking system such as Facebook.

SECTION 1.2

Purpose of Database Systems Database systems arose in response to early methods of computerized manage ment of commercial data. As an example of such methods, typical of the 1960s, consider part of a university organization that, among other data, keeps infor mation about all instructors, students, departments, and course offerings. One way to keep the information on a computer is to store it in operating system f iles. To allow users to manipulate the information, the system has a number of application programs that manipulate the files, including programs to: • Addnewstudents,instructors, and courses • Register students for courses and generate class rosters • Assigngradestostudents,computegradepointaverages(GPA),andgenerate transcripts System programmers wrote these application programs to meet the needs of the university. New application programs are added to the system as the need arises. For example, suppose that a university decides to create a new major (say, computer science). Asaresult,theuniversitycreatesanewdepartmentandcreatesnewper manentfiles(or addsinformationto existing files) to recordinformation about all the instructors in the department, students in that major, course offerings, degree requirements, etc. The university may have to write new application programs to deal with rules specific to the new major. New application programs may also have to be written to handle new rules in the university. Thus, as time goes by, the system acquires more files and more application programs. This typical file-processing system is supported by a conventional operat ing system. The system stores permanent records in various files, and it needs different application programs to extract records from, and add records to, the ap propriate files. Before database management systems (DBMSs) were introduced, organizations usually stored information in such systems. Keepingorganizational information in a file-processing system has a number of major disadvantages: 4 Chapter 1 Introduction • Data redundancy and inconsistency. Since different programmers create the files and application programs over a long period, the various files are likely to have differentstructures and the programsmaybewritteninseveral programminglanguages.Moreover,thesameinformationmaybeduplicated in several places (files). For example, if a student has a double major (say, music and mathematics) the address and telephone number of that student mayappear in a file that consists of student records of students in the Music department and in a file that consists of student records of students in the Mathematicsdepartment.Thisredundancyleadstohigherstorageandaccess cost. In addition, it may lead to data inconsistency;thatis,thevariouscopies ofthesamedatamaynolongeragree.Forexample,achangedstudentaddress may be reflected in the Music department records but not elsewhere in the system. • Difficulty in accessing data. Suppose that one of the university clerks needs to find out the namesof all students who live within a particular postal-code area. The clerk asks the data-processing department to generate such a list. Because the designers of the original system did not anticipate this request, there is no application program on hand to meet it. There is, however, an application program to generate the list of all students. The university clerk has now two choices: either obtain the list of all students and extract the needed information manually or ask a programmer to write the necessary application program.Bothalternativesareobviouslyunsatisfactory. Suppose that such a program is written, and that, several days later, the same clerk needs to trim that list to include only those students who have taken at least 60 credit hours. As expected, a program to generate such a list does not exist. Again, the clerk has the preceding two options, neither of which is satisfactory. The point here is that conventional file-processing environments do not allow neededdata to beretrievedin a convenient and efficient manner. More responsive data-retrieval systems are required for general use. • Data isolation. Because data are scattered in various files, and files may be in different formats, writing new application programs to retrieve the appropriate data is difficult. • Integrity problems. The data values stored in the database must satisfy cer tain types of consistency constraints. Suppose the university maintains an account for each department, and records the balance amount in each ac count. Suppose also that the university requires that the account balance of a department may never fall below zero. Developers enforce these constraints in the system by adding appropriate code in the various application pro grams. However, when new constraints are added, it is difficult to change the programstoenforcethem.Theproblemiscompoundedwhenconstraints involve several data items from different files. • Atomicity problems. A computer system, like any other device, is subject to failure. In many applications, it is crucial that, if a failure occurs, the data 1.2 Purpose of Database Systems 5 be restored to the consistent state that existed prior to the failure. Consider a program to transfer $500 from the account balance of department A to the account balance of department B. If a system failure occurs during the execution of the program, it is possible that the $500 was removed from the balance of departmentAbutwasnotcreditedtothebalanceofdepartmentB, resulting in an inconsistent database state. Clearly, it is essential to database consistency that either both the credit and debit occur, or that neither occur. That is, the funds transfer must be atomic—it must happen in its entirety or not at all. It is difficult to ensure atomicity in a conventional file-processing system. • Concurrent-accessanomalies.Forthesakeofoverallperformanceofthesys tem and faster response, many systems allow multiple users to update the data simultaneously. Indeed, today, the largest Internet retailers may have millions of accesses per day to their data by shoppers. In such an environ ment, interaction of concurrent updates is possible and may result in incon sistent data. Consider department A, with an account balance of $10,000. If two department clerks debit the account balance (by say $500 and $100, re spectively) of department A at almost exactly the same time, the result of the concurrent executions may leave the budget in an incorrect (or inconsistent) state. Supposethattheprogramsexecutingonbehalfofeachwithdrawalread the oldbalance, reducethat valuebytheamountbeingwithdrawn,andwrite the result back. If the two programs run concurrently, they may both readthe value $10,000, and write back $9500 and $9900, respectively. Depending on which one writes the value last, the account balance of department A may contain either $9500 or $9900, rather than the correct value of $9400. To guard against this possibility, the system must maintain some form of supervision. Butsupervisionis difficult to providebecause datamaybe accessedbymany different application programs that have not been coordinated previously. As another example, suppose a registration program maintains a count of students registered for a course, in order to enforce limits on the number of students registered. When a student registers, the program reads the current count for the courses, verifies that the count is not already at the limit, adds one to the count, and stores the count back in the database. Suppose two students register concurrently, with the count at (say) 39. The two program executions may both read the value 39, and both would then write back 40, leading to an incorrect increase of only 1, even though two students suc cessfully registered for the course and the count should be 41. Furthermore, suppose the course registration limit was 40; in the above case both students would be able to register, leading to a violation of the limit of 40 students. • Security problems. Not every user of the database system should be able to access all the data. For example, in a university, payroll personnel need to see only that part of the database that has financial information. They do not need access to information about academic records. But, since applica tion programs are added to the file-processing system in an ad hoc manner, enforcing such security constraints is difficult. 6 Chapter 1 Introduction These difficulties, among others, prompted the developmentof database sys tems. In what follows, we shall see the concepts and algorithms that enable database systems to solve the problems with file-processing systems. In most of this book, we use a university organization as a running example of a typical data-processing application.

CHAPTER-2

2.1Consider the relational database of Figure 2.14. What are the appropriate primary keys?

2.2 Considertheforeignkeyconstraintfromthedept nameattributeofinstructor to the department relation. Give examples of inserts and deletes to these relations, which can cause a violation of the foreign key constraint.

2.3 Consider the time slot relation. Given that a particular time slot can meet more than once in a week, explain why day and start time are part of the primary key of this relation, while end time is not.

2.4 In the instance of instructor shown in Figure 2.1, no two instructors have the same name. From this, can we conclude that name can be used as a superkey (or primary key) of instructor?

2.5 What is the result of first performing the cross product of student and advisor, and then performing a selection operation on the result with the predicate s id = ID? (Using the symbolic notation of relational algebra, this query can be written as sid= ID(student × advisor).) employee (person name, street, city) works (person name, company name, salary) company (company name, city)

branch(branch name, branch city, assets) customer (customer name, customer street, customer city) loan (loan number, branch name, amount) borrower (customer name, loan number) account (account number, branch name, balance) depositor (customer name, account number)

Figure 2.15 Banking database for Exercises 2.8, 2.9, and 2.13.

2.6 Consider the following expressions, which use the result of a relational algebra operation as the input to another operation. For each expression, explain in words what the expression does. a. b. c. year≥2009(takes) year≥2009(takes student student) ID,name,course id(student takes)

2.7 Consider the relational database of Figure 2.14. Give an expression in the relational algebra to express each of the following queries: a. Findthe names of all employees who live in city “Miami”. b. Findthenamesofallemployeeswhosesalaryisgreaterthan$100,000. c. Find the names of all employees who live in “Miami” and whose salary is greater than $100,000.

2.8 Consider the bank database of Figure 2.15. Give an expression in the rela tional algebra for each of the following queries. a. Findthe names of all branches located in “Chicago”. b. Find the names of all borrowers who have a loan in branch “Down town”.

2.9 Consider the bank database of Figure 2.15. a. Whatarethe appropriate primary keys? b. Givenyourchoiceofprimarykeys,identifyappropriateforeignkeys.

2.10 Consider the advisor relation shown in Figure 2.8, with s id as the primary key of advisor. Suppose a student can have more than one advisor. Then, would s id still be a primary key of the advisor relation? If not, what should the primary key of advisor be?

2.11 Describe the differences in meaning between the terms relation and relation schema. Bibliographical Notes 55

2.12 Consider the relational database of Figure 2.14. Give an expression in the relational algebra to express each of the following queries: a. Find the names of all employees who work for “First Bank Corpora tion”. b. Findthenamesandcitiesofresidenceofall employeeswho work for “First Bank Corporation”. c. Findthenames,streetaddress,andcitiesofresidenceofallemployees whoworkfor “First Bank Corporation” and earn more than $10,000.

2.13 Consider the bank database of Figure 2.15. Give an expression in the rela tional algebra for each of the following queries: a. Findall loan numbers with a loan value greater than $10,000. b. Find the names of all depositors who have an account with a value greater than $6,000. c. Find the names of all depositors who have an account with a value greater than $6,000 at the “Uptown” branch.

2.14 List two reasons why null values might be introduced into the database.

2.15 Discuss the relative merits of procedural and nonprocedural languages.