

A Graduate Student Administration System using Delphi 3.0 and MS-Access

Xinfa Cai and T. E. O'Neil
Department of Computer Science
University of North Dakota
Grand Forks, ND 58202-9015
Cai@cs.und.edu oneil@cs.und.edu

Abstract

MS-Access and Delphi are powerful tools for building databases and associated applications. To build a sophisticated graduate student database management system for the Computer Science Department, MS-Access is used to create and hold database tables and Delphi 3.0 is used to develop the student administration system. The administration system includes 11 MS-Access tables, 2 Delphi data modules, and 47 Delphi forms. The database tables hold information on each applicant and graduate student, including general information, academic record, history of assistantships, program of study, project/thesis outline, and after-graduation information. The user-friendly application interface includes multi-layer forms. The main form is the access point with a few menus. Other forms are accessed from those menus. The users can access and update the database tables more conveniently and efficiently by using the developed application than by using MS-Access directly. The users can also retrieve information from multiple tables and produce summary reports. The management system will benefit the department's administration of graduate students. It can also be modified to apply to other departments and degree programs.

1. Introduction

In the last twenty years, the development of computers and computing technology has had incredible impact on academics, business and the lives of individuals. It has also resulted in an explosive increase in the amount of information we deal with in our daily lives. This explosion of information has not occurred in a logical, well-thought-out manner, so we need figure out how to collect, store, retrieve, update and analyze the information. Database technology helps us in this regard.

In the Computer Science Department at the University of North Dakota, there are approximately 20 enrolled graduate students, and 5-10 new graduate students are recruited every year. The program started in 1986, so we currently have a filing cabinet full of information for more than 100 students. The mass of information has now reached the point where automated record keeping is essential.

Anyone who has ever served as a graduate director knows how difficult it is to keep track of graduate students. Responding to a request for a list of current graduate students is a major research project. We need to keep a wide variety of information on each student and update it on a regular basis. For applicants, we need to know who they are, where they're from, what their grades and test scores are, what courses they have taken to meet the program prerequisites, and their acceptance status. After students begin the program, we need to keep a history of what courses they take, what responsibilities they have relating to teaching, research, or service assistantships, and what degree program requirements they have completed or not completed. For students who have left the program, we need to know whether they graduated and where they went afterward.

The tracking system doesn't just store information, it also implements common queries and facilitates the preparation of various forms and reports required by the graduate school. We decided to use MS Access for the database and Delphi for designing the interface. We chose Access because it is available on all the desktop machines in the department office. We decided to use Delphi to maximize power and flexibility in creating user-friendly forms and applications running on top of the database tables.

2. The Underlying Database System

This project includes several components, including the actual database, the management application, and associated help files, data files, and image files. The database stores the actual information. It is logically organized in four sections and implemented in 11 linked tables. The tables are summarized below and in Figure 1.

General information: StudentGeneral and CourseInformation.

The StudentGeneral table contains name, address, birthdate, and origin of each student in the database. The CourseInformation table contains the names and numbers of courses that graduate students in computer science are likely to take.

Applicant information: Applicant and Prerequisite.

The Applicant table contains information on the undergraduate institution and test scores of applicants. The Prerequisite table identifies the courses an applicant has taken to meet the nine prerequisites of the M.S. program at UND.

Academic information: Advisor, RAInformation, TAInformation, TakenCourse, TakenCoreExam and StudyProgram.

The Advisor table contains the history of advisor appointments for each student, whether he/she has chosen the computer science track or the applied computing track, and whether he/she has submitted a thesis/project outline and a program of study to the graduate school. The StudyProgram table contains the program of study information -- the list of courses the student will use to complete the M.S. requirements. The TakenCourse table lists the courses each student has completed. The RAInformation and TAInformation tables contain a history of appointments and activities for students who are awarded assistantships. The TakenCoreExam table

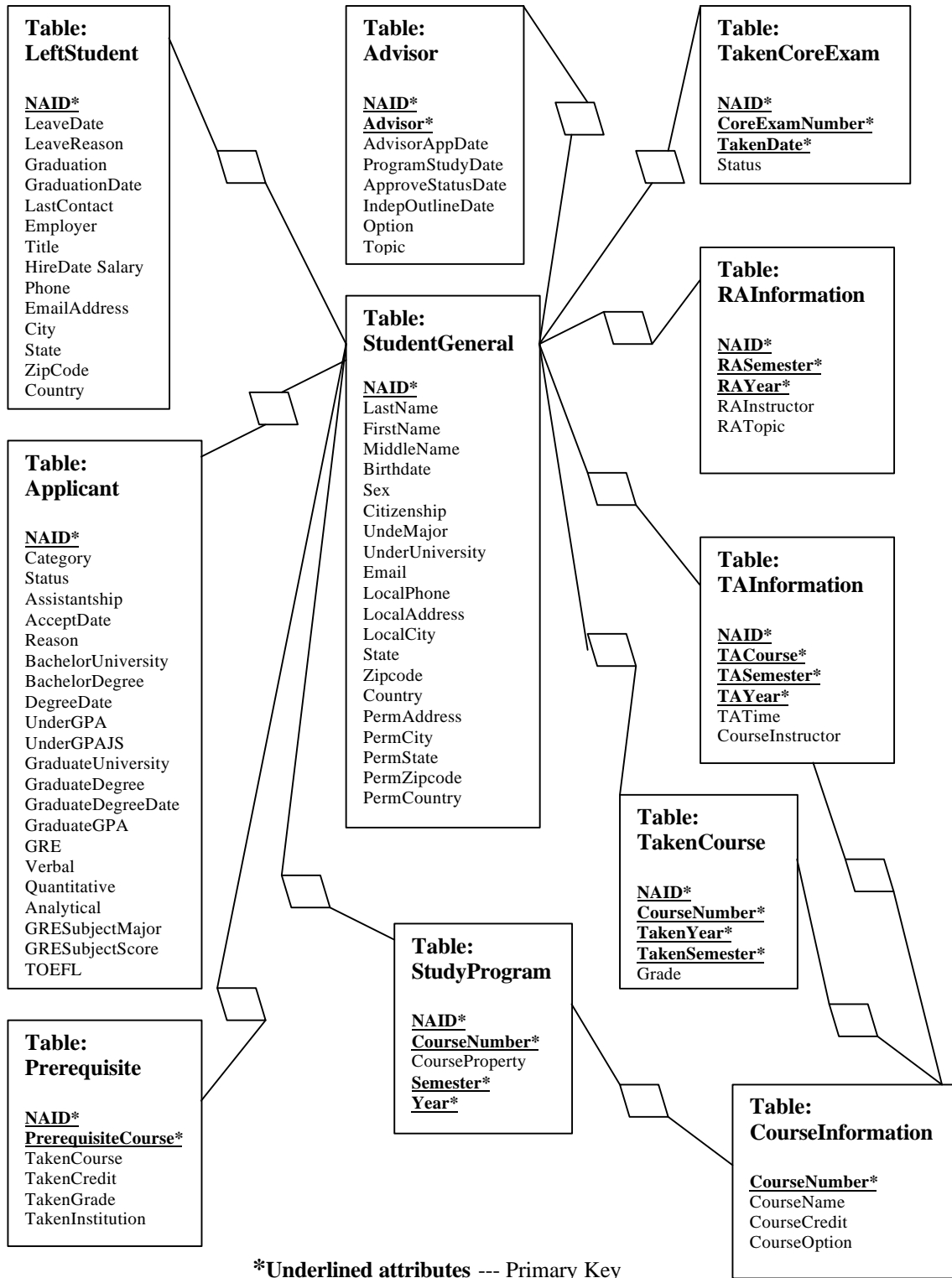


Figure 1. The relationship between the 11 tables of the database

contains the history of results of the comprehensive examination, which is offered in four sections each semester.

Alumni information: LeftStudent.

The LeftStudent table contains address and employment information for students who left the program. It also indicates whether they graduated, and if not, their reason for leaving.

3. The Management Application

The Delphi project contains a source code directory, an image directory, a data file directory, a help file, and the management application. All source files and codes are included within the source code directory. Those source files are compiled and linked to form the application.

The images, help file and data files are used by the running application. The image directory stores all image files needed by the application. There are 36 bitmap files in this directory. The data file directory includes various files containing information needed by the application, such as a list of the names of current faculty members.

The source directory includes a project file, 50 Pascal units, 48 form files, 2 data module files and a help file. The forms are visual components (windows) that appear at run-time. The Pascal units contain code for procedures that are invoked when various events occur at run-time. The main form appears when the program is started. The others appear in response to user actions. Figure 2 shows how the forms are related to each other.

The main menu of the application offers the user four choices: **Database**, **Table**, **View**, and **Help**. The **Database** choice just displays the name of the database file to which the system is connected. The **Help** choice invokes the on-line users' manual, complete with contents, index, and item-search capabilities. The **View** option allows users to quickly look up information on students and applicants. The **Table** option presents a list of forms that provide full editing capabilities for the associated database tables.

The form associated with the general student information table is shown in Figure 3. Its format is typical of many of the other forms. The upper section directly displays the database table information as a list of records, and the lower section gives an organized view of the data fields for a single record. The NAID number is the student's university identification number. It serves as the primary key for searching the database tables.

The bar that separates the upper and lower sections of the form contains some search tools, including navigation buttons and a **Filter** button. The **Filter** button invokes a form that allows the user to craft an SQL query. A list of field names and relational operators is presented to aid in building the query. When the query is applied, only those records that match the query are displayed on the form.

Fields can be edited by typing information in either the upper or lower section of the form. Records can be added or deleted by using the '+' and '-' buttons on the navigation bar. When the **Print** button is pressed, an image of the form is sent to a local printer.

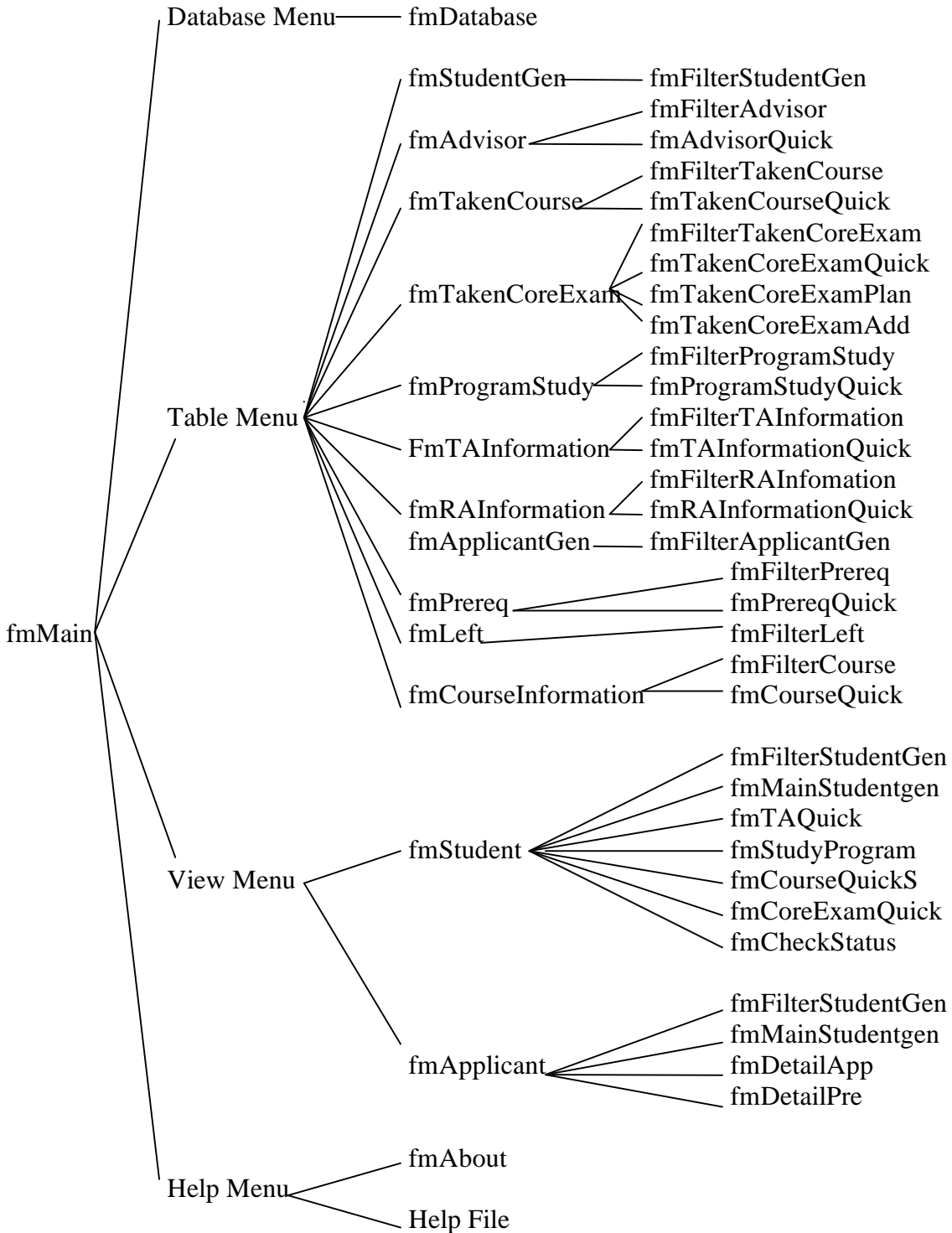


Figure 2. The relationship between the forms

NAID	LastName	FirstName	MiddleName	BirthDate	Sex	Citizenship
▶ 1024876	Ken	Hanks	J	12/5/1959	Male	USA.
1103261	Manda	Bake	E	5/23/1963	Male	USA.
1107101	Jo	Harry	M	10/19/1964	Female	Hong Kong
1187830	Cruise	Tom		5/9/1962	Male	USA.

You Can Edit, Add, Delete and Save Records in This Page

NAID: Sex: Citizenship:

First Name: Undergraduate Major:

Last Name: Undergraduate University:

Middle Name: Email:

Birth Date: Phone:

Local Address
 Address:
 City: Zip Code:
 State: Country:

Permanent Address
 Address:
 City: Zip Code:
 State: Country:

Figure 3. The General Student Information Form

The **View** option on the main menu allows the user to retrieve summarized information on various topics for current students and applicants. The **View Student** form is shown in Figure 4. The form has a similar layout to the general information form described above, except that the lower section contains buttons, and no editing is allowed. The

NAID	LastName	FirstName	MiddleName	BirthDate	Sex	Citizenship
▶ 1024876	Ken	Hanks	J	12/5/1959	Male	USA.
1103261	Manda	Bake	E	5/23/1963	Male	USA.
1107101	Jo	Harry	M	10/19/1964	Female	Hong Kong

Filter Student

Detail Information

Figure 4. The View Student Form

buttons on this form provide a means of tracking the progress of a student. Pressing the **General Information** button causes the general background of a student to be displayed. The **TA/RA** button displays the student's semester-by-semester history of appointments as a teaching or research assistant. The **Core Exam** button produces a report of the student's performance in the different areas of the department's comprehensive examination required for the M. S. degree. The **Courses Taken** button shows the student's history of coursework and grades. The **Program of Study** button displays the selection of courses the student will use to complete the M. S. requirements. The **Status Check** button gives information on various other requirements relating to the student's thesis or applied computing project. This information helps in advising the student on what requirements remain to be fulfilled before graduation.

The form is divided into several sections:

- Drag Name Here:** A list of students with their names and IDs:
 - Len , Jenke : 2681854
 - Manda , Mike : 2455838
 - Lister , Lisa : 2745135
 - Bin , Tian : 2826925
 - Ken , Ben : 2858959
 - Poland , Lisa : 2358379
 - Silk , Jenifer : 1201638
 - Kent , Tom : 2570203
- Select Semester:** Three buttons:
 - Spring (with a calendar icon)
 - Fall (with a calendar icon)
 - Clear Box (with a trash can icon)
- Core Exams:** Four boxes representing exam areas:
 - Algorithms(CSci575):** Contains Len , Jenke : 2681854, Lister , Lisa : 2745135, Bin , Tian : 2826925.
 - Operating Systems(CSci551):** Contains Len , Jenke : 2681854, Lister , Lisa : 2745135, Bin , Tian : 2826925.
 - Database Systems(CSci513):** Contains Len , Jenke : 2681854, Lister , Lisa : 2745135, Poland , Lisa : 2358379.
 - Theoretical Foundations(CSci522):** Contains Len , Jenke : 2681854, Silk , Jenifer : 1201638, Kent , Tom : 2570203.

Figure 5. The Core Exam Planning Form

The form designed for planning the core exam, shown in Figure 5, takes advantage of Delphi's drag-and-drop programming utilities. With most objects in Delphi's component hierarchy, the developer can use the events `OnStartDrag`, `OnDragOver`, `OnDragDrop`, and `OnEndDrag` to program what happens when the user drags an item from one object to another on a form. These events are employed in the core exam planning form to allow the user to drag names from a list of students and drop them in boxes representing areas of the exam. The exam is offered in four areas each semester, and students can sign up for one or more areas. The drag-and-drop interface makes it easy to specify which students will take which sections of the exam. A similar form is used to record grades for the exam.

4. Conclusion

The Graduate Student Administration System is being deployed in the Computer Science Department at UND in the current semester. The system will be refined and enhanced as we gain experience with it. Entering ten years' worth of data to initialize the system is a substantial task, and updating the system will require extra time on a day-to-day basis. However, we anticipate that retrieving information on graduate students will be much quicker with the automated system, and the time saved on retrieving information will far exceed the time spent on data entry.

References

- [1] Date, C. J., and H. Darwen, *A Guide to the SQL Standard*, Addison-Wesley, Reading, MA, 1994.
- [2] *Delphi 3 User's Guide*, Borland International, Inc., Scotts Valley, CA, 1997.
- [3] Newman, W. M., and M. G. Lamming, *Interactive System Design*, Addison-Wesley, Workingham, England, 1995.
- [4] Silberschatz, A., H. Korth, and S. Sudarshan, *Database System Concepts*, McGraw-Hill, New York, 1997.
- [5] *UND 1997-1999 Academic Catalog*, University of North Dakota, Grand Forks, ND, 1997.