tutors
students
tutoring schedule
courses
requests

She thinks a bit longer. "We need to track whether students attended the sessions they scheduled. That is important, but is it a new topic? It could be part of scheduling." Terry wanted one more thing, she remembers. She wanted to track student success. To Sharon that seems like a different topic entirely. She recalls that Bill Collins in his class always insisted that a good database like a good table should be focused on a single topic. She decides to leave the list as it is.

ATTRIBUTES

These are things that define entities (The entity customer has attributes like name and address).

Things You Should Know

Identifying the major topics of a database is an important exercise. It helps provide a clearer sense of just what the database is about. It is also the first step toward identifying the "entities" that will be used in the database design.

One way to begin identifying the major themes is to look at the nouns in your notes. See if they cluster together around certain themes. These themes are most likely the major topics of your database. We will look at this technique more closely later when we talk about defining entities and attributes.

It is important to note that a database may contain several themes, but all those themes should relate to a single overarching topic like tutoring. If there is more than one overarching topic, it may indicate that you should develop additional databases.

WRITING THE STATEMENT OF WORK

Now that she has the big topics in mind, she begins to compose the statement of work. She begins with the history. The history is a statement of the problem. It can narrate how the current situation came to be the way it is. Sharon thinks about the things she saw and the things that Terry told her.

> For a long time the tutoring program has used a paper schedule to sign students up for tutoring. Tutors identify their schedule for a 2-week period, and then a schedule is printed and placed in the computer lab. Students look through the schedule for sessions that match courses they are taking and the times they have available. This system has worked and continues to work, but it has several significant problems. For one, it can be difficult for students to find appropriate tutoring sessions. The paper forms are difficult to navigate and understand. Additionally, it is very difficult for the tutoring program to track the students using the tutoring. It is difficult or impossible to track demographic information. It is also difficult to assure that students are enrolled in the courses they receive tutoring in. Even tracking tutors' hours can be difficult.
>
> A database with a client application could significantly improve the situation by providing a flexible, searchable schedule for students; better tracking of demographics and eligibility; and better tracking of hours tutored.

She pauses. That was hard to get going, but once she got started, it flowed pretty well.

> The tutoring database will be designed to manage the tutoring program at the college.

She isn't real happy with that as an opening sentence. She modifies it a little and forges ahead. It proves to be a lot harder than she imagined. The statement has to

include all the general points but still be concise enough to give a clear indication of the purpose and functions of the database. After a lot of effort, she had this preliminary statement:

> The tutoring database will manage data for the tutoring program at the college. It will track available tutors and the courses they can tutor. It will also track each tutor's tutoring schedule. The database will store demographic information for students who register for tutoring. This information will be private and used only to generate general reports that include no personal information. Students, who have registered, will be able to sign up for available tutoring sessions for courses in which they are enrolled. The database will track whether students attended their scheduled sessions.

Sharon looks it over carefully. What about the data about student success? Should that be a part of this database, or should that be a separate project? She decides to set it aside until she has talked with Terry.

She also wonders if she should state some of the things the database *won't* do. Things such as the following:

> The database can be used to get the hours worked for each tutor, but it will not process pay or provide any payroll information.
>
> The database will not validate student information against the school's registration database.

CONSTRAINTS

These are limits on what the database will do. Later we will see that you can also set constraints on the types and range of data that can be entered into a column in a table.

For the moment, she can't think of any other *constraints.*

She consults an example her instructor gave her to look at. The next step is to set out the objectives for the database. She spends some time thinking about this. Most of the objectives are spelled out in the scope. She pulls out some of the main points and makes a list.

- Streamline the process by which tutors enter their schedules and students sign up for them
- Improve tracking of demographic data of students using the tutoring program
- Improve tracking of tutors' hours and students' use of tutoring sessions

Next she needs to add tasks and a timeline. She jots down some notes on a paper. The first thing she will have to do is to gather information. She needs to know all the relevant data and processes. How long will that take? She makes a rough guess of 2–3 weeks. Then she will have to evaluate all the information she has gathered and use it to start developing a list of business rules and the first rough model of the data. That could take another couple weeks. Next she will have to refine and normalize the model. Sharon thinks she can do this in 2 or 3 days. Then she needs to actually make the database. That won't take long. She can probably do that part in a couple of hours. What then? Sharon muses for a while. The last part may take a fair amount of time. She will need to test the database and make sure that it meets all of Terry's needs. She will also have to test for security issues and privacy. That could take two or more weeks of intense work. Where does that put her? Sharon calculates and taking the longer times in each case comes up with 9 or 10 weeks. None of this is counting the fact that it will take a completely different development project to create a client application for Terry, the tutors, and students to interact with the database. But, Sharon says to herself, one project at a time.

Sharon almost has everything she needs for the statement of work, but there is still something missing. After a while it occurs to her: Every task should also have a deliverable, something concrete she can show Terry to let her know that the database is on track.

Sharon spends the next couple of hours completing her statement of work.

THINGS TO THINK ABOUT

Estimating Times

One of the most difficult things for anyone who is new to developing databases is estimating the time it will take to complete the various tasks. Experience will help, but before you have enough experience, how do you even begin to guess an appropriate time?

There are some techniques that can help. One is to make a weighted average. To do this, write down your most optimistic time (O)—if everything goes perfect; your best guess at the probable time it will take (Pt); and your most pessimistic time estimate (p)—if everything goes wrong. Add them all together, but multiply your most probable estimate by 3, then divide the sum by five.

$$(0 + Pt \times 3 + p)/5$$

What other ways can you think of to help your time estimates be more accurate?

REVIEWING THE STATEMENT OF WORK

The following afternoon Sharon returns to Terry's office and shows her the statement. As Terry looks it over, Sharon says, "It is important that we both are clear about what we are working on. I don't want to go off and make a database and then find out it is not what you had in mind at all."

"No, I can see that is a really good idea." She sets the paper down. "What about the surveys of student success?"

"I thought about that, and I am not sure. Sometimes I think that does belong in this project, and other times, I think that it is a separate project on its own. I am not sure how we could get objective data on their success, but we could include evaluations by students or a quarterly survey. If we build the database as I have described it, we should be able to add the success-tracking features later or we could look at adding a second database devoted to tracking student success."

"OK, I can live with that. It would be nice if you could validate student information."

"Yes, but I don't really know how to do that. I also think it unlikely that I would be granted the permissions I would need on the school's registration database. You might be able to get the school's developers to look at that piece later."

"Fair enough. One other thing you don't have here, and I am not sure we talked about it, but it would be nice if students could request tutoring in courses that we don't currently have tutors for. It would help us know where the need is and where we need to try to recruit new tutors."

"That shouldn't be a problem. I can add that."

"Good. What do you need to proceed?"

"Well, let's go over the tasks and timeline. First, I am going to need to gather some information. I am going to need to see how you have been doing things. I will need to talk to some tutors, and maybe some students, and I probably need to see the reports you make to ensure that the database contains all the information you require. Then I will need to analyze all the information I get and begin to make a data model. After all that, I can actually make the database and test it."

Terry studies the timeline. "This is very clear and well done. How realistic do you think this timeline is?"

Sharon smiles. "It represents my very best guess. It could go faster if everything works out well, but it could also go slower if I encounter problems. I tried to be very conservative on the times, so I think there is a good chance it can be completed on schedule."

"Good, it would be ideal if the database could be in place by the beginning of next term."

Sharon warns, "There is another piece to all this. A client application needs to be developed so you, the students, and tutors can interact safely and easily with the database. But that is really a separate project."

Terry smiles. "You're right. We can tackle that when we have finished with the database."

"Tell you what, I will come by tomorrow with a revised version of this statement, and I will give you a preliminary plan of where we go next."

Terry stands up and puts out her hand to shake. "Sounds good. I look forward to working with you on this."

THE STATEMENT OF WORK

Home, later. Sharon revised the statement of work to include student requests. Here is her completed statement of work:

Statement of Work: Tutoring Database Project

History

For a long time the tutoring program has used a paper schedule to sign students up for tutoring. Tutors identify their schedule for a 2-week period, and then a schedule is printed and placed in the computer lab. Students look through the schedule for sessions that match courses they are taking and the times they have available. This system has worked and continues to work, but it has several significant problems. For one, it can be difficult for students to find appropriate tutoring sessions. The paper forms are difficult to navigate and understand. Additionally, it is very difficult for the tutoring program to track the students using the tutoring. It is difficult or impossible to track demographic information. It is also difficult to assure that students are enrolled in the courses they receive tutoring in. Even tracking tutors' hours can be difficult.

A database with a client application could significantly improve the situation, by providing a flexible, searchable schedule for students; better tracking of demographics and eligibility; and better tracking of hours tutored.

Scope

The tutoring database will manage data for the tutoring program at the college. It will track available tutors and the courses they can tutor. It will also track each tutor's tutoring schedule. The database will store demographic information for students who register for tutoring. This information will be private and used only to generate general reports that include no personal information. Students who have registered will be able to sign up for available tutoring sessions for courses in which they are enrolled. The database will track whether students attended their scheduled sessions. It will also track student requests for tutoring in additional courses and subjects.

Constraints

The database can be used to get the hours worked for each tutor, but it will not process pay or provide any payroll information. The database will not validate student information against the school's registration database.

Objectives

- Streamline the process by which the tutors enter their schedules and students sign up for them
- Improve tracking of demographic data of students using the tutoring program
- Improve tracking of tutors' hours and students' use of tutoring sessions
- Track student requests for additional tutoring

Tasks and Timeline

1. **Gathering Data:** This task will consist in a number of interviews, questionnaires, and observations. Time allotted: 3 weeks.

Deliverable: A list of scheduled interviews and observations and, text of the questionnaires.

2. **Analyzing Data:** The data gathered will be analyzed to determine business rules and preliminary data modeling. Time allotted: 2 weeks.

Deliverable: List of business rules—their basic entities and attributes—to be reviewed.

3. **Normalization:** The data model will be completed with entities and relationships normalized. Time allotted: 1 week.

Deliverables: Entity relation diagram for review.

4. **Building the Physical Database:** The data model will be translated to the RDBMS. Tables containing columns with specific data types and relational and other constraints created. Time allotted: 3 days.

Deliverables: The schema of the database for review.

5. **Testing and Security:** Sample data will be entered and each of the business rules and requirements will be tested. General database security and security related to business rules will also be tested. Time allotted: 3 weeks.

Deliverables: Documented test results.

6. **Database Completion and Installation:** Final changes and corrections are made. Sample data will be removed, and the database installed on a server. Final testing for server access and connections. Time allotted: 2 weeks.

Deliverables: The working database.

Total time between beginning and end of the project: 11 weeks, 3 days.

DOCUMENTATION

Documentation is a lot like flossing: Nobody likes to do it, and far more claim to do it than actually do. Developers want to work on their plan. The last thing they want to do, generally, is to take time out and describe what they are developing and how they are going about it. And yet, like flossing, few things are as important to a healthy database enterprise.

Imagine you have been hired to work as a database administrator for some company. They have a large and complex database, but the former administrator, who was also the developer, left no documentation. To do your job properly, you need to understand what each object in the database is meant to do. You also need to know what it is supposed to do and how data is processed. Managers expect you to be able to provide them with the data they need when they need it. Some pieces probably make sense right away, but several pieces remain obscure. You try to ask people about them, but managers are not database designers and, generally, they don't have a clue. Many of the people who were involved in the creation of the database have moved on, and it is difficult to get a clear sense of the original intentions or purpose of the database. Eventually you may solve the problems, but you will have spent countless hours in investigation, hours that could have been saved by a little documentation.

Documentation is one of the most important and one of the most neglected aspects of any database project. When you look at a database built by someone else, or even one that you may have made some time ago, it is often difficult to see why certain decisions were made, why the tables are the way they are, and why certain columns were included or left out. Without documentation, it can take a great deal of research and guesswork to understand the database. You may never understand all of its original logic.

So what does it mean to document a database? There are really two main aspects that need to be documented: the structure of the database itself and the process by which the database was developed.

Documenting the existing structure of the database includes describing the tables, the columns and their data types, and the relations between tables and any other database objects and constraints. This kind of documentation is often called a "data dictionary." Anyone can use this dictionary to look up any table and find out what columns and key fields it contains. He or she can also look up a column and determine its data type and what constraints, if any, were placed on the column. This is important information for anyone who needs to maintain the database or for application developers who wish to build software based on the database.

Documenting the process of developing the database should include recording the original intent of the database, the problems that it was meant to solve, the business rules to which it must conform, and important decisions that were made throughout the process. This information is essential to anyone who needs to maintain or modify the database. Such an individual needs to first understand why the database is as it is. Then he or she needs to understand how his or her changes will affect the original purposes of the database.

As part of the development process, you should keep one or more notebooks in which you put all the documents and notes related to the project. The first thing you should add is the statement of work. The statement of work is one of the first and most important pieces of documentation. The history section captures the original reasons for developing the database. The scope and objectives provide insight into the specific tasks the database was intended to perform.

In the following scenario sections and in the rest of the book, there will be "to do" items that are labeled "Documentation" to help you record your development process.

Things We Have Done

In this chapter we have

- identified a situation in which a database could prove valuable
- reviewed briefly the history of databases
- identified some of the components of relational databases such as entities and key fields
- observed an interview to gather general information about a database
- broken the general information into major topics
- used the major topics to develop a statement of work for the database

Vocabulary

Match the definitions to the vocabulary words

1. Attribute	—	a. A type of database that uses "relations," tables, to store and relate tables.
2. Foreign key	—	b. The process of organizing data into tables or entities and then determining the relations among them.
3. Statement of work	—	c. The language relational databases used to create their objects and to modify and retrieve data.
4. Primary key	—	d. These files have some sort of character separating columns of data. The delimiter is often a comma or tab, but it can be any non-alphanumeric character.
5. Data integrity	—	e. Files where the length in characters of each column is the same.
6. Redundancy	—	f. Refers to the accuracy and the correctness of the data in the database.
7. Delimited files	—	g. Refers to storing the same data in more than one place in the database.
8. Relational database	—	h. This key uniquely identifies each row in the table.
9. Entity	—	i. This key is the primary key repeated in another table to create a link between the tables.
10. Relational design	—	j. A short statement of one or more paragraphs that says in clear, but general, terms what the project will do.
11. SQL	—	k. Something that the database is concerned with, about which data can be stored.
12. Constraints	—	l. Things that define aspects of entities.
13. Fixed-width files	—	m. Limits on what the database will do.
14. Statement of scope	—	n. A document including the scope, objectives, and timeline for a given project.