Chapter Introduction

Learning Objectives

- Examine JC Consulting (JCC), the company used for many of the examples throughout the text
- Define basic database terminology
- Describe database management systems (DBMSs)
- Explain the advantages and key factors for a healthy relational database system
- Prepare for a career in database administration
- Review Pitt Fitness, a company used in a case that appears at the end of each module
- Review Sports Physical Therapy, a company used in another case that appears at the end of each module

Introduction

In this module, you will examine the requirements of JC Consulting (JCC), a company that will be used in many examples in this text. You will learn how JCC initially stored its data, what problems employees encountered with that storage method, and why management decided to employ a database management system (DBMS). You will also study the basic terminology and concepts of relational databases, database management systems, and big data. You will learn the advantages and key factors of a properly designed relational database. Finally, you will examine the database requirements for Pitt Fitness and Sports Physical Therapy, the companies featured in the cases that appear at the end of each module.

Chapter 1: Introduction to Database Management Chapter Introduction
Book Title: Concepts of Database Management
Printed By: Michael Ammerman (mammerman0005@kctcs.edu)
© 2020 Cengage Learning, Cengage Learning

Chapter 1: Introduction to Database Management: 1-1 JC Consulting Company Background
Book Title: Concepts of Database Management
Printed By: Michael Ammerman (mammerman0005@kctcs.edu)

1-1 JC Consulting Company Background

JC Consulting (JCC) is a digital development and consulting business. The founder, Jacqueline Cabrero, started the business in the mid-1990s when the Internet became publicly available. Jacqueline grew the business from a sole proprietorship that built static webpages for small businesses to a firm with more than 20 employees. JCC's services range from building websites and web apps to back-end database conversions and programming.

Initially, Jacqueline kept track of her clients and project bids in a spreadsheet. As the company grew, she used a homegrown project estimator program to bid new projects. Jacqueline has now determined that the company's recent growth means it is no longer feasible to use those programs to maintain its data.

What led JCC to this decision? One of the company's spreadsheets, shown in Figure 1-1, displays project estimates, and illustrates JCC's problems with the spreadsheet approach. For each estimate, the spreadsheet displays the number and name of the client, the project estimate number and date, the task ID, a description of the task, and a quoted price for that task. Tri-Lakes Realtors received two different project estimates (ProjectIDs 1 and 31). In the first project estimate, Tri-Lakes Realtors needed general help to establish online goals. In the second estimate, the agency needed help with relational database design and data conversion. The result was seven lines in the spreadsheet, two project estimate numbers, and several task IDs.

Figure 1-1

Project Estimates Spreadsheet

Data redundancy (The duplication of data, or the storing of the same data in more than one place.) is one problem that employees have with the project estimates spreadsheet. Data redundancy, sometimes shortened to **redundancy** (The duplication of data, or the storing of the same data in more than one place.) , is the duplication of data, or the storing of the same data in more than one place. In the project estimates spreadsheet, redundancy occurs in the ClientName column because the name of a client is listed for each line item in each project estimate. Redundancy also occurs in other columns, such as in the ProjectID column when a project has more than one task, or in the TaskID and Description columns when two projects contain the same tasks.

Q&A

1-1

What problems does redundancy cause?

Difficulty accessing related data is another problem that employees at JCC encounter with their spreadsheets. For example, if you want to see a client's address, you must open and search another spreadsheet that contains this data because the client's address is not currently stored in the project estimates spreadsheet.

Spreadsheets also have limited security features to protect data from being accessed by unauthorized users. In addition, a spreadsheet's data-sharing features prevent multiple employees from updating data in one spreadsheet at the same time. Finally, if JCC estimates continue to increase at their planned rate, spreadsheets have inherent size

limitations that will eventually force the company to split the project estimates into multiple spreadsheets. Splitting the project estimates into more than one spreadsheet would create further redundancy, data organization, and reporting problems. For these reasons, JCC decided to replace the estimating spreadsheet with a **database** (A collection of data organized in a manner that allows access, retrieval, and use of that data.) , a collection of data organized in a manner that allows access, retrieval, and use of that data.

After making the decision, management has determined that JCC must maintain the following information about its employees, clients, tasks, and project estimates:

- For employees: Employee ID, last name, first name, hire date, title, and salary
- For clients: Client ID, name, address, and government status
- For projects: Project ID, start date, task IDs, task descriptions, costs, project notes, and task notes

Figure 1-2 shows a sample project estimate.

Figure 1-2

Sample Project Estimate

- The top of the estimate contains the company name, JC Consulting, the words “Project Estimate,” and company contact information.
- The body of the estimate contains the name of the client for which the project estimate has been created, a brief description of the project, the project ID, an estimated project start date, and one or more line items. Each line item contains a task description and the estimated price for that task.
- The bottom of the estimate contains the total estimated price for the project.

Internally, JCC also must store the following items for each client’s estimate:

- For each project estimate, JCC must store the client's address as well as the employee assigned as the project leader for the project.
- For each task line item, JCC not only stores the project ID but also the task ID, the estimated completion date for that task, and task notes. If the task is considered complex or risky, a factor is applied to increase the estimate. The task description and task category (coding, meeting, planning, and so forth) are also stored for each task.
- The overall project estimate total is not stored. Instead, the total is calculated whenever a project estimate is printed or displayed on the screen.

The problem facing JCC is common to many businesses and individuals that need to store and retrieve data in an efficient and organized way. JCC is interested in several areas of information such as employees, clients, estimates, and tasks. A school is interested in students, faculty, and classes; a real estate agency is interested in clients, houses, and agents; a distributor is interested in customers, orders, and inventory; and a car dealership is interested in clients, vehicles, and manufacturers.

The difficult question is not how to manage single categories or lists of information, but how to manage the lists of data and the relationships between the lists. For example, besides being interested in client and project estimate information, JCC also wants to know which clients have received more than one project estimate. The company wants to know which employees are assigned as the lead to which projects and which tasks are most commonly added to which projects.

Likewise, a school is not only interested in students and classes but also which students are enrolled in which classes. A real estate agency is not only interested in their lists of agents and homes for sale; they also want to know which agents are listing or selling the most homes. A distributor wants to know which customers are ordering specific inventory items, and a car dealership not only wants to know about their customer base and car inventory but also which customers are buying multiple cars over time.

1-2 Selecting a Database Solution

After studying the alternatives to using spreadsheet software, JCC decided to switch to a relational database system. A **relational database** (A collection of related tables of data.) is a structure that contains data about many categories of information as well as the relationships between those categories. The JCC database, for example, will contain information about employees, clients, project estimates, and tasks. It also will provide facts that relate employees to the projects they manage, clients to their project estimates, and the project estimates to the tasks that are contained within each project.

With a relational database, JCC will be able to retrieve a particular project estimate and identify which client and tasks belong to that estimate. In addition, employees can start with a client and find all project estimates, including the individual tasks within the project that have been prepared for that client. Using a relational database, JCC can use the data to produce a variety of regular periodic or ad hoc reports to summarize and analyze the data in an endless number of ways.

Chapter 1: Introduction to Database Management: 1-2 Selecting a Database Solution
Book Title: Concepts of Database Management
Printed By: Michael Ammerman (mammerman0005@kctcs.edu)
© 2020 Cengage Learning, Cengage Learning

Chapter 1: Introduction to Database Management: 1-2a Defining Database Terminology
Book Title: Concepts of Database Management
Printed By: Michael Ammerman (mammerman0005@kctcs.edu)

1-2a Defining Database Terminology

Some terms and concepts in the database environment are important to know. The terms *entity*, *attribute*, and *relationship* are fundamental when discussing databases. An **entity** (A person, place, event, item, or other transaction for which you want to store and process data. Also called a *table*.) is a person, place, event, item, or other transaction for which you want to store and process data. The entities of interest to JCC, for example, are employees, clients, project estimates, and tasks. Entities are represented by a **table** (A person, place, event, item, or other transaction for which you want to store and process data. Also called an *entity*.) of data in relational database systems.

An **attribute** (A characteristic or property of an entity; also called a *column* or a *field*.) is a characteristic or property of an entity. For the entity *employee*, for example, attributes might include such characteristics as first and last name, employee number, and date of hire. For JCC, the attributes of interest for the *client* entity include client name, street, city, state, zip code, and whether the client is a government body. An attribute is also called a **field** (A characteristic or property of an entity; also called an *attribute* or a *column*.) or **column** (A characteristic or property of an entity; also called an *attribute* or a *field*.) in many database systems.

Figure 1-3 shows two entities, Clients and Projects, along with the attributes for each entity. The Clients entity has seven attributes: ClientID, ClientName, Street, City, State, Zip, and Government (whether the client is any type of government institution). Attributes are similar to columns in a spreadsheet. The Projects entity (which represents project estimates) has five attributes: ProjectID, ProjectStartDate, ClientID, EmployeeID, and ProjectNotes. Entity (table) names and attribute (field) names should be easy to understand, concise, indicative of their content, and contain no spaces or other special characters.

Figure 1-3

Entities and Attributes

A **relationship** (An association between entities.) is an association between entities. For example, there is an association between clients and projects. A client is *related to* all of its projects, and a project is *related to* its client.

This relationship is called a **one-to-many relationship** (A relationship between two entities in which each occurrence of the first entity is related to many occurrences of the second entity, and each occurrence of the second entity is related to at most one occurrence of the first entity.) because each client may be associated with *many* projects, but each project is associated with only *one* client. In this type of relationship, the word *many* is used differently from everyday English because it does not always indicate a large number. In this context, the term *many* means that a client can be associated with *any* number of projects. That is, a given client can be associated with zero, one, or more projects.

A one-to-many relationship often is represented visually as shown in Figure 1-4. In such a diagram, entities and attributes are represented in precisely the same way as they are shown in Figure 1-3. A line connecting the entities represents the relationship. The *one* entity of the relationship (in this case, Clients) does not have an arrow on its end of the line, and the *many* entity in the relationship (in this case, Projects) is indicated by a single-headed arrow.

Figure 1-4

One-to-Many Relationship

Q&A

1-2

What happens when the relationship between two entities is best defined as “many-to-many” because one record in one entity relates to many records in the other entity and vice versa? For example, at a college, one student may be related to many classes, and one class is also related to many students.

Chapter 1: Introduction to Database Management: 1-3 Storing Data
Book Title: Concepts of Database Management
Printed By: Michael Ammerman (mammerman0005@kctcs.edu)

1-3 Storing Data

A spreadsheet that is used to store data, often called a **data file** (A file that stores data about a single entity in one table or list.) , typically stores data as one large table. Data stored this way is also referred to as a **flat file** (A file that stores data in lists that have no relationship to other lists.) because lists in a spreadsheet have no relationships with other lists. A relational database, however, not only stores information about multiple entities in multiple tables but also identifies the relationships between those tables.

For example, in addition to storing information about projects and clients, the JCC database will hold information relating clients to the various project estimates that were created for that client, which employee is assigned as the project leader for that particular project, and more. A relational database can store information about multiple types of entities, the attributes of those entities, and the relationships among the entities.

How does a relational database handle these entities, attributes, and relationships among entities? Entities and attributes are fairly straightforward. Each entity has its own table. The JCC database, for example, will have one table for employees, one table for clients, one table for the project estimates, and so on. The attributes of an entity become the columns in the table. Within each table, a **row** (A collection of related fields. Also called a *record* or *tuple*.) of data corresponds to one record. A **record** (A collection of related fields that describe one item in a table; can be thought of as a row in a table. Also called a *row* or *tuple*.) is a group of fields (attributes) that describe one item in the table (entity).

What about relationships between entities? At JCC, there is a one-to-many relationship between clients and projects. But how is this relationship established in a relational database system? It is handled by using a common field in the two tables to tie the related records from each table together. Consider Figure 1-4. The ClientID column in the Clients table and the ClientID column in the Projects table are used to build the relationship between clients and projects. Given a particular ClientID, you can use these columns to determine all the projects that have been estimated for that client; given a ProjectID, you can use the ClientID columns to find the client for which that project estimate was created.

How will JCC store its data via tables in a database? Figure 1-5 shows sample data for JCC.

Figure 1-5

Sample Data for JCC

Employees

EmployeeID	LastName	FirstName	HireDate	Title	Salary
19	Kohn	Ali	01-Jan-20	Project Leader	$5,000.00
22	Kaplan	Franco	01-Feb-20	Programmer	$5,500.00
35	Prohm	Nada	29-Feb-20	Customer Support Specialist	$4,000.00

EmployeeID	LastName	FirstName	HireDate	Title	Salary
47	Alvarez	Benito	31-Mar-20	Front End Developer	$5,200.00
51	Shields	Simone	30-Apr-20	Network Specialist	$7,000.00
52	Novak	Stefan	01-Jan-19	Project Leader	$8,000.00
53	Anad	Sergei	01-Jan-19	Front End Developer	$5,300.00
54	Allen	Sasha	01-Jan-19	Programmer	$7,000.00
55	Winter	Wendy	31-Dec-20	Front End Developer	$4,300.00
56	Reddy	Kamal	01-Sep-19	Programmer	$6,200.00
57	Yang	Tam	30-Apr-21	Front End Developer	$5,000.00
58	Young	Solomon	01-Jan-19	Programmer	$5,500.00
59	Santana	Carmen	01-Jan-19	Front End Developer	$4,800.00
60	Lu	Chang	01-Mar-19	Database Developer	$7,900.00
61	Smirnov	Tovah	01-Oct-19	Programmer	$6,000.00
62	Turner	Jake	31-Mar-21	Database Developer	$7,800.00
63	Geller	Nathan	01-Jan-19	Project Leader	$8,100.00
64	Lopez	Miguel	01-Jan-19	Programmer	$6,200.00
65	Garcia	Hector	01-Apr-23	UI Designer	$7,000.00
66	Roth	Elena	31-Oct-20	Network Specialist	$7,000.00
67	Horvat	Nigel	30-Apr-24	UI Designer	$6,300.00

Clients

ClientID	ClientName	Street	Zip	Government
1	Tri-Lakes Realtors	135 E Jefferson St	02447	FALSE
2	Project Lead The Way	762 Saratoga Blvd	02446	TRUE
3	Midstates Auto Auction	9787 S Campbell Ln	01355	FALSE
4	Bretz & Hanna Law Firm	8101 N Olive Dr	01431	FALSE
5	Aspire Associates	5673 South Ave	01431	FALSE
6	Bounteous	9898 Ohio Ave	02770	FALSE

ClientID	ClientName	Street	Zip	Government
7	Netsmart Solutions	4091 Brentwood Ln	01354	FALSE
8	Loren Group	9565 Ridge Rd	02466	FALSE
9	Associated Grocers	231 Tecumsa Rd	02532	FALSE
10	Jobot Developers	1368 E 1000 St	02330	FALSE
11	Harper State Bank	1865 Forrest Dr	01571	FALSE
12	MarketPoint Sales	826 Hosta St	01983	FALSE
13	SecureCom Wireless	5280 Industrial Dr	01852	FALSE
14	The HELPCard	840 Boonville Ave	02466	TRUE
15	Jillian Henry & Associates	815 E California St	02113	FALSE
16	Pediatric Group	4940 W Farm Rd	02113	FALSE
17	SkyFactor	1736 Sunshine Dr	02726	FALSE
18	NuCamp	2500 E Kearny St	01431	FALSE
19	Wu Electric	5520 S Michigan	02447	FALSE
20	Juxly Engineering	4238 Rumsfield Rd	02148	FALSE
21	Carta Training	2445 N Airport Dr	02446	FALSE

Projects

ProjectID	ProjectStartDate	ClientID	EmployeeID	
1	06-Feb-19	1	52	Client wants digital solutions to help rebrand company name to emphasize commercial real estate.
2	07-Feb-19	2	63	Client needs help converting, organizing, and managing various sources/formats of donor and donation data.
3	11-Mar-19	3	52	Client wants to establish SEO goals.
4	10-Apr-20	4	52	Client wants to set up an internal server as well as help with a domain name.
7	02-Sep-19	2	63	Client has used the database for several months and now needs new report
8	06-Jan-20	3	52	Develop and implement website SEO strategy.

ProjectID	ProjectStartDate	ClientID	EmployeeID	
9	10-Feb-20	6	63	Needs help to manage and organize internal data.
10	31-Mar-21	7	19	Develop new website content.
11	30-Apr-20	9	19	Client needs internal database to manage personnel.
13	30-Nov-20	10	64	Client needs subcontracting help installing a new database for a WordPress site.
14	09-Dec-20	15	19	Client needs new functionality for current JavaScript application.
15	21-Dec-20	14	19	Client needs new functionality for current Ruby/Rails application.
16	04-Jan-21	11	52	Client needs help with server security.
17	15-Feb-21	12	52	Current online sales solution is unreliable.
18	14-Apr-21	6	63	Client needs internal database to manage inventory.
19	04-Jun-21	13	52	Client needs new functionality for current C# / ASP.NET application.
20	30-Jul-21	1	63	Client needs full website reskin.
21	31-Aug-21	16	19	Client needs help with data analytics.
22	30-Sep-21	20	19	Client needs an online reference database
23	12-Nov-21	18	63	Client needs new blog and current pages updated to include responsive web design principles for mobile devices.

ProjectLineItems

ProjectLineItemID	ProjectID	TaskID	TaskDate	Quantity	Factor	ProjectLineItemNotes

ProjectLineItemID	ProjectID	TaskID	TaskDate	Quantity	Factor	ProjectLineItemNotes
1	1	MEET00	06-Feb-19	1	1.00	
2	1	PLAN01	06-Feb-19	1	1.00	
4	2	MEET00	07-Feb-19	1	1.00	
5	2	PLAN01	07-Feb-19	1	1.00	
6	2	DB01	15-Mar-19	1	1.30	Data is stored in multiple spreadsheets.
7	2	DB02	15-Apr-19	20	1.30	Data is not consistent between spreadsheets.
8	3	MEET00	11-Mar-19	1	1.00	
9	3	PLAN01	11-Mar-19	1	1.20	Owner is difficult to pin down.
10	4	MEET00	10-Apr-20	1	1.00	
11	4	PLAN01	10-Apr-20	1	1.20	Two principal attorneys must agree.
12	4	SERV01	11-May-20	1	1.00	
13	4	SERV02	10-Jun-20	1	1.30	Security is a paramount issue.
17	11	MEET00	30-Apr-20	1	1.00	
18	11	PLAN01	30-Apr-20	1	1.00	
19	9	MEET00	10-Feb-20	1	1.00	
20	9	PLAN01	10-Feb-20	1	1.00	
25	9	PLAN10	17-Feb-20	1	1.00	
26	18	MEET00	14-Apr-21	1	1.00	
27	20	MEET00	30-Jul-21	1	1.00	
28	20	PLAN01	30-Jul-21	1	1.00	
29	20	PLAN02	30-Jul-21	1	1.00	

TaskMasterList

TaskID	Description	CategoryID	Per	Estimate
CODE01	Code PHP	Coding	Hour	$150.00
CODE02	Code C# in ASP.NET	Coding	Hour	$150.00
CODE03	Code Ruby on Rails	Coding	Hour	$150.00
CODE04	Code SQL	Coding	Hour	$150.00
CODE05	Code HTML	Coding	Hour	$100.00
CODE06	Code CSS	Coding	Hour	$100.00
CODE07	Code JavaScript	Coding	Hour	$125.00
CODE08	Perform analytics	Coding	Hour	$100.00
CODE09	Select technology stack	Coding	Hour	$200.00
CODE10	Apply SEO	Coding	Hour	$125.00
CODE12	Create prototype	Coding	Hour	$150.00
CODE13	Code WordPress	Coding	Hour	$100.00
CODE14	Code Python	Coding	Hour	$150.00
CODE15	Create shopping cart	Coding	Hour	$125.00
CODE16	Code other	Coding	Hour	$150.00
DB01	Design relational database	Database	Project	$1,000.00
DB02	Convert data	Database	Hour	$125.00
DB03	Install MySQL database	Database	Project	$500.00
DB04	Install SQL Server database	Database	Project	$500.00
DB05	Install Access database	Database	Project	$400.00
MEET00	Initial customer meeting	Meeting	Project	$0.00

In the Employees table, each employee has a unique EmployeeID number in the first column. The name of the employee with the EmployeeID value of 19 in the first record is Ali Kohn. His hire date was 01-Jan-20, his title is Project Leader, and his monthly salary is $5,000.00.

The Clients table contains one record for each client, which is uniquely identified by a ClientID number in the first column. The client name, street, zip, and whether the client is a governmental entity are also stored for each record.

In the Projects table, a unique ProjectID number for each project is positioned in the first column. The project start date and project notes are also recorded in fields named ProjectStartDate and ProjectNotes. The ClientID field contains a number that connects the Projects table with the Clients table. In the first record for ProjectID 1, the ClientID value is also 1, which connects with Tri-Lakes Realtors in the Clients table. The EmployeeID value of 52 connects with Stefan Novak in the Employees table.

In the table named ProjectLineItems, each record represents one task for each project. The ProjectID value connects each record to a specific project in the Projects table. Note that the first two records contain a ProjectID value of 1 connecting them with the first project in the

Projects table, which in turn connects them with the Tri-Lakes Realtors record in the Clients table. The TaskID column connects each line item with a record in the TaskMasterList table that further describes that task. The ProjectLineItems table also contains fields named TaskDate, Quantity, Factor, and ProjectLineItemNotes, which further describe each task for that project. The Quantity field is used to identify the estimated hours for the hourly tasks. The Factor field is a multiplier that represents additional risk or complexity. For example, $1.1 = 10\%$ increase in perceived complexity or risk for that task. Both the Quantity and Factor fields are used to calculate the price for that task.

The TaskMasterList table uniquely identifies the different tasks that may appear on a project estimate with the TaskID field, and further describes each task with the Description, CategoryID, Per (per hour or per project), and Estimate fields. The Estimate field contains the dollar amount for that task. It is multiplied by the Quantity and Factor fields in the ProjectLineItems table to calculate the total estimated cost for that line item.

The table named ProjectLineItems might seem strange at first glance. Why do you need a separate table for the project line items? Couldn't the project line items be included in the Projects table? The Projects table *could* be structured as shown in Figure 1-6. Notice that this table contains the same projects and line items as those shown in Figure 1-5, with the same fields and data. However, the TaskID, TaskDate, Quantity, Factor, and ProjectLineItemNotes fields contain multiple entries.

Figure 1-6

Alternative Projects Table Structure

ProjectID	ProjectStartDate	ClientID	EmployeeID	ProjectNotes	TaskID	TaskDate	Quantity	Factor	Project
1	06-Feb-19	1	52	Client wants digital solutions to emphasize commercial real estate.	MEET00 PLAN01	06-Feb-19 06-Feb-19	1 1	1.00 1.00	

ProjectID	ProjectStartDate	ClientID	EmployeeID	ProjectNotes	TaskID	TaskDate	Quantity	Factor	Project
2	07-Feb-19	10	63	Client needs help converting, organizing, and managing donor and donation data.	MEET00	07-Feb-19	1	1.00	Data is
					PLAN01		1	1.00	multiple
					DB01	07-Feb-19	1	1.30	Data is
					DB02		20	1.30	betwee spreads
					CODE04	15-Mar-19	4	1.00	Code S
					TEST01	15-Apr-19	8	1.00	queries
					TEST02		8	1.00	
					MEET01	15-May-19	2	1.00	
					SUPP03	03-Jun-19	8	1.00	
						03-Jun-19			
						03-Jun-19			
						03-Jun-19			
3	11-Mar-19	3	52	Client wants to establish SEO goals.	MEET00	11-Mar-19	1	1.00	Owner i
					PLAN01		1	1.20	down.
						11-Mar-19			
4	10-Apr-20	4	52	Client wants to set up an internal server as well as help with a domain name.	MEET00	10-Apr-20	1	1.00	Two pri
					PLAN01		1	1.20	must ag
					SERV01	10-Apr-20	1	1.00	Security
					SERV02		1	1.30	paramo
					TEST01	11-May-20	16	1.00	
					TEST02	10-Jun-20	16	1.00	
					SUPP03		4	1.00	
						15-Jun-20			
						15-Jun-20			
						15-Jun-20			

Q&A

1-3

How is the information in Figure 1-5 represented in Figure 1-6?

Q&A

1-4

Why does ProjectID 2 have such a large row in Figure 1-6?

Figure 1-5 shows a single entry in each field of the ProjectLineItems table. In Figure 1-6, the fields that describe tasks contain multiple entries such as the TaskID, TaskDate, Quantity, Factor, and ProjectLineItemNotes. For example, ProjectID 1 consists of two tasks, and therefore two entries are placed in the TaskID, TaskDate, Quantity, and Factor fields because those fields describe the two tasks for that project. Other projects contain many more tasks and would have many more entries in those fields.

In general, tables that contain more than one piece of information per attribute (column or field) create several problems that can be eliminated with a proper relational database design. The following are some warning signs that your entities, attributes, and relationships are not properly designed:

- You need to enter more than one value in a particular field (see Figure 1-6).
- You are asked to enter two or more pieces of information in a field. For example, using one field to enter both first and last names means you cannot quickly and easily search, sort, and filter on either part of a person's name.
- You are asked to enter both values and units of measure in the same field. Entering numbers and text in the same field generally prevents you from calculating on the numeric part of the data.
- You find yourself adding new columns to handle multiple values for the same type of data. For example, to track employee salaries over time, you wouldn't want to create additional attributes in the Employees table with names such as Salary1, Salary2, and Salary3. A better approach would be to create a Salaries table and relate it to the Employees table. One employee record would be related to many records in the Salaries table. When an employee earned a salary increase, you would add a new record in the Salaries table for that employee with the new salary value as well as the new salary effective date.

In the tables shown in Figure 1-5, each field is named concisely yet clearly. Each field contains one and only one piece of information, and fields have been created to properly relate the tables in one-to-many relationships. To test your understanding of the JCC data, use the data shown in Figure 1-5 to answer the following questions.

- **Question:** What is the client name for ClientID 10?

  Answer: Jobot Developers

- **Question:** What ProjectIDs and ProjectNotes are connected with ClientID 10?

Answer: ClientID 10, Jobot Developers, is connected with ProjectID 13, “Client needs subcontracting help installing a new database for a WordPress site.”

- **Question:** What are all the TaskIDs that appear for ProjectID 1?

  Answer: MEET00 and PLAN01

- **Question:** What fields are needed to calculate a line item cost estimate for a task such as CODE04 (Code SQL) that requires several hours of effort?

  Answer: The line item cost value is not physically stored in the tables, but calculated using three fields: the Quantity and Factor fields in the ProjectLineItems table are multiplied by the Estimate field in the TaskMasterList table.

- **Question:** What is the Per field?

  Answer: The Per field in the TaskMasterList table identifies the unit of measure for each task. If the task is an hourly task, the value is Hour. If the task is completed once per project, the value is Project.

- **Question:** How is the Per field used in the calculation for the line item cost estimate?

  Answer: The Per field is not directly used in the line item cost estimate. (See the previous question and answer about calculating a line item cost estimate for a task.) If the value in the Per field is Project, the Quantity field value is always 1, indicating that the task happens only once per project. If the value in the Per field is Hour, the Quantity field represents the number of hours estimated for that task.

- **Question:** What is the Factor field?

  Answer: The Factor field is used to increase the cost estimate for a line item based on perceived complexity or risk. By default, the Factor field is 1, but an entry of 1.1 would increase the total line item cost by 10%.

- **Question:** How could you reduce redundancy in the City, State, and Zip fields of the Clients table?

  Answer: You could separate those fields into their own table, perhaps called Zips, and relate the Zips table to the Clients table using a common Zip field. One record in the Zips table would then be related to many records in the Clients table. Doing this would eliminate redundant City and State values in the Clients table.

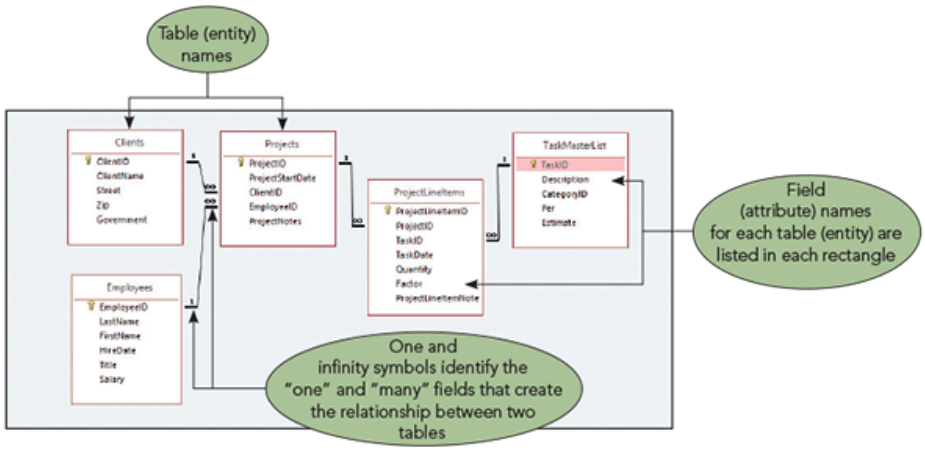
- **Question:** How could you ensure consistency in the CategoryID field of the TaskMasterList table?

  Answer: You could separate that field into its own table, perhaps called Categories, which stores the individual CategoryID values, one per record such as Coding, Database, and Meeting. One record in the Categories table would be related to many records in the TaskMasterList table. Doing this would prevent the user from entering inconsistent CategoryID values in the TaskMasterList table.

Many database administrators and computer science professionals use a visual way to represent and analyze a database called an **entity-relationship (E-R) diagram** (A graphic model for database design in which entities are represented as rectangles and relationships are represented as either arrows or diamonds connected to the entities they relate. Also called an *ERD*.) (sometimes referred to as an **ERD**). In an E-R diagram, rectangles represent entities and display their attributes; lines represent relationships between connected entities. The E-R diagram for the JCC database appears in Figure 1-7.

Figure 1-7

E-R Diagram for the JCC Database



Each of the five entities in the JCC database appears as a rectangle in the E-R diagram shown in Figure 1-7. The name of each entity appears at the top of the rectangle. The attributes (fields or columns) for each entity appear within the rectangle. Because the Clients and Projects entities have a one-to-many relationship, a line connects these two entities; similarly, a line connects the Employees and Projects entities, the Projects and ProjectLineItems entities, and the TaskMasterList and ProjectLineItems entities. The number 1 indicates the “one” part of the one-to-many relationship between two entities. The infinity symbol (∞) at the end of a line indicates the “many” part of the one-to-many relationship between two entities. Some E-R diagrams represent the relationship lines in a different way. You will learn more about E-R diagrams in module 6.

Chapter 1: Introduction to Database Management: 1-4 Identifying Database Management Systems
Book Title: Concepts of Database Management
Printed By: Michael Ammerman (mammerman0005@kctcs.edu)

1-4 Identifying Database Management Systems

A **database management system (DBMS)** (A program through which users interact with the data stored in a database.) is a program through which users interact with the data stored in a database. Access works well for JC Consulting because only a handful of trusted users will be simultaneously using the application and they are all located in the same building. Fortunately, the relational database design concepts by which you determine and create entities (tables), attributes (fields), and relationships are similar between all relational database management systems, but Access is mostly limited to those situations where a small number of simultaneous trusted users are located in the same physical area.

If the requirements for your relational database application exceed these basic requirements, other relational database management systems that can handle a larger number of users and data, which have more security and application development features, and which can be used across wide area intranets or even the Internet are available as described in Figure 1-8.

Figure 1-8

Popular Relational Database Management System Software

Name	Company	Website
Oracle	Oracle Corporation	oracle.com/database/
MySQL	Oracle/Open source	mysql.com/
SQL Server	Microsoft	microsoft.com/en-us/sql-server/default.aspx
PostgreSQL	Open source	postgresql.org/
Db2	IBM	ibm.com/products/db2-database
SQLite	Open source	sqlite.org
MariaDB	Open source	mariadb.org/
Access	Microsoft	products.office.com/en-us/access

A **software stack** (A group of programs and technologies that commonly work together to build a digital solution. Also called a *solution stack* or *technology stack*.) (also called a solution stack or technology stack) groups several software components that are commonly used to build a new application. A software stack often includes a programming language such as C#, Java, PHP, Ruby, or Python; a framework that provides tools to help write code and applications in a standardized and organized way; and a database management system to organize and store the raw data. Some stacks include operating systems, web servers, and other hardware and software components.

Theoretically, the relational database management programs in Figure 1-8 can be interchanged within most solution stacks given they all conceptually structure and store the entities, attributes, and relationships using the same relational database concepts. They do, however, have slightly different implementations of these concepts, and they each have different size limitations, security features, and application development tools. Over time, specific software stacks with certain relational database systems have become favorites for reasons that include business incentives and mergers, performance and reliability, available documentation, and personal developer preferences. A few popular software stacks are listed in Figure 1-9.

Figure 1-9

Popular Software Stacks

Stack Name	Operating System	Web Server	Database	Programming Language	Framework(s)
LAMP	Linux	Apache	MySQL	PHP	Laravel, Zend, Drupal, Joomla, and others
XAMP	X (for “any” operating system)	Apache	MariaDB	PHP	
WINS (Microsoft)	Windows Server	Internet Information Services	SQL Server	C#	ASP.NET
Ruby on Rails			SQLite or PostgreSQL	Ruby	Rails

Q&A

1-5

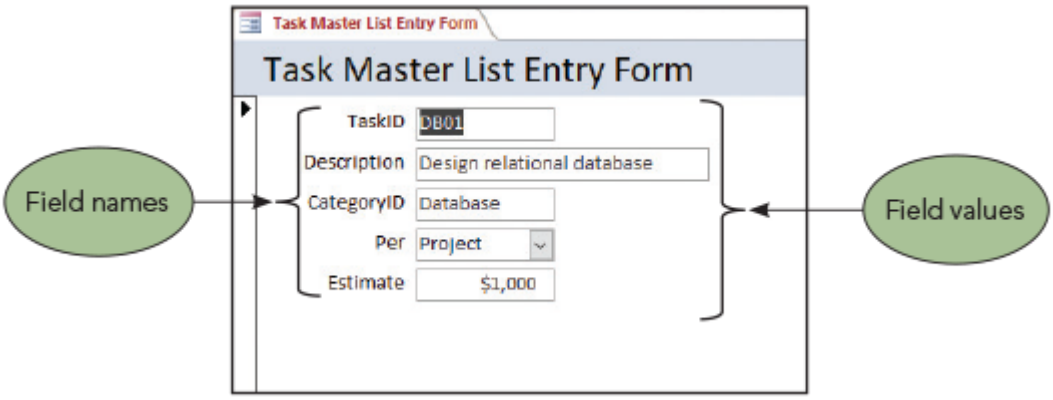
Which DBMS should JC Consulting use to create and maintain its database?

The **database design** (The process of creating the entities, attributes, and relationships between tables of data.) phase is the process of creating the entities, attributes, and relationships between the tables of data shown in Figure 1-5. When that is completed, the database developer creates **forms** (A screen object used to maintain and view data from a database.) , screen objects used to maintain and view data from a database. Employees then use these forms to find, enter, and edit data.

The form that employees use to enter the tasks that may appear on each project estimate is shown in Figure 1-10. Employees can use this form to enter a new task; to view, change, or delete an existing task; or to print the information for a task. Using the form development tools provided by Access, you can create the form without having programming knowledge.

Figure 1-10

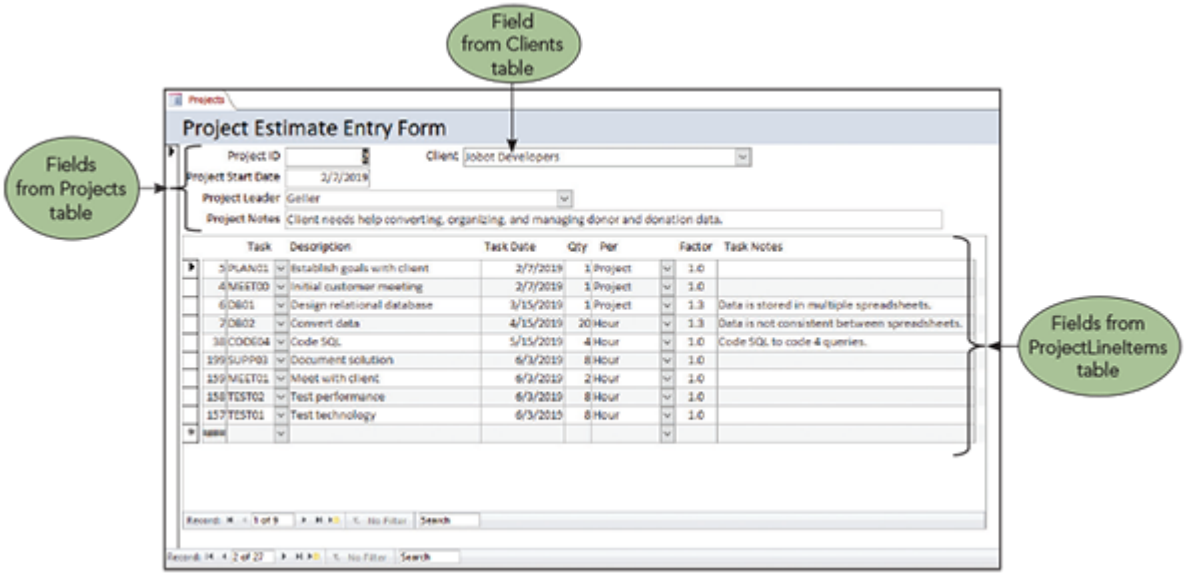
Task Master List Entry Form



A more complicated form for entering the information needed for each project estimate is shown in Figure 1-11. This form displays data about each project using data from the Clients, Projects, and ProjectLineItems tables.

Figure 1-11

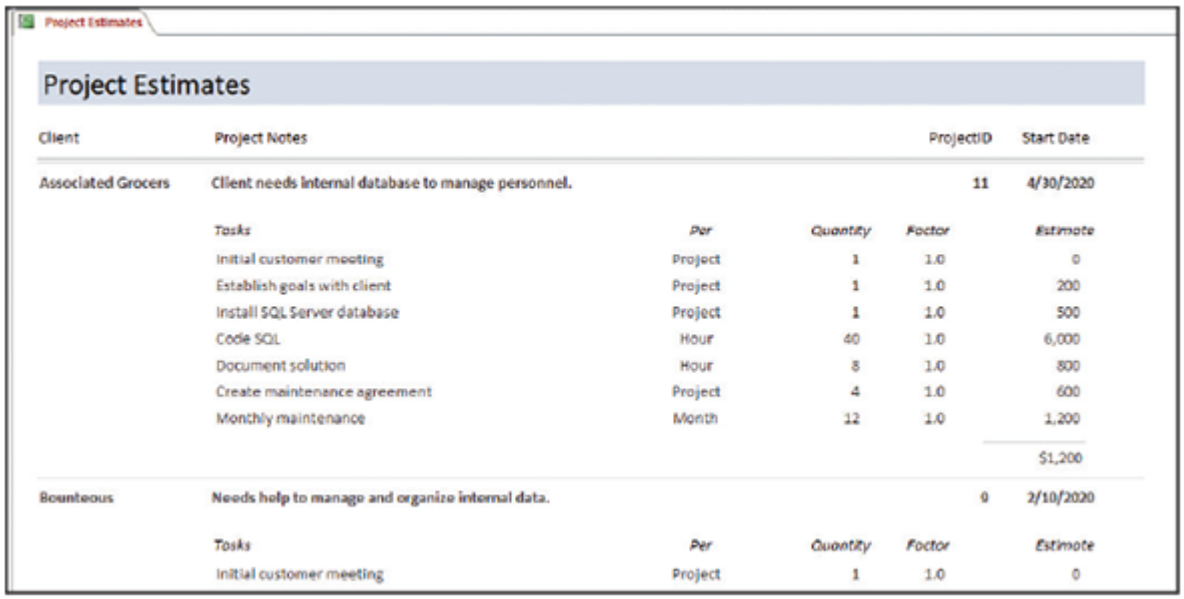
Project Estimate Entry Form



JCC can create the **reports** (A database object that creates a professional printout of data.) it needs using the report generation tools provided by Access. The Project Estimates report, which lists the client, project notes, project ID, project start date, and task information for each project, is shown in Figure 1-12.

Figure 1-12

Project Estimates Report



Project Estimates

Client	Project Notes				ProjectID	Start Date
Associated Grocers	Client needs internal database to manage personnel.				11	4/30/2020
	Tasks	*Per*	*Quantity*	*Factor*		*Estimate*
	Initial customer meeting	Project	1	1.0		0
	Establish goals with client	Project	1	1.0		200
	Install SQL Server database	Project	1	1.0		500
	Code SQL	Hour	40	1.0		6,000
	Document solution	Hour	8	1.0		800
	Create maintenance agreement	Project	4	1.0		600
	Monthly maintenance	Month	12	1.0		1,200
						$1,200
Bounteous	Needs help to manage and organize internal data.				9	2/10/2020
	Tasks	*Per*	*Quantity*	*Factor*		*Estimate*
	Initial customer meeting	Project	1	1.0		0

Chapter 1: Introduction to Database Management: 1-5 Advantages of a Properly Designed Relational Database
Book Title: Concepts of Database Management
Printed By: Michael Ammerman (mammerman0005@kctcs.edu)

1-5 Advantages of a Properly Designed Relational Database

The relational database approach to organizing and managing data offers several clear advantages over alternative data management methods. These advantages are listed in Figure 1-13.

Figure 1-13

Advantages of a Properly Designed Relational Database System

1. Better information is provided.
2. Data and information are shared.
3. Multiple business information requirements are addressed.
4. Data redundancy is minimized.
5. Data consistency is applied.
6. Referential integrity is enforced.
7. Security is increased.
8. Productivity is increased.
9. Data is freed from individual applications.

1. ***Better information is provided.*** A fundamental goal of a database system is to turn data (recorded facts) into information (the knowledge gained by processing those facts). In a flat-file environment, data is partitioned into several disjointed systems, lists, and files. Any request for information that involves accessing data from more than one of these areas can be difficult to fulfill, especially summarized data that helps confirm assumptions, analyze trends, and spot exceptions.

2. ***Data and information are shared.*** Several users have access to the same data—for example, a client's correct name and address. When one user changes a client's address, the new address immediately becomes available to all users.

3. ***Multiple business information requirements are addressed.*** For the database approach to function adequately within an organization, a person or group should be in charge of the database, especially if the database will serve many users. This person or group is often called the **database administrator** (The person or group in charge of a database. Also called *database administration (DBA).*) or **database administration (DBA)** (The person or group in charge of a database. Also called the *database administrator.*) . By keeping the overall needs of the organization in mind, a DBA can structure the database to benefit the entire organization, not just a single group.

4. ***Data redundancy is minimized.*** With database processing, data that formerly was kept in separate file-oriented systems is integrated into a single database, so multiple copies of the same data are minimized. Minimizing data redundancy makes the process of updating data simpler and less error prone.

5. ***Data consistency is applied.*** Using a relational database, attribute values (field values) are entered consistently, which helps users correctly find, filter, and analyze the data. For example, a properly organized relational database will not allow a user to enter a State field value in different ways (TX versus Texas).

6. ***Referential integrity is enforced.*** **Referential integrity** (A relational database concept that sets rules called integrity constraints on table relationships primarily to prevent the creation of orphan records.) is a relational database concept that sets rules called **integrity constraints** (A rule that must be followed by data in a database.) on table relationships primarily to prevent the creation of orphan records. **Orphan records** (A record in the “many” (child) table that has no match in the “one” (parent) table in a one-to-many relationship.) are records in an entity (table) on the “many” side of a relationship that do not have a matching record with the entity (table) on the “one” side of a relationship.

   For example, a project estimate cannot be created for a client that doesn’t yet exist in the database because that would create a project for an unknown client. Also, a client cannot be deleted from the database if it has related projects as that would also create projects with unknown clients. A database has **integrity** (A database has integrity if the data in it satisfies all established integrity constraints.) when the data in it satisfies all established integrity constraints.

7. ***Security is increased.*** A secure DBMS will have features such as the assignment of user IDs, groups, passwords, and permissions that allow access to certain areas of the database. A secure DBMS will also provide for data **encryption** (Converting the data in a database to a format that is indecipherable by anyone other than an authorized user.) , which protects the data as it moves through a network. Other

processes that some DBMS systems provide to ensure **data security** (Protection of data from threats including the prevention of unauthorized access to the database, encryption of data as it travels through a network, protection against data corruption, and protection against all other electronic and physical attacks to the data.) are methods to prevent and recover from data corruption, protections from unauthorized access and malware attacks, and automatic rollbacks and backups to provide reliable starting points in the event of a security breach.

8. ***Productivity is increased.*** With a DBMS, the programmers who are writing programs that use data from the database do not have to engage in mundane data manipulation activities, such as adding new data and deleting existing data, thus making the programmers more productive. These features increase the productivity of programmers as well as nonprogrammers, who may be able to get the results they seek from the data in a database without waiting for a program to be written for them. Accurate, reliable data and reports improve the productivity as well as the effectiveness of all analysts, managers, and employees who use the information.

9. ***Data is freed from individual applications.*** A good DBMS provides **data independence** (A quality that allows you to change the structure of a database without requiring you to make major changes to the programs that access the database.), a quality that allows you to change the structure of a database without requiring major changes to the programs that access the database. Without data independence, programmers often need to expend effort to update programs to match the new database structure. The presence of many programs in the system can make this effort so prohibitively difficult that management might decide to avoid changing the database, even though the change might improve the database's performance or add valuable data.

Q&A

1-6

What is data security?

Chapter 1: Introduction to Database Management: 1-5a Key Factors for a Healthy Relational Database
Book Title: Concepts of Database Management
Printed By: Michael Ammerman (mammerman0005@kctcs.edu)

1-5a Key Factors for a Healthy Relational Database

While the advantages of a relational database are compelling, they are predicated on the relational database being healthy. Figure 1-14 identifies some of the key factors for developing and maintaining a healthy relational database system.

Figure 1-14

Key Factors for a Healthy Relational Database System

1. Design.
2. Security.
3. Talent.

1. ***Design.*** If the database is not designed properly, meaning that if the entities, attributes, and relationships are not correctly identified and created, it is difficult to enter, edit, find, and analyze data. Improperly designed relational databases mean that queries and reports may be incomplete, inconsistent, or inaccurate, which undermines high-quality decisions.

   In addition, entity (table) and attribute (field) names should be as short as possible, yet long enough to be clear and descriptive. Naming conventions should be established and applied so that all developers can easily read and maintain the relational database system. Poor naming conventions or lack of programming standards and disciplines creates unnecessary confusion and costly programming mistakes.

2. ***Security.*** When several users are sharing the same database, a security breach to the database has a dramatic impact on the business. Security and backup processes and protocols need to be established. A business must commit to hiring experienced talent and modern tools and resources to protect any data that is vital to the operation of the company.

3. ***Talent.*** The process of creating, maintaining, and securing a healthy relational database requires a high level of talent and experience. A business must commit to

either hiring or training individuals that can successfully manage these important responsibilities.

1-6 Big Data

Companies have access to more and different kinds of data than ever before. The term **big data** (The large volume of data produced by every digital process, system, sensor, mobile device, and even social media exchange.) describes the large volume of data produced by every digital process, system, sensor, mobile device, and even social media exchange. To extract meaningful information from big data, companies need optimal processing power, analytics capabilities, and up-to-date technical skills.

Big data may be structured, unstructured, or a combination of both. **Structured data** (Data that is traditional in its retrieval and storage in database management systems.) is traditional in its retrieval and storage DBMS, similar to the data in the JCC database in this module. **Unstructured data** (Data not organized or easily interpreted by traditional databases or data models, which may involve a lot of text and metadata.) is not organized or easily interpreted by traditional databases or data models. Unstructured data may involve **metadata** (Descriptive data stored with input sources.) —descriptive data stored with input sources.

Q&A

1-7

What are other examples of unstructured data?

Unstructured data is often stored in nonrelational database systems. The term **NoSQL** (A database management system such as MongoDB that uses a document model made up of collections and documents to store data.) refers to a database management system such as MongoDB that uses a document model made up of collections and documents to store data.

Insights derived from big data enable companies to make better decisions about trends, customer engagement, threats, and new revenue streams. Big data is a source for ongoing discovery and analysis, and the demand for information from big data will require new approaches to database management, architecture, tools, and practices.

Chapter 1: Introduction to Database Management: 1-6 Big Data
Book Title: Concepts of Database Management
Printed By: Michael Ammerman (mammerman0005@kctcs.edu)
© 2020 Cengage Learning, Cengage Learning

1-7 Preparing for a Career in Database Administration and Data Analysis

Database administrators (DBAs) manage and maintain database management systems and software. Their responsibilities range from managing the physical aspects of the database such as installing, maintaining, and testing hardware and software to designing and improving the database to provide efficient and effective access to the information. Most DBAs for large companies have college degrees and certifications from software vendors.

Data analysts (A job title for a person who uses tools and algorithms to mine a database for answers, information, trends, and insights in internal organization data.) use tools and algorithms to mine a database for answers, information, trends, and insights. An **algorithm** (A defined set of steps to solve a problem.) is a set of rules, calculations, and assumptions used to solve a problem. For example, a data analyst might apply an algorithm that includes historical information about product sales and advertising to project future sales. Data analysts in one company might be called business analysts, operations research analysts, management analysts, statisticians, or data scientists in another depending on their focus and responsibilities. Data analysts generally have a college degree and command a healthy annual salary.

Q&A

1-8

What are typical salaries for a database administrator and data analyst?

Chapter 1: Introduction to Database Management: 1-7 Preparing for a Career in Database Administration and Data Analysis
Book Title: Concepts of Database Management
Printed By: Michael Ammerman (mammerman0005@kctcs.edu)
© 2020 Cengage Learning, Cengage Learning

1-8 Introduction to the Pitt Fitness Database Case

Pitt Fitness is a chain of three fitness centers that offer classes to the residents of Pittsburgh, Pennsylvania. The centers design the classes for all ages and do not charge a monthly fee. The owners of Pitt Fitness have found that their clients enjoy the concept of paying per class, resulting in a lively revenue stream with this model.

The managers at Pitt Fitness use a database management system to record their customers, instructors, classes, and class reservations. Figure 1-15 displays the instructors that work at Pitt Fitness. Each instructor is identified by a unique InstructorID, which contains two letters and two numbers. Other information about the instructors are their full name, street address, email address, and telephone number.

Figure 1-15

Instructor Data for Pitt Fitness

Instructors

InstructorID	InstructorLastName	InstructorFirstName	InstructorStreetAddress	InstructorCity	InstructorState	InstructorZ
JV01	Varlano	Juan	15A Penn Ave	Pittsburgh	PA	15219
LL01	Lane	Luke	2661 Butler St	Pittsburgh	PA	15201
MA01	Nguyen	Michael	2367 Barn Way	Pittsburgh	PA	15219
MD01	D'Angelo	Maria	245D Wylie Ave	Pittsburgh	PA	15219
MK01	Kobinski	Megan	9087 Monanca Pl	Pittsburgh	PA	15219
MS01	Said	Memo	45 Webster Ave	Pittsburgh	PA	15219
NT01	Tahan	Neda	4588 Penn Ave	Pittsburgh	PA	15219
RS01	Stein	Raymond	3254 Forward Ave	Pittsburgh	PA	15217
RS02	Sisto	Robert	2836 Maryland Ave	Pittsburgh	PA	15232
VP01	Pegues	Vicki	3700 Murray Ave	Pittsburgh	PA	15217

Figure 1-16 shows some of the classes that Pitt Fitness offers. Each class has a ClassID, which is a unique identification number. Other class information includes class name, center and room identification, length of time, maximum size, category, day of the week and time, plus a brief description.

Figure 1-16

Class Data for Pitt Fitness

Classes

ClassID	ClassName	Room	Location	LengthOfTime	MaxSize	TypeOfClass	Day	Time	Description
1	Combination	159B	Shadyside	60	40	Combo	Saturday	9:00	Cardio, strength, flexibility
2	Yogalates	342	Downtown	60	25	Core	Monday	6:00	Yoga and Pilates fundamentals

ClassID	ClassName	Room	Location	LengthOfTime	MaxSize	TypeOfClass	Day	Time	Description
3	Barbell Power	Main	Oakland	60	25	Strength	Tuesday	8:00	Barbell program
4	Intense Cycle	163	Shadyside	30	25	Cardio	Saturday	10:00	Sprint
5	Zumba	159B	Shadyside	60	25	Cardio	Saturday	10:00	Zumba dance
6	Low Impact Aerobics	Main	Oakland	60	25	Cardio	Tuesday	11:00	Low impact aerobics
7	Yoga	342	Downtown	60	25	Core	Tuesday	7:00	Traditional yoga
8	Yoga Balance	Main	Oakland	60	25	Core	Thursday	8:00	Balance yoga
9	Barre	Main	Oakland	60	25	Core	Thursday	9:00	Pilates yoga and aerobics
10	Barre Limited	342	Downtown	30	25	Core	Wednesday	17:30	Barre brief
11	Maturity Endurance and Strength	Main	Oakland	60	25	Strength	Friday	10:30	Cardio, strength, flexibility for older adults
12	Maturity Classics	159B	Shadyside	60	25	Strength	Wednesday	7:00	Strength and movement for older adults
13	Agility for Seniors	Main	Oakland	60	25	Core	Friday	9:30	Agility for older adults
14	HIIT	315	Downtown	30	25	Cardio	Thursday	18:00	High intensity interval drills
15	Bootcamp	159B	Shadyside	30	25	Cardio	Saturday	11:00	Weight training and cardio
16	Cycle	320	Downtown	60	25	Cardio	Friday	6:30	Sprints and climbs
17	Cycle and Strength	163	Shadyside	60	25	Cardio	Sunday	12:30	Cycle and strength
18	Aquasize	Pool	Oakland	60	25	Cardio	Tuesday	14:00	Low impact cardio water exercise
19	Aqua Strength and Cardio	125P	Shadyside	60	25	Cardio	Wednesday	6:00	Strength and cardio in the water
20	Aqua Calm	Pool	Oakland	45	25	Strength	Tuesday	13:00	Low impact water exercise
21	Combination	159B	Shadyside	60	40	Combo	Sunday	9:00	Cardio, strength, flexibility
22	Yogalates	342	Downtown	60	25	Core	Tuesday	6:00	Yoga and Pilates fundamentals
23	Barbell Power	Main	Oakland	60	25	Strength	Wednesday	8:00	Barbell program

ClassID	ClassName	Room	Location	LengthOfTime	MaxSize	TypeOfClass	Day	Time	Description
24	Intense Cycle	163	Shadyside	30	25	Cardio	Sunday	10:00	Sprint
25	Zumba	159B	Shadyside	60	25	Cardio	Sunday	10:00	Zumba dance
26	Low Impact Aerobics	Main	Oakland	60	25	Cardio	Wednesday	11:00	Low impact aerobics
27	Yoga	342	Downtown	60	25	Core	Wednesday	6:00	Traditional yoga
28	Yoga Balance	Main	Oakland	60	25	Core	Friday	8:00	Balance yoga
29	Barre	Main	Oakland	60	25	Core	Friday	9:00	Pilates yoga and aerobics
30	Barre Limited	342	Downtown	30	25	Core	Thursday	17:30	Barre brief
31	Maturity Endurance and Strength	Main	Oakland	60	25	Strength	Saturday	10:30	Cardio, strength, flexibility for older adults
32	Maturity Classics	159B	Shadyside	60	25	Strength	Thursday	7:00	Strength and movement for older adults
33	Agility for Seniors	Main	Oakland	60	25	Core	Saturday	9:30	Agility for older adults
34	HIIT	315	Downtown	30	25	Cardio	Friday	18:00	High intensity interval drills
35	Bootcamp	159B	Shadyside	30	25	Cardio	Sunday	11:00	Weight training and cardio
36	Cycle	320	Downtown	60	25	Cardio	Saturday	6:30	Sprints and climbs
37	Cycle and Strength	163	Shadyside	60	25	Cardio	Monday	12:30	Cycle and strength
38	Aquasize	Pool	Oakland	60	25	Cardio	Wednesday	14:00	Low impact cardio water exercise
39	Aqua Strength and Cardio	125P	Shadyside	60	25	Cardio	Thursday	6:00	Strength and cardio in the water
40	Aqua Calm	Pool	Oakland	45	25	Strength	Wednesday	13:00	Low impact water exercise

Figure 1-17 shows customer data for Pitt Fitness. Each customer is identified by a unique customer number. In addition, management stores each customer's last name, first name, street address, city, state, zip code, email address, phone number, and birth date.

Figure 1-17

Customer Data for Pitt Fitness

Customers

CustomerID	LastName	FirstName	StreetAddress	City	State	ZipCode	EmailAddress	PhoneNu
101	Aboud	Nour	4898 Negley Ave	Pittsburgh	PA	15232	NourAboud@cengage.com	412-255-
102	Waldron	Tony	766 Myrtle Way	Pittsburgh	PA	15232	TonyLWaldron@cengage.com	617-825-
103	Arian	Farah	998 Forward Ave	Pittsburgh	PA	15217	FarahArian@cengage.com	878-243-
104	Cane	James	4310 Elmer St	Pittsburgh	PA	15232	JamesJCane@cengage.com	610-555-
105	Brescia	Keith	1595 Holden St	Pittsburgh	PA	15232	KeithWBrescia@cengage.com	724-943-
106	Conner	Don	3004 Alder St	Pittsburgh	PA	15232	DonSConner@cengage.com	412-281-
107	Feldman	Terrance	2158 Semple St	Pittsburgh	PA	15213	TerranceLFeldman@cengage.com	559-673-
108	Gregor	Alice	392 Murray Ave	Pittsburgh	PA	15217	AliceRGregor@cengage.com	770-832-
109	Miller	Roland	2542 York Way	Pittsburgh	PA	15213	RolandJMiller@cengage.com	412-266-
110	Sanchez	Ramiro	4983 McKee Pl	Pittsburgh	PA	15213	RamiroSanchez@cengage.com	878-576-
111	Barry	Juan	4534 Urie Way	Pittsburgh	PA	15213	JuanRBarry@cengage.com	210-444-
112	Smith	Sharon	1937 Bates St	Pittsburgh	PA	15213	SharonMSmith@cengage.com	878-268-
113	Hatcher	Charles	3218 Louisa St	Pittsburgh	PA	15213	CharlesMHatcher@cengage.com	561-550-
114	Spencer	Glenn	2764 Oakland Ave	Pittsburgh	PA	15213	GlennJSpencer@cengage.com	878-688-
115	Hearn	James	1252 Pier St	Pittsburgh	PA	15213	JamesJHearn@cengage.com	231-640-
116	Cornett	Albert	3545 Ermine Way	Pittsburgh	PA	15213	AlbertACornett@cengage.com	724-975-
117	Devito	Sylvia	3281 Fiber Way	Pittsburgh	PA	15213	SylviaKDevito@cengage.com	256-486-
118	Hill	Jess	3161 Argyle Way	Pittsburgh	PA	15213	JessAHill@cengage.com	480-854-
119	Benavides	Philip	1465 Butler St	Pittsburgh	PA	15201	PhilipCBenavides@cengage.com	412-868-
120	Patterson	Margo	3291 Mahon St	Pittsburgh	PA	15219	MargoFPatterson@cengage.com	412-290-
121	Thorn	Debbie	4434 Watt St	Pittsburgh	PA	15219	DebbieLThorn@cengage.com	412-323-
122	Shaffer	Gene	2041 Wylie Ave	Pittsburgh	PA	15219	GeneKShaffer@cengage.com	281-817-
123	Brough	Pablo	1812 Penn Ave	Pittsburgh	PA	15224	PabloRBrough@cengage.com	909-577-
124	Agnew	Betty	201 Barn Way	Pittsburgh	PA	15219	BettyBAgnew@cengage.com	878-263-

CustomerID	LastName	FirstName	StreetAddress	City	State	ZipCode	EmailAddress	PhoneNu
125	McCauley	Raymond	3254 Memory Lane	Pittsburgh	PA	15219	RaymondCMcCauley@cengage.com	878-345-
126	Sisto	Robert	2836 Bedford Ave	Pittsburgh	PA	15219	RobertJSisto@cengage.com	646-373-
127	Lara	Tamara	4588 Monaca Pl	Pittsburgh	PA	15219	TamaraRLara@cengage.com	417-359-
128	Allen	Michael	2367 Cliff St	Pittsburgh	PA	15219	MichaelHAllen@cengage.com	412-790-
129	Pegues	Vicki	3700 Penn Ave	Pittsburgh	PA	15219	VickiJPegues@cengage.com	412-597-
130	Choi	Min Jee	2661 Webster Ave	Pittsburgh	PA	15219	MinJeeChoi@cengage.com	412-868-

Figure 1-18 shows reservations data for Pitt Fitness. Each reservation is identified by a unique number that uses the last two digits of the current year followed by a five-digit number that is incremented sequentially as each reservation is received. The table also stores the identification number of the class, the date of the class, the price of the class, and any other fees associated with the class. Finally, the customer's unique identification number is included.

Figure 1-18

Reservation Data for Pitt Fitness

Reservations

ReservationID	ClassID	ClassDate	ClassPrice	OtherFees	CustomerID
2100001	39	01-07-2021	$9.00		102
2100002	24	01-03-2021	$11.00	$5.00	102
2100003	10	01-06-2021	$9.00		103
2100004	9	01-07-2021	$9.00		104
2100005	7	01-05-2021	$9.00		106
2100006	1	01-02-2021	$9.00		106
2100007	2	01-04-2021	$9.00		107
2100008	3	01-05-2021	$9.00		108
2100009	5	01-02-2021	$9.00		109
2100010	8	01-07-2021	$9.00		111
2100011	4	01-02-2021	$9.00		111
2100012	9	01-07-2021	$9.00		113
2100013	10	01-06-2021	$9.00		113
2100014	14	01-07-2021	$9.00		120
2100015	15	01-02-2021	$9.00		121
2100016	17	01-03-2021	$11.00	$5.00	122
2100017	18	01-05-2021	$10.00		123
2100018	35	01-03-3021	$9.00		124
2100019	36	01-02-2021	$11.00	$5.00	119

ReservationID	ClassID	ClassDate	ClassPrice	OtherFees	CustomerID
2100020	2	01-04-2021	$9.00		120
2100021	9	01-07-2021	$9.00		121
2100022	6	01-05-2021	$9.00		122
2100023	4	01-02-2021	$11.00	$5.00	123
2100024	21	01-03-2021	$9.00		124
2100025	24	01-03-2021	$11.00	$5.00	125
2100026	26	01-06-2021	$9.00		126
2100027	15	01-02-2021	$9.00		127
2100028	25	01-03-2021	$9.00		127
2100029	29	01-08-2021	$9.00		129
2100030	27	01-06-2021	$9.00		130
2100031	31	01-02-2021	$9.00		101
2100032	32	01-07-2021	$9.00		105
2100033	33	01-02-2021	$9.00		110
2100034	40	01-06-2021	$10.00		112
2100035	20	01-05-2021	$10.00		114
2100036	11	01-08-2021	$9.00		115
2100037	12	01-06-2021	$9.00		116
2100038	13	01-08-2021	$9.00		117
2100039	31	01-02-2021	$9.00		118

The table named ClassInstructors shown in Figure 1-19 is used to relate classes and instructors. It includes the class number and the instructor number. The class number in the ClassInstructors table matches a class number in the Classes table, and the instructor number in the ClassInstructors table matches an instructor number in the Instructors table. Note that some classes use more than one instructor.

Figure 1-19

Table Used to Relate Classes and Instructors

ClassInstructors

ClassID	InstructorID
1	RS02
2	RS01
3	VP01
4	MS01
5	MK01
6	MK01
7	RS01
8	LL01
9	LL01

ClassID	InstructorID
10	LL01
11	MA01
11	MD01
12	MA01
12	MD01
13	MA01
13	MD01
14	RS01
15	RS02
16	MS01
17	MS01
18	NT01
19	NT01
20	JV01
21	MD01
22	RS01
23	RS01
24	MS01
25	MK01
26	NT01
27	LL01
28	RS01
29	VP01
30	VP01
31	MA01
31	MD01
32	MA01
32	MD01
33	MA01
33	MD01
34	RS02
35	RS02
36	MS01
37	MS01
38	JV01
39	JV01
40	NT01

Q&A

1-9

To check your understanding of the Pitt Fitness data, which classes does InstructorID NT01, Neda Tahan, teach?

To check your understanding of the relationship between instructors, classes, and class instructors, answer the following questions.

- **Question:** What instructor teaches Barbell Power on Wednesday in Oakland?

  Answer: First look at the Classes table, shown in Figure 1-16, to determine the ClassID of that specific class. The Wednesday Barbell Power class has a ClassID of 23. Refer to the ClassInstructors table, shown in Figure 1-19, to find that InstructorID RS01 teaches the class with ClassID 23. Finally, in the Instructors table, shown Figure 1-15, look for InstructorID RS01 to find that Raymond Stein teaches Barbell Power, ClassID 23, on Wednesday in Oakland.

- **Question:** Customer Philip Benavides cannot remember his class reservation. Which class did he sign up for?

  Answer: First look at the Customers table, shown Figure 1-17, and find out that customer Philip Benavides is CustomerID 119. In the Reservations table, shown in Figure 1-18, read that CustomerID 119 signed up for ClassID 36 under reservation number 2100019. In the Classes table, shown in Figure 1-16, look up ClassID 36 to find that the class Philip Benavides signed up for is Cycle.

- **Question:** Which classes does Michael Nguyen teach? What are those classes called?

  Answer: First look up the instructor, Michael Nguyen, in the Instructors table, shown in Figure 1-15. Reading that Michael Nguyen has InstructorID MA01, you can use the ClassInstructors table, shown in Figure 1-19, to look up which classes Michael Nguyen teaches. The ClassInstructors table shows that Instructor MA01 teaches classes with ClassIDs 11, 12, 13, 31, 32, and 33. Finally, use the Classes table, shown in Figure 1-16, to look up the names of those classes: Maturity Endurance and Strength, Maturity Classics, and Agility for Seniors.

The E-R diagram for the Pitt Fitness database appears in Figure 1-20.

Figure 1-20

E-R Diagram for the Pitt Fitness Database

Chapter 1: Introduction to Database Management: 1-9 Introduction to the Sports Physical Therapy Database Case
Book Title: Concepts of Database Management
Printed By: Michael Ammerman (mammerman0005@kctcs.edu)

1-9 Introduction to the Sports Physical Therapy Database Case

Sports Physical Therapy provides evaluation, treatment, and rehabilitation of all types of acute and chronic injuries for both athletes and non-athletes. The highly skilled, certified therapists use their background of biomechanics, sport mechanics, and clinical experience to provide one-on-one comprehensive rehabilitation for all types of injuries. The company stores information about their patients, therapists, therapies, and sessions.

In the Patient table shown in Figure 1-21, Sports Physical Therapy stores information about its patients. Each patient is identified by a unique, four-digit patient number. The patient's name and address, as well as the balance due on their bill, also are stored in the table.

Figure 1-21

Patient Data for Sports Physical Therapy

Patient

PatientNum	LastName	FirstName	Address	City	State	ZipCode	Balance
1010	Koehler	Robbie	119 West Bay Dr.	San Vista	TX	72510	$1,535.15
1011	King	Joseph	941 Treemont	Oak Hills	TX	74081	$212.80
1012	Houghland	Susan	7841 Lake Side Dr.	Munster	TX	72380	$1,955.40
1013	Falls	Tierra	44 Applewood Ave.	Palm Rivers	TX	72511	$1,000.35
1014	Odepaul	Ben	546 WCR 150 South	Munster	TX	74093	$525.00
1015	Venable	Isaiah	37 High School Road	Waterville	TX	74183	$432.30

PatientNum	LastName	FirstName	Address	City	State	ZipCode	Balance
1016	Waggoner	Brianna	2691 Westgrove St.	Delbert	TX	72381	$714.25
1017	Short	Tobey	1928 10th Ave.	Munster	TX	72512	$967.60
1018	Baptist	Joseph	300 Erin Dr.	Waterville	TX	76658	$1,846.75
1019	Culling	Latisha	4238 East 71st St.	San Vista	TX	74071	$1,988.50
1020	Marino	Andre	919 Horton Ave.	Georgetown	TX	72379	$688.95
1021	Wilson	Tammy	424 October Blvd.	Waterville	TX	76658	$2,015.30

Sports Physical Therapy records information about each of its therapy sessions. The fields of data are stored in the Session table shown in Figure 1-22. A session record will have a unique number, the session date, the patient number, and the length of the session, as well as the therapist number and therapy code.

Figure 1-22

Session Data for Sports Physical Therapy

Session

SessionNum	SessionDate	PatientNum	LengthOfSession	TherapistID	TherapyCode
27	10/10/2021	1011	45	JR085	92507
28	10/11/2021	1016	30	AS648	97010
29	10/11/2021	1014	60	SW124	97014
30	10/12/2021	1013	30	BM273	97033
31	10/15/2021	1016	90	AS648	98960

SessionNum	SessionDate	PatientNum	LengthOfSession	TherapistID	TherapyCode
32	10/16/2021	1018	15	JR085	97035
33	10/17/2021	1017	60	SN852	97039
34	10/17/2021	1015	45	BM273	97112
35	10/18/2021	1010	30	SW124	97113
36	10/18/2021	1019	75	SN852	97116
37	10/19/2021	1020	30	BM273	97124
38	10/19/2021	1021	60	AS648	97535

To check your understanding of the relationship between patients and sessions, which patient had therapy on October 15, 2021? What therapy code was listed for the session belonging to Isaiah Venable? What session number(s) is (are) listed for Tierra Falls?

Answer: The Session table (Figure 1-22) lists PatientNum 1016 as having therapy on October 15, 2021. When you look that patient up in the Patient table (Figure 1-21), you see that it is Brianna Waggoner. To find the therapy code for Isaiah Venable, you must start with the Patient table, look up his number, 1015, and then examine the Session table. Patient number 1015 had the therapy coded as 97112. Finally, Tierra Falls is patient number 1013 (Patient table). Looking up her session number in the Session table, it is 30.

Sports Physical Therapy stores information about the therapists who work in their office, as shown in the Therapist table in Figure 1-23. Each therapist is identified by a unique ID number that consists of two uppercase letters followed by a three-digit number. For each therapist, the table also includes the last name, first name, street, city, state, and zip code.

Figure 1-23

Therapist Data for Sports Physical Therapy

Therapist

TherapistID	LastName	FirstName	Street	City	State	ZipCode
AS648	Shields	Anthony	5222 Eagle Court	Palm Rivers	TX	72511

TherapistID	LastName	FirstName	Street	City	State	ZipCode
BM273	McClain	Bridgette	385 West Mill St.	Waterville	TX	76658
JR085	Risk	Jonathan	1010 650 North	Palm Rivers	TX	72511
SN852	Nair	Saritha	25 North Elm St.	Livewood	TX	72512
SW124	Wilder	Steven	7354 Rockville Road	San Vista	TX	72510

To check your understanding of the relationship between therapists and sessions, which therapist worked with patient 1021? How many patients did Bridgette McClain work with? What were the therapy codes (TherapyCode) for those sessions?

Answer: To determine which therapist worked with patient 1021, first examine the Session table (Figure 1-22). Find patient 1021; look across the table to see the TherapistID, AS648. Then look up the TherapistID in the Therapist table (Figure 1-23) to find the name, Anthony Shields.

To determine the number of patients that Bridgette McClain worked with, look up her TherapistID number in the Therapist table (Figure 1-23). You will see that it is BM273. Then look at the Session table (Figure 1-22). In the TherapistID column, count the number of times you see BM273—it should be three. Finally, look at the TherapyCode column for each of those three sessions. You should identify therapies 97033, 97112, and 97124.

In the Therapies table, each kind of therapy is identified by a unique number, which corresponds to the medical physical therapy code sent to insurance companies for reimbursement. The TherapyCode, a description, and the billable unit of time, if any, are included in the table. Time-based therapies are billed in groups of minutes (listed in the table). Service-based therapies are billed per service (no time is listed in the table). Figure 1-24 displays data for therapies.

Figure 1-24

Therapies Data for Sports Physical Therapy

Therapies

TherapyCode	Description	UnitOfTime
90901	Biofeedback training by any modality	
92240	Shoulder strapping	
92507	Treatment of speech	15
92530	Knee strapping	
92540	Ankle and/or foot strapping	
95831	Extremity or trunk muscle testing	
97010	Hot or cold pack application	
97012	Mechanical traction	
97014	Electrical stimulation	
97016	Vasopneumatic devices	
97018	Paraffin bath	
97022	Whirlpool	
97026	Infrared	
97032	Electrical stimulation	15
97033	Iontophoresis	15
97035	Ultrasound	15
97039	Unlisted modality	15
97110	Therapeutic exercises to develop strength and endurance, range of motion, and flexibility	15
97112	Neuromuscular re-education of movement, balance, coordination, etc.	15
97113	Aquatic therapy with therapeutic exercises	15
97116	Gait training	15
97124	Massage	15
97139	Unlisted therapeutic procedure	
97140	Manual therapy techniques	15
97150	Group therapeutic procedure	15

TherapyCode	Description	UnitOfTime
97530	Dynamic activities to improve functional performance, direct (one-on-one) with the patient	15
97535	Self-care/home management training	15
97750	Physical performance test or measurement	15
97799	Unlisted physical medicine/rehabilitation service or procedure	
98941	CMT of the spine	
98960	Education and training for patient self-management	30

Q&A

1-10

To check your understanding of the relationship between therapies and the other tables, answer the following questions: Did any patient have a hot or cold pack application? Which therapist(s) helped a patient with gait training? How many minutes did Jonathan Risk work with his patient on speech therapy, and how many units will be billed to insurance?

The E-R diagram for the Sports Physical Therapy database appears in Figure 1-25.

Figure 1-25

E-R Diagram for the Sports Physical Therapy Database

Chapter 1: Introduction to Database Management: 1-9 Introduction to the Sports Physical Therapy Database Case
Book Title: Concepts of Database Management
Printed By: Michael Ammerman (mammerman0005@kctcs.edu)