Microsoft

Microsoft Azure Data Fundamentals

Exam Ref DP-900

Daniel A. Seara
Francesco Milano

FREE SAMPLE CHAPTER

Microsoft

Exam Ref DP-900 Microsoft Azure Data Fundamentals

Daniel A. Seara
Francesco Milano

Exam Ref DP-900 Microsoft Azure Data Fundamentals

Published with the authorization of Microsoft Corporation by:
Pearson Education, Inc.

ISBN-13: 978-0-13-725216-9
ISBN-10: 0-13-725216-1

Library of Congress Control Number: 2021931458

ScoutAutomatedPrintCode

SPECIAL SALES

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at corpsales@pearsoned.com or (800) 382-3419.

For government sales inquiries, please contact governmentsales@pearsoned.com.

For questions about sales outside the U.S., please contact intlcs@pearson.com.

CREDITS

EDITOR-IN-CHIEF
Brett Bartow

EXECUTIVE EDITOR
Loretta Yates

DEVELOPMENT EDITOR
Songlin Qiu

SPONSORING EDITOR
Charvi Arora

MANAGING EDITOR
Sandra Schroeder

SENIOR PROJECT EDITOR
Tracey Croom

COPY EDITOR
Liz Welch

INDEXER
Valerie Haynes Perry

PROOFREADER
Scout Festa

TECHNICAL EDITOR
Herbert Albert

EDITORIAL ASSISTANT
Cindy Teeters

COVER DESIGNER
Twist Creative, Seattle

Contents at a glance

Contents

Acknowledgments

I would like to thank the following people, who helped me during the work on this book and in my life, both professional and personal.

First, thank you to my wife, Nilda Beatriz Díaz, for helping me daily be a better person and a better professional, and for sharing with me the adventure of this life and this astounding work, all around the world.

I would also like to thank all the members of our team at Lucient, who walk with me in the path of knowledge and in the process of providing our customers with the services they deserve. For this particular book, one of them, Herbert Albert, was especially helpful, reviewing all our technical content. Thanks again, my friend; I owe you another set of Argentinian-style pizzas.

And finally, I would like to thank Lilach Ben-Gan, who makes my English writing more readable and clearer for you, the reader, and keeps our writing work flowing smoothly and on time.

Daniel Seara

While I am used to preparing and delivering live sessions, courses, and short articles, this was my first time writing a technical book. It is a very intensive and unique experience and, at the same time, the perfect occasion to rearrange and extend my knowledge about the topics covered. But also, it is something I could not have achieved alone.

I have to say a big thank-you to my wife and daughters for living many hours with a “ghost” in their house. It must not have been easy at times, but they heartfully managed to give me all the time I needed.

I would also like to thank everyone at Lucient, in particular the Italian team that took care of additional work to compensate for my months-long disappearance. Two special mentions: One is for Lilach Ben-Gan, who had the thankless task of improving my English and making it understandable, and the other one is for Herbert Albert, whose precious suggestions helped immensely in shaping the technical content to its best possible form.

Finally, a big hug goes to my parents and parents-in-law for being our great helping hand. I really appreciate all your unrelenting efforts, and knowing you were there made the writing of this book more feasible.

Francesco Milano

The authors would also like to thank the team at Pearson who helped with the production of this book: Loretta Yates, Charvi Arora, Songlin Qiu, Liz Welch, Danielle Foster, and Tracey Croom.

About the authors

Daniel A. Seara is an experienced software developer. He has more than 20 years' experience as a technical instructor, developer, and development consultant.

Daniel has worked as a software consultant in a wide range of companies in Argentina, Spain, and Peru. He has been asked by Peruvian Microsoft Consulting Services to help several companies in their migration path to .NET Framework development.

Daniel was Argentina's Microsoft Regional Director for 4 years and was the first nominated Global Regional Director, a position he held for two years. He was also the manager of the Desarrollador Cinco Estrellas I (Five-Star Developer) program, one of the most successful training projects in Latin America. Daniel held a Visual Basic MVP status for more than 10 years, as well as a SharePoint Server MVP status from 2008 until 2014. Additionally, Daniel is the founder and "Dean" of Universidad .NET, the most visited Spanish language site on which to learn .NET.

In 2005, he joined Lucient, the leading global company on the Microsoft Data Platform, where he has been working as a trainer, consultant, and mentor.

Francesco Milano has been working with Microsoft technologies since 2000.

Francesco specializes in the .NET Framework and SQL Server platform, and he focuses primarily on back-end development, integration solutions, relational model design, and implementation.

Since 2013 Francesco has also been exploring emerging trends and technologies pertaining to the big data and advanced analytics world, consolidating his knowledge of products like Azure HDInsight, Databricks, Azure Data Factory, and Azure Synapse Analytics.

Francesco is a speaker at prominent Italian data platform conferences and workshops.

In 2015, he joined Lucient, the leading global company on the Microsoft Data Platform, where he has been working as a trainer, consultant, and mentor.

Introduction

In this connected era, it is important to determine how and when your data can be stored in the cloud. This book, both a reference and a tutorial, covers the different approaches to storing information in the Microsoft Azure environment. The book discusses and compares various storage options, helping you make better choices based on each particular need, and guides you through the steps to prepare, deploy, and secure the most appropriate storage environment.

This book covers every major topic area found on the exam, but it does not cover every exam question. Only the Microsoft exam team has access to the exam questions, and Microsoft regularly adds new questions to the exam, making it impossible to cover specific questions. You should consider this book a supplement to your relevant real-world experience and other study materials. If you encounter a topic in this book that you do not feel completely comfortable with, use the “Need more review?” links you'll find in the text to find more information and take the time to research and study the topic. Great information is available on MSDN, on TechNet, and in blogs and forums.

Organization of this book

This book is organized by the “Skills measured” list published for the exam. The “Skills measured” list is available for each exam on the Microsoft Learn website: *http://aka.ms/examlist*. Each chapter in this book corresponds to a major topic area in the list, and the technical tasks in each topic area determine a chapter’s organization. If an exam covers six major topic areas, for example, the book will contain six chapters.

Preparing for the exam

Microsoft certification exams are a great way to build your résumé and let the world know about your level of expertise. Certification exams validate your on-the-job experience and product knowledge. Although there is no substitute for on-the-job experience, preparation through study and hands-on practice can help you prepare for the exam. This book is *not* designed to teach you new skills.

We recommend that you augment your exam preparation plan by using a combination of available study materials and courses. For example, you might use the Exam Ref and another study guide for your ”at home” preparation and take a Microsoft Official Curriculum course for the classroom experience. Choose the combination that you think works best for you.

Learn more about available classroom training and find free online courses and live events at *http://microsoft.com/learn*. Microsoft Official Practice Tests are available for many exams at *http://aka.ms/practicetests*.

Note that this Exam Ref is based on publicly available information about the exam and the authors' experience. To safeguard the integrity of the exam, authors do not have access to the live exam.

Microsoft certifications

Microsoft certifications distinguish you by proving your command of a broad set of skills and experience with current Microsoft products and technologies. The exams and corresponding certifications are developed to validate your mastery of critical competencies as you design and develop, or implement and support, solutions with Microsoft products and technologies both on-premises and in the cloud. Certification brings a variety of benefits to the individual and to employers and organizations.

> ***MORE INFO*** **ALL MICROSOFT CERTIFICATIONS**
>
> **For information about Microsoft certifications, including a full list of available certifications, go to *http://www.microsoft.com/learn*.**

Check back often to see what is new!

Quick access to online references

Throughout this book are addresses to webpages that the author has recommended you visit for more information. Some of these links can be very long and painstaking to type, so we've shortened them for you to make them easier to visit. We've also compiled them into a single list that readers of the print edition can refer to while they read.

Download the list at *MicrosoftPressStore.com/ExamRefDP900AzureFundamentals/downloads*.

The URLs are organized by chapter and heading. Every time you come across a URL in the book, find the hyperlink in the list to go directly to the webpage.

Errata, updates & book support

We've made every effort to ensure the accuracy of this book and its companion content. You can access updates to this book—in the form of a list of submitted errata and their related corrections—at:

MicrosoftPressStore.com/ExamRefDP900AzureFundamentals/errata

If you discover an error that is not already listed, please submit it to us at the same page.

For additional book support and information, please visit *http://www.MicrosoftPressStore.com/Support*.

Please note that product support for Microsoft software and hardware is not offered through the previous addresses. For help with Microsoft software or hardware, go to *http://support.microsoft.com*.

Stay in touch

Let's keep the conversation going! We're on Twitter: *http://twitter.com/MicrosoftPress*.

CHAPTER 2

Describe how to work with relational data on Azure

Relational data is the most used storage since the last quarter of the past century. It is likely the concept most students study at the very beginning of their careers. You will find concepts about how the data is stored, and the best ways to design them, in hundreds of books. No matter what kind of information you want to preserve, a relational database is most likely a good option.

> ***NOTE*** **OTHER OPTIONS**
>
> **As you will read in the next chapter, a relational database is not the only option, and in some cases, relational data storage is not the best choice.**

Skills covered in this chapter:

- Skill 2.1: Describe relational data workloads
- Skill 2.2: Describe relational Azure data services
- Skill 2.3: Identify basic management tasks for relational data
- Skill 2.4: Describe query techniques for data using SQL language

Skill 2.1: Describe relational data workloads

Relational data storage is described as storing information based on a predefined structure of the information. Depending on the use of your data and your workload, you must select the technique that best matches your needs. Conceptually, in relational databases you try to define things to represent the entities in the real world, like persons, companies, products, bills, and so on. We use the term "relational" to describe the relation in the data representing an entity, and not just because, for example, one bill could be related to a person and a customer and was generated by a company. Moreover, it can have several products in the details, and all these elements are related. All this information must be stored in some way, and that is what we will cover here.

This skill covers how to:

- Identify the right data offering for a relational workload
- Describe relational data structures

Identify the right data offering for a relational workload

If you analyze how your data has been managed in the past, usually you find one or more applications storing information in a centralized storage, probably a single database. Unless different business processes, or different areas, are involved with specific privacy or security reasons, you will find a lot of applications storing all the information in just one database. However, during recent years, this has been changing. A lot of information is now stored in several formats and places all around the world (in fact, all around the "cloud").

And this is an important matter to consider. Not only must you manage the data, but you also must get information from several sources and, probably, adapt it to match the way your business uses the information.

NOTE **INFORMATION JOURNEY**

Consider the information traveling in an information pipeline, where each station can modify, extract, change, or refine information. That is the way information is managed these days.

Online transaction processing (OLTP)

This workload is what we typically get from business transactions, like bank transfers, online shopping, and cash machines, that are preserved in a data store. It is the repository for any transaction related to the activities.

In a health-care system, the information about every patient and each event—disease or symptom, treatment, blood analysis, X-ray, and so forth—consists of activities for the system, and usually they are related in order to manage the information clearly.

The concepts about OLTP are well known. The workload has been deeply analyzed, and many rules have been defined to make OLTP work better. Probably the most important is the atomicity, consistency, isolation, durability (ACID) concept, which defines the properties of database transactions that must be completed to guarantee sustainable operations.

EXAM TIP

ACID is a very important concept. In this book, you have the basic definitions, but other resources elaborate on it. As a starting point, you can read the first article about this concept, "Principles of transaction-oriented database recovery," at *https://dl.acm.org/doi/10.1145/289.291*.

ATOMICITY

The name "atomicity" derives from the concept of an atom. It is something that must be together. It is "all or nothing."

Consider this scenario: A patient requires treatment in the ER. The doctor needs some laboratory checks for diagnostics purposes. The doctor performs some procedures to cure the diagnosed disease.

When the procedures are completed, several pieces of information must be recorded:

1. The patient's symptoms
2. The list of laboratory checks
3. The result of those checks
4. Each procedure, medical instrument, medication and dosage
5. The closure: recommendations, future follow-up procedures, and so on

All this information and all the detailed costs of the procedures must be recorded as a single unit. It is not useful, for example, to have the symptoms without the laboratory results.

Ensuring that all the information is stored as one block, as an atom including all the parts at the same time, is *atomicity*.

CONSISTENCY

The information stored in a relational database usually has defined rules to ensure that all the information makes sense. Using the previous example, there is no sense in having the laboratory results without any indication of which patient they belong to, or the exact definition of the procedure.

Ensuring that the information can be related in a specific way in the future is *consistency*.

ISOLATION

Isolation ensures that other actors in the process do not access partial information.

Two different areas in the hospital using the same information must access the same data. If someone at the ER office is entering the information at the same time another person is preparing the bill, it will not be good if the second person obtains the already stored laboratory checks while the first person is still completing the registration of the procedures or drugs used to treat the patient.

During the update procedure, until the consistency has been maintained, the information for this specific transaction must be isolated from others.

EXAM TIP

There is some fine-tuning of isolation, the so-called *isolation levels*. It is important to understand how they modify the behavior of the reads in a database environment. You can learn more here: *https://docs.microsoft.com/en-us/sql/connect/jdbc/understanding-isolation-levels*.

DURABILITY

Durability ensures that the information can be accessed later even after a system crash. Most relational database systems (RDBSs) use a mechanism to quickly store each step of an activity and then confirm all of them at the same time (known as a *commit*).

After the commit succeeds, the information is secure. Of course, IT departments must deal with external factors, but from a relational database point of view, the information is safe.

Online analytical processing (OLAP)

The OLAP workload, even when still a relational workload, was developed with data analysis in mind. You can think of it as looking to the past. The important element here is analyzing what happened instead of registering what is going on.

Using the previous example, OLAP will be used to evaluate how many patients the ER treated in the last week, or month, or year; how many require follow-up; the average number of laboratory procedures per patient; and so on.

The most important difference between OLTP and OLAP is that OLAP is implemented for reading big amounts of data for data analysis, whereas OLTP is designed for many parallel write transactions.

Another difference you can find in OLAP implementations is the fact that, usually, the OLAP data has been restructured to facilitate the queries.

Look at the partial entity-relationship diagram of products in the Adventure Works OLTP database, shown in Figure 2-1, and compare it with the diagram for products in the Adventure Works OLAP database, shown in Figure 2-2. The second one is more simplistic, but the tables contain more columns. Moreover, if you look at the Product table in the OLAP version, you will see that it has columns that are in other related tables in the OLTP model. That is because the OLAP data is *flattened* several times to accelerate the reads during the query process.

> ***NOTE*** **DIFFERENT SCHEMAS**
>
> **Notice that the entities in both schemas do not have exact matches; they are used just as a sample to better illustrate OLAP database design and do not necessarily match the structured database design rules.**

The OLAP database uses a *semantic model* instead of a *database schema*. The semantic model redefines the information from a business point of view, rather than using a structured point of view as the OLTP database schema does. This is because the business user, who is the final consumer for an OLAP implementation, knows the business entities but not the underlying data schema.

The semantic model usually contains calculations already performed, time-oriented calculations, aggregation from different tables to make it easier to read the information, and in some cases, aggregation from different sources.

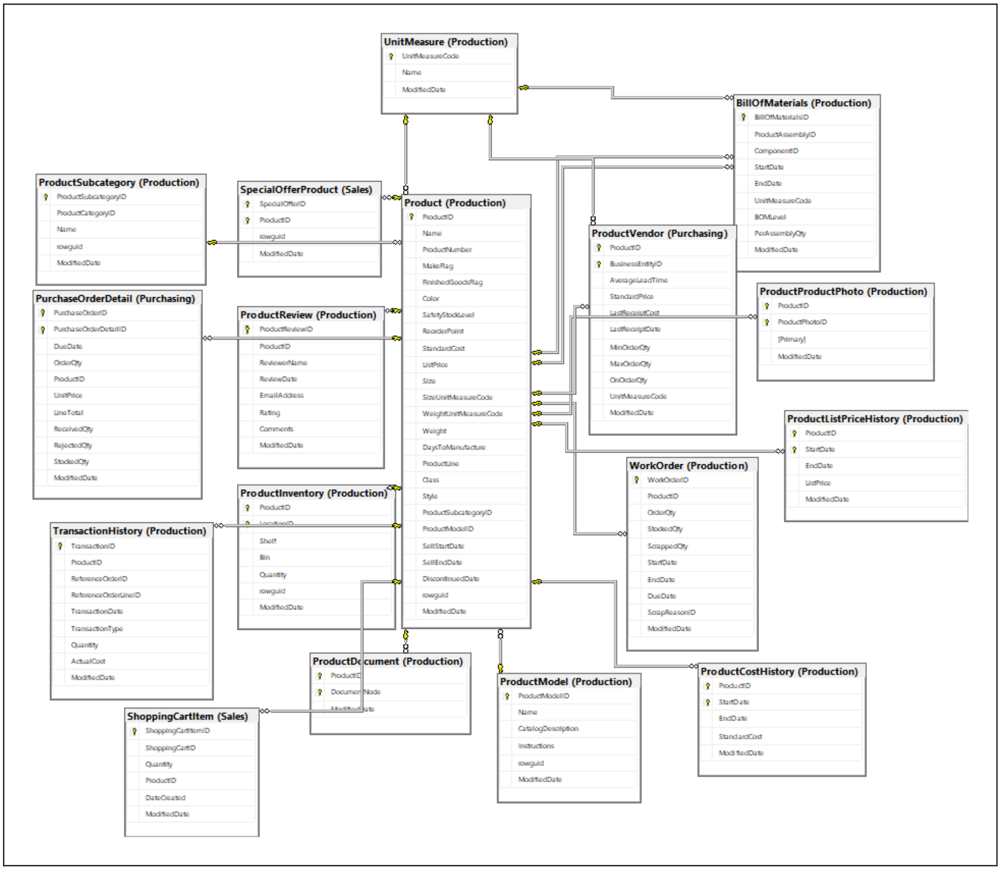

FIGURE 2-1 OLTP database product relationships

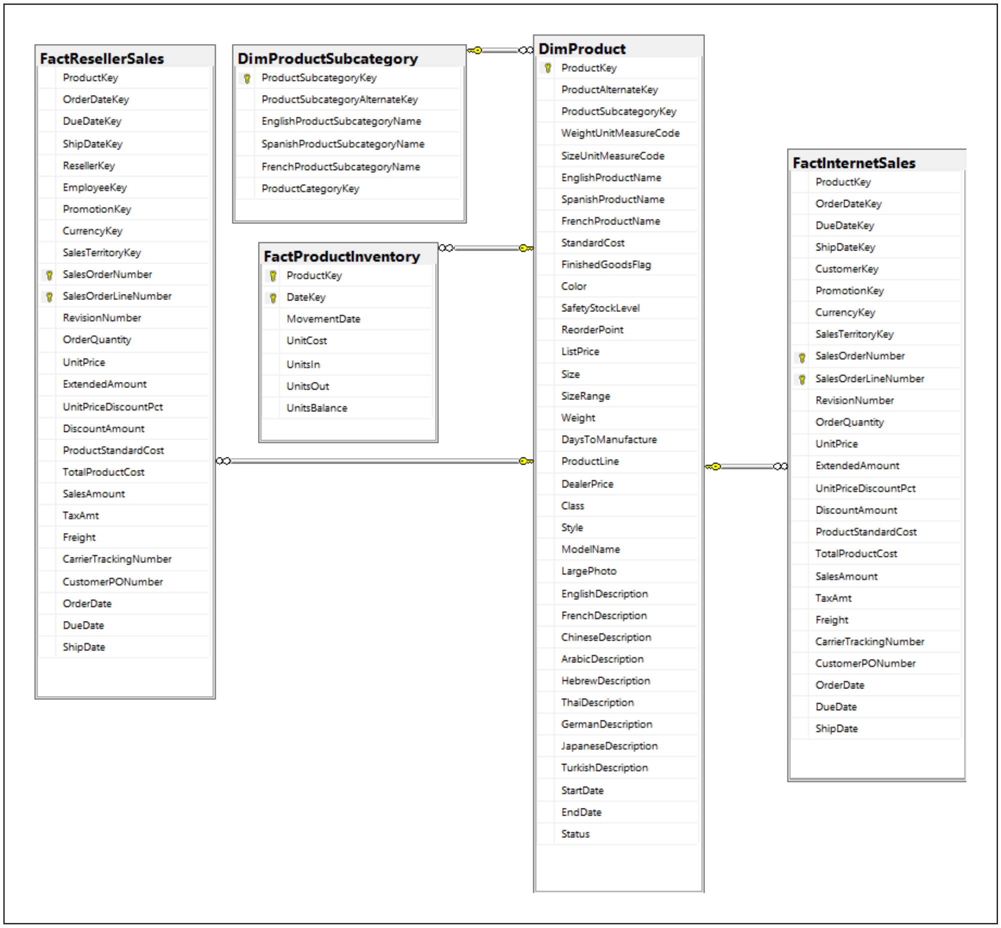

FIGURE 2-2 OLAP database product relationship

When you define an OLAP workload, you must decide which kind of semantic model to use, as shown in Table 2-1.

TABLE 2-1 OLAP semantic models

OLAP Model	Description
Tabular	Like OLTP models, this model uses concepts such as tables, columns, and relationships.
Multidimensional	A more traditional OLAP approach is used, based on cubes, dimensions, and measures.

Data warehousing

Using information from different sources, during a long period of time, implies keeping historical information in a secure, consistent way. Moreover, the storage solution must not burden the other workloads with the analytical process. This is where a data warehouse comes in.

A data warehouse is the place to store historical and current information, preprocessed in ways that facilitate the business analytical queries to get better results. In the implementation of a data warehouse, procedures are used to cleanse the data and make it consistent. Because the information can come from disparate sources, it must be preprocessed to facilitate better results from the business analytical queries.

Several different tools and procedures are available to keep the information up-to-date in a data warehouse, but all can be defined as a three-part process: extract the information from the sources; store the results in the data warehouse; and transform, process, and ensure data quality in some parts of the process.

Sometimes, you prefer to transform the data before storing it in the data warehouse (the extract, transform, and load [ETL] process). In other circumstances, it could be more reliable, more secure, or simply cheaper to move all the information into the data warehouse and then process it (the extract, load, and transform [ELT] process).

NEED MORE REVIEW? **TRANSFORMATION PROCESSES**

For more information about the transformation processes, review Skill 1.2, “Describe data analytics core concepts,” in this book.

Describe relational data structures

Relational data is about having the information stored according to specific structures and predefined elements. This ensures the quality of the queries, the relationships, and the consistency of the information. The following are several concepts related to how the information is structured in relational data structures.

Tables

A *table* is the basic structure where data is stored. A table predefines the parts of the data, and the information stored in it must match the defined schema.

A table defines *columns* to identify each piece of information about the entity it stores. Consider the set of information in Table 2-2 (let’s say it is information about sales regions).

TABLE 2-2 Table data sample

Name	Country	Start	SalesLastYear
North	US	05/01/2010	$ 3,298,694.49
Central	US	06/01/2012	$ 3,205,014.08
South	US	03/01/2008	$ 5,366,575.71
Canada	CA	08/01/2010	$ 5,693,988.86
France	FR	09/01/2006	$ 2,396,539.76
Germany	DE	10/01/2012	$ 1,307,949.79
Australia	AU	11/01/2018	$ 2,278,548.98

To store the information, a relational database must have a table that defines the columns, including their properties. The column definition specifies not only the name of each column (which must be unique to the table), but also the type of information the column will contain in each entry.

In some cases, when the entities you want to store have different sizes, most database engines allow you to define a specific or a maximum size.

Also, you can apply other kinds of restrictions. In this example, just one column is allowed to have no value, since the first time a new entry is added, no value for that column is added (for example, a new region will not have sales from the previous year, since it is new). This concept is represented in Table 2-3.

TABLE 2-3 Data columns and constrains

Column name	Type	Size	Allow empty
Name	Characters	100	No
Country	Characters	2	No
Start	Date		No
SalesLastYear	Money		Yes

Each database engine has its own data type definitions. However, most of them define the same standards, often with different nomenclatures and some specific data types not shared with others. But the most important types are the same for all of them. Table 2-4 shows the various data types.

TABLE 2-4 Standard data types

Information Type	Standard Data types	
Characters	**Size**	**Data Types**
	Fixed length	char nchar (Unicode)
	Variable length	varchar nvarchar (Unicode)
Numbers	**Size**	**Data Types**
	Integer	integer smallinteger biginteger tinyinteger
	Non-integer	decimal numeric float real double money
Other data	**Size**	**Data Types**
	Dates	smallDateTime dateTime time timespan
	Logical	**bit**
	Other	binary image Etc.

EXAM TIP

The name of *nvarchar*, or *nchar*, stands for *N*ational *CHAR*acters. Using the N at the beginning of the name signals that the data type is for Unicode/double-byte characters.

Indexes

When you have a lot of information stored in a table, finding a specific entry could be time consuming. Imagine yourself in a room with hundreds of thousands of folders of information, trying to find a specific entry. Without classifications, you are in for a lot of work to find the information you are searching for.

Now think about having each folder with hundreds of pages . . . you will have to lift each of the folders to see if it is the correct one. That can be heavy work!

Something similar occurs in the database engine.

Finding your folder will be so much easier if you have a collection of tabs, with the tabs ordered and just the most important information to identify each one of your folders. That way, you can quickly locate the folder you are looking for in all your libraries.

That is the concept behind indexes. Instead of you reading each entire row, one at a time, to find the entry you need, the system searches an index to get the exact location of the information in the table.

In Figure 2-3, you can see how the index search works.

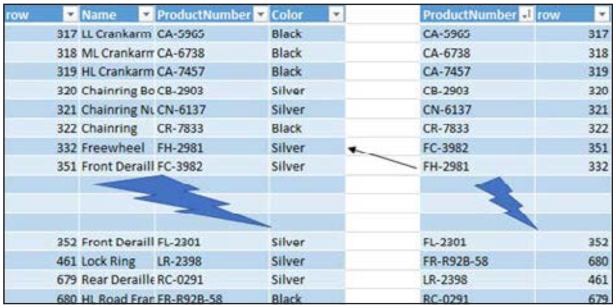

FIGURE 2-3 Index search

In a similar way, indexes can combine more than one column for lookup purposes.

Indexes can be used to:

- Ensure uniqueness of each key in a table, defined as the unique key.
- Establish the most important key to search, called the primary key.
- Use relationships to speed up search correlation between data in columns in one table and the values of the column(s) of the primary key of another table.

Views

Once you have data stored in tables, you probably need to filter or regroup information in different ways for different users. Most important, it is often the case that not all the information stored in each table can be viewed by all your users. You might have sensitive information intended only for a subset of users or just a couple of columns some users need to view. In that case, you can use *views* to redefine the data to make it accessible in a reliable and secure form.

Consider a table with employee information. Any person in the company may need information from this table. However, salaries must not be visible to anyone except Human Resources personnel.

Here is another example. Suppose management needs the total sales by vendor, employee, year, and month. Instead of making management perform the calculation, you can have the information ready, in an already prepared view.

Keep in mind that the view does not *store* information. It is a virtual definition of how you want to see the information. Every time you query the view, the database platform will query the original table(s) to show you only the information you need.

A view is just a statement to query data from the table(s), not the final data. To enhance performance, when the database engine receives the order to store a view, it performs the following steps:

1. Checks the correctness of the statement itself
2. Verifies that all the columns and tables in use are present in the database
3. Determines the best plan to query the different parts of the data retrieved
4. Compiles the statement with that best plan (usually named the query or execution plan)

By doing this, the database engine, once executed the first time, will have the query plan in the cache and can use it.

EXAM TIP

Data changes with time. When the engine estimates a query plan, different tables can have a different number of rows, and the tables can have different amounts of data when it is required by the view.

That is why the data engine uses statistics to evaluate how much the data has changed.

If the statistics of one or more tables implied in a view are changed, the engine recalculates the query plan and stores the new one, before extracting the results.

Listing 2-1 is a sample of a view created to get information from five different tables.

LISTING 2-1 View sample

```
CREATE VIEW [Salestotal]
AS
   SELECT
      YEAR([Soh].[Duedate]) AS                        [Year]
    , MONTH([Soh].[Duedate]) AS                       [Month]
    , [Prod].[Name] AS                                [Product]
    , [Per].[Lastname] + ', ' + [Per].[Firstname] AS [Vendor]
    , SUM([Sod].[Orderqty]) AS                        [Quantity]
    , SUM([Sod].[Linetotal]) AS                       [Total]
     FROM
          [Sales].[Salesorderdetail] AS [Sod]
          INNER JOIN
          [Sales].[Salesorderheader] AS [Soh]
          ON
             [Sod].[Salesorderid]
             = [Soh].[Salesorderid]
     INNER JOIN
     [Sales].[Salesperson] AS [Sp]
     ON
        [Soh].[Salespersonid]
        = [Sp].[Businessentityid]
        AND
        [Soh].[Salespersonid]
        = [Sp].[Businessentityid]
     INNER JOIN
     [Production].[Product] AS [Prod]
```

Index

NUMBERS

SYMBOLS

A

B

C

D

O

P

Q

R

S

T

U

V

W

X

Z