



Microsoft Azure Data Fundamentals

Exam Ref DP-900

Daniel A. Seara
Francesco Milano

FREE SAMPLE CHAPTER

SHARE WITH OTHERS



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.

COPYRIGHT © 2021 BY LUCIENT DATA SA.

All rights reserved. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, request forms, and the appropriate contacts within the Pearson Education Global Rights & Permissions Department, please visit www.pearson.com/permissions

No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

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

ISBN-10: 0-13-725216-1

Library of Congress Control Number: 2021931458

ScoutAutomatedPrintCode

TRADEMARKS

Microsoft and the trademarks listed at <http://www.microsoft.com> on the "Trademarks" webpage are trademarks of the Microsoft group of companies. Lucient is a trademark of Lucient Data SA and the Lucient group of companies. All other marks are property of their respective owners.

WARNING AND DISCLAIMER

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an "as is" basis. The author, the publisher, and Microsoft Corporation shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the programs accompanying it.

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

	<i>Introduction</i>	<i>xiii</i>
CHAPTER 1	Describe core data concepts	1
CHAPTER 2	Describe how to work with relational data on Azure	47
CHAPTER 3	Describe how to work with non-relational data on Azure	135
CHAPTER 4	Describe an analytics workload on Azure	203
	<i>Index</i>	<i>305</i>

Contents

Introduction	xiii
Organization of this book	xi
Preparing for the exam	xi
Microsoft certifications	xii
Quick access to online references	xii
Errata, updates & book support	xiii
Stay in touch	xiii
Chapter 1 Describe core data concepts	1
Skill 1.1: Describe types of core data workloads	1
Describe streaming data	3
Describe batch data	10
Describe the difference between batch and streaming data	19
Describe the characteristics of relational data	20
Skill 1.2: Describe data analytics core concepts	22
Describe analytics techniques	23
Describe the concepts of ETL, ELT, and data processing	28
Describe data visualization and basic chart types	36
Chapter summary	43
Thought experiment	44
Thought experiment answers	44
Chapter 2 Describe how to work with relational data on Azure	47
Skill 2.1: Describe relational data workloads	47
Identify the right data offering for a relational workload	48
Describe relational data structures	53

Skill 2.2: Describe relational Azure data services	58
Describe and compare PaaS, IaaS, and SaaS delivery models	60
Describe Azure SQL Database	63
Describe Azure Synapse Analytics	69
Describe SQL Server on Azure Virtual Machine	74
Describe Azure Database for PostgreSQL, Azure Database for MariaDB, and Azure Database for MySQL	79
Describe Azure SQL Managed Instance	83
Skill 2.3: Identify basic management tasks for relational data	87
Describe provisioning and deploying relational data services	87
Describe method for deployment including ARM templates and Azure Portal	90
Identify data security components (e.g., firewall, authentication)	107
Identify basic connectivity issues (e.g., accessing from on-premises, access with Azure VNets, access from internet, authentication, firewalls)	112
Identify query tools (e.g., Azure Data Studio, SQL Server Management Studio, sqlcmd utility, etc.)	114
Skill 2.4: Describe query techniques for data using SQL language	122
Compare DDL versus DML	123
Query relational data in PostgreSQL, MySQL, and Azure SQL Database	126
Chapter summary	131
Thought experiment	132
Thought experiment answers	133

Chapter 3 Describe how to work with non-relational data on Azure 135

Skill 3.1: Describe non-relational data workloads	135
Describe the characteristics of non-relational data	136
Describe the types of non-relational and NoSQL data	137
Choose the correct data store	142
Determine when to use non-relational data	143

Skill 3.2: Describe non-relational data offerings on Azure	143
Identify Azure data services for non-relational workloads	144
Describe Azure Cosmos DB API	144
Describe Azure Storage	155
Describe Azure Table storage	158
Describe Azure Blob storage	163
Describe Azure File storage	170
Skill 3.3: Identify basic management tasks for non-relational data	175
Describe provisioning and deployment of non-relational data services	175
Describe method for deployment including the Azure portal, Azure Resource Manager templates, Azure PowerShell, and the Azure command-line interface (CLI)	176
Identify data security components (e.g., firewall, authentication, encryption)	182
Identify basic connectivity issues (e.g., accessing from on-premises, access with Azure VNets, access from internet, authentication, firewalls)	190
Identify management tools for non-relational data	194
Chapter summary	198
Thought experiment	199
Thought experiment answers	201
Chapter 4 Describe an analytics workload on Azure	203
Skill 4.1: Describe analytics workloads	203
Skill 4.2: Describe the components of a modern data warehouse	207
Describe modern data warehousing architecture and workload	207
Describe Azure data services for modern data warehousing such as Azure Data Lake, Azure Synapse Analytics, Azure Databricks, and Azure HDInsight	208

Skill 4.3: Describe data ingestion and processing on Azure	232
Describe the components of Azure Data Factory (e.g., pipeline, activities, etc.)	233
Describe data processing options (e.g., Azure HDInsight, Azure Databricks, Azure Synapse Analytics, Azure Data Factory)	254
Describe common practices for data loading	276
Skill 4.4: Describe data visualization in Microsoft Power BI	278
Describe the workflow in Power BI	279
Describe the role of interactive reports	279
Describe the role of dashboards	294
Describe the role of paginated reporting	297
Chapter summary	299
Thought experiment	302
Thought experiment answers	304
Index	305

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*.

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.

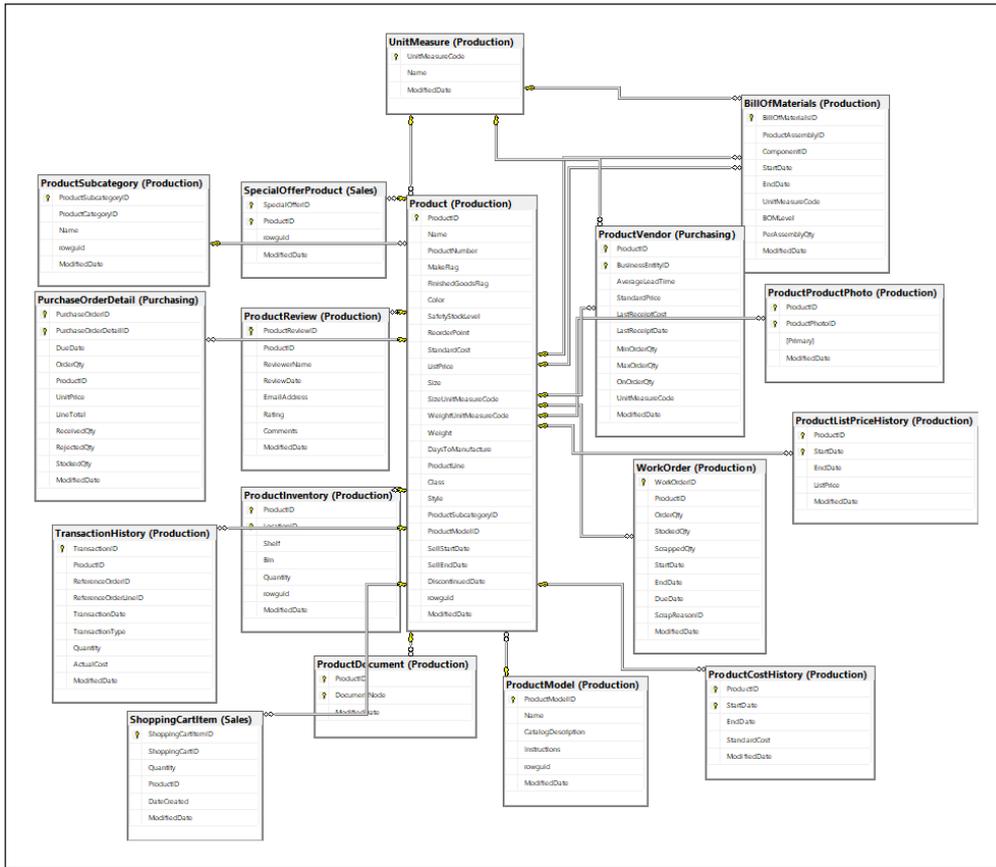


FIGURE 2-1 OLTP database product relationships

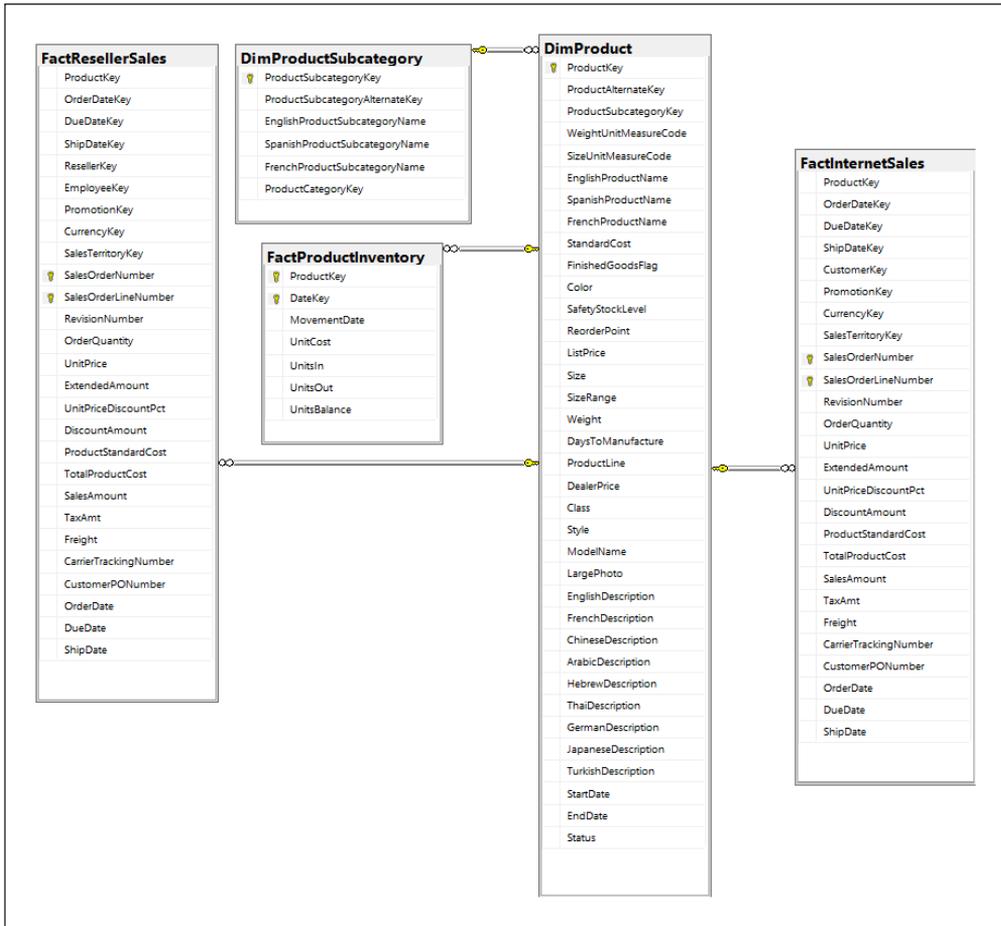


FIGURE 2-2 OLAP database product relationship

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.

row	Name	ProductNumber	Color	ProductNumber	row
317	LL Crankarm	CA-5965	Black	CA-5965	317
318	ML Crankarm	CA-6738	Black	CA-6738	318
319	HL Crankarm	CA-7457	Black	CA-7457	319
320	Chainring Bo	CB-2903	Silver	CB-2903	320
321	Chainring NL	CN-6137	Silver	CN-6137	321
322	Chainring	CR-7833	Black	CR-7833	322
332	Freewheel	FH-2981	Silver	FC-3982	351
351	Front Derail	FC-3982	Silver	FH-2981	332
352	Front Derail	FL-2301	Silver	FL-2301	352
461	Lock Ring	LR-2398	Silver	FR-R92B-58	680
679	Rear Derail	RC-0291	Silver	LR-2398	461
680	HL Road Fran	FR-R92B-58	Black	RC-0291	679

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

5 V's of big data, 12–19

SYMBOLS

* (asterisk), using with columns, 126

@ (at) symbol, using in SQL Server, 111

A

ABFS (Azure Blob Filesystem), 165, 168

ABFS or WASB drivers, 226

access management, 109–110

ACID (atomicity, consistency, isolation, durability), 48–50

acquisition, 4

AD (Active Directory). *See* Azure Active Directory

ADF (Azure Data Factory) v2, 31–34

adjacency list, managing, 140–141

adjacency matrix, 140–141

AES (Advanced Encryption Standard), 108

aggregation and transformation, 4

AI, integrating in pipelines, 16

aliasing long names, 128

ALTER command, 124

Always On availability groups, SQL Server, 84

analytical access, 37

analytics curve, 23

analytics process, 42. *See also* data analytics

analytics workloads, overview, 203–207

Apache Hive and HiveQL, 257

Apache Kafka, 5

Apache Oozie, 11, 233

Apache Spark, 35–36, 268. *See also* Azure Databricks Runtime

ARM (Azure Resource Manager) templates, 95–103, 176–179

 creating Cosmos DB account from, 178

 getting from Cosmos DB account, 176–177

 in GitHub, 181

asterisk (*), using with columns, 126

at (@) symbol, using in SQL Server, 111

atomicity in ACID, 49

attributes, 21

authentication

 managing, 108–111

 issues with non-relational data, 193–194

authorization

 managing, 110

 non-relational data, 189–190

AutoML (automated machine learning), 26. *See also* machine learning prediction

auto-scaling, Azure HDInsight, 211

az module, using with PowerShell, 104

AzCopy, 196

Azure

 datacenter drawing, 59

 event hubs and Apache Kafka, 5

 services components, 62

 services for non-relational workloads, 144

Azure account, obtaining, 1

Azure Active Directory, 110–111, 171

Azure Blob API, 167–168

Azure Blob storage

Azure Blob storage. *See also* blob storage

- API, 167–168
 - Azure Storage Explorer, 166–167
 - Blob driver, 165
 - containers, 164
 - content from PowerShell, 168–170
 - content types, 164
 - data for data analysis, 165
 - hierarchical structure, 165
 - .NET client library, 170
 - organization, 163–164
 - overview, 163
 - and Table storage, 144
- Azure Cache for Redis, 144
- Azure Calculator, 76
- Azure CLI (command-line interface), 105–107, 181
- Azure Cognitive Services, 16, 144
- Azure Cosmos DB API
- account and database, 151–155
 - and Azure Data Explorer, 153
 - Azure Table, 151
 - Cassandra, 150–151
 - configuring consistency, 154
 - consistency levels, 145–147
 - Core (SQL), 150
 - creating account, 177–179
 - Data Migration tool, 154–155
 - deploying, 176–179
 - geo-redundancy, 153–154
 - getting ARM template from, 176–177
 - Gremlin (graph storage), 151
 - high availability, 147
 - importing data to, 154–155
 - IOPS (input/output operations per second), 147
 - JSON example, 149
 - MongoDB, 149–150
 - Notebook feature, 153
 - overview, 144–145
 - request units, 147–149
 - tokens, 146
 - using, 149–151

- Azure Data Catalog, 18
- Azure Data Explorer, 153, 194–196
- Azure Data Factory
- Author menu item, 239–240
 - Azure Blog storage linked service, 248
 - characteristics, 233
 - components, 238–239
 - connectors, 245
 - Copy Data activity, 242–244
 - data integration units, 238
 - data sets, 244–245
 - data stores connectors, 245–247
 - expressions and functions, 249
 - features, 11, 17
 - home page, 235–236
 - IR (integration runtime), 237–238
 - mapping data flows, 271–276
 - monitoring, 254
 - parameters, 240
 - pipeline activities, 239–241
 - provisioning, 234–235
 - SSIS (SQL Server Integration Services), 237–238
 - templates, 235–236
 - testing connection, 248–249
 - triggers, 254
- Azure Data Lake, accessing, 168
- Azure Data Studio, 116–118
- Azure databases
- MariaDB, 81–82
 - MySQL, 82–83
 - PostgreSQL, 79–81
- Azure Databricks. *See also* Spark
- access to storage layer, 225
 - and Apache Spark, 207
 - Azure portal, 220–221
 - characteristics, 217–218
 - clusters, 227–229
 - connector for Azure Synapse Analytics, 278
 - control plane, 221–226
 - Delta Lake, 218
 - features, 35–36, 217–229, 262–263

- joining data, 267
- mounting external storage, 227–229
- provisioning, 220–224, 227–229
- PySpark, 267–268
- source data in PySpark, 266
- and Spark, 263–264
- Spark RDDs, 219–220
- specifying schemas, 264–265
- Azure Databricks File System, 226
- Azure Event Hubs, 206
- Azure File Sync, 174
- Azure Files storage. *See also* storage
 - account mapping information, 173
 - authentication, 171–173
 - authorization, 173
 - Azure File Sync, 174
 - file share to local drive, 173–174
 - Kerberos authentication, 171–173
 - net use command, 174
 - overview, 170
 - SMB (Server Message Block) protocol, 170, 174
 - uploading files, 196
- Azure HDInsight
 - Ambari views, 214–215
 - appending content, 261
 - auto-scaling, 211
 - and Azure portal, 212–216
 - and Azure SQL Database, 213
 - clusters, 210, 212–216
 - considerations, 211
 - data processing options, 257–262
 - destination tables, 260–261
 - features, 11, 35, 206, 209–217
 - Hive optimization, 262
 - Hive tables and Azure Blob storage, 260
 - Hive View console, 258
 - internal and external tables, 258–259
 - and Linux, 212
 - programming languages, 216
 - provisioning Hadoop cluster, 216–217
 - SDKs and IDEs, 215
 - versions of components, 211
- Azure Key Vault, 108, 185, 248. *See also* encryption
- Azure Kubernetes Service (AKS), 27
- Azure Marketplace, 18
- Azure ML (Machine Learning) Designer, 26–27
- Azure .NET libraries, 181–182
- Azure portal
 - ARM templates, 102–103
 - and Azure HDInsight, 212–216
 - creating Cosmos DB account from, 177
 - features, 67–69, 90–95
- Azure services, authorized by default, 183
- Azure SQL Database. *See also* SQL (Structured Query Language)
 - and Azure HDInsight, 213
 - creating, 67–69
 - features, 63
 - purchasing models, 64
 - segmentations, 64–66
 - service models, 67–69
 - using, 15, 89, 92, 108
- Azure SQL-MI (Managed Instance), 83–86, 108
- Azure Storage
 - access tiers, 156–157
 - account types, 156
 - creating accounts, 158
 - deploying, 178–179
 - lifecycle management, 157
 - NFS v3, 158
 - performance levels, 155
 - replication options, 157
 - service exposition, 156
- Azure Stream Analytics, 10, 206
- Azure Synapse Analytics. *See also* PolyBase T-SQL query language; Synapse pool
 - configuring PolyBase in, 269
 - CREATE EXTERNAL TABLE AS SELECT statement, 270
 - creating pools, 74
 - data processing capabilities, 231

- language; Synapse pool (*continued*)
 - external tables, 269–270
 - overview, 229–231
 - private and public preview, 276
 - provisioning SQL pool, 231–232
 - using, 12–13, 36, 69–74, 229–232
 - writing to blob storage, 270–271

- Azure Table API, 151

- Azure Table storage. *See also* tables API, 159–161
 - connecting to, 162–163
 - creating tables, 161–162
 - non-relational workloads, 144
 - OData specification, 159
 - overview, 158–159

B

- batch data

- approach, 18–19
 - described, 10–12
 - versus streaming data, 19–20
 - workload type, 2

- batch layer versus speed layer, 14

- batch workload, 205–206

- BI (business intelligence) projects, 29, 42. *See also* Power BI service

- big data

- value, 18
 - variety, 14–16
 - velocity, 14
 - veracity, 17–18
 - volume, 12–14

- binary file formats, 16

- Blob driver, 165

- Blob service REST API, 167–168

- blob storage. *See* Azure Blob storage

- managing content in, 168–170
 - writing to in Azure Synapse Analytics, 270–271

- blobs, uploading content to, 196

- blocking transformations, 31

- BSON (Binary JSON), 139

- B-tree storage, 137–138

C

- Cached mode versus DirectQuery, 71

- card visual, using in reports, 41

- Cassandra API, 150–151

- character data type, 55

- charts

- and data visualization, 36–42
 - using in reports, 38–39, 41

- CLI (command-line interface), using with Blob storage, 169–170

- cloud, moving data to, 89

- Cloud Shell, using with Azure CLI, 106–107

- clustered columnstore indexes, 63. *See also* indexes

- Codd, Edgar F., 20

- columnar data store, 139–140, 144

- columnstore indexes, 63

- commands

- ALTER, 124

- DROP, 124

- RENAME, 125

- computing nodes, 72

- connectivity issues, non-relational data, 190–194

- connectivity issues, identifying, 112–114. *See also* network security

- consistency in ACID, 49

- constraints, 21

- consumer groups, 5–6

- consumers, 4

- Copy Data Wizard, 250–254

- Core (SQL) API, 150

- Cosmos Explorer, 197

- costs, estimating for SQL server in VM, 76

- CREATE EXTERNAL TABLE AS SELECT statement, 270

- CREATE TABLE AS SELECT statement, 277

- CRM (customer relationship management), 62

- cube structure, SQL Server analysis services, 70

- customer churn, 24

D

- dashboarding access, 37
- dashboards, 294–297. *See also* Power BI service
- data
 - layered access to, 37
 - migrating with DMS, 84
 - volatility of, 14
- data analytics. *See also* analytics process; on-demand
- data analysis
 - core concepts, 23
 - techniques, 23–28
- Data Control Language, 126
- data encryption, 108, 184–185. *See also* Azure Key Vault
- data flows, mapping in Azure Data Factory, 271–276
- data governance, 18
- data ingestion and processing
 - Azure Data Factory, 233–249
 - overview, 232
- data integration units, Azure Data Factory, 238
- data lake, 2
- data loading, 276–278
- Data Migration tool, downloading, 154–155. *See also* migrating data with DMS
- data pipelines
 - data transformations, 28
 - sources and destinations, 28
- data processing options
 - Azure Data Factory, 271–276
 - Azure Databricks, 262–268
 - Azure HDInsight, 257–262
 - Azure Synapse Analytics, 268–271
 - output field list, 256–257
 - overview, 254–257
- data protection, non-relational data, 185–186
- Data Quality Services, 17
- data secure layers, 107
- data security components
 - AzCopy, 196
 - Azure Data Explorer, 194–196
 - connectivity issues, 190–194
 - Cosmos Explorer, 197
 - management tools, 194–198
 - non-relational data workloads, 182–190
- data security components, identifying, 107–110
- data sets, 294
- data stores for non-relational data
 - choosing, 142
 - types, 137–142
- data storytelling, 294
- data stream, 3
- data stream flow
 - acquisition, 4
 - aggregation and transformation, 4
 - production, 3
 - storage, 4
- data structures, classifications, 15–16
- data virtualization, 13, 35
- data visualization and chart types, 36–42
- data warehouse architecture, 2, 19, 53, 207–208. *See also* modern data warehouses
- data warehousing
 - Azure data services, 208–209
 - Azure HDInsight, 209–216
- data workloads
 - batch data, 10–19
 - streaming data, 3–10
 - types of, 1–2
- database schema versus semantic model, 50. *See also* multidimensional databases
- DBFS (Databricks File System), 225
- DDL (Data Definition Language), 123–125
- Delete structure, 125
- Delta Lake, Azure Databricks, 218
- deployment
 - managing with Azure CLI, 105–107
 - managing with PowerShell, 103–105
- descriptive analysis, 23
- diagnostic analysis, 23
- dictionary and key-value store, 137
- directed graph, 140
- DirectQuery versus Cached mode, 71

disk images, creating

- disk images, creating, 76
- DISM (Deployment Image Servicing and Management), 76
- DML (Data Manipulation Language), 125–126
- DMS (Database Migration Service), 84
- document store, 138–139, 144
- domains, 21
- DROP command, 124
- DTS (Data Transformation), 30
- DTU-based purchase model, 64–67
- durability in ACID, 50
- DWUs (data warehouse units), 73
- Dynamic Data Masking, 108

E

- EDW (enterprise data warehouse), 42
- elasticity, 60–61
- ELT (extract-load-transform), 34–36, 205
- encryption. *See* data encryption, 108, 184–185. *See also* Azure Key Vault
- encryption keys, configuring and storing, 185
- ERP (enterprise resource planning), 62
- errors, searching for, 114
- ETL (extract-transform-load), 29–34, 205
- Event Hub Capture, 5
- event hubs, auditing, 109
- events and session windows, 9–10
- external storage, mounting, 227–229

F

- Fiddler tool and connectivity issues, 190–191
- file storage. *See* Azure Files storage
- firewall rules, non-relational data, 182–183
- firewalls
 - managing, 107–111
 - non-relational data, 182–183
- form recognition, 16
- FQDN (fully qualified domain name), 94
- fraud detection, 24

- Free Tier account, non-relational data, 177
- FROM clause, 128
- Fuzzy Lookup/Fuzzy Grouping, 17

G

- geo-redundancy, Azure Cosmos DB API, 153–154
- GIGO (garbage-in, garbage-out), 17
- global distribution, 59
- graph store, 140–141, 144
- Gremlin (graph storage) API, 151
- GRS (geo-redundant storage), 157
- GZRS (geo-zone-redundant storage), 157

H

- Hadoop framework, 12–13, 209
- hash table and key-value store, 137
- HBase in HDInsight, 144
- HDFS (Hadoop Distributed File System), 35, 210
- HiveQL, 257
- hopping window, 7–8
- horizontal partitioning, 30

I

- IaaS (infrastructure as a service), 17, 60, 62
- image classification, 16
- incoming events, off-loading, 5
- indexed information, searching, 144
- indexes, relational database structures, 55–56. *See also* clustered columnstore indexes
- information
 - protection, 108
 - retrieval, 16
- in-memory technologies, 63
- INNER JOIN clause, 128
- input attributes as features, 24
- input rate versus processing rate, 5
- Insert structure, 125

IntelliSense, Azure Data Studio, 118
 isolation in ACID, 49
 IT (information technology) infrastructure, 58

J

JSON (JavaScript Object Notation), 15–16, 138–139

K

Kerberos authentication, Azure File storage, 171–173
 key-value store, 137–138, 144
 KPI (key performance indicator)
 and dashboards, 294
 and SQL Server analysis services, 71
 using in reports, 41

L

law of the instrument, 254
 LEFT OUTER JOIN, 129
 lifecycle management, Azure Storage, 157
 line chart, using in reports, 39
 Linux and Azure HDInsight, 212
 long names, aliasing, 128
 LRS (locally redundant storage), 157

M

machine learning prediction, 16. *See also* AutoML
 (automated machine learning)
 map chart, using in reports, 41–42
 MariaDB, 81–82
 Master Data Services, 17
 matrix visual, using in reports, 38
 message identifications, 192
 Microsoft
 Learn Path, 10, 19
 SQL Server, 15

Microsoft SQL Server Enterprise Edition, 17
 migrating data with DMS, 84. *See also* Data Migration tool
 MLFlow platform, 26
 MLOPs (machine learning operations), 26
 modern data warehousing. *See also* data warehouse
 architecture
 architecture and workload, 207–208
 Azure Databricks, 217–229
 Azure HDInsight, 209–217
 Azure Synapse Analytics, 229–232
 defined, 204
 services, 208–209
 MongoDB API, 149–150
 monitor logs, 109
 mounting external storage, 227–229
 MPP (massively parallel processing) architectures, 12, 35,
 72, 205
 multidimensional databases, 69. *See also* database
 schema versus semantic model
 MySQL, 82–83, 131

N

names, aliasing, 128
 NetMon (Network Monitor) tool, 191
 network security, 111–112, 192. *See also* connectivity
 issues
 NFS v3, 158
 non-relational data. *See also* relational data
 authentication, 186–189, 193–194
 authorization, 189–190
 Azure data services for, 144
 characteristics, 136
 choosing data stores, 142
 columnar data store, 139–140
 connectivity issues, 190–193
 data protection, 185–186
 deploying Azure Cosmos DB, 176
 document store, 138–139
 firewall rules, 182–183
 Free Tier account, 177

non-relational data (continued)

- non-relational data (*continued*)
 - graph store, 140–141
 - identifying data services for, 144
 - key-value store, 137
 - object data store, 142
 - provisioning and deployment, 175
 - reasons for using, 143
 - secure transfer, 183–184
 - storage data encryption, 184–185
 - time series store, 141
 - TLS (Transport Layer Security) version, 184
- non-relational workload, 205
- NoSQL databases, 15, 136–142
- numbers data type, 55
- nvarchar (nchar) (National CHARacters), 55

O

- object data store, 142, 144
- OData specification, Azure Table storage API, 159–160
- ODBC (Open Database Connectivity), 14
- Office 365 SaaS, 62
- OLAP (online analytical processing), 19, 22, 52
- OLTP (online transaction processing), 22, 48–51
- on-demand data analysis, 13. *See also* data analytics
- ONNX (Open Neural Network Exchange), 28
- operating systems and SQL versions, 75
- ORDER BY, 127, 130

P

- PaaS (platform as a service), 61–62
- paginated reporting, 297–299. *See also* reports
- paginated reports, 42
- partitioning, 30
- pie chart, using in reports, 39–40
- pinning objects on dashboards, 294–295
- pipeline
 - checking wealth of, 5
 - creating and running, 250–254

- PoCs (proofs of concept), 207
- policies and data protection, 186
- PolyBase T-SQL query language. *See also* Azure Synapse Analytics
 - configuring for Azure Blob storage, 268–269
 - configuring in Azure Synapse Analytics, 269
 - using, 14, 71–72
- pool tables, 71
- PostgreSQL databases, 79–81, 129–131
- Power BI Desktop, downloading, 280
- Power BI Report Builder, 298
- Power BI Report Server, 299
- Power BI service. *See also* BI (business intelligence) projects; dashboards
 - connectors, 281–283
 - dashboards, 294–297
 - data alerting/notifications, 295
 - data modeling, 280
 - Data view, 285
 - dual tables, 283
 - exploring data, 296
 - Get Data window, 282
 - interactive reports, 279–293
 - Model view, 284, 287
 - overview, 278
 - paginated reporting, 297–299
 - plans and features, 291
 - portal, 292
 - Power Query Editor, 286, 288
 - publishing and sharing, 293
 - reading view, 293
 - release of, 42
 - Report view, 289–291
 - workflow, 279
- PowerShell
 - and Azure Storage, 178–179
 - commands for Cosmos DB, 179
 - creating Cosmos DB account from, 177
 - managing deployment, 103–105
 - storage account and container, 180–182
- PowerShell Azure library, using with blob storage, 168–170

- predictive analysis, 24
- prescriptive analysis, 24
- primary key, 21
- private and public preview, 276
- private connections, using, 192–193
- procedures, relational databases, 58
- process rate versus input rate, 5
- producers, 3
- PySpark
 - file schema in, 265
 - joining data in, 267
 - source data in, 266
 - writing data to files, 267–268

Q

- Query Editor, 115
- Query Performance Insight, 64
- query techniques, SQL language, 122–126
- query tools, identifying, 114–122
- querying data
 - MySQL, 131
 - PostgreSQL databases, 129–131
 - SQL Server databases, 126–129

R

- raw data, 4
- RBAC (role-based access control), 171
- regex (regular expressions), 17
- relational Azure data services
 - IaaS (infrastructure as a service), 60
 - overview, 58–59
 - PaaS (platform as a service), 61
 - SaaS (software as a service), 61–62
- relational data. *See also* non-relational data
 - characters, 55
 - columns and constraints, 53–55
 - deployment of services, 87–90
 - indexes, 55–56

- numbers, 55
- procedures, 58
- provisioning, 87–90
- querying, 126–131
- tables, 53–55
- theory and practice, 20–22
- types of, 55
- views, 56–58
- relational data services, provisioning and deploying, 87–90
- relational workload, 204–205
- RENAME command, 125
- replication and data protection, 185
- reporting access, 37
- reports, visuals used in, 38–41. *See also* paginated reporting; SSRS (SQL Server Reporting Services)
- resource errors, 114
- resource groups, 92
- resource providers, ARM, 95–96
- resources
 - increasing and reducing, 60
 - overuse errors, 114
- retention period, 5
- road vehicle trips analysis, 4
- routing preferences, non-relational data, 183

S

- SaaS (software as a service), 61–62
- scatter chart, using in reports, 40
- schema drift, 14
- secure transfer, non-relational data, 183–184
- security. *See* data security components
- SELECT statement, 126–128
- semantic model versus database schema, 50
- semi-structured data, 15
- sentiment analysis, 16
- servers, registering, 174
- serving layer, 42
- session window, 9–10
- shared resource model, 66
- single source of truth, 22

sliding window

- sliding window, 8–9
- SLOs (service level objectives), 73–74
- SMB (Server Message Block) protocol, Azure File storage, 170
- SMEs (subject matter experts), 23
- SMP (symmetric multiprocessing) systems, 12
- soundex, 17
- Spark, 35–36, 268. *See also* Azure Databricks
- speed layer versus batch layer, 14
- SQL (Structured Query Language). *See also* Azure SQL Database
 - operators, 21
 - query techniques, 122–131
- SQL DW (Datawarehouse), 71
- SQL Server
 - adding, 69
 - Always On availability groups, 84
 - analysis services, 69–74
 - avoiding syntax errors, 111
 - on Azure VM (virtual machine), 74–79, 89–90
 - in-engine prediction, 27–28
 - implementing inside VM, 74–79
 - querying data, 126–129
- SQL Server Reporting Services, 297
- SQL versions and operating systems, 75
- sqlcmd utility, 115–116
- SQL-MI (Managed Instance), 83–86, 89
- SSIS (SQL Server Integration Services) Enterprise Edition, 17, 30–33
- SSIS workloads, migrating, 238
- SSMS (SQL Server Management Studio), 110–111, 113, 118–121, 124
- SSOT (single source of truth), 208
- SSRS (SQL Server Reporting Services), 42. *See also* reports
- stacked column chart, using in reports, 38–39
- statistics, use of, 57
- storage, 4. *See also* Azure Files storage

- storage account
 - Azure portal, 95
 - and container using PowerShell, 180–182
 - creating, 158
- storage data encryption, non-relational data, 184–185
- storage devices, 144
- Storage Explorer, 161–162, 166–167
- stream processing overview, 4
- stream processing pipeline, 3
- stream window aggregation, 6
- streaming
 - data sets, 294
 - workload, 206
- streaming data
 - versus batch data, 19–20
 - use case, 4
- structured data, 15

T

- table visual, using in reports, 38
- tables, relational database structures, 53–55. *See also* Azure Table storage
- tabular models, 71
- TCO (total cost of ownership), 58
- TDE (Transparent Data Encryption), 108
- TDSP (Team Data Science Process), 24–26
- Template URI, using for ARM templates, 181
- threat protection, 109
- time series store, 141, 144
- time window aggregations, 5–6
- TLS (Transport Layer Security), 184
- transformations
 - and aggregation, 4
 - performing, 34
 - processes, 53
- T-SQL (Transact Structured Query Language), 21, 126–129
- T-SQL query language, PolyBase, 14
- tumbling window, 7
- tuples, 21

U

- unified analytics platform, 217
- unstructured data, 16
- Update structure, 125

V

- value and big data, 18
- variety and big data, 14–16
- vCore-based purchasing model, 64–67
- velocity and big data, 14
- veracity and big data, 17–18
- views, relational database structures, 56–58
- Visual Studio Cloud Explorer, 197–198
- visualization, 18
- VM (virtual machine)
 - implementing SQL Server in, 74–79
 - templates, 61
- volatility of data, 14
- volume and batch data, 12–14
- VPNs (virtual private networks), 192

W

- WASB or ABFS drivers, 226
- watermarks, 5–6
- WHERE predicate, 127
- Wireshark network analyzer, 191
- Word document, structure of, 136
- workloads, types of, 2

X

- XML (Extensible Markup Language) documents, 15, 138

Z

- ZRS (zone-redundant storage), 157