CHAPTER 3

Security

Fifteen percent (15%) of the *DB2 10.1 Fundamentals* certification exam (Exam 610) is designed to test your knowledge of the mechanisms DB2 uses to protect data and database objects against unauthorized access and modification. The questions that make up this portion of the exam are intended to evaluate the following:

- Your ability to identify ways in which access to instances, databases, and user data can be restricted
- Your ability to identify the authorization levels and privileges that are available with DB2
- Your ability to identify how specific authorizations and privileges are given (granted) to others
- Your ability to identify how specific authorizations and privileges are taken away (revoked) from others
- Your knowledge of how to define and use roles
- Your knowledge of how to use Row and Column Access Control (RCAC) to prevent unauthorized access to sensitive data
- Your knowledge of how and when to use trusted contexts

This chapter introduces you to the authorizations and privileges that are available with DB2. It also provides you with information about the tools that are used to give

(grant) and take away (revoke) authorizations and privileges to/from individuals, groups, and roles.

In this chapter, you will become familiar with the two mechanisms that DB2 uses to control access to instances, databases, data objects, and data: *authorities* and *privileges*. You will also learn how to grant authorities and privileges to specific users, groups, and roles, as well as how to revoke authorities and privileges when it is appropriate to do so. And you will learn how to use tools like RCAC and Label-Based Access Control (LBAC) to secure sensitive data in ways that meet the strictest of security requirements or that adhere to rigid government security standards.

Controlling Database Access

Identity theft—a crime in which someone wrongfully obtains another person's personal data (such as a Social Security number, bank account number, or credit card number) and uses it in a fraudulent or deceptive manner for economic gain—is the fastest-growing crime in our nation today. Criminals obtain the information needed to steal an identity in a variety of ways: through overheard conversations made on cell phones; through telephone and email "phishing" scams; by stealing wallets, purses, and personal mail; by taking discarded documents from the trash; and by exploiting careless online shopping and banking habits. If that is not frightening enough, studies show that up to 70 percent of all identity theft cases are "inside jobs"—that is, they are perpetrated by a co-worker or by an employee of a business that individuals frequently patronize. In these cases, all that is needed to commit identity theft is access to a company database.

That is why it is essential that a relational database management system be able to protect data against unauthorized access and modification. With DB2, a combination of external security services and internal access controls are used to perform this vital task. Furthermore, three different layers of security are employed: the first controls access to the instance a database was created under, the second controls access to the database itself, and the third controls who has access to the data and data objects that reside within the database.

Authentication

The first security portal most users must pass through on their way to gaining access to a DB2 instance or database is a process known as *authentication*. The purpose of authentication is to verify that users really are who they say they are. And in most cases, an external security facility that is not part of DB2 is used to perform this task. This

facility might be part of the operating system, which is the case when DB2 is deployed on Linux, AIX, Solaris, HP-UX, and recent versions of the Windows operating system. Or it can be a separate add-on product such as Distributed Computing Environment (DCE) Security Services. In either case, the external security facility used often must be presented with two specific pieces of information before a user can be authenticated: a unique *user ID* and a corresponding *password*. The user ID identifies the user to the security facility, and the password, which is information that is supposedly known only by the user and the security facility, verifies that the user is indeed who he or she claims to be.

Important: Because passwords are an important tool for authenticating users, they should always be required at the operating system level if an operating system will be used to perform authentication. Keep in mind that on most UNIX operating systems, undefined passwords are treated as NULL, and any user who has not been assigned a password will be treated as having a NULL password. Thus, from the operating system's perspective, if no password is provided when a user with a NULL password attempts to log in, authentication will be deemed successful, and the user will be given access to the operating system as well as to DB2.

Where Authentication Takes Place

Because DB2 can reside in environments that consist of multiple clients, gateways, and servers (each of which can be running a different operating system), deciding where authentication is to take place can sometimes be challenging. For this reason, DB2 often relies on the security facility that either the client or the server's operating system provides to control how users are authenticated.

With DB2 for Linux, UNIX, and Windows, a parameter in each DB2 Database Manager configuration file (which is a file that is associated with every instance) controls where and how authentication takes place. The value assigned to this parameter, often referred to as the *authentication type*, is set initially when an instance is first created. (On the server side, the authentication type is specified during the instance creation process; on the client side, the authentication type is stipulated when a remote database is cataloged.) Only one authentication type exists for each instance, and it controls

access to that instance, as well as to all databases that fall under that instance's control. The following authentication types are available with DB2 10.1 for Linux, UNIX, and Windows:

- **CLIENT**: Authentication occurs at the client workstation or database partition where a client application is invoked, using the security facility that the client's operating system provides (assuming one is available). The user ID and password supplied by users wishing to access an instance or database are compared with the user ID and password combinations stored at the client or node to determine whether access is permitted.
- **SERVER**: Authentication occurs at the server workstation using the security facility that the server's operating system provides. The user ID and password supplied by users wishing to access an instance or database are compared with the user ID and password combinations stored at the server to determine whether access is permitted. Unless otherwise specified, this is the default authentication type used.
- **SERVER_ENCRYPT**: Authentication occurs at the server workstation using the security facility that the server's operating system provides. However, the password supplied by users wishing to access an instance or database can be encrypted at the client workstation before it is sent to the server for validation.
- **DATA_ENCRYPT**: Authentication occurs at the server workstation using the SERVER_ENCRYPT authentication method. In addition, all user data is encrypted before it is passed from the client to the server and vice versa.
- **DATA_ENCRYPT_CMP**: Authentication occurs at the server workstation using the SERVER_ENCRYPT authentication method. And all user data is encrypted before it is passed from the client to the server and vice versa. In addition, compatibility for down-level products that do not support the DATA_ENCRYPT authentication type is provided. (Such products connect using the SERVER_ENCRYPT authentication type, and user data is not encrypted.)
- **KERBEROS**: Authentication occurs at the server workstation using a security facility that supports the Kerberos security protocol. This protocol performs authentication as a third-party service by using conventional cryptography to create a shared secret key—the key becomes the credentials used to verify the user's identity whenever local or network services are requested. (This eliminates the need to pass a user ID and password across the network as ASCII text.) If both the client and the

server support the Kerberos security protocol, the user ID and password provided by users wishing to access an instance or database are encrypted at the client workstation and sent to the server for validation.

- **KRB_SERVER_ENCRYPT**: Authentication occurs at the server workstation using either the KERBEROS or the SERVER_ENCRYPT authentication method. If the client's authentication type is set to KERBEROS, authentication takes place at the server using the Kerberos security system; if the client's authentication type is set to anything other than KERBEROS or if the Kerberos authentication service is unavailable, the server acts as if the SERVER_ENCRYPT authentication type was specified, and the rules for that authentication method are applied.
- **GSSPLUGIN**: Authentication occurs at the server workstation using a Generic Security Service Application Program Interface (GSS-API) plug-in. If the client's authentication type is not specified, the server returns a list of server-supported plug-ins to the client. (This list is stored in the *srvcon_gssplugin_list* database manager configuration parameter.) The client selects and uses the first plug-in it finds that is identified in the supported list; if no supported plug-in is found, the client is authenticated using the KERBEROS authentication method.
- **GSS_SERVER_ENCRYPT**: Authentication occurs at the server workstation using either the GSSPLUGIN or the SERVER_ENCRYPT authentication method. That is, if client authentication occurs through a GSS-API plug-in, the client is authenticated using the first client-supported plug-in found in the list of server-supported plug-ins supplied. However, if the client does not support any plug-ins in this list, it is authenticated by using the KERBEROS authentication method—if the client does not support the Kerberos security protocol, it is authenticated using the SERVER_ENCRYPT authentication method instead.

It is important to note that if the authentication type a client workstation employs will encrypt user ID and password information before sending it to a server, the server must use an authentication type that can decipher this information. Otherwise, the encrypted data cannot be processed, and an error will result.

Authorities and Privileges

After a user has been authenticated and an attachment to an instance (or a connection to a database) has been established, DB2 evaluates the set of *authorities* and *privileges* that have been assigned to the user to determine which operations, if any, he or she is allowed to perform. Authorities convey the right to perform high-level administrative and

maintenance/utility operations on an instance or a database. Privileges, on the other hand, convey the right to perform certain actions against specific database resources (such as tables, indexes, and views).

Together, authorities and privileges control access to an instance, to one or more databases under a specific instance's control, to a database, and to a database's data objects and data. Users can work only with instances, databases, and objects they have been given authorization for—that is, only if they possess the specific authority or privilege needed.

Administrative Authorities

An *administrative authority* is a set of related privileges that controls which administrative and maintenance operations a user can perform against a DB2 instance or database. Individuals who have been given (granted) administrative authority are responsible both for controlling an instance or database and for ensuring the safety and integrity of any data that might come under that instance/database's control. The following administrative authorities are available:

- **System Administrator (SYSADM) authority:** The highest level of administrative authority available; users who have been granted this authority can run most DB2 utilities, execute most DB2 commands, and perform any SQL or XQuery operation that does not attempt to access data that is protected by RCAC or LBAC. Users with this authority also have the ability to create databases and database objects such as tables, indexes, and views.

  Individuals who hold SYSADM authority are implicitly given all the rights that are granted to users who hold any of the other system-level administrative authorities available.
- **Installation System Administrator (Installation SYSADM) authority:** Assigned to a limited number of users when DB2 is first installed, this authority conveys the same set of abilities that SYSADM authority provides. However, unlike with SYSADM authority, information about who holds Installation SYSADM authority is not stored in the system catalog. Consequently, users with this authority can perform recovery operations when the system catalog for a database is inaccessible or unavailable.
- **System Control (SYSCTRL) authority:** The highest level of system and instance control authority available; users who have been granted this authority can create and drop DB2 databases, use almost all of the DB2 utilities, and execute the majority of the DB2 commands available. However, they cannot access user data

APPENDIX B

Practice Questions

Welcome to the section that really makes this book unique. In my opinion, one of the best ways to prepare for the *DB2 10.1 Fundamentals* certification exam (Exam 610) is by answering practice questions that are similar to, and are presented in the same format as, the questions you will see when you take the actual exam. In this last part of the book, you will find 150 practice questions, as well as comprehensive answers for each question. (It's not enough to know *which* answer is correct; it's also important to know *why* a particular answer is correct—and why the other choices are wrong!)

All of the questions presented here were developed using copious notes that were taken during the exam development process. (As a member of the team that developed the *DB2 10.1 Fundamentals* certification exam, I had the opportunity to see every question that was created for this exam!) I trust you will find these practice questions helpful.

Roger E. Sanders

Planning

Question 1

A database will be used primarily to identify sales patterns for products sold within the last three years and to summarize sales by region, on a quarterly basis. Which type of system is needed?

- ❍ A. Analytical
- ❍ B. DB2 pureScale
- ❍ C. Data warehouse
- ❍ D. Online transaction processing (OLTP)

Question 2

Which product can be used to tune performance for a single query?

- ❍ A. IBM Data Studio
- ❍ B. IBM Control Center
- ❍ C. IBM Data Administrator
- ❍ D. IBM Workload Manager

Question 3

Which two DB2 products are suitable for very large data warehouse applications? (Choose two.)

- ❒ A. DB2 for i
- ❒ B. DB2 for AIX
- ❒ C. DB2 for z/OS
- ❒ D. DB2 pureScale
- ❒ E. DB2 Express-C

C

APPENDIX

Answers to Practice Questions

Planning

Question 1

The correct answer is **C**. Data warehouses are typically used to store and manage large volumes of data that is often historical in nature and that is used primarily for analysis. Thus, a data warehouse could be used to identify sales patterns for products sold within the past three years or to summarize sales by region, on a quarterly basis.

Online transaction processing (OLTP) systems (*Answer D*), on the other hand, are designed to support day-to-day, mission-critical business activities such as web-based order entry and stock trading.

IBM offers two solutions that are tailored specifically for one system workload type or the other: *InfoSphere Warehouse* for data warehouses and the *DB2 pureScale Feature* (*Answer B*) for OLTP workloads.

Analytical workloads (*Answer A*) are better handled by a specialized product known as DB2 for i and by IBM BLU Acceleration, which is currently available only with DB2 10.5 for Linux, UNIX, and Windows.

Question 2

The correct answer is **A**. IBM Data Studio is an Eclipse-based, integrated development environment (IDE) that can be used to perform instance and database administration, routine (SQL procedures, SQL functions, etc.) and application development, and performance-tuning tasks. It replaces the DB2 Control Center (*Answer B*) as the standard GUI tool for DB2 database administration and application development.

IBM Workload Manager, or WLM (*Answer D*) is a comprehensive workload management feature that can help identify, manage, and control database workloads to maximize database server throughput and resource utilization.

There is no such product as IBM Data Administrator (*Answer C*).

Question 3

The correct answers are **B** and **C**. DB2 for z/OS is a multiuser, full-function database management system that has been designed specifically for z/OS, IBM's flagship mainframe operating system. Tightly integrated with the IBM mainframe, DB2 for z/OS leverages the strengths of System z 64-bit architecture to provide, among other things, the ability to support complex data warehouses.

In addition to DB2 for z/OS, all of the DB2 Editions available *except* DB2 Express-C (*Answer E*) and DB2 Express Edition can be used to create data warehouse and OLTP environments. However, IBM offers two solutions that are tailored specifically for one workload type or the other: *InfoSphere Warehouse* for data warehousing workloads and the *DB2 pureScale Feature* (*Answer D*) for OLTP workloads.

DB2 for i (*Answer A*), formerly known as DB2 for i5/OS™, is an advanced, 64-bit Relational Database Management System that leverages the high performance, virtualization, and energy efficiency features of IBM's Power Systems; its self-managing attributes, security, and built-in analytical processing functions make DB2 for i an ideal database server for applications that are analytical in nature.

Question 4

The correct answer is **B**. DB2 Workload Manager (WLM) is a comprehensive workload management feature that can help identify, manage, and control database workloads to maximize database server throughput and resource utilization; with DB2 Workload Manager, it is possible to customize execution environments so that no single workload can control and consume all of the system resources available. (This prevents any one department or service class from overwhelming the system.)

IBM InfoSphere Optim Performance Manager Extended Edition can be used to identify, diagnose, solve, and prevent performance problems in DB2 products and associated applications (*Answer A*). The Self-Tuning Memory Manager (STMM) responds to significant changes in a database's workload