1. Record Nr. UNINA9910556881603321

Autore	Haines Scott
Titolo	Modern data engineering with Apache Spark : a hands-on guide for building mission-critical streaming applications / / Scott Haines
Pubbl/distr/stampa	[Place of publication not identified] : , : Apress, , [2022] ©2022
ISBN	1-4842-7452-0
Descrizione fisica	1 online resource (595 pages) : illustrations
Disciplina	006.312
Soggetti	Data mining
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Includes index.
Nota di contenuto	Intro -- Table of Contents -- About the Author -- About the Technical Reviewer -- Acknowledgments -- Introduction -- Part I: The Fundamentals of Data Engineering with Spark -- Chapter 1: Introduction to Modern Data Engineering -- The Emergence of Data Engineering -- Before the Cloud -- Automation as a Catalyst -- The Cloud Age -- The Public Cloud -- The Origins of the Data Engineer -- The Many Flavors of Databases -- OLTP and the OLAP Database -- The Trouble with Transactions -- Analytical Queries -- No Schema. No Problem. The NoSQL Database -- The NewSQL Database -- Thinking about Tradeoffs -- Cloud Storage -- Data Warehouses and the Data Lake -- The Data Warehouse -- The ETL Job -- The Data Lake -- The Data Pipeline Architecture -- The Data Pipeline -- Workflow Orchestration -- The Data Catalog -- Data Lineage -- Stream Processing -- Interprocess Communication -- Network Queues -- From Distributed Queues to Repayable Message Queues -- Fault-Tolerance and Reliability -- Kafka's Distributed Architecture -- Kafka Records -- Brokers -- Why Stream Processing Matters -- Summary -- Chapter 2: Getting Started with Apache Spark -- The Apache Spark Architecture -- The MapReduce Paradigm -- Mappers -- Durable and Safe Acyclic Execution -- Reducers -- From Data Isolation to Distributed Datasets -- The Spark Programming Model -- Did You Never Learn to Share? -- The Resilient Distributed Data Model -- The Spark Application Architecture -- The Role of the Driver Program

-- The Role of the Cluster Manager -- Bring Your Own Cluster -- The Role of the Spark Executors -- The Modular Spark Ecosystem -- The Core Spark Modules -- From RDDs to DataFrames and Datasets -- Getting Up and Running with Spark -- Installing Spark -- Downloading Java JDK -- Downloading Scala -- Downloading Spark -- Taking Spark for a Test Ride -- The Spark Shell.
Exercise 2-1: Revisiting the Business Intelligence Use Case -- Defining the Problem -- Solving the Problem -- Problem 1: Find the Daily Active Users for a Given Day -- Problem 2: Calculate the Daily Average Number of Items Across All User Carts -- Problem 3: Generate the Top Ten Most Added Items Across All User Carts -- Exercise 2-1: Summary -- Summary -- Chapter 3: Working with Data -- Docker -- Containers -- Docker Desktop -- Configuring Docker -- Apache Zeppelin -- Interpreters -- Notebooks -- Preparing Your Zeppelin Environment -- Running Apache Zeppelin with Docker -- Docker Network -- Docker Compose -- Volumes -- Environment -- Ports -- Using Apache Zeppelin -- Binding Interpreters -- Exercise 3-1: Reading Plain Text Files and Transforming DataFrames -- Converting Plain Text Files into DataFrames -- Peeking at the Contents of a DataFrame -- DataFrame Transformation with Pattern Matching -- Exercise 3-1: Summary -- Working with Structured Data -- Exercise 3-2: DataFrames and Semi-Structured Data -- Schema Inference -- Using Inferred Schemas -- Using Declared Schemas -- Steal the Schema Pattern -- Building a Data Definition -- All About the StructType -- StructField -- Spark Data Types -- Adding Metadata to Your Structured Schemas -- Exercise 3-2: Summary -- Using Interpreted Spark SQL -- Exercise 3-3: A Quick Introduction to SparkSQL -- Creating SQL Views -- Using the Spark SQL Zeppelin Interpreter -- Computing Averages -- Exercise 3-3: Summary -- Your First Spark ETL -- Exercise 3-4: An End-to-End Spark ETL -- Writing Structured Data -- Parquet Data -- Reading Parquet Data -- Exercise 3-4: Summary -- Summary -- Chapter 4: Transforming Data with Spark SQL and the DataFrame API -- Data Transformations -- Basic Data Transformations -- Exercise 4-1: Selections and Projections -- Data Generation -- Selection -- Filtering -- Projection.
Exercise 4-1: Summary -- Joins -- Exercise 4-2: Expanding Data Through Joins -- Inner Join -- Right Join -- Left Join -- Semi-Join -- Anti-Join -- Semi-Join and Anti-Join Aliases -- Using the IN Operator -- Negating the IN Operator -- Full Join -- Exercise 4-2: Summary -- Putting It All Together -- Exercise 4-3: Problem Solving with SQL Expressions and Conditional Queries -- Expressions as Columns -- Using an Inner Query -- Using Conditional Select Expressions -- Exercise 4-3: Summary -- Summary -- Chapter 5: Bridging Spark SQL with JDBC -- Overview -- MySQL on Docker Crash Course -- Starting Up the Docker Environment -- Docker MySQL Config -- Exercise 5-1: Exploring MySQL 8 on Docker -- Working with Tables -- Connecting to the MySQL Docker Container -- Using the MySQL Shell -- The Default Database -- Creating the Customers Table -- Inserting Customer Records -- Viewing the Customers Table -- Exercise 5-1: Summary -- Using RDBMS with Spark SQL and JDBC -- Managing Dependencies -- Exercise 5-2: Config-Driven Development with the Spark Shell and JDBC -- Configuration, Dependency Management, and Runtime File Interpretation in the Spark Shell -- Runtime Configuration -- Local Dependency Management -- Runtime Package Management -- Dynamic Class Compilation and Loading -- Spark Config: Access Patterns and Runtime Mutation -- Viewing the SparkConf -- Accessing the Runtime Configuration -- Iterative Development with the Spark Shell -- Describing Views and Tables -- Writing DataFrames to External MySQL Tables -- Generate Some New

Customers -- Using JDBC DataFrameWriter -- SaveMode -- Exercise 5-2: Summary -- Continued Explorations -- Good Schemas Lead to Better Designs -- Write Customer Records with Minimal Schema -- Deduplicate, Reorder, and Truncate Your Table -- Drop Duplicates -- Sorting with Order By -- Truncating SQL Tables -- Stash and Replace. Summary -- Chapter 6: Data Discovery and the Spark SQL Catalog -- Data Discovery and Data Catalogs -- Why Data Catalogs Matter -- Data Wishful Thinking -- Data Catalogs to the Rescue -- The Apache Hive Metastore -- Metadata with a Modern Twist -- Exercise 6-1: Enhancing Spark SQL with the Hive Metastore -- Configuring the Hive Metastore -- Create the Metastore Database -- Connect to the MySQL Docker Container -- Authenticate as the root the MySQL User -- Create the Hive Metastore Database -- Grant Access to the Metastore -- Create the Metastore Tables -- Authenticate as the dataeng User -- Switch Databases to the Metastore -- Import the Hive Metastore Tables -- Configuring Spark to Use the Hive Metastore -- Configure the Hive Site XML -- Configure Apache Spark to Connect to Your External Hive Metastore -- Using the Hive Metastore for Schema Enforcement -- Production Hive Metastore Considerations -- Exercise 6-1: Summary -- The Spark SQL Catalog -- Exercise 6-2: Using the Spark SQL Catalog -- Creating the Spark Session -- Spark SQL Databases -- Listing Available Databases -- Finding the Current Database -- Creating a Database -- Loading External Tables Using JDBC -- Listing Tables -- Creating Persistent Tables -- Finding the Existence of a Table -- Databases and Tables in the Hive Metastore -- View Hive Metastore Databases -- View Hive Metastore Tables -- Hive Table Parameters -- Working with Tables from the Spark SQL Catalog -- Data Discovery Through Table and Column-Level Annotations -- Adding Table-Level Descriptions and Listing Tables -- Adding Column Descriptions and Listing Columns -- Caching Tables -- Cache a Table in Spark Memory -- The Storage View of the Spark UI -- Force Spark to Cache -- Uncache Tables -- Clear All Table Caches -- Refresh a Table -- Testing Automatic Cache Refresh with Spark Managed Tables -- Removing Tables -- Drop Table.
Conditionally Drop a Table -- Using Spark SQL Catalyst to Remove a Table -- Exercise 6-2: Summary -- The Spark Catalyst Optimizer -- Introspecting Spark's Catalyst Optimizer with Explain -- Logical Plan Parsing -- Logical Plan Analysis -- Unresolvable Errors -- Logical Plan Optimization -- Physical Planning -- Java Bytecode Generation -- Datasets -- Exercise 6-3: Converting DataFrames to Datasets -- Create the Customers Case Class -- Dataset Aliasing -- Mixing Catalyst and Scala Functionality -- Using Typed Catalyst Expressions -- Exercise 6-3: Summary -- Summary -- Chapter 7: Data Pipelines and Structured Spark Applications -- Data Pipelines -- Pipeline Foundations -- Spark Applications: Form and Function -- Interactive Applications -- Spark Shell -- Notebook Environments -- Batch Applications -- Stateless Batch Applications -- Stateful Batch Applications -- From Stateful Batch to Streaming Applications -- Streaming Applications -- Micro-Batch Processing -- Continuous Processing -- Designing Spark Applications -- Use Case: CoffeeCo and the Ritual of Coffee -- Thinking about Data -- Data Storytelling and Modeling Data -- Exercise 7-1: Data Modeling -- The Story -- Breaking Down the Story -- Extracting the Data Models -- Customer -- Store -- Product, Goods and Items -- Vendor -- Location -- Rating -- Exercise 7-1: Summary -- From Data Model to Data Application -- Every Application Begins with an Idea -- The Idea -- Exercise 7-2: Spark Application Blueprint -- Default Application Layout -- README.md -- build.sbt -- conf -- project -- src -- Common Spark Application

	Components -- Application Configuration -- Application Default Config -- Runtime Config Overrides -- Common Spark Application Initialization -- Dependable Batch Applications -- Exercise 7-2: Summary -- Connecting the Dots -- Application Goals. Exercise 7-3: The SparkEventExtractor Application.
Sommario/riassunto	Leverage Apache Spark within a modern data engineering ecosystem. This hands-on guide will teach you how to write fully functional applications, follow industry best practices, and learn the rationale behind these decisions. With Apache Spark as the foundation, you will follow a step-by-step journey beginning with the basics of data ingestion, processing, and transformation, and ending up with an entire local data platform running Apache Spark, Apache Zeppelin, Apache Kafka, Redis, MySQL, Minio (S3), and Apache Airflow.