Study guide for Exam DP-203: Data Engineering on Microsoft Azure

Purpose of this document

This study guide should help you understand what to expect on the exam and includes a summary of the topics the exam might cover and links to additional resources. The information and materials in this document should help you focus your studies as you prepare for the exam.

Useful links	Description
Review the skills measured as of February 6, 2023	This list represents the skills measured AFTER the date provided. Study this list if you plan to take the exam AFTER that date.
Review the skills measured prior to February 6, 2023	Study this list of skills if you take your exam PRIOR to the date provided.
Change log	You can go directly to the change log if you want to see the changes that will be made on the date provided.
How to earn the certification	Some certifications only require passing one exam, while others require passing multiple exams.
Certification renewal	Microsoft associate, expert, and specialty certifications expire annually. You can renew by passing a **free** online assessment on Microsoft Learn.
Your Microsoft Learn profile	Connecting your certification profile to Microsoft Learn allows you to schedule and renew exams and share and print certificates.
Exam scoring and score reports	A score of 700 or greater is required to pass.
Exam sandbox	You can explore the exam environment by visiting our exam sandbox.

Useful links	Description
Request accommodations	If you use assistive devices, require extra time, or need modification to any part of the exam experience, you can request an accommodation.
Take a free Practice Assessment	Test your skills with practice questions to help you prepare for the exam.

Updates to the exam

Our exams are updated periodically to reflect skills that are required to perform a role. We have included two versions of the Skills Measured objectives depending on when you are taking the exam.

We always update the English language version of the exam first. Some exams are localized into other languages, and those are updated approximately eight weeks after the English version is updated. Other available languages are listed in the **Schedule Exam** section of the **Exam Details** webpage. If the exam isn't available in your preferred language, you can request an additional 30 minutes to complete the exam.

Note

The bullets that follow each of the skills measured are intended to illustrate how we are assessing that skill. Related topics may be covered in the exam.

Note

Most questions cover features that are general availability (GA). The exam may contain questions on Preview features if those features are commonly used.

Skills measured as of February 6, 2023

Candidates for this exam should have subject matter expertise in integrating, transforming, and consolidating data from various structured, unstructured, and streaming data systems into a suitable schema for building analytics solutions.

Azure data engineers help stakeholders understand the data through exploration, and they build and maintain secure and compliant data processing pipelines by using different tools and techniques. These professionals use various Azure data services and frameworks to store and produce cleansed and enhanced datasets for analysis. This data store can be designed with different architecture patterns based on business requirements, including modern data warehouse (MDW), big data, or lakehouse architecture.

Azure data engineers also help to ensure that the operationalization of data pipelines and data stores are high-performing, efficient, organized, and reliable, given a set of business requirements and constraints. These professionals help to identify and troubleshoot operational and data quality issues. They also design, implement, monitor, and optimize data platforms to meet the data pipelines.

Candidates for this exam must have solid knowledge of data processing languages, including SQL, Python, and Scala, and they need to understand parallel processing and data architecture patterns. They should be proficient in using Azure Data Factory, Azure Synapse Analytics, Azure Stream Analytics, Azure Event Hubs, Azure Data Lake Storage, and Azure Databricks to create data processing solutions.

- Design and implement data storage (15–20%)
- Develop data processing (40–45%)
- Secure, monitor, and optimize data storage and data processing (30–35%)

Design and implement data storage (15–20%)

Implement a partition strategy

- Implement a partition strategy for files
- Implement a partition strategy for analytical workloads
- Implement a partition strategy for streaming workloads
- Implement a partition strategy for Azure Synapse Analytics
- Identify when partitioning is needed in Azure Data Lake Storage Gen2

Design and implement the data exploration layer

- Create and execute queries by using a compute solution that leverages SQL serverless and Spark cluster
- Implement Azure Synapse Analytics database templates
- Recommend Azure Synapse Analytics database templates
- Push new or updated data lineage to Microsoft Purview
- Browse and search metadata in Microsoft Purview Data Catalog

Develop data processing (40–45%)

Ingest and transform data

- Design and implement incremental loads
- Transform data by using Apache Spark
- Transform data by using Transact-SQL (T-SQL)
- Ingest and transform data by using Azure Synapse Pipelines or Azure Data Factory
- Transform data by using Azure Stream Analytics
- Cleanse data
- Handle duplicate data
- Handle missing data
- Handle late-arriving data
- Split data
- Shred JSON
- Encode and decode data
- Configure error handling for a transformation

- Normalize and denormalize values
- Perform data exploratory analysis

Develop a batch processing solution

- Develop batch processing solutions by using Azure Data Lake Storage, Azure Databricks, Azure Synapse Analytics, and Azure Data Factory
- Use PolyBase to load data to a SQL pool
- Implement Azure Synapse Link and query the replicated data
- Create data pipelines
- Scale resources
- Configure the batch size
- Create tests for data pipelines
- Integrate Jupyter or Python notebooks into a data pipeline
- Upsert data
- Revert data to a previous state
- Configure exception handling
- Configure batch retention
- Read from and write to a delta lake

Develop a stream processing solution

- Create a stream processing solution by using Stream Analytics and Azure Event Hubs
- Process data by using Spark structured streaming
- Create windowed aggregates
- Handle schema drift
- Process time series data
- Process data across partitions
- Process within one partition
- Configure checkpoints and watermarking during processing
- Scale resources
- Create tests for data pipelines
- Optimize pipelines for analytical or transactional purposes
- Handle interruptions
- Configure exception handling
- Upsert data
- Replay archived stream data

Manage batches and pipelines

- Trigger batches
- Handle failed batch loads
- Validate batch loads

- Manage data pipelines in Azure Data Factory or Azure Synapse Pipelines
- Schedule data pipelines in Data Factory or Azure Synapse Pipelines
- Implement version control for pipeline artifacts
- Manage Spark jobs in a pipeline

Secure, monitor, and optimize data storage and data processing (30–35%)

Implement data security

- Implement data masking
- Encrypt data at rest and in motion
- Implement row-level and column-level security
- Implement Azure role-based access control (RBAC)
- Implement POSIX-like access control lists (ACLs) for Data Lake Storage Gen2
- Implement a data retention policy
- Implement secure endpoints (private and public)
- Implement resource tokens in Azure Databricks
- Load a DataFrame with sensitive information
- Write encrypted data to tables or Parquet files
- Manage sensitive information

Monitor data storage and data processing

- Implement logging used by Azure Monitor
- Configure monitoring services
- Monitor stream processing
- Measure performance of data movement
- Monitor and update statistics about data across a system
- Monitor data pipeline performance
- Measure query performance
- Schedule and monitor pipeline tests
- Interpret Azure Monitor metrics and logs
- Implement a pipeline alert strategy

Optimize and troubleshoot data storage and data processing

- Compact small files
- Handle skew in data
- Handle data spill
- Optimize resource management
- Tune queries by using indexers
- Tune queries by using cache

- Troubleshoot a failed Spark job
- Troubleshoot a failed pipeline run, including activities executed in external services

Study resources

We recommend that you train and get hands-on experience before you take the exam. We offer self-study options and classroom training as well as links to documentation, community sites, and videos.

Study resources	Links to learning and documentation
Get trained	Choose from self-paced learning paths and modules or take an instructor led course
Find documentation	Azure Data Lake Storage Azure Synapse Analytics Azure Databricks Data Factory Azure Stream Analytics Event Hubs Azure Monitor
Ask a question	Microsoft Q&A | Microsoft Docs
Get community support	Analytics on Azure | TechCommunity Azure Synapse Analytics | TechCommunity
Follow Microsoft Learn	Microsoft Learn - Microsoft Tech Community
Find a video	Exam Readiness Zone Data Exposed Browse other Microsoft Learn shows

Change log

Key to understanding the table: The topic groups (also known as functional groups) are in bold typeface followed by the objectives within each group. The table is a comparison between the two versions of the exam skills measured and the third column describes the extent of the changes.

Skill area prior to February 6, 2023	Skill area as of February 6, 2023	Changes
Audience profile		Major

Skill area prior to February 6, 2023	Skill area as of February 6, 2023	Changes
Design and implement data storage	**Design and implement data storage**	% of exam decreased
Design a Data Storage Structure		Removed
Design a Partition Strategy	Implement a partition strategy	Major
	Design and implement the data exploration layer	Added
Design the Serving Layer		Removed
Implement Physical Data Storage Structures		Removed
Implement Logical Data Structures		Removed
Implement the Serving Layer		Removed
Design and Develop Data Processing	**Develop data processing**	% of exam increased
Ingest and Transform Data	Ingest and transform data	Major
Design and Develop a Batch Processing Solution	Develop a batch processing solution	Major
Design and Develop a Stream Processing Solution	Develop a stream processing solution	Major
Manage Batches and Pipelines	Manage batches and pipelines	Minor
Design and Implement Data Security		Removed
Design Security for Data Policies and Standards		Removed
Implement Data Security		Removed

Skill area prior to February 6, 2023	Skill area as of February 6, 2023	Changes
Monitor and Optimize Data Storage and Data Processing	**Secure, monitor, and optimize data storage and data processing**	% of exam increased
	Implement data security	Added
Monitor Data Storage and Data Processing	Monitor data storage and data processing	Major
Optimize and Troubleshoot Data Storage and Data Processing	Optimize and troubleshoot data storage and data processing	Major

Skills measured prior to February 6, 2023

Audience profile

Candidates for this exam should have subject matter expertise integrating, transforming, and consolidating data from various structured and unstructured data systems into a structure that is suitable for building analytics solutions.

Azure Data Engineers help stakeholders understand the data through exploration, and they build and maintain secure and compliant data processing pipelines by using different tools and techniques. These professionals use various Azure data services and languages to store and produce cleansed and enhanced datasets for analysis.

Azure Data Engineers also help ensure that data pipelines and data stores are high-performing, efficient, organized, and reliable, given a set of business requirements and constraints. They deal with unanticipated issues swiftly, and they minimize data loss. They also design, implement, monitor, and optimize data platforms to meet the data pipelines needs.

A candidate for this exam must have strong knowledge of data processing languages such as SQL, Python, or Scala, and they need to understand parallel processing and data architecture patterns.

- Design and implement data storage (40–45%)
- Design and develop data processing (25–30%)
- Design and implement data security (10–15%)
- Monitor and optimize data storage and data processing (10–15%)

Design and implement data storage (40–45%)

Design a data storage structure

- Design an Azure Data Lake solution

- Recommend file types for storage
- Recommend file types for analytical queries
- Design for efficient querying
- Design for data pruning
- Design a folder structure that represents the levels of data transformation
- Design a distribution strategy
- Design a data archiving solution

Design a partition strategy

- Design a partition strategy for files
- Design a partition strategy for analytical workloads
- Design a partition strategy for efficiency/performance
- Design a partition strategy for Azure Synapse Analytics
- Identify when partitioning is needed in Azure Data Lake Storage Gen2

Design the serving layer

- Design star schemas
- Design slowly changing dimensions
- Design a dimensional hierarchy
- Design a solution for temporal data
- Design for incremental loading
- Design analytical stores
- Design metastores in Azure Synapse Analytics and Azure Databricks

Implement physical data storage structures

- Implement compression
- Implement partitioning
- Implement sharding
- Implement different table geometries with Azure Synapse Analytics pools
- Implement data redundancy
- Implement distributions
- Implement data archiving

Implement logical data structures

- Build a temporal data solution
- Build a slowly changing dimension
- Build a logical folder structure
- Build external tables
- Implement file and folder structures for efficient querying and data pruning

Implement the serving layer

- Deliver data in a relational star
- Deliver data in Parquet files
- Maintain metadata
- Implement a dimensional hierarchy

Design and develop data processing (25–30%)

Ingest and transform data

- Transform data by using Apache Spark
- Transform data by using Transact-SQL
- Transform data by using Data Factory
- Transform data by using Azure Synapse Pipelines
- Transform data by using Stream Analytics
- Cleanse data
- Split data
- Shred JSON
- Encode and decode data
- Configure error handling for the transformation
- Normalize and denormalize values
- Transform data by using Scala
- Perform data exploratory analysis

Design and develop a batch processing solution

- Develop batch processing solutions by using Data Factory, Data Lake, Spark, Azure Synapse Pipelines, PolyBase, and Azure Databricks
- Create data pipelines
- Design and implement incremental data loads
- Design and develop slowly changing dimensions
- Handle security and compliance requirements
- Scale resources
- Configure the batch size
- Design and create tests for data pipelines
- Integrate Jupyter/Python notebooks into a data pipeline
- Handle duplicate data
- Handle missing data
- Handle late-arriving data
- Upsert data
- Regress to a previous state
- Design and configure exception handling

- Configure batch retention
- Design a batch processing solution
- Debug Spark jobs by using the Spark UI

Design and develop a stream processing solution

- Develop a stream processing solution by using Stream Analytics, Azure Databricks, and Azure Event Hubs
- Process data by using Spark structured streaming
- Monitor for performance and functional regressions
- Design and create windowed aggregates
- Handle schema drift
- Process time series data
- Process across partitions
- Process within one partition
- Configure checkpoints/watermarking during processing
- Scale resources
- Design and create tests for data pipelines
- Optimize pipelines for analytical or transactional purposes
- Handle interruptions
- Design and configure exception handling
- Upsert data
- Replay archived stream data
- Design a stream processing solution

Manage batches and pipelines

- Trigger batches
- Handle failed batch loads
- Validate batch loads
- Manage data pipelines in Data Factory/Synapse Pipelines
- Schedule data pipelines in Data Factory/Synapse Pipelines
- Implement version control for pipeline artifacts
- Manage Spark jobs in a pipeline

Design and implement data security (10–15%)

Design security for data policies and standards

- Design data encryption for data at rest and in transit
- Design a data auditing strategy
- Design a data masking strategy
- Design for data privacy

- Design a data retention policy
- Design to purge data based on business requirements
- Design Azure role-based access control (Azure RBAC) and POSIX-like Access Control List (ACL) for Data Lake Storage Gen2
- Design row-level and column-level security

Implement data security

- Implement data masking
- Encrypt data at rest and in motion
- Implement row-level and column-level security
- Implement Azure RBAC
- Implement POSIX-like ACLs for Data Lake Storage Gen2
- Implement a data retention policy
- Implement a data auditing strategy
- Manage identities, keys, and secrets across different data platform technologies
- Implement secure endpoints (private and public)
- Implement resource tokens in Azure Databricks
- Load a DataFrame with sensitive information
- Write encrypted data to tables or Parquet files
- Manage sensitive information

Monitor and optimize data storage and data processing (10–15%)

Monitor data storage and data processing

- Implement logging used by Azure Monitor
- Configure monitoring services
- Measure performance of data movement
- Monitor and update statistics about data across a system
- Monitor data pipeline performance
- Measure query performance
- Monitor cluster performance
- Understand custom logging options
- Schedule and monitor pipeline tests
- Interpret Azure Monitor metrics and logs
- Interpret a Spark directed acyclic graph (DAG)

Optimize and troubleshoot data storage and data processing

- Compact small files
- Rewrite user-defined functions (UDFs)
- Handle skew in data

- Handle data spill
- Tune shuffle partitions
- Find shuffling in a pipeline
- Optimize resource management
- Tune queries by using indexers
- Tune queries by using cache
- Optimize pipelines for analytical or transactional purposes
- Optimize pipeline for descriptive versus analytical workloads
- Troubleshoot a failed spark job
- Troubleshoot a failed pipeline run