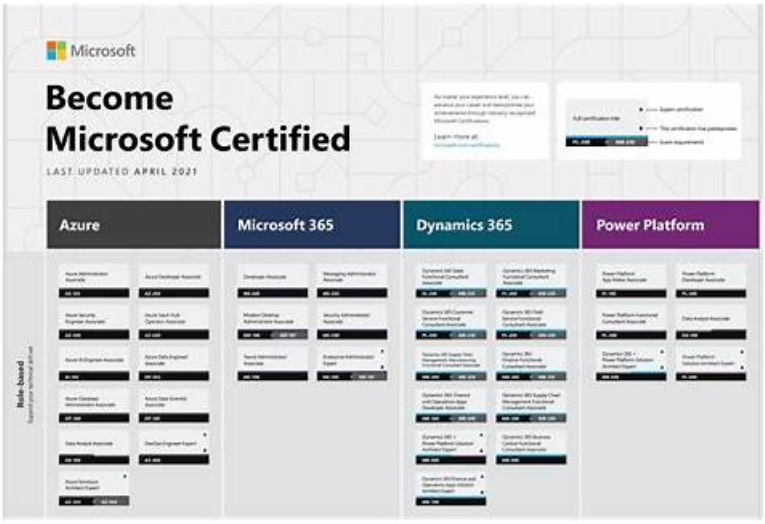
最高なMicrosoftのDP-750認定試験テストソフトウェア



テストが来るのを静かに待っている場合は、目を覚まして、別の方法でDP-750試験を受ける準備ができている必要があります。最近のDP-750ガイド急流の効果が資格試験を通じて受験者の秘密兵器になったことを示した後、DP-750トレーニング資料を勉強して「テストデータ」を書くことがあなたの選択に最適です。 DP-750ガイドトレントのユーザーは、DP-750試験で予期しない結果を得ることができます。

我々はあなたが我々のDP-750問題集を通して試験に合格できるのを保証しています。もし不幸であなたが試験に失敗しましたら、あなたの成績書のスキャンをもらって、我々は全額であなたにDP-750問題集の金額を返金して、あなたの失敗するための経済損失を減少します。

>> DP-750日本語版サンプル <<

DP-750資格勉強 & DP-750無料模擬試験

余分な課税を受けている場合は、DP-750信頼性の高い学習ガイド資料を購入する前に時間内にお知らせください。 キーポイントが情報税である場合があります。 一部の国では、追加情報税の支払いを購入者に要求する場合があります。 Microsoft DP-750の信頼できる学習ガイド教材を購入する際にこの税を回避するにはどうすればよいですか? クレジットカードでPayPalで支払うことを選択できます。 PayPalには追加費用はありません。 ここでは、PayPalアカウントは必要ありません。 クレジットカードは、DP-750の信頼できる学習ガイドを購入するために必要です。

Microsoft Implementing Data Engineering Solutions Using Azure Databricks 認定 DP-750 試験問題 (Q70-Q75):

質問 # 70
What is the best way to reduce Databricks compute cost for intermittent workloads?

- A. Enable autoscaling and auto-termination
- B. Use all-purpose clusters
- C. Use single-node clusters only
- D. Increase cluster size

正解：A

解説：
Autoscaling adjusts resources based on workload demand, and auto-termination shuts down idle clusters, reducing cost significantly. All-purpose clusters are always on. Increasing size raises cost. Single-node clusters limit scalability and are not suitable for production workloads.

質問 # 71
What improves join performance for small lookup tables?

- A. Shuffle join
- B. Cartesian join
- C. Sort merge join
- D. Broadcast join

正解：D

質問 # 72
You need to develop the task logic for a new job in Lakeflow Jobs that processes telemetry data.
Each task must contain only the appropriate logic for its step in the pipeline. The solution must support the planned changes and meet the data ingestion and processing requirements.
What should you do?

- A. Use a single Databricks notebook task that performs ingestion, cleansing, and curation in one script.
- B. Create three tasks that each contains the identical logic and use task retries.
- C. Create separate tasks for ingestion, cleansing, and curation.
- D. Use a single SQL task that performs ingestion, cleansing, and curation by running merge commands.

正解：C

解説：
The correct answer is D. Breaking the pipeline into separate tasks for ingestion, cleansing, and curation is the foundation of well-designed Lakeflow Jobs pipelines. Each task should own one responsibility - when a task does too much, debugging a failure becomes a hunt through unrelated code, and retry logic becomes expensive because you re-execute work that already succeeded. Contoso's planned changes explicitly call for 'a clear execution order and dependencies' and 'orchestrate multi- step ingestion and transformation workflows.' Separate tasks map directly to those goals: Lakeflow Jobs tracks each task's status independently, so if cleansing fails, ingestion doesn't re-run.
Option A bundles everything into one notebook, which means a curation bug forces a full re-ingestion. Option B copies logic three times - any future change must be applied in triplicate, which is a maintenance hazard.
Option C forces everything through SQL MERGE, which is the wrong tool for raw-event ingestion and doesn't address cleansing or schema drift.
Reference: https://learn.microsoft.com/en-us/azure/databricks/jobs/
Topic 1, Contoso Case Study
Overview
Contoso has a single Azure Databricks workspace named Workspace1 in the West US Azure region.
Workspace1 is enabled for Unity Catalog.
Workspace1 contains all-purpose clusters for both development and production workloads.
The company's Azure environment contains:
* In the West US, Central US, and East US Azure regions, Azure event hubs that stream telemetry data and an Azure Data Lake Storage Gen2 account in each region for each hub
* A single Azure SQL database in the West US region that hosts enterprise resource planning (ERP) data
* An Azure Database for PostgreSQL server in the West US region that stores operational maintenance data Company information
Contoso, Inc. is a renewable energy provider that operates solar and wind farms across North America.
Data Environment
Contoso ingests the following operational and business data:
* Telemetry data: More than 40,000 IoT sensors across 28 sites emit JSON telemetry events every few seconds. Each site sends the events to the nearest event hub, which writes the data into the corresponding Data Lake Storage Gen2 account. These files frequently experience schema drift.
* Maintenance logs: Maintenance systems generate historical repair logs, daily incremental updates, technician notes, and unstructured attachments that are stored in the Data Lake Storage Gen2 accounts.
* Operational maintenance data: Structured operational maintenance data is stored on the Azure Database for PostgreSQL server.
* External weather data: Hourly weather forecasts are retrieved from a REST API and written to the Data Lake Storage Gen2 accounts.
* ERP data: Daily CSV extracts of 50 to 100 GB contain equipment metadata, work orders, and purchase order information.
Problem Statements

The company's existing analytics environment has several issues:
Ingestion
* Telemetry pipelines fall behind during peak loads.
* Telemetry ingestion fails when schema drift occurs.
* Streaming pipelines reprocess events after a pipeline restarts.
Compute
* Production and development workloads run on the same all-purpose clusters.
* Production and development workloads do NOT support autoscaling or workload isolation.
Governance
* The ERP data is duplicated across systems and development teams.
* Naming conventions are inconsistent across development teams, regions, and products.
* Ownership of the loT sensors changes over time, and analysts must track the full history of the ownership.
* Occasionally, equipment manufacturers must correct data-entry mistakes in equipment names. Historical values are NOT required.
Pipeline operations
* Pipelines lack resiliency, alerting, and centralized scheduling.
Planned Changes
Contoso plans to implement the following changes:
* Implement scalable data pipeline orchestration.
* Create a managed analytics catalog in Unity Catalog.
* Implement a consistent approach to creating curated datasets.
* Establish a centralized governance model across ingestion, cleansed, and curated layers.
* Grant data engineers access to the ERP tables by using minimal development effort.
* Adopt a compute strategy that isolates production workloads and supports autoscaling.
* Adopt a slowly changing dimension (SCD) approach to address current data modeling issues.
Technical Requirements
Contoso identifies the following environment and compute requirements:
* Ensure that production ingestion workloads run on compute clusters that can scale automatically during telemetry spikes.
* Provide fast and consistent performance for business intelligence (Bl) workloads.
* Prevent development activity from affecting production pipelines.
* Production ingestion workloads must run as scheduled, non-interactive pipelines rather than on shared interactive development clusters.
Contoso identifies the following data ingestion and processing requirements:
* Auto-scale ingestion pipelines to handle bursty workloads.
* Handle schema drift for the maintenance and telemetry data.
* Ingest file-based telemetry data by using minimal operational effort.
* Store all the ingested data in a format that supports incremental processing.
* Support the continuous ingestion of telemetry data from the event hubs by using exactly-once semantics.
* Support the ingestion of the structured maintenance data from the Azure Database for PostgreSQL server.
* Build a new telemetry pipeline that ingests raw events from the event hubs, cleanses the data, and publishes curated tables to Unity Catalog.
* Ensure that the Apache Spark Structured Streaming pipelines reading from the event hubs write the data into a managed Delta table named telemetry.raw_events. The pipelines must support schema drift and resume processing after failures without reprocessing the data.
Contoso identifies the following data modeling and optimization requirements:
* Build curated tables that standardize business logic.
* Overwrite equipment metadata attributes, such as name, manufacturer, model, and commissioning date, when the attributes change. Historical values are NOT required.
Contoso identifies the following pipeline deployment and operation requirements: |