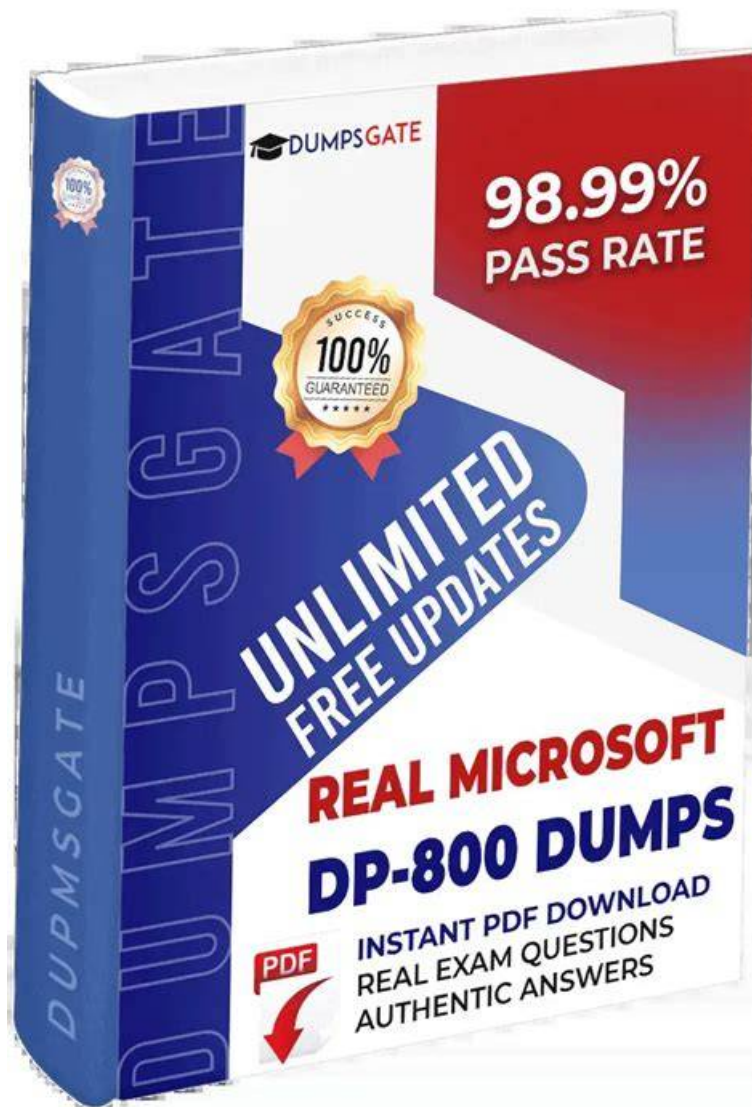


Microsoft DP-800 Desktop-Based Practice Program



Before the clients purchase our DP-800 study materials, they can have a free trial freely. The clients can log in our company's website and visit the pages of our products. The pages of our products lists many important information about our DP-800 study materials and they include the price, version and updated time of our products, the exam name and code, the total amount of the questions and answers, the merits of our DP-800 Study Materials and the discounts. You can have a comprehensive understanding of our DP-800 study materials after you see this information. Then you can look at the free demos and try to answer them to see the value of our DP-800 study materials and finally decide to buy them or not.

Microsoft DP-800 Exam Syllabus Topics:

Section	Weight	Objectives

Topic 1: Secure, optimize, and deploy database solutions	35-40%	- Deploy and maintain solutions • 1. Deploy across SQL Server, Azure SQL, and Microsoft Fabric • 2. Implement CI/CD for databases • 3. Manage updates and versioning - Optimize performance and reliability • 1. Optimize queries and indexes • 2. Ensure high availability and disaster recovery • 3. Monitor and troubleshoot performance - Secure database solutions • 1. Protect data at rest and in transit • 2. Implement authentication and authorization • 3. Implement compliance and auditing
Topic 2: Design and develop database solutions	35-40%	- Develop database solutions • 1. Implement database objects and structures • 2. Implement data integration and transformation • 3. Write and optimize T-SQL code - Design database solutions • 1. Design for maintainability and DevOps integration • 2. Design solutions for structured and semi-structured data • 3. Design for scalability and performance
Topic 3: Implement AI capabilities in database solutions	25-30%	- Integrate AI models and services • 1. Integrate with Azure AI services and Microsoft Foundry • 2. Create and manage external models • 3. Implement vector data and embeddings - Use AI-assisted development tools • 1. Leverage AI tools for T-SQL development • 2. Validate and test AI-enhanced solutions - Build intelligent search and retrieval • 1. Optimize search performance and relevance • 2. Implement semantic search and RAG patterns • 3. Design and implement embedding pipelines

>> Valid DP-800 Test Question <<

TOP Valid DP-800 Test Question 100% Pass | The Best Microsoft Developing AI-Enabled Database Solutions Reliable Dumps Files Pass for sure

The DP-800 prep guide adopt diversified such as text, images, graphics memory method, have to distinguish the markup to learn information, through comparing different color font, as well as the entire logical framework architecture, let users of the DP-800 training dump on the premise of grasping the overall layout, better clues to the formation of targeted long-term memory, and through the cycle of practice, let the knowledge more deeply printed in my mind. The DP-800 Exam Questions are so scientific and reasonable that you can easily remember everything of the DP-800 exam.

Microsoft Developing AI-Enabled Database Solutions Sample Questions

(Q27-Q32):

NEW QUESTION # 27

You have an Azure SQL database that contains the following SQL graph tables:

- A NODE table named dbo.Person
- An EDGE table named dbo.Knows

Each row in dbo.Person contains the following columns:

- PersonID (int)
- DisplayName (nvarchar(100))

You need to use a MATCH operator and exactly two directed Knows relationships to return the PersonID and DisplayName of people that are reachable from the person identified by an input parameter named @StartPersonId.

Which Transact-SQL query should you use?

```
SELECT p3.PersonId, p3.DisplayName
FROM dbo.Person AS p1, dbo.Knows AS k1, dbo.Person AS p2, dbo.Knows AS k2, dbo.Person AS p3
WHERE p1.PersonId = @StartPersonId
AND MATCH(p1-(k1)->p2) AND MATCH(p2-(k2)->p3);
```

- A.
- B.

```
SELECT p3.PersonId, p3.DisplayName
FROM dbo.Person AS p1, dbo.Knows AS k1, dbo.Person AS p2, dbo.Knows AS k2, dbo.Person AS p3
WHERE p1.PersonId = @StartPersonId
AND MATCH(p1-(k1)->p2-(k2)->p3);
```

```
SELECT p3.PersonId, p3.DisplayName FROM dbo.Person AS p1
JOIN dbo.Knows AS k1 ON 1 = 1
JOIN dbo.Person AS p2 ON 1 = 1
JOIN dbo.Knows AS k2 ON 1 = 1
JOIN dbo.Person AS p3 ON 1 = 1
WHERE p1.PersonId = @StartPersonId
AND MATCH(p3<-(k2)-p2<-(k1)-p1);
```

- C.

```
SELECT p2.PersonId, @StartPersonId
FROM dbo.Person AS p1, dbo.Knows AS k1, dbo.Person AS p2, dbo.Knows AS k2, dbo.Person AS p3
WHERE p1.DisplayName = p2.DisplayName
AND MATCH(p1-(k1)->p2-(k2)->p3);
```

- D.

Answer: B

Explanation:

To achieve this using SQL Graph in Azure SQL, you use the MATCH clause to define the path.

Since you need exactly two directed relationships (A → B → C), you must chain the edge table twice in the same direction.

Transact-SQL Statement

SELECT

Person3.ID,

Person3.NameColumn -- Replace with your actual nvarchar(100) column name FROM PersonNodeTable AS Person1,

KnowsEdgeTable AS Knows1, PersonNodeTable AS Person2, KnowsEdgeTable AS Knows2, PersonNodeTable AS Person3

WHERE MATCH(Person1-(Knows1)->Person2-(Knows2)->Person3) AND Person1.ID = @InputID; Use code with caution.

Key Components

MATCH Clause: Defines the traversal pattern. The syntax (Node)-(Edge)->(Node) ensures the relationship is directed.

Chaining: To get "exactly two" steps, you define three node aliases and two edge aliases.

Aliases: Each instance of the table must have a unique alias (e.g., Person1, Person2, Person3) so the engine can distinguish between the different points in the path.

Filtering: The WHERE clause uses your input parameter (@InputID) to set the starting point of the graph traversal Incorrect:

[Not A]

Need three persons, not two.

[Not B]

Do not use JOIN.

[Not C]

Incorrect MATCH statement.

Reference:

<https://learn.microsoft.com/en-us/sql/relational-databases/graphs/sql-graph-sample>

NEW QUESTION # 28

You have an Azure SQL database.

You deploy Data API builder (DAB) to Azure Container Apps by using the `mcr.microsoft.com/azure-databases/data-api-builder:latest` image.

You have the following Container Apps secrets:

- * `MSSQL_CONNECTION_STRING` that maps to the SQL connection string

- * `DAB_CONFIG_BASE64` that maps to the DAB configuration

You need to initialize the DAB configuration to read the SQL connection string.

Which command should you run?

- **A. `dab init --database-type mssql --connection-string "@env('MSSQL_CONNECTION_STRING')" --host-mode Production --config dab-config.json`**
- B. `dab init --database-type mssql --connection-string "@env('DAB_CONFIG_BASE64')" --host-mode Production --config dab-config.json`
- C. `dab init --database-type mssql --connection-string "secretrefmssql-connection-string" --host-mode Production --config dab-config.json`
- D. `dab init --database-type mssql --connection-string "secretrefDAB_CONFIG_BASE64" --host-mode Production --config dab-config.json`

Answer: A

Explanation:

Data API builder supports reading the database connection string from an environment variable by using the syntax:

`@env('MSSQL_CONNECTION_STRING')`

Microsoft's DAB documentation explicitly shows that `@env('MSSQL_CONNECTION_STRING')` tells Data API builder to read the connection string from an environment variable at runtime.

That fits this scenario because Azure Container Apps secrets are typically exposed to the container as environment variables.

Microsoft's Azure Container Apps documentation states that environment variables can reference secrets, and DAB's Azure Container Apps deployment guidance shows a secret being mapped into an environment variable that DAB then reads.

Why the other options are wrong:

- * A and D incorrectly point the connection string to `DAB_CONFIG_BASE64`, which is the config payload secret, not the SQL connection string.

- * C uses `secretref` syntax inside `dab init`, but DAB expects the connection string parameter in the config to use the environment-variable reference syntax `@env(...)`. The `secretref` pattern is for Azure Container Apps environment variable configuration, not for the DAB CLI connection-string argument itself.

So the correct command is:

```
dab init --database-type mssql --connection-string "@env('MSSQL_CONNECTION_STRING')" --host-mode Production --config dab-config.json
```

NEW QUESTION # 29

Case Study 2 - Fabrikam

Existing Environment

Azure Environment

Fabrikam has a single Azure subscription in the East US 2 Azure region. The subscription contains an Azure SQL database named DB1. DB1 contains the following tables:

- * Patients

- * Employees

- * Procedures

- * Transactions

- * UsefulPrompts

- * ProcedureDocuments

You store a column master key as a secret in Azure Key Vault.

You have an on-premises application named TransactionProcessing that uses a hard-coded username and password in a connection string to access DB1.

Problem Statements

Users report that after executing a long-running stored procedure named `sp_UpdateProcedureForPatient`, updates to the underlying data are sometimes inconsistent.

Requirements

Planned Changes

Fabrikam plans to manage all changes to Azure SQL Database objects by using source control in GitHub. Every pull request submitted to production will be validated before it can be merged.

Deployments must use the Release configuration.

Security Requirements

Fabrikam identifies the following security requirements:

- * The `TransactionProcessing` application must use a passwordless connection to DB1.
- * The `Employees` table contains two columns named `TaxID` and `Salary` that must be encrypted at rest.
- * Auditors must have a tamper-evident history of transactions with cryptographic proof of changes to the employee data.

Database Performance Requirements

Records accessed by using `sp_UpdateProcedureForPatient` must NOT be changed by other transactions while the stored procedure runs.

AI Search, Embeddings, and Vector Indexing

Fabrikam identifies the following AI-related requirements:

- * Queries to the `ProcedureDocuments` table must use Reciprocal Rank Fusion (RRF).
- * Users must be able to query the data in DB1 by using prompts in Copilot in Microsoft Fabric.
- * The `UsefulPrompts` table will store prompts that doctors can use to help diagnose patient illness by connecting to an Azure OpenAI endpoint.

Development Requirements

Fabrikam identifies the following development requirements:

- * Provide the functionality to retrieve all the transactions of a given patient between two dates, showing a running total.
- * Expose a Data API builder (DAB) configuration file to enable Azure services to perform the following operations over a REST API:
 - Read data from the `procedures` table without authentication.
 - Read and insert data into the `Transactions` table once authenticated.
 - Execute the `sp_UpdateProcedurePatient` stored procedure.
- * Provide the functionality to retrieve a list of the names of patients who underwent medical procedures during the last 30 days.
- * Information for each medical procedure will be stored in a table. The table will be used with a large language model (LLM) for user querying and will have the following structure.

```
CREATE TABLE dbo.ProcedureDocuments
(
    DocumentId INT IDENTITY PRIMARY KEY,
    SourceId NVARCHAR(200) NULL,
    Content NVARCHAR(MAX) NOT NULL,
    Embedding VECTOR(1536) NOT NULL,
    CreatedAt DATETIME2 NOT NULL DEFAULT SYSUTCDATETIME()
);
```

DAB

You create a DAB configuration file that meets the development requirements for DB1 and includes the following entities.

```

"entities": {
  "Procedures": {
    "source": "dbo.Procedures",
    "rest": true,
    "graphql": true,
    "permissions": [
      {
        "role": "anonymous",
        "actions": [ "read" ]
      }
    ]
  },
  "Transactions": {
    "source": "dbo.Transactions",
    "rest": true,
    "graphql": true,
    "permissions": [
      {
        "role": "authenticated",
        "actions": [ "read", "create" ]
      }
    ]
  },
  "UpdateProcedurePatient": {
    "source": "dbo.sp_UpdateProcedurePatient",
    "rest": {
      "enabled": true,
      "method": "post",
      "path": "/procedurepatient"
    },
    "graphql": false,
    "permissions": [
      {
        "role": "authenticated",
        "actions": [ "execute" ]
      }
    ]
  }
}

```

You implement ProcedureDocuments to support the planned changes.

When users consume data through the Retrieval Augmented Generation (RAG) pattern, they experience data retrieval delays.

You need to improve the data retrieval performance and reduce the number of tokens per retrieval.

What should you implement?

- A. embeddings
- B. a small language model (SLM)
- C. JSON content
- D. chunking

Answer: A

Explanation:

Scenario: Fabrikam identifies the following AI-related requirements: Queries to the ProcedureDocuments table must use Reciprocal Rank Fusion (RRF).

To remedy data retrieval delays in a Retrieval Augmented Generation (RAG) pattern using Reciprocal Rank Fusion (RRF) on an Azure SQL Database table, you should use embeddings.

In a RAG architecture, retrieval delays often stem from inefficient or computationally heavy search processes. While RRF is excellent

for merging results from multiple sources (like combining keyword and vector searches), the core of the speed problem typically lies in how the initial data is indexed and retrieved.

Role of Embeddings

Vector Search Acceleration: Embeddings convert text into high-dimensional vectors. Azure SQL Database can perform similarity searches on these vectors much faster than complex semantic text matching.

Hybrid Search Synergy: RRF is most effective when it fuses results from a keyword search (fast) and a vector search (powered by embeddings). Using embeddings ensures that the "semantic" side of the retrieval is streamlined.

Pre-computation: Since embeddings are generated once during ingestion, the retrieval phase only requires a distance calculation (e.g., Cosine Similarity), which is significantly faster than real-time natural language parsing during each query.

Reference:

<https://pratikbarjatya.medium.com/unlocking-the-power-of-language-with-retrieval-augmented-generation-rag-14123cc275e6>

NEW QUESTION # 30

Hotspot Question

You have an SDK-style SQL database project named `MyDatabaseProject.sqlproj` stored in a private GitHub repository. The repository contains the following GitHub Actions workflow.

```
name: Build and Deploy SQL Project
```

```
on:
```

```
  push:
```

```
    branches:
```

```
      - main
```

```
  workflow_dispatch:
```

```
jobs:
```

```
  build-and-deploy:
```

```
    runs-on: ubuntu-latest
```

```
    permissions:
```

```
      id-token: write
```

```
      contents: read
```

```
    steps:
```

```
      - uses: actions/checkout@v4
```

```
      - name: Build SQL project
```

```
        run: dotnet build MyDatabaseProject.sqlproj
```

```
      - name: Publish DACPAC
```

```
        uses: azure/sql-action@v2
```

```
        with:
```

```
          action: publish
```

```
          path: bin/Debug/MyDatabaseProject.dacpac
```

```
          connection-string: ${ secrets.AZURE_SQL_CONNECTION_STRING }}
```

The repository contains the `AZURE_SQL_CONNECTION_STRING` secrets.

The target is an Azure SQL database that allows access to Azure services and is configured to support mixed authentication.

The workflow runs successfully.

For each of the following statements, select Yes if the statement is true. Otherwise, select No.

NOTE: Each correct selection is worth one point.

Answer Area Microsoft

Statements

Yes

No

The workflow relies on a Microsoft Entra workload identity to access the target Azure SQL database.

☐☐

Changing the Build SQL project step to run:
`dotnet build MyDatabaseProject.sqlproj -c Release` will result in a successful deployment.

☐☐

The workflow can be triggered manually, without making changes to the repository.

☐☐

Answer:

Explanation:

Statements	Yes	No
The workflow relies on a Microsoft Entra workload identity to access the target Azure SQL database.	☐	☒
Changing the Build SQL project step to run: <code>dotnet build MyDatabaseProject.sqlproj -c Release</code> will result in a successful deployment.	☐	☒
The workflow can be triggered manually, without making changes to the repository.	☒	☐

NEW QUESTION # 31

You need to enable similarity search to provide the analysts with the ability to retrieve the most relevant health summary reports. The solution must minimize latency.

What should you include in the solution?

- A. a standard nonclustered index on the Embeddings (vector (1536)) column
- B. a full-text index on the Embeddings (vector (1536)) column
- C. a computed column that manually compares vector values
- D. a vector index on the Embedding* (vector (1536)) column

Answer: D

Explanation:

The correct answer is D because the requirement is to enable similarity search over embedding vectors and to minimize latency. Microsoft documents that CREATE VECTOR INDEX is specifically used to create an index on vector data for approximate nearest neighbor (ANN) search, which is designed to accelerate vector similarity queries compared to exact k-nearest-neighbor scans.

This matches the scenario exactly. The VehicleHealthSummary table already includes an Embeddings (vector (1536)) column. In Microsoft SQL platforms, embeddings are stored in vector columns and queried for semantic similarity. To improve performance and reduce response time, Microsoft recommends a vector index, not a regular B-tree nonclustered index and not a full-text index.

The other options are not appropriate:

* B is incorrect because a standard nonclustered index is not the index type used for vector similarity operations.

Microsoft's current documentation is explicit that vector indexes support approximate nearest neighbor search, and that the optimizer can use the ANN index automatically for vector queries. That is the exam-aligned design choice when the goal is fast retrieval of the most relevant health summary reports from an embeddings column.

• • • • •

DP-800 Reliable Dumps Files: <https://www.prepawayete.com/Microsoft/DP-800-practice-exam-dumps.html>

- All-in-One Exam Guide Prep Guide - DP-800 < and easily obtain a free download on ➡ www.practicevce.com □□□ □DP-800 Interactive Course
- DP-800 Study Materials - DP-800 Exam Preparatory - DP-800 Test Prep □ Open ➡ www.pdfvce.com □ and search for 《 DP-800 》 to download exam materials for free □Instant DP-800 Discount
- DP-800 Interactive Course □ ExamDP-800 Material □ DP-800 Interactive Course □ Go to website ➤ www.validtorrent.com □ open and search for □ DP-800 □ to download for free □Latest DP-800 Study Guide
- DP-800 Interactive Course □ DP-800 Best Practice □ DP-800 Valid Braindumps □ Immediately open ➡ www.pdfvce.com □ and search for ⇒ DP-800 ⇐ to obtain a free download □Actual DP-800 Tests
- Test DP-800 Study Guide □ Exam Cram DP-800 Pdf □ Actual DP-800 Tests ♣ Search for （ DP-800 ） and obtain a free download on ⇒ www.examcollectionpass.com ⇐ □Reliable DP-800 Exam Preparation
- DP-800 Test Lab Questions □ Exam Cram DP-800 Pdf □ Test DP-800 Study Guide □ Copy URL ➤ www.pdfvce.com □ open and search for [DP-800] to download for free □Exam DP-800 Material
- DP-800 Valid Exam Camp □ DP-800 Test Lab Questions □ New DP-800 Test Papers □ Enter ➡ www.dumpsquestion.com □ and search for “ DP-800 ”to download for free □DP-800 Certification Practice
- DP-800 exam dumps and Microsoft DP-800 exam Simulator □ Search for ➡ DP-800 □ and download it for free on ➡ www.pdfvce.com □ website □DP-800 Actual Exams
- Exam Cram DP-800 Pdf □ Actual DP-800 Tests □ Instant DP-800 Discount □ Download ➡ DP-800 □□□ for free by simply searching on▷ www.prepaywayexam.com◁ □Latest DP-800 Study Guide
- Don't Miss Amazing Offers - Buy Microsoft DP-800 Actual Dumps Today □ ✓ www.pdfvce.com □✓□ is best website to obtain ⇒ DP-800 ⇐ for free download □DP-800 Test Lab Questions
- DP-800 Best Practice ☎ Actual DP-800 Tests □ DP-800 Valid Exam Camp □ Search for 「 DP-800 」 and download it for free on ➡ www.pdffidumps.com □ website □Instant DP-800 Discount
- www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw,
www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, Disposable vapes