100% Pass Newest DP-800 - Developing AI-Enabled Database Solutions Trusted Exam Resource

Our DP-800 test prep is of high quality. The passing rate and the hit rate are both high. The passing rate is about 98%-100%. We can guarantee that you have a very high possibility to pass the exam. The DP-800 guide torrent is compiled by the experts and approved by the professionals with rich experiences. The DP-800 prep torrent is the products of high quality complied elaborately and gone through strict analysis and summary according to previous exam papers and the popular trend in the industry. The language of the DP-800 exam material is simple and easy to be understood.

When we started offering Microsoft DP-800 exam questions and answers and exam simulator, we did not think that we will get such a big reputation. What we are doing now is incredible form of a guarantee. PassSureExam guarantee passing rate of 100%, you use your Microsoft DP-800 Exam to try our Microsoft DP-800 training products, this is correct, we can guarantee your success.

>> DP-800 Trusted Exam Resource <<

DP-800 Test Question | Associate DP-800 Level Exam

Our DP-800 practice exams lower the likelihood of failing and are optimal for self-evaluation. You can access the web-based Developing AI-Enabled Database Solutions (DP-800) practice exam online without having to install any software. All browsers and operating systems support our web-based DP-800 Practice Test. Every Windows computer supports our desktop Microsoft DP-800 practice exam software, enabling you to prepare for the DP-800 test without an active internet connection. You can customize the time and types of these Microsoft Questions.

Microsoft DP-800 Exam Syllabus Topics:

Topic	Details
Topic 1	• Secure, optimize, and deploy database solutions: This domain focuses on implementing data security measures like encryption, masking, and row-level security, optimizing query performance, managing CI • CD pipelines using SQL Database Projects, and integrating SQL solutions with Azure services including Data API builder and monitoring tools.
Topic 2	• Design and develop database solutions: This domain covers designing and building database objects such as tables, views, functions, stored procedures, and triggers, along with writing advanced T-SQL code and leveraging AI-assisted tools like GitHub Copilot and MCP for SQL development.
Topic 3	• Implement AI capabilities in database solutions: This domain covers designing and managing external AI models and embeddings, implementing full-text, semantic vector, and hybrid search strategies, and building retrieval-augmented generation (RAG) solutions that connect database outputs with language models.

Microsoft Developing AI-Enabled Database Solutions Sample Questions (Q31-Q36):

NEW QUESTION # 31
You have a GitHub Enterprise subscription.
Your team is developing an Azure SQL dataset solution from a locally cloned GitHub repository by using Microsoft Visual Studio Code and GitHub Copilot Chat.
A mix of GitHub Copilot instructions is configured at different levels, including organization-wide, repository-wide, agent-specific, and personal.
Which instructions will take precedence over the others?

- A. personal
- B. organization-wide
- C. repository-wide
- D. agent-specific

Answer: A

Explanation:
In a GitHub Enterprise environment, personal instructions (user-level) generally have the highest precedence for individual interactions in VS Code.
Order of Precedence
When GitHub Copilot processes instructions, it follows a "specific-to-general" hierarchy. The most specific context (the individual user) typically overrides broader organizational settings.
1. Personal Instructions
Set via .github/copilot-instructions.md in your local home directory or VS Code settings.
Overrides all other layers for your specific session.
2. Agent-Specific / Extension Instructions
Specific instructions provided to a custom agent (like @workspace or a custom Chat participant).
3. Repository-wide Instructions
Stored in the .github/copilot-instructions.md file within the specific repository.
4. Organization-wide Policies
Set by Enterprise/Org admins. These usually act as "guardrails" (e.g., blocking suggestions) rather than stylistic instructions.
Reference:
https://aiddbot.com/vscode-and-github-copilot

NEW QUESTION # 32
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 Fmbeddings (vector (1536)) column
- B. a vector index on the Embedding* (vector (1536)) column
- C. a full-text index on the Fmbeddings (vector (1536)) column
- D. a computed column that manually compares vector values

Answer: B

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. A vector index is purpose-built for finding the most similar vectors efficiently.
The other options are not appropriate:
* A would require manual comparison logic and would increase latency rather than minimize it.
* B is incorrect because a standard nonclustered index is not the index type used for vector similarity operations.
* C is incorrect because full-text indexes are for textual token-based search, not numeric vector embeddings.
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.

NEW QUESTION # 33
You have a database named DB1. The schema is stored in a Git repository as an SDK-style SQL database project.
You have a GitHub Actions workflow that already runs dotnet build and produces a database artifact.
You need to add a deployment step that publishes the dacpac file to an Azure SQL database by using the secrets stored in GitHub repository secrets What should you include in the workflow?

- A. 
```
- name: Publish
  uses: azure/sql-action@v2
  with:
    action: extract
    path: bin/Debug/db1.dacpac
    connection-string: ${{ secrets.SQL_CONNECTION_STRING }}
```
- B. 
```
env:
  SQL_CONNECTION_STRING: Server-tcp:myserver.database.windows.net;
...
steps:
- name: Publish
  uses: azure/sql-action@v2
  with:
    action: publish
    path: bin/Debug/db1.dacpac
    connection-string: ${{ env.SQL_CONNECTION_STRING }}
```
- C. 
```
- name: Publish
  uses: azure/sql-action@v2
  with:
    action: publish
    path: bin/Debug/db1.dacpac
    connection-string: ${{ secrets.SQL_CONNECTION_STRING }}
```
- D. 
```
- name: Publish
  run: |
    dotnet build db1.sqlproj \
      /p:TargetConnectionString="${{ secrets.SQL_CONNECTION_STRING }}"
```

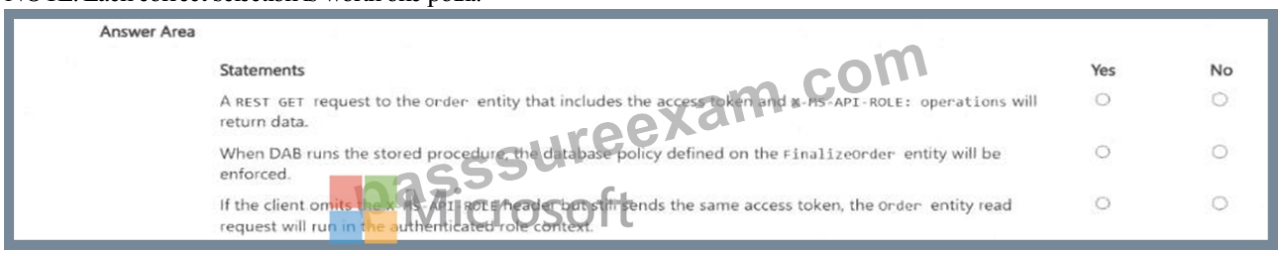
Answer: C

Explanation:
The correct workflow step is Option C because it uses the Azure SQL GitHub Action to publish a .dacpac file and reads the connection string from GitHub repository secrets , which is exactly what the requirement asks for. Microsoft's Azure SQL GitHub Actions guidance shows using azure/sql-action@v2 with a connection string stored in secrets and a DACPAC path for deployment.
The key parts that make C correct are:
* uses: azure/sql-action@v2
* action: publish
* path: bin/Debug/db1.dacpac
* connection-string: ${{ secrets.SQL_CONNECTION_STRING }}
That matches the documented publish pattern for deploying a DACPAC to Azure SQL Database from GitHub Actions. Microsoft and the Azure SQL action documentation both describe Publish as the deployment action for applying a DACPAC to a target database, while Extract is used to create a DACPAC from an existing database, not deploy one.
Why the other options are incorrect:
* A uses an environment variable defined inline with a visible connection string rather than using GitHub repository secrets , which does not meet the requirement.
* B uses action: extract, which would create a DACPAC from a database instead of publishing the existing DACPAC artifact.
* D passes a target connection string to dotnet build, but the question says the workflow already runs dotnet build and produces a database artifact . The missing step is the deployment/publish step, not another build step. Microsoft's SQL project automation guidance separates build the DACPAC from publish the DACPAC .

NEW QUESTION # 34
You have an Azure SQL database named SalesDB
You have a Data API builder (DAB) instance that exposes the following entities in SalesDB
* A table entity named Order mapped to a table named dbo. Orders
* A stored procedure entity named FinalizeOrder mapped to a stored procedure named dbo.usp_FinalizeOrder The DAB runtime configuration includes the following permissions.

```
{
  "entities": {
    "Order": {
      "source": "dbo.Orders",
      "permissions": [
        {
          "role": "authenticated",
          "actions": [ "read" ]
        }
      ]
    },
    "FinalizeOrder": {
      "source": "dbo.usp_FinalizeOrder",
      "permissions": [
        {
          "role": "operations",
          "actions": [ "execute" ],
          "policy": {
            "database": "TenantId = @claims.tenantId"
          }
        }
      ]
    }
  }
}
```

Client requests include a Microsoft Entra access token. The client also sends HTTP header x-MS-APl-ROlE: operations for both REST and GraphQL requests.
For each of the following statements, select Yes if the statement is true. Otherwise, select No.
NOTE: Each correct selection is worth one point.



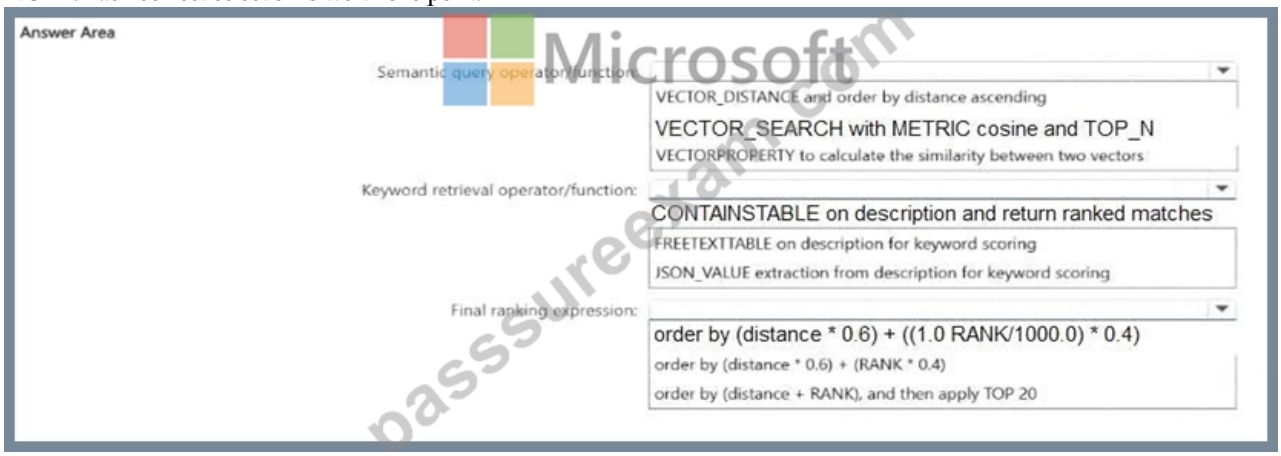
Answer:

Explanation:

Explanation:
* A REST GET request to the order entity that includes the access token and x-MS-API-ROLE: operations will return data. # No
* When DAB runs the stored procedure, the database policy defined on the FinalizeOrder entity will be enforced. # Yes
* If the client omits the x-MS-API-ROLE header but still sends the same access token, the order entity read request will run in the

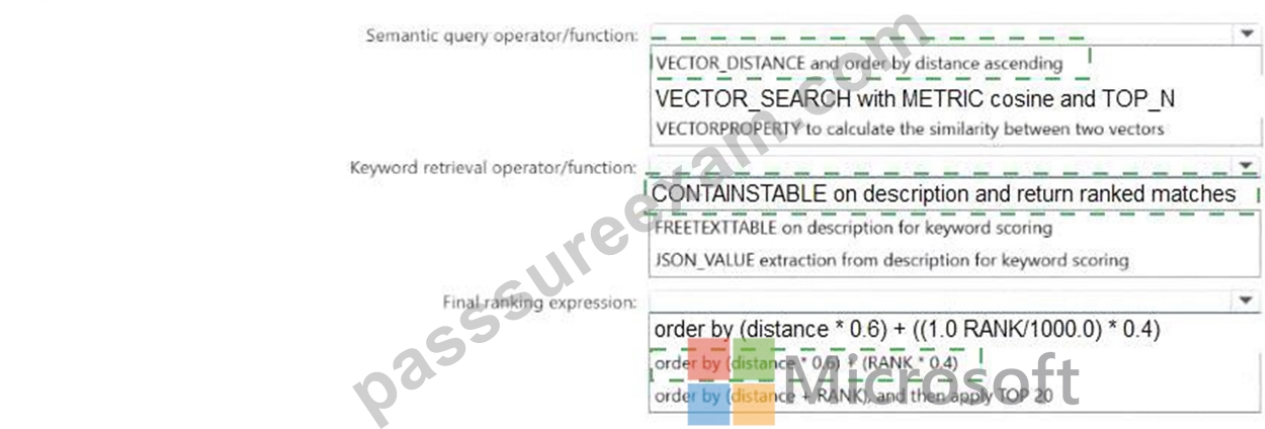
authenticated role context. # Yes The first statement is No . In Data API builder, when a valid token is sent with X-MS-API-ROLE, the request runs in that requested role if that role is present in the token . Here, that means the effective role becomes operations , not authenticated. But the order entity grants read only to the authenticated role, not to operations, so the GET request would not be authorized to return data.
The second statement is Yes . DAB evaluates the request against the permissions and policies configured for the effective role on the requested entity. The FinalizeOrder entity grants execute to role operations and includes a database policy of TenantId = @claims.tenantid, so that policy is part of the enforced authorization
/filtering behavior when the stored procedure entity is executed.
The third statement is Yes . If the client sends a valid access token without X-MS-API-ROLE, DAB uses the built-in Authenticated system role by default. Since the order entity allows read for the authenticated role, that read request runs in the authenticated role context.
Top of Form
Bottom of Form

NEW QUESTION # 35
You have an Azure SQL database that contains a table named stores, stores contains a column named description and a vector column named embedding.
You need to implement a hybrid search query that meets the following requirements:
* Uses full-text search on description for the keyword portion
* Returns the top 20 results based on a combined score that uses a weighted formula of 60% vector distance and 40% full-text rank
How should you configure the query components? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.



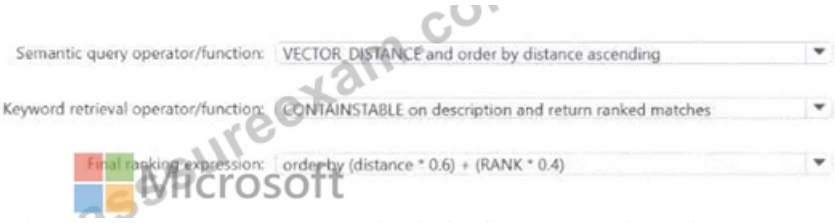
Answer:

Explanation:



Explanation:

Answer Area



For the vector portion, the correct choice is VECTOR_DISTANCE and order by distance ascending . The requirement is to build a combined weighted formula using the actual vector distance. Microsoft documents that VECTOR_DISTANCE returns the exact distance between two vectors. Since lower distance means greater similarity, ascending distance is the right direction for ranking. VECTOR_SEARCH is for ANN retrieval, but this hotspot specifically asks for a weighted formula based on distance , so VECTOR_DISTANCE is the appropriate operator.
For the keyword portion, the correct choice is CONTAINSTABLE on description and return ranked matches . Microsoft documents that CONTAINSTABLE returns a RANK column from 0 through 1000 , which is exactly what is needed for weighted scoring in a hybrid formula.
For the final ranking expression, the best choice is order by (distance * 0.6) + ((1.0 - RANK/1000.0) * 0.4) .
This works because vector distance is a lower-is-better metric, while full-text RANK is a higher-is-better metric. Dividing RANK by 1000 normalizes it to the documented range, and subtracting from 1.0 converts it into a lower-is-better term so both components can be combined consistently in one ascending score. This final step is a sound inference based on Microsoft's documented distance semantics and full-text rank range.

NEW QUESTION # 36
.....