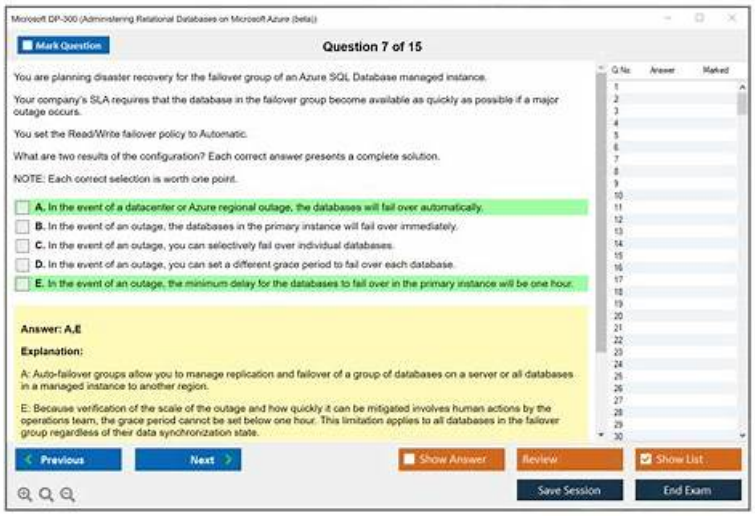
DP-800 Latest Practice Questions, Accurate DP-800 Study Material



A generally accepted view on society is only the professionals engaged in professionally work, and so on, only professional in accordance with professional standards of study materials, as our Developing AI-Enabled Database Solutions study questions, to bring more professional quality service for the user. Our study materials can give the user confidence and strongly rely on feeling, lets the user in the reference appendix not alone on the road, because we are to accompany the examinee on DP-800 Exam, candidates need to not only learning content of teaching, but also share his arduous difficult helper, so believe us, we are so professional company.

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	• 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.
Topic 3	• 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.

>> DP-800 Latest Practice Questions <<

Useful DP-800 Latest Practice Questions | DP-800 100% Free Accurate Study Material

The main reason why people look for Microsoft DP-800 practice test is that these help them to prepare for the exam. Even if you study well but with no idea of the Developing AI-Enabled Database Solutions DP-800 exam pattern, it will be tough to crack the nut. You shall waste your time thinking about the pattern and how to attempt the Developing AI-Enabled Database Solutions DP-800 Exam Questions. On the other hand, if you know the Developing AI-Enabled Database Solutions DP-800 exam questions well, you can use that time to solve the queries and improve your chances to score well in the exam.

Microsoft Developing AI-Enabled Database Solutions Sample Questions (Q71-Q76):

NEW QUESTION # 71
You have an Azure SQL database.
You need to create a scalar user-defined function (UDF) that returns the number of whole years between an input parameter named @OrderDate and the current date/time as a single positive integer. The function must be created in Azure SQL Database.
You write the following code.

```
01 CREATE FUNCTION dbo.ufnYearsSinceOrder (@OrderDate datetime2)
02 RETURNS int
03 AS
04 BEGIN
05
06 END
```


What should you insert at line 05?

- A. RETURN DATEDIFF(year, @OrderDate, GETDATE());
- B. DATEPART(year, GETDATE()) - DATEPART(year, @orderdate)
- C. DATEDIFF(month, @orderdate, GETDATE()) / 12
- D. RETURN DATEDIFF(year, GETDATE(), @OrderDate);

Answer: A

Explanation:
Use RETURN to produce the scalar value of the function.
In an Azure SQL Database scalar function (a user-defined function that returns a single value), you must use the RETURN statement to return the scalar value.
The RETURN statement immediately terminates the function's execution and returns the value specified in its argument to the calling statement or procedure. The value returned must be of the data type specified in the RETURNS clause of the function definition.
The second argument to DATEDIFF should be @OrderDate as it is the start date, while the third argument is the end date, which is the current date.
Note:
DATEDIFF (Transact-SQL)
This function returns the count (as a signed integer value) of the specified datepart boundaries crossed between the specified startdate and enddate.
Syntax
DATEDIFF (datepart , startdate , enddate)
Arguments
datepart
Specifies the units in which DATEDIFF reports the difference between the startdate and enddate.
Commonly used datepart units include month or second.
Reference:
https://learn.microsoft.com/en-us/sql/t-sql/functions/datediff-transact-sql

NEW QUESTION # 72
Case Study 1 - Contoso
Existing Environment
Azure Environment
Contoso has an Azure subscription in North Europe that contains the corporate infrastructure.
The current infrastructure contains a Microsoft SQL Server 2017 database. The database contains the following tables.

Table names	Column names
CustomerFeedback	• FeedbackId (int) (primarykey) • FeedbackJson (nvarchar (max))
Fleets	• FleetId (int) (primarykey) • FleetName (nvarchar(100)) • Description (nvarchar(256))
MaintenanceEvents	• MaintenanceId (int) (primarykey) • VehicleId (int) • LastModifiedUTC (datetime2) • Description (nvarchar(256))
SupportTickets	• TicketId (int) (primarykey) • FleetId (int) • CreatedUtc (datetime2)
UserAccounts	• UserId (int) (primarykey) • UserPrincipalName (nvarchar(256)) • JobRole (nvarchar(256)) • StartDate (datetime2)
VehicleIncidentReports	• IncidentId (int) (primarykey) • VehicleId (int) • FleetId (int) • IncidentType (nvarchar(50)) • VehicleLocation (nvarchar(200)) • IncidentDescription (nvarchar(max)) • SeverityScore (int)
Vehicles	• VehicleId (int) (primarykey) • VIN ((nvarchar(50)) • VehicleDescription (nvarchar(256))
VehicleHealthSummary	• VehicleId (int) (primarykey) • FleetId (int) • Summary (nvarchar(2000)) • LastUpdatedUtc (datetime2) • EngineStatus [bit] • EngineStatusLastUpdatedUtc (datetime2) • BatteryHealth (int) • Embeddings (vector (1536))

The FeedbackJsoncolumn has a full-text index and stores JSON documents in the following format.

```
"text": "The battery drains too fast when driving uphill.",
"category": "Battery",
"metadata": {
  "appVersion": "5.2.1",
  "device": "Android",
  "language": "en-GB"
}
```


The support staff at Contoso never has the UNMASKpermission.

Problem Statements

Contoso is deploying a new Azure SQL database that will become the authoritative data store for the following:

* AI workloads
* Vector search
* Modernized API access
* Retrieval Augmented Generation (RAG) pipelines

Sometimes the ingestion pipeline fails due to malformed JSON and duplicate payloads.

The engineers at Contoso report that the following dashboard query runs slowly.

```
SELECT VehicleId, LastUpdatedUtc, EngineStatus, BatteryHealth
FROM dbo.VehicleHealthSummary
WHERE FleetId = @FleetId
ORDER BY LastUpdatedUtc DESC;
```

You review the execution plan and discover that the plan shows a clustered index scan.

VehicleIncidentReportsoften contains details about the weather, traffic conditions, and location. Analysts report that it is difficult to find similar incidents based on these details.

Requirements

Planned Changes

Contoso wants to modernize Fleet Intelligence Platform to support AI-powered semantic search over incident reports.

Security Requirements

Contoso identifies the following security requirements:

* Restrict the support staff from viewing Personally Identifiable Information (PII) data, which is full email addresses and phone numbers.
* Enforce row-level filtering so that analysts see only incidents for the fleets to which they are assigned. The analysts can be assigned to multiple fleets.

Database Performance and Requirements

Contoso identifies the following telemetry requirements:

* Telemetry data must be stored in a partitioned table.
* Telemetry data must provide predictable performance for ingestion and retention operations.
* latitude, longitude, and accuracyJSON properties must be filtered by using an index seek.

Contoso identifies the following maintenance data requirements:

* Ensure that any changes to a row in the MaintenanceEventstable updates the corresponding value in the LastModifiedUtccolumn to the time of the change.
* Avoid recursive updates.

AI Search, Embeddings, and Vector Indexing

Contoso plans to implement semantic search over incident data to meet the following requirements:

* Embeddings must be stored in dedicated Azure SQL Database tables.
* Embeddings must be generated from rich natural language fields.
* Chunking must preserve semantic coherence.
* Hybrid search must combine the following:

- Vector similarity
- Keyword filtering or boosting

Development Requirements

The development team at Contoso will use Microsoft Visual Studio Code and GitHub Copilot and will retrieve live metadata from the databases.

Contoso identifies the following requirements for querying data in the FeedbackJsoncolumn of the CustomerFeedbacktable:

* Extract the customer feedback text from the JSON document.
* Filter rows where the JSON text contains a keyword.

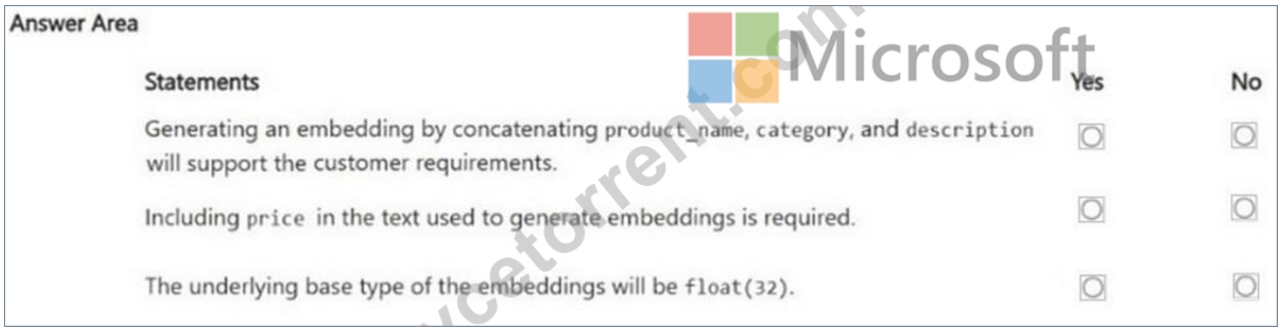
* Calculate a fuzzy similarity score between the feedback text and a known issue description.
* Order the results by similarity score, with the highest score first.
You need to enable similarity search to provide the analysts with the ability to retrieve the most relevant health summary reports. The solution must minimize latency. What should you include in the solution?

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

Answer: C

Explanation:
Scenario: There is a VehicleHealthSummary table.
To enable similarity search on your health summary data while minimizing latency, you should use the native VECTOR data type and a DiskANN vector index, which are now available in public preview for Azure SQL Database.
Solution Implementation
1. Define the Vector Column: Ensure your embeddings are stored using the native VECTOR(1536) type rather than NVARCHAR or VARBINARY. This format is optimized for high- dimensional data and mathematical operations.
ALTER TABLE HealthSummaries
ADD SummaryVector VECTOR(1536);
2. Create the Vector Index: Use the CREATE VECTOR INDEX statement. In Azure SQL, this uses the DiskANN algorithm, which is specifically designed to provide high-speed Approximate Nearest Neighbor (ANN) searches for large datasets.
CREATE VECTOR INDEX idx_health_summary_vector
ON HealthSummaries (SummaryVector)
WITH (METRIC = 'COSINE', TYPE = 'DISKANN');
3. Perform the Similarity Search: To leverage the index for low-latency retrieval, use the VECTOR_SEARCH function rather than VECTOR_DISTANCE. While VECTOR_DISTANCE calculates exact values (resulting in a full table scan), VECTOR_SEARCH utilizes the DiskANN index to find the most relevant reports quickly.
SELECT TOP(10) *
FROM HealthSummaries
ORDER BY VECTOR_DISTANCE('cosine', SummaryVector, @query_vector);
Reference:
https://learn.microsoft.com/en-us/samples/azure-samples/azure-sql-db-openai/azure-sql-db- openai/

NEW QUESTION # 73
Hotspot Question
Your company has an ecommerce catalog in a Microsoft SQL Server 2025 database named SalesDB. SalesDB contains a table named products. products contains the following columns:
- product_id (int)
- product_name (nvarchar(200))
- description (nvarchar(max))
- category (nvarchar(50))
- brand (nvarchar(50))
- price (decimal)
- sku (nvarchar(40))
The description fields are updated daily by a content pipeline, and price can change multiple times per day.
You want customers to be able to submit natural language queries and apply structured filters for brand and price.
You plan to store embeddings in a new VECTOR(1536) column and use VECTOR_SEARCH(...
METRIC='cosine' ...).
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:

NEW QUESTION # 74

You have an Azure SQL database that contains a table named dbo.ManualChunks. dbo.HonualChunks contains product manuals A retrieval query already returns the top five matching chunks as nvarchar(max) text.

You need to call an Azure OpenAI REST endpomt for chat completions. The request body must include both the user question and theretiieved chunks.

You write the following Transact-SQL code.

```
01 CREATE DATABASE SCOPED CREDENTIAL AzureOpenAIHeaders
02 WITH IDENTITY = 'HTTPEndpointHeaders',
03   SECRET = N'{"api-key":"<YOUR_AZURE_OPENAI_API_KEY>"}';
04 GO
05 CREATE OR ALTER PROCEDURE dbo.AskManuals
06   . . .
07   SELECT @chunks =
08   (
09     SELECT TOP (5)
10       mc.ChunkText AS [text]
11     FROM dbo.ManualChunks AS mc
12     ORDER BY mc.Score DESC
13     FOR JSON PATH
14   );
15   SET @payload =
16   (
17     SELECT
18       'system' AS [messages[0].role],
19       'Use only the provided manual chunks.' AS [messages[0].content],
20       'user' AS [messages[1].role],
21       CONCAT(@question, CHAR(10), JSON_QUERY(@chunks)) AS [messages[1].content]
22
23   );
24
25   EXEC @retval =
26     . . .
27 END;
28 GO
```


What should you insert at line 22?

- A. FOR XHL AUTO, TYPE, XML SCHEMA,
- B. FOR XHL PATH, INCLUDE_NULL_VAIUES
- C. FOR 3SON AUTO, IMCLUDE_MULL_VAIUES
- D. FOR JSON PATH, WITHOUT_ARRAY_WRAPPER

Answer: D

Explanation:
The correct insertion at line 22 is FOR JSON PATH, WITHOUT_ARRAY_WRAPPER .
The request body for the Azure OpenAI chat completions call must be a single JSON object containing the messages array with both the system/user content and the retrieved chunks. Microsoft documents that FOR JSON PATH is the preferred way to shape JSON output, especially when you want precise control over nested property names like messages[0].role and messages[1] .content.
The key detail is WITHOUT_ARRAY_WRAPPER . By default, FOR JSON returns results enclosed in square brackets as a JSON array. Microsoft documents that WITHOUT_ARRAY_WRAPPER removes those brackets so a single JSON object is produced instead. That is exactly what is needed here for @payload, because the stored procedure is building one request body, not an array of request bodies.

NEW QUESTION # 75
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" ]
      }
```


```
    ]
  }
}
```

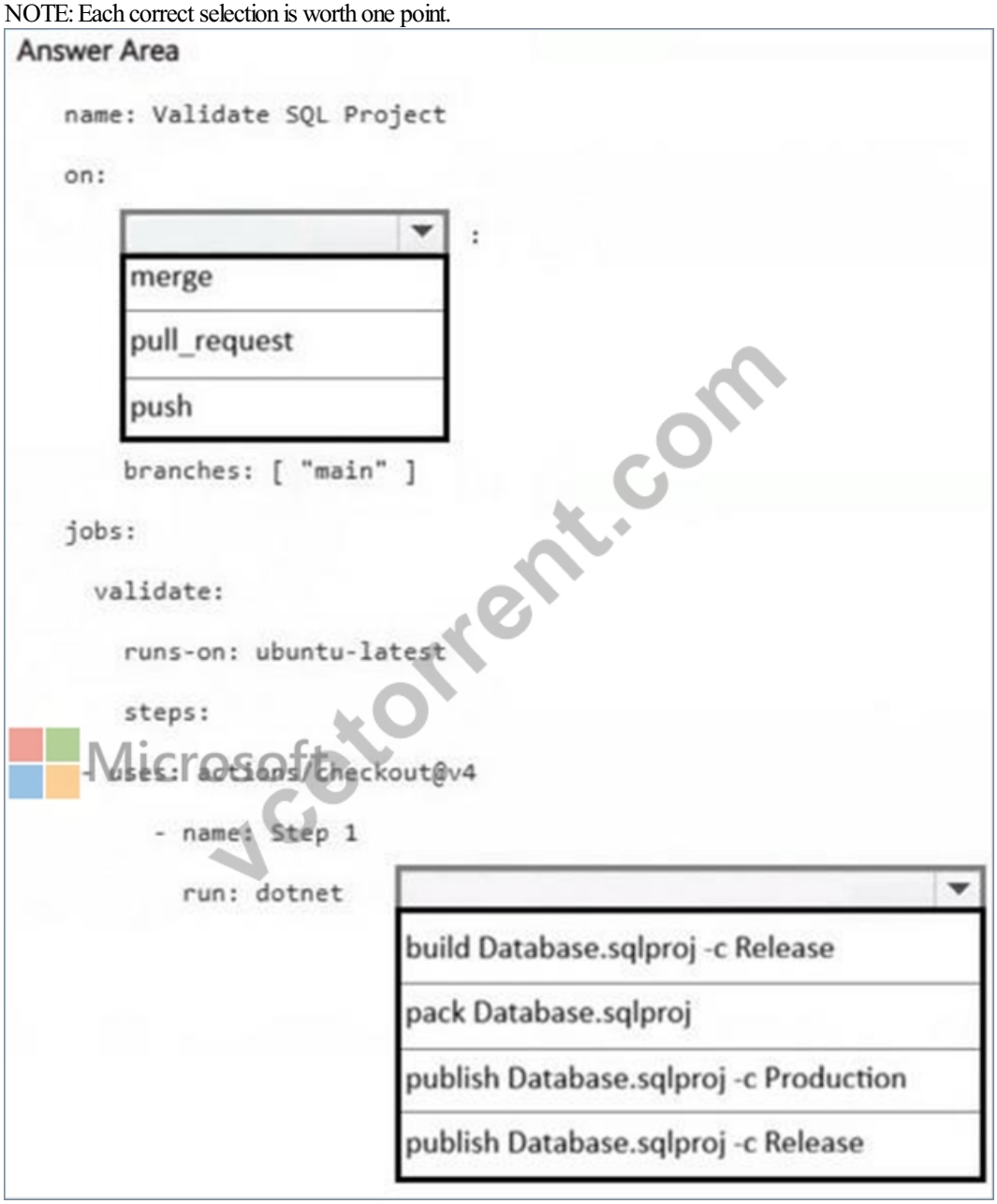
Hotspot Question

You create an SDK-style SQL database project in Microsoft Visual Studio Code named Database.sqlproj and add the project to a GitHub repository.

You need to configure a GitHub Actions workflow to support the planned changes for DB1.

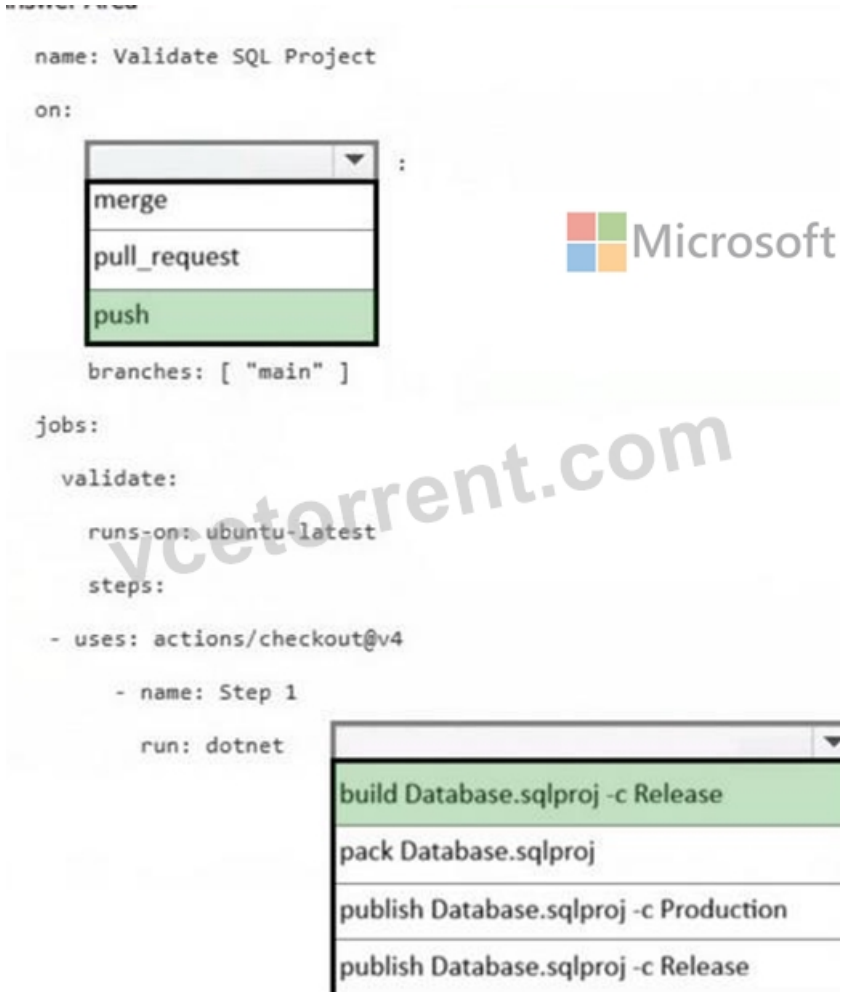
How should you complete the workflow? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.



Answer:

Explanation:



NEW QUESTION # 76

.....

Someone around you must be using our DP-800 exam questions. The users of our DP-800 exam materials are really very extensive. Or, you can consult someone who has participated in the DP-800 exam. They must know or use our products. We can confidently say that our products are leading in the products of the same industry. The richness and authority of DP-800 Exam Materials are officially certified.

www.practicevce.com □ open and search for " DP-800 " to download for free □Guaranteed DP-800 Success

- DP-800 Test Testking □ DP-800 Exam Simulator Free □ DP-800 Examcollection Dumps Torrent □ Enter ➽ www.pdfvce.com □ and search for ⇛ DP-800 ⇚ to download for free □Latest DP-800 Dumps Sheet
- DP-800 Valid Exam Pattern ℹ DP-800 Vce Format □ DP-800 Exam Tutorials □ Search for 【 DP-800 】 and obtain a free download on 【 www.prepawayexam.com 】 □DP-800 Latest Exam Experience
- webtechdirectory.com, mariamnurh740462.slypage.com, jayzzgr495280.blog2freedom.com, caoimhevtwm042136.vidublog.com, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, robustdirectory.com, carazwdy111105.blog-mall.com, lilianpgcd761708.daneblogger.com, tinybookmarks.com, montyonje786326.bloggosite.com, Disposable vapes