QUESTIONS & *ANSWERS*

verified

DBCLOUD EXAMS

EXPERIENICIA Y TECNOLOGÍA

DBCLOUD CERTIFIED

Technology Provider: Microsoft

Exam: Developing AI-Enabled Database Solutions

Number of questions in the database: 79

Exam Version: OCT.2026

Topic 1

Question #1*Topic 1*

HOTSPOT -

Case Study -
This is a case study. Case studies are not timed separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided.
To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other questions in this case study.
At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next section of the exam. After you begin a new section, you cannot return to this section.

To start the case study -
To display the first question in this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question.

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.

https://dbcloud.academy/

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 FeedbackJson column has a full-text index and stores JSON documents in the following format.

https://dbcloud.academy/

```
{
  "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 UNMASK permission.

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.
VehicleIncidentReports often 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

https://dbcloud.academy/

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 accuracy JSON 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 MaintenanceEvents table updates the corresponding value in the LastModifiedUtc column 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 FeedbackJson column of the CustomerFeedback table:
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 meet the development requirements for the FeedbackJson column.
How should you complete the Transact-SQL query? To answer, select the appropriate options in the answer area.

https://dbcloud.academy/

NOTE: Each correct selection is worth one point.

Answer Area

```
SELECT
  f.FeedbackId,
  f.VehicleId,
  [                                                                    ],
  EDIT_DISTANCE_SIMILARITY(
    JSON_VALUE(f.FeedbackJson, '$.text'),
    @KnownIssueDescription
  ) AS SimilarityScore
FROM
  dbo.CustomerFeedback f
WHERE
  [                                                                    ]
ORDER BY
  [                                                                    ]
  DESC;
```

- CONTAINS(FeedbackJson, @Keyword)
- EDIT_DISTANCE(JSON_VALUE(f.FeedbackJson, '$.details.comment'), @Keyword) < 3
- EDIT_DISTANCE(JSON_VALUE(f.FeedbackJson, '$.text'), @Keyword) < 3
- JSON_QUERY(f.FeedbackJson, '$.text', @KnownIssueDescription) AS FeedbackText
- JSON_VALUE(f.FeedbackJson, '$.text') AS FeedbackText
- SimilarityScore

Correct Answer:

https://dbcloud.academy/

Answer Area

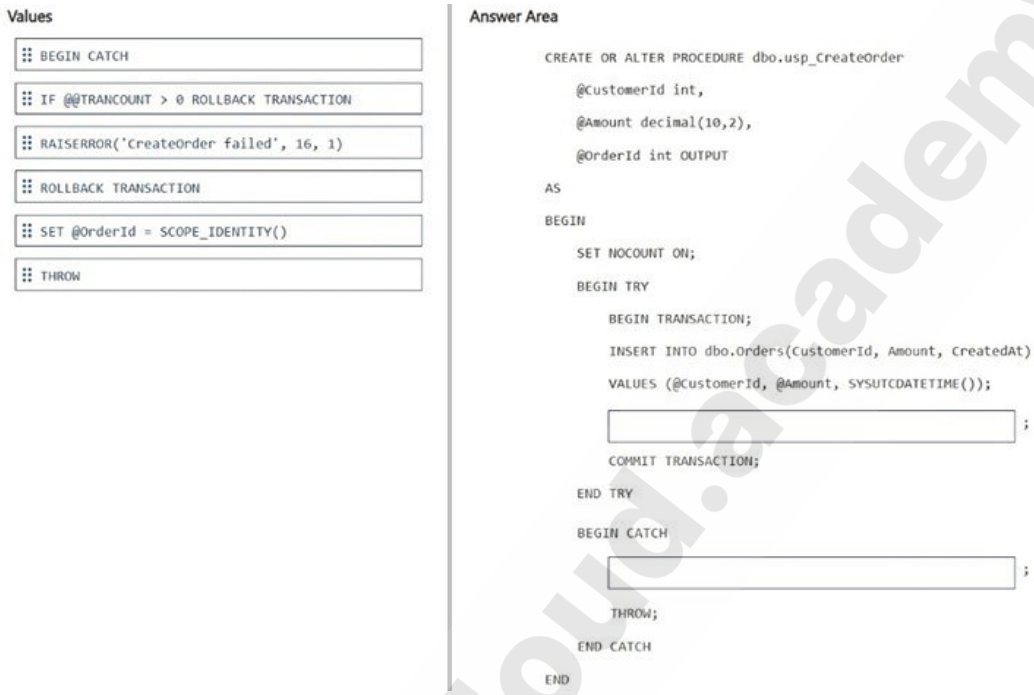
```
SELECT
  f.FeedbackId,
  f.VehicleId,
  [ CONTAINS(FeedbackJson, @Keyword)
    EDIT_DISTANCE(JSON_VALUE(f.FeedbackJson, '$.details.comment'), @Keyword) < 3
    EDIT_DISTANCE(JSON_VALUE(f.FeedbackJson, '$.text'), @Keyword) < 3
    JSON_QUERY(f.FeedbackJson, '$.text', @KnownIssueDescription) AS FeedbackText
    JSON_VALUE(f.FeedbackJson, '$.text') AS FeedbackText
    SimilarityScore ],
  EDIT_DISTANCE_SIMILARITY(
    JSON_VALUE(f.FeedbackJson, '$.text'),
    @KnownIssueDescription
  ) AS SimilarityScore
FROM
  dbo.CustomerFeedback f
WHERE
  [ CONTAINS(FeedbackJson, @Keyword)
    EDIT_DISTANCE(JSON_VALUE(f.FeedbackJson, '$.details.comment'), @Keyword) < 3
    EDIT_DISTANCE(JSON_VALUE(f.FeedbackJson, '$.text'), @Keyword) < 3
    JSON_QUERY(f.FeedbackJson, '$.text', @KnownIssueDescription) AS FeedbackText
    JSON_VALUE(f.FeedbackJson, '$.text') AS FeedbackText
    SimilarityScore ]
ORDER BY
  [ CONTAINS(FeedbackJson, @Keyword)
    EDIT_DISTANCE(JSON_VALUE(f.FeedbackJson, '$.details.comment'), @Keyword) < 3
    EDIT_DISTANCE(JSON_VALUE(f.FeedbackJson, '$.text'), @Keyword) < 3
    JSON_QUERY(f.FeedbackJson, '$.text', @KnownIssueDescription) AS FeedbackText
    JSON_VALUE(f.FeedbackJson, '$.text') AS FeedbackText
    SimilarityScore ]
DESC;
```

Question #2*Topic 1*

DRAG DROP -
You have an Azure SQL database that contains a table named dbo.Orders.
You have an application that calls a stored procedure named dbo.usp_CreateOrder to insert rows into dbo.Orders.
When an insert fails, the application receives inconsistent error details.

https://dbcloud.academy/

You need to implement error handling to ensure that any failures inside the procedure abort the transaction and return a consistent error to the caller.
How should you complete the stored procedure? To answer, drag the appropriate values to the correct targets. Each value may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.
NOTE: Each correct selection is worth one point.



Correct Answer:



https://dbcloud.academy/

Question #3Topic 1

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.

You need to disable the GitHub Copilot repository-level instructions for yourself without affecting other users.

What should you do?

A. From Visual Studio Code, modify your GitHub Copilot Chat user settings.

B. Add a --debug flag when you start the GitHub Copilot Chat extension.

C. Delete .github/copilot-instructions.md.

Correct Answer: A

Question #4*Topic 1*

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?

A.

```
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);
```

B.

https://dbcloud.academy/

```
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 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);
```

D.

```
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);
```

Correct Answer: D

Question #5*Topic 1*

You have a SQL database in Microsoft Fabric that contains a column named Payload. Payload stores customer data in JSON documents that have the following format.

```
{
"date": "2020-01-25",
"customer_email": "user@contoso.com",
...
}
```

Data analysis shows that some customers have subaddressing in their email address, for example, user1+promo@contoso.com.

You need to return a normalized email value that removes the subaddressing, for example, user1 +promo@contoso.com must be normalized to user1@contoso.com.

Which Transact-SQL expression should you use?

A. REGEXP_REPLACE(JSON_VALUE(Payload, '$.customer_email'), '\+.*$', '')

B. REGEXP_SUBSTR(JSON_VALUE(Payload, '$.customer_email'), '^[^+]+@.*$=')

C. REGEXP_REPLACE(JSON_VALUE(Payload, '$.customer_email'), '\+.*@', '@')

D. REGEXP_REPLACE(JSON_VALUE(Payload, '$.customer_email'), '\+.*', '')

Correct Answer: C

https://dbcloud.academy/