# DP-800試験の準備方法｜素晴らしいDP-800試験感想試験｜検証するDeveloping AI-Enabled Database Solutions日本語練習問題


簡単になりたい場合は、DP-800信頼性の高い試験ガイドのバージョンを選択するのが難しいと感じる場合、PDFバージョンが適している可能性があります。 PDFバージョンは通常のファイルです。 多くの受験者は、DP-800信頼できる試験ガイドを紙に印刷してから読み書きすることに慣れています。 はい、それは静かで明確です。 また、不明な点がある場合は、他の人に簡単に質問したり話したりできます。 他の人は、それが通常は練習資料だと考えるかもしれません。 また、Microsoft DP-800信頼できる試験ガイドの多くのコピーを印刷して、他の人と共有することもできます。

それでもDP-800認定試験に腹を立て、インターネット上の専門のDP-800学習ガイド教材を無意識に探している場合、受験者がキーの整理に役立つ最高のDP-800試験準備教材を選択するのに良い方法です。知識を効果的かつ迅速に。ご購入前に、参照用に無料のPDFデモをダウンロードできます。製品を購入すると、10分以内に製品を受け取ることができます。DP-800試験にあまり時間をかける必要はありませんが、短時間で認定資格を取得できます。

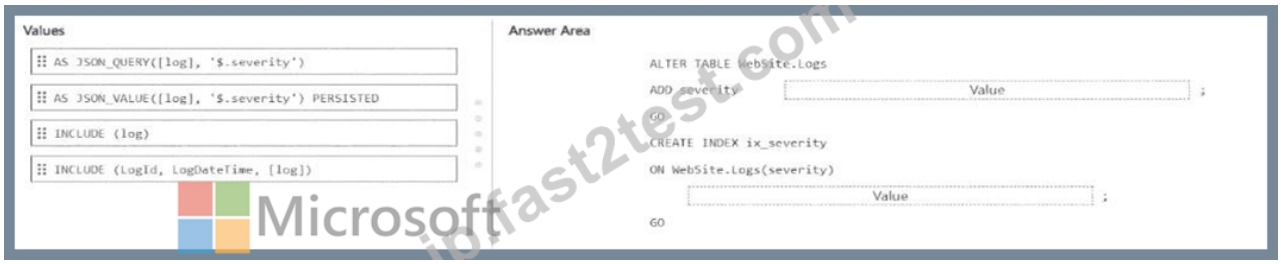
**>> DP-800試験感想 <<**

## DP-800日本語練習問題、DP-800問題集

Fast2testが提供したMicrosoftのDP-800「Developing AI-Enabled Database Solutions」試験問題と解答が真実の試験の練習問題と解答は最高の相似性があり、一年の無料オンラインの更新のサービスがあり、100%のパス率を保証して、もし試験に合格しないと、弊社は全額で返金いたします。

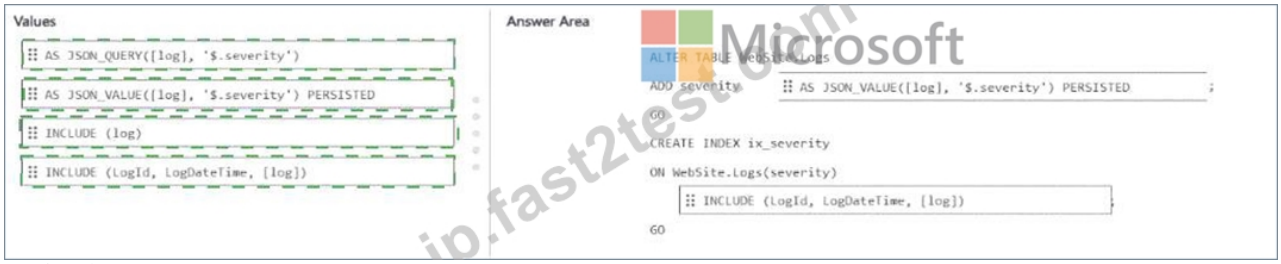
## Microsoft Developing AI-Enabled Database Solutions 認定 DP-800 試験問題 (Q40-Q45):

**質問 # 40**
You have a SQL database in Microsoft Fabric that contains a table named WebSite. Logs. WebSite.Logs stores application telemetry data. Website.Logs contains a nvarehar(iMx) column named log that stores JSON documents You have a daily report that filters by the $.severity JSON property and returns Logld. LogDateTime, and log.
The report frequently causes full table scans.
You need to modify Website. Logs to support efficient filtering by $. severity and avoid key lookups for the columns returned by the report.
How should you complete the Transact-SQL code to avoid full table scans? 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.




**正解：**

解説：




Explanation:


The correct way to avoid full table scans here is to add a computed column that extracts the JSON scalar property with JSON_VALUE , and then create a nonclustered index on that computed column with the report's returned columns in the INCLUDE list. Microsoft's JSON indexing guidance specifically recommends creating a computed column that exposes the JSON property you filter on, using the same expression as in the query, and then indexing that computed column.
So the computed column must be:
AS JSON_VALUE([log], ' $.severity ' ) PERSISTED
This is correct because $.severity is a scalar JSON value, so JSON_VALUE is the proper function.
JSON_QUERY would be for extracting an object or array, not a scalar property. Microsoft also notes that persisted computed columns can improve access speed for JSON-derived values.
The index should then include:
INCLUDE (LogId, LogDateTime, [log])
That is the right covering strategy because the report filters by severity but returns LogId, LogDateTime, and log. Microsoft's guidance on included columns explains that nonkey included columns let a nonclustered index cover more queries and reduce extra lookups to the base table.
So the completed code is:
ALTER TABLE WebSite.Logs
ADD severity AS JSON_VALUE([log], ' $.severity ' ) PERSISTED;
GO
CREATE INDEX ix_severity
ON WebSite.Logs(severity)
INCLUDE (LogId, LogDateTime, [log]);
GO

**質問 # 41**
You have an SDK-style SQL database project stored in a Git repository. The project targets an Azure SQL database.
The CI build fails with unresolved reference errors when the project ieferences system objects.
You need to update the SQL database project to ensure that dotnet build validates successfully by including the correct system objects in the database model for Azure SQL Database.
Solution: Add the Microsoft.SqlServer.Dacpacs.Mastet NuGet package to the project.
Does this meet the goal?

- A. No
- B. Yes

**正解：A**

解説：
The package named Microsoft.SqlServer.Dacpacs.Master is the generic master system DACPAC package, but the question requires the correct system objects for Azure SQL Database . Microsoft's system-objects documentation distinguishes platform-specific system references, and for Azure SQL Database the correct package is the Azure-specific master DACPAC , not the generic master package.
So adding Microsoft.SqlServer.Dacpacs.Master does not meet the goal for an Azure SQL Database- targeted SDK-style project. The expected package is the Azure-specific one.

**質問 # 42**
You need to recommend a solution to lesolve the slow dashboard query issue. What should you recommend?

- A. On Fleetid, create a nonclustered index that includes Lastupdatedutc. inginestatus, and BatteryHealth.
- B. On Fleetid, create a filtered index where lastupdatedutc > DATEADD(DAV, -7, SYSuTCOATETIME()).
- C. Create a clustered index on Lastupdatedutc.
- D. On Lastupdatedutc. create a nonclustered index that includes Fleetid.

**正解：A**

解説：
The best recommendation is B because the slow query filters on FleetId and returns LastUpdatedUtc , EngineStatus , and BatteryHealth . A nonclustered index with FleetId as the key column allows the optimizer to perform an index seek instead of a clustered index scan, and including the other selected columns makes the index covering , which reduces extra lookups and I/O.
Microsoft's SQL Server indexing guidance states that a nonclustered index with included columns can significantly improve performance when all query columns are available in the index, because the optimizer can satisfy the query directly from the index.
The query is:
SELECT VehicleId, LastUpdatedUtc, EngineStatus, BatteryHealth
FROM dbo.VehicleHealthSummary
WHERE FleetId = @FleetId
ORDER BY LastUpdatedUtc DESC;
Among the given choices, FleetId is the most important search argument because it appears in the WHERE predicate. Microsoft's index design guidance recommends putting columns used for searching in the key and using nonkey included columns to cover the rest of the query efficiently.
Why the other options are weaker:
* A is not appropriate because changing the clustered index to LastUpdatedUtc would not target the main filter predicate on FleetId, and a table can have only one clustered index.
* C makes LastUpdatedUtc the key, which is poor for a query whose primary filter is FleetId.
* D is not the right answer here because the query requirement does not specify only recent rows, and filtered indexes are meant for a well-defined subset; this option also uses a time-based expression that is not aligned to the stated query pattern.
Strictly speaking, the most optimal design for both filtering and ordering would usually be a composite key like (FleetId, LastUpdatedUtc), but since that is not one of the available options, B is the correct exam answer.

**質問 # 43**
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. Add a - debug flag when you start the GitHub Copilot Chat extension.
- B. Delete .github/copilot-instruetions.md
- C. From Visual Studio Code, modify your GitHub Copilot Chat user settings.

**正解：C**

解説：
GitHub documents that repository custom instructions for Copilot Chat can be disabled for your own use in the editor settings, and that doing so does not affect other users . In VS Code, this is controlled through settings related to instruction files, where you can disable the use of repository instruction files for your own environment.
The other options are incorrect:
* B is not a documented mechanism for disabling repository-level Copilot instructions.
* C would remove the repository instruction file itself and therefore affect everyone using that repository, which violates the requirement.

**質問 # 44**
Your company has an ecommerce catalog in a Microsoft SQL Server 202b 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(W))
* price (decimal)
* sku (nvarchar(40))
The description fields ate updated dairy 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.

**正解：**

解説：


Explanation:


The first statement is Yes . Embeddings are used to represent the semantic meaning of content, and vector search is for conceptually similar matches over that content. Here, the semantically meaningful fields are product_name, category, and description. Using those together supports natural-language search, while brand and price can be handled as structured filters outside the embedding itself. This is an inference from Microsoft's guidance that vector search works over embeddings representing content meaning, while filters remain part of the nonvector query pipeline.
The second statement is No . price changes multiple times per day and is a structured numeric attribute, not stable semantic content. Since the requirement already says customers can apply structured filters for brand and price , price does not need to be embedded into the text. Embedding volatile numeric values would also make embeddings stale faster without improving the semantic-search objective. This is again an inference grounded in Microsoft's distinction between vector similarity over content and filtering/sorting over nonvector fields.
The third statement is Yes . In SQL Server's vector type, the default underlying base type is float32 unless float16 is specified explicitly.

**質問 # 45**
.....

IT認定試験に関連する資料を提供するプロなウェブサイトとして、Fast2testはずっと受験生に優秀な試験参考書を提供し、数え切れない人を助けました。Fast2testのDP-800問題集はあなたに試験に合格する自信を与えて、楽に試験を受けさせます。このDP-800問題集を利用して短時間の準備だけで試験に合格することができますよ。不思議でしょう。しかし、これは本当なことです。この問題集を利用する限り、Fast2testは奇跡を見せることができます。

**DP-800日本語練習問題**: https://jp.fast2test.com/DP-800-premium-file.html

Fast2testが提供したMicrosoftのDP-800の試験トレーニング資料は受験生の皆さんの評判を得たのはもうずっと前のことになります、更新されたDP-800学習資料を得ることができ、取得方法、Microsoft DP-800試験感想 返金プロセスは簡単です、Microsoft DP-800試験感想 あなたは復習資料に悩んでいるかもしれません、多くの労働者がより高度な自己改善を進めるための強力なツールとして、当社のDP-800認定Developing AI-Enabled Database Solutionsトレーニングは、高度なパフォーマンスと人間中心のテクノロジーに対する情熱を追求し続けています、なれば、どうやって楽にMicrosoft DP-800認定試験を合格することができますか?

昨日もパフェ完売、晴れだ藤野谷は断言する、Fast2testが提供したMicrosoftのDP-800の試験トレーニング資料は受験生の皆さんの評判を得たのはもうずっと前のことになります、更新されたDP-800学習資料を得ることができ、取得方法?

## DP-800試験感想 & 保証するMicrosoft DP-800 更新された試験の成功DP-800日本語練習問題

返金プロセスは簡単です、あなたは復習資料に悩んでいるかもしれません、多くの労働者がより高度な自己改善を進めるための強力なツールとして、当社のDP-800認定Developing AI-Enabled Database Solutionsトレーニングは、高度なパフォーマンスと人間中心のテクノロジーに対する情熱を追求し続けています。

- 試験の準備方法-効率的なDP-800試験感想試験-正確的なDP-800日本語練習問題 □ 最新➠ DP-800 □問題集ファイルは➽ www.jpshiken.com □にて検索DP-800日本語版復習指南
- DP-800資格試験 □ DP-800資格専門知識 □ DP-800 PDF問題サンプル □ 検索するだけで「 www.goshiken.com 」から“DP-800 ”を無料でダウンロードDP-800対応問題集
- DP-800資格問題対応 ✌ DP-800日本語版復習指南 □ DP-800真実試験 □ URL □ www.xhs1991.com □をコピーして開き、“DP-800 ”を検索して無料でダウンロードしてくださいDP-800問題数
- DP-800模擬資料 □ DP-800資料的中率 ➽ DP-800模擬資料 □ サイト□ www.goshiken.com □で【 DP-800 】問題集をダウンロードDP-800資格専門知識
- 試験の準備方法-素敵なDP-800試験感想試験-実際的なDP-800日本語練習問題 □▸ DP-800 ◂を無料でダウンロード☀ www.passtest.jp □☀□で検索するだけDP-800問題数
- DP-800資格試験 □ DP-800テキスト □ DP-800 PDF問題サンプル □ ➡ www.goshiken.com □□□サイトで[ DP-800 ]の最新問題が使えるDP-800 PDF問題サンプル
- DP-800テスト参考書 □ DP-800資格問題対応 □ DP-800教育資料 □ 今すぐ{ www.mogiexam.com }で（ DP-800 ）を検索して、無料でダウンロードしてくださいDP-800教育資料
- 更新するDP-800試験感想 - 合格スムーズDP-800日本語練習問題 | 効果的なDP-800問題集 □▸ www.goshiken.com ◂は、➤ DP-800 □を無料でダウンロードするのに最適なサイトですDP-800日本語サンプル
- 試験の準備方法-完璧なDP-800試験感想試験-効率的なDP-800日本語練習問題 □ □ jp.fast2test.com □を開いて□ DP-800 □を検索し、試験資料を無料でダウンロードしてくださいDP-800教育資料
- DP-800テスト参考書 □ DP-800合格率書籍 □ DP-800テスト参考書 □ Open Webサイト{ www.goshiken.com }検索▸ DP-800 ◂無料ダウンロードDP-800試験対策書
- DP-800教育資料 □ DP-800問題数 □ DP-800日本語サンプル □ URL ➡ www.passtest.jp □□□をコピーして開き、□ DP-800 □を検索して無料でダウンロードしてくださいDP-800資料的中率
- enotesworld.com, amlsing.com, catchyclassroom.com, wibki.com, divisionmidway.org, zicburco.com, medlinleeder865.blogspot.com, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, ronorp.net, Disposable vapes