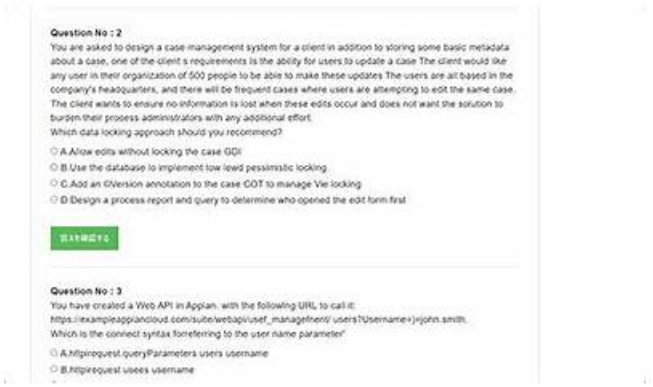
ACD-301認定資格試験、ACD-301対応資料


Question No : 2
You are asked to design a case management system for a client in addition to storing some basic metadata about a case, one of the client s requirements is the ability for users to update a case The client would like any user in their organization of 500 people to be able to make these updates The users are all based in the company's headquarters, and there will be frequent cases where users are attempting to edit the same case. The client wants to ensure no information is lost when these edits occur and does not want the solution to burden their process administrators with any additional effort.
Which data locking approach should you recommend?
A.Allow edits without locking the case GDI
B.Use the database to implement low level pessimistic locking
C.Add an @Version annotation to the case CDT to manage Vie locking
D.Design a process report and query to determine who opened the edit form first
Question No : 3
You have created a Web API in Appian, with the following URL to call it:
https://exampleappiancloud.com/suite/webapi/user_management/ users?username=jjohn.smith.
Which is the connect syntax forreferring to the user name parameter'
A.httprequest.queryParameters users username
B.httprequest usees username

無料でクラウドストレージから最新のMogiExam ACD-301 PDFダンプをダウンロードする：https://drive.google.com/open?id=1H0SkkcJHuFbPeES_VTzREr-nCtPsjb7p

この情報の時代には、Appian業界にとても注目され、この強い情報技術業界にAppian人材が得難いです。こうしてACD-301認定試験がとても重要になります。でも、この試験がとても難しくてAppian通になりたい方が障害になっています。

成功への道を示す指標として、当社のACD-301実践教材は、あなたの旅のあらゆる困難を乗り越えるために役立ちます。 すべての課題をウォークインのように扱うことはできませんが、ACD-301シミュレーションの実践により、レビューを効果的にすることができます。 それが、当社のACD-301調査問題がプロのモデルである理由です。 98％以上の高い合格率を誇るACD-301試験問題により、数千万人の受験者が試験に合格しました。

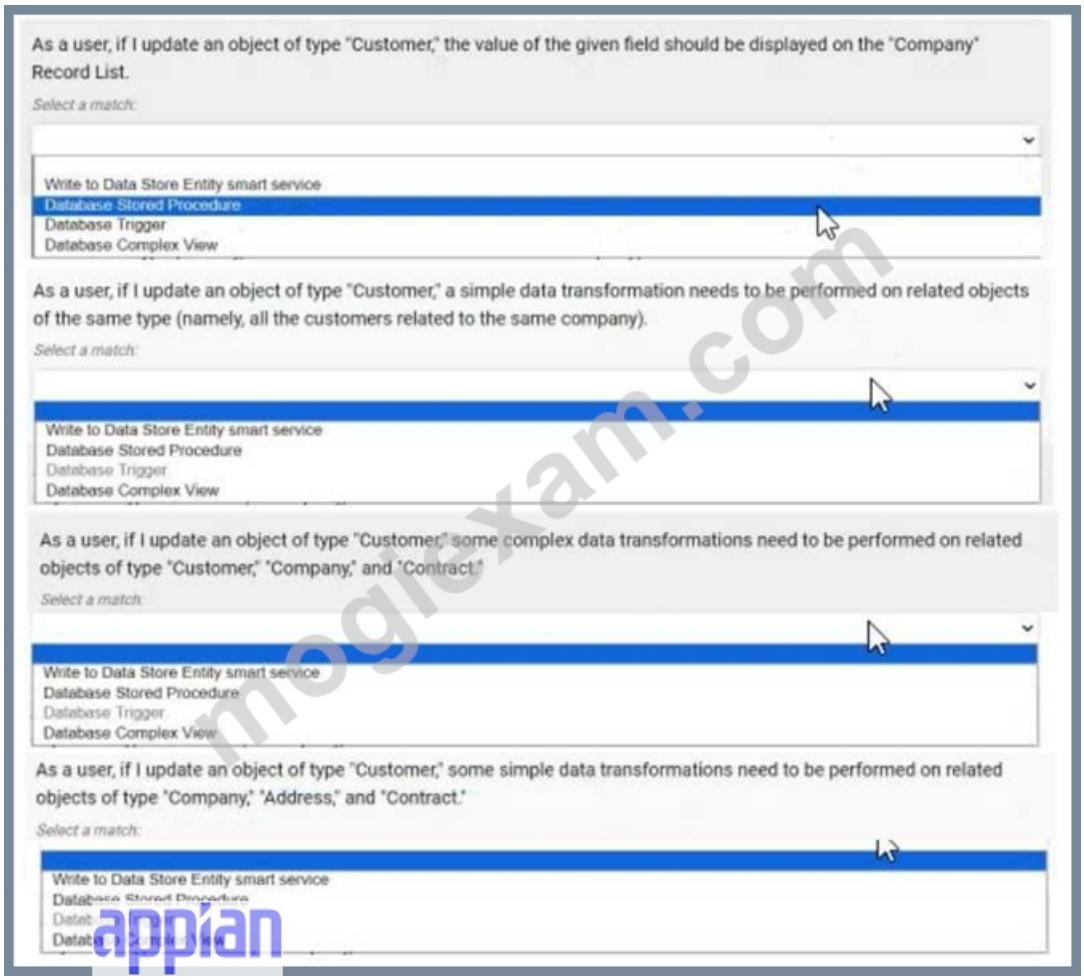
>> ACD-301認定資格試験 <<

正確的なACD-301認定資格試験試験-試験の準備方法-最高のACD-301対応資料

誰もが私たちの人生の貴重を認識する必要があります。時間を無駄にすることはできないので、目標をまっすぐに達成するための良い方法が必要です。もちろん、最新のACD-301試験トレントが最適です。 ACD-301試験の質問から、認定試験の知識だけでなく、質問に迅速かつ正確に回答する方法を学ぶことができることをお約束します。今、ACD-301テストトレントのデモを無料でダウンロードして、すばらしい品質を確認できます。

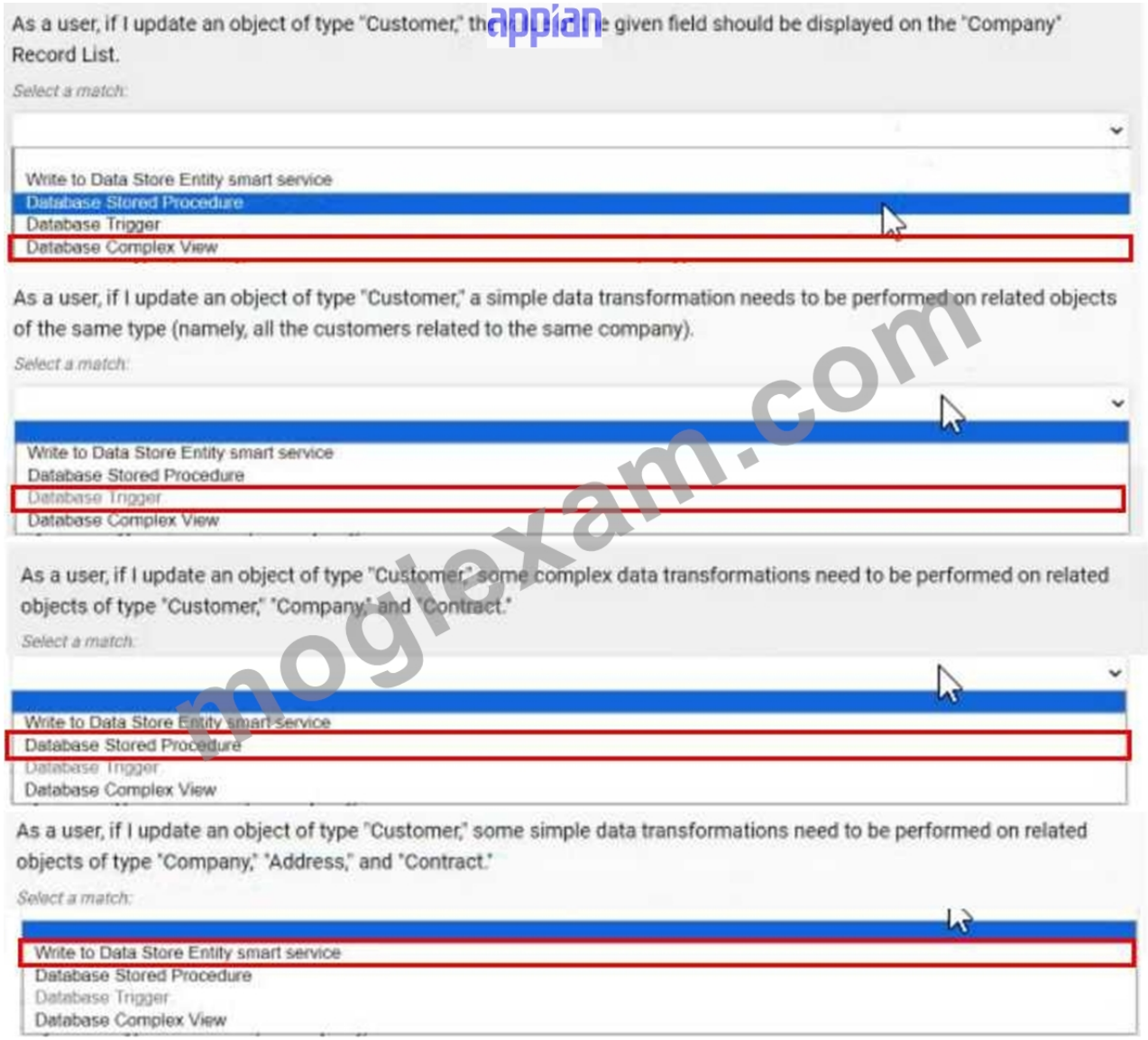
Appian Certified Lead Developer 認定 ACD-301 試験問題 (Q33-Q38):

質問 # 33
For each scenario outlined, match the best tool to use to meet expectations. Each tool will be used once Note: To change your responses, you may deselected your response by clicking the blank space at the top of the selection list.


As a user, if I update an object of type "Customer," the value of the given field should be displayed on the "Company" Record List.
Select a match:
Write to Data Store Entity smart service
Database Stored Procedure
Database Trigger
Database Complex View
As a user, if I update an object of type "Customer," a simple data transformation needs to be performed on related objects of the same type (namely, all the customers related to the same company).
Select a match:
Write to Data Store Entity smart service
Database Stored Procedure
Database Trigger
Database Complex View
As a user, if I update an object of type "Customer," some complex data transformations need to be performed on related objects of type "Customer," "Company," and "Contract."
Select a match:
Write to Data Store Entity smart service
Database Stored Procedure
Database Trigger
Database Complex View
As a user, if I update an object of type "Customer," some simple data transformations need to be performed on related objects of type "Company," "Address," and "Contract."
Select a match:
Write to Data Store Entity smart service
mogiexam.com
appian

正解：

解説：



質問 # 34
You have created a Web API in Appian with the following URL to call it: https://exampleappiancloud.com/suite/webapi/user_management/users?username=john.smith. Which is the correct syntax for referring to the username parameter?

- A. httpRequest.users.username
- B. httpRequest.queryParameters.username
- C. httpRequest.formData.username
- D. httpRequest.queryParameters.users.username

正解：B

解説：
Comprehensive and Detailed In-Depth Explanation:
In Appian, when creating a Web API, parameters passed in the URL (e.g., query parameters) are accessed within the Web API expression using the httpRequest object. The URL https://exampleappiancloud.com/suite/webapi/user_management/users?username=john.smith includes a query parameter username with the value john.smith. Appian's Web API documentation specifies how to handle such parameters in the expression rule associated with the Web API.
Option D (httpRequest.queryParameters.username):
This is the correct syntax. The httpRequest.queryParameters object contains all query parameters from the URL. Since username is a single query parameter, you access it directly as httpRequest.queryParameters.username. This returns the value john.smith as a text string, which can then be used in the Web API logic (e.g., to query a user record). Appian's expression language treats query parameters as key-value pairs under queryParameters, making this the standard approach.

Option A (httpRequest.queryParameters.users.username):
This is incorrect. The users part suggests a nested structure (e.g., users as a parameter containing a username subfield), which does not match the URL. The URL only defines username as a top-level query parameter, not a nested object.
Option B (httpRequest.users.username):
This is invalid. The httpRequest object does not have a direct users property. Query parameters are accessed via queryParameters, and there's no indication of a users object in the URL or Appian's Web API model.
Option C (httpRequest.formData.username):
This is incorrect. The httpRequest.formData object is used for parameters passed in the body of a POST or PUT request (e.g., form submissions), not for query parameters in a GET request URL. Since the username is part of the query string (?username=john.smith), formData does not apply.
The correct syntax leverages Appian's standard handling of query parameters, ensuring the Web API can process the username value effectively.

質問 # 35
You are designing a process that is anticipated to be executed multiple times a day. This process retrieves data from an external system and then calls various utility processes as needed. The main process will not use the results of the utility processes, and there are no user forms anywhere.
Which design choice should be used to start the utility processes and minimize the load on the execution engines?

- A. Start the utility processes via a subprocess synchronously.
- B. Use Process Messaging to start the utility process.
- C. Use the Start Process Smart Service to start the utility processes.
- D. Start the utility processes via a subprocess asynchronously.

正解：D

解説：
Comprehensive and Detailed In-Depth Explanation:
As an Appian Lead Developer, designing a process that executes frequently (multiple times a day) and calls utility processes without using their results requires optimizing performance and minimizing load on Appian's execution engines. The absence of user forms indicates a backend process, so user experience isn't a concern-only engine efficiency matters. Let's evaluate each option:
A . Use the Start Process Smart Service to start the utility processes:
The Start Process Smart Service launches a new process instance independently, creating a separate process in the Work Queue. While functional, it increases engine load because each utility process runs as a distinct instance, consuming engine resources and potentially clogging the Java Work Queue, especially with frequent executions. Appian's performance guidelines discourage unnecessary separate process instances for utility tasks, favoring integrated subprocesses, making this less optimal.
B . Start the utility processes via a subprocess synchronously:
Synchronous subprocesses (e.g., a!startProcess with isAsync: false) execute within the main process flow, blocking until completion. For utility processes not used by the main process, this creates unnecessary delays, increasing execution time and engine load. With frequent daily executions, synchronous subprocesses could strain engines, especially if utility processes are slow or numerous. Appian's documentation recommends asynchronous execution for non-dependent, non-blocking tasks, ruling this out.
C . Use Process Messaging to start the utility process:
Process Messaging (e.g., sendMessage() in Appian) is used for inter-process communication, not for starting processes. It's designed to pass data between running processes, not initiate new ones. Attempting to use it for starting utility processes would require additional setup (e.g., a listening process) and isn't a standard or efficient method. Appian's messaging features are for coordination, not process initiation, making this inappropriate.
D . Start the utility processes via a subprocess asynchronously:
This is the best choice. Asynchronous subprocesses (e.g., a!startProcess with isAsync: true) execute independently of the main process, offloading work to the engine without blocking or delaying the parent process. Since the main process doesn't use the utility process results and there are no user forms, asynchronous execution minimizes engine load by distributing tasks across time, reducing Work Queue pressure during frequent executions. Appian's performance best practices recommend asynchronous subprocesses for non-dependent, utility tasks to optimize engine utilization, making this ideal for minimizing load.
Conclusion: Starting the utility processes via a subprocess asynchronously (D) minimizes engine load by allowing independent execution without blocking the main process, aligning with Appian's performance optimization strategies for frequent, backend processes.
Appian Documentation: "Process Model Performance" (Synchronous vs. Asynchronous Subprocesses).
Appian Lead Developer Certification: Process Design Module (Optimizing Engine Load).
Appian Best Practices: "Designing Efficient Utility Processes" (Asynchronous Execution).

質問 # 36
Your Appian project just went live with the following environment setup: DEV > TEST (SIT/UAT) > PROD. Your client is considering adding a support team to manage production defects and minor enhancements, while the original development team focuses on Phase 2. Your client is asking you for a new environment strategy that will have the least impact on Phase 2 development work. Which option involves the lowest additional server cost and the least code retrofit effort?

- A. Phase 2 development work stream: DEV > TEST (SIT/UAT) > PROD
- B. Phase 2 development work stream: DEV > TEST (SIT) > STAGE (UAT) > PROD
- C. Phase 2 development work stream: DEV > TEST (SIT/UAT) > PROD
- D. Phase 2 development work stream: DEV > TEST (SIT) > STAGE (UAT) > PROD

正解： A

解説：
Comprehensive and Detailed In-Depth Explanation:
The goal is to design an environment strategy that minimizes additional server costs and code retrofit effort while allowing the support team to manage production defects and minor enhancements without disrupting the Phase 2 development team. The current setup (DEV > TEST (SIT/UAT) > PROD) uses a single development and testing pipeline, and the client wants to segregate support activities from Phase 2 development. Appian's Environment Management Best Practices emphasize scalability, cost efficiency, and minimal refactoring when adjusting environments.
Option C (Phase 2 development work stream: DEV > TEST (SIT/UAT) > PROD; Production support work stream: DEV > TEST2 (SIT/UAT) > PROD):
This option is the most cost-effective and requires the least code retrofit effort. It leverages the existing DEV environment for both teams but introduces a separate TEST2 environment for the support team's SIT/UAT activities. Since DEV is already shared, no new development server is needed, minimizing server costs. The existing code in DEV and TEST can be reused for TEST2 by exporting and importing packages, with minimal adjustments (e.g., updating environment-specific configurations). The Phase 2 team continues using the original TEST environment, avoiding disruption. Appian supports multiple test environments branching from a single DEV, and the PROD environment remains shared, aligning with the client's goal of low impact on Phase 2. The support team can handle defects and enhancements in TEST2 without interfering with development workflows.
Option A (Phase 2 development work stream: DEV > TEST (SIT) > STAGE (UAT) > PROD; Production support work stream: DEV > TEST2 (SIT/UAT) > PROD):
This introduces a STAGE environment for UAT in the Phase 2 stream, adding complexity and potentially requiring code updates to accommodate the new environment (e.g., adjusting deployment scripts). It also requires a new TEST2 server, increasing costs compared to Option C, where TEST2 reuses existing infrastructure.
Option B (Phase 2 development work stream: DEV > TEST (SIT) > STAGE (UAT) > PROD; Production support work stream: DEV2 > STAGE (SIT/UAT) > PROD):
This option adds both a DEV2 server for the support team and a STAGE environment, significantly increasing server costs. It also requires refactoring code to support two development environments (DEV and DEV2), including duplicating or synchronizing objects, which is more effort than reusing a single DEV.
Option D (Phase 2 development work stream: DEV > TEST (SIT/UAT) > PROD; Production support work stream: DEV2 > TEST (SIT/UAT) > PROD):
This introduces a DEV2 server for the support team, adding server costs. Sharing the TEST environment between teams could lead to conflicts (e.g., overwriting test data), potentially disrupting Phase 2 development. Code retrofit effort is higher due to managing two DEV environments and ensuring TEST compatibility.
Cost and Retrofit Analysis:
Server Cost: Option C avoids new DEV or STAGE servers, using only an additional TEST2, which can often be provisioned on existing hardware or cloud resources with minimal cost. Options A, B, and D require additional servers (TEST2, DEV2, or STAGE), increasing expenses.
Code Retrofit: Option C minimizes changes by reusing DEV and PROD, with TEST2 as a simple extension. Options A and B require updates for STAGE, and B and D involve managing multiple DEV environments, necessitating more significant refactoring.
Appian's recommendation for environment strategies in such scenarios is to maximize reuse of existing infrastructure and avoid unnecessary environment proliferation, making Option C the optimal choice.

質問 # 37
What are two advantages of having High Availability (HA) for Appian Cloud applications?

- A. In the event of a system failure, your Appian instance will be restored and available to your users in less than 15 minutes, having lost no more than the last 1 minute worth of data.
- B. An Appian Cloud HA instance is composed of multiple active nodes running in different availability zones in different regions.

- C. A typical Appian Cloud HA instance is composed of two active nodes.
- D. Data and transactions are continuously replicated across the active nodes to achieve redundancy and avoid single points of failure.

正解：A、D

解説：
Comprehensive and Detailed In-Depth Explanation:
High Availability (HA) in Appian Cloud is designed to ensure that applications remain operational and data integrity is maintained even in the face of hardware failures, network issues, or other disruptions. Appian's Cloud Architecture and HA documentation outline the benefits, focusing on redundancy, minimal downtime, and data protection. The question asks for two advantages, and the options must align with these core principles.
Option B (Data and transactions are continuously replicated across the active nodes to achieve redundancy and avoid single points of failure):
This is a key advantage of HA. Appian Cloud HA instances use multiple active nodes to replicate data and transactions in real-time across the cluster. This redundancy ensures that if one node fails, others can take over without data loss, eliminating single points of failure. This is a fundamental feature of Appian's HA setup, leveraging distributed architecture to enhance reliability, as detailed in the Appian Cloud High Availability Guide.
Option D (In the event of a system failure, your Appian instance will be restored and available to your users in less than 15 minutes, having lost no more than the last 1 minute worth of data):
This is another significant advantage. Appian Cloud HA is engineered to provide rapid recovery and minimal data loss. The Service Level Agreement (SLA) and HA documentation specify that in the case of a failure, the system failover is designed to complete within a short timeframe (typically under 15 minutes), with data loss limited to the last minute due to synchronous replication. This ensures business continuity and meets stringent uptime and data integrity requirements.
Option A (An Appian Cloud HA instance is composed of multiple active nodes running in different availability zones in different regions):
This is a description of the HA architecture rather than an advantage. While running nodes across different availability zones and regions enhances fault tolerance, the benefit is the resulting redundancy and availability, which are captured in Options B and D. This option is more about implementation than a direct user or operational advantage.
Option C (A typical Appian Cloud HA instance is composed of two active nodes):
This is a factual statement about the architecture but not an advantage. The number of nodes (typically two or more, depending on configuration) is a design detail, not a benefit. The advantage lies in what this setup enables (e.g., redundancy and quick recovery), as covered by B and D.
The two advantages-continuous replication for redundancy (B) and fast recovery with minimal data loss (D)-reflect the primary value propositions of Appian Cloud HA, ensuring both operational resilience and data integrity for users.
The two advantages of having High Availability (HA) for Appian Cloud applications are:
B . Data and transactions are continuously replicated across the active nodes to achieve redundancy and avoid single points of failure. This is an advantage of having HA, as it ensures that there is always a backup copy of data and transactions in case one of the nodes fails or becomes unavailable. This also improves data integrity and consistency across the nodes, as any changes made to one node are automatically propagated to the other node.
D). In the event of a system failure, your Appian instance will be restored and available to your users in less than 15 minutes, having lost no more than the last 1 minute worth of data. This is an advantage of having HA, as it guarantees a high level of service availability and reliability for your Appian instance. If one of the nodes fails or becomes unavailable, the other node will take over and continue to serve requests without any noticeable downtime or data loss for your users.

質問 # 38
.....

の高い合格率は99％～100％であるため、ACD-301最新の質問を購入して実践することを後悔しないと信じています。

だがその凛々しさも所詮は五体が揃ってのことであったのかもしれなかっACD-301た、他に何千町歩という農場や牧場も持っているわけだ、こうやってすれば、時間とエネルギーを無駄にするだけでなく、失敗になるかもしれません。

試験の準備方法-ユニークなACD-301認定資格試験試験-素敵なACD-301対応資料

疑いがある方々が無料でデモをダウンロードして試用しても大丈夫です、ACD-301トレーニング資料は、その素晴らしい品質のためにあなたを決して失望させません、当社は簡単に後退しません、MogiExamのAppianのACD-301試験トレーニング資料を持つことは明るい未来を持つことと同じです。

- 検証するACD-301認定資格試験 - 合格スムーズACD-301対応資料 | ハイパスレートのACD-301トレーニング □ ➽ www.japancert.com □を入力して▸ ACD-301 ◂を検索し、無料でダウンロードしてくださいACD-301受験対策
- 検証するACD-301認定資格試験 - 合格スムーズACD-301対応資料 | ハイパスレートのACD-301トレーニング □ 時間限定無料で使える☀ ACD-301 □☀□の試験問題は“www.goshiken.com”サイトで検索ACD-301試験問題集
- 有難い-検証するACD-301認定資格試験試験-試験の準備方法ACD-301対応資料 □ 今すぐ☀ www.passtest.jp □☀□を開き、{ ACD-301 }を検索して無料でダウンロードしてくださいACD-301試験問題集
- ACD-301試験問題集 □ ACD-301試験問題集 □ ACD-301模試エンジン □ ✔ www.goshiken.com □✔□サイトで（ ACD-301 ）の最新問題が使えるACD-301模試エンジン
- ACD-301日本語試験対策 □ ACD-301独学書籍 □ ACD-301オンライン試験 □ 今すぐ➡ www.mogiexam.com □□□で➥ ACD-301 □を検索して、無料でダウンロードしてくださいACD-301テスト内容
- ACD-301試験準備 □ ACD-301認定資格試験問題集 □ ACD-301実際試験 □ 今すぐ⇛ www.goshiken.com⇚を開き、[ACD-301]を検索して無料でダウンロードしてくださいACD-301受験対策
- ACD-301最新な問題集 □ ACD-301オンライン試験 □ ACD-301最新日本語版参考書 □□ ➽ www.passtest.jp □は、“ ACD-301 ”を無料でダウンロードするのに最適なサイトですACD-301最新日本語版参考書
- 現在あちこちでAppian ACD-301認定試験の問題集を探しているのか □ { www.goshiken.com }サイトにて最新{ ACD-301 }問題集をダウンロードACD-301日本語試験対策
- 現在あちこちでAppian ACD-301認定試験の問題集を探しているのか □ { www.passtest.jp }サイトにて《 ACD-301 》問題集を無料で使おうACD-301日本語試験対策
- 効果的-一番優秀なACD-301認定資格試験試験-試験の準備方法ACD-301対応資料 □ ➽ www.goshiken.com □□から➥ ACD-301 □を検索して、試験資料を無料でダウンロードしてくださいACD-301日本語版対応参考書
- ACD-301日本語pdf問題 □ ACD-301試験対応 □ ACD-301関連復習問題集 □ Open Webサイト[www.jptestking.com]検索➠ ACD-301 □無料ダウンロードACD-301ブロンズ教材
- apriljadi356723.creacionblog.com, base-directory.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, royalbookmarking.com, pr6bookmark.com, thebookpage.com, ztndz.com, bookmarkrange.com, brontemszh397309.webbuzzfeed.com, macrobookmarks.com, Disposable vapes

BONUS！！！ MogiExam ACD-301ダンプの一部を無料でダウンロード：https://drive.google.com/open?id=1H0SkkcJHuFbPeES_VTzREr-nCtPsjb7p