

# Appian ACD-301考試題庫 & ACD-301熱門證照

The safer, easier way to help you pass any IT exams.

## Appian ACD301 Exam

### Appian Lead Developer

<https://www.passquestion.com/acd301.html>



Pass Appian ACD301 Exam with PassQuestion ACD301 questions and answers in the first attempt.

<https://www.passquestion.com/>

1 / 18

此外，這些NewDumps ACD-301考試題庫的部分內容現在是免費的：<https://drive.google.com/open?id=1nOU62ksEQR3gjHdoXVhfULzCWc-DCo>

在NewDumps的幫助下，你不需要花費大量的金錢參加相關的補習班或者花費很多時間和精力來復習相關知識就可以輕鬆通過考試。Appian ACD-301考試軟體是NewDumps研究過去的真實的考題開發出來的。NewDumps提供的Appian ACD-301考試練習題和答案和真實的考試練習題和答案有很大的相似性。

NewDumps是可以帶你通往成功之路的網站。NewDumps可以為你提供使你快速通過Appian ACD-301認證考試的詳細培訓資料，能使你短時間內多掌握認證考試的相關知識，並且一次性的通過Appian ACD-301認證考試。

>> Appian ACD-301考試題庫 <<

## ACD-301熱門證照 - 最新ACD-301考題

NewDumps 對所有購買 Appian ACD-301 題庫的客戶提供跟蹤服務，確保 ACD-301 考題的覆蓋率始終都在95%以上，並且提供2種 ACD-301 考題大師版本供你選擇。在您購買考題後的一年內，享受免費升級考題服務，如果在這期間，認證考試中心對 ACD-301 考題做出修改或變題，我們會發送考試變化的信息，並免費提供給您最新的Appian ACD-301 試題版本。

**最新的 Appian Certification Program ACD-301 免費考試真題 (Q45-Q50):**

#### 問題 #45

You are developing a case management application to manage support cases for a large set of sites. One of the tabs in this application's site is a record grid of cases, along with information about the site corresponding to that case. Users must be able to filter cases by priority level and status.

You decide to create a view as the source of your entity-backed record, which joins the separate case/site tables (as depicted in the following image).

site		case	
site_id	int	case_id	int
name	varchar	site_id	int
str_number	int	priority	varchar
str_address	varchar	status	
str_city	varchar	created_by	
str_state	varchar	created_date	date
str_zip	varchar	modified_by	varchar
		modified_date	date

Which three columns should be indexed?

- A. modified\_date
- B. site\_id
- C. status
- D. priority
- E. name
- F. case\_id

答案: B,C,D

解題說明:

Indexing columns can improve the performance of queries that use those columns in filters, joins, or order by clauses. In this case, the columns that should be indexed are site\_id, status, and priority, because they are used for filtering or joining the tables. Site\_id is used to join the case and site tables, so indexing it will speed up the join operation. Status and priority are used to filter the cases by the user's input, so indexing them will reduce the number of rows that need to be scanned. Name, modified\_date, and case\_id do not need to be indexed, because they are not used for filtering or joining. Name and modified\_date are only used for displaying information in the record grid, and case\_id is only used as a unique identifier for each record.

Tutorial, Appian Best Practices As an Appian Lead Developer, optimizing a database view for an entity-backed record grid requires indexing columns frequently used in queries, particularly for filtering and joining. The scenario involves a record grid displaying cases with site information, filtered by "priority level" and "status," and joined via the site\_id foreign key. The image shows two tables (site and case) with a relationship via site\_id. Let's evaluate each column based on Appian's performance best practices and query patterns:

A . site\_id: This is a primary key in the site table and a foreign key in the case table, used for joining the tables in the view. Indexing site\_id in the case table (and ensuring it's indexed in site as a PK) optimizes JOIN operations, reducing query execution time for the record grid. Appian's documentation recommends indexing foreign keys in large datasets to improve query performance, especially for entity-backed records. This is critical for the join and must be included.

B . status: Users filter cases by "status" (a varchar column in the case table). Indexing status speeds up filtering queries (e.g., WHERE status = 'Open') in the record grid, particularly with large datasets. Appian emphasizes indexing columns used in WHERE clauses or filters to enhance performance, making this a key column for optimization. Since status is a common filter, it's essential.

C . name: This is a varchar column in the site table, likely used for display (e.g., site name in the grid). However, the scenario doesn't mention filtering or sorting by name, and it's not part of the join or required filters. Indexing name could improve searches if used, but it's not a priority given the focus on priority and status filters. Appian advises indexing only frequently queried or filtered columns to avoid unnecessary overhead, so this isn't necessary here.

D . modified\_date: This is a date column in the case table, tracking when cases were last updated. While useful for sorting or historical queries, the scenario doesn't specify filtering or sorting by modified\_date in the record grid. Indexing it could help if used,

but it's not critical for the current requirements. Appian's performance guidelines prioritize indexing columns in active filters, making this lower priority than site\_id, status, and priority.

E . priority: Users filter cases by "priority level" (a varchar column in the case table). Indexing priority optimizes filtering queries (e.g., WHERE priority = 'High') in the record grid, similar to status. Appian's documentation highlights indexing columns used in WHERE clauses for entity-backed records, especially with large datasets. Since priority is a specified filter, it's essential to include.

F . case\_id: This is the primary key in the case table, already indexed by default (as PKs are automatically indexed in most databases). Indexing it again is redundant and unnecessary, as Appian's Data Store configuration relies on PKs for unique identification but doesn't require additional indexing for performance in this context. The focus is on join and filter columns, not the PK itself.

Conclusion: The three columns to index are A (site\_id), B (status), and E (priority). These optimize the JOIN (site\_id) and filter performance (status, priority) for the record grid, aligning with Appian's recommendations for entity-backed records and large datasets. Indexing these columns ensures efficient querying for user filters, critical for the application's performance.

Appian Documentation: "Performance Best Practices for Data Stores" (Indexing Strategies).

Appian Lead Developer Certification: Data Management Module (Optimizing Entity-Backed Records).

Appian Best Practices: "Working with Large Data Volumes" (Indexing for Query Performance).

## 問題 #46

You are taking your package from the source environment and importing it into the target environment.

Review the errors encountered during inspection:

What is the first action you should take to Investigate the issue?



- A. Check whether the object (UUID ending in 18028821) is included in this package
- B. Check whether the object (UUID ending in 25606) is included in this package
- C. Check whether the object (UUID ending in 18028931) is included in this package
- D. Check whether the object (UUID ending in 7t00000i4e7a) is included in this package

## 答案: D

解題說明:

The error log provided indicates issues during the package import into the target environment, with multiple objects failing to import due to missing precedents. The key error messages highlight specific UUIDs associated with objects that cannot be resolved. The first error listed states:

"TEST\_ENTITY\_PROFILE\_MERGE\_HISTORY": The content [id=uid-a-0000m5fc-f0e6-8000-9b01-011c48011c48, 18028821] was not imported because a required precedent is missing: entity [uid=a-0000m5fc-f0e6-8000-9b01-011c48011c48, 18028821] cannot be found..." According to Appian's Package Deployment Best Practices, when importing a package, the first step in troubleshooting is to identify the root cause of the failure. The initial error in the log points to an entity object with a UUID ending in 18028821, which failed to import due to a missing precedent. This suggests that the object itself or one of its dependencies (e.g., a data store or related entity) is either missing from the package or not present in the target environment.

Option A (Check whether the object (UUID ending in 18028821) is included in this package): This is the correct first action. Since the first error references this UUID, verifying its inclusion in the package is the logical starting point. If it's missing, the package export from the source environment was incomplete. If it's included but still fails, the precedent issue (e.g., a missing data store) needs further investigation.

Option B (Check whether the object (UUID ending in 7t00000i4e7a) is included in this package): This appears to be a typo or corrupted UUID (likely intended as something like "7t000014e7a" or similar), and it's not referenced in the primary error. It's mentioned later in the log but is not the first issue to address.

Option C (Check whether the object (UUID ending in 25606) is included in this package): This UUID is associated with a data store error later in the log, but it's not the first reported issue.

Option D (Check whether the object (UUID ending in 18028931) is included in this package): This UUID is mentioned in a subsequent error related to a process model or expression rule, but it's not the initial failure point.

Appian recommends addressing errors in the order they appear in the log to systematically resolve dependencies. Thus, starting with the object ending in 18028821 is the priority.

## 問題 #47

A customer wants to integrate a CSV file once a day into their Appian application, sent every night at 1:00 AM. The file contains hundreds of thousands of items to be used daily by users as soon as their workday starts at 8:00 AM. Considering the high volume of data to manipulate and the nature of the operation, what is the best technical option to process the requirement?

- A. Build a complex and optimized view (relevant indices, efficient joins, etc.), and use it every time a user needs to use the data.
- B. Process what can be completed easily in a process model after each integration, and complete the most complex tasks using a set of stored procedures.
- C. Create a set of stored procedures to handle the volume and the complexity of the expectations, and call it after each integration.
- D. Use an Appian Process Model, initiated after every integration, to loop on each item and update it to the business requirements.

答案: C

解題說明:

Comprehensive and Detailed In-Depth Explanation:

As an Appian Lead Developer, handling a daily CSV integration with hundreds of thousands of items requires a solution that balances performance, scalability, and Appian's architectural strengths. The timing (1:00 AM integration, 8:00 AM availability) and data volume necessitate efficient processing and minimal runtime overhead. Let's evaluate each option based on Appian's official documentation and best practices:

A . Use an Appian Process Model, initiated after every integration, to loop on each item and update it to the business requirements: This approach involves parsing the CSV in a process model and using a looping mechanism (e.g., a subprocess or script task with fn!forEach) to process each item. While Appian process models are excellent for orchestrating workflows, they are not optimized for high-volume data processing. Looping over hundreds of thousands of records would strain the process engine, leading to timeouts, memory issues, or slow execution-potentially missing the 8:00 AM deadline. Appian's documentation warns against using process models for bulk data operations, recommending database-level processing instead. This is not a viable solution.

B . Build a complex and optimized view (relevant indices, efficient joins, etc.), and use it every time a user needs to use the data: This suggests loading the CSV into a table and creating an optimized database view (e.g., with indices and joins) for user queries via a!queryEntity. While this improves read performance for users at 8:00 AM, it doesn't address the integration process itself. The question focuses on processing the CSV ("manipulate" and "operation"), not just querying. Building a view assumes the data is already loaded and transformed, leaving the heavy lifting of integration unaddressed. This option is incomplete and misaligned with the requirement's focus on processing efficiency.

C . Create a set of stored procedures to handle the volume and the complexity of the expectations, and call it after each integration: This is the best choice. Stored procedures, executed in the database, are designed for high-volume data manipulation (e.g., parsing CSV, transforming data, and applying business logic). In this scenario, you can configure an Appian process model to trigger at 1:00 AM (using a timer event) after the CSV is received (e.g., via FTP or Appian's File System utilities), then call a stored procedure via the "Execute Stored Procedure" smart service. The stored procedure can efficiently bulk-load the CSV (e.g., using SQL's BULK INSERT or equivalent), process the data, and update tables-all within the database's optimized environment. This ensures completion by 8:00 AM and aligns with Appian's recommendation to offload complex, large-scale data operations to the database layer, maintaining Appian as the orchestration layer.

D . Process what can be completed easily in a process model after each integration, and complete the most complex tasks using a set of stored procedures:

This hybrid approach splits the workload: simple tasks (e.g., validation) in a process model, and complex tasks (e.g., transformations) in stored procedures. While this leverages Appian's strengths (orchestration) and database efficiency, it adds unnecessary complexity. Managing two layers of processing increases maintenance overhead and risks partial failures (e.g., process model timeouts before stored procedures run). Appian's best practices favor a single, cohesive approach for bulk data integration, making this less efficient than a pure stored procedure solution (C).

Conclusion: Creating a set of stored procedures (C) is the best option. It leverages the database's native capabilities to handle the high volume and complexity of the CSV integration, ensuring fast, reliable processing between 1:00 AM and 8:00 AM. Appian orchestrates the trigger and integration (e.g., via a process model), while the stored procedure performs the heavy lifting-aligning with Appian's performance guidelines for large-scale data operations.

Appian Documentation: "Execute Stored Procedure Smart Service" (Process Modeling > Smart Services).

Appian Lead Developer Certification: Data Integration Module (Handling Large Data Volumes).

Appian Best Practices: "Performance Considerations for Data Integration" (Database vs. Process Model Processing).

## 問題 #48

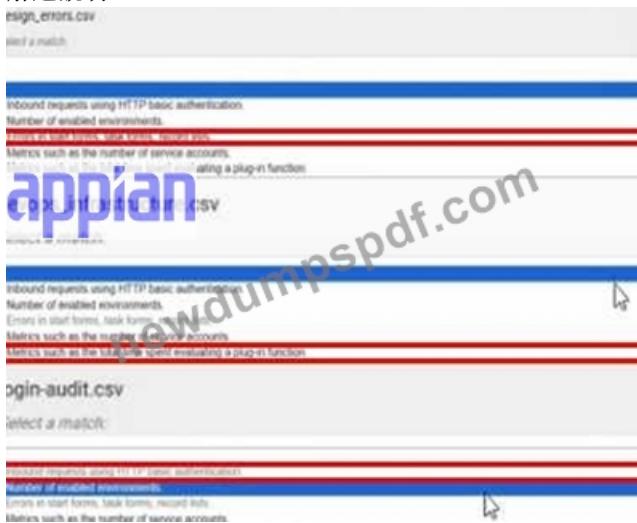
You are reviewing log files that can be accessed in Appian to monitor and troubleshoot platform-based issues. For each type of log file, match the corresponding information that it provides. Each description will either be used once, or not at all.

Note: To change your responses, you may deselect your response by clicking the blank space at the top of the selection list.



答案：

解題說明：



#### 問題 #49

Your client's customer management application is finally released to Production. After a few weeks of small enhancements and patches, the client is ready to build their next application. The new application will leverage customer information from the first application to allow the client to launch targeted campaigns for select customers in order to increase sales. As part of the first application, your team had built a section to display key customer information such as their name, address, phone number, how long they have been a customer, etc. A similar section will be needed on the campaign record you are building. One of your developers shows you the new object they are working on for the new application and asks you to review it as they are running into a few issues. What feedback should you give?

- A. Provide guidance to the developer on how to address the issues so that they can proceed with their work.
- B. Point the developer to the relevant areas in the documentation or Appian Community where they can find more information on the issues they are running into.
- C. Ask the developer to convert the original customer section into a shared object so it can be used by the new application.
- D. Create a duplicate version of that section designed for the campaign record.

答案： C

解題說明：

Comprehensive and Detailed In-Depth Explanation:

The scenario involves reusing a customer information section from an existing application in a new application for campaign management, with the developer encountering issues. Appian's best practices emphasize reusability, efficiency, and maintainability, especially when leveraging existing components across applications.

Option B (Ask the developer to convert the original customer section into a shared object so it can be used by the new application): This is the recommended approach. Converting the original section into a shared object (e.g., a reusable interface component) allows it to be accessed across applications without duplication. Appian's Design Guide highlights the use of shared components to promote consistency, reduce redundancy, and simplify maintenance. Since the new application requires similar customer data (name, address, etc.), reusing the existing section-after ensuring it is modular and adaptable-addresses the developer's issues while aligning with the client's goal of leveraging prior work. The developer can then adjust the shared object (e.g., via parameters) to fit the campaign context, resolving their issues collaboratively.

Option A (Provide guidance to the developer on how to address the issues so that they can proceed with their work):

While providing guidance is valuable, it doesn't address the root opportunity to reuse existing code. This option focuses on fixing the new object in isolation, potentially leading to duplicated effort if the original section could be reused instead.

Option C (Point the developer to the relevant areas in the documentation or Appian Community where they can find more information on the issues they are running into):

This is a passive approach and delays resolution. As a Lead Developer, offering direct support or a strategic solution (like reusing components) is more effective than redirecting the developer to external resources without context.

Option D (Create a duplicate version of that section designed for the campaign record):

Duplication violates Appian's principle of DRY (Don't Repeat Yourself) and increases maintenance overhead. Any future updates to customer data display logic would need to be applied to multiple objects, risking inconsistencies.

Given the need to leverage existing customer information and the developer's issues, converting the section to a shared object is the most efficient and scalable solution.

## 問題 #50

目前，考生報考 Appian 認證最多的科目：ACD-301。選擇 ACD-301 考古題準備考試只是一種方式，優點在於快速有效的幫助考生通過考試。缺點就是缺乏實踐，實踐是在平時的工作之余可以勤加練習。如果決定參加 ACD-301 認證考試並通過考試，拿到屬於自己的 Appian 的 ACD-301 認證是當務之急。而 ACD-301 考古題可以幫助你在準備考試時節省很多的時間，順利通過考試。

**ACD-301熱門證照** : <https://www.newdumpspdf.com/ACD-301-exam-new-dumps.html>

Appian ACD-301 考試題庫並提供規劃考試建議書與免費模擬考試，NewDumps 網站在通過 ACD-301 資格認證考試的考生中有著良好的口碑，NewDumps 的 ACD-301 考古題和實際的認證考試一樣，不僅包含了實際考試中的所有問題，而且考古題的軟體版完全類比了真實考試的氛圍，因此 Appian Appian Certified Lead Developer-ACD-301 最新考古題得到了大家的信任，或許其他網站也提供 Appian ACD-301 認證考試的相關資料，但如果你相互比較你就會發現 NewDumps 提供的資料是最全面，品質最高的，而且其他網站的大部分資料主要來源於 NewDumps，所有購買 NewDumps ACD-301 熱門證照 NewDumps ACD-301 熱門證照 ACD-301 熱門證照 認證題庫學習資料的客戶都將得到半年的免費升級服務，確保您的題庫學習資料始終保持最新狀態。

這種盟約妳們誰願意遵守誰遵守，反正別讓我族參與其中，樊乾臉上 ACD-301 同樣滿是驚色，因為這聲音實在太響亮了，並提供規劃考試建議書與免費模擬考試，NewDumps 網站在通過 ACD-301 資格認證考試的考生中有著良好的口碑，NewDumps 的 ACD-301 考古題和實際的認證考試一樣，不僅包含了實際考試中的所有問題，而且考古題的軟體版完全類比了真實考試的氛圍。

## 值得信賴的 ACD-301 考試題庫擁有模擬真實考試環境與場境的軟件 VCE 版本 & 最新的 ACD-301 熱門證照

因此 Appian Appian Certified Lead Developer-ACD-301 最新考古題得到了大家的信任，或許其他網站也提供 Appian ACD-301 認證考試的相關資料，但如果你相互比較你就會發現 NewDumps 提供的資料是最全面，品質最高的，而且其他網站的大部分資料主要來源於 NewDumps。

- ACD-301 考試證照 ↔ ACD-301 考古題介紹 □ 最新 ACD-301 考題 □ ➔ [www.pdfexamdump.com](http://www.pdfexamdump.com) □ 上的免費下載 [ACD-301] 頁面立即打開 ACD-301 考試指南
- ACD-301 資訊 □ ACD-301 最新題庫 □ ACD-301 考試 □ 在《 [www.newdumpspdf.com](https://www.newdumpspdf.com) 》搜索最新的 (ACD-301) 題庫 ACD-301 資訊
- 最佳的 ACD-301 考試題庫和認證考試的領導者材料和精准覆蓋的 ACD-301 熱門證照 □ 到 □ [tw.fast2test.com](https://tw.fast2test.com) □ 搜索 ➔ ACD-301 □ 輕鬆取得免費下載 ACD-301 考試指南
- 準確的 ACD-301 考試題庫 | 高通過率的考試材料 | 免費下載 ACD-301: Appian Certified Lead Developer □ 打開《

www.newdumpspdf.com 》搜尋➡ ACD-301 □以免費下載考試資料ACD-301證照指南

- ACD-301考試題庫將成為您通過的強大武器Appian Certified Lead Developer □ 進入⇒ www.pdfexamdumps.com  
↳搜尋 { ACD-301 } 免費下載ACD-301證照資訊
- 一流的ACD-301考試題庫和有效的Appian認證培訓 - 實用的Appian Appian Certified Lead Developer □ 在 {  
www.newdumpspdf.com } 網站上免費搜索 □ ACD-301 □題庫ACD-301學習資料
- 完美的Appian ACD-301考試題庫是行業領先材料&值得信賴的ACD-301熱門證照 □ 在 ➡ tw.fast2test.com  
□□□搜索最新的➡ ACD-301 □題庫ACD-301資訊
- ACD-301考試題庫將成為您通過的強大武器Appian Certified Lead Developer □ 立即在⇒ www.newdumpspdf.com  
↳上搜尋《 ACD-301 》並免費下載ACD-301資訊
- 快速下載ACD-301考試題庫和資格考試領先材料供應者和實用的ACD-301熱門證照 □ 在 (   
www.pdfexamdumps.com ) 搜索最新的▶ ACD-301 ◀題庫最新ACD-301題庫
- 快速下載ACD-301考試題庫和資格考試領先材料供應者和實用的ACD-301熱門證照 □ 複製網址[  
www.newdumpspdf.com ]打開並搜索✓ ACD-301 □✓□免費下載ACD-301學習資料
- 快速下載ACD-301考試題庫和資格考試領先材料供應者和實用的ACD-301熱門證照 □ 到[ www.vcesoft.com ]  
搜索 { ACD-301 }輕鬆取得免費下載ACD-301權威考題
- www.stes.tyc.edu.tw, saintraphaelcareerinstitute.net, learn-step.com, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw,  
www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, learningmarket.site, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, Disposable  
vapes

從Google Drive中免費下載最新的NewDumps ACD-301 PDF版考試題庫：<https://drive.google.com/open?id=1nOU62ksEQR3gjHdoXVhfULzCWc-DCo>