ACD301인기자격증시험덤프 & ACD301인기자격증덤프문제

Itexamdump ACD301 최신 PDF 버전 시험 문제집을 무료로 Google Drive에서 다운로드하세요: https://drive.google.com/open?id=1mYEkkNiTatVbP7n4_0uL854II3q7Tqw8

어떻게 하면 가장 편하고 수월하게 Appian ACD301시험을 패스할수 있을가요? 그 답은 바로 Itexamdump에서 찾아볼수 있습니다. Appian ACD301덤프로 시험에 도전해보지 않으실래요? Itexamdump는 당신을 위해Appian ACD301덤프로Appian ACD301인증시험이라는 높은 벽을 순식간에 무너뜨립니다.

Itexamdump의 Appian 인증 ACD301시험덤프공부자료는 pdf버전과 소프트웨어버전 두가지 버전으로 제공되는데 Appian 인증 ACD301실제시험예상문제가 포함되어있습니다.덤프의 예상문제는 Appian 인증 ACD301실제시험의 대부분 문제를 적중하여 높은 통과율과 점유율을 자랑하고 있습니다. Itexamdump의 Appian 인증 ACD301덤프를 선택하시면 IT자격증 취득에 더할것 없는 힘이 될것입니다.

>> ACD301인기자격증 시험덤프 <<

ACD301인기자격증 덤프문제 - ACD301퍼펙트 최신 덤프

Appian ACD301 덤프는 Appian ACD301 시험의 모든 문제를 커버하고 있어 시험적중율이 아주 높습니다. Itexamdump는 Paypal과 몇년간의 파트너 관계를 유지하여 왔으므로 신뢰가 가는 안전한 지불방법을 제공해드립니다. Appian ACD301시험탈락시 제품비용 전액환불조치로 고객님의 이익을 보장해드립니다.

Appian ACD301 시험요강:

주제	소개
주제 1	• Application Design and Development: This section of the exam measures skills of Lead Appian Developers and covers the design and development of applications that meet user needs using Appian functionality. It includes designing for consistency, reusability, and collaboration across teams. Emphasis is placed on applying best practices for building multiple, scalable applications in complex environments.
주제 2	• Extending Appian: This section of the exam measures skills of Integration Specialists and covers building and troubleshooting advanced integrations using connected systems and APIs. Candidates are expected to work with authentication, evaluate plug-ins, develop custom solutions when needed, and utilize document generation options to extend the platform's capabilities.

주제 3	• Project and Resource Management: This section of the exam measures skills of Agile Project Leads and covers interpreting business requirements, recommending design options, and leading Agile teams through technical delivery. It also involves governance, and process standardization.

최신 Lead Developer ACD301 무료샘플문제 (Q30-Q35):

질문 # 30
You are deciding the appropriate process model data management strategy.
For each requirement. match the appropriate strategies to implement. Each strategy will be used once.
Note: To change your responses, you may deselect your response by clicking the blank space at the top of the selection list.

정답:

설명:

Explanation:
* Archive processes 2 days after completion or cancellation. # Processes that need to be available for 2 days after completion or cancellation, after which are no longer required nor accessible.
* Use system default (currently: auto-archive processes 7 days after completion or cancellation). # Processes that remain available for 7 days after completion or cancellation, after which remain accessible.
* Delete processes 2 days after completion or cancellation. # Processes that need to be available for 2 days after completion or cancellation, after which remain accessible.
* Do not automatically clean-up processes. # Processes that need remain available without the need to unarchive.
Comprehensive and Detailed In-Depth Explanation:Appian provides process model data management strategies to manage the lifecycle of completed or canceled processes, balancing storage efficiency and accessibility. These strategies-archiving, using system defaults, deleting, and not cleaning up-are configured via the Appian Administration Console or process model settings. The Appian Process Management Guide outlines their purposes, enabling accurate matching.
* Archive processes 2 days after completion or cancellation # Processes that need to be available for
2 days after completion or cancellation, after which are no longer required nor accessible:
Archiving moves processes to a compressed, off-line state after a specified period, freeing up active resources. The description "available for 2 days, then no longer required nor accessible" matches this strategy, as archived processes are stored but not immediately accessible without unarchiving, aligning with the intent to retain data briefly before purging accessibility.
* Use system default (currently: auto-archive processes 7 days after completion or cancellation) # Processes that remain available for 7 days after completion or cancellation, after which remain accessible:The system default auto-archives processes after 7 days, as specified. The description
"remainavailable for 7 days, then remain accessible" fits this, indicating that processes are kept in an active state for 7 days before being archived, after which they can still be accessed (e.g., via unarchiving), matching the default behavior.
* Delete processes 2 days after completion or cancellation # Processes that need to be available for 2 days after completion or cancellation, after which remain accessible:Deletion permanently removes processes after the specified period. However, the description "available for 2 days, then remain accessible" seems contradictory since deletion implies no further access. This appears to be a misinterpretation in the options. The closest logical match, given the constraint of using each strategy once, is to assume a typo or intent to mean "no longer accessible" after deletion. However, strictly interpreting the image, no perfect match exists. Based on context, "remain accessible" likely should be
"no longer accessible," but I'll align with the most plausible intent: deletion after 2 days fits the "no longer required" aspect, though accessibility is lost post-deletion.
* Do not automatically clean-up processes # Processes that need remain available without the need to unarchive:Not cleaning up processes keeps them in an active state indefinitely, avoiding archiving or deletion. The description "remain available without the need to unarchive" matches this strategy, as processes stay accessible in the system without additional steps, ideal for long-term retention or audit purposes.
Matching Rationale:
* Each strategy is used once, as required. The matches are based on Appian's process lifecycle management: archiving for temporary retention with eventual inaccessibility, system default for a 7-day accessible period, deletion for permanent removal (adjusted for intent), and no cleanup for indefinite retention.
* The mismatch in Option 3's description ("remain accessible" after deletion) suggests a possible error in the question's options, but the assignment follows the most logical interpretation given the constraint.
References:Appian Documentation - Process Management Guide, Appian Administration Console - Process Model Settings, Appian Lead Developer Training - Data Management Strategies.

질문 # 31
You are just starting with a new team that has been working together on an application for months. They ask you to review some of their views that have been degrading in performance. The views are highly complex with hundreds of lines of SQL. What is the first step in troubleshooting the degradation?

- A. Go through the entire database structure to obtain an overview, ensure you understand the business needs, and then normalize the tables to optimize performance.
- B. Browse through the tables, note any tables that contain a large volume of null values, and work with your team to plan for table restructure.
- C. Run an explain statement on the views, identify critical areas of improvement that can be remediated without business knowledge.
- D. Go through all of the tables one by one to identify which of the grouped by, ordered by, or joined keys are currently indexed.

정답： C

설명:
Comprehensive and Detailed In-Depth Explanation:Troubleshooting performance degradation in complex SQL views within an Appian application requires a systematic approach. The views, described as having hundreds of lines of SQL, suggest potential issues with query execution, indexing, or join efficiency. As a new team member, the first step should focus on quickly identifying the root cause without overhauling the system prematurely. Appian's Performance Troubleshooting Guide and database optimization best practices provide the framework for this process.
* Option B (Run an explain statement on the views, identify critical areas of improvement that can be remediated without business knowledge):This is the recommended first step. Running an EXPLAIN statement (or equivalent, such as EXPLAIN PLAN in some databases) analyzes the query execution plan, revealing details like full table scans, missing indices, or inefficient joins. This technical analysis can identify immediate optimization opportunities (e.g., adding indices or rewriting subqueries) without requiring business input, allowing you to address low-hanging fruit quickly. Appian encourages using database tools to diagnose performance issues before involving stakeholders, making this a practical starting point as you familiarize yourself with the application.
* Option A (Go through the entire database structure to obtain an overview, ensure you understand the business needs, and then normalize the tables to optimize performance):This is too broad and time-consuming as a first step. Understanding business needs and normalizing tables are valuable but require collaboration with the team and stakeholders, delaying action. It's better suited for a later phase after initial technical analysis.
* Option C (Go through all of the tables one by one to identify which of the grouped by, ordered by, or joined keys are currently indexed):Manually checking indices is useful but inefficient without first knowing which queries are problematic. The EXPLAIN statement provides targeted insights into index usage, making it a more direct initial step than a manual table-by-table review.
* Option D (Browse through the tables, note any tables that contain a large volume of null values, and work with your team to plan for table restructure):Identifying null values and planning restructures is a long-term optimization strategy, not a first step. It requires team input and may not address the immediate performance degradation, which is better tackled with query-level diagnostics.
Starting with an EXPLAIN statement allows you to gather data-driven insights, align with Appian's performance troubleshooting methodology, and proceed with informed optimizations.
References:Appian Documentation - Performance Troubleshooting Guide, Appian Lead Developer Training
- Database Optimization, MySQL/PostgreSQL Documentation - EXPLAIN Statement.

질문 # 32
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).
Which three column should be indexed?

- A. name
- B. priority
- C. site_id
- D. case_id
- E. status
- F. modified_date

정답： B,C,E

설명:
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.
Verified References: Appian Records 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.
References:
* 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).

질문 # 33
You are planning a strategy around data volume testing for an Appian application that queries and writes to a MySQL database. You have administrator access to the Appian application and to the database. What are two key considerations when designing a data volume testing strategy?

- A. Large datasets must be loaded via Appian processes.
- B. The amount of data that needs to be populated should be determined by the project sponsor and the stakeholders based on their estimation.
- C. Data from previous tests needs to remain in the testing environment prior to loading prepopulated data.
- D. Data model changes must wait until towards the end of the project.
- E. Testing with the correct amount of data should be in the definition of done as part of each sprint.

정답： B,E

설명:
Comprehensive and Detailed In-Depth Explanation:Data volume testing ensures an Appian application performs efficiently under

realistic data loads, especially when interacting with external databases like MySQL. As an Appian Lead Developer with administrative access, the focus is on scalability, performance, and iterative validation. The two key considerations are:
* Option C (The amount of data that needs to be populated should be determined by the project sponsor and the stakeholders based on their estimation):Determining the appropriate data volume is critical to simulate real-world usage. Appian's Performance Testing Best Practices recommend collaborating with stakeholders (e.g., project sponsors, business analysts) to define expected data sizes based on production scenarios. This ensures the test reflects actual requirements-like peak transaction volumes or record counts-rather than arbitrary guesses. For example, if the application will handle 1 million records in production, stakeholders must specify this to guide test data preparation.
* Option D (Testing with the correct amount of data should be in the definition of done as part of each sprint):Appian's Agile Development Guide emphasizes incorporating performance testing (including data volume) into the Definition of Done (DoD) for each sprint. This ensures that features are validated under realistic conditions iteratively, preventing late-stage performance issues. With admin access, you can query/write to MySQL and assess query performance or write latency with the specified data volume, aligning with Appian's recommendation to "test early and often."
* Option A (Data from previous tests needs to remain in the testing environment prior to loading prepopulated data):This is impractical and risky. Retaining old test data can skew results, introduce inconsistencies, or violate data integrity (e.g., duplicate keys in MySQL). Best practices advocate for a clean, controlled environment with fresh, prepopulated data per test cycle.
* Option B (Large datasets must be loaded via Appian processes):While Appian processes can load data, this is not a requirement. With database admin access, you can use SQL scripts ortools like MySQL Workbench for faster, more efficient data population, bypassing Appian process overhead.
Appian documentation notes this as a preferred method for large datasets.
* Option E (Data model changes must wait until towards the end of the project):Delaying data model changes contradicts Agile principles and Appian's iterative design approach. Changes should occur as needed throughout development to adapt to testing insights, not be deferred.
References:Appian Lead Developer Training - Performance Testing Best Practices, Appian Documentation - Data Management and Testing Strategies.

질문 # 34

You are running an inspection as part of the first deployment process from TEST to PROD. You receive a notice that one of your objects will not deploy because it is dependent on an object from an application owned by a separate team.
What should be your next step?

- A. Create your own object with the same code base, replace the dependent object in the application, and deploy to PROD.
- B. Check the dependencies of the necessary object. Deploy to PROD if there are few dependencies and it is low risk.
- C. Halt the production deployment and contact the other team for guidance on promoting the object to PROD.
- D. Push a functionally viable package to PROD without the dependencies, and plan the rest of the deployment accordingly with the other team's constraints.

정답： C

설명：
Comprehensive and Detailed In-Depth Explanation:
As an Appian Lead Developer, managing a deployment from TEST to PROD requires careful handling of dependencies, especially when objects from another team's application are involved. The scenario describes a dependency issue during deployment, signaling a need for collaboration and governance. Let's evaluate each option:
A . Create your own object with the same code base, replace the dependent object in the application, and deploy to PROD:
This approach involves duplicating the object, which introduces redundancy, maintenance risks, and potential version control issues. It violates Appian's governance principles, as objects should be owned and managed by their respective teams to ensure consistency and avoid conflicts. Appian's deployment best practices discourage duplicating objects unless absolutely necessary, making this an unsustainable and risky solution.
B . Halt the production deployment and contact the other team for guidance on promoting the object to PROD:
This is the correct step. When an object from another application (owned by a separate team) is a dependency, Appian's deployment process requires coordination to ensure both applications' objects are deployed in sync. Halting the deployment prevents partial deployments that could break functionality, and contacting the other team aligns with Appian's collaboration and governance guidelines. The other team can provide the necessary object version, adjust their deployment timeline, or resolve the dependency, ensuring a stable PROD environment.
C . Check the dependencies of the necessary object. Deploy to PROD if there are few dependencies and it is low risk:
This approach risks deploying an incomplete or unstable application if the dependency isn't fully resolved. Even with "few dependencies" and "low risk," deploying without the other team's object could lead to runtime errors or broken functionality in PROD. Appian's documentation emphasizes thorough dependency management during deployment, requiring all objects (including those from other applications) to be promoted together, making this risky and not recommended.

D . Push a functionally viable package to PROD without the dependencies, and plan the rest of the deployment accordingly with the other team's constraints:
Deploying without dependencies creates an incomplete solution, potentially leaving the application non-functional or unstable in PROD. Appian's deployment process ensures all dependencies are included to maintain application integrity, and partial deployments are discouraged unless explicitly planned (e.g., phased rollouts). This option delays resolution and increases risk, contradicting Appian's best practices for Production stability.
Conclusion: Halting the production deployment and contacting the other team for guidance (B) is the next step. It ensures proper collaboration, aligns with Appian's governance model, and prevents deployment errors, providing a safe and effective resolution.
Reference:
Appian Documentation: "Deployment Best Practices" (Managing Dependencies Across Applications).
Appian Lead Developer Certification: Application Management Module (Cross-Team Collaboration).
Appian Best Practices: "Handling Production Deployments" (Dependency Resolution).

질문 # 35

......

IT자격증을 많이 취득하여 IT업계에서 자신만의 단단한 자리를 보장하는것이 여러분들의 로망이 아닐가 싶습니다. Itexamdump의 완벽한 Appian인증 ACD301덤프는 IT전문가들이 자신만의 노하우와 경험으로 실제Appian인증 ACD301시험문제에 대비하여 연구제작한 완벽한 작품으로서 100%시험통과율을 보장합니다.

ACD301인기자격증 덤프문제: https://www.itexamdump.com/ACD301.html

- ACD301덤프문제집 □ ACD301최고품질 예상문제모음 □ ACD301시험대비 덤프 최신자료 □ □ www.itdumpskr.com □웹사이트에서【 ACD301 】를 열고 검색하여 무료 다운로드ACD301인증시험 인기덤프
- 인기자격증 ACD301인기자격증 시험덤프 시험 덤프자료 □ 무료 다운로드를 위해 지금【 www.itdumpskr.com 】에서▷ ACD301 ◁검색ACD301인증시험공부
- 완벽한 ACD301인기자격증 시험덤프 시험덤프공부 □ ▶ www.koreadumps.com◀에서 검색만 하면【 ACD301 】를 무료로 다운로드할 수 있습니다ACD301최신버전 공부자료
- 시험패스 가능한 ACD301인기자격증 시험덤프 최신버전 덤프 □ 검색만 하면[www.itdumpskr.com]에서【 ACD301 】무료 다운로드ACD301인증덤프샘플 다운
- ACD301최신 업데이트 인증덤프 □ ACD301최신 시험대비자료 □ ACD301최신 시험대비자료 □ 무료 다운로드를 위해 지금□ www.exampassdump.com □에서“ ACD301 ”검색ACD301최신 인증시험 기출자료
- ACD301최신 시험대비자료 □ ACD301인증시험 인기덤프 □ ACD301인증시험 인기 덤프자료 □ ▶ www.itdumpskr.com◀웹사이트를 열고▷ ACD301 ◁를 검색하여 무료 다운로드ACD301최신 인증시험 기출자료
- ACD301인증시험 인기 덤프자료 □ ACD301최신시험 □ ACD301최신 인증시험 기출자료 㶞 ▶ kr.fast2test.com◀을(를) 열고➥ ACD301 □를 입력하고 무료 다운로드를 받으십시오ACD301높은 통과율 덤프자료
- ACD301인기자격증 시험덤프 최신 인기 인증시험 □ ➡ www.itdumpskr.com □□□을 통해 쉽게[ACD301]무료 다운로드 받기ACD301최고품질 예상문제모음
- ACD301인기자격증 시험덤프 인기시험 기출문제 □ ✔ www.passtip.net □✔□웹사이트를 열고《 ACD301 》를 검색하여 무료 다운로드ACD301시험대비 덤프 최신자료
- 완벽한 ACD301인기자격증 시험덤프 시험덤프공부 □ 무료로 쉽게 다운로드하려면⇛ www.itdumpskr.com⇚에서➽ ACD301 □를 검색하세요ACD301최고덤프자료
- ACD301인기자격증 시험덤프 시험준비에 가장 좋은 최신 공부자료 □ ☀ www.dumptop.com □☀□은{ ACD301 }무료 다운로드를 받을 수 있는 최고의 사이트입니다ACD301인증시험 인기 덤프자료
- www.stes.tyc.edu.tw, 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, www.stes.tyc.edu.tw, myportal.utt.edu.tt, bbs.t-firefly.com, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, myportal.utt.edu.tt, Disposable vapes

참고: Itexamdump에서 Google Drive로 공유하는 무료 2026 Appian ACD301 시험 문제집이 있습니다: https://drive.google.com/open?id=1mYEkkNiTatVbP7n4_0uL854II3q7Tqw8