Appian Associate Developer (ACD101) Practice Exam (Sample)

Study Guide

BY EXAMZIFY

Everything you need from our exam experts!

Notice: Examzify makes every reasonable effort to obtain accurate, complete, and timely information about this product from reliable sources.

Table of Contents

Introduction

Preparing for a certification exam can feel overwhelming, but with the right tools, it becomes an opportunity to build confidence, sharpen your skills, and move one step closer to your goals. At Examzify, we believe that effective exam preparation isn't just about memorization, it's about understanding the material, identifying knowledge gaps, and building the test-taking strategies that lead to success.

This guide was designed to help you do exactly that.

Whether you're preparing for a licensing exam, professional certification, or entry-level qualification, this book offers structured practice to reinforce key concepts. You'll find a wide range of multiple-choice questions, each followed by clear explanations to help you understand not just the right answer, but why it's correct.

The content in this guide is based on real-world exam objectives and aligned with the types of questions and topics commonly found on official tests. It's ideal for learners who want to:

- **Practice answering questions under realistic conditions,**
- **Improve accuracy and speed,**
- **Review explanations to strengthen weak areas, and**
- **Approach the exam with greater confidence.**

We recommend using this book not as a stand-alone study tool, but alongside other resources like flashcards, textbooks, or hands-on training. For best results, we recommend working through each question, reflecting on the explanation provided, and revisiting the topics that challenge you most.

Remember: successful test preparation isn't about getting every question right the first time, it's about learning from your mistakes and improving over time. Stay focused, trust the process, and know that every page you turn brings you closer to success.

Let's begin.

How to Use This Guide

This guide is designed to help you study more effectively and approach your exam with confidence. Whether you're reviewing for the first time or doing a final refresh, here’s how to get the most out of your Examzify study guide:

1. Start with a Diagnostic Review

Skim through the questions to get a sense of what you know and what you need to focus on. Your goal is to identify knowledge gaps early.

2. Study in Short, Focused Sessions

Break your study time into manageable blocks (e.g. 30 – 45 minutes). Review a handful of questions, reflect on the explanations.

3. Learn from the Explanations

After answering a question, always read the explanation, even if you got it right. It reinforces key points, corrects misunderstandings, and teaches subtle distinctions between similar answers.

4. Track Your Progress

Use bookmarks or notes (if reading digitally) to mark difficult questions. Revisit these regularly and track improvements over time.

5. Simulate the Real Exam

Once you're comfortable, try taking a full set of questions without pausing. Set a timer and simulate test-day conditions to build confidence and time management skills.

6. Repeat and Review

Don’t just study once, repetition builds retention. Re-attempt questions after a few days and revisit explanations to reinforce learning. Pair this guide with other Examzify tools like flashcards, and digital practice tests to strengthen your preparation across formats.

There’s no single right way to study, but consistent, thoughtful effort always wins. Use this guide flexibly, adapt the tips above to fit your pace and learning style. You've got this!

Questions

1. **Given ri!name is 'Maria' and ri!directory is a list of maps including Maria and setven, what is the expected output of the expression whereContains(ri!name, index(ri!directory, 'name'))?**
   - A. True
   - B. False
   - C. Null
   - D. Error

2. **Which statement is true about memory optimization in a process model?**
   - A. Splitting into sub processes reduces memory usage by allowing only part of the process to run at a time.
   - B. Increasing the number of process variables reduces memory usage.
   - C. Storing large data chunks in process variables improves performance.
   - D. Compressing text before storing in process variables is never beneficial.

3. **Which set of test scenarios should you configure for validating an email address?**
   - A. Invalid email without @, invalid email address: null, valid email address: jane.doe@example.com
   - B. An invalid email address that is missing the @ character; an invalid email address: null; a valid email address: jane.doe@example.com
   - C. Invalid email address with spaces; invalid email address: null; valid email address: jane.doe@example.com
   - D. Null email; valid email address: jane.doe@example.com; invalid email address missing @ character

4. **If you need to remove an unused field from an existing record type backed by a database, what is the recommended action?**
   - A. Delete the field from the record type and optionally delete the column from the database table
   - B. Deactivate the record type
   - C. Rename the field in the database
   - D. Archive the table

5. **Which Appian design object connects to a relational database, establishing a relationship between custom data types and database tables?**

   A. CDT

   B. Process Model

   C. Constant

   D. Data Store

6. **An expression rule deployed to production on January 12, 2022 has form type as a rule input and several other inputs describing the logged in user. Given values for these inputs, the rule returns a set of form instructions for the specific form as applicable to the logged in user. What is the most important action to take with regards to the object description?**

   A. Set an object description that reads: "returns a set of form instructions to display, given the type of form and characteristics of the logged in user"

   B. Update the deployment date in the description

   C. Describe the data model

   D. Add unit tests

7. **Which code snippet retrieves the first 10 entries from the abc_book table as CDT objects and returns the total count?**

   A. a!queryEntity(

   entity: cons!ABC_DSE_BOOK

   query: a!query(

   pagingInfo: a!pagingInfo(

   startIndex: 1,

   batchSize: 10,

)

)

   fetchTotalCount: true

)

   B. a!queryEntity(entity: cons!ABC_DSE_BOOK, query: a!query(pagingInfo: a!pagingInfo(startIndex: 0, batchSize: 10)), fetchTotalCount: true)

   C. a!queryEntity(entity: cons!ABC_DSE_BOOK, query: a!query(pagingInfo: a!pagingInfo(startIndex: 1, batchSize: 10)), fetchTotalCount: false)

   D. a!queryEntity(entity: cons!ABC_DSE_BOOK, query: a!query(pagingInfo: a!pagingInfo(startIndex: 2, batchSize: 10)), fetchTotalCount: true)

8. **In Appian expressions, what does ri! denote?**

   A. Process variable

   B. Rule input

   C. Global constant

   D. Local variable

9. **What does len(ri!userComment) > 10 evaluate?**

   A. The length of the string is greater than 10

   B. The length equals 10

   C. The length is less than 10

   D. The string is empty

10. A user needs to navigate from a record summary to an external URL. Which interface component supports this?

- **A. Card Layout with a link**
- **B. List View with a URL field**
- **C. Chart component with an external URL**
- **D. Form element linking to an external page**

Answers

1. **A**
2. **B**
3. **B**
4. **A**
5. **D**
6. **A**
7. **A**
8. **B**
9. **A**
10. **A**

Explanations

1. **Given ri!name is 'Maria' and ri!directory is a list of maps including Maria and setven, what is the expected output of the expression whereContains(ri!name, index(ri!directory, 'name'))?**

   A. True

   B. False

   C. Null

   D. Error

You’re testing how to check if a specific name appears in the set of names pulled from a list of records. ri!directory is a list of maps where each map includes a name field, such as Maria and Setven. The expression index(ri!directory, 'name') collects the values of the name field from every map in the list, giving something like ['Maria', 'Setven']. Then whereContains(ri!name, index(ri!directory, 'name')) checks whether the left-side string, Maria, is contained in that collection of names. Since Maria is indeed one of the names in the directory, the expression evaluates to True.

2. **Which statement is true about memory optimization in a process model?**

   A. Splitting into sub processes reduces memory usage by allowing only part of the process to run at a time.

   B. Increasing the number of process variables reduces memory usage.

   C. Storing large data chunks in process variables improves performance.

   D. Compressing text before storing in process variables is never beneficial.

Memory usage in a process model is tied to how much data you keep active in memory at once. If you store large data items directly in process variables, the in-memory footprint grows quickly. Increasing the number of process variables can help reduce peak memory usage because you can split data into smaller pieces, keep only what you need in memory, and store the bulk of data externally while holding lightweight references or smaller chunks in variables. This on-demand or chunked approach minimizes the amount of data resident in memory at any moment. Other options tend to be less reliable. Splitting into sub-processes can help with organization and control flow, but memory reduction isn’t guaranteed and depends on the overall design. Storing large data chunks in process variables generally increases memory usage rather than reduces it. Compressing text before storing can be beneficial in many cases, especially for large text, so the claim that it’s never beneficial isn’t accurate.

3. Which set of test scenarios should you configure for validating an email address?

A. Invalid email without @, invalid email address: null, valid email address: jane.doe@example.com

B. An invalid email address that is missing the @ character; an invalid email address: null; a valid email address: jane.doe@example.com

C. Invalid email address with spaces; invalid email address: null; valid email address: jane.doe@example.com

D. Null email; valid email address: jane.doe@example.com; invalid email address missing @ character

When validating an email address, test scenarios should cover common invalid formats, the possibility of no input, and a valid example. Including an invalid email that is missing the @ symbol ensures the validator rejects inputs that lack essential structure. Including a null input checks that the function handles absence of data gracefully without errors. Including a known valid email like jane.doe@example.com confirms the validator accepts correct formats and returns a positive result. Other sets might miss one of these critical paths, such as not testing a null input or not including a clearly invalid format, which leaves gaps in how well the validation logic handles real-world cases.

4. If you need to remove an unused field from an existing record type backed by a database, what is the recommended action?

A. Delete the field from the record type and optionally delete the column from the database table

B. Deactivate the record type

C. Rename the field in the database

D. Archive the table

When a field in a record type that is backed by a database is no longer needed, clean up both the model and the underlying table to keep things in sync. The recommended action is to delete the field from the record type, and if the column in the database is not used anywhere else, you can drop that column as well. This prevents orphaned fields or columns that could cause confusion, errors, or maintenance headaches later. Before dropping the database column, make sure no reports, integrations, or other parts of the system rely on it, and consider any data retention requirements. Deactivating the record type stops it from being used but leaves the database and field intact. Renaming the field wouldn’t remove it, and archiving the table is overly drastic and would disrupt existing data and relationships.

5. Which Appian design object connects to a relational database, establishing a relationship between custom data types and database tables?

   A. CDT

   B. Process Model

   C. Constant

   D. Data Store

In Appian, the component that connects to a relational database and links your custom data types to actual database tables is the Data Store. A Data Store defines how Appian talks to the database and provides the mapping between a CDT (custom data type) and the corresponding database tables through Data Store Entities. This mapping tells Appian how to persist, retrieve, and update data using the CDT’s structure in relation to the physical table columns. The CDT defines the in-app data structure, but it’s the Data Store that establishes the database connection and the relationship to the tables. Process Models handle workflow logic, and Constants are fixed values, so they don’t establish database connections or mappings.

6. An expression rule deployed to production on January 12, 2022 has form type as a rule input and several other inputs describing the logged in user. Given values for these inputs, the rule returns a set of form instructions for the specific form as applicable to the logged in user. What is the most important action to take with regards to the object description?

   A. Set an object description that reads: "returns a set of form instructions to display, given the type of form and characteristics of the logged in user"

   B. Update the deployment date in the description

   C. Describe the data model

   D. Add unit tests

Documenting what the rule does is the key purpose of its object description. The rule takes the form type as a rule input along with user-related inputs and returns a set of form instructions tailored to that form and the logged-in user. The best description states this behavior clearly, so future developers understand exactly what the rule produces and under what conditions. That’s why phrasing the description to reflect the output—“returns a set of form instructions to display, given the type of form and characteristics of the logged in user”—is the most precise and helpful option.   Other choices don’t serve the same purpose: updating the deployment date doesn’t convey what the rule does, describing the data model focuses on structure rather than behavior, and while unit tests are important, they belong in testing artifacts, not the object description itself.

7. Which code snippet retrieves the first 10 entries from the abc_book table as CDT objects and returns the total count?

A. a!queryEntity(

entity: cons!ABC_DSE_BOOK

query: a!query(

pagingInfo: a!pagingInfo(

startIndex: 1,

batchSize: 10,

)

)

fetchTotalCount: true

)

B. a!queryEntity(entity: cons!ABC_DSE_BOOK, query: a!query(pagingInfo: a!pagingInfo(startIndex: 0, batchSize: 10)), fetchTotalCount: true)

C. a!queryEntity(entity: cons!ABC_DSE_BOOK, query: a!query(pagingInfo: a!pagingInfo(startIndex: 1, batchSize: 10)), fetchTotalCount: false)

D. a!queryEntity(entity: cons!ABC_DSE_BOOK, query: a!query(pagingInfo: a!pagingInfo(startIndex: 2, batchSize: 10)), fetchTotalCount: true)

To get the first 10 entries as CDT objects and also know how many total records exist, you page the query from the first record and request the total count. In Appian, the first page starts at startIndex 1 with a batchSize of 10, and you enable total counting with fetchTotalCount: true. This combination returns the first 10 records from the abc_book CDT as objects and includes the overall total number of matching records. That's why the snippet using the entity cons!ABC_DSE_BOOK, pagingInfo with startIndex: 1 and batchSize: 10, and fetchTotalCount: true is the correct choice. Using a startIndex of 0 would not align with Appian's 1-based paging for the first page, startIndex: 2 would skip the first page, and fetching the total count as false would omit the total number of records.

8. In Appian expressions, what does ri! denote?

A. Process variable

B. Rule input

C. Global constant

D. Local variable

ri! denotes Rule Inputs. In Appian, a rule is defined with named inputs, and the values passed to those inputs when the rule runs are read inside the rule using ri! followed by the input's name. This makes the rule parameterized and reusable. For example, if a rule accepts an input called amount, you would access its value with ri!amount inside the rule. This is different from process variables (pv!) which reside in a running process, or from global constants and local variables used in other parts of the application.

9. What does len(ri!userComment) > 10 evaluate?

A. The length of the string is greater than 10

B. The length equals 10

C. The length is less than 10

D. The string is empty

The expression is counting characters and then checking that count against a threshold. len(ri!userComment) returns how many characters are in the string provided by the rule input userComment. The comparison > 10 asks if that count is greater than 10, so the result is a boolean: true when the comment has 11 or more characters, false when it has 10 or fewer. This is a common way to enforce minimum length on user input. If the input could be missing or null, you'd want to guard against that to avoid errors.

10. A user needs to navigate from a record summary to an external URL. Which interface component supports this?

A. Card Layout with a link

B. List View with a URL field

C. Chart component with an external URL

D. Form element linking to an external page

Embedding a navigable link in a record summary is best achieved with a Card Layout that includes a hyperlink. Card Layouts are designed to present summary content in a compact, clickable form, so adding a link there lets users jump directly to an external URL from the record view. A URL field in a List View would show the URL value, but isn't inherently a single-click navigation action from the summary. A Chart component focuses on data visualization, and a Form element is meant for data capture, not navigation. So the Card Layout with a link best supports navigating from the record summary to an external site.

Next Steps

Congratulations on reaching the final section of this guide. You've taken a meaningful step toward passing your certification exam and advancing your career.

As you continue preparing, remember that consistent practice, review, and self-reflection are key to success. Make time to revisit difficult topics, simulate exam conditions, and track your progress along the way.

If you need help, have suggestions, or want to share feedback, we’d love to hear from you. Reach out to our team at hello@examzify.com.

Or visit your dedicated course page for more study tools and resources:

https://appianacd101.examzify.com

We wish you the very best on your exam journey. You've got this!