

Salesforce Platform Developer Practice Exam (Sample)

Study Guide



Everything you need from our exam experts!

This is a sample study guide. To access the full version with hundreds of questions,

Copyright © 2026 by Examzify - A Kaluba Technologies Inc. product.

ALL RIGHTS RESERVED.

No part of this book may be reproduced or transferred in any form or by any means, graphic, electronic, or mechanical, including photocopying, recording, web distribution, taping, or by any information storage retrieval system, without the written permission of the author.

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

SAMPLE

Table of Contents

Copyright	1
Table of Contents	2
Introduction	3
How to Use This Guide	4
Questions	6
Answers	10
Explanations	12
Next Steps	19

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. Don't worry about getting everything right, 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, and take breaks to retain information better.

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.

7. Use Other Tools

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!

SAMPLE

Questions

- 1. Which data structure is returned after performing a SOSL search?**
 - A. A list of lists of sObjects.**
 - B. A map of sObject types to a list of sObjects.**
 - C. A map of sObject types to a list of lists of sObjects.**
 - D. A list of sObjects.**
- 2. What is the maximum number of custom fields that can be created on a Salesforce object?**
 - A. 100**
 - B. 200**
 - C. 500**
 - D. 800**
- 3. If the assignee of a workflow task is a role containing more than one assigned user, who is designated as the assignee?**
 - A. Account Owner**
 - B. System Administrator**
 - C. The person who triggered the rule**
 - D. The record will show an error message**
- 4. Which of the following statements is true regarding an auto number field?**
 - A. The data remains unchanged if changed to a text field**
 - B. All records created as auto-number will be deleted**
 - C. Data contained in a text field will be deleted when changed to auto number**
 - D. None of the above**
- 5. What action can a developer take in a before update trigger?**
 - A. Update the original object using an update DML operation.**
 - B. Delete the original object using a delete DML operation.**
 - C. Change field values using the Trigger.new context variable.**
 - D. Display a custom error message in the application interface.**

- 6. What are some ways to keep email templates organized? Select all that apply.**
- A. Preface any template name with the name of the app**
 - B. Create a public email template folder with the name of the app**
 - C. None of the above**
 - D. Use consistent naming conventions across templates**
- 7. To track engines and their parts without orphan records, the developer should?**
- A. Create a champion object to relate many engines to many parts.**
 - B. Create a master-detail relationship for engines and parts.**
 - C. Create a lookup relationship for each part to the engine object.**
 - D. Create a junction object for many-to-many relationships.**
- 8. Which code block correctly returns the ListView of an Account object using a debug statement?**
- A. `ApexPages.StandardSetController controller = new ApexPages.StandardSetController([SELECT Id FROM Account LIMIT 1]);`**
 - B. `ApexPages.StandardController controller = new ApexPages.StandardController(Database.getQueryLocator('select Id from Account Limit 1'));`**
 - C. `ApexPages.StandardController controller = new ApexPages.StandardController([SELECT Id FROM Account LIMIT 1]);`**
 - D. `ApexPages.StandardSetController controller = new ApexPages.StandardSetController(Database.getQueryLocator('select Id from Account Limit 1'));`**
- 9. What is the primary purpose of dynamic dashboards in Salesforce?**
- A. To display information statically**
 - B. To allow each user to see data based on their permissions**
 - C. To combine various dashboard types**
 - D. To enhance report access**

10. Which statement ensures that a Salesforce application can scale effectively for multiple users?

- A. Implement custom user sessions.**
- B. Utilize sharing settings effectively.**
- C. Optimize SOQL queries to retrieve maximum records.**
- D. Design for bulk operations within Apex.**

Answers

SAMPLE

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

SAMPLE

Explanations

1. Which data structure is returned after performing a SOSL search?

A. A list of lists of sObjects.

B. A map of sObject types to a list of sObjects.

C. A map of sObject types to a list of lists of sObjects.

D. A list of sObjects.

When performing a SOSL (Salesforce Object Search Language) search, the correct data structure returned is a map of sObject types to a list of lists of sObjects. This means that the SOSL search results will categorize the returned records based on the type of sObject, and each type will contain a list of lists, where each inner list represents the sObjects that match the search criteria. In SOSL, you can search across multiple objects and fields simultaneously, and the system organizes the results to ensure that you can easily identify which records belong to which object type. For instance, if you searched for a keyword in both Contacts and Accounts, the result would give you a separate list of records for each object. Choosing a list of lists of sObjects only captures the records without the context of the object types they belong to, which is not sufficient for most use cases where clear categorization is needed. Using a simple list of sObjects would miss the necessary organization by object type, while a map of just object types to lists misses the granularity of lists under each type. Therefore, the structure that incorporates these aspects fulfills the requirements of the SOSL search effectively.

2. What is the maximum number of custom fields that can be created on a Salesforce object?

A. 100

B. 200

C. 500

D. 800

The limit of custom fields that can be created on a Salesforce object is determined by the object type and the Salesforce edition being used. For standard objects, the maximum number of custom fields that can be created is typically around 500 fields. This limit is applicable to most standard Salesforce objects, making it significant for organizations that need to customize their Salesforce instance extensively to meet unique business requirements. Custom fields allow organizations to extend the functionality of standard objects and tailor data storage to fit specific needs. The ability to create up to 500 custom fields per object enables developers and administrators to accommodate a wide range of business processes, making Salesforce a flexible platform for managing data. In contrast, other choices would suggest much lower limits, which do not reflect the current capabilities Salesforce provides for custom field creation on standard objects. Understanding the limits for custom fields is crucial for effective object management and ensuring that data architecture remains efficient and scalable in the Salesforce environment.

3. If the assignee of a workflow task is a role containing more than one assigned user, who is designated as the assignee?

A. Account Owner

B. System Administrator

C. The person who triggered the rule

D. The record will show an error message

When a workflow task is assigned to a role that includes multiple users, the system needs to determine a specific individual to designate as the assignee for that task. In this situation, the person who triggered the workflow rule will be designated as the assignee. This approach helps ensure that the task is effectively connected to a relevant user who initiated the action that led to the workflow being enacted. The reasoning behind this is rooted in the design of Salesforce's workflow management. By assigning the task to the person who triggered the rule, it maintains a clear connection between the action taken and the action required (the task). This enhances accountability and ensures that the individual who initiated the process is directly responsible for following up on the task at hand. In scenarios where roles are involved, but no specific user is clearly defined as the assignee, the system defaults to the individual who triggered the workflow to prevent ambiguity and ensure that workflow actions remain traceable and manageable. This choice maximizes the effectiveness of workflow automation within the Salesforce environment.

4. Which of the following statements is true regarding an auto number field?

A. The data remains unchanged if changed to a text field

B. All records created as auto-number will be deleted

C. Data contained in a text field will be deleted when changed to auto number

D. None of the above

An auto number field in Salesforce is designed to automatically generate a unique number for each record, based on a specified format. When an auto number field is changed to a text field, the data in that field remains unchanged. This means that the previously generated numbers will stay the same, but the field type will allow for more flexibility in data entry and manipulation. In contrast, when a field type is changed, the data integrity must be maintained. In this case, changing from an auto number to a text field does not delete or alter the existing data; it simply changes how the information can be interacted with going forward. For instance, records that were previously assigned an auto number will retain their values when viewed in the text format, which enhances usability. Thus, the reason this choice is correct lies in the fact that the content of the records created remains intact, and users have the potential to further utilize this data. The other responses address scenarios that are either incorrect or do not accurately represent how Salesforce handles these field type changes. Transitioning from auto number to text does not lead to the deletion of records or their data; therefore, the selected statement is indeed the only one that stands true in this context.

5. What action can a developer take in a before update trigger?
- A. Update the original object using an update DML operation.
 - B. Delete the original object using a delete DML operation.
 - C. Change field values using the Trigger.new context variable.**
 - D. Display a custom error message in the application interface.

In a before update trigger, a developer is allowed to manipulate the records that are being updated before they are saved to the database. This is achieved using the Trigger.new context variable, which contains the new version of the records being updated. By modifying the field values within Trigger.new, the developer can ensure that specific fields are set to desired values before the update transaction is completed. This capability is essential for enforcing business rules, performing validations, or modifying data as part of the processing logic. For example, if a certain field must be adjusted based on conditions within the logic, the developer can simply modify the corresponding values in Trigger.new. The other options involve operations that are not permissible in a before update trigger context. Updating or deleting the original object through DML operations in a before update trigger would result in recursive calls and is not allowed, as it can lead to unintended consequences and performance issues. Displaying a custom error message is also not feasible directly within a trigger context; rather, error messages can be managed through exceptions thrown during validation, but this requires a different approach than simply displaying them through the trigger logic.

6. What are some ways to keep email templates organized? Select all that apply.
- A. Preface any template name with the name of the app**
 - B. Create a public email template folder with the name of the app
 - C. None of the above
 - D. Use consistent naming conventions across templates

Keeping email templates organized is essential for efficient use and quick retrieval within Salesforce. Prefacing a template name with the name of the app helps identify its purpose and context immediately. This approach allows users to recognize which templates belong to which applications, thus minimizing confusion, especially in environments where multiple apps and templates are used. Creating a public email template folder named after the app serves a similar purpose. This organization method allows for clearer categorization, making it easier for all users to find related templates in one location. It enhances collaboration as team members can easily access templates that are relevant to their work. Using consistent naming conventions across templates is another effective strategy. By establishing a standard format for naming, users can quickly identify the template's content, purpose, and target audience without ambiguity. Consistency aids in maintaining clarity and reduces the likelihood of miscommunication among team members. Together, these strategies promote an organized approach to managing email templates, making it easier for users to locate and utilize the templates they need efficiently.

7. To track engines and their parts without orphan records, the developer should?

- A. Create a champion object to relate many engines to many parts.**
- B. Create a master-detail relationship for engines and parts.**
- C. Create a lookup relationship for each part to the engine object.**
- D. Create a junction object for many-to-many relationships.**

Creating a junction object for many-to-many relationships is the correct approach for tracking engines and their parts without orphan records. A junction object serves as an intermediary that associates records from two different objects, which in this case would be engines and parts. By implementing a junction object, you can effectively create a relationship where multiple engines can be associated with multiple parts, and vice versa. This method ensures that each part is tied directly to its respective engine and that the relationship is stable, preventing orphan records, which could occur if parts are not linked to any engines. This setup also allows for ease of queries and maintenance, as you can manage the associations through the junction object. Using a master-detail relationship for engines and parts would imply a parent-child relationship, which isn't suitable for many-to-many scenarios. Only one part could be linked to one engine in that case, leading to potential loss of information and flexibility. Similarly, creating a lookup relationship for each part to the engine object assumes a one-to-many relationship, which also wouldn't capture the necessary complexity of multiple engines sharing parts without duplicating records. An alternative option like creating a champion object might not fully resolve the need for an explicit many-to-many relationship. Overall, a junction object is the most efficient

8. Which code block correctly returns the ListView of an Account object using a debug statement?
- A. `ApexPages.StandardSetController controller = new ApexPages.StandardSetController([SELECT Id FROM Account LIMIT 1]);`
 - B. `ApexPages.StandardController controller = new ApexPages.StandardController(Database.getQueryLocator('select Id from Account Limit 1'));`
 - C. `ApexPages.StandardController controller = new ApexPages.StandardController([SELECT Id FROM Account LIMIT 1]);`
 - D. `ApexPages.StandardSetController controller = new ApexPages.StandardSetController(Database.getQueryLocator('select Id from Account Limit 1'));`**

The selected answer utilizes the `ApexPages.StandardSetController` with a `Database.getQueryLocator` method, which is the appropriate way to handle large sets of records in Salesforce Visualforce pages. The `StandardSetController` is designed to work with large data sets efficiently. By utilizing `Database.getQueryLocator`, it allows for pagination and better memory management, which is vital when dealing with scenarios where the number of records can be substantial. The use of the query string retrieves a set of records (in this case, Account records) that can be displayed in a ListView format. Additionally, this approach enables developers to implement features like pagination in the user interface without loading all records at once, which enhances performance. This usage is essential for adhering to limits in Salesforce, particularly with governor limits around heap size and query results. In contrast, other choices either misuse `ApexPages.StandardController`, which is meant for single records, or employ direct queries without the pagination benefits provided by `StandardSetController`, therefore not facilitating the management of potentially large result sets optimally. Such configurations would not yield the intended ListView behavior for larger datasets effectively.

9. What is the primary purpose of dynamic dashboards in Salesforce?

- A. To display information statically**
- B. To allow each user to see data based on their permissions**
- C. To combine various dashboard types**
- D. To enhance report access**

The primary purpose of dynamic dashboards in Salesforce is to allow each user to see data based on their permissions. Dynamic dashboards enable personalization by changing the underlying data displayed according to the criteria set by the current user viewing the dashboard. This means that each user can see different information that is relevant to them, depending on their role, profile, or specific permissions set within Salesforce. This capability is crucial for organizations where users have varying levels of access to data, as it ensures that sensitive information is only visible to those who are authorized to see it. By tailoring the data being shown, dynamic dashboards not only improve data security but also enhance the relevance of the information each user receives, thereby driving better decision-making and engagement. In contrast, static dashboards present the same data to everyone, regardless of their permission levels, which limits the utility of the information displayed. Combining various dashboard types doesn't speak directly to user-specific access and is more about the configuration of the presentation. Enhancing report access relates more to how reports are designed and made available rather than the personalized visibility offered through dynamic dashboards.

10. Which statement ensures that a Salesforce application can scale effectively for multiple users?

- A. Implement custom user sessions.**
- B. Utilize sharing settings effectively.**
- C. Optimize SOQL queries to retrieve maximum records.**
- D. Design for bulk operations within Apex.**

Designing for bulk operations within Apex is crucial for scaling a Salesforce application effectively for multiple users. When an application is designed with bulk processing in mind, it can handle large volumes of data and multiple transactions simultaneously without degrading performance. This is particularly important in a multi-tenant environment like Salesforce where resources are shared among numerous users. Apex provides specific features that enable developers to process records in bulk, such as trigger patterns that operate on collections of records instead of executing one record at a time. By using bulk processing techniques, developers can minimize resource consumption, adhere to governor limits, and provide a more responsive and efficient experience for users, regardless of the number of users accessing the application at any given time. This approach allows the application to remain performant as the number of users and transactions increases, which is essential for maintaining a smooth experience in a scalable way.

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://salesforce-platformdeveloper.examzify.com>

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