Salesforce
Plat-Arch-205
Salesforce Certified Platform Sharing and Visibility Architect

For More Information – Visit link below:
https://www.examsempire.com/

Product Version

1. Up to Date products, reliable and verified.
2. Questions and Answers in PDF Format.



Latest Version: 27.1

Question: 1

Universal Containers (UC) is looking to expand its delivery capabilities through a network of distributors that use a Partner Community license. UC employees currently can view all delivery records through the organization-wide default (OWD) setting of Public Read-Only. Which approach should an architect recommend to limit the records a distributor can see?

A. Create an ownership-based sharing rule to grant access to the distributor.
B. Create a criteria-based sharing rule to grant access to the distributor.
C. Set the External OWD to Private for the Delivery object.

Answer: C

Explanation:
Comprehensive and Detailed 150 to 250 words of Explanation From Platform Sharing and Visibility Architect/Course Guide/topics:
External organization-wide defaults are designed for this exact separation of internal and external access. UC can keep the internal default for Delivery at the level required by employees while setting Default External Access to Private for authenticated external users. That creates a restrictive baseline for distributors without reducing internal visibility. Sharing rules are additive: they can open access beyond the organization-wide default, but they cannot take away access already granted by a public OWD. For that reason, neither an owner-based nor a criteria-based rule can be used to make external access more restrictive. After the external OWD is Private, UC can selectively grant the required records through the appropriate Experience Cloud mechanism, such as sharing sets, partner roles, sharing rules, or other supported sharing constructs depending on the relationship model. This follows the least-privilege design principle used throughout Salesforce security architecture: establish the most restrictive baseline that satisfies the broad population, then add narrowly scoped exceptions. Study Guide Access to Records - organization-wide defaults, external OWDs, Experience Cloud record access, sharing rules, and least-privilege sharing.
==========

Question: 2

Sales operations at Universal Containers (UC) has created Public Reports and Dashboards folders for sales managers. Sales operations and sales managers report to the VP of Sales. Sales operations currently spends a few hours each month updating users that should have access to edit reports and dashboards in these folders. How should UC grant access to sales managers to automate access to these Public Reports and Dashboards folders?

A. Share the folders with the lowest roles in the Role Hierarchy; superiors will get access automatically.
B. Share the folders with the Sales Managers profile.
C. Share the folders with a "Sales Managers" public Group.

Answer: C

Explanation:
Comprehensive and Detailed 150 to 250 words of Explanation From Platform Sharing and Visibility Architect/Course Guide/topics:
A public group provides the cleanest reusable security boundary for report and dashboard folder access. UC can create a Sales Managers public group and maintain the appropriate managers as members, then grant that group the required folder access. Folder sharing remains attached to the group instead of being maintained user by user, which removes the recurring monthly administration described in the scenario. Enhanced folder sharing supports audiences such as users, roles, territories, and public groups, with the folder access level determining whether members can view, edit, or manage the content. Profiles are not the normal sharing target for public report and dashboard folders, so changing a profile does not solve the folder-membership problem. The role hierarchy also should not be treated as though folder access automatically rolls upward in the same way as record access. Folder security is a separate authorization layer. Public groups are particularly effective when a functional population must receive identical access even though membership changes over time. Study Guide Access to Other Data - report and dashboard folders, enhanced folder sharing, public groups, analytics permissions, and scalable administration.
==========

Question: 3

Universal Containers (UC) has a custom Apex class that enforces a business process and updates opportunities. UC has noticed that fields with Field-Level Security permission of Read- Only on certain users' profiles are being updated by this class. How should the architect fix this problem?

A. Add the With Sharing keyword to the class.
B. Put the code in an inner class that uses the With Sharing keyword.
C. Use the isUpdateable() Apex method to test each field prior to allowing update.

Answer: C

Explanation:
Comprehensive and Detailed 150 to 250 words of Explanation From Platform Sharing and Visibility Architect/Course Guide/topics:
The security defect is field-level write authorization, not record-level sharing. Apex can execute with elevated behavior, so code that performs DML must explicitly respect whether the running

user is allowed to update the fields being changed. The Schema Describe check represented by isUpdateable() verifies field update permission before DML and directly addresses the Read-Only Field-Level Security condition in the scenario. By contrast, the with sharing keyword controls record visibility through the sharing model; it does not itself enforce object CRUD or Field-Level Security. Moving the same logic into another with sharing class therefore would not stop a field from being updated when the user lacks field edit permission. In modern Salesforce development, user-mode database operations and Security.stripInaccessible() can provide broader and often simpler enforcement of CRUD and FLS, but the explicit isUpdateable() check remains technically valid for the option set given. The architectural rule is to evaluate object permissions, field permissions, and record sharing as separate layers. Study Guide Permissions to Standard Objects, Custom Objects, and Fields - CRUD, Field-Level Security, secure Apex, Schema Describe checks, and user-mode operations.
==========

Question: 4

Users at Universal Containers are complaining that a field has disappeared from the Account page after last weekend's deployment. The page layout did not change with this deployment. How should the admin troubleshoot this issue?

A. Run a Who Sees What report, filtering on Account.
B. Log in as a user and check several accounts to isolate the problem records.
C. View Field Accessibility in the Object Manager.

Answer: C

Explanation:
Comprehensive and Detailed 150 to 250 words of Explanation From Platform Sharing and Visibility Architect/Course Guide/topics:
Because the Account page layout did not change, the next control to inspect is Field-Level Security and effective field accessibility. Salesforce can hide a field from a user even when that field remains present on the assigned page layout. The Field Accessibility view in Object Manager provides a consolidated way to determine whether a field is visible, read-only, or hidden for the relevant profile and configuration. This is a better diagnostic than logging in and checking several records because the symptom is field-level, not record-specific. A Who Sees What analysis focuses on record visibility and therefore does not explain why one field disappeared while the user can still open the Account. The administrator should verify the affected field's FLS for the user's profile and any relevant permission-set assignments, then confirm that the page layout and record type do not impose additional presentation restrictions. Field-Level Security is the enforceable security boundary; page layout placement alone is not. Study Guide Permissions to Standard Objects, Custom Objects, and Fields - Field-Level Security, field accessibility, profiles, permission sets, and page layouts.
==========

Question: 5

A consulting company uses the Salesforce mobile app for its field consultants and uses Case object to track customer specific consulting done by field consultants. The company also has a large number of customer service representatives who takes calls from customers on company issued desktops and uses case object to track customer issues and grievances. The company would like to capture images of customer site captured by field consultants while they are editing the case record during customer site visit. The Director of IT wants to minimize customization and promote reusability of code artifacts wherever possible. What recommendations should an architect give to the company to implement the image capture requirement, while ensuring customer that the service rep can continue to use same lightning pages they were trained to use?

A. Use Lightning Component as an override for "Edit" action on mobile view allowing image capture feature. No Change required for desktop users.
B. Create a separate button "Edit in Mobile", which opens a custom lightning component that will allow field consultants to add an image. No change required for desktop users.
C. Use Lightning Component as an override for "Edit" action on lightning experience allowing image capture feature. Detect the form factor of the device and redirect the user to the default not-overridden view.

Answer: A

Explanation:
Comprehensive and Detailed 150 to 250 words of Explanation From Platform Sharing and Visibility Architect/Course Guide/topics:
Salesforce supports experience-specific standard-action overrides, so the Case Edit action can be customized for the Salesforce mobile app while the desktop Lightning Experience continues to use the standard page. That directly satisfies the requirement to add image capture for field consultants without retraining the large desktop service population or replacing their existing Lightning pages. A mobile-specific Lightning component override keeps the customization localized to the form factor that needs it and maximizes reuse of standard platform behavior elsewhere. Creating a separate Edit in Mobile button would introduce a parallel workflow that users must remember and maintain. Overriding the desktop Lightning Experience and then detecting form factor in code would be more complex still: it would add routing logic, testing surface, and code merely to reproduce the existing desktop behavior. The architectural preference is to constrain customization to the user experience that actually needs it and avoid custom code when standard Salesforce configuration can isolate the behavior cleanly. Study Guide Implications of Security Model Choice - Lightning action overrides, mobile versus desktop behavior, reusable components, maintainability, and declarative-first design.
==========

Thank You for Trying Our Product

Special 16 USD Discount Coupon: NSZUBG3X

Email: support@examsempire.com

Check our Customer Testimonials and ratings available on every product page.

Visit our website.

https://examsempire.com/