DUMPSBOSS.

Salesforce Certified Sharing and Visibility Architect

Salesforce Sharing-and-Visibility-Architect

Version Demo

Total Demo Questions: 35

Total Premium Questions: 357

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, Sharing and Visibility	295
Topic 2, Performance	21
Topic 3, User Management	30
Topic 4, Mix Questions	11
Total	357

QUESTION NO: 1

Universal container (UC) use External Object to retrieve Invoice data from a Legacy ERP. A finance team requested to have access to the Invoice records in the account page.

In addition to objects access in the finance users profile, what other feature should a Sales Architect recommend?

A. Create a criteria-based sharing rule to grant access to the records.

B. Include the Invoice Related List On Account Page layout.

C. Create an owner-based sharing rule to grant access to the records.

D. Use APEX managed sharing to grant access to the records.

ANSWER: B

Explanation:

Include the Invoice Related List On Account Page layout. is the appropriate recommendation because the requirement is to make externally stored Invoice records visible in the context of an Account record. When an external object is related to Account through a supported external-object relationship, Salesforce can expose the related Invoice records as a related list on the Account page layout. Finance users who have the required object permissions for the external object can then navigate to an Account and view the associated invoice data without copying that data into Salesforce.

This approach fits Salesforce Connect's purpose: Salesforce displays data that remains in the external ERP in real time or on demand. Adding the related list is a user-interface configuration that places the relationship where the finance team needs it, on the Account record page. The relationship must be configured between the Invoice external object and Account, and the external data source must be available to the users. Salesforce documents external-object relationships and their use in displaying related external data at External Object Relationships. Additional Salesforce Connect implementation information is available at Salesforce Connect Developer Guide.

QUESTION NO: 2

Susan posts a file to the chatter fees for a record of an object which OWD is private. Which

two statements accurately describe who can view the file by default?

Choose 2 answers.

A. Susan and users with the View All Data permission.

B. Susan and users with access to the record.

C. Susan and users with a shared chatter post link to the file.

D. Susan only.

ANSWER: A B

Explanation:

When Susan posts a Salesforce File to the Chatter feed for a record, the file is shared in the context of that record. Consequently, Susan and users who have access to the record can view the file by default. Record access can come from ownership, role hierarchy access where applicable, sharing rules, manual sharing, teams, or any other mechanism that grants the user access to the private record. The object's private organization-wide default limits baseline record visibility, but it does not prevent users who are explicitly granted record access from viewing files attached or posted to that record's feed.

Users with the *View All Data* permission can view all records regardless of sharing settings and can also view all Salesforce Files. This broad administrative permission therefore includes visibility to the file that Susan posted, even if the user would not otherwise receive access through the record's normal sharing model. Salesforce documents that files posted to a record are available to people with access to that record, and that *View All Data* provides access to all files. See Salesforce Files sharing documentation and Salesforce system permissions documentation.

QUESTION NO: 3

To grant Universal Containers sales managers access to shipment records properly it was necessarily to the IT Team is worried about improper access to records.

Which two features and best practices should a Salesforce architect recommended to mitigate the risk?

A. USe isShareable keyword in Apex classes to assure record visibility will be followed

B. Use runAs system method in test classes to test using different users and profiles.

C. Use With Sharing keyword in Apex classes to assure record visibility will be followed

D. User isAccessable keyword Apex classes to assure recor visibility will be followed.

ANSWER: B C

Explanation:

Use With Sharing keyword in Apex classes to assure record visibility will be followed is appropriate because Apex ordinarily runs in system mode and can bypass record-sharing rules. Declaring a class `with sharing` causes Salesforce to apply the current user's record-level sharing when the class accesses records, helping ensure sales managers can retrieve only shipment records to which they have been granted access through the configured sharing model. This should be a deliberate part of the Apex design whenever code performs record queries or updates on behalf of users.

Use runAs system method in test classes to test using different users and profiles. is also a sound practice. `System.runAs` lets tests execute as a specified user, so the team can validate the intended access behavior for users with different roles, profiles, and sharing circumstances. Tests should create representative users and records, then confirm that the sharing design and Apex code produce the expected visibility outcomes. Together, sharing-aware Apex and tests that exercise realistic user contexts reduce the chance that code unintentionally exposes shipment records. See Salesforce's Apex sharing documentation and System.runAs documentation.

QUESTION NO: 4

A junior account manager owns an account and creates a new opportunity to manage a complex deal. She needs the help of the product specialist and solution engineer. Given the size of this deal, she knows the account is likely to be reassigned to a senior account manager in the near future.

What is the optimal way for the junior account manager to share the opportunity, given the private sharing model?

A. Manual share on the opportunity

B. Manual share on the account

C. Opportunity Team

ANSWER: C

Explanation:

Opportunity Team is the optimal approach because it is designed for focused collaboration on a particular opportunity. The junior account manager can add the product specialist and solution engineer as team members and assign each person the appropriate opportunity access level, such as read-only or read/write. This gives the deal team access to the specific opportunity while the organization-wide default for opportunities remains Private, preserving the intended record-security model.

Opportunity teams also fit the anticipated ownership transition. Team membership is associated with the opportunity itself rather than relying on a one-time owner-created sharing grant. When responsibility for the account and deal moves to the senior account manager, the specialists' established access through the opportunity team continues to support the ongoing sales process. The new owner can manage the team as needed, without needing to reconstruct the collaboration arrangement from scratch.

Salesforce documents opportunity teams as a feature for granting selected users access to an opportunity and defining their roles in the sales effort. For implementation details, see Salesforce Help: Opportunity Teams and Salesforce Help: Organization-Wide Defaults.

QUESTION NO: 5

What should an architect recommend to make sure that users that gained access to a custom object record through Apex managed sharing do not lose access to it when its owner is changed?

A. Use "With Sharing" keyword to make sure record visibility will beconsidered.

B. Create a new record in _Share object with RowCause "Manual".

C. Create aspecific Apex Sharing Reason for the custom object.

ANSWER: C

Explanation:

Create a specific Apex Sharing Reason for the custom object. Apex managed sharing on custom objects uses share records, and the `RowCause` value identifies why access was granted. Salesforce removes share records created with the `Manual` row cause when record ownership changes, because that access is treated as owner-dependent manual sharing. A custom Apex Sharing Reason creates a durable, developer-defined row cause that can be used when inserting records into the custom object's share table. Shares created with this custom reason are retained when ownership changes, so the users or groups granted access continue to have the intended record visibility. The sharing reason is configured in the custom object's Sharing Reasons area, then referenced through the corresponding generated `Schema.<Object>__Share.RowCause` value in Apex. This approach also makes the business purpose of the programmatic share explicit and supports controlled maintenance of the sharing logic. Salesforce documents that Apex sharing reasons are available for custom objects and are specifically intended to preserve Apex-managed shares through ownership changes. See Salesforce Apex Developer Guide: Creating Apex Managed Sharing and Salesforce Help: Define Apex Sharing Reasons.

QUESTION NO: 6

What should the Architect do to ensure Field-Level Security is enforced on a custom Visualforce page using the Standard Lead Controller?

A. Use the "With Sharing" keyword on the Standard Lead Controller.

B. Nothing; Field-Level Security will automatically be enforced.

C. Use the {!Schema.sObjectType.Lead.fields.isAccessible()} expression

D. Use the Schema.SObject.Lead.isAccessible() method.

ANSWER: B

Explanation:

Nothing; Field-Level Security will automatically be enforced. A Visualforce page that uses a standard controller benefits from the platform's built-in security enforcement. Standard controllers respect the running user's object permissions and field-level security when they retrieve, display, and save the controller record. Therefore, a user who lacks access to a Lead field cannot use the standard-controller-backed page to view or edit that field merely because the field is referenced in page markup. This behavior is a key benefit of using standard controllers rather than writing custom data-access logic.

The Architect should retain the standard controller for the Lead record and avoid introducing custom Apex or other direct data-access paths unless needed. If a custom controller or extension performs SOQL, DML, or exposes data independently of the standard controller, the developer must explicitly enforce appropriate object and field permissions in that custom logic. For the stated implementation, however, no additional field-access expression or sharing declaration is required to ensure

field-level security. Salesforce documents the security behavior of standard controllers and Visualforce security considerations at Standard Controllers and Visualforce Security.

QUESTION NO: 7

Which two options can be selected to share data with when creating a sharing rule?

Choose 2 answers

A. Roles

B. Public Groups

C. Users

D. Profiles

ANSWER: A B

Explanation:

Roles and **Public Groups** are valid targets for Salesforce sharing rules. A role-based sharing rule can grant record access to users assigned to a selected role, and it can also be configured around the role hierarchy where applicable. This supports scalable access management because access follows a user's organizational function rather than requiring individual record sharing. For example, a rule can share records owned by one role with users in another role to support cross-functional collaboration.

Public Groups are also designed as sharing-rule recipients. A public group can contain individual users, roles, roles and subordinates, territories, or other public groups, allowing an administrator to define a reusable audience with the needed access. Sharing rules can then grant that group read-only or read/write access, depending on the object and organization-wide default configuration. This approach centralizes maintenance: when group membership changes, the sharing rule continues to apply to the updated group automatically. Salesforce documents that owner-based and criteria-based sharing rules can share records with public groups and roles. See Salesforce Help: Sharing Rules and Salesforce Help: Control Access to Records.

QUESTION NO: 8

Which three areas should the Architect review in order to increase performance of "Record Access" and "Sharing" calculations?

Choose 3 answers.

A. Custom Object data, to ensure that no Account has more than 10,000 Custom Objects that look up to it.

B. Opportunity data, to ensure that no Account has more than 10,000 Opportunity records that are related to it.

C. Record ownership, to ensure that no user owns more than 10,000 Object records in the system.

D. Apex Managed Sharing triggers, to ensure that no trigger is querying more that 10,000 Object records.

ANSWER: A B C

Explanation:

Record-access performance is strongly affected by data-skew patterns because sharing recalculation work can expand substantially when many records are concentrated around the same parent or owner. Reviewing custom-object data where more than 10,000 records look up to one Account is important because this is a form of lookup skew. Salesforce must account for the relationship and applicable access when parent-related access changes occur. Reviewing Opportunity data associated with a single Account is equally important: a very large concentration of related records can create account-data skew and increase the scope and duration of sharing-related processing. Reviewing record ownership is also essential. When one user owns more than 10,000 records, ownership skew can make ownership changes, role changes, and sharing

recalculations more expensive because access derived from that owner may need to be evaluated for a large record population. An architect should identify these concentrations early, distribute ownership where feasible, and model relationships to limit skew before large-scale sharing changes are deployed. Salesforce describes these skew types and their implications in its Managing Data Skew guidance and provides additional sharing-design considerations in the Sharing Architecture documentation.

QUESTION NO: 9

Partner users can access records belonging to users in their account at their same role or lower in the role hierarchy, for Cases, Leads, Opportunities and Custom Objects. Which of the following access has to be given ?

A. Super user permission

ANSWER: A

Explanation:

Super user permission is required. In an Experience Cloud partner setup, enabling the Partner Super User capability gives a partner user broader access to records associated with other partner users in the same partner account. Specifically, the user can access eligible account-related records owned by users in their account who are at the same role or below them in the partner role hierarchy. This access model is intended for partner managers or senior partner contacts who need to oversee their organization's sales pipeline, service activity, and custom business processes without requiring record-by-record sharing.

The capability applies to the supported standard objects identified in the question—Cases, Leads, and Opportunities—as well as custom objects, subject to the organization's sharing configuration and object permissions. Administrators grant this through the partner user's contact/user configuration by enabling the super user setting, rather than by creating separate sharing rules for every partner user and record. Salesforce documents this feature as Partner Super User access and describes its role-hierarchy-based access within the partner account. See Salesforce Help: Grant Super User Access to Your Partners and Salesforce Developer Documentation: Partner Super Users.

QUESTION NO: 10

Universal Containers has set Opportunity Sharing to Private with Opportunity Teams enabled. Which three options can change the Owner of the Opportunity?Choose 3 answers.

A. Any Opportunity Team Member on the current Opportunity.

B. The current Opportunity Owner can transfer the current ownership.

C. The System Administrator or a user with the "Transfer Records" permission.

D. The user specified as the Manager on the Owner's User Profile.

E. Someone above the Opportunity Owner in the Role Hierarchy.

ANSWER: B C E

Explanation:

The current Opportunity Owner can transfer the current ownership because record owners are permitted to transfer records they own, subject to the required object-level access. Ownership transfer is a distinct action from simply granting record access, and it updates the Opportunity owner to another eligible user.

The System Administrator or a user with the "Transfer Records" permission can change ownership because administrative authority or the Transfer Record system permission enables a user to perform record transfers when the applicable object permissions are present. This permission supports centrally managed reassignment without requiring the user to be the current record owner.

Someone above the Opportunity Owner in the Role Hierarchy can change ownership of records owned by users below them. The role hierarchy provides upward record access, and Salesforce supports managers transferring records owned by subordinate users, assuming the manager has the necessary Opportunity access. Private Opportunity sharing limits broad peer access but does not prevent the owner, authorized transfer users, or eligible managers in the hierarchy from performing ownership transfers. Opportunity Team membership itself grants the configured record access; it does not make a team member an owner or confer ownership-transfer authority. See Salesforce's guidance on transferring records and role hierarchies.

QUESTION NO: 11

Universal Containers (UC) has a Private Account organization-wide default. Sales representatives regularly involve the same solution engineer and finance analyst on new customer accounts. These users need access to the Account and related Opportunity and Case records.

Which two features should an architect recommend to reduce manual sharing effort? Choose 2 answers

A. Configure a default Account Team for each sales representative.

B. Use Account Team access levels for Accounts, Opportunities, and Cases.

C. Assign View All Data to solution engineers and finance analysts.

D. Create a queue that owns all customer Accounts.

ANSWER: A B

Explanation:

A default Account Team lets a sales representative define recurring collaborators once and automatically add them to Accounts that the representative creates. This reduces repetitive setup when the same solution engineer and finance analyst support the representative's customers.

Account Team access levels can be configured separately for the Account, related Opportunities, related Cases, and Contacts. The representative can grant the appropriate levels of access to each team member without changing Account ownership or broadening the organization-wide default for all users.

See Salesforce documentation on default Account Teams and Account Teams.

QUESTION NO: 12

Universal Containers (UC) wants to reduce the amount of redundant leads entered into the system. UC also only edited/reassigned by the lead owner.

What organization-wide default (OWD) approach should be recommended to help UC implement these requirements?

A. Implement a Public Read Only OWD on Lead.

B. Implement a Public Read Only/Transfer OWD on Lead

C. Implement a private OWD on Lead.

D. Implement a Public Read/Write OWD on Lead.

ANSWER: A

Explanation:

Implement a Public Read Only OWD on Lead. This baseline sharing model lets all applicable users see lead records across the organization, even when they do not own them. Broad visibility is important for reducing duplicate lead entry: before creating a lead, a representative can search for and review existing lead records to determine whether the prospect is already represented in Salesforce. At the same time, Public Read Only does not grant nonowners the ability to edit lead

details. Modification remains controlled by record ownership, subject to any separately configured permissions, sharing mechanisms, or administrative access. This supports the requirement that a lead's owner retains responsibility for maintaining and reassigning the record while other users can identify existing leads. In Salesforce, organization-wide defaults establish the most restrictive baseline access level for records, and additional access can be granted only where a documented business need exists. Keeping Lead records publicly readable provides the visibility needed for duplicate prevention without making routine lead updates available to all users. See Salesforce's organization-wide defaults documentation and the Trailhead module on organization-wide defaults.

QUESTION NO: 13

Universal Containers (UC) has a custom object to track the internal net promoter score (NPS) for all of its employees. The manager is in the role above the owner and there are no sharing rules on the object.

How should UC ensure that NPS records cannot be accessed by the owner's manager?

A. RemoveCreate, Read, Edit, and Delete fromManager profiles and permission sets.

B. Use Apex sharing to remove NPS object share records for Manager profiles.

C. Set organization-wide default to Private and uncheck the Access Using Hierarchies option for the NPS object.

ANSWER: C

Explanation:

Set organization-wide default to Private and uncheck the Access Using Hierarchies option for the NPS object. A Private organization-wide default establishes that only record owners and users granted access through an explicit sharing mechanism can access NPS records. For custom objects, Salesforce also provides the *Access Using Hierarchies* setting. When this setting is disabled, users higher in the role hierarchy do not automatically receive access to records owned by users below them. Therefore, a manager who is above the record owner's role will not gain record access solely because of that hierarchy relationship. This configuration is appropriate for sensitive employee-feedback data such as internal NPS, where management should not automatically be able to view an employee-owned response. Object permissions should still be assigned as needed so users can create and work with their own NPS records; record-level access is then controlled separately by the Private default and the disabled hierarchy inheritance. Salesforce documents that hierarchy-based access can be disabled for custom objects, unlike standard objects, for which hierarchy access is generally enabled. See Organization-Wide Defaults and Role Hierarchy and Record Access.

QUESTION NO: 14

At Universal Containers, users should only see Accounts they or their subordinates own. All Accounts with the custom field "Kay Customer" should be visible to all Senior Account Managers. There is a custom field on the Account record that contains sensitive information and should be hidden from all users, except 3 designated users who require view and edit access. These three users come from different user groups, and will change occasionally. Which three platform security features are required to support these requirements with the minimum amount of effort? Choose 3 answers

A. Criteria-Based Sharing Rules

B. Owner-Based Sharing Rules

C. Role Hierarchy

D. Apex Managed Sharing

E. Permission Sets

ANSWER: A C E

Explanation:

Role Hierarchy supports the requirement that users can access Account records owned by users below them in the hierarchy. With Account organization-wide defaults set appropriately restrictively, the hierarchy grants managers record access to records owned by their subordinates while preserving a controlled sharing model. Salesforce describes how role hierarchies provide record access upward through an organization's reporting structure.

Criteria-Based Sharing Rules can automatically share every Account whose *Kay Customer* field meets the defined criterion with the Senior Account Managers, typically through a public group. This is low maintenance because records gain or lose access automatically as the field value changes, without requiring administrators to manage individual record shares.

Permission Sets provide the most maintainable way to grant the three designated users read and edit field-level security for the sensitive custom Account field. A permission set can expose the field and make it editable without changing the access of users who do not receive that permission set. As the designated users change, an administrator only assigns or removes the permission set. See Salesforce guidance on record access and sharing and field permissions in permission sets.

QUESTION NO: 15

Universal Containers maintains Job information in a Custom Object that contains sensitive information. The only users who should be able to view and edit Job records are the user who owns the record and all users in the Delivery profile. Which three platform sharing tools are required to support the above requirements?

Choose 3 answers.

A. Grant access Using Hierarchy sharing setting on the Job Object set to false.

B. "Modify All" permission for Job Object on the Delivery Profile.

C. Criteria-Based sharing rule for the Delivery Profile on the Job Object.

D. Organization-Wide Default sharing setting of Private on the Job Object.

E. "View All Data" profile permission on the Delivery Profile.

ANSWER: A B D

Explanation:

Setting the Job object's organization-wide default to Private establishes the required restrictive baseline: record owners retain access to their own Job records, while users who do not own a record receive no access merely because they are internal users. For a sensitive custom object, disabling Grant Access Using Hierarchies prevents users above a record owner in the role hierarchy from automatically gaining access. This ensures that access remains limited to the intended populations rather than being expanded through management reporting relationships.

Granting the Delivery profile the Modify All permission on the Job object gives every user assigned to that profile the ability to view, edit, and manage every Job record, independent of ownership and record-level sharing. The object permission therefore supplies the required broad access for the Delivery users, while the private default and disabled hierarchy access preserve the restriction for everyone else. Together, these settings provide owner access, Delivery-user access, and a private record-sharing model. Salesforce documents private organization-wide defaults and hierarchy behavior in its record access and sharing documentation, and describes object-level administrative permissions in Salesforce user permissions documentation.

QUESTION NO: 16

Universal Containers is a fast-growing company that sells containers globally. It has thousands of dealerships throughout the world where local dealers service Containers sold locally. They recently opened two dealerships in California: NorthCal and SoCal. Universal Containers implemented a new partner community to enable their dealers. Each dealership has a dealer Manager who has all service agents report into them. Assuming a private sharing model, what is the best option to enable dealer managers to have visibility to customer cases within their dealership and not across all dealerships?

A. Create sharing groups that share all cases to all agents under the Dealer manager.

B. Create a batch job that creates sharing rules as needed, based on the cases created.

C. Build a trigger that create manual sharing of cases as needed whenever a new case is created.

D. No changes are needed to the sharing and visibility model to implement this requirement.

ANSWER: D

Explanation:

No changes are needed to the sharing and visibility model to implement this requirement. In a partner community, partner users can be assigned to partner roles associated with their dealer account. The dealer manager is positioned above that dealership's service agents in the partner role hierarchy. With the Case organization-wide default set to Private, the role hierarchy provides the manager access to records owned by users in subordinate roles, including cases owned by the service agents who report to that manager. This access is inherited within that dealer's role branch, rather than being granted across unrelated dealer-account branches. As a result, the NorthCal manager can access cases owned by NorthCal service agents, while the SoCal manager can access cases owned by SoCal service agents, preserving dealership-level isolation. This design uses Salesforce's standard role-based record access model and remains maintainable as agents are added or moved within a dealership. Salesforce documents that roles control the level of access users have to records and that partner accounts have a corresponding partner role hierarchy. See Salesforce Roles documentation and Salesforce Partner Roles documentation.

QUESTION NO: 17

Universal Containers (UC) has a private Opportunity sharing model. Sales Operations has created dashboards showing pipeline data that should be available only to regional sales directors.

Which two actions should an architect recommend to meet these requirements?

Choose 2 answers

A. Share the dashboard folder with a public group containing the regional sales directors.

B. Grant the regional sales directors Modify All Data.

C. Create sharing rules that grant the regional sales directors access to the relevant Opportunities.

D. Make the Opportunity organization-wide default Public Read/Write.

ANSWER: A C

Explanation:

Share the dashboard folder with a public group containing the regional sales directors. Folder sharing controls who can open and run the dashboards stored in that folder. A public group provides a maintainable audience when directors are distributed across different regions or roles.

Also create sharing rules that grant the directors access to the relevant Opportunity records. Dashboard-folder access does not bypass object, field, or record-level security. When a director runs a dashboard, the underlying components return only data that the director is authorized to access, unless the dashboard is configured with an authorized running-user context. Therefore, the directors need appropriate Opportunity record access in addition to dashboard-folder access. See Salesforce guidance on sharing dashboard folders and sharing rules.

QUESTION NO: 18

An External Object is created to show Invoices from an external accounting system. When viewing the External Object, a user should only access invoice records the user is authorized to see.

What two actions are required to achieve the above requirement? Choose 2 answers

A. Setup External Object to use OAuth to connect to the Accounting system.

B. Create an owner based sharing rule to grant visibility to the Invoice object.

C. Restrict access to data in the accounting system.

D. Grant access to the External Object to only the Account Manager profile.

ANSWER: A C

Explanation:

External objects expose data that remains in an external system; Salesforce does not store those invoice records or independently apply Salesforce record-sharing rules to them. Consequently, record-level authorization must be evaluated by the external accounting system when it receives a request for invoice data. Restricting access to data in the accounting system ensures that its authorization model returns only the invoices that the requesting user is permitted to view.

Setting up the external object to use OAuth establishes a secure authenticated connection to the accounting system. To enforce access on a user-by-user basis, the integration should use an authentication configuration that conveys the individual user's identity or credentials to the external system, rather than relying solely on one shared integration identity. The accounting system can then apply its own permissions while servicing each request made through the external object. Salesforce documentation describes how external objects access data through external data sources and adapters, while authentication and authorization are handled as part of the external connection configuration. See Salesforce External Objects documentation and Salesforce external data source authentication guidance.

QUESTION NO: 19

A banking company uses a VIP Flag in the Contact Object that they want only Private Banking Reps to see.

Which approach is recommended to meet this requirement?

A. Change the type of VIP Flag field to a picklist, define a new record type for the Contact Object and make the picklist field available for Editing.

B. Define a page layout for Contact Object and add the VIP Flag field for that layout. Remove the VIP Flag field from other layouts.

C. Set the Field Level Security for the VIP Flag field so that it is visible to Private Banking Rep Profile.

ANSWER: C

Explanation:

Set the Field Level Security for the VIP Flag field so that it is visible to Private Banking Rep Profile. Field-level security is the appropriate control because the requirement is to protect the value of a sensitive Contact field, independently of a user's access to an individual Contact record. Configure the VIP Flag field as visible for the Private Banking Rep profile and keep it hidden for other relevant profiles. Salesforce enforces field-level security throughout the platform, including record detail pages, list views, reports, search results, and API access. This means users without field visibility cannot retrieve the VIP Flag value merely because they can access the Contact record through organization-wide defaults, sharing rules, teams, or other record-sharing mechanisms. The configuration therefore follows a defense-in-depth model: record access determines whether a user can access a Contact, while field-level security determines whether that user can access the VIP Flag data on Contacts they are permitted to view. If permission sets are used for the private-banking population, field visibility can also be granted through a permission set or permission set group, but the underlying recommended mechanism remains field-level security. Salesforce recommends using field-level security to control access to sensitive fields rather than relying on presentation-layer configuration. See Salesforce Field-Level Security documentation and Salesforce Security Implementation Guide: Field Permissions.

QUESTION NO: 20

Universal Containers (UC) operates worldwide with offices in more than 100 regions in 10 different countries and has established a very complex role hierarchy to control data visibility. In the new fiscal year UC is planning to reorganize the roles and reassign accounts owners.

Which three features could an architect recommend to avoid problems on this operation? Choose 3 answers

A. Partition data using Divisions

B. Deferred Sharing Recalculation

C. Parallel Sharing Rule recalculation

D. Skinny table

E. Granular Locking

ANSWER: B C E

Explanation:

Deferred Sharing Recalculation is appropriate because it lets administrators defer costly sharing recalculations while making a sequence of ownership, role, group, territory, or sharing-configuration changes. After the reorganization is complete, UC can resume calculations and process the resulting sharing updates in a controlled manner, rather than repeatedly recalculating access after every individual change. This is particularly valuable where a complex role hierarchy causes many inherited access changes.

Parallel Sharing Rule recalculation helps reduce the elapsed time for recalculating sharing rules by allowing Salesforce to process eligible recalculation work concurrently. That capability is useful when the organizational change affects a large number of records and sharing-rule-derived access must be rebuilt.

Granular Locking reduces lock contention during high-volume ownership updates by using more targeted locks for account-related operations. Since account ownership changes can update related sharing and hierarchy access, minimizing lock conflicts makes a large reassignment operation more reliable and easier to execute at scale. Salesforce documents deferred calculations and related sharing-recalculation behavior in its Deferred Sharing Calculations guidance, and explains lock behavior and mitigation in the Record Locking Cheat Sheet.

QUESTION NO: 21

Universal Containers would like to control access to records and objects according to the following business requirements: Sales users can view all account records but only edit their own records. Sales managers can view all account records but only edit records of their team. Service users can view all account records that are not marked with a RecordType of Prospect.

Which organization-wide default configuration should an architect recommend to fulfill these requirements?

A. Public Read Write

B. Public Read/Transfer

C. Private

ANSWER: C

Explanation:

Private is the appropriate Account organization-wide default because it establishes the restrictive baseline needed to grant access selectively by user population and record criteria. With a private Account model, sales users initially have access only to the records they own, preserving the requirement that they can edit only their own accounts. An owner-based sharing rule can then grant the Sales user group read-only access to all Account records, allowing universal visibility without expanding edit capability.

Sales managers can receive access to records owned by users below them through the role hierarchy. When Account access through hierarchies is enabled, this provides managers with the necessary read/write access to their team's records while retaining the private baseline for records outside their reporting hierarchy. A criteria-based sharing rule can separately grant Service users read-only access to Account records whose record type is not Prospect. This rule can target the required records without exposing Prospect accounts to Service users.

Using Private as the baseline and explicit sharing mechanisms for each requirement follows Salesforce's least-privilege approach to record access. See Salesforce's organization-wide defaults documentation and sharing rules documentation.

QUESTION NO: 22

Which two options can help mitigate the risks of import failures associated with large-volume bulk data loads?

Choose 2 answers.

A. Minimize user group hierarchy.

B. Defer Sharing Calculation.

C. Increase batch size.

D. Group records by ParentID within a batch.

ANSWER: B D

Explanation:

Defer Sharing Calculation. helps make a large data load more reliable when record changes would otherwise trigger extensive sharing recalculations. Deferring those calculations lets Salesforce process the import without repeatedly recalculating sharing access for each inserted or updated record. The sharing work can then be resumed and processed after the load, reducing contention and the likelihood that the bulk operation is disrupted by sharing-related processing overhead.

Group records by ParentID within a batch. is also a key bulk-loading practice for records that have parent-child relationships. Keeping child records for the same parent together improves processing locality and reduces cross-batch locking contention on parent records. This is particularly important when many child records reference a smaller set of parent records, because concurrent batches that touch the same parents can encounter lock-related failures. Organizing batches by parent identifier therefore makes the import more predictable and more resilient at high volume. Salesforce documents these approaches in its data-load optimization guidance and guidance for deferring sharing calculation.

QUESTION NO: 23

Universal Containers has the following requirements:

* The Commercial Account and Consumer Account support departments should not collaborate.

* The Commercial and Consumer sales users roll up to the same VP of Sales, but there should be no collaboration between sales departments.

* The Commercial sales department should share its customers with the Commercial support department.

* The Consumer sales department shares its customers with the Consumer support department.

* The Commercial and Consumer support departments roll up to the same Support Director.

* The sales departments will remain the Account Owner for the Accounts that they sell to.

What is the recommended Org-Wide Sharing Default for Accounts, and how would the

Architect enable proper Commercial and Consumer Sales to Support Account Sharing for this scenario?

A. Private Account Sharing with Sharing Rules from Commercial Sales Role(s) to Consumer Support Role(s) and Consumer Sales Role(s) to Commercial Support Role(s).

B. Private Account Sharing with Sharing Rules from Commercial support Role(s) to Commercial Support Role(s) and Consumer Sales Role(s) to Consumer Support Role(s).

C. Read-Only Account Sharing with Sharing Rules from Commercial Sales Role(s) to Consumer Support Group(s) and Consumer Sales Role(s) to Commercial Support Groups(s).

D. Private Account Sharing with Sharing Rules from Commercial Sales Group(s) to Commercial Support Groups(s) and Consumer Sales Group(s) to Consumer Support Group(s).

ANSWER: D

Explanation:

Private Account Sharing with Sharing Rules from Commercial Sales Group(s) to Commercial Support Groups(s) and Consumer Sales Group(s) to Consumer Support Group(s). is correct because a Private organization-wide default establishes least-privilege access: Account records are initially available only to the owner, users above that owner in the role hierarchy where applicable, and users granted access through explicit sharing. This is appropriate because Account ownership remains with the relevant sales users and neither sales nor support population should receive broad default access to the other department's customers.

Owner-based Account sharing rules can then selectively grant access based on the group to which the Account owner belongs. A rule can share Accounts owned by the Commercial Sales public group with the Commercial Support public group, while a separate rule shares Accounts owned by the Consumer Sales public group with the Consumer Support public group. Public groups provide a maintainable way to represent each business population and ensure that sharing is aligned to the intended Commercial-to-Commercial and Consumer-to-Consumer relationships. The resulting access model preserves sales ownership while giving the matching support team the Account visibility needed to provide service.

See Salesforce's organization-wide default sharing guidance and sharing rules documentation.

QUESTION NO: 24

For the Universal Containers Commercial and Consumer support departments, having access to Activities for Contacts with which they interact is important. Commercial support users should not see Consumer Accounts/Contacts and Consumer support users should not see Commercial Accounts/Contacts. Assuming the Organization-Wide Default for Activities is set to "Controlled by Parent" what is the minimum level of Sharing access a support user would need to Accounts/Contacts to view associated Activities?

A. Private Account/Contact Sharing Default with a Sharing Rule for each department set to Public Read/Write access to Accounts/Contacts.

B. Private Account/Contact Sharing Default with a Sharing Rule for each department set to Private access to Accounts/Contacts.

C. The users need no access to Accounts/Contacts with the proper Activity Sharing Rules and Profile Permissions for the Accounts Tab.

D. Private Account/Contact Sharing Default with a Sharing Rule for each department set to Public Read only access to Accounts/Contacts.

ANSWER: D

Explanation:

Private Account/Contact Sharing Default with a Sharing Rule for each department set to Public Read only access to Accounts/Contacts is correct because an activity whose organization-wide default is *Controlled by Parent* derives access from its related parent record. For activities related to contacts, the user must therefore be able to read the relevant contact (and, where applicable, the related account) in order to view the associated activity. Read access is the minimum necessary level; edit access to the parent is not required merely to view activities. Keeping the Account and Contact defaults private establishes the required separation between Commercial and Consumer records. Department-specific sharing rules can then grant only the appropriate users read access to the records belonging to their own department, which in turn makes the related controlled-by-parent activities visible. This design applies least-privilege access while preserving the intended departmental data boundary. Salesforce documents that activity access can be controlled by the related record through the Controlled by Parent setting, and that sharing rules extend record access to selected groups of users. See Salesforce Activity Sharing and Salesforce Sharing Rules.

QUESTION NO: 25

Which features does Salesforce provide for restricting login access to the application?

Choose 2 answers.

A. Profile-based login hour restrictions

B. Role-based IP restrictions

C. Organization-wide login hour restrictions

D. Profile-based IP restrictions

ANSWER: A D

Explanation:

Salesforce supports restricting when and where users can access the application through settings configured on profiles. **Profile-based login hour restrictions** let an administrator define the days of the week and time ranges during which users assigned to a profile may log in. This is useful when access must align with operating hours, staffing schedules, or internal security policies. When a user attempts to log in outside the allowed hours, Salesforce prevents the login.

Profile-based IP restrictions provide a location-based control by defining trusted IP address ranges on a profile. Users assigned to that profile can log in only from an IP address within one of the configured ranges. This enables differentiated restrictions for groups with distinct access needs, such as employees who must use a corporate network or administrators who require tightly controlled access. These profile settings can be combined, so a user must satisfy both the applicable login-hour window and allowed IP-range requirement. Salesforce documents login-hour and login-IP-range controls as profile login restrictions: Salesforce Help: Restrict Login Access and Salesforce Help: Set Login Restrictions.

QUESTION NO: 26

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 aWho Sees What report, filtering on Account.

B. Leg in as 3 user and check several accounts to isolate the problem records.

C. View Field Accessibility in the Object Manager.

ANSWER: C

Explanation:

View Field Accessibility in the Object Manager. is the appropriate troubleshooting step because a field can be absent from a user's Account record page even when the assigned page layout has not changed. Field-level security controls whether a field is visible, read-only, or editable for users based on their profile and permission sets. A deployment can modify field permissions, permission sets, profiles, or permission-set-group assignments, causing users to lose visibility to an otherwise unchanged field.

In Object Manager, the administrator can open the Account field and use Field Accessibility to review the field's visibility across profiles and permission sets. This provides a direct way to identify whether the affected users' access changed and whether the field is hidden by field-level security. The administrator should also verify the affected users' assigned profile and permission sets, including any recent deployment changes affecting those assignments. Salesforce documents field-level security as a separate control from page layouts: a field must be visible through field-level security before it can appear to a user on a layout. See Salesforce Field-Level Security documentation and Salesforce Field Accessibility documentation.

QUESTION NO: 27

Assuming Person Account is enabled in a Salesforce organization, which three objects can be configured as "Controlled by Parent" in Sharing Settings?

Choose 3 answers

A. Opportunity

B. Order

C. Asset

D. Contact

E. Case

ANSWER: B C D

Explanation:

With Person Accounts enabled, **Order**, **Asset**, and **Contact** can be configured with the *Controlled by Parent* organization-wide default. This sharing model derives a child record's access from the related parent account. Consequently, a user's ability to view or edit an order, asset, or contact is determined by that user's access to the associated account, including a person account where applicable. This model helps maintain consistent visibility for records that belong to the same customer while avoiding separate, record-level sharing administration for each child object. For a person account, Salesforce represents an individual through an account record paired with an underlying contact record, so parent-based access is particularly important for keeping the person account and its associated customer records aligned. Administrators configure these defaults from Sharing Settings, then use account access and sharing mechanisms to control the resulting access to related records. Salesforce describes person accounts as a combined account-and-contact model and documents the account relationship that underpins this behavior. See Salesforce Help: Person Accounts and Salesforce Help: Organization-Wide Defaults.

QUESTION NO: 28

Which functionality does the system method "runAs()" verify when writing test methods?

A. Enforcement of a user's record sharing

B. Enforcement of a user's Field Level Security

C. Enforcement of a user's permissions

ANSWER: A

Explanation:

Enforcement of a user's record sharing is correct because `System.runAs()` executes the enclosed Apex test code in the context of a specified user for record-sharing purposes. This lets a test create users with different roles or sharing access, run queries or operations as each user, and confirm that the organization's sharing model produces the intended record visibility. For example, a test can use `System.runAs(user)` to verify that a user can access a record shared through an ownership-based, role-based, criteria-based, manual, or Apex-managed sharing mechanism. This is especially important when validating custom sharing logic and the effects of organization-wide defaults. Salesforce documents that the method changes the user context for sharing enforcement in test methods, enabling tests to evaluate whether records are available to the intended users under the configured sharing rules. The method is therefore a core testing tool for confirming record-level access behavior in a multiuser Salesforce implementation. See Salesforce's System.runAs documentation and the Apex testing guide.

QUESTION NO: 29

Universal Container has developed a custom Visualforce page that will accept user input and must prefer returning the results to the users.

Which two techniques should be used to ensure the users cannot perform a SOQL injection attack?

A. Escape double quotes in the user input.

B. Use bind variable in the SOQL query.

C. Use the escapesinglequotes() method to sanitize user input.

D. Use the with Sharing keyword on the controller.

ANSWER: B C

Explanation:

Using bind variables in the SOQL query is the preferred defense whenever user-supplied input is used as a value in a query. A bind variable keeps the input separate from the SOQL command structure, so Salesforce treats the supplied value as data rather than as executable query syntax. For example, a query such as `WHERE Name = :userInput` prevents embedded quotes or SOQL keywords in `userInput` from changing the query's intended conditions.

Using the `String.escapeSingleQuotes()` method is also an appropriate safeguard when dynamic SOQL must be built as a string and a user value cannot be represented by a bind variable. The method escapes single-quote characters before the value is concatenated into the dynamic query, preventing that input from prematurely terminating a quoted string literal and altering the SOQL statement. In practice, developers should use bind variables wherever possible and apply escaping carefully to values incorporated into dynamic SOQL. Salesforce's secure coding guidance identifies both bind variables and escaping single quotes as SOQL-injection mitigations. See Salesforce Secure Coding Guide: SOQL Injection and Salesforce Apex String Methods Reference.

QUESTION NO: 30

To grant Universal Containers sales manager access to shipment records properly, it was necessary to leverage Apex managed sharing. The IT team is worried about improper access to records.

Which two features and best practices should a Salesforce architect recommend to mitigate this risk?

A. Use runAs system method in test classes to test using different users and profiles.

B. Use with Sharing keyword in Apex classes to assure record visibility will be followed.

C. Use isShareable in Apex classes to assure record visibility will be followed.

D. Use isAccessible keyword in Apex classes to assure record visibility will be followed

ANSWER: A B

Explanation:

Use `with Sharing` keyword in Apex classes to assure record visibility will be followed. This declaration causes Apex code to honor the current user's record-level sharing rules, including organization-wide defaults, role hierarchy access, sharing rules, teams, and manually or programmatically created shares. For code that queries or updates shipment records after Apex managed sharing has been applied, explicitly declaring the sharing mode makes the intended security behavior clear and helps prevent accidental execution in a context that bypasses record sharing.

Use `runAs` system method in test classes to test using different users and profiles. Tests should create representative users, execute relevant logic under each user context, and assert both the expected access for sales managers and the absence of access for users who should not see the shipment records. This verifies that the managed-sharing implementation and the Apex sharing context produce the intended record visibility before deployment and after future changes. Salesforce documents the effect of Apex sharing declarations at Apex Sharing and describes user-context testing with `System.runAs` at Using runAs in Tests.

QUESTION NO: 31

Universal Containers uses Person Accounts to represent retail customers and Business Accounts to represent commercial customers. The retail sales team should not have access to commercial customers but should have access to ALL retail customers.

With the organization-wide default on Account set to Private, how should the architect meet these requirements?

A. Create an owner-based sharing rule on AccountContactRelation to grant a@coess to all account contact roles records owned by retail sales reps.

B. Update the Retall Sales profile to grant access to Person Account record type.

C. Create o criteria-based sharing rule giving the Retail Sales role access to Accounts of type PersonAccount.

ANSWER: C

Explanation:

Create a criteria-based sharing rule giving the Retail Sales role access to Accounts with the Person Account record type. Person Accounts are implemented as Account records that use a Person Account record type, so they follow the Account object's organization-wide default and its sharing model. Because the Account default is Private, retail users do not automatically receive access to retail customer records owned by other users. A criteria-based Account sharing rule can select every Account whose record type is the Person Account record type and share those records with the Retail Sales role. This provides the retail team access across all retail customers while leaving Business Account records outside the sharing-rule criteria.

The architect should select the appropriate access level required by the sales process, such as Read Only when viewing is sufficient or Read/Write when retail representatives must update customer records. Using a record-type criterion is durable as new Person Account records are created because qualifying records are shared automatically. Salesforce documents that Person Accounts are Account records with person-specific behavior and are identified through their record type; sharing rules can grant access based on record criteria. See Salesforce Help: Person Accounts and Salesforce Help: Sharing Rules.

QUESTION NO: 32

In order to comply with regulatory requirements, Universal Health must encrypt all Personally Identifiable Information (PII), both while it is being transmitted over the network and while it is at rest. Universal Health has completed a data audit and has determined that 12 fields on the contact record can contain PII, including the contact name and several other standard fields. Universal Health would like the fields to remain accessible in Salesforce. Which two options does Universal Health have to maintain compliance?

Choose 2 answers.

A. Implement a custom Apex trigger to automatically encrypt the PII data using the Apex Crypto Class.

B. Update the field type of each of the 12 fields to "Text (Encrypted)" so that they are encrypted at rest.

C. Enable Salesforce Platform Encryption and select the 12 contact fields to be encrypted.

D. Use an external, third party encryption service to encrypt PII before it enters Salesforce.

ANSWER: C D

Explanation:

Enabling Salesforce Platform Encryption and selecting the applicable Contact fields is an appropriate Salesforce-native approach for protecting supported PII fields at rest. Platform Encryption is part of Salesforce Shield and encrypts selected field data while preserving Salesforce functionality according to the capabilities and limitations of the selected encryption scheme. Salesforce also protects data in transit through TLS for supported browser and API connections, so this approach addresses both the platform's at-rest protection and secure network transport requirements. See Salesforce's Platform Encryption overview.

Using an external, third-party encryption service before PII enters Salesforce is also a valid architecture when the organization needs to control encryption outside the Salesforce trust boundary. Salesforce stores the encrypted ciphertext, while an authorized integration, application, or controlled user experience can use the external service and its key-management process to decrypt values when access is required. Transport to Salesforce should use TLS, and the organization must design the encryption/decryption workflow so authorized users can continue to access the information as

needed. Salesforce's data-encryption guidance describes the distinction between platform encryption and customer-managed encryption approaches.

QUESTION NO: 33

A sales rep at Universal Containers (UC) is a member of the Default Opportunity team for an

account manager. The account manager created an opportunity and the sales rep is added to that Opportunity team.

The sales rep is complaining about no longer having access to an opportunity record that the sales rep was helping with.

What is the cause of this problem?

A. The Account team was changed and consequently the Opportunity team members were replaced by the Account team members.

B. The Sales rep was manually removed from the Opportunity team.

C. The Sales rep was removed from the Opportunity team in another opportunity record of the same account.

D. The opportunity owner can enable/disable if the "Default Opportunity team" is able to access the record

ANSWER: B

Explanation:

The Sales rep was manually removed from the Opportunity team. Opportunity teams grant their members record-level access to a specific opportunity, with the access level defined for that team member. A default opportunity team is a reusable set of users that an opportunity owner can apply when creating opportunities; once those users are included on an individual opportunity's team, their access is associated with that particular opportunity-team membership. If the sales rep subsequently loses access to the opportunity, removal of that membership from the opportunity team is the direct cause described by the scenario. Salesforce supports adding, changing, and removing members for an individual opportunity team, and a user's access provided by the team no longer applies after the user is removed. This behavior enables the opportunity owner to keep collaboration aligned to the people currently working on each deal. See Salesforce's guidance on opportunity teams and the use of default opportunity teams.

QUESTION NO: 34

Universal Containers (UC) has created a public group with certain Sales Engineers to help on complex deals and a sharing rule to grant access to these opportunities. Opportunity OWD is private.

What is the impact of these sharing settings?

A. Subordinates of Managers who have Sales Engineers in the public group will also have access to these records.

B. Sales Engineers that have a similar role of the Sales Engineers of the public group will also have access to these records.

C. Sales Engineers Managers and their managers in the role hierarchy will also have access to these records.

D. Sales Engineers direct reports will also have access to these records.

ANSWER: C

Explanation:

Sales Engineers Managers and their managers in the role hierarchy will also have access to these records. Opportunities are a standard object, and access through the role hierarchy is enabled for standard objects. This means that when a sharing rule grants members of the public group access to private Opportunity records, users above those group members in the role hierarchy can inherit that access. The inherited access supports the management structure: managers can review and collaborate on records available to users beneath them, including records made available through sharing.

The key distinction is that the sharing rule explicitly grants access to the public group, while the role hierarchy extends record visibility upward from each recipient to their managers. The Opportunity organization-wide default being Private prevents broad baseline access, but it does not disable the role-hierarchy behavior for this standard object. The resulting access is limited to the managers above the Sales Engineers who receive the shared Opportunity access, continuing upward through the hierarchy as applicable. Salesforce documents that users higher in a role hierarchy can access records owned by, or shared with, users below them. See Salesforce role hierarchy documentation and Salesforce sharing rules documentation.

QUESTION NO: 35

Mary is Joe's manager in the role hierarchy. The OWD for a custom Invoice object is Public ReadOnly and Mary's profile is not granted the Read permission for the Invoice object.

What action can Mary take on Joe's Invoice records?

A. Read/Write

B. Edit Only

C. None

D. View Only

ANSWER: C

Explanation:

None is correct because Salesforce requires object-level permission before a user can access any records of that object. Organization-Wide Defaults, role hierarchy access, sharing rules, teams, and manual sharing determine record-level access only; they do not grant the underlying object permissions. Although the Invoice object is Public Read Only, that setting would make records available for viewing only to users who already have permission to read the Invoice object. Similarly, Mary's position above Joe in the role hierarchy can extend access to records owned by users below her, but it cannot bypass the absence of the Read object permission on Mary's profile or permission set. Because Mary has no Read permission for Invoice, she cannot open, view, edit, or otherwise access Joe's Invoice records. An administrator must grant Mary object-level Read access, such as through her profile or a permission set, before Public Read Only sharing or hierarchy-based access can provide visibility to the records. Salesforce describes object permissions as the baseline gate for access, with sharing settings applying afterward at the record level. See Salesforce object permissions documentation and Salesforce organization-wide defaults documentation.