DUMPSBOSS.

Salesforce Certified B2C Commerce Architect

Salesforce B2C-Commerce-Architect

Version Demo

Total Demo Questions: 13

Total Premium Questions: 139

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, Architecture and Design	25
Topic 2, Account and Site Structure	10
Topic 3, Data Management	39
Topic 4, Application Development	16
Topic 5, Integrations	38
Topic 6, Security	11
Total	139

QUESTION NO: 1

Northern Trail Outfitters (NTO) operate 200 physical stores. NTO has products that are available in some of the physical stores and not available in others. The closest physical store is determined based on customer s post zip code when they are shopping online. Only the products that are available in the customer's closest physical store should be presented to the customer to the search results.

What are the two feasible technical approaches to meet these requirements?

Choose 2 answers

A. Create a separate shipping method per physical store. Use post/zip code to determine the applicable shipping method. Show only the products that are not excluded from the shipping method.

B. Create a separate category per physical store use post/Tip code with a mapping to determine the relevant category. Show only the products from this category.

C. Create a separate site per physical store. Use post/zip code to redirect the customer to the relevant site. Show the products from the site navigation catalog.

D. Create a separate pricebook per physical store. Use post/zip code to activate this pricebook through a customer group. Show only the products with price by applying price refinement.

ANSWER: A D

Explanation:

Creating a separate shipping method per physical store can support store-specific assortment filtering because B2C Commerce shipping methods can be configured with product exclusions. After the shopper’s postal code is used to identify the applicable store and corresponding shipping method, the storefront can apply that shipping method to the product search. The search result then reflects the products eligible for that method, which models inventory availability for the selected physical store. This approach also keeps the availability rule in standard Business Manager shipping configuration.

Creating a separate pricebook per physical store is also feasible when a store’s sellable assortment is represented by the products that have prices in that store-specific pricebook. Postal-code logic can identify the store, and the relevant pricebook can be activated through the applicable customer group or otherwise applied to the shopper context. A price refinement can then limit displayed search results to products with an applicable price, ensuring that the shopper sees the store’s available assortment. B2C Commerce supports pricebook assignment and price-based product search behavior, making this a viable configuration-led solution. See Salesforce documentation on working with price books and configuring shipping methods.

QUESTION NO: 2

a B2C Commerce developer has Implemented a job that connects to an SFTP, loops through a specific number of .csv rtes. and Generates a generic mapping for every file. In order to keep track of the mappings imported, if a generic mapping is created successfully, a custom object instance w created with the .csv file name. After running the job in the Development instance, the developer checks the Custom Objects m Business Manager and notices there Isn’t a Custom Object for each csv file that was on SFTP.

What are two possible reasons that some generic mappings were not created? Choose 2 answers

A. The maximum number of generic mappings was reached.

B. The generic mappings definition need to be replicated from Staging before running the job.

C. Invalid format in one or more of the .csv files.

D. The job needs to run on Staging and then replicate the generic mappings and custom objects on Development

ANSWER: A C

Explanation:

The maximum number of generic mappings was reached because B2C Commerce places a platform limit on the number of generic mappings that can exist in an instance. Once that limit is reached, a job cannot create additional mappings. Since the custom object is created only after successful mapping creation, files processed after the limit is encountered will have no corresponding tracking object in Business Manager. A robust job should check for this condition, log the affected file names, and apply a retention or cleanup process for mappings that are no longer needed.

Invalid format in one or more of the .csv files is also a valid cause. A generic mapping is built from structured source data, so each input file must conform to the expected delimiter, column layout, encoding, and mapping requirements used by the job. If a file is malformed or cannot be parsed, the mapping creation operation can fail for that specific file. Because the custom object is conditional on successful creation, no object is written for that failed import. The job should validate input before creating the mapping and record parsing or validation errors in job logs. Salesforce documents job execution and troubleshooting concepts in the B2C Commerce Jobs guide and provides Business Manager administration guidance in the Business Manager documentation.

QUESTION NO: 3

There Is an Issue with the site when the domain Is opened from Google search results. After researching the problem. It turns out that the site returns * 404 page error when accessed with a parameter in the URL.

What should the Architect recommend to fix that issue?

A. Add dynamic catch-all rule to redirect to home page.

B. Add the provided snippet to the aliases configuration for the domain.

C. Add dynamic redirect if the URL contains parameter to Home Show.
Add this snippet to the aliases configuration for the domain

ANSWER: B

Explanation:

Add the provided snippet to the aliases configuration for the domain. In B2C Commerce, aliases determine how an incoming host, path, and applicable request conditions resolve to a storefront route. A URL reached from a Google result can include a query parameter, such as an analytics or campaign parameter. The alias configuration must therefore include the appropriate conditional mapping so that the parameterized request is still resolved to the intended storefront route, typically `Home-Show`, rather than falling through to a 404 response. This is a routing configuration concern at the site-domain boundary, not an application-level redirect concern. Configuring the alias allows Commerce to recognize the domain request consistently while preserving the expected canonical storefront behavior and avoids relying on a catch-all redirect that could mask legitimate missing-page conditions. The Architect should update the aliases file for the relevant domain and replicate or activate the configuration through the normal B2C Commerce configuration process. Salesforce documents aliases and URL-rule configuration as the supported mechanism for mapping incoming storefront URLs to pipelines/controllers and managing URL behavior: Salesforce B2C Commerce URL Rules and Salesforce B2C Commerce Aliases.

QUESTION NO: 4

A developer is remotely fetching the reviews for a product.

Assume that it's an HTTP GET request and caching needs to be implemented, what consideration should the developer keep in mind for building the caching strategy?

A. Cache the HTTP service request

B. Remote include with caching only the reviews

C. Use custom cache

D. Cached remote include with cache of the HTTP service

ANSWER: A

Explanation:

Cache the HTTP service request is correct because product reviews retrieved through an HTTP GET service call are well suited to response caching at the service layer. Salesforce B2C Commerce service definitions support caching for eligible GET requests, allowing Commerce to reuse a previously retrieved response rather than making the same outbound call for every shopper request. This reduces latency for the storefront, lowers load on the external reviews provider, and centralizes the caching behavior with the integration that owns the remote request.

The cache duration should reflect how frequently review data changes and the business need for freshness. The request must also include all inputs that affect the response, such as the product identifier, so responses remain correctly scoped to the requested product. Service-level caching is particularly appropriate here because the data originates from an external endpoint and the HTTP method is safe to retrieve repeatedly. Salesforce documents service integration and configuration concepts in the B2C Commerce Web Services guide, and provides general cache-behavior guidance in the B2C Commerce caching guide.

QUESTION NO: 5

A developer is checking for Cross Site Scripting (XSS) and found that the quick search is not escaped (allows inclusion of Javascript) in the following script:

How would the developer resolve this issue?

A. Replace ' with double Quote*

B. Use

C. Use < isprint value= ' ${searchPhrase} encoding- ' jsblock " / >

D. Use < toprint value= " ${searchPhrase} " / >

ANSWER: B

Explanation:

Use `<isprint value="${searchPhrase}" encoding="jshtml" />` to render the search phrase safely where JavaScript is present in an HTML page. The `isprint` tag is the B2C Commerce ISML mechanism for output encoding dynamic values. Its `jshtml` encoding mode protects a value that is placed into JavaScript embedded in HTML by encoding characters that could otherwise alter the JavaScript or HTML parsing context. As a result, data submitted through quick search is treated as text rather than as executable markup or script content. This is particularly important for search terms because they originate from an HTTP request and must be considered untrusted, even when they are later displayed back to the shopper. Applying contextual output encoding at the point where `searchPhrase` is rendered is the appropriate XSS defense; it preserves benign search text while preventing attacker-controlled content from becoming active code. Salesforce documents the available `isprint` encoding contexts, including `jshtml`, in the ISML isprint documentation. Salesforce's B2C Commerce security guidance also emphasizes encoding untrusted output in its destination context.

QUESTION NO: 6

An Architect is designing an integration that sends order-confirmation data to an external loyalty platform after an order is placed. The loyalty platform can occasionally be unavailable, but an outage must not prevent customers from placing orders.

Which two design decisions should the Architect include?

Choose 2 answers

A. Persist the outbound loyalty request before asynchronous processing.

B. Use an order-based identifier to make the loyalty request idempotent.

C. Call the loyalty platform synchronously before placing the order.

D. Store failed loyalty requests only in the shopper session.

E. Mark the request as successfully delivered before the platform responds.

ANSWER: A B

Explanation:

Persisting the outbound loyalty request with a unique order-based identifier before asynchronous processing provides durable recovery state when the loyalty platform is unavailable. A scheduled job can select pending records, submit them outside the checkout request, and retain failed records for later retry. This prevents a temporary external outage from affecting order placement.

Making the loyalty-platform request idempotent using an order-based identifier is also necessary. Jobs can retry after network failures where it is unknown whether the external platform received the first request. An idempotency key lets the receiving platform recognize a repeated submission and avoid issuing duplicate loyalty points. Together, durable pending state and idempotent delivery support reliable asynchronous integration processing. Salesforce documents background processing through the B2C Commerce Jobs documentation and custom persistence through Custom Objects.

QUESTION NO: 7

A business wants to migrate its customerservice provider from provider A to provider B. Provider e offers a LINK cartridge to integrate with its commerce solution.

Which three artifacts need to be created by the Architect? Choose 3 answers

A. Document the design of implementing a new B2C Commerce cartridge following the Industry standard best practices

B. Document the data objects, the interface, and data synchronization frequency between the systems.

C. Document the data mapping between commerce and customer service provider.

D. Document the customizations required on top of the LINK cartridge based on current commerce implementation and business needs.

E. Document how the customer online journey flown from landing on the page to placing of the order

ANSWER: A B D

Explanation:

“Document the design of implementing a new B2C Commerce cartridge following the Industry standard best practices,” “Document the data objects, the interface, and data synchronization frequency between the systems,” and “Document the customizations required on top of the LINK cartridge based on current commerce implementation and business needs” are the required architecture artifacts. A LINK cartridge provides an established integration foundation, but it must be assessed and incorporated into the merchant’s existing cartridge architecture in a maintainable way. The implementation design establishes cartridge responsibilities, integration touchpoints, deployment considerations, and alignment with established B2C Commerce development conventions.

The integration artifact must define the participating business objects, the systems’ interface contract, and when synchronization occurs. This gives both systems a shared, implementable understanding of ownership, data exchange, and operational timing. Finally, an architect must identify the modifications needed beyond the packaged cartridge. These customizations should be explicitly documented so the implementation supports the merchant’s current storefront behavior and business requirements while preserving a clear upgrade and support strategy. Salesforce describes LINK cartridges as prebuilt integrations and provides implementation resources through its LINK program documentation: Salesforce B2C Commerce LINK Cartridges. For the broader platform integration model and service considerations, see B2C Commerce Web Services.

QUESTION NO: 8

A third party survey provider offers both an API endpoint for individual survey data and an SFTP server endpoint that can accept batch survey data. The initial implementation of the integration includes

1. Marking the order as requiring a survey before order placement

2. On the order confirmation pace, the survey form is displayed for the customer to fill

3. The data is sent to the survey provider API, and the order it marked as not requiring a survey

Later it was identified that this solution is not fit for purpose as the following issues and additional requirements were identified:

1. If the API call fails, the corresponding survey data is lost. The Business requires to avoid data loss.

2. Some customers skipped the form. The Business require sending a survey email to such customers.

3. The Order Management System (OMS) uses a non-standard XML parser it did not manage to parse orders with the survey, until the survey attribute was manually removed from the xml.

How should the Architect address the issues and requirements described above?

A. Create a custom session attribute when the survey is required. Send to the API endpoint in real-time. On failure, capture the survey data in the session and reprocess, use me session attribute to send emails for the cases when survey was skipped.

B. Create a custom object to store the survey data. Send to the API endpoint using a job. On success, remove the custom object. On failure, send the survey data with API from the next execution of the same job. Use the custom object to send emails for the cases when the survey was skipped.

C. Create a custom object when the survey is required Send to the API endpoint in real-time. On success, remove the object. On failure, capture the survey data in the custom object and later reprocess with a job. Use the custom object to send emails for the cases when survey was skipped.

D. Send the survey data to the API endpoint in real-time until the survey data is successfully captured. Instruct the OMS development team to update their XML parser, use the Order survey attribute to send emails for the cases when the survey was skipped.

ANSWER: C

Explanation:

Create a custom object when the survey is required Send to the API endpoint in real-time. On success, remove the object. On failure, capture the survey data in the custom object and later reprocess with a job. Use the custom object to send emails for the cases when survey was skipped. This design creates durable, independent integration state before the external call is attempted. The custom object can retain the survey payload, customer/order correlation information, completion status, and retry metadata, so a transient API outage cannot cause the submitted survey data to disappear. A successful synchronous call still provides timely delivery; deleting or marking the custom-object record as complete after success prevents duplicate processing.

When delivery fails, a scheduled B2C Commerce job can locate pending records and retry them outside the storefront request. This makes recovery reliable and avoids tying retry behavior to a shopper session or to the original confirmation-page request. The same persisted record also identifies surveys that were required but never submitted, enabling an email job or workflow to target the appropriate customers. Crucially, keeping survey state outside the Order removes the incompatible survey attribute from order XML consumed by the OMS, while retaining all data needed for the survey integration. B2C Commerce custom objects are intended for storing custom business data, and jobs support scheduled background processing: Custom Objects and Job Framework.

QUESTION NO: 9

A retailer uses product-level custom attributes for color, material, and collection. Merchandisers want shoppers to refine search results by these values, and the values must be updated by a nightly product feed.

Which two actions should the Architect include in the design?

Choose 2 answers

A. Define controlled product custom attributes with the correct data type.

B. Configure the attributes as searchable refinements and rebuild the search index.

C. Store all attribute values in a single comma-separated product string.

D. Create a separate storefront site for each color value.

E. Use customer-profile attributes to define product refinements.

ANSWER: A B

Explanation:

Defining controlled product custom attributes with the correct data type provides a consistent product-data model for the nightly feed. For example, a single-select enum can be used where a product has one permitted value, while a multi-value attribute should be used only when the business model allows multiple values. Controlled values help prevent inconsistent feed data from producing unusable refinement values.

Configuring the attributes as searchable refinements and rebuilding the search index is also required. Attribute definitions alone do not expose values in storefront search. The relevant search configuration must make each attribute available as a refinement, and index processing must occur after feed changes so the storefront reflects the updated product data. Salesforce documents product search configuration in the B2C Commerce search refinements documentation and catalog data in the B2C Commerce catalog data documentation.

QUESTION NO: 10

The Client has implemented a different category/search layout for mobile and desktop. The code uses a session attribute called deviceType to choose the corresponding layout. This attribute it populated from the browser user agent. After this implementation they have run into these problems:

â€¢ Sometimes desktop pages are being served to both desktop and mobile customers.

â€¢ Sometimes mobile pages ate being served to both desktop and mobile customers.

The page has caching implementedthat depends; on promotions. SEC is very important and the site traffic is high.

Which solution should the Architect select to resolve the issue without impacting the existing requirements?

A. Create customer groups for desktop and mobile users and uhremote includes based on these groups to render the mobile and desktop pages

B. Create customer groups for desktop and mobile users and empty promotions linked to these groups to ensure different cached versions of the page.

C. Disable caching for thesepages to ensure that the correct template is used to render the mobile and desktop pages.

D. Change the URL structure to include desktop and mobile as URL parameters to ensure different cached versions of the page

ANSWER: B

Explanation:

Create customer groups for desktop and mobile users and empty promotions linked to these groups to ensure different cached versions of the page. is the appropriate approach because the existing pages are already promotion-sensitive. B2C Commerce page caching does not automatically vary a cached response merely because a session attribute such as `deviceType` has a different value. As a result, the first rendered device-specific layout can be reused for requests that should receive the other layout.

Associating each device audience with a distinct customer group and activating a zero-discount, or otherwise non-visible, promotion for that group makes the applicable promotion context part of the cache variation. The platform can then maintain distinct cached page variants for the mobile and desktop audiences while retaining full-page caching. This preserves the high-traffic performance benefit of caching and continues to work with the site's existing promotion-dependent caching

design. It also keeps the public page URL unchanged, avoiding a device-specific URL scheme that could complicate canonicalization and search-engine indexing.

Salesforce's caching guidance explains that promotion-sensitive pages are cached according to applicable promotion context: B2C Commerce caching strategies. Customer-group behavior is documented in the CustomerGroup Script API.

QUESTION NO: 11

During load testing, a third party service isconstantly failing to respond in a timely manner on the Product Listing Page. The page is not affected at it is collecting data with the server side call, however the loading time b increasing.

Which two recommendations should the developer take in order to minimize the risk and Improve the loading time?

Choose 2 answers

A. Ask the third party to improve the reliability of the service.

B. Decrease the service timeout.

C. Enable the Circuit Breaker.

D. Remove the service.

E. Load the data asynchronously after the page is loaded

ANSWER: C E

Explanation:

Enable the Circuit Breaker. is appropriate because the B2C Commerce service framework protects the storefront when an external dependency repeatedly fails or is unavailable. Once the configured failure threshold is reached, the circuit breaker opens and prevents further calls for a defined interval. Requests can then fail fast instead of repeatedly waiting on an unhealthy remote system, which limits cascading performance impact and gives the third-party service time to recover. This behavior is especially valuable for a frequently requested page such as a Product Listing Page.

Load the data asynchronously after the page is loaded separates nonessential third-party enrichment from the critical server-rendered page response. The shopper can receive the primary product-listing content without waiting for the remote service. A subsequent asynchronous browser request can retrieve and display the supplemental information when it is available, improving perceived responsiveness while isolating intermittent third-party latency from the initial page render. The asynchronous endpoint should still use the service framework and handle unavailable data gracefully. Salesforce describes service configuration, timeouts, and circuit-breaker behavior in the B2C Commerce web services documentation; its storefront performance guidance is available in the B2C Commerce performance documentation.

QUESTION NO: 12

A Client uses an external fulfillment system that sends shipment updates to B2C Commerce. The fulfillment system can send the same shipment notification more than once if it does not receive a response before its own timeout.

What should the Architect recommend for the shipment-notification controller?

A. Make the shipment update idempotent by storing and validating the external shipment event identifier.

B. Reject every notification received after the first shipment is created for an order.

C. Store the external event identifier only in the shopper session.

D. Disable controller error handling so the fulfillment system retries indefinitely.

ANSWER: A

Explanation:

Make the shipment update idempotent by storing and validating the external shipment event identifier. is correct because duplicate callbacks are an expected condition in distributed integrations. The controller should identify each notification using a stable external event or shipment identifier and determine whether that event has already been successfully processed. If it has, the controller can return an appropriate successful response without creating duplicate shipment records or repeating order-state changes.

Idempotent processing is more reliable than assuming a callback will be delivered exactly once. The implementation should also authenticate the callback, validate its payload, and log correlation identifiers without exposing sensitive data. Salesforce provides order and shipment APIs for server-side order processing; see the Order Script API and the B2C Commerce security guidance.

QUESTION NO: 13

A company manages its regional operations as separate businesses. The regional sites (Site A and Site B) operate with:

• Separate realms

• Deferent code bates

• Different category navigation menus

• Frequent updates on category structure

The requirement from the business is to provide hreflang link tags on category pages pointing to the same category on the other regional site. Example MTML for one of these links as displayed on Site A is:

```
<link rel="alternate" href="https://www.siteB.com/en_US/womens-new-arrivals"
hreflang="en_US" />
```

Which solution should the Architect choose while keeping performance in mind?

A. Create a new custom attribute on the Category. Populate the attribute with the other entire site URLs corresponding to locales In JSON Format. Use the attribute to display the hreflang link tag.

B. Create a new custom object type Populate the hreflang mapping for each category and locale in this custom object. Use the custom object to display the hreflang link tag.

C. Create additional locales in al realms create a new custom attribute on the category that is localized. Populate the attribute with the other site URLs and use it to display the hreflang tag.

D. Create a custom Business Manager module. Ask the business to maintain the hreflang link tags for each regional site in this Business Manager module.

ANSWER: A

Explanation:

Create a new custom attribute on the Category. Populate the attribute with the other entire site URLs corresponding to locales In JSON Format. Use the attribute to display the hreflang link tag. This is the appropriate design because the sites are in separate realms, use independent code bases, and have different category hierarchies. Consequently, there is no dependable assumption that a category ID, URL rule, locale, or navigation path on one site can be derived from its counterpart on the other site. Storing the explicit cross-site URL mapping on the category makes the relationship clear and lets the page render the required canonical regional URL without a runtime lookup, cross-realm integration call, or traversal of another catalog structure.

A category is an extensible B2C Commerce object, so a custom attribute is a natural place to maintain metadata that belongs to that category. The value can hold a compact JSON map keyed by language/region, allowing templates to select and emit the relevant `rel="alternate" hreflang` link efficiently. Business users can update the mapping alongside category-structure changes in Business Manager, while storefront rendering remains a direct attribute read. Salesforce documents category extensibility in the Category Script API and custom-attribute access through ExtensibleObject.