

DUMPSQUEEN

Adobe Commerce Architect Master

Adobe AD0-E718

Version Demo

Total Demo Questions: 10

Total Premium Questions: 50

Buy Premium PDF

<https://dumpsqueen.com>

support@dumpsqueen.com

dumpsqueen.com

QUESTION NO: 1

An Adobe Commerce Architect notices that the product price index takes too long to execute. The store is configured with multiple websites and dozens of customer groups.

Which two ways can the Architect shorten the full price index execution time? (Choose two.)

- A. Enable price index customer group merging for products without tier prices
- B. Set Customer Share Customer Accounts Option to Global
- C. Edit customer groups to exclude websites that they are not using
- D. Set MaGE_INDEXER_THREADS_COUNT environment variable to enable parallel mode
- E. Move catalog price_index indexer to another custom indexer group

ANSWER: A D

Explanation:

The two best ways the Architect can shorten the full price index execution time are Option A. Enable price index customer group merging for products without tier prices, and Option D. Set MaGEINDEXER_THREADS_COUNT environment variable to enable parallel mode. Enabling customer group merging will help reduce the number of customer groups that need to be processed, while setting the environment variable will allow the indexer to use multiple threads and run in parallel mode, thus reducing the overall execution time.

Enabling price index customer group merging allows Magento to merge the price index rows for products that have the same price for all customer groups. This reduces the number of rows in the price index table and improves the performance of the indexer. Setting the MaGE_INDEXER_THREADS_COUNT environment variable allows Magento to run the indexer in parallel mode, which splits the index into multiple batches and processes them simultaneously. This reduces the execution time of the indexer. References: <https://devdocs.magento.com/guides/v2.4/extension-dev-guide/indexing.html#customer-group-merging> <https://devdocs.magento.com/guides/v2.4/extension-dev-guide/indexing.html#parallel-mode>

QUESTION NO: 2

A merchant notices that product price changes do not update on the storefront.

The index management page in the Adobe Commerce Admin Panel shows the following:

- All indexes are set to 'update by schedule'
- Their status is 'ready'
- There are no items in the backlog
- The indexes were last updated 1 minute ago

A developer verifies that updating and saving product prices adds the relevant product IDs into the catalog_product_price_cl changelog table.

Which two steps should the Architect recommend to the developer to resolve this issue? (Choose two.)

- A. Invalidate the catalog_product_price indexer in the Adobe Commerce Admin Panel so that it is fully reindexed next time the cron runs.
- B. Manually reindex the catalog_product_price index from the Command line:bin\magento indexer:reindex catalog_product_price.
- C. Make sure that no custom or third-party modules modify the changelog and indexing process.
- D. Make sure that the version_id for the price indexer in the mview_state table is not higher than the last entry for the same column in the changelog table and re-synchronize.
- E. Reduce the frequency of the cron job to 5 minutes so the items have more time to process.

ANSWER: A B

QUESTION NO: 3

An Adobe Commerce Architect needs to set up two websites on a single Adobe Commerce instance with base URLs: example.com and website2.example.com.

How should the Architect configure this project so that both websites can use the same customer base?

- A. Change Session Cookie attribute to "SameSite=None"
- B. Disable Session Validation for "HTTP_X_FORWARDED_FOR" header
- C. Set Cookie Domain for both websites to ".example.com"

ANSWER: C

Explanation:

By setting the same cookie domain for both websites, the customer base can be shared between both websites, as the customer will be authenticated by the same cookie across both sites. This will ensure that customers don't have to log in twice when switching between the two sites.

Setting Cookie Domain for both websites to ".example.com" will allow both websites to use the same customer base. This is because the cookie domain determines which websites can access the customer information stored in the cookie. By using a common domain, both websites can share the same customer cookie. See Multiple websites or stores in the Adobe Commerce Help Center¹. References: <https://experienceleague.adobe.com/docs/commerce-operations/configuration-guide/multi-sites/ms-overview.html?lang=en1>

QUESTION NO: 4

An Adobe Commerce Architect is supporting deployment and building tools for on-premises Adobe Commerce projects. The tool is executing build scripts on a centralized server and using an SSH connection to deploy to project servers.

A client reports that users cannot work with Admin Panel because the site breaks every time they change interface locale.

Considering maintainability, which solution should the Architect implement?

- A. Edit project env.php file, configure 'admin_locales_for.build' value, and specify all required locales
- B. Adjust the tool's build script and specify required locales during 'setup:static-content:deploy' command
- C. Modify project config.php file, configure 'admin_locales_for_deploy' value, and specify all required locales

ANSWER: A

Explanation:

Editing project env.php file, configuring 'admin_locales_for.build' value, and specifying all required locales is the solution that the Architect should implement. This is because this configuration allows the tool to generate static content for different locales during the build phase, which improves performance and avoids breaking the site when changing interface locale.

[See Deploy static view files in the Adobe Commerce Help Center](#)¹. References:

<https://experienceleague.adobe.com/docs/commerce-operations/configuration-guide/cli/static-view/static-view-file-deployment.html?lang=en1>

QUESTION NO: 5

An Adobe Commerce Architect needs to log the result of a ServiceClass : : ge-Dara method execution after all plugins have executed. The method is public, and there are a few plugins declared for this method. Among those plugins are after and around types, and all have sortOrder specified.

Which solution should be used to meet this requirement?

- A. Declare a new plugin with the sortOrder value higher than the highest declared plugin sortOrder and implement afterGetData method.
- B. Declare a new plugin with the sortOrder value lower than the lowest declared plugin sortOrder and implement aroundGetData method
- C. Declare a new plugin with the sortOrder value higher than the highest declared plugin sortOrder and implement aroundGetData method

ANSWER: A

Explanation:

To log the result of a method execution after all plugins have executed, the Architect should declare a new plugin with the sortOrder value higher than the highest declared plugin sortOrder and implement an after method. The after method receives the result of the original method and any previous plugins as a parameter and can log it accordingly. The around method is not suitable for this requirement because it executes before and after the original method and any other plugins, so it cannot capture the final result. References: <https://devdocs.magento.com/guides/v2.4/extension-dev-guide/plugins.html>

QUESTION NO: 6

An Architect wants to create an Integration Test that does the following:

- Adds a product using a data fixture
- Executes `$this->someLogic->execute($product)` on the product

- Checks if the result is true.

\$this->someLogic has the correct object assigned in the setup () method-Product creation and the tested logic must be executed in the context of two different store views with IDs of 3 and 4, which have been created and are available for the test.

How should the Architect meet these requirements?

- A.** Create one test class with one test method. Use the `\Magento\TestFramework\store\ExecutionStoreContext` class once in the fixture and another time in the test.
- B.** Create two test Classes With one test method each. Use the `@magentoExecuteInStoreContext 3` and `@magentoExecuteInStoreContext 4` annotations on the class level.
- C.** Create one test class with two test methods. Use the `@magentoStoreContext 3` annotation in one method and `@magentoStoreContext 4` in the other one.

ANSWER: C

Explanation:

The `@magentoStoreContext` annotation allows the test to run in the context of a specific store view. This annotation can be used on the method level to specify different store views for different test methods. This way, the product creation and the tested logic will be executed in the context of the same store view for each test method. References: <https://devdocs.magento.com/guides/v2.4/test/integration/annotations/magento-store-context.html>

QUESTION NO: 7

A merchant is using a unified website that supports native Adobe Commerce B2B and B2C with a single store view.

The merchant wants to show the B2B account features like negotiable quotes and credit limits in the header of the site on every page for the logged-in users who are part of a B2B company account.

Each B2B company has its own individual shared catalog and customer group, and many customer groups for non B2B customers change. The merchant requests that this should not be tied to customer groups.

Which two solutions should the Architect recommend considering public data and caching? (Choose two.)

- A.** Create a plugin that switches the theme when a user is part of a B2B company so the output can be modified accordingly in the alternate theme.
- B.** Check if the current user is part of a B2B company within a block class and modify the output accordingly.
- C.** Create a new custom condition for customer segments that allow for choosing whether a user is part of a B2B company and then use this segment to modify the output accordingly.
- D.** Set whether the current user is part of a B2B company in the customer session and use that data directly to modify the output accordingly.
- E.** Create a new HTTP Context variable to allow for separate public content to be cached for users in B2B companies where the output can be modified accordingly.

ANSWER: C E

Explanation:

C would involve creating a new custom condition for customer segments that allow for choosing if a user is part of a B2B company, and then use this segment to modify the output accordingly. E would involve creating a new HTTP Context variable to allow for separate public content to be cached for users in B2B companies, where the output can be modified accordingly.

To show the B2B account features in the header of the site on every page for the logged-in users who are part of a B2B company account, the Architect should recommend two solutions: C) Create a new custom condition for customer segments that allow for choosing whether a user is part of a B2B company and then use this segment to modify the output accordingly. This solution will allow the merchant to create a customer segment based on the custom condition and use it to display different content in the header for B2B users. E) Create a new HTTP Context variable to allow for separate public content to be cached for users in B2B companies where the output can be modified accordingly. This solution will ensure that the public content cache is varied based on the custom HTTP Context variable, which can be set based on whether the user is part of a B2B company or not. Option A is incorrect because switching the theme based on the user's B2B status is not a scalable or maintainable solution, and it will also affect the entire site's appearance, not just the header. Option B is incorrect because checking the user's B2B status within a block class will not work with public content cache, as it will not vary the cache based on that condition. Option D is incorrect because setting the user's B2B status in the customer session will not work with public content cache, as it will not vary the cache based on that data. References:

<https://devdocs.magento.com/guides/v2.4/extension-dev-guide/segmentation.html>

<https://devdocs.magento.com/guides/v2.4/extension-dev-guide/cache/page-caching/public-content.html>

QUESTION NO: 8

An external system integrates functionality of a product catalog search using Adobe Commerce GraphQL API. The Architect creates a new attribute `my_attribute` in the admin panel with frontend type select.

Later, the Architect sees that `ProductInterface` already has the field `my_atribute`, but returns an `mc` value. The Architect wants this field to be a new type that contains both option id and label.

To meet this requirement, an Adobe Commerce Architect creates a new module and file `etc/schema.graphqls` that declares as follows:

```
interface ProductInterface {
    my_attribute: SelectableOption @resolver(class:"Vendor\\CatalogGraphQL\\Model\\Resolver\\SelectableOption")
}
```

After calling command `setup:upgrade`, the introspection of `ProductInterface` field `xy_attribute` remains `int`. What prevented the value type of field `my_attribute` from changing?

- A.** The fields of `ProductInterface` are checked during processing `schema.graphqls` files. If they have a corresponding attribute, then the backend type of product attribute is set for field type.
- B.** The interface `ProductInterface` is already declared in `Magento.CatalogGraphQL` module. Extending requires use of the keyword `-xceni` before a new declaration of `ProductInterface`.
- C.** The `Magento.CatalogGraphQL` module occurs later in sequence than the `Magento.GraphQL` module and merging output of dynamic attributes schema reader overrides types declared in `schema.graphqls`

ANSWER: C

Explanation:

`products query` is a GraphQL query that returns information about products that match specified search criteria. It also shows how to use `ProductInterface` fields to retrieve product data.

<https://devdocs.magento.com/guides/v2.3/graphql/queries/products.html>

QUESTION NO: 9

An Adobe Commerce store owner sets up a custom customer attribute "my.attribute" (type int).

An Architect needs to display customer-specific content on the home page to Customers with "my.attribute" greater than 3. The website is running Full Page Cache.

Using best practices, which two steps should the Architect take to implement these requirements? (Choose two.)

- A. Use customer-data JS library to retrieve "my.attribute" value
- B. Add a new context value of "my.attribute" to Magento\Framework\App\Http\Context
- C. Add a custom block and a phtml template with the content to the cmsjindexindex.xml layout
- D. Create a Customer Segment and use "my.attribute" in the conditions
- E. Add a dynamic block with the content to the Home Page

ANSWER: B E

Explanation:

To display customer-specific content on the home page with Full Page Cache enabled, the Architect needs to add a new context value of "my.attribute" to Magento\Framework\App\Http\Context. This will allow the cache to vary based on the value of "my.attribute". Then, the Architect needs to add a dynamic block with the content to the Home Page. A dynamic block is a type of content block that can be personalized based on customer segments or other conditions. References:

<https://devdocs.magento.com/guides/v2.4/extension-dev-guide/cache/page-caching/public-content.html>
<https://docs.magento.com/user-guide/marketing/page-builder-add-content-block.html>

QUESTION NO: 10

A company has an Adobe Commerce store. An attribute named "my.attribute" (type "text") is created to save each product's global ID that is shared between multiple systems.

Several months after going live, the values of "my.attribute" are all integer. This causes a problem for the other systems when those systems receive this data.

An Adobe Commerce Architect needs to recommend a solution to change the type of "my.attribute" from text to int

Which two steps should the Architect take to achieve this? (Choose two.)

- A. Migrate data from table "catalog_product_entity_text" to "catalog_product_entity_int" for the attribute.id
- B. Go to Admin > Stores > Attributes > Product, edit "my.attribute" and update type from "text" to "int"
- C. Write a plugin for \Magento\Eav\Model\Entity\Attribute\Backend\AbstractBackend::afterLoad() and load data from "catalog_product_entity_int"
- D. Create a Data Patch and update 'my.attribute' type from "text" to "int"
- E. Run the Command bin/magento indexer:reset catalog_product_attribute

ANSWER: A D

Explanation:

[Option A is correct because it will migrate data from one table to another based on the attribute id, which is required when changing the attribute type¹⁴. Option D is correct because it will create a data patch that will update 'my.attribute' type from "text" to "int", which is a recommended way of changing the attribute type programmatically⁴.](#)

To change the type of "my.attribute" from text to int, the Architect should take two steps: A) Migrate data from table "catalog_product_entity_text" to "catalog_product_entity_int" for the attribute_id. This step will move the existing values of the attribute from the text table to the int table, which corresponds to the new type. D) Create a Data Patch and update 'my.attribute' type from "text" to "int". This step will update the attribute metadata in the database and reflect the new type. Option B is incorrect because the Admin panel does not allow changing the type of an existing attribute. Option C is incorrect because writing a plugin for \Magento\Eav\Model\Entity\Attribute\Backend\AbstractBackend::afterLoad() will not change the type of the attribute, but only load data from a different table. Option E is incorrect because running the command bin/magento indexer:reset catalog_product_attribute will not change the type of the attribute, but only reset the indexer status. References: <https://devdocs.magento.com/guides/v2.4/extension-dev-guide/attributes.html>