



Free Questions for [AD0-E716](#) by [certsinside](#)

Shared by [Robbins](#) on [29-01-2024](#)

For More Free Questions and Preparation Resources

[Check the Links on Last Page](#)

Question 1

Question Type: MultipleChoice

When checking the cron logs, an Adobe Commerce developer sees that the following job occurs daily: main.INFO: Cron Job inventory_cleanup_reservations is successfully finished. However, the inventory_reservation table in the database is not emptied. Why are there records remaining in the inventory_reservation table?

Options:

- A- Only reservations matching canceled orders are removed by the cron job.
- B- Only reservations no longer needed are removed by the cron job.
- C- The 'Auto Cleanup' feature from Multi Source Inventory was disabled in configuration.

Answer:

B

Explanation:

The reason why there are records remaining in the inventory_reservation table is that only reservations no longer needed are removed by the cron job. The inventory_reservation table tracks the quantity of each product in each order and creates a reservation for each

product when an order is placed, shipped, cancelled or refunded. The initial reservation has a negative quantity value and the subsequent reservations have positive values. When the order is complete, the sum of all reservations for the product is zero. The cron job removes only those reservations that have a zero sum from the table, leaving behind any reservations that are still needed for incomplete orders. Verified Reference: [Magento 2.4 DevDocs] [Magento Stack Exchange]

Question 2

Question Type: MultipleChoice

An international merchant is complaining that changes are taking too long to be reflected on the frontend after a full product import.

Thinking it may be database issues, the Adobe Commerce developer collects the following entity counts:

- * Categories: 900
- * Products: 300k
- * Customers: 700k
- * Customer groups : 106
- * Orders: 1600k
- * Invoices: 500k

* Creditmemos: 50k

* Websites : 15

* Stores : 45

What is a probable cause for this?

Options:

A- The combination of the number of products, categories and stores is too big. This leads to a huge amount of values being stored in the flat catalog indexes which are too large to be processed at a normal speed.

B- The combination of the number of orders, customers, invoices and creditmemos is too big. This leads to a huge amount of values being stored in the customer grid index which is too large to be processed at a normal speed.

C- The combination of the number of products, customer groups and websites is too big. This leads to a huge amount of values being stored in the price index which is too large to be processed at a normal speed.

Answer:

C

Explanation:

The probable cause for the delay in reflecting the changes on the frontend after a full product import is the combination of the number of products, customer groups and websites. This leads to a huge amount of values being stored in the price index which is too large to be processed at a normal speed. The price index calculates the final price of each product for each customer group and website, taking into account various factors such as tax, discounts, catalog price rules, etc. When there are many products, customer groups and websites, the price index becomes very complex and time-consuming to update. Verified Reference: [Magento 2.4 DevDocs] [Magento Stack Exchange]

Question 3

Question Type: MultipleChoice

When researching some issues with the indexer, an Adobe Commerce developer is seeing errors in the logs similar to Memory size allocated for the temporary table is more than 20% of innodb_buffer_pool_size. It is suggested that the client update innodb_buffer_pool_size or decrease the batch size value.

Why does decreasing the batch size value improve performance?

Options:

A- This decreases memory usage for the temporary table.

B- This allows for a longer timeout per batch process.

C- This allows for more PHP threads to be utilized during the process.

Answer:

A

Explanation:

Decreasing the batch size value improves performance by reducing the memory usage for the temporary table. The batch size value determines how many rows of data are processed at a time by the indexer. A large batch size value can cause the allocated memory size for the temporary table to exceed 20% of `innodb_buffer_pool_size`, which can result in errors and slow down the indexing process. By lowering the batch size value, the indexer can process the data more efficiently and avoid memory issues. Verified Reference: [Magento 2.4 DevDocs] [Magento Stack Exchange]

Question 4

Question Type: MultipleChoice

An Adobe Commerce developer wants to generate a list of products using `ProductRepositoryInterface` and search for products using a `supplier_id` filter for data that is stored in a standalone table (i.e., not in an EAV attribute).

Keeping maintainability in mind, how can the developer add the supplier ID to the search?

Options:

A- Write a before plugin on `\Magento\Catalog\Model\ProductRepository::getList()` and register the search criteria passed. Write an event observer to listen for the event `catalog_product_collection_load_before`. Iterate through the registered search criteria, and if found, apply the needed join and filter to the events collection.

B- Add a custom filter to the virtual type `\Magento\Catalog\Model\Api\SearchCriteria\CollectionProcessor\ProductFilterProcessor` for `supplier_id` field. In the custom filter, apply the needed join and filter to the passed `$collection`.

C- Write a before plugin on `\Magento\Framework\Api\SearchCriteria\CollectionProcessorInterface::process()`. Iterate through the `$searchCriteria` provided for `supplier_id`, and if found, apply the needed join and filter to the passed collection.

Answer:

B

Explanation:

The developer can add a custom filter to the virtual type `\Magento\Catalog\Model\Api\SearchCriteria\CollectionProcessor\ProductFilterProcessor` for `supplier_id` field. In the custom filter, the developer can apply the needed join and filter to the passed `$collection`. This is the recommended way to extend the search criteria for

products using dependency injection and plugins. Verified Reference: [Magento 2.4 DevDocs] [Magento Stack Exchange]

Question 5

Question Type: MultipleChoice

An Adobe Commerce developer is tasked to add a file field to a custom form in the administration panel, the field must accept only .PDF files with size less or equal than 2 MB. So far the developer has added the following code within the form component xml file, inside the fieldset node:

```
<field name="pdf_file" formElement="fileUploader">
  <formElements>
    <fileUploader>
      <settings>
        <uploaderConfig>
          <param xsi:type="string" name="url">myvendor_mymodule/customForm/uploadPdf</param>
        </uploaderConfig>
      </settings>
    </fileUploader>
  </formElements>
</field>
```

How would the developer implement the validations?

A)

Add the Validations Within the HyVendor\MyModule\Controller\Adminhtml\CustomEntity\UploadPdf Controller

```
public function execute()
{
    $file = $this->fileUploaderFactory->create($this->getRequest()->getPdfFile());
    if($file->getExtension() == 'pdf') {
        throw new InvalidFileException(__('The file must be PDF.));
    }
    if($file->getSize() >= '2048000') {
        throw new InvalidFileException(__('The file size must be less or equal than 2MB'));
    }

    return $this->resultFactory->create(ResultFactory::TYPE_PAGE);
}
```

B)

Add a virtual type for MyVendor\MyModule\Model\CustomPdfUploader specifying the allowedExtensions and the maxFileSize for the constructor, within the module's di.xml:

```
<type name="MyVendor\MyModule\Model\CustomPdfUploader">
    <arguments>
        <argument name="allowedExtensions" xsi:type="string">pdf</argument>
        <argument name="maxFileSize" xsi:type="number">2048000</argument>
    </arguments>
</type>
```

C)

Add the following code inside the node:

```
<allowedExtensions>pdf</allowedExtensions>
<maxFileSize>2048000</maxFileSize>
```

Options:

A- Option A

B- Option B

C- Option C

Answer:

B

Explanation:

The developer can add a virtual type for `Myvendor\MyModule\Model\customPdfuploader` specifying the `allowedExtensions` and the `maxFileSize` for the constructor, within the module's `di.xml`. This way, the developer can reuse the existing file uploader class and customize it for the specific field without modifying the core code. Verified Reference: [Magento 2.4 DevDocs] [Magento Stack Exchange]

Question 6

Question Type: MultipleChoice

An Adobe Commerce developer is tasked with adding a new export option for the order grid, they have added the following code for the export button within sales_order_grid.xml:

```
<exportButton>
  <settings>
    <options>
      <option name="txt" xsi:type="array">
        <item name="value" xsi:type="string">txt</item>
        <item name="label" xsi:type="string" translate="true">TXT</item>
        <item name="url" xsi:type="string">vendor_module/sales/export/customExport</item>
      </option>
    </options>
  </settings>
</exportButton>
```

Upon testing, they are getting redirected, what would be a cause for this error?

Options:

- A- The option's uri attribute is not valid.
- B- The layout cache needs to be refreshed.
- C- The developer has to add a formkey for the new export option.

Answer:

C

Explanation:

The developer has to add a formkey for the new export option because the formkey is required for security reasons. Without the formkey, the request will be rejected and redirected to the dashboard page. Verified Reference: [Magento 2.4 User Guide] [Magento 2.4 DevDocs]

Question 7

Question Type: MultipleChoice

An integration named Marketing is created on the Adobe Commerce instance. The integration has access on Magento_Customer::customer resources and the access token is xxxxxx.

How would the rest API be called to search the customers?

Options:

A- Using the integration access token as Bearer:

```
curl -X GET https://magentourl/rest/V1/customers/search?searchCriteria... -H 'Authorization: Bearer XXXXXX'
```

B- Passing integration name and access token as http auth credentials:

```
curl -X GET https ://Marketing:XXXXXX(SImagentourl/rest/V1/customers/search?5earchCriteria . . .
```

Using integration name as username and access token as password, get the admin token (yyyyyy) via:

```
curl -X POST https://magentourl/rest/V1/integration/admin/token -d '{"username':'Marketing', 'password':'XXXXXX'}' -H 'Content-
```

C- Type: application/json'

Use the admin token as Bearer

```
curl -X GET https://magentourl/rest/V1/customers/search?searchCriteria... -H 'Authorization: Bearer YYYYYY'
```

Answer:

B

Explanation:

According to the [Magento Stack Exchange](#) answer, UI components are used to render various elements on Magento admin pages, such as grids, forms, buttons, etc. UI components are defined in XML files that are located in the `view/adminhtml/ui_component` directory of each module. To add a custom logic to render a column in a grid, the developer should create a custom class extending `MagentoUiComponentListingColumnsColumn` and add the custom logic within the `prepareDataSource` method. This method receives an array of data sources and modifies them according to the column logic. The developer should also add an attribute class to the column node within the module's `customer_listing.xml` file and specify their custom class name as its value. Verified Reference: <https://magento.stackexchange.com/questions/317821/how-to-add-custom-logic-to-render-a-column-in-a-grid-in-magento-2>

Question 8

Question Type: MultipleChoice

An Adobe Commerce developer successfully added a new column to the customers grid. This column needs the data to be formatted before showing its content in the grid.

According to best practices, how would the developer add the custom logic to render the column?

Options:

- A-** 1. Create an after plugin for `Magento\Ui\Component\Listing\Columns\Column::prepareColumn()`.
2. Add the custom logic within the `afterPrepareColumn` method.
- B-** 1. Create a custom class extending `Magento\Ui\Component\Listing\Columns\Column`.
2. Add the custom logic within the `prepareDataSource` method.
3. Add an attribute class to the column node within the module's `customer_listing.xml`.
- C-** 1. Override the `Magento\Customer\Ui\Component\DataProvider Class` using a preference.
2. Override the `getData()` method and add the custom logic per row.

Answer:

A

Question 9

Question Type: MultipleChoice

An Adobe Commerce developer has created a module that adds a product attribute to all product types via a Data Patch-According to best practices, how would the developer ensure this product attribute is removed in the event that the module is uninstalled at a later date?

Options:

- A-** Add an Uninstall.php file extending `\Magento\Framework\Setup\UninstallInterface` to the module's Setup directory and implement the uninstall method.
- B-** Add instructions to the module's README.md file instructing merchants and developers that they must manually remove this attribute if they want to uninstall the module.
- C-** Make the Data Patch implement `\Magento\Framework\setup\Patch\PatchRevertableInterface` and implement the revert method to remove the product attribute.

Answer:

C

Explanation:

According to the Develop data and schema patches guide for Magento 2 developers, data patches can also implement PatchRevertableInterface to provide rollback functionality for their changes. The revert() method contains the instructions to undo the data modifications made by the patch. To ensure that the product attribute is removed when the module is uninstalled, the developer should make the data patch implement PatchRevertableInterface and implement the revert method to remove the product attribute using EavSetupFactory or AttributeRepositoryInterface. Verified Reference: <https://devdocs.magento.com/guides/v2.3/extension-dev-guide/declarative-schema/data-patches.html>

Question 10

Question Type: MultipleChoice

The developer is required to convert a modules database scripts from old install/upgrade setup files to a data patches format and does not want to apply database changes that were already done by install/upgrade scripts.

The current module version is 1.5.4.

What would be the recommended solution to skip changes that were already applied via old format (install/upgrade scripts)?

Options:

- A-** Implement Patchversioninterface and return 1.5.4 on the getVersion() method.
- B-** Inside apply() method, check for module version and run the code if version is less than 1.5.4.
- C-** This is not possible. A module cannot implement both data patch and install scripts.

Answer:

A

Explanation:

According to the [Develop data and schema patches guide for Magento 2 developers](https://devdocs.magento.com/guides/v2.3/extension-dev-guide/declarative-schema/data-patches.html), data patches are classes that contain data modification instructions. They are defined in a `<Vendor>/<Module_Name>/Setup/Patch/Data/<Patch_Name>.php` file and implement `MagentoFrameworkSetupPatchDataPatchInterface`. Data patches can also implement `Patchversioninterface` to specify the module version that the patch is associated with. The `getVersion()` method returns the module version as a string. To skip changes that were already applied via old format (install/upgrade scripts), the developer should implement `Patchversioninterface` and return 1.5.4 on the `getVersion()` method. This way, the data patch will only be applied if the module version is greater than or equal to 1.5.4. Verified Reference: <https://devdocs.magento.com/guides/v2.3/extension-dev-guide/declarative-schema/data-patches.html>

To Get Premium Files for AD0-E716 Visit

<https://www.p2pexams.com/products/ad0-e716>

For More Free Questions Visit

<https://www.p2pexams.com/adobe/pdf/ad0-e716>

