



Free Questions for [AD0-E717](#) by [actualtestdumps](#)

Shared by [Weber](#) on [24-05-2024](#)

For More Free Questions and Preparation Resources

[Check the Links on Last Page](#)

Question 1

Question Type: MultipleChoice

How should a record be removed from a database which is using a model that was inherited from the \Magento\Framework\Model\AbstractModel class?

Options:

- A- Call the 'unset' method on this model object
- B- Call the 'remove' method on this model object
- C- Call the 'delete' method on this model object

Answer:

C

Explanation:

The 'delete' method on the \Magento\Framework\Model\AbstractModel class is used to remove a record from the database. This method will also cascade the delete to any related records.

In Magento, models that inherit from the \Magento\Framework\Model\AbstractModel class interact with the database through the ORM (Object-Relational Mapping) layer. To remove a record from the database using such a model, the delete method is used. This method encapsulates the logic for deleting the record associated with the model instance from its corresponding database table. By calling \$model->delete(), where \$model is an instance of a model inheriting from AbstractModel, Magento will perform the necessary operations to remove the record from the database, ensuring data integrity and consistency within the application.

Question 2

Question Type: MultipleChoice

How are multiple EAV attributes belonging to the same entity grouped in the database?

Options:

- A- Based on the sizes of values they contain
- B- Based on all numeric values being stored in one table while text values are stored in the other
- C- Based on the types of values they contain

Answer:

C

Explanation:

Multiple EAV attributes belonging to the same entity are grouped in the database based on their data types, such as datetime, decimal, int, text, or varchar. For example, all attributes with datetime values are stored in one table, while all attributes with text values are stored in another table.

The sizes or numeric/text values of attributes do not determine how they are grouped in the database.

Verified Reference: [Adobe Commerce Developer Guide - EAV data model]

Magento's EAV (Entity-Attribute-Value) model organizes attributes based on their data types to optimize storage and retrieval. Attributes are grouped into different tables based on whether they store values of types such as integer, varchar (text), decimal, datetime, etc. This organization allows Magento to efficiently manage the diverse data types associated with products, customers, and other entities, ensuring data integrity and optimizing database performance by using appropriate indexing and storage mechanisms for each data type.

Question 3

Question Type: MultipleChoice

Which two attribute input types does Magento already have by default? (Choose two.)

Options:

- A- Multiple Select
- B- Text Field
- C- Geographic coordinate
- D- Numeric Field

Answer:

A, B

Explanation:

The two attribute input types that Adobe Commerce already has by default are Multiple Select and Text Field. Multiple Select allows the user to select multiple values from a list of options. Text Field allows the user to enter text in a single line.

The Geographic coordinate and Numeric Field input types do not exist in Adobe Commerce.

Verified Reference: [Adobe Commerce User Guide - Create a product attribute]

Magento, by default, provides various attribute input types to accommodate different data entry needs for product and customer attributes. The 'Text Field' input type allows for single-line text input, suitable for short, textual data such as names, titles, or any other brief information. The 'Multiple Select' input type enables the selection of multiple options from a predefined list, useful for attributes with multiple applicable values like colors, sizes, or features. These input types are part of Magento's flexible attribute system, allowing for customizable data entry fields that cater to diverse product and customer data requirements.

Question 4

Question Type: MultipleChoice

Which two attribute input types can be used for a date? (Choose two.)

Options:

- A- Timezone
- B- Schedule
- C- Date and Time
- D- Date

Answer:

C, D

Explanation:

The two attribute input types that can be used for a date are Date and Time and Date. These input types allow the user to select a date or a date and time from a calendar widget.

The Timezone and Schedule input types do not exist in Adobe Commerce.

Verified Reference: [Adobe Commerce User Guide - Create a product attribute]

In Magento, attribute input types define the type of data an attribute can hold and how it should be inputted or displayed in the UI. For dates, Magento provides specific input types to handle date-related data efficiently. The 'Date' input type is used for attributes that require only a date, without a time component, suitable for birthdays, anniversaries, or any date-specific information. The 'Date and Time' input type, on the other hand, includes both date and time components, ideal for events, promotions, or any scenario where the time of day is relevant. These input types ensure data consistency and provide a user-friendly interface for selecting dates and times.

Question 5

Question Type: MultipleChoice

A product has been added to the Adobe Commerce Store, and it contains a value for the custom product attribute. A merchant reports that the attribute value is not displayed in the Additional Information tab on the product detail page.

Which action will correct this problem?

Options:

- A- The attribute must be moved to the specific group in the attribute set
- B- The attribute property 'Use in Product Tab' must be set to 'yes'
- C- The attribute property 'Visible on Catalog Pages on Storefront' must be set to 'yes'.

Answer:

C

Explanation:

The 'Visible on Catalog Pages on Storefront' attribute property determines whether or not the attribute value is displayed in the Additional Information tab on the product detail page. If this property is set to 'no', the attribute value will not be displayed.

For a custom product attribute to be displayed in the Additional Information tab on the product detail page in Magento, it needs to be visible on the catalog pages on the storefront. This visibility is controlled by the attribute property 'Visible on Catalog Pages on Storefront'. When this property is set to 'yes', Magento includes the attribute in the Additional Information tab, making it visible to customers browsing the product. This setting ensures that only relevant and intended attributes are shown on the storefront, allowing for better product information management and customer experience.

Question 6

Question Type: MultipleChoice

A developer is working on a task that includes a custom controller creation. A controller should forward the request to a different action.

How can the developer complete this task?

Options:

- A-** Specify the forward action in the controllerforward.xml configuration file.
- B-** Implement a forwardToAction method in the controller, and return the action where the request should be forwarded.
- C-** Return the forward object with action as an argument in the object's forward method

Answer:

C

Explanation:

To forward the request to a different action, the developer can use the following code in the controller:

```
return $resultForward->forward('action');
```

where `$resultForward` is an instance of `\Magento\Framework\Controller\Result\ForwardInterface` and 'action' is the name of the action where the request should be forwarded.

There is no `controllerjorward.xml` configuration file or `forwardToAction` method in Adobe Commerce.

Verified Reference: [Adobe Commerce Developer Guide - Forward action result]

In Magento, to forward a request from one controller action to another, a developer can utilize the `forward` method available in the controller action class. This is achieved by returning a result from the action method that instructs Magento to forward the request to another action. The forward object is obtained by calling the `$this->resultForwardFactory->create()` method within the controller action. Then, the target action is specified by calling the `forward` method on this object with the action name as the argument, such as `$resultForward->forward('targetAction')`. This approach is consistent with Magento's emphasis on using result objects to control the flow of request processing within its MVC architecture.

Question 7

Question Type: MultipleChoice

Which CLI command should be used to determine that static content signing is enabled?

Options:

A- bin/magento config:show dev/static/status

B- bin/magento config:show dev/static/sign

C- bin/magento config:show dev/static/sign/status

Answer:

B

Explanation:

After a thorough search of the provided documents, I couldn't find a direct reference to the specific CLI command for determining if static content signing is enabled in Magento. However, the typical command for checking configuration settings in Magento is bin/magento config:show , where is the configuration path for the setting you wish to view. Based on Magento's configuration path patterns and the options provided, the most logical choice would be B. bin/magento config:show dev/static/sign, although this cannot be confirmed without further context or documentation.

Question 8

Question Type: MultipleChoice

Which Magento project directory is the recommended webroot for the web server?

Options:

A- Pub/

B- app/

C- bin/

Answer:

A

Explanation:

The Pub/ directory is the recommended webroot for the web server in Magento. This is because it contains all of the static content that is used by the Magento store, such as images, CSS, and JavaScript files.

The recommended webroot for the web server in a Magento project is the pub/ directory. This directory is used to access Magento static files and is the safer option for webroot as it minimizes security risks by not exposing the entire application structure to the web.

The pub/ directory in Magento is designed to be the root directory for web servers. This design choice is intended to enhance security and efficiency by serving static files directly from this location, thereby reducing the need for PHP processing for these files. By setting the webroot to pub/, it ensures that only the necessary files are publicly accessible, protecting the application's core code and structure from external access. This setup is a recommended best practice in Magento development for maintaining a secure and well-organized file system structure.

Question 9

Question Type: MultipleChoice

Which two actions will the developer need to take to translate strings added in JS files? (Choose two.)

Options:

- A- `define ([
'jquery',
'mage/translate'
], function ($, $t) {.`
- C- `$ trans(,<string>')`
- D- `$.mage._('<string>');`
- D- `translate('<string>');`

Answer:

A, D, D

Explanation:

To translate strings added in JavaScript files in Adobe Commerce, developers need to use the `mage/translate` RequireJS module along with the `$.mage.__('<string>')` function to mark strings for translation. This approach ensures that any text strings embedded within JavaScript code can be localized according to the store's current locale, providing a consistent and accessible user experience across different languages and regions. The `mage/translate` module and the `$.mage.__()` function work together to retrieve the corresponding translated strings from Magento's translation dictionaries, dynamically replacing the original text in the JavaScript code with the appropriate translations.

Question 10

Question Type: MultipleChoice

Which method type can be intercepted by plugins?

Options:

A- final

B- static

C- public

Answer:

C

Explanation:

In Magento, plugins (Interceptors) can only intercept public methods. This is because the plugin system relies on Magento's object manager to dynamically create proxy classes that can intercept method calls. Since private and final methods are not accessible from outside the class they are defined in, and static methods are not called on an object instance, these method types cannot be intercepted. This mechanism allows for the extension and customization of Magento's core behavior in a transparent and non-intrusive manner.

Question 11

Question Type: MultipleChoice

Which Adobe Commerce table stores all cron data?

Options:

A- schedule

B- cronjob

C- cron_schedule

Answer:

C

Explanation:

The Adobe Commerce table that stores all cron job data is cron_schedule. This table maintains records of all scheduled cron tasks, including their statuses, execution times, and any messages related to their execution. It plays a central role in Magento's scheduling system, allowing for the management and monitoring of background tasks that are essential for various system functions and integrations.

Question 12

Question Type: MultipleChoice

Which command can be used to display a full list of enabled and disabled Magento modules?

Options:

A- bin/magento module:all

B- bin/magento modulestatus

C- bin/magento module:show

Answer:

B

Explanation:

The command to display a full list of enabled and disabled Magento modules is bin/magento module:status. This command provides a comprehensive overview of all modules within the Magento instance, categorizing them into enabled and disabled modules. This information is crucial for debugging and development purposes, as it allows developers to quickly understand which components of Magento are active and which are not, facilitating troubleshooting and configuration adjustments.

To Get Premium Files for AD0-E717 Visit

<https://www.p2pexams.com/products/ad0-e717>

For More Free Questions Visit

<https://www.p2pexams.com/adobe/pdf/ad0-e717>

