Free Questions for AD0-E704 by dumpssheet

Shared by Roman on 29-01-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

You are troubleshooting an issue where a related product rule is not showing the product expected by a user. You run this command:

bin magento cache:clean target_rule

What impact does this have?

Options:

A- Products to Display calculation will be cleared from the Magento cache

B- HTML output of target rule blocks will be cleared from the Magento cache

C- Cached select statements will be cleared from the Magento cache

D- Cached select statements will be cleared from action_select column of the magento_targetrule table

Answer:

D

Question 2

A merchant has created a segment that applies to registered customers whose gender is female. You then migrate customers by copying data directly from one database to another. What two ways are used to assign migrated customers with a female gender to the segment?

Options:

A- They will be automatically assigned by a cron job at the scheduled time

B- Run bin Mgento index:reinde.\ customer_segment

C- Click on the Refresh Segment Data button in the Admin

D- Re-save the customer segment in the Admin

Answer:

B, C

Question 3

You are working on a project with custom code located in an observer MyCompany\MyModule\Observer\Custom on the catalog_product_load_before event. How do you prevent your custom observer from being executed on a staging preview?

Options:

A- By adding your observer to the bannedobserver parameter of Magento\staging\Model\Event\Manager using di. xml

B- By moving your observer to the staged_catalog_product_load_before event

C- By specifying an attribute excude_from_staging_previwe*="i" in the declaration of the observer in the events.xml file

D- No action is required, the catalog_product_ioad_before event is excluded from the execution on staging preview

Answer:

C

Question 4

Question Type: **MultipleChoice**

A merchant is asking you to organize a catalog of 9000 rings which come in six sizes, each with unique inventory. Additionally, the customers will have 1000 unique engravings to choose from for their ring. The engraving is made on the selected ring just before the order is shipped.

How do you organize such a catalog to achieve optimal store performance and ease of management?

Options:

A- 9000 configurable products which have 6x1000 variants each

B- 9000 configurable products which have 6 variants each with 1000 values for an engraving custom option

C- 9000 simple products with 6 values for size custom option and 1000 values for an engraving custom option

D- 9000x6 simple products with 1000 values for engraving custom option

Answer:

C

Question 5

Question Type: **MultipleChoice**

A Magento site is experiencing an issue where a fatal out of memory error occurs during a custom bulk catalog import process. Here is the code:

```
/** @var \Magento\Catalog\Model\ResourceModel\Product\Collection */
protected $collection;

public function import()
{
    foreach ($this->collection as $product) {
        //
    }
}
```

Options:

A- Call Magento\Framework\Data\Collection: setMemoryLimit to increase PHP's memory limit

B- Inject an instance Of Uagento\Catalog\model\ResourceModel\Product\IteratingCollectionFactory into your code instead

C- Use a \Magento\Framework\Model\ResourceModel\iterator to walk through the collection row-by-row

D- UseMagento\Model\Entity\Collection\AbstractCollection: :setPage to iterate through the collection in chunks

Answer:

B, C

Question 6

Question Type: MultipleChoice

You are starting a project and realized you need to clean the content of a container. How do you achieve this?

A- You create a view/layout/mycompany_mymodule_index. xml and add a referenceContainer. You will then remove the existing blocks one by one

B- YOU create a view/layout/mycompany_mymodule_index. xmlWith the referenceContainer and remove all the blocks by adding remove name='*' />

C- The layout merging logic will override a container if you reference the container and load your module after the original one using sequence in your module. xml

D- YOU create a view\layout override/mycoapany_mymodule_index. xml and add the new blocks to the referenceContainer to override the previous ones

Answer:

D

Question 7

Question Type: **MultipleChoice**

You are working on a Magento module for a merchant that does business in the UK. This module exposes a REST API endpoint for sending text messages to customers. There is an app based on this REST API which allows admin users to stay in touch with the customers. You would like to restrict the admin's ability to send text messages after work hours.

Keeping simplicity in mind, how do you implement this?

Options:

A- Use the built-in scheduled ACL functionality to configure allow-deny rules in the admin panel

B- Add the check directly to the service method implementation

C- Implement a cron job to change the admin permissions twice a day for everyone who has these permissions set

D- Add aclResolver='Mycompany\MyModule\Model\PathToAc1Resolver' declaration for the relevant methods in webapi. xbl

Answer:

D

Question 8

Question Type: **MultipleChoice**

You are developing an in-house solution for catalog management which is implemented as a Magento module called MyCompany_MyModule. To streamline integration with the ERP system you decided to expose MyCompany_MyModule'S functionality via REST.

What are the three minimum requirements to achieve this?

Options:

A- Add webapi. xml declarations for all methods to be exposed

B- Specify the parameter and return types in the PHP method signatures

C- Represent all functionalities which need to be exposed as service contracts

D- Specify the parameter and return types in the method annotations

E- Add Cap: annotations to all methods which need to be exposed

Answer:

A, C, E

Question 9

Question Type: MultipleChoice

You are programmatically creating a customer EAV attribute and want to add it to the customer segment's condition options. How do you do this?

A)

Add in customer_segment.xml

```
<customer_segment>
    <argument name="customer_segment" xsi:type="array">
        <item name="attribute" xsi:type="string">YOUR_ATTRIBUTE</item>
    </argument>
</customer_segment>
```

B)

Add to the attribute property array

```
'options' => [
    'option' => [
        [
            'customer_segment' => true
        ]
    ]
]
```

C)

Add the new attribute to the Customer_Segment attribute set

D)

Set the attribute property is_used_for_customer_segment to 1

Options:

A- Option A

B- Option B

C- Option C

D- Option D

Answer:

B