Question 1

You are debugging a problem with a shopping cart price rule which gives free shipping for the whole cart if the subtotal is greater than $100. You are seeing that when a custom shipping method is selected, the shipping price is still present even though the subtotal is greater than $100.

How do you fix this problem?

Options:

A- Create an option allow_freeshiping in the system configuration of the shipping method and set its value to 1

B- Add the shipping method's code to the price rule action

C- Modify the carrier class to process the free_shipping flag of the quote items

D- Add the shipping method's code to the price rule condition

Answer:

A

Question 2

While debugging a problem with a scheduled product update, you want to get a list of products that are scheduled for the next update. How does Magento Staging store scheduled entities?

Options:

A- Scheduled entities are stored in a copy of the entity table with a _ version suffix

B- Scheduled entities are stored in their tables as separate rows with version stored in the created_in updated_in fields

C- In the table staging_update as a serialized array of ids in the field entity_ids

D- In the table sTaging_scheduied_entity where each scheduled entity is represented by a row related to the staging_update table

Answer:
D

Question 3

A merchant is planning to create a single scheduled update for one million products. They are wondering about website performance. What two performance issues will the update cause?

Question 4

Question Type: MultipleChoice

After the installation of several 3rd-party extensions to the merchant's store, it was noticed the admin panel authentication consistently fails with the credentials used before. Where do you start debugging to understand why the authentication is failing?

Options:

A- \Magento\Backend\App\Action\Composite\Authentication::dispatch

B- \Magento\Backend\App\Act ion\Observer\Authentication::execute

C- \Magento\Backend\App\Act ion\Plugin\Authentication::aroundDispatch

D- \Magento\Backend\App\Act ion\AuthenticationWrapper::dispatch

Answer:

D

Question 5

Question Type: MultipleChoice

Magento regenerates the admin URL secret key for each new session. What is the purpose of this key?

Options:

A- To prevent brute-force attacks

B- To prevent admin pages from being cached

C- To validate ACL permissions for the current admin user

D- To prevent admin panel Cross Site Request Forgery attacks

Answer:

C

Question 6

Question Type: MultipleChoice

You have a requirement to customize the Product Detail Page so users can enter their ID in a custom field and get a personalized price. The user has to see the lower of the personalized and the Magento price.

Which two actions do you take to enable the personalized price on the Product Detail Page and the checkout?

Options:

A- Add a price class to the prices parameter of the Magento\Catalog\Pricing\Price\Pool class and implement the getvalue method

B- Create an observer for the event catalog_product_load_after, access the $product object and set a new price using the setFinalPrice method

C- Create a plugin for the Magento\Catalog\Model\Product::getFinalPrice method

D- Modify the indexer so all possible prices are listed in the price index table

Answer:

B, C

Question 7

You are developing a new payment method. It is required that any new order created with this payment method needs to have a specific custom status. Assume the status has been added to the system already, assigned to the processing state, and the new order state for the payment method is processing. How do you set the custom status on orders placed with the new payment method?

Options:

A- Add a custom status code to the Magento\Sales\Model\Order-\Payment class as a $newOrderStatus constructor parameter using di.xml

B- Create a system configuration option order.status and set the status code as its value

C- Create an observer for the event sales_order_set_status, get the $status parameter and set the new status using $status->setStatus()

D- Create a command setOrderStatus and add it to a commandPool with a name set_status

Answer:

C

Question 8

Question Type: MultipleChoice

Your company asks you to create the option to disable RMA for the customer group named CustomerGroup1. How do you do it?

A) It will require a code customization

B) This feature already exists and can be set in:

Admin > Store > Settings > Configuration > Sales > Rma Settings > Exclude Customer Group

C)

Create rma.xml and add

```
<arguments>
    <argument name="banned" xsi:type="array">
        <item name="CustomerGroup" xsi:type="string">CustomerGroup1</item>
    </argument>
</arguments>
```

D)

In `di.xml`

```xml
<type name="Magento\Rma\Model\CustomerGroup">
    <arguments>
        <argument name="banned" xsi:type="array">
            <item name="CustomerGroup" xsi:type="string">CustomerGroup1</item>
        </argument>
    </arguments>
</type>
```

Options:

A- Option

B- Option

C- Option

D- Option

Answer:

B

Question 9

Question Type: MultipleChoice

You have created a module MyCompany_MyModule that adds a custom category attribute. The attribute is not showing up on the Category Edit Page in the Admin. How do you correct this?

Options:

A- Define the field in MyCompany MyModule/view/admintml\ui_component\category_form xml

B- Call 3attribute->setIsSystem(0); in data patch after creating the attribute

C- Call Sattribute-setIsVisible(1): in data patch after creating the attribute

D- Define the field in MyCompany MyModule\view\adminhtml\layout\catalog_category_edit.xml

Answer:

D

Question 10

Question Type: **MultipleChoice**

You are working on a custom page where the content of a CMS block has to be rendered. You know the merchant is using Staging for CMS blocks. In your code you have access to the $repository object which is an instance of \Magento\Cms\Api\BlockRepositoryInterface. $blockId which is the original block ID, and $current Version which is the current Staging

version.

How do you load the CMS block with the right Staging version?

A)

```
$repository->getByIdAndVersion($blockId, $versionId);
```

B)

```
$repository->getById($blockId, $currentVersion);
```

C)

```
$repository->getById($blockId);
```

D)

```
/** @var \Magento\Staging\Helper\Entity $entityHelper */
$rowId = $entityHelper->getIdentifier('cms_block', $blockId, $currentVersion);
$repository->getById($rowId);
```

Options:

A- Option A

B- Option B

C- Option C

D- Option D

Answer:

D

Question 11

Question Type: MultipleChoice

A client uses an external PIM for managing product information. They want to run a cron job every minute to update and create products.
The site is using the "Update by schedule" index mode and the full page cache is enabled.

What is required to make these updates visible on the frontend?

Options:

A- The indexer_update_all_views cron will reindex the products and clean only the necessary entries from cache

B- The indxer_update_all_views cron will reindex the products, however the full page cache will need to be manually flushed

C- The indexer_update_all_views cron will reindex the products and clean the entire full page cache

D- The cron job will also need to run a reindex and flush full page cache on each run for updates to be reflected on the front end of the website

Answer:

D

Question 12

You are reviewing a custom observer for the event sales_order_save_after which is fired when an order is placed on the storefront. The observer is sending a remote request to an analytics system with the order data.

What issue may this observer cause?

Options:

A- In case of an error in the observer the order will not be saved in the database

B- The payment status is not available at the moment when the order is saved

C- The final price for the order is not available on the sales_order_save_after event

D- This event will not be fired if the order is placed through the WebAPI

Answer:

A