Question 1

Question Type: MultipleChoice

An Adobe Commerce Developer has written an importer and exporter for a custom entity. The client is using this to modify the exported data and then re-importing the file to batch update the entities.

There is a text attribute, which contains information related to imagery in JSON form, media_gallery. This is not a field that the client wants to change, but the software they are using to edit the exported data seems to be modifying it and not allowing it to import correctly.

How would the developer prevent this?

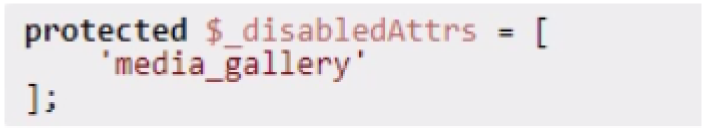
A) Specify a serializer class for the attribute using the $_transformAttrs class property array for both the exporter and importer so it gets converted:

```
protected $_transformAttrs = [
    'media_gallery' => \Magento\Framework\Serialize\Serializer\Json::class
];
```

B) Strip the attribute from the imported file by adding it to the s_strippedAttrs class property array:

```
protected $_strippedAttrs = [
    'media_gallery'
];
```

C) Prevent it from being exported by adding it to the $_disat>iedAttrs class property array:



```
protected $_disabledAttrs = [
    'media_gallery'
];
```

Options:

A- Option A

B- Option B

C- Option C

Answer:

A

Explanation:

The _transformAttrs class property array of the importer and exporter classes can be used to specify a serializer class for a particular attribute. The serializer class will be used to convert the attribute value from one format to another when the data is exported or imported.

In this case, the developer can specify a serializer class that will convert the JSON data in the media_gallery attribute to a string. This will prevent the software that the client is using to modify the exported data from changing the JSON data.

The following code shows how to specify a serializer class for the media_gallery attribute:

PHP

```
class MySerializer

{

public function serialize($value)

{

return json_encode($value);

}

public function deserialize($value)

{

return json_decode($value);

}

}

$importer->setSerializer('media_gallery', MySerializer::class);

$exporter->setSerializer('media_gallery', MySerializer::class);
```

Once the serializer class has been specified, the JSON data in the media_gallery attribute will be converted to a string when the data is exported or imported. This will prevent the software that the client is using to modify the exported data from changing the JSON data.

Question 2

Question Type: MultipleChoice

An Adobe Commerce developer wants to create a product EAV attribute programmatically which should appear as WYSIWYG in the admin panel. They have made sure that wysiwyg_enabled has been set to true, however, the attribute is not appearing as WYSIWYG in the admin panel.

What would be a possible reason?

Options:

A- The is_html_allowed_on_front Option iS Set tO false.

B- The input type is not set to text.

C- The input type is not set to textarea.

Answer:

C

Explanation:

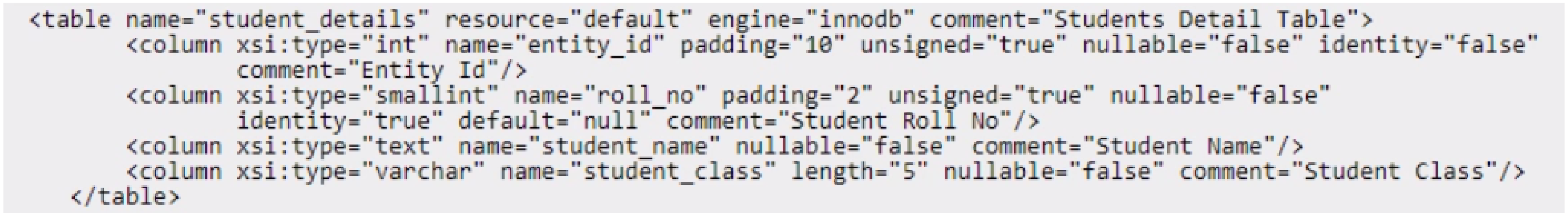
The input_type attribute of a product EAV attribute specifies the type of input field that will be used to enter the value of the attribute in the admin panel. The textarea input type is used for WYSIWYG fields. If the input_type attribute is not set to textarea, then the attribute will not appear as WYSIWYG in the admin panel.

To fix this, the developer should set the input_type attribute to textarea.

Question 3

Question Type: MultipleChoice

An Adobe Commerce developer is trying to create a custom table using declarative schema, but is unable to do so.



```
<table name="student_details" resource="default" engine="innodb" comment="Students Detail Table">
       <column xsi:type="int" name="entity_id" padding="10" unsigned="true" nullable="false" identity="false"
               comment="Entity Id"/>
       <column xsi:type="smallint" name="roll_no" padding="2" unsigned="true" nullable="false"
               identity="true" default="null" comment="Student Roll No"/>
       <column xsi:type="text" name="student_name" nullable="false" comment="Student Name"/>
       <column xsi:type="varchar" name="student_class" length="5" nullable="false" comment="Student Class"/>
   </table>
```

What are two errors in the snippet above? (Choose two.)

Options:

A- Column (roll_no) does not have index. It is needed since attribute identity is set to true.

B- Column (entity_id) does not have index. It is needed since attribute identity is set to false.

C- Column (student_name) does not have attribute length.

D- null is not a valid value for column (roll_no).

Answer:

A, C

Explanation:

The correct answers are A and C.

The errors in the snippet are:

Columnroll_nodoes not have an index. It is needed sinceattribute_identityis set totrue.

Columnstudent_namedoes not have an attribute length.

The attribute_identity attribute specifies whether the primary key of the table should be auto-incremented. If attribute_identity is set to true, then the roll_no column must have an index. The student_name column does not have an attribute length, which is required for string columns.

The following code shows how to fix the errors:

XML

```
<table name='vendor_module_table'>

<entity_id>

<type>int</type>

<identity>true</identity>

<unsigned>true</unsigned>

<nullable>false</nullable>

</entity_id>

<roll_no>

<type>int</type>

<identity>false</identity>

<unsigned>true</unsigned>

<nullable>false</nullable>

true
```

<index>true</index>

</roll_no>

<student_name>

<type>string</type>

<length>255</length>

<nullable>false</nullable>

</student_name>

</table>

Once the errors have been fixed, the table can be created successfully.

Question 4

Question Type: **MultipleChoice**

An Adobe Commerce developer is creating a module (Vendor.ModuleName) to be sold on the Marketplace. The new module creates a database table using declarative schema and now the developer needs to make sure the table is removed when the module is disabled.

What must the developer do to accomplish this?

Options:

A- There is nothing further the developer needs to do. The table will be removed when the module is disabled and bin/magento setup:upgrade is run.

B- There is nothing further the developer needs to do. The table will be removed when the when bin/magento module:uninstall vendor_ModuleName is run.

C- Add a schema patch that implements Magento\Framework\setup\Patch\PatchRevertabieinterface and drops the table in the revert function.

Answer:

C

Explanation:

According to the Declarative Schema Overview guide for Magento 2 developers, declarative schema is a new feature that allows developers to declare the final desired state of the database and has the system adjust to it automatically, without performing redundant operations. However, declarative schema does not support uninstalling modules or reverting changes. To remove a table when a module is disabled, the developer needs to add a schema patch that implements Magento\Framework\setup\Patch\PatchRevertabieinterface and drops the table in the revert function. The revert function will be executed when the module is disabled using bin/magento module:disable command. Verified Reference:

https://devdocs.magento.com/guides/v2.3/extension-dev-guide/declarative-schema/

Question 5

Question Type: MultipleChoice

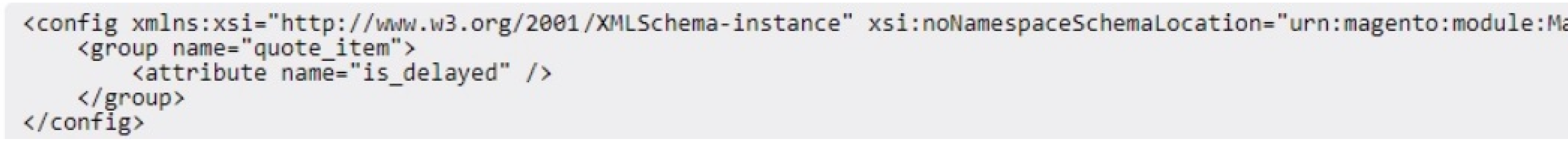
There is the task to create a custom product attribute that controls the display of a message below the product title on the cart page, in order to identify products that might be delivered late.

The new EAV attribute is_delayed has been created as a boolean and is working correctly in the admin panel and product page.

What would be the next implementation to allow the is_delayed EAV attribute to be used in the .phtml cart page such as $block->getProduct()->getIsDelayed()?

A)

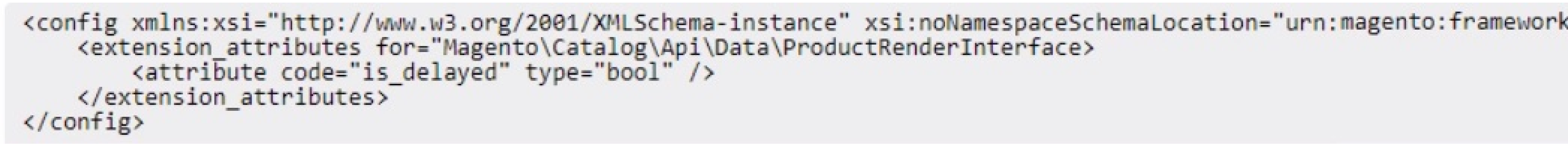
Create a new file etc/catalog_attributes.xmi:



```
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:module:Ma
    <group name="quote_item">
        <attribute name="is_delayed" />
    </group>
</config>
```

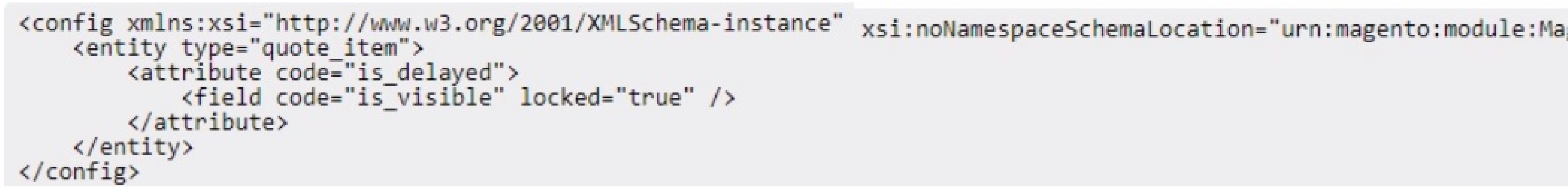
B)

Create a new file etc/extension attributes.xmi:



```
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:framework
    <extension_attributes for="Magento\Catalog\Api\Data\ProductRenderInterface>
        <attribute code="is_delayed" type="bool" />
    </extension_attributes>
</config>
```

Create a new file etc/eav attributes.xmi:



```
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:module:Ma
    <entity type="quote_item">
        <attribute code="is_delayed">
            <field code="is_visible" locked="true" />
        </attribute>
    </entity>
</config>
```

Options:

A- Option A

B- Option B

C- Option C

Answer:

A

Explanation:

To allow the is_delayed EAV attribute to be used in the .phtml cart page, the developer needs to create a new file called etc/catalog_attributes.xmi. This file will contain the definition of the is_delayed attribute.

The following code shows how to create the etc/catalog_attributes.xmi file:

XML

```
<?xml version='1.0'?>

<catalog_attributes>


<label>Is Delayed</label>

<note>This attribute indicates whether the product is delayed.</note>

<sort_order>10</sort_order>

<required>false</required>


</catalog_attributes>
```

Once the etc/catalog_attributes.xmi file has been created, the is_delayed attribute will be available in the .phtml cart page. The attribute can be accessed using the getIsDelayed() method of the Product class.

PHP

$product = $block->getProduct();

$isDelayed = $product->getIsDelayed();

The isDelayed variable will contain the value of the is_delayed attribute. If the value of the attribute is 1, then the product is delayed. If the value of the attribute is 0, then the product is not delayed.

Question 6

Question Type: MultipleChoice

An Adobe Commerce developer is tasked with creating a custom block that will be displayed on every page in the footer of the site.

After completing and optimizing the development, the developer notices that the block takes too much time to be generated on each page and decides to store it in the system cache after enabling it for all cache groups.

What would be the minimum requirement to achieve this?

Options:

A- Set a value for the cache_Lifetime data property of the block.

B- Set a value for cache_key data property of the block.

C- Set values for both cache_lifetime and cache_key data properties of the block.

Answer:

C

Explanation:

To store a block in the system cache, the developer needs to set values for both the cache_lifetime and cache_key data properties of the block. The cache_lifetime property specifies how long the block should be cached, and the cache_key property specifies a unique identifier for the block.

The following code shows how to set the cache_lifetime and cache_key data properties of a block:

PHP

```
$block->setData('cache_lifetime', 600);

$block->setData('cache_key', 'my_custom_block');
```

Once the cache_lifetime and cache_key data properties have been set, the block will be stored in the system cache and will not be regenerated on each page load.

Question 7

Question Type: **MultipleChoice**

An Adobe Commerce developer has added a new configuration field to the admin are

a. The path for this option is general/store_information/out_of_hours_phone.

Keeping simplicity in mind, how would the developer ensure this option contains a valid US telephone number?

Options:

A- Add <validate>phoneUS</validate> to the field in system.xml.

B- Create a backend model to check the validity of the phone number entered.

C- Add <validate type='phoneUS'/> to the field in system.xml.

Answer:

A

Explanation:

According to the Magento Stack Exchange answer, system.xml is a file that defines the configuration fields for the admin area. Each field can have a validate attribute that specifies a validation rule for the field value. Magento provides some built-in validation rules, such as phoneUS, which validates a US telephone number. Therefore, to ensure that the option contains a valid US telephone number, the developer needs to add <validate>phoneUS</validate> to the field in system.xml. Verified Reference: https://magento.stackexchange.com/questions/104570/magento-2-system-xml-validation-rules

20%
DISCOUNT