DUMPSBOSS.

Salesforce Certified Industries CPQ Developer

Salesforce Industries-CPQ-Developer

Version Demo

Total Demo Questions: 48

Total Premium Questions: 485

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, Product Catalog	121
Topic 2, Product Configuration	158
Topic 3, Pricing	75
Topic 4, Promotions	45
Topic 5, Digital Commerce	56
Topic 6, Integration	24
Topic 7, Mix Questions	6
Total	485

QUESTION NO: 1

Which two aspects of a promotion are determined by its time policy?

Choose 2 answers

A. The duration

B. The start date

C. The end date

D. The purchase date

ANSWER: A D

Explanation:

In Salesforce Industries CPQ, a promotion's time policy governs both the period for which the promotion remains applicable and the point in the customer lifecycle when it can be purchased. Therefore, **The duration** is determined by the time policy: the policy specifies how long the promotional offering, discount, or benefit applies after activation. **The purchase date** is also determined by the policy because it defines the permitted purchase or availability window for the promotion. Together, these settings let an implementation support common commercial rules, such as allowing a promotion to be purchased only during a campaign period while granting its discounted price for a defined number of months. This separates the promotion's commercial timing rules from the underlying product configuration and pricing setup. Salesforce Industries CPQ promotion configuration is part of the broader catalog and offer-management capabilities used to control how products and promotions are made available to customers. See Salesforce's Industries CPQ promotions documentation and the Salesforce Industries CPQ foundations learning module for related catalog and promotion concepts.

QUESTION NO: 2

The options for pricing bundles include: (Choose TWO)

Note: This question displayed answer options in random order when taking this Test.

A. Assigning the parent a zero price

B. Assigning a price to the parent and zero prices to the children

C. Assigning only pricing elements to the parent and children

D. Assigning only pricing variables to the parent and children

ANSWER: A B

Explanation:

"Assigning the parent a zero price" is a supported bundle-pricing approach because the bundle parent can serve as the commercial and configuration container while the individual child products carry the charges. This model is useful when the customer must select or configure components independently and the resulting price should be the sum of those component prices. The parent still provides the bundle structure, eligibility context, and relationship among the child products even when it has no direct charge.

"Assigning a price to the parent and zero prices to the children" is also supported. In this approach, the parent represents the sellable commercial offer and holds the bundle's price, while included children are zero-priced components. It is appropriate for a packaged offer sold at one price, where the children must be visible for configuration, fulfillment, decomposition, or technical purposes but are not separately charged. Industries CPQ supports defining product relationships and pricing behavior so a bundle can be modeled according to the intended commercial offer and downstream order requirements. See Salesforce's Salesforce Industries developer documentation and the Salesforce CPQ product overview for product-catalog and CPQ concepts.

QUESTION NO: 3

Which OmniScript actions can invoke server-side processing? (Choose THREE)

Note: This question displayed answer options in random order when taking this Test.

A. Integration Procedure Action

B. DataRaptor Extract Action

C. Remote Action

D. Text Block

E. Step

ANSWER: A B C

Explanation:

Integration Procedure Action, **DataRaptor Extract Action**, and **Remote Action** can invoke server-side processing from an OmniScript. An Integration Procedure Action calls an Integration Procedure that can orchestrate multiple server-side operations. A DataRaptor Extract Action retrieves Salesforce data through a configured DataRaptor. A Remote Action invokes an Apex remote method when custom server-side logic is required.

A Text Block and a Step are presentation and navigation elements. They organize the guided interaction but do not by themselves call server-side services. See Salesforce Help: OmniScript and Salesforce Trailhead: OmniScripts.

QUESTION NO: 4

Which customer lifecycle actions are supported by asset-based ordering in Salesforce Industries CPQ? (Choose FOUR)

Note: This question displayed answer options in random order when taking this Test.

A. Adding a product or service

B. Changing an existing product or service

C. Moving a service

D. Disconnecting a service

E. Creating a new Salesforce user

ANSWER: A B C D

Explanation:

Asset-based ordering supports lifecycle transactions against products and services that a customer already owns. It can support **adding a product or service**, **changing an existing product or service**, **moving a service**, and **disconnecting a service**. These actions use the existing asset context to identify what the customer has and to calculate the changes required for the transaction.

Asset-based ordering is particularly important for communications and subscription businesses because customer requests frequently amend an existing service rather than create a completely unrelated order. Creating a new Salesforce user is an administrative setup task and is not an asset lifecycle action. See Trailhead: Industries CPQ Foundations and Salesforce Help: Assets Overview.

QUESTION NO: 5

Which four of the following would require compatibility rules?

Note: This question displayed answer options in random order when taking this Test.

A. Apple iPhone requires SIM card

B. MetroE cannot be delivered to Consumer customers

C. VPN Access can only be delivered over high speed internet

D. Only DSL customers can purchase a modem

E. High Def set top box is required for High Def TV packages

ANSWER: A C D E

Explanation:

Compatibility rules in Industries CPQ define valid relationships among products and enforce configuration dependencies during cart-based product selection. They are appropriate when selecting one offer, product, or service requires, permits, excludes, or conditions the selection of another product. “Apple iPhone requires SIM card” is a required-product dependency: configuring the phone must result in a compatible SIM card being included. “VPN Access can only be delivered over high speed internet” is a dependency between the VPN service and its required underlying connectivity product. “Only DSL customers can purchase a modem” expresses a compatibility condition between the modem and the DSL service context; the modem is valid only when the qualifying DSL product is present. “High Def set top box is required for High Def TV packages” is likewise a required-product relationship between the television package and the necessary hardware. These conditions are modeled so the cart can guide the user to valid combinations and prevent an incomplete or unsupported configuration before order submission. Salesforce Industries CPQ documentation describes using product relationships and rules to control compatible product selections and configuration behavior. See Salesforce Help: Compatibility Rules and Salesforce Developers: Product Relationships.

QUESTION NO: 6

What must you do before you can create a product attribute?

A. Create the product

B. Create the object type

C. Create an attribute category

D. Create the picklist

ANSWER: C

Explanation:

Create an attribute category is correct because Salesforce Industries CPQ uses attribute categories to organize and manage product attributes. An attribute category is established first and then selected as the grouping context when defining an attribute. This setup gives attributes a consistent business classification, such as specifications, service characteristics, or eligibility-related details, and supports organized presentation and maintenance in the product catalog.

Attributes are reusable definitions of product characteristics that can later be assigned to products and configured with details such as data type, values, and validation behavior. Defining the category first establishes the catalog structure into which those reusable definitions belong. This is particularly important in Industries CPQ implementations, where a large catalog can contain many attributes shared across offers and products; categories make those attributes easier for catalog administrators to locate, govern, and present coherently during product configuration.

Salesforce’s Industries CPQ learning materials describe product attributes as catalog metadata organized through attribute categories and then associated with products. See the Salesforce Industries CPQ Foundations module and the Salesforce Help documentation for product attributes.

QUESTION NO: 7

Product attributes are stored:

Note: Thisquestion displayed answer options in random order when taking this Test.

A. in JSON format in a field on the Product2 object

B. As name/value pairs on the Attribute2 object

C. In individual fields on the product record

D. As a binary attachment

ANSWER: B

Explanation:

As name/value pairs on the Attribute2 object is correct because Salesforce Industries CPQ uses the Attribute2 data model to represent product attributes in a flexible, metadata-driven way. Rather than requiring a fixed field on every product for every possible characteristic, an attribute is represented by a name and its associated value. This lets a product catalog support configurable information such as capacity, color, speed, eligibility criteria, or other industry-specific characteristics without altering the core product data structure whenever the catalog changes.

Attribute definitions and their values can be used by the Industries CPQ configuration and pricing framework when determining which product choices are available, applying rules, and presenting configurable product information to users. This model is particularly useful for complex catalogs because attributes can be reused and managed independently of the standard product record structure. Salesforce's Industries data-model documentation describes the catalog and product data architecture used across Communications Cloud and Industries implementations, while Salesforce's Industries CPQ resources describe the metadata-driven catalog approach. See the Salesforce Communications Cloud data model reference and Salesforce Industries CPQ overview.

QUESTION NO: 8

What does the HTML template control inthe Guided Selling cart?

Note: This question displayed answer options in random order when taking this Test.

A. JSON for the price list Id

B. The order status

C. Product filters to retrieve specific products

D. The vertical or horizontal appearance of thecart

E. Remote methods to be used for the cart

ANSWER: D

Explanation:

The vertical or horizontal appearance of thecart is correct because the Guided Selling cart's HTML template determines the cart's presentation layout in the user interface. In particular, the template defines whether the cart renders selections in a vertical format or a horizontal format. This is a display-level configuration that shapes how users visually review products, product details, and their selections while progressing through a Guided Selling flow. Choosing the appropriate template helps ensure that the cart layout fits the intended experience, such as a compact horizontal arrangement or a more detailed vertically oriented view. The template is therefore part of the cart UI configuration, rather than a mechanism for defining transactional data or server-side processing. Salesforce Industries CPQ Guided Selling is designed to provide configurable guided product-selection experiences, and its cart presentation can be adapted to the business's required user experience. See Salesforce's Industries CPQ Developer Guide and the Industries CPQ Developer certification-maintenance learning module for Industries CPQ configuration concepts.

QUESTION NO: 9

Which of these can you use in a promotion to change the price of a child product? (Choose TWO

Note: This question displayed answer options in random order when taking this Test.

A. MACD

B. Adjustment

C. Override

D. New order

E. Different price list

ANSWER: B C

Explanation:

Adjustment and **Override** are promotion pricing actions that can alter the price of a child product within an Industries CPQ promotion. An adjustment changes the product's calculated price relative to its normal pricing result. Depending on the configured adjustment, this can apply a discount, markup, or other price modification while retaining the underlying price-list-driven calculation as the starting point. This is useful when a promotion should offer a defined incentive on a particular included child item.

An override sets the child product to a specified promotional price rather than allowing its usual calculated price to remain in effect. It is appropriate when the promotion requires a fixed, known price for the child product, such as making an accessory free or setting a promotional monthly charge. Both actions can be scoped to the relevant child product in the promotion configuration, allowing the promotion to affect that item's price independently of other items in the offer.

These actions are part of the promotion and pricing configuration model described in the Salesforce Industries CPQ documentation. See the Salesforce Industries CPQ documentation and the Industries CPQ Foundations Trailhead module.

QUESTION NO: 10

The developerprepares to text the rule by running by running the appropriate API cache jb, but ...

Which two properties are causing the job to fail?

Choose 2 answers

A. Code

B. Default Value

C. Data Type

D. Values for caching

ANSWER: A C

Explanation:

Code and **Data Type** are the properties that determine whether the API cache job can correctly identify and process a rule input. The code is the programmatic identifier used to reference the rule property consistently during cache generation. It must be present and conform to the expected naming conventions so that the cache job can resolve the rule definition. The data type establishes how the cache engine interprets the property value, such as text, numeric, Boolean, or date information. A missing, unsupported, or inconsistent data type prevents the job from serializing and evaluating the rule values reliably, causing cache processing to fail. These fields form the required structural contract for a cacheable rule: one identifies the input and the other specifies how its value is handled. Salesforce Industries CPQ uses metadata-driven product configuration and rule processing, so valid identifiers and correctly defined value types are essential before testing or

populating cached results. See the Salesforce Industries CPQ developer reference at Salesforce Industries CPQ Reference and the Industries CPQ Advanced Product Configuration Trailhead module for related configuration concepts.

QUESTION NO: 11

A product does not need a baseprice to:

Note: This question displayed answer options in random order when taking this Test.

A. Show up as a pricing variable

B. Appear in the cart

C. Indicate the type of price it is

D. Have rules associated with it

ANSWER: D

Explanation:

Have rules associated with it is correct. In Salesforce Industries CPQ, product rules are configuration and qualification logic associated with products, product bundles, and catalog behavior. They can be used to control eligibility, enforce compatibility, add or remove products, display messages, and apply other guided-selling behavior. This relationship is established through the product-rule configuration and does not depend on defining a base-price record for the product.

Base pricing belongs to the pricing model: it supplies the starting monetary amount from which the pricing engine can calculate charges, adjustments, promotions, and totals. Product rules, by contrast, evaluate conditions and take configured actions while the customer configures an offer. Therefore, a product can participate in rule-driven catalog or configuration logic even when its commercial price is not defined through a base price. This separation lets an implementation model conditional items, noncommercial components, and eligibility logic independently of pricing setup. Salesforce's Industries CPQ documentation describes the platform's catalog, configuration, and pricing capabilities, while the product-rules documentation covers rules as a separate mechanism for controlling product behavior. See Salesforce Industries CPQ Developer Guide and Salesforce Help: Product Rules.

QUESTION NO: 12

What is Vlocity Guided Selling used for? (Choose FOUR)

Note: This question displayed answer options in random order when taking this Test.

For customers and partners who can self-order

A. To reduce average call handling time and agent training

B. To provide a better customer experience

C. To replace the regular Vlocity Cart for order capture

D. To use functions not found with the Vlocity Cart for order capture

E. To train agents in selling processes before they use the regular Vlocity Cart

F. To include helpful instructions for agents

ANSWER: A B C F

Explanation:

Vlocity Guided Selling is designed to make product selection and order capture easier through a guided, business-defined experience. It can support both assisted channels, such as service or sales agents, and self-service channels for customers

and partners. Consequently, *To reduce average call handling time and agent training* is a valid use: guided steps, eligibility logic, and contextual direction help users reach an appropriate product choice efficiently and consistently. *To provide a better customer experience* is also central, because the guided interaction can simplify complex offers and present relevant choices at the right point in the journey.

Guided Selling can be used *To replace the regular Vlocity Cart for order capture* when a tailored guided experience is preferable to the standard cart-led interaction. The guided application still supports the broader Industries CPQ ordering process while presenting a purpose-built user experience. Finally, *To include helpful instructions for agents* reflects its ability to display contextual guidance and business-process instructions during the selling flow. Salesforce Industries CPQ is intended to support configurable products, guided processes, and consistent omnichannel commerce experiences; see the Salesforce Industries CPQ Foundations Trailhead module and the Salesforce Industries CPQ developer documentation.

QUESTION NO: 13

Which of the following entities can be managed using Vlocity Product Console? (Choose FOUR)

Note: This question displayed answer options in random order when taking this Test.

A. Picklists

B. Product attributes

C. Object types

D. Products

E. Attribute categories

F. Configuration/Validation rules

G. Advanced Rules

H. Field mappings

ANSWER: A B C D

Explanation:

Vlocity Product Console provides a centralized catalog-management experience for defining the core metadata used by an Industries CPQ product catalog. **Products** are managed there as the catalog items that can be sold, bundled, and configured. **Product attributes** are also managed as reusable characteristics that capture configurable product information, such as a service tier, color, speed, or term. **Object types** establish the business context and target object for attributes, allowing the catalog to use attributes consistently for the appropriate product or related entity type. **Picklists** supply controlled sets of selectable values that attributes can use, ensuring configuration choices are standardized and reusable across catalog definitions. Together, these entities support the foundational relationship between a sellable offering, its configurable properties, the context in which those properties apply, and the permitted values presented to users. This is why they are catalog entities that Product Console can maintain directly. Salesforce's Industries CPQ learning materials describe Product Console as the interface for creating and maintaining product-catalog definitions and their supporting configuration metadata. See Salesforce Trailhead: Industries CPQ Foundations and Salesforce Help: Product Console.

QUESTION NO: 14

Which of these questions would you use to decide that you should use a promotion instead of an offer to market products? (Choose THREE)

Note: This question displayed answer options in random order when taking this Test.

A. Product is available to limited set of customers.

B. Product is available only to B2C customers.

C. Product is available for a limited time.

D. Product needs to be sold quickly.

E. Product cost is increasing.

ANSWER: A C D

Explanation:

Promotions in Salesforce Industries CPQ are designed to support time-bound, targeted, and commercially urgent marketing initiatives. “Product is available to limited set of customers.” is an appropriate promotion decision point because promotions can be qualified for specific customer segments, allowing the business to make a benefit available only to customers who meet defined eligibility criteria. “Product is available for a limited time.” also indicates a promotion: promotions commonly have effective dates and expiration dates that control when the promotional pricing, discount, or benefit can be applied. “Product needs to be sold quickly.” is likewise a promotion use case, since a temporary incentive can increase customer interest and create urgency for inventory clearance, a campaign, or a short-term sales objective. In contrast to a standard offer, which represents the regular commercial package being sold, a promotion overlays targeted eligibility and temporary commercial benefits on the underlying product or offer. This distinction enables the catalog to retain reusable base offers while marketing teams configure campaign-specific incentives separately. See Salesforce’s Industries CPQ promotions documentation and the Salesforce Industries CPQ foundations learning module.

QUESTION NO: 15

Context rules frameworks uses a single interface for all of its operations.

A. True

B. False

ANSWER: A

Explanation:

True is correct. Salesforce Industries CPQ’s Context Rules framework is designed around a single common interface for its operations. This consistent interface provides the contract through which the framework invokes context-rule functionality, allowing implementations to be handled uniformly rather than requiring a separate interface for each type of context operation. In practical development, this architecture makes custom context-rule behavior more predictable: developers implement the framework’s expected interface and return data in the format required by the Context Rules engine. The engine can then evaluate context data consistently when it determines whether a rule applies during CPQ processing. A single-interface approach also supports extensibility, because custom implementations can be substituted or enhanced without changing the mechanism used by the framework to call them. This is an important design concept when working with Industries CPQ configuration and custom rule logic, where reusable, standardized integration points help keep implementations maintainable. Salesforce’s Industries CPQ resources and contextual help provide additional material on configuring and extending Context Rules: Salesforce Help: Context Rules search and Trailhead: Industries CPQ Context Rules resources.

QUESTION NO: 16

Which of the following are the three Configuration/Validation implementations provided by Vlocity?

Note: This question displayed answer options in random order when taking this Test.

A. AdvancedCompatibilityImplementaion

B. DefaultRulesImplementation

C. ProductRelationshipValidationImpl

D. ValidationRulesImplementation

E. DefaultProductValidationImplementation

F. CustomValidationFramework

ANSWER: C D E

Explanation:

The provided Industries CPQ (formerly Vlocity) configuration and validation implementation classes are `ProductRelationshipValidationImpI`, `ValidationRulesImplementation`, and `DefaultProductValidationImplementation`. These implementations support the product-configuration validation lifecycle by evaluating whether a selected product and its relationships are allowed, applying validation rules defined for the catalog, and delivering default product-validation behavior. In practice, the configuration engine invokes the relevant implementation during product configuration so that invalid combinations, missing dependencies, and catalog rule violations can be identified before the cart or quote proceeds.

`ProductRelationshipValidationImpI` is the relationship-focused implementation, used when compatibility or dependency logic must be evaluated between products. `ValidationRulesImplementation` provides rule-based validation behavior for configured products. `DefaultProductValidationImplementation` supplies the standard validation implementation used by the Industries CPQ framework. These extension points allow an implementation to use the platform's standard validation model while supporting catalog-specific configuration requirements. Salesforce Industries CPQ developer documentation covers the product-configuration framework and its extensibility model in the Industries CPQ Developer Guide; related implementation guidance is available through Salesforce Industries documentation.

QUESTION NO: 17

What could be wrong if a promotion does not appear THREE) in the Promotions list of the cart? (Choose

Note: This question displayed answer options in random order when taking this Test.

A. The Effective From date of the promotion is missing.

B. The Effective Until date of the promotion is in the past.

C. The promotion status is not active.

D. No time policy is assigned to the promotion.

E. No time plan is assigned to the promotion.

ANSWER: A B D

Explanation:

A promotion's availability in the cart is governed by its effective-dating configuration and by the time-policy configuration used to determine whether it is currently eligible to be presented. Therefore, **The Effective From date of the promotion is missing.** can prevent the promotion from being evaluated as available for the cart date. Likewise, **The Effective Until date of the promotion is in the past.** means the promotion has expired and should not be offered to the user. Finally, **No time policy is assigned to the promotion.** can leave the promotion without the required temporal eligibility configuration for promotion selection and display. These settings are evaluated as part of catalog and promotion eligibility, so administrators should validate date ranges and associate the correct time policy before testing the promotion in the cart. Salesforce Industries CPQ uses effective dates and time-based eligibility to control when catalog items and commercial offers can be presented. See Salesforce's Promotion Management documentation and the Time Policies documentation for configuration guidance.

QUESTION NO: 18

What is the syntax to indicate a merge code or token in Vlocity Document Templates?

Note: This question displayed answer options in random order when taking this Test.

A. %field%

B. {{field}}

C. %%field%%

D. {%field%}

ANSWER: C

Explanation:

`%%field%%` is the correct token syntax for Vlocity (Salesforce Industries) Document Templates. Document Generation processes a template by locating text enclosed by two percent signs on each side, treating the enclosed value as a merge code, and replacing it with data from the document-generation data JSON. For example, a token such as `%%Account.Name%%` identifies the account-name value in the available data context. The double-percent delimiters are important because they allow the document-generation engine to distinguish dynamic merge codes from ordinary text that should remain unchanged in the generated document. Tokens can reference data made available by the configured DataRaptor, Integration Procedure, or other document-generation data source; therefore, the path inside the delimiters must match the structure and field names of that data. This convention is part of the template-authoring approach used by Salesforce Industries Document Generation, including templates created in supported document formats. See Salesforce's Document Generation documentation and the Salesforce Industries developer documentation for the Document Generation framework and template-related guidance.

QUESTION NO: 19

When viewed in the Price Details window, which of these can be used to detect the adjustments that have been applied to a line item in the cart? (Choose TWO)

Note: This question displayedanswer options in random order when taking this Test.

A. The display text of the base price

B. The display text of the pricing adjustment

C. The display text of the pricing override

D. The display text of the product description

E. The display text of theprice list

ANSWER: B C

Explanation:

The display text of the pricing adjustment and the display text of the pricing override identify changes that affect a cart line item's calculated price in the Price Details window. A pricing adjustment represents a pricing change applied by the CPQ pricing process, such as an adjustment derived from configured pricing rules, promotions, discounts, or other adjustment logic. Its display text gives the cart user a human-readable indication of the adjustment that contributed to the final price.

The display text of the pricing override similarly provides visibility when the normal calculated price has been superseded by an override. Because an override is a distinct price-affecting action, its display text enables users to recognize that the displayed result includes an overridden value rather than relying solely on the standard pricing calculation. Together, these two display-text values support transparent pricing review by exposing the applied price modifications directly in the line item's pricing breakdown. Salesforce Industries CPQ cart and pricing capabilities are documented in the Salesforce Industries Cart-Based APIs Developer Guide and the Salesforce Industries CPQ documentation.

QUESTION NO: 20

What tasks must be done when moving product catalogdata containing product attributes between orgs or spinning a sandbox? (Choose TWO)

Note: This question displayed answer options in random order when taking this Test.

A. Regenerate the attribute record ids stored in the JSONAttribute field

B. Refresh Pricebook, which runs the ProductAttributeBatchProcessor job

C. Run the Product Hierarchy Maintenance batch job

D. Use Data Loader to load attribute record ids

ANSWER: A C

Explanation:

When product catalog data that includes product attributes is moved to another org or introduced through a sandbox refresh, the attribute references embedded in product data must be made valid for the target org. **Regenerate the attribute record ids stored in the JSONAttribute field** is required because Salesforce record IDs are org-specific. Product attribute JSON can contain references to attribute records, and those source-org IDs cannot be assumed to identify the corresponding records after migration. Regenerating the IDs ensures that the JSON configuration points to the target-org attribute records and that attribute-driven catalog behavior remains intact.

Run the Product Hierarchy Maintenance batch job is also required to rebuild and maintain the derived hierarchy information used by the Industries product catalog. This processing ensures that the transferred catalog's parent-child product relationships and related hierarchy structures are correctly synchronized after the data move. Performing both tasks helps make the imported catalog usable rather than merely present as records in the target environment. Salesforce's product-catalog guidance describes the catalog and its product relationships, while Salesforce's deployment guidance explains the need to account for environment-specific record references during data movement. See Salesforce Industries Product Catalog documentation and Salesforce Data Loader documentation.

QUESTION NO: 21

Which two custom settings need to be set to true in order to avoid passing large sets of items to the configuration rules and the pricing engine?

Choose 2 answers

A. CacheAPI.TimeToLiveInDays

B. Cache APIFields

C. DeltaPrice

D. DeltaValidate

ANSWER: C D

Explanation:

DeltaPrice and **DeltaValidate** enable delta processing for Cart-Based APIs in Salesforce Industries CPQ. With delta processing enabled, the platform identifies the cart items affected by a change and sends only that relevant subset for pricing or validation rather than repeatedly sending the full cart item set. This is especially important for large or complex carts, where processing every line item after each update can increase response time and consume unnecessary server resources.

DeltaPrice applies this optimization to the pricing engine. It limits repricing activity to items whose prices may have been affected by the cart change, including related items where applicable. **DeltaValidate** applies the corresponding optimization to configuration-rule validation, focusing rule evaluation on the changed items and the portions of the cart impacted by those changes. Setting both custom settings to true therefore reduces the amount of cart data passed into the relevant processing flows while preserving the intended pricing and validation behavior. Salesforce's Industries CPQ documentation describes the Cart-Based API architecture and its performance-oriented processing model in the Salesforce Industries CPQ Developer Guide. Additional platform guidance for implementing scalable custom configuration behavior is available in the Salesforce Apex Developer Guide.

QUESTION NO: 22

When creating a pricing adjustment, what should a developer do to make sure that the user can understand why the price has been adjusted?

A. Use a dear name for the pacing adjustment

B. Include help text for the pricing adjustment

C. Include a configuration message for the pricing adjustment

D. Include display text for the pricing adjustment

ANSWER: D

Explanation:

Include display text for the pricing adjustment. In Salesforce Industries CPQ, a pricing adjustment can affect the amount presented to a sales representative or customer in the cart. The display text provides the user-facing description associated with that adjustment, making the reason for the changed price understandable at the point where the price is reviewed. A developer can use this text to communicate the business context of the adjustment, such as an eligible promotion, a negotiated discount, a surcharge, or another pricing rule. Clear display text supports price transparency because it connects the visible price change to a meaningful explanation rather than requiring users to infer why the total changed. This is particularly important when multiple pricing rules are evaluated and applied during cart pricing. Salesforce Industries CPQ pricing is driven by configured rules and adjustments, and the user experience should expose appropriate descriptive information alongside the resulting price. See the Salesforce Industries CPQ Developer Guide and Salesforce's pricing documentation for related pricing and quote-management concepts.

QUESTION NO: 23

A developer selects a smart thermostat from an account's list of assets and dicks Change to Order. While reviewing the order in the Cart, the developer notices the price of the existing asset in the Change order does not match the price that was displayed in the Service Asset Management view.

What is the reason for this discrepancy?

A. The asset was originally priced for a different user.

B. asset has a status of Disconnect in Change orders.

C. One-time charges are zeroed out on Change orders.

D. The product's price has changed.

ANSWER: C

Explanation:

One-time charges are zeroed out on Change orders. In Salesforce Industries CPQ asset-based ordering, a Change order represents a modification to an already purchased and installed asset rather than a new sale of the asset itself. The Service Asset Management view can display the asset's existing price information, including one-time charges that were applicable when the asset was originally ordered. However, when that asset is brought into the Cart through Change to Order, Industries CPQ sets the one-time charge amounts for the existing asset to zero. This prevents the customer from being charged again for a nonrecurring amount that was already paid as part of the original purchase. The Cart price therefore reflects the financial impact of the requested change, including any applicable recurring-charge adjustments, instead of duplicating the historic one-time price shown on the asset. This behavior is important when implementing asset lifecycle transactions because the asset record retains commercial history while the Change order calculates only charges relevant to the current transaction. Salesforce's Industries CPQ documentation covers asset-based ordering and change transactions in the Industries CPQ Developer Guide. Additional product documentation is available from Salesforce Industries Products and Services documentation.

QUESTION NO: 24

When usingrange attribute-based pricing, where do you specify the values for each range?

Note: This question displayed answer options in random order when taking this Test.

A. Pricing plan step parameters

B. Vlocity Picklists

C. Columns in the RangeAttributePricingMatrix

D. JSONAttribute field

ANSWER: C

Explanation:

Columns in the RangeAttributePricingMatrix is correct because range attribute-based pricing models price variation by defining the range values directly in the matrix's columns. The matrix is the configuration artifact that relates an attribute's range breakpoints to the applicable pricing result. For example, when a product's price varies according to a numeric or otherwise ordered attribute, the matrix columns hold the defined range values used during pricing evaluation. Industries CPQ can then evaluate the product's selected attribute value against those configured ranges and retrieve the associated price or adjustment from the matching matrix configuration. This separates the reusable pricing logic from the product's transactional data and allows administrators to maintain multiple range tiers without changing product code or pricing-plan logic. In practice, the range definitions must be structured consistently with the attribute and the pricing configuration so the pricing engine can identify the intended range at runtime. Salesforce's Industries CPQ learning materials describe attribute-based pricing as a matrix-driven approach to determining prices from product attributes, including advanced pricing scenarios. See Salesforce Trailhead: Industries CPQ Foundations and Salesforce Help: Industries Product Console.

QUESTION NO: 25

In the line item configuration window of Vlocity Cart, what information can be modified? (Choose TWO)

Note: This question displayed answer options in random order when taking this Test.

A. Service account

B. Billing account

C. Prices

D. Applied promotion

ANSWER: A B

Explanation:

Service account and **Billing account** can be modified from the line item configuration window in Vlocity Cart. These account selections determine important commercial context for the individual cart line. The service account identifies the customer account associated with receiving or using the selected product or service. This is particularly important when an order contains services for multiple locations, subscribers, or existing customer accounts. The billing account identifies the account responsible for billing the item, enabling the cart to support scenarios in which the party receiving a service differs from the party that pays for it.

Allowing these values to be updated at the line-item level lets a sales representative accurately model account relationships before pricing, validation, and order submission occur. Industries CPQ cart functionality is designed to manage complex communications and subscription ordering journeys, where service and billing responsibilities can vary by product or line item. The account assignments therefore form part of the line item's configuration context and are available for adjustment while configuring that item. Salesforce's Industries product documentation and learning materials provide broader context on Communications Cloud and Industries CPQ ordering capabilities: Salesforce Communications Cloud and Salesforce Trailhead: Industries CPQ for Communications Cloud.

QUESTION NO: 26

A developer needs to create a context rule in a Digital Commerce guided selling experience to evaluate the technology configuration installed at the account's location. The developer configures a new context dimension for the rule as shown below.

The developer prepares to test the rule by running the appropriate API cache job, but the job fails. Which two properties are causing the job to fail?

Choose 2 answers

A. Code

B. Values for caching

C. Data Type

D. Default Value

ANSWER: B D

Explanation:

Values for caching and **Default Value** are the properties that affect whether Industries CPQ can generate the context cache for this dimension. A context dimension represents a value that the context-rule engine uses when it evaluates eligibility and guided-selling rules. When the cache job runs, it builds the required cached combinations from the configured values. Therefore, the values supplied for caching must be valid for the dimension and must be configured in a format the cache process can resolve.

The default value must also be a valid value for the dimension's configured data type and cacheable value set. The cache-generation process uses that default when a request does not supply a context value. An invalid, unavailable, or inconsistent default prevents the process from constructing a valid context combination, causing the API cache job to fail. Correcting the cacheable values and ensuring that the default is one of the valid values enables the cache job to generate the context data used by Digital Commerce guided selling.

For background on Industries CPQ configuration and its Digital Commerce capabilities, see Salesforce Trailhead: Industries CPQ Foundations and Salesforce Help: Context Rules.

QUESTION NO: 27

Which capabilities are provided by Calculation Matrices? (Choose THREE)

Note: This question displayed answer options in random order when taking this Test.

A. Input columns used to evaluate a lookup

B. Output columns that return a price or adjustment value

C. Versions with effective dates

D. Parent-child product relationships

E. Cart session persistence

ANSWER: A B C

Explanation:

Calculation Matrices provide a declarative way to manage commercial lookup logic. They can define **input columns used to evaluate a lookup**, **output columns that return a price or adjustment value**, and **versions with effective dates**. Effective-dated versions allow a business to schedule pricing changes without replacing the entire matrix.

Calculation Matrices do not create product relationships or replace the cart APIs. They are pricing artifacts that are invoked by Calculation Procedures and Calculation Steps as part of the pricing runtime. See Salesforce Industries CPQ Developer Guide.

QUESTION NO: 28

Which rule type allows you toauto-add products based on other products in the Cart?

A. Context

B. Advanced

ANSWER: B

Explanation:

Advanced is the rule type used to automatically add products in response to products that are already present in the cart. In Salesforce Industries CPQ, advanced rules support product-configuration logic that evaluates cart contents and then performs actions, including adding a required or related product. This is useful when a selected base product requires an accessory, service component, add-on, or other dependent offering to be included without requiring the user to locate and add it manually. The rule can use conditions based on the selected products and their attributes, then execute an auto-add action when those conditions are met. As a result, the cart stays aligned with commercial and technical dependencies while the sales process remains efficient and guided. Salesforce describes Industries CPQ as supporting guided product configuration and rules-driven selling behavior; advanced rules provide the appropriate mechanism when that behavior must evaluate and act on cart-product relationships. See Salesforce's Advanced Rules documentation and the Salesforce Industries CPQ Foundations learning module for the relevant configuration concepts.

QUESTION NO: 29

What tasks must be done when new product bundles are added to the product catalog? (Choose TWO)

Note: This question displayed answer options in random order when taking this Test.

A. Clear the platform cache

B. Refresh Pricebook, which runs the ProductHierarchyBatchProcessor job

C. Run the Product Hierarchy Maintenance batch job

D. No manual action is required

ANSWER: B C

Explanation:

When new product bundles are introduced, Industries CPQ must rebuild and maintain the derived product-hierarchy data used to present and process bundle structures efficiently. **Refresh Pricebook, which runs the ProductHierarchyBatchProcessor job** is required because the price book refresh updates the catalog's generated pricing and hierarchy-related data after catalog changes. Running this process ensures that the newly added bundle and its relationships are represented in the data consumed by CPQ product-selection and pricing processes.

Run the Product Hierarchy Maintenance batch job is also required to maintain the product hierarchy after the bundle is added. This batch operation updates hierarchy maintenance records and supports accurate traversal of bundle parent-child relationships. Together, the refresh and hierarchy-maintenance processes make the new bundle available with a consistent structure and associated catalog data, rather than relying on stale derived information. Salesforce Industries CPQ uses asynchronous and batch processing for catalog operations because product catalogs can contain deeply nested bundle structures and large numbers of related records. See Salesforce's Product Hierarchy documentation and the Industries CPQ Developer Guide for the catalog and product-hierarchy processing model.

QUESTION NO: 30

In Vlocity Cart, what are the buttons in the top right of the cart header?

Note: This question displayed answer options in random order when taking this Test.

A. Vlocity Actions

B. Buttons defined on the Salesforce layout

C. Buttons defined in the cpq-cart-setup template

D. Buttons defined on the Order sObject

ANSWER: A

Explanation:

Vlocity Actions is correct. In the Vlocity (Salesforce Industries) Cart user interface, the controls displayed at the upper right of the cart header are action controls supplied through the Vlocity Actions framework. These actions provide contextual operations that a user can perform while working with a cart, such as progressing the cart process or invoking a configured business operation. The action framework lets an implementation expose behavior appropriate to the cart's state, the user's context, and the applicable business process, rather than treating these controls as ordinary record-page buttons.

This distinction is important when configuring and troubleshooting a Cart implementation. Cart behavior is driven by Industries CPQ and OmniStudio configuration, including the cart's actions and their associated logic, so developers should inspect the configured Vlocity Actions when determining which header controls appear and what they do. Salesforce's Industries CPQ learning material describes the product's cart-based guided selling and ordering capabilities, while OmniStudio documentation provides the broader configuration model used by Industries solutions. See Salesforce Trailhead: Industries CPQ Foundations and Salesforce OmniStudio Developer Guide.

QUESTION NO: 31

In Vlocity Context Rules, what is a context scope?

Note: This question displayed answer options in random order when taking this Test.

A. The relational path from a root sObject, such as an Order, to related sObjects, such as Account

B. Avariable that stores rule condition values

C. A link to data stored in sObjects, calculated using a function, or typed in during design-time

D. A multi-dimensional array of sObject data

ANSWER: A

Explanation:

The relational path from a root sObject, such as an Order, to related sObjects, such as Account is correct. In Industries CPQ Context Rules, a context scope establishes the portion of the Salesforce object graph that a rule can traverse and evaluate. It begins with a designated root object that supplies the rule's starting context—for example, an Order or Cart—and defines the relationship path to the related records whose fields may be needed by conditions and actions. This lets a rule evaluate information that is meaningful only when viewed through its relationship to the transaction, such as account-level attributes associated with an order. The scope is therefore a navigation and data-context definition, rather than a place to store a value or an independently structured collection of records. Defining this path makes the available context predictable and enables the rule engine to resolve related data consistently while processing CPQ transactions. Salesforce object relationships provide the underlying parent-child and lookup relationships that make such traversal possible; see the Salesforce Object Reference: Relationships Among Objects. For broader Industries CPQ implementation and configuration guidance, consult the Salesforce Industries documentation.

QUESTION NO: 32

Which of these is true for a time plan?

Note: This question displayed answer options in random order when takingthis Test.

A. it has a start date.

B. it has an end date.

C. it has a length of time.

D. it can be prorated.

ANSWER: D

Explanation:

“it can be prorated.” is correct. In Salesforce Industries CPQ, a time plan supports time-based charging and establishes how a charge is applied across a defined period. Proration is an important capability for recurring or usage-period charges because a customer’s service can begin, change, or end partway through a standard billing interval. Rather than charging the full periodic amount when only part of that interval is applicable, the system can calculate an adjusted amount that reflects the applicable portion of the period. This enables accurate commercial handling for scenarios such as mid-cycle activation, cancellation, suspension, or a subscription change.

Time-plan configuration is part of the broader Industries product catalog and pricing model, which supports defining commercial offers and the charging behavior associated with those offers. The ability to prorate helps ensure that billing outcomes align with the actual effective service duration and with the organization’s billing policies. Salesforce describes Industries CPQ as supporting complex, configurable product and pricing experiences, including recurring charges and lifecycle-based changes. See Salesforce’s Industries CPQ overview and the Salesforce Industries Communications, Media, and Energy developer documentation for the platform context.

QUESTION NO: 33

All Vlocity Picklist definition fields are not visible at run-time during order capture in Vlocity Cart.

A. True

B. False

ANSWER: A

Explanation:

True is correct. A Vlocity (Salesforce Industries) Picklist Definition is configuration metadata used to control the values and behavior of a picklist. Its definition can include administrative and implementation-oriented information, such as identifiers, sequencing, activation and default-value settings, effective usage, and display-related attributes. During order capture, Vlocity Cart renders the picklist as an end-user control and exposes only the values and presentation information needed for the shopper or sales agent to make a valid selection. It does not expose the complete underlying Picklist Definition record and every configuration field at runtime.

This separation is intentional: catalog and configuration metadata supports product-model maintenance and runtime rules, while the cart experience should present a concise, guided interface. The runtime UI therefore uses the picklist’s configured display values and applicable eligibility or context behavior rather than displaying all administrative definition fields. This is consistent with Salesforce Industries CPQ’s catalog-driven approach, in which product configuration data is authored in the catalog and consumed by guided selling and cart experiences. See Salesforce’s Communications Cloud overview and the Salesforce Industries documentation entry point for product and documentation context.

QUESTION NO: 34

In which sequence do rules execute to ensure a perfect order in the Cart?

A. Advanced Rules Framework, Context Rules Framework, Compatibility Rules

B. Eligibility Rules, Context Rules, Advanced Rules

C. Context Rules Framework, Advanced Rules Framework.

D. Context Rules Framework, Compatibility Rules Framework, Advanced Rules Framework.

ANSWER: C

Explanation:

Context Rules Framework, Advanced Rules Framework. is the correct execution sequence for Industries CPQ cart rule processing. Context rules run first because they establish the applicable product and pricing context for the cart interaction. They evaluate contextual information—such as attributes, customer data, account characteristics, or selections already present in the cart—to determine which rules and outcomes are relevant at that point in the guided selling or cart journey. Establishing this context before later rule evaluation ensures that downstream logic works with the appropriate set of products, promotions, and conditions.

After the contextual evaluation is complete, the Advanced Rules Framework runs. Advanced Rules can then apply their more detailed business logic to the cart using the context that has already been resolved. This ordered processing is important for predictable cart behavior: the cart first identifies the relevant context and then applies advanced actions, validations, or updates based on that context. Salesforce's Industries CPQ documentation describes Context Rules and Advanced Rules as separate rule frameworks used to control product configuration and cart behavior. See the Salesforce Industries Context Rules documentation and the Salesforce Industries Advanced Rules documentation.

QUESTION NO: 35

A company has two telesales teams: one that does business in Russian and another that does business in English. The developer decides to enable the multi-language feature in the shared catalog. However, during testing, Russian translations do not display in the Cart.

Which three things should the developer verify?

Choose 3 answers

A. User's Locale is set to Russian.

B. Russian translations for the products exist in the StringTranslation object.

C. The Russian Locale is activated in the Vlocity Locale Code picklist.

D. Russian translations for the products exist in the Translation Workbench.

E. Users Language is setto Russian.

ANSWER: A B E

Explanation:

The developer should verify the user's language, the user's locale, and the translated catalog strings. In Industries CPQ, the running user's language controls the language context used when Cart retrieves localized text. A Russian telesales user therefore needs their Language setting configured for Russian. The user's Locale should also be Russian so regional formatting and the locale context used by the application are aligned with the intended Russian experience. Salesforce distinguishes language from locale: language controls displayed text where translations are available, while locale controls regional presentation settings such as formats.

The shared catalog must also contain Russian entries in the `StringTranslation` object. Industries CPQ uses these translation records to return localized values for catalog and Cart content when multi-language is enabled. Enabling multi-language alone does not create translated product strings; the required Russian translation records must be present and associated with the applicable source strings. Confirming these three items ensures the Cart has both a Russian user

context and the Russian text to resolve for the catalog content. For Salesforce's definitions of user language and locale, see Manage User Settings. For Industries product documentation, see Salesforce Industries Documentation.

QUESTION NO: 36

In an attribute-based pricing matrix, which field is used asthe key by the matching algorithm?

Note: This question displayed answer options in random order when taking this Test.

A. Source Product Code

B. Source Product Name

C. Source Product RecordId

D. Source Product RecordId and Source Product Name

ANSWER: A

Explanation:

Source Product Code is the key used by the attribute-based pricing matrix matching algorithm. Attribute-based pricing first establishes the product context for the item being priced, and the source product code supplies the stable product identifier used to locate the applicable matrix rows. Once the relevant product context is identified through this code, the algorithm can evaluate the configured attribute values and select the matching pricing result, such as a price, adjustment, or related pricing action. Using a product code is appropriate because it is a business-level identifier intended to distinguish product definitions consistently across the catalog and pricing configuration. This enables one pricing matrix to support product-specific attribute combinations while preserving predictable matching behavior when the matrix is evaluated during cart or quote pricing. Salesforce Industries CPQ pricing capabilities are part of the Industries product catalog and CPQ framework, which uses catalog metadata and configured rules to determine commercial outcomes. See Salesforce's Industries Cloud overview and the Industries CPQ Foundations Trailhead module for platform context.

QUESTION NO: 37

A guided process allows you to create new product bundles in the Cart.

A. True

B. False

ANSWER: A

Explanation:

True is correct. In Salesforce Industries CPQ, a guided process supports a structured, user-facing workflow for performing tasks in the Cart, including creating and configuring a new product bundle. The process can walk a user through selections in a defined order, applying the product-model rules, eligibility logic, and configuration requirements that govern the bundle. This is especially useful for complex offerings, where agents should not need to manually locate every child product or understand all dependency rules before building a valid configuration. Guided processes improve consistency because the Cart interaction is driven by the configured product catalog and its associated rules rather than an ad hoc sequence of manual actions. The resulting bundle is added to the cart as a configured commercial product structure, where it can continue through pricing, validation, and checkout or order-capture activities. Salesforce's Industries CPQ developer documentation describes the Cart as the central runtime experience for adding and configuring products, while the Industries CPQ documentation provides the relevant implementation context for guided and catalog-driven commerce flows. See the Salesforce Industries Communications, Media, and Energy developer documentation and Salesforce Help search for Industries CPQ guided processes.

QUESTION NO: 38

A company isbuilding a B2C website for its customers. The company wants to allow its customers to buy its products online, so they are designing the website to support a self-service selling journey, when the company launches new promotions, website traffic dramatically increases by several million API calls.

Which type of APIs should the developer use to meet these requirements?

A. Digital Commerce APIs

B. integration Procedure APIs

C. Data Raptor APIs

D. Cart-Based APIs

ANSWER: A

Explanation:

Digital Commerce APIs are correct because they are intended for customer-facing, digital self-service commerce experiences, such as a business-to-consumer website where shoppers browse a catalog, configure products, manage a cart, and place an order. These APIs expose Industries CPQ commerce capabilities through REST resources designed for digital channels, allowing the website or mobile experience to drive the buying journey without requiring a sales representative to operate an internal console.

The scenario's very large traffic spikes during promotional launches are also central to this choice. A public commerce channel must support large numbers of independent shopper interactions and requests across catalog discovery, eligibility, pricing, cart operations, and checkout. Digital Commerce APIs provide the API layer built for those scalable online-selling use cases and integrate with the Industries product catalog and cart capabilities that support the underlying transaction flow. The developer should therefore use Digital Commerce APIs as the contract between the B2C storefront and Salesforce Industries CPQ. Salesforce's Digital Commerce API documentation provides the relevant overview and resource guidance: Digital Commerce API Developer Guide. Additional Industries CPQ API documentation is available at Salesforce Industries Communications, Media, and Energy Developer Guide.

QUESTION NO: 39

Select all that apply: What are the phases of the Digital Commerce user experience model?

A. Configure

B. Basket

C. Cart

D. Payment

E. Submit

F. Checkout

ANSWER: A B D

Explanation:

The Digital Commerce user experience model is organized around **Configure**, **Basket**, and **Payment**. Configure is the guided product-selection and product-configuration phase. It presents eligible offers and captures the attributes, quantities, and relationships needed to build a valid commercial solution. Industries CPQ rules, pricing, compatibility logic, and context-based eligibility are applied as the shopper configures the offer.

Basket is the phase in which the shopper reviews the accumulated configured items as a commercial basket. This stage supports reviewing the overall selection and its commercial context before continuing the transaction flow. It provides the consolidated handoff from configuration to the final transaction activity.

Payment is the final experience phase, where the commerce flow gathers the information needed to complete the purchase and proceed with transaction submission. Keeping payment as a defined phase enables implementations to tailor payment, billing, fulfillment, and order-capture behavior without changing the earlier configuration experience. Salesforce Industries documentation and learning resources describe Digital Commerce and its guided, API-driven commerce capabilities; see the Salesforce Help documentation for Digital Commerce and the Salesforce Trailhead Digital Commerce learning resources.

QUESTION NO: 40

The options forpricing bundles include: (Choose TWO)

Note: This question displayed answer options in random order when taking this Test.

A. Assigning the parent a zero price

B. Assigning a price to the parent and zero prices to the children

C. Assigning only pricing elements to the parent and children

D. Assigning only pricing variables to the parent and children

ANSWER: A B

Explanation:

Salesforce Industries CPQ supports flexible bundle-pricing designs because the commercial value can be held either at the bundle parent level or at the individual child-product level. **Assigning the parent a zero price** is appropriate when the bundle acts principally as a container for configuration and the selected child products carry the sellable prices. The resulting quote total is therefore derived from the components that the customer chooses within the bundle. This is especially useful where optional components, quantities, or attributes cause the bundle's price to vary.

Assigning a price to the parent and zero prices to the children is the complementary model. In that design, the parent represents the priced commercial offering, while the child products provide the included structure, eligibility, and configuration detail without independently increasing the price. This allows the quote and order to show the bundle's constituent products while charging a single bundle amount. Both approaches preserve the parent-child product hierarchy while placing the price at the level that best represents the offer. Salesforce Industries CPQ documentation and learning materials describe product configuration and pricing capabilities for these catalog-driven offers: Salesforce Industries CPQ Developer Guide and Salesforce Trailhead: Industries CPQ Foundations.

QUESTION NO: 41

Which of these uses a calculation of the base price?

A. Pricing element adjustment

B. Pricing element override

C. Price list

ANSWER: A

Explanation:

Pricing element adjustment is correct because an adjustment derives a price change by applying configured calculation logic to the product's base price. In Salesforce Industries CPQ, pricing elements are evaluated as part of the pricing procedure, and an adjustment can increase or decrease the base price using a percentage, amount, or rule-driven value. This makes the resulting charge dependent on the calculated base-price context rather than simply replacing that price with a fixed value. For example, a percentage-based promotional discount is evaluated against the applicable base price, so the adjustment amount changes when the base price changes. This behavior supports reusable commercial rules, such as discounts, markups, surcharges, and conditional price modifications, while preserving the base price as the foundation of the calculation. The pricing procedure sequences these elements so that the platform can calculate the appropriate final price

consistently for the configured product and context. Salesforce's Industries CPQ learning materials describe pricing configuration and the role of pricing elements in building calculated prices: Salesforce Trailhead: Industries CPQ Foundations. Additional Industries Communications, Media, and Energy developer documentation is available at Salesforce Industries CME Developer Guide.

QUESTION NO: 42

By what mechanism is a Document Template linked to the Opportunity, Quote or Order objects?

Note: This question displayed answer options in random order when taking this Test.

A. PDF Document Template custom setting

B. Applicable Objects in Template Settings

C. Applicable Item Types in Template Settings

D. Document Template object picklist

ANSWER: B

Explanation:

Applicable Objects in Template Settings is correct because this setting defines the Salesforce record types and object contexts in which an Industries CPQ Document Template can be used. A document template is configured for a business transaction context, such as an Opportunity, Quote, or Order, by listing the relevant supported objects as applicable objects. This association lets Industries CPQ determine when the template is available for document generation and which record supplies the data context used to merge information into the generated output.

Configuring applicable objects is important when a template is intended for a particular stage of the sales or fulfillment process. For example, a proposal-oriented template can be made available in the quote context, while an order summary template can be associated with orders. The template configuration thereby governs the template's functional availability rather than merely defining its file format or its content structure. This object-level applicability works with the template's data mappings and document-generation configuration to ensure that users generate the intended document from the appropriate business record. Salesforce's Industries CPQ resources describe the product's transaction-based configuration model and document-generation capabilities; see Salesforce Trailhead: Industries CPQ Foundations and Salesforce Industries product documentation.

QUESTION NO: 43

A company has two telesales teams: one that does business in Russian and another that does business in English. The developer decides to enable the multi-language feature in the shared catalog. However, during testing, Russian translations do not display in the Cart.

Which three things should the developer verify?

Choose 3 answers

A. User's Locale is set to Russian.

B. Russian translations for the products exist in the StringTranslation object.

C. The Russian Locale is activated in the Vlocity Locale Code picklist.

D. Russian translations for the products exist in the Translation Workbench.

E. Users Language is set to Russian.

ANSWER: A B C

Explanation:

Russian Cart translations depend on the Industries CPQ multilingual catalog configuration resolving a Russian locale and finding matching translated strings. The user's Locale must be set to Russian so the Cart requests localized catalog content using the applicable locale context. Salesforce distinguishes locale settings, which control regional presentation and locale-dependent behavior, from other user preferences; the platform exposes the locale through the UserInfo APIs described in the Salesforce UserInfo reference.

The shared catalog must also contain Russian product translations as StringTranslation records. Industries CPQ uses these translated strings for catalog and Cart content rather than relying solely on standard Salesforce metadata translations. The translation records must be associated with the relevant catalog entities and locale so that the Cart can resolve a Russian value at runtime.

Finally, Russian must be an active value in the Vlocity Locale Code picklist. This configuration makes the locale available to Industries CPQ's multilingual framework and allows the locale used by the user and translations to be recognized consistently. These locale and translation prerequisites support Salesforce's general localization model, including locale-aware application behavior documented in Salesforce user profile settings.

QUESTION NO: 44

Tightest Match calculates multiple conditions as follows:

A. AND rule condition weights are summed together

B. OR rule conditions select the highest weighted value

C. Both of the above

ANSWER: C

Explanation:

Both of the above is correct because Tightest Match evaluates composite rule criteria according to their logical grouping and associated weights. When conditions are joined with AND, every condition must be satisfied for that branch to qualify. The weights of the satisfied AND conditions are added, producing a combined score that reflects how specifically the full set of conditions matches the transaction context. When conditions are joined with OR, only one alternative condition needs to qualify. In that case, Tightest Match uses the highest applicable weighted value rather than accumulating every alternative, because OR represents competing paths rather than requirements that must all be met. This scoring behavior enables Industries CPQ to select the most specific applicable rule when several rules could otherwise apply, such as for pricing, promotions, eligibility, or configuration behavior. The resulting comparison is based on the calculated match score, ensuring that a narrowly targeted rule can take precedence over a broader matching rule. Salesforce describes Industries Communications, Media, and Energy capabilities and implementation resources in its Industries CME Developer Guide, and provides related product-learning material through Salesforce Trailhead for Communications.

QUESTION NO: 45

Which three of the following product relationship types are available when creating a Vlocity Configuration/Validation rule?

Note: This question displayed answer options in random order when taking this Test.

A. Requires

B. Excludes

C. Recommends

D. Optional

E. Limited Quantity

F. Suggests

G. Not Compatible

H. Requires, Excludes, and Recommends

ANSWER: H

Explanation:

Requires, Excludes, and Recommends is correct because these are the product relationship behaviors supported for configuration and validation in Salesforce Industries CPQ (formerly Vlocity CPQ). A *Requires* relationship enforces the inclusion of a related product when the source product is selected. An *Excludes* relationship prevents incompatible products from being selected together. A *Recommends* relationship presents an associated product as a recommendation during configuration, allowing guided selling without making that associated product mandatory. These relationship types let an implementation model compatibility, dependency, and guided-selling requirements directly within the product catalog and configuration experience. The relationship is evaluated as part of the configuration process so that the resulting product selection complies with defined catalog rules while still supporting relevant recommendations to the user. Salesforce Industries CPQ configuration capabilities are covered in the Salesforce Industries CPQ Foundations Trailhead module and the Industries Communications, Media, and Energy Developer Guide.

QUESTION NO: 46

Which of the following are features of the Product List in Vlocity Cart? (Choose TWO)

Note: This question displayed answer options in random order when taking this Test.

A. It is only displayed on the Opportunity tab.

B. It is based on a specific price list.

C. It uses Vlocity Rules to display available and eligible products.

D. Allows hierarchical product structures.

ANSWER: B C

Explanation:

It is based on a specific price list. is correct because Industries CPQ uses price lists to determine the catalog of products and prices available for a particular sales context. The Product List retrieves products associated with the applicable price list, allowing the cart experience to present the appropriate commercial catalog for the customer, channel, market, or transaction. This price-list-driven approach supports controlled product availability as well as consistent pricing throughout the cart process.

It uses Vlocity Rules to display available and eligible products. is also correct. Product availability in the cart can be evaluated dynamically using Industries CPQ rules and qualification logic. These rules assess the transaction context and product relationships to determine what a user can see and select. Consequently, the Product List can show products that are eligible for the current customer and order scenario rather than presenting an unrestricted catalog. This helps guide users toward valid product selections before configuration and checkout. Salesforce Industries CPQ documentation describes the cart as a guided selling experience that uses catalog, pricing, and rules capabilities to manage product selection. See Salesforce Help: Industries Products and Pricing and Trailhead: Industries CPQ Foundations.

QUESTION NO: 47

MACD or Asset-Based Ordering capabilities are powered by what Vlocity technology?

Note: This question displayed answer options in random order when taking this Test.

A. OmniScript and Vlocity Cart

B. Vlocity Cart

C. OmniScript

D. Vlocity Rules

ANSWER: B

Explanation:

Vlocity Cart is the technology that powers MACD (move, add, change, and disconnect) and asset-based ordering in Salesforce Industries CPQ. Asset-based ordering starts from a customer's existing installed products and services, represented as assets, rather than treating each transaction as an entirely new order. The cart evaluates the requested changes against the current asset structure, identifies the affected products and relationships, and builds the appropriate order actions needed to add, modify, move, or disconnect services. It also maintains the context needed to support complex commercial-product configurations and lifecycle changes across orders. This cart-driven approach is essential for industries such as communications, media, and energy, where orders commonly alter existing subscriptions, bundles, and service hierarchies. Salesforce Industries CPQ documentation describes asset-based ordering as a capability for managing changes to existing customer assets, while the Industries CPQ cart provides the ordering framework used to create and process those changes. See Salesforce's Asset-Based Ordering documentation and the Salesforce Industries CPQ Developer Guide for product architecture and implementation guidance.

QUESTION NO: 48

You apply rule sets rather than individual context rules to products or promotions.

A. True

B. False

ANSWER: A

Explanation:

True is correct. In Salesforce Industries CPQ, context rules define conditional logic that determines whether a product or promotion is available, eligible, visible, or subject to a particular action for a given sales context. Individual context rules are organized into a context rule set, which provides the reusable, manageable unit for associating that logic with the catalog. The rule set can contain one or more context rules and can be configured to evaluate them according to the intended business behavior. Products and promotions reference the context rule set rather than requiring each underlying rule to be attached separately. This design lets an administrator maintain a related collection of eligibility or availability conditions in one place and reuse that collection across catalog items. It also makes changes safer and more efficient: updating a rule within the assigned set affects the relevant product or promotion behavior without having to recreate every association. Salesforce's Industries CPQ documentation describes context rules and their organization into rule sets for catalog configuration. See the Salesforce Help documentation on context rules and the Salesforce Industries reference documentation.