

# DUMPS ARENA

## Oracle CPQ 2024 Implementation Professional

Oracle 1z0-1033-24

Version Demo

Total Demo Questions: 12

Total Premium Questions: 124

Buy Premium PDF

<https://dumpsarena.co>

[sales@dumpsarena.co](mailto:sales@dumpsarena.co)

[sales@dumpsarena.co](mailto:sales@dumpsarena.co)  
[dumpsarena.co](https://dumpsarena.co)

## Topic Break Down

| Topic                        | No. of Questions |
|------------------------------|------------------|
| Topic 1, CPQ Administration  | 27               |
| Topic 2, CPQ Configuration   | 41               |
| Topic 3, CPQ Commerce        | 23               |
| Topic 4, CPQ Process         | 8                |
| Topic 5, CPQ Document Engine | 8                |
| Topic 6, CPQ Integration     | 16               |
| Topic 7, Mix Questions       | 1                |
| Total                        | 124              |

QUESTION NO: 1

Identify the three data types available for part custom fields. (Choose three.)

- A. "Date," can include time
- B. "String," can be changed to "Single Select Menu"
- C. "Boolean"
- D. "Float," can be changed to "Integer"
- E. "Currency"
- F. "Multi Select menu"

**ANSWER: B D E**

**Explanation:**

Part custom fields support the core types used to store and present part-level data in the CPQ part repository: String, Float, and Currency. A String field stores text-based values and can be configured for controlled selection behavior, including use as a Single Select Menu where appropriate. This makes it suitable for values such as classifications, categories, and other text or list-driven part properties. A Float field stores numeric values that may require decimal precision; where a whole-number representation is appropriate, it can be changed to Integer. Currency is provided specifically for monetary part values, allowing CPQ to handle amounts in the context of the applicable currency configuration. These types enable administrators to model descriptive, numeric, and monetary characteristics directly on parts, so the data can be maintained in the part catalog and used by commerce processes. Oracle's CPQ documentation library is available at [Oracle CPQ Documentation](#); related implementation and administration learning resources are available through [Oracle CPQ Training](#).

QUESTION NO: 2

You want to perform a configuration migration from a development site to a QA site. What is the Oracle-recommended way to accomplish this?

- A. Using a DB Link, log in to the development site through the Migration Center section, select the components that need to be migrated, and start the migration.
- B. Open Migration Center in the development site, log in to the QA site through the Migration Center section, select the components that need to be migrated, and start the migration.
- C. Open Migration Center in the QA site, log in to the development site through the Migration Center section, select the components that need to be migrated, and start the migration.
- D. Download the components that need to be migrated from the development site and upload them to the QA site.

**ANSWER: C**

**Explanation:**

Open Migration Center in the QA site, log in to the development site through the Migration Center section, select the components that need to be migrated, and start the migration. This follows Oracle CPQ's target-initiated migration model: the receiving environment, in this case QA, is the environment in which an administrator opens Migration Center and establishes a connection to the source environment. The administrator then chooses the required configuration components from the development source and imports them into QA. Performing the migration from the destination helps ensure that the selected artifacts are brought into the intended target site and that the target site's migration process controls the import. This approach is appropriate for promoting tested configuration changes through the normal development-to-QA lifecycle while preserving the distinction between the source and receiving environments. Oracle's CPQ documentation describes Migration Center as the mechanism for moving configuration data between CPQ environments; consult the [Oracle CPQ documentation library](#) and the [Oracle CPQ Administration documentation](#) for the applicable release-specific migration procedures and component considerations.

### QUESTION NO: 3

The commerce step "Pending" has four participant profiles in the following order: SalesRep, SalesManager, Owner, and Admin. The Admin profile allows access with the FullAccess user type. The Owner profile has at performer steps "start" step selected. The step has an auto-forwarding rule.

A FullAccess user with the SalesGroupA and SalesManagerA groups creates a new quote and saves it, which moves the quote to "Pending."

The auto-forwarding rule returned the following two lines:

SalesManagerA~Company~SalesManager

SalesGroupA~Company~SalesRep

Which profile will be used to display the quote in the "Pending" step to this user?

- A. SalesManager profile
- B. Owner profile
- C. Admin profile
- D. SalesRep profile

**ANSWER: D**

#### Explanation:

**SalesRep profile** will be used. Oracle CPQ evaluates the participant profiles configured for a commerce process step in their displayed order and uses the first profile for which the user qualifies. The auto-forwarding result establishes that SalesGroupA is forwarded as a participant with the SalesRep role. Because the user belongs to SalesGroupA, that user qualifies for the SalesRep participant profile when the quote enters Pending.

The same user also has SalesManagerA, which is returned as a SalesManager participant, and the user's FullAccess type can satisfy the relevant access condition. However, the governing behavior for profile selection is the configured participant-profile sequence. SalesRep appears before SalesManager, Owner, and Admin, so it is the first applicable profile and therefore determines how the Pending-step quote is displayed to this user. The forwarding output's line order does not replace the participant-profile order configured on the commerce step.

This behavior is part of CPQ Commerce process access and participant-profile configuration, where profiles control the user experience and permissions available at a process step. See Oracle's [Oracle CPQ documentation](#) and the [Oracle CPQ product documentation resources](#).

### QUESTION NO: 4

A FullAccess User is attempting to update parts through the SOAP API using Web Services 2.0.

When attempting to update a specific part, the user gets the following error:

"The security token could not be authenticated or authorized."

What should the user verify?

- A. That the user is a Partner User
- B. That the part number is valid
- C. That the password used in the header of the SOAP call is correct
- D. That the user has API access permissions

**ANSWER: C**

**Explanation:**

That the password used in the header of the SOAP call is correct is the appropriate verification because Web Services 2.0 authenticates the calling CPQ user from the WS-Security credentials included in the SOAP request header. The security token contains the username-token authentication data, including the password or password digest as applicable. If the password supplied by the client does not match the current password for the named CPQ user, CPQ cannot authenticate the token and returns the stated security-token authentication or authorization error before it can process the requested part update.

This should be checked especially after a password reset, password expiration, account update, or a change to the credentials stored in the integration middleware or SOAP client. The integration should use the current credentials for the same FullAccess user configured as the username in the security header. Once the SOAP header contains valid, current credentials in the expected WS-Security format, CPQ can authenticate the request and proceed to evaluate the update operation. Oracle's CPQ documentation landing page is available at [Oracle Configure, Price, Quote documentation](#), and Oracle describes WS-Security UsernameToken credential handling in its [WebLogic Server secure web services documentation](#).

**QUESTION NO: 5**

What are three valid reasons for a Configuration rule failing to execute? (Choose three.)

- A. A runtime error occurs within the rule, which is firing before the current rule.
- B. The status of the rule is set to Inactive.
- C. The rule's Condition attributes are hidden in Configuration Layout.
- D. The attributes used in the rule are not added in Configuration Layout.

**ANSWER: A B D****Explanation:**

A runtime error in a configuration rule that fires earlier can prevent later rule processing in the same configuration evaluation cycle. Configuration rules are processed according to their configured sequencing and dependencies, so an unhandled error must be resolved before reliable downstream execution can occur. A rule whose status is set to Inactive is intentionally excluded from the runtime rule evaluation process; it remains available for administration but does not execute for users configuring a transaction. In addition, attributes used by a configuration rule must be included in the applicable Configuration Layout. The layout establishes the configuration model presented and evaluated for the user session, allowing the rule engine to obtain and react to the relevant attribute values. Ensuring that required attributes are present in the layout, that the rule is active, and that earlier rules complete without runtime errors is therefore essential when troubleshooting rule execution. Oracle's CPQ documentation provides the configuration administration context and release documentation at [Oracle CPQ Documentation](#) and the product's configuration capabilities are described at [Oracle Configure, Price, Quote](#).

**QUESTION NO: 6**

Select two statements that are true about Test Script found in a function debugger. (Choose two.)

- A. Test Script is template of how the BML function should be written.
- B. Test Script provides a way to test a BML Library function when array type attributes are input parameters.
- C. Test Script describes the various functions used in BML.
- D. Test Script can be used to compute multiple iterations of the library functions and print each.

**ANSWER: B D****Explanation:**

Test Script provides a way to test a BML Library function when array type attributes are input parameters because it lets an implementer construct and pass inputs that are more complex than the values conveniently entered through the standard Function Inputs interface. This is particularly useful when validating library functions that accept array-based data, ensuring that the function receives representative input structures during debugging.

Test Script can also be used to compute multiple iterations of the library functions and print each result. A script can invoke a library function repeatedly, vary values between calls, and use print statements to display outputs. This supports efficient validation of repeated calculations, different input combinations, and returned values without changing the deployed library-function logic. The debugger is therefore a controlled test harness for executing BML library-function calls and inspecting their behavior. Oracle CPQ documentation describes BML development and administration capabilities in the [Oracle CPQ documentation library](#); Oracle's broader CPQ product documentation is also available through [Oracle CPQ](#).

#### QUESTION NO: 7

Identify four types of attributes that can be set by formulas. (Choose four.)

- A. Integer Attributes
- B. Single Select Menu Attributes
- C. Date Attributes
- D. Text Attributes
- E. Boolean Attributes
- F. Currency Attributes

**ANSWER: A C E F**

#### Explanation:

Oracle CPQ formulas can calculate and assign values to supported scalar attribute data types used in configuration rules. **Integer Attributes** are formula-settable because formulas can return whole-number results for quantities, counts, dimensions, and similar numeric configuration inputs. **Date Attributes** can receive date results, enabling formulas to derive dates from configuration logic and date calculations. **Boolean Attributes** are formula-settable when formula logic evaluates to a true/false outcome; this supports conditional behavior and rule-driven selections. **Currency Attributes** can be set from formula calculations for monetary values, such as calculated costs, adjustments, or amounts associated with configured products.

These attribute types correspond to values formulas can produce and assign in the configurator. Formula design should ensure that the returned result matches the target attribute's data type, particularly for date and currency values, so that downstream configuration, pricing, and display behavior use the calculated value correctly. Oracle CPQ documentation and product resources are available through the [Oracle CPQ Documentation library](#) and the [Oracle CPQ product page](#).

#### QUESTION NO: 8

A Configuration model contains an array set that is populated by Configuration rules. Users must be able to view the resulting rows, but they must not be able to manually add or remove rows.

Which standard configuration approach should be used?

- A. Hide the control attribute.
- B. Mark every attribute in the array set as forced set.
- C. Hide the array set by using a Configuration hiding rule.
- D. Use CSS to hide the add and delete icons.

**ANSWER: A**

**Explanation:**

**Hide the control attribute** is correct. The control attribute provides the user-facing controls for managing rows in an array set. Hiding this attribute removes the controls that allow users to add or remove rows while retaining the array set and its system-populated rows in the Configurator.

A forced-set setting controls whether a value can be overridden; it does not specifically remove the array-set row controls. CSS-based changes are not the standard CPQ configuration method and do not provide reliable rule-driven behavior.

**QUESTION NO: 9**

Which statement is FALSE about live data tables?

- A. Live data tables do not need to be deployed after modification.
- B. Web services are recommended to update live data tables over BMQL functions.
- C. Insert, Modify, Delete, and Update are supported BMQL functions for live data tables.
- D. Live data tables can be modified using UI, Web Services, BMQL, and Bulk Uploads.

**ANSWER: C**

**Explanation:**

**Insert, Modify, Delete, and Update are supported BMQL functions for live data tables.** is the false statement because `MODIFY` is not a supported BMQL data-manipulation command for a live data table. BMQL supports the SQL-style commands `INSERT`, `UPDATE`, and `DELETE` when working with live data tables, along with querying data through `SELECT`. Consequently, a change to an existing live-data-table row is performed with `UPDATE`; it is not performed through a separate `MODIFY` function. This distinction matters when implementing BML/BMQL logic, because the statement passed to `bmql()` must use the supported command syntax for the operation to execute successfully. Live data tables are intended to support runtime data maintenance, so developers should use the documented DML verbs precisely when inserting, changing, or removing rows. Oracle's CPQ documentation describes data-table administration and the supported integration and scripting mechanisms; see the [Oracle CPQ documentation](#) and the [Oracle CPQ REST API documentation](#).

**QUESTION NO: 10**

A Document Designer template has a default-language view and translated views for several nondefault languages. The document structure must remain common across all languages.

Which two changes can be made in a nondefault language view? (Choose two.)

- A. Change text content.
- B. Modify date and currency formats.
- C. Add an image element that exists only in the translated view.
- D. Add or remove rows and columns from a table element.
- E. Move a text element to a different layout.

**ANSWER: A B**

**Explanation:**

**Changing text content** is supported so that labels, headings, legal text, and other textual elements can be translated for the selected language. **Modifying date and currency formats** is also supported to accommodate locale-specific presentation requirements.

The common document structure is controlled by the default-language design. Adding new images, changing table rows or columns, or moving elements to different layouts would alter the shared structure and is not a nondefault-language-view capability.

### QUESTION NO: 11

Given the following JSON:

```
jsonString=" {\ \"asset-101\":" : {\ \"lines\":" : [ {\ \"documentNumber\":" : \ \"45\"," , \ \"quantity\":" : \ \"3\"," ,  
 \ \"price \":" : 9.1 }, {\ \"documentNumber\":" : \ \"40\"," , \ \"quantity\":" : \ \"30\"," , \ \"price\":" :62}],  
 \ \"finalDate\":" :null, \ \"finalAction\":" : \ \"UPDATE\"," , \ \"available\":"true} }";
```

Which option gets the reference of a JSON object with a specific Document Number '40'?

- A. `jobjLine=jsonpathgetsingle (" $.lines [? (@.documentNmber == '40' )]" , "json", jobjAsset);`
- B. `jobjLine=jsongetsingle (jobjAsset, "$.lines [? (@dicumentNumber== '40')]" , "json");`
- C. `jobjLine=jsonpath.getsingle (jobAsset, "$.lines[? (@.documentNumber== '40')]" , "json");`
- D. `jobjLine=jsonpathgetsingle (jobjAsset, "$.lines[? (@.documentNumber== '40')]" , "json");`

- A. Option A
- B. Option B
- C. Option C
- D. Option D

### ANSWER: D

#### Explanation:

Option D is correct because it uses the appropriate JSON-object selection pattern to obtain the object whose Document Number is 40, rather than merely returning a primitive field value or treating the document number as an array position. The required result is a reference to the matching object itself, so subsequent logic can read or update its properties through that selected object. When JSON data contains a collection of document objects, the selection must evaluate each object's Document Number property against the required value and return the matching object reference. This is particularly important where document numbers are business identifiers: they are not necessarily equivalent to zero-based array indexes, and they may be represented as strings in the JSON payload. The expression shown in Option D performs that property-based lookup and therefore supplies the required object reference for Document Number 40. This follows standard JavaScript collection-query behavior, in which a predicate is evaluated for each element and the matching element can be returned for later use. For background on JSON structures and object access, see [MDN's JSON guide](#). Oracle CPQ implementation resources are available through the [Oracle CPQ documentation library](#).

### QUESTION NO: 12

Which three discount types are available in pricing? (Choose three.)

- A. amount discount
- B. absolute price
- C. promotion discount
- D. quantity-dependent discount

E. percent discount

**ANSWER: A D E**

**Explanation:**

Oracle CPQ pricing supports amount discount, quantity-dependent discount, and percent discount as discount methods that can be applied when calculating a configured item's price. An amount discount reduces the applicable price by a specified monetary value, which is useful when the commercial policy calls for a fixed currency reduction. A percent discount calculates the reduction as a percentage of the relevant price, allowing the discount value to scale with the price of the item or transaction. A quantity-dependent discount supports volume-based commercial policies by associating the discount with the quantity purchased, so the resulting pricing can reflect quantity thresholds or tiers. Together, these methods give pricing administrators the ability to model fixed-value concessions, proportional concessions, and volume-sensitive pricing behavior within the pricing configuration. The Oracle CPQ documentation describes the product's pricing and pricing-administration capabilities, including the configuration of pricing logic and rules. See the [Oracle Configure, Price, and Quote documentation](#) and Oracle's [CPQ product overview](#) for the official product and documentation resources.