DUMPSBOSS.

Microsoft Dynamics 365: Finance and Operations Apps Developer

Microsoft MB-500

Version Demo

Total Demo Questions: 35

Total Premium Questions: 359

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, Plan architecture and solution design	13
Topic 2, Develop new elements by using Application Explorer	54
Topic 3, Read and write basic X++	24
Topic 4, Develop and test code	46
Topic 5, Implement security and optimize performance	42
Topic 6, Integrate with Finance and Operations apps	77
Topic 7, Extend Finance and Operations apps	86
Topic 8, Case Study 1	6
Topic 9, Case Study 2	2
Topic 10, Case Study 3	2
Topic 11, Case Study 4	3
Topic 12, Case Study 5	4
Total	359

QUESTION NO: 1

You are a Dynamics 365 Finance and Operations developer.

You need to create an extension class.

Which action should you perform?

A. Mark the class as final.

B. Add the suffix .extension to the file name.

C. Mark the class as protected.

D. Mark the class as private.

ANSWER: A

Explanation:

Mark the class as final. In Dynamics 365 Finance and Operations, extension classes are implemented as final X++ classes and are associated with the type being extended by using the `ExtensionOf` attribute. For example, a class extension commonly follows the pattern `[ExtensionOf(classStr(SomeClass))]` followed by `final class SomeClass_Extension`. Declaring the extension class as final establishes that the extension itself is not intended to participate in further inheritance, which is the required class declaration pattern for the extension framework.

This pattern supports the platform's extensibility model, including chain of command method wrapping. A method added to the final extension class can wrap an eligible method on the augmented class, allowing custom logic to execute before and/or after the next implementation in the chain. The extension remains a separate customization artifact rather than changing the Microsoft-owned base object, which helps preserve upgradeability. Microsoft's documentation shows extension classes declared with the `final` modifier in its chain of command examples and explains how the `ExtensionOf` attribute identifies the extended artifact. See Method wrapping and Chain of Command and Customization through extensions.

QUESTION NO: 2

You are a Dynamics 365 Finance developer. You make changes to an existing custom class.

The code comparison tool shows version conflicts between your version and the latest checked-in version of the custom class.

In Visual Studio, you open the Source Control Explorer. You locate the latest changeset for the class and you open the changeset in Team Explorer.

You need to compare the latest code that is in source control with the code that you created.

Which option should you use?

A. Compare with Workspace Version

B. Compare with Previous Version

C. View History

D. Compare with Latest Version

ANSWER: A

Explanation:

Compare with Workspace Version is the appropriate command because the workspace contains the developer's local, created version of the custom class. From the changeset details in Team Explorer, this command compares the file revision represented by the selected changeset with the corresponding version in the current Visual Studio workspace. Because the selected changeset is the latest checked-in revision, the comparison directly shows the differences between the latest

source-controlled code and the developer's local edits. This is particularly useful for resolving the reported version conflict: the comparison view identifies the changes in each version so the developer can determine how to merge or reconcile the local modifications with the current server version before checking in. In TFVC, workspace versions are the local files mapped into the active workspace, whereas a changeset identifies a server-side version of files. Microsoft documents the ability to compare a server version with a workspace version through the TFVC comparison experience. See Compare files in TFVC and Use Source Control Explorer.

QUESTION NO: 3

You are developing a form for Dynamics 365 Finance.

You need to add an Action Pane button that opens a custom form.

Which type of menu item should you create?

A. A display menu item

B. An action menu item

C. An output menu item

D. A URL menu item

ANSWER: A

Explanation:

A **display menu item** is the appropriate menu-item type because display menu items open forms. The menu item references the target form and can be assigned to an Action Pane button through the button's menu-item properties.

Using a display menu item also provides a reusable navigation target and allows access to be controlled through the standard security framework. An action menu item is used to invoke a class, while an output menu item is used to run a report. For more information, see Menu items and Action controls.

QUESTION NO: 4

You are training a new Dynamics 365 Finance developer.

You need to recommend where asynchronous integrations should be used instead of synchronous integrations.

For which scenario should you recommend asynchronous integration?

A. A warehouse wants to track movement of all inventory from scanners to the system.

B. A retailer wants to ensure gift card balances are communicated back to the system from point of sale (POS) terminals in near real-time.

C. A retailer requires all new customer data captured at point of sale (POS) terminals through the day sent back to the system.

D. When products are updated in Finance and Operations, a third-party application that contains the same product information needs to also be updated in near real-time.

ANSWER: C

Explanation:

A retailer requires all new customer data captured at point of sale (POS) terminals through the day sent back to the system is the appropriate asynchronous integration scenario. This requirement describes data that can be accumulated during business operations and transferred in batches, rather than requiring Finance and Operations to immediately process each customer record while the POS transaction is occurring. Asynchronous integration decouples the source system from the

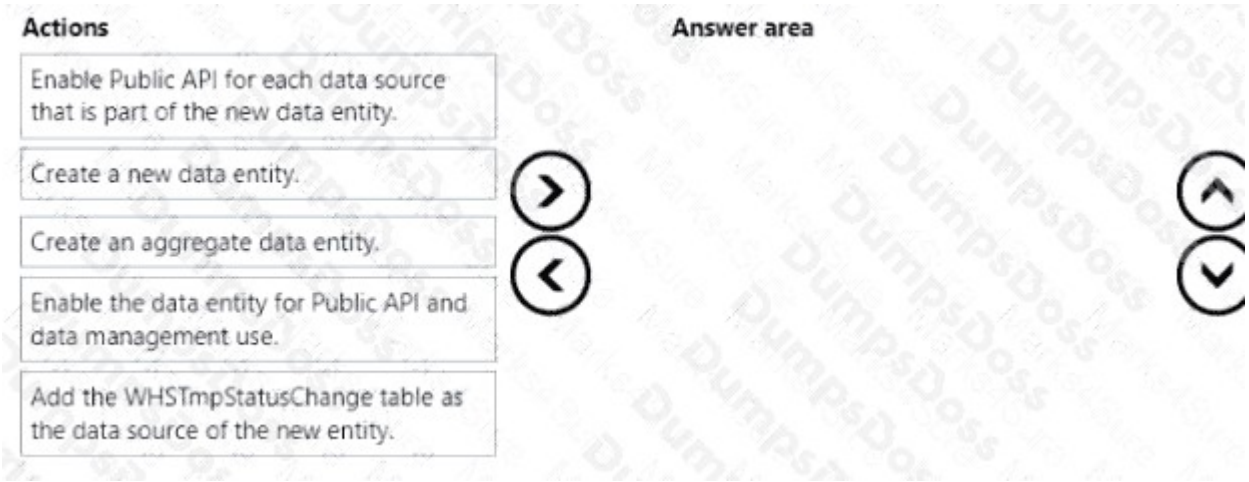
destination system: the POS process can continue capturing customers, and the records can be submitted and processed later through a scheduled or queued mechanism. This pattern is well suited to higher-volume data exchange and helps tolerate temporary connectivity or service availability issues without blocking store activity. In Finance and Operations, data-management capabilities and recurring data jobs support asynchronous, batch-oriented exchange of entity data. Microsoft guidance distinguishes this from synchronous patterns, which are intended when the calling process needs an immediate response from Finance and Operations before it can proceed. Sending the customer data collected throughout the day does not impose that immediate-response requirement, so asynchronous processing is the recommended design. See Microsoft's synchronous versus asynchronous integration guidance and Data management package API documentation.

QUESTION NO: 5 - (DRAG DROP)

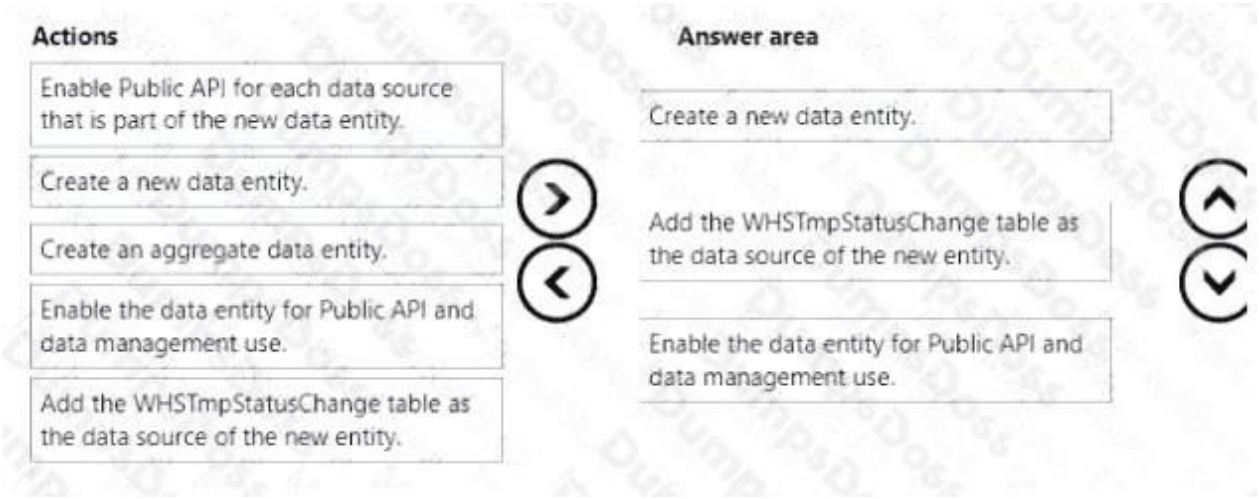
You are working on the data structures for the company's implementation.

You need to ensure that the data from the source system is mapped into the new staging table and can be used for synchronous and asynchronous integrations.

Which three actions should you perform in sequence? To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.



ANSWER:



Explanation:

Start by creating a new data entity. A data entity is the integration-facing representation of application data in Dynamics 365 Finance and Operations, and it provides the structure that external consumers and the Data management framework use. Once the entity exists, add the *WHSTmpStatusChange* table as the data source of that entity. This establishes the staging table as the entity's backing source and allows the appropriate table fields to be represented and mapped in the data entity.

After the entity has a defined data source, enable the data entity for Public API and data management use. Enabling Public API support exposes the entity for synchronous integration scenarios, including OData endpoints. Enabling data management support makes the same entity usable by asynchronous data import and export processes, such as projects handled by the Data management framework. Using one properly configured data entity therefore provides a consistent integration contract over the staging data for both required interaction patterns.

This sequence is important because exposure settings apply to the completed entity definition, including its underlying source and field mapping. Microsoft's documentation describes data entities as the abstraction used for integration and data management, and describes OData as the public REST endpoint for enabled entities. See Data entities overview, Build and consume data entities, and Open Data Protocol (OData).

QUESTION NO: 6

A company uses Dynamics 365 Finance.

Users report that the performance of the open vendor invoice form in the production environment takes longer than usual to load. When users test the invoice form in the sandbox environment the performance is fine.

You need to troubleshoot the performance issue.

Which two tools can you use? Each correct answer presents a complete solution. Choose two.

NOTE: Each correct selection is worth one point.

A. Visual Studio debugger

B. Windows Task Manager

C. Trace parser

D. Environment monitoring in Lifecycle Services

ANSWER: C D

Explanation:

Trace parser is appropriate because it enables detailed analysis of a captured client and server trace for a specific user scenario, such as opening the vendor invoice form. The trace can expose the elapsed time of form initialization, X++ methods, database calls, service activity, and other operations involved in the request. This level of timing information helps a developer isolate the component responsible for the slower production behavior and compare the behavior with a healthy environment when necessary. Microsoft documents trace capture and analysis procedures in Trace Parser.

Environment monitoring in Lifecycle Services is also appropriate for a production-only issue. Lifecycle Services provides environment-level telemetry and monitoring data that can be used to investigate system health and performance conditions affecting users, including activity, errors, and resource-related trends. Reviewing this data for the period in which users experience slow form loading helps determine whether the issue correlates with production load, platform events, database activity, or other environmental conditions. Microsoft describes the available monitoring capabilities in Monitoring and diagnostics in Lifecycle Services.

QUESTION NO: 7

A company has a cloud-based installation of Dynamics 365 Finance.

A user reports slow response times in the application when accessing the production environment.

You need to diagnose the performance issue by using the Environment monitoring tool.

Which two features of the Environment monitoring tool should you use?

NOTE: Each correct answer presents part of the solution.

A. Customization analysis

B. System diagnostics

C. SQL insights

D. Health metrics

E. Availability monitoring

ANSWER: C D

Explanation:

SQL insights and **Health metrics** are the appropriate Environment monitoring features for investigating slow interactive response in a production Finance environment. SQL insights provides visibility into database activity and query performance. It helps an administrator identify expensive or long-running SQL queries, evaluate query duration and execution characteristics, and determine whether database workload is contributing to the reported application delay. This makes it particularly useful when the slowness occurs during operations that retrieve, calculate, or update substantial amounts of business data.

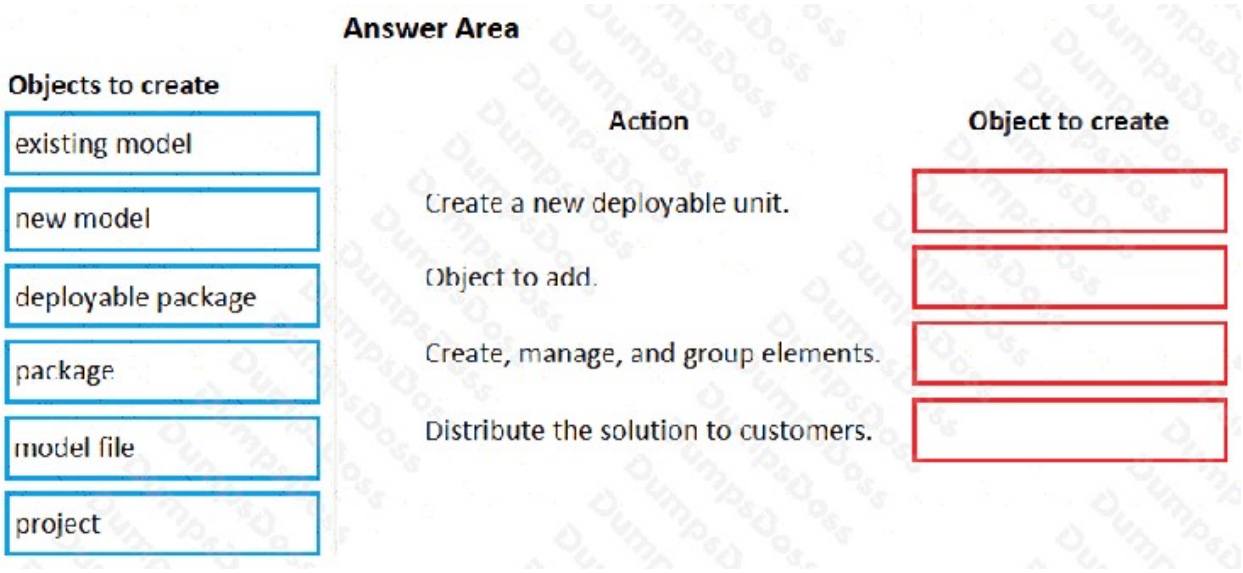
Health metrics provides the broader environment-level telemetry needed to correlate the user's experience with platform conditions. The metrics can reveal resource and service trends over the selected period, enabling investigation of conditions that affect application responsiveness, such as sustained load or unhealthy performance patterns. Reviewing these trends alongside the SQL workload helps distinguish a database-query issue from a more general environment health condition and establishes the time window for deeper analysis. Microsoft describes the available production monitoring and diagnostic capabilities in Monitoring and diagnosing issues in finance and operations apps and documents environment monitoring through Environment monitoring.

QUESTION NO: 8 - (SIMULATION)

You need to develop, test, and deploy the Vendor Exclusion list solution.

What should you create? To answer, drag the appropriate objects to the correct actions. Each element may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

NOTE: Each correct selection is worth one point.



ANSWER: See the explanation for the answer

Explanation:

Drag the objects as follows:

A package is the deployable/compilation unit that contains one or more models. Create a new model in the package, use a project to organize and manage its development elements, and then create a deployable package to distribute and deploy the completed solution.

QUESTION NO: 9

You have a table in which multiple properties must be changed. Multiple table properties are locked down at

the base table and must not be overwritten.

You need to modify the table properties by extending the table.

Which table property can you populate in a table extension by using the property sheet?

A. Primary index

B. Configuration key

C. Save data per company

D. Modified date time

ANSWER: D

Explanation:

Modified date time is a table property that can be populated through the property sheet on a table extension. This supports extension-based customization without overlayering or directly changing the base-table metadata. When the ModifiedDateTime property is enabled, Finance and Operations maintains modification timestamp information for records in the table. This provides a standard system-managed audit value indicating when a record was most recently changed, rather than requiring custom code to set and maintain a timestamp.

Table extensions are the appropriate mechanism when a developer needs to add supported metadata to an existing application table while preserving the base object and maintaining upgradeability. The property-sheet support is important because it applies the metadata change through the extension model, so the customization remains isolated from Microsoft or other publisher code. This approach aligns with the platform extensibility model and allows the table to participate in standard record-tracking behavior while retaining the base table as delivered.

For the supported metadata changes available through extensions, see Modify properties through extensions. For background on table design and system fields, see Tables and data in X++.

QUESTION NO: 10

A developer submits code for a quality assurance review. Several lines of code use the var data type.

You need to validate the segments tor the correct X++structure.

Which code statement will pass validation?

A. var var1 = systemDateGet (); var1 = var1 ? today(); "No today";

B. var var1 = true ? 10:"10"

C. var var1 = true; var1 = true ? 10:falase;

D. var var1 = (var1 >= false) ? true :10;

E. var var1 = systemDateGet(); var1 = today();

ANSWER: E

Explanation:

`var var1 = systemDateGet(); var1 = today();` passes validation because X++ infers the type of a `var` variable from its initializing expression. `systemDateGet()` returns a value of the X++ `date` type, so `var1` is established as a date variable at declaration time. The subsequent assignment uses `today()`, which also returns a date value. Therefore, the variable's inferred type and the assigned value's type remain consistent.

Type inference with `var` does not create an untyped or dynamically changing variable. It provides a concise declaration while preserving compile-time type checking. Consequently, developers should initialize a `var` variable with an expression whose type is known and subsequently assign values that are compatible with that inferred type. Using date-returning functions for both the declaration and later assignment is a clear, valid X++ pattern. Microsoft's X++ documentation describes the available primitive data types, including `date`, and the X++ language reference provides the relevant syntax and development guidance: X++ data types and X++ language reference.

QUESTION NO: 11

You are training a new Dynamics 365 Finance developer.

You need to explain the relationships between models, packages, and projects to the new hire.

Which three design concepts should you explain? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

A. A project can contain elements from multiple models.

B. A model is a group or collection of elements that constitute a distributable software solution.

C. A Visual Studio project can belong to more than one model.

D. A model is a design time concept.

E. A package is a deployment unit that may contain one or more models.

ANSWER: B D E

Explanation:

A model is a design-time construct used to organize application artifacts for a functional area or a custom solution. It groups related metadata and source elements, such as tables, classes, forms, reports, and extensions, into a coherent software solution. This organization lets developers identify the ownership and purpose of application elements while working in the Dynamics 365 Finance and Operations development environment.

A package is the higher-level compilation and deployment boundary. It can contain one or more models and is used to build, distribute, and deploy the compiled application components together. Packages also establish dependencies between application components, which enables the platform to resolve referenced functionality during development, compilation, and runtime.

Therefore, the key concepts are that a model is a group or collection of elements that constitute a distributable software solution, that a model is a design time concept, and that a package is a deployment unit that may contain one or more models. These distinctions are foundational when creating extensions, managing source code, and planning deployable packages. For further details, see Models in Dynamics 365 Finance and Operations and Packages in Dynamics 365 Finance and Operations.

QUESTION NO: 12

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You create a new form in a project.

You need to display tabs in a vertical alignment.

Solution: Apply the Simple List pattern.

Does the solution meet the goal?

A. Yes

B. No

ANSWER: B

Explanation:

No is correct. The Simple List form pattern is intended for displaying and maintaining a straightforward collection of records in a list or grid. It provides the standard structure for a list page, including the action pane and a primary grid, but it is not the pattern that supplies a vertically arranged tab navigation experience. Therefore, applying this pattern does not meet a requirement specifically to display tabs vertically.

In Dynamics 365 Finance and Operations, a form's design must use a pattern that supports the required navigation and layout structure. Patterns are not merely cosmetic templates; they prescribe supported control arrangements and interaction behavior. When the intended experience is navigation through sections by using vertically presented tabs, the developer must select a form pattern designed for that navigation model rather than use a simple record-list pattern. Applying the correct pattern also helps ensure the form complies with the platform's user-interface guidelines and can pass pattern validation during development.

Microsoft documents the purpose and required structure of the Simple List pattern in its Simple List form pattern guidance. For the broader pattern-selection and implementation guidance, see Form patterns.

QUESTION NO: 13

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution. Determine whether the solution meets the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

A company uses Dynamics 365 finance and operations apps.

You have a custom enumeration named CarType. The enumeration has the following elements: Sedan, SUV.

You must extend CarType and add a new element named MUV to CarType.

You need to develop a solution that meets the requirements.

Solution: Set the Is Extensible property to true for the CarType enumeration. Create a new enumeration to add the MUV element.

Does the solution meet the goal?

A. Yes

B. No

ANSWER: B

Explanation:

No is correct because setting *Is Extensible* to *Yes* permits an enumeration to be extended, but the required implementation is an *enum extension* for the existing *CarType* base enumeration. The extension adds *MUV* as another element of *CarType*, so code and metadata that use *CarType* can recognize the new value alongside *Sedan* and *SUV*. Creating a separate enumeration would define a different type rather than extending *CarType*; consequently, its element would not become a member of the required enumeration. In Dynamics 365 Finance and Operations, extensions are implemented in a separate extension object and retain the base object's identity, which supports customization without overlayering. For this scenario, the correct development approach is to enable extensibility on the base enum when necessary and create an enum extension that declares the *MUV* element. Microsoft's extensibility guidance describes extending enums and the requirement for an enum to be marked extensible: Extend enums. Additional information about the model-based extension architecture is available in Extensions overview.

QUESTION NO: 14

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You are adding a new field to the SalesTable form.

You must use an extension to add a status field onto the form.

You need to create the extension in the Application Object Tree (AOT) and add the extension to the demoExtensions model.

Solution: Add a new form object to the project and name the form SalesTable.Extension.

Does the solution meet the goal?

A. Yes

B. No

ANSWER: B

Explanation:

No is correct because an extension must be created as a form extension of the existing `SalesTable` form, rather than as an independent new form object whose name happens to include `.Extension`. In Visual Studio, the supported process is to locate the existing form in Application Explorer, right-click it, and select **Create extension**. This creates extension metadata that is explicitly associated with the base form and allows custom controls, data sources, and other permitted elements to be added without overlayering the Microsoft object. The resulting extension can then be included in the `demoExtensions` model.

A status field also normally requires the underlying field to exist on a table extension, such as an extension of `SalesTable`, before it can be exposed through a form extension. Creating a separate form does not extend the existing sales order form that users open, and it does not establish the required extension relationship in the AOT. Microsoft's extensibility model uses extensions to preserve upgradeability and isolate customizations from the base application. See Customize model elements by using extensions and Create table extensions.

QUESTION NO: 15

You are a Dynamics 365 Finance developer.

You need to deploy a new report in a development environment.

From which two locations can you deploy the report? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

A. Package deployment

B. Build project

C. Application Explorer

D. Solution Explorer

E. Build Models options

ANSWER: C D

Explanation:

Application Explorer and **Solution Explorer** are the two Visual Studio locations from which a developer can deploy a Dynamics 365 Finance report in a development environment. In Solution Explorer, deployment is initiated from the report project, which is useful when working directly with the project and its report artifacts. This publishes the report definition and associated items to the development environment's reporting infrastructure.

Application Explorer also exposes report metadata under the reporting node. A developer can select an individual report there and use the deployment command, providing a metadata-centric workflow that does not require navigating from the project structure. This is particularly convenient when the report is part of an existing model and the developer wants to deploy a specific report artifact.

Both locations support the standard developer workflow for making a newly created or updated report available for execution in the development environment. Microsoft's report development guidance describes using Visual Studio and Application Explorer to work with reporting artifacts, while the deployment guidance explains how Finance and Operations application artifacts are packaged and deployed across environments. See Create reports by using SQL Server Reporting Services and Create and apply deployable packages.

QUESTION NO: 16

You need to configure security for the Vendor Exclusion List report.

What are two possible ways to achieve this goal? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

A. Create a new privilege, add a reference to an output menu item, and then add the new privilege to a role.

B. Create a new security policy and add an output menu item. Add the new security policy to duty extension and then to a role extension.

C. Create a new privilege extension and add an output menu item. Add the privilege to a duty extension and then to a role extension.

D. Create a new privilege and assign it to an output menu item. Add the new privilege to a duty extension and then add the duty extension to a role.

ANSWER: A D

Explanation:

A new privilege is the appropriate security artifact for granting access to a new report entry point. The privilege includes the required permission for the output menu item that launches the Vendor Exclusion List report. The privilege can then be assigned directly to a security role, making this a complete way to grant report access to the users represented by that role.

It is also valid to place the new privilege in a duty and grant that duty through a role. In an extension-based customization, a duty extension can add the privilege and a role extension can add the duty. This approach is especially suitable when the report permission belongs to a defined business responsibility and must be incorporated into an existing security model without overlayering Microsoft application objects.

Dynamics 365 Finance security is structured around roles, duties, privileges, and permissions. Privileges provide access to securable entry points, while duties and roles group permissions into business-appropriate assignments. Security role and duty extensions are specifically supported for adding custom security artifacts. See Security role and duty extensions and Security architecture for Finance and Operations apps.

QUESTION NO: 17

A company uses Dynamics 365 Finance.

The company deploys a custom table named LoyaltyCustomerTable. The table has a 1:1 relationship with a table named CustTable. Both tables contain a unique field named AccountNum and share the same data. Using event handlers, all edits to CustTable are reflected within LoyaltyCustomerTable.

The LoyaltyCustomerTable data is maintained by using the LoyaltyCustomerTable form. This form selects all data from both LoyaltyCustomerTable and CustTable and displays the records sorted by LoyaltyCustomerTableAccountNum.

LoyaltyCustomerTable was mistakenly created with no indexes. A significant delay occurs when users open the LoyaltyCustomerTable form.

You need to improve performance by creating an index.

Which method should you use?

A. Create an index on CustTable for Redd.

B. Create an index on LoyaltyCustomerTable for AccountNum.

C. Create a foreign key relation on CustTable with Related Table set to LoyaltyCustomerTable and Index set to RecId.

D. Create a foreign key relation on CustTable with Related Table set to LoyaltyCustomerTable and Index set to AccountIdx.

ANSWER: B

Explanation:

Create an index on LoyaltyCustomerTable for AccountNum. The form's query reads LoyaltyCustomerTable together with CustTable and orders the displayed result by the AccountNum field from LoyaltyCustomerTable. Because the custom table has no indexes, SQL Server may need to scan the table and perform a costly sort before the form can display its records. An index whose key is AccountNum gives the database an ordered access path for the table data used by the form. This is particularly appropriate because AccountNum is unique for the stated one-to-one data design, so the index can efficiently support locating the matching record and returning rows in the requested order. In Dynamics 365 Finance and Operations, table indexes are defined in the table metadata and are synchronized to the underlying database; they are an essential part of designing performant queries, joins, and sorts. The index should be added beneath the custom table's Indexes node, with AccountNum included as its indexed field, then the project should be built and database synchronized so that the SQL index is created. For Microsoft guidance on defining table indexes and their role in data-model performance, see Indexes and Table browser.

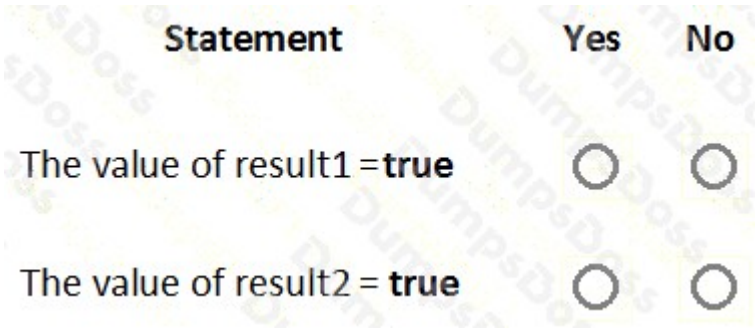
QUESTION NO: 18 - (HOTSPOT)

You have the following code:

```
public void CompareValues()
{
  Date myDate = str2date('01/08/1901',123);
  boolean result1, result2, result3;
  result1 = (myDate + 1 == str2date('02/08/1901', 123));
  result2 = (int2str(real2int(myDate)) + 1 == str2date('02/08/1901', 123));
}
```

For each of the following statements, select Yes if the statement is true. Otherwise, select No.

NOTE: Each correct selection is worth one point.



Statement	Yes	No
The value of result1 = **true**	○	○
The value of result2 = **true**	○	○

ANSWER:

Explanation:

The first statement is true because X++ supports date arithmetic with an integer offset. When an integer is added to a `Date` value, the integer represents a number of days. Therefore, adding `1` to `myDate` advances it by exactly one calendar day. The resulting date matches the date created from `02/08/1901` using the same date-sequence parameter, so the equality comparison assigned to `result1` evaluates to `true`. The correct selection for the first statement is Yes.

The date value in the second statement cannot be converted by calling `real2int` directly. That function is for converting a `real` value to an integer; it is not a date conversion function. Date values are a distinct X++ primitive type and should be processed through supported date operations or appropriate type-specific conversion logic. As a result, the expression shown for `result2` cannot establish the stated true result. The correct selection for the second statement is No. Microsoft documents the X++ primitive types and conversion functions in the X++ data types reference and the available language features in the X++ language programming guide.

QUESTION NO: 19

You are developing an extension for Dynamics 365 Finance.

You need to run custom logic after a record is inserted into the CustTable table.

You decide to use a table data event handler.

Which two actions should you perform? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

A. Create a class that contains the event handler method.

B. Decorate a static method with the DataEventHandler attribute for CustTable and DataEventType::Inserted.

C. Modify the CustTable insert method directly.

D. Create an output menu item for CustTable.

E. Set the AllowEdit property of CustTable to Yes.

ANSWER: A B

Explanation:

Create a class that contains the event handler method. Event handlers are implemented in separate classes so that the extension logic remains isolated from the base table artifact.

Decorate a static method with the `DataEventHandler` attribute and specify `tableStr(CustTable)` and `DataEventType::Inserted`. The attribute subscribes the method to the inserted event for CustTable. The framework invokes the handler after a CustTable record is inserted.

Data events provide an extensibility mechanism for reacting to table operations without overlayering table methods. For more information, see Customize model elements by using extensions and Events and event handlers.

QUESTION NO: 20

You create a custom data entity for an external application.

You need to expose the entity through the Dynamics 365 Finance OData endpoint.

Which two data entity properties should you configure? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

A. Set IsPublic to Yes.

B. Set PublicEntityName.

C. Set AllowEdit to No.

D. Set CacheLookup to EntireTable.

E. Set TableType to TempDB.

ANSWER: A B

Explanation:

Set the **IsPublic** property to Yes. This marks the data entity as public and makes it eligible for exposure through the OData endpoint after the entity metadata is refreshed.

Set the **PublicEntityName** property to the name that external consumers use in OData requests. The public entity name identifies the entity in the OData service metadata and URL path. Together, these properties provide the entity's external OData exposure and name.

For more information, see Open Data Protocol (OData) and Data entities overview.

QUESTION NO: 21

An organization has two million customers that are part of the International customer group.

Validation must occur when customer records are updated. For all customers where the value of the customer group field is international, you must the delivery mode to Air.

You need to update the customer records.

Which two code segments can you use? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

A.
```
ttsbegin;
while select forupdate CustTable
  where CustTable.CustGroup == 'International'
{
  CustTable.DlvMode = 'Air';
  CustTable.update();
}
ttscommit;
```

B.
```
ttsbegin;
update CustTable
set DlvMode = 'Air'
where CustTable.Custgroup == 'International';
ttscommit;
```

C.
```
ttsbegin;
update_recordset CustTable
  where CustTable.Custgroup == 'International'
  setting DlvMode = 'Air';
ttscommit;
```

D.
```
ttsbegin;
while select forupdate CustTable
  where CustTable.CustGroup == 'International'
{
    CustTable.DlvMode = 'Air';
    CustTable.doUpdate();
}
ttscommit;
```

E.
```
ttsbegin;
update_recordset CustTable
  setting DlvMode = 'Air'
  where CustTable.Custgroup == 'International';
ttscommit;
```

A. Option A

B. Option B

C. Option C

D. Option D

E. Option E

ANSWER: D E

Explanation:

The two selected code segments provide complete ways to update all customer records whose customer group is International and assign the delivery mode value Air. A record-by-record approach selects the matching records for update, changes the delivery mode on each buffer, and calls `update()` inside a transaction. This is appropriate when the update must use the table update pipeline, including validation and any business logic implemented through table methods.

A set-based `update_recordset` operation is also designed to update a field for every record satisfying a `where` clause. It is especially appropriate for a population of two million customers because it lets the database execute one set-oriented update rather than requiring application code to fetch and update each row individually. Finance and Operations can fall back to row-by-row processing for set-based operations when required by table behavior, preserving the applicable data-method processing unless it has explicitly been skipped. The selection condition must target the International customer group, and the assignment must set the delivery mode to Air.

For Microsoft guidance on X++ set-based statements and the situations in which they fall back to record-by-record behavior, see X++ data performance. Transaction handling for record updates is documented in X++ transactional integrity.

QUESTION NO: 22

You use the performance timer to monitor and optimize Dynamics 365 performance.

You need to view information about how many users are currently using the application.

Which performance counter should you use?

A. Services Session Provider

B. Web Client Session

C. Forms

D. GC

ANSWER: B

Explanation:

Web Client Session is the appropriate performance-counter category for viewing the number of users currently using the Dynamics 365 Finance and Operations application through the web client. The Performance timer provides multiple counter categories that expose operational information for different application tiers and subsystems. Web-client session counters are specifically intended to show session and user activity associated with browser-based application usage, which is the relevant measure when determining current user load.

Monitoring this information helps administrators and developers establish the active workload on an environment while investigating responsiveness, validating performance tests, or correlating usage volume with resource consumption. Because users normally access Finance and Operations through the browser-based client, the session information exposed under **Web Client Session** provides the directly applicable view of concurrent application use. This makes it useful for monitoring trends and evaluating whether increased user activity coincides with slower page loads, server pressure, or other performance symptoms.

Microsoft documents the available Performance timer counters and their use for analyzing Finance and Operations performance in the Performance timer documentation. Additional guidance for performance testing and monitoring is available in Performance testing.

QUESTION NO: 23

A company uses Dynamics 365 Finance.

You have an entity named VendVendorV2Entity. You add a new column to the entity.

You need to ensure that the new column is available in the entity for data transfer activity.

Which two actions should you perform? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

A. Select the Generate mapping button on the Mapping details form.

B. Select the Validate button on the Entity list page.

C. Select the Entity structure button on the Entity list page.

D. Select the Modify target mapping button on the Entity list page.

ANSWER: A B

Explanation:

Selecting *Select the Validate button on the Entity list page.* validates the data entity's metadata for Data management. Validation checks the entity against the metadata that is available after the entity-list refresh process, including configuration-key-related metadata. This confirms that the changed entity is valid for use by the Data management framework and that its data-source and field metadata can be used for transfer operations.

Selecting *Select the Generate mapping button on the Mapping details form.* then generates the staging-to-target field mapping used for import and export. Because the new column is now part of the entity definition, regenerating the mapping incorporates that field into the entity mapping available to data projects. Together, validation and mapping generation make the updated entity metadata usable for data transfer activity rather than retaining an earlier mapping definition.

Microsoft describes data entities as the abstraction used for data import and export, and documents the Data management entity-list actions for validating entities and generating mappings. See Data entities overview and Data management overview.

QUESTION NO: 24

You are a Dynamics 365 Finance developer.

You must extend the validate method of the SalesLine table by using Chain of Command. The value of a variable named SalesPrice must be greater than or equal to zero when adding new lines.

You need to create an extension class.

Which two code segments can you use? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

A.

```
[ExtensionOf(tableStr(SalesLine))]
final class SalesLine_Extension
{
  boolean validateWrite(boolean _skipCreditLimitCheck)
  {
    boolean ret;
    ret = next validateWrite(_skipCreditLimitCheck);
    if (ret && this.SalesPrice < 0)
    {
      ret = false;
    return ret;
  }
}
```

B.

```
[ExtensionOf(tableStr(SalesLine))]
final class SalesLine_Extension
{
  boolean validateWrite(boolean _skipCreditLimitCheck = false)
  {
    boolean ret;
    ret = next validateWrite(_skipCreditLimitCheck);
    if (ret && this.SalesPrice < 0)
    {
      ret = false;
    return ret;
  }
}
```

C.
```
[ExtensionOf(tableStr(SalesLine))]
public class SalesLine_Extension
{
  boolean validateWrite(boolean _skipCreditLimitCheck)
  {
    boolean ret;
    ret = next validateWrite(_skipCreditLimitCheck);
    if (ret && this.SalesPrice < 0)
    {
      ret = false;
    return ret;
  }
}
```

D.
```
[ExtensionOf(tableStr(SalesLine))]
final class SalesLine_Extension
{
  boolean validateWrite(boolean _skipCreditLimitCheck)
  {
    boolean ret;
    try
    {
       ret = next validateWrite(_skipCreditLimitCheck);
    }
    catch
    {
      ret = false;
    }
    if (ret && this.SalesPrice < 0)
    {
      ret = false;
    }
    return ret;
  }
}
```

A. Option A

B. Option B

C. Option C

D. Option D

ANSWER: C D

Explanation:

Option C and Option D are correct because each provides a valid Chain of Command implementation for the SalesLine validation logic and preserves the standard table-validation behavior. A table extension class must be decorated with `ExtensionOf` for the SalesLine table, and the wrapper method must match the signature of the method being wrapped. The validation must evaluate the new line's `SalesPrice` value and return a failed validation result when that value is below zero. This prevents persistence of an invalid SalesLine while allowing the normal validation process to continue for valid values.

Crucially, a Chain of Command wrapper calls the base implementation through `next`. This ensures that standard Finance validation and any other extensible validation behavior are retained rather than replaced. The wrapper can combine the result from standard validation with the custom nonnegative-price rule and return the final Boolean result. This is the supported extensibility approach for augmenting application logic without overlayering Microsoft code. See the Microsoft guidance for method wrapping and Chain of Command and the table-validation guidance in validations and defaults.

QUESTION NO: 25 - (HOTSPOT)

HOTSPOT

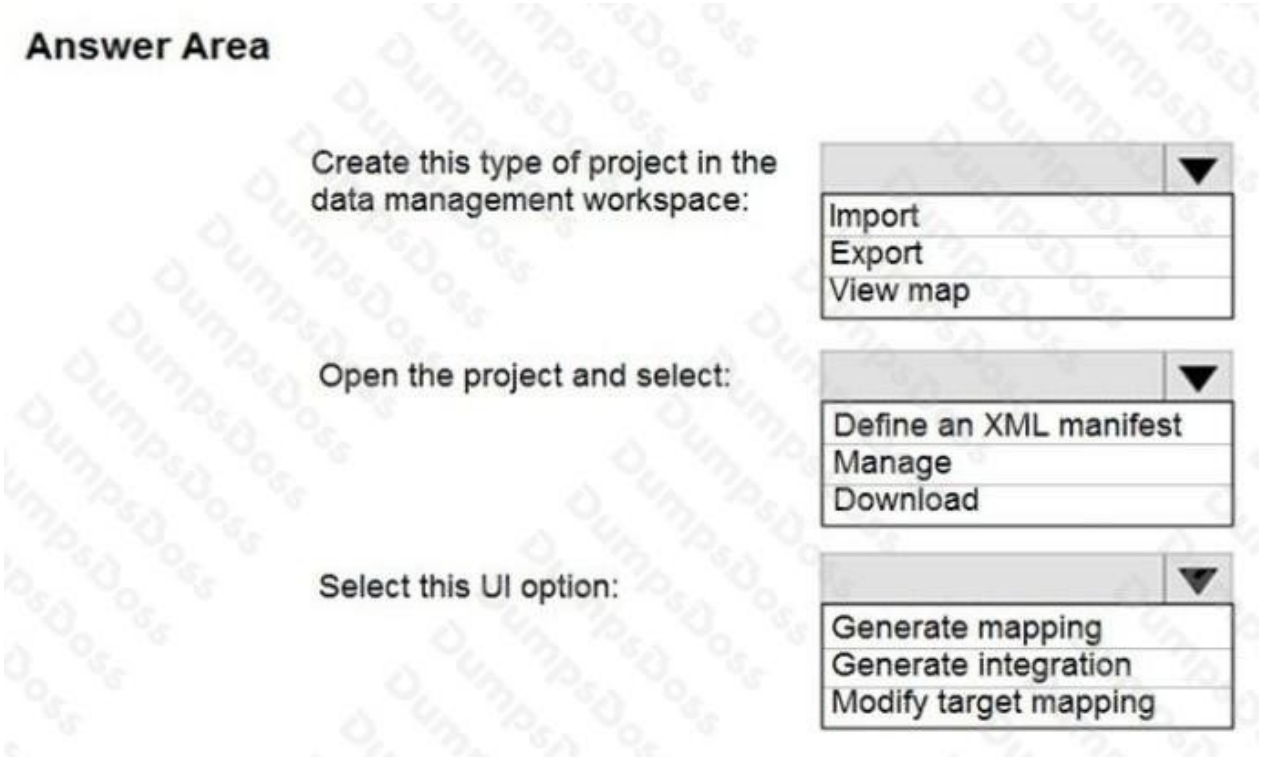
A company plans to migrate purchase orders from an older system into Dynamics 365 Finance.

You need to generate field mapping between the source and the target data structures.

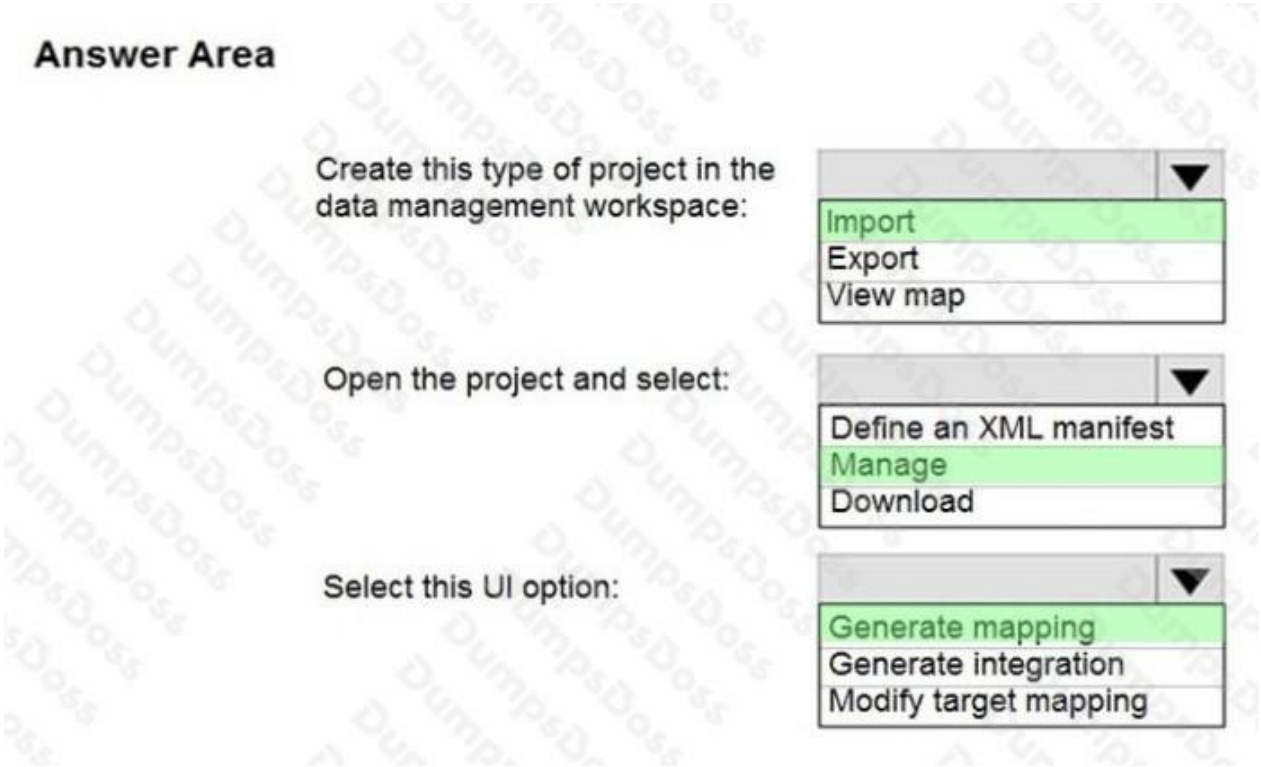
What should you do? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Hot Area:



ANSWER:



Explanation:

Use an **Import** project in the Data management workspace because the purchase-order data is moving from an older external system into Dynamics 365 Finance. An import project provides the target data entities and processing framework used to load inbound data into Finance. After the project is created and its entities are configured, open the project and select **Manage**. This area contains the project-level capabilities for working with the data package, mappings, and related import artifacts.

From the Manage actions, select **Generate mapping**. This creates the field mapping between the source data structure and the target data entity structure used by the project. The generated mapping provides a starting point for translating inbound purchase-order fields to the appropriate Finance entity fields, allowing the integration to process the source data in the format required by the target entity. The mapping can subsequently be reviewed and adjusted as needed for the legacy system's specific field names and data layout.

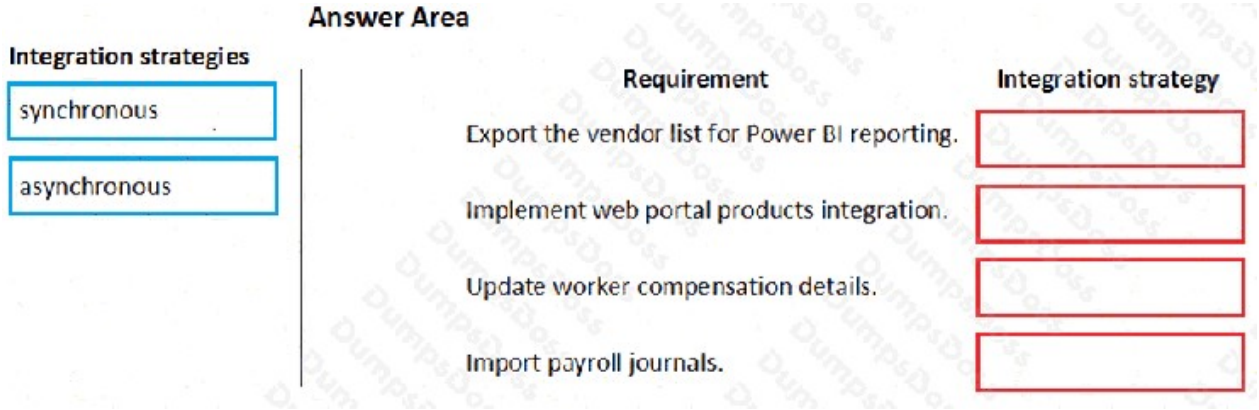
This workflow aligns with the Data management framework's import-project process: define an import project, add/configure the data entity, and generate the mapping used for incoming files or packages. Microsoft's documentation on data management overview describes the framework for importing data through projects and data entities. See also the data import and export jobs documentation for the project-based import workflow.

QUESTION NO: 26 - (SIMULATION)

You need to implement the companyâ€™s integration requirements.

Which integration strategies should you use? To answer, drag the appropriate integration strategies to the correct requirements. Each integration strategy may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

NOTE: Each correct selection is worth one point.



ANSWER: See the explanation for the answer

Explanation:

Assign the integration strategies as follows:

Use synchronous integration when the calling application requires an immediate request/response result, such as retrieving products for a web portal or confirming a compensation update. Use asynchronous integration for reporting exports and high-volume/batch-oriented journal imports, where the caller does not need to wait for processing to finish.

QUESTION NO: 27 - (DRAG DROP)

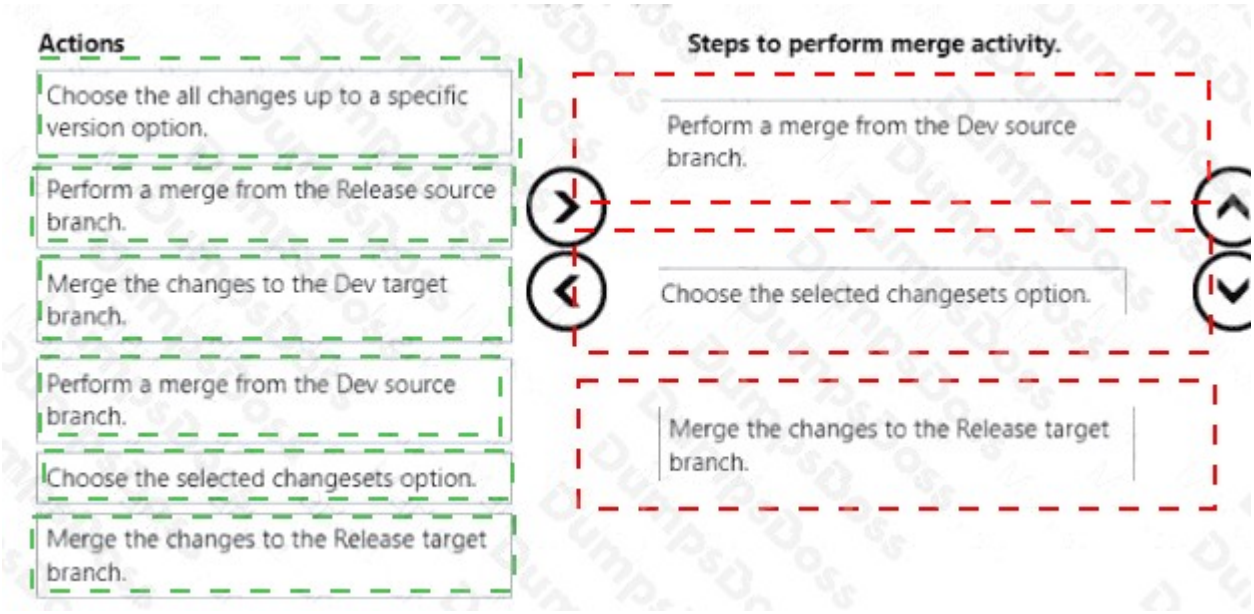
You must merge the code change that you performed and which was tested by QA to the production environment.

You need to merge a specific changeset.

Which three actions should you perform in sequence? To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.

ANSWER:

Answer:



Answer:

Explanation:

To promote a single QA-tested change to production, begin the merge from the Dev branch because it contains the validated development work that must be moved forward. In a branch-based lifecycle, the Release branch is the controlled branch used for production-ready code, so the merge direction needs to be from Dev as the source to Release as the target. This preserves the intended promotion path and keeps the production branch aligned with approved work.

Because the requirement is to merge a specific changeset, select the selected-changesets merge mode. That mode allows the developer to locate and include only the changeset that was tested and approved by QA. It avoids bringing in unrelated work that might also exist in Dev but has not been approved for production. After identifying that specific changeset, complete the operation by merging it to the Release target branch. The resulting sequence is: perform a merge from the Dev source branch, choose the selected changesets option, and merge the changes to the Release target branch.

This approach matches the standard Azure DevOps TFVC branching workflow for selectively promoting changes between branches. Microsoft's guidance on merging folders and files in TFVC describes selecting source and target branches and choosing the relevant changes to merge. The branch roles and promotion flow are also consistent with the branching concepts described in TFVC branching guidance.

QUESTION NO: 28

D18912E1457D5D1DDCBD40AB3BF70D5D

You are a Dynamics 365 Finance and Operations developer.

You need to deploy a new report in a development environment.

From which two locations can you deploy the report? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

A. Package deployment

B. Build project

C. Application Explorer

D. Solution Explorer

E. Build Models options

ANSWER: A D

Explanation:

Package deployment is a valid deployment method because Finance and Operations deployable packages contain the compiled application artifacts for one or more models, including report artifacts and associated resources. Applying the package to an environment installs those artifacts as part of the application deployment process. This approach is particularly applicable when the report is delivered with other model changes as a packaged build output. Microsoft documents the process for creating and applying deployable packages at Create and apply deployable packages.

Solution Explorer is also a valid location for report deployment during development. A developer can select the report project/report artifact in Visual Studio's Solution Explorer and use the available deployment command to build and deploy the report to the connected development environment. This supports the normal developer workflow of editing report metadata and designs, building the project, and deploying the resulting report artifact for validation. Finance and Operations development uses Visual Studio projects and model-based application artifacts; Microsoft's development environment guidance is available at Access Visual Studio.

QUESTION NO: 29

A company uses Dynamics 365 finance and operations apps.

A customer wants to ensure future customizations or application upgrades do not interfere with or interrupt business process steps. The customer must document the business process steps a user performs in the user interface.

You need to select the options for exporting a task recording for testing.

Which three options could you select? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

A. Generate Test Execution files only

B. Save as developer recording

C. Export as Word document

D. Save to Lifecycle Services

E. Create derived test case

ANSWER: B C D

Explanation:

Save as developer recording, **Export as Word document**, and **Save to Lifecycle Services** are supported Task recorder output options in Dynamics 365 finance and operations apps. A developer recording preserves the recorded user actions in a format that developers can use when reviewing and supporting implementation work. Exporting as a Word document produces human-readable process documentation, including the captured steps and screenshots, which enables business users and test teams to review the expected process and retain it as evidence of the baseline procedure. Saving the recording to Lifecycle Services (LCS) makes it available for Business process modeler-based process documentation and test-case management. From LCS, teams can organize the process into business-process libraries and use it to support regression testing as the implementation changes. Together, these export targets provide both maintainable process documentation and artifacts that support validation of recorded business flows after customizations or upgrades. Microsoft documents Task recorder's available save and export capabilities at Task recorder, and describes managing business processes and test cases in Business process modeler.

QUESTION NO: 30

You are developing a report for Dynamics 365 Finance.

The report process requires temporary data that must support SQL joins, SQL filtering, and transaction processing.

Which table type should you use?

A. TempDB

B. InMemory

C. Regular

D. View

ANSWER: A

Explanation:

You should use a **TempDB** table. TempDB tables are physically created in the SQL Server tempdb database and support SQL-based operations, including joins, filtering, and transaction processing. They are appropriate for report and processing scenarios that need temporary relational data while benefiting from database execution.

An InMemory table is held in application memory and is better suited to smaller temporary datasets that do not require SQL joins. A regular table persists data in the application database and is not intended solely for temporary processing. For more information, see Temporary tables.

QUESTION NO: 31

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You are adding a new field to the SalesTable form.

You must use an extension to add a status field onto the form.

You need to create the extension in the Application Object Tree (AOT) and add the extension to the demoExtensions model.

Solution: Navigate to the user interface forms node for the SalesTable form and modify the form.

Does the solution meet the goal?

A. Yes

B. No

ANSWER: B

Explanation:

No is correct because the requirement explicitly calls for an extension, whereas directly modifying the existing SalesTable form is an overlayering-style change and does not create a form extension in the demoExtensions model. In Dynamics 365 Finance and Operations, customizations to Microsoft-provided application artifacts should be implemented through extensions so that the base object remains unchanged and the customization can be maintained across platform and application updates. The appropriate AOT action is to create an extension of the SalesTable form under the User Interface > Forms node, then add the status control to that extension. If the status field itself is new data that must be stored, it must first be created through a SalesTable table extension; the form extension can then expose that field by adding a control bound to the extended data source field. This approach keeps the customization isolated in the demoExtensions model and follows the extensibility architecture used by Finance and Operations. Microsoft's extension model is designed specifically to enable this type of additive customization without altering the original element. See Customize model elements through extension and Add fields through extension.

QUESTION NO: 32

You need to resolve the issue for UserB.

Which framework should you use?

A. SysExtension

B. SysOperationSandBox

C. SysOperation

D. RunBaseBatch

ANSWER: B

Explanation:

SysOperationSandBox is the appropriate framework when the required operation must run in an isolated sandbox session. This framework supports executing server-side logic outside the caller's session context, which is useful for operations that need isolation from the interactive user session and its client-side state. The sandbox execution model helps ensure that the operation is performed in the intended server environment rather than relying on the user's current session behavior.

In Finance and Operations development, the SysOperation architecture is the underlying pattern for implementing service operations and controllers. SysOperationSandBox builds on that architecture for scenarios specifically requiring sandboxed execution. A developer should define the operation so that it has a clear service boundary and can execute independently of the initiating user session. This approach also aligns with the platform's separation of UI interaction from server-side business processing.

Microsoft describes the SysOperation framework and its service/controller-based programming model in the batch processing overview. For the broader Finance and Operations development architecture and X++ guidance, see X++ language programming guide.

QUESTION NO: 33

A company has an extension that makes changes to a base form.

You need to identify all extension changes made to the base form.

What are two possible ways to achieve the goal? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

A. Right-click the base form and select view references

B. In search bar on the form extension element, enter the text e:

C. In the search bar on the base form, enter the text e:.

D. In the search bar on the form extension element, enter the text c:

E. In the search bar on the base form, enter the text c:.

ANSWER: A B

Explanation:

Right-click the base form and select view references is a valid way to locate extension artifacts that reference the base form. In Visual Studio's Application Explorer, references provide a way to discover dependent metadata, including form extension elements. Opening the relevant form extension then exposes the extension's added or modified designer elements, enabling the developer to identify the changes associated with the base form.

In search bar on the form extension element, enter the text e: is also correct because the Form Designer supports search prefixes that filter nodes by their origin. When an extension form is open, the `e:` filter displays only nodes that belong to the current extension. This makes it practical to isolate extension-added controls, data sources, and other metadata from the inherited base-form structure. Together, reference discovery and the extension-node filter support efficient inspection of form extension work without manually comparing every node in the full form hierarchy.

Microsoft describes form extensions and their development model in Customize model elements by using extensions and documents form development concepts in Build extensible forms.

QUESTION NO: 34

You are Dynamics 365 Finance developer.

You need to initialize an embedded Microsoft Power BI report.

Which code segment should you add to the form initialization method?

A. addReportControl(formGroupControl)

B. initializeReportControlInternal(powerBIConfiguration, reportparameters, reportControl, showError)

C. InitializeReportControl(workspaceName, Formgroup)

D. deployorUpdateReport(powerBIConfiguration, reportName, resourceName)

E. PBIReportHelper::initializeReportControl(powerBIConfiguration, reportParameters, reportControl, showError)

ANSWER: E

Explanation:

`PBIReportHelper::initializeReportControl(powerBIConfiguration, reportParameters, reportControl, showError)` is the supported Finance and Operations helper call for initializing an embedded Power BI report control on a form. It is intended to be called during form initialization after the relevant configuration and parameter objects have been prepared and the form group control that hosts the report is available. The helper connects the report configuration, report parameters, and target form control, then performs the required initialization for rendering the embedded report. The final Boolean argument controls whether an initialization error should be displayed to the user. This public helper is the appropriate extension point for form code because it encapsulates the platform behavior needed to initialize the embedded Power BI experience, rather than requiring the form to invoke lower-level implementation details. Microsoft's workspace analytics guidance demonstrates using `PBIReportHelper::initializeReportControl` when adding an Analytics tab or embedding a Power BI report in the Finance and Operations user interface. See Add an Analytics tab to workspaces and Power BI integration with Finance and Operations.

QUESTION NO: 35

A company uses Dynamics 365 Finance.

You need to migrate all custom report artifacts from a development environment to a cloud-hosted Tier-1 test environment that has Visual Studio installed.

Which two actions should you perform? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

A. Apply a deployment package

B. Export the project as an axppfile

C. Create a deployment package

D. Import the project into a cloud-hosted environment from an axpp file and ensure that the Overwrite Elements option is selected

ANSWER: B D

Explanation:

Export the project as an axppfile and then import the project into a cloud-hosted environment from an axpp file with the Overwrite Elements option selected. An AXPP file is the project export format used by the Dynamics 365 Finance and Operations development tools to transfer project metadata and its included artifacts between development environments. This method is appropriate for moving report artifacts to a Tier-1 environment that has Visual Studio, because the artifacts can be brought into the target developer workspace and built there.

During import, selecting *Overwrite Elements* ensures that target elements with the same identities are replaced by the versions contained in the exported project. This makes the imported project consistent with the source project and is especially important when the target environment already contains prior iterations of the custom report artifacts. After import, the project can be built and synchronized as needed through the normal development workflow. Microsoft describes Finance and Operations projects and their role in organizing development artifacts in Projects in Finance and Operations development tools. For the related lifecycle distinction between developer changes and deployable packages, see Create deployable packages.