MB-820 Q&As

Microsoft Dynamics 365 Business Central Developer

Pass Microsoft MB-820 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.certbus.com/mb-820.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Microsoft Official Exam Center

QUESTION 1

DRAG DROP

You are developing an XMLport to export data from the parent Item table and a related child "Item Unit of Measure\\' table. The XMLport configuration must provide the following:

1.

Link the child table to its parent.

2.

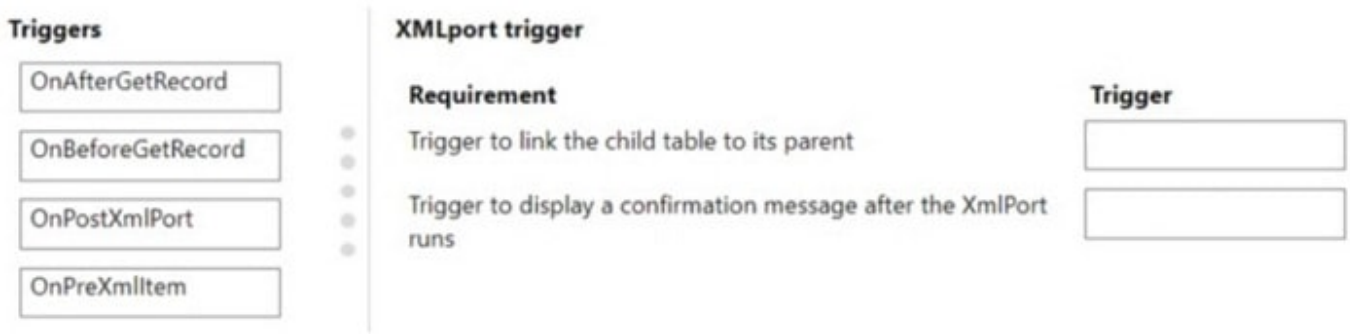
Display a confirmation message after the XMLport runs.

You need to generate the XMLport.

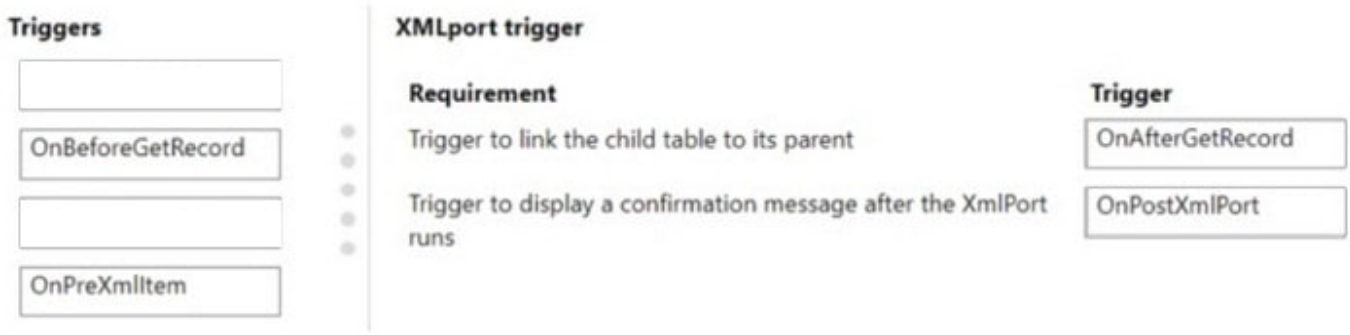
What should you do? To answer, move the appropriate triggers to the correct requirements. You may use each trigger once, more than once, or not at all. You may need to move the split bar between panes or scroll to view content.

NOTE: Each correct selection is worth one point.

Select and Place:



Correct Answer:



To meet the XMLport configuration requirements:

Link the child table to its parent: Use the OnAfterGetRecord trigger. Display a confirmation message after the XMLport runs: Use the OnPostXMLPort trigger.

In Business Central, when you are developing an XMLport for data export, triggers are used to perform actions at

different stages of the XMLport\\'s operation:

OnAfterGetRecord Trigger:This trigger fires after a record is retrieved from the database but before it is processed for output in the XMLport. It is the ideal place to link child table records to their parent because you have access to the current

record that can be used to set filters or modify data in the child table before it is written to the XML file.

OnPostXMLPort Trigger:This trigger fires after the XMLport has finished processing all records. It is the correct place to display a confirmation message because it ensures that the message will appear after the entire XMLport operation is

complete. Here, you can use application-specific functions to show the message, such as MESSAGE function in AL code.

By placing the appropriate triggers in these positions, you can ensure that the XMLport will link the child records to their parent records during the data export process and will notify the user with a confirmation message once the operation is

successfully completed.

QUESTION 2

HOTSPOT

You plan to create a table to hold client data.

You have the following data integrity requirements:

1.

Lookups into other records must be established.

2.

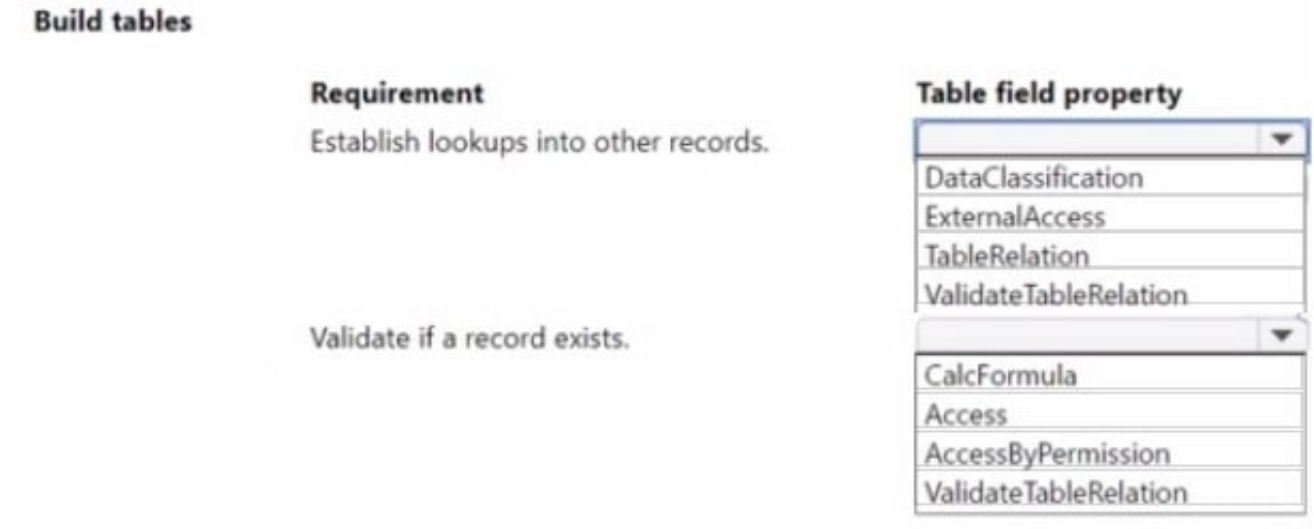
Validate if a record exists in a destination record.

You need to select the table field property to use for each requirement.

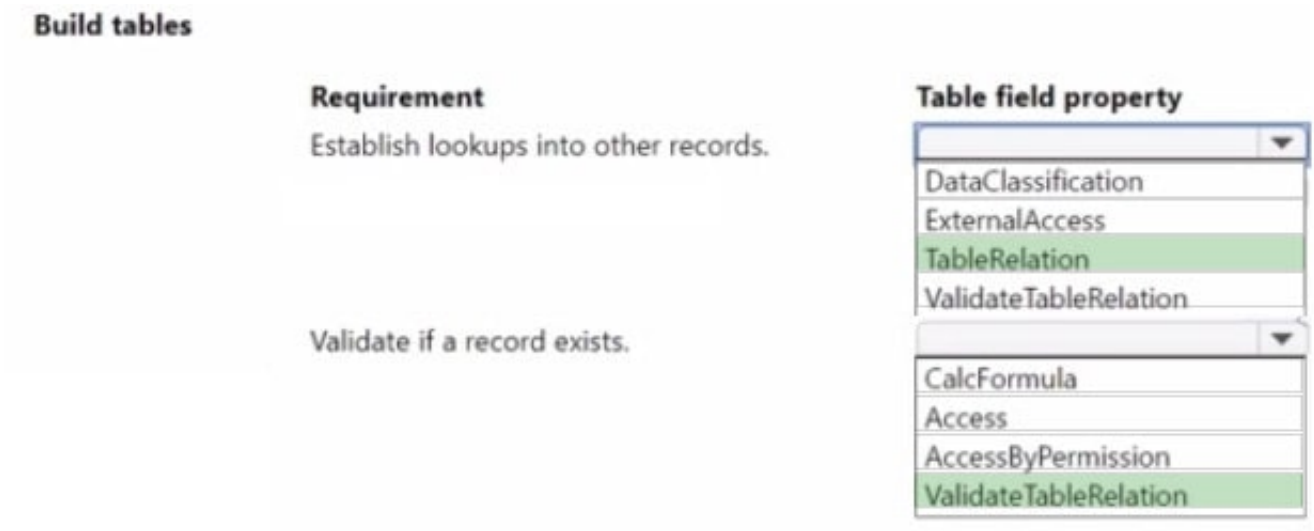
Which table field property should you use? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Hot Area:



Correct Answer:



For the data integrity requirements, the table field properties to use are:

To establish lookups into other records, use the TableRelation property. To validate if a record exists in a destination record, use the ValidateTableRelation property.

In Business Central, when creating tables to hold data, maintaining data integrity is crucial:

TableRelation Property:This property is used to create a relationship between the field in one table and a field in another table, which is typically used for lookups. When you set the TableRelation property on a field, it allows users to select

from a list of values that exist in the related table.

ValidateTableRelation Property:This property is used to ensure that the value entered in a field matches one of the values in a related table. If a user tries to enter a value that doesn\\'t exist in the related table, an error will occur.

QUESTION 3

You need to allow debugging in an extension to view the source code. In which file should you specify the value of the

allowDebugging property?

A. settings.json

B. rad.json

C. app.json

D. launchjson

Correct Answer: C

To enable debugging in an extension and allow the source code to be viewed, the allowDebugging property should be specified in the app.json file (C). The app.json file serves as the manifest for an AL project in Microsoft Dynamics 365 Business Central, defining the project\\'s properties, dependencies, and features. By setting the allowDebugging property to true in this file, developers enable the debugging of the extension\\'s source code, facilitating troubleshooting and development. This is essential for analyzing the behavior of the extension and identifying issues during the development process.

QUESTION 4

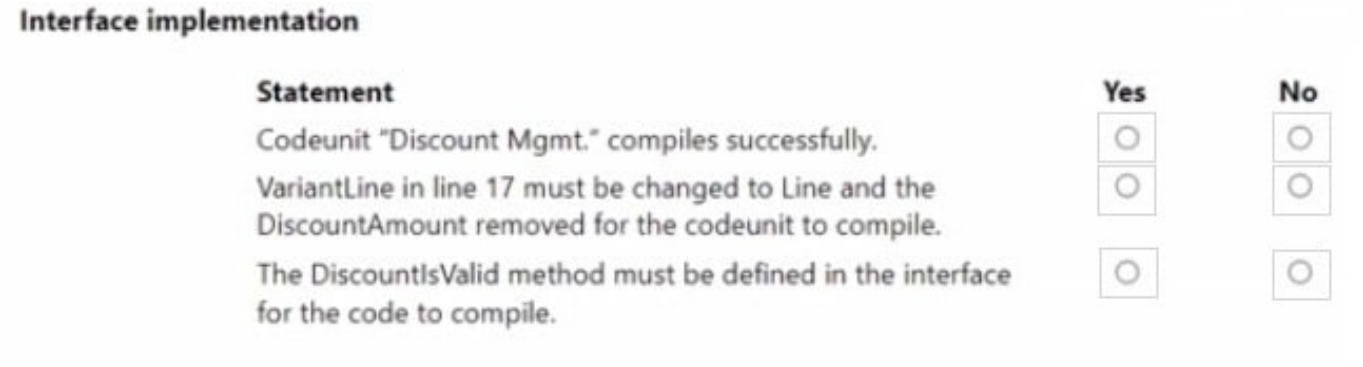
HOTSPOT

You have a per tenant extension that contains the following code.

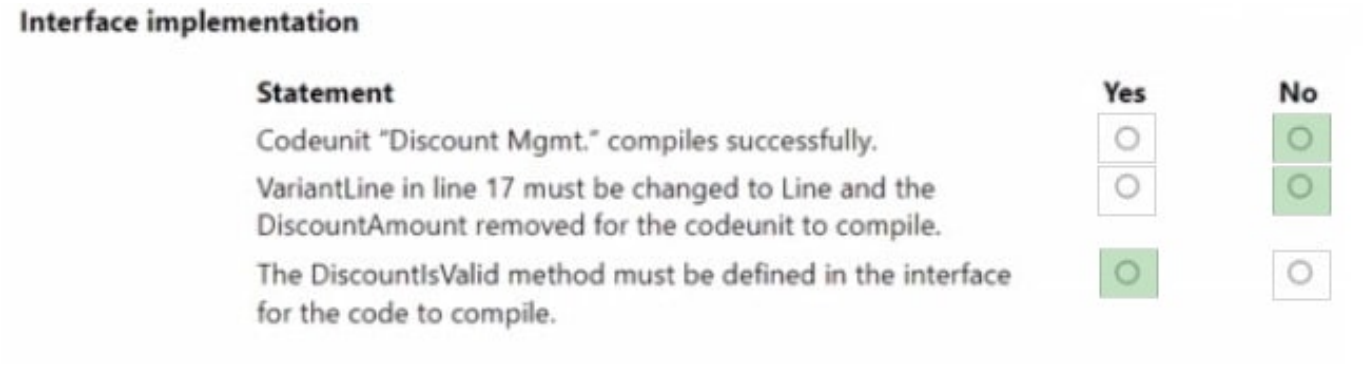
```
10 interface "IDiscount Calculation"
11 {
12     procedure GetLine(var Line: Variant)
13     procedure GetDiscount() : Decimal
14 }
15 codeunit 50100 "Discount Mgmt." implements "IDiscount Calculation"
16 {
17     procedure GetLine(var VariantLine: Variant)
18     begin
19     end;
20     procedure GetDiscount() DiscountAmount : Decimal
21     begin
22     end;
23     procedure DiscountIsValid(DocumentDate: Date): Boolean
24     begin
25     end;
26 }
```

For each of the following statements, select Yes if the statement is true. Otherwise, select No. NOTE: Each correct selection is worth one point.

Hot Area:



Correct Answer:



Codeunit "Discount Mgmt." compiles successfully. = NO VariantLine in line 17 must be changed to Line and the DiscountAmount removed for the codeunit to compile. = NO

The DiscountIsValid method must be defined in the interface for the code to compile. = YES

The codeunit "Discount Mgmt." will not compile successfully as is because the DiscountIsValid method is not defined in the "IDiscount Calculation" interface, yet it is being declared in the codeunit which implements this interface. AL requires

that all procedures in the codeunit that implements an interface must be defined in the interface itself.

The VariantLine in line 17 does not need to be changed to Line, nor does the DiscountAmount need to be removed for the codeunit to compile. These are valid declarations in AL and they are correctly implemented in the codeunit. The Variant

data type in AL is used to handle various data types and DiscountAmount is a valid return type for a procedure.

For the code to compile successfully, the DiscountIsValid method must be included in the interface because AL enforces that any codeunit implementing an interface must implement all the methods defined in that interface.

QUESTION 5

DRAG DROP

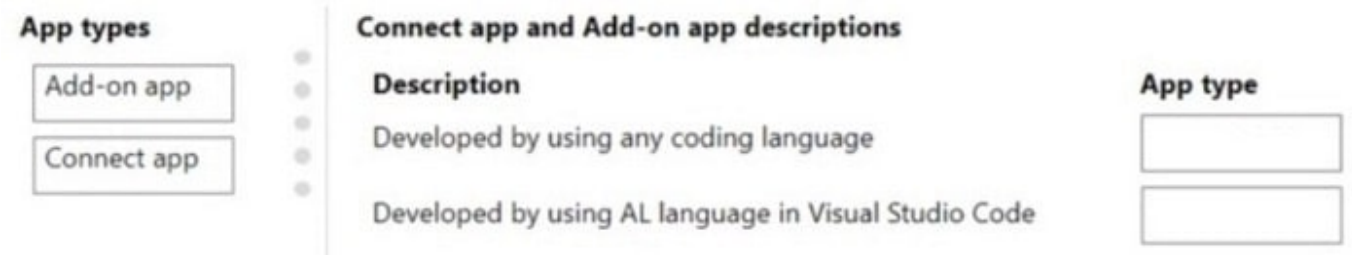
A company is examining Connect apps and Add-on apps for use with Business Central. You need to describe the development language requirements for Connect apps and Add- on apps.

How should you describe the app language requirements? To answer, move the appropriate app types to the correct

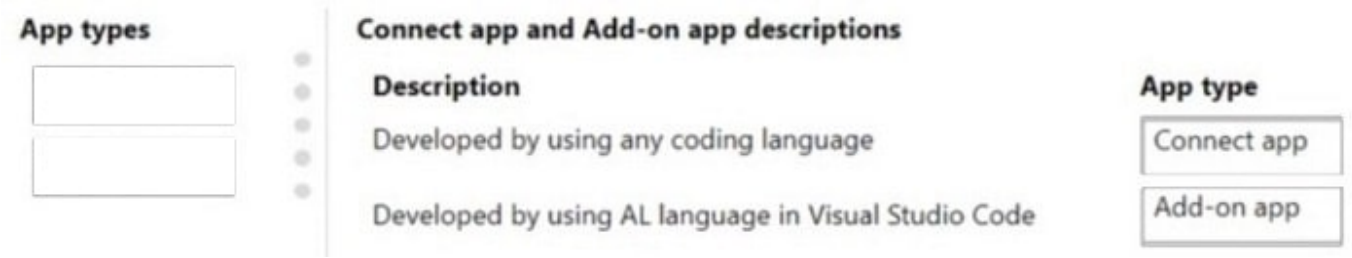
descriptions. You may use each app type once, more than once, or not at all. You may need to move the split bar between panes or scroll to view content.

NOTE: Each correct selection is worth one point.

Select and Place:



Correct Answer:



Developed by using any coding language: Connect app Developed by using AL language in Visual Studio Code: Add-on app

In Microsoft Dynamics 365 Business Central, there are distinct types of applications that can be developed: Connect apps and Add-on apps. Each has its own development language requirements:

Connect apps:

Add-on apps:

The language and environment used for developing these apps are key differentiators between Connect apps and Add-on apps.

QUESTION 6

HOTSPOT

You create a table with fields.

You observe errors in the code

You need to resolve the errors.

```
01 field(1; "Job No."; Code[20])
02 {
03     Caption = 'Job No.';
04 }
05 field(2; Description; text[150])
06 {
07     Caption = 'Description';
08 }
09 field(3; "Sales Amount"; Decimal)
10 {
11     AutoFormatType = 1;
12     CalcFormula = sum("Job Task"."Recognized Sales Amount" where("Job No." = field("No.")));
13     Caption = 'Calc. Recog. Sales Amount';
14     Editable = false;
15 }
16 field(5; "Over Budget"; Boolean)
17 {
18     Caption = 'Over Budget';
19 }
20 field(6; "Project Manager"; Code[50])
21 {
22     Caption = 'Project Manager';
23     TableRelation = "User Setup";
24 }
```

For each of the following statements, select Yes if the statement is true. Otherwise, select No. NOTE: Each correct selection is worth one point.

Hot Area:

Error resolution

Statement	Yes	No
In line 12, declare "Job Task" as a variable.	○	○
Add the property FieldClass = FlowField; for field 3.	○	○
Add the property FieldClass = FlowFilter; for field 3.	○	○
In line 23, assign the "User Setup" table to a field.	○	○

Correct Answer:

Error resolution

Statement	Yes	No
In line 12, declare "Job Task" as a variable.	○	○
Add the property FieldClass = FlowField; for field 3.	○	○
Add the property FieldClass = FlowFilter; for field 3.	○	○
In line 23, assign the "User Setup" table to a field.	○	○

In line 12, declare "Job Task" as a variable. = NO Add the property FieldClass = FlowField; for field 3. = YES Add the property FieldClass = FlowFilter; for field 3. = NO In line 23, assign the "User Setup" table to a field. = YES

For "In line 12, declare \\'Job Task\\' as a variable": In the AL code provided, the "Job Task" appears to be part of a CalcFormula of a FlowField, which means it references a table and not a variable. The "Job Task" does not need to be declared as a variable because it is used to reference a table in a CalcFormula expression. For "Add the property FieldClass = FlowField; for field 3": The line of code CalcFormula = sum("Job Task"."Recognized Sales Amount" where("Job No." = field ("No."))); indicates that this field is calculated from other table data, which is the definition of a FlowField. Therefore, adding the property FieldClass = FlowField; is necessary for the field to function correctly. For "Add the property FieldClass = FlowFilter; for field 3": FlowFilters are used to filter data based on the value in a flow field. Since field 3 is using a CalcFormula to sum values, it is a FlowField and not a FlowFilter. Therefore, this statement is not correct. For "In line 23, assign the \\'User Setup\\' table to a field": The line TableRelation = "User Setup"; suggests that the "Project Manager" field has a relation to the "User Setup" table, which is a method of assigning a table to a field to ensure that the values in "Project Manager" correspond to values in the "User Setup" table. Hence, this statement is true.

QUESTION 7

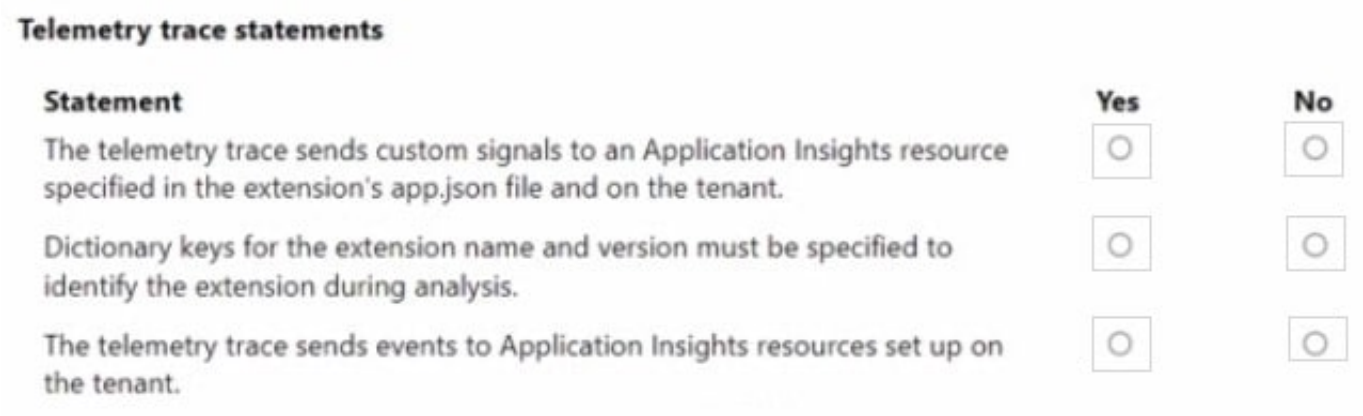
HOTSPOT

A company is setting up a custom telemetry trace signal to send traces on failed customer statement emails.

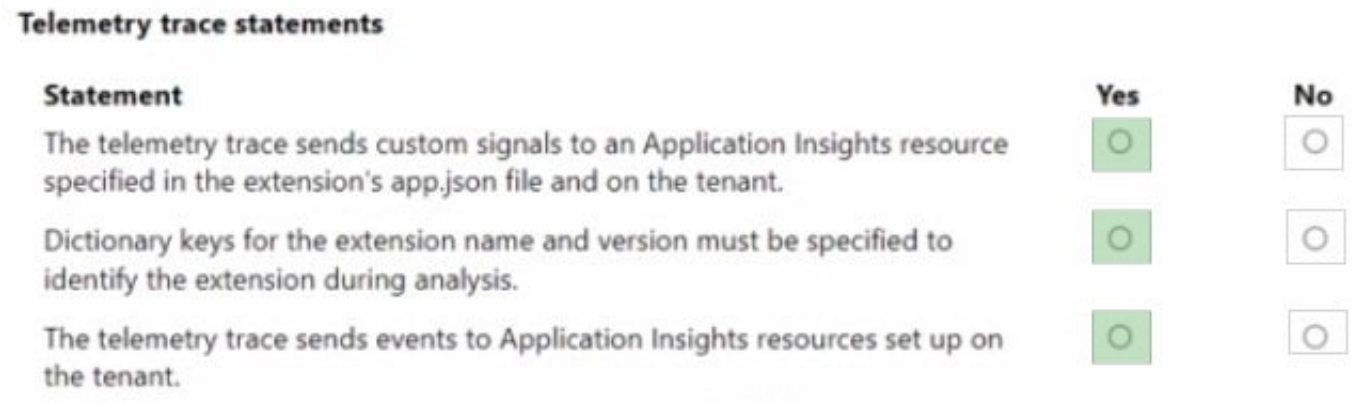
```
05  local procedure SendTraceOnFailedToEmailCustomerStatement(Customer: Record Customer)
06  var
07      Dimensions: Dictionary of [Text, Text];
08      FailedEmailLbl: Label 'Failed to email customer statement';
09  begin
10       Dimensions.Add('systemId', Customer.SystemId);
11      Session.LogMessage('FCUSTSTMT', FailedEmailLbl, Verbosity::Error,
12      DataClassification::SystemMetadata, TelemetryScope::ExtensionPublisher, Dimensions);
13  end;
```

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

Hot Area:



Correct Answer:



The telemetry trace sends custom signals to an Application Insights resource specified in the extension\\'s app.json file and on the tenant. = YES Dictionary keys for the extension name and version must be specified to identify the extension

during analysis. = YES

The telemetry trace sends events to Application Insights resources set up on the tenant. = YES

Telemetry in Business Central allows developers to collect custom telemetry for extensions using Application Insights. The telemetry trace is used to send custom signals to an Application Insights resource. This resource is typically specified

in the app.json file of the extension and must be configured on the tenant where the extension is installed. The use of dictionary keys for the extension name and version is a best practice to identify the extension during analysis in Application

Insights. These keys can be added to the telemetry trace to ensure that when the data is collected, it\\'s clear which extension the data is associated with.

Finally, it is correct that the telemetry trace sends events to Application Insights resources that are set up on the tenant, enabling the collection and analysis of telemetry at the tenant level.

QUESTION 8

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 set 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 on the review screen.

A company creates a Business Central app and a table named MyTable to store records when sales orders are posted.

Users report the following issues:

1.

The users receive permission errors related to MyTable.

2.

Users are no longer able to post sales orders since installing the new app.

3.

The users cannot access the list page created in MyTable.

You need to resolve the user issues without creating new permission sets. You must use the principle of least privilege.

Solution: Decorate the event subscriber used for inserting data in MyTable by entering (InherentPermissions(PermissionOb]ectType:TableData. Database:MyTable. \\'R\\')]

Does the solution meet the goal?

A. Yes

B. No

Correct Answer: A

Using InherentPermissions in an event subscriber with the specified syntax could potentially resolve the permission issues related to MyTable, provided that the permissions specified (in this case, \\'R\\' for Read) align with the minimum necessary for the users to perform their tasks. This approach allows the app to grant permissions dynamically based on the context of the event subscriber, which in this case is involved with inserting data into MyTable. By granting Read permission at the event level, it ensures that users have the necessary permissions to interact with MyTable in the context of the operations facilitated by the event subscriber, without needing to alter existing permission sets or grant broader permissions than necessary. This solution adheres to the principle of least privilege by ensuring that permissions are granted only within the narrow scope needed for specific operations, thereby potentially resolving the reported user issues in a secure and controlled manner.

QUESTION 9

You are developing an app that will be published to Microsoft AppSource.

The app requires code analyzers to enforce some rules. You plan to add the analyzers to the settings.json file.

You need to activate the analyzers for the project.

Which three code analyzers should you activate to develop the app for AppSource? Each correct answer presents part

of the solution

NOTE: Each correct selection is worth one point.

A. CodeCop

B. UlCop

C. a custom rule set

D. PerTenantExtensionCop

E. AppSourceCop

Correct Answer: ADE

When developing an app for Microsoft AppSource, it is crucial to adhere to specific guidelines and standards to ensure compatibility and compliance. The three code analyzers you should activate are:

CodeCop (A): This is the default analyzer for AL language extensions. It enforces the AL Coding Guidelines, ensuring that the code follows best practices for readability, maintainability, and performance. It checks for a wide range of issues,

from syntax errors to best practice violations, making it essential for any AL development.

PerTenantExtensionCop (D): This analyzer is specifically designed for extensions that are intended to be installed for individual tenants. It includes rules that ensure the extension does not interfere with the per-tenant customizations and

adheres to the guidelines for extensions that can be safely installed and uninstalled without affecting the underlying application. AppSourceCop (E): This analyzer is tailored for extensions that are intended for publication on Microsoft

AppSource. It enforces additional rules that are specific to AppSource submissions, such as checking for the use of reserved object ranges and ensuring that all prerequisite dependencies are correctly declared. This is crucial for ensuring

that your app meets all the requirements for listing on AppSource. By activating these three analyzers, developers can ensure their app adheres to the standards required for AppSource, as well as maintain high code quality and compatibility

with Business Central.

QUESTION 10

HOTSPOT

You need to create the API page according to the requirements.

How should you complete the code segment? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Hot Area:

Code segment for API page

```
page 50100 ItemAPI
{
    PageType = API;
    Caption = 'ItemApi';
    APIPublisher = 'contoso';
    APIGroup = 'sales';
    APIVersion = 'v1.0';
    EntityName = 'item';
    EntitySetName = 'items';
    SourceTable = Item;
ODataKeyFields = SystemId;
```

[InsertAllowed = false; | DelayedInsert = true; | DataAccessIntent = ReadOnly;]

[ModifyAllowed = false; | Editable = false; | UsageCategory = Lists;]

```
    layout
    {
        area(Content)
        {
            repeater(GroupName)
            {
                field(no; Rec."No.")
                {
                }
                field(salesTotal; Rec."Sales (LCY)")
                {
                }
            }
        }
    }
}
```

Correct Answer:

Code segment for API page

```
page 50100 ItemAPI
{
    PageType = API;
    Caption = 'ItemApi';
    APIPublisher = 'contoso';
    APIGroup = 'sales';
    APIVersion = 'v1.0';
    EntityName = 'item';
    EntitySetName = 'items';
    SourceTable = Item;
ODataKeyFields = SystemId;
```

[InsertAllowed = false; | DelayedInsert = true; | DataAccessIntent = ReadOnly;]

[ModifyAllowed = false; | Editable = false; | UsageCategory = Lists;]

```
    layout
    {
        area(Content)
        {
            repeater(GroupName)
            {
                field(no; Rec."No.")
                {
                }
                field(salesTotal; Rec."Sales (LCY)")
                {
                }
            }
        }
    }
}
```

Box 1: DataAccessIntent = readOnly;

Scenario:

Contoso, Ltd. must create an API in Business Central to expose item details to the mobile application.

The API must have the lowest possible impact on the production environment when used during working hours.

The API must only support Get operations.

DataAccessIntent Property

Sets the data access intent of the page.

Applies to

Page

Report

Query

Property Value

*

ReadOnly

Intent to access records, but not to modify them. Read-only pages are run against a replica of the database leading to improved performance, but preventing modifications to the database records.

*

ReadWrite

Intent to access and modify records.

Syntax

DataAccessIntent = ReadOnly|ReadWrite;

Remarks

For reports, API pages, and API queries, the Business Central server can use read-only database replicas on Azure SQL Database and SQL Server. If replicas are enabled, use this property to reduce the load on the primary database. Using

ReadOnly might also improve performance when viewing objects. ReadOnly works as a hint for the server to route the connection to a secondary (read-only) replica, if one is available. When a workload is executed against the replica, insert/

delete/modify operations aren\\'t possible. If any of these operations are executed against the replica, an exception is thrown at runtime.

Box 2: Editable = false;

Note: It only applies to pages of the type API. For such, The Editable property must be set to false.

Reference: https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/properties/devenv-dataaccessintent-property