

DUMPS ARENA

Building Intelligent Applications

Microsoft AB-410

Version Demo

Total Demo Questions: 6

Total Premium Questions: 69

Buy Premium PDF

<https://dumpsarena.co>

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.co

Topic Break Down

Topic	No. of Questions
Topic 1, Create an Azure AI Foundry project	1
Topic 2, Develop an AI agent	3
Topic 3, Implement retrieval-augmented generation (RAG) with Azure AI Search	1
Topic 4, Mix Questions	64
Total	69

QUESTION NO: 1

A company uses Power Apps.

Users must be able to view only the address1 columns in the Account table.

You need to ensure other address columns are not visible to users when creating views and filters.

What should you do?

- A. Use column-level security to remove read access to all users.
- B. Create business rules to hide the other address columns.
- C. Delete the other address columns from the table.
- D. Disable the Search option for the columns.

ANSWER: D

Explanation:

Disable the Search option for the columns. In Dataverse, the column *Searchable* metadata setting determines whether a column is made available in search-oriented query experiences, including the filtering and column-selection experiences used to build views. Setting this property to *No* for the unwanted address columns prevents makers and users from selecting those columns while configuring views and filters, while leaving the required *address1* columns available. This meets the requirement by controlling the columns' discoverability in the relevant authoring interfaces rather than changing the data itself. The setting is a column-level metadata configuration, so it can be applied individually to each unwanted address column and does not require deleting standard Dataverse columns or modifying forms. Microsoft documents the Searchable column property as controlling whether a column is available for search, and Dataverse view configuration relies on column metadata when presenting fields that can be used in a view. See [Create and edit columns in Dataverse](#) and [Create and edit public views in model-driven apps](#).

QUESTION NO: 2

A company uses a Copilot Studio agent to support account managers.

The agent must answer questions about product policy by using approved policy documents stored in a SharePoint site. The agent must also retrieve a customer's current entitlement from a Dataverse table when an account manager asks about that customer.

You need to configure the agent.

Which two actions should you perform? Each correct answer presents part of the solution. NOTE: Each correct selection is worth one point.

- A. Add the SharePoint site as a knowledge source.
- B. Embed the policy documents in topic messages.
- C. Add an agent flow that retrieves the current entitlement from Dataverse.
- D. Export the Dataverse table to a static spreadsheet.
- E. Create a business rule on the Dataverse table.

ANSWER: A C

Explanation:

Add the SharePoint site as a knowledge source so the agent can ground policy-related responses in the approved documents. A knowledge source is appropriate for reference content that the agent uses to formulate answers.

Add an agent flow that retrieves the current entitlement from Dataverse. Current customer entitlement is transactional data that must be retrieved at the time of the request. An agent flow can be invoked by the agent to query Dataverse and return the result for use in its response. Embedding policy text in topic messages is harder to maintain and does not provide the required grounding. A Dataverse table alone does not make the live retrieval operation available to the agent.

QUESTION NO: 3 - (HOTSPOT)

A company deploys a model-driven app to multiple departments.

The company requires the following app functionality:

• Enable users to open the app.

• Determine why users cannot access the app.

You need to configure app access.

Which actions should you perform? To answer, select the appropriate options in the answer area. NOTE: Each correct selection is worth one point.

Model-driven app access configuration

Requirement	Action
Enable users to open the app.	Share the app and assign appropriate security roles. Configure app visibility privileges. Configure the app to the site map. Grant table-level permissions to security roles.
Determine why users cannot access the app.	Run the App access checker. Review table permissions. Reassign security roles. Run the solution checker.

ANSWER:

Explanation:

To enable users to open a model-driven app, share the app and assign appropriate security roles. Model-driven apps are made available to users through Dataverse security roles. Sharing the app with the roles used by the relevant departments establishes which groups of users can see and launch it, while assigning users to those roles gives them the app entitlement. The roles must also provide the privileges needed for the Dataverse tables and features exposed by the app, so users can successfully work with the app after opening it. Microsoft documents this role-based approach in its guidance for [sharing a model-driven app using security roles](#).

To determine why a particular user cannot access the app, run the App access checker. This checker is designed specifically to evaluate a user's access to a model-driven app. It analyzes the effective app-sharing configuration and relevant user requirements, helping an administrator identify the condition preventing access. This provides a focused diagnostic path rather than requiring the administrator to infer the cause from multiple individual configuration areas. The checker can be run from the app's sharing experience for the affected user, allowing the administrator to validate access against the specific app and user context. Microsoft's [App access checker documentation](#) describes using it to assess whether a user can access a shared model-driven app and to identify applicable access issues.

QUESTION NO: 4 - (SIMULATION)

A company is creating a canvas app and a model-driven app to manage their customer accounts.

The canvas app requires a business rule to set the Business Type column to large if the customer size is greater than a specific currency value.

The model-driven app requires a business rule to recommend the account rating be re-evaluated when the account goes on credit hold for this app only.

You need to configure the scope for the business rules.

Which scope should you use? To answer, drag the appropriate scopes to the correct business rules. Each scope 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.

Scopes

All forms

Specific form

Table

Answer Area

Business rule

Business Type column setting for customer size

Account rating re-evaluation

Scope

ANSWER: See the explanation for the answer

Explanation:

Configure the business-rule scopes as follows:

Use **Table** scope for the Business Type calculation so the rule is applied at the Dataverse table level and is available to the canvas app. Use **Specific form** for the account-rating recommendation because it must appear only in the model-driven app experience that uses that particular form, rather than on every form for the Account table.

QUESTION NO: 5

A company tracks order processing duration using a model-driven app.

The company requires a system that provides the following functionality:

- ‘ The age of the order, based on the created date, must be displayed on forms and views.
- ‘ The value must update dynamically when viewed without storing the result.
- ‘ The value must NOT be stored in the database.

You need to configure the column types.

Which two column types should you use? Each correct answer presents part of the solution. NOTE: Each correct selection is worth one point.

- A. Formula
- B. Calculated
- C. BigInt
- D. Prompt

ANSWER: A

Explanation:

Formula is the appropriate column type because Dataverse formula columns are virtual: their values are evaluated when they are retrieved rather than being persisted as row data. This directly meets the requirement for an order-age value that is displayed in both model-driven app forms and views, remains current as time passes, and is not stored in the database. A Power Fx formula can derive the elapsed time from the system-created date and the current date or time, subject to the functions and types supported for formula columns. Since retrieval evaluates the expression against the current time, an order opened tomorrow will show a greater age without requiring a scheduled flow, plug-in, or update to the order record. Formula columns are therefore well suited to transient, derived information such as order age. Microsoft documents that formula columns use Power Fx and are calculated at runtime, with the resulting values not stored in Dataverse. See [Formula columns in Microsoft Dataverse](#) and [Define calculated columns in Dataverse](#).

QUESTION NO: 6 - (DRAG DROP)

A company develops and deploys AI-enabled Power Platform solutions across multiple environments. Environment access is controlled through security groups.

The development team requires Power Platform environments to store, manage, and share the solutions across the company.

All developers require access to the same environment. The environment must be subject to ALM controls.

You need to set up the environments.

Which environment types should you use? To answer, move the appropriate environment types to the correct requirements. You may use each environment 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.

Environment types

Environment type selection

	Requirement	Environment type
Default	Build and validate new features	
Sandbox	Host stable business workloads	
Developer		
Production		

ANSWER:

Explanation:

A Sandbox environment is the correct environment for building and validating new features because it provides a separate, nonproduction boundary where the team can work on shared Power Platform solutions without affecting the live business workload. The scenario requires all developers to access the same environment and requires lifecycle controls. A shared Sandbox meets that need: access can be restricted with the specified security group, makers can collaborate on solutions, and changes can be tested and validated before release. Sandbox environments also support administration and lifecycle activities appropriate for controlled development and test work, including copying, resetting, and administration mode when required. Microsoft's environment guidance describes Sandbox as a nonproduction environment intended for development and testing, and its environment strategy guidance emphasizes separating development and validation work from live operations. See [Power Platform environments overview](#) and [application lifecycle management overview](#).

A Production environment is the correct environment for hosting stable business workloads. After a solution has passed the team's validation process, it should be deployed as the approved workload in a controlled operational environment. Production is designed for business use and provides the appropriate foundation for the live application, its data, user access, governance, monitoring, and ongoing managed releases. Security groups can continue to limit access to the

intended users in this environment, while solution-based deployment preserves the lifecycle boundary between the team's work and the stable workload. This separation supports reliable releases and prevents unvalidated development changes from being made directly to the business application. Microsoft documents Production as an environment type intended for permanent work in an organization; see [create and manage environments in the Power Platform admin center](#).