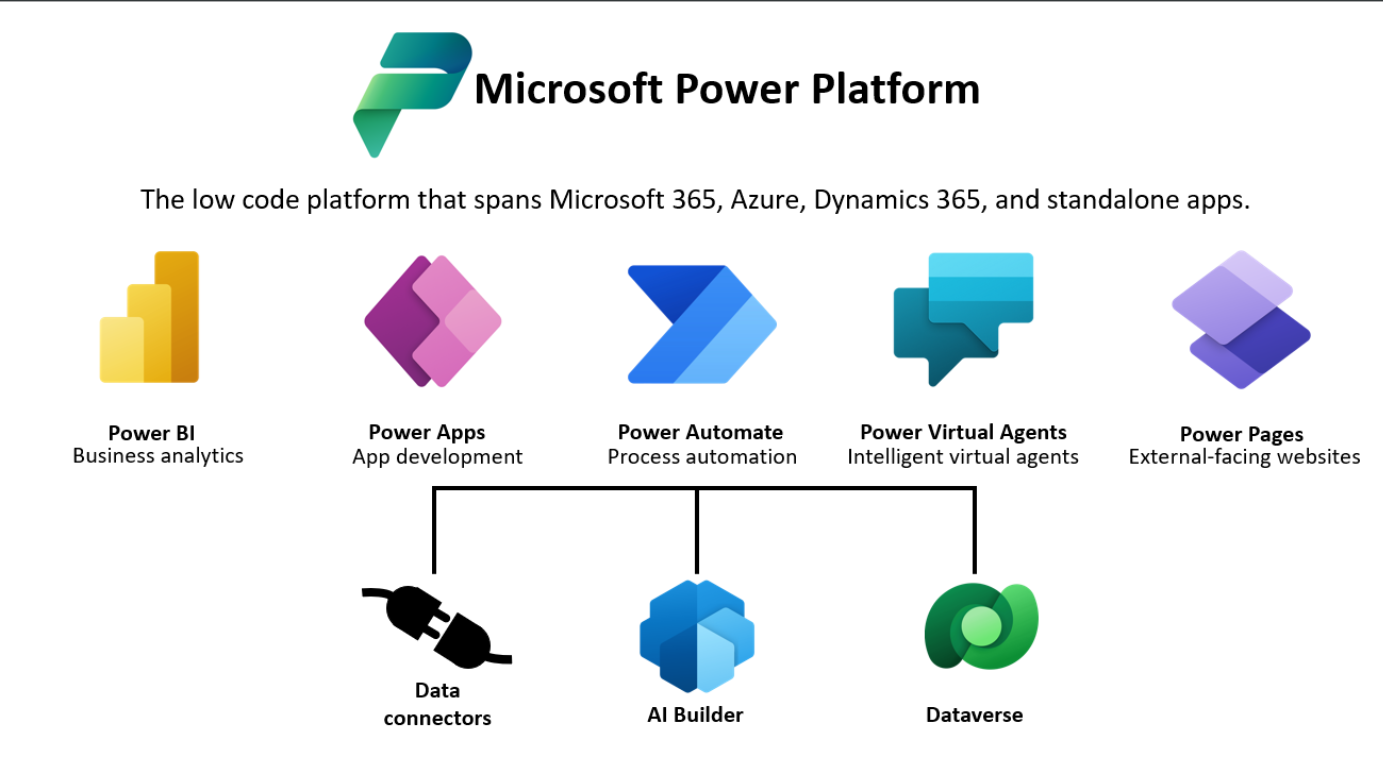
Module-1: Describe the business value of the Microsoft Power Platform

Microsoft Power Platform enables your business to craft solutions while empowering you to unite customized technology to help everyone, from CEO to front line workers, and to drive the business with data-driven insights.

**Explore Microsoft Power Platform**

With Power Platform, solutions can be built in days or weeks, as opposed to months or years. Microsoft Power Platform is composed of five key products: Power Apps, Power Automate, Power BI, Power Virtual Agents, and Power Pages.



In addition to the products listed above, there are additional tools that enhance the solutions you create on the Power Platform. Some of these are:

AI Builder

Microsoft Dataverse

Connectors

**Describe the business value of the Power Platform**

As the business climate changes, new factors are impacting businesses.

These impacts include:

Changing workforce expectations

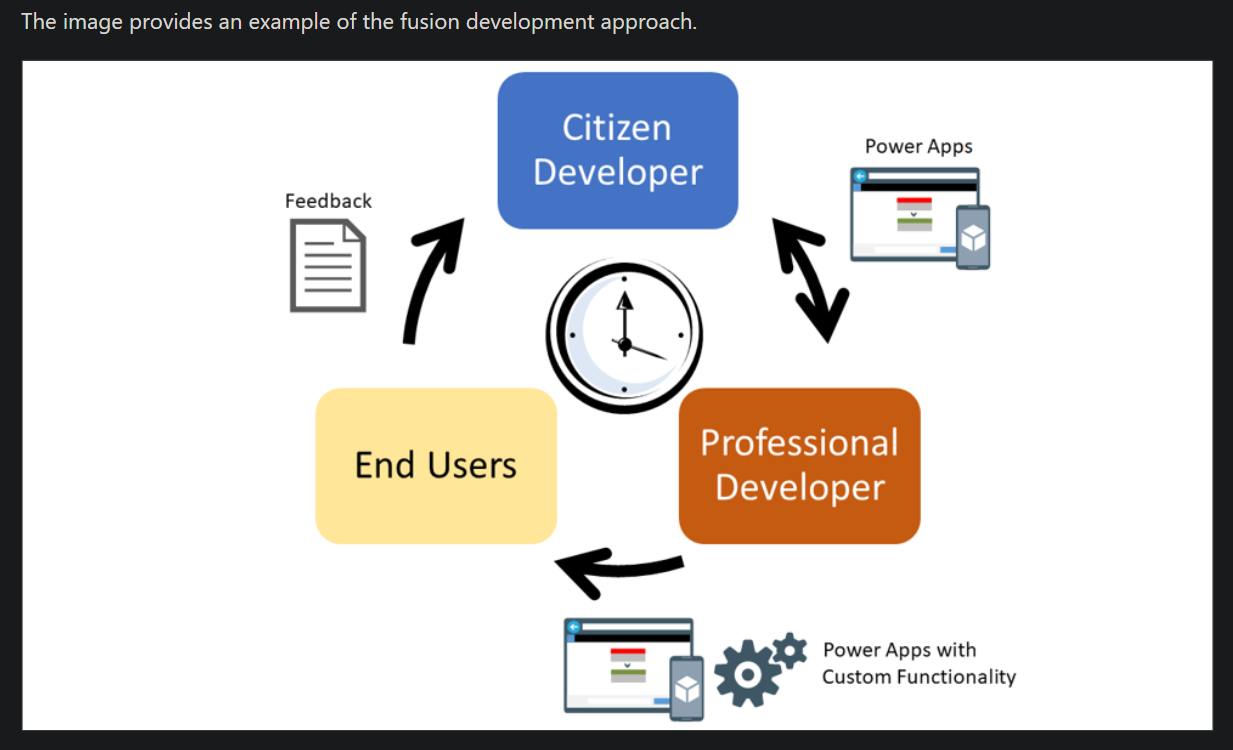
Increased costs for custom application development

Need to become more agile

Need to scale development efficiently

The Power Platform makes it easy for organizations to address all the challenges mentioned.

The APIs can be easily added to Power Apps and Power Automate. This collaborative approach to creating solutions is often referred to as fusion development, allowing organizations to use the best resources for the required task.



**Explore connectors and Microsoft Dataverse**

Creating solutions that span multiple services can create challenges. One of those challenges is ensuring that you can perform the necessary operations in all systems.

**Connectors**

Creating solutions that span multiple services can create challenges. One of those challenges is ensuring that you can perform the necessary operations in all systems

With Connectors:

The process is automated, efficient, and reduces human intervention.

The data is updated automatically in relevant systems, minimizing errors and delays.

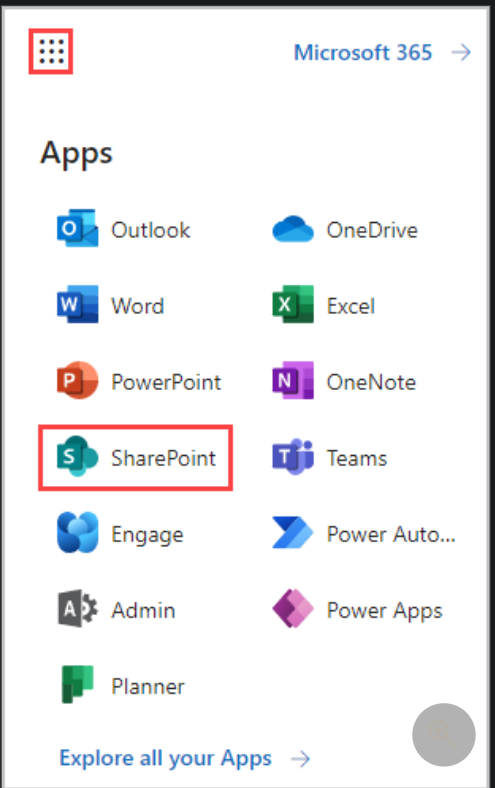
Without Connectors:

The process is manual, time-consuming, and prone to errors.

There's a higher likelihood of delays and miscommunication, impacting efficiency and productivity.

Examples of Connectors:

Microsoft 365 Connectors:



Microsoft Outlook: Send and receive emails, manage calendars, and access other mailbox functionalities.

Microsoft SharePoint: Integrate with SharePoint lists, libraries, and documents for collaboration and data management.

Microsoft Teams: Automate actions and notifications within Microsoft Teams, like sending messages or creating channels.

Microsoft Forms: Automate actions based on form responses and manage forms and responses.

Database Connectors:

Microsoft SQL Server: Connect to and work with data in Microsoft SQL Server databases.

Common Data Service (CDS) and Microsoft Dataverse: Integrate with the centralized data platform for storing and managing data.

Azure SQL Database: Access and manage data in Azure SQL databases.

Cloud Service Connectors:

Azure Logic Apps: Connect to Azure Logic Apps for advanced workflow automation and integration with Azure services.

Azure Blob Storage: Access and manage files in Azure Blob Storage.

Azure Table Storage: Work with data in Azure Table Storage.

**Microsoft Dataverse**

Microsoft Dataverse allows organizations to securely store and manage data used by your business applications. Dataverse data is stored in tables. A table is a set of rows and columns. Each column in a table stores specific type of data such as names, locations, ages, dates, salaries, and so on. In addition to data storage, Dataverse also has other elements that help with securing data, data validation, and productivity.

With Microsoft Dataverse:

Data is centralized, secure, and easily accessible across the organization.

Integration, automation, and analytics are streamlined, promoting efficient operations and informed decision-making.

Without Microsoft Dataverse:

Data is scattered and difficult to manage, leading to inefficiencies and potential inaccuracies.

Integration and reporting are complex and often require custom solutions, resulting in higher development costs and longer project timelines.

Examples:

Customer Relationship Management (CRM) System:

Human Resources (HR) Application:

Inventory Management System:

Many solutions built on Power Platform use both Dataverse and connectors heavily. Dataverse acts as the primary business data storage mechanism, and connectors are used in the different apps and automations that are connecting to different data services.

**Describe how Power Platform works with Microsoft 365 apps and services**

Organizations around the world use Microsoft technologies in the enterprise capacity. They often use Microsoft 365 for as their email and productivity platform. They might also use SharePoint as their document management platform. Because of the pervasive use of Microsoft 365 services, there are multiple built-in scenarios where the Power Platform can help you to create a more streamlined solution by interacting directly with Microsoft 365 services.

**Explore how Power Platform works with Microsoft Teams**

Microsoft Teams has filled that need for many organizations. It provides a central point where users can collaborate with other users, have meetings, manage projects, and more. One key advantage of Microsoft Teams is how extensible and adaptable it is. Microsoft Power Platform is the innovative gateway to rapidly build Teams compatible apps using low-code attributes. All Power Platform components can be used with Microsoft Teams.

**Describe how Power Platform works with Microsoft Dynamics 365 apps**

Dynamics 365 is a set of intelligent business applications that help organizations run their entire business and deliver greater results through predictive

Examples of Microsoft Dynamics 365 Apps:

Dynamics 365 Sales:

Dynamics 365 Marketing:

Dynamics 365 Customer Service:

Dynamics 365 Field Service:

Dynamics 365 Finance:

There are many ways that the Power Platform works with Dynamics 365 products. First, all Dynamics 365 customer engagement apps are what are referred to as model-driven applications. Model-driven applications are built using Power Apps. They are based on a data model store within Microsoft Dataverse. Components such as forms, views, charts, and dashboards are used to present data to end users.

Model-driven applications are not the only component of the Power Platform used by Dynamics 365 applications. Other Power Platform components used by Dynamics 365 customer engagement apps can include:

Power BI

Power Automate

Power Virtual Agents

Power Pages

**Describe how Power Platform solutions consume Microsoft Azure services**

The Azure cloud platform consists of hundreds of products and cloud services designed to help you bring new solutions to life to solve today's and future challenges. These services range from data storage services to virtual machines, to artificial intelligence services.

Power Platform and Azure services are a perfect complement for each other, and the possibilities for using them together are endless. Azure services can be used with Power Platform to help modernize legacy systems, automate processes, and create advanced analytical solutions.

**Explore how Microsoft Power Platform apps work together**

Power Platform can add value to any business by helping you to analyze, act, and automate.

Act by building custom apps in Power Apps, automate processes based on the data you collect in Power Automate, and analyze the data you have collected in Power BI.

Module 2: How to build your first model-driven app with Dataverse

**Introduction to Microsoft Dataverse**

To thrive and grow, businesses need to capture, analyze, predict, present, and report data and do it all with a high level of agility.

Microsoft Dataverse

Microsoft Dataverse is the cloud data platform for the Microsoft Power Platform that is easy to use, compliant, secure, scalable, and globally available. Using Microsoft Dataverse as the data store for your apps has many **benefits**:

Metadata: Properties you define on your data model are used by Power Apps, speeding up the building of apps.

Data access: You can control who can access which tables, rows, and columns.

Logic: Calculations and rules can be added to table columns.

Import and export: You can choose from multiple tools to manage your data, including Excel.

Audit: You can track who accesses and changes data.

Processes: Business processes can be added to ensure data quality and perform automation.

Managed data in the cloud: You don't need to perform backups or configure the database because this is performed for you by Microsoft.

Storage: You don't need to worry about how the data is stored because Microsoft takes care of this for you.

Skills: You don't need to be a DBA to manage Dataverse.

Secure: All data is encrypted at rest and in transit.

Integration: Dataverse is deeply integrated with Microsoft cloud services such as Microsoft 365 and Microsoft Azure.

Proven: Microsoft Dataverse is used by Dynamics 365 apps such as Dynamics 365 Sales and Dynamics 365 Customer Service.

Connectors: Microsoft Dataverse has connectors for Power Apps, Power Automate, and Power BI.

Tables

Data within Dataverse is stored within a set of records called tables. A table is a logical structure of rows and columns that represent a set of data, similar to how a table stores data within a database.

Microsoft Dataverse is designed to let you quickly and easily create a data model for your application, based on the tables and the table metadata that you include in your app.

Dataverse includes a base set of standard tables that support common business scenarios. You can also create custom tables specific to your organization.

**Table characteristics**

In Microsoft Dataverse, you create and edit tables by using the Power Apps maker portal.

Tables in Dataverse have the following **properties:**

Name: The simple name defining your table such as Pets

Rows: Specific records in your data (such as a cat named Ashley)

Columns: Metadata for each record (such as pet Species and Breed)

Relationships: Describe the links to other tables (such as a pet belonging to an owner)

Keys: Column, or columns, to uniquely identify a row in the table (such as an ID number)

Forms: Forms used by model-driven apps to view and edit a table row

Views: Define the rows and columns for displaying table rows

Charts: Visualizations of table rows

Dashboards: Provide a customizable glance at your data through charts and filtered data

Business rules: Logic that can be applied to columns in a table row

Metadata: Properties of the table that affect the capabilities of the table and how apps and flows can use the table

Commands: You can customize the buttons in your command bar for your model driven app

Row keys:

Database tables have a primary key. A primary key uniquely identifies a single row in the table. In Dataverse, the primary key is a Globally Unique Identifier (GUID), a 32-character string such as 123e4567-e89b-12d3-a456-426655440000. The GUID for the primary key is generated automatically when a row is created in a Dataverse table.

Because GUIDs aren't user friendly, in Dataverse there's also a column called the primary column. This is a text string that is used to represent the row in apps and flows. When you create a Dataverse table, this column is defaulted to Name.

Table types:

Standard: A table where you can store data and add to the navigation in model-driven apps. Most tables you create are standard tables. You can't delete standard tables, but you can hide them.

Activity: Activity tables store interactions such as phone calls, tasks, and appointments. Dataverse has a set of activity tables. These tables share the same set of columns and share security privileges. Many of the table options, including the primary column, are fixed and can't be changed. Activity tables appear in the timeline on model-driven app forms.

Virtual: Virtual tables connect to data from an external data source such as Microsoft Azure SQL Database or a SharePoint List.

Elastic: Used when your table includes a large dataset, the elastic tables offer performance benefits over standard tables and are powered by Azure Cosmos DB.

Table ownership:

When you create a custom table, the options for ownership are User or team owned, or Organization-owned. After a table is created, you can’t change the ownership.

User or team owned: Actions that can be performed on these rows can be controlled at the user level.

Organization-owned: Access to the data is controlled at the organization level.

**Create a Microsoft Dataverse table**

You create a table and then customize key components:

Create a custom table.

Add custom columns to your table.

Customize a view.

Customize a form.

**Import data into your Microsoft Dataverse database**

You can import data into your Microsoft Dataverse database in bulk from various sources including Excel, Access, SharePoint, SQL and many others.

**Table relationships**

When working with Microsoft Dataverse, you need to try to keep information about different sets of data in separate tables. This is known as data normalization. Table relationships define the different ways table rows can be associated with rows from other tables.

**Relationship types**

One-to-many relationships: In a one-to-many table relationship (1:N), many referencing (related) table rows can be associated with a single referenced (primary) table row. The referenced table row is sometimes referred to as the ”parent” and rows of the referencing table are referred to as ”children.” A many-to-one relationship is just the child perspective of a one-to-many relationship

Many-to-many relationships: In a many-to-many (N:N) table relationship many table rows can be associated with many other table rows. Rows related using a many-to-many relationship can be considered peers and the relationship is reciprocal.

Remember that One-to-many relationships have a hierarchy between the rows in the table relationship. Rows that are related through Many-to-many table relationships are considered peers.

Many-to-many relationships are not supported by relational databases and Dataverse creates a hidden Intersect table to link the table rows. This intersect table does not have a form and you cannot edit its properties or add columns to the table.

**Create table relationships**

Create a relationship by using a lookup column.

Add a One-to-Many table relationship.

Add a Many-to-Many table relationship.

**Dataverse logic and security**

Microsoft Dataverse is more than a data store, it contains features to apply business logic to data and securing access to data.

Business rules: Business rules validate data across multiple columns in a table, and provide warning and error messages, regardless of the app that's used to create the data.

Business process flows: Business process flows guide users to ensure they enter data consistently and follow the same steps every time. Business process flows are currently supported only for model-driven apps.

Real-time workflows: Workflows automate business processes without requiring user interaction.

Business logic with code: Business logic supports advanced developer scenarios that extend the application directly through code.

Business rules are created and managed in the Power Apps maker portal. Business rules are defined for a Dataverse table and are applied based on the setting of the Scope on the rule:

Individual form: The rule applies only to the specified model-driven app form.

All forms: The rule applies to all model-driven app forms.

Entity: The rule applies to all model-driven app forms and when the row is created or updated on the Dataverse table. This is the default setting.

Dataverse security:

Dataverse has a comprehensive security model that controls access to environments, tables, table rows, and other Dataverse features such as importing and exporting of data.

Access to data in Dataverse is controlled using role-based access control (RBAC).

You can assign several built-in security roles:

System Administrator: Has full permission to customize or administer the environment.

Environment Maker: Can create new resources associated with an environment, including apps, flows, and connections.

Basic user: Can run an app within the environment and perform common tasks for rows on the out-of-the-box tables such as Account and Contact.

Security roles are managed from the Power Platform admin center

Within the security role, you can specify the access levels for each privilege for every table: the Create, Read, Write, Delete, Append, Append To, Assign, and Delete privileges

**Create a custom table and import data:**

Use Microsoft Dataverse to store data:

Create a custom table.

Add custom columns to your table.

Create a calculated column.

Configure a business rule.

Import data from a Microsoft Excel workbook.

**Dataverse auditing:**

Microsoft Dataverse supports an auditing feature designed to meet the external and internal auditing, compliance, security, and governance policies that are common to many enterprises. Dataverse auditing logs changes that are made to customer records in an environment with a Dataverse database. Dataverse auditing also logs user access through an app or through the SDK in an environment.

Audit logs can also be retrieved using the Web API or the Organization Service.

Key concepts

There are three levels where auditing can be configured: an environment, table, and column. For table auditing to take place, auditing must be enabled at the environment level. For column auditing to take place, auditing must be enabled at the environment level and the table level.

To enable user access auditing (Log access) or activity logging (Read logs), auditing must be enabled at the environment level. The option to enable activity logging is only visible when the minimum Office licensing requirements are met.

You must have System Administrator or System Customizer role or equivalent permissions to enable or disable auditing.

Auditing can be configured manually via the Power Platform admin center and the Power Apps portal. Auditing can also be configured programmatically.

Configure tables and columns for auditing in Power Apps

The following task requires the System Administrator or System Customizer role or equivalent permissions.

With this simple setup, Dataverse can track changes to a table or a specified column. It's possible to refer to this data in a model driven app under Audit history. Remember that you can view auditing on an environmental level if you use the Audit Summary View, which is a comprehensive list of all audit logs in an environment. The Audit Summary View is available in the Power Platform admin center.

**Dual-write vs. virtual tables**

1. Dual Write:

Definition:

Dual Write is a feature that enables real-time, bidirectional data synchronization between Dynamics 365 Finance and Operations (F&O) and Dynamics 365 Customer Engagement (CE) or other Dynamics 365 applications. It ensures that data changes in one system are automatically reflected in the other system in near real-time.

When to Use Dual Write:

Use Dual Write when you have scenarios where you need to maintain synchronized and up-to-date data between Finance and Operations (F&O) and Customer Engagement (CE).

This is particularly useful when you need to seamlessly integrate financial data (F&O) with customer, sales, or service data (CE) to ensure a unified view of your operations.

2. Virtual Tables:

Definition:

Virtual Tables in Dataverse are read-only tables that allow users to view and interact with data from external data sources directly within Dataverse. These tables are not physically stored in Dataverse; instead, they provide a virtual representation of the data from the connected external system.

When to Use Virtual Tables:

Use Virtual Tables when you want to access and utilize data from external systems (e.g., SQL databases, SharePoint, Azure SQL) within Dataverse without physically moving or duplicating the data.

This is helpful when you need to present data from various external sources alongside your Dataverse data, providing a consolidated view for reporting, analytics, or business processes.

Comparison:

Dual Write is used for real-time bidirectional data synchronization between specific Dynamics 365 applications (e.g., F&O and CE), ensuring consistent data across the integrated systems.

Virtual Tables, on the other hand, provide a way to view data from external systems within Dataverse without actually moving or storing that data in Dataverse.

Module-3: [How to build your first model-driven app with Dataverse](https://learn.microsoft.com/en-us/training/modules/build-first-model-driven-app-dataverse/)

**Introduction to Dataverse:**

Dataverse is the ideal data source because it's the foundational data source of Microsoft Power Platform. Another benefit of using Dataverse is that all your data is stored in tables. A table is a set of records that are used to store data, similar to how a table stores data within a database. Dataverse help provide a secure and cloud-based storage option for your data.

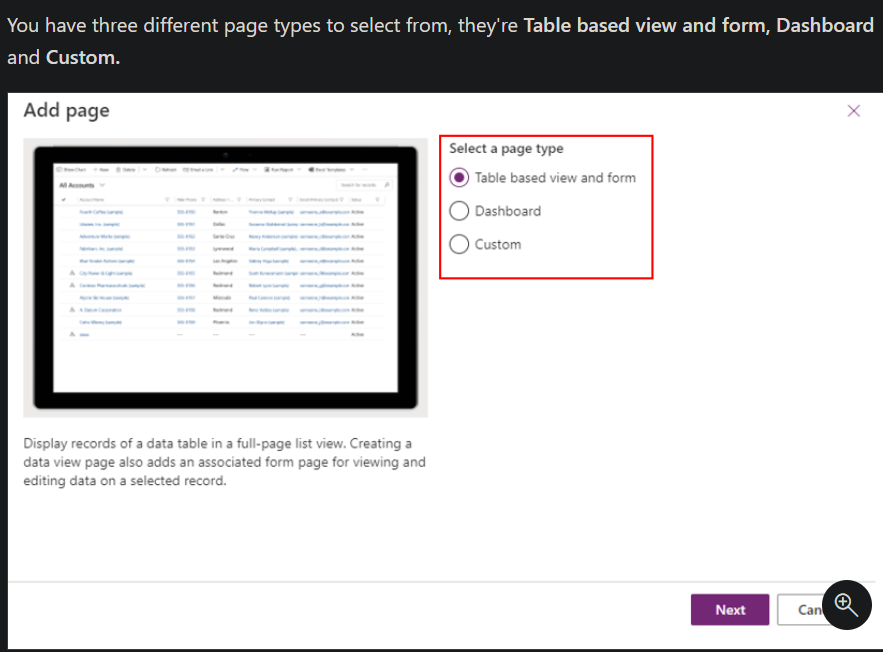


Contoso Corp Tenant where Azure, Microsoft 365 and power platform services reside.

It contains environment where data source tables and apps and flows can be managed as a container. We can have multiple container.

**Introduction to model-driven apps**

Unlike canvas apps, where you build out an app screen-by-screen by adding logic and code as you go, model-driven apps can be created with a few simple steps. Model-driven apps use a component-focused approach to develop the app.



Through model-driven apps, you can create business process flows.

**Model-driven apps, powered by Microsoft Dataverse**

It's important to ensure that your business data and business processes at the data level are structured properly before you compose your app. Model-driven apps will automatically generate a UI that is responsive across devices; however, this outcome relies heavily on how your data is modeled in Dataverse.

Approach to model-driven app making

When creating model-driven apps, it's important to focus on three areas:

Modeling business data

Defining business processes

Composing the app

**Explore sample apps:**

In Power Apps, you can use a sample app to explore design possibilities. You'll also discover concepts that you can apply as you develop your own apps. Every sample app uses fictitious data to showcase a real-world scenario.

**Summary:**

To review, this module explained the following concepts:

Dataverse data is stored in tables.

There are numerous default tables available out-of-the box to help you quickly build apps.

You can create custom tables.

Data is stored in a way that users can only view it if you grant them access.

Sample model-driven apps and templates are available to help you learn and create your own apps.

**Module 4: Get started with model-driven apps in Power Apps**

Model-driven app design is an approach that focuses on quickly adding components to your apps. These components include dashboards, forms, views, and charts. With little or no code, you can make apps that are simple or complex.

**Introducing model-driven apps**

The approach to making model-driven apps

Model-driven apps have three design phases:

Model your business data

Define your business processes

Build the app

**Components of model-driven apps**

To understand how each of these components relates to app design, they're separated here into data, UI, logic, and visualization categories.

Data components:

Table

Column

Relationship

Choice column

User interface components:

App

Site map

Form

View

Logic:

Business process flow

Workflow

Actions

Business rule

Power Automate Flow

Visualizations:

Chart

Dashboard

Embedded Power BI

Advanced model-driven apps:

Solution Explorer is used to make advanced model-driven apps.

**Design model-driven apps:**

When you design your Power Apps solution, there are several different factors to consider:

Business requirements

Data Model

Business Logic

Output

Your primary design goal is to get your Microsoft Dataverse data model in order. With that in place, you can connect Power Apps, and a Model-driven app will be created for you from that model.

Model-driven apps are created using the App Designer.

Business requirements

The first step in the process is to understand your business requirements. Work with the app stakeholders to consider your security, accessibility, data, and design needs.

For security, the Dataverse has a robust security model.

Remember, model-driven applications use a metadata-driven architecture.

Note:

If a column type needs to be changed to a different column type, (i.e. text column to an choice), then you will need to delete that column and recreate with the correct column type. This will cause you to lose any data associated with that column.

User Interface (UI) and User Experience (UX):

When you build a Model-driven app, most of the UI and UX are predetermined for you.

Business logic:

When you incorporate business logic in your app, there are two primary options available. You can set Business Rules on your Dataverse tables or you can build Business Process Flows.

With Business Rules, you'll define behaviors at the data layer. Business rules are great for changing when a column is required, setting a default value, or even showing or hiding a column based on a criteria.

Output:

A common output need for apps is to visualize the data.

**Control security when sharing model-driven apps**

To control who can access restricted or sensitive data and resources and what they can do with them, you can assign users security roles. Microsoft Power Apps uses role-based security for sharing.

Privilege Description

Create Required to make a new record.

Read Required to open a record to view the contents.

Write Required to make changes to a record.

Delete Required to permanently remove a record.

Append Required to associate the current record with another record. For example, if users have Append rights on a note, they can attach the note to an opportunity. If there are many-to-many relationships, a user must have Append privilege for both tables being associated or disassociated.

Append to Required to associate a record with the current record. For example, if users have Append To rights on an opportunity, they can add a note to the opportunity.

Assign Required to give ownership of a record to another user.

Share Required to give access to a record to another user while keeping your own access.

**Summary:**

Defining and enforcing consistent business processes is a key aspect of model-driven app design. Consistent processes help ensure that your app users can focus on their work and not have to remember to perform a set of manual steps.

Let's review what you've learned:

Model-driven app design is an approach that focuses on quickly adding components to apps. These components include dashboards, forms, views, and charts.

Little or no code is required to build model-driven apps.

Model-driven design uses metadata-driven architecture so that designers can customize their apps.

The best way to get started building model-driven apps is to use sample apps and data; then customize the apps.