**Configure Microsoft Dataverse (25–30%)**

**Manage the data model**

* Create or modify standard, activity, or virtual tables
* Create new tables or modify existing tables
* Determine which type of relationship to implement, including one-to-many and many-to-many
* Configure table relationships including behaviours and cascading rules
* Create new or modify existing columns, including calculated, rollup, and formula
* Configure table properties

**Manage Dataverse**

* Configure Dataverse search, and manage the search index
* Manage auditing
* Describe options for importing and exporting data
* Configure duplicate detection settings
* Configure bulk deletion

**Configure security settings**

* Manage business units
* Create and manage security roles
* Create and manage users and teams
* Create and manage column security
* Configure hierarchy security
* Configure Microsoft Entra ID group teams

## Getting Started with Dataverse

Each ***environment*** is created under a Microsoft Entra ID tenant, and only users within that tenant can access its resources. An environment (Dataverse DB and resources created in that env) is also bound to a geographic location, like the United States. Can create more than 1 env (e.g., dev and prod).

Dev env - A Power Platform **admin** can create Developer environments for other users and can restrict users from creating their own Developer environments.

Creating env - once you create an environment with Dataverse in it, you can access that Dataverse data from other environments as well. Dataverse creation with an environment reserves up to 1 GB in storage.

Dataverse

Dev env

Dev env

Dataverse

Dataverse

Prod env

Prod env

Users & Roles - After you have created an environment, users from Microsoft Entra ID that are associated with your tenant and have a selected security role are automatically added to the environment.

\*We recommend that you create Microsoft Entra ID groups and associate roles with those security groups to simplify permissions and data access.

Notes: The two Environment **roles** that are built into **every** environment are System Administrator and Environment Maker. All other roles are user security roles.

The following **operation** can be done on the **Environment**:

Change env type – prod to sandbox, sandbox to prod  
Delete env - to recover storage space and to remove personal data  
Recover env - within seven days of deletion  
Reset env – useful when we need to create new project or free up resources and personal data  
Copy env – there are two level of copy 1) Everything, and 2) Customizations and schemas only  
Restriction apply such as:  
- can only copy to an environment in the same tenant and region  
- can’t copy to prod env  
- copy and restore operations can take up to 8 hours – 24 hours

Backup and Restore

System backup – with Dynamic enabled, it stored 28 days, if not, only 7 days

Manual backup – sandbox/dev for 7 days, prod with Dynamic is 28 days

Restoring env - To restore to a production environment, you must first change its type to sandbox.

Administration mode - is useful when you want to make operational changes and not have regular users affect your work, and not have your work affect end users (non-admins).

Table

To create a table, go to <https://make.powerapps.com/>, you must be in correct environment.

Click “Table” menu item from the left menu bar. Select Create. Below are three properties:

* 1. Schema name: This is the internal name of the table created in the database
  2. Type: Standard, Activity, Virtual or Elastic.
  3. Record ownership: User or Team or Organizational

You can’t change these options once a table is created.

Table types:

1. **Standard**: A table where you can store data and add to the navigation in model-driven apps.
2. **Activity**: Store interactions with time dimensions 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.
3. **Virtual**: Connect to data from an external data source such as Microsoft Azure SQL Database. Virtual tables appear in your app to users as regular table rows but contain data that is sourced from an external database dynamically at runtime.
4. **Elastic**: Used when your table includes a large dataset in excess of tens of millions of rows. Elastic tables are powered by Azure Cosmos DB.

Licensing Requirements for each table type

A screenshot of a computer program

Description automatically generated

Tables with complex business logic (**or Complex tables**) such as code plug-ins or real time workflows will require that users have a Power Apps or Power Automate per user license.

**Restricted tables** aren't standard tables within Dataverse but are included in one of the customer engagement apps in Dynamics 365 or a third-party solution. They'll require a Dynamics 365 license.

More details on Licensing Requirements for each table type are [here](https://learn.microsoft.com/en-us/power-apps/maker/data-platform/data-platform-entity-licenses).

More details on Restricted tables requiring Dynamic 365 license are [here](https://learn.microsoft.com/en-us/power-apps/maker/data-platform/data-platform-restricted-entities).

Table Ownership

1. User or team owned: Data belongs to a user or team. Actions that can be performed on these rows can be controlled at the user level.
2. Organization-owned: Data belongs to the organization. Access to the data is controlled at the organization level.

Tips: Dataverse came with many standard tables and recommended using them (by changing the column name if needed) instead of creating a new table. You can also hide the standard tables if not using them. To hide a standard table, change the security role privileges for your organization to remove the Read privilege for that table. This removes the table from most parts of the application.

A screenshot of a computer

Description automatically generated

We can customize Views and Forms for a table.

Create a Dataverse Table:

<https://learn.microsoft.com/en-us/training/modules/get-started-with-powerapps-common-data-service/3-create-a-cds-entity>

Import SharePoint list data into Dataverse Table:

<https://learn.microsoft.com/en-us/training/modules/get-started-with-powerapps-common-data-service/4-import-data-into-your-database>

Create table relationships:

<https://learn.microsoft.com/en-us/training/modules/get-started-with-powerapps-common-data-service/create-table-relationships>

Create custom table, business rule and import Excel spreadsheet data to Dataverse table

<https://learn.microsoft.com/en-us/training/modules/get-started-with-powerapps-common-data-service/4a-use-data-cds-exercise>

Dataverse Logic and Security

Reduce repetitive code in app - Tables within Dataverse can use rich server-side logic and validation to ensure data quality.

* 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

Business rules can be used for data validation, enforce requirement, set value, perform calculation, etc. Business rules depend on the scope of the rule which can be:

1. **Individual form**: The rule applies only to the specified model-driven app form.
2. **All forms**: The rule applies to all model-driven app forms.
3. **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.

Two main components here are Flow and Actions. Flow is the condition check and action is the tasks to perform such as showing error messages, setting field visibility or setting/clearing default value, etc.

Note: Business rules defined for a table apply to both canvas apps and model-driven apps if the table is used in the app BUT in canvas apps, show or hide, enable or disable and creating business recommendations based on business intelligence are not supported.

Dataverse Security

Standard security roles System Administrator, Environment Maker and Basic User. But should create custom security roles because standard security roles cannot be customized for your tables. You can only modify custom security roles.

You can manage security roles from the Power Platform admin center. Select your environment and select Settings, expand Users + permissions and select Security roles.

Dataverse Auditing

Three levels of auditing: Environment, Table and Column.

Dataverse auditing is supported on all custom and most customizable tables and columns. Audit logs are stored in Dataverse and **consume log storage capacity**. You can view audit logs in the Audit History tab for a single record. You can view audit logs in the Audit Summary view for all audited operations in a single environment. Audit logs can also be retrieved using the Web API or the SDK for .NET.

It's possible to refer to audit log data in a model driven app under **Audit history**.

Note that auditing won’t run if is turned off for your environment.

Dua-write vs Virtual tables/entities

Dual-write – is an out-of-box infrastructure (automated data flow) that provides near-real-time interaction between customer engagement apps and finance and operations apps. Dual-write also supports online and offline mode, and it follows the no-code/low-code principle.

Virtual Entities – tables dose not reside in Dataverse. Instead, it continues to reside in the app where it belongs but support CRUD operations. The entities must be made available as virtual entities in Dataverse to support CRUD.

Together, virtual entities and dual write are part of the shared data layer for the convergence of finance and operations apps and the Dataverse platform. They're complementary technologies that are intended to work **together**.

When you must have access to your data offline, consider Dual write. If data is for read-only then consider virtual entities which also save the storage space in Dataverse.

**Primary Column** – a way to identify a record in a table and is not a unique identifier (NOT a Primary Key). The Unique identifier column is a GUID (autogenerated). To make the primary column unique, then create an alternate key (with unique value) and assign the Primary Name column to the new key. You can create up to five keys for each table. All values in the key are unique.

**Choice Column** – Two options: Local Choice vs Global Choice.

Local choices can only be used by the table and column that they're created against and can't be reused on other tables. A global choice is a separate component and can be reused for multiple columns on multiple tables. The list of choices is shared for the table columns that reference the global choice.

**Dataverse Data Export Options** – Beside exporting Excel format, offer both Power Automate and Azure Synapse Link.

Power Automate – use cloud flows with the Dataverse connector to query table rows and generate output files. Power Automate can extract exactly the information that you require in the format that you need.

Azure Synapse Link – after enabling the Azure Synapse Link for Dataverse in the Power Apps maker portal, you choose the tables in Dataverse and Dataverse will perform a continuous replication of data to Azure to either an Azure Data Lake storage Gen2 account or an Azure Synapse Analytics workspace.

Using Dataflow to import data into Dataverse

Create dataflow with destination storage and schedule data refresh. This can be done in Power App Portal under “Dataflows” (may need to click on …More menu option).

Importing data using dataflow link is [here](https://learn.microsoft.com/en-us/training/modules/load-export-data-create-data-views/5a-import).

**Action-based Data Source (Service Name or Action Connectors)**

Differ from the more popular tabular data sources. You use functions to interact with the data source instead of just reading and writing data. But it cannot be used with **Forms Control**. Most action-based data sources provide functions for updating the data as appropriate. An action-based data source can also be for things like sending emails or other notifications, not necessarily only for reading and writing data. Power App supports both tables and action-based data sources.

More on how to work with action-based data source is [here](https://learn.microsoft.com/en-us/training/modules/connect-to-other-data-in-powerapps-canvas-app/2-work-with-action-based-data-sources).

**Power Automate is a companion of Power App**

Power Automate helps you to pull data from many different tables and action connectors, so it's a valuable companion for Power Apps. Power Automate is a connector to Power App.

Use case: Send email when someone submit form – this can be done using Power App, but

Use case: Start approval process after sending email – this is where Power Automate comes in

Explanation: You can have Power Apps trigger a Power Automate flow when the user submits the data. Power Automate can then look up who the user's manager is and send the manager an approval request. Power Automate will then help get a response from the manager, update the data source with the status based on their response, and send the original submitter an update.

User case: Consuming third-party API that return complex JSON data and need to purify before storing it to our table. Power Automate can do that and store data to table and then Power App will take care of displaying or working with those data further.

Power App  
is a Trigger Point

Trigger (with params)

Power Automate  
take those params and perform action on the flow

More use cases can be found [here](https://learn.microsoft.com/en-us/training/modules/connect-to-other-data-in-powerapps-canvas-app/3-flow-and-powerapps).

## Manage permissions and administration for Dataverse

More on Environment - Each environment can have zero or one instance of a Dataverse database, and it serves as a collection of objects such as tables, flows, apps, connections, and security roles. Typically, an organization might use multiple environments to support different groups of users simultaneously. This setup allows you to limit who can access specific data, apps, and workflows in each environment. One common practice is to create separate environments for different stages of application lifecycle management (ALM), such as development, testing, and production. In production environment, you configure permissions to allow access to the data, apps from Microsoft Power Apps, and Microsoft Power Automate workflows.

Security Roles – Go to environment -> Settings -> Security Roles.

After you define a security role in the environment, you can:

* Associate it directly with users.
* Associate it with Dataverse business units and teams.

Individually assigning security roles to users can be a tedious process for groups of users. Instead, you can associate a security role with a team and then add all users to the team. After you assign a user to the team, they receive the security role's privileges. Often, organizations use this approach to accelerate the assignment of security roles to groups of users. A key concept of Dataverse security is that privileges are accumulative across all assigned security roles, with the highest privilege access winning.

List of actions can perform are: Create, Read, Write, Delete, Append, Append To, Assign, Share. These actions make up the ***privileges*** of a security role.

Business Units - Go to environment -> Settings -> Business Units.

Business units provide security and structure for grouping users. Often, you would use business units to mimic an organization’s departmental structure (but they don't need to). Business units are hierarchical, and every Dataverse instance has a single root business unit. Beneath the root business unit, you can create child business units to further segment and secure your data.

Important: If you move a user to a different business unit, you need to reassign their security roles. Each user is a member of only one business unit, but a team can have user members from multiple business units.

Access Level – Each privilege (Create, Read, etc.) can be granted different access levels.

* None – No access
* User – access to record owned by users, shared with them or their teams
* Business Unit – access to records in user’s business unit. Because this access level gives access to information throughout the business unit, it should be restricted to match the organization's data security plan. For managers with authority over the business unit.
* Parent: Child Business Unit – access to records in users’ business units or all subordinate units. This level gives access to information throughout the business unit and subordinate business units. For managers with authority over the business units.
* Organization – access to all records in the organization regardless of the business unit hierarchical level they belong to. For managers with authority over the organization.

**Tips: Together the privileges and access levels make up role-based access control in Dataverse.**

Teams – important security building blocks. Business units own teams. Every business unit has a default team that the system automatically creates when you create the business unit. Dataverse manages members of the default team, which contains all users who are associated with that business unit. You can’t add or remove members from the default team manually. Instead, the system dynamically adjusts the members as you add new users to or remove them from business units.

Tips: Users can be members of multiple teams, which allows for a powerful way of granting permissions to users without micromanaging access at the individual user level.

Hierarchy Security – In addition to business units, you can help secure Dataverse by using one of the out-of-the-box hierarchical security models. The hierarchy security model extends Dataverse security. Two security models that you can use for hierarchies are:

* Manager hierarchy - To access data the subordinate's data, a user must be in the same business unit as the subordinate or in the parent business unit of the subordinate’s business unit. You can establish the manager and subordinate relationship by using the Manager field on the User table.
* Position hierarchy - Allows data access across business units. With position hierarchy security, you can define various job positions in the organization and arrange them in a hierarchy by using the Position table. Then, you can add users to any given position by using the Position lookup column on the User record.

To configure “Hierarchy Security”, Go to environment -> Settings -> Users + permissions -> Hierarchy security. Then select “Enable Manager hierarchy Model” option or “Enable Position hierarchy Model”, you may click “Configure” hyperlink next to it to configure one of the selected hierarchy models.

Tips: For performance, avoid creating many business units. Instead, create fewer business units and add hierarchy security.

Table/record ownership – Dataverse supports two types of record ownership: organization owned, and user or team owned. Record ownership is a choice that happens when you create the table, and you can't change it afterward.

For organization-owned records, the only access level choices are that the user can do the operation, or they can’t. For user and team-owned records, the access level choices for most privileges are: (1) Organizational, (2) Business unit, (3) Business unit and child business unit, (4) Only the user’s own records.

Record-level security in Dataverse – A user's access to a record is the combination of all their security roles, the business unit that they're associated with, the teams that they're members of, and the records that are shared with them. All access in Dataverse is accumulative across all of those concepts in the scope of a Dataverse database environment. These entitlements are only granted in a single database and are individually tracked in each Dataverse database, requiring a user to have an appropriate license to access Dataverse.

Column-level security to control access - You can turn on column-level security on a column-by-column basis. Then, you can manage access by creating a Column Security Profile. The profile contains all columns that have column-level security turned on and the access granted by that specific profile. You can control each column in the profile for Create, Update, and Read access. Then, you would associate Column Security Profiles with a user or teams to grant users with those privileges to the records that they already have access to.

Important: A user must already have access to the record (row-level) for the Column Security Profile to grant them access to the columns.

Configure user’s environment security - After you create roles, teams, and business units in an environment, you can assign users with their security configurations.

1. When you create a user, associate the user with a business unit. By default, this business unit is the root business unit in the organization. The system also adds the user to the default team of that business unit.
2. Assign security roles that the user needs and/or add them as members of certain teams. Remember, teams can also have security roles, and the effective rights of the user is the combination of directly assigned security roles with those of any teams that the user is a member of.

Additionally, if you use column-level security, you need to associate the user, or a team of the user, to a Column Security Profile that you created.

## Get started with security roles in Dataverse

A security role has certain privileges associated with it, and you can associate a user with one or many security roles.

Environment roles – Environments have two predefined roles that provide access to permissions within an environment.

* Environment admin - Before a Dataverse database is added to the environment, the Environment Admin role can perform all administrative actions on an environment.
* Environment Maker - can create resources within an environment including apps, connections, custom connectors, gateways, and flows using Power Automate.

The following rules apply to members of the ***Environment Maker*** role:

* Environment Makers can distribute the apps that they build in an environment to other users within an organization. They share the app with individual users, security groups, or all users in the organization.
* Users or groups that are assigned to these environment roles aren't automatically given access to the environment's database (if it exists). They must be given access separately by a Database owner.
* Whenever a new user signs up for Power Apps, they're automatically added to the Maker role of the default environment.

You assign users to one of these two roles when considering what permissions, you want to give to a user in an environment. However, if the environment has a Dataverse database, more roles are added, and the permissions options broaden.

Environments with a Dataverse datastore - When an environment has a Dataverse datastore, users must be assigned the System Administrator role instead of the Environment Admin role for full admin privileges.

Important: Users who make apps that connect to Dataverse and need to create or update table and security roles need to be assigned the System Customizer role in addition to the Environment Maker role. This is necessary because the Environment Maker role doesn't have privileges on the environment's data.

Refer following table for summary of resources available for predefined security roles.

A screenshot of a computer

Description automatically generated

Permission Settings - Configuration of table permissions can also be done with predefined groups of permissions. Select a table, and then select Permission Settings in the command bar or select More Actions (…) > Permission Settings.

A screenshot of a computer

Description automatically generated

Beside permission settings on a table, “Miscellaneous privileges” (such as Bulk Edit, Merge, etc.) and “Privacy related privileges” (such as Export to Excel), can also be configured.

Tips: Privacy related privileges only use the None or Organization access levels. Most miscellaneous privileges only use the None or Organization access levels but there are some miscellaneous privileges that can have all access levels set.

Configure Dataverse team for security - Using Microsoft Dataverse teams is optional. However, teams provide an easy way to share business objects and let you collaborate with other people across business units. Although a team belongs to one business unit, it can include users from other business units. You can associate a user with more than one team.

Team Types - can be one of the following: Owner, Access, Microsoft Entra ID Security group, or Microsoft Entra ID Office group. Go to environment -> Settings -> Users + permissions -> Teams.

* Owner team: an owner team owns records and has security roles assigned to the team. A user's privileges can come from their individual security roles, those of the teams that they're part of or the ones they inherit. Team members are added manually to the owner team.
* Access team: an access team doesn't own records and doesn't have security roles assigned to the team. The team members have privileges defined by their individual security roles and by roles from the teams they're members of.
* Microsoft Entra ID group team: Like owner teams, a Microsoft Entra ID group team can own records and can have security roles assigned to the team. Security and Office are two group team types, and they correspond directly to Microsoft Entra ID group types. Team members are dynamically derived (added and removed) when they access an environment based on their Microsoft Entra ID group membership.

The administrator can create Microsoft Entra ID group teams that are associated to the Microsoft Entra ID groups in each of the Dataverse environments. Then they can assign a security role to these group teams. For each Microsoft Entra ID group, the administrator can create group teams based on the Microsoft Entra ID group Members, and/or Owners, or Guests, and assign a respective security role to each of these teams. When members of these group teams access these environments, their access rights are automatically granted based on the group team's security role.

Provision and deprovision users - Once the group team and its security role are established in an environment, user access to the environment is based on the user membership of the Microsoft Entra ID groups. When a new user is created in the tenant, all the administrator needs to do is assign the user to the appropriate Microsoft Entra ID group, and assign Dataverse licenses. The user can immediately access the environment without the need to wait for the administrator to assign a security role.

When users are deleted or disabled in Microsoft Entra ID or removed from the Microsoft Entra ID groups, they lose their group membership. These users won't be able to access the environment when they try to sign in.

Important: The memberships for the user's Microsoft Entra ID groups and Dataverse group teams are synchronized, and the user's access rights are dynamically derived at run time.

## Use administration options for Dataverse

Tenant Storage Capacity - Dataverse storage is broken into three sources:

* **Database:** Refers to the data that is being stored in tables in your Dataverse database. This includes item like out-of-the-box tables, custom tables, and more.
* **Log:** Refers to log files and that are being used to capture information such as audit logs, or plug-in trace logs.
* **File:** Refers to images or files that are being stored in your Dataverse environment such as a customer’s picture stored on their contact record.

Dataverse Search - delivers fast and comprehensive search results in a single list, sorted by relevance. With Dataverse search enabled, a search box is always available at the top of every page in all the model-driven apps in the environment.

Search Index - The search index defines which tables appear in search results. You can manage search index from the **Overview** tab within a solution.

Auditing - Data change auditing, is configured at the Dataverse environment, table, and column level. Enabling and disabling of auditing can only be done by someone in the System Administrator or System Customizer role.

Auditing settings for tables and columns are solution aware and will be transported with the schema as the solution is installed in another environment. However, each environment must have auditing enabled at the environment level for these settings to perform as expected.

Tips: If auditing is disabled at the environment level, no auditing data is captured, even if auditing is configured for tables and columns in the environment.

Go to environment -> Settings -> Audit & Logs -> Audit settings

Enabling auditing on tables and columns - By default, new Dataverse tables that are created have auditing disabled. You can enable or disable auditing for tables using the classic solution explorer. When auditing is enabled for a table, auditing is enabled for all eligible columns of that table. When editing a column, you can find the auditing switch under the advanced options section on the column properties.

Important: After changing any of the column audit configurations to enable or disable, you must publish changes for the table or publish all changes.

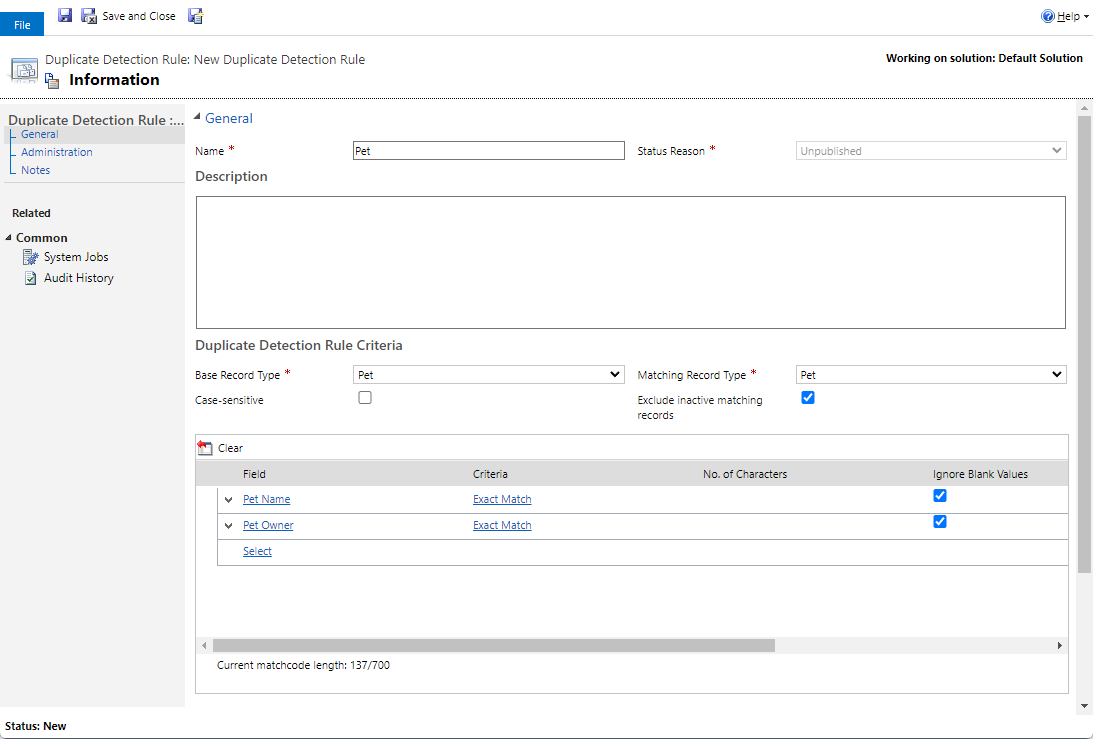
Duplicate Detection - Microsoft Dataverse includes basic duplicate detection functionality. Duplicate detection must be enabled in the environment's settings in the Power Platform admin centre under the **Data management** section.

Important: To detect duplicates for a table's rows, you must first edit the table's properties in the Power Apps maker portal and enable the Apply duplicate detection rules option.

Configure duplication detection rules - To detect duplicates, you need to define rules to determine potential duplicate data. You can only create duplicate detection rules for tables that have been enabled for duplicate detection. Duplicate detection rules are managed in the environment's settings in the Power Platform admin center under the Data management section.

To create a duplicate detection rule: Select the base and matching tables. The matching table defaults to the selected base table but can be changed. Select column and then Select matching criteria.

The matching criteria you can select is simple.



Tips: You should select **Exclude inactive matching records** to prevent duplicates being found for deactivated rows and you should select **Ignore blank values** to prevent multiple blank to blank duplicates being detected.

Duplicate detection jobs **-** Duplicate detection jobs run as scheduled background jobs to find duplicates on a table. Duplicate detection jobs are managed in the environment's settings in the Power Platform admin center under the **Data management** section. Duplicate detection jobs are created by using a wizard.

Bulk Delete - Bulk delete is managed in the environment's settings in the Power Platform admin center under the Data management section. Dataverse has many bulk delete jobs to clean up its own data.

Long term data retention - capability of Microsoft Dataverse that enables you to transfer your data from a Dataverse datastore to a managed data lake and securely retain unlimited data long term in a cost-efficient way. Dataverse long term data retention requires a **Managed** environment. Once data is retained as long term (inactive) data, it can't be moved back to the active data store. Set up retention policies by defining criteria for a given data table. The retained data is never moved out of Dataverse, it's stored in a Dataverse managed data lake. The data is always secured with Dataverse security backed by Microsoft Entra ID. Enable long term data retention in a table's properties in the Power Apps maker portal. Then you can create a data retention policy in the Power Apps maker portal. When creating a data retention policy, select a table, choose an existing table view, and then define the schedule as Daily, Weekly, Monthly, or Yearly. To view retained data you require organizational access level read privilege on the table.

## Visualize data with Dataverse views

Dataverse table views - three types of views are:

* Personal - This view is owned by individuals and is visible to other users whom it's shared with.
* System - Special views that Power Apps applications depend on. They're automatically created, and only users with the system administrator or system customizer security role can edit them. System views are defined as follows:
  + Quick Find - Use for searches in Power Apps applications.
  + Advanced Find - Use for searches in Power Apps applications.
  + Associated - Use to list records that are associated with another table's record.
  + Lookup - This system view is for lookup fields.
* Public - This view is created by individuals but is accessible by all app users. It's useful for organization-wide scenarios.

These views are made ready to use in power apps, both model-driven and canvas app. In model-driven app, user can create a personal view, but it needs to share so that other users can access.

## Use Power Query to load data in Dataverse

Power Query - provides a complete solution to transform and prepare data by using a graphical user interface (GUI). Two versions of Power Query are available, one for desktop and one online. In the context of Microsoft Power Platform, the applicable version is Power Query Online and is experienced via the **Dataflow** option. Two steps required when considering using Power Query to import data are:

1. Get Data – Based on the data source, this may require having proper connection credentials.
2. Transform Data – This performs various transformations, such as grouping and merging.

Dataflow - create a new Dataflow from a source (can be CSV uploaded in OneDrive or Share Point list using connectors), and transform data, select table, map data columns and load. Final step will publish the dataflow. Make sure to set appropriate refresh data setting (refresh manually or automatically).

Refresh Data - A dataflow can update the destination table based on updates of the source table or file. To properly match rows from the two sets, a unique ID column must be part of both tables. In many cases, this column is defined as an alternate key in the destination table. In Dataverse, alternate keys are configured via the **Keys** option in the **Schema** section of the table.

For dataflows that are required to be processed automatically, select **Refresh automatically**, and then configure **Frequency-based refresh** or **Refresh on specific days and times** based on the business scenario.

For data sources that have large quantities of data, you may want to change the **Incremental Refresh** settings that are configured to process a full refresh by default. To access this configuration, select Edit incremental refresh for the specific dataflow.

Tips: To use incremental refresh, you must provide date/time columns (i.e., your dataflow should include date/time columns to compare against with destination table).

### Use Microsoft Word and Excel templates with Dataverse

Dataverse allows to use Word and Excel templates to address following scenarios:

* Provide a work order detail to a field technician.
* Generate an insurance policy certificate for a customer.
* Share a prospect list with a colleague.

To use the native Word and Excel templates integration with Dataverse, you'll need to first enable the integration through the Power Platform admin center. Go to **Environment -> Settings -> Templates -> Select Document templates**. Upload the template here. To create template, first download the base template (with XML data to customize later) via Power App list view -> select one record screen. The top ribbon bar should have “Word Template”. Hands-on [tutorial here](https://learn.microsoft.com/en-us/training/modules/use-word-excel-dataverse/create-document). More can be found [here](https://learn.microsoft.com/en-us/training/modules/use-word-excel-dataverse/exercise-create-work-order) and [here](https://learn.microsoft.com/en-us/training/modules/use-word-excel-dataverse/exercise-create-sales-forecast).

## Export data from Dataverse and use Microsoft Excel to edit records

From a model-driven app, you can select a view, or you can filter rows from a view, add columns to it, and then export the records to Excel. Different export options are available:

Open in Excel Online: useful when a user wants to manipulate data directly in Excel with simple synchronization between the Excel spreadsheet and the source Dataverse table.

Static Worksheet and Static Worksheet (Page only): useful when a user wants to share Dataverse records with users who don't have access to Dataverse, such as partners or service providers. Page only option generates record for current page view and if not, static worksheet can generate up to 100,000 rows. Making changes won’t sync back to Dataverse table. The Excel table includes three hidden columns that you can use as a reference when importing the file to update Dataverse. Don't modify those values.

Dynamic Worksheet: a user wants to manipulate or use current and updated data for analysis in Excel. The security permissions and restrictions will be applied when the Excel file is opened, based on the user's profile in Dataverse. Record won’t be visible if user has no access rights to that record. Same three hidden columns that should not touch. Can add/remove columns to be exported without modifying view.

Dynamic PivotTable: same as Dynamic Worksheet but PivotTable isn't automatically created in the Excel spreadsheet. Doesn't include the three hidden columns that you can use as a reference. Because the three required hidden columns aren't part of the generated file, the Import from Excel option will generate an error message stating that this format isn't supported.

Use Azure and external tools to manipulate data

Azure Synapse Link for Dataverse - Microsoft Azure Synapse Analytics is a service that’s designed for enterprise big data analytics. Combined with Microsoft Azure Synapse Link for Dataverse, Azure Synapse Analytics enables an organization to get near real–time insights over Dataverse.

First, through the Microsoft Azure portal, you'll need to create a Synapse workspace prior to linking a Dataverse database to it.

Second, create a new Azure Synapse Link for Dataverse in the Microsoft Power Apps portal. Select **+ New link** in the **Azure Synapse Link** section of **Dataverse**. Then, select the storage account configuration to connect to the Azure Synapse Analytics workspace (workspace was created in Azure portal before).

Third, Select the tables to connect with Azure Synapse Analytics, and then select Next to create the link. As the link is created and Dataverse tables are synchronized, the Sync status for each table will change to Active, which means that the process is complete. These tables are automatically created in Azure Synapse Analytics. Different Microsoft analytics solutions are readily available such as:

* **Power BI** - Visualize Dataverse data in highly interactive dashboards and reports.
* **Dynamics 365 Customer Insights - Data** - Create a 360-degree view of your customer data.
* **Azure Synapse** - Benefit from the data warehousing and big data analytics features of Azure Synapse.
* **Azure Data Factory** - Process complex data transformation and integrate with other data silos.
* **Azure Databricks** - Take advantage of extensive AI solutions.

XrmToolBox – community-based, external tool (a Windows application) for data manipulation in Dataverse database. Refer here for [instructions](https://learn.microsoft.com/en-us/training/modules/understand-azure-dataverse/data-manipulation-tools).

Create relationships, business rules, calculations, and rollups in Dataverse

Two types of relationships are supported. One to many and Many to many.

One-to-many relationship (which is also called 1:N or parent-child) includes a primary (parent) table that can be associated to many other related (child) table rows by using a lookup column on the related (child) table. The primary row is the parent and the related table rows are called child rows.

Creating **a lookup column** creates a many-to-one relationship. Correspondingly, creating a **one-to-many** relationship creates a lookup column on the related table.

Many-to-many relationship (which is also called N:N) includes a special third table called a relationship table, sometimes called an intersect table, which maps how the many rows of one table can be related to the many rows of another table.

The intersect table is not visible to users. More importantly, you cannot add columns to the intersect table or trigger workflow or Power Automate cloud flows when rows are associated, or disassociated, with each other. Not all tables can be used with many-to-many relationships. If the table is not listed in the designer, you cannot create a new many-to-many relationship with this table. You cannot edit the tables in a many-to-many relationship after it has been created; you can only delete it.

Behaviors & Actions - several kinds of **behaviors** that can be applied when certain **actions** occur. Actions are **controlled** by the behaviors. List of all behaviors with descriptions:

* Cascade All – Perform the action on all related table rows.
* Cascade Active – Perform the action on all active related table rows.
* Cascade User Owned – Perform the action on all related table rows owned by the same user as the primary table row.
* Cascade None – Do nothing.
* Remove Link – Remove the lookup value for all related rows.
* Restrict – Prevent the primary table row from being deleted when related table rows exist.

List of **actions** that are controlled by behaviors are:

|  |  |  |
| --- | --- | --- |
| Action Name | Description | Behaviors |
| Delete | What should happen when the primary table row is deleted? | Cascade All, Remove Link, Restrict |
| Assign | What should happen when the primary table row is assigned to someone else? | Cascade All, Cascade Active, Cascade User-owned, Cascade None |
| Reparent | What should happen when the lookup value of a related table in a parental relationship is changed? | Cascade All, Cascade Active, Cascade User-owned, Cascade None |
| Share | What should happen when the primary table row is shared? | Cascade All, Cascade Active, Cascade User-owned, Cascade None |
| Unshare | What should happen when a primary table row is unshared? | Cascade All, Cascade Active, Cascade User-owned, Cascade None |
| Merge | What should happen when a primary table row is merged? | Cascade All, Cascade None |
| Rollup View | What is the desired behavior of the rollup view associated with this relationship? | Cascade All, Cascade Active, Cascade User-owned, Cascade None |

Types of Behavior – to simplify setting relationship behaviors, can group the behaviors into types:

Parental – Any action taken on a row of the parent table is also taken on the related child table rows.

Referential, Restrict Delete – Any related rows can be navigated to. Actions taken on the parent row won't be applied to the child row, but the parent row can't be deleted while the child row exists.

Referential, Remove Link – Any related rows can be navigated to, and actions taken on one won't affect the other. This is the default.

Custom – The behavior for each possible action can be selected.

Hierarchical relationships - Some standard tables in Dataverse already have hierarchies defined such as Account. The rows in a hierarchy can be displayed visually in model-driven apps.

The option to define a hierarchical relationship **is only available for self-referential relationships**. A self-referential relationship is a One-to-many relationship between a table and itself where the same table is defined as both the primary and the related table.

Connections - a way for users to record associations **between almost any two rows** and provide additional information to describe how the records are linked to one another. With Dataverse, you can **define connections between table rows** without creating a table relationship.

For example, connections between accounts and contacts might include employees, board members, shareholders, customers, suppliers, etc. Some connections can also be reciprocal, such as child and parent, husband and wife, or doctor and patient.

To allow users to make connections to a table's rows, enable “**Can have connections**” on the table's properties.

Connection roles – are used to name or describe the relationship between the two rows. Dataverse includes connection roles such as Employer and Former Employer. You can delete these connection roles and create your own connection roles.

Define and create business rules in Dataverse

Business rules – are declarative logic and validation that is associated with a Dataverse table without writing code. The logic defined by a business rule can be applied client-side within a model-driven app form and/or server-side when a Dataverse row is created or updated. In server-side, logic is performed during the transaction before the data is saved into the database. Therefore, the conditions and actions in the rule applies to Dataverse data changed by model-driven apps, canvas apps, Power Pages sites, Power Automate cloud flows, or the Dataverse APIs.

Scope – defines when the logic in the business rule is applied such as Entity, All forms or Specific form. To use a business rule with a canvas app you should set the scope to **Entity**. Business rules can only perform actions on table columns on the form and cannot change the properties of sections or tabs. Business rules are not applied in the form's save event. When the business rule's scope is set to Entity, the rule's conditions are evaluated when the row is saved. The logic is performed during the pre-operation stage of the database transaction. A business rule that shows an error message will prevent the save and rollback the database transaction.

Components of a business rule - encapsulate logic in a predefined set of steps. The steps run each time data is entered or modified, and the data meets certain criteria to trigger the business rule. Each rule is made up of **conditions** and **actions**.

* Conditions – all business rules start with a condition. Conditions are shown in purple in the business rule editor. A condition is always evaluated as either true or false. Within a condition, you can add multiple clauses with either AND or OR logic.
* Actions - a logic step that runs on either the true or false branch of a condition. Actions are shown in green in the business rule editor. You can chain multiple actions by adding a new action.

Create relationships, business rules, calculations, and rollups in Dataverse