

## Power Platform Overview

### What is Dataverse?

Microsoft Dataverse is a cloud-based data platform that allows you to securely store and manage data used by business applications. It provides a standardized way to organize information and is tightly integrated with Microsoft Power Platform (Power Apps, Power Automate, Power BI, Copilot, etc.) and Dynamics 365.

Dataverse is like the **database** 'behind the scenes' for low-code/no-code solutions, giving consistency, security, and scalability.

### Core Elements

- **Tables:** Logical containers for data, like spreadsheets or database tables. For example, Accounts, Contacts, Invoices.
- **Columns (Fields):** Define attributes of a table. For example, in Contacts → First Name, Email, Phone.
- **Rows (Records):** Each entry in a table, similar to a row in Excel. For example, a single customer record in the Contacts table.
- **Relationships:** Define how tables connect. Example:
  - One-to-Many: One Account can have many Contacts.
  - Many-to-Many: Contacts can be linked to many Opportunities.

### What is the Common Data Model (CDM)?

The Common Data Model is a standardized, shared definition of business entities (like Account, Customer, Invoice) used across Microsoft apps and services.

It ensures consistency across systems by using common naming, data types, and relationships.

Example: Both Dynamics 365 Sales and a custom Power App can use the same Contact table format, making data portable and easier to integrate.

### Business Rules

Business Rules in Dataverse allow you to apply logic and enforce consistency without code.

They can:

- Validate data before saving (e.g., Discount cannot exceed 20%).
- Show/hide fields.

- Set default values.
- Make fields required.

These run client-side in apps or server-side when saving data.

### Power Fx

Power Fx is the low-code formula language of Power Apps, inspired by Excel.

Used for UI logic, data manipulation, and expressions.

Example: `If(Quantity > 100, "Bulk Order", "Standard Order")`

Makes apps dynamic by controlling visibility, calculations, and user interactions.

### Power Automate

Power Automate allows you to create workflows (flows) that automate tasks.

Integrates Dataverse with Microsoft 365, Dynamics, and third-party apps.

Example use cases:

- When a new row is created in Dataverse → Send a Teams notification.
- On record update → Run approval workflow.
- Daily schedule → Export data from Dataverse to SharePoint.

### Importing and Exporting Data

- **Import:** Data can be imported into Dataverse via Excel, CSV, Power Query, or dataflows. Useful for migrations, integrations, or seeding initial data.

- **Export:** You can export Dataverse data directly to Excel, Power BI, or external systems. Features like Export to Data Lake or Export to Azure Synapse allow large-scale reporting and analytics. Dataflows support ongoing synchronization with external sources.

### Power Platform Security Model

The security model in Power Platform (and Dataverse) is designed to ensure data integrity, controlled access, and compliance. It is layered and flexible:

- **Authentication:** Handled by Microsoft Entra ID (formerly Azure AD). Users must be authenticated before accessing apps or data.
- **Authorization:** Controlled through security roles and permissions.
  - Security Roles: Define what actions a user can perform (read, write, append, share, delete) on specific tables.
  - Privileges: The specific rights within a security role (e.g., read on Contacts).

- Access Levels: Define scope (user-level, business unit-level, organizational-level).
- Business Units: Logical divisions for structuring security (like departments or regions).
- Teams: Users can inherit permissions through team membership.
- Field-level & Row-level Security: Restrict access to specific fields or rows.

Key takeaway: The model ensures the right people see and do the right things—nothing more, nothing less.

## Power Platform Environments

An environment is a container for apps, data, flows, and resources in Power Platform. They provide isolation, governance, and lifecycle control.

### Types of Environments:

- Default Environment: Created automatically, available for all users. Best for personal productivity, not enterprise apps.
- Sandbox: Used for development and testing. Can be reset or copied without impacting production.
- Production: Stable environments intended for live apps and data.
- Trial: Temporary environments for evaluations or POCs.

### Key Points:

- Each environment has its own Dataverse database (optional).
- Security, apps, and flows are environment-specific.
- Environment strategy is crucial for governance (e.g., Dev → Test → Prod).

## Power Platform Administrative Portals & Experiences

Admins have multiple portals to manage the platform:

- Power Platform Admin Center ([admin.powerplatform.microsoft.com](https://admin.powerplatform.microsoft.com)): Manage environments, security roles, capacity, analytics, and policies.
- Microsoft 365 Admin Center: User and license management.
- Dataverse Security Settings: Configure roles, teams, auditing, business units.
- Power BI Admin Portal: Manage Power BI workspaces, governance, and monitoring.
- Microsoft Entra Admin Center: Identity and authentication, conditional access, MFA policies.

Experience: Admins use these portals to balance governance and empowerment—giving makers tools to build while ensuring compliance.

## Application Lifecycle Management (ALM) in Power Platform

ALM is the process of managing apps from idea → build → test → release → maintain. Power Platform provides both manual and automated (DevOps) approaches.

Key Components:

- Solutions: Packaging mechanism for apps, flows, tables, and components.
  - Unmanaged → for development (editable).
  - Managed → for deployment (locked, production-ready).
- Environment Strategy: Dev → Test → Production (separation ensures stability).
- Version Control: Track changes to solutions (GitHub/Azure DevOps integration available).
- Automation: Power Platform Build Tools, GitHub Actions for CI/CD, automatic deployment, validation, and rollback.

Benefit: ALM ensures apps are delivered reliably, consistently, and securely across the organization.