WORKSHOP

All-In! A Deep-Dive On Model-Driven Power Apps

Data Modeling & Dataverse

Part I

PPCC 2025

Data Modeling Concepts 🧠

Importance of Data Architecture before Data Modeling

Data architecture defines how data supports a business's strategic objectives.

It is important to understand the "who/what/where/why" (data architecture) before applying the tactical process of "how" (data modeling).

Important to answers these types of who/what/where/why questions:

- *What data is needed?*
- *How is the data used?*
- *Where is data stored?*
- *Who needs access to this data?*

Categorizing Data Helps Inform Data Modeling

Master Data
Usually represents real world entities. Data is controlled and maintained for accuracy. Low to medium volumes typically.

Examples:
- Customers
- Vendors
- Products

Transactional Data
Records of business activities or events, typically related to master data. High volume of data.

Examples:
- Purchase Orders, Invoices,
- Inspections, Service Calls
- Bank Transactions

Reference/Configuration Data
Used for classification and categorization typically. Often required to configure a system for operational use. Data rarely changes and often used as “lookup” tables.

Examples:
- Tax Codes
- Cities
- Vehicle Model

Inferred Data
Usually generated automatically and based on predictions or insights. Lower accuracy certainty as a result

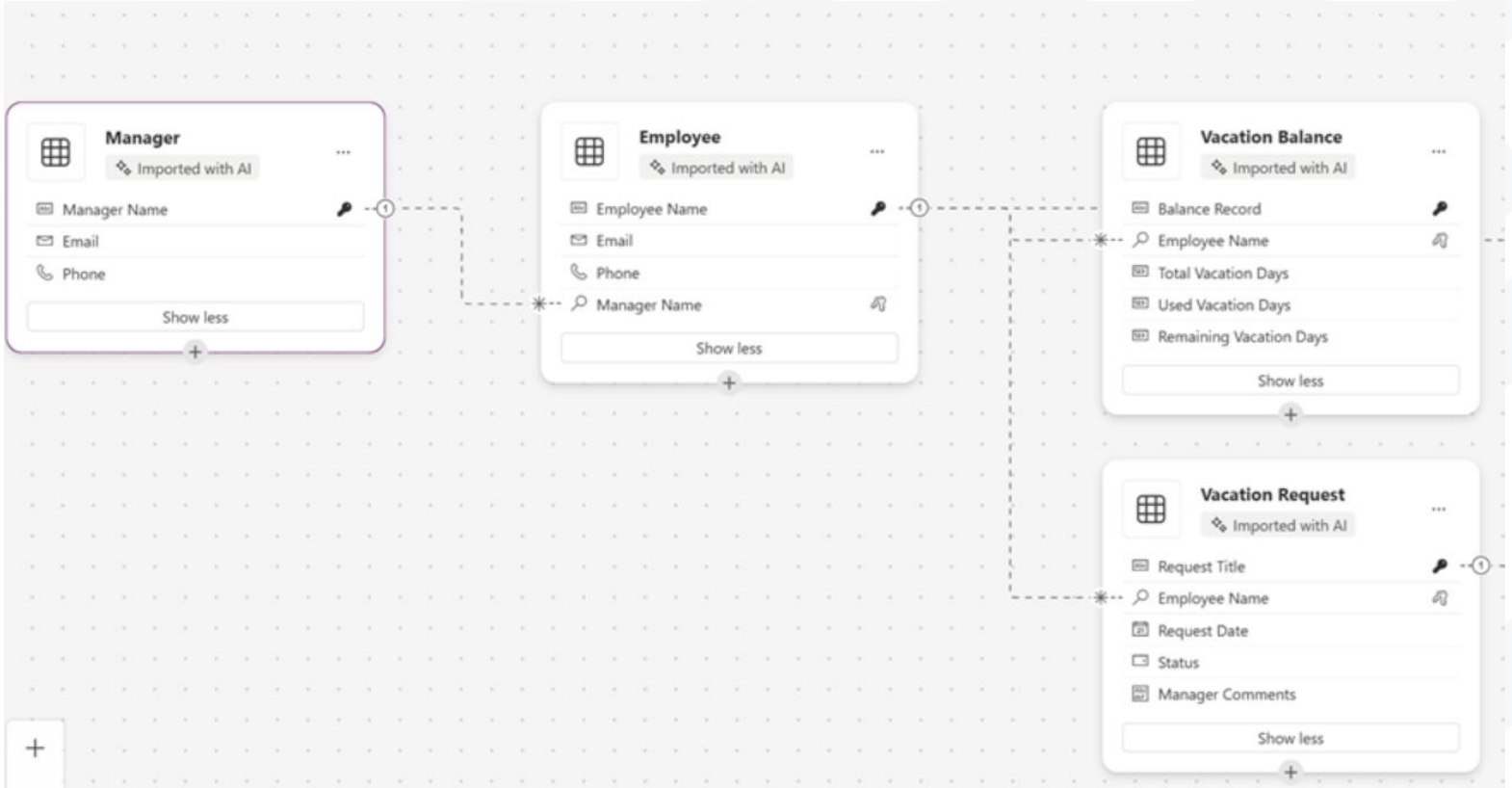
Examples:
- Social Media posts
- Credit Score
- Segmentation

Data Modeling

Data modeling is a tactical level of design that defines detailed data flows, structure, and relationships.

A data model is a visual representation of the detailed design.

An **entity relationship diagram (ERD)** is the output that depicts the design showing table structures and relationships between them.



Principals of Good Data Models

- **Accuracy** – reflects the business domain and terminology
- **Clarity** – tables & columns are intuitive and self-describing
- **Flexibility** – able to adapt to future business needs
- **Performance** – optimized for expected workloads (transactional, analytical)
- **Security** – aligns with access requirements and regulatory compliance in mind

Metadata and Tables

Metadata

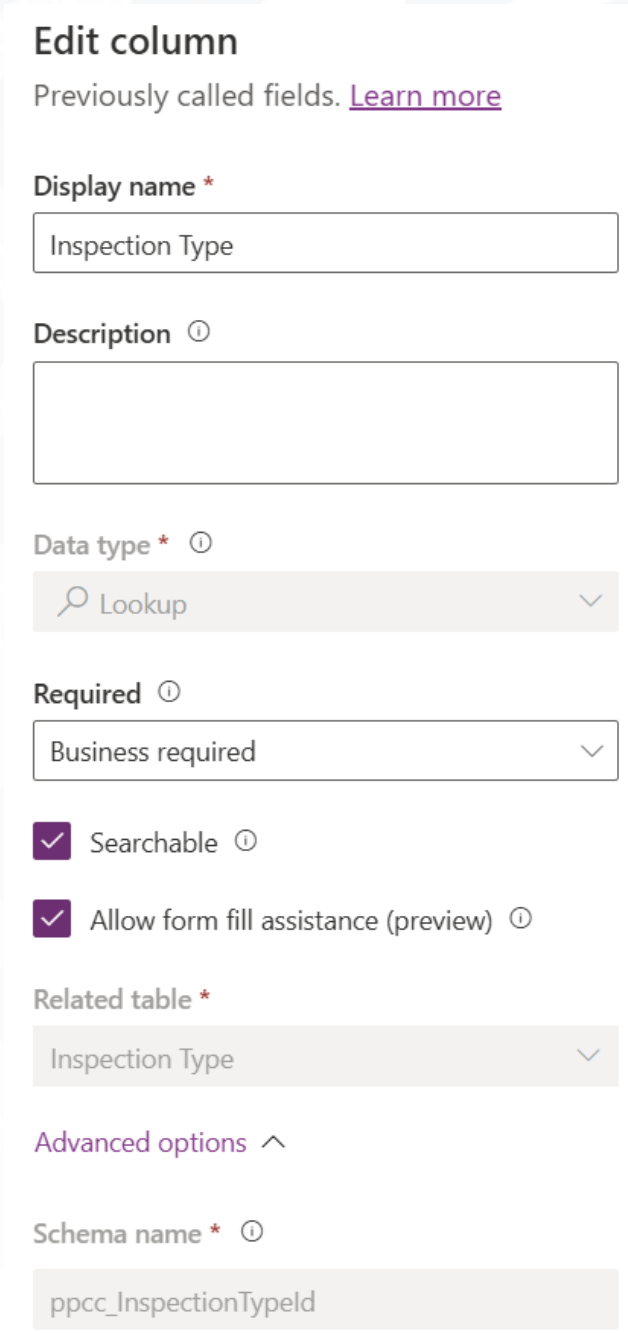
"**Data about the data**"

Describes the definition, structure, and behavior of components across Dataverse.

Visual designers in Dataverse allow for easy configuration of metadata.

Examples:

- Tables (display name, schema name, etc.)
- Columns (display name, data type, etc.)
- Forms (name, description, sections, tabs, etc.)
- Apps (name, app type, etc.)
- Workflows (name, workflow type, status, etc.)



Tables

Models and manages your business data.

There are four types of tables:

- **Standard** – Most commonly used for storing relational datacustom and out of the box tables such as Account, Contact, etc
- **Activity** – Best used for interactions (e.g. phone call, email) and schedule-based activities (e.g. task, appointment, meeting)
- **Virtual** – Interact with data in an external source as if it was natively in Dataverse.
- **Elastic** – Designed for high volume semi-structured data, backed by Azure Cosmos DB under the hood.

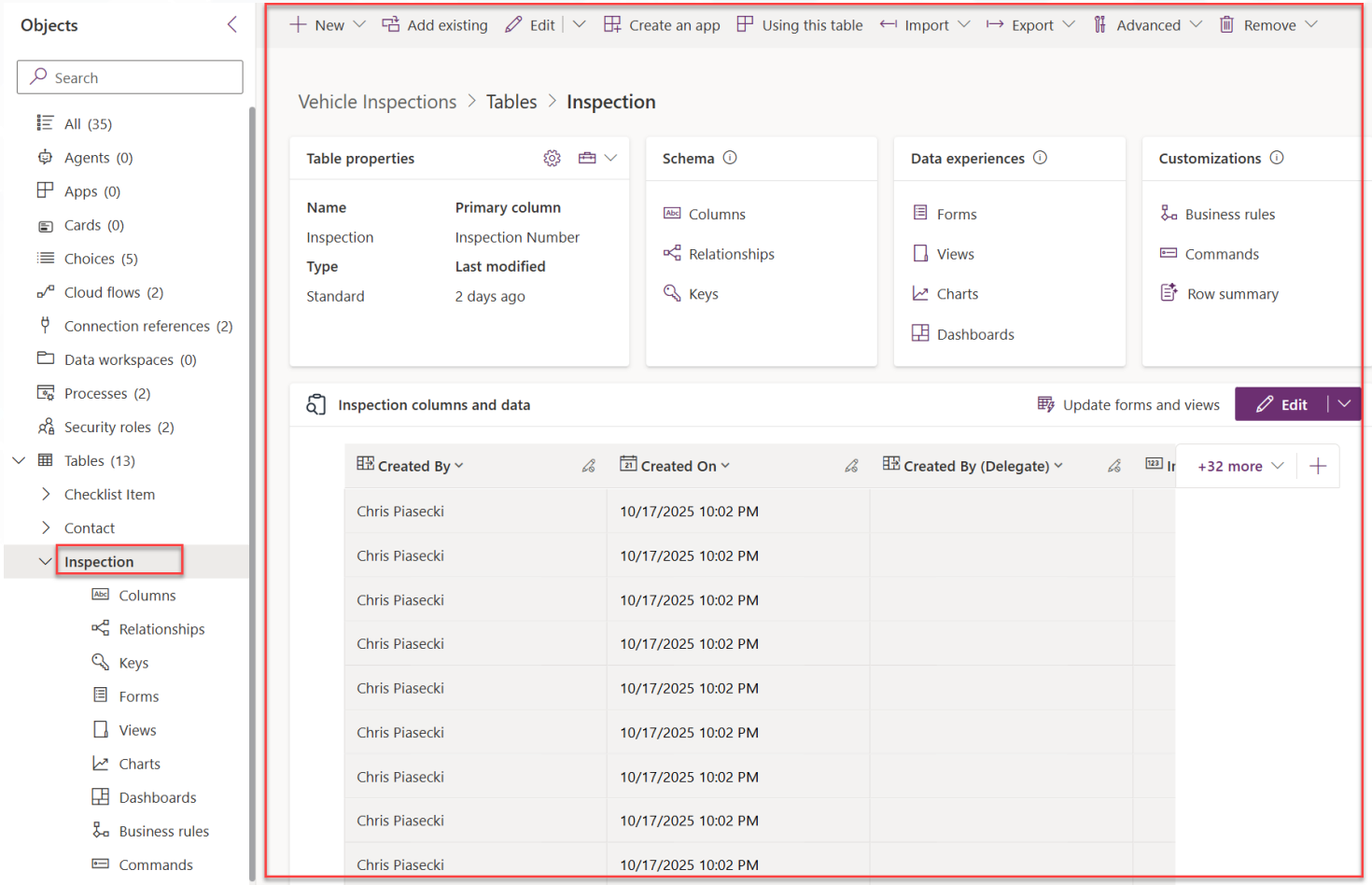
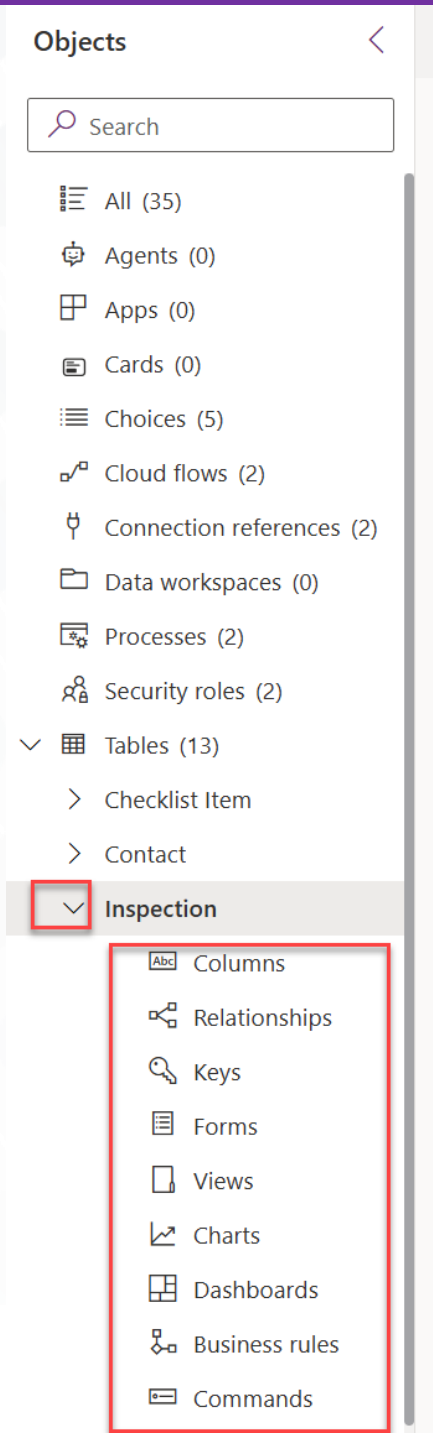


Table Components

Several sub-component types are available for a table.

- **Table** – Definition and its metadata/options
- **Columns** – Attributes that hold data
- **Relationships** – Relate two tables together
- **Keys** – Define alternate key columns
- **Forms** – working with a single row
- **Views** – List of rows with filtering/sorting/etc.
- **Charts** – Visualization on top of a view
- **Dashboards** – Collection of charts/views
- **Business Rules** – Declarative business logic
- **Commands** – Buttons that execute logic



Columns & Data Types

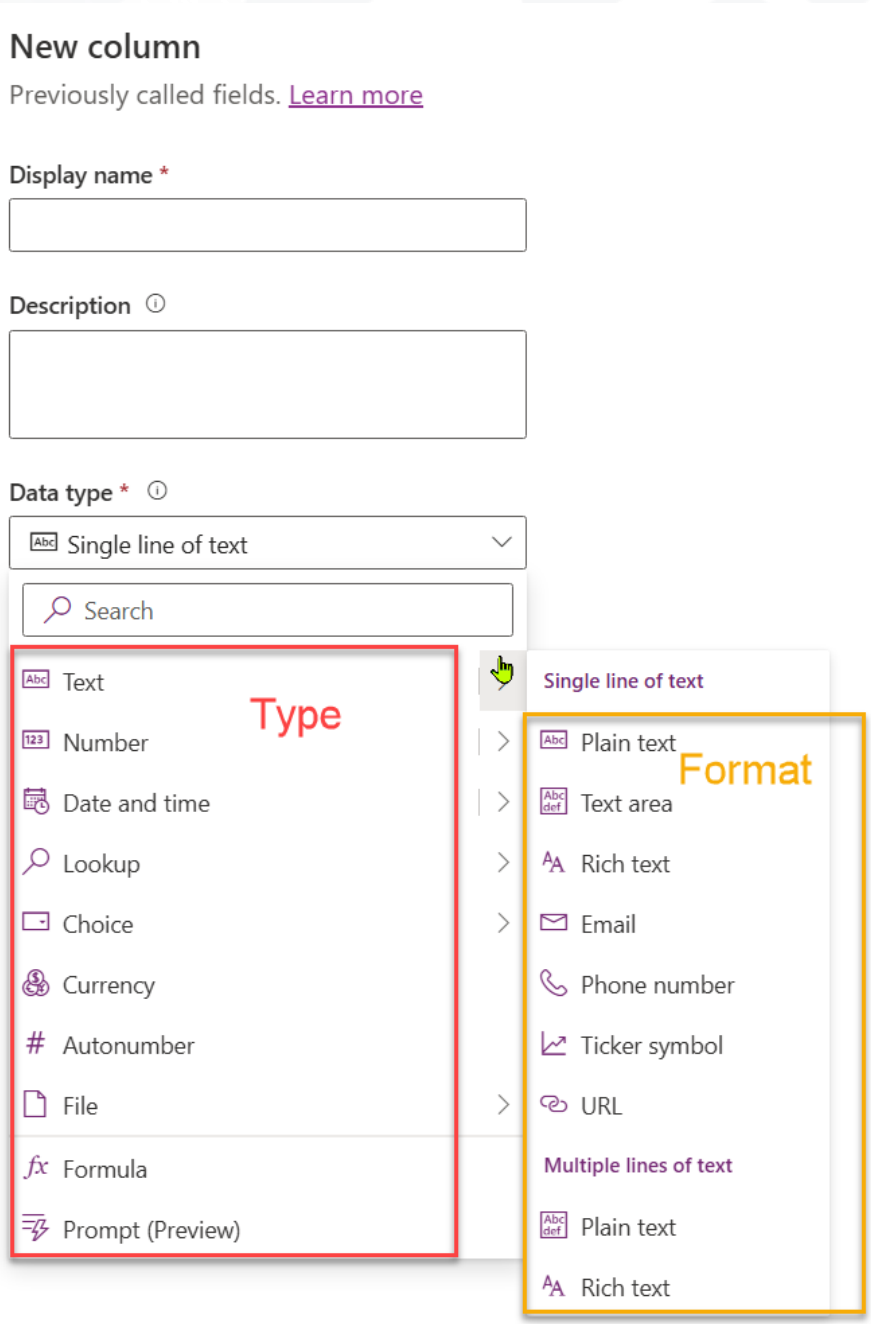
There are several basic column data types.

- Text
- Number
- Date and time
- Choice
- Lookup

Some types have specialized formatting options.

- Phone Number
- Rich Text
- Decimal
- Date Only

25+ data type and format combinations



System Columns

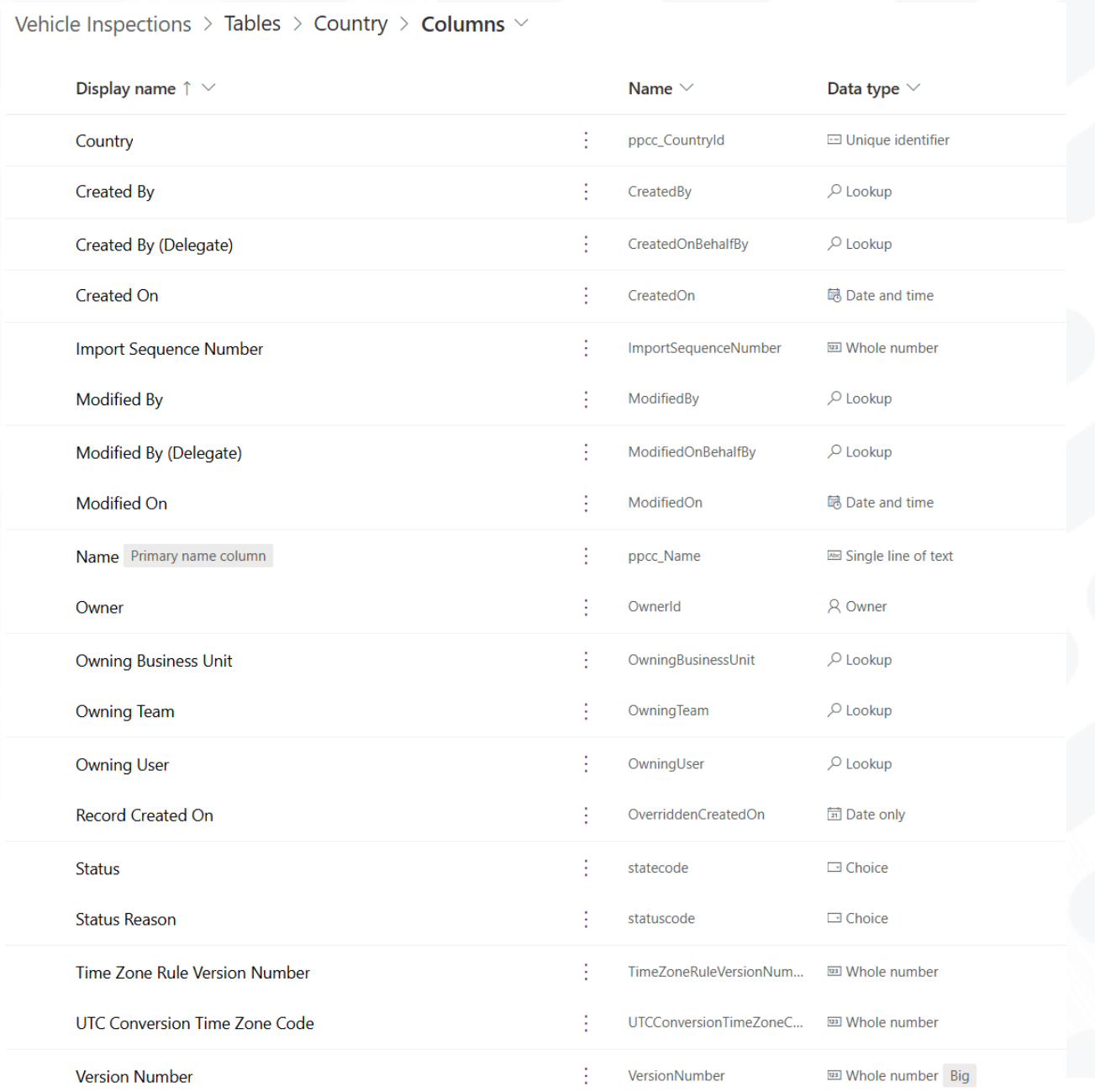
Unique Identifier – a globally unique identifier (GUID) for the row

Primary Name – a text (or autonumber) column which shows as the record title

Owner – a lookup that references the user or team that is assigned to the record

Status – Active and Inactive states, the latter making the record read-only. Not configurable.

Status Reason – Configurable statuses for active and inactive states



Display name	Name	Data type
Country	ppcc_Countryld	Unique identifier
Created By	CreatedBy	Lookup
Created By (Delegate)	CreatedOnBehalfBy	Lookup
Created On	CreatedOn	Date and time
Import Sequence Number	ImportSequenceNumber	Whole number
Modified By	ModifiedBy	Lookup
Modified By (Delegate)	ModifiedOnBehalfBy	Lookup
Modified On	ModifiedOn	Date and time
Name Primary name column	ppcc_Name	Single line of text
Owner	Ownerld	Owner
Owning Business Unit	OwningBusinessUnit	Lookup
Owning Team	OwningTeam	Lookup
Owning User	OwningUser	Lookup
Record Created On	OverriddenCreatedOn	Date only
Status	statecode	Choice
Status Reason	statuscode	Choice
Time Zone Rule Version Number	TimeZoneRuleVersionNum...	Whole number
UTC Conversion Time Zone Code	UTCConversionTimeZoneC...	Whole number
Version Number	VersionNumber	Whole number Big

Rollup Column

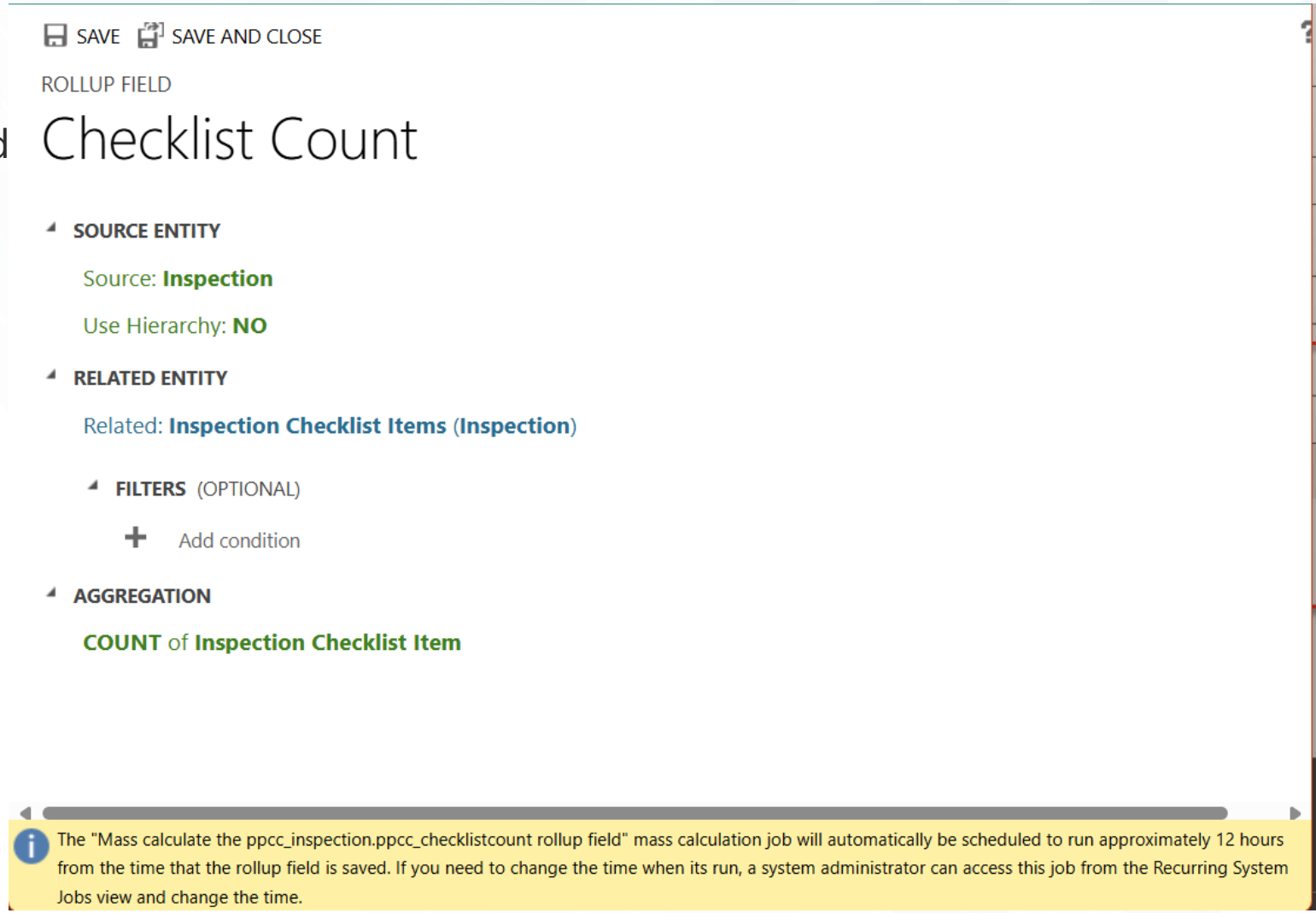
Rollup columns (fields) perform a system calculated aggregation against a set of child records.

Examples:

- Count the number of child Inspection Checklist Items for an Inspection
- Calculate the total amount of Invoice Line items

Value is read-only and is recalculated by a system job that can be configured to run as often as every hour.

Can force recalculation on a specific row in the UI.



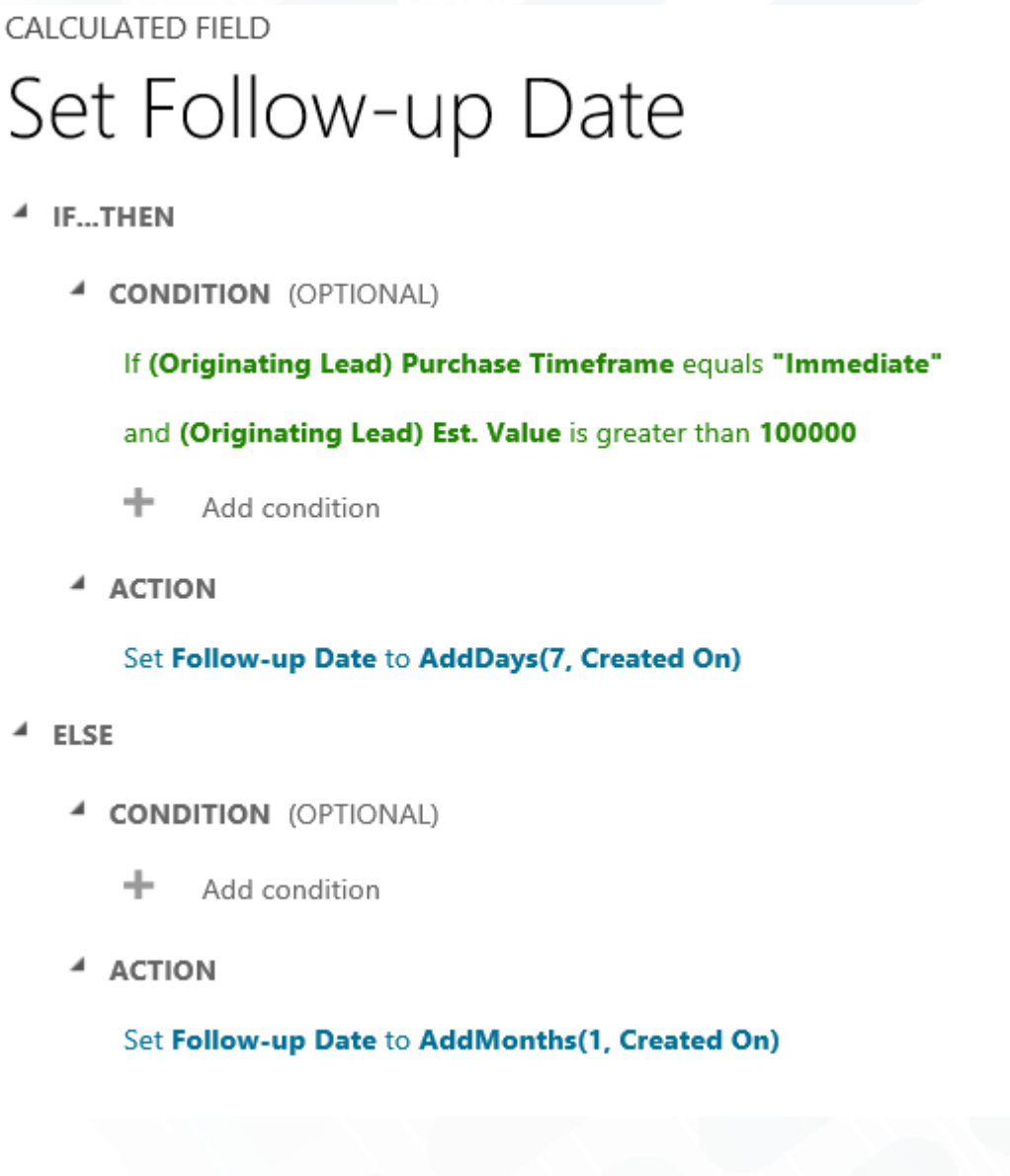
Calculated Column

Calculated columns perform a real-time calculation using field values on the table or parent records

The value is not persisted, instead it is computed when retrieving the row

Limited set of built in functions:

- Date/time: *ADDHOURS, ADDMINUTES, SUBTRACTDAYS,*
- Whole Number*: DIFFINMINUTES, DIFFINDAYS*.
- Text functions: *CONCAT, TRIMLEFT, TRIMRIGHT*



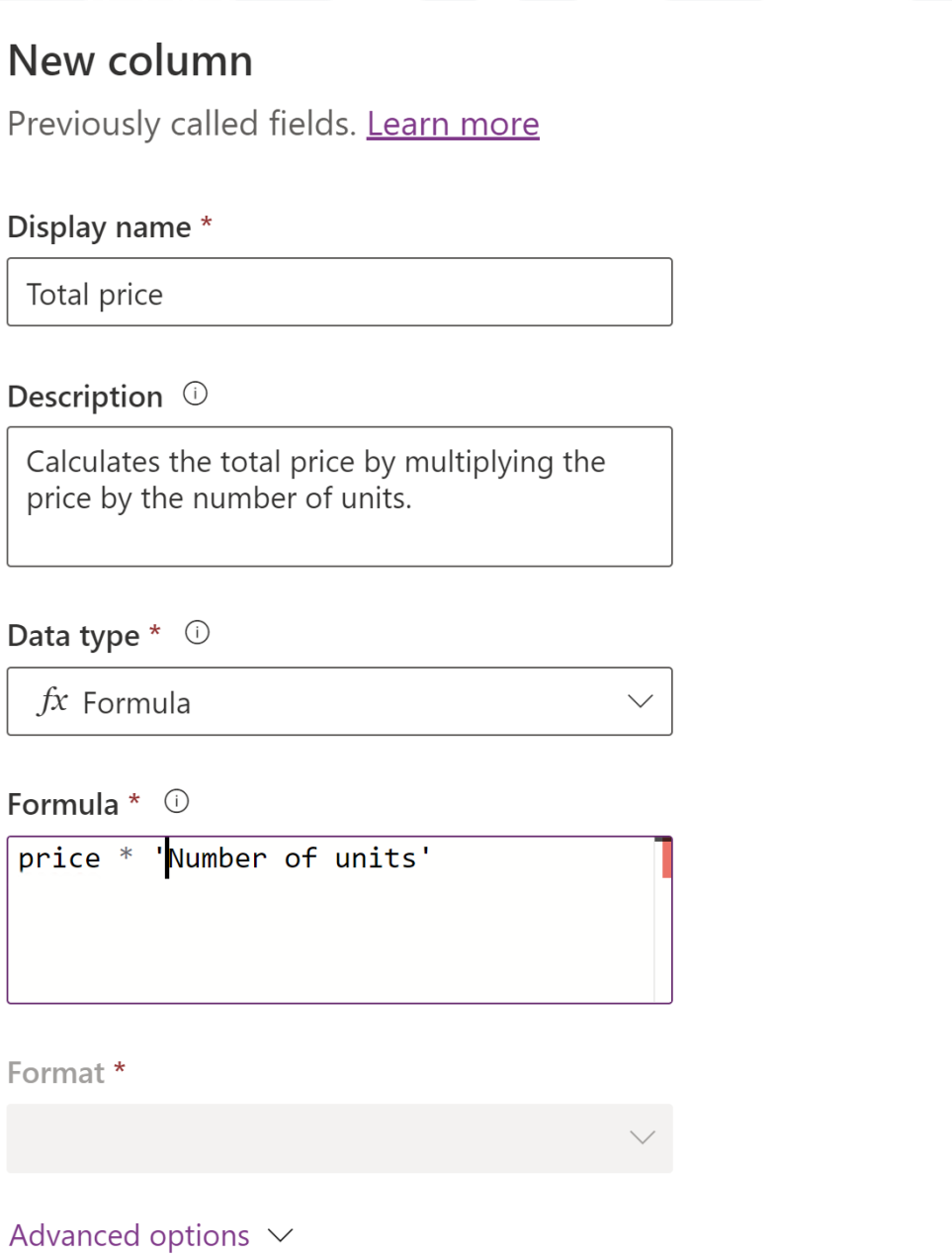
Fx Formula Column

Formula columns are a newer version of calculated column that are authored using a Power Fx expression.

The format (e.g. text, decimal, etc.) is inferred upon creation of column and can't be changed afterward.

A subset of Power Fx formulas are supported.

AI can provide formula suggestions based on a descriptive prompt.



Relationships

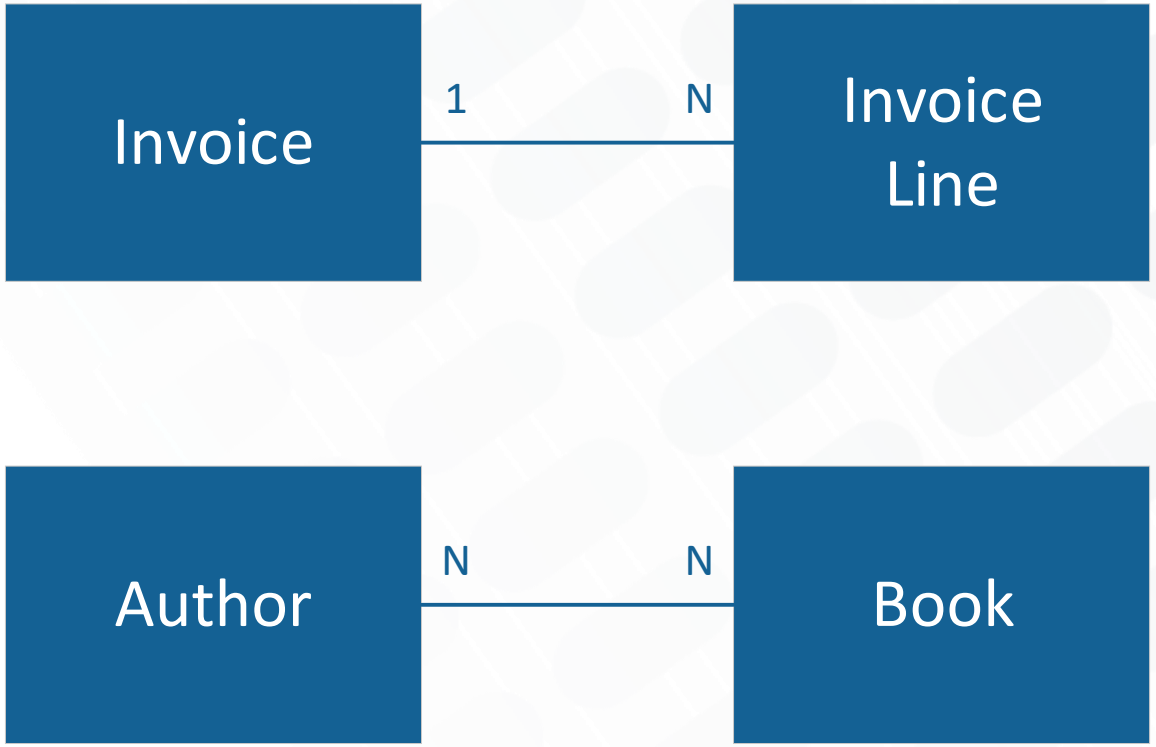
Relationships

Relationships are a formal way to relate two tables using a foreign key reference

Relationship types:

- One-to-Many (1:N)

- Many-to-Many (N:N)



Connections / Connection Roles

Connections create a link between two records without a formal relationship

- The records can be from any table that has the Connections option enabled.
- Useful for when there are many ways two records can be related

Connection Roles describe the relationship between the records

Examples:

- Grandparent <-> Grandchild
- Doctor <-> Patient
- Teacher <-> Student

Relationship Behavior

Helps ensure data integrity by defining the automatic behavior applied to child records when specific actions are performed against the parent record.

Action	Description
Assign	What should happen when the primary table row is assigned to someone else?
Reparent	What should happen when the lookup value of a related table in a parental relationship is changed?
Share	What should happen when the primary table row is shared?
Delete	What should happen when the primary table row is deleted?
Unshare	What should happen when a primary table row is unshared?
Merge	What should happen when a primary table row is merged?
Rollup View	What is the desired behavior of the rollup view associated with this relationship?

Relationship Behavior

Specific behaviors are available to perform against the child records

Behavior	Description
Cascade Active	Perform the action on all active related table rows.
Cascade All	Perform the action on all related table rows.
Cascade All	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.
Cascade User Owned	Perform the action on all related table rows owned by the same user as the primary table row.

Relationship Behavior

Each action has different behavior options

Action	Options
Assign	Cascade All, Cascade Active, Cascade User-owned, Cascade None
Reparent	Cascade All, Cascade Active, Cascade User-owned, Cascade None
Share	Cascade All, Cascade Active, Cascade User-owned, Cascade None
Delete	Cascade All, Remove Link, Restrict
Unshare	Cascade All, Cascade Active, Cascade User-owned, Cascade None
Merge	Cascade All, Cascade None
Rollup View	Cascade All, Cascade Active, Cascade User-owned, Cascade None

Alternate Keys

Keys

- Useful for uniquely identifying a row other than by the unique identifier (GUID)
- Can include a combination of multiple column values
- Commonly used for integration when storing the unique value from an external data store

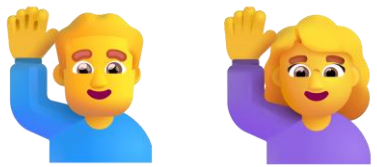
Examples:

- Vehicle Identification Number
- Social Security Number
- Vehicle Make, Model, Year

Demo

Data Modeling: Best Practices ⭐

- Use **Keys** to enforce uniqueness and improve read performance
- Create an **intersect table** instead of using a many-to-many relationship if you need additional columns
- **Do not over-normalize** as it can affect user experience, performance, and reporting
- Prefer **constrained** types like Choice/Lookup instead of text fields to control data quality and consistency.
- Use **Choice** instead of **Yes/No** unless you are certain there will only ever be a true/false state
- Prefer to use **Date** instead of **Yes/No** for better data insights/reporting
- Use **Date only** with a behavior of **Time zone Independent** or **Date only**
- Use a **Lookup** instead of a **Choice** when:
  - Options will change over time
  - Number of options are very large
  - Additional columns are needed to describe an option
  - Cascading filtering
  - Need to apply row level security for options



Q&A Time ⏰

Softball questions only. Unless you have a difficult problem.