outsystems

Modeling Data Relationships

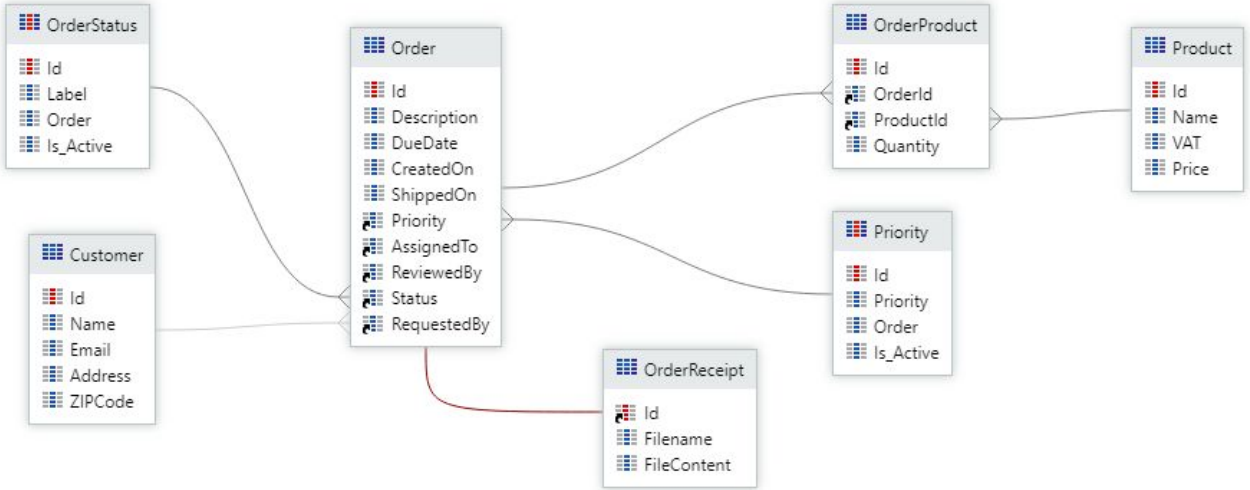
Topics

- Building Data Relationships
- Entity & Reference Identifiers
- Relationships
  - 1 to 1
  - 1 to Many
  - Many to Many
- Indexes
- Referential Integrity (Delete rule)
- Entity Diagrams

Building Data Relationships

Data types are rarely isolated

- A rich data model establishes the relationships between data
- The Relationships can be as important as the data itself

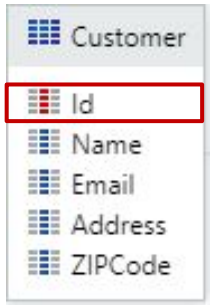


outsystems

Entity Identifier

An Entity must have an **Identifier** to allow relationships

- **Id** attribute is a long integer and automatically numbered by default
- Mandatory
- Possible types
  - Text
  - Integer / Long Integer
  - (Another) Entity Identifier

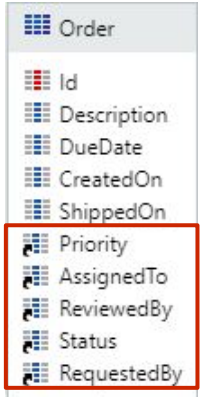


Represents the database table's Primary Key

- OutSystems support **simple primary keys**
- **NO** composite keys!

Referencing Data

- Entities can be referenced by their identifier
  - Create an attribute of type *Entity* Identifier
  - Can be mandatory or not
  - Static Entities can only reference other Static Entities
- Represents the database table's Foreign Key
- *NullIdentifier()* is the default value for reference attributes



Priority
Entity Attribute

Name	Priority
Description	...
Label	Priority
Data Type	Priority Identifier
Is Mandatory	No
Delete Rule	Protect

outsystems

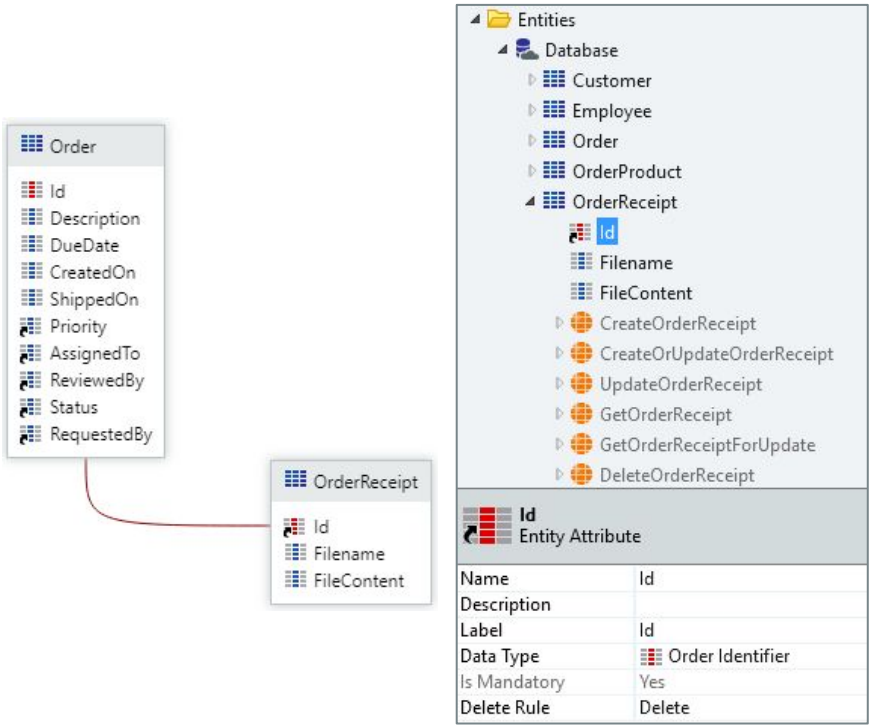
1-to-1 Relationship

1-to-1

- Each Receipt belongs to **one** Order
- Each Order has at **most one** Receipt

OrderReceipt is an **extension** of Order

- OrderReceipt has an identifier of type ***Order Identifier***
- Is Mandatory is set to *Yes*
- OrderReceipt Identifier <u>must</u> be explicitly assigned with the *Order* identifier when creating the record

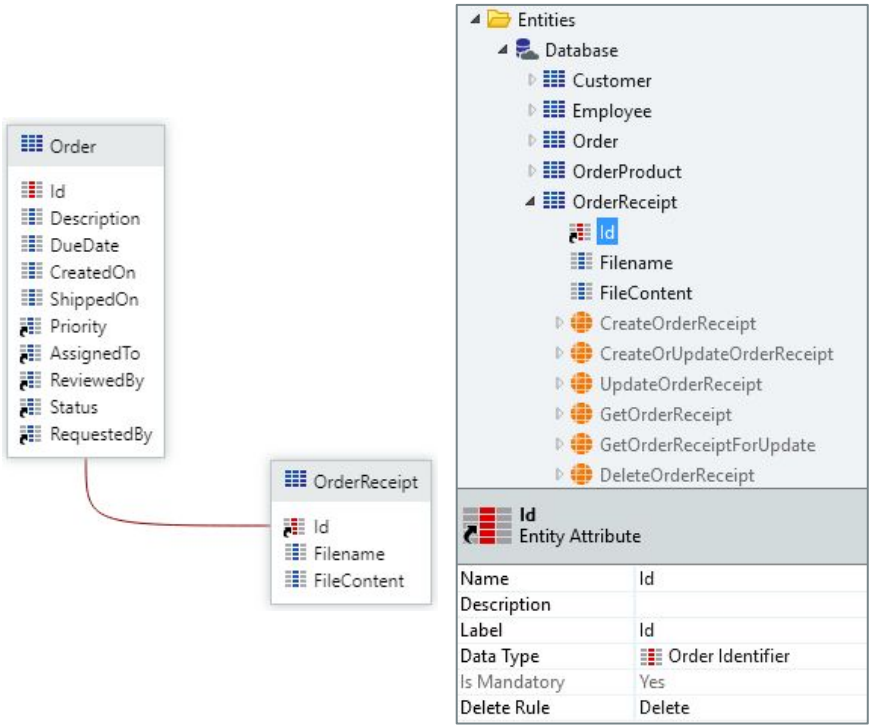


outsystems

Extension Entity

Extension Entity

- Shares its identifier with the base Entity
- The two Entities could be merged
- Usually split apart due to:
  - Performance issues
  - Base Entity is read only (e.g. User)



outsystems

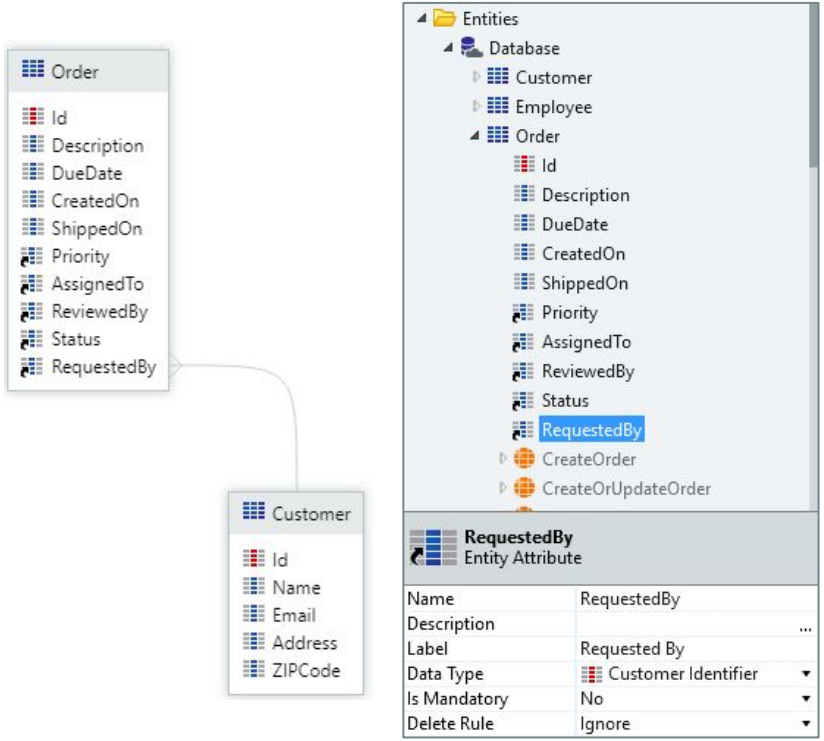
1-to-many Relationship

1-to-many

- Each Customer may have **many** Orders
- An Order was requested by **one** Customer

A Customer submits multiple Orders

- Order Entity has a reference attribute of type *Customer Identifier (RequestedBy)*
- Reference attribute may or may not be mandatory
  - If not mandatory, some Orders may not know which Customer requested the Order

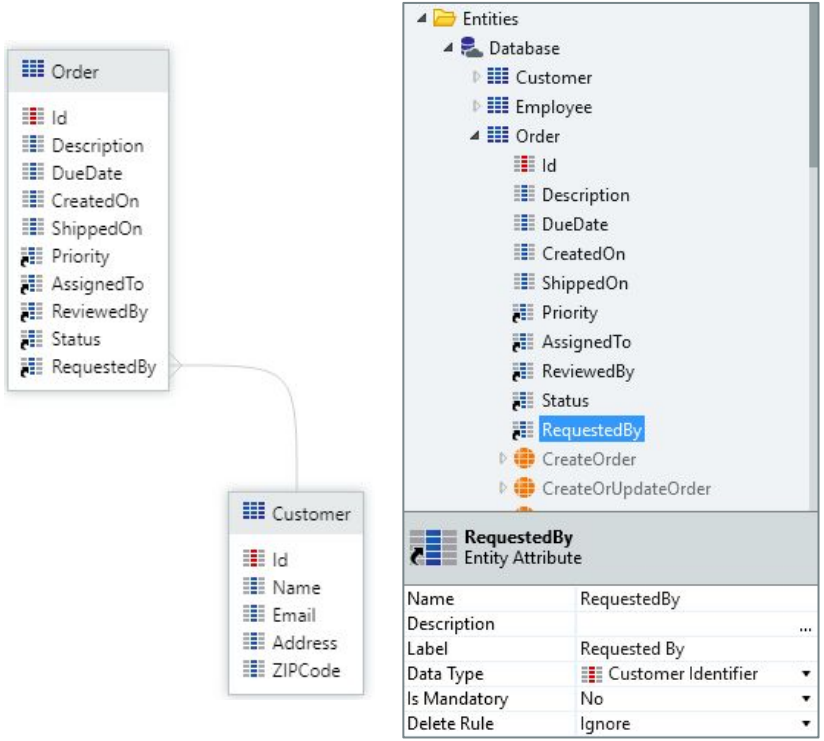


outsystems

Master-Detail Entities

Master-Detail

- Detail Entity references the Master Entity
- Detail Entity has a reference attribute of type *MasterEntity Identifier*
- An Entity can have multiple reference attributes



outsystems

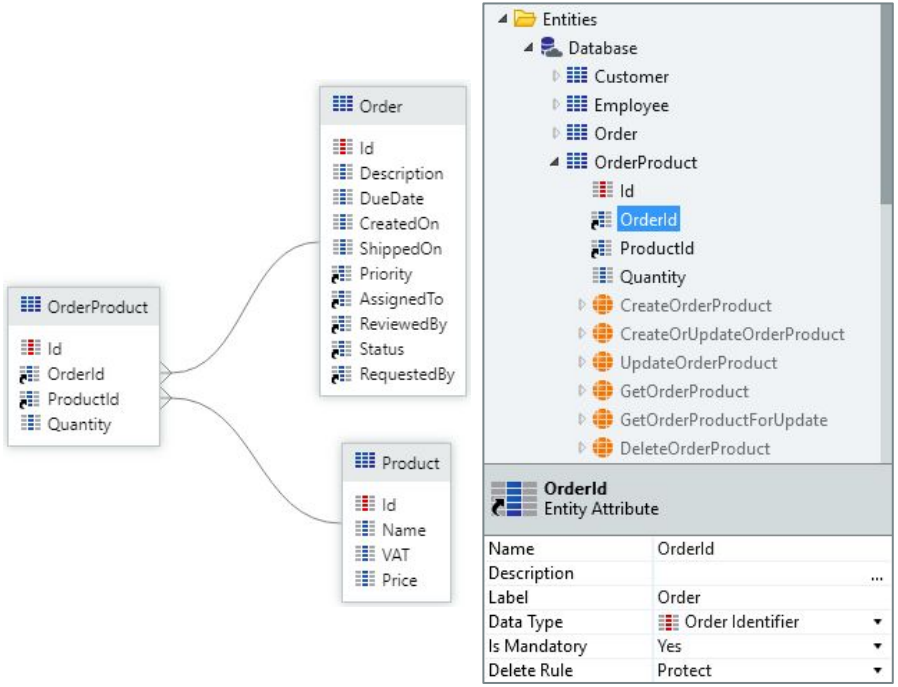
Many-to-many Relationship

Many-to-many

- Each Product may be part of **many** Orders
- Each Order may have **many** Products

An OrderProduct **Junction Entity** is required

- Reference attribute *OrderId* of type *Order Identifier*
- Reference attribute *ProductId* of type *Product Identifier*

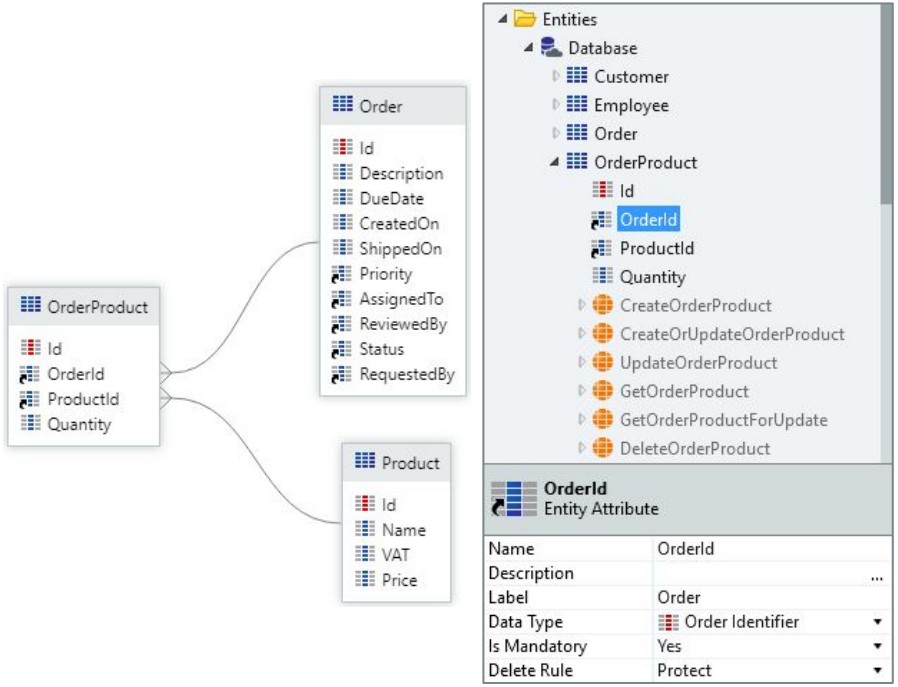


outsystems

Junction Entity

Junction Entity

- New Entity with its own unique **Id**
- One reference attribute per each Entity of the relationship
  - *MasterEntity1 Identifier*
  - *MasterEntity2 Identifier*
- Unique Index with both reference attributes may be useful



outsystems

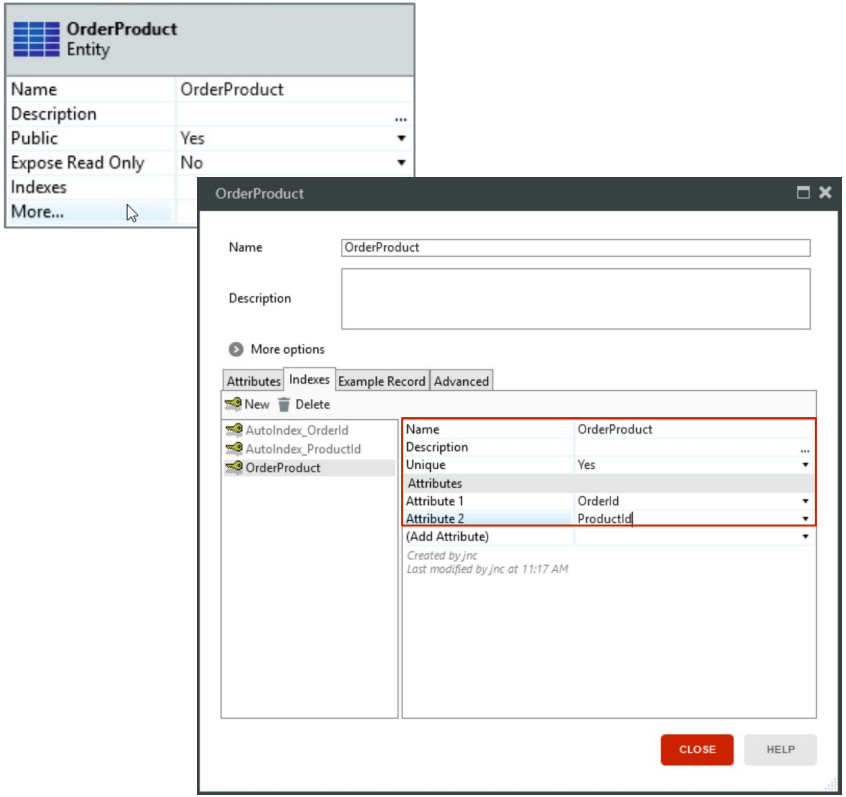
Indexes

Speed up data retrieval over certain attributes

- Costs additional writes and storage space
- Reference attributes have automatically created indexes
- Custom indexes can be created

Avoid duplicates

- Define the Index as **Unique**
- Combine one or more attributes



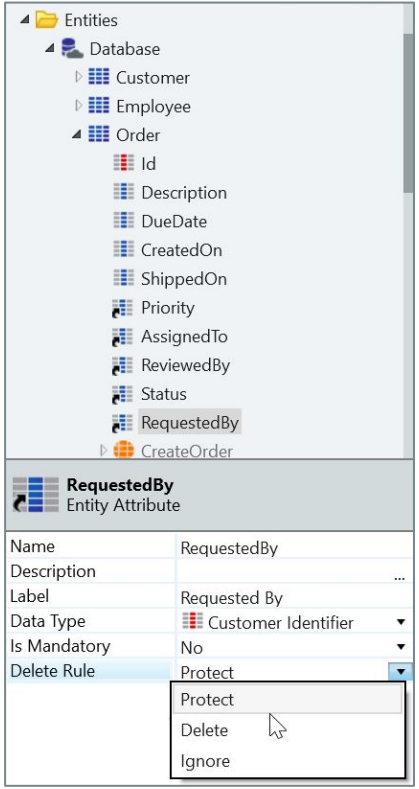
outsystems

Referential Integrity

Delete Rule property of the reference attribute

Applied when deleting a record from the referenced Entity (e.g. Customer)

- **Protect** - does not allow deleting the record
  - Ex: Customer with Orders is not deleted
- **Delete** - deletes the record and cascades delete all the records that reference it
  - Ex: Customer and all its Orders are deleted
- **Ignore** - does not guarantee referential integrity
  - Ex: Customer is deleted. Orders are kept



outsystems

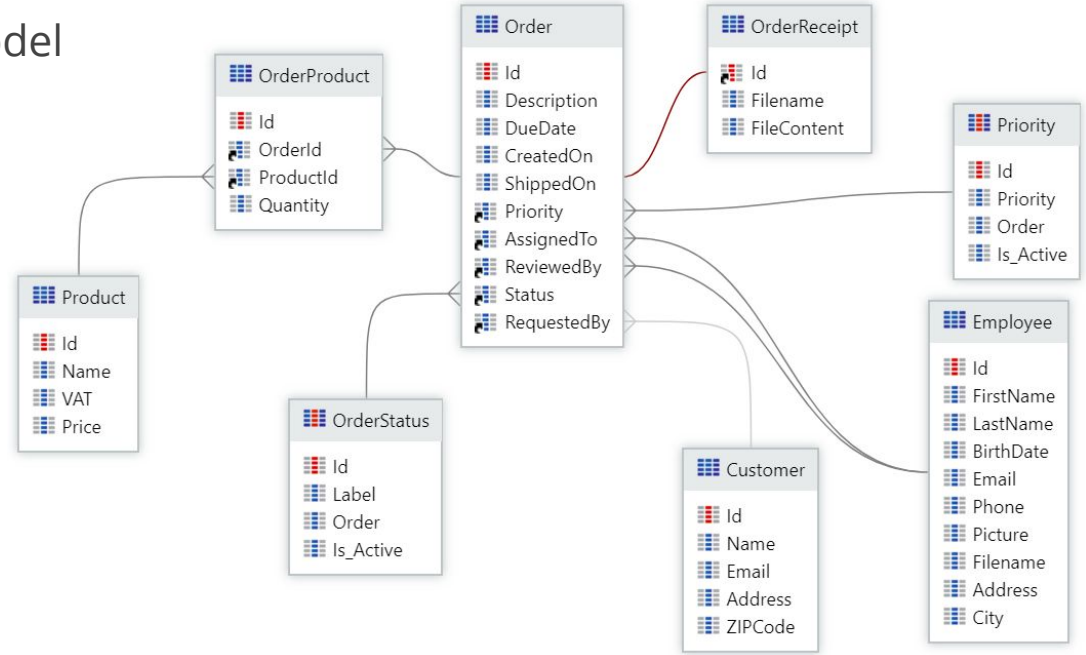
Entity Diagram

Visual representation of the Data Model

Designed by the developer

The Entity Diagram represents:

- Entities
- Relationships
- Delete Rules
  - Protect
  - Delete
  - Ignore



outsystems

Summary

- Building Data Relationships
- Entity & Reference Identifiers
- Relationships
  - 1 to 1
  - 1 to Many
  - Many to Many
- Indexes
- Referential Integrity (Delete rule)
- Entity Diagrams


Modeling Data Relationships
Thank You!