outsystems

OutSystems Database Entities

Topics

- Data Modeling in OutSystems
- Entities
  - Entity Attributes
  - Basic Data Types
  - Entity Actions
- Static Entities
  - Static Entity Records
  - Using Static Entities

Data Modeling in OutSystems

Many applications work with data that needs to be persisted in the Database

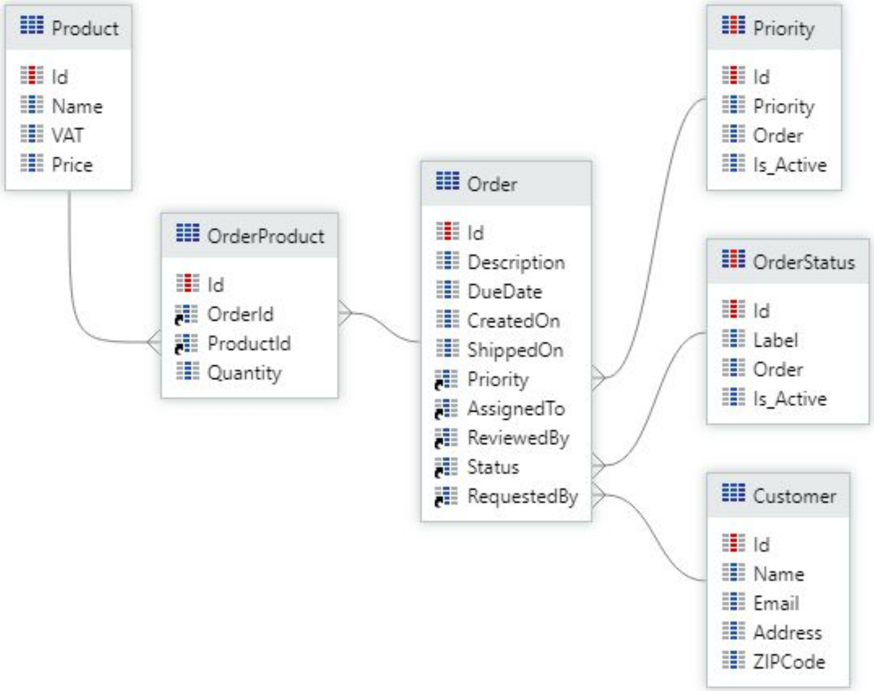
- Compound data with multiple fields

OutSystems enables persisting data with:

- Entities
- Static Entities

Entities and Static Entities are created in Service Studio

- OutSystems manages the creation of the underlying Database tables



Product
- Id
- Name
- VAT
- Price

OrderProduct
- Id
- OrderId
- ProductId
- Quantity

Order
- Id
- Description
- DueDate
- CreatedOn
- ShippedOn
- Priority
- AssignedTo
- ReviewedBy
- Status
- RequestedBy

Priority
- Id
- Priority
- Order
- Is_Active

OrderStatus
- Id
- Label
- Order
- Is_Active

Customer
- Id
- Name
- Email
- Address
- ZIPCode

Entities

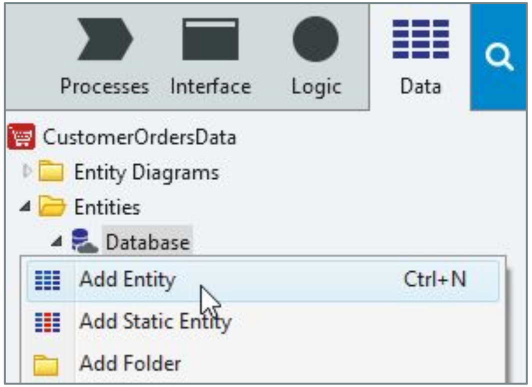
An Entity is persisted in a **Database Table**

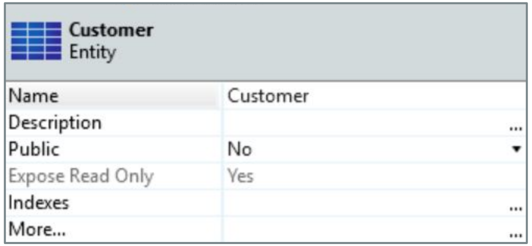
- Each new instance or record of an Entity is **inserted as a row** on the corresponding table

A different Entity should be created for each application concept

- Customer, Order and Order Item

An Entity is defined by fields called **Attributes**





Customer
Entity

Name	Customer
Description	...
Public	No
Expose Read Only	Yes
Indexes	...
More...	...

Entity Attributes

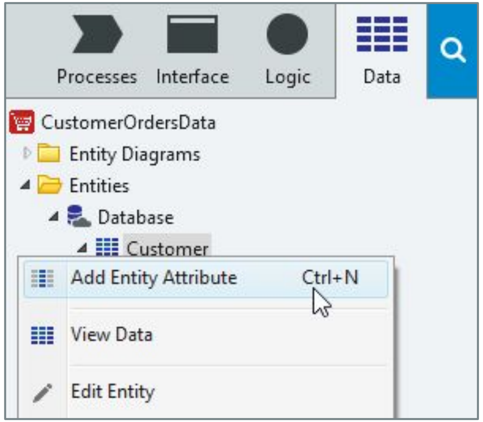
Entity data is stored in its **attributes**

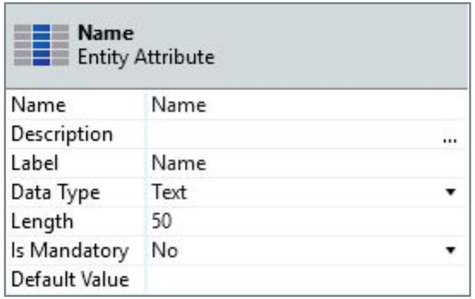
Each Attribute

- Is **a Column** on the respective database Table
- Must be one of the Basic Types

Every Entity is created with a special Id Attribute

- **Primary key** in the database
- Supports relational database operations




Name
Entity Attribute

Name	Name
Description	...
Label	Name
Data Type	Text ▾
Length	50
Is Mandatory	No ▾
Default Value	

Basic Data Types

Family	OutSystems Data Type	Example	Default Value
Alphanumeric	**Text**	“OutSystems”	“”
	Phone Number	“+1 555 111 222”	“”
	Email	“support@mail.com”	“”
Numeric	**Integer**	42	0
	Long Integer	4200000000	0
	Decimal	11.08	0.0
	Currency	9.99	0.0
Logic	**Boolean**	True	False
Dates and Times	**Date Time**	#2011-12-13 14:15:16#	#1900-01-01 00:00:00#
	Date	#2011-12-13#	#1900-01-01#
	Time	#14:15:16#	#00:00:00#
Large Object	**Binary Data**	<Unprintable>	<0 bytes>
Referential	**Entity Identifier**	<Positive Integer or Text>	NullIdentifier()

Data Type Inference

When naming a new attribute (or variable), OutSystems tries to infer its type

- Integer ← x, y, z, *Count, *Number
- Date ← *Date
  e.g. *BirthDate*
- DateTime ← *DateTime, *On, *Instant
  e.g. Datetime, CreatedOn
- Time ← *Time
- Boolean ← Is*, Has*
  e.g. IsCompleted, HasDocument
- Email ← *Email

- PhoneNumber ← *Phone, *Mobile
  e.g. HomePhone, WorkMobile
- Currency ← *Amount, *Price
- User Identifier ← *By
  e.g. CreatedBy
- Entity Identifier ← <*Entity*>Id
  e.g. CustomerId
- Entity Record ← <*Entity*>
- Entity Record List ← <*Entity*>s/ies

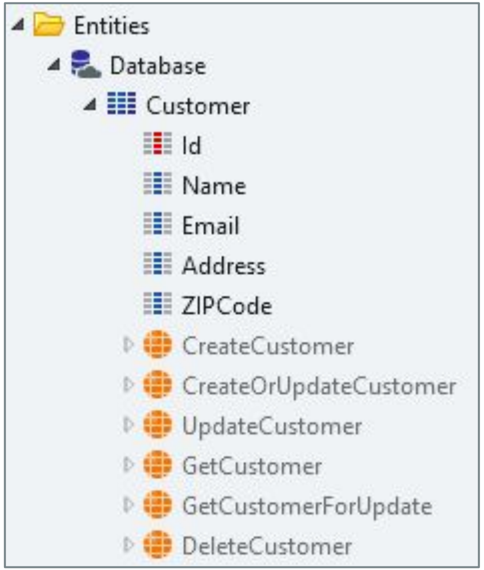
Entity Actions

OutSystems automatically creates **Entity Actions** for each of the CRUD data operations

Data operations refer to the Entity by its Name:

- **C**reate: CreateCustomer
- **R**etrieve: GetCustomer
- **U**pdate: UpdateCustomer
- **D**elete: DeleteCustomer

Entity Actions can be used directly in the application logic



Entity Actions in Detail

Entity Action	Parameters	Description
Create<*EntityName*>	Source: Entity record Id: Entity Id	Inserts a new record in the database, returning the Id of the newly created row
CreateOrUpdate<*EntityName*>	Source: Entity record Id: Entity Id	Inserts in the database (or updates if it exists) the record, returning the appropriate row Id
Update<*EntityName*>	Source: Entity record	Updates the record in the database. If it doesn't exist, an error occurs
Get<*EntityName*>	Id: Entity Id Record: Entity record	Retrieves from the database the record with the provided Id. If it doesn't exist, an error occurs
GetForUpdate<*EntityName*>	Id: Entity Id Record: Entity record	Similar to the Get entity action. Database row is locked until the database transaction finishes
Delete<*EntityName*>	Id: Entity Id in database	Deletes from the database the record with the provided Id. If it doesn't exist, an error occurs

Static Entities

Static Entities are Entities that have a predefined set of possible values

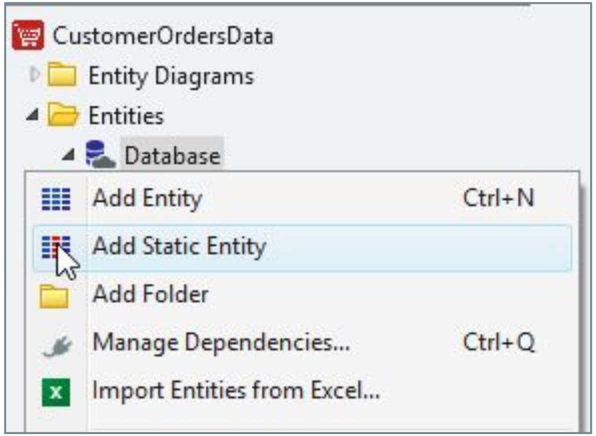
- Attribute values are defined at design time

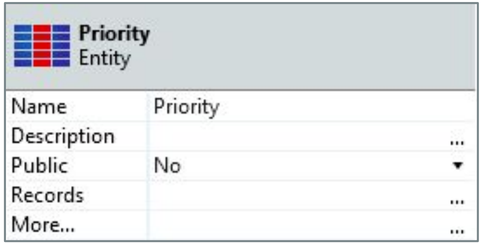
Each value of a Static Entity is a **Record**

Static Entity records **can NOT be created, updated, or deleted**; only retrieved

- **Get<StaticEntityName>** is the only Entity Action

Static Entities can be used like enumerations



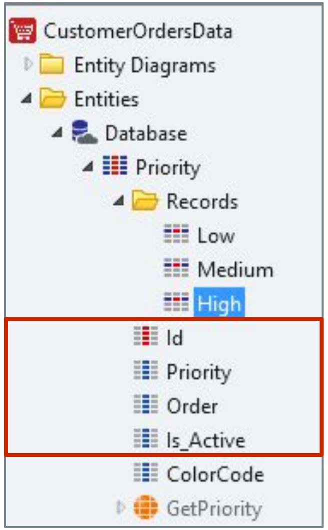


Priority
Entity

Name	Priority
Description	...
Public	No
Records	...
More...	...

Static Entity Records

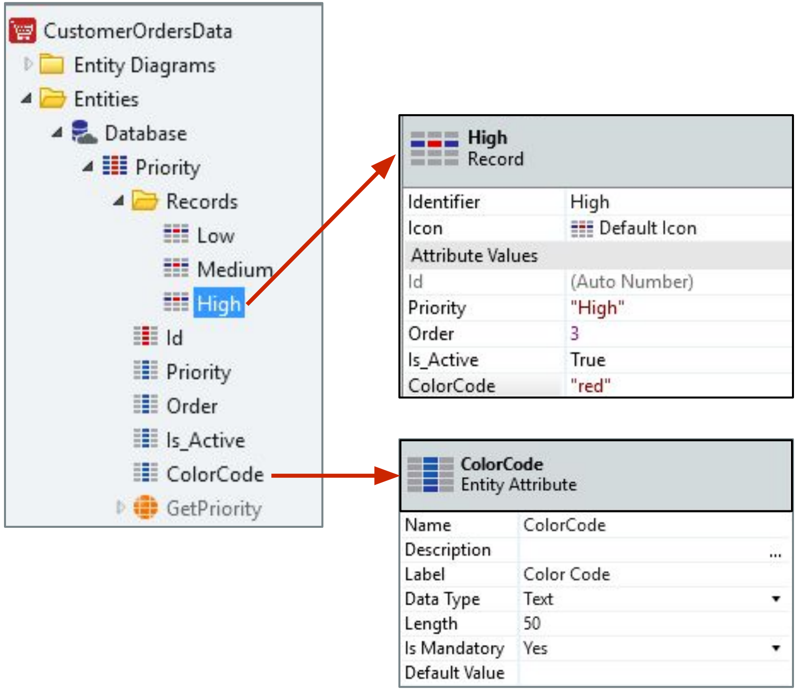
- Static Entities Records start with 4 automatically created attributes
  - Id
  - Label
  - Order
  - Is Active



CustomerOrdersData
Entity Diagrams
Entities
Database
Priority
Records
Low
Medium
High
Id
Priority
Order
Is_Active
ColorCode
GetPriority

Static Entity Records

- Static Entities are created with 4 automatically created attributes
  - Id
  - Label
  - Order
  - Is Active
- These attributes can be changed or deleted and others can be added
- Define each Record by setting the value of each Attribute
  - In Service Studio (development time)



High
Record

Identifier	High
Icon	Default Icon
Attribute Values	
Id	(Auto Number)
Priority	"High"
Order	3
Is_Active	True
ColorCode	"red"

ColorCode
Entity Attribute

Name	ColorCode
Description	...
Label	Color Code
Data Type	Text
Length	50
Is Mandatory	Yes
Default Value	

outsystems

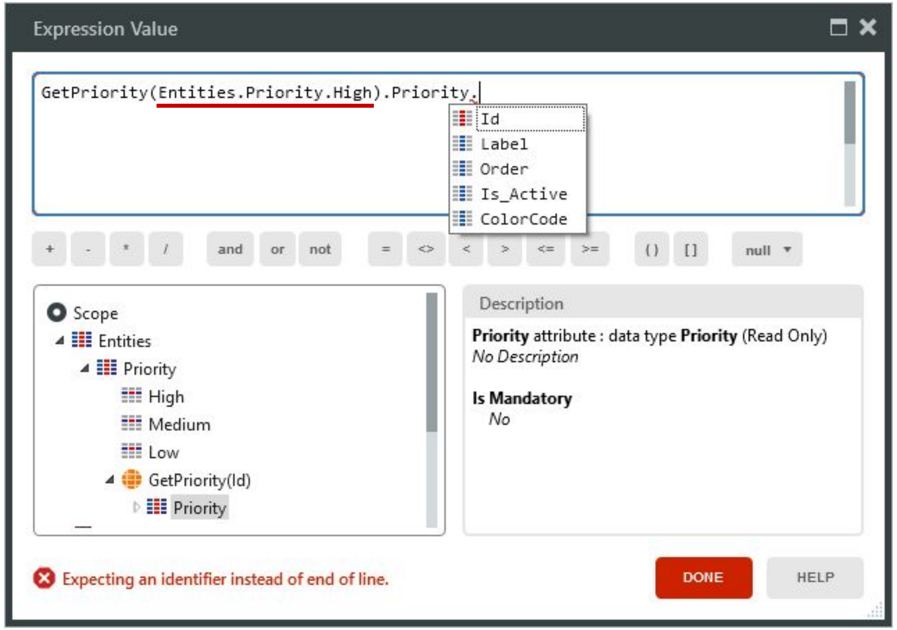
Using Static Entities

Static Entity Records can be **referenced by Record name**

- Known at design time

Record name can be used in place of the Static Entity's Identifier

- In an Input of an action or a function (e.g. GetPriority Entity Action)
- Directly in an Expression or Assign



```
GetPriority(Entities.Priority.High).Priority.
```

Priority attribute : data type **Priority** (Read Only)
No Description

Is Mandatory
No

Expecting an identifier instead of end of line.

Summary

- Data Modeling in OutSystems
- Entities
  - Entity Attributes
  - Basic Data Types
  - Entity Actions
- Static Entities
  - Static Entity Records
  - Using Static Entities

OutSystems

Database Entities

Thank You!