outsystems

Basic Data Queries

a.k.a. Aggregates

Topics

- Aggregates Overview
- Creating Aggregates
- Aggregates
  - Sources
  - Filters
  - Sorting
  - Testing
  - Output
  - Properties

Aggregates Overview

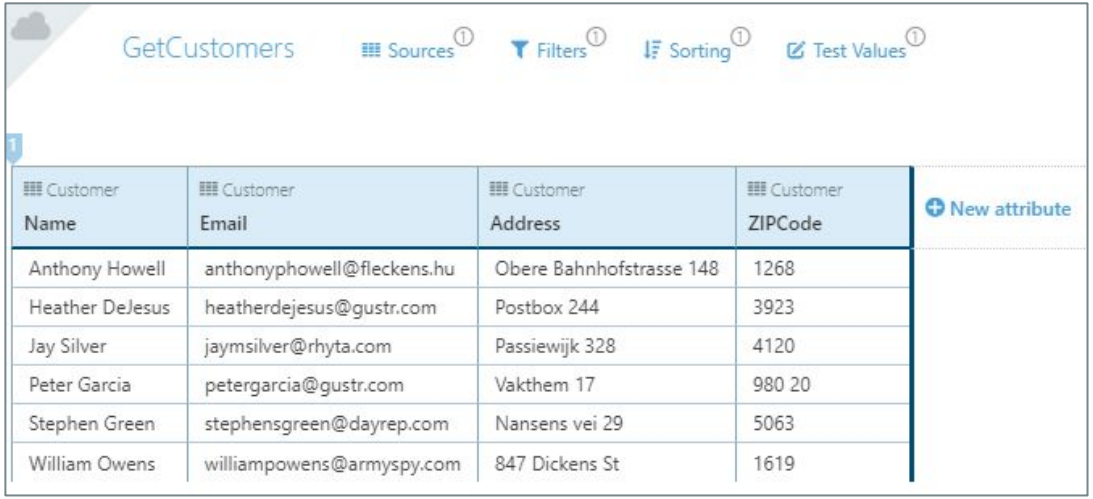
Most web applications need to fetch data from the database

Aggregates allow us to define database queries in a visual way

- Add Sources
- Create filters
- Define sorting

Aggregates are easy to create and maintain

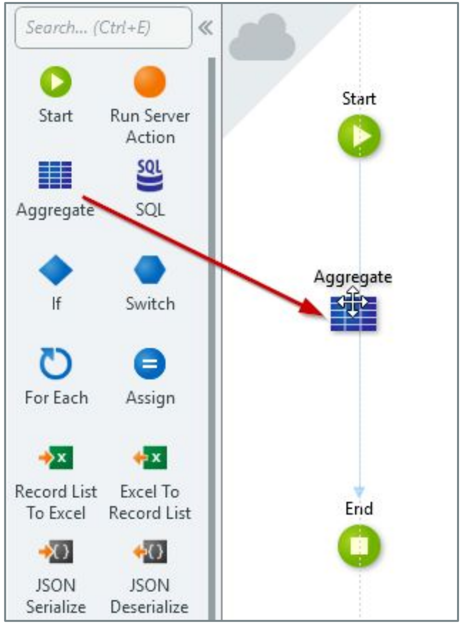
- Excel-like display of real data
- SQL knowledge is NOT required



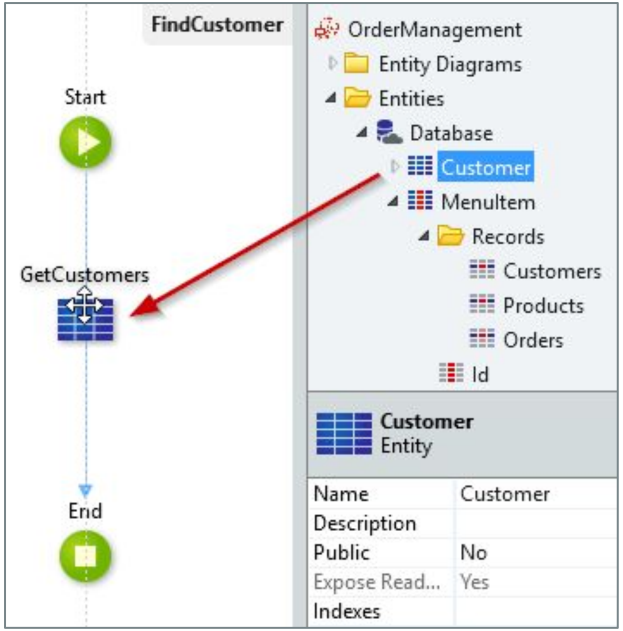
Customer Name	Customer Email	Customer Address	Customer ZIPCode
Anthony Howell	anthonyphowell@fleckens.hu	Obere Bahnhofstrasse 148	1268
Heather DeJesus	heatherdejesus@gustr.com	Postbox 244	3923
Jay Silver	jaymsilver@rhyta.com	Passiewijk 328	4120
Peter Garcia	petergarcia@gustr.com	Vakthem 17	980 20
Stephen Green	stephensgreen@dayrep.com	Nansens vei 29	5063
William Owens	williampowens@armyspy.com	847 Dickens St	1619

Creating an Aggregate

Drag an Aggregate from the Toolbox to an Action flow



Drag an Entity to an Action Flow (accelerator)



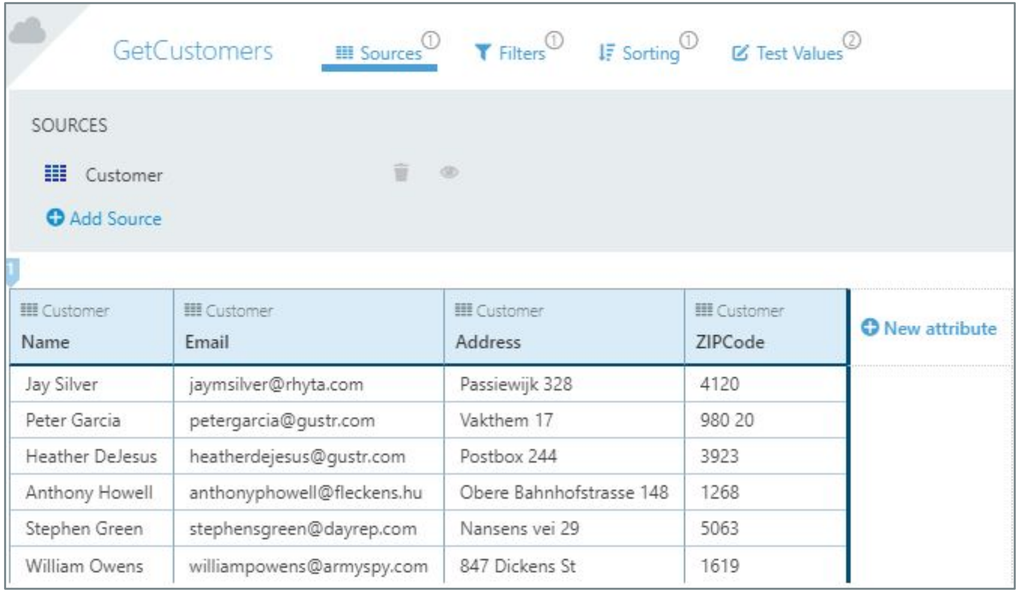
Name	Customer
Description	
Public	No
Expose Read...	Yes
Indexes	

Aggregate Sources

Aggregates support one or more source Entities

The sources determine the type of the Aggregate's Output List

- e.g. Customer List



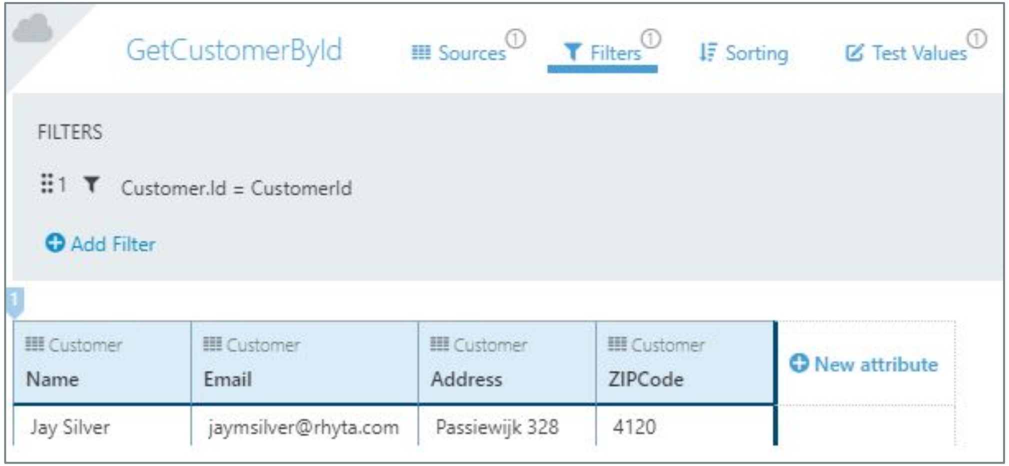
Customer	Customer	Customer	Customer
Name	Email	Address	ZIPCode
Jay Silver	jaymsilver@rhyta.com	Passiewijk 328	4120
Peter Garcia	petergarcia@gustr.com	Vakthem 17	980 20
Heather DeJesus	heatherdejesus@gustr.com	Postbox 244	3923
Anthony Howell	anthonyphowell@fleckens.hu	Obere Bahnhofstrasse 148	1268
Stephen Green	stephensgreen@dayrep.com	Nansens vei 29	5063
William Owens	williampowens@armyspy.com	847 Dickens St	1619

Aggregate Filters

Adds one or more conditions to the query to filter the output records

- Support for multiple filters
- Supports logical operators
  - =, <>, and, or, ...
- Support for some built-in functions
  - CurrDateTime()
  - If(Condition, True, False)

Equivalent to a SQL **WHERE** Clause



FILTERS

1 Customer.Id = CustomerId

Customer Name	Customer Email	Customer Address	Customer ZIPCode
Jay Silver	jaymsilver@rhyta.com	Passiewijk 328	4120

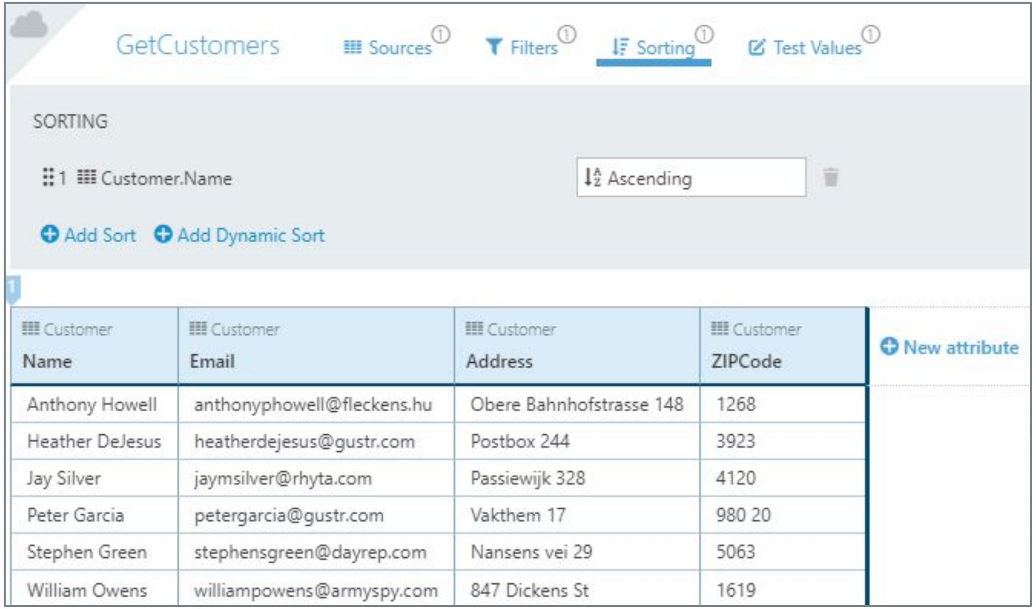
Aggregate Sorting

Defines the Entity's attribute to sort by and in which direction

- Ascending
- Descending

When defining multiple sorts the order is relevant to the result

Equivalent to a SQL **ORDER BY**



SORTING

1 Customer.Name — Ascending

Customer Name	Customer Email	Customer Address	Customer ZIPCode
Anthony Howell	anthonyphowell@fleckens.hu	Obere Bahnhofstrasse 148	1268
Heather DeJesus	heatherdejesus@gustr.com	Postbox 244	3923
Jay Silver	jaymsilver@rhyta.com	Passiewijk 328	4120
Peter Garcia	petergarcia@gustr.com	Vakthem 17	980 20
Stephen Green	stephensgreen@dayrep.com	Nansens vei 29	5063
William Owens	williampowens@armyspy.com	847 Dickens St	1619

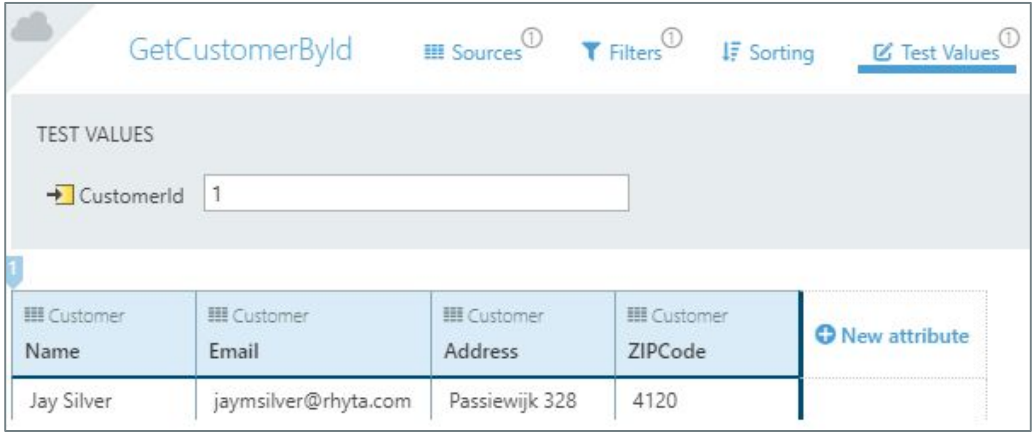
Aggregate Test Values

Allows specifying values for external variables used in the filter or sort conditions

- Has as many inputs as external variables

Used to test the Aggregate and preview the output records

- Does not have any influence on the actual output



TEST VALUES

CustomerId: 1

Customer Name	Customer Email	Customer Address	Customer ZIPCode
Jay Silver	jaymsilver@rhyta.com	Passiewijk 328	4120

Aggregates Output

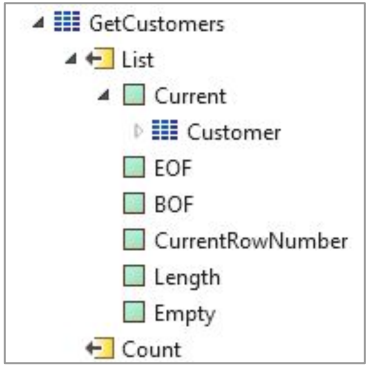
List of records with several properties

- Length - number of elements returned
- Empty - True if no records were returned
- EOF, BOF, CurrentRowNumber - filled while iterating the List

Aggregate Get<Entity> contains a “List.Current”

- Type matches the definition in the query
- The “Current” cursor moves through the List when iterating
  - Points at first row by default
- Filled in with default values if no rows returned by query
  - Empty List

Count has the total number of records that match the criterias defined in the Aggregate



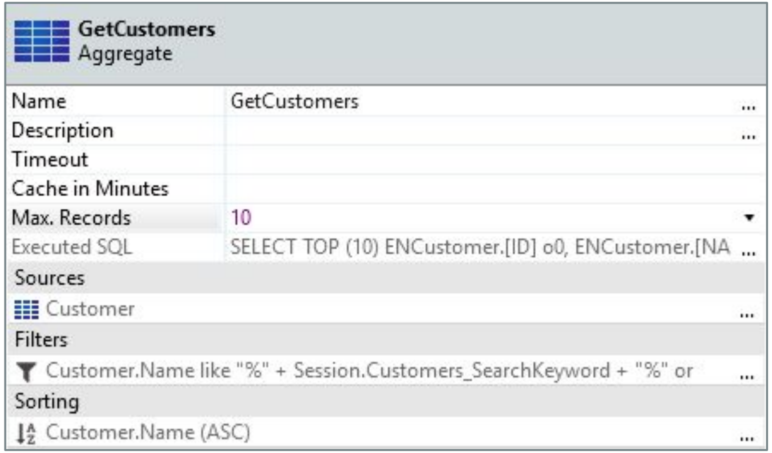
Aggregate Properties

It is possible to limit the Aggregate’s output to a maximum number of records

- **Max. Records** property
- Does not impact the Count output

Aggregate Editor is SQL-dialect agnostic

- SQL is generated according to the DBMS used
- **Executed SQL** property shows the SQL statement generated from the Aggregate
- SQL SELECT statements are optimized by the platform according to the output usage



GetCustomers
Aggregate

Name	GetCustomers
Description	
Timeout	
Cache in Minutes	
Max. Records	10
Executed SQL	SELECT TOP (10) ENCustomer.[ID] o0, ENCustomer.[NA
Sources	
Customer	
Filters	
Customer.Name like "%" + Session.Customers_SearchKeyword + "%" or	
Sorting	
Customer.Name (ASC)	

Summary

- Aggregates Overview
- Creating Aggregates
- Aggregates
    - Sources
    - Filters
    - Sorting
    - Testing
    - Output
    - Properties

Basic Data Queries

Thank You!