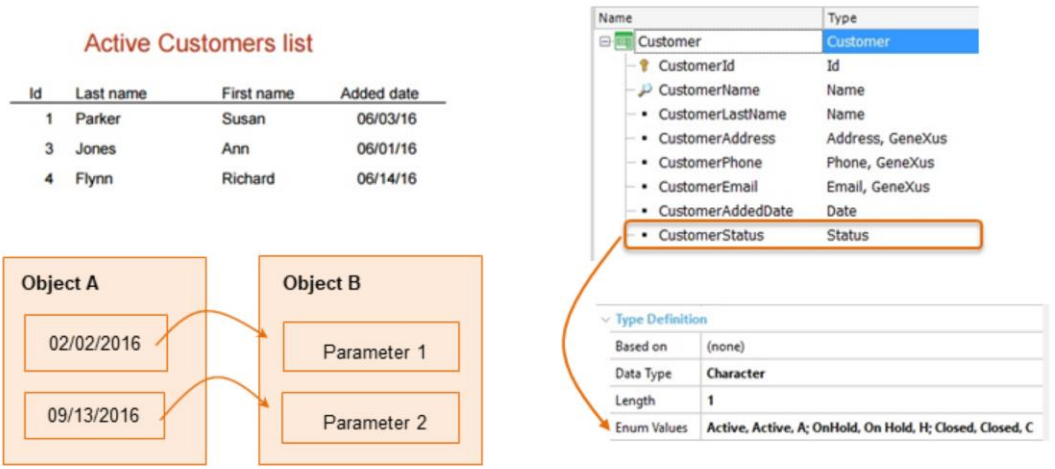
Data Selectors

Reusing definitions

GeneXus 16

Scenario

- Active customers entered between two given dates ... in many queries!



Suppose that we have added the CustomerStatus attribute to the Customer transaction, in order to represent one of the three statuses (active, on hold, closed) that a customer can have in the travel agency system. To this end, an enumerated **Status** data type has been created, as shown in the image.

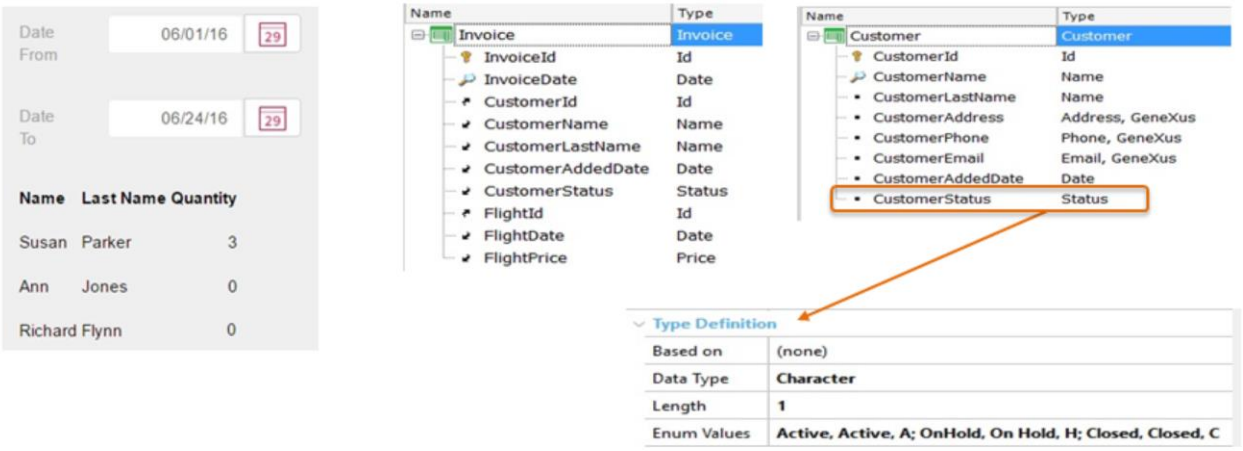
Suppose that in several places of the application we need to work with the active customers entered between two given dates. For example:

1. A PDF list that receives a date range (&start and &end) and shows the active customers that were entered into the system between these two given dates.

In this example, we will talk about something that will be dealt with in more detail later: an object can receive values from the caller object. These values are necessary for it to perform its action (as in our example, a date range) but to do so it has to be able to receive them. To enable an object to receive values (which we call parameters), they have to be stated (we will see how it's done).

Scenario

- Active customers entered between two given dates ... in many queries!

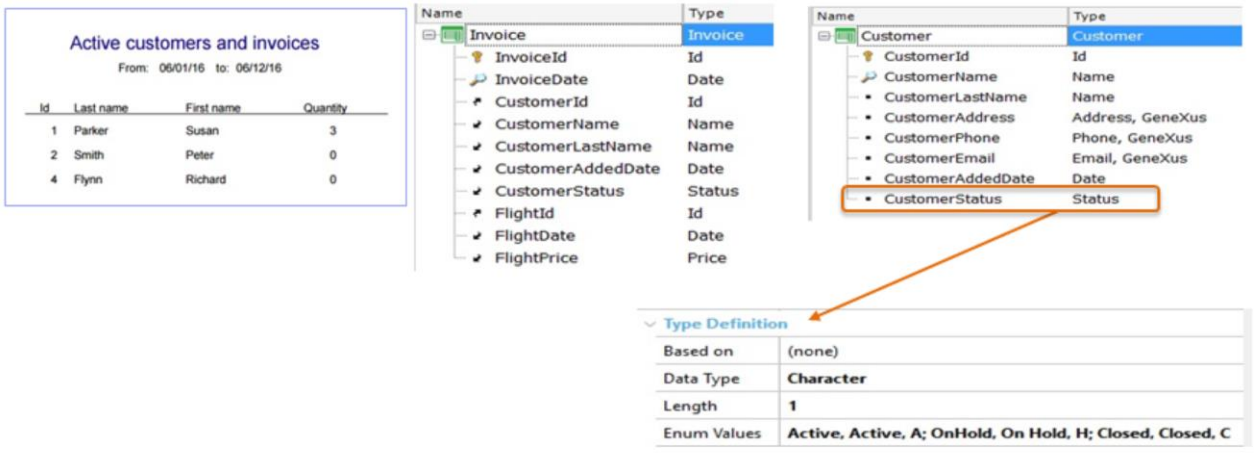


2 - In a web panel that shows all the customers who have invoices and their number of invoices, with the possibility for the user to enter a date range to count only the invoices corresponding to active customers entered into the system between those dates. If a customer is inactive or was entered outside those dates, he will be included in the list but his number of invoices will be zero.

It should be mentioned that a web panel is a very flexible type of GeneXus object that allows designing all kinds of interactive queries to the database. Later in this course we will look at this object in more detail.

Scenario

- Active customers entered between two given dates ... in many queries!



3. In a PDF list we need to show the same as in the previous web panel.

SOLUTION using what we know so far

- List of Active customers entered between two dates.

```
Print Title
For each Customer
    Where CustomerStatus = Status.Active
    Where CustomerAddedDate >= &DateFrom
    Where CustomerAddedDate <= &DateTo
    Print Customer
Endfor
```

```
Parm(in: &DateFrom, in: &DateTo);
```

Values received from another object (parameters)

- List of Active customers entered between two dates and their number of invoices.

```
Print Title
For each Invoice
    Unique CustomerId
    &Qty = Count(InvoiceDate,
          CustomerStatus=Status.Active and CustomerAddedDate>=&DateFrom and CustomerAddedDate<=&DateTo)
    Print Customer
Endfor
```

If we implemented the queries mentioned above using the knowledge we have so far, we would see that in both cases the three conditions indicated above are repeated.

Remember that writing three Where clauses is the same as writing only one whose conditions are joined by AND.

Note:
In general, an object states the parameters used to exchange information with the caller through a rule: the parm rule. In this case, in both lists we will create two variables: &DateFrom and &DateTo, to receive (that's why “in” is entered) the date range from the caller.

Definition

Data Selector

```
CustomerStatus = Status.Active
and
CustomerAddedDate >= &DateFrom
and
CustomerAddedDate <= &DateTo
```

- ✓ Savings and reuse
- ✓ Maintenance

Data Selector Structure | Documentation

Structure	Type	Description
ActiveCustomers		Active Customers
Parameters		
DateFrom	Date	Date From
DateTo	Date	Date To
Conditions		
CustomerStatus=Status.Active		
CustomerAddedDate>=&DateFrom		
CustomerAddedDate<=&DateTo		
Orders		
CustomerStatus		
DefinedBy		

To avoid having to repeat the same specifications everywhere we need them (the web panel and the previous procedures, as well as in objects of another type that we will see later), we can make these definitions in a single place, giving them a name, and from then on we can use that name as a reference. That place is the Data Selector object.

To optimize the query in a For Each command we would order by CustomerStatus, because we have an equality filter by that attribute.

As we can see, in the example we create a data selector called "ActiveCustomers", and there we state the conditions and order, and state the &DateFrom and &DateTo parameters, variables that are used in two of the conditions.

This centralized definition will allow us to reuse it everywhere this query is needed, making its maintenance easier (if we need to change something in the definition, it is made in one place and automatically applied everywhere the KB is used).

Let's see how, once defined the data selector, we would use in the examples mentioned.

Use

In Using clause of For Each command

```
Print Title
For each Customer
    Using ActiveCustomers(&DateFrom, &DateTo)
    Print Customer
Endfor
```

```
Parm(in: &DateFrom, in: &DateTo);
```

Active Customers list

Id	Last name	First name	Added date
1	Parker	Susan	06/03/16
3	Jones	Ann	06/01/16
4	Flynn	Richard	06/14/16

We use it through the **using** clause. Its behavior is the same as in the previous specification. Here we see the case of the first list.

For the For Each command, we could use the Data Selector by running it as a query that is independent from the database. We won't talk about it in this course, but it's worth noting that the **in** operator is used in this case. For example, if we added the customers' country, and we wanted to list the countries that have active customers who have been entered into the system between two given dates, we would enter this code:

For each Country
Where CountryId **in** ActiveCustomers(&DateFrom, &DateTo)
...
endfor

Here we have two queries to the database: one of the Data Selector, which will return the set of active customers who have been entered between the two dates indicated and their corresponding countries. The other, corresponding to the For Each command, will filter the countries included in that set.

You can find more documentation about Data selectors in For Each commands in our wiki
http://wiki.genexus.com/commwiki/servlet/wiki?5312,Data+Selectors+in+For+Each+command

In Formula

```
Print Title
For each Invoice
    Unique CustomerId
    &Qty = Count(InvoiceDate, using ActiveCustomers(&DateFrom, &DateTo))
    Print Customer
Endfor
```

Active customers and invoices

From: 06/01/16 to: 06/12/16

Id	Last name	First name	Quantity
1	Parker	Susan	3
2	Smith	Peter	0
4	Flynn	Richard	0

Here we can see the case of the second list, where we're using the data selector within the Count formula. Remember that the second parameter of an aggregate formula is for writing the conditions that must be met by the records in order to be “aggregated”.

We don’t show the web panel example here because we haven’t studied this object. When we have studied it, we will explain where to use the Data Selector to filter the data that will be shown in a grid.

For Each command syntax

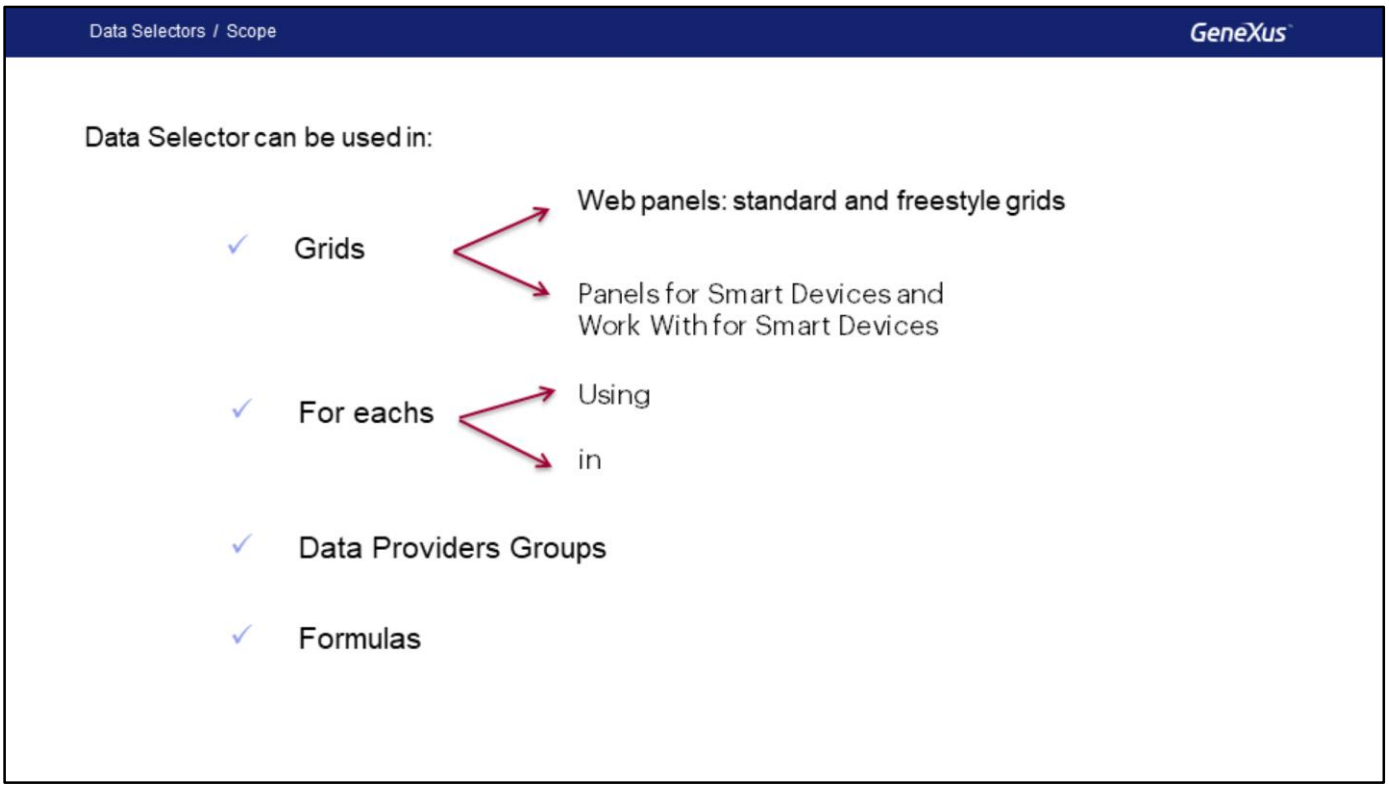
```
For each      BaseTransaction
    order att1, att2, ... , attn [when condition]
    order att1, att2, ... , attn [when condition]
    unique att1, att2, ... , attn
    using DataSelector( parm1, parm2, ... , parmn)
    where condition [when condition]
    where condition [when condition]
    where att IN DataSelector( parm1, parm2, ... , parmn)
        main code
When none
        ...
Endfor
```

A Data Selector specifies, based on the parameters received, a set of conditions and orders for the data in a centralized manner, so as to avoid having to repeat the Order, Where and Defined by clauses everywhere they are needed.

When we indicate the For Each command to use (using) a Data Selector, we will be telling it to add its orders and filters to that of the For Each command. For this reason, the attributes that are included in the Data Selector will have to belong to the extended table of the For Each command base table.

The other possibility that we had mentioned, which implied using the in operator to filter in a Where clause, is shown in the syntax but it has been left out of this explanation. If you want, you can read about it in the following link: http://wiki.genexus.com/commwiki/servlet/wiki?5312,Data+Selectors+in+For+Each+command.

Data Selectors scope



The Data Selector is an object to store a set of parameters, conditions, orders and defined by clauses, to be used/invoked from different queries and calculations, and reuse the same definitions (navigation) several times.

So, where a Data Selector can be used? In all places where queries to the database are specified.

So far we only know the for eachs and formulas. In further chapters we will study the Grids and the Data Providers.

More documentation about Data selectors:
http://wiki.genexus.com/commwiki/servlet/wiki?5271,Category%3AData+Selector+object

GeneXus™

The power of doing.

Videos	training.genexus.com
Documentation	wiki.genexus.com
Certifications	training.genexus.com/certifications