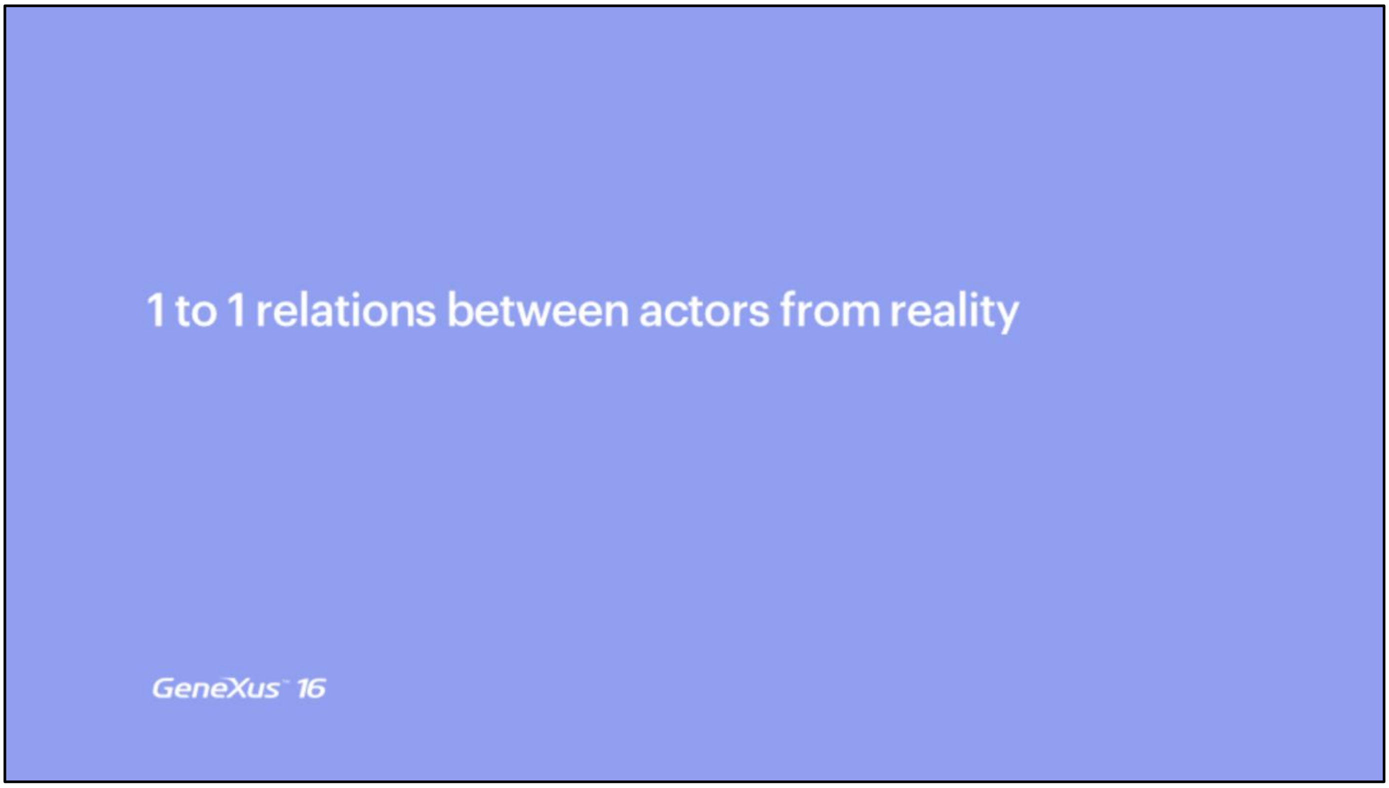
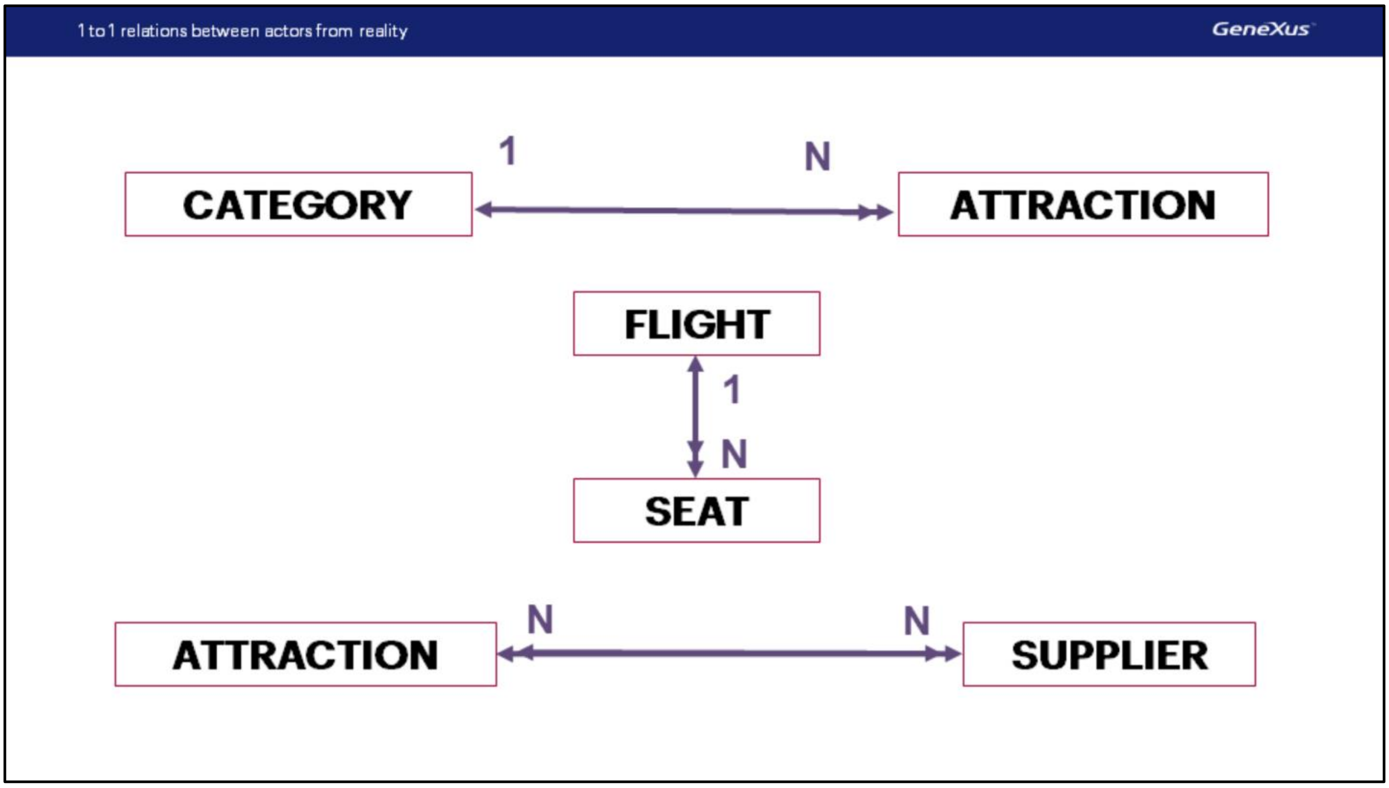


1 to 1 relations between actors from reality

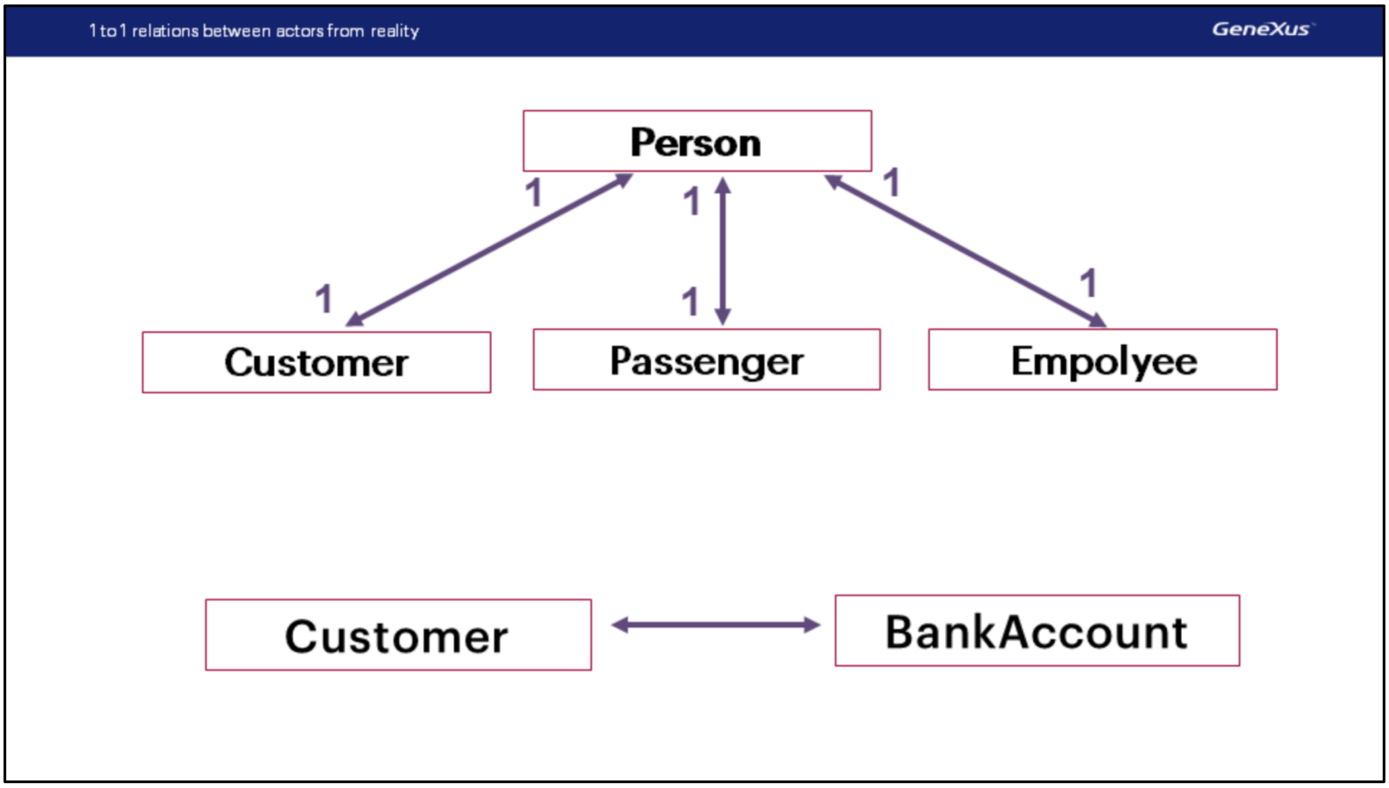
GeneXus 16



1 to 1 relations between actors from reality

GeneXus

We already saw that, with transactions and their attributes, we could represent strong and weak 1 to N relations between actors from our reality, in addition to N to N relations.

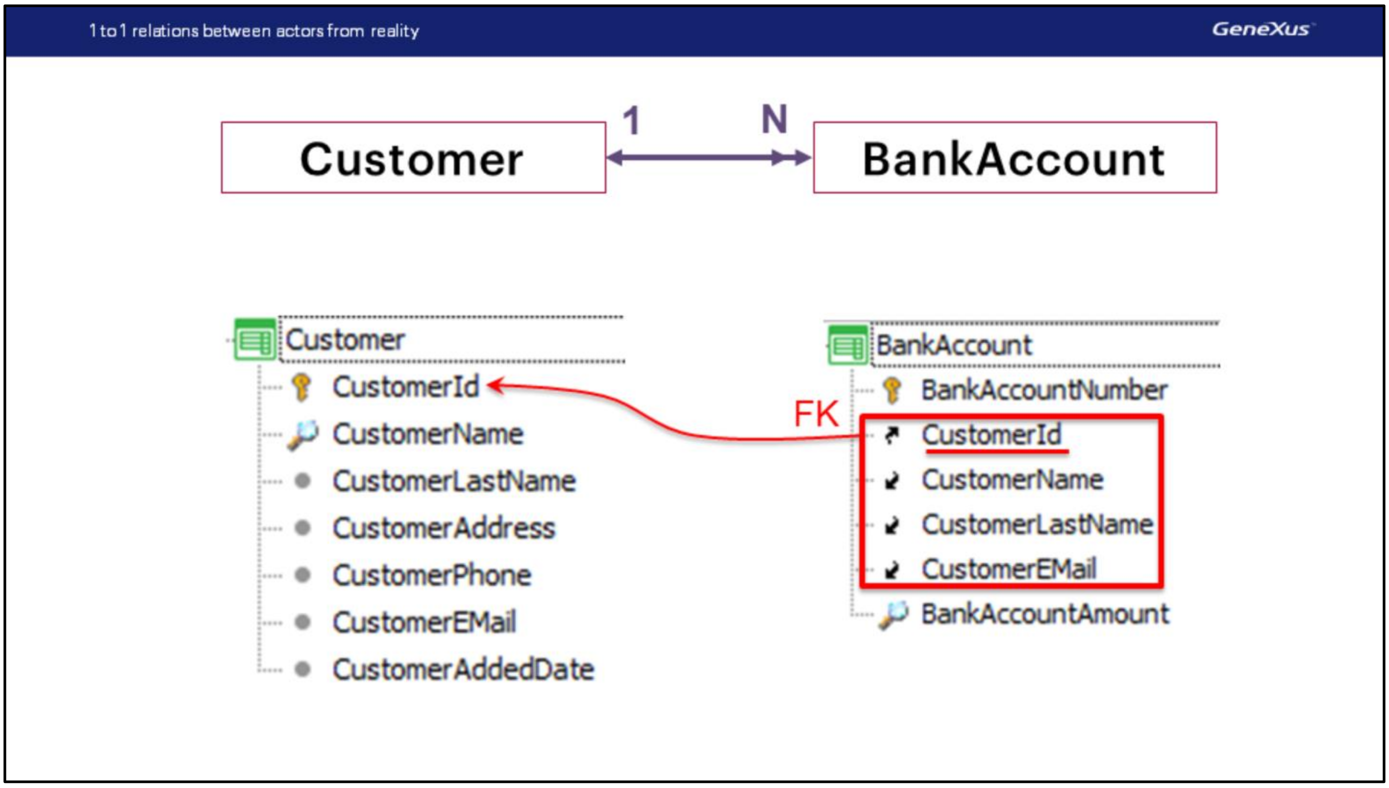


And we had mentioned for the 1 to 1 relations resulting from specialization, this was studied in the subtypes class as one of the use cases.

Here we will see the other case of 1 to 1 relations that we introduced before.

This is the case, for instance, of when the travel agency needs to associate each customer with the bank account that is opened for making the payments of services contracted.

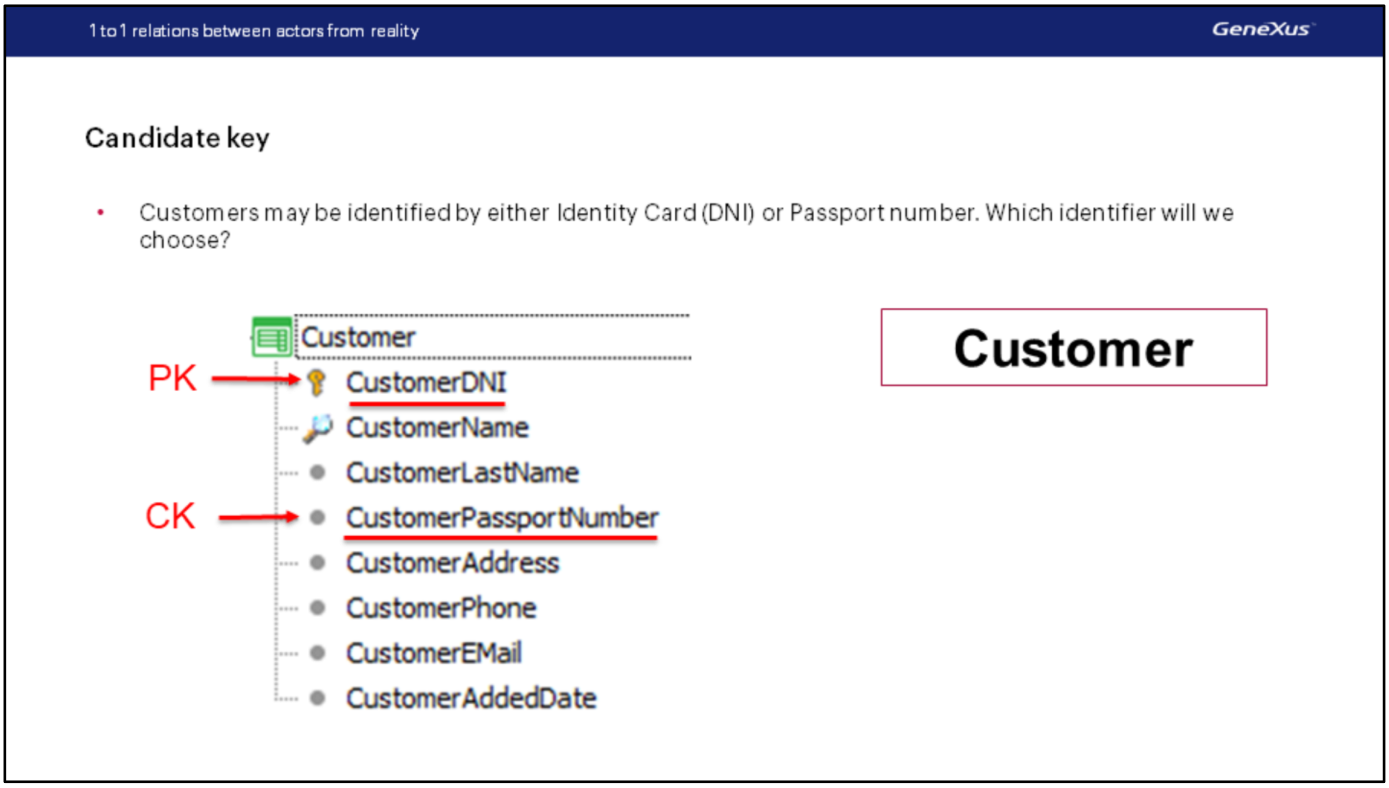
1 to 1 relations between actors from reality



In addition to the Customer transaction we will also have the BankAccount transaction, identified by a number, and with only one associated customer.

But we need to do something else, because this design corresponds to a 1 to N relation. This means that, even when each account may have only one customer, each customer could be associated with many bank accounts. Note that CustomerId is **foreign key** in BankAccount, referencing the Customer table.

We must also ensure that the CustomerId attribute will not be repeated in BankAccount. In other words, there may not be two or more records with the same value for that attribute. It is like saying that we want the attribute to be a **candidate key.** We will be getting back to this further ahead when we consider indices.



Candidate key

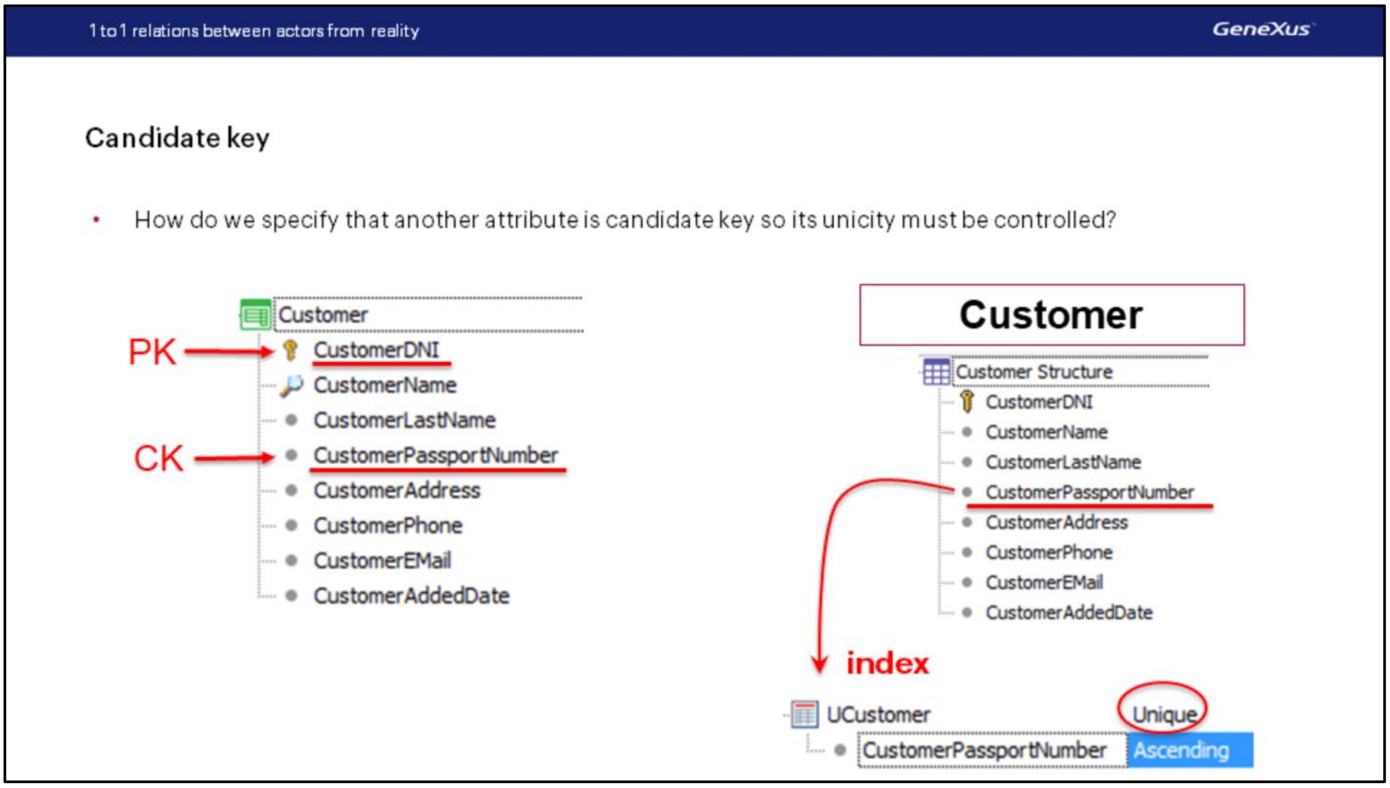
- Customers may be identified by either Identity Card (DNI) or Passport number. Which identifier will we choose?

Customer
PK → CustomerDNI
CustomerName
CustomerLastName
CK → CustomerPassportNumber
CustomerAddress
CustomerPhone
CustomerEMail
CustomerAddedDate

Customer

For each level of each transaction me must define the attribute or group of attributes that comprise the level identifier. That identifier, at the physical table level, will become the primary key of the table. This means that the values of that attribute or group of attributes may not be repeated.

However, we often have more than one attribute or group of attributes that must fulfill that condition. For example, we decide to identify the customer with his-her identity card issued by the customer's country of origin. But we could also have a passport number as secondary attribute, which should also be one and only. Since we have to choose one of the two to identify the entity, if we do nothing else, then the other one will remain as secondary attribute, with the possibility of being repeated.



GeneXus

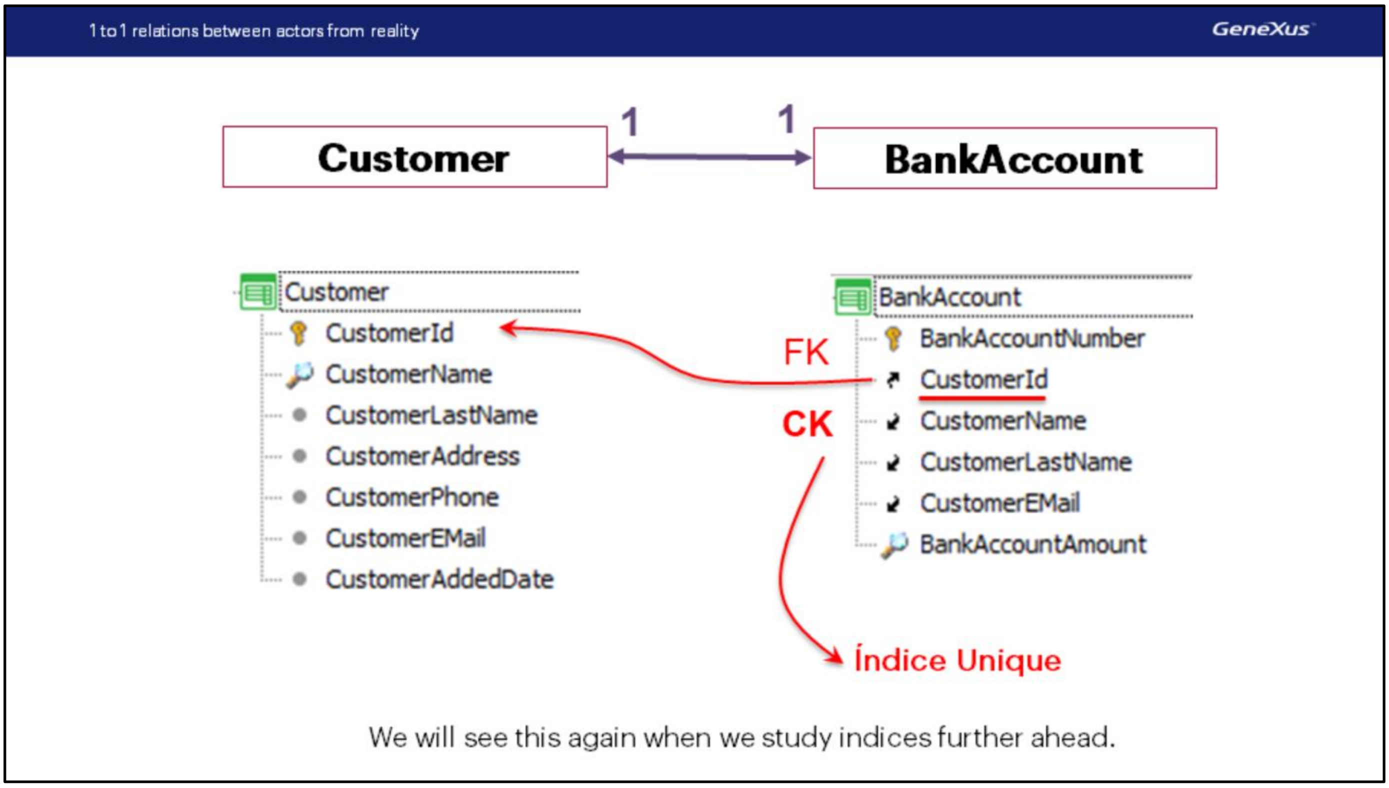
Candidate key

- How do we specify that another attribute is candidate key so its unicity must be controlled?

The way to tell GeneXus that the CustomerPassportNumber attribute is a candidate key is to define an index at the table level, on that attribute, and establish that the index must control that values are not repeated, in other words: a UNIQUE index.

We should recall that **indices** are efficient ways to access data. We could think, for example, of a recipe book with numerous pages and indices (alphabetic index, type of food index, etc.) Likewise, tables used to store records also have indices to sort the records and making it easier to find them.

We will be seeing this again later on, including the indices created automatically by GeneXus, based on the tables, and user indices. We will see how they are defined, but for the time being we will just mention the concept.



We will see this again when we study indices further ahead.

The design solution that will allow us to control that the customer is not repeated on the BankAccount table will be to define it not only as foreign key (which is done automatically just by giving it the same name as the primary key attribute of another table) but also as candidate key, that is, through a **Unique** index.

GeneXus™

The power of doing.

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