Subtipos

Una mirada integradora

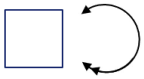
GeneXus™

Los subtipos nos permiten indicarle a GeneXus cómo asociar distintos nombres de atributo a un mismo concepto.

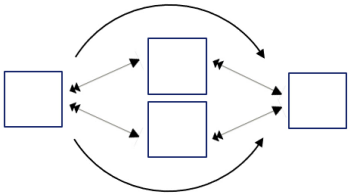
Cases of Subtypes

Direct references

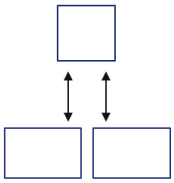
Recursive Subtypes



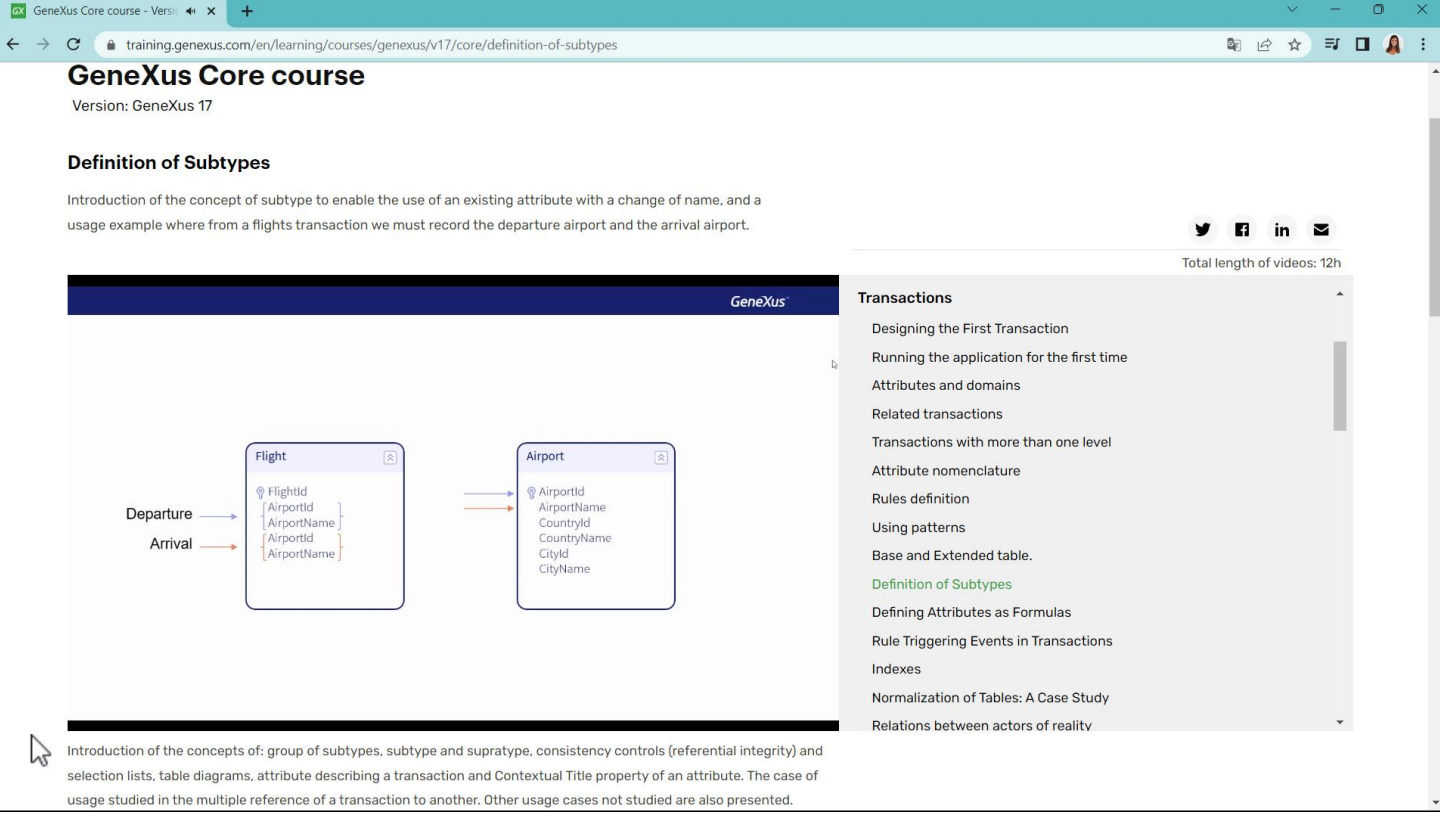
Indirect references



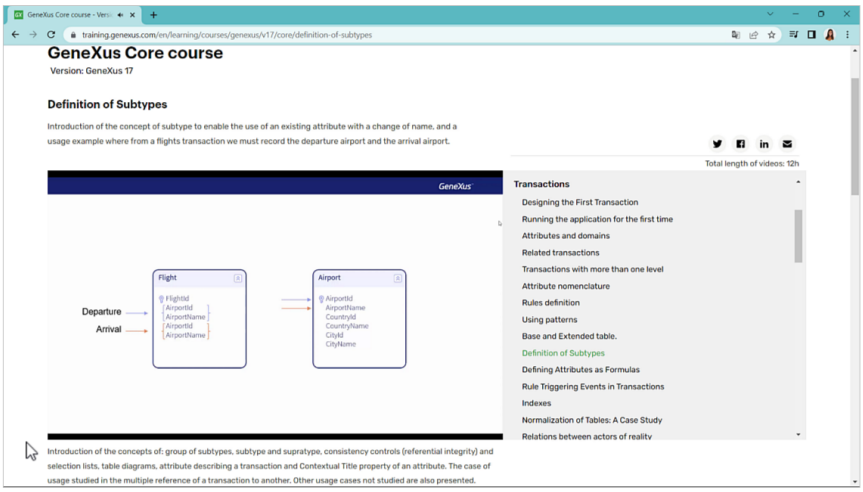
Specialization



En cursos anteriores, hemos analizado y estudiado diferentes casos en los que definimos grupos de subtipos para resolver los conflictos o ambigüedades que surgen en nuestras aplicaciones.



Comenzamos con el ejemplo más simple, donde teníamos una doble referencia a un mismo concepto, pero con distintos roles.



Era el caso en el que nos solicitaban poder registrar tanto el aeropuerto de partida como el aeropuerto de llegada de determinado vuelo:

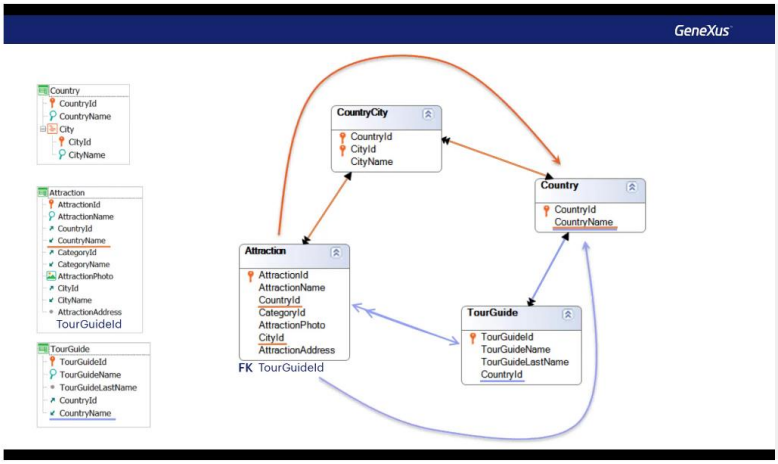
GeneXus Advanced course

Version: GeneXus 17

Subtypes: Multiple References and Specialization

A case of indirect multiple references and a specialization case are presented.

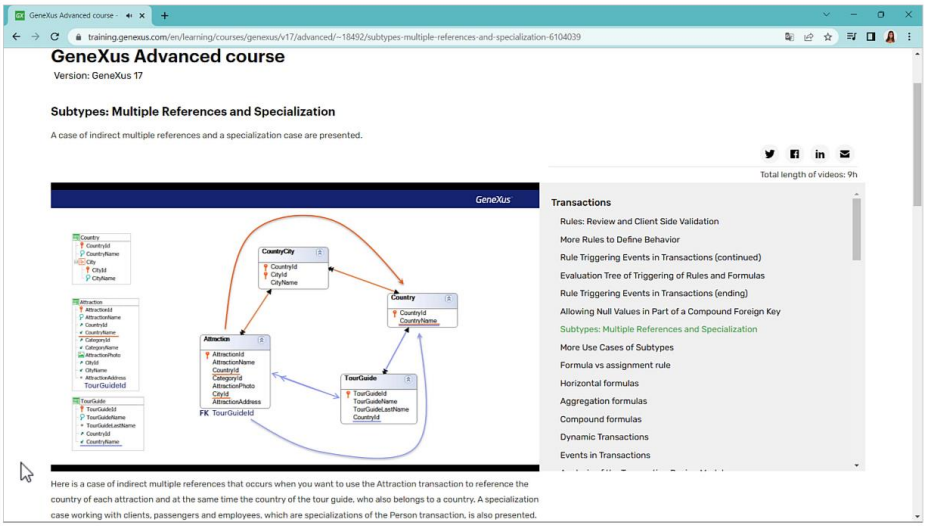
Total length of videos: 9h



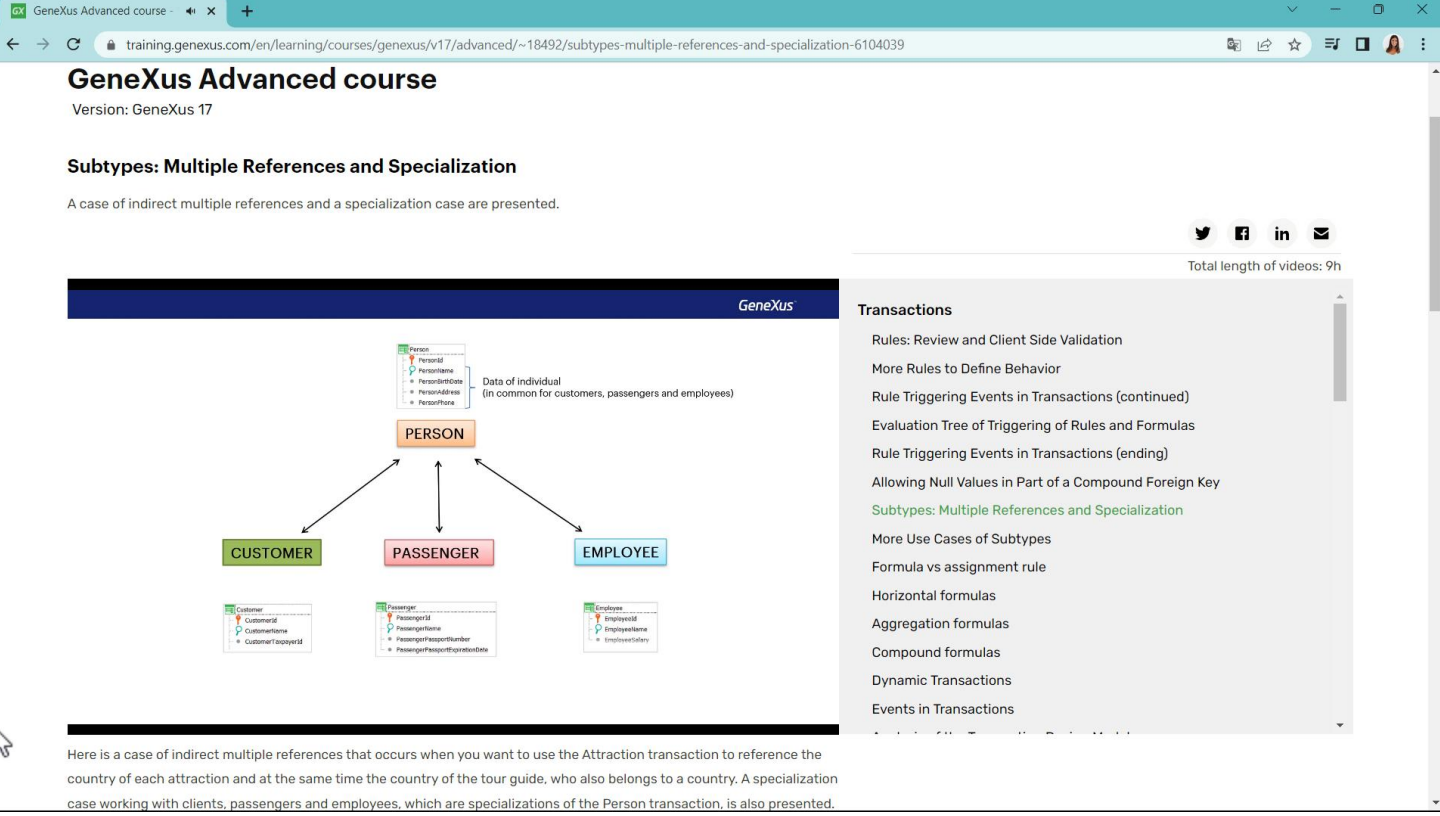
Here is a case of indirect multiple references that occurs when you want to use the Attraction transaction to reference the country of each attraction and at the same time the country of the tour guide, who also belongs to a country. A specialization case working with clients, passengers and employees, which are specializations of the Person transaction, is also presented.

Transactions

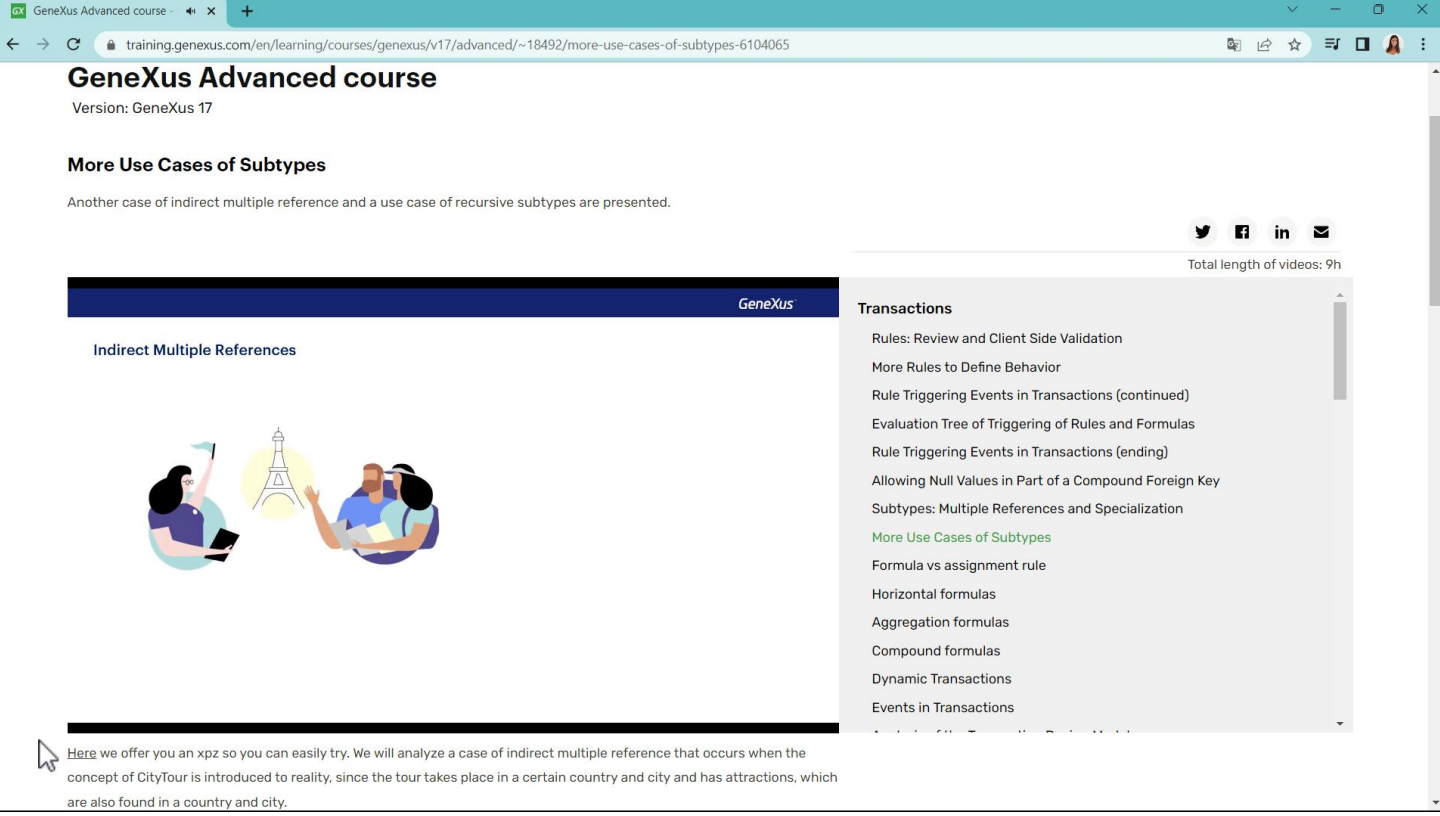
Más adelante, estudiamos un caso de referencias múltiples indirectas, ya que, dada una tabla, teníamos dos caminos para llegar a otra:



Era el caso en el que debíamos registrar la información de los guías turísticos, donde se nos presentaban dos caminos para llegar al atributo CountryId para identificar su país.



En este mismo video, vimos un caso de uso de subtipos para representar una especialización, donde una transacción registraba la información común de las personas, y otras transacciones (especializaciones de la primera) registraban la información particular.



En este otro video, nos adentramos aún más en un caso de referencia múltiple indirecta, cuando debimos registrar los tours ofrecidos a los clientes de la agencia de viajes para visitar las atracciones turísticas de una ciudad determinada.

GeneXus Advanced course - training.genexus.com/en/learning/courses/genexus/v17/advanced/~18492/more-use-cases-of-subtypes-6104065

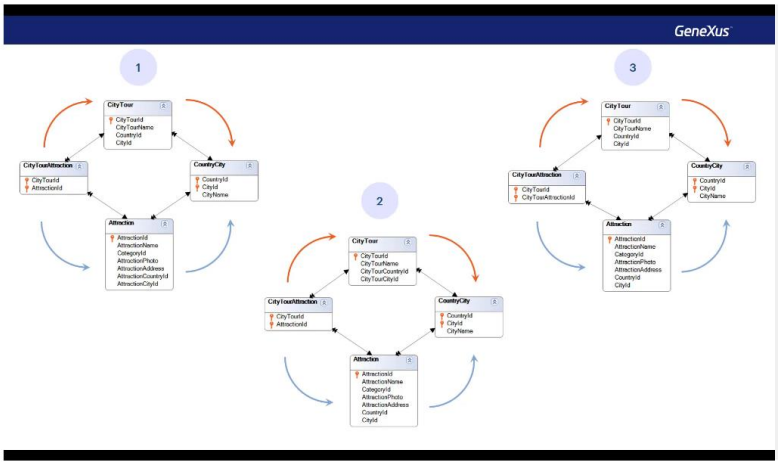
GeneXus Advanced course

Version: GeneXus 17

More Use Cases of Subtypes

Another case of indirect multiple reference and a use case of recursive subtypes are presented.

Total length of videos: 9h

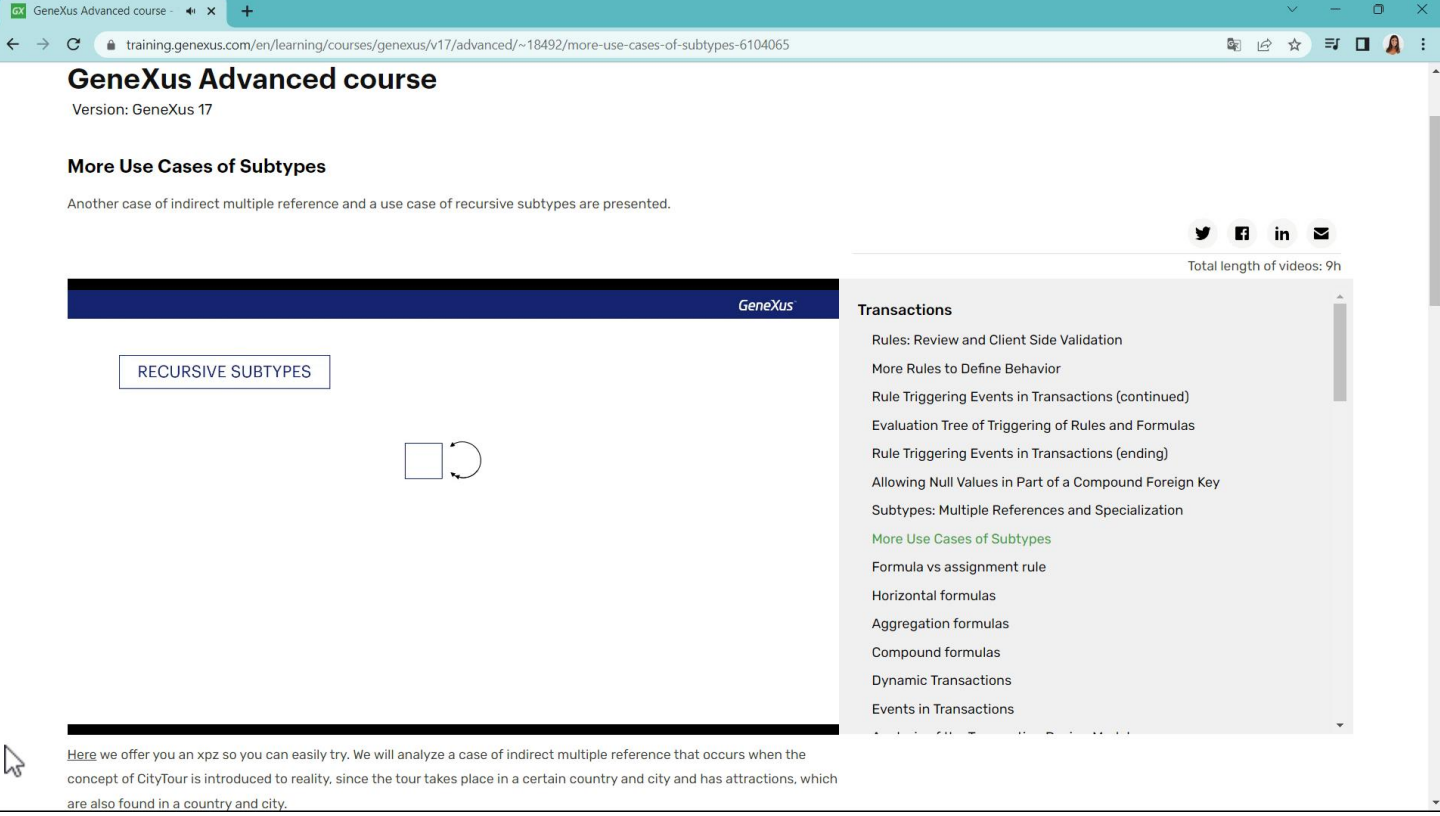


Here we offer you an xpz so you can easily try. We will analyze a case of indirect multiple reference that occurs when the concept of CityTour is introduced to reality, since the tour takes place in a certain country and city and has attractions, which are also found in a country and city.

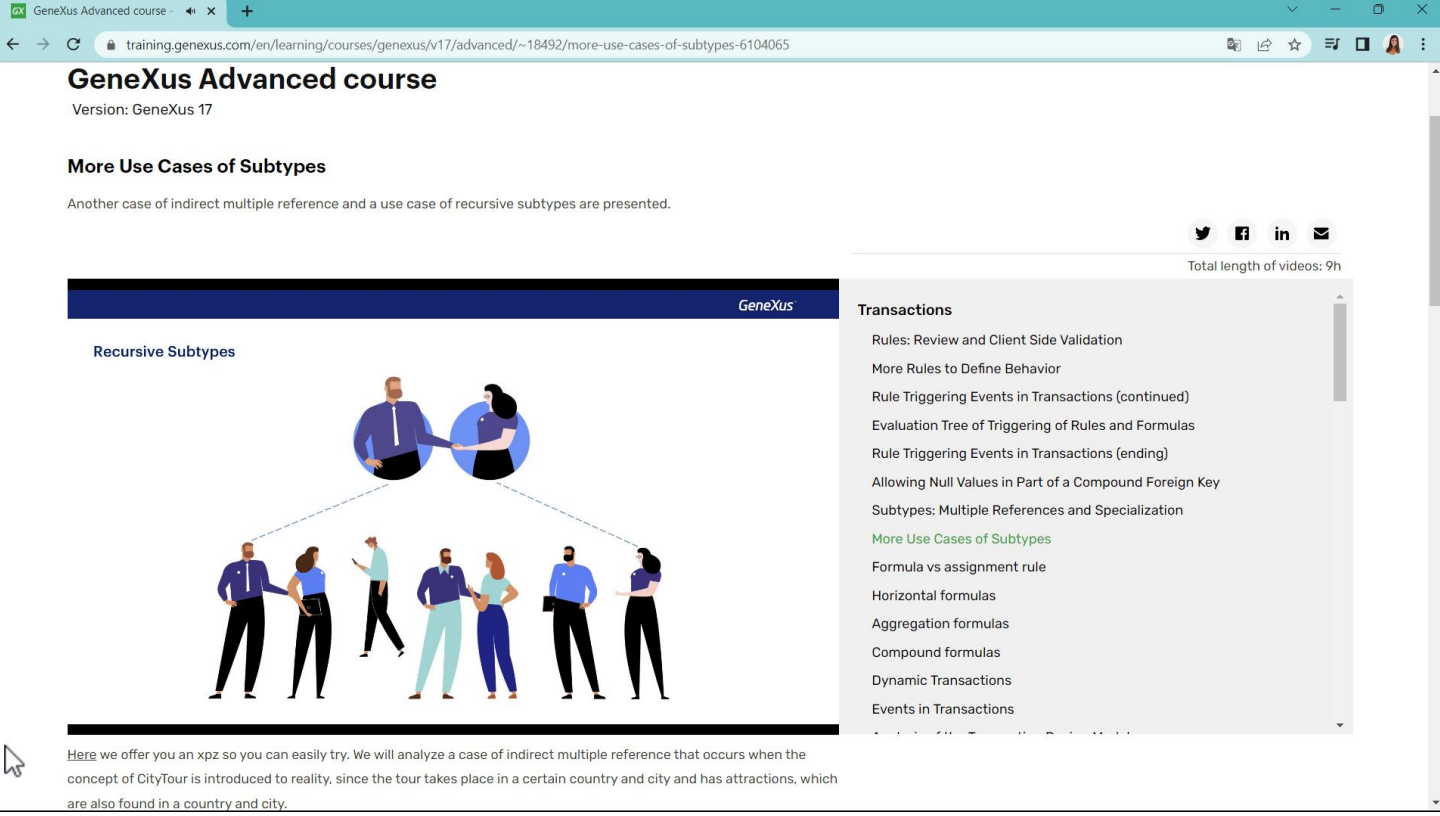
Transactions

- Rules: Review and Client Side Validation
- More Rules to Define Behavior
- Rule Triggering Events in Transactions (continued)
- Evaluation Tree of Triggering of Rules and Formulas
- Rule Triggering Events in Transactions (ending)
- Allowing Null Values in Part of a Compound Foreign Key
- Subtypes: Multiple References and Specialization
- More Use Cases of Subtypes
- Formula vs assignment rule
- Horizontal formulas
- Aggregation formulas
- Compound formulas
- Dynamic Transactions
- Events in Transactions

En este ejemplo, analizamos exhaustivamente el problema con sus diferentes soluciones, cada una con sus ventajas y desventajas, destacando la importancia de estudiar cada caso particular y determinar utilizar subtipos realmente cuando y donde se necesitan y no arbitrariamente.

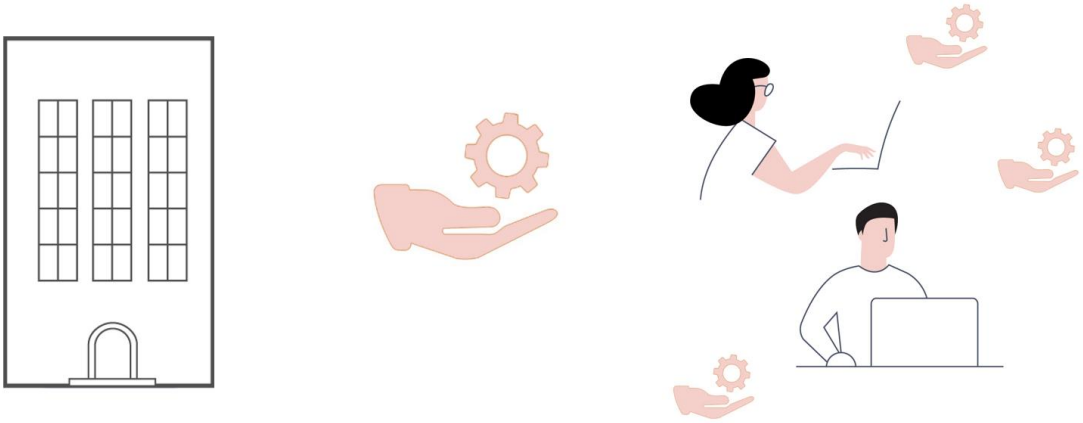


Presentamos también un caso de uso de subtipos recursivos, donde una entidad debía auto referenciarse:



Se trataba de la realidad en la que representábamos la información de los empleados de la agencia de viajes, donde cada empleado podía ser, a su vez, gerente de otro u otros empleados.

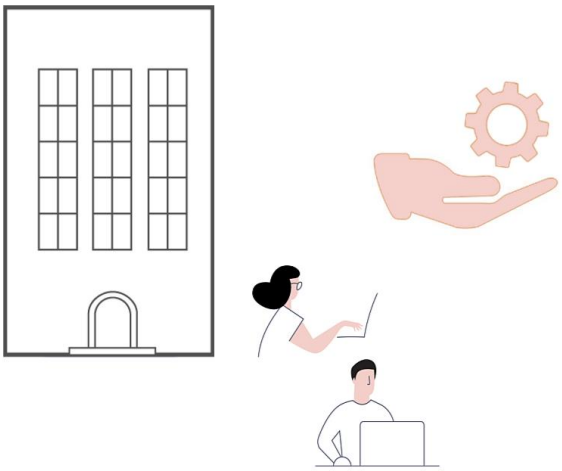
New case: Avoid referential relationship



Estudiemos ahora un último ejemplo en el que debemos evitar la relación referencial.

Supongamos que debemos modelar las transacciones para una realidad en la que se tienen empresas y servicios que éstas pueden contratar (como, por ejemplo, un servicio de emergencia médica). A su vez, las empresas tienen empleados que también pueden tener contratados servicios que no tienen por qué coincidir con los de la empresa para la que trabajan. Nos interesa registrar esos servicios de los empleados porque así, si, por ejemplo, muchos empleados tienen contratado determinado servicio de emergencia médica, puede intentarse un convenio con ese servicio para obtener algún beneficio.

New case: Avoid referential relationship

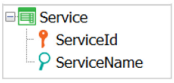


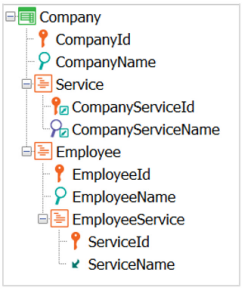
Employees can only work in one company, but they are not to be represented as a strong entity, but as dependent on the company.

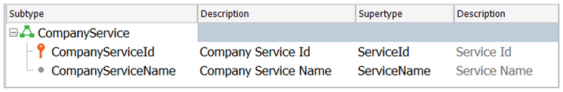
En nuestra realidad, los empleados solamente pueden trabajar en una empresa, pero no se los quiere representar como una entidad fuerte, sino dependiente de la empresa. Veamos estas dos soluciones propuestas, donde una es correcta y la otra no lo es.

Two possible solutions

Solution A)





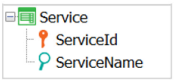


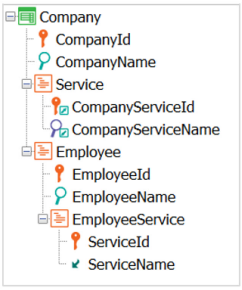
Subtype	Description	Supertype	Description
CompanyService			
CompanyServiceId	Company Service Id	ServiceId	Service Id
CompanyServiceName	Company Service Name	ServiceName	Service Name

A) La primera solución sería creando estas dos transacciones y el siguiente grupo de subtipos:

Two possible solutions

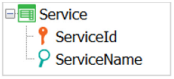
Solution A)

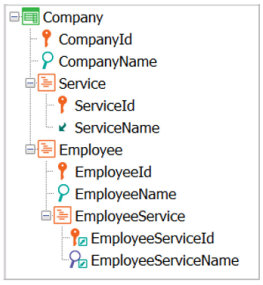




Subtype	Description	Supertype	Description
CompanyService			
CompanyServiceId	Company Service Id	ServiceId	Service Id
CompanyServiceName	Company Service Name	ServiceName	Service Name

Solution B)



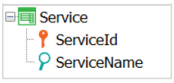


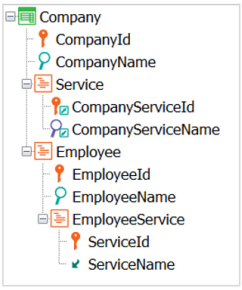
Subtype	Description	Supertype	Description
EmployeeService			
EmployeeServiceId	Employee Service Id	ServiceId	Service Id
EmployeeServiceName	Employee Service Name	ServiceName	Service Name

B) y la segunda solución sería creando estas dos transacciones y el siguiente grupo de subtipos:

Two possible solutions

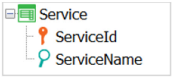
Solution A) ✓

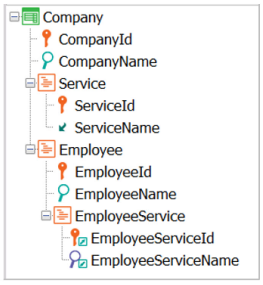




Subtype	Description	Supertype	Description
CompanyService			
CompanyServiceId	Company Service Id	ServiceId	Service Id
CompanyServiceName	Company Service Name	ServiceName	Service Name

Solution B) ✗



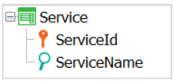


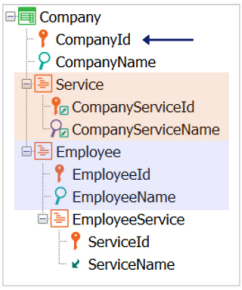
Subtype	Description	Supertype	Description
EmployeeService			
EmployeeServiceId	Employee Service Id	ServiceId	Service Id
EmployeeServiceName	Employee Service Name	ServiceName	Service Name

La solución correcta es la A) y no la B).

Two possible solutions

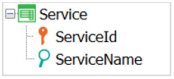
Solution A)

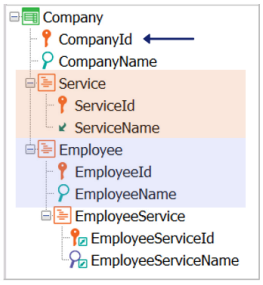




Subtype	Description	Supertype	Description
CompanyService			
CompanyServiceId	Company Service Id	ServiceId	Service Id
CompanyServiceName	Company Service Name	ServiceName	Service Name

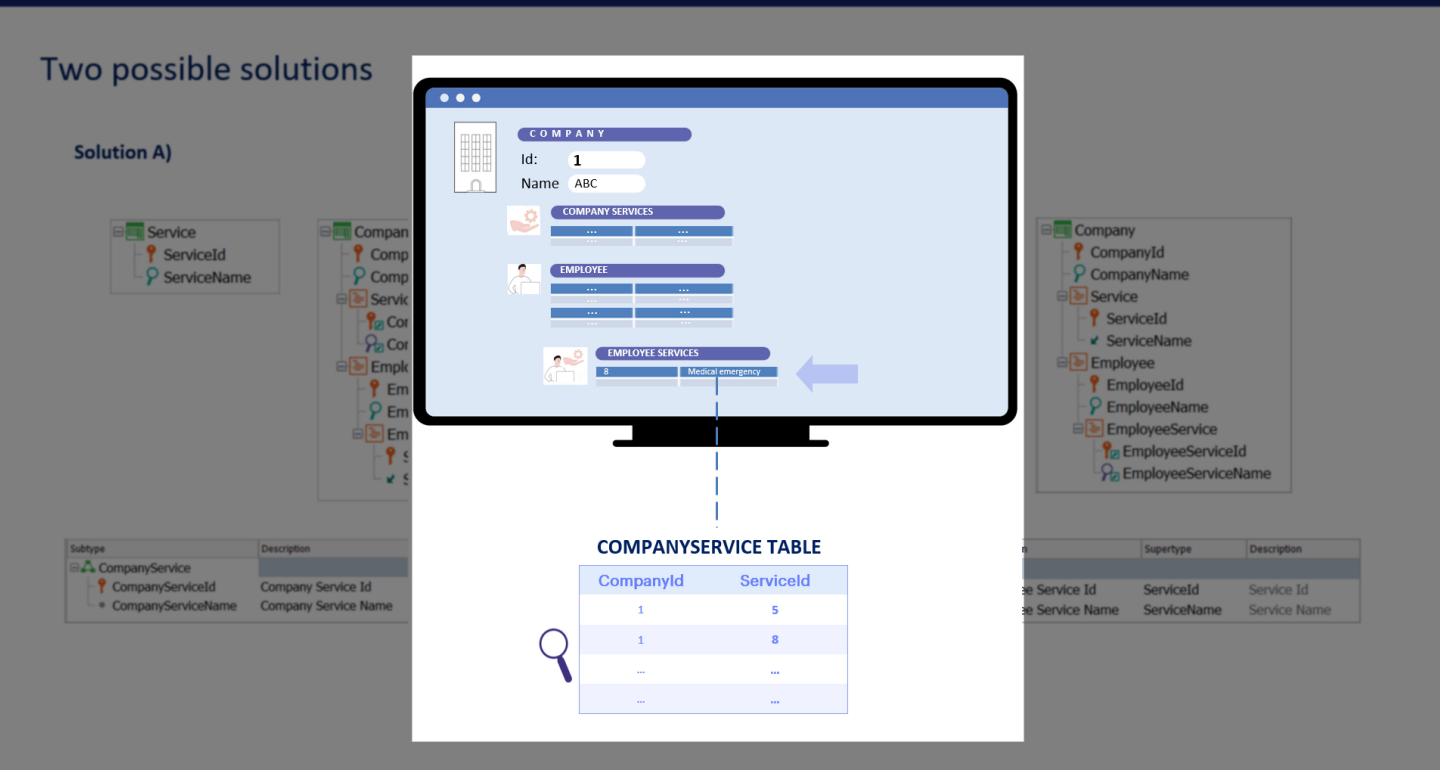
Solution B)





Subtype	Description	Supertype	Description
EmployeeService			
EmployeeServiceId	Employee Service Id	ServiceId	Service Id
EmployeeServiceName	Employee Service Name	ServiceName	Service Name

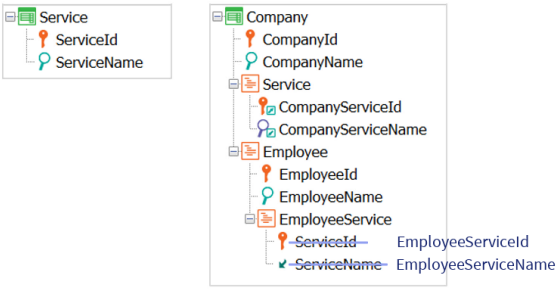
Si observamos con detenimiento, en la transacción Company tenemos dos niveles paralelos: Service y Employee. Esto significa que todo lo que se infiera de cualquiera de estos niveles corresponderá a la misma compañía. Sin embargo, no queremos que el servicio del empleado exista como servicio de la empresa, puesto que en nuestra realidad el empleado podría tener contratados servicios diferentes a los de la compañía para la que trabaja.



Dicho de otro modo: no queremos que se chequee cuando el usuario ingresa en el grid de servicios del empleado que el servicio ingresado exista como registro en la tabla correspondiente a Company.Service.

Two possible solutions

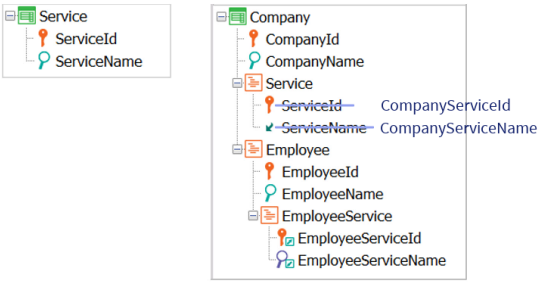
Solution A)



Subtype	Description	Supertype	Description
CompanyService			
CompanyServiceId	Company Service Id	ServiceId	Service Id
CompanyServiceName	Company Service Name	ServiceName	Service Name

EmployeeService			
EmployeeServiceId	...	ServiceId	...
EmployeeServiceName	...	ServiceName	...

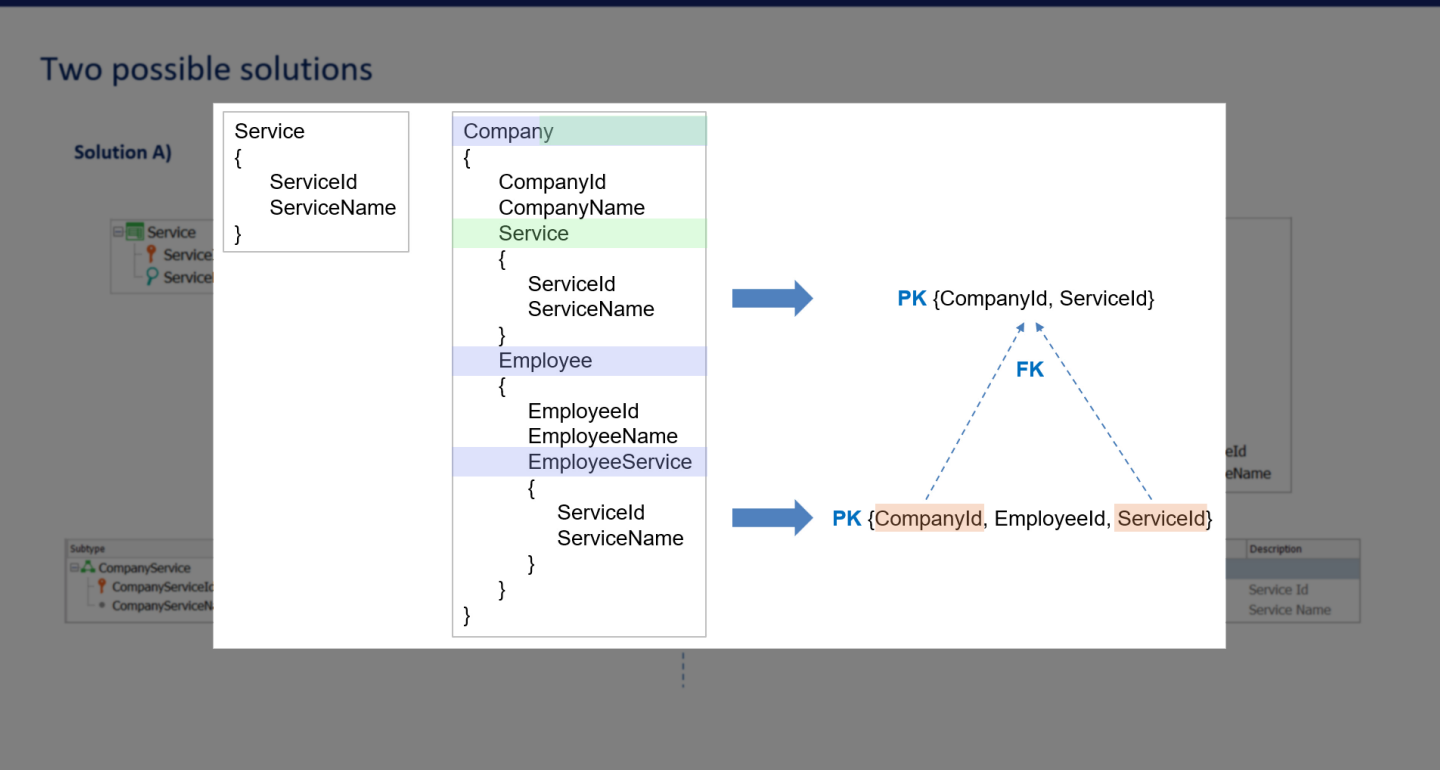
Solution B)



Subtype	Description	Supertype	Description
EmployeeService			
EmployeeServiceId	Employee Service Id	ServiceId	Service Id
EmployeeServiceName	Employee Service Name	ServiceName	Service Name

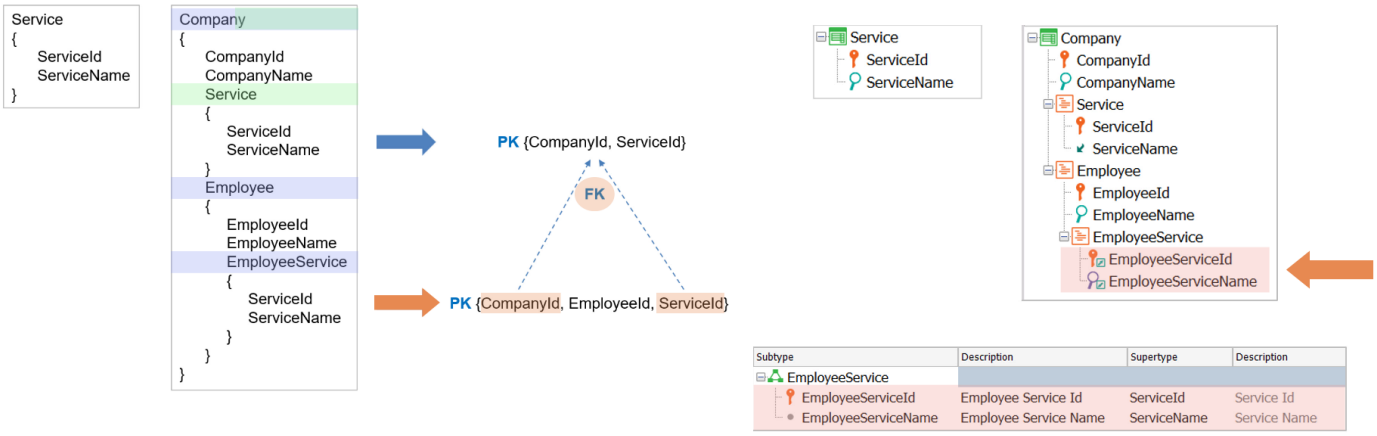
CompanyService			
CompanyServiceId	...	ServiceId	...
CompanyServiceName	...	ServiceName	...

Es claro que necesitamos definir un grupo de subtipos, puesto que en la misma transacción GeneXus no nos permitirá repetir el mismo nombre de atributo. La pregunta que surge entonces es: ¿da lo mismo definirlo en un nivel que en el otro? La respuesta es no. Podríamos definir dos grupos de subtipos y acabaríamos con el problema, pero, como ya hemos visto en videos anteriores, no es buena práctica definir más subtipos que los estrictamente necesarios, pues nunca es exactamente lo mismo tener el subtipo que tener el supertipo, como quedará claro con este ejemplo.



Por lo tanto, para resolver el problema nos alcanza con un único grupo. ¿Por qué, entonces, la solución correcta es la A) y no la B)? Es que, si GeneXus nos permitiera repetir el mismo nombre de atributo, claramente encontraría que en la tabla asociada al nivel Company.Employee.EmployeeService, de clave primaria {CompanyId, EmployeeId, ServiceId} los atributos {CompanyId, ServiceId} confromarían una clave foránea a la tabla correspondiente al nivel Company.Service (ya que su clave primaria sería {CompanyId, ServiceId}).

Two possible solutions

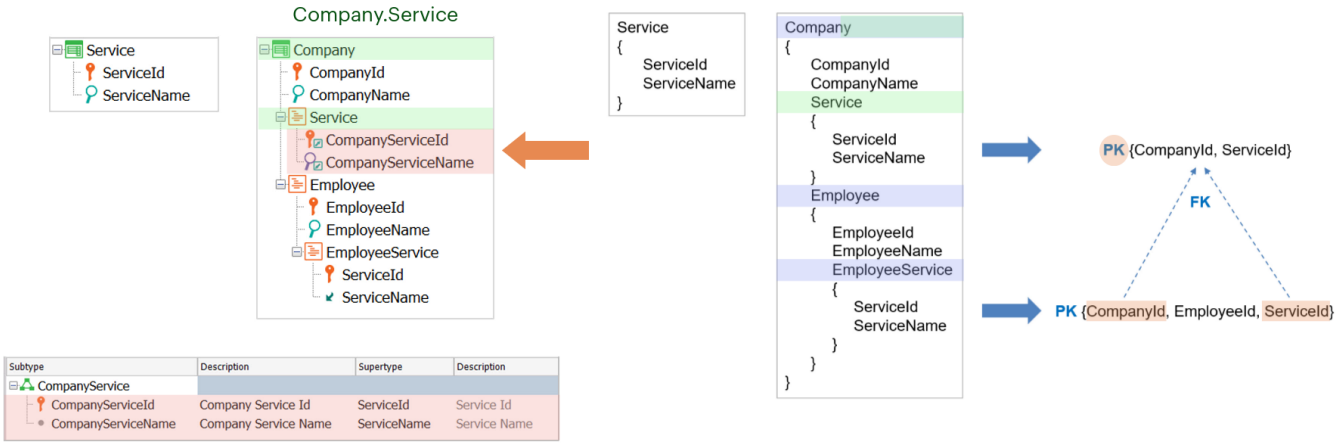


This does not delete for GeneXus its referential function

Pero si lo que hacemos es cambiar el nombre (con un subtipo) de ServiceId en la tabla en la que este atributo es parte de una clave foránea, esto no elimina para GeneXus su función referencial.

Two possible solutions

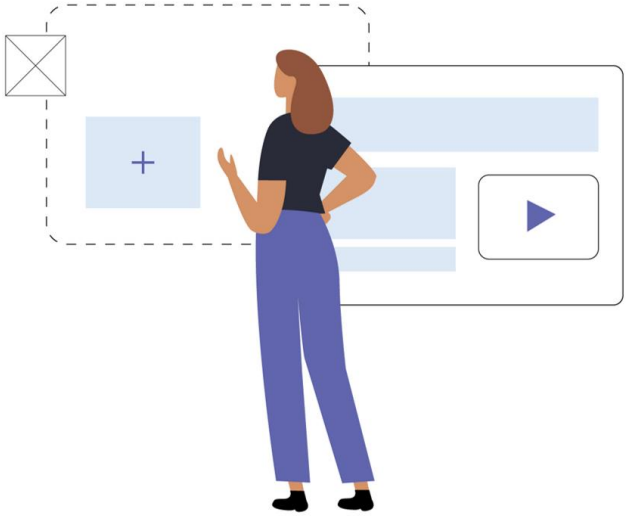
Solution A)



GeneXus does not establish the referential relationship

En cambio, si el atributo al que le cambiamos el nombre (usando un subtipo) es el que cumple la función de clave primaria, entonces en la tabla en la que el atributo supertipo aparece, no establece la relación referencial.

Subtypes are a compromise solution to solve problems and should be used cautiously.



Tanto estos casos más complejos como los más simples son comunes en aplicaciones de la vida real, y es el desarrollador quien deberá analizar los pros y contras de las diferentes soluciones para encontrar la que más se adecúe a cada caso en particular, recordando siempre que los subtipos son una solución de compromiso para resolver problemas y que deben usarse a discreción.

training.genexus.com
wiki.genexus.com
training.genexus.com/certifications