Subtipos

Uma visão integradora

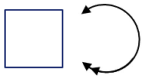
GeneXus

Os subtipos permitem indicar a GeneXus como associar diferentes nomes de atributo a um mesmo conceito.

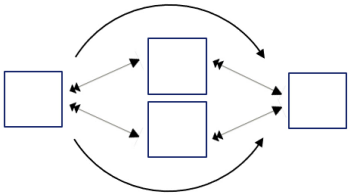
Cases of Subtypes

Direct references

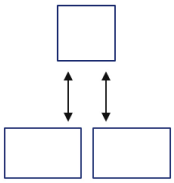
Recursive Subtypes



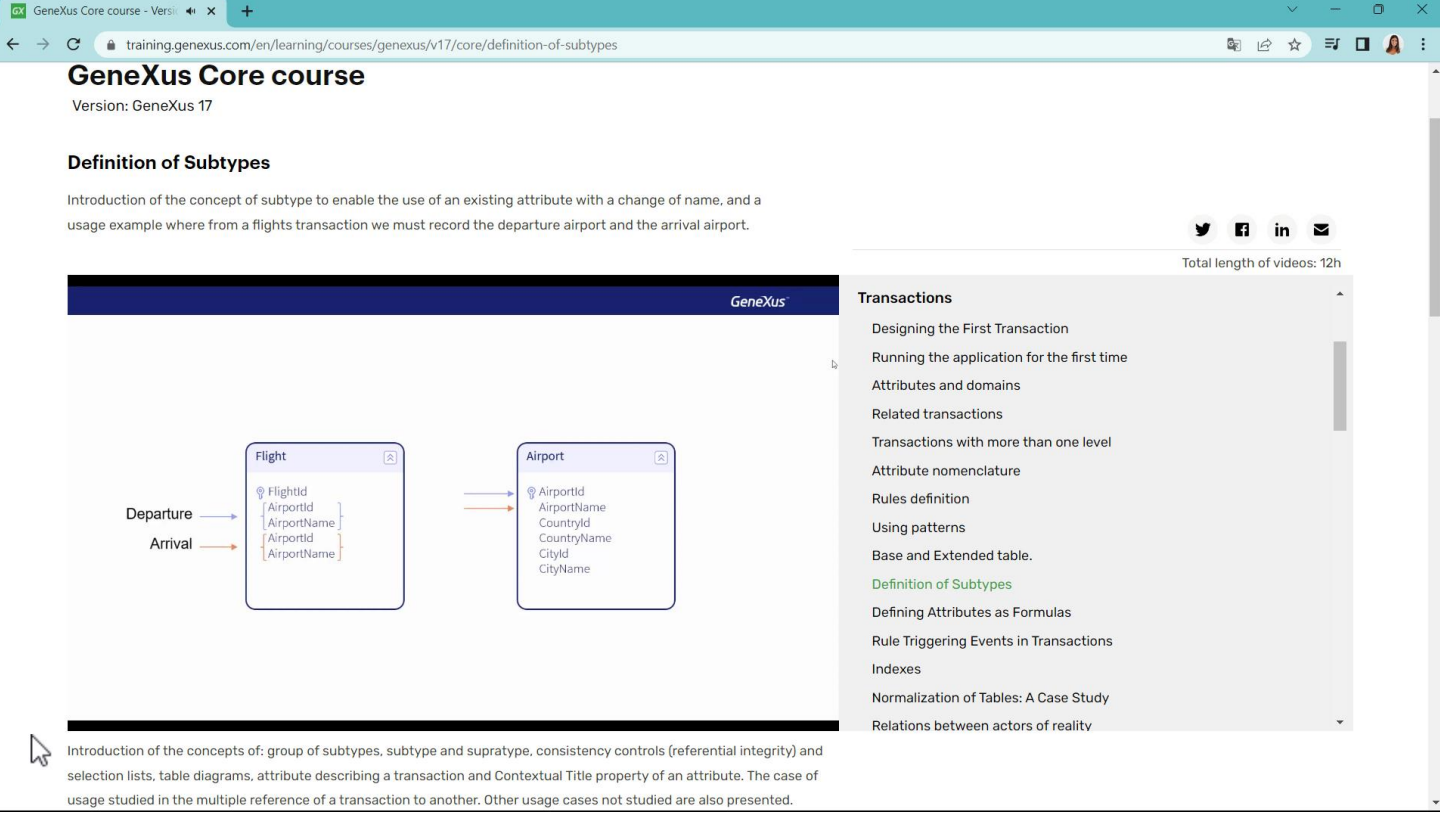
Indirect references



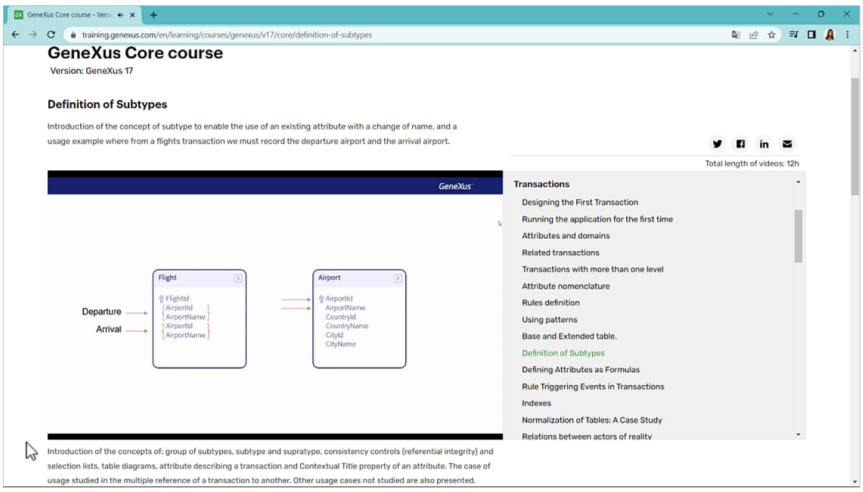
Specialization



Em cursos anteriores, analisamos e estudamos diferentes casos em que definimos grupos de subtipos para resolver os conflitos ou ambiguidades que surgem em nossas aplicações.



Começamos com o exemplo mais simples, onde tínhamos uma dupla referência a um mesmo conceito, mas com diferentes roles.



Era o caso em que nos solicitaram poder registrar tanto o aeroporto de partida como o aeroporto de chegada de determinado voo:

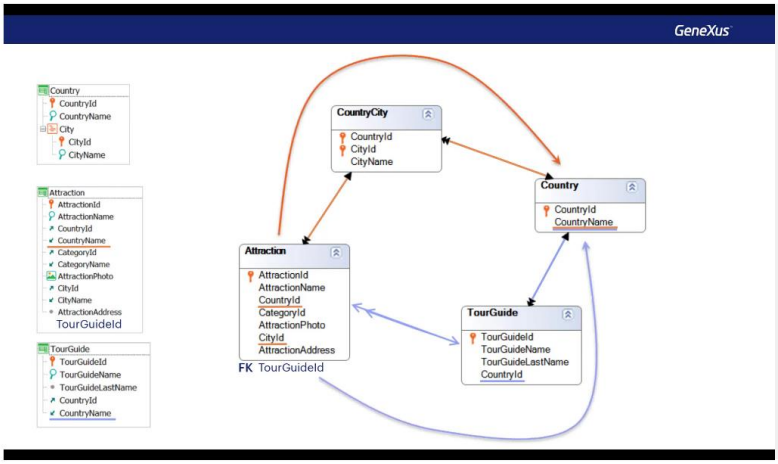
GeneXus Advanced course

Version: GeneXus 17

Subtypes: Multiple References and Specialization

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

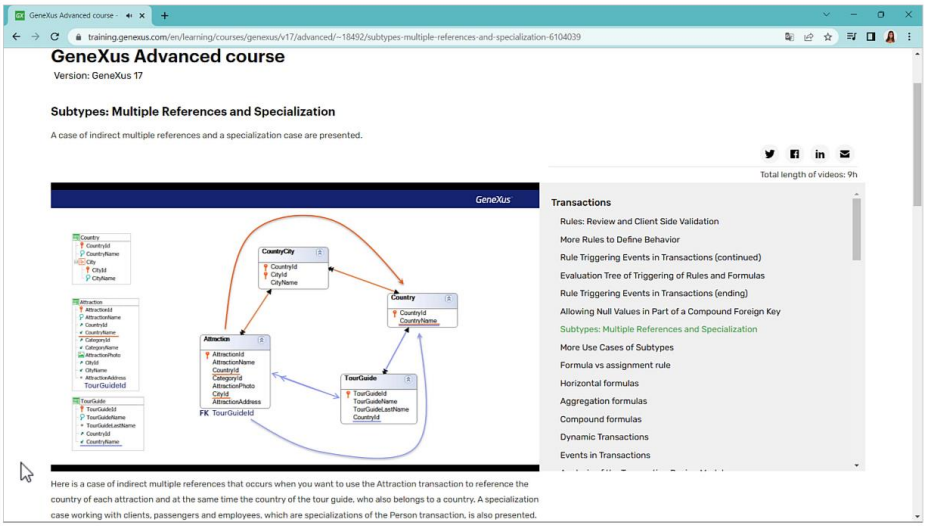
Total length of videos: 9h



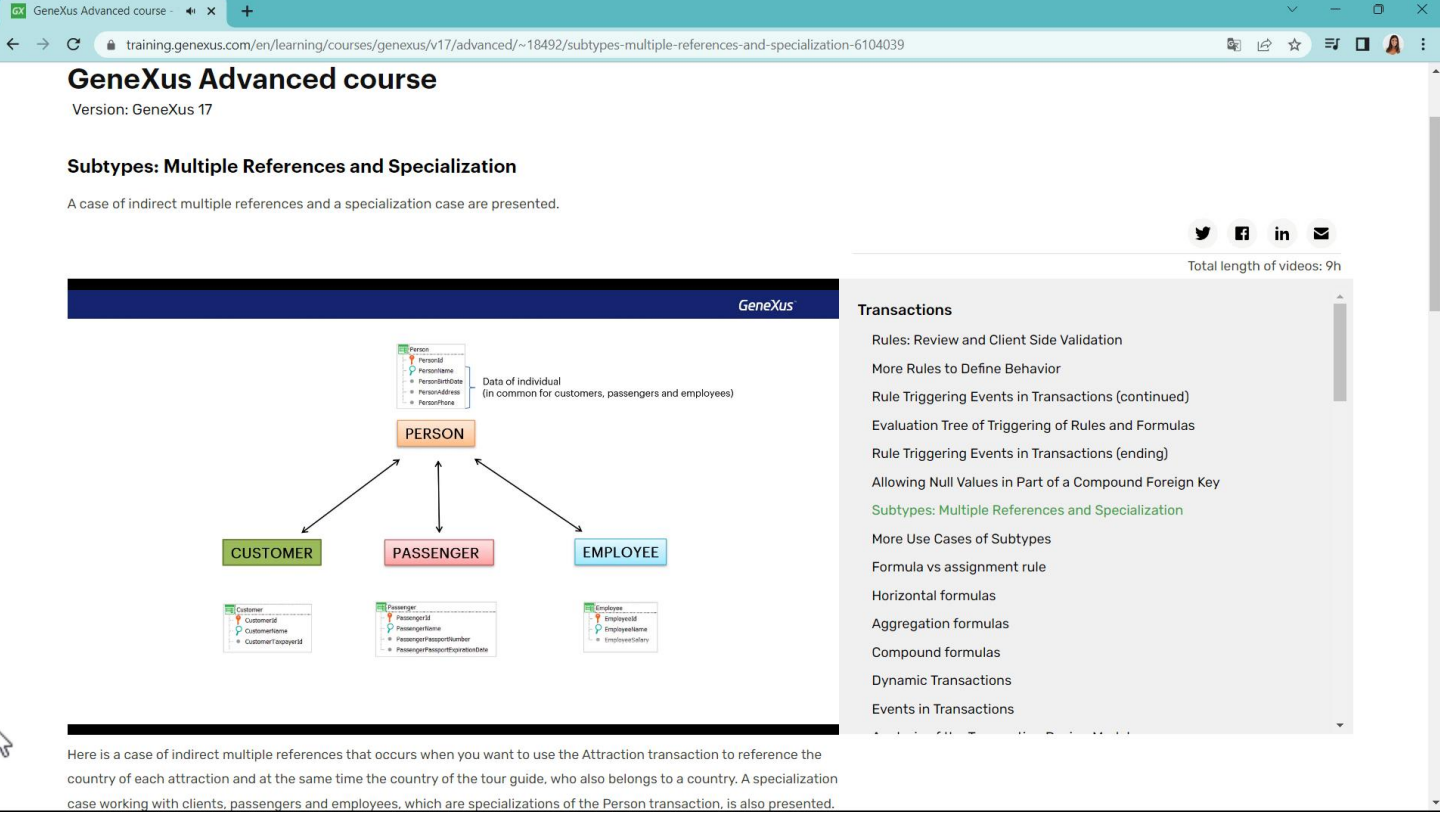
Here is a case of indirect multiple references that occurs when you want to use the Attraction transaction to reference the country of each attraction and at the same time the country of the tour guide, who also belongs to a country. A specialization case working with clients, passengers and employees,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

Transactions

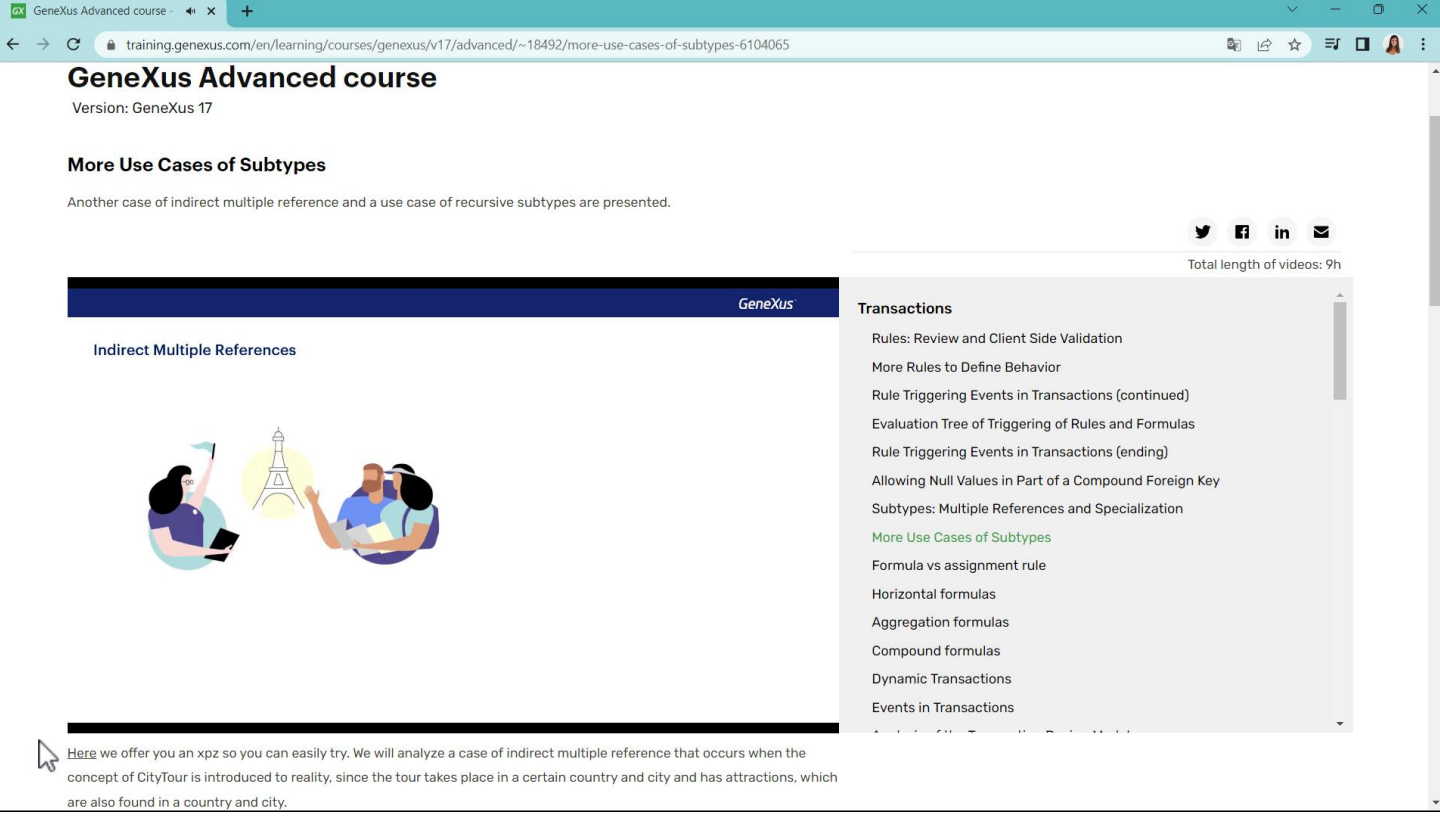
Mais adiante, estudamos um caso de referências múltiplas indiretas, pois, dada uma tabela, tínhamos dois caminhos para chegar a outra:



Era o caso em que devíamos registrar a informação dos guias turísticos, onde nos eram apresentados dois caminhos para chegar ao atributo CountryId para identificar seu país.



Neste mesmo vídeo, vimos um caso de uso de subtipos para representar uma especialização, onde uma transação registrava a informação comum das pessoas, e outras transações (especializações da primeira) registravam a informação particular.



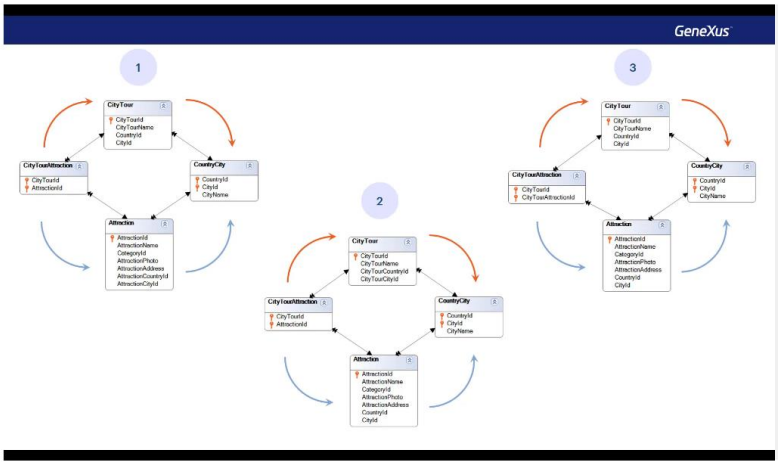
Neste outro vídeo, aprofundamos ainda mais em um caso de referência múltipla indireta, quando tivemos que registrar os tours oferecidos aos clientes da agência de viagens para visitar as atrações turísticas de uma cidade determinada.

GeneXus Advanced course

Version: GeneXus 17

More Use Cases of Subtypes

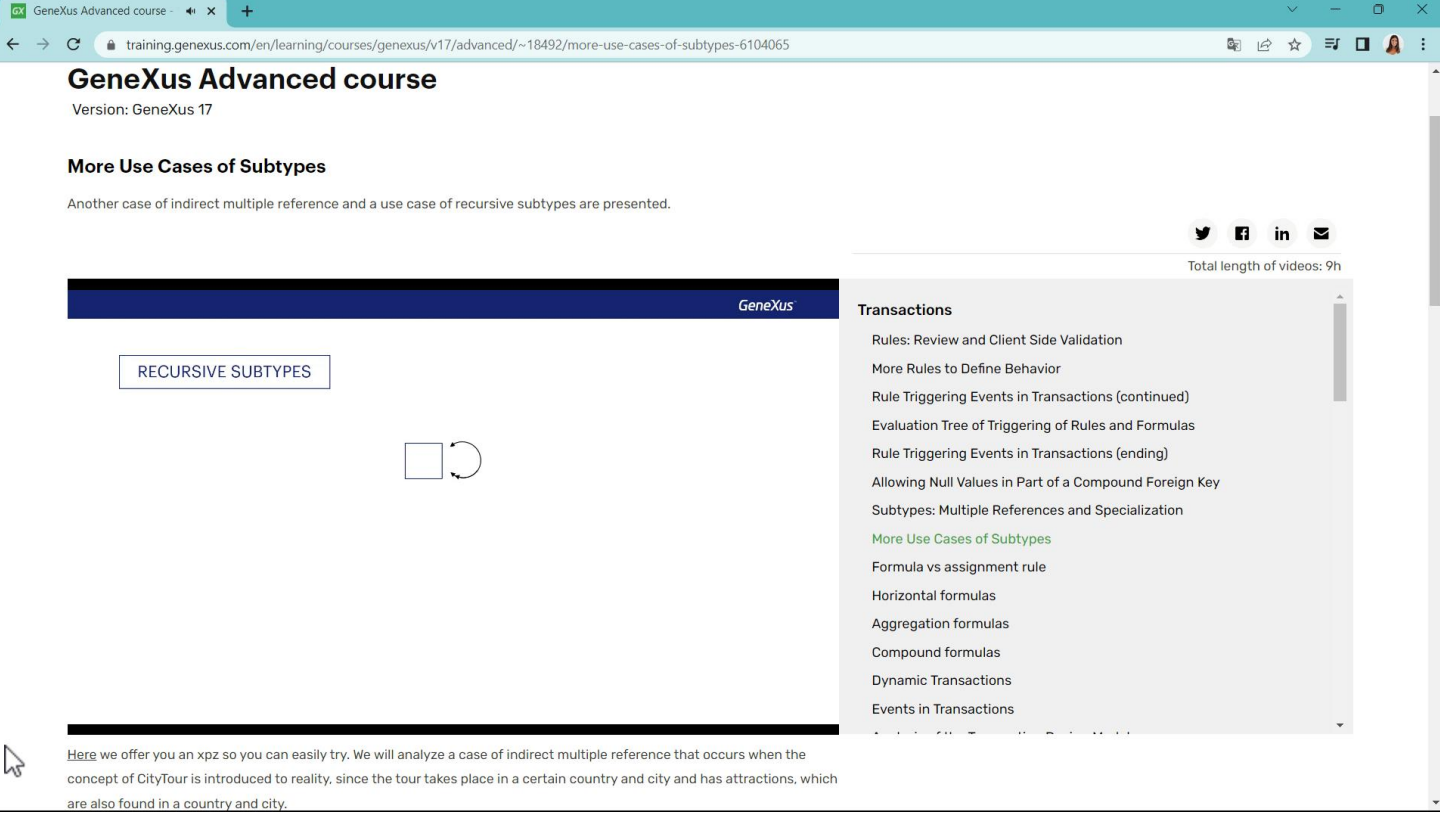
Another case of indirect multiple reference and a use case of recursive subtypes are presented.



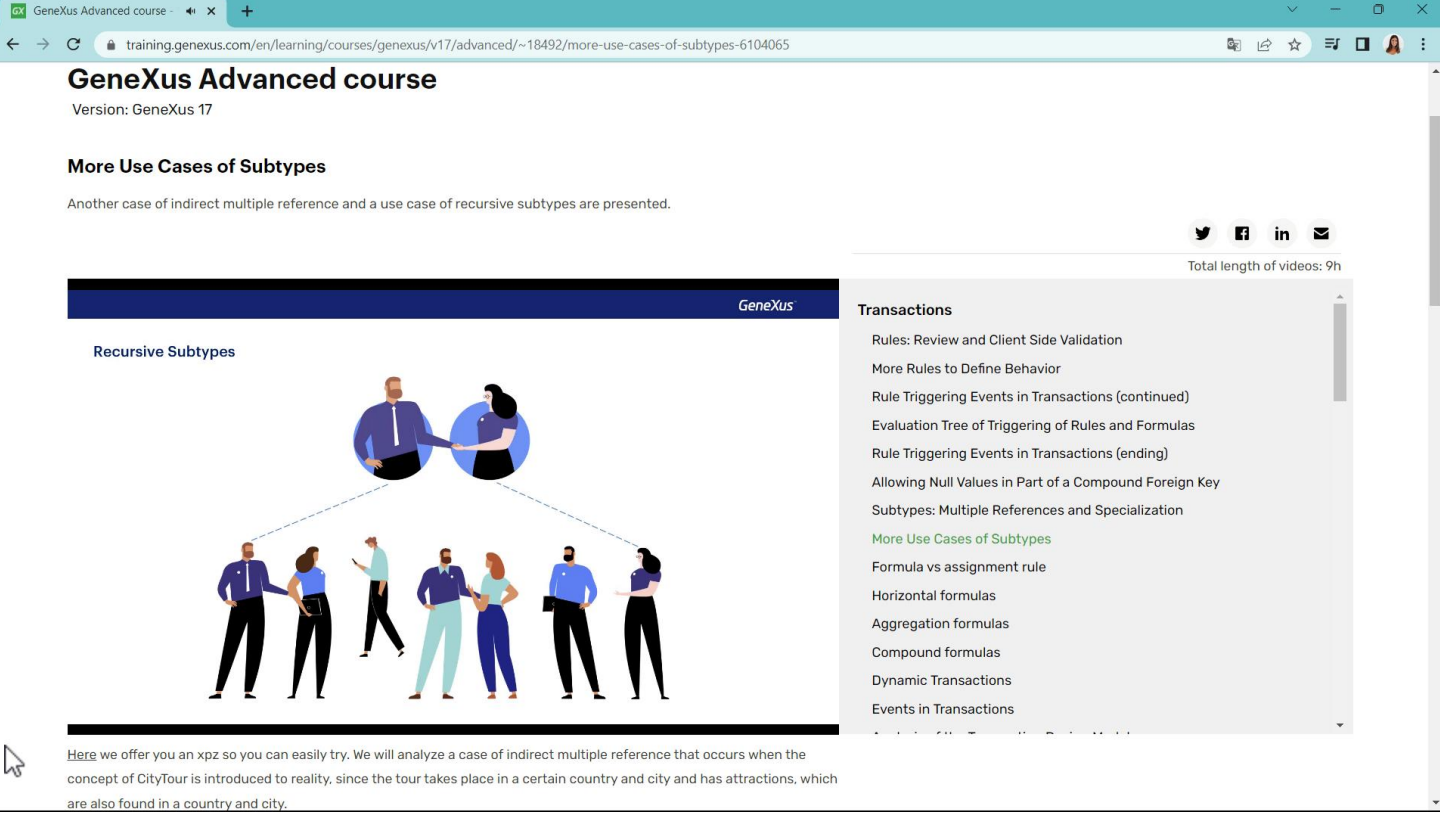
Here we offer you an xpz so you can easily try. We will analyze a case of indirect multiple reference that occurs when the concept of CityTour is introduced to reality, since the tour takes place in a certain country and city and has attractions, which are also found in a country and city.

Total length of videos: 9h

Neste exemplo, analisamos amplamente o problema com suas diferentes soluções, cada uma com suas vantagens e desvantagens, destacando a importância de estudar cada caso particular e determinar utilizar subtipos realmente quando e onde forem necessários e não arbitrariamente.

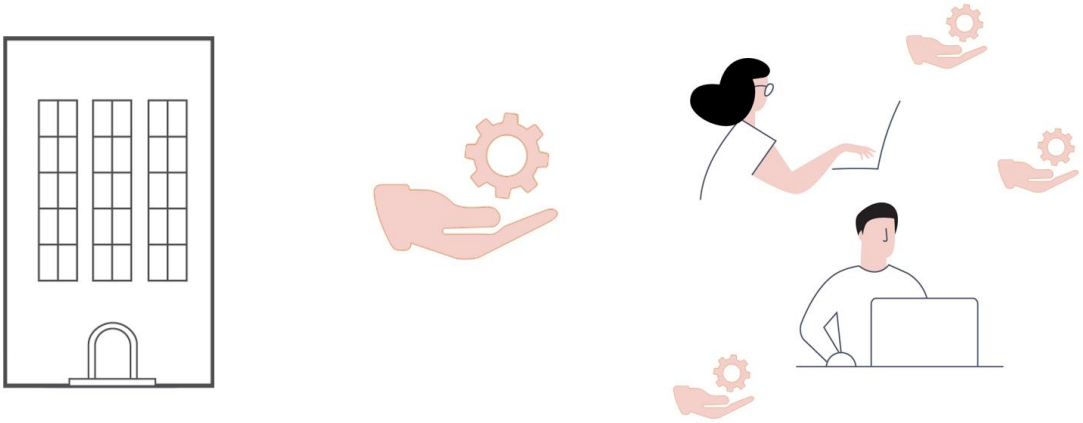


Apresentamos também um caso de uso de subtipos recursivos, onde uma entidade devia autorreferenciar-se:



Tratava-se da realidade em que representávamos a informação dos funcionários da agência de viagens, onde cada funcionário poderia ser, por sua vez, gerente de outro ou outros funcionários.

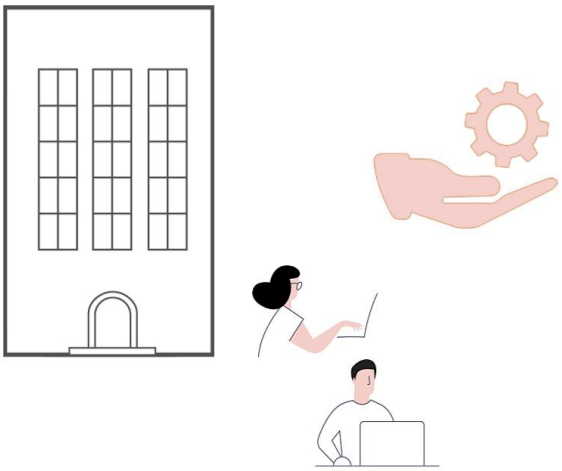
New case: Avoid referential relationship



Vamos estudar agora um último exemplo em que devemos evitar a relação referencial.

Suponhamos que devemos modelar as transações para uma realidade em que existem empresas e serviços que elas podem contratar (como, por exemplo, um serviço de emergência médica). Por sua vez, as empresas têm funcionários que também podem ter contratados serviços que não têm de coincidir com os da empresa para a qual trabalham. Nos interessa registrar esses serviços dos funcionários porque assim, por exemplo, se muitos funcionários contrataram determinado serviço de emergência médica, pode-se tentar um convênio com esse serviço para obter algum benefício.

New case: Avoid referential relationship

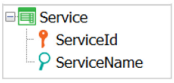


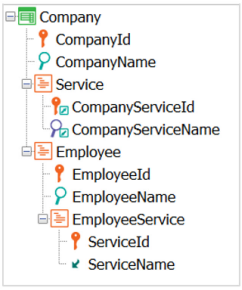
Employees can only work in one company, but they are not to be represented as a strong entity, but as dependent on the company.

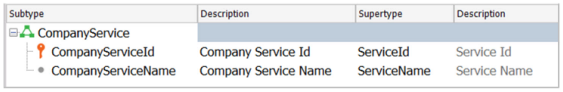
Em nossa realidade, os funcionários só podem trabalhar em uma empresa, mas não querem que sejam representados como uma entidade forte, mas dependente da empresa. Vejamos estas duas soluções propostas, onde uma está correta e a outra não está.

Two possible solutions

Solution A)





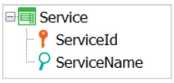


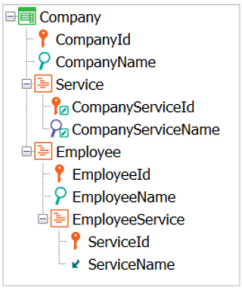
Subtype	Description	Supertype	Description
CompanyService			
CompanyServiceId	Company Service Id	ServiceId	Service Id
CompanyServiceName	Company Service Name	ServiceName	Service Name

A) A primeira solução seria criando estas duas transações e o seguinte grupo de subtipos:

Two possible solutions

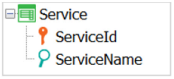
Solution A)

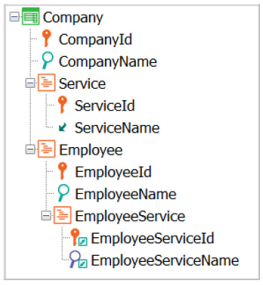




Subtype	Description	Supertype	Description
CompanyService			
CompanyServiceId	Company Service Id	ServiceId	Service Id
CompanyServiceName	Company Service Name	ServiceName	Service Name

Solution B)



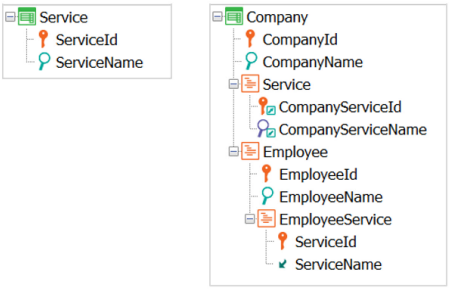


Subtype	Description	Supertype	Description
EmployeeService			
EmployeeServiceId	Employee Service Id	ServiceId	Service Id
EmployeeServiceName	Employee Service Name	ServiceName	Service Name

B) e a segunda solução seria criando estas duas transações e o seguinte grupo de subtipos:

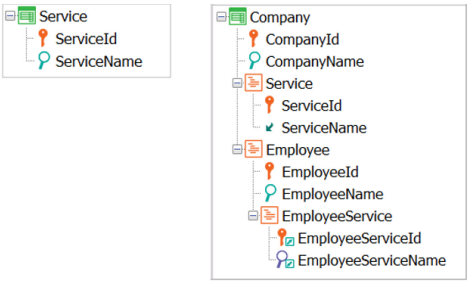
Two possible solutions

Solution A) ✓



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

Solution B) ✗

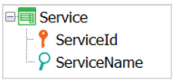


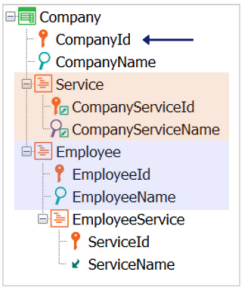
Subtype	Description	Supertype	Description
EmployeeService			
EmployeeServiceId	Employee Service Id	ServiceId	Service Id
EmployeeServiceName	Employee Service Name	ServiceName	Service Name

A solução correta é a A) e não a B).

Two possible solutions

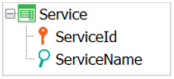
Solution A)

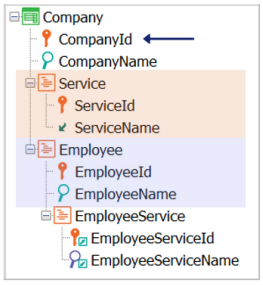




Subtype	Description	Supertype	Description
CompanyService			
CompanyServiceId	Company Service Id	ServiceId	Service Id
CompanyServiceName	Company Service Name	ServiceName	Service Name

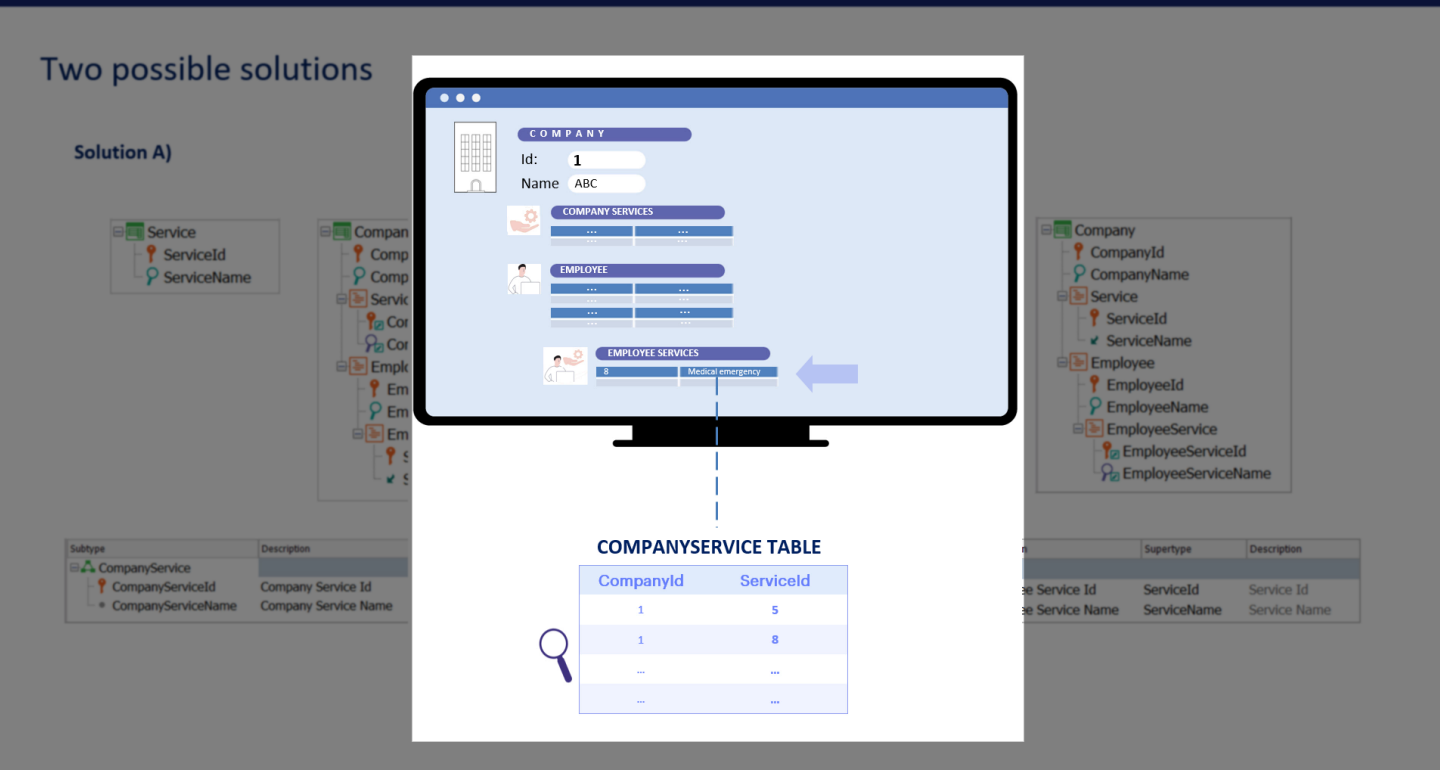
Solution B)





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

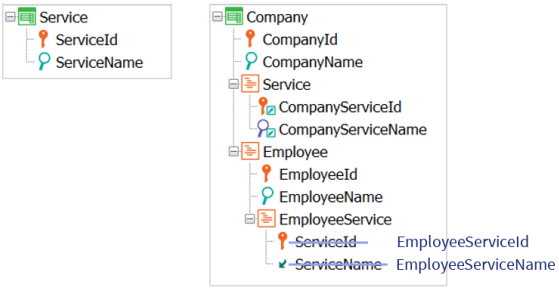
Se observarmos com atenção, na transação Company temos dois níveis paralelos: Service e Employee. Isto significa que tudo o que for inferido de qualquer um desses níveis corresponderá à mesma empresa. No entanto, não queremos que o serviço do funcionário exista como serviço da empresa, pois em nossa realidade o funcionário poderia ter contratado serviços diferentes daqueles da empresa para a qual trabalha.



Em outras palavras: não queremos que seja verificado quando o usuário insere no grid de serviços do funcionário que o serviço inserido exista como registro na tabela correspondente a Company.Service.

Two possible solutions

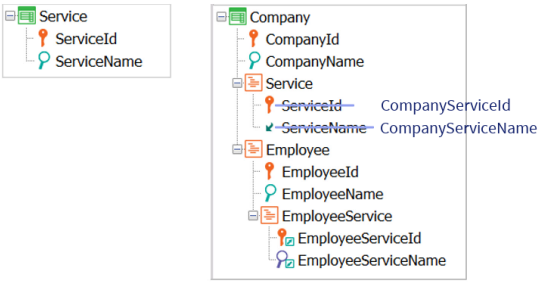
Solution A)



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)

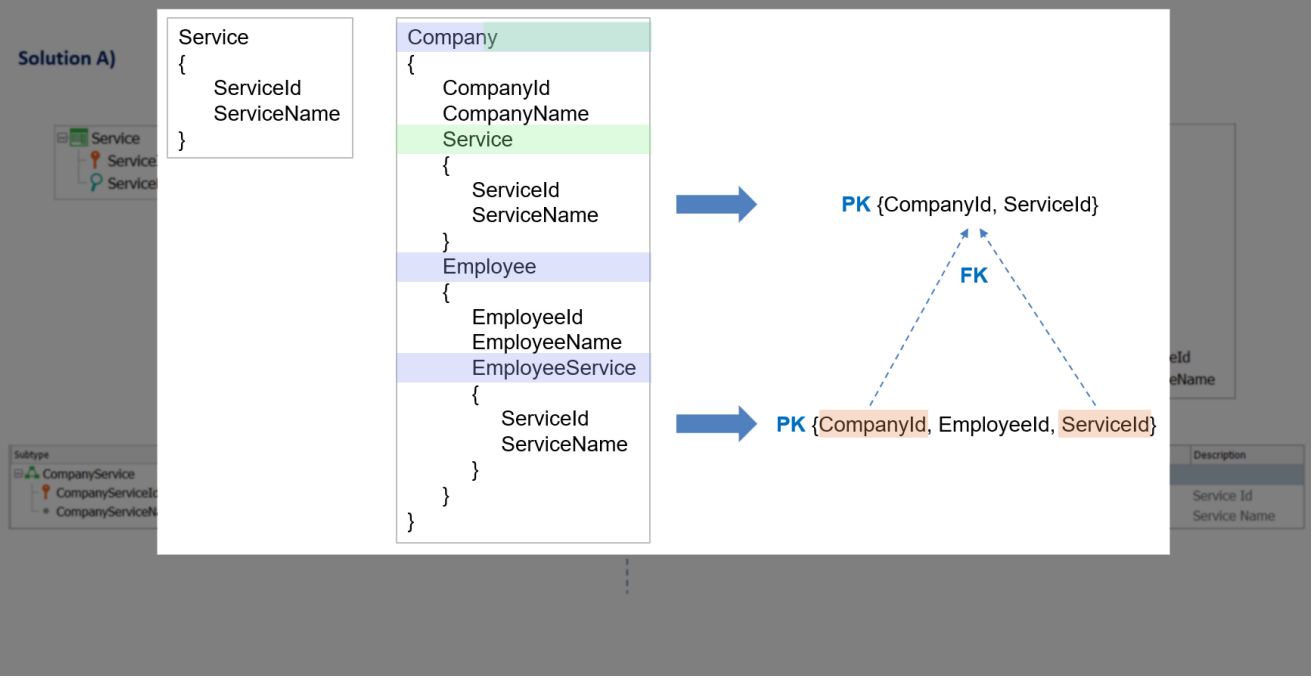


Subtype	Description	Supertype	Description
EmployeeService			
EmployeeServiceId	Employee Service Id	ServiceId	Service Id
EmployeeServiceName	Employee Service Name	ServiceName	Service Name

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

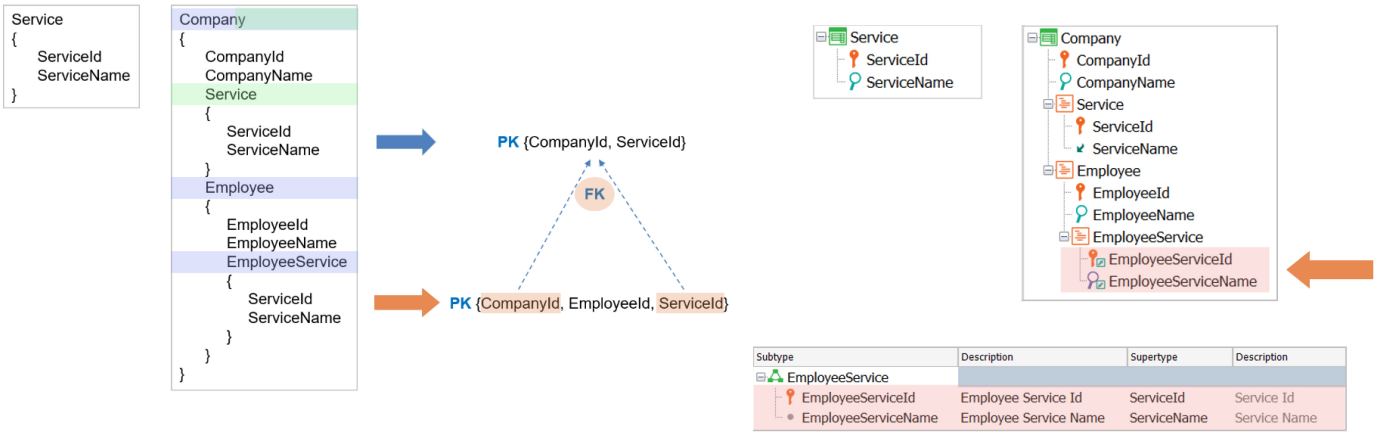
É claro que precisamos definir um grupo de subtipos, já que na mesma transação GeneXus não nos permitirá repetir o mesmo nome de atributo. A pergunta que surge então é: dá no mesmo defini-lo em um nível que no outro? A resposta é não. Poderíamos definir dois grupos de subtipos e acabaríamos com o problema, mas, como já vimos em vídeos anteriores, não é boa prática definir mais subtipos do que os estritamente necessários, pois nunca é exatamente igual ter o subtipo que ter o supertipo, como ficará claro com este exemplo.

Two possible solutions



Portanto, para resolver o problema basta com um único grupo. Por que, então, a solução correta é a A) e não a B)? É que, se GeneXus nos permitisse repetir o mesmo nome de atributo, claramente encontraria que na tabela associada ao nível Company.Employee.EmployeeService, de chave primária {CompanyId, EmployeeId, ServiceId} os atributos {CompanyId, ServiceId} formariam uma chave estrangeira para a tabela correspondente ao nível Company.Service (já que sua chave primária seria {CompanyId, ServiceId}).

Two possible solutions

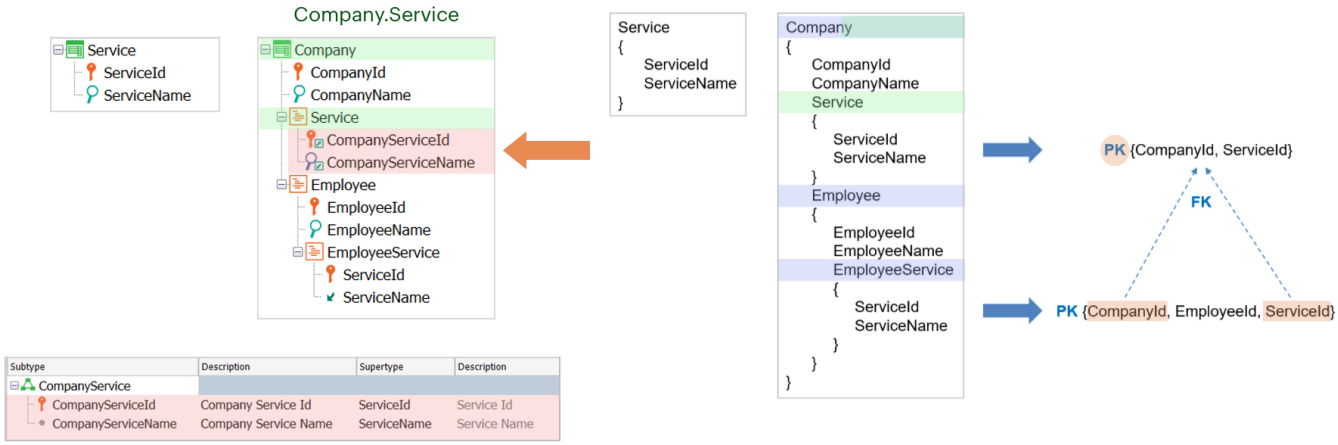


This does not delete for GeneXus its referential function

Mas se o que fazemos é alterar o nome (com um subtipo) de ServiceId na tabela em que este atributo faz parte de uma chave estrangeira, isto não elimina para GeneXus sua função referencial.

Two possible solutions

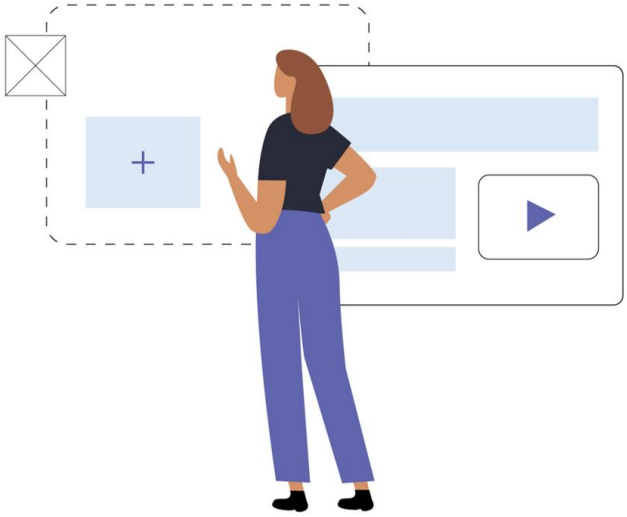
Solution A)



GeneXus does not establish the referential relationship

Em vez disso, se o atributo ao qual mudamos o nome (usando um subtipo) é aquele que cumpre a função de chave primária, então na tabela em que o atributo supertipo aparece, não estabelece a relação referencial.

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



Tanto esses casos mais complexos quanto os mais simples são comuns em aplicações da vida real, e é o desenvolvedor quem deverá analisar os prós e contras das diferentes soluções para encontrar a que melhor se adequa a cada caso em particular, lembrando sempre que os subtipos são uma solução de compromisso para resolver problemas e que devem ser usados com cautela.

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