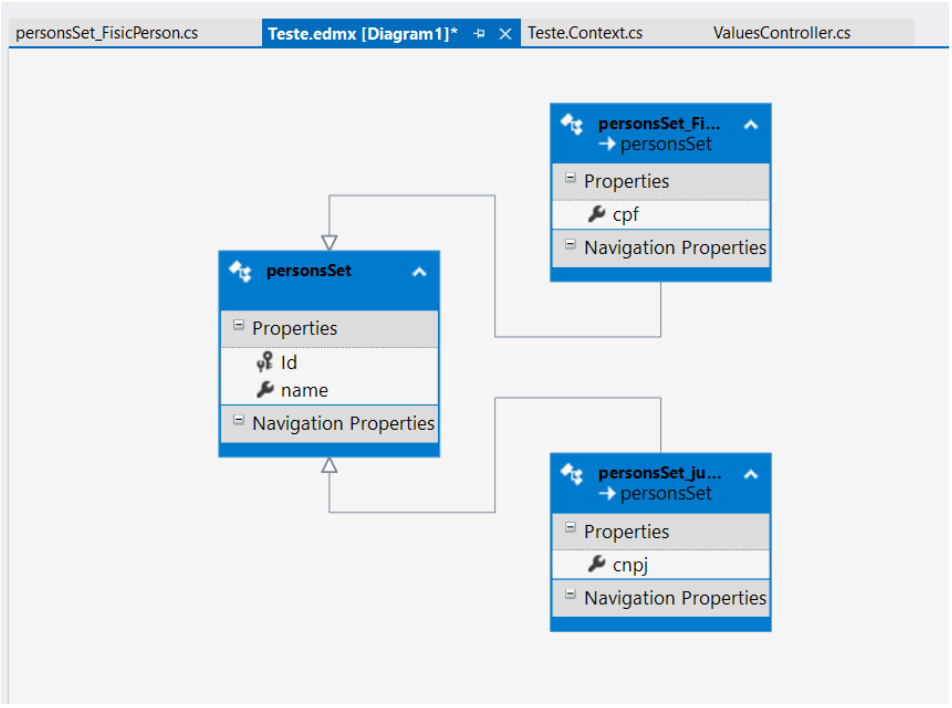
Trabalhando com EDMX com modelo DATABASE FIRST!!

Modelo EDMX já mapeado:



Criando tabelas novas no banco

```
10
11 CREATE TABLE pessoa (
12     Id int IDENTITY(1,1) NOT NULL,
13     name nvarchar(MAX) NOT NULL,
14     CONSTRAINT pk_person PRIMARY KEY (Id)
15 )
16
17 CREATE TABLE pessoa_fisica (
18     cpf nvarchar(MAX) NOT NULL,
19     pessoa_id int NOT NULL,
20     CONSTRAINT pessoafisicafkpessoa FOREIGN KEY (pessoa_id) REFERENCES pessoa(Id) ON DELETE CASCADE
21 )
22
23 CREATE TABLE pessoa_juridica (
24     cnpj nvarchar(MAX) NOT NULL,
25     pessoa_id int NOT NULL,
26     CONSTRAINT pessoajuridicafkpessoa FOREIGN KEY (pessoa_id) REFERENCES pessoa(Id) ON DELETE CASCADE
27 )
28
```

Incluindo uma coluna **na tabela já existente** e previamente mapeada no EDMX:

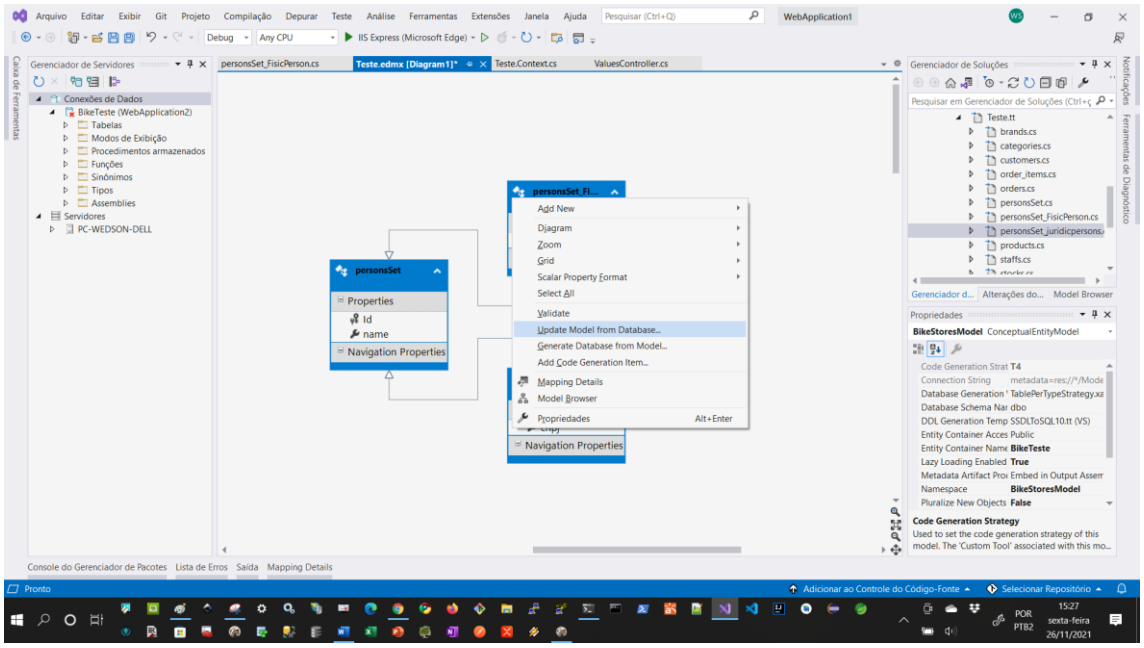
```
28
29
30
31 alter table personsSet ADD description VARCHAR (255);
```

Statistics

alter table personsSet ADD description VARCHAR (Enter a SQL expression to filter results (us

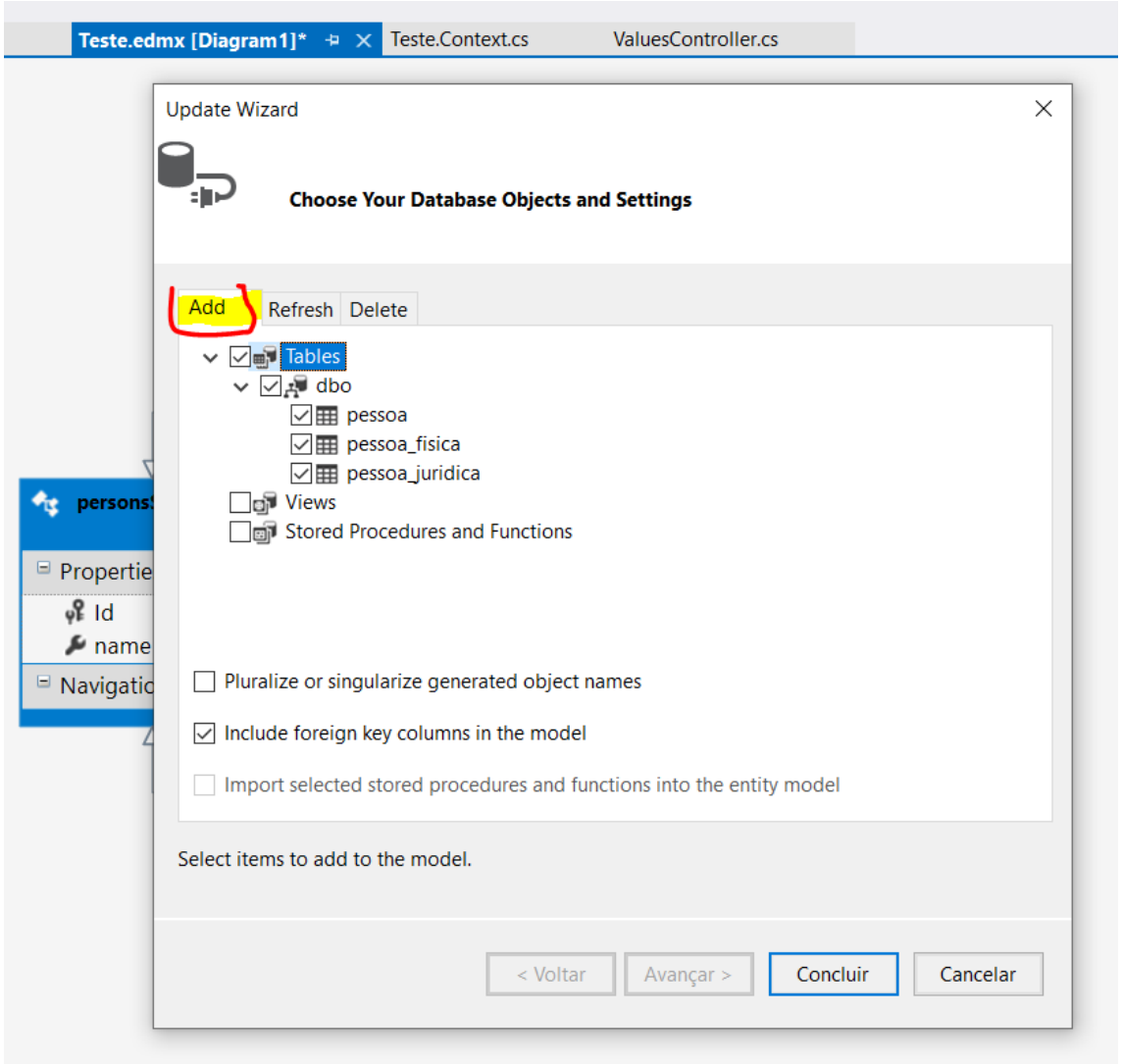
me	Value
dated Rows	0
ery	alter table personsSet ADD description VARCHAR (255);
ish time	Fri Nov 26 15:23:01 BRT 2021

Para importar as novas tabelas criadas bem como as alterações em tabelas pré-existentes, é necessário atualizar o modelo a partir do banco de dados:

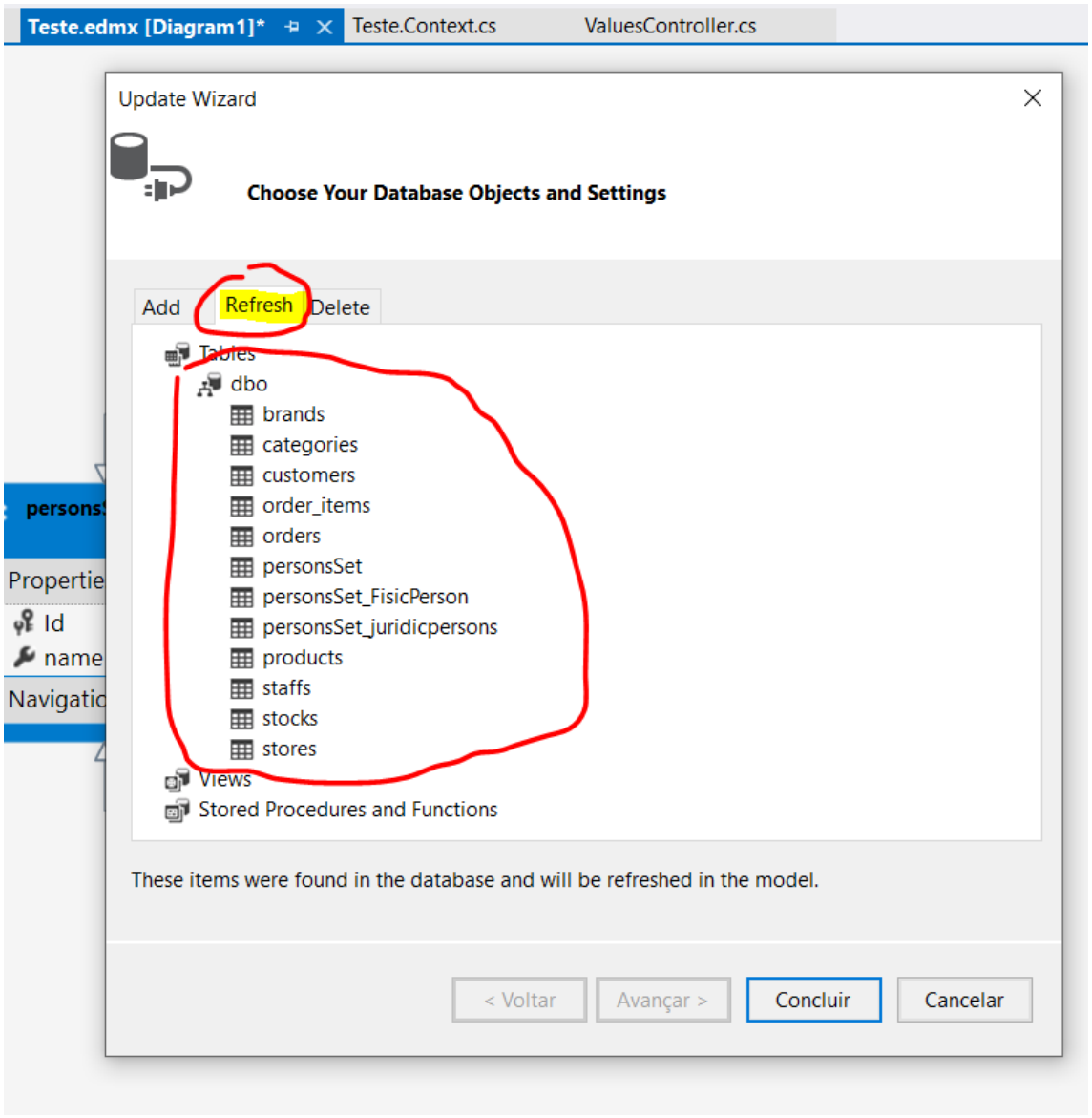


Menu de contexto acionado a partir do botão direito do mouse

Na janela exibida, na guia “Add”, marcar todas as tabelas que deseja importar para o modelo:

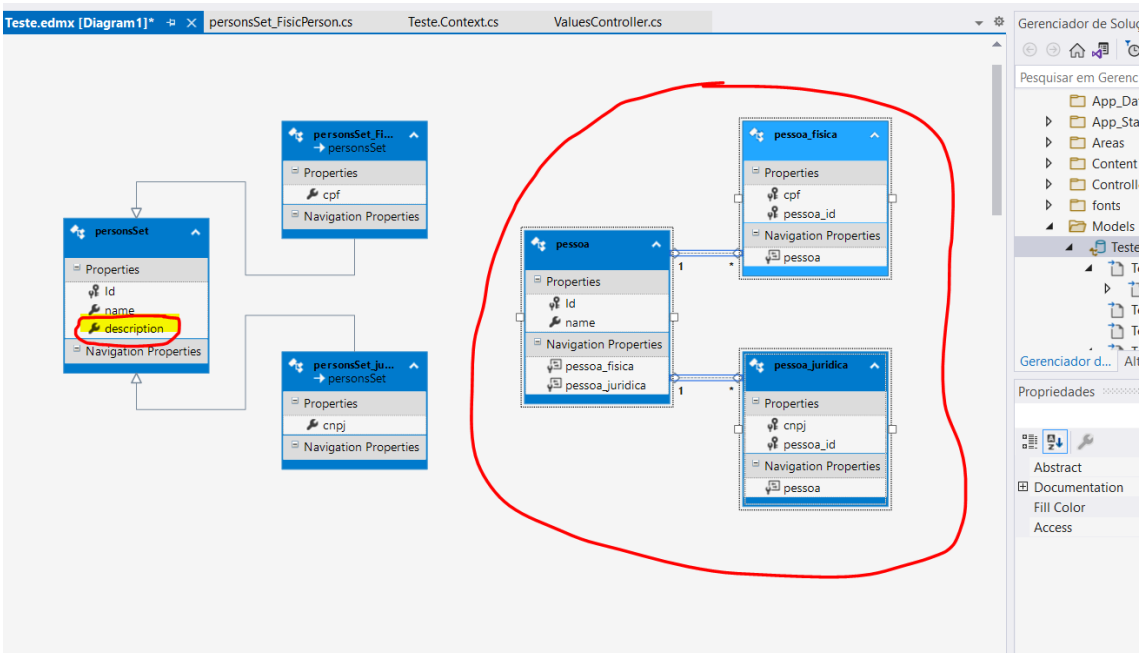


A guia “Refresh” descreve os objetos do banco de dados já mapeados e que, se porventura sofram alguma modificação no banco, eles serão atualizados no modelo:



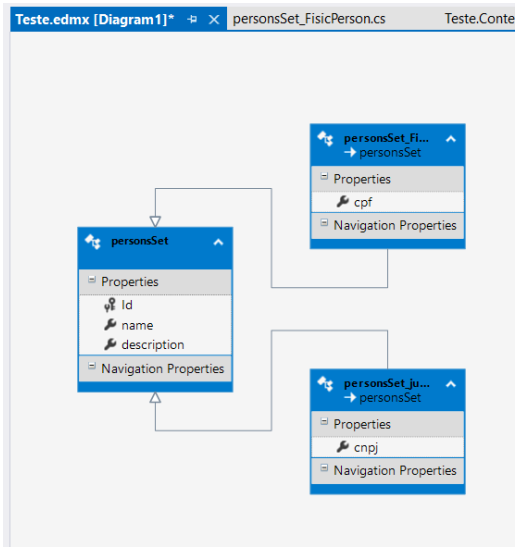
Clique em “Concluir” para iniciar o processo de atualização.

Como resultado da importação, o EDMX foi atualizado e o seguinte modelo foi apresentado:



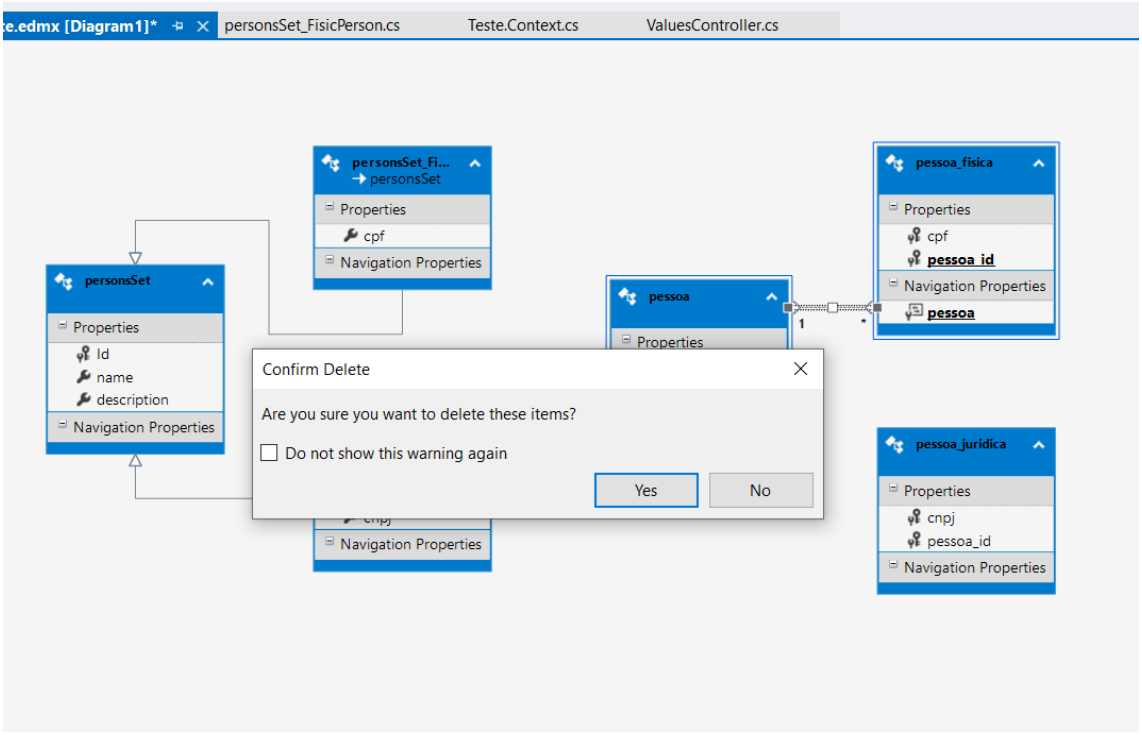
Note que as novas tabelas foram importadas e o novo campo exibido no model pré-existente.

*Veja também que o relacionamento do tipo herança entre as tabelas antigas que **já estava ajustados em importação anterior, foram mantidos**!*

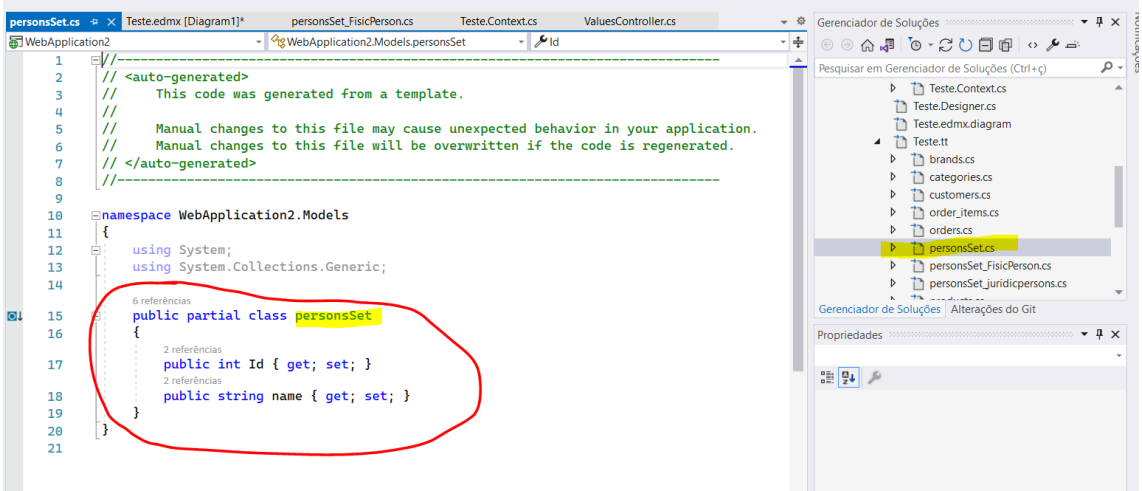


O importador colocou as tabelas pessoas, pessoa_fisica e pessoa_juridica com relacionamentos normais, porém, queremos que esta relação seja representada por meio de herança (modelo TPT) no modelo. Para isso, devemos excluir manualmente este relacionamento e dizer ao EF que ele se dá através de herança.

Selecione a linha do relacionamento e pressione “delete”. Confirme a operação de exclusão no diálogo que vai abrir:



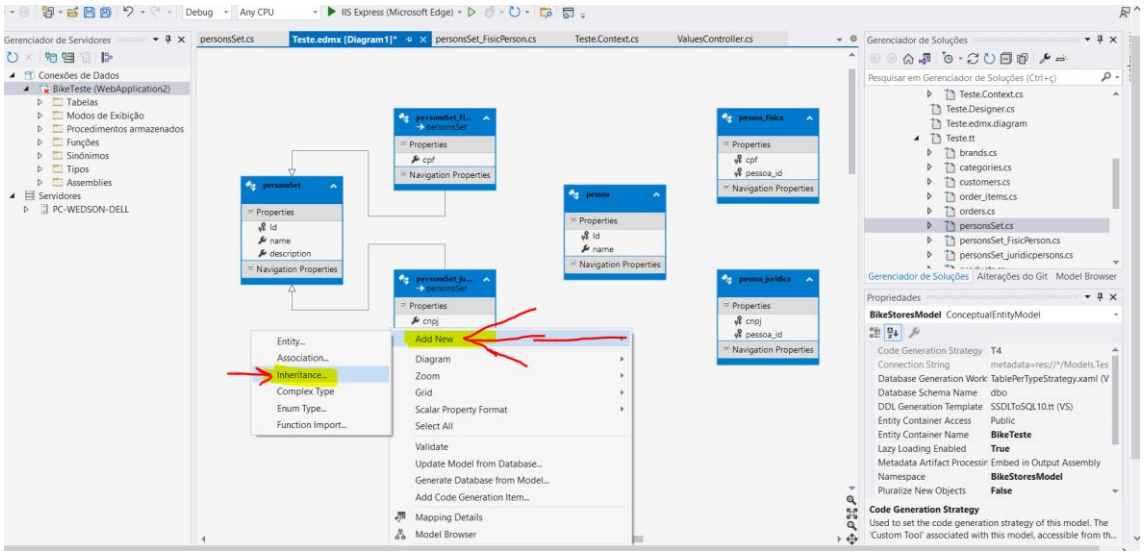
Neste primeiro momento, as entidades geradas não estão conforme representação do modelo, elas serão atualizadas após “Validar” e “Salvar” o modelo EDMX.



Veja na imagem acima que tanto a herança quanto o novo atributo acrescentado à tabela não estão no código fonte.

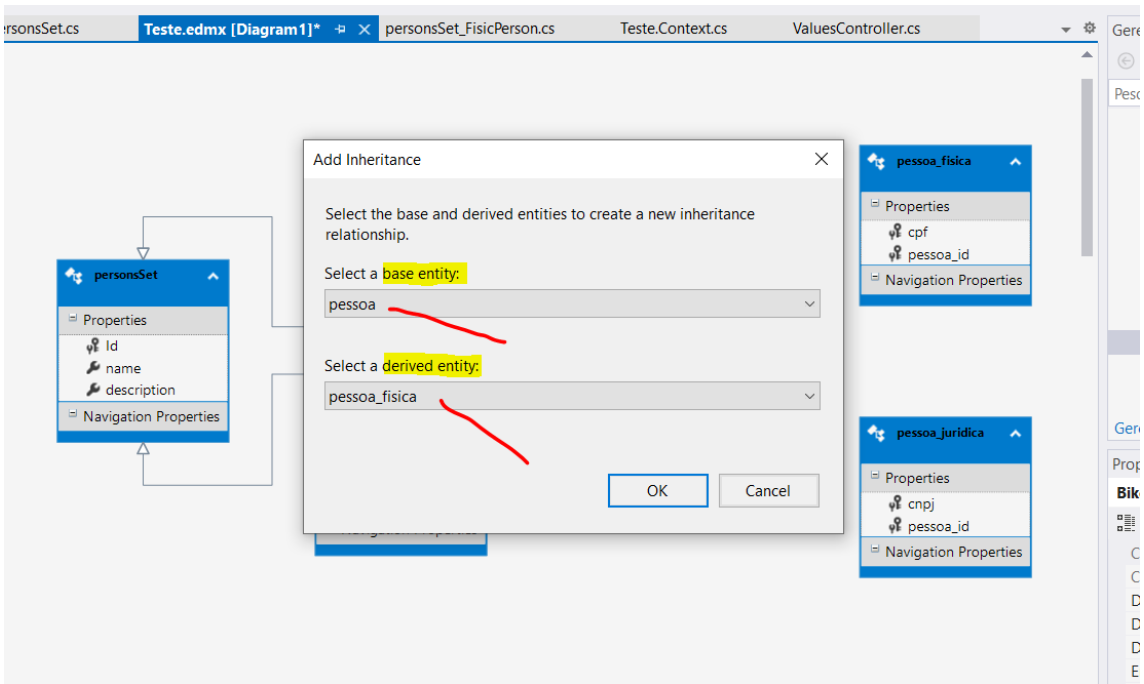
Agora vamos fazer as modificações no relacionamento das tabelas novas que foram importadas. Mais uma vez, o mapeamento no modelo EDMX das tabelas antigas que já haviam sido modificadas em commits anteriores, não sofreram alterações com esta atualização (Update from database), elas foram mantidas e o que precisamos é alterar somente as novas tabelas.

Para isso, clique com o botão direito do mouse em qualquer área em branco no modelo e escolha as ações “Add new -> Inheritance...”:

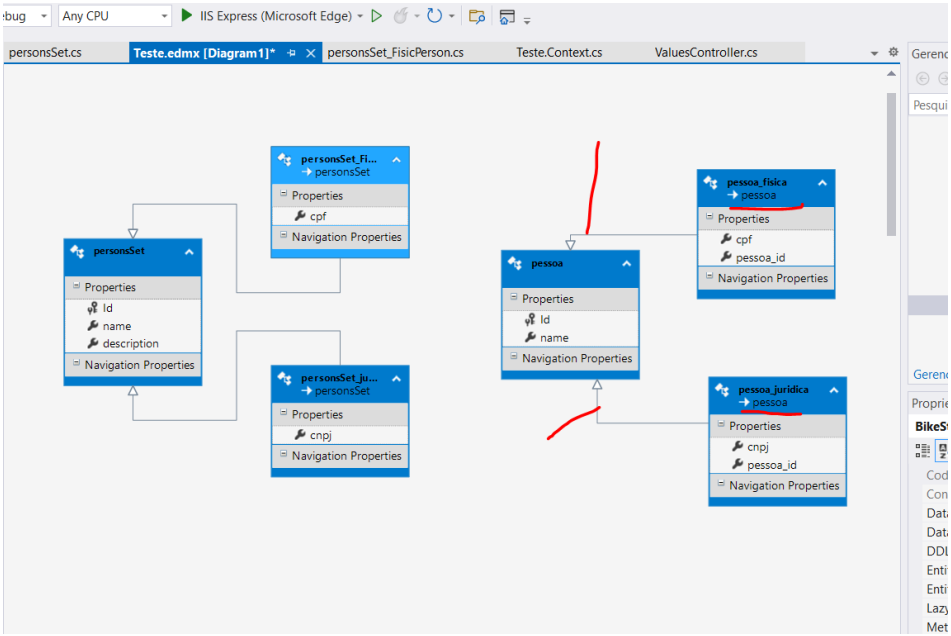


Defina a herança da entidade “pessoa” para “pessoa_fisica”.

Neste nosso caso, Pessoa é a superclasse e Pessoa_Fisica a classe derivada:



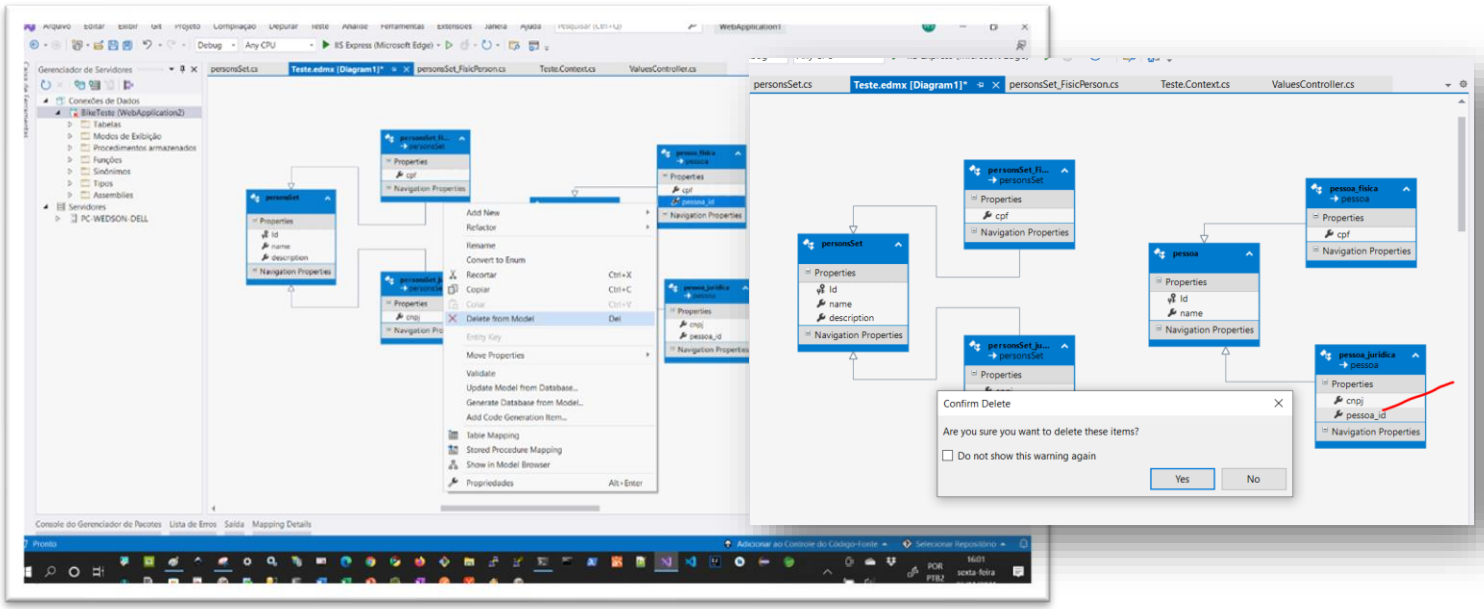
Repita o mesmo procedimento e coloque a relação para Pessoa_Juridica como derivada de pessoa. Ao término, você terá um modelo semelhante a este:



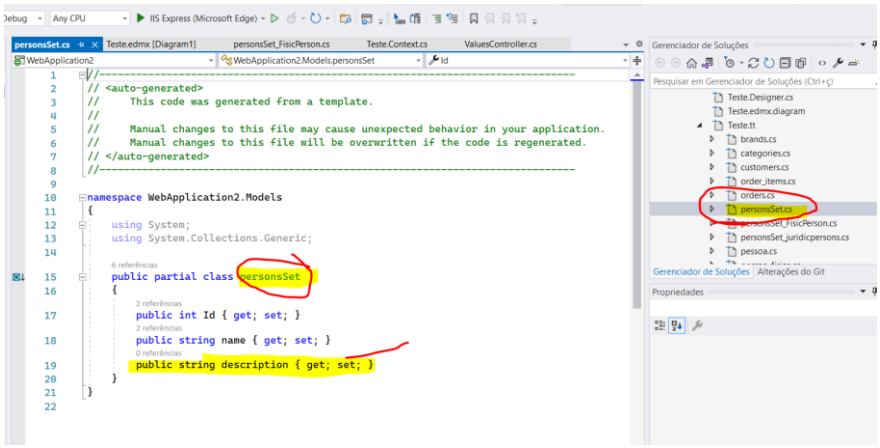
Note que o EDMX mostra qual é a entidade base no topo da própria entidade.

Agora, precisamos excluir o mapeamento dos atributos das classes derivadas que apontam para a chave (primary key) da classe base. Este mapeamento deve ser removido para que o EF saiba qual entidade deve ser persistida primeiro no banco de dados e também para que ele propague o ID da classe base para a classe filha quando for realizar a operação no banco.

Para isso, selecione o atributo no modelo EDMX e pressione “delete” .

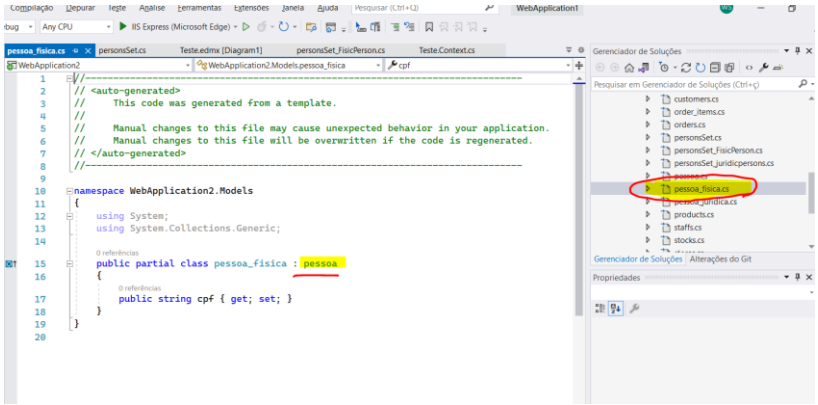


Após realizar estas mudanças, salve as alterações no EDMX. Neste momento, as classes de entidade serão alteradas para representar as alterações efetuadas no modelo:

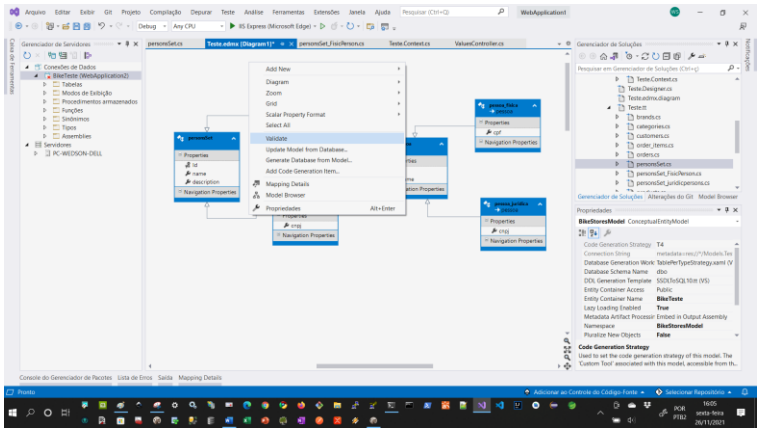


Veja que só após salvar é que o atributo novo foi acrescentado à entidade que já existia antes da atualização.

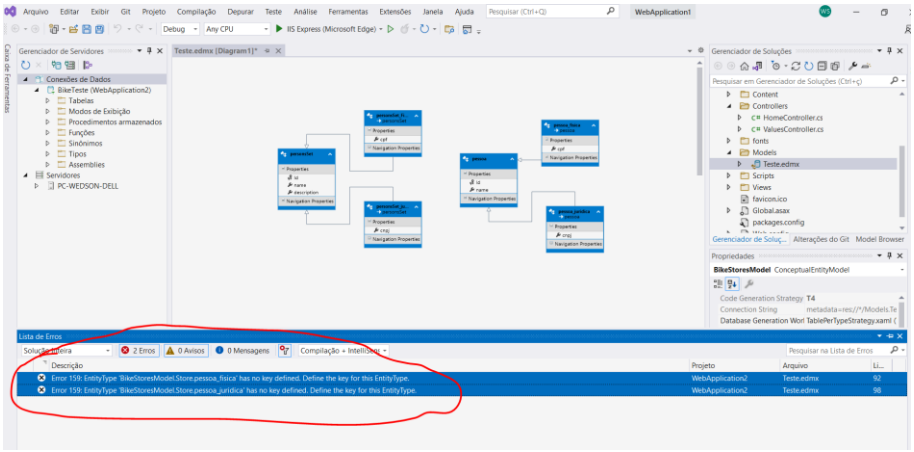
Bem como só após salvar que a herança foi acrescentadas às entidades pessoa_fisica e pessoa_juridica:



Agora, com o botão direito numa área em branco do modelo, acione o menu de contexto e pressione a opção “Validate”:



Veja que o resultado foi que ele reclama que não há uma chave primária para as entidades modeladas:



Atenção a este erro comum de modelagem.

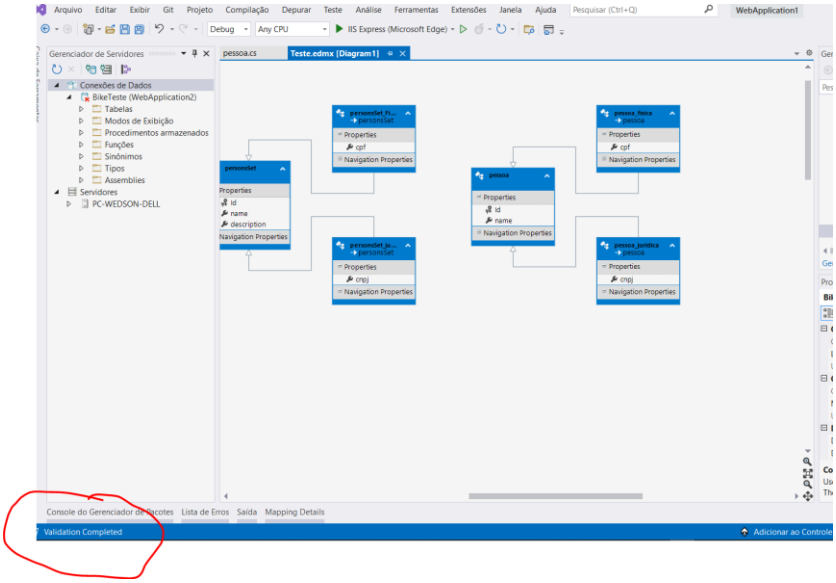
Isso ocorre porque não definimos um atributo identificador para as tabelas mapeadas pelas classes derivadas: pessoa_fisica e pessoa_juridica.

Tenha o cuidado de colocar o nome do atributo na tabela igual ao nome do atributo chave primária na tabela base!

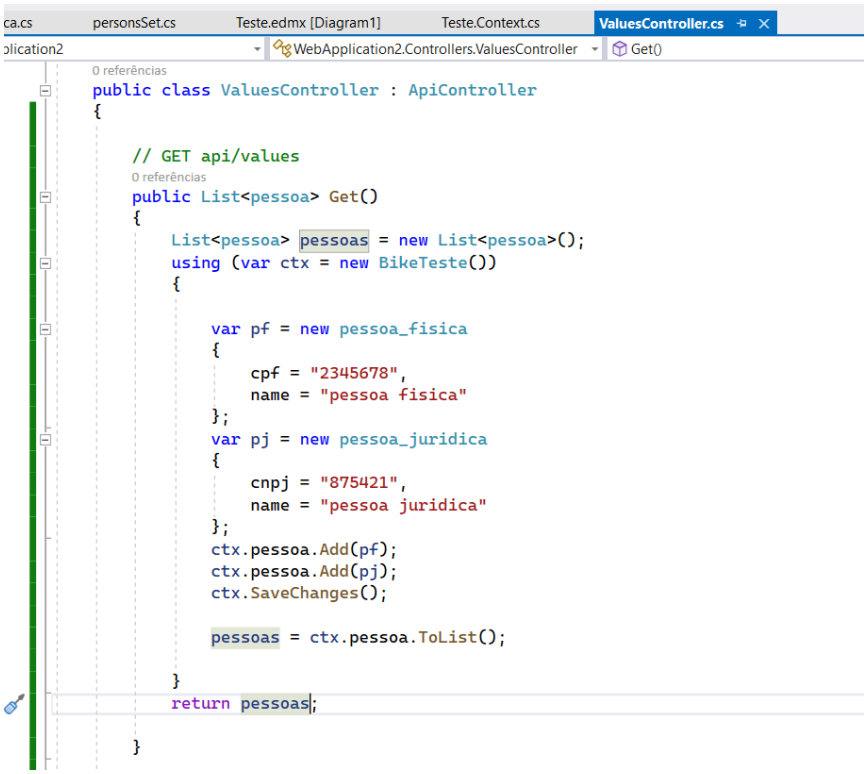
Para solucioná-lo, basta incluir, primeiro no banco de dados a primary key para estas duas tabelas e realizar a atualização novamente do modelo.

```
CREATE TABLE pessoa (
    Id int IDENTITY(1,1) NOT NULL,
    name nvarchar(MAX) NOT NULL,
    CONSTRAINT pk_person PRIMARY KEY (Id)
)


CREATE TABLE BikeStores.dbo.pessoa_fisica (
    cpf nvarchar(MAX) NOT NULL,
    Id int NOT NULL,
    CONSTRAINT pessoa_fisica_PK PRIMARY KEY (Id),
    CONSTRAINT pessoafisicafkpessoa FOREIGN KEY (Id) REFERENCES BikeStores.dbo.pessoa(Id) ON DELETE CASCADE
)


CREATE TABLE BikeStores.dbo.pessoa_juridica (
    cnpj nvarchar(MAX) NOT NULL,
    Id int NOT NULL,
    CONSTRAINT pessoa_juridica_PK PRIMARY KEY (Id),
    CONSTRAINT pessoajuridicafkpessoa FOREIGN KEY (Id) REFERENCES BikeStores.dbo.pessoa(Id) ON DELETE CASCADE
)
```

Se por acaso os nomes das chaves estrangeiras forem diferentes do nome da chave primária na tabela base, é preciso ajustar no banco, excluir as classes derivadas do modelo EDMX e repetir o processo "Update from database", removendo as associações e os atributos do modelo importado e incluindo novamente a relação de herança no modelo, salvando e validando o modelo. Após efetuar estes passos, o seu modelo estará válido!



Por fim, veja um exemplo de uso no controller (apenas um exemplo). Ele adiciona dois registros no banco de dados e depois consulta e retorna para quem chamou a API:



Resultado no banco: