Passando um ER para Modelo Relacional

Apenas relembrando, **ER (Entidade relacional)** é um tipo de representação que **ainda não é um modelo próximo ao banco de dados** e sim um modelo que possibilite ao cliente enxergar como o negócio funciona.

Vamos só abordar algumas regras que ajudam na hora de adaptar um ER para um Modelo Relacional.

Regras

Entidade forte: Uma entidade forte é uma entidade que não depende de nenhuma outra para existir. Devemos criar a **Relação(tabela)** com todos atributos simples que vem da entidade, devemos escolher um dos seus atributos para ser o atributo chave ***(aquele que não se repete entre registros dessa entidade)***.

Um ponto **importante** aqui é que se um atributo é composto, então devemos considerar os componentes dessa composição como atributos da Relação. Por exemplo, lá no **ER tem um atributo que é nome**, e ele é **composto por nome do meio e ultimo nome**. Então minha Relação vai ter os campos: ***nome, nome_meio e ultimo_nome.***

Entidade fraca: Uma entidade fraca é uma entidade que DEPENDE de outra para existir. Para passar esse tipo de Entidade para um modelo relacional, primeiro precisamos saber de quem ela depende.

Então minha **entidade fraca = F** pertente a minha **entidade forte = T**, sabendo disso, agora eu crio uma **Relação(tabela)** onde eu possuo as **chaves primárias** de F e T como chaves estrangeiras da **Relação** além de todos os atributos de F.

No final minha **Relação** vai ser um mapeamento de F com a adição das **primary key** de T como **chave estrangeira** para que haja o relacionamento.

Relacionamento 1:1 binários: Em uma **relação um para um binário (Duas entidades envolvidas)**, eu posso escolher qual entidade irá conter os atributos da relação. Por exemplo, um relacionamento um para um de **Funcionario** que **GERENCIA** um **Departamento,** qualquer uma das minhas duas entidades envolvidas pode ter os campos que dizem respeito ao relacionamento de **GERENCIA.**

GERENCIA tem o atributo de ***Data_inicio_gerente***, e a relação de **Um Funcionario(gerente) para Um Departamento**.

Posso então adicionar minha **chave estrangeira de funcionário(gerente)** e ***data_inicio_gerente*** como a**tributo da minha Relação(tabela) Departamento**.

O inverso também funcionaria, adicionar como **atributo da Relação(tabela) Funcionário** a **chave estrangeira de departamento** e o atributo ***data_inicio_gerente***.

IMPORTANTE: por mais que a teoria diga que pode ser em ambas as entidades o relacionamento, ainda É NECESSÁRIO pensar onde vai se encaixar melhor os atributos.

Relacionamento 1:N binários: A relação de **1:N** é dada da seguinte forma: Identificamos o **lado da relação que representa muitos** e então colocamos a **chave estrangeira da relação que representa o lado 1**, juntamente com os atributos próprios da relação.

Pra ficar mais claro: meu **DEPARTAMENTO CONTROLA PROJETO**, um departamento pode controlar vários projetos, e vários projetos podem ser controlados por um departamento, essa é nossa situação 1:N.

Desse modo p**odemos identificar o PROJETO** como sendo o **lado N *(já que UM departamento pode ter vários projetos)*** então meu lado **N(Projeto)** vai ter os atributos da **Relação (Controla)** juntamente com as **chaves estrangeiras que vem de Departamento.**

Relação 1:N recursiva: *Um funcionário é supervisor de outros funcionários, então além da minha chave primária de funcionário (CPF) eu preciso de um campo que faça essa relação recursiva na tabela de funcionário, então eu crio* ***CPF_FUNCIONARIO.***

Relacionamento N:M binários: Quando existe um relacionamento de **N:M (*Muitos para muitos*)** temos que **CRIAR UMA NOVA RELAÇÃO(Tabela)** que irá conter a chave estrangeira de ambas as relações mais seus atributos próprios caso haja algum.

Por exemplo: *vários(N)* **Funcionário** podem **TRABALHAR** em *vários(M)* **Projetos,** portanto, trabalhar vai virar uma nova relação (**TRABALHA_EM),** e essa nova relação vai ter minhas ***chaves estrangeiras que vem de Funcionário e Projeto***, além de um atributo próprio chamado horas.

Atributos multivalorados: Em todos os relacionamentos até agora esquecemos a questão do **atributo multivalorado**, então vamos ver a regra dele agora.

Então quando uma **entidade = T** tem atributos **multivalorados,** então na **Relação(tabela**) vamos ter a **chave estrangeira de T** e o valor do atributo **multivalorado que também vai ser uma chave da Relação.**

Por exemplo: Um **Departamento** tem a **LOCALIZAÇÃO** como sendo um atributo **multivalorado.** Então agora eu vou ter uma **Relação(Localizacoes_Departamento)** com os atributos **sendo a chave estrangeira de Departamento** mais um **ATRIBUTO QUE REPRESENTA O PRÓPRIO VALOR DO ATRIBUTO MULTIVALORADO.**

Generalização/Especialização: Esse conceito diz que um atributo só vai existir dependendo de outro.

Por exemplo, um **Funcionário** tem o atributo **Tipo**, esse atributo pode ter o valor de ***"contratado" ou "terceiro"***. Dependendo desse resultado eu vou ter atributos diferentes, como o contratado vai ter o **número da carteira** e o terceiro vai ter o **cnpj da empresa dele.**

Na regra desse cara o que vai acontecer é o seguinte, eu vou ter novas relações baseadas nesses atributos de generalização**.** Onde essas relações vão ter como chave primária as chaves estrangeiras da relação já pré-estabelecida além é claro dos seus próprios atributos.