MER e DER

MER e DER são dois tipos de modelos de dados usados na área de banco de dados para representar a estrutura e as relações entre os dados de um sistema.

MER, ou Modelo de Entidade-Relacionamento, é um modelo conceitual que descreve as entidades (objetos) de um sistema e as relações entre elas. Em um MER, as entidades são representadas por retângulos e as relações são representadas por linhas que conectam esses retângulos. Cada relação pode ter um tipo, como "um para muitos", "muitos para muitos" ou "um para um". O MER é uma representação abstrata e não está ligado a um sistema específico de banco de dados.

Já o DER, ou Diagrama de Entidade-Relacionamento, é uma versão mais detalhada do MER que mostra como as entidades e relações são implementadas em um sistema de banco de dados específico. O DER inclui detalhes como os tipos de dados para os campos, as chaves primárias e estrangeiras e outras restrições. Em um DER, as tabelas do banco de dados são representadas por retângulos e as relações entre elas são representadas por linhas com setas que indicam a direção da relação.

Modelo de Entidade e Relacionamento (MER)

Um projeto de banco de dados não se resume somente a código SQL, muitas vezes precisamos fazer um desenho que o defina ou um esboço de qual será a estrutura que desejamos. Esse processo chamamos de modelagem, e podemos partir para essa etapa quando já temos as regras de negócio definidas, e a partir dela vamos extrair as informações que condizem com o que nosso banco de dados armazenará.

Basicamente podemos dizer que o Modelo de Entidade e Relacionamento (MER) são diagramas utilizados para projetar bancos de dados relacionais, utilizando como base a relação de objetos reais, e sendo representado por meio de entidades e relacionamentos.

É possível usar o MER para ilustrar como os dados são estruturados nos processos de negócios ou para detalhar como os dados são armazenados nos bancos de dados relacionais.

O que são entidades?

Entidades são formas gráficas que representam objetos do mundo real e que possuem existência independente, podemos dizer que representam por exemplo **pessoas**, **empresas**, **automóveis**, **casa**, **departamentos**, **produtos** e muitas outras coisas.

De forma simples podemos dizer que existem três tipos de entidades, as entidades fortes, que são independentes, as entidades fracas que dependem de outras entidades para existir, ou seja, podemos dizer que elas dependem de uma entidade forte para existir. E por último temos as

entidades associativas, que são utilizadas quando precisamos associar uma entidade a um relacionamento.

Podemos ir um pouco além e citar alguns exemplos para cada uma:

Entidades fortes: são entidades que não dependem de outras entidades para existirem. Um exemplo seria em um sistema de recursos humanos (RH), onde a entidade **cargos**, por exemplo, independe de quaisquer outras para existir.

Entidades fracas: as entidades fracas seguem o lado oposto das fortes, elas **dependem** de outras entidades para existirem. Utilizando o exemplo anterior do sistema de recursos humanos, uma entidade de colaborador dependeria da entidade cargo, pois um colaborador sem cargo não faria sentido.

Entidades associativas: as entidades associativas são utilizadas quando temos a necessidade de associar o relacionamento entre duas ou mais entidades, neste caso um exemplo que podemos utilizar é com base no sistema de recursos humanos anteriores, se pensarmos que temos uma entidade chamada **cargo** e uma entidade chamada **área** que pode ter valores como gestão, tecnologia da informação, recrutador etc. Podemos pensar no seguinte cenário: Um **cargo** pode estar em várias **áreas** ao mesmo tempo, por exemplo, um cargo de **gestor** pode administrar várias áreas, da mesma forma que várias áreas são administradas pelo cargo de gestor, mais um exemplo seria no caso de um recrutador que pode recrutar para várias áreas, ao mesmo tempo que várias áreas têm recrutamento pelo cargo recrutador, neste caso estabelecemos um relacionamento que chamamos de muitos para muitos que veremos mais especificamente no tópico de **Relacionamentos** logo abaixo. Levando em consideração estes exemplos, devemos criar uma entidade associativa que terá como atributo as chaves primárias das tabelas que fazem parte do relacionamento.

O que são atributos?

Atributos são representações de características de uma entidade, podemos utilizar como exemplo os atributos definidos para as entidades abaixo:

Entidade	Atributos
Pessoas	Nome, CPF, Endereço, Telefone, E-mail, Estado Civil, Data de Nascimento
Produtos	Descrição do Produto, Dimensões, Cor, Categoria
Automóveis	Nome do automóvel, Placa, Cor, Chassi

Os atributos podem ser definidos em algumas categorias, e elas são:

Atributos Simples: são atributos indivisíveis, ou seja, é composto por uma característica que não tem como dividir/destrinchar em outros atributos, um exemplo seria o atributo RG, ele não pode ser dividido em partes menores para formar outros atributos, o que o torna indivisível;

Atributos Compostos: são atributos que podem ser divididos em uma ou mais partes que o representam, como o atributo Endereço, ele pode ser subdividido em atributos menores, como, por exemplo, cidade, estado, rua, cep, número e bairro. Também temos o atributo Nome na entidade de pessoas, pois podemos dividir ele em dois atributos menores que seriam Primeiro Nome e Sobrenome;

Atributos Monovalorados: são atributos que possuem um único valor para a entidade, como por exemplo, o campo nome relacionado a entidade automóvel;

Atributos Multivalorados: são atributos que possuem mais de um valor. Por exemplo, o caso da categoria associada à entidade produtos, pois é possível não ter nenhuma categoria ou ter várias.

Atributos Referenciais: são atributos que representam a ligação de uma entidade com outra em um relacionamento e são chamados de Chave Estrangeira (FK(Foreign Key)), ou seja, eles são ligados a chave primária de outra entidade. Um exemplo seria em um banco de dados de e-commerce onde temos uma entidade categoria que tem um atributo que armazenará o código de identificação dela no sistema que é sua chave primária (PK), e uma entidade de produto que possui uma categoria, então temos o relacionamento da entidade de produto com a de categoria.

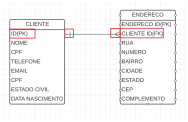
Existem alguns atributos que representam valores únicos que identificam a entidade, podemos utilizar como exemplo um cadastro de clientes, onde temos um atributo que é o CPF, um CPF é um documento único que representa cada cliente, então chamamos ele de **chave primária** ou utilizamos a sigla **PK(Primary Key)** para identificá-lo.

Relacionamentos

A partir do momento que as entidades são identificadas, devemos definir como será o relacionamento entre elas. Os tipos de relacionamentos entre as entidades são três:

Relacionamento 1 para 1 (1:1): é quando temos duas entidades envolvidas e elas referenciam obrigatoriamente apenas uma a outra. Um exemplo seria em um banco de dados de gestão de clientes, e cada cliente tem seu **único** endereço, assim como **um** endereço pertence somente a um cliente.

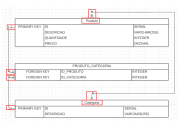
Relacionamento 1 para muitos (1:N): é quando uma das entidades envolvidas, que vamos chamar de X pode referenciar várias unidades da outra que podemos nomear de Y, porém, a outra entidade Y só pode referenciar uma unidade da entidade X. Um exemplo seria em um banco de dados de gestão de clientes, onde cada cliente pode ter vários endereços, assim como vários endereços pertencem a um único cliente.



Fonte: Autoria própria.

Observe que a entidade cliente tem a parte da relação com a ramificação simples que é somente um traço, e a entidade endereço além de ter um campo chamado **cliente id** referenciando a *primary key* da entidade de cliente, contém uma ramificação com vários traços, o que identifica que essa entidade recebe a parte N do relacionamento.

Relacionamento muitos para muitos (N:N): é quando cada entidade, de ambos os lados, podem referenciar múltiplas unidades da outra. Por exemplo em um sistema de e-commerce, um produto pode ter várias categorias, assim como várias categorias contém vários produtos.



Fonte: Autoria própria.

Observe que entre a entidade categoria e produto ambos recebem a ramificação N, ou seja, ambas guardam N informações uma da outra, no caso N produtos podem ter várias categorias, assim como N categorias podem ter vários produtos.

Neste caso é gerado uma entidade intermediária que vai ser nomeada com a combinação dos nomes das duas entidades, no caso produto_categoria, e nesta nova entidade teremos 2 campos de chaves estrangeiras, herdadas das chaves primárias das duas entidades garantindo assim o nosso relacionamento N-N.

DER (Diagrama Entidade e Relacionamento)

O diagrama de entidade e relacionamento é a representação gráfica do MER, porque na maioria dos casos somente o modelo conceitual é abstrato demais para auxiliar no desenvolvimento.

O diagrama auxilia bastante na comunicação entre as equipes, pois oferece uma linguagem comum que é utilizada tanto pelo analista que é responsável por levantar os requisitos, quanto pelos desenvolvedores que são responsáveis por implementar tudo que foi modelado.

Além de muitas vezes auxiliar no desenvolvimento de aplicações com uma estrutura definida do banco de dados que não precisará ficar sendo ajustada diversas vezes em tempo de desenvolvimento.

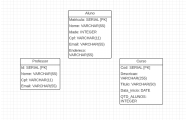
Em uma notação mais atual da UML no diagrama entidade-relacionamento, as entidades devem ser representadas por retângulo, e neste retângulo teremos os atributos da entidade, e os relacionamentos entre elas.

Praticando

Agora que já sabemos o que é um MER e o que é um DER, podemos praticar criando um cenário fictício onde fomos contratados para criar um sistema para uma escola de programação, e este sistema será para gestão de cursos. Bom, sabemos que para gerenciar os cursos, precisaremos de alunos e professores também, e para cada curso armazenaremos as informações sobre: descrição, título, quantidade de alunos, professor e data de início. Para os alunos precisaremos armazenar informações como nome, idade, e-mail, CPF e endereço. Para os professores precisaremos armazenar informações como nome, CPF e e-mail. O que devemos fazer como primeiro passo é identificar as entidades e seus atributos, assim teremos:

Entidade **Professor**: tem os atributos de nome, CPF e e-mail;
Entidade **Aluno**: tem os atributos de nome, idade, e-mail, CPF e endereço;
Entidade **Curso**: tem os atributos de descrição, título, quantidade de alunos e data de início.

Teremos o seguinte diagrama abaixo representando as entidades:

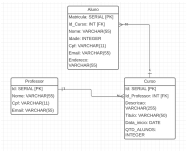


Fonte: Autoria própria.

Após identificarmos as entidades e relacionamentos existentes, podemos identificar os relacionamentos entre elas.

> A descrição do relacionamento entre as entidades de **Aluno** e **Curso** seria a seguinte: um curso pode ter muitos(**N**) alunos matriculados e um aluno pode estar matriculado somente em **1** curso, então neste caso temos um relacionamento **1(Curso):N(Alunos)**;
>
> A descrição do relacionamento entre as entidades de **Professor** e **Curso** seria a seguinte: um professor pode instruir muitos(**N**) cursos e um (**1**) curso pode ser instruído por 1 professor, então neste caso temos um relacionamento **1(Professor):N(Curso)**.

Este será o resultado do diagrama:



Fonte: Autoria própria.

Aqui vai a recomendação de algumas ferramentas para construir seu DER:

Ferramentas	Link de Acesso
Draw.io	https://app.diagrams.net/
Lucidchart	https://lucid.app/
brModelo	http://www.sis4.com/brModelo/
StarUML	https://staruml.io/

Referências e materiais complementares

MER e DER: o que é, as principais diferenças e como usar

O que é UML (Unified Modeling Language)

Diagrama entidade relacionamento