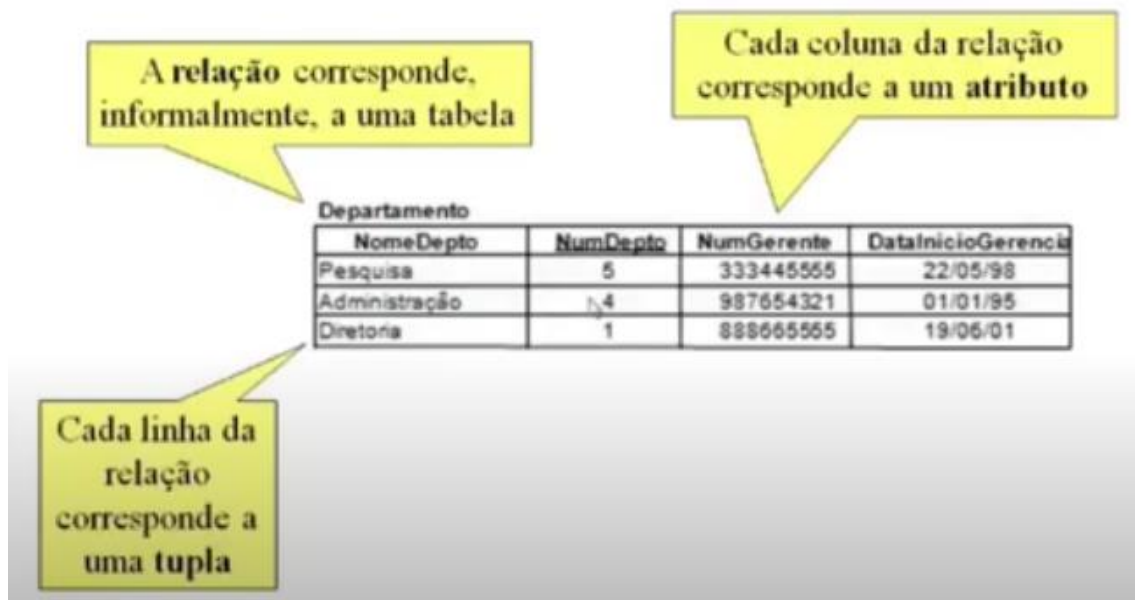


1. Modelo Relacional

1.1– Conceitos Básicos

O modelo relacional se aproxima mais de como um banco de dados é realmente implementado. Ele representa o banco de dados como um conjunto de relações, que são representados em forma de tabelas. Os nomes das tabelas e das colunas são utilizados para auxiliar na interpretação da relação.



Cada coluna corresponde a um atributo da relação, também podendo indicar uma conexão com outra tabela

O tipo do dado de cada atributo (coluna) é chamado de domínio. Esse domínio é um conjunto de valores atômicos, ou seja, indivisível. Definir o domínio do BD consegue evitar que valores inválidos sejam atribuídos aos atributos.

Um esquema de relação diz respeito a expressão da “tabela” em questão: $R(A_1, A_2, A_3, \dots, A_n)$, onde:

- R: nome da relação/tabela;
- A_i : nome de um atributo, onde o domínio em R é expressado por “dom(A_i)”;
- n: grau da relação, ou seja, quantidade de atributos.

Exemplos:

- Estudante (nome, matrícula, endereço, telefone, dataNasc)
- Disciplina (nome, código, cargaHorária, numCréditos)

Estudante possui grau 5 e Disciplina possui grau 4.

Uma instancia de uma relação são as tuplas que preenchem a tabela, desde que respeitem o domínio do banco de dados ou tenha um valor especial nulo. A ordem das tuplas não é relevante, mas a ordem dos valores dentro de cada tupla é, pois, cada valor tem que respeitar o domínio de cada atributo especificado.

Um esquema de um banco de dados relacional diz respeito a um conjunto de esquemas de relação e um conjunto de restrições de integridade. E uma instancia de um BD é um conjunto de instancias de relação.

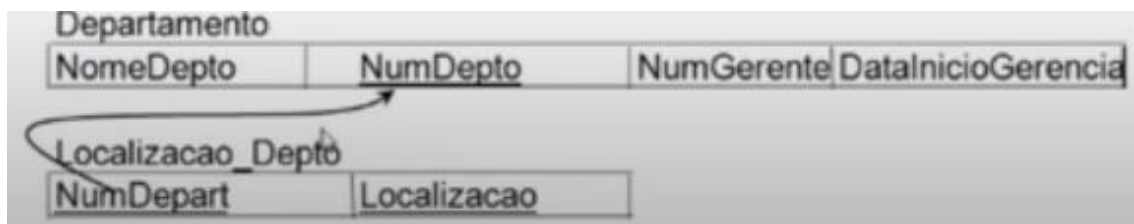
1.2– Restrições do Modelo Relacional

Restrição de domínio é uma regra em que cada valor de cada atributo deve ser um valor indivisível no domínio do atributo, ou seja, deve respeitar o domínio que foi indicado.

A restrição de chave diz respeito que toda relação deve ter uma chave. A chave é um conjunto mínimo de valores que identifica unicamente a tupla.

A restrição de integridade de entidade especifica que chaves primárias não podem ter valor nulo.

A restrição de integridade referencial é especificada como uma relação conversa com outra. Essa “conversa” se da por meio da chave primária delas, que recebe o nome de chave estrangeira. Estas são obtidas por meio dos relacionamentos entre as entidades e podem ser representadas graficamente pelo seguinte:



Também pode ser representado pela notação $R1[fk] \rightarrow R2[pk]$, onde fk é a chave estrangeira e pk é a chave primaria e $R1$ depende do valor de $R2$. Também é definido uma opção de exclusão para a relação nas quais são:

- Bloqueio (restrict): se a tupla referencia a tupla a ser excluída por meio de uma chave estrangeira, a exclusão não acontece.

- Propagação (cascade): todas as tuplas que referenciam a tupla a ser excluída por meio de uma chave estrangeira, são excluídas automaticamente
- Substituição por nulo (set null): todas as tuplas que referenciam a tupla a ser excluída tem os valores das chaves estrangeiras modificadas para nulo (se for permitido) e a exclusão é efetuada.

2 Mapeamento ER para relacional

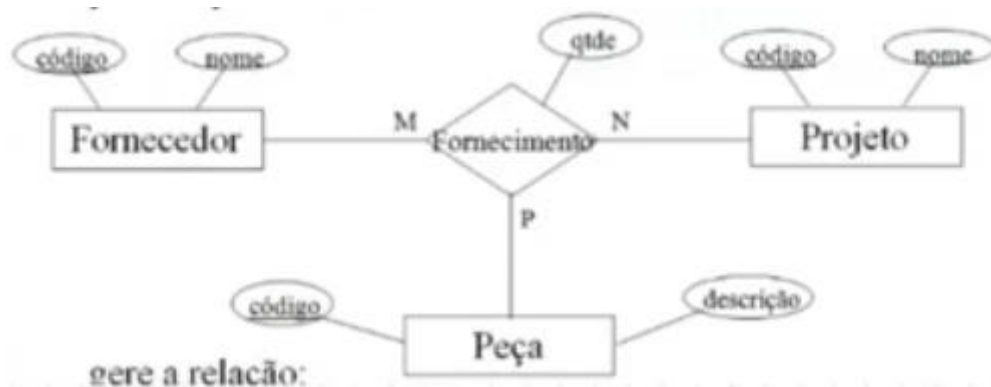
Para cada tipo de entidade E no esquema, cria-se uma relação R que contenha os atributos simples de E. Todos os atributos compostos entrarão como os atributos que os compõe, para obedecer a regra de que todos os atributos na relação sejam indivisíveis. Também escolhemos uma chave primária dentre as chaves candidatas. Caso possua uma chave estrangeira, é representado por $R1[fk] \rightarrow R2[pk]$, onde “ \rightarrow ” tem uma função que define como ocorrerá a exclusão caso essa for solicitada.



Para entidade empregado, gera-se: Empregado (cpf, primeiroNome, sobrenome, salário).

3 Mapeamento ERE para relacional

Para o relacionamento n-ário ($n \geq 2$), cria-se uma relação para o relacionamento em que terá as chaves primárias de cada relação que faz parte do relacionamento.



Para esse exemplo, temos que:

- Fornecedor: (código, nome);
- Projeto: (código, nome);
- Peça: (código, descrição);

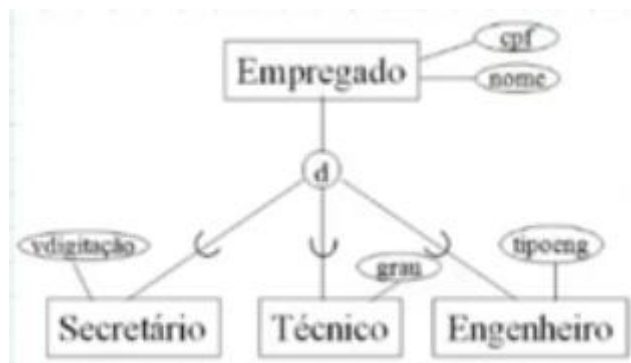
Então, temos que o relacionamento entre os três se dá por:

- Fornecimento: (codForn, codProj, codPeça, qtde, data).

Como cada “cod” é uma chave estrangeira originada das relações, é necessário definir uma restrição de integridade:

- Fornecimento [codForn] → {b ou p} Fornecedor [código]
- Fornecimento [codProj] → {b ou p} Projeto [código]
- Fornecimento [codPeca] → {b ou p} Peça [código]

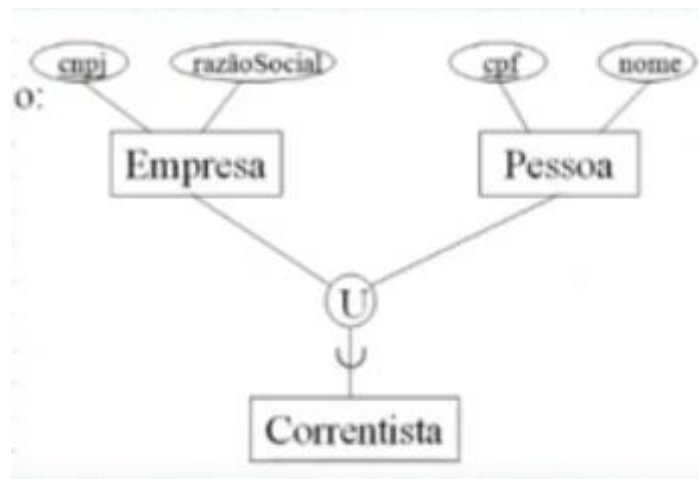
3.1 – Mapeamento de Especialização



Existem 4 opções para criar o relacional:

- Opção 1: serve para todos os casos. Cria uma nova relação para cada superclasse e para cada subclasse. Então, é incluída a chave primaria em cada subclasse que herdou da superclasse.
- Opção 2: cria-se apenas as subclasses com todos os atributos da superclasse herdados e os atributos isolados de cada subclasse. Funciona apenas para especialização total e disjunta para evitar redundâncias.
- Opção 3: cria-se apenas a superclasse. Todos os atributos das subclasses são representados na superclasse e cria-se um novo tipo para identificar qual tipo aquela relação representa.
- Opção 4: similar a opção 3, apenas mapea-se a superclasse. Porém, cria um flag para cada subclasse. Essa opção serve para contornar o erro caso a especialização não for disjunta.

3.1– Mapeamento da União



Em uma união, na subclasse cria-se uma chave substituta apenas para a união. Após isso, inclui-se como chave estrangeira em cada superclasse. Então, são feitas restrições de integridade para cada superclasse.

- Correntista (código, ...)
- Empresa (cnpj, razãoSocial, ..., codCorr)
- Pessoa (cpf, nome, ..., codCorr)
- Empresa [codCorr] → {n} Correntista [código]
- Pessoa [codCorr] → {n} Correntista [código]

3.2– Mapeamento de Agregação

Basta criar uma relação para o relacionamento que já está representando a agregação.