

INSTITUTO FEDERAL DE CIÊNCIA E TECNOLOGIA DO RIO G. DO NORTE – IFRN
DISCIPLINA DE BANCOS DE DADOS
RESUMO DE AULA – CONCEITOS INICIAIS

1. INTRODUÇÃO

Pode se desenvolver qualquer software (programa de computador) utilizando os conhecimentos adquiridos na disciplina Lógica de Programação e Algoritmos, porém para criar um sistema de informação, é preciso que esse software guarde informações. Seja o nome dos usuários que podem ter acesso, endereços, boletos ou qualquer outro. A melhor forma de armazenar essas informações, na maioria das vezes é utilizando um banco de dados.

Um banco de dados nada mais é que um software criado apenas para tornar mais fácil armazenar, alterar e consultar esses dados em curto intervalo de tempo, seja quais forem os dados, mantendo-os organizados.

1.1. TABELA

Antes de estudarmos banco de dados, vamos pensar como poderíamos guardar dados em nosso computador de forma simples e organizada sem uso de nenhuma ferramenta nova. A primeira idéia que surge em mente é guardar esses dados em planilhas/tabelas, seja utilizando o MS Excel, LO Calc, ambos criados para a mesma tarefa.

Ao guardarmos informações em um desses softwares, usaremos tabelas. Imagine guardar, por exemplo, os dados de uma agenda telefônica, para isso, criaríamos uma tabela conforme a Tabela 1, abaixo.

Nome	Telefone	Nacionalidade
João	(00) 0000-0000	Brasileiro
Maria	(00) 0000-0000	Brasileiro
José	(00) 0000-0000	Brasileiro
Joaquim	(00) 0000-0000	Espanhol

Tabela 1: Agenda Telefônica

No banco de dados acima, a nacionalidade foi repetida várias vezes. Imagine um banco de dados com milhares ou milhões de usuários. Esses dados precisariam ser guardados repetidamente gastando memória do computador e causando demora para consultar esses dados por ser necessário consultar em um conjunto maior de informações. Contudo a forma de separar os dados em tabela é bem interessante, pois qualquer informação pode ser guardada de forma organizada usando este recurso. Tabelas e objetos são formas de representar entidades.

Note também que o telefone, neste caso, repete para todos os usuários. Este campo precisará ser guardado de forma repetida, pois apesar de ser possível sua repetição, isto ocorre de forma **eventual**.

Já a nacionalidade, embora possa ser alterada caso o cidadão perca sua nacionalidade, constará de repetições **em massa**. Neste caso, guardando sem repetição, nos permitirá evitar as repetições. Veremos adiante como fazer isto.

1.2 COLUNAS

Em uma tabela, temos colunas. Existem bancos de dados orientados a objeto e, se fosse esse

o caso, diríamos que as colunas representam atributos. Existem bancos de dados orientados a objetos, então também é correto o uso do termo atributo, mas nesse caso esses bancos de dados não são baseados no **modelo entidade-relacionamento** e sim no **modelo de orientação a objetos**. Em bancos de dados, usa-se os termos coluna e atributo como sinônimos, embora o nome atributo gere confusão por se referir ao modelo de OO.

1.3 LINHA

De forma semelhante, temos as linhas da tabela. Em bancos de dados orientados a objeto, as linhas seriam chamadas de instâncias do objeto. Em geral, usa-se os termos linhas e registros como sinônimos.

2. BANCO DE DADOS

Para evitar o problema apontado na Tabela 1, temos de separar os dados que se repetem em tabelas diferentes, isto é, uma tabela apenas com nome e telefone e uma outra tabela apenas com nacionalidades, por exemplo. Assim, já que uma tabela ficaria apenas o nome nacionalidade, nela, não seria necessário repetições. Veja abaixo como ficaria, no Quadro 1.

Nome	Telefone	Nacionalidade
João	(00) 0000-0000	Brasileiro
Maria	(00) 0000-0000	Espanhol
José	(00) 0000-0000	
Joaquim	(00) 0000-0000	

Quadro 1: Agenda Telefônica sem repetição

A solução acima, resolve os problemas de repetição de dados, porém agora não temos como identificar qual usuário tem qual nacionalidade. Para remover esse problema podemos rotular essas linhas, e repetir esse rótulo de uma tabela para outra, por exemplo, veja o Quadro 2, abaixo. Note que foram dados nomes as tabelas e renomeado colunas.

Usuário

Nome	Telefone	Rotulo 1
João	(00) 0000-0000	A
Maria	(00) 0000-0000	A
José	(00) 0000-0000	A
Joaqui m	(00) 0000-0000	B

Nacionalidade

Nome	Rotulo 2
Brasileiro	A
Espanhol	B

Quadro 2: Agenda Telefônica com rótulos

Note, no Quadro 2, que João, Maria e José estão com o mesmo rótulo, a letra A, enquanto apenas Joaquim está rotulado como B. Note também que na outra tabela, temos a informação que o rótulo A representa a nacionalidade Brasileiro, enquanto o rótulo B representa Espanhol.

No Quadro 2, as tabelas foram nomeadas, assim temos agora uma tabela cujo nome é Usuário e uma outra tabela cujo nome é Nacionalidade, e temos duas novas colunas, Rótulo 1 e

Rótulo 2.

Caso desejemos incluir uma nova possibilidade de nacionalidade, poderíamos incluir mais um registro na tabela nacionalidade, porém note que não faz sentido que o Rótulo 2 se repita, pois caso mais de uma nacionalidade tenha o mesmo rótulo, e um usuário tenha um destes, não teríamos como identificar qual seria correto, por existir mais de uma possibilidade com mesmo rótulo.

Já na tabela Usuário, faz sentido repetir este rótulo, pois mais de um usuário pode morar possuir mesma nacionalidade, por exemplo, João, Maria e José são todos brasileiros.

2.1 Chaves

Notamos que no Quadro 2, temos um dos rótulos, um que não faz sentido repetir e outro que faz sentido repetir.

- *Chave Primária*

A um rótulo que seu conteúdo não pode repetir, chamaremos de chave primária. A partir de agora, sempre que formos definir uma chave primária, o nome da coluna que armazenará essa chave, daremos o nome de pk (do inglês, Primary Key) ou id (Identificador).

- *Chave Estrangeira*

A um rótulo que pode repetir, mas que necessariamente seu conteúdo deve ser uma cópia de uma chave primária, chamaremos de agora em diante de chave estrangeira. O nome da coluna que armazenará essa chave, a partir de agora nomearemos de fk (do inglês, Foreign Key) e após o fk, colocaremos o nome da tabela de onde será feita a cópia da chave primária, por exemplo, fk_nacionalidade.

2.2 Relacionamentos

Analisando, intuitivamente o Quadro 2, temos duas tabelas e sabemos que os dados de uma tabela estão relacionados com os de outra, isto é, precisamos olhar as duas tabelas para interpretarmos a informação. Quando duas tabelas estão relacionadas, devemos desenhar um traço unindo essas tabelas, como por exemplo, mostrado no Quadro 3.

Além disso, devemos saber qual a relação entre elas. Por exemplo, responderemos duas perguntas de acordo com o diagrama mostrado no Quadro 2 (sem considerar suas filosofias de vida, apenas os dados das tabelas acima).

A primeira: Um usuário pode ter quantas nacionalidades? A resposta a esta primeira pergunta deve ficar sobre o traço que indica o relacionamento e do lado de nacionalidade, pois a pergunta se referiu a quantas nacionalidades.

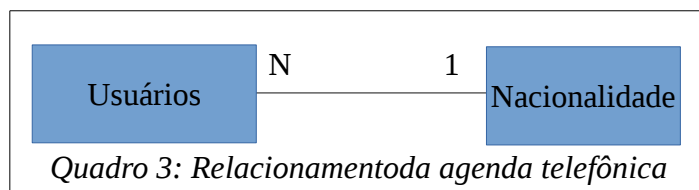
A segunda: Uma nacionalidade pode ter quantos usuários? A resposta a esta segunda pergunta deve ficar também sobre o traço do relacionamento, porém do lado de usuário, pois agora a pergunta se referiu a quantos usuários.

A resposta da primeira questão é que, como um usuário só pode ter um rótulo, e cada rótulo só pode representar uma nacionalidade, então, um usuário só pode, neste exemplo, ter uma nacionalidade. Para a segunda questão, uma nacionalidade só pode ter um rótulo e este rótulo pode estar repetido entre vários usuários, então uma nacionalidade pode ter diversos usuários.

Veja no Quadro 3, abaixo, que a resposta das perguntas feitas devem aparecer no diagrama. Por simplicidade, neste diagrama em específico não aparecerão colunas nem registros. Note que quando se pergunta sobre um usuário, quantas nacionalidades, a resposta fica do lado de nacionalidade, pois a pergunta sobre quantos, se refere a nacionalidades. No caso da pergunta de uma nacionalidade, quantos usuários, a pergunta se refere a quantos usuários, então a resposta fica do lado de usuários. Note também que a pergunta sempre parte de UMA unidade de uma tabela a

perguntar quantos da tabela oposta.

Para qualquer diagrama acharemos esses valores com o mesmo tipo de pergunta, UM de uma tabela, tem quantos da outra tabela. Depois fazemos a pergunta inversa. No lugar de “vários” usaremos a letra N, porém em outros modelos do mesmo diagrama, será encontrado também o símbolo * simbolizando também a mesma coisa.



2.2.1 Cardinalidade

Aos valores colocados sobre o relacionamento, chamamos de cardinalidade do relacionamento. Em geral são colocados os valores 1 ou N para representar, respectivamente, um para muitos. No exemplo do Quadro 3, podemos ler o relacionamento como “uma nacionalidade possui muitos usuários”.

3. DER

DER é a abreviação de Diagrama de Entidade-Relacionamento. Nada mais é que uma forma de desenhar o banco de dados com todos os dados e conceitos vistos nesse resumo de aula de forma que pessoas do mundo inteiro compreendam o que foi escrito. Abaixo existe um exemplo, porém existem várias representações para simbolizar o mesmo conceito. Abaixo temos duas dessas representações.

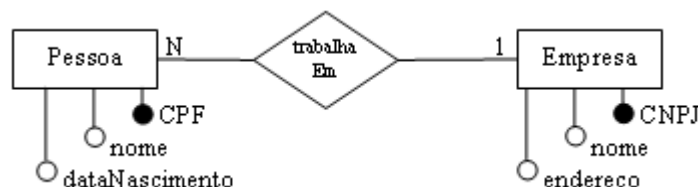


Figura 1: Losangos nomeiam o relacionamento, bolhas pretas simbolizam chaves primárias, chave estrangeira não definida no diagrama.

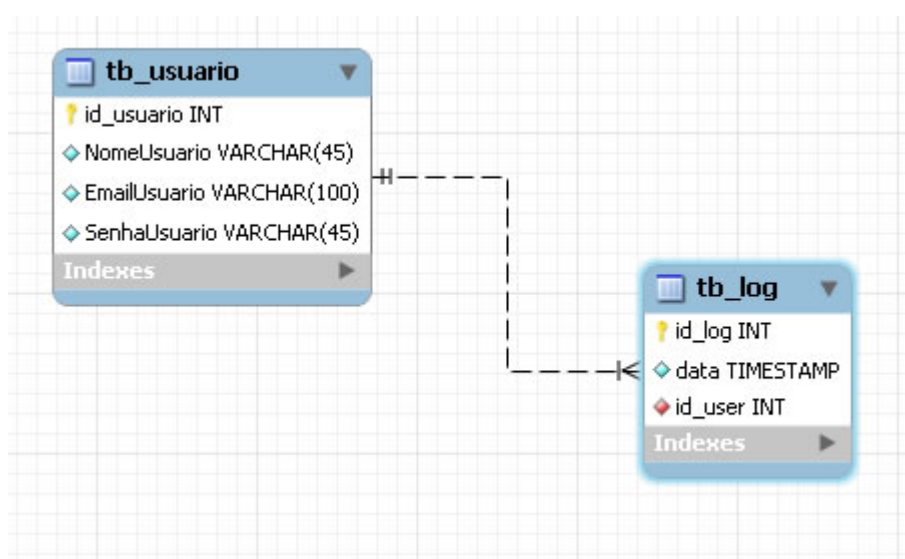


Figura 2: Pés de galinha simbolizam o N, cada traço simboliza o 1, numero 0 simboliza o zero, desenho dourado da chave simboliza chave primária, losango vermelho, chave estrangeira.

3.1 MODELO CONCEITUAL

O modelo conceitual tem o objetivo de criar uma representação do banco de dados de forma que seja totalmente independente de tecnologia que irá implementar o banco de dados. Devemos notar e chamar atenção aqui que qualquer forma de armazenar dados pode ser chamada de banco de dados. E.g. Arquivos e diretórios, entidade-relacionamento, modelo orientado a objetos, entre outros.

Assim, nem todos estes meios fazem uso de chaves primárias, cabeçalhos de tipo de arquivo, mas mesmo arquivos e diretórios podem ser considerados entidades, conter informações sequenciais ao invés da forma tabular usada em tabelas e relacionar-se entre arquivos.

No modelo conceitual nosso objetivo deve ser apenas criar um diagrama informando quais entidades de informação serão armazenadas pelo nosso sistema e suas relações, independente da forma de implementação. Um exemplo pode ser visto acima na Figura 1.

3.2 MODELO LÓGICO

Enquanto o modelo conceitual não se preocupa com a forma de implementação, o modelo lógico traz essa preocupação e para isso não nos é mais importante o nome de suas relações e sim como elas serão implementadas no banco de dados.

Deve-se atentar aqui que no modelo entidade-relacionamento, usando tabelas, as suas relações são criadas pelas chaves estrangeiras que são dependentes das chaves primárias, assim, para criação do modelo lógico de um banco de dados usando tabelas, devemos além do nome das entidades e suas relações, informar também quais as chaves primárias e estrangeiras presentes para criar os relacionamentos.

LISTA DE EXERCÍCIOS - \ô/

Para os exercícios abaixo, construa um DER (modelo conceitual e lógico) para cada questão.

Lembre que cada relacionamento dá origem necessariamente a uma chave primária e uma estrangeira. Uma chave estrangeira só serve para um relacionamento.

Obs.: Pense bem no problema, pois guardaremos no banco de dados informações da vida real, embora o comum seja a casa estar em uma rua, ela pode ter, por exemplo, uma fachada residencial para uma rua e comercial para outra. Busque informações na internet caso necessário para entender cada problema.

Obs.2: Pode ser necessário criar alguma tabela adicional para que as tabelas tenham sentido juntas. Se sim, justifique, mas só adicione caso seja realmente necessário.

1. Em um sistema de vendas de supermercado temos as tabelas: Produto (Itens a ser vendidos), Cliente (Comprador de produtos) e Cupom Fiscal (Representa uma nota de compra).
2. Na base de dados dos correios temos Rua (Informações da rua, incluso o cep, numero inicial e final da rua), Bairro, Cidade e Estado (Unidades de localização hierárquica).
3. Em uma rádio temos Música (Canções), Autor (Mesmo que compositor), Cantor (Quem canta), Ano (Data de gravação da música), Gravadora (Quem gravou o disco físico), Disco (Unidade física contendo a música).
4. Um sistema de vendas online temos Pedido (Agrupamento de produtos com dados da venda), Cliente (Quem compra produtos), Produto (Item a ser comprado), Endereço.
5. Um restaurante tem Pedido (Um pedido individual, não reflete a conta total), Mesa (Local onde se é atendido), Garçom (Quem atende a mesa), Cliente (Consumidor), Pagamento (Pagamento do cliente por seus gastos, não tem relação com a conta total).
6. Uma escola tem em seu banco de dados Aluno (Estudante), Turma (Agrupamento de estudantes), Professor (Quem leciona), Notas (Considerar sistema de notas semelhante a um boletim, de ensino médio), Disciplina (Considerar as disciplinas de ensino médio).
7. Em um banco de dados para armazenar dados de aulas, temos as tabelas Usuário (Dados pessoais do usuário), Questão (Dados de questões de multipla escolha), Lista de Exercícios (Agrupação de questões), Aula (Itens para estudo referentes a uma aula), Disciplina (Uma matéria onde a aula está inserida).
8. Em um sistema de almoxarifado temos Requesições (Pedidos de saída de material), Produtos, Setor (local requisitante do material), Funcionário (solicitante e responsável pela entrega do material) e Notas de compra (Dados de compra de novos materiais).
9. Em um banco, temos Conta (local virtual onde é gerenciado valores monetários), Tipo de Conta (Informação do tipo, por exemplo, poupança, conta salário, ...), Usuário (Dados pessoais de um usuário), Movimentação (Movimentação financeira entre contas).
10. Em uma biblioteca temos Cliente, Livro, Estante (Onde está guardado o livro), Prateleira, Corredor e Bloco.