



NORMATIZAÇÃO DE DADOS E ÍNDICES

Análise e Desenvolvimento de Sistemas - Senai 2020



SQL SERVER

E aí pessoas? De boas?

Vamos falar um pouco sobre normalização em banco de dados, beleza?

Mas antes de aprofundarmos nossos conhecimentos sobre este importante tema, vamos repassar uma breve revisão sobre banco de dados e modelagem de dados, ok?

Banco de Dados

"Bancos de dados são coleções de dados/informações que se relacionam de forma dinâmica de modo a criar sentido para determinado fim e são de vital importância para o mundo empresarial.

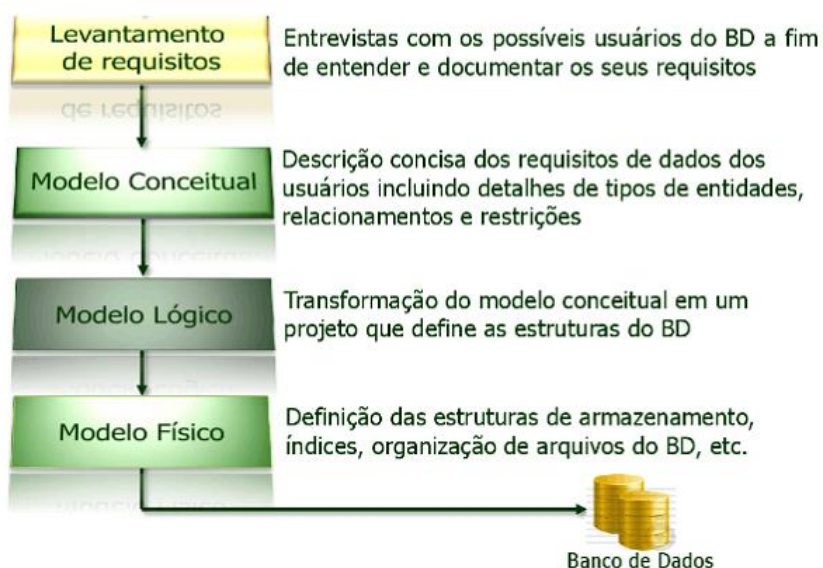
Dados são fatos brutos, em sua forma primária. E muitas vezes os dados podem não fazer sentido sozinhos.

Informação consiste no agrupamento de dados de forma organizada para fazer sentido, gerar conhecimento.

Por exemplo, o número 2001 isoladamente faz algum sentido? Não! Isso é um dado. E se eu dissesse: "Ano do atentado terrorista às torres gêmeas: 2001"? Agora faz sentido! Isso é uma informação. Um banco de dados é uma estrutura de dados organizada que permite a extração de informações no banco de dados.

Etapas na modelagem de um banco de dados

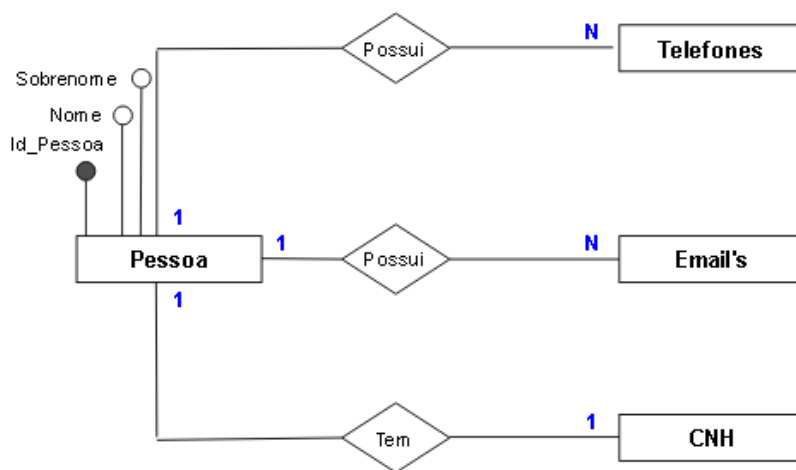
Etapas de um projeto de BD



Modelagem Conceitual

O objetivo aqui é criar um modelo conceitual de forma gráfica, sendo este chamado de Diagrama Entidade e Relacionamento (DER), que identificará todas as entidades e relacionamentos de uma forma global.

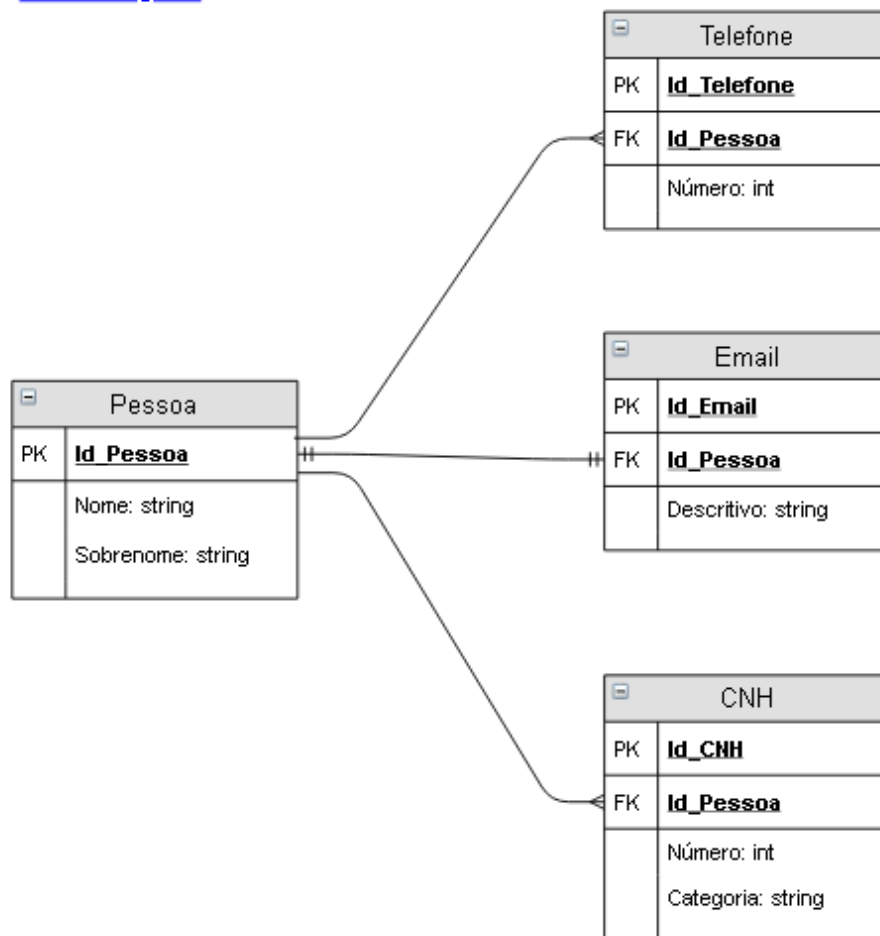
Modelo Conceitual : Uma pessoa pode ter vários telefones e vários emails, mas pode conter apenas uma CNH.



Modelagem Lógica

Depois que selecionamos um SGBD, iremos mapear o modelo conceitual para ele. O projetista relaciona as características e restrições do modelo conceitual com as do modelo selecionado para implementação. O modelo lógico constitui uma representação específica de um modelo interno, utilizando as estruturas de BD suportada pelo banco escolhido.

Modelo Lógico :



Modelagem Física

Aqui é trabalhado no nível mais baixo de abstração, descrevendo o modo como os dados são salvos em meios de armazenamentos, como discos e fitas, sendo exigido a definição tanto dos dispositivos de armazenamento físico como dos métodos de acesso (físico) necessários para se chegar aos dados nesses dispositivos, o que o torna dependente tanto de software como de hardware. Os projetistas precisam de um conhecimento detalhado do hardware e do software utilizado para implementar o projeto de BD.

Modelo Físico_Pessoas

Pessoa		
Id_Pessoa	Nome	Sobrenome
1	João	Brigadeiro
2	Maria	Cookie
3	Justino	Justo

Email		
Id_Email	Id_Pessoa	Descritivo
7	1	brigadeiro@gmail.com
8	2	cookie@gmail.com
9	3	justo@yahoo.com

Telefone		
Id_Telefone	Id_Pessoa	Número
4	1	1150506070
5	2	1190909090
6	3	1191919191

CNH			
Id_CNH	Id_Pessoa	Número	Categoria
1	1	11111111111	A
2	2	22222222222	B
3	3	33333333333	E

Ressaltando que, a criação do banco de dados na linguagem SQL é uma só, porém ela é dividida em tipos de acordo com a funcionalidade dos comandos.

Os tipos da linguagem SQL são:

DDL - Data Definition Language - Linguagem de Definição de Dados.

São os comandos que interagem com os objetos do banco.

São comandos DDL: CREATE, ALTER e DROP

DML - Data Manipulation Language - Linguagem de Manipulação de Dados.

São os comandos que interagem com os dados dentro das tabelas.

São comandos DML: INSERT, DELETE e UPDATE

DQL - Data Query Language - Linguagem de Consulta de dados.

São os comandos de consulta.

São comandos DQL: SELECT (é o comando de consulta)

O que diferencia na identificação destas siglas é a letra do meio.

DML = "M" do meio vem de Manipulação.

DDL = "D" do meio vem de Definição.

DQL = "Q" do meio vem de "Consulta".

Agora sim, vamos ao que nos interessa para o momento.

Normalização em Banco de Dados

O processo de normalização compreende o uso de um conjunto de regras, chamados de formas normais. É um processo a ser realizado nas tabelas do banco de dados para evitar falhas no projeto, como por exemplo redundância de dados, evitando assim, que o banco de dados fique mais lento, pesado ou com informações repetidas desnecessariamente.

O banco de dados deve ser consistente e íntegro em uma boa modelagem de dados. Para garantir integridade e consistência na construção do banco de dados, algumas normas devem ser observadas, entre as quais a três primeiras são vitais para um bom banco de dados.

Primeira Forma Normal (1FN)

Uma tabela deve possuir:

- Apenas valores atômicos (campos compostos por apenas um valor);
- Não pode haver grupos de atributos repetidos;
- Deve conter uma chave primária;
- Não pode conter atributos multivalorados, compostos ou relações aninhadas (uma tabela dentro da outra).

Primeira Forma Normal (1FN)

Pessoa

CPF	Nome	Sexo	Localização	Telefone
333	Lia	F	Rio de Janeiro, RJ, Brasil	999-555, 777-333
222	Caio	M	Brasília, DF, Brasil	555-888, 333-222
111	Ana	F	São Paulo, SP, Brasil	444-999
444	Beto	M	Porto Alegre, RS, Brasil	888-222



O campo telefone é um campo multivalorado, pois uma pessoa pode ter mais de um contato/telefone.



Trata-se de um campo composto

Adaptado de You Tube: EdukaTI

Primeira Forma Normal (1FN)

Pessoa

CPF	Nome	Sexo	Localização	Telefone
333	Lia	F	Rio de Janeiro, RJ, Brasil	999-555, 777-333
222	Caio	M	Brasília, DF, Brasil	555-888, 333-222
111	Ana	F	São Paulo, SP, Brasil	444-999
444	Beto	M	Porto Alegre, RS, Brasil	888-222

CPF	Nome	Sexo	Cidade	UF	País
333	Lia	F	Rio de Janeiro	RJ	Brasil
222	Caio	M	Brasília	DF	Brasil
111	Ana	F	São Paulo	SP	Brasil
444	Beto	M	Porto Alegre	RS	Brasil

CPF	Telefone
333	999-555
333	777-333
222	555-888
222	333-222
111	444-999
444	888-222

Obs: Os campos composto e os multivalorados devem ser descritos corretamente conforme as figuras em detalhes.

Adaptado de You Tube: EdukaTI

Segunda Forma Normal (2FN)

Antes de mais nada, para a tabela atender a 2FN é preciso estar na 1FN.

Além disso, todos os atributos não chaves da tabela devem depender unicamente da chave primária (não podendo depender apenas de parte dela).

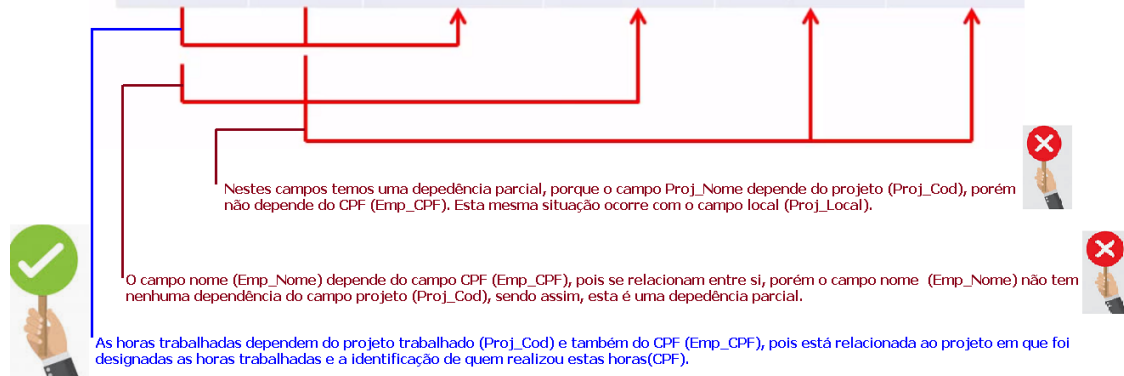
Para dispor a tabela na segunda forma normal, é necessário identificar as colunas que não são funcionalmente dependentes da chave primária da tabela e, em seguida, remover essa coluna da tabela principal e criar uma nova tabela com esses dados.

Exemplo:

Segunda Forma Normal (2FN)

Empregado_Projeto

Emp_CPF	Proj_Cod	Horas_Trabalho	Emp_Nome	Proj_Nome	Proj_Local
333	1A	100	Lia	SisRH	RJ
222	3C	400	Caio	GTI	DF
111	3C	300	Ana	GTI	DF
333	2B	200	Lia	SisCorp	RS



Adaptado de You Tube: EdukaTI

Esta tabela esconde uma relação N para N, o que gera os seguintes problemas:

Anomalias/Erro inserção e exclusão de dados

Empregado_Projeto

Emp_CPF	Proj_Cod	Horas_Trabalho	Emp_Nome	Proj_Nome	Proj_Local
333	1A	100	Lia	SisRH	RJ
222	3C	400	Caio	GTI	DF
111	3C	300	Ana	GTI	DF
333	2B	200	Lia	SisCorp	RS
555	1A	50	Igor	SisRH	RJ

Anomalia de inserção de dados: Considerando que inserimos o funcionário Igor devido a um novo projeto, as informações da lotação do mesmo (SisRH e RJ) dispostas no campos Proj_Nome e Proj_Local simultaneamente) se tornam redundantes, pois na linha da tabela onde se localiza a funcionária Lia, já constam esses dados de lotação, tornando-os duplicados na tabela.

Por outro lado, aqui temos uma anomalia de exclusão de dados, pois se por um acaso fosse necessário retirar a funcionária Lia neste campo, o dado de lotação no estado "RS" seria perdido do banco de dados.

Adaptado de You Tube: EdukaTI

Podemos normalizar esta tabela, modelando estes dados em três tabelas, conforme a seguir:



Adaptado de You Tube: EdukaTI

Desta forma, as alterações simuladas podem ser efetuadas normalmente, sem que haja anomalias na manipulação dos dados destas tabelas.

Terceira Forma Normal (3FN)

Para estar na 3FN, é preciso estar na 2FN. Além disso, os atributos não chave de uma tabela devem ser mutuamente independentes e dependentes unicamente e exclusivamente da chave primária (um atributo B é funcionalmente dependente de A se, e somente se, para cada valor de A só existe um valor de B).

Para atingir essa forma normal, é preciso identificar as colunas que são funcionalmente dependentes das outras colunas não chave e extraí-las para outra tabela.

Ou seja,

“Uma relação está em 3FN se e somente se estiver na 2FN e nenhum atributo não-primário (isto é, que não seja membro de uma chave) for transitivamente dependente da chave primária.”

- ❑ Dependência Transitiva: ocorre quando uma coluna, além de depender da chave primária de uma tabela, depende de outra coluna ou conjunto de colunas da tabela.
- ❑ Uma relação para estar na 3FN não deve ter um atributo não-chave funcionalmente determinado por um outro atributo não-chave.

Exemplo:

Terceira Forma Normal (3FN)

Empregado_Departamento

Emp_CPF	Emp_Nome	Emp_Sexo	Dep_Cod	Dep_Nome	Dep_Ger
333	Lia	F	1A	RH	Igor
222	Caio	M	2B	Adm	Edu
111	Ana	F	3C	TI	Tiago
444	Beto	M	2B	Adm	Edu

Neste exemplo a chave primária é CPF e as colunas determinadas por CPF são: nome (Emp_Nome), sexo (Emp_Sexo) e código do departamento a qual pertence o funcionário (Dep_Cod).

Porém as colunas nome do departamento (Dep_Nome) e departamento geral (Dep_Geral) que são atributos não chave, dependem somente da chave código do departamento (Dep_Cod), que não é a chave primária e não se relacionam neste caso, com CPF e nem com nome do funcionário.

Neste exemplo temos uma dependência transitiva e uma dependência parcial, podendo causar anomalias de inserção, exclusão ou atualização.

Empregado

CPF	Nome	Sexo	Dep_Cod
333	Lia	F	1A
222	Caio	M	2B
111	Ana	F	3C
444	Beto	M	2B

Departamento

Cod	Nome	Ger
1A	RH	Igor
2B	Adm	Edu
3C	TI	Tiago

Para normalizar esta modelagem é necessário separar as colunas, criando uma tabela para empregado e outra para departamento, conforme o exemplo ao lado.

ÍNDICES - O que são? Para que servem? Onde vivem? Qual o impacto no banco de dados?

Um índice é uma estrutura em disco associada a uma tabela ou exibição, que agiliza a recuperação das linhas de uma tabela ou exibição.

Um índice contém chaves criadas de uma ou mais colunas da tabela ou exibição. Essas chaves são armazenadas em uma estrutura (árvore B) que habilita o SQL Server a localizar a linha ou as linhas associadas aos valores de chave de forma rápida e eficaz.

Clusterizado

Os índices clusterizados classificam e armazenam as linhas de dados da tabela ou exibição com base em seus valores de chave. Essas são as colunas incluídas na definição do índice. Pode haver apenas um índice clusterizado por tabela, pois as linhas de dados podem ser classificadas somente em uma única ordem.

Não clusterizado

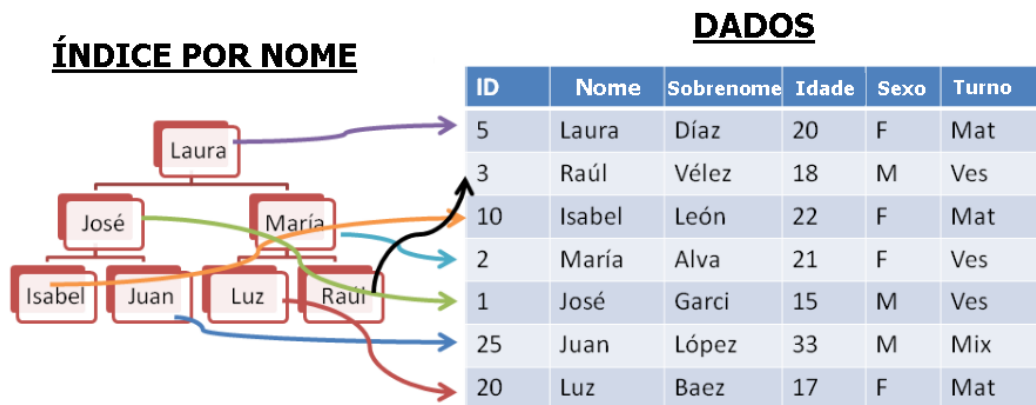
Os índices não clusterizados têm uma estrutura distinta das linhas de dados. O índice não clusterizado contém os valores de chave de índice não clusterizado e cada entrada de valor de chave tem um ponteiro para a linha de dados que contém o valor de chave.

O ponteiro de uma linha de índice em um índice não clusterizado de uma linha de dados é denominado localizador de linhas. A estrutura do localizador de linhas depende de as páginas de dados serem armazenadas em um heap ou em uma tabela clusterizada. Para o heap, o localizador de linhas é um ponteiro para a linha. Para a tabela clusterizada, o localizador de linhas é a chave de índice clusterizado.

Índices e restrições

Os índices são criados automaticamente quando as restrições PRIMARY KEY e UNIQUE são definidas em colunas de tabelas. Por exemplo, quando você criar uma tabela com uma restrição UNIQUE, Mecanismo de Banco de Dados criará automaticamente um índice não clusterizado. Se você configurar uma PRIMARY KEY, Mecanismo de Banco de Dados criará automaticamente um índice clusterizado, a menos que já exista um. Quando você tentar impor uma restrição

PRIMARY KEY em uma tabela existente e já houver um índice clusterizado nessa tabela, o SQL Server irá impor a chave primária usando um índice não clusterizado.



Bibliografia:

Canal you tube - EdukaTI

<https://docs.microsoft.com/pt-br/sql/relational-databases/indexes/clustered-and-nonclustered-indexes-described?view=sql-server-ver15>

<https://pt.stackoverflow.com/>

<http://www.regilan.com.br/wp-content/uploads/2013/10/Apostila-Banco-de-Dados.pdf>

http://www.nrsystem.com.br/Fund_Banco_Dados.pdf

