

FACULDADE SENAC GOIÁS
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
2º PERÍODO

BANCO DE DADOS
PESQUISA

LUANA FERREIRA DE ALMEIDA.

GOIÂNIA-GOIÁS
2019

BANCO DE DADOS
PESQUISA

LUANA FERREIRA DE ALMEIDA.

Exercício solicitado pelo Professor Gildenor
Redes da disciplina de Banco de Dados do
2º período do curso de Análise e
Desenvolvimento de Sistemas da Faculdade
Senac-Goiás.

GOIÂNIA-GOIÁS
2019

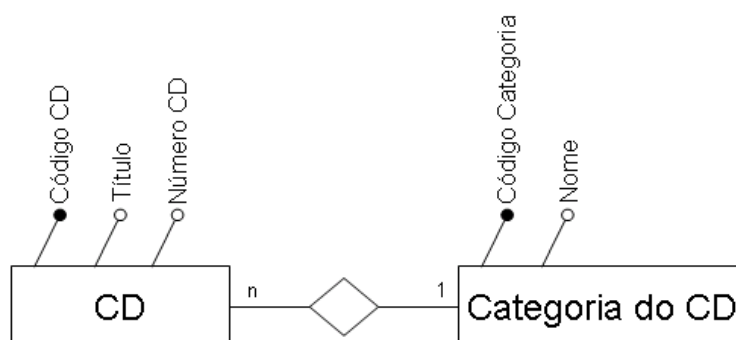
PESQUISA

Modelo de Bancos de Dados

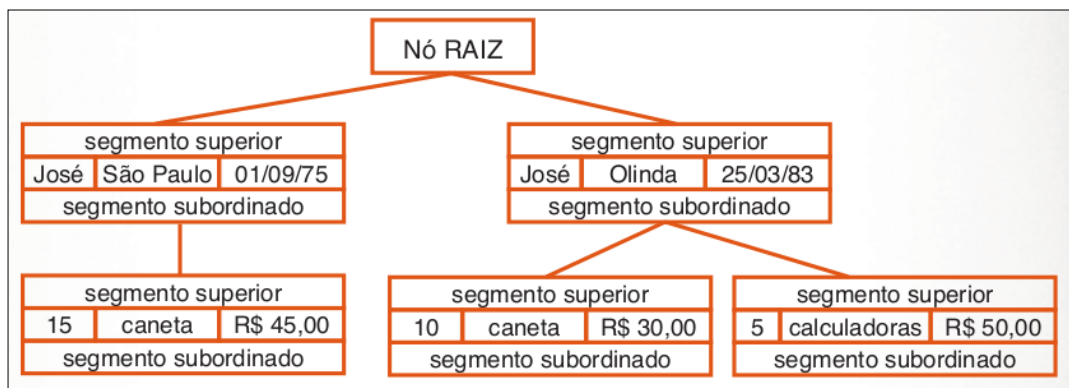
É a descrição das informações armazenadas em um banco de dados. Citando como exemplo o nosso estudo de caso, podemos dizer que o modelo de dados informa que o banco de dados armazena informações sobre CDs e que, para cada CD, estão armazenadas informações sobre o autor, título, número de CDs e as músicas. Em resumo, um modelo de banco de dados nada mais é do que a descrição formal da estrutura de um banco de dados;

Modelo Conceitual

É uma descrição do banco de dados porém de uma forma independente da implementação que será feita, em outras palavras, ele independe de qual SGBD será utilizado na implementação. Desta forma, um mesmo modelo conceitual poderá ser utilizado para implementação em Oracle, DB2, PostgreSQL, etc. O modelo conceitual indica quais os dados que poderão aparecer no banco de dados, mas não informa de que forma estes mesmos dados serão armazenados no nível do SGBD. Em resumo, um modelo conceitual é um modelo abstrato que descreve a estrutura de um banco de dados de forma independente do SGBD.

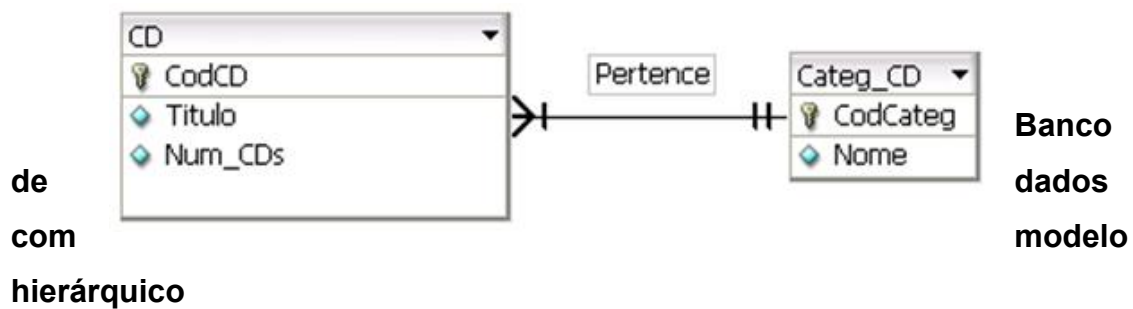


Exemplo de Modelo Conceitual.



Estrutura de banco de dados no nível do SGBD.

Exemplo de Modelo Lógico.



de
com
hierárquico

Banco
dados
modelo

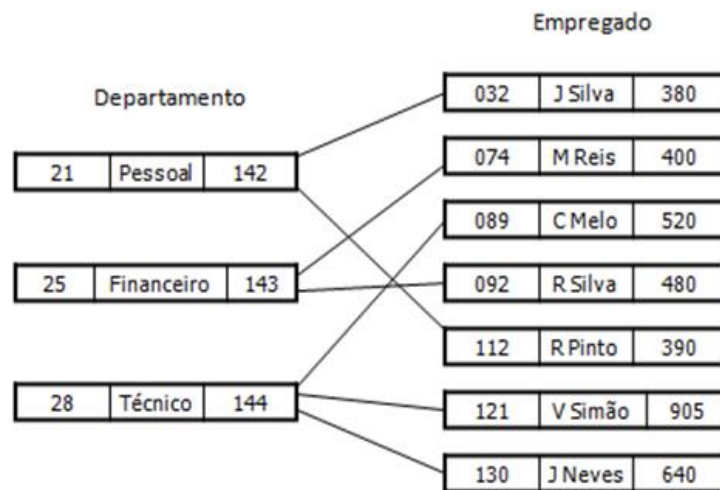
No ano de 1960, foi desenvolvido pela IBM o IMS (Information Management System), que possui como base o modelo hierárquico. Esse modelo representou um grande avanço nas técnicas de armazenamento de dados, utilizando como referência uma estrutura em árvore. Essa estrutura possibilita a organização dos registros através de ligações, nas quais cada registro possui vários campos e cada campo possui somente um valor.

Cada um dos registros possui dois segmentos: o superior e o subordinado. Eles são utilizados para a ligação com um outro registro, em um relacionamento pai-filho. O registro pai pode possuir vários filhos, mas os registros filhos só podem possuir um registro pai, como demonstrado na figura abaixo que mostra um esquema de banco de dados hierárquico:

Houve grandes melhoras na forma que o armazenamento era feito, e, pela primeira vez, os dados de diferentes assuntos poderiam ser relacionados.

Banco de dados com modelo de rede

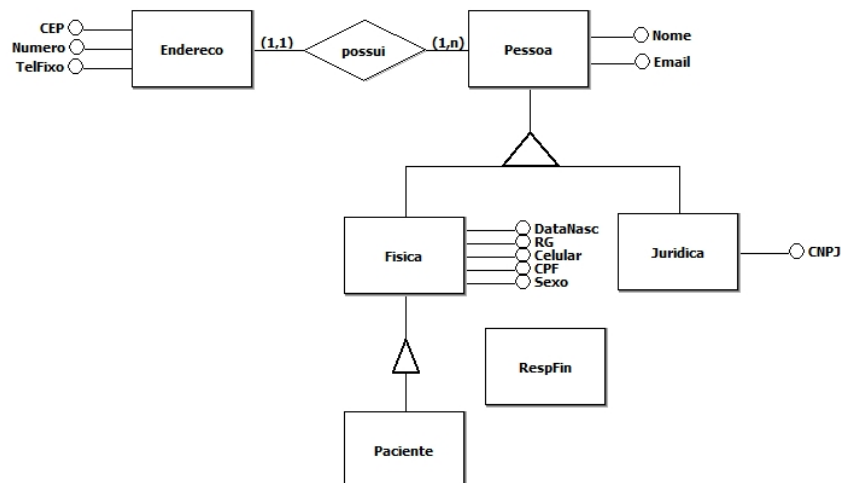
Na tentativa de melhorar o modelo hierárquico, foi criado o modelo de rede, representado por um diagrama similar ao hierárquico. No entanto, este novo modelo poderia relacionar cada registro filho com vários pais diferentes. Sua estrutura utilizava termos como entidades e atributos (registros e itens do registro, respectivamente).



As melhorias propostas acarretaram em um sistema com buscas mais rápidas e complexas, pois o modelo não dependia de um único registro para levar aos próximos. No entanto, o modelo de rede apresentava os mesmos problemas de seu antecessor, ao se deparar com grandes quantidades de dados, ou então na difícil manutenção das estruturas quando alterações eram solicitadas.

Banco de dados com modelo relacional

O modelo relacional se destaca pela simplicidade e flexibilidade, utilizando como base nos mesmo conceitos dos arquivos de dados. Os dados são armazenados em tabelas que, por sua vez, são compostas de campos e registros.



O banco de dados relacional foi criado com base na teoria matemática de conjuntos, e, por isso, seus componentes principais supracitados também podem ser conhecidos por seus correspondentes em linguagem matemática. Os campos podem ser chamados de atributos, os registros são chamados de tuplas, e as tabelas são conhecidas como relação.

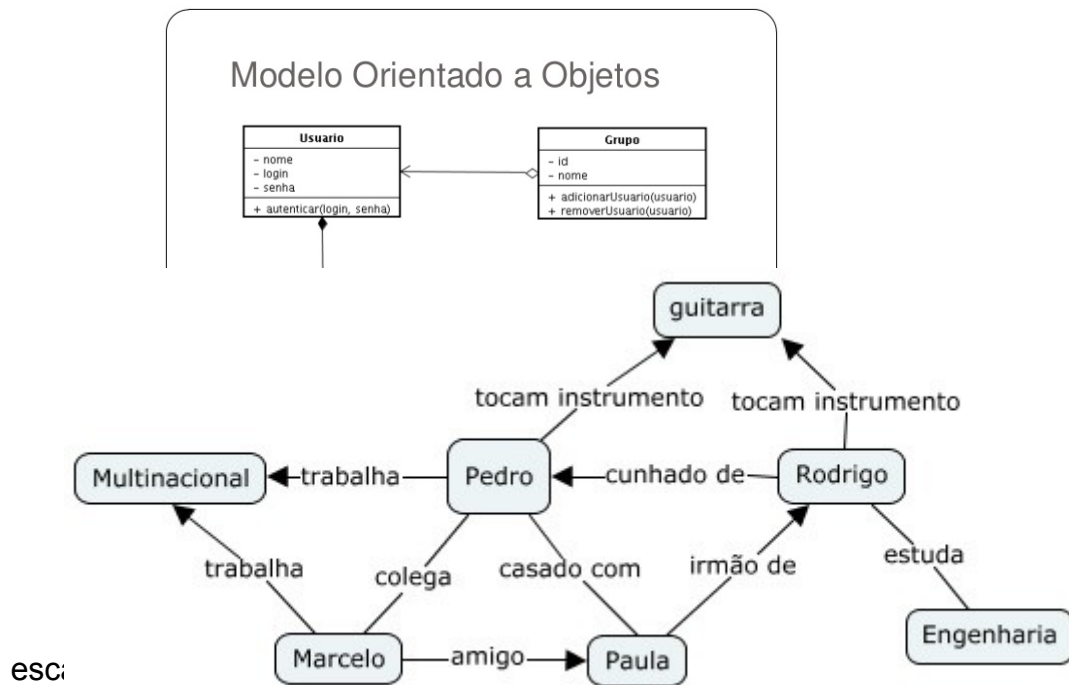
A estrutura pode se comparada a uma planilha eletrônica, com suas linhas (tuplas) e colunas (atributos). O conteúdo de um conjunto linha-coluna específico é denominado célula, tanto para a planilha eletrônica como para o banco de dados relacional. A célula é responsável por conter o valor do atributo que, por definição, deve ser um valor atômico.

Banco de dados Orientado a Objetos

Uma visão geral de Bancos de Dados Orientados a Objetos (BDOOs), e um exemplo de integração em aplicações Java através da implementação de um mesmo diagrama de classes em duas ferramentas (db4o e ObjectDB). Exploramos ainda as APIs e utilitários que elas fornecem para manipulação de objetos.

Para que serve:

Proporcionar uma alternativa à utilização de *frameworks* objeto-relacionais, unificando os modelos de classes e de dados de uma aplicação Java através da persistência dos próprios objetos manipulados.



esc:

promoveram uma alternativa de alto armazenamento com velocidade e grande disponibilidade, procurando se livrar de certas regras e estruturas que norteiam o Modelo Relacional. Se por um lado havia um rompimento das regras do Modelo Relacional, por outro lado havia ganho de performance, flexibilizando os sistemas de banco de dados para as diversas características que são peculiares de cada empresa. Esta flexibilidade passou a se tornar fundamental para suprir os requisitos de alta escalabilidade necessários para gerenciar grandes quantidades de dados, assim como para garantir uma alta disponibilidade destes, característica fundamental para as aplicações Web 2.0. Algumas grandes organizações passaram a investir em seus próprios SGBDs baseando-se na ideia do NoSQL.

MySQL Workbench

Desenvolvida exclusivamente para se trabalhar com o SGBD MySQL, é uma das ferramentas gratuitas mais conhecidas e mais utilizadas para quem trabalha com este SGBD, tanto para modelagem ER, como para execução de scripts SQL.

Sua principal característica é a facilidade de se trabalhar com o SGBD MySQL, onde seu foco está na modelagem física. É a ferramenta ideal para pequenos e médios projetos, bem como, para quem está iniciando ou já tem experiência com modelagem de banco de dados.

Entidade- Relacionamento

É uma maneira sistemática de descrever e definir um processo de negócio. O processo é modelado como componentes (entidades) que são ligadas umas as outras por relacionamentos que expressam as dependências e exigências entre elas, como: um edifício pode ser dividido em zero ou mais apartamentos, mas um apartamento pode estar localizado em apenas um edifício. Entidades podem ter várias propriedades (atributos) que os caracterizam. Diagramas criados para representar graficamente essas entidades, atributos e relacionamentos são chamados de diagramas entidade relacionamento.

Um modelo MER é normalmente implementado como um banco de dados. Nos casos de um banco de dados relacional, que armazena dados em tabelas, as próprias tabelas representam as entidades. Alguns campos de dados nestas tabelas apontam para índices em outras tabelas. Tais ponteiros representam relacionamentos.

Cardinalidade e exemplo

Nela são definidos o grau de relação entre duas entidades ou tabelas.

Uma das principais funções de cardinalidade, é manter a integridade do banco de dados, em associação com as regras de negócio, não permitindo que essas regras sejam quebradas causando anomalias no SGBD, dados repetidos ou fora de normalização. Essas associações são ligadas através de chaves (chave estrangeira e chave primária) que são registro de indexação que não se repetem e que podem ser usadas como um índice para os demais campos da tabela do banco de dados. Em chaves primárias, não pode haver valores nulos nem repetição. Em modelos de dados complexos, o relacionamento poderá ocorrer centenas de vezes, envolvendo dezenas de tabelas. O renomado cientista da computação, C.J. Date, criou um método sistemático para essa organização do modelos de banco de dados. Esse

modelo é o que conhecemos como normalização de dados. A normalização de dados permite-nos entre outras coisas, evitar anomalias em comando de delete e updates. Na vida real o processo de normalização de banco de dados consiste em dividir uma grande tabela com diversas colunas em tabelas menores.