

Administrador de Banco de Dados - Turma 2024A

1.3 Fundamentos de BD

Independência de Dados

Explica-se através da arquitetura de três esquemas.

- **Conceito:** capacidade de mudar um esquema em um nível de sistema de banco de dados sem que ocorram alterações do esquema do nível mais alto.

Dois tipos de independência de dados:

- **Independência lógica de dados:** capacidade de alterar o esquema conceitual sem mudar o esquema externo ou os programas.
- **Independência física de dados:** capacidade de alterar o esquema interno sem ter de mudar o esquema conceitual (nem o esquema externo, por consequência).

Modelos de Dados

Fornecem a abstração: omissão de detalhes do armazenamento de dados.

Os conjuntos de conceitos usados para descrever a estrutura de um banco de dados são: tipos de dados, relacionamentos, restrições, etc.

Categorias de Modelos de Dados

Classificam os modelos de dados de acordo com os tipos de conceitos que utilizam para descrever a estrutura BD

- **Conceituais:** Fornecem conceitos próximos à percepção dos usuários.
- **Lógicos:** Fornecem conceitos que descrevem como os dados são armazenados.
- **Físicos:** Fornecem conceitos de acordo com as regras de implementação.

Modelos Lógicos

Estruturados em registros de formatos fixos, de diversos tipos.

Cada tipo tem sua colocação de atributos.

Há linguagens para expressar consultas e atualizações.

Aqui nesta tabela há duas contas com o mesmo saldo, conta e saldo é o mesmo, com pessoas diferentes em cidades diferentes. Tem-se então uma redundância.

Nome	Rua	Cidade	Conta	Saldo
José	Figueiras	Campinas	900	55
João	Laranjeiras	Campinas	556	1.000
João	Laranjeiras	Campinas	647	5.366
Antônio	Ipê	São Paulo	647	5.366
Antônio	Ipê	São Paulo	801	10.533

Descrição da tabela:

Nome: José; Rua: Figueiras; Cidade: Campinas; Conta: 900; Saldo: 55.

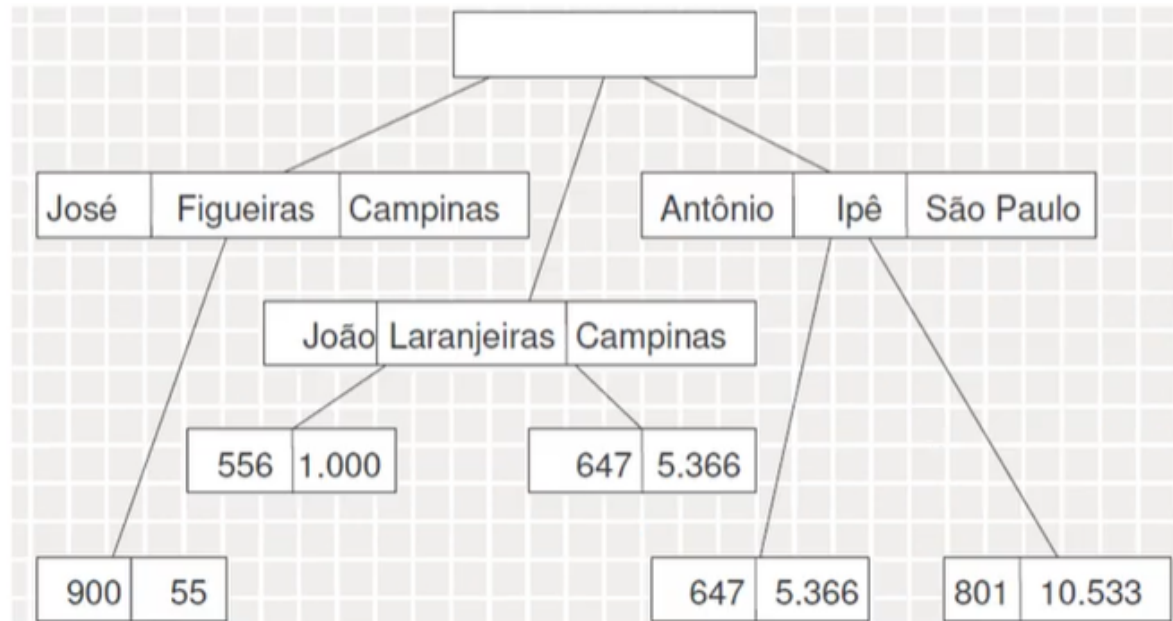
Nome: João; Rua: Laranjeiras; Cidade: Campinas; Conta: 556; Saldo: 1.000.

Nome: João; Rua: Laranjeiras; Cidade: Campinas; Conta: 647; Saldo: 5.366.

Nome: Antônio; Rua: Ipê; Cidade: São Paulo; Conta: 647; Saldo: 5.366.

Nome: Antônio; Rua: Ipê; Cidade: São Paulo; Conta: 801; Saldo: 10.533.

Hierárquico



Descrição: Esquema em que tem-se um retângulo em branco no topo central, deste retângulo, saem três linhas. Da esquerda pra direita, a primeira linha, se conecta a um retângulo dividido em três partes, sendo que cada uma delas tem algo escrito dentro, da esquerda pra direita tem-se [José], [Figueiras] e [Campinas]. Este retângulo se conecta por uma linha a um retângulo dividido em duas partes, na primeira está escrito [900] e na segunda [55]. A segunda linha que sai do retângulo central, se conecta a um retângulo dividido em três partes, a primeira, tem

escrito [João], a segunda, [Laranjeiras] e a terceira [Campinas], sendo que deste retângulo, saem duas linhas que se conectam a retângulos divididos em duas partes, sendo que no primeiro tem-se escrito nelas [556] e [56], já no segundo, [647 e 5.366]. Por fim, a terceira linha que sai do retângulo central, se conecta a um retângulo dividido em três partes, na primeira está escrito [Antônio], na segunda [Ipê] e na terceira [São Paulo], sendo que deste retângulo, saem duas linhas que se conectam a retângulos divididos em duas partes, sendo que no primeiro tem-se escrito nelas [647] e [5.366], já no segundo, [801] e [10.533].

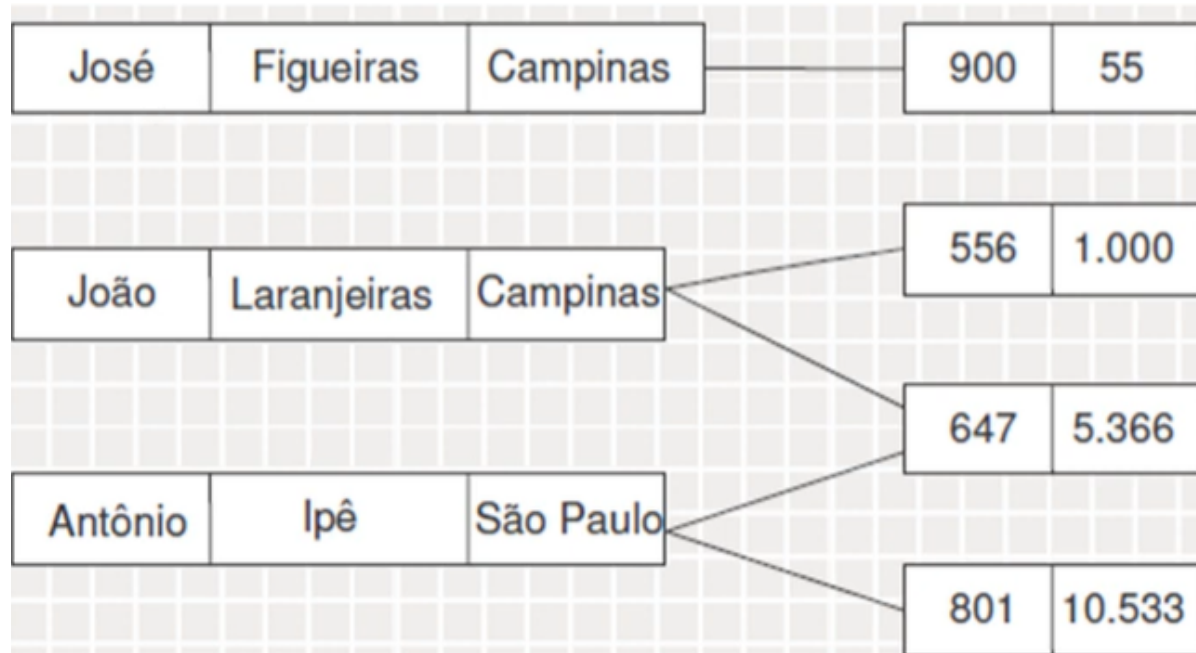
Neste exemplo mais uma vez ocorre uma redundância, pois dois correntistas dividem a mesma conta, e novamente é necessário demonstrar repetidamente duas vezes a mesma conta, para indicar que ela está ligada ao cliente João e também ao cliente Antônio, ao mesmo tempo, esta é uma outra forma de representação que acarreta demonstração redundante.

Os dados estão estruturados em hierarquias de árvores e sua estrutura utiliza as organizações de endereçamento físico dos discos

Os nós das hierarquias contém ocorrências de registros, onde cada registro é uma coleção de atributos, é um modelo orientado a registros, ou seja, operam um registro de cada vez.

Um registro pode estar associado a vários registros, porém ele deve ser replicado.

Em Rede



Descrição: Esquema em que tem-se dois esquemas. No do topo, da esquerda pra direita, se tem um retângulo dividido em três partes, sendo que em cada uma há algo escrito, na primeira [José], na segunda [Figueiras] e na terceira [Campinas]. Deste retângulo, sai uma linha pra direita que o conecta a um retângulo dividido em duas partes, na primeira, tem-se escrito [900] e na segunda, [55]. No esquema de baixo, no topo superior esquerdo, tem-se um retângulo dividido em três partes, na primeira tem-se escrito [João], na segunda [Laranjeiras] e na terceira [Campinas]. Este retângulo se conecta a dois retângulos, por uma linha cada. Os dois estão divididos em duas partes, sendo que no de cima tem-se escrito [556] na primeira e [1.000] na segunda, já no de baixo, tem-se [647] na primeira e [5.366] na segunda. No topo inferior esquerdo, tem-se um retângulo dividido em três partes, escrito na primeira [Antônio], na segunda [Ipê] e na terceira [São Paulo]. Este retângulo se conecta ao retângulo [647 | 5.366] e a um retângulo dividido em duas partes, na primeira escrito [801] e na segunda [10.533].

É uma extensão do modelo hierárquico, porém elimina o conceito de hierarquia, visto que os registros estão organizados e em grafos.

Permite que um mesmo registro esteja envolvido em várias associações. Também opera um registro de cada vez, por ser orientado a registros.

Modelo Relacional

Tabela Cliente (dados)

Cód-cliente	Nome	Rua	Cidade
015	José	Figueiras	Campinas
021	João	Laranjeiras	Campinas
037	Antônio	Ipê	São Paulo

Descrição da tabela:

Cód-cliente: 015; Nome: José; Rua: Figueiras; Cidade: Campinas.

Cód-cliente: 021; Nome: João; Rua: Laranjeiras; Cidade: Campinas.

Cód-cliente: 037; Nome: Antônio; Rua: Ipê; Cidade: São Paulo.

Tabela

Conta

(dados)

Nro-conta	Saldo
900	55
556	1.000
647	5.366
801	10.533

Descrição da tabela:

Nro-conta: 900; Saldo: 55.

Nro-conta: 556; Saldo: 1.000.

Nro-conta: 647; Saldo: 5.366.

Nro-conta: 801; Saldo: 10.533.

Tabela Cliente- Conta (relacionamento)

Cód-cliente	Nro-conta
015	900
021	556
021	647
037	647
037	801

Descrição da tabela:

Cód-cliente: 015; Nro-conta: 900.

Cód-cliente: 021; Nro-conta: 566.

Cód-cliente: 021; Nro-conta: 647.

Cód-cliente: 037; Nro-conta: 647.

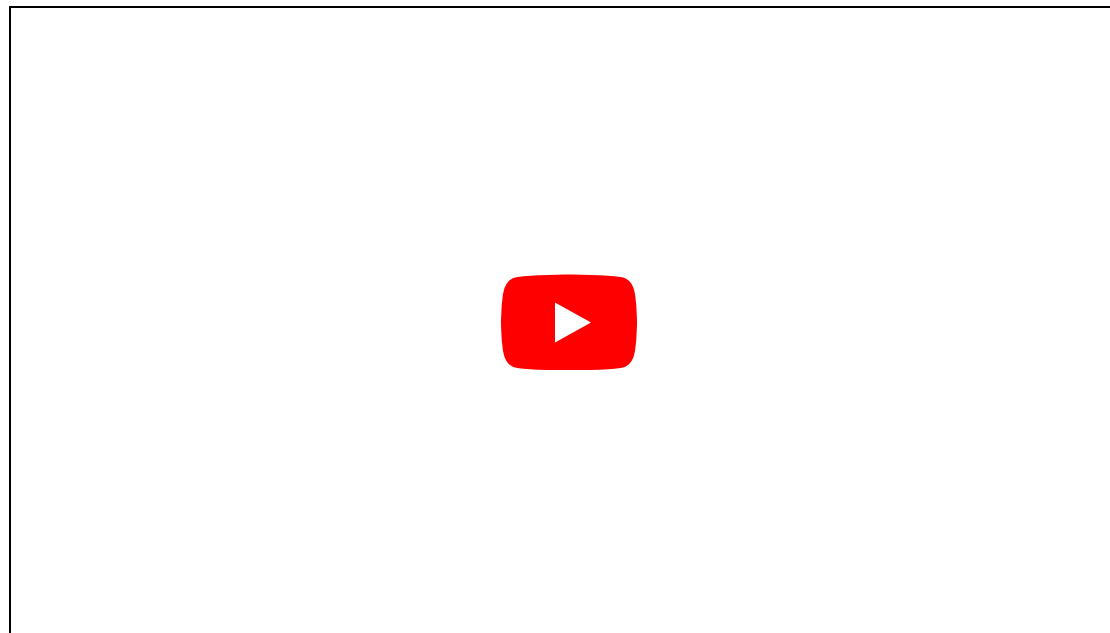
Cód-cliente: 037; Nro-conta: 801.

O modelo relacional é composto por um conjunto de tabelas ou relações, onde cada tabela é um conjunto não-ordenado de linhas (tuplas), que por sua vez, são uma série de campos (atributos)

Simplicidade dos conceitos básicos e poder dos operandos de manipulação.

Cada campo possui um nome e o conjunto de campos que possuem um mesmo nome formam uma coluna na tabela.

Abaixo, segue a aula que fala sobre o que vimos até agora.



▼  Transcrição do vídeo

[O vídeo acima é do curso Administrador de Banco de Dados. Apresentado pelo professor Róger Sá da Silva e abordará o conteúdo Bancos de dados 1. O vídeo é uma apresentação de slides e a voz do professor narrando as explicações].

Olá pessoal, tudo bem, começando mais uma aula de banco de dados 1, a primeira aula desta sequência, a gente vai abordar o que exatamente nessa primeira aula? Os fundamentos de banco de dados para gente conseguir compreender, depois a sequência de conteúdos que a gente precisa ver a partir de projeto e modelagem de banco de dados durante essa disciplina.

O que são os bancos de dados? É isso que está aparecendo na foto, bancos formados por dados? Claro que não, os bancos de dados estão no contexto que essa figura representa, onde todos os meios, nosso computador, dispositivos móveis, automóveis, roteadores de casa, equipamento de internet, a nuvem, pagamentos eletrônicos, inclusive documentos físicos, cartas, e-mail, planilhas, relatórios, chips, tudo isso em alguma medida vai precisar armazenar dados, então o banco de dados quanto conceito está vinculado a diversas áreas da computação, senão todas.

Para a gente dar o passo inicial é preciso definir alguns conceitos, o primeiro é o conceito de dado, que é a representação da informação, ou seja, o é um dado? Ele é um fato do mundo real que está registrado e possui algum significado implícito no domínio daquela aplicação, por exemplo, uma data num cadastro de pessoa pode ser uma data de nascimento, pode ser a data de cadastro ou pode ser a data que um empregado ou um funcionário entrou na empresa. Se for um estudante pode ser a data de matrícula, então vejam que o dado data ele é um fato, é uma data de um calendário que todo mundo segue e o significado dele vai depender do contexto do domínio da aplicação. Já informação é o significado desse dado, aqui sim é um fato útil que pode ser extraído direta ou indiretamente a partir de um dado, então vejam que a gente pode ter um dado que é a data de nascimento e a partir daí num cadastro de pessoas por exemplo, e a partir daí eu consigo extrair uma informação, significado desse dado, por exemplo, em alguma data, hoje, amanhã, mês que vem, qual é a idade dessa pessoa, então se extrai uma informação que é a idade a partir de um dado que a data de nascimento, esses conceitos é bom que a gente tenha em mente porque eles vão ser precisos ao longo da disciplina.

O que são bancos de dados? Agora que a gente sabe o que é dado, dependendo dos livros que a gente vai seguir cada um deles conceito de uma forma diferente, uns trazem que é o equivalente eletrônico de um armário de arquivamento, outros chamam que é uma coleção de dados relacionados, o que eu prefiro, que eu acho que tem mais a ver com o curso, é uma coleção de dados inter-relacionados representando informações sobre um domínio específico, por que eu prefiro essa? Porque a gente geralmente parte de um domínio, de um cenário para começar a modelar e projetar um banco de dados, esses dados vão representar situações e informações que tem significado neste domínio, eu prefiro esse último conceito para um melhor entendimento.

O que é um sistema de banco de dados? São um sistema que está em um computador, computadorizado que faz a manutenção desses registros, ou seja, ele mantém os dados e torna disponíveis quando a gente solicita para fazer alguma consulta, ele me permite armazenar e manter ao longo do tempo esses dados e eles estão disponíveis quando eu preciso. Então, este sistema de banco de dados, ele é formado

por várias partes, pelos dados em si, por conjuntos de softwares o gerenciador e por alguns programas e aplicativos que funcionam junto, o hardware que no fim do dia armazena em si os dados, e as pessoas, usuários ou programadores técnicos, administradores, também fazem parte do que a gente chama de um sistema de banco de dados, tudo isso é algo sistêmico que funciona de forma sistêmica.

Aqui está uma figura que representa mais ou menos isso, o sistema de banco de dados, esse quadrado mais externo, tem os usuários e programadores interagindo como parte desse sistema, esses programas de apoio que fazem essa interface com o software SGBDs, que é o sistema gerenciador de banco de dados. Então, geralmente, tem um software interface com esse SGBDs, muitas vezes fornecidos pelo próprio SGBDs, dentro é a parte mais, digamos, escondida, para o usuário e para o programador a não ser a pessoa que trabalha de fato com a administração de banco de dados mais a fundo. Aí tem os processadores, com que ele utiliza as consultas quando eu preciso consultar dados, como ele faz esses acessos, às bases físicas em si, esses cilindros recortados é a representação gráfica da parte física, são os dados armazenados em si e a definição dos dados armazenados. A gente vai ver isso ao longo, que a gente precisa definir que dados são armazenados para depois eu consigo armazená-los em si, todo esse conjunto é uma figura que representa um sistema de banco de dados, uma coisa genérica, o conceito de sistema de banco de dados.

Então, ele tem os dados, que é o que está armazenado em si, eles são integrados, todos reunidos de uma forma integrada, são passíveis de serem compartilhados, vocês viram, usuários acessando e etc. O software, que é a camada entre os usuários e os dados, tem essas duas partes que eu comentei, os aplicativos que fazem a interface direta, e o gerenciador que é o software, digamos mais pesado que de fato gerencia e coordena acessos, coordena as questões de segurança e de integridade desses dados. O hardware, onde estão literalmente armazenados, um disco é um SSD, um HD etc, da máquina onde aquele sistema de banco de dados está rodando, e as pessoas que têm os administradores, que é quem tem mais alto nível de acesso às partes mais internas do SGBDs. Os programadores de aplicativo que seriam digamos, nós em uma medida que interagimos através do gerenciador para configurar e formatar o banco de dados para uma aplicação nossa, um sistema que a gente esteja desenvolvendo, e também tem a ideia de usuário final, que vai acessar o banco de dados ou via aplicativos, feitos por alguém, ou até inclusive via gerenciador eventualmente se preciso for, dependendo do tipo de cenário de situação.

Além disso, tem diversos programas utilitários que a gente chama, ou ferramentas que um sistema de banco de dados vai trabalhar junto, então, tem ferramentas eventualmente de carga, carga inicial, quando tem o banco de dados zerado eu preciso dar a carga nele, várias informações cadastrais que eu já possuo no cenário que eu estou trabalhando, de backup e de retomada dos dados, de conseguir fazer para salvar cópias de segurança, tempo. Utilitários para reorganizar os dados em alguma medida, com o tempo eu preciso fazer uma reorganização para utilizar o acesso aos dados quando eu tenho muitas e muitas e muitas dados e tabelas e coisas diferentes ao longo do tempo armazenado, tenho software para estatística que fazem análises também sobre esses dados, e os dicionários de dados que são aquilo que eu falei, os dados precisam ter algum significado alguma explicação, a gente precisa definir algumas coisas quando vai

armazenar os dados para que o SGBDs consiga manter eles e manter a integridade deles ao longo do tempo usando, alguns desses se não a maioria, os próprios sistemas gerenciadores de banco de dados já fornecem várias dessas ferramentas dentro deles mesmos.

Os bancos de dados evoluíram em quatro gerações, o que a gente entende, na primeira geração bem lá atrás, não importa a data agora, é o processamento eram chamados arquivos independentes, então vejam, cada cilindrinho é um arquivo, que tem o cadastro, então eu cadastrar projetos e se eu queria extrair as relações de projetos eu tinha que extrair um relatório a partir desse arquivo independente, outro arquivo de cadastro horas de trabalho, outro arquivo de cadastrar empregados para depois conseguir emitir a folha, vejam nada integrado, tudo separado. Independente, com o tempo, na segunda geração houve um passo adiante, que é integrar, eu tenho um arquivo integrado e aí todos os processos acessam esse arquivo integrado e emitem os relatórios, e na terceira geração a gente chega bem mais próximo do que a gente trabalha hoje, que a chamada geração do processamento da informação, ou seja, a gente começa a mapear conceitos que a gente trabalha intrinsecamente ao banco de dados, então vocês vão ver ao longo dessa disciplina que a gente começa a modelar algumas coisas e começa a trabalhar com qual é o dado que é identificador no cadastro de pessoas? Qual é o dado que é liquidificador no cadastro de clientes? algum dos dados que eu armazeno de um cliente é o identificador daquele cliente que eu uso para localizar aquele cliente, a gente começa a mapear conceitos para dentro do banco de dados, depois vocês vão entender de forma mais tranquila isso, a quarta e última geração, onde os SGBDs dão um grande salto porque o hardware permite isso, como aconteceu com a computação geral é o chamado geração do processamento ampliado, que começou a ver a ampliação das aplicações de sistemas de banco de dados, aí os SGBDs vieram com tudo, a gente tem todas essas ferramentas e essas funcionalidades que eu tenho tratado com vocês de forma muito contundente na quarta geração, que a geração que a gente hoje trabalha e vai trabalhar aqui.

Quais são as características de um sistema de banco de dados? Tem a natureza alto descritiva do sistema, ou seja, os próprios dados e o que eu estou armazenando descrevem esse sistema de banco de dados, a um isolamento entre os programas e os dados, vocês viram naquele esqueminha que os dados ficam isolados do problema, eu posso eventualmente atualizar algo do meu sistema de banco de dados e mantém os dados armazenados, garantidos sem problema algum, ele dá suporte para múltiplas visões dos dados, ou seja, um administrador tem uma visão dos dados com mais detalhes e mais acessos, um programador um pouco menos e usuário final menos ainda, só tem a visão do que o sistema ou software que ele tá usando e demonstra os dados para ele permite, é isso que veio também como uma característica dos sistemas de banco de dados, compartilhamento de dados e processamento de transações de multiusuários, ou seja, a ideia de que todos acessem os mesmos dados e quando alguém está editando, por exemplo, um registro e alguém pede para visualizar aquele registro, como que eu coordeno isso? mostro o registro com o dado antigo? não mostro o registro para aquele usuário que pediu para ver? primeiro atualizo e depois mostro para ele o registro já atualizado? Como é que eu organizo essas transações? mesma coisa de uma conta corrente, estou fazendo depósito e alguém pede para ver o saldo, aguardo para liberar o saldo antes ou depois do depósito para pessoa que tá pedindo para consultar o saldo? Essas transações são organizadas pelo sistema de banco de dados.

Então de forma resumida, porque usar banco de dados? uma, a gente resolve uma questão de densidade, não há mais necessidade de arquivos, antigamente pessoal um arquivo de papel, aqueles de gavetas, aquilo era um banco de dados analógico, um banco de dados de registro de coisas que uma empresa tinha, cadastro de clientes, vendas, emissão de notas fiscais, era no papel depois os arquivos Independentes também eram de forma digital mas ainda tinha uma densidade muito grande, hoje a gente tem isso, mas o SGBDs máscara em alguma medida, então eu, usuário administrador ou usuário não preciso lidar com a densidade dos arquivos de um banco de dados, o SGBDs resolve isso para mim, velocidade nem se fala, rapidez em obter e realizar os dados, agilidade, às tarefas mecânicas realizadas por uma máquina são muito mais ágeis do que qualquer busca manual feita por alguém, um ser humano buscando em arquivos individuais, catalogados da melhor forma possível a máquina vai sempre ganhar numa busca de algum dado que alguém deseja consultar, atualidade, as informações precisam estar atualizadas e um banco de dados permite que eu as mantenha assim, e a proteção, eu consigo proteger os dados contra perda e o acesso ilegal, tudo isso o banco de dados me garante e são a grande motivação que ele tem começado a ser utilizado cada vez mais, agora é algo meio difundido e não a quem não utilize.

Ele tem uma grande característica que é o controle Centralizado, é muito importante em ambientes que tem multiusuários que são basicamente todos os SGBDs que a gente usa. Papéis de destaque desse controle Centralizado, eu tenho o administrador de dados, esse cara deve entender dos dados, ele deve decidir quais dados armazenar quando eu estou fazendo um projeto de um banco de dados, já o administrador de banco de dados é o cara que cria os bancos de dados, define as estruturas, implementa controles, assegura o desempenho, então vejam que esse segundo administrador ou administrador de banco de dados é necessariamente alguém da área da tecnologia porque é ele quem administra o sistema de banco de dados em si, porém ele precisa do apoio do administrador de dados, que é a pessoa que entende dos dados, do contexto do domínio da aplicação que ele tá fazendo, então sendo uma empresa, eu vou fazer uma parte de controle de cadastro de clientes, de vendas e etc, tem que falar com o pessoal do comercial, das vendas para entender os dados que eles têm para eu conseguir criar o banco, definir as estruturas e implementar controles necessários de acordo com a necessidade, eventualmente pode ser a mesma pessoa, o administrador de banco de dados pode ter escutado bastante o contexto, cenário que está trabalhando e começa a entender os dados e depois ele faz os dois papéis.

Vantagens disso tudo, os dados podem ser compartilhados, eu reduzo e muito a redundância, então não corro risco de armazenar dados duplicados, duas vezes, ou de forma recorrente, evito, possibilito evitar inconsistência, garanto que a qualidade dos dados vai ser Ok, ele dá o suporte a transações, essa questão de estou atualizando quem pede para acessar, quem faz primeiro, se não realizar essa atualização não faz outra, isso tudo, esse controle de transações é ele quem me permite, consigo manter a integridade, ou seja, aquilo que eu inseri, aquilo que eu garantir, configurei, fica garantido ao longo do tempo, a segurança vai ser reforçada, eu consigo ter controle de acesso, diferentes níveis de acesso para diferentes tipos de usuário, se eu tenho algum requisito contraditório um com outro eu consigo equilibrar

isso dentro do meu sistema de banco de dados com controle centralizado, e eu consigo impor padrões, que especialmente o que me garante a integridade ao longo do tempo, eu consigo que uma data tenha sempre um padrão, senão não aceita, não armazena, que um CPF tem sempre o mesmo padrão, que algo numérico sempre obedeça um certo padrão, que questões de somatório, de totalização eu consiga sempre verificar e não guardar totais ou somatório errados, e etc, tudo isso funciona, como que a gente chama, arquitetura de três esquemas, ela foi proposta para que a gente consiga realizar e visualizar todas essas características que a gente está comentando, Qual é o objetivo principal dessa arquitetura de três esquemas? é separar o usuário da aplicação, que está de fato manipulando, do banco de dados físico, que é o que está armazenado em disco, como é que se separa o usuário? através de três níveis de abstração dos dados, por isso os três esquemas, eu tenho três níveis de abstração o nível de visão do usuário, nível conceitual e o nível físico, vejam, aqui os usuários finais têm uma visão externa no nível externo, aqui no meio eu tenho um esquema conceitual, que é basicamente quem vai fazer esse mapeamento da parte interna do esquema interno para a visão externa dos usuários, esse esquema interno obviamente está no nível interno, que é o banco de dados armazenados, são os dados armazenados, através desses três níveis de abstração que a gente trabalha.

Nível de visão do usuário, é o mais próximo do usuário, ele vai selecionar a forma como os dados são vistos pelos usuários individuais, de fato é a visão dos usuários, como ele vê os dados. Nível conceitual, ele representa a abstração do banco de dados, partindo de algumas estruturas que vão descrever quais dados estão armazenados e como eles se relacionam, mas não quer dizer que é como ele está armazenado e relacionado nos discos fisicamente numa máquina, ele é uma forma já de abstração disso, eu vejo de uma forma que não é a realidade física, mas por quê? Porque através destes conceitos eu consigo compreender e modelar o banco de dados e não preciso entender do nível físico, que é um nível mais difícil e complexo. Por fim, eu tenho esse nível físico, que é o mais próximo ao armazenamento físico em si, e essa sim é a forma como os dados são realmente armazenados e acessados fisicamente do disco de uma máquina.

Isso tudo gera uma independência de dados, ela é oriunda dessa arquitetura de três esquemas que criou esses níveis diferentes de abstração, o que são a independência de dados? É a capacidade de eu conseguir mudar o esquema em nível sem que ocorra alterações desse esquema no nível mais alto, então os usuários estão enxergando uma visão externa através do meu nível conceitual, eu posso mudar o esquema interno eventualmente? mudei o disco, mudei a forma de armazenamento mas o meu esquema conceitual que faz esse mapeamento se mantém o mesmo e os usuários continuam tendo a mesma visão dos dados e não fazem ideia que houve uma mudança no esquema interno no nível interno, isso é a chamada independência de dados que essa arquitetura de três esquemas promove, eu tenho esses dois tipos, a independência a lógica que eu mudo o esquema conceitual sem mudar o esquema externo ou a dependência física, que eu consigo mudar o esquema interno sem ter que mudar o esquema conceitual e por consequência nem o esquema externo que é mais da visão do usuário.

Por fim, toda essa questão de banco de dados é calcada em modelos de dados, porque eu só consigo fazer essa abstração e especialmente

aqueles modelos conceituais, se eu partir de um modelo que vai fornecer essa abstração, ou seja, a omissão de detalhes do armazenamento de dados, então modelo de dados é o conjunto de conceitos que eu vou usar para descrever a estrutura de um banco de dados, eu vou dizer que tipos de dados eu trabalho, como esses dados se relacionam, se ele tem alguma restrição de preenchimento, de ter que existir ou não, de poder ser nulo e etc, eu consigo aquele modelo conceitual, consigo aquela abstração através de um modelo de dados, e existem alguns modelos, eu tenho categorias para esses modelos, eu tenho alguns tipos de conceito que eu consigo descrever uma estrutura de um banco de dados, tenho os tipos de modelo de dados conceituais, lógicos e físicos, vamos ver rapidamente com eles são conceituados.

Os modelos conceituais fornecem conceitos próximos a percepção do usuário, conceitos físicos vão fornecer conceitos que descrevem como os dados são armazenados, os conceitos lógicos vão fornecer conceitos de acordo com as regras de implementação, como esse banco de dados é implementado na prática, então vejam que eu tenho modelos de dados para diferentes níveis de abstração e aí eu vou usá-los de acordo com as necessidades quando eu tenho que trabalhar com algum daqueles níveis de abstração, os modelos lógicos eles são estruturados em registros de formato fixo, de diversos tipos, cada tipo deles tem uma coleção de atributos e eu vou ter linguagem para expressar consultas e atualizações, aqui eu tenho uma tabela, tem sempre esse mesmo formato fixo e cada tipo tem uma coleção de atributos, então, nome, rua, cidade, conta e saldo, vejam que nesse modelo lógico tenho duas contas com o mesmo número e o mesmo saldo que indicam que é a mesma conta, possivelmente essas duas pessoas têm uma conta conjunta, eu só consigo ver isso nesse modelo aqui ao observar que conta e saldo é o mesmo, são pessoas diferentes com endereços diferentes em cidades diferentes, porém compartilham o mesmo número de conta, vejam que esse modelo promove uma redundância, eu preciso armazenar duas vezes as informações para conseguir demonstrar o relacionamento de que dois correntistas compartilham uma conta corrente, já o modelo hierárquico ele usa esse formato de hierarquia, então partindo do que seria por exemplo um banco eu tenho o cadastro dos clientes, que ligam hierarquicamente a esse banco, então o José da Rua Figueiras na cidade de Campinas, o Antônio na Rua Ipê de São Paulo, tenho três clientes vinculados, e ligado a cada cliente eu tenho uma conta com o número da conta e o saldo, vejam que ligado ao João eu tenho a conta 647 e ligado ao Antônio eu tenho a conta 647 e cada um deles, por outro lado, também tem uma outra conta corrente 556 [está ligada ao João e] 801 [está ligada ao Antônio], com saldo tal [cada uma delas], mas de novo eu tenho uma redundância, preciso demonstrar duas vezes a conta corrente para indicar que ela está ligada ao João e ligada ao Antônio ao mesmo tempo, essa é uma outra forma de representação que também me demanda uma redundância, então eles trabalham hierárquico a essas estruturas em árvores, em cada registro é uma coleção de atributos, como eu falei, para ele estar associado a vários registros eu tenho que replicar aquela conta para poder vincular a mais de um cliente.

Eu tenho esse outro modelo lógico em rede, que é parecido com o hierárquico mas é uma rede sem a hierarquia em si, e vejam que aqui em alguma medida eu represento uma vez só conta 647 e faço duas ligações uma para cada um dos clientes e consigo representar uma vez só

a conta mas indicando que ela está vinculada a dois clientes diferentes, ela é uma extensão do hierárquico, porém sem o conceito de hierarquia, porque os registros estão organizados, como a gente chama, em grafos, depois vocês vão estudar Isso em estrutura de dados, ele permite que o mesmo registro, como eu mostrei, tenha várias associações, por fim, eu tenho o modelo relacional, que eu tenho tabelas para armazenar os dados, Imaginem como se fosse as planilhas do Excel, para início de conversa, vamos ver aqui, o João e o Antônio, para cada um deles eu criei um código de cliente, 021 João, 037 é o Antônio e tem um cadastro só dos clientes, código, nome, rua e cidade, tem um segundo cadastro de dados da tabela conta, que é só o número da conta e o saldo, então um cadastro Independente de clientes e um cadastro Independente de conta, mas professora e a gente perdeu qual conta é de qual cliente? não, neste modelo relacional eu tenho uma terceira tabela, que é a tabela que faz o relacionamento, através do código do cliente e do número da conta eu faço os pares de quem é de quem, a conta 900 é do cliente 0 15, então a conta 900 é do José, a conta 556 é do 0 21, então a conta 556 de saldo mil é do João, o João 021 tem outra conta, a 647 que é aquela conta compartilhada, como eu sei que é compartilhada? esta duas vezes o código da conta, o número da conta, vinculada a dois código de cliente o 021 e o 037, 021 João, 037 Antônio, então vejam que eu não precisei replicar, a única questão é esse número identificador da conta e o código identificador de um cliente, mas eu posso manter os cadastros todos individuais Independentes de clientes e contas e usar só essa tabela para montar o par de quem se relaciona com quem, e se um cliente tiver mais de uma conta, eu jogo duas contas, e se uma conta for de dois clientes eu jogo duas vezes ou três o número da conta e vínculo os códigos de cliente relacionados a essa conta, esse relacional é o conjunto de tabelas, cada tabela é um conjunto não ordenado de linhas, cada uma dessas linhas tem um monte de campos, nome, rua, cidade, para o cadastro de clientes por exemplo, aqui eu tenho os conceitos mais básicos e eu consigo usar o poder dos operandos de manipulação, depois a gente vai ver com calma isso quando a gente estudar o modelo relacional mais adiante, por hora esses conceitos essenciais ficam bem explicados dessa forma, realizam leituras complementares sobre o conteúdo, façam a atividade proposta na sequência, tá bom pessoal, valeu, muito obrigado.

Referências:

DATE, C. J. Introdução aos sistemas de Banco de Dados. 8. Ed. Rio de Janeiro: Campus, 2004.

ELMASRI, R. e NAVATHE, S. Sistemas de Banco de Dados. São Paulo: Pearson/Addison Wesley, 2011.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. Sistema de banco de dados. São Paulo: Elsevier, 2012.

Última atualização: terça, 27 fev 2024, 09:30

◀ 1.2 Fundamentos de BD

Seguir para...

1.4 Sistemas Gerenciadores de Banco
de Dados ▶