

Questões básicas sobre os BDs

Agenda:

- Vantagens da abordagem de utilização de SGBDs;
- Conceitos e Arquitetura



Questões básicas sobre os BDs

- **Vantagens da abordagem de utilização de SGBDs:**
- **--Controle de redundância**
- Dicionário>Qualidade do que é repetitivo e supérfluo: subir para cima é redundância!
-
- A situação real:
- Cada software utilizando processamento de arquivos mantém seus próprios arquivos, de maneira independente.
- Os mesmos dados, ou de um mesmo domínio, estão guardados em locais diferentes, o que complica a tarefa de atualização ou manipulação destas informações.

Questões básicas sobre os BDs

- **Vantagens da abordagem de utilização de SGBDs:**
- **--Controle de redundância**

id	nm	codigo
01	Aluno 1	1010
02	Aluno 2	2020
03	Aluno 3	3030

matricula	nm	data
131415	Aluno 1	10-10
131516	Aluno 2	11-10
131617	Aluno 3	09-10

disciplina	nm	nota
BD1	Aluno 1	8,9
BD2	Aluno 2	7,0
ED1	Aluno 3	10,0

id	disciplina	codigo
01	BD1	1011
02	BD2	2021
03	ED1	3031

Questões básicas sobre os BDs

- **Vantagens da abordagem de utilização de SGBDs:**
- **--Controle de redundância**
- A redundância é, então, o fato de guardarmos a mesma informação em lugares diferentes!!!
- Isso gera muitos problemas:
 - Preocupação com atualizações lógicas;
 - Trabalhos duplicados;
 - Desperdício de espaço;
 - Inconsistência dos dados repetidos.

Questões básicas sobre os BDs

- **Vantagens da abordagem de utilização de SGBDs:**

- **--Controle de redundância**

- É bastante comum que a redundância não seja intencionalmente criada pelo projetista do banco de dados, e sim por um descontrole das fases de levantamento de requisitos e demais fases do projeto do sistema:

- Diferentes grupos de usuários apresentam visões diferentes para elementos muito parecidos;
- Erros na modelagem → erros na implementação;
- Problemas na identificação de dados;

Necessário se ter técnicas apropriadas para integrar as visões diferentes.

Questões básicas sobre os BDs

- **Vantagens da abordagem de utilização de SGBDs:**

- **--Controle de redundância**

- Exemplo de redundância gerada pela visão duplicada de um mesmo fato:

- Em um projeto de sistema de informação para uma empresa de varejo, os stakeholders descrevem seus pontos de vista:

- Depto Mkt: “Precisamos controlar todas as vendas, relacionando as categorias de produtos aos respectivos clientes, para que possamos criar campanhas específicas e categorizadas à nossos clientes”;
- Depto Financeiro: “Precisamos registrar todas as vendas, pois precisamos ter registro de todos os produtos vendidos, com dados que possibilitem a identificação da venda para efeitos fiscais e de faturamento”;
- Depto de Estoque: “Precisamos saber de todas as saídas de produtos, para sabermos quais produtos temos em estoque”.

Questões básicas sobre os BDs

- **Vantagens da abordagem de utilização de SGBDs:**
- **--Controle de redundância**
- Controlando-se a redundância, busca-se obter consistência e otimização do sistema. Porém, em alguns casos, este trabalho gera complicações na utilização do BD pelo sistema.
- Para tanto, é necessário a construção de modelos de dados com uso de **redundância controlada**, de forma a resguardar os pontos positivos, mas garantindo boa execução de manipulações de consumo do BD.
- O SGBD deve ter capacidade de controlar certas redundâncias, impedindo inconsistências entre os arquivos. Cabe ao projetista incluir restrições às redundâncias.

Questões básicas sobre os BDs

- **Vantagens da abordagem de utilização de SGBDs:**
- **--Restrição de acesso não autorizado:**
- Garantir que somente tenha acesso aos dados aqueles que possam realmente ter acesso à eles: **confidencialidade**.
- Ex.: dados de movimentação financeira de uma conta bancária.
- O SGBD deve garantir um subsistema de autorização, usado pelo DBA para criar contas e definir restrições para cada uma.
- Parecido com o controle de contas de um SO.

Questões básicas sobre os BDs

- **Vantagens da abordagem de utilização de SGBDs:**
- **--Armazenamento persistente para programas:**
- Armazenamento de dados gerados/capturados pelos programas e estruturas de dados, garantindo que estes não se percam ao desligar do programa e descarregamento das variáveis.
- Variáveis de uma linguagem de programação ficam na memória RAM, e são voláteis.

Questões básicas sobre os BDs

- **Vantagens da abordagem de utilização de SGBDs:**
- **--Armazenamento de estruturas para processamento de consultas:**
- Tão importante quanto dar suporte para se guardar dados, um SGBD só terá funcionalidade plena se possibilitar formas de se recuperar tais informações.
-
- Pelo fato de termos os dados armazenados em disco, um SGBD deve possuir estruturas de dados especializadas para gerar velocidade de pesquisa em disco para recuperação da informação desejada.
- O SGBD deve possuir um módulo para gerenciamento e escalonamento de comandos de busca:
- Índices, buffering, processamento de consulta e otimização.

Questões básicas sobre os BDs

- **Vantagens da abordagem de utilização de SGBDs:**
- **--Backup e recuperação:**
- Um SGBD deve prover facilidades para a restauração de falhas de hardware ou de software.
- O módulo de backup e recuperação é o responsável por isso.
- Se um sistema de computador falhar no meio de uma transação complexa, este módulo deve garantir que o banco de dados seja realocado no mesmo estado em que estava.
- Alternativamente o módulo pode garantir que retorne exatamente no ponto onde parou.

Questões básicas sobre os BDs

- **Vantagens da abordagem de utilização de SGBDs:**
- **--Representação de relacionamentos complexos de dados:**
- Bancos de dados reais possuem a representação de dados que se relacionam para formar informações muitas vezes de forma complexa.
-
- O SGBD deve ter a capacidade de representar a variedade de relacionamentos complexos que o sistema demandar, bem como garantir que os dados sejam guardados e recuperados dentro desta complexidade de representatividade.
- Ex.: moléculas associadas às alterações em cromossomos do DNA de um indivíduo adulto de determinada faixa etária e possíveis causas de câncer.

Questões básicas sobre os BDs

- **Vantagens da abordagem de utilização de SGBDs:**

- **--Forçar restrições de integridade:**

- Integridade significa confiabilidade!!!

- A maioria dos sistemas modelados possuem restrições de integridade, ou seja, fatos que precisam ocorrer para que certa informação possa existir.

- O SGBD deve prover funcionalidades para a definição e a garantia dessas restrições.
- Exemplos de restrições: - impedir que um texto seja inserido no campo de data de nascimento;
- - Impedir que um filho seja cadastrado sem seu pai/mãe;
- - Impedir que um produto seja vendido sem ter sido cadastrado;
- - Impedir que um dado seja excluído após ter sido referenciado.

Questões básicas sobre os BDs

- **Vantagens da abordagem de utilização de SGBDs:**
- **--Inferências e Ações Usando Regras:**
- Alguns SGBDs fornecem capacidades para definir regras de dedução por inferência gerando novas informações de fatos armazenados no BD: sistemas de banco de dados dedutivos.
- Ex.: regras complexas no minimundo de um sistema escolar que determina se um aluno está reprovado. Isso pode estar no programa, no seu algoritmo, mas pode ser um recurso do BD, uma regra mantida pelo SGBD, dando capacidade de processamento ao SGBD.
- Dados armazenados *vs* dados processados.

Conceitos e Arquiteturas

• Evolução

- A arquitetura evoluiu de pacotes de softwares SGBDs monolíticos, onde era um único sistema fortemente integrado, para modernos conjuntos de softwares modulares, com arquitetura cliente/servidor.

Por que??

- Computadores de grande porte e centralizados foram sendo substituídos por vários PCs e estações de trabalho, conectados via redes de comunicações a vários servidores:
 - Servidor Web, servidor de banco de dados, servidor de arquivos, servidor de aplicações.
 -
 - Duas partes: Módulo cliente e Módulo Servidor.

Conceitos e Arquiteturas

Modelos de dados, esquemas e instâncias

- Uma característica dos BDs: permitem a **abstração de dados**, ocultando detalhes do armazenamento de dados, que são desnecessários para a maioria dos usuários do BD.
- **Modelo de dados** é um conjunto de conceitos que podem ser usados para descrever a estrutura de um banco de dados. Fornece significado necessário para permitir a abstração.

Conceitos e Arquiteturas

Modelos de dados, esquemas e instâncias

- Estrutura de um banco de dados: os tipos de dados, relacionamentos e restrições.
-
- Os modelos devem incluir as operações básicas para a recuperação e atualização no BD.
-
- Fornece conceitos para especificar o aspecto dinâmico ou o comportamento de uma aplicação de banco de dados.

Conceitos e Arquiteturas

Modelos de dados, esquemas e instâncias

- Categorias de modelos de dados:
- **Modelos de alto nível:** modelos de dados conceituais, que possuem conceitos que descrevem os dados como os usuários os percebem;
- **Modelos de baixo nível:** modelos físicos, contêm conceitos que descrevem os detalhes de como os dados serão armazenados no computador.
-
- Modelos de baixo nível são significativos para especialistas computacionais, mais não para seus usuários finais.

Conceitos e Arquiteturas

Modelos de dados, esquemas e instâncias

Modelo de Dados?

O primeiro Sistema Gerenciador de Banco de Dados (SGBD) comercial surgiu no final de 1960 com base nos primitivos sistemas de arquivos disponíveis na época, os quais não controlavam o acesso concorrente por vários usuários ou processos. Os SGBDs evoluíram desses sistemas de arquivos de armazenamento em disco, criando novas estruturas de dados com o objetivo de armazenar informações. Com o tempo, os SGBDs passaram a utilizar diferentes formas de representação, ou **modelos de dados**, para descrever a estrutura das informações contidas em seus bancos de dados. Atualmente, os seguintes modelos de dados são normalmente utilizados pelos SGBD's: modelo hierárquico, modelo em redes, modelo relacional (amplamente usado) e o modelo orientado a objetos.

Conceitos e Arquiteturas

Modelos de dados, esquemas e instâncias

Modelagem de Dados → modelo de dados

Modelar significa criar um modelo que represente as características de funcionamento e comportamento de um objeto qualquer.

Tendo em vista que um sistema de banco de dados deve prover recursos para armazenar dados de um domínio real em unidades de armazenamento secundário a modelagem de dados deve fornecer um modelo de dados de alto nível(???).

Uma modelagem leva à um modelo!!

Conceitos e Arquiteturas

Modelos de dados, esquemas e instâncias

Tipos de modelagem:

Modelagem Conceitual: Busca construir um modelo conceitual de dados, definindo as entidades do modelo e seus relacionamentos básicos;

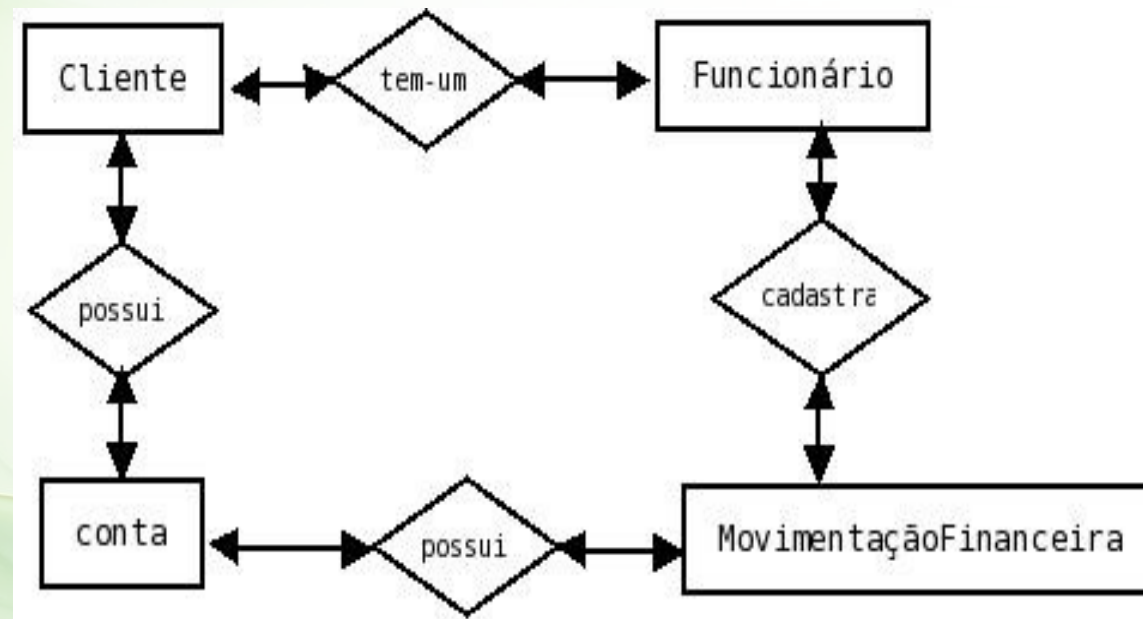
Modelagem Lógica: Constrói um modelo que demonstre as ligações entre as entidades e a lógica destas ligações;

Modelagem Física: Demonstrar como os dados serão organizados fisicamente, como será construído no sistema e implementado para garantir que o modelo lógico seja composto.

Conceitos e Arquiteturas

- **Modelos de dados, esquemas e instâncias**

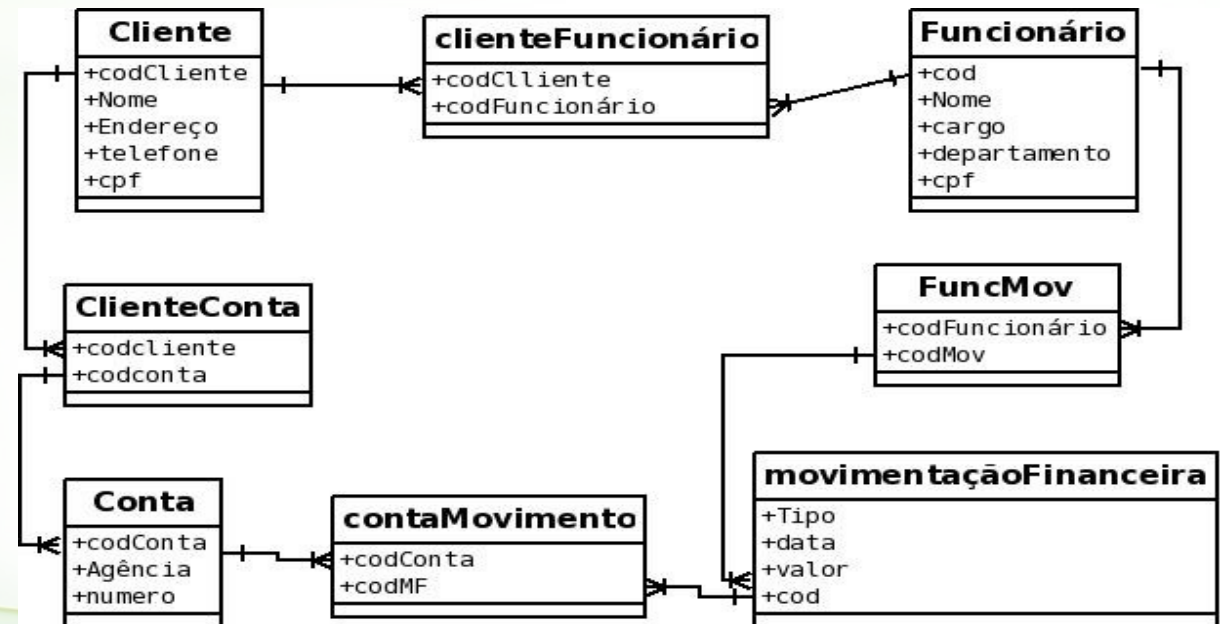
Modelo Conceitual - banco de dados: banco



Conceitos e Arquiteturas

- Modelos de dados, esquemas e instâncias**

Modelo Lógico de banco de dados: banco



Conceitos e Arquiteturas

Modelos de dados, esquemas e instâncias

Modelo físico-banco de dados: banco

DDL

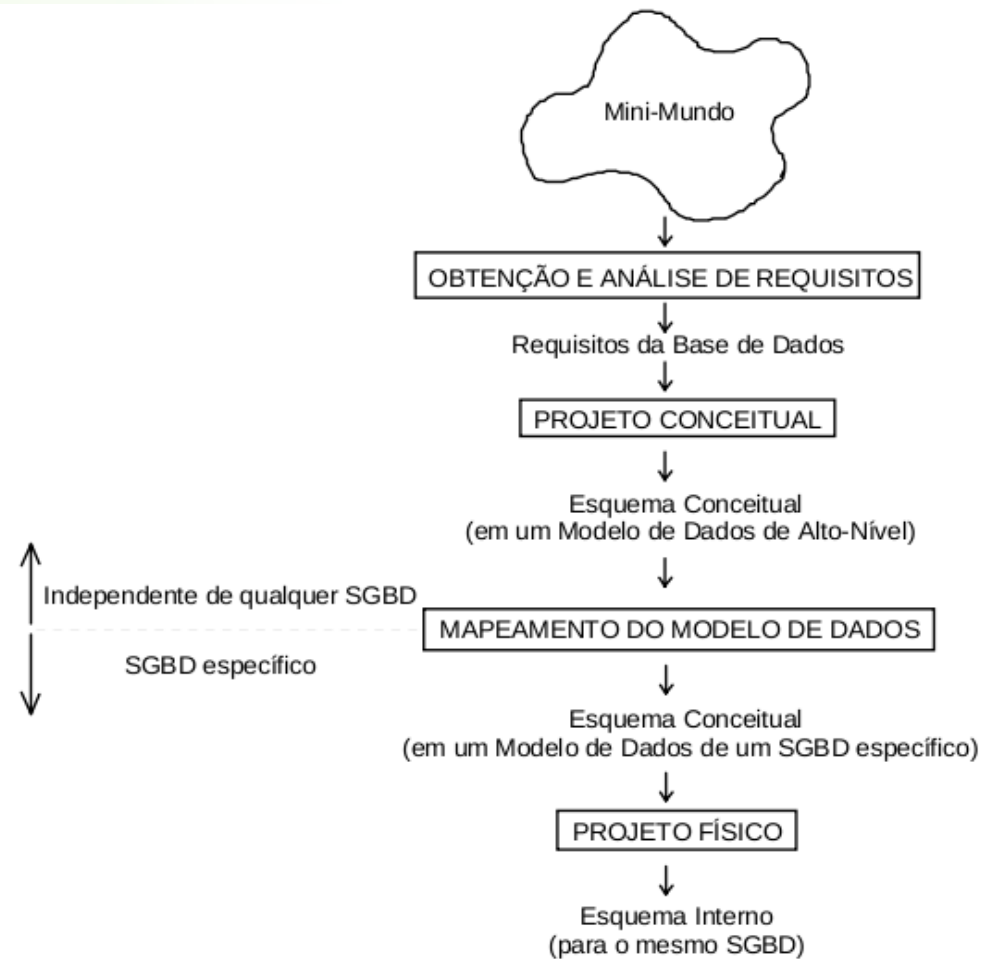
```
create table Cliente(  
    codcliente serial not null primary key,  
    nome string not null,  
    Endereco text not null,  
    telefone text not null,  
    cpf integer not null  
);  
  
create table ClienteConta(  
    codcliente integer not null,  
    codconta integer not null  
);  
  
alter table Cliente add foreign key (codcliente)
```

```
references ClienteConta (codcliente)
```

```
on delete restrict on update restrict;
```

Conceitos e Arquiteturas

Modelos de dados, esquemas e instâncias



Conceitos e Arquiteturas

Modelos de dados, esquemas e instâncias

Há de se compreender que quando falamos de modelos de dados, pode-se estar referindo à três extremos:

- **Modelos de dados representacionais:** ou de implementação, que oferece conceitos que podem ser entendidos pelos usuários finais, mais não estão muito distantes da forma como os dados estarão definidos em nível de armazenamento no computador.
- Estes modelos ocultam alguns detalhes de armazenamento dos dados, porém, podem ser implementados em sistemas gerenciadores de bancos de dados de forma mais direta.
 - Modelo de dados relacional
 - Modelo de dados em rede
 - Modelo de dados hierárquico
 - Modelo de dados orientados à objetos

Conceitos e Arquiteturas

Modelos de dados, esquemas e instâncias

Há de se compreender que quando falamos de modelos de dados, pode-se estar referindo à três extremos:

- **Modelos de dados conceituais:** utilizam conceitos como entidades, atributos e relacionamentos para descrever conceitualmente a representação e dos dados e informações do mundo real observado.
-
- Basicamente, uma entidade representa um objeto do mundo real ou um conceito, como um departamento, um cliente, um produto, um pedido.
- Um atributo corresponde às propriedades que estas entidades possuem, que permitem a identificação de elementos distintos.
-
- No mundo real, estas entidades possuem relacionamentos, ou seja, se associam.

Conceitos e Arquiteturas

Modelos de dados, esquemas e instâncias

Há de se compreender que quando falamos de modelos de dados, pode-se estar referindo à três extremos:

- **Modelos de dados físicos:** Descrevem como os dados estão armazenados em arquivos no computador, pela representação da informação como o formato do registro, com suas respectivas ordens de criação e rotas de acesso.

Conceitos e Arquiteturas

Modelos de dados, esquemas e instâncias

Modelo Hierárquico:

Primeiro modelo reconhecido como modelo de dados. Possível depois do surgimento dos discos de armazenamento endereçáveis, pois é possível a exploração de sua estrutura de endereçamento físico para viabilizar a representação hierárquica das informações.

Os dados são estruturados em hierarquias ou árvores. Os nós das hierarquias contêm ocorrências de registros, onde cada registro é a coleção de campos.

O registro que precede outro é o registro-pai, e os outros são registros-filho. Cada filho só tem um pai.

Para se navegar nos registros é preciso ir do topo para as folhas, da esquerda para a direita no mesmo nível.

Conceitos e Arquiteturas

Modelos de dados, esquemas e instâncias

Modelo Hierárquico:

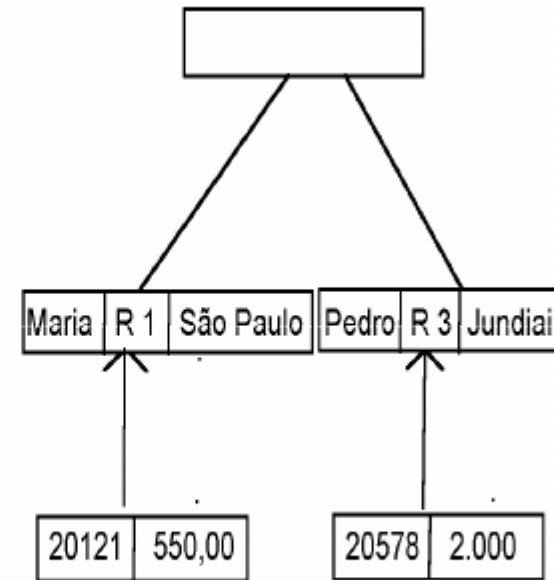
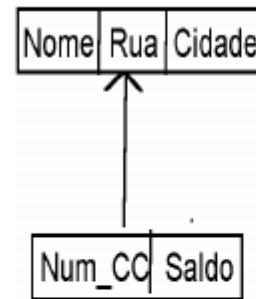


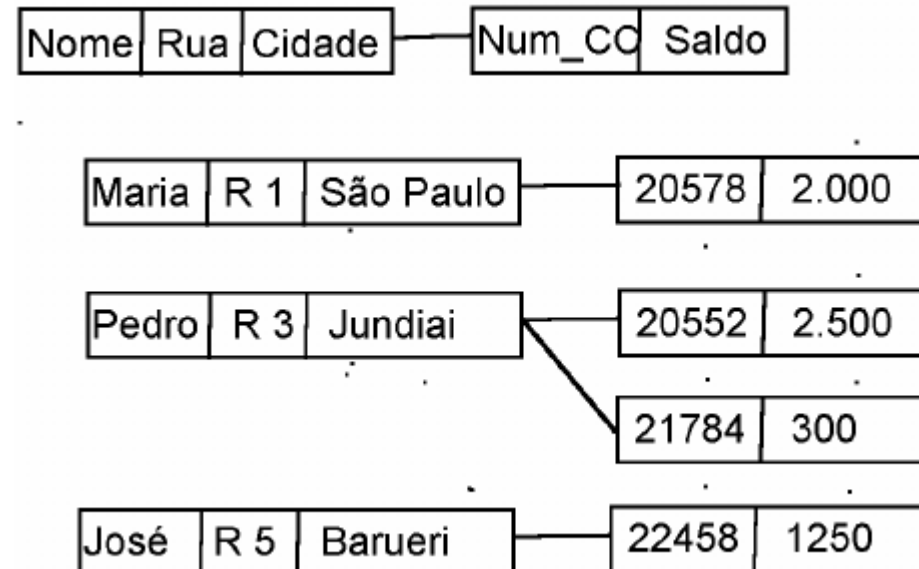
Figura 1.1 - Diagrama de estrutura de árvore Cliente - Conta Corrente

Conceitos e Arquiteturas

Modelos de dados, esquemas e instâncias

Modelo em Rede:

Extensão do modelo hierárquico, eliminando o conceito de hierarquia, permitindo que um mesmo registro estivesse envolvido em várias associações. Grafo.



Conceitos e Arquiteturas

Modelos de dados, esquemas e instâncias

Modelo Relacional:

Apareceu devido as necessidades de aumentar a independência de dados nos SGBDs, de prover um conjunto de funções apoiadas em álgebra relacional para armazenamento e recuperação de dados, além de permitir o processamento ad hoc (dedicado e exclusivo).

Tem por base a teoria dos conjuntos e álgebra relacional. Revela-se mais flexível e adequado ao solucionar problemas que se colocam no nível da concepção e implementação da base de dados.

Sua estrutura básica é a relação, ou RelVar, que pode ser definida como uma tabela, que contém um ou mais atributos relacionados. Cada instância do esquema (linha) é chamada de tupla.

Não há caminho pré-definido para se acessar os dados.

Conceitos e Arquiteturas

Modelos de dados, esquemas e instâncias

Modelo Relacional:

Cod_Cliente	Nome	Rua	Cidade
1	Pedro	A	São Paulo
2	Maria	B	Jundiai

Num_CC	Saldo
20121	1200
21582	1320
21352	652

Cod_Cliente	Num_CC
1	20121
2	21582
2	21352

Conceitos e Arquiteturas

Modelos de dados, esquemas e instâncias

Esquema de banco de dados: é a descrição do banco de dados, e não o banco de dados em si.

Um esquema pode ser uma forma de representação, inclusive visual, de um banco de dados, como com a utilização de um diagrama esquemático.

Um diagrama esquemático mostra somente alguns aspectos do esquema, como os nomes dos tipos de registro e itens de dados, além de alguns tipos de restrições.

Um esquema de banco de dados é utilizado pelo SGBD para definir e manter um BD.

É definido durante o projeto do BD e se espera que não se altere frequentemente.

É um catálogo de recursos mantidos pelo SGBD.

Conceitos e Arquiteturas

Modelos de dados, esquemas e instâncias

Arquitetura de Três-Esquemas (ANSI/SPARC):

Arquitetura de SGBDs que permite cumprir parte dos objetivos do seu uso (relembrando):

- a) separação de programas e dados;
- b) suporte a múltiplas visões de usuários;
- c) uso de catálogo para armazenar a descrição do BD (esquemas).

Esquema interno + Esquema Conceitual + Esquema Externo = Independência dos dados.

Conceitos e Arquiteturas

Modelos de dados, esquemas e instâncias

Camadas de esquemas: Objetivo de separar o usuário da aplicação do banco de dados físico.

- **Nível interno ou esquema interno**
 - **Descreve a estrutura de armazenamento físico do BD;**
 - **Utiliza o modelo de dados físico e descreve os detalhes complexos do armazenamento de dados e caminhos de acesso.**

Conceitos e Arquiteturas

Modelos de dados, esquemas e instâncias

Camadas de esquemas: Objetivo de separar o usuário da aplicação do banco de dados físico.

- **Nível conceitual ou esquema conceitual**
 - **Descreve a estrutura de todo o banco de dados para a comunidade de usuários. Oculta os detalhes das estruturas de armazenamento físico e se concentra na descrição das entidades, atributos e relacionamentos, além das operações dos usuários e suas restrições;**
 - **Utiliza modelos de dados representacionais;**

Conceitos e Arquiteturas

Modelos de dados, esquemas e instâncias

Camadas de esquemas: Objetivo de separar o usuário da aplicação do banco de dados físico.

- **Nível externo ou esquema externo**
 - **Abrange os esquemas externos ou visões de usuários.**
 - **Cada esquema externo descreve a parte do BD que um grupo de usuários tem interesse em particular e oculta o restante do BD.**

Conceitos e Arquiteturas

Modelos de dados, esquemas e instâncias

Independência dos dados

A arquitetura de camadas de esquemas é utilizada na independência dos dados, que é a capacidade de mudar o esquema em um nível do sistema de banco de dados sem que ocorram alterações do esquema no próximo nível mais alto. Tipos de independência:

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

Conceitos e Arquiteturas

Modelos de dados, esquemas e instâncias

Independência Física

Exemplo:

- Alterações no esquema interno para permitir uma reorganização física dos dados;
- Alterações para aperfeiçoar o desempenho de recuperações de informações;
- Alterações no hardware de armazenamento;
- Alterações nas versões do SGBD.

Conceitos e Arquiteturas

Modelos de dados, esquemas e instâncias

Instâncias

O dado no banco de dados pode ser alterado frequentemente. O banco de dados muda toda vez que inserimos um aluno ou registramos uma nova nota para uma avaliação.

Os dados no banco de dados são chamados de Ocorrências, ou instâncias no banco de dados.

Também chamado de ESTADO de um banco de dados.

•

Conceitos e Arquiteturas

Linguagem de banco de dados

Assim que o projeto de um BD está completo e o SGBD foi escolhido, o primeiro ponto é especificar o esquema conceitual e interno para o banco de dados.

Grande parte do trabalho de construção de esquemas e mapeamentos é feita por intermédio de linguagens.

- **DDL – Data Definition Language, ou linguagem de definição de dados.**
- Usada pelo DBA e projetistas para criar os esquemas: conceitual e interno.
- É compilado pelo SGBD, que processa o DDL a fim de identificar os construtores e para armazenar a descrição do esquema no catálogo do SGBD.

Conceitos e Arquiteturas

Linguagem de banco de dados

Assim que o projeto de um BD está completo e o SGBD foi escolhido, o primeiro ponto é especificar o esquema conceitual e interno para o banco de dados.

Grande parte do trabalho de construção de esquemas e mapeamentos é feita por intermédio de linguagens.

- **SDL – Storage Definition Language, ou linguagem de definição de armazenamento.**
- Em SGBDs que há uma clara distinção entre os níveis conceitual e interno, DDL é usada só para especificar o esquema conceitual.
- SDL então é utilizada para definição do esquema interno.

Conceitos e Arquiteturas

Linguagem de banco de dados

Assim que o projeto de um BD está completo e o SGBD foi escolhido, o primeiro ponto é especificar o esquema conceitual e interno para o banco de dados.

Grande parte do trabalho de construção de esquemas e mapeamentos é feita por intermédio de linguagens.

- **VDL – View Definition Language, ou linguagem de definição de visões.**
- **Útil em SGBDs com arquitetura Três-Esquemas pura.**
- **Especifica as visões dos usuários e os seus mapeamentos para o esquema conceitual.**

Na maioria dos SGBDs atuais, DDL agrupa tanto VDL quanto a SDL.

Conceitos e Arquiteturas

- **Linguagem de banco de dados**

Assim que o projeto de um BD está completo e o SGBD foi escolhido, o primeiro ponto é especificar o esquema conceitual e interno para o banco de dados.

Grande parte do trabalho de construção de esquemas e mapeamentos é feita por intermédio de linguagens.

- **DML – Data Manipulation Language, ou linguagem de manipulação de dados.**
- **Operações com os dados do BD implementado no SGBD**
 - **Create**
 - **Read**
 - **Update**
 - **Delete**

-

Conceitos e Arquiteturas

Linguagem de banco de dados

Assim que o projeto de um BD está completo e o SGBD foi escolhido, o primeiro ponto é especificar o esquema conceitual e interno para o banco de dados.

Grande parte do trabalho de construção de esquemas e mapeamentos é feita por intermédio de linguagens.

- **DML – Data Manipulation Language, ou linguagem de manipulação de dados.**
- **Dois tipos de DML: Alto nível ou não procedural, utilizada para especificar operações complexas no BD (query language), e DML de nível baixo, ou procedural, que precisa ser embutida em uma linguagem de programação de propósito geral, que possibilita a recuperação de registros, sendo possíveis serem manipulados pela linguagem de programação.**

Conceitos e Arquiteturas

Arquitetura SGBD

Centralizada:

Acompanhando a visão da época em que os computadores eram muito caros, e que as empresas compravam poucos destes computadores e os mantinham via acesso centralizados.

Mesmo em sistemas de computadores pessoais, com baixo custo, ainda é bastante usual os SGBDs centralizados.

Todo o processamento do SGBD é realizado no servidor, e enviado como resposta ao cliente solicitante.

Arquitetura Cliente-servidor: presença de camadas entre o usuário solicitante e o servidor do SGBD.

Conceitos e Arquiteturas

Arquitetura SGBD

