## Navathe – resumo + exercícios

(primeiras 17 páginas)

Resumo

O livro do Navathe começa apresentando os conceitos básicos sobre bancos de dados e sistemas de gerenciamento de banco de dados (SGBDs). Um banco de dados é descrito como uma coleção organizada de dados relacionados, que representam aspectos do mundo real (ou “minimundo”, como o autor chama), com um objetivo específico e voltado a um grupo definido de usuários. Já o SGBD entra como o software que permite criar, manipular e manter esses dados de forma eficiente (em outras palavras, é quem faz a mágica acontecer por trás dos bastidores).

Logo de início, o autor destaca que o SGBD armazena metadados (dados sobre os dados, tipo nome de tabela, tipos de atributos etc.) em um catálogo, o que deixa o sistema mais flexível e fácil de adaptar para diferentes aplicações. Outra característica importante é a abstração de dados: o SGBD esconde os detalhes do armazenamento físico, e isso facilita muito a vida de quem desenvolve, porque permite alterar estruturas internas sem quebrar os sistemas que já estão rodando (ou seja, programas e dados ficam independentes entre si).

Também rola a ideia das múltiplas visões do banco. Isso significa que cada usuário pode acessar só aquilo que precisa, da forma que faz mais sentido pra ele (por exemplo, o gerente vê uma tela diferente do atendente). E mesmo quando várias pessoas estão acessando ao mesmo tempo, o sistema tem mecanismos para garantir que tudo continue funcionando direitinho, sem bagunçar os dados (isso é o controle de concorrência).

No ambiente de banco de dados, existem vários perfis. O DBA (Administrador de Banco de Dados) é quem cuida da segurança, da integridade e do desempenho geral. Os projetistas decidem quais dados precisam ser armazenados e como organizá-los. Os usuários finais variam bastante: alguns só fazem consultas rápidas (usuários casuais), enquanto outros criam aplicações mais complexas (usuários sofisticados). Também temos os programadores e analistas, que são os responsáveis por desenvolver sistemas que se conectam ao banco (a galera de ADS, por exemplo).

Usar um SGBD traz várias vantagens. Reduz a redundância (em vez de repetir os mesmos dados em vários lugares, armazena tudo de forma centralizada), oferece controle de acesso (cada um vê só o que pode), armazena objetos complexos de forma persistente (o que é útil pra sistemas mais avançados), e torna as consultas mais rápidas com o uso de índices e técnicas de otimização. Além disso, o sistema é preparado para lidar com falhas (tem backup e recuperação) e ainda garante a integridade dos dados com regras definidas no próprio banco.

O capítulo também traz um panorama histórico. Os primeiros sistemas eram hierárquicos e em rede (bem limitados e com pouca flexibilidade). Depois vieram os bancos relacionais (que separaram o físico do lógico e trouxeram linguagens de consulta como o SQL). Mais tarde, surgiram os bancos orientados a objetos (que prometiam lidar melhor com dados complexos, mas não pegaram muito). Com a popularização da internet, o XML ganhou força (por facilitar a troca de dados entre sistemas) e começaram a surgir aplicações mais específicas, como comércio eletrônico, sistemas GIS e mineração de dados.

Mas nem sempre um SGBD é a melhor escolha. Em sistemas muito simples e estáticos (tipo uma planilha que quase nunca muda), pode ser exagero usar um. O mesmo vale para aplicações embarcadas (que têm pouco recurso) ou sistemas de tempo real (em que cada milissegundo conta, e o overhead do SGBD pode atrapalhar).

Pra fechar, o autor reforça como os bancos de dados são fundamentais na nossa sociedade atual (presentes em tudo, de apps de delivery até sistemas hospitalares) e mostra a diferença entre os bancos de dados tradicionais e os sistemas de recuperação de informação (tipo os buscadores da web, que lidam com dados não estruturados e têm objetivos diferentes).

Exercícios de revisão

**1.1. Definições**

**Dados:** Fatos que podem ser registrados e têm significado implícito.

**Banco de Dados (BD):** Coleção de dados relacionados.

**SGBD (Sistema Gerenciador de Banco de Dados):** Conjunto de programas para gerenciar um BD.

**Sistema de Banco de Dados**: Combinação do BD, SGBD e aplicações associadas.

**Catálogo de Banco de Dados:** Armazena a descrição (metadados) da estrutura do banco.

**Independência entre dados e programas:** Capacidade de alterar estrutura dos dados sem alterar os programas.

**Visão do Usuário:** Parte do banco visível a um grupo de usuários.

**DBA (Administrador de BD):** Responsável pela supervisão e controle do BD.

**Usuário Final:** Pessoa que acessa o banco para realizar tarefas do dia a dia.

**Transação Programada:** Unidade de programa que executa uma função no BD.

**Sistema de BD Dedutivo:** Sistema que usa regras lógicas para inferência de dados.

**Objeto Persistente:** Dados que continuam existindo após o fim do programa.

**Metadados:** Dados que descrevem outros dados.

**Aplicação para Processamento de Transação:** Programas que acessam BD para registrar transações rotineiras.

**1.2. Quatro tipos principais de ações**

1. **Definição**: Criar estruturas de dados (tabelas, índices, visões).
2. **Construção**: Inserir dados iniciais no banco.
3. **Manipulação**: Consultas e atualizações (inserção, exclusão, modificação).
4. **Compartilhamento**: Permitir acesso simultâneo por múltiplos usuários .

**1.3. Características da abordagem de banco de dados**

* Redução da redundância.
* Compartilhamento de dados.
* Controle de acesso.
* Persistência.
* Restrições de integridade.
* A principal diferença para arquivos tradicionais é a centralização e o controle dos dados, enquanto arquivos são controlados por programas individuais .

**1.4. Responsabilidades**

* **DBA**: Definir estrutura de armazenamento, esquemas, autorizações, desempenho.
* **Projetistas de BD**: Projetam o esquema conceitual, lógico e físico do banco .

**1.5. Tipos de usuários finais**

* **Usuários casuais**: Usam consultas ocasionais.
* **Usuários avançados**: Conhecem linguagem de consulta.
* **Usuários paramétricos**: Executam tarefas repetitivas.
* **Usuários móveis/autônomos**: Usam o banco remotamente.
* **Programadores de aplicações**: Desenvolvem sistemas que usam o BD.

**1.6. Capacidades de um SGBD**

* Definição de dados.
* Manipulação de dados.
* Segurança e autorização.
* Confiabilidade e recuperação.
* Controle de concorrência.
* Interfaces de usuário e APIs.
* Catálogo de metadados.

**1.7. Diferenças entre banco de dados e recuperação de informações**

* **Banco de Dados**: Dados estruturados, consultas exatas.
* **Recuperação de Informações**: Dados não estruturados (como texto), busca por relevância.

Exercícios

**1.8. Operações informais de consulta e atualização esperadas:**

**Consultas:**

* Listar todas as disciplinas que "Silva" cursou e as respectivas notas.
* Consultar quais disciplinas têm como pré-requisito "Estrutura de Dados".
* Obter todas as turmas oferecidas pela disciplina "Banco de Dados".
* Verificar qual o curso do aluno de número 8.

**Atualizações:**

* Inserir uma nova turma para "Banco de Dados".
* Atualizar a nota de "Silva" em determinada turma.
* Mudar o curso do aluno "Braga" para "ENG".
* Adicionar um novo aluno com seus dados completos.

**1.9. Diferença entre redundância controlada e não controlada (com exemplos):**

* **Redundância controlada**: intencional, planejada para melhorar desempenho. Ex: repetir Nome\_aluno no HISTORICO\_ESCOLAR para evitar junções frequentes.
* **Redundância não controlada**: ocorre por má prática, causa inconsistência. Ex: Braga ter duas datas de nascimento diferentes em arquivos distintos do sistema.

**1.10. Relacionamentos entre os registros da Figura 1.2:**

* **ALUNO ↔ HISTORICO\_E**
* **TURMA ↔ HISTORICO\_ESCOLAR**: por Identificacao\_turma.
* **DISCIPLINA ↔ PRE\_REQUISITO**: por Numero\_disciplina e Numero\_pre\_requi **SCOLAR:** por Numero\_aluno.
* **DISCIPLINA ↔ TURMA: por Numero\_disciplina.**
* sito.

**1.11. Visões adicionais possíveis para outros grupos:**

* **Secretaria Acadêmica**: visão com nome do aluno, curso, e disciplinas cursadas com notas.
* **Coordenação de Curso**: visão que mostra disciplinas ofertadas por semestre e pré-requisitos.
* **Professores**: visão com alunos por turma e suas respectivas notas.
* **Financeiro** (hipotético): visão apenas com nome, número do aluno e curso (sem notas).

**1.12. Restrições de integridade aplicáveis:**

* Numero\_aluno deve ser único (chave primária).
* Numero\_disciplina deve ser único.
* Identificacao\_turma deve ser único.
* Nota deve ser um dos seguintes caracteres: {‘A’, ‘B’, ‘C’, ‘D’, ‘F’}.
* Tipo\_aluno deve estar entre 1 e 5.
* Integridade referencial entre HISTORICO\_ESCOLAR.Numero\_aluno e ALUNO.Numero\_aluno.

**1.13. Exemplos de quando usar processamento de arquivos tradicional:**

* Cadastro simples de convidados para uma festa.
* Controle de inventário em uma loja pequena com poucos produtos e nenhuma relação complexa.
* Sistema embarcado em microcontrolador para leitura de sensores (tempo real, limitado).

**1.14. a) Colunas que precisariam ser atualizadas se 'CC' mudar para 'CCES':**

* DISCIPLINA.Departamento: para todas as disciplinas do departamento CC.
* ALUNO.Curso: para alunos do curso CC.

b) **Reestruturação para minimizar atualização:**

Uma solução seria criar uma tabela separada de **Departamentos**, com código (CC, MAT etc.) e nome completo. Então, as tabelas DISCIPLINA e ALUNO apenas referenciariam esse código. Assim, ao mudar o nome ou prefixo de um departamento, seria necessário alterar apenas essa nova tabela de departamentos.