Introdução a Bancos de Dados

Conceitos - Modelos e Arquitetura

Clodoveu Davis

DCC/UFMG

Modelos de dados

- **Modelo de dados**
  - Um conjunto de conceitos usados para descrever a ***estrutura*** de um banco de dados, as ***operações*** para manipular tais estruturas, e as ***restrições*** a que o banco de dados tem que obedecer.

Modelos de dados

- **Estrutura e restrições do modelo de dados**
  - ***Primitivas*** são usadas para definir a estrutura do banco de dados
  - As primitivas tipicamente incluem ***elementos*** de dados (e seus tipos), ***grupos de elementos*** (p. ex. entidade, registro, tabela), e ***relacionamentos*** entre esses grupos
  - ***Restrições*** estabelecem os critérios para que os dados sejam considerados válidos; essas regras devem ser aplicadas todo o tempo

Modelos de dados

- **Operações do modelo de dados**
  - Operações são usadas para especificar a recuperação e atualização dos dados, fazendo referência aos artefatos do modelo de dados
  - As operações podem incluir ações básicas sobre o modelo (p. ex., inserção, exclusão, atualização) e ações definidas pelo usuário (p. ex., regras de negócio como “cálculo de RSG” e “atualização do estoque”)

Categorias de modelos de dados

- **Conceituais (de alto nível, semânticos)**
  - Envolvem conceitos que estão próximos da maneira Segundo a qual muitos usuários percebem os dados
    - (Também chamados de ***modelos de entidades*** ou ***modelos de objetos***)
- **Físicos (de baixo nível, internos)**
  - Envolvem conceitos que descrevem detalhes de como os dados são armazenados no computador
- **De implementação (representacionais)**
  - Envolvem conceitos que estão entre os dois anteriores, usados por muitas implementações comerciais de SGBD (p. ex. o modelo relacional é usado em muitos SGBDs de mercado)
- **Autodescritivos**
  - Combinam a descrição dos dados com os valores propriamente ditos. Exemplos: XML, repositórios chave-valor e alguns sistemas NoSQL

Esquemas versus Instâncias

- Esquema de um banco de dados
  - Uma ***descrição*** do banco de dados
  - Inclui detalhes sobre a estrutura do banco de dados, tipos de dados usados, e restrições aplicáveis sobre os dados
- Diagrama do esquema
  - Uma ***figura ilustrativa*** de (grande parte dos) aspectos do esquema de um banco de dados
- Primitiva do esquema
  - Um ***componente do esquema*** ou um objeto inserido no esquema, como por exemplo ESTUDANTE, DISCIPLINA

Esquemas versus Instâncias

- Estado do banco de dados
  - Os dados efetivamente armazenados em um banco de dados em um ***instante particular de tempo***. Isso inclui toda a coleção de dados presentes no banco de dados.
  - Também chamado de ***instância do banco de dados*** (ou ***ocorrência***, ou ***instantâneo***, ou ***snapshot***)
    - A palavra instância também é usada para referenciar componentes individuais do banco de dados, como *instância de registro*, *instância de tabela*, *instância de entidade*
      - *P. ex., UM estudante na tabela ESTUDANTE é uma instância de ESTUDANTE.*

Esquema do banco de dados versus Estado do banco de dados

- Estado do banco de dados
  - Refere-se ao ***conteúdo*** do banco de dados em um determinado momento
- Estado **inicial** do banco de dados
  - Refere-se ao estado do banco de dados quando ele é inicialmente carregado em um sistema
- Estado **válido**
  - Um estado do banco de dados que satisfaça a estrutura e as restrições definidas para o banco de dados

Esquema do banco de dados versus Estado do banco de dados

- Distinção
  - O ***esquema do banco de dados*** muda muito raramente
  - O ***estado do banco de dados*** muda toda vez que o banco de dados é atualizado

Exemplo de um esquema de BD

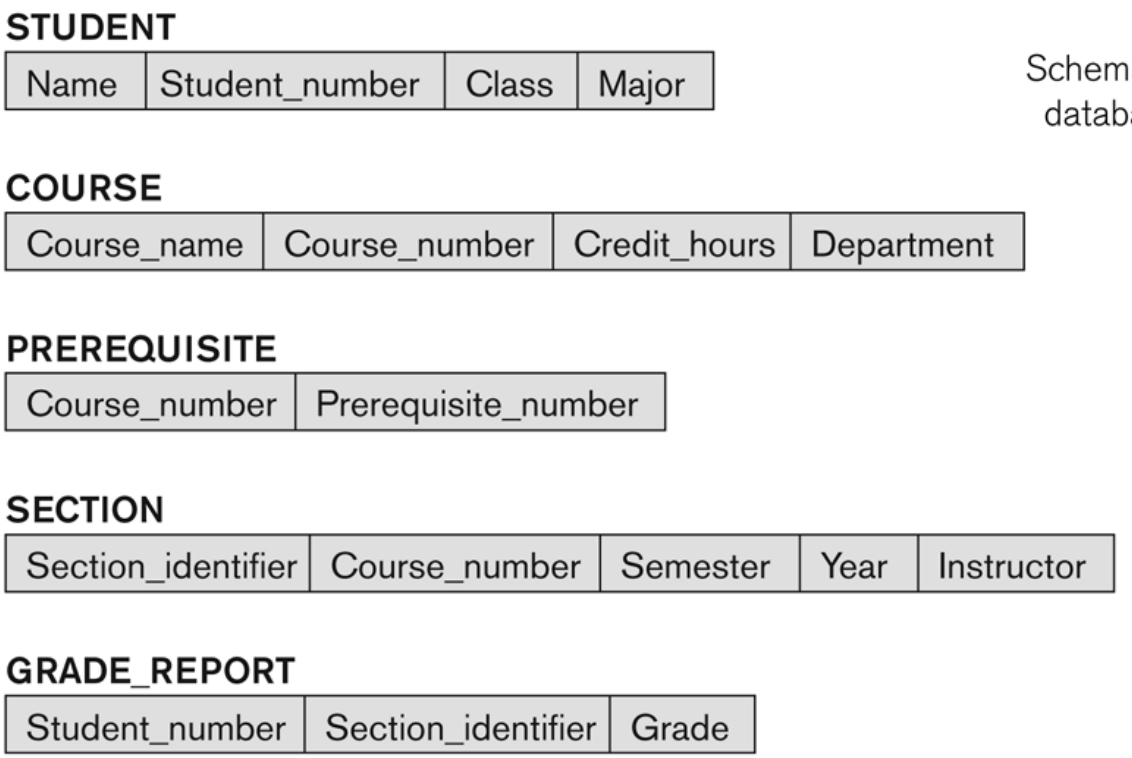


Figure 2.1
Schema diagram for the database in Figure 1.2.

Exemplo do estado de um banco de dados

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	04	King
92	CS1310	Fall	04	Anderson
102	CS3320	Spring	05	Knuth
112	MATH2410	Fall	05	Chang
119	CS1310	Fall	05	Anderson
135	CS3380	Fall	05	Stone

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Figure 1.2
A database that stores student and course information.

Arquitetura de Três Esquemas

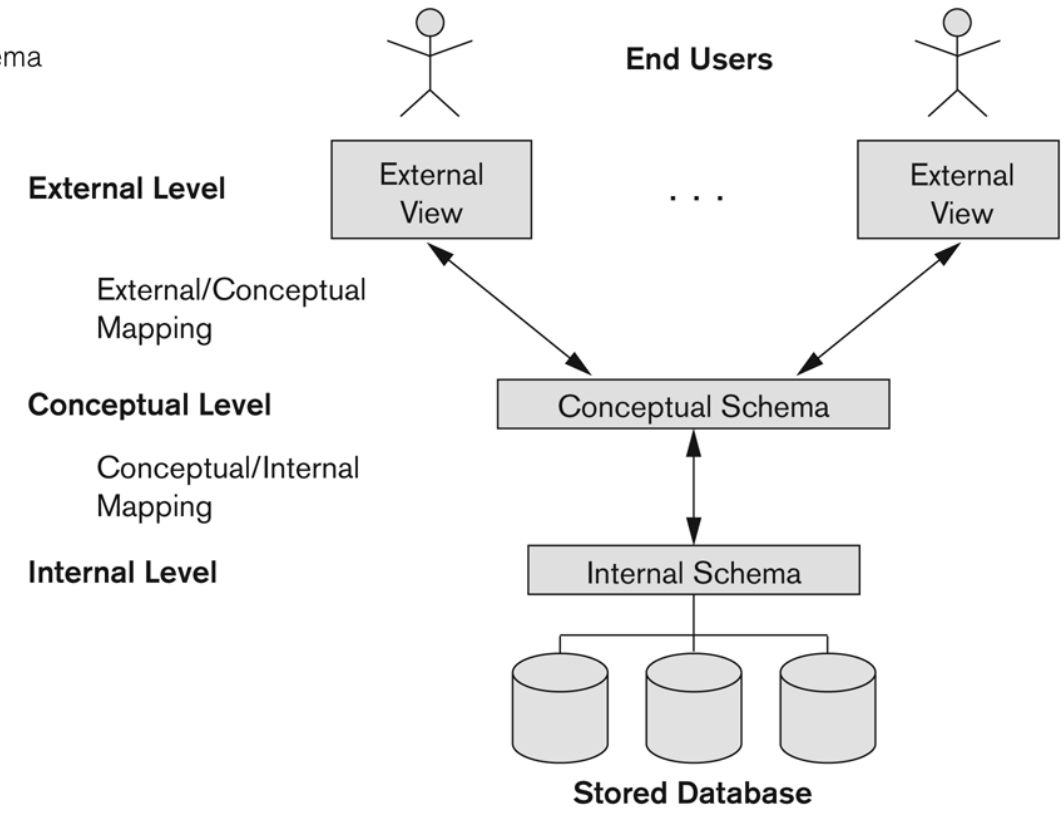
- Proposta para suportar as seguintes características do enfoque de bancos de dados:
  - **Independência entre programas e dados**
  - Suporte a **múltiplas visões** dos dados
- Não é explicitamente usada em SGBD comerciais, mas é útil para se perceber a organização de um sistema de banco de dados

Arquitetura de Três Esquemas

- Define os esquemas de SGBDs em três níveis:
  - **Esquema interno,** no nível interno, para descrever as estruturas de armazenamento físico e caminhos de acesso (p. ex., índices)
    - Tipicamente usa um modelo de dados **físico**
  - **Esquema conceitual,** no nível conceitual, para descrever a estrutura e restrições para todo o banco de dados e para a comunidade de usuários
    - Usa um modelo de dados **conceitual** ou **de implementação**
  - **Esquemas externos** no nível externo, para descrever as várias visões de usuários
    - Em geral usa o mesmo modelo de dados que o esquema conceitual

Arquitetura de três esquemas

Figure 2.2
The three-schema architecture.



Arquitetura de três esquemas

- São realizados **mapeamentos** entre os níveis do esquema para transformar os requisitos e os dados
  - Programas se referem a um esquema externo, e são mapeados pelo SGBD para um esquema interno para execução
  - Dados extraídos do nível interno do SGBD são reformatados para que correspondam à visão externa do usuário (exemplo: formatar os resultados de uma consulta SQL para apresentação em uma página Web)

Independência de Dados

- **Independência Logica de Dados**
  - A capacidade de alterar o esquema conceitual sem ter que alterar o esquema externo e os programas de aplicação associados
- **Independência Física de Dados**
  - A capacidade de alterar o esquema interno sem ter que alterar o esquema externo
  - Por exemplo, o esquema interno pode ser alterado quando certas estruturas de arquivos são reorganizadas, ou quando índices são criados para melhorar o desempenho no acesso

Independência de Dados

- Quando um esquema de nível baixo é alterado, só os **mapeamentos** entre esse esquema e esquemas de nível superior precisam ser ajustados, se o SGBD suportar totalmente a independência de dados
- Os esquemas de alto nível **não são alterados**
  - Portanto, os programas de aplicação não precisam ser alterados, já que se referem ao esquema externo

Linguagens de SGBD

- Data Definition Language (DDL) – linguagem de definição de dados
- Data Manipulation Language (DML) – linguagem de manipulação de dados
  - Linguagens de alto nível ou não procedurais: incluem a álgebra relacional e SQL
    - Podem ser usadas diretamente ou embutidas em uma linguagem de programação
  - Linguagens de baixo nível ou procedurais
    - Devem ser embutidas em uma linguagem de programação

Linguagens de SGBD

- **Data Definition Language (DDL)**
  - Usada pelo DBA (Database Administrator, administrador do BD) e pelos projetistas de bancos de dados para especificar o esquema conceitual de um banco de dados
  - Em muitos SGBDs, a DDL é também usada para definer esquemas internos e externos (visões)
  - Em alguns SGBDs, uma **storage definition language (SDL)** separada e uma **view definition language (VDL)** são usadas para definir os esquemas interno e externo
    - Funções de SDL são tipicamente implementadas por meio de comandos ao SGBD emitidos pelo DBA e pelos projetistas do BD

Linguagens de SGBD

- **Data Manipulation Language (DML)**
  - Usada para recuperar dados e para realizar atualizações
  - Comandos DML (uma sublinguagem de dados) podem ser embutidos em uma linguagem de programação de uso geral (linguagem hospedeira), como COBOL (!), C, C++ ou Java
    - Uma biblioteca de funções pode estar disponível para permitir acesso ao SGBD a partir de uma linguagem de programação
  - Como alternativa, comandos individuais de DML podem ser executados diretamente (linguagem de consulta – *query language*)

Tipos de DML

- **De alto nível ou não procedurais**
  - Por exemplo, SQL
  - São orientadas a conjuntos e especificam ***os dados a recuperar***, e não ***como recuperá-los***
  - Também chamadas de **linguagens declarativas**
- **De baixo nível ou procedurais**
  - Recuperam dados um registro por vez
  - Construções como loops são necessárias para recuperar múltiplos registros, posicionando apontadores

Interfaces entre SGBDs e Linguagens de Programação

- Interfaces para incorporar a DML em linguagens de programação
  - **Embutido**: e.g SQL embutido (para C, C++, etc.), SQLJ (for Java)
  - **Chamada de funções**: e.g. JDBC para Java, ODBC (Open Database Connectivity) para outras linguagens, funcionam como APIs (application programming interfaces)
  - **Linguagem de programação do banco de dados**: e.g. o ORACLE tem a PL/SQL, uma linguagem de programação baseada em SQL; em PostgreSQL existe a pg/plsql; são linguagens procedurais que incorporam o SQL e seus tipos como components integrais de sua implementação
  - **Linguagens de scripting:** PHP (client-side scripting) e Python (server-side scripting) são usadas para escrever programas para bancos de dados.

Módulos e componentes típicos de SGBDs

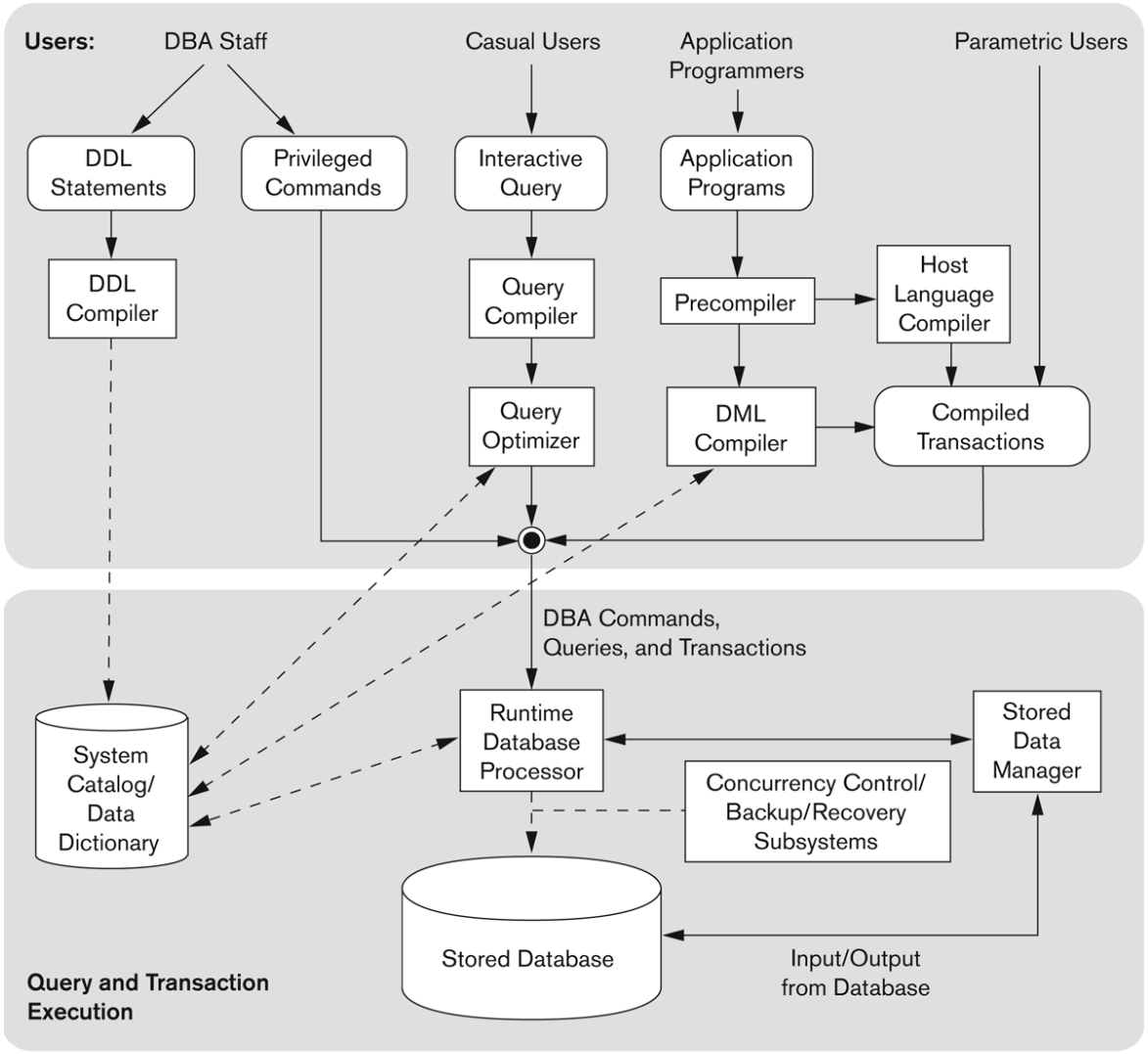


Figure 2.3
Component modules of a DBMS and their interactions.

Conceitos importantes: BDA

- Indexação
- Transação
- Controle de concorrência
- Recuperação de falhas
- Backup
- Log
- Processamento de consultas

Indexação

- Apenas uma pequena parte do BD precisará ser acessada a cada momento
- Para localizar algo no BD de maneira a reduzir a quantidade de acessos a disco necessária, cria-se **índices**
- Há vários tipos de índices (estruturas de dados em disco)
- Não é interessante ter índices para tudo, apenas para o que é mais usado na seleção e mais frequentemente modificado
- Escolher o que será indexado é parte do projeto físico

Transação

- Unidade lógica de processamento em relação ao banco de dados
- Inclui potencialmente várias operações (consulta, inclusão, exclusão, atualização)
- O SGBD tenta executar cada operação conforme programado; caso ocorra alguma falha, todos os efeitos da transação têm que ser desfeitos
- O processamento de múltiplas transações simultâneas, de diversos usuários, precisa ser coordenado para não introduzir inconsistências no BD

Controle de concorrência

- O processamento de múltiplas transações simultâneas, de diversos usuários, precisa ser coordenado para não introduzir inconsistências no BD
- O SGBD tem mecanismos que permitem travar partes do BD aguardando a conclusão de uma transação, impedindo que outras modifiquem essas partes
  - Se muita coisa for bloqueada, o desempenho de todo o SGBD pode ser afetado, pois outras transações ficarão esperando muito tempo

Recuperação de falhas e backup

- Quando ocorre alguma falha física (desconexão do servidor, falha de energia, falha da unidade de disco), uma transação pode ser impedida de completar
- Nesse caso, é necessário desfazer seus efeitos
  - Mas se a falha impedir o acesso ao disco, como desfazer?
- O SGBD mantém um controle de todas as operações realizadas desde um determinado instante: **log**
- Em caso de falha catastrófica, pode-se recorrer ao log para recuperar o estado do banco
- A cada backup, o log é salvo e inicia-se um novo ciclo

Log

- O log é uma lista ordenada das operações realizadas no BD
- Depende fundamentalmente de um mecanismo de determinação do tempo: **timestamp**
- A partir de um estado do banco garantido por um backup, o log pode ser usado para refazer tudo o que for perdido em uma falha catastrófica
- O log também pode ser usado para desfazer operações noc aso de falha nas transações
- Problema: não se pode gravar registros no log a cada instante, pois o tempo de acesso a disco é grande. Por outro lado, se os registros ficarem em buffers de memória, uma falha catastrófica pode causar sua perda

Processamento de consultas

- A consulta recebida em SQL precisa ser transformada em um **plano de execução**, para uso do processador de runtime
- O SGBD precisa decidir a melhor maneira de compor o plano de execução, de modo a minimizar o número de acessos a disco
  - Índices são considerados
  - Dados sobre as tabelas são considerados: volume, número de registros, organização física
  - Dados sobre atributos são considerados: tamanho, valores permitidos/distintos, variabilidade
- Problema: como confiar em dados sobre as tabelas e atributos, se esses mudam o tempo todo?

clodoveu@dcc.ufmg.br

Links

vCard