

21053 - Fundamentos de Bases de Dados

Professor:

Paulo Pombinho

S1 - Introdução



Exemplos de Aplicações de BD - 1

- Informação da Empresa
 - Vendas: clientes, produtos, compras
 - Contabilidade: pagamentos, recibos, ativos
 - Recursos Humanos: Informação sobre empregados, salários, impostos sobre os salários
- Fabrico: gestão da produção, inventário, encomendas, cadeia de abastecimento
- Banca e finanças
 - informação do cliente, contas, empréstimos e transações bancárias
 - Transações de cartões de crédito
 - Finanças: vendas e aquisições de instrumentos financeiros (por exemplo, ações e obrigações; armazenamento de dados do mercado em tempo real)

Exemplos de Aplicações de BD - 2

- Universidades: inscrições, notas
- Companhias aéreas: reservas, horários
- Telecomunicações: registos de chamadas, textos e utilização de dados, geração de faturas mensais, manutenção de saldos em cartões pré-pagos
- Serviços baseados na Web
 - Lojas online: rastreio de encomendas, recomendações personalizadas
 - Anúncios online

Exemplos de Aplicações de BD - 3

- Bases de dados de documentos
- Sistemas de navegação: Para manter as localizações de locais de interesse variados, juntamente com as rotas exatas de estradas, sistemas de comboios, autocarros, etc.

Finalidade dos SGBD - 1

Inicialmente, as aplicações de base de dados eram construídas diretamente em cima de sistemas de ficheiros, o que originava:

- Redundância e inconsistência de dados: os dados são armazenados em vários formatos de ficheiros, resultando na duplicação de informação em diferentes ficheiros
- Dificuldade no acesso aos dados
 - Necessário escrever um novo programa para realizar cada nova tarefa
- Isolamento de dados
 - Vários ficheiros e formatos

Finalidade dos SGBD - 2

- Problemas de integridade
 - As restrições de integridade (por exemplo, saldo da conta > 0) ficam “escondidas” no código do programa em vez de serem explicitamente declaradas
 - Difícil adicionar novas restrições ou alterar as existentes
- Atomicidade das atualizações
 - Falhas podem deixar base de dados num estado inconsistente com atualizações parciais realizadas
 - Exemplo: Transferência de dinheiro de uma conta para outra deve ter sido completada ou então não acontecer!

Finalidade dos SGBD - 3

- Acesso simultâneo por vários utilizadores
 - Acesso concorrente necessário para desempenho
 - Acessos concorrentes descontrolados podem originar inconsistências
 - Ex: Duas pessoas vendo um saldo (e.g. 100 €) e atualizando-o, retirando dinheiro (e.g. 50€) ao mesmo tempo
- Problemas de segurança
 - Difícil fornecer acesso ao utilizador a apenas alguns dos dados

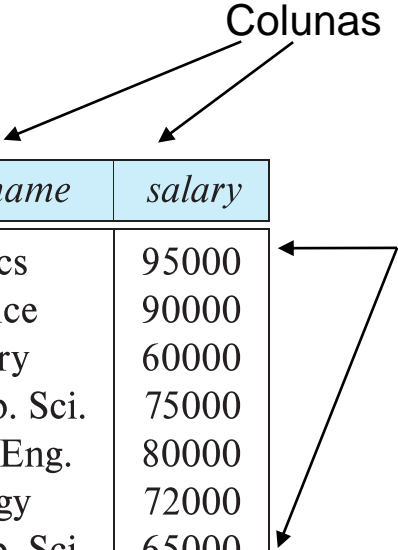
Os sistemas de gestão de base de dados oferecem soluções para todos os problemas mencionados!

Modelos de Dados

- Uma coleção de ferramentas para descrever
 - Dados
 - Relações dos dados
 - Semântica dos dados
 - Restrições aos dados
- Modelo relacional
- Modelo de dados Entidade-Relação (principalmente para desenho de bases de dados)

Modelo Relacional

- Todos os dados são armazenados em várias tabelas.
- Exemplo de tabela de dados num modelo relacional



<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>salary</i>
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

Exemplo de BD Relacional

<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>salary</i>
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

<i>dept_name</i>	<i>building</i>	<i>budget</i>
Comp. Sci.	Taylor	100000
Biology	Watson	90000
Elec. Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

(b) The *department* table

Instâncias e Esquemas

- Semelhante a tipos e variáveis em linguagens de programação
- **Esquema Lógico** – a estrutura lógica global da base de dados
 - Exemplo: A base de dados consiste em informações sobre um conjunto de clientes e contas num banco e a relação entre eles
 - Análogo ao tipo de informação de uma variável num programa
- **Esquema físico** – a estrutura física global da base de dados
- **Instância** – o conteúdo real da base de dados num determinado momento
 - Análogo ao valor de uma variável

Independência de Dados Físicos

- **Physical Data Independence** – a capacidade de modificar o esquema físico sem alterar o esquema lógico
 - As aplicações dependem do esquema lógico
 - Em geral, as interfaces entre os vários níveis e componentes devem ser bem definidas de modo a que as alterações em algumas partes não influenciem seriamente outras.

Linguagem de definição de dados (DDL)

- Notação da especificação da definição do esquema da base de dados
- Exemplo: **create table** *instructor* (
 ID **char**(5),
 name **varchar**(20),
 dept_name **varchar**(20),
 salary **numeric**(8,2))
- O compilador DDL gera um conjunto de modelos de tabela armazenados num **dicionário de dados**
- O dicionário de dados contém metadados (i.e., dados sobre dados)
 - Esquema de base de dados
 - Restrições de integridade
 - Chave primária (ID identifica univocamente cada instrutor)
 - Permissões
 - Quem pode aceder ao que
 -

Linguagem de manipulação de dados (DML)

- Linguagem para aceder e atualizar os dados organizados pelo modelo de dados apropriado
 - DML também conhecido como linguagem de consulta ou interrogações
- Existem basicamente dois tipos de linguagem de manipulação de dados
 - **DML Procedimental** -- requerem que um utilizador especifique quais os dados necessários e como obter esses dados.
 - **DML Declarativo** -- exigir que um utilizador especifique quais os dados necessários sem especificar como obter esses dados.
- Os DMLs declarativos são geralmente mais fáceis de aprender e usar do que os DMLs procedimentais.
- Os DMLs declarativos também são referidos como DMLs não procedimentais.
- A parte de um DML que envolve a recuperação de informações é chamada de linguagem de **consulta (interrogação ou query)**.

Linguagem de consulta SQL

- A linguagem de consulta SQL não é procedimental. Uma consulta tem como entrada várias tabelas (possivelmente apenas uma) e devolve sempre apenas uma única tabela.
- Exemplo para encontrar todos os instrutores do departamento de “Comp. Sci.”

```
select name  
from instructor  
where dept_name = 'Comp. Sci.'
```

- SQL **NÃO** é Turing Completa
- Para ser capaz de calcular funções complexas o SQL é habitualmente incorporada em linguagem de nível superior
- As aplicações geralmente acedem às bases de dados através de:
 - Extensões à linguagem que permitem o uso incorporado do SQL
 - Interfaces de Programação de aplicações (API's, por exemplo, ODBC/JDBC) que permite que as consultas SQL sejam enviadas para uma base de dados

Acesso à base de dados a partir de Aplicações

- Linguagens de consulta não procedimentais, como o SQL, não são tão poderosas como outras linguagens Turing completas.
- A SQL não suporta ações como o *input* dos utilizadores, *output* para apresentação ou comunicação sobre a rede.
- Tais computações e ações devem ser escritas numa outra linguagem, como C/C++, Java ou Python, com consultas SQL incorporadas que acessem aos dados na base de dados.

Desenho de base de dados

O processo de conceção da estrutura geral da base de dados:

- Desenho Lógico – A decidir sobre o esquema da base de dados. O design da base de dados requer que encontremos uma "boa" coleção de esquemas de relação.
 - Decisão de negócio – Que atributos devemos guardar na base de dados?
 - Decisão da Ciência da Computação – Que esquemas de relação devemos ter e como os atributos devem ser distribuídos entre os vários esquemas de relação?
- Desenho Físico – Decidir sobre o layout físico da base de dados

Motor de base de dados

- Um sistema de base de dados é dividido em módulos que lidam com cada uma das responsabilidades do sistema global.
- Os componentes funcionais de um sistema de base de dados podem ser divididos em:
 - O gestor de armazenamento,
 - O processador de consultas,
 - A componente de gestão de transações.

Gestor de Armazenamento - 1

- Um módulo que fornece a interface entre os dados de baixo nível, armazenados na base de dados, e as aplicações e consultas submetidas ao sistema.
- O gestor de armazenamento é responsável pelas seguintes tarefas:
 - Interação com o gestor de ficheiros do sistema operativo
 - Armazenamento, recuperação e atualização eficientes de dados
- Os componentes do gestor de armazenamento incluem:
 - Gestor de autorização e integridade
 - Gestor de transações
 - Gestor de ficheiros
 - Gestor de *buffer*

Gestor de Armazenamento - 2

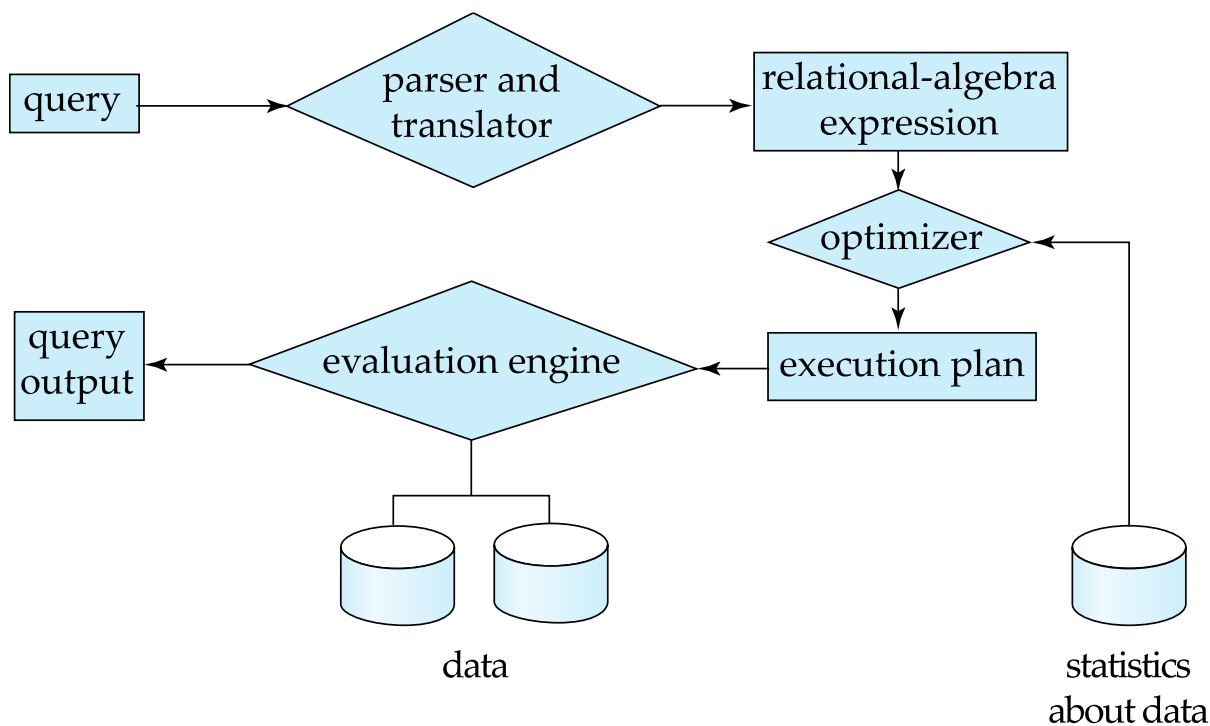
- O gestor de armazenamento implementa várias estruturas de dados como parte da implementação do sistema físico:
 - Ficheiros de dados -- armazenam a própria base de dados
 - Dicionário de dados -- armazena metadados sobre a estrutura da base de dados, em particular o esquema da base de dados.
 - Índices -- pode fornecer acesso rápido a itens de dados. Um índice de base de dados fornece ponteiros para os itens de dados que detêm um valor particular.

Processador de consultas - 1

- Os componentes do processador de consultas incluem:
 - Intérprete DDL -- interpreta declarações DDL e regista as definições no dicionário de dados.
 - Compilador DML -- traduz declarações DML numa linguagem de consulta num plano de avaliação constituído por instruções de baixo nível que o motor de avaliação de consulta entende.
 - O compilador DML executa a otimização de consultas; ou seja, escolhe o plano de avaliação de custos mais baixo entre as várias alternativas.
 - Motor de avaliação de consultas -- executa instruções de baixo nível geradas pelo compilador DML.

Processamento de consultas - 2

1. Análise e tradução
2. Otimização
3. Avaliação



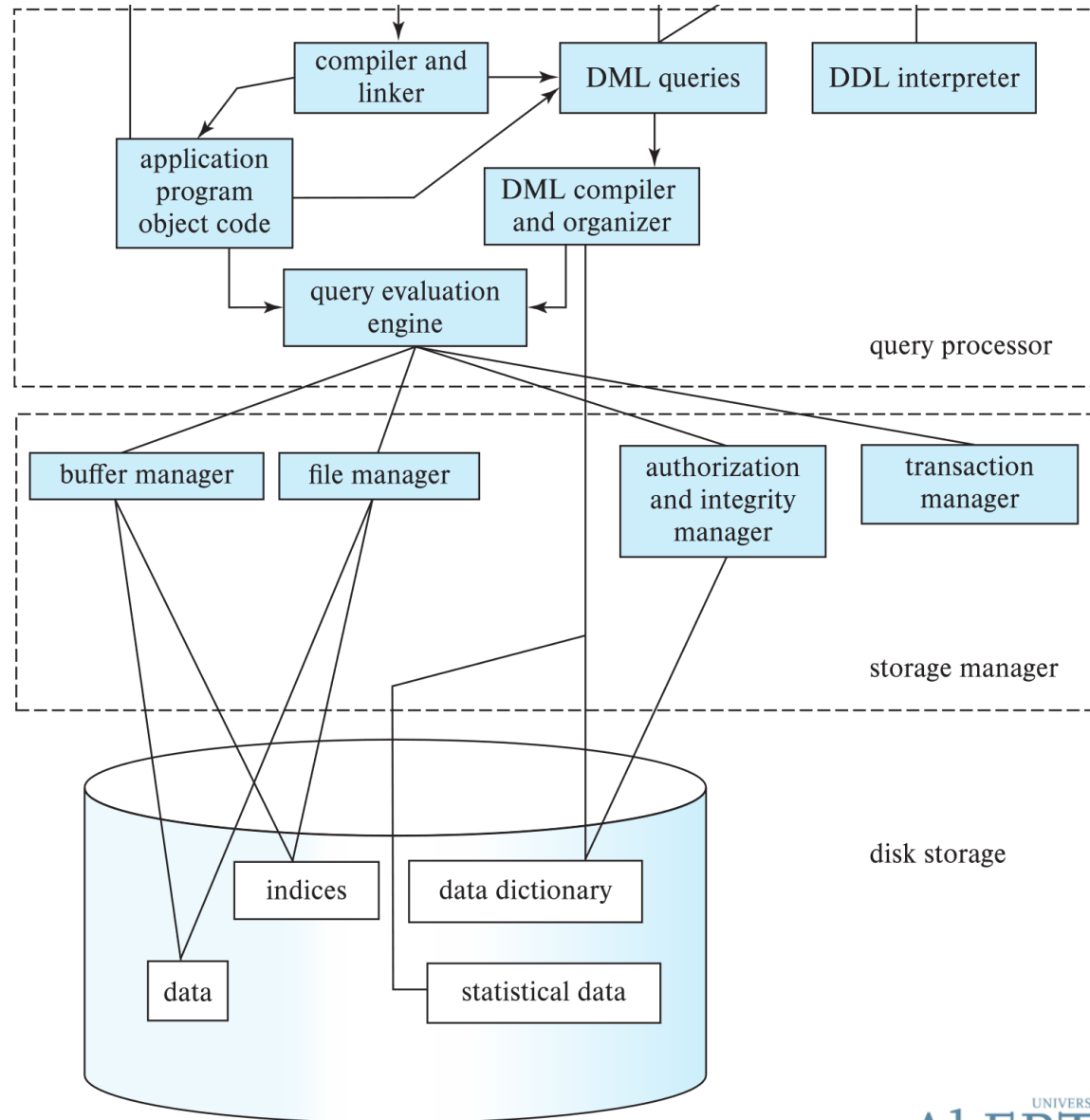
Gestão de Transações

- Uma **transação** é uma coleção de operações que executam uma única função lógica numa aplicação de base de dados
- **Componente de gestão de transações** garante que a base de dados permanece num estado consistente (correto), apesar das falhas do sistema (por exemplo, falhas de energia e falhas no sistema operativo) e falhas de transações.
- **Gestor de controlo da concorrência** controla a interação entre as transações simultâneas, para garantir a consistência da base de dados.

Arquitetura de Base de Dados

- Bases de dados centralizadas
 - Um a alguns núcleos, memória partilhada
- Cliente-servidor,
 - Um servidor executa trabalho em nome de várias máquinas de clientes.
- Bases de dados paralelas
 - Muitos núcleos, memória partilhada
 - Disco partilhado
- Bases de dados distribuídas
 - Distribuição geográfica
 - Heterogeneidade de esquema/dados

Arquitetura da base de dados (Centralizada/Memória Partilhada)

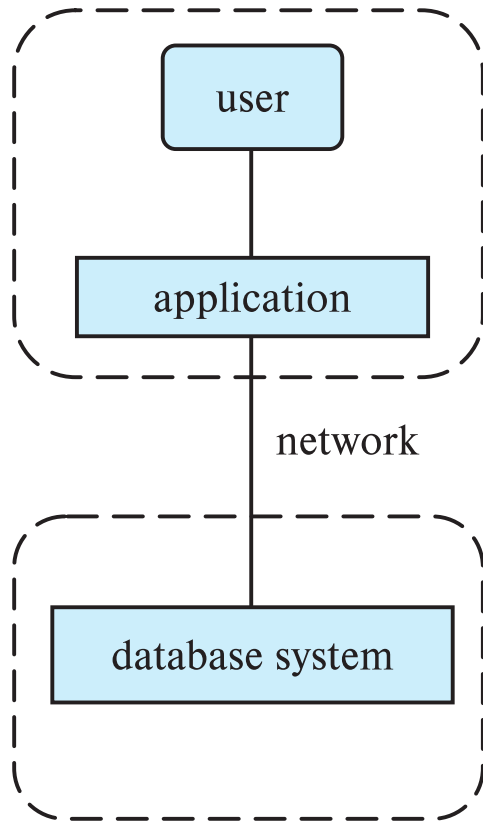


Aplicações de base de dados

As aplicações da base de dados são geralmente divididas em duas ou três camadas

- Arquitetura de duas camadas -- a aplicação reside na máquina do cliente, onde invoca a funcionalidade do sistema de base de dados na máquina do servidor
- Arquitetura de três camadas - a máquina cliente funciona como *front-end* e não contém nenhuma ligação direta à base de dados.
 - O cliente comunica com um servidor de aplicações, geralmente através de uma interface de formulários.
 - O servidor de aplicações, por sua vez, comunica com um sistema de base de dados para aceder aos dados.

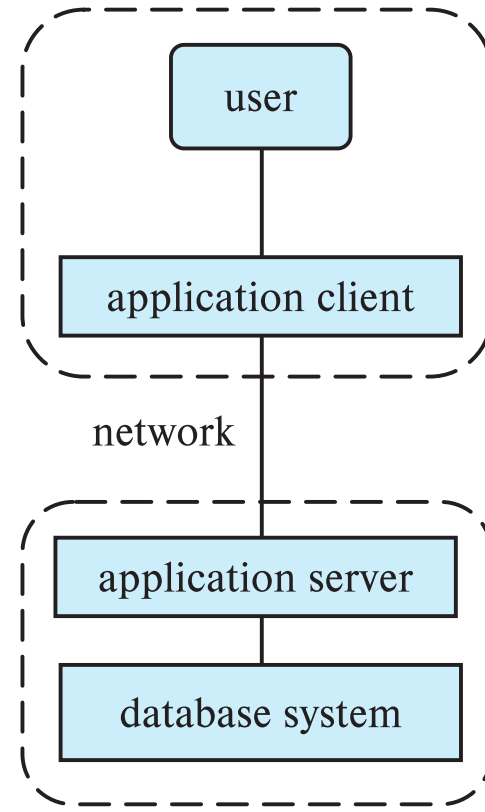
Arquiteturas de duas e três camadas



(a) Two-tier architecture

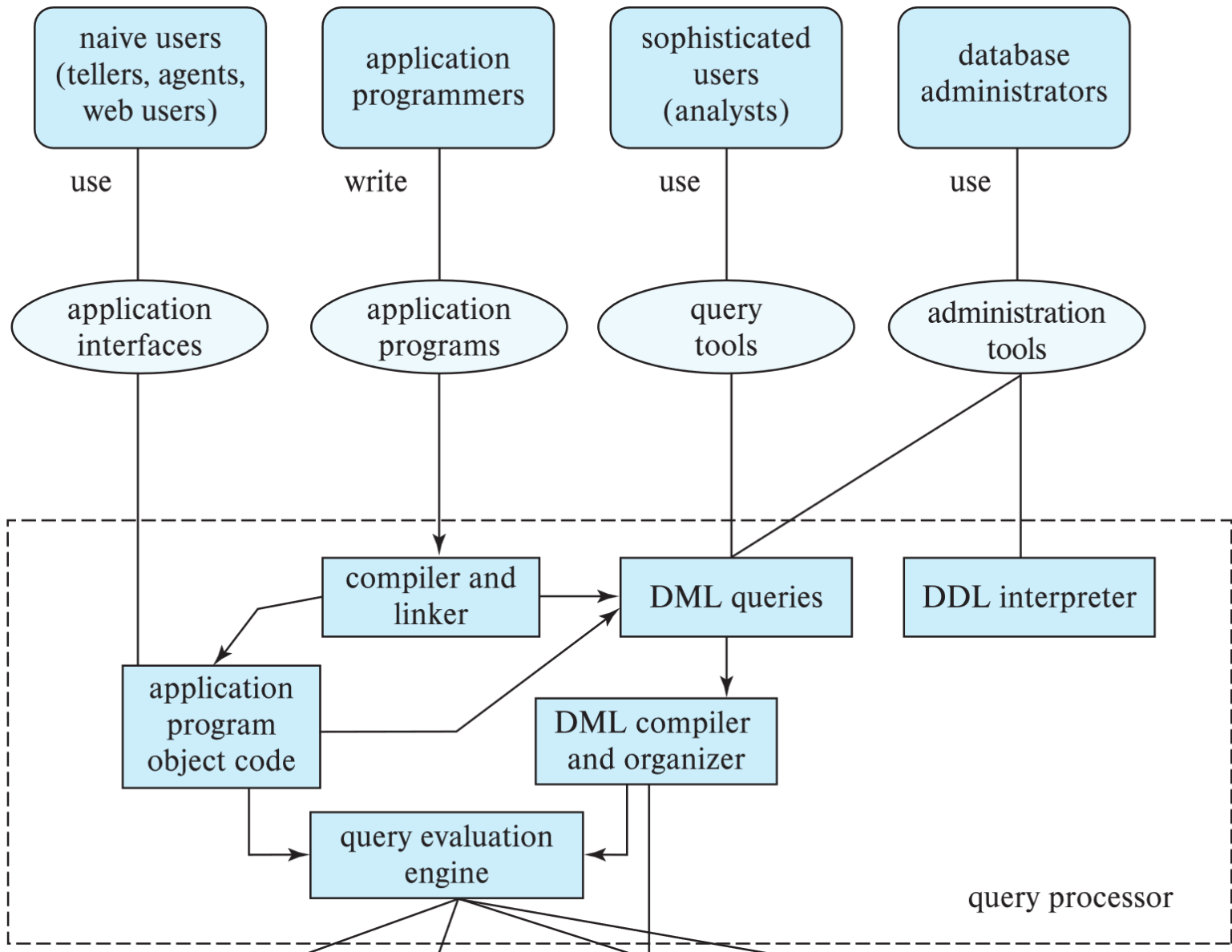
client

server



(b) Three-tier architecture

Utilizadores de bases de dados



Administrador de base de dados

Uma pessoa que tem controlo central sobre o sistema é chamada de **administrador de base de dados (DBA)**. Funções de um DBA incluem:

- Definição do esquema
- Estrutura de armazenamento e definição de método de acesso
- Alteração de esquema e organização física
- Concessão de autorização para acesso a dados
- Manutenção de rotina
- Periodicamente fazer o backup da base de dados
- Garantir que há espaço livre em disco suficiente para operações normais e melhorando o espaço do disco conforme necessário
- Monitorização de trabalhos a correr na base de dados

Histórico de Sistemas de Bases de Dados

- Anos 50 e início dos anos 60:
 - Processamento de dados utilizando fitas magnéticas para armazenamento
 - Fitas fornecem apenas acesso sequencial
 - Cartões perfurados para entrada

- Finais dos anos 60 e 70:
 - Discos rígidos permitiram acesso direto aos dados
 - Modelos de dados de rede e hierárquicos em uso generalizado
 - Ted Codd define o modelo de dados relacionais
 - Ganharia o Prémio ACM Turing por este trabalho
 - IBM Research inicia protótipo do Sistema R
 - UC Berkeley (Michael Stonebraker) inicia protótipo ingres
 - Oracle lança primeira base de dados relacional comercial
 - Processamento de transações de alto desempenho (para a época)

History of Database Systems (Cont.)

- Anos 80:
 - Protótipos relacionais de investigação evoluem para sistemas comerciais
 - SQL torna-se padrão industrial
 - Sistemas de base de dados paralelos e distribuídos
 - Wisconsin, IBM, Teradata
 - Sistemas de base de dados orientados para objetos
- Anos 90:
 - Grandes aplicações de apoio à decisão e de extração de dados
 - Grandes bases de dados com vários terabytes
 - Surgimento do comércio web

History of Database Systems (Cont.)

- Anos 2000
 - Big data
 - Google BigTable, Yahoo PNuts, Amazon,
 - Sistemas “NoSQL”.
 - Análises Big data: beyond SQL
 - Map reduce
- Anos 2010
 - SQL reloaded
 - Sistemas com SQL front end para Map Reduce
 - Sistemas de base de dados massivamente paralelos
 - Bases de dados Multi-core main-memory

21053 - Fundamentos de Bases de Dados

Professor:

Paulo Pombinho

Bom Estudo!

