

Bases de Dados



2018/19

<http://bd.ssdi.di.fct.unl.pt>

Carlos Viegas Damásio (cd@fct.unl.pt)
João Leite e José Alferes

Equipa Docente



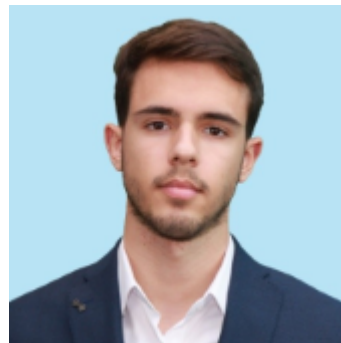
José Júlio Alferes
Teóricas e Regência



Matthias Knorr



Nuno Marques



Igor Ruivo

Funcionamento e Avaliação

A thick, yellow, brushstroke-style underline that spans the width of the slide, positioned directly beneath the title text.

Funcionamento

- As aulas teóricas vão funcionar de forma concentrada, mais no início do semestre
 - ✦ 4 horas por semana em vez de 3, e a terminar a 14 de maio
 - ✦ Dois turnos, a que os estudantes podem ir de forma livre
- Esta forma de funcionamento permite:
 - ✦ Uma melhor distribuição do esforço na UC entre teóricas e trabalho prático
 - ✦ Que o trabalho final da UC arranque mais cedo, já com toda a matéria relevante dada nas teóricas
 - ✦ Que os estudantes coordenem melhor o esforço em articulação com as restantes UC
- Mas é muito importante que não *"percam o comboio"* no início do semestre!

Funcionamento das práticas

- Ao todo, 8 turnos de
- As práticas começam ainda esta semana, a 7, 8 ou 11 de março (consoante o turno)
- De início as práticas são de exercícios relativamente à matéria dada nas teóricas
 - ✦ As primeiras até são de “papel e lápis”
 - ✦ Mas não se preocupem! Depois terão tempo para mexer no computador e fazer coisas/aplicações
- Depois, em especial após as teóricas terminarem, as práticas serão para elaboração do projeto final da UC

Avaliação

■ Componente teórica

- ✦ 2 testes, todos com o mesmo peso
- ✦ Ou exame de recurso
- ✦ O primeiro teste é no dia 8 de abril!

■ Componente prática

- ✦ Projeto final, de construção de uma base de dados completa
- ✦ A fazer em grupos de exatamente 3 estudantes, todos inscritos no mesmo turno de práticas
- ✦ Com várias fases (ver detalhes em <http://bd.ssdi.di.fct.unl.pt>)
- ✦ Duas entregas: 10 de abril e 28 de maio
 - ❖ A nota da componente é a média de cada uma destas entregas, com a nota da discussão (cada um a valer 1/3)

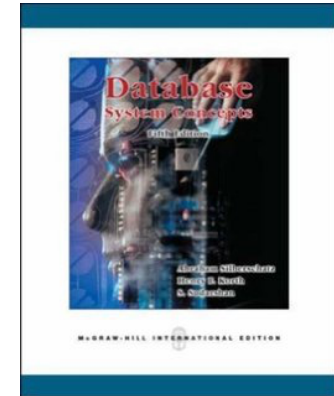
■ Nota final

- ✦ Teórica 60% + Prática 40%
- ✦ Aprovação com nota final ≥ 10 valores e nota da componente teórica ≥ 10 valores (ambas as notas arredondadas às unidades)

Recursos

■ Recursos online

- ✳ CLIP
- ✳ Página da UC (<http://bd.ssdi.di.fct.unl.pt>)
- ✳ Google Calendar



- Em todos eles poderão encontrar os vários materiais, avisos, regras, etc

- ★ Bibliografia, Slides, Fichas de Exercícios, Outros

Bases de Dados - FCT NOVA

Ano letivo 2016/17

Bases de Dados é uma unidade curricular obrigatória de 9 créditos ECTS, do 4º semestre do [Mestrado Integrado em Engenharia Informática da FCT NOVA](#).

Datas Importantes (a confirmar)

- 6 de março - Início das aulas teóricas
- 9 de março - Início das aulas práticas
- 22 de março - Registo dos grupos
- 29 de março - 1º teste
- 5 de abril - Registo do tema do projeto
- 19 de abril - Entrega da 1ª fase do projeto
- 29 de abril - 2º teste
- 17 de maio - 3º teste
- 28 de maio - Entrega da 2ª fase do projeto
- 12, 14 e 16 de junho - Apresentação dos projetos

Início ▶▶▶

- Objetivos
- Programa
- Bibliografia
- Docentes
- Avaliação
- Teóricas
- Práticas
- Projeto
- Baú
- Site Map

AVISO

As aulas começam logo no dia 6 de março às 9 da manhã!

E as práticas começam também na 1ª semana de aulas!

© 2017 FCT-NOVA, José Júlio Alferes [Contact Me](#)

Today   **6 - 12 Mar 2017** 

	Mon 6/3	Tue 7/3	Wed 8/3	Thu 9/3	Fri 10/3	Sat 11/3	Sun 12/3
09:00	09:00 - 11:00 BD - T1		09:00 - 11:00 BD - T1	09:00 - 09:00 - P1 P2	09:00 - 09:00 - P10 P7		
10:00							
11:00	11:00 - 13:00 BD - T2		11:00 - 13:00 BD - T2	11:00 - 11:00 - P3 P4	11:00 - 13:00 P8		
12:00				P6 When Thu, 9 March, 15:00 - 17:00 Description Exercícios de Modelo ER more details» copy to my calendar»			
13:00							
14:00							
15:00				15:00 - 15:00 - P5 P6			
16:00							
17:00							

Events shown in time zone: Lisbon

 Google Calendar

Algumas Recomendações

■ Erros habituais:

- ✱ As aulas práticas são suficientes

 - ❖ Não o são, de todo!

- ✱ Basta ir às aulas e dar uma vista de olhos pelos acetatos nas vésperas dos testes/exame

■ Aos estudantes que só vêm às primeiras três/quatro aulas teóricas

- ✱ Não é boa ideia! ...pelo menos, estatisticamente, não tem sido.

- ✱ Devem ler o livro com muita atenção para perceber bem os conceitos. **Ler os acetatos não chega!**

Algumas Recomendações

- A matéria não é complicada.
 - ✦ No entanto, é exigido um conhecimento **profundo** e **rigoroso**.
 - ✦ Este conhecimento **não se adquire na véspera** do teste/exame
 - ✦ Requer um certo amadurecimento que só se obtém com o acompanhamento da matéria, e estudo ao longo do semestre

Datas importantes

- Formação de grupos: 18 de março de 2019
- Registo de tema: 26 de março de 2019
- Primeiro teste: 8 de abril de 2019
- Primeira entrega do projecto: 10 de abril de 2019
- Entrega final do projecto: 28 de maio de 2019
- Segundo teste: 3 de junho de 2019

Introdução

A thick, yellow, horizontal bar with slightly irregular, hand-drawn edges, spanning across the width of the slide below the title.

Objetivo:

Dotar os estudantes das bases necessárias à conceção, construção, manipulação e análise de Bases de Dados relacionais.

Aplicações de Bases de Dados

■ Como é que fariam para lidar com as seguintes aplicações?

- ✦ Sistema académico (e.g. CLIP)
- ✦ Clientes de um banco, contas bancárias, movimentos, etc
- ✦ Facebook

■ Problemas

- ✦ Como lidar com vários utilizadores a aceder simultaneamente a dados das aplicações?
- ✦ Como lidar com vários programas a aceder/manipular os dados da aplicação?
 - ❖ Programas feitos por pessoas diferentes, e em linguagens diferentes?

Foco nos Dados

- Os programas não são “donos” dos dados
 - ✦ A ideia de programa, que tem e manipula os seus próprios dados, independentemente do resto, não se aplica de todo!
 - ❖ E.g. as estruturas de dados para acesso rápido deixam de poder ser geridas pelos programas individuais
- O mais importante nestas aplicações são os dados!
 - ✦ Tudo o resto (programas, utilizadores) gira à volta dos dados

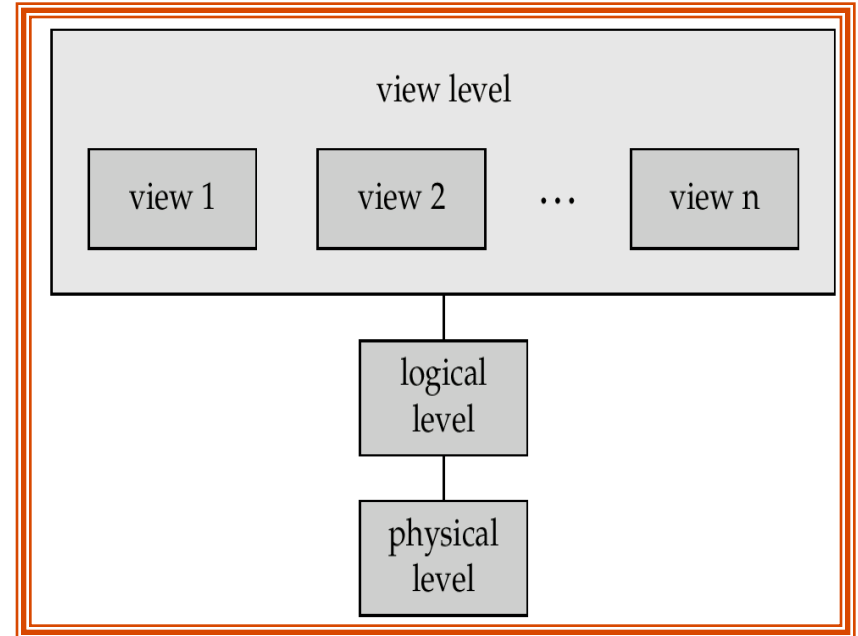


Desafios

- Se os dados são para ser usados por diferentes programas, eventualmente em linguagens diferentes:
 - ✦ Como é que todos podem “compreender” os dados?
 - ✦ Como é que todos lhes podem aceder?
 - ✦ Como é que se garante que ninguém “estraga” os dados?
- É crucial que haja alguma abstração na representação dos dados!
- É crucial que haja algo entre os dados e os programas, que trata de:
 - ✦ implementar e fornecer essa abstração;
 - ✦ garantir a integridade dos dados;
 - ✦ fornecer formas de acesso aos dados
- Sistema de Gestão de Bases de Dados

Níveis de Abstracção

- **Nível físico:** descreve como um registo (e.g., cliente) é armazenado.
- **Nível lógico:** descreve os dados armazenados na base de dados, assim como as relações entre os dados.
- **Nível das vistas:** as aplicações ocultam os detalhes dos tipos de dados. As vistas também podem esconder informação por motivos de segurança.



- Semelhante a tipos de dados nas linguagens de programação
 - **Nível físico:** blocos de dados no disco.
 - **Nível lógico:** tipos e relações entre tipos, usados ao nível da programação.
 - **Nível das vistas:** aplicações que escondem os tipos do utilizador.

Instâncias e Esquemas

- Os dados variam ao longo do tempo...
- **Instância** – o conteúdo de uma base de dados num instante de tempo
 - Análogo ao valor de uma variável
- **Esquema** – a estrutura lógica da base de dados
 - e.g., a base de dados é constituída por informação sobre clientes, contas e as relações entre si.
 - Análogo ao tipo de uma variável num programa
 - **Esquema físico**: desenho da base de dados ao nível físico
 - **Esquema lógico**: desenho da base de dados ao nível lógico
- **Independência física dos dados** – a capacidade de modificar o esquema físico sem alterar o esquema lógico
 - As aplicações dependem do esquema lógico
 - Em geral, as interfaces no SGBD entre os vários níveis e componentes devem estar bem definidas de modo a que alterações numa parte não influenciem grandemente outras partes.

Modelos de Dados

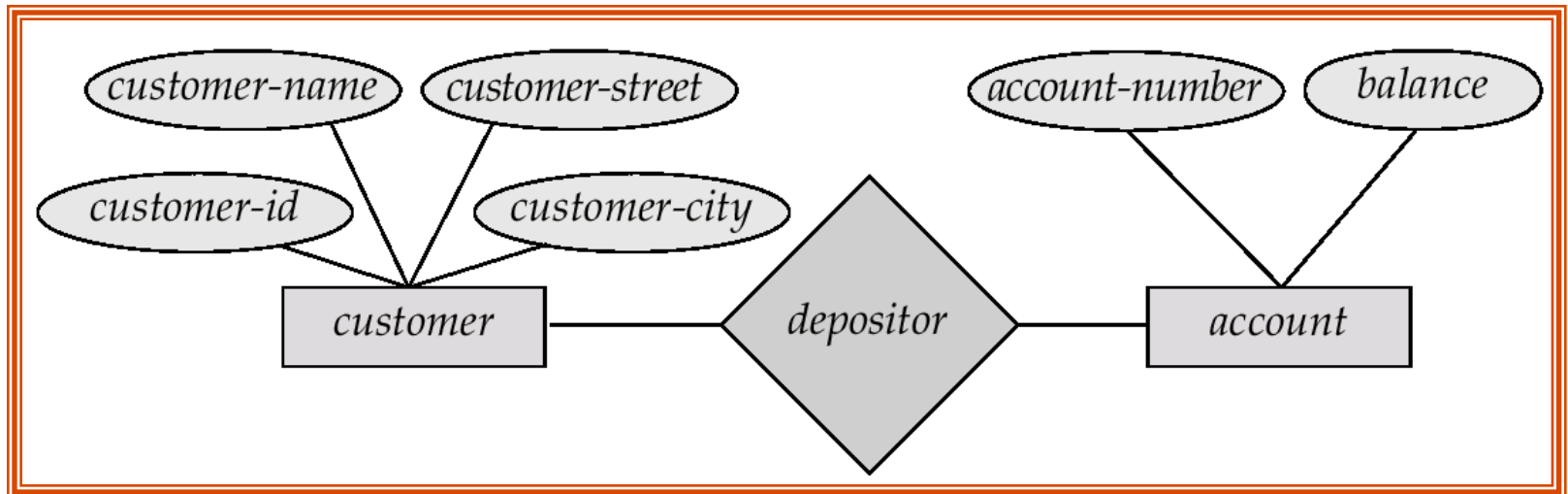
- Um conjunto de ferramentas para descrever
 - dados
 - relações entre dados
 - semântica dos dados
 - restrições sobre os dados
- Exemplos:
 - ✦ Modelo Entidade-Relações (para desenho de Bases de Dados)
 - ✦ Modelo Relacional
 - ✦ Modelos de dados baseados em objetos
 - ✦ Modelos de dados semiestruturados (XML)
 - ✦ Outros modelos:
 - Modelo hierárquico
 - Modelo em rede

Modelo de Entidades e Relações

- Modelização ER do mundo
 - Entidades (objetos)
 - E.g. clientes, contas, agências, etc
 - Relações entre entidades
 - E.g., a conta A-101 pertence ao cliente Johnson
 - O conjunto de associações *depositante* associa clientes a contas
- Amplamente utilizado no desenho de bases de dados
 - O desenho de base de dados no modelo ER é normalmente convertido para o desenho no modelo relacional (a seguir) que é utilizado como abstração para processamento

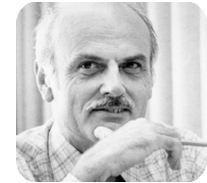
Modelo ER (cont.)

Exemplo de um esquema no modelo ER



Modelo Relacional

- Os dados e as associações entre eles são representados por um conjunto de relações (semelhante a tabelas)



Ted Codd
Turing Award 1981

<i>Customer-id</i>	<i>customer-name</i>	<i>customer-street</i>	<i>customer-city</i>	<i>account-number</i>
192-83-7465	Johnson	Alma	Palo Alto	A-101
019-28-3746	Smith	North	Rye	A-215
192-83-7465	Johnson	Alma	Palo Alto	A-201
321-12-3123	Jones	Main	Harrison	A-217
019-28-3746	Smith	North	Rye	A-201

Exemplo de Base de Dados Relacional

<i>customer-id</i>	<i>customer-name</i>	<i>customer-street</i>	<i>customer-city</i>
192-83-7465	Johnson	12 Alma St.	Palo Alto
019-28-3746	Smith	4 North St.	Rye
677-89-9011	Hayes	3 Main St.	Harrison
182-73-6091	Turner	123 Putnam Ave.	Stamford
321-12-3123	Jones	100 Main St.	Harrison
336-66-9999	Lindsay	175 Park Ave.	Pittsfield
019-28-3746	Smith	72 North St.	Rye

(a) The *customer* table

<i>account-number</i>	<i>balance</i>
A-101	500
A-215	700
A-102	400
A-305	350
A-201	900
A-217	750
A-222	700

(b) The *account* table

<i>customer-id</i>	<i>account-number</i>
192-83-7465	A-101
192-83-7465	A-201
019-28-3746	A-215
677-89-9011	A-102
182-73-6091	A-305
321-12-3123	A-217
336-66-9999	A-222
019-28-3746	A-201

(c) The *depositor* table

Integridade dos Dados

- O modelo de dados deve conter restrições que garantam a integridade dos dados
 - ✦ E o SGBD deve garantir essa integridade
- Exemplos:
 - ✦ Cada cliente de um banco deve ter no máximo uma data de nascimento
 - ✦ Uma conta pode pertencer a vários clientes, mas está associada a uma única sucursal do banco
 - ✦ Um estudante só se pode inscrever a UCs oferecidas ao curso em que está matriculado
- Como é que isso se pode garantir?
 - ✦ Por exemplo, garantindo que só há um local para guardar a data de nascimento de cada cliente
 - ❖ Se e.g. os dados do cliente, incluindo data de nascimento, forem guardados nas contas:
 - Há redundância (espaço gasto desnecessariamente)
 - É fonte de inconsistência (se se muda a data numa conta e não noutra)

Primeiro Objetivo

- Dado um problema, saberem conceber um modelo, e construir uma Bases de Dados relacional que:
 - ✦ Permita armazenar os dados relativos ao problema
 - ✦ Garanta a integridade dos dados
 - ✦ Permita a vários programas aceder aos dados sem perigo de “estragarem”
- Esta é a matéria do 1º teste (8 de abril), e para a 1ª fase do projeto (10 de abril)

Objetivo:

Dotar os estudantes das **bases necessárias à conceção, construção**, manipulação e análise **de Bases de Dados relacionais**.

Acesso e manipulação de Dados

- Mas como é que depois os programas acedem aos dados, se nem sequer “sabem” como é que eles estão implementados fisicamente?
- Linguagens Declarativas para acesso a dados
 - ✦ **A ideia** é dizer que dados se quer obter/introduzir/editar/apagar
 - ✦ **Não é** dizer como é que se obtêm/introduzem/editam/apagam os dados
- Essas linguagens devem
 - ✦ ser independentes da implementação física do armazenamento dos dados
 - ✦ possibilitar o uso por vários tipos de linguagens, e até de consola
- O SGBD é responsável pela implementação destas linguagens

Exemplo do SQL

■ SQL: linguagem declarativa de uso generalizado

- E.g. encontrar o nome de cliente com customer-id 192-83-7465

```
select  customer.customer-name  
from    customer  
where customer.customer-id = '192-83-7465'
```

- E.g. procurar os saldos de todas as contas detidas pelo cliente com customer-id 192-83-7465

```
select  account.balance  
from    depositor, account  
where depositor.customer-id = '192-83-7465' and  
        depositor.account-number = account.account-number
```

■ As aplicações geralmente acedem a bases de dados por intermédio de:

- Extensões às linguagens permitindo SQL embutido:
- Interface de aplicações (e.g. ODBC/JDBC) permitindo o envio de consultas SQL para a base de dados

Segundo Objetivo

- Dada uma Bases de Dados relacional saber:
 - ✦ Aceder aos dados dessa Base de Dados
 - ✦ Alterar os dados
 - ✦ Extrair indicadores (complexos) a partir desses dados
- Esta é a matéria essencial do 2º teste (a 3 de junho)

Objetivo:

Dotar os estudantes das bases necessárias à conceção, construção, **manipulação e análise de Bases de Dados relacionais**.

Então e depois?

- Como é que os programas podem aceder aos dados e comunicar com bases de dados?
 - ✦ Embedded SQL, Open DataBase Connectivity (ODBC) e API Java (JDBC)
- E exatamente como é que se lida com acessos concorrentes?
 - ✦ Problemas de atomicidade e isolamento
 - ✦ Transações
- O modelo relacional é de longe o mais usado. Mas há outros.
 - ✦ Vamos ver dois:
 - ❖ Modelo Object-Relacional (tabelas com tabelas lá dentro)
 - ❖ XML e dados semiestruturados
 - Especialmente adequado para transferência de dados
 - A ideia é juntar tudo o que é relativo a uma entidade, em vez de ter em tabelas separadas

Programa

- Introdução (1 aula – esta)
- Modelos de dados
 - ✧ Modelo de Entidades e Relações (2 aulas)
 - ✧ Modelo Relacional (1 aula)
- Normalização de Bases de Dados
 - ✧ Dependências funcionais e multi valor (1½ aula)
 - ✧ Formas normais: 3ª, 4ª e de Boyce-Cood (1½ aula)
- Linguagens de interrogação e manipulação de bases de dados
 - ✧ Álgebra relacional (2 aulas)
 - ✧ SQL (3 aulas)
 - ✧ Outras linguagens (1 aula)

1º Teste

- Integridade de Bases de Dados
 - ✧ Integridade de referência (½ aula)
 - ✧ Asserções e triggers (½ aula)
- Interação com bases de dados (2 aulas)
 - ✧ Embedded SQL, ODBC, JDBC
 - ✧ Segurança e autorizações
 - ✧ Transações
- Discussão de outros modelos de bases de dados
 - ✧ Bases de dados objeto/relacional (1 aula)
 - ✧ XML (2 aulas)

2º Teste

Aulas Práticas

■ 1º teste

- ✦ Exercícios de modelo ER (2 aula)
- ✦ Exercícios de Normalização de BDs (1 aula)
- ✦ Exercícios de álgebra relacional (1 aula)

■ 2º teste

- ✦ Exercícios de SQL (3 aulas)
- ✦ Introdução ao Oracle® Application Express (2 aulas)
- ✦ Exercícios de Transações e XML (1 aula)
- ✦ Implementação do trabalho prático (1 aula)

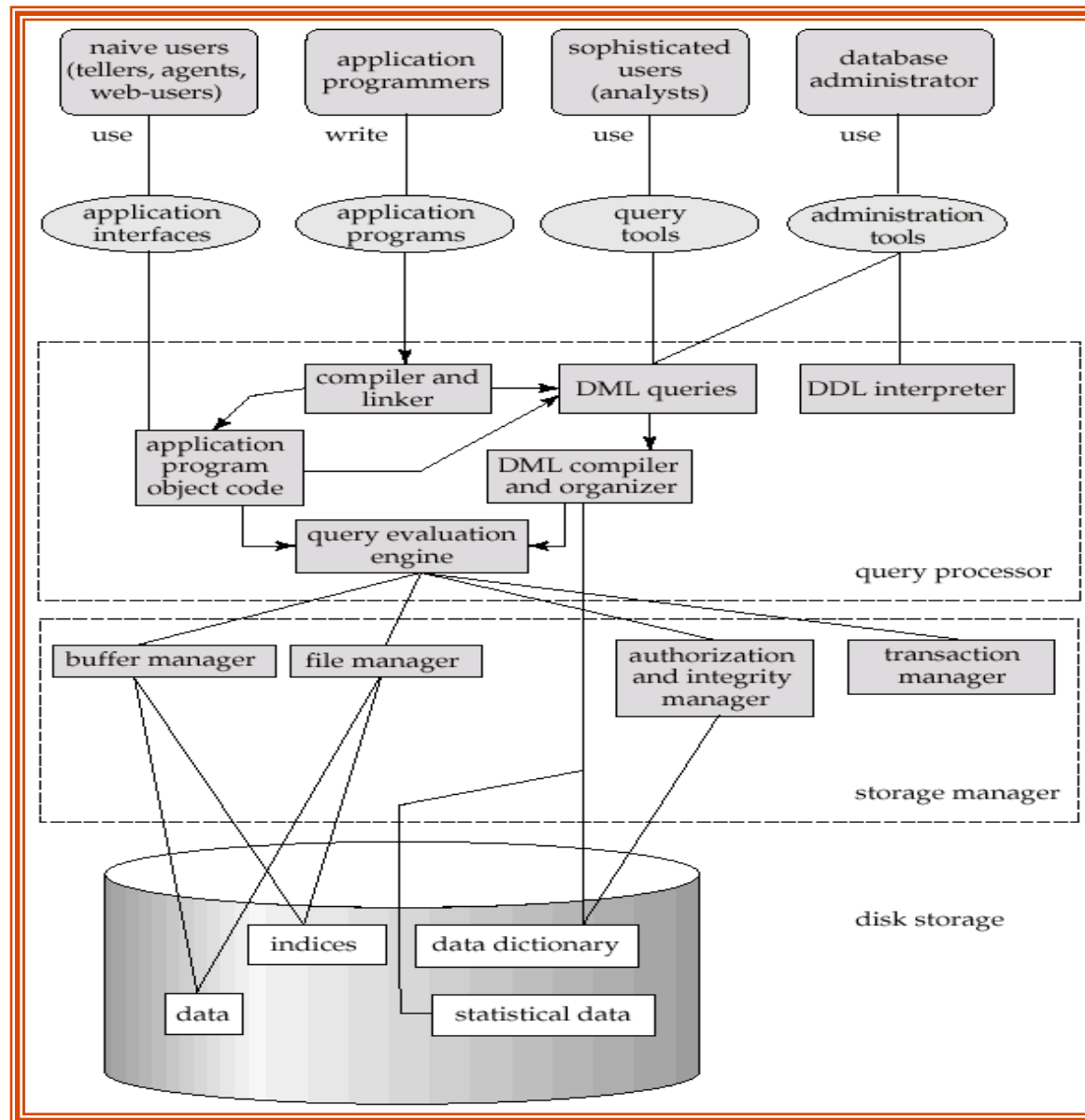
O que fica de fora

- E se a grande base de dados fosse a Web?
 - ✦ Bases de dados heterogêneas e distribuídas
 - ✦ Como fazer modelos disto (tabelas não serve de certeza)?
 - ✦ Como interrogar os dados nesses modelos?
 - ✦ Isto fica para *“Representação de Conhecimento”* e *“Modelação de dados”*
- E se eu quiser fazer análise de históricos de dados?
 - ✦ Convinha ter um armazém de dados, que permitisse fazer essas análises de alto nível
 - ✦ Como fazer a modelação nesses casos?
 - ✦ Isto fica para *“Modelação de dados”* e *“Prospecção de Dados”*
- E como se visualizam dados com relações complexas entre eles?
 - ✦ Isso fica para *“Visualização de Dados”*

O que mais fica de fora

- E então se os dados estiverem sempre a chegar?
 - ✦ E.g. redes de sensores, dados de mercados
 - ✦ Neste caso o que temos é tabelas em evolução, e os resultados também estão em evolução
 - ✦ Isto fica para *“Processamento de Streams”*
- E se houver muitos dados geográficos?
 - ✦ Será que não há formas específicas de tratar este tipo de dados?
 - ✦ Há e isso é estudado em *“Tecnologias de Informação Geográfica”*
- E como raio se implementam mesmo os SGBDs?
 - ✦ Um Eng. Informático deve saber isto!
 - ✦ E isto ajuda no Tuning de BDs (para as tornar mais eficientes)
 - ✦ Isso fica para *“Sistemas de Bases de Dados”*

Um SGBD por dentro



Gestão do Armazenamento

- O gestor do armazenamento é um módulo de programa que fornece uma 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
 - armazenamento, recuperação e alteração eficientes dos dados

Processamento de consultas

■ Módulos:

- Interpretação e tradução
- Otimização
- Avaliação

■ Formas alternativas de avaliação de uma dada consulta

- Expressões equivalentes
- A diferença de custo entre uma boa e uma má forma de avaliação de uma consulta pode ser enorme

■ É necessário estimar o custo das operações

- Depende de dados estatísticos mantidos pela Base de Dados sobre as relações que armazena
- Necessita estimar estatísticas para resultados intermédios para calcular o custo de uma expressão complexa

Gestão de Transacções

- Uma *transação* é um conjunto de operações que efetuam uma operação lógica na aplicação de base de dados
- A componente de gestão de transações garante que a base de dados se mantém num estado consistente (correto) apesar de falhas no sistema (e.g., falta de energia elétrica e paragens abruptas do sistema operativo) e de transações falhadas.
- O gestor de controlo de concorrência coordena a interação entre transações concorrentes para garantir a consistência da base de dados.

Utilizadores da Base de Dados

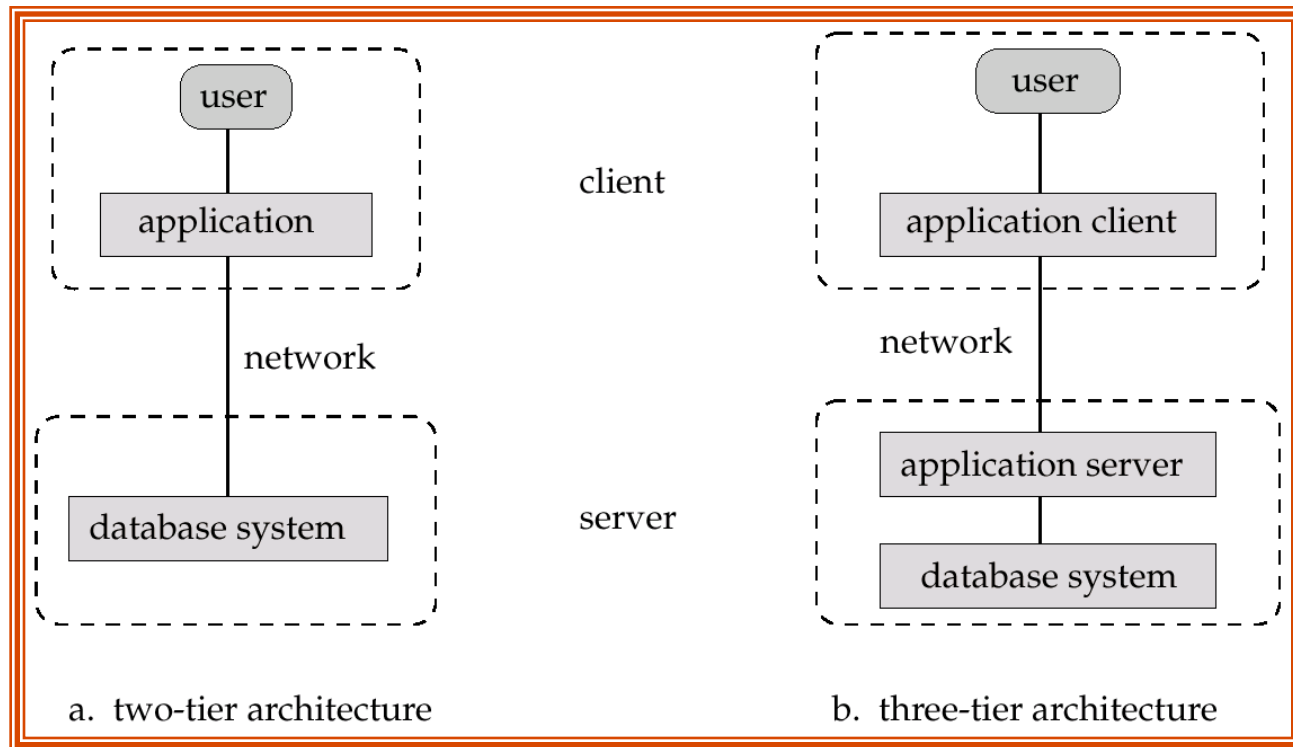
Os utilizadores diferenciam-se pela forma esperada de interacção com o sistema

- **Programadores de aplicações** – interagem com o sistema através de chamadas DML
- **Utilizadores sofisticados** – constroem pedidos numa linguagem de consulta a bases de dados
- **Utilizadores especializados** – escrevem aplicações de bases de dados especializadas que não se enquadram no espírito do processamento de dados tradicional
- **Utilizadores** – chamam uma das aplicações que foi construída previamente
 - E.g. pessoas a acederem a uma base de dados através da Web, caixas, pessoal de secretariado

Administrador da Base de Dados

- Coordena todas as atividades do sistema de base de dados; o administrador da base de dados compreende bem dos recursos e necessidades de informação da empresa.
- As funções do administrador de bases de dados incluem:
 - Definição do esquema
 - Definição dos métodos de acesso e estrutura de armazenamento
 - Modificação do esquema e da organização física
 - Dar aos utilizadores autorizações de acesso à base de dados
 - Especificar restrições de integridade
 - Servir de ligação entre os utilizadores
 - Monitorar a performance e responder a alterações nos requisitos.

Arquiteturas de Aplicação



- **Arquitetura de duas camadas:** E.g. programas clientes recorrendo a ODBC/JDBC para comunicar com a base de dados
- **Arquitetura de três camadas:** E.g. aplicações web e aplicações construídas recorrendo a "software intermediário", e.g. APEX)

Só as relacionais

☐ include secondary database models

137 systems in ranking, March 2019

Rank			DBMS	Database Model	Score		
Mar 2019	Feb 2019	Mar 2018			Mar 2019	Feb 2019	Mar 2018
1.	1.	1.	Oracle	Relational, Multi-model	1279.14	+15.12	-10.47
2.	2.	2.	MySQL	Relational, Multi-model	1198.25	+30.96	-30.62
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	1047.85	+7.79	-56.94
4.	4.	4.	PostgreSQL	Relational, Multi-model	469.81	-3.75	+70.46
5.	5.	5.	IBM Db2	Relational, Multi-model	177.20	-2.23	-9.47
6.	6.	6.	Microsoft Access	Relational	146.20	+2.18	+14.26
7.	7.	7.	SQLite	Relational	124.87	-1.29	+10.06
8.	8.	9.	MariaDB	Relational, Multi-model	84.31	+0.89	+21.21
9.	9.	8.	Teradata	Relational	75.22	-0.75	+2.76
10.	10.	11.	Hive	Relational	73.00	+0.71	+16.00
11.	11.	12.	FileMaker	Relational	58.13	+0.34	+3.00
12.	13.	10.	SAP Adaptive Server	Relational	56.03	+0.29	-6.58
13.	12.	13.	SAP HANA	Relational, Multi-model	55.51	-1.03	+6.99
14.	14.	15.	Microsoft Azure SQL Database	Relational, Multi-model	27.93	+0.81	+3.31
15.	15.	14.	Informix	Relational, Multi-model	26.90	+0.54	-0.61
16.	16.	16.	Vertica	Relational, Multi-model	22.07	-0.74	+1.77
17.	17.	19.	Amazon Redshift	Relational	20.89	-0.10	+6.70
18.	19.	17.	Firebird	Relational	20.17	+0.82	+2.14
19.	18.	18.	Netezza	Relational	19.63	-0.15	+2.64
20.	20.	21.	Google BigQuery	Relational	19.57	+0.82	+7.08
21.	21.	22.	Spark SQL	Relational	16.31	-0.25	+5.23

db-engines.com

345 systems in ranking, March 2019

Rank			DBMS	Database Model	Score		
Mar 2019	Feb 2019	Mar 2018			Mar 2019	Feb 2019	Mar 2018
1.	1.	1.	Oracle	Relational, Multi-model	1279.14	+15.12	-10.47
2.	2.	2.	MySQL	Relational, Multi-model	1198.25	+30.96	-30.62
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	1047.85	+7.79	-56.94
4.	4.	4.	PostgreSQL	Relational, Multi-model	469.81	-3.75	+70.46
5.	5.	5.	MongoDB	Document	401.34	+6.24	+60.82
6.	6.	6.	IBM Db2	Relational, Multi-model	177.20	-2.23	-9.47
7.	9.	7.	Microsoft Access	Relational	146.20	+2.18	+14.26
8.	7.	8.	Redis	Key-value, Multi-model	146.12	-3.32	+14.90
9.	8.	9.	Elasticsearch	Search engine, Multi-model	142.79	-2.46	+14.25
10.	10.	11.	SQLite	Relational	124.87	-1.29	+10.06
11.	11.	10.	Cassandra	Wide column	122.80	-0.58	-0.69
12.	12.	15.	MariaDB	Relational, Multi-model	84.31	+0.89	+21.21
13.	13.	13.	Splunk	Search engine	83.10	+0.29	+17.44
14.	14.	12.	Teradata	Relational	75.22	-0.75	+2.76
15.	15.	18.	Hive	Relational	73.00	+0.71	+16.00
16.	16.	14.	Solr	Search engine	60.01	-0.95	-4.80
17.	17.	17.	HBase	Wide column	58.80	-1.48	-2.14
18.	18.	19.	FileMaker	Relational	58.13	+0.34	+3.00
19.	20.	16.	SAP Adaptive Server	Relational	56.03	+0.29	-6.58
20.	19.	20.	SAP HANA	Relational, Multi-model	55.51	-1.03	+6.99
21.	21.	21.	Amazon DynamoDB	Multi-model	54.49	-0.45	+12.04
22.	22.	22.	Neo4j	Graph	48.58	+0.72	+7.68

História

■ 1950s e princípio 1960s:

- ✦ Processamento de dados utilizando bandas magnéticas para armazenamento
 - ❖ Bandas magnéticas só permitem acesso sequencial
- ✦ Cartões perfurados para input

■ Finais 1960s e 1970s:

- ✦ Discos magnéticos permitem acesso direto aos dados
- ✦ Modelos em rede e hierárquicos amplamente utilizados
- ✦ Ted Codd define o modelo relacional
 - ❖ Ganhará o ACM Turing Award por este trabalho
 - ❖ IBM Research inicia o desenvolvimento do protótipo System R
 - ❖ UC Berkeley (Michael Stonebraker) inicia o protótipo Ingres
 - ❖ Oracle lança a primeira base de dados relacional
- ✦ Processamento eficiente de transacções (à época)

História (cont.)

■ 1980s:

- ★ Protótipos de investigação evoluem para sistemas comerciais
 - ❖ SQL torna-se um standard industrial
- ★ SGBD paralelos e distribuídos
 - ❖ Wisconsin, IBM, Teradata
- ★ SGBD orientados a objetos

■ 1990s:

- ★ Grandes aplicações de apoio à decisão e de mineração de dados
- ★ Grandes armazéns de dados com vários terabyte
- ★ Emergência do comércio electrónico

História (cont.)

■ 2000s

- ★ Sistemas de armazenamento Big data
 - ❖ Google BigTable, Yahoo PNuts, Amazon,
 - ❖ Sistemas “NoSQL”.
- ★ Análise Big data : para além de SQL
 - ❖ Map reduce e variantes

■ 2010s

- ★ SQL novamente
 - ❖ SQL front end para sistemas Map Reduce
 - ❖ SGBD massivamente paralelos
 - ❖ Bases de dados em memória multi-core