

■ Aula 02

- Fundamentos e conceitos básicos
- Sistema de Gerenciamento de Banco de Dados
- Tipos de banco de dados
- A importância do projeto de banco de dados
- Sistemas de arquivos



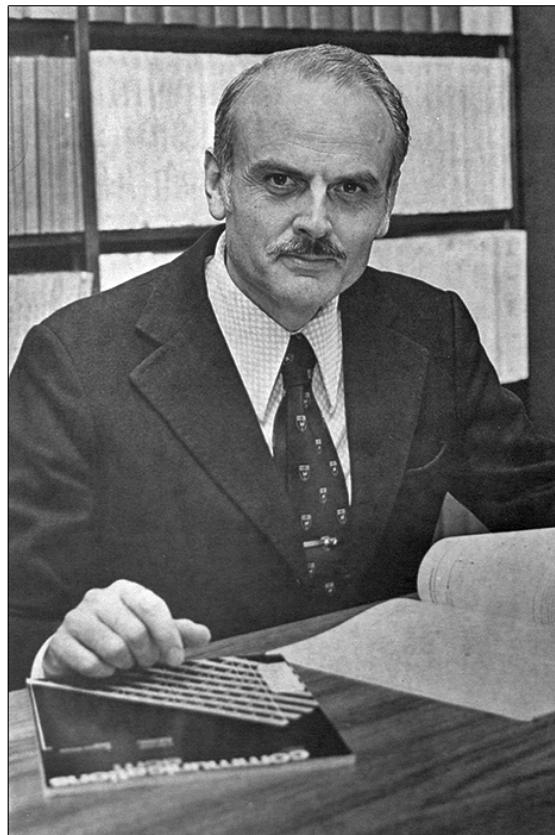
■ Introdução

- Banco de dados e sistemas de banco de dados são um componente essencial na vida da sociedade moderna.
- Atualmente, grande parte da população realiza diversas atividades que envolvem alguma interação com um banco de dados.
- Essas interações são exemplos do que podemos chamar de aplicações de banco de dados tradicionais. Porém, nem sempre foi assim...



Fundamentos e Conceitos Básicos

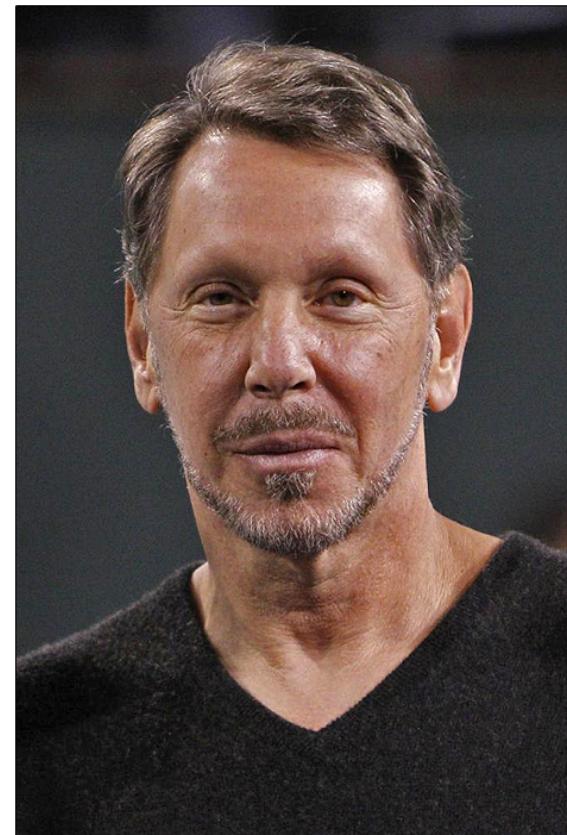
■ A Revolução Relacional...



Edgar Frank Codd



Donald Chamberlin



Larry Ellison



■ Introdução

- Os bancos de dados e sua tecnologia têm um impacto importante sobre o uso crescente dos computadores.
- Eles desempenham um papel crítico em quase todas as áreas em que os computadores são utilizados, como por exemplo: negócios, comércio eletrônico, engenharia, medicina, genética, direito, educação, jogos, etc.



■ Definição

- De maneira simplificada, um **banco de dados** representa uma coleção de dados relacionados.
- Eses **dados relacionados** dizem respeito a fatos conhecidos que podem ser registrados e possuem algum significado.
- Como exemplo de dados, temos: nome, número do telefone, endereço, data de nascimento, e-mail, etc.



■ Definição

- Um banco de dados é projetado, construído e **populado** com dados para uma finalidade específica.
- Geralmente, ele armazena informações que facilitam a **tomada de decisão**.
- Logo, é necessário que essas informações estejam dentro de um **contexto**.



Fundamentos e Conceitos Básicos

■ Exemplo de aplicação

Business Computer Lab Satisfaction Survey

This survey is designed to obtain student feedback regarding the services provided by the Business Computer Lab and identify areas in which we need to improve. Please answer each question as accurately as possible.

Using the Lab

What is your academic classification?

Freshman Sophomore Junior Senior Graduate Student Other

Do you own a computer?

Yes No

How often do you use the Business Computer Lab?

Few or more times / week Three or four times / week Once or twice / week Once / month or less

What do you **primarily** use the Business Computer Lab for?

You may check more than one.

Internet (i.e. Web Browsing, Chat) Presentations (i.e. MS PowerPoint)

Email Access Academic Programs (i.e., Minibase, Cobol, MS Visio, etc.)

Word Processing (i.e., MS Word) Games

Spreadsheets (i.e., MS Excel) Other

What do you like **MOST** about the Business Computer Lab?

You may check more than one.

Email Up-to-date software

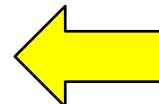
24-hour schedule Laser printers

Availability of computers Scanners

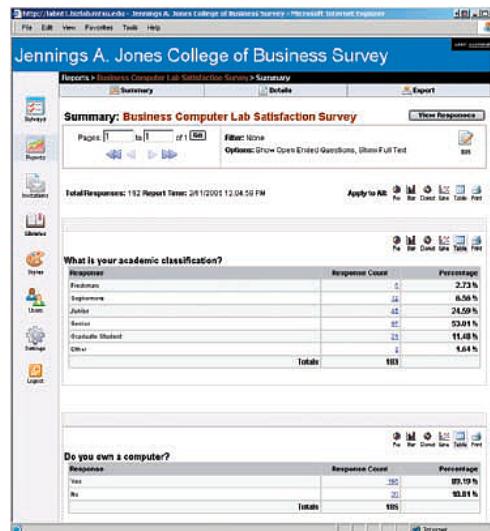
Done

Internet

SurveyData.xls



Tela Inicial de Pesquisa



Resumos



Gráficos



■ Exemplo de aplicação: Amazon.com

- Um exemplo de um grande banco de dados comercial é a **Amazon.com**.
- Possui mais de 20 milhões de livros, CDs, vídeos, DVDs, jogos eletrônicos, roupas e outros itens.
- Ocupa mais de 42 terabytes de dados.
- Está armazenado em mais de 200 computadores diferentes, ou **servidores**.
- Possui mais de 60 milhões de clientes ativos.
- Diariamente, mais de 15 milhões de visitantes acessam o site e utilizam o banco de dados para realizar compras.



Fundamentos e Conceitos Básicos

- Como lidar com todas essas informações?



■ Sistema de Gerenciamento de Banco de Dados

- Para lidar com as informações armazenadas em um banco de dados utilizamos um **Sistema de Gerenciamento de Banco de Dados**, ou **SGBD**.
- O SGBD é uma coleção de programas que permite aos usuários criar e gerenciar a estrutura de um banco de dados.
- A utilização de um SGBD facilita o processo de definição, construção, manipulação e compartilhamento de bancos de dados entre diversos usuários e aplicações.
- Exemplos de SGBD: **Microsoft SQL Server**, Oracle, MySQL, **MongoDB**, Cassandra, Neo4j, ArangoDB, etc.



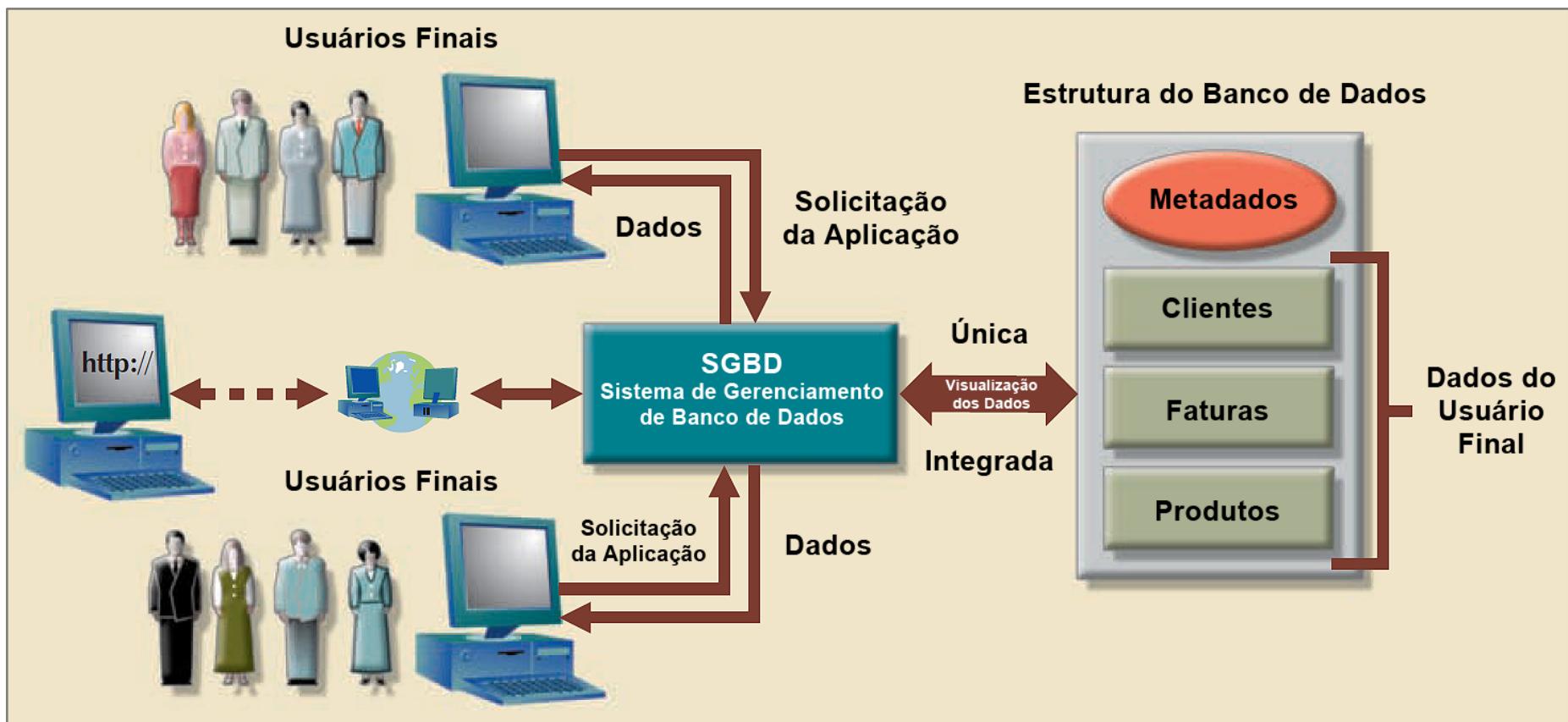
■ Sistema de Gerenciamento de Banco de Dados

- A **definição** de um banco de dados envolve a especificação dos tipos, estruturas e restrições dos dados a serem armazenados. O SGBD armazena essas definições dentro de um catálogo conhecido como **metadados**.
- A **construção** do banco de dados representa o processo de armazenar os dados em algum meio controlado pelo SGBD.
- A **manipulação** de um banco de dados inclui a realização de consultas para recuperar informações específicas, gerar relatórios ou atualizar dados já cadastrados.
- O **compartilhamento** de um banco de dados permite o acesso simultâneo de diversos usuários e programas.



Fundamentos e Conceitos Básicos

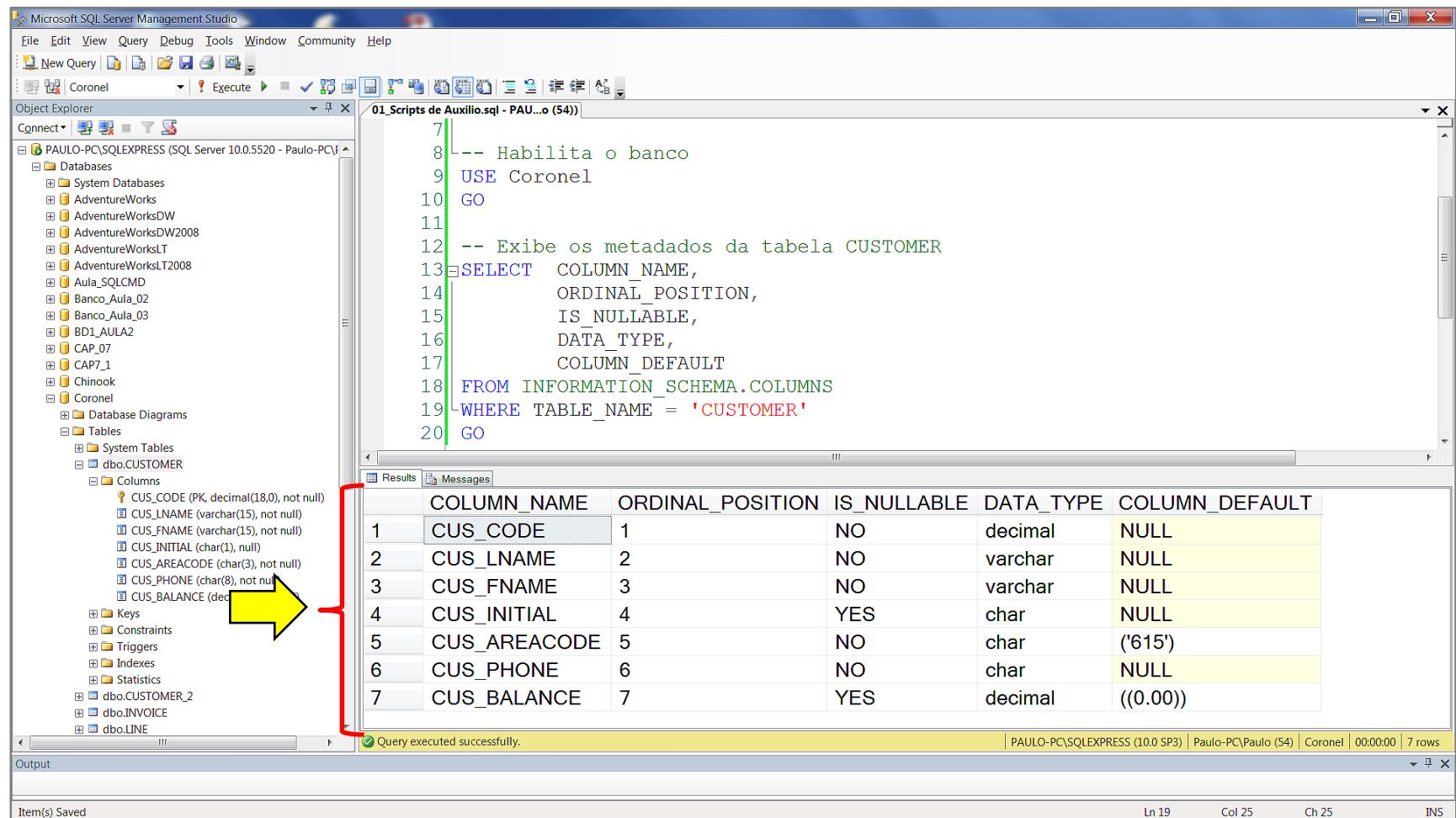
■ Sistema de Gerenciamento de Banco de Dados



O SGBD gerencia a interação entre o usuário final e o banco de dados.



■ Exemplo de metadados: Microsoft SQL Server



The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer pane displays the database structure, including the 'Coronel' database and its tables. A red arrow points from the 'Columns' node of the 'CUSTOMER' table in the Object Explorer to the results grid. The central pane contains a script named '01_Scripts de Auxilio.sql' with the following T-SQL code:

```
7
8 -- Habilita o banco
9 USE Coronel
10 GO
11
12 -- Exibe os metadados da tabela CUSTOMER
13 SELECT COLUMN_NAME,
14     ORDINAL_POSITION,
15     IS_NULLABLE,
16     DATA_TYPE,
17     COLUMN_DEFAULT
18 FROM INFORMATION_SCHEMA.COLUMNS
19 WHERE TABLE_NAME = 'CUSTOMER'
20 GO
```

The results grid shows the metadata for the 'CUSTOMER' table columns:

	COLUMN_NAME	ORDINAL_POSITION	IS_NULLABLE	DATA_TYPE	COLUMN_DEFAULT
1	CUS_CODE	1	NO	decimal	NULL
2	CUS_LNAME	2	NO	varchar	NULL
3	CUS_FNAME	3	NO	varchar	NULL
4	CUS_INITIAL	4	YES	char	NULL
5	CUS_AREACODE	5	NO	char	('615')
6	CUS_PHONE	6	NO	char	NULL
7	CUS_BALANCE	7	YES	decimal	((0.00))

At the bottom of the results pane, a message indicates: 'Query executed successfully.'

■ **Vantagens de utilizar um SGBD**

- Persistência das informações;
- Aprimoramento do compartilhamento de dados;
- Aprimoramento da segurança e do controle de acesso;
- Minimização das inconsistências e redundâncias;
- Gerenciamento da integridade dos dados;
- Processamento e otimização de consultas;
- Facilidade para backup e restauração.



- **Desvantagens de utilizar um SGBD**
 - Aumento de custos: hardware, software e treinamento de pessoal;
 - Complexidade do gerenciamento;
 - Manutenção do banco de dados;
 - Dependência do fornecedor;
 - Atualização ou substituição dos softwares utilizados pela empresa.



■ Tipos de bancos de dados

- O SGBD pode dar suporte a muitos tipos de banco de dados, os quais podem ser classificados de acordo com:
 1. Número de usuários;
 2. Localização dos dados;
 3. Tipo e extensão do uso esperado.



■ Número de usuários

- **Monousuário**: oferece suporte a apenas um usuário por vez.
- **Desktop**: banco de dados monousuário, executado em um computador pessoal.
- **Multiusuário**: oferece suporte a vários usuários simultaneamente.
- **Grupo de trabalho**: oferece suporte para um número relativamente pequeno de usuários.
- **Empresarial**: utilizado por uma organização inteira, oferece suporte para muitos usuários.



■ Localização dos dados

- **Centralizado**: possui os dados localizados em um único local.
- **Distribuído**: possui os dados distribuídos por vários locais diferentes.



■ **Tipo e extensão do uso esperado**

- **Operacional**: oferece suporte às operações diárias de uma empresa. Também é denominado de banco de dados **transacional** ou de **produção**.
- **Data Warehouse**: armazena dados utilizados para gerar informações necessárias à tomada de decisões táticas e estratégicas.
- **Semiestruturado**: oferece suporte ao armazenamento e gerenciamento de dados semiestruturados em formato XML ou similar.



■ Dados semiestruturados

- Os **dados semiestruturados** não têm um esquema definido.
- Ao invés de serem organizados em formato de tabela, geralmente são organizados por meio de etiquetas ou *tags*, que permitem agrupá-los e criar hierarquias.
- São também conhecidos como **dados não-relacionais** ou **NoSQL**.



■ Exemplos de dados semiestruturados

- **E-mails**: os metadados nativos permitem a classificação e pesquisa de palavras-chave na linguagem de marcação;
- **XML**: sua estrutura flexível, baseada em etiquetas, permite o armazenamento e transporte de dados na web.
- **JSON**: formato utilizado na transmissão de dados entre aplicações web e servidores.



■ Exemplos de dados XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Edited by XMLSpyB. -->
<?xml-stylesheet type="text/css" href="estilos.css"?>
<Clientes>
    <Entrada>
        <Nome>Paulo Giovani</Nome>
        <Endereco>
            <Tipo>Rua</Tipo>
            <Descricao>Monsenhor José Vita</Descricao>
            <Numero>280</Numero>
            <Bairro>Vila Abernéssia</Bairro>
            <Cidade>Campos do Jordão</Cidade>
            <Estado>São Paulo</Estado>
        </Endereco>
        <Email>paulogiovani@ifsp.edu.br</Email>
    </Entrada>

    <Entrada>
        <Nome>Renata Aparecida</Nome>
        <Endereco>
            <Tipo>Rua</Tipo>
            <Descricao>Débora</Descricao>
            <Numero>11</Numero>
            <Bairro>Floresta Negra</Bairro>
            <Cidade>Campos do Jordão</Cidade>
            <Estado>São Paulo</Estado>
        </Endereco>
        <Email>renata@gmail.com.br</Email>
    </Entrada>
</Clientes>
```

Paulo Giovani

Rua Monsenhor José Vita 280
Vila Abernéssia
Campos do Jordão
São Paulo
paulogiovani@ifsp.edu.br

Renata Aparecida

Rua Débora 11
Floresta Negra
Campos do Jordão
São Paulo
renata@gmail.com.br



■ Exemplos de dados JSON

```
{  
    "Clientes": {  
        "Entrada": [  
            {  
                "Nome": "Paulo Giovani",  
                "Endereco": {  
                    "Tipo": "Rua",  
                    "Descricao": "Monsenhor José Vita",  
                    "Numero": 280,  
                    "Bairro": "Vila Abernéssia",  
                    "Cidade": "Campos do Jordão",  
                    "Estado": "São Paulo"  
                },  
                "Email": "paulogiovani@ifsp.edu.br"  
            },  
            {  
                "Nome": "Renata Aparecida",  
                "Endereco": {  
                    "Tipo": "Rua",  
                    "Descricao": "Débora",  
                    "Numero": 11,  
                    "Bairro": "Floresta Negra",  
                    "Cidade": "Campos do Jordão",  
                    "Estado": "São Paulo"  
                },  
                "Email": "renata@gmail.com.br"  
            }  
        ]  
    }  
}
```

```
{  
    "Clientes": {  
        "Entrada": [  
            {  
                "Nome": "Paulo Giovani",  
                "Endereco": {  
                    "Tipo": "Rua",  
                    "Descricao": "Monsenhor José Vita",  
                    "Numero": 280,  
                    "Bairro": "Vila Abernéssia",  
                    "Cidade": "Campos do Jordão",  
                    "Estado": "São Paulo"  
                },  
                "Email": "paulogiovani@ifsp.edu.br"  
            },  
            {  
                "Nome": "Renata Aparecida",  
                "Endereco": {  
                    "Tipo": "Rua",  
                    "Descricao": "Débora",  
                    "Numero": 11,  
                    "Bairro": "Floresta Negra",  
                    "Cidade": "Campos do Jordão",  
                    "Estado": "São Paulo"  
                },  
                "Email": "renata@gmail.com.br"  
            }  
        ]  
    }  
}
```

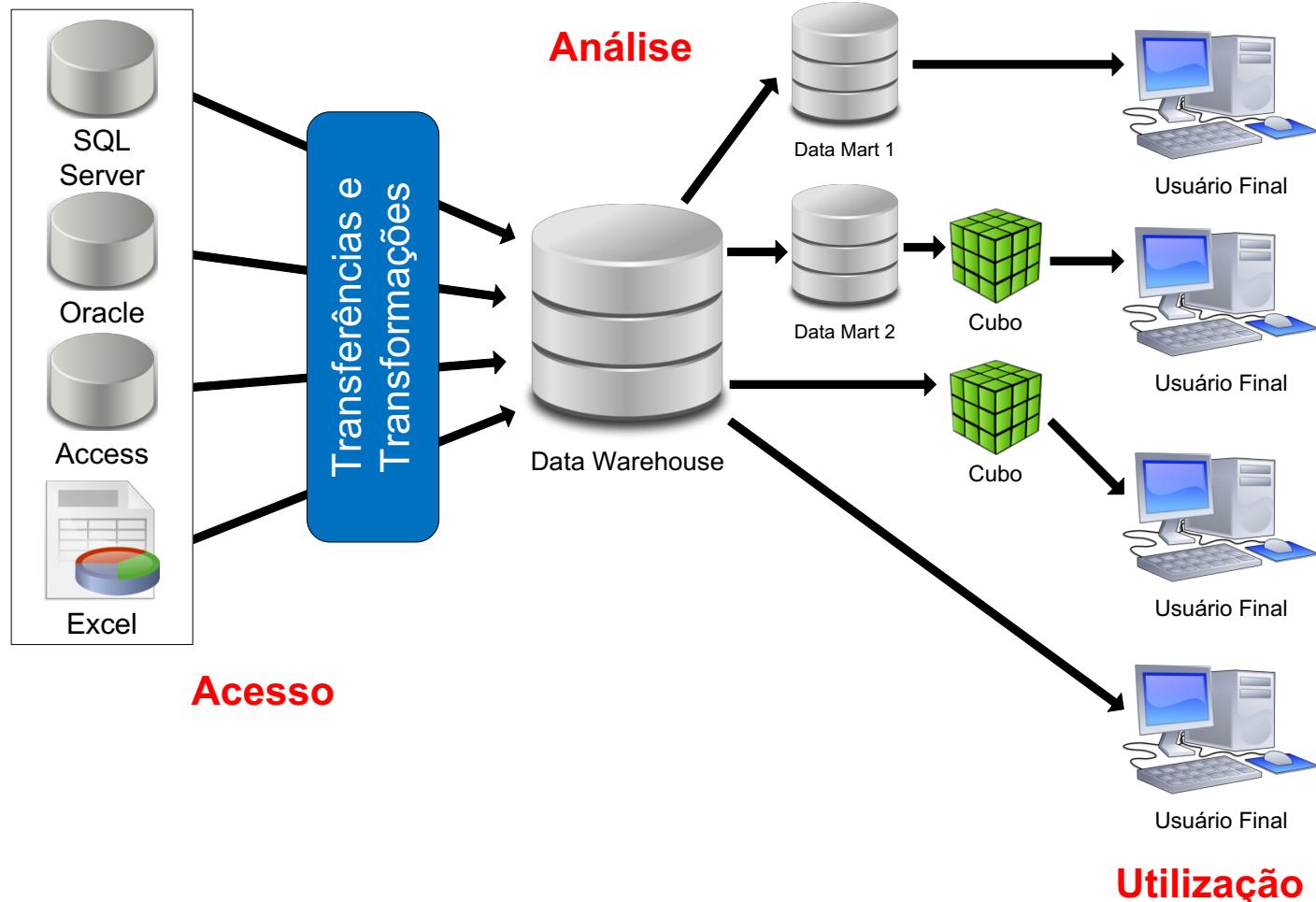


■ Data Warehouse

- O **Data Warehouse** é um sistema de computação utilizado para armazenar informações relativas às atividades de uma organização, dentro de um banco de dados, de forma consolidada.
- Ele possibilita a análise de um grande volume de dados, coletados dos sistemas transacionais, oferecendo suporte para a tomada de decisões no presente, além da previsão de eventos futuros.

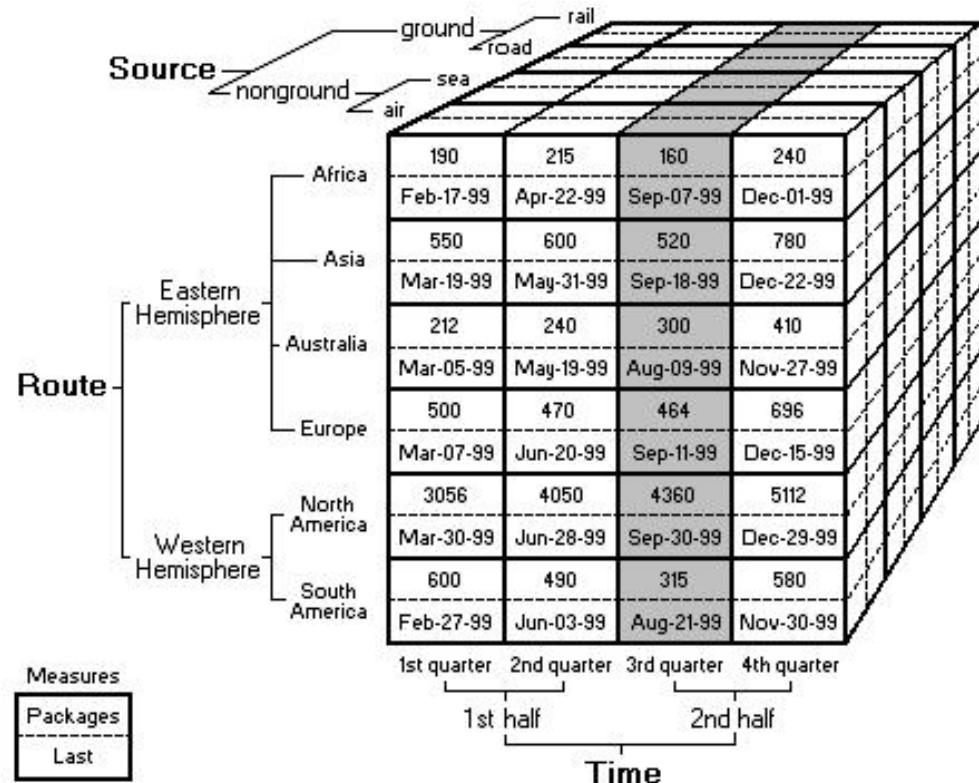
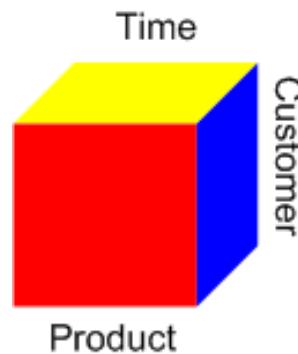


■ Data Warehouse



Fundamentos e Conceitos Básicos

■ Cubos OLAP



Extração de Dados - OLTP

Transformação e Padronização

Importação Modelo OLAP

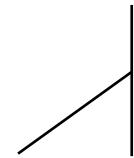
Construção dos Cubos

Produção de Relatórios



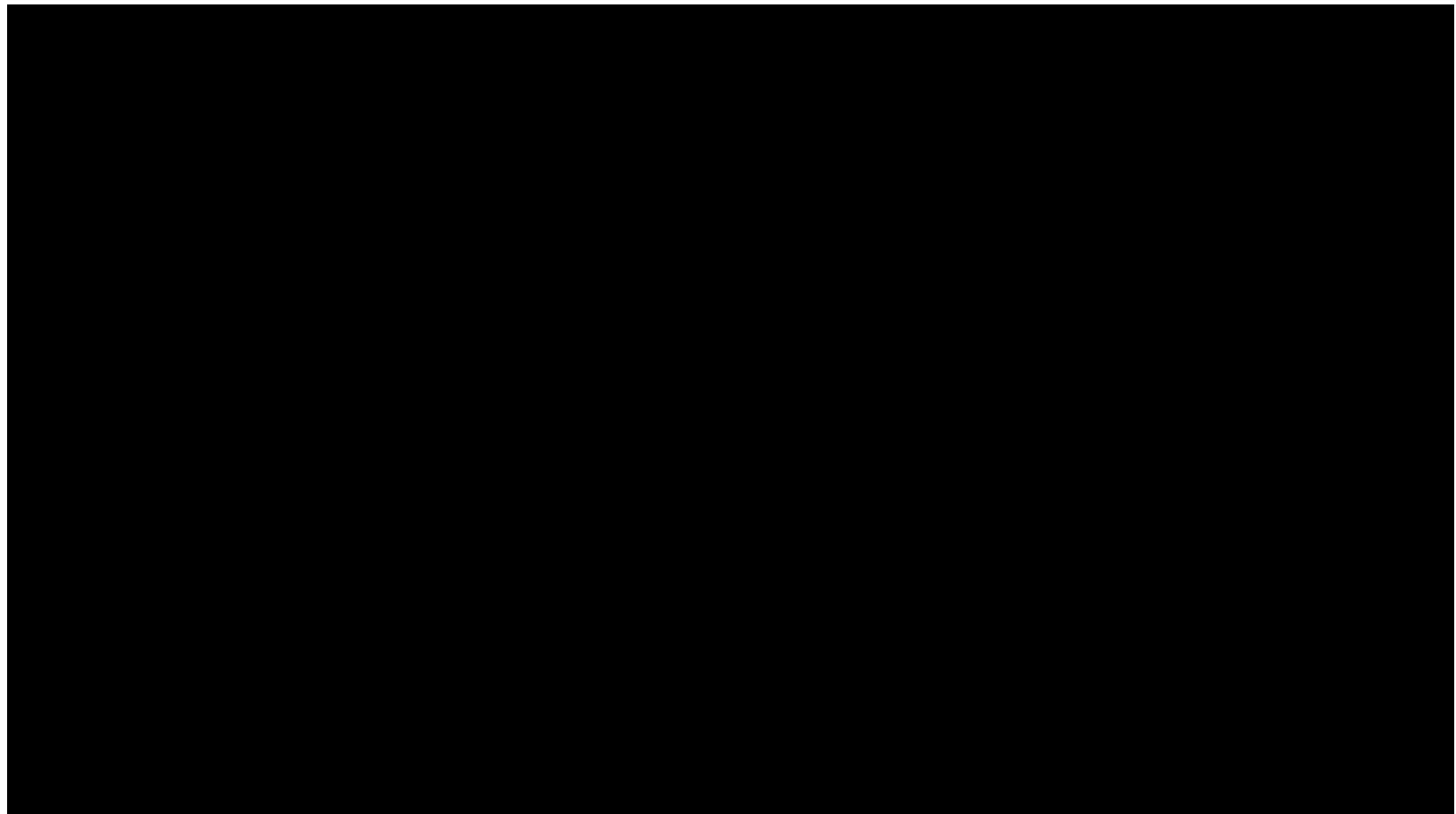
Fundamentos e Conceitos Básicos

- Importância do projeto de banco de dados
 - O projeto de banco de dados refere-se às atividades que focam na elaboração da estrutura que será utilizada para armazenar e gerenciar dados do usuário final.



Tá, e qual a
importância disso?

- **História dos sistemas de banco de dados**



<https://youtu.be/KG-mqHoXOXY>



- Evolução das técnicas de modelagem

Época Atual						
1990						
1980						
1970						
1960						
1950						
Antes de 1950						
	Sistemas de Arquivos	Hierárquico	Rede	Relacional	Objeto	Objeto Relacional e Novas Tecnologias

■ Sistemas de arquivos

- As bases de dados evoluíram a partir de **sistemas de arquivos**. Uma compreensão de suas características relativamente simples torna mais fácil entender a complexidade de um projeto de bancos de dados.
- Em um sistema de arquivos, os dados são armazenados em arquivos independentes, onde cada um necessita de seus próprios programas de gerenciamento de dados.



■ Sistemas de arquivos

- Esse sistema procurava simular os processos existentes em um ambiente composto por pastas e ficheiros utilizadas, para armazenar os dados;
- A organização dos dados era determinada pela sua utilização;
- Cada arquivo era propriedade de um indivíduo ou departamento;
- Cada arquivo utilizava seu próprio aplicativo para armazenar, recuperar e modificar dados, o que exigia uma programação extensiva;
- A alteração das estruturas existentes era mais difícil;
- Os recursos de segurança eram inadequados;
- A administração do sistema poderia se tornar muito complexa;
- A conversão do processo manual para o processo informatizado exigia a presença de um **especialista em processamento de dados**.



- **Sistemas de arquivos**



Exemplo de armários com ficheiros de arquivos.

- **Sistemas de arquivos**



Ken Thompson e Dennis Ritchie, criadores da linguagem C.

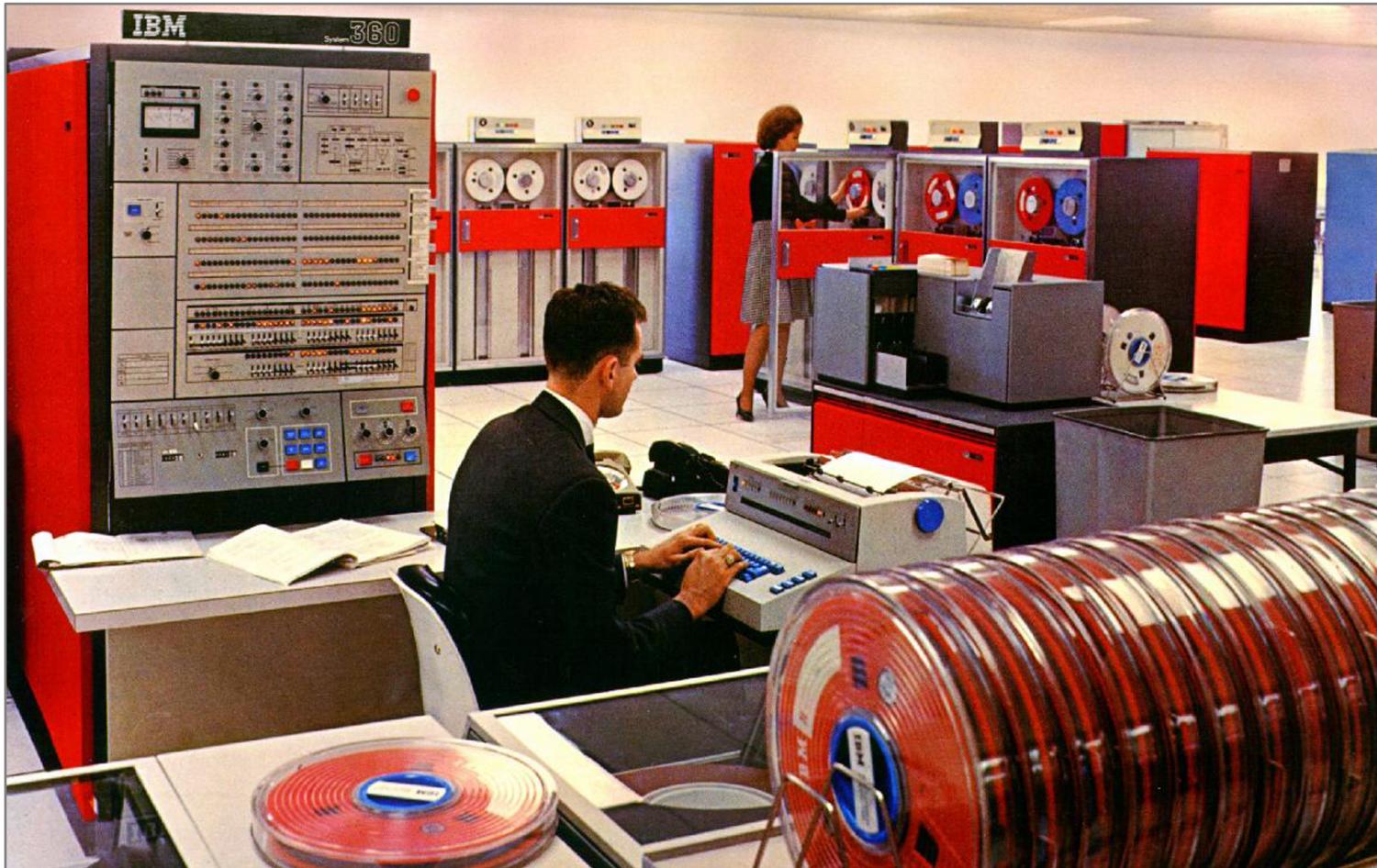


- **Sistemas de arquivos**



Exemplo de departamento de processamento de dados.

- **Sistemas de arquivos**



Exemplo de departamento de processamento de dados.



■ Sistemas de arquivos

- O método de sistema de arquivos foi um aprimoramento em relação ao sistema manual.
- Ele foi utilizado por mais de duas décadas, sendo que a compreensão de suas falhas auxilia no desenvolvimento dos bancos de dados mais atuais e modernos.
- Um sistema de arquivos apresenta **dependência estrutural**, o que significa que o acesso a um arquivo é dependente de sua estrutura.

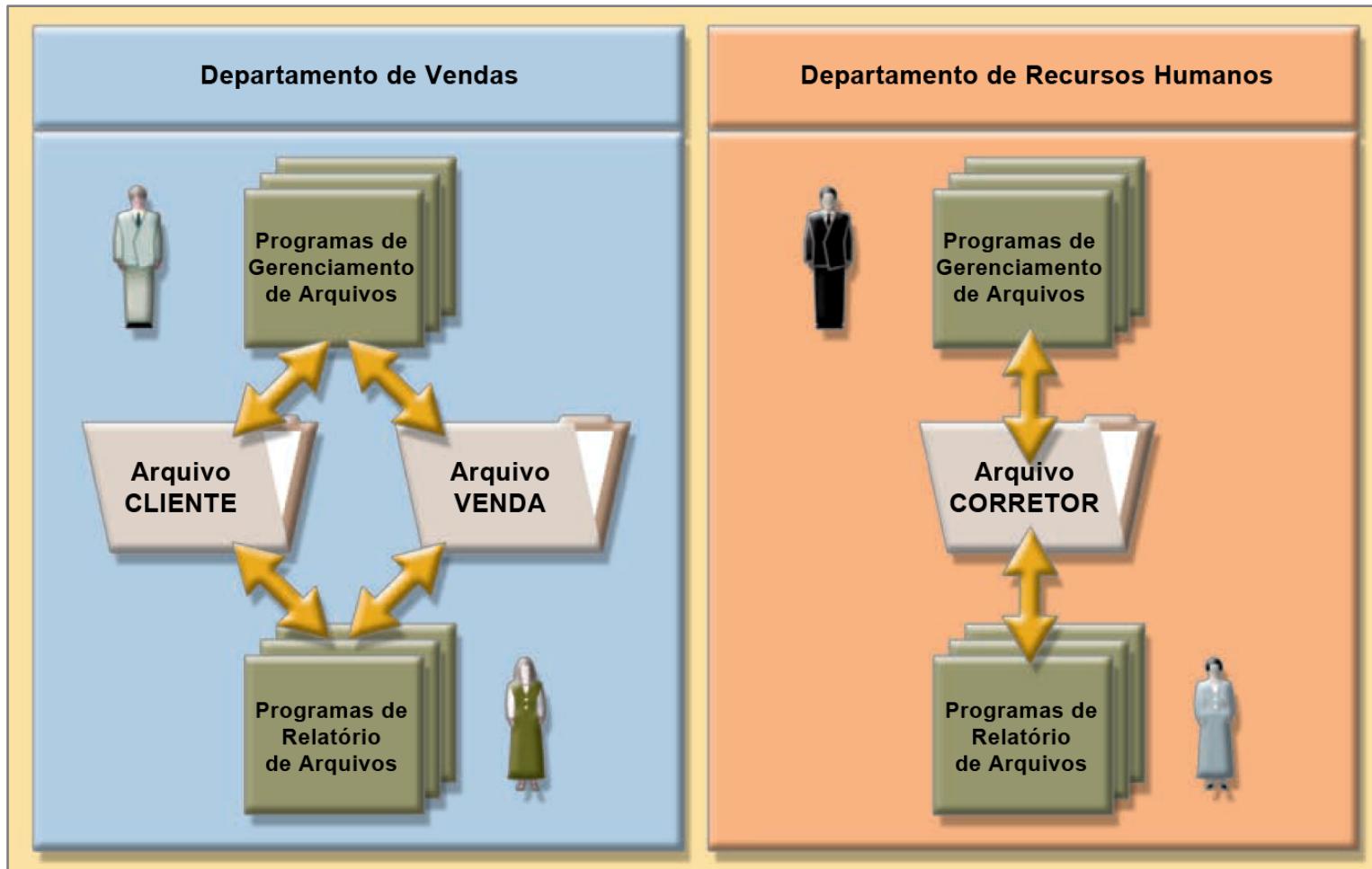


■ Sistemas de arquivos

- Todos os programas do sistema de arquivos devem ser modificados em conformidade com essa estrutura, além de possuir os seguintes elementos:
 1. Linhas que especifiquem a abertura de um tipo específico de arquivo;
 2. A especificação de cada registro;
 3. As definições para cada campo utilizado pelo programa.



■ Sistemas de arquivos



■ Sistemas de arquivos

--- MENU DE OPCOES ---

1. Inserir um cadastro;
2. Remover um cadastro;
3. Listar cadastros;
4. Buscar cadastro;
5. Salvar arquivo;
6. Carregar arquivo;
7. Sair.

Informe sua escolha: 1

Nome: Renata Aparecida
Tipo: Rua
Descricao: Debora
Numero: 11
Bairro: Floresta Negra
Cidade: Campos do Jordao
Estado: Sao Paulo
Email: renata@gmail.com

--- MENU DE OPCOES ---

1. Inserir um cadastro;
2. Remover um cadastro;
3. Listar cadastros;
4. Buscar cadastro;
5. Salvar arquivo;
6. Carregar arquivo;
7. Sair.

Informe sua escolha: 3

Exibindo dados..

Renata Aparecida
Rua Debora, 11
Floresta Negra - Campos do Jordao / Sao Paulo
renata@gmail.com



■ Sistemas de arquivos

```
// Exemplo de armazenamento de informações utilizando C
// Autor: Paulo Giovani
// Data: 27/12/2014

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

// Estrutura para armazenamento dos dados
struct clientes
{
    char nome[50];
    char tipo[20];
    char descricao[100];
    char numero[10];
    char bairro[50];
    char cidade[50];
    char estado[30];
    char email[100];

    struct clientes *next;    // Ponteiro para a próxima entrada
    struct clientes *prior;   // Ponteiro para a entrada anterior
};

struct clientes *start;      // Ponteiro para a primeira entrada na lista
struct clientes *last;       // Ponteiro para a última entrada na lista
struct clientes *find(char *);
```



■ Sistemas de arquivos

```
// Menu para o usuário selecionar uma opção
int menu_select(void)
{
    char s[80];
    int c;

    printf ("\n--- MENU DE OPCOES ---\n\n");

    printf ("1. Inserir um cadastro;\n");
    printf ("2. Remover um cadastro;\n");
    printf ("3. Listar cadastros;\n");
    printf ("4. Buscar cadastro;\n");
    printf ("5. Salvar arquivo;\n");
    printf ("6. Carregar arquivo;\n");
    printf ("7. Sair.\n");

    do
    {
        printf ("\nInforme sua escolha: ");
        gets(s);
        c = atoi(s);

    } while(c<0 || c>7);

    return c;
}
```



■ Sistemas de arquivos

```
// Insere um cadastro
void enter(void)
{
    struct clientes *info;

    for(;;)
    {
        info = (struct clientes *)malloc(sizeof(struct clientes));

        if(!info)
        {
            printf("\nFalta de Memoria!\n");
            return;
        }

        inputs("\nNome: ", info->nome, 50);

        if(!info->nome[0])
            break; // Interrompe a entrada se o nome for deixado em branco

        inputs("Tipo: ", info->tipo, 20);
        inputs("Descricao: ", info->descricao, 100);
        inputs("Numero: ", info->numero, 10);
        inputs("Bairro: ", info->bairro, 50);
        inputs("Cidade: ", info->cidade, 50);
        inputs("Estado: ", info->estado, 30);
        inputs("Email: ", info->email, 100);
```



■ Sistemas de arquivos

```
// Remove um elemento da lista
void mldelete(struct clientes **start, struct clientes **last)
{
    struct clientes *info;
    char s[80];

    inputs("\nInforme o nome desejado: ", s, 30);

    info = find(s);

    if(info)
    {
        if(*start==info)
        {
            *start=info->next;

            if(*start)
                (*start)->prior = NULL;
            else
                *last = NULL;
        }
        else
        {
            info->prior->next = info->next;

            if(info!=*last)
                info->next->prior = info->prior;
        }
    }
}
```



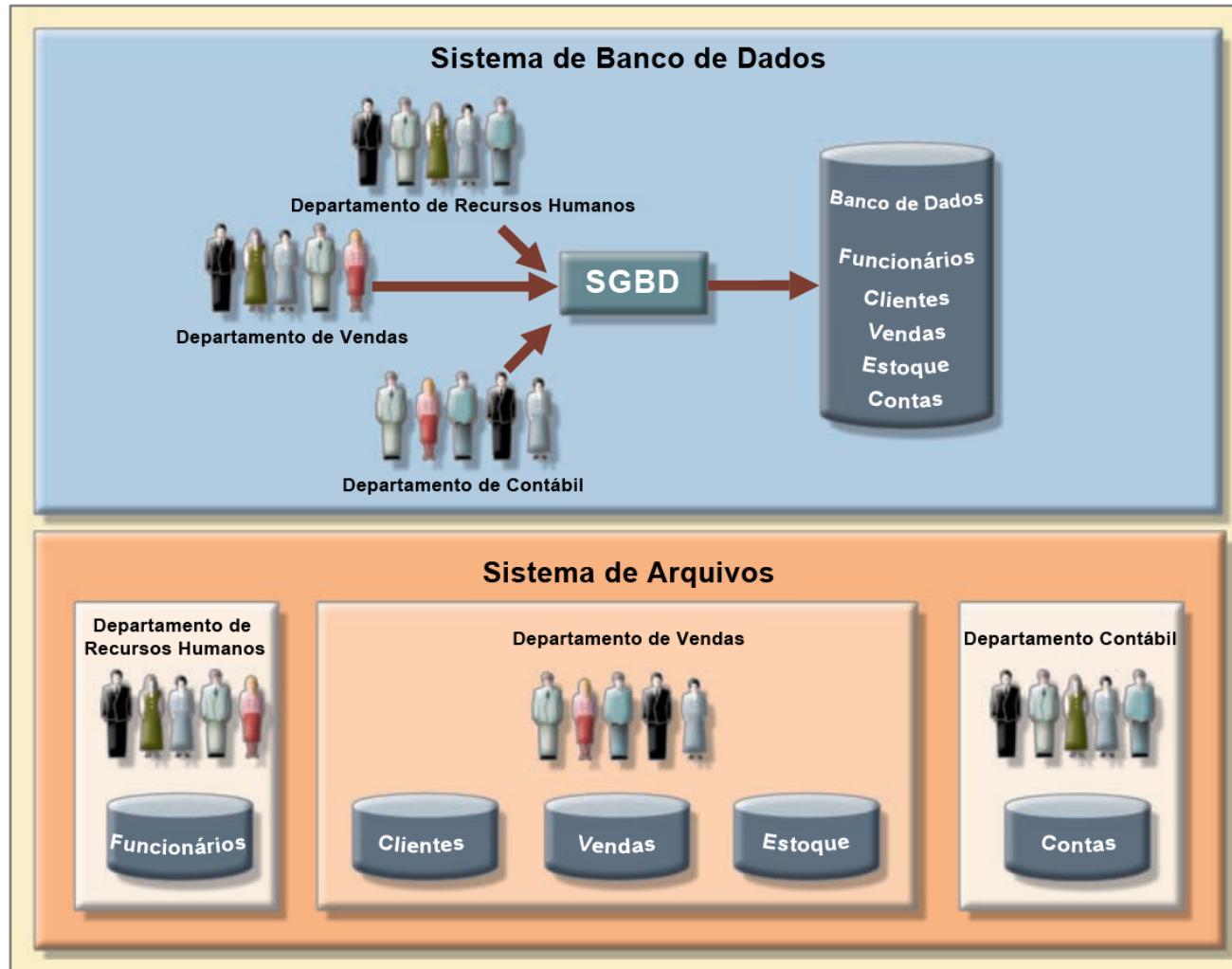
■ Sistemas de arquivos

- A estrutura do sistema de arquivos dificulta a combinação de dados a partir de várias fontes, o que pode causar diversos problemas.
- Por exemplo, o armazenamento dos mesmos dados que são utilizados em vários departamentos pode ocorrer de forma redundante, gerando **ilhas de informação**, o que pode ocasionar diversos tipos de anomalias e inconsistências.



Fundamentos e Conceitos Básicos

- Comparação entre os sistemas de armazenamento de dados**



■ Ambiente do sistema de banco de dados

- O termo **sistema de banco de dados** refere-se a uma organização de componentes que define e regula a coleta, o armazenamento, o gerenciamento e a utilização de dados em um ambiente de banco de dados.
- Esse sistema é composto por cinco partes principais: hardware, software, pessoas, procedimentos e dados.



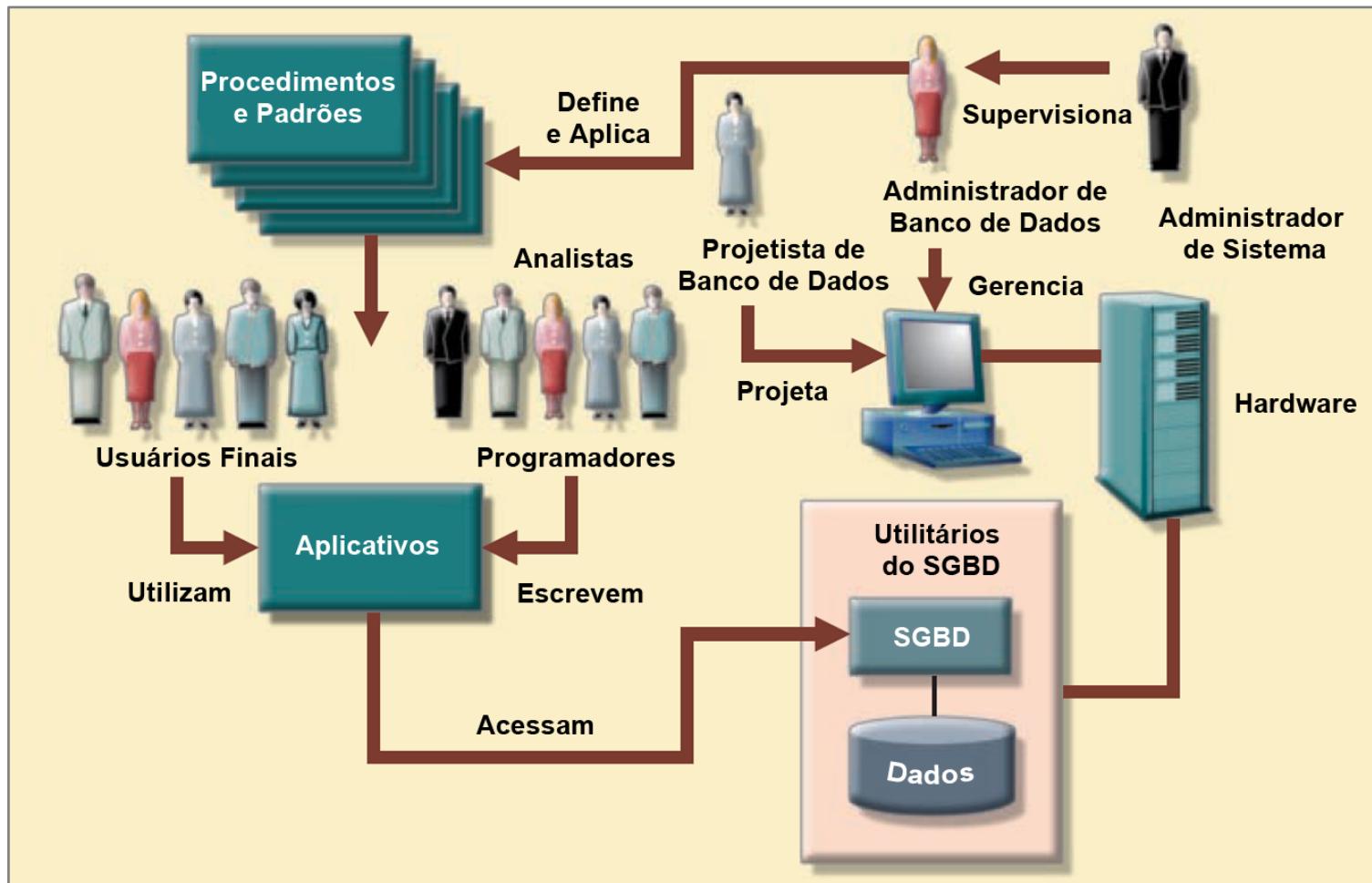
- **Ambiente do sistema de banco de dados**

- A introdução de um sistema de banco de dados em um ambiente de sistema de arquivos fornece um modelo no qual podem ser aplicados procedimentos e padrões mais rigorosos.
- O papel do componente humano muda da ênfase em programação para focar nos aspectos mais amplos de gerenciamento.



Fundamentos e Conceitos Básicos

■ Ambiente do sistema de banco de dados



■ Ambiente do sistema de banco de dados

- O SGBD fornece acesso aos dados por meio de uma linguagem não-procedural.
- Nesse tipo de linguagem, o usuário descreve a informação desejada sem fornecer um procedimento específico para a obtenção dessas informações.
- Em bancos de dados relacionais, a linguagem utilizada atualmente é denominada **Linguagem de Consulta Estruturada (Structured Query Language)**, ou simplesmente **SQL**.



- Na próxima aula veremos...
 - Modelos de Dados.

