

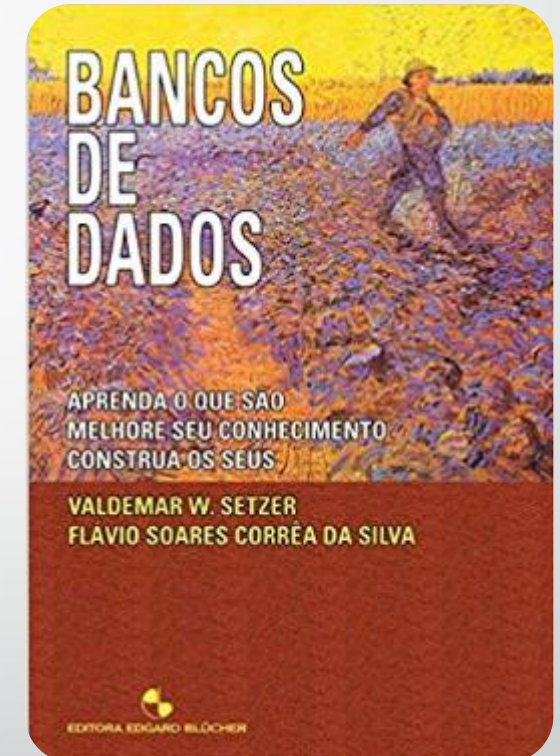
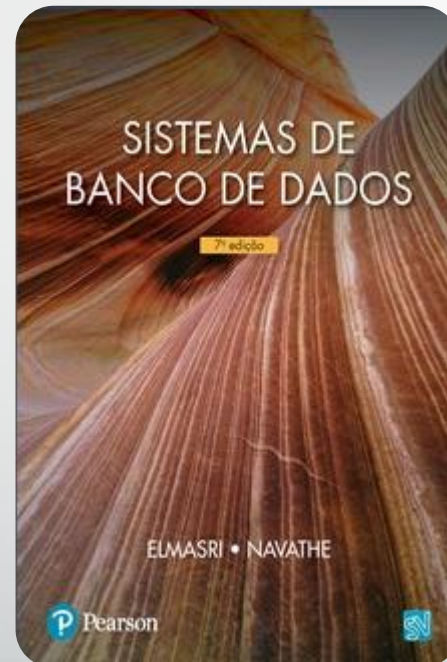
S112 – Banco de Dados

Professor MSc. Eng. Márcio José de Lemos

E-mail: marcio.lemos@senairs.org.br

<http://lattes.cnpq.br/4769158065464009>

Bibliografia Básica



- Aula de Revisão:
- Modelo Conceitual
- Modelo Lógico
- Modelo Físico



HI 0898 0050



SEM INTERNET

REQUISITO

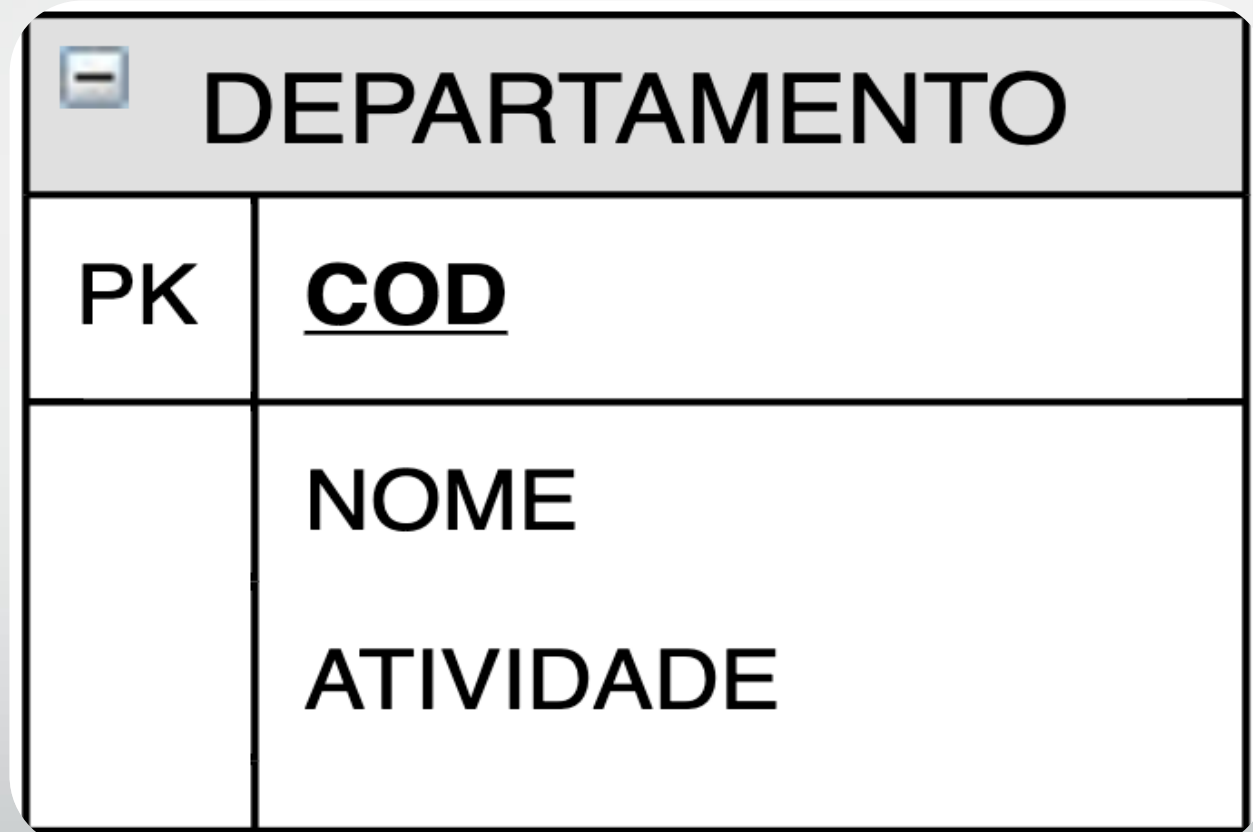
- O cliente precisa de um sistema que guarde dados de vários departamentos!!



MODELO CONCEITUAL



MODELO LÓGICO VISUAL



ABRA O XAMPP 1

XAMPP Control Panel v3.2.4 [Compiled: Jun 5th 2019]

XAMPP Control Panel v3.2.4

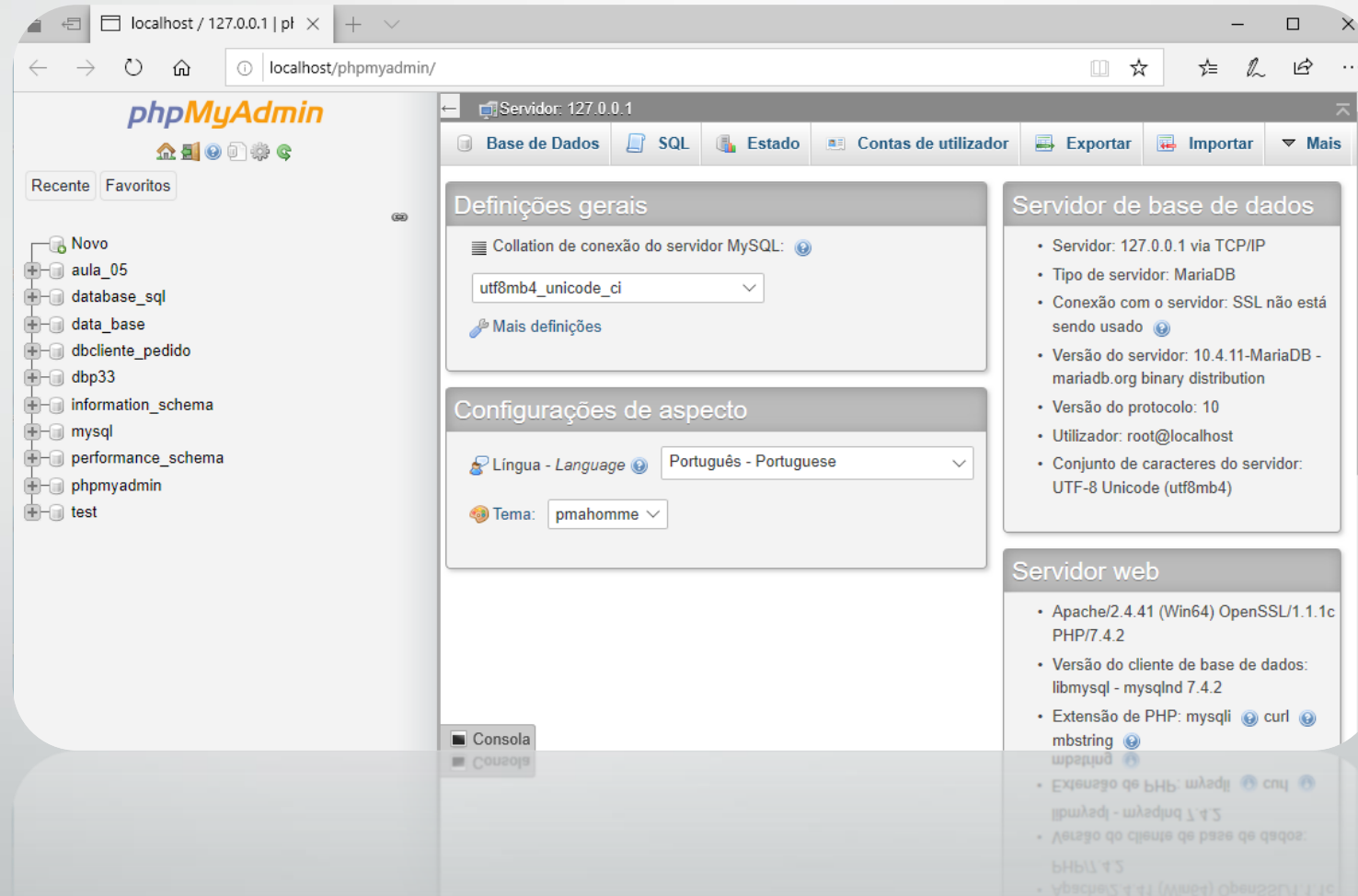
Modules

Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache	21984 31072	80, 443	Stop Admin Config Logs
<input type="checkbox"/>	MySQL	26840	3306	Stop Admin Config Logs
<input type="checkbox"/>	FileZilla			Start Admin Config Logs
<input type="checkbox"/>	Mercury			Start Admin Config Logs
<input type="checkbox"/>	Tomcat			Start Admin Config Logs

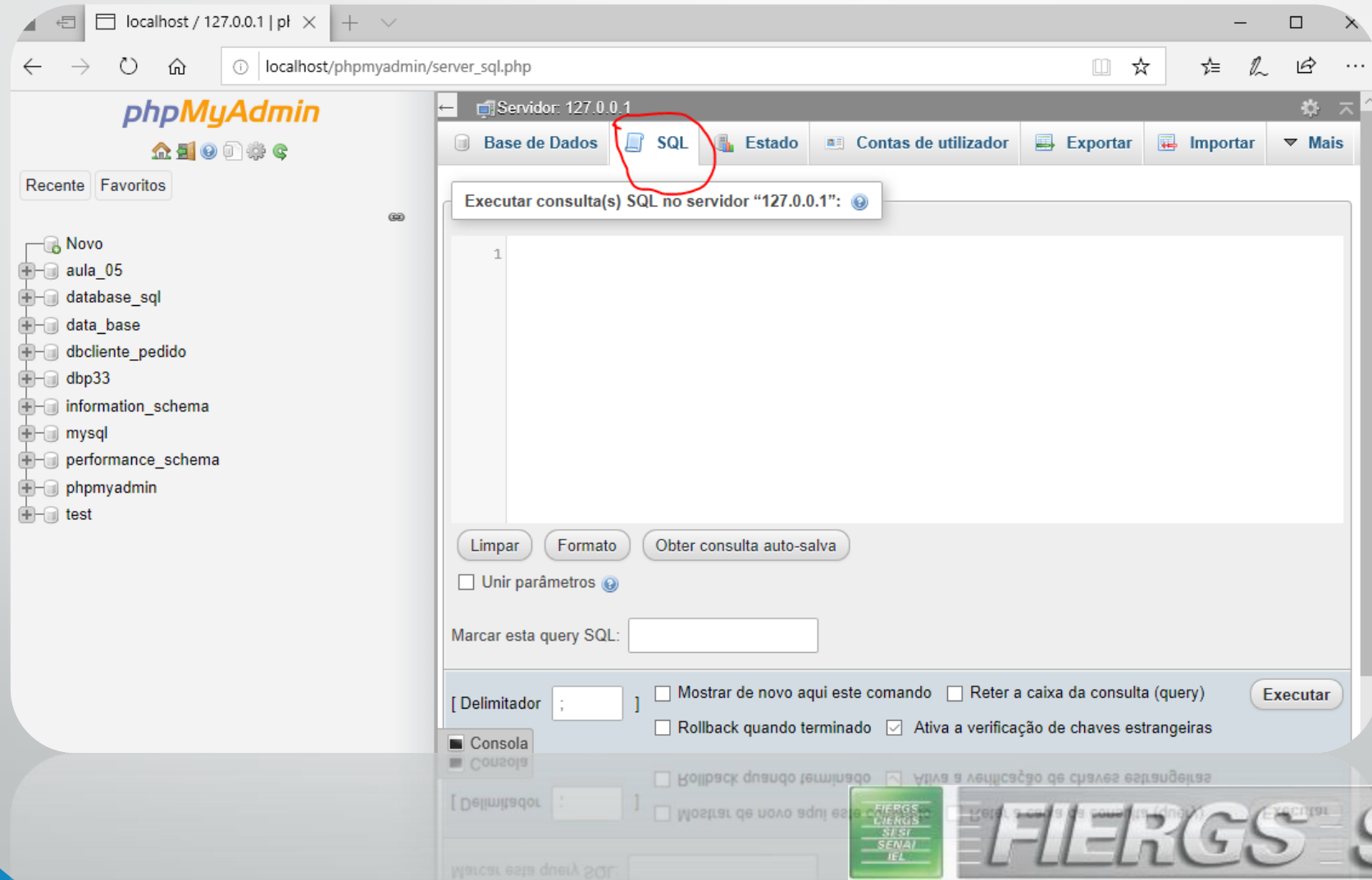
Config Netstat Shell Explorer Services Help Quit

19:12:56 [main] Initializing Control Panel
19:12:56 [main] Windows Version: Enterprise 64-bit
19:12:56 [main] XAMPP Version: 7.4.2
19:12:56 [main] Control Panel Version: 3.2.4 [Compiled: Jun 5th 2019]
19:12:56 [main] You are not running with administrator rights! This will work for most application stuff but whenever you do something with services there will be a security dialogue or things will break! So think about running this application with administrator rights!
19:12:56 [main] XAMPP Installation Directory: "c:\xampp\xampp\xampp\
19:12:56 [main] Checking for prerequisites

ABRA O XAMPP 2



ABRA O XAMPP 3



ETAPA 1 – MODELO FÍSICO

--PARTE 1 (comando SQL):

```
CREATE DATABASE topico_05;
```

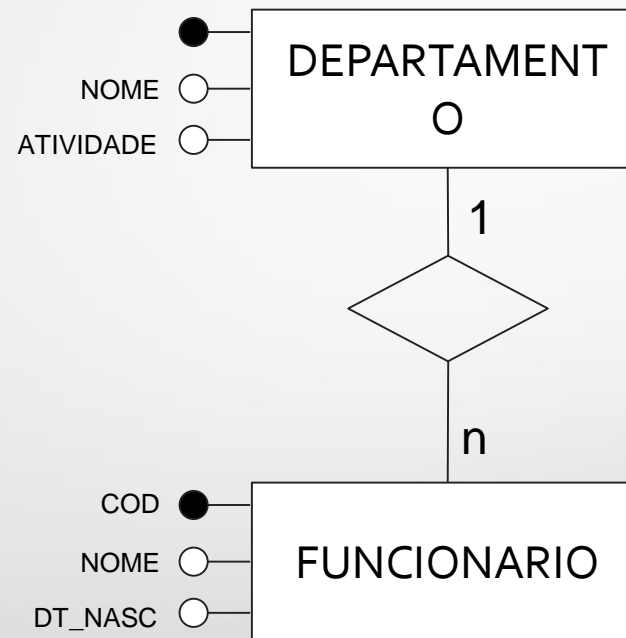
```
CREATE TABLE departamento(  
    cod int NOT NULL,  
    nome varchar(25) NOT NULL,  
    atividade varchar(250) NOT NULL);
```

```
ALTER TABLE departamento ADD PRIMARY KEY (cod);
```

ETAPA 2



ETAPA 2 - MODELO CONCEITUAL (Opção 1)



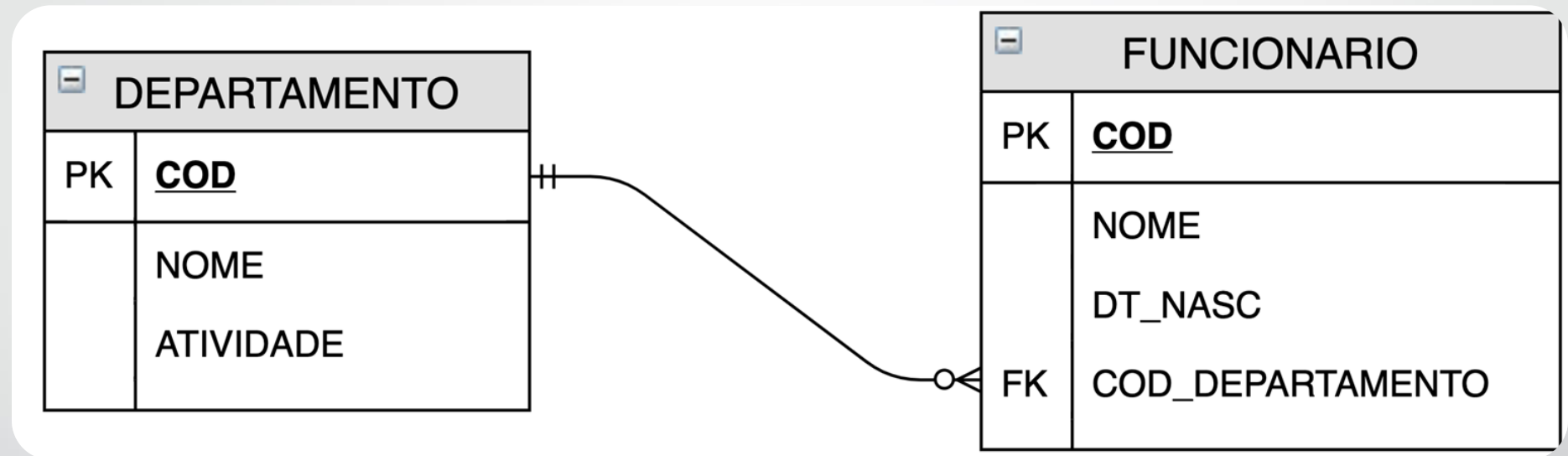


ETAPA 2 - MODELO LÓGICO TEXTUAL

DEPARTAMENTO (cod, nome, atividade)

FUNCIONARIO (cod, nome, dt_nasc, cod_departamento)

ETAPA 2 - MODELO LÓGICO VISUAL



ETAPA 2 - MODELO LÓGICO VISUAL

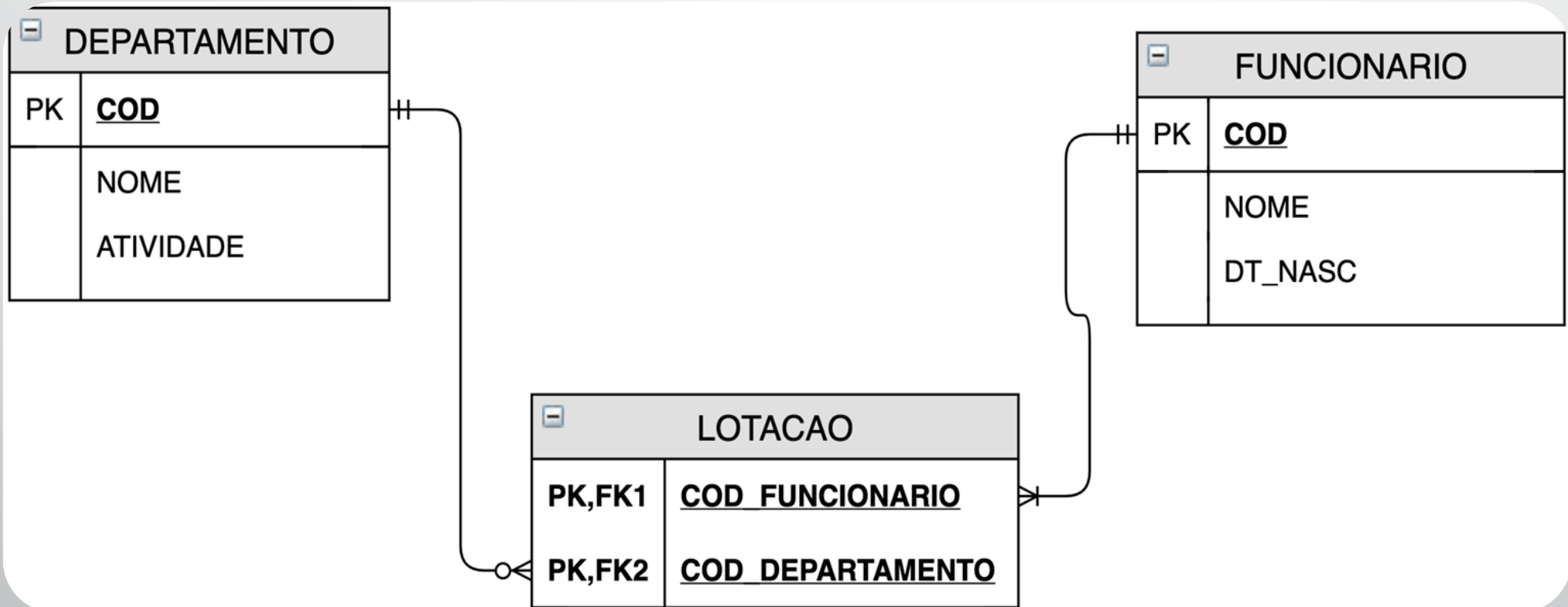
--PARTE 2

```
CREATE TABLE funcionario(  
    cod int(11) NOT NULL,  
    nome varchar(25) NOT NULL,  
    cod_departamento INT(11) NOT NULL);  
  
ALTER TABLE funcionario ADD PRIMARY KEY (cod);  
  
ALTER TABLE funcionario  
    ADD CONSTRAINT fk_funcionario_departamento  
    FOREIGN KEY (cod_departamento)  
    REFERENCES departamento (cod);
```


ETAPA 3



ETAPA 3 – MODELO LÓGICO VISUAL





ETAPA 3 – MODELO FÍSICO

--PARTE 3

```
ALTER TABLE `funcionario` DROP INDEX `fk_funcionario_departamento`;
```

```
ALTER TABLE `funcionario` DROP INDEX `cod_departamento`;
```

```
CREATE TABLE lotacao(  
    cod_funcionario int(11) NOT NULL,  
    cod_departamento int(11) NOT NULL  
);
```

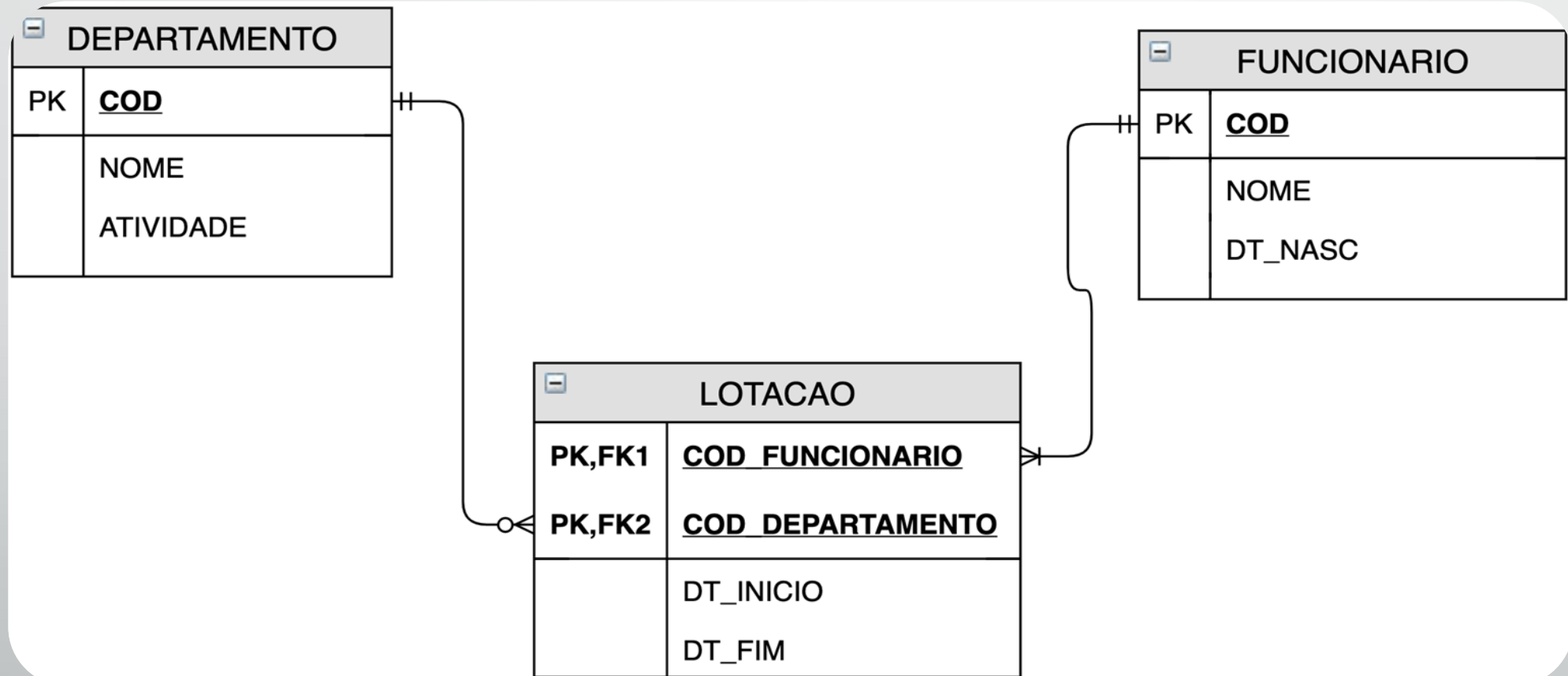
ETAPA 3 – MODELO FÍSICO

```
ALTER TABLE lotacao  
    ADD PRIMARY KEY (cod_funcionario, cod_departamento);
```

```
ALTER TABLE lotacao  
    ADD CONSTRAINT fk_lotacao_funcionario  
    FOREIGN KEY (cod_funcionario)  
    REFERENCES funcionario (cod);
```

```
ALTER TABLE lotacao  
    ADD CONSTRAINT fk_lotacao_departamento  
    FOREIGN KEY (cod_departamento)  
    REFERENCES departamento (cod);
```

ETAPA 4 – MODELO LÓGICO VISUAL



Modelagem de dados

- É a criação de uma estrutura de dados eletrônica (banco de dados) que representa um conjunto de informações.
- Esta estrutura permite ao usuário recuperar dados de forma rápida e eficiente.

Modelagem de dados

- O objetivo é incluir dados em uma estrutura que possibilite transformar os dados originais em vários tipos de saídas como formulários, relatórios, etiquetas ou gráficos.

Modelagem de dados



Modelagem de dados

- Essa capacidade de transformar informações caracteriza as operações de banco de dados e é a chave de sua utilidade.
- Um Banco de Dados – BD, representa uma coleção de dados que possui algum significado e objetiva atender a um conjunto de usuários.

Modelagem de dados

Por exemplo: um catálogo telefônico pode ser considerado um BD.

Sendo assim, um BD não necessariamente está informatizado.

Modelagem de dados

- Quando resolvemos informatizar um BD, utilizamos um programa especial para realizar essa tarefa, o SGBD – Sistema Gerenciador de Banco de Dados.
- Em um SGBD relacional, enxergamos os dados armazenados em uma estrutura chamada tabela.

Sem
Internet



Modelagem de dados

- Neste modelo, as tabelas de um BD são relacionadas, permitindo assim que possamos recuperar informações envolvendo várias delas.

Modelagem de dados

- Observe o exemplo abaixo:

CLIENTES		
Código	Nome	Data de nascimento
1	Regilan Meira Silva	13/02/1983
2	Aline Araujo Freitas	27/08/1986
3	Joaquim José Pereira da Silva	12/05/1967
4	Maria Aparecida Gomes da Costa	06/01/1995

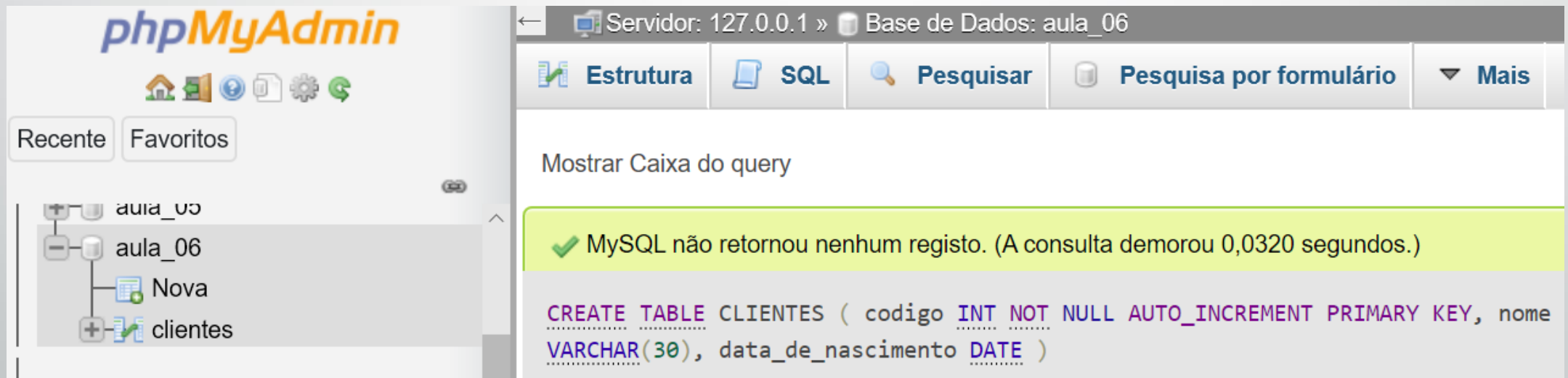
Modelagem de dados

- Observe o exemplo abaixo:

```
CREATE TABLE CLIENTES (  
    codigo INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(30),  
    data_de_nascimento DATE  
);
```

Modelagem de dados

- Observe o exemplo abaixo:



The screenshot displays the phpMyAdmin web interface. On the left, the database structure tree shows 'aula_06' expanded, with a new table 'clientes' being created. The main panel shows the 'SQL' tab selected, with a message indicating that the query was successful: 'MySQL não retornou nenhum registro. (A consulta demorou 0,0320 segundos.)'. Below the message, the SQL query is displayed: `CREATE TABLE CLIENTES (codigo INT NOT NULL AUTO_INCREMENT PRIMARY KEY, nome VARCHAR(30), data_de_nascimento DATE)`.

phpMyAdmin

Recente Favoritos

Servidor: 127.0.0.1 » Base de Dados: aula_06

Estrutura SQL Pesquisar Pesquisa por formulário Mais

Mostrar Caixa do query

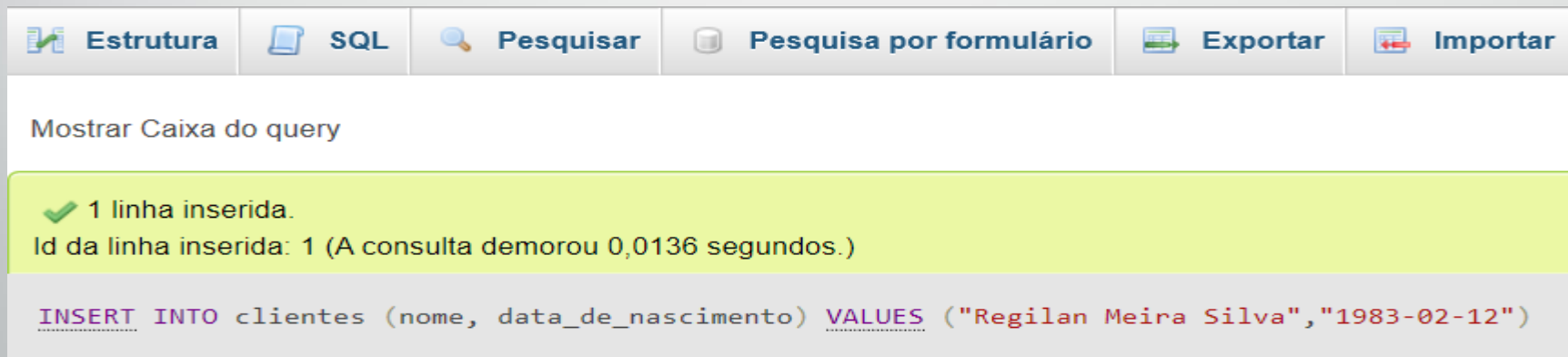
✓ MySQL não retornou nenhum registro. (A consulta demorou 0,0320 segundos.)

```
CREATE TABLE CLIENTES ( codigo INT NOT NULL AUTO_INCREMENT PRIMARY KEY, nome VARCHAR(30), data_de_nascimento DATE )
```

Modelagem de dados

- Observe o exemplo abaixo:

```
INSERT INTO clientes (nome, data_de_nascimento)  
VALUES ("Regilan Meira Silva", "1983-02-12");
```



The screenshot shows a database management interface with a menu bar at the top containing icons and labels for 'Estrutura', 'SQL', 'Pesquisar', 'Pesquisa por formulário', 'Exportar', and 'Importar'. Below the menu bar, there is a section titled 'Mostrar Caixa do query'. A green status bar displays a checkmark icon and the text '1 linha inserida. Id da linha inserida: 1 (A consulta demorou 0,0136 segundos.)'. At the bottom, the executed SQL query is shown: `INSERT INTO clientes (nome, data_de_nascimento) VALUES ("Regilan Meira Silva", "1983-02-12")`.

Modelagem de dados




- Observe o exemplo abaixo:

✓ A mostrar registos de 0 - 0 (1 total, A consulta demorou 0,0017 segundos.)

```
SELECT * FROM clientes
```

☐ Mostrar tudo | Número de registos: 25 ▾ Filtrar registos:

+ Opções

	codigo	nome	data_de_nascimento
<input type="checkbox"/>  Editar  Copiar  Apagar	1	Regilan Meira Silva	1983-02-12

Modelagem de dados

- Observe o exemplo abaixo:

```
CREATE TABLE telefones(  
    codigo int NOT NULL,  
    numero VARCHAR(15),  
    tipo VARCHAR(15)  
);
```



CONHEÇA O PRIMEIRO COMPUTADOR DO MUNDO

WEBTUDO CURIOSIDADES

Modelagem de dados

- Observe o exemplo abaixo:

```
ALTER TABLE telefones  
    ADD CONSTRAINT fk_Telefone_clientes  
    FOREIGN KEY (codigo)  
    REFERENCES clientes(codigo);
```

Modelagem de dados

- Podemos verificar que a tabela CLIENTES está relacionada com a tabela Telefones.
- Note que o cliente Regilan Meira Silva possui dois telefones: um celular e um residencial.

Modelagem de dados

- A cliente Aline Araujo Freitas possui um telefone celular, Maria Aparecida Gomes da Costa possui um celular e Joaquim José Pereira da Silva não possui telefone.

Modelagem de dados

- Tal constatação é verificada após comparar a coluna CÓDIGO da tabela CLIENTES com a coluna CÓDIGO da tabela TELEFONES.
- A coluna CÓDIGO é utilizada para fazer o relacionamento entre as Tabelas

Modelagem de dados

- Entretanto, para que possamos implementar, de forma correta, um BD utilizando algum SGBD, temos que passar por uma fase intermediária – e não menos importante - chamada modelagem de dados.

Modelagem de dados

- Quando estamos aprendendo a programar e desenvolver os algoritmos, em geral dividimos esta tarefa em três fases:
 1. Entendimento do problema;
 2. Construção do algoritmo;
 3. Implementação (linguagem de programação).



Modelagem de dados

- Em se tratando de banco de dados não é muito diferente:
 1. Entendimento do problema;
 2. Construção do modelo ER – entidade e relacionamento;
 3. Implementação (SGBD).

Modelagem de dados

- Entender determinado problema nem sempre é uma tarefa fácil, principalmente se você não está familiarizado com a área de atuação de seu cliente, e/ou o cliente não sabe descrever sobre o problema a qual você foi contratado para resolver.

Modelo Entidade-Relacionamento

- Antes da implementação em um SGBD, precisamos de uma descrição formal da estrutura de um banco de dados, de forma independente do SGBD.
- Essa descrição formal é chamada modelo conceitual.

Modelo Entidade-Relacionamento

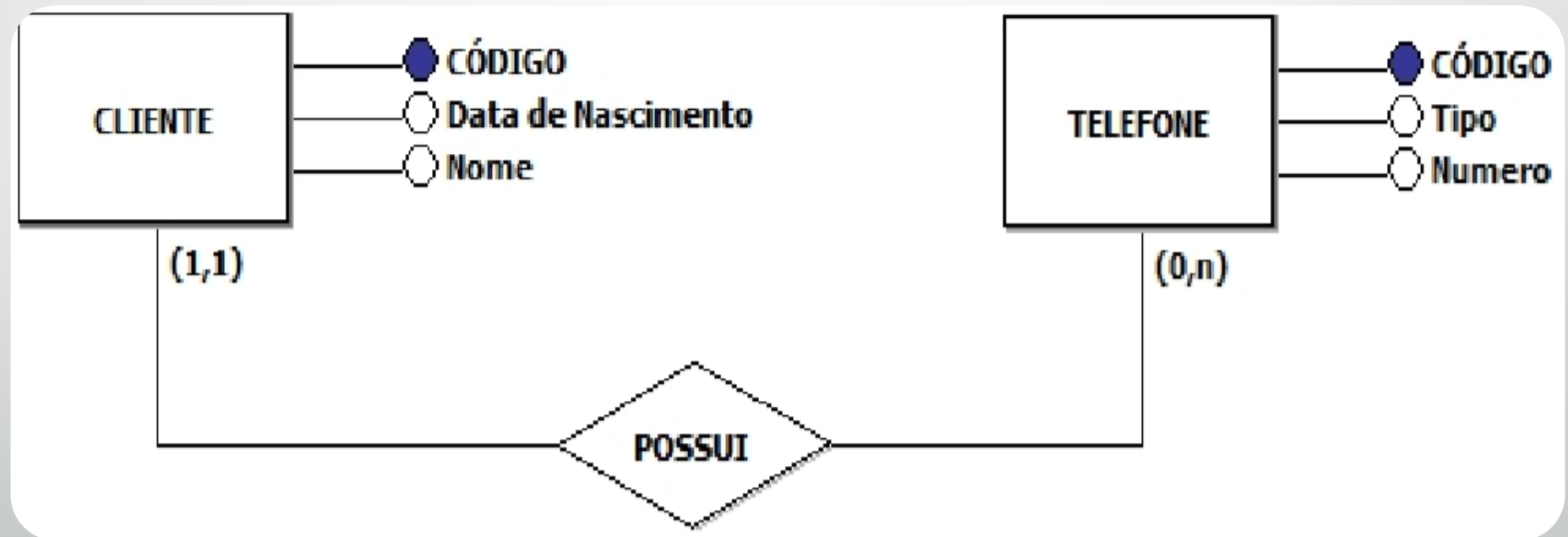
- Podemos comparar o modelo conceitual com o pseudocódigo/português estruturado em algoritmos, na qual construímos os algoritmos independentes de que linguagem de programação iremos desenvolver nossos programas.

Modelo Entidade-Relacionamento

- Costumamos representar um modelo conceitual através da abordagem entidade–relacionamento (ER).
- Nesta abordagem construímos um diagrama, chamado diagrama entidade-relacionamento (DER).
- Observe abaixo o diagrama que originou as tabelas CLIENTES e TELEFONES:

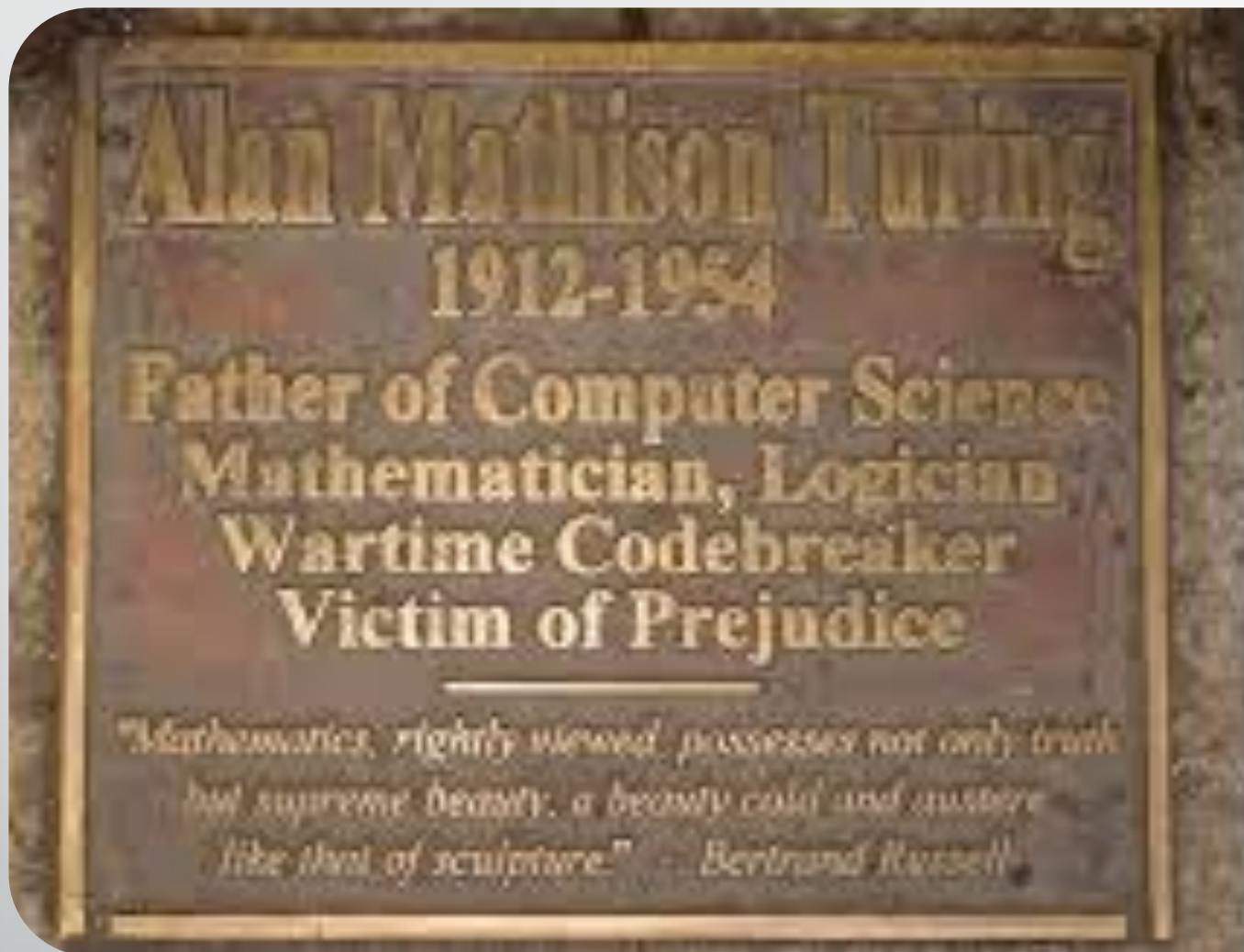
Modelo Entidade-Relacionamento

- Observe abaixo o diagrama que originou as tabelas CLIENTES e TELEFONES:



Modelo Entidade-Relacionamento

- Entidade pode ser entendida como uma “coisa” ou algo da realidade modelada onde deseja-se manter informações no banco de dados (BD).
- No exemplo anterior, as tabelas Clientes e Telefones são as entidades no modelo Entidade Relacionamento.



Modelo Entidade-Relacionamento

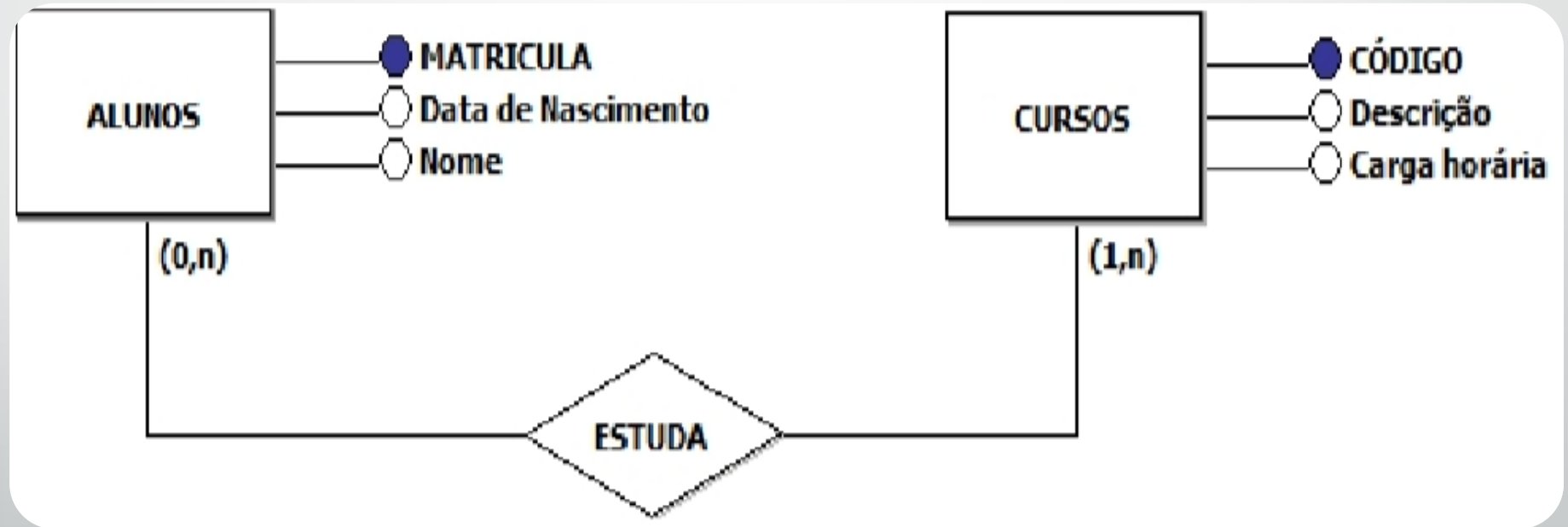
- Outro exemplo seria em um sistema escolar, algumas entidades podem ser os alunos, professores, horário, disciplinas e avaliações.

Modelo Entidade-Relacionamento

- Note que uma entidade pode representar tanto objetos concretos (alunos), quanto objetos abstratos (horário).
- A entidade é representada por um retângulo.
- Uma entidade se transformará em uma tabela no modelo físico de banco de dados.

Modelo Entidade-Relacionamento

- Uma entidade se transformará em uma tabela no modelo físico de banco de dados.



Modelo Entidade-Relacionamento

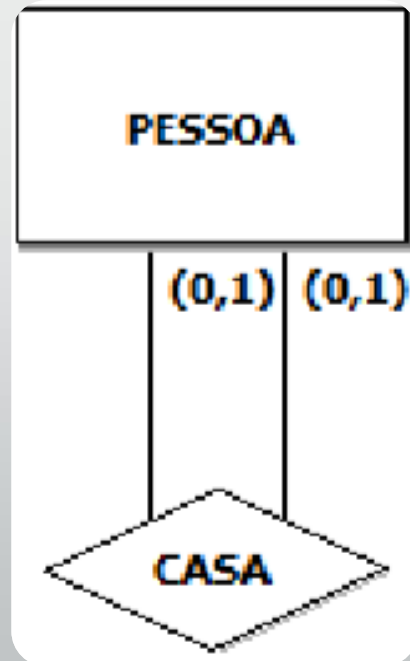
- A entidade ALUNO representa todos os estudantes sobre as quais se deseja manter informações no BD.
- Relacionamento é um conjunto de associações entre entidades.

Modelo Entidade-Relacionamento

- O relacionamento é representado por um LOSANGO e o nome do relacionamento (POSSUI, ESTUDA).
- Esse losango é ligado por linhas aos retângulos que representam as entidades participantes do relacionamento.

Modelo Entidade-Relacionamento

- Um relacionamento pode envolver ocorrências de uma mesma entidade.
- Neste caso, CASA é um relacionamento que envolve duas ocorrências da entidade PESSOA, ou seja PESSOA casa com PESSOA.



Modelo de dado Orientado a Objetos (OO)

Aplicações mais complexas

- CAD - Projeto Auxiliado por Computador – armazenar dados de projeto de engenharia.
- CASE - Engenharia de software Auxiliada por Computador – armazenar dados para apoiar desenvolvedores de software.
- Banco de dados hipertexto - armazenar textos com links.

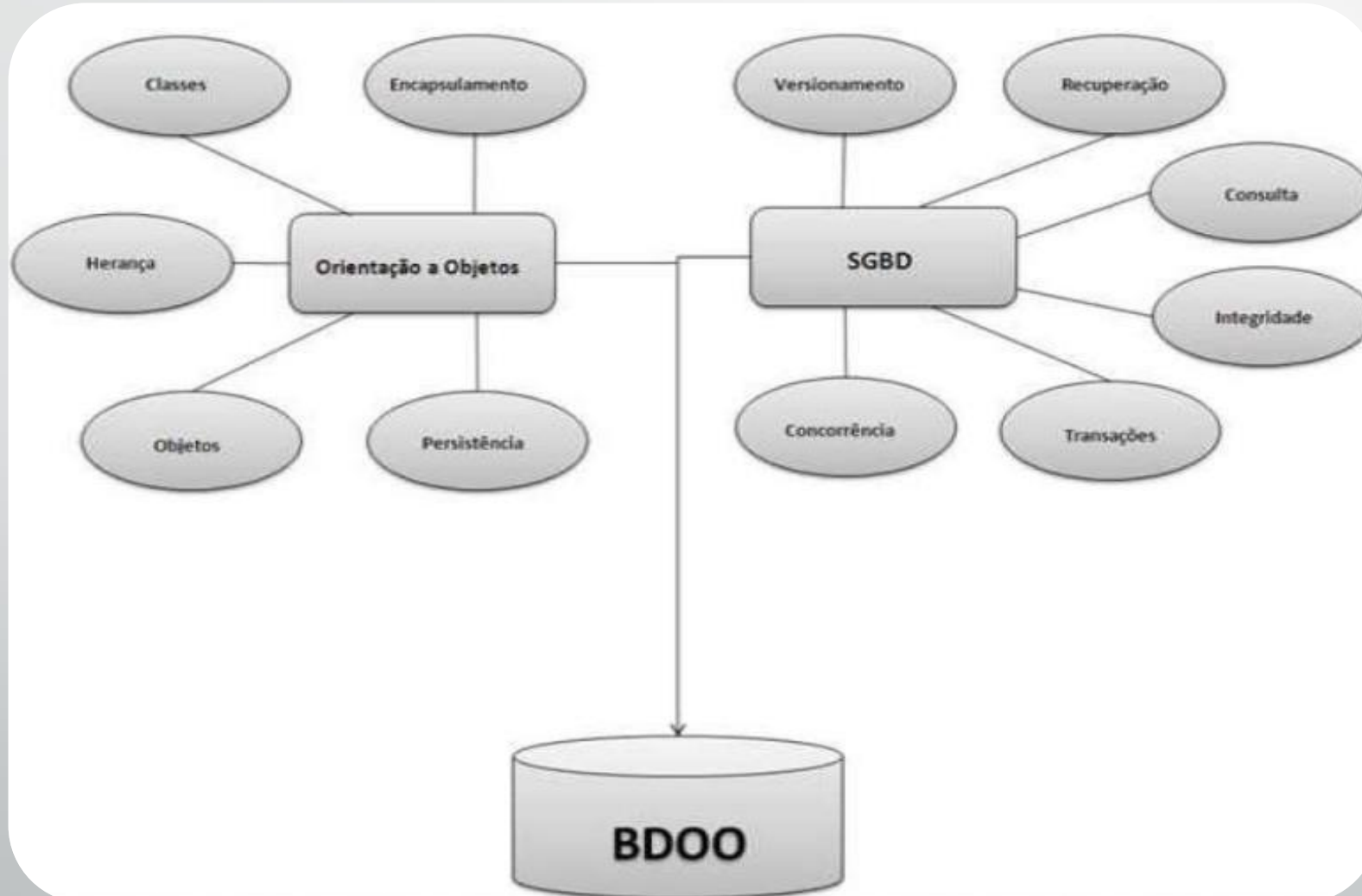
Modelo de dado Orientado a Objetos (OO)

- Maior motivação para o surgimento de novas abordagens para o gerenciamento de dados.
- BDOO - Integração do paradigma de orientação a objetos a sistemas de banco de dados.

BDOO - Conceito

- É um banco em que cada informação é armazenada na forma de objetos, e só podem ser manipuladas através de métodos pela classe que esteja o objeto.
- Junção dos conceitos de O.O com os conceitos de SGBD
- Unidade de armazenamento: objeto.

BDOO - Conceito



BDOO - Características

PERSISTÊNCIA

A característica que diferencia os BDOO em relação às LPOO.

- Nas LPOO os objetos existem apenas durante a execução do programa.

BDOO - Características

PERSISTÊNCIA

- Nos BDOO o objeto continua existindo a após o encerramento do programa manipuladas através de métodos pela classe que esteja o objeto.

BDOO - Características

Abordagens para persistência dos objetos

- Persistência por classe – objetos pertencentes à classe assim declaradas são persistentes.

BDOO - Características

Abordagens para persistência dos objetos

- Persistência por criação – objetos podem se tornar persistentes após sua criação.
- Persistência por referência – objetos referenciados por outros objetos se tornam persistentes.

BDOO - Características

OBJETOS COMPLEXOS

- Objetos que contém outros objetos.
- São formados por construtores (conjuntos, listas, tuplas, registros, coleções, arrays) aplicados a objetos simples(inteiros, booleanos, strings).
- Operações de acesso mais complexas.

BDOO - OPERAÇÕES MAIS COMUNS

- Criar um objeto
- Destruir um objeto
- Atualizar um objeto
- Recuperar dados do objeto
- Efetuar algum cálculo

HIERARQUIA DE TIPOS (CLASSES) E HERANÇA

- Classes similares postas em uma hierarquia de especialização.
- É possível a definição de novos tipos a partir de tipos outros predefinidos.
- O novo tipo herdará todas as funções (atributos e operações) do primeiro (supertipo).

Especificação e Padronização

- ODMG - Object Database Management Group responsável pela padronização de especificações para persistência de objetos de linguagem de programação OO para objetos em banco de dados;

Object Model – Modelo de Objeto

- É importante porque especifica os tipos de semântica que podem ser definidas explicitamente para um ODMS.
- Determina:
 - As características dos objetos;
 - Os relacionamentos;
 - Chamadas e identificadores;

Object Specification Languages – Linguagem de Especificação do Objeto

- São independentes da linguagem de programação e são usados para representar os sistemas de gerenciamento de dados de objeto
- Definem esquemas, estados e operações de um ODMS
- Facilitam a migração dos dados entre o banco e a aplicação, garantindo a interoperabilidade

Object Query Language – Linguagem de Consulta de Objetos

- É a linguagem de consulta que suporte o modelo de objetos
- É próxima a linguagem SQL e agrega as noções de objetos complexos, herança, identidade do objeto, polimorfismo, operações, etc.

Object Query Language – Linguagem de Consulta de Objetos

- Fornece um acesso declarativo ao objeto
- Não é uma linguagem completa e pode ser chamada dentro de uma linguagem de programação

ODMG - Especificação

- Define padrões para linguagens:
- C++
- Smalltalk
- Java

Exemplos de banco de dados OO

Principais SGBDOOs do mercado				
Nome	Open Source	Sistema Operacional	Fabricante	Site Oficial
EnterpriseDB	SIM	Linux, MacOS, Solaris e Windows	EnterpriseDB	http://www.enterprisedb.com/
Objectivity/DB	NÃO	Linux, Windows e Unix Posix	Objectivity Database Systems	http://www.objectivity.com
GemStone	NÃO	Windows, UNIX e Linux	GemStone System Inc.	http://www.gemstone.com
ConteXT	NÃO	Windows e UNIX	Unixspace	http://www.contextsoft.com/
Versant	NÃO	Windows, Linux e Unix	Versant Corp.	http://www.versant.com
Caché	NÃO	Windows, Linux e UNIX	Intersystems Software	http://www.intersystems.com.br
EyeDB	SIM	Linux e UNIX	Sysra Informatique	http://www.eyedb.org/
Jasmine	NÃO	Windows, Linux e UNIX	Computer Associates	http://www3.ca.com/
ORION	SIM	Linux e UNIX	Orion Group (Purdue University)	http://orion.cs.purdue.edu/
ObjectStore	NÃO	Windows, Linux e UNIX	Progress Software	http://www.objectstore.com

Exemplo - PostgreSQL

```
CREATE TYPE NOME AS (  
    primeiro_nome VARCHAR(20),  
    sobrenome VARCHAR(20)  
);
```

```
CREATE TYPE ENDERECO AS (  
    rua VARCHAR(40),  
    numero INTEGER,  
    bairro VARCHAR(40),  
    cep VARCHAR(10),  
    cidade VARCHAR(40),  
    estado CHAR(2),  
    pais VARCHAR(40)  
);
```

```
CREATE TYPE TELEFONE AS (  
    ddi CHAR(3),  
    ddd CHAR(3),  
    telefone VARCHAR(20)  
);
```

```
CREATE TYPE CLIENTE AS (  
    nm_cliente NOME,  
    end_cliente ENDERECO,  
    tel_cliente TELEFONE  
);
```

```
CREATE TABLE TB_CLIENTE(  
    client CLIENTE,  
    fg_ativo INTEGER  
);
```

Vantagens

- Capacidade de armazenamento de objetos;
- Poder de processamento de requisições;
- Não possui chave primária nem estrangeira, aumentando o desempenho das consultas e processos;
- Objetos se comunicam entre si através de mensagens;

Desvantagens

- Falta de padronização das linguagens de manipulação de dados;
- Alto custo de aquisição das novas tecnologias;
- Curva de aprendizagem e adaptação ao novo ambiente demorada;

Obrigado pela atenção

Sigo à disposição pelo e-mail:

marcio.lemos@senairs.org.br