



Universidade Federal do Rio Grande do Norte (UFRN)
Departamento de Informática e Matemática Aplicada (DIMAP)

Banco de dados

(IMD0401) - Aula 7 - Modelo físico e SQL

Prof^a: Ingridy Barbalho



Julho/2024

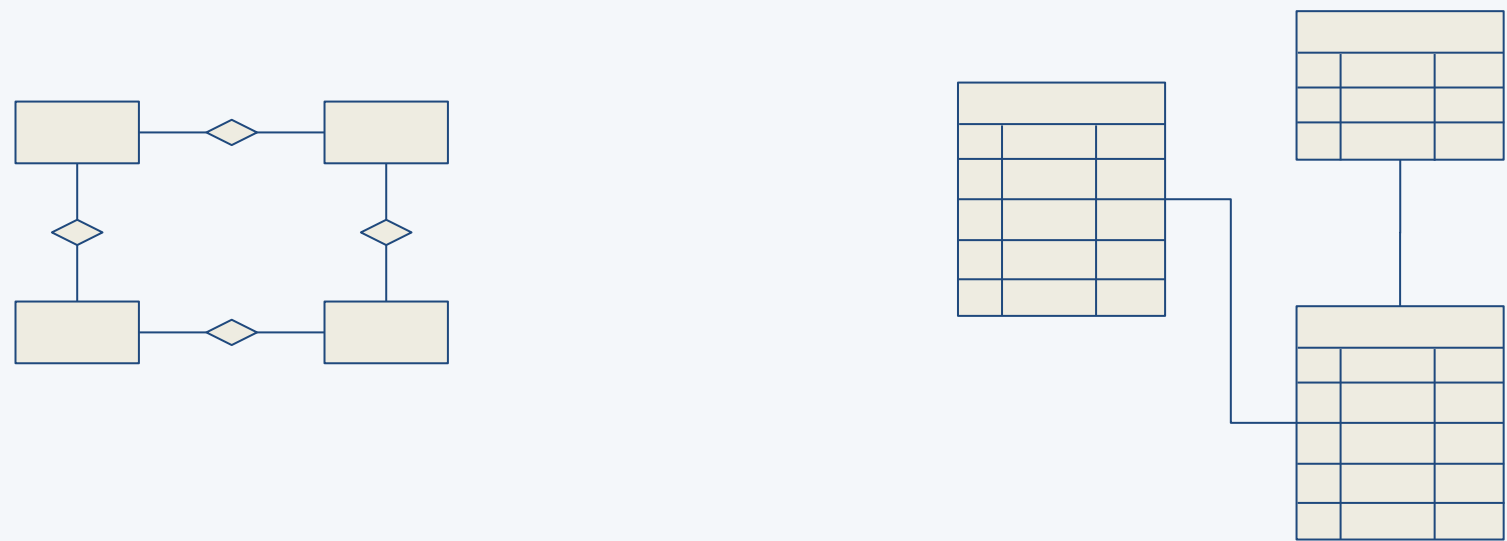


Roteiro

- Modelo físico
- Comandos SQL
- Postgresql

Modelo físico

- É uma representação detalhada de como os dados serão armazenados, organizados e gerenciados em um SGBD.



```
CREATE TABLE cliente (  
  cod serial PRIMARY KEY,  
  nome varchar(255) NOT NULL,  
  idade int,  
  email varchar(255)  
);
```

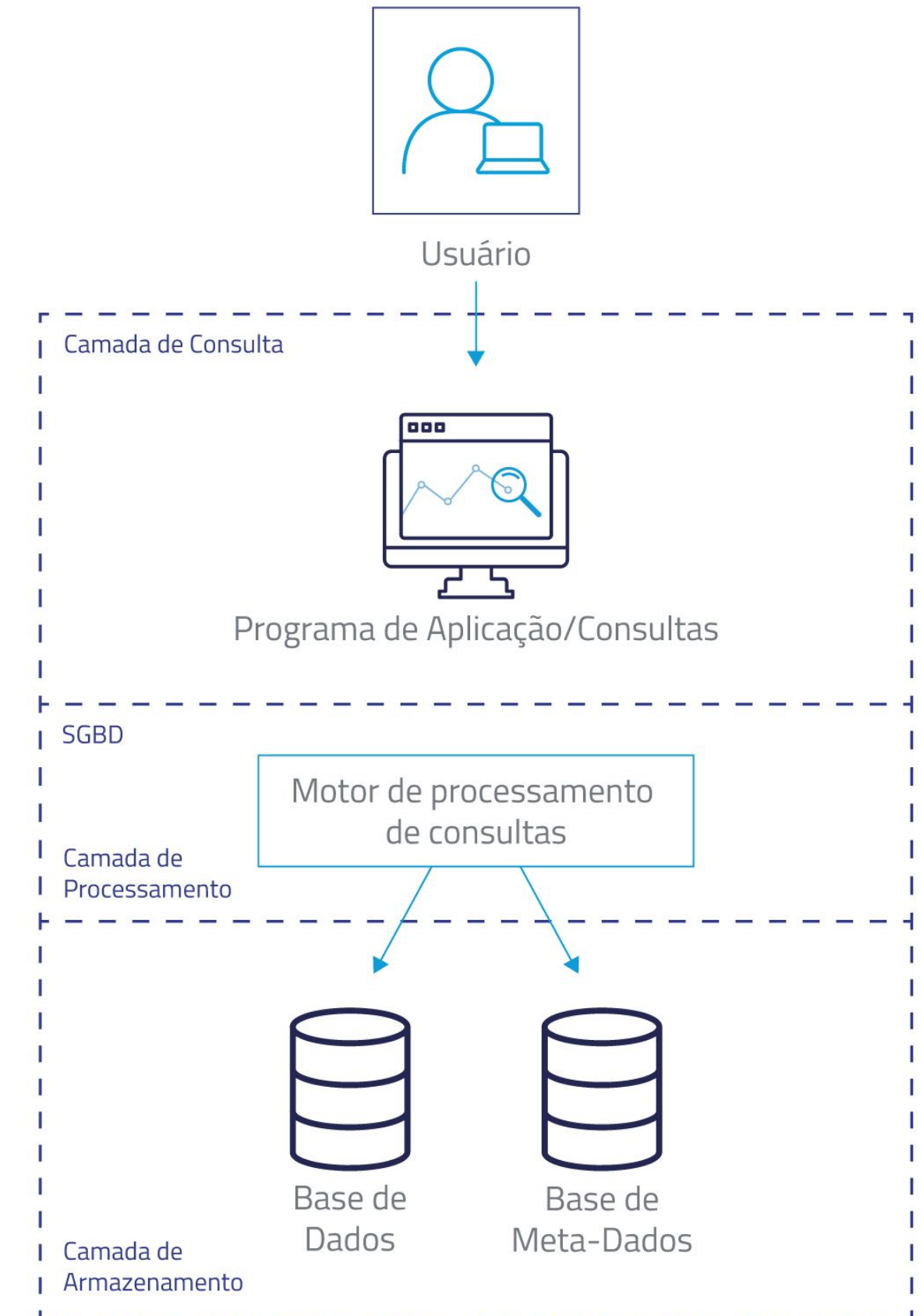


cliente			
cod	nome	idade	email
1	Maria	30	maria@gmail.com
2	João	40	joao@gmail.com

Structured Query Language (SQL)

Linguagem de consulta estruturada

- Linguagem padrão utilizada para gerenciar e manipular bancos de dados relacionais
- Amplamente adotada para realizar operações de definição, consulta, atualização e gerenciamento de dados em SGBD
 - Cláusulas
 - Operadores lógicos e relacionais
 - Funções de agregação
 - Expressões
- Postgresql (PGAdmin4)
- One Compiler - Simulador online
 - <https://onecompiler.com/postgresql/>



SQL

Subconjuntos SQL

DDL

Data Definition Language

Comandos usados para gerenciar as estruturas do banco de dados

DML

Data Manipulation Language

Comandos usados para manipular os dados armazenados em um banco

DQL

Data Query Language

Comando usados para consultar os dados que armazenamos no banco de dados

DCL

Data Control Language

Comandos para controlar o acesso aos dados do banco

DTL

Data Transaction Language

Comandos usados para gerenciar transações feitas no banco

Data Definition Language (DDL)

Criação de tabelas

- CREATE TABLE

```
CREATE TABLE nome (  
    coluna1 tipo(tamanho),  
    coluna2 tipo (tamanho),  
    ...  
);
```

tb_funcionario	
id_funcionario	serial
no_nome	varchar(255) NN
no_cargo	varchar(255)
dt_admissao	date NN
dt_demissao	date
id_endereco	int
dt_criacao	timestamp
dt_atualizacao	timestamp



```
CREATE TABLE tb_funcionario (  
    id_funcionario serial PRIMARY KEY,  
    no_nome varchar(255) NOT NULL,  
    no_cargo varchar(255),  
    dt_admissao date NOT NULL,  
    dt_demissao date,  
    id_endereco int,  
    dt_criacao timestamp,  
    dt_atualizacao timestamp  
);
```

- CREATE
- ALTER
- DROP
- RENAME

Data Definition Language (DDL)

Adicionar chave estrangeira

Criar a tabela com a chave primária e posteriormente criar a tabela que tem a chave estrangeira

```
CREATE TABLE tb_endereco (  
    id_endereco serial PRIMARY KEY,  
    no_logradouro varchar(255),  
    no_numero varchar(20),  
    no_bairro varchar(255),  
    no_cidade varchar(255),  
    no_cep char(8),  
    no_estado varchar(255),  
    no_pais varchar(255),  
    dt_criacao timestamp,  
    dt_atualizacao timestamp  
);
```

```
CREATE TABLE tb_funcionario (  
    id_funcionario serial PRIMARY KEY,  
    no_nome varchar(255) NOT NULL,  
    no_cargo varchar(255),  
    dt_admissao date NOT NULL,  
    dt_demissao date,  
    id_endereco int,  
    dt_criacao timestamp,  
    dt_atualizacao timestamp,  
  
    FOREIGN KEY (id_endereco) references  
    tb_endereco(id_endereco)  
);
```

Se você já tiver criado a tabela e desejar adicionar uma chave estrangeira posteriormente, você pode usar o comando ALTER TABLE.

```
ALTER TABLE tb_funcionario  
ADD CONSTRAINT id_endereco  
FOREIGN KEY (id_endereco)  
REFERENCES tb_endereco(id_endereco);
```


Data Definition Language (DDL)

ALTER TABLE

- Modificação de estruturas existentes no banco de dados

Adiciona uma nova coluna a uma tabela já existente

```
ALTER TABLE nome_da_tabela ADD COLUMN nome_da_coluna type;
```

Altera o tipo da coluna de uma determinada tabela

```
ALTER TABLE nome_da_tabela ALTER COLUMN nome_da_coluna type;
```

Renomeia uma coluna já existente*

```
ALTER TABLE nome_da_tabela RENAME COLUMN nome_da_coluna_para_renomear TO novo_nome;
```

Adiciona uma restrição a uma coluna já existente

```
ALTER TABLE nome_da_tabela ADD CONSTRAINT nome_da_restricao tipo_da_restricao (coluna);
```

Renomeia uma tabela do banco de dados*

```
ALTER TABLE nome_da_tabela RENAME TO novo_nome;
```


Data Definition Language (DDL)

DROP

- Remoção de objetos do banco de dados

Remove a coluna de uma tabela

```
ALTER TABLE nome_da_tabela DROP COLUMN nome_da_coluna;
```

Remove uma restrição já existente

```
ALTER TABLE nome_da_tabela DROP CONSTRAINT nome_da_restricao;
```

Remove uma tabela

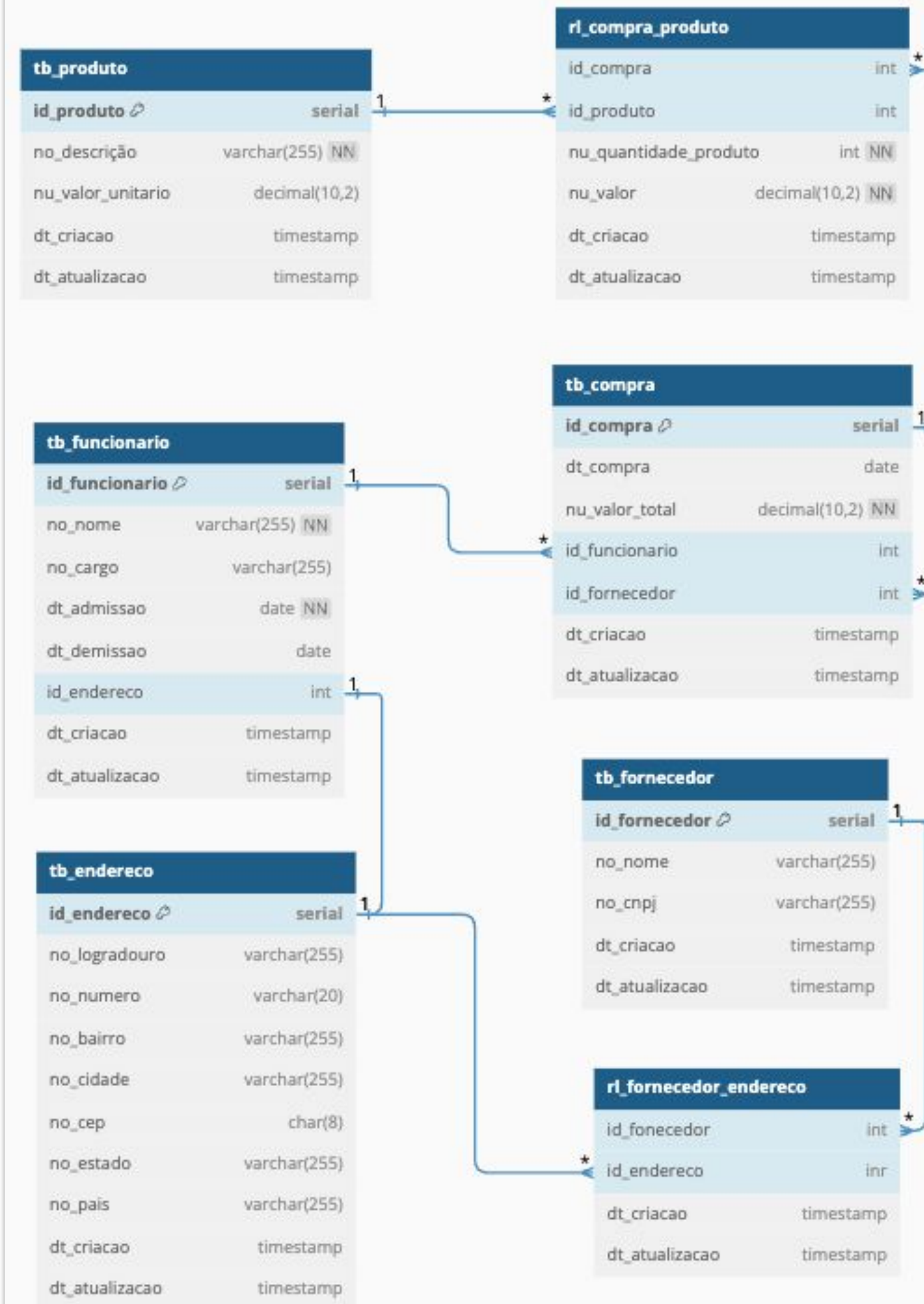
```
DROP TABLE nome_da_tabela;
```

Remove apenas os registros existente na tabela

```
TRUNCATE TABLE nome_da_tabela;
```

Vamos praticar?

- Elabore o modelo físico do seguinte banco de dados
- Adicione todas as chaves primárias e estrangeiras
- Adicione uma nova coluna na tabela de fornecedor
- Altere o tipo de dado dessa nova coluna
- Delete essa nova coluna
- Adicione uma nova tabela com novas colunas
- Altere o nome dessa coluna
- Apague apenas os registros dessa coluna
- Apague a tabela que você criou



Obrigada!

Contato: ingridy.marina@lais.huol.ufrn.br