

1



2

SQL – Structured Query Language

- ▶ Fundamentada no modelo Relacional de Codd (1970)
- ▶ Primeira versão – SEQUEL (1974)
- ▶ Padrão – BDs Relacionais
- ▶ Oficializado como linguagem – Ambiente Relacional – American National Standard Institute (ANSI) (1982)




3

3

Funções de um SGBD Convencional

- ▶ Definição de dados (DDL – *Data Definition Language*)
 - Criar (Create)
 - Destruir (Drop)
 - Modificar (Alter)
- ▶ Manipulação de dados (DML – *Data Manipulation Language*)
 - Consultar (Select) – DQL – *Data Query Language* (Linguagem de Consulta de Dados)
 - Inserir (Insert)
 - Excluir (Delete)
 - Atualizar (Update)
- ▶ Linguagem de Controle de Dados (DCL – *Data Control Language*)
 - Grant
 - Revoke
- ▶ Linguagem de Transação de Dados (DTL – *Data Transaction Language*)
 - Commit
 - Rollback
- ▶ SDL – *Storage Definition Language*
- ▶ VDL – *View Definition Language*



4

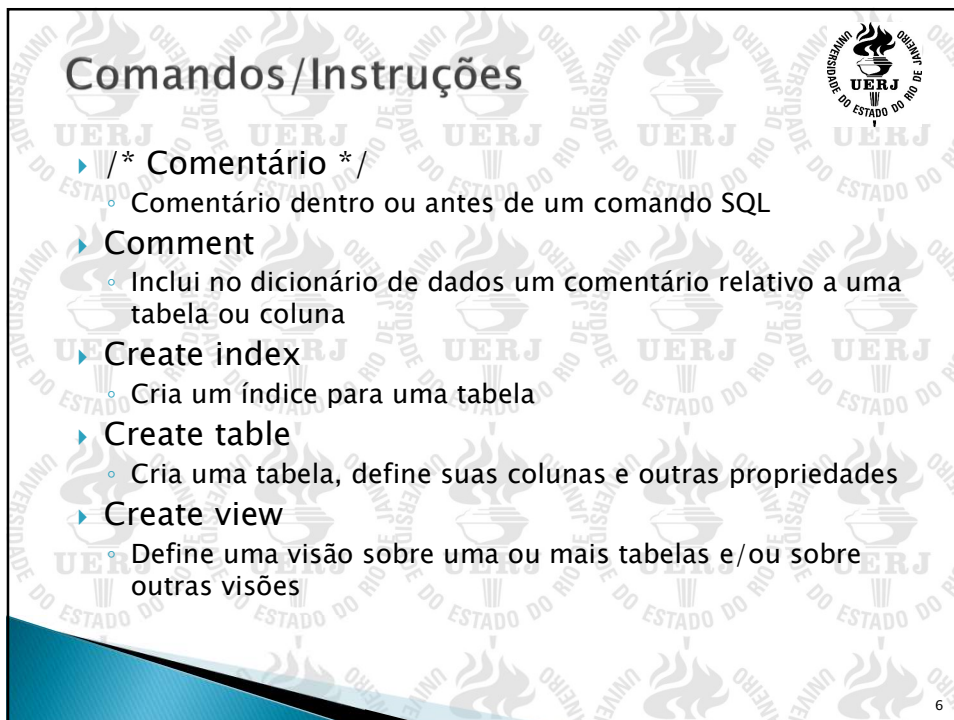
4



Vantagens da SQL

- ▶ Independência de fabricante
- ▶ Portabilidade entre computadores
- ▶ Múltiplas visões dos dados
- ▶ Definição dinâmica dos dados

5




Comandos/Instruções

- ▶ **/* Comentário */**
 - Comentário dentro ou antes de um comando SQL
- ▶ **Comment**
 - Inclui no dicionário de dados um comentário relativo a uma tabela ou coluna
- ▶ **Create index**
 - Cria um índice para uma tabela
- ▶ **Create table**
 - Cria uma tabela, define suas colunas e outras propriedades
- ▶ **Create view**
 - Define uma visão sobre uma ou mais tabelas e/ou sobre outras visões

6

Comandos/Instruções

- ▶ Drop
 - Exclui uma tabela, índice entre outros objetos de um banco de dados
- ▶ Alter table
 - Inclui ou redefine uma coluna em uma tabela existente
- ▶ Delete
 - Exclui linhas de uma tabela
- ▶ Update
 - Muda o valor de um ou mais atributos de uma tabela
- ▶ Insert
 - Inclui linhas em uma tabela ou visão
- ▶ Select
 - Seleciona linhas e colunas de uma ou mais tabelas




7

7

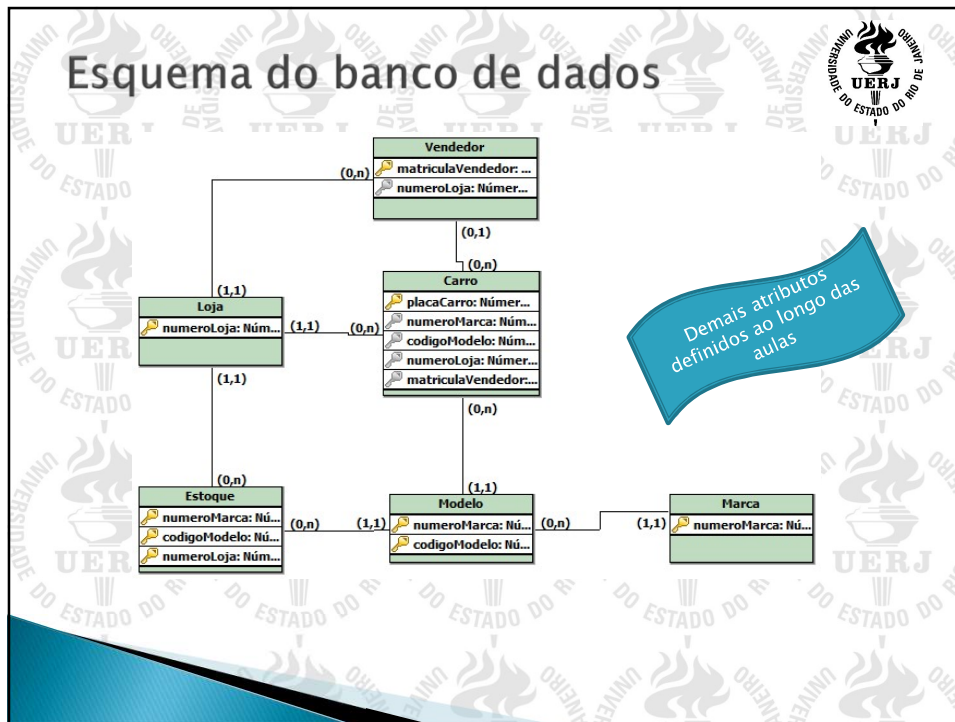
Comandos/Instruções

- ▶ Grant
 - Atribui privilégios relativos a tabelas e visões
- ▶ Revoke
 - Revoga privilégios relativos ao BD ou acesso a tabelas
- ▶ Commit
 - Efetiva no banco de dados, tornando permanentes, as modificações que já foram completadas
- ▶ Rollback
 - Desfaz, em caso de falha do sistema, qualquer alteração que ainda não tenha sido efetivada, de modo que a integridade do BD seja mantida
 - Permite ao usuário desfazer qualquer alteração antes de um COMMIT



8

8



9

Por que PostgreSQL?!?!?

- ▶ SGBD gratuito
- ▶ SGBD de código-fonte aberto
- ▶ SGBD popular – mundo acadêmico e no mercado
 - Mais popular em empresas públicas
- ▶ **Qualquer aluno pode baixar e instalar em sua casa!**
- ▶ Documentação Original
 - www.postgresql.org
- ▶ Documentação em Português
 - www.postgresql.org.br
- ▶ Como instalar (sem ser a partir do código-fonte)?
 - Linux (<http://www.vivaolinux.com.br/dica/Instalando-o-PostgreSQL-e-pgAdmin3-no-Ubuntu>)
 - Windows (<http://www.devmedia.com.br/instalando-postgresql/23364>)
- ▶ Ferramentas para manipular o banco de dados
 - PGADMIN III (<http://www.pgadmin.org/>)
 - PHPPGADMIN (<http://phppgadmin.sourceforge.net/doku.php>)

Scripts de aula disponíveis no moodle

10

Acessando o servidor de BD...

- SGBD PostgreSQL
- Endereço: labdb.ime.uerj.br/phppgadmin

postgres
labime@2019

11

DDL – Criação do banco de dados

PostgreSQL 8.4.21 running on 127.0.0.1:5432 – You are logged in as user "postgres"

phpPgAdmin: PostgreSQL

Create database?

Name:

Template:

Encoding:

Collation:

Character Type:

Comment:

Create Cancel

auto_<matricula>

template0

UTF-8

nome

Exercício Prático

collation é a forma como tipos textuais serão armazenados, comparados e ordenados.
 Character Type: classificação do caractere. Ex.: lower, upper and digit.

```
CREATE DATABASE nome
[ [ WITH ] [ OWNER [=] user_name ]
[ TEMPLATE [=] template ]
[ ENCODING [=] encoding ]
[ LC_COLLATE [=] lc_collate ]
[ LC_CTYPE [=] lc_ctype ]
[ TABLESPACE [=] tablespace ]
[ CONNECTION LIMIT [=] connlimit ] ]
```

12

Esquema

- ▶ Esquema (*Schema*)
 - Descrição do banco de dados
 - Coleção de objetos do banco de dados pertencente a um usuário específico.
 - Especificado durante a modelagem do banco de dados (diagrama do esquema)
 - Espera-se que tenha poucas modificações após sua criação
- ▶ Objetos do banco de dados
 - Tabelas, índices, visões, procedimentos armazenados etc.
- ▶ SGBDs tradicionais – Descrição (esquema) é separada dos dados
- ▶ Diagrama do esquema
 - Estrutura de cada tipo de registro, mas não as suas instâncias reais de registros

13

DDL – Criação do Esquema

- ▶ Padrão
 - Public
- ▶ Criando esquemas personalizados
 - SQL / Selecionar o seu banco de dados
- ▶ Concedendo autorização, para o novo esquema, a outro usuário, no momento de sua criação

```
create schema meuesquema1;
create schema meuesquema2;
```

```
CREATE SCHEMA schemaname AUTHORIZATION username;
```

Exemplos retirados da documentação da PostgreSQL: <https://www.postgresql.org/docs/9.1/ddl-schemas.html>

Exercício Prático


14

DDL – Criação de tabelas

► Formato básico:

```
CREATE TABLE <nome-da-tabela>
(<descrição das colunas>)
(<restrições de integridades>);
```

- ✓ <nome-da-tabela> – Nome da tabela a ser criada
- ✓ <descrição das colunas> – Lista de colunas da tabela; campos e seus respectivos tipos de dados (smallint, varchar2, varchar, integer, decimal, float, real, date, time)
- ✓ <restrições de integridade> – Lista de colunas que são tratadas como chave primária, chaves candidatas, chaves estrangeiras, checks constraints etc.



15

DDL – Criação da tabela Loja

- ✓ SQL
- ✓ Selecionar o nome do seu banco de dados
- ✓ Esquema: *public*

Loja(numeroLoja, nomeFantasiaLoja, enderecoLoja, bairroLoja, cidadeLoja, estadoLoja, CNPJLoja)

```
CREATE TABLE LOJA
(
    NUMEROLOJA          NUMERIC(3)      NOT NULL,
    NOMEFANTASIALOJA    VARCHAR(50)     NOT NULL,
    ENDerecoLOJA        VARCHAR(50)     NOT NULL,
    BAIRROLOJA          VARCHAR(25)     NOT NULL,
    CIDADELOJA          VARCHAR(30)     NOT NULL,
    ESTADOLOJA          VARCHAR(2)      NOT NULL,
    CNPJLOJA            NUMERIC(14)     NOT NULL
);
```

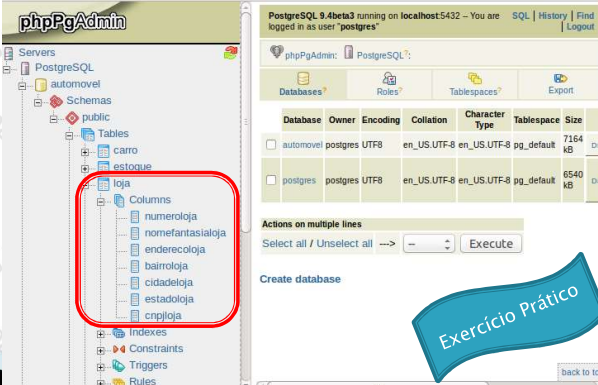
Exercício Prático

16

Verificando como ficou a tabela Loja

PLSQL:
\d loja

PHPPGADMIN:
SELECT column_name, is_nullable, data_type
FROM information_schema.columns
WHERE table_name = 'loja';



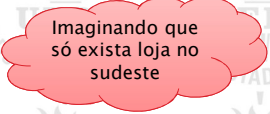
Exercício Prático

17

Integridade de colunas – CHECK

► Durante a criação da tabela

```
CREATE TABLE LOJA
(
    NUMEROLOJA          NUMERIC(3)          NOT NULL,
    NOMEFANTASIALOJA    VARCHAR(50)         NOT NULL,
    ENDERECOLOJA        VARCHAR(50)         NOT NULL,
    BAIRROLOJA          VARCHAR(25)         NOT NULL,
    CIDADELOJA          VARCHAR(30)         NOT NULL,
    ESTADOLOJA          VARCHAR(2)          NOT NULL,
    CNPJLOJA            NUMERIC(14)         NOT NULL,
    CONSTRAINT CK_ESTADO_LOJA
        CHECK (ESTADOLOJA in ('RJ', 'SP', 'ES', 'MG'))
);
```



Imaginando que só exista loja no sudeste

18

Estabelecendo *default* para o valor de uma coluna

► Durante a criação da tabela

```
CREATE TABLE LOJA
(
    NUMEROLOJA          NUMERIC(3)          NOT NULL,
    NOMEFANTASIALOJA    VARCHAR(50)         NOT NULL,
    ENDERECOLOJA        VARCHAR(50)         NOT NULL,
    BAIRROLOJA          VARCHAR(25)         NOT NULL,
    CIDADELOJA          VARCHAR(30)         NOT NULL,
    ESTADOLOJA          VARCHAR(2)          NOT NULL,
    CNPJLOJA            NUMERIC(14)         NOT NULL,
    DEFAULT 'RJ',
);
```

19

Identificando a chave primária

► Durante a criação da tabela

```
CREATE TABLE LOJA
(
    NUMEROLOJA          NUMERIC(3)          NOT NULL,
    NOMEFANTASIALOJA    VARCHAR(50)         NOT NULL,
    ENDERECOLOJA        VARCHAR(50)         NOT NULL,
    BAIRROLOJA          VARCHAR(25)         NOT NULL,
    CIDADELOJA          VARCHAR(30)         NOT NULL,
    ESTADOLOJA          VARCHAR(2)          NOT NULL,
    CNPJLOJA            NUMERIC(14)         NOT NULL,
    CONSTRAINT LOJA_PK PRIMARY KEY (NUMEROLOJA)
);
```

←

```
CREATE TABLE LOJA
(
    NUMEROLOJA          NUMERIC(3)          NOT NULL,
    NOMEFANTASIALOJA    VARCHAR(50)         NOT NULL,
    ENDERECOLOJA        VARCHAR(50)         NOT NULL,
    BAIRROLOJA          VARCHAR(25)         NOT NULL,
    CIDADELOJA          VARCHAR(30)         NOT NULL,
    ESTADOLOJA          VARCHAR(2)          NOT NULL,
    CNPJLOJA            NUMERIC(14)         NOT NULL,
    PRIMARY KEY ,
);
```

Quando for simples

20

Identificando a chave estrangeira

► Durante a criação da tabela

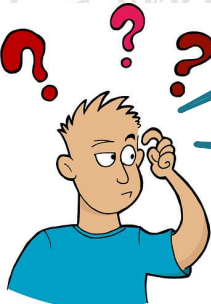
```
CREATE TABLE VENDEDOR
(
  MATRICULAVENDEDOR NUMERIC(5) NOT NULL PRIMARY KEY,
  NOMEVENDEDOR VARCHAR(50) NOT NULL,
  CPFVENDEDOR NUMERIC(11) NOT NULL,
  NUMEROLOJA NUMERIC(3) NOT NULL REFERENCES LOJA(NUMEROLOJA)
);
```

```
CREATE TABLE VENDEDOR
(
  MATRICULAVENDEDOR NUMERIC(5) NOT NULL PRIMARY KEY,
  NOMEVENDEDOR VARCHAR(50) NOT NULL,
  CPFVENDEDOR NUMERIC(11) NOT NULL,
  NUMEROLOJA NUMERIC(3) NOT NULL,
  CONSTRAINT VENDEDOR_LOJA_FK
  FOREIGN KEY (NUMEROLOJA)
  REFERENCES LOJA(NUMEROLOJA)
);
```

Exercício Prático

21

Impossibilidade de declarar FK?



Erro?!?

Ah! Falta declarar a PK!

```
CREATE TABLE VENDEDOR
(
  MATRICULAVENDEDOR NUMERIC(5) NOT NULL PRIMARY KEY,
  NOMEVENDEDOR VARCHAR(50) NOT NULL,
  CPFVENDEDOR NUMERIC(11) NOT NULL,
  NUMEROLOJA NUMERIC(3) NOT NULL
);
```

Exercício Prático

22

Modificando as definições em tabelas

```
ALTER TABLE <nome-da-tabela>
ADD (<especificações das colunas> | <restrição>, ...);
```

```
ALTER TABLE <nome-da-tabela>
ALTER (<especificações das colunas> | <restrição>, ...);
```

```
ALTER TABLE <nome-da-tabela>
DROP Constraint <nome da restrição>;
```

23

Acrescentando a integridade de colunas

```
ALTER TABLE LOJA
ADD CONSTRAINT CK_ESTADO_LOJA
CHECK (ESTADOLOJA in ('RJ', 'SP', 'ES', 'MG'));
```

Violação de integridade

```
INSERT INTO LOJA
(
    numeroLoja, nomeFantasiaLoja, enderecoLoja, bairroLoja, cidadeLoja, estadoLoja, CNPJLoja
)
VALUES
(
    1, 'DISTAC', 'RUA DAS LARANJEIRAS, 291', 'LARANJEIRAS', 'RIO DE JANEIRO', 'PB', '10101010000101'
);
```

```
INSERT INTO LOJA
(
    numeroLoja, nomeFantasiaLoja, enderecoLoja, bairroLoja, cidadeLoja, estadoLoja, CNPJLoja
)
VALUES
(
    1, 'DISTAC', 'RUA DAS LARANJEIRAS, 291', 'LARANJEIRAS', 'RIO DE JANEIRO', 'RJ', '10101010000101'
);
```

Exercício Prático

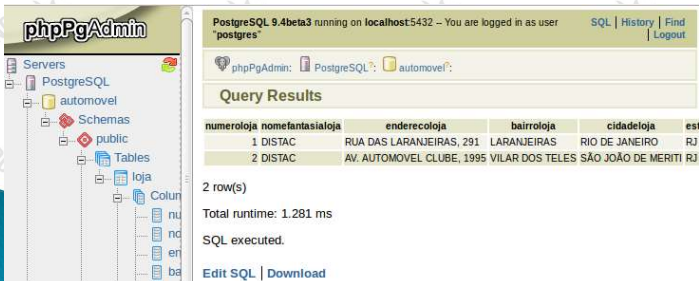
24

Acrescentando valor *default* para uma coluna

ALTER TABLE LOJA
ALTER COLUMN ESTADOLOJA SET DEFAULT 'RJ';

INSERT INTO LOJA
 (
 numeroLoja, nomeFantasiaLoja, enderecoLoja, bairroLoja, cidadeLoja, CNPJLoja
)
 VALUES
 (
 2, 'DISTAC', 'AV. AUTOMOVEL CLUBE, 1995', 'VILAR DOS TELES', 'SÃO JOÃO DE MERITI', '10011101000100'
);

SELECT * FROM LOJA;



Exercício Prático

25

Acrescentando chave primária

ALTER TABLE LOJA
ADD CONSTRAINT LOJA_PK PRIMARY KEY (NUMEROLOJA);

INSERT INTO LOJA
 (
 numeroLoja, nomeFantasiaLoja, enderecoLoja, bairroLoja, cidadeLoja, estadoLoja, CNPJLoja
)
 VALUES
 (
 1, 'IMCAUTO', 'R. VOLUNTÁRIOS DA PÁTRIA, 54', 'BOTAFOGO', 'RIO DE JANEIRO', 'RJ', '10202222000102'
);

INSERT INTO LOJA
 (
 numeroLoja, nomeFantasiaLoja, enderecoLoja, bairroLoja, cidadeLoja, estadoLoja, CNPJLoja
)
 VALUES
 (
 3, 'IMCAUTO', 'R. VOLUNTÁRIOS DA PÁTRIA, 54', 'BOTAFOGO', 'RIO DE JANEIRO', 'RJ', '10202222000102'
);

Mesmo
 identificador
 da "Distac"

Exercício Prático

26

Não existe esse identificador de Loja

Exercício Prático

Alterando coluna para *null* ou *not null*

Cuidado caso a
tabela já
possua dados!

Exercício Prático

14

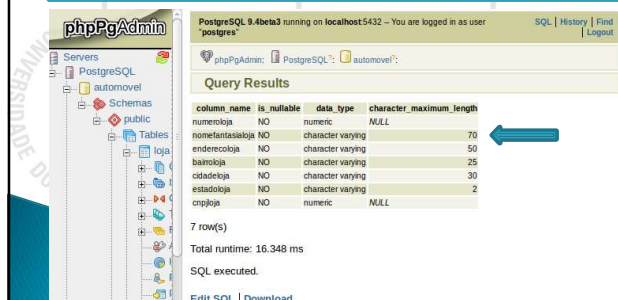
Alterando o tamanho de uma coluna

```
SELECT column_name, is_nullable, data_type, character_maximum_length  
FROM information_schema.columns  
WHERE table_name = 'loja';
```

```
ALTER TABLE LOJA  
ALTER COLUMN NOMEFANTASIALOJA TYPE VARCHAR(70);
```

VARCHAR(50)

```
SELECT column_name, is_nullable, data_type, character_maximum_length  
FROM information_schema.columns  
WHERE table_name = 'loja';
```



column_name	is_nullable	data_type	character_maximum_length
numerosloja	NO	numeric	NULL
nomefantasia	NO	character varying	70
enderecologja	NO	character varying	50
baixologja	NO	character varying	25
cidadeologja	NO	character varying	30
estadologja	NO	character varying	2
cepologja	NO	numeric	NULL

Exercício Prático

29

Removendo a chave estrangeira

```
ALTER TABLE VENDEDOR  
DROP CONSTRAINT VENDEDOR_LOJA_FK;
```

```
INSERT INTO VENDEDOR  
(  
    matriculaVendedor, nomeVendedor, numeroLoja, CPFVendedor  
)  
VALUES  
(  
    2, 'PEDRO DA SILVA', 5, 10508050701  
);
```

```
ALTER TABLE VENDEDOR  
ADD CONSTRAINT VENDEDOR_LOJA_FK  
FOREIGN KEY (NUMEROLOJA)  
REFERENCES LOJA (NUMEROLOJA);
```


```
DELETE FROM VENDEDOR WHERE MATRICULAVENDEDOR = 2;
```

Exercício Prático

30

Criando as chaves candidatas

```
CREATE TABLE LOJA
(
    NUMEROLOJA          NUMERIC(3)          NOT NULL,
    NOMEFANTASIALOJA    VARCHAR(50)         NOT NULL,
    ENDERECOLOJA        VARCHAR(50)         NOT NULL,
    BAIRROLOJA          VARCHAR(25)         NOT NULL,
    CIDADELOJA          VARCHAR(30)         NOT NULL,
    ESTADOLOJA          VARCHAR(2)          NOT NULL,
    CNPJLOJA            NUMERIC(14)         NOT NULL
);
```



Exercício Prático

```
ALTER TABLE LOJA
ADD CONSTRAINT UK_CNPJ_LOJA
UNIQUE (CNPJLOJA);
```

```
INSERT INTO LOJA
(
    numeroLoja, nomeFantasiaLoja, enderecoLoja, bairroLoja, cidadeLoja, estadoLoja, CNPJLoja
)
VALUES
(
    4, 'BARRAFOR', 'AV. AYRTON SENNA, 2541', 'BARRA DA TIJUCA', 'RIO DE JANEIRO', 'RJ', '10202222000102'
);
```


Mesmo CNPJ da Simcauto!

31

Verificando as restrições da tabela

```
SELECT CONSTRAINT_NAME, CONSTRAINT_TYPE
FROM information_schema.table_constraints
WHERE table_name='loja';
```

constraint_name	constraint_type
ck_estado_loja	CHECK
loja_pk	PRIMARY KEY
uk_cnpj_loja	UNIQUE
2200_17168_1_not_null	CHECK
2200_17168_2_not_null	CHECK
2200_17168_3_not_null	CHECK
2200_17168_4_not_null	CHECK
2200_17168_5_not_null	CHECK
2200_17168_6_not_null	CHECK
2200_17168_7_not_null	CHECK



Exercício Prático

32

Criando tabela a partir de outra tabela

```
CREATE TABLE nova_loja as
SELECT numeroLoja, nomeFantasiaLoja, enderecoLoja,
bairroLoja, cidadeLoja, estadoLoja, CNPJLoja
FROM loja;
```

```
SELECT *
FROM nova_loja;
```

numeroLoja	nomeFantasiaLoja	enderecoLoja	bairroLoja	cidadeLoja	estadoLoja	cnnpjLoja
1	DISTAC	RUA DAS LARANJEIRAS, 291	LARANJEIRAS	RIO DE JANEIRO	RJ	10101010000101
2	DISTAC	AV. AUTOMOVEL CLUBE, 1995	VILAR DOS TELES	SÃO JOÃO DE MERITI	RJ	10011101000100
3	SIMCAUTO	R. VOLUNTARIOS DA PATRIA, 54	BOTAFOGO	RIO DE JANEIRO	RJ	10202222000102

Incluindo as *constraints*...

```
CREATE TABLE tablename
(LIKE parent_table INCLUDING CONSTRAINTS)
```

33

Exclusões

- ▶ Banco de dados


```
DROP DATABASE [IF EXISTS] nome_bd;
```
- ▶ Esquema


```
DROP SCHEMA [IF EXISTS] nome_esquema [CASCADE | RESTRICT];
```
- ▶ Tabela


```
DROP TABLE nova_loja;
```

CASCADE
Apaga automaticamente todos os objetos contidos no esquema

RESTRICT
Se existir objeto, ele não remove o esquema

34