



Bases de Dados

- Linguagem SQL -

António Sousa

antonioribeirosousa@gmail.com

Workbench



Delegação Regional do Norte
Centro de Emprego e Formação Profissional do Porto





Workbench

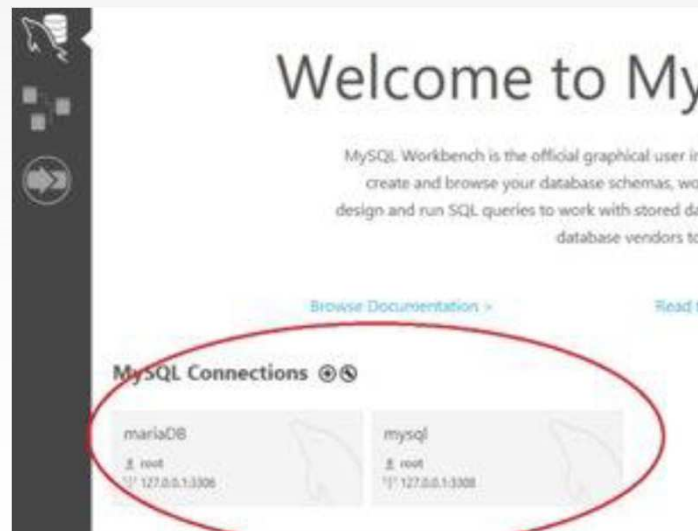
- Para usufruir da **área de administração do MySQL Workbench** é necessário ter um SGBD instalado
- Esse SGBD pode ser o **MySQL** ou o **MariaDB**





Ligação ao SGBD

- Para que consigam tornar o **Workbench** apto para **programação com SQL** terão de definir uma conexão ao SGBD (MySQL ou MariaDB)





SQL –Structured Query Language

- O SQL é uma poderosa linguagem declarativa que permite definir, questionar e manipular bases de dados.
- Para além das funcionalidades básicas, o SQL incorpora uma série de facilidades adicionais que permitem:
 - Definir restrições de integridade
 - Definir visões sobre os dados
 - Especificar transacções
 - Especificar permissões de segurança e de acesso
 - Criar índices de forma a otimizar o acesso
 - Ligar-se a outras linguagens de programação
 - ...
- O SQL é simultaneamente uma DDL (Data Definition Language) e uma DML (Data Manipulation Language)



SQL –Structured Query Language

- Originalmente, o SQL foi desenhado e desenvolvido pela IBM Research e era chamado de SEQUEL (Structured English QUery Language)
- Actualmente, o SQL é a linguagem standard para todos os sistemas comerciais de gestão de bases de dados relacionais
- A 1ª versão standard do SQL foi definida em 1986, o SQL1 ou SQL-86
- A 2ª versão standard foi definida em 1992, o SQL2 ou SQL-92
- A versão mais recente é o SQL3 ou SQL-99



Esquemas de base de dados

- Criar e eliminar um esquema de base de dados
 - Criar um novo esquema de base de dados:
 - CREATE DATABASE <nome da BD>;
 - Eliminar um esquema de base de dados:
 - DROP DATABASE <nome da BD>;



Selecionar uma base de dados criada

Selecionar uma base de dados já criada

- USE <nome da BD>;

Comentar código(sintaxe)

- -- <texto do comentário>;



Criação de Tabelas

- Permite criar uma nova relação (tabela) indicando o seu nome, atributos (nome, domínio, restrições e valores por omissão) e restrições de integridade (integridade da chave e integridade referencial).

```
CREATE TABLE <TABELA> (  
<ATRIB_1> <DOMÍNIO_1> [<OPÇÕES_1>],  
... ,  
<ATRIB_N> <DOMÍNIO_N> [<OPÇÕES_N>],  
[<RESTR_1>, ..., <RESTR_M>] );
```

- O SQL permite que uma tabela possua **tuplos repetidos**.



Criação de Tabelas

```
CREATE TABLE table_name (  
    column1 datatype,  
    column2 datatype,  
    column3 datatype,  
    ....  
);
```

Exemplo

```
CREATE TABLE Persons (  
    PersonID int,  
    LastName varchar(255),  
    FirstName varchar(255),  
    Address varchar(255),  
    City varchar(255)  
);
```



Regras para nomes de tabelas e colunas

- Nomes de **tabelas** e **colunas** não podem ter espaços ou caracteres especiais

1 **Nome Inválido**

```
2 CREATE TABLE minha tabela(  
3     col1 int,  
4     col2 varchar(255)  
5 );
```

The image shows a SQL statement with a red circle around the space in 'minha tabela', indicating it is an invalid name.

1 **Nome Válido**

```
2 CREATE TABLE minha_tabela(  
3     col1 int,  
4     col2 varchar(255)  
5 );
```

The image shows a SQL statement with a green circle around 'minha_tabela', indicating it is a valid name.



Domínio dos atributos

- Valores numéricos
 - **TINYINT**: 1 byte
 - **SMALLINT**: 2 bytes
 - **INT**: 4 bytes
 - **BIGINT**: 8 bytes
 - **FLOAT**: 4 bytes
 - **DOUBLE**: 8 bytes
 - **DECIMAL(N, D)**: N dígitos com D dígitos depois do ponto decimal



Domínio dos atributos

- Valores temporais
 - **DATE**: formato 'YYYY-MM-DD' ('2004-01-30')
 - **TIME**: formato 'HH:MM:SS' ('09:12:47')
 - **DATETIME**: formato 'YYYY-MM-DD HH:MM:SS' ('2004-01-30 09:12:47')
 - **TIMESTAMP**: formato YYYYMMDDHHMMSS (20040130091247)



Domínio dos atributos

- Valores lógicos
 - **BOOLEAN**: TRUE e FALSE
- Sequências de texto (*strings*)
 - **CHAR(N)**: *string* de comprimento fixo de N caracteres, $0 \leq N \leq 255$
 - **VARCHAR(N)**: *string* de comprimento variável até N caracteres, $0 \leq N \leq 65,535$
 - **TEXT**: *string* de comprimento variável até 65 Kbytes
 - **LONGTEXT**: *string* de comprimento variável até 4.3 Gbytes
- Criar a tabela para a relação DEPARTAMENTO



Domínio dos atributos

- Criar a tabela para a relação DEPARTAMENTO

```
CREATE TABLE DEPARTAMENTO (  
  Nome VARCHAR(50),  
  Num INT,  
  GerenteBI CHAR(9),  
  GerenteData DATE );
```



Valores por omissão e restrições sobre os atributos

- Definir o valor por defeito para um atributo:
`<ATRIB> <DOMÍNIO> DEFAULT <VAL>`
- Não permitir que um atributo possua valores NULL:
`<ATRIB> <DOMÍNIO> NOT NULL`
- Restringir os valores que um atributo pode tomar:
`<ATRIB> <DOMÍNIO> CHECK (<COND>)`



Valores por omissão e restrições sobre os atributos

- Criar a tabela para a relação DEPARTAMENTO

```
CREATE TABLE DEPARTAMENTO (  
Nome VARCHAR(50) NOT NULL,  
Num INT NOT NULL CHECK (Num > 0),  
GerenteBI CHAR(9) DEFAULT '000000000',  
GerenteData DATE);
```



Restrições de Integridade na Chave

- Definir a chave primária da tabela:
PRIMARY KEY (<ATRIB_1>, ..., <ATRIB_N>)
- Definir chaves secundárias da tabela:
UNIQUE (<ATRIB_1>, ..., <ATRIB_N>)



Restrições de Integridade na Chave

- Criar a tabela para a relação DEPARTAMENTO

```
CREATE TABLE DEPARTAMENTO (  
Nome VARCHAR(50) NOT NULL,  
Num INT NOT NULL CHECK (Num > 0),  
GerenteBI CHAR(9) DEFAULT '0000000000',  
GerenteData DATE,  
PRIMARY KEY (Num),  
UNIQUE (Nome) );
```



Restrições de Integridade na Chave

```
CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    PRIMARY KEY (ID)  
);
```



Auto Incremento

- Permite que um atributo definido como chave primária possa gerar um valor incremental sempre que for gerado um novo tuplo

Criar a tabela para a relação DEPARTAMENTO

```
CREATE TABLE DEPARTAMENTO ( Nome  
VARCHAR(50) NOT NULL,  
Num INT NOT NULL AUTO_INCREMENT,  
GerenteBI CHAR(9) DEFAULT '0000000000',  
GerenteData DATE, PRIMARY  
KEY (Num), UNIQUE (Nome) );
```



Auto Incremento

```
1 CREATE TABLE animals (  
2     id MEDIUMINT NOT NULL AUTO_INCREMENT,  
3     name CHAR(30) NOT NULL,  
4     PRIMARY KEY (id)  
5 );  
6
```



Restrições de Integridade Referencial

- Definir uma chave externa da tabela:
FOREIGN KEY (<ATRIB_1>, ..., <ATRIB_N>)
REFERENCES <TABELA> (<CHAVE_1>, ..., <CHAVE_N>)



Restrições de Integridade Referencial

- Criar a tabela para a relação DEPARTAMENTO.

```
CREATE TABLE DEPARTAMENTO (  
Nome VARCHAR(50) NOT NULL,  
Num INT NOT NULL AUTO_INCREMENT,  
GerenteBI CHAR(9) DEFAULT '000000000',  
GerenteData DATE,  
PRIMARY KEY (Num),  
UNIQUE (Nome),  
FOREIGN KEY (GerenteBI) REFERENCES  
EMPREGADO (NumBI) ) ;
```




Restrições de Integridade Referencial

```
CREATE TABLE Orders (  
    OrderID int NOT NULL,  
    OrderNumber int NOT NULL,  
    PersonID int,  
    PRIMARY KEY (OrderID),  
    FOREIGN KEY (PersonID) REFERENCES Persons(PersonID)  
);
```



Manutenção da Integridade Referencial

- As restrições de integridade referencial podem ser violadas quando inserimos ou removemos tuplos ou quando alteramos o valor de uma chave primária ou chave externa
- Quando isso acontece, o SQL por defeito rejeita essas operações



Manutenção da Integridade Referencial

- No entanto, é possível modificar esse comportamento para as operações de remoção (ON DELETE) e alteração (ON UPDATE) que violem a integridade referencial sobre tuplos que são referenciados pela chave externa de outras tabelas:
 - **ON DELETE SET NULL / ON UPDATE SET NULL:** coloca o valor NULL na chave externa dos tuplos que referenciam o tuplo removido/alterado.
 - **ON DELETE SET DEFAULT / ON UPDATE SET DEFAULT:** coloca o valor por defeito na chave externa dos tuplos que referenciam o tuplo removido/alterado
 - **ON DELETE CASCADE:** remove todos os tuplos que referenciam o tuplo removido
 - **ON UPDATE CASCADE:** actualiza com o novo valor a chave externa dos tuplos que referenciam o tuplo alterado



Manutenção da Integridade Referencial

- Criar a tabela para a relação DEPARTAMENTO.

```
CREATE TABLE DEPARTAMENTO (  
...  
GerenteBI CHAR(9) DEFAULT '0000000000',  
...  
FOREIGN KEY (GerenteBI) REFERENCES  
EMPREGADO (NumBI)  
ON DELETE SET DEFAULT ON UPDATE CASCADE );
```



Manutenção da Integridade Referencial

- Criar a tabela para a relação LOCALIZAÇÕES_DEP.

```
CREATE TABLE LOCALIZAÇÕES_DEP (  
  NumDep INT NOT NULL,  
  Localização VARCHAR(50) NOT NULL,  
  PRIMARY KEY (NumDep, Localização),  
  FOREIGN KEY (NumDep) REFERENCES DEPARTAMENTO (Num)  
  ON DELETE CASCADE ON UPDATE CASCADE );
```



Simplificações de Notação

- Se a chave primária for definida por um só atributo:
`<ATRIB> <DOMÍNIO> PRIMARY KEY`
- Se uma chave externa for definida por um só atributo.
`<ATRIB> <DOMÍNIO> REFERENCES <TABELA> (<CHAVE>)`



Simplificações de Notação

- Criar a tabela para a relação DEPARTAMENTO.

```
CREATE TABLE DEPARTAMENTO (  
Nome VARCHAR(50) ,  
Num INT PRIMARY KEY,  
GerenteBI CHAR(9) REFERENCES EMPREGADO (NumBI) ,  
GerenteData DATE ) ;
```



Criar Domínio

- Permite declarar um novo domínio para ser usado na definição de atributos.

```
CREATE DOMAIN <NOME> AS <DOMÍNIO> [CHECK (<COND>) ] ;
```




Criar Domínio

- Criar a tabela para a relação DEPARTAMENTO

```
CREATE DOMAIN CHARNOME AS VARCHAR(50);
```

```
CREATE DOMAIN INTPOS AS INT CHECK (INTPOS > 0);
```

```
CREATE TABLE DEPARTAMENTO (
```

```
Nome CHARNOME NOT NULL,
```

```
Num INTPOS NOT NULL,
```

```
GerenteBI CHAR(9) DEFAULT '000000000',
```

```
GerenteData DATE );
```



Criar Domínio

- Criar a tabela para a relação DEPARTAMENTO

```
CREATE DOMAIN CHARNOME AS VARCHAR(50);
```

```
CREATE DOMAIN INTPOS AS INT CHECK (INTPOS > 0);
```

```
CREATE TABLE DEPARTAMENTO (
```

```
Nome CHARNOME NOT NULL,
```

```
Num INTPOS NOT NULL,
```

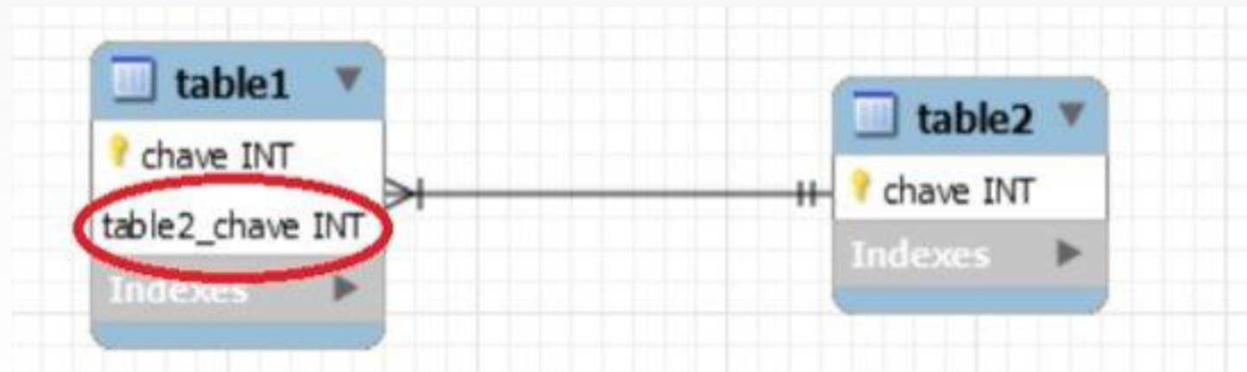
```
GerenteBI CHAR(9) DEFAULT '000000000',
```

```
GerenteData DATE );
```



Tipos de Relacionamento

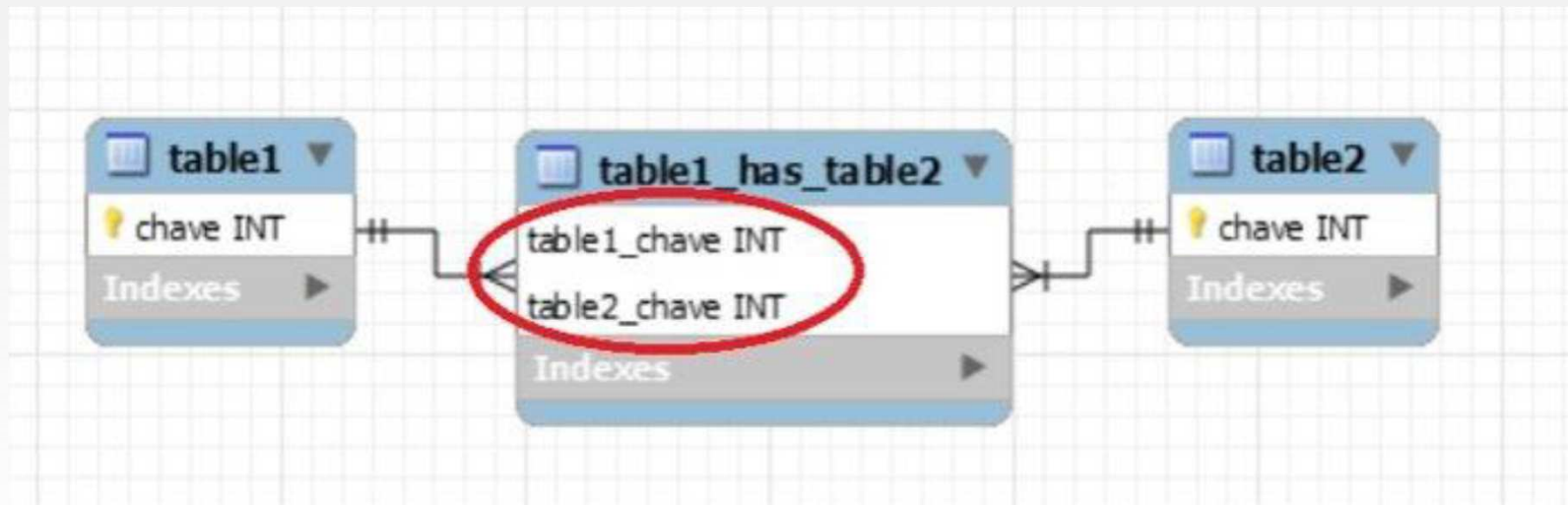
- Conforme visto na fase de desenho, o tipo de relacionamento está dependente da localização da chave estrangeira
- **Relacionamento de 1 para MUITOS** – Existe uma chave estrangeira na tabela de muitos





Tipos de Relacionamento

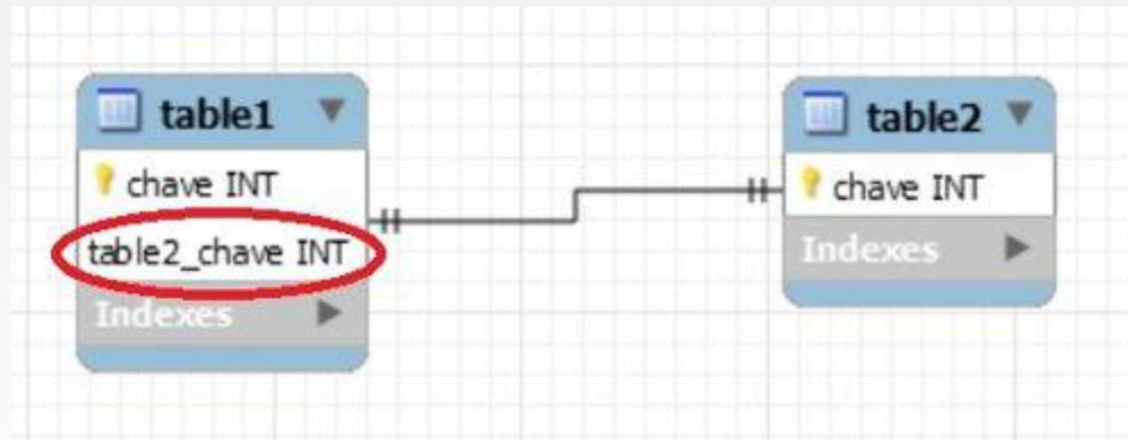
- **Relacionamento de MUITOS para MUITOS** – Existe uma tabela intermédia com as chaves estrangeiras





Tipos de Relacionamento

- **Relacionamento de 1 para 1** – Existe uma chave estrangeira numa das duas tabelas





Eliminar Tabelas

- Permite remover uma tabela (definição + dados).
DROP TABLE <TABELA> [**CASCADE** | **RESTRICT**];
- Opção **CASCADE**
 - Remove também todas as restrições (chaves externas em outras tabelas) e vistas que referenciam a tabela a remover
- Opção **RESTRICT**
 - Só remove a tabela se esta não for referenciada por nenhuma restrição ou vista
- Remover a tabela para a relação LOCALIZAÇÕES_DEP.
DROP TABLE LOCALIZAÇÕES_DEP CASCADE;



Alterar Tabelas

- Permite alterar os atributos ou restrições de uma tabela.

ALTER TABLE <TABELA> [**ADD** | **DROP** | **ALTER**] <OPÇÕES>;

- Opção **ADD** <ATRIB> <DOMÍNIO>

- Permite adicionar um novo atributo à tabela. O novo atributo terá valores NULL em todos os tuplos. A restrição NOT NULL não é permitida com esta opção.

- Opção **DROP** <ATRIB> [**CASCADE** | **RESTRICT**]

- Permite remover um atributo da tabela. As opções CASCADE e RESTRICT têm o mesmo significado que em DROP TABLE.

- Opção **ALTER** <ATRIB> [**SET** | **DROP**] <OPÇÕES>

- Permite alterar as restrições de um atributo da tabela.



Alterar Tabelas

- Adicionar um novo atributo à tabela DEPARTAMENTO.
ALTER TABLE DEPARTAMENTO **ADD** DataCriação DATE;
- Remover um atributo da tabela DEPARTAMENTO.
ALTER TABLE DEPARTAMENTO **DROP** DataCriação **CASCADE**;
- Remover uma restrição de um atributo da tabela DEPARTAMENTO.
ALTER TABLE DEPARTAMENTO **ALTER** GerenteBI **DROP DEFAULT**;
- Adicionar uma nova restrição a um atributo da tabela DEPARTAMENTO.
ALTER TABLE DEPARTAMENTO **ALTER** GerenteBI
SET DEFAULT '0000000000' ;



Alterar Tabelas

- Adicionar uma chave estrangeira à tabela DEPARTAMENTO.
ALTER TABLE DEPARTAMENTO **ADD FOREIGN KEY** (DocID)
REFERENCES Documentos (DocID) ;
- Remover uma chave estrangeira à tabela DEPARTAMENTO.
ALTER TABLE DEPARTAMENTO **DROP FOREIGN KEY** DocID;
- Adicionar uma chave primária à tabela DOCUMENTO.
ALTER TABLE DOCUMENTO **ADD PRIMARY KEY** (ID) ;
- Remover uma chave primária à tabela DOCUMENTO.
ALTER TABLE DOCUMENTO **DROP PRIMARY KEY**;



Entidade (Relação) vs Tabela

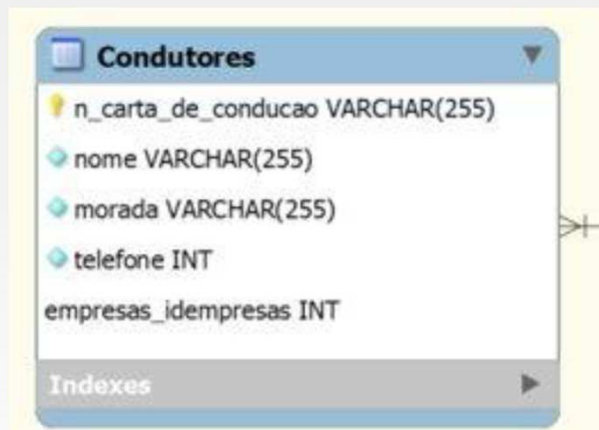

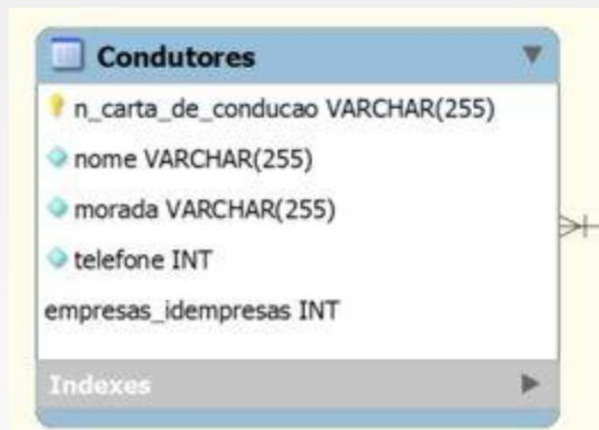


Tabela: Condutores

 N_CARTA	NOME	MORADA	TELEFONE	ID_EMPRESAS
Dado	Dado	Dado	Dado	Dado
Dado	Dado	Dado	Dado	Dado
Dado	Dado	Dado	Dado	Dado



Entidade (Relação) vs Script SQL



```
1 • CREATE DATABASE lista_negra;  
2 • CREATE TABLE condutores(  
3     n_carta int NOT NULL,  
4     nome varchar(255) NOT NULL,  
5     morada varchar(255) NOT NULL,  
6     telefone int NOT NULL,  
7     PRIMARY KEY (n_carta),  
8     FOREIGN KEY (id_empresas) REFERENCES empresas(id_empresas)  
9 );  
10  
11
```