

Banco de Dados – Ciência da Computação

SQL (*Structured Query Language*)

Prof. Mariane Moreira

mariane.souza@unifal-mg.edu.br

Universidade Federal de Alfenas

Instituto de Ciências Exatas

Departamento de Ciência da Computação

Operações de exclusão em tabelas – mais detalhes

- Exclui uma tabela em um esquema:
 - **DROP TABLE** empresa
- Exclui uma tabela em um esquema desde que ela **não** esteja sendo referenciada em nenhuma restrição de chave estrangeira ou visões
 - **DROP TABLE** empresa **RESTRICT**
- A cláusula **DELETE** também aceita a cláusula **RESTRICT**.

Operações de alteração estrutural em tabelas – mais detalhes

- A **estrutura** da tabela pode mudar utilizando o comando ALTER para vários tipos de mudança:
 - ALTER TABLE nomedoesquema.nomedatabela ALTERACAO
 - ALTER TABLE empresa.empregado ADD funcao varchar(12)
 - ALTER TABLE empresa.empregado DROP endereco
 - ALTER TABLE empresa.empregado DROP funcao RESTRICT
 - ALTER TABLE empresa.empregado ALTER salario SET DEFAULT=2000,00
 - ALTER TABLE empresa.empregado ADD CONSTRAINT chaveestrangeirasuper definição da constraint
 - ALTER TABLE empresa.empregado DROP CONSTRAINT chaveestrangeirasuper CASCADE
- obrigatório

Restrições em SQL

- Restrições permitidas no comando create table
 - Not null – ao inserir um registro é obrigatório informar algum valor para o atributo que seja “not null”

Create table Cliente

```
(  
    matricula char(5) not null,  
    nome char(50) not null  
)
```

Restrições em SQL

- Restrições permitidas no comando create table
 - Unique – determina que o atributo deve ser único, não pode se repetir em diferentes

Create table Cliente

```
(  
    matricula char(5) not null,  
    identidade char(7) not null,  
    nome char(50) not null,  
    primary key (matricula),  
    unique (identidade)  
)
```

Restrições em SQL

- Check
- Restringe valores que o atributo pode ter.

Create table Cliente

(

matricula char(5) not null,

identidade char(7) not null,

nome char(50) not null,

tipo char(30),

primary key (matricula),

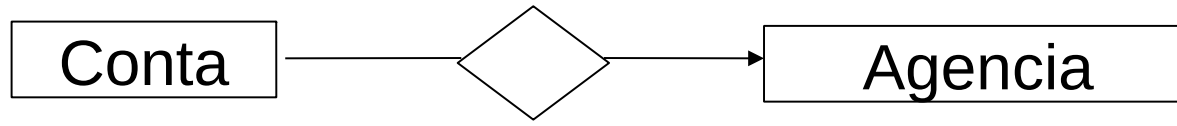
check (tipo in ('Graduacao', 'Pos-Graduacao'))

Restrições em SQL

- Check
- Pode também fazer comparações de valores.
Neste caso, a nota deve ser sempre maior que

```
Create table Avaliacao
```

```
(  
    matricula char(5) not null,  
    codigoDisciplina char(5) not null,  
    nota numeric(2,1) not null,  
    primary key (matricula, codigoDisciplina),  
    check (nota > 0)  
)
```



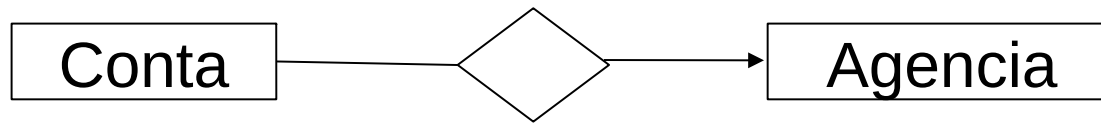
Restrições em SQL

- **Integridade Referencial**
 - Definição de chave estrangeira
 - Cláusula **foreign key**

Create table conta

```
(  
    numero char(5) not null,  
    saldo numeric(2,1) not null,  
    nome_agencia char(15) not null,  
    primary key (numero)  
    foreign key (nome_agencia) references agencia  
(nome_agencia)
```

Campo usado para referenciar na
tabela conta a chave de Agencia.



Restrições em SQL

- **Ações engatilhadas**
 - Foreign key com **cascade**
 - **Delete cascade** ou **Update cascade**

Neste caso, ao deletar/atualizar uma agencia, a tupla em conta, que faz referencia a essa agencia, também será deletada/atualizada.

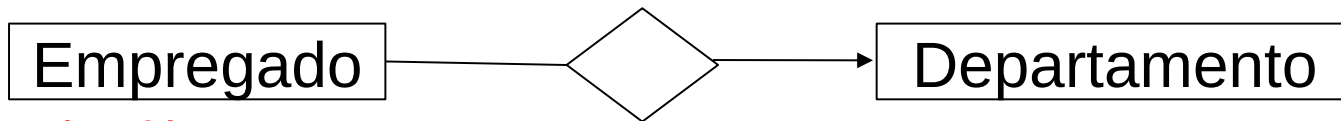
Create table conta

(

**numero char(5) not null,
saldo numeric(2,1) not null,
nome_agencia char(5) not null,
primary key (numero)**

foreign key (nome_agencia) references agencia

(nome_agencia) on delete cascade on update cascade)



Restrições em SQL

- Ações engatilhadas
 - Foreign key com **set null**
 - **Delete set null** ou **Update set null** (não muito usado, geralmente cascade)

Create table empregado

(

ssn integer(11) not null,
pnome varchar(30) not null,

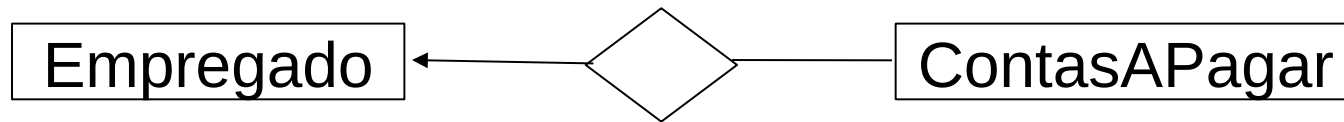
...

dnumero integer(11),
primary key (ssn)

foreign key (dnumero) **references** departamento

(dnumero) on delete set null on update cascade)

Neste caso, ao deletar/atualizar um departamento, a tupla em empregado, que faz referencia a esse departamento não será deletada. Neste caso, o valor nulo será colocado no lugar da referencia da chave. A atualização é em cascata.



Restrições em SQL

- **Ações engatilhadas**
 - Foreign key com **restrict**
 - **Delete restrict** (Update restrict é permitido, mas neste caso não faz muito sentido).

Create table contasAPagar

(

**idConta integer(11) not null,
montante double not null,
ssn integer(11),
primary key (idConta)**

**foreign key (ssn) references empregado(ssn) on
delete restrict)**

Neste caso, ao deletar um empregado, ele só poderá de fato ser deletado se não tiver nenhuma referencia sua em contas a pagar.

Inserção de tuplas em tabelas

- O comando **INSERT** pode ser utilizado para adicionar uma única tupla à tabela. Existem duas formas:
 - Informando todos os campos desejados **diretamente**:
 - **INSERT INTO** nometabela (campo1, campo2, ...) **VALUES** (valor1, valor2, ...)
 - Usado quando **não** se deseja inserir informação em **todos** os campos da tabela:
 - **INSERT INTO** empregado (ssn, pnome, minicial, unome, datanasc, salario) **VALUES** (123456789, 'Marcia', 'S.', 'Garcia', '1983-08-15', 30000)
 - Os campos desejados devem ser explicitados; Os demais serão definidos automaticamente com o seu valor *default*;

Inserção de tuplas em tabelas

- Informando apenas os valores:
 - `INSERT INTO` nometabela `VALUES` (valor1, valor2, ..., valor n)
 - Utilizado quando se deseja inserir informações **completas**;
 - `INSERT INTO` empregado `VALUES` (123456789, 'Marcia', 'S.', 'Garcia', '1983-08-15', 'F', 30000, 98765, 2)
 - Neste caso os campos **não** precisarão ser mencionados, mas os valores deverão estar na ordem especificada na tabela;

Inserção de tuplas em tabelas

- É possível adicionar **diversas** tuplas de **uma** só vez em uma tabela:
 - Os atributos de cada uma serão definidos entre parênteses;
 - Depois dos parênteses deve-se separar as tuplas usando vírgula;
 - **INSERT INTO** empregado (pnome, minicial, unome)
VALUES ('Marcio', 'M.', 'Rezende'), ('Mario', 'L.', 'Figueira')

Atualização de tuplas em tabelas

- A **atualização** de **valores** de **atributos** de uma tupla é feita utilizando o comando **UPDATE**:
 - **UPDATE** nometabela **SET** atributo1 = novovalor1, atributo 2 = novovalor2 ... **WHERE** condicao
 - Deseja-se alterar o **nome** do departamento número **2** para **DCC**. :
 - **UPDATE** departamento **SET** dnome = 'DCC' **WHERE** dnumero = 2;

Atualização de tuplas em tabelas

- Um único comando **UPDATE** pode alterar valores de **várias** tuplas;
 - Suponha que se deseja aumentar o salário de **todos** os empregados do departamento 'Exatas' em 10%.
 - **UPDATE** empregado **SET** salario = salario*1.1 **WHERE** dnumero **IN** (**SELECT** dnumero **FROM** departamento **WHERE** dnome = 'Exatas')
 - Neste caso, **várias** tuplas (referente a cada empregado que trabalha no departamento 'Exatas') serão atualizadas em **um** só comando **UPDATE**;

Atualização de tuplas em tabelas

- Caso sejam definidas ações engatilhadas do tipo **ON UPDATE CASCADE** as atualizações poderão ser **propagadas** para outras tabelas;
- Suponha a definição da seguinte **ação engatilhada** na restrição de chave estrangeira na tabela **empregado**:
 - **FOREIGN KEY** (dnumero) **REFERENCES** departamento(dnumero)
ON DELETE SET NULL ON UPDATE CASCADE

Atualização de tuplas em tabelas

- Neste caso, com a seguinte atualização da tabela departamento:
 - `UPDATE departamento SET dnumero = 2 WHERE dnumero = 5`
- O que acontece?
 - **Todas** as tuplas na tabela empregado que tiverem **relacionadas** com o departamento **5** terão seus valores do campo '**dnumero**' atualizados para **2**;

Remoção de tuplas em tabelas

- A remoção de **tuplas** de uma tabela é feita através do comando **DELETE**:
 - **DELETE FROM** nometabela **WHERE** condicao
 - O número de tuplas deletadas depende das condições definidas na cláusula **WHERE**.
 - Caso a cláusula **WHERE** não seja colocada, **todas** as tuplas da tabela mencionada serão deletadas!
 - Entretanto, a tabela continuará no banco de dados (somente **DROP** exclui uma tabela).

Remoção de tuplas em tabelas

- Suponha que se deseja remover todos os empregados que ganhem mais de 40.000 reais na empresa;
 - `DELETE FROM empregado WHERE empregado.salario > 40.000;`
- Suponha que se deseja excluir todos os empregados do departamento 'Solos';
 - `DELETE FROM empregado WHERE empregado.dnumero IN (SELECT dnumero FROM departamento WHERE departamento.dnome = 'Solos')`

Remoção de tuplas em tabelas

- Da mesma forma que acontece com **UPDATE**:
 - Caso sejam definidas **ações engatilhadas** do tipo **ON DELETE CASCADE** as exclusões poderão ser **propagadas** em outras tabelas;
- Suponha a definição da seguinte ação engatilhada na restrição de chave estrangeira na tabela dependente:
 - **FOREIGN KEY (essn) REFERENCES empregado(ssn) ON DELETE CASCADE ON UPDATE CASCADE**

Remoção de tuplas em tabelas

- Neste caso, com a seguinte exclusão na tabela empregado:
 - `DELETE FROM empregado WHERE ssn = 12345678;`
- **Todas** as tuplas na tabela dependente que tiverem **relacionadas** com o empregado cujo ssn é '**12345678**' serão **deletadas**;

Próximas aulas...

- Correção de exercícios SQL