

FACEC
F a c u l d a d e

Cont. Modelo Relacional e Introdução a SQL

Professor: Yuri Ferreira

Aula 5 – Modelo Relacional e Introdução a SQL

- Revisão aula anterior:
 - Conceitos principais de Bancos de Dados Relacionais;
 - Restrições de Esquema:
 - Restrições de domínio;
 - Restrições de chave;
 - Restrições de entidade;
 - Restrições de Integridade Referencial;
 - Chave estrangeira;
 - Diagrama de Esquema;

➤ Conteúdo:

- Restrições de Integridade Semântica;
- Operações de inserção e tratamento violação de restrições;
- Operações de delete e tratamento de violação de restrições;
- História da Linguagem SQL;
- Partes da Linguagem SQL;
- Recursos Básicos SQL;
- Tipos de domínios básicos e avançados;
- Definição de Esquema e Catalogo;
- Criação de Tabelas;

➤ Conteúdo:

➤ Especificando Restrições na tabela:

- Atributo;

- chave;

- Integridade referencial;

➤ Tratamento de restrições de integridade referencial;

- Restrições de Integridade Semântica
 - Podem ser especificadas na **aplicação** que atualiza o BD (mais comum) ou por mecanismos chamados **triggers** (gatilhos) e **assertions** (afirmações) usados na linguagem SQL;
 - Ex: “O salário de um empregado não pode ser maior que o do seu supervisor”;
 - “O número máximo de horas que um empregado pode trabalhar, em todos os projetos, é 56”;

- Operações de **atualização** e tratamento de **violações de restrição**:
 - BD relacional temos dois tipos de operações:
 - Recuperações e
 - Atualizações;
 - Operações básicas de atualização:
 - *Insert*: insere uma linha;
 - *Delete*: remove uma linha;
 - *Update*: modifica o valor de atributos de linhas existentes;

Aula 5 – Modelo Relacional e Introdução a SQL

➤ Exemplo “Esquema da relação Empregado”

EMPREGADO	PNOME	MINICIAL	UNOME	<u>SSN</u>	DATANASC	ENDERECO	SEXO	SALARIO	SUPERSSN	DNO
	John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
	Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
	Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
	Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
	Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
	Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
	Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
	James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	<i>null</i>	1

					DEPT LOCALIZACOES		<u>DNUMERO</u>	<u>DLOCALIZACAO</u>
							1	Houston
							4	Stafford
							5	Bellaire
							5	Sugarland
								Houston
DEPARTAMENTO	DNOME	<u>DNUMERO</u>	GERSSN	GERDATAINICIO				
	Pesquisa	5	333445555	1988-05-22				
	Administração	4	987654321	1995-01-01				
	Sede administrativa	1	888665555	1981-06-19				

Fonte: RUIZ, USP.

➤ Operação *Insert*:

- Ex. 1: inserir um novo Empregado com chave null;
- Inserir empregado <'Alfredo', 'B', 'Ribeiro', **null**, '1960-04-05', 'Rua A, 388', 'M', 28000, null, 4>
- Resultado:
- Viola restrição de integridade de entidade;
- Chave primária não pode ser null;

➤ Operação *Insert* (cont.):

- Ex. 2: Inserir um novo Empregado com chave já existente;
- Inserir empregado <'Alfredo', 'B', 'Ribeiro', **99988777**, '1960-04-05', 'Rua A, 388', 'M', 28000, null, 4>
- Resultado:
- Viola restrição de chave;
- Já existe uma linha com o mesmo valor de SSN;

➤ Operação *Insert* (cont.):

- Ex. 3: inserir um novo Empregado com chave estrangeira inexistente;
- Inserir empregado <'Alfredo', 'B', 'Ribeiro', 123456788, '1960-04-05', 'Rua A, 388', 'M', 28000, null, **7**>
- Resultado:
- Viola restrição de integridade referencial;
- Não existe valor de DNUMERO=7;

- Como resolver problemas com INSERT ?
 - Por padrão é rejeitada a inserção;
 - Poderia ser tratado a exceção na aplicação mostrando o motivo ao usuário;
 - Tentar corrigir o motivo da rejeição (No caso do exemplo de chave nula e chave existente);
 - Inserir uma nova linha que corresponda a relação (No exemplo da chave estrangeira inexistente, inserir uma linha com o número do Departamento = 7);

Aula 5 – Modelo Relacional e Introdução a SQL

- Operação *delete*: só viola a restrição de integridade referencial; Ex: delete tupla EMPREGADO com SSN='333445555';

EMPREGADO	PNOME	MINICIAL	UNOME	SSN	DATANASC	ENDERECO	SEXO	SALARIO	SUPERSSN	DNO
	John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
	Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
	Alicia	J	Zeh	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
	Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
	Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
	Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
	Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
	James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	null	1

DEPT	LOCALIZACOES	DNUMERO	DLOCALIZACAO
		1	Houston
		4	Stafford
		5	Bellaire
		5	Sugarland
			Houston

DEPARTAMENTO	DNOME	DNUMERO	GERSSN	GERDATAINICIO
	Pesquisa	5	333445555	1988-05-22
	Administração	4	987654321	1995-01-01
	Sede administrativa	1	888665555	1981-06-19

Fonte: RUIZ, USP.

➤ Operação *delete*:

- Só é possível deletar linhas de uma tabela se nenhuma das colunas forem referenciadas em outras tabelas;
- Por exemplo o Empregado John poderia ser deletado, pois ele não consta na tabela Departamento;
- Agora o Empregado “Franklin” não pode ser deletado, pois ele esta sendo referenciado no departamento “Pesquisa” como gerente;

- Como resolver problemas com *delete*:
 - Detectado a violação da restrição as **opções** para tratamento podem ser especificadas nas restrições da tabela, como:
 - *Restrict*: rejeitar sempre a exclusão;
 - *Cascade*: propagar exclusão nas demais linhas referenciadas em outras tabelas;
 - Set NULL ou Set Default: é possível modificar os valores das linhas referenciadas como NULL (caso não seja chave primária) ou outra linha de valor válido;
 - Ex: FOREIGN KEY (GERSSN) REFERENCES EMPREGADO ON DELETE **CASCADE** ou **SET NULL**;

A linguagem SQL

➤ História do SQL:

- IBM desenvolveu a versão original do SQL, inicialmente chamada de Sequel, teve primeiros passos por volta de 1970;
- Passado alguns anos, evoluiu para SQL (*Structured Query Language*);
- Se tornou a linguagem padrão para banco de dados Relacionais;
- Padrão SQL-86 ou SQL1 foi publicado pelos órgãos ANSI e ISO;
- A próxima expansão da linguagem veio com SQL-92, ou SQL2;
- O padrão SQL:1999 começou como SQL3.
- 2 atualizações vieram em sequência SQL:2003 e SQL:2006, acrescentaram recursos XML;
- Outras atualizações surgiram em 2008 com incorporação de dados de objeto e a SQL:2011;

➤ Partes da Linguagem SQL:

- DDL: Definição de **esquemas** de tabelas; **exclusão** de tabelas; e **modificação** de esquemas;
- DML: Linguagem de consulta baseada em álgebra relacional e no cálculo relacional (operações matemáticas);
- Integridade: DDL especifica **restrições** de integridade e os dados armazenados precisam satisfazer essas restrições;
- Definição de *Views* (visões): inclui comando para definição de visões;

- Partes da Linguagem SQL (Cont.):
 - Controle de Transações: início e fim das transações;
 - SQL embutida e SQL dinâmica: definem como as instruções são incorporadas dentro das linguagens de programação (C, C++, Java, PHP, Python, NodeJS, etc);
 - Autorização: DDL inclui comandos para especificar direitos de acesso para tabelas e visões;

- Recursos Básicos DML, DDL e da SQL:
 - Esquema para cada relação;
 - Domínio de cada atributo;
 - Restrições de Integridade;
 - Conjunto de índices mantidos para cada relação;
 - Informações de segurança e Autorização para cada relação;
 - Estrutura de armazenamento físico;

- Tipos de **Domínios** básicos e avançados:
 - Tipo **texto**: Char, Character ou varchar;
 - Tipo **inteiro**: Int, Integer ou Smallint;
 - Tipo **ponto flutuante**: float, real, double;
 - Tipo ponto flutuante estendidos: Numeric(p,d), Decimal(p,d), onde p é de precisão (total de números) e d números de ponto flutuante;
 - Ex: Numeric(3,1) => 44,5 e não pode ser 444,5 e nem 0,32; o mesmo se aplica para Decimal;
 - Tipo de **datas e horas**: Date, Time, Timestamp;
 - Ex: pode ser extraído ano, mês, dia, hora, minuto, segundo;
 - Também pode ser utilizado tipos estáticos como current_date, current_time, current_timestamp, ambos fornecerão dados sobre o tempo atual;

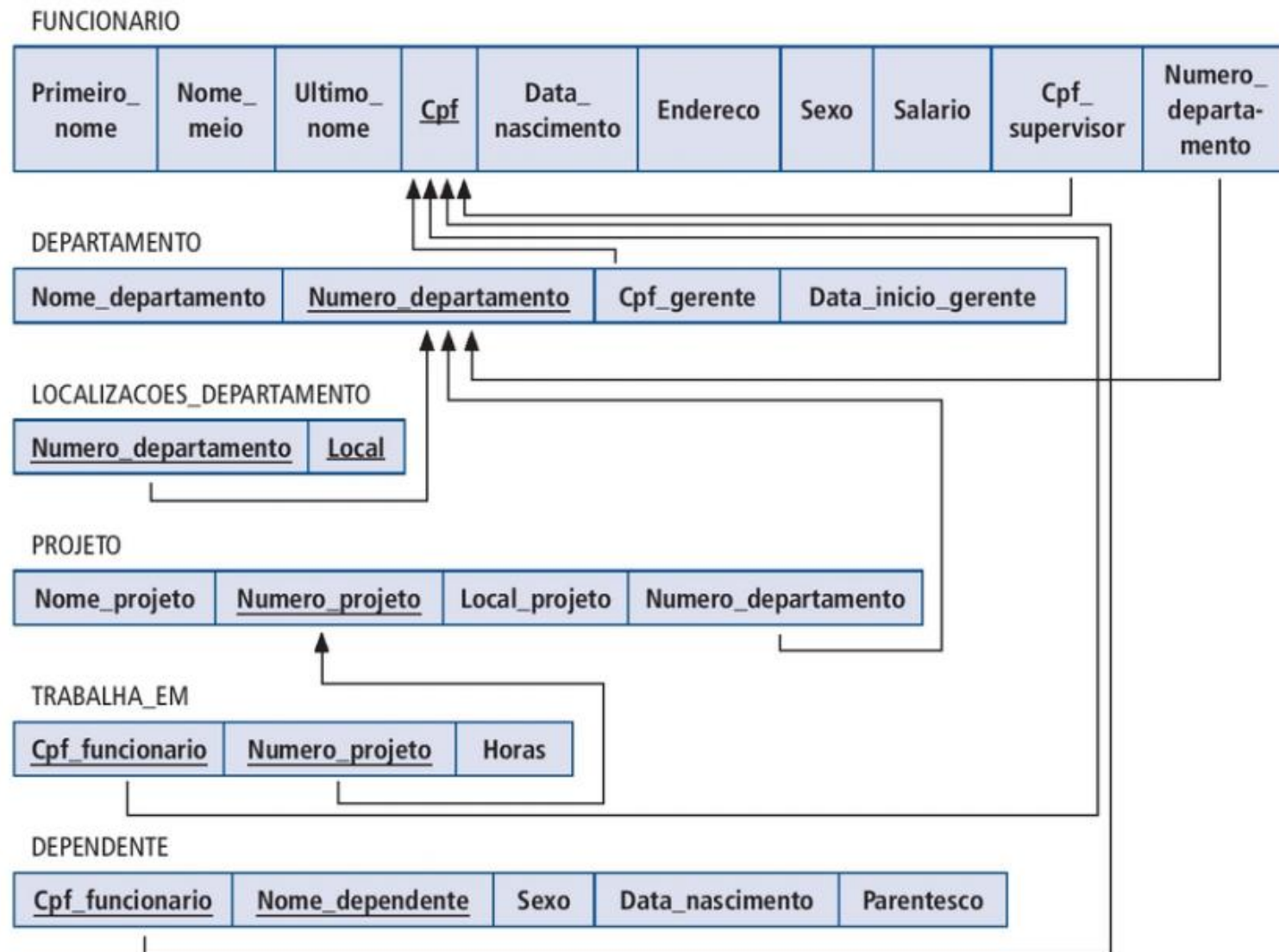
- Tipos de Domínios básicos e Avançados (Cont.):
 - Tipos de **grandes capacidades**: CLOB, BLOB;
 - CLOB: tipo caractere, pode ser utilizado para armazenar grandes arquivos texto, Ex: CLOB(20M);
 - BLOB: tipo binário, pode ser utilizado para armazenar imagens e filmes entre outros arquivos, Ex: BLOB(30G);
 - É possível também criar tipos definidos pelo usuário:
 - **CREATE TYPE DOLARES AS NUMERIC(12,2) FINAL;**
 - Neste caso o tipo poderia chegar de \$0 até \$9.999.999.999,99 (=~ 10 bilhões);

- Definição de **Esquema** e **Catalogo** em SQL:
 - Foi incorporado inicialmente para agrupar tabelas e outras construções que pertencem ao BD;
 - Em alguns SGDB`s os esquemas são chamados de BD;
 - Um Esquema é identificado por:
 - Um **nome de esquema** + **identificador de autorização** + descritores dos **elementos** do esquema (opcional);
 - Estes elementos incluem tabelas, tipos, restrições, views, domínios e concessões de privilégios);
 - Exemplo Instrução para criar um esquema:
- **CREATE SCHEMA** Empresa **AUTHORIZATION** 'Jsilva';

- O comando CREATE TABLE:
 - CREATE TABLE r(**A1D1**, ..., **AnDn**,
 <restrição_de_integridade1>,
 <restrição_de_integridaden>)
- Onde r é o nome da relação (tabela);
 - **A** são os nomes dos atributos;
 - **D** são os tipos dos domínios dos valores;
 - podem existir diversas **restrições**;

➤ Ex:

Tabelas do Esquema Empresa;



➤ Vamos criar 2 tabelas para exemplificar:

```
create table funcionario (  
    primeiro_nome varchar(15) not null,  
    nome_meio char,  
    ultimo_nome varchar(15) not null,  
    cpf char(11) not null,  
    dt_nascimento date,  
    endereco varchar(30),  
    sexo char,  
    salario decimal(10,2),  
    cpf_supervisor char(11),  
    numero_departamento int not null,  
    primary key (cpf)  
);
```

```
create table departamento (  
    nome_departamento varchar(15) not null,  
    numero_departamento int not null,  
    cpf_gerente char(11) not null,  
    data_inicio_gerente date,  
    primary key (numero_departamento),  
    unique (nome_departamento),  
    foreign key (cpf_gerente)  
        references funcionario(cpf)  
);
```

<https://kripken.github.io/sql.js/GUI/>

➤ Restrições de **atributo**:

➤ **NOT NULL** ou **DEFAULT** <valor>;

```
create table funcionario (  
    ...  
    numero_departamento int not null default 1,  
    ativo int not null default 1,  
    primary key (cpf)  
);
```

➤ Restrições de **atributo**:

- A Clausula **CHECK** também pode ser utilizada para limitar valores de atributo;

```
create table funcionario (  
    ...  
    numero_departamento int not null  
        CHECK (numero_departamento > 0 AND numero_departamento < 21),  
    ativo int not null default 1,  
    primary key (cpf)  
);
```

➤ Restrições de **chave**:

- A cláusula **PRIMARY KEY** pode especificar um ou mais atributos que compõem a chave primaria;
- Se ela for única pode ser passada diretamente a frente do atributo;
- A cláusula **UNIQUE** especifica as chaves candidatas;

```
create table departamento (  
    nome_departamento varchar(15) UNIQUE,  
    numero_departamento int PRIMARY KEY,  
    cpf_gerente char(11) not null,  
    data_inicio_gerente date,  
    foreign key (cpf_gerente)  
        references funcionario(cpf)  
);
```

- As restrições também podem ser **nomeadas** utilizando a palavra **CONSTRAINT**:

```
create table funcionario (  
    ...  
    cpf char(11) not null,  
    cpf_supervisor char(11),  
    numero_departamento int not null,  
    constraint chave_prim_func primary key(cpf),  
    constraint chave_estr_supervisor  
        foreign key (cpf_supervisor)  
        references funcionario(cpf),  
    constraint chave_estr_departamento  
        foreign key (numero_departamento)  
        references departamento(numero_departamento)  
);
```

- Restrições de **integridade referencial**:
 - É utilizada a cláusula **FOREING KEY** (chave estrangeira);
 - Lembrando: Ela pode ser **violada** quando linhas são **inseridas**, **excluídas** ou ainda quando uma chave é **atualizada**;
 - As ações de disparo referencial podem ser:
 - **NO ACTION**;
 - **CASCADE**;
 - **SET NULL**;
 - **SET DEFAULT**;

- Projetista do BD pode especificar uma alternativa quando violada a restrição:

```
create table funcionario (  
    ...  
    cpf char(11) primary key,  
    cpf_supervisor char(11),  
    numero_departamento int not null,  
    foreign key (cpf_supervisor)  
        references funcionario(cpf)  
        ON DELETE SET NULL  
        ON UPDATE CASCADE  
    foreign key (numero_departamento)  
        references departamento(numero_departamento)  
        ON DELETE SET DEFAULT  
        ON UPDATE CASCADE  
);
```


➤ Referências:

- SILBERSCHATZ, A.; KORTH, F.; SUDARSHA, S. Database System Concepts. 6. ed. Nova York: MC Graw Hill, 2011.
- ELMASRI, R.; NAVATHE B. Sistemas de banco de dados. 6. Ed. São Paulo, SP: Pearson Addison-Wesley, 2011.
- RUIZ E. E.; Modelo de Dados Relacional. USP, São Paulo.