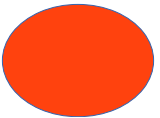


# Modelo Relacional

---

## Linguagens de Banco de Dados SQL - DDL

Prof. André Britto



# UNIQUE

---

- Especifica as chaves alternativas.
  - Podemos definir campos que não são a chave primária com valores únicos.
  - É uma restrição adiciona após o tipo do atributo.
  - Sintaxe
- ...

CONSTRAINT id UNIQUE (A1)

# UNIQUE

---

- A restrição de unicidade pode ser aplicada a mais de uma coluna:

```
CREATE TABLE example (
```

```
  a integer,
```

```
  b integer,
```

```
  c integer,
```

```
  UNIQUE (a, c)
```

```
);
```

# UNIQUE

---

- Vamos criar a seguinte tabela:
  - Tabela professor no schema universidade
  - Atributos:
    - mat\_professor: VARCHAR(7), não nulo
    - cpf: BIGINT, não nulo
    - cargo: INT
    - departamento: VARCHAR(5)
  - Restrições
    - Chave primária: mat\_professor
    - Chave estrangeira: cpf referencia a coluna cpf da tabela usuário
    - UNIQUE: cpf

# UNIQUE

---

```
CREATE TABLE universidade.professor(  
    mat_professor VARCHAR(7) NOT NULL,  
    cpf BIGINT NOT NULL,  
    cargo INT,  
    departamento VARCHAR(5),  
    CONSTRAINT pk_professor PRIMARY KEY(mat_professor),  
    CONSTRAINT fk_usuario FOREIGN KEY(cpf) REFERENCES  
    universidade.usuario(cpf),  
    CONSTRAINT uq_cpf UNIQUE(cpf)  
);
```

# UNIQUE

---

```
CREATE TABLE universidade.professor(  
    mat_professor VARCHAR(7) NOT NULL,  
    cpf BIGINT NOT NULL UNIQUE,  
    cargo INT,  
    departamento VARCHAR(5),  
    CONSTRAINT pk_professor PRIMARY KEY(mat_professor),  
    CONSTRAINT fk_usuario FOREIGN KEY(cpf) REFERENCES  
    universidade.usuario(cpf),  
);
```

# Default

---

- Quando não especificado, o valor default para um atributo é NULL.
  - Exceto campos NOT NULL ou chave primária.
- Podemos definir valores default diferente de null na especificação dos atributos.
- Sintaxes:  
A1 D1 [NOT NULL] [DEFAULT <valor>]

# Referential triggered action

---

- As modificações no banco de dados podem causar violações na integridade referencial.
- A ação padrão definida pela linguagem SQL é rejeitar a modificação caso isso ocorra.
- Podemos especificar ações alternativas quando ocorre essa violação.



# Referential triggered action

---

- Podemos efetuar essas ações na remoção e na atualização:
  - ON DELETE;
  - ON UPDATE;

# Referential triggered action

---

- Podemos efetuar as operações:
  - SET NULL;
    - O valor do atributo referenciado é alterado para NULL.
  - SET DEFAULT;
    - O valor do atributo referenciado é alterado para o valor default.
  - CASCADE.
    - A ação é feita em cascata, ou seja, todas as linhas que referenciam o valor modificado também são alterados.

# Referential triggered action

---

- Vamos criar a seguinte tabela:
  - Tabela departamento no schema universidade
  - Atributos:
    - cod\_depto: VARCHAR(5), não nulo
    - nome: VARCHAR, não nulo
    - chefe: VARCHAR(7)
    - orcamento: FLOAT
  - Restrições
    - Chave primária: cod\_depto
    - Chave estrangeira: chefe referencia mat\_professor da tabela professor
      - ON DELETE SET NULL
      - ON UPDATE CASCADE

# Referential triggered action

---

- ```
CREATE TABLE universidade.departamento(  
    cod_depto VARCHAR(5) NOT NULL,  
    nome VARCHAR NOT NULL,  
    chefe VARCHAR(11),  
    orcamento FLOAT,  
    CONSTRAINT pk_depto PRIMARY KEY (cod_depto),  
    CONSTRAINT fk_chefe FOREIGN KEY (chefe) REFERENCES  
    universidade.professor(mat_professor)  
    ON DELETE SET NULL ON UPDATE CASCADE  
  
);
```

# Criação de domínios

---

- Além de usar os tipos básicos do SQL, podemos criar domínios específicos.
- `CREATE DOMAIN universidade.tipo_matricula AS VARCHAR(7)`

# Criação de domínios

---

```
CREATE TABLE universidade.professor(  
    mat_professor tipo_matricula NOT NULL,  
    cpf BIGINT NOT NULL,  
    cargo INT,  
    departamento VARCHAR(5),  
    CONSTRAINT pk_professor PRIMARY KEY(mat_professor),  
    CONSTRAINT fk_usuario FOREIGN KEY(cpf) REFERENCES  
    universidade.usuario(cpf),  
    CONSTRAINT uq_cpf UNIQUE(cpf)  
);
```

# Check

---

- Um tipo de restrição que é usada para restringir um valor para um atributo.
- Sintaxe

```
CREATE TABLE R(  
    ...  
    CHECK(<predicado>  
)
```

# Check

---

- Vamos criar a seguinte tabela:

- Tabela disciplina no schema universidade

- Atributos:

- cod\_disc: VARCHAR(8), não nulo
    - nome: VARCHAR, não nulo
    - pre\_req VARCHAR(8)
    - creditos: SMALLINT
    - depto\_responsavel: VARCHAR(5)

- Restrições

- Chave primária: cod\_disc

CHECK:  $1 \leq \text{creditos}$  e  $\text{creditos} \leq 12$

- Chave estrangeira: depto\_responsavel referencia  
cod\_depto da tabela departamento
      - ON DELETE SET NULL
      - ON UPDATE CASCADE
    - pre\_req referencia cod\_disc da tabela disciplina
      - ON DELETE SET NULL
      - ON UPDATE CASCADE



# Check

- Vamos supor que os créditos de uma disciplina podem variar entre 1 e 12:

```
CREATE TABLE universidade.disciplina(  
    cod_disc VARCHAR(8),  
    nome VARCHAR NOT NULL,  
    pre_req VARCHAR(8),  
    credits SMALLINT,  
    depto_responsavel VARCHAR(5) NOT NULL,  
    CONSTRAINT pk_disciplina PRIMARY KEY(cod_disc),  
    CONSTRAINT ck_credits CHECK (1<=credits AND credits<=12),  
    CONSTRAINT fk_pre_req FOREIGN KEY(pre_req) REFERENCES universidade.disciplina(cod_disc)  
    ON DELETE SET NULL ON UPDATE CASCADE,  
    CONSTRAINT fk_depto FOREIGN KEY (depto_responsavel) REFERENCES universidade.departamento (cod_depto)  
    ON DELETE SET NULL ON UPDATE CASCADE  
);
```

# Check

---

```
CREATE TABLE universidade.disciplina(  
    cod_disc VARCHAR(8),  
    nome  VARCHAR NOT NULL,  
    pre_req VARCHAR(8),  
    creditos SMALLINT CONSTRAINT ck_creditos CHECK (1<=creditos AND creditos < 12),  
    depto_responsavel VARCHAR(5) NOT NULL,  
    CONSTRAINT pk_disciplina PRIMARY KEY(cod_disc),  
    CONSTRAINT ck_creditos CHECK (1<=creditos AND creditos<=12),  
    CONSTRAINT fk_pre_req FOREIGN KEY(pre_req) REFERENCES universidade.disciplina(cod_disc)  
    ON DELETE SET NULL ON UPDATE CASCADE,  
    CONSTRAINT fk_depto FOREIGN KEY (depto_responsavel) REFERENCES universidade.departamento (cod_depto)  
    ON DELETE SET NULL ON UPDATE CASCADE  
);
```

# Check

---

- A cláusula CHECK pode ser utilizada juntamente com a criação de um domínio.

```
CREATE DOMAIN tipo_creditos AS SMALLINT  
CHECK (tipo_creditos >= 1 AND tipo_creditos <= 12 )
```

# SQL – Comandos DDL

---

- Para remover uma tabela do banco de dados:
  - DROP TABLE R;
- Para adicionar os atributos de uma tabela:
  - ALTER TABLE R ADD A D;
- Para remover atributos de uma tabela:
  - ALTER TABLE R DROP A;

# SQL – Comandos DDL

---

- O comando de alteração é utilizado para alterar qualquer componente da tabela.
- Por exemplo,
  - Um professor precisa de um departamento e um departamento precisa de um professor como chefe.
  - Como podemos criar a chave estrangeira de uma tabela que ainda não existe?

# SQL – Comandos DDL

---

- Criamos as tabelas sem restrições e em seguida alteramos as tabelas.  

```
CREATE TABLE universidade.departamento(  
    cod_depto VARCHAR(5),  
    nome VARCHAR(50) NOT NULL,  
    chefe VARCHAR(7),  
    orcamento REAL CONSTRAINT ck_orcamento CHECK(orcamento > 0),  
    PRIMARY KEY(cod_depto)  
);
```

# SQL – Comandos DDL

---

- ALTER TABLE universidade.professor ADD  
CONSTRAINT fk\_alocacao FOREIGN KEY (departamento) REFERENCES  
universidade.departamento(cod\_depto)  
ON DELETE SET NULL ON UPDATE CASCADE;
- ALTER TABLE universidade.departamento or ADD  
CONSTRAINT fk\_chefe FOREIGN KEY (chefe) REFERENCES  
universidade.professor(mat\_professor)  
ON DELETE SET NULL ON UPDATE CASCADE;

# ALTER TABLE

---

- ALTER TABLE [nome da tabela] [ação]
- Ações
  - ADD
    - COLUMN
      - Nome, tipo e restrições
    - CONSTRAINT
      - Nome e restrição



# ALTER TABLE

---

- ALTER TABLE [nome da tabela] [ação]
- Ações
  - DROP
  - ALTER
    - Alterar tipo.
    - Adicionar ou remover valor DEFAULT
  - RENAME
    - Tabela, coluna

# ALTER TABLE

---

- ALTER TABLE universidade.departamento ADD CONSTRAINT orcamento\_valido CHECK (orcamento >= 0)
- ALTER TABLE universidade.departamento DROP CONSTRAINT orcamento\_valido
- ALTER TABLE universidade.professor ALTER COLUMN salario SET NOT NULL

## Leitura recomendada

---

ELMASRI, R; NAVATHE, S.B. **Sistemas de Banco de Dados**, Addison Wesley, 6ª Edição.

- Capítulo 4

Silberschatz, A; Korth H.F.; Sudarshan S. **Sistemas de Banco de Dados**, Editora Campus, 6ª Edição.

- Capítulo 3

