

Banco de Dados II

Data Definition Language - DDL (Table) e Constraints

Dr^a Alana Moraes

Aula Passada?

Interagimos com o PSQL

Interagimos com a sintaxe básica de interações com bancos de dados PostgreSQL

DDL - Linguagem de Definição de Dados

- O Comando **Create Table**:

CREATE TABLE <tabela>

(<descrição das **colunas+tipos**>)

(<descrição das **chaves/restrições**>);

- O Comando **Drop Table**

DROP TABLE <tabela>;

- O Comando **Truncate**

TRUNCATE TABLE <tabela>;

ID_EMP	NOME	SOBREN	RG	DT_NAS	ENDER	SEXO	SAL	DEPTO	ID_GER
17206-2	José	da Silva	24535357	25/02/1970	Rua Floriano Peixoto, 25	M	7000	1	
12584-7	Cecília	Ortiz Rezende	65784123	12/10/1980	Rua Marechal Deodoro da Fonseca, 335	F	3200	3	17206-2
16764-6	Pedro	Silvestre	24789635	15/05/1975	Rua Bela Cintra, 159	M	2800	4	17206-2
17987-5	Felipe	Gulhermino	25896314	19/08/1982	Rua Bolívia, 348	M	1800	3	12584-7
15698-3	Luciana	Feltosa	59874123	03/03/1980	Rua Presidente Dutra, 25	F	1500	4	16764-5
12357-1	Fabio	Santos Silva	25874137	29/07/1978	Rua Barão de Tefé, 30	M	1500	4	16764-5
16257-2	Elaine	Cristina	35789418	29/04/1974	Rua Itália, 258	F	2500	3	12584-7
15234-1	Cleiton	Fernandes	54698742	30/01/1979	Rua Andradina, 487	M	2200	3	12584-7

Tipo Serial

```
CREATE TABLE tablename (  
    colname SERIAL  
);
```

is equivalent to specifying:

```
CREATE SEQUENCE tablename_colname_seq;  
CREATE TABLE tablename (  
    colname integer NOT NULL DEFAULT nextval('tablename_colname_seq')  
);  
ALTER SEQUENCE tablename_colname_seq OWNED BY tablename.colname;
```

Exemplo

```
Create table teste2(  
    cod serial not null,  
    valor char(2));
```

```
Insert into teste2(valor) values('xx');
```

```
Insert into teste2(cod,valor) values (default, 'yy');
```

```
Insert into teste2 values (default, 'zz');
```

```
Select * from teste2;
```

ALTER TABLE

ALTER TABLE <tabela> <opção> <coluna>;

■ Principais opções:

- **ADD** - adiciona coluna
- **ADD CONSTRAINT** - adiciona restrição
- **MODIFY** - modifica definição de coluna
- **MODIFY CONSTRAINT** - modifica definição de restrição
- **DROP CONSTRAINT** - apaga restrição
- **DROP COLUMN** - apaga coluna
- **RENAME TO** - altera o nome da tabela
- **RENAME COLUMN** - altera nome de coluna
- **RENAME CONSTRAINT** - altera nome de constraint

ALTER TABLE - exemplos

ALTER TABLE Teste2 ADD telefone char(16);

ALTER TABLE Teste2 ADD fone char(16) default 'não listado';

Select * from teste2;

ALTER TABLE Teste2 DROP COLUMN telefone;

ALTER TABLE Teste2 alter column valor type varchar(30);

ALTER TABLE teste2 RENAME TO teste22;

ALTER TABLE Teste22 RENAME COLUMN valor TO descrição;

Select * from teste22;

Restrições (Constraints)

- Uma **restrição** ou **constraint** é um mecanismo que implementa **controle** para garantir a consistência dos dados (Em Modelos Relacionais)

Nome	Uso
NULL	Informa se o campo em questão pode ser nulo; caso contrário deve ser precedido de NOT.
UNIQUE	Indica que os valores na coluna ou conjunto de colunas não podem ser repetidos;
PRIMARY KEY	Identifica a chave primária da tabela;
FOREIGN KEY	Identifica uma chave estrangeira da tabela; implementada pela cláusula <i>references</i> .
CHECK	Determina uma regra de validação.

Constraint

[constraint <nome>] primary key | unique | not null | foreign key

- A *Constraint* pode ser nomeada
 - Caso não seja, o SGBD cria internamente
 - Exemplo:

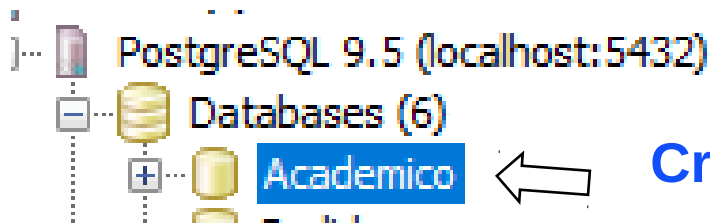
```
CREATE TABLE EMP (  
    EMPNO number constraint pk_emp primary key, . . . );
```

Opções

```
CREATE TABLE Produto (  
    CodPROD integer PRIMARY KEY,  
    Descricao Varchar(20),  
    Valor Numeric(10,2),  
    Unidade Char(2) );
```

```
CREATE TABLE Produto (  
    CodPROD integer,  
    Descricao Varchar(20),  
    Valor Numeric(10,2),  
    Unidade Char(2),  
    Constraint pk_prod PRIMARY KEY(codProd));
```

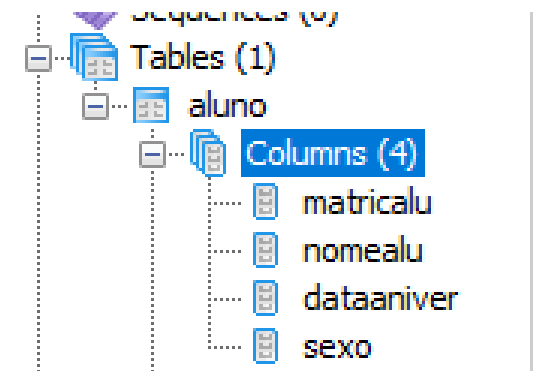
Exemplo



Criar novo banco: Academico

■ Tabela Aluno

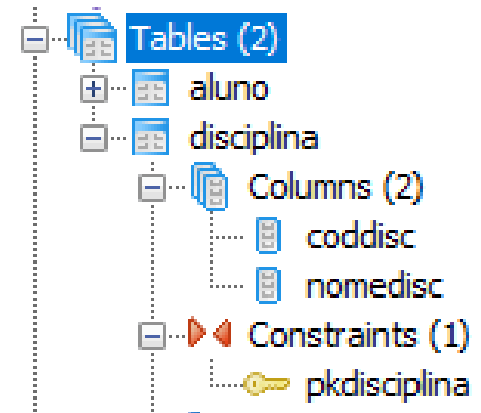
```
CREATE TABLE Aluno(  
    MatricAlu      serial NOT NULL,  
    NomeAlu        VARCHAR(40),  
    DataAniver     DATE default '01/01/1995',  
    Sexo            CHAR(1),  
    CONSTRAINT PKAluno PRIMARY KEY(MatricAlu) );
```



Exemplo

■ Tabela Disciplina

```
CREATE TABLE Disciplina (  
    CodDISC      serial NOT NULL,  
    NomeDISC     VARCHAR(30),  
    CONSTRAINT PKDisciplina PRIMARY KEY(CodDISC)  
);
```



**E se alguma coisa não for executada
ou sofrer alteração?**

ALTER TABLE para Constraints

ALTER TABLE <tabela> <opção> <coluna>;

- ADD CONSTRAINT - adiciona restrição
- MODIFY CONSTRAINT - modifica definição de restrição
- DROP CONSTRAINT - apaga restrição
- RENAME CONSTRAINT - troca o nome da restrição

Exemplo - Criação da Tabela Professor

```
CREATE TABLE Professor (  
    MatricProf          serial    NOT NULL,  
    NomeProf            VARCHAR(40),  
    DataAdmissao        DATE      );
```

```
ALTER TABLE Professor ADD CONSTRAINT PKProfessor  
PRIMARY KEY(MatricProf);
```

Ou

```
CREATE TABLE Professor (  
    Matricprof          serial    NOT NULL,  
    Nomeprof            VARCHAR(40),  
    Dataadmissao        DATE,  
    CONSTRAINT PKProfessor PRIMARY KEY(Matricprof));
```

Exemplos

- ALTER TABLE professor RENAME CONSTRAINT pkprofessor TO pkprof;
- ALTER TABLE professor DROP PRIMARY KEY;
ou
- ALTER TABLE professor DROP CONSTRAINT pkprof;

Relacionando Disciplina e Professor

Tabela: **Disciplina**

CodDisc	NomeDisc
1	Banco de Dados II
2	Sistemas Operacionais
3	POO

Tabela: **Professor**

Matric	Nome	DataAdm	CodDisc
1	Crishane	12/12/00	1
2	Damires	12/12/00	1
3	Luciana	12/12/00	2
4	Fausto	12/12/00	3

Integridade Referencial

- **Garante a não corrupção dos dados, de modo a não permitir um registro "filho" sem um registro "pai"**
- Define um conjunto de regras possíveis:
 - **RESTRICT/NO ACTION**
 - Desabilita o **delete/update** no dado referenciado (PKs não podem ser atualizadas ou excluídas)
 - **SET TO NULL:**
 - Quando o dado referenciado (PK) é **atualizado/excluído**, todos os dados dependentes associados são setados pra **NULL**.

Integridade Referencial

- SET TO DEFAULT:

- Quando o dado referenciado (PK) é **atualizado/excluído**, todos os dados dependentes associados são setados pra um valor **DEFAULT**.

- CASCADE:

- Quando o dado referenciado (PK) é **atualizado**, todos os dados dependentes associados são **atualizados**
- Quando uma linha referenciada (PK) é **deletada**, todas as linhas dependentes associadas são **deletadas**.

Vamos testar...

1. NO ACTION (default)

- As chaves especificadas **não** podem ser atualizadas ou excluídas

2. DELETE SET NULL

- Indica que, quando uma linha que contém PK for excluída, todas as linhas (FK) da tabela filha terão seu **valor ajustado para null**

3. DELETE CASCADE

- Quando uma linha contendo chave especificada for excluída, todas as linhas da tabela filha também serão;

4. UPDATE CASCADE

- Quando uma linha contendo chave especificada for atualizada, todas as linhas da tabela filha serão.

Exemplo – Integridade Referencial nº 1

```
ALTER TABLE professor add codDisc integer;
```

```
ALTER TABLE professor ADD CONSTRAINT FKprofdisc  
FOREIGN KEY(codDisc) REFERENCES Disciplina;
```

```
Insert into disciplina(nomeDisc) values ('BDII');
```

```
Insert into disciplina(nomeDisc) values ('Sistemas  
Operacionais');
```

```
Select * from disciplina;
```

Exemplo - Integridade Referencial nº 1

```
Insert into professor(nomeProf, dataadmissao,coddisc)  
values('Alana', null, 1);
```

```
Insert into professor(nomeProf, dataadmissao,coddisc)  
values('Damires', null, 1);
```

```
Insert into professor(nomeProf, dataadmissao,coddisc)  
values('Alex', null, 2);
```

```
Select * from professor;
```

```
Delete from disciplina where codDisc = 1;
```

- O que aconteceu???

Exemplo - Integridade Referencial nº 2

Alter table professor drop constraint fkprofdisc;

```
ALTER TABLE professor ADD CONSTRAINT  
FK2_prof_disc FOREIGN KEY(codDisc)  
REFERENCES Disciplina on delete set null;
```

Select * from disciplina;

Select * from professor;

Delete from disciplina where codDisc = 1;

O que aconteceu???

Select * from disciplina;

Select * from professor;

Exemplo - Integridade Referencial nº 3

```
Alter table professor drop constraint fk2_prof_disc;  
ALTER TABLE professor ADD CONSTRAINT  
FK3_prof_disc FOREIGN KEY(coddisc)  
REFERENCES Disciplina ON DELETE CASCADE;
```

```
Select * from disciplina;
```

```
Select * from professor;
```

Delete from disciplina where coddisc = 2;

O que aconteceu???

```
Select * from disciplina;
```

```
Select * from professor;
```


Exemplo - Integridade Referencial nº 4

Alter table professor drop constraint fk3_prof_disc;

```
ALTER TABLE professor ADD CONSTRAINT  
FK4_prof_disc FOREIGN KEY(coddisc) REFERENCES  
Disciplina ON UPDATE CASCADE;
```

**** Inserir/atualizar dados**

Insert into disciplina(coddisc,nomeDisc) values (1,'BD');

insert into disciplina(coddisc,nomeDisc) values (2,'Sistemas Operacionais');

Update professor set coddisc = 1 where nomeprof = 'Crishane' or nomeprof = 'Damires';

Insert into professor values(3,'Luciana', null, 2);

Exemplo – Integridade Referencial nº 4

Select * from disciplina;

Select * from professor;

Update disciplina

Set coddisc = 11

where coddisc = 1;

O que aconteceu???

Select * from disciplina;

Select * from professor;

Voltando Disciplina ao valor correto ...

```
Update disciplina  
Set coddisc = 1  
Where coddisc = 11;
```

```
Select * from disciplina;  
Select * from professor;
```

Outro Relacionamento

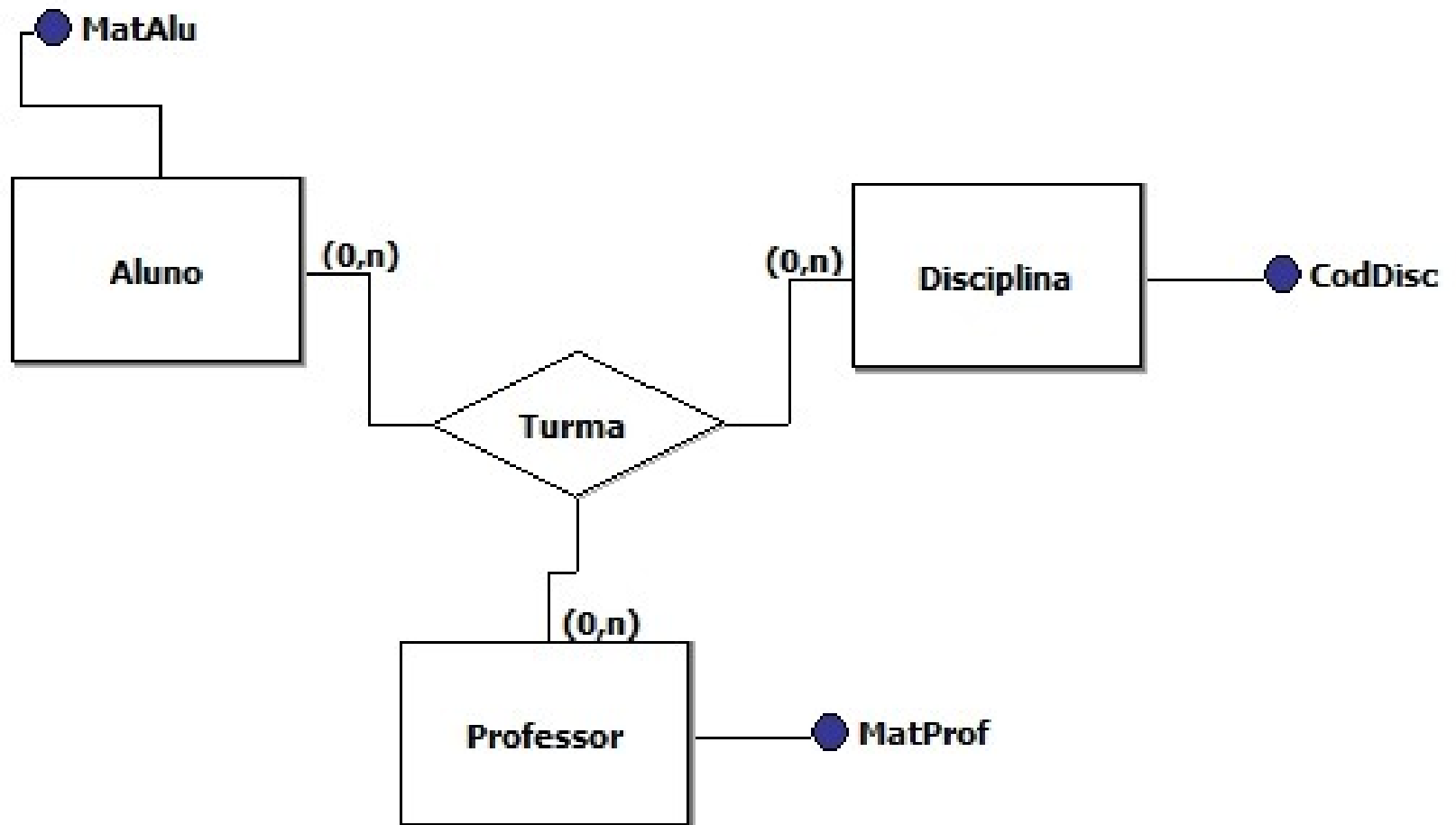


Tabela Turma

```
CREATE TABLE turma (  
    MatricAlu integer NOT NULL,  
    MatricProf integer NOT NULL,  
    CodDISC integer NOT NULL);
```

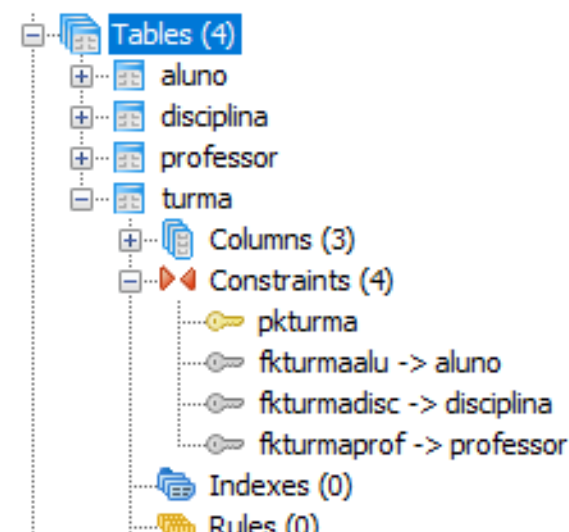
```
ALTER TABLE turma ADD CONSTRAINT PKturma  
PRIMARY KEY(MatricAlu,MatricProf,CodDISC);
```

Chaves estrangeiras

```
ALTER TABLE turma ADD  
CONSTRAINT FKturmaAlu  
FOREIGN KEY(MatricAlu)  
REFERENCES Aluno;
```

```
ALTER TABLE turma ADD  
CONSTRAINT FKturmaDisc  
FOREIGN KEY(CodDISC)  
REFERENCES Disciplina;
```

```
ALTER TABLE turma ADD  
CONSTRAINT FKturmaProf  
FOREIGN KEY(MatricProf)  
REFERENCES Professor;
```



Turma

```
Insert into aluno values(default,'X','12/12/80','M');  
Select * from aluno;
```

```
Insert into turma values(1,1,1);
```

```
Select * from turma;
```

matricalu integer	matricprof integer	coddisc integer
1	1	1

Eliminando tabela com dependências

DROP TABLE disciplina cascade constraints;

Check Constraint

- Uma **check constraint** permite a especificação de condições sobre cada linha da tabela, para **validação de dados**
 - Deve se referir a colunas naquela tabela
- Uma **check constraint** vai ser definida por meio dos comandos **CREATE TABLE** ou **ALTER TABLE**

Check Constraint

- Utilizando o comando Create Table

```
CREATE TABLE table_name  
  (column1 datatype null/not null,  
   column2 datatype null/not null,  
   ...  
   CONSTRAINT constraint_name CHECK  
   (column_name condition) [DISABLE]  
  );
```

Check Constraint - Exemplo

```
CREATE TABLE fornecedor (  
    id integer,  
    nome varchar(40),  
    CONSTRAINT checkFornecedor CHECK (id  
    BETWEEN 100 and 9999)  
);
```

Agora teste: **Insert into fornecedor values(1, 'XXX');**

Como corrigir??

Exemplos

```
CREATE TABLE products  
  (product_no integer,  
   name varchar(20), price numeric CHECK (price > 0)  
  );
```

```
ALTER TABLE professor ADD salario numeric(15,2) CHECK  
(salario >= 2000.00 and salario <= 10000.00);
```

Backup e Restore

**Cria o BD
Academico
e Restaura**

The screenshot illustrates the steps to create and restore a database in PostgreSQL 9.5. On the left, the 'Databases (11)' tree shows a context menu for the 'Academico' database. The 'Maintenance...' option is highlighted with a red rectangle. In the center, a second context menu is shown for the 'testeadad' database, with the 'Restore...' option highlighted in blue. On the right, the SQL editor displays the SQL script for creating the 'testeadad' database with specific encoding and collation settings.

PostgreSQL 9.5 (localhost:5432)

Databases (11)

- Academico
- Emprega
- Filmes
- Pedidos
- Pedidos2
- Teste
- Teste2
- alunosG
- postgis_
- postgres
- testeaca
- Cata
- Ever
- Ext
- Sche

Context menu for 'Academico':

- Refresh
- New Object >
- Disconnect database
- Delete/Drop...
- Search objects...
- CREATE Script
- Reports
- Maintenance...**
- Backup...
- Restore...
- Properties...

Context menu for 'testeadad':

- Refresh
- New Object >
- Disconnect database
- Delete/Drop...
- Search objects...
- CREATE Script
- Reports >
- Maintenance...
- Backup...
- Restore...**
- Properties...

SQL Editor (Default table ACL):

```
-- Database: testeadad
-- DROP DATABASE testeadad;

CREATE DATABASE testeadad
WITH OWNER = postgres
ENCODING = 'UTF8'
TABLESPACE = pg_default
LC_COLLATE = 'Portuguese_Brazil.1252'
LC_CTYPE = 'Portuguese_Brazil.1252'
CONNECTION LIMIT = -1;
```

Dúvidas?

Alanamm.prof@gmail.com