Constraints

DEFAULT – importante para campos que não tem valor obrigatório entrado pelo usuário

Exemplo, campos numéricos devem ter valor default 0 ou outro adequado. Ajuda, pois as variáveis já são inicializadas. Caso contrário o valor default será NULL, o que pode complicar em expressões com ele.

data date DEFAUL current_date;

NOT NULL – Importante para campos requeridos. Mas lembre que um simples espaço em branco já satisfaz. Para dados importantes, fazer outras críticas via programação ou via constraint CHECK.

CHECK – Esta checa se o campo satisfaz um valor ou expressão.

```
CREATE TABLE produtos (
produto_no integer,
descricao text,
desconto numeric CHECK (desconto > 0 AND desconto < 0.10),
preco numeric CONSTRAINT preco_positivo CHECK (preco > 0),
check (preco > desconto)
);
```

Simulando ENUM no PostgreSQL COM CHECK

Para simular a constraint enum do MySQL, podemos usar a constraint check.
Dica do site "PostgreSQL & PHP Tutorials".

```
CREATE TABLE pessoa(
codigo int null primary key,
cor_favorita varchar(255) not null,
check (cor_favorita IN ('vermelha', 'verde', 'azul'))
);
```

```
INSERT INTO pessoa (codigo, cor_favorita) values (1, 'vermelha'); -- OK
INSERT INTO pessoa (codigo, cor_favorita) values (1, 'amarela'); -- Erro, amarelo não consta
```

A versão 8.3 do PostgreSQL já traz o tipo ENUM.

UNIQUE – exige que o campo tenha valor exclusivo para todos os registros.

```
CREATE TABLE produtos (
cod_prod integer UNIQUE,
nome text,
preco numeric
);
```

```
CREATE TABLE produtos (
cod_prod integer,
nome text,
preco numeric,
UNIQUE (cod_prod)
);
```

```
CREATE TABLE exemplo (
a integer,
b integer,
c integer,
UNIQUE (a, c)
);
```

CHAVE PRIMÁRIA (PRIMARY KEY) – Exige que o campo tenha valores exclusivos em todos os registros e também que não seja NULL, ou seja, uma chave primária é a combinação das constraints NOT NULL e UNIQUE.

```
CREATE TABLE produtos (
cod_prod integer PRIMARY KEY,
nome text,
preco numeric
);
```

Composta (formada por mais de um campo)

```
CREATE TABLE exemplo (
a integer,
b integer,
c integer,
PRIMARY KEY (a, c)
);
```

CHAVE ESTRANGEIRA (FOREIGN KEY) - Criadas com o objetivo de relacionar duas tabelas, mantendo a integridade referencial entre ambas.

Tabela principal
```
CREATE TABLE produtos (
cod_prod integer PRIMARY KEY,
nome text,
preco numeric
);
```

Tabela referenciada
```
CREATE TABLE pedidos (
cod_pedido integer PRIMARY KEY,
cod_prod integer,
quantidade integer,
CONSTRAINT pedidos_fk FOREIGN KEY (cod_prod) REFERENCES produtos (cod_prod)
);
```

Selecionando o Campo para a Chave Primária
A chave primária é o campo ou campos que identificam de forma exclusiva cada registro.
• Não é permitido valores nulos na chave nem duplicados
• Caso a tabela não tenha um campo que a identifique, pode-se usar um campo que numere os registros seqüencialmente

Dica de Desempenho: O tamanho da chave primária afeta o desempenho das operações, portanto usar o menor tamanho que possa acomodar os dados do campo.

Uma chave primária pode ser formada por mais de um campo, quando um único campo não é capaz de caracterizar a tabela.

Cada tabela somente pode conter uma única chave primária.
Chave Primária Composta (dois campos)
```
CREATE TABLE tabela (
codigo INTEGER,
data DATE,
nome VARCHAR(40),
PRIMARY KEY (codigo, data)
);
```

Nomes de Campos com espaço ou acento
Devem vir entre aspas duplas.