

Constraints

Constraints (restrições) são regras que podem ser aplicadas a colunas ou tabelas inteiras em um banco de dados para garantir a integridade e consistência dos dados. As constraints podem ser aplicadas para impor limites, restrições ou regras que devem ser respeitadas para que os dados sejam armazenados no banco.

Imagine que as constraints são como regras em um jogo de tabuleiro. Cada regra especifica o que pode e o que não pode ser feito durante o jogo. Por exemplo, em um jogo de xadrez, existem regras que especificam como as peças podem se mover no tabuleiro. Se um jogador tentar mover uma peça de uma maneira que não está de acordo com as regras, a jogada será inválida e ele terá que voltar atrás. Da mesma forma, se você tentar inserir um valor que não atenda a uma constraint em uma coluna de um banco de dados, a inserção será inválida e você terá que corrigir o valor ou a constraint para continuar.

CHECK

A constraint CHECK é usada para impor uma condição específica em uma coluna. Por exemplo, podemos definir uma constraint CHECK para garantir que a coluna "idade" de uma tabela de usuários tenha valores entre 18 e 99 anos. Veja:

```
CREATE TABLE usuarios (  
    id SERIAL PRIMARY KEY,  
    nome VARCHAR(50),  
    idade INTEGER,  
    CONSTRAINT idade_maior CHECK (idade >= 18 AND idade <= 99)  
);
```

Isso garante que caso seja inserido um valor não previsto pela constraint, o banco de dados irá gerar um erro e o registro não será inserido.

NOT NULL

A constraint NOT NULL é usada para especificar que uma coluna não pode ter valores nulos. Por exemplo, podemos definir uma constraint NOT NULL para garantir que a coluna "email" de uma tabela de usuários não seja nula.

```
CREATE TABLE usuarios (  
    id SERIAL PRIMARY KEY,
```

```
    nome VARCHAR(50),  
    email VARCHAR(50) NOT NULL  
);
```

Isso garante que caso seja inserido um valor nulo na coluna email, o banco de dados irá gerar um erro e o registro não será inserido.

UNIQUE

A constraint UNIQUE é usada para garantir que os valores em uma coluna sejam exclusivos. Por exemplo, podemos definir uma constraint UNIQUE para garantir que a coluna "email" de uma tabela de usuários tenha valores únicos.

```
CREATE TABLE usuarios (  
    id SERIAL PRIMARY KEY,  
    nome VARCHAR(50),  
    email VARCHAR(50) UNIQUE  
);
```

Isso garante que caso seja inserido um valor na coluna email que já exista em outro registro, o banco de dados irá gerar um erro e o registro não será inserido.

Referências e materiais complementares

<https://www.postgresql.org/docs/current/ddl-constraints.html>