



WEBAULA 18

- Criação e Gerenciamento de Restrições
- Referências

QUESTION BANKS

Criação e Gerenciamento de Restrições

Na seção anterior você aprendeu a criar e manipular tabelas sem a utilização de restrições. Agora vamos continuar com a criação e manipulação de tabelas, porém com a utilização de restrições.

As restrições são regras apresentadas ao SGBD relacional para que ele possa garantir a integridade e a consistência dos dados manipulados pelo usuário (CARDOSO, 2009). Tais restrições podem sofrer alterações sem que as modificações realizadas afetem os dados já existentes. Porém, é fundamental um bom entendimento e compreensão do impacto delas no conjunto de dados para não gerar severas consequências na consistência dos dados já armazenados e que ainda serão mantidos pelo SGBD.

Assim, podemos concluir que as restrições são regras impostas em uma tabela ou coluna. Os tipos de restrições válidas nos SGBDs relacionais são:

- NOT NULL
- UNIQUE KEY
- PRIMARY KEY
- FORFIGN KFY
- CHECK

Dica



(i) As restrições são conhecidas nos SGBDs como constraints. Mais adiante iremos apresentar individualmente mais detalhes sobre cada tipo de restrição e sua aplicação.

A definição de uma restrição pode ser classificada como em nível de coluna ou em nível de tabela. Dizemos que ela é em nível de coluna quando sua indicação acompanha a definição da coluna. De igual forma, dizemos que é em nível de tabela quando sua indicação não está associada diretamente à definição da coluna, e sim complementarmente à definição da(s) coluna(s).

As restrições podem ser definidas no momento da criação da tabela. O Quadro 16 ilustra a sintaxe do comando CREATE TABLE, onde podemos observar a sintaxe de uma restrição em nível de coluna e uma restrição em nível de tabela.

Quadro 16 - Sintaxe de definição de restrição no momento da criação da tabela

```
CREATE TABLE nome_tabela
  coluna1 tipo_de_dado restrição_de_coluna,
  coluna2 tipo_de_dado,
  coluna3 tipo_de_dado,
 restrição_de_tabela
);
```

Considere o cenário no qual a tabela já existe e se faz necessária a inclusão de uma restrição. Neste caso, devemos utilizar o comando ALTER TABLE, conforme ilustrado no Quadro 17.

Quadro 17 – Sintaxe de definição de restrição com a tabela existente

Agora que você já conhece as possibilidades que temos para definir uma restrição, passaremos para a descrição de cada uma das cinco restrições conhecidas nos SGBDs relacionais.

A primeira restrição é a NOT NULL. Esta restrição faz com que valores nulos não sejam permitidos em determinada coluna, isto é, quando uma coluna apresentar esta restrição, obrigatoriamente um valor deverá ser informado no momento da inserção de um registro. Observe na Figura 40 um cenário no qual a tabela EMP apresenta a restrição NOT NULL em duas colunas (ename e deptno).

EMP

EMPNO	ENAME	JOB		сомм	DEPTNO
7839	KING	PRESIDENT			10
7698	BLAKE	MANAGER			30
7782	CLARK	MANAGER			10
7566	JONES	MANAGER			20

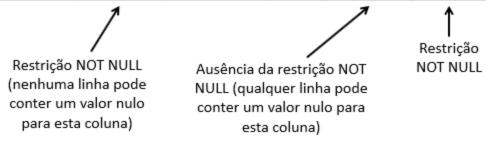


Figura 40

(i) Reflexão

Talvez você não se recorde, mas quando apresentamos aspectos sobre a Modelagem de Dados (Ciclo 2) falamos no conceito de obrigatoriedade do relacionamento. Pois bem, o uso da restrição NOT NULL é o efeito na prática da presença de uma obrigatoriedade na relação entre duas tabelas. Mais adiante falaremos sobre chave

primária e chave estrangeria e aos poucos você irá percebendo os aspectos conceituais sendo aplicados na prática.

Na prática, a restrição NOT NULL força o preenchimento de uma determinada coluna. No exemplo acima, a coluna "ename" apresenta restrição, o que faz com que todas as linhas devam apresentar um valor para a coluna, exatamente como ocorre também na coluna "deptno". Já a coluna "comm", que indica um valor de comissão para o empregado, não apresenta a restrição NOT NULL e, portanto, poderá não apresentar valores para ela.

Normalmente, a restrição NOT NULL é definida em nível de coluna, pois no momento de criação da coluna já se tem uma leitura preliminar da necessidade do seu preenchimento. O Quadro 18 apresenta um exemplo de sentença SQL na qual ocorre a criação da tabela EMP.

Quadro 18 – Exemplo de sentença de criação de tabela com a restrição NOT NULL

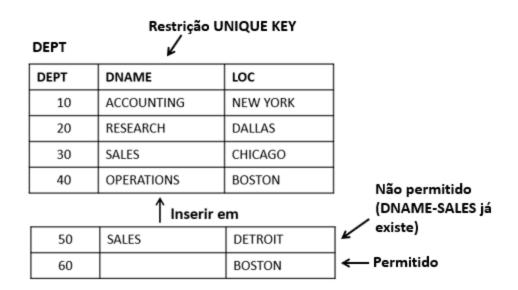
```
CREATE TABLE emp
(
   empno INTEGER,
   ename VARCHAR (10) NOT NULL,
   job VARCHAR (9),
   mgr INTEGER,
   hiredate DATE,
   sal FLOAT (7,2),
   comm FLOAT (7,2),
   deptno INTEGER NOT NULL
);
```

No entanto, caso a tabela já exista e seja necessária a indicação de uma obrigatoriedade de coluna, isto pode ser feito através do comando ALTER TABLE em conjunto com a cláusula ALTER COLUMN ou MODIFY, como visto na seção anterior. O Quadro 19 ilustra dois exemplos onde é realizada a alteração da coluna "job", indicando que seu preenchimento é obrigatório.

Quadro 19 – Exemplo de sentença de alteração de coluna com a restrição NOT NULL

-- sintaxe no SGBD MySQL
ALTER TABLE emp
ALTER COLUMN (job NOT NULL);
OU
-- sintaxe no SGBD Oracle
ALTER TABLE emp
MODIFY (job NOT NULL);

Há situações nas quais desejamos garantir que o valor informado para uma determinada coluna, ou mesmo uma combinação de colunas, deva ser único. Por exemplo, considere o cadastro de pessoa física, onde a coluna número do registro geral (RG) não deve ser repetido para diferentes linhas (registros). Neste caso, os SGBDs relacionais apresentam uma restrição conhecida como UNIQUE. Esta restrição não permite valores repetidos. Observe na Figura 41 um cenário no qual a tabela DEPT apresenta a restrição UNIQUE na coluna "dname".



No exemplo acima vemos a aplicação da restrição UNIQUE KEY na coluna "dname". Esta restrição impede que valores idênticos sejam inseridos.



Atenção para alguns SGBDs relacionais que podem apresentar o contexto conhecido como case sensitivity, o que torna diferente o conteúdo com variantes em maiúsculas e minúsculas. Da mesma forma, eles podem aceitar várias linhas com valor para a coluna correspondente a nulo (null), enquanto outros, apenas uma linha (registro) com indicação de nulidade para a referida coluna.

A restrição UNIQUE pode ser definida tanto em nível de tabela quanto em nível de coluna. O Quadro 20 apresenta dois exemplos de utilização da restrição UNIQUE. O primeiro ilustra a sentença para a inclusão da restrição em nível de coluna, enquanto o segundo em nível de tabela.

Quadro 20 - Exemplo de criação de coluna com a restrição UNIQUE

```
CREATE TABLE dept
  deptno INTEGER,
  dname VARCHAR (14) UNIQUE,
  loc VARCHAR (13)
);
OU
CREATE TABLE dept
```

```
deptno INTEGER,
 dname VARCHAR (14),
 loc VARCHAR (13),
 CONSTRAINT dept_name_uk UNIQUE (dname)
);
```

Acima é possível observar que no segundo exemplo (onde a restrição é definida em nível de tabela) há a presença da palavra reservada CONSTRAINT, seguida de um identificador. Este identificador (dept_name_uk) é o nome atribuído para a restrição.



(i) Dica

A utilização da palavra "reserva" seguida da identificação, como no exemplo apresentado, não é obrigatória. No entanto, no momento em que é feita a identificação (nome) da restrição, que no exemplo é composta pelo nome da tabela e nome da coluna seguida da abreviação "uk", amplia-se o horizonte da legibilidade do código escrito, o que facilita significativamente a interpretação tanto da sentença como de possíveis erros gerados pelo SGBD relacional.

No contexto onde a tabela já tenha sido criada, se faz necessária a inclusão da restrição através do comando ALTER TABLE em conjunto com a cláusula ADD. O Quadro 21 apresenta o exemplo de utilização da restrição UNIQUE.

Quadro 21 - Exemplo de sentença de alteração de coluna com a restrição UNIQUE

ALTER TABLE dept ADD CONSTRAINT dept_name_uk **UNIQUE** (dname);

(i) Dica

No caso da inclusão de uma restrição UNIQUE em uma coluna já existente, esta só será aceita pelo SGBD relacional se todas as linhas já inseridas na tabela satisfizerem a condição de unicidade imposta na restrição.

A próxima restrição não é largamente utilizada, porém pode ser de grande utilidade. A restrição CHECK define uma condição que cada linha deve satisfazer. Uma única coluna pode ter várias restrições CHECK que fazem referência à coluna na sua definição. Não há limite quanto ao número de restrições CHECK que você pode definir em uma coluna. Da mesma forma como nas demais restrições já apresentadas, a restrição CHECK pode ser definida no nível da coluna ou no nível da tabela. O Quadro 22 apresenta dois exemplos de utilização de uma restrição CHECK para a coluna "deptno".

Quadro 22 – Exemplo de sentença de criação de coluna com a restrição CHECK

```
CREATE TABLE dept
  deptno INTEGER CHECK (deptno BETWEEN 10 AND 99),
  dname VARCHAR (14),
  loc VARCHAR (13)
);
OU
CREATE TABLE dept
  deptno INTEGER,
  dname VARCHAR (14),
  loc VARCHAR (13),
  CONSTRAINT emp_deptno_ck
    CHECK (DEPTNO BETWEEN 10 AND 99)
);
```

Acima, no Quadro 22, é possível observar os dois exemplos de utilização da restrição CHECK. O primeiro ilustra a sentença para a inclusão da restrição em nível de coluna, enquanto o segundo em nível de tabela. Em ambos, a restrição indica que os valores que serão aceitos pelo SGBD relacional para a coluna "deptno" deverão estar na faixa entre 10 e 99, inclusive as extremidades. Igualmente, aqui vemos a presença da palavra reservada CONSTRAINT seguida de um identificador, cuja explicação já demos anteriormente.

De igual forma, no contexto onde a tabela já tenha sido criada, se faz necessária a inclusão da restrição através do comando ALTER TABLE em conjunto com a cláusula ADD CONSTRAINT. O Quadro 23 apresenta o exemplo de utilização da restrição CHECK em uma coluna já existente em uma tabela.

Quadro 23 – Exemplo de sentença de alteração de coluna com a restrição CHECK

ALTER TABLE dept

ADD CONSTRAINT emp_deptno_ck

CHECK (DEPTNO BETWEEN 10 AND 99);



No caso da inclusão de uma restrição CHECK em uma coluna já existente, esta só será aceita pelo SGBD relacional se todas as linhas já inseridas na tabela satisfizerem a condição de checagem imposta na expressão (deptno between 10 and 99) apresentada

para a restrição.

As próximas duas restrições são fundamentais para que o SGBD relacional possa garantir a integridade referencial do banco de dados. São elas: PRIMARY KEY e FOREIGN KEY.

A restrição PRIMARY KEY é uma coluna ou conjunto de colunas que identifica exclusivamente cada linha da tabela. Essa restrição impõe a unicidade de valores e assegura que nenhuma coluna que seja parte da chave primária possa conter um valor nulo.

Observe na Figura 42 um cenário no qual a tabela DEPT esteja recebendo linhas de entrada do cadastro de um usuário.

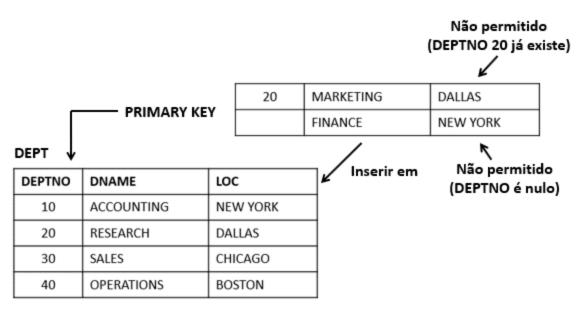


Figura 42

No exemplo acima é possível observar a tentativa de inserção de uma linha onde o valor da coluna "deptno" (20) já existe na tabela e, portanto, não será permitido. Da mesma forma, o SGBD relacional não irá permitir a inclusão de uma linha onde o valor da "deptno" for nulo (*null*).



Somente uma chave primária pode ser criada para cada tabela. Na prática, podemos realizar uma analogia a uma PRIMARY KEY como sendo uma combinação de uma restrição UNIQUE e uma restrição NOT NULL juntas.

O Quadro 24 apresenta dois exemplos de utilização da restrição PRIMARY KEY. O primeiro ilustra a sentença para a inclusão da restrição em nível de coluna, enquanto o segundo, em nível de tabela.

Quadro 24 – Exemplo de sentença de criação de coluna com a restrição PRIMARY KEY

```
CREATE TABLE dept
(
deptno INTEGER PRIMARY KEY,
dname VARCHAR (14),
loc VARCHAR (13)
);

OU

CREATE TABLE dept
(
deptno INTEGER,
dname VARCHAR (14),
loc VARCHAR (13),
CONSTRAINT emp_deptno_pk
PRIMARY KEY (deptno)
);
```

Em existindo a tabela, faz-se necessária a inclusão da restrição através do comando ALTER TABLE em conjunto com a cláusula ADD CONSTRAINT, como já visto anteriormente. O Quadro 25 apresenta o exemplo de utilização da restrição PRIMARY KEY em uma coluna de uma tabela já existente.

Quadro 25 – Exemplo de sentença para alteração de coluna com a restrição PRIMARY KEY

ALTER TABLE dept ADD CONSTRAINT dept_deptno_pk PRIMARY KEY (deptno);

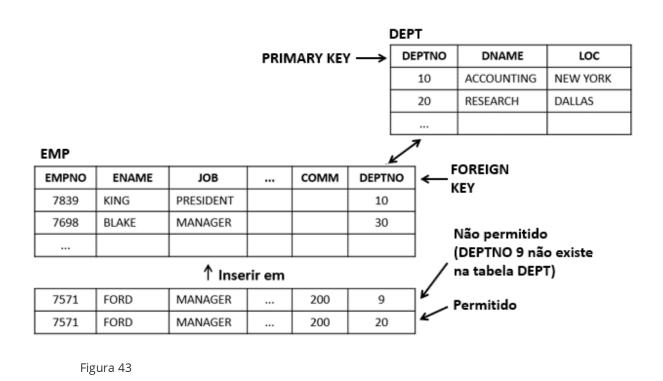


(i) Dica

Da mesma forma como na utilização da restrição CHECK, no caso da inclusão de uma restrição PRIMARY KEY em uma coluna, ou combinação de colunas já existentes, esta só será aceita pelo SGBD relacional se todas as linhas já inseridas na tabela satisfizerem a condição imposta na restrição.

Por último, mas não menos importante, temos a restrição da chave estrangeira. A FOREIGN KEY designa uma coluna ou combinação de colunas com a chave estrangeira estabelecendo um relacionamento entre tabelas. Ao estabelecer a relação entre as tabelas, a restrição FOREIGN KEY impõe a regra de integridade referencial, não permitindo que dados se tornem órfãos na medida em que linhas da tabela de origem sofrem manipulações. Da mesma forma, não permite que valores para a coluna que referencia sejam diferentes dos da coluna referenciada.

A Figura 43 apresenta um cenário no qual a tabela DEPT apresenta uma coluna (deptno) PRIMARY KEY que é utilizada na tabela EMP como sendo uma FOREIGN KEY para esta tabela DEPT. A coluna (deptno) na tabela EMP representa esta chave estrangeira, não por acaso, com o mesmo nome para facilitar a legibilidade da estrutura.



No exemplo acima há a tentativa de inserir valor (9) para a coluna "deptno" na tabela EMP, onde o valor (9) não existe previamente na tabela DEPT. Esta inclusão não será permitida. Já na ação de inclusão de uma linha onde o valor (20) da coluna "deptno" existe na tabela DEPT (de origem), a ação é bemsucedida.

A restrição FOREIGN KEY pode ser definida tanto em nível de tabela quanto em nível de coluna. O Quadro 26 apresenta dois exemplos de utilização de uma restrição FOREIGN KEY para a coluna "deptno". No primeiro temos a sentença para a inclusão da restrição em nível de coluna; no segundo, em nível de tabela.

Quadro 26 – Exemplo de sentença para criação de coluna com a restrição FOREIGN KEY

```
CREATE TABLE emp
```

```
empno INTEGER NOT NULL,
  ename VARCHAR (10),
 job VARCHAR (9),}
  mgr INTEGER,
  hiredate DATE,
  sal FLOAT (7,2),
  comm FLOAT (7,2),
  deptno INTEGER REFERENCES dept (deptno)
);
CREATE TABLE emp
  empno INTEGER NOT NULL,
  ename VARCHAR (10),
 job VARCHAR (9),}
  mgr INTEGER,
  hiredate DATE,
  sal FLOAT (7,2),
  comm FLOAT (7,2),
  deptno INTEGER,
  CONSTRAINT emp_deptno_fk FOREIGN KEY (deptno)
    REFERENCES dept (deptno)
);
```

(i) Dica

O fato de utilizarmos o mesmo nome para a coluna referenciada e a coluna de referência (deptno) obviamente não garante nenhuma relação de integridade referencial. Assim, podemos afirmar que qualquer conjunto de nomes poderia ser utilizado. No entanto, reforçamos a importância de seguir um padrão de nomenclatura e utilização de identificadores para favorecer a legibilidade do modelo.

No quadro acima pode-se observar que na primeira sentença SQL é utilizada apenas a palavra reservada REFERENCES para indicar a tabela (DEPT) e a coluna (deptno) que será referenciada. Já no segundo exemplo é utilizada a palavra reservada CONSTRAINT, seguida de um identificador para o nome da restrição, a expressão FOREIGN KEY com a indicação do nome da coluna na tabela criada, e só aí a palavra reservada REFERENCES para indicar a tabela (DEPT) e a coluna (deptno) que está sendo referenciada.

Caso a tabela já exista, de igual forma como nas restrições anteriores, você pode definir a chave estrangeira utilizando o comando ALTER TABLE com a cláusula ADD CONSTRAINT. Veja no Quadro 27 o exemplo de utilização da restrição FOREIGN KEY em uma coluna de uma tabela já existente na tabela EMP.

Quadro 27 – Exemplo de sentença para alteração de coluna com a restrição FOREIGN KEY

ALTER TABLE emp

ADD CONSTRAINT emp_deptno_fk

FOREIGN KEY (deptno) REFERENCES dept (deptno);



Novamente reforçamos a informação de que, no caso da inclusão de uma restrição FOREIGN KEY em um coluna, ou combinação de colunas já existentes, esta só será aceita pelo SGBD relacional se todas as linhas já inseridas na tabela satisfizerem a condição imposta na restrição, ou seja, os valores previamente armazenados estejam de acordo com os valores armazenados na coluna da tabela referenciada.

Portanto, após a exposição e exemplos utilizados para demonstrar o uso da restrição de chave estrangeira, podemos afirmar que ela é definida através da combinação das seguintes palavras-chave:

- FOREIGN KEY é usada para definir a coluna na tabela filha no nível de restrição da tabela;
- REFERENCES identifica a tabela e a coluna na tabela mãe.



Fique atento à grafia da palavra "foreign", pois ela é responsável por grande parte dos erros de sintaxe nos comandos de definição de estrutura de tabelas.

Para finalizarmos esta seção é necessário que você saiba como remover uma restrição de uma tabela. Para isso, utilize o comando ALTER TABLE em conjunto com a cláusula DROP CONSTRAINT. A sintaxe utilizada para remover restrições é apresentada no Quadro 28.

Quadro 28 – Sintaxe para remoção de restrição em uma tabela

ALTER TABLE nome_tabela

DROP PRIMARY KEY

| UNIQUE (coluna)

| CONSTRAINT nome_da_restrição;

Caso tivéssemos, por exemplo, que remover a restrição do departamento (DEPT) na tabela EMP, utilizaríamos o comando apresentado no Quadro 29.

Quadro 29 - Exemplo de sentença para remoção de restrição em uma tabela

ALTER TABLE emp

DROP CONSTRAINT emp_deptno_fk;



(i) Dica

Aqui no exemplo do Quadro 29 temos a constatação da importância de nomearmos as restrições criadas. Observe que o comando utilizou o nome da restrição de chave estrangeira (emp_deptno_fk) para eliminação.

Encerramento

Nesta seção você teve a oportunidade de conhecer as cinco restrições encontradas nos SGBDs relacionais: NOT NULL, UNIQUE, CHECK, PRIMARY KEY e FOREIGN KEY. Agora é chegado o momento de conhecermos os comandos de manipulação de dados, também conhecidos como DML (Data Manipulation Language).

CONTINUE

Referências

Referências

ORACLE. Documentação de utilização do sistema (Versão 11g). Califórnia: Oracle Corp., 2017.

CARDOSO, Vírginia M. Linguagem SQL: fundamentos e práticas. São Paulo: Saraiva, 2009.

Referências de Imagens

Divisão de Modalidades de ensino (DME), Fundação Universidade Regional de Blumenau (FURB), 2019.