

BANCO DE DADOS - RELACIONAL

PROF. NILTON

Aula de Hoje

- Implementação de restrições de integridade

Restrições de Integridade

Restrições de Integridade são usados para garantir a exatidão e a consistência dos dados em um Banco de dados relacional. Ou seja, garantir que dados representem assertivamente a realidade modelada. A integridade dos dados é tratada nas bases de dados através do conceito de integridade relacional e é garantida pelo próprio SGBD.

Restrições de Integridade

Existem vários tipos de restrições de integridade.

Restrição de Chave

Impede que uma chave primária se repita. Um campo chave primária diferencia de forma única os registros (linhas) de uma relação (tabela).

Restrição de Domínio

Definir o conjunto de valores possíveis ou permitidos que um campo pode ter.

O domínio indica os possíveis valores de um atributo. A integridade de domínio verifica se os dados são do tipo permitido (alfanumérico, numérico, etc.), tamanho do campo, se ele pode ser nulo ou não.

Restrições de Integridade

Integridade de vazios

Verifica se um campo pode ou não receber valor NULL. Subitem da integridade de domínio.

Integridade Referencial

Uma chave estrangeira de uma relação tem que coincidir com uma chave primária da sua tabela "pai" a que a chave estrangeira se refere. Ou seja, não só deve existir o atributo (campo), como também, o valor referenciado.

Restrições de Integridade

Antes de uma linha ser inserida ou atualizada, todas as restrições são avaliadas na ordem em que são definidas. Se qualquer expressão de restrição retornar falso, a linha não será inserida ou atualizada.

CHECK

Verifica se os dados atendem à condição fornecida

- CHECK (expressão) fornecida como parte de uma definição de coluna.
- CONSTRAINT [constraint_name] CHECK (expressão)

CHECK

```
CREATE TABLE t1 (  
  a INT CHECK (a>2),  
  b INT CHECK (b>2),  
  CONSTRAINT a_greater CHECK (a>b)  
);
```


Opções de Chave Estrangeira

Existem algumas opções aplicáveis às chaves estrangeiras que auxiliam a manter a integridade dos dados nas tabelas do banco de dados. Vamos lembrar a sintaxe SQL para criação de uma chave estrangeira em uma definição de tabela:

```
[CONSTRAINT [symbol]] FOREIGN KEY
```

```
  [index_name] (index_col_name, ...)
```

```
  REFERENCES tbl_name (index_col_name,...)
```

```
  [ON DELETE reference_option]
```

```
  [ON UPDATE reference_option]
```

reference_option:

RESTRICT | CASCADE | SET NULL | NO ACTION

Opções de Chave Estrangeira

Os itens entre colchetes [] são opcionais. ON DELETE significa que a ação referencial será executada quando um registro for excluído da tabela pai, e ON UPDATE indica que a ação referencial será executada quando um registro for modificado na tabela pai.

Opções de Chave Estrangeira

As principais opções para as ações referenciais são as seguintes:

CASCADE: A opção CASCADE permite excluir ou atualizar os registros relacionados presentes na tabela filha automaticamente, quando um registro da tabela pai for atualizado (ON UPDATE) ou excluído (ON DELETE). É a opção mais comum aplicada.

RESTRICT: Impede que ocorra a exclusão ou a atualização de um registro da tabela pai, caso ainda haja registros na tabela filha. Uma exceção de violação de chave estrangeira é retornada. A verificação de integridade referencial é realizada antes de tentar executar a instrução UPDATE ou DELETE

Opções de Chave Estrangeira

SET NULL: Esta opção é usada para definir com o valor NULL o campo na tabela filha quando um registro da tabela pai for atualizado ou excluído.

NO ACTION: Essa opção equivale à opção RESTRICT, porém a verificação de integridade referencial é executada após a tentativa de alterar a tabela. É a opção padrão, aplicada caso nenhuma das opções seja definida na criação da chave estrangeira.

Exemplo

```
CREATE TABLE Pai (  
  ID_Pai SMALLINT PRIMARY KEY,  
  Nome_Pai VARCHAR(50)  
);
```

```
CREATE TABLE Filho (  
  ID_Filho SMALLINT AUTO_INCREMENT PRIMARY KEY,  
  Nome_Filho VARCHAR(50),  
  ID_Pai SMALLINT,  
  CONSTRAINT fk_id_pai FOREIGN KEY (ID_Pai)  
    REFERENCES Pai(ID_Pai)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE  
);
```

Exemplo com SET NULL

Suponha que, ao excluir um pai do banco de dados, em vez de excluir imediatamente seus filhos (cascadeamento) nós queiramos manter esses registros, e o campo de ID_Pai da tabela de filhos passe então a conter um valor NULL (“filhos órfãos”).

Neste caso, a tabela de filhos deve ser criada da maneira mostrada a seguir, substituindo a cláusula ON DELETE CASCADE por ON DELETE SET NULL:

SET NULL

Como utilizar

```
CREATE TABLE Pai (  
  ID_Pai SMALLINT PRIMARY KEY,  
  Nome_Pai VARCHAR(50)  
);
```

```
CREATE TABLE Filho (  
  ID_Filho SMALLINT AUTO_INCREMENT PRIMARY KEY,  
  Nome_Filho VARCHAR(50),  
  ID_Pai SMALLINT,  
  CONSTRAINT fk_id_pai FOREIGN KEY (ID_Pai)  
    REFERENCES Pai(ID_Pai)  
    ON DELETE SET NULL  
    ON UPDATE CASCADE  
);
```

OBRIGADO!

nilton.sacco@fatec.sp.gov.br