

[◀ VOLTAR](#)

Trabalhando a estrutura das tabelas, regras e relacionamento

Apresentação da criação da estrutura das tabelas, regras e relacionamento usando comandos DDL

NESTE TÓPICO



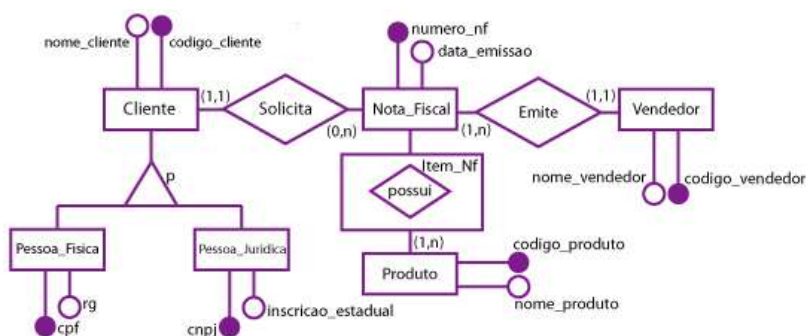
Marcar
tópico



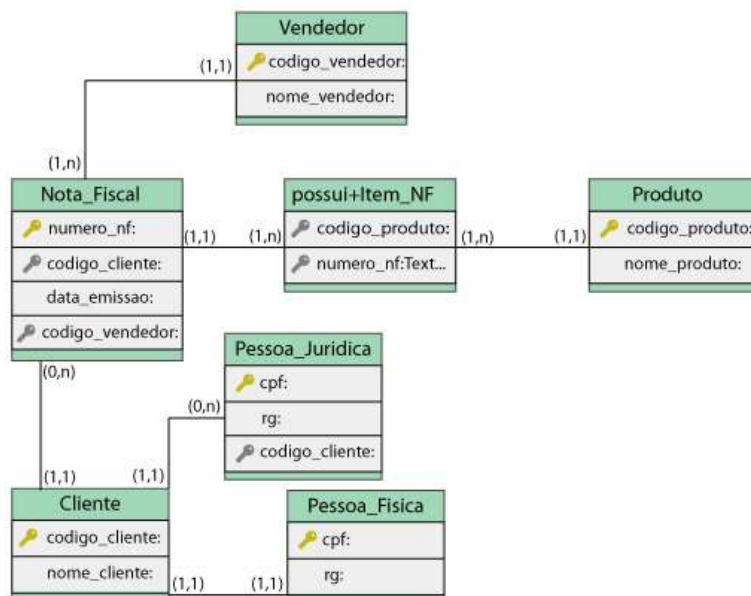
Trabalhando o relacionamento entre as tabelas

Nesta aula, demonstraremos a criação dos relacionamentos entre as tabelas existentes nos modelos apresentados no início deste curso: modelo conceitual, relacional/lógico e físico. É importante salientar que estes exemplos são retirados de materiais oficiais e livros que ministram curso de SQL para Oracle, que é o nosso SGDBR utilizado. Por se tratar de uma linguagem universal padronizada, a maioria dos comandos são aceitos em outros SGBDR, entretanto, alguns são diferentes.

Dado o modelo conceitual:



Convertido para o modelo relacional/lógico:



Transformado no modelo físico:

Tabela Cliente

| Nome da coluna | Tipo de dados | Tamanho da coluna | Regra - Constraint |
|----------------|---------------|-------------------|--------------------|
| codigo_cliente | Numérico | 4 | Chave primária |
| Nome_cliente | Alfanumérico | 20 | Não nulo |

Tabela Vendedor

| Nome da coluna | Tipo de dados | Tamanho da coluna | Regra - Constraint |
|------------------|---------------|-------------------|--------------------|
| Codigo_vendedor | Numérico | 4 | Chave primária |
| Nome_vendedor | Alfanumérico | 20 | Não nulo |
| Salario_vendedor | Numérico | 5,2 | |
| Sexo_vendedor | Character | 2 | M ou F |

Tabela Produto

| Nome da coluna | Tipo de dados | Tamanho da coluna | Regra - Constraint |
|----------------|---------------|-------------------|--------------------|
| Codigo_produto | Numérico | 4 | Chave primária |
| Nome_produto | Alfanumérico | 20 | Não nulo |
| preco_produto | Numérico | 5,2 | |

Tabela Nota_Fiscal

| Nome da coluna | Tipo de dados | Tamanho da coluna | Regra - Constraint |
|----------------|---------------|-------------------|--------------------|
| Numero_nf | Numérico | 4 | Chave primária |

| | | |
|-----------------|------|-------------------|
| Data_emissao | data | Não nulo |
| Codigo_cliente | | Chave estrangeira |
| Codigo_vendedor | | Chave estrangeira |

Após a criação dos modelos, o que nos interessa agora é o modelo físico, visto que este modelo traz as informações necessárias para o desenvolvimento prático das tabelas. Você deve prestar atenção ao nome da tabela, ao nome da coluna, aos tipos de dados e às regras existentes. Analisando o modelo físico, pode-se perceber que existem colunas que não possuem tipo e tamanho, mas são chave estrangeira, neste caso estas colunas, obrigatoriamente, possuem o mesmo tipo e tamanho de suas chaves primárias. Geralmente, sabe-se que são as ligações a partir dos modelos já existentes (Conceitual e Relacional/Lógico) ou pelo nome da coluna. Por exemplo, na tabela nota fiscal, a coluna refere-se ao cliente- código_cliente, por este motivo pode-se perceber que esta coluna estará ligada ou referenciada à tabela cliente na sua chave primária.

Exemplo da criação de uma chave estrangeira, apelidada de FK (*foreign key*).

Neste momento, é necessário seguir uma regra de criação das tabelas. Em primeiro lugar, é necessário criar tabelas que tenham apenas chave primária e que não tenham chave estrangeira. No modelo conceitual ou relacional/lógico, pode-se observar que são as tabelas: "Cliente", "Vendedor" ou "Produto". Após a identificação, a sequência de criação ficará a critério do desenvolvedor

| Nome da coluna | Tipo de dados | Tamanho da coluna | Regra - Constraint |
|-----------------|---------------|-------------------|--------------------|
| Codigo_produto | Numérico | 4 | Chave primária |
| Nome_produto | Alfanumérico | 20 | Não nulo |
| preco_produto | Numérico | 5,2 | |
| Nome da coluna | Tipo de dados | Tamanho da coluna | Regra - Constraint |
| Numero_nf | Numérico | 4 | Chave primária |
| Data_emissao | data | | Não nulo |
| Codigo_cliente | | | Chave estrangeira |
| Codigo_vendedor | | | Chave estrangeira |

Criando a tabela cliente:

```
1. create table cliente
2. (codigo_cliente number(4) constraint clie_cpf_pk primary key,
3. nome_cliente varchar2(20) constraint clie_nome_nn not null);
4.
```

Criando a tabela vendedor:

1. create table vendedor
2. (codigo_vendedor number(4) constraint ven_cod_pk primary key,
3. nome_vendedor varchar2(20) constraint ven_nome_nn not null,
4. salario_vendedor number(7,2),
5. sexo char(1) constraint ven_sexo_ck check (sexo in ('M','F')));

Nesta criação, usamos uma regra chamada check, que tem o objetivo de validar um dado antes de seu cadastro, ou seja, será verificado se o dado está em uma lista de valores que podem ser cadastrados. Por exemplo, no caso de cadastro do sexo, se o usuário informar uma letra diferente de M ou F, a instrução não será completada, ocorrerá um erro.

Criando a tabela "produto":

1. create table produto
2. (codigo_produto number(4) constraint pro_cod_pk primary key,
3. nome_produto varchar2(20) constraint pro_nome_nn not null
4. constraint pro_nome_uk unique,
5. preco_produto number(7,2));

Criando a tabela "nota fiscal", inserindo a chave estrangeira:

1. create table nota_fiscal
2. (numero_nf number(4) constraint nf_num_pk primary key,
3. data_nf date constraint nf_dt_nn not null,
4. codigo_cliente number (4) constraint nf_cli_fk references cliente,
5. codigo_vendedor number(4) constraint nf_ven_fk references vendedor);

Para ser criada a chave estrangeira, foreign key, primeiro, deve haver uma coluna que é chave primária, em seguida, a coluna deve ser ligada à chave primária, como demonstrado anteriormente. Existem maneiras diferentes de realizar a mesma instrução, nesta foi informado o tipo de dados, tamanho, nome da regra e a tabela a ser relacionada, a conexão é automática à chave primária da tabela informada. Vamos aos tipos de criação da FK.

Versão 1: mostrada no exemplo acima

1. nome_coluna tipo_dados(tamanho) constraint nome_constraint references nome_tabela
2. codigo_cliente number (4) constraint nf_cli_fk references cliente
3. Versão1.1: resumida
4. nome_coluna references nome_tabela
5. codigo_cliente number (4) references cliente

Nesta versão, a chave estrangeira assume o tipo da coluna e tamanho da chave primária relacionada, mas perceba que não existe um nome para a regra, quem insere este nome na regra é SGBDR – Oracle.

Versão 1.3: criando a chave estrangeira no final da tabela

1. create table nota_fiscal
2. (numero_nf number(4) constraint nf_num_pk primary key,
3. data_nf date constraint nf_dt_nn not null,
4. codigo_cliente number (4),
5. codigo_vendedor number(4),
6. constraint nf_cli_fk foreign key (codigo_cliente) references cliente,
7. constraint nf_ven_fk foreign key (codigo_vendedor) references vendedor);

Em algumas versões, após o nome da tabela, é colocado o nome da coluna chave primária:

1. `constraint nf_cli_fk foreign key (codigo_cliente) references cliente (codigo_cliente)`

Lembre-se: a linguagem SQL é bem versátil, existem diversas possibilidades de realizar a mesma instrução, dependerá da forma como deseja detalhar o comando, por isso algumas versões podem não estar neste curso. Mas como saber se o comando foi realmente realizado com sucesso?

O SQL Plus sempre retornará, através de mensagens, informando se o comando foi realizado com sucesso. Por exemplo, na criação da tabela, ele mostrará a mensagem "Tabela Criada", caso esta mensagem não apareça, você deverá visualizar o asterisco e provavelmente lá estará o problema.

Após a criação do relacionamento, para que consiga apagar uma tabela que tenha a sua chave primária – *primary key* relacionada a outra tabela – *foreign key*, é necessário mudar o comando, pois, por segurança da base de dados, não é possível apagar uma tabela que está relacionada a outra, ou seja, o comando - *drop table* cliente; - não funcionará mais. É necessário apagar o relacionamento através do comando `<alter table>` ou, então, reescrever o comando da seguinte forma:

1. `drop table cliente cascade constraints;`

O comando `<cascade constraints>` elimina, primeiramente, o relacionamento, para depois eliminar a tabela.

Este comando deve ser usado quando uma tabela que precisa ser eliminada tiver sua chave primária relacionada a uma chave estrangeira, caso seja o processo inverso, apague uma tabela que tenha a chave estrangeira relacionada à primária e, assim, poderá usar o comando sem o `cascade constraints`.

Se precisar renomear uma tabela, use o seguinte comando:

1. `rename nome_tabela_atual to novo_nome_tabela;`
2. `rename cliente to novocliente;`

Referências

BEIGHLEY, Lynn. *Use a Cabeça SQL*. Rio de Janeiro: Alta Books, 2008.

FANDERUFF, Damaris. *Dominando o Oracle 9i: Modelagem e Desenvolvimento*, São Paulo: Makron, 2003.

GRAVES, Mark. *Projeto de banco de dados com XML*. São Paulo: Pearson, 2003.

MORELLI, Eduardo Terra. *Oracle 9i Fundamental: SQL, PL/SQL e Administração*, São Paulo, Editora Érica, 2002.

PRICE, Jason. *Oracle Database 11g SQL*. (tradução: João Eduardo Nóbrega Tortello). Porto Alegre: Bookman, 2009.

SILVA, Robson. *Oracle Database 10g Express Edition*. São Paulo: Editora Érica, 2007.



Avalie este tópico



ANTERIOR

Exemplificando a criação da estrutura das tabelas, regras e eliminação da estrutura

Biblioteca (https://www.uninove.br/conheca-

a-

uninove/biblioteca/sobre-

a-

biblioteca/apresentacao/)

Portal Uninove

(http://www.uninove.br)

Mapa do Site



Índice

Ajuda?

PRÓXIMO
(https://ava.un
id=)

Alteração de estrutura de uma tabela

© Todos os direitos reservados

