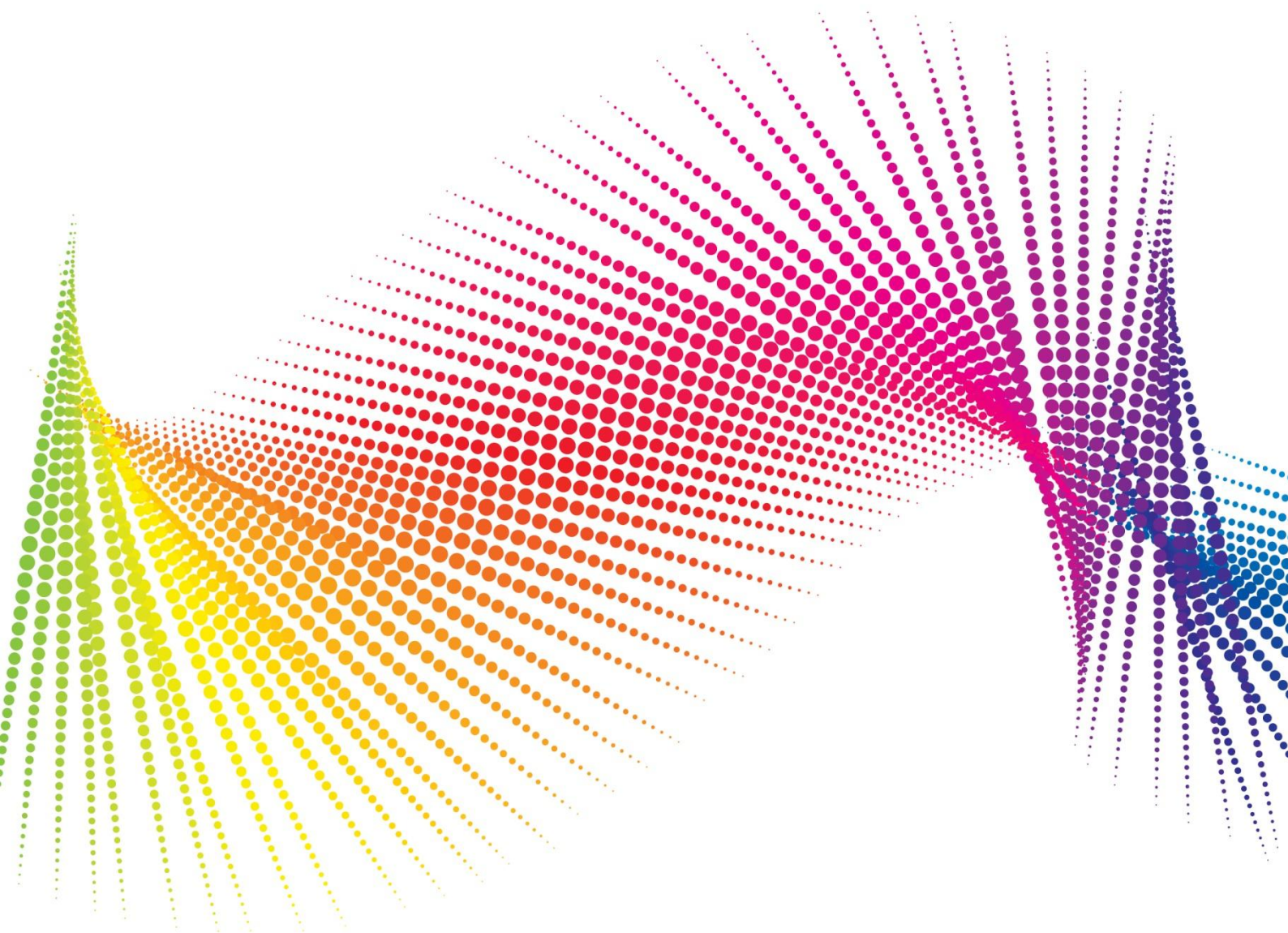


Programação para Banco de Dados

Aula 02



Este material é parte integrante da disciplina, oferecida pela UNINOVE.

O acesso às atividades, conteúdos multimídia e interativo, encontros virtuais, fóruns de discussão e a comunicação com o professor devem ser feitos diretamente no ambiente virtual de aprendizagem UNINOVE.

Uso consciente do papel.

Cause boa impressão, imprima menos.

Aula 02: SQL – Linguagem de Definição de Dados (DDL)

Objetivo: Apresentar os principais comandos da Linguagem de Definição de Dados (DDL – Data Definition Language), um subconjunto de instruções SQL que define a estrutura de um banco de dados.

Criação de tabelas

Para criar uma tabela utilizamos o comando CREATE TABLE. A sintaxe básica desse comando é a seguinte:

```
CREATE TABLE NOME_DA_TABELA (  
  NOME_DA_PRIMEIRA_COLUNA TIPO_DE_DADO (TAMANHO) ,  
  NOME_DA_SEGUNDA_COLUNA TIPO_DE_DADO (TAMANHO) ,  
  NOME_DA_ÚLTIMA_COLUNA TIPO_DE_DADO (TAMANHO) ) ;
```

Observações:

1. Os nomes das colunas não podem começar com números e não podem conter espaços: NOME, NOME1 e NOME_CLIENTE são válidos. 1NOME e NOME CLIENTE não são válidos.
2. O tamanho da coluna é expresso em caracteres ou dígitos (no caso de campos numéricos).

No exemplo abaixo vamos criar uma tabela chamada CLIENTE, com a seguinte estrutura:

NOME DA COLUNA	TIPO DE DADO	TAMANHO
CODIGO	NUMBER	5
NOME	VARCHAR2	60

```
CREATE TABLE CLIENTE (  
  CODIGO NUMBER(5) ,  
  NOME VARCHAR2(60)) ;
```

Para verificar a estrutura da tabela utilizamos o comando DESCRIBE ou a forma abreviada DESC:

```
DESCRIBE CLIENTE;
```

Ou

```
DESC CLIENTE;
```

Alguns tipos de dados não exigem que seja informado o tamanho. Observe o exemplo a seguir:

```
CREATE TABLE PEDIDO (  
  NUMERO_PEDIDO NUMBER(5) ,  
  DATA_EMISSAO DATE) ;
```

NOTA: Para o tipo de dado DATE (acima) não precisamos informar o tamanho.

Alteração de tabelas

Para alterar a estrutura de uma tabela utilizamos o comando ALTER TABLE. Podemos realizar basicamente quatro tipos de alterações em uma tabela:

- ADD: adiciona uma coluna
- MODIFY: altera a largura (o tamanho) de uma coluna
- RENAME COLUMN: altera o nome de uma coluna
- DROP COLUMN: elimina uma coluna

Adicionar uma coluna

Para adicionar uma coluna utilizamos o comando ALTER TABLE **ADD**.

A sintaxe básica desse comando é a seguinte:

```
ALTER TABLE NOME_DA_TABELA  
ADD NOME_DA_COLUNA TIPO_DE_DADO (TAMANHO) ;
```

No exemplo abaixo vamos adicionar à tabela CLIENTE uma coluna chamada TELEFONE:

NOME DA COLUNA	TIPO DE DADO	TAMANHO
TELEFONE	CHAR	9

```
ALTER TABLE CLIENTE  
ADD TELEFONE CHAR (9) ;
```

Alterar o tamanho (a largura) de uma coluna

Para alterar a largura de uma coluna utilizamos o comando ALTER TABLE **MODIFY**.

A sintaxe básica desse comando é a seguinte:

```
ALTER TABLE NOME_DA_TABELA  
MODIFY NOME_DA_COLUNA TIPO_DE_DADO (NOVO_TAMANHO) ;
```

No exemplo abaixo vamos alterar na tabela CLIENTE o tamanho da coluna NOME:

NOME DA COLUNA	TIPO DE DADO	TAMANHO
NOME	VARCHAR2	80

```
ALTER TABLE CLIENTE  
MODIFY NOME VARCHAR2 (80) ;
```

Alterar o nome de uma coluna

Para alterar o nome de uma coluna utilizamos o comando **ALTER TABLE RENAME COLUMN**.

A sintaxe básica desse comando é a seguinte:

```
ALTER TABLE NOME_DA_TABELA  
RENAME COLUMN NOME_DA_COLUNA TO NOVO_NOME_DA_COLUNA;
```

No exemplo abaixo vamos alterar na tabela **CLIENTE** o nome da coluna de **TELEFONE** para **FONE**:

```
ALTER TABLE CLIENTE  
RENAME COLUMN TELEFONE TO FONE;
```

Excluir uma coluna

Para eliminar uma coluna utilizamos o comando **ALTER TABLE DROP COLUMN**.

A sintaxe básica deste comando é a seguinte:

```
ALTER TABLE NOME_DA_TABELA  
DROP COLUMN NOME_DA_COLUNA;
```

No exemplo abaixo vamos excluir da tabela **CLIENTE** a coluna chamada **FONE**:

```
ALTER TABLE CLIENTE  
DROP COLUMN FONE;
```

Exclusão de tabelas

Para excluir uma tabela utilizamos o comando DROP TABLE.

A sintaxe básica deste comando é a seguinte:

```
DROP TABLE NOME_DA_TABELA;
```

No exemplo a seguir vamos excluir uma tabela chamada CLIENTE:

```
DROP TABLE CLIENTE;
```

Nota: algumas tabelas têm restrições ou constraints (que serão abordadas na próxima aula) associadas a elas. Neste caso, devemos utilizar o seguinte comando para eliminar a tabela e suas respectivas constraints:

```
DROP TABLE NOME_DA_TABELA CASCADE CONSTRAINTS;
```

Alterando o nome de uma tabela

Para alterar o nome de uma tabela utilizamos o comando RENAME.

A sintaxe básica deste comando é a seguinte:

```
RENAME NOME_DA_TABELA TO NOVO_NOME_DA_TABELA;
```

No exemplo abaixo vamos alterar o nome da tabela CLIENTE para CLIENTES:

```
RENAME CLIENTE TO CLIENTES;
```

Criando uma tabela com base em outra

Para criar uma nova tabela com base em outra já existente utilizamos o seguinte comando:

```
CREATE TABLE NOME_DA_NOVA_TABELA AS  
SELECT * FROM NOME_DA_TABELA;
```

No exemplo a seguir vamos criar uma tabela chamada CLIENTE_2 com base na tabela CLIENTE_1:

```
CREATE TABLE CLIENTE_2 AS SELECT * FROM CLIENTE_1;
```

Nota: O comando `SELECT * FROM CLIENTE_1` seleciona todos os dados (todas as colunas e linhas) da tabela CLIENTE_1. Portanto, a tabela CLIENTE_2 será criada não somente com a mesma estrutura da tabela CLIENTE_1, mas também com uma cópia de todos os dados dessa tabela.

Visite o Ambiente Virtual de Aprendizagem para visualizar o slideshow com telas extraídas do ambiente de trabalho SQL Plus (fornecido com o Oracle Database).

Chegamos ao final de nossa segunda aula. Agora acesse o AVA da UNINOVE e faça os exercícios propostos. Se ficar com dúvidas, não deixe de esclarecê-las com o seu professor.

REFERÊNCIAS

FEUERSTEIN, Steven. *Linguagem PL/SQL Oracle*. Rio de Janeiro: Alta Books, 2006.

McLAUGHLIN, Michael. *Oracle Database 11g PL/SQL Programação*. Rio de Janeiro: Alta Books, 2009.

MORELLI, Eduardo Terra. *Oracle 9i fundamental SQL, PL/SQL e Administração*. São Paulo: Érica, 2005.

RAMALHO, José Antonio. *Oracle 10g*. São Paulo: Thomson, 2005.

URMAN, Scott. *Oracle 9i programação PL/SQL*. Rio de Janeiro: Elsevier, 2002.