

---

# SQL - Structured Query Language

## Unidade 2 – Manipulação Básica de Dados – Parte 2

Prof. Daniel Callegari  
Escola Politécnica – PUCRS

---

### 1. Valores Nulos

Quando criamos uma tabela, podemos especificar, para cada campo, se ele é obrigatório ou não. Para especificar que um campo é obrigatório indicamos NOT NULL, e para especificar que um campo é de conteúdo opcional indicamos NULL. O padrão da linguagem SQL é NULL (campo opcional). Por exemplo:

```
CREATE TABLE PESSOAS
(
    cpf          VARCHAR(20)  NOT NULL,
    nome         VARCHAR(150) NOT NULL,
    idade        NUMBER(3)     NULL,
    endereco     VARCHAR(150)

    -- o campo endereço é NULL implicitamente
);
```

Depois, para inserir registros podemos especificar cada um dos campos da tabela (e inclusive a sua ordem).

```
-- ordem normal das colunas:
INSERT INTO PESSOAS (cpf, nome, idade, endereco)
VALUES ('32809', 'Maria', 25, 'Rua A, 20');

-- outra ordem qualquer das colunas:
INSERT INTO PESSOAS (idade, endereco, cpf, nome)
VALUES (25, 'Rua A, 20', '30599', 'Pedro');

-- valores nulos:
INSERT INTO PESSOAS (cpf, nome, idade, endereco)
VALUES ('29385', 'Carlos', NULL, NULL);

INSERT INTO PESSOAS (cpf, nome, idade, endereco)
VALUES ('39582', 'Alice', 80, NULL);

INSERT INTO PESSOAS (cpf, nome, idade, endereco)
VALUES ('78838', 'Antonio', NULL, 'Rua B, 80');
```


... ou podemos omitir alguns dos campos. Neste caso, o valor NULL é implícito. Mas, para isso, temos que omitir os nomes dos campos também no comando de inserção:

```
INSERT INTO PESSOAS (cpf, nome)
VALUES ('90038', 'Ana Paula');


INSERT INTO PESSOAS (cpf, nome, idade)
VALUES ('23487', 'Patricia', 18);

INSERT INTO PESSOAS (cpf, nome, endereco)
VALUES ('23363', 'Jose', 'Rua C, 50');
```

Repare que no comando INSERT agora especificamos o conjunto e a ordem das colunas. Essa é uma boa prática e deve ser mantida a partir deste ponto.

 Experimente adicionar outros registros.

 Experimente especificar valores nulos para campos obrigatórios.

 Experimente alguns comandos SELECT para ver como os dados aparecem. Por exemplo, descubra quantas pessoas omitiram alguma informação para o cadastro.

Para fazer busca por valores nulos, não devemos usar o operador de igualdade convencional (=). É necessário usar os operadores especiais IS NULL e IS NOT NULL:

```
-- Pessoas sem especificação de idade:
SELECT *
FROM PESSOAS
WHERE idade IS NULL;

-- Pessoas que forneceram algum endereço:
SELECT *
FROM PESSOAS
WHERE endereco IS NOT NULL;
```

## 2. Operadores LIKE e IN

O operador LIKE é usado para localizar textos. O símbolo '%' substitui zero ou mais caracteres.

```
-- Pessoas com nomes iniciando com a letra 'A':  
SELECT *  
FROM PESSOAS  
WHERE nome LIKE 'A%';  
  
-- Pessoas com nomes iniciando com 'Ana':  
SELECT *  
FROM PESSOAS  
WHERE nome LIKE 'Ana%';  
  
-- Pessoas com nomes que terminam com 'Silva':  
SELECT *  
FROM PESSOAS  
WHERE nome LIKE '%Silva';  
  
-- Pessoas com nomes que contenham 'Carlos':  
SELECT *  
FROM PESSOAS  
WHERE nome LIKE '%Carlos%';
```

O símbolo '\_' substitui exatamente um caractere.

```
-- Pode corresponder a Maria ou Mario:  
SELECT *  
FROM PESSOAS  
WHERE nome LIKE 'Mari_ da Silva';
```

O operador IN determina se um valor corresponde a qualquer um dos valores de uma lista:

```
-- Pessoas que tenham 25, 30 ou 40 anos de idade:  
SELECT *  
FROM PESSOAS  
WHERE idade IN (25, 30, 40);
```

### 3. Removendo e adicionando novas colunas em tabelas

Remover e adicionar colunas em uma tabela significa alterar a sua estrutura. Para isso usamos o comando ALTER TABLE.

Para remover uma coluna de uma tabela usamos o comando ALTER TABLE com a cláusula DROP COLUMN:

```
ALTER TABLE PESSOAS
DROP COLUMN idade;
```

Para adicionar uma nova coluna em uma tabela usamos o comando ALTER TABLE com a cláusula ADD:

```
ALTER TABLE PESSOAS
ADD sexo CHAR(1);
```

### 4. Manipulando Datas

Nossa tabela PESSOAS armazena a idade de cada pessoa, mas essa não foi uma boa escolha de projeto porque precisaríamos atualizar as idades ano após ano (qual seria o comando?). Uma escolha melhor teria sido:

```
CREATE TABLE PESSOAS
(
    cpf          VARCHAR(20)  NOT NULL,
    nome         VARCHAR(150) NOT NULL,
    datanasc     DATE          NULL,    -- alterado!
    endereco     VARCHAR(150)  NULL
);
```

Usando os comandos para alteração da tabela, vamos modificar a sua estrutura para substituir idade por data de nascimento:

```
ALTER TABLE PESSOAS
DROP COLUMN idade;

ALTER TABLE PESSOAS
ADD datanasc DATE NULL;
```

**Observação:** A manipulação de datas é muito dependente do produto de banco de dados que se está usando. As informações que constam aqui se referem ao SGBD da Oracle. Consulte a documentação do SGBD que você está usando.


O tipo DATE armazena o século, todos os quatro dígitos de um ano, o mês, o dia, a hora (formato 24h), os minutos e os segundos.

#### 4.1 Inserindo datas em um campo DATE

O formato padrão varia conforme a instalação do SGBD. Supondo que a sua instalação apresenta o formato DD-MON-YYYY, temos um dia com 2 dígitos, as três primeiras letras do mês (inglês) e um ano com 4 dígitos. Exemplo:

```
INSERT INTO PESSOAS (cpf, nome, datanasc, endereco)
VALUES
('29048', 'Roberto', '03-FEB-1980', 'Rua D, 80');

-- Formato alternativo (padrão ANSI YYYY-MM-DD):
-- Deve-se acrescentar a palavra DATE antes da data.
INSERT INTO PESSOAS (cpf, nome, datanasc, endereco)
VALUES
('29048', 'Roberto', DATE '1980-02-03', 'Rua D, 80');
```

 Veja como ficou usando o comando SELECT. Repare que o formato padrão é DD-MON-YY.

#### 4.2 Funções de conversão de datas

As funções TO\_CHAR() e TO\_DATE() convertem uma data/horário em uma string e vice-versa.


A sintaxe geral é TO\_CHAR(x [, formato]) e TO\_DATE(x [, formato]).

O elemento SYSDATE captura data e horas atuais.

Os especificadores de formato podem ser obtidos nos links a seguir:

[http://www.oradev.com/oracle\\_date\\_format.jsp](http://www.oradev.com/oracle_date_format.jsp)

[http://docs.oracle.com/cd/B19306\\_01/server.102/b14200/sql\\_elements004.htm](http://docs.oracle.com/cd/B19306_01/server.102/b14200/sql_elements004.htm)


 Experimente os seguintes comandos:

```
SELECT TO_CHAR(SYSDATE, 'MONTH, DD, YYYY HH24:MI:SS')
FROM DUAL;

-- DUAL é uma tabela interna do Oracle usada com o
-- comando SELECT quando não precisamos de uma tabela
-- real do banco de dados.

SELECT nome, TO_CHAR(datanasc, 'MONTH, DD, YYYY')
FROM PESSOAS;
```

 Especifique outros formatos com base na documentação da Oracle.

 Experimente agora os seguintes comandos:

```
INSERT INTO PESSOAS (cpf, nome, datanasc, endereco)
VALUES (
    '29920',
    'Beto',
    TO_DATE('25-FEB-1979 21:36:28',
           'DD-MON-YYYY HH24:MI:SS'),
    'Rua E, 80'
);
```

Por fim, é possível alterar o formato de data padrão da sessão atual com o comando:

```
ALTER SESSION SET NLS_DATE_FORMAT = 'MONTH-DD-YYYY';
```

Observação: Cada sessão pode ter um formato padrão diferente. Uma sessão é iniciada quando abrimos uma conexão com o banco de dados e é terminada quando a conexão é fechada.

#### 4.4 Aritmética de Datas

Em SQL é possível realizar as seguintes operações sobre datas:

- DATE + NUMBER = DATE
- DATE – NUMBER = DATE
- DATE – DATE = número de dias entre as datas

Por exemplo:

```
SELECT SYSDATE + 1
FROM DUAL;
```

Observação para Oracle: A tabela DUAL é uma tabela “dummy”. Ela contém somente uma coluna chamada “dummy” e apenas uma linha que contém o valor 'X'. Ela é utilizada sempre que se deseja retornar uma única linha em uma consulta e também porque todo comando SELECT deve possuir uma cláusula FROM.

Observação para SQL Server: No SGDB da Microsoft utilizamos `getdate()` e não incluímos a cláusula FROM DUAL.

#### Dicas finais desta aula

- Lembre-se de que buscas por valores nulos são feitas usando o operador IS.
- Pesquise sobre outros tipos de dados temporais, como TIMESTAMP.

-X-