

# Tipos de Dados do SQL da Oracle

Paulino Ng

2020-09-23

# Tipos de Dados

- ▶ O tipo de dado de uma coluna determina o tipo de informação que uma coluna guarda.
- ▶ Resumo de tipos de dados:
  - ▶ Caracteres
  - ▶ Inteiros
  - ▶ Decimais
  - ▶ Monetários
  - ▶ Data e hora
  - ▶ Binários
  - ▶ Seriais

# CHAR

- ▶ Usado para *strings* de **comprimento fixo**, isto é, você é obrigado a dizer o tamanho da string, pode ter de 1 a 2000 bytes.
- ▶ Se você fornecer uma *string* de tamanho menor do que o pré-especificado, espaços brancos serão inseridos na *string*.
- ▶ Se o valor inserido for maior, o BD da Oracle vai retornar *erro*.
- ▶ Ao comparar valores do tipo CHAR, o BD da Oracle completa a *string* menor com *brancos*.

# VARCHAR2

- ▶ Usado para armazenar *strings* de tamanho variado. De 1 a 4000 bytes, você deve especificar o máximo de caracteres.
- ▶ Se tentar inserir uma quantidade maior de bytes, o BD vai retornar erro.

## Semântica do comprimento para tipos caracteres

- ▶ Caracteres são representados usando diferentes codificações. A mais simples é o antigo código ASCII de 7 bits. Infelizmente, esta codificação só permite representar os caracteres do inglês americano, ele sequer tem o símbolo de £ (libra), usado no inglês britânico.
- ▶ O comprimento das *strings* é simples quando a codificação usa um único byte, mas torna-se mais complexa quando são usados códigos de tamanho variado como o **UTF-8**.
- ▶ O BD da Oracle permite trabalhar com comprimento em bytes (`VARCHAR2(20 BYTE)` e `SUBSTRB(<string>, 1, 5)`) ou em caracteres (`VARCHAR2(10 CHAR)` e `SUBSTR(<string>, 1, 5)`).
- ▶ O parâmetro `NLS_LENGTH_SEMANTICS` decide se uma nova coluna usa a semântica de byte ou a de carácter. O padrão (*default*) é byte.

# NCHAR e NVARCHAR2

- ▶ Estes são novos tipos de dados para *strings* que usam codificação UTF-8.
- ▶ Eles usam sempre a semântica de caracteres.

# NUMBER

- ▶ Armazena números de ponto fixo e flutuantes.
- ▶ Os número inteiros têm até 38 dígitos de precisão e vão de:
  - ▶ positivos: de  $1 \times 10^{-130}$  a  $9.99...9 \times 10^{125}$
  - ▶ negativos: de  $-1 \times 10^{-130}$  a  $-9.99...9 \times 10^{125}$
  - ▶ zero
- ▶ Você pode especificar o número máximo de dígitos e a quantidade de dígitos depois da vírgula com:  
`NUMBER(precisao, escala)`

## Números em ponto-flutuante

- ▶ A Oracle oferece 2 tipos de números em ponto flutuante. Eles são mais eficientes do que os números com ponto fixo.
- ▶ `BINARY_FLOAT` e `BINARY_DOUBLE` são os tipos em ponto flutuante. O primeiro usa ponto flutuante de 32 bits e o segundo de 64 bits.
- ▶ Os tipos `BINARY_FLOAT` e `BINARY_DOUBLE` seguem quase toda a norma *IEEE 754*.



# DATE

- ▶ Serve para armazenar datas (dia, mês e ano) e hora (hora, minutos e segundos).
- ▶ O formato padrão é DD-MON-YY, ex.: '13-NOV-92'.
- ▶ Para mudar o formato padrão use: `ALTER SESSION SET NLS_DATE_FORMAT = 'YYYY-MM-DD';`
- ▶ O formato padrão para as horas é: HH:MI:SS
- ▶ Use a função `TO_DATE()` para converter de um formato qualquer para o formato padrão. Por exemplo: `SQL INSERT INTO birthdays (bname, bday) VALUES ('ANDY', TO_DATE('13-AUG-66 12:56 A.M.', 'DD-MON-YY HH:MI A.M.'))`;
- ▶ Novos tipos de dados para data: `TIMESTAMP`, `TIMESTAMP WITH TIME ZONE` E `TIMESTAMP WITH LOCAL TIME ZONE`.

## LOB (Large Object)

- ▶ Permite armazenar grandes quantidades de dados com os tipos BLOB, CLOB, NCLOB e BFILE.
- ▶ Serve para armazenar grandes blocos de dados não estruturados como:
  - ▶ textos
  - ▶ imagens
  - ▶ vídeos
  - ▶ áudio
- ▶ BLOB é usado para armazenar dados binários não estruturados com até 128 TB.
- ▶ CLOB e NCLOB servem para armazenar até 128TB de caracteres.
- ▶ BFILE armazena dados binários fora do BD em arquivos do SO.