



**CWI SOFTWARE**

banco de dados 2

**Crescer 2017-1**

**ORACLE®**

VS

Microsoft®  
**SQL Server™**

---

# Diferenças entre SQL Server e Oracle

André Luís Nunes

Junho/2017

# Diferenças

Esta apresentação resume as principais diferenças entre as 2 tecnologias que dominam o mercado de desenvolvimento de sistemas corporativos.

Em relação ao que foi visto com SQL Server apenas o que é diferente no Oracle será visto aqui.

The Oracle logo, featuring the word "ORACLE" in a bold, red, sans-serif font with a registered trademark symbol.

VS

The Microsoft SQL Server logo, with "Microsoft" in a small grey font above "SQL Server" in a large, bold, dark grey font.



## Ferramentas utilizadas

Será realizada a instalação do banco de dados Oracle 11g XE (*free*).

- Instância (XE) instalada em cada instação;
  - Criação de bases de dados (schema/owner) e restauração de backup;
  - Criação de usuário e administração de permissões.
  
- Ferramenta de acesso ao banco de dados:
  - SQL Developer e Data Modeler

# OBJETOS

Os principais objetos de um SGBD são:

- Tables
- Indexes
- Sequences
- Views
- Triggers
- Procedures
- Functions
- Packages (permite empacotar procedures e functions).

# OBJETOS: Triggers

Gatilhos relacionados a eventos (insert/update/delete) de uma tabela.

→ São usados geralmente para criar auditoria sobre operações nas tabelas. Permitindo identificar quem executou determinada operação, quando e quais foram as alterações.

→ Oracle: permite a definição de execução por linha afetada, enquanto no SQL Server é necessário utilizar um cursor.

## **FOR EACH ROW**

Com isso o código do gatilho será executado 1 vez para cada linha afetada na operação (insert/update/delete).

# OBJETOS: Packages

Pacotes com 2 “partes”, a **especificação** e o **corpo** do pacote.  
Permite que tenha variáveis, procedures e functions declarados internamente (eles podem ser privados ou públicos).

Permite o encapsulamento e *overload* de métodos (subprogramas).

Ver arquivo **Exemplo-Package.sql**

# OBJETOS: tipos de dados

Os tipos de dados mais comuns do SQL Server são os seguintes:

- INTEGER: inteiro.
- NUMBER(I,D): valores flutuantes.
- VARCHAR2(N): caracteres de tamanho variável.
- CHAR (N): caracteres de tamanho fixo.
- DATE: data e hora.
- CLOB: texto variável (até 128 terabytes).



## TIPOS DE COMANDOS: DDL – Create > Table - autoincremento

Oracle não possui autoincremento como propriedade de coluna.

É necessário criar uma **sequence** (outro objeto) e utilizá-lo na inserção.

Não fica vinculado ao objeto (é isolado);

Normalmente é criada um gatilho (trigger) que vai utilizar a sequence no insert, desta forma “criamos um autoincremento” para a tabela.

Verificar arquivo **Exemplo-Trigger-Autoincremento.sql**

## TIPOS DE COMANDOS: DDL – Create > Table - **sequence**

Criação de uma sequence:

```
CREATE SEQUENCE SEQ_CIDADE  
  MINVALUE 1  
  MAXVALUE 999999999  
  START WITH 150  
  INCREMENT BY 1  
  NOCACHE;
```

Exemplo do uso da sequence:

```
Insert Into Cidade  
  (IDCidade, Nome_Cidade, UF)  
Values  
  (SEQ_CIDADE.NextVal, 'Morro da Pedra', 'RS');
```

## TIPOS DE COMANDOS: DDL – Create > Table - copy

Criando uma tabela a partir de outra já existente.

### “Copy table”

Executamos um “CREATE TABLE ...” para uma consulta/tabela já existente:

```
CREATE TABLE <NovaTabela> AS  
SELECT <ColumnNameA>, <ColumnNameB>, ...  
FROM <TabelaOrigem>;
```

Este comando incluirá todos os registros na tabela nova. Para restringir os registros leia mais em Select->Where (a seguir)

# MANIPULAÇÃO

# TIPOS DE COMANDOS: DML – Manipulação

Comandos que manipulam dados:

→ INSERT

→ UPDATE

→ DELETE

→ SELECT

```
Insert into Cidade  
  (IDCidade, Nome, UF)  
Values  
  (1, 'São Leopoldo', 'RS');
```

» O comando SELECT não manipula nenhuma informação, mas é considerado um DML.

» Os dados inseridos são CASE SENSITIVE, porém nas pesquisas não são considerados no SQL Server, diferente do Oracle.

CONSULTA

# TIPOS DE COMANDOS: DML – Manipulação > Select

Comandos que manipulam dados:

→ INSERT

→ UPDATE

→ DELETE

→ **SELECT**

```
Select <Column1>, <Column2>  
From   <TableName>  
[Where <predicate>]  
[Order by <column>];
```

```
Select IDCidade, Nome  
From   Cidade;
```

» Para consultar os dados deve ser considerado o CASE SENSITIVE (upper/lower) utilizados na grãção do registro.

» Os comandos SQL **não são** CASE SENSITIVE.

## SQL – Select > Order by

No Oracle é possível ordenar os nulos primeiro (first) ou por último (last), o padrão é por último.

» Por padrão a opção de ordenar nulos é NULLS LAST:

```
Select NomeEmpregado, Comissao  
From   Empregado  
Order by Comissao [NULLS LAST];
```

» É possível especificar que os valores nulos serão exibidos no início.

```
Select NomeEmpregado, Comissao  
From   Empregado  
Order  by Comissao NULLS FIRST
```



# SQL – Select > FUNÇÕES SQL

Para testar funções no Oracle utilize a tabela do sistema DUAL:

```
Select sysdate as DataAtual,  
        upper('rio grande do sul') as Estado  
From dual;
```

No SQL Server não é necessário informar o FROM

```
Select getdate() as DataAtual,  
        upper('rio grande do sul') as Estado  
go
```

# SQL – Select > FUNÇÕES SQL - conversões

Permite converter determinado tipo de dados.

❑ **TO\_CHAR**: converte para caracter.

`TO_CHAR( <expressão>, [<formato>] )`

❑ **TO\_NUMBER**: converte para número.

`TO_NUMBER( <expressão> )`

❑ **TO\_DATE**: converte para data/hora.

`TO_DATE( <expressão>, <formato> )`

# SQL – Select > FUNÇÕES SQL - TO\_CHAR

Exemplos de conversão para String (to\_char):

```
Select
  TO_CHAR(1034.2) Numero_pra_char,
  TO_CHAR(1034.2, '999g990d00') Numero_pra_char,
  TO_CHAR(sysdate, 'DD-MM-YYYY HH24:MI') DataH24_pra_char,
  TO_CHAR(sysdate, 'DD-MM-YYYY HH:MI:SS') DataH12_pra_char,
  TO_CHAR(sysdate, 'DD-MON-YYYY') Data_pra_char
From
  dual;
```

# SQL – Select > FUNÇÕES SQL - TO\_NUMBER

Exemplos de conversão para número (to\_number):

```
Select  
    TO_NUMBER('30495,3234')    Char_Para_Numerico  
From  
    dual;
```

# SQL – Select > FUNÇÕES SQL - TO\_DATE

Exemplos de conversão para data (to\_date):

```
Select
  TO_DATE(201107, 'YYYYMM')    AnoMes,
  TO_DATE('2011-09-20', 'YYYY-MM-DD')    DataNova,
  TO_DATE('2011-09-20 15:04', 'YYYY-MM-DD HH24:MI')    DtHora
From
  dual;
```

## EXTRAS - copy table

Copiando uma tabela (estrutura + dados):

```
CREATE TABLE CopiaCidade  
SELECT *  
FROM Cidade;
```

Inserindo os dados de uma tabela a partir de uma consulta:

```
INSERT INTO CopiaCidade  
    (IDCidade, Nome)  
SELECT IDCidade, Nome  
FROM Cidade;
```

# EXTRAS - view

Quando utilizar/criar uma view (visão) ?

- Quando a consulta será reutilizada.

```
Create view vwCargo as
  Select Cargo,
         Count(1) TotalEmpregados
From   Empregado
Group by Cargo
```

Consultando/executando uma view:

```
Select * From vwCargo;
```

# EXTRAS - top

Oracle não possui o "TOP"

- É necessário utilizar uma sub-query e o recurso ROWNUM.

```
Select Cargo, TotalEmpregados
From
  (Select Cargo, TotalEmpregados
   From vwCargo
   Order by TotalEmpregados desc)
Where rownum = 1
```

Não possui o recurso WITH TIES

Mais informações: [http://www.orafaq.com/faq/how\\_does\\_one\\_select\\_the\\_top\\_n\\_rows\\_from\\_a\\_table](http://www.orafaq.com/faq/how_does_one_select_the_top_n_rows_from_a_table)



# EXTRAS - top com rank()

A função RANK()

- Permite a criação de um ranking, permitindo o uso de paginação nas consultas.

```
Select Cargo, TotalEmpregados
From
    (Select Cargo,
        TotalEmpregados,
        Rank() Over (Order by TotalEmpregados desc) ordem
    From vwCargo)
Where Ordem between 2 and 4
```

Na consulta acima exibirá as linhas entre a 2ª e a 4ª posição.

Funciona também no SQL Server.