Executando Comandos SQL Básicos

Curso de Introdução a Oracle 10g: SQL

Prof.: Marlon Mendes Minussi
marlonminussi@gmail.br

Objetivos

- Listas as características do comando SQL SELECT
- Executar um comando SELECT básico
- Diferenciar comandos SQL de comandos SQL*Plus
- Para extrair dados a partir de um banco de dados você precisa utilizar o comando SQL (structured query language) SELECT. Você pode precisar restringir as colunas que serão exibidas.

Características do Comando SQL SELECT

Seleção

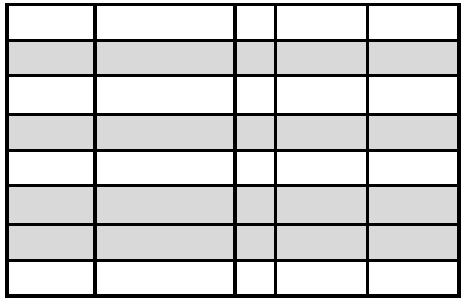


Tabela 1

Projeção

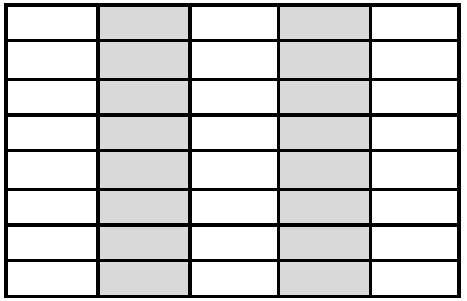


Tabela 1

Join

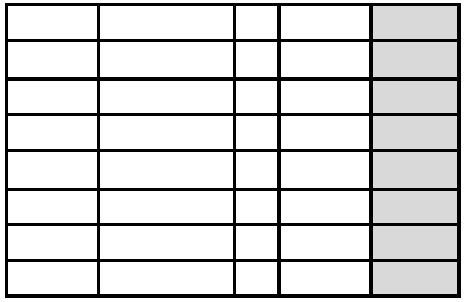
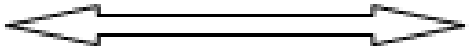


Tabela 1



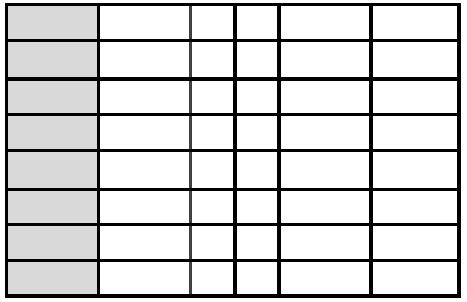


Tabela 2

Características do Comando SQL SELECT

- Um comando SELECT recupera informações a partir do banco de dados. Usando um comando SELECT você pode fazer o seguinte:
  - Seleção: você pode usar a capacidade de seleção em SQL para escolher as linhas de uma tabela que você deseja recuperar através de uma consulta. Você pode usar vários critérios para seletivamente restringir as linhas que serão visualizadas.
  - Projeção: você pode usar a capacidade de projeção em SQL para escolher as colunas de uma tabela que você deseja recuperar através de uma consulta. Você pode escolher algumas ou todas as colunas de uma tabela, de acordo com sua necessidade.
  - Join: você pode usar a capacidade de JOIN em SQL para reunir dados que estão armazenados em tabelas diferentes, criando um vinculo através de colunas que ambas as tabelas compartilhem.

Comando SELECT Básico

- Em sua forma mais simples, um comando SELECT deve incluir o seguinte:
  - Uma cláusula SELECT que especifica as colunas a serem exibidas.
  - Uma cláusula FROM que especifica as tabelas que possuem as colunas listadas na cláusula SELECT.
- Sintaxe:

```
SELECT [DISTINCT] {*, column [alias],...}
FROM       table;
```

Comando SELECT Básico

- Na sintaxe:
- SELECT: é uma lista de uma ou mais colunas;
- DISTINCT: suprime duplicidades;
- *: seleciona todas as colunas;
- column: seleciona a coluna nomeada;
- alias: fornece para as colunas selecionadas títulos diferentes;
- FROM table: especifica a tabela que contem as colunas.

Escrevendo Comandos SQL

- Seguindo regras simples e as diretrizes apresentadas abaixo, você pode construir comandos válidos que são tanto fáceis de ler quando de editar:
- Comandos SQL não fazem distinção entre maiúsculas e minúsculas, a menos que especificado;
- Podem ser escritos em uma ou mais linhas;
- Palavras chaves (keywords) não podem ser divididas em mais de uma linha ou abreviadas;
- As diferentes cláusulas são normalmente separadas em linhas distintas para facilitar a visualização e edição do comando;
- Tabulações e indentações podem ser utilizadas para tornar o código mais legível;
- Palavras chaves normalmente são escritas em maiúsculo, enquanto que as outras palavras, como nomes de tabelas e colunas, são escritas em minúsculo;

Executando Comandos SQL

- Coloque um ponto e vírgula (;) ao término da última cláusula;
- Coloque uma barra (/) na última linha do buffer;
- Coloque uma barra (/) no prompt de SQL;
- Execute o comando RUN do SQL*Plus no prompt de SQL.

Selecionando todas as Colunas

- Ex.:

```
SELECT *
FROM   paciente;
```

```
  COD_PAC NOME                           DT_NASC          RG S FONE         EST_CIVIL  END
--------- ------------------------------ -------- ---------- - ------------ ---------- ------------------------------
        1 PEDRO BIAL PEREIRA             15/08/64 1234567890 M (21)31111587 Solteiro   RUA BARATA RIBEIRO, 320
        2 BILL GATES                     28/10/55 7784512397 M 85478965     Casado     Vale do Silício, CA, USA
        3 Steve Jobs                     24/02/55 8745874584 M 98754123     Casado     VALE DO SILÍCIO, CA, USA
        4 LUANA PIOVANI                  29/08/76 7845784541 F (21)88694545 Solteira   RUA BARATA RIBEIRO, 321
        5 Renato Gaucho                  09/09/62 7711847579 M 5188979651   Casado     Porto Alegre, RS
        6 DANRLEI                        18/04/73 8475123654 M 5132589745   Solteiro   Porto Alegre, RS
        7 CLINICA                        10/03/75 7064476299 M (51)97389225 Solteiro   Av. Brasil
        8 Joaquim                        18/11/73 8475123654 M 5132589745   Solteiro   Porto Alegre, RS
       10 Victor                         21/01/83 7711847579 M 5188979651   Casado     Porto Alegre,RS
       11 ROTH                           18/11/73 8475123654 M 5132589745   Solteiro   Porto Alegre, RS

10 linhas selecionadas.
```

Selecionando todas as colunas e todas as linhas

- Você pode exibir todas as colunas de dados de uma tabela colocando um asterisco (*) logo após a palavra chave SELECT. No exemplo acima, a tabela de pacientes possui dez colunas: COD_PAC, NOME, DT_NASC, RG, SEXO, FONE, EST_CIVIL, END, CIDADE e ESTADO. A tabela possui dez linhas, uma para cada paciente.
- Você também pode exibir todas as colunas da tabela listando-as depois da palavra chave SELECT. Por exemplo, o seguinte comando SQL, como no exemplo acima, também exibe todas as colunas e todas as linhas da tabela PACIENTE:
- Ex.:
- SELECT cod_pac, nome, dt_nasc, rg, sexo, fone, est_civil, end, cidade, estado
- FROM paciente/

Selecionando Colunas Específicas

Ex.:

```
SELECT cod_pac, nome
FROM   pacientes;
```

- Você pode usar o comando SELECT para exibir colunas específicas da tabela especificando os nomes das colunas, separados por vírgulas.
- O exemplo acima exibe todos os códigos e nomes de pacientes a partir da tabela PACIENTES.

Selecionando Colunas Específicas e todas as Linhas

- Na cláusula SELECT, especifique as colunas que você quer ver, na ordem na qual você quer que elas sejam mostradas. Por exemplo, para exibir a data de nascimento antes do nome do cliente, você utiliza o seguinte comando:

Ex.:

```
SELECT dt_nasc, nome
FROM    paciente/
```

Padrões de cabeçalho de colunas

- Cabeçalhos e dados de colunas tipo caractere como cabeçalhos e dados tipo data são alinhados à esquerda dentro do tamanho da coluna.
- Cabeçalhos e dados de colunas numéricas são alinhados à direita.
- Ex.:

```
SELECT cod_pac, nome, dt_nasc
FROM   paciente;
```

Expressões Aritméticas

- Crie expressões em dados tipo NUMBER e DATE utilizando operadores aritméticos.

Operador	Descrição
+	Soma
-	Subtração
*	Multiplicação
/	Divisão

Expressões Aritméticas

- Você pode precisar modificar a forma como os dados são exibidos, por exemplo, executando cálculos.
- Isto é possível através do uso de expressões aritméticas.
- Uma expressão aritmética pode conter nomes de colunas, valores numéricos constantes, e os operadores aritméticos.
- A figura acima mostra os operadores aritméticos disponíveis em SQL.
- Você pode usar os operadores aritméticos em qualquer cláusula de um comando SQL, exceto a cláusula FROM.

Utilizando Operadores Aritméticos

- Ex.:

```
SELECT cod_atendimento, valor, valor+300
FROM   atendimento;
```

- No exemplo utilize o operador de adição para calcular um aumento do preço de R$300 para todos os atendimentos e mostrar uma nova coluna VALOR+300 na tela.
- Note que a coluna resultante VALOR+300 não é uma nova coluna da tabela VALOR; sendo utilizada somente na exibição. Por default, o nome da coluna nova é obtido a partir da expressão de cálculo que a gerou, neste caso, VALOR+300.

Utilizando Parênteses

- Ex.:

```
SELECT cod_atendimento, valor, 10*(valor+valor)*0.17
FROM   atendimento;
```

- Você pode alterar as regras de precedência usando parênteses para especificar a ordem na qual devem ser executados os operadores.
- O exemplo exibe o código do atendimento, valor da consulta, e o cálculo de 10 consultas mais o imposto adicional. Ele calcula o imposto a partir da soma dos dois preços, multiplicando o resultado por 10 e depois por 0.17. Devido ao uso dos parênteses, a adição recebe prioridade sobre a multiplicação.

Precedência dos Operadores

*** / + -**

- Se uma expressão aritmética possui mais de um operador, os de multiplicação e divisão são avaliados primeiro. Se os operadores dentro de uma expressão são da mesma prioridade, então a avaliação é feita da esquerda para a direita.
- Você pode usar parênteses para forçar a expressão colocada dentro deles a ser avaliada primeiro.
- Ex.:

```
SELECT cod_atendimento, valor, 10*valor+valor*0.17
FROM   atendimento;
```

Precedência dos Operadores

- O exemplo exibe o código da consulta, o valor do consulta e o valor de 10 consultas mais o imposto adicional.
- Observe que as multiplicações são executadas antes da adição.
- Nota: use parênteses para reforçar a ordem padrão de precedência e melhorar a clareza do comando.
- Por exemplo, expressão poderia ser escrita desta forma, sem mudança no resultado: (10*valor)+(valor*0.17).
- Ex.:

```
SELECT cod_atendimento, valor, (10*valor)+(valor*0.17)
FROM   atendimento;
```

Definindo um Valor Nulo

- Ex.:

```
SELECT cod_atendimento, cod_pac, desconto, valor
FROM   atendimento;
SELECT cod_atendimento, cod_pac, desconto, valor
FROM   atendimento;
```

COD_ATENDIMENTO	COD_PAC	DESCONTO	VALOR
6	5		100
7	6		100
8	2		75,5
2	2	15	100
3	3	7,5	70,5
4	4		100
5	2	11,32	70,5
9	1	7,5	75,5
13	2		100
15	2		70,5
16	2		75,5
1	1	7,5	70,5

Valores nulos (null values)

- Se uma linha não possui valor para uma coluna específica é dito que o valor para esta coluna é nulo, ou que ela contém nulo.
- Um valor nulo é um valor que é indisponível, não atribuído, desconhecido, ou inaplicável. Um valor nulo não é o mesmo que zero ou um espaço. Zero é um numero, e um espaço é um caractere.
- Colunas de qualquer tipo de dado podem conter valores nulos, a menos que a coluna tenha sido definida como NOT NULL ou como PRIMARY KEY (chave primária) quando criada.
- Na coluna DESCONTO da tabela ATENDIMENTO, você pode observar que algumas consultas possuem valores de desconto sobre o valor do atendimento. Um valor nulo representa o fato do paciente não possuir desconto.

Valores Nulos em Expressões Aritméticas

- Ex.:

```
SELECT cod_atendimento, cod_pac, valor-desconto
FROM   atendimento;
```

- Se o valor de alguma coluna em uma expressão aritmética é nulo, o resultado da expressão também é nulo.
- Por exemplo, se você tentar executar uma divisão por zero, você obtém um erro.
- Porém, se você divide um número por nulo, o resultado é nulo ou desconhecido.
- No exemplo acima, os atendimentos que não possuem desconto possuem o cálculo do atendimento nulo. Uma vez que a coluna DESCONTO na expressão aritmética é nula, o resultado também é nulo.

Definindo um Alias de Coluna

- Para exibir o resultado de uma consulta, o SQL*Plus normalmente utiliza o nome da coluna selecionada como seu cabeçalho.
- Em muitos casos, este título pode não ser descritivo e conseqüentemente pode ser difícil de entender.
- Você pode mudar o cabeçalho de uma coluna utilizando um alias (apelido) de coluna.
- Especifique o alias depois da coluna na lista da cláusula SELECT utilizando um espaço como separador.
- Por default, cabeçalhos baseados em alias aparecerem em maiúsculas. Se o alias possui espaços, caracteres especiais (como # ou $), ou deve diferenciar maiúsculas e minúsculas, coloque o alias entre aspas duplas (“ “).

Utilizando um Alias de Coluna

- Ex.:

```
SELECT cod_pac codigo, nome AS paciente
FROM paciente;

SELECT nome “Paciente Preferencial”
FROM   paciente;
```

- O primeiro exemplo exibe o código e o nome de todos os pacientes. Observe que a palavra chave opcional AS foi utilizada antes do nome do alias da coluna. O resultado da consulta seria o mesmo se a palavra chave AS fosse utilizada ou não. Observe também que o comando SQL tem os alias de coluna, cod_pac e nome em minúsculas, sendo que o resultado da consulta exibe os cabeçalhos de coluna em maiúsculas. Como mencionado anteriormente, os títulos de coluna aparecem em maiúsculas por default.
- O segundo exemplo exibe o nome de todos os clientes. Uma vez que o alias “Paciente Preferencial’ contem espaços, foi colocado entre aspas duplas. Observe que o cabeçalho da coluna ficou exatamente igual ao seu alias.

Operador de Concatenação

- Você pode unir colunas, expressões aritméticas ou valores constantes para criar uma expressão de caracteres usando o operador de concatenação (||).
- Colunas em qualquer lado do operador são combinadas para fazer uma única coluna de saída.

```
SELECT cod_pac || nome AS "Pacientes"
FROM   paciente;
```

Operador de Concatenação

- No exemplo, são concatenadas as colunas COD_PAC e NOME, sendo que o resultado recebe o alias de “Pacientes”.
- Observe que o código do cliente e o seu nome são combinados obtendo-se uma única coluna de saída.
- A palavra chave AS antes do nome do alias torna a cláusula SELECT mais legível.

Strings de Caracteres Literais

- Um literal é qualquer caractere, expressão ou números incluídos na lista da cláusula SELECT que não é um nome de coluna ou um alias de coluna.
- Ele é impresso para cada linha retornada. Podem ser incluídos strings literais de texto no resultado da consulta e podem ser tratadas como uma coluna da lista do SELECT.
- Literais do tipo data e caractere devem ser incluídas dentro de aspas simples(‘ ‘), enquanto que literais numéricas não necessitam de aspas.
- Ex.:

```
SELECT nome || ' ' || 'nasceu em' || ' ' ||
                      dt_nasc AS "Nascimento do Paciente"
FROM        paciente;
```

Strings de Caracteres Literais

- O exemplo mostra os nomes e as datas de nascimento de todos os pacientes.
- A coluna possui o cabeçalho “Nascimento do Paciente”. Observe os espaços entre as aspas simples no comando SELECT. Os espaços melhoram a visualização do resultado.
- No próximo exemplo, o cod_pac, nome e telefone de cada paciente são concatenados como um literal para dar mais significado às linhas retornadas.
- Ex.:

```
SELECT cod_pac || ':  ' ||  nome ||' telefone: ' || fone  PACIENTE
FROM        paciente/
```

Linhas Duplicadas

- Ex.:

```
SELECT cidade
FROM   paciente;
```

- A menos que você indique o contrário, a consulta exibe os resultados de uma consulta sem eliminar as linhas duplicadas.
- O exemplo acima exibe todas as cidades da tabela de pacientes.
- Observe que algumas são repetidas.

Eliminando Linhas Duplicadas

- Ex.:

```
SELECT DISTINCT(cidade)
FROM   paciente;
```

- Para eliminar linhas duplicadas do resultado da consulta, inclua a palavra chave DISTINCT imediatamente após a palavra SELECT.
- No exemplo, a tabela PACIENTE na verdade possui 10 linhas, mas existe somente 4 cidades diferentes na tabela e uma nula.

Eliminando Linhas Duplicadas

- Você pode especificar múltiplas colunas depois da palavra DISTINCT.
- O qualificador DISTINCT afeta todas as colunas selecionadas, e o resultado representa uma combinação distinta das colunas.
- Ex.

```
SELECT DISTINCT(cidade) CIDADE, estado ESTADO
FROM paciente/
```