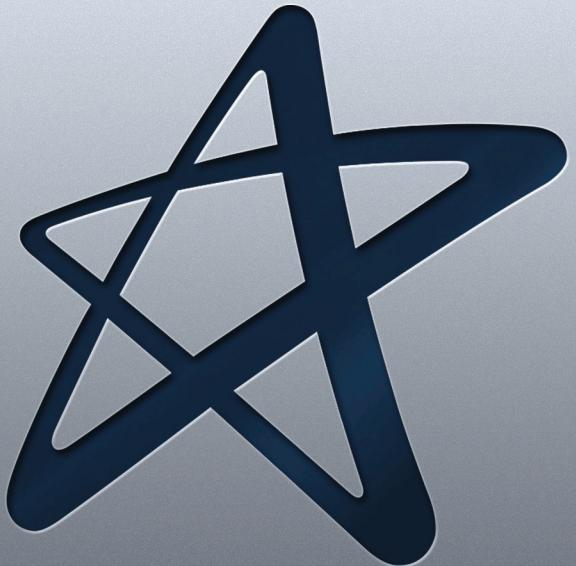


Linguagem de Banco de Dados



Material Teórico



Inserindo, alterando, excluindo e selecionando registros

Responsável pelo Conteúdo:

Prof. Ms. Luiz Carlos Reis

Revisão Textual:

Prof. Ms. Luciano Vieira Francisco

UNIDADE

Inserindo, alterando, excluindo
e selecionando registros



- Introdução
- Linhas Duplicadas
- Operadores SQL
- Expressões Negativas
- A Cláusula ORDER BY



OBJETIVO DE APRENDIZADO

- Conhecer rotinas de sincronização de dados em bancos relacionais por meio de instruções Data Manipulation Language (DML), tais como INSERT, UPDATE e DELETE.
- Verificar exemplos comuns desse tipo de tarefa, incluindo procedimentos para a inclusão, atualização e exclusão de listas de preços de produtos, lançamentos contábeis e/ou financeiros, apontamentos com horários de entrada e saída de funcionários, entre outras formas de informação.
- Conhecer a recuperação das informações em uma tabela.



Orientações de estudo

Para que o conteúdo desta Disciplina seja bem aproveitado e haja uma maior aplicabilidade na sua formação acadêmica e atuação profissional, siga algumas recomendações básicas:

Determine um horário fixo para estudar.

Mantenha o foco! Evite se distrair com as redes sociais.

Procure manter contato com seus colegas e tutores para trocar ideias! Isso amplia a aprendizagem.

Seja original! Nunca plágie trabalhos.

Aproveite as indicações de Material Complementar.

Conserve seu material e local de estudos sempre organizados.

Não se esqueça de se alimentar e se manter hidratado.

Assim:

- ✓ Organize seus estudos de maneira que passem a fazer parte da sua rotina. Por exemplo, você poderá determinar um dia e horário fixos como o seu “momento do estudo”.
- ✓ Procure se alimentar e se hidratar quando for estudar, lembre-se de que uma alimentação saudável pode proporcionar melhor aproveitamento do estudo.
- ✓ No material de cada Unidade, há leituras indicadas. Entre elas: artigos científicos, livros, vídeos e sites para aprofundar os conhecimentos adquiridos ao longo da Unidade. Além disso, você também encontrará sugestões de conteúdo extra no item **Material Complementar**, que ampliarão sua interpretação e auxiliarão no pleno entendimento dos temas abordados.
- ✓ Após o contato com o conteúdo proposto, participe dos debates mediados em fóruns de discussão, pois irão auxiliar a verificar o quanto você absorveu de conhecimento, além de propiciar o contato com seus colegas e tutores, o que se apresenta como rico espaço de troca de ideias e aprendizagem.

Introdução

Nesta Unidade veremos como incluir, alterar e excluir registros em um banco de dados e, para isso, utilizaremos os seguintes comandos:

- INSERT;
- UPDATE;
- DELETE.

Para recuperar dados de uma tabela, utilizaremos o comando SELECT e veremos também como restringir registros e expressões aritméticas, por meio de:

- SELECT;
- Distinct;
- Expressões aritméticas;
- Cláusula where;
- Alias.

! Importante!

Como pré-requisito para praticar nesta Unidade, considerando que utilizaremos o software SQLDeveloper para exemplificar os comandos, precisaremos de uma conexão AulaHR, configurada no tutorial de instalação do software.

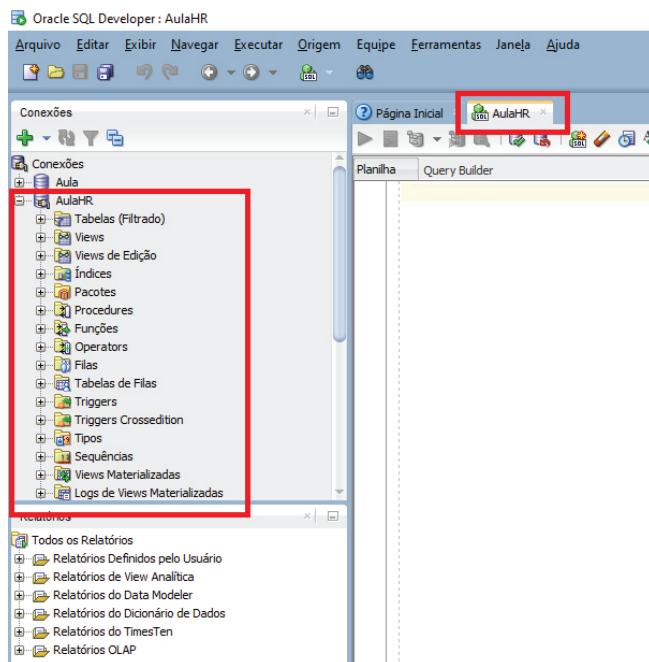


Figura 1

Criaremos a tabela pais para ajudar em nossos exemplos.

Vejamos:

```
CREATE TABLE pais
(
cd_pais number (2),
nm_pais char (45),
ds_nacionalidade char(35),
constraint pais_cd_pais_pk primary key (cd_pais),
constraint pais_nm_pais_uk unique(nm_pais)
)
```

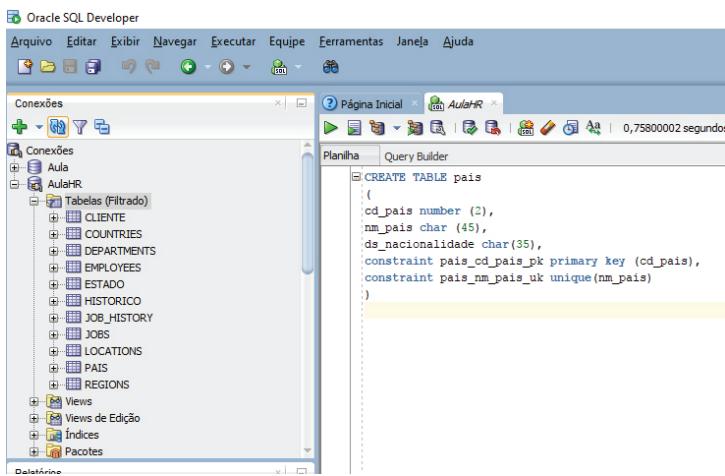


Figura 2

Inserindo registros nas tabelas:

```
Sintaxe:

INSERT INTO nome_da_tabela
  [(nome_da_coluna [, nome_da_coluna])]
VALUES
  (valor_da_coluna [, valor_da_coluna])
```

Figura 3

Valores alfanuméricos e datas deverão estar entre aspas simples (' '), conforme os seguintes casos:

- Exemplo 1:

```
INSERT INTO pais (cd_pais, nm_pais) VALUES (1,'Brasil')
```

- Exemplo 2:

```
INSERT INTO pais (cd_pais, nm_pais, ds_nacionalidade)
VALUES (2, 'Argentina',null);
```

- Exemplo 3:

```
INSERT INTO pais (cd_pais, nm_pais, ds_nacionalidade)
VALUES (4, 'Chile', 'Chilena');
INSERT INTO pais VALUES (2, 'Argentina',null);
```

- Exemplo 4:

```
INSERT INTO pais (nm_pais, ds_nacionalidade, cd_pais)
VALUES ('EUA', 'Americana', 3);
```

Para incluir dados em uma tabela, pode-se ou não inserir valores em todas as suas colunas – caso não sejam obrigatórias.

A relação de colunas da tabela poderá ser omitida quando forem incluídos valores para todas as colunas dessa tabela.

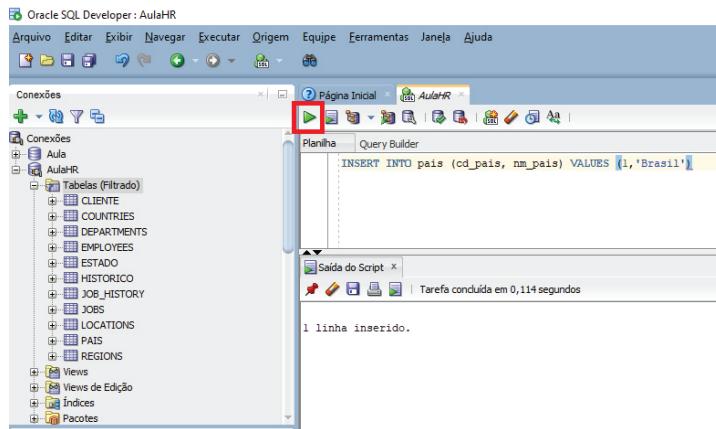


Figura 4

Perceba que caso for executada novamente a mesma consulta, o banco de dados Oracle apresentará erro de PK, pois fora implementada a restrição de Primary Key (PK) adicionada na criação da tabela.

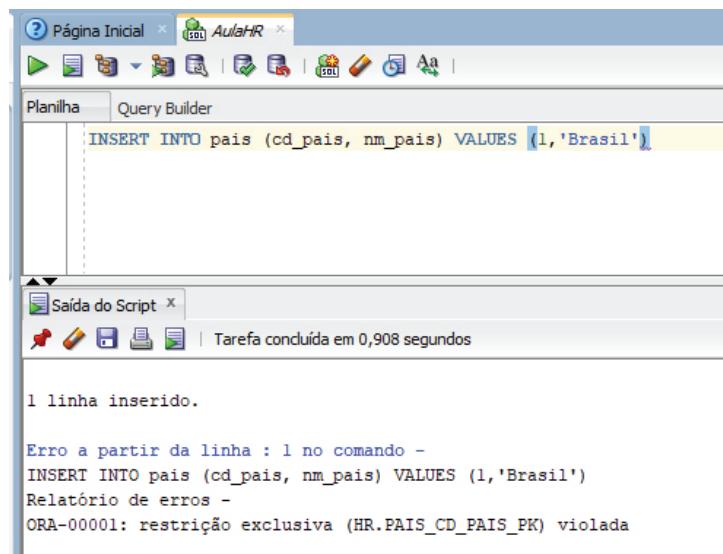


Figura 5

UPDATE

Modificar registros existentes com o comando UPDATE.

```
UPDATE Tabela
SET COLUNA = Valor
WHERE Condição
```

Figura 6

```
Update País set Nm_pais='Portugal', Ds_nacionalidade='Portug' Where Cd_pais = 3;
Update País set Nm_pais='Portugal', Ds_nacionalidade='Portug' Where Cd_pais = 3;
```

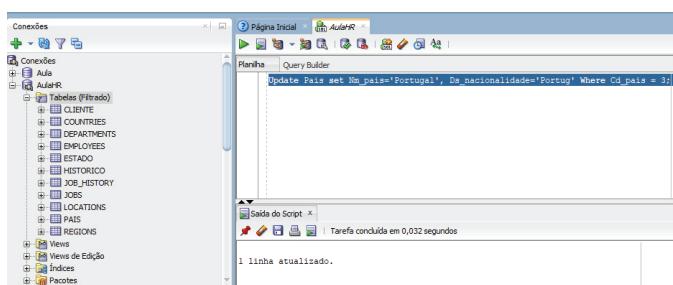


Figura 7



Importante!

Muito cuidado ao utilizar o comando UPDATE, pois caso seja omitido, a cláusula Where e todos os registros serão alterados.

DELETE

Remover registros com o comando DELETE.

```
DELETE FROM Tabela
WHERE Condição
```

Figura 8



Importante!

Como recomendação, uma boa prática é confirmar os registros que serão excluídos por meio do comando SELECT, antes da execução do comando DELETE.

```
DELETE FROM País WHERE Cd_pais = 3;
```

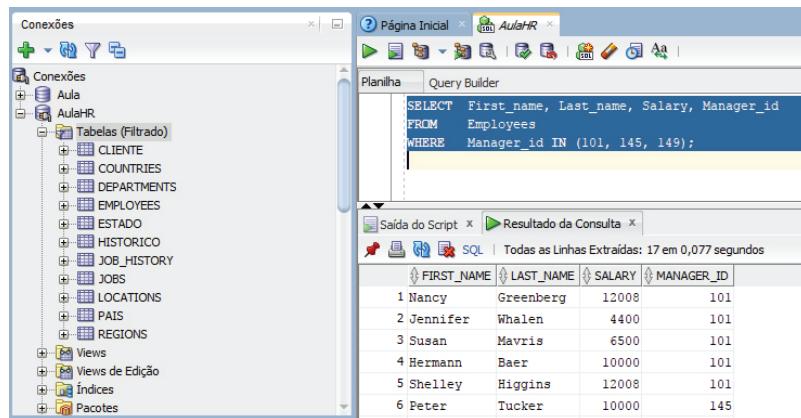


Figura 9



Importante!

Muito cuidado na utilização do comando DELETE, pois caso seja omitido, a cláusula Where e todos os registros serão excluídos.

SELECT

O comando SELECT permite a seleção e manipulação dos registros para visualização das informações que estão armazenadas no banco de dados.

Sintaxe:

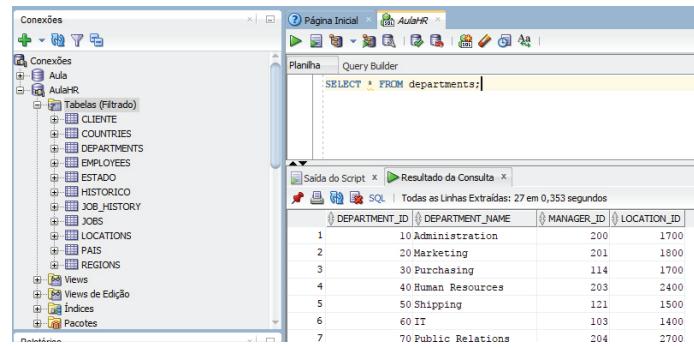
```
SELECT [distinct] col1, col2, coln  
FROM nome da tabela [alias]  
WHERE condição  
GROUP BY colunas  
HAVING condição  
ORDER BY expressão ou chave [desc]
```

Figura 10

O comando SELECT permite:

- Selecionar todas as colunas:

```
SELECT * FROM departments.
```

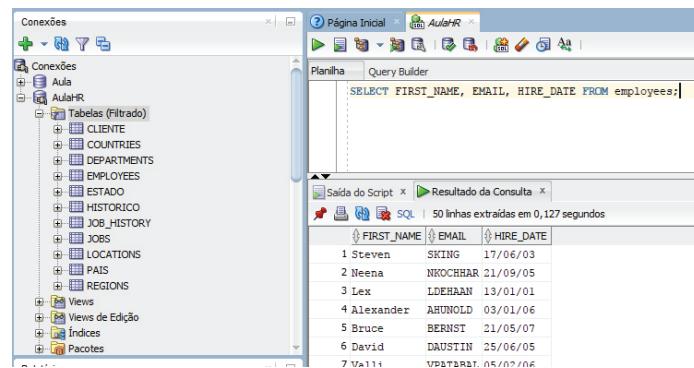


DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
1	Administration	200	1700
2	Marketing	201	1800
3	Purchasing	114	1700
4	Human Resources	203	2400
5	Shipping	121	1500
6	IT	103	1400
7	Public Relations	204	2700

Figura 11

- Selecionar colunas específicas:

```
SELECT FIRST_NAME, EMAIL, HIRE_DATE FROM employees.
```

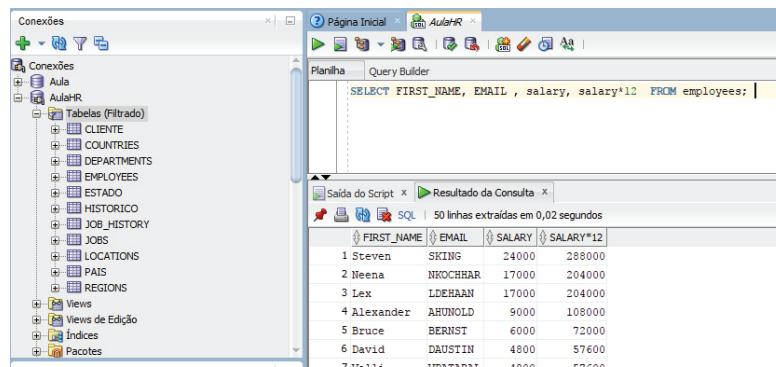


FIRST_NAME	EMAIL	HIRE_DATE
Steven	SKING	17/06/03
Neena	NKOCHHAR	21/09/05
Lex	LDEHMAN	13/01/01
Alexander	AHUNOLD	03/01/06
Bruce	BERNST	21/05/07
David	DAUSTIN	25/06/05
Valli	VPASTERI	05/07/06

Figura 12

- Selecionar expressões aritméticas:

```
SELECT FIRST_NAME, EMAIL , salary, salary*12 FROM
employees;
```



FIRST_NAME	EMAIL	SALARY	SALARY*12
Steven	SKING	24000	288000
Neena	NKOCHHAR	17000	204000
Lex	LDEHMAN	17000	204000
Alexander	AHUNOLD	9000	108000
Bruce	BERNST	6000	72000
David	DAUSTIN	4800	57600
Valli	VPASTERI	4800	57600

Figura 13

- Adicionar apelidos – alias – em determinada coluna:

```
SELECT FIRST_NAME, EMAIL , salary, salary*12 AS SALARY12 FROM
employees;
```

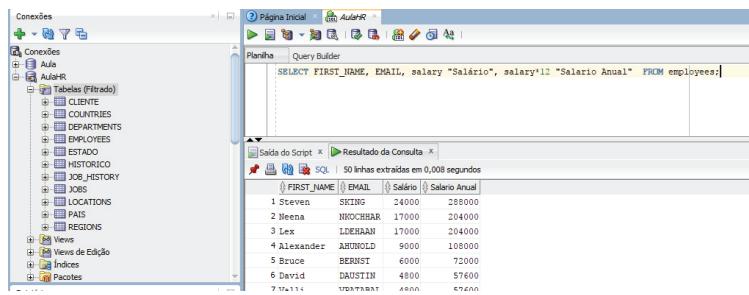


Figura 14

- Concatenar colunas:

```
SELECT FIRST_NAME, EMAIL, 'Salário: ' || salary "Salário",  
       salary*12 "Salario Anual" FROM employees;
```

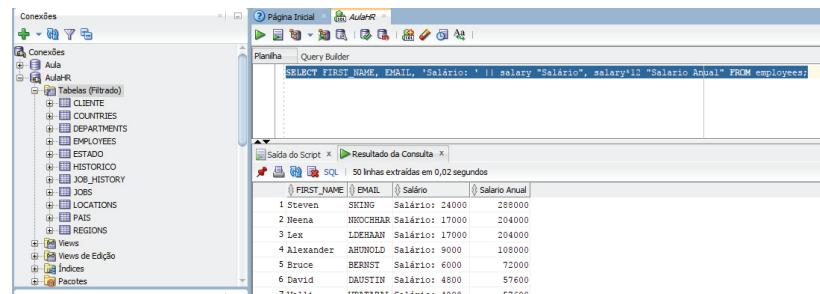


Figura 15

Linhas Duplicadas

A exibição default de consultas corresponde a todas as linhas, incluindo as duplicadas. Assim, para eliminar valores duplicados no resultado, incluímos o DISTINCT qualificador no comando SELECT:

```
SELECT distinct job_id FROM employees ORDER BY job_id;
```

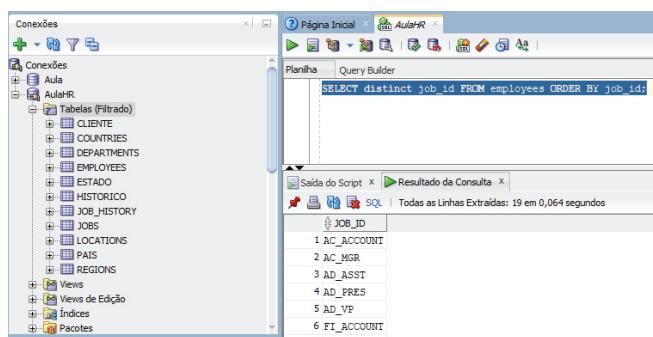


Figura 16

Perceba que não serão exibidos os cargos duplicados.

Ademais, a cláusula WHERE indica condição para um SELECT, no qual podemos ter os seguintes operadores:

Tabela 1

Operador	Descrição
=	Igual
<>	Diferente
>	Maior
<	Menor
>=	Maior ou igual
<=	Menor ou igual

Alfanuméricos e datas na cláusula WHERE devem estar entre aspas simples.
 Exemplo:

```
select first_name, hire_date, salary from employees where
salary > 10000;
```

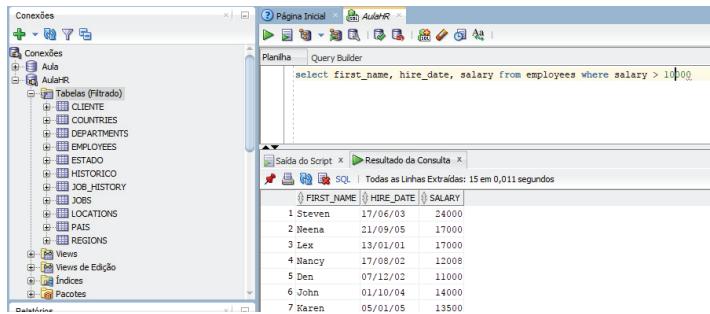


Figura 17

Operadores SQL

Existem quatro operadores SQL, os quais figuram com todos os tipos de dados:

Tabela 2

Operador	Significado
Between ... and ...	Entre dois valores – inclusive
In (lista)	Comparar uma lista de valores
Like	Comparar um parâmetro alfanumérico
Is Null	É um valor nulo

Operador *BETWEEN*

Compara uma faixa de valores, inclusive o menor e maior valor. Suponhamos que quiséssemos ver aqueles empregados cujos salários estão entre 1.000 e 2.000:

```
SELECT First_name, Last_name, Salary
FROM Employees
WHERE Salary BETWEEN 10000 AND 20000;
```

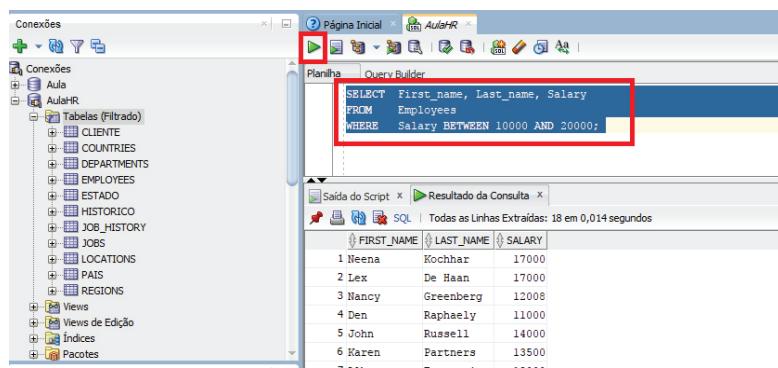


Figura 18

Para executar mais de uma linha, selecione-as e depois execute.

Operador *IN*

Compara os valores especificados dentro de uma lista. Para encontrar empregados que tenham um dos Manager com valores 101, 145 e 149, utilize o seguinte comando:

```
SELECT First_name, Last_name, Salary, Manager_id
  FROM Employees
 WHERE Manager_id IN (101, 145, 149);
```



Importante!

Se datas ou alfanuméricos forem usados na lista, precisam ser colocados entre aspas simples ('').

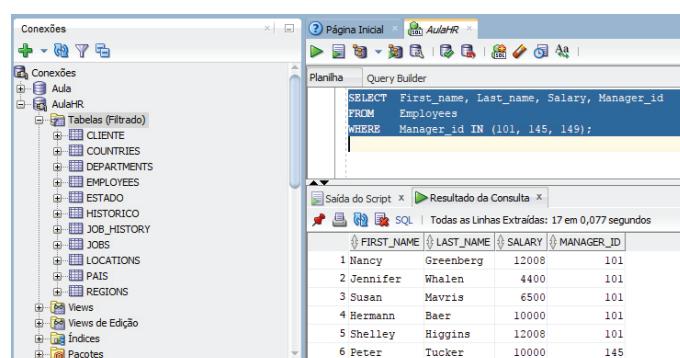


Figura 19

Operador LIKE

Algumas vezes você precisa procurar valores que não conhece exatamente. Assim, usando o operador LIKE é possível selecionar linhas combinando parâmetros alfanuméricos. Ademais, o caractere % é utilizado como “coringa” nas pesquisas. Veja um exemplo:

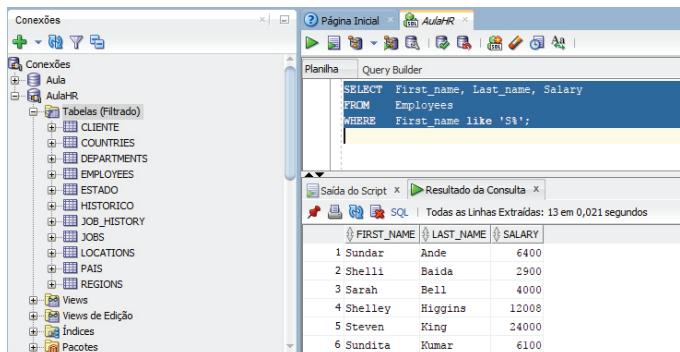
Tabela 3

Sintaxe	Descrição
Like '%cadeia'	Localiza valores que terminem com a cadeia de caracteres
Like 'cadeia%'	Localiza valores que comecem com a cadeia de caracteres
Like '%cadeia%'	Localiza valores que tenham a cadeia de caracteres em qualquer parte do campo

Fonte: elaborado pelo professor conteudista

Para listar todos os empregados que tenham o nome começando com a letra S, faça:

```
SELECT First_name, Last_name, Salary
FROM Employees
WHERE First_name like 'S%';
```



The screenshot shows the Oracle SQL Developer interface. On the left, the 'Conexões' (Connections) sidebar shows a connection named 'AulaHR'. In the center, the 'Planilha' (Worksheet) pane contains the following SQL query:

```
SELECT First_name, Last_name, Salary
FROM Employees
WHERE First_name like 'S%';
```

Below the worksheet, the 'Saída do Script' (Script Output) and 'Resultado da Consulta' (Query Result) panes are visible. The result pane displays a table with columns FIRST_NAME, LAST_NAME, and SALARY, containing the following data:

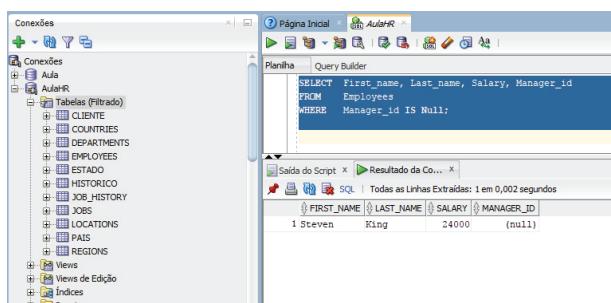
FIRST_NAME	LAST_NAME	SALARY
1 Sundar	Ande	6400
2 Shelli	Baida	2900
3 Sarah	Bell	4000
4 Shelley	Higgins	12008
5 Steven	King	24000
6 Sundita	Kumar	6100

Figura 20

Operador IS NULL

Verifica quais campos estão com valores nulos. Por exemplo, se unicamente encontrar todos os empregados que não tenham gerente, você testará um valor nulo:

```
SELECT First_name, Last_name, Salary, Manager_id
FROM Employees
WHERE Manager_id IS Null;
```



The screenshot shows the Oracle SQL Developer interface. On the left, the 'Conexões' (Connections) sidebar shows a connection named 'AulaHR'. In the center, the 'Planilha' (Worksheet) pane contains the following SQL query:

```
SELECT First_name, Last_name, Salary, Manager_id
FROM Employees
WHERE Manager_id IS Null;
```

Below the worksheet, the 'Saída do Script' (Script Output) and 'Resultado da Consulta' (Query Result) panes are visible. The result pane displays a table with columns FIRST_NAME, LAST_NAME, SALARY, and MANAGER_ID, containing the following data:

FIRST_NAME	LAST_NAME	SALARY	MANAGER_ID
1 Steven	King	24000	(null)

Figura 21

Expressões Negativas

Podemos também negar todos os operadores estudados:

Tabela 4

Sintaxe	Descrição
NOT Between	Tudo que estiver fora da faixa
NOT In	Tudo que não estiver na lista
NOT Like	Tudo que não conter a linha de caracteres
IS NOT Null	Tudo que não for nulo

```
SELECT First_name, Last_name, Salary
FROM Employees
WHERE Salary NOT BETWEEN 10000 AND 20000;
SELECT First_name, Last_name, Salary, Manager_id
FROM Employees
WHERE Manager_id NOT IN (101, 145, 149);
SELECT First_name, Last_name, Salary
FROM Employees
WHERE First_name NOT Like 'S%';
SELECT First_name, Last_name, Salary, Manager_id
FROM Employees
WHERE Manager_id IS NOT Null;
```

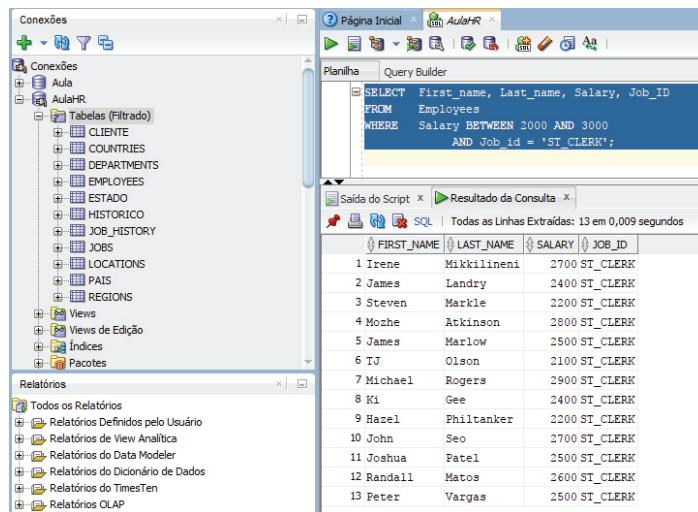
Pesquisando Dados com Múltiplas Condições

- Os operadores AND e OR devem ser usados para fazer composições de expressões lógicas;
- O predicado AND esperará que ambas as condições sejam verdadeiras;
- O predicado OR esperará que uma das condições seja verdadeira.

Nos dois seguintes exemplos as condições são as mesmas, mas cada resultado é diferente entre si. Veja como o resultado é dramaticamente alterado:

Para encontrar todos os escriturários do departamento estoque que ganhem entre 2.000 e 3.000:

```
SELECT First_name, Last_name, Salary, Job_ID
FROM Employees
WHERE Salary BETWEEN 2000 AND 3000
      AND Job_id = 'ST_CLERK';
```



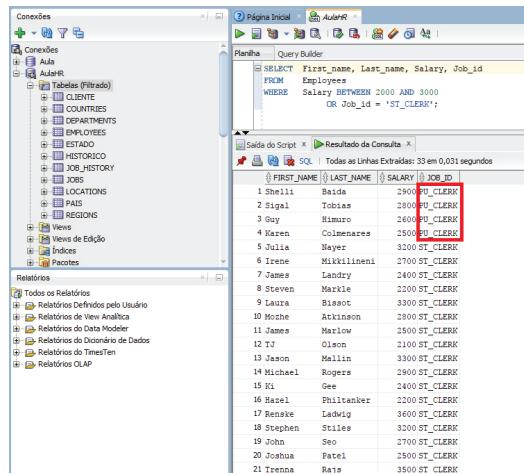
FIRST_NAME	LAST_NAME	SALARY	JOB_ID
Irene	Mikkilineni	2700	ST_CLERK
James	Landry	2400	ST_CLERK
Steven	Markle	2200	ST_CLERK
Moche	Atkinson	2800	ST_CLERK
James	Marlow	2500	ST_CLERK
TJ	Olison	2100	ST_CLERK
Michael	Rogers	2900	ST_CLERK
Ki	Gee	2400	ST_CLERK
Hazel	Philtanker	2200	ST_CLERK
John	Seo	2700	ST_CLERK
Joshua	Patel	2500	ST_CLERK
Randall	Matos	2600	ST_CLERK
Peter	Vargas	2500	ST_CLERK

Figura 22

Para encontrar todos os escriturários do departamento estoque e os funcionários que ganhem entre 2.000 e 3.000.

```
SELECT First_name, Last_name, Salary, Job_id
FROM Employees
WHERE Salary BETWEEN 2000 AND 3000
      OR Job_id = 'ST_CLERK';
```

Perceba que neste segundo caso a seleção foi bem maior, visto que existem funcionários que ganham entre 2.000 e 3.000 e que não são escriturários.



FIRST_NAME	LAST_NAME	SALARY	JOB_ID
Shelli	Baida	2900	ST_CLERK
Sigal	Tobias	2800	ST_CLERK
Guy	Himuro	2600	ST_CLERK
Karen	Colmenares	2500	ST_CLERK
Julia	Nayer	3200	ST_CLERK
Irene	Mikkilineni	2700	ST_CLERK
James	Landry	2400	ST_CLERK
Steven	Markle	2200	ST_CLERK
Laura	Bisot	3300	ST_CLERK
Moche	Atkinson	2800	ST_CLERK
James	Marlow	2500	ST_CLERK
TJ	Olison	2100	ST_CLERK
Jason	Mallin	3300	ST_CLERK
Michael	Rogers	2900	ST_CLERK
Ki	Gee	2400	ST_CLERK
Hazel	Philtanker	2200	ST_CLERK
Renske	Ledwig	3600	ST_CLERK
Stephen	Stiles	3200	ST_CLERK
John	Seo	2700	ST_CLERK
Joshua	Patel	2500	ST_CLERK
Trenna	Rais	3500	ST_CLERK

Figura 23

Você pode combinar AND e OR na mesma expressão lógica. Ou seja, quando AND e OR aparecerem na mesma cláusula WHERE, todos os AND serão feitos primeiro e, posteriormente, todos os OR o serão.

```
SELECT First_name, Last_name, Salary, Job_ID
FROM Employees
WHERE Salary > 8000 and
      Job_id = 'ST_CLERK' OR
      Job_id = 'ST_MAN'
```

Figura 24

Se você quiser selecionar todos os escriturários e gerentes que ganhem acima de 8.000, deverá empregar parênteses para estabelecer tal prioridade:

```
SELECT First_name, Last_name, Salary, Job_ID
FROM Employees
WHERE Salary > 8000 and
      (Job_id = 'ST_CLERK' OR
       Job_id = 'ST_MAN')
```

Figura 25

Os parênteses especificam a ordem pela qual os operadores devem ser avaliados – prioridade.

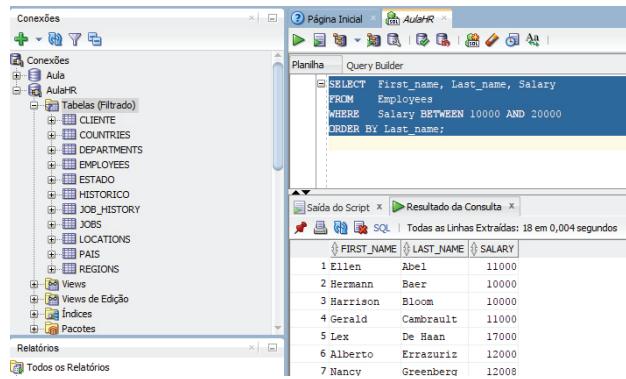
No segundo exemplo, o operador OR é avaliado antes de AND. Assim, sempre que você estiver em dúvida sobre qual dos dois operadores será feito primeiro quando a expressão é avaliada, use sempre parênteses para definir a prioridade das expressões.

A Cláusula *ORDER BY*

Normalmente, a ordem das linhas retornadas de uma pesquisa é indefinida. Assim, a cláusula ORDER BY pode ser empregada para ordenar as linhas.

Se utilizada, ORDER BY deve ser sempre a última cláusula da declaração SELECT. Para ordenar essa consulta pelo campo Last_Name, faça:

```
SELECT First_name, Last_name, Salary
FROM Employees
WHERE Salary BETWEEN 10000 AND 20000
ORDER BY Last_name;
```



The screenshot shows the Oracle SQL Developer interface. In the left sidebar, under 'Conexões', there is a connection named 'Aula-R' which is expanded to show tables like CLIENTE, COUNTRIES, DEPARTMENTS, EMPLOYEES, ESTADO, HISTORICO, JOB_HISTORY, JOBS, LOCATIONS, PAIS, and REGIONS. Below this is a 'Relatórios' section with 'Todos os Relatórios'. The main area has tabs for 'Planilha' and 'Query Builder'. The 'Planilha' tab contains the following SQL code:

```
SELECT First_name, Last_name, Salary
FROM Employees
WHERE Salary BETWEEN 10000 AND 20000
ORDER BY Last_name;
```

The 'Resultado da Consulta' tab shows the execution results:

FIRST_NAME	LAST_NAME	SALARY
Ellen	Abel	11000
Hermann	Baer	10000
Harrison	Bloom	10000
Gerald	Cambault	11000
Lex	De Haan	17000
Alberto	Errazuriz	12000
Nancy	Greenberg	12000

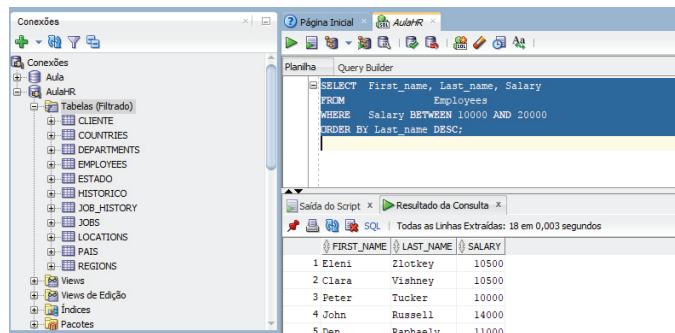
Figura 26

O padrão de ordenação é ascendente.

Invertendo o Padrão de Ordenação

Para inverter essa ordem, acrescente o comando DESC – decrescente – depois do nome das colunas da cláusula ORDER BY; enquanto que para inverter a ordem da coluna Last_name, faça:

```
SELECT First_name, Last_name, Salary
FROM Employees
WHERE Salary BETWEEN 10000 AND 20000
ORDER BY Last_name DESC;
```



The screenshot shows the Oracle SQL Developer interface, similar to Figure 26. The 'Planilha' tab contains the same SQL code as Figure 26, but with 'DESC' added to the ORDER BY clause:

```
SELECT First_name, Last_name, Salary
FROM Employees
WHERE Salary BETWEEN 10000 AND 20000
ORDER BY Last_name DESC;
```

The 'Resultado da Consulta' tab shows the execution results:

FIRST_NAME	LAST_NAME	SALARY
Eleni	Zlotkey	10500
Clara	Vishney	10500
Peter	Tucker	10000
John	Russell	14000
Den	Raphaely	11000

Figura 27

Note se tratar de ordenação por várias colunas, afinal, é possível utilizar mais de uma coluna na cláusula ORDER BY. O limite de colunas é o número dessas da tabela. Assim, na cláusula ORDER BY especifica-se as colunas que serão ordenadas, separando-as por vírgula. Se algumas ou todas serão invertidas, especifique DESC depois de cada uma das colunas.

Para ordenar por duas colunas e mostrar a ordem inversa do salário – e ordem crescente do sobrenome –, faça:

```

SELECT First_name, Last_name, Salary
FROM Employees
WHERE Salary BETWEEN 10000 AND 20000
ORDER BY Salary Desc, Last_name;

```

	FIRST_NAME	LAST_NAME	SALARY
1	Lex	De Haan	17000
2	Neena	Kochhar	17000
3	John	Russell	14000
4	Karen	Partners	13500
5	Michael	Hartstein	13000
6	Nancy	Greenberg	12008
7	Shelley	Higgins	12008

Figura 28

Para ordenar por uma coluna, esta não precisa necessariamente estar declarada no SELECT.

Ademais, a cláusula ORDER BY é empregada na pesquisa quando se quer mostrar as linhas em uma ordem específica – afinal, sem a cláusula ORDER BY, as linhas são retornadas na ordem conveniente para o Oracle.

Finalmente, tal comando não altera a ordem dos dados que estão armazenados no banco de dados

Material Complementar

Indicações para saber mais sobre os assuntos abordados nesta Unidade:



Livros

Oracle Database 11g Sql

Como complemento a esta Unidade, leia o oitavo capítulo do livro de J. Price, intitulado **Oracle Database 11g Sql**, publicado em 2009, pela editora Bookman, de Porto Alegre, RS.

SQL : guia prático

Ler do capítulo III do livro: Costa, Rogério Luis de C. **SQL : guia prático**. 2. ed. Rio de Janeiro: Brasport, 2006.



Vídeos

Oracle Server SQL 01 - Criar tabela, inserção de registros, descreve e dicas Post101

<https://youtu.be/J66VqmPd9II>

1.7 Inserir e Consultar dados de uma tabela de banco de dados Oracle

<https://youtu.be/2lsj3H1xDBI>

Referências

COSTA, R. L. de C. **SQL**: guia prático. 2. ed. Rio de Janeiro: Brasport, 2006.

MORELLI, E. M. T. **Oracle 9i fundamental**: Sql, Pl/SQL e administração. São Paulo: Érica, 2002.

PRICE, J. Oracle Database **11g Sql**. Porto Alegre, RS: Bookman, 2009.

SILBERSCHATZ, A. **Sistema de bancos de dados**. São Paulo: Pearson Education do Brasil, 2004



Cruzeiro do Sul Virtual
Educação a Distância

www.cruzeirodosulvirtual.com.br
Campus Liberdade
Rua Galvão Bueno, 868
CEP 01506-000
São Paulo - SP - Brasil
Tel: (55 11) 3385-3000



Cruzeiro do Sul
Educacional