



ORACLE

Academy



Database Programming with SQL

6-4

Autojunções e Consultas Hierárquicas

ORACLE
Academy



Objetivos

- Esta lição abrange os seguintes objetivos:
 - Construir e executar uma instrução SELECT para juntar uma tabela a si mesma por meio da autojunção
 - Interpretar o conceito de uma consulta hierárquica
 - Criar um relatório estruturado em árvore
 - Formatar dados hierárquicos
 - Excluir ramificações da estrutura em árvore

Finalidade

- Na modelagem de dados, às vezes era necessário mostrar uma entidade com uma relação consigo mesma
- Por exemplo, um funcionário também pode ser um gerente
- Mostrávamos isso usando a relação recursiva ou "orelha de porco"



Finalidade

- Quando temos uma tabela de funcionários real, um tipo de junção especial, chamado de autojunção, é necessário para acessar esses dados
- A autojunção é usada para unir uma tabela a si mesma, como se fossem duas tabelas

```
SELECT worker.last_name || ' works for ' || manager.last_name  
AS "Works for"  
FROM employees worker JOIN employees manager  
ON (worker.manager_id = manager.employee_id);
```

SELF-JOIN

- Para juntar uma tabela a si mesmo, é necessário dar dois nomes ou aliases a ela. Isso fará o banco de dados "pensar" que há duas tabelas

EMPLOYEES (trabalhador)

employee_id	last_name	manager_id
100	King	
101	Kochar	100
102	De Haan	100
103	Hunold	102
104	Ernst	103
107	Lorentz	103
124	Mourgos	100

EMPLOYEES (gerente)

employee_id	last_name
100	King
101	Kochar
102	De Haan
103	Hunold
104	Ernst
107	Lorentz
124	Mourgos

- Manager_id de trabalhadores é igual a employee_id na tabela de gerentes

SELF-JOIN

- Escolha aliases que se relacionem à associação dos dados com a tabela

EMPLOYEES (trabalhador)

employee_id	last_name	manager_id
100	King	
101	Kocher	100
102	De Haan	100
103	Hunold	102
104	Ernst	103
107	Lorentz	103
124	Mourgos	100



EMPLOYEES (gerente)

employee_id	last_name
100	King
101	Kocher
102	De Haan
103	Hunold
104	Ernst
107	Lorentz
124	Mourgos



- Manager_id na tabela de trabalhadores é igual a employee_id na tabela de gerentes

Exemplo de SELF-JOIN

```
SELECT worker.last_name, worker.manager_id, manager.last_name  
       AS "Manager name"  
FROM employees worker JOIN employees manager  
       ON (worker.manager_id = manager.employee_id);
```

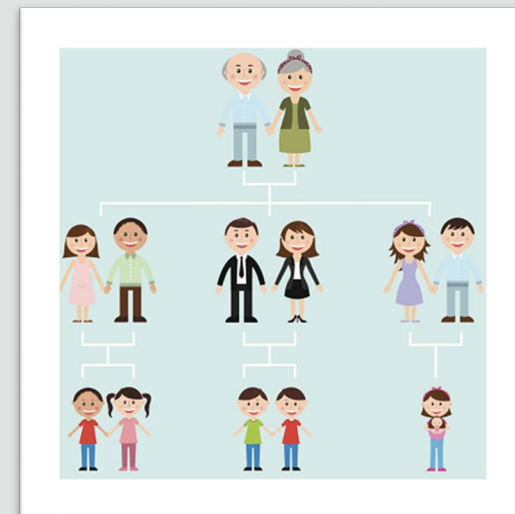
LAST_NAME	MANAGER_ID	Manager name
Kochhar	100	King
De Haan	100	King
Zlotkey	100	King
Mourgos	100	King
Hartstein	100	King
Whalen	101	Kochhar
Higgins	101	Kochhar
Hunold	102	De Haan
...

Consultas Hierárquicas

- As consultas hierárquicas têm uma relação muito próxima com as autojunções
- Nas páginas anteriores, você viu como usar as autojunções para ver o gerente direto de cada funcionário
- Com as consultas hierárquicas, também podemos ver para quem esse gerente trabalha, entre outras informações
- Com esse tipo de consulta, podemos criar um Organograma mostrando a estrutura de uma empresa ou departamento

Consultas Hierárquicas

- Imagine uma árvore genealógica com os membros mais velhos da família próximos à base ou como o tronco da árvore e os membros mais jovens como as ramificações
- As ramificações podem, por sua vez, ter outras ramificações



Usando Consultas Hierárquicas

- Ao usar as consultas hierárquicas, você pode recuperar dados com base na relação hierárquica natural entre as linhas de uma tabela
- Um banco de dados relacional não armazena registros de modo hierárquico
- No entanto, quando existe uma relação hierárquica entre as linhas de uma única tabela, um processo chamado de tree walking permite que a hierarquia seja construída
- Uma consulta hierárquica é um método para reportar as ramificações de uma árvore em uma ordem específica

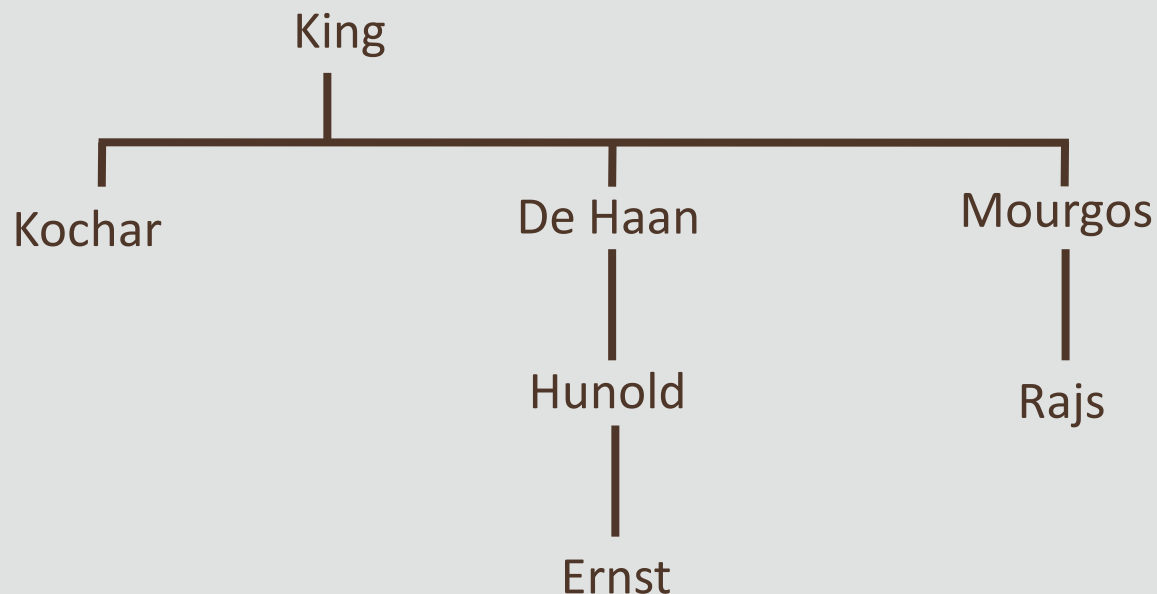
Os Dados das Consultas Hierárquicas

- Examine os dados de amostra da tabela **FUNCIONÁRIOS** abaixo e observe como você pode fazer manualmente as ligações para ver quem trabalha para quem, começando por Steven King e avançando pela árvore a partir dele

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMM_PCT	MGR_ID	DEPT_ID
100	Steven	King	SKING	515.123.4567	17-Jun-1987	AD_PRES	24000	(null)	(null)	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-Sep-1989	AD_VP	17000	(null)	100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	13-Jan-1993	AD_VP	17000	(null)	100	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	03-Jan-1990	IT_PROG	9000	(null)	102	60
104	Bruce	Ernst	BERNST	590.423.4568	21-May-1991	IT_PROG	6000	(null)	103	60
124	Kevin	Mourgos	KMOURGOS	650.123.5234	16-Nov-1999	ST_MAN	5800	(null)	100	50
141	Trenna	Rajs	TRAJS	650.121.8009	17-Oct-1995	ST_CLERK	3500	(null)	124	50

Ilustração de Consultas Hierárquicas

- O organograma que podemos criar com os dados na tabela EMPLOYEES ficará assim:



Palavras-chaves de Consultas Hierárquicas

- As consultas hierárquicas têm suas próprias palavras-chaves: **START WITH**, **CONNECT BY PRIOR** e **LEVEL**
- **START WITH** identifica qual linha usar como a Raiz da árvore que está sendo construída; **CONNECT BY PRIOR** explica como fazer as junções entre linhas; e **LEVEL** especifica quantas ramificações a árvore terá



Exemplo de Palavras-chaves de Consultas Hierárquicas

```
SELECT employee_id, last_name, job_id, manager_id
FROM employees
START WITH employee_id = 100
CONNECT BY PRIOR employee_id = manager_id
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	MANAGER_ID
100	King	AD_PRES	-
101	Kochhar	AD_VP	100
200	Whalen	AD_ASST	101
205	Higgins	AC_MGR	101
206	Gietz	AC_ACCOUNT	205
102	De Haan	AD_VP	100
103	Hunold	IT_PROG	102
104	Ernst	IT_PROG	103

Outro Exemplo de Consultas Hierárquicas

```
SELECT last_name || ' reports to ' || PRIOR last_name AS "Walk  
Top Down"  
FROM employees  
START WITH last_name = 'King'  
CONNECT BY PRIOR employee_id = manager_id;
```

Walk Top Down
King reports to
Kochhar reports to King
Whalen reports to Kochhar
Higgins reports to Kochhar
Gietz reports to Higgins
De Haan reports to King
Hunold reports to De Haan
Ernst reports to Hunold

Exemplo de Nível de Consultas Hierárquicas

- LEVEL é uma pseudocoluna usada com consultas hierárquicas e conta o número de pontos percorridos a partir da raiz da árvore

```
SELECT LEVEL, last_name ||  
' reports to ' ||  
PRIOR last_name  
  AS "Walk Top Down"  
FROM employees  
START WITH last_name = 'King'  
CONNECT BY PRIOR  
  employee_id = manager_id;
```

LEVEL	Walk Top Down
1	King reports to
2	Kochhar reports to King
3	Whalen reports to Kochhar
3	Higgins reports to Kochhar
4	Gietz reports to Higgins
2	De Haan reports to King
3	Hunold reports to De Haan
4	Ernst reports to Hunold

Relatório da Consulta Hierárquica

- Se você quisesse criar um relatório exibindo os níveis da gerência de uma empresa, começando do maior nível e criando um recuo para cada um dos níveis subsequentes, seria fácil usar a pseudocoluna LEVEL e a função LPAD para recuar os funcionários com base no nível deles

```
SELECT LPAD(last_name, LENGTH(last_name) + (LEVEL*2) - 2, '_')  
      AS "Org Chart"  
FROM employees  
START WITH last_name = 'King'  
CONNECT BY PRIOR employee_id = manager_id;
```

Níveis de Saída da Consulta Hierárquica

- Como você pode ver no resultado à direita, cada uma das linhas está recuada em dois sublinhados por nível

```
SELECT LPAD(last_name, LENGTH(last_name) +  
  (LEVEL*2)-2, '_') AS "Org_Chart"  
FROM employees  
START WITH last_name = 'King'  
CONNECT BY PRIOR employee_id = manager_id;
```

Org_Chart
King
__Kochhar
____Whalen
____Higgins
_____Gietz
__De Haan
____Hunold
_____Ernst
_____Lorentz
____Rajs
____Davies
____Matos
____Vargas
__Zlotkey
____Abel
____Taylor
____Grant
__Hartstein
____Fay

Consulta Hierárquica Ascendente

- Como você pode ver no resultado abaixo, este exemplo mostra como criar uma Consulta Hierárquica Ascendente movendo a palavra-chave PRIOR para depois do sinal de igual e usando 'Grant' na cláusula START WITH

```
SELECT LPAD(last_name, LENGTH(last_name) +  
(LEVEL*2)-2, '_') AS ORG_CHART  
FROM employees  
START WITH last_name = 'Grant'  
CONNECT BY employee_id = PRIOR manager_id
```

ORG_CHART
Grant
__Zlotkey
____King

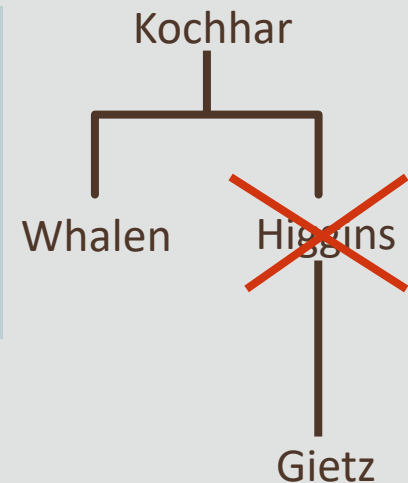
Remoção de Consultas Hierárquicas

- A remoção de ramificações da árvore pode ser feita usando a cláusula `WHERE` ou `CONNECT BY PRIOR`
- Se a cláusula `WHERE` for usada, somente a linha nomeada na instrução será excluída; se a cláusula `CONNECT BY PRIOR` for usada, a ramificação inteira será excluída

Remoção de Consultas Hierárquicas

- Por exemplo, se quisesse excluir uma única linha do resultado, você usaria a cláusula WHERE. No entanto, pareceria que Gietz trabalha diretamente para Kochhar, o que não é o caso

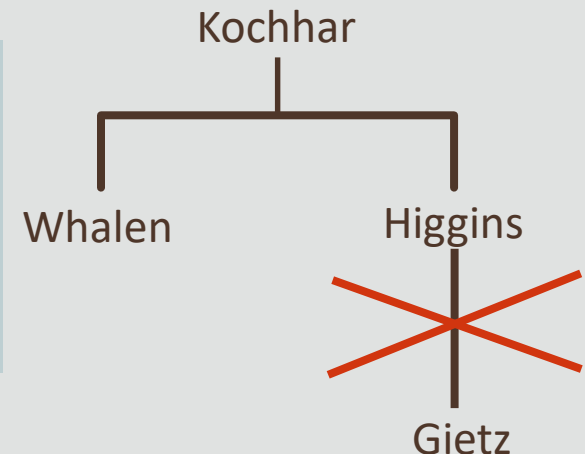
```
SELECT last_name  
FROM employees  
WHERE last_name != 'Higgins'  
START WITH last_name = 'Kochhar'  
CONNECT BY PRIOR employee_id = manager_id;
```



Remoção de Consultas Hierárquicas

- Se, no entanto, quisesse excluir uma linha e todas as outras abaixo dela, você deveria fazer a parte da exclusão da instrução CONNECT BY
- No exemplo abaixo, que exclui Higgins, estamos excluindo Gietz também

```
SELECT last_name  
FROM employees  
START WITH last_name = 'Kochhar'  
CONNECT BY PRIOR employee_id = manager_id  
AND last_name != 'Higgins';
```



Terminologia

- Estes são os principais termos usados nesta lição:
 - Connect By prior
 - Consultas hierárquicas
 - Level
 - Self join
 - Start with
 - Estrutura em Árvore
 - Tree Walking
 - Ramificações

Resumo

- Nesta lição, você deverá ter aprendido a:
 - Construir e executar uma instrução SELECT para juntar uma tabela a si mesma por meio da autojunção
 - Interpretar o conceito de uma consulta hierárquica
 - Criar um relatório estruturado em árvore
 - Formatar dados hierárquicos
 - Excluir ramificações da estrutura em árvore



ORACLE

Academy

