

SELECT

Parte 3

Exibindo Dados de Mais de Uma tabela
(JUNÇÃO)



Luciano Melo
profluciano.melo@fiap.com.br



Usando Apelidos de Tabelas

- Use prefixos de tabela para qualificar nomes de colunas presentes em várias tabelas.
- Use apelidos de coluna para distinguir as colunas com nomes idênticos, mas que residem em tabelas diferentes.
- Use apelidos de tabelas para simplificar consultas.
- Use apelidos de tabelas para melhorar o desempenho.

```
SELECT e.employee_id, e.last_name  
FROM   employees e;
```

Obtendo Dados de Várias Tabelas

EMPLOYEES

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
100	King	90
101	Kochhar	90
...		
202	Fay	20
205	Higgins	110
206	Gietz	110

DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID
10	Administration	1700
20	Marketing	1800
50	Shipping	1500
60	IT	1400
80	Sales	2500
90	Executive	1700
110	Accounting	1700
190	Contracting	1700



EMPLOYEE_ID	DEPARTMENT_ID	DEPARTMENT_NAME
200	10	Administration
201	20	Marketing
202	20	Marketing
...		
102	90	Executive
205	110	Accounting
206	110	Accounting

Tipos de Junções

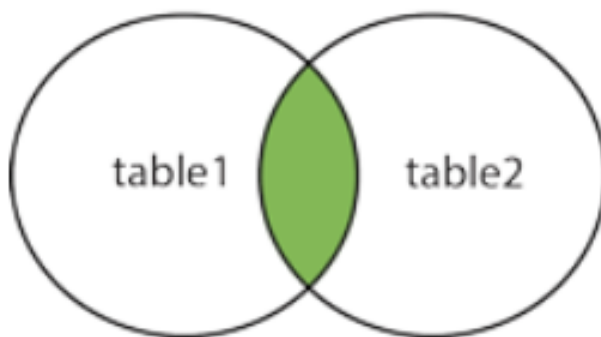
Quando se precisa projetar dados de mais de uma tabela, é necessário uma operação de junção entre as tabelas. Esta operação é conhecida como JOIN.

Tipos de Joins:

- Junções internas (INNER JOIN)
- Junções externas (OUTER JOIN)
- Produtos Cartesianos (CROSS JOIN)

Junções Internas (INNER JOIN)

Junções internas são usadas quando se deseja unir apenas as linhas que existem correspondência nas duas tabelas envolvidas na junção.



Portanto, uma junção interna é dada por uma condição – um filtro – que garanta que linhas das **duas tabelas podem ser “juntadas” por colunas, normalmente PK e FKs, que representam a mesma informação em ambas as tabelas.**

Matematicamente, Podemos dizer quem é representada por: $A \cap B$

Junção Interna (INNER JOIN)

- **Use a cláusula ON para especificar condições de junções a serem utilizadas em operações de join.**
- **A condição de join é separada de outras condições de pesquisa.**
- **A cláusula ON facilita a compreensão do código.**

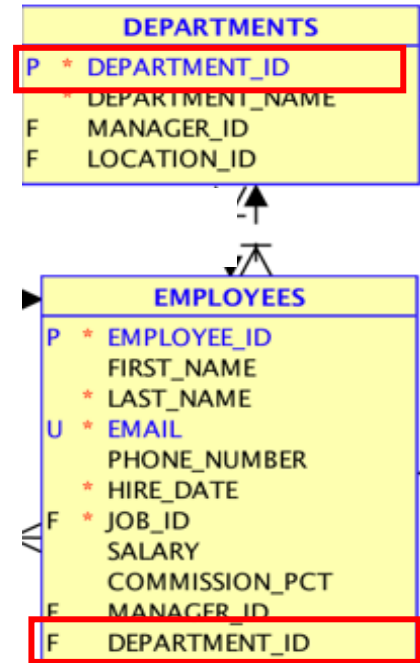
Junção com a cláusula ON

```
SELECT e.employee_id, e.last_name, e.department_id,  
       d.department_id, d.location_id  
FROM   employees e JOIN departments d  
ON     (e.department_id = d.department_id);
```

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_ID	LOCATION_ID
200	Whalen	10	10	1700
201	Hartstein	20	20	1800
202	Fay	20	20	1800
124	Mourgos	50	50	1500
141	Rajs	50	50	1500
142	Davies	50	50	1500
143	Matos	50	50	1500

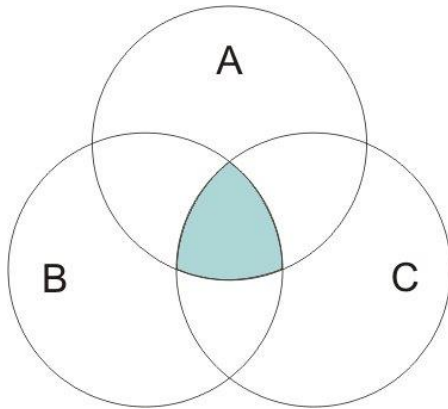
...

19 rows selected.

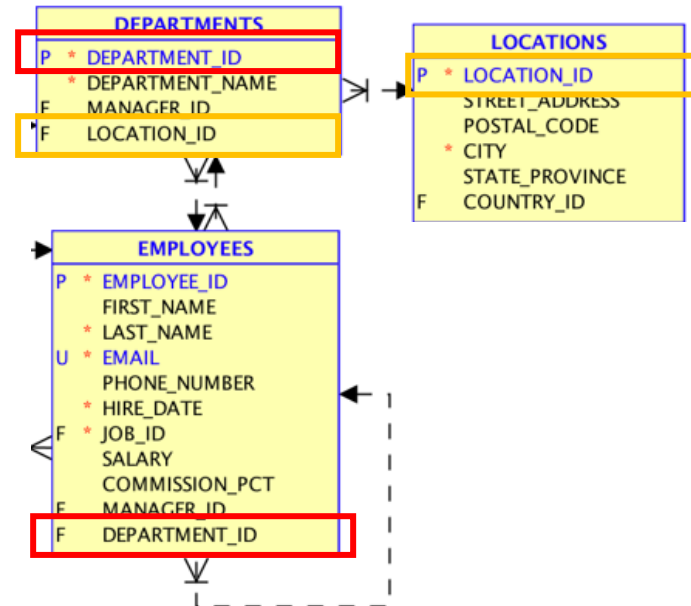


Junção com mais de duas tabelas

```
SELECT employee_id, city, department_name
FROM   employees e
JOIN   departments d
ON     d.department_id = e.department_id
JOIN   locations l
ON     d.location_id = l.location_id;
```



19 rows selected.



Auto-Joins (Self Join)

EMPLOYEES
employee_id
first_name
last_name
email
phone_number
hire_date
job_id
salary
commission_pct
manager_id
department_id

EMPLOYEES (WORKER)

EMPLOYEE_ID	LAST_NAME	MANAGER_ID
100	King	
101	Kochhar	100
102	De Haan	100
103	Hunold	102
104	Ernst	103
107	Lorentz	103
124	Mourgos	100

...

EMPLOYEES (MANAGER)

EMPLOYEE_ID	LAST_NAME
100	King
101	Kochhar
102	De Haan
103	Hunold
104	Ernst
107	Lorentz
124	Mourgos

...



MANAGER_ID na tabela “WORKER” é igual a EMPLOYEE_ID na tabela “MANAGER”.

Self Join - Exemplo

```
SELECT e.last_name emp, m.last_name mgr
FROM   employees e JOIN employees m
ON     (e.manager_id = m.employee_id);
```

EMP	MGR
Hartstein	King
Zlotkey	King
Mourgos	King
De Haan	King
Kochhar	King

...

19 rows selected.

Junções Externas (OUTER JOIN)

- Uma join entre duas tabelas que retorna os resultados da join interna, bem como as linhas não correspondentes da tabela esquerda (ou direita), é chamada de junção externa esquerda (ou direita).
- Uma join entre duas tabelas que retorna os resultados de uma join interna, bem como os resultados de uma join esquerda e direita, é uma junção externa integral.

Junções Externas (OUTER JOIN)

DEPARTMENTS

DEPARTMENT_NAME	DEPARTMENT_ID
Administration	10
Marketing	20
Shipping	50
IT	60
Sales	80
Executive	90
Accounting	110
Contracting	190

8 rows selected.

EMPLOYEES

DEPARTMENT_ID	LAST_NAME
90	King
90	Kochhar
90	De Haan
60	Hunold
60	Ernst
60	Lorentz
50	Mourgos
50	Rajs
50	Davies
50	Matos
50	Vargas
80	Zlotkey

...

20 rows selected.

Não há funcionários no departamento 190.

Junções Externas (OUTER JOIN)

DEPARTMENTS

DEPARTMENT_NAME	DEPARTMENT_ID
Administration	10
Marketing	20
Shipping	50
IT	60
Sales	80
Executive	90
Accounting	110
Contracting	190

8 rows selected.

EMPLOYEES

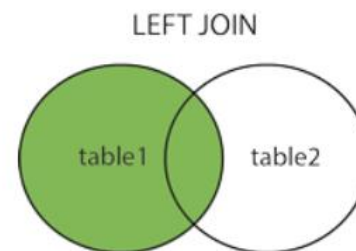
DEPARTMENT_ID	LAST_NAME
90	King
90	Kochhar
90	De Haan
60	Hunold
60	Ernst
60	Lorentz
50	Mourgos
50	Rajs
50	Davies
50	Matos
50	Vargas
80	Zlotkey

...

20 rows selected.

Não há funcionários no departamento 190.

Junções Externa a esquerda (*LEFT OUTER JOIN*)

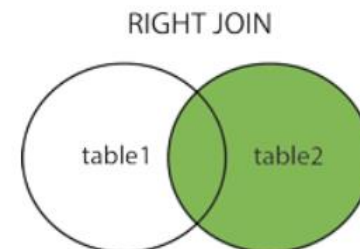


```
SELECT e.last_name, e.department_id, d.department_name
FROM   employees e LEFT OUTER JOIN departments d
ON     (e.department_id = d.department_id) ;
```

LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Whalen	10	Administration
Fay	20	Marketing
Hartstein	20	Marketing
...		
De Haan	90	Executive
Kochhar	90	Executive
King	90	Executive
Gietz	110	Accounting
Higgins	110	Accounting
Grant		

20 rows selected.

Junções Externa a direita (*RIGHT OUTER JOIN*)



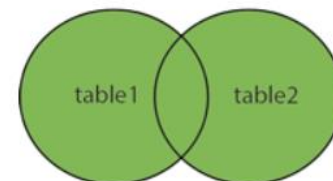
```
SELECT e.last_name, e.department_id, d.department_name
FROM   employees e RIGHT OUTER JOIN departments d
ON     (e.department_id = d.department_id) ;
```

LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Whalen	10	Administration
Fay	20	Marketing
Hartstein	20	Marketing
Davies	50	Shipping
...		
Kochhar	90	Executive
Gietz	110	Accounting
Higgins	110	Accounting
	190	Contracting

20 rows selected.

Junções Externa integral (*FULL OUTER JOIN*)

FULL OUTER JOIN



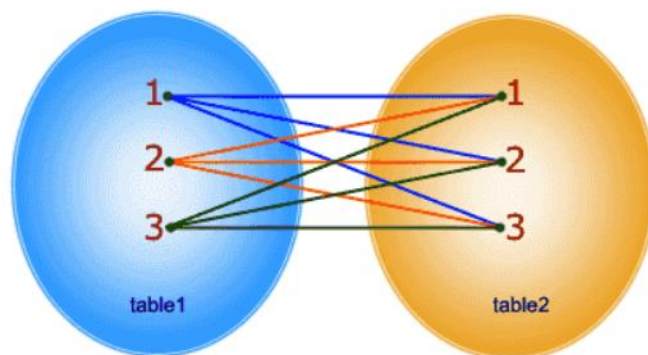
```
SELECT e.last_name, d.department_id, d.department_name
FROM   employees e FULL OUTER JOIN departments d
ON     (e.department_id = d.department_id) ;
```

LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Whalen	10	Administration
Fay	20	Marketing
Hartstein	20	Marketing
...		
King	90	Executive
Gietz	110	Accounting
Higgins	110	Accounting
Grant		
	190	Contracting

21 rows selected.

Produtos Cartesianos

- Um produto cartesiano será formado quando:
 - Uma condição de join for omitida
 - Uma condição de join for inválida
 - Todas as linhas da primeira tabela se unirem a todas as linhas da segunda tabela
- Para evitar um produto cartesiano, inclua sempre uma condição de join válida.



Gerando um produto cartesiano

EMPLOYEES (20 linhas)

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
100	King	90
101	Kochhar	90
...		
202	Fay	20
205	Higgins	110
206	Gietz	110

20 rows selected.

DEPARTMENTS (8 linhas)

DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID
10	Administration	1700
20	Marketing	1800
50	Shipping	1500
60	IT	1400
80	Sales	2500
90	Executive	1700
110	Accounting	1700
190	Contracting	1700

8 rows selected.

Produto cartesiano:
20 x 8 = 160 linhas

EMPLOYEE_ID	DEPARTMENT_ID	LOCATION_ID
100	90	1700
101	90	1700
102	90	1700
103	60	1700
104	60	1700
107	60	1700

...

160 rows selected.

Junções Cruzadas (Cross Join)

- A cláusula **CROSS JOIN** gera o produto cruzado de duas tabelas.
- Ele também é chamado de produto cartesiano entre as duas tabelas.

```
SELECT last_name, department_name  
FROM employees  
CROSS JOIN departments ;
```

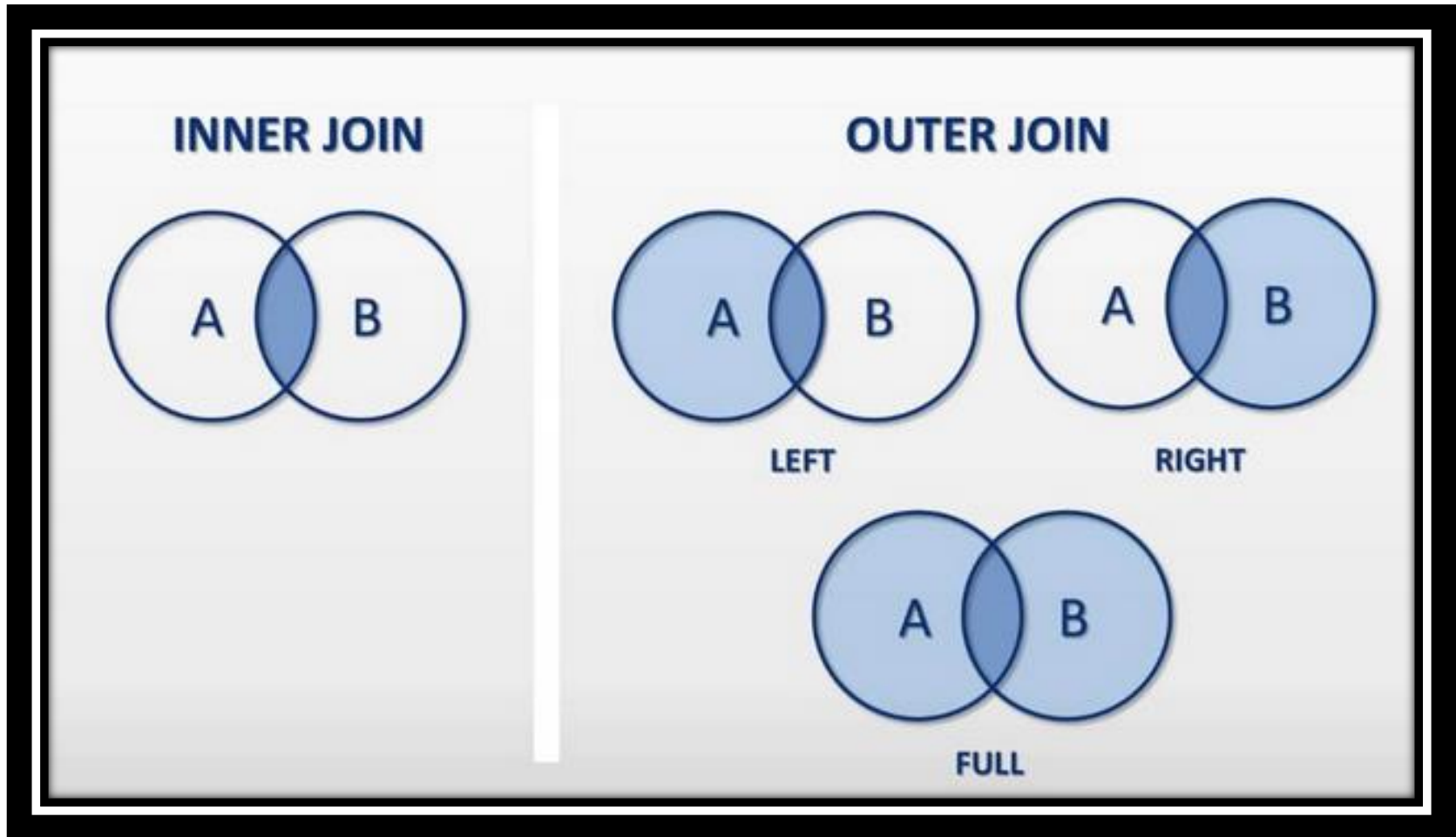
LAST_NAME	DEPARTMENT_NAME
King	Administration
Kochhar	Administration
De Haan	Administration
Hunold	Administration

...

160 rows selected.

JOIN

FIAP

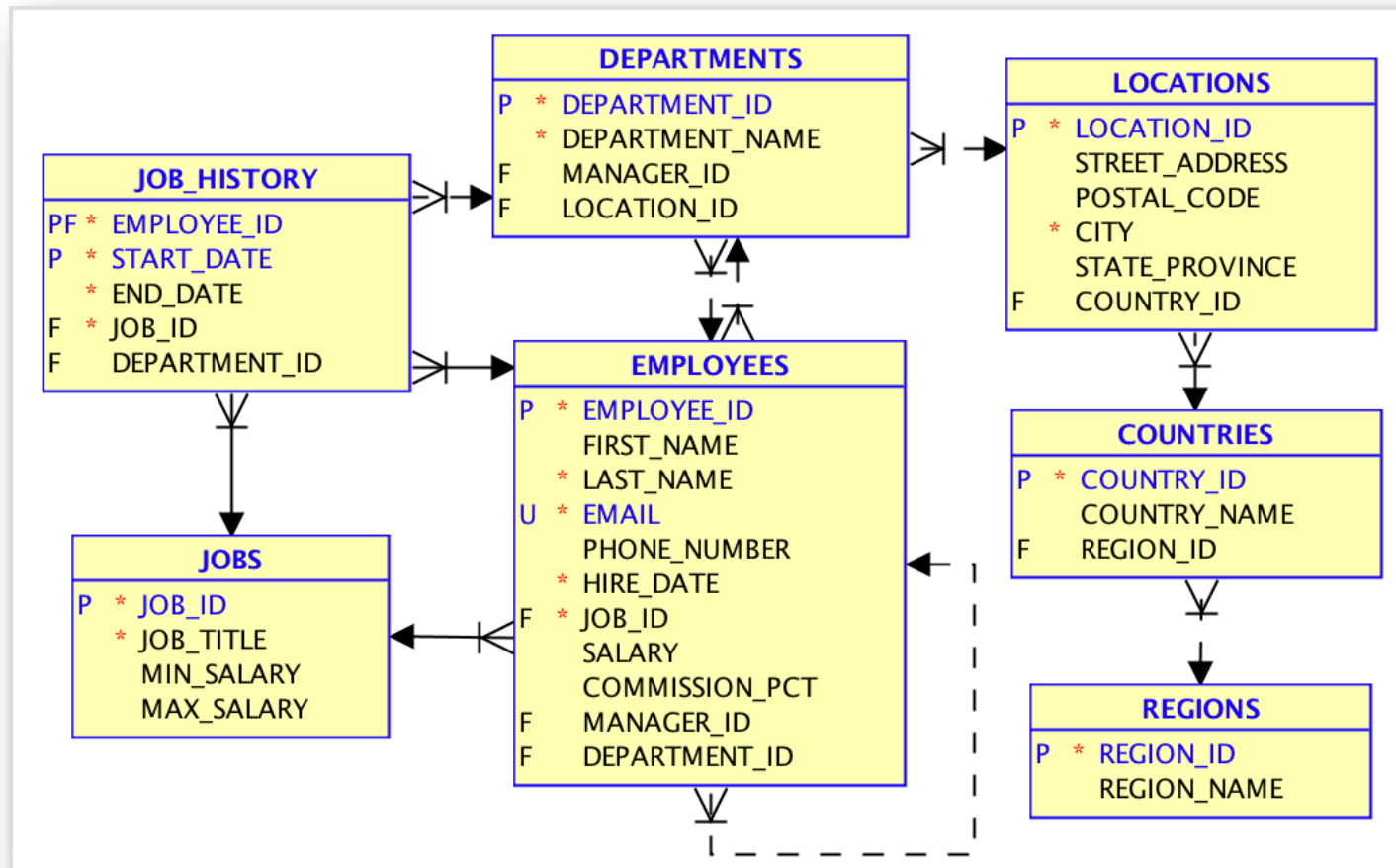




Hands on

profluciano.melo@fiap.com.br

Modelo de dados do schema de RH



Usar o comando SELECT para:



1. Mostrar o nome completo, o salário, o título da profissão e o nome do departamento de todos os empregados que trabalham em departamentos. Ordene o resultado de maneira alfabética pelo nome do empregado.
2. Alterar a consulta anterior para mostrar todos os empregados.
3. Mostrar o nome do departamento e o nome dos gerentes de cada departamento de todos os departamentos que tem empregados que comecem com a letra A.
4. Mostrar o primeiro nome do empregado, o nome do departamento e o nome da cidade em que o empregado trabalha. Todos os empregados devem aparecer
5. Listar as profissões que não tem nenhum empregado alocado na empresa?
6. Listar o nome das cidades que não tem nenhum empregado trabalhando
7. Mostrar o primeiro nome do empregado, o salário anual (sem comissão), e o nome da região do mundo que ele trabalha. Todos os empregados devem ser listados. Ordene o resultado pelo nome da região em ordem alfabética e para cada região os empregados que ganham os maiores salários primeiros.
8. Alterar a consulta anterior para mostra apenas os empregados com os 10 maiores salários
9. Mostrar o primeiro nome do empregado e o primeiro nome do gerente de cada empregado ordenados de maneira alfabética pelo primeiro nome do empregado.
10. Mostrar o nome do empregado, a data de admissão e quantos dias eles trabalham na empresa. Ordene de forma a mostrar os mais antigos primeiro.



profluciano.melo@fiap.com.br

Bibliografia Utilizada



Oracle SQL References: <http://docs.oracle.com>
Manuais Oracle – Introdução ao Oracle e SQL I e II

*Esta apresentação possui material de referência com propriedade da Oracle.
Copyright © 2004, Oracle. Todos os direitos reservados.*

