FACULDADE DAS AMÉRICAS

Banco de Dados

Exibindo Dados de Múltiplas Tabelas

Agenda

- O que é um JOIN
- Natural Join
- Simple Join – EquiJoin
- Alias
- SelfJoin
- NonEquijoins
- OuterJoin
  - Left Outer Join
  - Right Outer Join
  - Full Outer Join
- Cross Join
- Exercícios
- Bibliografia

FAM
FACULDADE DAS AMÉRICAS

Extraindo Dados de Mais de uma Tabela

Ligações (Joins)

Uma ligação é usada quando a pesquisa SQL requer dados de mais de uma tabela do Banco de Dados.

Linhas em uma tabela devem ser ligadas a outras linhas de acordo com o valor comum existente na coluna correspondente.
Existem três tipos principais de condições de ligações:

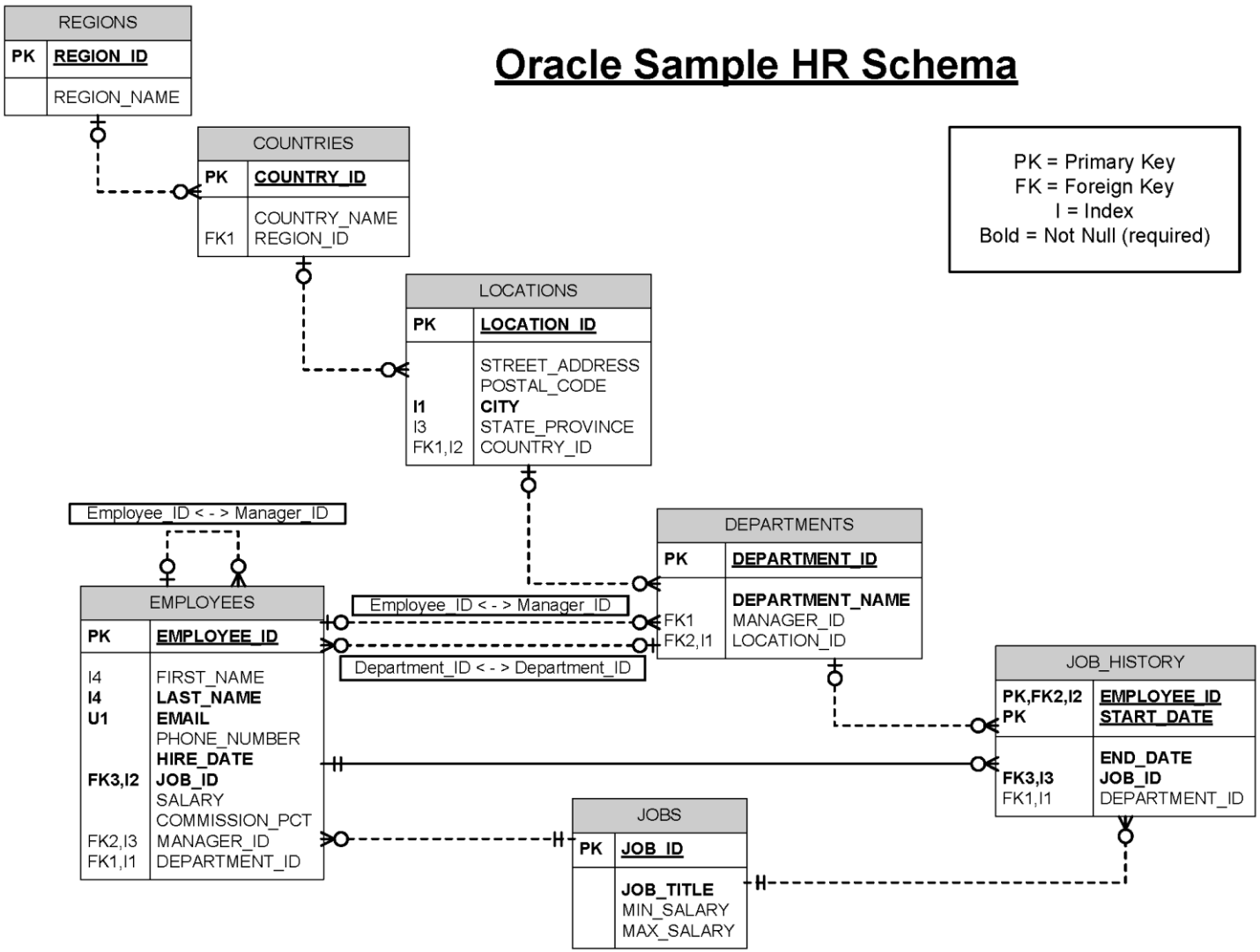
1) Equi-join
2) Non-equi-join
3) Outer-Join

Extraindo Dados de Mais de uma Tabela

Para as próximas atividades, iremos utilizar mais de uma tabela para extrair dados.

No próximo slide, está o DER das tabelas do usuário HR no Oracle 10g e seus relacionamentos.

Ligações


Oracle Sample HR Schema
REGIONS
PK REGION_ID
REGION_NAME
COUNTRIES
PK COUNTRY_ID
COUNTRY_NAME
FK1 REGION_ID
LOCATIONS
PK LOCATION_ID
STREET_ADDRESS
POSTAL_CODE
I1 CITY
I3 STATE_PROVINCE
FK1,I2 COUNTRY_ID
PK = Primary Key
FK = Foreign Key
I = Index
Bold = Not Null (required)
Employee_ID < - > Manager_ID
DEPARTMENTS
PK DEPARTMENT_ID
DEPARTMENT_NAME
FK1 MANAGER_ID
FK2,I1 LOCATION_ID
EMPLOYEES
PK EMPLOYEE_ID
I4 FIRST_NAME
I4 LAST_NAME
U1 EMAIL
PHONE_NUMBER
HIRE_DATE
FK3,I2 JOB_ID
SALARY
COMMISSION_PCT
FK2,I3 MANAGER_ID
FK1,I1 DEPARTMENT_ID
Employee_ID < - > Manager_ID
Department_ID < - > Department_ID
JOB_HISTORY
PK,FK2,I2 EMPLOYEE_ID
PK START_DATE
END_DATE
FK3,I3 JOB_ID
FK1,I1 DEPARTMENT_ID
JOBS
PK JOB_ID
JOB_TITLE
MIN_SALARY
MAX_SALARY

Natural Join

É baseada em todos os campos de mesmo nome em 2 tabelas.

Se os nomes dos campos e os tipos de dados forem iguais então retorna a consulta com sucesso;

Se os nomes forem iguais porém os tipos de dados forem diferentes, retorna erro.

Recuperando registros: Natural Join

```
SELECT department_id, department_name,
       location_id, city
FROM   departments
NATURAL JOIN locations ;
```

DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID	CITY
60	IT	1400	Southlake
50	Shipping	1500	South San Francisco
10	Administration	1700	Seattle
90	Executive	1700	Seattle
110	Accounting	1700	Seattle
190	Contracting	1700	Seattle
20	Marketing	1800	Toronto
80	Sales	2500	Oxford

8 rows selected.

```
SELECT  department_id, department_name,
           location_id, city
   FROM    departments
   NATURAL JOIN locations
   WHERE   department_id IN (20, 50);
```

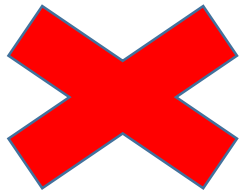
Natural Join + USING

Quando temos várias colunas com o mesmo nome e tipos diferentes, podemos escolher o campo de junção através do (USING).

Não use alias de tabela ou campo no using e no where, pois lembre-se: o campo deve ser comum às duas tabelas.

```
SELECT l.city, d.department_name
FROM   locations l JOIN departments d
USING (location_id)
WHERE  location_id = 1400;
```

```
SELECT l.city, d.department_name
FROM locations l JOIN departments d USING
(location_id)
WHERE d.location_id = 1400;
```



EquiJoin

Para descobrir, qual departamento os empregados estão, nós comparamos a coluna Department_ID da tabela Employees com o mesmo valor de Department_ID na tabela Departments. O relacionamento entre a tabela Employees e a Departments é um equi-join, em que o valor da coluna Department_ID seja igual para ambas as tabelas.

Uma condição de ligação é especificada na cláusula WHERE:

```
SELECT    coluna(s)
FROM      tabela(s)
WHERE     condição de ligação
```

Extraindo Dados de Mais de uma Tabela – Equi-Join

Para ligar as duas tabelas Employees e Departments, faça:

```
SELECT  Employees.Employee_ID,
        Employees.Department_ID, Department_Name
FROM    Employees, Departments
WHERE  Employees.Department_ID =Departments.Department_ID;
```

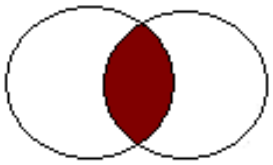
Ligação das tabelas (JOIN)

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



Inner Join

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

FAM FACULDADE DAS AMÉRICAS

Extraindo Dados de Mais de uma Tabela – Equi-Join

Para ligar as duas tabelas Employees e Departments utilizando JOIN, faça:

```
SELECT  Employees.Employee_ID,
        Employees.Department_ID, Department_Name
FROM    Employees JOIN Departments
ON      Employees.Department_ID =Departments.Department_ID;
```

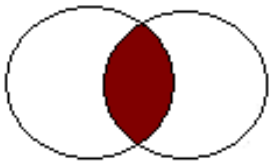
Ligação das tabelas (JOIN)

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



Inner Join

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

Utilizando Alias – "Apelidos"

- O 'Alias' também pode ser utilizado no comando select para apelidar tabelas conforme exemplo abaixo.
- O uso do alias simplifica a consulta;

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

FAM FACULDADE DAS AMÉRICAS

Extraindo Dados de Mais de uma Tabela - Equi-Join

Utilizando Join e condições juntas:

```
SELECT  E.First_Name, E.Last_Name, D.Department_Name
FROM    Employees E, Departments D
WHERE   E.Department_ID =D.Department_ID
        AND E.Department_ID = 30;
```

FIRST_NAME	LAST_NAME	DEPARTMENT_NAME
Den	Raphaely	Purchasing
Alexander	Khoo	Purchasing
Shelli	Baida	Purchasing
Sigal	Tobias	Purchasing
Guy	Himuro	Purchasing
Karen	Colmenares	Purchasing

Utilizando Join e condições separadas:

```
SELECT  E.First_Name, E.Last_Name, D.Department_Name
FROM    Employees E JOIN Departments D
ON      E.Department_ID =D.Department_ID
WHERE   E.Department_ID = 30;
```

FACULDADE DAS AMÉRICAS

Extraindo Dados de Mais de uma Tabela - Equi-Join

Caso não seja realizado o relacionamento, o SQL não está errado mas o resultado estará errado.

Portanto: na ausência da condição WHERE (relacionamento), cada linha da tabela Employees estará ligada com cada linha de Departments ...

Caso a tabela Employees tenha 14 registros e a tabela Departments tenha 4 linhas, será feito o produto cartesiano (14X4=56), e retornará 56 registros.

A EXIBIÇÃO DESSES DADOS ESTARÁ ERRADO!!!

Self Join

EMPLOYEES (WORKER)

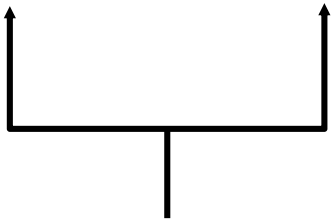
EMPLOYEE_ID	LAST_NAME	MANAGER_ID
100	King	
101	Kochhar	100
102	De Haan	100
103	Hunold	102
104	Ernst	103
107	Lorentz	103
124	Mourgos	100

...

EMPLOYEES (MANAGER)

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

...



MANAGER_ID da tabela WORKER é igual ao EMPLOYEE_ID da tabela MANAGER.

Self Join

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

Ou

```
SELECT e.last_name emp, m.last_name mgr
FROM   employees e, employees m
WHERE e.manager_id = m.employee_id;
```

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

...

19 rows selected.

Join entre três 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;
```

Ou

```
SELECT employee_id, city, department_name
FROM   employees e, departments d, locations l
WHERE  d.department_id = e.department_id AND
       d.location id = l.location id;
```

EMPLOYEE_ID	CITY	DEPARTMENT_NAME
103	Southlake	IT
104	Southlake	IT
107	Southlake	IT
124	South San Francisco	Shipping
141	South San Francisco	Shipping
142	South San Francisco	Shipping
143	South San Francisco	Shipping
144	South San Francisco	Shipping

19 rows selected.

NonEquijoins - [rodar script job_grades.sql]

FACULDADE DAS AMÉRICAS

EMPLOYEES

LAST_NAME	SALARY
King	24000
Kochhar	17000
De Haan	17000
Hunold	9000
Ernst	6000
Lorentz	4200
Mourgos	5800
Rajs	3500
Davies	3100
Matos	2600
Vargas	2500
Zlotkey	10500
Abel	11000
Taylor	8600

...

20 rows selected.

JOB_GRADES

GRA	LOWEST_SAL	HIGHEST_SAL
A	1000	2999
B	3000	5999
C	6000	9999
D	10000	14999
E	15000	24999
F	25000	40000

O salário dos empregados deve estar compreendido entre o menor e o maior salário da tabela JOB_GRADES.

Recuperando Registros com NonEquijoins

```
SELECT e.last_name, e.salary, j.grade_level
FROM   employees e JOIN job_grades j
ON     e.salary
       BETWEEN j.lowest_sal AND j.highest_sal;
```

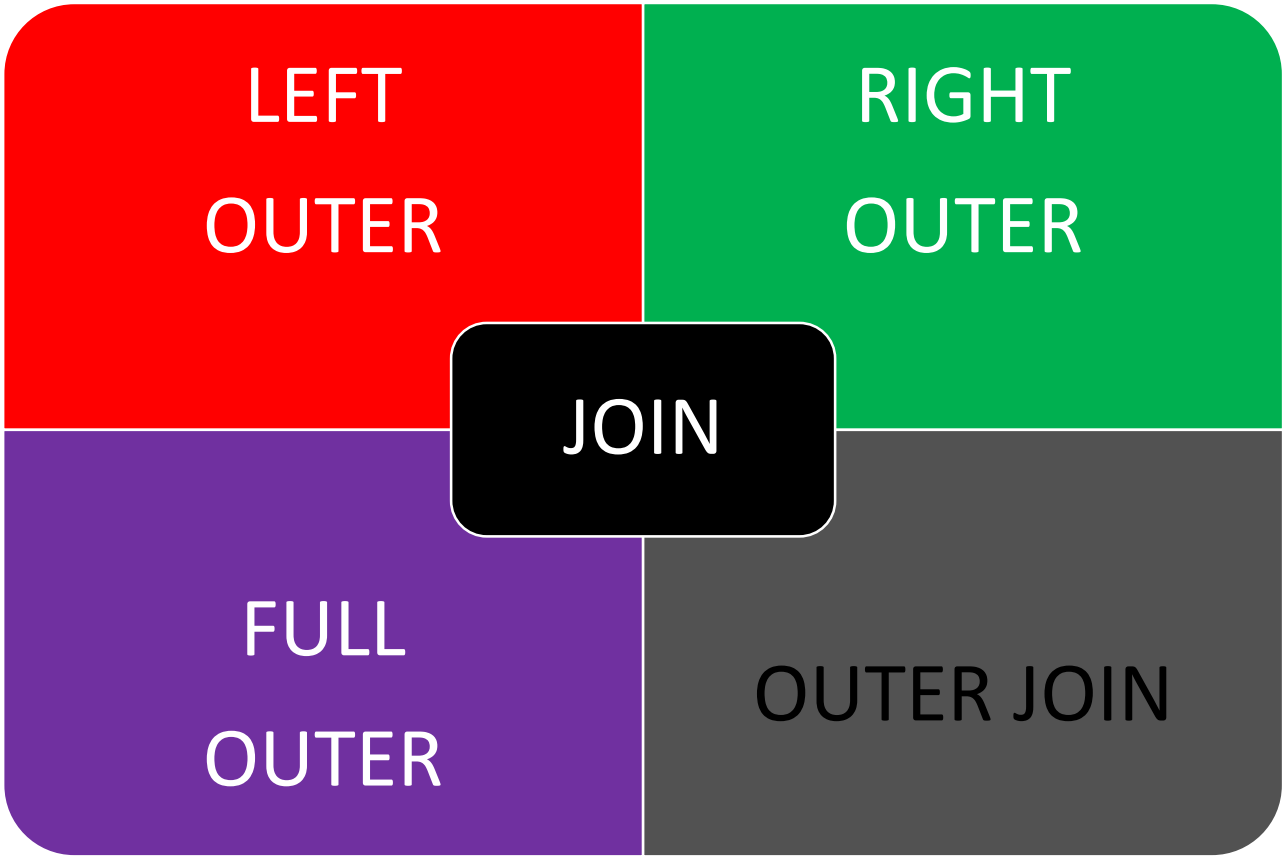
Ou

```
SELECT e.last_name, e.salary, j.grade_level
FROM   employees e, job_grades j
WHERE  e.salary BETWEEN j.lowest_sal AND
       j.highest_sal;
```

LAST_NAME	SALARY	GRA
Matos	2600	A
Vargas	2500	A
Lorentz	4200	B
Mourgos	5800	B
Rajs	3500	B
Davies	3100	B
Whalen	4400	B
Hunold	9000	C
Ernst	6000	C

...

20 rows selected.


LEFT
OUTER
RIGHT
OUTER
JOIN
FULL
OUTER
OUTER JOIN

Outer Join

Além de mostrar registros cujos campos em comum estejam presentes nas duas tabelas, ainda mostram os que faltam.

- Neste caso, a tabela DEPARTMENTS possui o departamento de Contracting que não possui nenhum EMPLOYEES alocado.
- No caso de uma consulta na tabela de EMPLOYEES o departamento 190 é o único que não aparecerá.

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

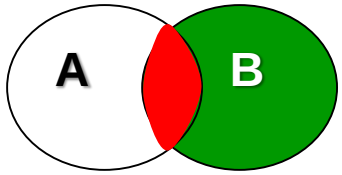
DEPARTMENT_ID	LAST_NAME
90	King
90	Kochhar
90	De Haan
60	Hunold
60	Ernst
60	Lorentz
50	Mourgos
50	Rajs
50	Davies
50	Matos
50	Vargas
80	Zlotkey

...

20 rows selected.

Não há empregados no departamento 190.

Right Outer Join



Right Outer Join

- Apresenta os departamentos que tem e que não tem funcionários.

```
SELECT e.last_name, d.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.

Right Outer Join (+)

FACULDADE DAS AMÉRICAS

Usando o Operador (+)

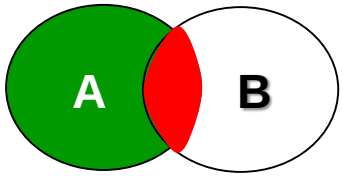
O símbolo (+) é o operador para junção externa , ou seja, se um dado existir somente em uma tabela, com este símbolo será possível retornar linha(s) faltando o outro valor.

O símbolo (+) deve ser inserido junto ao nome da tabela que não tem linhas correspondentes.

```
SELECT  E.First_Name, E.Last_Name,
        D.Department_Name
FROM    Employees E, Departments D
WHERE   E.Department_ID (+) =
        D.Department_ID
ORDER BY D.Department_Name;
```

FIRST_NAME	LAST_NAME	DEPARTMENT_NAME
William	Gietz	Accounting
Shelley	Higgins	Accounting
Jennifer	Whalen	Administration
-	-	Benefits
-	-	Construction
-	-	Contracting
-	-	Control And Credit
-	-	Corporate Tax
Lex	De Haan	Executive
Neena	Kochhar	Executive

Left Outer Join



Left Outer Join

- Apresenta os empregados que tem e que não tem departamentos.

```
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.

Left Outer Join

Usando o Operador (+)

```
SELECT  E.First_Name, E.Last_Name,  D.Department_Name
FROM    Employees E, Departments D
WHERE  E.Department_ID = D.Department_ID (+)
ORDER BY D.Department_Name
```

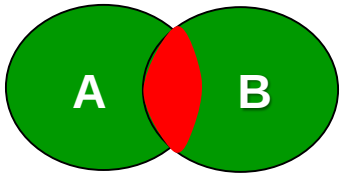
FIRST_NAME	LAST_NAME	DEPARTMENT_NAME
Shelley	Higgins	Accounting
William	Gietz	Accounting
Jennifer	Whalen	Administration
Steven	King	Executive
Neena	Kochhar	Executive
Lex	De Haan	Executive
Daniel	Faviet	Finance

■ ■ ■

Ki	Gee	Shipping
Hazel	Philtanker	Shipping
Renske	Ladwig	Shipping
Stephen	Stiles	Shipping
John	Seo	Shipping
Joshua	Patel	Shipping
Trenna	Rajs	Shipping
Curtis	Davies	Shipping
Randall	Matos	Shipping
Jean	Fleaur	Shipping
Kimberely	Grant	-

106 linhas retornadas em 0,00 segundos Exportação

Full Outer Join



Full Outer Join

- Retorna um Right e um Left Join ao mesmo tempo.

```
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.

Produto Cartesiano

EMPLOYEES (20 rows)

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 rows)

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 rows

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.

Cross Join

- O Cross Join retorna o produto cartesiano de 2 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.

FACULDADE DAS AMÉRICAS

Resumo do comando SELECT

```
SELECT    [DISTINCT] coluna(s), expr, alias...
FROM      tabelas [alias]...
WHERE     [condição de ligação]...
AND       [condição de linha]...
OR        [outras condições de linhas]..
GROUP BY  [expr/coluna]
HAVING    [grupo de condições]
ORDER BY  [expr/coluna] [DESC/ASC]
```

Exercícios

Vide Arquivo Exercício 12.doc em anexo

Referências Bibliográficas

[1] Fanderuff, Damaris. Dominando o Oracle 9i: Modelagem e desenvolvimento. São Paulo:Pearson Education do Brasil, 2003.

[2] Costa, Rogério Luis de C., SQL : guia prático. 2. ed. Rio de Janeiro : Brasport, 2006.

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

[4] Morelli, Eduardo M. Terra, 1996. Oracle 9i Fundamental: Sql, Pl/SQL e Administração. São Paulo: Érica, 2002.