

A large, light gray, stylized number '5' is centered in the background of the slide.

Consultado Múltiplas Tabelas JOIN e UNION

Objetivos

- Tipos de JOINS
- JOIN tradicional : INNER JOIN
 - LEFT OUTER JOIN
 - RIGHT OUTER JOIN
 - CROSS JOIN
- Table Alias
- Diferenças de sintaxes
- UNION e UNION ALL

JOIN

- O JOIN (junção) é utilizado para acessar e combinar dados de múltiplas tabelas
- INNER JOIN (combinações perfeitas)
- LEFT OUTER JOIN (todas linhas da tab. esquerda)
- RIGHT OUTER JOIN (todas linhas da tab. direita)
- FULL OUTER JOIN (todas as linhas de ambas)
- CROSS JOIN (produto cartesiano)

JOIN



Left outer join



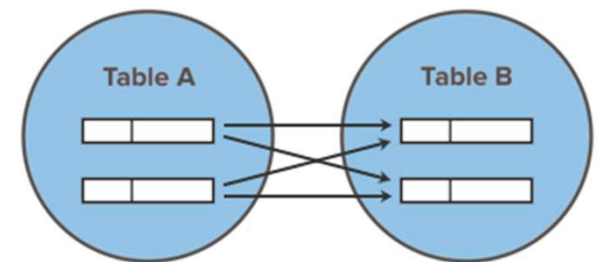
Inner join



Right outer join



Full outer join

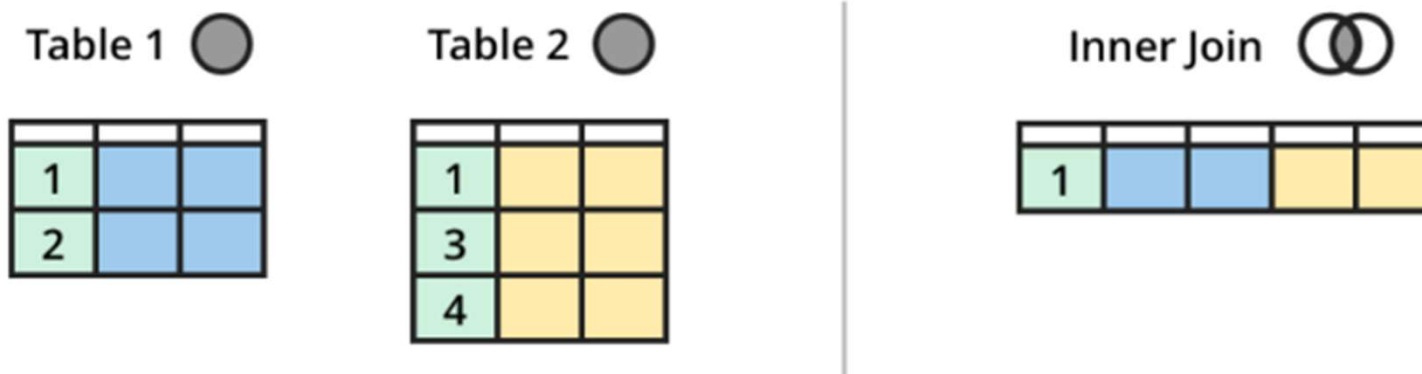


SQL CROSS JOIN

INNER JOIN

- INNER JOIN

- Considerando a “chave” de junção
traz apenas as linhas que existem em ambas as tabelas
- Combinações perfeitas




INNER JOIN

| Checks | | | |
|--------|-----------|-----------|-------------|
| Check | Vendor ID | Date | Amount |
| 1 | B | 4/11/2008 | \$ 451.58 |
| 2 | D | 4/14/2008 | \$ 4,483.99 |
| 3 | B | 4/15/2008 | \$ 848.48 |
| 4 | A | 4/18/2008 | \$ 8,564.99 |
| 5 | E | 4/19/2008 | \$ 1,941.80 |

| Vendors | |
|-----------|--------------|
| Vendor ID | Name |
| A | Adams Corp. |
| B | Blette, Inc. |
| C | Carlson Co. |
| E | ERT Corp. |
| F | Franks, Inc. |

| Check | Vendor ID | Name | Date | Amount |
|-------|-----------|--------------|-----------|------------|
| 1 | B | Blette, Inc. | 4/11/2008 | \$ 451.58 |
| 3 | B | Blette, Inc. | 4/15/2008 | \$ 848.48 |
| 4 | A | Adams Corp. | 4/18/2008 | \$8,564.99 |
| 5 | E | ERT Corp. | 4/19/2008 | \$1,941.80 |

INNER JOIN



| Checks | | | |
|--------|-----------|-----------|-------------|
| Check | Vendor ID | Date | Amount |
| 1 | B | 4/11/2008 | \$ 451.58 |
| 2 | D | 4/14/2008 | \$ 4,483.99 |
| 3 | B | 4/15/2008 | \$ 848.48 |
| 4 | A | 4/18/2008 | \$ 8,564.99 |
| 5 | E | 4/19/2008 | \$ 1,941.80 |

| Vendors | |
|-----------|--------------|
| Vendor ID | Name |
| A | Adams Corp. |
| B | Blette, Inc. |
| C | Carlson Co. |
| E | ERT Corp. |
| F | Franks, Inc. |

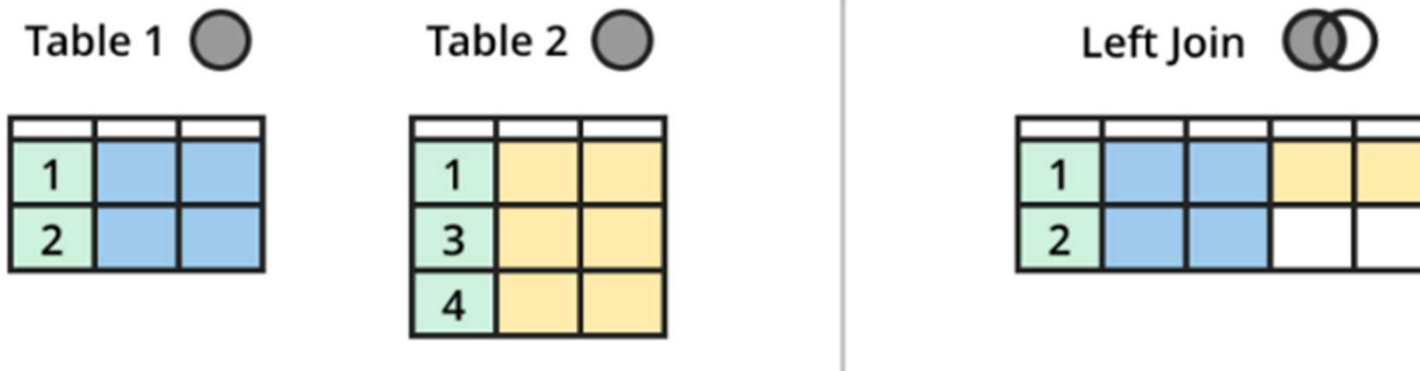
| Check | Vendor ID | Name | Date | Amount |
|-------|-----------|--------------|-----------|------------|
| 1 | B | Blette, Inc. | 4/11/2008 | \$ 451.58 |
| 3 | B | Blette, Inc. | 4/15/2008 | \$ 848.48 |
| 4 | A | Adams Corp. | 4/18/2008 | \$8,564.99 |
| 5 | E | ERT Corp. | 4/19/2008 | \$1,941.80 |

- Atenção: apenas linhas com correspondências formam o resultset !

LEFT OUTER JOIN

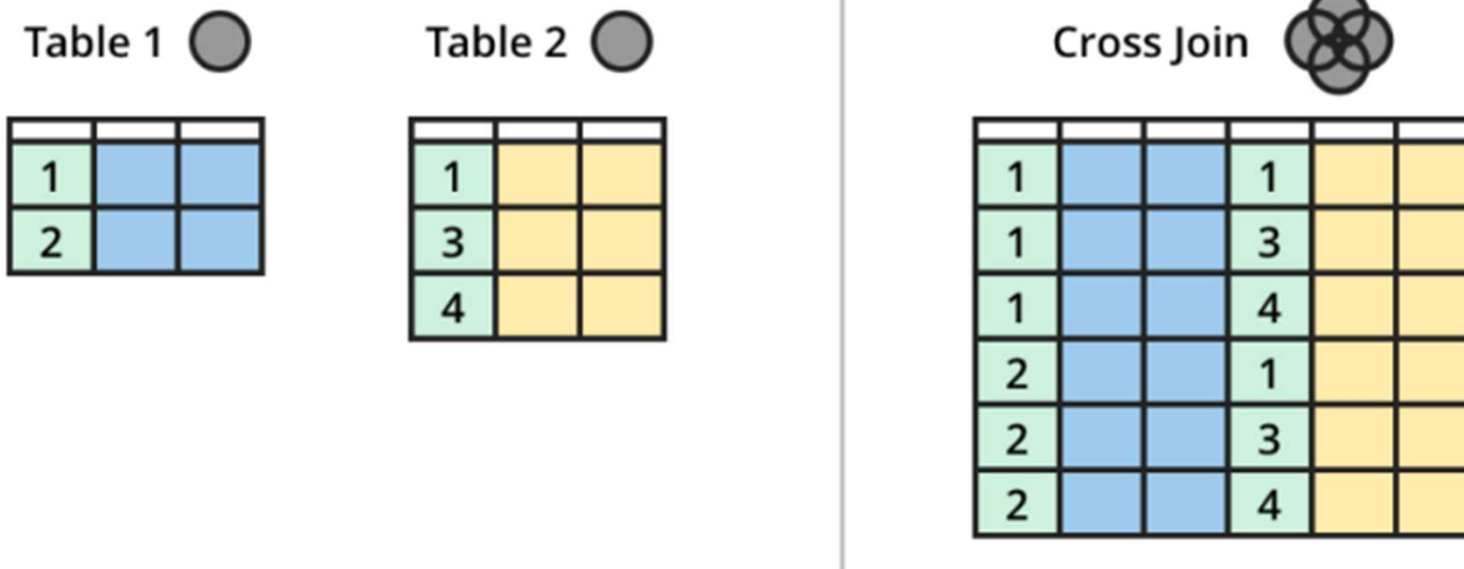
- LEFT OUTER JOIN

- Considerando a “chave” de junção
traz todas as linhas da tabela à esquerda,
e os dados das linhas com correspondência na outra tabela



CROSS JOIN

- CROSS JOIN
 - Produto cartesiano
 - Multiplicação de todas as linhas das duas tabelas



JOIN

- Como é a sintaxe?

```
select  col1, col2, col3, tabela1.colX, ...
from    tabela1
        <JOIN_TYPE> tabela2 on (tabela1.colX = tabela2.colX);
```

- Caso tenha colunas de nomes repetidos entre as tabs precisa identificar!

```
select  col1, col2, col3, tabela1.colX, ...
from    tabela1
        <JOIN_TYPE> tabela2 on (tabela1.colX = tabela2.colX);
```

JOIN

- Caso não identifique as colunas repetidas: ERRO!

```
select Check, Vendor_id, Name, Date, Amount
from Checks
inner join Vendors on (Vendors.Vendor_ID = Checks.Vendor_ID);
```

```
select Employee_id, Department_id, Department_name, hire_date, salary
from hr.Employees
inner join hr.Departments on (Departments.Department_id = Employees.Department_id);
```

ORA-00918: column ambiguously defined

ALIAS para as tabelas!

- Colunas com mesmo nome em +1 tabela
 - Identificar a tabela
 - Usar aliases para as tabelas (legibilidade + performance)

```
select Employee_id, Employees.Department_id, Department_name, hire_date, salary
from hr.Employees
inner join hr.Departments on
        (Departments.Department_id = Employees.Department_id)
```

```
select e.Employee_id, e.Department_id, d.Department_name, e.hire_date, e.salary
from hr.Employees e
inner join hr.Departments d on (d.Department_id = e.Department_id);
```

JOIN – Sintaxe SQL ANSI vs. Oracle

- Sintaxe Oracle:
 - Todas as tabelas no from, separadas por virgulas
 - Condições de junções todas no where...
 - Ambas tem o mesmo efeito e resultado

* sintaxe SQL ANSI

```
select e.Employee_id, e.Department_id, d.Department_name, e.hire_date, e.salary
from   hr.Employees e
       inner join hr.Departments d on (d.Department_id = e.Department_id);
```

* sintaxe ORACLE

```
select e.Employee_id, e.Department_id, d.Department_name, e.hire_date, e.salary
from   hr.Employees e, hr.Departments d
where  d.Department_id = e.Department_id;
```

JOIN – Sintaxe SQL ANSI vs. Oracle

- Sintaxe Oracle:
 - Todas as tabelas no from, separadas por virgulas
 - Condições de junções todas no where...
 - Em OUTER JOINS usar (+) à direita da coluna da tabela “opcional”

* sintaxe SQL ANSI

```
select e.Employee_id, e.Department_id, d.Department_name, e.hire_date, e.salary
from   hr.Employees e
       LEFT OUTER join hr.Departments d on (d.Department_id = e.Department_id);
```

* sintaxe ORACLE

```
select e.Employee_id, e.Department_id, d.Department_name, e.hire_date, e.salary
from   hr.Employees e, hr.Departments d
where  d.Department_id(+) = e.Department_id;
```

UNION: junção de 2 consultas em 1 resultset

```
1 v select Employee_id, Department_id, first_name,  
2      salary, salary*1.1 as salary_aum  
3 from   hr.Employees  
4 where  department_id = 40;
```

| EMPLOYEE_ID | DEPARTMENT_ID | FIRST_NAME | SALARY | SALARY_AUM |
|-------------|---------------|------------|--------|------------|
| 203 | 40 | Susan | 6500 | 7150 |

```
1 v select Employee_id, Department_id, first_name,  
2      salary, salary*1.25 as salary_aum  
3 from   hr.Employees  
4 where  department_id = 90;
```

| EMPLOYEE_ID | DEPARTMENT_ID | FIRST_NAME | SALARY | SALARY_AUM |
|-------------|---------------|------------|--------|------------|
| 100 | 90 | Steven | 24000 | 30000 |
| 101 | 90 | Neena | 17000 | 21250 |
| 102 | 90 | Lex | 17000 | 21250 |

UNION: pode alterar posição

```
1 v select Employee_id, Department_id, first_name,  
2       salary, salary*1.1 as salary_aum  
3 from   hr.Employees  
4 where  department_id = 40  
5 UNION  
6 select Employee_id, Department_id, first_name,  
7       salary, salary*1.25 as salary_aum  
8 from   hr.Employees  
9 where  department_id = 90;
```

| EMPLOYEE_ID | DEPARTMENT_ID | FIRST_NAME | SALARY | SALARY_AUM |
|-------------|---------------|------------|--------|------------|
| 100 | 90 | Steven | 24000 | 30000 |
| 101 | 90 | Neena | 17000 | 21250 |
| 102 | 90 | Lex | 17000 | 21250 |
| 203 | 40 | Susan | 6500 | 7150 |

Selecionar **Valores fixos** e unir seleções

- Quantos UNIONS forem necessários ...

```
1 v select 'Sul' as Regiao, 'PR' as Sigla, 'Paraná' as Estado from dual
2 UNION ALL
3 select 'Sul' as Regiao, 'SC' as Sigla, 'Santa Catarina' as Estado from dual
4 UNION ALL
5 select 'Sul' as Regiao, 'RS' as Sigla, 'Rio Grande do Sul' as Estado from dual;
```

| REGIAO | SIGLA | ESTADO |
|--------|-------|-------------------|
| Sul | PR | Paraná |
| Sul | SC | Santa Catarina |
| Sul | RS | Rio Grande do Sul |

Selecionar **Valores fixos** e unir seleções

- Basta alias nas colunas do 1o. select

```
1 v select 'Sul' as Regiao, 'PR' as Sigla, 'Paraná' as Estado from dual
2 UNION ALL
3 select 'Sul' as Reg, 'SC' as Sig, 'Santa Catarina' as Est from dual
4 UNION ALL
5 select 'Sul', 'RS', 'Rio Grande do Sul' from dual;
```

| REGIAO | SIGLA | ESTADO |
|--------|-------|-------------------|
| Sul | PR | Paraná |
| Sul | SC | Santa Catarina |
| Sul | RS | Rio Grande do Sul |

Resumo e Dúvidas

- Dúvidas ou comentários ... ?

