

MBA em IA e Big Data



Curso 01 - Linguagens e Ferramentas para Inteligência Artificial e Big Data (Python e SQL)

Consultas avançadas com SQL - Junções e agregações

Jose Fernando Rodrigues Junior
ICMC-USP São Carlos

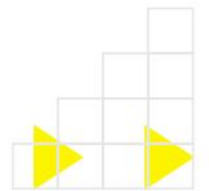
Objetivo: prática com conceitos de junção, agregação, ordenação, operações com conjuntos, e consultas aninhadas





Data Manipulation Language (DML)

SELECT

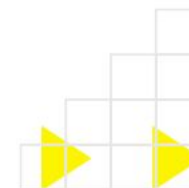




Comandos DML

- **SELECT** – comando de consulta
 - retorno \Rightarrow tabela resultado (**multiconjunto – potencialmente um conjunto com repetições**)

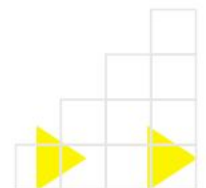
```
SELECT [DISTINCT|ALL] <lista de atributos>  
FROM <lista de tabelas>  
[WHERE <condições>]  
[GROUP BY atributo]  
[HAVING <condições>]  
[ORDER BY atributo [ASC|DESC]]
```



JUNÇÃO

- Cláusula **FROM** com mais de uma tabela
 - **Junção interna** (*Inner Join*)
 - **WHERE** \Rightarrow condição de junção
 - em geral: atributos com relacionamento PK - FK

```
SELECT [DISTINCT|ALL] <atributos>  
FROM tabela1, tabela2  
WHERE tabela1.atributo1 =  
       tabela2.atributo2
```



Exemplo: Junção Interna

Aluno = {Nome, NUSP}

{<Zeca, 11111>,
<Zico, 22222>,
<Juca, 33333>,
<Tuca, 44444> }

Matricula= {Sigla, Numero, Aluno, Ano}

{<SCC-125, 1, 11111, 2010>,
<SCC-148, 1, 11111, 2010>,
<SCC-125, 2, 22222, 2010>,
<SCC-148, 1, 22222, 2009>}

```
select A.nome, A.nusp, M.Sigla
from Aluno A, Matricula M
where A.nusp = M.aluno
```

{Nome, NUSP, Sigla}
{<Zeca, 11111, SCC-125>,
<Zeca, 11111, SCC-148>,
<Zico, 22222, SCC-125>,
<Zico, 22222, SCC-148 >}

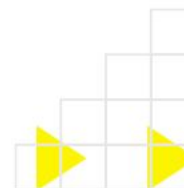
Aluno = {Nome, Nusp, Idade, DataNasc}

Professor = {Nome, NFunc, Idade, Titulação}

Disciplina = {Sigla, Nome, NCred, Professor, Livro}

Turma = {Sigla, Numero, NAlunos}

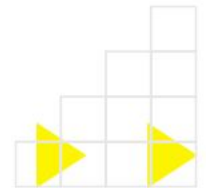
Matrícula = {Sigla, Numero, Aluno, Ano, Nota}



Inner Join

```
SELECT [DISTINCT|ALL] <atributos>  
FROM tabela1 T1  
[INNER] JOIN tabela2 T2  
ON T1.atributo1 = T2.atributo2
```

Exemplo: select A.nome, A.nusp, M.Sigla
from Aluno A JOIN Matricula M
ON A.nusp = M.aluno



Junção Interna

SELECT <atributos>

FROM tabela1 T1 , tabela2 T2

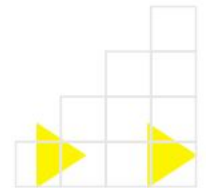
WHERE T1.atributo1 = T2.atributo2



SELECT <atributos>

FROM tabela1 T1 JOIN tabela2 T2

ON T1.atributo1 = T2.atributo2



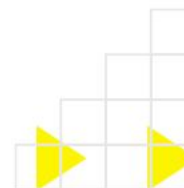
Junções Externas



```
SELECT [DISTINCT|ALL] <atributos>  
FROM tabela1 T1  
[LEFT | RIGHT | FULL] JOIN tabela2 T2  
ON T1.atributo1 = T2.atributo2
```

```
SELECT [DISTINCT|ALL] <atributos>  
FROM tabela1 T1, tabela2 T2  
WHERE T1.atributo1[(+)] = T2.atributo2[(+)]
```

Exemplo: SELECT *
FROM Professor P [LEFT|RIGHT|FULL] Disciplina D
ON P.NFunc = D.Professor



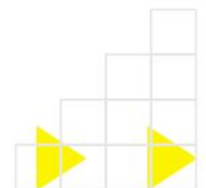
Junções Externas, notação Oracle+

- *LEFT JOIN COM (+)*

```
SELECT [DISTINCT|ALL] <atributos>  
      FROM tabela1 T1, tabela2 T2  
      WHERE T1.atributo1 = T2.atributo2 (+)
```

- *RIGHT JOIN COM (+)*

```
SELECT [DISTINCT|ALL] <atributos>  
      FROM tabela1 T1, tabela2 T2  
      WHERE T1.atributo1 (+) = T2.atributo2
```



Junções Externas

- **LEFT JOIN** (*outer*)

SELECT [

FROM

WHERE

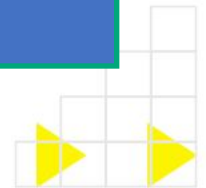
- **RIGHT JOIN** (*outer*)

SELECT [

FROM

WHERE *T1.atributo1 (+) = T2.atributo2*

O (+) não é interpretado de acordo com qual lado ele está (left ou right); ele apenas indica de qual tabela serão aceitos valores null.

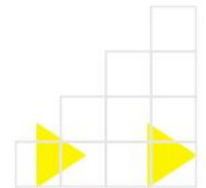


Exemplo: Junção Externa

Aluno = {Nome, <u>NUSP</u> }	Matricula= { <u>Sigla</u> , <u>Numero</u> , <u>Aluno</u> , <u>Ano</u> }
{<Zeca, 11111>, <Zico, 22222>, <Juca, 33333>, <Tuca, 44444> }	{<SCC-125, 1, 11111, 2010>, <SCC-148, 1, 11111, 2010>, <SCC-125, 2, 22222, 2010>, <SCC-148, 1, 22222, 2009>}

```
select A.nome, A.nusp, M.Sigla
from Aluno A left join Matricula M
where A.nusp = M.aluno
```

```
{Nome, NUSP, Sigla}
{<Zeca, 11111, SCC-125>,  
<Zeca, 11111, SCC-148>,  
<Zico, 22222, SCC-125>,  
<Zico, 22222, SCC-148>,  
<Juca, 33333, NULL >,  
<Tuca, 44444, NULL>}
```



Exemplo:

Aluno = {Nome, Nusp, Idade, DataNasc}

Professor = {Nome, NFunc, Idade, Titulação}

Disciplina = {Sigla, Nome, NCred, Professor, Livro}

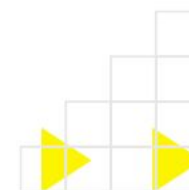
Turma = {Sigla, Numero, NAlunos}

Matrícula = {Sigla, Numero, Aluno, Ano, Nota}



- Selecionar nome e nro funcional dos professores DOUTORES que ministram ou não ministram disciplinas.

```
select P.Nome, P.NFunc, D.Sigla
  from Disciplina D right join Professor P
    on D.Professor = P.NFunc
 where UPPER(P.Titulacao) = UPPER('doutor')
```



INNER JOIN

1
2
3

INNER
JOIN

A
B
C

=

1 B
2 A

Only returns rows that meet the join condition

RIGHT OUTER JOIN

1
2
3

RIGHT
OUTER
JOIN

A
B
C

=

1 B
2 A
C

Returns all rows from the table on the right side of JOIN and matched rows from the left side of the JOIN

LEFT OUTER JOIN

1
2
3

LEFT
OUTER
JOIN

A
B
C

=

1 B
2 A
3

Returns all rows from the table on the left side of JOIN and matched rows from the right side of the JOIN

FULL OUTER JOIN

1
2
3

FULL
OUTER
JOIN

A
B
C

=

1 B
2 A
3
C

Returns all rows from both sides even if join condition is not met

CROSS JOIN

1
2
3

CROSS
JOIN

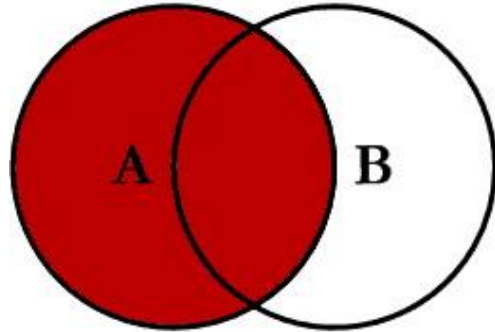
A
B
C

=

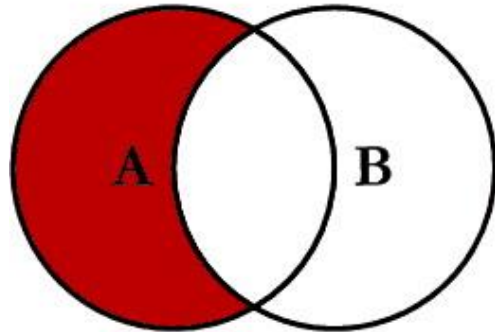
1 A
1 B
1 C
2 A
2 B
2 C
3 A
3 B
3 C

Cartesian product between the two sides is a join but without a join condition. Returns all rows joined from both sides

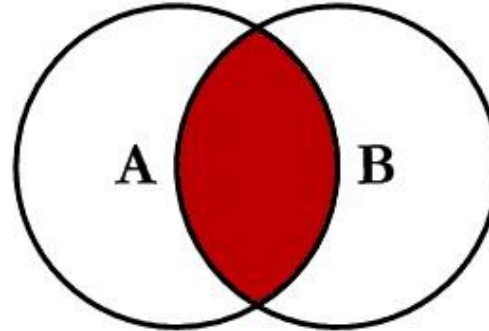
SQL JOINS



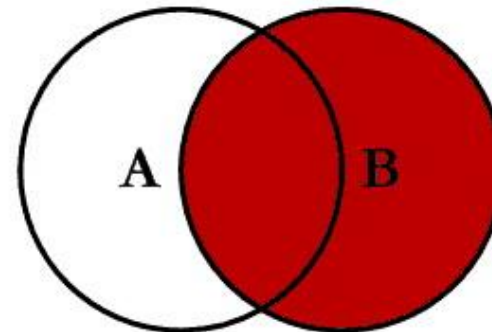
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```



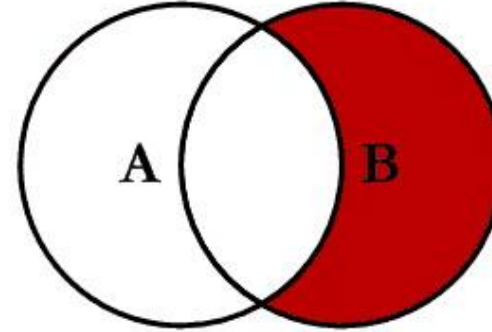
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key  
WHERE B.Key IS NULL
```



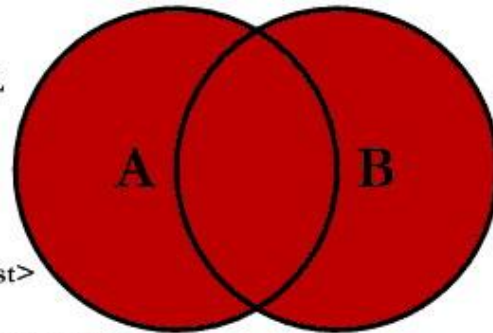
```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```



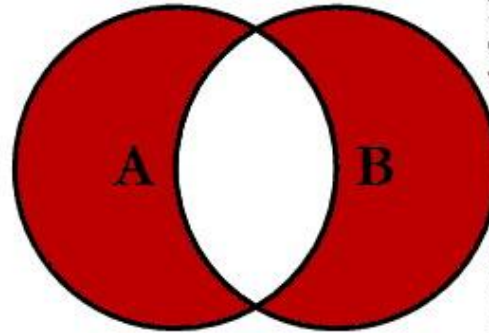
```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL  
OR B.Key IS NULL
```

