

# SELECT (selecionar)

## SELEÇÃO COM JUNÇÃO

- Quando os dados requeridos são de tabelas distintas, isto é, atributos de mais uma tabela são requeridos.
- Existe uma CONDIÇÃO DE JUNÇÃO, na qual os atributos **chave primária e chave estrangeira** das relações devem ser relacionados.

# SELECT (selecionar)

## SELEÇÃO COM JUNÇÃO

- **Ex.1:** Obter os nomes dos técnicos com experiência em secadora.

SELECT Nome

FROM TÉCNICOS, EXPERIÊNCIA

WHERE NumTec = NumTecnico AND Tipo = 'Secadora'

NumTec	Nome	Cargo
297	Marco	Trainee
553	Hélio	Sênior
062	Tião	Sênior
718	Sílvio	Estagiário

NumTecnico	Tipo	AnosExp
553	Secadora	15
062	Lavadora	18
297	Torradeira	1
297	Secadora	1
718	Lavadora	5
062	Congelador	10
062	Secadora	12

# SELECT (selecionar)

## SELEÇÃO COM JUNÇÃO

- Ex.2: Obter nomes dos técnicos com experiência maior que 10 anos.

SELECT Nome

FROM TÉCNICOS, EXPERIÊNCIA

WHERE NumTec = NumTecnico AND AnosExp > 15

NumTec	Nome	Cargo
297	Marco	Trainee
553	Hélio	Sênior
062	Tião	Sênior
718	Sílvio	Estagiário

NumTecnico	Tipo	AnosExp
553	Secadora	15
062	Lavadora	18
297	Torradeira	1
297	Secadora	1
718	Lavadora	5
062	Congelador	10
062	Secadora	12

# SELECT (selecionar)

## SELEÇÃO COM JUNÇÃO

**Ex.3:** Liste o nome dos técnicos e sua experiência em aparelhos da categoria 1.

Tipo	Categoria	Taxa
Lavadora	1	20,00
Secadora	1	20,00
Torradeira	2	10,00
Congelador	1	8,00
Batedeira	2	25,00

NumTec	Nome	Cargo
297	Marco	Trainee
553	Hélio	Sênior
062	Tião	Sênior
718	Sílvio	Estagiário

NumTecnico	Tipo	AnosExp
553	Secadora	15
062	Lavadora	18
297	Torradeira	1
297	Secadora	1
718	Lavadora	5
062	Congelador	10
062	Secadora	12

# SELECT (selecionar)

## SELEÇÃO COM JUNÇÃO

**Ex.3:** Liste o nome dos técnicos e sua experiência em aparelhos da categoria 1.

// colocar nomes das tabelas antes dos atributos

```
SELECT  TECNICOS.Nome, EXPERIÊNCIA.AnosExp,  
        TIPOS.Tipo  
FROM    TECNICOS, TIPOS, EXPERIENCIA  
WHERE   TIPOS.Tipo = EXPERIENCIA.Tipo AND  
        EXPERIENCIA.NumTecnico = TECNICOS.NumTec  
        AND TIPOS.Categoria = 1
```

# SELECT (selecionar)

## USO DE ALIASES

- Aliases → “apelidos” para as tabelas, usando a palavra reservada “AS”
- Permite associar um “nome de variável” para cada relação, a fim de simplificar os comandos SQL.
- Criando um SELECT com Aliases, temos:

```
SELECT  C.Nome, E.AnosExp, TP.Tipo
FROM    TECNICOS AS T, TIPOS AS TP, EXPERIENCIA AS E
WHERE   TP.Tipo = E.Tipo AND E.NumTecnico = T.NumTec
        AND TP.Categoria = 1
```

# CONSULTAS EM SQL - RESUMO

- Conjunto completo de cláusulas no comando SELECT-FROM
- Uma consulta em SQL pode consistir em até seis cláusulas:

```
SELECT [*] [DISTINCT] <lista de atributos> <funções agregação>
FROM <lista de tabelas>
[WHERE <condicao>]
[GROUP BY <lista de atributos para agrupamento>]
[HAVING <condicao para agrupamento, aceita funções agregação>]
[ORDER BY <atributo1, atributo2,..., atributo N> [ASC] [DESC]]
```

## OBSERVAÇÕES

- Apenas as cláusulas **SELECT** e **FROM** são obrigatórias.
- Quando existentes, as cláusulas devem aparecer na ordem especificada acima.
- O **ORDER BY** só pode ser utilizado após o ultimo **SELECT** (se a linguagem permitir).
- As cláusulas **GROUP BY** e **HAVING** só podem ser usados nos comandos **SELECT** individuais.

# JUNÇÕES

- É a possibilidade de se criar relacionamentos entre tabelas de forma a poder **recuperar dados de todas elas através de uma única consulta.**
- Através de uma operação chamada **JOIN** (junção) é possível consultar e manipular dados de mais de uma tabela usando apenas um comando **SELECT**.



# JUNÇÕES

- É importante utilizá-lo, porque tira da cláusula WHERE condições que são estritamente das junções (chave primária igual a chave estrangeira, por exemplo).
- Existem as variações de **junções internas e externas**.
- Internas: INNER JOIN, NATURAL JOIN

OBS.: A palavra INNER pode ser omitida.

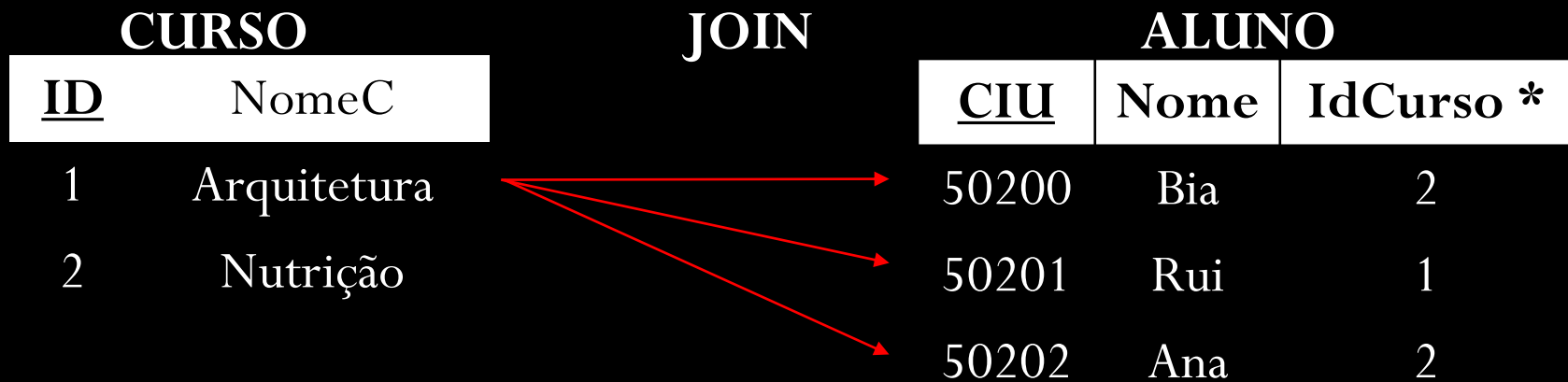
# JUNÇÕES

## COMO FUNCIONA O COMANDO JOIN?

- Quando um comando **SELECT** especifica campos de duas tabelas **sem nenhuma restrição ou filtro**, o resultado será um número de linhas iguais à multiplicação do total de linhas da primeira tabela (N) pela segunda tabela (M) =  $N \times M$  tuplas.
- Isso ocorre devido ao fato de que, para cada linha da primeira tabela, todas as linhas da segunda são processadas.
- **Operações de junções tomam duas relações e têm como resultado uma outra relação.**

# JUNÇÕES

## FUNCIONAMENTO DO COMANDO JOIN



### RESULTADO:

2 linhas x 3 linhas  
= 6 linhas

ID	NomeC	CIU	Nome	IdCurso
1	Arquitetura	50200	Bia	2
1	Arquitetura	50201	Rui	1
1	Arquitetura	50202	Ana	2
2	Nutrição	50200	Bia	2
2	Nutrição	50201	Rui	1
2	Nutrição	50202	Ana	2

# JUNÇÕES


## FUNCIONAMENTO DO COMANDO JOIN

### RESULTADO:

Porém, o resultado só fará sentido quando as chaves PK e FK forem equivalentes, ou seja,  $ID = IDCURSO$  (isso é feito usando o ON).

### RESULTADO CORRETO:

IDs devem ser iguais



ID	NomeC	CIU	Nome	IdCurso
1	Arquitetura	50200	Bia	2
1	Arquitetura	50201	Rui	1
1	Arquitetura	50202	Ana	2
2	Nutrição	50200	Bia	2
2	Nutrição	50201	Rui	1
2	Nutrição	50202	Ana	2

ID	NomeC	CIU	Nome	IdCurso
1	Arquitetura	50201	Rui	1
2	Nutrição	50200	Bia	2
2	Nutrição	50202	Ana	2

# JUNÇÕES

- **INNER JOIN (ou somente JOIN)**
- O inner join é uma junção interna.
- Junta os registros da tabela que tiver um **correspondente** na outra tabela, através das chaves primária e estrangeira.

NumTec	Nome	Cargo	NumTecnico	Tipo	AnosExp
--------	------	-------	------------	------	---------

→ TECNICOS **INNER JOIN** EXPERIENCIA **ON**  
NumTec=NumTecnico

# JUNÇÕES

## TIPOS DE JUNÇÕES E CONDIÇÃO DE JUNÇÃO

Cada uma das variantes das operações de JUNÇÃO consiste em um *tipo de junção* e em uma *condição de junção*.

- **Condição de junção (ON)**
  - Definem quais tuplas das duas relações apresentam correspondência e quais atributos são apresentados no resultado de uma junção.
- **Tipo de junção (inner ou outer)**
  - Define como as tuplas em cada relação que não possuam nenhuma correspondência com as tuplas da outra relação devem ser tratadas.

# JUNÇÕES

- **USANDO INNER JOIN** com 2 tabelas.
- **Ex.3:** Liste o nome dos técnicos que possuem experiência em congelador.

SELECT Nome

FROM (TECNICOS INNER JOIN EXPERIENCIA ON  
NumTec=NumTecnico)

WHERE Tipo = 'congelador'

NumTec	Nome	Cargo
297	Marco	Trainee
553	Hélio	Sênior
062	Tião	Sênior
718	Sílvio	Estagiário

NumTecnico	Tipo	AnosExp
553	Secadora	15
062	Lavadora	18
297	Torradeira	1
297	Secadora	1
718	Lavadora	5
062	Congelador	10
062	Secadora	12

# JUNÇÕES

- **USANDO INNER JOIN**  
com 3 tabelas.
- **Exemplo:** Liste o nome dos técnicos e sua experiência em aparelhos da categoria 1.

Tipo	Categoria	Taxa
Lavadora	1	20,00
Secadora	1	20,00
Torradeira	2	10,00
Congelador	1	8,00
Batedeira	2	25,00

NumTecnico	Tipo	AnosExp
553	Secadora	15
062	Lavadora	18
297	Torradeira	1
297	Secadora	1
718	Lavadora	5
062	Congelador	10
062	Secadora	12

NumTec	Nome	Cargo
297	Marco	Trainee
553	Hélio	Sênior
062	Tião	Sênior
718	Sílvio	Estagiário



# JUNÇÕES

USANDO INNER JOIN com 3 tabelas.

- **Exemplo:** Liste o nome dos técnicos e sua experiência em aparelhos da categoria 1.

```
SELECT Nome, AnosExp
```

```
FROM ((TECNICOS INNER JOIN EXPERIENCIA ON  
      NumTec = NumTecnico) INNER JOIN TIPOS ON Tipo =  
      Tipo)
```

```
WHERE Categoria = 1
```

# JUNÇÕES

- **NATURAL JOIN**
- Com ele você não precisa identificar quais colunas serão comparadas, pois ele fará a comparação entre campos com mesmo nome.

→ TECNICOS **NATURAL JOIN** EXPERIENCIA

Tipo	Categoria	Taxa	NumTecnico	Tipo	AnosExp
------	-----------	------	------------	------	---------

-----

→ Junção natural entre TECNICOS e EXPERIÊNCIA não dá (atributos com nomes diferentes)!!!

NumTec	Nome	Cargo	NumTecnico	Tipo	AnosExp
--------	------	-------	------------	------	---------

# JUNÇÕES

- Existem as variações de junções internas e externas.
- Externas:
- O **OUTER JOIN** é uma junção externa.
- Temos:
  - **LEFT OUTER JOIN**
  - **RIGHT OUTER JOIN**

OBS.: A palavra **OUTER** pode ser omitida.

# JUNÇÕES

- **LEFT OUTER JOIN**
- A precedência é da tabela da Esquerda, isto é, todos os registros da primeira tabela serão mostrados independente se houver correspondente na outra tabela, após a equivalência das chaves através do “ON”.
- Pega todos os atributos da relação que está à esquerda, verifica se existe algum correspondendo à direita, caso afirmativo, retorna os atributos da direita e caso negativo, coloca o valor nulo (NULL) nos atributos.

# JUNÇÕES

- **LEFT OUTER JOIN**

→ LISTAR TODOS OS FUNCIONÁRIOS PERTENCENTES AS LOJAS.

```
SELECT F.NOMEFUN, L.NOME  
FROM FUNCIONARIOS AS F LEFT JOIN LOJAS AS L ON  
F.CODIGOL = L.CODIGO
```

<u>CODIGOFUN</u>	<u>CODIGOL*</u>	NOMEFUN
1	1	JOÃO
2	1	JOAQUIM
3	2	JOSÉ
4	3	MÁRCIO
5	NULL	PEDRO

<u>CÓDIGO</u>	NOME
1	MATRIZ
2	FILIAL 1
3	FILIAL 2

# JUNÇÕES

- **LEFT OUTER JOIN**

Neste caso serão listados TODOS os funcionários mesmo os que não haja lojas cadastradas, POIS A PRIORIDADE É DA TABELA FUNCIONÁRIO.

RESULTADO:

NOMEFUN	NOME
JOÃO	MATRIZ
JOAQUIM	FILIAL 1
JOSÉ	FILIAL 1
MÁRCIO	FILIAL 2
PEDRO	NULL

# JUNÇÕES

- **RIGHT OUTER JOIN (join)**
- Prioridade da tabela à direita, isto é, todos os registros da segunda tabela serão mostrados independente se houver correspondente na outra tabela.
- Pega todos os atributos da relação que esta à direita, verifica se existe algum correspondente á esquerda, caso afirmativo, retorna os atributos da esquerda e caso negativo, coloca o valor nulo nos atributos.

# JUNÇÕES

- **RIGHT OUTER JOIN**

→ LISTAR TODOS OS FUNCIONÁRIOS PERTENCENTES AS LOJAS.

```
SELECT F.NOMEFUN, L.NOME  
FROM  FUNCIONARIOS AS F RIGHT JOIN LOJAS AS L ON  
      F.CODIGOL = L.CODIGO
```

CODIGOFUN	CODIGOL	NOMEFUN
1	1	JOÃO
1	1	JOAQUIM
3	2	JOSÉ
4	3	MÁRCIO
5	NULL	PEDRO

CÓDIGO	NOME
1	MATRIZ
2	FILIAL 1
3	FILIAL 2



# JUNÇÕES

- **RIGHT OUTER JOIN**

Neste caso serão listados TODAS AS LOJAS com seus respectivos funcionários, POIS A PRIORIDADE É DA TABELA LOJAS.

## RESULTADO:

NOMEFUN	NOME
JOÃO	MATRIZ
JOAQUIM	FILIAL 1
JOSÉ	FILIAL 1
MÁRCIO	FILIAL 2

Neste caso o resultado do RIGHT JOIN seria igual ao resultado do INNER JOIN.