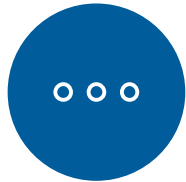


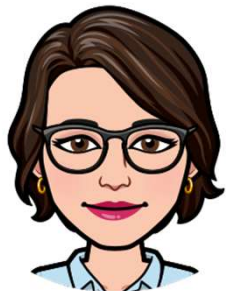
SENAI DENDEZEIROS

CURSO: DESENVOLVIMENTO DE SISTEMAS

UNIDADE CURRICULAR/DISCIPLINA: BANCO DE DADOS



Sistema FIEB



Christiane Santana
christiane.soares@ba.docente.senai.br



JOIN

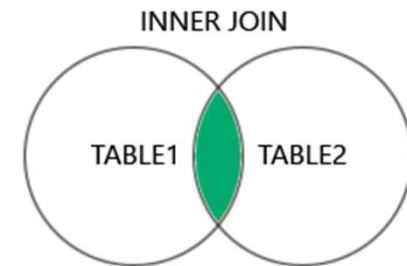
Junção de 2 ou mais tabelas



SQL – Structured Query Language

✓ INNER JOIN

- Retorna registros que possuem valores correspondentes em ambas as tabelas



Fonte: https://www.w3schools.com/mysql/mysql_join.asp

```
SELECT Table1.CampoSelecioneadoTab1, Table2.CampoSelecioneadoTab2
```

```
FROM Table1
```

```
INNER JOIN Table2 ON Table1.CampoReferenciaTab1 = Table2.CampoReferenciaTab2;
```

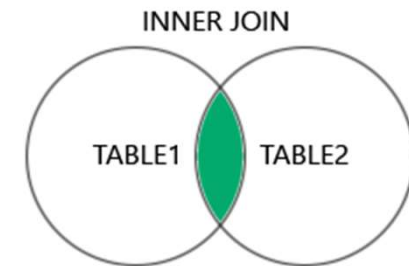
SQL – Structured Query Language

Exemplo:

SELECT nomeCurso, nome, sobrenome, ultimosobrenome

FROM aluno

INNER JOIN curso **ON** fk_Curso_IdCurso = idCurso



Fonte: https://www.w3schools.com/mysql/mysql_join.asp

Tabela: ALUNO

nome	sobrenome	ultimosobrenome	FK_CURSO_IDCURSO
ANA	SANTOS	SANTANA	1
CARLA	LIMA	SANTOS	2
CARLOS	PEREIRA	SILVA	1
JOAO	FERREIRA	MENDES	2
JOSE	SILVA	SANTOS	1
AMELIA	LIMA	SILVA	2
RODRIGO	NULL	SANTOS	1
VIVIANE	MENEZES	SILVA	2
CELIA	RODRIGUES	SANTANA	NULL

Tabela: CURSO

IDCURSO	NOMECURSO
1	Desenvolvimento de Sistemas
2	Redes de Computadoes
3	Automação

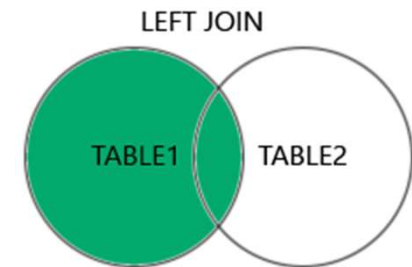
Resultado: INNER JOIN

nomeCurso	nome	sobrenome	ultimosobrenome
Desenvolvimento de Sistemas	ANA	SANTOS	SANTANA
Desenvolvimento de Sistemas	CARLOS	PEREIRA	SILVA
Desenvolvimento de Sistemas	JOSE	SILVA	SANTOS
Desenvolvimento de Sistemas	RODRIGO	NULL	SANTOS
Redes de Computadoes	CARLA	LIMA	SANTOS
Redes de Computadoes	JOAO	FERREIRA	MENDES
Redes de Computadoes	AMELIA	LIMA	SILVA
Redes de Computadoes	VIVIANE	MENEZES	SILVA

SQL – Structured Query Language

✓ LEFT JOIN

- Retorna todos os registros da tabela esquerda (Table1) e os registros correspondentes da tabela direita (Table2)



Fonte: https://www.w3schools.com/mysql/mysql_join.asp

```
SELECT Table1.CampoSelecioneadoTab1, Table2.CampoSelecioneadoTab2
```

```
FROM Table1
```

```
INNER JOIN Table2 ON Table1.CampoReferenciaTab1 = Table2.CampoReferenciaTab2;
```

SQL – Structured Query Language

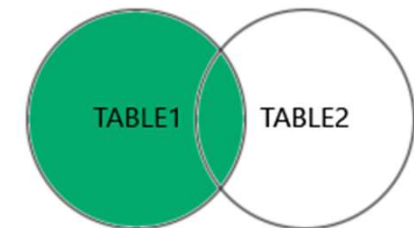
Exemplo:

SELECT nomeCurso, nome, sobrenome, ultimosobrenome

FROM aluno

LEFT JOIN curso **ON** fk_Curso_IdCurso = idCurso

LEFT JOIN



Fonte: https://www.w3schools.com/mysql/mysql_join.asp

Tabela: ALUNO

nome	sobrenome	ultimosobrenome	FK_CURSO_IDCURSO
ANA	SANTOS	SANTANA	1
CARLA	LIMA	SANTOS	2
CARLOS	PEREIRA	SILVA	1
JOAO	FERREIRA	MENDES	2
JOSE	SILVA	SANTOS	1
AMELIA	LIMA	SILVA	2
RODRIGO	NULL	SANTOS	1
VIVIANE	MENEZES	SILVA	2
CELIA	RODRIGUES	SANTANA	NULL

Tabela: CURSO

IDCURSO	NOMECURSO
1	Desenvolvimento de Sistemas
2	Redes de Computadoes
3	Automação

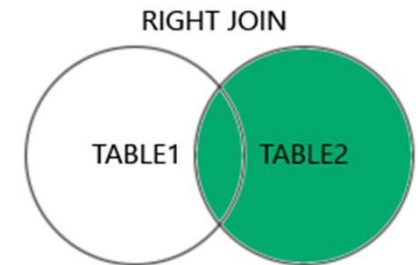
Resultado: LEFT JOIN

nomeCurso	nome	sobrenome	ultimosobrenome
NULL	CELIA	RODRIGUES	SANTANA
Desenvolvimento de Sistemas	ANA	SANTOS	SANTANA
Desenvolvimento de Sistemas	CARLOS	PEREIRA	SILVA
Desenvolvimento de Sistemas	JOSE	SILVA	SANTOS
Desenvolvimento de Sistemas	RODRIGO	NULL	SANTOS
Redes de Computadoes	CARLA	LIMA	SANTOS
Redes de Computadoes	JOAO	FERREIRA	MENDES
Redes de Computadoes	AMELIA	LIMA	SILVA
Redes de Computadoes	VIVIANE	MENEZES	SILVA

SQL – Structured Query Language

✓ RIGHT JOIN

- Retorna todos os registros da tabela direita (Table2) e os registros correspondentes da tabela esquerda (Table1)



Fonte: https://www.w3schools.com/mysql/mysql_join.asp

```
SELECT Table1.CampoSelecioneadoTab1, Table2.CampoSelecioneadoTab2
```

```
FROM Table1
```

```
RIGHT JOIN Table2 ON Table1.CampoReferenciaTab1 = Table2.CampoReferenciaTab2;
```

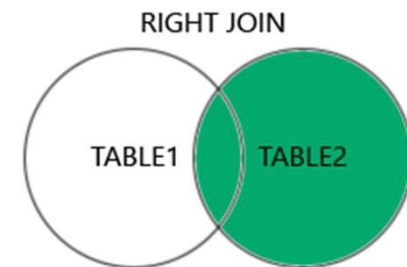
SQL – Structured Query Language

Exemplo:

SELECT nomeCurso, nome, sobrenome, ultimosobrenome

FROM aluno

RIGHT JOIN curso **ON** fk_Curso_IdCurso = idCurso



Fonte: https://www.w3schools.com/mysql/mysql_join.asp

Tabela: ALUNO

nome	sobrenome	ultimosobrenome	FK_CURSO_IDCURSO
ANA	SANTOS	SANTANA	1
CARLA	LIMA	SANTOS	2
CARLOS	PEREIRA	SILVA	1
JOAO	FERREIRA	MENDES	2
JOSE	SILVA	SANTOS	1
AMELIA	LIMA	SILVA	2
RODRIGO	NULL	SANTOS	1
VIVIANE	MENEZES	SILVA	2
CELIA	RODRIGUES	SANTANA	NULL

Tabela: CURSO

IDCURSO	NOMECURSO
1	Desenvolvimento de Sistemas
2	Redes de Computadoes
3	Automação

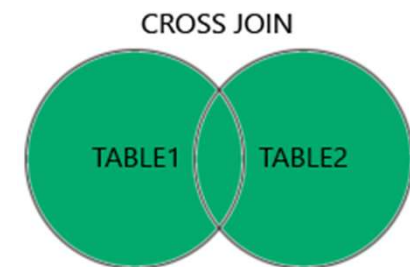
Resultado: RIGHT JOIN

nomeCurso	nome	sobrenome	ultimosobrenome
Automação	NULL	NULL	NULL
Desenvolvimento de Sistemas	ANA	SANTOS	SANTANA
Desenvolvimento de Sistemas	CARLOS	PEREIRA	SILVA
Desenvolvimento de Sistemas	JOSE	SILVA	SANTOS
Desenvolvimento de Sistemas	RODRIGO	NULL	SANTOS
Redes de Computadoes	CARLA	LIMA	SANTOS
Redes de Computadoes	JOAO	FERREIRA	MENDES
Redes de Computadoes	AMELIA	LIMA	SILVA
Redes de Computadoes	VIVIANE	MENEZES	SILVA

SQL – Structured Query Language

✓ CROSS JOIN

- Retorna todos os registros de ambas as tabelas



Fonte: https://www.w3schools.com/mysql/mysql_join.asp

```
SELECT Table1.CampoSelecioneadoTab1, Table2.CampoSelecioneadoTab2
```

```
FROM Table1
```

```
CROSS JOIN Table2
```

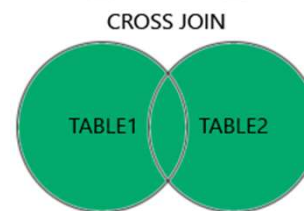
SQL – Structured Query Language

Exemplo:

SELECT nomeCurso, nome, sobrenome, ultimosobrenome

FROM aluno

CROSS JOIN curso



Fonte: https://www.w3schools.com/mysql/mysql_join.asp

Tabela: ALUNO

nome	sobrenome	ultimosobrenome	FK_CURSO_IDCURSO
ANA	SANTOS	SANTANA	1
CARLA	LIMA	SANTOS	2
CARLOS	PEREIRA	SILVA	1
JOAO	FERREIRA	MENDES	2
JOSE	SILVA	SANTOS	1
AMELIA	LIMA	SILVA	2
RODRIGO	NULL	SANTOS	1
VIVIANE	MENEZES	SILVA	2
CELIA	RODRIGUES	SANTANA	NULL

Tabela: CURSO

IDCURSO	NOMECURSO
1	Desenvolvimento de Sistemas
2	Redes de Computadoes
3	Automação

Resultado: CROSS JOIN

nomeCurso	nome	sobrenome	ultimosobrenome
Automação	ANA	SANTOS	SANTANA
Automação	CARLA	LIMA	SANTOS
Automação	CARLOS	PEREIRA	SILVA
Automação	JOAO	FERREIRA	MENDES
Automação	JOSE	SILVA	SANTOS
Automação	AMELIA	LIMA	SILVA
Automação	RODRIGO	NULL	SANTOS
Automação	VIVIANE	MENEZES	SILVA
Automação	CELIA	RODRIGUES	SANTANA
Desenvolvimento de Sistemas	ANA	SANTOS	SANTANA
Desenvolvimento de Sistemas	CARLA	LIMA	SANTOS
Desenvolvimento de Sistemas	CARLOS	PEREIRA	SILVA
Desenvolvimento de Sistemas	JOAO	FERREIRA	MENDES
Desenvolvimento de Sistemas	JOSE	SILVA	SANTOS
Desenvolvimento de Sistemas	AMELIA	LIMA	SILVA
Desenvolvimento de Sistemas	RODRIGO	NULL	SANTOS
Desenvolvimento de Sistemas	VIVIANE	MENEZES	SILVA
Desenvolvimento de Sistemas	CELIA	RODRIGUES	SANTANA
Redes de Computadoes	ANA	SANTOS	SANTANA
Redes de Computadoes	CARLA	LIMA	SANTOS
Redes de Computadoes	CARLOS	PEREIRA	SILVA
Redes de Computadoes	JOAO	FERREIRA	MENDES
Redes de Computadoes	JOSE	SILVA	SANTOS
Redes de Computadoes	AMELIA	LIMA	SILVA
Redes de Computadoes	RODRIGO	NULL	SANTOS
Redes de Computadoes	VIVIANE	MENEZES	SILVA
Redes de Computadoes	CELIA	RODRIGUES	SANTANA



INSERT, UPDATE e DELET
com SELECT

SQL – Structured Query Language

✓ INSERT com SELECT

```
INSERT INTO <tabela destino> (<campos da tabela destino>)
```

```
SELECT <campos da tabela origem> FROM <tabela origem> [WHERE <condição>]
```

SQL – Structured Query Language

✓ INSERT com SELECT

```
INSERT INTO <tabela destino> (<campos da tabela destino>)
```

```
SELECT <campos da tabela origem> FROM <tabela origem> [WHERE <condição>]
```

Exemplo: Insere na tabela ALUNO2 todos os registros da tabela ALUNO. Nesse caso os valores que retornam do SELECT serão inseridos nos campos da tabela destino obedecendo a ordem dos campos das tabelas (posicional)

```
INSERT INTO ALUNO2
```

```
SELECT * FROM ALUNO;
```

SQL – Structured Query Language

✓ INSERT com SELECT

`INSERT INTO <tabela destino> (<campos da tabela destino>)`

`SELECT <campos da tabela origem> FROM <tabela origem> [WHERE <condição>]`

Exemplo: Insere na tabela SELECIONADOS todos os registros da tabela ALUNO com BAIRRO = “BONFIM”. Nesse caso, só serão preenchidos os campos mencionados da tabela SELECIONADOS e os valores que retornam do SELECT serão inseridos obedecendo a ordem dos campos selecionados(posicional)

`INSERT INTO SELECIONADOS (NumCPF, nomeALuno, sobrenomeAluno, ultimosobrenomeAluno, idadeAluno)`

`SELECT CPF, nome, sobrenome, ultimosobrenome, idade FROM ALUNO WHERE bairro = "Bonfim"`

SQL – Structured Query Language

✓ UPDATE com SELECT

UPDATE <tabela>

SET <campo a ser alterado> = <novo valor>

WHERE <campo condicionado> IN

SELECT <campo a ser comparado> FROM <tabela> [WHERE <condição>]

SQL – Structured Query Language

✓ UPDATE com SELECT

`UPDATE <tabela>`

`SET <campo a ser alterado> = <novo valor>`

`WHERE <campo condicionado> IN`

`(SELECT <campo a ser comparado> FROM <tabela> [WHERE <condição>])`

Exemplo: Altera a tabela ALUNO2, setando o valor do campo CONVOCADO para 1 de todos os registros que tem o valor do campo NUMCPF igual a um dos valores dos CPFs selecionados no SELECT (CPF de alunos com bairro é igual a Bonfim)

`UPDATE ALUNO2`

`SET CONVOCADO = 1`

`WHERE NUMCPF IN (SELECT CPF FROM ALUNO WHERE bairro = "Bonfim")`

SQL – Structured Query Language

- ✓ DELETE com SELECT

DELETE FROM <tabela>

WHERE <campo condicionado> IN

SELECT <campo a ser comparado> FROM <tabela> [WHERE <condição>]

SQL – Structured Query Language

✓ DELETE com SELECT

`DELETE FROM <tabela>`

`WHERE <campo condicionado> IN`

`SELECT <campo a ser comparado> FROM <tabela> [WHERE <condição>]`

Exemplo: Deleta os registros da tabela ALUNO2, que tem o valor do campo NUMCPF igual a um dos valores dos CPFs selecionados no SELECT (CPF de alunos com bairro é igual a Brotas)

`DELETE FROM ALUNO2`

`WHERE NUMCPF IN (SELECT CPF FROM ALUNO WHERE bairro = "Brotas")`



VIEWS

Tabela virtual baseada no conjunto de resultados de uma instrução SQL.

FONTE: https://www.w3schools.com/mysql/mysql_view.asp



SQL – Structured Query Language

✓ VIEW

- O que é?
- Vantagens ?

SQL – Structured Query Language

- ✓ Criação de uma VIEW

```
CREATE VIEW <nome_view> AS
```

```
SELECT <campos> FROM <tabela> [WHERE <condições>]
```

SQL – Structured Query Language

- ✓ Alteração de uma View

```
ALTER VIEW <nome_view> AS
```

```
SELECT <campos> FROM <tabela> [WHERE <condições>]
```

SQL – Structured Query Language

- ✓ Exclusão de uma View

```
DROP VIEW <nome_view>
```

SQL – Structured Query Language

✓ Exemplo

```
/* VIEW: Seleção dos livros com seus respectivos autores */  
CREATE VIEW LIVRO_AUTOR AS  
    SELECT L.tituloLivro, A.nomeAutor  
    FROM livro L, autor A, autor_livro AL  
    WHERE (AL.fk_Livro_ISBN = L.ISBN) and (AL.fk_Autor_idAutor = A.idAutor);
```

/* ou */

```
CREATE VIEW LIVRO_AUTOR AS  
    SELECT L.tituloLivro, A.nomeAutor  
    FROM autor_livro AL  
    INNER JOIN livro L ON AL.fk_Livro_ISBN = L.ISBN  
    INNER JOIN autor A ON AL.fk_Autor_idAutor = A.idAutor;
```




STORED PROCEDURE

Um procedimento armazenado com comandos que
podem ser reutilizados

FONTE: https://www.w3schools.com/sql/sql_stored_procedures.asp

SQL – Structured Query Language

✓ STORED PROCEDURE

- O que é?
- Vantagens?

SQL – Structured Query Language

✓ CREATE PROCEDURE <nome> (<parametros>)

<COMANDOS>

SQL – Structured Query Language

✓ CREATE PROCEDURE <nome> (<parametros>)

<COMANDOS>

✓ CALL <nome>(<parâmetros>)

SQL – Structured Query Language

✓ Parâmetros

- IN (entrada)
- OUT (saída)
- INOUT (entrada e saída)

SQL – Structured Query Language

- ✓ BLOCO BEGIN ... END (para mais de um comando)
- ✓ Quando uma rotina tem mais de um comando, o delimitador da rotina deve ser alterado, pois todo comando interno deve ser finalizado com ;
- ✓ DELIMITER <delimitador> (muda o delimitador, e a rotina passa a ser finalizada com esse novo delimitador)

SQL – Structured Query Language

DELIMITER <delimitador>

CREATE PROCEDURE [IF NOT EXISTS] <nome> (<parametros>)

BEGIN

<COMANDOS>;

END<delimitador>

DELIMITER ;

SQL – Structured Query Language

Exemplo

```
/* STORED PROCEDURE: Seleção da quantidade de exemplares dos livros de uma determinada categoria */  
  
CREATE PROCEDURE Verificar_Quantidade_Exemplares_categoria (IN categoria INT)  
    SELECT SUM(quantExemplares) FROM livro  
    GROUP BY fk_Categoria_idCategoria  
    HAVING fk_Categoria_idCategoria = categoria;  
  
/* CALL: Chama a procedure passando por parâmetro o valor 3. Assim, seleciona a quantidade de exemplares  
    dos livros de categoria 3 */  
  
CALL Verificar_Quantidade_Exemplares_categoria(3);
```


SQL – Structured Query Language

Exemplo

```
/* STORED PROCEDURE: Seleção da quantidade de livros de uma determinada categoria - retornando essa  
quantidade através do parametro "quantidade" */
```

```
CREATE PROCEDURE Verificar_Quantidade_livros (OUT quantidade INT)
```

```
    SELECT COUNT(*) INTO quantidade FROM livro;
```

```
/* CALL: Chama a procedure passando por parâmetro a variável quant (pois o parâmetro é de saída OUT).  
Assim, seleciona a quantidade de exemplares dos livros de categoria 3 */
```

```
CALL Verificar_Quantidade_livros(@quant);
```

```
/* SELECT: Seleciona o valor da variável quant e mostra o resultado da chamada a procedure */
```

```
SELECT @quant;
```



FUNCTION

SQL – Structured Query Language

✓ FUNCTION

- O que é?
- Vantagens?

SQL – Structured Query Language

✓ CREATE FUNCTION <nome> (<parametros>)

<COMANDOS>

✓ SELECT <nome>(<parâmetros>)

SQL – Structured Query Language

DELIMITER <delimitador>

CREATE FUNCTION [IF NOT EXISTS] <nome> (<parametros>)

RETURNS <tipoRetorno>

BEGIN

<COMANDOS>

RETURN <valorRetorno>

END<delimitador>

DELIMITER ;

SQL – Structured Query Language

✓ SET @<nomeVariavel> = <valor>

SQL – Structured Query Language

✓ Exemplo

```
/* FUNCTION: retorna a quantidade de livros de uma determinada categoria */  
DELIMITER $  
CREATE FUNCTION Ver_Quantidade_livros(categoria INT)  
RETURNS int  
READS SQL DATA  
BEGIN  
    return (SELECT COUNT(*) FROM livro WHERE fk_Categoria_idCategoria = categoria);  
END$  
DELIMITER ;  
SELECT Ver_Quantidade_livros(3);
```



TRIGGER



SQL – Structured Query Language

✓ TRIGGER

- O que é?
- Vantagens?
- Desvantagens?

SQL – Structured Query Language

✓ Exemplo

```
DELIMITER $  
CREATE TRIGGER TGR_EMPRESTIMO_BI AFTER INSERT  
ON emprestimo_livro  
FOR EACH ROW  
BEGIN  
    UPDATE livro SET quantExemplares = quantExemplares - 1 WHERE ISBN = NEW.fk_Livro_ISBN;  
END$  
DELIMITER ;
```

Referências

ELMASRI, R., NAVATHE, S. - Sistemas de Banco de Dados - 7ª ed - São Paulo: Pearson Education do Brasil, 2018



Créditos e Agradecimentos



Infográficos com post-its

<https://slidesgo.com/pt/tema/infograficos-com-post-its>

Emoji pessoal

<https://www.bitmoji.com/>





Até a próxima aula...



Sistema FIEB



PELO FUTURO DO TRABALHO