

Trabalho de Banco de Dados - Projeto – Parte I

Aluno: 3º SGT F. Leite RG 93.177

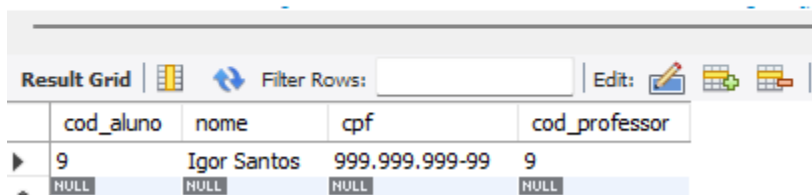
Parte 2

1. 2 consultas com SELECT e WHERE

Filtra todos os dados da tabela Aluno onde o campo nome é exatamente "Igor Santos".

```
SELECT * FROM Aluno where nome = 'Igor Santos';
```

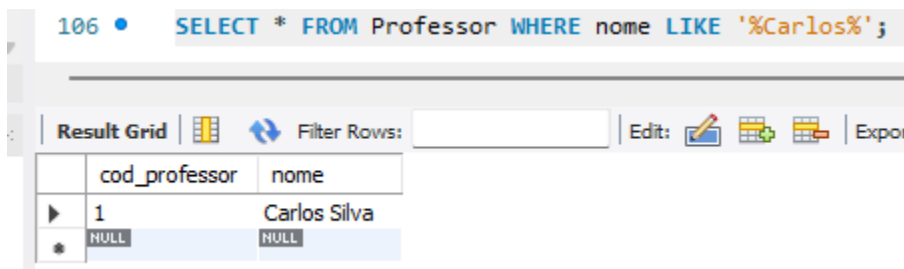
```
104 -- 2 consultas com SELECT e WHERE |
105 • SELECT * FROM Aluno where nome = 'Igor Santos';
106
```



	cod_aluno	nome	cpf	cod_professor
▶	9	Igor Santos	999.999.999-99	9
*	NULL	NULL	NULL	NULL

Mostra todos os professores cujo nome contém a palavra "Carlos".

```
SELECT * FROM Professor WHERE nome LIKE '%Carlos%';
```



```
106 • SELECT * FROM Professor WHERE nome LIKE '%Carlos%';
```

	cod_professor	nome
▶	1	Carlos Silva
*	NULL	NULL

2. 2 consultas com GROUP BY e ORDER BY com funções de agregação

Conta quantos alunos estão vinculados a cada cod_professor.

```
SELECT cod_professor, COUNT(*) AS total_alunos FROM Aluno GROUP BY  
cod_professor ORDER BY total_alunos DESC;
```

	cod_professor	total_alunos
▶	1	1
	2	1
	3	1
	4	1
	5	1

Result 18

Soma todas as mensalidades recebidas por secretaria.

```
SELECT cod_secretaria, SUM(valor_mensalidade) AS total_recebido FROM
Mensalidade GROUP BY cod_secretaria ORDER BY total_recebido DESC;
```

	cod_secretaria	total_recebido
▶	10	540.00
	8	530.00
	9	520.00
	7	510.00
	1	500.00

Result 19

3. 2 consultas com operadores aritméticos (+, -, *, /)

Aplica um aumento de 10% na mensalidade (valor_mensalidade * 1.10).

```
SELECT cod_mensalidade, valor_mensalidade, valor_mensalidade * 1.10 AS
valor_ajustado FROM Mensalidade;
```

	cod_mensalidade	valor_mensalidade	valor_ajustado
▶	1	500.00	550.0000
	2	480.00	528.0000
	3	470.00	517.0000
	4	460.00	506.0000
	5	450.00	495.0000

Result 21

Subtrai R\$ 100,00 de cada valor pago, como se fosse um desconto.

```
SELECT id_pagamento, valor_pago, valor_pago - 100 AS valor_descontado FROM
Efetua_Pagamento;
```

	id_pagamento	valor_pago	valor_descontado
▶	1	3000.00	2900.00
	2	2800.00	2700.00
	3	2600.00	2500.00
	4	3100.00	3000.00
	5	2700.00	2600.00

4. 3 consultas com operadores de comparação (=, !=, <, >, etc.)

Mostra mensalidades com valor exato de R\$ 500,00.

SELECT * FROM Mensalidade WHERE valor_mensalidade = 500;

	cod_mensalidade	valor_mensalidade	cod_aluno	cod_secretaria
▶	1	500.00	1	1
*	NULL	NULL	NULL	NULL

Mostra pagamentos que não são R\$ 3000.

SELECT * FROM Efetua_Pagamento WHERE valor_pago != 3000;

Result Grid					
		Filter Rows:		Edit:	
				Export/Import:	
	id_pagamento	cod_professor	cod_secretaria	valor_pago	data_pagamento
▶	2	2	2	2800.00	2025-07-01
	3	3	3	2600.00	2025-07-01
	4	4	4	3100.00	2025-07-01
	5	5	5	2700.00	2025-07-01
	6	6	6	2900.00	2025-07-01

Efetua_Pagamento 25 x

Filtra mensalidades com **valor maior** que R\$ 500.

SELECT * FROM Mensalidade WHERE valor_mensalidade > 500;

	cod_mensalidade	valor_mensalidade	cod_aluno	cod_secretaria
▶	7	510.00	7	7
	8	530.00	8	8
	9	520.00	9	9
	10	540.00	10	10
*	NULL	NULL	NULL	NULL

5. 3 consultas com operadores lógicos (AND, OR)

Mostra alunos que Têm a letra "a" no nome e Estão com o professor de `cod_professor = 1`.

```
SELECT * FROM Aluno WHERE nome LIKE '%a%' AND cod_professor = 1;
```

	cod_aluno	nome	cpf	cod_professor
▶	1	João Pedro	111.111.111-11	1
*	NULL	NULL	NULL	NULL

Mostra mensalidades com **valor acima de R\$ 450** ou com secretaria 3.

```
SELECT * FROM Mensalidade WHERE valor_mensalidade > 450 OR cod_secretaria = 3;
```

	cod_mensalidade	valor_mensalidade	cod_aluno	cod_secretaria
▶	1	500.00	1	1
	2	480.00	2	2
	3	470.00	3	3
	4	460.00	4	4
	6	490.00	6	6

Mensalidade31 x

Filtra pagamentos **altos e feitos em 1º de julho de 2025**.

```
SELECT * FROM Efetua_Pagamento WHERE valor_pago > 2800 AND data_pagamento = '2025-07-01';
```

	id_pagamento	cod_professor	cod_secretaria	valor_pago	data_pagament
▶	1	1	1	3000.00	2025-07-01
	4	4	4	3100.00	2025-07-01
	6	6	6	2900.00	2025-07-01
	7	7	7	3000.00	2025-07-01
	8	8	8	3200.00	2025-07-01

Efetua_Pagamento 32 v

6. 2 consultas com operadores lógicos e negação (NOT)

Mostra todos os alunos **exceto** "João Pedro".

```
SELECT * FROM Aluno WHERE NOT nome LIKE '%João Pedro%';
```

	cod_aluno	nome	cpf	cod_professor
▶	2	Mariana Lima	222.222.222-22	2
	3	Lucas Rocha	333.333.333-33	3
	4	Beatriz Souza	444.444.444-44	4
	5	Rafael Silva	555.555.555-55	5
	6	Amanda Teixeira	666.666.666-66	6

Aluno 35

Exclui mensalidades entre R\$ 460 e R\$ 500.

```
SELECT * FROM Mensalidade WHERE NOT valor_mensalidade BETWEEN 460 AND 500;
```

	cod_mensalidade	valor_mensalidade	cod_aluno	cod_secretaria
▶	5	450.00	5	5
	7	510.00	7	7
	8	530.00	8	8
	9	520.00	9	9
	10	540.00	10	10

Mensalidade 36

7. 3 consultas com operadores auxiliares (IS NULL, BETWEEN, LIKE, IN)

Mostra alunos que **têm CPF preenchido**.

```
SELECT * FROM Aluno WHERE cpf IS NOT NULL;
```

	cod_aluno	nome	cpf	cod_professor
▶	1	João Pedro	111.111.111-11	1
	2	Mariana Lima	222.222.222-22	2
	3	Lucas Rocha	333.333.333-33	3
	4	Beatriz Souza	444.444.444-44	4
	5	Rafael Silva	555.555.555-55	5

Aluno 39

Mostra mensalidades entre **R\$ 480 e R\$ 520**.

```
SELECT * FROM Mensalidade WHERE valor_mensalidade BETWEEN 480 AND 520;
```

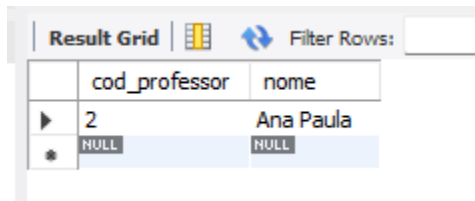
	cod_mensalidade	valor_mensalidade	cod_aluno	cod_secretaria
▶	1	500.00	1	1
	2	480.00	2	2
	6	490.00	6	6
	7	510.00	7	7
	9	520.00	9	9

Mensalidade 40

Output

Mostra professores cujo **nome começa com "A"**.

```
SELECT * FROM Professor WHERE nome LIKE 'A%';
```

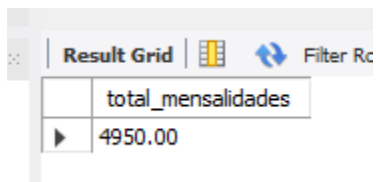


	cod_professor	nome
▶	2	Ana Paula
*	NULL	NULL

8. 3 consultas com funções de agregação (SUM(), AVG(), etc.)

Soma todas as mensalidades do banco.

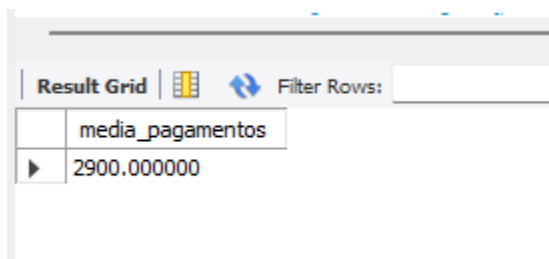
```
SELECT SUM(valor_mensalidade) AS total_mensalidades FROM Mensalidade;
```



	total_mensalidades
▶	4950.00

Mostra o valor **médio** dos pagamentos feitos.

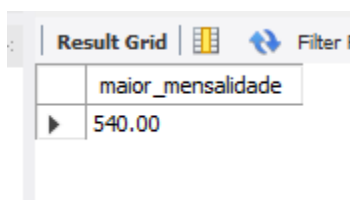
```
SELECT AVG(valor_pago) AS media_pagamentos FROM Efetua_Pagamento;
```



	media_pagamentos
▶	2900.000000

Mostra a **maior mensalidade** cadastrada.

```
SELECT MAX(valor_mensalidade) AS maior_mensalidade FROM Mensalidade;
```

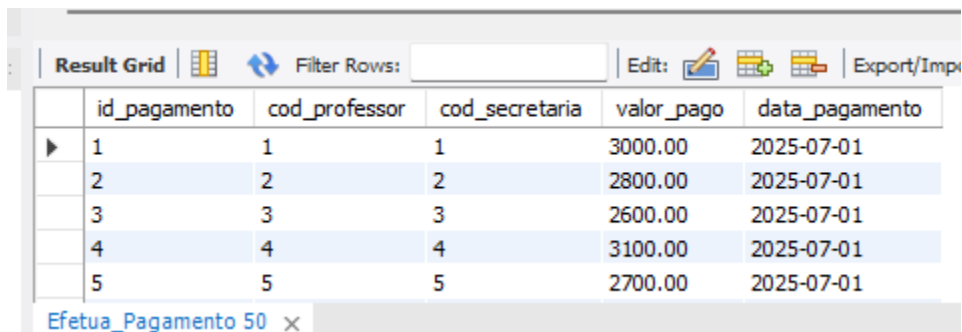


	maior_mensalidade
▶	540.00

9. 2 consultas com funções de datas (NOW(), DATE(), YEAR(), etc.)

Filtra os pagamentos feitos **no ano de 2025**.

```
SELECT * FROM Efetua_Pagamento WHERE YEAR(data_pagamento) = 2025; SELECT *
FROM Efetua_Pagamento WHERE YEAR(data_pagamento) = 2025;
```



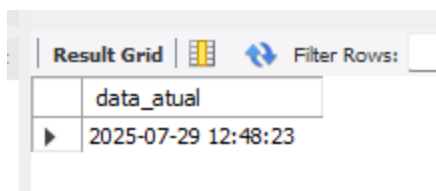
The screenshot shows a database interface with a 'Result Grid' tab. The grid displays five rows of payment data for the year 2025. The columns are: id_pagamento, cod_professor, cod_secretaria, valor_pago, and data_pagamento. The data is as follows:

	id_pagamento	cod_professor	cod_secretaria	valor_pago	data_pagamento
▶	1	1	1	3000.00	2025-07-01
	2	2	2	2800.00	2025-07-01
	3	3	3	2600.00	2025-07-01
	4	4	4	3100.00	2025-07-01
	5	5	5	2700.00	2025-07-01

Below the grid, there is a tab labeled 'Efetua_Pagamento 50' with a close button (x).

Mostra a data e hora **atuais do sistema**.

```
SELECT NOW() AS data_atual;
```



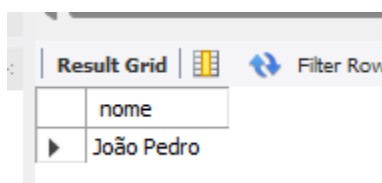
The screenshot shows a database interface with a 'Result Grid' tab. The grid displays a single row of data for the current system date and time. The column is: data_atual. The data is as follows:

	data_atual
▶	2025-07-29 12:48:23

10.3 sub-consultas com agrupamento e união de dados

Mostra nomes de alunos que têm professores com sobrenome "**Silva**".

```
SELECT nome FROM Aluno WHERE cod_professor IN (
    SELECT cod_professor FROM Professor WHERE nome LIKE '%Silva%'
);
```

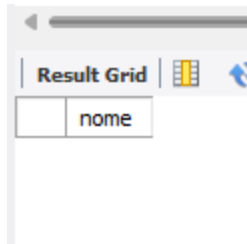


The screenshot shows a database interface with a 'Result Grid' tab. The grid displays a single row of data for the name of a student. The column is: nome. The data is as follows:

	nome
▶	João Pedro

Mostra professores que têm mais de **1 aluno**. (Nenhum professor foi cadastrado com mais de um aluno)

```
SELECT nome FROM Professor WHERE cod_professor IN ( SELECT cod_professor
FROM Aluno GROUP BY cod_professor HAVING COUNT(*) > 1 );
```



Junta os nomes dos alunos e professores (sem repetição).

```
SELECT nome FROM Aluno UNION SELECT nome FROM Professor;
```



11.3 consultas com JOIN e visualização de tabelas

Junta alunos e professores, mostrando quem está com quem.

```
SELECT a.nome AS aluno, p.nome AS professor FROM Aluno a JOIN Professor p ON
a.cod_professor = p.cod_professor;
```


Result Grid			Filter Rows:
	aluno	professor	
▶	João Pedro	Carlos Silva	
	Mariana Lima	Ana Paula	
	Lucas Rocha	Ricardo Gomes	
	Beatriz Souza	Marina Costa	
	Rafael Silva	José Martins	

Result 64 x

Mostra mensalidade com o nome do aluno correspondente.

```
SELECT m.valor_mensalidade, a.nome AS aluno FROM Mensalidade m JOIN Aluno a
ON m.cod_aluno = a.cod_aluno;
```

Result Grid			Filter Rows:
	valor_mensalidade	aluno	
▶	500.00	João Pedro	
	480.00	Mariana Lima	
	470.00	Lucas Rocha	
	460.00	Beatriz Souza	
	450.00	Rafael Silva	

Result 65 x

Mostra os pagamentos junto com o nome do professor envolvido.

```
SELECT ep.valor_pago, p.nome AS professor FROM Efetua_Pagamento ep JOIN
Professor p ON ep.cod_professor = p.cod_professor;
```

Result Grid			Filter Rows:
	valor_pago	professor	
▶	3000.00	Carlos Silva	
	2800.00	Ana Paula	
	2600.00	Ricardo Gomes	
	3100.00	Marina Costa	
	2700.00	José Martins	

Result 66 x

12.4 consultas com tipos de JOIN: INNER, LEFT, RIGHT

Só mostra quando há correspondência nas duas tabelas

```
SELECT a.nome, p.nome AS professor FROM Aluno a INNER JOIN Professor p ON
a.cod_professor = p.cod_professor;
```

Result Grid | Filter Rows:

	nome	professor
▶	João Pedro	Carlos Silva
	Mariana Lima	Ana Paula
	Lucas Rocha	Ricardo Gomes
	Beatriz Souza	Marina Costa
	Rafael Silva	José Martins

Result 67 x

Mostra todos os alunos, mesmo sem mensalidade

```
SELECT a.nome, m.valor_mensalidade FROM Aluno a LEFT JOIN Mensalidade m ON
a.cod_aluno = m.cod_aluno;
```

Result Grid | Filter Rows:

	nome	valor_mensalidade
▶	João Pedro	500.00
	Mariana Lima	480.00
	Lucas Rocha	470.00
	Beatriz Souza	460.00
	Rafael Silva	450.00

Result 69 x

Mostra todas as mensalidades, mesmo se aluno não existir

```
SELECT m.valor_mensalidade, a.nome FROM Mensalidade m RIGHT JOIN Aluno a ON
m.cod_aluno = a.cod_aluno;
```

Result Grid | Filter Rows:

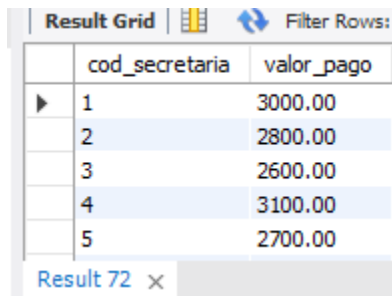
	valor_mensalidade	nome
▶	500.00	João Pedro
	480.00	Mariana Lima
	470.00	Lucas Rocha
	460.00	Beatriz Souza
	450.00	Rafael Silva

Result 71 x

Output

Mostra todas as secretarias, mesmo se não tiverem pagamentos

```
SELECT s.cod_secretaria, ep.valor_pago FROM Secretaria s LEFT JOIN  
Efetua_Pagamento ep ON s.cod_secretaria = ep.cod_secretaria;
```



The image shows a screenshot of a database query result grid. The grid has two columns: 'cod_secretaria' and 'valor_pago'. There are five rows of data. The first row has a value of 1 for 'cod_secretaria' and 3000.00 for 'valor_pago'. The second row has 2 and 2800.00. The third row has 3 and 2600.00. The fourth row has 4 and 3100.00. The fifth row has 5 and 2700.00. The grid is titled 'Result Grid' and has a 'Filter Rows' button. Below the grid, it says 'Result 72' with a close button.

	cod_secretaria	valor_pago
▶	1	3000.00
	2	2800.00
	3	2600.00
	4	3100.00
	5	2700.00

Result 72 x