



Estácio

Universidade Estácio de Sá

Curso Desenvolvimento Full Stack - Turma 9001

**Disciplina: RPG0015 - Nível 2: Vamos Manter as
Informações?**

Semestre Letivo: 2024.4

Repositorio Git:

<https://github.com/SashaCardoso/Missao-N2-M3-Estacio>

【SASHA CARDOSO】

Missão Prática | Nível 2 | Mundo 3

Objetivo: Modelagem e implementação de um banco de dados simples.

2º Procedimento

Todos os códigos estão disponíveis no repositório Github
linkado acima.

Resultados da execução:

Language: English

MySQL » db » shop » SQL command

Adminer 4.8.1

DB: shop

SQL command [Import](#)
[Export](#) [Create table](#)

[select companies](#)
[select individuals](#)
[select products](#)
[select transactions](#)
[select users](#)

SQL command

SELECT * FROM companies INNER JOIN users ON companies.user_id = users.id

user_id	cnpj	id	login	password	name	address	email	phone
1	12345678000199	1	johndoe	password123	John Doe	123 Main St, Cityville, Country	johndoe@email.com	123-456-7890
2	23456789000188	2	janedoe	mypassword	Jane Doe	456 Oak St, Townsville, Country	janedoe@email.com	234-567-8901
3	34567890000177	3	bobjohnson	securepassword	Bob Johnson	789 Pine St, Villagetown, Country	bobjohnson@email.com	345-678-9012
4	45678901000166	4	emilywhite	strongpassword	Emily White	101 Maple St, Hamlet, Country	emilywhite@email.com	456-789-0123

4 rows (0.000 s) [Edit](#), [Explain](#), [Export](#)

SELECT * FROM companies INNER JOIN users ON companies.user_id = users.id

Limit rows: ☐ Stop on error ☐ Show only errors

[History](#)

Language: English

MySQL » db » shop » SQL command

Adminer 4.8.1

DB: shop

SQL command [Import](#)
[Export](#) [Create table](#)

[select companies](#)
[select individuals](#)
[select products](#)
[select transactions](#)
[select users](#)

SQL command

SELECT * FROM individuals INNER JOIN users ON individuals.user_id = users.id

user_id	cpf	id	login	password	name	address	email	phone
1	12345678901	1	johndoe	password123	John Doe	123 Main St, Cityville, Country	johndoe@email.com	123-456-7890
2	23456789012	2	janedoe	mypassword	Jane Doe	456 Oak St, Townsville, Country	janedoe@email.com	234-567-8901
3	34567890123	3	bobjohnson	securepassword	Bob Johnson	789 Pine St, Villagetown, Country	bobjohnson@email.com	345-678-9012
4	45678901234	4	emilywhite	strongpassword	Emily White	101 Maple St, Hamlet, Country	emilywhite@email.com	456-789-0123

4 rows (0.001 s) [Edit](#), [Explain](#), [Export](#)

SELECT * FROM individuals INNER JOIN users ON individuals.user_id = users.id

Limit rows: ☐ Stop on error ☐ Show only errors

[History](#)

Language: English ▼

MySQL » db » shop » SQL command

Adminer 4.8.1

DB: shop ▼

SQL command [Import](#)
[Export](#) [Create table](#)

[select companies](#)
[select individuals](#)
[select products](#)
[select transactions](#)
[select users](#)

SQL command

```
SELECT p.name AS product_name,  
       SUM(t.quantity * p.price) AS total_value  
FROM transactions t  
JOIN products p ON t.product_id = p.id  
GROUP BY p.id
```

product_name	total_value
Product A	21.98
Product B	102.5
Product C	15.75
Product D	90

4 rows (0.000 s) [Edit](#), [Explain](#), [Export](#)

```
SELECT p.name AS product_name,  
       SUM(t.quantity * p.price) AS total_value  
FROM transactions t  
JOIN products p ON t.product_id = p.id  
GROUP BY p.id;
```

Limit rows: ☐ Stop on error ☐ Show only errors

[History](#)

Language: English ▾

MySQL » db » shop » SQL command

Adminer 4.8.1

DB: shop ▾

SQL command [Import](#)
[Export](#) [Create table](#)

[select companies](#)
[select individuals](#)
[select products](#)
[select transactions](#)
[select users](#)

SQL command

```
SELECT p.name AS product_name,  
       SUM(t.quantity * p.price) / SUM(t.quantity) AS weighted_mean_price  
FROM transactions t  
JOIN products p ON t.product_id = p.id  
GROUP BY p.id
```

product_name	weighted_mean_price
Product A	10.99
Product B	20.5
Product C	15.75
Product D	30

4 rows (0.000 s) [Edit](#), [Explain](#), [Export](#)

```
SELECT p.name AS product_name,  
       SUM(t.quantity * p.price) / SUM(t.quantity) AS weighted_mean_price  
FROM transactions t  
JOIN products p ON t.product_id = p.id  
GROUP BY p.id;
```

Limit rows: ☐ Stop on error ☐ Show only errors

[History](#)

Análise e Conclusão

1. Quais as diferenças no uso de *sequence* e *identity*?

- *Sequence* e *identity* são ambas utilizadas para gerar um número automaticamente, mas a principal diferença é que a *identity* depende da tabela e a *sequence* é independente da tabela.

2. Qual a importância das chaves estrangeiras para a consistência do banco?

- *Foreign Keys* providenciam uma conexão entre tabelas, organizando uma estrutura robusta ao banco.

3. Quais operadores do SQL pertencem à álgebra relacional e quais são definidos no cálculo relacional?

- *SELECT* e *JOIN* são exemplos de álgebra relacional, enquanto *EQUALS* (=) e *LESS THAN* (<) são exemplos de cálculo relacional.

4. Como é feito o agrupamento em consultas, e qual requisito é obrigatório?

- É feito usando *GROUP BY*, e o requisito é que *GROUP BY* sempre venha depois de *FROM* e *WHERE*.

