

# Banco de Dados

# Grand Finale

GRUPO: HUMBERTO CARNEIRO, LUCCA MILANO, RIAN DELOU  
E VICTOR SANTOS

# Temas

**1 Triggers**

**2 Procedures**

**3 Views**

**4 Segurança**

**5 CRUD**

# Triggers

4 Triggers criados:

- atualizarNumeroCliente()
- atualizarValorDesconto()
- atualizarValorProduto()
- removerCliente()

```
-- trigger 1 (trigger para atualizar o valor do produto)
```

```
CREATE TRIGGER tg_atualiza_valor_produto
AFTER INSERT OR UPDATE ON PRODUTO
FOR EACH ROW
EXECUTE PROCEDURE atualizarValorProduto();
```

```
UPDATE PRODUTO SET valor= 15 WHERE idProduto= 2001;
UPDATE PRODUTO SET valor= 20 WHERE idProduto= 2004;
```

```
SELECT * FROM PRODUTO;
```

```
SELECT * FROM ATT_PRODUTO;
```

```
-- trigger 2
```

```
-- Função para trigger 2
```

```
CREATE OR REPLACE FUNCTION atualizarValorDesconto()
RETURNS trigger AS $$
BEGIN
    IF (OLD.valorDesconto IS DISTINCT FROM NEW.valorDesconto) THEN
        INSERT INTO ATT_cupomDesconto (idcupomdesconto, descricaoDesconto, valorDesconto, data_modificacao)
        VALUES (NEW.idcupomdesconto, NEW.descricaoDesconto, NEW.valorDesconto, NOW());
    END IF;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

nome	valor	avaliacaomedia	idproduto	totalvendas
eBook2	12.31	4	2002	231
acessório1	12.31	4.5	2003	231
curso1	12.31	1.5	2005	142
curso2	12.31	3.5	2006	104
curso3	12.31		2007	104
curso4	12.31		2008	104
curso5	12.31		2009	104
curso6	12.31		2010	104
eBook1	15	3	2001	100
acessório2	20	0.5	2004	100

(10 rows)

idproduto	nome	valor	data_modificacao
2001	eBook1	15	2024-06-10 01:47:58.125304
2004	acessório2	20	2024-06-10 01:47:58.126535

(2 rows)

# Procedures

4 Procedures criados:

- trocar\_ead()
- trocar\_obrigatoriedade()
- qtd\_alunos()
- qtd\_funcionarios()

```
-- procedure 1 (atualiza uma disciplina de presencial para remoto, ou vice-versa)
```

```
SELECT * FROM DISCIPLINA;  
CREATE OR REPLACE PROCEDURE trocar_ead()  
LANGUAGE plpgsql  
AS $$  
BEGIN  
  
    UPDATE DISCIPLINA  
    SET Remota = true WHERE idDisciplina = 2;  
  
    UPDATE DISCIPLINA  
    SET Presencial = false WHERE idDisciplina = 2;
```

```
END $$;
```

```
CALL trocar_ead();  
SELECT * FROM DISCIPLINA;
```

iddisciplina	presencial	remota	eletiva	obrigatoria
1	t	f	f	t
2	t	f	f	t
3	t	f	f	t
4	t	f	f	t
5	t	f	f	t
6	t	f	f	t

(6 rows)

```
CREATE PROCEDURE
```

```
CALL
```

iddisciplina	presencial	remota	eletiva	obrigatoria
1	t	f	f	t
3	t	f	f	t
4	t	f	f	t
5	t	f	f	t
6	t	f	f	t
2	f	t	f	t

(6 rows)

# Views

4 Views criados:

- Cliente\_Pedido\_info
- Cliente\_Removidos
- Pedidos\_Metodos\_Pagamento
- Produtos\_Mais\_Vendidos

Cliente\_Pedido\_info

GeneralDefinitionCodeSecuritySQL

1

2

3

4

5

6

7

SELECT

c.idcliente,

c.nome,

c.email,

count(p.idpedido) AS totalvendas

FROM cliente c

LEFT JOIN pedido p ON c.idcliente = p.idcliente

GROUP BY c.idcliente, c.nome, c.email;

Data OutputMessagesNotifications

≡+

▼

▼

SQL

	idcliente integer	nome character varying	email character varying	totalvendas bigint
1	5	Cliente Dois	cliente2@email.com	1
2	6	Cliente Três	cliente3@email.com	0
3	2	Cliente 2	cliente2@example.com	3
4	7	Cliente Quatro	cliente4@email.com	0
5	1	Cliente 1	cliente1@example.com	3
6	8	Cliente Cinco	cliente5@email.com	0
7	3	Cliente 3	cliente3@example.com	2
8	1000	luca	luca@gmail	0

# Segurança


```
-- SEGURANÇA
```

```
CREATE ROLE usuario_admin WITH PASSWORD 'admin123';  
CREATE ROLE usuario_gerente WITH PASSWORD 'gerente123';  
CREATE ROLE usuario_funcionario WITH PASSWORD 'funcionario123';  
CREATE ROLE usuario_padrao WITH PASSWORD 'usuario123';
```

```
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO usuario_admin;  
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO usuario_gerente;  
GRANT SELECT, INSERT, UPDATE ON CLIENTE, PEDIDO TO usuario_funcionario;  
GRANT SELECT ON CLIENTE, PEDIDO TO usuario_padrao;
```

4 usuários criados:

- usuario\_admin
- usuario\_gerente
- usuario\_funcionario
- usuario\_padrao

	rolname name	
1	pg_database_owner	
2	pg_read_all_data	
3	pg_write_all_data	
4	pg_monitor	
5	pg_read_all_settings	
6	pg_read_all_stats	
7	pg_stat_scan_tables	
8	pg_read_server_files	
9	pg_write_server_files	
10	pg_execute_server_program	
11	pg_signal_backend	
12	pg_checkpoint	
13	pg_use_reserved_connections	
14	pg_create_subscription	
15	postgres	
16	usuario_admin	
17	usuario_gerente	
18	usuario_funcionario	
19	usuario_padrao	



**ADONIS**

**CRUD**

**Prática!**

lucidadonisjs.com/docs/crud-operations

GoogleRDelou63's ProfileTurmasDI3DCaixa de entrada (1...EAQ - Brasil Banda...Todos os marcadores

lucid

Search

Transactions

Pagination

Database seeders

Query builders

Select query builder

Insert query builder

Raw query builder

Migrations

Introduction

Schema builder

Table builder

Models

Introduction

CRUD operations

Hooks

Query builder

Naming strategy

Query scopes

Serializing models

Relationships

Model factories

## Using the query builder

The above-mentioned static methods cover the common use cases for querying the database. However, you are not only limited to these methods and can also leverage the query builder API for making advanced SQL queries.

The `ModelQueryBuilder` returns an array of model instances and not the plain JavaScript object(s).

You can get an instance of a query builder for your model using the `.query` method.

```
const users = await User
  .query()
  .where('countryCode', 'IN')
  .orWhereNull('countryCode')
```

To fetch a single row, you can make use of the `.first` method. There is also a `firstOrFail` method.

```
const users = await User
  .query()
  .where('countryCode', 'IN')
  .orWhereNull('countryCode')
  .first()
```

## Update

USD/EUR  
+0.28%

Search

11:11 PM  
2024-06-09

bd-logical-physical

EXPLORER

BD-LOGICAL-PHYSICAL

app

controllers

TS professores\_contr...

TS produtos\_control...

TS psicologos\_contr...

exceptions

middleware

models

TS cliente.ts

TS produto.ts

TS professor.ts

TS psicologo.ts

bin

TS console.ts

TS server.ts

TS test.ts

config

database

node\_modules

start

TS env.ts

TS kernel.ts

TS routes.ts

tests

tmp

.editorconfig

.env

.env.example

ace.js

adonisrc.ts

package-lock.json

package.json

tsconfig.json

Prints

Conceitual\_Definitivo...

Fisico\_Definitivo.txt

Lógico\_Definitivo.brM3

README.md

OUTLINE

TIMELINE

CRUD-adonisjs > app > controllers > TS professores\_controller.ts > ProfessorsController > update

4 export default class ProfessorsController {

20 async show({}: HttpContext) {

22 return professor

23 }

24 }

25 async edit({}: HttpContext) {}

26 }

27 async update({}: HttpContext) {

28 const professor = await Professor.findOrFail(1) // CRUD "U" -> UPDATE P

29 professor.nome = 'CARAMBAAA!! ELE N TEM DIPLOMA N TEM NOME TB'

30 professor.diploma = false

31 console.log('update!')

32 await professor.save()

33 }

34 }

35 async destroy({}: HttpContext) {

36 const professor = await Professor.findOrFail(2) // CRUD "D" -> DELETE P

37 await professor.delete()

38 }

39 console.log('delete!')

40 }

41 }

42 }

PROBLEMS

OUTPUT

DEBUG-CONSOLE

TERMINAL

PORTS

[ info ] starting HTTP server...

Server address: http://localhost:3333

Watch Mode: HMR

Ready in: 528 ms

[02:00:09.859] INFO (18288): started HTTP server on localhost:3333

invalidated app\controllers\professors\_controller.ts

invalidated app\controllers\professors\_controller.ts

invalidated app\controllers\professors\_controller.ts

[02:09:23.667] WARN (18288): Row not found

request\_id: "og3ls9bvmosyenxygy9j3w41"

x-request-id: "og3ls9bvmosyenxygy9j3w41"

invalidated app\controllers\professors\_controller.ts

delete!

Ln 33, Col 4 Spaces: 2 UTF-8 LF TypeScript Go Live Prettier