

atividades - Junções, Views, Subconsultas e Check

a) moodJunções necessárias (inner join, left join e right join)

- Ao acessar as categorias de lugares é exibido o **nome**, a **descrição** e o **link** da imagem referente a categoria. Logo, são acessados dados das tabelas categoria e imagem:

```
SELECT nome_categoria, descricao, link
```

```
from categoria
```

```
inner join imagem
```

```
on (categoria.id_imagem = imagem.id_imagem);
```

HIT on postgres@PostgreSQL 9.5

1

2

3

4

5

```
SELECT nome_categoria, descricao, link
from categoria
inner join imagem
on (categoria.id_imagem = imagem.id_imagem);
```

Data Output

Explain

Messages

History

	nome_categoria character varying (256)	descricao character varying (256)	link character varying (256)
1	Shopping	Centro comercial com lo...	link_Imagem1_Shopping
2	Praça	Espaço urbano para co...	link_Imagem4_Praca
3	Praia	Area recreativa para ba...	link_Imagem3_Praia
4	Lanchonete	Estabelecimento comer...	link_Imagem6_Lanchon...
5	Supermercado	Comércio com grande q...	link_Imagem5_Superm...
6	Bar	Estabelecimento comer...	link_Imagem7_Bar

- Ao selecionar uma categoria são exibidos os lugares, sendo apresentados o **nome**, **avaliação geral** quando possuir (obtida da média das avaliações individuais), **imagem** do lugar e a **localização** (latitude e longitude do lugar):

```
SELECT nome_lugar, link, lat, long, AVG(nota) as Nota
```

```
from lugar
```

```
inner join imagem
```

```
on (lugar.id_lugar = imagem.id_lugar)
```

```
left outer join avaliacao
```

```
on (avaliacao.id_lugar = lugar.id_lugar)
```

```

WHERE lugar.id_categoria IN (
                                select distinct id_categoria as ID_Categoria
                                from categoria
                                where id_categoria = 1
                                )
GROUP BY nome_lugar, link, lat, long;

```

HIT on postgres@PostgreSQL 9.5

```

1  from lugar
2
3  inner join imagem
4  on (lugar.id_lugar = imagem.id_lugar)
5  left outer join avaliacao
6  on (avaliacao.id_lugar = lugar.id_lugar)
7  WHERE lugar.id_categoria IN (
8                                select distinct id_categoria as ID_Categoria
9                                from categoria
10                               where id_categoria = 1
11                               )
12  GROUP BY nome_lugar, link, lat, long;
13
14

```

	nome_lugar character varying (256)	link character varying (256)	lat real	long real	nota numeric
1	Shopping Vitória	link_Imagem14_ShoppingVitoria2	-20.3127	-40.2878	4.5000000000000000
2	Shopping Vila Velha	link_Imagem18_ShoppingVilaVelha1	-20.3526	-40.3002	[null]
3	Shopping Vitória	link_Imagem13_ShoppingVitoria1	-20.3127	-40.2878	4.5000000000000000
4	Shopping Mestre Alvaro	link_Imagem16_ShoppingMestreAlvaro1	-20.2402	-40.2771	[null]
5	Shopping Montserrat	link_Imagem17_ShoppingMontserrat1	-20.193	-40.2684	[null]
6	Boulevard Shopping Vil...	link_Imagem19_BoulevardShoppingVV1	-20.3922	-40.3203	[null]
7	Shopping Laranjeiras	link_Imagem20_ShoppingLaranjeiras1	-20.1969	-40.2572	[null]

- Ao selecionar o lugar, é exibida a tela de avaliação do lugar sendo exibido os comentários individuais, sendo compostos de **nome da pessoa**, **imagem da pessoa**, **nota** atribuída ao lugar pela pessoa e o **comentário**:

```

SELECT nome_pessoa, link, comentario, nota
from pessoa
inner join imagem
on(pessoa.id_imagem = imagem.id_imagem)
inner join avaliacao
on(avaliacao.id_pessoa = pessoa.id_pessoa)
WHERE avaliacao.id_lugar IN(
                                select distinct id_lugar as ID_Lugar
                                from lugar
                                where id_lugar = 1
                                );

```

HIT on postgres@PostgreSQL 9.5

```
1  SELECT nome_pessoa, link, comentario, nota
2  from pessoa
3  inner join imagem
4  on(pessoa.id_imagem = imagem.id_imagem)
5  inner join avaliacao
6  on(avaliacao.id_pessoa = pessoa.id_pessoa)
7  WHERE avaliacao.id_lugar IN(
8      select distinct id_lugar as ID_Lugar
9      from lugar
10     where id_lugar = 1
11 );
```

Data Output Explain Messages History

	nome_pessoa character varying (255)	link character varying (256)	comentario character varying (256)	nota integer
1	Jorge Silva	link_Imagem11_pessoa...	Shopping bem organiza...	4
2	Jean Costa	link_Imagem11_pessoa...	ótimo Shopping, maneir...	5

- Ao acessar a timeline do lugar são apresentadas as postagens, sendo mostradas informações de **nome da pessoa**, **imagem** da pessoa, **postagem** (texto e **imagem** quando houver) e **data** da postagem:

```
SELECT nome_pessoa, link, postagem, data
from postagem
inner join pessoa
on(postagem.id_pessoa = pessoa.id_pessoa)
inner join imagem
on(pessoa.id_imagem = imagem.id_imagem)
WHERE postagem.id_lugar IN(
    select distinct id_lugar as ID_Lugar
    from lugar
    where id_lugar = 12
);
```

HIT on postgres@PostgreSQL 9.5

```
1 SELECT nome_pessoa, link, postagem, data
2 from postagem
3 inner join pessoa
4 on(postagem.id_pessoa = pessoa.id_pessoa)
5 inner join imagem
6 on(pessoa.id_imagem = imagem.id_imagem)
7 WHERE postagem.id_lugar IN(
8         select distinct id_lugar as ID_Lugar
9         from lugar
10        where id_lugar = 12
11        );
```

Data Output

Explain

Messages

History

	nome_pessoa character varying (255)	link character varying (256)	postagem character varying (256)	data date
1	Eduardo Vasconcelos	link_Imagem10_pessoa4	Lugar maneiro, muito bom!!	2017-09-02

•

f)Check constraint

* Tabela Pessoa

```
CREATE TABLE public.pessoa
(
    email character varying(255) COLLATE pg_catalog."default" CONSTRAINT proper_email
CHECK (email ~* '^[A-Za-z0-9._%~]+@[A-Za-z0-9.-]+[.][A-Za-z]+$'),
    colaboracao integer,
    nome_pessoa character varying(255) COLLATE pg_catalog."default",
    autenticidade integer,
    senha character varying(20) COLLATE pg_catalog."default",
    login character varying(20) COLLATE pg_catalog."default",
    bairro character varying(100) COLLATE pg_catalog."default",
    cidade character varying(100) COLLATE pg_catalog."default",
    estado character varying(100) COLLATE pg_catalog."default",
    rua character varying(100) COLLATE pg_catalog."default",
    numero character varying(5) COLLATE pg_catalog."default",
    cep character varying(8) COLLATE pg_catalog."default" CONSTRAINT proper_cep CHECK
(cep ~* '^[0-9]{2}.[0-9]{3}-[0-9]{3}$'),
    ativo character(1) COLLATE pg_catalog."default",
    id_pessoa integer NOT NULL,
```

```

id_imagem integer,
)
CONSTRAINT pessoa_pkey PRIMARY KEY (id_pessoa),
CONSTRAINT pessoa_id_imagem_fkey FOREIGN KEY (id_imagem)
REFERENCES public.imagem (id_imagem) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION
)
WITH (
oids = false
)
TABLESPACE pg_default;

ALTER TABLE public.pessoa
OWNER to postgres;

```

*** Tabela Avaliação**

```

CREATE TABLE public.avaliacao
(
comentario character varying(256) COLLATE pg_catalog."default",
id_avaliacao integer NOT NULL,
nota integer check (nota >= 1 and nota <=5),
id_lugar integer,
id_pessoa integer,
CONSTRAINT avaliacao_pkey PRIMARY KEY (id_avaliacao),
CONSTRAINT avaliacao_id_lugar_fkey FOREIGN KEY (id_lugar)
REFERENCES public.lugar (id_lugar) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION,
CONSTRAINT avaliacao_id_pessoa_fkey FOREIGN KEY (id_pessoa)

```

```

REFERENCES public.pessoa (id_pessoa) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

```

```

ALTER TABLE public.avaliacao
    OWNER to postgres;

```

g)Views

```

create view categorias as
SELECT nome_categoria, descricao, link
from categoria
inner join imagem
on (categoria.id_imagem = imagem.id_imagem);

```

```

create view lugares as
SELECT nome_lugar, link, lat, long, AVG(nota) as Nota
from lugar
inner join imagem
on (lugar.id_lugar = imagem.id_lugar)
left outer join avaliacao
on (avaliacao.id_lugar = lugar.id_lugar)
WHERE lugar.id_categoria IN (

```

```

    select distinct id_categoria as ID_Categoria
    from categoria
    where id_categoria = 1

```

)

GROUP BY nome_lugar, link, lat, long;

create view avaliacao_lugar as

SELECT nome_pessoa, link, comentario, nota

from pessoa

inner join imagem

on(pessoa.id_imagem = imagem.id_imagem)

inner join avaliacao

on(avaliacao.id_pessoa = pessoa.id_pessoa)

WHERE avaliacao.id_lugar IN(

select distinct id_lugar as ID_Lugar

from lugar

where id_lugar = 1

);