

RESIDÊNCIA DE SOFTWARE



Projeto Banco de Dados

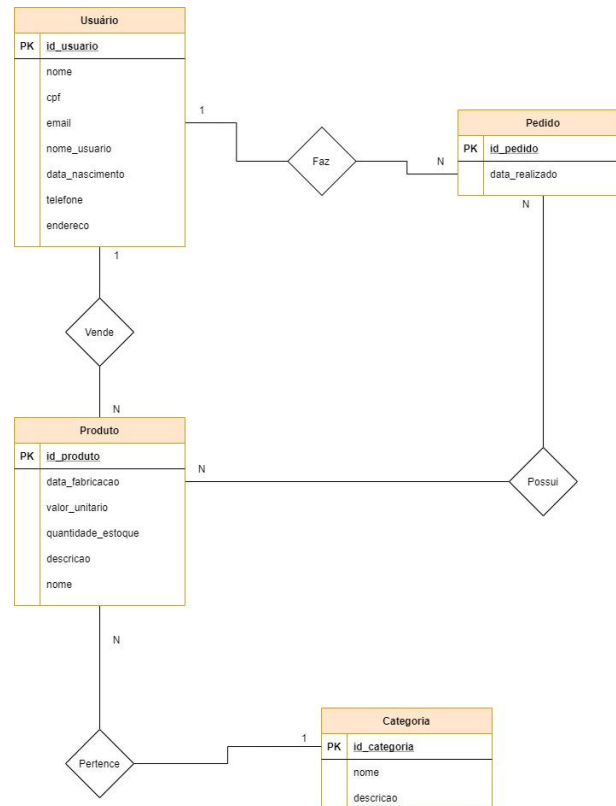
Grupo G2AC

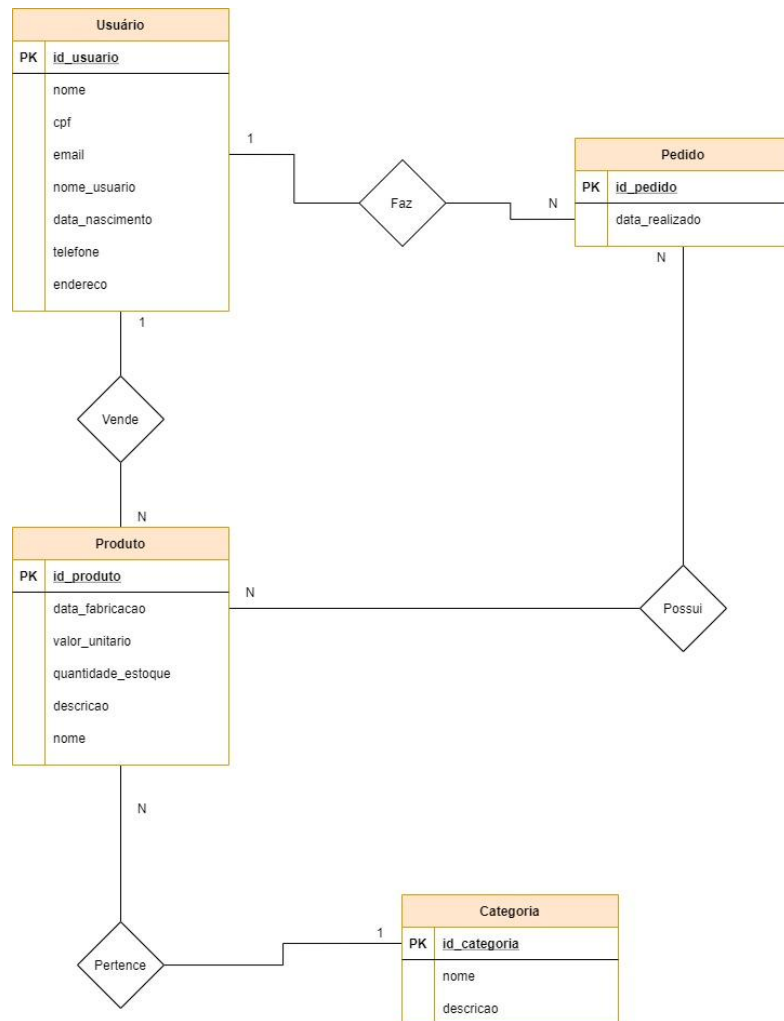


Modelo Entidade de Relacionamento

No MER, obtivemos o raciocínio a seguir:

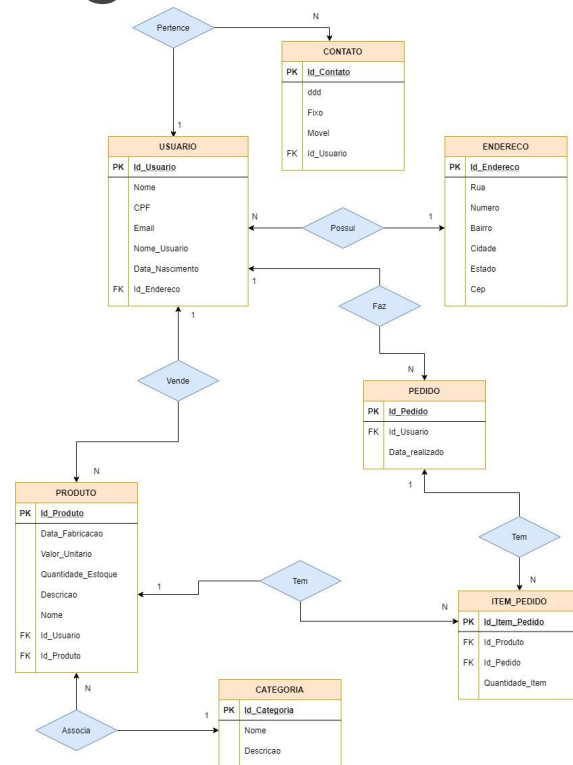
Usuário
Pedidos
Produto
Categoria.

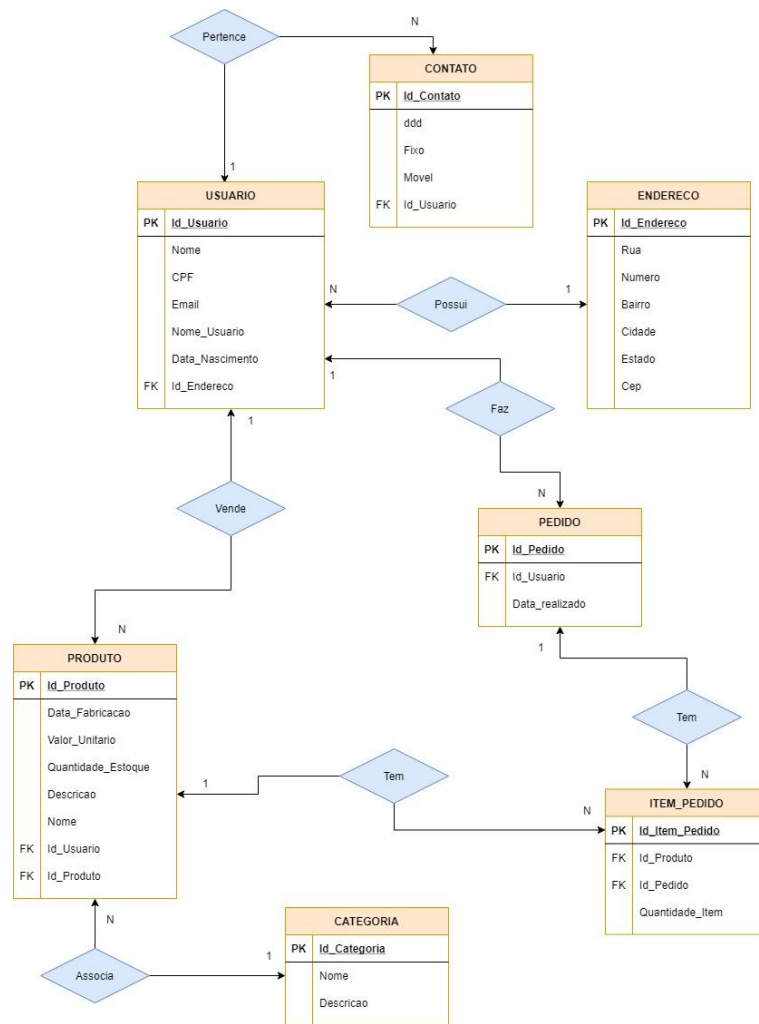




Normalização

Na normalização, tivemos a necessidade de criar novas tabelas, para se encaixar nas 3 formas normais.



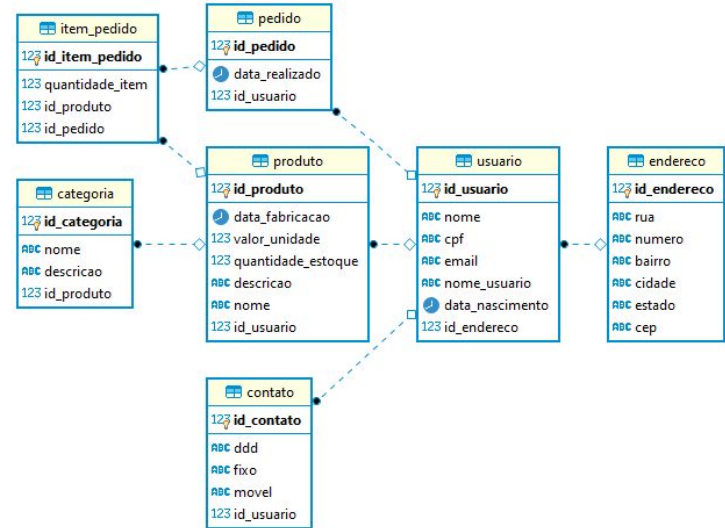


Banco de Dados

DDL - Linguagem de Definição de Dados

CREATE TABLE

- create table **ENDERECO**,
- create table **USUARIO**,
- create table **CONTATO**,
- create table **PRODUTO**,
- create table **CATEGORIA**,
- create table **PEDIDO**,
- create table **ITEM_PEDIDO**;




```
create table ENDERECO(  
  ID_ENDERECO SERIAL primary key not null,  
  RUA VARCHAR(40) not null,  
  NUMERO VARCHAR(15) not null,  
  BAIRRO VARCHAR(30) not null,  
  CIDADE VARCHAR(40) not null,  
  ESTADO CHAR(2) not null,  
  CEP CHAR(8) not null);
```

create table ENDERECO

```
create table USUARIO(  
    ID_USUARIO SERIAL primary key not null,  
    NOME VARCHAR not null,  
    CPF CHAR(11) unique not null,  
    EMAIL VARCHAR unique not null,  
    NOME_USUARIO VARCHAR(15) unique not null,  
    DATA_NASCIMENTO DATE not null,  
    ID_ENDERECO INTEGER not null,  
    foreign key(ID_ENDERECO) references ENDERECO);
```

create table USUARIO

```
create table CONTATO(  
    ID_CONTATO SERIAL primary key not null,  
    DDD CHAR(2) not null,  
    FIXO CHAR(8) UNIQUE,  
    MOVEL CHAR(9) UNIQUE,  
    ID_USUARIO INTEGER not null,  
    foreign key(ID_USUARIO) references USUARIO);
```

create table CONTATO

```
create table PRODUTO(  
    ID_PRODUTO SERIAL primary key not null,  
    DATA_FABRICACAO DATE not null,  
    VALOR_UNIDADE REAL not null,  
    QUANTIDADE_ESTOQUE INTEGER not null,  
    DESCRICAO VARCHAR(150),  
    NOME VARCHAR(30) not null,  
    ID_USUARIO INTEGER not null,  
    ID_CATEGORIA INTEGER not null,  
    foreign key(ID_USUARIO) references USUARIO,  
    foreign key(ID_CATEGORIA) references CATEGORIA);
```

create table PRODUTO

```
create table CATEGORIA(  
  ID_CATEGORIA SERIAL primary key not null,  
  NOME VARCHAR(30) not null,  
  DESCRICAO VARCHAR(150));
```

create table CATEGORIA

```
create table PEDIDO(  
    ID_PEDIDO SERIAL primary key not null,  
    DATA_REALIZADO TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP not null,  
    ID_USUARIO INTEGER not null,  
foreign key(ID_USUARIO) references USUARIO);
```

create table PEDIDO

```
create table ITEM_PEDIDO(  
    ID_ITEM_PEDIDO SERIAL primary key not null,  
    QUANTIDADE_ITEM INTEGER not null,  
    ID_PRODUTO INTEGER not null,  
    ID_PEDIDO INTEGER not null,  
foreign key(ID_PRODUTO) references PRODUTO,  
foreign key(ID_PEDIDO) references PEDIDO);
```

create table ITEM_PEDIDO

Banco de Dados

DML - Linguagem de Manipulação de Dados

Insert into

Populando as tabelas endereço, usuario, contato, categoria e produto.

```
insert into endereco (rua, numero, bairro, cidade, estado, cep)
values ('Ladeira Elisa Reuther da Silva', '65', 'Mosela', 'Petrópolis', 'RJ', '25675171'),
('Sergipe', '95', 'Quitandinha', 'Petrópolis', 'RJ', '25650490'),
('Vila Nicolau Antonio Mayworm', '36', 'Mosela', 'Petrópolis', 'RJ', '25675311'),
('Travessa José Lafaiete', '469', 'Mosela', 'Petrópolis', 'RJ', '25675155'),
('Siméria', '99', 'Siméria', 'Petrópolis', 'RJ', '25600000');

insert into usuario (nome, CPF, email, nome_usuario, data_nascimento, id_endereco)
values ('Abner Joia', '13129182764', 'abnerjoia1991@gmail.com', 'AbJoia', '1991-02-08', 1),
('Alessandro Felix', '65129415568', 'alessandro.felix.1973@gmail.com', 'Felix', '1973-06-01', 2),
('Cainã Machado', '80440771793', 'cwaxtd@gmail.com', 'Cwax', '2002-01-21', 3),
('Giulia Mendes', '16946186761', 'giulliamendes67@gmail.com', 'GiuliaMarcela', '1998-11-20', 4),
('Camila Ferreira Costa Silva', '00000000000', 'camilaferreira@gmail.com', 'CamilaFerreira', '1990-08-06', 5);

insert into contato (ddd, fixo, movel, id_usuario)
values ('24', null, '981573502', 1),
('24', '22358317', '992761939', 2),
('24', null, '988319994', 3),
('24', null, '988317240', 4),
('24', null, '000000000', 5);

insert into categoria (nome, descricao)
values ('Tecnologia', 'Informática, kindle'),
('Vestário', 'Roupas, Sapatos e Acessórios'),
('Música', 'Instrumentos Musicais, CDs e Equipamentos de Estúdio'),
('Livraria', 'Ebook, Livros e Revistas'),
('Automóvel', 'Carros e Motos'),
('Pet', 'Acessórios, Artigos de Pet Shop e Animais Domésticos');

insert into produto (data_fabricacao, nome, descricao, quantidade_estoque, valor_unidade, id_usuario, id_categoria)
values ('2017/02/20', 'Smartphone Galaxy A30', 'Aparelho em ótimo estado, com 2 meses de uso', '1', 800.00, 4, 1),
('2020/07/23', 'Jaqueta de Couro', 'Produto novo, 100% pele animal', '100', 600.00, 2, 2),
('1959/08/20', 'Guitarra Gibson Les Paul', 'Edição Jimmy Page 1959', '1', 30000.00, 1, 3),
('1943/04/18', 'Livro: O Pequeno Príncipe', 'Novo, sem marcas de uso', '20', 30.00, 4, 4),
('1971/11/02', 'Fusca 71', 'Cor laranja, em bom estado de conservação. Aceito moto de entrada', '1', 12990.00, 3, 5),
('2020/06/23', 'Cachorro', 'Completo com 4 patas, pedigree Golden, caramelo', '5', 2500.00, 5, 6);
```

Insert/update

Populando as tabelas pedido, item_pedido.

E atualizando a quantidade do estoque do usuário(vendedor).

Abner efetuou a compra de 10 jaquetas

```
insert into pedido (id_usuario)
values (1);
insert into item_pedido (quantidade_item , id_produto , id_pedido)
values (2, 4, 1);
update produto
set quantidade_estoque = quantidade_estoque - 2
where id_produto = 4;
```

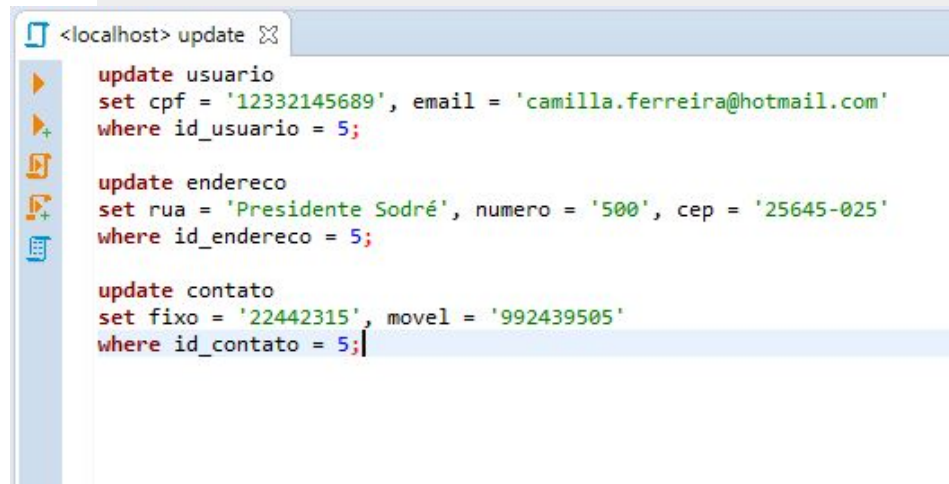
```
insert into pedido (id_usuario)
values (1);
insert into item_pedido (quantidade_item , id_produto , id_pedido)
values (10, 2, 2);
update produto
set quantidade_estoque = quantidade_estoque - 10
where id_produto = 2;
```

```
insert into pedido (id_usuario)
values (3);
insert into item_pedido (quantidade_item , id_produto , id_pedido)
values (1, 6, 3);
update produto
set quantidade_estoque = quantidade_estoque - 1
where id_produto = 6;
```

```
insert into pedido (id_usuario)
values (5);
insert into item_pedido (quantidade_item , id_produto , id_pedido)
values (1, 1, 4);
update produto
set quantidade_estoque = quantidade_estoque - 1
where id_produto = 1;
```

Update

Atualizando os dados da Camila.

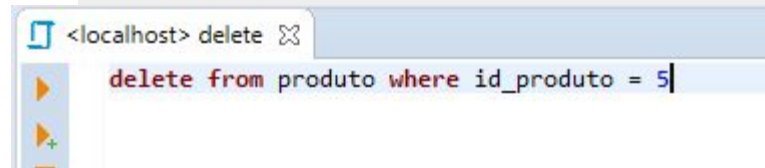


The screenshot shows a web browser window with a single tab titled "<localhost> update". The browser's address bar is empty. On the left side of the browser, there is a vertical toolbar with five icons: a play button, a plus sign, a document icon, a plus sign, and a calendar icon. The main content area of the browser displays three SQL update queries, each on a new line. The first query updates the 'usuario' table, setting 'cpf' to '12332145689' and 'email' to 'camilla.ferreira@hotmail.com' for 'id_usuario = 5'. The second query updates the 'endereco' table, setting 'rua' to 'Presidente Sodr ', 'numero' to '500', and 'cep' to '25645-025' for 'id_endereco = 5'. The third query updates the 'contato' table, setting 'fixo' to '22442315' and 'movel' to '992439505' for 'id_contato = 5'. The third query is highlighted with a light blue background.

```
<localhost> update  
  
update usuario  
set cpf = '12332145689', email = 'camilla.ferreira@hotmail.com'  
where id_usuario = 5;  
  
update endereco  
set rua = 'Presidente Sodr ', numero = '500', cep = '25645-025'  
where id_endereco = 5;  
  
update contato  
set fixo = '22442315', movel = '992439505'  
where id_contato = 5;
```

Delete

Deletando um produto.

A screenshot of a web browser's developer console. The top bar shows the URL as '<localhost> delete' with a close icon. The console area displays a SQL statement: 'delete from produto where id_produto = 5'. The text is color-coded: 'delete' is red, 'from' is blue, 'produto' is black, 'where' is red, 'id_produto' is black, and '5' is blue. The cursor is positioned at the end of the statement.

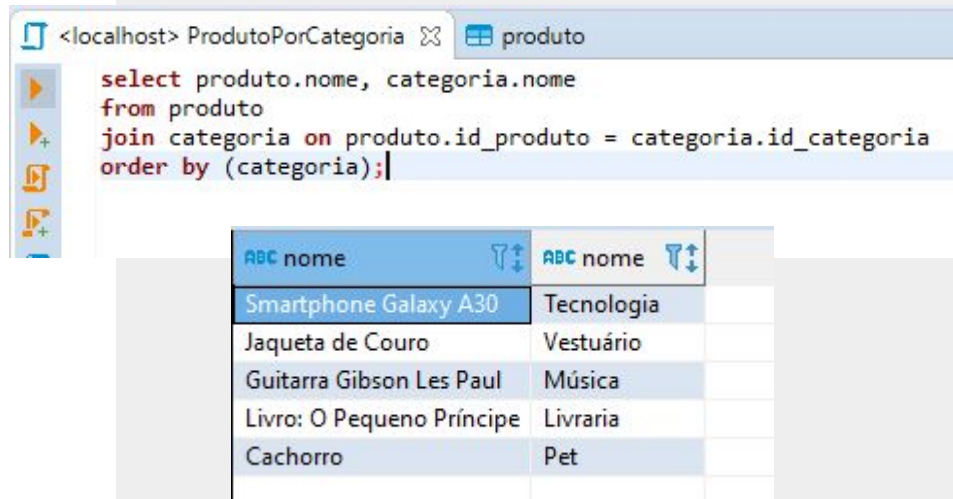
```
delete from produto where id_produto = 5
```

Banco de Dados

DQL - Linguagem de Consulta de Dados

Join

Retorna relação produto categoria
ordenando com order by



The screenshot shows a database query window with the title "<localhost> ProdutoPorCategoria" and a tab for the "produto" table. The query is as follows:

```
select produto.nome, categoria.nome
from produto
join categoria on produto.id_produto = categoria.id_categoria
order by (categoria);
```

Below the query, the results are displayed in a table with two columns, both labeled "ABC nome". The results are ordered by the category name.

ABC nome	ABC nome
Smartphone Galaxy A30	Tecnologia
Jaqueta de Couro	Vestuário
Guitarra Gibson Les Paul	Música
Livro: O Pequeno Príncipe	Livraria
Cachorro	Pet

Inner Join

Retorna relação de usuários e seus produtos anunciados com quantidade.

```
select p.quantidade_estoque as "Quantidade Estoque",  
       p.nome as "Produto", u.nome as "Vendedor"  
from produto p  
inner join usuario u  
on p.id_usuario = u.id_usuario;
```

123 Quantidade Estoque	ABC Produto	ABC Vendedor
1	Guitarra Gibson Les Paul	Abner Joia
90	Jaqueta de Couro	Alessandro Felix
1	Fusca 71	Cainã Machado
0	Smartphone Galaxy A30	Giulia Mendes
18	Livro: O Pequeno Príncipe	Giulia Mendes
1	Cachorro	Camila Ferreira Costa Silva

Count

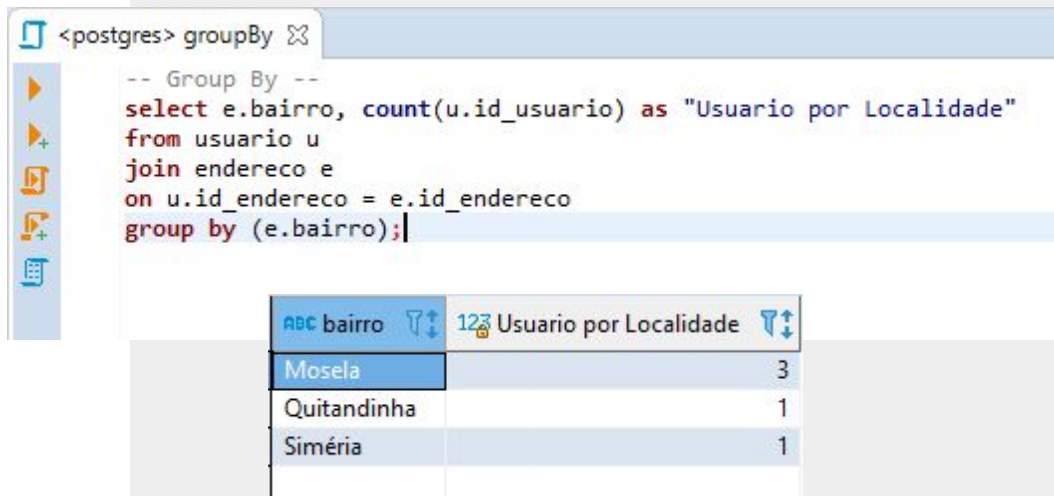
Contando quantos pedidos foram gerados no sistema.

```
select COUNT(id_pedido ) as "Quantidade de pedidos gerados"  
from pedido;
```

123 Quantidade de pedidos gerados
5

Group by

Retorna relação de usuários por localidade agrupado com group by



The screenshot shows a PostgreSQL query editor window titled "<postgres> groupBy". The query is as follows:

```
-- Group By --
select e.bairro, count(u.id_usuario) as "Usuario por Localidade"
from usuario u
join endereco e
on u.id_endereco = e.id_endereco
group by (e.bairro);
```

Below the query, the results are displayed in a table with two columns: "bairro" and "Usuario por Localidade". The results are grouped by neighborhood.

bairro	Usuario por Localidade
Mosela	3
Quitandinha	1
Siméria	1

Nota fiscal

Consulta SQL de Nota Fiscal

```
select pd.id_pedido as "Nº Pedido", p.nome as "Produto", p.valor_unidade as "Valor unitario",  
       ip.quantidade_item as "Quantidade", pd.data_realizado as "Data Emissão", u.nome as "Comprador(a)",  
       u.cpf, p.valor_unidade * ip.quantidade_item as "Total", v.nome as "Vendedor(a)"  
  from produto p  
 inner join usuario v  
    on p.id_usuario = v.id_usuario  
 inner join item_pedido ip  
    on p.id_produto = ip.id_produto  
 inner join pedido pd  
    on ip.id_pedido = pd.id_pedido  
 inner join usuario u  
    on pd.id_usuario = u.id_usuario  
 where pd.id_pedido = 2;
```

123 Nº Pedido	ABC Produto	123 Valor unitario	123 Quantidade	Data Emissão	ABC Comprador(a)	ABC cpf	123 Total	ABC Vendedor(a)
2	Jaqueta de Couro	600	10	2020-07-26 18:06:05	Abner Joia	13129182764	6,000	Alessandro Felix