

Objetivos:

- I. Características da linguagem SQL - Structured Query Language;
- II. SQL DDL;
- III. SQL DCL;
- IV. SQL DTL;
- V. SQL DML;
- VI. Comando SQL para inserir registros na tabela;
- VII. Comando SQL para listar registros da tabela;
- VIII. Operador de comparação para texto.

I. Características da linguagem SQL - Structured Query Language

O SQL é uma linguagem declarativa em oposição as linguagens procedurais (utilizadas nas linguagens de programação). Ela possui uma notação mais próxima da forma de escrita humana do que as linguagens de programação.

Ela deixa claro **o que deve ser feito** e **não como deve ser feito**. A instrução SQL a seguir diz **que deve ser criada a tabela tbdepartamento**, cabe ao SGBD saber **como será criada a tabela** no sistema de armazenamento do computador:

```
CREATE TABLE if not exists tbdepartamento (  
    iddepartamento INTEGER PRIMARY KEY,  
    nome VARCHAR(20) NOT NULL  
);
```

O SQL é a linguagem de consulta utilizada pelos SGBDs relacionais e objeto-relacionais. Os comandos da linguagem podem divididos em:

- Linguagem de Definição de Dados (SQL DDL - Data Definition Language);
- Linguagem de Manipulação de Dados (SQL DML - Data Manipulation Language);
- Linguagem de Controle de Dados (SQL DCL - Data Control Language);
- Linguagem de Transação de Dados (SQL DTL - Data Transaction Language).

II. SQL DDL

Os comandos DDL são usados para criar, apagar e alterar bancos, tabelas e outros elementos do BD.

Instruções:

- **create** usado para criar banco, tabela ou outro elemento do BD, tais como, view e funções armazenadas (procedure e function). Exemplos:

Comando para criar o BD de nome bdteste. Observação ele não aceita o termo *if not exists*:

```
create database bdteste;
```

Comando para criar a tabela de nome tbdepartamento:

```
create table if not exists tbdepartamento (  
    iddepartamento INTEGER PRIMARY KEY,  
    nome VARCHAR(20) NOT NULL  
);
```

- **drop** usado para deletar um banco, tabela ou outro elemento do BD;

Comando para excluir o BD de nome bdteste:

```
drop database if exists bdteste;
```

Comando para excluir a tabela de nome tbdepartamento:

```
drop table if exists tbdepartamento;
```

- **alter** usado para alterar alguma propriedade do banco, tabela ou outro elemento do BD, tais como, alterar o nome do BD e alterar o tipo de dado de uma coluna da tabela;
- **truncate** usada para remover todos os registros da tabela.

Os comandos DDL devem ser usados raramente, pois eles têm o poder de alterar drasticamente a estrutura das tabelas e BD.

III. SQL DCL

Os comandos DCL são usados para conceder e revogar acesso do usuário a bancos, tabelas e outros elementos do BD. No SGBD todo tipo de recurso requer permissão de acesso do usuário e não é possível logar no SGDB sem ter um usuário.

Instruções:

- **grant** usado para **conceder** acesso do usuário a um determinado recurso do SGBD;
- **revoke** usado para **revogar** acesso do usuário de um determinado recurso do SGBD.

Entende-se como recurso: banco, tabela, coluna da tabela, visão (view), funções e procedures.

IV. SQL DTL

Os comandos DTL são usados para controlar a efetivação das operações no SGBD.

Instruções:

- **start transaction** ou **begin** usados para marcar o início de uma transação no SGBD;
- **commit** usado para efetivar todas as mudanças permanentemente desde o último start transaction ou begin;
- **rollback** usado para desfazer todas as mudanças existentes desde o último start transaction ou begin.

Considere o seguinte exemplo:

Deseja-se registrar a venda de um produto para um cliente. Essa operação é formada por duas instruções:

- Dar baixa no estoque;
- Registrar a venda para o cliente.

Considere uma queda de energia entre uma operação e outra, desta forma, a base de dados ficaria inconsistente. Já usando controle de transações as operações não são efetivadas no sistema de armazenamento até encontrar um commit:

- start transaction - a partir daqui as instruções não serão efetivadas até receber um commit;
- Dar baixa no estoque;
- Registrar a venda para o cliente;
- commit - efetiva as instruções desde o start transaction.

O exemplo a seguir faz uso dos comandos.

A tabela tbdepartamento foi criada usando as instruções:

```
drop table if exists tbdepartamento;  
create table if not exists tbdepartamento (  
    iddepartamento INTEGER PRIMARY KEY,  
    nome VARCHAR(20) NOT NULL  
);
```

Use o comando a seguir para inserir um registro na tbdepartamento:

```
insert into tbdepartamento  
values  
(1, 'Estoque');
```

Fica mais fácil perceber ao usar o ambiente de linha de comando do psql, mas você pode testar usando o pgAdmin. A seguir usando o comando rollback para desfazer as operações desde o último begin:

```
SQL Shell (psql)
bdaula=# begin; ← 1 – O bloco começa aqui e termina no commit ou rollback
BEGIN
bdaula=# insert into tbdepartamento
bdaula=# values (2,'RH');
INSERT 0 1
bdaula=# select * from tbdepartamento;
 iddepartamento | nome
-----+-----
                1 | Estoque
                2 | RH
(2 rows)

bdaula=# rollback; ← 4 – O insert foi descartado ao usar rollback
ROLLBACK
bdaula=# select * from tbdepartamento;
 iddepartamento | nome
-----+-----
                1 | Estoque
(1 row)
```

2 – O registro inserido não será efetivado no sistema de armazenamento

3 – O registro está na tbdepartamento. Porém outra conexão no SGBD não poderá ver este registro

5 – O registro de RH não existe

A seguir usando o comando commit para efetivar as operações desde o último begin:

```
SQL Shell (psql)
bdaula=# begin; ← 1 – Início do bloco de transações
BEGIN
bdaula=# insert into tbdepartamento
bdaula=# values (3,'Compras');
INSERT 0 1
bdaula=# select * from tbdepartamento;
 iddepartamento | nome
-----+-----
                1 | Estoque
                3 | Compras
(2 rows)

bdaula=# commit; ← 2 – O insert foi efetivado ao usar commit
COMMIT
bdaula=# select * from tbdepartamento;
 iddepartamento | nome
-----+-----
                1 | Estoque
                3 | Compras
(2 rows)
```

3 – O registro foi efetivado

Para mais detalhes acesse <https://www.postgresql.org/docs/current/sql-start-transaction.html>

V. SQL DML

Os comandos DML são usados para selecionar, inserir, atualizar e apagar registros, ou seja, esses comandos não são capazes de alterar a estrutura do BD ou tabela. São operações que atuam nas linhas/registros da tabela. Os comandos DML constituem a maioria dos comandos utilizados no cotidiano do BD.

Instruções:

- **select** usado para listar os registros de uma ou mais tabelas de acordo com a cláusula;
- **insert** usado para inserir um ou mais registros na tabela;
- **update** usado para alterar os valores de um ou mais campos de um ou mais registros da tabela;
- **delete** usado para remover registros da tabela.

VI. Comando SQL para inserir registros na tabela

Na linguagem SQL a instrução para inserir um registro (linha) na tabela possui a seguinte estrutura:

```
insert into nomeDaTabela (nomeDaColuna1, nomeDaColuna2, nomeDaColuna3)
values
(valorDaColuna1, valorDaColuna2, valorDaColuna3),
(valorDaColuna1, valorDaColuna2, valorDaColuna3);
```

As cláusulas são terminadas pelo **ponto e vírgula**.

À direita do nome da tabela precisamos colocar, entre parênteses, as colunas que receberão os valores. Após o termo **values** colocamos, entre parênteses, cada registro a ser inserido na tabela. A ordem dos valores precisa ser a mesma ordem das colunas.

Considere como exemplo a tabela **tbpessoa**:

```
drop table if exists tbpessoa;
create table if not exists tbpessoa (
    id INTEGER PRIMARY KEY,
    nome VARCHAR(40) NOT NULL,
    mail VARCHAR(30),
    idade integer
);
```

Cláusula para inserir três registros na **tbpessoa**.

```
insert into tbpessoa(id,nome,mail,idade)
values
(1,'Ana Maria', 'ana@teste.com', 22),
(2,'Pedro Souza', NULL, 25),
(3,'Mara Silva', 'mara@teste.com', 27);
```

Observações:

- Os valores do tipo texto precisam estar envolvidos por aspas simples;
- **null** é um valor e não poder estar envolvido por aspas;

- As colunas id e nome não podem receber o valor null, pelo fato delas serem not null. A chave primária é not null por padrão;
- Cada registro é delimitado pelos parênteses laranja;
- Os valores são separados por vírgula – marcadas em azul;
- Os registros são separados por vírgula – marcadas em vermelho.

Não é obrigatório manter o nome das colunas quando a cláusula possui todos os campos do registro, assim como é mostrado no exemplo a seguir:

```
insert into tbpessoa
values
(4, 'Lucas Naves', 'lucas@teste.com', 25),
(5, 'Ivo Gomes', NULL, 29);
```

Os campos que aceitam nulo não são obrigatórios na cláusula. Porém, neste caso é obrigatório informar o nome das colunas:

```
insert into tbpessoa(id,nome)
values
(6, 'Maria Ribeiro'),
(7, 'Otávio Neves');
```

Os valores podem ser apresentados em qualquer ordem no registro, mas os nomes das colunas precisam ser especificados:

```
insert into tbpessoa(mail,id,idade,nome)
values
('carla@teste.com', 8, 24, 'Carla Silva'),
('joao@teste.com', 9, 23, 'João Santos');
```

VII. Comando SQL para listar registros da tabela

Na linguagem SQL a instrução para listar os registros (linhas) da tabela possui a seguinte estrutura:

```
select nomeDaColuna1, nomeDaColuna2, nomeDaColuna3
from nomeDaTabela
where expressão que resulta em verdadeiro/falso
order by nomeDaColuna asc/desc;
```

As cláusulas são terminadas pelo ponto e vírgula.

Apesar da cláusula select ser apenas um comando. Ela possui os termos select, from, where e order by, e cada termo tem um papel único na cláusula:

- À direita do termo **select** colocamos os nomes das colunas que queremos ver no resultado. O asterisco significa todas as colunas. Como exemplo:

Listar somente as colunas **nome** e **mail** de todos os registros da **tbpessoa**. Os nomes das colunas são separados por vírgulas

Listar todas as colunas e todos os registros da **tbpessoa**

Query Editor

Query History

1 select nome,mail

2 from tbpessoa;

Data Output

Explain

Messages

Notifications

	nome character varying (40)	mail character varying (30)
1	Ana Maria	ana@teste.com
2	Pedro Souza	[null]
3	Mara Silva	mara@teste.com
4	Lucas Naves	lucas@teste.com
5	Ivo Gomes	[null]
6	Maria Ribeiro	[null]
7	Otávio Neves	[null]
8	Carla Silva	carla@teste.com
9	João Santos	joao@teste.com

Query Editor

Query History

1

select *

← Todas as colunas da tbpessoa

2

from tbpessoa;

Data Output

Explain

Messages

Notifications

	id [PK] integer	nome character varying (40)	mail character varying (30)	idade integer
1	1	Ana Maria	ana@teste.com	22
2	2	Pedro Souza	[null]	25
3	3	Mara Silva	mara@teste.com	27
4	4	Lucas Naves	lucas@teste.com	25
5	5	Ivo Gomes	[null]	29
6	6	Maria Ribeiro	[null]	[null]
7	7	Otávio Neves	[null]	[null]
8	8	Carla Silva	carla@teste.com	24
9	9	João Santos	joao@teste.com	23

- À direita do termo **from** colocamos os nomes das tabelas envolvidas na consulta. Se tiver mais de uma tabela, o nome delas deverá estar separado por vírgula;
- À direita do termo **where** deve estar uma expressão lógica que resulta em true/false. A expressão lógica é formada por:
 - ✓ Operadores de comparação: **>** (maior), **>=** (maior ou igual), **<** (menor), **<=** (menor ou igual), **=** (igual) e **!=** (diferente). O operador de comparação compara os dois lados e resulta em true/false:
 - **4 < 3**: o resultado é false;
 - **3 <= 3**: o resultado é true;
 - **'Ana' != 'Ana'**: o resultado é false.
 - ✓ O valor null não pode ser comparado com os operadores **=** e **!=**. Ele requer os operadores **is null** (é nulo) e **is not null** (não é nulo):
 - **null is null**: o resultado é true;
 - **null is not null**: o resultado é false.
 - ✓ Operadores lógicos: **and** (e) e **or** (ou) são usados para combinar operadores de comparação:
 - **and**: o resultado é true se os dois lados são true:
 - **true and true** o resultado é true;
 - **2 < 3 and 3 < 5** o resultado é true pelo fato de 2 < 3 ser true e 3 < 5 ser true;

- $2 < 3$ **and** $5 < 3$ o resultado é false pelo fato de $5 < 3$ ser false.
- or: o resultado é false se os dois lados são false:
 - false **or** false o resultado é false;
 - $2 < 3$ **or** $5 < 3$ o resultado é true pelo fato de $2 < 3$ ser true. Na comparação or basta um dos lados ser true.

Listar as colunas nome e mail dos registros que possuem e-mail, ou seja, os registros cujo mail não seja nulo

```
select nome,mail
from tbpessoa
where mail is not null;
```

	nome	mail
1	Ana Maria	ana@teste.com
2	Mara Silva	mara@teste.com
3	Lucas Naves	lucas@teste.com
4	Carla Silva	carla@teste.com
5	João Santos	joao@teste.com

Listar as colunas nome e mail dos registros que não possuem e-mail, ou seja, os registros cujo mail seja nulo

```
select nome,mail
from tbpessoa
where mail is null;
```

	nome	mail
1	Pedro Souza	[null]
2	Ivo Gomes	[null]
3	Maria Ribeiro	[null]
4	Otávio Neves	[null]

Listar todas as colunas dos registros que possuem idade menor que 25

```
select *
from tbpessoa
where idade < 25;
```

id	nome	mail	idade
1	Ana Maria	ana@teste.com	22
8	Carla Silva	carla@teste.com	24
9	João Santos	joao@teste.com	23

Listar todas as colunas dos registros que possuem idade maior que 24 e menor que 28

```
select *
from tbpessoa
where idade > 24 and idade < 28;
```

id	nome	mail	idade
2	Pedro Souza	[null]	25
3	Mara Silva	mara@teste.com	27
4	Lucas Naves	lucas@teste.com	25

Um erro comum é achar que o intervalo pode ser testado usando $24 < idade < 28$.
Intervalos requerem o uso do operador **and** para unir os testes lógicos

- À direita do termo **order by** deve estar os nomes das colunas usadas para ordenar o resultado. Em outras palavras, o termo order by muda a ordem que os registros são apresentados. Existem as ordens **asc** (crescente) e **desc** (decrescente). Como exemplo:

Listar as colunas nome e idade de todos os registros e ordenar o resultado em ordem crescente de nome. O termo where é opcional

Listar as colunas nome e idade de todos os registros e ordenar o resultado em ordem crescente de idade e decrescente de nome. O nome só será considerado

	quando ocorrer empate de idade																																								
<pre>select nome, idade from tbpessoa order by nome asc;</pre> <table> <thead> <tr> <th>nome</th><th>idade</th></tr> </thead> <tbody> <tr><td>Ana Maria</td><td>22</td></tr> <tr><td>Carla Silva</td><td>24</td></tr> <tr><td>Ivo Gomes</td><td>29</td></tr> <tr><td>João Santos</td><td>23</td></tr> <tr><td>Lucas Naves</td><td>25</td></tr> <tr><td>Mara Silva</td><td>27</td></tr> <tr><td>Maria Ribeiro</td><td>[null]</td></tr> <tr><td>Otávio Neves</td><td>[null]</td></tr> <tr><td>Pedro Souza</td><td>25</td></tr> </tbody> </table>	nome	idade	Ana Maria	22	Carla Silva	24	Ivo Gomes	29	João Santos	23	Lucas Naves	25	Mara Silva	27	Maria Ribeiro	[null]	Otávio Neves	[null]	Pedro Souza	25	<pre>select nome, idade from tbpessoa order by idade asc, nome desc;</pre> <table> <thead> <tr> <th>nome</th><th>idade</th></tr> </thead> <tbody> <tr><td>Ana Maria</td><td>22</td></tr> <tr><td>João Santos</td><td>23</td></tr> <tr><td>Carla Silva</td><td>24</td></tr> <tr><td>Pedro Souza</td><td>25</td></tr> <tr><td>Lucas Naves</td><td>25</td></tr> <tr><td>Mara Silva</td><td>27</td></tr> <tr><td>Ivo Gomes</td><td>29</td></tr> <tr><td>Otávio Neves</td><td>[null]</td></tr> <tr><td>Maria Ribeiro</td><td>[null]</td></tr> </tbody> </table> <p>Idades iguais</p> <p>Idades iguais</p>	nome	idade	Ana Maria	22	João Santos	23	Carla Silva	24	Pedro Souza	25	Lucas Naves	25	Mara Silva	27	Ivo Gomes	29	Otávio Neves	[null]	Maria Ribeiro	[null]
nome	idade																																								
Ana Maria	22																																								
Carla Silva	24																																								
Ivo Gomes	29																																								
João Santos	23																																								
Lucas Naves	25																																								
Mara Silva	27																																								
Maria Ribeiro	[null]																																								
Otávio Neves	[null]																																								
Pedro Souza	25																																								
nome	idade																																								
Ana Maria	22																																								
João Santos	23																																								
Carla Silva	24																																								
Pedro Souza	25																																								
Lucas Naves	25																																								
Mara Silva	27																																								
Ivo Gomes	29																																								
Otávio Neves	[null]																																								
Maria Ribeiro	[null]																																								

O comando **select** é a única forma de ver os registros de uma tabela.

VIII. Operador de comparação para texto

Os operadores **like** e **not like** são usados para fazer comparações com textos. Eles podem ser combinados com os caracteres especiais **%** e **_**.

O caractere especial **%** significa qualquer conjunto de caracteres e o caractere especial **_** significa exatamente um caractere. Por exemplo:

- `select 'Ana Maria' like '%i%'` resulta em true pelo fato de existir a letra **i** no texto Ana Maria;
- `select 'Ana Maria' like '%a'` resulta em true pelo fato de existir a letra **a** e a direita da letra **a** não podem existir outros caracteres;
- `select 'Ana Maria' like 'A%'` resulta em true pelo fato de existir a letra **A** e a esquerda da letra **A** não podem existir outros caracteres;
- `select 'Ana Maria' like 'a%'` resulta em false pelo fato de o operador like ser sensível ao case da letra. Case é maiúsculo;
- `select 'Ana Maria' like '__a%'` resulta em true pelo fato de existir a letra **a** na 3ª posição, pois a esquerda da letra **a** podem ter apenas 2 caracteres.

Os operadores **ilike** e **not ilike** são semelhantes aos operadores **like** e **not like**, com a diferença que eles não são sensíveis ao case da letra:

- `select 'Ana Maria' like 'a%'` resulta em true pelo fato de existir a letra **A** e a esquerda da letra **A** não podem existir outros caracteres.

Exercícios

Para fazer os exercícios use as cláusulas do arquivo Aula2 - Clausula SQL.txt para criar a tbfaculdade e colocar os 50 registros.

Exercício 1: Fazer uma consulta para listar os registros da tbfaculdade.

O resultado terá 50 registros.

Dica: use os termos select e from.

id	unidade	municipio	logradouro	numero	bairro	cep
1	Fatec Adamantina	Adamantina	Rua Paraná	400	Jardim Brasil	17800000
2	Fatec Americana - Ministro Ralph Biasi	Americana	R. Emílio de Menezes	(null)	Gleba B - Vila Amorim	13469111
3	Fatec Araçatuba - Prof. Fernando Amaral de Almeida Prado	Araçatuba	Av. Prestes Maia	1764	Jardim Ipanema	16052045
4	Fatec Araraquara	Araraquara	R. Precide Scarpino	126	Jardim Santa Clara	14811373
5	Fatec Araras	Araras	R. Jarbas Leme de Godoy	875	Jardim José Ometto II	13606389

Exercício 2: Fazer uma consulta para listar os registros da tbfaculdade em ordem decrescente de município.

O resultado terá 50 registros.

Dica: use os termos select, from e order by.

id	unidade	municipio	logradouro	numero	bairro	cep
50	Fatec Presidente Prudente	Presidente Prudente	R. Teresina	75	Vila Paulo Roberto	19046230
49	Fatec Praia Grande	Praia Grande	Praça 19 de Janeiro	144	Boqueirão	11700100
48	Fatec Pompéia - Shunji Nishimura	Pompéia	Av. Shunji Nishimura	605	Distrito Industrial	17580000
47	Fatec Piracicaba - Dep. Roque Trevisan	Piracicaba	R. Diácono Jair de Oliveira	651	Santa Rosa	13414155
46	Fatec Pindamonhangaba	Pindamonhangaba	Rod. Vereador Abel Fabricio Dias (SP 62)	4010	Água Preta	12414000

Exercício 3: Fazer uma consulta para listar o registro da tbfaculdade que está no município de Jacareí.

O resultado terá 1 registro.

Dica: use os termos select, from e where.

id	unidade	municipio	logradouro	numero	bairro	cep
33	Fatec Jacareí - Prof. Francisco de Moura	Jacareí	Av. Faria Lima	155	Jardim Santa Maria	12328070

Exercício 4: Fazer uma consulta para listar os registros da tbfaculdade que não possuem número.

O resultado terá 4 registros.

Dica: use os termos select, from e where, e o operador is null.

id	unidade	municipio	logradouro	numero	bairro	cep
2	Fatec Americana - Ministro Ralph Biasi	Americana	R. Emílio de Menezes	(null)	Gleba B - Vila Amorim	13469111
11	Fatec Botucatu	Botucatu	Av. José Ítalo Bacchi	(null)	Jardim Aeroporto	18606855
35	Fatec Jahu	Jaú	R. Frei Galvão	(null)	Jardim Pedro Ometto	17212599
41	Fatec Mococa	Mococa	Av. Dr. Américo Pereira Lima	(null)	Jardim Lavínia	13736260

Exercício 5: Fazer uma consulta para listar os registros da tbfaculdade que possuem o nome do município começando pela letra **f**.

O resultado terá 3 registros.

Dica: use os termos select, from e where, e o operador like ou ilike.

id	unidade	municipio	logradouro	numero	bairro	cep
20	Fatec Ferraz de Vasconcelos	Ferraz de Vasconcelos	R. Carlos de Carvalho	200	Jardim São João	08545120
21	Fatec Franca - Dr. Thomaz Novelino	Franca	R. Irênio Grecco	4580	Vila Imperador	14405191
22	Fatec Franco da Rocha	Franco da Rocha	Rod. Prefeito Luiz Salomão Chamma	240	Pouso Alegre	07857050

Exercício 6: Fazer uma consulta para listar os registros da tbfaculdade que estão em um bairro que possui o termo **vila**.

O resultado terá 11 registros.

Dica: use os termos select, from e where, e o operador like ou ilike.

id	unidade	municipio	logradouro	numero	bairro	cep
2	Fatec Americana - Ministro Ralph Biasi	Americana	R. Emílio de Menezes	(null)	Gleba B - Vila Amorim	13469111
15	Fatec Carapicuíba	Carapicuíba	Av. Francisco Pignatari	650	Vila Gustavo Correia	06310390
18	Fatec Cruzeiro - Prof. Waldomiro May	Cruzeiro	Av. Rotary	383	Vila Paulista	12701170
21	Fatec Franca - Dr. Thomaz Novelino	Franca	R. Irênio Grecco	4580	Vila Imperador	14405191
27	Fatec Itapetininga - Prof. Antonio Belizandro Barbosa Rezende	Itapetininga	R. Dr. João Vieira de Camargo	104	Vila Barth	18205600
29	Fatec Itaquaquecetuba	Itaquaquecetuba	Av. Itaquaquecetuba	711	Vila Monte Belo	08577210
32	Fatec Jaboticabal - Nilo de Stéfani	Jaboticabal	Av. Eduardo Zambianchi	31	Vila Industrial	14883130
40	Fatec Mauá	Mauá	Av. Antonia Rosa Fioravante	804	Vila Fausto Morelli	09390120
42	Fatec Mogi das Cruzes	Mogi das Cruzes	R. Carlos Barattino	908	Vila Nova Mogilar	08773600
44	Fatec Osasco - Pref. Hirant Sanazar	Osasco	R. Pedro Rissato	30	Vila dos Remédios	06296220
50	Fatec Presidente Prudente	Presidente Prudente	R. Teresina	75	Vila Paulo Roberto	19046230

Exercício 7: Fazer uma consulta para listar os registros da tbfaculdade que possuem o CEP com os três últimos dígitos sendo zero.

O resultado terá 6 registros.

Dica: use os termos select, from e where, e o operador like.

id	unidade	municipio	logradouro	numero	bairro	cep
1	Fatec Adamantina	Adamantina	Rua Paraná	400	Jardim Brasil	17800000
23	Fatec Garça - Dep. Julio Julinho Marcondes de Moura	Garça	Av. Presidente Vargas	2331	José Ribeiro	17400000
37	Fatec Lins - Prof. Antonio Seabra	Lins	Estrada Mario Covas Jr.	Km 1	Estrada Vicinal Lins - Guaimbe - Zona Rural	16400000
38	Fatec Marília - Estudante Rafael Almeida Camarinha	Marília	Av. Castro Alves	62	Somenzari	17506000
46	Fatec Pindamonhangaba	Pindamonhangaba	Rod. Vereador Abel Fabricio Dias (SP 62)	4010	Água Preta	12414000
48	Fatec Pompéia - Shunji Nishimura	Pompéia	Av. Shunji Nishimura	605	Distrito Industrial	17580000

Exercício 8: Fazer uma consulta para listar os registros da tbfaculdade que possuem o dígito 5 na 2ª posição do CEP.

O resultado terá 3 registros.

Dica: use os termos select, from e where, e o operador like.

id	unidade	municipio	logradouro	numero	bairro	cep
16	Fatec Catanduva	Catanduva	R. Maranhão	898	Centro	15800020
34	Fatec Jales - Prof. José Camargo	Jales	R. Vicente Leporace	2630	Jardim Trianon	15703116
39	Fatec Matão	Matão	Av. Habib Gabriel	1360	Residencial Olívio Benassi	15990539

Exercício 9: Fazer uma consulta para listar os registros da tbfaculdade que possuem o dígito 5 na 2ª posição e o dígito 2 na penúltima posição do CEP.

O resultado terá 1 registro.

Dica: use os termos select, from e where. Use o operador like.

id	unidade	municipio	logradouro	numero	bairro	cep
16	Fatec Catanduva	Catanduva	R. Maranhão	898	Centro	15800020

Exercício 10: Fazer uma consulta para listar os registros da tbfaculdade que possuem exatamente dois dígitos no número.

O resultado terá 7 registros.

Dica: use os termos select, from e where, e o operador like.

id	unidade	municipio	logradouro	numero	bairro	cep
14	Fatec Capão Bonito	Capão Bonito	R. Amantino de Oliveira Ramos	60	Terras de Embiruçu	18304755
25	Fatec Guarulhos	Guarulhos	R. Cristóbal Cláudio Eliilo	88	Parque Cecap	07190065
26	Fatec Indaiatuba - Dr. Archimedes Lammoglia	Indaiatuba	R. D. Pedro I	65	Cidade Nova I	13334100
32	Fatec Jaboticabal - Nilo de Stéfani	Jaboticabal	Av. Eduardo Zambianchi	31	Vila Industrial	14883130
38	Fatec Marília - Estudante Rafael Almeida Camarinha	Marília	Av. Castro Alves	62	Somenzari	17506000
44	Fatec Osasco - Pref. Hiran Sanazar	Osasco	R. Pedro Rissato	30	Vila dos Remédios	06296220
50	Fatec Presidente Prudente	Presidente Prudente	R. Teresina	75	Vila Paulo Roberto	19046230