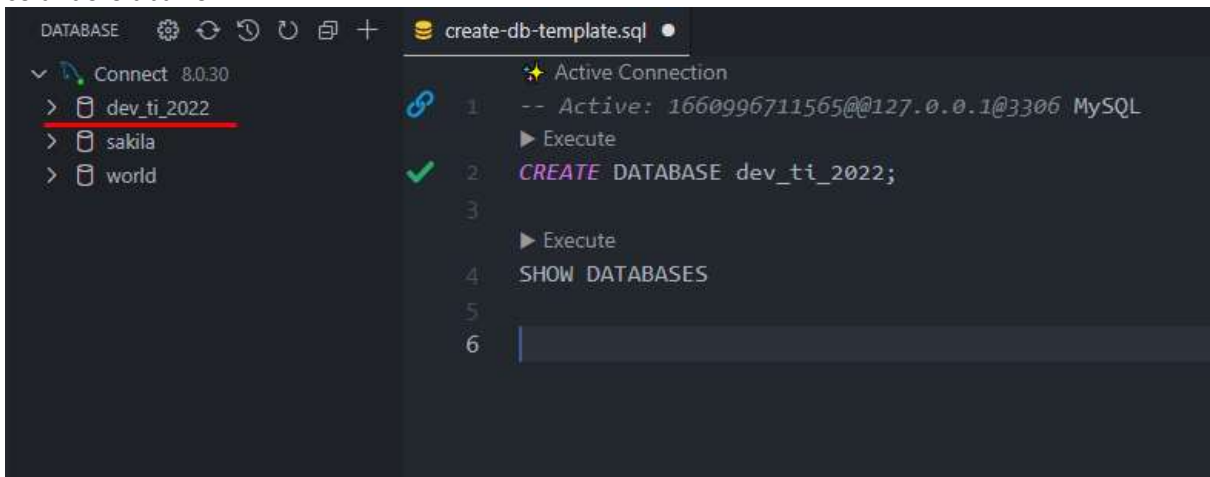
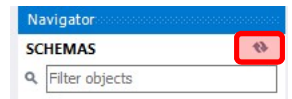


# SQL Básico (MySQL)

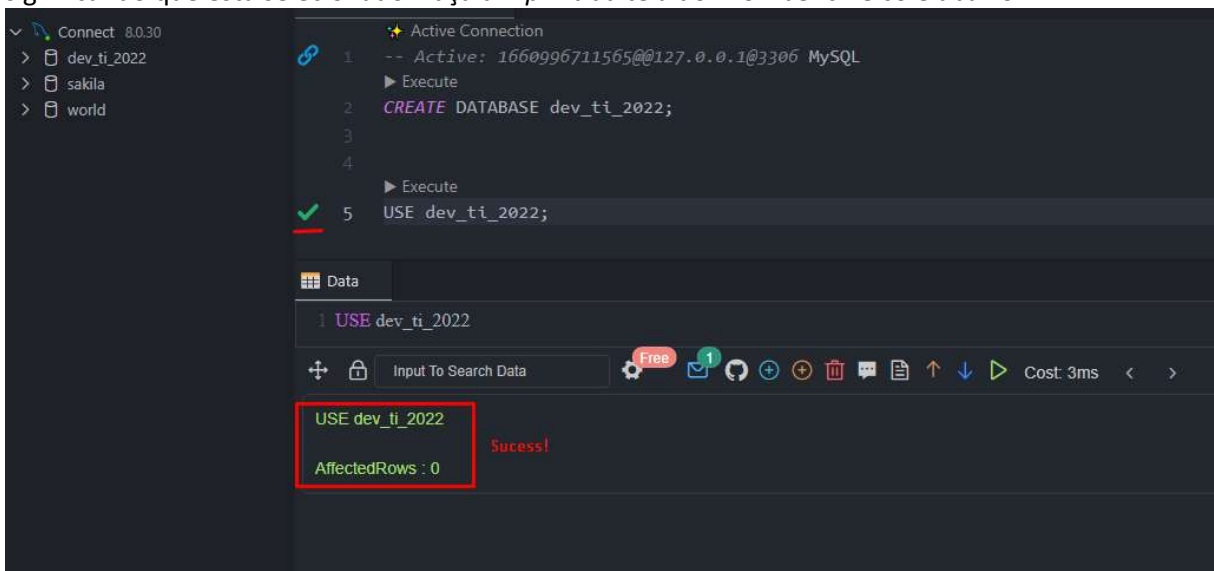
1. Execute o comando `SHOW DATABASES` para verificar os bancos de dados existentes. Quais os esquemas extras que apareceram no resultado e que não são listados na lista dos esquemas do MySQL Workbench?

R.: **information\_schema, mysql, performance\_schema, sys.**

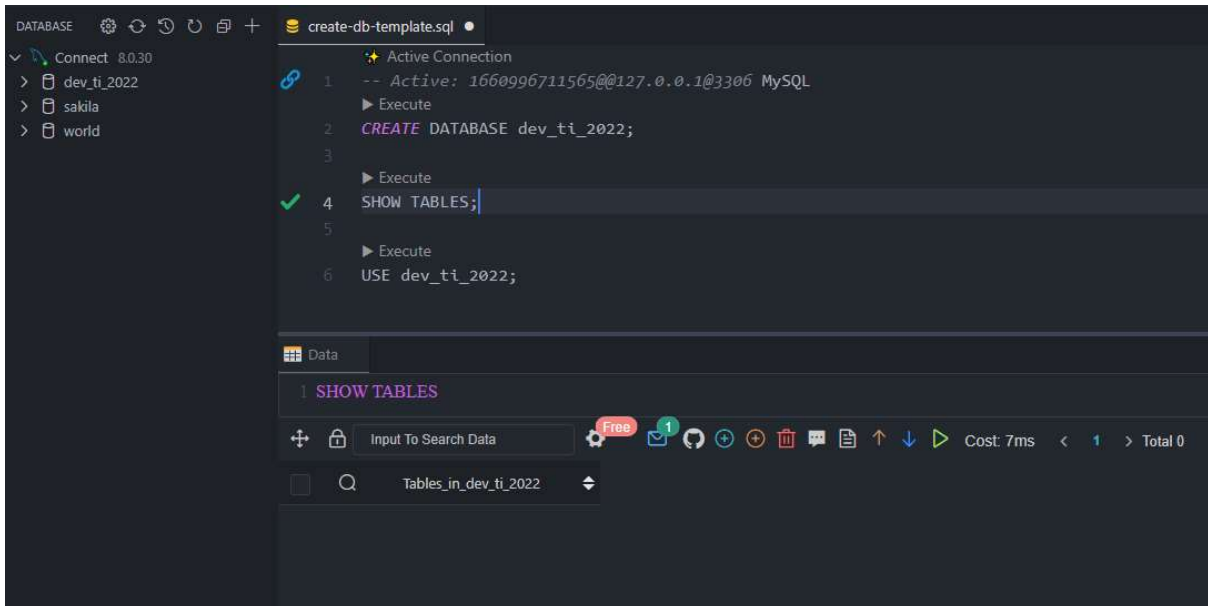
2. Rode o comando `CREATE DATABASE dev_ti_2022` para criar o banco de dados. Atualize (dê um *refresh*) na lista de esquemas do Workbench e faça um *print* da tela colando-o abaixo.



3. Execute o comando `USE` apropriado para selecioná-lo. Confira para ver se o nome do banco está em negrito significando que está selecionado. Faça um *print* da tela do Workbench e cole abaixo.



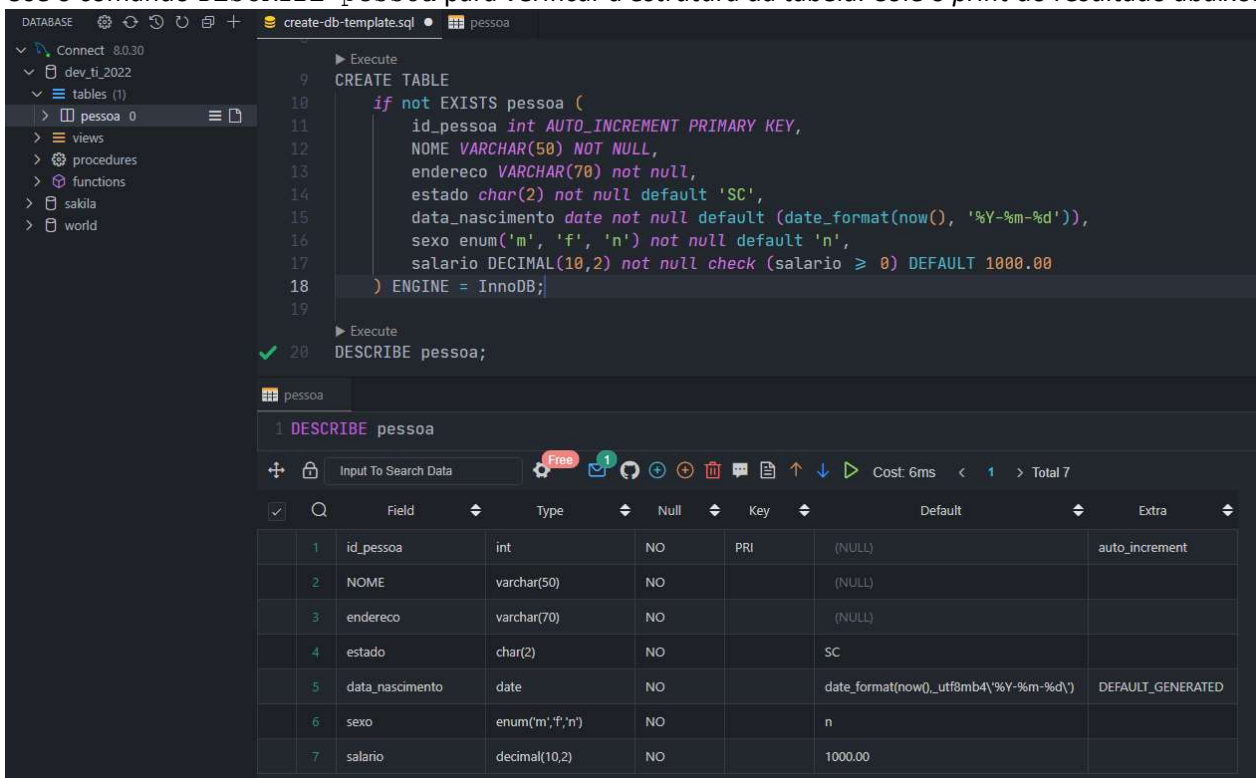
4. Use o comando `SHOW TABLES` para listar as tabelas existentes neste banco. A consulta não deve retornar nenhum resultado.



5. Cria a tabela com o comando SQL abaixo:

```
CREATE TABLE IF NOT EXISTS pessoa (
    id_pessoa INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(50) NOT NULL,
    endereco VARCHAR(70) NOT NULL,
    estado CHAR(2) NOT NULL DEFAULT 'SC',
    data_nascimento DATE NOT NULL DEFAULT (date_format(now(), '%Y-%m-%d')),
    sexo ENUM('m', 'f', 'n') NOT NULL DEFAULT 'n',
    salario DECIMAL(10, 2) NOT NULL CHECK (salario >= 0) DEFAULT 1000.00
) ENGINE = InnoDB;
```

6. Use o comando `DESCRIBE` `pessoa` para verificar a estrutura da tabela. Cole o *print* do resultado abaixo.



7. Insira na tabela os seguintes registros.

id_pessoa	nome	endereço	estado	data_nascimento	sexo	salário
1	André	Av.X, 10	RJ	1980-01-01	m	1.000,00
2	Maria	Av.X, 50	RJ	1970-10-31	f	3.000,00
3	Luis	Rua Y, 25	PR	1990-05-15	m	1.000,00
4	Ana	Rua Y, 50	PR	2000-02-28	f	4.300,00
5	Augusto	Av.Z, 80	SC	1995-08-17	m	8.600,00
6	Ricardo	Av.X, 50	PR	2005-07-16	m	1.500,00
7	Thais	Rua Z, 20	SP	1960-06-06	f	10.000,00
8	Pedro	Rua Z, 100	SP	1980-09-05	m	8.900,00
9	Flavia	Av.Z, 40	SC	1980-01-01	f	2.300,00
10	Patrícia	Alameda W, 42	SP	1985-11-23	f	15.000,00

O primeiro registro pode ser inserido com o comando a seguir. Use-o como base para inserir todos os restantes.

```
INSERT INTO pessoa VALUES (null, 'André', 'Av.X, 10', 'RJ', '1980-01-01', 'm', 1000.00);
```

8. Modifique a data de nascimento de Ana para 7 de setembro de 1998 (use o campo `id_pessoa` para fazer isso).

9. Modifique o salário de Pedro para 9 mil reais e o de Augusto para 8500 (use dois comandos diferentes).

10. Experimente agora inserir o registro abaixo:

```
INSERT INTO pessoa VALUES (null, 'Fulano', 'Rua F', '2000-01-01', 'p', 1000.00);
```

Foi possível inserir o registro? Se não, qual foi o motivo?

**R.: o valor 'p' não é um valor aceito pela coluna sexo, os valores aceitos são: 'f', 'm', 'n'**

11. Tente inserir o registro abaixo:

```
INSERT INTO pessoa VALUES (null, 'Fulano', 'Rua F', '2000-01-01', 'm', -1000.00);
```

Foi possível inserir o registro? Se não, qual foi o motivo?

**R.: a coluna 'salário' não aceita valores negativos**

12. Faça agora as seguintes consultas e escreva os comandos utilizados:

a. Liste todos os registros e todas as colunas.

```
SELECT * from pessoa
```

b. Liste todos os registros mas somente as colunas `nome` e `salário`.

```
SELECT nome, salario from pessoa
```

c. Mostre somente os nomes em ordem crescente.

```
SELECT nome from pessoa ORDER BY nome asc;
```

d. Mostre os nomes com respectivas datas de nascimento e salários em ordem decrescente de nascimento e, para nascimentos iguais, ordem crescente de salário.

```
SELECT nome, data_nascimento, salario from pessoa ORDER BY data_nascimento asc, salario asc;
```

e. Exiba o campo `nome` somente dos registros cujo `sexo = 'f'`.

```
SELECT nome from pessoa where sexo = 'f';
```

- f. Mostre a média salarial, agrupada por sexo e dê apelido de 'média salarial' para a coluna.

```
SELECT
    sexo,
    avg(salario) as 'media salarial'
FROM pessoa
group BY sexo;
```

- g. Melhore a consulta anterior usando a função `round()` para arredondar o resultado para 2 dígitos.

```
SELECT
    sexo,
    round(avg(salario),2) as 'media salarial'
FROM pessoa
group BY sexo;
```

- h. Melhore ainda mais a consulta mostrando o símbolo 'R\$' (utilize a função `concat()`). Em vez da função `round()` use a função `format()` para formatar a média salarial no formato monetário.

```
SELECT
    sexo,
    concat('R$ ',format(avg(salario),2, 'de_DE'),20,' ') as 'media
salarial'
FROM pessoa
group BY sexo;
```

- i. Agora, após aplicar a formatação monetária mas antes de concatenar com o 'R\$' use a função `lpad()` para produzir uma saída de 20 caracteres preenchidas com espaços à esquerda, como mostrado abaixo.

	sexo	média salarial
►	m	R\$ 4.200,00
	f	R\$ 6.920,00

```
SELECT
    sexo,
    concat('R$ ', lpad (format(avg(salario),2, 'de_DE'),20,' ')) as
'media salarial'
FROM pessoa
group BY sexo;
```

- j. Finalmente, tente unir as funções `lpad()` e `rpadd()` para centralizar o caractere do campo sexo, preenchendo com 3 asteriscos à direita e 3 à esquerda. Dê o apelido de 'sexo' para esta coluna.

	sexo	média salarial
►	***m***	R\$ 4.200,00
	***f***	R\$ 6.920,00

```
SELECT
    rpadd (lpad (sexo,4,'*'),7,'*') as sexo,
    concat('R$ ', lpad (format(avg(salario),2, 'de_DE'),20,' ')) as
'media salarial'
FROM pessoa
group BY sexo;
```

- k. Utilize a cláusula `HAVING` para mostrar somente o sexo cuja média salarial é maior ou igual a 5 mil reais.

```
SELECT
    rpad (lpad (sexo,4,'*'),7,'*') as sexo,
    concat('R$ ', lpad (format(avg(salario),2, 'de_DE'),20,' ')) as
'media_salarial'
FROM pessoa
group BY sexo
HAVING avg(salario) >= 5000.00;
```

- l. Mostre o maior e o menor dos salários na mesma consulta e modifique o nome das colunas produzindo o resultado abaixo.

	maior salário	menor salário
▶	15000.00	1000.00

```
SELECT max(salario) as 'maior salario', min(salario) as 'menor salario'
from pessoa;
```

- m. Liste a média dos salários por estado ordenando de forma decrescente por média salarial.

```
SELECT
    estado, concat('R$ ', lpad (format(avg(salario),2, 'de_DE'),20,'
')) as 'media_salarial'
FROM pessoa
group by estado
ORDER BY avg(salario) desc;
```

- n. Liste os estados em ordem decrescente de média salarial e cuja média dos salários seja maior ou igual a R\$ 5.000,00.

```
SELECT
    estado, concat('R$ ', lpad (format(avg(salario),2, 'de_DE'),20,'
')) as 'media_salarial'
FROM pessoa
group by estado
HAVING avg(salario) >= 5000.00
ORDER BY avg(salario) desc;
```

- o. Mostre o nome e o estado das pessoas que moram na região Sul.

```
select nome, estado from pessoa where estado in ('SC', 'PR', 'RS')
```

- p. Utilize a função `year()` para mostrar o nome, data de nascimento e estado de todas as pessoas que nasceram em 1980 ou antes ( $\leq$ ).

```
select
    nome,
    data_nascimento,
    estado from pessoa
where
    year(data_nascimento) <= 1980;
```

- q. Faça uma consulta que retorne o nome, data de nascimento e estado de todas as pessoas que nasceram em 1980 ou antes ( $\leq$ ) e que sejam de 'SP'.

	nome	estado	data_nascimento
►	Thais	SP	1960-06-06
	Pedro	SP	1980-09-05

```
select nome, data_nascimento, estado from pessoa where
year(data_nascimento) <= 1980 and estado = 'SP';
```

- r. Insira a pessoa com o comando a seguir.

```
INSERT INTO pessoa (nome, endereco)
VALUES ('Fulano', 'Rua dos Fulanos');
```

Execute uma consulta que busque somente pelo nome 'Fulano' e mostre todos seus campos. Qual é o conteúdo dos campos data de nascimento, estado, sexo e salário? Por que o MySQL atribuiu esses valores aos campos?

**R.: O SQL adicionou esses dados pois especificamos na criação da tabela os valores default**

- s. Exclua o registro criado na questão anterior utilizando para isso o campo de identificação da pessoa.

```
delete from pessoa where id_pessoa = 11;
```

- t. Mostre todas as pessoas que morem em ruas.

```
select nome from pessoa where endereco like 'Rua%';
```

- u. Mostre todas as pessoas que não morem em alamedas.

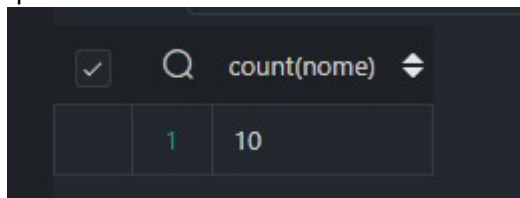
```
select nome from pessoa where endereco not like 'Alameda%';
```

- v. Utilize o operador `BETWEEN` para mostrar todas as pessoas nascidas entre 1º de novembro de 1985 e 15 de abril de 2000. Use a função `date_format()` com a máscara '%d de %M de %Y' para formatar a saída. Dê o apelido de 'nascimento' a esse campo.

```
select nome, date_format(data_nascimento, '%d de %M de %Y') as
nascimento
from pessoa
WHERE data_nascimento BETWEEN '1985-11-01' and '2000-04-15';
```

- w. O resultado da consulta anterior retornou as datas em inglês. Modifique a 'localização' do sistema para português do Brasil com o comando `SET lc_time_names='pt_BR'` e refaça a consulta.

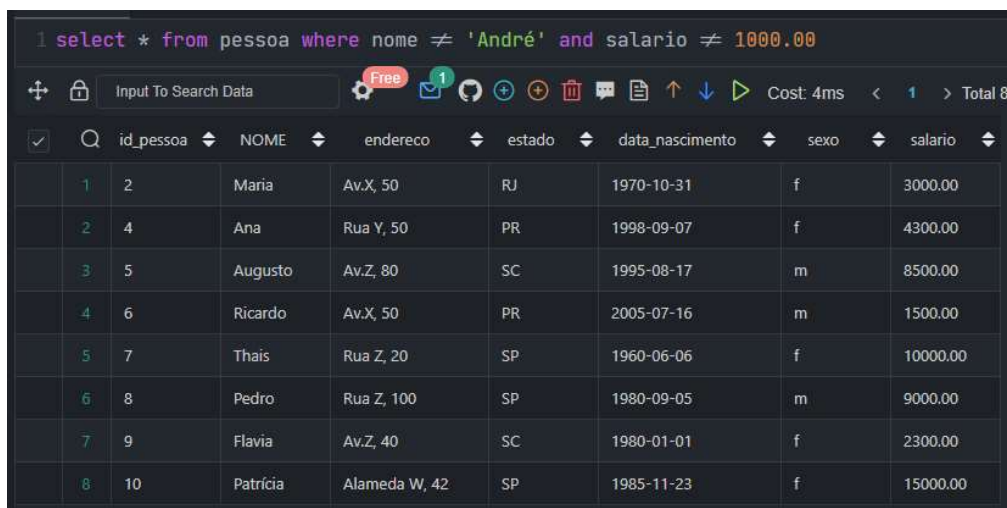
- x. Utilize a função `count()` para retornar o número de registros que atenderam ao critério da consulta da questão anterior.



A screenshot of a database query result. At the top, there is a column labeled 'count(nome)'. Below it is a table with one row showing the count '10'.

count(nome)
10

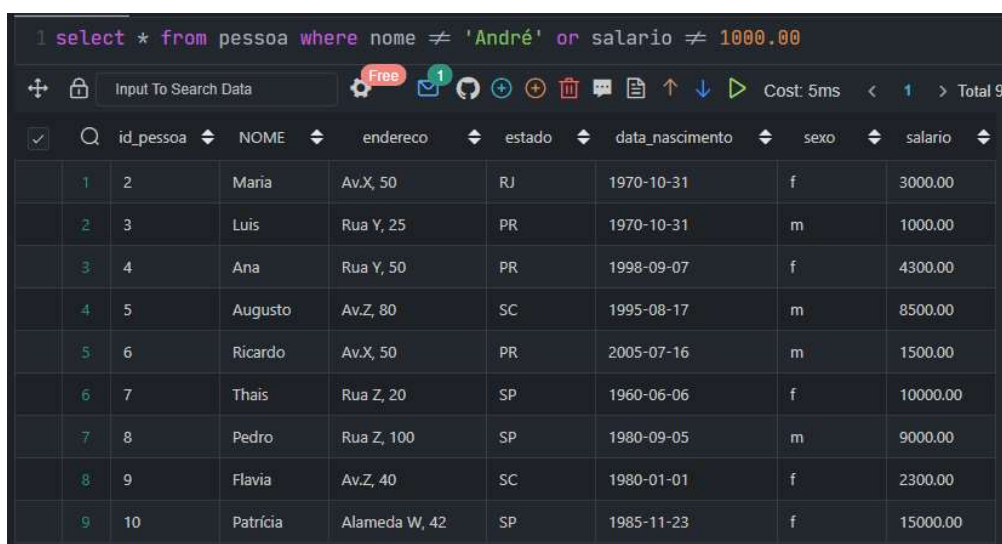
- y. Faça uma consulta que liste todos os registros mas **não** deve mostrar **nem** as pessoas que se chamem 'André' e **nem** as pessoas que ganhem R\$ 1.000,00. Quantos registros atenderam a esse critério?



A screenshot of a database query result. The query is `select * from pessoa where nome != 'André' and salario != 1000.00`. The result shows a table with 8 rows of data. The columns are id\_pessoa, NOME, endereco, estado, data\_nascimento, sexo, and salario.

	id_pessoa	NOME	endereco	estado	data_nascimento	sexo	salario
1	2	Maria	Av.X, 50	RJ	1970-10-31	f	3000.00
2	4	Ana	Rua Y, 50	PR	1998-09-07	f	4300.00
3	5	Augusto	Av.Z, 80	SC	1995-08-17	m	8500.00
4	6	Ricardo	Av.X, 50	PR	2005-07-16	m	1500.00
5	7	Thais	Rua Z, 20	SP	1960-06-06	f	10000.00
6	8	Pedro	Rua Z, 100	SP	1980-09-05	m	9000.00
7	9	Flavia	Av.Z, 40	SC	1980-01-01	f	2300.00
8	10	Patrícia	Alameda W, 42	SP	1985-11-23	f	15000.00

- z. Faça uma consulta que liste todos os registros **exceto** se for uma pessoa que ganhe R\$ 1.000,00 e ao mesmo tempo se chame 'André'. Quantos registros atenderam a esse critério?

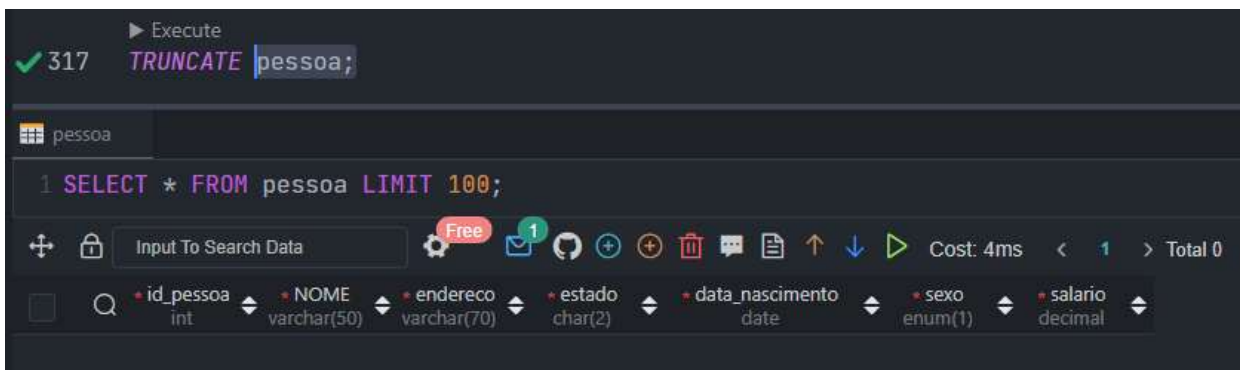


A screenshot of a database query result. The query is `select * from pessoa where nome != 'André' or salario != 1000.00`. The result shows a table with 9 rows of data. The columns are id\_pessoa, NOME, endereco, estado, data\_nascimento, sexo, and salario.

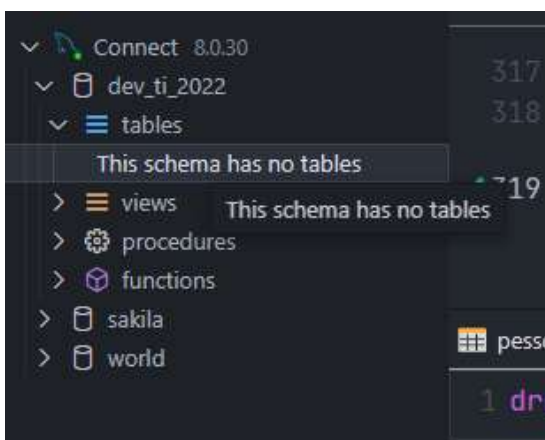
	id_pessoa	NOME	endereco	estado	data_nascimento	sexo	salario
1	2	Maria	Av.X, 50	RJ	1970-10-31	f	3000.00
2	3	Luis	Rua Y, 25	PR	1970-10-31	m	1000.00
3	4	Ana	Rua Y, 50	PR	1998-09-07	f	4300.00
4	5	Augusto	Av.Z, 80	SC	1995-08-17	m	8500.00
5	6	Ricardo	Av.X, 50	PR	2005-07-16	m	1500.00
6	7	Thais	Rua Z, 20	SP	1960-06-06	f	10000.00
7	8	Pedro	Rua Z, 100	SP	1980-09-05	m	9000.00
8	9	Flavia	Av.Z, 40	SC	1980-01-01	f	2300.00
9	10	Patrícia	Alameda W, 42	SP	1985-11-23	f	15000.00



13. Execute o comando para truncar (apagar todos os dados) da tabela `pessoa`. Dê um *refresh* e verifique que a tabela continua dentro do banco, mas está vazia.



14. Agora remova realmente a tabela `pessoa`. Dê um *refresh* para se assegurar que a tabela não existe mais.



15. Execute o comando que apaga o banco de dados `dev_ti_2022`

