



**Trabalho de Banco de Dados**  
**Estudo de Caso D: Sistema de Denúncia de Violação de Direitos Humanos.**

**Apresentamos o**  
**SIGVID – Sistema Integrado de Gestão de Violações e Denúncias.**

**Alunos:**

**Matheus Henrique dos Santos  
Gomes**

**Paulo Antonio Blasque Fernandes**

**Professor(a):**

**Ivone Penque Matsuno Yugoshi**



## Sumário

<b>1. INTRODUÇÃO.....</b>	<b>3</b>
<b>2. DESCRIÇÃO DO CASO DE USO.....</b>	<b>3</b>
<b>3. ESQUEMA CONCEITUAL (DER).....</b>	<b>4</b>
<b>4. ESQUEMA LÓGICO (ESQUEMA RELACIONAL).....</b>	<b>5</b>
<b>5. SCRIPTS PARA CRIAÇÃO DO BANCO DE DADOS.....</b>	<b>5</b>
5.1 Tabela “Violacao”.....	5
5.2 Tabela “Prova”.....	6
5.3 Tabela “Pessoa”.....	6
5.4 Tabela “Advogado”.....	7
5.5 Tabela “Contrato”.....	7
5.6 Tabela “Orgao_responsavel”.....	8
5.7 Tabela “Pessoa_denunciada”.....	9
5.8 Tabela “Denuncia”.....	10
5.9 Tabela “Pessoa_denunciante”.....	11
5.10 Tabela “telefonePessoa”.....	11
5.11 Tabela “Denuncia_identificada”.....	12
5.12 Tabela “Denuncia_anonima”.....	13
5.13 Tabela “Advogado_acompanha_denuncia”.....	13
5.14 Tabela “Denuncia_contem_provas”.....	14
5.15 Tabela “Denuncia_contem_violacao”.....	14
<b>6. CONSULTAS EM ÁLGEBRA RELACIONAL E CÓDIGO SQL.....</b>	<b>15</b>
6.1 Listar todas as pessoas denunciante.....	15
6.2 Listar todas as denúncias com relatos, status e órgão responsável.....	15
6.3 Listar denúncias identificadas com nome e idade do denunciante.....	16
6.4 Mostrar denúncias com seus relatos e respectivas violações separadas por uma vírgula.....	16
6.5 Listar denúncias e seus respectivos tipos de provas com descrição, ordenado de maneira crescente pelo número identificador.....	17
6.6 Listar denúncias feitas por pessoa com mais de 60 anos.....	17
6.7 Listar denúncias com mais de uma violação e sua respectiva quantidade de violações.....	18
6.8 Listar denúncias com mais de uma prova e sua respectiva quantidade de provas.....	18
6.9 Listar pessoas denunciadas e o nome do órgão para onde a denúncia foi.....	19
6.10 Listar as denúncias com provas do tipo 'Áudio' associadas a denúncias de assédio e a descrição da prova.....	19
6.11 Contar quantas denúncias cada órgão recebeu.....	20
6.12 Listar o número de identificação de denúncias já encerradas.....	20
6.13 Listar denúncias com mais de uma pessoa envolvida.....	21
6.14 Listar denúncias que estão em análise por perícia técnica.....	21
6.15 Listar todas as violações com suas descrições.....	21
6.16 Listar as provas do tipo 'Documento' associadas a denúncias com mais de uma violação.....	22
6.17 Contar quantas vezes cada tipo de prova aparece.....	22
6.18 Ver denúncias em que a pessoa denunciante também é a pessoa denunciada.....	23
6.19 Listar denúncias acompanhadas por advogados.....	23
<b>7. INFORMAÇÕES DOS ARQUIVOS.....</b>	<b>24</b>



## **1. INTRODUÇÃO**

Em um sistema nacional de monitoramento e denúncia de violações de direitos humanos são registradas informações essenciais para documentar, investigar e acompanhar denúncias. O sistema SIGVID foi projetado para garantir flexibilidade no registro das ocorrências, bem como a rastreabilidade das informações processadas, sendo assim a melhor maneira de satisfazer essa necessidade.

## **2. DESCRIÇÃO DO CASO DE USO**

O sistema chamado SIGVID (Sistema Integrado de Gestão de Violações e Denúncias) tem como objetivo registrar e acompanhar denúncias de irregularidades envolvendo órgãos públicos ou privados. O sistema tem como foco:

- O cadastro de pessoas envolvidas nos casos (denunciantes, denunciados e advogados),
- O controle de denúncias (identificadas ou anônimas),
- O armazenamento de provas e violações relacionadas a cada denúncia,
- O acompanhamento de casos por advogados contratados ou autorizados.

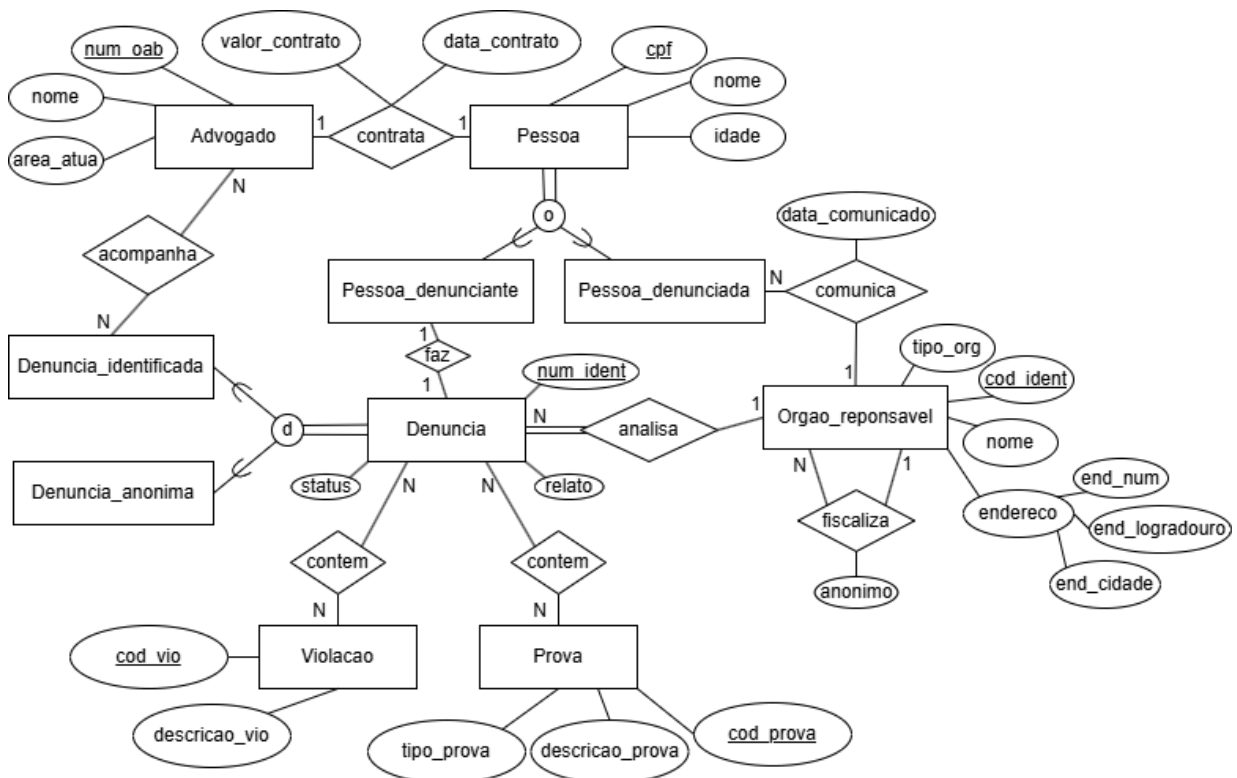
### **Requisitos para funcionamento do sistema:**

- Toda denúncia registrada deve ter um número de identificação único, um status (Aberta, Em análise ou Encerrada), um órgão responsável associado e uma descrição do método de análise (Automatizada ou Perícia técnica).
- A denúncia pode ser identificada, associada a um denunciante com CPF, ou anônima, sem vínculo com uma pessoa.
- Um denunciante pode ter uma relação com o caso (por exemplo: vítima, testemunha) e pode participar de várias denúncias.
- Um advogado pode acompanhar denúncias identificadas se tiver sido autorizado para o caso. Advogados são identificados por seu número de OAB, área de atuação (por padrão "Civil") e nome.



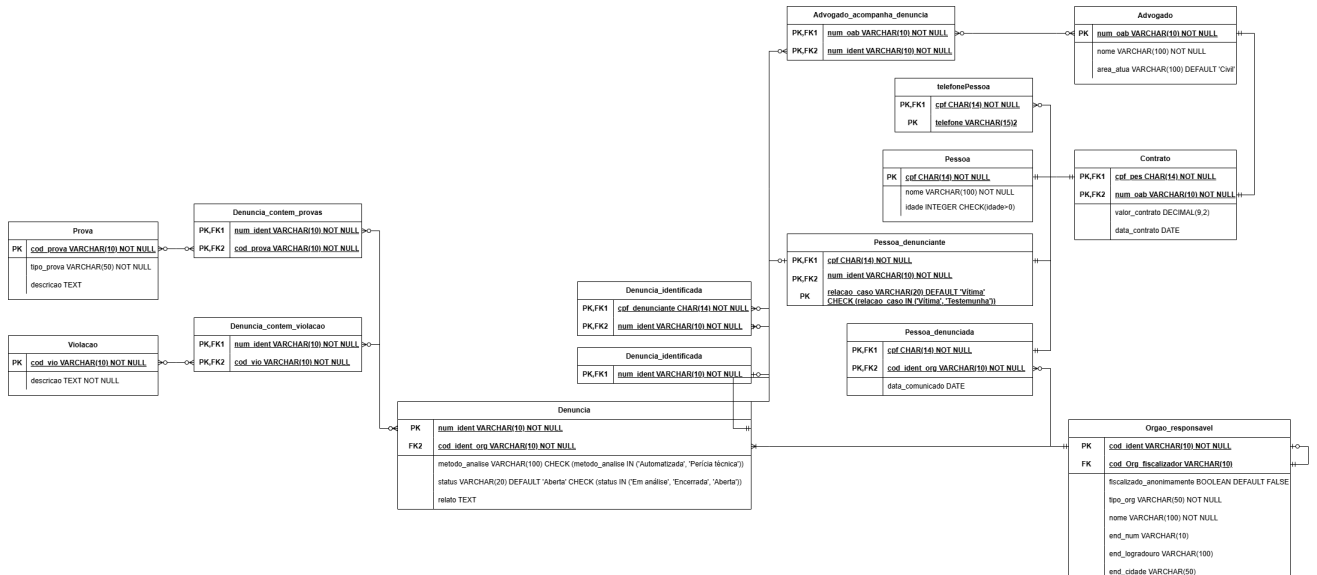
- Pessoas físicas com CPF, nome e idade podem ser denunciadas em um determinado órgão, e deve ser registrada a data em que o comunicado foi feito ao órgão.
- O sistema também registra provas que apoiam cada denúncia, especificando o tipo e a descrição da prova.
- Cada denúncia também pode estar associada a uma ou mais violações legais, registradas com código e descrição.
- Pessoas físicas podem contratar advogados, informando o valor do contrato e a data do acordo. O mesmo advogado pode ter contratos com várias pessoas.
- Os órgãos responsáveis são identificados por código e podem possuir vínculo hierárquico com outro órgão fiscalizador). Também são cadastradas informações como nome, endereço (número, logradouro e cidade) e se a fiscalização é feita anonimamente.
- Pessoas cadastradas podem ter um ou mais telefones vinculados.

### 3. ESQUEMA CONCEITUAL (DER)





#### 4. ESQUEMA LÓGICO (ESQUEMA RELACIONAL)



#### 5. SCRIPTS PARA CRIAÇÃO DO BANCO DE DADOS

##### 5.1 Tabela “Violacao”

Tabela01_Violacao				
Atributo	Tipo de Dados	Restrição Domínio	Restrição de Entidade	Restrição Referencial
cod_vio	VARCHAR(10)	NOT NULL	PK(chave primaria)	
descricao	TEXT	NOT NULL		

SQL

```
CREATE TABLE Violacao(
  cod_vio VARCHAR(10) NOT NULL,
  descricao TEXT NOT NULL,
  PRIMARY KEY (cod_vio)
);
```



### 5.2 Tabela “Prova”:

Tabela02_Provas				
Atributo	Tipo de Dados	Restrição Domínio	Restrição de Entidade	Restrição Referencial
cod_prova	VARCHAR(10)	NOT NULL	PK(chave primaria)	
tipo_prova	VARCHAR(50)	NOT NULL		
descricao	TEXT			

SQL

```
CREATE TABLE Prova(  
    cod_prova VARCHAR(10) NOT NULL,  
    tipo_prova VARCHAR(50) NOT NULL,  
    descricao TEXT,  
    PRIMARY KEY (cod_prova)  
);
```

### 5.3 Tabela “Pessoa”:

Tabela03_Pessoa				
Atributo	Tipo de Dados	Restrição Domínio	Restrição de Entidade	Restrição Referencial
cpf	VARCHAR(14)	NOT NULL	PK(chave primaria)	
nome	VARCHAR(100)	NOT NULL		
idade	INTEGER	CHECK(idade>0)		

SQL

```
CREATE TABLE Pessoa(  
    cpf CHAR(14) NOT NULL,  
    nome VARCHAR(100) NOT NULL,  
    idade INTEGER CHECK(idade>0),  
    PRIMARY KEY (cpf)  
);
```



#### 5.4 Tabela “Advogado”:

Tabela04_Advogado				
Atributo	Tipo de Dados	Restrição Domínio	Restrição de Entidade	Restrição Referencial
num_oab	VARCHAR(10)	NOT NULL	PK(chave primaria)	
nome	VARCHAR(100)			
area_atua	VARCHAR(100)	DEFAULT 'Civil'		

SQL

```
CREATE TABLE Advogado(  
    num_oab VARCHAR(10) NOT NULL,  
    nome VARCHAR(100),  
    area_atua VARCHAR(100) DEFAULT 'Civil',  
    PRIMARY KEY (num_oab)  
);
```

#### 5.5 Tabela “Contrato”:

Tabela05_Contrato				
Atributo	Tipo de Dados	Restrição Domínio	Restrição de Entidade	Restrição Referencial
cpf_pes	VARCHAR(14)	NOT NULL	PK(chave primaria)	FK(chave estrangeira de Pessoa)
num_oab	VARCHAR(10)	NOT NULL	PK(chave primaria)	FK(chave estrangeira de Advogado)
valor_contrato	DECIMAL(9, 2)			
data_contrato	DATE			



SQL

```
CREATE TABLE Contrato(  
    cpf_pes CHAR(14) NOT NULL,  
    num_oab VARCHAR(10) NOT NULL,  
    valor_contrato DECIMAL(9,2),  
    data_contrato DATE,  
    PRIMARY KEY (cpf_pes, num_oab),  
    FOREIGN KEY (cpf_pes) REFERENCES Pessoa(cpf),  
    FOREIGN KEY (num_oab) REFERENCES Advogado(num_oab)  
);
```

### 5.6 Tabela “Orgao\_responsavel”:

Tabela06_Orgao_responsavel				
Atributo	Tipo de Dados	Restrição Domínio	Restrição de Entidade	Restrição Referencial
cod_ident	VARCHAR(10)	NOT NULL	PK(chave primaria)	
cod_Org_fiscalizador	VARCHAR(10)	NOT NULL		FK(chave estrangeira de Orgao_responsavel)
fiscalizado_anomamente	BOOLEAN	DEFAULT FALSE		
tipo_org	VARCHAR(50)	NOT NULL		
nome	VARCHAR(100)			
end_num	VARCHAR(10)			
end_logradouro	VARCHAR(100)			
end_cidade	VARCHAR(50)			





SQL

```
CREATE TABLE Orgao_responsavel(  
    cod_ident VARCHAR(10) NOT NULL,  
    cod_Org_fiscalizador VARCHAR(10),  
    fiscalizado_anonimamente BOOLEAN DEFAULT FALSE,  
    tipo_org VARCHAR(50) NOT NULL,  
    nome VARCHAR(100),  
    end_num VARCHAR(10),  
    end_logradouro VARCHAR(100),  
    end_cidade VARCHAR(50),  
    PRIMARY KEY (cod_ident),  
    FOREIGN KEY (cod_Org_fiscalizador) REFERENCES  
Orgao_responsavel(cod_ident)  
);
```

### 5.7 Tabela “Pessoa\_denunciada”:

Tabela07_Pessoa_denunciada				
Atributo	Tipo de Dados	Restrição Domínio	Restrição de Entidade	Restrição Referencial
cpf	CHAR(14)	NOT NULL	PK(chave primaria)	FK(chave estrangeira de Pessoa)
cod_ident_org	VARCHAR(10)	NOT NULL	PK(chave primaria)	FK(chave estrangeira de Orgao_responsavel)
data_comunicado	DATE			

SQL

```
CREATE TABLE Pessoa_denunciada(  
    cpf CHAR(14) NOT NULL,  
    cod_ident_org VARCHAR(10) NOT NULL,  
    data_comunicado DATE,  
    PRIMARY KEY (cpf, cod_ident_org),  
    FOREIGN KEY (cpf) REFERENCES Pessoa(cpf),  
    FOREIGN KEY (cod_ident_org) REFERENCES Orgao_responsavel(cod_ident)  
);
```



### 5.8 Tabela “Denuncia”:

Tabela08_Denuncia				
Atributo	Tipo de Dados	Restrição Domínio	Restrição de Entidade	Restrição Referencial
num_ident	VARCHAR(10)	NOT NULL	PK(chave primaria)	
status	VARCHAR(20)	DEFAULT 'Aberta' CHECK (status='Em análise' OR status='Encerrada' OR status='Aberta')		
cod_ident_org	VARCHAR(10)	NOT NULL		FK(chave estrangeira de Orgao_responsavel)
metodo_analise	VARCHAR(100)	CHECK (metodo_analise='Automatizada' OR metodo_analise='Perícia técnica')		
relato	TEXT			

SQL

```
CREATE TABLE Denuncia(  
    num_ident VARCHAR(10) NOT NULL,  
    status VARCHAR(20) DEFAULT 'Aberta' CHECK (status IN ('Em análise',  
'Encerrada', 'Aberta')),  
    cod_ident_org VARCHAR(10) NOT NULL,  
    metodo_analise VARCHAR(100) CHECK (metodo_analise IN  
('Automatizada', 'Perícia técnica')),  
    relato TEXT,  
    PRIMARY KEY (num_ident),  
    FOREIGN KEY (cod_ident_org) REFERENCES Orgao_responsavel(cod_ident)  
);
```



### 5.9 Tabela “Pessoa\_denunciante”:

Tabela09_Pessoa_denunciante				
Atributo	Tipo de Dados	Restrição Domínio	Restrição de Entidade	Restrição Referencial
cpf	CHAR(14)	NOT NULL	PK(chave primaria)	FK(chave estrangeira de Pessoa)
relacao_caso	VARCHAR(20)	DEFAULT 'Vítima'	PK(chave primaria)	
num_ident	VARCHAR(10)	NOT NULL	PK(chave primaria)	FK(chave estrangeira de Denuncia)

SQL

```
CREATE TABLE Pessoa_denunciante (  
    cpf CHAR(14) NOT NULL,  
    relacao_caso VARCHAR(20) DEFAULT 'Vítima' CHECK (relacao_caso IN  
    ('Vítima', 'Testemunha')),  
    num_ident VARCHAR(10) NOT NULL,  
    PRIMARY KEY (cpf, relacao_caso, num_ident),  
    FOREIGN KEY (cpf) REFERENCES Pessoa(cpf),  
    FOREIGN KEY (num_ident) REFERENCES Denuncia(num_ident)  
);
```

### 5.10 Tabela “telefonePessoa”:

Tabela10_telefone_Pessoa				
Atributo	Tipo de Dados	Restrição Domínio	Restrição de Entidade	Restrição Referencial
cpf	CHAR(14)	NOT NULL	PK(chave primaria)	FK(chave estrangeira de Pessoa)
telefone	VARCHAR(15)		PK(chave primaria)	



SQL

```
CREATE TABLE telefonePessoa(  
    cpf CHAR(14) NOT NULL,  
    telefone VARCHAR(15),  
    PRIMARY KEY (cpf, telefone),  
    FOREIGN KEY (cpf) REFERENCES Pessoa(cpf));
```

### 5.11 Tabela “Denuncia\_identificada”:

Tabela11_Denuncia_identificada				
Atributo	Tipo de Dados	Restrição Domínio	Restrição de Entidade	Restrição Referencial
num_ident	VARCHAR(10)	NOT NULL	PK(chave primaria)	FK(chave estrangeira de Denuncia)
cpf_denunciante	CHAR(14)	NOT NULL		FK(chave estrangeira de Pessoa)

SQL

```
CREATE TABLE Denuncia_identificada(  
    num_ident VARCHAR(10) NOT NULL,  
    cpf_denunciante CHAR(14) NOT NULL,  
    PRIMARY KEY (num_ident),  
    FOREIGN KEY (cpf_denunciante) REFERENCES Pessoa(cpf),  
    FOREIGN KEY (num_ident) REFERENCES Denuncia(num_ident)  
);
```



### 5.12 Tabela “Denuncia\_anonima”:

Tabela12_Denuncia_anonima				
Atributo	Tipo de Dados	Restrição Domínio	Restrição de Entidade	Restrição Referencial
num_ident	VARCHAR(10)	NOT NULL	PK(chave primaria)	FK(chave estrangeira de Denuncia)

SQL

```
CREATE TABLE Denuncia_anonima(  
    num_ident VARCHAR(10) NOT NULL,  
    PRIMARY KEY (num_ident),  
    FOREIGN KEY (num_ident) REFERENCES Denuncia(num_ident)  
);
```

### 5.13 Tabela “Advogado\_acompanha\_denuncia”:

Tabela13_Advogado_acompanha_denuncia_identificada				
Atributo	Tipo de Dados	Restrição Domínio	Restrição de Entidade	Restrição Referencial
num_oab	VARCHAR(10)	NOT NULL	PK(chave primaria)	FK(chave estrangeira de Advogado)
num_ident	VARCHAR(10)	NOT NULL	PK(chave primaria)	FK(chave estrangeira de Denuncia_identificada)

SQL

```
CREATE TABLE Advogado_acompanha_denuncia(  
    num_oab VARCHAR(10) NOT NULL,  
    num_ident VARCHAR(10) NOT NULL,  
    PRIMARY KEY (num_oab, num_ident),  
    FOREIGN KEY (num_oab) REFERENCES Advogado(num_oab),  
    FOREIGN KEY (num_ident) REFERENCES Denuncia_identificada(num_ident)  
);
```



#### 5.14 Tabela “Denuncia\_contem\_provas”:

Tabela14_Denuncia_contem_provas				
Atributo	Tipo de Dados	Restrição Domínio	Restrição de Entidade	Restrição Referencial
num_ident	VARCHAR(10)	NOT NULL	PK(chave primaria)	FK(chave estrangeira de Denuncia)
cod_prova	VARCHAR(10)	NOT NULL	PK(chave primaria)	FK(chave estrangeira de Prova)

SQL

```
CREATE TABLE Denuncia_contem_provas(  
    num_ident VARCHAR(10) NOT NULL,  
    cod_prova VARCHAR(10) NOT NULL,  
    PRIMARY KEY (num_ident, cod_prova),  
    FOREIGN KEY (num_ident) REFERENCES Denuncia(num_ident),  
    FOREIGN KEY (cod_prova) REFERENCES Prova(cod_prova));
```

#### 5.15 Tabela “Denuncia\_contem\_violacao”:

Tabela15_Denuncia_contem_violacao				
Atributo	Tipo de Dados	Restrição Domínio	Restrição de Entidade	Restrição Referencial
num_ident	VARCHAR(10)	NOT NULL	PK(chave primaria)	FK(chave estrangeira de Denuncia)
cod_vio	VARCHAR(10)	NOT NULL	PK(chave primaria)	FK(chave estrangeira de Violacao)



SQL

```
CREATE TABLE Denuncia_contem_violacao(  
    num_ident VARCHAR(10) NOT NULL,  
    cod_vio VARCHAR(10) NOT NULL,  
    PRIMARY KEY (num_ident, cod_vio),  
    FOREIGN KEY (num_ident) REFERENCES Denuncia(num_ident),  
    FOREIGN KEY (cod_vio) REFERENCES Violacao(cod_vio)  
);
```

## 6. CONSULTAS EM ÁLGEBRA RELACIONAL E CÓDIGO SQL

### 6.1 Listar todas as pessoas denunciantes:

SQL

**ÁLGEBRA RELACIONAL**  
**Pessoa\_denunciante**

SQL

```
SELECT *  
FROM Pessoa_denunciante;
```

### 6.2 Listar todas as denúncias com relatos, status e órgão responsável:

SQL

**ÁLGEBRA RELACIONAL**

$\pi \{d.num\_ident, d.relato, d.status, o.nome\} (Denuncia \bowtie \{d.cod\_ident\_org = o.cod\_ident\} Orgao\_responsavel)$

SQL

```
SELECT d.num_ident, d.relato, d.status, o.nome AS orgao  
FROM Denuncia AS d  
JOIN Orgao_responsavel AS o ON d.cod_ident_org = o.cod_ident  
ORDER BY d.num_ident;
```



### 6.3 Listar denúncias identificadas com nome e idade do denunciante:

SQL

#### ALGEBRA RELACIONAL

```
 $\pi$  {d.num_ident, p.nome, p.idade} (
  (Denuncia_identificada  $\bowtie$  {cpf_denunciante = cpf} Pessoa)
   $\bowtie$  {Denuncia_identificada.num_ident = Denuncia.num_ident} Denuncia
)
```

#### SQL

```
SELECT d.num_ident, p.nome, p.idade
FROM Denuncia_identificada di
JOIN Pessoa AS p ON di.cpf_denunciante = p.cpf
JOIN Denuncia AS d ON di.num_ident = d.num_ident
ORDER BY d.num_ident;
```

### 6.4 Mostrar denúncias com seus relatos e respectivas violações separadas por uma vírgula:

SQL

#### ALGEBRA RELACIONAL

```
 $\gamma$  {d.num_ident, d.relato, STRING_AGG(v.descricao)} (
  (Denuncia  $\bowtie$  {d.num_ident = dv.num_ident} Denuncia_contem_violacao)
   $\bowtie$  {dv.cod_vio = v.cod_vio} Violacao
)
```

#### SQL

```
SELECT d.num_ident, d.relato, STRING_AGG(v.descricao, ', ') AS
violacoes
FROM Denuncia AS d
JOIN Denuncia_contem_violacao AS dv ON d.num_ident = dv.num_ident
JOIN Violacao AS v ON dv.cod_vio = v.cod_vio
GROUP BY d.num_ident, d.relato
ORDER BY d.num_ident;
```





**6.5 Listar denúncias e seus respectivos tipos de provas com descrição, ordenado de maneira crescente pelo número identificador:**

SQL

**ALGEBRA RELACIONAL**

```
 $\pi$  {d.num_ident, p.tipo_prova, p.descricao} (  
  (Denuncia_contem_provas  $\bowtie$  {cod_prova = p.cod_prova} Prova)  
   $\bowtie$  {Denuncia_contem_provas.num_ident = d.num_ident} Denuncia  
)
```

SQL

```
SELECT d.num_ident, p.tipo_prova, p.descricao  
FROM Denuncia_contem_provas AS dp  
JOIN Prova AS p ON dp.cod_prova = p.cod_prova  
JOIN Denuncia AS d ON dp.num_ident = d.num_ident  
ORDER BY num_ident, tipo_prova;
```

**6.6 Listar denúncias feitas por pessoa com mais de 60 anos:**

SQL

**ALGEBRA RELACIONAL**

```
 $\pi$  {d.num_ident, p.nome, p.idade} (  
   $\sigma$  {p.idade > 60} (  
    (Denuncia_identificada  $\bowtie$  {cpf_denunciante = cpf} Pessoa)  
     $\bowtie$  {Denuncia_identificada.num_ident = Denuncia.num_ident} Denuncia  
  )  
)
```

SQL

```
SELECT d.num_ident, p.nome, p.idade  
FROM Denuncia_identificada di  
JOIN Pessoa AS p ON di.cpf_denunciante = p.cpf  
JOIN Denuncia AS d ON di.num_ident = d.num_ident  
WHERE p.idade > 60;
```



### 6.7 Listar denúncias com mais de uma violação e sua respectiva quantidade de violações:

SQL

ALGEBRA RELACIONAL

```
 $\sigma$  {count > 1} (  
   $\gamma$  {num_ident; count(*) -> count} (Denuncia_contem_violacao)  
)
```

SQL

```
SELECT num_ident, COUNT(*) AS total_violacoes  
FROM Denuncia_contem_violacao  
GROUP BY num_ident  
HAVING COUNT(*)>1;
```

### 6.8 Listar denúncias com mais de uma prova e sua respectiva quantidade de provas:

SQL

ALGEBRA RELACIONAL

```
 $\sigma$  {count > 1} (  
   $\gamma$  {num_ident; count(*) -> count} (Denuncia_contem_provas)  
)
```

SQL

```
SELECT num_ident, COUNT(*) AS total_provas  
FROM Denuncia_contem_provas  
GROUP BY num_ident  
HAVING COUNT(*)>1;
```



### 6.9 Listar pessoas denunciadas e o nome do órgão para onde a denúncia foi:

SQL

ALGEBRA RELACIONAL

```
 $\pi$  {p.nome, o.nome} (
  (Pessoa_denunciada  $\bowtie$  {cpf = p.cpf} Pessoa)
   $\bowtie$  {cod_ident_org = o.cod_ident} Orgao_responsavel
)
```

SQL

```
SELECT p.nome AS pessoa_denunciada, o.nome AS orgao
FROM Pessoa_denunciada AS pd
JOIN Pessoa AS p ON pd.cpf = p.cpf
JOIN Orgao_responsavel AS o ON pd.cod_ident_org = o.cod_ident;
```

### 6.10 Listar as denúncias com provas do tipo 'Áudio' associadas a denúncias de assédio e a descrição da prova:

SQL

ALGEBRA RELACIONAL

```
 $\pi$  {d.num_ident, p.descricao} (
   $\sigma$  {dv.cod_vio = 'VI000005'  $\wedge$  p.tipo_prova = 'Áudio'} (
    ((Denuncia  $\bowtie$  {d.num_ident = dp.num_ident} Denuncia_contem_provas)
     $\bowtie$  {dp.cod_prova = p.cod_prova} Prova)
     $\bowtie$  {d.num_ident = dv.num_ident} Denuncia_contem_violacao))
```

SQL

```
SELECT d.num_ident, p.descricao
FROM Denuncia AS d
JOIN Denuncia_contem_provas AS dp ON d.num_ident=dp.num_ident
JOIN Prova AS p ON dp.cod_prova=p.cod_prova
JOIN Denuncia_contem_violacao AS dv ON d.num_ident=dv.num_ident
WHERE dv.cod_vio='VI000005' AND p.tipo_prova='Áudio';
```



### 6.11 Contar quantas denúncias cada órgão recebeu:

SQL

**ALGEBRA RELACIONAL**

```
γ {o.nome; count(d.num_ident) -> total} (  
  Orgao_responsavel ⋈ {o.cod_ident = d.cod_ident_org} Denuncia  
)
```

**SQL**

```
SELECT o.nome AS orgao, COUNT(d.num_ident)  
FROM Orgao_responsavel AS o  
JOIN Denuncia AS d ON o.cod_ident=d.cod_ident_org  
GROUP BY o.nome;
```

### 6.12 Listar o número de identificação de denúncias já encerradas:

SQL

**ALGEBRA RELACIONAL**

```
σ {status = 'Encerrada'} (π {num_ident, status} Denuncia)
```

**SQL**

```
SELECT num_ident, status  
FROM Denuncia  
WHERE status='Encerrada';
```



### 6.13 Listar denúncias com mais de uma pessoa envolvida:

SQL

**ALGEBRA RELACIONAL**

```
 $\pi$  \{d.num\_ident\} (\n  (Denuncia \bowtie \{d.num\_ident = pd.num\_ident\} Pessoa\_denunciante)\n  \bowtie \{d.cod\_ident\_org = pda.cod\_ident\_org\} Pessoa\_denunciada\n)
```

**SQL**

```
SELECT DISTINCT d.num_ident\nFROM Denuncia AS d\nJOIN Pessoa_denunciante AS pd ON d.num_ident = pd.num_ident\nJOIN Pessoa_denunciada AS pda ON d.cod_ident_org = pda.cod_ident_org;
```

### 6.14 Listar denúncias que estão em análise por perícia técnica:

SQL

**ALGEBRA RELACIONAL**

```
 $\sigma$  \{metodo\_analise = 'Perícia técnica'\} (\pi \{num\_ident, metodo\_analise, relato\} Denuncia)
```

**SQL**

```
SELECT num_ident, metodo_analise, relato\nFROM Denuncia\nWHERE metodo_analise='Perícia técnica';
```

### 6.15 Listar todas as violações com suas descrições:

SQL

**ALGEBRA RELACIONAL**

```
 $\pi$  \{cod\_vio, descricao\} (Violacao)
```

**SQL**

```
SELECT cod_vio, descricao FROM Violacao;
```



### 6.16 Listar as provas do tipo 'Documento' associadas a denúncias com mais de uma violação:

SQL

ALGEBRA RELACIONAL

```
Temp <- σ {count > 1} (γ {num_ident; count(*) -> count}  
(Denuncia_contem_violacao))
```

```
π {p.cod_prova, p.descricao} (  
  σ {p.tipo_prova = 'Documento'} (  
    (Prova ⋈ {p.cod_prova = dp.cod_prova} Denuncia_contem_provas)  
    ⋈ {dp.num_ident = Temp.num_ident} Temp  
  )  
)
```

SQL

```
SELECT DISTINCT p.cod_prova, p.descricao  
FROM Prova AS p  
JOIN Denuncia_contem_provas AS dp ON p.cod_prova=dp.cod_prova  
JOIN (  
  SELECT num_ident  
  FROM Denuncia_contem_violacao  
  GROUP BY num_ident  
  HAVING COUNT(*) > 1  
) AS dv ON dp.num_ident = dv.num_ident  
WHERE p.tipo_prova = 'Documento';
```

### 6.17 Contar quantas vezes cada tipo de prova aparece:

SQL

ALGEBRA RELACIONAL

```
γ {tipo_prova; count(*) -> total} (Prova)
```

SQL

```
SELECT tipo_prova, COUNT(*) AS total FROM Prova  
GROUP BY tipo_prova;
```



### 6.18 Ver denúncias em que a pessoa denunciante também é a pessoa denunciada:

SQL

#### ALGEBRA RELACIONAL

```
 $\pi \{di.num\_ident, p.nome\} ($   
     $(Denuncia\_identificada \bowtie \{di.cpf\_denunciante = pd.cpf\}$   
     $Pessoa\_denunciada)$   
     $\bowtie \{pd.cpf = p.cpf\} Pessoa$   
)
```

SQL

```
SELECT di.num_ident, p.nome  
FROM Denuncia_identificada AS di  
JOIN Pessoa_denunciada AS pd ON di.cpf_denunciante = pd.cpf  
JOIN Pessoa AS p ON p.cpf = pd.cpf;
```

### 6.19 Listar denúncias acompanhadas por advogados:

SQL

#### ALGEBRA RELACIONAL

```
 $\pi \{aad.num\_ident, a.nome\} ($   
     $Advogado\_acompanha\_denuncia \bowtie \{num\_oab = a.num\_oab\} Advogado$   
)
```

SQL

```
SELECT aad.num_ident, a.nome AS nome_advogado  
FROM Advogado_acompanha_denuncia AS aad  
JOIN Advogado AS a ON aad.num_oab = a.num_oab;
```



## 7. INFORMAÇÕES DOS ARQUIVOS

- O arquivo “tabelasSIGVID.sql” é para criação das tabelas do banco de dados para o SIGVID.
- O arquivo “insertsSIGVID.sql” é para o povoamento do banco de dados criado.
- O arquivo “selectsSIGVID.sql” contém exemplos de consultas no banco de dados da SIGVID.
- O arquivo “SIGVID – Sistema Integrado de Gestão de Violações e Denúncias.pdf” é o relatório técnico com as informações necessárias para todo o entendimento do estudo de caso “Sistema de Denúncia de Violação de Direitos Humanos”.
- O arquivo “BDDERSIGVID.png” é uma representação gráfica conhecida como Diagrama Entidade-Relacionamento.
- O arquivo “BDMERSIGVID.png” é uma representação gráfica conhecida como Mapeamento Entidade-Relacionamento.

Todos os arquivos são de autoria própria e foram feitos nas seguintes ferramentas: “[Visual Studio Code](#)”, “[Draw.io](#)”, “[Google Docs](#)” e “[Google Sheets](#)”. Sendo utilizado como ferramenta de testes a “pdAdmin4” do “[PostgreSQL](#)”.