

# Notificações de síndrome gripal leve - RJ 2021

Integrantes: Marcelle Iannuzzi, Beatriz Santos, Guilherme Soares, Daniel Firmino, Gabriel Quintela  
DRE's: 118056472, 116054270, 117247410, 115160387, 119062905

# Tópicos abordados

01

Tema do trabalho

02

Modelo Conceitual

03

Modelo Lógico

04

Modelo Físico



Conversão de dados

05

Aplicação Web -  
Linguagem e Framework

06

Banco de dados

07

Consultas

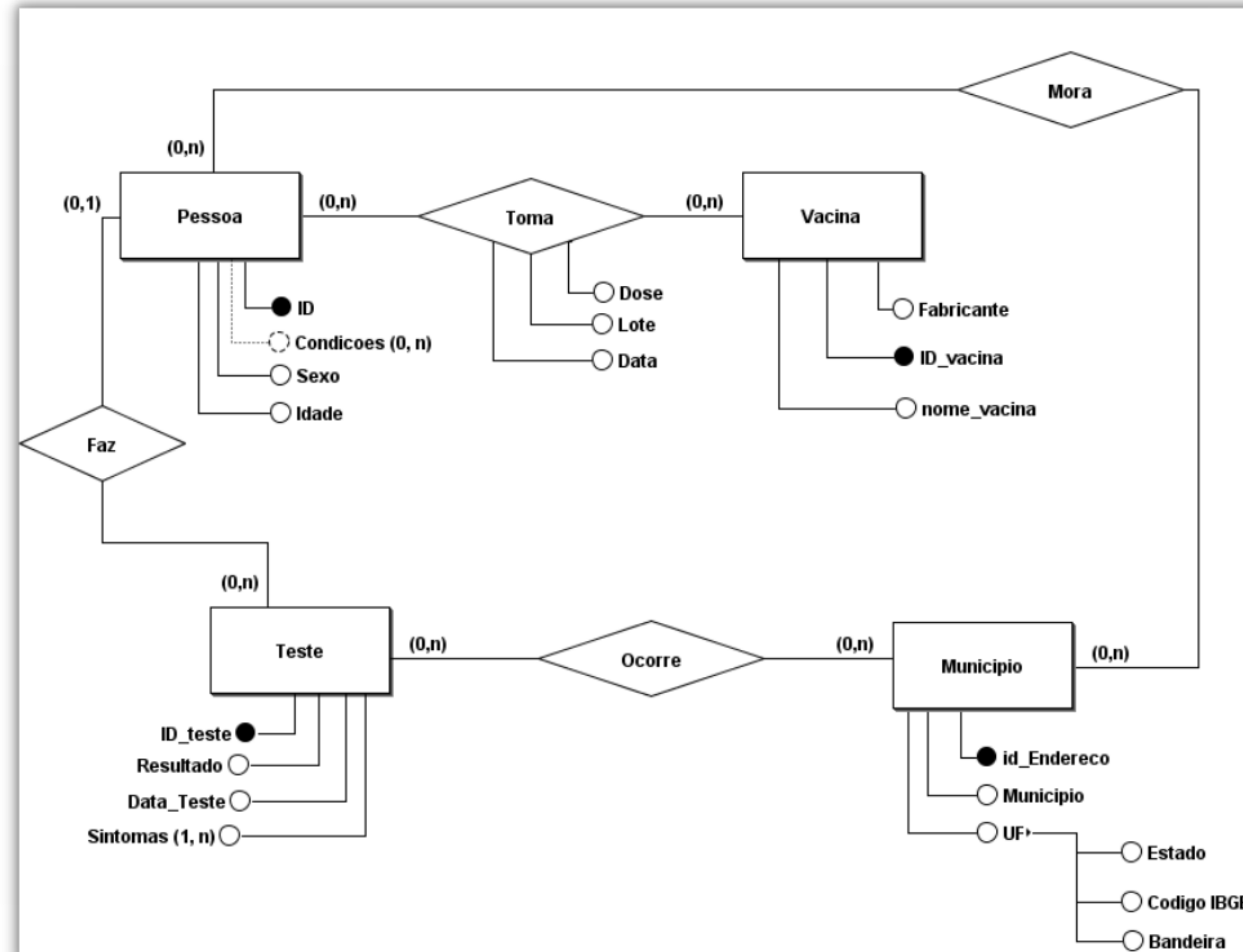
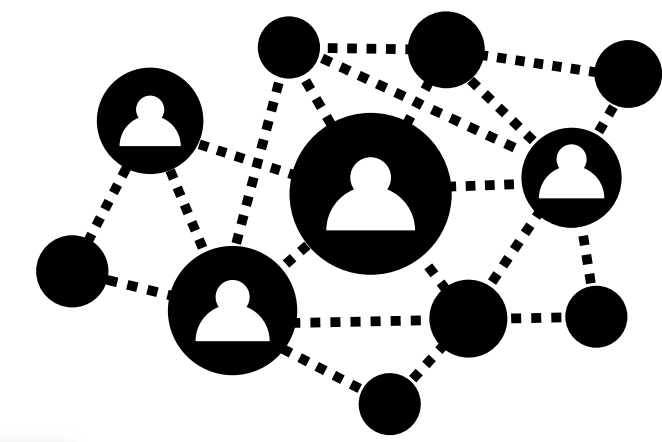
08



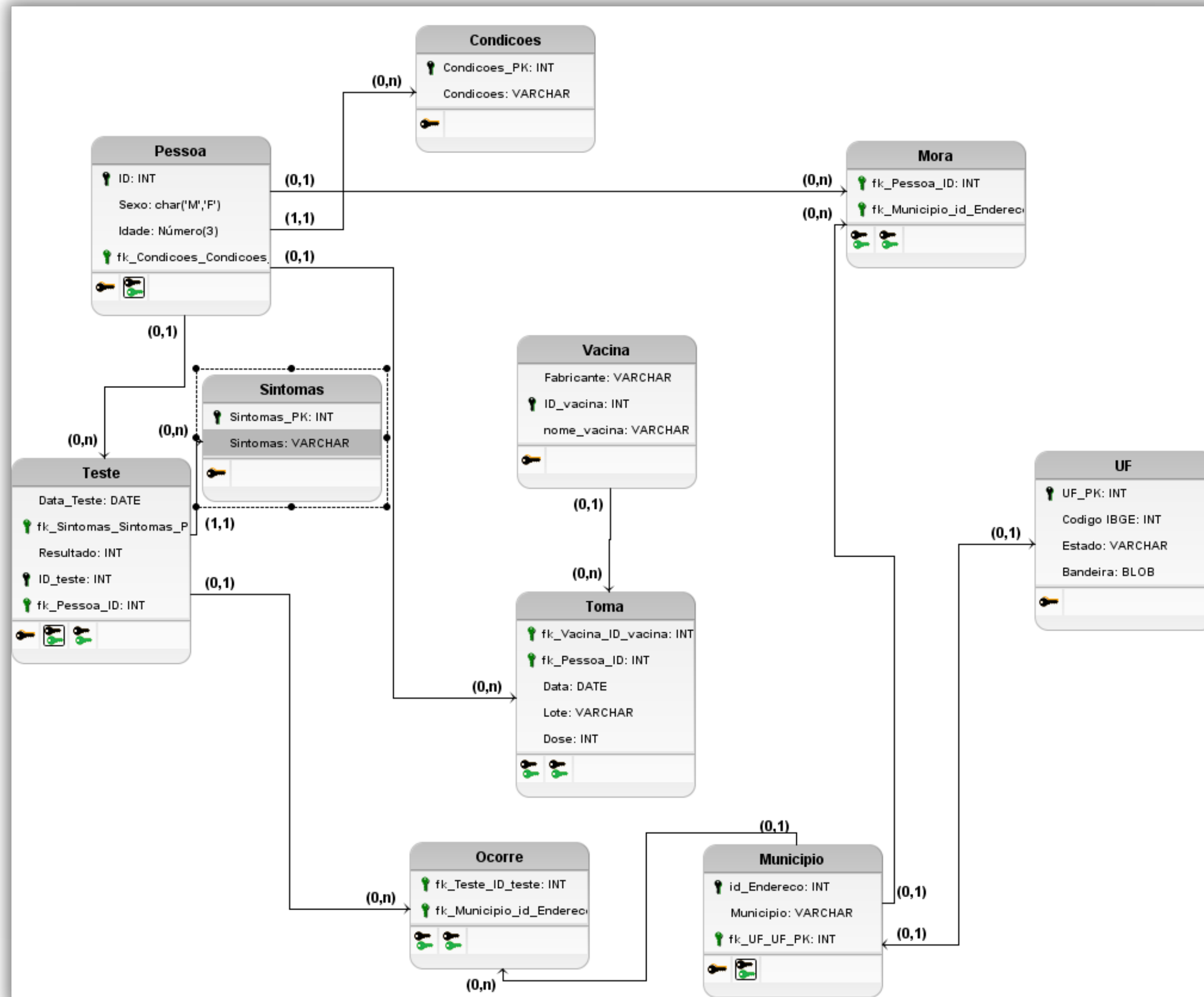
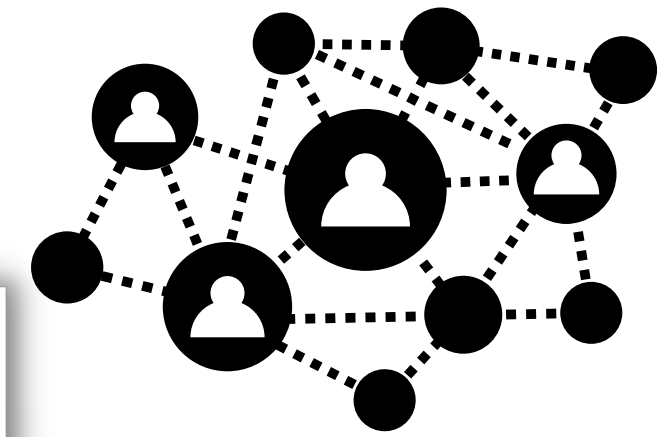
Nosso tema foram as Notificações de síndrome gripal no ano de 2021. No site do DataSUS, pegamos os dados restritos ao Rio de Janeiro, referentes ao Lote 1 do conjunto total de dados.

# Tema do trabalho

# Modelo Conceitual / Diagrama ER



# Modelo Lógico





# Modelo Físico

1

```
CREATE DATABASE bd_Open_DataSus_BD1;

USE bd_Open_DataSus_BD1;

CREATE TABLE Pessoa (
  id_Pessoa int PRIMARY KEY,
  sexo varchar(1),
  idade int
);

CREATE TABLE Condicao (
  id_Condicao int PRIMARY KEY,
  condicao varchar(256),
  fk_id_Pessoa int
);

CREATE TABLE Vacina (
  id_Vacina int PRIMARY KEY,
  fabricante varchar(256),
  nome_Vacina varchar(256)
);

CREATE TABLE Municipio (
  id_Municipio int PRIMARY KEY,
  municipio varchar(256),
  fk_id_Uf int
);

CREATE TABLE Teste (
  id_Testes int PRIMARY KEY,
  data_Testes DATE,
  resultado int,
  fk_id_Pessoa int
);
```

2

```
CREATE TABLE Uf (
  id_Uf int NOT NULL PRIMARY KEY,
  codigo_ibge int,
  estado varchar(256),
  bandeira BLOB
);

CREATE TABLE Sintomas (
  id_Sintomas int NOT NULL PRIMARY KEY,
  sintomas varchar(256),
  fk_id_Testes int
);

CREATE TABLE Toma (
  fk_id_Vacina int,
  fk_id_Pessoa int,
  data DATE,
  lote varchar(256),
  dose int
);

CREATE TABLE Ocorre (
  fk_id_Testes int,
  fk_id_Municipio int
);

CREATE TABLE Mora (
  fk_id_Pessoa int,
  fk_id_Municipio int
);
```

3

```
ALTER TABLE Municipio ADD CONSTRAINT fk_Municipio_Uf
FOREIGN KEY (fk_id_Uf)
REFERENCES Uf(id_Uf)
ON DELETE SET NULL;

ALTER TABLE Sintomas ADD CONSTRAINT fk_Sintomas_Testes
FOREIGN KEY (fk_id_Testes)
REFERENCES Testes(id_Testes)
ON DELETE NO ACTION;

ALTER TABLE Condicao ADD CONSTRAINT fk_Condicao_Pessoa
FOREIGN KEY (fk_id_Pessoa)
REFERENCES Pessoa(id_Pessoa)
ON DELETE NO ACTION;

ALTER TABLE Testes ADD CONSTRAINT fk_Testes_Pessoa
FOREIGN KEY (fk_id_Pessoa)
REFERENCES Pessoa(id_Pessoa)
ON DELETE SET NULL;

ALTER TABLE Toma ADD CONSTRAINT fk_Toma_Vacina
FOREIGN KEY (fk_id_Vacina)
REFERENCES Vacina(id_Vacina)
ON DELETE SET NULL;
```

4

```
ALTER TABLE Toma ADD CONSTRAINT fk_Toma_Pessoa
FOREIGN KEY (fk_id_Pessoa)
REFERENCES Pessoa(id_Pessoa)
ON DELETE SET NULL;

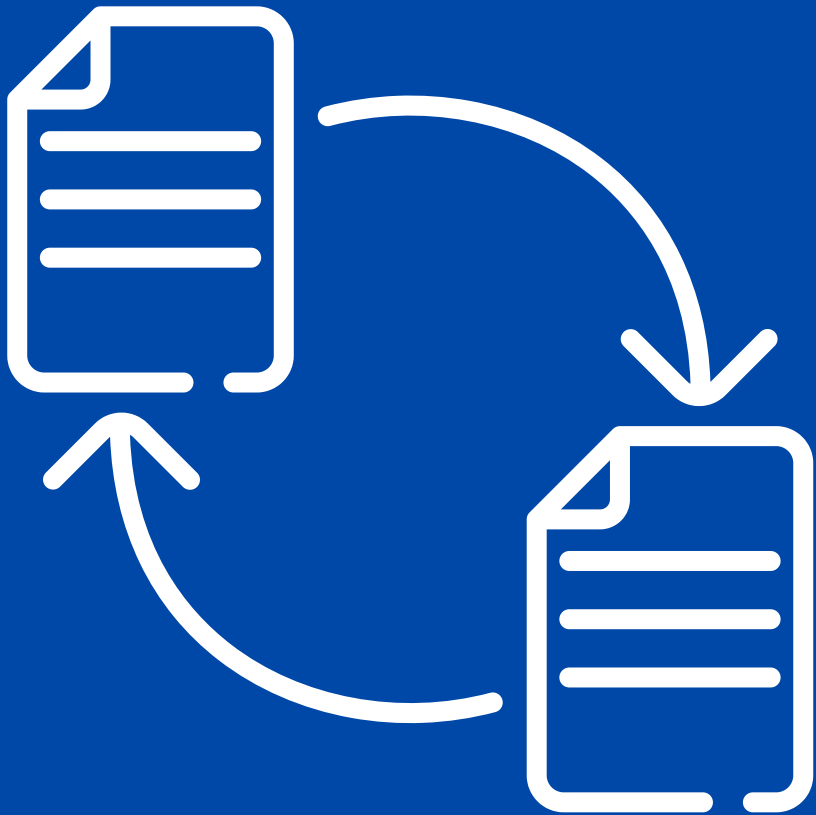
ALTER TABLE Ocorre ADD CONSTRAINT fk_Ocorre_Testes
FOREIGN KEY (fk_id_Testes)
REFERENCES Testes(id_Testes)
ON DELETE SET NULL;

ALTER TABLE Ocorre ADD CONSTRAINT fk_Ocorre_Municipio
FOREIGN KEY (fk_id_Municipio)
REFERENCES Municipio(id_Municipio)
ON DELETE SET NULL;

ALTER TABLE Mora ADD CONSTRAINT fk_Mora_Pessoa
FOREIGN KEY (fk_id_Pessoa)
REFERENCES Pessoa(id_Pessoa)
ON DELETE SET NULL;

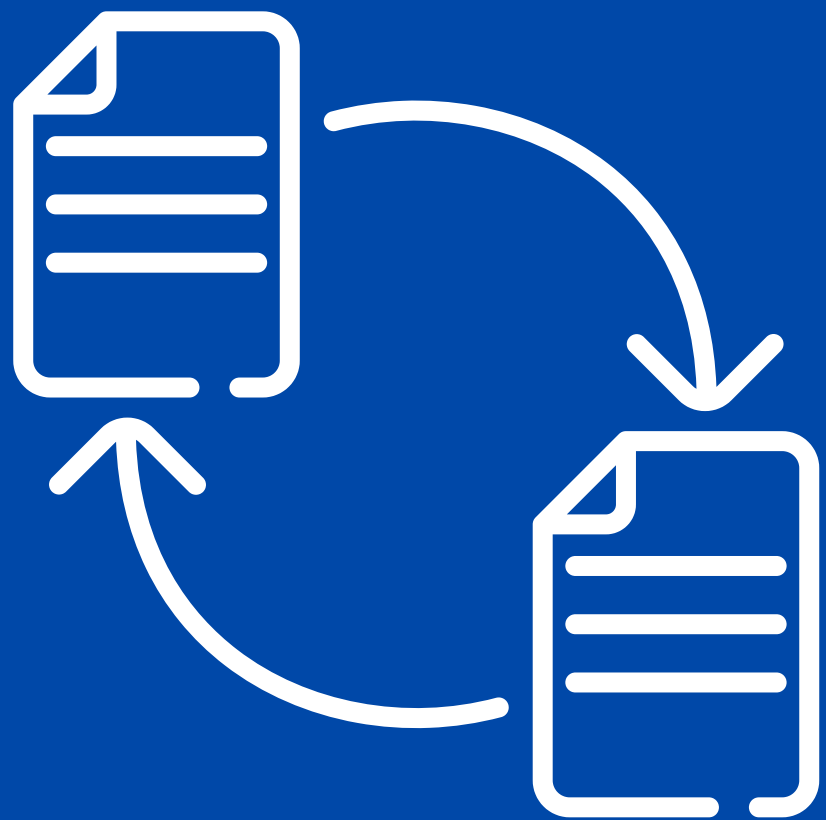
ALTER TABLE Mora ADD CONSTRAINT fk_Mora_Municipio
FOREIGN KEY (fk_id_Municipio)
REFERENCES Municipio(id_Municipio)
ON DELETE SET NULL;
```

# Conversão de dados



| <u>condicoes</u>     | sexo      | idade     | estado         | <u>estado</u> BGE |
|----------------------|-----------|-----------|----------------|-------------------|
|                      | Feminino  | 26        | Rio de Janeiro |                   |
| <u>AXICA, CORIZA</u> | Masculino | 50        | Rio de Janeiro |                   |
|                      | Feminino  | 22        | Rio de Janeiro |                   |
|                      | Masculino | 18        | Rio de Janeiro |                   |
|                      | Feminino  | 49        | Rio de Janeiro |                   |
| Doenças respirató    | Feminino  | 32        | Rio de Janeiro |                   |
|                      | Masculino | 46        | Rio de Janeiro |                   |
|                      | Feminino  | 21        | Rio de Janeiro |                   |
|                      | Feminino  | 55        | Rio de Janeiro |                   |
|                      | Masculino | 45        | Rio de Janeiro |                   |
| Portador de doenç    | Masculino | 25        | Rio de Janeiro | RJ                |
|                      | Masculino | 27        | Rio de Janeiro |                   |
|                      | Feminino  | 42        | Rio de Janeiro | RJ                |
|                      | Feminino  | 24        | Rio de Janeiro |                   |
|                      | Feminino  | 39        | Rio de Janeiro |                   |
| Doenças cardíaca     | Feminino  | 47        | Rio de Janeiro |                   |
|                      | Masculino | 40        | Rio de Janeiro |                   |
|                      | Feminino  | 31        | Rio de Janeiro |                   |
|                      | Feminino  | 33        | Rio de Janeiro |                   |
|                      | Feminino  | 86        | Rio de Janeiro |                   |
|                      | Masculino | 54        | Rio de Janeiro |                   |
|                      | Masculino | 35        | Rio de Janeiro |                   |
|                      | Feminino  | 41        | Rio de Janeiro |                   |
|                      |           | Masculino | 68             | Rio de Janeiro    |
|                      | Feminino  | 28        | Rio de Janeiro |                   |
| Doenças cardíaca     | Feminino  | 72        | Rio de Janeiro |                   |
|                      | Feminino  | 52        | Rio de Janeiro |                   |
| Distúrbios           | Masculino | 77        | Rio de Janeiro |                   |

# Conversão de dados



```
class Toma:
    def __init__(self, vacina_id, pessoa_id, data, lote, dose):
        # divide nome e fabricante separados por '/'
        self.vacina = vacina_id
        self.pessoa = pessoa_id
        self.data = data
        self.lote = lote
        self.dose = dose

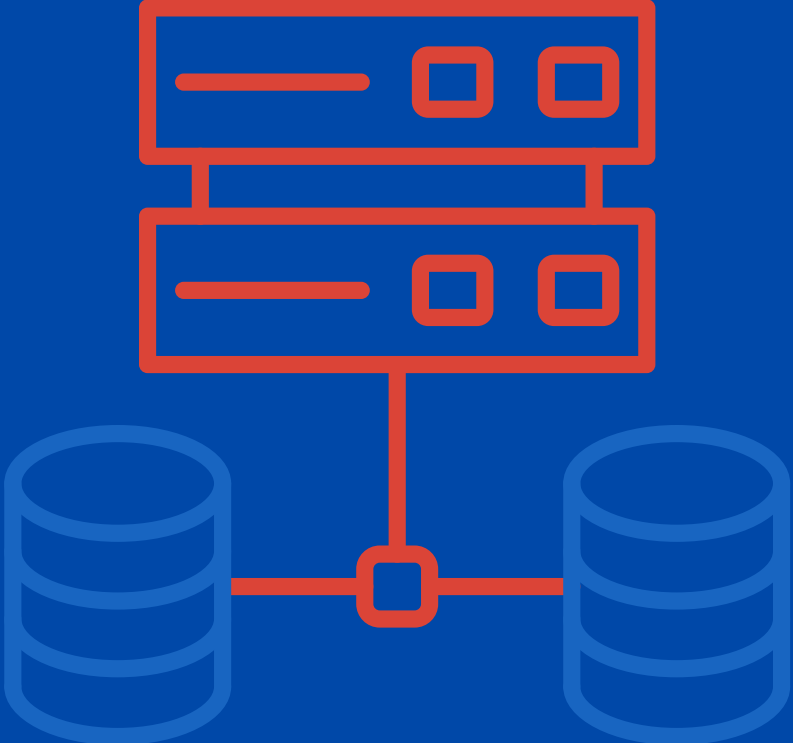
    def to_sql(self):
        return "INSERT INTO Toma VALUES (" + str(self.vacina) + ", " + str(self.pessoa) + ", '" + \
            + self.data + "', '" + str(self.lote) + "', " + str(self.dose) + ") ;\n"

    def to_string(self):
        return "(" + str(self.vacina) + ", " + str(self.pessoa) + ", '" + \
            + self.data + "', '" + str(self.lote) + "', " + str(self.dose) + ")"

# a partir de um prefixo "INSERT INTO Table VALUES " gera sql para inserir multiplas
# tuplas modeladas em tuples
def gera_insert_sql(prefix, tuples):
    # insere as tuplas separadas por virgula apos o prefixo terminando com ;
    return prefix + ",".join(tuples) + ";\n"
```



# Consultas



Retornar todos os sintomas encontrados:

Select sintomas From Sintomas;

Qual a vacina mais aplicada no município 'Rio de Janeiro':

```
SELECT nome_vacina, num_vac FROM
(SELECT count(nome_vacina) as num_vac, nome_vac FROM Toma NATURAL JOIN Mora INNER JOIN
Vacina ON Toma.fk_id_Vacina = id_vacina INNER JOIN Municipio ON Mora.fk_id_municipio = id_Municipio
WHERE municipio = 'Rio de Janeiro' GROUP BY nome_vacina) AS cont_vac WHERE num_vac >= ALL (
SELECT count(nome_vacina) FROM Toma NATURAL JOIN Mora INNER JOIN Vacina ON Toma.fk_id_Vacina
= id_vacina INNER JOIN Municipio ON Mora.fk_id_municipio = id_MunicipioToma NATURAL JOIN Mora
INNER JOIN Vacina ON Toma.fk_id_Vacina = id_vacina
WHERE municipio = 'Rio de Janeiro' GROUP BY nome_vacina);
```

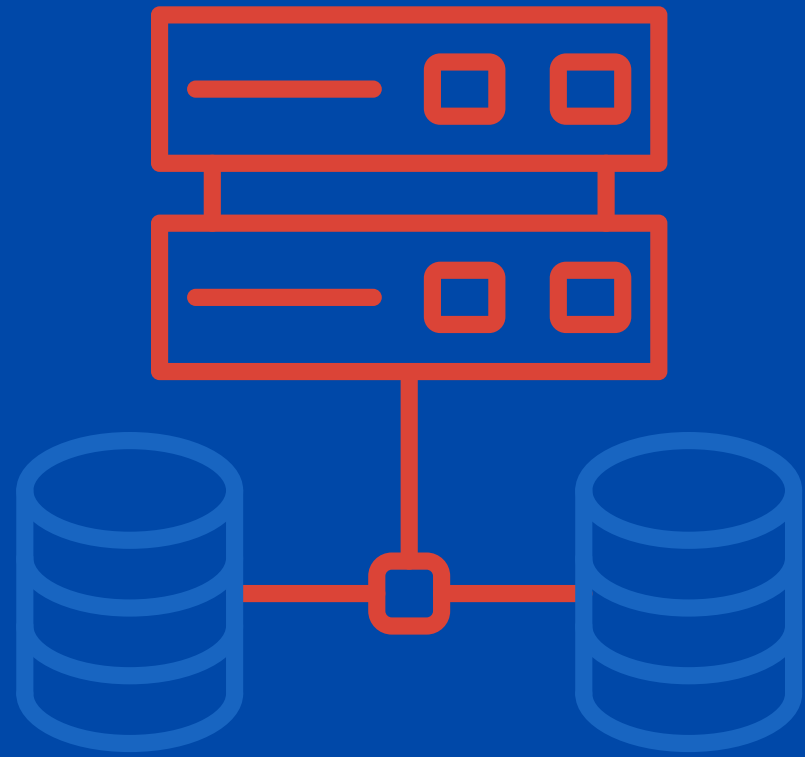
Qual o numero de pessoas que apresentaram o sintoma 'febre':

```
SELECT sintomas, count(*) FROM Teste INNER JOIN Sintomas
ON Sintomas.fk_id_Testes = id_Testes
WHERE Sintomas.sintomas = 'Febre'
GROUP BY sintomas;
```

Qual o caso de teste positivo mais recente por município:

```
SELECT MAX(data_Testes), municipio FROM Teste NATURAL
JOIN Mora INNER JOIN Municipio ON Municipio.id_Municipio =
Mora.fk_id_Municipio WHERE Testes.resultado = 1 GROUP BY
Municipio;
```

# Consultas



Mostrar idade da pessoa e data de vacinação de todas as pessoas que tomaram coronavac (SINOVAC):

```
SELECT idade, data FROM Toma INNER JOIN Pessoa ON Pessoa.ID = Toma.fk_id _pessoa INNER JOIN Vacina ON Toma.fk_id_vacina = Vacina.id_vacina WHERE nome_vacina = 'SINOVAC';
```

Mostrar todas as pessoas que moram no municipio Rio de Janeiro e tem idade acima de 95:

```
SELECT id FROM Pessoa INNER JOIN Mora ON id_pessoa = fk_id_pessoa INNER JOIN Municipio ON fk_id_municipio = id_municipio WHERE idade > 95 AND municipio = 'Rio de Janeiro';
```

Mostrar todas as pessoas que não tomaram nenhuma dose de Vacina:

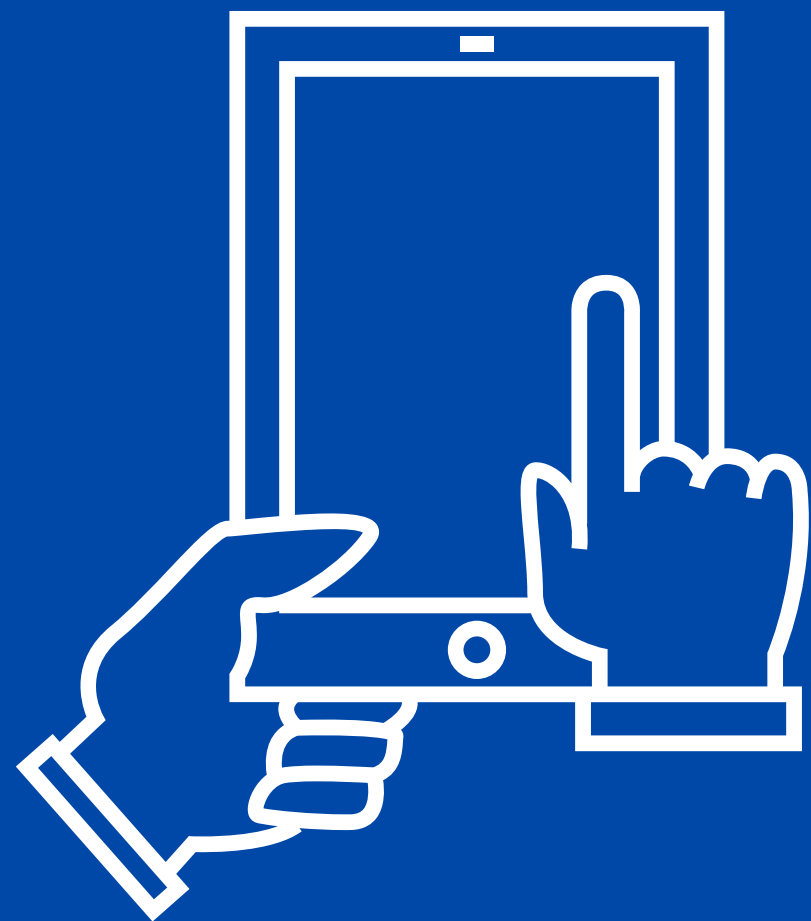
```
SELECT ID_pessoa FROM Pessoa LEFT OUTER JOIN Toma ON Pessoa.ID_pessoa = Toma.fk_id_Pessoa WHERE fk_ID_vacina IS NULL;
```

Mostrar todos os municipios que houveram testes feitos

```
SELECT Municipio FROM Municipio INNER JOIN Ocorre ON Municipio.id_municipio = Ocorre.fk_id_Municipio GROUP BY Municipio;
```

Listar todos os dias que aconteceu vacinação e/ou teste:

```
(SELECT distinct (Data_Testes) FROM Testes) UNION (SELECT distinct (Data) FROM Toma);
```



# Aplicação Web - Linguagem e Framework



**Perguntas?**