

## Covid Tracking

Relatório BDAD 2020/2021

G201

André de Jesus Fernandes Flores – up201907001

Diogo Luís Araujo de Faria – up201907014

Tiago Andre Batista Rodrigues – up201906807

# Table of Contents

Introdução .....	4
Classes .....	5
Location .....	5
Country .....	5
District .....	5
County .....	5
Parish .....	5
Vaccine .....	5
VaccinationAmount .....	6
Manufacturer .....	6
Case .....	6
EmploymentSector .....	6
Ethnicity .....	7
Outcome .....	7
NursingHome .....	7
Hospitalization .....	7
ICUStay .....	7
Ventilation .....	8
Hospital .....	8
Modelo Conceptual .....	9
Esquema Relacional .....	10
Análise a Forma Normal .....	11
Vaccine .....	11
EmploymentSector .....	11
Hospitalization .....	12
EthnicityCOVIDCase e EmployedIn .....	12
Strain .....	12
Geral .....	12
Restrições .....	13
Country .....	13
District .....	13
County .....	13
Parish .....	13
Vaccine .....	13

VaccinationAmount.....	14
Manufacturer .....	14
VaccineManufacturer.....	14
Strain .....	14
NursingHome .....	15
EmploymentSector.....	15
Ethnicity.....	15
COVIDCase.....	15
EthnicityCOVIDCase.....	15
Hospital .....	16
Hospitalization.....	16
ICUStay .....	16
Ventilation.....	16
EmployedIn .....	17
Restrições Adicionais.....	17
Interrogações .....	17

# Introdução

A base de dados visa guardar informação relativa aos casos de COVID-19 num dado país de modo a ser possível analisar a evolução do estado da pandemia ao longo do tempo. Para a esta análise ser o mais detalhada possível a informação geográfica dos casos será dividida por várias unidades administrativas sendo a mais pequena a freguesia. Também se intende providenciar contexto a cada caso de COVID-19 registado através de outra informação relevante.

Uma grande inspiração para este projeto é o [COVID HYPERLINK "https://covidtracking.com/Tracking](https://covidtracking.com/Tracking) [HYPERLINK "https://covidtracking.com/](https://covidtracking.com/) Project.

# Classes

## Location

A classe “Location” serve de generalização para os elementos que apresentem como atributos comuns um identificador locID, um name, /caseNumber e /noVaccinated. É uma classe completa e disjunta.

### Country

Esta classe representa um país, que possui um atributo a mais, o /population que representa a população do país, derivado a partir da população de cada distrito, “**District**”.

Cada país tem como suas partes pelo menos um distrito, “**District**”, e pode ser o país de origem de alguma variante de SARS-CoV-2, guardada na classe “**Strain**”.

### District

Esta classe representa um distrito, que possui um atributo a mais, o /population que representa a população do distrito, derivado a partir da população de cada “**County**”.

Cada distrito faz parte de um e um só país, “**Country**”, sendo que tem como suas partes pelo menos um concelho, “**County**”.

### County

Esta classe representa um concelho, que possui um atributo a mais, o /population que representa a população do concelho, derivado a partir da população de cada freguesia, “**Parish**”.

Cada concelho faz parte de um e um só distrito, “**District**”, sendo que tem como suas partes pelo menos uma freguesia, “**Parish**”.

### Parish

Esta classe representa uma freguesia, que possui um atributo a mais, o population que representa a sua população.

Cada freguesia faz parte de um e um só concelho, “**County**”. Associada a cada freguesia pode haver inúmeros casos, “**Case**”, e também inúmeras vacinas administradas, “**Vaccine**”, numa relação que apresenta uma classe de associação, “**VacciantionAmount**”. Em cada freguesia podem não existir ou existir inúmeros hospitais, “**Hospital**”.

## Vaccine

Esta classe representa uma vacina e tem como atributos um identificador vacID, um name, que representa a designação da vacina utilizado, e um doseNumber, que representa o número de doses necessárias.

Cada vacina específica pode existir em várias ou inúmeras freguesias, **“Parish”**, numa relação em que apresenta uma classe de associação, **“VaccinationAmount”**. Cada vacina pode ter sido criada apenas por um ou por uma colaboração de vários fabricantes, **“Manufacturer”**.

## VaccinationAmount

Esta classe é uma classe de associação entre freguesia, **“Parish”**, e vacina, **“Vaccine”**, que representa o número de vacinas de um certo tipo presentes numa certa freguesia e que tem como atributos `vaccinationNumber` e `date`, que representa a data em que se verificou a quantidade de vacinas existentes de forma a se poder estudar a variação da mesma em relação aos casos, **“Case”**.

## Manufacturer

Esta classe representa um fabricante de uma determinada vacina e tem como atributos um identificador `manId` e um `name`, que representa o nome do fabricante.

Como na base de dados se está apenas a considerar fabricantes que realmente já desenvolveram uma vacina e, pelos dados que obtivemos, considera-se que um fabricante vai estar ligado uma e uma só vez a uma vacina, **“Vaccine”**.

## Strain

Esta classe representa cada variante do vírus SARS-CoV-2, tendo como atributos um identificador `strainID` e `designation`.

Cada variante é originada em um e uma só país, **“Country”**, existindo então uma ligação com a classe mencionada e também pode estar associada a inúmeros casos, **“Case”**.

## Case

Esta classe representa um caso, tendo como atributos um identificador `caseID`, uma `detectionDate` e `endDate`, que representam a data de deteção do caso e de fim, respetivamente, e `birthYear`, que representa o ano de nascimento da pessoa que foi infetada.

Cada caso vai representar alguém infetado com uma e uma só variante do vírus, **“Strain”**, que foi detetado numa só freguesia, **“Parish”**, que representa alguém de uma ou mais etnias, **“Ethnicity”**, e que podem ter inúmeros empregos ou nenhum em certos setores, **“EmploymentSector”** ou estar num lar, **“NursingHome”**. Em cada caso pode ou não acontecer uma e uma só hospitalização, **“Hospitalization”**, visto que de acordo com os dados que obtemos, uma pessoa apenas sai de um hospital apenas se já foi curada ou se faleceu.

## EmploymentSector

Esta classe representa o setor de emprego em que uma pessoa representada por um caso, **“Case”**, trabalha e tem como atributos um identificador `sectorId`, um `name`, que representa o nome do setor e `contactFrequency`, que é dada por um número entre 0 e 5 e que representa a frequência de contacto entre pessoas em diferentes setores.

Cada setor pode estar ligado a nenhum ou inúmeros casos, **“Case”**.

## Ethnicity

Esta classe representa a etnia de alguém representado num caso, “**Case**”, e tem como atributos um identificador, etnlID, e um name, que representa o nome da etnia.

Cada etnia diferente pode estar ligada a nenhum ou inúmeros casos diferentes, “**Case**”.

## Outcome

Esta classe representa o resultado de uma infecção do vírus SARS-CoV-2 e tem como atributo um type, que representa o tipo de resultado, sendo que pode ter 2 valores: 0, que significa que o caso, “**Case**”, a que está relacionada acabou em morte com causa de COVID-19, 1, que significa que houve um óbito não relacionado a COVID-19, ou 2, que significa que houve recuperação.

Visto que vários casos, “**Case**”, podem e vão ter apenas um de dois resultados, esta classe está pode estar ligada a nenhum ou inúmeros casos, “**Case**”.

## NursingHome

Esta classe representa um lar e tem como atributos um identificador nursingHomeID e um name, que representa o nome do lar.

Cada lar pode estar associado a nenhum ou inúmeros casos, “**Case**”.

## Hospitalization

Esta classe representa uma hospitalização e tem como atributos um identificador hospStayID, e uma startDate e endDate, que representam a data de início e fim da hospitalização, respetivamente.

Em cada hospitalização podem acontecer ou não inúmeras vezes a necessidade de se estar na unidade de cuidados intensivos, “**ICUStay**”. Uma hospitalização vai estar associada a um ou mais hospitais, “**Hospital**”, visto que existe a possibilidade de um doente ser reencaminhado para diferentes hospitais em certas situações.

## ICUStay

Esta classe representa estar numa unidade de cuidados intensivos e tem como atributos um identificador ICUStayID, e uma startDate e endDate, que representam a data de início e fim em que se está na unidade, respetivamente.

Cada visita à ICU vai ser associada a uma e uma só hospitalização, “**Hospitalization**”, e a zero ou inúmeras vezes à necessidade de ser utilizado um ventilador, “**Ventilation**”.

## Ventilation

Esta classe representa o uso de um ventilador e tem como atributos um identificador ventlID, e uma startDate e endDate, que representam a data em que se começou e se acabou de usar um ventilador, respetivamente.

Cada uso de ventilador vai estar associado a uma e uma só estadia numa unidade de cuidados intensivos, "**ICUStay**", visto que apenas se utilizam ventiladores quando se está na unidade.

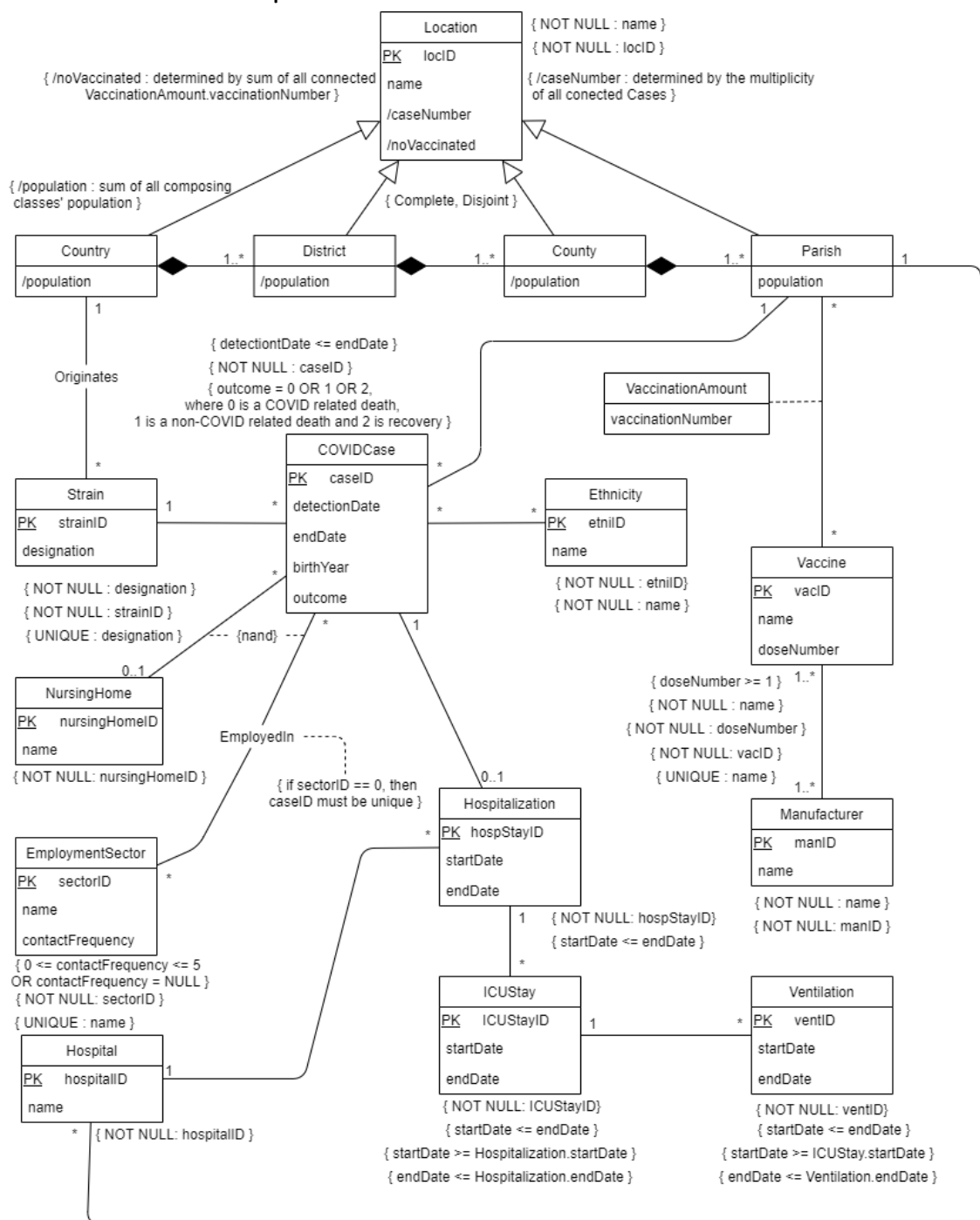
## Hospital

Esta classe representa um hospital e tem como atributos um identificador hospitalID e um name, que representa o nome do hospital.

Cada hospital está localizado numa e numa só freguesia, "**Parish**".



## Modelo Conceptual



# Esquema Relacional

**Country**(locID, name, caseNumber, noVaccinated, population)

- locID -> name, caseNumber, noVaccinated, population

**District**(locID, name, caseNumber, noVaccinated, population, countryID -> Country)

- locID -> name, caseNumber, noVaccinated, population, countryID

**County**(locID, name, caseNumber, noVaccinated, population, districtID -> District)

- locID -> name, caseNumber, noVaccinated, population, districtID

**Parish**(locID, name, caseNumber, noVaccinated, population, countyID -> County)

- locID -> name, caseNumber, noVaccinated, population, countyID

**Vaccine**(vacID, name, doseNumber)

- vacID -> name, doseNumber
- name -> doseNumber

**VaccinationAmount**(locID -> Parish, vacID -> Vaccine, vaccinationNumber)

- locID, vacID -> vaccinationNumber

**Manufacturer**(manID, name)

- manID -> name

**VaccineManufacturer**(vacID -> Vaccine, manID -> Manufacturer)

**Strain**(strainID, designation, countryID -> Country)

- strainID -> designation, countryID
- designation -> strainID, countryID

**Ethnicity**(ethniID, name)

- ethniID -> name

**EthnicityCOVIDCase**(caseID -> COVIDCase, ethniID -> ethnicity)

**NursingHome**(nursingHomeID, name)

- nursingHomeID -> name

**COVIDCase**(caseID, detectionDate, endDate, birthyear, outcome, parishID -> Parish, strainID -> Strain, nursingHomeID -> NursingHome)

- caseID -> detectionDate, endDate, birthyear, outcome, parishID, strainID, nursingHomeID

**EmploymentSector**(sectorID, name, contactFrequency)

- sectorID -> name, contactFrequency

- name -> sectorID, contactFrequency

**EmployedIn**(caseID -> COVIDCase, sectorID -> EmploymentSector)

**Hospitalization**(hospStayID, startDate, endDate, hospitalID -> Hospital, caseID->COVIDCase)

- hospStayID -> startDate, endDate, hospitalID, caseID
- caseID-> hospStayID, startDate, endDate, hospitalID

**ICUStay**(ICUStayID, startDate, endDate, hospStayID -> Hospitalization)

- ICUStayID -> startDate, endDate, hospStayID

**Ventilation**(ventID, startDate, endDate, ICUStayID -> ICUStay)

- ventID -> startDate, endDate, ICUStayID

**Hospital**(hospitalID, name, parishID -> Parish)

- hospitalID -> name, parishID

## Análise a Forma Normal

Vaccine

Na relação Vaccine, em name -> doseNumber, devido ao facto de que o atributo do lado esquerdo (name) nem é uma chave nem superchave, pode-se considerar uma violação BCNF e, também, devido a que o atributo do lado direito (doseNumber) não é um atributo primo, pode-se considerar uma violação à 3NF.

Uma possível solução, seria decompor a relação em:

- VaccineA(name, doseNumber), com name -> doseNumber
- VaccineB(vacID, name), com vacID -> name

Estas, já se encontram na BCNF, e por consequência na 3NF, visto que apenas apresentam uma relação não trivial em que do lado esquerdo está a sua chave primária.

No entanto, neste contexto, não se utilizam as relações decompostas, visto que vacID é um atributo utilizado apenas de forma a simplificar a identificação da relação, sendo mais simples do que uma string, que neste contexto pode ser longa, e permitindo uma menor existência de erros no povoamento.

EmploymentSector

Nesta relação, as duas dependências funcionais apresentadas apresentam do lado esquerdo uma chave primária (caseID) ou uma superchave (name), pelo que se pode considerar a relação como estando em BCNF e, por sua vez, em 3NF.

#### Hospitalization

Na relação Hospitalization, apresentam-se duas dependências funcionais, em que uma no lado esquerda tem uma chave primária e outra um superchave, podendo então se considerar a relação como estando em BCNF e, por consequência, em 3NF.

#### EthnicityCOVIDCase e EmployedIn

Estas relações não apresentam nenhuma dependência funcional não trivial, pelo que podem ser consideradas com estando em BCNF e, por sua vez, em 3NF.

#### Strain

Nesta relação existem duas dependências funcionais, as quais apresentam do lado esquerdo uma chave (strainID) ou superchave (designation). Logo, podem-se considerar em BCNF, e por consequência, em 3NF.

#### Geral

As relações restantes apresentam apenas uma dependência funcional, em que do lado esquerdo está sua chave primária, podendo, assim, ser consideradas como estando em BCNF e, por consequência, em 3NF.

# Restrições

## Country

- Não podem haver dois países com o mesmo locID
  - locID PRIMARY KEY
- Não podem haver países com o campo name NULL
  - name NOT NULL

## District

- Não podem haver dois distritos com o mesmo locID
  - locID PRIMARY KEY
- Não podem haver distritos com o campo name NULL
  - name NOT NULL
- Cada distrito tem de estar obrigatoriamente ligado a um e um só país
  - countryID REFERENCES Country(locID) NOT NULL

## County

- Não podem haver dois concelhos com o mesmo locID
  - locID PRIMARY KEY
- Não podem haver concelhos com o campo name NULL
  - name NOT NULL
- Cada concelho tem de estar obrigatoriamente ligado a um e um só distrito
  - districtID REFERENCES District(locID) NOT NULL

## Parish

- Não podem haver duas freguesias com o mesmo locID
  - locID PRIMARY KEY
- Não podem haver freguesias com o campo name NULL
  - name NOT NULL
- Cada freguesia tem de estar obrigatoriamente ligada a um e um só concelho
  - countyID REFERENCES County(locID) NOT NULL

## Vaccine

- Não podem haver duas vacinas com o mesmo vacID
  - vacID PRIMARY KEY
- Todas as vacinas têm de ter um nome e este não pode ser repetido

- name UNIQUE NOT NULL
- Todas as vacinas têm de ter um número de doses a ser administrado e este tem de ser maior ou igual que um
  - doseNumber CHECK(doseNumber >= 1) NOT NULL

### VaccinationAmount

- Não podem haver duas VaccinationAmount com o mesmo par de (parishID, vacID)
  - PRIMARY KEY (parishID, vacID)
- O parishID deve corresponder a um locID da tabela Parish e o vacID a um vacID da tabela Vaccine, nenhum pode ser NULL
  - parishID REFERENCES Parish(locID) NOT NULL
  - vacID REFERENCES Vaccine(locID) NOT NULL
- O número de vacinas administrado deve ser sempre maior ou igual a zero
  - CHECK(vaccinationNumber >= 0)

### Manufacturer

- Não podem haver dois fabricantes com o mesmo manID
  - manID PRIMARY KEY
- Todos os fabricantes têm de ter um nome e este não pode ser repetido
  - name UNIQUE NOT NULL

### VaccineManufacturer

- Não podem haver dois VaccineManufacturer com o mesmo par (vacID, manID)
  - PRIMARY KEY (vacID, manID)
- O vacID tem de corresponder a um vacID da tabela Vaccine e o manID tem de corresponder a um manID da tabela Manufacturer, nenhum pode ser NULL
  - vacID REFERENCES Vaccine(vacID) NOT NULL
  - manID REFERENCES Manufacturer(manID) NOT NULL

### Strain

- Não podem haver duas estirpes com o mesmo strainID
  - strainID PRIMARY KEY
- Todas as estirpes têm de ter um designação e esta não pode ser repetida
  - Designation UNIQUE NOT NULL
- O countryID tem de corresponder a um locID da tabela Country
  - countryID REFERENCES Country(locID)

## NursingHome

- Não podem haver dois lares de repouso com o mesmo nursingHomeID
  - nursingHomeID PRIMARY KEY

## EmploymentSector

- Não podem haver dois setores de emprego com o mesmo sectorID
  - sectorID PRIMARY KEY
- Todos os setores de emprego têm de ter um nome e este não pode ser repetido
  - name UNIQUE NOT NULL
- A frequência de contacto com outras pessoas terá de ser um valor entre 0 e 5 (inclusive) ou poderá ser NULL se o sectorID for 0 (row reservada para Desempregado)
  - CHECK((contactFrequency >= 0 AND contactFrequency <= 5 and contactFrequency IS NOT NULL AND sectorID <> 0) OR (sectorID = 0 AND contactFrequency IS NULL))

## Ethnicity

- Não podem haver duas etnias com o mesmo ethniID
  - ethniID PRIMARY KEY
- Todas as etnias têm de ter um nome e este não pode ser repetido
  - name UNIQUE NOT NULL

## COVIDCase

- Não podem haver dois casos com o mesmo caseID
  - caseID PRIMARY KEY
- O parishID tem de corresponder a um locID da tabela Parish e não pode ser NULL, o mesmo com o strainID e o strainID da tabela Strain. O nursingHomeID tem de corresponder a um nursingHomeID da tabela NursingHome e pode ser NULL.
  - parishID REFERENCES Parish(locID) NOT NULL
  - strainID REFERENCES Strain(strainID) NOT NULL
  - nursingHomeID REFERENCES NursingHome(nursingHomeID)
- A data da deteção do caso tem de ser menor ou igual a data do seu fim
  - CHECK(detectionDate <= endDate)
- O resultado do caso tem de estar entre 0 e 2 inclusive
  - CHECK(outcome >= 0 AND outcome <= 2)

## EthnicityCOVIDCase

- Não podem haver dois EthnicityCOVIDCase com o mesmo par (caseID, ethniID)
  - PRIMARY KEY (caseID, ethniID)

- O caseID tem de corresponder a um caseID da tabela COVIDCase e não pode ser NULL, o mesmo para o ethniID e o ethniID da tabela Ethnicity
  - caseID REFERENCES COVIDCase(caseID) NOT NULL
  - ethniID REFERENCES Ethnicity(ethniID) NOT NULL

### Hospital

- Não podem haver dois hospitais com o mesmo hospitalID
  - hospitalID PRIMARY KEY
- O parishID tem de corresponder a um locID da tabela Parish e não pode ser NULL
  - parishID REFERENCES Parish(locID) NOT NULL

### Hospitalization

- Não podem haver duas hospitalizações com os mesmo hospStayID
  - hospStayID PRIMARY KEY
- O caseID tem de corresponder a um caseID da tabela COVIDCase e não pode ser NULL, o mesmo para o hospitalID e o hospitalID da tabela Hospital
  - caseID REFERENCES COVIDCase(caseID) NOT NULL
  - hospitalID REFERENCES Hospital(hospitalID) NOT NULL
- A data do começo de um hospitalização tem de ser menor ou igual a data do seu fim
  - CHECK(startDate <= endDate)

### ICUStay

- Não podem haver duas ICUStays com o mesmo ICUStayID
  - ICUStayID PRIMARY KEY
- O hospStayID tem de corresponder a um hospStayID da tabela Hospitalization e não pode ser NULL
  - hospStayID REFERENCES Hospitalization(hospStayID) NOT NULL
- A data de começo da estadia na Unidade de Cuidados Intensivos tem de ser menor ou igual a data do seu fim
  - CHECK(startDate <= endDate)
- A data de começo da estadia na Unidade de Cuidados Intensivos tem de ser maior ou igual a data de começo da sua hospitalização correspondente
- A data de fim da estadia na Unidade de Cuidados Intensivos tem de ser menor ou igual a data de fim da sua hospitalização correspondente

### Ventilation

- Não podem haver duas Ventilations com o mesmo ventID
  - ventID PRIMARY KEY



- O ICUStayID tem de corresponder a um ICUStayID da tabela ICUStay e não pode ser NULL
  - ICUStayID REFERENCES ICUStay(ICUStayID) NOT NULL
- A data de começo da ventilação tem de ser menor ou igual a data do seu fim
  - CHECK(startDate <= endDate)
- A data de começo da ventilação tem de ser maior ou igual a data de começo da sua estadia na Unidade de Cuidados Intensivos correspondente
- A data de fim da ventilação tem de ser menor ou igual a data de fim da sua estadia na Unidade de Cuidados Intensivos correspondente

### EmployedIn

- Não podem haver dois EmployedIn com o mesmo par (caseID, sectorID)
  - PRIMARY KEY (caseID, sectorID)
- O caseID corresponde a um caseID da tabela COVIDCase e não pode ser NULL, o mesmo para o sectorID e o sectorID da tabela EmploymentSector
  - caseID REFERENCES COVIDCase(caseID) NOT NULL
  - sectorID REFERENCES EmploymentSector(sectorID) NOT NULL
- Se um COVIDCase estiver ligado ao EmploymentSector designado para sem ocupação (sectorID == 0), não pode estar ligado a outros EmploymentSector.

### Restrições Adicionais

Um COVIDCase não poderá estar ligado a uma NursingHome e a um EmploymentSector ao mesmo tempo

NOTA: As restrições especificadas mas cuja implementação não está definida são impossíveis de implementar sem recurso a triggers e, como tal, apenas serão implementadas na 3ª entrega.

Para além disso, na base de dados em SQL também estão implementadas restrições ON UPDATE e ON DELETE para manter a integridade dos dados armazenados.

## Interrogações

1. As três freguesias com mais casos (sem utilizar max());
2. Comparação da taxa de mortalidade de pessoas hospitalizadas com pessoas não hospitalizadas;
3. Hospitais que apenas tem casos da sua propria freguesia;
4. Casos internados em comum;
5. Numero de novas hospitalizações por mês;
6. Estirpe com mais casos por freguesia;
7. Casos detetados em Abril de 2020 ordenados pelo tipo de trabalho respetivo;

8. Cidade mais afetada em 2021;
9. Número de casos ativos em ICU por hospital;
10. Média de casos em cidades com hospitais vs cidades sem hospitais.

## Gatilhos

1. Atualizar população de cidade depois de ser eliminado algum tuplo de freguesia;
2. Impedir pessoas que estejam ao mesmo tempo associadas desemprego e a emprego nalgum setor;
3. Atualizar número de casos de uma freguesia quando se adiciona um caso nessa mesma.