



Universidade do Minho

**Escola de Engenharia**

Mestrado Integrado em Engenharia Biomédica

**Bases de Dados Clínicas e de Gestão Hospitalar**

# **Reações Adversas às vacinas da COVID-19**

Trabalho elaborado pelo grupo nº 4:

António da Cruz Castelo Carneiro da Rocha (A84647)

Lara Alexandra Pereira Novo Martins Vaz (A88362)

Manuel Pinto de Matos (A88357)

Mariana Lindo Carvalho (A88360)

Braga, 7 de junho de 2021

## Sumário

No presente trabalho desenvolveu-se uma base de dados para analisar os efeitos adversos associados às vacinas da COVID-19, de forma a preveni-los e/ou minimizá-los em futuras vacinações de outros pacientes. Para tal, foi utilizado o *dataset* “COVID-19 World Vaccine Adverse Reactions” proveniente do *kaggle*, cujos dados foram submetidos a um tratamento prévio. De seguida, abordou-se a construção dos respetivos modelos conceptual, relacional e físico, bem como todos os procedimentos, funções e *triggers* envolvidos no povoamento e análise do modelo.

Com base no trabalho desenvolvido, verificou-se que bebés e homens com mais de 65 anos com hiperlipidemia, hipertensão, hipotireoidismo ou GERD são os principais grupos de risco na vacinação contra a COVID-19.

# Índice

1. Introdução .....	10
1.1. Contextualização .....	10
1.2. Motivação e objetivos .....	10
2. Análise do <i>Dataset</i> .....	11
3. Modelo Conceptual .....	15
4. Modelo lógico .....	16
4.1. Entidades .....	17
4.2. Atributos .....	18
4.3. Chave-primária e chave-estrangeira .....	20
5. Modelo físico .....	21
5.1. Pré-povoamento .....	21
5.1.1. Tabela temp_doenca .....	23
5.1.2. Tabela temp_fornecedor .....	25
5.1.3. Tabela temp_historico .....	26
5.1.4. Tabela temp_hospital .....	29
5.1.5. Tabela temp_paciente .....	31
5.1.6. Tabela temp_sintomas .....	35
5.1.7. Tabela temp_sintomas2 .....	36
5.1.8. Tabela temp_vacina .....	38
5.1.9. Tabela temp_vacinar .....	40
5.2. Povoamento .....	42
5.2.1. Tabela fornecedor .....	42
5.2.2. Tabela vacina .....	43
5.2.3. Tabela paciente .....	44
5.2.4. Tabela vacinar .....	45
5.2.5. Tabela historico_clinico .....	47
5.2.6. Tabela doenca_afeta_paciente .....	49
5.2.7. Tabela hospitalizacao .....	50
5.2.8. Tabela paciente_e_hospitalizado .....	51
5.2.9. Tabela sintomas .....	52
5.2.10. Tabela sintomas_afeta_paciente .....	54
5.2.11. Tabela vacina_causa_sintomas .....	55
6. Resultados e Discussão .....	56
6.1. Query's, funções e procedimentos .....	56

6.1.1.	Análise da amostra .....	56
6.1.2.	Indicadores de saúde .....	60
6.1.3.	Indicadores das vacinas.....	64
6.1.4.	Indicadores da vacinação .....	66
6.1.5.	Sintomas.....	72
6.1.6.	Morte.....	90
6.2.	Triggers.....	105
7.	Interface .....	107
8.	Análise crítica e conclusão .....	113
9.	Bibliografia .....	115

## Índice de Figuras

<b>Figura 1.</b>	Modelo conceptual. ....	16
<b>Figura 2.</b>	Modelo lógico. ....	17
<b>Figura 3.</b>	Query's efetuadas na tabela temp_doenca. ....	24
<b>Figura 4.</b>	Resultado obtido na query efetuada na zona 3 na tabela temp_da_doenca.....	24
<b>Figura 5.</b>	Resultado obtido na query efetuada na zona 4 na tabela temp_da_doenca.....	24
<b>Figura 6.</b>	Parte da informação presente na tabela temp_da_doenca. ....	24
<b>Figura 7.</b>	Query's efetuadas na tabela temp_fornecedor.....	25
<b>Figura 8.</b>	Resultado obtido na query efetuada na zona 3 na tabela temp_do_fornecedor. ....	26
<b>Figura 9.</b>	Parte da informação presente na tabela temp_do_fornecedor.....	26
<b>Figura 10.</b>	Query's efetuadas na tabela temp_historico (parte 1).....	27
<b>Figura 11.</b>	Query's efetuadas na tabela temp_historico (parte 2).....	28
<b>Figura 12.</b>	Resultado obtido na query efetuada na zona 3 na tabela temp_do_historico. ....	28
<b>Figura 13.</b>	Parte da informação presente na tabela temp_do_historico (parte 1).....	28
<b>Figura 14.</b>	Parte da informação presente na tabela temp_do_historico (parte 2).....	29
<b>Figura 15.</b>	Query's efetuadas na tabela temp_hospital (parte 1).....	30
<b>Figura 16.</b>	Query's efetuadas na tabela temp_hospital (parte 2).....	30
<b>Figura 17.</b>	Resultado obtido na query efetuada na zona 3 na tabela temp_do_hospital. ....	30
<b>Figura 18.</b>	Resultado obtido na query efetuada na zona 5 na tabela temp_do_hospital. ....	31
<b>Figura 19.</b>	Parte da informação presente na tabela temp_do_hospital.....	31
<b>Figura 20.</b>	Script desenvolvida para converter o formato das datas de mm/dd/yyyy para o formato yyyy-mm-dd. ....	33
<b>Figura 21.</b>	Query's efetuadas na Tabela temp_paciente (parte 1). ....	33
<b>Figura 22.</b>	Query's efetuadas na Tabela temp_paciente (parte 2). ....	34
<b>Figura 23.</b>	Resultado obtido na query efetuada na zona 5 na tabela temp_dos_pacientes. ....	34
<b>Figura 24.</b>	Parte da informação presente na tabela temp_dos_pacientes. ....	34
<b>Figura 25.</b>	Query's efetuadas na tabela temp_sintomas (parte 1). ....	35
<b>Figura 26.</b>	Query's efetuadas na tabela temp_sintomas (parte 2). ....	36
<b>Figura 27.</b>	Resultado obtido na query efetuada na zona 3 na tabela temp_dos_sintomas.....	36
<b>Figura 28.</b>	Parte da informação presente na tabela temp_dos_sintomas. ....	36
<b>Figura 29.</b>	Query's efetuadas na tabela temp_sintomas2. ....	37
<b>Figura 30.</b>	Resultado obtido na query efetuada na zona 4 na tabela temp_dos_sintomas2. ....	38

<b>Figura 31.</b> Parte da informação presente na tabela temp_dos_sintomas2. ....	38
<b>Figura 32.</b> Query's efetuadas na tabela temp_vacina (parte 2). ....	39
<b>Figura 33.</b> Query's efetuadas na tabela temp_vacina (parte 1). ....	39
<b>Figura 34.</b> Resultado obtido na query efetuada na zona 5 na tabela temp_das_vacinas. ....	40
<b>Figura 35.</b> Parte da informação presente na tabela temp_das_vacinas. ....	40
<b>Figura 36.</b> Query's efetuadas na Tabela temp_vacinar. ....	41
<b>Figura 37.</b> Resultado obtido na query efetuada na zona 2 na tabela temp_vacinar. ....	41
<b>Figura 38.</b> Parte da informação presente na tabela temp_vacinar. ....	41
<b>Figura 39.</b> Query's efetuadas para povoar a tabela fornecedor e confirmar os resultados obtidos no povoamento. ....	43
<b>Figura 40.</b> Tabela fornecedor. ....	43
<b>Figura 41.</b> Query's efetuadas para povoar a tabela vacina e confirmar os resultados obtidos no povoamento. ....	44
<b>Figura 42.</b> Parte da informação presente na tabela vacina. ....	44
<b>Figura 43.</b> Query's efetuadas para povoar a tabela paciente e confirmar os resultados obtidos no povoamento. ....	45
<b>Figura 44.</b> Parte da informação presente na tabela paciente. ....	45
<b>Figura 45.</b> Query's efetuadas para povoar a tabela vacinar e confirmar os resultados obtidos no povoamento. ....	46
<b>Figura 46.</b> Parte da informação presente na tabela vacinar. ....	47
<b>Figura 47.</b> Query's efetuadas para povoar a tabela historico_clinico e confirmar os resultados obtidos no povoamento. ....	48
<b>Figura 48.</b> Tabela historico_clinico. ....	48
<b>Figura 49.</b> Query's efetuadas para povoar a tabela doenca_afeta_paciente e confirmar os resultados obtidos no povoamento. ....	49
<b>Figura 50.</b> Parte da informação presente na tabela doenca_afeta_paciente. ....	50
<b>Figura 51.</b> Query's efetuadas para povoar a tabela hospitalizacao e confirmar os resultados obtidos no povoamento. ....	50
<b>Figura 52.</b> Tabela hospitalizacao. ....	51
<b>Figura 53.</b> Query's efetuadas para povoar a tabela paciente_e_hospitalizado e confirmar os resultados obtidos no povoamento. ....	51
<b>Figura 54.</b> Parte da informação presente na tabela paciente_e_hospitalizado. ....	52
<b>Figura 55.</b> Query's efetuadas para povoar a tabela sintomas e confirmar os resultados obtidos no povoamento. ....	53
<b>Figura 56.</b> Query's efetuadas para povoar a tabela sintomas e confirmar os resultados obtidos no povoamento (Continuação). ....	53
<b>Figura 57.</b> Parte da informação presente na tabela sintomas. ....	53
<b>Figura 58.</b> Query's efetuadas para povoar a tabela sintomas_afeta_paciente e confirmar os resultados obtidos no povoamento. ....	54
<b>Figura 59.</b> Parte da informação presente na tabela sintomas_afeta_paciente. ....	55
<b>Figura 60.</b> Query's efetuadas para povoar a tabela sintomas_afeta_paciente e confirmar os resultados obtidos no povoamento. ....	55
<b>Figura 61.</b> Parte da tabela vacina_causa_sintomas. ....	56
<b>Figura 62.</b> Query efetuada para determinar o número de pacientes da amostra. ....	56
<b>Figura 63.</b> Número de pacientes da amostra. ....	57
<b>Figura 64.</b> Query's efetuadas para a percentagem de mulheres e homens na amostra. ....	57
<b>Figura 65.</b> Percentagem de mulheres na amostra. ....	57
<b>Figura 66.</b> Percentagem de homens na amostra. ....	57

<b>Figura 67.</b> Procedure PERCENTAGEM_GRUPO_PACIENTES. ....	58
<b>Figura 68.</b> Call's do Procedure PERCENTAGEM_GRUPO_PACIENTES. ....	58
<b>Figura 69.</b> Percentagem de bebês na amostra. ....	58
<b>Figura 70.</b> Percentagem de crianças na amostra. ....	59
<b>Figura 71.</b> Percentagem de adolescentes na amostra. ....	59
<b>Figura 72.</b> Percentagem de jovens na amostra. ....	59
<b>Figura 73.</b> Percentagem de adultos na amostra. ....	59
<b>Figura 74.</b> Percentagem de idosos na amostra. ....	59
<b>Figura 75.</b> Query efetuada para determinar os estados americanos mais frequentes. ....	60
<b>Figura 76.</b> 3 estados americanos mais frequentes e respetiva percentagem de pacientes. ....	60
<b>Figura 77.</b> Query efetuada para determinar a percentagem de pacientes sintomáticos. ....	60
<b>Figura 78.</b> 3 Percentagem de pacientes sintomáticos. ....	60
<b>Figura 79.</b> Query efetuada para determinar os sintomas mais frequentes associados à vacinação. ....	61
<b>Figura 80.</b> Top 10 dos sintomas mais frequentes associados à vacinação. ....	61
<b>Figura 81.</b> Query efetuada para determinar a percentagem de pacientes hospitalizados. ....	62
<b>Figura 82.</b> Percentagem de pacientes hospitalizados. ....	62
<b>Figura 83.</b> Query efetuada para determinar o top 10 dos dias em que as pessoas ficaram hospitalizadas. ....	62
<b>Figura 84.</b> Top 10 dos dias em que as pessoas ficaram hospitalizadas. ....	63
<b>Figura 85.</b> Query efetuada para determinar a percentagem de pacientes que morreram. ....	63
<b>Figura 86.</b> Percentagem de pacientes que morreram. ....	63
<b>Figura 87.</b> Query efetuada para determinar a percentagem de pacientes que têm doenças. ..	63
<b>Figura 88.</b> Percentagem de pacientes que têm doenças. ....	64
<b>Figura 89.</b> Query efetuada para determinar o top 3 das patologias mais frequentes. ....	64
<b>Figura 90.</b> Top 3 das patologias mais frequentes. ....	64
<b>Figura 91.</b> Query efetuada para determinar a percentagem de inoculação por marca de vacina. .....	65
<b>Figura 92.</b> Percentagem de inoculação por marca de vacina. ....	65
<b>Figura 93.</b> Query efetuada para determinar a percentagem de inoculação das vacinas provenientes dos lotes mais frequentes. ....	66
<b>Figura 94.</b> Percentagem de inoculação das vacinas provenientes dos lotes mais frequentes. ...	66
<b>Figura 95.</b> Query efetuada para determinar a percentagem de inoculação por número de doses de vacina. ....	67
<b>Figura 96.</b> Percentagem de inoculação por número de doses de vacina. ....	68
<b>Figura 97.</b> Query efetuada para determinar a percentagem de inoculação das vacinas por forma de inoculação. ....	68
<b>Figura 98.</b> Percentagem de inoculação das vacinas por forma de inoculação. ....	69
<b>Figura 99.</b> Query efetuada para determinar a percentagem de inoculação por local de injeção da vacina. ....	69
<b>Figura 100.</b> Percentagem de inoculação por local de injeção da vacina. ....	70
<b>Figura 101.</b> Query efetuada para determinar a percentagem de inoculação das vacinas por setor. ....	70
<b>Figura 102.</b> Percentagem de inoculação das vacinas por setor. ....	71
<b>Figura 103.</b> Query efetuada para determinar a percentagem de inoculação das vacinas por mês. ....	71
<b>Figura 104.</b> Percentagem de inoculação das vacinas por mês. ....	72

<b>Figura 105.</b> Procedure efetuado para determinar a percentagem de pacientes que foram às urgências e/ou receberam visita do médico por sintoma. ....	73
<b>Figura 106.</b> Call's do procedure SINTOMAS_HOSPITALIZACAO para as 4 combinações possíveis da tabela hospitalizacao. ....	74
<b>Figura 107.</b> Percentagem de pacientes que foram às urgências e receberam visita do médico por sintoma. ....	74
<b>Figura 108.</b> Percentagem de pacientes que foram às urgências, mas não receberam visita do médico por sintoma. ....	74
<b>Figura 109.</b> Percentagem de pacientes que foram não foram às urgências, mas receberam visita do médico por sintoma. ....	75
<b>Figura 110.</b> Percentagem de pacientes que não foram às urgências, nem receberam visita do médico por sintoma. ....	75
<b>Figura 111.</b> Query efetuada para determinar os sintomas que as pessoas que tiveram mais tempo hospitalizadas tinham. ....	76
<b>Figura 112.</b> Sintomas que as pessoas que tiveram mais tempo hospitalizadas tinham. ....	76
<b>Figura 113.</b> Query efetuada para determinar o tempo médio até se iniciarem os sintomas. ...	77
<b>Figura 114.</b> Função Diferença. ....	77
<b>Figura 115.</b> Tempo médio até se iniciarem os sintomas. ....	78
<b>Figura 116.</b> Query efetuada para determinar o tempo médio até se iniciarem os sintomas. ...	78
<b>Figura 117.</b> Tempo médio até se iniciarem os sintomas. ....	78
<b>Figura 118.</b> Query efetuada para determinar a percentagem de pacientes que recuperaram completamente dos sintomas. ....	79
<b>Figura 119.</b> Percentagem de pacientes que recuperaram completamente dos sintomas. ....	79
<b>Figura 120.</b> Procedure efetuado para determinar a percentagem de pacientes que apresentam determinada patologia por sintoma. ....	80
<b>Figura 121.</b> Call's do procedure SINTOMAS_PATOLOGIAS para as 7 primeiras patologias da tabela historico_clinico. ....	81
<b>Figura 122.</b> Call's do procedure SINTOMAS_PATOLOGIAS para as restantes patologias da tabela historico_clinico. ....	81
<b>Figura 123.</b> Percentagem de pacientes que já tiveram COVID-19 por sintoma. ....	81
<b>Figura 124.</b> Percentagem de pacientes que têm obesidade por sintoma. ....	82
<b>Figura 125.</b> Percentagem de pacientes que têm colesterol elevado por sintoma. ....	82
<b>Figura 126.</b> Percentagem de pacientes que têm elevado pressão sanguínea por sintoma. ....	82
<b>Figura 127.</b> Percentagem de pacientes que têm MIGRAINES por sintoma. ....	83
<b>Figura 128.</b> Percentagem de pacientes que têm GERD por sintoma. ....	83
<b>Figura 129.</b> Percentagem de pacientes que têm depressão por sintoma. ....	83
<b>Figura 130.</b> Percentagem de pacientes que têm ansiedade por sintoma. ....	84
<b>Figura 131.</b> Percentagem de pacientes que têm osteoartrite por sintoma. ....	84
<b>Figura 132.</b> Percentagem de pacientes que têm hipotireoidismo por sintoma. ....	84
<b>Figura 133.</b> Percentagem de pacientes que têm hipertensão por sintoma. ....	85
<b>Figura 134.</b> Percentagem de pacientes que têm asma por sintoma. ....	85
<b>Figura 135.</b> Percentagem de pacientes que têm hiperlipidemia por sintoma. ....	85
<b>Figura 136.</b> Percentagem de pacientes que têm uma doença de risco por sintoma. ....	86
<b>Figura 137.</b> Percentagem de pacientes que têm deficiência por sintoma. ....	86
<b>Figura 138.</b> Procedure efetuado para determinar a percentagem de pacientes que tomaram vacina de uma dada marca por sintoma. ....	87
<b>Figura 139.</b> Call's do procedure SINTOMAS_VACINA. ....	88
<b>Figura 140.</b> Percentagem de pacientes que tomaram vacina da Moderna por sintoma. ....	88

<b>Figura 141.</b> Percentagem de pacientes que tomaram vacina da Pfizer por sintoma. ....	88
<b>Figura 142.</b> Percentagem de pacientes que tomaram vacina da Janssen por sintoma. ....	89
<b>Figura 143.</b> Query efetuada para determinar a percentagem de sintomas provocados por um dado lote de vacinas. ....	90
<b>Figura 144.</b> Percentagem de sintomas provocados por um dado lote de vacinas. ....	90
<b>Figura 145.</b> Query efetuada para determinar a percentagem de pessoas que morreram e foram às urgências ou receberam visita do médico. ....	91
<b>Figura 146.</b> Percentagem de pessoas que morreram e foram às urgências ou receberam visita do médico. ....	91
<b>Figura 147.</b> Query efetuada para determinar o número médio de dias em que as pessoas que morreram ficaram hospitalizadas. ....	92
<b>Figura 148.</b> Número médio de dias em que as pessoas que morreram ficaram hospitalizadas. ....	92
<b>Figura 149.</b> Query efetuada para determinar o número médio de dias em que as pessoas que não morreram ficaram hospitalizadas. ....	92
<b>Figura 150.</b> Número médio de dias em que as pessoas que não morreram ficaram hospitalizadas. ....	92
<b>Figura 151.</b> Procedure efetuado para determinar a percentagem de mortes por género. ....	94
<b>Figura 152.</b> Percentagem de mulheres que faleceram. ....	94
<b>Figura 153.</b> Percentagem de homens que faleceram. ....	94
<b>Figura 154.</b> Procedure efetuado para determinar a percentagem de mortes de uma determinada faixa etária dos pacientes. ....	95
<b>Figura 155.</b> Comandos CALL para determinar a percentagem de mortes de uma determinada faixa etária após a toma da vacina contra a COVID-19. ....	95
<b>Figura 156.</b> Percentagem de bebés, dos 0 a 1 ano de idade, que morreram após a toma da vacina contra a COVID-19. ....	96
<b>Figura 157.</b> Percentagem de crianças, de 1 aos 12 anos de idade, que morreram após a toma da vacina contra a COVID-19. ....	96
<b>Figura 158.</b> Percentagem de adolescentes, dos 12 aos 16 anos, que morreram após a toma da vacina contra a COVID-19. ....	96
<b>Figura 159.</b> Percentagem de jovens, dos 16 aos 25 anos, que morreram após a toma da vacina contra a COVID-19. ....	96
<b>Figura 160.</b> Percentagem de adultos, dos 25 aos 65 anos, que morreram após a toma da vacina contra a COVID-19. ....	96
<b>Figura 161.</b> Percentagem de idosos, de 65 ou mais anos, que morreram após a toma da vacina contra a COVID-19. ....	96
<b>Figura 162.</b> Query efetuada para determinar a percentagem de pessoas que morreram com as patologias mais frequentes. ....	98
<b>Figura 163.</b> Percentagem de pessoas que morreram e que tinham as patologias mais frequentes. ....	98
<b>Figura 164.</b> Percentagem de pessoas que morreram com os sintomas mais frequentes e respetivas patologias. ....	100
<b>Figura 165.</b> Percentagem de pessoas que morreram e que tinham os sintomas mais frequentes. ....	100
<b>Figura 166.</b> Query efetuada para determinar o tempo médio que decorreu entre a vacinação de um paciente e a sua morte. ....	101
<b>Figura 167.</b> Tempo médio de dias que decorreu entre a vacinação de um paciente e a sua morte. ....	101



<b>Figura 168.</b> Query efetuada para determinar a percentagem de doentes que morreram por número de doses de vacinas que tomaram. ....	102
<b>Figura 169.</b> Percentagem de pacientes que morreram e tomaram uma dose da vacina contra a COVID-19 e daqueles que morreram e tomaram duas doses. ....	102
<b>Figura 170.</b> Procedure efetuado para determinar a percentagem de mortes associadas a cada marca de vacina. ....	103
<b>Figura 171.</b> Percentagem de pacientes que morreram e tomaram a Moderna. ....	103
<b>Figura 172.</b> Percentagem de pacientes que morreram e tomaram a Pfizer. ....	103
<b>Figura 173.</b> Percentagem de pacientes que morreram e tomaram a Janssen. ....	104
<b>Figura 174.</b> Query efetuada para determinar a percentagem de mortes associada a cada marca e respetivo lote da vacina contra a COVID-19. ....	105
<b>Figura 175.</b> Percentagem de mortes associadas a cada marca e respetivo lote da vacina contra a COVID-19. ....	105
<b>Figura 176.</b> Trigger efetuado para atualizar a tabela paciente sempre que é feito um INSERT INTO na tabela temp_dos_pacientes. ....	106
<b>Figura 177.</b> Trigger efetuado para atualizar a tabela paciente sempre que é feito um DELETE na tabela temp_dos_pacientes. ....	107
<b>Figura 178.</b> Trigger efetuado para atualizar a tabela vacina sempre que é feito um INSERT INTO na tabela temp_das_vacinas. ....	107
<b>Figura 179.</b> Página inicial da interface, onde é feito o login. ....	108
<b>Figura 180.</b> Menu da ficha técnica do paciente a ser preenchido pelo profissional de saúde com as secções existentes representadas. ....	109
<b>Figura 181.</b> Subsecção relativa a informação genérica do paciente, na secção "Informação do paciente". ....	109
<b>Figura 182.</b> Subsecção relativa ao histórico clínico do paciente, na secção "Informação Paciente" (Continuação). ....	110
<b>Figura 183</b> Subsecção relativa ao histórico clínico do paciente, na secção "Informação Paciente". ....	110
<b>Figura 184..</b> Subsecção relativa à hospitalização do paciente. ....	111
<b>Figura 185.</b> Subsecção relativa ao histórico clínico do paciente, na secção "Informação Paciente" (Continuação). ....	111
<b>Figura 186.</b> Secção Vacinação. ....	112
<b>Figura 187.</b> Secção Sintomas pós-vacinação. ....	112

## Índice de Tabelas

<b>Tabela 1.</b> Colunas presentes no dataset “COVID-19 World Vaccine Adverse Reactions” do Kaggle <sup>[3]</sup> . .....	11
<b>Tabela 2.</b> Motivos para a não utilização de colunas do dataset na base de dados. ....	14
<b>Tabela 3.</b> Entidades, entidades relacionamento, respetivos atributos, tipo de dados e quantidade de caracteres que estes armazenam. ....	18
<b>Tabela 4.</b> Entidades, entidades relacionamento e respetivos atributos PK e FK. ....	20
<b>Tabela 5.</b> Designação de cada uma das tabelas temporárias criadas para o import dos dados e colunas contidas em cada uma delas. ....	22
<b>Tabela 6.</b> Nomes das tabelas onde se encontram os dados para povoar as tabelas do modelo lógico. ....	42

# 1. Introdução

## 1.1. Contextualização

Uma base de dados é uma coleção partilhada de dados relacionados logicamente e uma descrição dos mesmos, projetada para atender às necessidades de informação de uma organização. Todos os acessos à base de dados são feitos através do Sistema de Gestão de Base de Dados (SGBD), uma estrutura subjacente de um sistema de informação, que fornece acesso controlado à base de dados, segurança, integridade, simultaneidade e controlo de recuperação, bem como um catálogo acessível ao utilizador <sup>[1]</sup>.

A linguagem SQL é uma linguagem de consulta estruturada que facilita a consulta dos dados organizados na BD e que permite ao utilizador criar a mesma, estruturar as relações, fazer a sua manutenção e realizar consultas, tanto simples como complexas. Esta linguagem tem quatro componentes que são a *Data Definition Language* (DDL), a *Data Manipulation Language* (DML), a *Data Control Language* (DCL) e a *Transaction Control Language* (TCL), das quais se destacam a linguagem de definição de dados (DDL) que possibilita que os utilizadores definam a base de dados e a linguagem de manipulação de dados (DML) que permite inserir, atualizar, excluir e recuperar dados <sup>[1]</sup>.

Atualmente, com a pandemia COVID-19 cada vez mais pessoas estão a ser vacinadas, de forma a tornarem-se imunes ao vírus SARS-CoV-2. Contudo, como estas vacinas foram desenvolvidas num curto período de tempo, ainda se desconhecem muitos dos seus efeitos secundários. Deste modo, é bastante pertinente proceder a uma análise dos efeitos de cada uma das vacinas para a COVID-19.

## 1.2. Motivação e objetivos

Neste trabalho procedeu-se à análise do *dataset* “COVID-19 World Vaccine Adverse Reactions” proveniente do *kaggle*, que contém dados do Sistema de Notificação de Eventos Adversos de Vacinas (VAERS). O VAERS foi criado pela *Food and Drug Administration* (FDA) e pelo Centro para Controle e Prevenção de Doenças (CDC), para receber relatórios sobre eventos adversos que podem estar associados às vacinas. Nenhum medicamento de prescrição ou produto biológico, como uma vacina, está completamente livre de efeitos colaterais. As vacinas protegem muitas pessoas de doenças perigosas, mas podem causar efeitos colaterais, cuja

gravidade é considerável. Assim, o VAERS monitoriza continuamente os relatórios para determinar se alguma vacina ou lote de vacina tem uma taxa de eventos adversos superior à esperada [2].

Deste modo, no *dataset* em causa, estão registados diversos dados relativos a indivíduos americanos, de diferentes faixas etárias, que tomaram uma das vacinas para a COVID-19. De entre os dados analisados destacam-se as patologias associadas a cada um dos indivíduos, os sintomas após a toma e também algumas características das vacinas como o seu fabricante e lote. A partir deste *dataset*, foi possível criar uma base de dados para relacionar toda a informação relativa aos pacientes, à vacinação e seus efeitos [2].

Posto isto, o principal objetivo deste trabalho consiste em analisar os efeitos associados à vacinação contra a COVID-19, nomeadamente os sintomas e a mortalidade, de forma que possam ser tomadas medidas para diminuir as taxas de pacientes com sintomas graves e de mortalidade.

## 2. Análise do *Dataset*

Após escolhido o *dataset*, procedeu-se à sua análise, de forma a perceber quais serão as informações importantes para o desenvolvimento da base de dados. O *dataset* possui 54 colunas, cujas designações e respetivo significado estão registados na Tabela 1.

**Tabela 1.** Colunas presentes no dataset “COVID-19 World Vaccine Adverse Reactions” do Kaggle [3].

Coluna	Significado	Coluna	Significado
<b>VAERS_ID</b>	Número de identificação do VAERS.	<b>HISTORY</b>	Alergias, defeitos congénitos ou condições médicas pré-diagnosticadas.
<b>RECVDATE</b>	Data de recebimento dos dados.	<b>PRIOR_VAX</b>	Informações recebidas antes da vacinação.
<b>STATE</b>	Estado americano a que pertence o indivíduo.	<b>SPLTTYPE</b>	Nº do fabricante.
<b>AGE_YRS</b>	Idade em anos (decimal).	<b>FORM_VERS</b>	Versão do VAERS.

<b>CAGE_YR</b>	Idade em anos (inteiro).	<b>TODAYS_DATE</b>	Data de preenchimento do formulário.
<b>CAGE_MO</b>	Idade em meses (inteiro).	<b>BIRTH_DEFECT</b>	Anomalia congênita ou defeito de nascença (Yes/No).
<b>SEX</b>	Sexo do indivíduo.	<b>OFC_VISIT</b>	Cuidados de saúde fornecidos por medico /clínica (Yes/No).
<b>RPT_DATE</b>	Data de conclusão do formulário.	<b>ER_ED_VISIT</b>	Ida às urgências (Yes/No).
<b>SYMPTOM_TEXT</b>	Texto de relato dos sintomas.	<b>ALLERGIES</b>	Alergias.
<b>DIED</b>	Morte (Yes/No).	<b>SYMPTOM1</b>	Termo 1 de Evento Adverso MedDRA.
<b>DATEDIED</b>	Data do falecimento.	<b>SYMPTOMVERSION1</b>	Versão do dicionário MedDRA nº1.
<b>L_THREAT</b>	<i>Life-Threatening Illness</i> (Yes/No).	<b>SYMPTOM2</b>	Termo 2 de Evento Adverso MedDRA.
<b>ER_VISIT</b>	Ida à emergência ou visita do médico.	<b>SYMPTOMVERSION2</b>	Versão do dicionário MedDRA nº2.
<b>HOSPITAL</b>	Hospitalização (Yes/No).	<b>SYMPTOM3</b>	Termo 3 de Evento Adverso MedDRA.
<b>HOSPDAYS</b>	Nº de dias de hospitalização.	<b>SYMPTOMVERSION3</b>	Versão do dicionário MedDRA nº3.
<b>X_STAY</b>	Hospitalização prolongada (Yes/No).	<b>SYMPTOM4</b>	Termo 4 de Evento Adverso MedDRA.
<b>DISABLE</b>	Deficiência (Yes/No).	<b>SYMPTOMVERSION4</b>	Versão do dicionário MedDRA nº4.
<b>RECOVD</b>	Recuperou totalmente dos sintomas (Yes/No).	<b>SYMPTOM5</b>	Termo 5 de Evento Adverso MedDRA.
<b>VAX_DATE</b>	Data de vacinação.	<b>SYMPTOMVERSION5</b>	Versão do dicionário MedDRA nº5.
<b>ONSET_DATE</b>	Data da ocorrência de eventos adversos.	<b>VAX_TYPE</b>	Tipo de vacina administrada.
<b>NUMDAYS</b>	Nº de dias de persistência dos sintomas.	<b>VAX_MANU</b>	Fabricante da vacina.

<b>LAB_DATA</b>	Dados laboratoriais de diagnóstico.	<b>VAX_LOT</b>	Lote da vacina de cada fabricante.
<b>V_ADMINBY</b>	Vacinas administradas em (PUB = Publico, PVT = Privado, OTH = Other, MIL = Military).	<b>VAX_DOSE_SERIES</b>	Nº de doses de vacina administradas.
<b>V_FUNDBY</b>	Vacinas compradas com fundos ((PUB = Publico, PVT = Privado, OTH = Other, MIL = Military).	<b>VAX_ROUTE</b>	Forma de inoculação da vacina.
<b>OTHER_MEDS</b>	Outras medicações.	<b>VAX_SITE</b>	Local de injeção da vacina.
<b>CUR_ILL</b>	Doença no momento da vacinação.	<b>VAX_NAME</b>	Nome da vacina.

No final da década de 1990, o Conselho Internacional de Harmonização de Requisitos Técnicos para Produtos Farmacêuticos para Uso Humano (ICH) desenvolveu o Dicionário Médico para Atividades Regulatórias (MedDRA), que é um conjunto de termos usados internacionalmente relacionados a condições médicas, medicamentos e dispositivos médicos. A MedDRA está disponível para todos e é utilizada para o registo, documentação e monitorização da segurança de produtos médicos antes e depois de um produto ter sido autorizado para venda. Atualmente, o seu uso crescente em todo o mundo por autoridades regulatórias, empresas farmacêuticas, organizações de pesquisa clínica e profissionais de saúde permite uma melhor proteção global da saúde do paciente <sup>[4]</sup>.

Os sintomas presentes na Tabela 1 fornecem os termos codificados dos eventos adversos utilizando o Dicionário MedDRA. Os codificadores procuraram termos específicos nas colunas SYMPTOM\_TEXT e LAB\_DATA e codificaram-nos em termos MedDRA. Para cada paciente existem até 5 colunas com sintomas organizados segundo as versões do dicionário. Contudo, como existem sintomas iguais em versões diferentes, as versões não serão tidas em conta na elaboração da base de dados, pois esta distinção de sintomas não acrescenta informação relevante para possíveis conclusões <sup>[4]</sup>.

Na base de dados desenvolvida apenas se utilizou a informação relativa às colunas sombreadas a azul na Tabela 1. A restantes não foram utilizadas por diversas razões enumeradas na Tabela 2.

**Tabela 2.** Motivos para a não utilização de colunas do dataset na base de dados.

Coluna	Motivos para não utilização na base de dados
<b>RCVDATE, RPT_DATE, SPLTTYPE, FORM_VERS, TODAYS_DATE</b>	Os dados não acrescentam nenhuma informação relativamente a possíveis efeitos adversos provocados pelas vacinas.
<b>CAGE_YR</b>	Fornece a idade em anos (inteiro), não sendo possível averiguar a idade dos bebés com menos de 1 ano.
<b>CAGE_MO</b>	Fornece a idade em meses o que não é tão fácil de analisar, comparativamente com anos.
<b>SYMPTOM_TEXT, LAB_DATA</b>	A informação foi tratada à posteriori (de acordo com o dicionário MedDRA) pelo VAERS para construir as colunas SYMPTOM (1 a 5) que foram consideradas para o povoamento da base de dados.
<b>DIED</b>	Quando esta coluna apresenta “Yes”, a coluna HOSPDAYS apresenta a respetiva data de falecimento. Assim, basta utilizar a coluna DATEDIED, pois quando esta não é nula, é porque o indivíduo faleceu.
<b>ER_VISIT, X_STAY, V_FUNDBY</b>	100% dos valores eram nulos.
<b>HOSPITAL</b>	Quando esta coluna apresenta “Yes”, a coluna HOSPDAYS apresenta os respetivos dias de internamento. Assim, basta utilizar a coluna HOSPDAYS, pois quando esta não é nula, é porque o indivíduo esteve hospitalizado.
<b>OTHER_MEDS</b>	39% dos valores eram nulos, 5% eram “none” e os restantes eram medicamentos muitos variados e com nomes diferentes, assim como diferentes suplementos vitamínicos, sendo muito difícil estabelecer algum tipo de comparação.
<b>PRIOR_VAX</b>	96% dos valores eram nulos e os restantes 4% eram “OTH”.
<b>ALLERGIES</b>	44% dos valores eram nulos, 9% eram “none” e os restantes eram alergias muito diversas a diferentes alimentos, medicamentos, animais, entre outros, sendo muito difícil estabelecer algum tipo de comparação.

<b>SYMPTOMVERSION1, SYMPTOMVERSION2, SYMPTOMVERSION3, SYMPTOMVERSION4, SYMPTOMVERSION5</b>	Os dicionários da MedDRA apresentam diferentes versões para os diferentes termos, contudo, esta informação não é relevante para avaliar efeitos das vacinas.
<b>VAX_TYPE</b>	Como o objetivo do trabalho é analisar os efeitos adversos apenas de vacinas da COVID-19, não se considerou esta coluna, pois todos os pacientes cujo VAX_TYPE não era COVID-19 foram descartados através de um pré-povoamento descrito na secção 5.2.8.
<b>VAX_NAME</b>	Uma vez que o VAX_MANU já contém o nome da marca da vacina, que coincidia sempre com o nome da vacina, esta coluna foi descartada, pela sua ambiguidade.

### 3. Modelo Conceptual

Após ter sido analisado o *dataset* e terem-se definido os dados a utilizar na criação da base de dados, foi possível construir o modelo conceptual, utilizando o software TerraER. Desta forma, utilizou-se um Diagrama Entidades-Relacionamentos (ou Diagrama ER), o qual é constituído pelas entidades, pelos atributos e pelas relações que existem entre si. No que toca à notação Peter Chen, as entidades são representadas por retângulos verdes e os atributos por elipses a azul, estando os atributos-chave sublinhados e a negrito e os atributos derivados a traço interrompido. Os relacionamentos são representados por losangos amarelos e as entidades-relacionamento por losangos amarelos contidos em retângulos verdes. A cardinalidade é importante para ajudar a definir o relacionamento, pois esta define o número de ocorrências num relacionamento. Entre entidades podem estabelecer-se diferentes relações: relações de 1 para 1 (1:1), relações de 1 para muitos (1:N) e relações de muitos para muitos (N:M) <sup>[5]</sup>.

Na Figura 1, está representado o modelo conceptual da BD a desenvolver. Este modelo é constituído por uma relação (1:N) entre as entidades fornecedor e vacina. Como a vacina tem obrigatoriamente de ter um fornecedor, então verifica-se a presença de uma *identify relationship*, representada pela linha dupla. As restantes relações são N:M e de carácter opcional.



Relativamente às entidades, existem 6 entidades (fornecedor, paciente, vacina, historico\_clinico, hospitalizacao, sintomas) e 5 entidades relacionamento (histórico\_clinico afeta paciente, hospitalizacao trata paciente, sintomas afetam paciente, vacina causa sintomas e vacinar). Todos os atributos foram retirados de colunas da Tabela 1, exceto os ids das tabelas fornecedor, vacina, hospitalizacao, sintomas e historico\_clinico, o atributo designacao da tabela sintomas e o atributo designacao da tabela historico\_clinico. Relativamente à tabela sintomas, cada designacao corresponde a um sintoma diferente, proveniente das 5 colunas de SYMPTOM. Quanto à tabela historico\_clinico, cada atributo designacao corresponde a uma doença diferente. Para a construção desta tabela, foram consideradas as colunas DISABLE e L\_THREAT da Tabela 1, assim como as doenças mais frequentes e mais graves das colunas CUR\_ILL, BIRTH\_DEFECT e HISTORY.

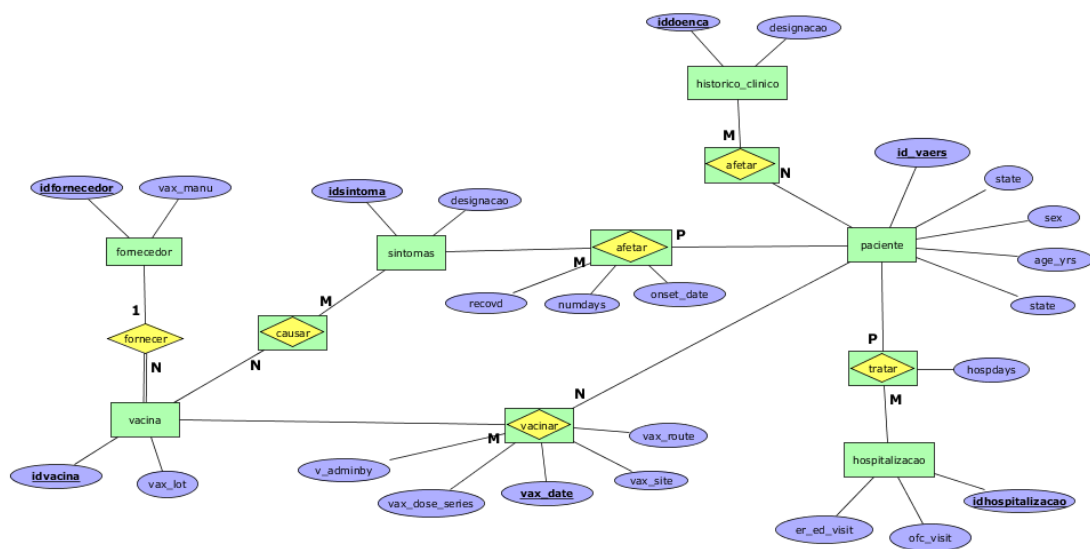


Figura 1. Modelo conceitual.

## 4. Modelo lógico

O modelo relacional foi proposto por EF Codd em 1970, representando a segunda geração de SGBD's. Os objetivos do modelo relacional são permitir um elevado grau de independência de dados, fornecer motivos para lidar com a semântica dos dados, problemas de consistência e redundância (relações normalizadas) e permitir a expansão de linguagens de manipulação de dados orientadas para conjuntos [6].

Posto isto, define-se como base de dados relacional uma coleção de relações normalizadas com distintos nomes para as relações [6]. O modelo lógico é construído com base no modelo conceptual desenvolvido, sendo este uma representação simples e intuitiva da base de dados que se pretende construir, tal como se observa na Figura 2.

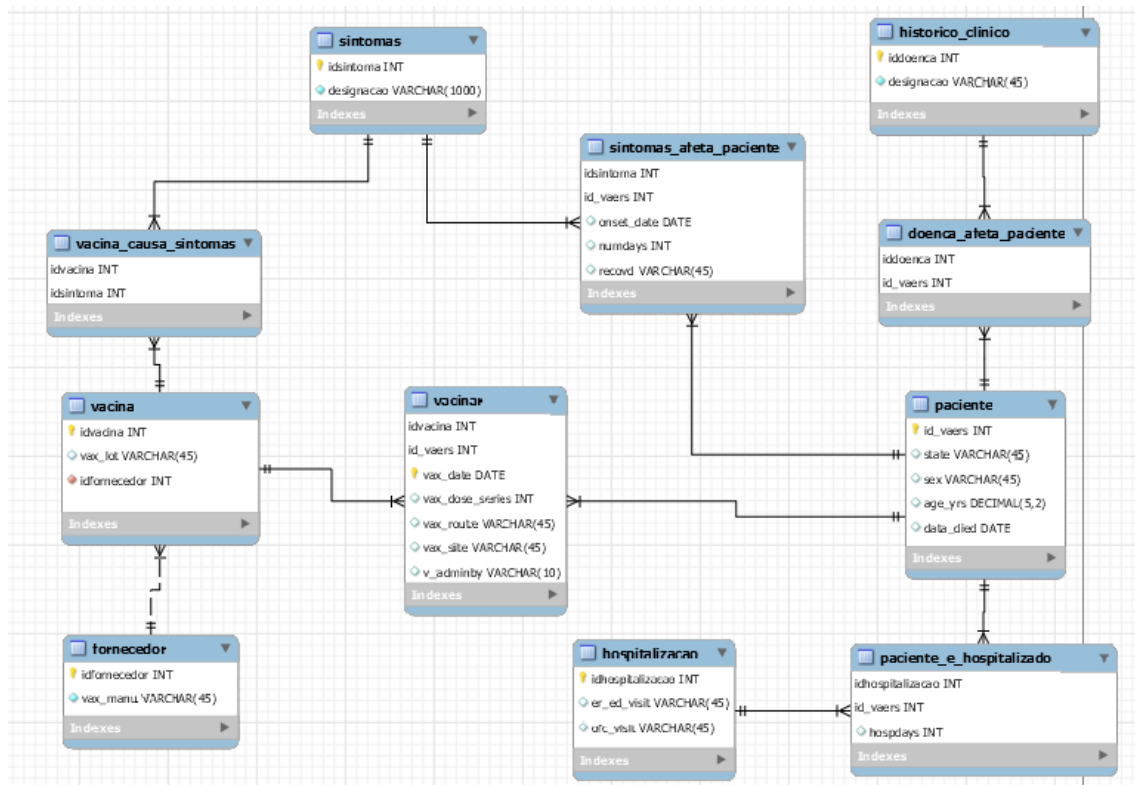


Figura 2. Modelo lógico.

#### 4.1. Entidades

As entidades representam uma classe de objetos sobre os quais se pretende guardar informação na base de dados [6]. Estas encontram-se sob a forma de tabelas no modelo lógico, tal como se pode observar pelas formas retangulares da Figura 2, correspondendo as colunas a cada um dos seus atributos. Na Figura 2 enumeram-se as seguintes entidades, incluindo as entidades-relacionamento que advém de relacionamentos com cardinalidade N:N no modelo conceptual[5]:

- Entidades:
  - ✓ fornecedor;
  - ✓ vacina;
  - ✓ sintomas;
  - ✓ historico\_clinico;

- ✓ paciente;
  - ✓ hospitalizacao.
- Entidades relacionamento:
    - ✓ vacinar;
    - ✓ vacina\_causa\_sintomas;
    - ✓ sintomas\_afeta\_paciente;
    - ✓ doenca\_afeta\_paciente;
    - ✓ paciente\_e\_hospitalizado.

## 4.2. Atributos

Os atributos correspondem a itens informativos específicos de cada entidade, que se encontram armazenados nas colunas das tabelas. Existem diferentes tipos de atributos e a estes encontra-se associado um determinado tipo de dados, como por exemplo, sequência de caracteres (VARCHAR), numérico inteiro (INT), numérico decimal (DECIMAL) e data (DATE) <sup>[6]</sup>. É de referir que certas características associadas a cada atributo são divulgadas por um código simbólico no modelo lógico, tal como se pode observar na Figura 2, nomeadamente o facto do atributo não assumir valor nulo, representado por um losango azul anterior à designação do atributo. Na Tabela 3 estão identificados os atributos de todas as entidades descritas anteriormente, bem como o tipo de dados e quantidade de caracteres que armazenam.

**Tabela 3.** Entidades, entidades relacionamento, respetivos atributos, tipo de dados e quantidade de caracteres que estes armazenam.

Entidades e entidades relacionamento	Atributos	
	Designação	Tipo de dados
<b>fornecedor</b>	idfornecedor	INT
	vax_manu	VARCHAR (45)
<b>vacina</b>	idvacina	INT
	vax_lot	VARCHAR (45)
	idfornecedor	INT
<b>vacina_causa_sintomas</b>	idvacina	INT

	idsintoma	INT
<b>vacinar</b>	idvacina	INT
	id_vaers	INT
	vax_date	DATE
	vax_dose_series	INT
	vax_route	VARCHAR (45)
	vax_site	VARCHAR (45)
	v_adminby	VARCHAR (10)
<b>sintomas</b>	idsintoma	INT
	designacao	VARCHAR (1000)
<b>sintomas_afeta_paciente</b>	idsintoma	INT
	id_vaers	INT
	onset_date	DATE
	numdays	INT
	recovd	VARCHAR (45)
<b>historico_clinico</b>	iddoenca	INT
	designacao	VARCHAR (45)
<b>doenca_afeta_paciente</b>	iddoenca	INT
	id_vaers	INT
<b>paciente</b>	id_vaers	INT
	state	VARCHAR (45)
	sex	VARCHAR (45)
	age_yrs	DECIMAL (3,2)
	data_died	DATE
<b>paciente_e_hospitalizado</b>	idhospitalizacao	INT
	id_vaers	INT
	hospdays	INT
<b>hospitalizacao</b>	idhospitalizacao	INT
	er_ed_visit	VARCHAR (45)
	ofc_visit	VARCHAR (45)

### 4.3. Chave-primária e chave-estrangeira

Nas tabelas, existem atributos cuja finalidade é identificar inequivocamente um tuplo, permitindo um simples e fácil acesso à informação lá armazenada. Estes designam-se por chaves candidatas, sendo que nenhum tuplo pode ser reduzido por forma a perder este cariz de unicidade. Desta forma, define-se como chave-primária (PK, da expressão anglo-saxónica Primary Key) todos o(s) atributo(s) que são chaves candidatas não nulas que identificam em regime de exclusividade uma só linha da tabela, representados por uma chave amarela no modelo relacional (Figura 2). Existem também as chaves-estrangeiras, (FK, do inglês Foreign Key) que se definem como o(s) atributo(s) presentes numa tabela que são chaves-primárias de outra relação. Quando estas são obrigatoriamente não nulas são representadas por um losango vermelho e quando não têm de ser obrigatoriamente não nulas, são representadas por um losango com bordas vermelhas e fundo branco. É de referir que podem existir atributos numa mesma tabela que sejam simultaneamente chaves-primárias e chaves-estrangeiras – não possuem símbolo identificativo no modelo relacional <sup>[6]</sup>. Na Tabela 4 apresentam-se as chaves-primárias e estrangeiras de cada entidade.

**Tabela 4.** Entidades, entidades relacionamento e respetivos atributos PK e FK.

Entidades e entidades relacionamento	Atributos	
	PK	FK
<b>fornecedor</b>	idfornecedor	---
<b>vacina</b>	idvacina	idfornecedor
<b>vacina_causa_sintomas</b>	idvacina	idvacina
	idsintoma	idsintoma
<b>vacinar</b>	idvacina	idvacina
	id_vaers	id_vaers
	vax_date	---
<b>sintomas</b>	idsintoma	---
<b>sintomas_afeta_paciente</b>	idsintoma	idsintoma
	id_vaers	id_vaers
<b>historico_clinico</b>	iddoenca	---
<b>doença_afeta_paciente</b>	iddoenca	iddoenca

	id_vaers	id_vaers
<b>paciente</b>	id_vaers	---
<b>paciente_e_hospitalizado</b>	idhospitalizacao	idhospitalizacao
	id_vaers	id_vaers
<b>hospitalizacao</b>	idhospitalizacao	---

Posto isto, é perceptível que a construção das tabelas no modelo relacional não é aleatória, pois as chaves-estrangeiras demonstram dependência de informação entre relações. Assim, primeiramente são geradas as tabelas com apenas chaves-primárias, seguindo-se as com menor número de chaves-estrangeiras e, posteriormente, as de maior número, finalizando-se com o preenchimento das tabelas que possuam chaves simultaneamente PK e FK.

## 5. Modelo físico

No SGBD MySQL, utilizou-se o comando FORWARD ENGINEER para gerar o modelo físico a partir do modelo lógico representado na Figura 2.

### 5.1. Pré-povoamento

Com o modelo físico criado, através do modelo lógico anteriormente exposto, procedeu-se ao povoamento das tabelas. Contudo, devido à complexidade do povoamento, dividiu-se este processo em 2 etapas. Na 1ª etapa procedeu-se ao pré-povoamento que consiste numa preparação dos dados presentes nas tabelas importadas através do *Table Data Import Wizard*.

Quando se procedeu ao *import* das tabelas, utilizando o comando *Table Data Import Wizard*, foi necessário importar todas as colunas como sendo TEXT, à exceção das colunas com VAERS\_ID que foram importadas como INT. Isto foi necessário, uma vez que quando se tentou submeter os dados com o *datatype* correspondente, todas as linhas que possuíam alguma célula vazia no Excel não eram importadas. Ao assumir todos os dados como TEXT, cada vez que aparecia uma célula vazia, estas eram assumidas como um TEXT do tipo "" e por isso foi possível importar todas as linhas. As colunas do tipo VAERS\_ID não apresentaram esse problema, pois não existia nenhuma célula vazia nestas colunas.

O *dataset* proveniente do *kaggle* possuía 3 documentos em Excel, contudo, não foi possível fazer diretamente o *import* de todos os dados existentes em cada um dos Excel, porque, como eram muitos dados, o *mySQL* desligava-se. Deste modo, procedeu-se à divisão de todas as colunas em vários documentos de Excel diferentes, conforme representado na Tabela 5.

É também importante referir que nem todos os dados que estão em cada uma das tabelas serão povoados numa só tabela. Inicialmente esse era o objetivo, contudo, à medida que se foi desenvolvendo o trabalho, foram surgindo novas ideias de melhoria da base de dados e por isso existem dados na mesma tabela que serão povoados em tabelas distintas.

**Tabela 5.** Designação de cada uma das tabelas temporárias criadas para o import dos dados e colunas contidas em cada uma delas.

Nome do documento Excel	Colunas contidas no documento
<b>temp_doenca</b>	VAERS_ID, RECOVD, DISABLE
<b>temp_fornecedor</b>	VAERS_ID, VAX_MANU
<b>temp_historico</b>	VAERS_ID, Obesity, High cholesterol, High Blood Pressure, Migraines, GERD, Depression, Anxiety, Osteoarthritis, Hypothyroidism, Hypertension, Asthma, Hyperlipidemia, COVID-19, L_THREAT V_ADMINBY
<b>temp_hospital</b>	VAERS_ID, HOSPDAYS, OFC_VISIT, ER_ED_VISIT
<b>temp_paciente</b>	VAERS_ID, STATE, AGE_YRS, SEX, DATEDIED
<b>temp_sintomas</b>	VAERS_ID, SYMPTOM1, SYMPTOM2, SYMPTOM3, SYMPTOM4, SYMPTOM5
<b>temp_sintomas2</b>	VAERS_ID, ONSET_DATE, NMDAYS
<b>temp_vacina</b>	VAX_TYPE, VAX_LOT, VAX_DOSE_SERIES, VAX_ROUTE, VAX_SITE
<b>temp_vacinar</b>	VAERS_ID, VAX_DATE

Todas as tabelas contêm o VAERS\_ID, uma vez que sem esta coluna é impossível saber a que paciente correspondem aqueles dados, e consequentemente, não se consegue povoar as

tabelas no modelo físico. O *import* e tratamento de cada uma dessas tabelas foi realizado no *Schema* trabalho\_covid.

De seguida será analisada de forma detalhada o pré-povoamento de cada uma das tabelas presentes na Tabela 5.

#### 5.1.1. Tabela temp\_doenca

Na Figura 3 estão representadas todas as *query's* efetuadas na tabela **temp\_doenca**. Na zona 1 converteram-se as *string's* vazias em valores nulos, de forma a facilitar a análise das *query's* a realizar nas secções 5.2 e 6. Para tal, utilizou-se o comando UPDATE, que atualizou os valores "" da **temp\_doenca** para NULL. Na zona 2 criou-se uma tabela **temp\_da\_doenca** cujos VAERS\_ID existiam simultaneamente na tabela **temp\_doenca** e na tabela **temp\_vacinar**. Esta operação foi necessária, uma vez que na tabela vacinar do modelo lógico, a PK é composta pelo id\_vaers, pelo idvacina e pela vax\_date, e assim sendo, não faz sentido povoar o modelo com pacientes que não existam na tabela **temp\_vacinar**, pois estes não terão vax\_date, e consequentemente não será possível analisar os efeitos provocados pela vacinação em pacientes que não foram vacinados. Para realizar esta *query*, recorreu-se ao comando CREATE TABLE, tendo sido realizado à posteriori o INSERT INTO na tabela **temp\_da\_doenca** das linhas da tabela **temp\_doenca** cujos VAERS\_ID da tabela **temp\_doenca** eram iguais aos VAERS\_ID da tabela **temp\_vacinar**.

Na zona 3 (Figura 4) verificou-se se existia mais do que 1 linha por paciente, uma vez que não faz sentido existir 1 paciente que simultaneamente tenha uma deficiência e não tenha uma deficiência (DISABLE = "Y" e DISABLE = "") ou então que tenha e não tenha simultaneamente recuperado de todos os sintomas (RECOVD = "Y" e RECOVD = ""). Para tal, utilizou-se a função de agregação COUNT, em que se o COUNT fosse superior a 1, significaria que existia mais do que 1 linha por paciente, o que não se verificou. Utilizou-se também o GROUP BY para agrupar linhas que possuíam o mesmo VAERS\_ID.

Por último, na zona 4 converteu-se o *datatype* da coluna RECOVD em VARCHAR (45), para ficar em conformidade com o *datatype* dessa coluna no modelo lógico. Para tal recorreu-se ao comando ALTER TABLE MODIFY COLUMN e verificou-se se a operação tinha sido corretamente efetuada (Figura 5). É de realçar que não se alterou o *datatype* da coluna DISABLE, uma vez que no modelo lógico não existe nenhuma coluna com esta designação. A informação desta coluna foi incorporada na tabela **doenca\_afeta\_paciente** conforme será explicado na secção 5.2.6.

Na Figura 6 está representada parte da informação presente na tabela final **temp\_da\_doenca**, obtida através da *query* SELECT \* FROM **temp\_da\_doenca**.



```
-- Tabela temp_doenca
-- 1. Converter strings vazias em valores nulos
UPDATE temp_doenca SET RECOVD = NULL WHERE RECOVD = '';
UPDATE temp_doenca SET DISABLE = NULL WHERE DISABLE = '';

-- 2. Criar tabela temp_da_doenca cujos pacientes são iguais aos do temp_vacinar
CREATE TABLE temp_da_doenca (VAERS_ID INT, RECOVD TEXT, DISABLE TEXT);

INSERT INTO temp_da_doenca (VAERS_ID, RECOVD, DISABLE)
SELECT d.VAERS_ID, d.RECOVD, d.DISABLE
FROM temp_vacinar vr, temp_doenca d
WHERE vr.VAERS_ID = d.VAERS_ID;

-- 3. Verificar se existe mais do que 1 linha por paciente (não existe)
SELECT * FROM temp_da_doenca GROUP BY VAERS_ID HAVING COUNT(VAERS_ID)>1;

-- 4. Converter RECOVD em VARCHAR(45)
ALTER TABLE temp_da_doenca MODIFY COLUMN RECOVD VARCHAR(45);
SELECT COLUMN_NAME, data_type FROM information_schema.COLUMNS WHERE table_schema = 'trabalho_covid'
AND TABLE_NAME = 'temp_da_doenca';
```

Figura 3. Query's efetuadas na tabela temp\_doenca.

VAERS_ID	RECOVD	DISABLE

Figura 4. Resultado obtido na query efetuada na zona 3 na tabela temp\_da\_doenca.

COLUMN_NAME	DATA_TYPE
DISABLE	text
RECOVD	varchar
VAERS_ID	int

Figura 5. Resultado obtido na query efetuada na zona 4 na tabela temp\_da\_doenca.

VAERS_ID	RECOVD	DISABLE
916600	Y	NULL
916601	Y	NULL
916602	NULL	NULL
916603	Y	NULL
916604	N	NULL
916606	Y	NULL
916607	Y	NULL
916608	NULL	NULL
916609	N	NULL
916610	N	NULL
916611	N	NULL
916613	NULL	NULL
916614	Y	NULL
916615	N	NULL
916617	Y	NULL

Figura 6. Parte da informação presente na tabela temp\_da\_doenca.

### 5.1.2. Tabela temp\_fornecedor

Na Figura 7 estão representadas todas as *query's* efetuadas na tabela **temp\_fornecedor**. Na zona 1 converteram-se as *string's* vazias em valores nulos, pelos motivos anteriormente explicados na secção 5.1.1. Para tal, utilizou-se o comando UPDATE, que atualizou os valores "" da **temp\_fornecedor** para NULL. Na zona 2 criou-se uma tabela **temp\_do\_fornecedor** cujos VAERS\_ID existiam simultaneamente na tabela **temp\_fornecedor** e na tabela **temp\_vacinar**. Esta operação foi necessária pelos motivos anteriormente explicados na secção 5.1.1. Para realizar esta *query*, recorreu-se ao comando CREATE TABLE, tendo sido realizado à posteriori o INSERT INTO na tabela **temp\_do\_fornecedor** das linhas da tabela **temp\_fornecedor** cujos VAERS\_ID da tabela **temp\_fornecedor** eram iguais aos VAERS\_ID da tabela **temp\_vacinar**.

Por último, na zona 3 converteu-se o *datatype* da coluna VAX\_MANU em VARCHAR (45), para ficar em conformidade com o *datatype* dessa coluna no modelo lógico. Para tal recorreu-se ao comando ALTER TABLE MODIFY COLUMN e verificou-se se a operação tinha sido corretamente efetuada (Figura 8).

Na Figura 9 está representada parte da informação presente na tabela final **temp\_do\_fornecedor**, obtida através da *query* SELECT \* FROM temp\_do\_fornecedor.

```
-- Tabela temp_fornecedor
-- 1. Converter strings vazias em valores nulos
UPDATE temp_fornecedor SET VAX_MANU = NULL WHERE VAX_MANU = '';

-- 2. Criar tabela temp_do_fornecedor cujos pacientes são iguais aos do temp_vacinar
CREATE TABLE temp_do_fornecedor (VAERS_ID INT, VAX_MANU TEXT);

INSERT INTO temp_do_fornecedor (VAERS_ID, VAX_MANU)
SELECT f.VAERS_ID, f.VAX_MANU
FROM temp_vacinar vr, temp_fornecedor f
WHERE vr.VAERS_ID = f.VAERS_ID;

-- 3. Converter VAX_MANU em VARCHAR(45)
ALTER TABLE temp_do_fornecedor MODIFY COLUMN VAX_MANU VARCHAR(45);
SELECT COLUMN_NAME, data_type FROM information_schema.COLUMNS WHERE table_schema = 'trabalho_covid'
AND TABLE_NAME = 'temp_do_fornecedor';
```

Figura 7. Query's efetuadas na tabela temp\_fornecedor.

COLUMN_NAME	DATA_TYPE
VAERS_ID	int
VAX_MANU	varchar

**Figura 8.** Resultado obtido na query efetuada na zona 3 na tabela temp\_do\_fornecedor.

VAERS_ID	VAX_MANU
916600	MODERA
916601	MODERA
916602	PFIZER\BIOTECH
916603	MODERA
916604	MODERA
916606	MODERA
916608	MODERA
916609	MODERA
916610	MODERA
916611	MODERA
916613	MODERA
916614	MODERA
916615	MODERA
916618	MODERA

**Figura 9.** Parte da informação presente na tabela temp\_do\_fornecedor.

### 5.1.3. Tabela temp\_historico

A tabela **temp\_historico** contém 16 colunas, o VAERS\_ID comum a todas as tabelas temp, 14 patologias e o V\_ADMINBY que apesar de não estar relacionado com o histórico clínico dos pacientes, está nesta tabela pois, tendo sido adicionado à posteriori, foi mais conveniente adicioná-lo nesta tabela, em que os VAERS\_ID são os mesmos e estão na ordem correta pelo qual os dados aparecem na coluna V\_ADMINBY.

Para a construção da tabela **temp\_historico**, foi considerada a coluna L\_THREAT da Tabela 1, assim como as doenças mais frequentes e mais graves das colunas CUR\_ILL, BIRTH\_DEFECT e HISTORY. Analisando essas 3 colunas presentes no *dataset*, verificou-se para cada uma delas quais os pacientes que apresentavam determinada patologia. Para isso, foram utilizadas inúmeras ferramentas do Excel. Posteriormente, procedeu-se a uma seleção dos pacientes, pelo VAERS\_ID, que continham essa mesma doença. Para isso, a ferramenta Ordenar e Filtrar foi essencial, em que com a opção 'Contém' (nome da patologia) foi possível isolar todas as células de uma coluna que continham uma determinada patologia. Após isso, todas as linhas já filtradas foram substituídas por 'Sim', e novamente recorrendo à filtração, por coluna, isolaram-se as

células com a opção “É diferente de...”, para substituir as restantes células que eram diferentes de “Sim”, ou seja, que não continham a patologia, por “Não”. Inicialmente, as respostas lógicas eram em português, mas como o *dataset* apresentava células que continham “Yes” noutras colunas, optou-se por substituir o “Sim” por “Yes”, assim como o “Não” foi substituído por “No”.

Nas Figuras 10 e 11 estão representadas todas as *query’s* efetuadas na tabela **temp\_historico**. Na zona 1 converteram-se as *string’s* vazias em valores nulos, pelos motivos anteriormente explicados na secção 5.1.1. Para tal, utilizou-se o comando UPDATE, que atualizou os valores “” da **temp\_historico** para NULL. Na zona 2 criou-se uma tabela **temp\_do\_historico** cujos VAERS\_ID existiam simultaneamente na tabela **temp\_historico** e na tabela **temp\_vacinar**. Esta operação foi necessária, pelos motivos anteriormente explicados na secção 5.1.1. Para realizar esta *query*, recorreu-se ao comando CREATE TABLE, tendo sido realizado à posteriori o INSERT INTO na tabela **temp\_do\_historico** das linhas da tabela **temp\_historico** cujos VAERS\_ID da tabela **temp\_historico** eram iguais aos VAERS\_ID da tabela **temp\_vacinar**.

Por último, na zona 3 converteu-se o *datatype* da coluna V\_ADMINBY em VARCHAR (10), para ficar em conformidade com o *datatype* dessa coluna no modelo lógico. Para tal recorreu-se ao comando ALTER TABLE MODIFY COLUMN e verificou-se se a operação tinha sido corretamente efetuada (Figura 12). É de realçar que não se alterou o *datatype* das restantes colunas, uma vez que no modelo lógico não existe nenhuma coluna com estas designações. A informação destas colunas foi incorporada na tabela **doenca\_afeta\_paciente** conforme será explicado na secção 5.2.6.

Nas Figura 13 e 14 está representada parte da informação presente na tabela final **temp\_do\_historico**, obtida através da *query* SELECT \* FROM **temp\_do\_historico**.

```
-- Tabela temp_historico
-- 1. Converter strings vazias ou com UNK ou OTH em valores nulos
UPDATE temp_historico SET L_THREAT = NULL WHERE L_THREAT = '';
UPDATE temp_historico SET V_ADMINBY = NULL WHERE V_ADMINBY = '' OR V_ADMINBY = 'UNK' OR V_ADMINBY = 'OTH';

-- 2. Criar tabela temp_do_historico cujos pacientes são iguais aos do temp_vacinar
CREATE TABLE temp_do_historico (VAERS_ID INT, OBESITY TEXT, HIGH_CHOLESTEROL TEXT, HIGH_BLOOD_PRESSURE TEXT,
MIGRAINES TEXT, GERD TEXT, DEPRESSION TEXT, ANXIETY TEXT,
OSTEOARTHRITIS TEXT, HYPOTHYROIDISM TEXT, HYPERTENSION TEXT, ASTHMA TEXT, HYPERLIPIDEMIA TEXT, COVID19 TEXT, L_THREAT TEXT,
V_ADMINBY TEXT);

INSERT INTO temp_do_historico (VAERS_ID, OBESITY ,HIGH_CHOLESTEROL, HIGH_BLOOD_PRESSURE, MIGRAINES , GERD , DEPRESSION ,
ANXIETY, OSTEOARTHRITIS, HYPOTHYROIDISM, HYPERTENSION , ASTHMA , HYPERLIPIDEMIA, COVID19, L_THREAT, V_ADMINBY)
SELECT h.VAERS_ID, h.OBESITY, h.`HIGH CHOLESTEROL`, h.`HIGH BLOOD PRESSURE`, h.MIGRAINES, h.GERD, h.DEPRESSION, h.ANXIETY,
h.OSTEOARTHRITIS, h.HYPOTHYROIDISM, h.HYPERTENSION, h.ASTHMA, h.HYPERLIPIDEMIA, h.`Covid 19`, h.L_THREAT, h.V_ADMINBY
FROM temp_vacinar vr, temp_historico h
WHERE vr.VAERS_ID = h.VAERS_ID;
```

Figura 10. *Query’s* efetuadas na tabela **temp\_historico** (parte 1).

```
-- 3. Converter V_ADMINBY em VARCHAR(10)
ALTER TABLE temp_do_historico MODIFY COLUMN V_ADMINBY VARCHAR(10);
SELECT COLUMN_NAME, data_type FROM information_schema.COLUMNS WHERE table_schema = 'trabalho_covid'
AND TABLE_NAME = 'temp_do_historico';
```

**Figura 11.** Query's efetuadas na tabela temp\_historico (parte 2).

COLUMN_NAME	DATA_TYPE
COVID19	text
DEPRESSION	text
GERD	text
HIGH_BLOOD_PRESSURE	text
HIGH_CHOLESTEROL	text
HYPERLIPIDEMIA	text
HYPERTENSION	text
HYPOTHYROIDISM	text
L_THREAT	text
MIGRAINES	text
OBESITY	text
OSTEOARTHRITIS	text
V_ADMINBY	varchar
VAERS_ID	int

**Figura 12.** Resultado obtido na query efetuada na zona 3 na tabela temp\_do\_historico.

VAERS_ID	OBESITY	HIGH_CHOLESTEROL	HIGH_BLOOD_PRESSURE	MIGRAINES	GERD	DEPRESSION	ANXIETY	OSTEOARTHRITIS	HYPOTHYROIDISM
916600	No	No	No	No	No	No	No	No	No
916601	No	No	No	No	No	No	No	No	No
916602	No	No	No	No	No	No	No	No	No
916603	No	No	No	No	No	No	No	Yes	No
916604	No	No	No	No	No	No	No	No	No
916606	No	No	No	No	No	No	No	No	No
916607	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
916608	No	No	No	No	No	No	No	No	No
916609	No	No	No	No	No	Yes	No	No	No
916610	No	No	No	No	No	No	No	No	No
916611	Yes	No	No	No	No	No	Yes	No	No
916613	No	No	No	No	No	No	No	No	No
916614	No	No	No	No	No	No	No	No	No

**Figura 13.** Parte da informação presente na tabela temp\_do\_historico (parte 1).

HYROIDISM	HYPERTENSION	ASTHMA	HYPERLIPIDEMIA	COVID19	L_THREAT	V_ADMINBY
	No	No	No	No	NULL	PVT
	No	No	No	No	NULL	SEN
	No	No	No	No	NULL	SEN
	No	No	No	No	NULL	WRK
	No	No	No	No	NULL	PUB
	No	No	No	No	NULL	PVT
	Yes	Yes	Yes	Yes	NULL	PUB
	No	No	No	No	NULL	NULL
	Yes	No	No	No	NULL	PUB
	No	No	No	No	NULL	PVT
	No	No	No	No	NULL	NULL
	No	No	No	No	NULL	NULL
	No	No	No	No	NULL	PVT
					NULL	

**Figura 14.** Parte da informação presente na tabela temp\_do\_historico (parte 2).

#### 5.1.4. Tabela temp\_hospital

Nas Figuras 15 e 16 estão representadas todas as *query's* efetuadas na tabela **temp\_hospital**. Na zona 1 converteram-se as *string's* vazias em valores nulos, pelos motivos anteriormente explicados na secção 5.1.1. Para tal, utilizou-se o comando UPDATE, que atualizou os valores "" da **temp\_hospital** para NULL. Na zona 2 criou-se uma tabela **temp\_do\_hospital** cujos VAERS\_ID existiam simultaneamente na tabela **temp\_hospital** e na tabela **temp\_vacinar**. Esta operação foi necessária, pelos motivos anteriormente explicados na secção 5.1.1. Para realizar esta *query*, recorreu-se ao comando CREATE TABLE, tendo sido realizado à posteriori o INSERT INTO na tabela **temp\_do\_hospital** das linhas da tabela **temp\_hospital** cujos VAERS\_ID da tabela **temp\_hospital** eram iguais aos VAERS\_ID da tabela **temp\_vacinar**.

Na zona 3 (Figura 17) verificou-se se existia mais do que 1 linha por paciente, uma vez que não faz sentido existir 1 paciente que simultaneamente tenha uma visitado e não visitado o médico (OFC\_VISIT = "Y" e OFC\_VISIT = "") ou então que tenha ido e não ido às urgências (ER\_ED\_VISIT = "Y" e ER\_ED\_VISIT = ""). Para tal, utilizou-se a função de agregação COUNT, em que se o COUNT fosse superior a 1, significaria que existia mais do que 1 linha por paciente, o que não se verificou. Utilizou-se também o GROUP BY para agrupar linhas que possuíam o mesmo VAERS\_ID.

Na zona 4 converteu-se o *datatype* das colunas ER\_ED\_VISIT e OFC\_VISIT em VARCHAR (45), para ficar em conformidade com o *datatype* dessa coluna no modelo lógico. Para tal recorreu-se ao comando ALTER TABLE MODIFY COLUMN. Por último, na zona 5 converteu-se o *datatype* da coluna HOSPDAYS em INT, para ficar em conformidade com o *datatype* dessa coluna no modelo lógico. Para tal recorreu-se ao comando ALTER TABLE MODIFY COLUMN e verificou-se se a operação tinha sido corretamente efetuada (Figura 18).

Na Figura 19 está representada parte da informação presente na tabela final **temp\_do\_hospital**, obtida através da *query* SELECT \* FROM **temp\_do\_hospital**.

```
-- Tabela temp_hospital
-- 1. Converter strings vazias em valores nulos
UPDATE temp_hospital SET HOSPDAYS = NULL WHERE HOSPDAYS = '';
UPDATE temp_hospital SET OFC_VISIT = NULL WHERE OFC_VISIT = '';
UPDATE temp_hospital SET ER_ED_VISIT = NULL WHERE ER_ED_VISIT = '';

-- 2. Criar tabela temp_do_hospital cujos pacientes são iguais aos do temp_vacinar
CREATE TABLE temp_do_hospital (VAERS_ID INT, HOSPDAYS INT, OFC_VISIT TEXT, ER_ED_VISIT TEXT);

INSERT INTO temp_do_hospital (VAERS_ID, HOSPDAYS, OFC_VISIT, ER_ED_VISIT)
SELECT h.VAERS_ID, h.HOSPDAYS, h.OFC_VISIT, h.ER_ED_VISIT
FROM temp_vacinar vr, temp_hospital h
WHERE vr.VAERS_ID = h.VAERS_ID;

-- 3. Verificar se existe mais do que 1 linha por paciente (não existe)
SELECT * FROM temp_do_hospital GROUP BY VAERS_ID HAVING COUNT(VAERS_ID)>1;
```

Figura 15. Query's efetuadas na tabela temp\_hospital (parte 1).

```
-- 4. Converter OFC_VISIT e ER_ED_VISIT em VARCHAR(45)
ALTER TABLE temp_do_hospital MODIFY COLUMN OFC_VISIT VARCHAR(45);
ALTER TABLE temp_do_hospital MODIFY COLUMN ER_ED_VISIT VARCHAR(45);

-- 5. Converter HOSPDAYS em INT
ALTER TABLE temp_do_hospital MODIFY COLUMN HOSPDAYS INT;
SELECT COLUMN_NAME, data_type FROM information_schema.COLUMNS WHERE table_schema = 'trabalho_covid'
AND TABLE_NAME = 'temp_do_hospital';
```

Figura 16. Query's efetuadas na tabela temp\_hospital (parte 2).


Result Grid		Filter Rows:		Exp
VAERS_ID	HOSPDAYS	OFC_VISIT	ER_ED_VISIT	

Figura 17. Resultado obtido na query efetuada na zona 3 na tabela temp\_do\_hospital.

COLUMN_NAME	DATA_TYPE
ER_ED_VISIT	varchar
HOSPDAYS	int
OFC_VISIT	varchar
VAERS_ID	int

**Figura 18.** Resultado obtido na *query* efetuada na zona 5 na tabela temp\_do\_hospital.

VAERS_ID	HOSPDAYS	OFC_VISIT	ER_ED_VISIT
916600	NULL	Y	NULL
916601	NULL	Y	NULL
916602	NULL	NULL	Y
916603	NULL	NULL	NULL
916604	NULL	NULL	NULL
916606	NULL	NULL	NULL
916607	NULL	NULL	NULL
916608	NULL	NULL	NULL
916609	NULL	NULL	NULL
916610	NULL	NULL	NULL
916611	NULL	Y	NULL
916613	NULL	NULL	NULL
916614	NULL	NULL	Y
916615	NULL	NULL	NULL

**Figura 19.** Parte da informação presente na tabela temp\_do\_hospital.

#### 5.1.5. Tabela temp\_paciente

No *dataset* do *kaggle*, as datas estavam no formato mm/dd/yyyy que não é reconhecido como um formato de data no Excel. Deste modo, não foi possível converter as datas presentes no *dataset* no formato yyyy-mm-dd, que é o formato de datas do *mySQL*. Apesar do *import* das datas ter sido em formato TEXT, não foi possível utilizar o comando ALTER TABLE MODIFY COLUMN para converter essas datas de TEXT para DATE, já que não se encontravam no formato assumido pelo *mySQL*. Assim sendo, foi necessário desenvolver um script em *Python*, que convertesse as datas do formato mm/dd/yyyy para o formato yyyy-mm-dd.

Na Figura 20 está representada a *script* desenvolvida. Como se pode observar, esta *script* aplicou-se à tabela em bruto proveniente do *dataset*, cobrindo por isso todas as tabelas temp. Na *script*, inicialmente é aberto o ficheiro em bruto (f) e também f2 que é o novo ficheiro onde



ficarão as datas no formato pretendido. Para cada coluna do *dataset* que contém datas (2, 6, 9, 10), faz-se o *split* da data pelas “/” e adicionam-se esses *split’s* a uma lista *split* que será *split* = [mm, dd, yyyy]. A *newDate* será obtida através da troca dos elementos da lista para que passará a ser [yyyy, mm, dd] e adicionando “-” entre eles. Assim, o formato será yyyy-mm-dd. Por último, as novas datas são colocadas no ficheiro f2.

Nas Figuras 21 e 22 estão representadas todas as *query’s* efetuadas na tabela **temp\_paciente**. Na zona 1 converteram-se as *string’s* vazias em valores nulos, pelos motivos anteriormente explicados na secção 5.1.1. Para tal, utilizou-se o comando UPDATE, que atualizou os valores “” da **temp\_paciente** para NULL. Na zona 2 criou-se uma tabela **temp\_dos\_pacientes** cujos VAERS\_ID existiam simultaneamente na tabela **temp\_paciente** e na tabela **temp\_vacinar**. Esta operação foi necessária, pelos motivos anteriormente explicados na secção 5.1.1. Para realizar esta *query*, recorreu-se ao comando CREATE TABLE, tendo sido realizado à posteriori o INSERT INTO na tabela **temp\_dos\_pacientes** das linhas da tabela **temp\_paciente** cujos VAERS\_ID da tabela **temp\_paciente** eram iguais aos VAERS\_ID da tabela **temp\_vacinar**.

Na zona 3 converteu-se o *datatype* (previamente sujeito à *script*) da coluna DATEDIED em DATE, para ficar em conformidade com o *datatype* dessa coluna no modelo lógico. Para tal recorreu-se ao comando ALTER TABLE MODIFY COLUMN. Na zona 4 converteu-se o *datatype* das colunas STATE e SEX em VARCHAR (45), para ficar em conformidade com o *datatype* dessa coluna no modelo lógico. Para tal recorreu-se ao comando ALTER TABLE MODIFY COLUMN. Por último, na zona 5 converteu-se o *datatype* da coluna AGE\_YRS em DECIMAL (5,2), para ficar em conformidade com o *datatype* dessa coluna no modelo lógico. Para tal recorreu-se ao comando ALTER TABLE MODIFY COLUMN e verificou-se se a operação tinha sido corretamente efetuada (Figura 23).

Na Figura 24 está representada parte da informação presente na tabela final **temp\_dos\_pacientes**, obtida através da *query* SELECT \* FROM **temp\_dos\_pacientes**.

```

f = open("2021VAERSDATA_Formatado-Cópia.csv", "r+")
f2 = open("2021VAERSDATA_Formatado-Cópia2.csv", "w+")

split = []
newLine = ""

f.readline()
for line in f.readlines():
    split = line.split(";")

    if(split[1]):
        data = split[1]
        nd = data.split("/")
        newDate = nd[2] + "-" + nd[0] + "-" + nd[1]
        split[1] = newDate

    if(split[8]):
        data = split[8]
        nd = data.split("/")
        newDate = nd[2] + "-" + nd[0] + "-" + nd[1]
        split[8] = newDate

    if(split[5]):
        data = split[5]
        nd = data.split("/")
        newDate = nd[2] + "-" + nd[0] + "-" + nd[1]
        split[5] = newDate

    if(split[9]):
        data = split[9]
        nd = data.split("/")
        newDate = nd[2] + "-" + nd[0] + "-" + nd[1]
        split[9] = newDate

    newLine = ';' .join(split)
    f2.writelines(newLine)

```

Figura 20. Script desenvolvida para converter o formato das datas de mm/dd/yyyy para o formato yyyy-mm-dd.

```

-- Tabela temp_paciente
-- 1. Converter strings vazias em valores nulos
UPDATE temp_paciente SET STATE = NULL WHERE STATE = '';
UPDATE temp_paciente SET AGE_YRS = NULL WHERE AGE_YRS = '';
UPDATE temp_paciente SET SEX = NULL WHERE SEX = '';
UPDATE temp_paciente SET DATEDIED = NULL WHERE DATEDIED = '';

-- 2. Criar tabela temp_dos_pacientes cujos pacientes são iguais aos do temp_vacinar
CREATE TABLE temp_dos_pacientes (VAERS_ID INT, STATE TEXT, AGE_YRS DECIMAL (5,0), SEX TEXT, DATEDIED DATE);

INSERT INTO temp_dos_pacientes (VAERS_ID, STATE, AGE_YRS, SEX, DATEDIED)
SELECT p.VAERS_ID, p.STATE, p.AGE_YRS, p.SEX, p.DATEDIED
FROM temp_vacinar vr, temp_paciente p
WHERE vr.VAERS_ID = p.VAERS_ID;

-- 3. Converter DATEDIED em DATE
ALTER TABLE temp_dos_pacientes MODIFY COLUMN DATEDIED DATE;

```

Figura 21. Query's efetuadas na Tabela temp\_paciente (parte 1).

```
-- 4. Converter STATE e SEX em VARCHAR(45)
ALTER TABLE temp_dos_pacientes MODIFY COLUMN STATE VARCHAR(45);
ALTER TABLE temp_dos_pacientes MODIFY COLUMN SEX VARCHAR(45);

-- 5. Converter AGE_YRS em DECIMAL(5,2)
ALTER TABLE temp_dos_pacientes MODIFY COLUMN AGE_YRS DECIMAL(5,2);
SELECT COLUMN_NAME, data_type FROM information_schema.COLUMNS WHERE table_schema = 'trabalho_covid'
AND TABLE_NAME = 'temp_dos_pacientes';
```

**Figura 22.** Query's efetuadas na Tabela temp\_paciente (parte 2).

COLUMN_NAME	DATA_TYPE
ER_ED_VISIT	varchar
HOSPDAYS	int
OFC_VISIT	varchar
VAERS_ID	int

**Figura 23.** Resultado obtido na query efetuada na zona 5 na tabela temp\_dos\_pacientes.

VAERS_ID	STATE	AGE_YRS	SEX	DATEDIED
916600	TX	33.00	F	NULL
916601	CA	73.00	F	NULL
916602	WA	23.00	F	NULL
916603	WA	58.00	F	NULL
916604	TX	47.00	F	NULL
916606	NV	44.00	F	NULL
916607	KS	50.00	M	NULL
916608	OH	33.00	M	NULL
916609	TN	71.00	F	NULL
916610	VA	18.00	F	NULL
916611	NC	33.00	F	NULL
916613	CA	40.00	F	NULL
916614	NY	29.00	F	NULL
916615	NY	38.00	F	NULL

**Figura 24.** Parte da informação presente na tabela temp\_dos\_pacientes.

### 5.1.6. Tabela temp\_sintomas

Nas Figuras 25 e 26 estão representadas todas as *query's* efetuadas na Tabela **temp\_sintomas**. Na zona 1 converteram-se as *string's* vazias em valores nulos, pelos motivos anteriormente explicados na secção 5.1.1. Para tal, utilizou-se o comando UPDATE, que atualizou os valores "" da **temp\_sintomas** para NULL. Na zona 2 criou-se uma tabela **temp\_dos\_sintomas** cujos VAERS\_ID existiam simultaneamente na tabela **temp\_sintomas** e na tabela **temp\_vacinar**. Esta operação foi necessária, pelos motivos anteriormente explicados na secção 5.1.1. Para realizar esta *query*, recorreu-se ao comando CREATE TABLE, tendo sido realizado à posteriori o INSERT INTO na tabela **temp\_dos\_sintomas** das linhas da tabela **temp\_sintomas** cujos VAERS\_ID da tabela **temp\_sintomas** eram iguais aos VAERS\_ID da tabela **temp\_vacinar**.

Por último, na zona 3 converteu-se o *datatype* das colunas SYMPTOM1, SYMPTOM2, SYMPTOM3, SYMPTOM4 e SYMPTOM 5 em VARCHAR (1000), para ficar em conformidade com o *datatype* dessa coluna no modelo lógico. Optou-se por não deixar TEXT, pois apesar dos sintomas poderem ser *string's* longas, estas continuam a não ser um texto. Para tal recorreu-se ao comando ALTER TABLE MODIFY COLUMN e verificou-se se a operação tinha sido corretamente efetuada (Figura 27).

Na Figura 28 está representada parte da informação presente na tabela final **temp\_dos\_sintomas**, obtida através da *query* SELECT \* FROM **temp\_dos\_sintomas**.

```
-- Tabela temp_sintomas
-- 1. Converter strings vazias em valores nulos
UPDATE temp_sintomas SET SYMPTOM1 = NULL WHERE SYMPTOM1 = '';
UPDATE temp_sintomas SET SYMPTOM2 = NULL WHERE SYMPTOM2 = '';
UPDATE temp_sintomas SET SYMPTOM3 = NULL WHERE SYMPTOM3 = '';
UPDATE temp_sintomas SET SYMPTOM4 = NULL WHERE SYMPTOM4 = '';
UPDATE temp_sintomas SET SYMPTOM5 = NULL WHERE SYMPTOM5 = '';

-- 2. Criar tabela temp_dos_sintomas cujos pacientes são iguais aos do temp_vacinar
CREATE TABLE temp_dos_sintomas (VAERS_ID INT, SYMPTOM1 TEXT, SYMPTOM2 TEXT, SYMPTOM3 TEXT, SYMPTOM4 TEXT, SYMPTOM5 TEXT);

INSERT INTO temp_dos_sintomas (VAERS_ID, SYMPTOM1, SYMPTOM2, SYMPTOM3, SYMPTOM4, SYMPTOM5)
SELECT s.VAERS_ID, s.SYMPTOM1, s.SYMPTOM2, s.SYMPTOM3, s.SYMPTOM4, s.SYMPTOM5
FROM temp_vacinar vr, temp_sintomas s
WHERE vr.VAERS_ID = s.VAERS_ID;
```

Figura 25. Query's efetuadas na tabela temp\_sintomas (parte 1).

COLUMN_NAME	DATA_TYPE
SYMPTOM1	varchar
SYMPTOM2	varchar
SYMPTOM3	varchar
SYMPTOM4	varchar
SYMPTOM5	varchar
VAERS_ID	int

Figura 27. Resultado obtido na query efetuada na zona 3 na tabela temp\_dos\_sintomas..

```
-- 3. Converter SYMPTOM (1 a 5) em VARCHAR(1000)
ALTER TABLE temp_dos_sintomas MODIFY COLUMN SYMPTOM1 VARCHAR(1000);
ALTER TABLE temp_dos_sintomas MODIFY COLUMN SYMPTOM2 VARCHAR(1000);
ALTER TABLE temp_dos_sintomas MODIFY COLUMN SYMPTOM3 VARCHAR(1000);
ALTER TABLE temp_dos_sintomas MODIFY COLUMN SYMPTOM4 VARCHAR(1000);
ALTER TABLE temp_dos_sintomas MODIFY COLUMN SYMPTOM5 VARCHAR(1000);
SELECT COLUMN_NAME, data_type FROM information_schema.COLUMNS WHERE table_schema = 'trabalho_covid'
AND TABLE_NAME = 'temp_dos_sintomas';
```

Figura 26. Query's efetuadas na tabela temp\_sintomas (parte 2).

VAERS_ID	SYMPTOM1	SYMPTOM2	SYMPTOM3	SYMPTOM4	SYMPTOM5
916600	Dysphagia	Epiglottitis	NULL	NULL	NULL
916601	Anxiety	Dyspnoea	NULL	NULL	NULL
916602	Chest discomfort	Dysphagia	Pain in extremity	Visual impairment	NULL
916603	Dizziness	Fatigue	Mobility decreased	NULL	NULL
916604	Injection site erythema	Injection site pruritus	Injection site swelling	Injection site warmth	NULL
916606	Pharyngeal swelling	NULL	NULL	NULL	NULL
916607	Abdominal pain	Chills	Sleep disorder	NULL	NULL
916608	Diarrhoea	Nasal congestion	NULL	NULL	NULL
916609	Vaccination site erythema	Vaccination site pruritus	Vaccination site swelling	NULL	NULL
916610	Rash	Urticaria	NULL	NULL	NULL
916611	Blood pressure decreased	Chest pain	Chills	Confusional state	Decreased...
916611	Dyspnoea	Fatigue	Feeling abnormal	Head discomfort	Headache
916611	Heart rate decreased	Heart rate increased	Hypertension	Injection site pain	Musculosk...

Figura 28. Parte da informação presente na tabela temp\_dos\_sintomas.

### 5.1.7. Tabela temp\_sintomas2

Na Figura 29 estão representadas todas as query's efetuadas na tabela **temp\_sintomas2**. Na zona 1 converteram-se as *string's* vazias em valores nulos, pelos motivos anteriormente explicados na secção 5.1.1. Para tal, utilizou-se o comando UPDATE, que atualizou os valores ""

da **temp\_sintomas2** para NULL. Na zona 2 criou-se uma tabela **temp\_dos\_sintomas2** cujos VAERS\_ID existiam simultaneamente na tabela **temp\_sintomas2** e na tabela **temp\_vacinar**. Esta operação foi necessária, pelos motivos anteriormente explicados na secção 5.1.1. Para realizar esta *query*, recorreu-se ao comando CREATE TABLE, tendo sido realizado à posteriori o INSERT INTO na tabela **temp\_dos\_sintomas2** das linhas da tabela **temp\_sintomas2** cujos VAERS\_ID da tabela **temp\_sintomas2** eram iguais aos VAERS\_ID da tabela **temp\_vacinar**.

Na zona 3 converteu-se o *datatype* (previamente sujeito à *script*) da coluna ONSET\_DATE em DATE, para ficar em conformidade com o *datatype* dessa coluna no modelo lógico. Para tal recorreu-se ao comando ALTER TABLE MODIFY COLUMN. Por último, na zona 5 converteu-se o *datatype* da coluna NMDAYS em INT, para ficar em conformidade com o *datatype* dessa coluna no modelo lógico. Para tal recorreu-se ao comando ALTER TABLE MODIFY COLUMN e verificou-se se a operação tinha sido corretamente efetuada (Figura 30).

Na Figura 31 está representada parte da informação presente na tabela final **temp\_dos\_sintomas2**, obtida através da *query* SELECT \* FROM **temp\_dos\_sintomas2**.

```
-- Tabela temp_sintomas2
-- 1. Converter strings vazias em valores nulos
UPDATE temp_sintomas2 SET ONSET_DATE = NULL WHERE ONSET_DATE = '';
UPDATE temp_sintomas2 SET NMDAYS = NULL WHERE NMDAYS = '';

-- 2. Criar tabela temp_dos_sintomas2 cujos pacientes são iguais aos do temp_vacinar
CREATE TABLE temp_dos_sintomas2 (VAERS_ID INT, ONSET_DATE DATE, NMDAYS INT);

INSERT INTO temp_dos_sintomas2 (VAERS_ID, ONSET_DATE, NMDAYS)
SELECT si.VAERS_ID, si.ONSET_DATE, si.NMDAYS
FROM temp_sintomas2 si, temp_vacinar vr
WHERE si.VAERS_ID = vr.VAERS_ID;

-- 3. Converter ONSET_DATE em DATE
ALTER TABLE temp_dos_sintomas2 MODIFY COLUMN ONSET_DATE DATE;

-- 4. Converter NMDAYS em INT
ALTER TABLE temp_dos_sintomas2 MODIFY COLUMN NMDAYS INT;
SELECT COLUMN_NAME, data_type FROM information_schema.COLUMNS WHERE table_schema = 'trabalho_covid'
AND TABLE_NAME = 'temp_dos_sintomas2';
```

Figura 29. Query's efetuadas na tabela temp\_sintomas2.

COLUMN_NAME	DATA_TYPE
NUMDAYS	int
ONSET_DATE	date
VAERS_ID	int

**Figura 30.** Resultado obtido na *query* efetuada na zona 4 na tabela temp\_dos\_sintomas2.

VAERS_ID	ONSET_DATE	NUMDAYS
916600	2020-12-30	2
916601	2020-12-31	0
916602	2020-12-31	0
916603	2020-12-23	0
916604	2020-12-29	7
916606	2020-12-29	0
916607	2020-12-29	1
916608	2020-12-31	2
916609	2020-12-31	8
916610	2020-12-30	1
916611	2020-12-29	0
916613	2020-12-30	0
916614	2020-12-22	0
916615	2020-12-31	8
916617	2020-12-30	0
916618	2020-12-31	0

**Figura 31.** Parte da informação presente na tabela temp\_dos\_sintomas2.

#### 5.1.8. Tabela temp\_vacina

Nas Figuras 32 e 33 estão representadas todas as *query's* efetuadas na tabela **temp\_vacina** Na zona 1 eliminaram-se todas as linhas em que VAX\_TYPE não era "COVID19", uma vez que o intuito deste trabalho é analisar os efeitos provocados por vacinas exclusivamente para a COVID-19. Para tal, recorreu-se ao comando DELETE.

Na zona 2 converteram-se as *string's* vazias em valores nulos, pelos motivos anteriormente explicados na secção 5.1.1. Para tal, utilizou-se o comando UPDATE, que atualizou os valores "" da **temp\_vacina** para NULL. Na zona 3 criou-se uma tabela **temp\_das\_vacinas** cujos VAERS\_ID existiam simultaneamente na tabela **temp\_vacina** e na tabela **temp\_vacinar**. Esta operação foi necessária, pelos motivos anteriormente explicados na secção 5.1.1. Para realizar esta *query*, recorreu-se ao comando CREATE TABLE, tendo sido realizado à posteriori o INSERT INTO na

tabela **temp\_das\_vacinas** das linhas da tabela **temp\_vacina** cujos VAERS\_ID da tabela **temp\_vacina** eram iguais ao VAERS\_ID da tabela **temp\_vacinar**. Além disso, adicionou-se na tabela **temp\_das\_vacinas** uma coluna com o idvacina auto incremental, de forma a facilitar o povoamento da tabela vacinar apresentado na secção 5.2.4.

Na zona 4 converteu-se o *datatype* da coluna VAX\_DOSE\_SERIES em INT, para ficar em conformidade com o *datatype* dessa coluna no modelo lógico. Para tal recorreu-se ao comando ALTER TABLE MODIFY COLUMN. Por último, na zona 5 converteu-se o *datatype* das colunas VAX\_LOT, VAX\_ROUTE e VAX\_SITE em VARCHAR (45), para ficar em conformidade com o *datatype* dessa coluna no modelo lógico. Para tal recorreu-se ao comando ALTER TABLE MODIFY COLUMN e verificou-se se a operação tinha sido corretamente efetuada (Figura 34).

Na Figura 35 está representada parte da informação presente na tabela final **temp\_das\_vacinas**, obtida através da *query* SELECT \* FROM **temp\_das\_vacinas**.

```
-- Tabela temp_vacina
-- 1. Eliminar as linhas que não possuem pacientes que tomaram vacinas da covid
DELETE FROM temp_vacina WHERE VAX_TYPE != 'COVID19';

-- 2. Converter strings vazias em valores nulos
UPDATE temp_vacina SET VAX_DOSE_SERIES = NULL WHERE VAX_DOSE_SERIES = '';
UPDATE temp_vacina SET VAX_TYPE = NULL WHERE VAX_TYPE = '';
UPDATE temp_vacina SET VAX_LOT = NULL WHERE VAX_LOT = '';
UPDATE temp_vacina SET VAX_ROUTE = NULL WHERE VAX_ROUTE = '';
UPDATE temp_vacina SET VAX_SITE = NULL WHERE VAX_SITE = '';

-- 3. Criar tabela temp_das_vacinas cujos pacientes são iguais aos do temp_vacinar e com atributo idvacina
-- (para facilitar povoamento tabela vacinar)
CREATE TABLE temp_das_vacinas (VAERS_ID INT, VAX_LOT TEXT, VAX_DOSE_SERIES INT, VAX_ROUTE TEXT, VAX_SITE TEXT,
idvacina INT NOT NULL AUTO_INCREMENT, PRIMARY KEY (idvacina));
```

Figura 33. Query's efetuadas na tabela temp\_vacina (parte 1).

```
-- 4. Converter VAX_DOSE_SERIES em INT
ALTER TABLE temp_das_vacinas MODIFY COLUMN VAX_DOSE_SERIES INT;

-- 5. Converter VAX_LOT, VAX_ROUTE, VAX_SITE em VARCHAR(45)
ALTER TABLE temp_das_vacinas MODIFY COLUMN VAX_LOT VARCHAR(45);
ALTER TABLE temp_das_vacinas MODIFY COLUMN VAX_ROUTE VARCHAR(45);
ALTER TABLE temp_das_vacinas MODIFY COLUMN VAX_SITE VARCHAR(45);
SELECT COLUMN_NAME, data_type FROM information_schema.COLUMNS WHERE table_schema = 'trabalho_covid'
AND TABLE_NAME = 'temp_das_vacinas';
```

Figura 32. Query's efetuadas na tabela temp\_vacina (parte 2).



COLUMN_NAME	DATA_TYPE
idvacina	int
VAERS_ID	int
VAX_DOSE_SERIES	int
VAX_LOT	varchar
VAX_ROUTE	varchar
VAX_SITE	varchar

**Figura 34.** Resultado obtido na *query* efetuada na zona 5 na tabela temp\_das\_vacinas.

	VAERS_ID	VAX_LOT	VAX_DOSE_SERIES	VAX_ROUTE	VAX_SITE	idvacina
▶	916600	03K20A	1	IM	LA	1
	916601	025L20A	1	IM	RA	2
	916602	EL1284	1	IM	LA	3
	916603	NULL	NULL	NULL	NULL	4
	916604	NULL	1	IM	LA	5
	916606	011J20A	1	IM	LA	6
	916608	NULL	1	IM	LA	7
	916609	011J201A	1	IM	LA	8
	916610	NULL	1	SYR	LA	9
	916611	039k20a	1	SYR	RA	10
	916613	025J20-2A	1	SYR	LA	11
	916614	011J20A	1	IM	LA	12
	916615	025J20A	1	IM	LA	13

**Figura 35.** Parte da informação presente na tabela temp\_das\_vacinas.

### 5.1.9. Tabela temp\_vacinar

Na Figura 36 estão representadas todas as *query's* efetuadas na Tabela **temp\_vacinar**. Na zona 1 eliminaram-se todas as linhas em que não existe VAERS\_ID ou não existe VAX\_DATE, uma vez que estes serão PK na tabela vacinar e por isso não podem ser nulos. Além disso, não faz sentido analisar pacientes que não tenham sido vacinados com uma vacina da COVID-19.

Por último, na zona 2 converteu-se o *datatype* (previamente sujeito à *script*) da coluna VAX\_DATE em DATE, para ficar em conformidade com o *datatype* dessa coluna no modelo lógico. Para tal recorreu-se ao comando ALTER TABLE MODIFY COLUMN e verificou-se se a operação tinha sido corretamente efetuada (Figura 37).

Na Figura 38 está representada parte da informação presente na tabela final **temp\_vacinar**, obtida através da *query* `SELECT * FROM temp_vacinar`.

```
-- Tabela temp_vacinar
-- 1. Eliminar as linhas em que não existe VAERS_ID ou VAX_DATE, agora não existem valores nulos!
DELETE FROM temp_vacinar WHERE VAERS_ID = '' OR VAX_DATE = '';

-- 2. Converter VAX_DATE em date
ALTER TABLE temp_vacinar MODIFY COLUMN VAX_DATE DATE;
SELECT COLUMN_NAME, data_type FROM information_schema.COLUMNS WHERE table_schema = 'trabalho_covid'
AND TABLE_NAME = 'temp_vacinar';
```

Figura 36. Query's efetuadas na Tabela temp\_vacinar.

COLUMN_NAME	DATA_TYPE
VAERS_ID	int
vax_date	date

Figura 37. Resultado obtido na query efetuada na zona 2 na tabela temp\_vacinar.

VAERS_ID	vax_date
916600	2020-12-28
916601	2020-12-31
916602	2020-12-31
916603	2020-12-23
916604	2020-12-22
916606	2020-12-29
916607	2020-12-28
916608	2020-12-29
916609	2020-12-23
916610	2020-12-29
916611	2020-12-29
916613	2020-12-30
916614	2020-12-22

Figura 38. Parte da informação presente na tabela temp\_vacinar.

## 5.2. Povoamento

Após se ter feito todo o pré-povoamento dos dados, estes estão prontos para povoar as tabelas do modelo lógico. Na Tabela 6 constam as tabelas que serão utilizadas nos povoamentos das tabelas do modelo. A ordem pelo qual se procedeu ao povoamento foi: fornecedor, paciente, vacina, vacinar, historico\_clinico, doenca\_afeta\_paciente, hospitalizacao, paciente\_e\_hospitalizado, sintomas, sintomas\_afeta\_paciente e vacina\_causa\_sintomas.

**Tabela 6.** Nomes das tabelas onde se encontram os dados para povoar as tabelas do modelo lógico.

Nome da tabela do modelo lógico	Nome(s) da(s) tabela(s) temp
fornecedor	temp_do_fornecedor
vacina	temp_do_fornecedor, temp_das_vacinas
vacina_causa_sintomas	---
vacinar	temp_vacinar, temp_das_vacinas, temp_do_historico
sintomas	temp_dos_sintomas
sintomas_afeta_paciente	temp_dos_sintomas, temp_da_doenca, temp_dos_sintomas2
historico_clinico	---
doenca_afeta_paciente	temp_da_doenca, temp_do_historico
paciente	temp_dos_pacientes
paciente_e_hospitalizado	temp_do_hospital
hospitalizacao	---

De seguida será explicado detalhadamente o povoamento de cada uma das tabelas. É de realçar que o povoamento dessas tabelas foi realizado no *Schema* covid\_efeitos.

### 5.2.1. Tabela fornecedor

Na Figura 39 estão representadas todas as *query's* efetuadas para povoar a tabela **fornecedor**. Pela análise da figura, verifica-se que se recorreu ao comando INSERTO INTO para colocar na tabela **fornecedor** todos os VAX\_MANU diferentes que existem na tabela **temp\_do\_fornecedor**

e em que VAX\_MANU não fosse NOT NULL. Sendo idfornecedor AI, para cada linha da tabela **fornecedor**, o idfornecedor auto incrementa o seu valor INT de 1 até ao nº total de linhas.

Posteriormente, recorrendo à *query* `SELECT * FROM fornecedor`; verificou-se que o povoamento tinha sido corretamente efetuado, com um total de 13 fornecedores de vacinas para a COVID-19 (Figura 40).

```
-- Tabela fornecedor
INSERT INTO fornecedor (vax_manu)
SELECT DISTINCT VAX_MANU FROM trabalho_covid.temp_do_fornecedor WHERE VAX_MANU IS NOT NULL;
SELECT * FROM fornecedor;
```

**Figura 39.** Query's efetuadas para povoar a tabela fornecedor e confirmar os resultados obtidos no povoamento.

idfornecedor	vax_manu
1	MODERA
2	PFIZER\BIOTECH
3	SAOFI PASTEUR
4	MERCK & CO. IC.
5	MAUFACTURER
6	GLAXOSMITHKLINE BIOLOGICALS
7	OVTIS VACCIES AD DIAGOSTICS
8	SEQIRUS, IC.
9	PFIZER\WYETH
10	PROTEI SCIECES CORPORATIO
11	MEDIMME VACCIES, IC.
12	BERA BIOTECH, LTD.
13	JASSE
NULL	NULL

**Figura 40.** Tabela fornecedor.

### 5.2.2. Tabela vacina

Na Figura 41 estão representadas todas as *query's* efetuadas para povoar a tabela **vacina**. Pela análise da figura, verifica-se que se recorreu ao comando `INSERTO INTO` para colocar na tabela **vacina** todos os VAX\_LOT que existem na tabela **temp\_das\_vacinas** e todos os idfornecedor presentes na tabela **fornecedor**, em que os VAERS\_ID da tabela **temp\_das\_vacinas** fossem iguais aos VAERS\_ID da tabela **temp\_do\_fornecedor** e os VAX\_MANU da tabela

**temp\_do\_fornecedor** fossem iguais aos VAX\_MANU da tabela **fornecedor**. Estas condições foram necessárias para que se pudesse estabelecer a ligação entre cada paciente e o fabricante da vacina que este tomou. Para tal, recorreu-se à tabela **temp\_do\_fornecedor**, não para o *import* propriamente dito, mas para estabelecer essa ligação entre **vacina** e **fornecedor**. Sendo idvacina AI, para cada linha da tabela **vacina**, o idvacina auto incrementa o seu valor INT de 1 até ao nº total de linhas.

Posteriormente, recorrendo à *query* `SELECT * FROM vacina;` verificou-se que o povoamento tinha sido corretamente efetuado (Figura 42).

```
-- Tabela vacina
INSERT INTO vacina (vax_lot, idfornecedor)
SELECT v.VAX_LOT, f.idfornecedor
FROM fornecedor f, trabalho_covid.temp_do_fornecedor t, trabalho_covid.temp_das_vacinas v
WHERE v.VAERS_ID = t.VAERS_ID
AND t.VAX_MANU = f.vax_manu;
SELECT * FROM vacina;
```

**Figura 41.** *Query's* efetuadas para povoar a tabela vacina e confirmar os resultados obtidos no povoamento.

	idvacina	vax_lot	idfornecedor
	1	03K20A	1
	2	025L20A	1
	3	EL1284	2
	4	NULL	1
	5	NULL	1
	6	011J20A	1
	7	NULL	1
	8	011J201A	1
	9	NULL	1
	10	039k20a	1
	11	025J20-2A	1
	12	011J20A	1
	13	025J20A	1
	14	NULL	1
	15	NULL	2

**Figura 42.** Parte da informação presente na tabela vacina.

### 5.2.3. Tabela paciente

Na Figura 43 estão representadas todas as *query's* efetuadas para povoar a tabela **paciente**. Pela análise da figura, verifica-se que se recorreu ao comando `INSERTO INTO` para colocar na tabela

**paciente** todos os dados das colunas VAERS\_ID, STATE, SEX, AGE\_YRS, DATEDIED que existem na tabela **temp\_dos\_pacientes**, em que o VAERS\_ID fosse NOT NULL.

Posteriormente, recorrendo à *query* `SELECT * FROM paciente;` verificou-se que o povoamento tinha sido corretamente efetuado (Figura 44).

```
-- Tabela paciente
INSERT INTO paciente (id_vaers, state, sex, age_yrs, data_died)
SELECT VAERS_ID, STATE, SEX, AGE_YRS, DATEDIED FROM trabalho_covid.temp_dos_pacientes WHERE VAERS_ID IS NOT NULL;
SELECT * FROM paciente;
```

**Figura 43.** Query's efetuadas para povoar a tabela paciente e confirmar os resultados obtidos no povoamento.

id_vaers	state	sex	age_yrs	data_died
916600	TX	F	33.00	NULL
916601	CA	F	73.00	NULL
916602	WA	F	23.00	NULL
916603	WA	F	58.00	NULL
916604	TX	F	47.00	NULL
916606	NV	F	44.00	NULL
916607	KS	M	50.00	NULL
916608	OH	M	33.00	NULL
916609	TN	F	71.00	NULL
916610	VA	F	18.00	NULL
916611	NC	F	33.00	NULL
916613	CA	F	40.00	NULL
916614	NY	F	29.00	NULL
916615	NY	F	38.00	NULL
916617	CA	F	35.00	NULL

**Figura 44.** Parte da informação presente na tabela paciente.

#### 5.2.4. Tabela vacinar

Na Figura 45 estão representadas todas as *query's* efetuadas para povoar a tabela **vacinar**. Pela análise da figura, verifica-se que se recorreu ao comando `INSERTO INTO` para colocar na tabela **vacinar** os idvacina da tabela **vacina**, os id\_vaers da tabela **paciente**, as VAX\_DATE da tabela

**temp\_vacinar**, os dados presentes nas colunas VAX\_DOSE\_SERIES, VAX\_ROUTE, VAX\_SITE da tabela **temp\_das\_vacinas** e a coluna V\_ADMINBY da tabela **temp\_do\_historico**. Contudo, o INSERT foi sujeito a várias restrições: os VAERS\_ID das tabelas **temp\_das\_vacinas**, **temp\_vacinar** e **temp\_do\_historico** têm de ser iguais aos id\_vaers da tabela **paciente** e os idvacina da tabela **vacina** e da tabela **temp\_das\_vacinas** têm de ser iguais.

Estas condições foram necessárias para que se pudesse estabelecer a ligação entre cada paciente e a sua data de vacinação, o nº de doses de vacina que tomou, de que forma lhe foi aplicada a vacina e onde foi aplicada a vacina no corpo (braço esquerdo, direito, ...) e em termos de instituição (hospital público, privado,). Assim, igualaram-se os diferentes VAERS\_ID das várias tabelas temp aos id\_vaers da tabela **paciente**. Relativamente à igualdade dos idvacinas, como na tabela **vacina** a PK é o idvacina (que distingue inequivocamente uma vacina da outra), então apenas foi possível estabelecer a ligação entre as informações presentes nas tabelas **vacina** e **temp\_das\_vacinas** através dos id's.

Posteriormente, recorrendo à *query* SELECT \* FROM **vacinar**; verificou-se que o povoamento tinha sido corretamente efetuado (Figura 46)

```
-- Tabela vacinar
INSERT INTO vacinar (idvacina, id_vaers, vax_date, vax_dose_series, vax_route, vax_site, v_adminby)
SELECT v.idvacina, p.id_vaers, va.VAX_DATE, vac.VAX_DOSE_SERIES, vac.VAX_ROUTE, vac.VAX_SITE, h.V_ADMINBY
FROM vacina v, paciente p, trabalho_covid.temp_vacinar va, trabalho_covid.temp_das_vacinas vac,
trabalho_covid.temp_do_historico h
WHERE va.VAERS_ID = p.id_vaers
AND vac.VAERS_ID = p.id_vaers
AND h.VAERS_ID = p.id_vaers
AND v.idvacina = vac.idvacina;
SELECT * FROM vacinar;
```

**Figura 45.** Query's efetuadas para povoar a tabela vacinar e confirmar os resultados obtidos no povoamento.

idvacina	id_vaers	vax_date	vax_dose_series	vax_route	vax_site	v_adminby
1	916600	2020-12-28	1	IM	LA	PVT
2	916601	2020-12-31	1	IM	RA	SEN
3	916602	2020-12-31	1	IM	LA	SEN
4	916603	2020-12-23	NULL	NULL	NULL	WRK
5	916604	2020-12-22	1	IM	LA	PUB
6	916606	2020-12-29	1	IM	LA	PVT
7	916608	2020-12-29	1	IM	LA	NULL
8	916609	2020-12-23	1	IM	LA	PUB
9	916610	2020-12-29	1	SYR	LA	PVT
10	916611	2020-12-29	1	SYR	RA	NULL
11	916613	2020-12-30	1	SYR	LA	NULL
12	916614	2020-12-22	1	IM	LA	PVT
13	916615	2020-12-23	1	IM	LA	PVT
14	916618	2020-12-31	1	IM	LA	PVT
15	916619	2020-12-22	1	IM	LA	NULL

**Figura 46.** Parte da informação presente na tabela vacinar.

#### 5.2.5. Tabela historico\_clinico

Na Figura 47 estão representadas todas as *query's* efetuadas para povoar a tabela **historico\_clinico**. Pela análise da figura, verifica-se que se recorreu ao comando INSERTO INTO dos VALUES que são os nomes das colunas da tabela **temp\_do\_historico** e da tabela **temp\_da\_doenca**, no caso de "Disable". Sendo iddoenca AI, para cada linha da tabela **historico\_clinico**, o iddoenca auto incrementa o seu valor INT de 1 até ao nº total de linhas.

Posteriormente, recorrendo à *query* SELECT \* FROM **historico\_clinico**; verificou-se que o povoamento tinha sido corretamente efetuado (Figura 48).



```
-- Tabela histórico clínico
INSERT INTO historico_clinico (designacao) VALUE ('Obesity');
INSERT INTO historico_clinico (designacao) VALUE ('High_Cholesterol');
INSERT INTO historico_clinico (designacao) VALUE ('High_Blood_Pressure');
INSERT INTO historico_clinico (designacao) VALUE ('Migraines');
INSERT INTO historico_clinico (designacao) VALUE ('GERD');
INSERT INTO historico_clinico (designacao) VALUE ('Depression');
INSERT INTO historico_clinico (designacao) VALUE ('Anxiety');
INSERT INTO historico_clinico (designacao) VALUE ('Osteoarthritis');
INSERT INTO historico_clinico (designacao) VALUE ('Hypothyroidism');
INSERT INTO historico_clinico (designacao) VALUE ('Hypertension');
INSERT INTO historico_clinico (designacao) VALUE ('Asthma');
INSERT INTO historico_clinico (designacao) VALUE ('Hyperlipidemia');
INSERT INTO historico_clinico (designacao) VALUE ('L_threat');
INSERT INTO historico_clinico (designacao) VALUE ('COVID-19');
INSERT INTO historico_clinico (designacao) VALUE ('Disable');
SELECT * FROM historico_clinico;
```

**Figura 47.** Query's efetuadas para povoar a tabela historico\_clinico e confirmar os resultados obtidos no povoamento.

iddoenca	designacao
1	Obesity
2	High_Cholesterol
3	High_Blood_Pressure
4	Migraines
5	GERD
6	Depression
7	Anxiety
8	Osteoarthritis
9	Hypothyroidism
10	Hypertension
11	Asthma
12	Hyperlipidemia
13	L_threat
14	COVID-19
15	Disable
NULL	NULL

**Figura 48.** Tabela historico\_clinico.

### 5.2.6. Tabela doenca\_afeta\_paciente

Na Figura 49 estão representadas todas as *query's* efetuadas para povoar a tabela **doenca\_afeta\_paciente**. Pela análise da figura, verifica-se que se recorreu ao comando INSERTO INTO para colocar na tabela **doenca\_afeta\_paciente** os iddoenca da tabela **historico\_clinico** e os id\_vaers da tabela **paciente**, em que os VAERS\_ID das tabelas **temp\_da\_doenca** e **temp\_do\_historico** são iguais aos id\_vaers da tabela **paciente** e em que para cada designacao da tabela **historico\_clinico**, a coluna da tabela **temp\_do\_historico** (ou **temp\_da\_doenca** no caso da designacao ser "Disable") apresenta "Yes", ou seja o paciente apresenta essa doença. É de realçar que nesta condição se usou o operador OR entre as várias designações, uma vez que o povoamento é para ser feito quando o paciente tiver pelo menos 1 das patologias e não quando o paciente tem todas as patologias. Além disso, neste povoamento recorreu-se às tabelas **temp\_do\_historico** e **temp\_da\_doenca** não para o *import* propriamente dito, mas para estabelecer essa ligação entre **paciente** e **historico\_clinico**.

Posteriormente, recorrendo à *query* SELECT \* FROM **doenca\_afeta\_paciente**; verificou-se que o povoamento tinha sido corretamente efetuado (Figura 50).

```
-- Tabela doenca_afeta_paciente
INSERT INTO doenca_afeta_paciente (iddoenca, id_vaers)
SELECT h.iddoenca, p.id_vaers
FROM historico_clinico h, paciente p, trabalho_covid.temp_da_doenca d, trabalho_covid.temp_do_historico ho
WHERE p.id_vaers = d.VAERS_ID
AND p.id_vaers = ho.VAERS_ID
AND (h.designacao = 'Obesity' AND ho.Obesity = 'Yes'
OR h.designacao = 'High_Cholesterol' AND ho.High_Cholesterol = 'Yes'
OR h.designacao = 'High_Blood_Pressure' AND ho.High_Blood_Pressure = 'Yes'
OR h.designacao = 'Migraines' AND ho.Migraines = 'Yes'
OR h.designacao = 'GERD' AND ho.GERD = 'Yes'
OR h.designacao = 'Depression' AND ho.Depression= 'Yes'
OR h.designacao = 'Anxiety' AND ho.Anxiety = 'Yes'
OR h.designacao = 'Osteoarthritis' AND ho.Osteoarthritis = 'Yes'
OR h.designacao = 'Hypothyroidism' AND ho.Hypothyroidism = 'Yes'
OR h.designacao = 'Hypertension' AND ho.Hypertension = 'Yes'
OR h.designacao = 'Hyperlipidemia' AND ho.Hypertension = 'Yes'
OR h.designacao = 'Asthma' AND ho.Asthma = 'Yes'
OR h.designacao = 'L_threat' AND ho.L_threat = 'Y'
OR h.designacao = 'Disable' AND d.Disable = 'Y'
OR h.designacao = 'COVID-19' AND ho.COVID19 = 'Yes');
SELECT * FROM doenca_afeta_paciente;
```

**Figura 49.** *Query's* efetuadas para povoar a tabela **doenca\_afeta\_paciente** e confirmar os resultados obtidos no povoamento.

id_doenca	id_vaers
8	916603
1	916607
4	916607
5	916607
6	916607
7	916607
8	916607
9	916607
10	916607
11	916607
14	916607
6	916609
10	916609
1	916611
7	916611
4	916615
6	916617

**Figura 50.** Parte da informação presente na tabela `doenca_afeta_paciente`.

### 5.2.7. Tabela `hospitalizacao`

Na Figura 51 estão representadas todas as *query's* efetuadas para povoar a tabela **historico\_clinico**. Pela análise da figura, verifica-se que se recorreu ao comando `INSERTO INTO` dos `VALUES` que correspondem a todas as combinações possíveis que podem ocorrer para um dado paciente: ir às urgências e receber visita médica; ir às urgências e não receber visita médica; não ir às urgências e receber visita médica; não ir às urgências e não receber visita médica. Sendo `id_hospitalizacao` AI, para cada linha da tabela **hospitalizacao**, o `id_hospitalizacao` auto incrementa o seu valor INT de 1 até ao nº total de linhas.

Posteriormente, recorrendo à *query* `SELECT * FROM hospitalizacao`; verificou-se que o povoamento tinha sido corretamente efetuado (Figura 52).

```
-- Tabela hospitalizacao
INSERT INTO hospitalizacao (er_ed_visit, ofc_visit) VALUE ('Yes', 'Yes');
INSERT INTO hospitalizacao (er_ed_visit, ofc_visit) VALUE ('Yes', 'No');
INSERT INTO hospitalizacao (er_ed_visit, ofc_visit) VALUE ('No', 'Yes');
INSERT INTO hospitalizacao (er_ed_visit, ofc_visit) VALUE ('No', 'No');
SELECT * FROM hospitalizacao;
```

**Figura 51.** *Query's* efetuadas para povoar a tabela `hospitalizacao` e confirmar os resultados obtidos no povoamento.

idhospitalizacao	er_ed_visit	ofc_visit
1	Yes	Yes
2	Yes	No
3	No	Yes
4	No	No
NULL	NULL	NULL

Figura 52. Tabela hospitalizacao.

### 5.2.8. Tabela paciente\_e\_hospitalizado

Na Figura 53 estão representadas todas as *query's* efetuadas para povoar a tabela **paciente\_e\_hospitalizado**. Pela análise da figura, verifica-se que se recorreu ao comando INSERTO INTO para colocar na tabela **paciente\_e\_hospitalizado** os idhospitalizacao da tabela **hospitalizacao**, os id\_vaers da tabela **paciente** e os HOSPDAYS da tabela **temp\_do\_hospital**, em que os VAERS\_ID da tabela **temp\_do\_hospital** são iguais aos id\_vaers da tabela **paciente** e em que para cada uma das 4 combinações de er\_ed\_visit e ofc\_visit da tabela **hospitalizacao**, a coluna da tabela **temp\_do\_hospital** apresenta uma combinação igual, ou seja o paciente apresenta uma dessas 4 combinações. É de realçar que nesta condição se usou o operador OR entre as várias designações, uma vez que o povoamento é para ser feito quando o paciente tiver 1 das combinações e não quando o paciente tem todas as combinações.

Posteriormente, recorrendo à *query* SELECT \* FROM **paciente\_e\_hospitalizado**; verificou-se que o povoamento tinha sido corretamente efetuado (Figura 54).

```
-- Tabela paciente_e_hospitalizado
INSERT INTO paciente_e_hospitalizado (idhospitalizacao, id_vaers, hospdays)
SELECT h.idhospitalizacao, p.id_vaers, ho.HOSPDAYS
FROM hospitalizacao h, paciente p, trabalho_covid.temp_do_hospital ho
WHERE p.id_vaers = ho.VAERS_ID
AND ((h.er_ed_visit = 'Yes' AND h.ofc_visit = 'Yes' AND ho.er_ed_visit = 'Y' AND ho.ofc_visit = 'Y')
OR (h.er_ed_visit = 'Yes' AND h.ofc_visit = 'No' AND ho.er_ed_visit = 'Y' AND ho.ofc_visit IS NULL)
OR (h.er_ed_visit = 'No' AND h.ofc_visit = 'Yes' AND ho.er_ed_visit IS NULL AND ho.ofc_visit = 'Y')
OR (h.er_ed_visit = 'No' AND h.ofc_visit = 'No' AND ho.er_ed_visit IS NULL AND ho.ofc_visit IS NULL));
SELECT * FROM paciente_e_hospitalizado;
```

Figura 53. Query's efetuadas para povoar a tabela paciente\_e\_hospitalizado e confirmar os resultados obtidos no povoamento.

idhospitalizacao	id_vaers	hospdays
1	916683	NULL
1	916762	NULL
1	916837	NULL
1	916849	NULL
1	916853	NULL
1	916938	NULL
1	917005	NULL
1	917030	NULL
1	917114	NULL
1	917122	2
1	917139	NULL
1	917250	NULL
1	917260	NULL
1	917338	NULL
1	917342	NULL
1	917450	NULL
1	917538	NULL

**Figura 54.** Parte da informação presente na tabela paciente\_e\_hospitalizado.

#### 5.2.9. Tabela sintomas

Nas Figuras 55 e 56 estão representadas todas as *query's* efetuadas para povoar a tabela **sintomas**. Pela análise das figuras, verifica-se que se recorreu ao comando INSERTO INTO para colocar na tabela **sintomas** os SYMPTOM1, SYMPTOM2, SYMPTOM3, SYMPTOM4 e SYMPTOM5 da tabela **temp\_dos\_sintomas**. Como referido anteriormente, a distinção dos sintomas por versão (1 a 5) não foi considerada, sendo que o único requisito para o povoamento da tabela **sintomas** foi enumerar todas as designações de sintomas distintos. Foram feitos tantos INSERT INTO quanto o número de colunas da tabela **temp\_dos\_sintomas** para que, após o primeiro INSERT INTO, fosse possível verificar os valores já inseridos na tabela sintomas, e só acrescentar os não existentes. Para isso, foi utilizado o comando SELECT DISTINCT e WHERE...NOT IN. Como havia colunas da tabela **temp\_dos\_sintomas** que continham valores nulos também foi necessário utilizar o comando IS NOT NULL.

Posteriormente, recorrendo às *query* SELECT COUNT (\*) FROM **sintomas**; e SELECT COUNT(DISTINCT(designacao)) FROM **sintomas**; verificou-se que o povoamento tinha sido corretamente efetuado (Figura 57). Finalmente, recorreu-se à *query* SELECT \* FROM **sintomas**; para confirmar que o povoamento estaria correto.

```
-- Tabela sintomas
INSERT INTO sintomas (designacao)
SELECT DISTINCT SYMPTOM1 FROM trabalho_covid.temp_dos_sintomas
WHERE SYMPTOM1 IS NOT NULL;

INSERT INTO sintomas (designacao)
SELECT DISTINCT SYMPTOM2 FROM trabalho_covid.temp_dos_sintomas
WHERE SYMPTOM2 NOT IN (SELECT designacao FROM sintomas)
AND SYMPTOM2 IS NOT NULL;

INSERT INTO sintomas (designacao)
SELECT DISTINCT SYMPTOM3 FROM trabalho_covid.temp_dos_sintomas
WHERE SYMPTOM3 NOT IN (SELECT designacao FROM sintomas)
AND SYMPTOM3 IS NOT NULL;

INSERT INTO sintomas (designacao)
SELECT DISTINCT SYMPTOM4 FROM trabalho_covid.temp_dos_sintomas
WHERE SYMPTOM4 NOT IN (SELECT designacao FROM sintomas)
AND SYMPTOM4 IS NOT NULL;
```

**Figura 55.** Query's efetuadas para povoar a tabela sintomas e confirmar os resultados obtidos no povoamento.

```
INSERT INTO sintomas (designacao)
SELECT DISTINCT SYMPTOM5 FROM trabalho_covid.temp_dos_sintomas
WHERE SYMPTOM5 NOT IN (SELECT designacao FROM sintomas)
AND SYMPTOM5 IS NOT NULL;

-- Confirmação final de que as scripts implementaram corretamente
SELECT COUNT(*) FROM sintomas;
select COUNT(DISTINCT(designacao)) from sintomas;
SELECT * FROM sintomas;
```

**Figura 56.** Query's efetuadas para povoar a tabela sintomas e confirmar os resultados obtidos no povoamento (Continuação).

idsintoma	designacao
1	Dysphagia
2	Anxiety
3	Chest discomfort
4	Dizziness
5	Injection site erythema
6	Pharyngeal swelling
7	Abdominal pain
8	Diarrhoea
9	Vaccination site erythema
10	Rash
11	Blood pressure decreased
12	Dyspnoea
13	Heart rate decreased
14	Nausea
15	SARS-CoV-2 antibody test
16	Abdominal pain

**Figura 57.** Parte da informação presente na tabela sintomas.

### 5.2.10. Tabela sintomas\_afeta\_paciente

Na Figura 58 estão representadas todas as *query's* efetuadas para povoar a tabela **sintomas\_afeta\_paciente**. Pela análise da figura, verifica-se que se recorreu ao comando INSERTO INTO para colocar na tabela **sintomas\_afeta\_paciente** os idsintoma da tabela **sintomas**, os id\_vaers da tabela **paciente**, os ONSET\_DATE e NUMDAYS da tabela **temp\_dos\_sintomas2** e os RECOVD da tabela **temp\_da\_doenca**, em que os VAERS\_ID das tabelas **temp\_dos\_sintomas**, **temp\_dos\_sintomas2** e **temp\_da\_doenca** são iguais aos id\_vaers da tabela **paciente** e em que para cada designacao da tabela **sintomas**, as colunas SYMPTOM1, SYMPTOM2, SYMPTOM3, SYMPTOM4 ou SYMPTOM5 da tabela **temp\_dos\_sintomas** apresentam a mesma designação, ou seja, o paciente apresenta um determinado sintoma. É de realçar que nesta condição se usou o operador OR entre as várias designações, uma vez que o povoamento é para ser feito quando o paciente tiver pelo 1 sintoma.

Posteriormente, recorrendo à *query* SELECT \* FROM **sintomas\_afeta\_paciente**; verificou-se que o povoamento tinha sido corretamente efetuado (Figura 59).

```
-- Tabela sintomas_afeta_paciente
INSERT INTO sintomas_afeta_paciente (idsintoma, id_vaers, onset_date, numdays, recovd)
SELECT s.idsintoma, p.id_vaers, s2.ONSET_DATE, s2.NUMDAYS, d.RECOVD
FROM sintomas s, paciente p, trabalho_covid.temp_dos_sintomas si, trabalho_covid.temp_dos_sintomas2 s2,
trabalho_covid.temp_da_doenca d
WHERE si.VAERS_ID = p.id_vaers
AND (s.designacao = si.SYMPTOM1
OR s.designacao = si.SYMPTOM2
OR s.designacao = si.SYMPTOM3
OR s.designacao = si.SYMPTOM4
OR s.designacao = si.SYMPTOM5)
AND p.id_vaers = s2.VAERS_ID
AND p.id_vaers = d.VAERS_ID;
SELECT * FROM sintomas_afeta_paciente;
```

**Figura 58.** *Query's* efetuadas para povoar a tabela sintomas\_afeta\_paciente e confirmar os resultados obtidos no povoamento.

idsintoma	id_vaers	onset_date	numdays	recovd
1	916600	2020-12-30	2	Y
1	916602	2020-12-31	0	NULL
1	916720	2020-12-31	0	Y
1	916790	2020-12-26	0	NULL
1	916804	2020-12-30	0	Y
1	917026	2020-12-29	1	N
1	917168	2020-12-31	0	Y
1	917273	2020-12-31	0	Y
1	917338	2020-12-31	0	Y
1	917430	2020-12-24	1	N
1	917576	2021-01-02	0	Y
1	917618	2021-01-02	0	N
1	917771	2021-01-02	2	NULL
1	918008	2020-12-29	0	Y
1	918050	2020-12-31	0	N
1	918226	2020-12-30	0	Y

Figura 59. Parte da informação presente na tabela `sintomas_afeta_paciente`.

#### 5.2.11. Tabela `vacina_causa_sintomas`

Na Figura 60 estão representadas todas as *query's* efetuadas para povoar a tabela **`vacina_causa_sintomas`**. Pela análise da figura, verifica-se que se recorreu ao comando `INSERTO INTO` para colocar na tabela **`vacina_causa_sintomas`** os `idvacina` da tabela **`vacinar`**, os `idsintomas` da tabela **`sintomas_afeta_paciente`**, em que os `id_vaers` das tabelas **`sintomas_afeta_paciente`** e **`vacinar`** são iguais.

Posteriormente, recorrendo à *query* `SELECT * FROM vacina_causa_sintomas;` verificou-se que o povoamento tinha sido corretamente efetuado (Figura 61).

```
-- Tabela vacina_causa_sintomas
INSERT INTO vacina_causa_sintomas (idvacina, idsintoma)
SELECT v.idvacina,sa.idsintoma FROM sintomas_afeta_paciente sa, vacinar v
WHERE v.id_vaers=sa.id_vaers;
SELECT * FROM vacina_causa_sintomas;
```

Figura 60. *Query's* efetuadas para povoar a tabela `sintomas_afeta_paciente` e confirmar os resultados obtidos no povoamento.



idvacina	idsintoma
1	1
3	1
80	1
238	1
273	1
442	1
447	1
482	1
541	1
700	1
1091	1
1112	1
1877	1
1938	1
2258	1
...	...

**Figura 61.** Parte da tabela vacina\_causa\_sintomas.

## 6. Resultados e Discussão

### 6.1. *Query's*, funções e procedimentos

Após se ter efetuado o povoamento da base de dados, é agora possível efetuar comandos SQL que permitam analisar a toda a informação e tirar conclusões sobre as reações adversas às vacinas da COVID-19.

De forma a facilitar a análise, subdividiram-se as *query's* efetuadas em subsecções.

#### 6.1.1. Análise da amostra

Antes de se efetuarem *query's* para analisar as reações adversas às vacinas da COVID-19, foram previamente realizados alguns comandos SQL para analisar a amostra em estudo. Na Figura 62 está representada a *query* efetuada para determinar o número de pacientes na amostra, em que se utilizou o comando SELECT. Como se pode observar na Figura 63, a amostra tem 32622 pacientes.

```
-- 1. Número de pacientes da amostra
SELECT COUNT(*) FROM paciente;
```

**Figura 62.** Query efetuada para determinar o número de pacientes da amostra.

COUNT(*)
32622

**Figura 63.** Número de pacientes da amostra.

De seguida, determinou-se a percentagem de mulheres e de homens na amostra, recorrendo ao comando SELECT, para apresentar os resultados da razão entre o número de mulheres e a população total do estudo. O mesmo raciocínio aplicou-se para os homens (Figura 64). Pela observação das Figuras 65 e 66 é possível verificar que a amostra é maioritariamente constituída por mulheres (cerca de 73%).

```
-- 2. Sexo predominante
SELECT ((SELECT COUNT(*) FROM paciente WHERE sex = 'F')/COUNT(*)*100 AS percentagem_mulheres FROM paciente;
SELECT ((SELECT COUNT(*) FROM paciente WHERE sex = 'M')/COUNT(*)*100 AS percentagem_homens FROM paciente;
```

**Figura 64.** Query's efetuadas para a percentagem de mulheres e homens na amostra.

percentagem_mulheres
72.8833

**Figura 65.** Percentagem de mulheres na amostra.

percentagem_homens
25.7188

**Figura 66.** Percentagem de homens na amostra.

Posteriormente, determinou-se a percentagem de cada faixa etária na amostra. Para tal foi desenvolvido o *procedure* PERCENTAGEM\_GRUPO\_PACIENTES (Figura 67) que tem como parâmetros de entrada menor e maior, ambos INT, e retorna a variável percentagem FLOAT (inicialmente declarada como NULL). Quando se pretende determinar a percentagem de bebés presentes na amostra (idade inferior a 1 ano), o *procedure* executa a 1ª condição do IF, em que o maior é NULL, pois pretende-se determinar a idade até ao 1 ano. Para crianças (1-12 anos), adolescentes (12-16 anos), jovens (16-25) e adultos (25-65), o *procedure* executa a 2ª condição do IF, em que menor e maior são ambos NOT NULL. Por último, para idosos, é executada a 3ª condição do IF, com menor NULL (Figura 68). Cada vez que o *procedure* é executado, este atualiza (através do comando SET) a variável percentagem, igualando-a SELECT da razão do número de pacientes com determinada faixa etária pelo total de pacientes.

Nas Figuras 69 a 74 estão representados os resultados dos *call's* do *procedure*, que “chama” e executa o *procedure*, tendo em conta os argumentos de entrada dados. Verifica-se que a amostra é maioritariamente constituída por adultos (cerca de 66%) e idosos (cerca de 23%).

```
-- 3. Percentagem de bebés, crianças, adolescentes, jovens, adultos e idosos
DELIMITER $$
CREATE PROCEDURE PERCENTAGEM_GRUPO_PACIENTES(menor INT, maior INT)
BEGIN
    DECLARE percentagem FLOAT DEFAULT NULL;

    IF (menor IS NOT NULL AND maior IS NULL)
        THEN SET percentagem = (SELECT ((SELECT COUNT(*) FROM paciente WHERE age_yrs <= menor)/COUNT(*)*100 FROM paciente);

    ELSEIF (menor IS NOT NULL AND maior IS NOT NULL)
        THEN SET percentagem = (SELECT ((SELECT COUNT(*) FROM paciente WHERE age_yrs > menor AND age_yrs <= maior)/COUNT(*)*100 FROM paciente);

    ELSEIF (menor IS NULL AND maior IS NOT NULL)
        THEN SET percentagem = (SELECT ((SELECT COUNT(*) FROM paciente WHERE age_yrs > maior)/COUNT(*)*100 FROM paciente);
    END IF;
SELECT percentagem;
END $$
DELIMITER ;
```

Figura 67. Procedure PERCENTAGEM\_GRUPO\_PACIENTES.

```
-- Percentagem de bebés
CALL PERCENTAGEM_GRUPO_PACIENTES(1, NULL);

-- Percentagem de crianças
CALL PERCENTAGEM_GRUPO_PACIENTES(1, 12);

-- Percentagem de adolescentes
CALL PERCENTAGEM_GRUPO_PACIENTES(12, 16);

-- Percentagem de jovens
CALL PERCENTAGEM_GRUPO_PACIENTES(16, 25);

-- Percentagem de adultos
CALL PERCENTAGEM_GRUPO_PACIENTES(25, 65);

-- Percentagem de idosos
CALL PERCENTAGEM_GRUPO_PACIENTES(NULL, 65);
```

Figura 68. Call's do Procedure PERCENTAGEM\_GRUPO\_PACIENTES.

percentagem
0.141009

Figura 69. Percentagem de bebés na amostra.

percentagem
0.0337195

**Figura 70.** Percentagem de crianças na amostra.

percentagem
0.217644

**Figura 71.** Percentagem de adolescentes na amostra.

percentagem
4.97824

**Figura 72.** Percentagem de jovens na amostra.

percentagem
65.8727

**Figura 73.** Percentagem de adultos na amostra.

percentagem
22.681

**Figura 74.** Percentagem de idosos na amostra.

Por último, determinaram-se os 3 estados americanos mais frequentes na amostra, uma vez, que o *dataset* apenas utiliza informações de pacientes americanos (Figura 75). Para tal, recorreu-se ao comando SELECT do estado da tabela **paciente** e da percentagem de pacientes de cada estado agrupando os dados (através do GROUP BY) pelo estado e ordenando os 3 estados por ordem decrescente segundo a percentagem de pacientes (com os comandos ORDER BY DESC LIMIT 3). Na Figura 76 está representado o resultado da *query* efetuada, em que se pode verificar que os estados mais frequentes são a Califórnia, o Texas e Nova Iorque.

```
-- 4. 3 estados americanos mais frequentes na amostra
SELECT state, (COUNT(id_vaers)/(SELECT COUNT(id_vaers) FROM paciente))*100 AS percentagem_pacientes FROM paciente
WHERE state IS NOT NULL
GROUP BY state ORDER BY percentagem_pacientes DESC LIMIT 3;
```

**Figura 75.** Query efetuada para determinar os estados americanos mais frequentes.

state	percentagem_pacientes
CA	7.7003
TX	5.4135
NY	5.2940

**Figura 76.** 3 estados americanos mais frequentes e respetiva percentagem de pacientes.

### 6.1.2. Indicadores de saúde

Após se ter analisado a amostra, foram determinados alguns indicadores de saúde. Primeiramente, determinou-se a percentagem de pacientes que tiveram sintomas, através do quociente entre o número de pacientes diferentes que ocorrem na tabela **sintomas\_afeta\_paciente** e o número de pacientes total da amostra (Figura 77). Pela análise da Figura 78, verifica-se que todos os pacientes vacinados foram sintomáticos.

```
-- 1. Percentagem de pacientes que tiveram sintomas
SELECT ((SELECT COUNT(DISTINCT(id_vaers)) FROM sintomas_afeta_paciente)/COUNT(*))*100 AS percentagem_sintomas FROM paciente p;
```

**Figura 77.** Query efetuada para determinar a percentagem de pacientes sintomáticos.

percentagem_sintomas
100.0000

**Figura 78.** 3 Percentagem de pacientes sintomáticos.

De seguida, determinaram-se os 10 sintomas mais frequentes associados à vacinação. Pela análise da Figura 79, verifica-se que se recorreu ao comando SELECT do iddsintoma e designacao

da tabela **sintomas** e da percentagem das pessoas que tinham esses sintomas. Para determinar esta percentagem, procedeu-se à razão do número (COUNT) de pacientes que tinham um determinado sintoma (*query* principal), pelo total de pacientes que tinham pelo menos 1 sintoma (*subquery*).

Na *subquery* foi selecionado o número de id\_vaers da tabela **vacinar**, em que os id\_vaers da tabela **vacinar** e da tabela **paciente** são iguais. Quanto à *query* foram postas as seguintes condições: os idvacina da tabela **vacina** e da tabela **vacinar** são iguais, os idvacina da tabela **vacina** e da tabela **vacina\_causa\_sintomas** são iguais e os idsintoma da tabela **sintomas** e da tabela **vacina\_causa\_sintomas** são iguais. Os resultados foram agrupados com o comando GROUP BY designacao e ordenados de forma decrescente consoante o valor da respetiva percentagem\_pacientes, com um total de 10 sintomas diferentes.

Na Figura 80 está representado o resultado do comando apresentado na Figura 79, em que se pode observar que os sintomas mais frequentes são dor de cabeça (cerca de 22%), pirexia ou febre (cerca de 18%) e arrepios (cerca de 17%).

```
-- 3. 10 Sintomas mais frequentes associados à vacinação
SELECT s.idsintoma, s.designacao, (COUNT(vr.id_vaers)/(SELECT COUNT(vr.id_vaers) FROM vacinar vr, paciente p
WHERE p.id_vaers = vr.id_vaers))*100 AS percentagem_pacientes
FROM vacina v, vacinar vr, vacina_causa_sintomas vcs, sintomas s
WHERE v.idvacina = vr.idvacina
AND v.idvacina = vcs.idvacina
AND s.idsintoma = vcs.idsintoma
GROUP BY s.designacao
ORDER BY percentagem_pacientes DESC
LIMIT 10;
```

**Figura 79.** Query efetuada para determinar os sintomas mais frequentes associados à vacinação.

idsintoma	designacao	percentagem_pacientes
21	Headache	21.7176
37	Pyrexia	17.4160
18	Chills	16.6539
58	Fatigue	15.8253
59	Pain	14.7205
14	Nausea	12.6694
4	Dizziness	10.5263
27	Pain in extremity	9.4011
19	Injection site pain	8.7617
52	Myalgia	8.4292

**Figura 80.** Top 10 dos sintomas mais frequentes associados à vacinação.

Posteriormente, determinou-se a percentagem de pacientes que ficaram hospitalizados. Pela análise da Figura 81, verifica-se que se recorreu ao comando SELECT da razão do número

(COUNT) de pacientes que estiveram hospitalizados (*subquery*), pelo total de pacientes (*query* principal). Na *subquery* foi selecionado o número de id\_vaers da tabela **paciente**, em que os id\_vaers da tabela **paciente** e da tabela **paciente\_e\_hospitalizado** são iguais e os idhospitalizacao da tabela **paciente\_e\_hospitalizado** e da tabela **hospitalizacao** são iguais.

Na Figura 82 está representado o resultado do comando apresentado na Figura 81, em que se pode observar que apenas aproximadamente 9% dos pacientes foram hospitalizados.

```
-- 4. Percentagem de pacientes que ficaram hospitalizados
SELECT ((SELECT COUNT(DISTINCT(p.id_vaers)) FROM paciente p, paciente_e_hospitalizado ph, hospitalizacao h
WHERE p.id_vaers = ph.id_vaers
AND h.idhospitalizacao = ph.idhospitalizacao
AND ph.hospdays IS NOT NULL)/count(*))*100 AS percentagem_hospitalizacoes FROM paciente p;
```

**Figura 81.** Query efetuada para determinar a percentagem de pacientes hospitalizados.

percentagem_hospitalizacoes
8.6935

**Figura 82.** Percentagem de pacientes hospitalizados.

De seguida, determinou-se o top 10 dos dias de hospitalização. Pela análise da Figura 83, verifica-se que se recorreu ao comando SELECT do número de dias em que os pacientes estiveram hospitalizados da tabela **paciente\_e\_hospitalizado**, ordenando os resultados por ordem decrescente e limitado a 10 linhas (ORDER BY DESC LIMIT 10). Na Figura 84 está representado o resultado do comando apresentado na Figura 83, em que se pode observar que no máximo, os pacientes tiveram hospitalizados 39 dias.

```
-- 5. Top 10 dos dias que as pessoas ficaram hospitalizadas
SELECT hp.hospdays
FROM paciente p, hospitalizacao h, paciente_e_hospitalizado hp
WHERE p.id_vaers = hp.id_Vaers
AND hp.idhospitalizacao = h.idhospitalizacao
ORDER BY hp.hospdays DESC
LIMIT 10;
```

**Figura 83.** Query efetuada para determinar o top 10 dos dias em que as pessoas ficaram hospitalizadas.

hospdays
39
38
37
36
36
33
30
29
28
28

**Figura 84.** Top 10 dos dias em que as pessoas ficaram hospitalizadas.

Posteriormente, determinou-se a percentagem de pacientes que morreram, através da razão dos pacientes que morreram (data\_died não nulo) pelo total de pacientes (Figura 85). Pela observação da Figura 86 verifica-se que cerca de 5.5% dos pacientes faleceram, o que é bastante considerável, pois, mais de 5 a cada 100 pacientes morreram.

```
-- 6. Percentagem de pacientes que faleceram
SELECT ((SELECT COUNT(*) FROM paciente WHERE data_died IS NOT NULL)/COUNT(*)*100 AS percentagem_mortes FROM paciente;
```

**Figura 85.** Query efetuada para determinar a percentagem de pacientes que morreram.

percentagem_mortes
5.4718

**Figura 86.** Percentagem de pacientes que morreram.

De seguida, determinou-se a percentagem de pacientes que têm patologias associadas. Pela análise da Figura 87, verifica-se que se recorreu ao comando SELECT da razão do número (COUNT) de pacientes que têm patologias (*subquery*), pelo total de pacientes (*query* principal). Na *subquery* foi selecionado o número de id\_vaers da tabela **doenca\_afeta\_paciente**, em que os id\_vaers da tabela **paciente** e da tabela **paciente\_e\_hospitalizado** são iguais e os iddoenca da tabela **doenca\_afeta\_paciente** e da tabela **historico\_clinico** são iguais. Pela observação da Figura 88 verifica-se que cerca de 26% dos pacientes tinham pelo menos 1 patologia associada.

```
-- 8. Percentagem de pacientes que têm doenças
SELECT ((SELECT COUNT(DISTINCT(d.id_vaers)) FROM paciente p, doenca_afeta_paciente d, historico_clinico h
WHERE p.id_vaers = d.id_vaers AND h.iddoenca = d.iddoenca)/COUNT(*)*100 AS percentagem_doencas FROM paciente p;
```

**Figura 87.** Query efetuada para determinar a percentagem de pacientes que têm doenças.



percentagem_doencas
25.7710

**Figura 88.** Percentagem de pacientes que têm doenças.

Por último, determinou-se o top 3 das patologias mais frequentes. Pela análise da Figura 89, verifica-se que se recorreu ao comando SELECT da designação da patologia da tabela **historico\_clinico**, ordenando os resultados por ordem decrescente e limitado a 3 linhas (ORDER BY DESC LIMIT 3). Na Figura 90 está representado o resultado do comando apresentado na Figura 89, em que se pode observar que cerca de 12% dos pacientes apresentavam hipertensão, hiperlipidemia e/ou asma.

```
-- 7. Top 3 das patologias mais frequentes
SELECT DISTINCT h.designacao, (COUNT(p.id_vaers)/(SELECT COUNT(vr.id_vaers) FROM vacinar vr, paciente p
WHERE p.id_vaers = vr.id_vaers))*100 AS percentagem_pacientes
FROM paciente p, historico_clinico h, doenca_afeta_paciente d
WHERE p.id_vaers = d.id_vaers
AND d.iddoenca = h.iddoenca
GROUP BY h.designacao
ORDER BY percentagem_pacientes DESC
LIMIT 3;
```

**Figura 89.** Query efetuada para determinar o top 3 das patologias mais frequentes.

designacao	percentagem_pacientes
Hypertension	11.9994
Hyperlipidemia	11.9994
Asthma	11.9534

**Figura 90.** Top 3 das patologias mais frequentes.

Resumindo, com estes indicadores de saúde, é possível concluir que todos os pacientes foram sintomáticos, apresentando maioritariamente dor de cabeça, febre e arrepios. Cerca de 9% dos pacientes ficaram hospitalizados e mais de 5% dos pacientes morreram. Cerca de 26% dos indivíduos apresenta pelo menos uma patologia, sendo hipertensão, hiperlipidemia e asma as mais comuns.

### 6.1.3. Indicadores das vacinas

Assim como se determinaram alguns indicadores de saúde, determinaram-se também 2 indicadores relativos às vacinas. Primeiramente, determinaram-se as 3 marcas de vacina mais inoculadas e a percentagem de vacinas inoculadas de cada marca. Pela análise da Figura 91, verifica-se que se recorreu ao comando SELECT do vax\_manu da tabela **fornecedor** e da

percentagem de vacinas inoculadas por marca de vacina. Para determinar esta percentagem, procedeu-se à razão do número (COUNT) de vacinas inoculadas de uma dada marca (*query* principal), pelo total de vacinas (*subquery*).

Na *subquery* foi selecionado o número de idvacina da tabela **vacinar**, em que os idvacina da tabela **vacinar** e da tabela **vacina** e em que os idfornecedor da tabela **fornecedor** e da tabela **vacina** são iguais. Quanto à *query* foram impostas as mesmas condições que na *subquery*, com o comando GROUP BY vax\_manu as vacinas foram agrupadas por marca e através do ORDER BY, as marcas de vacina foram ordenadas de forma decrescente consoante o valor da respetiva percentagem\_vacinas\_inoculadas, com um total de 3 marcas diferentes.

Na Figura 92 está representado o resultado do comando apresentado na Figura 91, em que se pode observar que das vacinas inoculadas, aproximadamente metade foram da Moderna e as restantes foram da Pfizer (cerca de 48%) e da JANSSEN (cerca de 3%).

```
-- 1. 3 marcas de vacina mais inoculadas
SELECT f.vax_manu, (COUNT(vr.idvacina)/(SELECT COUNT(vr.idvacina)
FROM fornecedor f, vacina v, vacinar vr
WHERE f.idfornecedor = v.idfornecedor
AND v.idvacina = vr.idvacina))*100 AS percentagem_vacinas_inoculadas
FROM fornecedor f, vacina v, vacinar vr
WHERE f.idfornecedor = v.idfornecedor
AND v.idvacina = vr.idvacina
GROUP BY f.vax_manu
ORDER BY percentagem_vacinas_inoculadas DESC
LIMIT 3;
```

**Figura 91.** Query efetuada para determinar a percentagem de inoculação por marca de vacina.

vax_manu	percentagem_vacinas_inoculadas
MODERA	49.4246
PFIZER\BIOTECH	47.8390
JASSE	2.5830

**Figura 92.** Percentagem de inoculação por marca de vacina.

Posteriormente, determinaram-se os 3 lotes de vacinas mais frequentes, respetivos fornecedores desses lotes a percentagem de vacinas inoculadas desses lotes. Pela análise da Figura 93, verifica-se que se recorreu ao comando SELECT do vax\_manu e do vax\_lot da tabela **fornecedor** e da percentagem de vacinas inoculadas por lote de vacina. Para determinar esta percentagem, procedeu-se à razão do número (COUNT) de vacinas inoculadas de um dado lote (*query* principal), pelo total de lotes de vacinas (*subquery*).

Na *subquery* foi selecionado o número de idvacina da tabela **vacinar**, em que os idvacina da tabela **vacinar** e da tabela **vacina** e em que os idfornecedor da tabela **fornecedor** e da tabela **vacina** são iguais. Quanto à *query* foram impostas as mesmas condições que na *subquery*, e ainda, com o comando GROUP BY vax\_lot foram agrupadas as vacinas por lote e através do ORDER BY, os lotes foram ordenados de forma decrescente consoante o valor da respetiva percentagem\_vacinas\_inoculadas, com um total de 3 lotes diferentes.

Na Figura 94 está representado o resultado do comando apresentado na Figura 93, em que se pode observar que os dois lotes, com maior percentagem de vacinas inoculadas, são da Moderna. O 3º lote de vacina mais frequente é da Pfizer, com cerca de 3% de vacinas inoculadas desse lote.

```
-- 2. 3 lotes de vacinas mais frequentes e respetivos fornecedores desses lotes
SELECT f.vax_manu, v.vax_lot, (COUNT(vr.idvacina)/(SELECT COUNT(vr.idvacina)
FROM fornecedor f, vacina v, vacinar vr
WHERE f.idfornecedor = v.idfornecedor
AND v.idvacina = vr.idvacina))*100 AS percentagem_vacinas_inoculadas
FROM fornecedor f, vacina v, vacinar vr
WHERE f.idfornecedor = v.idfornecedor
AND v.idvacina = vr.idvacina
AND v.vax_lot IS NOT NULL
GROUP BY v.vax_lot
ORDER BY percentagem_vacinas_inoculadas DESC
LIMIT 3;
```

**Figura 93.** Query efetuada para determinar a percentagem de inoculação das vacinas provenientes dos lotes mais frequentes.

vax_manu	vax_lot	percentagem_vacinas_inoculadas
MODERA	039k20a	4.6903
MODERA	011J20A	3.6264
PFIZER/BIOTECH	EK9231	3.3195

**Figura 94.** Percentagem de inoculação das vacinas provenientes dos lotes mais frequentes.

Assim, com a análise dos indicadores de vacinas, é possível concluir que a Moderna é a marca de vacina mais inoculada nesta amostra de pacientes com cerca de metade das inoculações, seguindo-se a Pfizer com aproximadamente 48% de inoculações. Da mesma forma, os 2 lotes de vacinas mais frequentes são da Moderna.

#### 6.1.4. Indicadores da vacinação

Assim como se determinaram alguns indicadores de saúde e de vacinas, determinaram-se também indicadores relativos à vacinação. Primeiramente, determinou-se o número de doses inoculadas mais frequente nos pacientes e a que marca pertencem. Pela análise da Figura 95, verifica-se que se recorreu ao comando SELECT do vax\_manu da tabela **fornecedor**, do vax\_dose\_series da tabela **vacinar** e da percentagem de vacinas inoculadas por doses de vacina. Para determinar esta percentagem, procedeu-se à razão do número (COUNT) de vacinas inoculadas com uma determinada dose (*query* principal), pelo total de doses de vacinas inoculadas(*subquery*).

Na *subquery* foi selecionado o número de idvacina da tabela **vacinar**, em que os idvacina da tabela **vacinar** e da tabela **vacina** e em que os idfornecedor da tabela **fornecedor** e da tabela **vacina** são iguais. Quanto à *query* foram impostas as mesmas condições que na *subquery* e ainda, com o comando GROUP BY vax\_dose\_series foram agrupadas as vacinas por número de doses e através do ORDER BY, as doses de vacina foram ordenadas de forma decrescente consoante o valor da respetiva percentagem\_vacinas\_inoculadas, com um total de 3 doses diferentes.

Na Figura 95 está representado o resultado do comando apresentado na Figura 94, em que se pode observar que das vacinas inoculadas, aproximadamente 79% foram da 1ª dose, 21% da 2ª dose e da marca Moderna.

```
-- 1. Número de doses de vacinas mais frequente e de que vacina
SELECT f.vax_manu, vr.vax_dose_series, (COUNT(vr.idvacina)/(SELECT COUNT(vr.idvacina)
FROM fornecedor f, vacina v, vacinar vr
WHERE f.idfornecedor = v.idfornecedor
AND v.idvacina = vr.idvacina
AND vr.vax_dose_series IS NOT NULL))*100 AS percentagem_vacinas_inoculadas
FROM fornecedor f, vacina v, vacinar vr
WHERE f.idfornecedor = v.idfornecedor
AND v.idvacina = vr.idvacina
AND vr.vax_dose_series IS NOT NULL
GROUP BY vr.vax_dose_series
ORDER BY percentagem_vacinas_inoculadas DESC
LIMIT 3;
```

**Figura 95.** Query efetuada para determinar a percentagem de inoculação por número de doses de vacina.

vax_manu	vax_dose_series	percentagem_vacinas_inoculadas
MODERA	1	78.7077
MODERA	2	21.0878
MODERA	3	0.1143

**Figura 96.** Percentagem de inoculação por número de doses de vacina.

Posteriormente, determinaram-se as 3 formas mais comuns de aplicar a vacina e a percentagem de vacinas inoculadas para essas formas. Pela análise da Figura 97, verifica-se que se recorreu ao comando SELECT do vax\_route da tabela **vacinar** e da percentagem de vacinas inoculadas por forma de inoculação. Para determinar esta percentagem, procedeu-se à razão do número (COUNT) de vacinas inoculadas de uma dada forma (*query* principal), pelo total de formas de aplicação de vacinas (*subquery*).

Na *subquery* foi selecionado o número de idvacina da tabela **vacinar**, em que os idvacina da tabela **vacinar** e da tabela **vacina** são iguais e o vax\_route não é nulo. Quanto à *query* foram impostas as mesmas condições que na *subquery*, com o comando GROUP BY vax\_route foram agrupadas as vacinas por forma de aplicação e através do ORDER BY, as vacinas foram ordenadas de forma decrescente consoante o valor da respetiva percentagem\_vacinas\_inoculadas, com um total de 3 tipos de vacinas de inoculações diferentes.

Na Figura 98 está representado o resultado do comando apresentado na Figura 97, em que se pode observar cerca de 76% das vacinas foram administradas via intramuscular, aproximadamente 15% foram administradas através de uma agulha e quase 9% foram administradas por outras vias.

```
-- 2. 3 formas mais comuns de aplicar a vacina
SELECT vr.vax_route, (COUNT(vr.idvacina)/(SELECT COUNT(vr.idvacina)
FROM vacina v, vacinar vr
WHERE v.idvacina = vr.idvacina
AND vr.vax_route IS NOT NULL))*100 AS percentagem_vacinas_inoculadas
FROM vacina v, vacinar vr
WHERE v.idvacina = vr.idvacina
AND vr.vax_route IS NOT NULL
GROUP BY vr.vax_route
ORDER BY percentagem_vacinas_inoculadas DESC
LIMIT 3;
```

**Figura 97.** Query efetuada para determinar a percentagem de inoculação das vacinas por forma de inoculação.

vax_route	percentagem_vacinas_inoculadas
IM	75.4617
SYR	15.3968
OT	8.4928

**Figura 98.** Percentagem de inoculação das vacinas por forma de inoculação.

De seguida, determinaram-se os 3 locais mais comuns de injeção da vacina no corpo. Pela análise da Figura 99, verifica-se que se recorreu ao comando SELECT do vax\_site da tabela **vacinar** e da percentagem de vacinas inoculadas por local de injeção. Para determinar esta percentagem, procedeu-se à razão do número (COUNT) de vacinas inoculadas num determinado local (*query* principal), pelo total de locais de injeção de vacinas (*subquery*).

Na *subquery* foi selecionado o número de idvacina da tabela **vacinar**, em que os idvacina da tabela **vacinar** e da tabela **vacina** são iguais e vax\_site não é nulo. Quanto à *query* foram impostas as mesmas condições que na *subquery* e ainda, com o comando GROUP BY vax\_site foram agrupadas as vacinas pelo local de injeção e através do ORDER BY, os locais de injeção foram ordenados de forma decrescente consoante o valor da respetiva percentagem\_vacinas\_inoculadas, com um total de 3 locais de injeção diferentes.

Na Figura 100 está representado o resultado do comando apresentado na Figura 99, em que se pode observar que das vacinas inoculadas, aproximadamente 72% foram aplicadas no braço esquerdo, cerca de 28% foram aplicadas no braço direito e uma minoria foi aplicada noutras partes do corpo.

```
-- 3. 3 locais mais comuns onde foram aplicadas as vacinas no corpo
SELECT vr.vax_site,(COUNT(vr.idvacina) /(SELECT COUNT(vr.idvacina)
FROM vacina v, vacinar vr
WHERE v.idvacina = vr.idvacina
AND vr.vax_site IS NOT NULL))*100 AS percentagem_vacinas_inoculadas
FROM vacina v, vacinar vr
WHERE v.idvacina = vr.idvacina
AND vr.vax_site IS NOT NULL
GROUP BY vr.vax_site
ORDER BY percentagem_vacinas_inoculadas DESC
LIMIT 3;
```

**Figura 99.** Query efetuada para determinar a percentagem de inoculação por local de injeção da vacina.

vax_site	percentagem_vacinas_inoculadas
LA	71.7942
RA	28.1187
OT	0.0469

**Figura 100.** Percentagem de inoculação por local de injeção da vacina.

Posteriormente, determinaram-se os 3 setores mais comuns onde foram aplicadas as vacinas e a percentagem de vacinas inoculadas nesses setores. Pela análise da Figura 101, verifica-se que se recorreu ao comando SELECT do v\_adminby da tabela **vacinar** e da percentagem de vacinas inoculadas por setor. Para determinar esta percentagem, procedeu-se à razão do número (COUNT) de vacinas inoculadas num dado setor (*query* principal), pelo total de setores (*subquery*).

Na *subquery* foi selecionado o número de idvacina da tabela **vacinar**, em que os idvacina da tabela **vacinar** e da tabela **vacina** são iguais e o v\_adminby não é nulo. Quanto à *query* foram impostas as mesmas condições que na *subquery*, com o comando GROUP BY v\_adminby foram as vacinas foram agrupadas por setor e através do ORDER BY, os setores foram ordenados de forma decrescente consoante o valor da respetiva percentagem\_vacinas\_inoculadas, com um total de 3 setores diferentes.

Na Figura 102 está representado o resultado do comando apresentado na Figura 101, em que se pode observar cerca de 59% das vacinas foram administradas no setor privado, aproximadamente 13% foram administradas a militares e cerca de 12% foram administradas no setor público.

```
-- 4. 3 locais mais comuns onde foram aplicadas as vacinas (instituições)
SELECT vr.v_adminby,(COUNT(vr.idvacina)/(SELECT COUNT(vr.idvacina)
FROM vacina v, vacinar vr
WHERE v.idvacina = vr.idvacina
AND vr.v_adminby IS NOT NULL))*100 AS percentagem_vacinas_inoculadas
FROM vacina v, vacinar vr
WHERE v.idvacina = vr.idvacina
AND vr.v_adminby IS NOT NULL
GROUP BY vr.v_adminby
ORDER BY percentagem_vacinas_inoculadas DESC
LIMIT 3;
```

**Figura 101.** Query efetuada para determinar a percentagem de inoculação das vacinas por setor.

v_adminby	percentagem_vacinas_inoculadas
PVT	58.6945
WRK	12.6133
PUB	12.2705

**Figura 102.** Percentagem de inoculação das vacinas por setor.

Por último, determinaram-se os 3 meses em que ocorreram mais vacinações e a percentagem de vacinas inoculadas nesses meses. Pela análise da Figura 103, verifica-se que se recorreu ao comando SELECT do mês da `vax_date` da tabela **vacinar** e da percentagem de vacinas inoculadas por mês. É de realçar que para se obterem os meses das datas de vacinação, se utilizou a função MONTH do MySQL que retorna o mês de uma data. Para determinar a percentagem, procedeu-se à razão do número (COUNT) de vacinas inoculadas num dado mês (*query principal*), pelo total de vacinas (*subquery*).

Na *subquery* foi selecionado o número de `idvacina` da tabela **vacinar**, em que os `idvacina` da tabela **vacinar** e da tabela **vacina** são iguais. Quanto à *query* foi imposta a mesma condição que na *subquery*, com o comando GROUP BY mes foram agrupadas as vacinas por mês e através do ORDER BY, os meses foram ordenados de forma decrescente consoante o valor da respetiva `percentagem_vacinas_inoculadas`, com um total de 3 meses diferentes.

Na Figura 104 está representado o resultado do comando apresentado na Figura 103, em que se pode observar cerca de 46% das vacinas foram administradas em janeiro de 2021, aproximadamente 34% foram administradas em dezembro de 2021 e cerca de 17% foram administradas em fevereiro de 2021.

```
-- 5. 3 meses em que ocorreram mais vacinações
SELECT MONTH(vr.vax_date) AS mes, (COUNT(vr.idvacina)/(SELECT COUNT(vr.idvacina)
FROM vacina v, vacinar vr
WHERE v.idvacina = vr.idvacina))*100 AS percentagem_vacinas_inoculadas
FROM vacina v, vacinar vr
WHERE v.idvacina = vr.idvacina
GROUP BY mes
ORDER BY percentagem_vacinas_inoculadas DESC
LIMIT 3;
```

**Figura 103.** Query efetuada para determinar a percentagem de inoculação das vacinas por mês.



mes	percentagem_vacinas_inoculadas
1	45.8595
12	33.5379
2	16.9250

**Figura 104.** Percentagem de inoculação das vacinas por mês.

Assim, com a análise dos indicadores de vacinação, é possível concluir a maior parte das vacinas inoculadas na amostra em estudo foram vacinas da 1ª dose, administradas via intramuscular no braço esquerdo dos pacientes. Sabe-se também que os pacientes foram vacinados maioritariamente no setor privado e em janeiro deste ano.

### 6.1.5. Sintomas

Após se ter analisado a amostra e determinados alguns dados estatísticos relevantes, serão agora analisadas as relações que existem entre os sintomas e outras entidades, como a hospitalização, os pacientes, as patologias e as vacinas.

#### 6.1.5.1. Sintomas e hospitalização

Nesta secção, serão averiguados os sintomas apresentados pelos pacientes que foram às urgências e receberam visita de um médico. O procedimento SINTOMAS\_HOSPITALIZACAO, representado na Figura 105, recebe como parâmetro 'Yes' e 'No' consoante o paciente foi às urgências e/ou recebeu visita do médico. Dentro do mesmo, está representada a *query* efetuada para determinar os 10 sintomas mais comuns em pacientes que foram às urgências e/ou receberam visita do médico e a respetiva percentagem de pacientes. Pela análise da figura, verifica-se que se recorreu ao comando SELECT das tabelas **sintomas**, **paciente**, **paciente\_e\_hospitalizado**, **hospitalizacao**, **vacina**, **vacinar** e **vacina\_causa\_sintomas** para se seleccionar a razão entre o número de pacientes que tiveram um determinado sintoma e foram às urgências e/ou receberam visita do médico, pelo número de pacientes que e foram às urgências e/ou receberam visita do médico sem estarem agrupados por sintoma, sendo esta última uma *subquery*. Para isso, foi utilizado o comando COUNT, que devolve um determinado número de linhas em que os idsintoma da tabela **sintoma** e da tabela **vacina\_causa\_sintomas** são iguais, os idvacina das tabelas **vacina** e **vacina\_causa\_sintoma** e das tabelas **vacina** e **vacinar** são iguais, os id\_vaers das tabelas **paciente** e **vacinar** e das tabelas **paciente\_e\_hospitalizado** e

**paciente** são iguais e os **idhospitalizacao** das tabelas **hospitalizacao** e **paciente\_e\_hospitalizado** são iguais. Além disso, verifica-se também que os parâmetros **urgencia** e **medico** recebidos pelo *procedure* são iguais aos atributos **er\_ed\_visit** e **ofc\_visit**, respectivamente, da tabela **hospitalizacao**.

Nas Figuras 107 a 110 estão representados os resultados do *procedure*, após ter sido utilizado o comando **CALL**, apresentado na Figura 106. Pela análise das figuras, verifica-se que para os pacientes que foram às urgências e receberam visita do médico e para os pacientes que foram às urgências, mas não receberam visita do médico, os sintomas são muito parecidos, predominando a dispneia, a dor de cabeça e as tonturas. Relativamente aos pacientes que não foram às urgências, mas receberam visita do médico e aos pacientes que não foram às urgências, nem receberam visita do médico, verifica-se que os sintomas mais frequentes são dor de cabeça, fadiga, dor e febre. Além disso, à medida que se observa sequencialmente as Figuras 107 a 110, verifica-se que as percentagens de pacientes com os sintomas aumentam. Na Figura 107 as percentagens variam entre 1-3% aproximadamente, enquanto na Figura 110, as percentagens variam entre 2-6% aproximadamente. Assim sendo, verifica-se que apesar de muitos dos sintomas serem parecidos entre os grupos, a maior parte dos pacientes optou por não receber ajuda médica, o que poderá indicar uma desvalorização dos sintomas por parte dos pacientes ou ainda um esgotamento dos serviços de saúde e consequente falta de acompanhamento médico.

```
-- Relação sintomas - hospitalizacao
DELIMITER $$
CREATE PROCEDURE SINTOMAS_HOSPITALIZACAO(urgencias VARCHAR (45), medico VARCHAR (45))
BEGIN

    SELECT s.idsintoma AS idsintoma, s.designacao AS designacao, (COUNT(vr.id_vaers)/(SELECT COUNT(vr.id_vaers)
    FROM sintomas s, paciente p, hospitalizacao h, paciente_e_hospitalizado hp, vacina v, vacinar vr, vacina_causa_sintomas vcs
    WHERE s.idsintoma = vcs.idsintoma AND v.idvacina = vcs.idvacina
    AND v.idvacina = vr.idvacina AND vr.id_vaers = p.id_vaers
    AND p.id_vaers = hp.id_vaers AND hp.idhospitalizacao = h.idhospitalizacao
    AND h.er_ed_visit = urgencias AND h.ofc_visit = medico))*100 AS percentagem_pacientes_hospitalizados
    FROM sintomas s, paciente p, hospitalizacao h, paciente_e_hospitalizado hp, vacina v, vacinar vr, vacina_causa_sintomas vcs
    WHERE s.idsintoma = vcs.idsintoma AND v.idvacina = vcs.idvacina AND v.idvacina = vr.idvacina AND vr.id_vaers = p.id_vaers
    AND p.id_vaers = hp.id_vaers AND hp.idhospitalizacao = h.idhospitalizacao
    AND h.er_ed_visit = urgencias AND h.ofc_visit = medico
    GROUP BY s.designacao ORDER BY percentagem_pacientes_hospitalizados DESC LIMIT 10;

END $$
DELIMITER ;
```

**Figura 105.** *Procedure* efetuado para determinar a percentagem de pacientes que foram às urgências e/ou receberam visita do médico por sintoma.

```
-- 1. Quais os sintomas que as pessoas que foram às urgências e receberam visita do médico tinham?
CALL SINTOMAS_HOSPITALIZACAO ('Yes', 'Yes');

-- 2. Quais os sintomas que as pessoas que foram às urgências mas não receberam visita do médico tinham?
CALL SINTOMAS_HOSPITALIZACAO ('Yes', 'No');

-- 3. Quais os sintomas que as pessoas que não foram às urgências mas receberam visita do médico tinham?
CALL SINTOMAS_HOSPITALIZACAO ('No', 'Yes');

-- 4. Quais os sintomas que as pessoas que não foram às urgências nem receberam visita do médico tinham?
CALL SINTOMAS_HOSPITALIZACAO ('No', 'No');
```

**Figura 106.** Call's do procedure SINTOMAS\_HOSPITALIZACAO para as 4 combinações possíveis da tabela hospitalizacao.

idsintoma	designacao	percentagem_pacientes
12	Dyspnoea	2.0262
21	Headache	1.7605
37	Pyrexia	1.6442
4	Dizziness	1.5114
58	Fatigue	1.4948
59	Pain	1.3453
133	Blood test	1.3453
18	Chills	1.2290
26	Asthenia	1.0962
14	Nausea	1.0463

**Figura 107.** Percentagem de pacientes que foram às urgências e receberam visita do médico por sintoma.

idsintoma	designacao	percentagem_pacientes
4	Dizziness	2.6549
12	Dyspnoea	2.5053
21	Headache	2.1875
14	Nausea	2.1376
37	Pyrexia	1.5580
18	Chills	1.2838
30	Vomiting	1.2589
59	Pain	1.2402
58	Fatigue	1.1903
3	Chest discomfort	1.1031

**Figura 108.** Percentagem de pacientes que foram às urgências, mas não receberam visita do médico por sintoma.

idsintoma	designacao	percentagem_pacientes
21	Headache	3.5617
5	Injection site erythema	3.0730
59	Pain	3.0067
58	Fatigue	2.7334
37	Pyrexia	2.6340
19	Injection site pain	2.3938
18	Chills	2.2364
32	Injection site swelling	2.0956
14	Nausea	2.0790
27	Pain in extremity	2.0459

**Figura 109.** Percentagem de pacientes que foram não foram às urgências, mas receberam visita do médico por sintoma.

idsintoma	designacao	percentagem_pacientes
21	Headache	5.6437
37	Pyrexia	4.6003
18	Chills	4.5465
58	Fatigue	4.1718
59	Pain	3.7552
14	Nausea	3.0579
27	Pain in extremity	2.4060
52	Myalgia	2.3640
19	Injection site pain	2.2178
4	Dizziness	2.1859

**Figura 110.** Percentagem de pacientes que não foram às urgências, nem receberam visita do médico por sintoma.

Por último, foram também avaliados os sintomas que as pessoas que ficaram mais tempo hospitalizadas (entre 37 e 39 dias) tinham. Pela análise da Figura 111, verifica-se que se recorreu ao comando SELECT das tabelas **sintomas**, **paciente**, **paciente\_e\_hospitalizado**, **hospitalizacao**, **vacina**, **vacinar** e **vacina\_causa\_sintomas** e se utilizaram as seguintes condições: os idsintoma da tabela **sintoma** e da tabela **vacina\_causa\_sintomas** são iguais; os idvacina das tabelas **vacina** e **vacina\_causa\_sintoma** e das tabelas **vacina** e **vacinar** são iguais; os id\_vaers das tabelas **paciente** e **vacinar** e das tabelas **paciente\_e\_hospitalizado** e **paciente** são iguais; os idhospitalizacao das tabelas **hospitalizacao** e **paciente\_e\_hospitalizado** são iguais; os dias de hospitalização eram 37 ou 38 ou 39.

Na Figura 112 estão representados os resultados da *query* implementada na Figura 111, em que são visíveis sintomas mais graves, relativamente aos sintomas mais frequentes, como astenia (fraqueza), paragem cardíaca, incapacidade de andar e perda de sensibilidade.

```
-- 5. Quais os sintomas que as pessoas que ficaram mais tempo hospitalizadas tinham?
SELECT s.idsintoma, s.designacao
FROM sintomas s, paciente p, hospitalizacao h, paciente_e_hospitalizado hp,
vacina v, vacinar vr, vacina_causa_sintomas vcs
WHERE s.idsintoma = vcs.idsintoma
AND v.idvacina = vcs.idvacina
AND v.idvacina = vr.idvacina
AND vr.id_vaers = p.id_vaers
AND p.id_vaers = hp.id_vaers
AND hp.idhospitalizacao = h.idhospitalizacao
AND (hp.hospdays = 39 OR hp.hospdays = 38 OR hp.hospdays = 37);
```

**Figura 111.** *Query* efetuada para determinar os sintomas que as pessoas que tiveram mais tempo hospitalizadas tinham.

idsintoma	designacao
12	Dyspnoea
26	Asthenia
37	Pyrexia
111	Laboratory test normal
369	Cardiac arrest
409	Vaccination complication
596	Mechanical ventilation
857	Gait inability
924	Sudden death
44	Paraesthesia
129	Mobility decreased
347	Muscular weakness
1342	Sensory loss

**Figura 112.** Sintomas que as pessoas que tiveram mais tempo hospitalizadas tinham.

Assim, com a análise das relações entre sintomas e hospitalização, é possível concluir que a maior parte dos pacientes tiveram os mesmos sintomas, mas apenas uma minoria é que recebeu cuidados médicos. Além disso, as pessoas que tiveram mais tempo internadas manifestaram sintomas mais graves relativamente às restantes, como paragem cardíaca, incapacidade de andar e perda de sensibilidade.

#### 6.1.5.2. Sintomas e pacientes

Agora serão analisadas possíveis relações entre os sintomas e os pacientes. Na Figura 113 está representada a *query* efetuada para determinar o tempo médio que decorreu entre o início dos sintomas e a data de vacinação, ou seja, o tempo médio até se iniciarem os sintomas após a vacinação. Como se pode observar, é utilizada a função Diferença (Figura 114), que retorna o número de dias entre 2 datas, funcionando como uma função idade, mas com a diferença que o cálculo é efetuado em dias e não em anos. A função recebe como argumento duas datas, data\_o e data\_m e retorna um inteiro, que resulta da aplicação da função TIMESTAMPDIFF do MySQL.

Pela análise da Figura 113, verifica-se que se recorreu ao comando SELECT da média do atributo inicio\_sintomas que é resultado da *subquery* a1. Nesta *subquery* foi selecionado o número distinto de id\_vaers da tabela **paciente** e o respetivo intervalo de tempo entre o vax\_date da tabela **vacinar** e o onset\_date da tabela **sintomas\_afeta\_paciente**, em que os id\_vaers da tabela **paciente** e das tabelas **vacinar** e **vacina\_causa\_sintomas** são iguais. Além dessas condições, os resultados da *subquery* foram ordenados de forma decrescente consoante o valor do respetivo inicio\_sintomas.

Na Figura 115 está representado o resultado do comando apresentado na Figura 113, em que se pode observar que o tempo médio até se iniciarem os sintomas nos pacientes é de 4.8 dias.

```
-- Relação sintomas - paciente
-- 1. Tempo médio que decorreu entre início sintomas e data de vacinação
SELECT AVG(inicio_sintomas) AS tempo_medio_inicio_sintomas
FROM (SELECT DISTINCT p.id_vaers, Diferenca(vr.vax_date, sa.onset_date) AS inicio_sintomas
FROM paciente p, sintomas_afeta_paciente sa, vacinar vr
WHERE p.id_vaers = sa.id_vaers
AND vr.id_vaers = p.id_vaers
ORDER BY inicio_sintomas DESC) a1;
```

**Figura 113.** Query efetuada para determinar o tempo médio até se iniciarem os sintomas.

```
-- Função Diferença
DELIMITER $$
CREATE FUNCTION Diferenca (data_o date, data_m date)
RETURNS INT
DETERMINISTIC
BEGIN
    RETURN TIMESTAMPDIFF(DAY, data_o, data_m);
END $$
DELIMITER ;
```

**Figura 114.** Função Diferença.

tempo_medio_inicio_sintomas
4.7674

**Figura 115.** Tempo médio até se iniciarem os sintomas.

Na Figura 116 está representada a *query* efetuada para o número médio de dias que os sintomas persistiram. Pela análise da figura, verifica-se que se recorreu ao comando SELECT da média do atributo duracao\_sintomas que é resultado da *subquery* a1. Nesta *subquery* foi selecionado o número distinto de id\_vaers da tabela **paciente** e o número de dias que os sintomas persistiram da tabela **sintomas\_afeta\_paciente**, em que os id\_vaers da tabela **paciente** e das tabelas **sintomas\_afeta\_paciente** são iguais e o número de dias de sintomas não é nulo. Além dessas condições, os resultados da *subquery* foram ordenados de forma decrescente consoante o valor da respetiva duracao\_sintomas.

Na Figura 117 está representado o resultado do comando apresentado na Figura 116, em que se pode observar que o tempo médio de duração dos sintomas foi de 21 dias.

```
-- 2. Número médio de dias que os sintomas persistiram
SELECT AVG(duracao_sintomas) AS tempo_medio_duracao_sintomas
FROM (SELECT DISTINCT p.id_vaers, sa.numdays AS duracao_sintomas
FROM paciente p, sintomas_afeta_paciente sa
WHERE p.id_vaers = sa.id_vaers
AND sa.numdays IS NOT NULL
ORDER BY duracao_sintomas DESC) a1;
```

**Figura 116.** Query efetuada para determinar o tempo médio até se iniciarem os sintomas.

tempo_medio_duracao_sintomas
21.0771

**Figura 117.** Tempo médio até se iniciarem os sintomas.

Por último, determinou-se a percentagem de pacientes que recuperaram completamente dos sintomas, através do quociente entre o número de pacientes diferentes que ocorrem na tabela **sintomas\_afeta\_paciente** em que o recovd não é nulo e o número de pacientes total da amostra (Figura 118). Pela análise da Figura 119, verifica-se que apenas 36% dos pacientes recuperaram completamente dos sintomas.

```
-- 3. Percentagem de pacientes que recuperaram completamente dos sintomas
SELECT (COUNT(DISTINCT(sa.id_vaers)))/(SELECT COUNT(*) FROM paciente p))*100 AS percentagem_recuperados
FROM sintomas_afeta_paciente sa
WHERE sa.recovd = 'Y';
```

**Figura 118.** Query efetuada para determinar a percentagem de pacientes que recuperaram completamente dos sintomas.

percentagem_recuperados
36.0217

**Figura 119.** Percentagem de pacientes que recuperaram completamente dos sintomas.

Assim, com a análise das relações entre sintomas e pacientes, é possível concluir que em média, os sintomas iniciam-se 5 dias após a vacinação e estes perduram (em média) 21 dias. Contudo, muitos dos pacientes (cerca de 64%) consideram que até ao momento do preenchimento do formulário, não estavam completamente recuperados de todos dos sintomas.

#### 6.1.5.3. Sintomas e patologias

Agora serão analisados os sintomas mais frequentes que pessoas com determinada patologia tiveram. O procedimento SINTOMAS\_PATOLOGIAS, representado na Figura 120, recebe como parâmetro o nome da patologia. Dentro do mesmo, está representada a *query* efetuada para determinar os 10 sintomas mais comuns em pacientes com uma determinada patologia e a respetiva percentagem de pacientes associada a essa patologia. Pela análise da figura, verifica-se que se recorreu ao comando SELECT das tabelas **sintomas**, **paciente**, **doença\_afeta\_paciente**, **historico\_clinico**, **vacina**, **vacinar** e **vacina\_causa\_sintomas** para se selecionar a razão entre o número de pacientes que tiveram um determinado sintoma e uma determinada patologia pelo número de pacientes que têm patologias, sendo esta última uma *subquery*. Para isso, foi utilizado o comando COUNT, que devolve um determinado número de linhas em que os iddoença da tabela **historico\_clinico** e da tabela **doença\_afeta\_paciente** são iguais, os idvacina das tabelas **vacina** e **vacina\_causa\_sintoma** e das tabelas **vacina** e **vacinar** são iguais, os id\_vaers das tabelas **paciente** e **vacinar** e das tabelas **doença\_afeta\_paciente** e **paciente** são iguais e os idsintomas das tabelas **sintomas** e **vacina\_causa\_sintomas** são iguais. Além disso, verifica-se também que o parâmetro doença recebido pelo *procedure* é igual ao atributo designacao da tabela **historico\_clinico**.



Nas Figuras 123 a 137 estão representados os resultados do *procedure*, após terem sido utilizados os comandos CALL, apresentados nas Figuras 121 e 122. Pela análise das figuras, verifica-se que muitos dos sintomas eram comuns aos pacientes com as diferentes patologias como dor de cabeça, febre, dores no local de injeção, fadiga, náuseas, arrepios e dores nas extremidades. Muitos destes sintomas são os sintomas mais frequentes na vacinação, como visto anteriormente na Figura 80, sendo menos graves comparativamente com os sintomas manifestados pelas pessoas que tiveram cuidados médicos.

No entanto, doentes com hipertensão ou hiperlipidemia apresentaram sintomas frequentes mais graves, como aperto na garganta. Deste modo, é possível concluir que pacientes com hipertensão ou hiperlipidemia estão sujeitos a sintomas mais graves, inclusive morte. Por último, é de realçar que as patologias que apresentam maior e menor percentagem de pacientes para cada sintoma são a elevada pressão sanguínea e deficiência, respetivamente.

```
-- Relação sintomas - patologias
DELIMITER $$
CREATE PROCEDURE SINTOMAS_PATOLOGIAS(doenca VARCHAR (45))
BEGIN
    SELECT s.idsintoma, s.designacao, (COUNT(vr.id_vaers)/(SELECT COUNT(vr.id_vaers)
    FROM sintomas s, paciente p, doenca_afeta_paciente d, historico_clinico h, vacina v, vacinar vr, vacina_causa_sintomas vcs
    WHERE s.idsintoma = vcs.idsintoma AND v.idvacina = vcs.idvacina
    AND v.idvacina = vr.idvacina AND vr.id_vaers = p.id_vaers
    AND p.id_vaers = d.id_vaers AND d.iddoenca = h.iddoenca
    AND h.designacao = doenca))*100 AS percentagem_pacientes
    FROM sintomas s, paciente p, doenca_afeta_paciente d, historico_clinico h, vacina v, vacinar vr, vacina_causa_sintomas vcs
    WHERE s.idsintoma = vcs.idsintoma AND v.idvacina = vcs.idvacina
    AND v.idvacina = vr.idvacina AND vr.id_vaers = p.id_vaers
    AND p.id_vaers = d.id_vaers AND d.iddoenca = h.iddoenca AND h.designacao = doenca
    GROUP BY s.designacao ORDER BY percentagem_pacientes DESC LIMIT 10;
END $$
DELIMITER ;
```

**Figura 120.** *Procedure* efetuado para determinar a percentagem de pacientes que apresentam determinada patologia por sintoma.

```

-- 1. Quais os sintomas mais frequentes que pessoas que já tiveram COVID-19 tiveram?
CALL SINTOMAS_PATOLOGIAS ('COVID-19');

-- 2. Quais os sintomas mais frequentes que pessoas com obesidade tiveram?
CALL SINTOMAS_PATOLOGIAS ('Obesity');

-- 3. Quais os sintomas mais frequentes que pessoas com colesterol elevado tiveram?
CALL SINTOMAS_PATOLOGIAS ('High_Cholesterol');

-- 4. Quais os sintomas mais frequentes que pessoas com pressão alta tiveram?
CALL SINTOMAS_PATOLOGIAS ('High_Blood_Pressure');

-- 5. Quais os sintomas mais frequentes que pessoas com MIGRAINES tiveram?
CALL SINTOMAS_PATOLOGIAS ('MIGRAINES');

-- 6. Quais os sintomas mais frequentes que pessoas com GERD tiveram?
CALL SINTOMAS_PATOLOGIAS ('GERD');

-- 7. Quais os sintomas mais frequentes que pessoas com depressão tiveram?
CALL SINTOMAS_PATOLOGIAS ('Depression');

```

**Figura 121.** Call's do procedure SINTOMAS\_PATOLOGIAS para as 7 primeiras patologias da tabela historico\_clinico.

```

-- 8. Quais os sintomas mais frequentes que pessoas com ansiedade tiveram?
CALL SINTOMAS_PATOLOGIAS ('Anxiety');

-- 9. Quais os sintomas mais frequentes que pessoas com osteoartrite tiveram?
CALL SINTOMAS_PATOLOGIAS ('Osteoarthritis');

-- 10. Quais os sintomas mais frequentes que pessoas com hipotireoidismo tiveram?
CALL SINTOMAS_PATOLOGIAS ('Hypothyroidism');

-- 11. Quais os sintomas mais frequentes que pessoas com hipertensão tiveram?
CALL SINTOMAS_PATOLOGIAS ('Hypertension');

-- 12. Quais os sintomas mais frequentes que pessoas com asma tiveram?
CALL SINTOMAS_PATOLOGIAS ('Asthma');

-- 13. Quais os sintomas mais frequentes que pessoas com hiperlipidemia tiveram?
CALL SINTOMAS_PATOLOGIAS ('Hyperlipidemia');

-- 14. Quais os sintomas mais frequentes que pessoas com doença de risco tiveram?
CALL SINTOMAS_PATOLOGIAS ('L_threat');

-- 15. Quais os sintomas mais frequentes que pessoas com deficiência tiveram?
CALL SINTOMAS_PATOLOGIAS ('Disable');

```

**Figura 122.** Call's do procedure SINTOMAS\_PATOLOGIAS para as restantes patologias da tabela historico\_clinico.

idsintoma	designacao	percentagem_pacientes
21	Headache	4.5656
59	Pain	3.8681
37	Pyrexia	3.6779
18	Chills	3.3608
58	Fatigue	3.2974
27	Pain in extremity	2.2194
19	Injection site pain	2.1560
14	Nausea	2.0292
4	Dizziness	2.0292
12	Dyspnoea	1.7755

**Figura 123.** Percentagem de pacientes que já tiveram COVID-19 por sintoma.

idsintoma	designacao	percentagem_pacientes
21	Headache	3.8136
58	Fatigue	3.1780
37	Pyrexia	3.1174
59	Pain	3.0569
18	Chills	2.7542
19	Injection site pain	1.9370
14	Nausea	1.7554
27	Pain in extremity	1.6344
5	Injection site erythema	1.5133
12	Dyspnoea	1.5133

**Figura 124.** Percentagem de pacientes que têm obesidade por sintoma.

idsintoma	designacao	percentagem_pacientes
21	Headache	4.6607
59	Pain	4.0439
58	Fatigue	3.4955
37	Pyrexia	3.3585
18	Chills	3.2214
27	Pain in extremity	2.2618
19	Injection site pain	2.1933
4	Dizziness	2.0562
14	Nausea	1.8506
12	Dyspnoea	1.8506

**Figura 125.** Percentagem de pacientes que têm colesterol elevado por sintoma.

idsintoma	designacao	percentagem_pacientes
21	Headache	4.8312
59	Pain	3.5924
37	Pyrexia	3.5615
58	Fatigue	3.5615
18	Chills	3.4995
27	Pain in extremity	2.0749
19	Injection site pain	2.0440
14	Nausea	2.0440
4	Dizziness	1.9511
5	Injection site erythema	1.8582

**Figura 126.** Percentagem de pacientes que têm elevado pressão sanguínea por sintoma.

idsintoma	designacao	percentagem_pacientes
21	Headache	4.7097
59	Pain	3.7618
18	Chills	3.3175
37	Pyrexia	3.2583
58	Fatigue	3.2583
14	Nausea	2.5474
19	Injection site pain	2.2808
5	Injection site erythema	2.1327
27	Pain in extremity	2.1031
4	Dizziness	1.9846

**Figura 127.** Percentagem de pacientes que têm MIGRAINES por sintoma.

idsintoma	designacao	percentagem_pacientes
21	Headache	3.3326
37	Pyrexia	2.5752
59	Pain	2.4886
58	Fatigue	2.4670
18	Chills	2.3372
14	Nausea	1.9909
12	Dyspnoea	1.8827
4	Dizziness	1.8178
19	Injection site pain	1.5365
27	Pain in extremity	1.3201

**Figura 128.** Percentagem de pacientes que têm GERD por sintoma.

idsintoma	designacao	percentagem_pacientes
21	Headache	3.9880
58	Fatigue	3.2665
37	Pyrexia	3.1663
59	Pain	3.1663
18	Chills	2.9058
14	Nausea	2.4850
19	Injection site pain	1.9840
4	Dizziness	1.9238
27	Pain in extremity	1.7435
12	Dyspnoea	1.7234

**Figura 129.** Percentagem de pacientes que têm depressão por sintoma.

idsintoma	designacao	percentagem_pacientes
21	Headache	3.9482
58	Fatigue	3.2285
37	Pyrexia	3.0023
59	Pain	2.9611
14	Nausea	2.7349
18	Chills	2.7144
4	Dizziness	2.1386
19	Injection site pain	1.8301
27	Pain in extremity	1.7890
12	Dyspnoea	1.6451

**Figura 130.** Percentagem de pacientes que têm ansiedade por sintoma.

idsintoma	designacao	percentagem_pacientes
21	Headache	3.6098
58	Fatigue	2.9570
59	Pain	2.9570
37	Pyrexia	2.8034
18	Chills	2.6114
14	Nausea	1.9585
12	Dyspnoea	1.9201
27	Pain in extremity	1.8817
19	Injection site pain	1.7665
4	Dizziness	1.6897

**Figura 131.** Percentagem de pacientes que têm osteoartrite por sintoma.

idsintoma	designacao	percentagem_pacientes
21	Headache	3.5756
58	Fatigue	2.8991
18	Chills	2.8218
59	Pain	2.8218
37	Pyrexia	2.6672
14	Nausea	1.7588
27	Pain in extremity	1.6622
19	Injection site pain	1.6428
4	Dizziness	1.6428
12	Dyspnoea	1.3143

**Figura 132.** Percentagem de pacientes que têm hipotireoidismo por sintoma.

idsintoma	designacao	percentagem_pacientes
21	Headache	2.8344
37	Pyrexia	2.6155
58	Fatigue	2.3044
18	Chills	2.1431
59	Pain	2.0509
14	Nausea	1.7629
12	Dyspnoea	1.5900
94	Death	1.5555
4	Dizziness	1.5094
27	Pain in extremity	1.1753

**Figura 133.** Percentagem de pacientes que têm hipertensão por sintoma.

idsintoma	designacao	percentagem_pacientes
21	Headache	4.3847
58	Fatigue	3.5127
18	Chills	3.3383
37	Pyrexia	3.2262
59	Pain	3.1390
14	Nausea	2.6781
12	Dyspnoea	2.3169
4	Dizziness	2.2546
19	Injection site pain	1.9557
27	Pain in extremity	1.7439

**Figura 134.** Percentagem de pacientes que têm asma por sintoma.

idsintoma	designacao	percentagem_pacientes
21	Headache	2.8344
37	Pyrexia	2.6155
58	Fatigue	2.3044
18	Chills	2.1431
59	Pain	2.0509
14	Nausea	1.7629
12	Dyspnoea	1.5900
94	Death	1.5555
4	Dizziness	1.5094
27	Pain in extremity	1.1753

**Figura 135.** Percentagem de pacientes que têm hiperlipidemia por sintoma.

idsintoma	designacao	percentagem_pacientes
12	Dyspnoea	2.5401
21	Headache	1.6195
37	Pyrexia	1.4320
14	Nausea	1.3297
4	Dizziness	1.2104
18	Chills	1.0569
45	Anaphylactic reaction	0.9206
633	Intensive care	0.9206
58	Fatigue	0.9035
273	Throat tightness	0.8865

**Figura 136.** Percentagem de pacientes que têm uma doença de risco por sintoma.

idsintoma	designacao	percentagem_pacientes
59	Pain	1.8746
21	Headache	1.8453
20	Arthralgia	1.6989
27	Pain in extremity	1.6110
58	Fatigue	1.5231
35	Hypoaesthesia	1.2595
37	Pyrexia	1.2302
170	Computerised tomogram	1.1716
480	Magnetic resonance imaging	1.1716
292	Facial paralysis	1.1131

**Figura 137.** Percentagem de pacientes que têm deficiência por sintoma.

#### 6.1.5.4. Sintomas e vacinas

Agora serão analisados os sintomas mais frequentes que pessoas que tomaram uma vacina de uma dada marca tiveram. O procedimento SINTOMAS\_VACINA, representado na Figura 138 recebe como parâmetro o nome da marca da vacina. Dentro do mesmo, está representada a *query* efetuada para determinar os 10 sintomas mais comuns em pacientes que tomaram uma vacina de uma dada marca e a respetiva percentagem de pacientes. Pela análise da figura, verifica-se que se recorreu ao comando SELECT das tabelas **sintomas**, **paciente**, **doenca\_afeta\_paciente**, **historico\_clinico**, **vacina**, **vacinar**, **fornecedor** e **vacina\_causa\_sintomas** para se selecionar a razão entre o número de pacientes que tiveram um determinado sintoma e tomaram a vacina de uma certa marca pelo número de pacientes que tomaram vacina, sendo esta última uma *subquery*. Para isso, foi utilizado o comando COUNT,

que devolve um determinado número de linhas em que os iddoença da tabela **historico\_clinico** e da tabela **doença\_afeta\_paciente** são iguais, os idvacina das tabelas **vacina** e **vacina\_causa\_sintoma** e das tabelas **vacina** e **vacinar** são iguais, os id\_vuers das tabelas **paciente** e **vacinar** e das tabelas **doença\_afeta\_paciente** e **paciente** são iguais, os idfornecedor das tabelas **vacina** e **fornecedor** são iguais e os idsintomas das tabelas **sintomas** e **vacina\_causa\_sintomas** são iguais. Além disso, verifica-se também que o parâmetro vacina recebido pelo *procedure* é igual ao atributo vax\_manu da tabela **fornecedor**.

Nas Figuras 140 a 142 estão representados os resultados do *procedure*, após terem sido utilizados os comandos CALL, apresentados na Figura 139. Pela análise das figuras, verifica-se que os sintomas mais frequentes para os pacientes que tomaram vacinas da Moderna ou da Pfizer são exatamente os mesmos, apresentando percentagens bastantes parecidas. É também de realçar que são muitos destes sintomas são os sintomas mais frequentes na vacinação, como visto anteriormente na Figura 80.

Por outro lado, apesar da vacina da marca Janssen ter sido a menos inoculada nesta amostra de pacientes, esta foi não só a que teve maior percentagem de pacientes com sintomas, como também, foi a que apresentou sintomas com maior gravidade como desconforto no peito e ainda erupções cutâneas como eritemas e prurito.

```
-- Relação sintomas - vacinas
DELIMITER $$
CREATE PROCEDURE SINTOMAS_VACINA(vacina VARCHAR (45))
BEGIN
    SELECT s.idsintoma, s.designacao, (COUNT(vr.id_vuers)/(SELECT COUNT(vr.id_vuers)
    FROM sintomas s, paciente p, doença_afeta_paciente d, historico_clinico h, vacina v,
    vacinar vr, vacina_causa_sintomas vcs, fornecedor f
    WHERE s.idsintoma = vcs.idsintoma AND v.idvacina = vcs.idvacina
    AND v.idvacina = vr.idvacina AND vr.id_vuers = p.id_vuers
    AND p.id_vuers = d.id_vuers AND d.iddoença = h.iddoença
    AND f.idfornecedor = v.idfornecedor AND f.vax_manu = vacina))*100 AS percentagem_pacientes
    FROM sintomas s, paciente p, doença_afeta_paciente d, historico_clinico h,
    vacina v, vacinar vr, vacina_causa_sintomas vcs, fornecedor f
    WHERE s.idsintoma = vcs.idsintoma AND v.idvacina = vcs.idvacina
    AND v.idvacina = vr.idvacina AND vr.id_vuers = p.id_vuers
    AND p.id_vuers = d.id_vuers AND d.iddoença = h.iddoença
    AND f.idfornecedor = v.idfornecedor AND f.vax_manu = vacina
    GROUP BY s.designacao ORDER BY percentagem_pacientes DESC LIMIT 10;
END $$
DELIMITER ;
```

**Figura 138.** *Procedure* efetuado para determinar a percentagem de pacientes que tomaram vacina de uma dada marca por sintoma.



```
-- 1. Quais os sintomas mais frequentes nas pessoas que tomaram vacina da MODERNA?
CALL SINTOMAS_VACINA ('MODERA');

-- 2. Quais os sintomas mais frequentes nas pessoas que tomaram vacina da PFIZER\\BIOTECH?
CALL SINTOMAS_VACINA ('PFIZER\\BIOTECH');

-- 3. Quais os sintomas mais frequentes nas pessoas que tomaram vacina da JASSE?
CALL SINTOMAS_VACINA ('JASSE');
```

**Figura 139.** Call's do procedure SINTOMAS\_VACINA.

idsintoma	designacao	percentagem_pacientes
21	Headache	3.5010
37	Pyrexia	2.9619
58	Fatigue	2.9121
59	Pain	2.8975
18	Chills	2.6192
14	Nausea	2.1826
12	Dyspnoea	1.7930
19	Injection site pain	1.6963
4	Dizziness	1.6026
27	Pain in extremity	1.5381

**Figura 140.** Percentagem de pacientes que tomaram vacina da Moderna por sintoma.

idsintoma	designacao	percentagem_pacientes
21	Headache	3.4098
37	Pyrexia	2.5249
58	Fatigue	2.5048
18	Chills	2.4039
59	Pain	2.3462
4	Dizziness	1.7726
14	Nausea	1.7553
12	Dyspnoea	1.6977
27	Pain in extremity	1.5132
19	Injection site pain	1.3316

**Figura 141.** Percentagem de pacientes que tomaram vacina da Pfizer por sintoma.

idsintoma	designacao	percentagem_pacientes
4	Dizziness	4.0174
5	Injection site erythema	3.9088
14	Nausea	3.8002
21	Headache	2.7144
3	Chest discomfort	2.1716
40	Pruritus	2.0630
32	Injection site swelling	1.9544
26	Asthenia	1.8458
58	Fatigue	1.8458
59	Pain	1.7372

**Figura 142.** Percentagem de pacientes que tomaram vacina da Janssen por sintoma.

Por último, determinaram-se os lotes de vacinas que provocaram mais sintomas, respetivos fornecedores desses lotes a percentagem de sintomas provocados por esses lotes. Pela análise da Figura 143, verifica-se que se recorreu ao comando SELECT do `vax_manu` e do `vax_lot` da tabela **fornecedor** e da percentagem de sintomas provocados por lote de vacina. Para determinar esta percentagem, procedeu-se à razão do número (COUNT) de sintomas provocados por um dado lote (*query* principal), pelo total de sintomas provocados por todos os lotes (*subquery*).

Na *subquery* foi selecionado o `idsintoma` da tabela **sintomas**, em que os `idvacina` da tabela **vacina** e da tabela **vacina\_causa\_sintomas** e em que os `idfornecedor` da tabela **fornecedor** e da tabela **vacina** são iguais. Quanto à *query* foram impostas as mesmas condições que na *subquery*, e ainda, com o comando GROUP BY `vax_lot` foram agrupadas as vacinas por lote e através do ORDER BY, os lotes foram ordenados de forma decrescente consoante o valor da respetiva `percentagem_sintomas`, com um total de 10 lotes diferentes.

Na Figura 144 está representado o resultado do comando apresentado na Figura 143, em que se pode observar que os dois lotes que provocam mais sintomas são da Moderna e são respetivamente os dois lotes de vacinas mais inoculados, como se observou na Figura 92. O 3º lote de vacina que provoca mais sintomas é da Pfizer, sendo o 3º lote de vacina mais inoculado na amostra.

```
-- 4. Lote de vacina que provocou mais sintomas e de que marca é
SELECT v.vax_lot, f.vax_manu, (COUNT(vc.idsintoma)/(SELECT COUNT(vc.idsintoma)
FROM vacina v, fornecedor f, vacina_causa_sintomas vc
WHERE f.idfornecedor = v.idfornecedor
AND v.idvacina = vc.idvacina
AND vax_lot IS NOT NULL))*100 AS percentagem_sintomas
FROM vacina v, fornecedor f, vacina_causa_sintomas vc
WHERE f.idfornecedor = v.idfornecedor
AND v.idvacina = vc.idvacina
AND vax_lot IS NOT NULL
GROUP BY v.vax_lot
ORDER BY percentagem_sintomas DESC
LIMIT 10;
```

**Figura 143.** Query efetuada para determinar a percentagem de sintomas provocados por um dado lote de vacinas.

vax_lot	vax_manu	percentagem_sintomas
039k20a	MODERA	6.7422
011J20A	MODERA	5.0467
EK9231	PFIZER\BIOTECH	4.6228
026L20A	MODERA	4.4881
Ek530	PFIZER\BIOTECH	4.4667
EH9899	PFIZER\BIOTECH	3.6771
Ej1685	PFIZER\BIOTECH	3.1492
EL1284	PFIZER\BIOTECH	3.0987
03K20A	MODERA	2.9319
025L20A	MODERA	2.8279

**Figura 144.** Percentagem de sintomas provocados por um dado lote de vacinas.

Assim, com a análise das relações entre os sintomas e as vacinas, é possível concluir que apesar dos lotes da Moderna serem os que provocam maior percentagem de sintomas, são as vacinas da Jassen que são responsáveis pelo aparecimento de sintomas mais graves com maior frequência.

#### 6.1.6. Morte

Por último, serão analisadas as relações que existem entre a morte dos pacientes e outras entidades, como a hospitalização, os pacientes, as patologias, os sintomas, a vacinação e as vacinas.

##### 6.1.6.1. Morte e hospitalização

Na Figura 145 está representada a *query* efetuada para determinar a percentagem de pessoas que morreram e que foram às urgências ou receberam visita médica. Pela análise da figura, verifica-se que se recorreu ao comando SELECT das tabelas **paciente**, **hospitalização** e

**paciente\_e\_hospitalizado** para se selecionar a razão entre o número de pacientes que morreram e estiveram hospitalizadas ou receberam visita do médico, pelo total de pacientes que morreram, sendo esta última uma *subquery*. Para isso, foi utilizado também o comando COUNT nas condições anteriores, que devolve um determinado número de linhas em que os id\_vaers da tabela **paciente** e da tabela **paciente\_e\_hospitalizado** são iguais, assim como os idhospitalizacao das tabelas **hospitalizacao** e **paciente\_e\_hospitalizado** e em que data\_died é NOT NULL.

Na Figura 146 está representado o resultado do comando apresentado na Figura 145, em que se pode observar que cerca de 21% das pessoas que morreram, foram às urgências ou receberam visita médica.

```
-- Morte e hospitalização
-- 1. Percentagem de pessoas que morreram que foram às urgências ou receberam visita do médico
SELECT (COUNT(p.data_died)/(SELECT COUNT(*) FROM paciente WHERE data_died IS NOT NULL))*100 AS percentagem_mortes
FROM paciente p, hospitalizacao h, paciente_e_hospitalizado hp
WHERE p.id_vaers = hp.id_vaers
AND hp.idhospitalizacao = h.idhospitalizacao
AND (h.er_ed_visit = 'Yes' OR h.ofc_visit = 'Yes')
AND p.data_died IS NOT NULL;
```

**Figura 145.** Query efetuada para determinar a percentagem de pessoas que morreram e foram às urgências ou receberam visita do médico.

percentagem_mortes
20.6723

**Figura 146.** Percentagem de pessoas que morreram e foram às urgências ou receberam visita do médico.

Na Figura 147 está representada a *query* efetuada para determinar o número médio de dias em que as pessoas que morreram ficaram hospitalizadas. Pela análise da figura, verifica-se que se recorreu ao comando SELECT média (AVG) do atributo dias\_hospitalizacao que é resultado da *subquery* a1. Nesta *subquery* foi selecionado o número de dias de hospitalização da tabela **paciente\_e\_hospitalizado**, em que os id\_vaers da tabela **paciente** e da tabela **paciente\_e\_hospitalizado** são iguais, assim como os idhospitalizacao das tabelas **hospitalizacao** e **paciente\_e\_hospitalizado** e em que o data\_died não é um valor nulo. Além dessas condições, também se garantiu que o hospdays era NOT NULL, de forma a obter todos os pacientes que estiveram hospitalizados.

Na Figura 148 está representado o resultado do comando apresentado na Figura 147, em que se pode observar que os pacientes que morreram ficaram em média 5 dias e meio hospitalizados.

```
-- 2. Número médio de dias que as pessoas que morreram ficaram hospitalizadas
SELECT AVG(dias_hospitalizacao) AS tempo_medio_hospitalizacao
FROM (SELECT hp.hospdays AS dias_hospitalizacao
FROM paciente p, hospitalizacao h, paciente_e_hospitalizado hp
WHERE p.id_vaers = hp.id_vaers
AND hp.idhospitalizacao = h.idhospitalizacao
AND p.data_died IS NOT NULL
AND hp.hospdays IS NOT NULL) a1;
```

**Figura 147.** Query efetuada para determinar o número médio de dias em que as pessoas que morreram ficaram hospitalizadas.

tempo_medio_hospitalizacao
5.5023

**Figura 148.** Número médio de dias em que as pessoas que morreram ficaram hospitalizadas.

O mesmo comando foi utilizado para determinar o número médio de dias que pessoas que não morreram ficaram hospitalizadas, em que neste caso o atributo `data_died` da tabela **paciente** poderia ser nulo, como se pode observar na Figura 149. Na Figura 150 está representado o resultado do comando apresentado na Figura 149, em que se pode observar que em que se pode observar que os pacientes que não morreram ficaram em média 3 dias e meio hospitalizados.

```
-- 3. Número médio de dias que as pessoas que não morreram ficaram hospitalizadas
SELECT AVG(dias_hospitalizacao) AS tempo_medio_hospitalizacao
FROM (SELECT hp.hospdays AS dias_hospitalizacao
FROM paciente p, hospitalizacao h, paciente_e_hospitalizado hp
WHERE p.id_vaers = hp.id_vaers
AND hp.idhospitalizacao = h.idhospitalizacao
AND p.data_died IS NULL
AND hp.hospdays IS NOT NULL) a1;
```

**Figura 149.** Query efetuada para determinar o número médio de dias em que as pessoas que não morreram ficaram hospitalizadas

tempo_medio_hospitalizacao
3.5969

**Figura 150.** Número médio de dias em que as pessoas que não morreram ficaram hospitalizadas.

Após se terem obtido estes dados, é possível concluir que as pessoas que morreram ficaram mais tempo hospitalizadas, o que faz sentido, pois provavelmente teriam sintomas mais graves. Contudo, é também de realçar que apenas cerca de 21% das pessoas que morreram foram às urgências ou receberam visita do médico, o que pode indicar uma desvalorização dos sintomas por parte dos pacientes ou ainda impossibilidade de serem atendidos (esgotamento dos serviços de saúde).

#### 6.1.6.2. Morte e pacientes

O procedimento MORTE\_PACIENTES\_SEXO, presente na Figura 151, recebe como parâmetro o sexo do paciente. Dentro do mesmo, está representada a *query* efetuada para determinar a percentagem de mortes associadas ao género do paciente. Pela análise da figura, verifica-se que se recorreu ao comando SELECT das tabelas **vacinar e paciente** para se selecionar a razão entre o número de pacientes de um determinado género e o total de pacientes, sendo esta última uma *subquery*. Para isso, foi utilizado o comando COUNT, que devolve um determinado número de linhas em que os id\_vaers da tabela **paciente** e da tabela **vacinar** são iguais, em que data\_died é NOT NULL e em que o parâmetro recebido pelo *procedure* é igual ao atributo sex da tabela **paciente**.

Nas Figuras 152 e 153 estão representados os resultados dos *call's* do *procedure*, que “chama” e executa o *procedure*, em que se pode observar que apenas 3% das mulheres da amostra faleceram, enquanto que aproximadamente 11% dos homens faleceram. Deste modo, verifica-se uma diferença substancial nas percentagens de mortes para cada um dos géneros, sendo visível uma tendência bastante superior dos homens morrerem por vacinação contra a COVID-19, em relação às mulheres.

```

-- Morte e pacientes
DELIMITER $$
CREATE PROCEDURE MORTE_PACIENTES_SEXO(sexo VARCHAR (45))
BEGIN
    SELECT (COUNT(vr.id_vaers)/(SELECT COUNT(vr.id_vaers)
    FROM vacinar vr, paciente p
    WHERE vr.id_vaers = p.id_vaers AND p.sex = sexo))*100 AS percentagem_mortes
    FROM vacinar vr, paciente p WHERE vr.id_vaers = p.id_vaers
    AND p.sex = sexo AND p.data_died IS NOT NULL;

END $$
DELIMITER ;

-- 1. Percentagem de mulheres que morrerem
CALL MORTE_PACIENTES_SEXO ('F');

-- 2. Percentagem de homens que morrerem
CALL MORTE_PACIENTES_SEXO ('M');

```

**Figura 151.** Procedure efetuado para determinar a percentagem de mortes por género.

percentagem_mortes
3.0135

**Figura 152.** Percentagem de mulheres que faleceram.

percentagem_mortes
11.0794

**Figura 153.** Percentagem de homens que faleceram.

O procedimento `MORTE_PACIENTES_IDADE` presente na Figura 154, recebe como parâmetro um intervalo de idades, ambos INT, correspondente a uma determinada faixa etária onde os pacientes se inserem e retorna a variável `percentagem` FLOAT (inicialmente declarada como NULL). Quando se pretende determinar a percentagem de bebés que morreu (idade inferior a 1 ano), o *procedure* executa a 1ª condição do IF, em que o maior é NULL, pois pretende-se determinar a idade até ao 1 ano. Para crianças (1-12 anos), adolescentes (12-16 anos), jovens (16-25) e adultos (25-65), o *procedure* executa a 2ª condição do IF, em que menor e maior são ambos NOT NULL. Por último, para idosos, é executada a 3ª condição do IF, com menor NULL (Figura 155). Cada vez que o *procedure* é executado, este atualiza (através do comando SET) a variável `percentagem`, igualando-a SELECT da razão do número de pacientes com determinada

faixa etária que morreram pelo total de pacientes dessa mesma faixa etária. Nas Figuras 156 a 161 estão representados os resultados dos *call's* do *procedure*, que “chama” e executa o *procedure*, tendo em conta os argumentos de entrada dados. Verifica-se que a faixa etária dos idosos é a que regista maior taxa de mortalidade com cerca de 19%, seguida da faixa etária dos bebés, com aproximadamente 4 % e os adultos com cerca de 1%. As restantes apresentam uma taxa de mortalidade nula ou muito reduzida.

```
DELIMITER $$
CREATE PROCEDURE MORTE_PACIENTES_IDADE(menor INT, maior INT)
BEGIN
    DECLARE percentagem_mortes FLOAT DEFAULT NULL;

    IF (menor IS NOT NULL AND maior IS NULL)
    THEN SET percentagem_mortes = (SELECT (COUNT(vr.id_vaers)/(SELECT COUNT(vr.id_vaers) FROM vacinar vr, paciente p
    WHERE vr.id_vaers = p.id_vaers AND age_yrs <= menor))*100 AS percentagem_mortes
    FROM vacinar vr, paciente p WHERE vr.id_vaers = p.id_vaers AND age_yrs <= menor AND p.data_died IS NOT NULL);

    ELSEIF (menor IS NOT NULL AND maior IS NOT NULL)
    THEN SET percentagem_mortes = (SELECT (COUNT(vr.id_vaers)/(SELECT COUNT(vr.id_vaers) FROM vacinar vr, paciente p
    WHERE vr.id_vaers = p.id_vaers AND age_yrs > menor AND age_yrs <= maior))*100 AS percentagem_mortes FROM vacinar vr, paciente p
    WHERE vr.id_vaers = p.id_vaers AND age_yrs > menor AND age_yrs <= maior AND p.data_died IS NOT NULL);

    ELSEIF (menor IS NULL AND maior IS NOT NULL)
    THEN SET percentagem_mortes = (SELECT (COUNT(vr.id_vaers)/(SELECT COUNT(vr.id_vaers) FROM vacinar vr, paciente p
    WHERE vr.id_vaers = p.id_vaers AND age_yrs > maior))*100 AS percentagem_mortes
    FROM vacinar vr, paciente p WHERE vr.id_vaers = p.id_vaers AND age_yrs <> maior AND p.data_died IS NOT NULL);

    END IF;
    SELECT percentagem_mortes;
END $$
DELIMITER ;
```

**Figura 154.** *Procedure* efetuado para determinar a percentagem de mortes de uma determinada faixa etária dos pacientes.

```
-- 3. Percentagem de bebés que morreram
CALL MORTE_PACIENTES_IDADE (1,NULL);

-- 4. Percentagem de crianças que morreram
CALL MORTE_PACIENTES_IDADE (1,12);

-- 5. Percentagem de adolescentes que morreram
CALL MORTE_PACIENTES_IDADE (12,16);

-- 6. Percentagem de jovens que morreram
CALL MORTE_PACIENTES_IDADE (16,25);

-- 7. Percentagem de adultos que morreram
CALL MORTE_PACIENTES_IDADE (25,65);

-- 8. Percentagem de idosos que morreram
CALL MORTE_PACIENTES_IDADE (NULL,65);
```

**Figura 155.** Comandos CALL para determinar a percentagem de mortes de uma determinada faixa etária após a toma da vacina contra a COVID-19.



percentagem_mortes
3.5714

**Figura 156.** Percentagem de bebés, dos 0 a 1 ano de idade, que morreram após a toma da vacina contra a COVID-19.

percentagem_mortes
0.0000

**Figura 157.** Percentagem de crianças, de 1 aos 12 anos de idade, que morreram após a toma da vacina contra a COVID-19.

percentagem_mortes
0.0000

**Figura 158.** Percentagem de adolescentes, dos 12 aos 16 anos, que morreram após a toma da vacina contra a COVID-19.

percentagem_mortes
0.4145

**Figura 159.** Percentagem de jovens, dos 16 aos 25 anos, que morreram após a toma da vacina contra a COVID-19.

percentagem_mortes
1.3147

**Figura 160.** Percentagem de adultos, dos 25 aos 65 anos, que morreram após a toma da vacina contra a COVID-19

percentagem_mortes
18.7006

**Figura 161.** Percentagem de idosos, de 65 ou mais anos, que morreram após a toma da vacina contra a COVID-19.

Deste modo, a partir da análise da relação da morte com os pacientes, é possível concluir que as vacinas da COVID-19 provocam mortes maioritariamente em idosos e bebés, pelo que se pode afirmar que estes são grupos de risco para a vacinação contra a COVID-19. Apesar das taxas de mortalidade serem relativamente elevadas, nomeadamente a dos idosos, este grupo continua a ser vacinado, pois, claramente, os benefícios associados à vacinação superam os riscos. Sabe-se também que a taxa de mortalidade tem uma incidência superior em homens do que em mulheres, o que indica que as mulheres ou possuem menos recetores para o vírus SARS-CoV-2 ou possuem sistemas imunológicos mais fortes para o combate desse vírus.

#### 6.1.6.3. Morte e patologias

Na Figura 162 está representada a *query* efetuada para determinar a percentagem de pessoas que morreram e que tinham as patologias mais frequentes. Pela análise da figura, verifica-se que se recorreu ao comando SELECT do iddoenca e designacao da tabela **histórico\_clinico** e da percentagem das pessoas que morreram e que tinham uma determinada doença. Para determinar esta percentagem, procedeu-se à razão do número (COUNT) de pacientes que morreram (data\_died não nulo) e que tinham uma determinada doença (*query* principal), pelo total de pacientes que morreram e que tinham pelo menos 1 patologia (*subquery*).

Na *subquery* foi selecionado o número de iddoenca da tabela **doença\_afeta\_paciente**, em que os iddoenca da tabela **histórico\_clinico** e da tabela **doença\_afeta\_paciente**, em que os id\_vaers da tabela **paciente** e da tabela **doença\_afeta\_paciente** são iguais e em que o data\_died não é um valor nulo. Quanto à *query* foram postas as seguintes condições: os iddoenca da tabela **histórico\_clinico** e da tabela **doença\_afeta\_paciente**; os id\_vaers da tabela **paciente** e da tabela **doença\_afeta\_paciente** são iguais; o data\_died não é um valor nulo; os resultados da *query* foram agrupados por designações iguais da tabela **historico\_clinico** com o comando GROUP BY designacao e ordenados de forma decrescente consoante o valor da respetiva percentagem\_patologias, com um total de 10 patologias diferentes.

Na Figura 163 está representado o resultado do comando apresentado na Figura 162, em que se pode observar que mais de 20% dos pacientes que morreram tinham hiperlipidemia e/ou hipertensão. Além disso, mais de 8% dos pacientes que morreram tinham hipotireoidismo e/ou GERD. Assim sendo, é possível concluir que estas 4 doenças são doenças de risco para pacientes que se vacinam contra a COVID-19. Estes resultados estão de acordo com os obtidos na secção

6.1.5.3, em que se verificou que doentes com hipertensão e hiperlipidemia apresentaram sintomas frequentes mais graves.

```
-- Morte e patologias
-- 1. Percentagem de pessoas que morreram com as patologias mais frequentes
SELECT h.iddoenca, h.designacao, (COUNT(d.iddoenca)/(SELECT COUNT(d.iddoenca)
FROM historico_clinico h, paciente p, doenca_afeta_paciente d
WHERE h.iddoenca = d.iddoenca
AND p.id_vaers = d.id_vaers
AND p.data_died IS NOT NULL))*100 AS percentagem_patologias
FROM historico_clinico h, paciente p, doenca_afeta_paciente d
WHERE h.iddoenca = d.iddoenca
AND p.id_vaers = d.id_vaers
AND p.data_died IS NOT NULL
GROUP BY h.designacao
ORDER BY percentagem_patologias DESC
LIMIT 10;
```

**Figura 162.** Query efetuada para determinar a percentagem de pessoas que morreram com as patologias mais frequentes

iddoenca	designacao	percentagem_patologias
12	Hyperlipidemia	21.5577
10	Hypertension	21.5577
9	Hypothyroidism	8.9013
5	GERD	8.6926
7	Anxiety	6.1892
6	Depression	5.9805
8	Osteoarthritis	5.4242
1	Obesity	5.2156
11	Asthma	3.9638
3	High_Blood_Pressure	3.4771

**Figura 163.** Percentagem de pessoas que morreram e que tinham as patologias mais frequentes.

#### 6.1.6.4. Morte e sintomas

Na Figura 163 está representada a *query* efetuada para determinar a percentagem de pessoas que morreram com os 11 sintomas mais frequentes. Pela análise da figura, verifica-se que se recorreu ao comando SELECT do idsintoma e designacao da tabela **sintomas** e da percentagem das pessoas que morreram e que tiveram um determinado sintoma. Para determinar esta percentagem, procedeu-se à razão do número (COUNT) de pacientes que morreram (data\_died

não nulo) e que tinham um determinado sintoma (*query* principal), pelo total de pacientes que morreram e que tinham pelo menos 1 sintoma (*subquery*).

Na *subquery* foi selecionado o número de mortes da tabela **paciente**, em que o idsintoma da tabela **sintomas** e da **vacina\_causa\_sintomas** são iguais, em que o idvacina da tabela **vacina** e da tabela **vacina\_causa\_sintomas** também são iguais, que também é igual ao idsintoma da tabela **vacinar**, em que o id\_vaers da tabela **paciente** e **vacinar** e em que o data\_died não é um valor nulo. Quanto à *query* foram postas as seguintes condições: o idsintoma da tabela **sintomas** e da tabela **vacina\_causa\_sintomas** são iguais, o idvacina da tabela **vacina** e da tabela **vacina\_causa\_sintomas** também são iguais, que também é igual ao idsintoma da tabela **vacinar**, em que o id\_vaers da tabela **paciente** e **vacinar** e em que o data\_died da tabela **paciente** não é um valor nulo. Os resultados da *query* foram agrupados por designações iguais da tabela **sintomas** com o comando GROUP BY designacao, ordenados de forma decrescente consoante o valor da respetiva percentagem\_mortes e limitados aos onze sintomas mais frequentes. É de realçar que se escolheu 11, uma vez que o 1º sintoma é morte, pelo que se desprezou a 1ª linha dos resultados.

Na Figura 164 está representado o resultado do comando apresentado na Figura 163, em que se pode observar que mais de 2% dos pacientes que morreram tiveram dispneia (dificuldade em respirar) e/ou não respondiam a estímulos. Além disso, quase 2% dos pacientes que morreram tiveram paragem cardíaca. Verifica-se também que estes sintomas são mais graves do que os sintomas mais comuns associados à vacinação. Assim sendo, é possível concluir que estes 3 sintomas são sintomas que muito provavelmente podem estar associados à morte dos pacientes pela vacinação contra a COVID-19.

```
-- Análise da morte com os sintomas
-- 1. Quais os sintomas mais frequentes comuns às pessoas que morreram?
SELECT s.idsintoma, s.designacao, (COUNT(p.data_died)/(SELECT COUNT(p.data_died)
FROM sintomas s, paciente p, vacina v, vacinar vr, vacina_causa_sintomas vcs
WHERE s.idsintoma = vcs.idsintoma
AND v.idvacina = vcs.idvacina
AND v.idvacina = vr.idvacina
AND vr.id_vaers = p.id_vaers
AND p.data_died IS NOT NULL))*100 AS percentagem_mortes
FROM sintomas s, paciente p, vacina v, vacinar vr, vacina_causa_sintomas vcs
WHERE s.idsintoma = vcs.idsintoma
AND v.idvacina = vcs.idvacina
AND v.idvacina = vr.idvacina
AND vr.id_vaers = p.id_vaers
AND p.data_died IS NOT NULL
GROUP BY s.designacao
ORDER BY percentagem_mortes DESC
LIMIT 11;
```

Figura 164. Percentagem de pessoas que morreram com os sintomas mais frequentes e respetivas patologias.

idsintoma	designacao	percentagem_mortes
94	Death	16.8406
12	Dyspnoea	2.3532
770	Unresponsive to stimuli	2.3123
369	Cardiac arrest	1.8621
879	Resuscitation	1.6165
258	SARS-CoV-2 test positive	1.4733
187	COVID-19	1.3505
30	Vomiting	1.1254
58	Fatigue	1.0845
26	Asthenia	1.0640
14	Nausea	1.0436

Figura 165. Percentagem de pessoas que morreram e que tinham os sintomas mais frequentes.

#### 6.1.6.5. Morte e Vacinação

Na Figura 165 está representada a *query* efetuada para determinar o tempo médio de dias que decorreu entre a vacinação e a morte de um paciente. Pela análise da figura, verifica-se que se recorreu ao comando SELECT média (AVG) do atributo tempo\_ate\_morte que é resultado da *subquery* a1. Nesta *subquery* foi selecionado o número distinto de id\_vaers da tabela **paciente** e o respetivo intervalo de tempo entre o vax\_date da tabela **vacinar** e o data\_died da tabela **paciente**, em que os id\_vaers da tabela **paciente** e da tabela **vacinar** são iguais e em que o

data\_died não é um valor nulo. Além dessas condições, os resultados da *subquery* foram ordenados de forma decrescente consoante o valor do respetivo tempo\_ate\_morte.

Na Figura 166 está representado o resultado do comando apresentado na Figura 165, em que se pode observar que a média do intervalo de tempo entre a vacinação de um paciente e a sua morte é de 10 dias e meio.

```
-- 1. Tempo médio que decorreu entre a vacinação e a morte
SELECT AVG(tempo_ate_morte) AS tempo_medio_ate_morrer
FROM (SELECT DISTINCT p.id_vaers, Diferenca(vr.vax_date, p.data_died) AS tempo_ate_morte
FROM paciente p, vacinar vr
WHERE p.data_died IS NOT NULL
AND p.id_vaers = vr.id_vaers
ORDER BY tempo_ate_morte DESC) a1;
```

**Figura 166.** Query efetuada para determinar o tempo médio que decorreu entre a vacinação de um paciente e a sua morte.

tempo_medio_ate_morrer
10.6912

**Figura 167.** Tempo médio de dias que decorreu entre a vacinação de um paciente e a sua morte.

Na Figura 167 está representada a *query* efetuada para determinar a percentagem de doentes que morreram por número de doses de vacinas tomadas. Pela análise da figura, verifica-se que se recorreu ao comando SELECT das tabelas **paciente** e **vacinar** para se selecionar o número de doses da vacina já tomadas e a razão entre o número de pacientes que morreram e o número de doses tomadas, sendo esta última uma *subquery*. Para isso, foi utilizado o comando COUNT, que devolve um determinado número de linhas em que os id\_vaers da tabela **paciente** e da tabela **vacinar** são iguais, e em que data\_died e o vax\_dose\_Series são NOT NULL. Os resultados da query foram agrupados por nº de doses da vacina tomadas da tabela **vacinar** com o comando GROUP BY vr.vax\_dose\_series, ordenados consoante o valor da respetiva percentagem\_pacientes.

Na Figura 168 está representado o resultado do comando apresentado na Figura 167, em que se pode observar que cerca de 27% dos pacientes que morreram tomaram 2 doses da vacina contra a COVID-19, e aproximadamente 73% tomou apenas uma dose.

```
-- Percentagem dos doentes que morreram por nº de doses de vacinas que tomaram
SELECT vr.vax_dose_series, (COUNT(p.id_vaers)/(SELECT COUNT(p.id_vaers)
FROM paciente p, vacinar vr
WHERE vr.id_vaers = p.id_vaers
AND p.data_died IS NOT NULL AND vr.vax_dose_series IS NOT NULL))*100 AS percentagem_pacientes
FROM paciente p, vacinar vr
WHERE vr.id_vaers = p.id_vaers
AND p.data_died IS NOT NULL AND vr.vax_dose_series IS NOT NULL
GROUP BY vr.vax_dose_series
ORDER BY percentagem_pacientes;
```

**Figura 168.** Query efetuada para determinar a percentagem de doentes que morreram por número de doses de vacinas que tomaram.

vax_dose_series	percentagem_pacientes
2	26.7554
1	73.2446

**Figura 169.** Percentagem de pacientes que morreram e tomaram uma dose da vacina contra a COVID-19 e daqueles que morreram e tomaram duas doses.

Assim sendo, é possível concluir que maior parte das pessoas que morreram após a vacinação, tinham tomado apenas 1 dose da mesma, ou seja, à 1ª dose da vacina contra a COVID-19 está associada uma maior taxa de mortalidade. Quanto ao intervalo de tempo entre a data da vacinação e a data da morte de um paciente, conclui-se que os pacientes morrem pouco tempo depois, menos de 2 semanas, após terem recebido a vacina.

#### 6.1.6.6. Morte e Vacinas

O procedimento MORTE\_VACINAS, presente na Figura 169, recebe como parâmetro o nome da marca da vacina. Dentro do mesmo, está representada a *query* efetuada para determinar a percentagem de mortes associadas à respetiva marca de vacina. Pela análise da figura, verifica-se que se recorreu ao comando SELECT das tabelas **fornecedor**, **vacinar**, **paciente** e **vacina** para

se selecionar a razão entre o número de pacientes e o número de vacinas tomadas por marca, sendo esta última uma *subquery*. Para isso, foi utilizado o comando COUNT, que devolve um determinado número de linhas em que os id\_vaes da tabela **paciente** e da tabela **vacinar** são iguais, bem como o idfornecedor das tabelas **fornecedor** e **vacina**, o idvacina das tabelas **vacina** e **vacinar**, em que data\_died é NOT NULL e em que o parâmetro recebido pelo *procedure* é igual ao atributo vax\_manu da tabela **fornecedor**.

Nas Figuras 171 a 173 estão representados os resultados do *procedure*, após ter sido utilizado o comando CALL, apresentado na Figura 170, em que se pode observar que 5.2 % dos pacientes que tomaram a Moderna morreram, que o mesmo aconteceu a 4.8% dos pacientes que tomaram a Pfizer, bem como a 4.6% dos que tomaram a Janssen.

```
DELIMITER $$
• CREATE PROCEDURE MORTE_VACINAS(vacina VARCHAR (45))
BEGIN
    SELECT (COUNT(vr.id_vaes)/(SELECT COUNT(vr.id_vaes) FROM vacina v, fornecedor f, vacinar vr, paciente p
    WHERE f.vax_manu = vacina AND f.idfornecedor = v.idfornecedor
    AND v.idvacina = vr.idvacina AND vr.id_vaes = p.id_vaes))*100 AS percentagem_mortes_moderna
    FROM vacina v, fornecedor f, vacinar vr, paciente p
    WHERE f.vax_manu = vacina AND f.idfornecedor = v.idfornecedor
    AND v.idvacina = vr.idvacina AND vr.id_vaes = p.id_vaes AND p.data_died IS NOT NULL;
END $$
DELIMITER ;

• -- 2. Percentagem de mortes provocadas pela vacina MODERNA
CALL MORTE_VACINAS ('MODERA');

• -- 3. Percentagem de mortes provocadas pela vacina PFIZER\BIOTECH
CALL MORTE_VACINAS ('PFIZER\BIOTECH');

• -- 4. Percentagem de mortes provocadas pela vacina JASSE
CALL MORTE_VACINAS ('JASSE');
```

**Figura 170.** *Procedure* efetuado para determinar a percentagem de mortes associadas a cada marca de vacina.

percentagem_mortes_moderna
5.2261

**Figura 171.** Percentagem de pacientes que morreram e tomaram a Moderna.

percentagem_mortes_fizer
4.8006

**Figura 172.** Percentagem de pacientes que morreram e tomaram a Pfizer.



percentagem_mortes_jasse
4.5545

**Figura 173.** Percentagem de pacientes que morreram e tomaram a Janssen.

Na Figura 174 estão representadas as *query's* efetuadas para determinar a percentagem de mortes associadas às marcas e respetivos lotes de vacinas. Pela análise da figura, verifica-se que se recorreu ao comando SELECT das tabelas **fornecedor**, **vacinar**, **paciente** e **vacina** para se selecionar o lote da vacina, a respetiva marca e a razão entre o número de pacientes que tomaram uma vacina de uma determinada marca e o total de pacientes vacinados, sendo esta última uma *subquery*. Para isso, foi utilizado o comando COUNT, que devolve um determinado número de linhas em que os id\_vacins da tabela **paciente** e da tabela **vacinar** são iguais, bem como os idfornecedor das tabelas **fornecedor** e **vacina**, os idvacina das tabelas **vacina** e **vacinar** e em que data\_died é NOT NULL Os resultados da *query* foram agrupados pelo lote da vacina com o comando GROUP BY v.vax\_lot e ordenados consoante o valor da respetiva percentagem\_mortes, por ordem decrescente, limitando o resultando apenas aos 10 lotes com maior percentagem de mortes associadas.

Na Figura 175 está representado o resultado do comando apresentado na Figura 174, em que se pode observar que a maior percentagem de pacientes que faleceram teriam tomado as vacinas da Moderna dos lotes 039K20A e 026L20A, sendo o primeiro o que apresenta maior taxa de mortalidade, cerca de 6%. Seguido destes dois lotes da Moderna, está o lote EK9231 da Pfizer com uma taxa de mortalidade de 3.5%.

```
-- 5. Lote de vacina que provocou mais mortes e de que marca é
SELECT v.vax_lot, f.vax_manu, (COUNT(vr.id_vaers)/(SELECT COUNT(vr.id_vaers)
FROM vacina v, fornecedor f, vacinar vr, paciente p
WHERE f.idfornecedor = v.idfornecedor
AND v.idvacina = vr.idvacina
AND vr.id_vaers = p.id_vaers
AND p.data_died IS NOT NULL
AND vax_lot IS NOT NULL))*100 AS percentagem_mortes
FROM vacina v, fornecedor f, vacinar vr, paciente p
WHERE f.idfornecedor = v.idfornecedor
AND v.idvacina = vr.idvacina
AND vr.id_vaers = p.id_vaers
AND p.data_died IS NOT NULL
AND vax_lot IS NOT NULL
GROUP BY v.vax_lot
ORDER BY percentagem_mortes DESC
LIMIT 10;
```

**Figura 174.** Query efetuada para determinar a percentagem de mortes associada a cada marca e respetivo lote da vacina contra a COVID-19.

vax_lot	vax_manu	percentagem_mortes
039K20A	MODERA	6.0383
026L20A	MODERA	4.5655
EK9231	PFIZER\BIOTECH	3.5346
011J20A	MODERA	3.0928
025L20A	MODERA	2.7982
EL1284	PFIZER\BIOTECH	2.5037
EK530	PFIZER\BIOTECH	2.5037
011L20A	MODERA	2.3564
EL1283	PFIZER\BIOTECH	2.2091
03K20A	MODERA	2.2091

**Figura 175.** Percentagem de mortes associadas a cada marca e respetivo lote da vacina contra a COVID-19.

Assim sendo, pode-se concluir que a vacina da Moderna é a que tem associada a maior taxa de mortalidade, seguida da vacina da Pfizer. No entanto, verifica-se que existe uma variabilidade entre os lotes das respetivas marcas e da taxa de mortalidade associada a cada uma, ou seja, existem lotes da vacina Moderna que apresentam menor taxa de mortalidade que alguns lotes da vacina Pfizer. Verifica-se ainda que os lotes com maior taxa de mortalidade associada são também os lotes que provocaram mais sintomas, como visto anteriormente na secção 6.1.5.4.

## 6.2. Triggers

Um *trigger* contém um conjunto de ações em SQL que são ativadas automaticamente sempre que ocorra numa tabela um dos seguintes eventos: INSERT, UPDATE ou DELETE. Para este

projeto, foram desenvolvidos 3 *triggers* que servem para atualizar as tabelas do modelo lógico quando são introduzidos, Figura 176, ou eliminados, Figura 177, dados nas tabelas utilizadas no povoamento do modelo (**temp\_dos\_pacientes**, **temp\_dos\_sintomas**, **temp\_dos\_sintomas2**, **temp\_das\_vacinas**, **temp\_da\_doenca**, etc).

O *trigger* representado na Figura 176 permite atualizar as informações relativas a um novo paciente na tabela **paciente**, em que após um INSERT INTO na tabela **temp\_dos\_pacientes** (AFTER INSERT), a informação é automaticamente adicionada na tabela **paciente**. Para além disto, também é possível eliminar eventuais pacientes que já não sejam relevantes para o estudo em causa, ou que não pretendam ficar registados na base de dados (AFTER DELETE), sendo esta a função do *trigger* presente na Figura 177. Assegura-se, desta forma, que após se ter eliminado a informação da tabela **temp\_dos\_pacientes**, a mesma é também eliminada da tabela **paciente**.

Após se ter adicionado, um novo paciente, é também necessário inserir informação sobre a vacinação do paciente. Para isso, recorre-se a outro *trigger* (Figura 178) muito semelhante ao anterior. É importante referir que a tabela **fornecedor**, nunca será atualizada, uma vez que a base de dados desenvolvida apenas serve para introduzir dados relativos a novos pacientes e não novas vacinas.

Apesar de terem sido implementados 3 *trigger's* que servem para ilustrar a atualização da informação na base de dados, na prática estes não seriam usados. No *dataset*, as informações estavam em 3 documentos Excel, um para o paciente, outro para a vacinação e outro para os sintomas. Se tivesse sido possível importar diretamente os dados desses Excel para o MySQL, teriam sido apenas desenvolvidos 6 *trigger's*, 2 por Excel (1 para INSERT e outro para DELETE), e não seriam necessários implementar tantos *trigger's* quantos os que seriam necessários neste caso.

```
1  -- Tabela paciente
2  -- INSERT na tabela temp_dos_pacientes
3  DELIMITER $$
4  ● CREATE TRIGGER Update_paciente AFTER INSERT ON trabalho_covid.temp_dos_pacientes
5    FOR EACH ROW
6    BEGIN
7      INSERT INTO covid_efeitos.paciente (id_vaers,state,age_yrs,sex,data_died)
8      VALUES ( NEW.VAERS_ID, NEW.STATE,NEW.AGE_YRS,NEW.SEX, NEW.DATEDIED );
9    END $$
10 DELIMITER ;
```

**Figura 176.** *Trigger* efetuado para atualizar a tabela paciente sempre que é feito um INSERT INTO na tabela temp\_dos\_pacientes.

```

14  -- DELETE tabela temp_dos_pacientes
15  DELIMITER $$
16  • CREATE TRIGGER Update_paciente_delete AFTER DELETE ON trabalho_covid.temp_dos_pacientes
17  FOR EACH ROW
18  BEGIN
19      DELETE FROM covid_efeitos.paciente WHERE id_vaers=OLD.VAERS_ID;
20  END $$
21  DELIMITER ;
22

```

**Figura 177.** Trigger efetuado para atualizar a tabela paciente sempre que é feito um DELETE na tabela temp\_dos\_pacientes.

```

36  -- Tabela vacina
37  -- INSERT na tabela temp_das_vacinas
38  DELIMITER $$
39  • CREATE TRIGGER Update_vacina AFTER INSERT ON trabalho_covid.temp_das_vacinas
40  FOR EACH ROW
41  BEGIN
42      INSERT INTO covid_efeitos.vacina (vax_lot, idfornecedor)
43      SELECT NEW.VAX_LOT, f.idfornecedor
44      FROM covid_efeitos.fornecedor f, trabalho_covid.temp_do_fornecedor t, trabalho_covid.temp_das_vacinas v
45      WHERE v.VAERS_ID = t.VAERS_ID
46      AND t.VAX_MANU = f.vax_manu;
47  END $$
48  DELIMITER ;

```

**Figura 178.** Trigger efetuado para atualizar a tabela vacina sempre que é feito um INSERT INTO na tabela temp\_das\_vacinas.

## 7. Interface

A interface desenvolvida vai de encontro com os *trigger's* desenvolvidos na secção anterior, cuja finalidade é atualizar a base de dados sempre que é introduzida ou eliminada informação sobre um paciente. Desta forma, o *mockup*, realizado no software de design Figma, tem como objetivo retratar da forma mais realista possível, um software que consuma o SBD desenvolvido, permitindo introduzir os dados de um paciente de forma intuitiva e organizada, por um profissional de saúde.

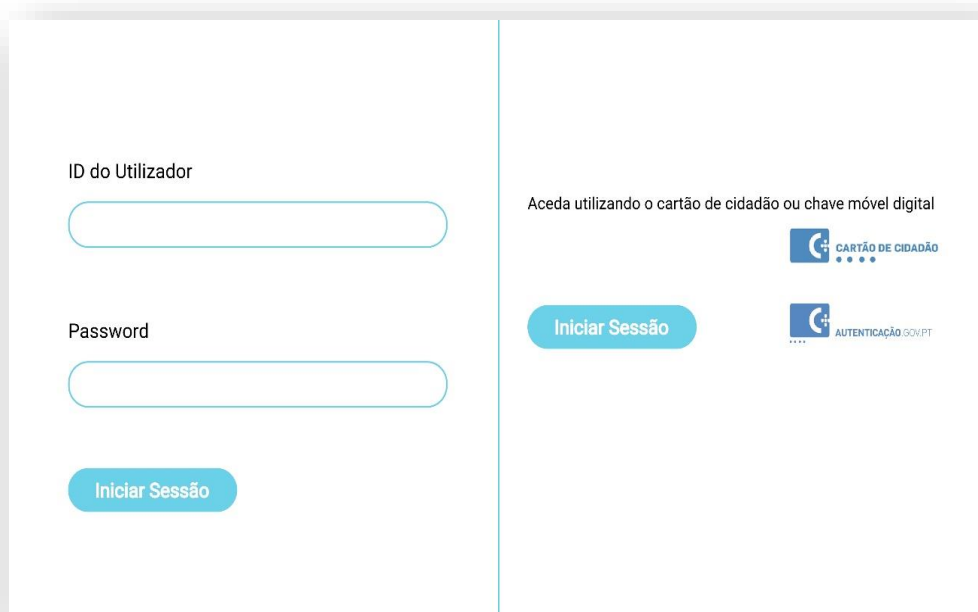
O princípio de funcionamento inicia-se quando, um profissional de saúde faz login na interface com o seu ID de utilizador e respetiva password, ou como alternativa, inicia sessão utilizando o cartão de cidadão ou a chave móvel digital, como se pode observar na Figura 179. Após ter a sessão iniciada, o profissional de saúde terá acesso a uma ficha técnica, onde serão introduzidos os dados e informações do paciente por secções, como está representado na Figura 180. Antes

de iniciar o preenchimento, dentro da secção “Informação do Paciente”, terá que ser verificado se o paciente já está ou não registado na base de dados, ou seja, se já tem o id\_vaers associado, caso contrário, será gerado automaticamente um id\_vaers, pelo sistema informático. É relevante perceber que se o paciente já existir na base de dados, os dados na interface já aparecerão preenchidos e a interface permite que os mesmos sejam alterados.

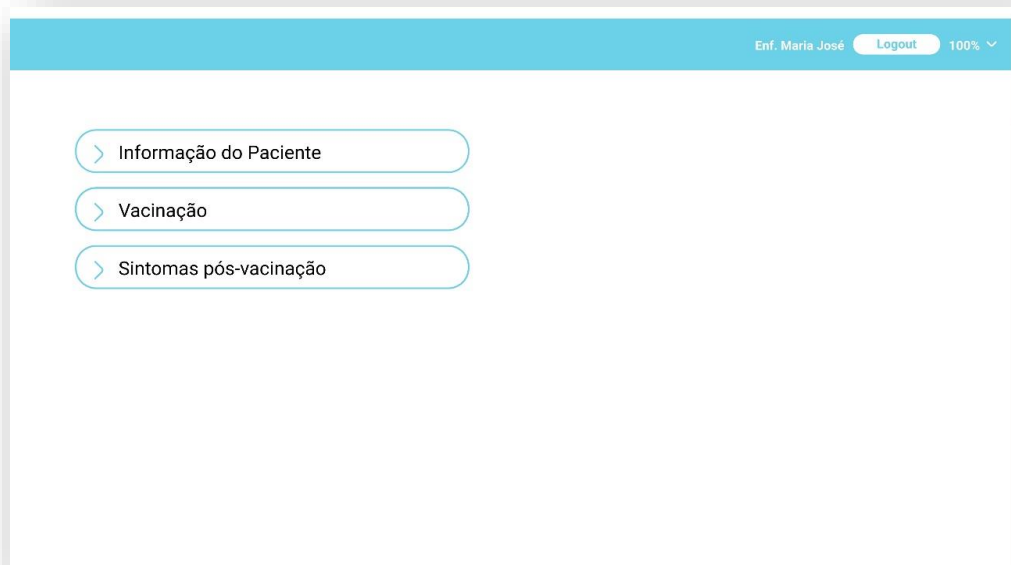
A primeira secção (“Informação do Paciente”) está dividida em três subsecções, uma relativa à informação genérica do paciente, Figura 181, como nome, sexo, data de nascimento e estado onde vive, outra em que o profissional de saúde seleciona as doenças coincidentes com o histórico clínico do paciente, Figura 182 a Figura 184 e outra subsecção onde são confirmados os dados acerca da hospitalização do paciente, se esteve internado e/ou precisou de cuidados médicos e se recorreu ao serviço de urgências, Figura 185. É importante referir que esta última subsecção é referente ao período após o paciente ter sido vacinado contra a COVID-19.

Relativamente à secção “Vacinação”, nesta são introduzidas todas as informações acerca do processo de vacinação, desde a data da vacinação até à forma como foi feita a inoculação da vacina, como se pode observar Figura 186.

Por último na secção “Sintomas pós-vacinação”, Figura 187, após ter registado a data de início dos sintomas, o profissional de saúde toma nota dos sintomas que o paciente relata que sentiu no período pós-vacinação e se este recuperou totalmente dos sintomas.



**Figura 179.** Página inicial da interface, onde é feito o login.



**Figura 180.** Menu da ficha técnica do paciente a ser preenchido pelo profissional de saúde com as secções existentes representadas.

Enf. Maria José Logout 100% ▾

▾ Informação do Paciente





O paciente já se encontra registado?

☐ Sim
 ☒ Não

Nome \_\_\_\_\_

Sexo \_\_\_\_\_

Data de Nascimento \_\_\_\_\_

Estado \_\_\_\_\_

▾ Vacinação

**Figura 181.** Subsecção relativa a informação genérica do paciente, na secção "Informação do paciente".

Information Patient

Progress bar with three icons: person, heart, and building.

**Obesidade**  
☒ Sim ☐ Não

**Colesterol Alto**  
☐ Sim ☒ Não

**Pressão Arterial Alta**  
☐ Sim ☐ Não

**Ansiedade**  
☐ Sim ☐ Não

**Covid-19**  
☐ Sim ☐ Não

**Figura 182** Subsecção relativa ao histórico clínico do paciente, na secção "Informação Paciente".

**Asma**  
☒ Sim ☐ Não

**GERD**  
☐ Sim ☒ Não

**Depressão**  
☐ Sim ☐ Não

**Hiperlipidemia**  
☐ Sim ☐ Não

**Enxaqueca**  
☐ Sim ☐ Não

**Cefaleia**  
☐ Sim ☐ Não

**Hipertensão**  
☐ Sim ☐ Não

**Figura 183.** Subsecção relativa ao histórico clínico do paciente, na secção "Informação Paciente" (Continuação).

**Osteoartritis**  
☐ Sim ☐ Não



**Hipotiroidismo**  
☐ Sim ☐ Não

> Vacinação

> Sintomas pós-vacinação

**Figura 184.** Subsecção relativa ao histórico clínico do paciente, na secção "Informação Paciente" (Continuação).

✓ Informação do Paciente

\*Esta informação é referente ao período pós-vacinação

**Recebeu cuidados médicos/clínicos?**  
☒ Sim ☐ Não

**Deu entrada nas urgências?**  
☐ Sim ☒ Não

**Necessitou de internamento hospitalar?**  
☐ Sim ☐ Não

Se sim, quantos dias esteve internado? \_\_\_\_\_

> Vacinação

**Figura 185.** Subsecção relativa à hospitalização do paciente.



Vacinação

Data da vacinação

Fabricante da vacina

Lote da vacina

Nº de doses previamente administradas

Forma de inoculação

O paciente encontrava-se doente no momento da vacinação?

☐ Sim
 ☒ Não

Setor da entidade onde foi administrada a vacina:

☐ Publico
 ☒ Privado
 ☐ Militar
 ☐ Outro

A injeção foi dada em que braço?

☐ Direito
 ☒ Esquerdo

Figura 186. Secção Vacinação.

Vacinação

Sintomas pós-vacinação

Data do início dos sintomas

Escreva aqui os sintomas do paciente após a toma da vacina

O paciente recuperou totalmente dos sintomas?

☐ Sim
 ☒ Não

Figura 187. Secção Sintomas pós-vacinação.

## 8. Análise crítica e conclusão

Diariamente milhões de pessoas são vacinadas contra a COVID-19, e simultaneamente, estão constantemente a ser identificados novos efeitos adversos provocados pelas vacinas nos pacientes. Deste modo, é de extrema importância para todos nós, que sejam recolhidas, registadas e analisadas informações relativas aos indivíduos que já se vacinaram, de forma a ser possível identificar grupos de risco para uma dada marca de vacina e, consequentemente, evitar que futuramente pacientes desses grupos de risco tomem vacinas desse fornecedor.

Tendo isso em conta e usando como inspiração os modelos de *machine learning*, neste trabalho foi analisado o *dataset* “COVID-19 World Vaccine Adverse Reactions” proveniente do *kaggle*, em que se procurou identificar sintomas graves, percentagens de hospitalização e de mortalidade e concluir acerca dos riscos associados à vacinação.

A amostra analisada continha 32622 pacientes, maioritariamente do sexo feminino e cuja faixa etária predominante compreendia idades entre os 25 e os 65 anos. Verificou-se que todos os pacientes foram sintomáticos, apresentando maioritariamente dor de cabeça, febre e arrepios. Cerca de 9% dos pacientes ficaram hospitalizados e mais de 5% dos pacientes morreram. Cerca de 26% dos indivíduos apresentava pelo menos uma patologia, sendo hipertensão, hiperlipidemia e asma as mais comuns.

Relativamente às hospitalizações, verificou-se que apenas cerca de 21% das pessoas que morreram foram às urgências ou receberam visita do médico, o que leva a deduzir que possivelmente muitas das mortes poderiam ser evitadas caso os pacientes fossem ao médico após manifestarem sintomas, nomeadamente dispneia, não reatividade a estímulos, náuseas e paragem cardíaca.

Dos pacientes que apresentavam patologias, apenas os doentes com hipertensão e hiperlipidemia apresentaram sintomas frequentes mais graves, como dores no peito e falta de ar. Além disso, mais de 20% dos pacientes que morreram tinham hiperlipidemia e/ou hipertensão e cerca de 8% dos pacientes que morreram tinham hipotireoidismo e/ou GERD. Assim sendo, é possível concluir que estas 4 doenças são doenças de risco para pacientes que se vacinam contra a COVID-19.

Relativamente às vacinas, verificou-se que apesar das vacinas da Moderna provocarem maior percentagem de sintomas, eram as vacinas da Jassen as responsáveis pelo aparecimento de

sintomas mais graves com maior frequência. Ao nível de mortalidade, as vacinas da Moderna, Pfizer e Jassen apresentam aproximadamente as mesmas taxas de mortalidade (entre 3-5%).

Com o estudo realizado, apurou-se também que as vacinas da COVID-19 provocam mortes maioritariamente em idosos e bebés, pelo que se pode afirmar que estes são grupos de risco para vacinação contra a COVID-19. Verificou-se ainda que a taxa de mortalidade teve uma incidência muito superior em homens do que em mulheres. Assim, homens com mais de 65 anos e bebés do sexo masculino são grupos de risco.

Sintetizando, com este trabalho, verificou-se que bebés e homens com mais de 65 anos com hiperlipidemia, hipertensão, hipotireoidismo ou GERD são os principais grupos de risco na vacinação. Sendo assim, algumas medidas a implementar de forma a diminuir as taxas de mortalidade seriam assegurar um acompanhamento especial destes pacientes após terem sido vacinados, obrigatoriedade de receberem ajuda médica caso sintomáticos e se possível não tomarem vacinas da Jassen, para diminuir a probabilidade de ocorrerem sintomas mais graves.

Por último, apesar do trabalho realizado ter cumprido os objetivos propostos, algumas propostas de melhorias seriam ter efetuado mais comandos SQL para obter mais conclusões e mais informações sobre como prevenir e/ou diminuir os efeitos adversos das vacinas e também um pré-tratamento mais profundo do *dataset*, de forma a eliminar pacientes cujos dados foram inseridos incorretamente e consequentemente interferiram com as conclusões finais. É ainda de realçar que se poderiam ter utilizado informações do *dataset* que se optou por não usar, uma vez, que quanto mais dados se tem sobre cada paciente, mais precisas e rigorosas são as conclusões obtidas.

## 9. Bibliografia

- [1] ANTÓNIO ABELHA, “INTRODUÇÃO ÀS BASES DE DADOS – AULA 2”, Universidade do Minho, Braga, 2021.
- [2] *kaggle*. <https://www.kaggle.com/ayushggarg/covid19-vaccine-adverse-reactions?select=2021VAERSVAX.csv>. Acedido a 24 abr de 2021.
- [3] VAERS. [https://vaers.hhs.gov/docs/VAERSDataUseGuide\\_June2017.pdf](https://vaers.hhs.gov/docs/VAERSDataUseGuide_June2017.pdf). Acedido a 29 mai de 2021.
- [4] MedDRA. <https://www.meddra.org/how-to-use/support-documentation/english/welcome>. Acedido a 29 mai de 2021.
- [5] DANIELA OLIVEIRA, “BASES DE DADOS CLÍNICAS E DE GESTÃO HOSPITALAR TP3”, Universidade do Minho, Braga, 2021.
- [6] ANTÓNIO ABELHA, “INTRODUÇÃO ÀS BASES DE DADOS – AULA 3”, Universidade do Minho, Braga, 2021.