Politecnico di Milano

Systems and methods for big and unstructured data

SMBUD project: second delivery

Group 24

Basso Paolo 10783951

Aiello Andrea 10863133

Borsatto Andrea 10628989

Cavalli Dario 10820532

Petriconi Emanuele 10577000

Academic Year 2021–2022

Specifications

The scope of this project is to design and store the information needed to support a certification app for COVID-19.

The database includes detailed information about each person, vaccine and authorized body that administers the vaccine doses.

For each individual, their personal data must be recorded (first and last name, telephone number, email and birthdate) along with information about their residence (address, country, city, etc...). Every person stored is identified by a fiscal code, that can be deduced from the data listed above.

The authorized bodies, in charge of administering the vaccines, can be of different types, going from hospitals to vaccination hub and they are further divided into different departments. The data stored regards their name, type, departments ("Immunology", "Epidemiology", "Covid Emergency") and location.

Four vaccines are stored and for each of them the dataset maintains their name, minimum number of doses to be effective, type, website and even some information about the manufacturer (name, website and telephone number).

People can get vaccine shots or take tests to obtain a certificate that verifies their negativity to a Covid-19 infection.

When a person gets a vaccine shot, the database must record the vaccine inoculated, the dose number, the authorized body which delivered the vaccine, time and date the vaccine was delivered, the lot number, the vaccine production date, the staff members who performed the shot and a single emergency contact designated by the patient (name and telephone would suffice).

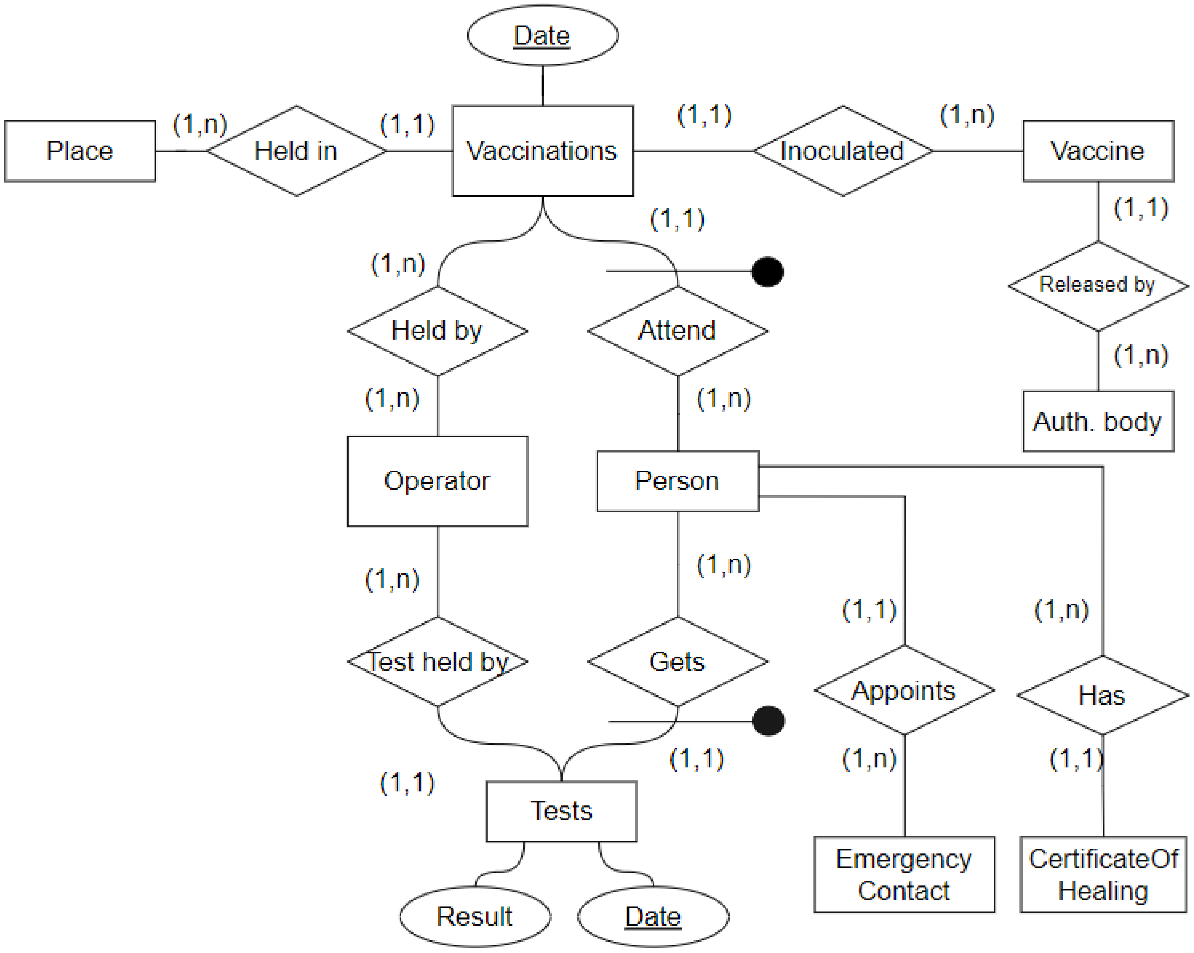
When a person does a test, it must record: the authorized body which performed the test, time and date test was performed, the operator that performed the test, the type (rapid, molecular, or serologic), the result and again a single emergency contact in case the test result is positive, and the person test is unreachable.

A certificate can be also obtained after a negative test result subsequently to a Covid-19 infection with a duration on 6 months.

Check Validity

It is the duty of the verification apps to decide whether a certificate is valid or not. This is a mandatory requirement because the rules of when a certificate is valid or not will likely change in the future (as they did in the past) and it is not feasible to edit all certificates. For this reason, updating the certificate verification app must be forced. In this particular case the validity of a certificate is proved by checking the date a vaccine was administered, a test was taken or a negative result after a recovery from the infection has been delivered.

Conceptual Model



The ER model translates into two collections in the MongoDB database, one for the certificates and the other for the authorized bodies. In APPENDIX A there are some tables which defines the structure of documents.

Dataset Generation

We created a python script which uses a library that interacts with randomuser.me API to generate random people data.

The script connects directly to a database using PyMongo and performs queries against it. At the end the script also exports the created database as json files.

In the script we first generate first a fixed set of vaccines, places, and staff members and then we populate the authorized collection with random data. Lastly, we populate the certificates collection with new random data and the data created before (vaccines, places, staff members, authorized bodies).

Assumptions

- We supposed to create the dataset for the full 2021, like if the dataset was created the 31st of December. This means that, for example, some people could not have yet completed the vaccination cycle because their second dose is scheduled for the 2022. No vaccinations nor tests performed in a year different from 2021 are present.
- If a person starts the vaccination cycle, then it will end it (no dropouts after the first dose).
- If a person results positive, it will take another test 10 days after.
- There are 3000 certificates.
- There are 100 places where people can perform tests or vaccinations.
- There are 500 staff members that can perform vaccinations and tests.
- There are 50 authorized bodies which issue the certificates.
- All the doses are made with the same vaccine.
- Some people get a booster dose.

In APPENDIX B all parameters used for the dataset generation are reported.

Importing the dataset

In the delivery there is a folder called *dumps* where each json file corresponds to a collection. To import the dataset using MongoDB Compass, it is needed to create a collection in the database for each file and then using the import utility selecting the corresponding json as source.

Queries

1. **Get top 10 places of vaccination (hospitals, vaccination centers, etc...) which delivered more vaccination doses**

```

db.certificates.aggregate([

// Filter people with a vaccination

{$match: {

’vaccinations.0’: {$exists: ’True’}

}},

// Unwind vaccinations

{$unwind: {

path: ’$vaccinations’

}},

// Group by place name

{$group: {

_id: ’$vaccinations.place.name’,

nVaccinations: {$sum: 1}

}},

// Sort by number of vaccinations DESC

{$sort: {

nVaccinations: -1

}},

// Get top 10

{$limit: 10}

])
```

2. Calculate the percentage of every vaccine used

```
    db.certificates.aggregate([

// Match vaccinated people

{$match: {

’vaccinations.0’: {$exists: ’True’}

}},

// Filter only the first dose

{ $project: {

"vaccinations": { $first: "$vaccinations" }

}},

// Count by vaccine name

{ $group: {

_id: "$vaccinations.vaccine.name",

nPeople: { $sum: 1 }

}},

// Find total and push counts to a vaccines array

{$group: {

_id:null,

total: {$sum:"$nPeople"},

vaccines: { $push: {name: "$_id", nPeople:"$nPeople"} }

}},

// Unwind vaccines array

{$unwind: {

path: "$vaccines"

}},

// Find percentage and project

{$project: {

_id:"$vaccines.name",

"percentage": {$multiply:[{$divide:["$vaccines.nPeople","$total"]},100]}

}},

])
```

3. Top 3 authorized bodies per country that delivered more doses

```
    db.certificates.aggregate([

// Filter people who got a vaccination

{ $match: {

’vaccinations.authorized_body_id’: {$exists: ’True’}

}},

// Unwind vaccinations

{ $unwind: ’$vaccinations’ },

// Count vaccination doses by authorized_body_id

{ $group: {

_id: ’$vaccinations.authorized_body_id’,

nVaccinationDoses: { $sum: 1 }

}},

// Sort by number of vaccination doses

{ $sort: {

nVaccinationDoses: -1

}},

{ $lookup: {

from: ’authorized_bodies’,

localField: ’_id’,

foreignField: ’_id’,

as: ’authorized_body’

}},

// Unwind authorized bodies

{ $unwind: ’$authorized_body’ },

// Group by country

{ $group: {

_id: ’$authorized_body.location.country’,

authorized_bodies: {$push: {

_id: "$authorized_body._id",

name: "$authorized_body.name",

nVaccinationDoses: "$nVaccinationDoses"

}}

}},

// Keep only top 3

{$project: {

_id: "$_id",

"authorized_bodies": {$slice: ["$authorized_bodies", 3]}

}}

])
```

4. **Determines if Leroy Payne (fiscalCode: "LRYPYN98A30R790Z") has a valid certificate, supposing that a certificate is valid if he is not currently positive and one of the following conditions is true:**

   - He got the first dose of a vaccine more than 15 days ago (before 2021-12-25)
   - He got a full vaccination cycle less than 9 months ago (270 days) and more than 15 days ago (from 2021-04-05 to 2021-12-16)
   - He took a negative test less than 48h ago (after 2021-12-30T00:00:00)
   - Healed from COVID less than 6 months ago (180 days, after 2021-06-04)

```
    db.certificates.aggregate([

  // Match the right person

  {

    $match: {

      fiscal_code: "LLYLMA84D02P040R", // LRYPYN98A30R790Z

    },

  },

  // Add useful fields

  { $addFields: { n_doses: { $size: "$vaccinations" } } },

  {

    $addFields: { total_doses: { $max: "$vaccinations.vaccine.total_doses" } },

  },

  { $addFields: { last_dose: { $first: "$vaccinations" } } },

  { $addFields: { last_test: { $first: "$tests" } } },

  // Project

  {

    $project: {

      fiscal_code: 1,

      n_doses: 1,

      total_doses: 1,

      last_dose: 1,

      last_test: 1,

      tests: 1,

      healings: 1,

    },

  },

  // Match not currently positive

  {

    $match: {

      $or: [

        { "last_test.result": "negative" },

        { last_test: { $exists: false } },

      ],

    },

  },

  // Match conditions

  {

    $match: {

      $or: [

        // 1. He got the first dose of a vaccine more than 15 days ago (before 2021-12-25)

        {

          "last_dose.datetime": {

            $lt: ISODate("2021-12-25T00:00:00.000Z"),

          },

        },

        // 2. He got a full vaccination cycle less than 9 months ago (270 days) and more than 15 days ago

        {

          $and: [

            { $expr: { $gte: ["$n_doses", "$total_doses"] } },

            {

              "last_dose.datetime": {

                $gte: ISODate("2021-04-05T00:00:00.000Z"),

              },

            },

            {

              "last_dose.datetime": {

                $lte: ISODate("2021-12-17T00:00:00.000Z"),

              },

            },

          ],

        },

        // 3. He took a negative test less than 48h ago

        {

          "last_test.datetime": {

            $gte: ISODate("2021-12-30T00:00:00.000Z"),

          },

        },

        // 4. Healed from COVID less than 6 months ago (180 days)

        {

          healings: {

            $elemMatch: {

              end: {

                $gte: ISODate("2021-06-04T00:00:00.000Z"),

              },

            },

          },

        },

      ],

    },

  },

]);
```

5. **Find fiscal code, first name and last name of the people who got a booster dose** (an additional dose after the normal vaccination cycle)

```
    db.certificates.aggregate([

// Match people who got a vaccination

{$match: {

’vaccinations.0’: {$exists: ’True’}

}},

// Project interesting fields

{$project: {

"first_name": 1,

"last_name": 1,

"fiscal_code": 1,

"vaccinations": 1

}},

// Match people who did more doses than the vaccine.total_doses

{$match: {

$expr: { $gt: [

{$size: ’$vaccinations’},

{$max: ’$vaccinations.vaccine.total_doses’}

]}

}},

// Project required fields

{$project: {

"first_name": 1,

"last_name": 1,

"fiscal_code": 1

}},

])
```

Commands

1. **Add a certificate to the database**

```
db.certificates.insertOne({

    first_name: "Mario",

    last_name: "Rossi",

    telephone: "051-806-7094",

    email: "mario.rossi@example.com",

    birth_datetime: ISODate("1984-10-22T06:10:00.000Z"),

    location: {

        address: "via delle primule, 56",

        country: "Italy",

        city: "Rovereto",

        zip_code: 26010,

        coordinates: {

            latitude: "45.8905",

            longitude: "11.0397",

        },

    },

});
```

2. Add an authorized body to the database

```
    db.authorized_bodies.insertOne({

    name: "provincial health office 1",

    type: "provincial health office",

    telephone: "10-988-7777",

    location: {

        address: "via delle azalee 88",

        country: "Italy",

        city: "Sermoneta",

        zip_code: 04013,

    },

});
```

3. Add a test to Mario Rossi (uuid: “61b63f7f98b9ace842754bf0”)

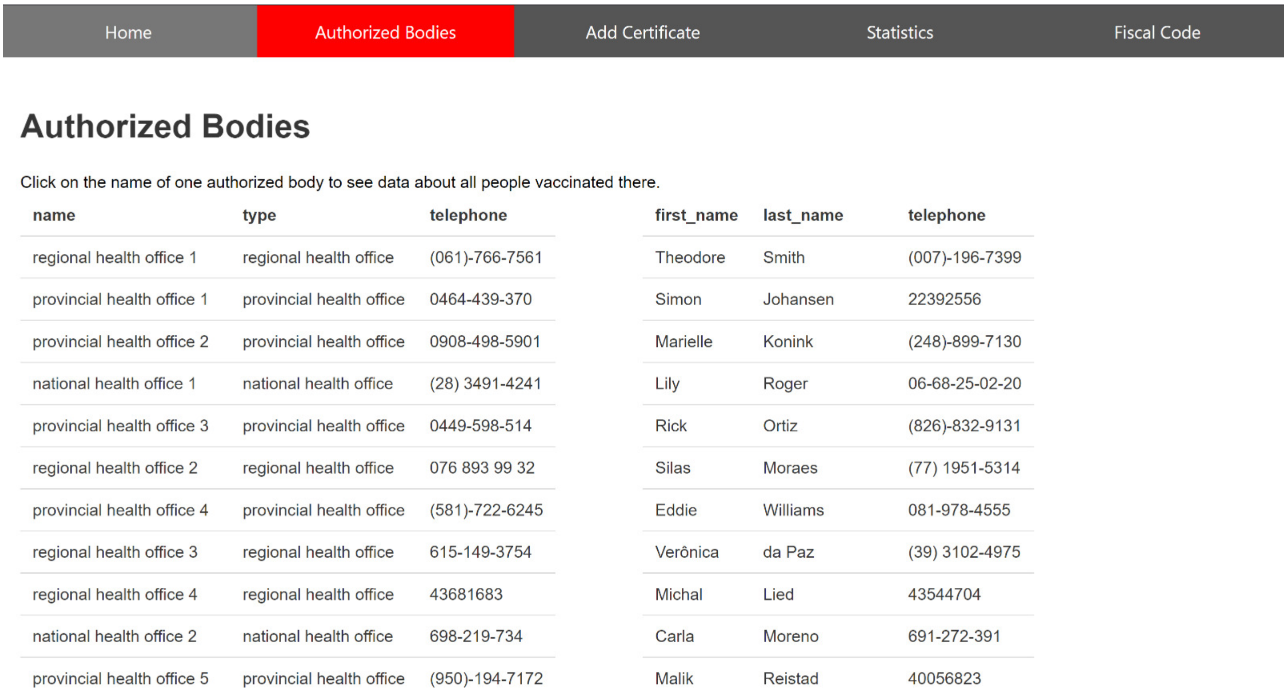
```
    db.authorized_bodies.insertOne({

    name: "provincial health office 1",

    type: "provincial health office",

    telephone: "10-988-7777",

    location: {

        address: "via delle azalee 88",

        country: "Italy",

        city: "Sermoneta",

        zip_code: 04013,

    },

});
```

4. **Add a vaccination to Mario Rossi** (uuid: “61b63f7f98b9ace842754bf0”)

```
db.certificates.updateOne(

{ _id: ObjectId("61b63f7f98b9ace842754bf0") },

{

    $push: {

        vaccinations: {

            datetime: ISODate("2021-09-29T08:04:45.000Z"),

            vaccine: {

                name: "COVID-19 Vaccine Janssen",

                total_doses: 1,

                type: "viral_vector",

                ema_url:

                    "https://www.ema.europa.eu/en/medicines/human/EPAR/covid-19-vaccine-janssen",

                manufacturers: [

                    {

                        name: "Johnson & Johnson",

                        website: "https://www.jnj.com/",

                        telephone: "(732) 524-0400",

                    },

                ],

                lot: 385,

                production_date: ISODate("2021-03-11T11:55:09.000Z"),

            },

            staff: [

                {

                    first_name: "Roseneide",

                    last_name: "Porto",

                    role: "nurse",

                    telephone: "(06) 7086-2238",

                        email: "roseneide.porto@example.com",

                        birth_date: ISODate("1975-09-15T00:00:00.000Z"),

                        location: {

                            address: "2397 Rua Sete de Setembro ",

                            country: "Brazil",

                            city: "Salvador",

                            zip_code: 43748,

                            latitude: 80.7739,

                            longitude: -170.5816,

                        },

                    },

                ],

                authorized_body_id: ObjectId("61b638eead9e15bece773fb7"),

                place: {

                    name: "hospital 43",

                    type: "hospital",

                    telephone: "0179-3597946",

                    location: {

                        address: "440 Gr ner Weg",

                        country: "Germany",

                        city: "Crimmitschau",

                        zip_code: 26168,

                        latitude: 18.2513,

                        longitude: -34.8376,

                    },

                },

            },

        },

    }

```

```
);
```

UI

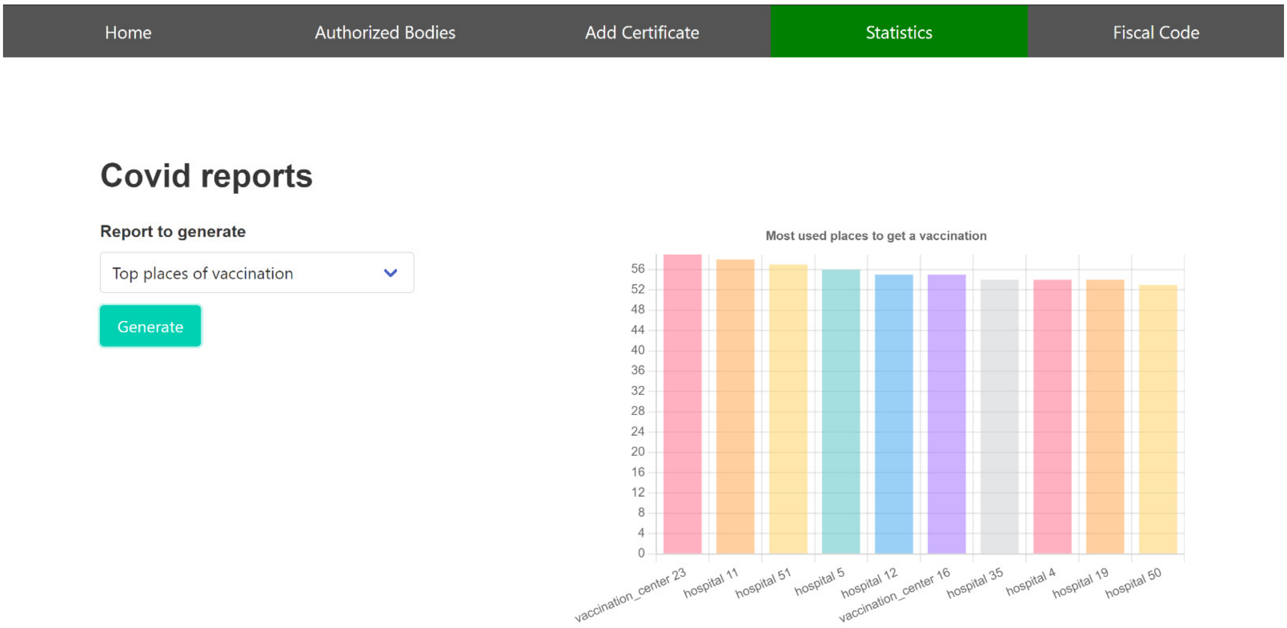


Authorized Bodies

Click on the name of one authorized body to see data about all people vaccinated there.

name	type	telephone
regional health office 1	regional health office	(061)-766-7561
provincial health office 1	provincial health office	0464-439-370
provincial health office 2	provincial health office	0908-498-5901
national health office 1	national health office	(28) 3491-4241
provincial health office 3	provincial health office	0449-598-514
regional health office 2	regional health office	076 893 99 32
provincial health office 4	provincial health office	(581)-722-6245
regional health office 3	regional health office	615-149-3754
regional health office 4	regional health office	43681683
national health office 2	national health office	698-219-734
provincial health office 5	provincial health office	(950)-194-7172

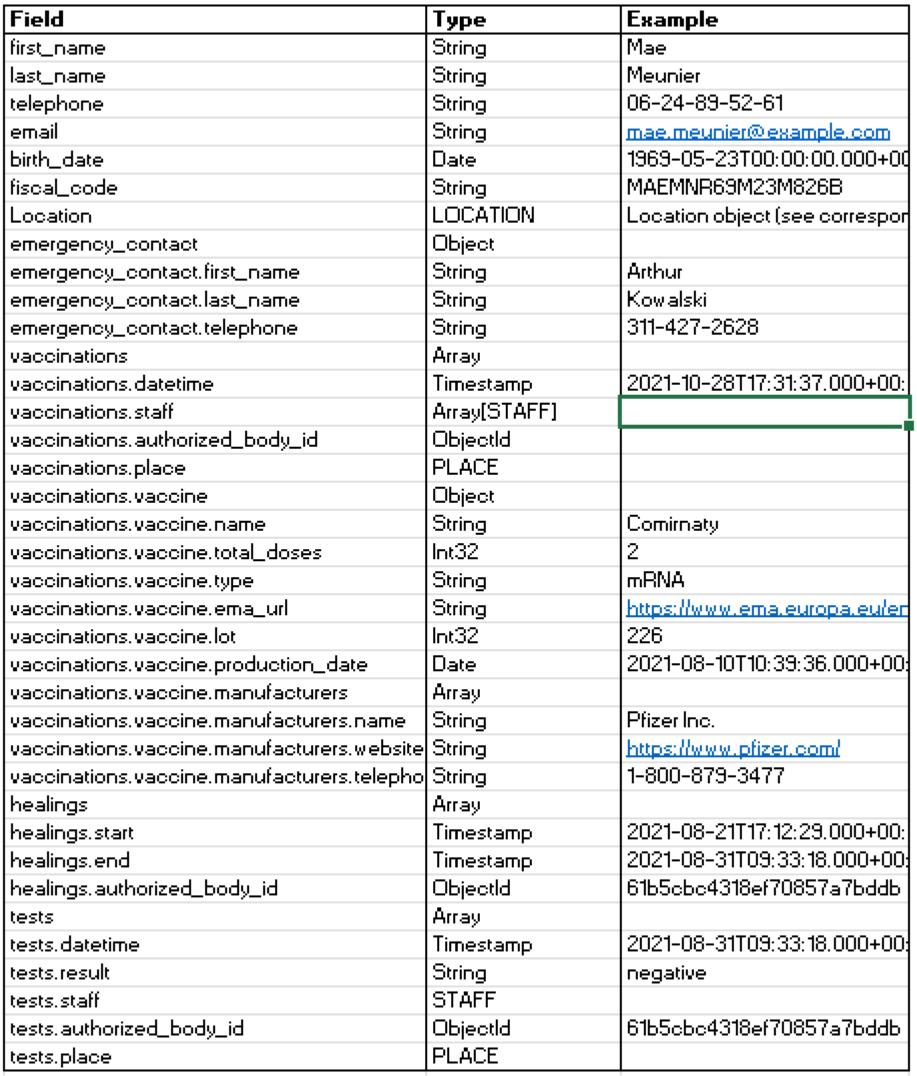
first_name	last_name	telephone
Theodore	Smith	(007)-196-7399
Simon	Johansen	22392556
Marielle	Konink	(248)-899-7130
Lily	Roger	06-68-25-02-20
Rick	Ortiz	(826)-832-9131
Silas	Moraes	(77) 1951-5314
Eddie	Williams	081-978-4555
Verônica	da Paz	(39) 3102-4975
Michal	Lied	43544704
Carla	Moreno	691-272-391
Malik	Reistad	40056823



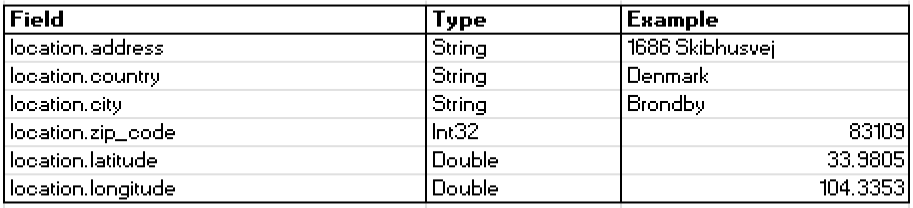
Covid reports

Report to generate: Top places of vaccination

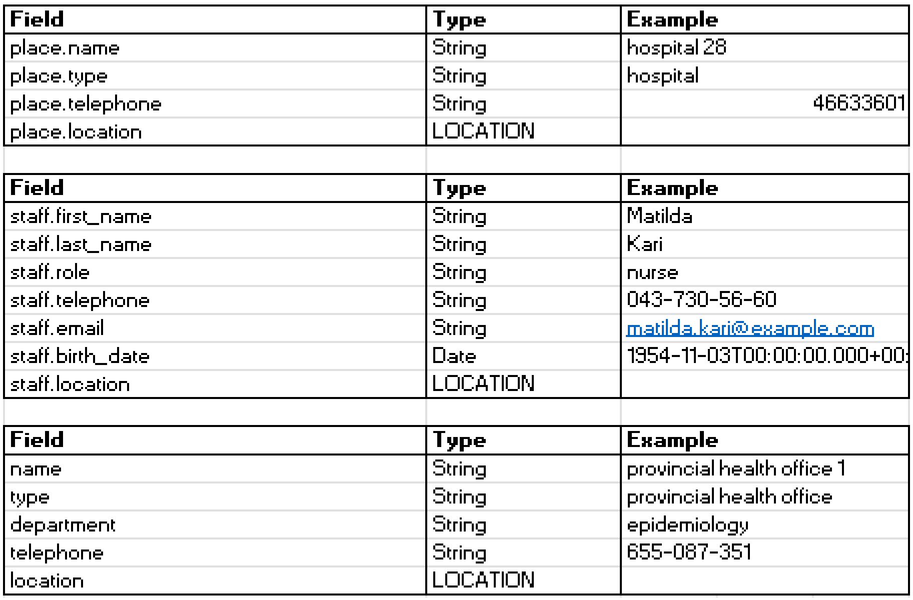
Appendix A



Field	Type	Example
first_name	String	Mae
last_name	String	Meunier
telephone	String	06-24-89-52-61
email	String	mae.meunier@example.com
birth_date	Date	1969-05-23T00:00:00.000+00
fiscal_code	String	MAEMNR69M23M826B
Location	LOCATION	Location object (see correspor
emergency_contact	Object	
emergency_contact.first_name	String	Arthur
emergency_contact.last_name	String	Kowalski
emergency_contact.telephone	String	311-427-2628
vaccinations	Array	
vaccinations.datetime	Timestamp	2021-10-28T17:31:37.000+00:
vaccinations.staff	Array[STAFF]	
vaccinations.authorized_body_id	ObjectId	
vaccinations.place	PLACE	
vaccinations.vaccine	Object	
vaccinations.vaccine.name	String	Comirnaty
vaccinations.vaccine.total_doses	Int32	2
vaccinations.vaccine.type	String	mRNA
vaccinations.vaccine.ema_url	String	https://www.ema.europa.eu/en
vaccinations.vaccine.lot	Int32	226
vaccinations.vaccine.production_date	Date	2021-08-10T10:39:36.000+00:
vaccinations.vaccine.manufacturers	Array	
vaccinations.vaccine.manufacturers.name	String	Pfizer Inc.
vaccinations.vaccine.manufacturers.website	String	https://www.pfizer.com/
vaccinations.vaccine.manufacturers.telepho	String	1-800-879-3477
healings	Array	
healings.start	Timestamp	2021-08-21T17:12:29.000+00:
healings.end	Timestamp	2021-08-31T09:33:18.000+00:
healings.authorized_body_id	ObjectId	61b5cbc4318ef70857a7bddb
tests	Array	
tests.datetime	Timestamp	2021-08-31T09:33:18.000+00:
tests.result	String	negative
tests.staff	STAFF	
tests.authorized_body_id	ObjectId	61b5cbc4318ef70857a7bddb
tests.place	PLACE	



Field	Type	Example
location.address	String	1686 Skibhusvej
location.country	String	Denmark
location.city	String	Brondby
location.zip_code	Int32	83109
location.latitude	Double	33.9805
location.longitude	Double	104.3353

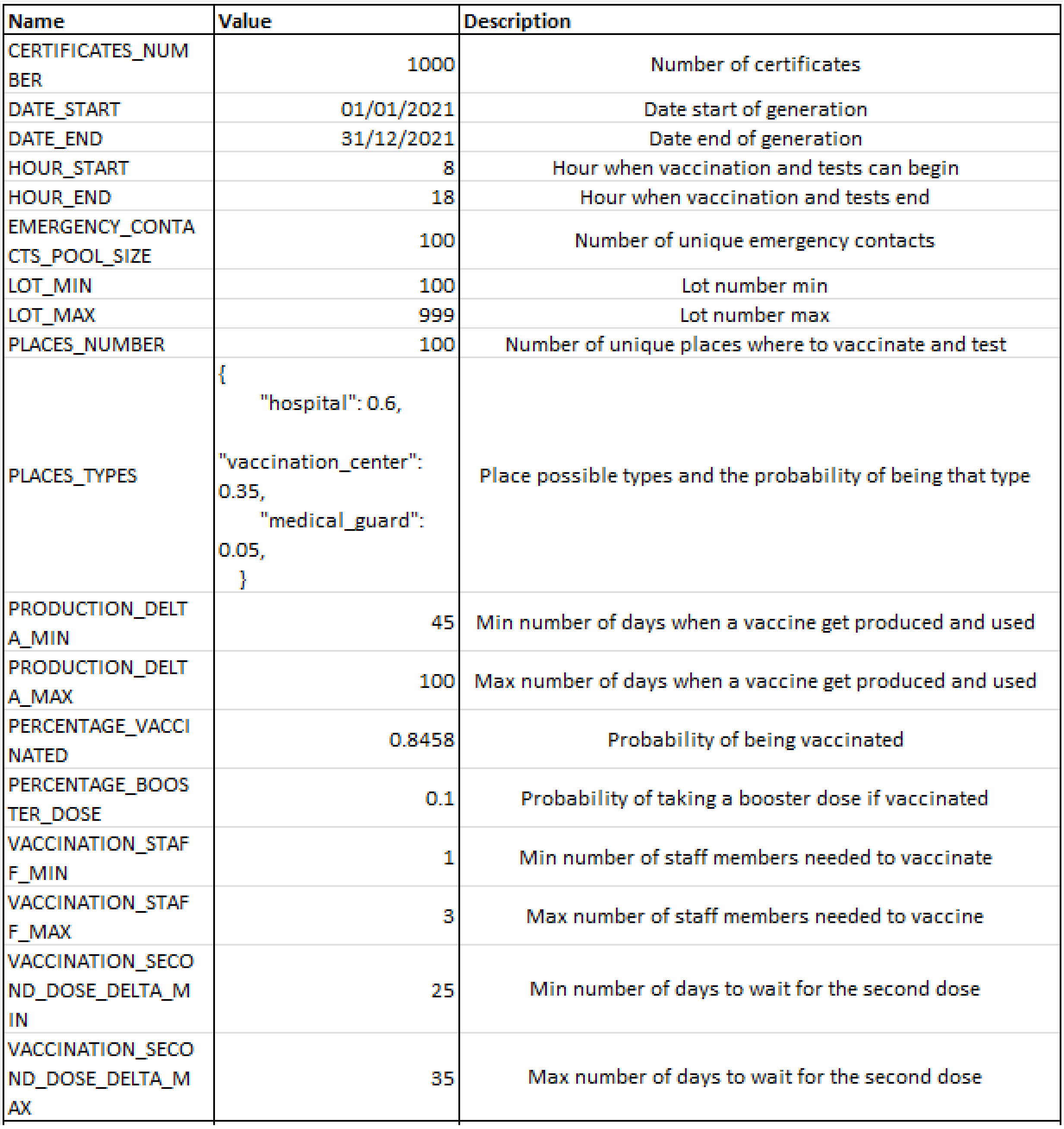


Field	Type	Example
place.name	String	hospital 28
place.type	String	hospital
place.telephone	String	46633601
place.location	LOCATION	

Field	Type	Example
staff.first_name	String	Matilda
staff.last_name	String	Kari
staff.role	String	nurse
staff.telephone	String	043-730-56-60
staff.email	String	matilda.kari@example.com
staff.birth_date	Date	1954-11-03T00:00:00.000+00:
staff.location	LOCATION	

Field	Type	Example
name	String	provincial health office 1
type	String	provincial health office
department	String	epidemiology
telephone	String	655-087-351
location	LOCATION	

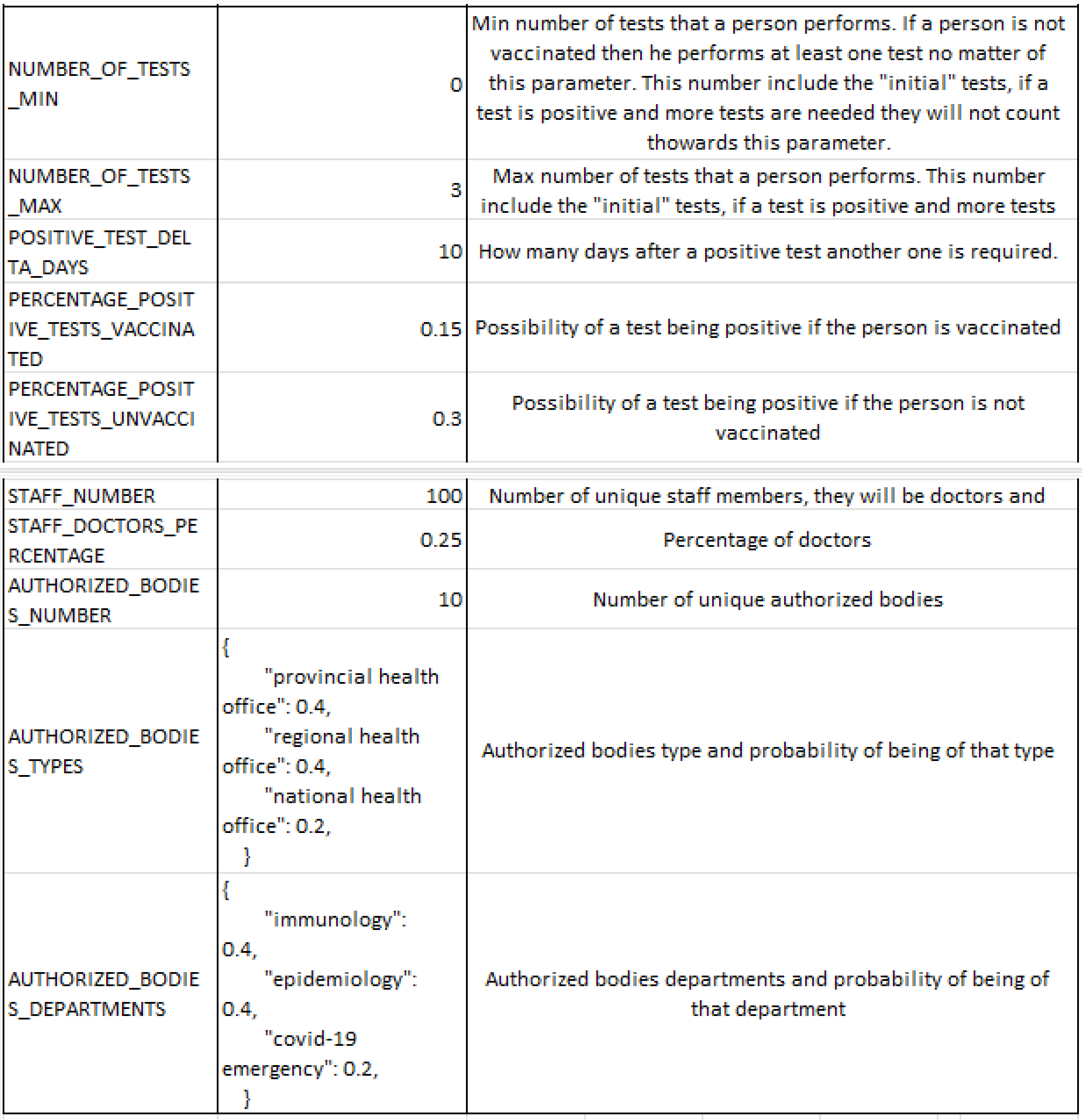
Appendix B



Name	Value	Description
CERTIFICATES_NUMBER	1000	Number of certificates
DATE_START	01/01/2021	Date start of generation
DATE_END	31/12/2021	Date end of generation
HOUR_START	8	Hour when vaccination and tests can begin
HOUR_END	18	Hour when vaccination and tests end
EMERGENCY_CONTACTS_POOL_SIZE	100	Number of unique emergency contacts
LOT_MIN	100	Lot number min
LOT_MAX	999	Lot number max
PLACES_NUMBER	100	Number of unique places where to vaccinate and test

PLACES_TYPES	{
	"hospital": 0.6,
	"vaccination_center": 0.35,
	"medical_guard": 0.05,
	}
		Place possible types and the probability of being that type

PRODUCTION_DELTA_MIN	45	Min number of days when a vaccine get produced and used
PRODUCTION_DELTA_MAX	100	Max number of days when a vaccine get produced and used
PERCENTAGE_VACCINATED	0.8458	Probability of being vaccinated
PERCENTAGE_BOOSTER_DOSE	0.1	Probability of taking a booster dose if vaccinated
VACCINATION_STAFF_MIN	1	Min number of staff members needed to vaccinate
VACCINATION_STAFF_MAX	3	Max number of staff members needed to vaccine
VACCINATION_SECOND_DOSE_DELTA_MIN	25	Min number of days to wait for the second dose
VACCINATION_SECOND_DOSE_DELTA_MAX	35	Max number of days to wait for the second dose



NUMBER_OF_TESTS_MIN	0	Min number of tests that a person performs. If a person is not vaccinated then he performs at least one test no matter of this parameter. This number include the "initial" tests, if a test is positive and more tests are needed they will not count thowards this parameter.
NUMBER_OF_TESTS_MAX	3	Max number of tests that a person performs. This number include the "initial" tests, if a test is positive and more tests
POSITIVE_TEST_DELTA_DAYS	10	How many days after a positive test another one is required.
PERCENTAGE_POSITIVE_TESTS_VACCINATED	0.15	Possibility of a test being positive if the person is vaccinated
PERCENTAGE_POSITIVE_TESTS_UNVACCINATED	0.3	Possibility of a test being positive if the person is not vaccinated
STAFF_NUMBER	100	Number of unique staff members, they will be doctors and
STAFF_DOCTORS_PERCENTAGE	0.25	Percentage of doctors
AUTHORIZED_BODIES_NUMBER	10	Number of unique authorized bodies

AUTHORIZED_BODIES_TYPES	{
	"provincial health office": 0.4,
	"regional health office": 0.4,
	"national health office": 0.2,
	}
		Authorized bodies type and probability of being of that type

AUTHORIZED_BODIES_DEPARTMENTS	{
	"immunology": 0.4,
	"epidemiology": 0.4,
	"covid-19 emergency": 0.2,
	}
		Authorized bodies departments and probability of being of that department