Databases
Informatics and Computing Engineering Department
Faculty of Engineering, University of Porto

Vaccination Monitoring Programme

A Research and Application on a Minimal Vaccination Campaign Database

Luís Tavares
Márcio Duarte
Nuno Costa

email: {up201809679, up201909936, up201906272}@fe.up.pt

2nd class, Group 205

April 4, 2021

Contents

1. Context

Nowadays, vaccination presents itself as one of the most critical problems in modern society. Many countries, over the years, have developed a National Vaccination Programme to prevent epidemics and improve citizens' health care. Portugal has had one since 1965, where a universal and free programme was born. Programmes like these require large amounts of data and adequate data structures to store reliable information. This project aims to describe a minimal vaccination campaign database.

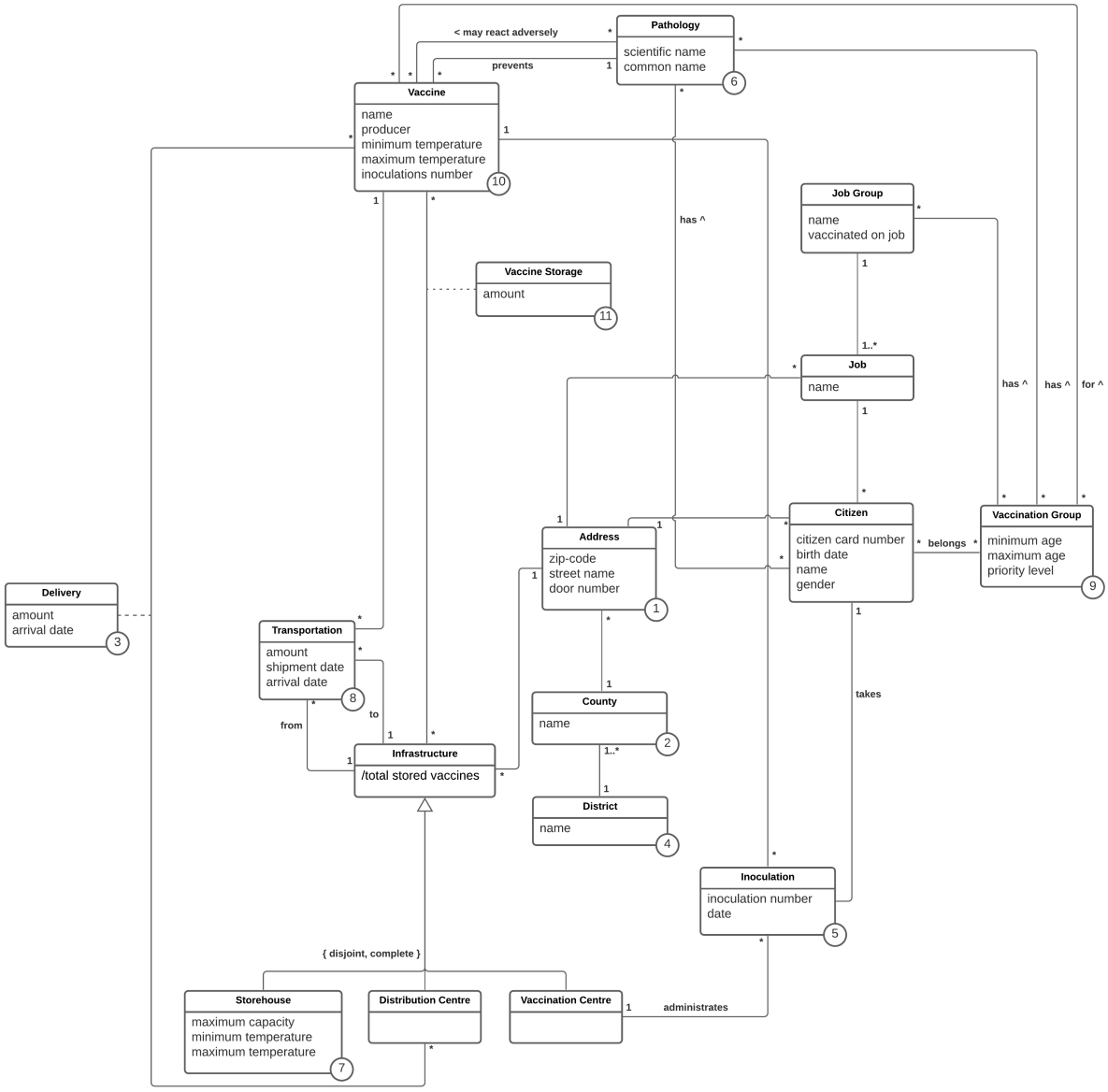
In this vaccination campaign database, citizens and vaccines are central entities. Citizens are identified by their citizen card number and hold sociodemographic data. On the other hand, vaccines provide information about the pathology it prevents and the ones it might react adversely to. Distributions centres store produced vaccines. Later, transportations between distribution centres and storehouses or vaccination centres may occur. These transportations only carry a single type of vaccine due to conservation restrictions (*e.g.* storage temperature range). After arriving at a storehouse, vaccines remain preserved in their storage conditions. Storehouses act as a middleware between a distribution centre and a vaccination centre. However, transportations may occur directly between a distribution centre and a vaccination centre. As some shipments have a long route, vaccine packages might travel between multiple storehouses to reach the final destination, allowing transportations to arise between two storehouses.

Vaccinating a population is particularly challenging. To ease the vaccination process, citizens have a defined vaccination group. A group for a vaccine is responsible for delineating its allocation criteria and its priority. After having a correctly structured collection of vaccination groups, the vaccination process starts. During the vaccination process, groups might suffer changes, allowing large groups to split into smaller ones.

The vaccination of a citizen takes place at a vaccination centre. The vaccination centre assigned to a citizen is dependent on sociodemographic data. For general purposes, the vaccination centre of a citizen is the closest to their address. However, the allocation criteria change according to the citizen vaccination group. An inoculation registry is maintained, holding information about the citizen, the vaccine and the vaccination centre.

2. Conceptual Modeling

2.1 UML Diagram



2.2 Class Definition and Restrictions

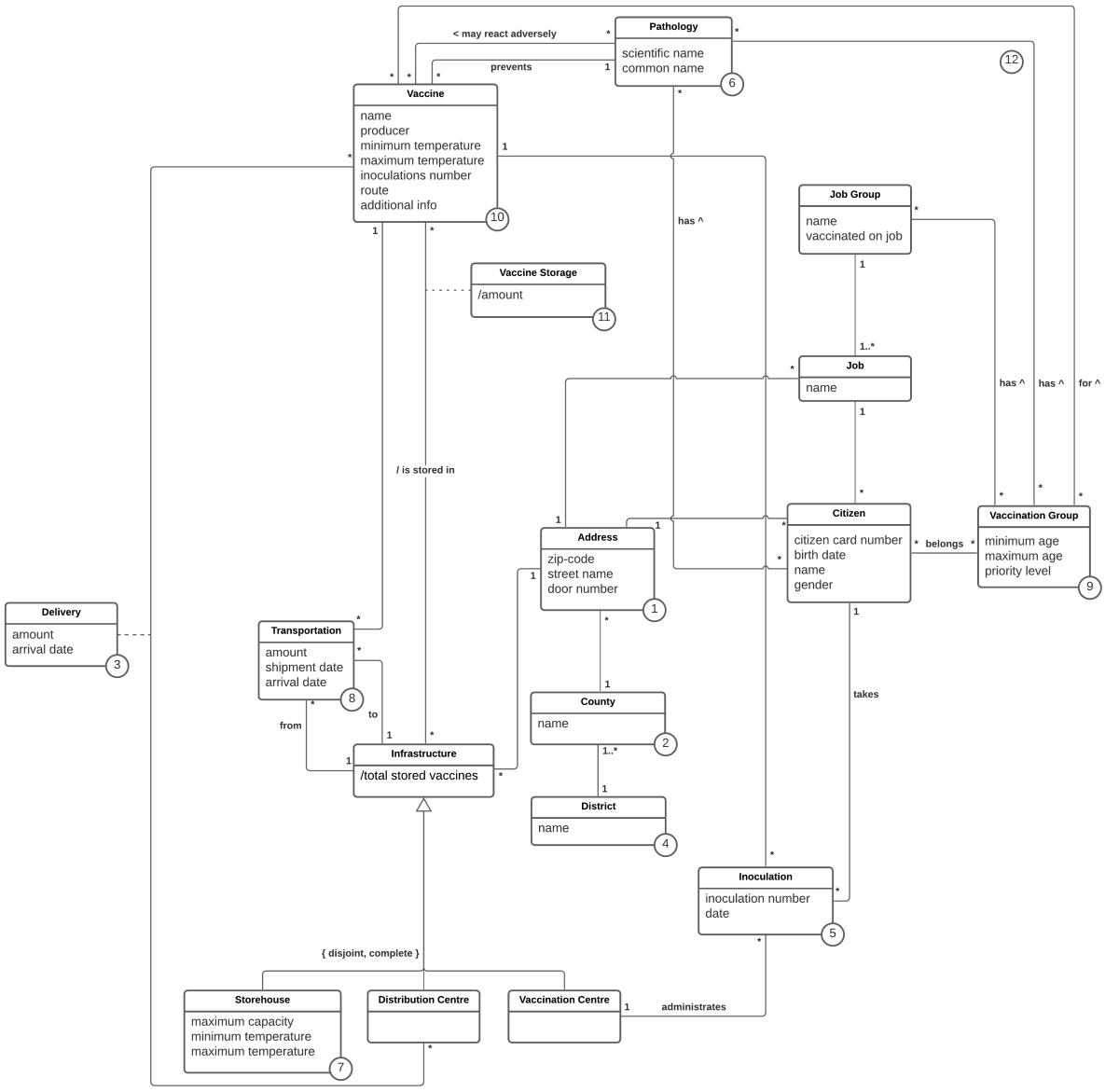
Name	Definition	Restrictions
Vaccine	Defines a vaccine by its name, producer, the number of inoculations it requires and the temperature range at which it must be reserved. The pathology it prevents, as well as the pathologies it may react adversely to, are also defined.	(10) a) inoculations num > 0 b) maximum temperature $\geq$ minimum temperature
Infrastructure	Defines an infrastructure by its address and the total (calculated) amount of vaccines it stores. It also keeps track of all of its vaccines' stock. It may receive or deliver transportations of vaccines.	None
Storehouse	It is a generalization of an Infrastructure. Represents a storehouse and, as such, has a given max capacity. It also defines at what temperature range vaccines can be stored.	(7) a) maximum temperature $\geq$ minimum temperature
Distribution Centre	It is a generalization of an Infrastructure. Represents a main centre of distribution and, as such, holds the first stop for all vaccines.	None
Vaccination Centre	It is a generalization of an Infrastructure. Defines the place where inoculations are taken.	None
Vaccine Storage	Defines the stock of a given vaccine in a given infrastructure.	(11) a) amount ≥ 0
Transportation	Defines the amount of vaccines that are being transported, as well as the shipment and arrival date, the vaccine being transported and the infrastructure from which they are being taken and to which they are to be delivered.	(8) a) amount > 0 b) Transportation can't be held to Distribution Centre

Delivery	Defines a shipment to a distribution centre of a certain vaccine. It holds the amount delivered and the date of arrival.	(3) a) amount > 0
Inoculation	Defines a given inoculation of a certain vaccine of a citizen in a certain vaccination centre. It is also described by its number (first, second (...) take of a given vaccine) and the date.	(4) a) $1 \leq$ inoculation number $\leq$ inoculations num of the associated Vaccine
Pathology	Defines a pathology, which is defined by its scientific name and common name. It may be prevented by or it may react adversely to a vaccine.	(6) a) scientific name must be unique
Vaccination Group	Defines a group by the minimum and maximum ages, as well as its priority level, represented by a number. The lower that number is, the higher the priority of that group. It is formed for a given vaccine and contains citizens with a certain job group and a certain set of pathologies.	(9) a) $0 \leq$ minimum age $\leq$ maximum age b) priority ≥ 0
Job Group	Defines a job group. A job group contains jobs of the same area (medical job group $\rightarrow$ nurse, doctor..). It is defined by its name and whether or not citizens that belong to it should be vaccinated in their job place (nursing homes, for example).	None
Job	Defines a job by its name.	None
Citizen	Defines a citizen by their citizenship card number, birth date, name and gender. His address is also defined, as well as his job and the group he belongs to (might not belong to any group).	None
Address	Defines the address of a given place with its zip-code, street name, door number and county.	(1) a) door number ≥ 1

County	Defines a county by its name.	(2) a) name must be unique for a specific District
District	Defines a district by its name.	(4) a) name must be unique

3. Revised Conceptual Modeling

3.1 UML Diagram



3.2 Extra Restrictions after Revision

Name	Definition	Restrictions
District	Association that defines the pathology a vaccination group is made to neutralize.	(12) a) the pathology must be vaccinable (there must exist an association between a vaccine and this pathology)

4. Relational Model Definition

In all the explicit relations, the primary key is defined with an underline.

district(id, name)

county(id, name, district_id)
district_id : FK(district)

zip_code(id, zip_code, county_id)
county_id : FK(county)

address(id, zip_code_id, street_name, door_number)
zip_code_id : FK(zip_code)

job_group(id, name, vaccinated_on_job)

job(id, name, group_id, address_id)
group_id : FK(job_group)
address_id : FK(address)

pathology(id, scientific_name, common_name)

vaccine(id, name, producer, minimum_temperature, maximum_temperature,
prevents_pathology_id, inoculations_number, route, additional_info)
prevents_pathology_id : FK(pathology)

vaccination_group(id, minimum_age, maximum_age)

pathology_reacts_adversely_to_vaccine(vaccine_id, pathology_id)
vaccine_id : FK(vaccine)
pathology_id : FK(pathology)

citizen(id, citizen_card_number, birth_date, name, minimum_temperature,
gender, job_id, address_id)
job_id : FK(job)
address_id : FK(address)

job_group_vaccination_group(job_group_id, vaccination_group_id)
job_group_id : FK(job_group)
vaccination_group_id : FK(vaccination_group)

pathology_vaccination_group(pathology_id, vaccination_group_id)
pathology_id : FK(pathology)
vaccination_group_id : FK(vaccination_group)

vaccination_group_vaccine(vaccination_group_id, vaccine_id)
vaccination_group_id : FK(vaccination_group)
vaccine_id : FK(vaccine)

citizen_has_pathology(citizen_id, pathology_id)
citizen_id : FK(citizen)
pathology_id : FK(pathology)

citizen_belongs_to_vaccination_group(citizen_id, vaccination_group_id)
citizen_id : FK(citizen)
vaccination_group_id : FK(vaccination_group)

infrastructure(id, address_id, total_stored_vaccines)
address_id : FK(address)

storehouse(infrastructure_id, maximum_capacity, minimum_temperature,
maximum_temperature)
infrastructure_id : FK(infrastructure)

distribution_centre(infrastructure_id)
infrastructure_id : FK(infrastructure)

vaccination_centre(infrastructure_id)
infrastructure_id : FK(infrastructure)

inoculation(id, inoculation, number, date, vaccination_centre_id, vaccine_id, citizen_id)
vaccination_centre_id : FK(vaccination_centre)
vaccine_id : FK(vaccine)
citizen_id : FK(citizen)

delivery(id, distribution_centre_id, vaccine_id, amount, arrival_date)
distribution_centre_id : FK(distribution_centre)
vaccine_id : FK(vaccine)

transportation(id, shipment_date, arrival_date, amount, from, to, vaccine_id)
from : FK(infrastructure)
to : FK(infrastructure)
vaccine_id : FK(vaccine)

vaccine_storage(vaccine_id, infrastructure_id, amount)
vaccine_id : FK(vaccine)
infrastructure_id : FK(infrastructure)

5. Functional Dependencies Analysis and Normal Forms

The following table presents all **non-trivial** functional dependencies.

Most of the relations that come from an association between two other relations have no other attributes than those derived directly from the associated tables, thus have no functional dependencies apart from the trivial ones. Consequently, these relations display no functional dependencies in this table and are indicated with *None*.

Since every attribute of every table corresponds to only one atomic type, one can conclude that all the relations follow the First Normal Form. Simultaneously, given that all non-prime-attributes never depend on a subset of a relation's key, the Second Normal Form is verified. On top of that, since every relation holds dependencies of prime attributes or dependencies on attributes that constitute a key, the Third Normal Form is also verified.

Given the lack of real-world unique attributes, some relations lay only on one functional dependency, where {id} → {everything else}, being therefore on BCNF, and indicated with *None* on its *Analysis*. With this, when this is not the case, the *Analysis* column provides an explanation of the keys, as well as the clarification of why it still holds the BCNF.

Relation	Functional Dependencies	Analysis
district	{id} → {name} {name} → {id}	The district's name is a natural key
county	{id} → {name, district_id} {name, district_id} → {id}	There can only exist one county per district
zip_code	{id} → {zip_code, county_id} {zip_code} → {id, county_id}	The zip-code is a natural key and the county can be extrapolated from it
address	{id} → {zip_code, street_name, door_number} {zip_code, street_name, door_number} → {id}	There can only exist one address with the same door number, in the same street, in the same zip code area

job_group	{id} → {name, vaccinated_on_job} {name} → {id, vaccinated_on_job}	The job group's name is a natural key
job	{id} → {name, group_id, address_id} {name, group_id, address_id} → {id}	There can only exist one job name per group in the same address
pathology	{id} → {scientific_name, common_name} {scientific_name} → {id, common_name}	The pathology's scientific name is a natural key
vaccine	{id} → {name, producer, inoculations_number, minimum_temperature, maximum_temperature} {name} → {id, producer, inoculations_number, minimum_temperature, maximum_temperature}	The vaccine's name is a natural key
vaccination_group	{id} → {minimum_age, maximum_age, priority}	None
pathology_reacts_adversely_to_vaccine	None	None
citizen	{id} → {citizen_card_number, birth_date, name, gender, job_id, address_id} {citizen_card_number} → {id, birth_date, name, gender, job_id, address_id}	The citizen's card number is a natural key
job_group_vaccination_group	None	None
pathology_vaccination_group	None	None
vaccination_group_vaccine	None	None
citizen_has_pathologoy	None	None
citizen_belongs_to_vaccination_group	None	None
infrastructure	{id} → {address_id, total_stored_vaccines}	None

storehouse	{infrastructure_id} → {maximum_capacity, minimum_temperature, maximum_temperature}	None
distribution_centre	None	None
vaccination_centre	None	None
inoculation	{id} → {inoculation_number, date, vaccination_centre_id, vaccine_id, citizen_id} {inoculation_number, date, vaccine_id, citizen_id} → {id, vaccination_centre_id}	A citizen can only take one vaccine shot in a day, and the vaccination centre can be extrapolated from it
delivery	{id} → {distribution_center_id, vaccine_id, amount, arrival_date}	None
transportation	{id} → {shipment_date, arrival_date, amount, from, to, vaccine}	None
vaccine_storage	{vaccine_id, infrastructure_id} → {amount}	Given an infrastructure and a vaccine, the amount of shots of the vaccine is unique and can be extrapolated from it

6. Restrictions

All restrictions that do not need the use of triggers were implemented and are listed bellow, according to each type of restriction.

6.1 Key Restrictions: Primary Key or Unique

Every table that does not come from an association has a parameter `id`. As such, all of them have a primary key constraint.

The association tables have a composite **primary key**, formed by the two `id`s of the tables that define the association. For example, pathology_reacts_adversely_to_vaccine has a composite primary key formed by vaccine_id and pathology_id.

There are several **unique** constraints. All of them are listed below.

- A district name must be unique;
- There can only be one unique county name for a given district;
- A zip code must be unique;
- An address, which is composed by a zip code, street name and door number, must be unique;
- A job-group name must be unique;
- A job, which is composed by its name, job-group and address, must be unique;
- A pathology's scientific name must be unique;
- A vaccine's name must be unique;
- A citizen's card number must be unique;
- An inoculation, which is defined by its number, date, vaccine and citizen, must be unique;

6.2 Referential Integrity Restrictions: Foreign Keys

Whenever there exists a parameter on a given table that references another table, a **foreign key** constraint is used. For example, in pathology_reacts_adversely_to_vaccine, both vaccine_id and pathology_id are foreign keys. Therefore, a constraint was set for both parameters. The aforementioned example can be applied to all foreign keys that were explicitly described in chapter 4.

6.3 Context Restrictions: Check

There are several check constraints. All of them are listed below.

- An address either has no door number or it must be greater than or equal to 1;
- A vaccine must require at least one inoculation;
- A vaccine's minimum temperature must be lower than or equal to its maximum temperature;
- A vaccination-group's age range must respect all the following conditions:
  1. the minimum age must be greater than or equal to 0;
  2. either the minimum age is lower than or equal to the maximum age or there is no maximum age;
- A vaccination's group priority level must be greater than or equal to 0;
- The number of an inoculation must be greater or equal to 1;
- The number of vaccines stored in an infrastructure must be greater than or equal to 0.
- A storehouse either has no defined maximum capacity or it is greater than 0.
- A storehouse's temperature range must respect one of the following conditions:
  1. there's no minimum temperature;
  2. there's no maximum temperature;
  3. the minimum temperature must be lower than or equal to the maximum temperature;
- A delivery must deliver at least one vaccine.
- A transportation must carry at least one vaccine.
- The dates of a transportation must respect one of the following conditions:
  1. there's no shipment date;
  2. there's no arrival date;
  3. the arrival date is greater than or equal to the shipment date;
- An infrastructure either has no information about the amount of vaccines it stores or it stores a number greater than or equal to 0.

6.4 Mandatory Parameter Restrictions: Not Null

Whenever a parameter is critical to a table and must exist, a **not null** constraint is set (except when it already is implied, *e.g.*, all primary keys). All of the parameters that are listed below cannot be null.

Table	**Parameter**
district	name
county	name, district_id
zip_code	zip_code, county_id
address	street_name
job_group	name, vaccinated_on_job
job	name
pathology	scientific_name
vaccine	name, producer, minimum_temperature, maximum_temperature, prevents_pathology_id, inoculations_number
vaccination_group	minimum_age, priority_level
citizen	citizen_card_number, name, birth_date, gender
inoculation	inoculation_number, vaccine_id, citizen_id
infrastructure	address_id
delivery	distribution_centre_id, vaccine_id, amount
transportation	amount, to, vaccine_id