UCLouvain-EPL

Database

Project 2: Healthcare Application

Teacher:
Siegfried Nijssen
Course assistants:
A. Mattenet,
Dam, G. Maudoux

Authors:
Nima Farnoodian - 68372000
nima.farnoodian@student.uclouvain.be

Andia Fathi - 56301900
andia.fathi@student.uclouvain.be

Group Q

1 Health Application

After peer-reviewing, we applied several modifications to our original diagrams to represent a better conceptual design for our database. In the following sections, we represent our updated diagrams (ER-chen and UML) and the required code for creating the database along with its physical diagram.

1.1 Updated Diagrams

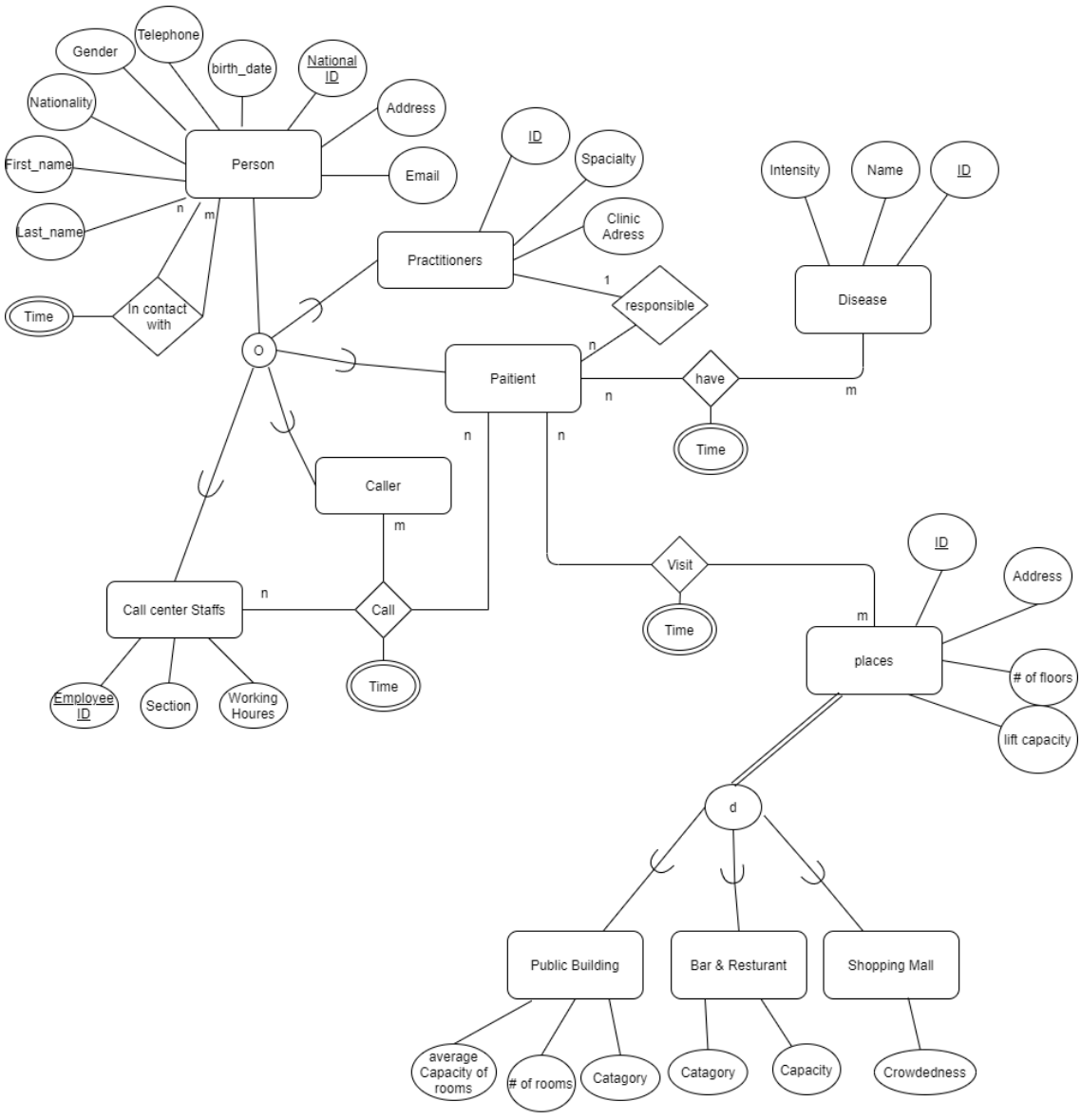

Figure 1: ER-chen diagram (updated after peer-reviewing)

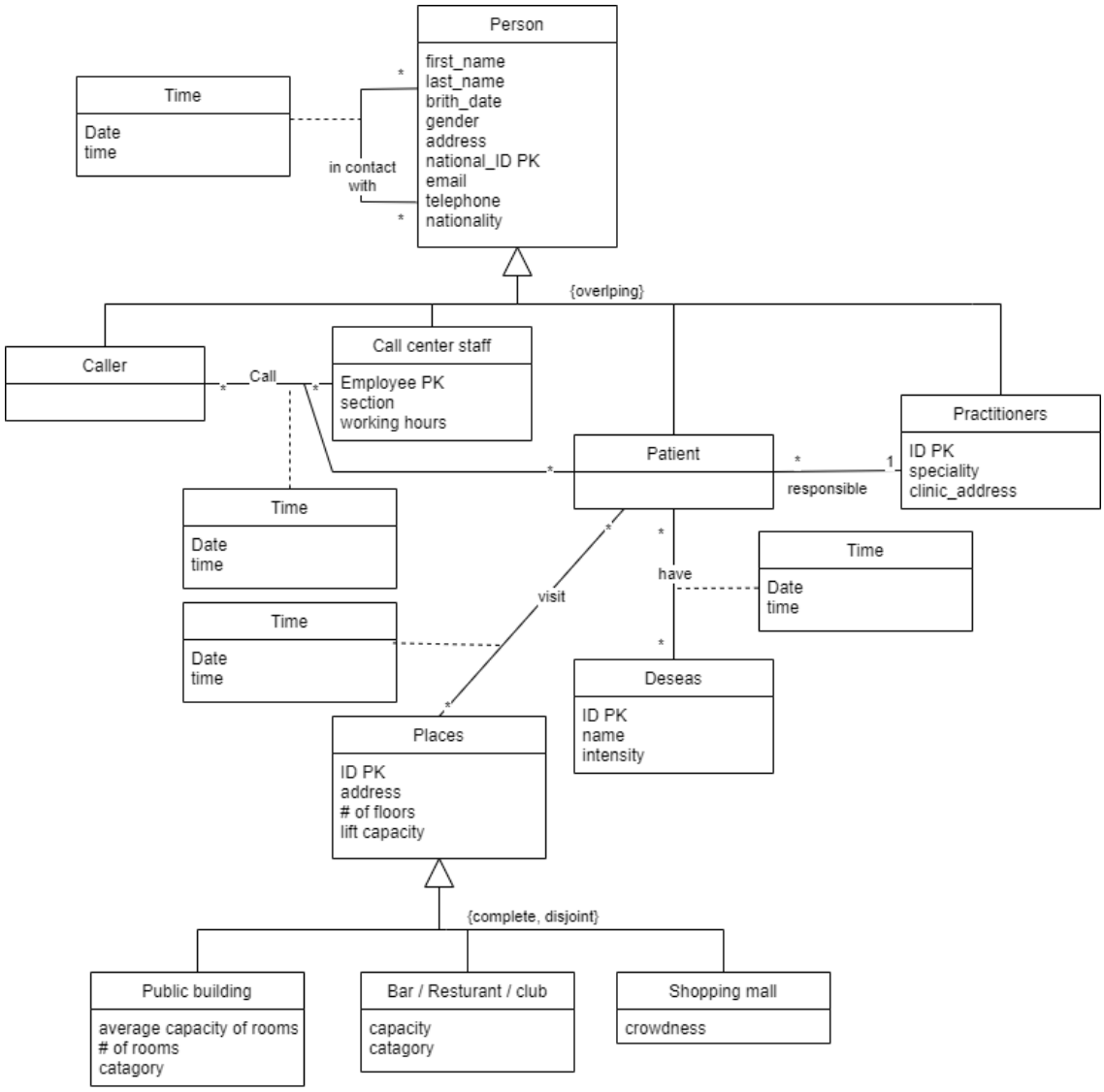

Figure 2: UML diagram (updated after peer-reviewing)

1.2 Physical Design

The below diagram was obtained by dbdiagram.io.

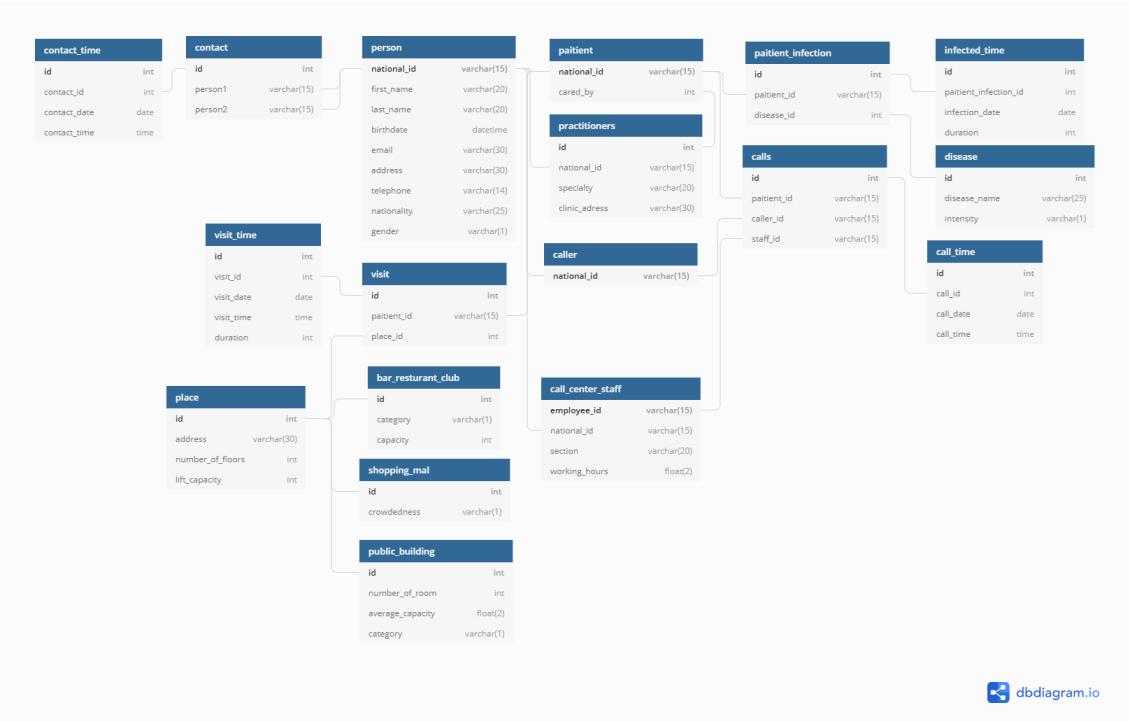

Figure 3: Physical design

1.2.1 SQL Code

In what follows, the SQL scripts required to create the related tables are provided in the order of creation. If the order changes, the errors might appear.

```
create table person (
national_id varchar (15) primary key,
first_name varchar (20) not null,
last_name varchar (20) not null,
birthdate datetime not null,
email varchar (30),
address varchar (30) not null,
telephone varchar(14) not null,
nationality varchar(25),
gender varchar(1) not null check (gender in ('M','F','O'))
);
```

Note: M means Male, F means Female, and O means other.

```
create table practitioners (
id int primary key not null auto_increment,
national_id varchar (15) not null,
specialty varchar (20) not null,
clinic_adress varchar(30) not null,
foreign key (national_id) references person(national_id) on delete cascade
);
```

```
create table call_center_staff (
employee_id varchar(15) primary key not null,
national_id varchar (15) not null,
section varchar (20) not null,
working_hours float(2) not null,
foreign key (national_id) references person(national_id) on delete cascade
);

create table paitient (
national_id varchar (15) primary key REFERENCES person( national_id ),
cared_by int,
foreign key (cared_by) references practitioners(id) on delete cascade
);


create table caller (
national_id varchar (15) primary key REFERENCES person( national_id )
);

create table place (
id int primary key not null auto_increment,
address varchar (30) not null,
number_of_floors int,
lift_capacity int
);

create table shopping_mal (
id int primary key REFERENCES place( id ),
crowdedness varchar(1) not null  check (crowdedness in ('H','M','L'))
);
```

Note: H means high, M means Medium, and L means low.

```
create table bar_resturant (
id int primary key REFERENCES place( id ),
category varchar(1) not null  check (category in ('B','R')),
capacity int
);
```

Note: B means bar, R means Restaurant.

```
create table public_building (
id int primary key REFERENCES place( id ),
number_of_room int not null,
average_capacity float(2) ,
category varchar(1) not null  check (category in ('H','A','O'))
);
```

Note: H means hotel, A means administration buildings, and O means other.

```
create table contact (
id int primary key not null auto_increment,
person1 varchar (15) not null,
person2 varchar (15) not null,
foreign key (person1) references person(national_id) on delete cascade,
foreign key (person2) references person(national_id) on delete cascade
);
```

```
create table contact_time (
id int primary key not null auto_increment,
contact_id int not null,
contact_date date not null,
contact_time time,
foreign key (contact_id) references contact(id) on delete cascade
);

create table calls (
id int primary key not null auto_increment,
paitient_id varchar (15) not null,
caller_id varchar (15) not null,
staff_id varchar(15) not null,
foreign key (paitient_id) references paitient(national_id) on delete cascade,
foreign key (caller_id) references caller(national_id) on delete cascade,
foreign key (staff_id) references call_center_staff(employee_id) on delete cascade
);

create table call_time (
id int primary key not null auto_increment,
call_id int not null,
call_date date not null,
call_time time not null,
foreign key (call_id) references calls(id) on delete cascade
);

create table disease(
id int primary key not null auto_increment,
disease_name varchar(25) not null,
intensity varchar(1) not null  check (intensity in ('H','M','L'))
);
```

Note: H means high, M means Medium, and L means low.

```
create table paitient_infection(
id int primary key not null auto_increment,
paitient_id varchar (15) not null,
disease_id int not null,
foreign key (paitient_id) references paitient(national_id) on delete cascade,
foreign key (disease_id) references disease(id) on delete cascade
);

create table infected_time(
id int primary key not null auto_increment,
paitient_infection_id int not null,
infection_date date not null,
duration int,
foreign key (paitient_infection_id) references paitient_infection(id) on delete cascade
)

create table visit(
id int primary key not null auto_increment,
paitient_id varchar (15) not null,
place_id int not null,
foreign key (paitient_id) references paitient(national_id) on delete cascade,
foreign key (place_id) references place(id) on delete cascade
);
```

```
create table visit_time (
id int primary key not null auto_increment,
visit_id int not null,
visit_date date not null,
visit_time time not null,
duration int,
foreign key (visit_id) references visit(id) on delete cascade
);
```

2 Soccer

In the original database, the match table includes redundant information about players of the match. Indeed, each player who played in the match is mentioned using its Foreign key. In total, considering the fact that in each match around 22 players play, 22 attributes are redundantly stored for each match. To this end, a third (helper) table that holds the players of the match should be added to the database and all foreign keys for the players in the match table should be removed.

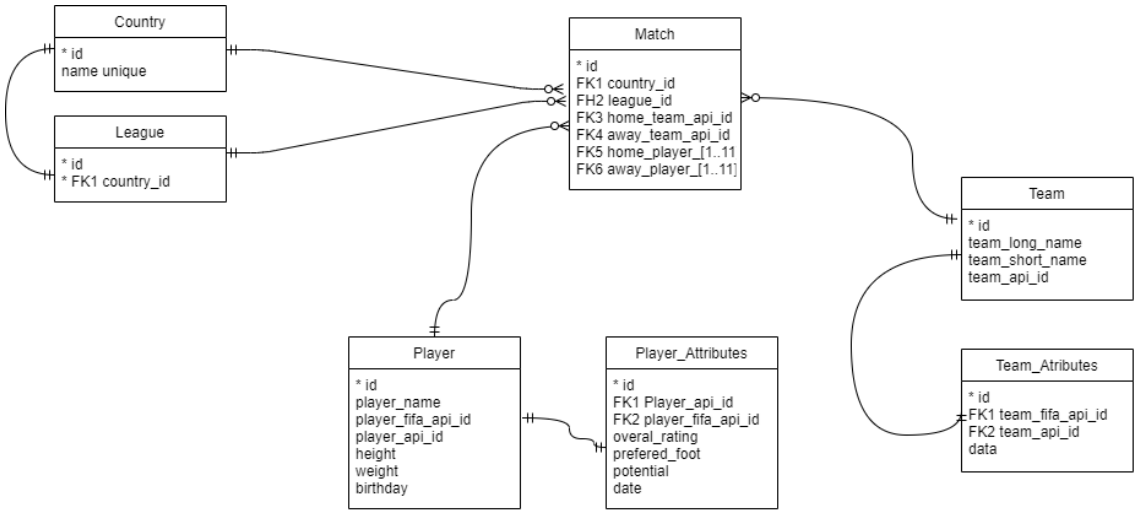

Figure 4: Physical Diagram of Soccer

3 Train

3.1 Entity-Relationship Diagram

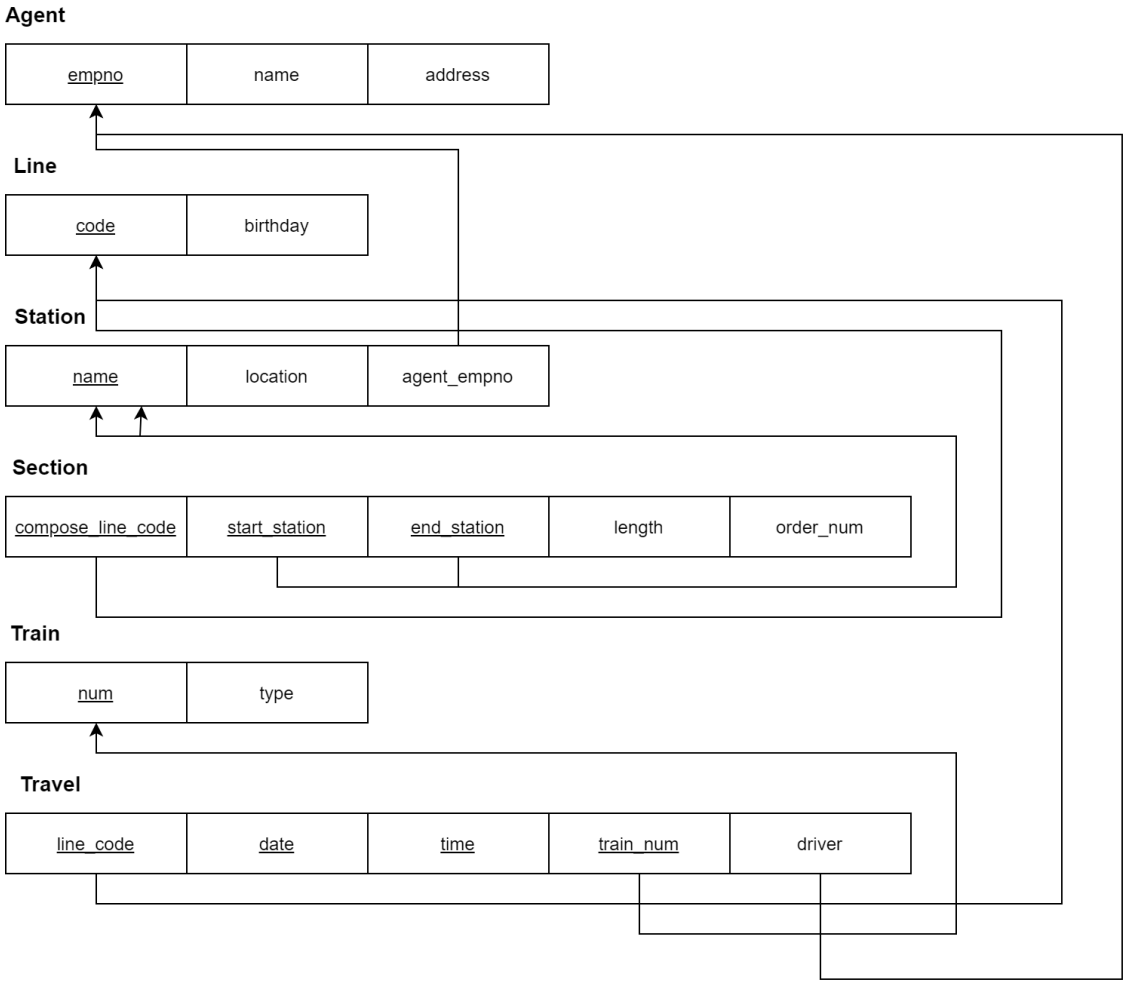

Figure 5: Entity-Relationship Diagram for Train Database

3.2 SQL Scripts

In order to keep consistency, the following tables should be created in the order of presentation.

```
create table Agent (
name varchar(20) not null,
empno int primary key auto_increment UNIQUE not null,
address varchar (30) not null
) ;

create table Line (
code int primary key auto_increment UNIQUE not null,
birthDate DATE not null) ;
```

```
create table Station (
name varchar(25)  primary key  UNIQUE not null,
location varchar(25) not null,
agent_empno varchar(25) REFERENCES Agent ( empno ) ) ;

create table Section (
compose_line_code int not null REFERENCES Line ( code ),
start_station varchar(25) not null REFERENCES Station (name),
end_station  varchar(25) not null REFERENCES Station (name ) ,
length float(2) not null,
order_num int,
primary key (compose_line_code , start_station , end_station )
) ;

create table Train (
num int primary key  auto_increment not null UNIQUE,
type varchar(15)) ;

create table Travel (
line_code int not null REFERENCES Line ( code ) ,
date DATE not null,
time TIME not null,
train_num int not null REFERENCES Train (num) ,
driver int not null  REFERENCES Agent (empno) ,
primary key ( line_code , date , time, train_num ) ) ;
;
```