# Inf553: 1st SQL assignment

Ioana Manolescu, Pierre Bourhis, Benoît Groz

September 20, 2019

## Database description

We consider a database of three tables created using the commands:

```
create table person(pid int primary key, firstname varchar, lastname varchar);
create table office(oid int primary key, positiontitle varchar, details varchar);
create table heldOffice(pid int foreign key references person(pid),
                        oid int foreign key references office(oid),
                        ystart int, yend int);
```

We assume that in the database, no two people hold the same position in the same year and no person holds two positions at the same time. (We omit the syntax of the constraints used to enforce this.)

**Write (in plain text) in the Moodle assignment** the SQL queries that return, on any instance of this schema that satisfies the above constraints:

1. Find the last name of those whose first name is Nicolas.
2. Find the first name and the last name of those who were or are Prime Minister.
3. Find the first name and last name of those who held a position in 2018.
4. For two distinct people A and B, we say B reported to A, if:
   - A was president and B held another office in year Y, *or*
   - A was prime minister in year Y, and B had an office in year Y which was neither president nor prime minister.

   Find all the pairs (name of A, name of B).
5. Find the first name and last name of people who were never Ministers of Defense.
6. We say a group of people *worked together* if the start and end year of their offices are identical. Find the start year and end year of the largest group of people who worked together.

**Sample instance** To help you think about the queries, we provide below the insert commands that would create *one* sample instance of the above schema. Recall however that your queries must be correct against *any* such instance.

```
insert into person values(1, 'Emmanuel', 'Macron');
insert into person values(2, 'Edouard', 'Philippe');
insert into person values(3, 'Elisabeth', 'Borne');
insert into person values(4, 'Florence', 'Parly');
insert into person values(5, 'Jean-Michel', 'Blanquer');
insert into person values(6, 'Benoit', 'Hamon');
insert into person values(7, 'Francois', 'Hollande');
insert into person values(8, 'Nicolas', 'Hulot');
insert into person values(9, 'Nicolas', 'Sarkozy');
insert into person values(10, 'Manuel', 'Valls');
```

```
insert into office values(1, 'PR', 'Président de la République');
insert into office values(2, 'PM', 'Premier Ministre');
insert into office values(3, 'MinEco', 'Ministre de l''Ecologie');
insert into office values(4, 'MinDef', 'Ministre de la Défense');
insert into office values(5, 'MinEdu', 'Ministre de l''Education');

insert into heldOffice values(1, 1, 2017, 2021);
insert into heldOffice values(2, 2, 2017, 2021);
insert into heldOffice values(3, 3, 2019, 2021);
insert into heldOffice values(4, 4, 2017, 2021);
insert into heldOffice values(5, 5, 2017, 2021);
insert into heldOffice values(6, 5, 2014, 2014);
insert into heldOffice values(7, 1, 2012, 2016);
insert into heldOffice values(8, 3, 2017, 2018);
insert into heldOffice values(9, 1, 2007, 2011);
insert into heldOffice values(10, 2, 2012, 2014);
```