Practice 1

Consider an ODL schema modeling information about a university. It includes the following class declarations:

```
class Person (extent persons key ssn) {
      attribute struct Pname {
              string fname,
              string lname } name;
      attribute set<Pname> prevnames;
      attribute string ssn;
      attribute Date birthdate;
}


class Faculty extends Person (extent faculty) {
      attribute string rank;
      attribute string phone;
      attribute string office;
      relationship set<GradStu> advises inverse GradStu::advisor;
}


class Dept (extent dept key dname) {
      attribute string dname;
      attribute struct Address {
              string street;
              string city;
              string state;
              int    zip;
              int     buildingCode; } address;
      relationship set<Faculty> has_faculty;
}
```

I.

Propose an SQL schema to model this data relationally. No need to provide SQL table declarations. Notation of the form

$$RelationName\ (type_1\ attrib_1, \ldots, type_n\ attrib_n)$$

suffices, complemented by explicit key and foreign key declarations.
List features of the ODL schema that cannot be captured in this way (if any).

II.
Express the following queries in OQL:

1.
Find the department (return the object itself) of Prof. "X".

2.
Find the department of the professor formerly known as "Charles Xavier".

3.
Now assume that the definition of class Faculty is enriched with the declaration

relationship Dept works_in **inverse** Dept::has_faculty;

and redo query 1.

4.
Under the same assumption as 3., return for each professor their contact info.
The query result should be of type

set<struct {string lname,
string ssn,
struct {int buildingCode, string office, string phone} contact}>

5.
Dropping the assumption of point 3., assume instead that Faculty is enriched with

relationship set<Dept> works_in **inverse** Dept::has_faculty;

Find the professors (return the objects themselves) affiliated with *all* departments.

III.

Express the queries II.1, II.2 and II.5 in SQL, on the schema of point I.
Instead of returning objects, return the key of the corresponding entities.