I.
Here is a possible relational schema capturing the given ODL schema. This is by no means the only solution.

Person (string ssn,
    string name_fname,
    string name_lname,
    Date birthdate)

Prevnames (    string ssn, string Pname_fname, string Pname_lname )
foreign key Prevnames.ssn references Person

Faculty (string ssn,
    string rank,
    string phone,
    string office)
foreign key Faculty.ssn references Person

Advises (    string fssn, string gssn )
foreign key Advises.fssn references Faculty
foreign key Advises.gssn references GradStu

Dept (    string dname,
    string address_street,
    string address_city,
    string address_state,
    int address_zip,
    int address_buildingCode)

Has_faculty (    string ssn, string dname )
foreign key has_faculty.ssn references Faculty
foreign key has_faculty.dname references Dept

GradStu (string ssn, string major, real gpa)
foreign key GradStu.ssn references Person

create view Advisor (gssn, fssn) as
    select gssn, fssn from Advises

    // note, instead of defining a new table to capture the Advisor relationship,
    // we define a view, which effectively gives another name to the Advises
    // relationship. In this way we do not need to worry about enforcing Advises and
    // Advisor to be inverses of each other. They are one and the same table, really.
II. Again, here is a possible solution (infinite variations exist).

1. 
   ```
   select d 
   from f in faculty, d in dept 
   where f.name.lname = 'X' and f in d.has_faculty
   ```

2. 
   ```
   select d 
   from f in faculty, pn in f.prevnames, d in dept 
   where pn.fname = 'Charles' and pn.lname = 'Xavier' 
     and f in d.has_faculty
   ```

3. 
   ```
   select f.works_in 
   from f in faculty 
   where f.name.lname = 'X'
   ```

4. 
   ```
   select struct ( lname: f.name.lname, 
                    ssn: f.ssn, 
                    contact: struct ( buildingCode: f.works_in.address.buildingCode, 
                                     office: f.office, 
                                     phone: f.phone ) 
                    ) 
   from f in faculty
   ```

5. 
   ```
   select f 
   from f in faculty 
   where for all d in dept: d in f.works_in
III.

1. 
   ```
   select h.dname 
   from Person p, Has_faculty h 
   where p.ssn = h.ssn and p.name_lname = 'X'
   ```

2. 
   ```
   select h.dname 
   from Person p, Prevnames n, Has_faculty h 
   where p.ssn = h.ssn and n.ssn = p.ssn 
   and n.Pname_fname = 'Charles' and n.Pname_lname = 'Xavier'
   ```

5. 
   ```
   select ssn 
   from faculty 
   where ssn not in 
     ( 
       select f.ssn 
       from Faculty f, Dept d 
       where not exists ( 
         select * 
         from Has_faculty h 
         where h.ssn = f.ssn and 
         h.dname = d.dname 
       ) 
     )
   ```