

HW16

22.2

a.

```
create table Emp
(
    ename char(20),
    ChildrenSet Children multiset,
    SkillSet Skills multiset,
)

create type Children as
(
    name varchar(20),
    birtyday date
)
not final;

create type Skills as
(
    type varchar(20),
    ExamSet Exams set
)
not final;

create type Exams as
(
    year year,
    city varchar(10)
)
not final;
```

b.

```

1.
select ename
from emp as e, e.ChildrenSet as c
where c.birthday.year >= 2000

2.
select e.ename
from Emp as e, e.SkillSet as s, s.ExamSet as x
where s.type = 'typing' and x.city = 'Dayton'

3.
select distinct s.type
from emp as e, e.SkillSet as s

```

22.7

Answer:

To put the schema into first normal form, we flatten all the attributes into a single relation schema.

Employee-details = (ename, cname, bday, bmonth, byear, stype, xyear, xcity)

We rename the attributes for the sake of clarity. cname is Children.name, and bday, bmonth, byear are the Birthday attributes. stype is Skills.type, and xyear and xcity are the Exams attributes. The FDs and multivalued dependencies we assume are:

```

ename, cname -> bday, bmonth, byear
ename ->-> cname, bday, bmonth, byear
ename, stype ->-> xyear, xcity

```

The FD captures the fact that a child has a unique birthday, under the assumption that one employee cannot have two children of the same name. The MVDs capture the fact there is no relationship between the children of an employee and his or her skills-information.

The redesigned schema in fourth normal is:

```

Employee = (ename)
Child = (ename, cname, bday, bmonth, byear)
Skill = (ename, stype, xyear, xcity)

```

ename will be the primary key of Employee, and (ename, cname) will be the primary key of Child. The ename attribute is a foreign key in Child and in Skill, referring to the Employee relation.