4.7

employee (employee_name, street, city)
works (employee_name, company_name, salary)
company (company_name, city)
manages (employee_name, manager_name)

```
create table employee
  (employee_name  char(20),
   street         char(30),
   city           char(30),
   primary key (employee_name))

create table works
   (employee_name  char(20),
    company_name   char(15),
    salary   integer,
    primary key (employee_name),
```

```
    foreign key (employee_name) references employee,
    foreign key (company_name) references company)

create table company
  (company_name  char(15),
   city          char(30),
   primary key   (company_name))
```

```
create table managers
    ( employee_name  char(20),
      manager_name   char(20),
      primary key    (employee_name),
      foreign key    (employee_name) references employee,
      foreign key    (manager_name) references employee)
```

4.9

The tuples of all employees of the manager, at all levels, get deleted as well! This happens in a series of steps. The initial deletion will trigger deletion of all the tuples corresponding to direct employees of the manager. These deletions will in turn cause deletions of second level employee tuples, and so on, till all direct and indirect employee tuples are deleted.

4.16

```
salaried_worker (name, office, phone, salary)
hourly_worker (name, hourly_wage)
address (name, street, city)
```

a.

foreign key (name) references salaried_worker or hourly_worker

b. To enforce this constriant, whenever a tuple is inserted into the address relation, a loop-up on the name value must be made on the salaried_worker relation and (if that lookup failed) on the hourly_worker relation (or vice-versa).

5.8

```
create trigger check_delete_trigger after delete on account
referencing old row as orow
for each row
delete from depositor
Where depositor.customer_name not in
    (select customer_name from depositor
     where account_number <> orow.account_number)
end
```

5.21

$r.B \rightarrow s.A$

We define triggers for each relation whose primary key B of r references the primary key A of S. Plus, the primary key is referred to by the foreign key of some other relation. The trigger would be activated whenever a tuple is deleted from the reffered to relation.

The action performed by the trigger would be to visit all the refering relations, and delete all the tuples in them whose foreign-key attribute value is the same as the primary key attribute value of the deleted tuple in the refered-to relation. These set of triggers will take care of the on delete cascade operation.