1. **Solution:**

(a)
```
create table Offices (
    officeId    integer,
    building    varchar(10) not null,
    level       integer not null,
    roomNumber  integer not null,
    area        integer,
    primary key (officeId),
    unique (building, level, roomNumber)
);

create table Employees (
    empId       integer,
    name        varchar(30) not null,
    officeId    integer not null,
    managerId   integer,
    primary key (empId),
    foreign key (officeId) references Offices (officeId),
    foreign key (managerId) references Employees (empId)
);
```

(b)
```
update Employees
set officeId =
    (select officeId from Offices
    where building = 'Tower1'
    and level = 5
    and roomNumber = 11)
where officeId = 123;
```

(c)
```
create table Employees (
    empId       integer,
    name        varchar(30) not null,
    officeId    integer,
    managerId   integer,
    primary key (empId, officeId),
    foreign key (officeId) references Offices (officeId)
);
```

Note that the above schema is unable to enforce the following constraints on Employees relation: (1) each non-null value for managerId must refer to some empId value; and (2) for each employee with more than one assigned office, the employee must have a unique name and a unique manager if he/she is managed by someone.

An alternative design that does not have these limitations is to keep the original Employees schema with `officeId` used to record an employee's first office, and to use an additional relation **EmployeeOffices** (empId, officeId) to store information about additional offices assigned to employees.

2. **Solution:**

(a)
```
select pizza
from Likes
where cname = 'Alice'
and pizza not in
    (select pizza
    from Likes
    where cname = 'Bob');
```

(b)
```
select distinct C.cname, S.pizza
from Customers C natural join Restaurant R natural join
Sells S;
```

(c)
```
select distinct C.cname, S.pizza
from Customers C natural left join (Restaurant R natural join
Sells S);
```

(d)
```
select distinct pizza
from Sells S1
where not exists (
    select 1
    from Sells S2, Sells S3, Restaurants R2, Restaurants R3
    where S2.pizza = S1.pizza
    and S3.pizza = S1.pizza
    and S2.rname <> S3.rname
    and R2.rname = S2.rname
    and R3.rname = S3.rname
    and R2.area = R3.area
);
```

(e)
```
select distinct S1.rname, S2.rname
from Sells S1, Sells S2
where (select max(price) from Sells where rname = S1.rname)
> (select max(price) from Sells where rname = S2.rname);
```

(f)
```
select pizza, rname
from Sells
where price >= all (select price from sells);
```