Landform Schema

```
create table Landform (
  id int primary key,
  name varchar(200) not null
);
create table Hill (
  id int primary key,
  foreign key(id) references Landform(id)
    on update cascade on delete cascade,
  height double not null,
  summitArea double,
  partOf int,
  foreign key(partOf) references Hill(id)
    on update cascade on delete cascade
);
create table Valley (
  id int primary key,
  foreign key(id) references Landform(id)
    on update cascade on delete cascade,
  area double not null
);
```

Landform Schema

```
create table Border (
  borders int not null,
  foreign key(borders) references Valley(id)
    on update cascade on delete cascade,
  isBorderedBy int not null,
  foreign key(isBorderedBy) references Hill(id)
    on update cascade on delete cascade,
  primary key(borders, isBorderedBy)
);
```

Valley Query Problem

List the valleys by name that are bordered by every hill that is part of a hill whose summit area exceeds 600.

Query Rewrites

List the valleys by name that are bordered by every hill that is part of a hill whose summit area exceeds 600.

First rewrite: List the valleys v by name such that for every hill h that is part of a hill whose summit area exceeds 600, the valley v is bordered by h.

Second rewrite: List the valleys v by name such that there does not exist a hill h that is part of a hill whose summit area exceeds 600, such that the valley v is not bordered by h.

Query Rewrites Alternative Approach

List the valleys by name that are bordered by every hill that is part of a hill whose summit area exceeds 600.

First rewrite: List the valleys v by name such that for every hill h **IF** h is part of a hill whose summit area exceeds 600 **THEN** the valley v is bordered by h.

Second rewrite: List the valleys v by name such that for every hill h **NOT** h is part of a hill whose summit area exceeds 600 **OR** the valley v is bordered by h.

Third rewrite: List the valleys v by name such that there does not exist a hill h such that **NOT** (**NOT** h is part of a hill whose summit area exceeds 600 **OR** the valley v is bordered by h).

Query Rewrites Alternative Approach

List the valleys by name that are bordered by every hill that is part of a hill whose summit area exceeds 600.

Fourth rewrite: List the valleys v by name such that there does not exist a hill h such that (h is part of a hill whose summit area exceeds 600) **AND NOT** (the valley v is bordered by h).

Fifth rewrite: List the valleys v by name such that there does not exist a hill h that is part of a hill whose summit area exceeds 600, such that the valley v is not bordered by h.

Query Solution

List the valleys v by name such that there does not exist a hill h that is part of a hill whose summit area exceeds 600, such that the valley v is not bordered by h.

```
select l.name
  from Valley v, Landform l
 where v.id = l.id
```

Query Solution

List the valleys v by name **such that there does not exist a hill h that is part of a hill whose summit area exceeds 600**, such that the valley v is not bordered by h.

```
select l.name
  from Valley v, Landform l
 where v.id = l.id
   and not exists(
         select *
           from Hill h, Hill k
          where h.partOf = k.id
            and k.summitArea > 600
```

Query Solution

List the valleys v by name such that there does not exist a hill h that is part of a hill whose summit area exceeds 600, **such that the valley v is not bordered by h**

```
select l.name
  from Valley v, Landform l
 where v.id = l.id
   and not exists(
         select *
           from Hill h, Hill k
          where h.partOf = k.id
            and k.summitArea > 600
            and not exists(
                  select *
                    from Border b
                   where b.borders = v.id
                     and b.isBorderedBy = h.id
                )
        )
```

Landform Schema

```
create table Landform (
  id int primary key,
  name varchar(200) not null
);
create table Hill (
  id int primary key,
  foreign key(id) references Landform(id)
    on update cascade on delete cascade,
  height double not null,
  summitArea double,
  partOf int,
  foreign key(partOf) references Hill(id)
    on update cascade on delete cascade
);
create table Valley (
  id int primary key,
  foreign key(id) references Landform(id)
    on update cascade on delete cascade,
  area double not null
);
```

The JOINED strategy is used for subclasses.

Relational Algebra Problem 1

Express the query “List the valleys by name whose area exceeds 600” using the relational algebra

```
select l.name
   from Valley v, Landform l
 where v.id = l.id
   and v.area > 600
```

Relational Algebra Solution

$$\pi_{\text{l.name}} (\sigma_{\text{v.area} > 600} (\text{Valley v} \bowtie_{\text{v.id=l.id}} \text{Landform l}))$$

Relational Algebra Problem 2

Express the query “List the hills by name that are part of a hill whose summit area exceeds 600” using the relational algebra.

```
select l.name
   from Hill h, Landform l, Hill k
 where h.id = l.id
   and h.partOf = k.id
   and k.summitArea > 600
```

Relational Algebra Solution

$$\pi_{\text{l.name}}\ \sigma_{\text{k.summitArea} > 600}\ ((\text{Hill h} \bowtie_{\text{h.id=l.id}} \text{Landform l}) \bowtie_{\text{h.partOf=k.id}} \text{Hill k})$$