Assertion Problem 3

```
create table Company(
  id int primary key,
  name varchar(500) not null,
  product varchar(500)
);
create table Person(
  id int primary key,
  name varchar(200) not null,
  worksFor int,
  foreign key (worksFor) references Company(id)
    on update cascade on delete no action
);
```

It is required that if a company has an employee, then the company has a product.

Assertion Solution Part 1

It is required that if a company has an employee, then the company has a product.

First rewrite: For every company c, if there exists an employee that works for c, then there exists a product of c

Second rewrite: For every company c, if there exists a person p who works for c, then the product field of c is not null

Assertion Solution Part 2

It is required that if a company has an employee, then the company has a product.

Use the fact that “if A then B” is logically equivalent to “NOT(A) OR B”

Third rewrite: For every company c, EITHER there does NOT exist a person p who works for c OR the product field of c is not null

Assertion Solution Part 3

It is required that if a company has an employee, then the company has a product.

Convert “for every” to “not exists ... not”

Fourth rewrite: There does not exist a company c such that NOT(EITHER there does NOT exist a person p who works for c OR the product field of c is not null)

Assertion Solution Part 4

It is required that if a company has an employee, then the company has a product.

Use De Morgan's Law

Fifth rewrite: There does not exist a company c such that there exists a person p who works for c AND the product field of c is null

Assertion Solution Part 5

There does not exist a company c such that there exists a person p who works for c AND the product field of c is null

```
create assertion CompanyHasAProduct check
  not exists (
    select *
      from Company c
     where exists (
             select *
               from Person p
              where p.worksFor = c.id
           )
       and c.product is null
  );
```

Assertion Solution Part 5

The existence of a person who works for a company c can also be specified by performing an inner join like this:

```
create assertion CompanyHasAProduct check
  not exists (
    select *
      from Company c, Person p
     where p.worksFor = c.id
       and c.product is null
  );
```

Landform Security Solution

Give the 'geographer' role permission to update the borders of a valley and to delegate this permission.

Changing a border means that one is either adding a new hill among the hills bordering a valley or one is removing an existing hill from the hills bordering a valley. One could also take the point of view of a hill which borders a set of valleys. Either way, one will need read access to Hill and Valley.

```
grant insert, update on Border to 'geographer'
  with grant option;
grant select on Hill to 'geographer'
  with grant option;
grant select on Valley to 'geographer'
  with grant option;
```