1.

select e1.\*

from E1 e1

where exists (select 1 from C)

UNION

select e2.\*

from E2 e2

where exists (select 1 from C);

=>

select e1.\*

from E1 e1, C c

UNION

select q.\* from

(select e2.\*

from E2 e2

EXCEPT

select e2.\*

from E2 e2, C c) q

RA:

2.

select true = all (select p1.y = p2.y

from F p1, F p2

where p1.x = p2.x) as isFunction ;

=>

Let be the query (if then we return True else False)

select 1 from F p1, F p2

where p1.x = p2.x and p1.y <> p2.y;

Converting Q to RA:

3.

For IN predicates:

select L1(r)

from R r, ( select s.B, r.\*

from S s, R r

where C2(s,r)

[union | intersect | except]

select t.C, r.\*

from T t, R r

where C3(t,r)

) q

where C1(r)

and r.A = q.B

RA for all IN predicates:

For NOT IN predicates:

select L1(r)

from R r

where C1(r)

EXCEPT

select L1(r)

from R r ,( select s.B, r.\*

from S s, R r

where C2(s,r)

[union | intersect | except]

select t.C, r.\*

from T t, R r

where C3(t,r)

) q

where C1(r)

and r.A = q.B;

RA for NOT IN:

4.

Because all the relations (A – E) have unary (same) schema and since & ,

…(since

5.

a.

b.

can be simplified into

This is because, b exists in U by default.

If b is being referred by V relation (since it is foreign key), we do not need to join U and V over ‘b’ to check if ‘b’ exists as the same in U as it is in V.

Hence the relation is as good as .

**Collaborated with: Akshat Arvind, Renuka Srishti**