$1.(\Pi_{e1.*}(E1) - \Pi_{e1.*}(E1 \times F)) \cup \Pi_{e2.*}(E2 \times F)$

$2.(\Pi_{true}(R) - (\Pi_{true}(R1 \bowtie_{r1.x<> r2.x} R2)) \cup \Pi_{false}(R1 \bowtie_{r1.x<> r2.x} R2)$

3.

a. $\Pi_{L2(s),L1(r)}(\sigma_{c2(s,r) \wedge c1(r)}(R \times S)) \cup \Pi_{L3(t),L1(r)}(\sigma_{c3(t,r) \wedge c1(r)}(T \times R))$

b. $\Pi_{L1}(\Pi_{L2(s),L1(r)}(\sigma_{c2(s,r) \wedge c1(r)}(R \times S)) \cap \Pi_{L3(t),L1(r)}(\sigma_{c3(t,r) \wedge c1(r)}(T \times R)))$

c. $\Pi_{L1}(\Pi_{L2(s),L1(r)}(\sigma_{c2(s,r) \wedge c1(r)}(R \times S)) - \Pi_{L3(t),L1(r)}(\sigma_{c3(t,r) \wedge c1(r)}(T \times R)))$

d. $\Pi_{L1(r)}(\sigma_{c1(r)}(R) - \Pi_{L1}(\Pi_{L2(s),L1(r)}(\sigma_{c2(s,r) \wedge c1(r)}(R \times S)) \cup \Pi_{L3(t),L1(r)}(\sigma_{c3(t,r) \wedge c1(r)}(T \times R)))$

e. $\Pi_{L1(r)}\ (\sigma_{c1(r)}(R) - \Pi_{L1}(\Pi_{\ L2(s),L1(r)}(\ \sigma_{\ c2(s,r)\ \wedge\ c1(r)}$ $(R\ X\ S)) \cap \Pi_{\ L3(t),L1(r)}(\ \sigma_{\ c3(t,r)\ \wedge\ c1(r)}(T\ X\ R)\ ))$

f. $\Pi_{L1(r)}\ (\sigma_{c1(r)}(R) - \Pi_{L1}(\Pi_{\ L2(s),L1(r)}(\ \sigma_{\ c2(s,r)\ \wedge\ c1(r)}$ $(R\ X\ S)) - \Pi_{\ L3(t),L1(r)}(\ \sigma_{\ c3(t,r)\ \wedge\ c1(r)}(T\ X\ R)\ ))$

4. To Prove - $_{a,d}(R \bowtie_{c=d} S) =_{a,d}(_{a,c}(R) \bowtie_{c=d}\ _{d}(S))$

Let LHS be equal to $_{a,d}(R \bowtie_{c=d} S)$

$=\{a,d \mid b\ c\ e\ (R(a,b,c) \wedge c = d \wedge S(d,e)\}$

$=\{a,d \mid b\ c\ \ (R(a,b,c) \wedge c = d \wedge (e\ S(d,e))\}$

$=\{a,d \mid c\ \ (b\ R(a,b,c) \wedge c = d \wedge (e\ S(d,e))\}$

$=\{a,d \mid c\ \ ((a,c)\ \Pi_{\ a,c}\ (R) \wedge c = (d\ _{d}\ (S))\}$

$=\{a,d \mid c\ ((a,c,d)\ {}_{a,c}(R) \bowtie_{c=d}\ {}_{d}\ (S))\}$

$=_{a,d}({}_{a,c}(R) \bowtie_{c=d}\ {}_{d}(S))$

=RHS

Hence the proof.

5. As given in the question we can assume that S has primary key d and R has foreign key c referencing this primary key in S.

$$\Pi_{a,d}(R \bowtie_{c=d} S) = \Pi_{a,d}(\Pi_{a,c}(R) \bowtie_{c=d} \Pi_{d}(S))$$

Consider RHS,

$=\Pi_{a,d}(\Pi_{a,c}(R) \bowtie_{c=d} \Pi_{d}(S))$

$=\Pi_{a,d}(\Pi_{d}(S) \bowtie_{c=d} (\Pi_{a,c}(R)))$

We can use the absorption rule as c is a foreign key referencing primary key d

$=\Pi_{a,d}(\Pi_{a,c}(R))$

$=\Pi_{a,d}(R)$

Considering the LHS,

$=\Pi_{a,d}(R \bowtie_{c=d} S)$

$=\Pi_{a,d}(S \bowtie_{c=d} R)$

Using the absorption law again we get,

$=\Pi_{a,d}(R)$

Since LHS is equal to the RHS we can conclude that the rewrite rule is correct.

6.Step-1. $\Pi_{c.cname,c.headquarter}$ (Company $\bowtie_{c.cname=w.cname}$ WorksFor$^{opt}{}_{t}$ $\bowtie_{w.pid=p.pid \wedge p.city<>'Bloomington'}$ Person)

Where,

WorksFor^{opt}=$\Pi_{w.*}$($\sigma_{w.salary<5500}$ $\bowtie$ WorksFor^{opt})

Step-2. $\Pi_{c.cname,c.headquarter}$ (Company $\bowtie_{c.cname=w.cname}$ WorksFor^{opt} $\bowtie_{w.pid=p.pid}$ Person^{opt})

Where,

Person^{opt}=$\Pi_{p.pid,p.city}$($\sigma_{p.city<>'Bloomington'}$ Person^{opt})

Step-3. $\Pi_{c.cname,c.headquarter}$ (Company $\bowtie$ WorksFor^{opt} $\bowtie$ Person^{opt}

7. $\Pi_{p.pid}$ ($\Pi_{p.pid}$(Person X Skill) - $\Pi_{p.pid,s.skill}$(Person $\bowtie_{p.pid=ps.pid}$ PersonSkill $\bowtie_{ps.skill=s.skill}$ Skill)) $\cap$ $\Pi_{p.pid}$ ($\Pi_{p.pid}$ (Person X Skill X

$\Pi_{p.pid}(\Pi_{p.pid,s1.skill,s2.skill}(Skill1 \bowtie_{s1.skill<>s2.skill} Skill2 \times Person) \cap (\Pi_{p.pid,s1.skill,s2.skill}(Skill1 \bowtie_{s1.skill<>s2.skill} Skill2 \times Person) - \Pi_{p.pid,s1.skill,s2.skill}(Skill1 \times Skill2 \times Person \bowtie_{p.pid=ps.pid \wedge s1.skill=ps.skill} PersonSkill)) \cap (\Pi_{p.pid,s1.skill,s2.skill}(Skill1 \bowtie_{s1.skill<>s2.skill} Skill2 \times Person) - \Pi_{p.pid,s1.skill,s2.skill}(Skill1 \times Skill2 \times Person \bowtie_{p.pid=ps.pid \wedge s2.skill=ps.skill} PersonSkill))$

Optimizing the above query

$\Pi_{p.pid}(\Pi_{p.pid}(Person^{opt} \times Skill) - \Pi_{p.pid,s.skill}(Person^{opt} \bowtie_{p.pid=ps.pid} PersonSkill \bowtie_{ps.skill=s.skill} Skill)) \cap \Pi_{p.pid}(\Pi_{p.pid}(Person^{opt} \times Skill \times \Pi_{p.pid}(\Pi_{p.pid,s1.skill,s2.skill}(Skill1 \bowtie_{s1.skill<>s2.skill} Skill2 \times Person^{opt}) \cap (\Pi_{p.pid,s1.skill,s2.skill}(Skill1 \bowtie$

$_{s1.skill<>s2.skill}$ Skill2 X Person^{opt})-$\Pi_{p.pid,s1.skill,s2.skill}$ (Skill1 X Skill2 X Person^{opt}$\bowtie_{p.pid=ps.pid \wedge s1.skill=ps.skill}$PersonSkill))

$\cap$ ($\Pi_{p.pid,s1.skill,s2.skill}$ (Skill1$\bowtie_{s1.skill<>s2.skill}$ Skill2 X Person^{opt}) - $\Pi_{p.pid,s1.skill,s2.skill}$(Skill1 X Skill2 X Person^{opt} $\bowtie_{p.pid=ps.pid \wedge s2.skill=ps.skill}$ PersonSkill))

Where,

Person^{opt}=$\Pi_{p.pid}$(Person)

8.$\Pi_{p.pid,p.pname}$ ($\sigma_{cl.city='Bloomington'}$ (Person $\bowtie_{p.pid=w.pid}$ WorksFor $\bowtie_{w.cname=cl.cname}$ CompanyLocation)) - $\Pi_{p.pid,p.pname}$ ($\sigma_{cl.city='Bloomington \wedge p1.city='Chicago'}$ (Person $\bowtie_{w.pid=p.pid}$ WorksFor $\bowtie_{w.cname=cl.cname}$ CompanyLocation $\bowtie_{p.pid=k.pid1}$ Knows $\bowtie_{k.pid2=p1.pid'}$ Person1))

Optimising the above expression

$\Pi_{p.pid,p.pname}$ ($\sigma_{cl.city='Bloomington'}$ ($Person^{opt} \bowtie_{p.pid=w.pid}$ $WorksFor^{opt} \bowtie_{w.cname=cl.cname}$ $CompanyLocation^{opt}$)) - $\Pi_{p.pid,p.pname}$ ($\sigma_{cl.city='Bloomington \wedge p1.city='Chicago'}$ ($Person^{opt} \bowtie_{w.pid=p.pid}$ $WorksFor^{opt} \bowtie_{w.cname=cl.cname}$ $CompanyLocation^{opt} \bowtie_{p.pid=k.pid1}$ $Knows^{opt} \bowtie_{k.pid2=p1.pid'}$ $Person1^{opt}$))

Where,

$Person^{opt=}\Pi_{p.pid,p.pname}$(Person)

$WorksFor^{opt=}\Pi_{w.pid,w.cname}$(WorksFor)

$CompanyLocation^{opt=}\Pi_{cl.cname}(\sigma_{cl.city='Bloomington}$(CompanLocation))

$Person1=\Pi_{p.pid}(\sigma_{city='Chicago'}$ (Person1)

9. $\Pi_{c.cname,c.headquarter}(Company \bowtie_{c.cname=w1.cname} worksFor1)$ -

$(\Pi_{c.cname,c.headquarter}(\Pi_{c.cname,c.headquarter,w.pid}(Company \bowtie_{w.cname=c.cname \wedge w.salary<70000} WorksFor) - \Pi_{c.cname,c.headquarter,w.pid}(Company \bowtie_{w.cname=c.cname \wedge w.salary<70000} WorksFor \bowtie_{ps.pid=w.pid \wedge ps.skill='Programming'} PersonSkill)) \cup \Pi_{c.cname,c.headquarter}(\Pi_{c.cname,c.headquarter,w.pid} (Company \bowtie_{w.cname=c.cname \wedge w.salary<70000} WorksFor) - \Pi_{c.cname,c.headquarter,w.pid} (Company \bowtie_{w.cname=c.cname \wedge w.salary<70000} WorksFor \bowtie_{ps1.skill=w.pid \wedge ps2.skill='AI'} PersonSkill2)))$

Optimising the above query

$\Pi_{c.cname,c.headquarter}(Company \bowtie_{c.cname=w1.cname} worksFor1^{opt})$ -

$(\Pi_{c.cname,c.headquarter}(\Pi_{c.cname,c.headquarter,w.pid}(Company \bowtie_{w.cname=c.cname \wedge w.salary<70000} WorksFor^{opt}) - \Pi_{c.cname,c.headquarter,w.pid}(Company \bowtie_{w.cname=c.cname \wedge w.salary<70000} WorksFor^{opt} \bowtie_{ps.pid=w.pid \wedge ps.skill='Programming'} PersonSkill^{opt})) \cup \Pi_{c.cname,c.headquarter}(\Pi_{c.cname,c.headquarter,w.pid} (Company \bowtie_{w.cname=c.cname \wedge w.salary<70000} WorksFor^{opt})-\Pi_{c.cname,c.headquarter,w.pid} (Company \bowtie_{w.cname=c.cname \wedge w.salary<70000} WorksFor^{opt} \bowtie_{ps1.skill=w.pid \wedge ps2.skill='AI'} PersonSkill1^{opt})))$

Where,

$WorksFor1^{opt}=\Pi_{c.cname} (WorksFor1)$

$WorksFor^{opt}=\Pi_{w.pid,w.cname} (\sigma_{salary<=70000} WorksFor)$

$PersonSkill^{opt} = \Pi_{ps.pid} (\sigma_{ps.skill='programming'} (PersonSkill))$

$PersonSkill1^{opt} = \Pi_{ps.pid} (\sigma_{ps.skill='AI'} (PersonSkill1))$

10. $\Pi_{p.pid,'t'} (Person \bowtie_{hm1.mid=p.pid} HasManager1 \bowtie_{hm2.mid<>hm1.mid \wedge hm2.mid=p.pid} HasManager2) \cup (\Pi_{p.pid,'f'} (Person \times HasManager1 \times HasManager2) - \Pi_{p.pid,'f'} (Person \bowtie_{hm1.mid=p.pid} HasManager1 \bowtie_{hm2.mid<>hm1.mid \wedge hm2.mid=p.pid} HasManager2$

Optimising the above expression

$\Pi_{p.pid,'t'} (Person^{opt} \bowtie_{hm1.mid=p.pid} HasManager1 \bowtie_{hm2.mid<>hm1.mid \wedge hm2.mid=p.pid}$

HasManager2) U ($\Pi_{p.pid,'f'}$ (Person^{opt} X HasManager1 X HasManager2) - $\Pi_{p.pid,'f'}$ (Person^{opt} $\bowtie_{hm1.mid=p.pid}$ HasManager1 $\bowtie_{hm2.mid<>hm1.mid \wedge hm2.mid=p.pid}$ HasManager2

Where,

Person^{opt}= $\Pi_{p.pid}$(Person)