Q1
a)

i) Select the names of suppliers who offer products tagged as 'PPE' and whose cost is less than 6.

ii) Select the names from the IDs of suppliers who offer products tagged as 'PPE' and whose cost is less than 6.

iii) Select the names of suppliers who both offer products tagged as 'PPE' and whose cost is less than 6, and products tagged as 'SuperTech' and whose cost is less than 6.

iv) Select the IDs of suppliers who either offer products tagged as 'PPE' and whose cost is less than 6, or products tagged as 'SuperTech' and whose cost is less than 6.

v) Select the names of suppliers who both offer products tagged as 'PPE' and whose cost is less than 6, and products tagged as 'SuperTech' and whose cost is less than 6.

b)

i) πsname(σtagname='PPE' ∨ tagname='Testing' (ProductTag ⋈ Catalog ⋈ Suppliers))

ii) πsid(σtagname='PPE' ∧ (cost < 10 ∨ cost > 420) (ProductTag ⋈ Catalog ⋈ Suppliers))

iii) πsid(σtagname='PPE' ∧ (cost >= 10 ∧ cost <= 1337)) (ProductTag ⋈ Catalog ⋈ Suppliers))

iv) πsid(σtagname='Cleaning' (ProductTag ⋈ Catalog ⋈ Suppliers))

v) Catalog1 := ρCatalog1(Catalog)
Catalog2 := ρCatalog2(Catalog)
CommonSuppliers := πCatalog1.sid, Catalog2.sid (σCatalog1.cost >= 1.2 × Catalog2.cost (Catalog1 × Catalog2))

vi) Catalog1 := ρCatalog1(Catalog)
Catalog2 := ρCatalog2(Catalog)
CommonProducts := πpid(σCatalog1.sid ≠ Catalog2.sid (Catalog1 × Catalog2))

vii)
SuperTechProducts := πsid, cost(σ(tagname='Super Tech' ∧ scountry='USA') (ProductTag ⋈ Catalog ⋈ Suppliers))
MaxCost := ρR2(sid2, cost2)(SuperTechProducts)
MatchingSuppliers := πsid, cost (SuperTechProducts ⋈ cost < cost2 MaxCost)

UniqueSuppliers := SuperTechProducts - MatchingSuppliers
Result := πsid(UniqueSuppliers)

viii) SuperTechProducts := πsid, cost(σ(tagname='Super Tech' ∧ scountry='USA') (ProductTag ⋈ Catalog ⋈ Suppliers))
MaxCost := ρR2(sid2, cost2)(SuperTechProducts)
LowerCost := πsid, cost (SuperTechProducts ⋈ cost < cost2 MaxCost)
SecondMaxCost := ρR4(sid3, cost3)(LowerCost)
SecondLowerCost := πsid, cost (LowerCost ⋈ cost < cost3 SecondMaxCost)
LowerThanSecondMax := LowerCost - SecondLowerCost
Result := πsid(LowerThanSecondMax)

ix) πpid(Product) - πpid(σcost >= 69(Catalog))

x) πpid(Product) - πpid(Inventory)

c)

i)
Subsuppliers1 := ρSubsuppliers1(Subsuppliers)
Subsuppliers2 := ρSubsuppliers2(Subsuppliers)

CommonProducts1 := πsid, subid, pid, cost(Subsuppliers1 ⋈ Subsuppliers2)
CommonProducts2 := πsid, subid, pid, cost(Subsuppliers2 ⋈ Subsuppliers1)

CommonProducts := πpid, sid1, sid2, cost1, cost2(σ(CommonProducts1.pid = CommonProducts2.pid ∧ CommonProducts1.sid ≠ CommonProducts2.sid ∧ CommonProducts1.cost = CommonProducts2.cost) (CommonProducts1 ⋈ CommonProducts2 ⋈ Inventory))

ii)
Catalog1 := ρCatalog1(Catalog)
Catalog2 := ρCatalog2(Catalog)

ProductPairs := πpid, sid, cost(Catalog1 ⋈ Catalog2)
UniqueProductPairs := ρR4(pid1, cost1)(ProductPairs)
UniquePrices := πpid, sid, cost(ProductPairs ⋈ (UniqueProductPairs.pid1 = ProductPairs.pid ∧ UniqueProductPairs.cost1 = ProductPairs.cost))

iii)
ProductTag1 := ρProductTag1(ProductTag)
ProductTag2 := ρProductTag2(ProductTag)
ProductTag3 := ρProductTag3(ProductTag)

PPEProducts := πpid(σ(tagname = 'PPE') ProductTag1)
NonSuperTechProducts := πpid(σ(tagname ≠ 'Super Tech') ProductTag2)
OtherProducts := πpid(σ(tagname ≠ 'Super Tech' ∧ tagname ≠ 'PPE') ProductTag3)

CommonProducts := PPEProducts ∩ NonSuperTechProducts ∩ OtherProducts

Result := πpid, pname, cost(CommonProducts ⋈ Product)

iv)
Subsuppliers1 := ρSubsuppliers1(Subsuppliers)
Subsuppliers2 := ρSubsuppliers2(Subsuppliers)

SuppliersWithBusinessRelationship := πsid, subid, sname, saddress(Subsuppliers1 ⋈ Subsuppliers2)
ReversedSuppliersWithBusinessRelationship := πsid, subid, sname, saddress(Subsuppliers2 ⋈ Subsuppliers1)

CommonSubsuppliers := πsubid, sid, sname, saddress(σ(SuppliersWithBusinessRelationship.subid = ReversedSuppliersWithBusinessRelationship.sid ∧ SuppliersWithBusinessRelationship.sid = ReversedSuppliersWithBusinessRelationship.subid) SuppliersWithBusinessRelationship)
UniqueSubsuppliers := πsubid, sid, sname, saddress(σ(SuppliersWithBusinessRelationship.subid = ReversedSuppliersWithBusinessRelationship.sid ∧ SuppliersWithBusinessRelationship.sid = ReversedSuppliersWithBusinessRelationship.subid) ReversedSuppliersWithBusinessRelationship)

UncommonSubsuppliers := CommonSubsuppliers ⋈ UniqueSubsuppliers
Result := SuppliersWithBusinessRelationship - UncommonSubsuppliers

d)
Revised version:
Supplier:
sid: Primary Key, integer
name: Supplier Name, text
address: Supplier Address, text
country: Supplier Country, text

Product:
pid: Primary Key, integer
name: Product Name, text

Product_Tag:

tid: Primary Key, integer
pid: Foreign Key, integer
tag_name: Tag Name, text

Catalog:
catalog_id: Primary Key, integer
sid: Foreign Key, integer
pid: Foreign Key, integer
cost: Product Cost, real

Inventory:
inventory_id: Primary Key, integer
pid: Foreign Key (references Product.pid), integer
quantity: Product Quantity, integer

The new schema improves upon the original by explaining employing clearer and better attribute names. For example, in the original schema, the 'Subsuppliers' table combines supplier and sub-supplier data, potentially leading to redundancy. In contrast, the new schema separates suppliers and sub-suppliers into distinct tables, enhancing clarity and reducing data duplication. Additionally, the new schema enforces referential integrity through foreign key constraints.

e)
Senior Developer: "I've used most of those products and I know those suppliers. I may manage the database better"
Me: "My dad runs some of those suppliers."
The senior developer was fired.

Q2:

i)

πutorid (Student) - πutorid (Approved ⋈ σ(roomname='IC404')(Room))

ii)

R1 ← π_utorid,roomid (Employee ⋈ Approved)

R2 ← ρR2(utorid2,roomid2) (R1)

R3 ← ρR3(utorid3,roomid3) (R1)

R4 ← R1 ⋈ utorid=utorid2 AND roomid<>roomid2 (R2)

R5 ← R4 ⋈ utorid=utorid3 AND roomid<>roomid3 AND roomid3<>roomid2 (R3)

R6 ← π_utorid (R5)

iii)

R1 ← π_utorid,roomid (Employee ⋈ Approved)

R2 ← ρR2(utorid2,roomid2) (R1)

R3 ← ρR3(utorid3,roomid3) (R1)

R4 ← ρR4(utorid4,roomid4) (R1)

R5 ← R1 ⋈ utorid=utorid2 AND roomid<>roomid2 (R2)

R6 ← R3 ⋈ utorid=utorid3 AND roomid<>roomid3 AND roomid3<>roomid2 (R5)

R7 ← R4 ⋈ utorid=utorid4 AND (roomid<>roomid4 OR roomid2<>roomid4 OR roomid3<>roomid4) (R5)

R8 ← π_utorid (R7)

iv)

R1 ← π_utorid,roomid (Employee ⋈ Approved)

R2 ← ρR2(utorid2,roomid2) (R1)

R3 ← ρR3(utorid3,roomid3) (R1)

R4 ← ρR4(utorid4,roomid4) (R1)

R5 ← R1 ⋈ utorid=utorid2 AND roomid<>roomid2 (R2)

R6 ← R5 ⋈ utorid=utorid3 AND roomid<>roomid3 AND roomid3<>roomid2 (R3)

R7 ← R6 ⋈ utorid=utorid4 AND roomid<>roomid4 AND roomid2<>roomid4 AND roomid3<>roomid4 (R4)

R8 ← πutorid (Employee) - π_utorid (R7)

v)

πroomid (σname='Oscar Lin' ∧ date>=2022-09-01 ∧ date<=2022-12-31 ∧ alertlevel > alertthreshold (Member ⋈ Approved ⋈ Room))

vi)

πutorid (Member) - πutorid (Approved ⋈ Room ⋈ σdate >= 2021-03-17 ∧ date <= 2022-12-31(Occupancy))

vii)

cannot be done in RA. We can not do summation in a specific tuple.

viii)
πutorid, email (σvaxstatus = 0 ∧ utorid = Member.utorid ∧ roomid = Approved.roomid ∧ alertlevel > alertthreshold (Occupancy ⋈ Room ⋈ Member))