Ex1:

Find the names of suppliers who supply some red part.
σ_sname(Π_sname(Π_sid((Π_pid(σ_color="red"(parts)))⋈catalog)⋈ suppliers))

Find the sids of suppliers who supply some red or green part.
σ_sid(Π_sid(Π_pid(σ_color="red" U color="green"(parts))⋈catalog))

Find the sids of suppliers who supply some red part or are at 221 Packer Street.
σ_sid(Π_sid((Π_pid(σ_color="red"(parts))) ⋈catalog) U Π_sid(σ_address="221 packer street"(suppliers)))

Find the sids of suppliers who supply some red part and some green part.
σ_sid(Π_sid((Π_pid(σ_color="red"(parts))) ⋈catalog) U Π_sid((Π_pid(σ_color="green"(parts))) ⋈catalog))

Find the sids of suppliers who supply every part.
σ_sid(Π_sid,pid(catalog) ÷ Π_pid(parts))

Find the sids of suppliers who supply every red part.
σ_sid(Π_sid,pid(catalog) ÷ Π_pid(σ_color="red"(parts)))

Find the sids of suppliers who supply every red or green part.
σ_sid(Π_sid,pid(catalog) ÷ Π_pid(σ_color="red" U color="green"(parts)))

Find the sids of suppliers who supply every red part or supply every green part.
σ_sid((Π_sid,pid(catalog) ÷ Π_pid(σ_color="red"(parts))) U (Π_sid,pid(catalog) ÷
Π_pid(σ_color="green"(parts))))

Find pairs of sids such that the supplier with the first sid charges more for some part than the supplier with the second sid.
ρCatalog →C1
ρCatalog →C2
σ_C1.sid, C2.sid(Π_C1.sid, C2.sid(σ_C1.cost > C1.cost ∩ C1.pid=C2.pid∩C1.sid != C2.sid(C1 x C2)))

Find the pids of parts supplied by at least two different suppliers.
ρCatalog →C1
 ρCatalog →C2
σ_C1.pid(Π_C1.pid(σ_C1.pid=C2.pid∩C1.sid != C2.sid(C1 x C2)))

Ex2:
1) names of suppliers who supply red parts that cost less than 100
2) names of suppliers who supply red parts that costs less than 100 and green parts that costs less than 100
3) sids of suppliers who supply red parts that costs less than 100 and green parts that costs less than 100
4) names of suppliers who supply red parts that costs less than 100 and green parts that costs less than 100