Omar Ahmadi **Exercise 3.18, 4.2, 4.3, 4.4, 4.6**

Exercise 3.18

```
CREATE TABLE  Artist(
        artistName CHAR(20),
        birthplace CHAR(20),
        age INTEGER,
        style CHAR(20),
        PRIMARY KEY (aristName) )
CREATE TABLE  Artwork(
        title CHAR(20),
        year INTEGER,
        type CHAR(15),
        price REAL,
        PRIMARY KEY (title) )
CREATE TABLE Paints (
        title CHAR (20),
        artist name CHAR(20) NOT NULL,
        type CHAR(20),
        price INTEGER,
        year INTEGER,
        PRIMARY KEY (title),
        FOREIGN KEY (artist_name) References Artist
        ON DELETE CASCADE ON UPDATE NO ACTION )
CREATE TABLE Group(
        groupName CHAR(20),
        PRIMARY KEY(groupName) )
CREATE Classify (
        title CHAR(20),
        groupName(20) ,
        PRIMARY KEY(title, name),
        FOREIGN KEY (title) REFERENCES Artwork Paints,
        FOREIGN KEY(groupName) REFERENCES Group )
CREATE TABLE Customer
        cust_id INTEGER,
        custName CHAR(20),
        address CHAR(20),
        amount INTEGER,
        PRIMARY KEY (cust_id) )
CREATE TABLE Like_Group (
        name CHAR(20),
        custName CHAR(20),
        PRIMARY KEY (groupName, custName),
        FOREIGN KEY (groupName) REFERENCES Group,
        FOREIGN KEY custName REFERENCES Customer )
CREATE TABLE Like_Artist (
        artistName CHAR(20),
        custName CHAR(20),
        PRIMARY KEY (artistName, custName),
        FOREIGN KEY (artistName) REFERENCES Artist,
        FOREIGN KEY custName REFERENCES Customer )
```

Exercise 4.2

Given two relations R1 and R2, where R1 contains N1 tuples, R2 contains N2 tuples, and N2 > N1 > 0, give the minimum and maximum possible sizes (in tuples) for the resulting relation produced by each of the following relational algebra expressions. In each case, state any assumptions about the schemas for R1 and R2 needed to make the expression meaningful:

Expression	Assumption	Min	Max
R1 ∪ R2	R1 and R2 are union compatible	N2	N1 + N2
R1 ∩ R2	R1 and R2 are union compatible	0	N1
R1 − R2	R1 and R2 are union compatible	0	N1
R1 × R2		N1 * N2	N1 * N2
$\sigma_{a=5}(R1)$	R1 has an attribute 'a'	0	N1
$\Pi_{a=5}(R1)$	R1 has an attribute 'a', N1 > 0	1	N1
R1 / R2		0	N2

Exercise 4.3

Suppliers(sid: integer, sname: string, address: string)
Parts(pid: integer, pname: string, color: string)
Catalog(sid: integer, pid: integer, cost: real)

1. Find the names of suppliers who supply some red part.

RA $\Pi_{sname}(\Pi_{sid}((\Pi_{pid}\ \sigma_{color='red'}\ Parts) \bowtie Catalog) \bowtie Suppliers$

TRC {T | ∃T1 ∈ Suppliers(∃X ∈ Parts(X.color = red ∧ ∃Y ∈ Catalog
(Y.pid = X.pid ∧ Y.sid = T1.sid)) ∧ T.sname=T1.sname)}

DRC {Y | <X, Y, Z>∈Suppliers∧∃P, Q, R(<P, Q, R> ∈ Parts
∧ R = red ∧ ∃I,J,K(<I,J,K> ∈ Catalog ∧ J = P ∧ I = X))}

2. Find the sids of suppliers who supply some red or green part.

RA $\Pi_{sid}(\Pi_{pid}(\sigma_{color='red' \vee color='green'}\ Parts) \bowtie Catalog)$

TRC {T | ∃ C∈ Catalog ∃ P∈ Parts ((P.color = 'red' ∨P.color = 'green') ∧C.pid =
P.pid ∧T.sid = C.sid)}

DRC {<Sid> | ∃ Pid, C (<Sid, Pid, C> ∈ Catalog ∧ ∃ Pn, Cl (<Pid, Pn, Cl> ∈ Parts
∧ (Cl = 'red' ∨ Cl = 'green'))))}

3. Find the sids of suppliers who supply some red part or are at 221 Packer Ave.

RA $\rho(R1, \Pi_{sid}((\Pi_{pid}\ \sigma_{color='red'}\ Parts) \bowtie Catalog))$
$\rho(R2, \Pi_{sid}\ \sigma_{address='221PackerStreet'}\ Suppliers)$
R1 ∪ R2

TRC {T | ∃ C∈ Catalog (∃ P∈ Parts (P.color = 'red' ∧C.pid = P.pid) ∧T.sid =
C.sid) ∨ ∃ S∈ Suppliers (S.address = '221 Packer Ave' ∧T.sid = C.sid)}

DRC {<Sid> | ∃ Pid, C (<Sid, Pid, C> ∈ Catalog ∧ ∃ Pn (<Pid, Pn, 'red'> ∈ Parts)
∨ ∃ Sn (<Sid, Sn, '221 Packer Ave'> ∈ Suppliers)}

4. Find the sids of suppliers who supply some rcd part and some green part.

RA $\rho(R1, \Pi_{sid}((\Pi_{pid}\ \sigma_{color='red'}\ Parts) \bowtie Catalog))$
$\rho(R2, \Pi_{sid}((\sigma_{color='green'}\ Parts) \bowtie Catalog))$
R1 ∩ R2

TRC {T | ∃ C1∈ Catalog (∃ P1∈ Parts (P1.color = 'red' ∧ C1.pid = P1.pid) ∧ T.sid = C.sid) ∧ C2∈ Catalog (∃ P2∈ Parts (P2.color = 'green' ∧ C2.pid = P2.pid) ∧ C1.sid = C2.sid) ∧ T.sid = C1.sid)}

DRC {<Sid> | ∃ Pid, C (<Sid, Pid, C> ∈ Catalog ∧ ∃ Pn (<Pid, Pn, 'red'> ∈ Parts) ∧ ∃ Pid2, C2 (<Sid, Pid2, C2> ∈ Catalog ∧ ∃ Pn2 (<Pid2, Pn2, 'green'> ∈ Parts)))}

5. Find the sids of suppliers who supply every part.

RA (Πsid,pid Catalog)/(Πpid Parts)

TRC {T | ∃ C∈ Catalog ∀ P∈ Parts (∃ C2∈ Catalog (C2.pid = P.pid ∧ C.sid = C2.sid) ∧ C.sid = T.sid)}

DRC {<Sid> | ∃ Pid, C (<Sid, Pid, C> ∈ Catalog ∧ ∀ <P, Pn, Cl> ∈ Parts (∃ Pid2, C2 (<Sid, Pid2, C2> ∈ Catalog (Pid2 = P)))}

6. Find the sids of suppliers who supply every red part.

RA (Πsid,pid Catalog)/(Πsid σcolor='red' Parts)

TRC {T | ∃ C∈ Catalog (∀ P∈ Parts (P.color = 'red' ⇒ (∃ C2∈ Catalog (C2.pid = P.pid ∧ C2.sid = C.sid))) ∧ T.sid = C.sid)}

DRC {<Sid> | ∃ Pid, C (<Sid, Pid, C> ∈ Catalog ∧ ∀ <P, Pn, Cl> ∈ Parts (Cl ≠ 'red' ∨ ∃ Pid2, C2 (<Sid, Pid2, C2> ∈ Catalog (Pid2 = P))))}

7. Find the sids of suppliers who supply every red or green part.

RA ($\pi_{sid, pid}$ Catalog) / (π_{pid} ($\sigma_{color = 'red' \vee color = 'green'}$ (Parts)))

TRC {T | ∃ C∈ Catalog (∀ P∈ Parts ((P.color = 'red' ∨ P.color = 'green') ⇒ (∃ C2 ∈ Catalog (C2.pid = P.pid ∧ C2.sid = C.sid))) ∧ T.sid = C.sid)}

DRC {<Sid> | ∃ Pid, C (<Sid, Pid, C> ∈ Catalog ∧ ∀ <P, Pn, Cl> ∈ Parts ((Cl ≠ 'red' ∧ Cl ≠ 'green') ∨ ∃ Pid2, C2(<Sid, Pid2, C2> ∈ Catalog (Pid2 = P))))}

8. Find the sids of suppliers who supply every red part or supply every green part.

RA (($\pi_{sid, pid}$ Catalog) / (π_{pid} ($\sigma_{color = 'red'}$ (Parts)))) ∪ (($\pi_{sid, pid}$ Catalog) / (π_{pid} ($\sigma_{color = 'red'}$ (Parts))))

TRC {T | ∃ C∈ Catalog ((∀ P∈ Parts (P.color = 'red' ⇒ (∃ C2∈ Catalog (C2.pid = P.pid ∧ C2.sid = C.sid))) ∨ ∀ P2∈ Parts (P2.color = 'green' ⇒ (∃ C3∈ Catalog (C3.pid = P2.pid ∧ C3.sid = C.sid)))) ∧ T.sid = C.sid)}

DRC {<Sid> | ∃ Pid, C (<Sid, Pid, C> ∈ Catalog ∧ (∀ <P, Pn, Cl> ∈ Parts (Cl ≠ 'red' ∨ ∃ Pid2, C2 (<Sid, Pid2, C2> ∈ Catalog(Pid2 = P))) ∨ ∀ <P2, Pn2, Cl2> ∈ Parts (Cl2 ≠ 'green' ∨ ∃ Pid3, C3 (<Sid, Pid3, C3> ∈ Catalog (Pid3 = P2)))))}

9. Find pairs of sids such that the supplier with the first sid charges more for some part than the supplier with the second sid.

RA ρ (C1, Catalog)
ρ (C2, Catalog)
$\pi_{C1.sid, C2.sid}$ ($\sigma_{C1.pid = C2.pid \wedge C1.sid \neq C2.sid \wedge C1.cost > C2.cost}$ (C1 × C2))

TRC {T | ∃ C1∈ Catalog (∃ C2∈ Catalog (C1.pid = C2.pid ∧ C1.sid ≠ C2.sid ∧ C1.cost > C2.cost) ∧ T.sid1 = C1.sid ∧ T.sid2 = C2.sid)}

DRC {<Sid1, Sid2> | ∃ Pid1, C1 (<Sid1, Pid1, C1> ∈ Catalog ∧ ∃ Pid2, C2 (<Sid2, Pid2, C2> ∈ Catalog ∧ Pid1 = Pid2 ∧ Sid1 ≠ Sid2 ∧ C1 > C2))}

10. Find the pids of parts supplied by at least two different suppliers.

RA ρ (C1, Catalog)
ρ (C2, Catalog)
$\pi_{C1.pid}$ ($\sigma_{C1.pid = C2.pid \wedge C1.sid \neq C2.sid}$ (C1 × C2))

TRC {T | ∃ C1∈ Catalog (∃ C2∈ Catalog (C1.pid = C2.pid ∧ C1.sid ≠ C2.sid) ∧

T.pid = C1.pid)}

DRC {<Pid> | ∃Sid, C (<Sid, Pid, C> ∈ Catalog ∧ ∃Sid2, C2 (<Sid2, Pid, C2> ∈ Catalog ∧ Sid ≠ Sid2))}

11. Find the pids of the most expensive parts supplied by suppliers named Yosemite Sham.

RA ρ (Temp1, π_{sid} $\sigma_{sname = 'Yosemite\ Sham'}$ Suppliers)
ρ (Temp2, Temp1 |×| Catalog)
ρ (Temp3, Temp2)
ρ (Temp4 (1→sid, 2→pid, 3→cost), $\sigma_{Temp3.cost < Temp2.cost}$ (Temp3 × Temp2))
π_{pid} (Temp2 - $\pi_{sid, pid, cost}$ Temp4)

TRC {T | ∃C∈Catalog (∃S∈ Suppliers (S.sid = C.sid ∧S.sname = 'Yosemite Sham') ∧ ¬(∃S2∈ Suppliers (S2.sname = 'Yosemite Sham' ∧ ∃C2∈ Catalog (C2.sid = S2.sid ∧C2.cost > C.cost))) ∧T.pid = C.pid)}

DRC {<Pid> | ∃Sid, C (<Sid, Pid, C> ∈Catalog (∃Addr (<Sid, 'Yosemite Sham', Addr> ∈ Suppliers) ∧ ¬(∃ Sid2, Addr2 (<Sid2, 'Yosemite Sham', Addr2>∈ Suppliers (∃Pid2, C2 (<Sid2, Pid2, C2> ∈Catalog (C2 > C)))))}

12. Find the pids of parts supplied by every supplier at less than $200. (If any supplier either does not supply the part or charges more than $200 for it, the part is not selected.)

RA ($\pi_{pid, sid}$ ($\sigma_{cost < 200}$ (Catalog)) / (π_{sid} Suppliers)

TRC {T | ∃C∈ Catalog (∀S∈ Suppliers (∃C2∈ Catalog (C2.pid = C.pid ∧C2.cost < 200 ∧C2.sid = S.sid))) ∧T.pid = C.pid)}

DRC {<Pid> | ∃Sid, C (<Sid, Pid, C> ∈ Catalog ∧ ∀<S, Sn, Addr> ∈ Suppliers (∃Sid2, C2 (<Sid2, Pid, C2> ∈ Catalog (C2 < 200 ∧Sid2 = S))))}

Exercise 4.4

1. Find the supplier names of suppliers with a red part that costs less than 100 dollars.

2. Does not exist, does not return anything

3. Find the supplier names of the suppliers with a red part that costs less than 100 dollars and a green part that costs less than 100 dollars.

4. Find the supplier ids of the suppliers with a red part that costs less than 100 dollars and a green part that costs less than 100 dollars.

5. Find the supplier names of the suppliers with a red part that costs less than 100 dollars and a green part that costs less than 100 dollars.

Exercise 4.6

Relational completeness means a query language can express every query that can be written in relational algebra. If a query language is relationally complete, it does not mean that the language can express any given query.