

Midterm

Express the following queries using relational algebra (ALG), tuple calculus (TRC), domain calculus (DRC), and SQL to express the following queries.

Person

<u>P#</u>	Name	Age	City
P1	Last	20	Ottawa
P2	Jones	30	Toronto
P3	Blake	25	Vancouver
P4	Clark	20	Montreal
P5	Adams	30	Calgary

Team

<u>T#</u>	Name	City
T1	Maple Leafs	Toronto
T2	Canucks	Vancouver
T3	Canadiens	Montreal
T4	Senators	Ottawa

Fan

<u>P#</u>	<u>T#</u>	Rank
P1	T1	2
P1	T2	3
P1	T3	4
P1	T4	1
P2	T1	1
P2	T2	2
P2	T3	4
P3	T1	1
P3	T2	2
P4	T1	1
P4	T2	2

1. Find the person who is not a fan of any team

ALG

T1:= project Name (Person);
T2:= project Name (Person njoin Fan);
T1 minus T2;

TRC

{P.Name | P in Person and not (exists F in Fan) (P.P# = F.P#)};

DRC

{N | (exists P#)(Person (P#, N, _, _) and not (exists T#) (Fan (P#,T#, _)));}

SQL

select P.Name from Person P where not exists (select * from Fan F where P.P# = F.P#);

Result: Adams

2. Find the person who is a fan of the team in the city where they live

ALG

T1: = project T#, City(Team)) ;
T2: = Person njoin Fan njoin T1;
Project Name (T2);

TRC

{P.name | P in Person and exists (F in Fan, T in Team)(P.P#=F.P# and F.T#=T.T# and
P.City =T.City)}

DRC

{N | (exists P, C, T)(Person(P, N, _, C) and Team(T, _, C) and Fan(P, T, _))};

SQL

select P.Name from Person P, Fan F, Team T where P.P# = F.P# and F.T# = T.T#
and P.City = T.City;

Result: Blake, Jones, Last

3. Find the person who is a fan of all teams

ALG

T1:= project T# (Team);
T2:= project P#,T#(Fan)
T3:= T2 divideby T1;
T4:= Person njoin T3;
project name (T4);

TRC

{P.Name | P in Person and (forall T in Team) (exists F in Fan)(P.P# =F.P# and
F.T# = N.T#)};

DRC

{Name | (exists P)(Person (P, Name, _, _) and (forall T)(not Team (T,_,_) or
Fan(P, T, _))};

SQL

select P.Name from Person P where not exists (
select * from Team T where not exists (
select * from Fan F where P.P# = F.P# and F.T# = T.T#));

Result: Last

4. Find the person who is a fan of all teams except Senators

ALG

T1:= project T# (select name != 'Senators' (Team));
 T2:= project P#,T#(Fan)
 T3:= T2 divideby T1;
 T4:= select name = 'Senators' (Team)
 T5:= project P# (Fan njoin T4))
 T6:= T3 minus T5;
 project name (Person njoin T5);

TRC

{P.Name | P in Person and (forall T in Team)
 (T.Name = 'Senators' and not (exists F in Fan) (P.P# = F.P# and F.T# =
 T.T#))
 or
 (N.Name != 'Senators' and (exists F in Fan) (P.P# = F.P# and F.T# =
 T.T#))});

DRC

{N | (exists P)(Person (P, N, _, _) and (forall T)
 (not (exists M)(Team(T, M, _) and (M!= 'Senators') or Fan (P, T, _))
 and
 (not (exists M)(Team(B, M, _) and M = 'Senators') or not Fan (P, T, _))));

SQL

select P.name from Person P where not exists(
 select * from Team T where (T.name != 'Senators' or exists (
 select * from Fan F where P.P# = F.F# and F.T# = T.T#))
 and (T.name = 'Senators' or not exists (
 select * from Fan F where P.P# = F.F# and F.T# = T.T#)));

Result: Jones

- Find the person who is fan of more than two teams

ALG

T1:= Person njoin Fan
 T2(Name, C) := aggregate name, count(*) (T1);
 project name (select C >=2 (T2));

TRC

T1(Name, C) := {P.name, count(*) | P in Person and (exists F in Fan)(P.P#=F.P#)};
 {P.name | P in T1 and P.Count > 2};

DRC

$T1(\text{Name}, C) := \{N, \text{count}(\ast) \mid (\text{exists } P)(\text{Person}(P, N, _, _) \text{ and } \text{Fan}(P, _, _))\};$
 $\{N \mid (\text{exists } C)(T1(N, C) \text{ and } C > 2)\};$

SQL

```
select P.name
from Person P, Fan F
where P.P# = F.P#
group by P.name
having count(*) > 2;
```

Result: Jones, Last