

COMP 3005
Assignment #3
Due: October 14

Instruction

1. You should do the assignments independently. Copying is not allowed
2. Submit your assignment as a single word/PDF document on [culearn](#).

Queries (64 marks)

Use both Tuple Relational Calculus (TRC) and Domain Relational Calculus (DRC) to express the following 12 queries and Query By Example (QBE) to express queries 1,2,4,5,7,8 based on the given suppliers-and-parts databases. You should include each question with its number, the two or three kinds of queries, and query results just for 11 and 12.

Suppliers

<u>S#</u>	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	30	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

Parts

<u>P#</u>	PNAME	COLOR	WEIGHT	CITY
P1	Nut	Red	12.0	London
P2	Bolt	Green	17.0	Paris
P3	Screw	Blue	17.0	Oslo
P4	Screw	Red	14.0	London
P5	Cam	Blue	12.0	Paris
P6	Cog	Red	19.0	London

SP

<u>S#</u>	<u>P#</u>	QTY
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P1	300
S3	P2	200
S4	P2	200
S4	P3	300
S4	P4	400
S4	P5	500
S4	P6	600

1. Get the part names that Adams provides.

```
TRC> {P.pname | P in Parts and (exists C in SP, S in suppliers)
      (S.s# = C.s# and C.P# = P.P# and S.sname = 'Adams')};
```

```
DRC> {PN | (exists P,S,SN) (Suppliers(S,SN,_,_)and
      Parts(P,PN,_,_,_)and SP(S,P,_) and SN = 'Adams')};
```

It can also be written in this way:

```
DRC> {PN | (exists P, S) (Suppliers(S, 'Adams', _, _) and SP(S, P, _) and
    Parts(P, PN, _, _, _))};
```

QBE>

Suppliers

S#	SNAME	STATUS	CITY
<u>S</u>	Adam		

SP

S#	P#	QTY
<u>S</u>	<u>P</u>	

Parts

P#	PNAME	COLOR	WEIGHT	CITY
<u>P</u>	P.			

2. Get supplier names for suppliers who supply part P2.

```
TRC> {S.sname | S in suppliers and (exists C in SP) (S.s# = C.s# and
    C.P# = 'P2')};
```

```
DRC> {SN|(exists S){Suppliers(S, SN, _, _, _) and SP(S, 'P2', _)}};
```

QBE>

Suppliers

S#	SNAME	STATUS	CITY
<u>S</u>	P.		

SP

S#	P#	QTY
<u>S</u>	P2	

3. Get supplier names and the total quantity of parts supplied.

```
TRC> {S.sname, sum(C.QTY)|S in suppliers and C in SP and S.s# = C.s#};
```

```
DRC> {SN,sum(QTY)|(exists S) (Suppliers(S, SN, _, _) and SP(S, _, QTY))};
```

4. Get supplier names for suppliers who supply parts with quantity less than 300

```
TRC> {S.sname |S in suppliers and (exists C in SP) (S.s#=C.s# and
    C.QTY< 300)};
```

```
DRC> {SNA|(exists S,QTY) (Suppliers(S, SN, _, _) and SP(S, _ ,QTY) and QTY<300)};
```

QBE>

Suppliers

S#	SNAME	STATUS	CITY
<u>S</u>	P.		

SP

S#	P#	QTY
<u>S</u>		<300

5. Get supplier names for suppliers who supply either blue or green parts.

```
TRC> {S.sname |S in suppliers and (exists C in SP, P in Parts)
    (S.S#=C.S# and C.P#=P.P# and (P.color='Blue' or P.color='Green'))};
```

```
TRC> {S.sname |S in Suppliers and (
    (exists S1 in SP, P1 in Parts)
    (P1.color='Blue' and S1.p# = P1.p# and S.s# = S1.s#)
    or
    (exists S2 in SP, P2 in Parts)
    (P2.color = 'Green' and S2.p# = P2.p# and S.s# = S2.s#))};
```

```
DRC> {SN |(exists S,P,CL) (Suppliers(S, SN, _, _) and Parts(P, _, CL, _, _) and
    SP(S, P, _) and (CL= 'Green' or CL= 'Blue'))};
```

QBE>

Suppliers

S#	SNAME	STATUS	CITY
_S	P.		

SP

S#	P#	QTY
_S	_P	

Parts

P#	PNAME	COLOR	WEIGHT	CITY
_P		blue		
_P		green		

6. Get supplier name/part name pairs such that the indicated supplier does not supply the indicated part.

TRC> {S.sname,P.pname|S in suppliers and P in Parts and
not (exists C in SP) (S.S# = C.S# and C.P# = P.P#)};

DRC> {SN,PN|(exists S,P) (Suppliers(S,SN,_,_)and Parts(P#,PN,_,_,_)and not SP(S,P,_))};

It CANNOT be written in this way:

{SN, PN |not(exists S, P(Suppliers(S,SN,_,_) and SP(S,P,_)
and Parts(P,PN,_,_,_))};

7. Get supplier names for suppliers who does not supply any parts.

TRC> {S.sname|S in suppliers and not(exists C in SP) (S.S# = C.S#)};

DRC> {SN|(exists S) (Suppliers(S,SN,_,_)and not SP(S,_,_))};

It cannot be written in this way:

{SN|not(exists S) (Suppliers(S,SN,_,_) and SP(S,_,_))};

QBE>

Suppliers

S#	SNAME	STATUS	CITY
_S	P.		

-

SP

S#	P#	QTY
_S		

8. Get supplier names for suppliers who supply all parts.

TRC> {S.sname | S in suppliers and (forall P in Parts) (exists C in SP)
(S.s# = C.s# and P.p# = C.p#) };

DRC> {SN | (exists S) (Suppliers(S,SN,_,_) and
(forall P) (Parts(P,_,_,_,_)and SP(S, P, _)))};

QBE>

Suppliers

S#	SNAME	STATUS	CITY
_S	P.		

-

Parts

P#	PNAME	COLOR	WEIGHT	CITY
_P				

-

SP

S#	P#	QTY
_S	_P	

9. Get supplier names for suppliers who supply all parts except P1.

TRC> {S.sname | S in suppliers and (forall P in Part)
(P.P#='P1' and not (exists C in SP) (S.s#=C.s# and P.p#=C.p#))
or
(P.p#!='P1' and (exists C in SP) (S.s#=C.s# andP.p#=C.p#))) };

DRC> {SN | (exists S) (Suppliers(S, SN,_,_) and

```
(forall P) ((Parts(P,_,_,_) and
              (P='P1' and not SP(S,P,_)
               or
               (P!='P1' and SP(S,P,_) ))));
```

10. Get supplier names and the number of parts and total quantity supplied.

```
TRC> {S.sname,count(C.p#),sum(C.QTY)|S in suppliers and C in SP and
      S.s# = C.s#};
```

```
DRC> {SN, count(P),sum(QTY)|(exists S,P)(Suppliers(S,_,_,_) and SP(S,P,QTY))};
```

11. Get supplier names for suppliers who supply all parts that Jones supplies

```
TRC> {S'.sname | S' in Supplier and S'.sname != 'Jones' and
      (exists S in Supplier)(S.sname = 'Jones' and
      (forall P in Parts)
      ((exists C in SP)(S.S# = C.S# and C.P# = P.P#) and
      (exists C' in SP)(S'.S# = C'.S# and C'.P# = P.P#)
      or
      not (exists C in SP)(S.S# = C.S# and C.P# = P.P#)))}
```

```
DRC> {N |(exists S',S)(Suppliers(S', N) and N != 'Jones' and Suppliers(S,'Jones',_)
      and(forall P)(Parts(P,_,_,_) and
      (SP(S,P,_) and SP(S',P)
      or
      not SP(S,P,_))))}
```

12. Get supplier names for suppliers who supply only those parts that Jones supplies

```
TRC> {S'.sname | S' in Supplier and S'.sname != 'Jones' and
      (exists S in Supplier)(S.sname = 'Jones' and
      (forall P in Parts)
      ((exists C in SP)(S.S# = C.S# and C.P# = P.P#) and
      (exists C' in SP)(S'.S# = C'.S# and C'.P# = P.P#)
      or
      not (exists C in SP)(S.S# = C.S# and C.P# = P.P#)and
      not (exists C' in SP)(S'.S# = C'.S# and C'.P# = P.P#)))}
```

```
DRC> {N |(exists S',S)(Suppliers(S', N) and N != 'Jones' and Suppliers(S,'Jones',_)
      and(forall P)(Parts(P,_,_,_) and
      (SP(S,P,_) and SP(S',P)
      or
      not SP(S,P,_)and not SP(S',P)))}
```