William O'Malley
10/09/2018
CS 348 Homework 2

1) (30 pts) Consider the following relational schema:
Suppliers (sid, sname, address)
Parts (pid, pname, color)
Catalog (sid, pid, cost)
Write the Relational Algebra (RA), Tuple Relational Calculus (TRC), and Domain Relational Calculus (DRC) expressions to answer each of the following queries. Note that some of these queries might not be expressible in Relational Algebra or Relational Calculus. For such queries, informally explain why they cannot be expressed:

a. Find the names of the suppliers who supply some red or green part.

(RA):

$\pi_{sname}(\pi_{sid}((\pi_{pid}\, \sigma_{color="red" \vee color="green"}\, (Parts)) \bowtie (Catalog)) \bowtie (Suppliers))$

(TRC):

$\{t \mid \exists s \in Suppliers(\exists u \in Parts((u.color = 'red' \vee u.color = 'green') \wedge \exists x \in Catalog\, (x.pid = u.pid \wedge x.sid = s.sid)) \wedge t.sname = s.sname)\}$

(DRC):

$\{<sname> \mid <ssid, sname, address> \in Suppliers \wedge \exists ppid, pname, color(<ppid, pname, color> \in Parts \wedge (color = 'red' \vee color = 'green') \wedge \exists csid, cpid, cost(<csid, cpid, cost> Catalog \wedge csid = ssid \wedge cpid = ppid))\}$

b. Find the sids of suppliers that supply every part.

(RA):

$(\pi_{sid,\ pid}\, Catalog) \div (\pi_{pid}\, Parts)$

(TRC):

$\{t \mid \exists s \in Catalog(\forall u \in Parts(\exists x \in Catalog(u.pid = x.pid \wedge x.sid = s.sid))\ t.sid = s.sid)\}$

(DRC):

$\{<csid> \mid <csid,cpid,cost> \in Catalog \wedge \forall <ppid,pname,color> \in Parts\ (\exists <csid2,cpid2,cost2> \in Catalog(csid = csid2 \wedge cpid2 = ppid))\}$

2) (45 pts) Consider the following relational schema:
Flights (flno, from, to, distance, departs, arrives)
Aircraft (aid, aname, cruisingrange)

Certified (eid, aid)
Employees (eid, ename, salary)
Write the Relational Algebra (RA), Tuple Relational Calculus (TRC), and Domain Relational Calculus (DRC) expressions to answer the following queries. Note that some of these queries might not be expressible in Relational Algebra or Relational Calculus. For such queries, informally explain why they cannot be expressed:

a. Find the names of pilots certified for some Boeing aircraft.

(RA):

$\pi_{ename}(\sigma_{aname="Boeing"}$ (Aircraft ⋈ Certified ⋈ Employees))

(TRC):

{t.ename | ∃t ∈ Employees ∧ ∃s ∈ Certified (∃u ∈ Aircraft(u.aid = s.aid ∧ u.aname = 'Boeing' ∧ t.eid = s.eid))}

(DRC):

{<ename> | <eeid, ename, salary> ∈ Employees ∧ ∃ceid, caid(<ceid, caid> ∈ Certified ∧ ∃aaid, aname, cruisingrange(<aaid, aname, cruisingrange> ∈ Aircraft ∧ aaid = caid ∧ aname = 'Boeing' ∧ eeid = ceid)}

b. Identify the flights that can be certified for all pilots whose salary is more than $100,000.

(RA):

$\pi_{flno}(\sigma_{distance<cruisingrange \wedge salary>100,000}$(Flights ⋈ Aircraft ⋈ Certified ⋈ Employees)))

(TRC):

{t.flno | t ∈ Flights ∧ ∃a ∈ Aircraft ∧ ∃c ∈ Certified ∧ ∃e ∈ Employees(e.salary > 100, 000 ∧ a.aid = c.aid ∧ e.eid = c.eid ∧ a.cruisingrange > t.distance)}

(DRC):

{<flno> | <flno, from, to, dist, departs, arrives> ∈ Flights ∧ ∃ceid, caid(<ceid, caid> ∈ Certified ∧ ∃aaid, aname, cruisingrange(<aaid, aname, cruisingrange> ∈ Aircraft ∧ ∃eeid, ename, salary(<eeid, ename, salary> ∈ Employees(salary > 10000 ∧ eeid = ceid ∧ caid = aaid ∧ cruisingrange > dist))))}

c. Find the eids of the employees that are certified for the largest number of aircrafts.

This is not possible for RA, TRC, or DRC because the question requires an answer that can count or order employees by the number of aircrafts they are certified for. There is not an operation in RA, TRC, or DRC to achieve this, however it is possible to do this in SQL using the COUNT operation along with ordering the table.

3) (25 pts) Consider the following relational schema for a library database:
Book (Book_id, Title, Publisher_name)
Book_Authors (Book_id, Author_name)
Publisher (Name, Address, Phone)
Book_Copies(Book_id, Branch_id, No_of_copies)
Book_Loans (Book_id, Branch_id, Card_no, Date_out, Due_date)
Library_Branch (Branch_id, Branch_name, Address)
Borrower (Card_no, Name, Address, Phone)
Write the Relational Algebra (RA), Tuple Relational Calculus (TRC), and Domain Relational Calculus (DRC) expressions to answer the following query:
Retrieve the names of borrowers who do not have any books checked out.

(RA):

Checked <- π_{Card_no} (Borrower) - π_{Card_no} (Book_Loans)

Res <- π_{Name} (Borrower ⋈ Checked)

(TRC):

{t.Name|t ∈ Borrower ∧ ¬(∃s ∈ Book_Loan(t.CardNo = s.CardNo))}

(DRC):

{<bname> | <bcard_no, bname, baddress, bphone> ∈ Borrower ∧ ¬(∃lbook_id, lbranch_id, lcard_no, ldate_out, ldue_date(<lbook_id, lbranch_id, lcard_no, ldate_out, ldue_date > ∈ Book_Loans(bcard_no = lcard_no)))}