CS348 - Homework 2 SPRING 2019

Due Sunday February 17, 2019 at 11:59PM on Blackboard. (There will be a 10% penalty for each late calendar day. After five calendar days, the homework will not be accepted.)

(100 Points)- Relational Algebra, Relational Calculus, and TRC

1) (30 pts) Consider the following relational schema:

Student (sid, sname, address)

Course (cid, cname, subject)

Enrolled (sid, cid)

Where subject can be Math, English, CS, Physics, etc. Now, write the Relational Algebra (RA), Tuple Relational Calculus (TRC), and Domain Relational Calculus (DRC) 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 students who are enrolled in some Math or CS courses.

Π sname (σ subject = "Math" OR "CS" ((Student ⋈ Enrolled) ⋈ Course))

{T | S ∈ Student(T[sname] = S[sname]

^ ∃ E ∈ Enrolled(S[sid] = E[sid])

^ ∃ C ∈ Course(E[cid] = C[cid])

^ (C[subject] = "Math" V C[subject] = "CS"))}

{ <sname> | ∃ sid, address <sname, sid, address> ∈ Student
^ ∃ cid <sid, cid> ∈ Enrolled
^ ∃ cname, subject <scid, cname, subject> ∈ Course
^ (subject = "Math" V subject = "CS")}

b. Find the sid(s) of the student(s) who enrolled in all the CS classes.

This query can't be expressed in relational algebra or TRC because it is an aggregate query

{ <sid> | ∃ sname, address <sname, sid, address> ∈ Student
^ ∀ cid (¬ (<cid, sname, subject> ∈ Course
V subject != "CS")

^ ∃ sid<sid, cid> ∈ Enrolled

}

2) (45 pts) Consider the following relational schema:

Factory(fid, fname, region)

Suppliers (sid, sname, region, commission)

Parts (pid, pname, color)

Catalog (sid, pid, cost)

a. Find the names of suppliers that provide "Car_part", i.e., with pname = "Car_part".

Π sname (σ pname = "Car_part" ((Suppliers ⋈ Catalog) ⋈ Parts))

{T | S ∈ Suppliers(T[sname] = S[sname]

^ ∃ C ∈ Catalog(S[sid] = C[sid])

^ ∃ P ∈ Parts(P[pid] = C[pid])

^ P[pname] = "Car_part")}

{ <sname> | ∃ sid, region, commission <sname, sid, address> ∈ Suppliers
^ ∃ pid <sid, pid, cost> ∈ Catalog
^ ∃ pname <sid, cid> ∈ Parts
^ (pname = "Car_part")}

b. Identify the factories that are in the same region as that of a supplier whose commission is less than 10%.

Π fid (σ commission < 10 (Factory ⋈ Suppliers))

{T | F ∈ Factory(T[fid, fname, region] = F[fid, fname, region]

^ ∃ S ∈ Suppliers(F[region] = S[region])

^ S[Commission] < 10)}

{ <fid, fname, region>|∃ sid, commission<sid, sname, region, commission> ∈ Suppliers
^ ∃ <fid, fname, region> ∈ Factory
^ commission < 10}

a. Find the names of the suppliers who supply the largest number of parts.

This query can't be expressed in relational algebra or TRC because it is an aggregate query

{ <sname>|∃ sid, region, commission<sid, sname, region, commission> ∈ Suppliers
^ ∃ pid <sid, pid, cost> ∈ Catalog
^ COUNT(sid) = MAX(sid)}

3) (25 pts) Consider the following relational schema for a library database:

Employee (eid, ename, dnum, address)

Department (dnum, dname, manager_id)

Dept_Location (dnum, dlocation)

Project (pnum, pname, dnum, dlocation)

Works-On (eid, pnum, hours)

Now, write the Relational Algebra (RA), Tuple Relational Calculus (TRC), and Domain Relational Calculus (DRC) to answer the following query: Retrieve the names of employees who do not work on any project.

Π ename (Π ename(Employee) – Π ename(Employee ⋈ Works-On))

{T | E ∈ Employee(T[ename] = T[ename]} –
{T | E ∈ Employee(T[ename] = T[ename]
^ ∃ W ∈ Works-On(E[eid] = W[eid])}

{ <ename>|∃ eid, dnum, address<eid, ename, dnum, address> ∈ Employee} –

{ <ename>|∃ eid, dnum, address<eid, ename, dnum, address> ∈ Employee

^ ∃ pnum, hours<eid, pnum, hours> ∈ Works-On}