EXERCISE 8

Andrea Crotti (#299466), Can Liu (#286105), Sebastian Roidl (#281941)

15 Dicembre 2009

1. Query Languages (5 pt.)

1. Assume that no two columns of relations have the same name.

Show that a query in the normal form c(F (R1 x R2 x ... x Rn)) can be expressed in Domain Relational Calculus (DRC), where c is a vector of columns, F is a boolean formula built from conjunctions of atoms in the form ci = cj or ci = constant (ci and ci are columns).

2. What does "relational completeness" mean?

Show that SQL is relational complete by enumerating SQL constructs corresponding to selection, projection, cartesian product, union, and difference.

Give two examples of SQL constructs/semantics not expressible in relational algebra (RA).

A language is *relational complete* if it can express all possible queries expressible by RA.

If we can map the RA operations to SQL constructs than we have shown that SQL is in fact relational complete.

selection	where
projection	select
Join	from
union	union
difference	except

3. Suppose we have three tables VIP(id), Employee(id), and Male(id). Translate the following SQL query into relational calculus and relational algebra.

select Male.id from VIP, Employee, Male where VIP.id=Male.id or Employee.id=Male.id

Relational calculus

- Domain

$\{\text{id} \mid \exists id VIP(id) \land Employee(id) \land Male(id)\}$

- Tuple

m.id OF EACH m in Male:

SOME v in VIP (Some e in Employee (m.id=v.id $\lor m.id = e.id$)

Relational algebra

$\pi_{\text{Male.id}}(\sigma_{\text{Vip.id=Male.id}\lor\text{Employee.id=Mal.id}}(\text{VIP} \times \text{Employee} \times \text{Male})$

4. For the following database (VIP is empty), what is the result of the query above? What is the result of $\pi_\phi(\textbf{Employee})$ (Employee)?

The result of the above query gives nothing as result, because the VIP table doesn't contain any tuple and the join becomes also empty.

```
CREATE table VIP(id INTEGER, PRIMARY KEY(id));
CREATE table EMPLOYEE(id INTEGER, PRIMARY KEY(id));
CREATE table MALE(id INTEGER, PRIMARY KEY(id));

insert into MALE VALUES (1);
insert into EMPLOYEE VALUES (1);
insert into EMPLOYEE VALUES (2);

select MALE.id
from EMPLOYEE, MALE, VIP
where VIP.id=MALE.id or EMPLOYEE.id=Male.id;
```

π_ϕ(**Employee**) (**Employee**)

5. Figure 1 shows the flow of a query through a DBMS, in which different forms are used to represent a query at different stages. Fill in the three blanks with the corresponding query languages (i.e., SQL, RC, RA).

2. Query Formulation (15 pt.)

Formulate the following queries as expressions in relational algebra, tuple relational calculus, domain relational calculus and SQL:

a) Find name and city of all persons who work for the company 'MyComp' and earn less than 10000.

Relational Algebra:

$\pi_{\text{person\_name,city}} \left(\sigma_{\text{company\_name}='MyComp'\wedge\text{salary}<10000} \left(\text{lives works} \right) \right.$

Domain Calculuis:

$\{[\text{p,c}] \mid \exists st([p, c, st] \in lives \wedge \exists f, g([p, f, g] \in works \wedge g < 10000))\}$

b) Find the names of all persons, who don't work for 'MyComp' (or do not work at all).

Relational Algebra:

$\pi_{\text{person\_name}}(\text{lives})$ - $\pi_{\text{person\_name}}\left(\sigma_{\text{company\_name}='MyComp'}(\text{works})\right)$

Tuple Calculus:

$\{\, \text{p.person\_name} \mid \text{p} \in lives \wedge \exists y \in works(p.person\_name = y.person\_name \wedge y.company\_name! = \text{'}MyComp'\text{)}\}$

Domain Calculus:

$\{\text{name} \mid \text{lives(name, \_ , }_) \wedge works(name, comp,_) \wedge comp! = \text{'}MyComp'\}$

c) Find the names of all persons, who live in a city that the company they are working for is not located in.

Relational Algebra:

$\pi_{\text{person\_name}}$ (lives works (π_{city} (lives) X $\pi_{\text{company\_name}}$(works) - located))

Tuple Calculus:

{ p.person_name | p $\in lives \land \exists x \in works \land \exists y \in located(p.person\_name = x.person\_name \land x.company\_name = y.company\_name \land y.city! = p.city)$}

Domain Calculus:

{ p | lives (p,c,st) $\land \exists works(p, f, g) \land \exists located(f, c_1) \land c! = c_1$

d) Find the names of all managers, whose company is not placed in Munich or Hamburg. e) Find the names of all companies that are located in exactly the same cities as 'My-Comp', assuming each company is located in some city.