EXERCISE 8

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

16 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 $\pi_c(\sigma_F$ (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.

RA	SQL
selection	where
projection	select
Join	from
union	union
difference	except

SQL is more expressive than relational algebra, for example these SQL operators are not found in relational algebra:

- ORDER BY

- GROUP BY
- UPDATE

  ⋮

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

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

- Tuple

m.id OF EACH m in Male: SOME v in VIP (Some e in Employee (m.id=v.id $\vee m.id = e.id$)

Relational algebra

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

4. For the following database (VIP is empty), what is the result of the query above? What is the result of π_ϕ(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).

STEP	LANG
1	SQL
2	RC
3	RA

2. What does "relational completeness" mean? Show that SQL is relational complete by enu- merating SQL constructs corresponding to selection, projection, cartesian product, union, and difference. Give two examples of SQL constructs/semantics not expressible in relational algebra (RA).

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

4. For the following database (VIP is empty), what is the result of the query above? What is the result of 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:

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

Tuple Calculus:

$\{$n.person_name,n.city | lives(n) $\wedge \exists y(works(y) \wedge n.person\_name = y.person\_name \wedge$
$y.salary < 10000 \wedge y.company\_name =' MyComp')\}$

Domain Calculuis:

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

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

Relational Algebra:

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

Tuple Calculus:

$\{$ p.person_name | lives(p) $\wedge \exists y \wedge works(y)(p.person\_name = y.person\_name \wedge$
$y.company\_name! = `MyComp')\}$

Domain Calculus:

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

c) Findthenamesofallpersons,wholiveinacitythatthecompanytheyareworkingforis not located in.

Relational Algebra:

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

Tuple Calculus:

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

Domain Calculus:

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

d) Find the names of all managers, whose company is not placed in Munich or Hamburg.

Relational Algebra:

$\pi_{\text{manager\_name}}$(boss)-($\pi_{\text{manager\_name}}$(boss $\bowtie \pi_{\text{manager\_name=person\_name}}$(works) $\bowtie$ ($\sigma_{\text{city='Munich'}}$ (located) U $\sigma_{\text{city='Hamburg'}}$(located))))

Tuple Calculus:

{m.manger_name | boss(m) $\wedge \exists x \wedge located(x) \exists y \wedge works(y)(m.manager\_name = y.person\_name \wedge x.city = y.city \wedge y.city! = `Hamburg' \wedge y.city! = `Munich')\}$

Domain Calculus:

{ m | $\exists p(boss(p, m) \backslash exists f, slocated(f, s) \exists g(works(m, f, g) \wedge s! = `Munich' \wedge s! = `Hamburg'))\}$

e) Find the names of all companies that are located in exactly the same cities as 'MyComp', assuming each company is located in some city.

Relational Algebra:

$\pi_{\text{company\_name}}\,(\sigma_{\text{C2='MyComp'}\wedge\text{city=city2}}(\text{located X p}_{\text{C2}\leftarrow\text{company\_name}}(\text{p}_{\{}city2 \leftarrow company\_name(located))))$

Tuple Calculus:

$\{\text{t(company\_name)} \mid \text{located (t)} \wedge \neq (\exists l2)(located(l2) \wedge (l2company\_name = tcompany\_name) \wedge \neq ((\exists myCL)(located(myCL) \wedge (myCLcompany\_name = `MyComp') \wedge \neq (\exists l_2)(\text{located(l}_2\text{)} \wedge (l_2company\_name = tcompany\_name) \wedge (l_2city = myCLcity)))\}$

Domain Calculus:

$\{\text{ f} \mid \exists s(located(f, s) \wedge located(myCompany, s) \wedge myComapny = `MyComp')\}$