The Australian National University, School of Computing
COMP2400/6240 (Relational Databases)
Semester 2, 2022

Lab 7, Week 9

RA, Query Processing and Optimization (Solutions)

The purpose of this lab is to help you understand relational algebra and the activities involved in processing SQL queries in relational databases. You will first practice on writing queries using relational algebra expressions, and translating SQL statements and relational algebra expressions from one into another. We will further do some exercises on: 1) how to represent a relational algebra expression using a query tree, 2:) how to apply heuristic rules to improve the efficiency of executing queries, and 3) how a relational algebra expression is evaluated over a given database state.

Recall that, the following operations are provided in relational algebra for expressing queries:

$\sigma_{\varphi}(R)$	Selection by condition φ
$\pi_{A_1,\dots,A_n}(R)$	Projection onto the set of attributes $\{A_1 \dots, A_n\}$
$\rho_{R'(A_1,\dots,A_n)}(R)$	Renaming the relation name to $R^{'}$ and attribute names to $A_1, \dots, A_n$
$\rho_{R'}(R)$	Renaming the relation name to $R^{'}$
$\rho_{(A_1,\dots,A_n)}(R)$	Renaming the attribute names to $A_1, \dots, A_n$
$R_1 \cup R_2$	Union of two relations R_1 and R_2
$R_1 \cap R_2$	Intersection of two relations R_1 and R_2
$R_1 - R_2$	Difference of two relations R_1 and R_2
$R_1 \times R_2$	Cartesian product of two relations R_1 and R_2
$R_1 \bowtie_{\varphi} R_2$	Join of two relations R_1 and R_2 with the join condition φ
$R_1 \bowtie R_2$	Natural join of two relations R_1 and R_2

We will be looking at queries in our exercises over the schema of the example database again.

- DEPARTMENT(dname, dnumber, mgrssn, mgrstartdate)
  with the primary key {dnumber};

- EMPLOYEE(fname, minit, lname, ssn, bdate, address, salary, superssn, dno) with the primary key {ssn};
- PROJECT(pname, pnumber, plocation, dnum) with the primary key {pnumber};
- WORKS_ON(ssn, pno, hours) with the primary key {ssn, pno};
- DEPT_LOCATION(dnumber, dlocation) with the primary key {dnumber, dlocation};

where the foreign keys are:

EMPLOYEE[dno]⊆DEPARTMENT[dnumber]

PROJECT[dnum]⊆DEPARTMENT[dnumber]

WORKS_ON[ssn]⊆EMPLOYEE[ssn]

WORKS_ON[pno]⊆PROJECT[pnumber]

DEPT_LOCATION[dnumber]⊆DEPARTMENT[dnumber]

Work in pairs to complete the following exercises.

1 Relational Algebra

Write the relational algebra expressions for the following queries.

(1) *List the last name and ssn of all employees with a salary of more than \$60,000 or less than \$10,000.*

Solution: $\pi_{lname,ssn}(\sigma_{salary>60000}(\text{EMPLOYEE}) \cup \sigma_{salary<10000}(\text{EMPLOYEE}))$

(2) *What is the name of the manager of the 'Information Technology' department?*

Solution: $\pi_{fname,lname}((\sigma_{dname=\text{'}Information\ Technology\text{'}}(\text{DEPARTMENT})) \bowtie_{mgrssn=ssn} \text{EMPLOYEE})$

(3) *List the first and last names of all employees who work at the 'Information Technology' department.*

Solution: $\pi_{fname,lname}((\sigma_{dname=\text{'}Information\ Technology'}(\text{DEPARTMENT})) \bowtie_{dnumber=dno} \text{EMPLOYEE})$

(4) *List the first and last names of all employees who have worked on the project 'Red tape is Fun'.*

Solution: $\pi_{fname,lname}(\sigma_{pname=\text{'Red tape is Fun'}}((\text{EMPLOYEE} \bowtie \text{WORKS\_ON}) \bowtie_{pno=pnumber} \text{PROJECT}))$

(5) *List the first and last names of all employees who work at the 'Information Technology' department but have* not *worked on the project 'Red tape is Fun'.*

Solution: $R_1 := \pi_{fname,lname,ssn}((\sigma_{dname=\text{'}Information\ Technology'}(\text{DEPARTMENT})) \bowtie_{dnumber=dno} \text{EMPLOYEE})$

$R_2 := \pi_{fname,lname,ssn}(\sigma_{pname=\text{'Red tape is Fun'}}((\text{EMPLOYEE} \bowtie \text{WORKS\_ON}) \bowtie_{pno=pnumber} \text{PROJECT}))$

$Result = \pi_{fname,lname}(R_1 - R_2)$.

(6) *List the first and last names of all employees who have* never *worked on any project.*

Solution:

$R_1 := \pi_{fname,lname,ssn}(\text{EMPLOYEE})$

$R_2 := \pi_{fname,lname,ssn}(\text{EMPLOYEE} \bowtie \text{WORKS\_ON})$

$Result = \pi_{fname,lname}(R_1 - R_2)$.

(7) *List the first and last names of all employees who have* only *worked on the project 'Red tape is Fun'.*

Solution:

$R_1 := \pi_{fname,lname,ssn}(\sigma_{pname=\text{'Red tape is Fun'}}((\text{EMPLOYEE} \bowtie \text{WORKS\_ON}) \bowtie_{pno=pnumber} \text{PROJECT}))$

$R_2 := \pi_{fname,lname,ssn}(\sigma_{pname\neq\text{'Red tape is Fun'}}((\text{EMPLOYEE} \bowtie \text{WORKS\_ON}) \bowtie_{pno=pnumber} \text{PROJECT}))$

$Result = \pi_{fname,lname}(R_1 - R_2)$.

Are the following two relational algebra expressions correct? If not, please explain why and suggest solutions to fix the problems.

(8) $\sigma_{pno=123}(\pi_{fname,lname,pno}(\text{EMPLOYEE} \bowtie \text{WORKS\_ON}) - \pi_{fname,pno,lname}(\sigma_{hours<200}(\text{EMPLOYEE} \bowtie \text{WORKS\_ON})))$

Solution:

It's not correct because it violates the type compatibility for the set operations. It can be fixed by swapping *lname* and *pno* as follows:

- $\sigma_{pno=123}(\pi_{fname,lname,pno}(\text{EMPLOYEE} \bowtie \text{WORKS\_ON}) - \pi_{fname,lname,pno}(\sigma_{hours<200}(\text{EMPLOYEE} \bowtie \text{WORKS\_ON})))$

(9) $\sigma_{hours>100}(\pi_{dname,fname,lname}(\text{EMPLOYEE} \bowtie \text{WORKS\_ON}))$

Solution:

It's not correct because (1) *hours* has been projected away before applying $\sigma_{hours>100}$; (2) *dname* is not an attribute of EMPLOYEE $\bowtie$ WORKS_ON. It can be fixed as follows:

- $\sigma_{hours>100}(\pi_{fname,lname,hours}(\text{EMPLOYEE} \bowtie \text{WORKS\_ON}))$

2 Equivalence between SQL and Relational Algebra

Translate the following SQL statements into relational algebra queries.

(10) `SELECT * FROM` EMPLOYEE, DEPARTMENT

Solution:

EMPLOYEE $\times$ DEPARTMENT

(11) `SELECT * FROM` EMPLOYEE, DEPARTMENT `WHERE` EMPLOYEE.*dno* = DEPARTMENT.*dnumber*

Solution:

EMPLOYEE $\bowtie_{dno=dnumber}$ DEPARTMENT or $\sigma_{dno=dnumber}(\text{EMPLOYEE} \times \text{DEPARTMENT})$

(12) $\pi_{dname,fname,lname}(\sigma_{ssn=mgrssn}(\text{EMPLOYEE} \times \text{DEPARTMENT}))$

Solution:

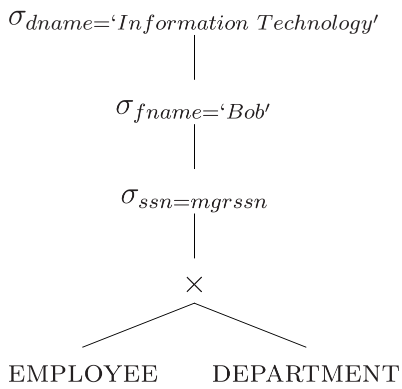
```
SELECT dname, fname, lname
  FROM EMPLOYEE, DEPARTMENT
 WHERE EMPLOYEE.ssn =DEPARTMENT.mgrssn
```

3 Query Trees

Draw the query trees for the following relational algebra expressions.

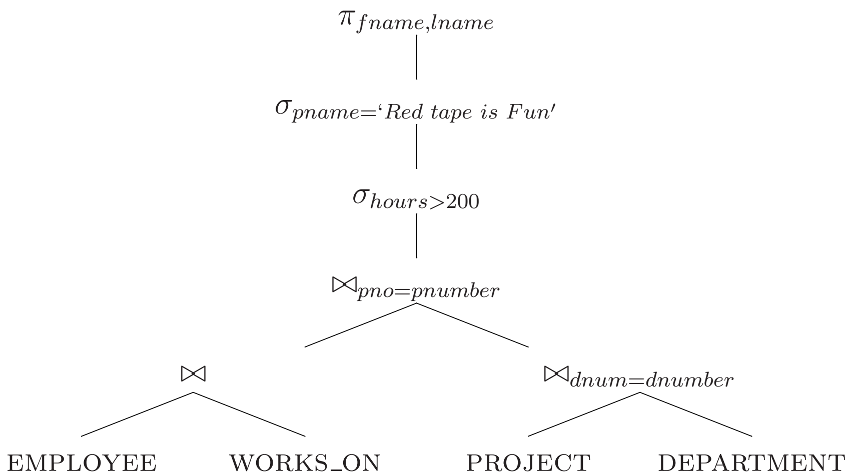
(13) $\sigma_{dname=\text{`}Information\ Technology'}(\sigma_{fname=\text{`}Bob'}(\sigma_{ssn=mgrssn}(\text{EMPLOYEE}\times\text{DEPARTMENT})))$

Solution:



(14) $\pi_{fname,lname}(\sigma_{pname=\text{`}Red\ tape\ is\ Fun'}(\sigma_{hours>200}((\text{EMPLOYEE} \bowtie \text{WORKS\_ON})$
$\bowtie_{pno=pnumber} (\text{PROJECT} \bowtie_{dnum=dnumber} \text{DEPARTMENT}))))$

Solution:



4 Query Optimization

Find two equivalent relational algebra expressions to each of the following relational algebra expressions, and discuss the difference of their performance. (**Hint:** apply algebraic heuristics rules).

(15) $\sigma_{dname=\text{`}Information\ Technology' \wedge fname=\text{`}John' \wedge ssn=mgrssn}(\text{EMPLOYEE} \times \text{DEPARTMENT})$

Solution:

- by applying $\sigma_{\varphi}(R_1 \times R_2) \equiv R_1 \bowtie_{\varphi} R_2$, we have:

  $\sigma_{dname=\text{`}Information\ Technology' \wedge fname=\text{`}John'}(\text{EMPLOYEE} \bowtie_{ssn=mgrssn} \text{DEPARTMENT})$.
- by applying $\sigma_{\varphi}(R_1 \bowtie R_2) \equiv R_1 \bowtie \sigma_{\varphi}(R_2)$, we have:

  $\sigma_{fname=\text{`}John'}(\text{EMPLOYEE} \bowtie_{ssn=mgrssn} \sigma_{dname=\text{`}Information\ Technology'}(\text{DEPARTMENT}))$.
- by applying $\sigma_{\varphi}(R_1 \bowtie R_2) \equiv R_1 \bowtie \sigma_{\varphi}(R_2)$ again, we have:

  $\sigma_{fname=\text{`}John'}(\text{EMPLOYEE}) \bowtie_{ssn=mgrssn} \sigma_{dname=\text{`}Information\ Technology'}(\text{DEPARTMENT})$.

(16) $\pi_{fname,lname}(\sigma_{pname=\text{`}Red\ tape\ is\ Fun'}(\sigma_{hours>200}((\text{EMPLOYEE} \bowtie \text{WORKS\_ON})$
$\bowtie_{pno=pnumber} (\text{PROJECT} \bowtie_{dnum=dnumber} \text{DEPARTMENT}))))$

Solution:

∗ You can remove DEPARTMENT as it will give you the same result (**Note**: this is often considered as semantic query optimization, not rule-based query optimization by applying algebraic heuristics rules):

  $\pi_{fname,lname}(\sigma_{pname=\text{`}Red\ tape\ is\ Fun'}(\sigma_{hours>200}((\text{EMPLOYEE} \bowtie \text{WORKS\_ON})$
  $\bowtie_{pno=pnumber} \text{PROJECT})))$

- push down selection $\sigma_{hours>200}$ (i.e., applying $\sigma_{\varphi}(R_1 \bowtie R_2) \equiv R_1 \bowtie \sigma_{\varphi}(R_2)$):

  $\pi_{fname,lname}(\sigma_{pname=\text{`}Red\ tape\ is\ Fun'}((\text{EMPLOYEE} \bowtie \sigma_{hours>200}(\text{WORKS\_ON}))$
  $\bowtie_{pno=pnumber} \text{PROJECT}))$
- push down selection $\sigma_{pname=\text{`}Red\ tape\ is\ Fun'}$ (i.e., applying $\sigma_{\varphi}(R_1 \bowtie R_2) \equiv R_1 \bowtie \sigma_{\varphi}(R_2)$):

  $\pi_{fname,lname}((\text{EMPLOYEE} \bowtie \sigma_{hours>200}(\text{WORKS\_ON}))$
  $\bowtie_{pno=pnumber} (\sigma_{pname=\text{`}Red\ tape\ is\ Fun'}(\text{PROJECT})))$fname
- push down projection (i.e., applying $\pi_X(R_1 \bowtie R_2) \equiv \pi_X(\pi_{X_1}(R_1) \bowtie \pi_{X_2}(R_2))$:

  $\pi_{fname,lname}(\pi_{fname,lname,pno}(\text{EMPLOYEE} \bowtie \sigma_{hours>200}(\text{WORKS\_ON}))$
  $\bowtie_{pno=pnumber} (\pi_{pnumber}(\sigma_{pname=\text{`}Red\ tape\ is\ Fun'}(\text{PROJECT}))))$

5 Evaluation of Relational Algebra Expressions

Evaluate the following relational algebra expressions over the database state provided in the lab in steps.

(17) WORK_ON_PROJECTS $:= \pi_{ssn,hours,pname}($WORKS_ON $\bowtie_{pno=pnumber}$ PROJECT$)$
MGR_WORKS_ON $:=$ DEPARTMENT $\bowtie_{mgrssn=ssn}$ WORKS_ON_PROJECTS
MGR_HOURS_WORKED $:= \pi_{mgrssn,pname,hours}(MGR_WORKS_ON)$
$(\pi_{fname,lname,ssn}($EMPLOYEE$)) \bowtie_{ssn=mgrssn}$ (MGR_HOURS_WORKED)

For example, we have the following result for WORKS_ON $\bowtie_{pno=pnumber}$ PROJECT:

WORKS_ON $\bowtie_{pno=pnumber}$ PROJECT						
ssn	pno	hours	pname	pnumber	plocation	dnum
20765	9000	100	'Difference Engine'	9000	'Canberra'	1000
20765	9001	500	'Red tape is Fun'	9001	'Canberra'	1001
20864	9002	50	'Object Oriented COBOL'	9002	'Sydney'	1000
20915	9000	250	'Difference Engine'	9000	'Canberra'	1000

(a) $\pi_{ssn,hours,pname}($WORKS_ON $\bowtie_{pno=pnumber}$ PROJECT$)$

Solution:

$\pi_{ssn,hours,pname}($WORKS_ON $\bowtie_{pno=pnumber}$ PROJECT$)$		
ssn	hours	pname
20765	100	'Difference Engine'
20765	500	'Red tape is Fun'
20864	50	'Object Oriented COBOL'
20915	250	'Difference Engine'

(b) DEPARTMENT $\bowtie_{mgrssn=ssn}$ WORKS_ON_PROJECTS

Solution:

DEPARTMENT $\bowtie_{mgrssn=ssn}$ WORKS_ON_PROJECTS						
dname	dnumber	mgrssn	mgrstartdate	ssn	hours	pname
'Information Technology'	1000	20765	'01/01/07'	20765	100	'Difference Engine'
'Information Technology'	1000	20765	'01/01/07'	20765	500	'Red tape is Fun'
'Administration'	1001	20915	'29/02/04'	20915	250	'Difference Engine'

(c) $\pi_{mgrssn,pname,hours}(MGR_WORKS_ON)$

Solution:

DEPARTMENT $\bowtie_{mgrssn=ssn}$ WORKS_ON_PROJECTS		
mgrssn	pname	hours
20765	'Difference Engine'	100
20765	'Red tape is Fun'	500
20915	'Difference Engine'	250

(d) $\pi_{fname,lname,ssn}$(EMPLOYEE)

Solution:

$\pi_{fname,lname,ssn}$(EMPLOYEE)		
fname	lname	ssn
'Michio'	'Morishima'	20118
'John'	'Backus'	20766
'Gramsci'	'Antonio'	20876
'Ada'	'Lovelace'	21286
'Milton'	'Friedman'	29057
'Edsger'	'Dijkstra'	20765
'Grace'	'Hopper'	20864
'Frederick'	'Taylor'	20915
'John'	'Keynes'	21287

(e) ($\pi_{fname,lname,ssn}$(EMPLOYEE)) $\bowtie_{ssn=mgrssn}$ (MGR_HOURS_WORKED)

Solution:

($\pi_{fname,lname,ssn}$(EMPLOYEE)) $\bowtie_{ssn=mgrssn}$ (MGR_HOURS_WORKED)					
fname	lname	ssn	mgrssn	pname	hours
'Edsger'	'Dijkstra'	20765	20765	'Difference Engine'	100
'Edsger'	'Dijkstra'	20765	20765	'Red tape is Fun'	500
'Frederick'	'Taylor'	20915	20915	'Difference Engine'	250