2007 Spring Database Systems

Homework III 參考解答

1. Specify the following queries on the COMPANY relational database schema shown in Figure 5.5, using the relational operators discussed in this chapter. Also show the result of each query as it would apply to the database state of Figure 5.6.

(a) Retrieve the names of employees in department 5 who work more than 10 hours per week on the 'ProductX' project.

(b) List the names of employees who have a dependent with the same first name as themselves.

(c) Find the names of employees that are directly supervised by 'Franklin Wong'.

(d) For each project, list the project name and the total hours per week (by all employees) spent on that project.

(e) Retrieve the names of employees who work on every project.

(f) Retrieve the names of employees who do not work on any project.

(g) For each department, retrieve the department name, and the average salary of employees working in that department.

(h) Retrieve the average salary of all female employees.

(i) Find the names and addresses of employees who work on at least one project located in Houston but whose department has no location in Houston.

(j) List the last names of department managers who have no dependents.

Answers:

In the relational algebra, as in other languages, it is possible to specify the same query in multiple ways. We give one possible solution for each query.

(a) $EMP\_W\_X \leftarrow (\sigma_{PNAME='ProductX'}(PROJECT)) \bowtie_{PNUMBER=PNO} (WORKS\_ON)$

$EMP\_WORK\_10 \leftarrow (EMPLOYEE) \bowtie_{SSN=ESSN} (\sigma_{HOURS>10}(EMP\_W\_X))$

$RESULT \leftarrow \Pi_{LNAME,FNAME}(\sigma_{DNO=5}(EMP\_WORK\_10))$

Result:

LNAME	FNAME
Smith	John
English	Joyce

(b) $E \leftarrow (EMPLOYEE) \bowtie_{SSN=ESSN \text{ AND } FNAME=DEPENDENT\_NAME} (DEPENDENT)$

$R \leftarrow \pi_{LNAME,FNAME} (E)$

Result (empty):

LNAME	FNAME

(c) $WONG\_SSN \leftarrow \pi_{SSN} (\sigma_{FNAME='Franklin' \text{ AND } LNAME='Wong'} (EMPLOYEE))$

$WONG\_EMPS \leftarrow (EMPLOYEE) \bowtie_{SUPERSSN=SSN} (WONG\_SSN)$

$RESULT \leftarrow \pi_{LNAME,FNAME} (WONG\_EMPS)$

Result:

LNAME	FNAME
Smith	John
Narayan	Ramesh
English	Joyce

(d) $PROJ\_HOURS(PNO,TOT\_HRS) \leftarrow {}_{PNO}\Im_{SUM\ HOURS} (WORKS\_ON)$

$RESULT \leftarrow \pi_{PNAME,TOT\_HRS} ((PROJ\_HOURS) \bowtie_{PNO=PNUMBER} (PROJECT))$

Result:

PNAME	TOT_HRS
ProductX	52.5
ProductY	37.5
ProductZ	50.0
Computerization	55.0
Reorganization	25.0
Newbenefits	55.0

(e) $PROJ\_EMPS(PNO,SSN) \leftarrow \pi_{PNO,ESSN} (WORKS\_ON)$

$ALL\_PROJS(PNO) \leftarrow \pi_{PNUMBER} (PROJECT)$

$EMPS\_ALL\_PROJS \leftarrow PROJ\_EMPS \div ALLPROJS$

$RESULT \leftarrow \pi_{LNAME,FNAME} (EMPLOYEE * EMP\_ALL\_PROJS)$

Result (empty):

LNAME	FNAME

(f) $ALL\_EMPS \leftarrow \pi_{SSN} (EMPLOYEE)$

$WORKING\_EMPS(SSN) \leftarrow \pi_{ESSN} (WORKS\_ON)$

NON_WORKING_EMPS ← ALL_EMPS - WORKING_EMPS

RESULT ← $\pi_{LNAME,FNAME}$ (EMPLOYEE * NON_WORKING_EMPS)

Result (empty):

LNAME	FNAME

(g) DEPT_AVG_SALS(DNUMBER,AVG_SAL) ← $_{DNO}\Im_{AVG\ SALARY}$ (EMPLOYEE)

RESULT ← π_{DNAME,AVG_SAL} (DEPT_AVG_SALS * DEPARTMENT)

Result:

DNAME	AVG_SAL
Research	33250
Administration	31000
Headquarters	55000

(h) RESULT(AVG_F_SAL) ← $\Im_{AVG\ SALARY}$ ($\sigma_{SEX='F'}$ (EMPLOYEE))

Result:

AVG_F_SAL
31000

(i) E_P_HOU(SSN) ←

π_{ESSN} (WORKS_ON $\bowtie_{PNO=PNUMBER}$ ($\sigma_{PLOCATION='Houston'}$ (PROJECT)))

D_NO_HOU ←

$\pi_{DNUMBER}$ (DEPARTMENT) - $\pi_{DNUMBER}$ ($\sigma_{DLOCATION='Houston'}$ (DEPARTMENT))

E_D_NO_HOU ← π_{SSN} (EMPLOYEE $\bowtie_{PNO=DNUMBER}$ (D_NO_HOU))

RESULT_EMPS ← E_P_HOU - E_D_NO_HOU

RESULT ← $\pi_{LNAME,FNAME,ADDRESS}$ (EMPLOYEE * RESULT_EMPS)

Result:

LNAME	FNAME	ADDRESS
Wallace	Jennifer	291 Berry, Bellaire, TX

(j) DEPT_MANAGERS(SSN) ← π_{MGRSSN} (DEPARTMENT)

EMPS_WITH_DEPENDENTS(SSN) ← π_{ESSN} (DEPENDENT)

RESULT_EMPS ← DEPT_MANAGERS - EMPS_WITH_DEPENDENTS

RESULT ← $\Pi_{\text{LNAME,FNAME}}$ (EMPLOYEE * RESULT_EMPS)

Result:

LNAME	FNAME
Borg	James

2.

(a) R1 ←STUDENT $\bowtie_{\text{S.SSN=E.SSN}}$ ENROLL

R2 ← $\Pi_{\text{COURSE\#}}$ $\sigma_{\text{NAME='JOHN SMITH' AND QUARTER='W99'}}$(R1))

Answer: R2

(b) R1← COURSE $\bowtie_{\text{C.COURSE\#=B.COURSE\#}}$ BOOK_ADOPTION

R2(COURSE#, BOOK_NO) ← $_{\text{COURSE\#}}\Im_{\text{COUNT BOOK\_ISBN}}$ (R1)

R3←$\sigma_{\text{CNAME='CS' AND BOOK\_NO>2}}$ (COURSE$\bowtie_{\text{C.COURSE\#=R2.COURSE\#}}$ R2)

R4← $\Pi_{\text{COURSE\#, BOOK\_ISBN}}$ (R3$\bowtie_{\text{R3.COURSE\#=B.COURSE\#}}$ BOOK_ADOPTION)

R5← $\Pi_{\text{COURSE\#, BOOK\_ISBN, BOOK\_TITLE}}$ (R4$\bowtie_{\text{R4.BOOK_ISBN=T.BOOK_ISBN}}$ TEXT)

Answer: R5

(c) R1← $\Pi_{\text{COURSE\#, PUBLISHER}}$ (BOOK_ADOPTION$\bowtie_{\text{B.BOOK_ISBN=T.BOOK_ISBN}}$ TEXT)

R2←$\sigma_{\text{COUNT\_PUB='1'}}$ ($_{\text{COURSE\#}}\Im_{\text{COUNT PUBLISHER}}$ (R1))

R3←$\sigma_{\text{PUBLISHER='AWL Publishing'}}$ (R2$\bowtie_{\text{R2.COURSE\#=R1.COURSE\#}}$ R1)

R4← Π_{CNAME} (COURSE$\bowtie_{\text{C.COURSE\#=R3.COURSE\#}}$ R3)

Answer: R4

3.

Answer:

(a) P Q R A B C
10 a 5 10 b 6
10 a 5 10 b 5
25 a 6 25 c 3

(b) P Q R A B C
15 b 8 10 b 6
15 b 8 10 b 5

(c) P Q R A B C
10 a 5 10 b 6
10 a 5 10 b 5
15 b 8 null null null
25 a 6 25 c 3

(d) P Q R A B C
15 b 8 10 b 6
null null null 25 c 3
15 b 8 10 b 5

(e) P Q R
10a 5
15 b 8
25 a 6
10b 6
25 c 3
10b 5

(f) P Q R A B C
10 a 5 10 b 5

4. Specify queries (a), (b), (c), (e), (f), (i), and (j) of Question 1 in both the tuple relational calculus and the domain relational calculus.

Answer:

(a) Retrieve the names of employees in department 5 who work more than 10 hours per week on the 'ProductX' project.

Tuple relational Calculus:

{ e.LNAME, e.FNAME | EMPLOYEE(e) AND e.DNO=5 AND (∃ p) (∃ w) (WORKS_ON(w) AND PROJECT(p) AND e.SSN=w.ESSN AND w.PNO=p.PNUMBER AND p.PNAME='ProductX' AND w.HOURS>10) }

Domain relational Calculus:

{ qs | EMPLOYEE(qrstuvwxyz) AND z=5 AND (∃ a) (∃ b) (∃ e)

(∃ f) (∃ g) (WORKS_ON(efg) AND PROJECT(abcd) AND t=e AND f=b AND a='ProductX' AND g>10) }

(b) List the names of employees who have a dependent with the same first name as themselves.

Tuple relational Calculus:

{ e.LNAME, e.FNAME | EMPLOYEE(e) AND (∃ d) (DEPENDENT(d) AND e.SSN=d.ESSN AND e.FNAME=d.DEPENDENT_NAME) }

Domain relational Calculus:

{ qs | (∃ t) (∃ a) (∃ b) (EMPLOYEE(qrstuvwxyz) AND DEPENDENT(abcde) AND a=t AND b=q) }

(c) Find the names of employees that are directly supervised by 'Franklin Wong'.

Tuple relational Calculus:

{ e.LNAME, e.FNAME | EMPLOYEE(e) AND (∃ s) (EMPLOYEE(s) AND s.FNAME='Franklin' AND s.LNAME='Wong' AND e.SUPERSSN=s.SSN) }

Domain relational Calculus:

{ qs | (∃ y) (∃ a) (∃ c) (∃ d) (EMPLOYEE(qrstuvwxyz) AND EMPLOYEE(abcdefghij) AND a='Franklin' AND c='Wong' AND y=d) }

(e) Retrieve the names of employees who work on every project.

Tuple relational Calculus:

{ e.LNAME, e.FNAME | EMPLOYEE(e) AND (FORALL p) (NOT(PROJECT(p)) OR (∃ w) (WORKS_ON(w) AND p.PNUMBER=w.PNO AND w.ESSN=e.SSN)) }

Domain relational Calculus:

{ qs | (∃ t) (EMPLOYEE(qrstuvwxyz) AND (FORALL b) (NOT(PROJECT(abcd)) OR (∃ e) (∃ f) (WORKS_ON(efg) AND e=t AND f=b)) }

(f) Retrieve the names of employees who do not work on any project.

Tuple relational Calculus:

{ e.LNAME, e.FNAME | EMPLOYEE(e) AND NOT(∃ w) (WORKS_ON(w) AND w.ESSN=e.SSN) }

Domain relational Calculus:

{ qs | (∃ t) (EMPLOYEE(qrstuvwxyz) AND NOT(∃ a) (WORKS_ON(abc) AND a=t)) }

(i) Find the names and addresses of employees who work on at least one project located in Houston but whose department has no location in Houston.

Tuple relational Calculus:

{ e.LNAME, e.FNAME, e.ADDRESS | EMPLOYEE(e) AND (∃ p) (∃ w) (WORKS_ON(w) AND PROJECT(p) AND e.SSN=w.ESSN AND w.PNO=p.PNUMBER AND p.PLOCATION='Houston' AND NOT(∃ l) (DEPT_LOCATIONS(l) AND e.DNO=l.DNUMBER AND l.DLOCATION='Houston')) }

Domain relational Calculus:

{ qsv | (∃ t) (∃ z) (EMPLOYEE(qrstuvwxyz) AND (∃ b) (∃ c) (∃ e) (∃ f) (WORKS_ON(efg) AND PROJECT(abcd) AND t=e AND f=b AND c='Houston' AND NOT(∃ h) NOT∃ i) (DEPT_LOCATIONS(hi) AND z=h AND i='Houston')) }

(j) List the last names of department managers who have no dependents.

Tuple relational Calculus:

{ e.LNAME | EMPLOYEE(e) AND (∃ d) (DEPARTMENT(d) AND e.SSN=d.MGRSSN AND NOT(∃ x) (DEPENDENT(x) AND e.SSN=x.ESSN)) }

Domain relational Calculus:

{ s | (∃ t) (EMPLOYEE(qrstuvwxyz) AND (∃ c) (DEPARTMENT(abcd) AND t=c AND NOT(∃ e) (DEPENDENT(efghi) AND e=t)) }