CHAPTER 6

Formal Relational Query Languages

In this chapter we study three additional formal relational languages. Relational Algebra, tuple relational calculus and domain relational calculus.

Of these three formal languages, we suggest placing an emphasis on relational algebra, which is used extensively in the chapters on query processing and optimization, as well as in several other chapters. The relational calculi generally do not merit as much emphasis.

Our notation for the tuple relational calculus makes it easy to present the concept of a safe query. The concept of safety for the domain relational calculus, though identical to that for the tuple calculus, is much more cumbersome notationally and requires careful presentation. This consideration may suggest placing somewhat less emphasis on the domain calculus for classes not focusing on database theory.

Exercises

6.10 Write the following queries in relational algebra, using the university schema.

a. Find the names of all students who have taken at least one Comp. Sci. course.

b. Find the IDs and names of all students who have not taken any course offering before Spring 2009.

c. For each department, find the maximum salary of instructors in that department. You may assume that every department has at least one instructor.

d. Find the lowest, across all departments, of the per-department maximum salary computed by the preceding query.

Answer:

employee (*person_name*, *street*, *city*)
works (*person_name*, *company_name*, *salary*)
company (*company_name*, *city*)
manages (*person_name*, *manager_name*)

Figure 6.22 Relational database for Exercises 6.2, 6.8, 6.11, 6.13, and 6.15

a. $\Pi_{name}\ (student \bowtie takes \bowtie \Pi_{course\_id}(\sigma_{dept\_name\ =\ 'Comp.Sci.'}(course)))$
Note that if we join *student*, *takes*, and *course*, only students from the Comp. Sci. department would be present in the result; students from other departments would be eliminated even if they had taken a Comp. Sci. course since the attribute *dept_name* appears in both *student* and *course*.

b. $\Pi_{ID,name}\ (student) - \Pi_{ID,name}\ (\sigma_{year<2009}(student \bowtie takes)$ Note that Spring is the first semester of the year, so we do not need to perform a comparison on *semester*.

c. ${}_{dept\_name}\mathcal{G}_{\textbf{max}(salary)}(instructor)$

d. $\mathcal{G}_{\textbf{min}(maxsal)}({}_{dept\_name}\mathcal{G}_{\textbf{max}(salary)}\ \textbf{as}\ maxsal(instructor))$

6.11 Consider the relational database of Figure 6.22, where the primary keys are underlined. Give an expression in the relational algebra to express each of the following queries:

a. Find the names of all employees who work for "First Bank Corporation".

b. Find the names and cities of residence of all employees who work for "First Bank Corporation".

c. Find the names, street addresses, and cities of residence of all employees who work for "First Bank Corporation" and earn more than $10,000.

d. Find the names of all employees in this database who live in the same city as the company for which they work.

e. Assume the companies may be located in several cities. Find all companies located in every city in which "Small Bank Corporation" is located.

Answer:

a. $\Pi_{person\_name}\ (\sigma_{company\_name\ =\ \text{"First Bank Corporation"}}\ (works))$

b. $\Pi_{person\_name,\ city}\ (employee \bowtie$
$(\sigma_{company\_name\ =\ \text{"First Bank Corporation"}}\ (works)))$

c. $\Pi_{person\_name,\ street,\ city}$
$(\sigma_{(company\_name\ =\ \text{“First Bank Corporation”}\ \wedge\ salary\ >\ 10000)}$
$works \bowtie employee)$

d. $\Pi_{person\_name}\ (employee \bowtie works \bowtie company)$

e. Note: Small Bank Corporation will be included in each answer.
$\Pi_{company\_name}\ (company \div$
$(\Pi_{city}\ (\sigma_{company\_name\ =\ \text{“Small Bank Corporation”}}\ (company))))$

6.12 Using the university example, write relational-algebra queries to find the course sections taught by more than one instructor in the following ways:

a. Using an aggregate function.

b. Without using any aggregate functions.

Answer:

a. $\sigma_{instrcnt>1}\ (_{course\_id,section\_id,year,semester}\mathcal{G}_{\textbf{count}(*)\ \textbf{as}\ instrcnt}(teaches))$

b. $\Pi_{course\_id,section\_id,year,semester}(\sigma_{ID<>ID2}(takes \bowtie$
$\rho_{takes1(ID2,course\_id,section\_id,year,semester)}(takes)))$

6.13 Consider the relational database of Figure 6.22. Give a relational-algebra expression for each of the following queries:

a. Find the company with the most employees.

b. Find the company with the smallest payroll.

c. Find those companies whose employees earn a higher salary, on average, than the average salary at First Bank Corporation.

Answer:

a. $t_1 \leftarrow\ _{company\_name}\mathcal{G}_{\textbf{count-distinct}(person\_name)}(works)$
$t_2 \leftarrow\ \mathcal{G}_{\textbf{max}(num\_employees)}(\rho_{company\_strength(company\_name,num\_employees)}(t_1))$
$\Pi_{company\_name}(\rho_{t_3(company\_name,num\_employees)}(t_1) \bowtie \rho_{t_4(num\_employees)}(t_2))$

b. $t_1 \leftarrow\ _{company\_name}\mathcal{G}_{\textbf{sum}(salary)}(works)$
$t_2 \leftarrow\ \mathcal{G}_{\textbf{min}(payroll)}(\rho_{company\_payroll(company\_name,payroll)}(t_1))$
$\Pi_{company\_name}(\rho_{t_3(company\_name,payroll)}(t_1) \bowtie \rho_{t_4(payroll)}(t_2))$

c. $t_1 \leftarrow\ _{company\_name}\mathcal{G}_{\textbf{avg}(salary)}(works)$
$t_2 \leftarrow\ \sigma_{company\_name\ =\ \text{“First Bank Corporation”}}(t_1)$
$\Pi_{t_3.company\_name}((\rho_{t_3(company\_name,avg\_salary)}(t_1))$
$\bowtie_{t_3.avg\_salary\ >\ first\_bank.avg\_salary}\ (\rho_{first\_bank(company\_name,avg\_salary)}(t_2)))$

6.14 Consider the following relational schema for a library:

member(*memb_no*, *name*, *dob*)
books(*isbn*, *title*, *authors*, *publisher*)
borrowed(*memb_no*, *isbn*, *date*)

Write the following queries in relational algebra.

a. Find the names of members who have borrowed any book published by "McGraw-Hill".

b. Find the name of members who have borrowed all books published by "McGraw-Hill".

c. Find the name and membership number of members who have borrowed more than five different books published by "McGraw-Hill".

d. For each publisher, find the name and membership number of members who have borrowed more than five books of that publisher.

e. Find the average number of books borrowed per member. Take into account that if an member does not borrow any books, then that member does not appear in the *borrowed* relation at all.

Answer:

a. $t_1 \leftarrow \Pi_{isbn}(\sigma_{publisher="McGraw-Hill"}(books))$
$\Pi_{name}((member \bowtie borrowed) \bowtie t_1))$

b. $t_1 \leftarrow \Pi_{isbn}(\sigma_{publisher="McGraw-Hill"}(books))$
$\Pi_{name,isbn}(member \bowtie borrowed) \div t_1$

c. $t_1 \leftarrow member \bowtie borrowed \bowtie (\sigma_{publisher="McGraw-Hill"}(books))$
$\Pi_{name}\ (\sigma_{countisbn > 5}\ ((\ memb\_no\ G_{\textbf{count-distinct}(isbn)\ \textbf{as}\ countisbn}(t_1))))$

d. $t_1 \leftarrow member \bowtie borrowed \bowtie books$
$\Pi_{publisher,name}\ (\sigma_{countisbn > 5}\ ((\ publisher,memb\_no\ G_{\textbf{count-distinct}(isbn)\ \textbf{as}\ countisbn}(t_1)))$

6.15 Consider the employee database of Figure 6.22. Give expressions in tuple relational calculus and domain relational calculus for each of the following queries:

a. Find the names of all employees who work for "First Bank Corporation".

b. Find the names and cities of residence of all employees who work for "First Bank Corporation".

c. Find the names, street addresses, and cities of residence of all employees who work for "First Bank Corporation" and earn more than $10,000.

d. Find all employees who live in the same city as that in which the company for which they work is located.

e. Find all employees who live in the same city and on the same street as their managers.

f. Find all employees in the database who do not work for "First Bank Corporation".

g. Find all employees who earn more than every employee of "Small Bank Corporation".

h. Assume that the companies may be located in several cities. Find all companies located in every city in which "Small Bank Corporation" is located.

Answer:

a. Find the names of all employees who work for First Bank Corporation:

i. $\{t \mid \exists s \in works\ (t[person\_name] = s[person\_name] \wedge s[company\_name] = \text{"First Bank Corporation"})\}$

ii. $\{< p > \mid \exists c, s\ (< p, c, s > \in works \wedge c = \text{"First Bank Corporation"})\}$

b. Find the names and cities of residence of all employees who work for First Bank Corporation:

i. $\{t \mid \exists r \in employee\ \exists s \in works\ (t[person\_name] = r[person\_name] \wedge t[city] = r[city] \wedge r[person\_name] = s[person\_name] \wedge s[company\_name] = \text{'First Bank Corporation'})\}$

ii. $\{< p, c > \mid \exists co, sa, st\ (< p, co, sa > \in works \wedge < p, st, c > \in employee \wedge co = \text{"First Bank Corporation"})\}$

c. Find the names, street address, and cities of residence of all employees who work for First Bank Corporation and earn more than $10,000 per annum:

i. $\{t \mid t \in employee \wedge (\exists s \in works\ (s[person\_name] = t[person\_name] \wedge s[company\_name] = \text{"First Bank Corporation"} \wedge s[salary] > 10000))\}$

ii. $\{< p, s, c > \mid < p, s, c > \in employee \wedge \exists co, sa\ (< p, co, sa > \in works \wedge co = \text{'First Bank Corporation'} \wedge sa > 10000)\}$

d. Find the names of all employees in this database who live in the same city as the company for which they work:

i. $\{t \mid \exists e \in employee \exists w \in works \exists c \in company$
$(t[person\_name] = e[person\_name]$
$\wedge e[person\_name] = w[person\_name]$
$\wedge w[company\_name] = c[company\_name] \wedge e[city] = c[city])\}$

ii. $\{< p > \mid \exists st, c, co, sa\ (< p, st, c > \in employee$
$\wedge < p, co, sa > \in works \wedge < co, c > \in company)\}$

e. Find the names of all employees who live in the same city and on the same street as do their managers:

i. $\{t \mid \exists l \in employee \exists m \in manages \exists r \in employee$
$(l[person\_name] = m[person\_name] \wedge m[manager\_name] = r[person\_name]$
$\wedge l[street] = r[street] \wedge l[city] = r[city] \wedge t[person\_name] = l[person\_name])\}$

ii. $\{< t > \mid \exists s, c, m\ (< t, s, c > \in employee \wedge < t, m > \in manages \wedge < m, s, c > \in employee)\}$

f. Find the names of all employees in this database who do not work for First Bank Corporation:
If one allows people to appear in the database (e.g. in *employee*) but not appear in *works,* the problem is more complicated. We give solutions for this more realistic case later.

i. $\{t \mid \exists w \in works\ (w[company\_name] \neq$ "First Bank Corporation"
$\wedge t[person\_name] = w[person\_name])\}$

ii. $\{< p > \mid \exists c, s\ (< p, c, s > \in works \wedge c \neq$ "First Bank Corporation"$)\}$

If people may not work for any company:

i. $\{t \mid \exists e \in employee\ (t[person\_name] = e[person\_name] \wedge \neg \exists w \in works$
$(w[company\_name] =$ "First Bank Corporation"
$\wedge w[person\_name] = t[person\_name]))\}$

ii. $\{< p > \mid \exists s, c\ (< p, s, c > \in employee) \wedge \neg \exists x, y$
$(y =$ "First Bank Corporation"$\wedge < p, y, x > \in works)\}$

g. Find the names of all employees who earn more than every employee of Small Bank Corporation:

i. $\{t \mid \exists w \in works\ (t[person\_name] = w[person\_name] \wedge \forall s \in works$
$(s[company\_name] =$ "Small Bank Corporation" $\Rightarrow w[salary] > s[salary]))\}$

ii. $\{< p > \mid \exists\, c,\ s\ (< p, c, s > \in\ works\ \wedge\ \forall\ p_2,\ c_2,\ s_2$
$(< p_2, c_2, s_2 > \notin\ works \vee c_2 \neq$ "Small Bank Corporation" $\vee$
$s\ >\ s_2))\}$

h. Assume the companies may be located in several cities. Find all companies located in every city in which Small Bank Corporation is located.
Note: Small Bank Corporation will be included in each answer.

i. $\{t \mid \forall s \in company\,(s[company\_name] =$ "Small Bank Corporation" $\Rightarrow$
$\exists\, r \in company\,(t[company\_name] = r[company\_name] \wedge$
$r[city] = s[city]))\}$

ii. $\{< co > \mid \forall\, co_2,\ ci_2\ (< co_2, ci_2 > \notin\ company$
$\vee co_2 \neq$ "Small Bank Corporation" $\vee\ < co, ci_2 > \in company)\}$

6.16 Let $R = (A,\ B)$ and $S = (A,\ C)$, and let $r(R)$ and $s(S)$ be relations. Write relational-algebra expressions equivalent to the following domain-relational-calculus expressions:

a. $\{< a > \mid \exists\, b\ (< a, b > \in\ r\ \wedge\ b = 17)\}$

b. $\{< a, b, c > \mid\ < a, b > \in\ r \wedge\ < a, c > \in\ s\}$

c. $\{< a > \mid \exists\, b\ (< a, b > \in\ r) \vee \forall\, c\ (\exists\, d\ (< d, c > \in\ s) \Rightarrow < a, c > \in\ s)\}$

d. $\{< a > \mid \exists\, c\ (< a, c > \in\ s\ \wedge\ \exists\, b_1, b_2\ (< a, b_1 > \in\ r\ \wedge\ < c, b_2 >$
$\in\ r\ \wedge\ b_1\ >\ b_2))\}$

Answer:

a. $\Pi_A\,(\sigma_{B\,=\,17}\,(r))$

b. $r\ \bowtie\ s$

c. $\Pi_A(r)\ \cup\ (r \div \sigma_B(\Pi_C(s)))$

d. $\Pi_{r.A}\,((r\ \bowtie\ s)\ \bowtie_{c\,=\,r2.A\,\wedge\,r.B\,>\,r2.B}\ (\rho_{r_2}(r)))$
It is interesting to note that (*d*) is an abstraction of the notorious query "Find all employees who earn more than their manager." Let $R = (emp,\ sal)$, $S = (emp,\ mgr)$ to observe this.

6.17 Repeat Exercise 6.16, writing SQL queries instead of relational-algebra expressions.
Answer:

a. **select** a
from r
where $b = 17$

b. **select** a, b, c
from r, s
where $r.a = s.a$

c. (**select** a
from r)
union
(**select** a
from s)

d. **select** a
from r **as** $r1, r$ **as** $r2, s$
where $r1.a = s.a$ **and** $r2.a = s.c$ **and** $r1.b > r2.b$

6.18 Let $R = (A, B)$ and $S = (A, C)$, and let $r(R)$ and $s(S)$ be relations. Using the special constant *null*, write tuple-relational-calculus expressions equivalent to each of the following:

a. $r \ltimes s$

b. $r \rtimes s$

c. $r \rtimes\!\ltimes s$

Answer:

a. $\{t \mid \exists r \in R\, \exists s \in S\, (r[A] = s[A] \wedge t[A] = r[A] \wedge t[B] = r[B] \wedge t[C] = s[C]) \vee$
$\exists s \in S(\neg \exists r \in R(r[A] = s[A]) \wedge t[A] = s[A] \wedge t[C] = s[C] \wedge t[B] = null)\}$

b. $\{t \mid \exists r \in R\, \exists s \in S\, (r[A] = s[A] \wedge t[A] = r[A] \wedge t[B] = r[B] \wedge t[C] = s[C]) \vee$
$\exists r \in R(\neg \exists s \in S(r[A] = s[A]) \wedge t[A] = r[A] \wedge t[B] = r[B] \wedge t[C] = null) \vee$
$\exists s \in S(\neg \exists r \in R(r[A] = s[A]) \wedge t[A] = s[A] \wedge t[C] = s[C] \wedge t[B] = null)\}$

c. $\{t \mid \exists r \in R\, \exists s \in S\, (r[A] = s[A] \wedge t[A] = r[A] \wedge t[B] = r[B] \wedge t[C] = s[C]) \vee$
$\exists r \in R(\neg \exists s \in S(r[A] = s[A]) \wedge t[A] = r[A] \wedge t[B] = r[B] \wedge t[C] = null)\}$

6.19 Give a tuple-relational-calculus expression to find the maximum value in relation $r(A)$.
Answer: $\{< a > \mid < a > \in r \wedge \forall < b > \in R\, a >= b\}$