type) compatible; these include UNION, INTERSECTION, an
CARTESIAN PRODUCT operation is a set operation that ca
s from two relations, producing all possible combinations. I
; however, we showed how CARTESIAN PRODUCT followe
o define matching tuples from two relations and leads to th
ent JOIN operations called THETA JOIN, EQUIJOIN, an
oduced. Query trees were introduced as a graphical represen
ra queries, which can also be used as the basis for interna
DBMS can use to represent a query.

ortant types of queries that *cannot* be stated with the basi
tions but are important for practical situations. We intro
ROJECTION to use functions of attributes in the projection
E FUNCTION operation to deal with aggregate types of sta
nmarize the information in the tables. We discussed recur
here is no direct support in the algebra but which can be
p approach, as we demonstrated. Then we presented the
R UNION operations, which extend JOIN and UNION and
source relations to be preserved in the result.

cribed the basic concepts behind relational calculus, which
of mathematical logic called predicate calculus. There are
calculi: (1) the tuple relational calculus, which uses tuple
tuples (rows) of relations, and (2) the domain relational
main variables that range over domains (columns of rela
ulus, a query is specified in a single declarative statement
rder or method for retrieving the query result. Hence, rela
onsidered to be a higher-level *declarative* language than the
se a relational calculus expression states *what* we want to
the query may be executed.

of relational calculus queries using both tuple and domain
query graphs as an internal representation for queries in
so discussed the existential quantifier (∃) and the universal
at relational calculus variables are bound by these quanti
tail how queries with universal quantification are written,
blem of specifying safe queries whose results are finite. We
ansforming universal into existential quantifiers, and vice
that give expressive power to the relational calculus, mak
asic relational algebra. There is no analog to grouping and
basic relational calculus, although some extensions have

ns

s of relational algebra and the purpose of each.

6.2. What is union compatibility? Why do the UNION, INTERSECTION, and DIFFERENCE operations require that the relations on which they are applied be union compatible?

6.3. Discuss some types of queries for which renaming of attributes is necessary in order to specify the query unambiguously.

6.4. Discuss the various types of *inner join* operations. Why is theta join required?

6.5. What role does the concept of *foreign key* play when specifying the most common types of meaningful join operations?

6.6. What is the FUNCTION operation? What is it used for?

6.7. How are the OUTER JOIN operations different from the INNER JOIN operations? How is the OUTER UNION operation different from UNION?

6.8. In what sense does relational calculus differ from relational algebra, and in what sense are they similar?

6.9. How does tuple relational calculus differ from domain relational calculus?

6.10. Discuss the meanings of the existential quantifier (∃) and the universal quantifier (∀).

6.11. Define the following terms with respect to the tuple calculus: *tuple variable, range relation, atom, formula,* and *expression.*

6.12. Define the following terms with respect to the domain calculus: *domain variable, range relation, atom, formula,* and *expression.*

6.13. What is meant by a *safe expression* in relational calculus?

6.14. When is a query language called relationally complete?

xercises

6.15. Show the result of each of the sample queries in Section 6.5 as it would apply to the database state in Figure 3.6.

6.16. Specify the following queries on the COMPANYrelational 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 in Figure 3.6.

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

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

c. Find the names of all employees who 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

—

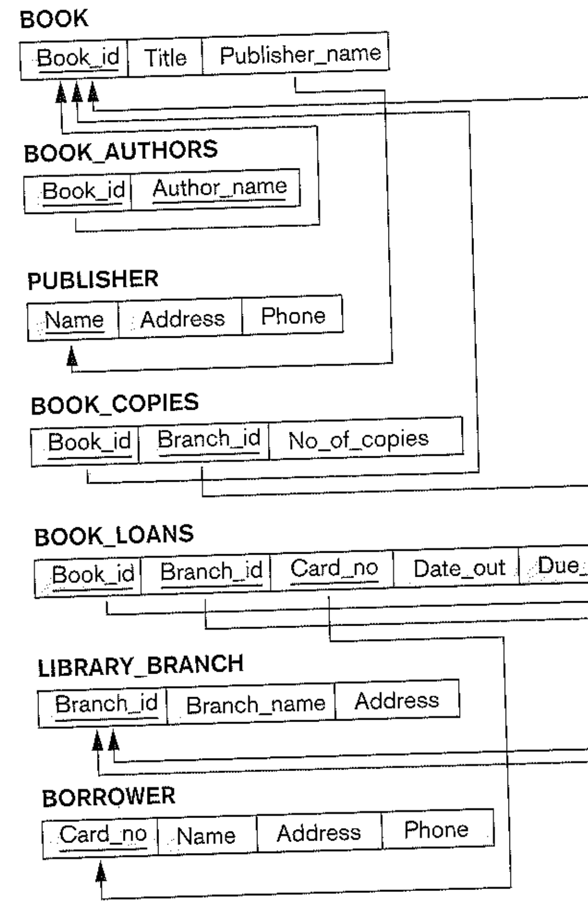
e. Retrieve the names of all employees who work on every project.
f. Retrieve the names of all employees who do not work on any project.
g. For each department, retrieve the department name and the average salary of all employees working in that department.
h. Retrieve the average salary of all female employees.
i. Find the names and addresses of all 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 all department managers who have no dependents.

6.17. Consider the AIRLINE relational database schema shown in Figure 3.8, which was described in Exercise 3.12. Specify the following queries in relational algebra:

a. For each flight, list the flight number, the departure airport for the first leg of the flight, and the arrival airport for the last leg of the flight.
b. List the flight numbers and weekdays of all flights or flight legs that depart from Houston Intercontinental Airport (airport code 'IAH') and arrive in Los Angeles International Airport (airport code 'LAX').
c. List the flight number, departure airport code, scheduled departure time, arrival airport code, scheduled arrival time, and weekdays of all flights or flight legs that depart from some airport in the city of Houston and arrive at some airport in the city of Los Angeles.
d. List all fare information for flight number 'CO197'.
e. Retrieve the number of available seats for flight number 'CO197' on '2009-10-09'.

6.18. Consider the LIBRARY relational database schema shown in Figure 6.14, which is used to keep track of books, borrowers, and book loans. Referential integrity constraints are shown as directed arcs in Figure 6.14, as in the notation of Figure 3.7. Write down relational expressions for the following queries:

a. How many copies of the book titled *The Lost Tribe* are owned by the library branch whose name is 'Sharpstown'?
b. How many copies of the book titled *The Lost Tribe* are owned by each library branch?
c. Retrieve the names of all borrowers who do not have any books checked out.
d. For each book that is loaned out from the Sharpstown branch and whose Due_date is today, retrieve the book title, the borrower's name, and the borrower's address.
e. For each library branch, retrieve the branch name and the total number of books loaned out from that branch.



f. Retrieve the names, addresses, and number of book
borrowers who have more than five books checked o
g. For each book authored (or coauthored) by Stephe
title and the number of copies owned by the library
is Central.

6.19. Specify the following queries in relational algebra on
given in Exercise 3.14:

a. List the Order# and Ship_date for all orders shipped f
b. List the WAREHOUSE information from which th
Jose Lopez was supplied his orders. Produce a listing

e names of all employees who work on every project.

e names of all employees who do not work on any project.

lepartment, retrieve the department name and the average
l employees working in that department.

e average salary of all female employees.

ames and addresses of all employees who work on at least one
cated in Houston but whose department has no location in

t names of all department managers who have no dependents.

AIRLINE relational database schema shown in Figure 3.8, which
d in Exercise 3.12. Specify the following queries in relational

ight, list the flight number, the departure airport for the first leg
nt, and the arrival airport for the last leg of the flight.

light numbers and weekdays of all flights or flight legs that
om Houston Intercontinental Airport (airport code 'IAH') and
Los Angeles International Airport (airport code 'LAX').

ight number, departure airport code, scheduled departure time,
rport code, scheduled arrival time, and weekdays of all flights or
that depart from some airport in the city of Houston and arrive
irport in the city of Los Angeles.

re information for flight number 'CO197'.

the number of available seats for flight number 'CO197' on
-09'.

he LIBRARY relational database schema shown in Figure 6.14,
d to keep track of books, borrowers, and book loans. Referential
nstraints are shown as directed arcs in Figure 6.14, as in the nota-
ure 3.7. Write down relational expressions for the following

ny copies of the book titled *The Lost Tribe* are owned by the
ranch whose name is 'Sharpstown'?

ny copies of the book titled *The Lost Tribe* are owned by each
ranch?

the names of all borrowers who do not have any books checked

book that is loaned out from the Sharpstown branch and whose
e is today, retrieve the book title, the borrower's name, and the
er's address.

n library branch, retrieve the branch name and the total number
s loaned out from that branch.

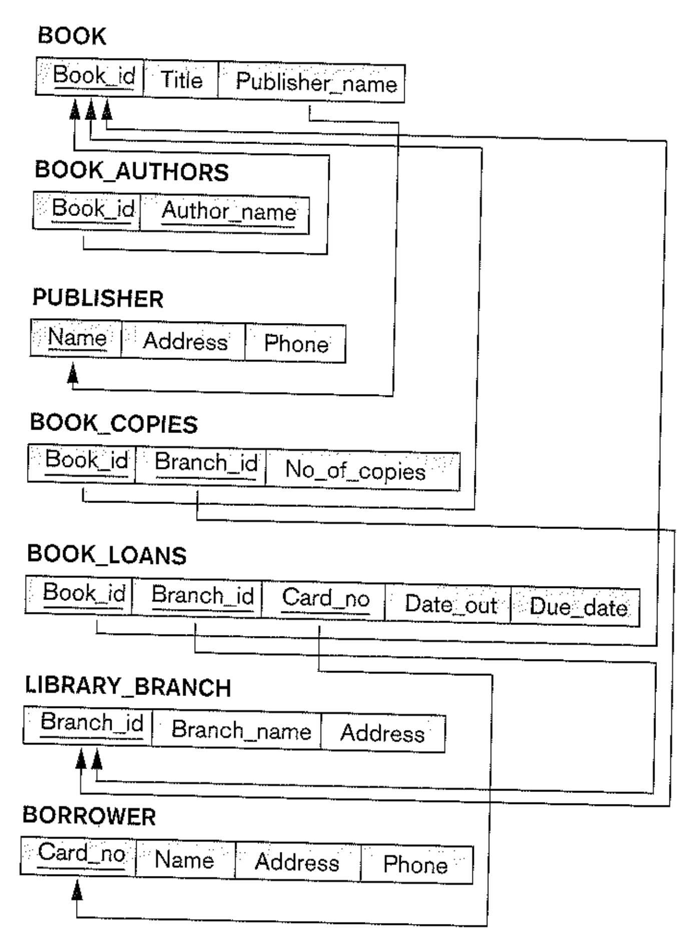


Figure 6.14
A relational database schema for a LIBRARY database.

f. Retrieve the names, addresses, and number of books checked out for all borrowers who have more than five books checked out.

g. For each book authored (or coauthored) by Stephen King, retrieve the title and the number of copies owned by the library branch whose name is Central.

6.19. Specify the following queries in relational algebra on the database schema given in Exercise 3.14:

a. List the Order# and Ship_date for all orders shipped from Warehouse# W2.

b. List the WAREHOUSE information from which the CUSTOMER named Jose Lopez was supplied his orders. Produce a listing: Order#, Warehouse#.

c. Produce a listing Cname, No_of_orders, Avg_order_amt, where the middle column is the total number of orders by the customer and the last column is the average order amount for that customer.

d. List the orders that were not shipped within 30 days of ordering.

e. List the Order# for orders that were shipped from *all* warehouses that the company has in New York.

6.20. Specify the following queries in relational algebra on the database schema given in Exercise 3.15:

a. Give the details (all attributes of trip relation) for trips that exceeded $2,000 in expenses.

b. Print the Ssns of salespeople who took trips to Honolulu.

c. Print the total trip expenses incurred by the salesperson with SSN = '234-56-7890'.

6.21. Specify the following queries in relational algebra on the database schema given in Exercise 3.16:

a. List the number of courses taken by all students named John Smith in Winter 2009 (i.e., Quarter=W09).

b. Produce a list of textbooks (include Course#, Book_isbn, Book_title) for courses offered by the 'CS' department that have used more than two books.

c. List any department that has all its adopted books published by 'Pearson Publishing'.

6.22. Consider the two tables $T1$ and $T2$ shown in Figure 6.15. Show the results of the following operations:

a. $T1 \bowtie_{T1.P = T2.A} T2$

b. $T1 \bowtie_{T1.Q = T2.B} T2$

c. $T1 ⟕_{T1.P = T2.A} T2$

d. $T1 ⟖_{T1.Q = T2.B} T2$

e. $T1 \cup T2$

f. $T1 \bowtie_{(T1.P = T2.A \textbf{ AND } T1.R = T2.C)} T2$

Figure 6.15
A database state for the relations $T1$ and $T2$.

TABLE T1

P	Q	R
10	a	5
15	b	8
25	a	6

TABLE T2

A	B	C
10	b	6
25	c	3
10	b	5

6.23. Specify the following queries in relational algebra on the da
Exercise 3.17:

a. For the salesperson named 'Jane Doe', list the following
all the cars she sold: Serial#, Manufacturer, Sale_price.

b. List the Serial# and Model of cars that have no options.

c. Consider the NATURAL JOIN operation between SALE
SALE. What is the meaning of a left outer join for thes
change the order of relations)? Explain with an example.

d. Write a query in relational algebra involving selection an
tion and say in words what the query does.

6.24. Specify queries a, b, c, e, f, i, and j of Exercise 6.16 in both tu
relational calculus.

6.25. Specify queries a, b, c, and d of Exercise 6.17 in both tuple ar
tional calculus.

6.26. Specify queries c, d, and f of Exercise 6.18 in both tuple an
tional calculus.

6.27. In a tuple relational calculus query with n tuple variables, wha
typical minimum number of join conditions? Why? What
having a smaller number of join conditions?

6.28. Rewrite the domain relational calculus queries that followed
6.7 in the style of the abbreviated notation of Q0A, where the
minimize the number of domain variables by writing consta
variables wherever possible.

6.29. Consider this query: Retrieve the Ssns of employees who wo
those projects on which the employee with Ssn=123456789 wo
be stated as (**FORALL** x) (**IF** P **THEN** Q), where

- x is a tuple variable that ranges over the PROJECT relation.
- $P \equiv$ EMPLOYEE with Ssn=123456789 works on PROJECT x.
- $Q \equiv$ EMPLOYEE e works on PROJECT x.

Express the query in tuple relational calculus, using the rules

- $(\forall x)(P(x)) \equiv \textbf{NOT}(\exists x)(\textbf{NOT}(P(x)))$.
- $(\textbf{IF } P \textbf{ THEN } Q) \equiv (\textbf{NOT}(P) \textbf{ OR } Q)$.

6.30. Show how you can specify the following relational algebra c
both tuple and domain relational calculus.

a. $\sigma_{A=C}(R(A, B, C))$

b. $\pi_{<A, B>}(R(A, B, C))$

c. $R(A, B, C) * S(C, D, E)$

d. $R(A, B, C) \cup S(A, B, C)$

e. $R(A, B, C) \cap S(A, B, C)$

type) compatible; these include UNION, INTERSEC
CARTESIAN PRODUCT operation is a set operation
es from two relations, producing all possible combi
; however, we showed how CARTESIAN PRODUCT
to define matching tuples from two relations and le
ent JOIN operations called THETA JOIN, EQUIJ
oduced. Query trees were introduced as a graphical
bra queries, which can also be used as the basis fo
DBMS can use to represent a query.

portant types of queries that *cannot* be stated with
ations but are important for practical situations.
PROJECTION to use functions of attributes in the p
TE FUNCTION operation to deal with aggregate typ
mmarize the information in the tables. We discuss
there is no direct support in the algebra but which
step approach, as we demonstrated. Then we prese
ER UNION operations, which extend JOIN and UN
n source relations to be preserved in the result.

described the basic concepts behind relational calcu
h of mathematical logic called predicate calculus. T
al calculi: (1) the tuple relational calculus, which us
ver tuples (rows) of relations, and (2) the domain r
domain variables that range over domains (columns
alculus, a query is specified in a single declarative sta
y order or method for retrieving the query result. Hen
n considered to be a higher-level *declarative* language
cause a relational calculus expression states *what* we
how the query may be executed.

tax of relational calculus queries using both tuple and
ced query graphs as an internal representation for qu
e also discussed the existential quantifier (∃) and the u
w that relational calculus variables are bound by these
n detail how queries with universal quantification are
problem of specifying safe queries whose results are fi
for transforming universal into existential quantifiers, a
fiers that give expressive power to the relational calcul
he basic relational algebra. There is no analog to groupi
ns in basic relational calculus, although some extensior

stions

ations of relational algebra and the purpose of each.

hat is union compatibility? Why do the UNION, INTERSECTION, and
FFERENCE operations require that the relations on which they are applied
union compatible?

iscuss some types of queries for which renaming of attributes is necessary
order to specify the query unambiguously.

iscuss the various types of *inner join* operations. Why is theta join
quired?

hat role does the concept of *foreign key* play when specifying the most
mmon types of meaningful join operations?

hat is the FUNCTION operation? What is it used for?

ow are the OUTER JOIN operations different from the INNER JOIN operations? How is the OUTER UNION operation different from UNION?

what sense does relational calculus differ from relational algebra, and in
hat sense are they similar?

ow does tuple relational calculus differ from domain relational calculus?

iscuss the meanings of the existential quantifier (∃) and the universal
uantifier (∀).

efine the following terms with respect to the tuple calculus: *tuple variable*,
ange relation, *atom*, *formula*, and *expression*.

efine the following terms with respect to the domain calculus: *domain variable*, *range relation*, *atom*, *formula*, and *expression*.

hat is meant by a *safe expression* in relational calculus?

hen is a query language called relationally complete?

ises

how the result of each of the sample queries in Section 6.5 as it would apply
o the database state in Figure 3.6.

pecify the following queries on the COMPANY relational database schema
hown in Figure 5.5, using the relational operators discussed in this chapter.
lso show the result of each query as it would apply to the database state in
igure 3.6.

a. Retrieve the names of all employees in department 5 who work more than 10 hours per week on the ProductX project.
b. List the names of all employees who have a dependent with the same first name as themselves.
c. Find the names of all employees who 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.

tabase
LIBRARY

e

r type) compatible; these include UNION, INTERSECTION, and
e CARTESIAN PRODUCT operation is a set operation that can
les from two relations, producing all possible combinations. It
ce; however, we showed how CARTESIAN PRODUCT followed
l to define matching tuples from two relations and leads to the
erent JOIN operations called THETA JOIN, EQUIJOIN, and
troduced. Query trees were introduced as a graphical represen-
ebra queries, which can also be used as the basis for internal
e DBMS can use to represent a query.

portant types of queries that *cannot* be stated with the basic
rations but are important for practical situations. We intro-
PROJECTION to use functions of attributes in the projection
ATE FUNCTION operation to deal with aggregate types of sta-
ummarize the information in the tables. We discussed recur-
n there is no direct support in the algebra but which can be
step approach, as we demonstrated. Then we presented the
TER UNION operations, which extend JOIN and UNION and
n source relations to be preserved in the result.

described the basic concepts behind relational calculus, which
h of mathematical logic called predicate calculus. There are
al calculi: (1) the tuple relational calculus, which uses tuple
ver tuples (rows) of relations, and (2) the domain relational
domain variables that range over domains (columns of rela-
alculus, a query is specified in a single declarative statement,
y order or method for retrieving the query result. Hence, rela-
a considered to be a higher-level *declarative* language than the
cause a relational calculus expression states *what* we want to
how the query may be executed.

ax of relational calculus queries using both tuple and domain
ced query graphs as an internal representation for queries in
e also discussed the existential quantifier (∃) and the universal
v that relational calculus variables are bound by these quanti-
detail how queries with universal quantification are written,
problem of specifying safe queries whose results are finite. We
r transforming universal into existential quantifiers, and vice
ers that give expressive power to the relational calculus, mak-
e basic relational algebra. There is no analog to grouping and
in basic relational calculus, although some extensions have

ions

ions of relational algebra and the purpose of each.

6.2. What is union compatibility? Why do the UNION, INTERSECTION, and DIFFERENCE operations require that the relations on which they are applied be union compatible?

6.3. Discuss some types of queries for which renaming of attributes is necessary in order to specify the query unambiguously.

6.4. Discuss the various types of *inner join* operations. Why is theta join required?

6.5. What role does the concept of *foreign key* play when specifying the most common types of meaningful join operations?

6.6. What is the FUNCTION operation? What is it used for?

6.7. How are the OUTER JOIN operations different from the INNER JOIN operations? How is the OUTER UNION operation different from UNION?

6.8. In what sense does relational calculus differ from relational algebra, and in what sense are they similar?

6.9. How does tuple relational calculus differ from domain relational calculus?

6.10. Discuss the meanings of the existential quantifier (∃) and the universal quantifier (∀).

6.11. Define the following terms with respect to the tuple calculus: *tuple variable, range relation, atom, formula,* and *expression.*

6.12. Define the following terms with respect to the domain calculus: *domain variable, range relation, atom, formula,* and *expression.*

6.13. What is meant by a *safe expression* in relational calculus?

6.14. When is a query language called relationally complete?

Exercises

6.15. Show the result of each of the sample queries in Section 6.5 as it would apply to the database state in Figure 3.6.

6.16. Specify the following queries on the COMPANYrelational 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 in Figure 3.6.

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

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

c. Find the names of all employees who 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.