```
person (<u>driver-id</u>, name, address)
car (<u>license</u>, model, year)
accident (report-number, <u>date</u>, location)
owns (<u>driver-id</u>, <u>license</u>)
participated (<u>driver-id</u>, <u>car</u>, report-number, damage-amount)
```

Figure 4.12. Insurance database.

```
insert into accident
    values (4007, '2001-09-01', 'Berkeley')

insert into participated
    select o.driver-id, c.license, 4007, 3000
    from person p, owns o, car c
    where p.name = 'Jones' and p.driver-id = o.driver-id and
        o.license = c.license and c.model = 'Toyota'
```

d. Delete the Mazda belonging to "John Smith".

Since *model* is not a key of the *car* relation, we can either assume that only one of John Smith's cars is a Mazda, or delete all of John Smith's Mazdas (the query is the same). Again assume *name* is a key for *person*.

```
delete car
where model = 'Mazda' and license in
  (select license
   from person p, owns o
   where p.name = 'John Smith' and p.driver-id = o.driver-id)
```

Note: The *owns*, *accident* and *participated* records associated with the Mazda still exist.

e. Update the damage amount for the car with license number "AABB2000" in the accident with report number "AR2197" to \$3000.

```
update participated

set damage-amount = 3000

where report-number = "AR2197" and driver-id in

(select driver-id

from owns

where license = "AABB2000")
```

- **4.2** Consider the employee database of Figure 4.13, where the primary keys are underlined. Give an expression in SQL 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 in the database who live in the same cities as the companies for which they work.
- **e.** Find all employees in the database who live in the same cities and on the same streets as do their managers.
- f. Find all employees in the database who do not work for First Bank Corporation.
- **g.** Find all employees in the database who earn more than each 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.
- **i.** Find all employees who earn more than the average salary of all employees of their company.
- j. Find the company that has the most employees.
- **k.** Find the company that has the smallest payroll.
- **1.** Find those companies whose employees earn a higher salary, on average, than the average salary at First Bank Corporation.

Answer:

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

```
select employee-name
from works
where company-name = '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 address, and cities of residence of all employees who work for First Bank Corporation and earn more than \$10,000.

If people may work for several companies, the following solution will only list those who earn more than \$10,000 per annum from "First Bank Corporation" alone.

```
select *
from employee
where employee-name in
   (select employee-name
   from works
   where company-name = 'First Bank Corporation' and salary ; 10000)
```

As in the solution to the previous query, we can use a join to solve this one

d. Find all employees in the database who live in the same cities as the companies for which they work.

e. Find all employees in the database who live in the same cities and on the same streets as do their managers.

```
select P.employee-name
from employee P, employee R, manages M
where P.employee-name = M.employee-name and
M.manager-name = R.employee-name and
P.street = R.street and P.city = R.city
```

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

The following solution assumes that all people work for exactly one company.

```
select employee-name from works where company-name ≠ 'First Bank Corporation'
```

If one allows people to appear in the database (e.g. in *employee*) but not appear in *works*, or if people may have jobs with more than one company, the solution is slightly more complicated.

```
select employee-name
from employee
where employee-name not in
  (select employee-name
  from works
  where company-name = 'First Bank Corporation')
```

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

The following solution assumes that all people work for at most one company.

```
select employee-name
from works
where salary > all
  (select salary
  from works
  where company-name = 'Small Bank Corporation')
```

If people may work for several companies and we wish to consider the *total* earnings of each person, the problem is more complex. It can be solved by using a nested subquery, but we illustrate below how to solve it using the **with** clause.

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.

The simplest solution uses the **contains** comparison which was included in the original System R Sequel language but is not present in the subsequent SQL versions.

Below is a solution using standard SQL.

i. Find all employees who earn more than the average salary of all employees of their company.

The following solution assumes that all people work for at most one company.

```
employee (employee-name, street, city)
works (employee-name, company-name, salary)
company (company-name, city)
manages (employee-name, manager-name)
```

Figure 4.13. Employee database.

j. Find the company that has the most employees.

```
select company-name
from works
group by company-name
having count (distinct employee-name) >= all
    (select count (distinct employee-name)
    from works
    group by company-name)
```

k. Find the company that has the smallest payroll.

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

- **4.3** Consider the relational database of Figure 4.13. Give an expression in SQL for each of the following queries.
 - **a.** Modify the database so that Jones now lives in Newtown.
 - **b.** Give all employees of First Bank Corporation a 10 percent raise.
 - **c.** Give all managers of First Bank Corporation a 10 percent raise.
 - **d.** Give all managers of First Bank Corporation a 10 percent raise unless the salary becomes greater than \$100,000; in such cases, give only a 3 percent raise.

e. Delete all tuples in the *works* relation for employees of Small Bank Corporation.

Answer: The solution for part 0.a assumes that each person has only one tuple in the *employee* relation. The solutions to parts 0.c and 0.d assume that each person works for at most one company.

a. Modify the database so that Jones now lives in Newtown.

```
update employee
set city = 'Newton'
where person-name = 'Jones'
```

b. Give all employees of First Bank Corporation a 10-percent raise.

```
update works
set salary = salary * 1.1
where company-name = 'First Bank Corporation'
```

c. Give all managers of First Bank Corporation a 10-percent raise.

d. Give all managers of First Bank Corporation a 10-percent raise unless the salary becomes greater than \$100,000; in such cases, give only a 3-percent raise.

```
update works T
set T.salary = T.salary * 1.03
where T.employee-name in (select manager-name
from manages)
and T.salary * 1.1 > 100000
and T.company-name = 'First Bank Corporation'

update works T
set T.salary = T.salary * 1.1
where T.employee-name in (select manager-name
from manages)
and T.salary * 1.1 <= 100000
and T.company-name = 'First Bank Corporation'
```

SQL-92 provides a **case** operation (see Exercise 4.11), using which we give a more concise solution:-

from manages) and

T.company-name = 'First Bank Corporation'

e. Delete all tuples in the works relation for employees of Small Bank Corporation.

delete works

where company-name = 'Small Bank Corporation'

4.4 Let the following relation schemas be given:

$$R = (A, B, C)$$

$$S = (D, E, F)$$

Let relations r(R) and s(S) be given. Give an expression in SQL that is equivalent to each of the following queries.

- **a.** $\Pi_A(r)$
- **b.** $\sigma_{B=17}(r)$
- c. $r \times s$
- **d.** $\Pi_{A,F}\left(\sigma_{C=D}(r\times s)\right)$

Answer:

a. $\Pi_A(r)$

select distinct A

from r

b. $\sigma_{B=17}(r)$

select * from r

where B = 17

c. $r \times s$

select distinct * from r, s

d. $\Pi_{A,F}\left(\sigma_{C=D}(r\times s)\right)$

select distinct A, F

from r, s

where C = D

- **4.5** Let R = (A, B, C), and let r_1 and r_2 both be relations on schema R. Give an expression in SQL that is equivalent to each of the following queries.
 - **a.** $r_1 \cup r_2$
 - **b.** $r_1 \cap r_2$

c.
$$r_1 - r_2$$

d. $\Pi_{AB}(r_1) \bowtie \Pi_{BC}(r_2)$

Answer:

a. $r_1 \cup r_2$

(select * from r1) union (select * from r2)

b. $r_1 \cap r_2$

We can write this using the **intersect** operation, which is the preferred approach, but for variety we present an solution using a nested subquery.

 $\begin{array}{c} \mathbf{select} \ ^* \\ \mathbf{from} \ r1 \\ \mathbf{where} \ (\mathbf{A}, \mathbf{B}, \mathbf{C}) \ \mathbf{in} \ (\mathbf{select} \ ^* \\ \mathbf{from} \ r2) \end{array}$

c. $r_1 - r_2$

 $\begin{array}{c} \mathbf{select} * \\ \mathbf{from} \ r1 \\ \mathbf{where} \ (\mathbf{A}, \mathbf{B}, \mathbf{C}) \ \mathbf{not} \ \mathbf{in} \ (\mathbf{select} * \\ \mathbf{from} \ r2) \end{array}$

This can also be solved using the **except** clause.

d. $\Pi_{AB}(r_1) \bowtie \Pi_{BC}(r_2)$

- **4.6** Let R=(A,B) and S=(A,C), and let r(R) and s(S) be relations. Write an expression in SQL for each of the queries below:
 - **a.** $\{ < a > \mid \exists \ b \ (< a, b > \in \ r \ \land \ b \ = \ 17) \}$
 - **b.** $\{ \langle a, b, c \rangle \mid \langle a, b \rangle \in r \land \langle a, c \rangle \in s \}$
 - **c.** $\{ \langle a \rangle \mid \exists c \ (\langle a, c \rangle \in s \land \exists b_1, b_2 \ (\langle a, b_1 \rangle \in r \land \langle c, b_2 \rangle \in r \land b_1 > b_2)) \}$

Answer:

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

select distinct Afrom rwhere B = 17

b.
$$\{ \langle a, b, c \rangle \mid \langle a, b \rangle \in r \land \langle a, c \rangle \in s \}$$