CS 4513 Dr. Le Gruenwald
Solutions for Practice Homework Assignment 2

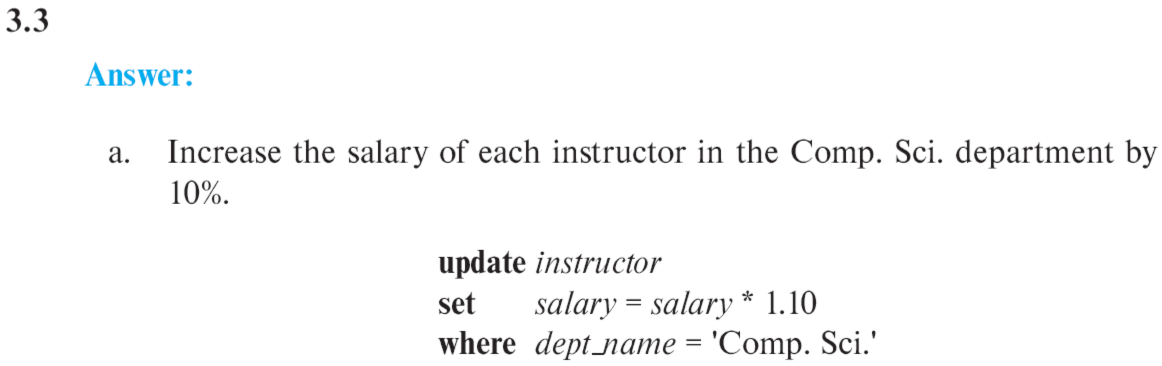
Problem 1

Employee (ename, ssn, dname, address)
Department (dname, number, ssn)
Project (pname, pno, location, dname)
Dependent (ssn, depname, relationship)
Works_on (ssn, pno, hours)

Problem 2

Office (building_name, number, phone)
Faculty_member (head, fname)
Company(type, address, cname)
Course (coursename, level, credit, sname, fname)
Staff (sname, address)
Tutorial (day, room, time, tutorial_no, coursename)
Student (address, studname, emergency_contact)
Undergraduate (studname, minor)
Graduate (studname, advisor, specialty)
Major (mname, no_of_reqd_courses)
Has_office (building_name, number, fname)
Manages (mname, fname)
Consults (fname, cname, address, hrs_per_week)
Enrolls (studname, mname)
Takes (coursename, studname, semester, year)
Prerequisites (coursename, prerequisite_name)
Tutors (tutorial_no, coursename, sname)

Problem 3

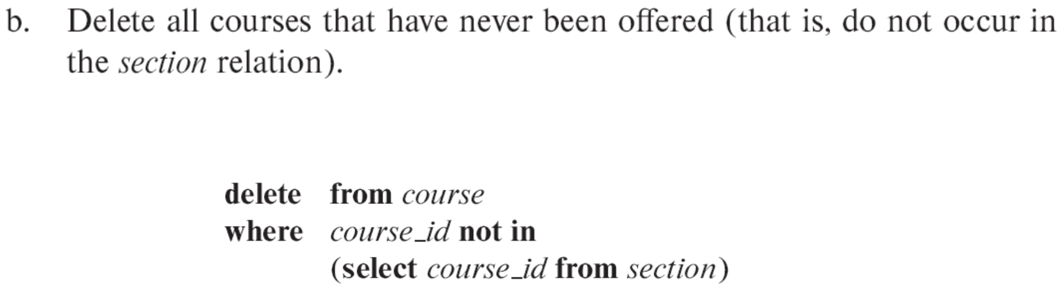


3.3

Answer:

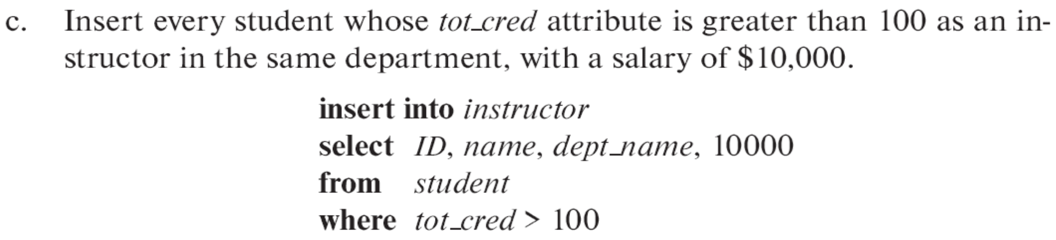
a. Increase the salary of each instructor in the Comp. Sci. department by 10%.

```
update instructor
set    salary = salary * 1.10
where  dept_name = 'Comp. Sci.'
```



b. Delete all courses that have never been offered (that is, do not occur in the *section* relation).

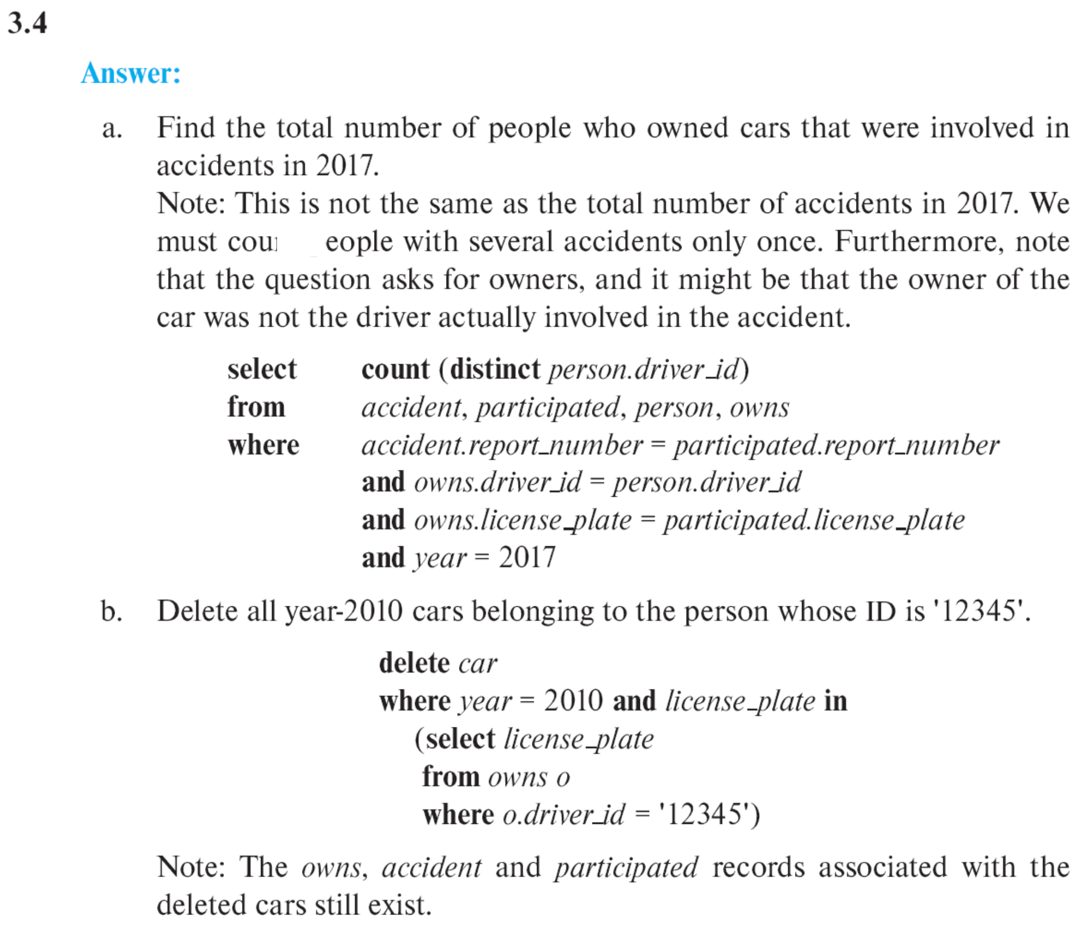
```
delete  from course
where   course_id not in
        (select course_id from section)
```



c. Insert every student whose *tot_cred* attribute is greater than 100 as an instructor in the same department, with a salary of $10,000.

```
insert into instructor
select  ID, name, dept_name, 10000
from    student
where   tot_cred > 100
```

Problem 4



3.4

Answer:

a. Find the total number of people who owned cars that were involved in accidents in 2017.
Note: This is not the same as the total number of accidents in 2017. We must cou eople with several accidents only once. Furthermore, note that the question asks for owners, and it might be that the owner of the car was not the driver actually involved in the accident.

```
select    count (distinct person.driver_id)
from      accident, participated, person, owns
where     accident.report_number = participated.report_number
          and owns.driver_id = person.driver_id
          and owns.license_plate = participated.license_plate
          and year = 2017
```

b. Delete all year-2010 cars belonging to the person whose ID is '12345'.

```
delete car
where year = 2010 and license_plate in
    (select license_plate
     from owns o
     where o.driver_id = '12345')
```

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

Problem 5

3.9

Answer:

a. Find the ID, name, and city of residence of each employee who works for "First Bank Corporation".

```
select e.ID, e.person_name, city
from employee as e, works as  w
where w.company_name = 'First Bank Corporation' and
        w.ID = e.ID
```

b. Find the ID, name, and city of residence of each employee who works for "First Bank Corporation" and earns more than $10000.

```
select *
from employee
where ID in
   (select ID
    from works
    where company_name = 'First Bank Corporation' and salary > 10000)
```

This could be written also in the style of the answer to part *a*.

c. Find the ID of each employee who does not work for "First Bank Corporation".

```
select ID
from works
where company_name <> 'First Bank Corporation'
```

If one allows people to appear in *employee* without appearing also in *works*, the solution is slightly more complicated. An outer join as discussed in Chapter 4 could be used as well.

```
select ID
from employee
where ID not in
   (select ID
    from works
    where company_name = 'First Bank Corporation')
```

d. Find the ID of each employee who earns more than every employee of "Small Bank Corporation".

```
select ID
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. But note that the fact that ID is the primary key for *works* implies that this cannot be the case.

e. Assume that companies may be located in several cities. Find the name of each company that is located in every city in which “Small Bank Corporation” is located.

```
select S.company_name
from company as S
where not exists ((select city
                   from company
                   where company_name = 'Small Bank Corporation')
                except
                  (select city
                   from company as T
                   where S.company_name = T.company_name))
```

f. Find the name of the company that has the most employees (or companies, in the case where there is a tie for the most).

```
select company_name
from works
group by company_name
having count (distinct ID) >= all
    (select count (distinct ID)
     from works
     group by company_name)
```

g. Find the name of each company whose employees earn a higher salary, on average, than the average salary at “First Bank Corporation”.

```
select company_name
from works
group by company_name
having avg (salary) > (select avg (salary)
                       from works
                       where company_name = 'First Bank Corporation')
```

Problem 6

3.10

Answer:

a. Modify the database so that the employee whose ID is '12345' now lives in "Newtown".

```
update employee
set city = 'Newtown'
where ID = '12345'
```

b. Give each manager of "First Bank Corporation" a 10 percent raise unless the salary becomes greater than $100000; in such cases, give only a 3 percent raise.

```
update works T
set T.salary = T.salary * 1.03
where T .ID in (select manager_id
                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 .ID in (select manager_id
                from manages)
      and T.salary * 1.1 <= 100000
      and T.company_name = 'First Bank Corporation'
```

The above updates would give different results if executed in the opposite order. We give below a safer solution using the **case** statement.

```
update works T
set T.salary = T.salary ∗
   (case
        when (T.salary ∗ 1.1  >  100000) then 1.03
        else 1.1
   end)
where T.ID in (select manager_id
               from manages) and
       T.company_name = 'First Bank Corporation'
```