3.11 Write the following queries in SQL, using the university schema.  
a. Find the names of all students who have taken at least one Comp. Sci. course; make sure there are no duplicate names in the result.  
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

Solution:

3.11.a:

***select*** name

***from***student **natural** **join** takes **natural** **join** course

***where*** course.dept = ‘Comp.Sci’

3.11.b:

***select*** ID, name

***from*** student

***except***

***select*** ID, name

***from*** student **natural** **join** takes

***where*** year < 2009

3.11.c:

***select*** dept, max(salary)

***from*** instructor

***group*** by dept

3.11.d:

***select*** min(maxSalary)

***from*** (***select*** dept, max(salary) as maxSalary

***from*** instructor

***group*** by dept)

3.12 Write the following queries in SQL, using the university schema.  
a. Create a new course “CS-001”, titled “Weekly Seminar”, with 0 credits.

b. Create a section of this course in Autumn 2009, with *section id* of 1.  
c. Enroll every student in the Comp. Sci. department in the above section.  
d. Delete enrollments in the above section where the student’s name is Chavez.  
e. Delete the course CS-001. What will happen if you run this delete statement without first deleting offerings (sections) of this course.  
f. Delete all *takes* tuples corresponding to any section of any course with the word “database” as a part of the title; ignore case when matching the word with the title.

Solution:

3.12.a:

***insert into*** course

***values***(‘CS-001’, ‘Weekly Seminar’, ‘Comp.Sci’, 0)

3.12.b:

***insert into*** section

***values***(‘CS-001’, 1, ‘Autumn’, 2009, null, null, null)

3.12.c:

***insert*** into takes

***select*** ID, ‘CS-001’,1 ‘Autumn’, 2009, null

***from*** student

***where*** dept\_name = ‘Comp.Sci.’

3.12.d:

***delete from*** takes

***where*** course\_id = ‘CS-001’ ***and*** section\_id = 1 ***and*** year = 2009 ***and*** semester = ‘Autumn’ ***and***

ID ***in*** (***select*** ID ***from*** student ***where*** name = ‘Chavez’)

3.12.e:

会直接破坏section course takes相互之间的外码参照关系

3.12.f:

***delete from*** takes

***where*** course\_id ***in*** (***select*** course\_id

***from*** course

***where*** lower(title) like ‘%database%’)

3.15 Consider the bank database of Figure 3.19, where the primary keys are underlined. Construct the following SQL queries for this relational database.

*branch*(*branch name*, *branch city, assets*)  
*customer* (*customer name*, *customer street, customer city*)  
*loan* (*loan number*, *branch name, amount*)  
*borrower* (*customer name*, *loan number*)  
*account* (*account number*, *branch name, balance* )  
*depositor* (*customer name*, *account number*)  
**Figure 3.19** Banking database for Exercises 3.8 and 3.15.

a. Find all customers who have an account at *all* the branches located in “Brooklyn”.  
b. Find out the total sum of all loan amounts in the bank.  
c. Find the names of all branches that have assets greater than those of at least one branch located in “Brooklyn”.

Solution:

3.15.a:

…..

3.15.b:

***select*** sum(amount)

***from*** loan

3.15.c:

***select*** branch\_name

***from*** branch

***where*** assets > some

(***select*** assets ***from*** branch ***where*** branch\_city = ‘Brooklyn’)

**3.17** Consider the relational database of Figure 3.20. Give an expression in SQL for each of the following queries.

*employee* (*employee name*, *street*, *city*)  
*works* (*employee name*, *company name*, *salary*)  
*company* (*company name*, *city*)  
*manages* (*employee name*, *manager name*)  
**Figure 3.20** Employee database for Exercises 3.9, 3.10, 3.16, 3.17, and 3.20

a. Give all employees of First Bank Corporation a 10 percent raise.  
b. Give all managers of First Bank Corporation a 10 percent raise.  
c. Delete all tuples in the *works* relation for employees of Small Bank Corporation.

Solution:

3.17.a:

***update*** works

***set*** salary = salary \* 1.1

***where*** company\_name = ‘First Bank Corporation’

3.17.b:

***update*** works

***set*** salary = salary \*1.1

***where*** employee\_name ***in*** (***select*** manager\_name ***from*** manages) ***and*** company\_name = ‘First Bank Corporation’

3.17.c:

***delete*** ***from*** works

***where*** company\_name = ‘Smakk Bank Corporation’

**3.24** Consider the query:

**with** *dept\_total* (*dept name*, *value*) **as**(**select** *dept name*, **sum**(*salary*)  
**from** *instructor***group by** *dept name*),  
*dept total avg*(*value*) **as**(**select avg**(*value*)  
**from** *dept total*)  
**select** *dept name***from** *dept total*, *dept total avg***where** *dept total.value* >= *dept total avg.value*;  
Rewrite this query without using the **with** construct.

Solution:

***select distinct*** dept\_name dn

***from*** instructor i

***where*** (***select*** ***sum***(salary) ***from*** instructo ***where*** department = d) >=

(***select*** ***avg***(sa)

***from*** (***select*** ***sum***(salary) ***as*** s ***from*** instructor ***group*** ***by*** department))