1. Write the following inserts, deletes or updates in SQL, using the university schema.

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

**update** instructor

**set** salary = salary \* 1.1

**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;

d. create a new course "CS-001", titled with "Weekly Seminar", with 0 credits.

**insert into** course (course\_id, title, dept\_name, credits)

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

e. Create a section of this course in Fall 2009, with sec\_id of 1.

**insert** **into** section (course\_id, sec\_id, semester, year, building, room\_number, time\_slot\_id)

**values** (‘CS-001’, 1, ‘Fall’, 2009, NULL, NULL, NULL);

f. Enroll every student in the Comp. Sci. department in the above section.

**insert into** takes (id, course\_id, sec\_id, semester, year, grade)

**select** id, ‘CS-001’, 1, ‘Fall’, 2009, NULL

**from** student

**where** dept\_name = ‘Comp.Sci’;

g. Delete enrollments in the above section where the student's name is Chavez.

**delete** **from** takes

**where** course\_id = ‘CS-001’

**and** sec\_id = 1

**and** semester = ‘Fall’

**and** year = 2009

**and** id **in** (select id

**from** student

**where** name = ‘Chavez’);

h. Delete the course CS-101. What will happen if you run this delete statement without first deleting offerings (sections) of this course.

**delete from** course

**where** course = ‘CS-101’;

可能会无法删除，因为别的表中可能有与 ’CS-101’ 相关的行，违反了外码约束。

i. 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.

**delete fro**m takes

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

**from** course

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

2. Consider the following employee database：

employee(employee\_name, street, city)

works(employee\_name, company\_name, salary)

company(company\_name, city)

manages(employee\_name, manager\_name)

Give an expression in SQL for each of the following queries.

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

**select** employee\_name, city

**from** employee e

**where** e. employee\_name **in** (

**select** employee\_name

**from** works w

**where** w. company\_name = ‘First Bank Corporation’);

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

**select** e.employee\_name, e.street, e.city

**from** employee e, works w

**where** e.employee\_name = w.employee\_name

**and** w.company\_name = ‘First Bank Corporation’

**and** w.salary > 10000;

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

**select** employee\_name

**from** employee

**where** employee\_name **not** **in** (

**select** employee\_name

**from** works

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

1. Find all employees in the database who earn more than each employee of "Small Bank Corporation".

**select** employee\_name

**from** works

**where** salary > ALL(

**select** salary

**from** works

**where** company\_name = ‘Small Bank Corporation’);

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

**select** c1.company\_name

**from** company c1

**where** c1.company\_name <> ‘Small Bank Corporation’

**and** **not** **exists**((select c2.city

**from** company c2

**where** c2.company\_name = ‘Small Bank Corporation’)

**except**

(**select** c3.city

**from** company c3

**where** c3.company\_name = c1.company\_name

));

1. Find the company that has the most employees.

**select** company\_name

**from** work

**group** **by** company\_name

**having** **count**(employee\_name) = (

**select** max(employee\_count)

**from**(select **count**(employee\_name) **as** employee\_count

**from** works

**group** **by** company\_name)

**as** employee\_count);

1. Find those companies 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');