Homework: Chapter 4

4.1 Complete the SQL DDL definition of the database. Identify referential-intergrity constrains that should hold, and include them in the DDL definition.

*employee(employee\_name, street, city)*

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

*company(company\_name, city)*

*manages(employee\_name, manager\_name)*

为上面的数据库定义的模式书写check条件，以保证：

1. 每个雇员工作的公司所在城市和其居住城市相同。
2. 没有雇员的工资比他的经理高。

**create table** *employee*

(*person name* char(20),

*street* char(30),

*city* char(30),

**primary key** (*person name*) )

**create table** *works*

(*person name* char(20),

*company name* char(15),

*salary* integer,

**primary key** (*person name*),

**foreign key** (*person name*) **references** *employee*,

**foreign key** (*company name*) **references** *company*)

**create table** *company*

(*company name* char(15),

*city* char(30),

**primary key** (*company name*))

**create table** *manages*

(*person name* char(20),

*manager name* char(20),

**primary key** (*person name*),

**foreign key** (*person name*) **references** *employee*,

**foreign key** (*manager name*) **references** *employee*)

create assertion city\_constrain check

(not exist (select \* from employee where city not in (select company\_city from works, company where works.company\_name = company.company\_name and works.employee\_name = employee.employee\_name)))

create assertion salary\_constrain check

(not exist (select \* from works as e\_s, works as m\_s, manages where e\_s.employee\_name = manages.employee\_name and m\_s.manager\_name = m\_s.employee\_name and e\_s.salary > m\_s.salary))

4.2 SQL允许外码依赖指向同一个关系，如下面的例子所示：

create table manager(

employee\_name char(20) not null,

manager\_name char(20) not null,

primary key employee\_name,

foreign key(manager\_name) reference manager on delete cascade)

这里employee\_name是关系manager的码，意味着每个employee都至多有一个经理。外码子句要求每个经理都是一个雇员。请准确解释当manager关系中一个元组被删除时会发生什么情况？

The tuples of all employees of the manager, at all levels, get deleted as well! This happens in a series of steps. The initial deletion will trigger deletion of all the tuples corresponding to direct employees of the manager. These deletions will in turn cause deletions of second level employee tuples, and so on, till all direct and indirect employee tuples are deleted.

4.3 请书写一个查询来定义关系

Empl\_depth(employee\_name, manager\_name, depth)

depth指出雇员和经理之间存在多少层中间管理。直接在某经理手下工作的雇员深度为0。

超纲！