

Database system exercise 4

3200105872 庄毅非

4.7 Consider the employee database of Figure 4.12. Give an SQL DDL definition of this database. Identify referential-integrity constraints that should hold, and include them in the DDL definition.

```
1 create table employee(  
2     ID int primary key,  
3     person_name varchar(30) not null,  
4     street varchar(30),  
5     city varchar(30)  
6 );  
7  
8 create table works(  
9     ID int primary key,  
10    company_name varchar(50) not null,  
11    salary double,  
12    foreign key(compant_name) references company(company_name)  
13 );  
14  
15 create table company(  
16     company_name varchar(50) not null,  
17     city varchar(30)  
18 );  
19  
20 create table manages(  
21     ID int primary key,  
22     manager_id int not null,  
23     foreign key(manager_id) references employee(ID)  
24 )
```

4.9 SQL allows a foreign-key dependency to refer to the same relation, as in the following example:

Answer: If this tuple get deleted, then all employees of this manager , not matter indirectly or directly, will be deleted as well. For the employee_ID is the primary key and manager_ID refer to employee_id of this table, when we delete this tuple, every tuple whose manager_ID is equal to current employee will also be deleted recursively. Finally all employees of this manager , not matter indirectly or directly, will be deleted.

4.12 Suppose a user wants to grant select access on a relation to another user. Why should the user include (or not include) the clause granted by current role in the grant statement?

Answer: In both case the granted user will get the select privilege on that table. But if we auth that user by current role, then even we deprecate current user's account, the authorization will persist. If we auth that user by current account, then if we deprecate current user's account, user been authorized will no longer own select permission on that table.