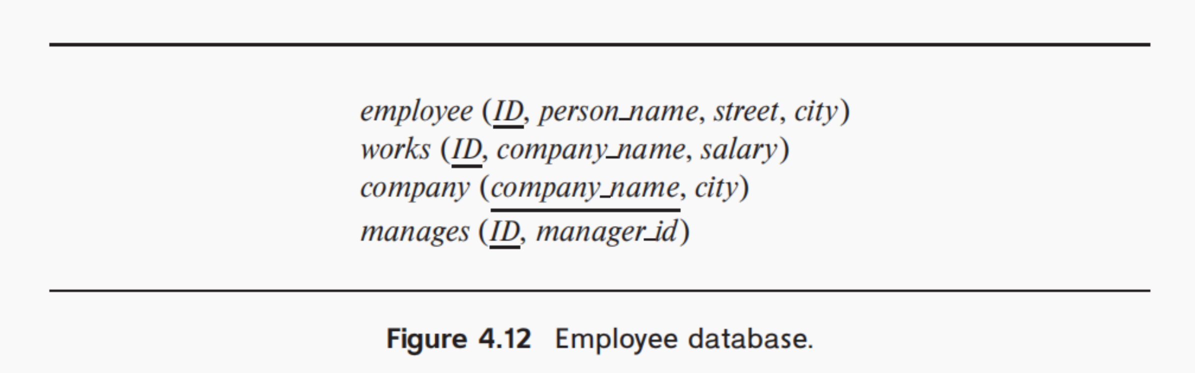
数据库作业week4_2

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.



employee (*ID*, *person_name*, *street*, *city*)
works (*ID*, *company_name*, *salary*)
company (*company_name*, *city*)
manages (*ID*, *manager_id*)

Figure 4.12 Employee database.

```
create table employee(
    ID integer,
    person_name varchar(25),
    street varchar(25),
    city varchar(25),
    PRIMARY KEY (ID)
);

create table works(
    ID integer,
    company_name varchar(25),
    salary numeric(10,2),
    PRIMARY KEY(ID),
    FOREIGN KEY (ID) REFERENCES employee(ID),
    FOREIGN KEY (company_name) REFERENCES company(company_name)
);

create table company(
    company_name varchar(25),
    city varchar(25),
    PRIMARY KEY(company_name)
);

create table manages(
    ID integer,
    manager_id varchar,
    PRIMARY KEY (ID),
    FOREIGN KEY (ID) REFERENCES employee (ID),
    FOREIGN KEY (manager_id) REFERENCES employee(ID)
)
```

4.9

> SQL allows a foreign-key dependency to refer to the same relation, as in the following example:
>
> ```
> create table manager
>     (employee ID char(20),
>     manager ID char(20),
>     primary key employee ID,
>     foreign key (manager ID) references manager(employee ID)
>                             on delete cascade )
> ```
>
> Here, *employee ID* is a key to the table *manager*, meaning that each employee has at most one manager. The foreign-key clause requires that every manager also be an employee. Explain exactly what happens when a tuple in the relation *manager* is deleted

所有此元组的*employee*的直接下级或是间接下级的信息都被删除。由于外键约束指向自己(本表)，同时设置了*on delete cascade*，那么当删除某一行时，假设该行的*employee*是A，那么由于级联的存在将删除所有*manager*为A的元组，之后将删除所有*manager*指向"被删除的*employee*信息'的行。然后连锁反应就导致关于所有此元组的*employee*的直接下级或是间接下级的元组都被删除咯。

All direct or indirect subordinates of the employee in this tuple are deleted. Since the foreign key constraint points to itself (this table) and on delete cascade is set, when a row is deleted, assuming that the employee of the row is A, then all tuples with manager as A will be deleted due to the cascade, and then all rows with manager pointing to the deleted employee information will be deleted. The chain reaction then causes all of the tuple's direct or indirect employees' tuples to 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?

if include the clause **granted by current role** ,the user who get access will have the right for **select** when another user who grants access to him leaves from database,because of the access is granted by role not by person. Similarly, if not include it, access to **select** will be revoked when the user who grants access to him leaves.

So,if you want to make the life of access be longer ,you should use **granted by current role** .