4.7

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

Figure 4.12 Employee database.

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 创建数据库

```
mysql> create database employee;
```

② 创建 company 表

```
mysql> create table company(
    -> company_name varchar(255),
    -> city varchar(255),
    -> primary key(company_name));
```

③ 创建 employee 表

```
mysql> create table employee(
    -> ID varchar(255),
    -> person_name varchar(255),
    -> street varchar(255),
    -> city varchar(255),
    -> primary key(ID),
    -> foreign key(city) references company(company_name));
```

④ 创建 works 表

```
mysql> create table works(
    -> ID varchar(255),
    -> company_name varchar(255),
    -> salary integer,
    -> primary key(ID),
    -> foreign key(company_name) references company(company_name),
    -> foreign key(company_name) references employee(ID));
```

⑤ 创建 manages 表

```
mysql> create table manages(
    -> ID varchar(255),
    -> manager_id varchar(255),
    -> primary key(ID),
    -> foreign key(ID) references employee(ID));
```

4.16

classroom(*building*, *room_number*, *capacity*)
department(*dept_name*, *building*, *budget*)
course(*course_id*, *title*, *dept_name*, *credits*)
instructor(*ID*, *name*, *dept_name*, *salary*)
section(*course_id*, *sec_id*, *semester*, *year*, *building*, *room_number*, *time_slot_id*)
teaches(*ID*, *course_id*, *sec_id*, *semester*, *year*)
student(*ID*, *name*, *dept_name*, *tot_cred*)
takes(*ID*, *course_id*, *sec_id*, *semester*, *year*, *grade*)
advisor(*s_ID*, *i_ID*)
time_slot(*time_slot_id*, *day*, *start_time*, *end_time*)
prereq(*course_id*, *prereq_id*)

Write an SQL query using the university schema to find the ID of each student who has never taken a course at the university. Do this using no subqueries and no set operations (use an outer join).

```
mysql> select name
    -> from student
    -> left join takes on student.id=takes.id
    -> where takes.id is null;
```