

2.2

Consider the foreign-key constraint from the *dept_name* attribute of *instructor* to the *department* relation. Give examples of inserts and deletes to these relations that can cause a violation of the foreign-key constraint.

1.

插入一个元组,其中元组的dept_name不在department中,这样会违反外键约束

2.

删除department中的(Biology,Watson,90000),这样会导致course中的老师违反外键约束

2.9

List two reasons why null values might be introduced into a database.

1.属性的值为空

2.属性的值不存在

3.13

Write SQL DDL corresponding to the schema in Figure 3.17. Make any reasonable assumptions about data types, and be sure to declare primary and foreign keys.

```

1 CREATE TABLE person (
2     driver_id VARCHAR(15) PRIMARY KEY,
3     name VARCHAR(30) NOT NULL,
4     address VARCHAR(40)
5 );
6
7 CREATE TABLE car (
8     license_plate VARCHAR(8) PRIMARY KEY,
9     model VARCHAR(7),
10    year NUMERIC(4,0) CHECK (year > 1701 AND year < 2100),
11 );
12
13 CREATE TABLE accident (
14     report_number VARCHAR(10) PRIMARY KEY,
15     year NUMERIC(4,0) CHECK (year > 1701 AND year < 2100),
16     location VARCHAR(30)
17 );
18
19 CREATE TABLE owns (
20     driver_id VARCHAR(15),
21     license_plate VARCHAR(8),
22     PRIMARY KEY (driver_id, license_plate),
23     FOREIGN KEY (driver_id) REFERENCES person(driver_id)
24     FOREIGN KEY (license_plate) REFERENCES car(license_plate)
25 );
26

```

```

27 CREATE TABLE participated (
28     report_number VARCHAR(10),
29     license_plate VARCHAR(8),
30     driver_id VARCHAR(15),
31     damage_amount NUMERIC(10,2),
32     PRIMARY KEY(report_number, license_plate),
33     FOREIGN KEY (report_number) REFERENCES accident(report_number),
34     FOREIGN KEY (license_plate) REFERENCES car(license_plate)
35 );

```

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 INTEGER PRIMARY KEY ,
3     person_name VARCHAR(50),
4     street VARCHAR(50),
5     city VARCHAR(50)
6 );
7
8 CREATE TABLE company (
9     company_name VARCHAR(50) PRIMARY KEY,
10    city VARCHAR(50),
11 );
12
13 CREATE TABLE works (
14     id INTEGER PRIMARY KEY,
15     company_name VARCHAR(50),
16     salary numeric(10,2),
17     FOREIGN KEY (id) REFERENCES employee(id),
18     FOREIGN KEY (company_name) REFERENCES company(company_name)
19 );
20
21 CREATE TABLE manages (
22     id INTEGER PRIMARY KEY,
23     manager_id INTEGER,
24     FOREIGN KEY (id) REFERENCES employee (id),
25     FOREIGN KEY (manager_id) REFERENCES employee (id)
26 )

```

4.9

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

```
1 CREATE TABLE manager (  
2   employee_id char(20),  
3   manager_id char(20),  
4   PRIMARY KEY employee_id,  
5   FOREIGN KEY (manager_id) REFERENCES manager (employee_id)  
6   ON DELETE CASCADE  
7 );
```

Here, *employee_id* is a key to the table *manager*, meaning that each employee has at 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.

该manager的所有employee的信息都会被删除