

CS304 Tutorial 2 Full Set — Relational Schema and SQL

Abstract

Now it is the time we map our design in the E-R diagram to relational schema, and executable SQL DDL statements. This tutorial contains 2 questions for you to get familiar with this process and a first glance to SQL.

Question 1

Following from Tutorial 1, Question 1. We have the E-R diagram as shown in figure 1. Now your task is to 1) generate from the E-R diagram the corresponding relational schema. 2) Write the corresponding CreateTable statements in SQL.

Comments:

1. first write schema using underline for keys.
2. Create SQL statement for all the attributes.
3. Add in constraints (Key, not_null, foreign key etc.)

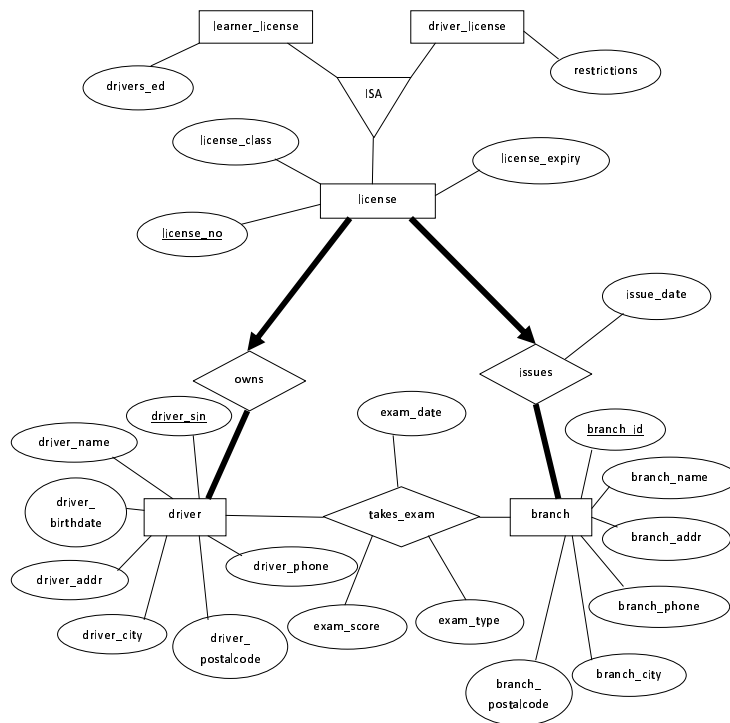


Figure 1: Motor-branch E-R diagram

Answer to Question 1

Schema:

license(license_no, license_class, license_expiry)

learner_license(license_no, drivers_ed)

driver_license(license_no, restrictions)

Driver(driver_sin, driver_name, driver_birthdate, driver_addr, driver_city, driver_postalcode, driver_phone)

Branch(branch_id, branch_name, branch_addr, branch_phone, branch_city, branch_postalcode)

owns(license_no, driver_sin)

takes_exam(driver_sin,branch_id,exam_date, exam_score, exam_type)

issues(license_no, branch_id issue_date)

Question 2

Suppose you execute the following SQL statements in sequence. What is the result at the end of each statement?

- 1) CREATE TABLE r1 (
 a1 INTEGER,
 a2 CHAR(20),
 PRIMARY KEY (a1))
- 2) CREATE TABLE r2(
 a3 INTEGER,
 a4 DATE,
 a5 INTEGER,
 PRIMARY KEY (a3)
 FOREIGN KEY (a5) REFERENCES r1(a1))
- 3) INSERT INTO r1 VALUES (111, '111')
- 4) INSERT INTO r1 VALUES (222, '222')
- 5) INSERT INTO r2 VALUES (1, '08-JUN-2003', 111)
- 6) INSERT INTO r2 VALUES (2, '08-JUN-2003', 333)
- 7) ALTER TABLE r2 ADD a6 CHAR(10)
- 8) INSERT INTO r2 VALUES (3, '08-JUN-2003', 111, 'a')
- 9) DELETE FROM r1 WHERE r1.a2 = '111'
- 10) DROP TABLE r2

Answer to Question 2

1. It creates a relation named r1 (under current workspace), with
 - attributes a1 typed integer,
 - a2 typed string of length 20,
 - let attribute a1 be the primary key of this relation.

2. It creates a relation named r2 with attributes

- a3 typed integer,
- a4 typed date
- a5 typed integer

It also lets attribute a3 be the primary key of this relation and adds a constraint on a5 a foreign key to the (key) attribute a1 in relation r1.

3. Inserts a tuple into relation r1.
4. Inserts a tuple into relation r1.
5. Inserts a tuple into relation r2
6. Inserts a tuple into relation r2
7. add a new attribute a6 typed string of length 10 to relation r2
8. Insert a new tuple to relation r2 (with extended attribute sets)
9. delete tuples in r1 whose value of attribute a2 is '111'
10. delete the relation r2 as well as all the tuples in it.

Question 3

Consider the following ER diagram. Write an SQL specification with primary and foreign key constraints to create tables for relations: Student, Course, and Enrolled.

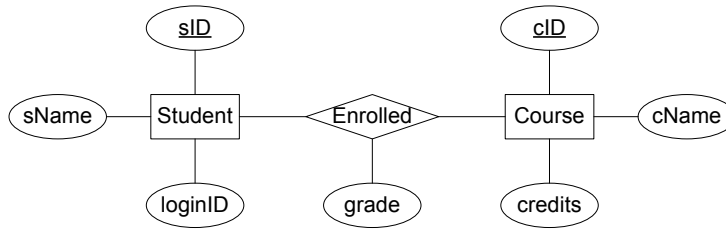


Figure 2: student-enrolment ER-diagram

Answers to Question 3

```
CREATE TABLE student (  
    sID CHAR(8),  
    sName CHAR(30) NOT NULL,  
    loginID CHAR(20) NOT NULL,  
    PRIMARY KEY(sID),  
    UNIQUE(loginID)  
)
```

```
CREATE TABLE course(  
    cID INTEGER,  
    cName CHAR(30) NOT NULL,  
    credit INTEGER NOT NULL,  
    PRIMARY KEY(cID)  
)
```

```
CREATE TABLE enrolled(  
    sID CHAR(8),  
    cID INTEGER,  
    grade REAL,  
    FOREIGN KEY(sID) REFERENCES student ON DELETE CASCADE,  
    FOREIGN KEY(cID) REFERENCES course ON DELETE CASCADE,  
    PRIMARY KEY(sID,cID)  
)
```