Assignment 1: ER Diagram and Relational Model

Due: Oct 7, Sat, 11:59pm

Weight: 8%. Total=61 marks

Submission instruction:

- **This assignment must be done by each student independently.**
- **Hand-written submissions are acceptable but the student is responsible for the clarity.**
- **Submission is through coursys.sfu.ca in a single pdf file with maximum file size of 10MB. Late submission will not be accepted.**

Marking remarks: For any marking related question, please contact the TA responsible for marking the question concerned, as follows. The TA office hours can be found in the introductory note.

Puru Arora paa56@sfu.ca:

AmirMohammad Deilami ada135@sfu.ca:

Obumneme Dukor osd@sfu.ca;

Question 1 (18 marks, 3 marks each). For each of the following statements, represent it using CREATE TABLE statement:

1. Patients attend doctors. You want to store the attending information and the date of attending. Patients identified by pid. Doctors is identified by did.

2. Continue 1, each patient attends doctors at most once.

3. Continue 2, each patient attends doctors at least once.

4. Continue 1, only existing doctors can be attended by a patient.

5. Continue 1, every doctor must be attended by a patient.

6. Continue 1, each of (Name, Address) and (Name,Age) uniquely identifies a patient.

Answer

1.

CREATE TABLE Attends (

did CHAR(20),

pid CHAR(20),

date DATE,

PRIMARY KEY (did,pid,date)) /* a patient can attend any number of doctors in any given day */

CREATE TABLE Patients (

pid CHAR(20),

…

PRIMARY KEY (pid)) /* pid is the primary key of a patient */

CREATE TABLE Doctors (

did CHAR(20),

…

PRIMARY KEY (did)) /* did is the primary key of a doctor */

2. Since each patient attends doctors at most once, we can store the attending information in the Patient table by including did and date of the only attending, and remove did from PRIMARY KEY that can now be determined from pid:

did CHAR(20),

date DATE,

PRIMARY KEY (pid)

Doctor table is same as part 1. Attends table is no longer needed.

3. Everything remains same as in part 2, except that to express “at least once”, in Patients table, we replace did CHAR(20) by

did CHAR(20) NOT NULL

4. This is a foreign key constraint that, so add the following to the Attends table in part 1:

FOREIGN KEY (did) REFERENCES Doctors), /* did is the primary key of Doctors and each did value in the current table must exist in Patients table */

5. This is the total participation of Doctors in the Attending relationship. The only way to represent this constraint is to specify “pid NOT NULL” within the Doctors table (where all doctors are stored). This is possible only if a doctor serves at most one patient (i.e., key constraint on Doctors side), in which case we can store pid directly in Doctors table, similar to Part 2. Since this key constraint is stated in Part 5, we can not represent this total participation constraint.

6. Add the following to the Patients table in Part 1:

UNIQUE (Name, Address),

UNIQUE (Name,Age)

TA’s notes:

Question 2 (18 marks, 3 marks each) A university database contains information about professors (identified by ssn) and courses (identified by courseid). Professors teach courses; each of the following situations concerns the Teaches relationship set. For each situation, draw an ER diagram that describes it (assuming that no further constraints hold) and using CREATE TABLE to model the information in the ER diagram.

1. Professors can teach the same course in several semesters, and each offering must be recorded.

2. Professors can teach the same course in several semesters, and only the most recent such offering needs to be recorded. (Assume this condition applies in all subsequent questions.)

3. Every professor must teach some course.

4. Every professor teaches exactly one course (no more, no less).

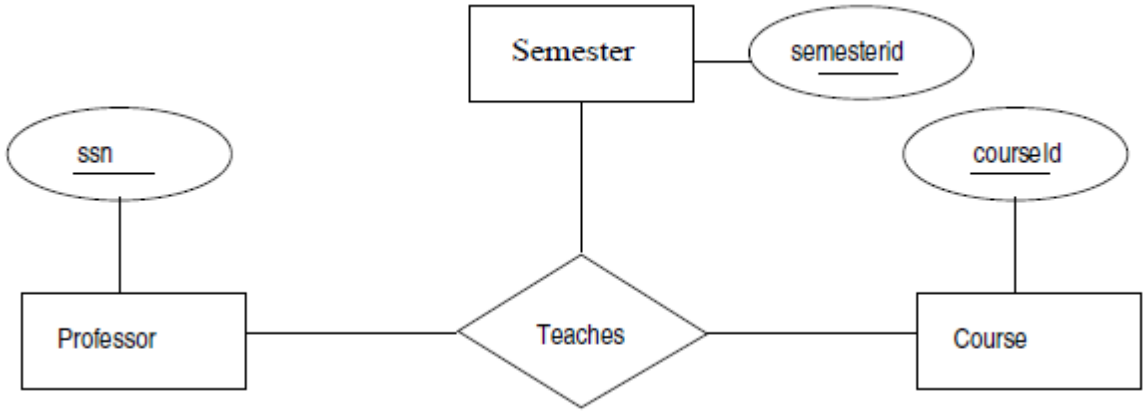
5. Every professor teaches exactly one course (no more, no less), and every course must be taught by some professor.

6. Now suppose that certain courses can be taught by a team of professors jointly, but it is possible that no one professor in a team can teach the course. Model this

situation, introducing additional entity sets and relationship sets if necessary.

Answer:

1.



CREATE TABLE Professor (

ssn CHAR(20)

PRIMARY KEY (ssn))

CREATE TABLE Course (

courseid CHAR(20)

PRIMARY KEY (courseid))

CREATE TABLE Senester (

semesterid CHAR(20)

PRIMARY KEY (semesterid))

CREATE TABLE Teaches (

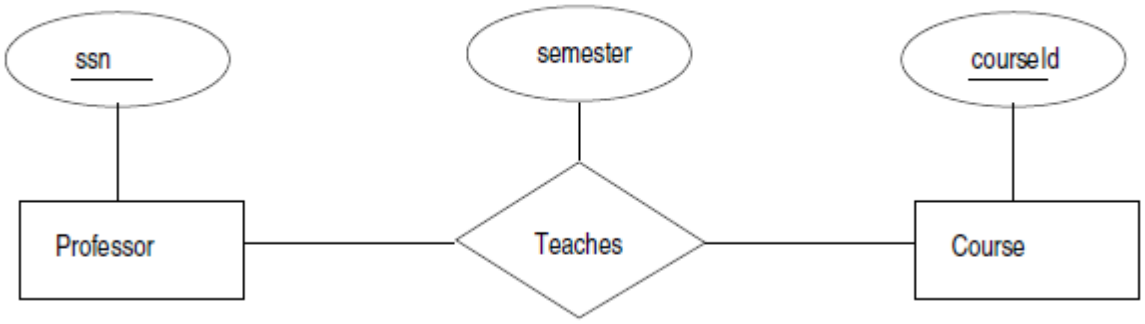
ssn CHAR(20)

courseid CHAR(20)

semesterid CHAR(20)

PRIMARY KEY (ssn, courseid, semesterid))

2.



Tables for Professor and Course are as in 1. Since only the most recent offering will be stored, and there is at most one such offering for each pair of (ssn, coursed), (ssn,coursed) is a key in Teaches table. This can be enforced by adding PRIMARY KEY (ssn, courseid) in the table as follows.

CREATE TABLE Teaches (

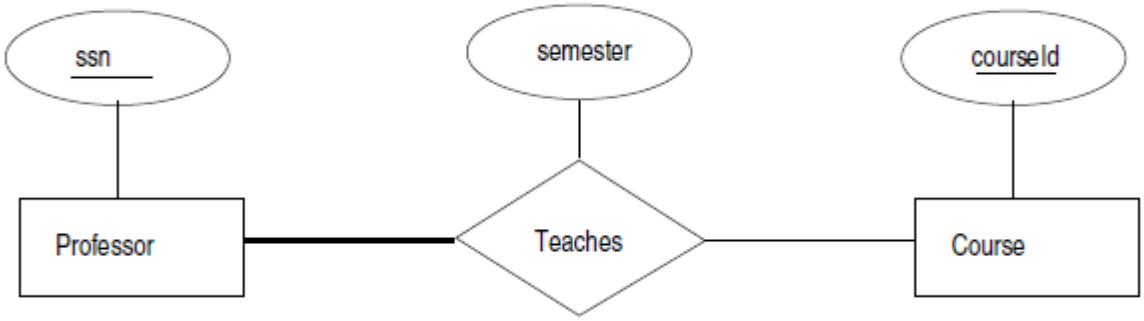
ssn CHAR(20)

courseid CHAR(20)

semesterid CHAR(20) /* which is the semester of the most recent offering */

PRIMARY KEY (ssn, courseid))

3.



Same as in 2. The total participation of Professors in Teaches cannot be modeled using CREATE TABLE without the key constraint on the Professor side.

4.

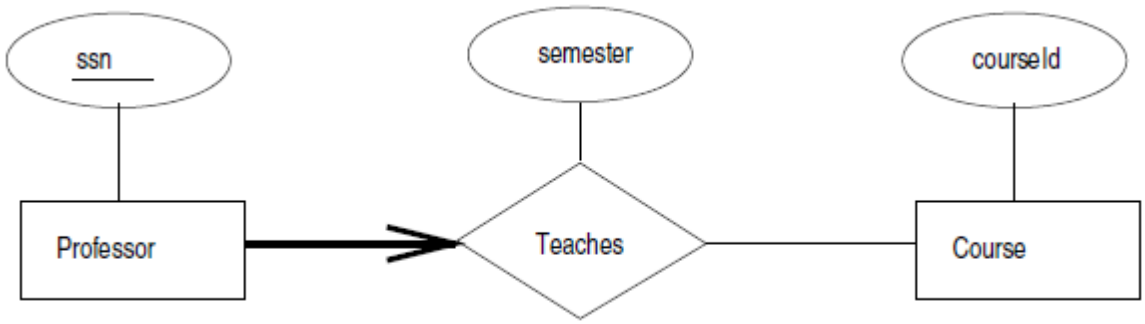


Table for Course is same as in 1. Thanks to the key constraint, Teaches information can be stored in the table for Professor and “at least one” is expressed by “courseid NOT NULL”:

CREATE TABLE Professor (

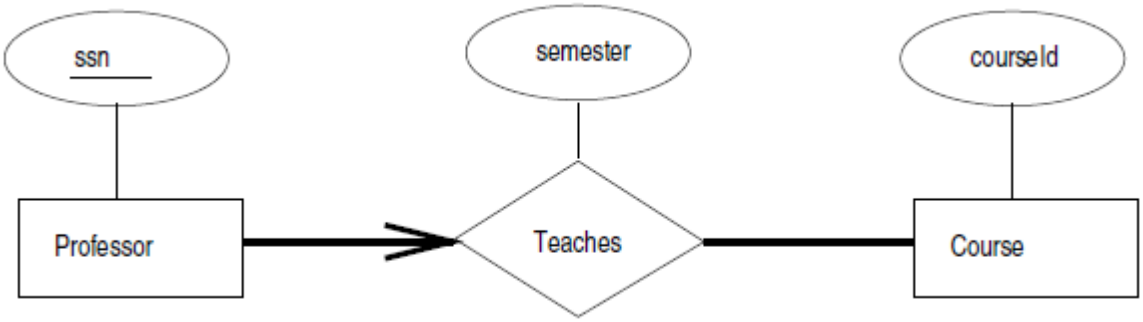
ssn CHAR(20)

courseid CHAR(20) NOT NULL

semesterid CHAR(20) NOT NULL

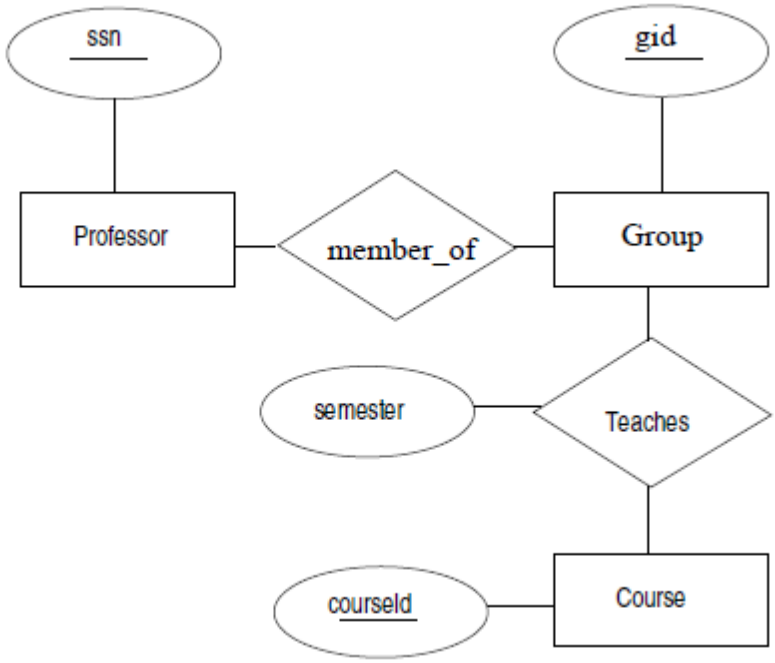
PRIMARY KEY (ssn))

5.



Same as 4. The total participation constraint of Course cannot be represented.

6.



Tables for Professor and Course are same as in 1.

CREATE TABLE Group (

gid CHAR(20)

PRIMARY KEY (gid))

CREATE TABLE Teaches (

gid CHAR(20)

courseid CHAR(20)

semesterid CHAR(20)

PRIMARY KEY (gid, courseid))

CREATE TABLE member_of (

ssn CHAR(20)

gid CHAR(20)

PRIMARY KEY (ssn, gid))

TA's notes:

Question 3 (25marks). You are asked to set up a database, ArtBase, for art galleries. This database will capture all the information that galleries need to

Maintain:

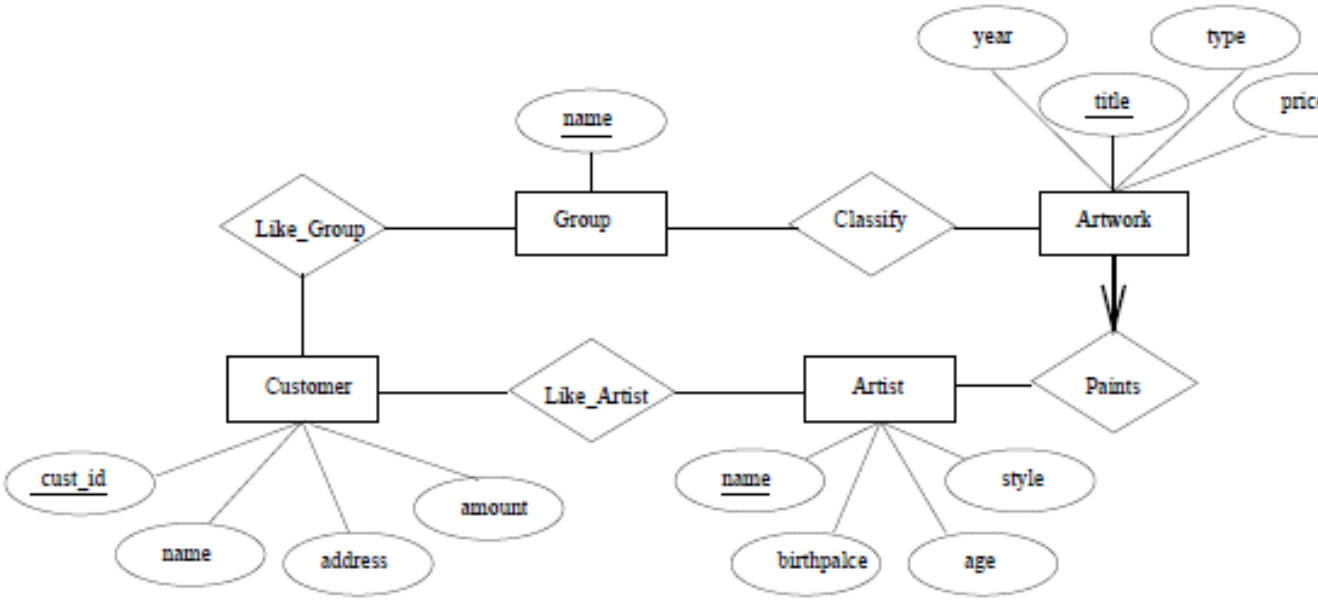
- Galleries keep information about artists, their names (which are unique), birthplaces, age, and style of art.
- For each piece of artwork, the artist, the year it was made, its unique title, its type of art (e.g., painting, lithograph, sculpture, photograph), and its price must be stored.
- Pieces of artwork are also classified into groups of various kinds, for example, portraits, still lifes, works by Picasso, or works of the 19th century; a given piece may belong to more than one group.
- Each group is identified by a name (like those above) that describes the group.
- Finally, galleries keep information about customers. For each customer, galleries keep their unique name, address, total amount of dollars they have spent in the gallery (very important!), and the artists and groups of art that each customer tends to like.

(1) Draw the ER diagram for the database (10 marks)

(2) Represent the data in the ER diagram using CREATE TABLE statements (10 marks)

(3) If Artwork as a weak entity set with the partial key title and the owner entity set Artist, describe the changes needed in (1) and (2) (5 marks).

Answer:



CREATE TABLE Artist (

Name CHAR(20),

Birthplace CHAR(20),

Age INTERGER,

Style CHAR(20),

PRIMARY KEY (name))

CREATE TABLE Artwork-Paints (/*merge the Paints information into Artwork */

Year INTEGER,

Title CHAR(20),

Type CHAR(20),

Price REAL

Name CHAR(20) NOT NULL

PRIMARY KEY (title),

FOREIGN KEY (name) REFERENCES Artist)

CREATE TABLE Group (

Name CHAR(20),

```
PRIMARY KEY (name))

CREATE TABLE Customer (
Cust_id INTEGER,
Name CHAR(20),
Address CHAR(20),
Amount REAL,
PRIMARY KEY (cust_id))

CREATE TABLE Like_Artist (
Name CHAR(20),
Cust_id INTEGER,
PRIMARY KEY (name, cust_id),
FOREIGN KEY (cust_id) REFERENCES Customer,
FOREIGN KEY (name) REFERENCES Artist)

CREATE TABLE Like_Group (
Name CHAR(20),
Cust_id INTEGER,
PRIMARY KEY (name, cust_id),
FOREIGN KEY (cust_id) REFERENCES Customer,
FOREIGN KEY (name) REFERENCES Group)

CREATE TABLE Classify (
Name CHAR(20),
Title CHAR(20)
PRIMARY KEY (name,title),
FOREIGN KEY (name) REFERENCES Group,
FOREIGN KEY (title) REFERENCES Artwork-Paints)
```

(3)

For (1), the rectangle of Artwork and the diamond of Paints will be darked, title of Artwork will be underlined by a dashed line. For (2), the primary key of Artwork-Paints now is (title,name) instead of title, which leads to the following changes in Artwork-Paints and Classify.

CREATE TABLE Artwork-Paints (/*merge the Paints information into Artwork */

Year INTEGER,

Title CHAR(20),

Type CHAR(20),

Price REAL

Name CHAR(20)

PRIMARY KEY (title, name),

FOREIGN KEY (name) REFERENCES Artist)

CREATE TABLE Classify (

Name CHAR(20),

Name_artist CHAR(20),

Title CHAR(20)

PRIMARY KEY (name, title, name_artist),

FOREIGN KEY (name) REFERENCES Group,

FOREIGN KEY (title, name_artist) REFERENCES Artwork-Paints)

TA's notes: