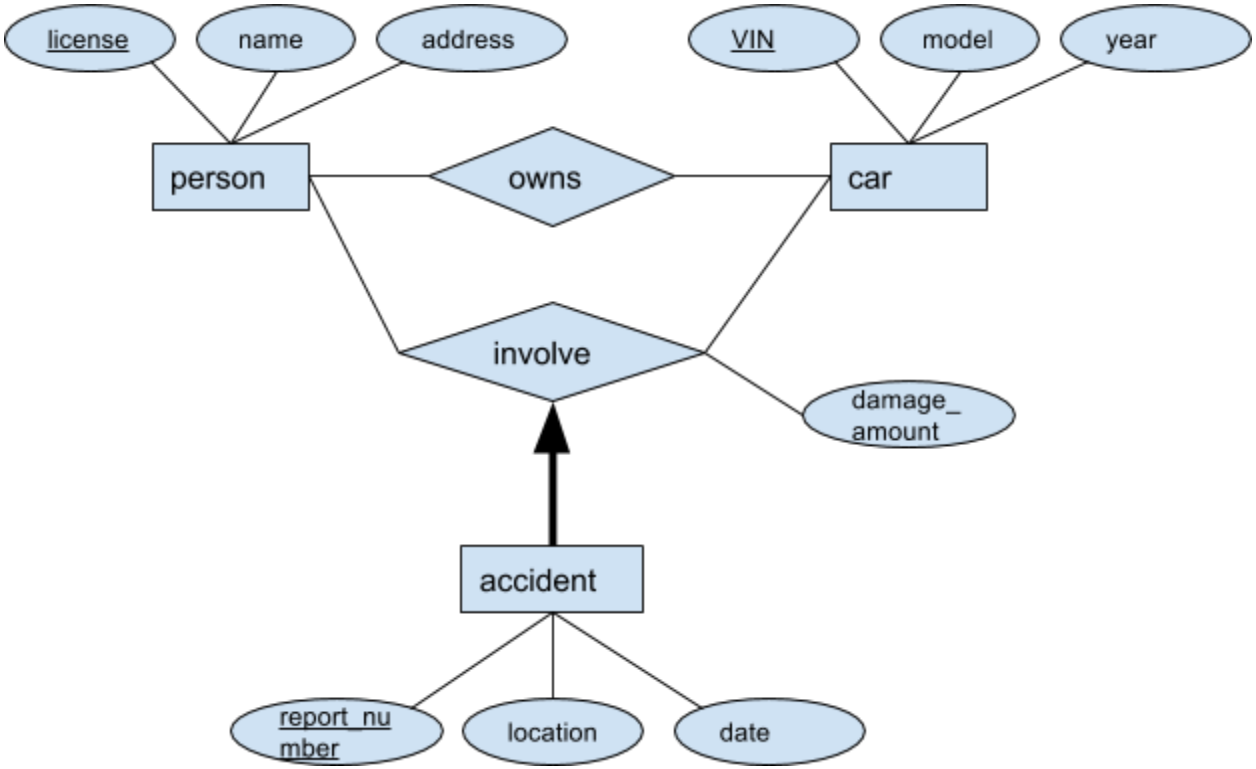
Homework 1 Solution

Discussion Session 3

Question 1

- Consider the following ER diagram



Question 1(a)

Q) If the company would like to make sure a car must be owned by exactly one person and a client should have at least one car. What modifications are needed?

A)

- *Key and participation constraint between car and owns, arrow points from car to owns*
- *Participation constraint between person and owns*

Question 1(b)

Q) What potential design problem do you observe regarding the ternary relationship “involve” and how would you rectify it? Make sure you explain your answer.

A)*It is likely an accident involves multiple person and cars, but bold arrow between accident and involve restricts only one person and car can be involved. Should change it to participation constraint, bold line between accident and involve*

Question 2

Consider a relation R with attributes A, B, and C. Suppose that you are given the following instance (i.e a snapshot) of this relation:

A	B	C
1	a	Sophia
2	f	Noah
3	c	Jacob
4	d	Isabella
1	e	Emily
5	f	Logan
6	c	Jacob

Which of the following attribute sets can you conclude are NOT keys for R? For each attribute set that is not a key, please explain why not.

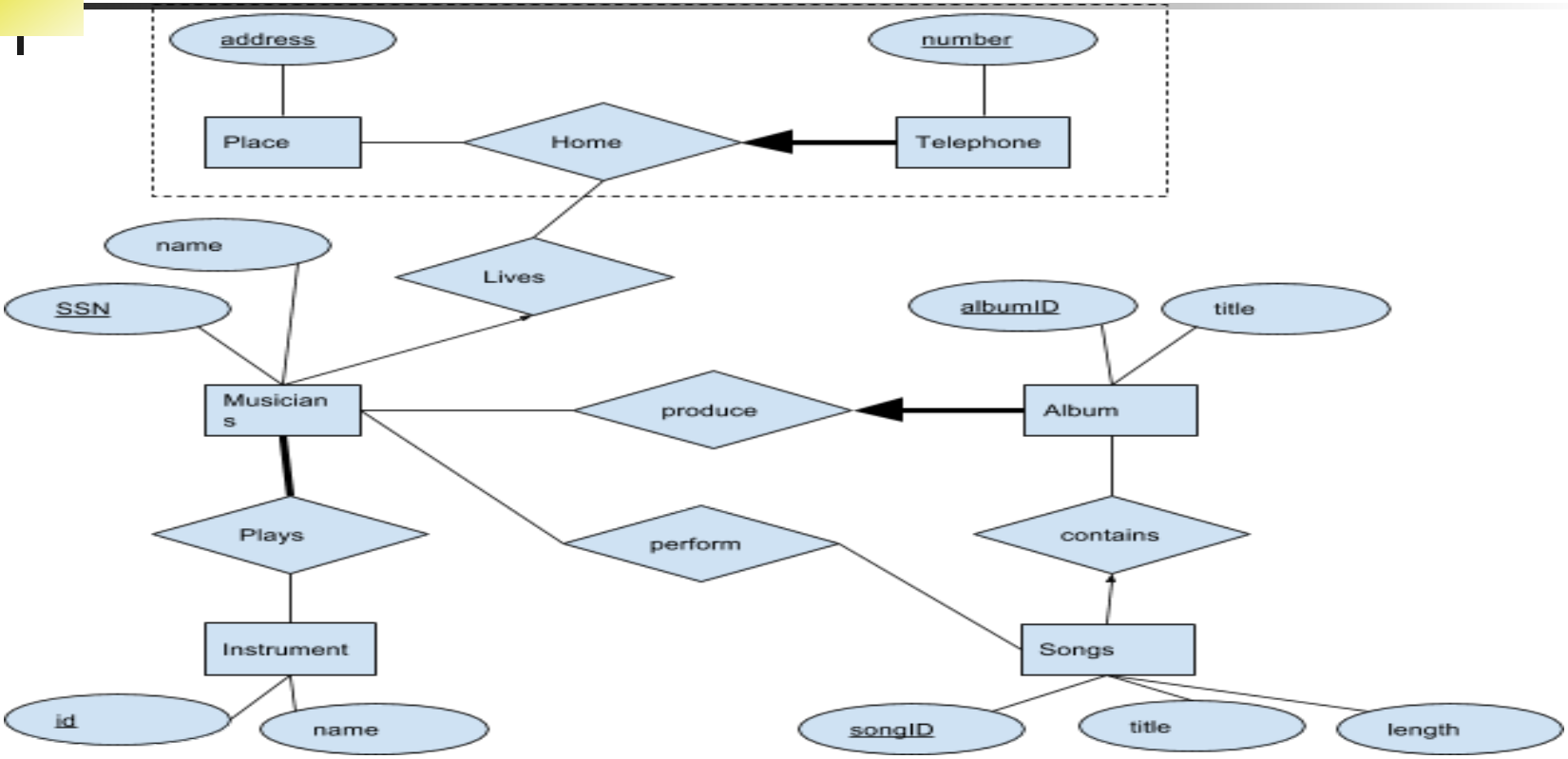
A; B; C; AB; AC; BC; ABC

Question 2 - Answer

A	B	C
1	a	Sophia
2	f	Noah
3	c	Jacob
4	d	Isabella
1	e	Emily
5	f	Logan
6	c	Jacob

A: not key, because not unique
B: not key, because not unique
C: not key, because not unique
AB: cannot tell
AC: cannot tell
BC: not key, because not unique
ABC: cannot tell, if AB or AC is key, then ABC is not key since it is not minimal

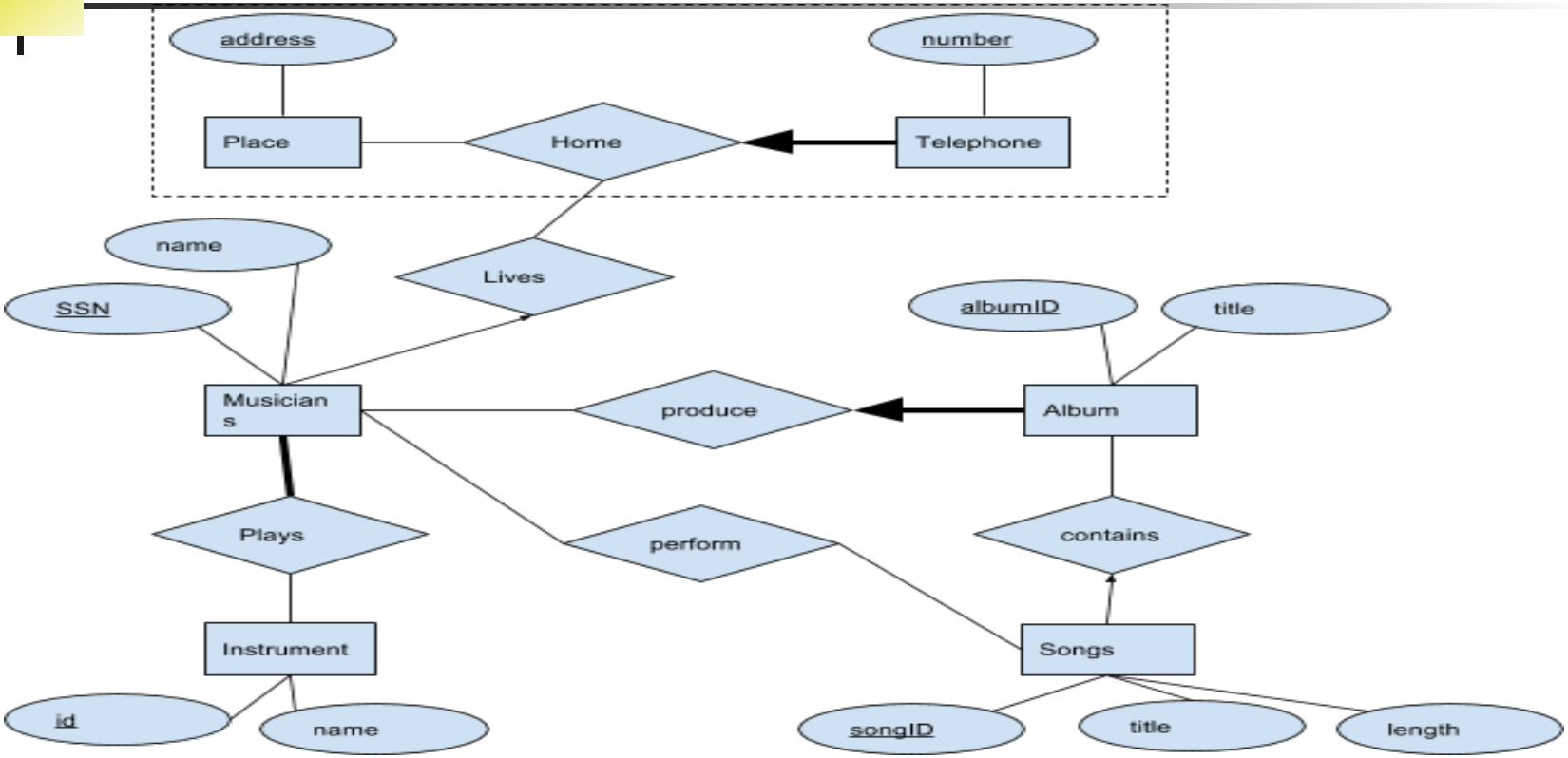
Question 3(a)



Suppose you have noticed that musicians often keep multiple addresses due to touring. Does this ER diagram allow musicians to have multiple addresses? Why or why not?

A) *This design doesn't allow multiple addresses.*
Reason: there is a key constraint between musician and lives, it allows at most one address.

Question 3(b)

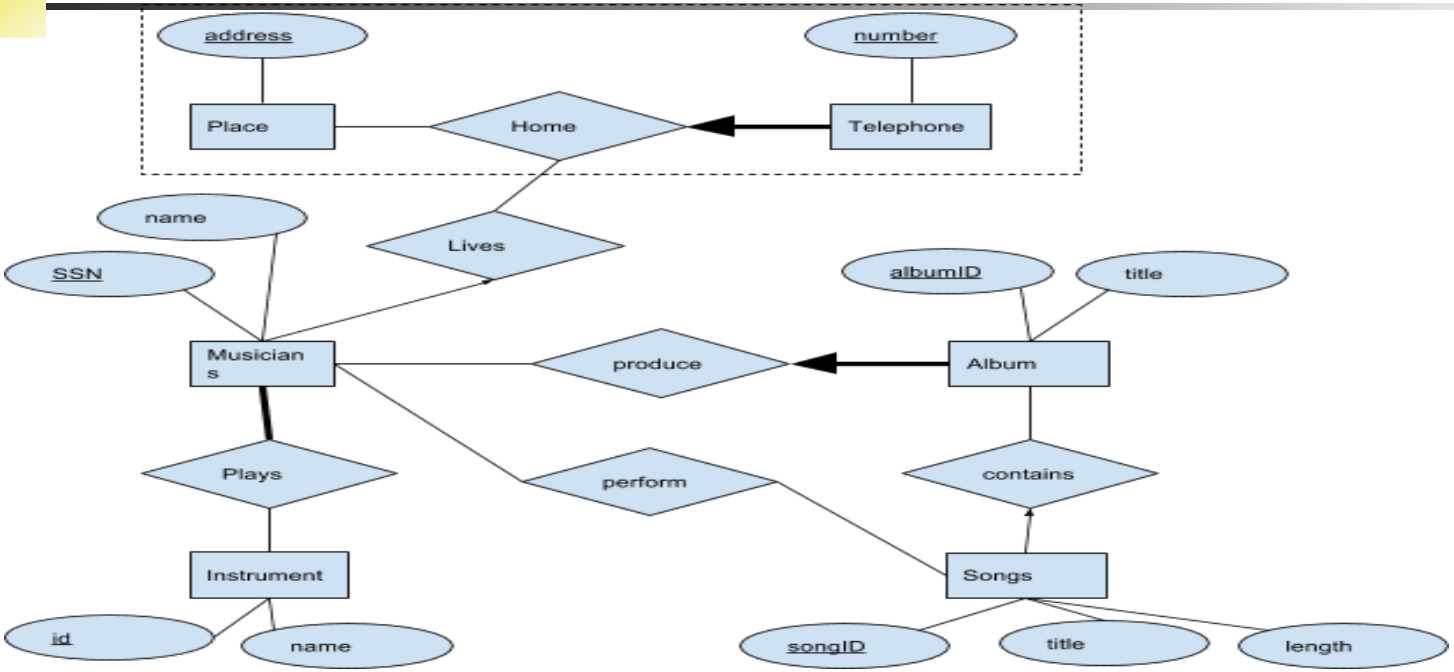


Suppose musicians sometimes collaborate with each other to produce an Album, does this design allow this? Why or why not?

A) *This design doesn't allow this.*

Reason: the key and participation constraint between Album and produce only allow an Album to map to one musician

Question 3(c)



Suppose you want to enforce a constraint that every musician should produce at least one album and when musician produce the Album, at least one instrument should be used. Does this design capture this constraint? If not, how would you do it?

A) This design does not capture this. It should be ternary relationship between "musician", "instrument", "album" and "produce". and bold line between "musician" and "produce".

Question 4

As the CDO (Chief Database Officer) of Sahara - world's largest online bookstore, you are tasked to design a new database to keep track of important data pertaining to books, authors, publishers, warehouses and customers. Unfortunately, all your employees are either on vacation or working on other projects, you have to do this by yourself. In the design, you need to capture the following:

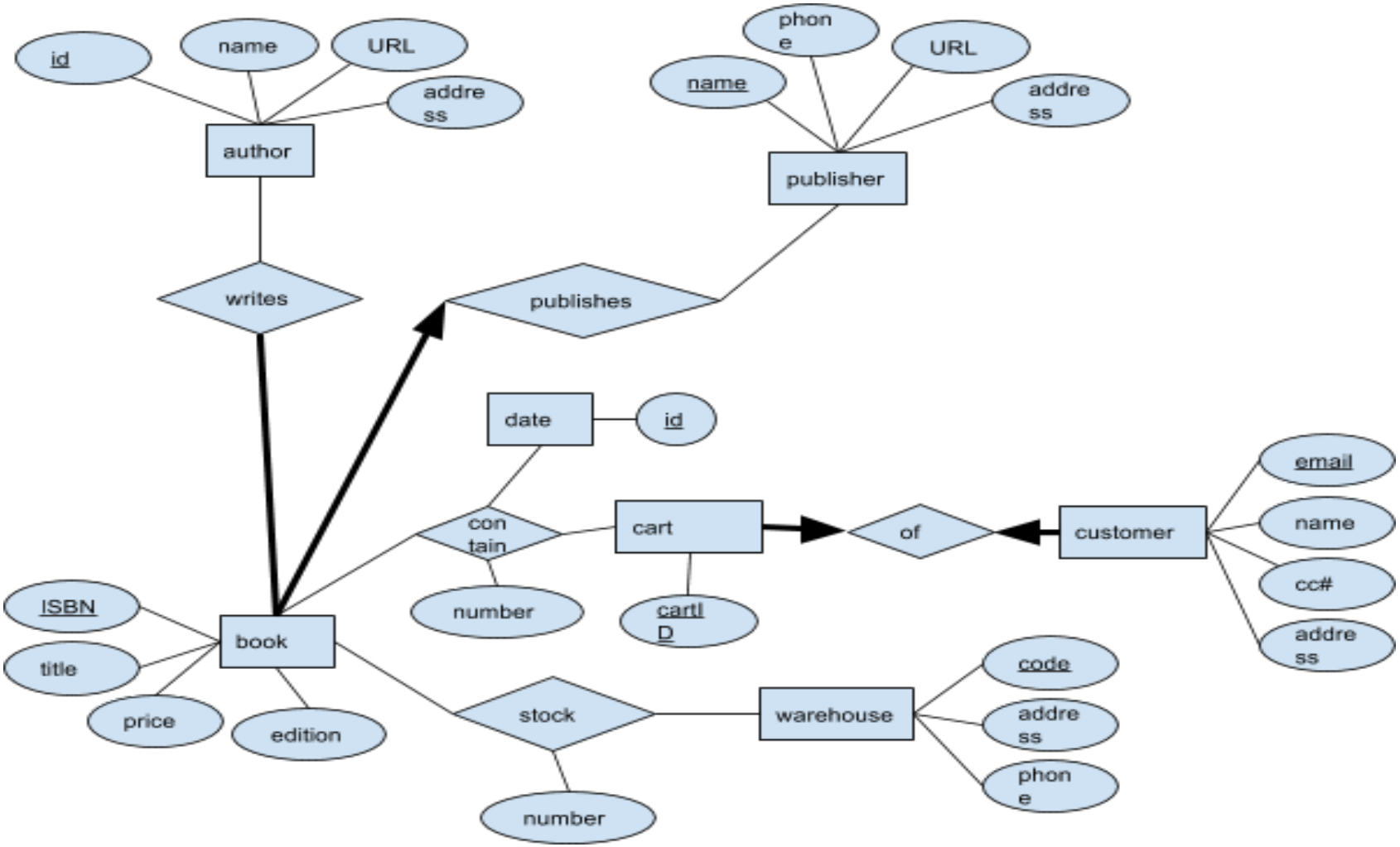
Each book have: ISBN (assume same ISBN for same book, different books will have different ISBN), title, edition and price

You also need to store information of author and publisher for each book. An author has an unique author ID, name, personal website URL and address. A publisher have name, phone number, address and website URL.

Each book has at least one author, possibly multiple authors and only one publisher.

As a valued employee of Sahara, you always keep the company's revenue in mind. In order to sell more books to customers, the database need to store the following information:
Each customer has an email address which also serves as their unique id, name, credit card number and address.
To determine which customer to send ads to based on their purchase history, the database needs to store a customer's shopping cart information:
each customer has exactly one cart and one cart belongs to exactly one customer; each cart has a cart id.
each cart contains a customer's purchase history: purchased book, number of copies, date of purchase. Note that it is possible a customer purchase the same book multiple times at different dates.
Since Sahara is the biggest online bookstore, it has warehouses all around the world. Each warehouse has: a unique code, address and phone number.
Keep track of books stored in each warehouse and the number of copies.

Question 4 - Answer

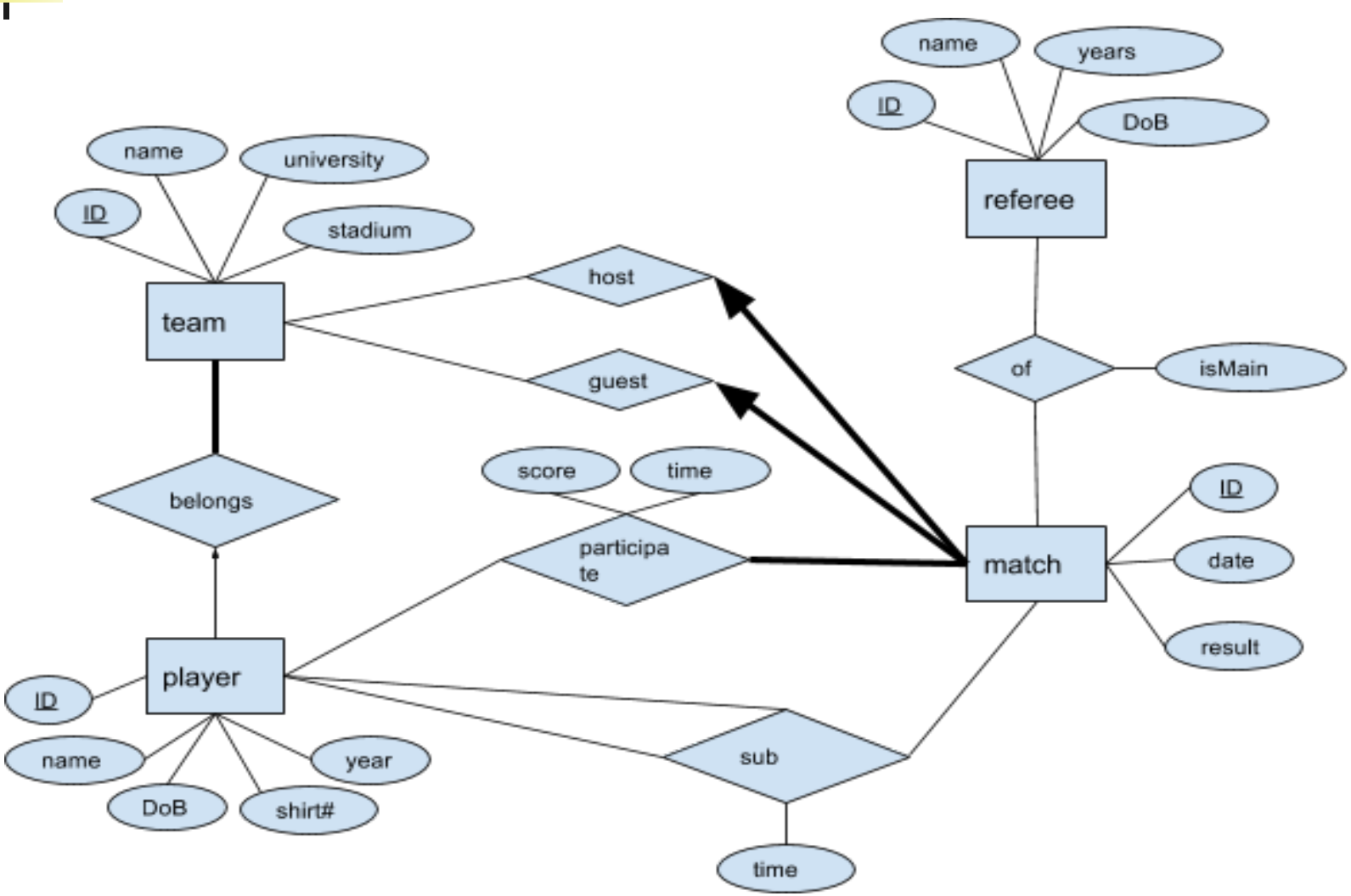


Question 5

Assume you are hired to design a database to model university football teams, the games they play, and the players. In the design, you need to capture the following:

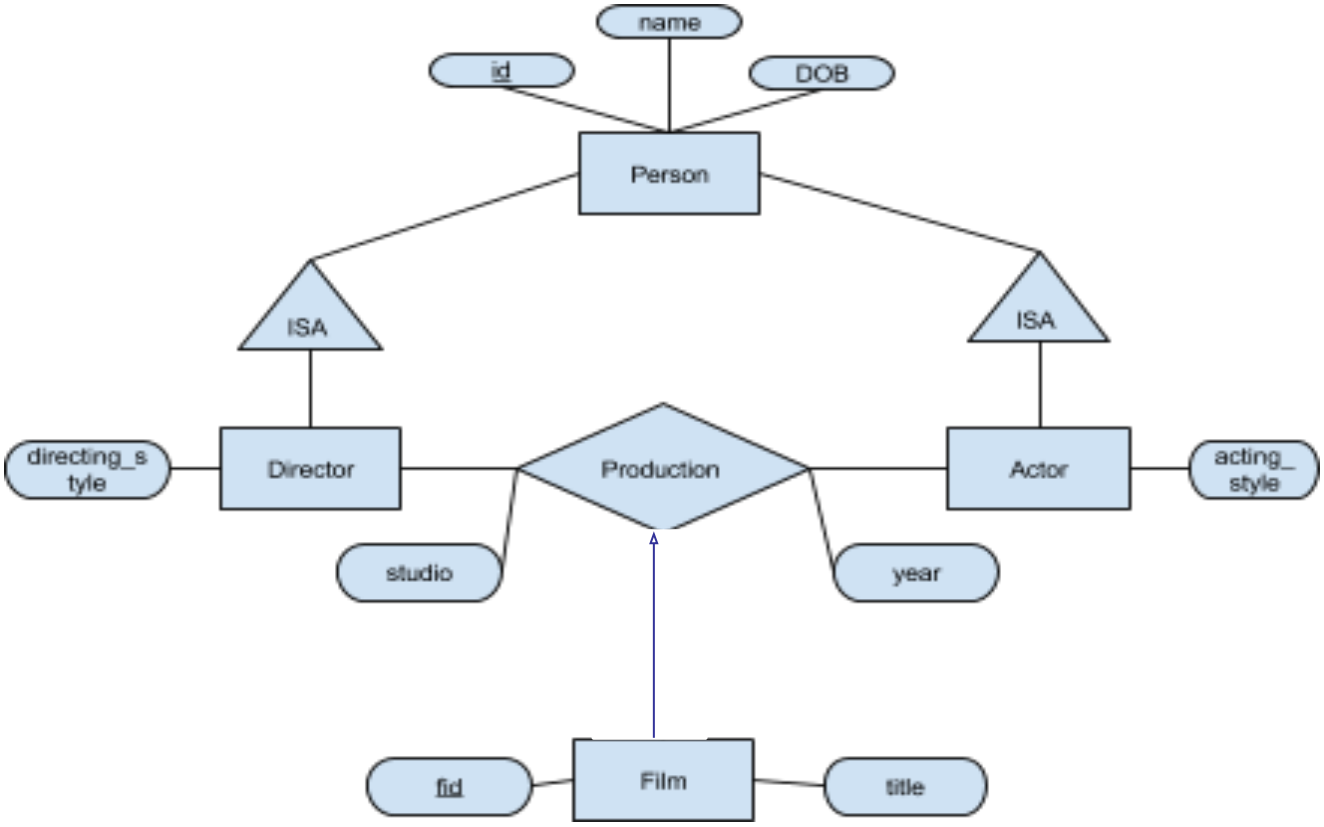
There are several teams: each team has an ID (unique identifier), name, main stadium and university to which this team belongs.
Each team has many players and each player belongs to at most one team. Each player has a unique identifier, name, DoB and shirt number that he uses. *It is possible that a player does not belong to any team.*
Teams play matches: in each match there is a host team and a guest team.
For each match we need to keep track of the following:
Match ID, Match date, Final result of the match
Players participated in the match. For each player, store his scores and total time of playing
During the match, one player may substitute another player. We'd like to capture this substitution and the time at which it took place. Assume that a player can only substitute the same player once in a match (A can substitute B only once in one match, but A can substitute other players in the same match and be substituted too).
Each match has multiple referees. For each referee we have an ID (unique identifier), name, DoB, years of experience.

Question 5 - Answer



Check your understanding

Translate the following ER diagram to relational tables and provide appropriate DDL(i.e. CREATE TABLE) statements. If there is more than one way for translation, provide all the possible translations.



Solution

```
CREATE TABLE Person
(
id INTEGER,
name VARCHAR(20),
dob  DATE or VARCHAR,
PRIMARY KEY(id)
);
```

```
CREATE TABLE Director
(
id INTEGER NOT NULL,
directing_style VARCHAR(20),
PRIMARY KEY(id),
FOREIGN KEY(id) References Person(id)
);

CREATE TABLE Actor
(
id INTEGER NOT NULL,
acting_style VARCHAR(20),
PRIMARY KEY(id),
FOREIGN KEY(id) References Person(id)
);
```

One table design

```
CREATE TABLE Film
(
fid INTEGER,
director_id INTEGER,
actor_id INTEGER,
title VARCHAR(20),
studio VARCHAR(20),
year VARCHAR(20),
PRIMARY KEY(fid),
FOREIGN KEY(actor_id) REFERENCES Actor(id),
FOREIGN KEY(director_id) REFERENCES Director(id)
);
```

Two table design

```
CREATE TABLE Film
{
fid INTEGER,
title VARCHAR(20),
PRIMARY KEY(fid),
}
```

```
CREATE TABLE Production
{
fid INTEGER,
director_id INTEGER,
actor_id INTEGER,
studio VARCHAR(20),
year VARCHAR(20),
PRIMARY KEY(fid),
FOREIGN KEY(fid) REFERENCES Film.
fid,
FOREIGN KEY(actor_id) REFERENCES
Actor.id,
FOREIGN KEY(director_id)
REFERENCES Director.id
}
```

Logging in to Oracle server

- module load oracle/muscle
  - Must be done every time you want to log in to the server
- If you add that line to your ~/.profile, that statement will be automatically executed every time you log in to CAEN and you will never have to type it again
  - For real: do this. Open/create your ~/.profile in your favorite text editor and add it.

Sqlplus Usability

- To get command line history: two options
  - rlwrap sqlplus
  - gqlplus
- Makes life better.

Some useful SQL operations

- Basic insert and select
  - Insert into table (field1) values (value1);
  - Select field1 from table where field1 > 2;
- Insert select
  - Insert into table1 (field1, field2)

    Select t2.field1, t3.field2

    From table2 t2, table3 t3

    Where t2.field3 = t3.field3;

Creating tables with select

- CREATE TABLE new_user AS (SELECT * FROM shravyak. public_user_information);

Constraint syntax

- Create table with constraints
  - Create table tname
    (field1 type1,
     field2 type2,
     field3 type3,
     constraint pk1 primary key (field1),
     constraint uniq23 unique (field2, field3)
    );