* Các khóa PK, FK
* Join
* Trigger
* Procedure
* Dạng câu 1 đề PE

* Thiết kế + đại số quan hệ
* FD

1 bảng chỉ có 1 khóa chính (primary key): tập các thuộc tính giúp xác định duy nhất 1 dòng trong bảng

--> 2 dòng khác nhau thì giá trị của khóa chính phải khác nhau

A(a1,a2,a3)

Create table A(

A1 int primary key,

A2 int,

A3 int,

)

Create table A(

A1 int,

A2 int,

A3 int,

primary key (a1,a2)

)

insert into A

values (1,1,1),

(1,2,1)

insert into A values (1,1,3)

alter table A

add primary key (a1,a2)

alter table A

add constraint PK\_A primary key (a1,a2)

-- foreign key: khóa tham chiếu

select \* from makers

select \* from products

insert into products(model, type)

values ('2008', 2)

select \* from products

update products

set maker = 'D'

where model = '2008'

create table B(

b1 int,

b2 int,

b3 int primary key,

foreign key (b1,b2) references A(a1,a2)

)

alter table B

add foreign key (b3) references A(a1)

select \* from B

insert into B(b1,b2,b3)

values (1,2,5)

-- Câu 1 đề PE: 2 điểm

Từ ERD tạo CSDL tương ứng với ERD đó

length 
to Stars 
Voices 
is 
title 
Movies 
Murder— 
Mysteries 

* Các thực thể: là các hình chữ nhật (dùng danh từ)
* Thuộc tính: hình oval
  + Thuộc tính có gạch chân: thuộc tính phải nằm trong khóa chính của thực thể hoặc mối quan hệ
* Mối quan hệ: hình thoi (dùng động từ)
* Thực thể yếu (weak-entity): hình chữ nhật có viền kép

Các quy tắc chuyển từ ERD sang bảng trong CSDL:

* Mỗi 1 thực thể không phải thực thể yếu -> 1 bảng

dept 
hkid 
PROF 
rank 

Tạo bảng PROF(pid, hkid, dept, rank, salary) với pid là khóa chính

* Mối quan hệ 1- 1:

MAN 
MARRY 
wid 
WOMAN 

Tạo các bảng:
MAN(mid, …) với mid là khóa chính
WOMAN(wid,…) với wid là khóa chính
mid là khóa ngoại của bảng MARRY tham chiếu đến MAN(mid)
wid là khóa ngoại của bảng MARRY tham chiếu đến WOMAN(wid)
1 người nam chỉ kết hôn với 1 người nữ
1 người nữ chỉ kết hôn với 1 người nam
-> khóa chính của bảng Marry là mid hoặc wid
Man
mid
…
M1

M2

M3

MARRY là mối quan hệ 1 - 1:
MARRY(mid,wid,year): có
Wid
wid
…
W1

W2

W3

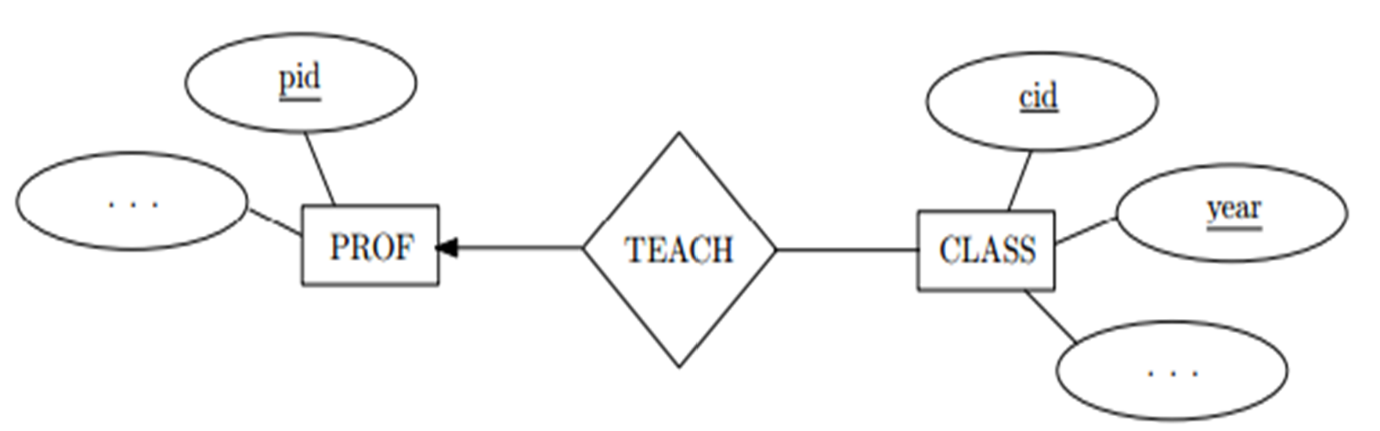
W4

Marry
mid
wid
year
M1
W2
2000
M2
W4
2001
M2
W3
2009



Ink Drawings


Quan hệ 1-M (one - many): 1 phía PROF, many phía class



Tạo các bảng:

* Bảng PROF(pid, …) với pid là khóa chính
* Bảng CLASS(cid,year,…, pid) với 1 khóa chính bao gồm 2 thuộc tính (cid, year)
  + Pid là khóa ngoại của CLASS tham chiếu đến PROF(pid)

Với quan hệ 1-M chúng ta không tạo thành bảng riêng mà đưa khóa chính của bảng bên phía 1 vào làm khóa ngoại của bảng phía Many

Quan hệ M-M (many - many):

TAKE 
ςτι: 

Taạo các bảng:

* Bảng STU(sid,…) với sid là khóa chính
* CLASS(cid,year,…) với 1 khóa chiính bao gồm 2 thuộc tính (cid,year)
* Bảng TAKE(sid, cid,year,grade):
  + Sid là khóa ngoọoại tham chiếu đến STU(sid)
  + (cid,year) là khóa ngọoại tham chiếu đến CLASS(cid,year)
  + Primary key (sid,cid,year)

IName 
hair 
times 
Name 
Presenter 
presents 
*Name 
station 
Show 
season 
represented 
rating 
prod by 
Agent 
ProdComp 
aN ame 
cName 
city 
phone 
cName 
n etW01th 
coun 

create table Agent(
aName nvarchar(50),
cName nvarchar(50),
city nvarchar(50),
phone nvarchar(50),
primary key (aName, cName)
)

create table ProdComp(
pcName nvarchar(50) primary key,
netWorth float,
country nvarchar(50)
)

create table Presenter(
SIN nvarchar(50) primary key,
fName nvarchar(50),
lName nvarchar(50),
hair nvarchar(50),
aName nvarchar(50),
cName nvarchar(50),
foreign key (aName, cName) references Agent(aName, cName)
)

create table Show(
sName nvarchar(50) primary key,
station nvarchar(50),
  
A screenshot of a computer

Description automatically generated  
A screenshot of a computer

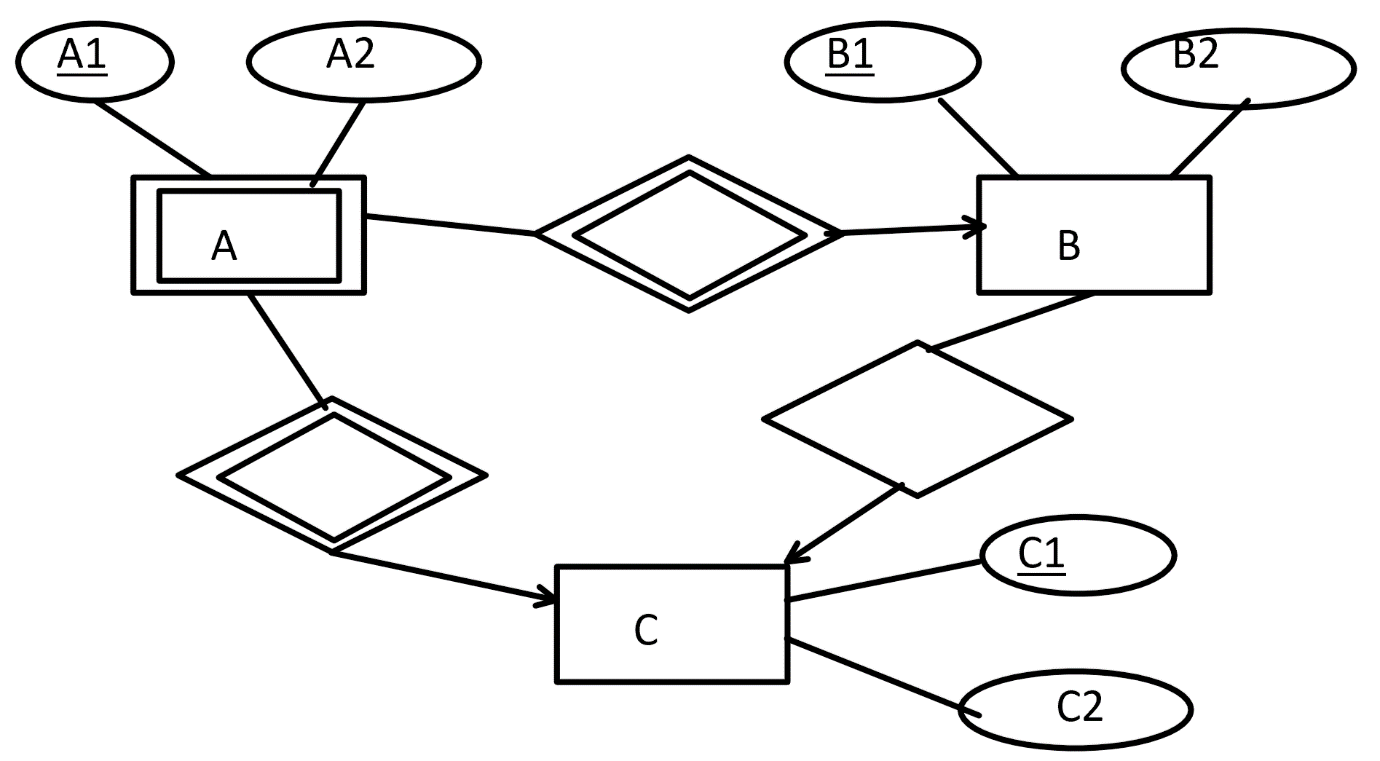
Description automatically generated

Cần tạo các bảng:

* Teams(name, address) với name là khóa chính
* Players(name, number, teamName) với:
  + teamName là khóa ngoại tham chiếu đến Teams(name)
  + Primary key (teamName, number)

--> với thực thể yếu cần có các thuộc tính của nó + khóa chính của các bảng mà nó phụ thuộc vào

--> khóa chính của thực thể yếu bao gồm khóa chính của bảng mà nó phụ thuộc vào + thuộc tính được gạch chân của thực thể yếu.



C(c1, c2) với C1 là khóa chính

B(B1, B2, C1) với B1 là khóa chính, C1 là khóa ngooại tham chiếu đến C(C1)

A(A1, A2, B1, C1) với:

* C1 là khóa ngooại tham chiếu đến C(C1)
* B1 là khóa ngooại tham chiếu đến B(B1)
* Khóa chính bao gồm (A1, B1, C1)

SubClass:

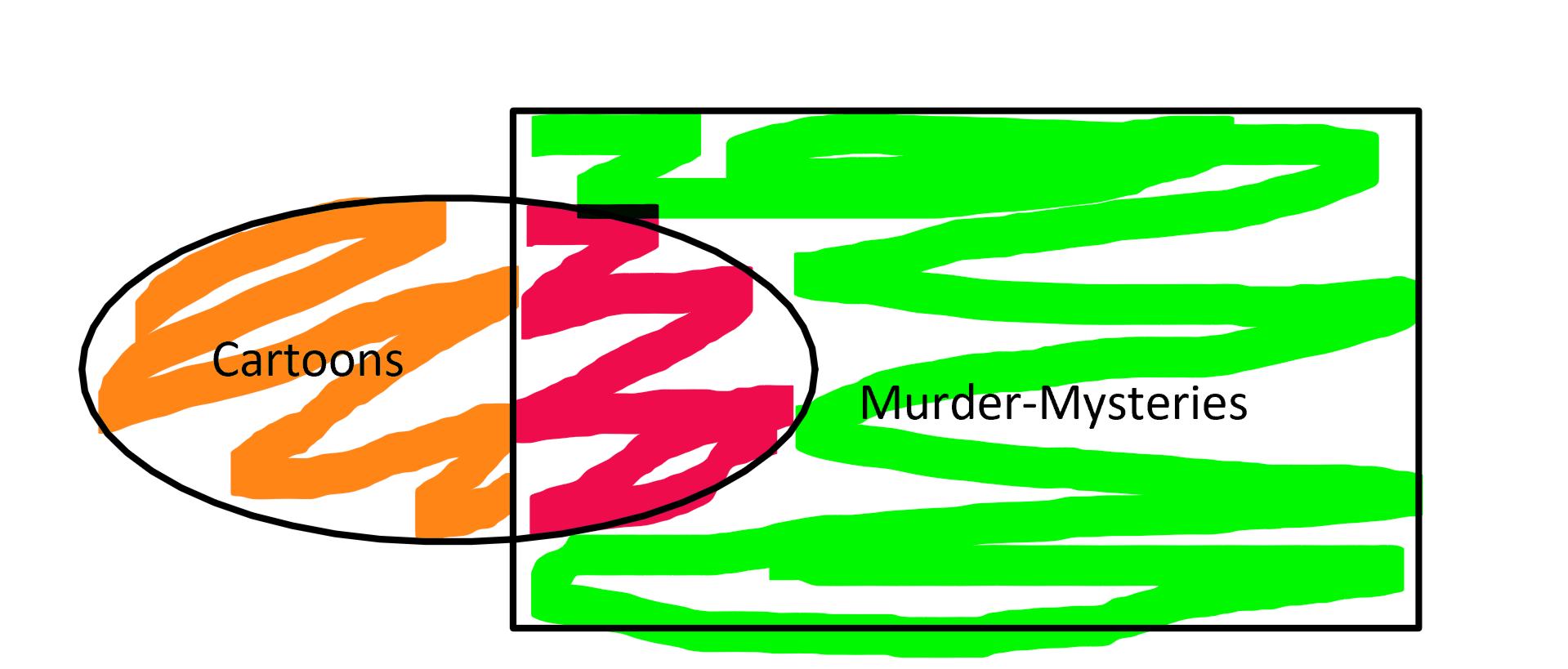
A diagram of a movie

Description automatically generated

**E/R style conversion**: tạo các bảng

* Movies(title, year, length, genre) với (title, year) là khóa chính chứa thông tin của tất cả các bộ phim kể cả các phim cartoons và murder-mysteries
* Murder-Mysteries(title, year, weapon) với:
  + (title, year) là khóa chính
  + (title, year) là khóa ngoại tham chiếu đến Movies(title,year)
* Cartoons(title, year) với:
  + (title, year) là khóa chính
  + (title, year) là khóa ngoại tham chiếu đến Movies(title,year)
* Voices(sName, title, year):
  + sName là khóa ngoại tham chiếu đến Stars(sName)
  + (title, year) là khóa ngoại tham chiếu đến Cartoons(title,year)
  + Khóa chính (sName, title, year)

**Object oriented approach**: mỗi bộ phim chỉ nằm trong 1 bảng



Tạo thành 4 bảng:

Cartoons(title, year, length, genre) với (title, year) là khóa chính -> chỉ là phim cartoon

Murder-mysteries(title, year, length, genre, weapon) với (title, year) là khóa chính -> chỉ là phim MM

Cartoon\_MM(title, year, length, genre, weapon) với (title, year) là khóa chính -> những phim vừa là cartoon vừa là murder-mysteries

Movies(title, year, length, genre) với (title, year) là khóa chính -> những phim còn lại

**Using Null values**: tạo 1 bảng

Movies(title, year, length, genre, weapon) với (title, year) là khóa chính

Inner join và outer join

A close-up of a label

Description automatically generated

Select \* from Class inner join Student on Class.Class\_ID = STUDENT.class\_ID :

chỉ giữ lại các dòng thoỏa mãn điều kiện join

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class.Class\_ID** | **Class\_Name** | **Sudent.Class\_ID** | **ID** | **Name** |
| 106 | Lop 106 | 106 | 1 | A |
| 106 | Lop 106 | 106 | 2 | B |
| 107 | Lop 107 | 107 | 3 | C |
| 107 | Lop 107 | 107 | 4 | D |

Outer join: cho phép giữ lại những dòng không thỏa mãn điều kiện join

**Select \* from Class left outer join Student on Class.Class\_ID = STUDENT.class\_ID**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class.Class\_ID** | **Class\_Name** | **Student.Class\_ID** | **ID** | **Name** |
| 106 | Lop 106 | 106 | 1 | A |
| 106 | Lop 106 | 106 | 2 | B |
| 107 | Lop 107 | 107 | 3 | C |
| 107 | Lop 107 | 107 | 4 | D |
| 201 | Lop 201 | Null | Null | Null |
| 202 | Lop 202 | Null | Null | Null |

Left outer join: giữ lại tất cả các dòng của bảng bên trái từ khóa join (Class)

-- giữ lại các dòng thỏa mãn điều kiện join (2 lớp 106, 107)

-- giữ lại các dòng của bảng bên trái không thoỏa mãn điều kiện join

**Select \* from Class right outer join Student on Class.Class\_ID = STUDENT.class\_ID**

Right outer join: giữ lại tất cả các dòng của bảng bên trái từ khóa join (Student)

-- giữ lại các dòng thỏa mãn điều kiện join (4 sinh viên A, B, C, D)

-- giữ lại các dòng của bảng bên phải không thoỏa mãn điều kiện join

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class.Class\_ID** | **Class\_Name** | **Student.Class\_ID** | **ID** | **Name** |
| 106 | Lop 106 | 106 | 1 | A |
| 106 | Lop 106 | 106 | 2 | B |
| 107 | Lop 107 | 107 | 3 | C |
| 107 | Lop 107 | 107 | 4 | D |
| Null | Null | Null | 5 | E |
|  |  |  | 6 | F |
|  |  |  | 7 | G |
|  |  |  | 8 | H |

**Select \* from Class full outer join Student on Class.Class\_ID = STUDENT.class\_ID**

Full outer join: giữ lại tất cả các dòng của cả 2 bảng

-- giữ lại các dòng thỏa mãn điều kiện join (4 sinh viên A, B, C, D)

-- giữ lại các dòng của bảng bên phải không thỏa mãn điều kiện join

-- giữ lại các dòng của bảng bên trái không thỏa mãn điều kiện join

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class.Class\_ID** | **Class\_Name** | **Student.Class\_ID** | **ID** | **Name** |
| 106 | Lop 106 | 106 | 1 | A |
| 106 | Lop 106 | 106 | 2 | B |
| 107 | Lop 107 | 107 | 3 | C |
| 107 | Lop 107 | 107 | 4 | D |
| Null | Null | Null | 5 | E |
| Null | Null | Null | 6 | F |
| Null | Null | Null | 7 | G |
| Null | Null | Null | 8 | H |
| 201 | Lop 201 | Null | Null | Null |
| 202 | Lop 202 | Null | Null | Null |

create table products(

maker nvarchar(10) references makers(code),

model nvarchar(10) primary key,

type int references [types](id)

)

**Products join makers on products.maker = makers.code**

**Products join types on products.type = types.id**