

CS3DB3/SE4DB3/ SE6DB3 TUTORIAL

YU HUANG
2015-01-21

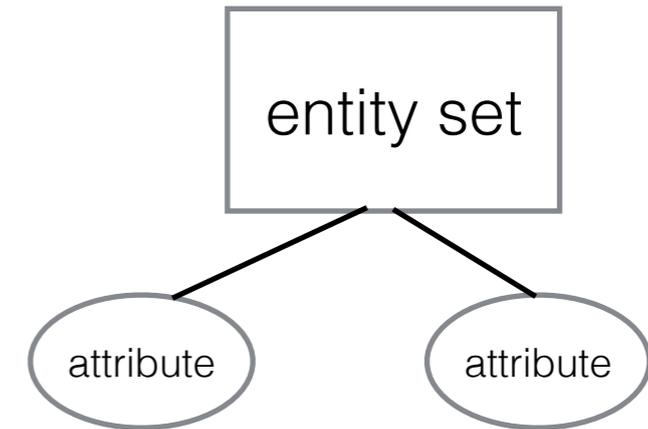
outline

- ER diagram
- Many-to-many
- Many-to-one
- One-to-one
- Access to DB2
- Upload file
- DB2 command

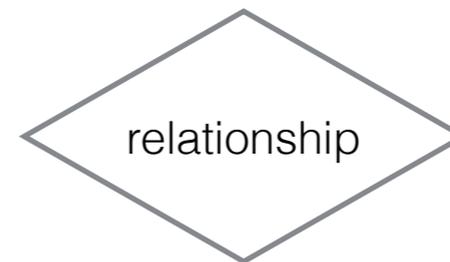
ER diagram

- E: Entity set, means “things” or object,
represented by rectangle

- attribute, property of entity set,
represented by oval

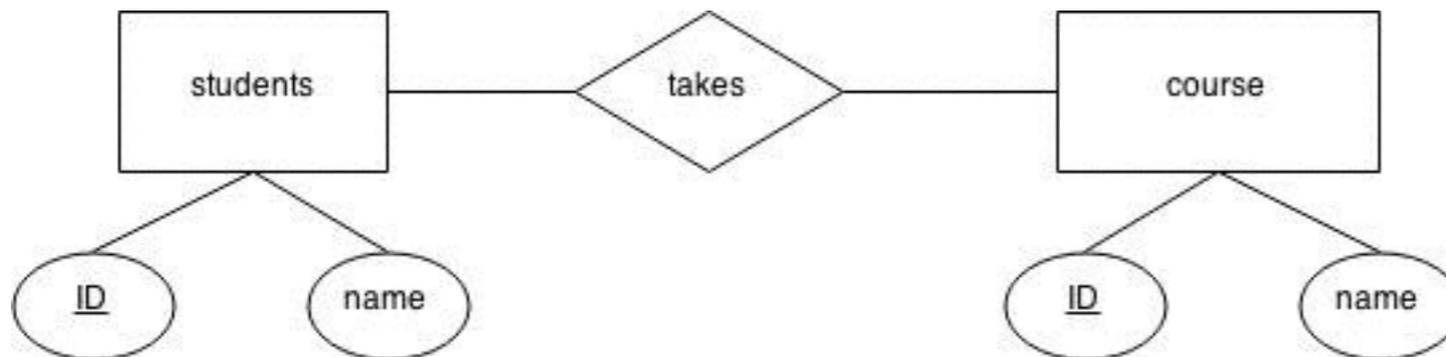


- R: relationship, connected two or more entity sets,
represented by diamond



Many-to-many

- students take courses



<u>ID</u>	name
101	Joe
102	Sue
103	Alex

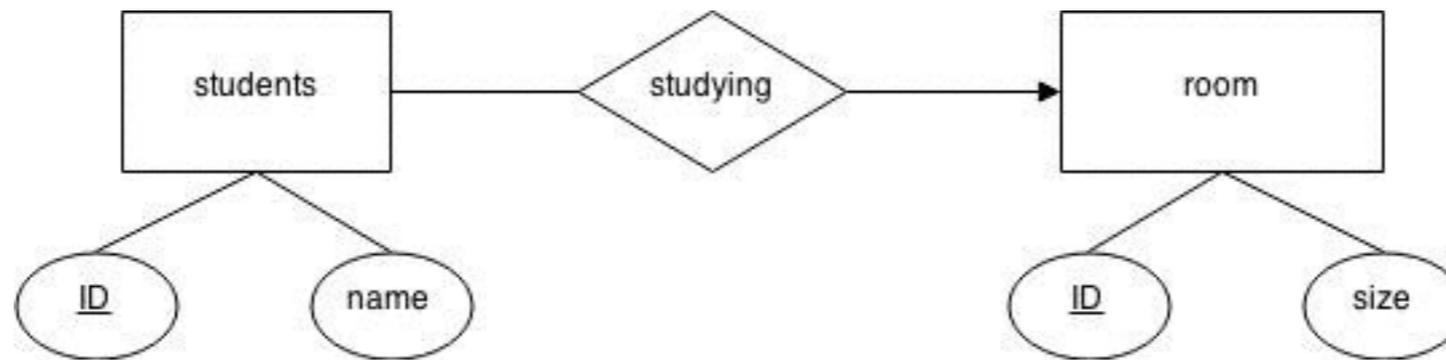
<u>studentID</u>	<u>courseID</u>
101	201
102	202
103	203

<u>ID</u>	name
201	database
202	JAVA
203	C++

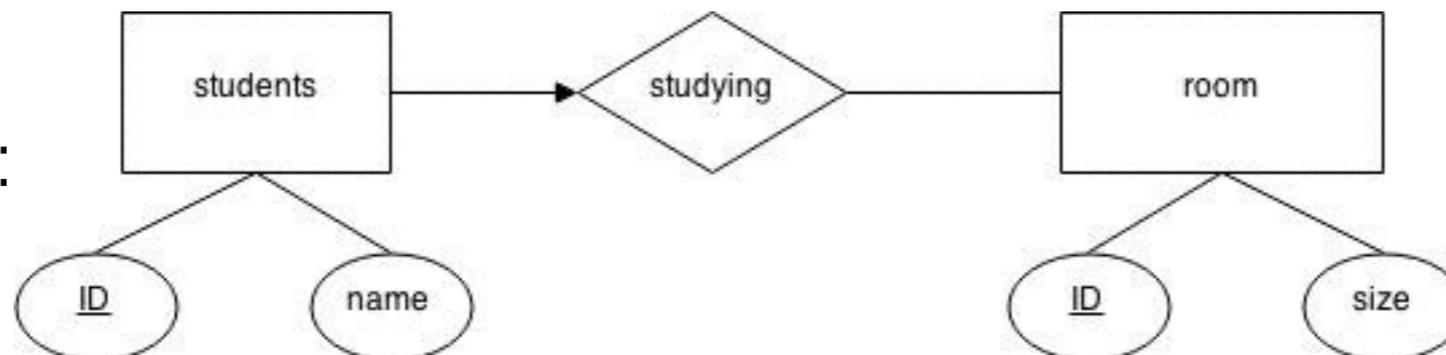
Many-to-one(partial participation)

- at most one (0 or 1), students are studying in room
- means some students are not studying in any room

slides:

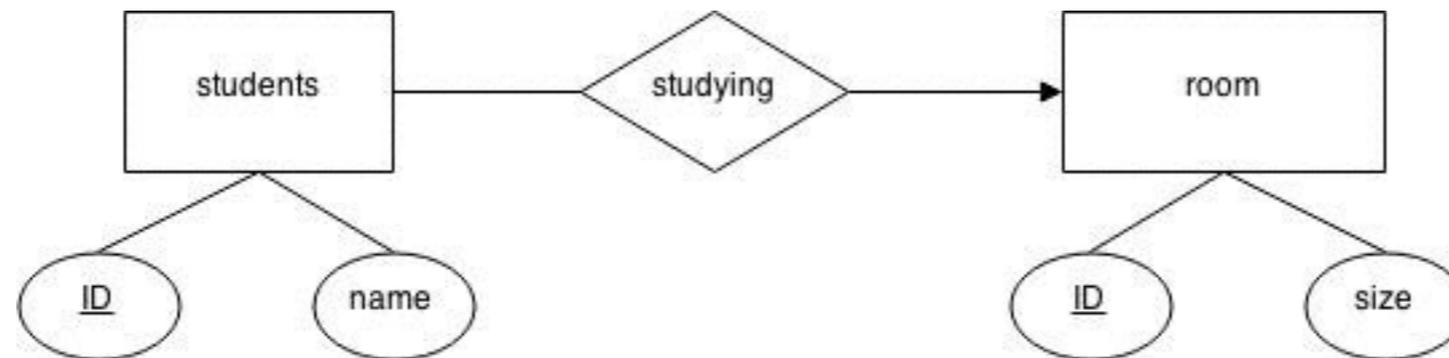


textbook:



- at most one (0 or 1) cont'

slides:



<u>ID</u>	name
101	Joe
102	Sue
103	Alex

<u>studentID</u>	<u>roomID</u>
101	201
102	201

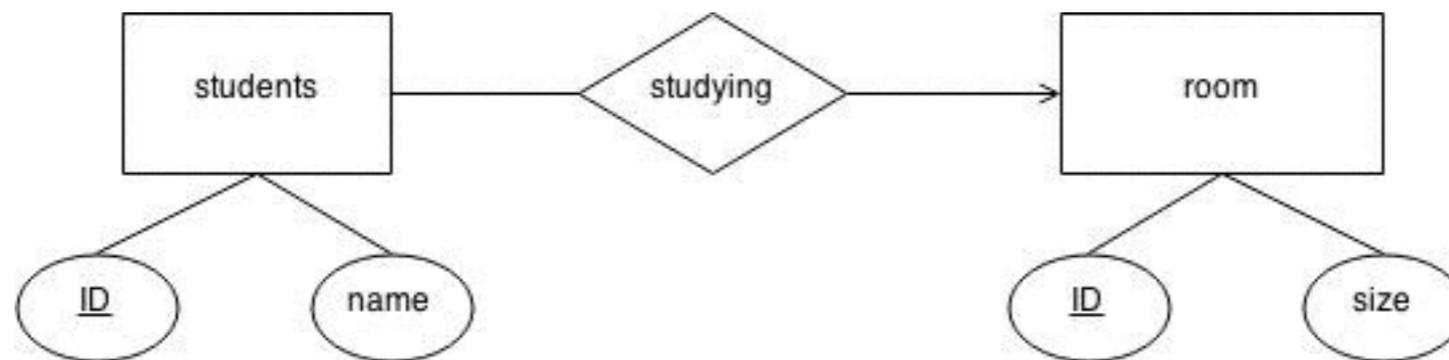
<u>ID</u>	size
201	20
202	50
203	100

Alex isn't studying in any room

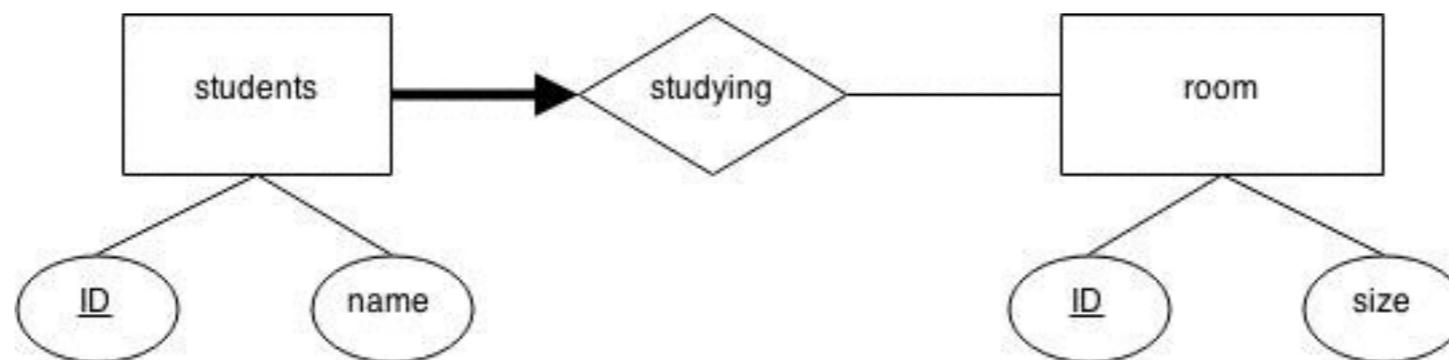
Many-to-one(total participation)

- exactly one (all student are studying in some room)

slides:

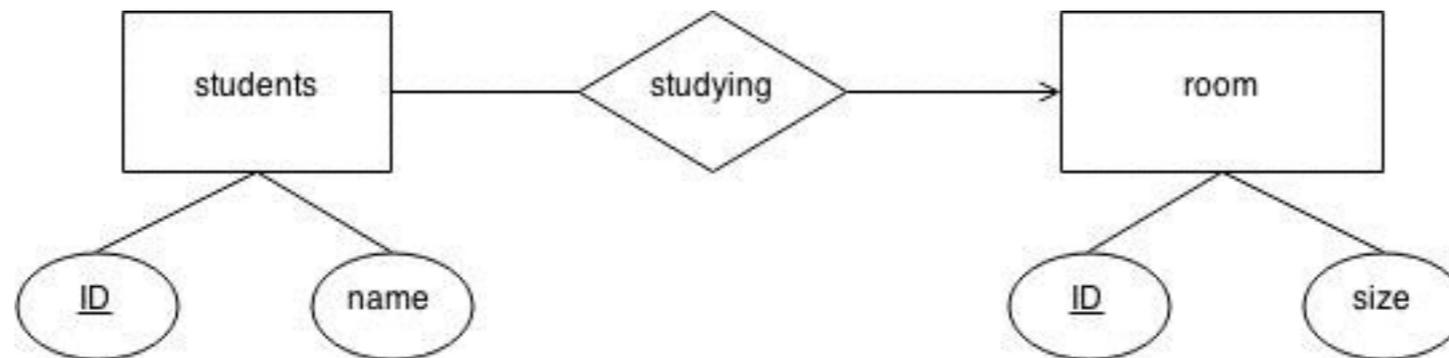


textbook:



- exactly one (all student are studying in some room)

slides:



<u>ID</u>	name
101	Joe
102	Sue
103	Alex

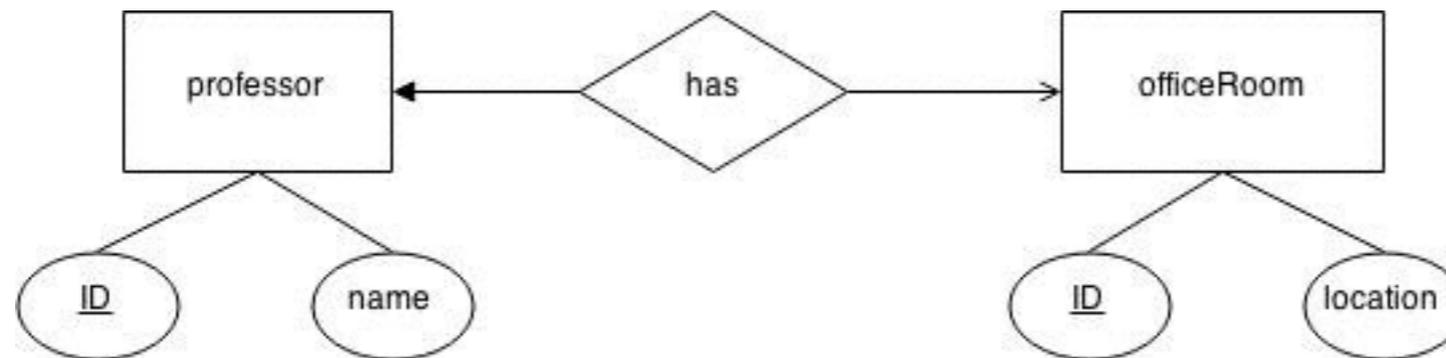
<u>studentID</u>	<u>roomID</u>
101	201
102	201
103	202

<u>ID</u>	size
201	20
202	50
203	100

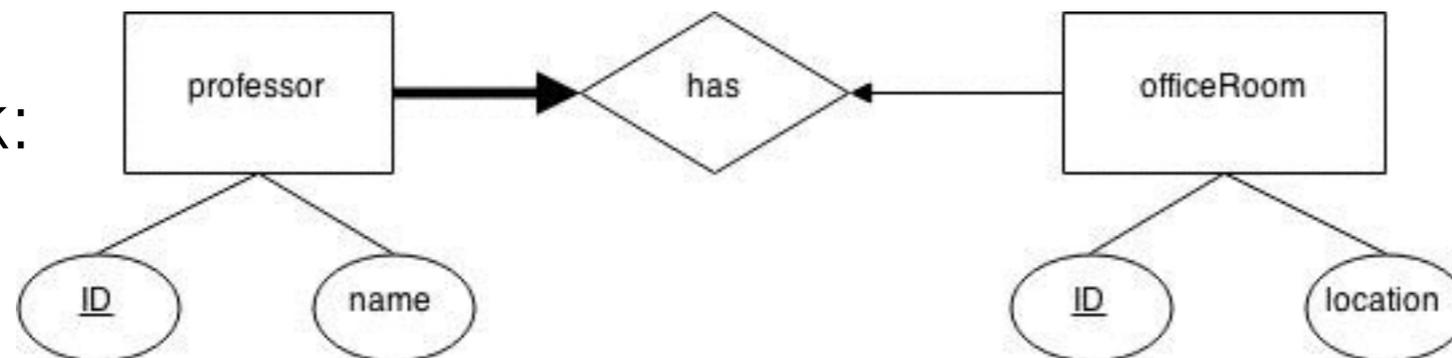
One-to-one(partial participation)

- professor has office room

slides:



textbook:

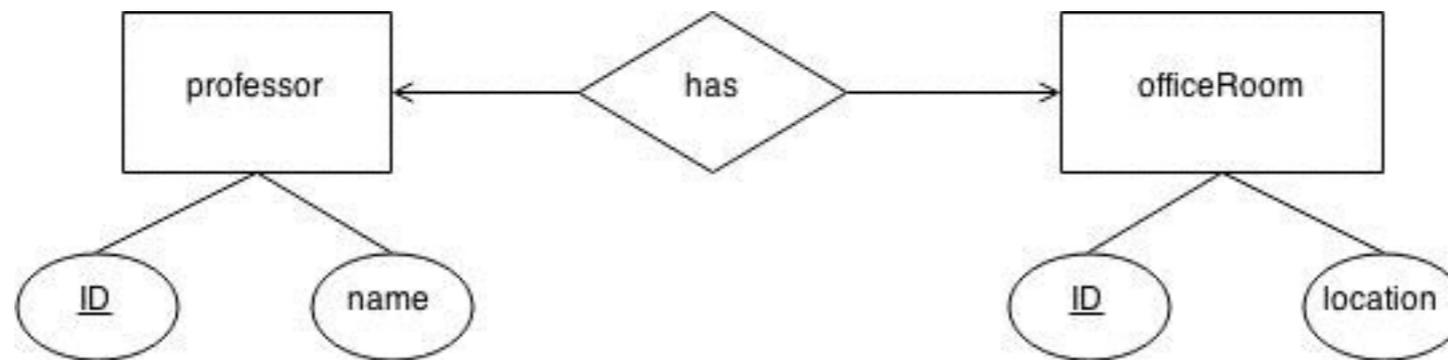


every professors has exactly one office room,
every office room has 0 or 1 professor (means
some office is empty).

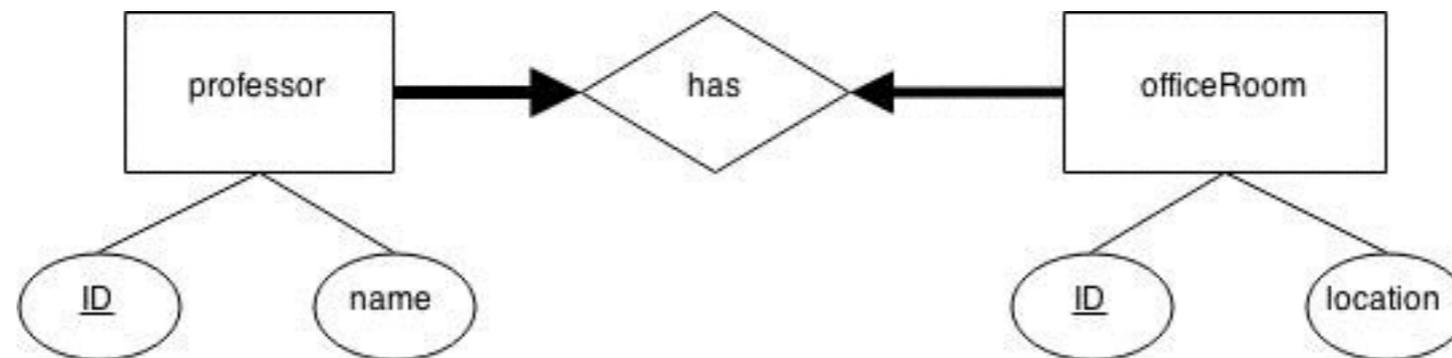
One-to-one(total participation)

- every professors has exactly one office room,
every office room has exactly one professor.
Means no office is empty.

slides:



textbook:



Access to DB2

- Windows

download the SSH software (putty or secureCRT)

<http://www.cas.mcmaster.ca/~fchiang/courses/db3/W15/help/ssh.html>

- Mac, linux etc

open Terminal, input `ssh`

Access to DB2

- two servers you can use
 - . CS3DB3 `ssh macid@db2srv2.mcmaster.ca`
 - . SE4DB3 `ssh macid@db2srv3.mcmaster.ca`

Tips:

1. please connect to the MacSecure WiFi, not MacConnect
2. Off-campus, please ssh to Mills first, then ssh to server from Mills (`ssh macid@mills.cas.mcmaster.ca`)
3. You don't need VPN !!

upload script

- Windows (use the software, very easy)
- Mac, Linux(use `scp` command)

`scp file's source path MACID@mills.mcmaster.ca:
destination path`

example:

```
scp /Users/you/script.txt MacID@mills.mcmaster.ca: /  
u20/somepath/
```

Basic command

- - - comment
- first line in your script

(login db2srv2.mcmaster.ca)

CONNECT TO CS3DB3;

- -your own code;

list tables;

Please don't include terminate;

Run Script

- run your script in server

```
db2 -tnf script.ddl
```

create table

```
CREATE TABLE Course(  
    cid CHAR(2) NOT NULL,  
    name CHAR(10),  
    PRIMARY KEY (cid)  
);
```

IMPORTANT:

1. Make sure you create all entity tables before you create any relationship tables !!!
2. Create tables that without foreign key before tables with foreign key!!!

view table

- view all tables

list tables

- view one specified table in detail

describe table tablename

Important tips

1. Make sure you create all entity tables **before**

you create any relationship tables !!!

2. Create tables that without foreign key **before** tables with foreign key!!!

3. when you write your ddl,

- - first two lines;

`connect to cs3db3;`

- -last line

`list tables;`

- - please do **NOT** include `Terminate;`

example code

```
1  -- connect to database;
2  connect to cs3db3;
3  --+++++
4  -- CREATE TABLES student, taking, course
5  --+++++
6
7  create table Student(
8      ID integer not null,
9      name varchar(10),
10     primary key (ID)
11 );
12
13 create table Course(
14     ID integer not null,
15     name varchar(10),
16     primary key (ID)
17 );
18
19 create table Taking(
20     studentID integer not null,
21     courseID integer not null,
22     primary key (studentID, courseID),
23     foreign key (studentID) references Student (ID) on delete cascade,
24     foreign key (courseID) references Course (ID) on delete cascade
25 );
26
27 list tables;
```

example code Drop table

```
-- connect to database;  
connect to cs3db3;  
--+++++  
-- DROP TABLES student, taking, course  
--+++++  
DROP table Student;  
DROP table Course;  
DROP table Taking;
```