

CSCI 403 Database Management

14 – Converting ERD to a SQL
Database

CS@Mines

ERD to SQL Summary

In general:

- Entities → tables
- 1:1 and 1:N relationships → Foreign keys
- N:M relationships → tables
- Multivalued attributes → tables

CS@Mines

2

7-Step Algorithm

Just follow the steps:

- Provided by book
- Some choices, but mostly deterministic
- Small examples along the way (you'll get to practice with a real schema later)

CS@Mines

3

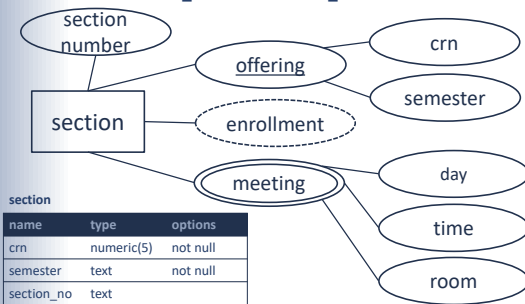
Step 1 – Regular Entities

- Regular entity → table
 - Use entity name or name of your choice
- All simple attributes → columns
 - No derived or multivalued
 - Only take components of a composite
 - Assign data types
 - Change name if desired
- Choose some key to be a primary key
- May get additional columns in later steps!

CS@Mines

4

Step 1 Example



section		
name	type	options
crn	numeric(5)	not null
semester	text	not null
section_no	text	

Primary key (crn, semester)

CS@Mines

5

Step 2 – Weak Entities

Like regular entities, but:

- Take primary key from owning entity's table:
 - Add to weak entity table
 - Make it a foreign key back to owning entity's table
- Since weak entity has no key, only a partial:
 - Make primary key a combination of partial key and "borrowed" key from owning entity

CS@Mines

6

Step 2 Example



name	type	options
room_no	text	not null
building_name	text	not null
...	...	...

Primary key (building_name, room_no)
Foreign key (building_name) references building (name)

CS@Mines

7

Step 3 – 1:1 Relationships

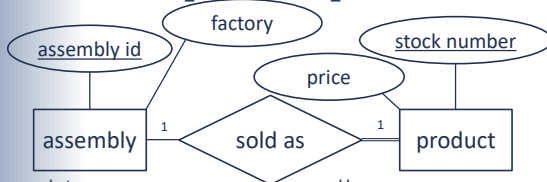
3 choices (depending somewhat on participation):

- Total participation on one or zero sides: create a foreign key
- Total participation on both sides: merge into one table
- Cross-reference table (discussed later - not recommended for 1:1)

CS@Mines

8

Step 3 Example 1



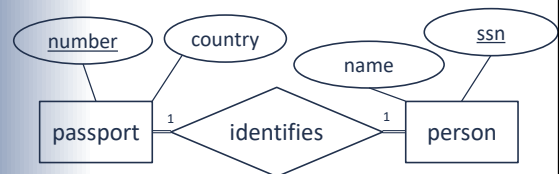
name	type	options
stock_no	text	primary key
assembly_id	integer	not null
price	numeric	
...	...	...

Foreign key (assembly_id) references assembly (id)

CS@Mines

9

Step 3 Example 2



name	type	options
ssn	integer	primary key
Name	text	
passport_no	integer	not null, unique
country	text	
...	...	...

CS@Mines

10

Step 4 – 1:N Relationships

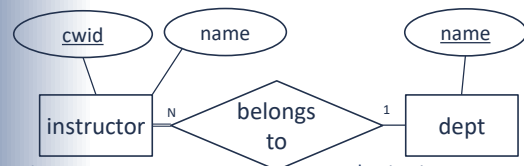
2 choices:

- Take primary key from "one" side:
 - Add to "many" side table
 - Create foreign key
- Cross-reference table (not recommended)

CS@Mines

11

Step 4 Example



name	type	options
cwid	integer	primary key
name	text	
department_name	text	not null
...	...	...

Foreign key (department_name) references department (name)

CS@Mines

12

Step 5 – N:M Relationships

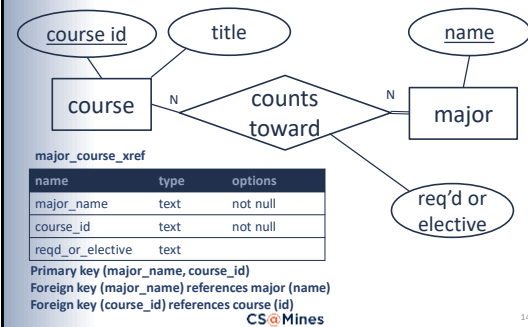
Only choice: cross-reference (xref) table

- Neither side can properly reference the other using a foreign key
- Create a new table whose entries represent connections between entries in the two tables
 - New table borrows primary key from each table
 - Each borrowed key is a foreign key back to original
 - Xref table also place for relationship attributes

CS@Mines

13

Step 5 Example



CS@Mines

14

Step 6 – Multivalued Attributes

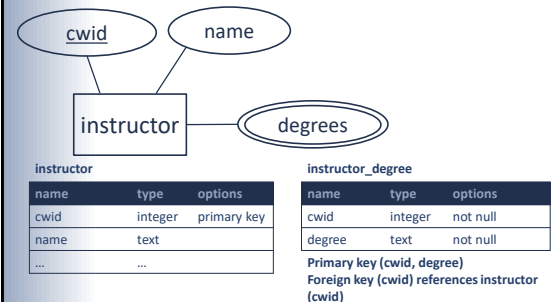
Treat essentially like weak entity

- Make a table for attribute
- In place of partial key, use attribute
- Borrow primary key of owning table
- Make borrowed key and attribute together the primary key
- Make foreign key back to owning table

CS@Mines

15

Step 6 Example



CS@Mines

16

Step 7 – N-ary Relationships

Only choice: cross-reference table

As for N:M relationships, but using primary keys from all involved tables.

CS@Mines

17

Up Next

- Next lecture:
 - Relational algebra
- Reading: Chapter 8:1 – 8.5: “The Relational Algebra and Relational Calculus”
- Coming soon:
 - Quiz on ERD

CS@Mines

18