CSCI 403
Database Management
14 – Converting ERD to a SQL Database

ERD to SQL Summary
In general:
- Entities → tables
- 1:1 and 1:N relationships → Foreign keys
- N:M relationships → tables
- Multivalued attributes → tables

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)

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!

Step 1 Example

<table>
<thead>
<tr>
<th>section number</th>
<th>offering</th>
<th>crn</th>
<th>semester</th>
<th>enrollment</th>
<th>day</th>
<th>time</th>
<th>meeting</th>
<th>room</th>
</tr>
</thead>
<tbody>
<tr>
<td>section</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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
Step 2 Example

```
building 1 contains N room
```

<table>
<thead>
<tr>
<th>name</th>
<th>type</th>
<th>options</th>
</tr>
</thead>
<tbody>
<tr>
<td>room_no</td>
<td>text</td>
<td>not null</td>
</tr>
<tr>
<td>building_name</td>
<td>text</td>
<td>not null</td>
</tr>
</tbody>
</table>

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

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)

Step 3 Example 1

```
assembly_id 1 assembly

factory

stock number

price

sold as 1 product

assembly

<table>
<thead>
<tr>
<th>name</th>
<th>type</th>
<th>options</th>
</tr>
</thead>
<tbody>
<tr>
<td>stock_no</td>
<td>text</td>
<td>primary key</td>
</tr>
<tr>
<td>assembly_id</td>
<td>integer</td>
<td>not null</td>
</tr>
<tr>
<td>price</td>
<td>numeric</td>
<td></td>
</tr>
</tbody>
</table>

Foreign key (assembly_id) references assembly(id)
```

Step 3 Example 2

```
number

country

name

ssn

passport

identifies 1 person

<table>
<thead>
<tr>
<th>name</th>
<th>type</th>
<th>options</th>
</tr>
</thead>
<tbody>
<tr>
<td>ssn</td>
<td>integer</td>
<td>primary key</td>
</tr>
<tr>
<td>passport_no</td>
<td>integer</td>
<td>not null, unique</td>
</tr>
<tr>
<td>country</td>
<td>text</td>
<td></td>
</tr>
</tbody>
</table>

person
```

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)

Step 4 Example

```
cwid

name

instructor N belongs to 1 dept

<table>
<thead>
<tr>
<th>name</th>
<th>type</th>
<th>options</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>integer</td>
<td>primary key</td>
</tr>
<tr>
<td>name</td>
<td>text</td>
<td></td>
</tr>
</tbody>
</table>

Foreign key (department_name) references department(name)
```
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

Step 5 Example

```
course id | title | name
-----------|-------|------
course     | N     | major
counts toward N
major_course_xref
name | type | options
major_name | text | not null
course_id | text | not null
reqd_or_elective | text
```

Primary key (major_name, course_id)
Foreign key (major_name) references major (name)
Foreign key (course_id) references course (id)

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

Step 6 Example

```
cwid | name | degree
----- | ---- | ----
    |     | degrees
```

```
instructor
cwid | name | ...
integer | text | ...
instructor_degree
name | type | options
cwid | integer | primary key
name | text | ...
```

Primary key (cwid, degree)
Foreign key (cwid) references instructor (cwid)

Step 7 – N-ary Relationships

Only choice: cross-reference table
As for N:M relationships, but using primary keys from all involved tables.

Up Next

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