Entity Relationship (ER) Modeling
ERDs

- A diagram of the end-user view of a DB
- Primary goal is to model
  - Attributes
  - Relationships
  - Entities (duh)
They look like this!

**FIGURE 2.5** The Chen and Crow’s Foot notations

**Chen Notation**

A One-to-Many (1:M) Relationship: a PAINTER can paint many PAINTINGs; each PAINTING is painted by one PAINTER.

![Diagram of a One-to-Many Relationship]

A Many-to-Many (M:N) Relationship: an EMPLOYEE can learn many SKILLs; each SKILL can be learned by many EMPLOYEES.

![Diagram of a Many-to-Many Relationship]

A One-to-One (1:1) Relationship: an EMPLOYEE manages one STORE; each STORE is managed by one EMPLOYEE.

![Diagram of a One-to-One Relationship]
The Types

- Chen
  - Emphasis on *modeling*
- Crow's Foot
- UML
  - Both of these focus more on *design and implementation*
- In reality you can implement your DB from either
- **Entities** → corresponds real-world objects
  - *Entity* and *object* are used interchangeably
  - Entity Set → collection of entities
  - Entity Instance or Occurrence → a particular entity, basically a row in a table
  - Represented by a named rectangle in ERDs

- **Attributes** → entity characteristics
  - Connected ovals in Chen and listed in the rectangle in Crow's
  - Can be required or optional (but optional can be bad if implemented wrong!)
Attributes Examples

**Figure 4.1** The attributes of the STUDENT entity

**Chen Model**

- STU_INITIAL
- STU_FNAME
- STU_LNAME
- STU_EMAIL
- STU_PHONE

**Crow’s Foot Model**

<table>
<thead>
<tr>
<th>STUDENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>STU_LNAME</td>
</tr>
<tr>
<td>STU_FNAME</td>
</tr>
<tr>
<td>STU_INITIAL</td>
</tr>
<tr>
<td>STU_EMAIL</td>
</tr>
<tr>
<td>STU_PHONE</td>
</tr>
</tbody>
</table>
Review

- **Attributes**
  - Have domains: rules about what values are valid
  - Can be **simple** or **composite** (composite can be a bit sticky as to whether they should be broken down)

- **Identifiers** (aka primary keys)
  - Can be based on a single attribute or multiple (**composite**)
  - When using composites you must be careful about making sure they will be unique
More on Attributes

- Can be single- or multi-valued
  - Be careful, there is not necessarily a relationship between single-valued and being a simple attribute
  - We don't implement multi-valued though! (pg 109)

- Derived Attributes → values that can be calculated or reasoned from other attributes
  - These may or may not be stored in the database
  - Why?
Derived Attributes in ERDs

FIGURE 4.6 Depiction of a derived attribute

Chen Model

EMP_FNAME
EMP_INITIAL
EMP_LNAME
EMP_NUM
EMP_DOB
EMP_AGE

Crow’s Foot Model

<table>
<thead>
<tr>
<th>PK</th>
<th>EMP_NUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMP_LNAME</td>
<td>EMP_FNAME</td>
</tr>
<tr>
<td>EMP_INITIAL</td>
<td>EMP_DOB</td>
</tr>
<tr>
<td>EMP_AGE</td>
<td></td>
</tr>
</tbody>
</table>
## Stored vs. Calculated

### TABLE 4.2

Advantages and Disadvantages of Storing Derived Attributes

<table>
<thead>
<tr>
<th>DERIVED ATTRIBUTE</th>
<th>STORED</th>
<th>NOT STORED</th>
</tr>
</thead>
</table>
| **Advantage**     | Saves CPU processing cycles  
                   Saves data access time  
                   Data value is readily available  
                   Can be used to keep track of historical data | Saves storage space  
                   Computation always yields current value |
| **Disadvantage**  | Requires constant maintenance to ensure derived value is current, especially if any values used in the calculation change | Uses CPU processing cycles  
                   Increases data access time  
                   Adds coding complexity to queries |
Connectivity and Cardinality

FIGURE 4.7 Connectivity and cardinality in an ERD

PROFESSOR

Connectivities

teaches

(1,1)

CLASS

Cardinalities

(1,4)
An entity is **existence dependent** when its only reason for being in the database is to be associated with another entity

- It has a mandatory foreign key attribute, meaning it cannot be null
- In terms of the college database, an Emergency Contact entity is existence dependent upon a Student

Something may also be **existence independent**

- An example might be in-state transfer credits
**Relationship Strength**

- **Weak or Non-identifying** → the FK and *only* the FK of the related table is the PK of another
  - Sales had the Agent's ID (the Agent PK) as a FK
  - Pretty much the standard type of FK relationship we've been talking about in our examples so far
  - Drawn as a dashed line in a Crow's Foot ERD

- **Strong or Identifying** → the FK of the related table is also involved in its PK, along with being the PK of another table (solid line in Crow's)
  - If Sales no longer has a numeric PK but a composite of sales date, agent ID, and VIN then the relationship becomes strong
When creating tables, you must create the 1 side of a 1:M relationship first... why?

Also note that the way a relationship is determined is by looking at the table that contains the foreign key (the \textit{related table})
Entities can be weak too!

- If a related entity is both existence-dependent and has a strong relationship with its parent, it is a **weak entity**
  - In Crow's foot these are represented by the combination of the solid line of a strong relationship and the PK/FK designation
  - These are not very common and are up to the database designer
Weak Entity Example

Figure 4.8: A weak (non-identifying) relationship between COURSE and CLASS

Table name: COURSE

<table>
<thead>
<tr>
<th>CRS_CODE</th>
<th>DEPT_CODE</th>
<th>CRS_DESCRIPTION</th>
<th>CRS_CREDITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCT-211</td>
<td>ACCT</td>
<td>Accounting I</td>
<td>3</td>
</tr>
<tr>
<td>ACCT-212</td>
<td>ACCT</td>
<td>Accounting II</td>
<td>3</td>
</tr>
<tr>
<td>CIS-220</td>
<td>CIS</td>
<td>Intro. to Microcomputing</td>
<td>3</td>
</tr>
<tr>
<td>CIS-420</td>
<td>CIS</td>
<td>Database Design and Implementation</td>
<td>4</td>
</tr>
<tr>
<td>MATH-243</td>
<td>MATH</td>
<td>Mathematics for Managers</td>
<td>3</td>
</tr>
<tr>
<td>QM-261</td>
<td>CIS</td>
<td>Intro. to Statistics</td>
<td>3</td>
</tr>
<tr>
<td>QM-362</td>
<td>CIS</td>
<td>Statistical Applications</td>
<td>4</td>
</tr>
</tbody>
</table>

Table name: CLASS

<table>
<thead>
<tr>
<th>CLASS_CODE</th>
<th>CRS_CODE</th>
<th>CLASS_SECTION</th>
<th>CLASS_TIME</th>
<th>ROOM_CODE</th>
<th>PROF_NUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>10012</td>
<td>ACCT-211</td>
<td>1</td>
<td>MMF 8:00-8:50 a.m.</td>
<td>BUS311</td>
<td>105</td>
</tr>
<tr>
<td>10013</td>
<td>ACCT-211</td>
<td>2</td>
<td>MMF 9:00-9:50 a.m.</td>
<td>BUS200</td>
<td>105</td>
</tr>
<tr>
<td>10014</td>
<td>ACCT-211</td>
<td>3</td>
<td>TTh 2:30-3:45 p.m.</td>
<td>BUS252</td>
<td>342</td>
</tr>
<tr>
<td>10015</td>
<td>ACCT-212</td>
<td>1</td>
<td>MMF 10:00-10:50 a.m.</td>
<td>BUS311</td>
<td>301</td>
</tr>
<tr>
<td>10016</td>
<td>ACCT-212</td>
<td>2</td>
<td>Th 8:00-8:40 p.m.</td>
<td>BUS252</td>
<td>301</td>
</tr>
<tr>
<td>10017</td>
<td>CIS-220</td>
<td>1</td>
<td>MMF 9:00-9:50 a.m.</td>
<td>KLR209</td>
<td>228</td>
</tr>
<tr>
<td>10018</td>
<td>CIS-220</td>
<td>2</td>
<td>MMF 9:00-9:50 a.m.</td>
<td>KLR211</td>
<td>114</td>
</tr>
<tr>
<td>10019</td>
<td>CIS-220</td>
<td>3</td>
<td>MMF 10:00-10:50 a.m.</td>
<td>KLR209</td>
<td>228</td>
</tr>
<tr>
<td>10020</td>
<td>CIS-420</td>
<td>1</td>
<td>vW 8:00-8:40 p.m.</td>
<td>KLR209</td>
<td>162</td>
</tr>
<tr>
<td>10021</td>
<td>QM-261</td>
<td>1</td>
<td>MMF 8:00-8:50 a.m.</td>
<td>KLR200</td>
<td>114</td>
</tr>
<tr>
<td>10022</td>
<td>QM-261</td>
<td>2</td>
<td>TTh 1:00-2:15 p.m.</td>
<td>KLR200</td>
<td>114</td>
</tr>
<tr>
<td>10023</td>
<td>QM-362</td>
<td>1</td>
<td>MMF 11:00-11:50 a.m.</td>
<td>KLR200</td>
<td>182</td>
</tr>
<tr>
<td>10024</td>
<td>QM-362</td>
<td>2</td>
<td>Th 2:30-3:45 p.m.</td>
<td>KLR200</td>
<td>182</td>
</tr>
<tr>
<td>10025</td>
<td>MATH-243</td>
<td>1</td>
<td>Th 8:00-8:40 p.m.</td>
<td>DRE159</td>
<td>325</td>
</tr>
</tbody>
</table>

Database name: Ch04_TinyCollege
Participation

- Entity relationships can be of two types in relation to whether there is a related entity for every parent.
- To determine relationship participation you have to look at it both ways and determine its categorization for each direction.
- The easiest way to determine these is to examine cardinality.
Participation Types

- **Optional** → When the 'parent' does not require a 'child' to be present
  - In Crow's Foot a circle is added to the related entity's end

- **Mandatory** → There is an entity required on each end for the relationship to make sense
  - In Crow's Foot no circle is assumed mandatory
  - The cardinality can also help, there can be no 0 in a mandatory relationship
How to 'read' the diagram

- Fig. 4.12

- "Professor teaches 0 to 3 classes" is the relationship
  - To get the participation you read backwards: "Class is optional to professor"

- "A class is taught by 1 professor" is the relationship
  - "Professor is mandatory to class"
Crow's Foot Diagrams

**FIGURE 4.13** CLASS is optional to COURSE

**FIGURE 4.14** COURSE and CLASS in a mandatory relationship
Gotcha

- Participation and Strength do not determine each other.
- A mandatory relationship does not imply a strong relationship (or vice versa) nor the same for optional/weak.
- Relationship *strength* is determined by the composition of the related table's PK.
- Relationship *participation* are based on business rules (which may contradict).
- See the previous diagrams.
Things to be careful about

- Relationships help determine the order in which you create tables and their rows
  - If you create a class entity with no associated course, a temporary course would have to be created until the official course is approved
- Make sure you understand the semantics of the relationship description!
  - For our examples, keeping course and class straight is important as well as getting the cardinality correct
## Crow's Foot Symbols

<table>
<thead>
<tr>
<th>CROW'S FOOT SYMBOL</th>
<th>CARDINALITY</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Symbol 1]</td>
<td>(0,N)</td>
<td>Zero or many. Many side is optional.</td>
</tr>
<tr>
<td>![Symbol 2]</td>
<td>(1,N)</td>
<td>One or many. Many side is mandatory.</td>
</tr>
<tr>
<td>![Symbol 3]</td>
<td>(1,1)</td>
<td>One and only one. 1 side is mandatory.</td>
</tr>
<tr>
<td>![Symbol 4]</td>
<td>(0,1)</td>
<td>Zero or one. 1 side is optional.</td>
</tr>
</tbody>
</table>
Associative Entities

- First heard about these when talking about M:N relationships and how a typical DB model does not allow them.

- To overcome the problem, an associative entity is created as a bridge between the two.

- The idea is to turn the M:N into two 1:M relationships.

- The associative entity will have the PKs from each table as FKs and its own PK.

- No special ERD notation, look at the keys and strong/identifying relationships.
What would it be?

Think about the relationship between the two, we won't worry about cardinality
  - Optional or mandatory?
The ERD

FIGURE 4.24  The M:N relationship between STUDENT and CLASS

Visio does not permit the definition of a M:N relationship. To make this illustration, two 1:M relationships have been superimposed.
Things to think about:

- How the relationships change; the optionalities get moved to the new relationships
- What are the keys?
- What else needs to be in the new table?
The new ERD

A composite entity in an ERD

FIGURE 4.25
Developing an ER Diagram

Section 4.2 – Pg 127
A highly recommended read
Homework!

- Review Questions → 2, 3, 4, 5, 8, 9, 10, 15
- Problems → 1, 2, 5, 7