CSC 310
Database Theory and Implementation

Lecture Set # 2
Blase B. Cindric
University of Mount Union
a Conceptual Data Model concerned with capturing essential features of the Miniworld to be represented

In this model, the real world is considered to consist of:

- a set of *entities*, and
- *relationships* between entities
Entity-Relationship Data Model

E-R is the most popular conceptual data model in use today

imparts the meaning of a real-world situation

Idea here: analyze a situation, identifying essential parts; then convert to the logical schema to be used

(in our case this semester, the Relational DB model)
Basics of the E-R Data Model

**Entity Sets**

*entity*:  
a person or thing from the real-world that may be distinguished from all other entities

*entity set*:  
a set of entities that are all of the same kind  
(ex.: students, faculty, courses)
Attributes

attribute:
some property possessed by an entity
(ex.: student number, name, phone #, gpa, course #, etc.)

domain:
a set of all possible values that an attribute may have
(ex.: student number --- 9-digit integer
gpa --- all real numbers between 0 & 4, inclusive)
domains are similar to, but distinct from, data types
Classifying attributes

Simple vs. Composite attributes:

- *Simple* -- attribute cannot be decomposed (atomic)
- *Composite* -- attribute is composed of parts

  ex.: name attr. composed of first name, m.i., last name

Single-valued vs. Multi-valued attributes:

- *Single-valued* -- only one value per attr. for each entity
  - ex.: e-mail addresses for students
- *Multi-valued* -- more than one value is possible for attr.
  - ex.: instructor(s) for a course
Classifying attributes (continued)

Null attributes:
- value is unknown or missing
- important for design methods to be encountered later

Derived attributes:
- not actually stored in the DB, but calculated from other fields
- ex.: GPA might be derived from
  \[ \frac{\text{Quality}_\text{Pts}}{\text{Credits}_\text{Earned}} \]
Entity Set Example

Entities are represented mathematically in a DB as sets of (attribute, value) pairs

\[
\text{STUDENT} = \{ 
\begin{array}{l}
\{ \text{(stNum, 11111)}, \text{(name, “Blough, Joe”)}, \text{(phone, 7777)} \}, \\
\{ \text{(stNum, 22222)}, \text{(name, “Smith, Sarah”)}, \text{(phone, 6767)} \}, \\
\{ \text{(stNum, 33333)}, \text{(name, “Jones, Helga”)}, \text{(phone, 7767)} \}, \\
\{ \text{(stNum, 44444)}, \text{(name, “Mars, Lars”)}, \text{(phone, 7000)} \} \}
\]

\[
\text{COURSES} = \{ 
\begin{array}{l}
\{ \text{(courseNum, “CSC 310”)}, \text{(title, “Database”)}, \text{(time, “TR 8”)} \}, \\
\{ \text{(courseNum, “PHE 999”)}, \text{(title, “Spelunking”)}, \text{(time, “MWF 1”)} \}, \\
\{ \text{(courseNum, “REL 777”)}, \text{(title, “Buddhism”)}, \text{(time, “TR 3”)} \} \}
\]
Entity Set Example

An Entity Set is the set of entities of the same type (a set of sets!)
Identifying Entities

We need a way to uniquely (and efficiently) identify entities.

Idea: find a group of attributes whose values are unique for each entity.

Such a set of attributes is called a key for the entity.
Identifying Entities (Keys)

**Candidate Key:**

a set of one or more attributes $K$ of entity set $E$ is a *candidate key* for $E$ if:

1) at any given time, no two entities in $E$ may have the same value(s) for $K$ (the Uniqueness property)

2) if $K$ is composite, no component of $K$ may be eliminated without destroying property 1 (the Minimality property)
Identifying Entities (Keys)

There may be more than one candidate key for an Entity Set

ex.: for Students, \{ Student\_Number \} would be a cand. key, as would \{ Social\_Security\_Num \}

*Primary Key:*

one of the candidate keys, chosen by the DBA as the main key for use with the DB
Identifying Entities (Keys)

Alternate Key(s):
all of the candidate keys not chosen as primary key

Superkey:
any set of attributes that has the Uniqueness property

ex.: \{ Student\_Number, Major \} is a superkey, not a candidate key
Relationship Sets

relationship:
an association between two or more entities
Blough, Joe is taking CSC 310 and REL 777

relationship set:
a set of all relationships between the same kind of entities
with 4 students and 3 courses, 12 relationships are possible (if all students take all courses)
Actual relationship set “Takes” would be a subset of all possible relationships
Relationship Set Example

11111, “Blough, Joe”, 7777
22222, “Smith, Sarah”, 6767
33333, “Jones, Helga”, 7767
44444, “Mars, Lars”, 7000

“CSC 310”, “Database”, “TR 8”
“PHE 999”, “Spelunking”, “MWF 1”
“REL 777”, “Buddhism”, “TR 5”
Relationship Set Example

Relationships may have attributes, in addition to entities having attributes

ex.: the grade for a student in a course

not part of the Student entity set, because each student has a different grade for every course

not part of the Course entity set, because each course has a different grade for each student registered for that course

the proper place to keep track of grades is as an attribute of the relationship
Structural Constraints (Sec. 12.6)

multiplicity

the number of other entities to which some particular entity may be associated via a relationship set

It is very important to identify the multiplicity of a relationship when designing a DB schema

each type of multiplicity is handled differently when implementing the DB in an actual DBMS
Multiplicity Types

Three main multiplicities:

• one - to - one ( 1 : 1 )
  ex. departments and chairpersons
  each department has only one chairperson
  each chairperson is in charge of only one department

• one - to - many ( 1 : * )
  ex. advisor relationship
  each student has only one advisor
  each faculty member may have many advisees

• many - to - many ( * : * )
  ex. course registration
  a student may take many courses in a semester
  a course may be taken by many students at once
Our textbook uses UML diagrams for E-R diagrams. This differs from traditional DB notation, but is becoming a new standard notation.

Entities are named in the top part of rectangles. Attributes are listed on separate lines in the bottom part of rectangles. (primary key attributes denoted by {PK} suffix)

Relationships are drawn as labeled directed lines. Multiplicities of Relationships shown for each direction. Format is [min #] .. [max #]
Entity-Relationship Diagram Example

A Simple Example:

<table>
<thead>
<tr>
<th>Entity_1</th>
<th>0..1</th>
<th>Owns</th>
<th>1..*</th>
<th>Entity_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>ent_1_key {PK}</td>
<td></td>
<td></td>
<td></td>
<td>ent_2_key {PK}</td>
</tr>
</tbody>
</table>

an Entity_1 can own one or more instances of Entity_2

an Entity_2 can be owned by zero or one instance of Entity_1

this is a one-to-many relationship
Entity-Relationship Diagram Example

Second Example: author(s) of book(s)

<table>
<thead>
<tr>
<th>Person</th>
<th>1 .. *</th>
<th>Wrote</th>
<th>0 .. *</th>
<th>Book</th>
</tr>
</thead>
<tbody>
<tr>
<td>name {PK}</td>
<td></td>
<td></td>
<td></td>
<td>title {PK}</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pages</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pub_date</td>
</tr>
</tbody>
</table>

- a Person may have written zero or more Books
- a Book may have been written by one or more People

this is a *many-to-many relationship*
In the notation for a multiplicity,

the first number is either 0 or 1
indicates whether the relationship is optional (0) or mandatory (1)
0 means an entity doesn’t have to be associated with any entity from the other Entity Set participating in the relationship
1 means that every entity must be associated with some other entity from the other Entity Set

the second number is either 1 or *
this determines one-to-one, one-to-many or many-to-many status when considering both directions of the relationship
the more important value!!
Entity-Relationship Relationship Diagram Example

A Larger Example:

Entity Sets:
Students, Courses, Faculty Members

Relationship Sets:
Students \( \text{Take} \) Courses
Faculty \( \text{Teach} \) Courses
Faculty \( \text{Advise} \) Students

Multiplicities:
\( \text{Takes} \) is a many-to-many relationship
\( \text{Teaches} \) is many-to-many
\( \text{Advises} \) is one-to-many
Entity-Relationship Diagram Example

A Larger Example:

- **Student**
  - studentNum {PK}
  - 1 .. *
  - 0 .. *

- **Course**
  - courseNum {PK}
  - section {PK}
  - 1 .. *

- **Takes**
  - 1 .. *

- **Faculty**
  - facIDNum {PK}
  - 1 .. *

- **Advises**
  - 1 .. 1

- **Teaches**
  - 1 .. *
Direction of Arrow is Important!

Consider this E-R Diagram:

<table>
<thead>
<tr>
<th>EntityA</th>
<th>1..*</th>
<th>RelatesTo</th>
<th>1..1</th>
<th>EntityB</th>
</tr>
</thead>
</table>

How do we determine whether a single entity from one Entity Set can be related to only one entity from the other Entity Set, or to more than one entity from the other Entity Set?

Must analyze the situation separately in **both** directions!

use only the max # when determining this
Direction of Arrow is Important!

In this situation:

<table>
<thead>
<tr>
<th>EntityA</th>
<th>1 .. *</th>
<th>RelatesTo</th>
<th>1 .. 1</th>
<th>EntityB</th>
</tr>
</thead>
</table>

Each EntityA relates to exactly one EntityB

Each EntityB may be related to many EntityA’s

How to do it:

[Entity] [Relationship phrase] [multiplicity] [Entity]
Assign multiplicities to the Relationships in this E-R diagram:

- **Buys**
  - **Buyer**
    - buyerID {PK}
    - name
    - phoneNum
  - **Property**
    - propID {PK}
    - askingPrice
    - salePrice
    - address

- **Sells**

- **WorksAt**
  - **StaffMember**
    - SocSecNum {PK}
    - lastName
    - firstName
    - phone
  - **Branch**
    - branchID {PK}
    - name
    - address
    - phone

- **Manages**
Try One Yourself!

Here are the correct multiplicities for this problem…

```
<table>
<thead>
<tr>
<th>Table</th>
<th>E-R Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buyer</td>
<td>0..1</td>
</tr>
<tr>
<td></td>
<td>Buys</td>
</tr>
<tr>
<td>buyerID {PK}</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>askingPrice</td>
</tr>
<tr>
<td>phoneNum</td>
<td>salePrice</td>
</tr>
</tbody>
</table>

| StaffMember      | 0..*      |
|                  | Sells     | 0..*     |
| SocSecNum {PK}   |           |          |
| lastName         | branchID {PK} |
| firstName        | name      |
| phone            | address   |

| Branch           | 1..1      |
| WorksAt          | 1..*      |
| branchID {PK}    |           |
| name             |          |
| address          |          |

| Branch           | 0..1      |
| Manages          | 1..1      |
| branchID {PK}    |           |
| name             |          |
| address          |          |
| phone            |          |
```
Creating an Entity-Relationship Diagram from a description

Create an E-R diagram for the following enterprise:

Each building in an organization has a different building name and address. The meeting rooms in each building have their own room numbers and seating capacities. Rooms may be reserved for meetings, and each meeting must start on the hour. The hour and length of each meeting are recorded. Each reservation is made by a group in the company. Each group has a group number and a contact phone.

Step 1: Identify the Entity Sets and Attributes for this problem

Step 2: Draw the E-R diagram
Creating an Entity-Relationship Diagram from a description

Identify the Entity Sets and Attributes for this problem

Entity Sets:
- Buildings
- MeetingRooms
- Meetings
- Groups

Attributes:
- Buildings: bldgName, address
- MeetingRooms: roomNum, capacity
- Meetings: meetingName, startHour, length
- Groups: groupName, phone
Creating an Entity-Relationship Diagram from a description

Here is one possible solution (there may be others):

- **Building**
  - bldgName {PK}
  - address

- **MeetingRoom**
  - roomNum {PK}
  - capacity

- **Group**
  - groupName {PK}
  - phone

- **Meeting**
  - meetingName {PK}
  - startHour
  - length

- **Holds**

- **IsIn**

- **HeldIn**

CSC 310 DB Theory & Impl.