Data Modeling and the Entity-Relationship Model

Ashraf Aboulnaga

David R. Cheriton School of Computer Science
University of Waterloo

CS 348
Introduction to Database Management
Winter 2013

Notes
Outline

1 Basic E-R Modeling
   Entities
   Attributes
   Relationships
   Roles

2 Constraints in E-R Models
   Primary Keys
   Relationship Types
   Existence Dependencies
   General Cardinality Constraints

3 Extensions to E-R Modeling
   Structured Attributes
   Aggregation
   Specialization
   Generalization

4 Design Considerations

5 Translating E-R Diagrams to a Relational Schema

Notes
Overview of E-R Model

Used for (and designed for) database (conceptual schema) design

⇒ Proposed by Peter Chen in 1976

World/enterprise described in terms of

- entities
- attributes
- relationships

Visualization: E-R diagram

N.B. Many variant notations are in common use
Basic E-R Modeling

**Entity**: a *distinguishable* object

**Entity set**: set of entities of same type

Examples:
- students currently at University of Waterloo
- flights offered by Air Canada
- burglaries in Ontario during 1994

**Graphical representation of entity sets:**

```
Student   Flight   Burglary
```
Attributes: describe properties of entities
Examples (for Student-entities): StudentNum, StudentName, Major, ...

Domain: set of permitted values for an attribute

Graphical representation of attributes:
**Relationship**: representation of the fact that certain entities are related to each other

**Relationship set**: set of relationships of a given type

Examples:
- students registered in courses
- passengers booked on flights
- parents and their children
- bank branches, customers and their accounts

In order for a relationship to exist, the participating entities must exist.
Graphical Representation

Student

\[ \text{StudentNum} \]

\[ \text{StudentName} \]

\[ \text{RegisteredIn} \]

Course

\[ \text{CourseNum} \]

Notes

_________________________________________________________________
_________________________________________________________________
_________________________________________________________________
_________________________________________________________________
_________________________________________________________________
_________________________________________________________________
_________________________________________________________________
_________________________________________________________________
_________________________________________________________________
_________________________________________________________________
Multiple Relationships and Role Names

**Role**: the function of an entity set in a relationship set

**Role name**: an explicit indication of a role

Example:

```
Team
TeamName
Match Location
Address
LocName
Visitor
HomeTeam
```

Role labels are needed whenever an entity set has multiple functions in a relationship set.
Relationships and Attributes

Relationships may also have attributes

Example:

```
Team
  TeamName
  HomeTeam
  Visitor

Match
  Score
  Location

Visitor

Address

LocName
```
• Primary keys
• Relationship types
• Existence dependencies
• General cardinality constraints
Primary Keys

Each entity must be distinguishable from any other entity in an entity set by its attributes.

**Primary key**: selection of attributes chosen by designer values of which determines the particular entity.

**Example 1**:

```
Dnum → Department → ManagerName

Dname \ Budge t
```

**Example 2**:

```
FirstName → Employee → Salary

Initial \ LastName
```

---

Notes

_________________________________________________________________________

_________________________________________________________________________

_________________________________________________________________________

_________________________________________________________________________

_________________________________________________________________________

_________________________________________________________________________

_________________________________________________________________________

_________________________________________________________________________

_________________________________________________________________________

_________________________________________________________________________
Relationship Types

- **many-to-many (N:N)**: an entity in one set can be related to many entities in the other set, and vice versa (This is the interpretation we have used so far.)

- **many-to-one (N:1)**: each entity in one set can be related to at most one entity in the other, but an entity in the second set may be related to many entities in the first

- **one-to-many (1:N)**: similar

- **one-to-one (1:1)**: each entity in one set can be related to at most one entity in the other, and vice versa

---

Notes

__________________________________________________________________________

__________________________________________________________________________

__________________________________________________________________________

__________________________________________________________________________

__________________________________________________________________________

__________________________________________________________________________

__________________________________________________________________________
Existence Dependencies

Sometimes the existence of an entity depends on the existence of another entity.

If \( x \) is existence dependent on \( y \), then

- \( y \) is a dominant entity
- \( x \) is a subordinate entity

Example: “Transactions are existence dependent on accounts.”
Identifying Subordinate Entities

**Weak entity set**: an entity set containing subordinate entities  
**Strong entity set**: an entity set containing no subordinate entities

Attributes of weak entity sets only form key relative to a given dominant entity

**Example**: “All transactions for a given account have a unique transaction number.”

![Diagram]

Notes
Identifying Subordinate Entities (cont’d)

A weak entity set must have a many-to-one relationship to a distinct entity set

Visualization: (distinguishing an identifying relationship)

Discriminator of a weak entity set: set of attributes that distinguish subordinate entities of the set, for a particular dominant entity

Primary key for a weak entity set: discriminator + primary key of entity set for dominating entities
General Cardinality Constraints

General cardinality constraints determine lower and upper bounds on the number of relationships of a given relationship set in which a component entity may participate.

Visualization:

Example:

Notes
Extensions to E-R Modeling

- Structured attributes
- Aggregation
- Specialization
- Generalization
- Disjointness
Structured Attributes

**Composite attributes:** composed of fixed number of other attributes

**Multi-valued attributes:** attributes that are set-valued

Example:

![Diagram showing Employee, Address, Hobbies, Street, City, Province, PostalCode]

Notes

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________
Aggregation

Relationships can be viewed as higher-level entities

Example: “Accounts are assigned to a given student enrollment.”

![Diagram showing relationships between entities: Student, EnrolledIn, Course, CourseAccount, Account, UserId, ExpirationDate, and CourseNum.]

Notes
Specialization

A specialized kind of entity set may be derived from a given entity set.

Example: “Graduate students are students who have a supervisor and a number of degrees.”
Several entity sets can be abstracted by a more general entity set.

Example: “A vehicle abstracts the notion of a car and a truck.”
Disjointness

Specialized entity sets are usually disjoint but can be declared to have entities in common

- By default, specialized entity sets are disjoint.
  Example: We may decide that nothing is both a car and a truck.
- However, we can declare them to overlap (to accommodate utility vehicles, perhaps).

```
MakeAndModel ____________________________________________
|                                                |
| LicenceNum                                      |
|                                               |
| Vehicle                                         |
|                                               |
| Price                                           |
|                                               |
| OVERLAPS                                        |
|                                               |
| Tonnage                                         |
|                                               |
| Truck                                           |
|                                               |
| Car                                             |
|                                               |
| MaxSpeed                                        |
|                                               |
| AxelCount                                       |
|                                               |
| PassengerCount                                  |
```

Notes

_____________________________________________________________________
_____________________________________________________________________
_____________________________________________________________________
_____________________________________________________________________
_____________________________________________________________________
_____________________________________________________________________
_____________________________________________________________________
_____________________________________________________________________
_____________________________________________________________________
_____________________________________________________________________
_____________________________________________________________________
_____________________________________________________________________
_____________________________________________________________________
Designing An E-R Schema

Usually many ways to design an E-R schema
Points to consider

- use attribute or entity set?
- use entity set or relationship set?
- degrees of relationships?
- extended features?
Attributes or Entity Sets?

Example: Should one model employees’ phones by a PhoneNumber attribute, or by a Phone entity set related to the Employee entity set?

Rules of thumb:
- Is it a separate object?
- Do we maintain information about it?
- Can several of its kind belong to a single entity?
- Does it make sense to delete such an object?
- Can it be missing from some of the entity set’s entities?
- Can it be shared by different entities?

An affirmative answer to any of the above suggests a new entity set.
Entity Sets or Relationships?

Instead of representing accounts as entities, we could represent them as relationships.
Binary vs. N-ary Relationships?
We can always represent a relationship on $n$ entity sets with $n$ binary relationships.
A Simple Methodology

1. Recognize entity sets
2. Recognize relationship sets and participating entity sets
3. Recognize attributes of entity and relationship sets
4. Define relationship types and existence dependencies
5. Define general cardinality constraints, keys and discriminators
6. Draw diagram

For each step, maintain a log of assumptions motivating the choices, and of restrictions imposed by the choices.
Example: A Registrar’s Database

- Zero or more sections of a course are offered each term. Courses have names and numbers. In each term, the sections of each course are numbered starting with 1.
- Most course sections are taught on-site, but a few are taught at off-site locations.
- Students have student numbers and names.
- Each course section is taught by a professor. A professor may teach more than one section in a term, but if a professor teaches more than one section in a term, they are always sections of the same course. Some professors do not teach every term.
- Up to 50 students may be registered for a course section. Sections with 5 or fewer students are cancelled.
- A student receives a mark for each course in which they are enrolled. Each student has a cumulative grade point average (GPA) which is calculated from all course marks the student has received.
Example: A Registrar’s Database (cont’d)

Course

Section

Off-Site Section

Professor

Student

CS 348  E-R Model  Winter 2013  31 / 50

Notes

_________________________________________________________________

_________________________________________________________________

_________________________________________________________________

_________________________________________________________________

_________________________________________________________________

_________________________________________________________________

_________________________________________________________________

_________________________________________________________________

_________________________________________________________________

_________________________________________________________________
Example: A Registrar’s Database (cont’d)

CourseNum → Course → CourseName

SectionOf

Term → SectionNum → Section

TaughtBy

Professor → ProfName → ProfNum

EnrolledIn

Student → Location → StudentNum → StudentName

Off–Site Section

Mark → GPA

Notes
Example: A Registrar’s Database (cont’d)

Notes

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________
Main ideas:

- Each entity set maps to a new table
- Each attribute maps to a new table column
- Each relationship set maps to either new table columns or to a new table
Representing Strong Entity Sets

Entity set $E$ with attributes $a_1, \ldots, a_n$ translates to table $E$ with attributes $a_1, \ldots, a_n$.

Entity of type $E \leftrightarrow$ row in table $E$.
Primary key of entity set $\rightarrow$ primary key of table.
Example:

\[
\begin{array}{|c|c|c|}
\hline
\text{StudentNum} & \text{StudentName} & \text{Major} \\
\hline
\end{array}
\]

Notes
Representing Weak Entity Sets

Weak entity set $E$ translates to table $E$

Columns of table $E$ should include
- Attributes of the weak entity set
- Attributes of the identifying relationship set
- Primary key attributes of entity set for dominating entities

Primary key of weak entity set $\rightarrow$ primary key of table
Representing Weak Entity Sets (cont.)

Example:

```
<table>
<thead>
<tr>
<th>AccNum</th>
<th>Balance</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>TransNum</th>
<th>AccNum</th>
<th>Date</th>
<th>Amount</th>
</tr>
</thead>
</table>
```

Notes
Representing Relationship Sets

- If the relationship set is an identifying relationship set for a weak entity set then no action needed

- If we can deduce the general cardinality constraint (1,1) for a component entity set $E$ then add following columns to table $E$
  - Attributes of the relationship set
  - Primary key attributes of remaining component entity sets

- Otherwise: relationship set $R \rightarrow$ table $R$
• Columns of table $R$ should include
  • Attributes of the relationship set
  • Primary key attributes of each component entity set

• Primary key of table $R$ determined as follows
  • If we can deduce the general cardinality constraint (0,1) for a component entity set $E$, then take the primary key attributes for $E$
  • Otherwise, choose primary key attributes of each component entity
Representing Relationship Sets (cont.)

Example:

Note that the role name of a component entity set should be prepended to its primary key attributes, if supplied.
Representing Aggregation

Tabular representation of aggregation of $R$

\[ \text{= tabular representation for relationship set } R \]

To represent relationship set involving aggregation of $R$, treat the aggregation like an entity set whose primary key is the primary key of the table for $R$
Representing Aggregation (cont.)

Example:

```
<table>
<thead>
<tr>
<th>Student</th>
<th>EnrolledIn</th>
<th>Course</th>
</tr>
</thead>
<tbody>
<tr>
<td>StudentNum</td>
<td></td>
<td>CourseNum</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CourseAccount</th>
<th>ExpirationDate</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserId</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Account</th>
<th>UserId</th>
<th>StudentNum</th>
<th>CourseNum</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>EnrolledIn</th>
<th>StudentNum</th>
<th>CourseNum</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>CourseAccount</th>
<th>UserId</th>
<th>StudentNum</th>
<th>CourseNum</th>
<th>ExpirationDate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Notes
Representing Specialization

Create table for higher-level entity set, and treat specialized entity subsets like weak entity sets (without discriminators)

Example:

```
+----------------+----------------+----------------+----------------+
|                |                |                |                |
| Student        | Graduate       | Degrees        | Professor      |
+----------------+----------------+----------------+----------------+
| StudentNumber  | StudentName    | StudentNumber  | ProfessorName  |
| StudentName    |                |                |                |
+----------------+----------------+----------------+----------------+
| (0, N)         | (1, 1)         | (0, N)         |                |
| SupervisedBy   |                |                |                |
+----------------+----------------+----------------+----------------+
| Professor      |                |                |                |
| ProfessorName  |                |                |                |
+----------------+----------------+----------------+----------------+
```

Notes
Representing Generalization (Approach #1)

Create a table for each lower-level entity set only

Columns of new tables should include
- Attributes of lower level entity set
- Attributes of the superset

The higher-level entity set can be defined as a view on the tables for the lower-level entity sets
Representing Generalization (Approach #1)

Example:

```
MakeAndModel
LicenceNum    Vehicle    Price

Tonnage    Truck    Car    MaxSpeed
AxelCount
PassengerCount

COVERS

LicenceNum    MakeAndModel    Price    Tonnage    AxelCount
Car
LicenceNum    MakeAndModel    Price    MaxSpeed    PassengerCount
Truck
```

Notes
Representing Generalization (Approach #2)

Treat generalization the same as specialization.

Example:

```
MakeAndModel

LicenceNum --> Vehicle --> Price

COVERS

Truck

Tonnage --> AxelCount

Car

MaxSpeed --> PassengerCount

Vehicle

<table>
<thead>
<tr>
<th>LicenceNum</th>
<th>MakeAndModel</th>
<th>Price</th>
</tr>
</thead>
</table>

Truck

| LicenceNum | Tonnage | AxelCount |

Car

| LicenceNum | MaxSpeed | PassengerCount |
```
Example Translation: ER Diagram

Notes