CS 331
- Kangmei Yang
Development Life Cycle

Planning
- Plan how the stages of the life cycle be more efficiently & effectively

System Definition
- Scope & boundaries of the database system

Requirements Collection & Analysis
- Collect & analyze of the requirements for the new database system

Database Design
- Support the organization’s mission statement & objectives for the required database

Application Design

Implementation
Data conversion & loading

Testing
Maintenance
Database Design Approaches

- **Bottom up**
  - Fundamental level – attributes
  - Small organization with small amount of attributes

- **Top down**
  - High-level entities & relationships to low-level entities, relationships & associated attributes
  - Medium to large organization
Phases of Database Design

- Conceptual
  - Construct a model

- Logical
  - Map conceptual data model to logical data model

- Physical
  - Produce a description of the implementation (how to implement)
Data Model

High level description of the schema

- Object based Data Model
  - ER (Entity-Relationship) 1976

- Record based Data Model
  - Relational
  - Network
  - Hierarchical
ER Model

- **Entity Types (Physical, Conceptual)**
  - A group of objects with the same properties, which are identified by the organization as having an independent existence

- **Entity Occurrence**
  - An uniquely identifiable object of an entity type.

- **Relationship Types**
  - A set of meaningful associations among entity types.

- **Relationship Occurrence**
  - An uniquely identifiable association between participating entity type
Semantic Net

- Object Level Model
- Represent each occurrence

Entity Occurrence
Student

S1
S2

Relationship Occurrence
Takes

T1
T2
T3

Entity Occurrence
Class

CS331-36
CS313-37

Student Takes Class

T1
T2
T3

S1
S2
Attributes

Property of an entity or relationship type
- **Simple Attribute (atomic)**
  - single component with independent existence

- **Composite Attribute**
  - multiple component each with an independent existence.

- **Single-valued Attribute**
  - holds a single value for each occurrence of an entity type.

- **Multi-valued Attribute**
  - holds a multiple value for each occurrence of an entity type, may have a limit

- **Derived Attribute**
  - derived from related attributes
Keys

Uniquely identifies each occurrence of an entity type
- Candidate Key
  - a minimal set of attributes that uniquely identifies each occurrence (can’t be null)

- Composite Key
  - a candidate key that consists of two or more attributes, each attribute may not be unique by itself

- Primary Key
  - a candidate key that is selected to uniquely identify each occurrence, ensures future uniqueness

- Alternate Key
Diagrammatic representation

- UML (Unified Modeling Language)

- Chen’s Notation

**EntityType1**
- primaryAttr (PK)
- compositeAttr
- cmpstA1
- cmpstA2

**EntityType2**
- partialPAttr1 (PPK)
- partialPAttr2
- multiVAttr [1..*]
- derivedAttr

**Relationship**

- primaryAttr
- partialPAttr1
- partialPAttr2

- compositeAttr
- multiVAttr
- cmpstA1
- cmpstA2
- derivedAttr
Degree of Relationship Type

Number of participating entity types in a relationship

- Binary (Most common)
  - a relationship of degree two, involves two entity types

- Complex Relationship – More than two, use diamond to represent relationships
  - ternary – a relationship of degree three, involves three entity types
  - quaternary – involves four entities

- Recursive Relationship (Unary)
  - same entity type participates more than once in different roles
Binary Relationship

- UML

Student ➔ Takes ➔ Class
Complex Relationship

- UML

Student

Instructor

Takes

Class

Location
Recursive Relationship

- UML

Diagram:

- Requires:
  - Pre-requisite: Advanced
  - Class
Strong and Weak Entity Types

- Strong Entity Type
  - not existence-dependent on other entity type

- Weak Entity Type
  - existence-dependent on other entity type
  - each entity occurrence cannot be uniquely identified using only the attributes associated with that entity type (No PK)
Multiplicities

A constraint on relationships that specify the possible occurrences of an entity type that may relate to a single occurrence of an associated entity type through a particular relationship.

- **1:1**
  - Staff-Chairman and Department

- **1:***
  - Ex: Advisor and Doctoral student

- ***:**
  - Ex: Student and Class

Cardinality and Participation
Cardinality and Participation

Cardinality
- Maximum number of possible relationship occurrence

Participation (Modality)
- Whether all or some entity occurrences participate in a relationship
  - mandatory (all)
  - optional (some)
Cardinality and Participation

UML
0..1
1..1
0..*
1..*
1..3

Crow’s Foot

[Diagram of Crow’s Foot notation]
Connection Traps

- **Fan Trap**
  - relationship between entity types where the pathway between certain entity occurrence is ambiguous

- **Chasm Trap**
  - existence of a relationship between entity type but the pathway does not exist between certain entity occurrence
Fan Traps

(M:1 and 1:M)

Sample:

<table>
<thead>
<tr>
<th>Entity</th>
<th>Relationship</th>
<th>Entity</th>
<th>Relationship</th>
<th>Entity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instructor</td>
<td>teaches</td>
<td>Class</td>
<td>has</td>
<td>Section</td>
</tr>
</tbody>
</table>

Which section does instructor I1 teach?
Fan Traps

Which section does instructor $I_1$ teach? $(M:1$ and $1:M)$ reconstruct the model to $1:M$ and $1:M(1)$

<table>
<thead>
<tr>
<th>Entity Class</th>
<th>Relationship has</th>
<th>Entity Section</th>
<th>Relationship by</th>
<th>Entity Instructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>331</td>
<td>h1</td>
<td>36</td>
<td>r1</td>
<td>I1</td>
</tr>
<tr>
<td>313</td>
<td>h2</td>
<td>26</td>
<td>r2</td>
<td>I2</td>
</tr>
<tr>
<td></td>
<td>h3</td>
<td>37</td>
<td>r3</td>
<td></td>
</tr>
</tbody>
</table>
### Chasm Traps

With optional participation

#### Sample:

<table>
<thead>
<tr>
<th>Entity Major</th>
<th>Relationship requires</th>
<th>Entity Class</th>
<th>Relationship taken by</th>
<th>Entity Student</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS BA</td>
<td>r1</td>
<td>CS 111</td>
<td>t1</td>
<td>S1</td>
</tr>
<tr>
<td>Acct</td>
<td>r2</td>
<td>CS 112</td>
<td>t2</td>
<td>S2</td>
</tr>
<tr>
<td></td>
<td>r3</td>
<td>Acc 101</td>
<td></td>
<td>S3</td>
</tr>
</tbody>
</table>

What major does student S2 declared?
Chasm Traps

What major does student S2 declared?
Add missing relationship (between Student and Major)

- Entity: Major
- Relationship: requires
- Entity: Class
- Relationship: taken by
- Entity: Student

Diagram:
- CS BA
- Acct
- CS 111
- CS 112
- Acc 101
- t1
- t2
- t3
- S1
- S2
- S3
- d1
- d2
- d3

Declared by r1, r2, r3
Enhanced ER Modeling

Is-A relationship (Specialization/Generalization)

Inheritance - superclasses and subclasses
- Process of specialization (Maximizing the differences between members of an entity, identify distinguishing characteristics)
  - subclass – distinct subgroupings of an entity type
- Process of generalization (Minimizing the differences, identify common characteristics)
  - superclass – includes one more distinct subgroupings of its occurrences

Multiple inheritance
Constrains

- Participation Constraints
  - determine whether every member in the superclass must participate as a member of a subclass
    - mandatory
    - optional

- Disjoint Constraints
  - indicate whether it is possible for a member of a superclass to be a member of more than one subclass
    - or
    - and

{mandatory, or}
{mandatory, and}
{optional, or}
{optional, and}
EER

Undergrad

{Mandatory, Or}

Graduate

Student
EER

Undergrad

{Optional, Or}

Graduate

Student
EER

Admin

Faculty

\{Mandatory, And\}

Staff
Aggregation

Has-A relationship between entity types

- Engine
- Wheel
- Seat

Car
Exam taking database:

Exam takers who’re identified by social security number are known by name.

There’re many different versions of the exam that’re available, which are numbered.

Proctors are also identified by social security number and know by name. Proctors may be contacted by email.

Once an exam has been taken, the exam will be graded.
Exam Taking

- Proctor
  - socSecNum
  - name
  - email

- ExamTaker
  - socSecNum
  - name

- Takes

- Exam
  - versionNum

- grades