Next: LECTURE 3 - RELATIONAL DATA MODEL, Previous: LECTURE 1 - INTRO TO DATABASES AND SQL, Up: Index

LECTURE 2 - E/R MODEL FILE

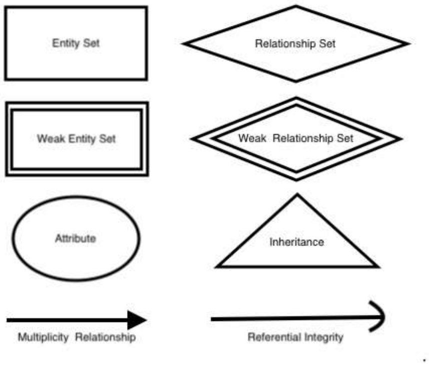
E/R = Entity Relationship Model
Object Oriented (ex: Object Definition Language)
Semi-structured Data (ex: XML)

Database Design

Database Design is made using a Data Model describing:

- data and relationships among data
- data semantics and constraints

Entity Relationship Model:



Entity:

- Real world object distinguishable from other objects
- *ODL = Object*

Entity Set:

- A collection of similar entities
- All entities in an entity set have the same set of attribute
- *ODL = Class*
- (Student, Course)

Attribute:

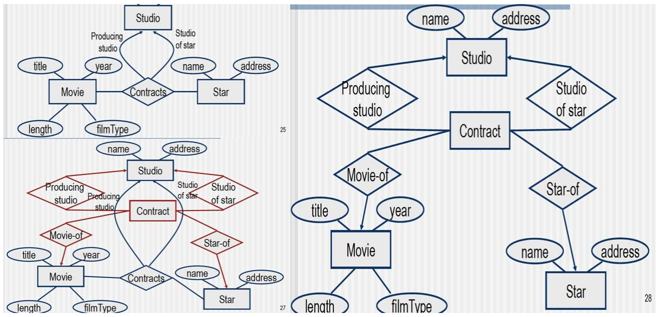
- Attributes (type) are atomic (1NF) (not allowed multiple values)
- (courseNumber, name, credits)(firstName, lastName, studentID, GPA)

Relationship:

- Association among entities
- one-one
- one-many
- many-many

Relationship Set:
Set of relationships of the same type
If E1, E2, . . ., En are entity sets, a relationship R on these sets is defined as:
R ⊆ E1 x … x En
HERE

Converting n-ary relationship to binary



Inheritance:
Represented by ISA
In OO = an object must be a member of exactly one class

Constraints

Things that cannot be represented using the **ODL** or **E/R** model introduced so far

Key K:
unique identifiers

Single-value Constraints:
1:1 — 1:M — M:1
The value for an attribute must exist if it is part of the key
The value for an attribute is optional, otherwise
Normalization = Atomic, means multiple values are not allowed

Referential Integrity Constraints:
Objects referred to by some object/entity must actually exist
Rounded Arrow Notation

Domain Constraints:
Restrict the values of an *attribute* to be drawn from a *set*

General Constraints:
Arbitrary constraints that must hold on the DB

Relationship degree constraints (multiplicity):
Restrict the number of entities in the entity sets involved in a relationship
ex: student cannot be enrolled in more than 5 courses

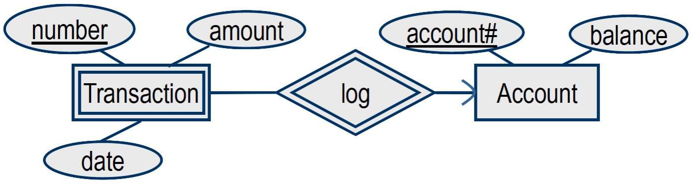
Keys

Superkey:
Set of attributes whose values uniquely identify an entity (object) in the entity set (class)

Minimal Superkey:
An entity can have *many* **Candidate/Alternate Keys**
Candidate keys are a special subset of super-keys that do not have any extraneous information in them
Only one of which is picked as the ~**Primary Key**~ (underlined in E/R)

Weak Entity & Weak Relationship Sets

ATM deposit transactions example: Different accounts might have transactions by the same number, on the same date, and for the same amount



- A **strong** entity set has a key
- A **weak**entity set does not have sufficient attributes of its own to form a key.
  - It participates in a M-1 relationship (with no descriptive attributes) with a strong entity set
- **Discriminator** of a **weak** entity set is a set of attributes that distinguishes among the entities corresponding to a strong entity (/#number)
- **key** of a weak entity set = key of the strong entity combined with the discriminator of the weak entity (ex above: number + account#)
- CES = Connecting Entity Sets = Always Weak
  - often have no attributes of their own

Design Principles

1. Faithful/Realistic
2. Avoid Redundancy
3. Simple

Choosing Attribute VS Entity Set or Class

- An **attribute** is simpler to implement than a class/entity set or a relationship.

- If something has more information associated with it than just its name, it should probably be modelled ed as an **entity set or a class**