

Entity Relationship Models 2

3 November 2009
Lecture 3

Topics for Today

- Entity Relationship Diagrams
 - Simple to Complex
 - Entities, Attributes, Entity Sets
 - Relationships and Relationship Sets
 - Additional Features
- Conceptual Design using Entity Relationship Diagrams
 - for Large Enterprises
- Source: Ramakrishnan and Gehrke 2.1-2.7

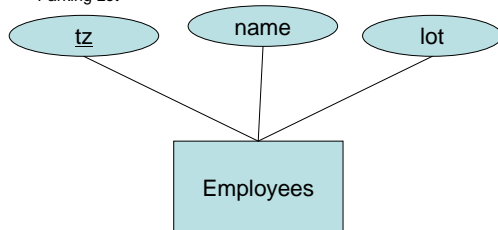
November 3, 2009

ISE 322: Database Systems

2

Employee Example and Diagram

- Employees have:
 - National Id
 - Name
 - Parking Lot



November 3, 2009

ISE 322: Database Systems

3

Relationships

- A **relationship** is an association between two or more entities
 - Relationships can be sets too

- Formally, a relationship set is of n-tuples where each entity belongs to some entity set:

$$\{(e_1, \dots, e_n) | e_1 \in E_1, \dots, e_n \in E_n\}$$

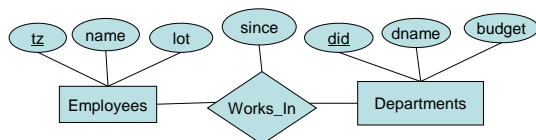
- This is read "A relationship is a set of tuples of the form e_1, e_2, \dots, e_n such that e_1 is a member of entity set E_1 , e_2 is a member of entity set E_2 , etc."

November 3, 2009

ISE 322: Database Systems

4

Relationship Example



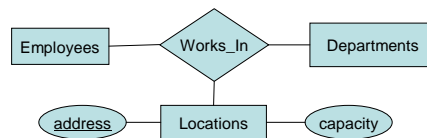
November 3, 2009

ISE 322: Database Systems

5

Relationships and Sets

- An **instance** of a relationship set is a relationship
- Relationships can binary:



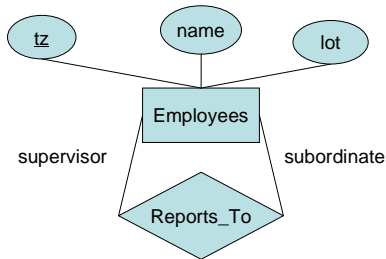
- Or Ternary
- Or more

November 3, 2009

ISE 322: Database Systems

6

Reflexive Relationships



- Sometimes a role label can help identify important aspects of a relationship

November 3, 2009

ISE 322: Database Systems

7

So far

- Entity Relationship Diagrams
 - Simple to Complex
 - Entities, Attributes, Entity Sets
 - Relationships and Relationship Sets
 - Additional Features
- Conceptual Design using Entity Relationship Diagrams
 - for Large Enterprises

November 3, 2009

ISE 322: Database Systems

8

Additional Features of ERD

- ERDs can also represent more complex aspects of relationships:
 - Key Constraints
 - Participation Constraints
 - Weak Entities
 - Inheritance
 - Aggregation

November 3, 2009

ISE 322: Database Systems

9

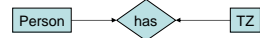
Key Constraints

- Types of Key Constraints:

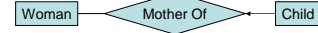
- One to one
- One to Many
- Many to Many

- Examples:

- One to One – 1 TZ to 1 person



- One to Many – 1 mother to Many children



- Many to Many – Many addresses to Many residents

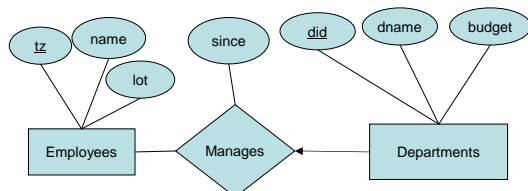


November 3, 2009

ISE 322: Database Systems

10

Key Constraint Example



November 3, 2009

ISE 322: Database Systems

11

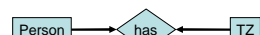
Participation Constraints

- Whether each member of an entity set must appear in the relationship

- Thick line on its side if yes

- Examples:

- Each person must have a TZ, every TZ is for a person



- Each child must have a mother



- Not everyone has an address, every address has someone there

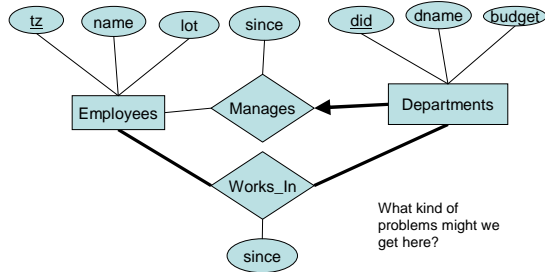


November 3, 2009

ISE 322: Database Systems

12

Participation Example



November 3, 2009

ISE 322: Database Systems

13

Weak Entities

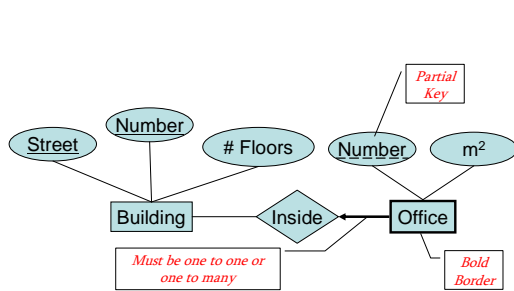
- These come up when there is something important which has no way of being identified except in terms of its relationship to another entity
- Example: We need to store data about offices in an office building
 - Each office has an office number and floor
 - ex. 308
 - Two offices in different buildings may have the same number
 - 308 in Building 1 and 308 in Building 2
 - We may identify an office based on its building

November 3, 2009

ISE 322: Database Systems

14

Weak Entities Example

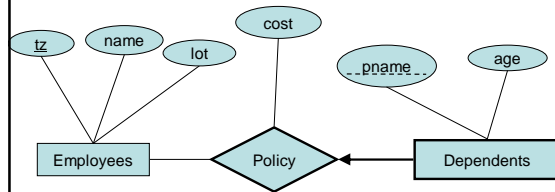


November 3, 2009

ISE 322: Database Systems

15

Weak Entity Example



November 3, 2009

ISE 322: Database Systems

16

Inheritance

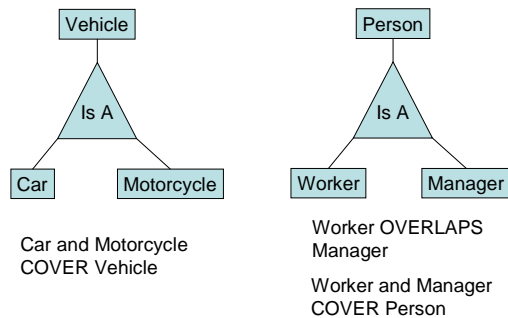
- ERDs can show hierarchical entity relationships
 - Inheritance in programming jargon
 - Limited to *Is A* relationships
- Examples:
 - A car *is a* vehicle
 - A motorcycle *is a* vehicle
 - A worker *is a* person
 - A manager *is a* person
- Has implicit containment semantics:
 - Every car/motorcycle entity is a vehicle as well
 - Every worker/manager is a person as well

November 3, 2009

ISE 322: Database Systems

17

Inheritance Examples



November 3, 2009

ISE 322: Database Systems

18

Aggregation

- Some complex relationships require relationships between relationships
 - Normally this is illegal in an ERD
 - Relationships can only connect entities
- We can allow such relationships by turning each instance of a relationship set into an entity
 - This is called **Aggregation**
 - We can then connect the aggregated relationship to other relationships
- BTW – it's hard to find simple examples of aggregation...
 - Think hard, because it might be doable with high order relationships instead

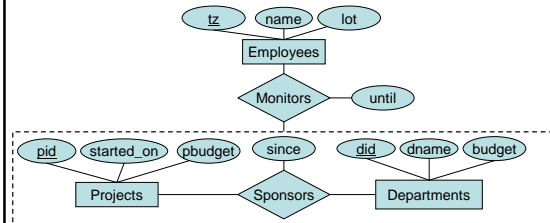
November 3, 2009

ISE 322: Database Systems

19

Aggregation Example

- Departments sponsor projects.
- There is an employee in charge of monitoring the project and its sponsorship until some date.

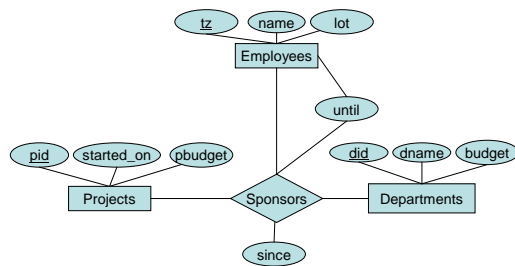


November 3, 2009

ISE 322: Database Systems

20

What else could we have done?



We need the aggregation to make the *until* attribute associated with the monitoring relationship alone.

November 3, 2009

ISE 322: Database Systems

21

So far

- Entity Relationship Diagrams
 - Simple to Complex
 - Entities, Attributes, Entity Sets
 - Relationships and Relationship Sets
 - Additional Features
- Conceptual Design using Entity Relationship Diagrams
 - for Large Enterprises

November 3, 2009

ISE 322: Database Systems

22

Conceptual Design using ERD

- Making good design using ERDs requires considering some tradeoffs:
 - Is it an **entity** or an **attribute**?
 - Is it an **entity** or a **relationship**?
 - Should the relationship be **binary** or **ternary**?
 - Should we use **aggregation** or a **higher order relationships**?
- Let's discuss some guidelines

November 3, 2009

ISE 322: Database Systems

23

Entity vs. Attribute

1. If an attribute must be multivalued (with unknown cardinality) then it must be an entity
 - **Example:** An Employee may have 1 or more addresses
 - Address must be an entity
 - If we know it's only 1-2, we can just make two Address attributes
2. If an attribute has structure or its own attributes, it must be an entity
 - **Example:** An Address must maintain a start date and an end date to indicate its validity period
 - Address must be an entity since the end date is not an attribute of the Employee
 - If it's multivalued too, the first reason applies anyway

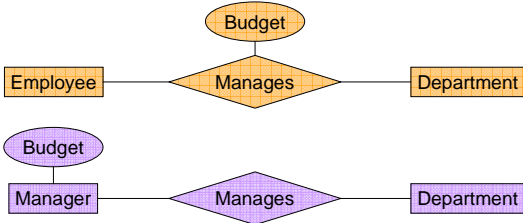
November 3, 2009

ISE 322: Database Systems

24

Entity vs. Relationship

- If a relationship needs attributes which are shared between relationship instances, it must be an entity
- Example:** A manager has a budget shared between multiple departments that she manages



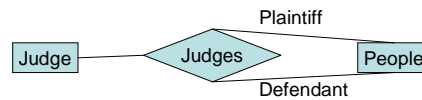
November 3, 2009

ISE 322: Database Systems

25

Binary vs. Ternary

- If a relationship is between three entities, a ternary (or higher) relationship is required
- Example:** The relationship between a Judge and two People



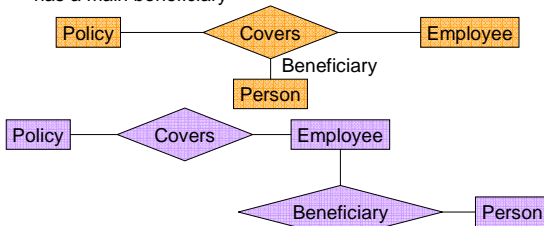
November 3, 2009

ISE 322: Database Systems

26

Binary vs. Ternary

- If relationship is really a series of binary relationships, then just show them
- Example:** A life insurance policy covers an individual and has a main beneficiary



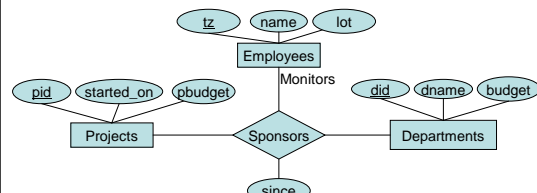
November 3, 2009

ISE 322: Database Systems

27

Aggregation vs. Ternary

- If the relationship doesn't require its own attributes, we can use ternary (or higher order) instead of aggregation
- Example:** If we didn't need the until attribute above, we could have made the relationship ternary instead



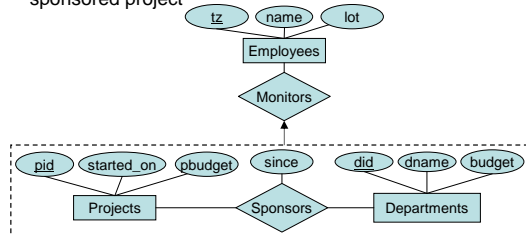
November 3, 2009

ISE 322: Database Systems

28

Aggregation vs. Ternary

- Some key constraints require aggregation
- Example:** If at most one employee can monitor each sponsored project



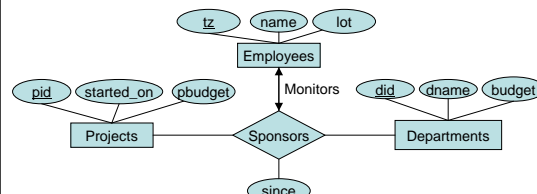
November 3, 2009

ISE 322: Database Systems

29

Aggregation vs. Ternary

- Some key constraints require aggregation
- Example:** If at most one employee can monitor each sponsored project



November 3, 2009

ISE 322: Database Systems

30

For a Team

- In a large organization, conceptual design is not done by one person
 - The team may not even be in the same location
 - They may not meet that often
- Usually separate teams will design separate aspects
 - But they all interact...
- This is why good planning **before** the design starts is essential
 - Determine central entities and relationships
 - Devise **interfaces** for the central entities or relationships ahead of time
 - Separation of responsibilities is key

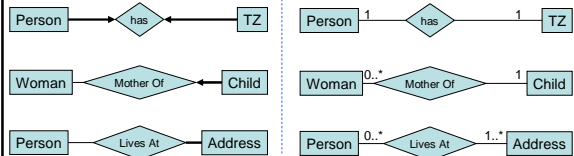
November 3, 2009

ISE 322: Database Systems

31

Tools and Techniques

- I presented here a particular style of ERDs
 - They are graphically simple and easy to learn
 - There are many other styles and techniques for ERDs
- Key constraints and participation constraints are expressed with arrows and heavy lines
 - The Information Systems Engineering book uses another style which makes them numerically explicit



November 3, 2009

ISE 322: Database Systems

32

Tools and Techniques

- Another common tool for design is the Unified Modeling Language
 - UML is a collection of documentation techniques
 - It includes a format for Entity Relationship Diagrams
 - Differs slightly from what I've presented
- You'll have a course on UML next semester

November 3, 2009

ISE 322: Database Systems

33

Conclusion

- Entity Relationship Diagrams
 - Simple to Complex
 - Entities, Attributes, Entity Sets
 - Relationships and Relationship Sets
 - Additional Features
- Conceptual Design using Entity Relationship Diagrams
 - for Large Enterprises

November 3, 2009

ISE 322: Database Systems

34

Word of the day

- Jargon – the language, especially the vocabulary, peculiar to a particular trade, profession, or group:

November 3, 2009

ISE 322: Database Systems

35