SK Alamgir Hossain

Assistant Professor
Computer Science & Engineering Discipline
Khulna University, Khulna, Bangladesh

http://alamgirhossain.com

CSE3203 - Software Engineering and Information System Design

Chapter 9

– Data Modeling

Chapter Outline

- Data Modeling
  - Entity
  - Relationships
  - ERD
  - Degree of Relationships
  - Cardinalities
  - Selecting Best Alternative Design Strategy

Lecture 17

Data Modeling

- Overview of Database Design
- The Entity-Relationship model
- ER Model Basics
- Degree of a relationship
- Strong versus Weak entity type
- Simple versus composite attributes
- Single-Valued versus Multivalued Attribute
- Entity vs. Attribute
- Entity vs. Relationship
- Binary vs. Ternary Relationships

Overview of Database Design

Khulna University

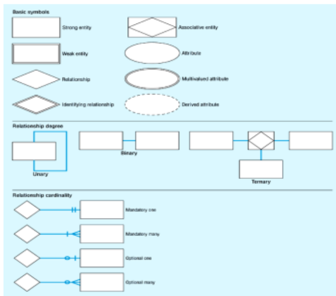
- Requirements Analysis: Understand what data will be stored in the database, and the operations it will be subject to.
- Conceptual Design: (ER Model is used at this stage.)
  - What are the entities and relationships in the enterprise?
  - What information about these entities and relationships should we store in the database?
  - What are the integrity constraints or business rules that hold?
- Logical Design: Convert the conceptual database design into the data model underlying the DBMS chosen for the application.
- Schema Refinement: (Normalization) Check relational schema for redundancies and anomalies.
- Physical Database Design and Tuning: Consider typical workloads and further refinement of the database design (v.g. build indices).
- Application and Security Design: Consider aspects of the application beyond data. Methodologies like UML often used for addressing the complete software development cycle.

The Entity-Relationship model

Khulna University

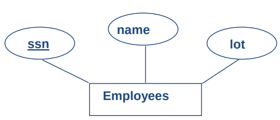
- The E-R model is a detailed, logical representation of the data for an organisation or business area
- It should be understandable to both the user and to the IT technologist
- The model must be as 'open' as possible and not tied to any technology or to any particular business methodology
- It must be flexible enough so that it can be used and understood in practically any environment where information is modelled
- It is expressed in terms of entities in the business environment, the relationships (or associations) among those entities and the attributes (properties) of both the entities and their relationships
- The E-R model is usually expressed as an E-R diagram

ER Model Basics: Notation



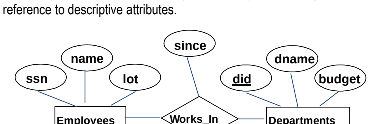
ER Model Basics (Contd.)

- Entity: Real-world object distinguishable from other objects. An entity is described using a set of attributes.
- Entity Set: A collection of entities of the same kind. E.g., all employees.
  - All entities in an entity set have the same set of attributes.
  - Each entity set has a key(a set of attributes uniquely identifying an entity).
  - Each attribute has a domain.



ER Model Basics (Contd.)

- Relationship: Association among two or more entities. E.g., Peter works in Pharmacy department.
- Relationship Set: Collection of similar relationships.
  - An **n-ary** relationship set R relates n entity sets E1 ... En; each relationship in R involves entities e1 $\in$ E1, ..., en $\in$ En
  - Same entity set could participate in different relationship sets, or in different "roles" in same set.
  - Relationship sets can also have descriptive attributes (e.g., the since attribute of Works_In). A relationship is uniquely identified by participating entities without reference to descriptive attributes.

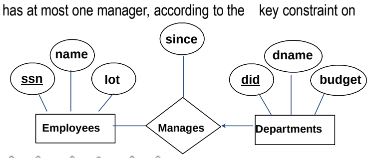


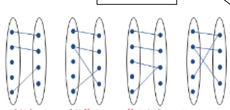
Attributes

- An attribute is a property or characteristic of an entity type, for example the entity EMPLOYEE may have attributes Employee_Name and Employee_Address.
- In ER diagrams place attributes name in an ellipse with a line connecting it to its associated entity
- Attributes may also be associated with relationships
- An attribute is associated with exactly one entity or relationship

Cardinality

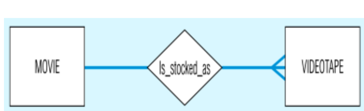
- Consider Works_In (in previous slide): An employee can work in many departments; a dept can have many employees.
- In contrast, each dept has at most one manager, according to the key constraint on Manages.





Cardinality constraints

- The number of instances of one entity that can or must be associated with each instance of another entity.
- If we have two entity types A and B, the cardinality constraint specifies the number of instances of entity B that can (or must) be associated with entity A
- e.g. a video store may stock more than one VIDEOTAPE for each MOVIE, this is a 'one-to-many' relationship as in the following Fig.



Minimum cardinality

- Yet there may be a more precise way of saying this
- The minimum cardinality of a relationship is the minimum number of instances of an entity B that may be associated with each instance of an entity A
- In our example, the minimum number of VIDEOTAPES of a MOVIE is zero (entity B is an optional participant in the 'Is_Stocked_As' relationship)
- This is signified by the symbol zero through the arrow near the VIDEOTAPE entity in the following Fig.

Maximum cardinality

- Is the maximum number of instances of an entity B that may be associated with each instance of entity A
- In the following Fig., the maximum cardinality for the VIDEOTAPE entity type is 'many' (an unspecified number greater than 1)
- This is indicated by the 'crow's foot' symbol on the arrow next to the VIDEOTAPE entity symbol

Degree of a relationship

- Is the number of entity types that participate in it.
- Thus 'Completes' has degree 2, since there are two participating entity types, EMPLOYEE and COURSE
- The three most common relationship degrees are unary (degree 1), binary (degree 2) and ternary (degree 3 –see following Fig.)
- Higher degree relationships are possible but rarely encountered in practice

Unary relationship

Khulna University

- Is between the instances of a single entity type (also called recursive relationships)
- 'Is_Married_To' is a one-to-one relationship between instances of the PERSON entity type
- 'Manages' is a one-to-many relationship between instances of the EMPLOYEE entity type

Binary relationships

Khulna University

- Between the instances of two entity types, and is the most common type of relationship encountered in data modelling. e.g. (one-to-one) an EMPLOYEE is assigned one PARKING_PLACE, and each PARKING_PLACE is assigned to one EMPLOYEE
- e.g. (one to many) a PRODUCT_LINE may contain many PRODUCTS, and each PRODUCT belongs to only one PRODUCT_LINE
- e.g. (many-to-many) a STUDENT may register for more than one COURSE, and each COURSE may have many STUDENTS

Ternary relationships

Khulna University

- A ternary relationship is a simultaneous relationship among the instances of 3 entity types
- It is the most common relationship encountered in data modelling
- The following Fig. shows a typical ternary relationship
- Here, vendors can supply various parts to warehouses

Ternary relationships

- The relationship 'Supplies' is used to record the specific PARTs supplied by a given VENDOR to a particular WAREHOUSE
- There are two attributes on the relationship 'Supplies', Shipping_Mode and Unit_Cost
- e.g. one instance of 'Supplies might record that VENDOR X can ship PART C to WAREHOUSE Y, that the Shipping_Mode is 'next_day_air' and the Unit_Cost is £5-00 per unit

Lecture 18

Data Modeling

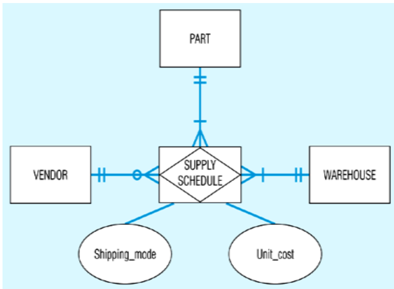
- Overview of Database Design
- The Entity-Relationship model
- ER Model Basics
- Degree of a relationship
- Strong versus Weak entity type
- Simple versus composite attributes
- Single-Valued versus Multivalued Attribute
- Entity vs. Attribute
- Entity vs. Relationship
- Binary vs. Ternary Relationships

An example using a ternary relationship

- PART and WAREHOUSE are mandatory participants in the relationship, whilst VENDOR is an optional participant
- The cardinality of each of the participating entities is mandatory one, since each SUPPLY_SCHEDULE instance must be related to exactly one instance of each of these participating entity types
- Each VENDOR can supply many PARTs to any number of WAREHOUSES, but need not supply any parts
- Each PART can be supplied by any number of VENDORs to more than one WAREHOUSE, but each part must be supplied by at least one vendor to a warehouse
- Each WAREHOUSE can be supplied with any number of PARTS from more than one VENDOR, but each warehouse must be supplied with at least one part

Ternary relationships

Khulna University



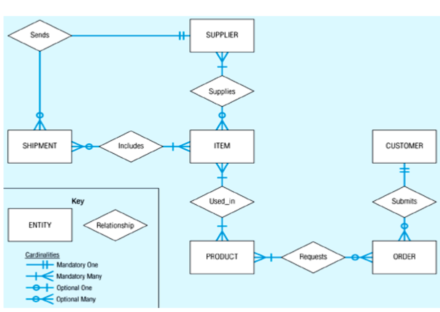
Ternary relationships

Khulna University

- We do not use diamond symbols on the lines from SUPPLY_SCHEDULE to the three entities, because these lines do not represent binary relationships
- It is recommended that all ternary (or higher) relationships are converted into associative entities (as in the Fig.), as it makes the representation of participation constraints (discussed later) easier
- Many CASE tools cannot represent ternary relationships, so you must represent the ternary relationship with an associative entity and three binary relationships

Sample E-R Diagram

Khulna University



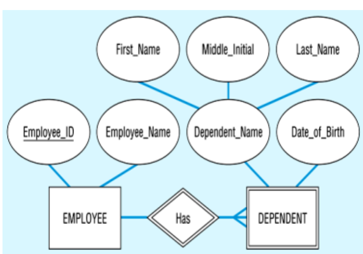
Strong versus Weak entity type

Khulna University

- Most of the basic entity types are classified as strong entity types [Rectangle] – one that exists independently from other entity types (such as EMPLOYEE)
- Always have a unique characteristic (identifier) – an attribute or combination of attributes that uniquely distinguish each occurrence of that identity
- A weak entity type [[Double Rectangle]] – existence depends on some other entity type. It has no meaning in the ER diagram without the entity on which it depends (such as DEPENDENT)
- The entity type on which the weak entity type depends is called the Identifying owner (or owner for short).
- Identifying relationship is the relationship between a weak entity type and and its owner (such as 'Has' in the following Fig.)
- Weak entity identifier is its partial identifier (double underline) combined with that of its owner. During a later design stage dependent name will be combined with Employee_ID (the identifier of the owner) to form a full identifier for DEPENDENT.

Example of a weak entity

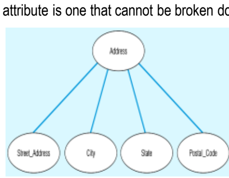
Khulna University



Simple versus composite attributes

Khulna University

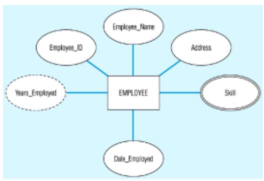
- Some attributes can be broken down into meaningful component parts, such as Address, which can be broken down into Street_Address, City..etc.
- The component attributes may appear above or below the composite attribute on an ER diagram
- Provide flexibility to users, as can refer to it as a single unit or to the individual components
- A simple (atomic) attribute is one that cannot be broken down into smaller components



Single-Valued versus Multivalued Attribute

Khulna University

- It frequently happens that there is an attribute that may have more than one value for a given instance, e.g. EMPLOYEE may have more than one Skill.
- A multivalued attribute is one that may take on more than one value – it is represented by an ellipse with double lines

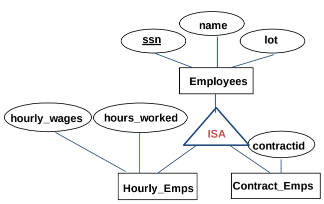


Chapter 9 – Data Modeling http://alamgirhossain.com Slide: 28

ISA (`is a') Hierarchies

Khulna University

- If we declare A **ISA** B, every A entity is also considered to be a B entity.



Chapter 9 – Data Modeling http://alamgirhossain.com Slide: 29

Entity vs. Attribute

Khulna University

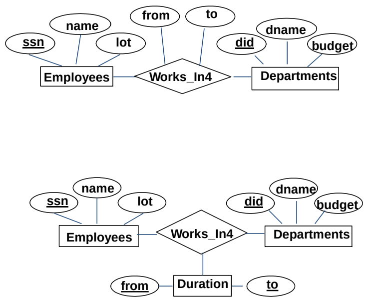
- Should address be an attribute of Employees or an entity (connected to Employees by a relationship)?
- Depends upon the use we want to make of address information, and the semantics of the data:
  - If we have several addresses per employee, address must be an entity (since attributes cannot be set-valued).
  - If the structure (city, street, etc.) is important, e.g., we want to retrieve employees in a given city, address must be modeled as an entity (since attribute values are atomic).

Chapter 9 – Data Modeling http://alamgirhossain.com Slide: 30

Entity vs. Attribute (Contd.)

Khulna University

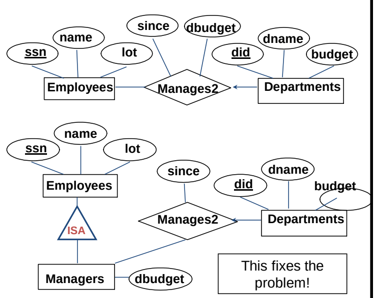
- Works_In4 does not allow an employee to work in a department for two or more periods (a relationship is identified by participating entities).
- Similar to the problem of wanting to record several addresses for an employee: We want to record several values of the descriptive attributes for each instance of this relationship. Accomplished by introducing new entity set, Duration.



Entity vs. Relationship

Khulna University

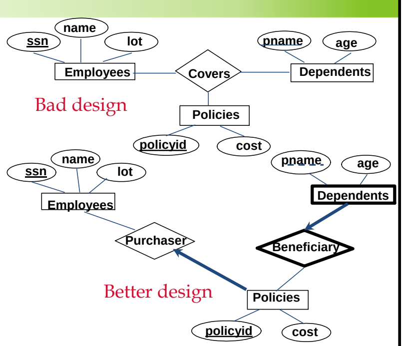
- First ER diagram OK if a manager gets a separate discretionary budget for each dept.
- What if a manager gets a discretionary budget that covers all managed depts?
  - Redundancy: dbudget stored for each dept managed by manager.
  - Misleading: Suggests dbudget associated with department-mgr combination.



Binary vs. Ternary Relationships

Khulna University

- Suppose:
  - A policy cannot be owned by more than one employee.
  - Every policy must be owned by some employee.
  - Dependent is a weak entity set, identified by policyId.



Binary vs. Ternary Relationships (Contd.)

- Previous example illustrated a case when two binary relationships were better than one ternary relationship.
- An example in the other direction: a ternary relation Contracts relates entity sets Parts, Departments and Suppliers, and has descriptive attribute qty. No combination of binary relationships is an adequate substitute:
  - Although S "can-supply" P, D "needs" P, and D "deals-with" S, all these do not imply that D has agreed to buy P from S (because D could buy P from another supplier).

Chapter 9 – Data Modeling http://alamgirhossain.com Slide: 34

Summary of Conceptual Design

- Conceptual design follows requirements analysis,
  - Yields a high-level description of data to be stored
- ER model popular for conceptual design
  - Constructs are expressive, close to the way people think about their applications.
- Basic constructs: entities, relationships, and attributes (of entities and relationships).
- Some additional constructs: weak entities, ISA hierarchies, and aggregation.
- Note: There are many variations on ER model.

Chapter 9 – Data Modeling http://alamgirhossain.com Slide: 35

Summary of ER (Contd.)

- Several kinds of integrity constraints can be expressed in the ER model: key constraints, participation constraints, and overlap/covering constraints for ISA hierarchies. Some foreign key constraints are also implicit in the definition of a relationship set.
  - Some constraints (notably, functional dependencies) cannot be expressed in the ER model.
  - Constraints play an important role in determining the best database design for an enterprise.

Chapter 9 – Data Modeling http://alamgirhossain.com Slide: 36

Summary of ER (Contd.)

- ER design is subjective. There are often many ways to model a given scenario! Analyzing alternatives can be tricky, especially for a large enterprise. Common choices include:
  - Entity vs. attribute, entity vs. relationship, binary or n-ary relationship, whether or not to use ISA hierarchies, and whether or not to use aggregation.
- Ensuring good database design: resulting relational schema should be analyzed and refined further. FD information and normalization techniques are especially useful.

THANK YOU