

Data Modeling with the Entity Relationship Model

CS157A

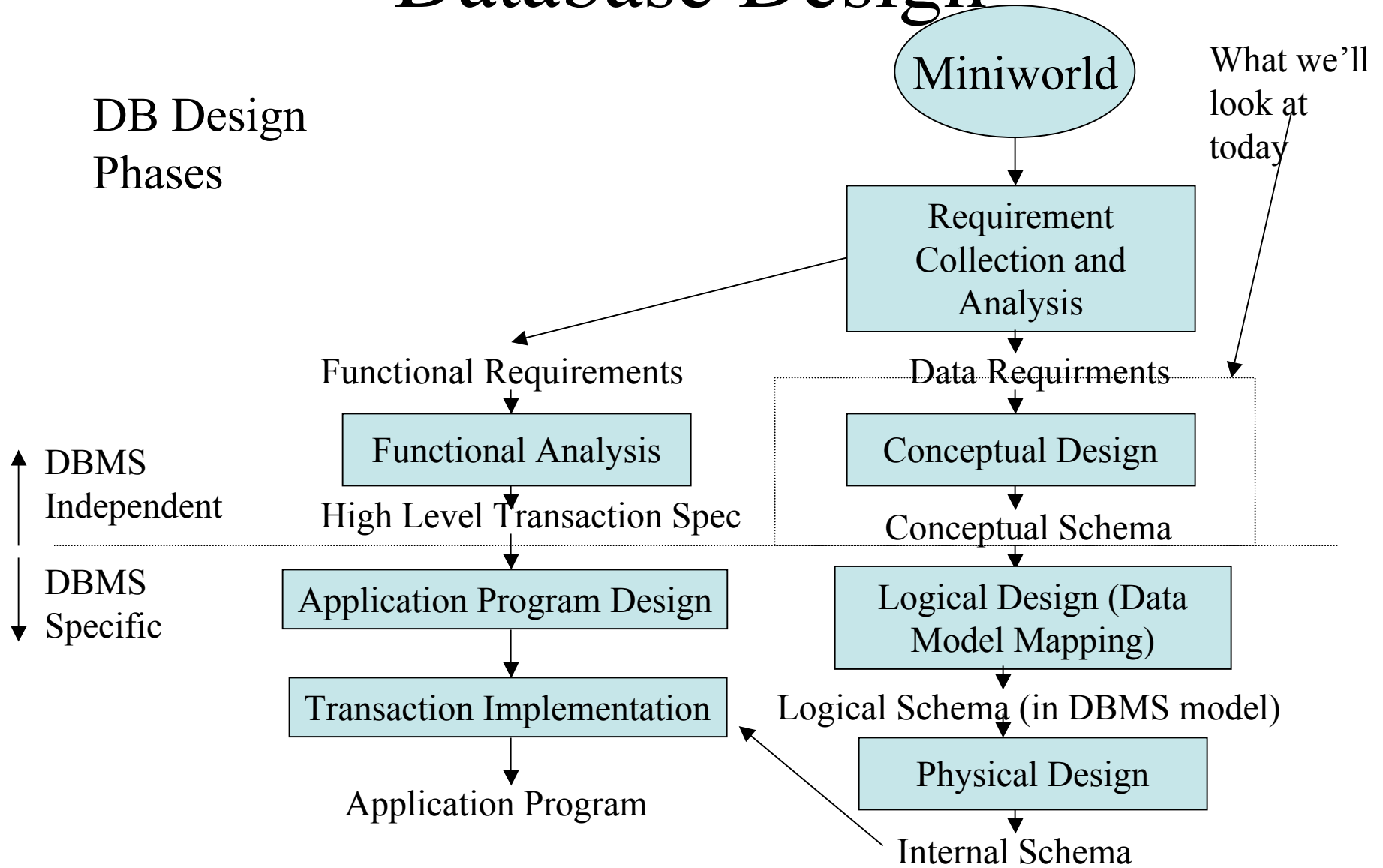
Chris Pollett

Sept. 7, 2005.

Outline

- Conceptual Data Models and Database Design
- An Example Application
- Entity Types, Sets, Attributes and Keys
- Relationship Types, Sets, Roles and Constraints

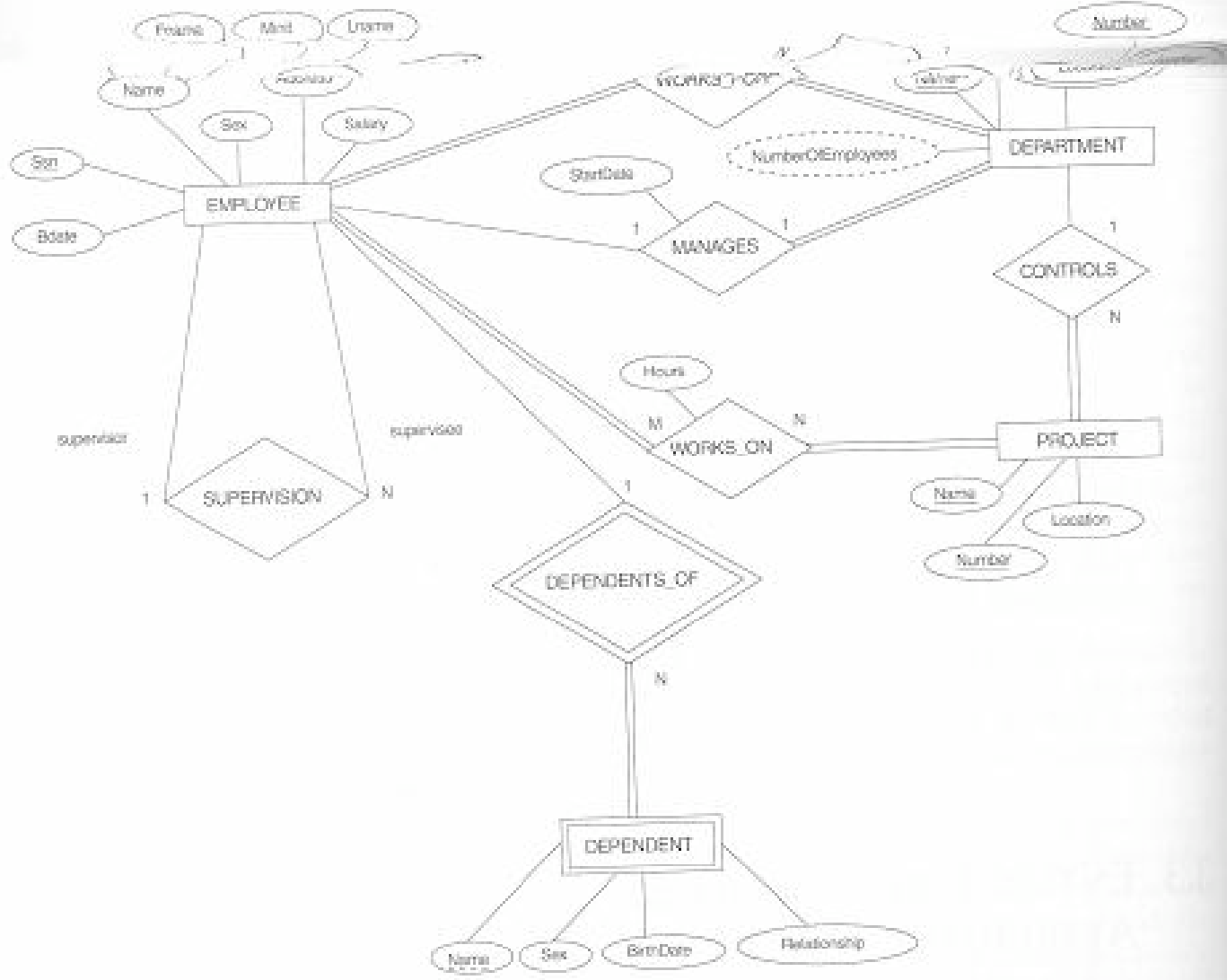
Conceptual Data Models and Database Design



Example Application

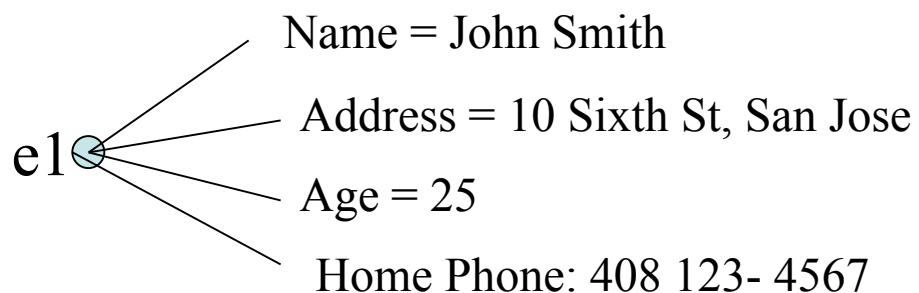
- Want to model a DB for a COMPANY
- A COMPANY keeps track of its employees, departments, and projects.
- A possible description of its mini-world after the requirement analysis phase might be:
 - The company is organized into departments.
 - Departments have a unique name, id, and manager who has a start date.
 - Departments might have several locations
 - Departments control a number of projects.
 - Each project has a name, number, and single location
 - Employee name, SSN, address, salary, sex, and birthdate must be recorded.
 - An employee works in one department but may work on several projects.
 - We must keep track of numbers of hours/week each employee works on a given project.
 - Each employee has a supervisor
 - We must keep track of the name, sex and relationship of each dependent of each employee for insurance purposes.

ER Diagram



Entity

- An **entity** is the basic object in the ER model.
- Represents a real world thing which has some independent existence.
- Each entity has **attributes**, which are the particular properties that describe the entity.



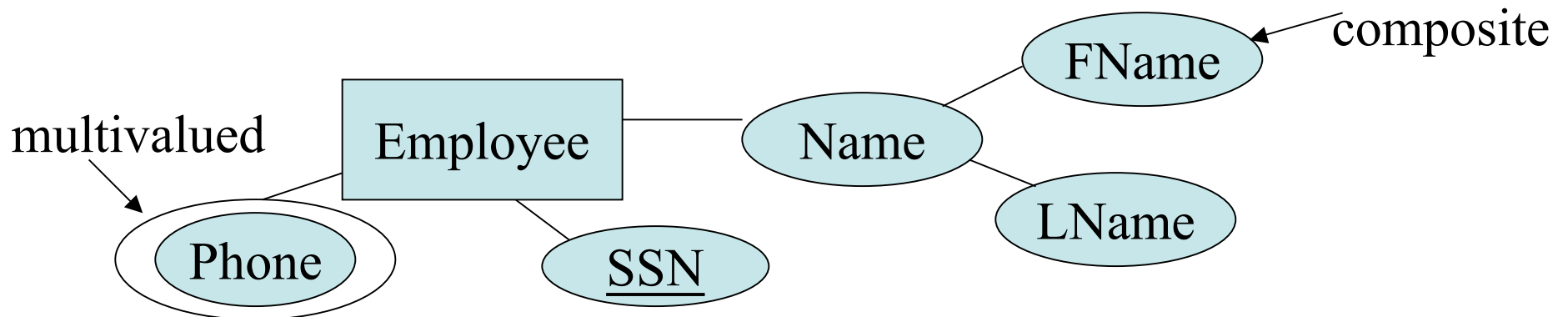
Example
employee entity

Simple, Composite, Multivalued, etc Attributes

- **Simple** (aka atomic): Attributes can't be divided into smaller parts. Ex: Gender
- **Composite** can be split into smaller parts. For example, address can be split into street address, city, state, and zip
- **Multivalued** attributes can have more than one value: CollegeDegrees, PhoneNumbers, etc.
- **Stored attributes** are physically stored in a DB. Ex. BirthDate. Other attributes like Age are **derived attributes** calculated from stored attributes.
- If an entity doesn't have a value for a particular attribute then say it has a **null value** for that attribute.
- Finally, **complex** attributes are attributes which are nestings of composite and multivalued attributes.

Entity Type, Entity Set

- An **entity type** is the collection of all entities with the same attributes.
- An **entity set** is subset of an entity type consisting of those entities at some fixed time in the DB.
- For example, EMPLOYEE might be an entity type and the e1 = (John Smith,...) example of the earlier slide is an element of this type.
- Represented in ER diagram with a box, with ovals for attributes.

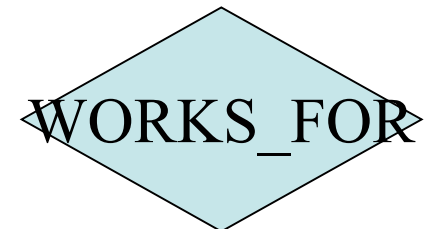


Key Attributes

- A **key attribute** is an attribute whose values are distinct for each entity in the entity set.
- For example, SSN for EMPLOYEE.
- We underline such attributes in the ER model.
- A key might be made of a set of attributes.
- For example, First Name and Last Name
- To be a key, this set must be minimal.

Relationship Types, Sets, Roles and Constraints

- Now we model relationships between entities.
- A **relationship type** R among entity types E_1, \dots, E_n defines a set of associations (called a **relationship set**) among entities in these entity types.
- We write $r1=(e_1, \dots, e_n)$ for a particular **relationship instance**.
- We can view R as a subset of $E_1 \times E_2 \times \dots \times E_n$
- We say that each of the E_i **participates** in the relationship.
- In ER model drawn with a diamond:



Properties of Relationships

- The **degree** of the relationship type is the number of entity types participating in the relationship type. For example for WORKS_FOR the degree is two.
- Each entity type participating in a relationship might have a **role name**. For example, supervisor, and supervisee.
- Role names are useful for **recursive relationship types**. These are relationship types where a single entity type participate more than once. For example, WORKS_FOR.

Constraints on Relationships

- We'll usually be interested in binary relationships.
- The **cardinality ratio** specifies the maximum number of times an entity can participate in a relationship. Typical values are: 1:1, 1:N, N:1, and N:M.
- These numbers can be marked in the diagram on the edges out of the diamond.

Participation Constraints

- A **participation constraint** specifies the minimum number of relationship instances an entity must belong to.
- There are two types of constraints that are considered: **Total** and **partial**.
- The first means each entity must participate in some relationship instance. The latter means each entity may or may not participate in a relationship instance.

Relationship Attributes

- Relationships are allowed to have attributes.
- These are also drawn with ovals.

