Database I

(60012301-1)

Lecture 3: Data Modeling Using the Entity-Relationship (ER) Model

Dr. Obead Alhadreti

Outline

- Overview of Database Design Process
- Example Database Application (COMPANY)
- ER Model Concepts
  - Entities and Attributes
  - Entity Types, Value Sets, and Key Attributes
- ER Diagrams - Notation
- ER Diagram for COMPANY Schema

Overview of Database Design Process

- Two main activities:
  - Database design
  - Applications design
- Focus here on database design
  - To design the conceptual schema for a database application
- Applications design focuses on the programs and interfaces that access the database
  - Generally considered part of software engineering

What is Database Design?

- **Database design** is the process of producing a detailed **data model** of a database.

- Database design involves identifying the existing relationships between separate pieces of data and mapping out those relationships in an organized way that makes sense.

- After analysis, Gather all the essential data required and understand how the data are related.

Main Phases of Database Design

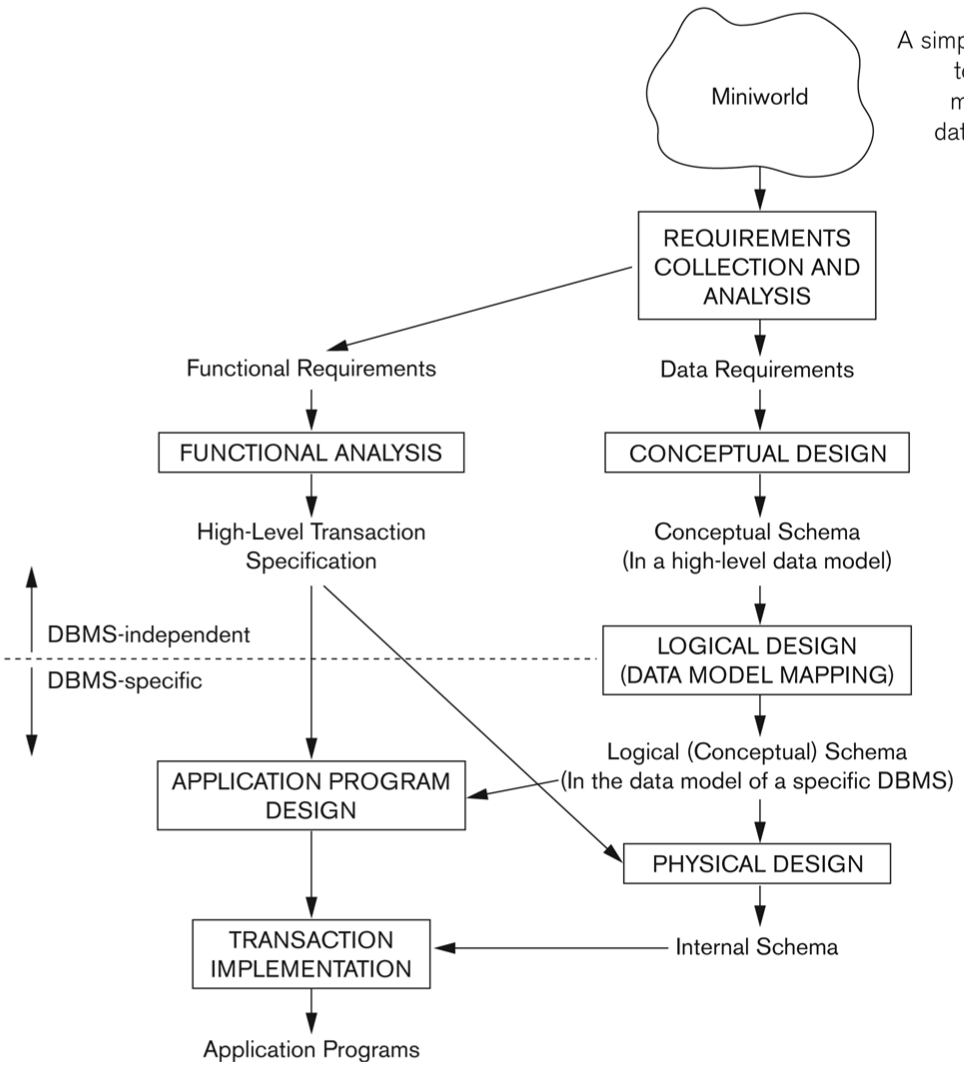

Miniworld
REQUIREMENTS COLLECTION AND ANALYSIS
Functional Requirements
Data Requirements
FUNCTIONAL ANALYSIS
CONCEPTUAL DESIGN
High-Level Transaction Specification
Conceptual Schema (In a high-level data model)
DBMS-independent
DBMS-specific
LOGICAL DESIGN (DATA MODEL MAPPING)
Logical (Conceptual) Schema (In the data model of a specific DBMS)
APPLICATION PROGRAM DESIGN
PHYSICAL DESIGN
TRANSACTION IMPLEMENTATION
Internal Schema
Application Programs

Figure 3.1
A simplified diagram to illustrate the main phases of database design.

Overview of Database Design Process (Contd.)

1. Requirements Collection and Analysis

- Database designers interview prospective database users to understand and document their **data requirements.**
- **Functional requirements:** user defined operations.
- **The output is:** A set of requirements.

2. Conceptual Design

- Detailed description of the entities, attributes, and their relationships.
- **The output is:** A conceptual schema (described using a conceptual data model like ***Entity-Relationship (ER) model***.

Overview of Database Design Process (Contd.)

3. Logical Design

- Mapping a conceptual schema (like ER model) into logical schema to provide a much detail description.
- **The output is:** A logical schema (described using a logical data model specific to the DBMS like relational model).

4. Physical Design

- Internal storage structures, file organizations, indexes, access paths, and physical design parameters for the database files are specified.
- **The output is:** An internal (physical) schema (described using a physical data model).

Example COMPANY Database

- We need to create a database schema design based on the following (simplified) **requirements** of the COMPANY Database:
  - The company is organized into DEPARTMENTs.
  - Each department has a name, number and an employee who *manages* the department. We keep track of the start date of the department manager. A department may have several locations.
  - Each department *controls* a number of PROJECTs.
  - Each project has a unique name, unique number and is located at a single location.

Example COMPANY Database (Contd.)

- We store each EMPLOYEE's social security number, address, salary, sex, and birthdate.
  - Each employee *works for* one department but may *work on* several projects.
  - We keep track of the number of hours per week that an employee currently works on each project.
  - We also keep track of the *direct supervisor* of each employee.
- Each employee may *have* a number of DEPENDENTs.
  - For each dependent, we keep track of their name, sex, birthdate, and relationship to the employee.

Entity-Relationship (ER) Model

- The ER model is a popular high-level conceptual data model, which is a representation of the structure of the database. It was proposed by Peter Chen in 1976
- The ER model aim to illustrate how relationships between entities are defined and refined.
- The ER model describes data as *entities*, *relationships*, and *attributes.*
- The diagrammatic ***notation*** associated with the ER model is known as **ER diagrams**.

Entities

- Entity: is a *thing* or *object* in the real world with an *independent existence.*
- An entity may be an object with a *physical existence* (for example, a particular person, car, house, or employee)
- It may also be an object with a *conceptual existence* (for instance, a company, a job, or a university course).
- Each entity has ***attributes*** that give them their identity.

Attributes

- Attributes: are properties used to describe an entity.
- For example, an EMPLOYEE entity may be described by the employee's name, age, address, salary, and job.
- A specific entity will have a value for each of its attributes.
  - For example a specific employee entity may have Name='John Smith', SSN='123456789', Address ='731, Fondren, Houston, TX', Sex='M', BirthDate='09-JAN-55'
- Each attribute has a ***value*** *set* (or data type) associated with it – e.g. integer, string, subrange, enumerated type, …

Attributes

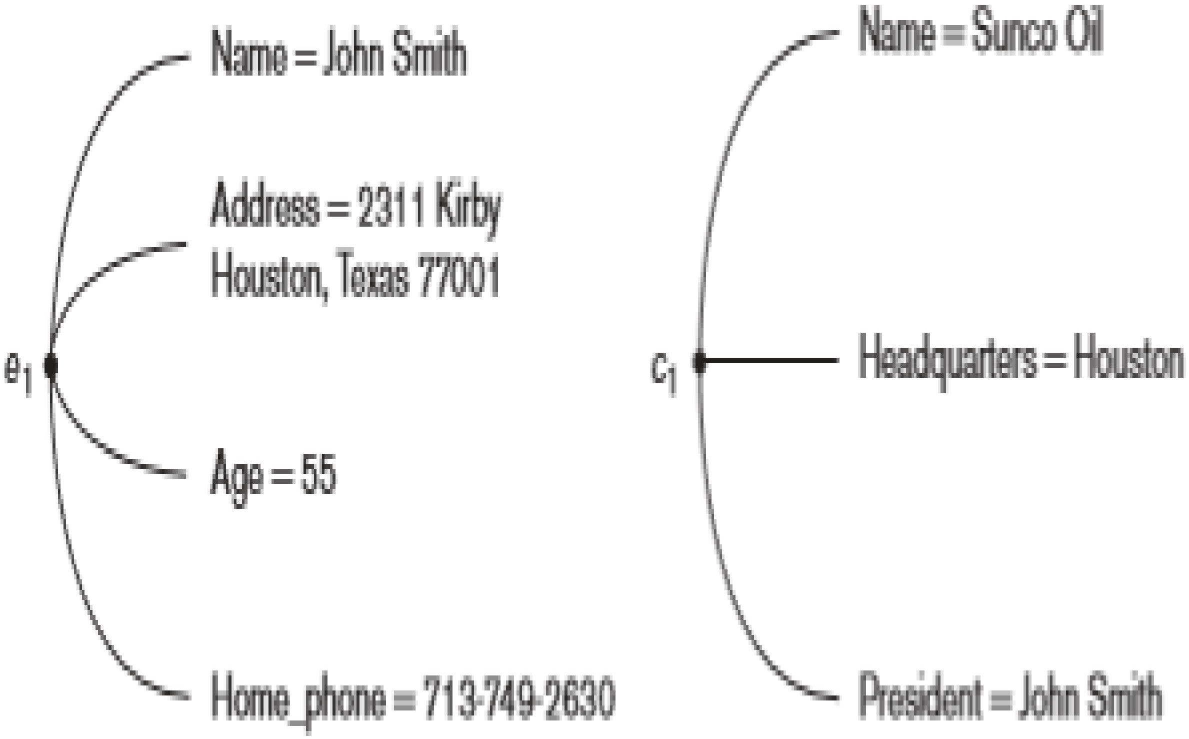


Figure 7.3
Two entities, EMPLOYEE e_1, and COMPANY c_1, and their attributes.

Types of Attributes

- Simple versus Composite attributes
  - Simple attribute: entity has a single atomic value for the attribute.

  - Composite attributes: The attribute can be divided into smaller subparts.
  - For example, the Address can be subdivided into (Apt#, House#, Street, City, State, ZipCode, Country).
  - Name(FirstName, MiddleName, LastName).
  - Composition may form a hierarchy where some components are themselves composite.

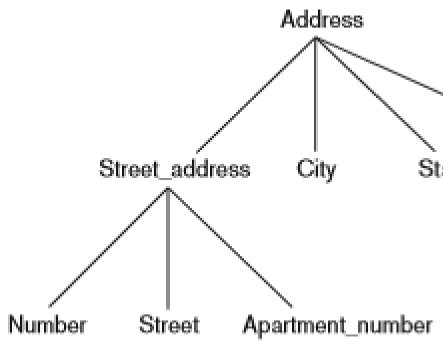


Figure 3.4
A hierarchy of composite attributes.

[3] Zip Code is the name used in the United States for a five-digit postal code, such as 76019, which can be extended to nine digits, such as 76019-0015. We use the five-digit Zip in our examples.

Types of Attributes (Contd.)

- Single-Valued versus Multi-Valued Attributes
  - Single-valued attributes: Most attributes have a single value for a particular entity; such attributes are called single-valued. For example, Age, height.
  - Single-valued attributes: An entity may have multiple values for that attribute. For example, Color of a CAR or PreviousDegrees of a STUDENT.
    - Denoted as {Color} or {PreviousDegrees}.
- Stored versus Derived Attributes
  - For example: Birth_date is a stored attribute.
  - Age can be derived from Birth_date. Therefore, Age is a called a derived attribute.

Types of Attributes (Contd.)

- NULL Values
  - A particular entity may not have an applicable value for an attribute (*not applicable value*).
  - For example, a College_degrees attribute applies only to people with college degrees. For such situations, a special value called NULL is created.
  - NULL can also be used if we do not know the value of an attribute for a particular entity (Unknown)- missing or really unknown.

- Complex Attributes
  - Mixing with composite and multivalued. For example, address and phones.

Entity Type

- Entity Type: a collection of entities that have the same attributes.

- So we may say **a name of table is an entity type.**

- Each entity type in the database is described by its name and attributes.

- Entity Instance: An entity instance is a single occurrence of an entity.

Entity Set

- Entity Set: The collection of all entities of a particular entity type in the database at any point in time is called an **entity set** or **entity collection.**
- Entity set is the current *state* of the entities of that type that are stored in the database.

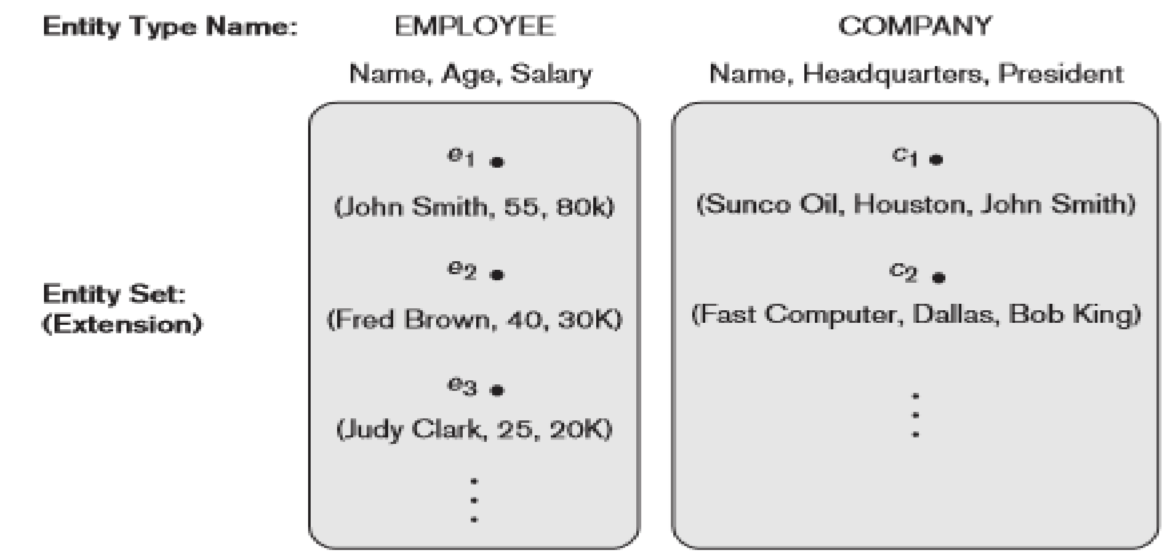


Figure 3.6
Two entity types, EMPLOYEE and COMPANY, and some member entities of each.

Exercise 1

Table name is STUDENT

ID	Name	Age
1	Ram	12
2	Sam	13

- Find out the entity type, entity attributes, entity instances, and entity set for the above table.

Key Attributes of an Entity Type

- Key Attributes are attributes whose values are distinct for each individual entity in the entity set.

- For example, the Name attribute is a **key** of the COMPANY entity type because no two companies are allowed to have the same name.

- An entity type may have more than one key, meaning that the *combination* of the attribute values must be distinct for each entity.
  - For example, The CAR entity type may have two keys:
    - VehicleIdentificationNumber (popularly called VIN)
    - VehicleTagNumber (Number, State), aka license plate number.

Key Attributes of an Entity Type (Contd.)

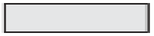
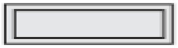
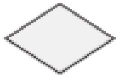
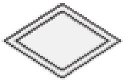
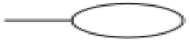
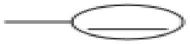
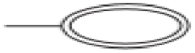
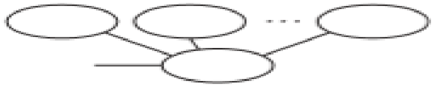
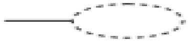
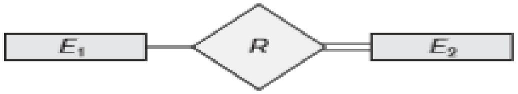
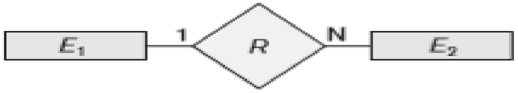
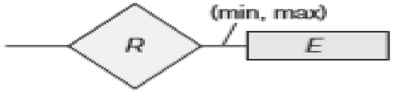
- A key attribute may be composite.
  - VehicleTagNumber is a key of the CAR entity type with components (Number, State).

- Must be minimal.

- Value Sets (domain of values)- specifies the set of values that may be assigned to that attribute for each individual entity. Not represented in ER, but specified in integer, Boolean, float, string, etc.

SUMMARY OF ER-DIAGRAM NOTATION FOR ER SCHEMAS

Figure 7.14
Summary of the notation for ER diagrams.

Symbol	Meaning
	Entity
	Weak Entity
	Relationship
	Indentifying Relationship
	Attribute
	Key Attribute
	Multivalued Attribute
	Composite Attribute
	Derived Attribute
	Total Participation of E_2 in R
	Cardinality Ratio 1: N for E_1:E_2 in R
	Structural Constraint (min, max) on Participation of E in R

Displaying an Entity type in ER diagram

- In ER diagrams, an entity type is displayed in a rectangular box.

- Attributes are displayed in ovals
  - Each attribute is connected to its entity type
  - Components of a composite attribute are connected to the oval representing the composite attribute
  - Each key attribute is underlined
  - Multivalued attributes displayed in double ovals

- See CAR example on next slide

Entity Type CAR with two keys and a corresponding Entity Set

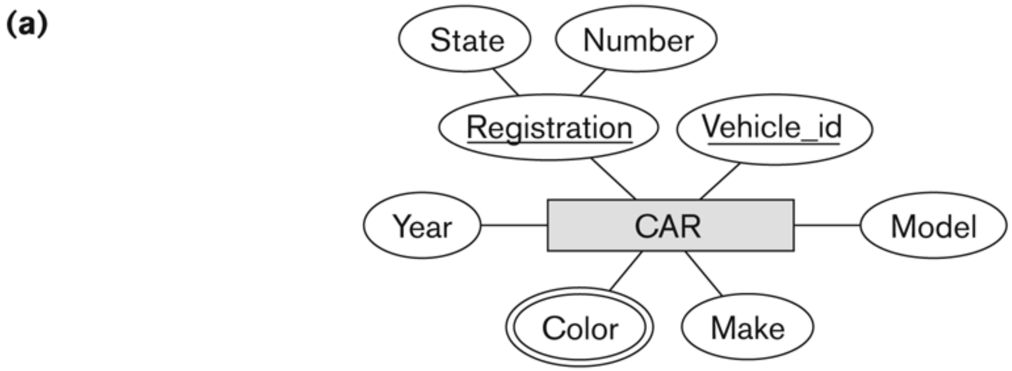


Figure 3.7
The CAR entity type with two key attributes, Registration and Vehicle_id. (a) ER diagram notation. (b) Entity set with three entities.

(b)

CAR
Registration (Number, State), Vehicle_id, Make, Model, Year, {Color}

CAR_1
((ABC 123, TEXAS), TK629, Ford Mustang, convertible, 2004 {red, black})

CAR_2
((ABC 123, NEW YORK), WP9872, Nissan Maxima, 4-door, 2005, {blue})

CAR_3
((VSY 720, TEXAS), TD729, Chrysler LeBaron, 4-door, 2002, {white, blue})

⋮

Initial Design of Entity Types for the COMPANY Database Schema

- Based on the requirements, we can identify four initial entity types in the COMPANY database:
  - DEPARTMENT
  - PROJECT
  - EMPLOYEE
  - DEPENDENT

- Their initial design is shown on the following slide

- The initial attributes shown are derived from the requirements description

Initial Design of Entity Types: EMPLOYEE, DEPARTMENT, PROJECT, DEPENDENT

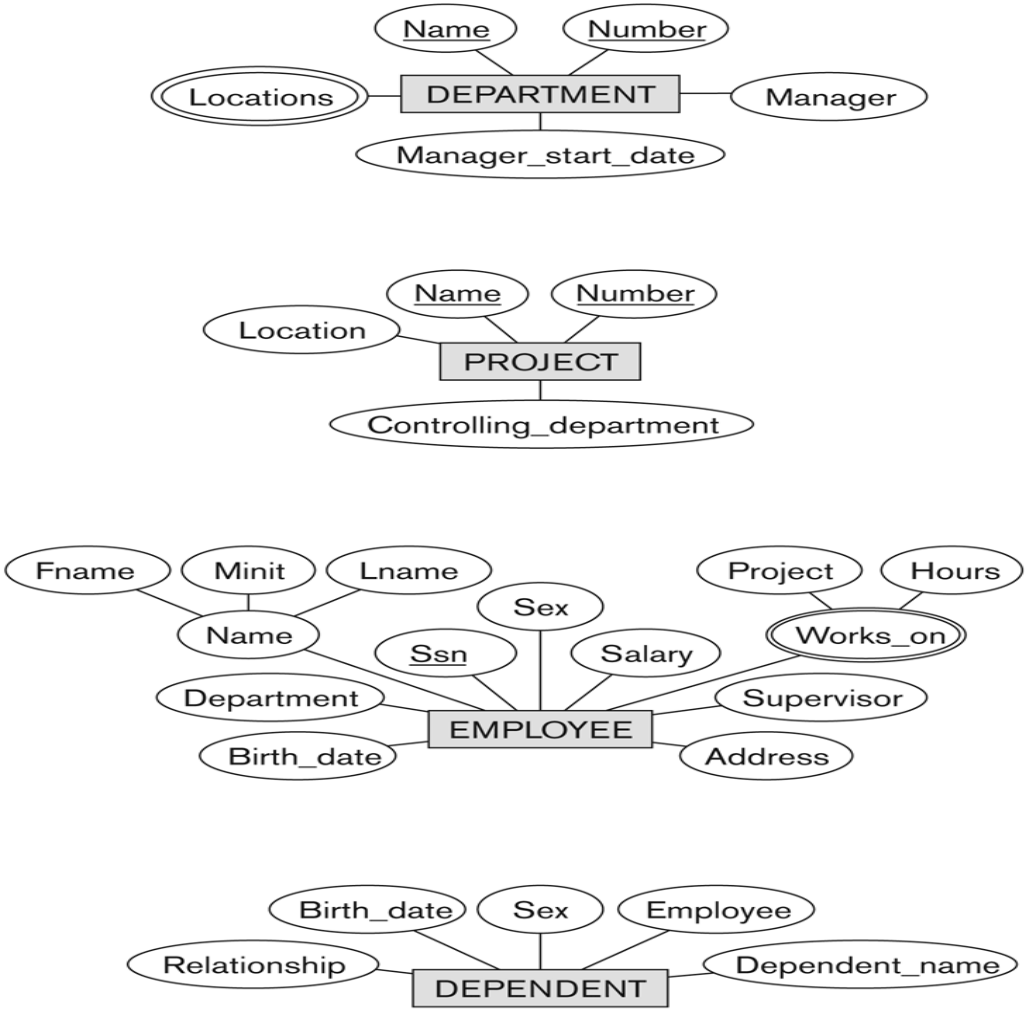


Figure 3.8
Preliminary design of entity types for the COMPANY database. Some of the shown attributes will be refined into relationships.

Initial Design of Entity Types for the COMPANY Database Schema

- Based on the requirements, we can identify four initial entity types in the COMPANY database:
  - DEPARTMENT
  - PROJECT
  - EMPLOYEE
  - DEPENDENT
- Their initial design is shown on the following slide
- The initial attributes shown are derived from the requirements description

Initial Design of Entity Types:
EMPLOYEE, DEPARTMENT, PROJECT, DEPENDENT

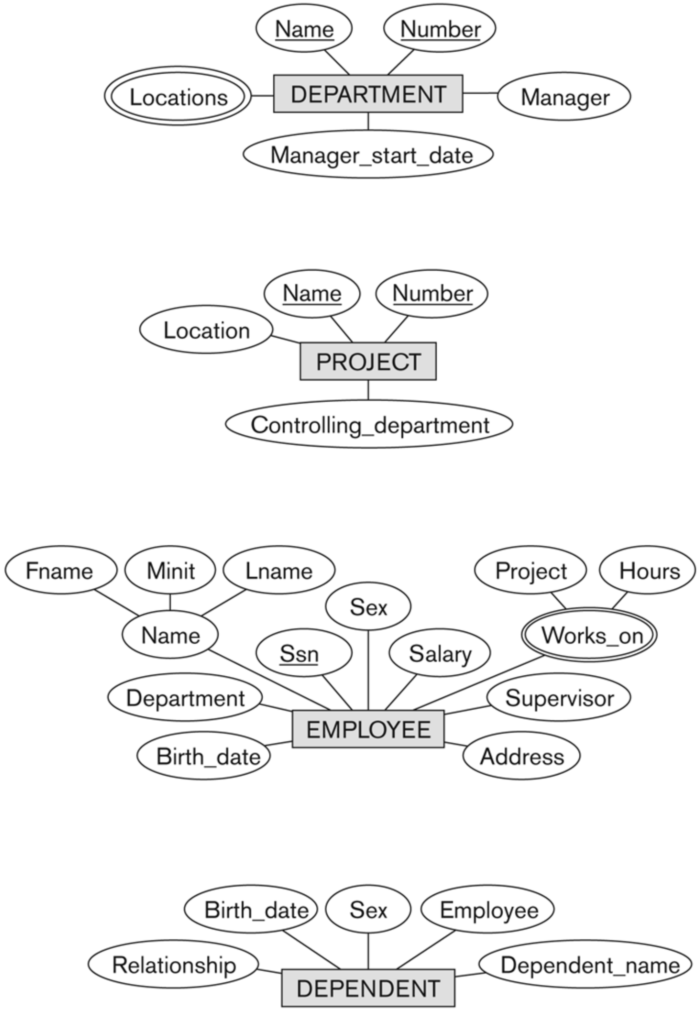


Figure 3.8
Preliminary design of entity types for the COMPANY database. Some of the shown attributes will be refined into relationships.

Exercise 2

Draw an initial conceptual design of the entity type: Students

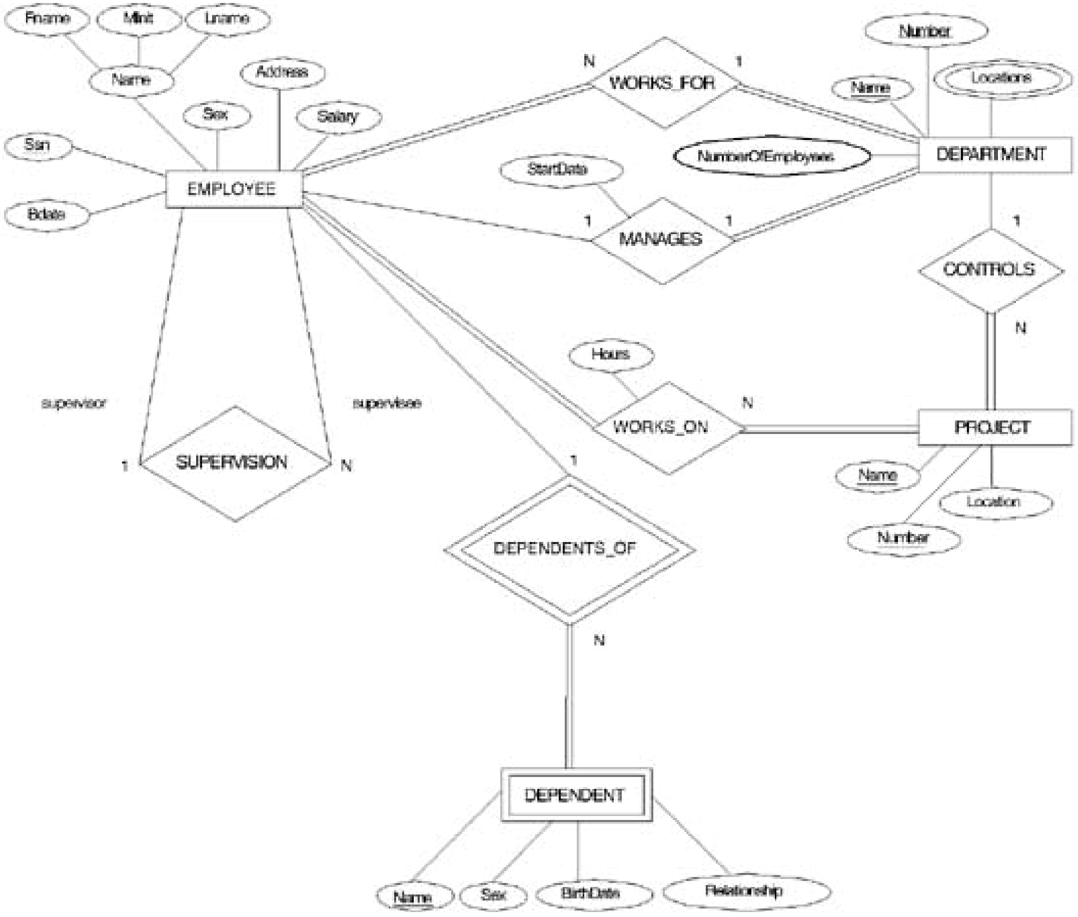
ID	First Name	Last name	Bdate	Age	sec
1	Ram	Jack	1980	12	M
2	Sam	Michel	1989	13	M

Refining the initial design by introducing **relationships**

- The initial design is typically not complete
- Some aspects in the requirements will be represented as **relationships**
- ER model has three main concepts:
  - Entities (and their entity types and entity sets)
  - Attributes (simple, composite, multivalued)
  - Relationships (and their relationship types and relationship sets)

Refining the initial design by introducing **relationships**

- We will introduce relationship concepts next lecture


Fname
Minit
Lname
Name
Address
Sex
Salary
Ssn
Bdate
EMPLOYEE
N
WORKS_FOR
1
Number
Name
Locations
DEPARTMENT
NumberOfEmployees
StartDate
1
MANAGES
1
1
CONTROLS
N
Hours
WORKS_ON
N
PROJECT
Name
Location
Number
supervisor
supervisee
1
SUPERVISION
N
1
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
BirthDate
Relationship