Database II

Lecture 1

Recap of database background

Dr. Doaa Elzanfaly

Recap of database background

- Basic Definitions
- DB System Architecture
- DBDLC
- Data Modeling
- Relational Data Model
  - Schema, relation, attribute, key …
  - Relational Model Constraints & Operations
  - Relational Algebra & SQL
  - ERD & ERD Mapping
  - Normalization

Basic Definitions

- **Data**

  Known *facts* that can be *recorded* and have an *implicit meaning*.

- **Database**

  A collection of related data.

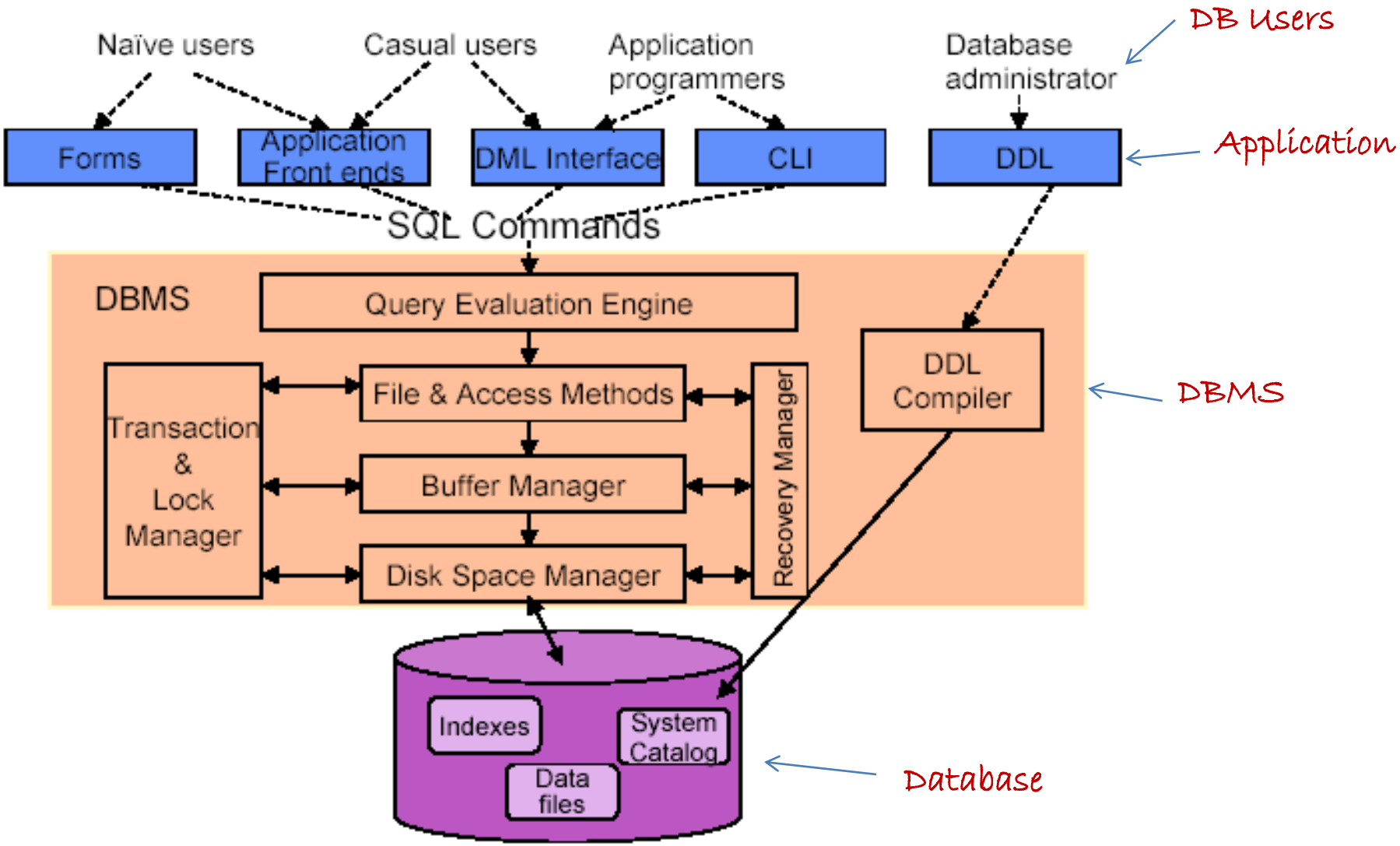
- **Database Management System (DBMS)**

  A software package/ system to facilitate the *creation*, *manipulation*, and *maintenance* of a computerized database.

- **Database System**

  The *DBMS* software together with the *data itself*. Sometimes, the applications are also included.

Database System Environment



Main Phases of DB Design

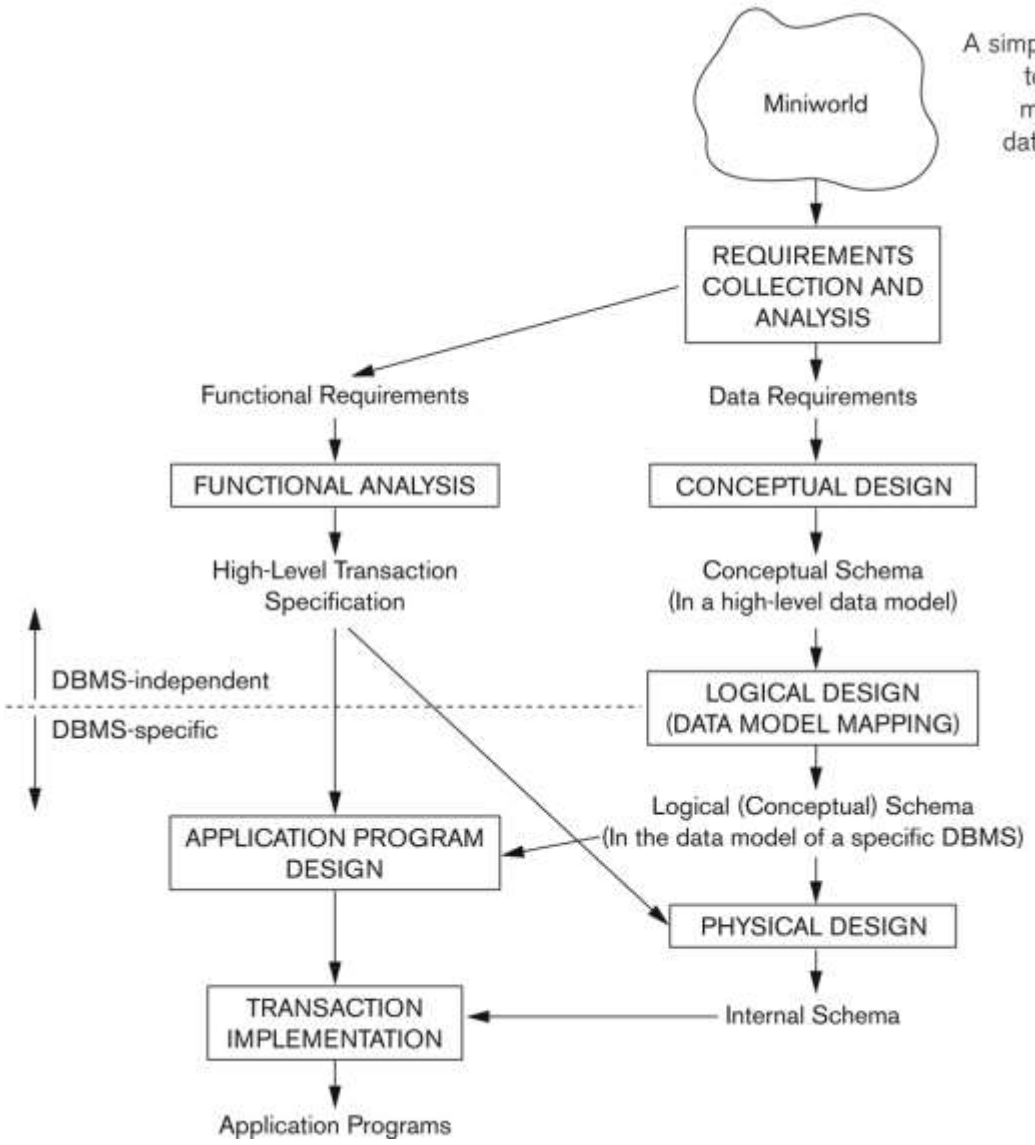


Figure 3.1
A simplified diagram to illustrate the main phases of database design.

Data Models

- **Data Model**

  A set of concepts to describe the *structure* of a database, and certain *constraints* that the database should obey.

- **Data Model Operations**

  Operations for specifying database retrievals and updates by referring to the concepts of the data model. Operations on the data model may include *basic operations* and *user-defined operations*.

Categories of data models

- **Conceptual (high-level, semantic)**

  Provide concepts that are close to the way many users *perceive* data. (Also called **entity-based** or **object-based** data models.)

- **Physical (low-level, internal)**

  Provide concepts that describe details of how data is stored in the computer and their access paths.

- **Implementation (representational)**

  Provide concepts that fall between the above two, balancing user views with some computer storage details.

Schemas versus Instances

- **Database Schema**

  The *description* of a database. Includes descriptions of the database *structure* and the *constraints* that should hold on the database.

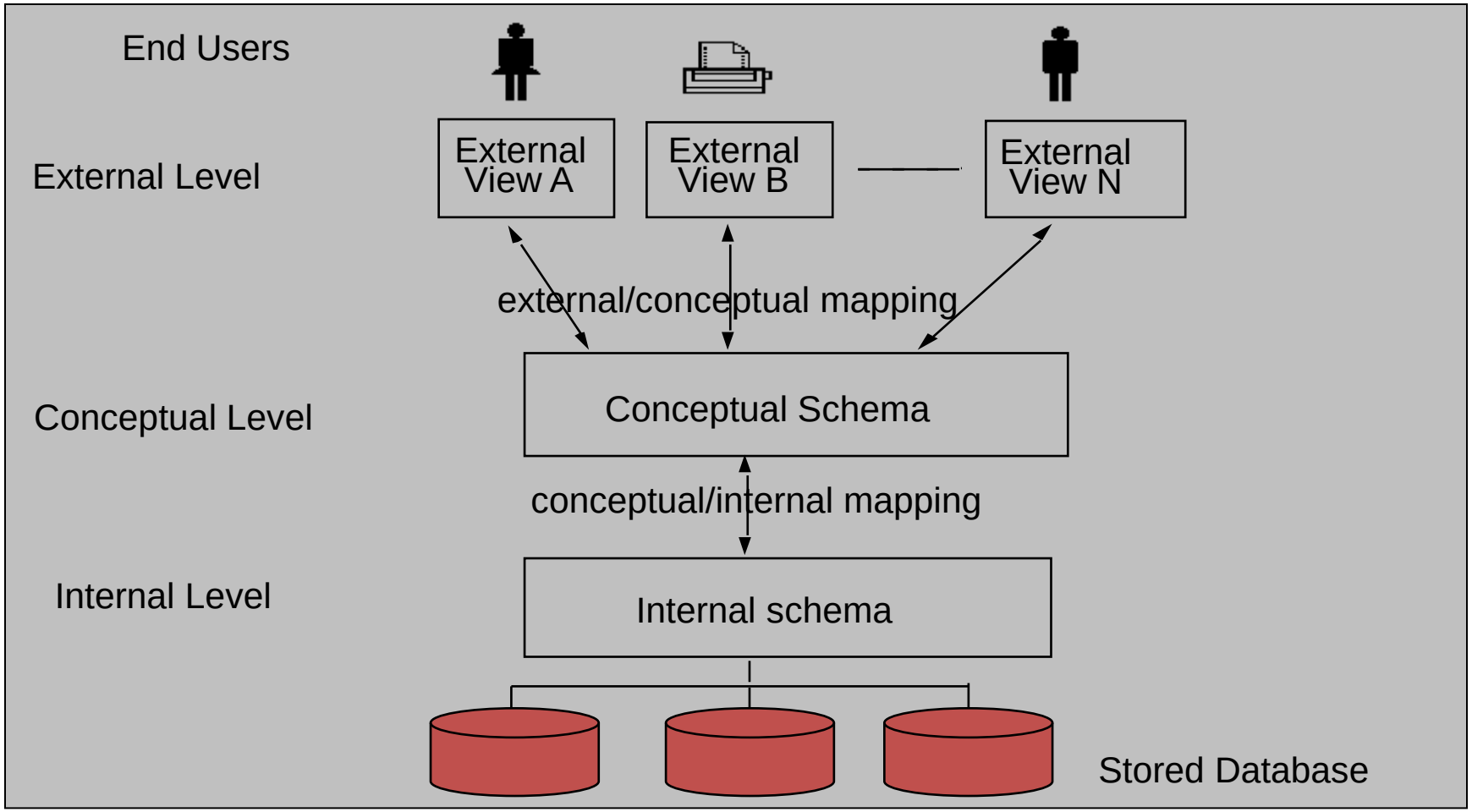
- **Database Instance**

  The *actual data* stored in a database at a *particular moment in time*. Also called **database state** (or **occurrence**).

`Student(studno,name,address)` ← Schema

`Student(123,Hassan,Maadi)` ← Instance

Three-Schema Architecture



Brief History of Databases

- **Early Database Applications**

  The *Hierarchical* and *Network* Models were introduced in mid 1960's and dominated during the seventies. A bulk of the worldwide database processing still occurs using these models.

- **Relational Model based Systems**

  The model that was originally introduced in 1970 was heavily researched and experimented with in IBM and the universities. Relational DBMS Products emerged in the 1980's.

Brief History of Databases (Cont.)

- **Object-Oriented and Object-Relational Applications**

  OODBMSs were introduced in late 1980's and early 1990's to cater to the need of complex data processing in CAD and other applications. Their use has not taken off much.

- **Data on the Web and E-commerce Applications:**

  Web contains data in HTML (Hypertext markup language) with links among pages. This has given rise to a new set of applications and E-commerce is using new standards like XML (eXtended Markup Language).

The Relational Data Model

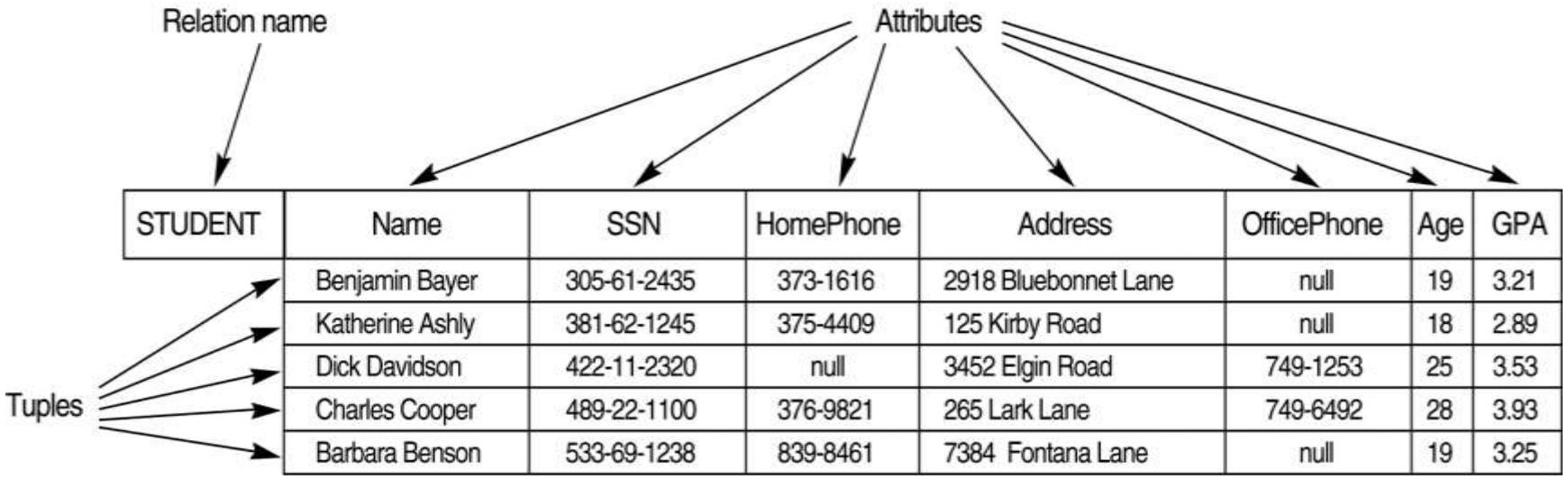
- The model behind most implementations of databases.
- Data is always represented as relations (2-dim. tables).
- SQL is a query language for developing and manipulating relational databases and is based on relational algebra.
- Some basic concepts:
  - Relation / Relationship
  - Attribute / Attribute Domain
  - Schema / Instance
  - Tuple
  - Key (Candidate, Primary, and Foreign)

How many of these do you recognize?

How many of these are you able to define now?

Relation

- A relation is a two-dimensional table:
  - Relation ≈ table.
  - Attribute ≈ column name.
  - Tuple ≈ row (not the header row).
- Database ≈ collection of relations.



STUDENT	Name	SSN	HomePhone	Address	OfficePhone	Age	GPA
	Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	null	19	3.21
	Katherine Ashly	381-62-1245	375-4409	125 Kirby Road	null	18	2.89
	Dick Davidson	422-11-2320	null	3452 Elgin Road	749-1253	25	3.53
	Charles Cooper	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
	Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	null	19	3.25

Relation Characteristics

- Each relation in the same relational database schema has a distinct name
- Each attribute in a relation has a distinct name.
- Values of an attribute are all from the same domain.
- Each tuple is distinct.
- Entity Degree is the number of fields/attributes in schema
- Entity Cardinality is the number of tuples in relation

Relational Model Integrity Constraints

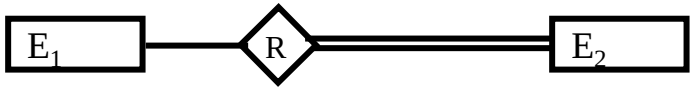
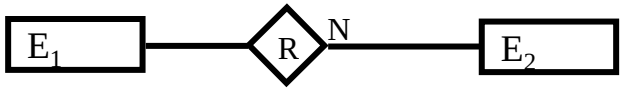
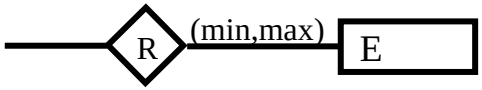
- Constraints are *conditions* that must hold on *all* valid relation instances. There are three main types of constraints:
  - **Key** constraints
  - **Entity integrity** constraints
  - **Referential integrity** constraints
  - **Semantic Integrity** constraints
- ICs are specified when schema is defined.
- ICs are checked when relations are modified.

Relational Data Model Operations

- There are two categories of relational data model operations:
  - Retrieval operations extract information from the relational database.
  - Update operations causes the relation (and the relational database) state changes. They Include:
    - Insert a tuple
    - Delete a tuple
    - Modify a value

Entity Relationship Diagram

Summary of ER-Diagram Notations

Symbol	Meaning
	ENTITY TYPE
	WEAK ENTITY TYPE
	RELATIONSHIP TYPE
	IDENTIFYING RELATIONSHIP TYPE
	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

ER-Diagram for the COMPANY database.

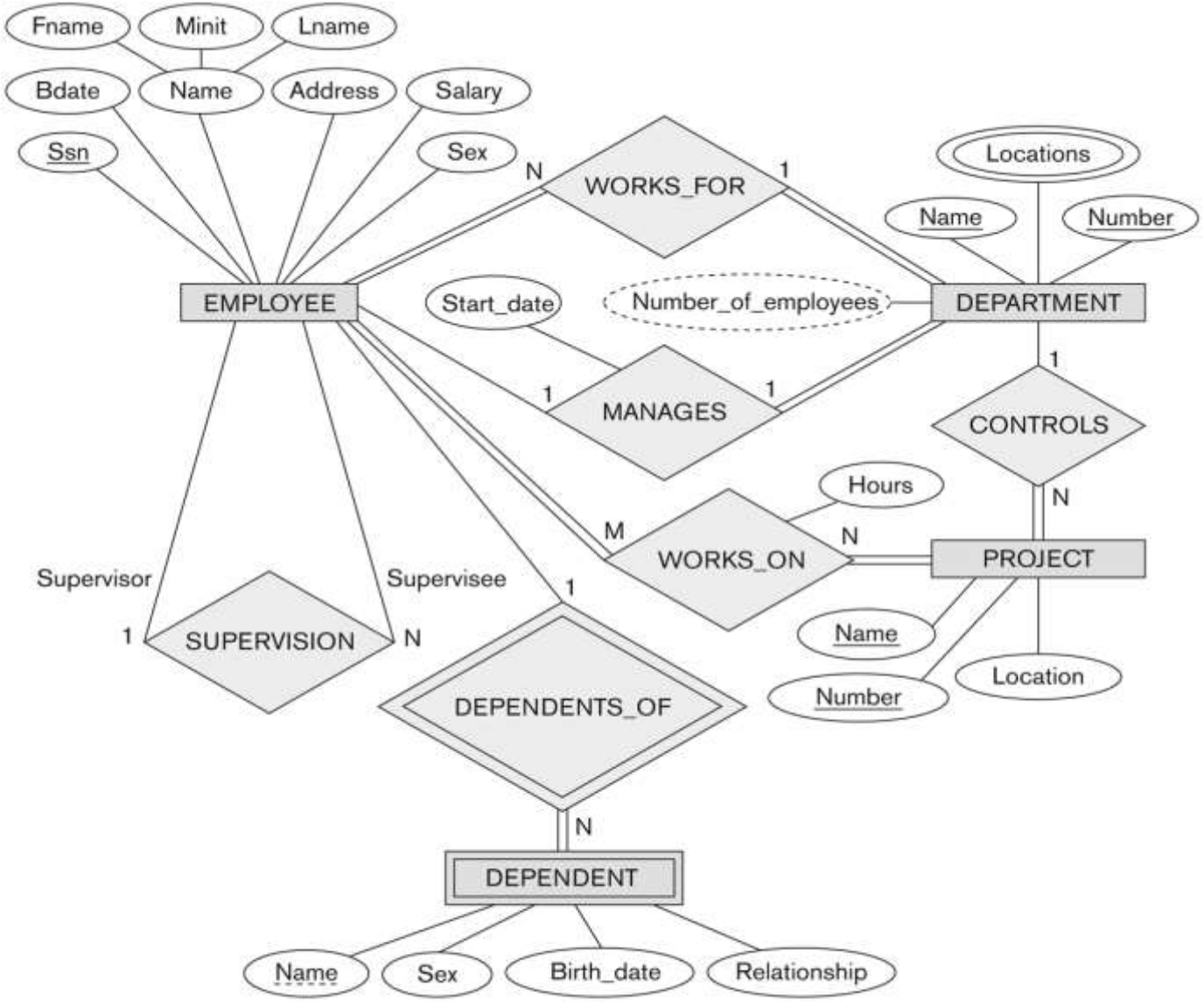
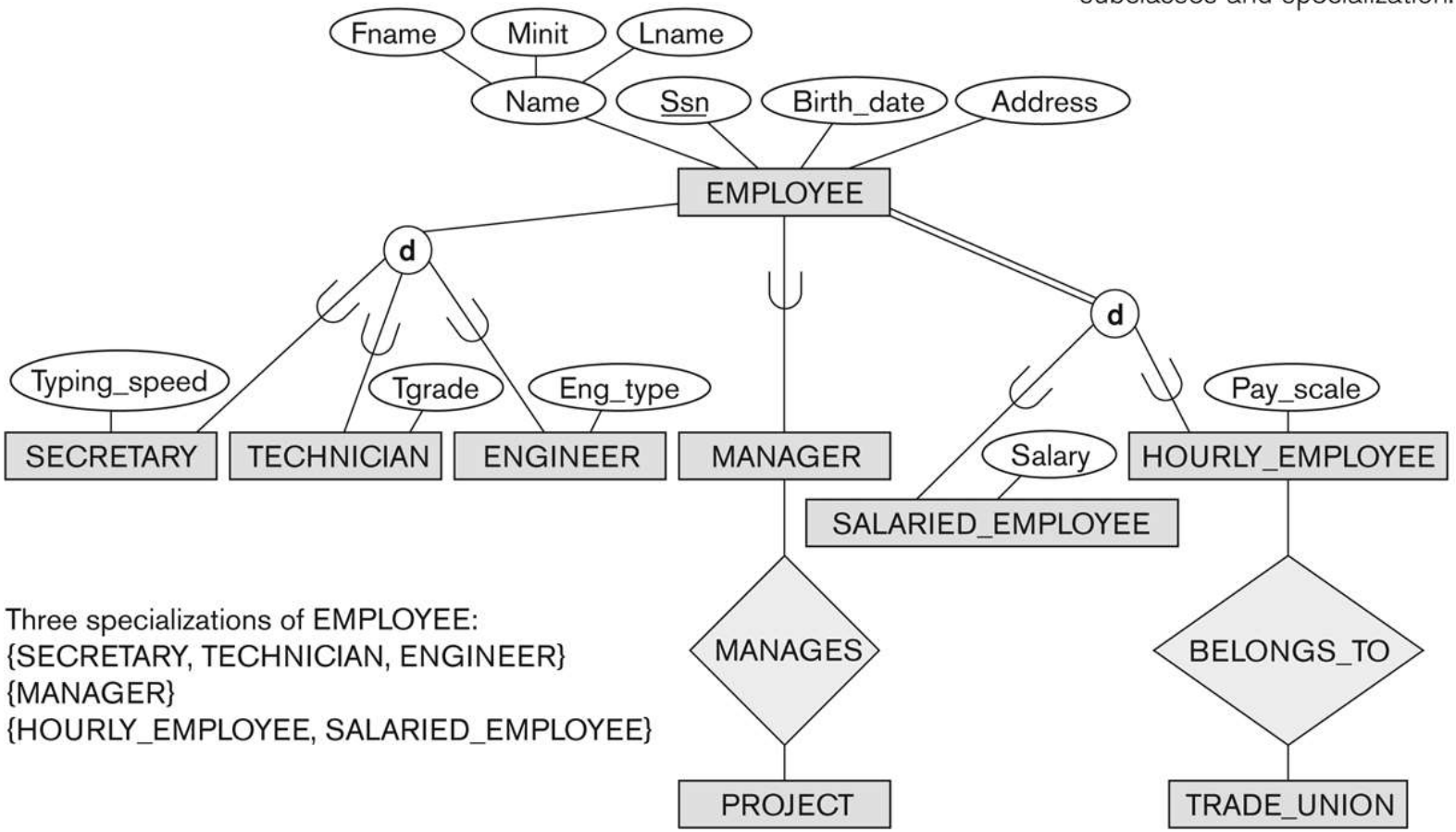


Figure 3.2
An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter.

EERD
Enhanced ER or Extended ER

- EER diagrams extend ER diagrams to represent additional subgroupings, called *subclasses* or *subtypes*
- These are called superclass/subclass relationships
- An entity that is member of a subclass *inherits*
  - All attributes of the entity as a member of the superclass
  - All relationships of the entity as a member of the superclass
- Specialization is the process of defining a set of subclasses of a superclass
- Generalization is the reverse of the specialization process

Figure 4.1
EER diagram notation to represent subclasses and specialization.



- SECRETARY (as well as TECHNICIAN and ENGINEER) inherit the attributes Name, SSN, …, from EMPLOYEE
- Every SECRETARY entity will have values for the inherited attributes

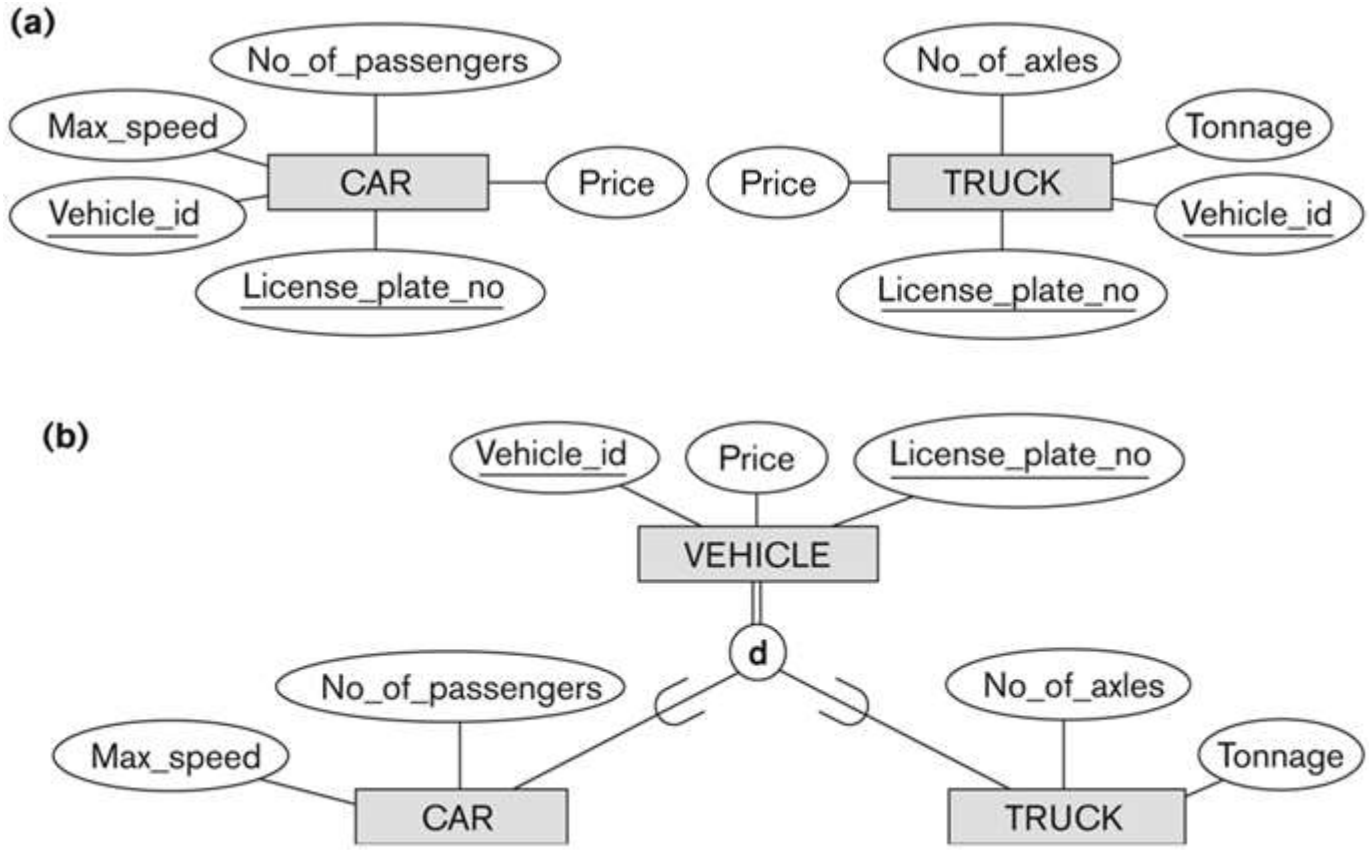


Figure 4.3
Generalization. (a) Two entity types, CAR and TRUCK.
(b) Generalizing CAR and TRUCK into the superclass VEHICLE.

Constraints on Specialization and Generalization

- Three basic constraints can apply to a specialization/generalization:
  - Condition-Defined Constraint
    - Attribute Defined Condition
    - User Defined Condition
  - Disjointness Constraint
    - Disjoint
    - Overlapping
  - Completeness Constraint
    - Total (mandatory)
    - Partial (optional)

Lattices & Shared Subclasses

- A subclass may itself have further subclasses specified on it
  - forms a hierarchy or a lattice
- ***Hierarchy*** has a constraint that every subclass has only one superclass (called ***single inheritance***); this is basically a ***tree structure***
- In a ***lattice***, a subclass can be subclass of more than one superclass (called ***multiple inheritance***)

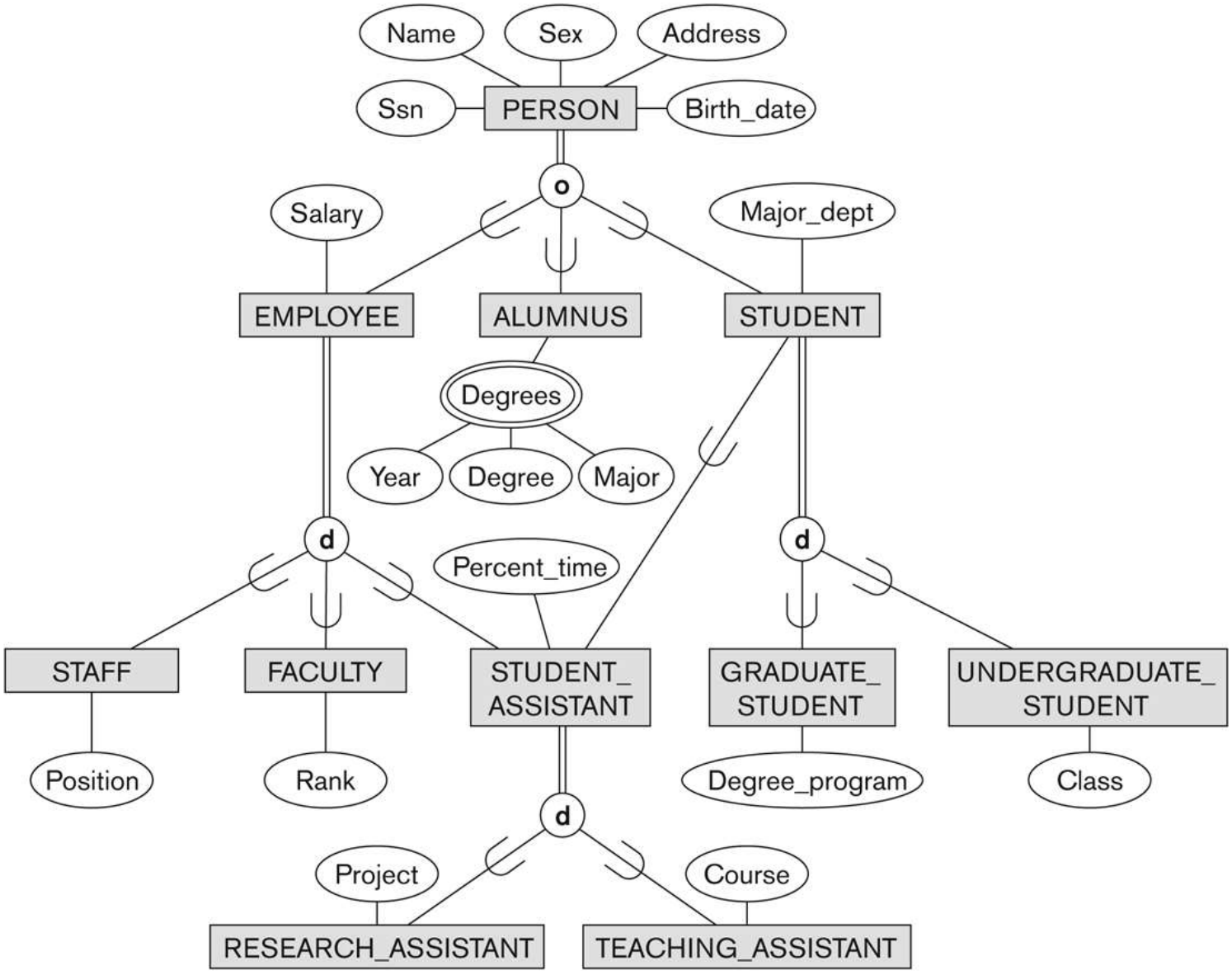


Figure 4.7
A specialization lattice with multiple inheritance for a UNIVERSITY database.

ER-to-Relational Mapping

Step 1: Mapping of Regular Entity Types

Step 2: Mapping of Weak Entity Types

Step 3: Mapping of Binary 1:1 Relation Types

Step 4: Mapping of Binary 1:N Relationship Types.

Step 5: Mapping of Binary M:N Relationship Types.

Step 6: Mapping of Multi-valued attributes.

Step 7: Mapping of N-ary Relationship Types.

Step 8: Mapping Super class/ Sub class relationship

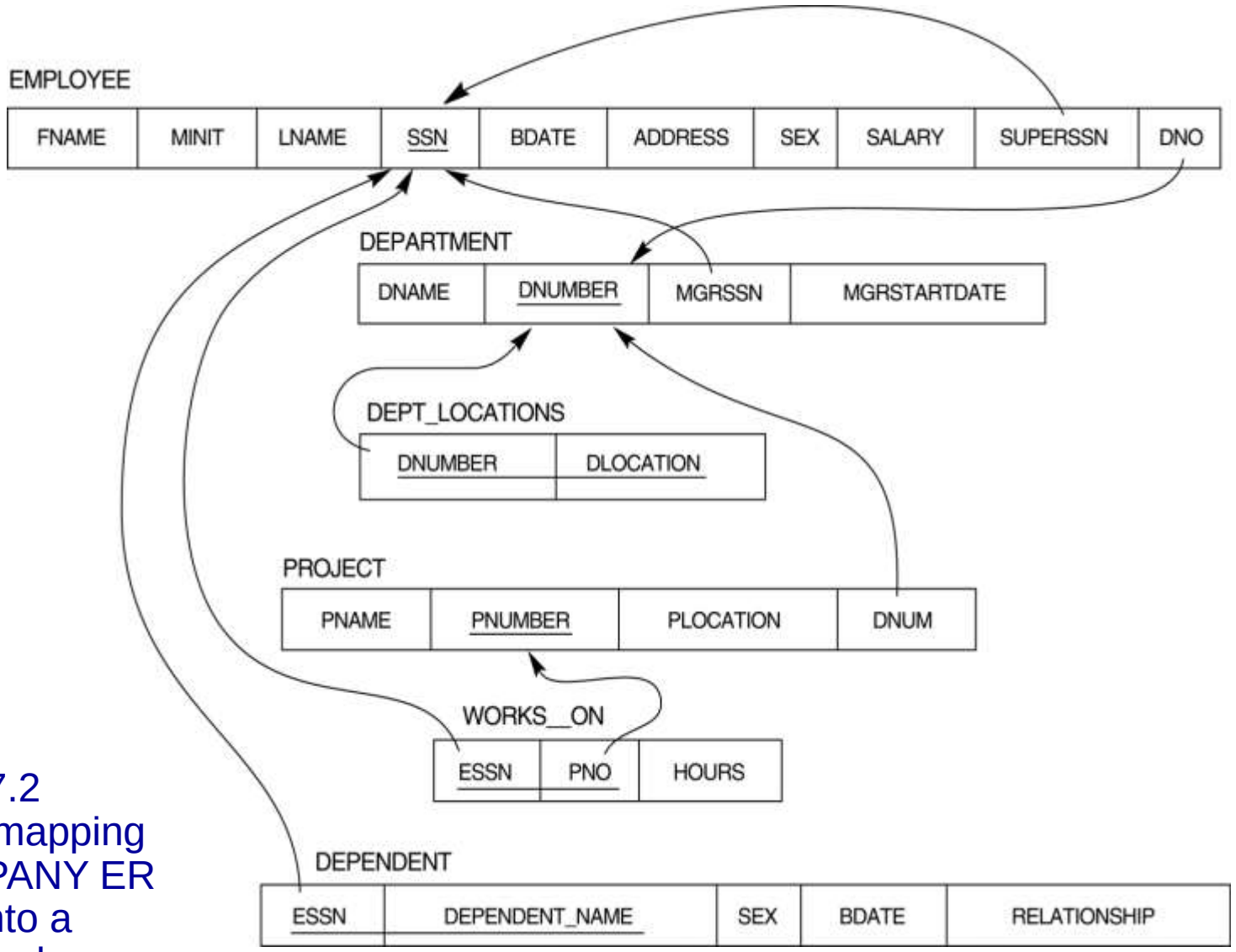
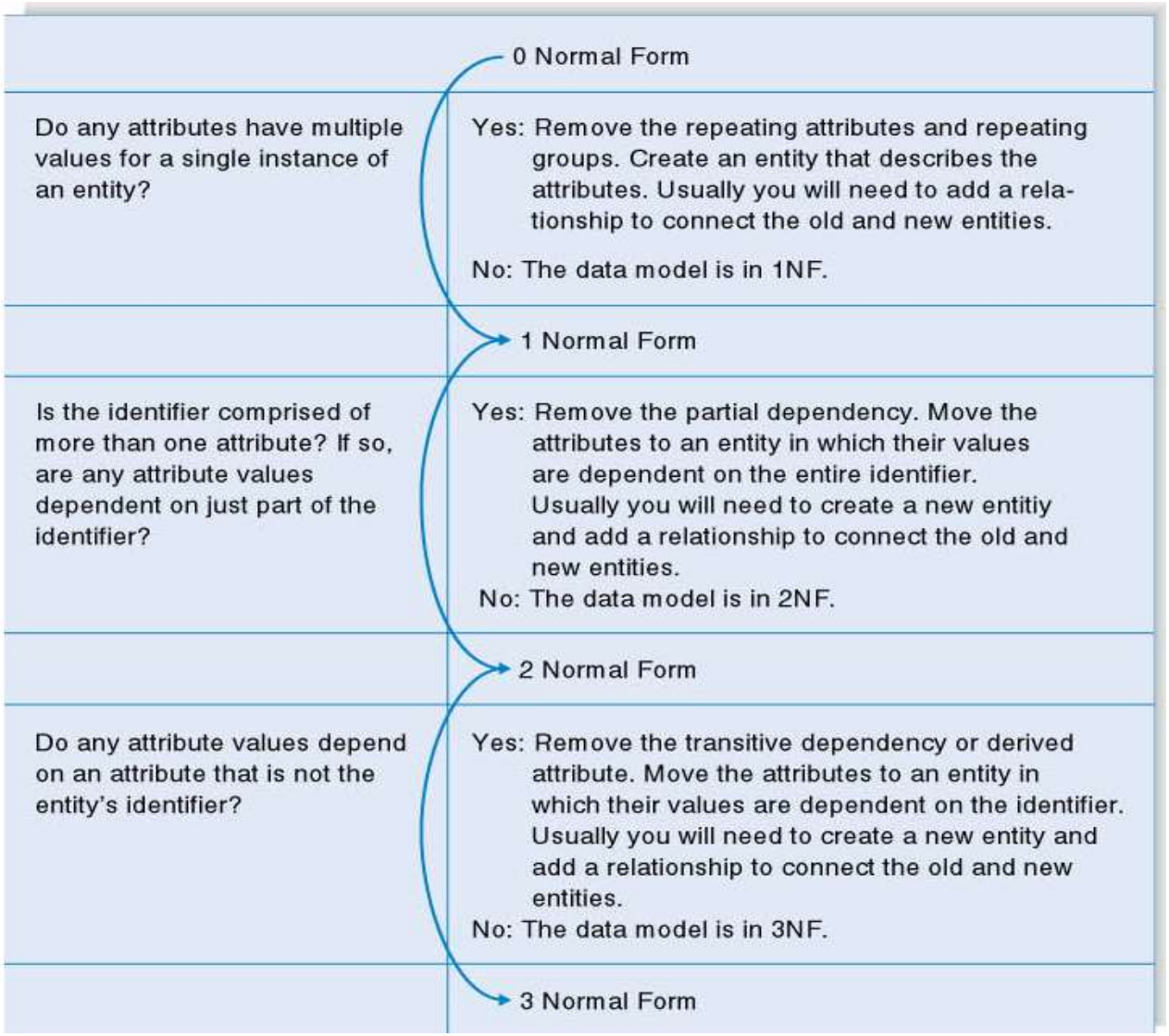


FIGURE 7.2
Result of mapping the COMPANY ER schema into a relational schema.

Normalization



Relational Algebra

- Operations that enable a user to specify basic retrieval requests.
- The result of a retrieval is a new relation, which may have been formed from one or more relations. The **algebra operations** thus produce new relations, which can be further manipulated using operations of the same algebra.

- Fundamental
  - union ∪
  - set difference -
  - selection σ
  - projection π
  - Cartesian product X
- Additional
  - rename ρ
  - intersection ∩
  - join ⋈
  - division ÷
- Type compatibility
  - same degree
  - corresponding attributes defined over the same domain

SQL

- SQL is a language that can be used for expressing queries on relations. It is based on a mixture of relational algebra for sets and bags.
- SQL also supports the creation and modification of relations.
- Some SQL examples:
  - CREATE TABLE *R* (<schema description>)
  - INSERT INTO *R* VALUES (v_1; ... ; v_n).
  - DELETE FROM *R* WHERE *C*.
  - UPDATE *R* SET *A* = *v* WHERE *C*.
  - SELECT .. FROM … WHERE…