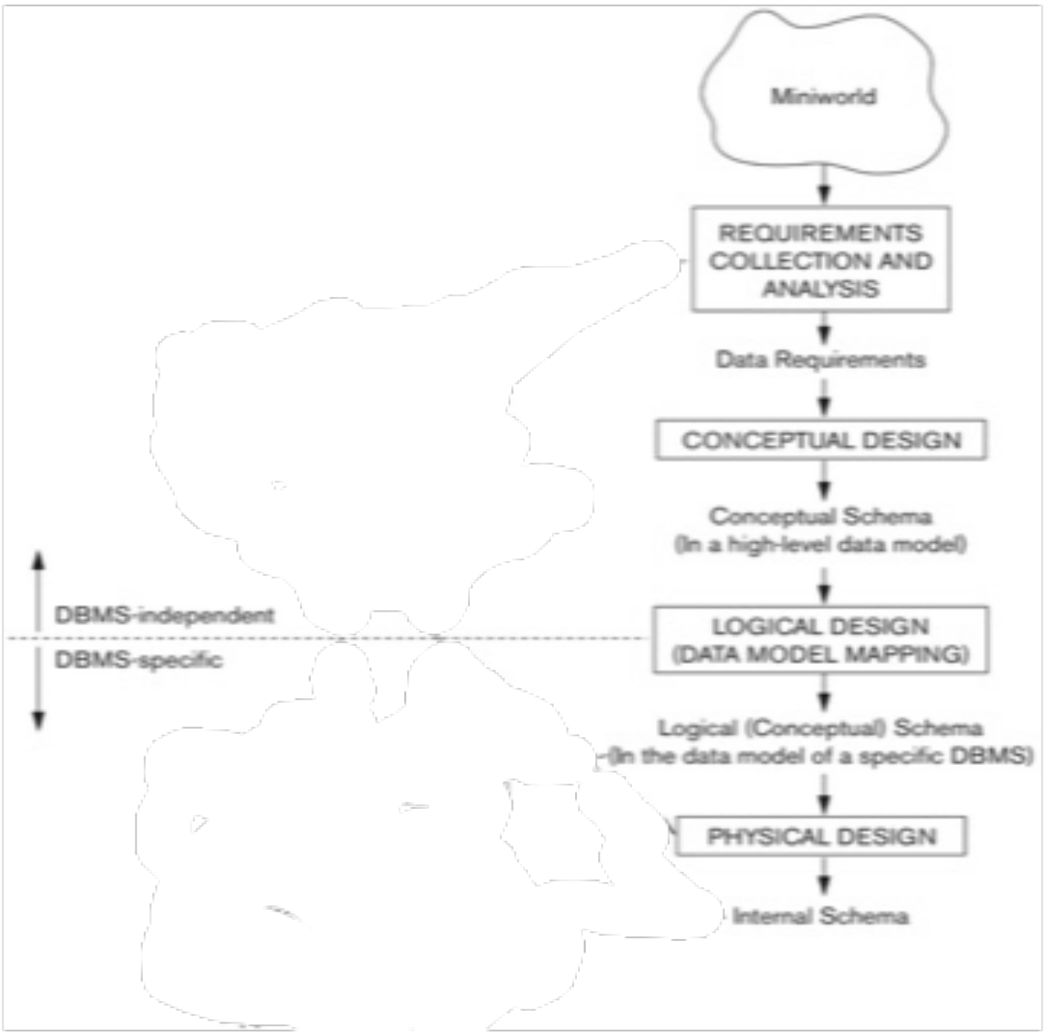
3. Data Modeling Using the Entity-Relationship (ER) Model

(Slide 4) Conceptual modeling is a very important phase in designing a successful database application. Generally, the term **database application** refers to a particular database and the associated programs that implement the database queries and updates. For example, a BANK database application that keeps track of customer accounts would include programs that implement database updates corresponding to customer deposits and withdrawals. These programs would provide user-friendly graphical user interfaces (GUIs) utilizing forms and menus for the end users of the application – the bank customers or bank tellers in this example. In addition, it is now common to provide interfaces to these programs to BANK customers via mobile devices using mobile app. A major part of the database application will require the design, implementation, and testing of these application programs.

In this chapter, we follow the traditional approach of concentrating on the database structures and constraints during conceptual database design. The design of application programs is typically covered in software engineering courses. We present the modeling concepts of the **entity-relationship (ER) model**, which is a popular high-level conceptual data model. This model and its variations are frequently used for the conceptual design of database applications, and many database design tools employ its concepts.

Object modeling methodologies such as the **Unified Modeling Language (UML)** are becoming increasingly popular in both database and software design. These methodologies go beyond database design to specify detailed design of software modules and their interactions using various types of diagrams. An important part of these metodologies – namely, *class diagrams* – is similar in many ways to the ER diagrams.

3.1 Using High-Level Conceptual Data Models for Database Design (Slide 5)



Miniworld
REQUIREMENTS COLLECTION AND ANALYSIS
Data Requirements
CONCEPTUAL DESIGN
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)
PHYSICAL DESIGN
Internal Schema

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

Figure 3.1 shows a simplified overview of the database design process. The fisrt step shown is requirements collection and analysis. During this step, the database designers interview prospective database users to understand and document their data requirements. These requirements should be specified in as detailed and complete a form as possible. In parallel with specifying the data requirements, it is useful to specify the known functional requirements of the application.

Once the requirements have been collected and analyzed, the next step is to create a conceptual schema for the database, using a high-level conceptual data model. This step is called conceptual design. The conceptual schema is a concise description of the data requirements of the users and includes detailed descriptions of the entity types, relationships, and constraints ; these are expressed using the concepts provided by the high-level data model. Because these concepts do not include implementation details, they are usually easier to understand and can be used to communicate with nontechnical users.

This approach enables database designers to concentrate on specifying the properties of the data, without being concerned with storage and implementation details, which makes it is easier to create a good conceptual database design.

The next step in database design is called logical design. The conceptual schema is transformed from the high-level data model into the implementation data model –such as the relational data model.

The last step is the physical design phase, during which the internal storage structures, file organizations, indexes, access paths, and physical design parameters for the database files.

3.2 A Sample Database Application

(Slides 8) In this section we describe a sample database application, called COMPANY, which serves to illustrate the basic ER model concepts and their use in schema design. We list the data requirements for the database here, and then create its conceptual schema step-by-step as we introduce the modeling concepts of the ER model. The COMPANY database keeps track of a company´s employees , departments and projects. Suppose that after the requirements collection and analysis phase, the database designers provide the following description of the *miniworld* –the part of the company that will be represented in database.

Figure 3.2 An ER schema diagram for the COMPANY database.

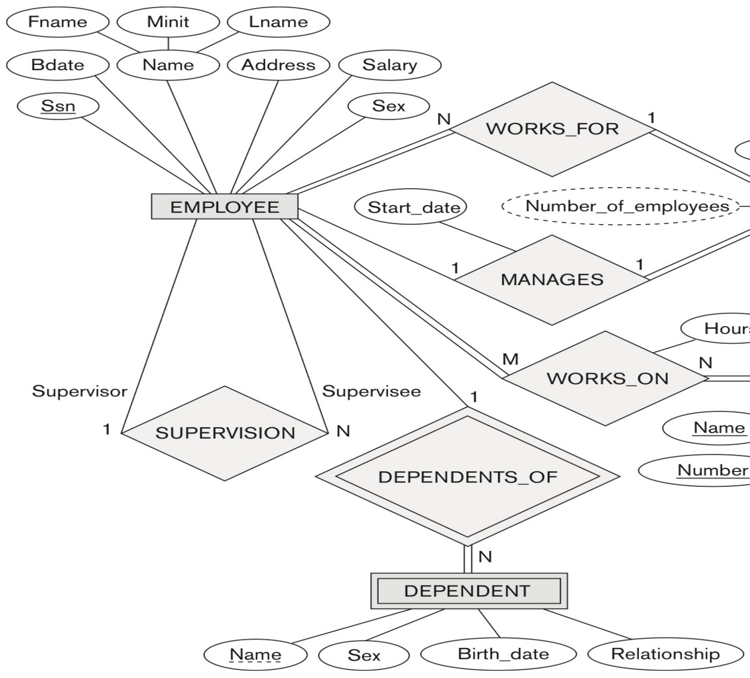


Figure 3.2 shows how the shema for this database application can be displayed by means of the graphical notation known as ER diagrams. This figure will be explained gradually as the ER model concepts are presented. We describe the step-by-step process of deriving this schema from the stated requirements.

(Slide 11) 3.3 E-R Model concepts

3.3.1 Entities and Attributes

Entities and their Attributes. The basic concept that the ER model represents is an entity, which 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) or it may be an object with a conceptual existence (for instance a company, a job, or a university course). Each entity has attributes – the particular properties that describe it. For example, an EMPLOYEE entity may be described by the employee´s name, age, address, salary, and job. A particular entity will have a value for each of its attributes. The attribute values that describe each entity become a major part of the data stored in the database.

Figure 3.3 Two entities, EMPLOYEE e_1, and COMPANY c_1, and their attributes

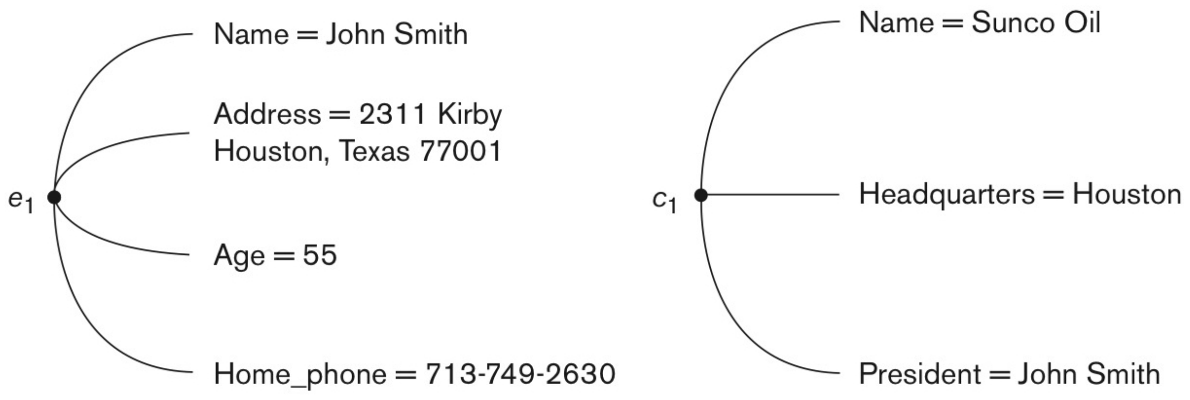


Figure 3.3 shows two entities and the values of their attributes. The EMPLOYEE entity **e1** has four attributes: Name, Address, Age and Home_phone; their values are ‘John Smith’, ‘2311 Kirby, Houston, Texas 77001’, ‘55’, and ‘713-749-2630’, respectively. The COMPANY entity c1 has three attributes: Name, Headquarters, and President; their values are ‘Sunco Oil’, ‘Houston’, and ‘John Smith’, respectively.