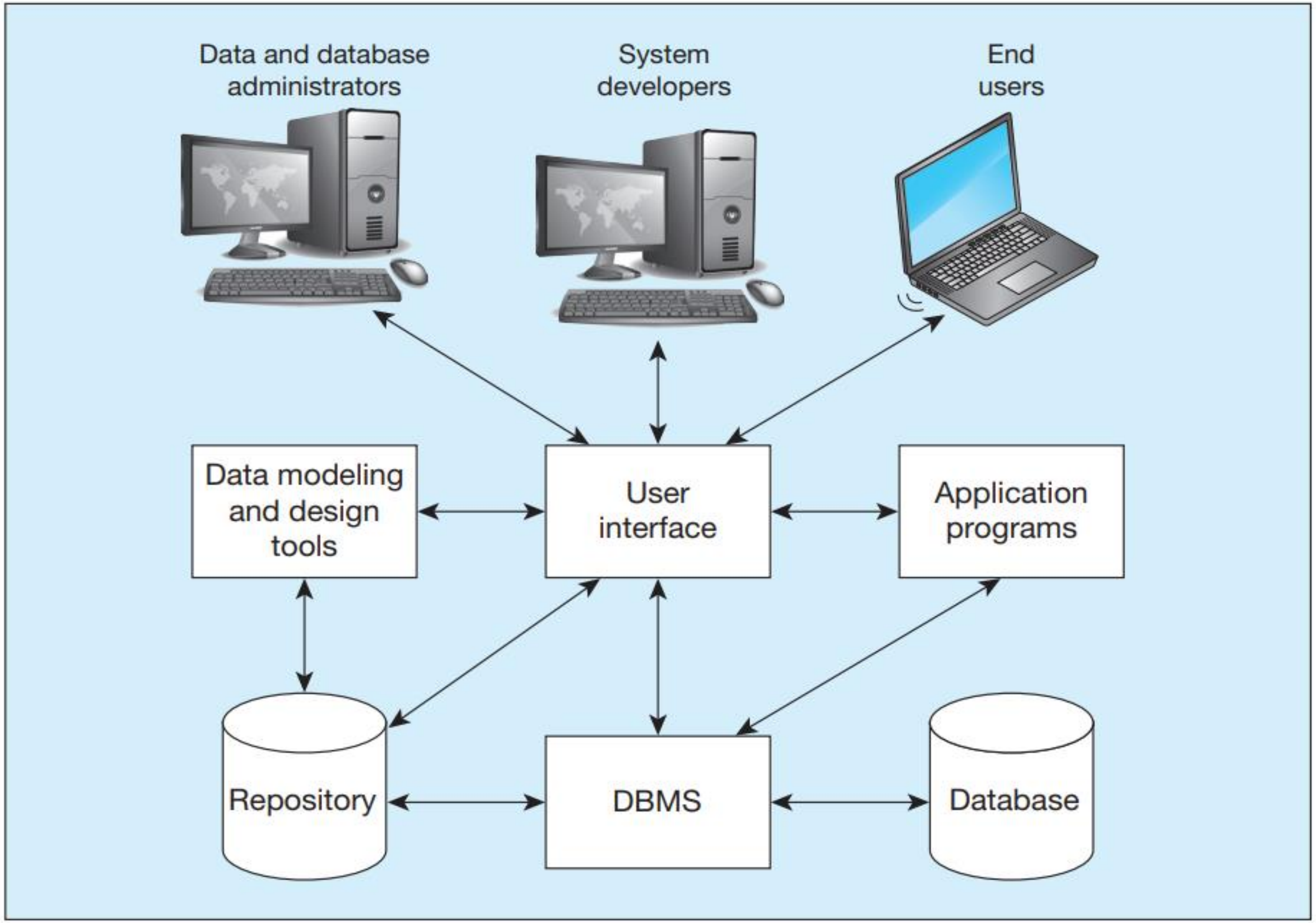
Database Systems CS-220

Saniya Ashraf

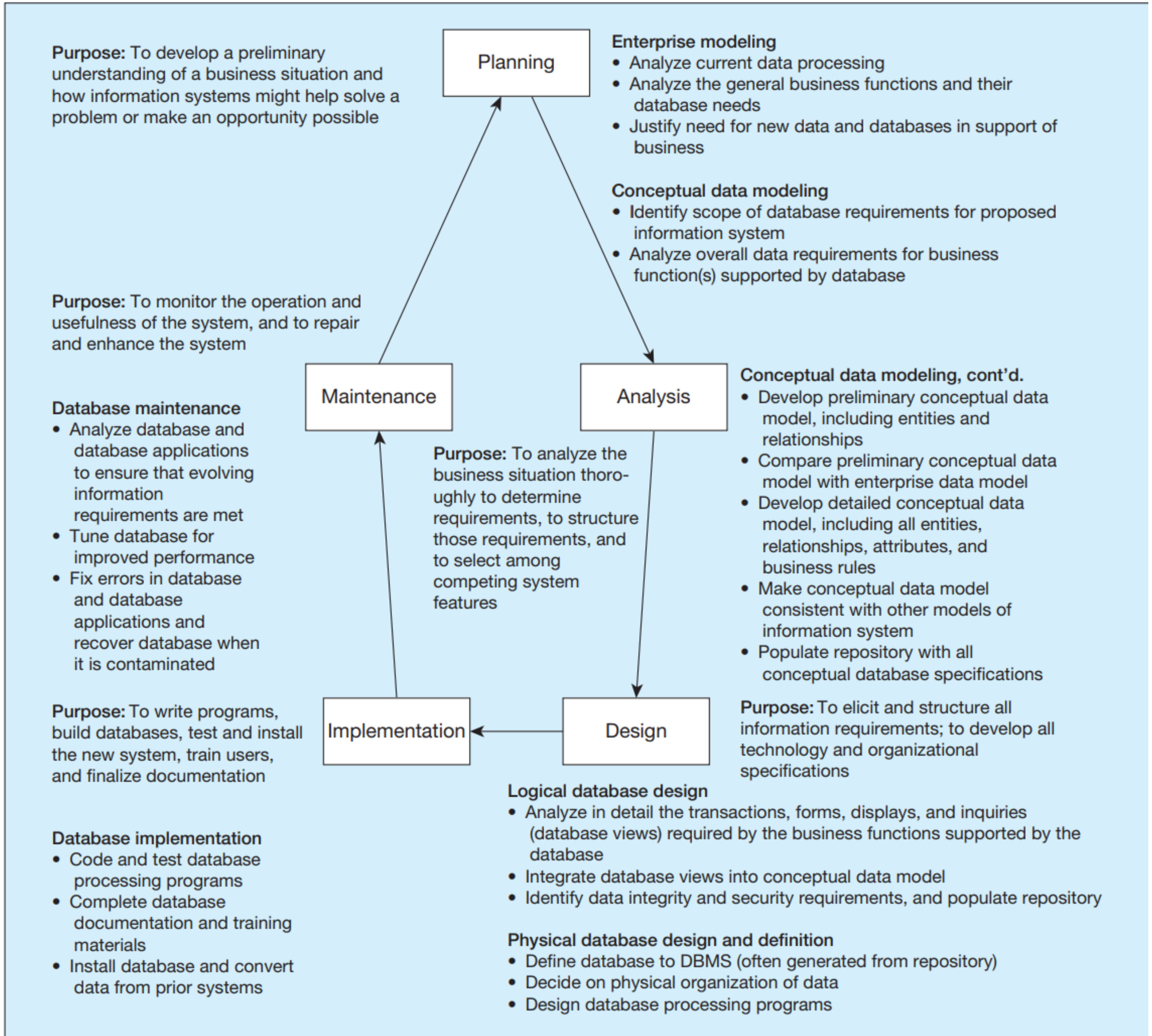
Lecture 4

COMPONENTS OF THE DATABASE ENVIRONMENT



Systems Development Life Cycle

- The Systems Development Life Cycle (SDLC) guides teams through building, maintaining, and replacing information systems.
- It's like a circle, showing how development is ongoing and can involve going back to earlier steps.
- SDLC has different phases, ranging from 3 to 20, with overlapping activities that can happen simultaneously.
- Each phase includes database development activities, like conceptual data modeling and planning data storage.
- The SDLC phases and database steps don't always perfectly align, but they work together to guide the development process.



Purpose: To develop a preliminary understanding of a business situation and how information systems might help solve a problem or make an opportunity possible

Enterprise modeling
- Analyze current data processing
- Analyze the general business functions and their database needs
- Justify need for new data and databases in support of business

Conceptual data modeling
- Identify scope of database requirements for proposed information system
- Analyze overall data requirements for business function(s) supported by database

Purpose: To analyze the business situation thoroughly to determine requirements, to structure those requirements, and to select among competing system features

Conceptual data modeling, cont'd.
- Develop preliminary conceptual data model, including entities and relationships
- Compare preliminary conceptual data model with enterprise data model
- Develop detailed conceptual data model, including all entities, relationships, attributes, and business rules
- Make conceptual data model consistent with other models of information system
- Populate repository with all conceptual database specifications

Purpose: To elicit and structure all information requirements; to develop all technology and organizational specifications

Logical database design
- Analyze in detail the transactions, forms, displays, and inquiries (database views) required by the business functions supported by the database
- Integrate database views into conceptual data model
- Identify data integrity and security requirements, and populate repository

Physical database design and definition
- Define database to DBMS (often generated from repository)
- Decide on physical organization of data
- Design database processing programs

Purpose: To write programs, build databases, test and install the new system, train users, and finalize documentation

Database implementation
- Code and test database processing programs
- Complete database documentation and training materials
- Install database and convert data from prior systems

Purpose: To monitor the operation and usefulness of the system, and to repair and enhance the system

Database maintenance
- Analyze database and database applications to ensure that evolving information requirements are met
- Tune database for improved performance
- Fix errors in database and database applications and recover database when it is contaminated

THE DATABASE DEVELOPMENT PROCESS

- So, when companies want to start making databases, they kick things off with something called **enterprise data modeling**. It's like creating a big picture of all the data the company needs, not just for one specific database, but for the whole organization.
- Think of it like **drawing a map of what data goes where and why**. This helps them figure out what databases they need and what they should contain.
- Sometimes, these projects are driven from the top down, where big goals like improving customer service or managing inventory better lead the way. Other times, they bubble up from the bottom, with users or IT folks saying, "Hey, we need a better way to handle this information!"
- If it's a bottom-up project, it might focus on building just one database to solve a specific problem. But usually, it's part of a bigger plan, where they're creating databases and the systems to use them all at once.

ENTERPRISE MODELING

- The first step in developing a database involves looking at the plans made during the information systems planning phase. This means checking out what data the organization already has and what it needs for upcoming projects.
- Analysts dive deep into current databases and systems, figuring out what kind of data they're dealing with and what data they'll need for new projects.
- Based on this analysis, they pick which projects to focus on next, considering how valuable each one will be for the organization.

What is a database schema?

- Designing the schema or structure of a database is the very first step in designing a database system.
- **how data is organized in a database.**

What is a database schema?

- The simplest way of understanding database schema is to think of it as the blueprint of a database.
- Before anyone can use a database to store and manipulate data, the database schema must first be designed.
- This process of database schema design is also known as **data modeling.**
- database schema is designed by **database designers.**
- database schema is just the skeleton of the database, and it doesn’t store any actual data.
- Once the designers have provided the database schema, the developers can understand how the data should be stored by the application that they are implementing.

What is a database schema?

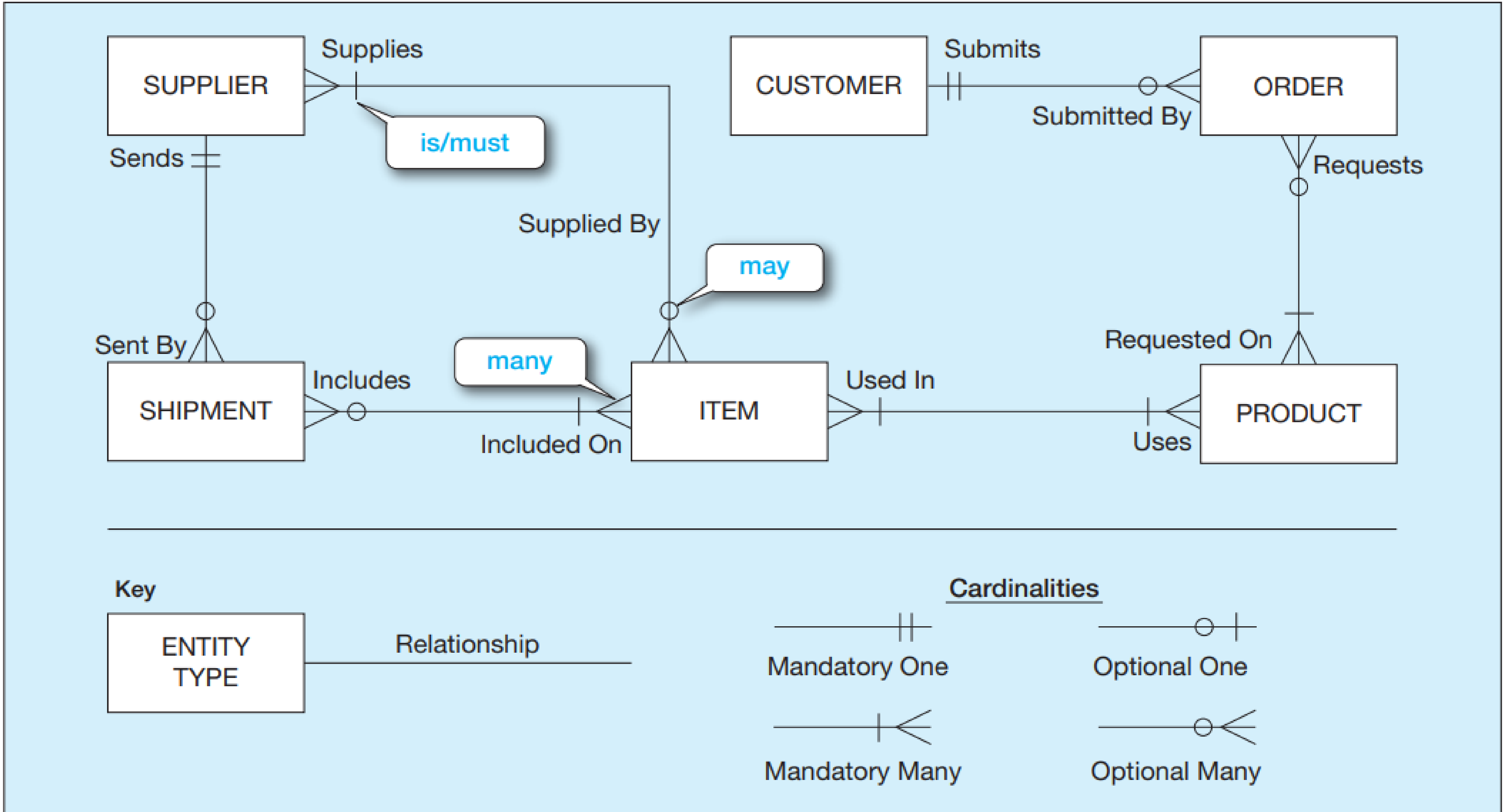
- Database schema can be broadly divided into three categories.

1. **Conceptual or logical** schema that defines entities, attributes and relationships.
2. Internal or physical schema that defines how data is stored in a secondary storage. In other words, the actual storage of data and access paths.
3. External or view schema that defines different user views.

Conceptual or logical schema

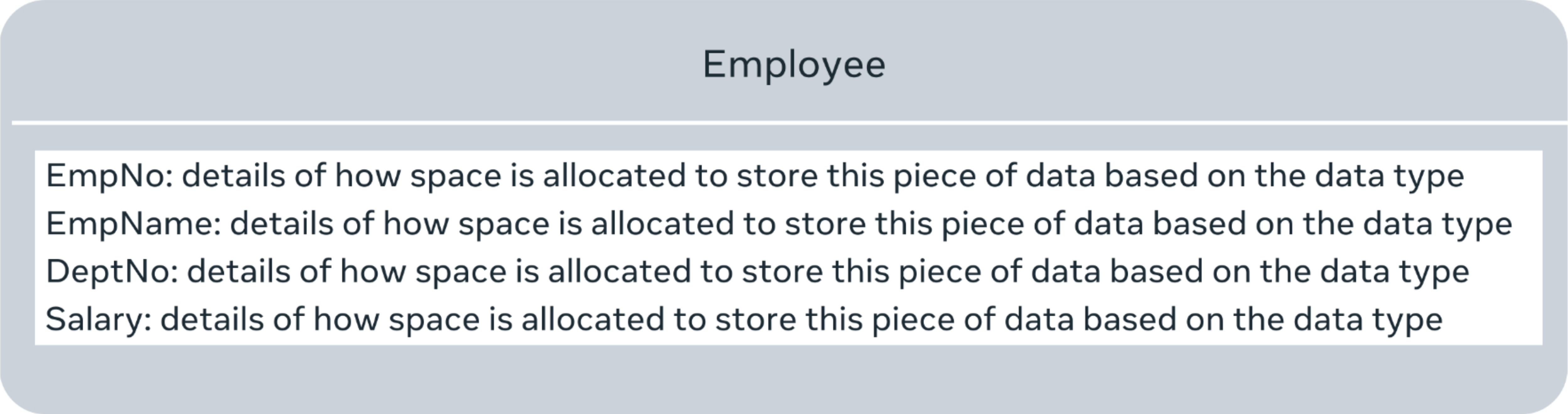
- The conceptual or logical schema describes the structure of the entire database for all the users.
- It describes the structure in terms of entities and features of the entities and the relationships between them.
- An Entity Relationship Diagram (ER-D) is usually drawn to represent the logical schema of a database.
- At this level, details about the physical storage and retrieval of data are hidden, and the database structure is described only at a concept level.
- The software developers work with the database at this level.

FIGURE 2-1 Sample E-R diagram



Internal or physical schema

- The internal or physical schema describes the physical storage of the database.
- It represents the entire database but at a very low level.
- This means it describes how the data is really stored on disk in the form of tables, columns and records.
- It defines what data is stored in the database and how.



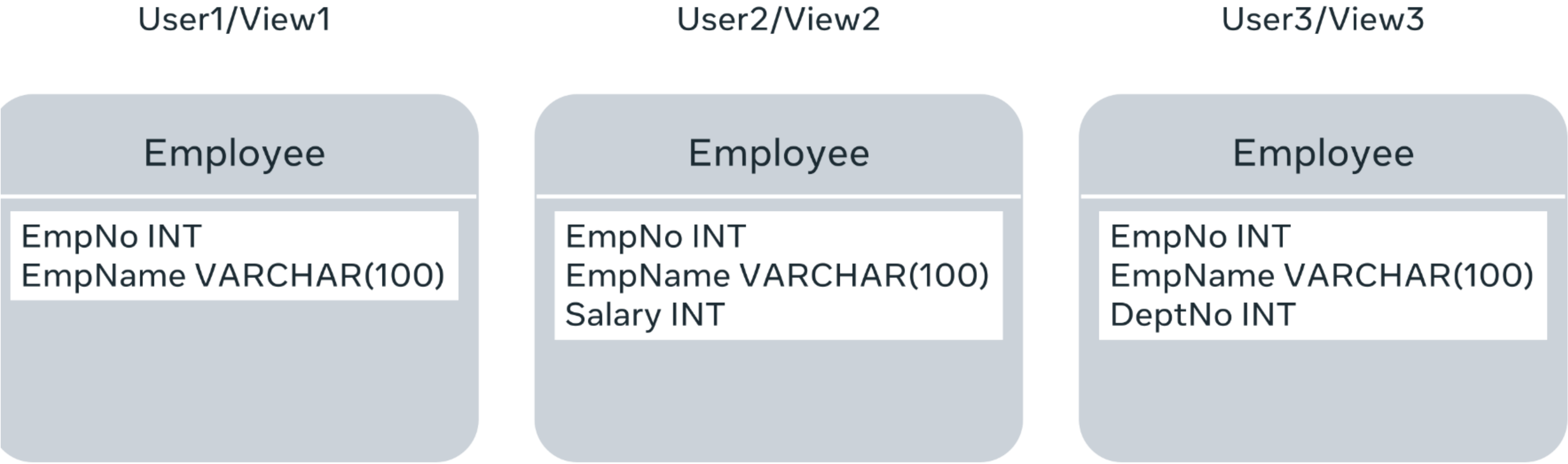
Employee

EmpNo: details of how space is allocated to store this piece of data based on the data type
EmpName: details of how space is allocated to store this piece of data based on the data type
DeptNo: details of how space is allocated to store this piece of data based on the data type
Salary: details of how space is allocated to store this piece of data based on the data type

This example depicts how the employee table should physically store its data. A real database would have more tables and the internal schema would describe the physical representation of all those tables in the entire database.

External or view schema

- The external or view schema describes the database like an external user would want to see it.
- This schema only describes the part of the database that the specific user is interested in.
- It hides the nonrelevant details of the database from a user.
- For example, a user from the sales department will see only sales-related data in a database.
- There can be many external schemas of a single database for different users.



User1/View1

Employee

EmpNo INT
EmpName VARCHAR(100)

User2/View2

Employee

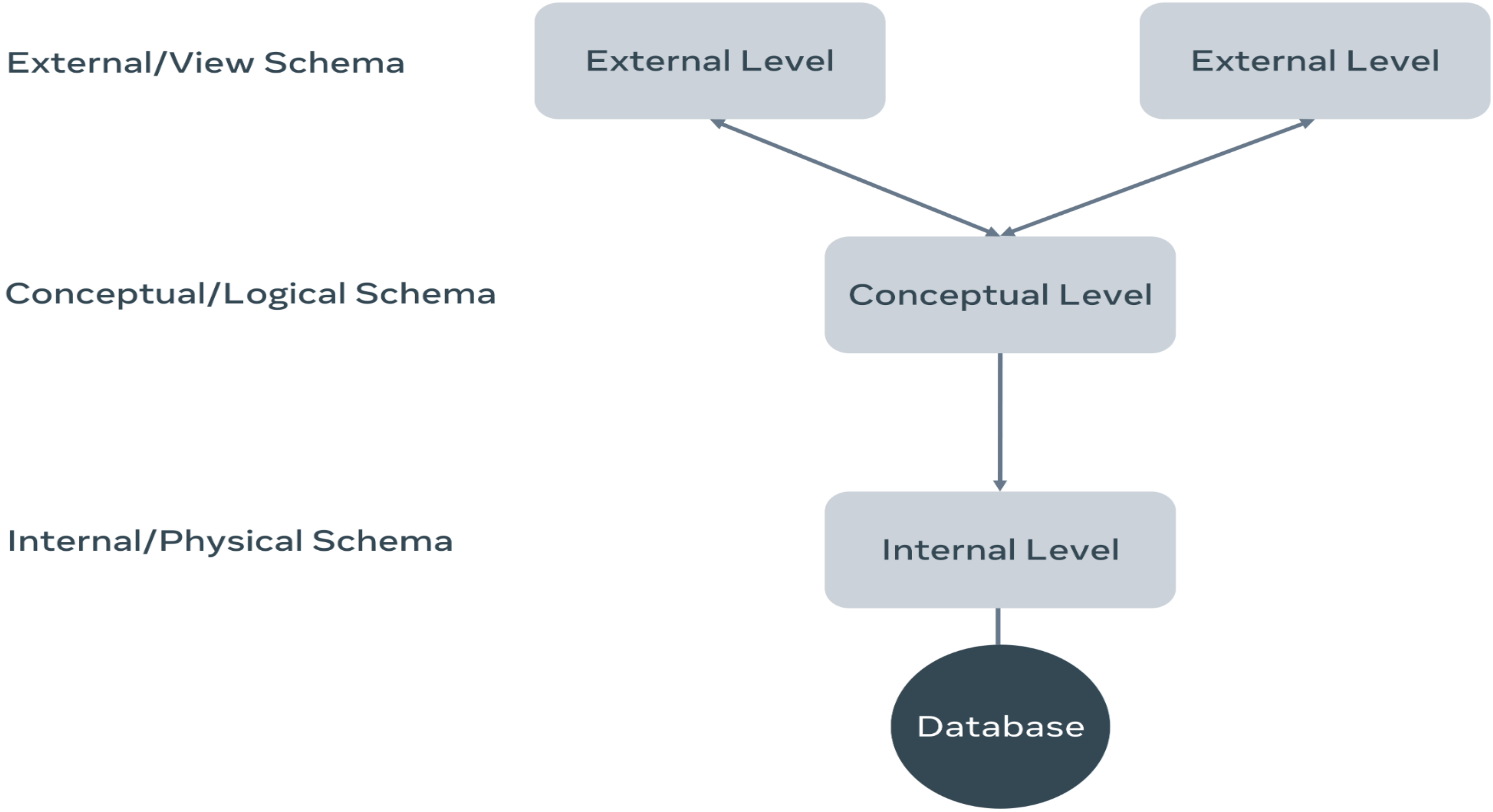
EmpNo INT
EmpName VARCHAR(100)
Salary INT

User3/View3

Employee

EmpNo INT
EmpName VARCHAR(100)
DeptNo INT

These three levels of schema are also known as the three-schema architecture. It can be diagrammatically depicted like this.



TASK

- Design Entity-Relationship Diagram (ERD) for a library system
- Steps you should consider while doing it
  - **User Requirements Gathering**
  - **Entity Identification**
  - **Attribute Specification**
  - **Relationship Determination**
  - **Cardinality(one to one, one to many etc) and Participation Constraints**