COMP2411

Tutorial 1 (Week 2)

Q1.

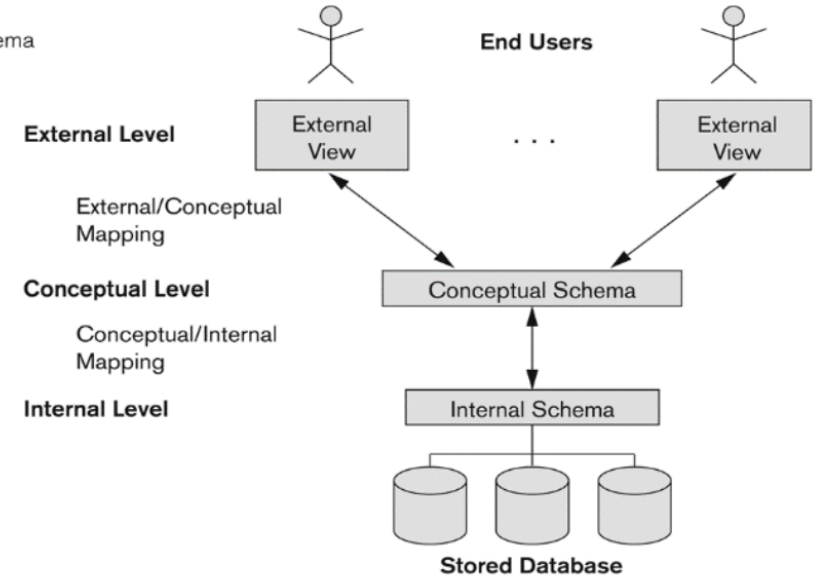
1. Consider a two-dimensional integer array of size $n*m$; the array is to be used in your favorite programming language. Illustrate the difference between
   a) the three levels of data abstraction, and
   b) a scheme and instances.

Q1.

Data Abstraction: 3-level architecture

- Data Abstraction
  - Abstract view of the data
    - simplify interaction with the system
    - hide details of how data is stored and manipulated
  - Levels of abstraction
    - physical/internal level: data structures; how data are actually stored
    - conceptual level: schema, what data are actually stored
    - view/external level: partial schema

Figure 2.2
The three-schema architecture.



Page 16, Lecture Notes 1.

Q1.

1. The **internal level** has an **internal schema**, which describes the physical storage structure of the database. The internal schema uses a physical data model and describes the complete details of data storage and access paths for the database.
2. The **conceptual level** has a **conceptual schema**, which describes the structure of the whole database for a community of users. The conceptual schema hides the details of physical storage structures and concentrates on describing entities, data types, relationships, user operations, and constraints. Usually, a representational data model is used to describe the conceptual schema when a database system is implemented. This *implementation conceptual schema* is often based on a *conceptual schema design* in a high-level data model.
3. The **external** or **view level** includes a number of **external schemas** or **user views**. Each external schema describes the part of the database that a particular user group is interested in and hides the rest of the database from that user group. As in the previous level, each external schema is typically implemented using a representational data model, possibly based on an external schema design in a high-level data model.

Elmasri, R. & Navathe, S.B. (2011). *Fundamentals of Database Systems: Chapter 2 Database System Concepts and Architecture* (6th ed.). Pearson.

Q1.

- **Physical level**. The lowest level of abstraction describes *how* the data are actually stored. The physical level describes complex low-level data structures in detail.
- **Logical level**. The next-higher level of abstraction describes *what* data are stored in the database, and what relationships exist among those data. The logical level thus describes the entire database in terms of a small number of relatively simple structures. Although implementation of the simple structures at the logical level may involve complex physical-level structures, the user of the logical level does not need to be aware of this complexity. This is referred to as **physical data independence**. Database administrators, who must decide what information to keep in the database, use the logical level of abstraction.
- **View level**. The highest level of abstraction describes only part of the entire database. Even though the logical level uses simpler structures, complexity remains because of the variety of information stored in a large database. Many users of the database system do not need all this information; instead, they need to access only a part of the database. The view level of abstraction exists to simplify their interaction with the system. The system may provide many views for the same database.

Silberschatz, A., Korth H.F. & Sudarshan, S. (2011). *Database System Concepts: Chapter1 Introduction* (6th ed.). McGraw-Hill.

Q1.

2.1.2 Schemas, Instances, and Database State

In any data model, it is important to distinguish between the *description* of the database and the *database itself*. The description of a database is called the **database schema**, which is specified during database design and is not expected to change frequently.[6] Most data models have certain conventions for displaying schemas as diagrams.[7] A displayed schema is called a **schema diagram**. Figure 2.1 shows a schema diagram for the database shown in Figure 1.2; the diagram displays the structure of each record type but not the actual instances of records. We call each object in the schema—such as STUDENT or COURSE—a **schema construct**.

Elmasri, R. & Navathe, S.B. (2011). *Fundamentals of Database Systems: Chapter 2 Database System Concepts and Architecture* (6th ed.). Pearson.

Q1.

1.3.2 Instances and Schemas

Databases change over time as information is inserted and deleted. The collection of information stored in the database at a particular moment is called an **instance** of the database. The overall design of the database is called the database **schema**. Schemas are changed infrequently, if at all.

Silberschatz, A., Korth H.F. & Sudarshan, S. (2011). *Database System Concepts: Chapter1 Introduction* (6th ed.). McGraw-Hill.

Q2. & Q3.

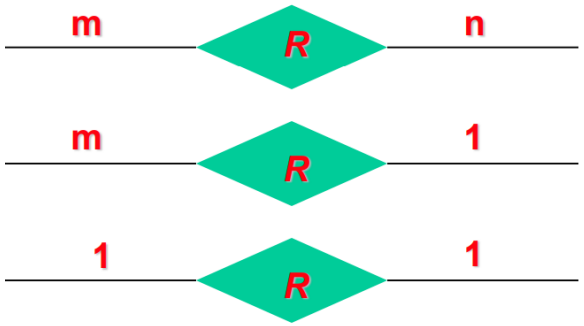
2. Construct an ER diagram for a car insurance company with a set of customers, each of whom owns a number of cars. Each car has a number of recorded accidents associated with it.

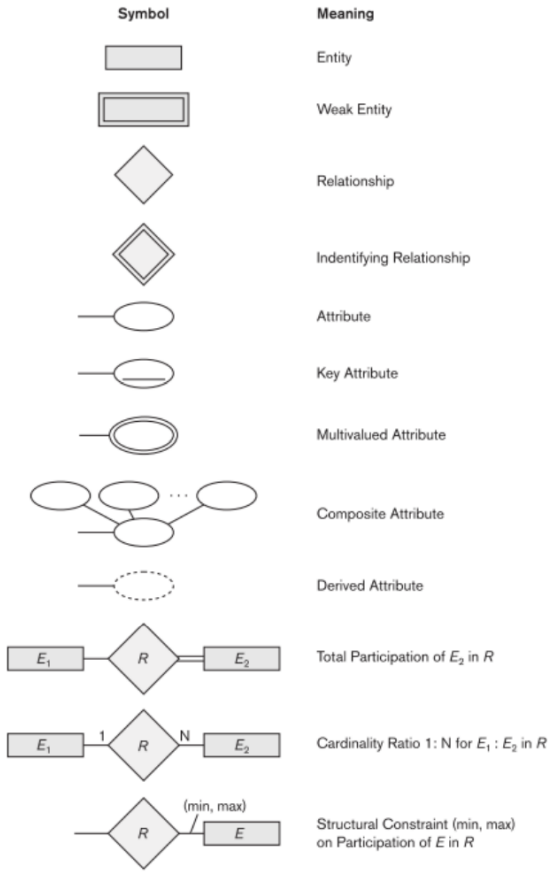
3. Construct an ER diagram for a hospital with a set of patients and a set of medical doctors. A log of the various conducted tests is associated with each patient.

Q2. & Q3.

The Entity-Relationship Model

- ER Diagram
  - Rectangles: Entity Sets
  - Ellipses: Attributes
  - Diamonds: Relationship Sets
  - Lines: Attributes to Entity/Relationship Sets
    or, Entity Sets to Relationship Sets





Page 43, Lecture Notes 1.

References

- Silberschatz, A., Korth H.F. & Sudarshan, S. (2011). *Database System Concepts: Chapter1 Introduction* (6th ed.). McGraw-Hill
- Elmasri, R. & Navathe, S.B. (2011). *Fundamentals of Database Systems: Chapter 2 Database System Concepts and Architecture* (6th ed.). Pearson.