COMPUTER ORGANIZATION AND ARCHITECTURE

Lecture 2

The Database Environment

Dr Victoria Mukami

INTRODUCTION

This lecture is an introduction to the database environment. We first do a review of the three-level database architecture. This will then pave way for a review of the external, conceptual, and internal levels of the environment. Next, we will review the evolution of various data models. Specifically, we will review the hierarchical and network models, the relational model, the entity-relationship model, the object-oriented model, and emerging data models.

Learning objectives

By the end of this topic, you should be able to:

1. Identify the purpose of the three-level database architecture
2. Distinguish between the external, conceptual, and internal levels
3. Understand the evolution process of data models
4. Understand the basic data modelling building blocks

OVERVIEW

In the last lecture, we did an introduction to databases including the evolution of databases. This lecture focuses mainly on the database environment itself. When the database evolved from the file-based system, an architecture was proposed and is still in use to date with some variations. This was important as it lay the foundation for any database that would be designed.

THREE-LEVEL DATABASE ARCHITECTURE

The three-level architecture was first proposed in 1971 as a two-level approach with a schema and a subschema [2]. This then led to a review by the American National Standards Institute (ANSI) Standards Planning and Requirements Committee (SPARC) to produce a similar architecture in 1972 [2]. The ANSI-SPARC architecture saw the need to have a three-level architecture as opposed to the predecessor two-level. These three levels are distinct and can be described as three levels of abstraction. The levels consist of an external, internal and conceptual level as shown in Figure 1. The three-level architecture aimed to separate the user from the physical database [3]. It aids a user to understand the structure of a database.

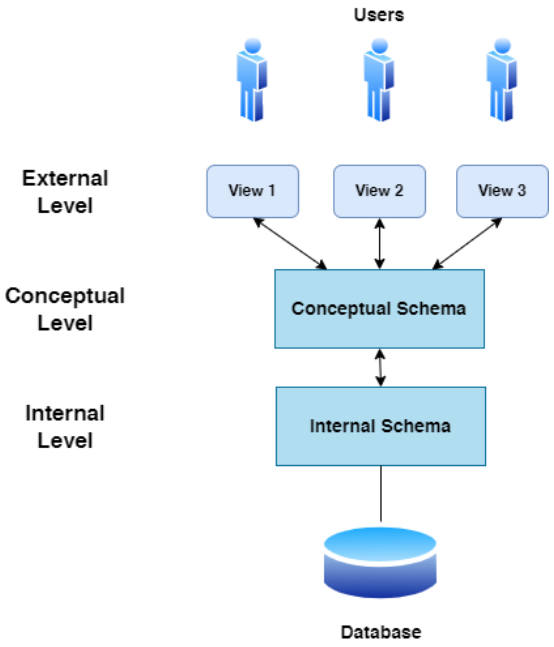

Figure 1: Adapted ANSI-SPARC three-level architecture [2]

While the three-level is the standard architecture, some databases may not explicitly separate the levels, while others may not include all the three levels. The three levels are in use in today's databases as it separates the various views of a database. We discuss the three levels individually in the section below.

The External Level

This is the users' view of the database. How the user perceives what the database looks like. The user refers to any person who uses the database either by adding or retrieving information. Each user will be interested in a specific part of the database and may not need to access all the components of the database. This is called the user view of the database or the external schema. For instance, let us take a finance office within a university. One employee within the finance department may be interested in pulling up student tuition payment details. A different employee may be interested in extracting payroll details involving employees. Each of these employees will only see data that involves their subject i.e., the student details or the employee details.

The Conceptual Level

This is the community view and describes all the data stored within the databases and all the relationships [2]. This is the level found between the external and internal levels.

The level not only describes the data but the structure of the entire database. This level hides any details of the physical storage of the database [3]. This is the database itself as seen in Figure 1. The conceptual level represents the following [2]

- Entities, attributes, and the relationships
- Constraints of the data
- Semantic information about the data
- Any security or data integrity information

The Internal Level

This is the physical representation of the database and describes how data is stored in the database [2]. This is the view that the database management system sees. This view works with the operating system concepts such as file management and storage techniques to retrieve, store and build up data. The internal level focuses on [2]

- Storage space allocation
- Record placement
- Data compression
- Data encryption

DATA MODELS

A data model is a simple representation of a more complex real-world data structure [1]. It helps users understand the complexities of the real-world environment. The data model will represent various data structures, their characteristics, relationships, constraints and transformations [1]. Data models are important as they help to improve an understanding of the business for which the database is developed. Data modelling is therefore considered as the first step in the design of a database. One of the reasons why database modelling is important is due to the various views, especially by the various database users. How an end-user views the database may not be the same as a database administrator views it. That said, all the views are necessary to give the entire picture or design of the database.

A data model has several building blocks which include:

- **Entity** - this is a person, place, or thing that the data is recorded about. Entities are considered to represent a real-world objects. An example of an entity would be a student or an employee
- **Attribute** - this is a characteristic of the entity. A student entity would have the following attributes associated with them, student registration number, student name, phone number, gender, country and many others.
- **Relationship** - This is a connection between entities. A relationship could exist between a student and a faculty member, where the faculty member could be an academic advisor of the student. Several relationships exist and these will be discussed in Lecture 4.
- **Constraint** - This is a restriction placed on data [1]. Constraints ensure that data integrity is upheld. For instance, a student can only be advised by one faculty member. Students can only take 1 - 7 courses per semester.

Evolution of Data Models

Throughout the evolution of databases, several data models have existed all with aim of trying to curb the limitations of previous data models.

1. **Hierarchical and Network Models [1]**

   The hierarchical model was developed in the 60s to deal with large amounts of data. The model was represented in a hierarchical scheme, similar to a pyramid. This meant that levels appearing at the top of the pyramid were considered as parents of components directly beneath them and shared a relationship called a parent-child relationship.

   The network model developed more complex relationships since the hierarchical model could only represent a one-to-many relationship. A parent can have many children, but a child can only have one parent. The network model dealt with the relationship limitations of the hierarchical model a many to many relationships could exist.

2. **Relational Model**

   This was introduced in the '70s and was a breakthrough for both users and designers [1]. Relations were introduced during this stage. A relation should not be confused by a relationship. A relation here is like a table that consists of rows and

columns. The columns represent fields while the rows entail various records. The relational model was implemented by a relational database management system. This is the stage where data complexities were hidden from users. This was the era where the Structured Query Language (SQL) was introduced. We review more on SQL in later lectures.

3. Entity-Relationship Model

With the success of the RDMS, there were more complex data and relationships that necessitated the need for more complex relational models. The entity-relationship model became the model of choice. It was introduced in 1976 when entities and their associated relationships were used. Entity Relationship diagrams were introduced during this ERA. We review more on Entity Relationship Diagrams in Lecture 4.

4. Object-Oriented Model

In the object-oriented data model (OODM) all the data and any associated relationships are contained within a single structure known as an object [1]. An object allows for various operations such as manipulation, querying, and storage of data within it. An **object** is the same as an entity and could be defined as the abstraction of a real-world entity [1]. OODM additionally uses **attributes** that describe the object. Similar objects are grouped and placed in a **class** that can be arranged hierarchically. An object can inherit the attributes and behaviours of the class it belongs to. This is known as **inheritance**. While entities and associated relationships are shown in an entity-relationship diagram, the OODM uses a **Unified Modelling Language** (UML).

5. Emerging Data Models

Throughout the development of various data models, one chief area has been the improvement of the previous data models either by simplifying complex relationships or improving the clarity of the data models. One area of concern has been the amount of data that organizations have collected over the years and the need to make the data usable [1]. Lots of the data is not found in a structured manner and may include web data that contains a combination of structured and unstructured data. **Unstructured** data is data that exists in its raw state, while

structured data is the result of formatting unstructured data. This has led to a term called Big Data which refers to finding better ways to manage large amounts of web and sensor-generated data [1]. Big Data has led to the development of various databases that can be used to manipulate the data. These emerging technologies include Hadoop and NoSQL. There are many other emerging technologies some still in development that is aimed toward harnessing the power of Big Data.

DEGREES OF ABSTRACTION

When a database is being designed and developed, there are specific ways in which these activities happen. As shown by the ANSI/SPARC there are various levels to a database. These levels reflect the various data models. The design of a database should start from the highest level, that is, the external level and complexities will keep increasing until the internal level is achieved.

The External Data Model

This is like the external view discussed previously. It is focused on the end-users and presents their view of the organization [2]. There are several advantages associated with the sternal level/data model [1].

1. It becomes easy to view any data that is associated with the business
2. The designer can provide any feedback that shows the model's accuracy [1].
3. Security is enhanced especially due to the use of constraints and rules.
4. Development especially of the application programs to access the database is made easy.

The Conceptual Model

This represents the community view of the DBMS [2]. There are important advantages that the conceptual level provides

1. It provides an easy to understand and overall view of the data environment [1].
2. The conceptual level is dependent on both the application software designed for access to the external level and the hardware needed [1]. It neither depends on the database management software nor the hardware used to implement the model.

The Internal Data Model

This maps the conceptual model to the DBMS and is how the DBMS sees the database [1]. The internal level is software dependent because if you change the DBMS then the internal level needs to be updated to fit the new requirements. The internal level is however hardware independent.

The Physical Data Model

This is considered the lowest level of abstraction. It describes how data is physically stored within the storage media. This model normally requires a definition of the physical storage devices and any access methods defined by the resident operating system. The physical model is neither hardware nor software dependent [1].

SUMMARY

This second lecture has been a review of the database environment. Specifically, we reviewed the various levels of the three-level ANSI/SPARC architecture. A review of data models was done where we learnt of the evolution of various data models including the hierarchical and network model, the relational model, the entity-relationship model, and the object-oriented data model. Additionally, a review of emerging data models was conducted. We finally, looked at the various degrees of data abstraction: external, conceptual, internal, and physical data models.

DISCUSSION TOPIC

Let us assume that you were designing a database for the lands office from the discussion topic in Lecture one. Which data model would you pick to represent your database? Why would you choose that data model?

REFERENCES

[1] Database systems: design, implementation, and management, Coronel, C., & Morris, S, Cengage Learning, 2019.

[2] Database Systems: A Practical Approach to Design, Implementation, and Management, Connolly, T., & Begg, C., Pearson, 2015.

[3] Fundamentals of database systems, Elmasri, R., & Navathe, S. B., Pearson Education Limited, 2016.