COMPUTER ORGANIZATION AND ARCHITECTURE

Lecture 3

Database Analysis and Design

Dr Victoria Mukami

INTRODUCTION

In the previous lecture, we looked at an introduction to the database environment. In this lecture, we follow the same path but go back toward the development and implementation of a database. A database like any other product needs specific steps for development. Specifically, for this lecture, we will start by reviewing the systems development life cycle (SDLC). Once we understand SDLC we will then focus on the database lifecycle (DBLC). Next, we will understand the categories of design including conceptual, logical, and physical design. Finally, we will finish the lecture by understanding the difference between centralized and decentralized database design.

Learning objectives

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

1. Identify the stages of the system development lifecycle
2. Understand the database lifecycle
3. Differentiate between conceptual, logical, and physical design
4. Distinguish between the centralized and decentralized design

OVERVIEW

A database just like any other product needs to be developed before it can be implemented and used. The development of a database follows specific steps to ensure that the organizations' operations are all met and catered for by the database. By taking a step back and remembering the users who use a database, some included the developers and designers. All these users will be used in the development of the database. Let’s take another step back.

Before a house can be built, several things need to be done. First, some land needs to exist where the house will be built. Before starting to build the house, one needs to acquire the services of an architect who will draw the house plans including the structural plans. This will also involve the development of something called a bill of quantities which will show the approximate materials and associated costs for the project. Once that is done of course various approvals will be applied for and finally construction of the house can be done.

This is a good example to explain the creation of a database. Before we can start creating the database, several steps need to be completed. In the next section, we look at the systems development lifecycle of any software within a computer.

SYSTEMS DEVELOPMENT LIFECYCLE (SDLC)

Systems development is the process of creating an information system [1]. An information system can be defined as a set of different components that are used to collect, store and process data. This data is then used as information or knowledge in various facets of an organization, especially in decision making. The systems development lifecycle (SDLC) is the key methodology used in the design and development of a database. There are five or six steps within the systems development lifecycle (Figure 1). The number of steps is dependent on the development team as some teams combine some of the steps. For this lecture, we will concentrate on five steps. These steps are discussed in detail within the next sections.

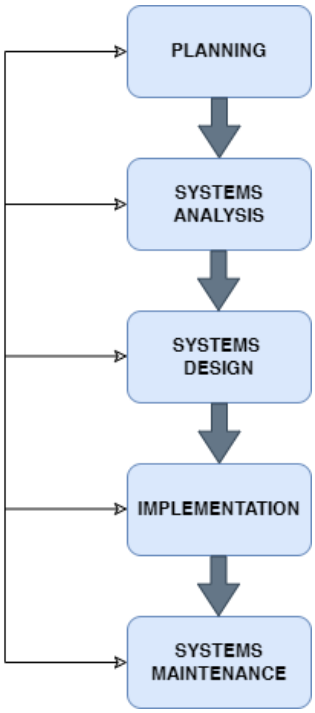

Figure 1: Systems Development Lifecycle

Planning

This is the initial assessment of a company and its needs. This is the first phase that is involved with finding whether the problem brought forward is worth solving and whether there is a need to push forward. The requirements gathering will involve a collection of various solutions to the problem brought forward. The solutions costs and individual benefits are extracted during this phase. This includes any additional resources that may be required such as hardware, technical resources, and personnel.

Systems Analysis

The second phase, system analysis, involves a review of how the current system works. This is to help the analyst understand in more detail the problems posed by the current system. The analyst works with the end-user to identify any problems with the current system that the analyst may have overlooked. The new requirements for the new system are outlined during this phase and several logical diagrams are used to present the findings. The diagrams used include use case diagrams, entity-relationship diagrams, and data flow diagrams.

Systems Design

This phase involves coming up with the various technical specifications that the new system will require. The phase is concerned with the look of the new system. This involves the design of any of the new user interfaces. Various diagrams are used to document and show how the system will work and function.

Implementation

This phase is sometimes split in two: testing and coding and implementation. Generally, this phase includes the development of the system, testing and debugging of the developed system as well as implementing the newly created system. If the organization opted to buy a new system rather than develop it, the phase would involve acquiring the necessary hardware to support the purchased system. Additionally, any employee training may happen during this phase.

Systems Maintenance

Since nothing is perfect, maintenance is an equally important phase. Once users start using the system, then there may be a time that they feel their needs are not being fully met, or that the system is lagging or some of their processes have changed. This brings up the need for various types of maintenance.

- Corrective maintenance is used to solve any system errors [1]
- Adaptive maintenance is necessary to deal with any changes in the way the business runs [1]
- Perfective maintenance is used to enhance and make the system much better [1]

DATABASE LIFECYCLE

The database lifecycle is concerned with the design, development, and maintenance of a database. It entails six steps which include: database planning, database design, implementation, testing, operation, and maintenance [1] as shown in Figure 2.

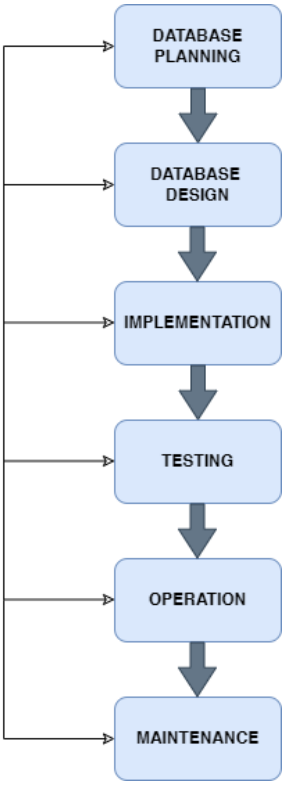

Figure 2: The Database Lifecycle

Each of the phases is discussed below.

Database Planning

This phase is a review of any previous databases in use. The database designer will review the current system in use and determine any problems with it. This phase is like the planning phase of the SDLC. Particularly, the database planning phase will analyse the current problems, design the problems and objectives while finally determining the scope of the new database [1].

Database Design

The database design focuses on the database model that will support the objectives laid out during the first phase [1]. This involves coming up with the user and system requirements. Again, this phase is similar to the SDLC. The output of this phase is the conceptual, logical, and physical design. These will be discussed later during this lecture.

Implementation

Implementation involves installing the Database Management System (DBMS), creating the database, and loading or converting data [1]. The output of the design phase feed into the implementation phase. All the design specifications that were created during the design phase are implemented here. The phase is mainly conducted by the database administrator (DBA).

Testing

Once the database has been developed and implemented, then it is necessary to test and ensure that data is stored correctly within the database. This is still the role of the DBA which ensures that any errors are handled. This is the phase where user access rights and any security policies are implemented. An evaluation of the database inner working is reviewed and corrected in the event there are any issues.

Operation

Once a database has passed the evaluation during the testing phase then it si considered operation [1]. This is the phase where the database starts being used by the various users. The DBA will still evaluate the database as it is being used to ensure that it meets its standards and performs well.

Maintenance

This phase is like the SDLC phase where corrective, preventive and adaptive maintenance is conducted. Maintenance is a necessary phase as it ensures that the DBA can keep track of the performance of the database and can make updates to deal with errors or to upgrade.

CONCEPTUAL, LOGICAL AND PHYSICAL DESIGN

The second phase of the DBLC is the design phase. Part of the output of this phase was the conceptual, logical, and physical design. We discuss each of these designs below.

Conceptual Design

The goal of the conceptual design is to design a database that is independent of the database software and physical details. The conceptual design has four steps

Data analysis and requirements [1]

This involves discovering the characteristics of the data elements. To develop a data model, the designer needs to understand the data types. Various business rules are also crafted here.

Entity-relationship modelling and normalization [1]

Entity Relationship Diagrams and normalization is discussed in detail during lecture 4 and 6 respectively. The business rules crafted during the data analysis and requirements then give way to the creation of the ERD diagrams. Normalization works when determining the relationships within the various tables.

Data model verification [1]

The ER model designed is normally reviewed against any proposed system processes. The aim is to check whether the data model is supported, and this is done using tests.

Distributed database design [1]

This is an optional requirement as not all databases are distributed geographically. Once the data model verification is conducted, then the database may be distributed geographically. Various processes may end up being physically stored in different places.

Logical Design

The logical database design points out that the process of constructing a model is independent of a particular DBMS [2]. This phase results in a logical model that is crafted from the conceptual model. Normalization is used here to test the correctness of the logical model [2]. There are four steps within the development of the logical model.

Map the conceptual model to logical model components [1].

During this step, the entities and relationships are discovered and mapped out. This involves mapping the entity-relationship diagram into a set of tables, columns and having various constraints.

Validate the logical model using normalization [1].

Once the tables and columns are created, normalization is used to ensure that the correct relationships are mapped out and to ensure multivalued attributes do not exist.

Validate the logical model integrity constraints [1].

All the constraints that have been identified need to be supported by the logical data model.

Validate the logical model against user requirements [1].

The logical model is finally validated against all the user requirements to check whether it meets them.

Physical Design

The physical design is the final sub-phase of the design phase. This is the phase where the designer decides how the database will be implemented [1]. The physical design involves:

- Define data storage organization [2].
- Define integrity and security measures [2].
- Determine performance measurements [2].

This phase involves physically knowing or calculating and understanding the physical storage required, ensuring integrity and security across the database, and finally checking on the database performance.

CENTRALIZED AND DECENTRALIZED DATABASE

In this final section, we discuss the centralized and decentralized database. There are two main approaches to database design, top-down and bottom-up approaches, that are used. The top-down design starts by identifying any data sets and then defining the elements of those data sets [1]. The bottom-up approach first identified the data elements and groups them into data sets [1]. These two approaches are influenced by the scope and size of the company, the management style, and the company structure (centralized or decentralized [1].

The **centralized design** is best used for small and simple databases and can be carried out by a DBA. The environment allows for a designer to take care of the design phase without the use of a very large team.

The **decentralized design** is used for complex databases that may have complex operations. The database team within a decentralized design may be made up of several individuals who can deal with the complexities of the database.

SUMMARY

During this lecture, we have conducted a review of the systems development lifecycle. We focused on the database life cycle and brought out the difference between the SDLC and the DBLC. This then led to a review of the design phase of the DBLC with a specific interest in the conceptual, logical, and physical design. Finally, a review of centralized and decentralized databases was covered.

DISCUSSION TOPIC

During Lecture 2, we picked a data model for the lands office database. Based on the data model that you picked and using the DBLC determine what each step of the lifecycle would entail and how each step might be completed and the individuals that might be required for each phase.

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.