Database Planning, Design, and Administration

Ms. Kapukha

Software Development

- Software has now surpassed hardware as the key to the success of many computerbased systems.
  - 80–90% do not meet their performance goals;
  - about 80% are delivered late and over budget;
  - around 40% fail or are abandoned;
  - under 40% fully address training and skills requirements;
  - less than 25% properly integrate enterprise and technology objectives;
  - Just 10–20% meet all their success criteria.
- There are several major reasons for the failure of software projects including:
  1. lack of a complete requirements specification;
  2. lack of an appropriate development methodology;
  3. Poor decomposition of design into manageable components.

ISLC VS SDLC

- As a solution to these problems, a structured approach to the development of software was proposed called the **Information Systems Lifecycle** (ISLC) or the **Software Development Lifecycle** (SDLC).
- However, when the software being developed is a database system the lifecycle is more specifically referred to as the Database System Development Lifecycle (DSDLC).
- The database is a fundamental component of an information system, and its development and usage should be viewed from the perspective of the wider requirements of the organization.
- Typically, the stages in the lifecycle of an information system include: planning, requirements collection and analysis, design, prototyping, implementation, testing, conversion, and operational maintenance.

The Database System Development Lifecycle

- The management activities that allow the stages of the database system development lifecycle to be realized as efficiently and effectively as possible:

1) Database Planning

- An important first step in **database planning** is to clearly define the **mission statement** for the database system. The mission statement defines the major aims of the database system.
- A mission statement helps to clarify the purpose of the database system and provide a clearer path towards the efficient and effective creation of the required database system.
- Once the mission statement is defined, the next activity involves identifying the **mission objectives**.
- Each mission objective should identify a particular task that the database system must support.
- Database planning should also include the **development of standards** that govern how data will be collected, how the format should be specified, what necessary documentation will be needed, and how design and implementation should proceed

2. System Definition

- Describes the scope and boundaries of the database application and the major user views.
- Before attempting to design a database system, it is essential that we first identify the boundaries of the system that we are investigating and how it interfaces with other parts of the organization's information system.
- It is important that we include within our system boundaries not only the current users and application areas, but also future users and applications.

3. Requirements Collection and Analysis

- The process of collecting and analyzing information about the part of the organization that is to be supported by the database system, and using this information to identify the requirements for the new system.
- This stage involves the collection and analysis of information about the part of the enterprise to be served by the database. There are many techniques for gathering this information, called **fact-finding techniques**.
- This information is then analyzed to identify the requirements (or features) to be included in the new database system. These requirements are described in documents collectively referred to as **requirements specifications** for the new database system.

4. Database Design

- Approaches of database design

  a) Bottom-up

  b) Top Down

- Data modelling
- The two main purposes of data modeling are to:

a) assist in the understanding of the meaning (semantics) of the data and

b) to facilitate communication about the information requirements.

Phases of Database Design

a)Conceptual Database Design Phase

- The process of constructing a model of the data used in an enterprise, independent of *all* physical considerations.
- Involves the creation of a conceptual data model of the part of the enterprise that we are interested in modelling.
- The data model is built using the information documented in the users' requirements specification.
- Conceptual database design is entirely independent of implementation details such as the target DBMS software, application programs, programming languages, hardware platform, or any other physical considerations.
- Throughout the process of developing a conceptual data model, the model is tested and validated against the users' requirements.

b) Logical database design

- This process of constructing a model of the data used in an enterprise based on a specific data model, but independent of a particular DBMS and other physical considerations.
- The conceptual data model created in the previous phase is refined and mapped on to a logical data model. The logical data model is based on the target data model for the database (for example, the relational data model).
- Whereas a conceptual data model is independent of all physical considerations, a logical model is derived knowing the underlying data model of the target DBMS.
- However, we ignore any other aspects of the chosen DBMS and, in particular, any physical details, such as storage structures or indexes.
- Throughout the process of developing a logical data model, the model is tested and validated against the users' requirements.
- The technique of normalization is used to test the correctness of a logical data model. Normalization ensures that the relations derived from the data model do not display data redundancy, which can cause update anomalies when implemented.

c.Physical database design

- The process of producing a description of the implementation of the database on secondary storage; it describes the base relations, file organizations, and indexes used to achieve efficient access to the data, and any associated integrity constraints and security measures.
- It is the phase during which the designer decides how the database is to be implemented
- In general, the main aim of physical database design is to describe how we intend to physically implement the logical database design.
- For the relational model, this involves:
  - creating a set of relational tables and the constraints on these tables from the information presented in the logical data model;
  - identifying the specific storage structures and access methods for the data to achieve an optimum performance for the database system;
  - Designing security protection for the system.

5. Application Design

- The design of the user interface and the application programs that use and process the database
- Database and application design are parallel activities of the database system development lifecycle.
- In most cases, it is not possible to complete the application design until the design of the database itself has taken place.
- We must ensure that all the functionality stated in the users' requirements specification is present in the application design for the database system.
- This involves designing the application programs that access the database and designing the transactions, (that is, the database access methods).
- In addition to designing how the required functionality is to be achieved, we have to design an appropriate user interface to the database system.
- This interface should present the required information in a 'user-friendly' way.

Prototyping

- Building a working model of a database system.
- A prototype is a working model that does not normally have all the required features or provide all the functionality of the final system.
- The main purpose of developing a prototype database system is to allow users to use the prototype to identify the features of the system that work well, or are inadequate, and if possible to suggest improvements or even new features to the database system.
- Prototypes should have the major advantage of being relatively inexpensive and quick to build.
- There are two prototyping strategies in common use today: requirements prototyping and evolutionary prototyping. **Requirements prototyping** uses a prototype to determine the requirements of a proposed database system and once the requirements are complete the prototype is discarded.
- While **evolutionary prototyping** is used for the same purposes, the important difference is that the prototype is not discarded but with further development becomes the working database system.

Implementation

- The physical realization of the database and application designs.
- On completion of the design stages (which may or may not have involved prototyping), we are now in a position to implement the database and the application programs.
- The database implementation is achieved using the Data Definition Language (DDL) of the selected DBMS or a Graphical User Interface (GUI), which provides the same functionality while hiding the low-level DDL statements.
- The DDL statements are used to create the database structures and empty database files. Any specified user views are also implemented at this stage.
- The application programs are implemented using the preferred third or fourth generation language (3GL or 4GL).

Data Conversion and Loading

- Transferring any existing data into the new database and converting any existing applications to run on the new database.
- This stage is required only when a new database system is replacing an old system. Nowadays, it is common for a DBMS to have a utility that loads existing files into the new database.
- The utility usually requires the specification of the source file and the target database, and then automatically converts the data to the required format of the new database files.
- Where applicable, it may be possible for the developer to convert and use application programs from the old system for use by the new system.
- Whenever conversion and loading are required, the process should be properly planned to ensure a smooth transition to full operation.

Testing

- The process of running the database system with the intent of finding errors.
- Before going live, the newly developed database system should be thoroughly tested.
- This is achieved using carefully planned test strategies and realistic data so that the entire testing process is methodically and rigorously carried out.
- If testing is conducted successfully, it will uncover errors with the application programs and possibly the database structure.
- As a secondary benefit, testing demonstrates that the database and the application programs *appear* to be working according to their specification and that performance requirements appear to be satisfied.
- In addition, metrics collected from the testing stage provide a measure of software reliability and software quality.

Operational Maintenance

- The process of monitoring and maintaining the database system following installation.
- In the previous stages, the database system has been fully implemented and tested. The system now moves into a maintenance stage, which involves the following activities:
  - Monitoring the performance of the system. If the performance falls below an acceptable level, tuning or reorganization of the database may be required.
  - Maintaining and upgrading the database system (when required). New requirements are incorporated into the database system through the preceding stages of the lifecycle.

Data Administration and Database Administration

- The Data Administrator (DA) and Database Administrator (DBA) are responsible for managing and controlling the activities associated with the corporate data and the corporate database, respectively.
- The DA is more concerned with the early stages of the lifecycle, from planning through to logical database design.
- The Data Administrator (DA) is responsible for the corporate data resource, which includes non-computerized data, and in practice is often concerned with managing the shared data of users or application areas of an organization.
- The database administration staff are more technically oriented than the data administration staff, requiring knowledge of specific DBMSs and the operating system environment.
- Although the primary responsibilities are cantered on developing and maintaining systems using the DBMS software to its fullest extent.

#END