Database Development Life Cycle

CONTENTS

A guide..®
can change your life

ABSTRACT

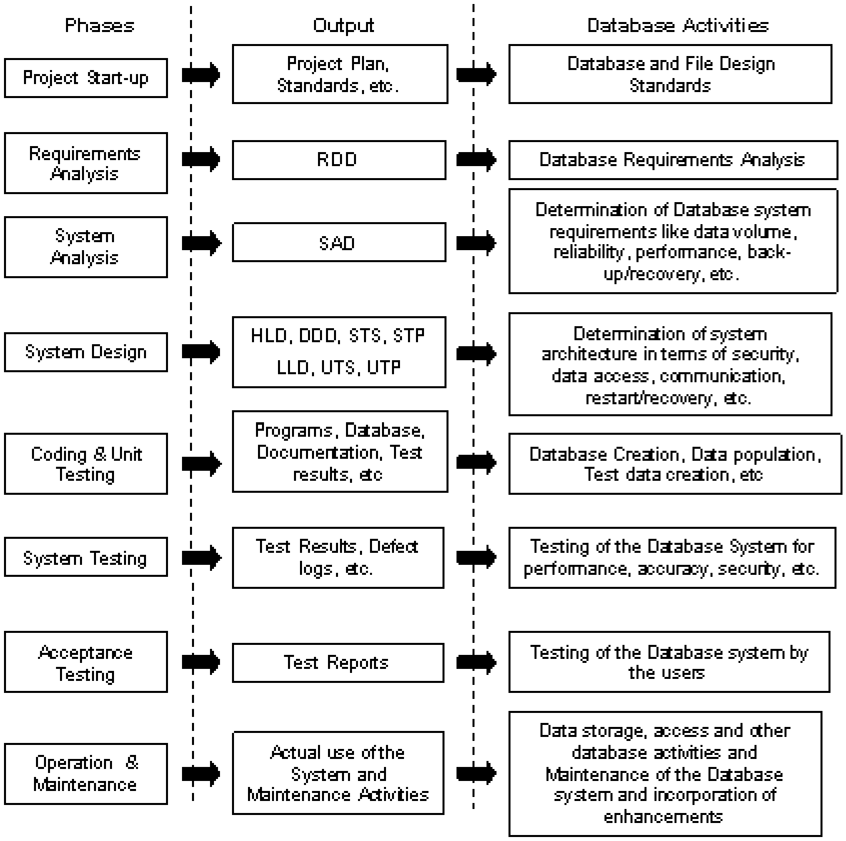
In this article, we will see the different phases of the software development life cycle and the database development life cycle. Database development lifecycle (DDLC) is a subset of the SDLC or we can say that the DDLC is part of the SDLC. The different phases of DDLC are requirements analysis, database design, DBMS evaluation, selection, implementation, data loading, testing, operation, performance tuning, and maintenance. We will also see the different activities performed in each phase and the output generated in each phase.

CONTENTS

INTRODUCTION

The software development is that set of actions required for efficiently transforming the user's need into an effective software solution. Software development process defines the activities required for building the software systems, incorporating the methods and practices to be adopted. It also includes the activities essential for planning the project, tracking its progress and managing the complexities of building the software.

The life span of software systems varies from product to product. During its lifetime, the software goes through various phases. IEEE defines software lifecycle as the period of time that starts when a software product is conceived and ends when the product is no longer available for use. The software development lifecycle (SDLC) typically includes a requirements phase, design phase, implementation phase, testing phase, operation and maintenance phase, and sometimes, retirement phase. The database development life cycle (DDLC) is a part of (or embedded inside) the software development life cycle. Figure 1 gives the various SDLC phases and their relationship with DDLC.



Phases	Output	Database Activities
Project Start-up	Project Plan, Standards, etc.	Database and File Design Standards
Requirements Analysis	RDD	Database Requirements Analysis
System Analysis	SAD	Determination of Database system requirements like data volume, reliability, performance, back-up/recovery, etc.
System Design	HLD, DDD, STS, STP LLD, UTS, UTP	Determination of system architecture in terms of security, data access, communication, restart/recovery, etc.
Coding & Unit Testing	Programs, Database, Documentation, Test results, etc	Database Creation, Data population, Test data creation, etc
System Testing	Test Results, Defect logs, etc.	Testing of the Database System for performance, accuracy, security, etc.
Acceptance Testing	Test Reports	Testing of the Database system by the users
Operation & Maintenance	Actual use of the System and Maintenance Activities	Data storage, access and other database activities and Maintenance of the Database system and incorporation of enhancements

Figure 1 SDLC phases and their relationship with DDLC

DDLC PHASES

In this section we will consolidate these different database related activities and group them into phases that form part of the database development life cycle (DDLC). The different phases of DDLC are:

- Requirements Analysis
- Database Design
- Evaluation and Selection
- Logical Database Design
- Physical Database Design
- Implementation
- Data Loading
- Testing and Performance Tuning
- Operation
- Maintenance

We will now see each of these steps in some detail and find out what exactly is done in each of these phases.

Requirements Analysis

The first step in implementing a database system is to find out what is required. What kind of a database is needed for the organization, what is the volume of the data that must be handled on a day-to-day basis (transaction data), how much data is to be stored in the master files, and so on. To get these information the database analysts should spend a lot of time in the organization talking to people—the end users—assessing their day-to-day tasks and analyzing the data needs and the amount of data they process and produce. In order to get an accurate estimate of the data needs, the volume of data that is to be handled by the proposed system, the database analysts need to spend time studying the system. Besides the data volume, the analysts should also gather detailed and accurate information on the number of users, the number of people simultaneously accessing the system, the performance expectations of the users, the rate at which the database will grow and so on. The main goals of this phase of DDLC are:

- Study the existing system – Here the objective is to identify the data needs and requirements, the data volume, the performance requirements, the security issues, data access restrictions, number and types of users, the growth rate of the database and so on.
- Define problems and constraints of the database environment – Here the objective is to find out the problems that will have to be solved when the database is designed. What are the areas where the designers have to be careful, in order to avoid producing a bad database design. Here the analysts should study the paper forms that are being used in the organization and then decide which data items are needed and which are redundant. This is a challenging job as it involves converting the grievances and complaints and expectations of the end users (who will not be familiar with the intricacies of database design) into meaningful requirement definitions. Here the analysts should sit with the users, question the need and necessity of each data item, the problems that can arise if a particular data item is missing, and so on and then arrive at the problem areas, constraints and limitations of the database system.
- Define the design objectives – The proposed database system should help in solving the information requirements of the organization and facilitate the smooth and efficient flow of information and help in better decision-making. Most organizations that plan to have a database usually have islands of information—each department having its own database. These islands should be removed and the information should be centralized. In order to do this the analysts should gather the information stored in the departmental databases and then integrate all of them to create an enterprise-wide information system. To do this—to create a centralized database that contains the information for the entire organization—the designers should have the overall view of the organizational goals and the organization’s structure. They should also address issues like data integrity, security, concurrency, performance and so on.
- Define standards and guidelines – The database design standards, the naming conventions, the documentation standards, diagramming standards and other conventions that are to be followed in the DDLC phases are also formulated during this phase.

Database Design

In this phase the database designers will decide on the database model that is ideally suited for the organization’s needs. The database designers will study the documents prepared by the analysts in the requirements analysis phase and then will go about developing a system that satisfies the requirements. In this phase the designers will try to find answers to the following questions:

- What are the problems in the existing system and how they could be overcome?

- What are the information needs of the different users of the system and how could the conflicting requirements be balanced?
- What data items are required for an efficient decision-making system?
- What are the performance requirements for the system?
- How should the data be structured?
- How will each user access the data?
- How the data is entered into the database or how is the data loaded to the database?
- How much data will be added to the database each day/week/year?

First a conceptual design of the database is created. In the conceptual design stage, data modeling is used to create an abstract database structure that represents the real-world scenario. The conceptual model will be a true representation of the real world, only if the requirements analysis is properly done, as it needs a thorough understanding of the business and functional areas. At this stage the hardware or the database model that is to be used are not decided—the conceptual design is hardware and software independent. This is important as the system can be implemented on any software/hardware platform chosen later. The different steps in the conceptual design are as follows:

- Data Analysis and Requirements Definition – In this step the data items and their characteristics are determined. The data items that are required for successful information processing and decision-making are identified and their characteristics are recorded. Questions like what kind of information is needed, what outputs (reports and queries) should the system generate, who will use the information, how and for what purpose it will be used, what are the sources of the information, etc. will be answered in this step.
- Data Modeling and Normalization – In this step the database designer creates a data model of the system. The business contains entities and relationships. Each entity will have attributes. In this step the business entities and relationships are transformed into a data model (usually an E-R model) using E-R diagrams. Now many designers have started using data modeling using UML (Unified Modeling Language) instead of the E-R diagrams. Once the data model is created, then the data will be available in a structured form. All objects (entities, attributes, relations and so on) are defined in a data dictionary and the data is normalized. During the process the designer will group the data items, define the tables, identify the primary keys, define the relationships (one-to-one, one-to-many or many-to-many), create the data model, normalize the data model and so on. Once the data model is created, it is verified against the proposed system in order to ascertain that the proposed model is capable of supporting the real-world system. So the data model is tested to find out whether the model can perform the various database operations (data loading, access, querying, insertion, modification and so on) and whether the data model takes care of the issues of data security, integrity, and concurrency and so on.

Evaluation and Selection

Once the data model is created, tested and verified, the next step is to evaluate the different database management systems and select the one that is ideally suited for the needs of the organization. Here the very important thing to be remembered is that the end-user's representatives should be made part of the group that evaluates and selects the database system for the organization. The main factors that influence the selection of the DBMS are:

- Cost of the System – The cost includes the purchase price, cost of operation, maintenance, site license, installation, training, data migration, data conversion, etc.
- Features and Tools – Not all database management systems are created equal. Some systems have more features than others. Some have a lot of data administration, querying and report writing tools as part of the system. For example, the availability of Query-By-Example (QBE), screen painters, report generators, query generators, data loaders, data dictionaries and so on make the DBMS easy-to-use and pleasant to work with. Similarly DBA utilities, automated back-up/recovery systems, access control and management systems all make the DBMS more attractive to the buyer. But here a word of caution: you should not go out and buy the DBMS that has the most number of features and tools. You should buy the one that has the most number of features and tools that you want.
- Customer Support and Training – Another factor that will influence the selection of the DBMS is the efficiency of the customer service (after sales service) that the DBMS vendor offers. Also the ease of training that is the ease with which users can be trained in the system is another factor.
- Underlying Data Model – The purchasing decision is to a very large extent influenced by the underlying data model—Hierarchical, Network, Relational, Object-Relational and so on.
- Portability – The DBMS selected should be portable across platforms and languages if there is a requirement for that.
- Hardware Requirements – The hardware requirements of the DBMS is another important factor. If the DBMS needs high-end systems to perform efficiently, then the cost of these hardware components should also be considered while making the selection.

Logical Database Design

Once the different database management systems are evaluated and the one best suited for the organization is selected, the next step in the DDLC is the logical database design. Logical design is dependent on the choice of the database model that is used. Once the database model is identified, the conceptual design can be mapped to the logical design that is tailored to the selected database model. So the logical design is software dependent.

In the logical design stage, the conceptual design is translated into internal model for the selected DBMS. This includes mapping all objects in the model to the specific constructs used by the selected database software. For example, for a RDBMS, the logical design includes the design of tables, indexes, views, transactions, access privileges, etc. Thus the logical design transforms the software-independent conceptual model into a software dependent model.

Physical Database Design

Physical database design is the process of selecting the data storage and data access characteristics of the database. The storage characteristics depends on the type of devices supported by the hardware, the type of data access methods supported by the system and the DBMS. Physical design translates the logical design into hardware dependent one. Physical design is particularly important for older database models like hierarchical and network models. Relational, object-relational, object-oriented and deductive models are much more insulated from the physical layer details than the older database models. But even in the case of modern database models, physical design has great significance as a bad design can result in poor performance. In the case of distributed databases the physical design becomes more complex as the networking and communications issues also come into the picture.

Implementation

In most databases a new database implementation requires the creation of special storage related constructs to house the end user tables. These constructs usually include storage group, tablespaces, data files, tables and so on.

Data Loading

After creating the database, the data must be loaded into the database. If the data to be loaded into the database is currently stored in a different system or in a different format, then the data needs to be converted and then migrated to the new database. Data conversion and migration tools and utilities are available with almost all database management systems in the marketplace. There are also a host of third-party tools to accomplish these tasks.

Testing and Performance Tuning

Once the data is loaded into the database the database is tested and fine-tuned for performance, integrity, concurrent access and security constraints. The testing and performance tuning occurs in parallel with the testing and performance tuning of the application programs. Sometimes, the performance degradation of the database is due to the inefficient code in the application program.So however hard the database administrator tries to fine-tune the database parameters, unless and otherwise, the application program logic is changed to a more efficient one, the performance will not improve. So it is important that the database administrators and application programmers work hand-in-hand during this phase.

Operation

Once the data is loaded to the database and it is tested, the database is released into production (along with the application programs). At this stage the database is considered to be operational and the database, its management, its users and the application programs together form an information system. During the operational phase, the database is accessed by the users and application programs, new data is added, the existing data is modified and some obsolete data is deleted. The database administrators perform the administrative tasks like performance tuning, storage space creation, access control, database back up and so on. It is during the operational phase that the database delivers its usefulness as a critical tool in management decision-making and help in the smooth and efficient functioning of the organization.

Maintenance

Once the database is released into production, it will not remain as it was designed, New business requirements, need for new information, acquisition of new data and similar factors will make it necessary to make modifications and enhancements to the existing design. So the database administrators will definitely receive requests for more storage space, changes in the database design, addition of tables, addition of new users, removal of users who have left the organization, changes in the access privileges of the users and so on. The main tasks in this phase are:

- Database backup and recovery

- Performance tuning
- Database design modifications
- Database access management
- Database audits (access audits, usage audits, security audits, etc.)
- Usage monitoring
- Hardware maintenance
- DBMS Software upgradation

SUMMARY

We have seen the different phases of the software development life cycle and the database development life cycle. Database development lifecycle (DDLC) is a subset of the SDLC or we can say that the DDLC is part of the SDLC. The different phases of DDLC are requirements analysis, database design, DBMS evaluation, selection, implementation, data loading, testing, operation, performance tuning, and maintenance. We saw the different activities performed in each phase and the output generated in each phase.

By
Hitesh Mahapatra
B.Tech (IT), M.Tech (CSE)