Chapter 4 : Database Design

Information System Lifecycle
Database System Development Lifecycle

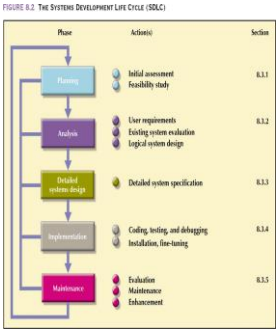
Software Depression

- In late 1960s, exist issue on 'software crisis', now refer to as the 'software depression'.
- Many major software projects were late, over budget, unreliable, difficult to maintain and performed poorly.
- Major reasons for failure of software projects includes:
  - lack of a complete requirements specification;
  - lack of appropriate development methodology;
  - poor decomposition of design into manageable components.
- Structured approach to development was proposed called **Information Systems Lifecycle (ISLC)** or **SDLC**.

The Systems Development Life Cycle (SDLC)

- Divided into five phases:
  - **Planning**
  - **Analysis**
  - **Detailed systems design**
  - **Implementation**
  - **Maintenance**
- Iterative rather than sequential process.

FIGURE 8.2 THE SYSTEMS DEVELOPMENT LIFE CYCLE (SDLC)



Database System Development Lifecycle (DDLC)

- Database planning
- System definition
- Requirements collection and analysis
- Database design
- DBMS selection (optional)
- Application design
- Prototyping (optional)
- Implementation
- Data conversion and loading
- Testing
- Operational maintenance

Database Planning

- ***Management activities that allow stages of database system development lifecycle to be realized as efficiently and effectively as possible.***
- Must be integrated with overall IS strategy of the organization.
- Mission statement for the database project defines major aims of database application.
- Those driving database project normally define the mission statement.
- Mission statement helps clarify purpose of the database project and provides clearer path towards the efficient and effective creation of required database system.

Database Planning – Mission Objectives

- Once mission statement is defined, *mission objectives* are defined.
- Each objective should identify a particular task that the database must support.
- May be accompanied by some additional information that specifies the ***work to be done***, the ***resources*** with which to do it, and the ***money*** to pay for it all.
- Database planning should also include development of standards that govern:
  - how data will be collected,
  - how the format should be specified,
  - what necessary documentation will be needed,
  - how design and implementation should proceed.

System Definition

- **Describes *scope* and *boundaries* of database system and the major user views.**
- User view defines what is required of a database system from perspective of:
  - a particular job role or
  - enterprise application area
- Identifying user views helps ensure that no major users of the database are forgotten when developing requirements for new system.

Requirements Collection and Analysis

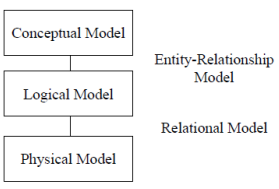
- ***Process of collecting and analyzing information about the part of organization to be supported by the database system and using this information to identify users' requirements of new system.***
- Information is gathered for each major user view including:
  - a description of data used or generated;
  - details of how data is to be used/generated;
  - any additional requirements for new database system.
- Information is analyzed to identify requirements to be included in new database system. Described in the requirements specification.

Database Design

- ***Process of creating a design for a database that will support the enterprise's mission statement and mission objectives for the required database system.***
- Two (2) main approaches include:
  - **Top-down**-starts with the development of data model that contain high level entities and relationship. Used for complex system. Represented by ERD.
  - **Bottom-up** – for simple database begins at the fundamental level of attribute through analysis of the association between attributes, entities and relationship. Represented by normalization.

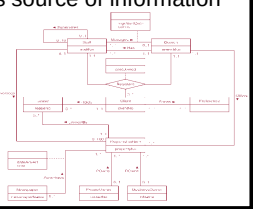
- Main purposes of data modeling include:
  - to assist in understanding the meaning (semantics) of the data;
  - to facilitate communication about the information requirements.
- Building data model requires answering questions about entities, relationships and attributes.
- A data model ensures we understand:
  - each user's perspective of the data;
  - nature of the data itself, independent of its physical representations;
  - use of data across user views.

- Three phases of database design:
  - Conceptual database design
  - Logical database design
  - Physical database design.



Conceptual Database Design

- ***Process of constructing a model of the data used in an enterprise, independent of all physical considerations***.
- Data model is built using the information in users' requirements specification.
- Conceptual data model is source of information for logical design phase.
- **Example :**
  Entity Relationship Modeling



Logical Database Design

- ***Process of constructing a model of the data used in an enterprise based on a specific data model (e.g. relational), but independent of a particular DBMS and other physical considerations***.
- Conceptual data model is refined and mapped on to a logical data model.
- **Example :** Relational Model

Relational Model

Activity Code	Activity Name
23	Patching
24	Overlay
25	Crack Sealing

Key = 24

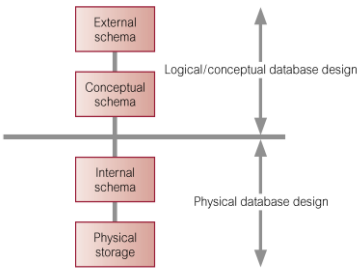
Activity Code	Date	Route No.
24	01/12/01	I-95
24	02/08/01	I-66

Date	Activity Code	Route No.
01/12/01	24	I-95
01/15/01	23	I-495
02/08/01	24	I-66

Physical Database Design

- ***Process of producing a description of the database implementation on secondary storage***.
- Describes base relations, file organizations and indexes used to achieve efficient access to data.
- Also describes any associated integrity constraints and security measures.
- Tailored to a specific DBMS system.

Three-Level ANSI-SPARC Architecture and Phases of Database Design



DBMS Selection

- ***Selection of an appropriate DBMS to support the database system.***
- Undertaken at any time prior to logical design provided sufficient information is available regarding system requirements.
- Main steps to selecting a DBMS:
  - define Terms of Reference of study;
  - shortlist two or three products;
  - evaluate products;
  - recommend selection and produce report.

Application Design

- ***Design of user interface and application programs that use and process the database.***
- Database design and application design are parallel activities.
- Includes two important activities:
  - transaction design;
  - user interface design.

Application Design - Transactions

- An action or series of actions carried out by a single user or application program, which accesses or changes content of the database.
- Should define and document the high-level characteristics of the transactions required.
- Important characteristics of transactions:
  - data to be used by the transaction;
  - functional characteristics of the transaction;
  - output of the transaction;
  - importance to the users;
  - expected rate of usage.
- Three main types of transactions: retrieval, update, and mixed.

User Interface Design

- **Useful guidelines** to follow when designing forms or reports:-
  - a) meaningful title
  - b) comprehensible instructions
  - c) logical grouping and sequencing of fields
  - d) visually appealing layout of the form/report
  - e) familiar field labels
  - f) consistent terminology and abbreviations
  - g) consistent use of colours
  - h) visible space and boundaries for data entry
  - i) convenient cursor movement

Copyright © Ani Yusnita Abdul Rahman 2016

- j) error correction for individual characters and entire fields
- k) error messages for unacceptable values
- l) optional fields marked clearly
- m) explanatory messages for fields
- n) completion signal

Prototyping

- ***Building working model of a database system, which allows the designers or users to visualize and evaluate the system.***
- 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 for the database system.
- It also use to clarify the user's requirements and to evaluate feasibility of a particular system design.
- A prototype is a working model that does not normally have all the required features or provide all functionality of the final system.

Copyright © Ani Yusnita Abdul Rahman 2016

Implementation

- ***Physical realization of the database and application designs.***
  - Use DDL to create database schemas and empty database files.
  - Use DDL to create any specified user views.
  - Use 3GL or 4GL to create the application programs. This will include the database transactions implemented using the DML, possibly embedded in a host programming language.

Copyright © Ani Yusnita Abdul Rahman 2016

Data Conversion and Loading

- ***Transferring any existing data into new database and converting any existing applications to run on new database.***
- Only required when new database system is replacing an old system.
  - DBMS normally has utility that loads existing files into new database.
- May be possible to convert and use application programs from old system to a new system.
- Should be properly planned to ensure a smooth transition to full operation.

Copyright © Ani Yusnita Abdul Rahman 2016

Testing

- ***Process of running the database system with intent of finding errors.***
- Use carefully planned test strategies and realistic data.
- Testing cannot show absence of faults; it can show only that software faults are present.
- Demonstrates the database and application programs *appear* to be working according to requirements.
- Should also test usability of system.
- Evaluation conducted against a usability specification.

Copyright © Ani Yusnita Abdul Rahman 2016

Examples of criteria include:

- Learnability – how long does it take a new user to become productive with the system?
- Performance – how well does the system response match the user's work practice?
- Robustness – How tolerant is the system of user error?
- Recoverability – How good is the system at recovering from user errors?
- Adaptability – How closely is the system tied to a single model of work?

Type of testing :

Top-down testing

Start with high-level system and integrate from the top-down, replacing individual components by stubs where appropriate. Stubs is a component that has the same interface with the module but no function code.

Bottom-up testing

An approach to integration testing where the lowest level components are tested first, then used to facilitate the testing of higher level components. The process is repeated until the component at the top of the hierarchy is tested.

Stress testing

Exercises the system beyond its maximum design load. Stressing the system often causes defects to come to light. Systems should not fail catastrophically. Stress testing checks for unacceptable loss of service or data. Particularly relevant to distributed systems which can exhibit severe degradation as a network becomes overloaded.

Thread testing

Tests the system's response to events as processing threads through the system. Difficult to test because interaction depends on time between processes in the system.

Operational Maintenance

Process of monitoring and maintaining database system following installation.

Involves activities below:

- **Monitoring performance of system** - if performance falls below acceptable level, may require tuning or reorganization of the database.
- **Maintaining and upgrading database system (when required)** – to ensure performance remains within acceptable levels.

Three types:

- ***Corrective maintenance*** in response to systems errors.
- ***Adaptive maintenance*** due to changes in the business environment.
- ***Perfective maintenance*** to enhance the system.

SDLC vs DDLC

SDLC	DDLC
- Identify business functions and develop application system to perform the functions. - function-oriented approach - Develop DFD (data flow diagram) * the flow of data from the functions or processes - System developed is for a short term duration and neglect the long term user requirements	- focus on data analysis use by the functions - System developed is very stable and fulfill the future requirements because data is more stable than system function. - Data-oriented approach - Developed based on database schema and structure