Database Design

Systems Development Life Cycle (SDLC)



Corrective maintenance: system errors Adaptive maintenance: business environment changes Perfective maintenance: system enhancement

The Database Life Cycle (DBLC)



Phase 1: The Database Initial Study

People-oriented: talk with end users

- Overall purpose:
 - Analyze the company situation Operating environment, mission, structure, etc.
 - Define problems and constraints
 - Define objectives Initialization, Interface? Data sharing?
 - Define scope and boundaries

Hardware, software, budget, etc.

 Interactive and iterative processes required to complete the first phase of the DBLC successfully

Summary of Activities in the Database Initial Study



Phase 2: Database Design

- Necessary to concentrate on the data
- Characteristics required to build database model
- Two views of data within system:
 - Business(Manager's) view of data as information source
 WHAT?
 - Designer's view of data structure, its access, and the activities required to transform the data into information

Procedure Flow in the Database Design



Verification VS Validation?

Conceptual Design Tools and Information Sources



Data Model Verification

End-user data views and their required transactions, Access paths and security, Business-imposed data requirements/constraints

- Model must be verified against proposed system processes to corroborate that intended processes can be supported by database model
- Revision of original design starts with a careful reevaluation of entities, followed by a detailed examination of attributes that describe these entities
- Define design's major components as *modules:*
 - An information system component that handles a specific function

DBMS Software Selection

Critical to the information system's smooth operation

Advantages and disadvantages should be

carefully studied

Logical Design

- Used to translate conceptual design into internal model for a selected database management system SQL Server, Oracle, Access, etc.
- Logical design is software-dependent
- Requires that all objects in the model be mapped to specific constructs used by selected database software

A Simple Conceptual Model

[Logic Design]

CLASS_CODE \rightarrow Type: numeric, Range: low value=1000, high value=9999, Display format: 9999, Length: 4 CLASS_DAYS \rightarrow Type: character, Display format: XXX, Valid entries: MWF, TTh, M, T, W, Th, F, S, Length: 3 CLASS_TIME \rightarrow Type: character, Display format: 99:99(24-hour clock), Range: 06:00 to 22:00, Length: 5



Physical Design

- Process of selecting data storage and data access characteristics of the database
- Storage characteristics are a function of device types supported by the hardware, type of data access methods supported by system, and DBMS
- Particularly important in the older hierarchical and network models
- Becomes more complex when data are distributed at different locations

LOCATION + PERFORMANCE!

Phase 3: Implementation and Data Loading

New database implementation requires the

creation of special storage-related constructs to

house the end-user tables and meet specified

technical requirements

Performance

- One of the most important factors in certain database implementations
- Not all DBMSs have performance-monitoring and fine-tuning tools embedded in their software
- There is no standard measurement for database performance

Security

- Data must be protected from access by unauthorized users
- Must provide for the following:
 - Physical security Personnel physical access
 - Password security
 - Access rights
 - Audit trails Check access violations. Although after-the-fact, its mere existence can discourage unauthorized use.
 - Data encryption Data 알호화
 - Diskless workstations No information download allowed

Backup and Recovery

- Database can be subject to data loss through unintended data deletion and power outages
- Data backup and recovery procedures
 - Create a safety valve
 - Allow database administrator to ensure availability of consistent data

Integrity

 Enforced through proper use of primary and foreign key rules

Concurrency Control

- Feature that allows simultaneous access to a database while preserving data integrity
- Failure to maintain can quickly destroy a database's effectiveness

Phase 4: Testing and Evaluation

- Occurs in parallel with applications programming
- Database tools used to prototype applications
- If implementation fails to meet some of the system's evaluation criteria
 - Fine-tune specific system and DBMS configuration parameters
 - Modify the physical design
 - Modify the logical design
 - Upgrade or change the DBMS software and/or the hardware platform

Phase 5: Operation

- Once the database has passed the evaluation stage, it is considered operational
- Beginning of the operational phase starts the process of system evolution

Problems that could not have been foreseen during the testing phase begin to surface!

Phase 6: Maintenance and Evolution

- Required periodic maintenance:
 - Preventive maintenance Action be
 - Corrective maintenance
 - Adaptive maintenance

Action before problem - backup

Action after problem - recovery

Enhance performance, Add new features, etc.

- Assignment of access permissions and their maintenance for new and old users
- Generation of database access statistics
- Periodic security audits
- Periodic system-usage summaries

Parallel Activities in the DBLC and the SDLC



Database Design Strategies

- Two classical approaches to database design:
 - Top-down design
 - Identifies data sets
 - Defines data elements for each of those sets
 - Bottom-up design
 - Identifies data elements (items)
 - Groups them together in data sets

Centralized vs. Decentralized Design

- Database design may be based on two very different design philosophies:
 - Centralized design
 - Productive when the data component is composed of a relatively small number of objects and procedures
 - Decentralized design
 - Used when the data component of system has considerable number of entities and complex relations on which very complex operations are performed

Synonyms and homonyms (같은 철자의 다른 뜻) Entity and entity subtypes Conflicting object definitions