Chapter (16)
Practical Database Design and Tuning

Objectives
• Learn about the design methodology which is a systematic process that the database design activity should undergo.

• Discuss the information system life cycle within organizations with a particular emphasis on the database system.

• Learn about the phases of a database design.

There are some tools designed by various vendors which implement different design methodologies. Examples are: Designer 2000 by Oracle, ERWin, BPWin, and Paradigm Plus by Platinum Technology, Sybase enterprise application studio, ER studio by Embarcadero Technologies, and System Architecture by Popkin Software.

The Organizational Context for Using Database Systems
Many organizations hire a DBA to oversee and control database life-cycle activities. Also, an Information Resource Manager (IRM) has been part of today’s large organizations. Why do we need these people?

• Data is a key resource and must be managed and controlled effectively.
• Organizations are computerized, thus there is a need to keep data on their current up-to-the-minute state.
• Data become more complex so as the relation among them, thus, we need to maintain and modeled such complex relationships.
• Information resources are consolidated in many organizations.

In addition database systems have additional characteristics that makes it attractive:
• Data independence – application is protected from changes in the underlying logical organization, in the physical access path, and storage structure.
• External schemas – same data, multiple applications with its own view of data.
What else make the DB attractive?

• Integration of data across multiple applications into a single database.

• Simplicity of developing new applications using high-level languages like SQL.

• Possibility of supporting casual access for browsing and querying by managers while supporting major production-level transaction processing.

• Availability of powerful resources
  • distributed and client-server DBMSs
  • Availability of tools and techniques (data dictionary and information repositories) that describe the DB structure, constraints, applications, authorizations, and so on.

The Information System Life Cycle

In a large organization, DBS is typically part of information system. That is the reason for looking at the information system life cycle.

The information life cycle is called macro life cycle and the database system life cycle is called micro life cycle.

The macro life cycle typically has the following life cycle:
  
  Feasibility Analysis
  Requirement Collection and Analysis
  Design
  Implementation
  Validation and Acceptance Testing
  Deployment
The Information System Life Cycle – cont.

• **Feasibility Analysis**
  Analyzing potential application areas, evaluate the economic aspects of information collection, evaluate cost-benefit, determine complexity of data and processes, and setting up priorities among applications.

• **Requirement Collection and Analysis**
  Collection of detailed requirement by interacting with potential users.

• **Design**
  1) Design of the database system
  2) Design of application

• **Implementation**
  Information system is implemented, data is loaded, and the database transactions are implemented and tested.

• **Validation and Acceptance Testing**
  System meets the requirements of users. It meets the performance criteria.

• **Deployment, operation, and maintenance**
  If necessary, conversion from an old version to a new one. Keep the system operational through training, and monitoring the system for possible problems and develop the fixes.
The Database Application System Life Cycle

• System Definition
  The scope of the database system, its users, and its application. The interface for different users, the response time constraints, storage and processing needs.

• Database Design
  A complete logical and physical design of the database system on the chosen DBMS is ready.

• Database Implementation
  • Specifying the conceptual, external, and internal database definition,
  • Create empty database files, and
  • Implement the software application.

The Database Application System Life Cycle – cont.

• Loading or Data Conversion
  The database is populated either by loading the data directly or by converting existing files into the database system format.

• Application Conversion
  Any software application from a previous system are converted to the new system.

• Testing and Validation
  Testing and validating the new system.

• Operation
  The new system will be put into operation in parallel with the old system.

• Monitoring and Maintenance
  The system is monitored and maintained. Modification will be made in the data and the software as required.
The Database Design Process

Design the logical and physical structure of one or more database to accommodate the information needs of the users in an organization for a defined set of application.

The goals of database design are:

• Satisfies the information content requirements of the specified users and applications,
• Provides a natural and easy-to-understand structuring of the information, and
• Supports processing requirements (performance, processing time, and storage space)

These goals are very hard to accomplish and measure:
More “naturalness” and “understandability” costs performance

The initial design may have informal or poorly defined requirements. The end product, on the other hand, has rigidly defined database scheme that cannot be modified easily.

Six main Phases of a Database Design Process

Phase (1): Requirements collection and analysis.
Phase (2): Conceptual database design.
Phase (3): Choice of a DBMS.
Phase (4): Data model mapping (also called logical database design).
Phase (5): Physical database design.
Phase (6): Database system implementation and tuning.
Phase (1): Requirements collection and analysis
Identify the new and existing users and applications. Common Activities in this phase:
1) Determine the areas that are affected by the DB. Identify the user groups who are affected.
2) Study if existing documentations and analysis.
3) Determine the current operating environment which includes:
   type of transactions, their frequencies, geographical characteristics regarding users, origin of transactions, destination of reports, the input ad output of transactions, and so forth…
4) Survey containing a set of questions for possible users.

This phase may be the most time consuming of all, but perhaps the most critical. Correcting a requirements error is much more expensive than correcting an error made during the implementation.

Customers participation in the development process increases their satisfactions. They are usually part of the development team.

Phase (2): Conceptual database design
There are two main parts in this phase:
1) Conceptual Schema Design:
The conceptual schema will be included in a DBMS-independent high-level data model, here is why:
   - complete understanding of the database,
   - it is a stable description of the database contents,
   - users and application designers can use it to understand the DBMS easier, and
   - a diagrammatic description of the conceptual schema is an excellent tools used by users, designers, and analysts.
A conceptual high-level data model is characterized by its:
   a) Expressiveness, b) simplicity and understandability, c) minimality,
   d) diagrammatic representation, and e) formality.
   - Approaches to Conceptual Schema Design
   - Strategies for Schema Design
   - Schema (View) Integration

2) Transaction Design
**Approaches to Conceptual Schema Design**

We must first identify the basic components of the schema: *the entity types, relationship types, and attributes.* Also we need to identify the key attributes, cardinality and participation constraints of relationships, weak entity types, and specialization/generalization hierarchies/lattices.

There are two approaches:

- **Centralized (one-shot) schema design approach**
  
The requirements of the different applications and user groups from Phase (1) are merged into a single set of requirements. Then a single schema corresponding to the merged set of requirements will be designed.

- **View integration approach**
  
The requirements are not merged. A schema is designed for each user group or application based only on its own requirements. Later individual schemas will be merged in the view integration phase. These schema are merged or integrated into a global conceptual schema.

**Strategies for Schema Design**

There are several strategies:

- **Top-down strategy**
  
  We start with a schema containing high-level abstractions and then apply successive top-down refinements.

- **Bottom-up strategy**
  
  Start with a schema containing basic abstractions and then combine or add to these abstractions.

- **Inside-out strategy**
  
  A special case of a bottom-up strategy. Central set of concepts are considered first, then the modeling plan spreads outward by considering new concepts.

- **Mixed strategy**
  
  The requirements are partitioned according to a top-down strategy, part of the schema is designed for each partition according to a bottom-up strategy. Then various schema parts are combined.
Schema (View) Integration

For large databases with many expected users and applications, the view integration approach of designing individual schemas and then merging them can be used.

Schema integration can be divided into the following subtasks:
1. Identifying correspondences and conflicts among the schemas
   a. Naming conflicts
   b. Type conflicts
   c. Domain
   d. Conflicts among constraints
2. Modifying views to confirm to one another
3. Merging of views
4. Restructuring