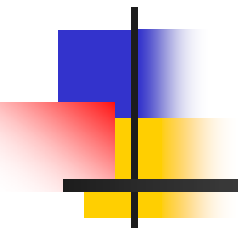
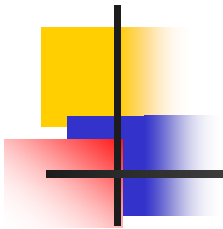
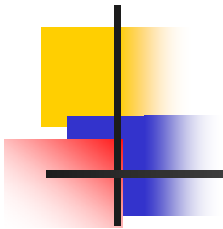


Practical Database Design



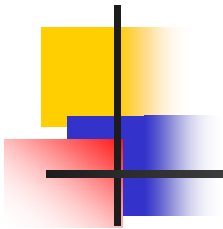
Objectives

- Introduction +
- Reasons for the Failure of DB projects +
- The Role of Information Systems in Organizations +
- Why Organizations Use DB Systems +
- Information System Life Cycle +
- Typical Phases of a Macro Life-Cycle +
- Typical Phases of a Micro Life-Cycle +
- Database Design Process +
- Six Phases of DB design Process +



- Introduction

- In this chapter we move from the theory to the practice of DB design.
- The overall DB design activity has to undergo a systematic process called **design methodology**.
- Generally, the design of small DBs need not be very complicated. But for medium or large DBs, a systematic approach to the overall DB design activity becomes necessary.
- Our goal in this chapter is to discuss DB design methodology in a broad context for a medium and large databases as it is done in large organizations.
  - By large DBs we mean DBs with several tens of gigabytes of data and a schema with more than 30 to 40 distinct entity types.

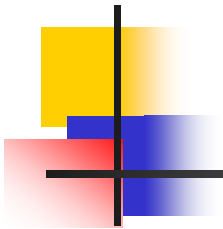


- Reasons for the Failure of DB projects

- The DB design could be simple for small DBs but could be very complex for some DBs with large number of schema objects.
- The more complex the DB design is the higher the probability that its corresponding project will fail.
- The major reasons for the failure of any DB or software project are:
  - Lack of complete requirements specification.
  - Lack of appropriate development methodology.
  - Poor decomposition of design into manageable components.
- The solution:
  - To understanding the role of information system in an organization and
  - To Follow a structured approach towards development of information system, called, Information System Development Life Cycle.

- Role of Information Systems in Organizations

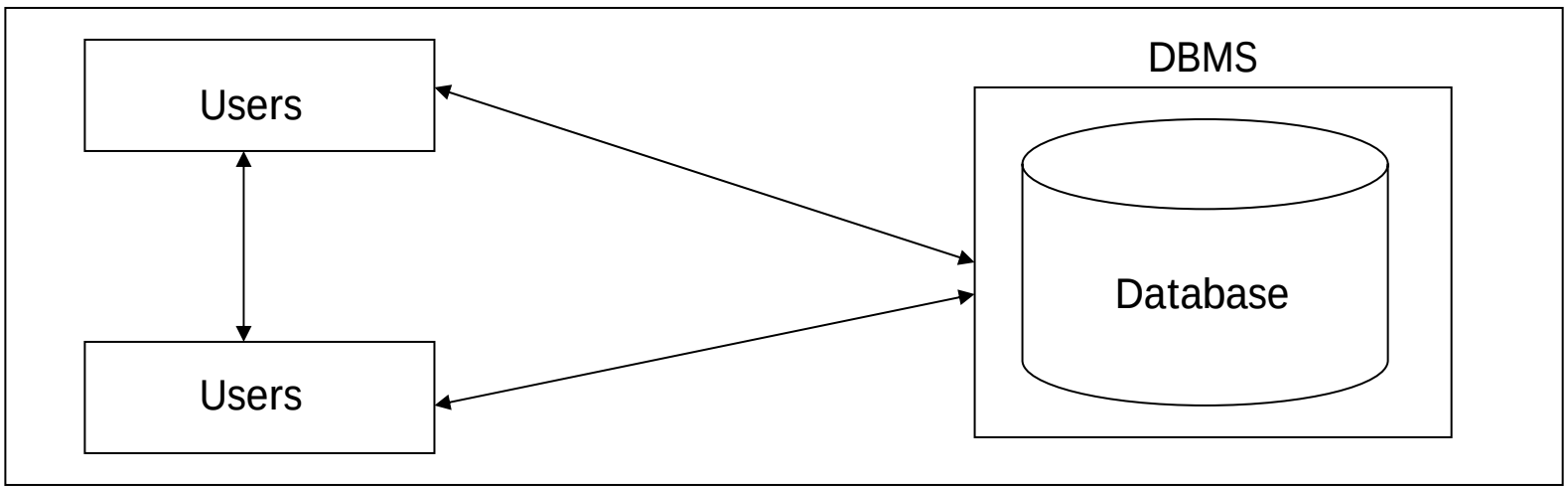
- Data is very valuable resource for any organization. Because when data is processed it changes to information.
- As a result Information Resource Management (IRM) is essential to any organization. The main reason for the need of IRM are:
  - More functions in organizations are computerized, increasing the need to keep large volumes of data available in an up-to-the minute current state.
  - Data is regarded as a corporate resource, its management and control is considered central to the effective working of the organization.
  - As the complexity of the data and the application grows, complex relationships among the data need to be modeled and maintained.
  - There is tendency towards consolidation of information resources in many organizations.

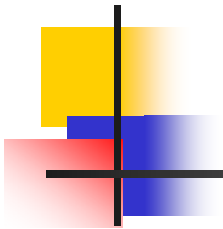


- Why Organizations Use DB Systems

- DB systems meet all the above mentioned IRM needs including:
  - Physical and logical data independence
  - External schema (Views) which allow the same data to be used by multiple applications each having its own view of data.
- Other important features provided by DB systems are:
  - Integration of data across multiple applications into a single database.
  - Simplicity of developing new applications using high-level language like SQL.
  - Possibility of supporting casual access for browsing and querying by managers while supporting major production-level transaction processing.

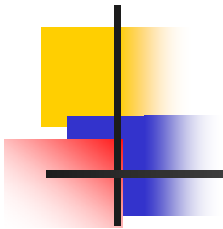
- Information System Life Cycle ...





... - Information System Life Cycle

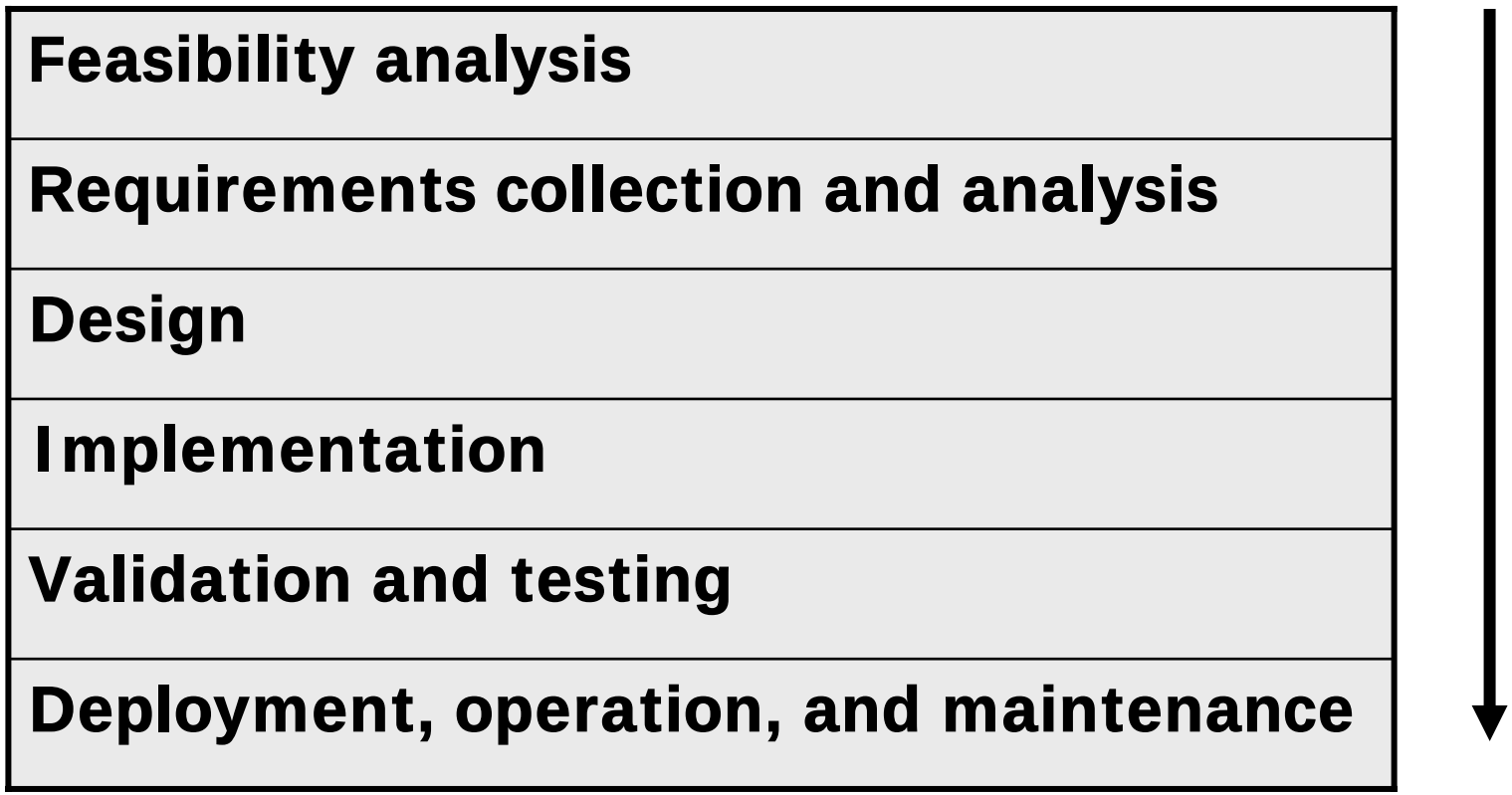
- Resources that enable the collection, management, control, and dissemination of information throughout an organization are called **Information System (IS).**
- Database is the main part of IS.
- Here we examine the typical life cycle of IS and how DB fits into this life cycle.
- The IS life cycle is often called the **macro life cycle** and the DB life cycle is referred to as **micro life cycle.**



- Typical Phases of a Macro Life-Cycle ...

1. **Feasibility analysis**: Analyzing potential application areas, costing, and setting priorities among applications.
2. **Requirements collection and analysis**: Gathering detailed requirements and specifications from users of the system.
3. **Design**: Design of DB systems and associated applications.
4. **Implementation**: IS is implemented, DB loaded and transactions performed.
5. **Validation and testing**: The system is tested against performance criteria and behavior specifications.
6. **Deployment, operation, and maintenance**: The system is deployed in real life. As new requirements crop up, they are passed through all previous phases and later incorporated into the system.

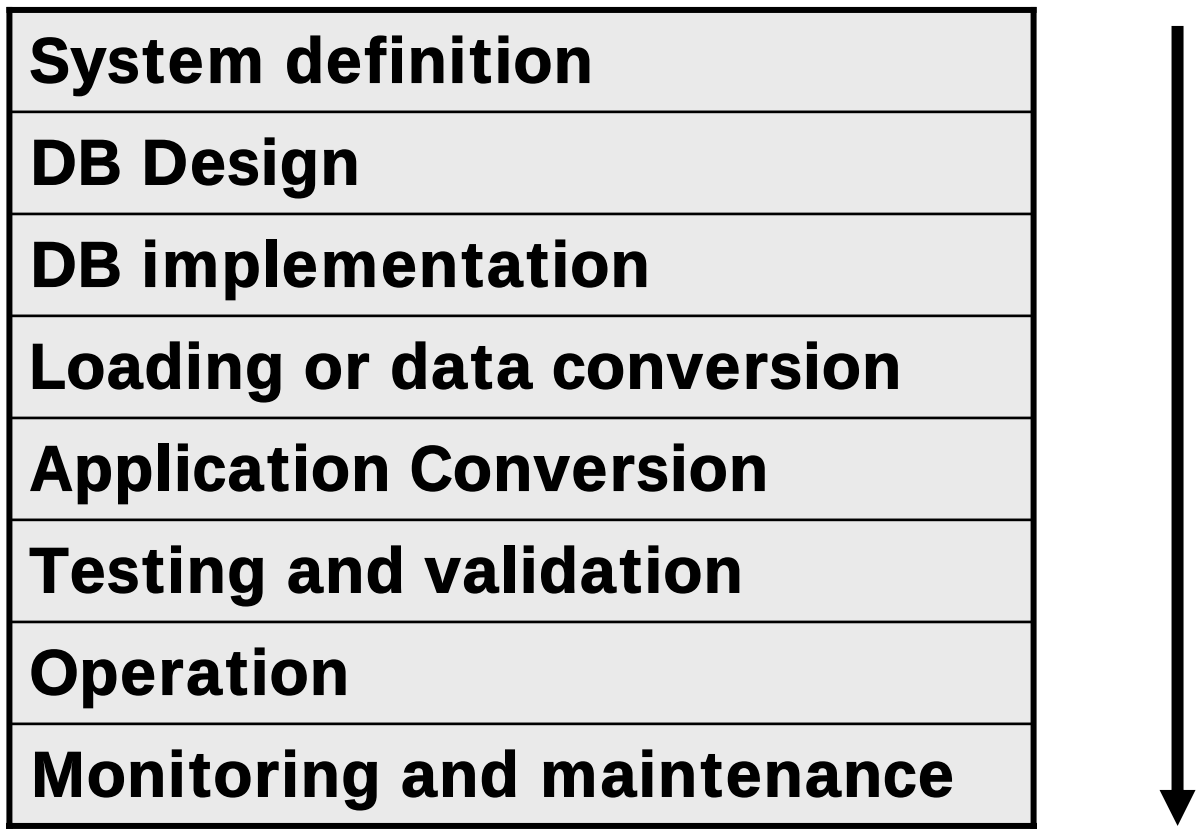
... - Typical Phases of a Macro Life-Cycle

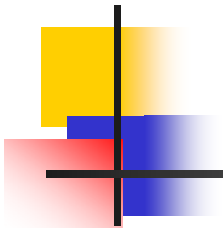


- Typical Phases of a Micro Life-Cycle ...

1. **System definition**: The scope of DB systems, its users, and its applications are defined.
2. **DB Design**: At the end of this phase, complete logical and physical design of the chosen DBMS is ready.
3. **DB implementation**: Define conceptual, external, internal database definitions, creating empty DB files.
4. **Loading or data conversion**: Loading the system data to the DB system format.
5. **Application Conversion**: Any software application from a previous systems are converted into the new system.
6. **Testing and validation**: The new system is tested and validated.
7. **Operation**: The DB system and its applications are put into operation.
8. **Monitoring and maintenance**: The system is constantly monitored and maintained.

... - Typical Phases of a Micro Life-Cycle

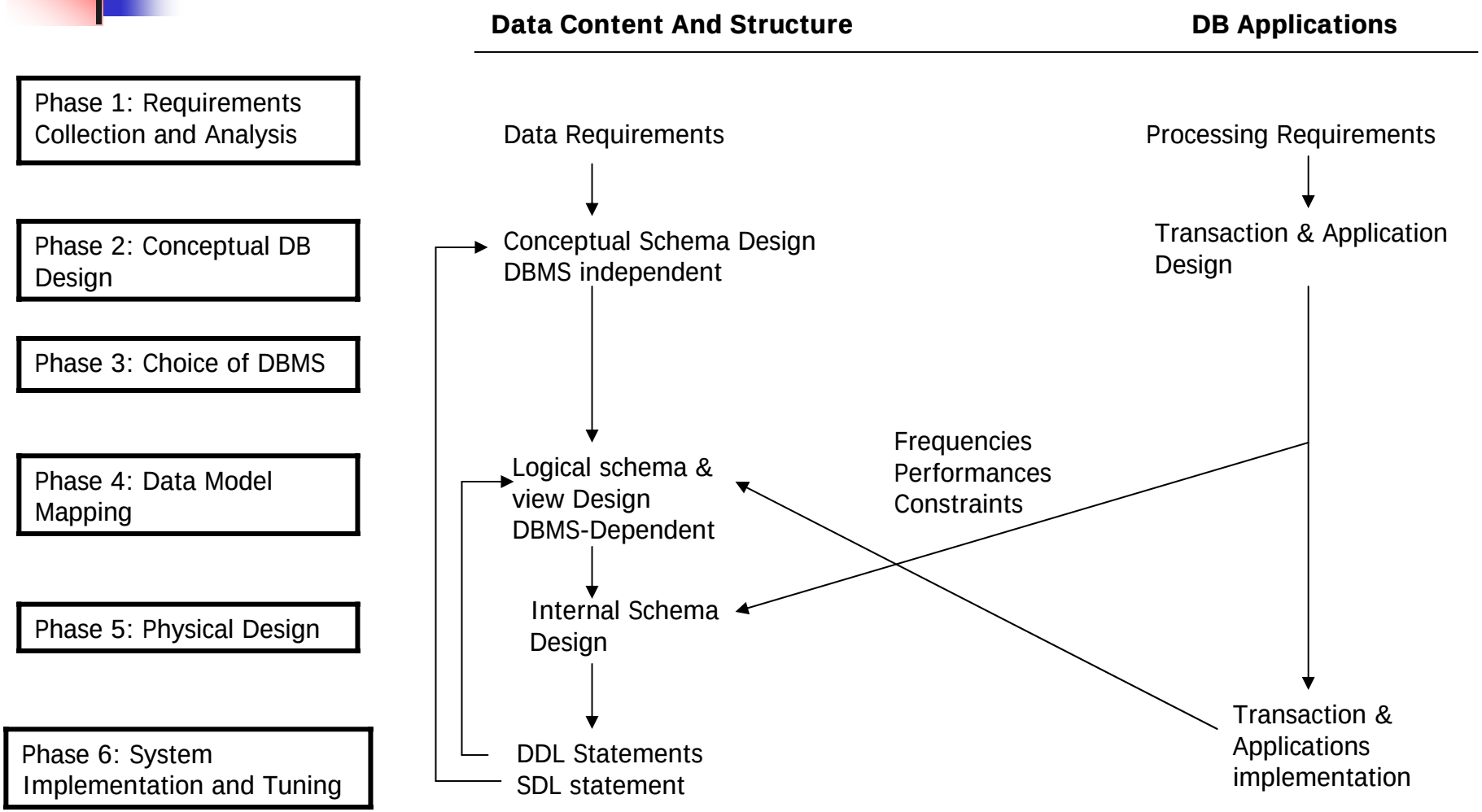


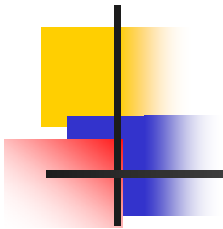


- Database Design Process

- We will now focus on step 2 of the database application life, cycle, which is the DB design.
- The problem of DB design can be stated as follows:
  - Design the logical and physical structures of one or more DBs to accommodate the information needs of the user in an organization for defined set of applications.
- Goals of the DB design:
  - Provide a natural and easy-to-understand structuring of the information.
  - Satisfy the data requirements of user or application.
  - Support processing requirements and any performance objectives such as response time, processing time, and storage space.

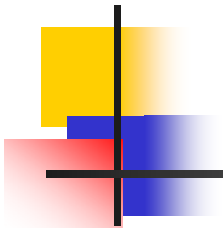
- Six Phases of DB design Process





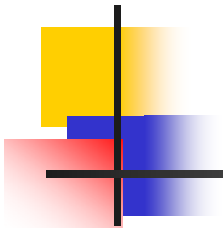
-- Phase 1: Requirements Collection and Analysis

- Activities:
  - Scope of use of application and users
  - Analysis of existing application and documentation
  - Transaction details
  - Feedback from users/customers
  - Interaction with customers/users
    - The designers live in the environment they are designing.
  - Formal Representation of the collected information
    - OOA (Object-oriented Analysis)
    - DFD (Data Flow Diagrams)
  - Use of CASE tools



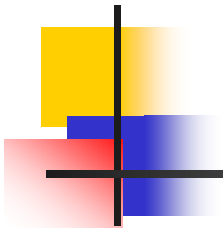
-- Phase 2: Conceptual DB Design

- This phase has two activities:
  - Phase 2a: Conceptual schema design
  - Phase 2b: Transaction and Application Design



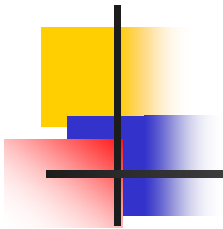
--- Phase 2a: Conceptual Schema Design ...

- Conceptual schema design is DBMS-independent because:
  - Each DBMS has idiosyncrasies and restrictions specific to it
  - Change of DBMS and design considerations should not affect the conceptual schema.
  - High level data model is more expressive and general than the data models of individual DBMS
  - Standard diagrammatic description of conceptual schema aids in communicating to DB users, designers, and analysts.



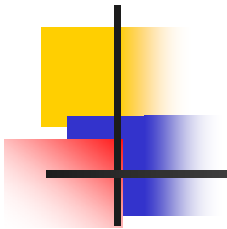
... --- Phase 2a: Conceptual Schema Design ...

- Characteristics of a data model:
  - Expressiveness
  - Simplicity and understandability
  - Minimality
  - Diagrammatic representation
  - Formality
- **Note 1**: Through out the course we have used the high level conceptual model as EER model.
- **Note 2**: UML has class diagrams that are largely based on EER model.



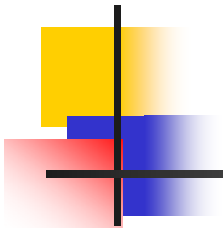
... --- Phase 2a: Conceptual Schema Design ...

- Approaches to Conceptual Schema Design
  - Centralized schema design:
    - Requirements from Phase 1 are merged into one set, then schema design begins. (merging is done by DBA).
    - If there are many users of DB and requirements are too many, then this approach would be too tedious.
  - View Integration approach
    - Requirements are not merged into one set.
    - For each user group a separate schema is designed, then all schemas are merged into one global conceptual schema for the entire DB.



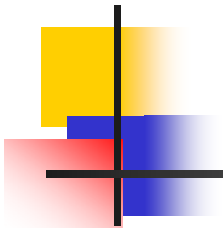
... --- Phase 2a: Conceptual Schema Design ...

- Strategies for schema design:
  - Top-down
  - Bottom-up
  - Inside-out
  - Mixed
- Schema (view) Integration
  - Identifying correspondences and conflicts among the schemas:
    - Naming conflicts, type conflicts, conflicts among constraints, and domain conflicts.
  - Modifying views to conform to one another
  - Merging of the views
  - Reconstructing



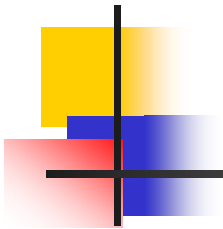
--- Phase 2b: Transaction Design

- The following should be specified in this phase:
  - Input/Output parameters
  - Functional behavior
- Transactions can be grouped in the following groups:
  - Retrieval Transactions
  - Update Transactions
  - Mixed Transactions
- Transactions or applications may originate in a front-end tool such as Power Builder or Oracle Developer.
- Transactions design as part of Software Engineering.



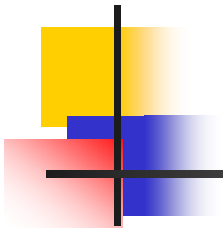
-- Phase 3: Choice of DBMS ...

- Factors affecting the Choice of DBMS
  - Technical
    - Type of DBMS (RDBMS, ORDBMS, etc)
    - Storage structures
    - User and Programmer interfaces available
    - Access paths that the DBMS supports
    - Users and programmer interfaces available
    - Client-server environment, etc
  - Cost
  - Political



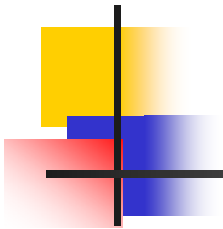
... -- Phase 3: Choice of DBMS ...

- Costs to be considered:
  - Software acquisition
  - Maintenance
  - Hardware acquisition
  - DB creation and conversion
  - Personnel
  - Training
  - Operating



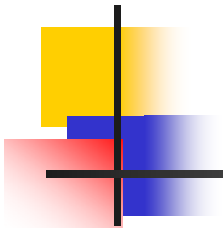
-- Phase 4: Data Modeling

- Logical DB design
  - System-independent Mapping
    - In this stage, the mapping does not consider any specific characteristics or special cases that apply to the DBMS implementation of the data model
  - Tailoring the schema to a specific DBMS
    - Different DBMS implement the data model by using specific modeling features.
    - It may be required to tailor the schema to conform to the DBMS chosen
- The result of this phase is DDL statements in the language of chosen DBMS. Many automated case tools will be of help in the task.



-- Phase 5: Physical DB Design ...

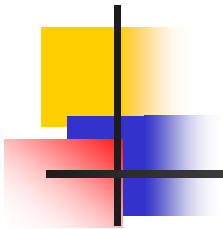
- Physical DB design is the process of choosing specific storage structures and access paths for the DB files to achieve good performance for the various DB applications.
- In this phase we mention generic guidelines which is true for any type of DBMS
  - Response time: The elapsed time between submitting a DB transaction for execution and receiving a response. Response time is influenced by:
    - DBMS
    - System load, Operating system scheduling, or communication delays



... -- Phase 5: Physical DB Design ...

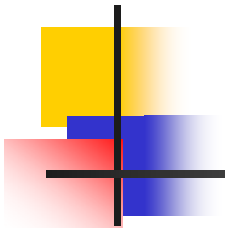
- Space utilization: The amount of storage space used by the DB files and their access path structures on disk, including indexes and other access paths.
- Transaction throughput: This is the average number of transactions that can be processed per unit time.

- Typical, average and Worst-case limits on above mentioned parameters are calculated.
- This will be considered as the specification of the system performance requirements.
- The result of the physical DB design is an initial determination of storage structures and access paths for the DB files.



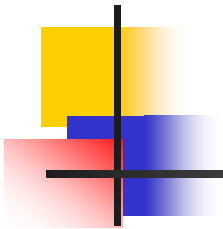
-- Phase 6: DB System Implementation and Tuning ...

- This phase is carried out by the DBA along with the help of DB designers
- DDL is used to create the empty of DB files and schemas.
- The second part of this phase is to implement the DB transactions.
- The DB is then populated with data, or the existing old data can be converted into this new system.
- With these activities this phase completes and the operation of the database application starts.



... -- Phase 6: DB System Implementation and Tuning

- As the transactions take place, the DB may need to be fine tuned. Due to large amount of DB transactions, the query results might take long time for processing, hence tuning is required here for the DB. This is done by different methods.
  - Example: adding indexes, reorganize some files, dropping some old indexes, analyzing the queries for their optimality.
- The DB tuning will continue as long as the DB is in use.



-- Summary of the Six Phases of DB design Process

Phase 1: Requirements collection and Analysis

Phase 2: Conceptual DB design

Phase 3: Choice of DBMS

Phase 4: Data model mapping

Phase 5: Physical Design

Phase 6: System implementation and tuning