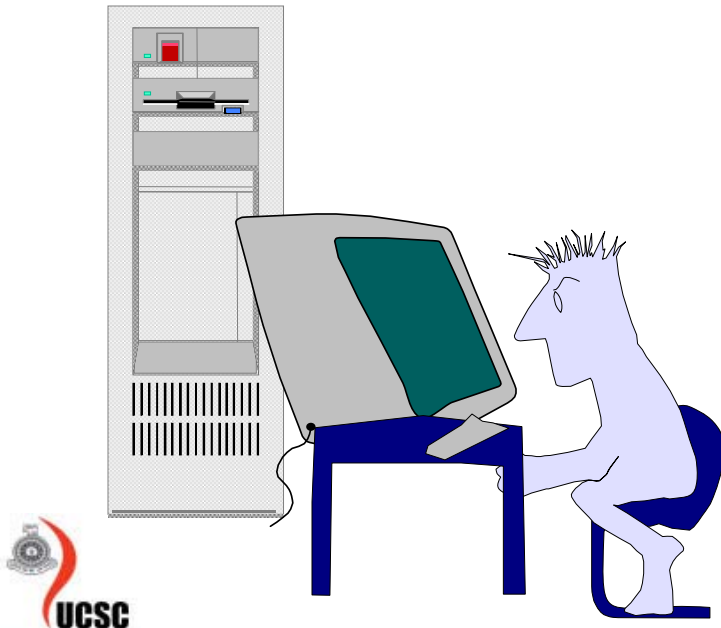


Database Design Process

Duration: 12hrs



Detailed Syllabus



6.1 Database Design Approach

- 6.1.1 Introduction: Benefits, Critical success factors, Where it fits into the application development process, Approach
- 6.1.2 Data requirement analysis: Gain an understanding of the business; Conceptual modeling: Identify the principal data objects, Diagram the data objects using the entity-relationship (ER) approach, Resolve the conceptual data model, Determine attribute specifications and data types, Verify the conceptual data model through normalization; Logical model; Physical model; Database Design tools.

6.2 ER Concepts and Terminology

- 6.2.1 The Role of ER Diagrams.
- 6.2.2 Three classes of objects: Entities, Relationships and Attributes.
- 6.2.3 Entities: Entity, Entity instance, Subtype and Super-type Entities, Strong and weak entities, Generalization, specialization and aggregation.
- 6.2.4 Relationships: Connectivity (binary, n-array), (1:1, 1:N, M:N), Determining the connectivity, Cardinality, Existence dependency (mandatory, optional).
- 6.2.5 Attributes: Identifying attributes, Attribute types (identifier, descriptor), Derived data, Domain, Composite attributes.

6.3 Mapping Conceptual model into relational schema.

6.3.1 Regular, weak, generalized and specialized entities, Relationship types, Multi-valued attributes.

6.3.2 Resolve the conceptual data model; Redundant Relationships; Recursive Relationships; Resolving Relationships: 1:1, M:N.

6.4 Attribute Specifications and Data types.

6.4.1 Attribute names, Naming conventions, Avoid Synonyms and Homonyms, Null Values, Entity integrity, Unique Requirement.

6.4.2 Categories of Data Types: Character, Numeric, Variable Character, Date, Serial, Money, Date-time, Interval.

6.4.3 Character: CHARACTER (CHAR); Numeric: INTEGER (INT), SMALLINT, FLOAT, SMALLFLOAT, DECIMAL; Variable Character: CHARACTER VARYING (VARCHAR); Binary Large Object (BLOB): Text, Byte.

- Database design and the creation of an entity relationship diagram (also known as an "ERD" or data model) is an important yet sometimes overlooked part of the application development lifecycle. An accurate and up-to-date data model can serve as an important reference tool for DBAs, developers, and other members of a JAD (joint application development) team. The process of creating a data model helps the team uncover additional questions to ask of end users. Effective database design also allows the team to develop applications that perform well from the beginning. By building quality into the project, the team reduces the overall time it takes to complete the project, which in turn reduces project development costs. The central theme behind database design is to "measure twice, cut once".

- *Data modeling is essential to building a well-functioning database. For a database to support the activities of a business, it needs a good blueprint and foundation: the data model. A data model represents a business' data.*

Effective database designers will keep in mind the principles of normalization while they design a database. Normalization is a database design approach that seeks the following four objectives:

- minimization of data redundancy,
- minimization of data restructuring,
- minimization of I/O by reduction of transaction sizes, and
- enforcement of referential integrity.

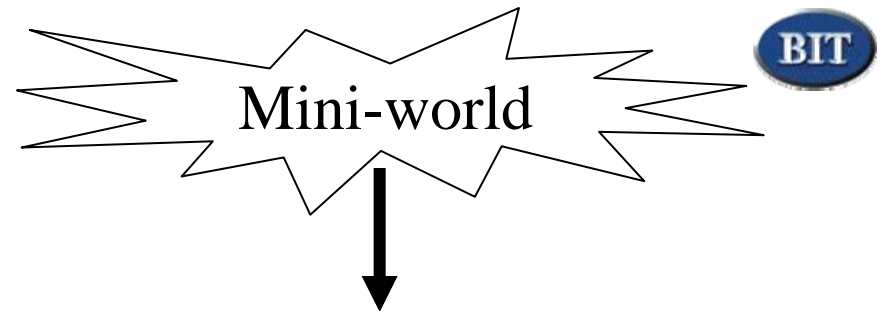
- *A poor database design can cripple an application, producing problems with redundancy, inaccuracy, consistency, and concurrency of your data.*
- *Normalization is a process that serves to reduce, if not eliminate, these problems with data.*

Database Design

The database design process can be broken down into four phases.

- Phase 1 - Requirements Collection and analysis phase
- Phase 2 - Conceptual Design
- Phase 3 - Logical Design
- Phase 4 - Physical Design

Database Design...



Phase 1 - Requirements Collection and Analysis phase

Functional
Requirements



Database
Requirements



Prospective database uses are interviewed to understand and document their data requirements.

From data view (e.g. ERD) rather than functional view (e.g. DFD)



Database Design...

Database
Requirements



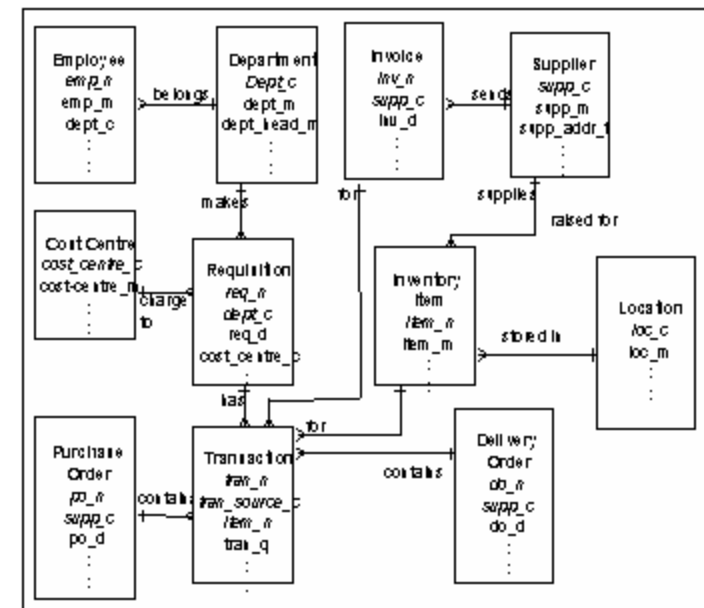
Phase 2 - Conceptual Design



Conceptual Design

This is high level description of the structure of a database. E.g. E-R diagram

Concise description of the data requirements of the users and includes detailed descriptions of the data, relationships and constraints.



Database Design...

Conceptual Design



Phase 3 - Logical Design

This is the process of mapping the database structure developed in the previous phase to a particular database model.
E.g. map E-R model to relational

Logical Design



| | | | | |
|-------|---------|-------------|------|-------|
| ATTR1 | ATTR2 A | TTTR3 ATTR4 | ATTR | LLL 5 |
|-------|---------|-------------|------|-------|

Table

| | | | |
|-------|----------|------|-------|
| ATTR1 | ATTR2 AT | TTR3 | ATTR4 |
|-------|----------|------|-------|

Table

| | | |
|-------|-----------|---------|
| ATTR1 | AT A TT R | 4 ATTR5 |
|-------|-----------|---------|

Table

| | | | |
|-------|---------|-------|---------|
| ATTR1 | ATTR2 A | TTTR3 | ATTRTR5 |
|-------|---------|-------|---------|

Table

Specific to a database model,
but independent of a particular
DBMS (product)

Database Design...



Logical Design



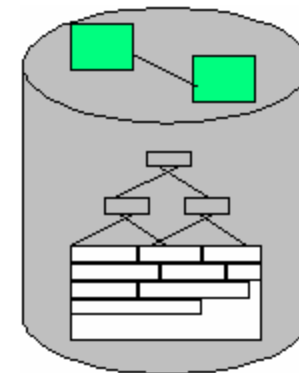
Phase 4 - Physical Design



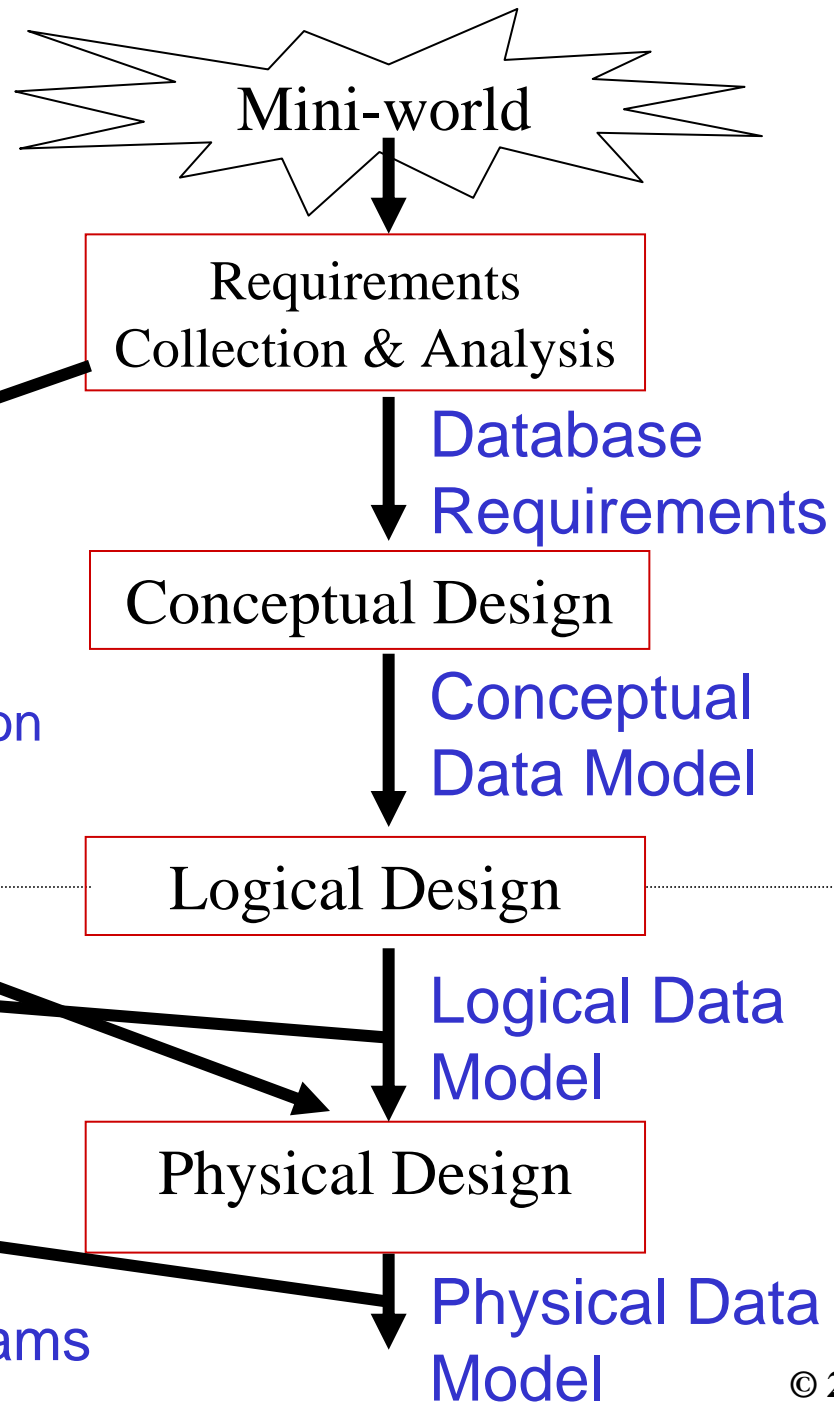
Physical Design

This is the process of defining structure that enables the database to be queried in an efficient manner.

E.g. index and hash file design, data partition

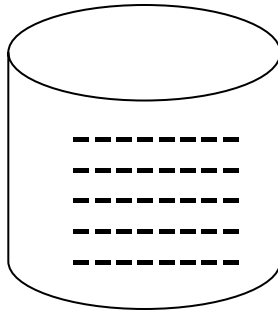


Phases of Database Design



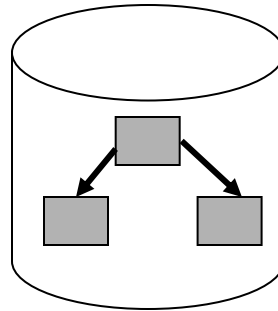
Types of Database Models

Traditional
Files



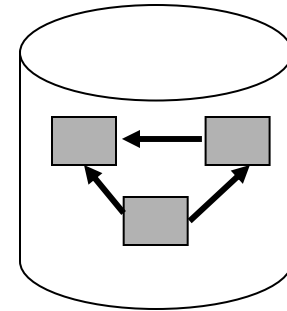
60s

Hierarchical
Database Model



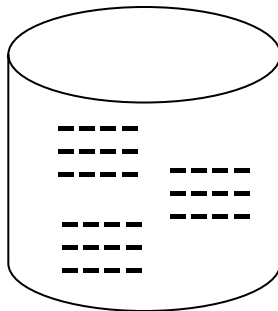
70s

Network
Database Model



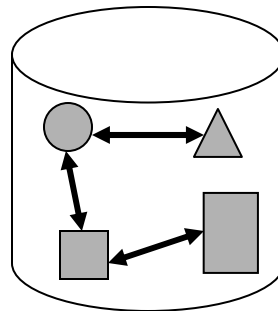
70s

Relational
Database Model



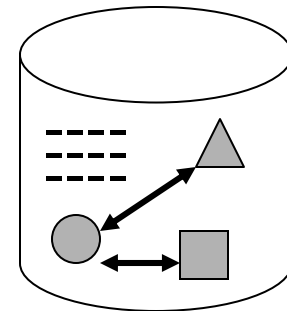
80s

Object-oriented
Database Model



90s

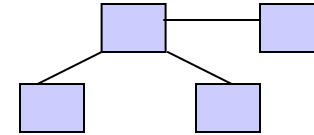
Object-relational
Database Model



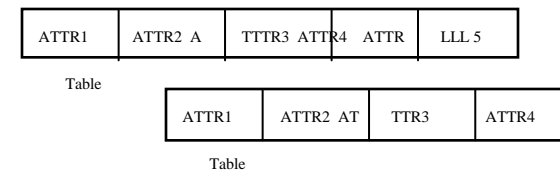
90s

Types of Data Models

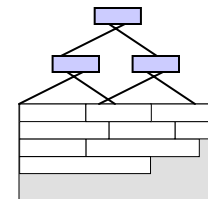
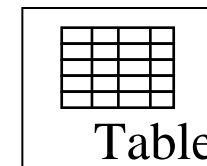
- **Conceptual Data Model**



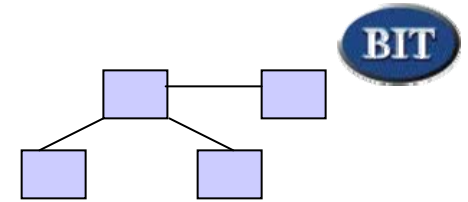
- **Logical Data Model**



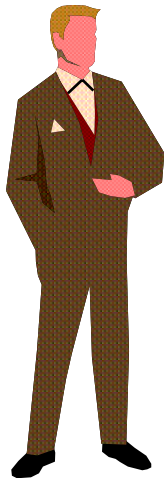
- **Physical Data Model**




Conceptual Data Model



- A data model representing the objects and business rules that govern the operation of an organisation
 - *Done by a Business Analyst*
 - *Not constrained by access requirement and technology*



Logical Data Model



| | | | | |
|-------|---------|-------------|------|-------|
| ATTR1 | ATTR2 A | TTTR3 ATTR4 | ATTR | LLL 5 |
|-------|---------|-------------|------|-------|

Table

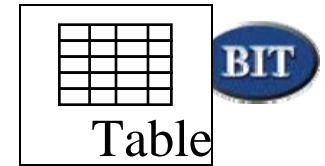
employee(empno, ...)

- A set of data structures assembled following rules that describe the processing requirements (access paths) of the data in terms of a logical database model
 - *Done by a Data Analyst*
 - *Not constrained by technology (?)*



Physical Data Model

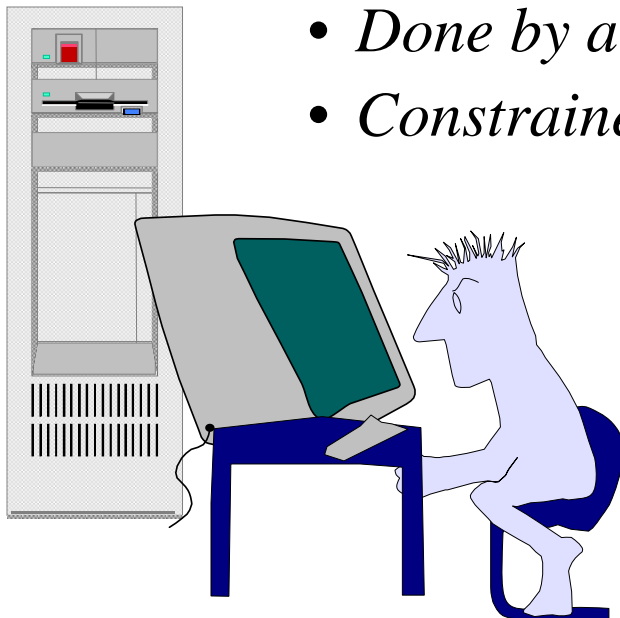
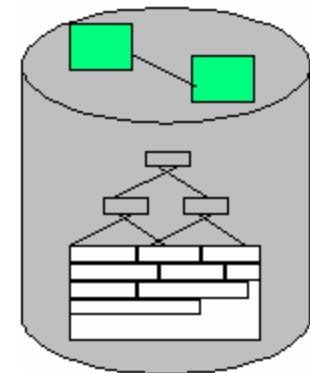
create table



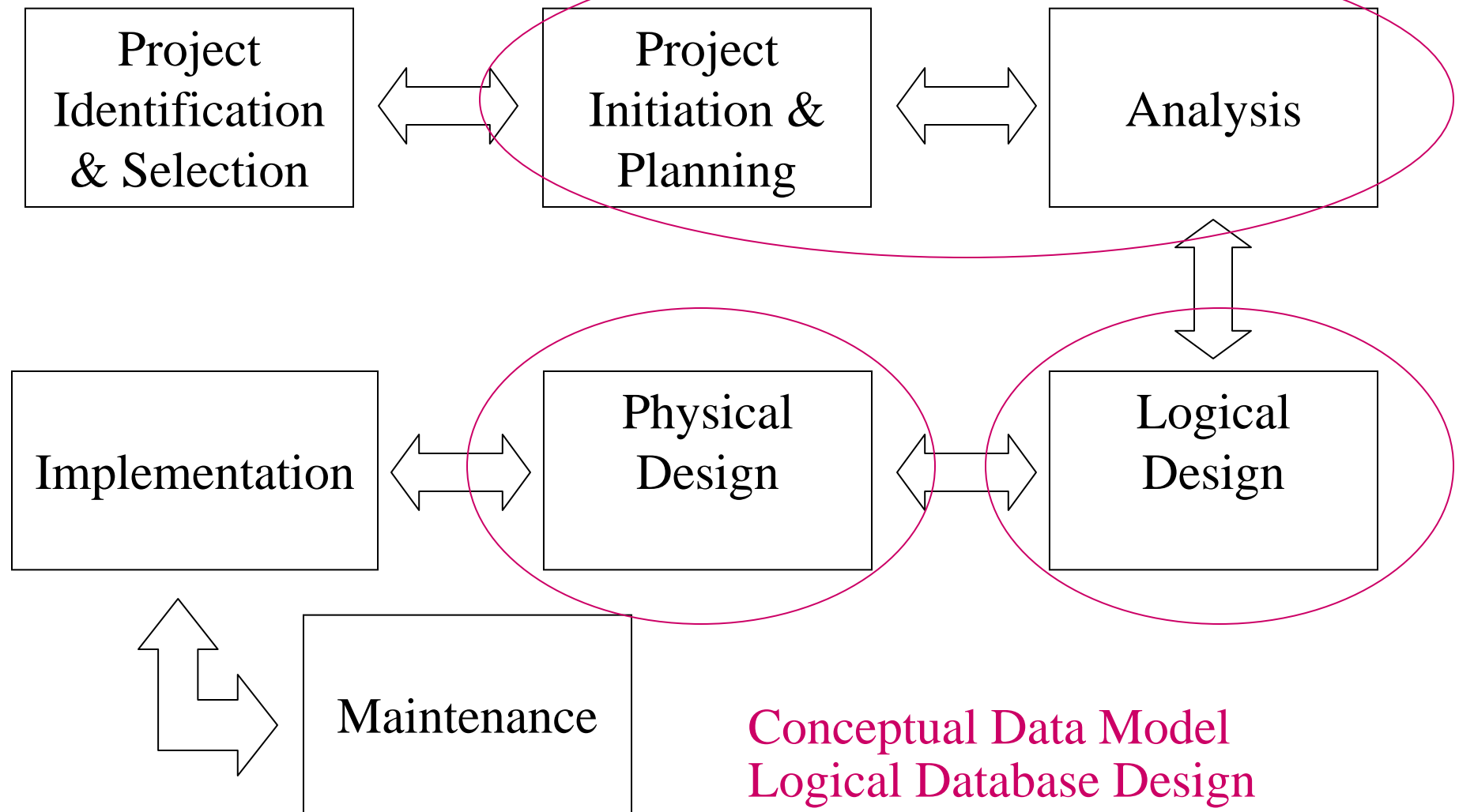
employee(empno ...)

- A model prepared for the purpose of implementing a database that runs under the control of a particular DBMS (product)

- *Done by a DBA*
- *Constrained by Technology*



Systems Development Life Cycle (SDLC)



Conceptual Data Model
Logical Database Design
Physical Database Design

Database Development Activities

- Enterprise Modelling
- Conceptual Data Modelling
- Logical Database Design
- Physical Database Design and Creation
- Database Implementation
- Database Maintenance

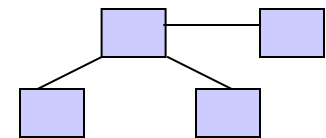
Enterprise Modelling

- Analyse current data processing
- Analyse the general business functions and their database needs
- Justify need for new data and databases in support of business

Project Identification & Selection

Conceptual Data Modelling

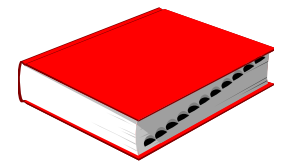
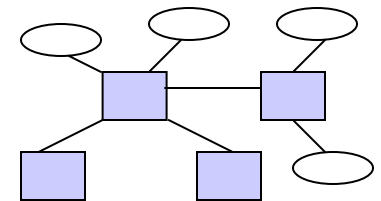
- Identify scope of database requirements for proposed information system
- Analyse overall data requirements for business function(s) supported by database
- Develop preliminary conceptual data model including entities and relationships
- Compare preliminary conceptual data model with enterprise data model



Analysis

Conceptual Data Modelling...

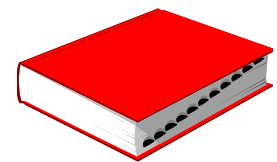
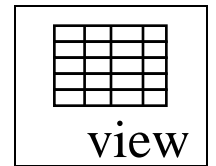
- Develop detailed conceptual data model, including all entities, relationships, attributes and business rules
- Make conceptual data model consistent with other models of information system
- Populate repository with all conceptual database specifications



Analysis

Logical Database Design

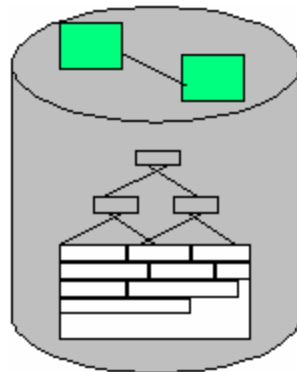
- Analyse in detail the transactions, forms, displays and inquires (data views) required by the business functions supported by the database
- Integrate database views into conceptual data model
- Identify data integrity and security requirements, and populate repository



Logical Design

Physical Database Design

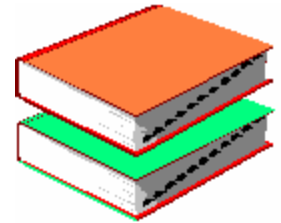
- Define database to DBMS (often generated from repository)
- Decide on physical organisation of data
- Design database processing programs



Physical Design

Database Implementation

- Code and test database processing programs
- Complete database documentation and training materials
- Install database and convert data from prior systems



Implementation

Database Maintenance

- Analyse database and database applications to ensure that evolving information requirements are met
- Tune database for improved performance
- Fix errors in database and database applications and recover database when it is contaminated

Maintenance