Database Management Systems

ICT1213

Database Development Life Cycle

Nuwan Laksiri
Department of ICT
Faculty of Technology
University of Ruhuna

Lecture 3

What we discuss Today………

- Desirable properties of a database
- Characteristics of Bad Database
- General Model of System development
- Database Development Life Cycle

Objectives

- Describe the key points of the waterfall model applied to database development
- Appreciate the roles of various development artefacts, such as the data requirements document, conceptual data model and such like used to communicate between activities in the database development life cycle
- Communicate effectively about aspects of the development of databases

Desirable properties of a database

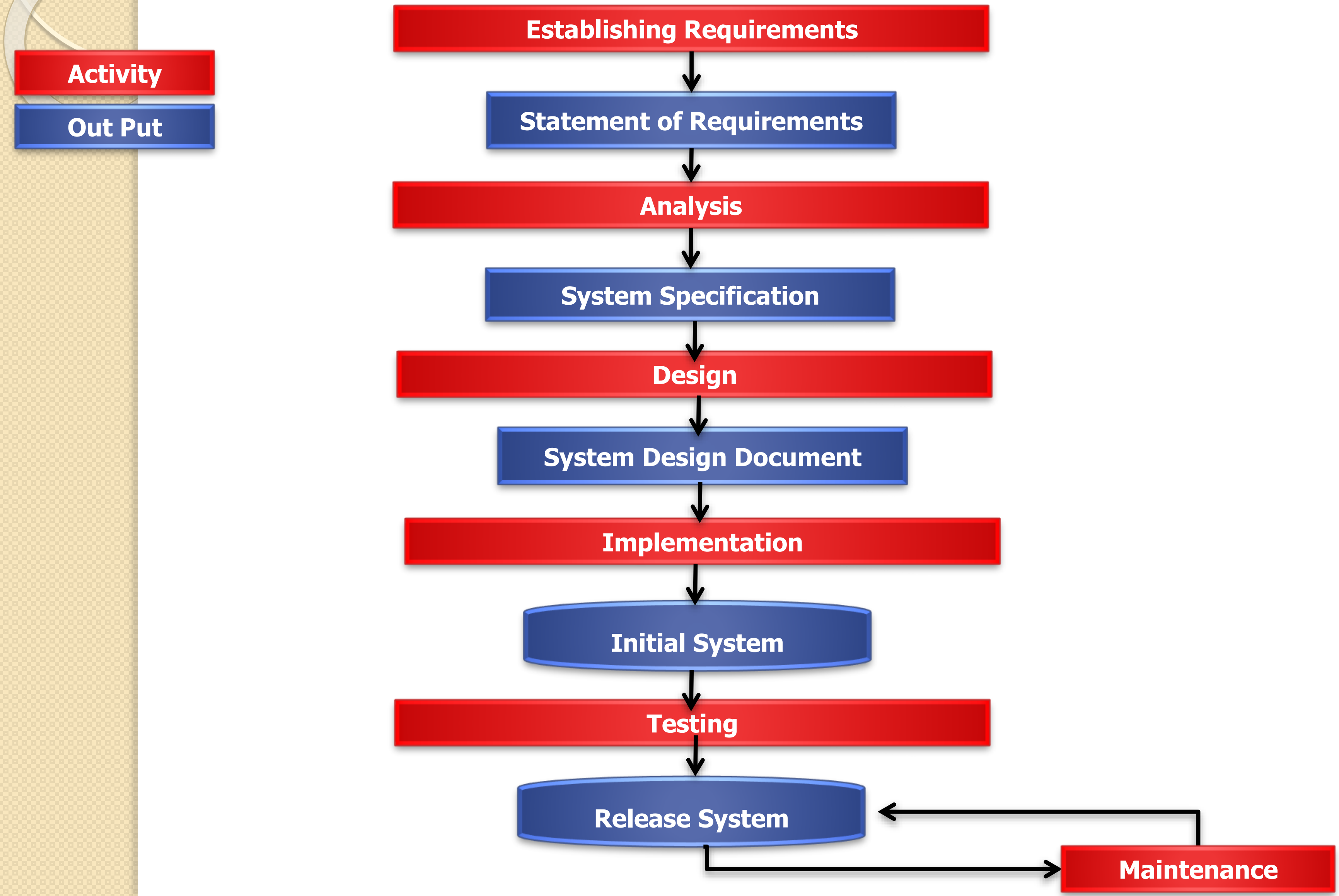
Completeness	Ensures that users can access the data they want. Note that this includes ad hoc queries, which would not be explicitly given as part of a statement of data requirements.
Integrity	Ensures that data is both consistent (no contradictory data) and correct (no invalid data), and ensures that users trust the database.
Flexibility	Ensures that a database can evolve (without requiring excessive effort) to satisfy changing user requirements.
Efficiency	Ensures that users do not have unduly long response times when accessing data.
Usability (ease of use)	Ensures that data can be accessed and manipulated in ways which match user requirements.

Characteristics of Bad Database

- Not satisfy all user requirements
- Would contain inconsistent and invalid data
- Would require excessive effort to change
- Would be slow and clumsy to use for achieving a desired outcome.

When trying to decide if some choices are better than others, you need to consider the key desirable properties of a database.

General Model of System Development



Activity

Out Put

Establishing Requirements

Statement of Requirements

Analysis

System Specification

Design

System Design Document

Implementation

Initial System

Testing

Release System

Maintenance

General Model of System Development

- **Establishing requirements**
  - Involves consultation with, and agreement among, stakeholders as to what they want of a system
  - Expressed as a statement of requirements.

- **Analysis**
  - Starts by considering the statement of requirements and finishes by producing a system specification.
  - The specification is a formal representation of what a system should do, expressed in terms that are independent of how it may be released.

- **Design**
  - Begins with a system specification and produces design documents,
  - Provides a detailed description of how a system should be constructed.

General Model of System Development

- **Implementation**
  - Construction of a computer system according to a given design document and taking account of the environment in which the system will be operating

    (for example specific hardware or software available for the development).
  - Implementation may be staged, usually with an initial system requirements can be validated and tested before a final system is released for use

- **Testing**
  - Compares the implemented system against the design documents and requirements specification
  - produces an acceptance report or, more usually, a list of errors and bugs that require a review of the analysis, design and implementation processes to correct

    (testing is usually the task that leads to the waterfall model iterating through the life cycle)

General Model of System Development

- **Maintenance**
  - Involves dealing with changes in the requirements,

    or the implementation environment,

    bug fixing or porting of the system to new environments

    (for example migrating a system from a standalone PC to a UNIX

    workstation or a networked environment)
  - Since maintenance involves the analysis of the changes required, design of a solution, implementation and testing of that solution over the lifetime of a maintained software system, the waterfall life cycle will be repeatedly revisited.

Exercise

The following are problems that have been identified during the testing process in the development of a new system. In which part of the life cycle do you think these problem could have originated and been identified by a thorough review following that stage in the development life cycle?

1. The performance of the system is poor – failing to respond quickly enough to meet the stated user requirement of interactive, screen-based use.
2. No backup facilities were included to meet the users' requirement of long-term archival of their data.
3. No user manuals were provided!

Answers

1. The user requirement for interactive, screen-based use forms part of the system specification, but actual performance can only be identified at implementation time, when the system has been built and is being evaluated. In the development of some systems, attempts at performance prediction can occur during the design stage; however, with database development it is normal to validate such design predictions at the testing stage.
2. The missing feature was in the requirements, so during analysis, design or implementation someone has overlooked this requirement. Without further information it is impossible to say when the feature fell out of the development life cycle.
3. No user manuals – were they asked for as part of the requirements? If they were then see the answer to (2). If they weren't part of the requirements, then such a basic omission should have been identified at the requirements gathering – user documentation should be considered as a standard requirement for any new system.

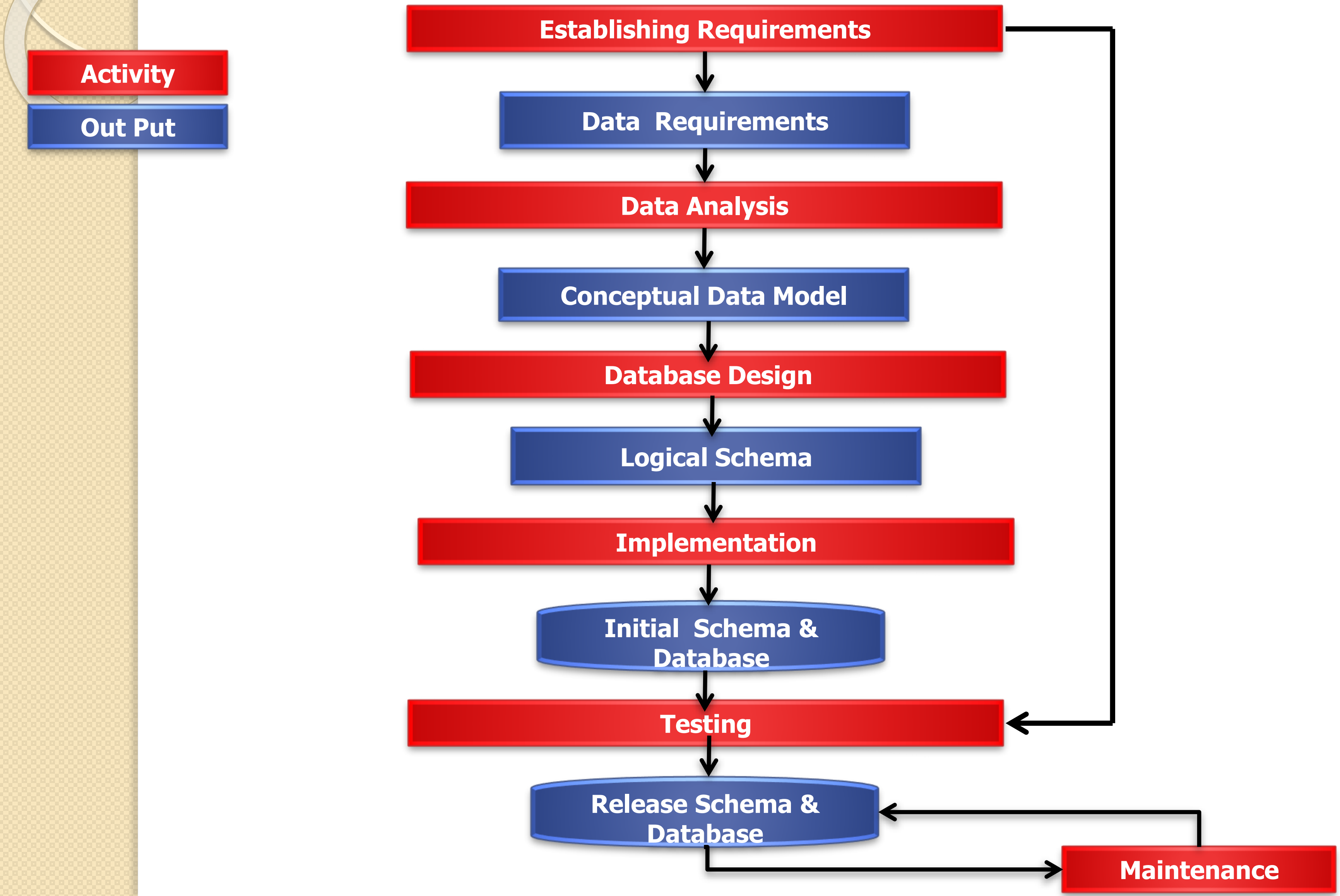
The Mapping

- we can use above discussed framework as the basis for a model of database development which incorporates following three assumptions:
  - We can separate the development of a database from the user processes that make use of the database

    (specification and creation of a schema to define data in

    a database)
  - We can use the three-schema architecture as a basis for distinguishing the activities associated with a schema.
  - We can represent the constraints to enforce the semantics of the data once, within a database, rather than within every user process that uses the data.

Database Development Life Cycle



Activity

Out Put

Establishing Requirements

Data Requirements

Data Analysis

Conceptual Data Model

Database Design

Logical Schema

Implementation

Initial Schema & Database

Testing

Release Schema & Database

Maintenance

Gather Requirements

- Concerned only with the requirements that relate specifically to the data
- Consultation with, and agreement among, all the users.
- Identifying meaning and interpretation of the data elements.
- Data administrator plays a key role in this process as they overview the business, legal and ethical issues within the organization that impact on the data requirements.

Data Requirement Document

Says nothing about how the data are processed, but it does state what the data items are, what attributes they have, what constraints apply and the relationships that hold between the data items.

- Used to agree requirements with users.
- Easily understood
- Should not be overly formal or highly encoded.
- document should give a concise summary of all users' requirements – not just a collection of individuals' requirements

Example DRD

- Each course which is available for study is given a course code, a title and a value for credit points – either a 2 credit course or a 3 credit course.

- A course may have a quota – the maximum number of students that can be enrolled on the course in any one year; each quota is reviewed regularly and the last date of review is recorded with the quota.

- A course need not (yet) have any students enrolled on it. Students may not enroll for more than 20 credit points' worth of courses at any one time.

Analysis

- Begins with the statement of data requirements and then produces a **conceptual data model**
- Obtain a detailed description of the data that will suit user requirements so that both high and low level properties of data and their use are dealt with.

Conceptual Data Model

- A conceptual data model is concerned with the meaning and structure of data, but not with the details affecting how they are implemented.
- Focuses on ‘What is required?’ not ‘How is it achieved?’
- Used by people for the database design activity; it is not used by any DBMS, nor is it a programming language.

Exercise

For each of the following statements decide which processes – requirements gathering or data analysis – would generate the statement as part of the documented output.

1. A customer record will allow for the storage of a name, address, evening and daytime phone numbers, one mobile phone number and as many email addresses as the customer wants to include.
2. We need to relate customer orders to their credit card details. If the credit card is invalid we need to know before any orders are accepted.
3. An order must have the opportunity to include a delivery address that is different from the customer's credit card billing address.

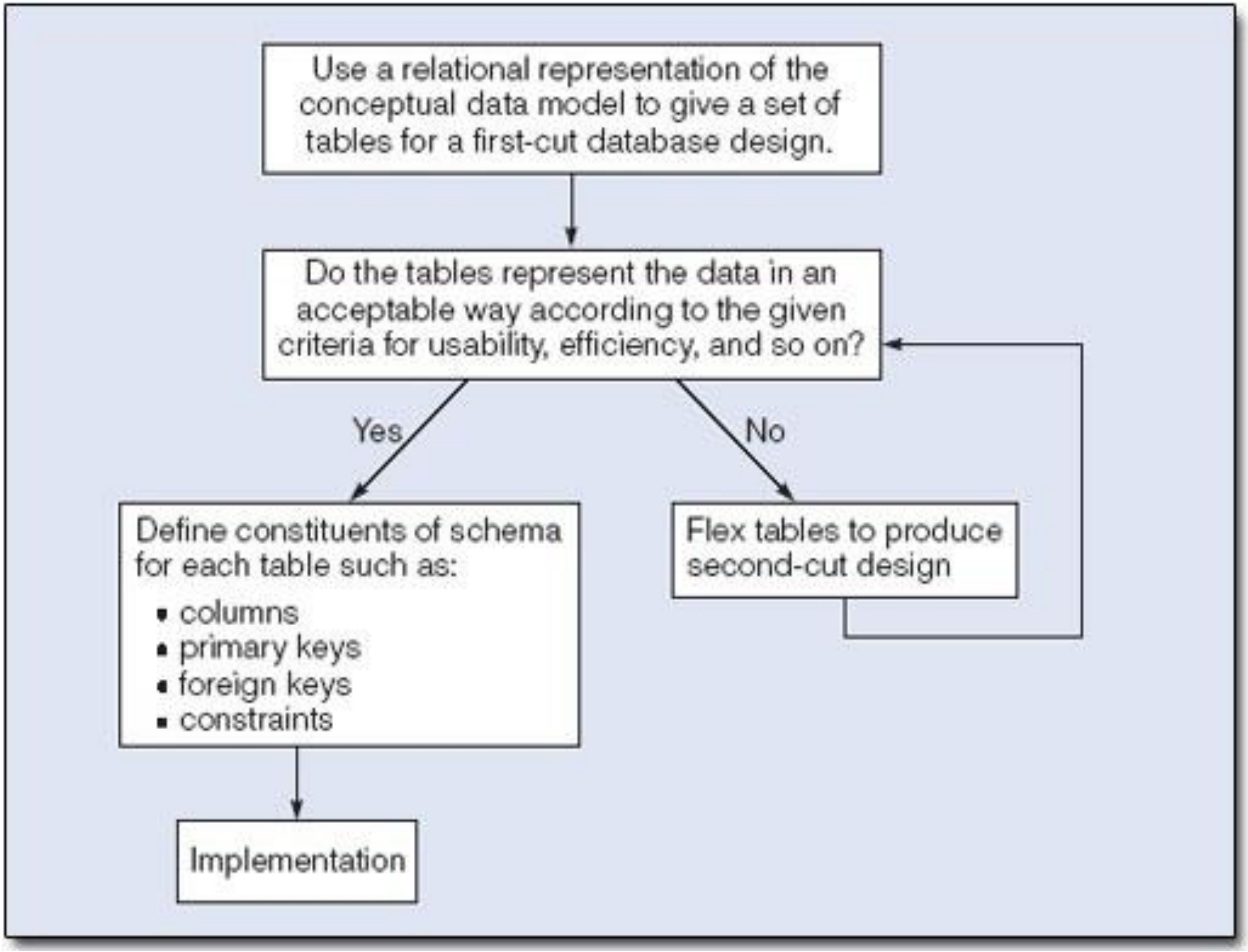
Answers

1. This is quite a detailed description of what data will be recorded about a customer so it is likely to be data analysis output – it might be included in the requirements analysis (a lot of information about data items usually is), but it will form part of the more formal conceptual data model.
2. This statement is about the requirements for recording a valid order, so it is part of the requirements specification. In effect, it is saying that an order can be recorded only if the credit card details are valid. If this appeared in the data analysis output we would need to include answers to several more questions: What is a valid credit card? What is an order? Can a customer use more than one credit card per order, or none? And so on.
3. This is another requirements specification statement – it says nothing about the data requirements in any detail. As a result, a lot more questions would need to be asked about the statement and what it references before the data analyst would be able to document this statement.

Design

- Starts with a conceptual data model and produces a specification of a logical schema
- Determine the specific type of database system (network, relational, object-oriented etc)
- **Logical schema** :
  - of all the tables and constraints needed to satisfy the description of the data in the conceptual data model.
- Choices are made as to which tables are most appropriate for representing the data in a database
- It is not uncommon to find that a single design cannot simultaneously satisfy all the properties of a good database.
- So it is important that the designer has prioritized these properties (usually using information from the requirements specification),
  - for example, to decide if integrity is more important than efficiency and whether usability is more important than flexibility in a given development.

Steps for database design


Use a relational representation of the conceptual data model to give a set of tables for a first-cut database design.
Do the tables represent the data in an acceptable way according to the given criteria for usability, efficiency, and so on?
Yes
No
Define constituents of schema for each table such as:
• columns
• primary keys
• foreign keys
• constraints
Flex tables to produce second-cut design
Implementation

3 general points of Design

For a given conceptual data model it is not necessary that all the user requirements it represents have to be satisfied by a single database.

separate the development of a database from the development of user processes that make use of it.

many aspects of database design and implementation depend on the particular DBMS being used.

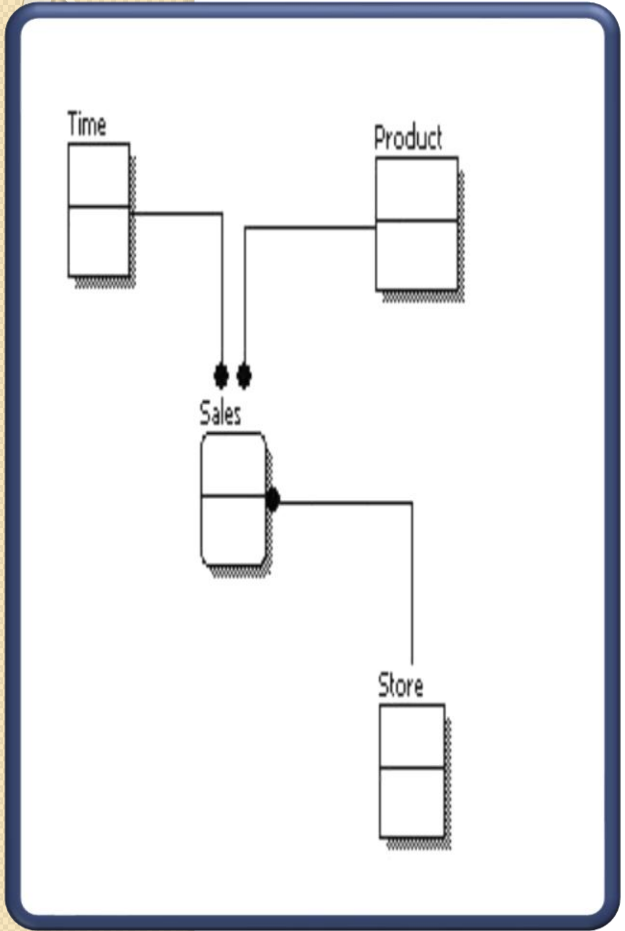
Implementation

- Construction of a database according to the specification of a logical schema.
- Include the specification of :-
  1. storage schema
  2. security enforcement
  3. external schema
- Heavily influenced by the choice of available DBMS, database tools and operating environment
- Requires a very detailed knowledge of the specific features and facilities that the DBMS has to offer

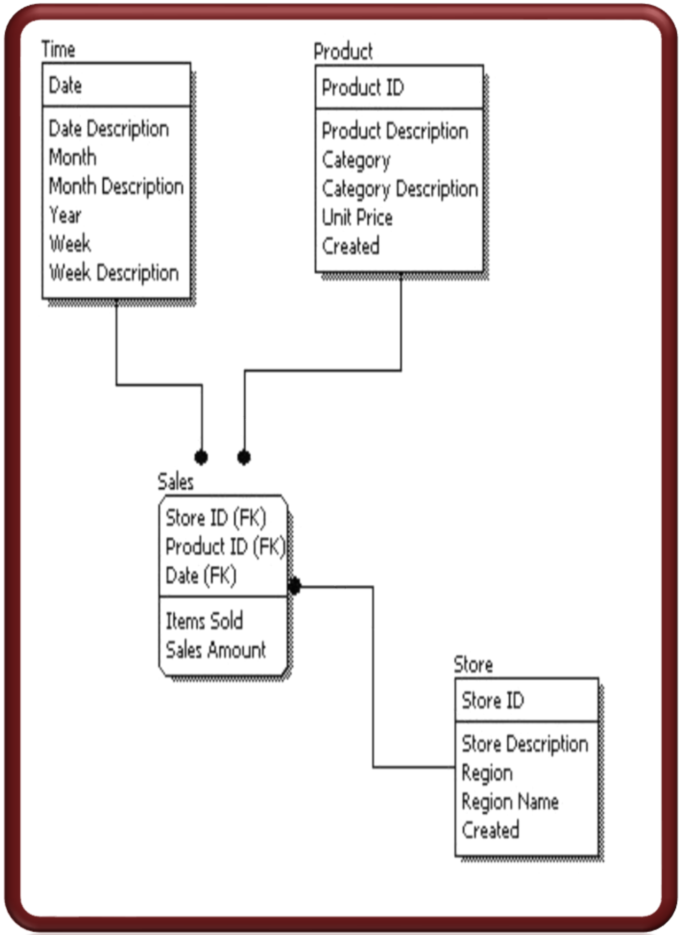
Implementation Activities

- Releasing the design
- Populating the database
- Supporting users and user processes
- Supporting data management strategies

Conceptual ,Logical ,Physical Schema



Conceptual


Time

- Date
- Date Description
- Month
- Month Description
- Year
- Week
- Week Description

Product

- Product ID
- Product Description
- Category
- Category Description
- Unit Price
- Created

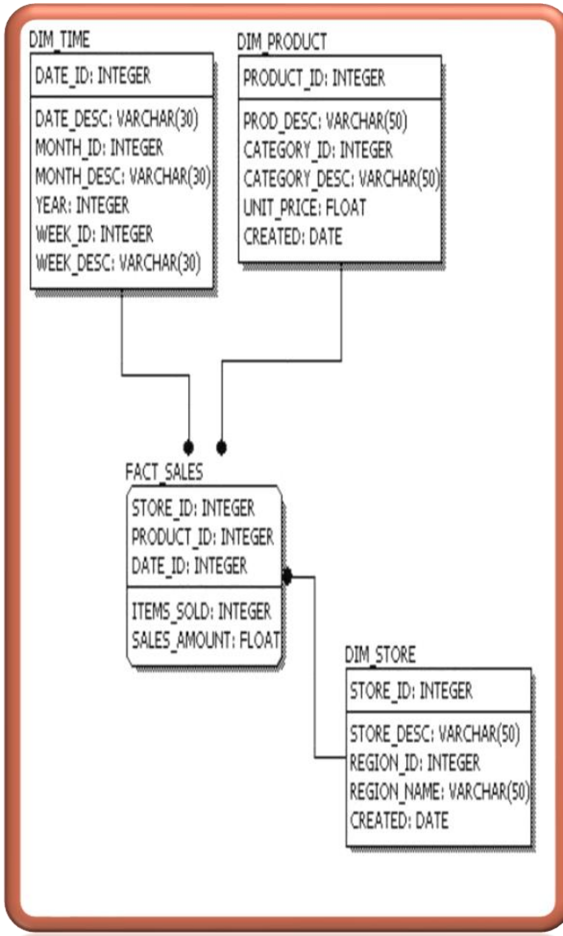
Sales

- Store ID (FK)
- Product ID (FK)
- Date (FK)
- Items Sold
- Sales Amount

Store

- Store ID
- Store Description
- Region
- Region Name
- Created

Logical


DIM_TIME

- DATE_ID: INTEGER
- DATE_DESC: VARCHAR(30)
- MONTH_ID: INTEGER
- MONTH_DESC: VARCHAR(30)
- YEAR: INTEGER
- WEEK_ID: INTEGER
- WEEK_DESC: VARCHAR(30)

DIM_PRODUCT

- PRODUCT_ID: INTEGER
- PROD_DESC: VARCHAR(50)
- CATEGORY_ID: INTEGER
- CATEGORY_DESC: VARCHAR(50)
- UNIT_PRICE: FLOAT
- CREATED: DATE

FACT_SALES

- STORE_ID: INTEGER
- PRODUCT_ID: INTEGER
- DATE_ID: INTEGER
- ITEMS_SOLD: INTEGER
- SALES_AMOUNT: FLOAT

DIM_STORE

- STORE_ID: INTEGER
- STORE_DESC: VARCHAR(50)
- REGION_ID: INTEGER
- REGION_NAME: VARCHAR(50)
- CREATED: DATE

Physical

Conceptual ,Logical ,Physical Schema

Conceptual

- A conceptual entity-relationship model shows how the business world sees information.
- It suppresses non-critical details in order to emphasize business rules and user objects.
- It typically includes only significant **entities** which have business meaning, along with their relationships.
- Many-to-many relationships are acceptable to represent entity associations.

Conceptual ,Logical ,Physical Schema

Logical

- A logical entity-relationship model is provable in the mathematics of data science.
- Given the current predominance of relational databases, logical models generally conform to relational theory. Thus a logical model contains only **fully normalized entities**.
- Some of these may represent logical domains rather than potential physical tables.

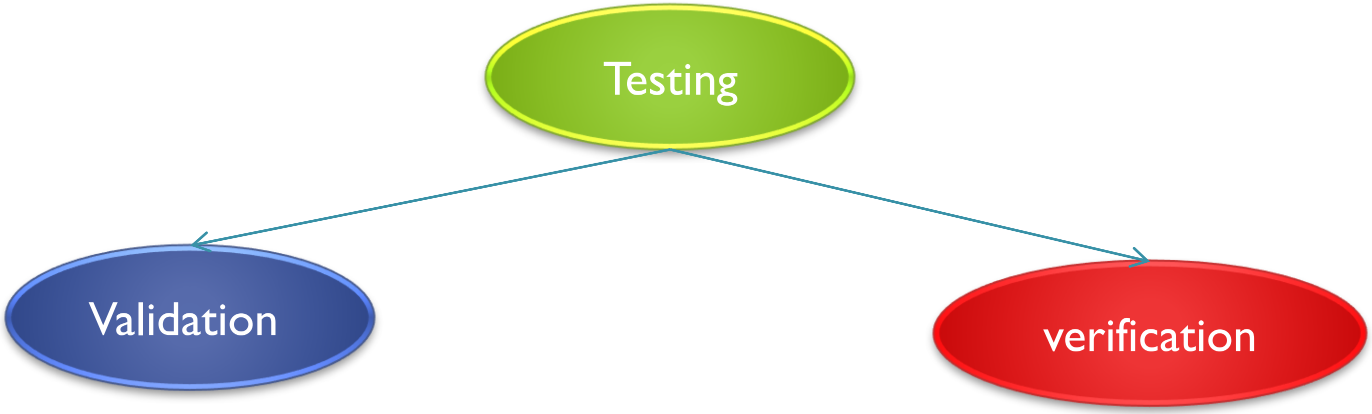
Conceptual ,Logical ,Physical Schema

Physical

- A physical data model is a single logical model instantiated in a specific database management product (e.g., Sybase, Oracle, Informix, etc.) in a specific installation.
- The physical data model specifies implementation details which may be features of a particular product or version, as well as configuration choices for that database instance.

Testing

Uncover errors in the design and implementation of the database, its structure, constraints and associated user and management support.



Testing

Validation

verification

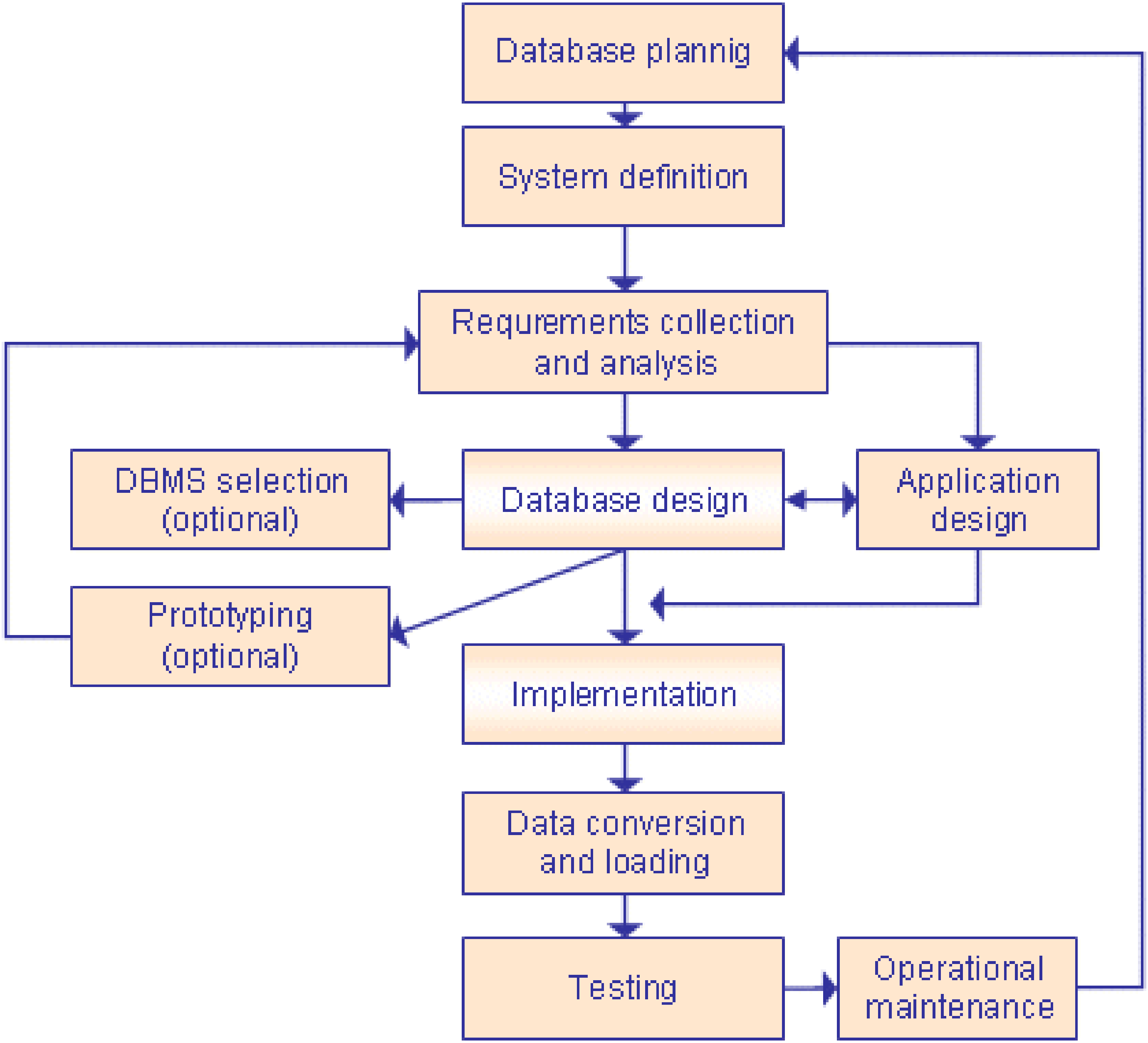
has the right database been developed to meet the requirements? It attempts to confirm that the right database has been constructed, with the right characteristics to meet the specified requirements.

has the database design been implemented correctly? Verification ensures that the processing steps, constraints and other ‘programmed’ components of the database (security, backup, recovery, audit trails, etc.) have been correctly implemented and contain no errors in program logic or execution sequences.

Maintenance

Operational maintenance	where the performance of the database is monitored.
Porting and implementation	In which the DBMS, the user processes, the underlying computer system or some other aspect undergoes changes that require the database implementation to be revised.
Requirements change	where the original requirement specification changes

A Modern Approach



Summary

- Desirable properties of a database
  - **Completeness, Integrity, Flexibility, Efficiency, Usability (ease of use)**
- General Model of System development
  - By referring waterfall model
- Model for database development
  - Requirements gathering
  - Analysis
  - Design
  - Implementation
    - Releasing the design
    - Populating the database
    - Supporting users and user processes
    - Supporting data management strategies
  - Testing
  - Maintenance

Refferance

- *Fundamentals of Database Systems*

(6th Edition) By Remez Elmasri & Shamkant B. Navathe

Questions ???

Thank You