Computer Science 202

Database Systems:
File systems and Database Structures

Objectives

- To develop the main database system concepts
- To show the evolution of database systems from computer file systems
- To show that database design is a crucial first step in the development and proper use of database systems
- To understand the main DBMS functions
- To describe the database environment
- To introduce the student to the basics of different types of database systems

Data vs. Information

- Data is the building blocks of information
- Information is produced by processing data
- Information makes meaningful use of data
- Information is used for decision making
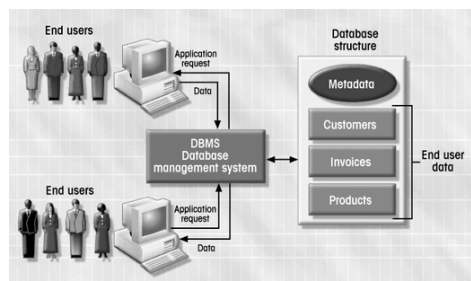- Good decision making is important for an organisation

What is a Database?

- Database is a repository which holds data items
  - User data
  - Meta data
- Allows for data to be stored in an orderly manner
- Database Management System (DBMS)
  - Manages data structures
  - Controls Access
  - Data Query language

DBMS roles

- More efficient management of data
- Query language allows for ad-hoc queries
- Reduces likelihood of inconsistent data
- Provides better Access to more and better managed data than historical methods

DBMS Interaction

Database Origins

- First 'applications' were manual clerical tasks
- Were difficult to fulfil requests for summary information , rather than just providing data
- File systems were developed to address the requirements:
  - Organised layout of data based on expected usage
  - Manual file systems were computerised
  - Creation of the job of a Data processing specialist

File System Terminology

- Data
  - The raw facts
- Field
  - A group of characters/bytes with a specific meaning
- Record
  - Logically related fields, that describe an object/entity
- File
  - A collection of related records

File Systems

- File System Data Management
  - Requires extensive complex programming in a 3GL
  - Time consuming and labour intensive
  - Ad-Hoc queries are difficult to run
  - Management systems produce a set of canned queries or reports
  - Difficult to integrate – leads to islands of information

File Systems

- Data Independence
  - A change in a file's data characteristics require rebuild of applications
  - Access programs have to be explicitly coded as to how to perform operations
  - Cumbersome to use form both the programming and management perspectives
- Structural dependence
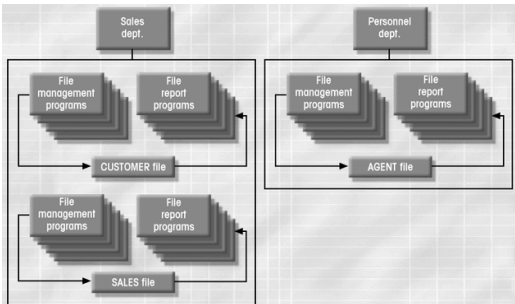 - A chance in the underlying file structure requires all related applications to be changed

File Systems

- Field Definitions and Naming Conventions
  - Flexible record definition anticipates reporting requirements
  - Selection of proper field names important
  - Attention to length of field names
  - Use of unique record identifiers

File Systems

- Data Redundancy
  - Different and conflicting versions of same data
  - Results of uncontrolled data redundancy
    - Data anomalies
      - Modification
      - Insertion
      - Deletion
    - Data inconsistency
      - Lack of data integrity
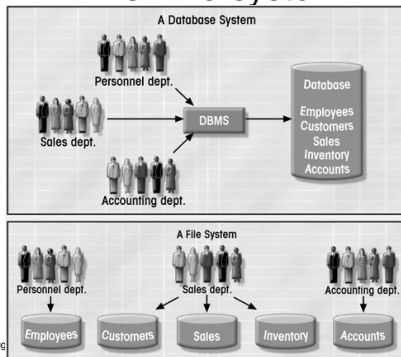
File Systems Example

Database Systems

- Database consists of logically related data stored in a single repository
- Provides advantages over file system management approach
  - Eliminates inconsistency, data anomalies, data dependency, and structural dependency problems
  - Stores data structures, relationships, and access paths

DB vs. File system

Database System Types

- Single-user vs. Multi-user Database
  - Desktop
  - Workgroup
  - Enterprise
- Centralised vs. Distributed
- Database Usage
  - Production or transactional
  - Decision support or data warehouse

DBMS Functions

- Data dictionary management
- Data storage management
- Data transformation and presentation
- Security management
- Multi-user access control
- Backup and recovery management
- Data integrity management
- Database language and application programming interfaces
- Database communication interfaces

Database Models

- Collection of logical constructs used to represent data structure and relationships within the database
  - Conceptual models: logical nature of data representation
  - Implementation models: emphasis on how the data are represented in the database
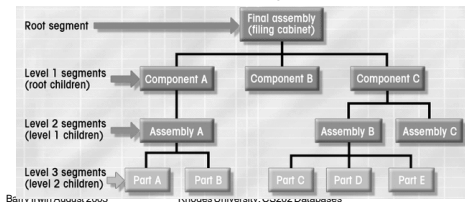
Database Models

- Collection of logical constructs used to represent data structure and relationships within the database
  - Conceptual Relationships in Conceptual Models
  - One-to-one (1:1)
  - One-to-many (1:M)
  - Many-to-many (M:N)
- Implementation Database Models
  - Hierarchical
  - Network
  - Relational
  - models: logical nature of data representation
  - Implementation models: emphasis on how the data are represented in the database

Database Structural Models

- Five main database models
  - Hierarchical
  - Network
  - Relational
  - Entity Relation
  - Object Orientated

Hierarchical

- Logically based on an inverted Tree
  - Consists of nodes with relationships
  - Each node can only have one parent
  - Each node can have multiple children

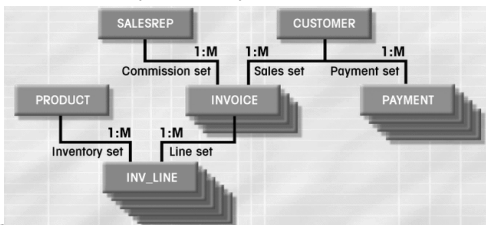|		
Root segment	Final assembly (filing cabinet)	
Level 1 segments (root children)	Component A	Component B	Component C
Level 2 segments (level 1 children)	Assembly A		Assembly B	Assembly C
Level 3 segments (level 2 children)	Part A	Part B	Part C	Part D	Part E

Hierarchical

- Advantages
  - Conceptual simplicity
  - Database security and integrity
  - Data independence
  - Efficiency
- Disadvantages
  - Complex implementation
  - Difficult to manage and lack of standards
  - Lacks structural independence
  - Applications programming and use complexity
  - Implementation limitations

Network

- Based on the concepts of Sets (Relations)
  - Set consists of an Owner and Member records
  - Member may have many owners

SALESREP		CUSTOMER	
	1:M	1:M	1:M
	Commission set	Sales set	Payment set
PRODUCT	INVOICE		PAYMENT
	1:M	1:M	
	Inventory set	Line set	
	INV_LINE		

Network

- Advantages
  - Conceptual simplicity
  - Handles more relationship types
  - Data access flexibility
  - Promotes database integrity
  - Data independence
  - Conformance to standards
- Disadvantages
  - System complexity
  - Lack of structural independence

Relational

- User perception is of data stored in a collection of tables
  - Tables are a collection of row/column intersections
  - Tables are inter-related through the sharing of a common entity attribute

Table name: AGENT

AGENT_CODE	AGENT_LNAME	AGENT_FNAME	AGENT_INITIAL	AGENT_AREACODE	AGENT_PHONE
501	Alby	Alex	B	713	228-1249
502	Hahn	Leah	F	615	882-1244
503	Okon	John	T	615	123-5589

Link through AGENT code

Table name: CUSTOMER

CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_AREACODE	CUS_PHONE	CUS_RENEW_DATE	AGENT_CODE
10010	Ramas	Alfred	A	615	844-2573	05-Apr-2002	502
10011	Dunne	Leona	K	713	894-1238	16-Jun-2002	501
10012	Smith	Kathy	W	615	894-2285	29-Jan-2001	502
10013	Olowski	Paul	F	615	894-2180	14-Oct-2002	502

Relational

- Advantages
  - Structural independence
  - Improved conceptual simplicity
  - Easier database design, implementation, management, and use
  - Ad hoc query capability with SQL
  - Powerful database management system

Disadvantages
  - Substantial hardware and system software overhead
  - Poor design and implementation is made easy
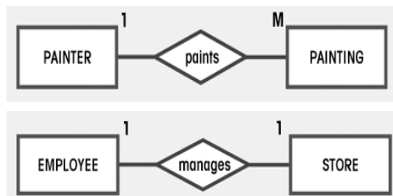 - May promote "islands of information" problems

---

Entity Relationship

- Based on the Relational Model concepts
  - Uses an Entity-Relation Diagram for representation
  - Based on Entities, Attributes and Relationships

Entity Relationship

- Advantages
  - Exceptional conceptual simplicity
  - Visual representation
  - Effective communication tool
  - Integrated with the relational database model
- Disadvantages
  - Limited constraint representation
  - Limited relationship representation
  - No data manipulation language
  - Loss of information content

Object Orientated

- Entity abstractions are stored as objects
  - Attributes are stored as object properties
  - Object Methods represent real-world actions
  - Similar objects are collected in a class
  - Classes are organised in a class hierarchy
  - Objects can inherit methods and properties from parent classes

---

Object Orientated

- Advantages
  - Adds semantic content
  - Visual presentation includes semantic content
  - Database integrity
  - Both structural and data independence
- Disadvantages
  - Lack of OODM
  - Complex navigational data access
  - Steep learning curve
  - High system overhead can lead to slows transactions