Overview of DBMS – CS502

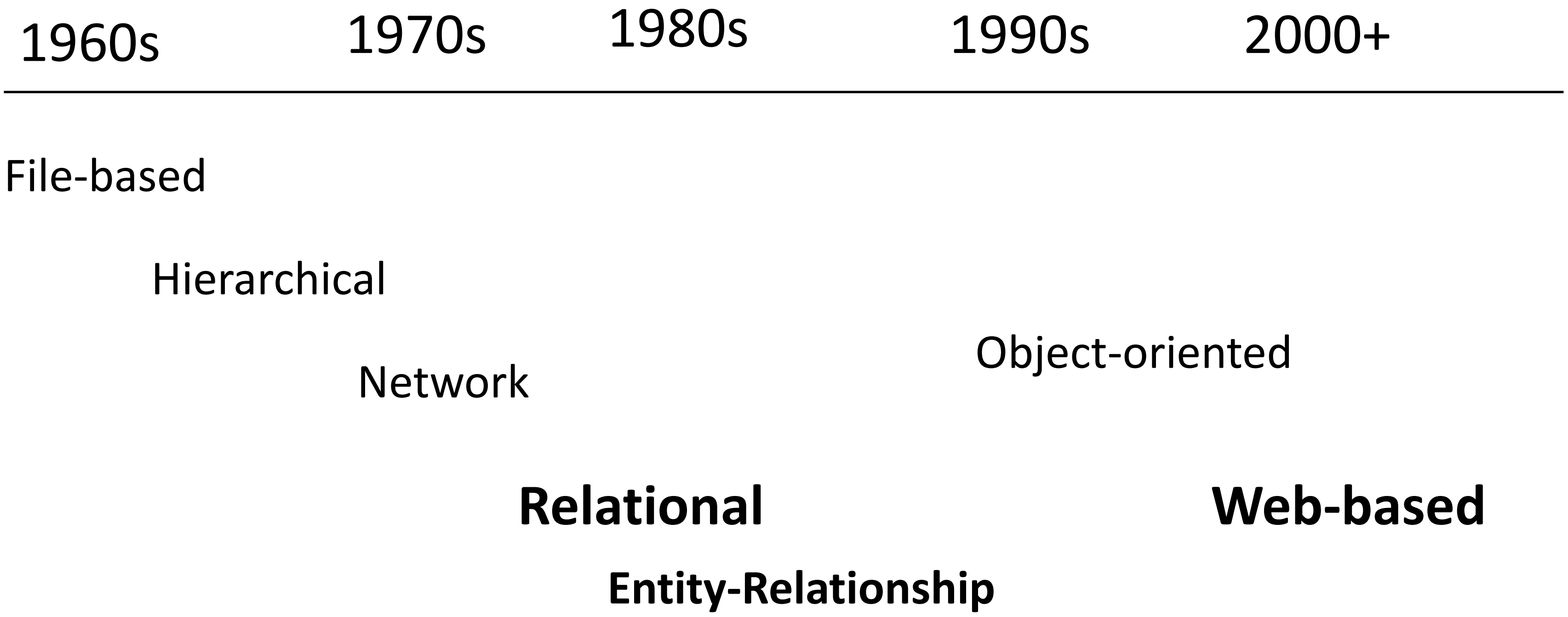
Uma Seshadri

Outline

- File system and Databases overview
- Database 3 – tier Architecture
- Database Characteristics
- Types of Databases
- Examples

Evolution of Database system

Timeline



1960s

1970s

1980s

1990s

2000+

File-based

Hierarchical

Network

Object-oriented

Relational

Web-based

Entity-Relationship

History of Database Systems

1950s and early 1960s:

- Data processing using magnetic tapes for storage
  - Tapes provided only sequential access
- Punched cards for input

Late 1960s and 1970s:

- Hard disks allowed direct access to data
- Network and hierarchical data models in widespread use
- Ted Codd defines the relational data model
- High-performance (for the era) transaction processing

History (cont.)

- 1980s:
  - Research relational prototypes evolve into commercial systems
    - SQL becomes industrial standard
  - Parallel and distributed database systems
  - Object-oriented database systems
- 1990s:
  - Large decision support and data-mining applications
  - Large multi-terabyte data warehouses
  - Emergence of Web commerce
- Early 2000s:
  - XML and XQuery standards
  - Automated database administration
- Later 2000s:
  - Giant data storage systems
    - Google BigTable, Yahoo PNuts, Amazon, ..

Database: Historical Roots

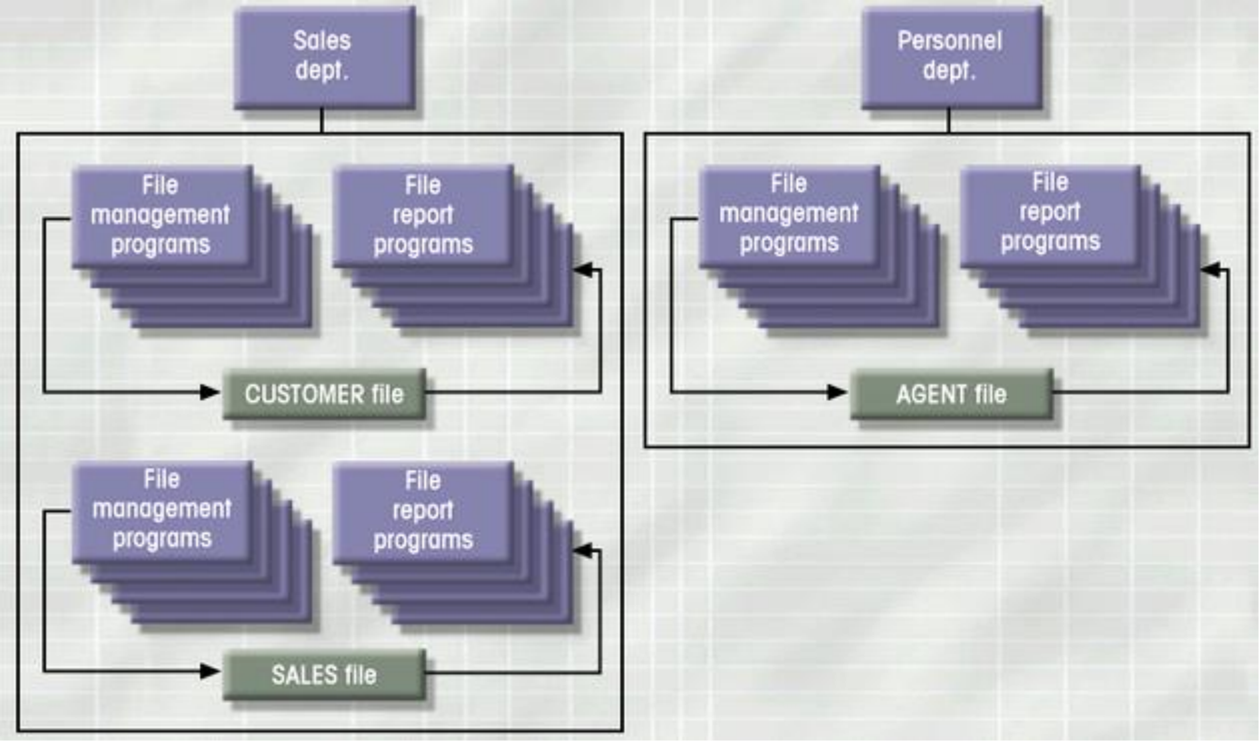
Manual File System

- to keep track of data
- used tagged file folders in a filing cabinet
- organized according to expected use
  - e.g. file per customer
- easy to create, but hard to
  - locate data
  - aggregate/summarize data

Computerized File System

- to accommodate the data growth and information need
- manual file system structures were duplicated in the computer
- Data Processing (DP) specialists wrote customized programs to
  - write, delete, update data (i.e. management)
  - extract and present data in various formats (i.e. report)

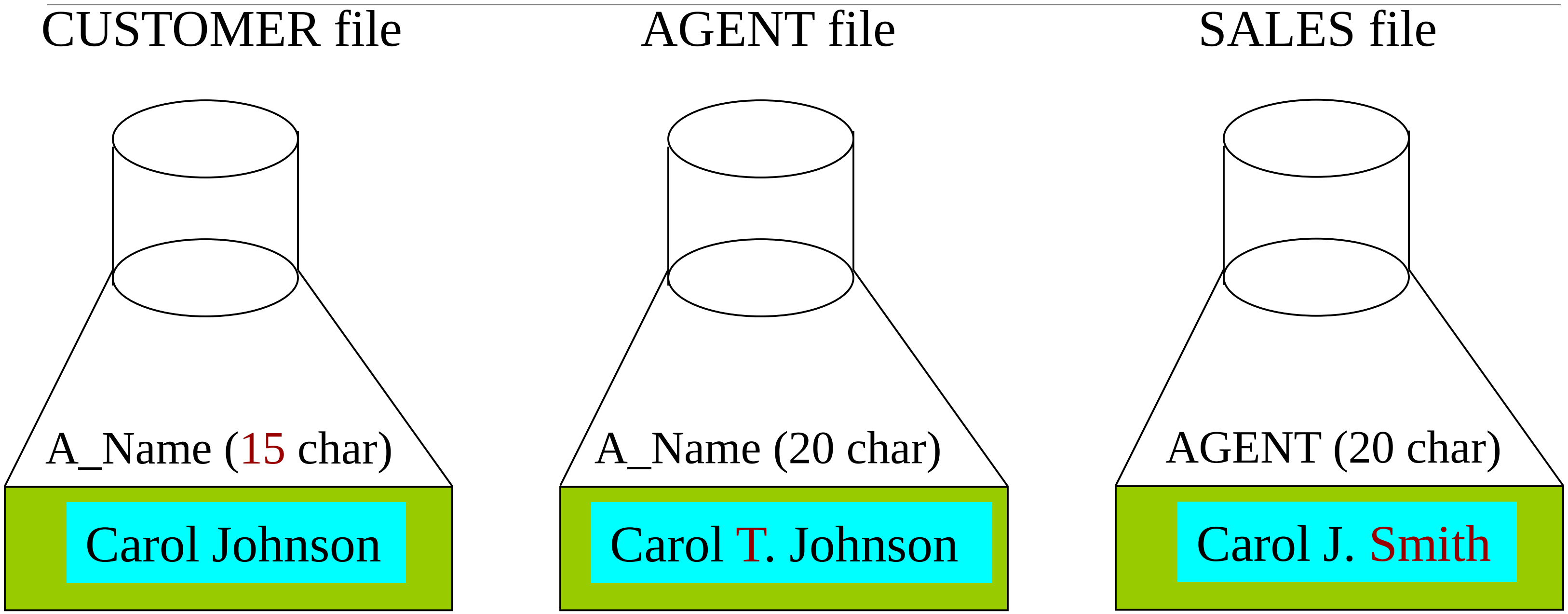
File System: Example



File System

- File System : Characteristics
  - Limited Storage space for Data
  - Data stored in files;
  - Concurrent users access many files
  - System crash affects File system
  - Password protected OS/Files

File System: Problem Case



- inconsistent field name, field size
- inconsistent data values
- data duplication

File System:

Weakness

- "Islands of data" in scattered file systems.

Challenges

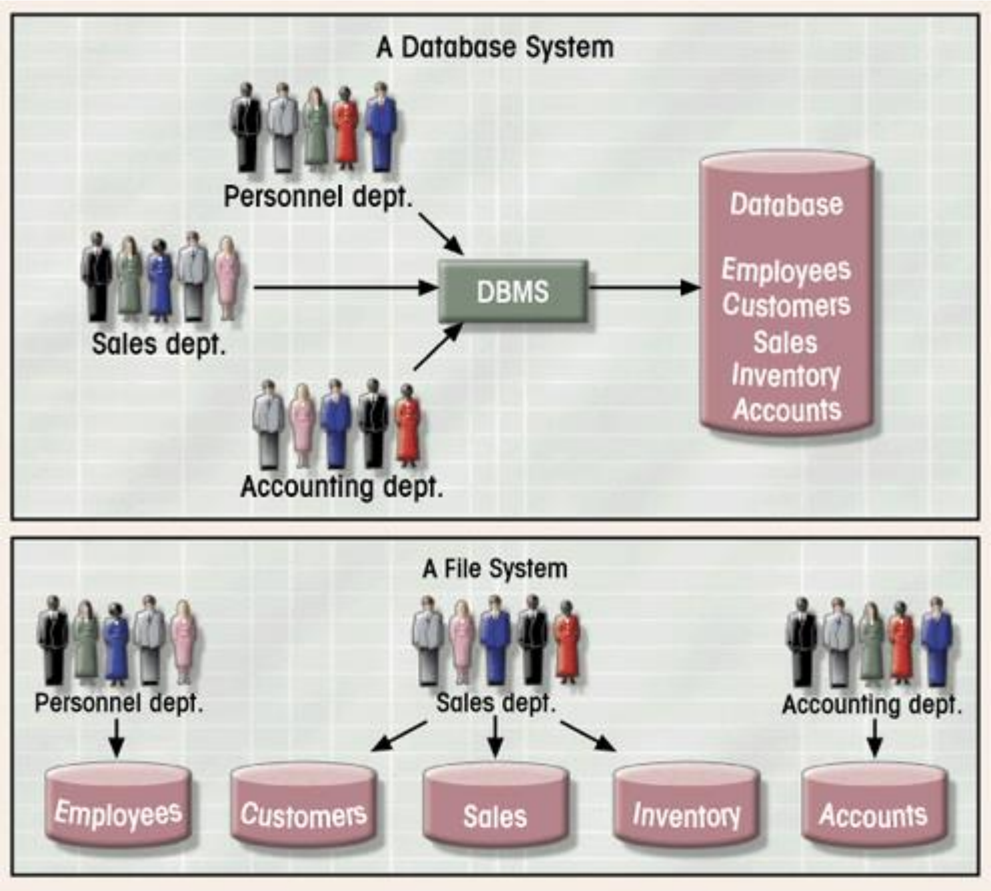
- Data redundancy and inconsistency
  - Multiple file formats, duplication of information in different files
- Difficulty in accessing data
  - Need to write a new program to carry out each new task
- Data isolation
  - Multiple files and formats
- Integrity problems
  - Integrity constraints (e.g., account balance > 0) Hard to add new constraints or change existing ones

File system - Disadvantages (Cont.)

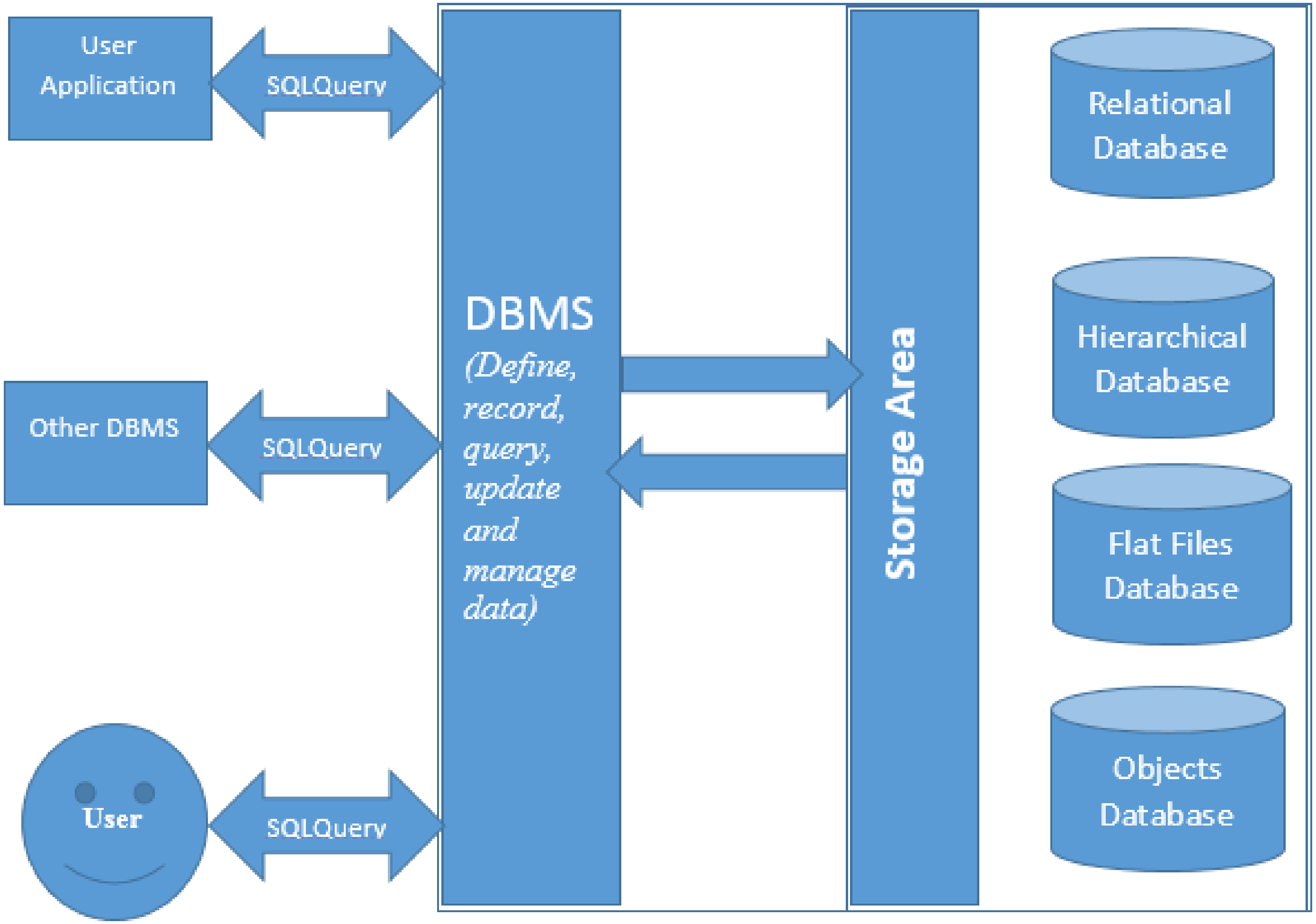
- Atomicity of updates
  - Failures may leave database in an inconsistent state with partial updates carried out
  - Example: Transfer of funds from one account to another should either complete or not happen at all
- Concurrent access by multiple users
  - Concurrent access needed for performance
  - Uncontrolled concurrent accesses can lead to inconsistencies
    - Example: Two people reading a balance (say 100) and updating it by withdrawing money (say 50 each) at the same time
- Security problems
  - Hard to provide user access to some, but not all, data

Database systems offer solutions to all the above problems

Database System vs. File System



DBMS system



User Application

SQLQuery

Other DBMS

SQLQuery

User

SQLQuery

DBMS

(Define, record, query, update and manage data)

Storage Area

Relational Database

Hierarchical Database

Flat Files Database

Objects Database

DBMS : Real life example

- Real Life Examples
  - Hotel : Taj, Le meridian (Users, Accommodation, tariff, locations)
  - Banking: SBI, CANARA,HDFC(Users, Transaction types, deposit, debits, transactions)
  - Airlines: Air India, Spice Jet, Indigo..(Passengers, Travel locations, Price)
  - Colleges: IIIT, IIM, IISc, Universities(Courses, Departments ,Students, Staff)
  - Online retailers: Amazon, Snap Deal, Flip cart (Buying and Selling)
  - Engineering: Inventory, Stores management,, Procurement, Manufacturing..
  - Company Database: Infosys, Wipro, Accenture (Employees, projects, clients)

DBMS - Overview

- What is **data** ?
  - Collection of known facts. Related to real life.

Eg: Student, Instructor, Department : ***Entities***

Admission No	Student	DoB	Gender	Year	Class
1001	Jayashree	18-12-1994	F	3	CSE
1002	John	19-12-1994	M	3	EE
1003	Smita	20-12-1993	F	3	ME
1004	Ravi	21-12-1994	M	3	EE

Instructor	Course	Admission No
Rajeev	MH101	1001
Srinivas	CV202	1002
Rama Devi	ME301	1003
Shivaram	EE203	1004

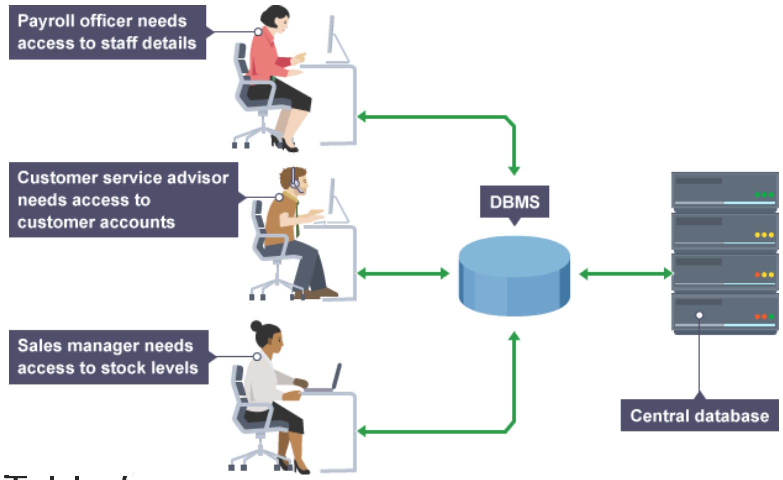
Dept.Name	Course	Instructor
EE	MH101	Rajeev
Civil	CV202	Srinivasa
ME	ME301	Rama Devi
CSE	EE203	Shivaram

Students course enrollment , Faculty teaching courses : ***Relationships***

- What is **Database**?
  - Collection of Organized data(relation data/Operational Data)
  - Processed to produce information
    - Eg: How many students enrolled for a particular course CV202?
    - How many Students are registered for CSE program?
    - How many faculties are employed in Civil dept?

DBMS - Overview

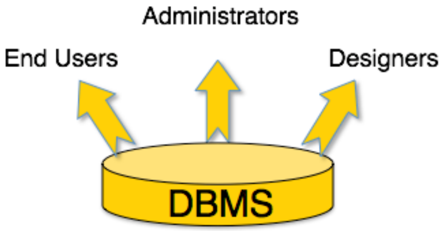
- **Database Management System**
  - Data is stored in a system
  - Helps users to Create, Maintain and Manipulate data
  - Collection of Programs

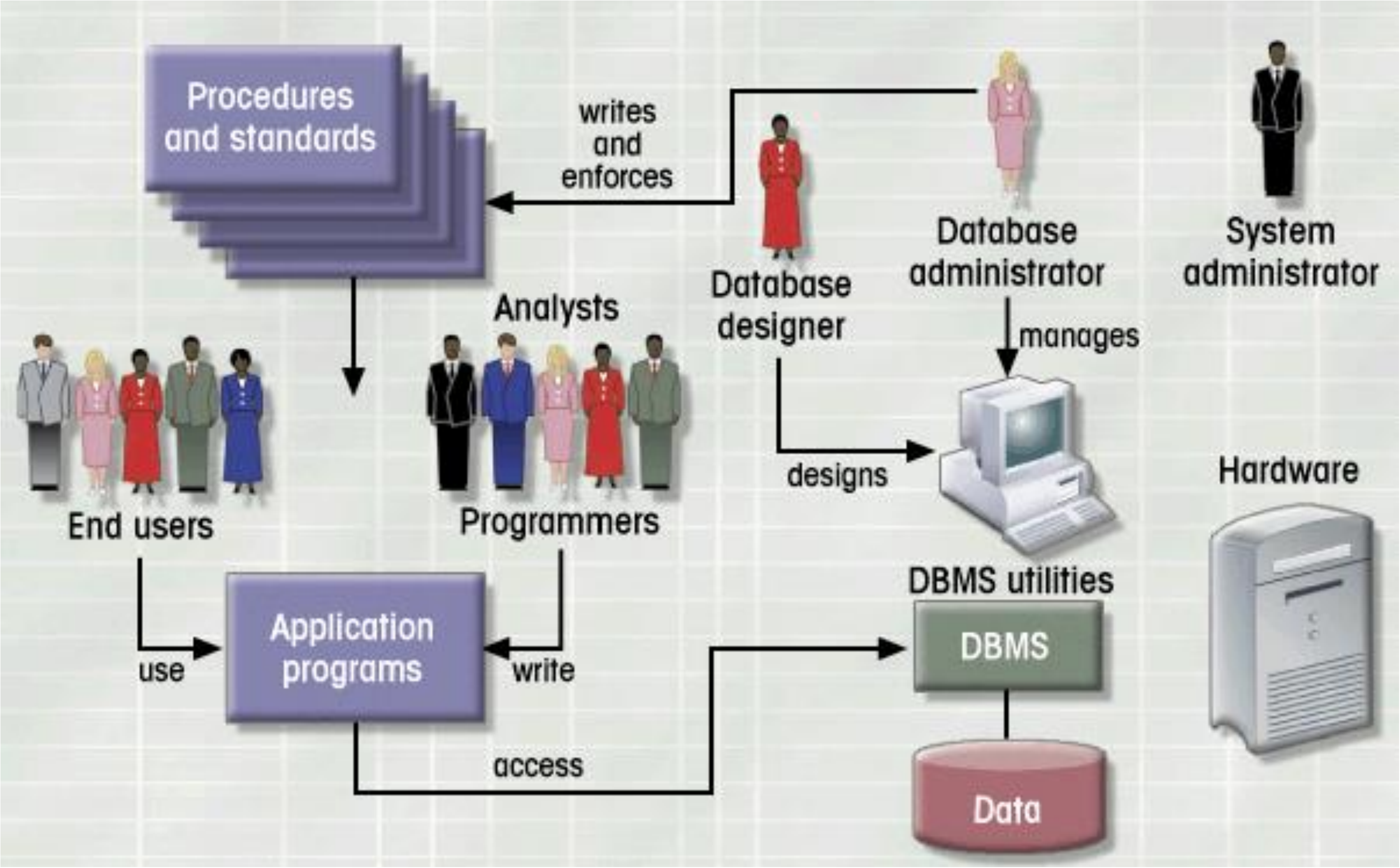

Payroll officer needs access to staff details
Customer service advisor needs access to customer accounts
Sales manager needs access to stock levels
DBMS
Central database

- Characteristics of DBMS:
  - **Real world data** (entities)
  - **Table form ;** Relations among data are represented in Table form
  - **Data program independence;** Isolation among data and application
  - **Minimize redundancy**; Tables are formed using normalization;
  - **Query processing** for Data access
  - **ACID** (Atomicity, Consistency, Isolation and Durability) based Transactions
  - **Multiuser and Concurrent Access**
  - **Multiple Views**

DBMS - Overview

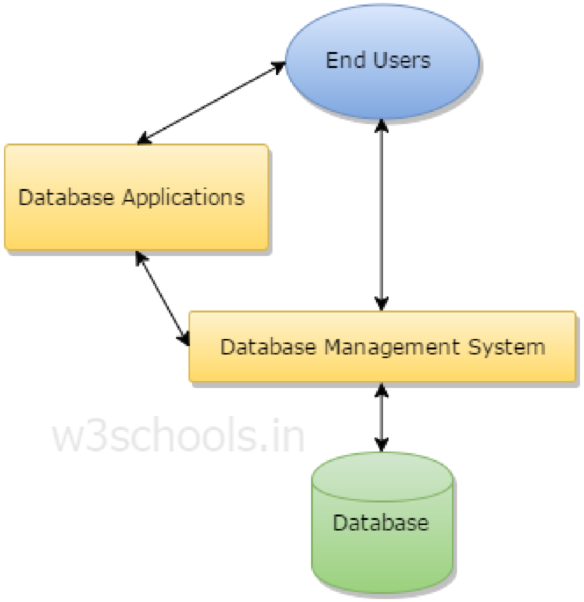
- **Users**
  - Database Administrator
    - Responsible for maintaining the database activities (access, coordination, manipulation…)
  - End Users
    - Users of the System.
  - Data base designers
    - Design the Data base, Architecture, Data structure..



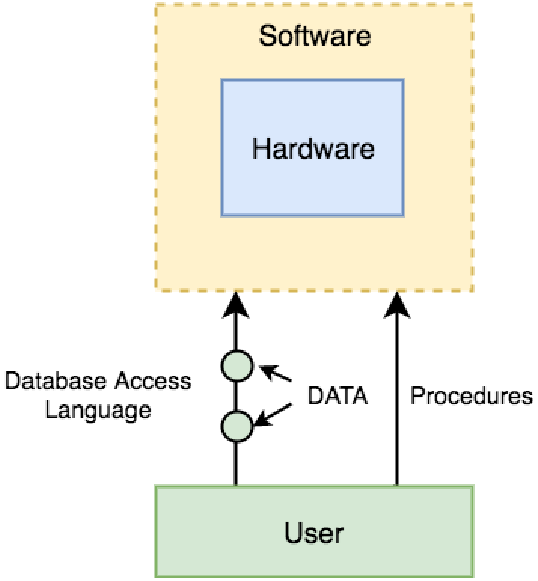

Procedures and standards
writes and enforces
Database administrator
System administrator
Database designer
Analysts
manages
End users
Programmers
designs
Hardware
DBMS utilities
Application programs
use
write
DBMS
access
Data

DBMS - Overview

- DBMS contains information about a particular enterprise
  - Users access and manipulate the data
  - Set of programs to access the data
  - An environment that is both *convenient* and *efficient* to use



Components of a Database Management System

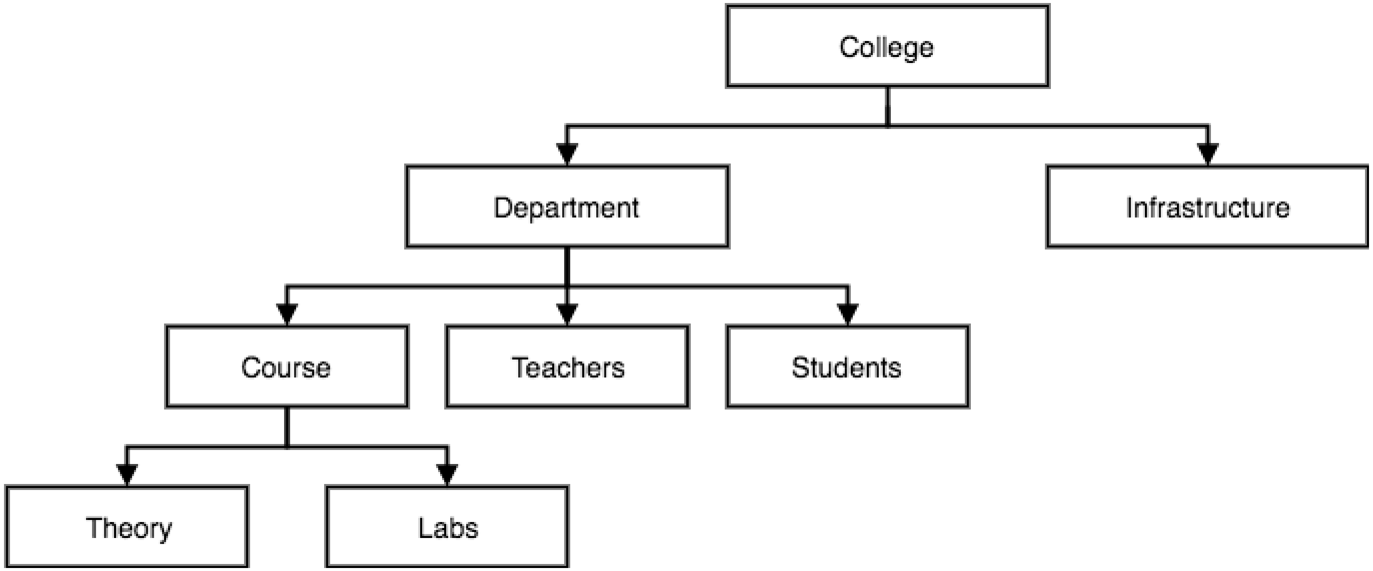


College DBMS– Example 1

Application program examples

- Add new students, instructors, and courses
- Register students for courses, and generate class rosters
- Assign grades to students, compute grade point averages (GPA) and generate transcripts

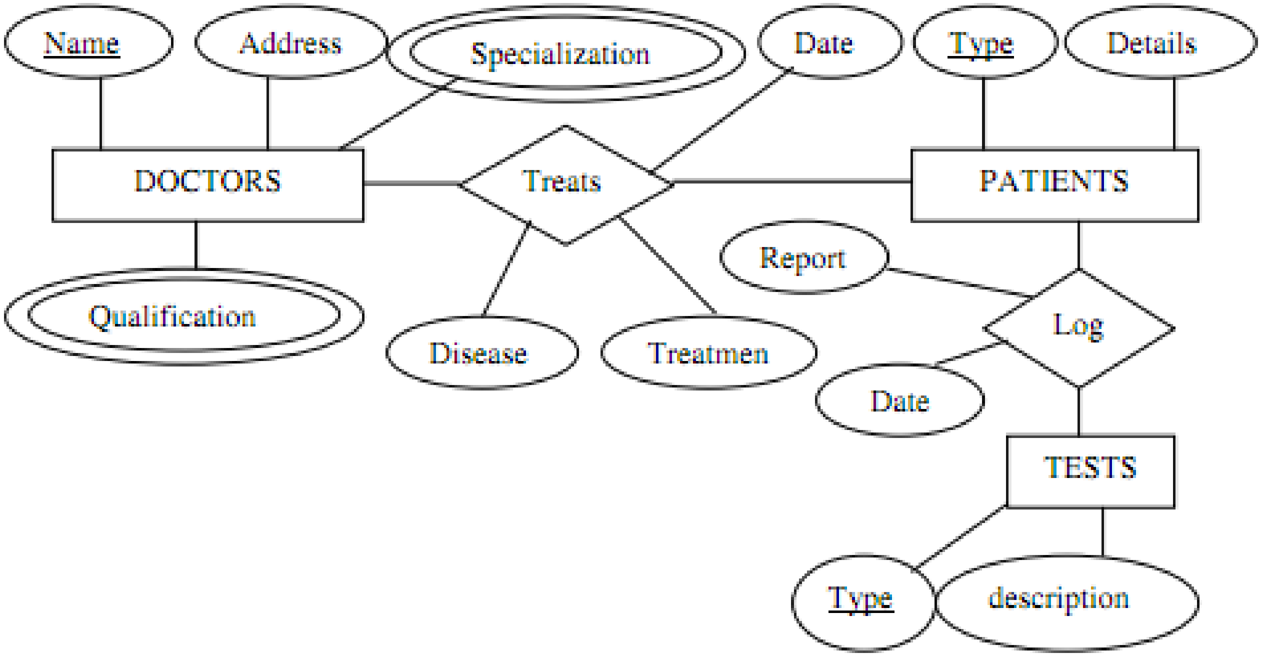
In the early days, database applications were built directly on top of file systems

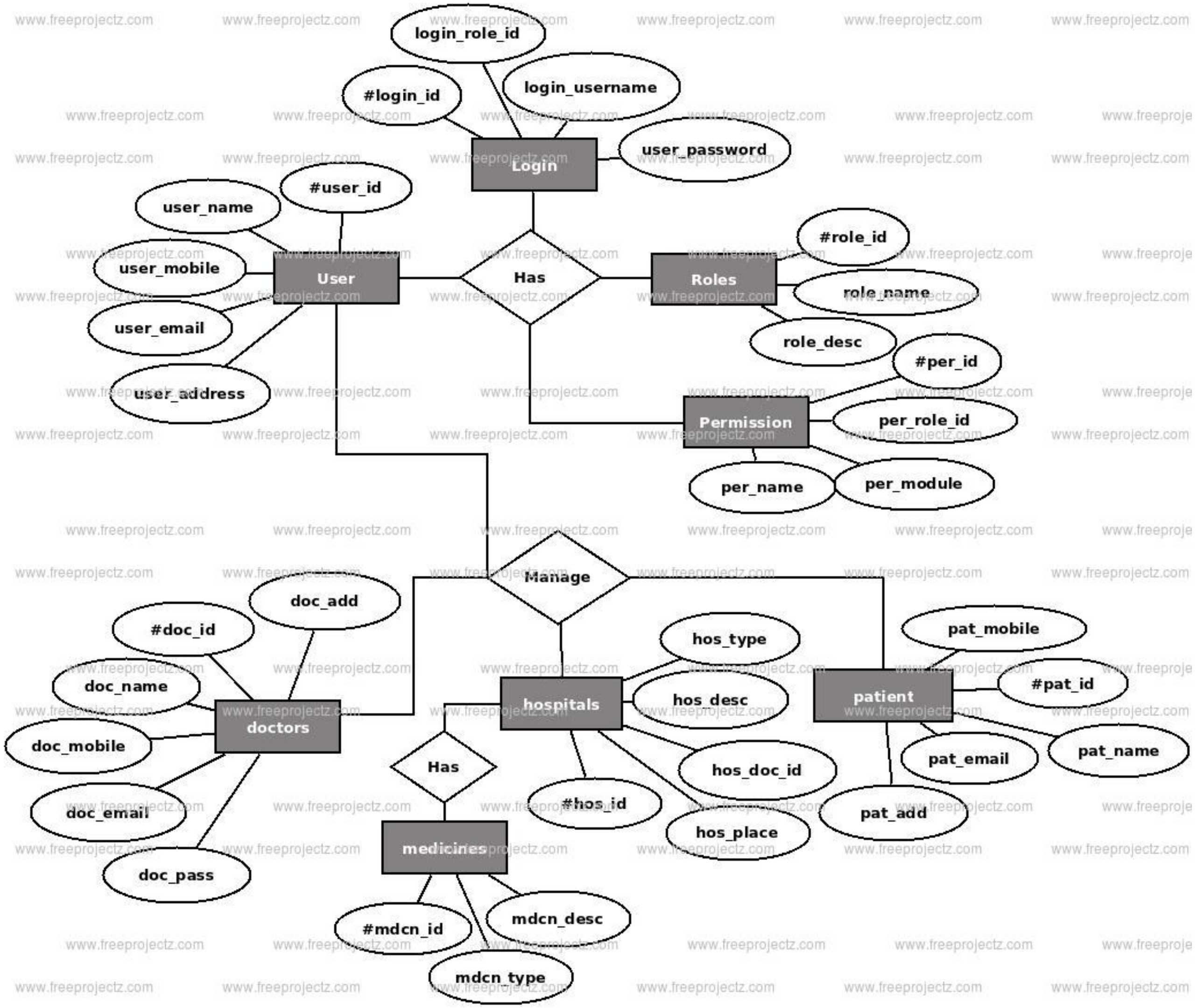


HOSPITAL DBMS – Example 2

Application program examples

- Add new patients, doctors, departments and treatment types
- Register patients, Conduct Tests (Diagnosis)
- Assign Doctors to Patients, Perform Treatments, Generate Health records and Collect Fees
- Discharge Patients



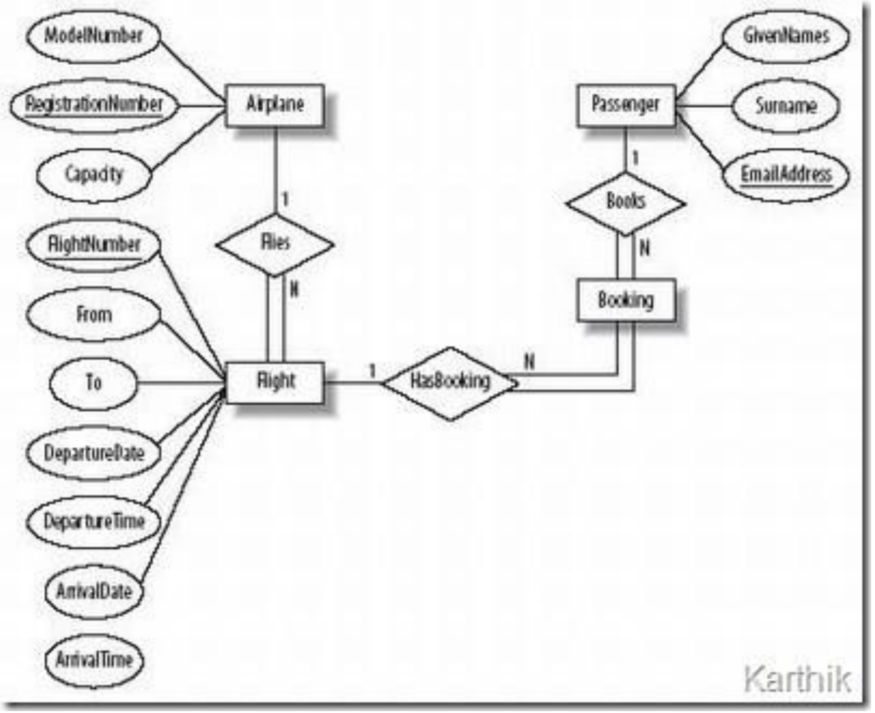


ER Diagram For Hospital Management System

Airline DBMS – Example 3

Application program examples

- Add new passenger, Users(Administrators, Database managers), Location details
- Register passengers ticket request(ticket booking, cancelation, modification) ..
- Generate Tickets/receipts and Collect Fees



DBMS : Characteristics

Real-world entity −

- DBMS is more realistic and uses real-world entities to design its architecture.
- It uses the behavior and attributes too.

For example, a school database may use students as an entity and their age as an attribute.

Relation-based tables −

- DBMS allows entities and relations among them to form tables.
- A user can understand the architecture of a database just by looking at the table names.

Characteristics

Isolation of data and application −

- A database system is entirely different than its data.
- A database is an active entity, whereas data is said to be passive, on which the database works and organizes.
- DBMS also stores metadata, which is data about data, to ease its own process.

Less redundancy −

- DBMS follows the rules of normalization, which splits a relation when any of its attributes is having redundancy in values.
- Normalization is a mathematically rich and scientific process that reduces data redundancy.

Characteristics

Consistency −

- Consistency is a state where every relation in a database remains consistent. There exist methods and techniques, which can detect attempt of leaving database in inconsistent state.
- A DBMS can provide greater consistency as compared to earlier forms of data storing applications like file-processing systems.

Query Language −

- DBMS is equipped with query language, which makes it more efficient to retrieve and manipulate data.
- A user can apply as many and as different filtering options as required to retrieve a set of data.
- it was not possible where file-processing system was used.

Characteristics

ACID Properties −

- DBMS follows the concepts of **A**tomicity, **C**onsistency, **I**solation, and **D**urability *normally shortened as ACID*.
- These concepts are applied on transactions, which manipulate data in a database. ACID properties help the database stay healthy in multitransactional environments and in case of failure.

Multiuser and Concurrent Access −

- DBMS supports multi-user environment and allows them to access and manipulate data in parallel.
- There are restrictions on transactions when users attempt to handle the same data item, but users are always unaware of them.

Characteristics

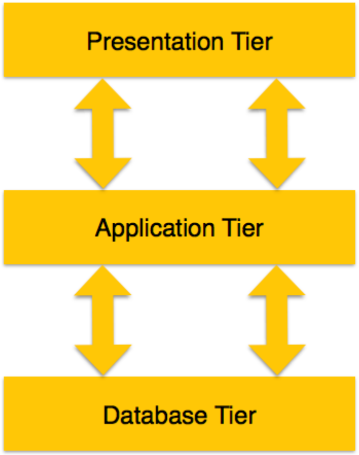
Multiple views –

- DBMS offers multiple views for different users.
- Eg: A user who is in the Sales department will have a different view of database than a person working in the Production department.
- This feature enables the users to have a concentrate view of the databaseaccording to their requirements.

Security –

- Features like multiple views offer security to some extent where users are unable to access data of other users and departments.
- DBMS offers methods to impose constraints while entering data into the database and retrieving the same at a later stage. DBMS offers

DBMS Architecture



One Tier Architecture

- DBMS is the only entity
- User directly uses the DBMS

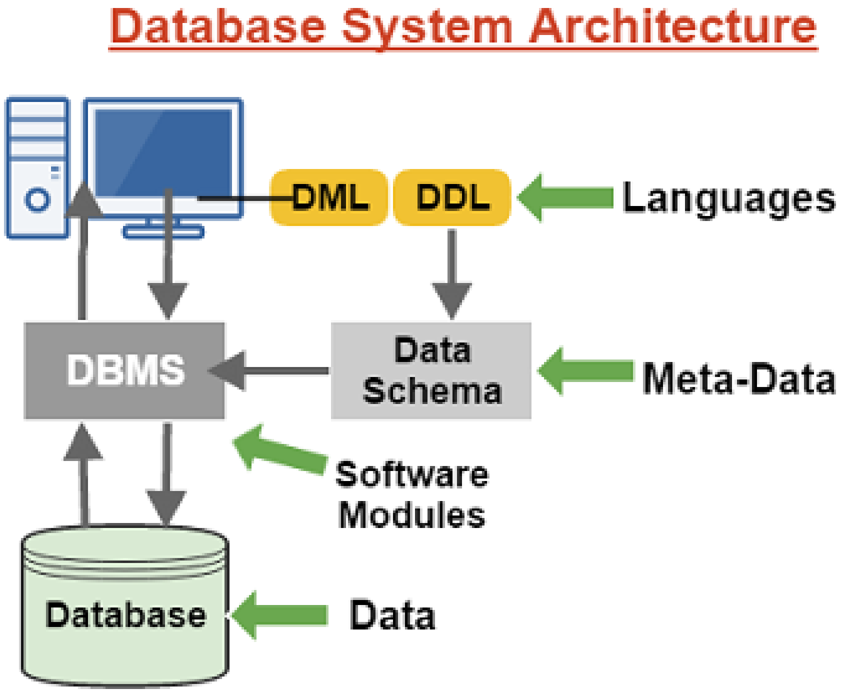
Two Tier Architecture:

- User uses/writes application to Access the data

Three Tier

- Database Tier : Database resides along with its query processing languages
- Application server Tier : programs reside that access the database
- End-users tier: Users access the data thru multiple Views

DBMS Architecture(Contd..)



Data Definition Language (DDL)

Specification notation for defining the database schema

Example:

```
create table Student (
 ID            char(5),
 name          varchar(25),
 dept_name     varchar(25),
 course fee    numeric(6,2)
```

DDL compiler generates a set of table templates stored in a ***data dictionary***

Data dictionary contains metadata (i.e., data about data)

- Database schema
- Integrity constraints
  - Primary key (ID uniquely identifies instructors)
- Authorization
  - Who can access what

Data Manipulation Language (DML)

Language for accessing and manipulating the data organized by the appropriate data model

- DML also known as query language

Two classes of languages

- **Pure** – used for proving properties about computational power and for optimization
  - Relational Algebra
  - Tuple relational calculus
  - Domain relational calculus
- **Commercial** – used in commercial systems
  - SQL is the most widely used commercial language

SQL

widely used commercial language

SQL is NOT a Turing machine equivalent language

To be able to compute complex functions SQL is usually embedded in some higher-level language

Application programs generally access databases through one of

- Language extensions to allow embedded SQL
- Application program interface (e.g., ODBC/JDBC) which allow SQL queries to be sent to a database

Hierarchical Database

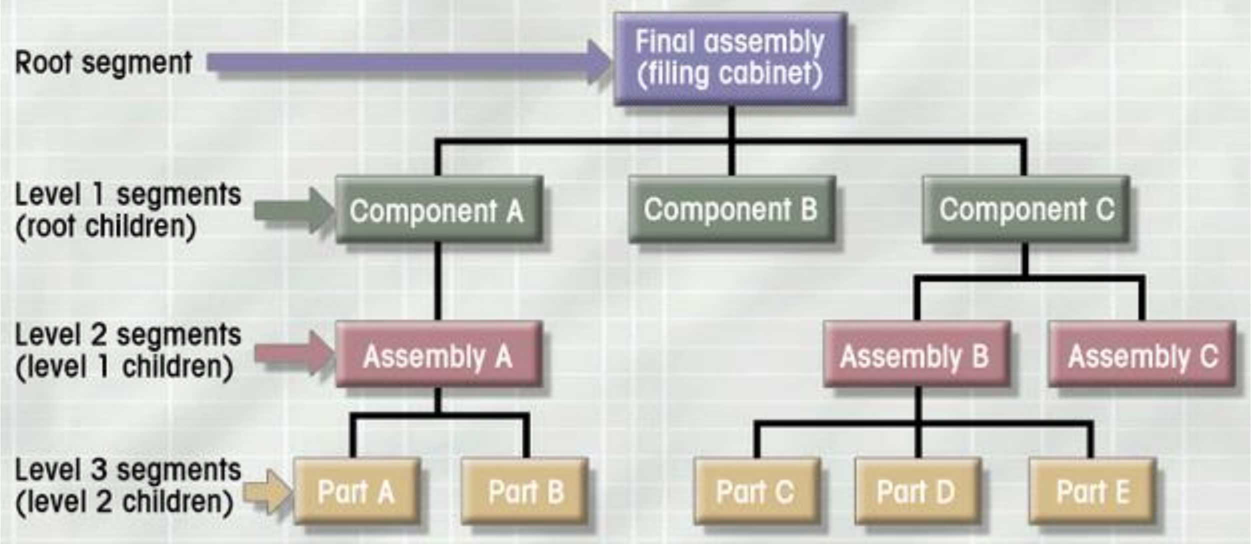
Background

- Developed to manage large amount of data for complex manufacturing projects
- e.g., Information Management System (IMS)
  - IBM-Rockwell joint venture
  - clustered related data together
  - hierarchically associated data clusters using pointers

Hierarchical Database Model

- Assumes data relationships are hierarchical
  - One-to-Many (1:M) relationships
    - Each parent can have many children
    - Each child has only one parent
- Logically represented by an upside down tree

Hierarchical Database: Example


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

Database Systems: Design, Implementation, & Management: Rob & Coronel

Hierarchical Database: Pros & Cons

Advantages

- Conceptual simplicity
  - groups of data could be related to each other
  - related data could be viewed together
- Centralization of data
  - reduced redundancy and promoted consistency

Disadvantages

- Limited representation of data relationships
  - did not allow Many-to-Many (M:N) relations
- Complex implementation
  - required in-depth knowledge of physical data storage
- Structural Dependence
  - data access requires physical storage path
- Lack of Standards
  - limited portability

Network Database

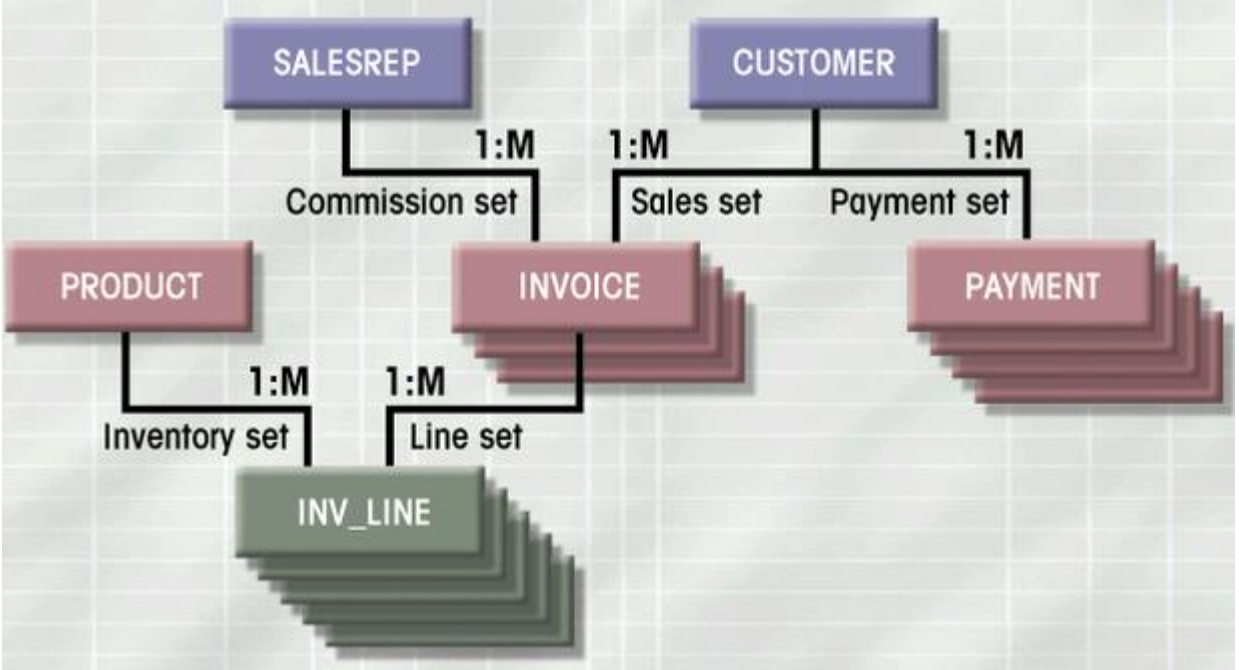
Objectives

- Represent more complex data relationships
- Improve database performance
- Impose a database standard

Network Database Model

- Similar to Hierarchical Model
  - Records linked by pointers
- Composed of sets
  - Each set consists of owner (parent) and member (child)
- Many-to-Many (M:N) relationships representation
  - Each owner can have multiple members (1:M)
  - A member may have several owners

Network Database: Example



Database Systems: Design, Implementation, & Management: Rob & Coronel

Network Database: Pros & Cons

Advantages

- More data relationship types
- More efficient and flexible data access
  - “network” vs. “tree” path traversal
- Conformance to standards
  - enhanced database administration and portability

Disadvantages

- System complexity
  - require familiarity with the internal structure for data access
- Lack of structural independence
  - small structural changes require significant program changes

Relational Database

Problems with legacy database systems

- Required excessive effort to maintain
  - Data manipulation (programs) too dependent on physical file structure
- Hard to manipulate by end-users
  - No capacity for ad-hoc query (must rely on DB programmers).

Evolution in Data Organization

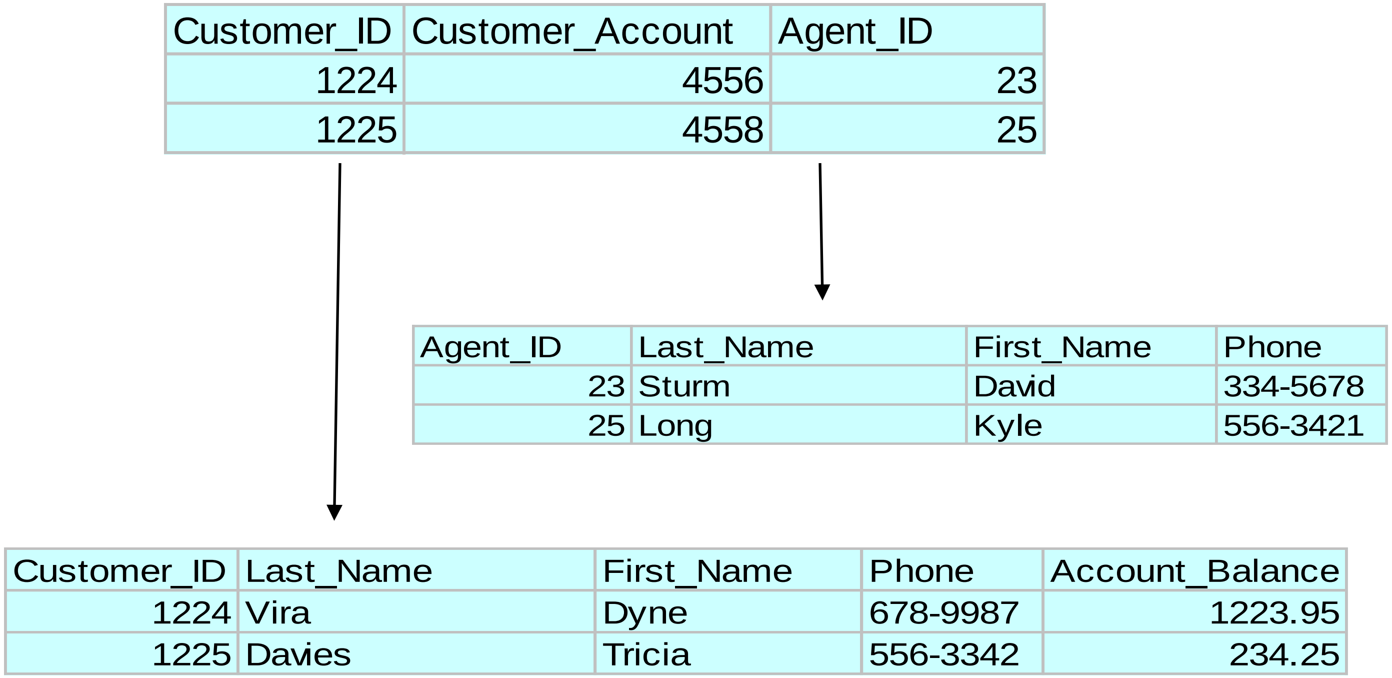
- E. F. Codd’s Relational Model proposal
  - Separated the notion of physical representation (machine-view) from logical representation (human-view)
  - Considered ingenious but computationally impractical in 1970

Relational Database (contd.)

- Relational Database Model
  - Dominant database model of today
  - Eliminated pointers and used tables to represent data
  - Tables
    - flexible logical structure for data representation
    - a series of row/column intersections
    - related by sharing common entity characteristic(s)

Relational Database: Example

- Provides a logical "human-level" view of the data and associations among groups of data (i.e., tables)



Customer_ID	Customer_Account	Agent_ID
1224	4556	23
1225	4558	25

Agent_ID	Last_Name	First_Name	Phone
23	Sturm	David	334-5678
25	Long	Kyle	556-3421

Customer_ID	Last_Name	First_Name	Phone	Account_Balance
1224	Vira	Dyne	678-9987	1223.95
1225	Davies	Tricia	556-3342	234.25

Relational Database: Pros & Cons

Advantages

- Structural independence
  - Separation of database design and physical data storage/access
  - Easier database design, implementation, management, and use
- Ad hoc query capability with Structured Query Language (SQL)
  - SQL translates user queries to codes

Disadvantages

- Substantial hardware and system software overhead
  - more complex system
- Poor design and implementation is made easy
  - ease-of-use allows careless use of RDBMS

End of Unit 1.1