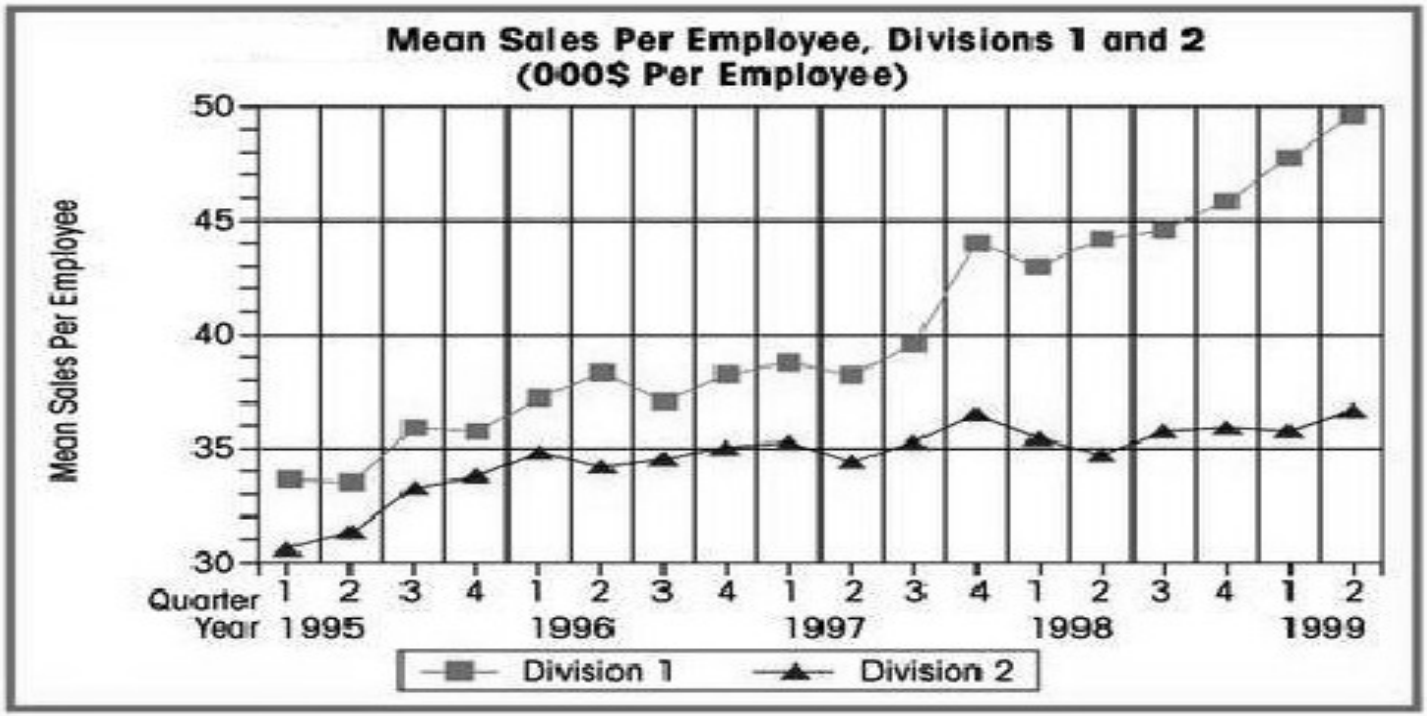
Review of Major DB Concepts

- Data and Information
  - Data: raw facts
  - Information: processed data
- Database
- Metadata
- DBMS: Database management system
- Database system

An Example of Data

- Sales per employee for each of X-company's two divisions



Data and Its Structure

- Data is actually stored as bits, but it is difficult to work with data at this level.
- It is convenient to view data at different levels of abstraction.
- **Schema**: Description of data at some level. Each level has its own schema.
- Three schemas: physical, conceptual, and external.

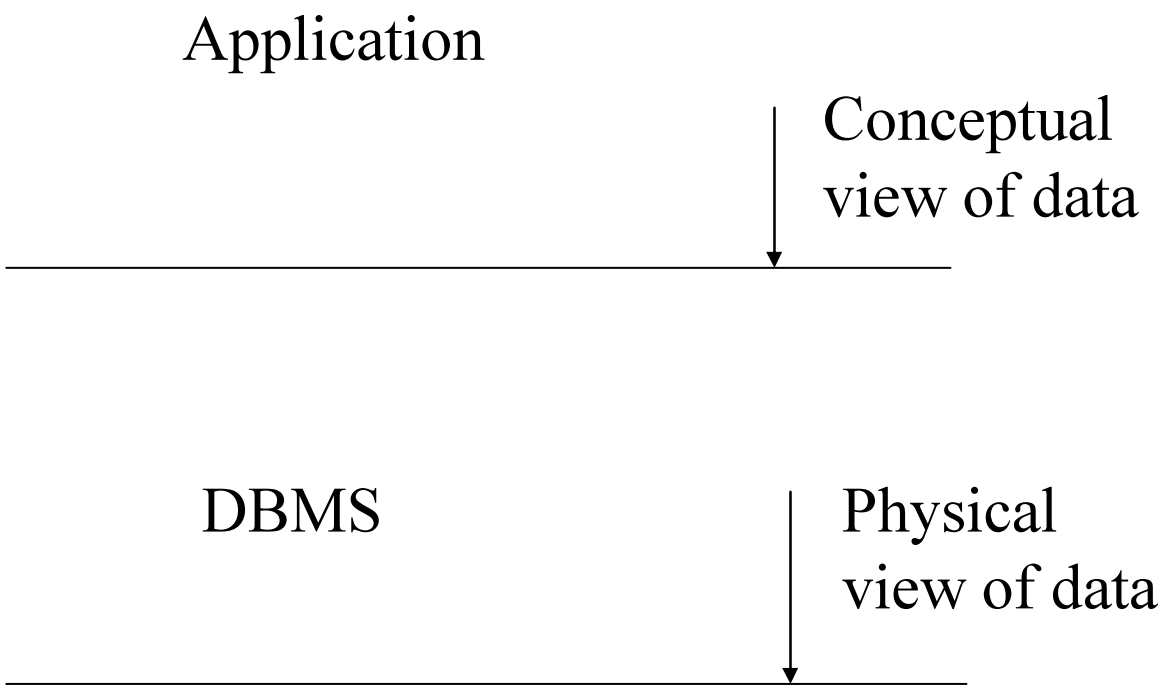
Physical Data Schema

- Describes details of **how** data is stored: tracks, cylinders, indices etc.
- Early applications worked at this level - explicitly dealt with details.
- **Problem:** Routines hard-coded to deal with physical representation.
  - Changes to data structure difficult to make.
  - Application code becomes complex since it must deal with details.
  - Rapid implementation of new features impossible.

Conceptual Data Level

- Hides details.
  - In the relational model, the conceptual schema presents data as a set of tables.
- DBMS maps from conceptual to physical schema automatically.
- Physical schema can be changed without changing application:
  - DBMS must change mapping from conceptual to physical.
- Referred to as *physical data independence*.

Conceptual Data Level (con't)



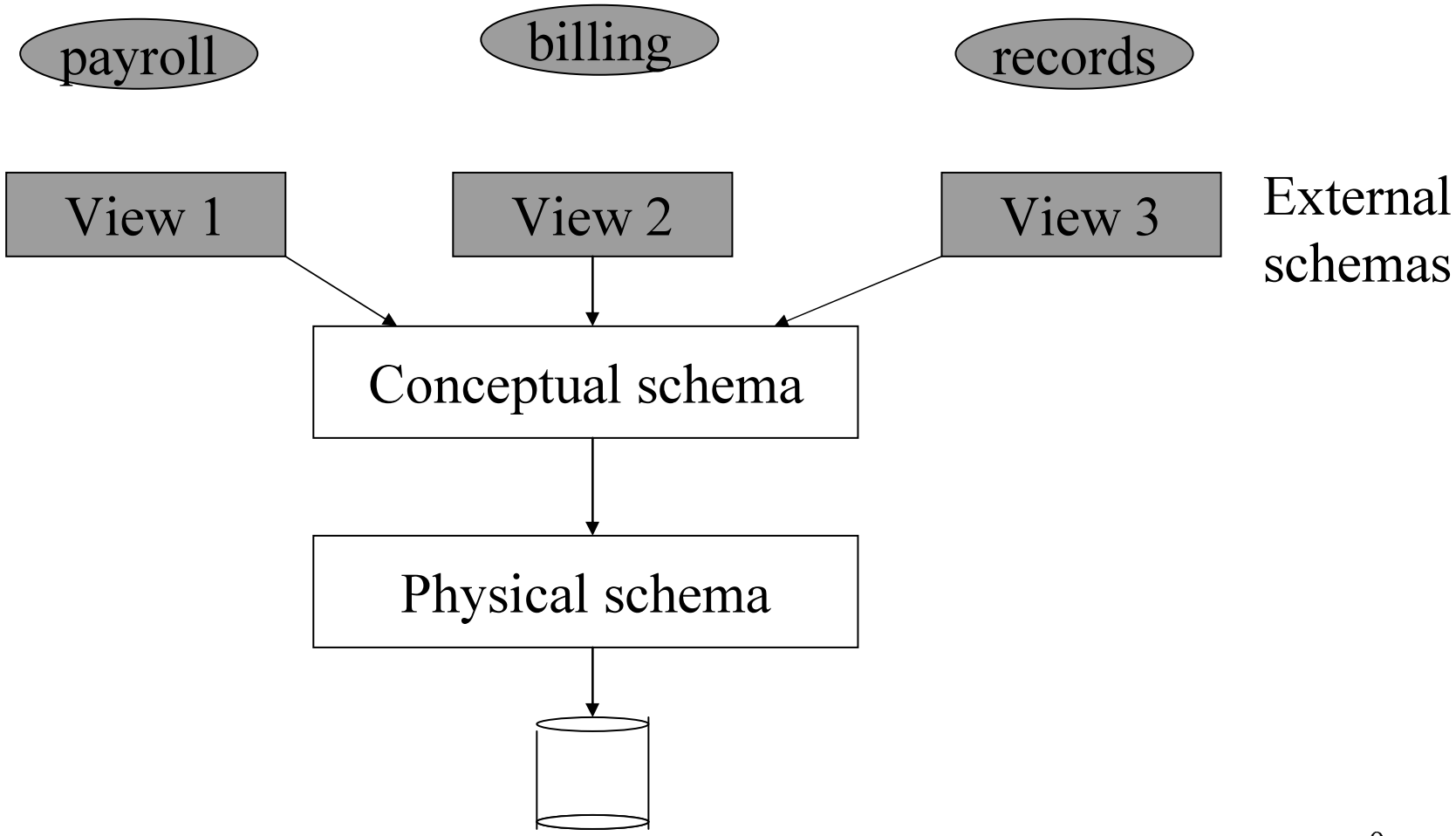
External Data Level

- In the relational model, the external schema also presents data as a set of relations.
- An external schema specifies a *view* of the data in terms of the conceptual level. It is tailored to the needs of a particular category of users.
  - Portions of stored data should not be seen by some users.
    - Students should not see faculty salaries.
    - Faculty should not see billing data.
  - Information that can be derived from stored data might be viewed as if it were stored.
    - GPA not stored, calculated when needed.

External Data Level (con't)

- Application is written in terms of an external schema.
- A view is computed when accessed (not stored).
- Different external schemas can be provided to different categories of users.
- Translation from external to conceptual done automatically by DBMS at run time.
- Conceptual schema can be changed without changing application:
  - Mapping from external to conceptual must be changed.
- Referred to as *conceptual data independence.*

Levels of Abstraction



Data Independence

- **Logical Data Independence**: The capacity to change the conceptual schema without having to change the external schemas and their application programs
- **Physical Data Independence**: The capacity to change the internal schema without having to change the conceptual schema.

TRANSCRIPT	StudentName	Student Transcript				
		CourseNumber	Grade	Semester	Year	SectionId
	Smith	CS1310	C	Fall	99	119
		MATH2410	B	Fall	99	112
	Brown	MATH2410	A	Fall	98	85
		CS1310	A	Fall	98	92
		CS3320	B	Spring	99	102
		CS3380	A	Fall	99	135

RADE_REPORT	StudentNumber	SectionIdentifier	Grade
	17	112	B
	17	119	C
	8	85	A
	8	92	A
	8	102	B
	8	135	A

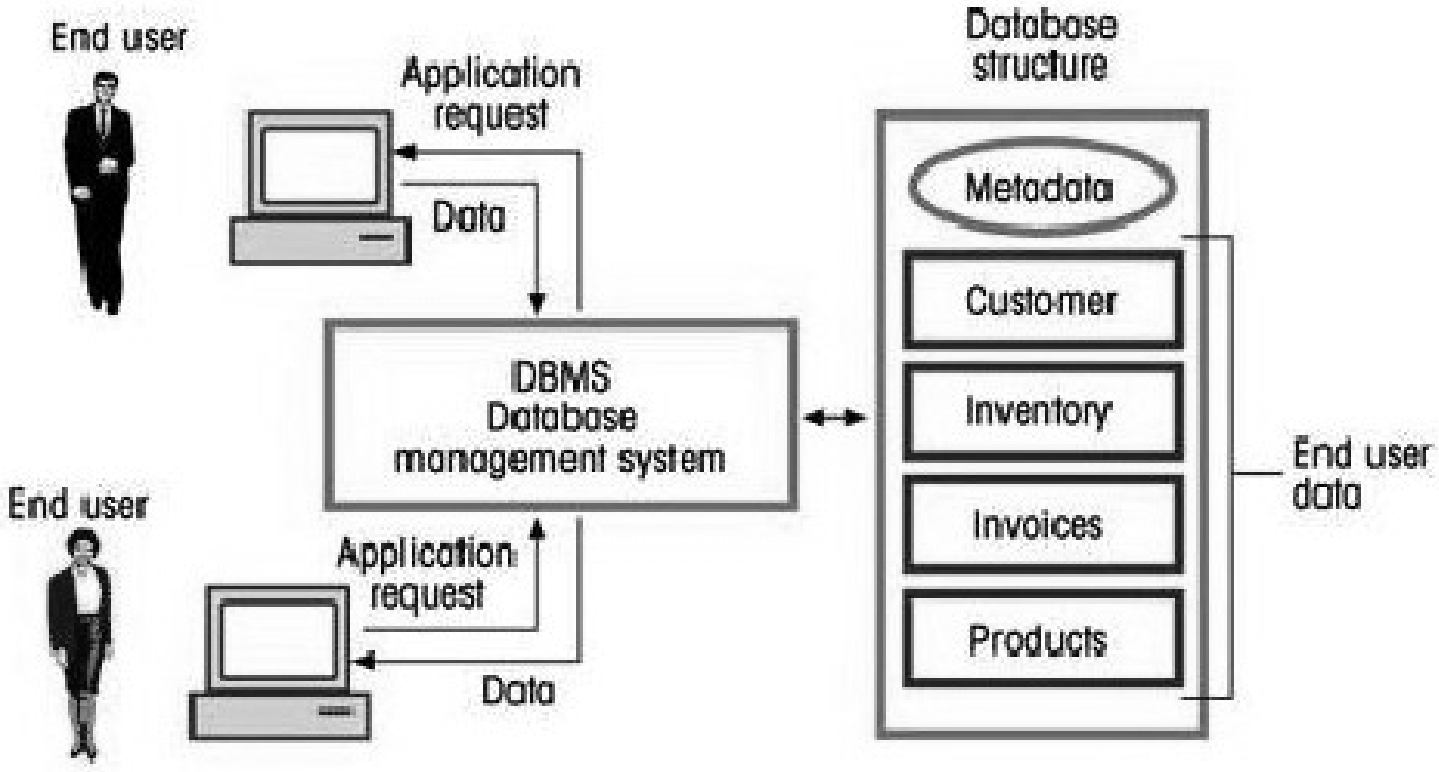
GRADE_REPORT	StudentNumber	StudentName	SectionIdentifier	CourseNumber	Grade
	17	Smith	112	MATH2410	B
	17	Smith	119	CS1310	C
	8	Brown	85	MATH2410	A
	8	Brown	92	CS1310	A
	8	Brown	102	CS3320	B
	8	Brown	135	CS3380	A

Data indep is accomplished:

changing a schema at a lower level of DB,

only the **mappings** between this schema and higher-level schemas need to be changed in a DBMS that fully supports data independence. The higher-level schemas themselves are *unchanged.* Hence, the application programs need not be changed since they refer to the external schemas.

A Database System



Historical Roots

- File systems
  - Provides historical perspective
  - Avoid pitfalls of data mgmt
  - Simple characteristics
  - Knowledge of converting a file system to a DB system

Contents of the CUSTOMER File

C_NAME	C_PHONE	C_ADDRESS	C_ZIP	A_NAME	A_PHONE	TP	AMT	REN
Alfred A. Ramas	615-844-2573	218 Fork Rd., Babs, TN	36123	Leah F. Hahn	615-882-1244	T1	$100.00	3/12/99
Leona K. Dunne	713-894-1238	Box 12A, Fox, KY	25246	Alex B. Alby	713-228-1249	T1	$250.00	5/23/99
Kathy W. Smith	615-894-2285	125 Oak Ln, Babs, TN	36123	Leah F. Hahn	615-882-2144	S2	$150.00	1/5/98
Paul F. Olowski	615-894-2180	217 Lee Ln., Babs, TN	36123	Leah F. Hahn	615-882-1244	S1	$300.00	9/20/99
Myron Orlando	615-222-1672	Box 111, New, TN	36155	Alex B. Alby	713-228-1249	T1	$100.00	12/4/99
Amy B. O'Brian	713-442-3381	387 Troll Dr., Fox, KY	25246	John T. Okon	615-123-5589	T2	$850.00	8/29/99
James G. Brown	615-297-1228	21 Tye Rd., Nash, TN	37118	Leah F. Hahn	615-882-1244	S1	$120.00	3/1/99
George Williams	615-290-2556	155 Maple, Nash, TN	37119	John T. Okon	615-123-5589	S1	$250.00	6/23/99
Anne G. Farriss	713-382-7185	2119 Elm, Crew, KY	25432	Alex B. Alby	713-228-1249	T2	$100.00	11/9/99
Olette K. Smith	615-297-3809	2782 Main, Nash, TN	37118	John T. Okon	615-123-5589	S2	$500.00	2/18/99

C_NAME = Customer name
C_PHONE = Customer phone
C_ADDRESS = Customer address
C_ZIP = Customer ZIP code

A_NAME = Agent name
A_PHONE = Agent phone
TP = Insurance type
AMT = Insurance policy amount, in thousands of $
REN = Insurance renewal date

Basic File Terminology

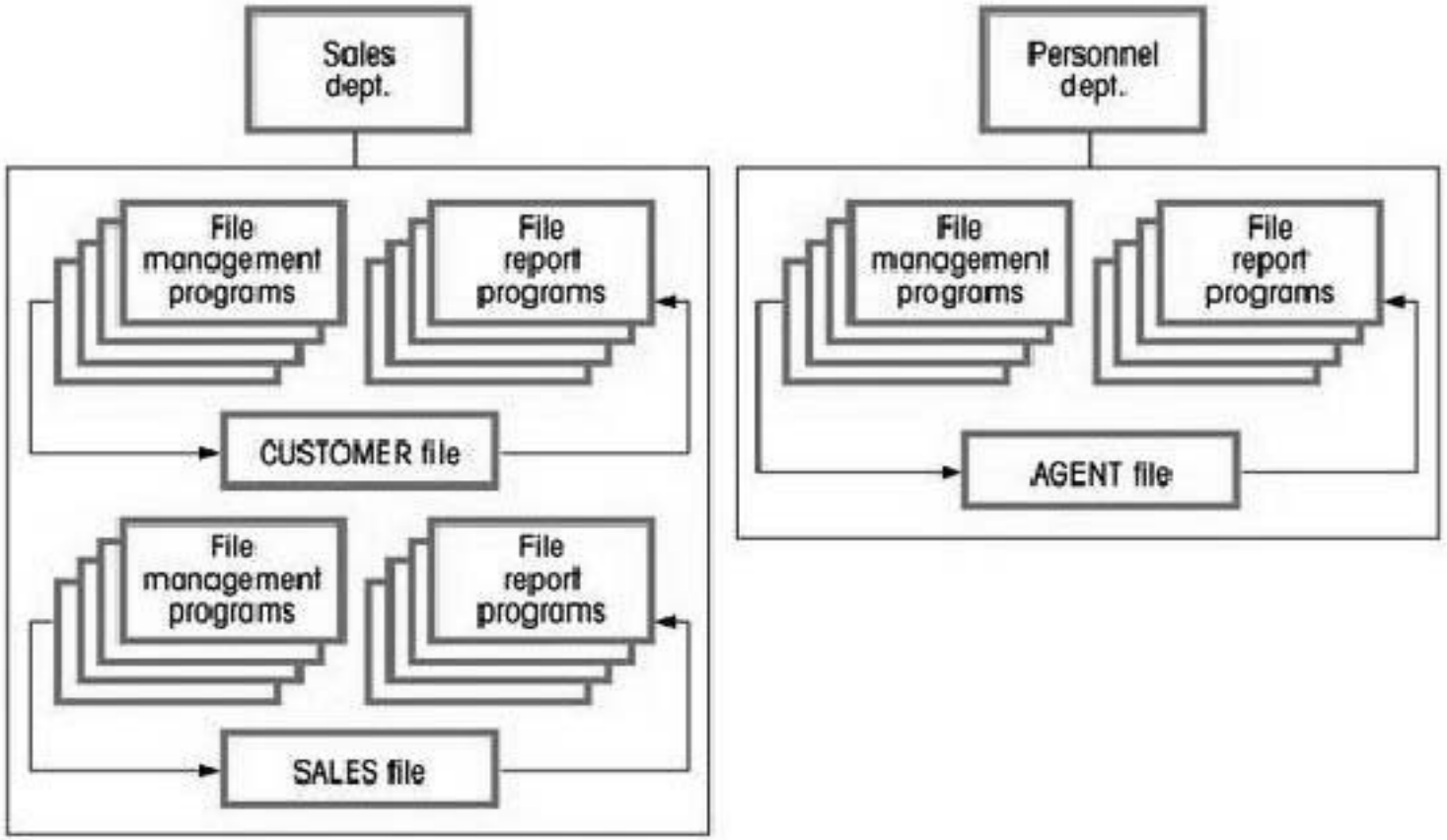
Data	"Raw" facts that have little meaning unless they have been organized in some logical manner. The smallest piece of data that can be "recognized" by the computer is a single character, such as the letter A, the number 5, or some symbol such as; ' ? > * +. A single character requires one byte of computer storage.
Field	A character or group of characters (alphabetic or numeric) that has a specific meaning. A field might define a telephone numbers, a birth date, a customer name, a year-to-date (YTD) sales value, and so on.
Record	A logically connected set of one or more fields that describes a person, place, or thing. For example, the fields that comprise a record for a customer named J. D. Rudd might consist of J. D. Rudd's name, address, phone number, date of birth, credit limit, unpaid balance, and so on.
File	A collection of related records. For example, a file might contain data about Company's vendors; or, a file might contain the records for the students currently enrolled at University.

Contents of the AGENT File

A_NAME	A_PHONE	A_ADDRESS	ZIP	HIRED	YTD_PAY	YTD_FIT	YTD_FICA	YTD_SLS	DEP
Alex B. Alby	713-228-1249	123 Toll, Nash, TN	37119	11/1/93	$20,586.24	$4,332.21	$1,534.57	$1,735.00	3
Leah F. Hahn	615-882-1244	334 Main, Fox, KY	25246	5/23/84	$25,213.76	$5,834.75	$1,788.36	$4,967.00	0
John T. Okon	615-123-5589	452 Elm, New, TN	36155	6/15/89	$23,198.29	$4,332.24	$1,689.44	$3,093.00	2

A_NAME = Agent name
A_PHONE = Agent phone
A_ADDRESS = Agent address
ZIP = Agent ZIP code
HIRED = Agent date of hire
YTD_PAY = Year-to-date pay
YTD_FIT = Year-to-date federal income tax paid
YTD_FICA = Year-to-date Social Security paid
YTD_SLS = Year-to-date sales, in thousands of $
DEP = Number of dependents

A Simple File System



File System Critique

- File System Data Management
  - Require extensive programming in a 3GL
  - System admini. becomes difficult, as the # of file expands
  - Difficult and important to make changes in existing file structure
  - Omit security feature to safeguard data
  - Island of data information

File System Critique

- Structural and Data Dependence
  - Structural Dependence
  - Data Dependence

- Data dependence makes file system cumbersome

File System Critique

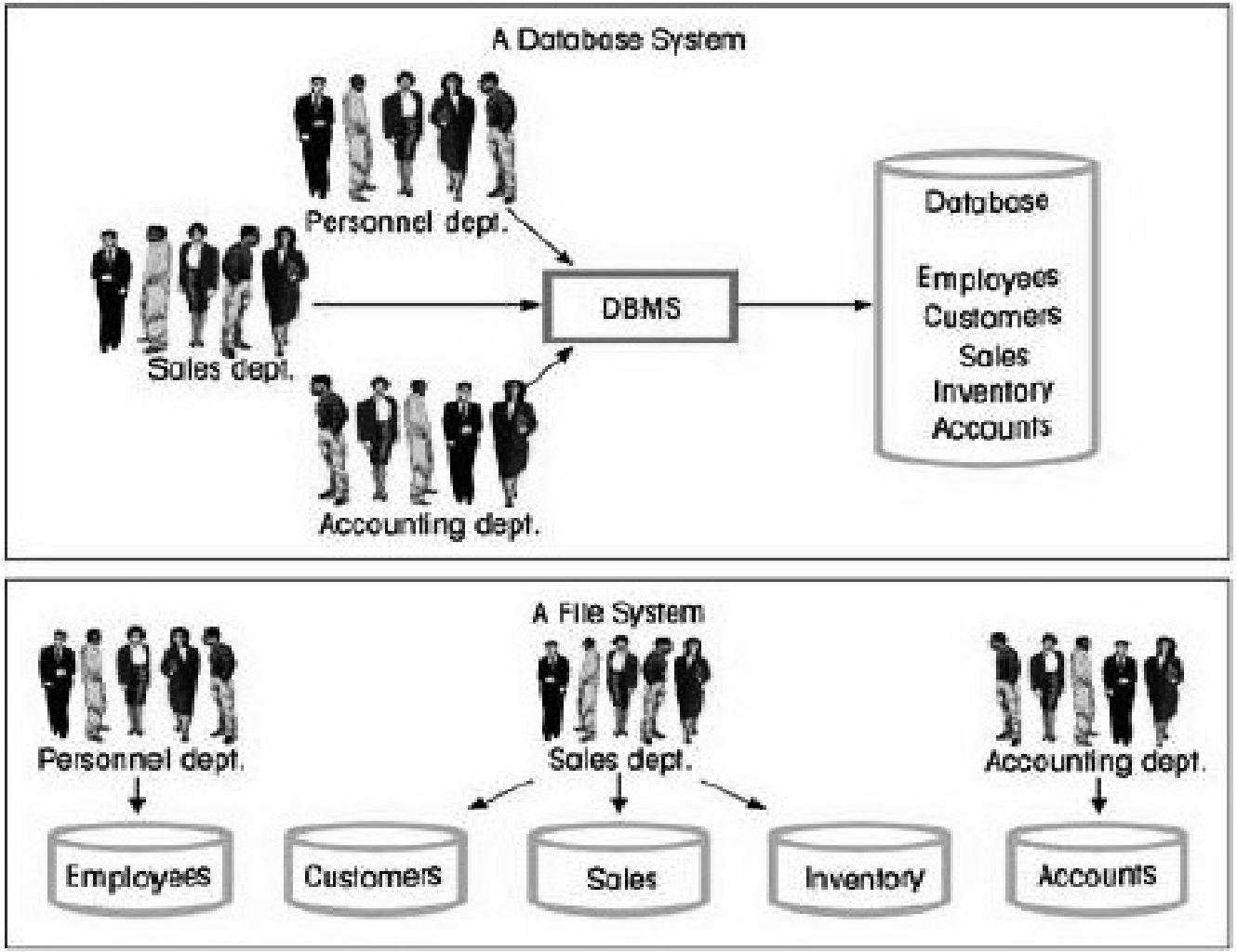
- Field Definitions and Name Conventions
  - A flexible (good) record definition anticipates reporting requirements by breaking up fileds into their components.
  - Example:
    - Customer Name ⇨ Last Name, First Name, Initial
    - Customer Address ⇨ Street Address, City, State

FIELD	CONTENTS
CUS_LNAME	Customer last name
CUS_FNAME	Customer first name
CUS_INITIAL	Customer initial
CUS_AREACODE	Customer area code
CUS_PHONE	Customer phone
CUS_ADDRESS	Customer street address or box number
CUS_CITY	Customer city
CUS_STATE	Customer state

File System Critique

- Selecting proper filed names is very important
  - descriptive within restrictions
  - Reflect documentation
- Data Redundancy
  - Data inconsistency (lack of data integrity)
  - Data anomalies
    - Modification anomalies
    - Insertion anomalies
    - Deletion anomalies

DB System Vs File System



DBMS Languages

- **Data Definition Language (DDL)**:
  - Used by the DBA and database designers to specify the *conceptual schema* of a database. In many DBMSs, the DDL is also used to define internal and external schemas (views). In some DBMSs, separate **storage definition language** (**SDL**) and **view definition language** (**VDL**) are used to define internal and external schemas.
  - Result of compiling DDL is **catalog** (a set of tables stored in a file)
- **Data Manipulation Language (DML)**: Used to specify database retrievals and updates.
  - High-level (nonprocedural, declarative) DML:
    - Describe what data is needed w/o specifying how to get it
    - DML commands (**data sublanguage**) can be *embedded* in a general-purpose programming language (**host language**), such as COBOL, PL/1 or PASCAL.
  - Low-level (procedural) DML:
    - Describe what & how
    - *stand-alone* DML commands can be applied directly (**query language**).

Data Model

- **Schema**: description of data at some level (*e.g.,* tables, attributes, constraints, domains)
- **Model**: tools and language for describing:
  - Conceptual and external schema
    - Data definition language (**DDL)**
  - Integrity constraints, domains (DDL)
  - Operations on data
    - Data manipulation language (**DML**)
  - Directives that influence the physical schema (affects performance, not semantics)
    - Storage definition language (**SDL**)

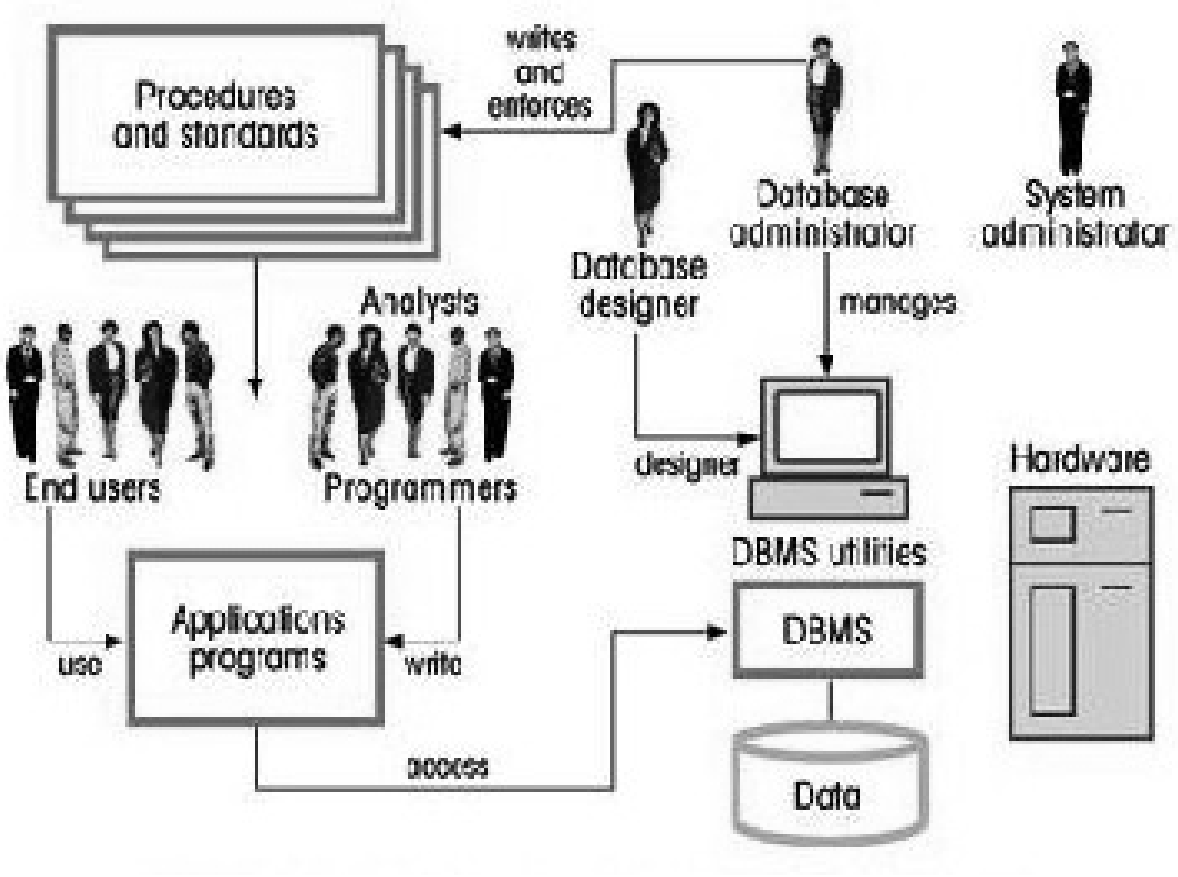
DBMS Interfaces

- Stand-alone query language interfaces
- Programmer interfaces for embedding DML in programming languages:
  - Pre-compiler Approach
  - Procedure (Subroutine) Call Approach
- User-friendly interfaces
  - Menu-based
  - Graphics-based (Point and Click, Drag and Drop etc.)
  - Forms-based
  - Natural language
  - Combinations of the above
  - Web Browser as an interface

DBMS Interfaces (Cont.)

- Parametric interfaces using function keys
- Report generation languages
- Interfaces for the DBA
  - Creating accounts, granting authorizations
  - Setting system parameters
  - Changing schemas or access path

Database System Environment



Database System Components

- Hardware
  - Computer
  - Peripherals
- Software
  - Operating system software
  - DBMS software
  - Applications programs and utilities software

Database System Components

- People
  - Systems administrators
  - Database administrators (DBAs)
  - Database designers
  - Systems analysts and programmers
  - End users
- Procedure
  - Instructions and rules
- Data
  - Collection of raw facts

Database System Components

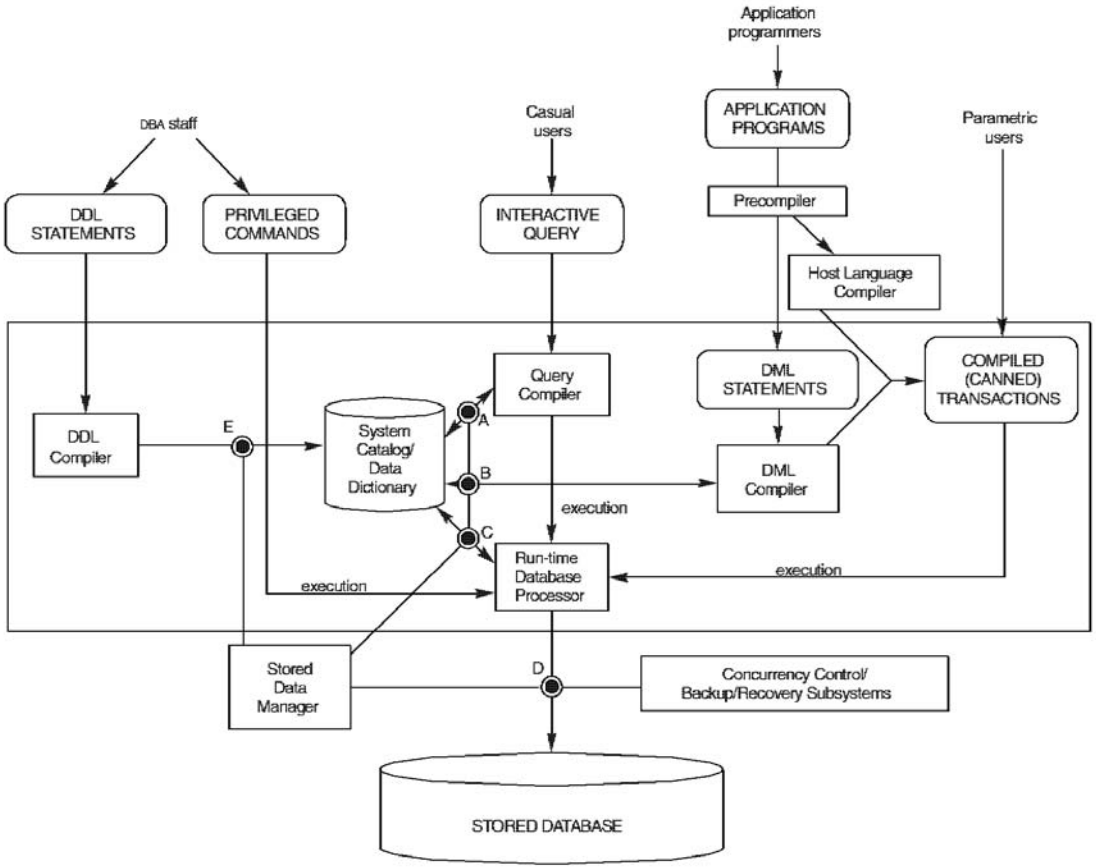
- Complexity of DB system depends on
  - Organization's size
  - Function
  - Corporate culture
  - Activities and environment
- DB solutions must be cost effective & strategically effective

DBMS Component Modules

- Stored data manager
- DDL compiler
- Run-time database processor
- Query compiler
- Pre-compiler

Typical DBMS Modules

Figure 2.3 Typical component modules of a DBMS. Dotted lines show accesses that are under the control of the stored data manager.



Database System Utilities

- *Loading* data stored in files into a database.
- *Backing up* the database periodically on tape.
- *Reorganizing* database file structures.
- *Report generation* utilities.
- *Performance monitoring* utilities.
- Other functions,
  - such as *sorting* , *user monitoring* , *security mgmt, data integrity, data compression*, etc.

Data dictionary / repository

- Used to store schema descriptions and other information such as design decisions, application program descriptions, user information, usage standards, etc.
- *Active* data dictionary is accessed by DBMS software and users/DBA.
- *Passive* data dictionary is accessed by users/DBA only

Types of Database Systems

- **Data model used**:
  - Traditional: Relational, Network, Hierarchical.
  - Emerging: Object-oriented, Object-relational, Temporal, Spatial
- **Other classifications:**
  - Single-user (typically used with micro- computers) vs. multi-user (most DBMSs).
  - Centralized (uses a single computer with one database) vs. distributed (uses multiple computers, multiple databases)
  - Use
    - Transactional (Production)
    - Decision support
    - Data warehouse

Database Models

- A set of logical constructs
  - Represent data structure & data relationships
- Two types of database models
  - Conceptual model
    - Logical nature of data representation
    - *What* is represented
  - Implementation model
    - *How* the data are represented
    - *How* the data structure are implemented

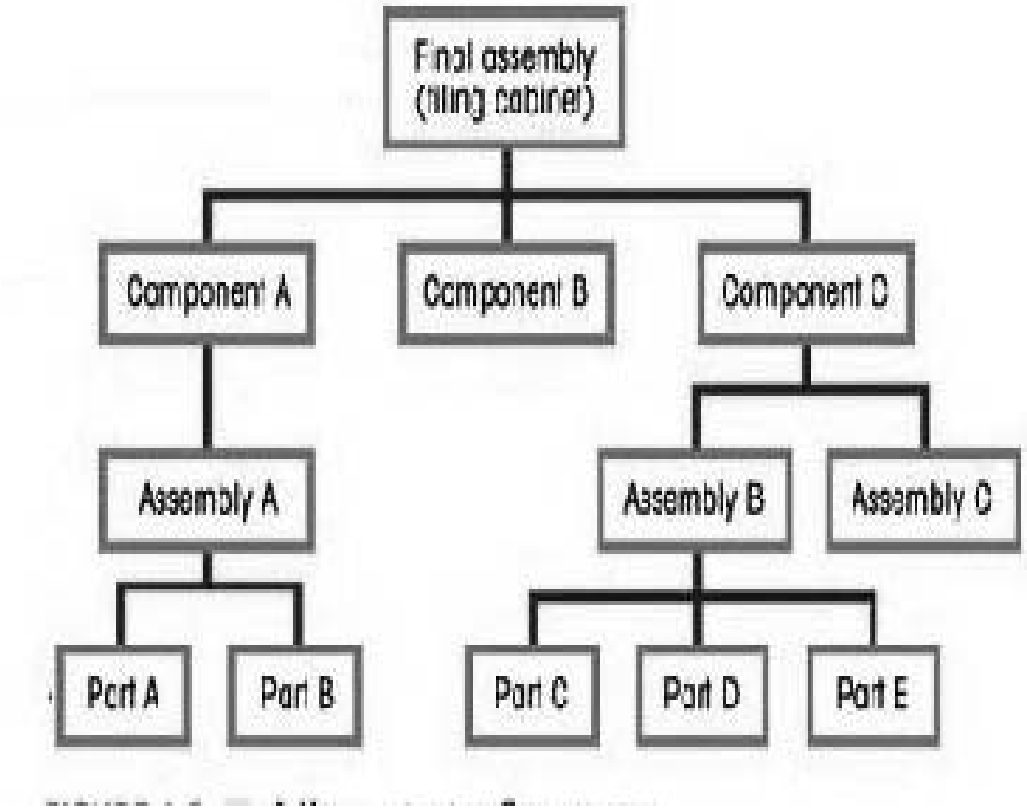
Database Models

- Three types of relationships
  - One-to-many (1:M)
    - PAINTER(1) *paints* PAINTING (M)
  - Many-to-many (N:M)
    - STUDENT(N) *takes* COURSE (M)
  - One-to-one (1:1)
    - EMPLOYER (1) *manages* STORE (1)

Implementation Database Models

- Hierarchical database model
- Network database model
- Relational database model

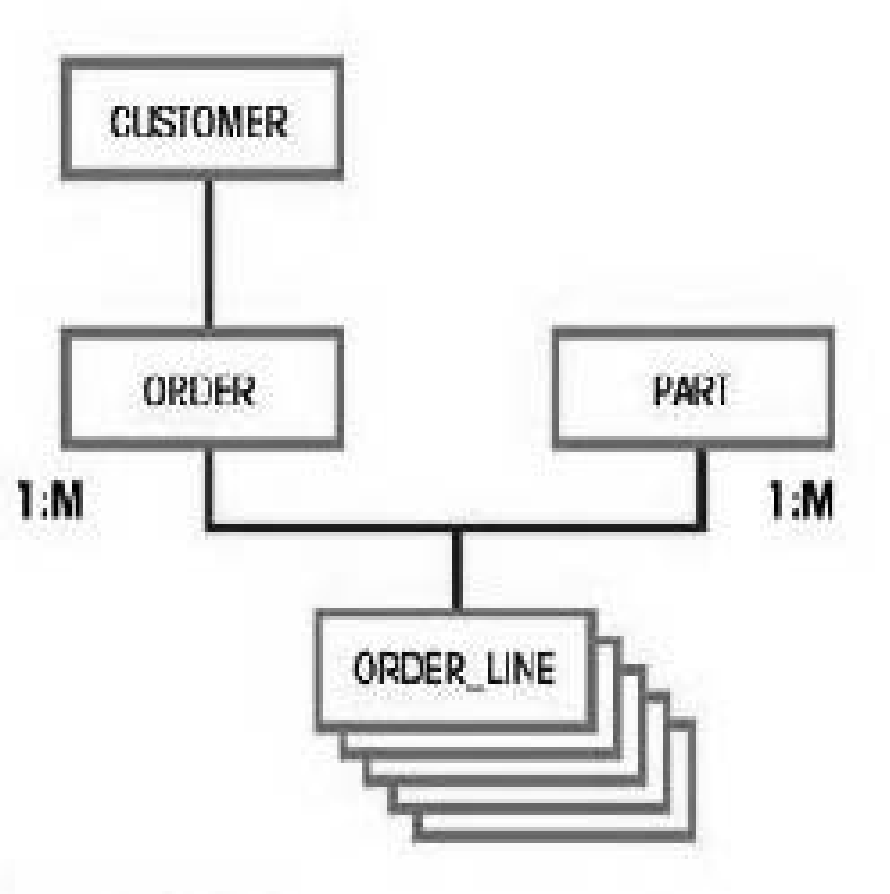
A Hierarchical Structure



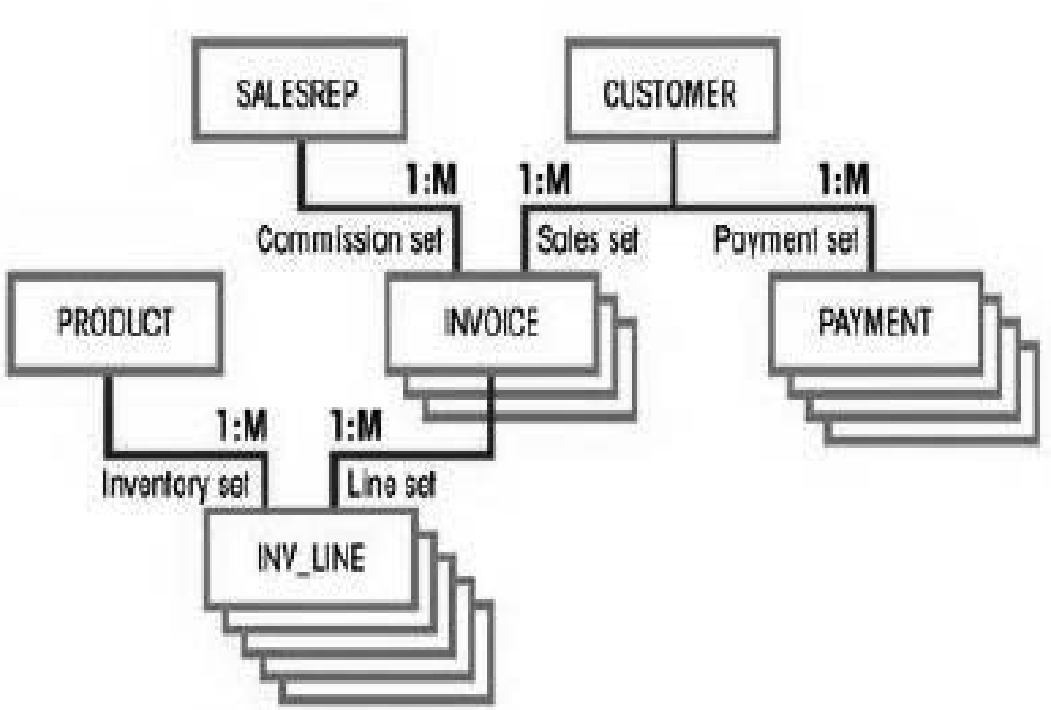
Hierarchical Data Model

- Pros
  - Conceptual simplicity
  - Database security
  - Data independence
  - Database integrity
  - Efficiency dealing with a large DB
- Cons
  - Complex implementation
  - Difficult to manage
  - Lacks structural independence
  - Applications programming and use complexity
  - Implementation limitation
  - Lack of standards

- Child with multiple parents



A Network Database Model



Network Database Model

- Pros
  - Conceptual simplicity
  - Handles more relationship types
  - Data access flexibility
  - Data independence
  - Conformance to standards
- Cons
  - System complexity
  - Lack of structural independence

Relational Database Model

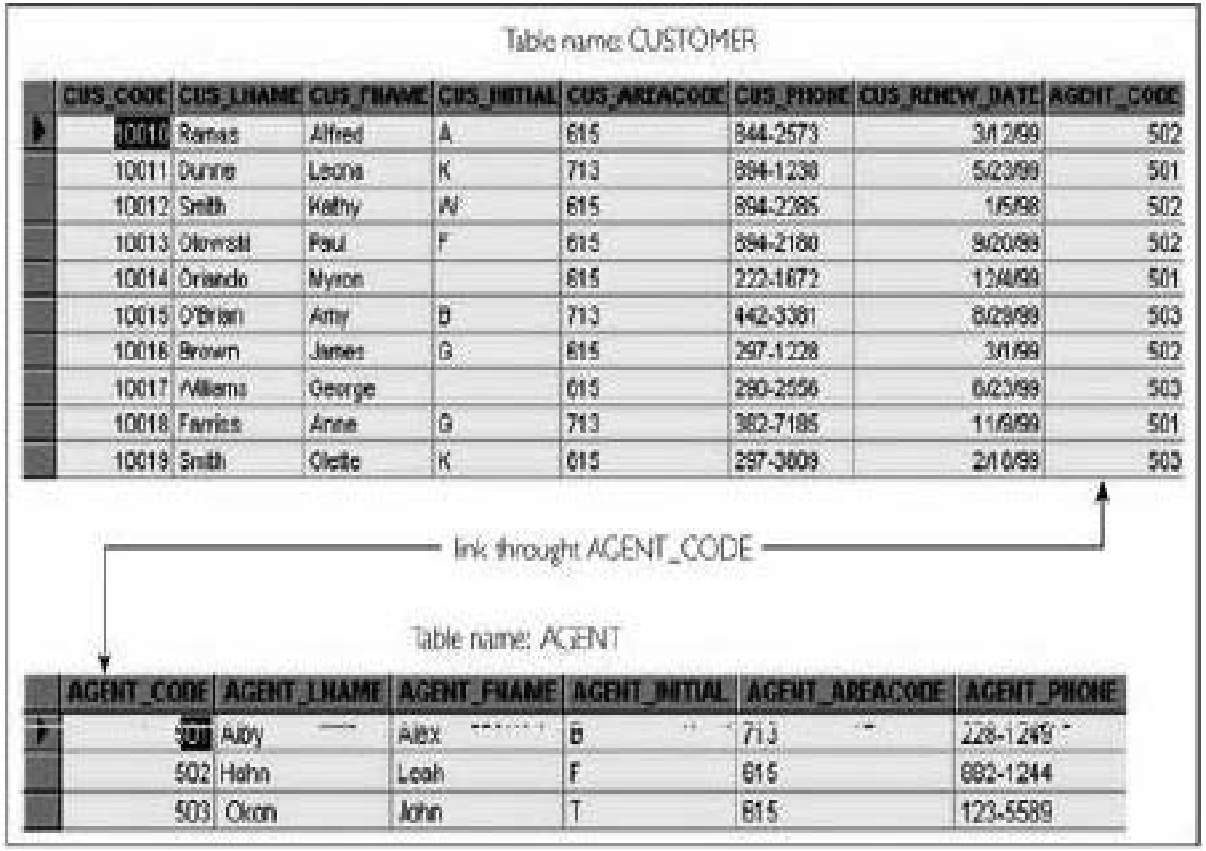


Table name: CUSTOMER

CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_AREACODE	CUS_PHONE	CUS_RENEW_DATE	AGENT_CODE
10010	Ramas	Alfred	A	615	844-2573	3/12/99	502
10011	Dunne	Leona	K	713	894-1238	5/23/99	501
10012	Smith	Kathy	W	615	894-2285	1/5/98	502
10013	Olowski	Paul	F	615	894-2180	9/20/99	502
10014	Orlando	Myron		615	222-1672	12/4/99	501
10015	O'Brian	Amy	B	713	442-3381	8/29/99	503
10016	Brown	James	G	615	297-1228	3/1/99	502
10017	Williams	George		615	290-2556	6/23/99	503
10018	Farriss	Anne	G	713	382-7185	11/8/99	501
10019	Smith	Olette	K	615	297-3809	2/10/99	503

link through AGENT_CODE

Table name: AGENT

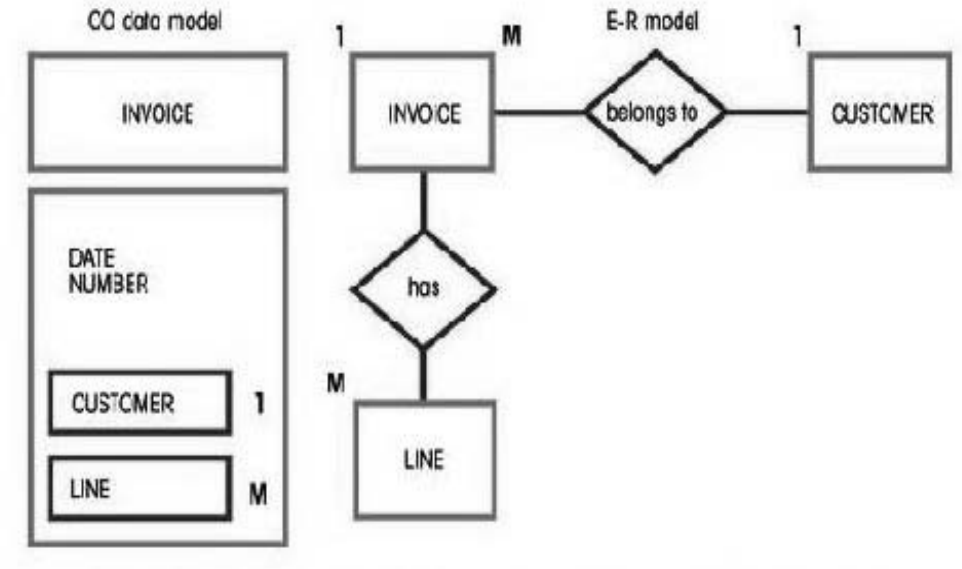
AGENT_CODE	AGENT_LNAME	AGENT_FNAME	AGENT_INITIAL	AGENT_AREACODE	AGENT_PHONE
501	Alby	Alex	B	713	228-1249
502	Hahn	Leah	F	615	882-1244
503	Okon	John	T	615	123-5589

Relational Database Model

- Pros
  - Structural independence
  - Improved conceptual simplicity
  - Easier database design, implementation, mgmt, and use
  - Ad hoc query capability (SQL)
  - Powerful DBMS
- Cons
  - Hardware and software overhead
  - Possibility of poor design and implementation
  - Potential “islands of information” problems

Object-Oriented Database Model

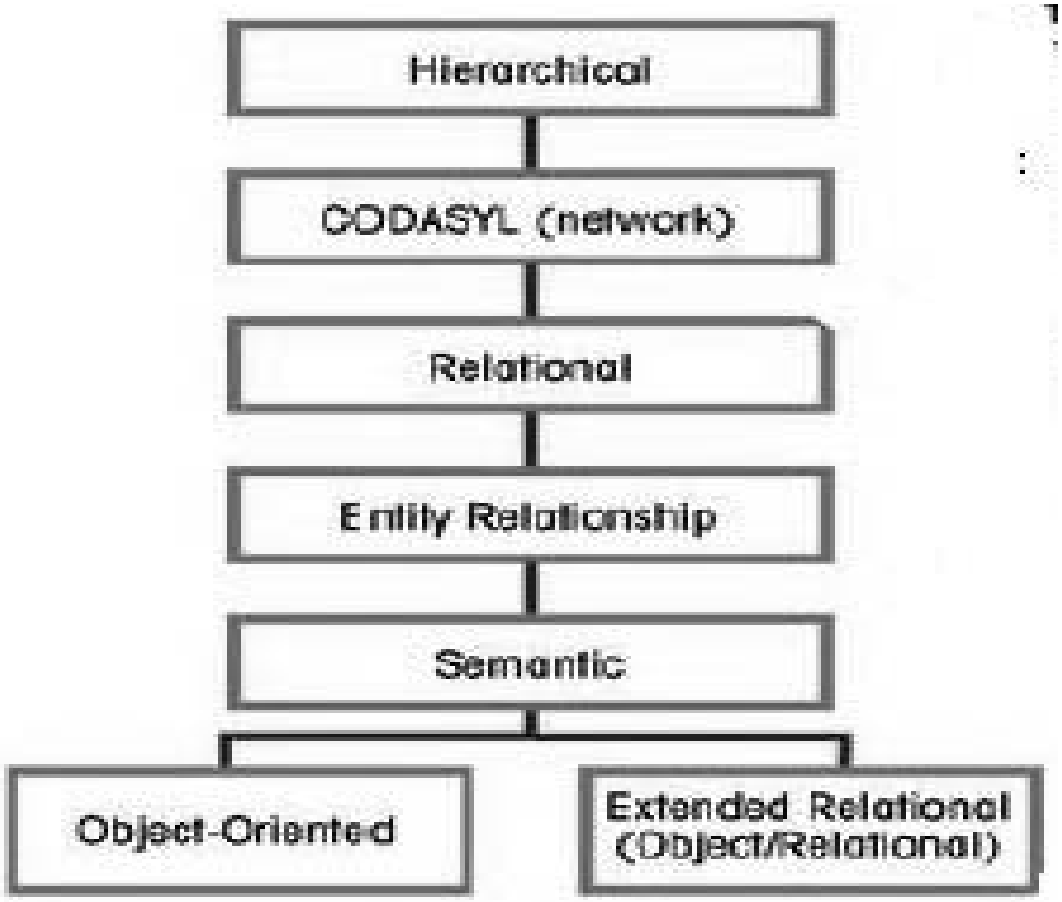
- Basic Structure
  - Objects
  - Attribute
  - Class
- Comparison of OO data model and ER data model

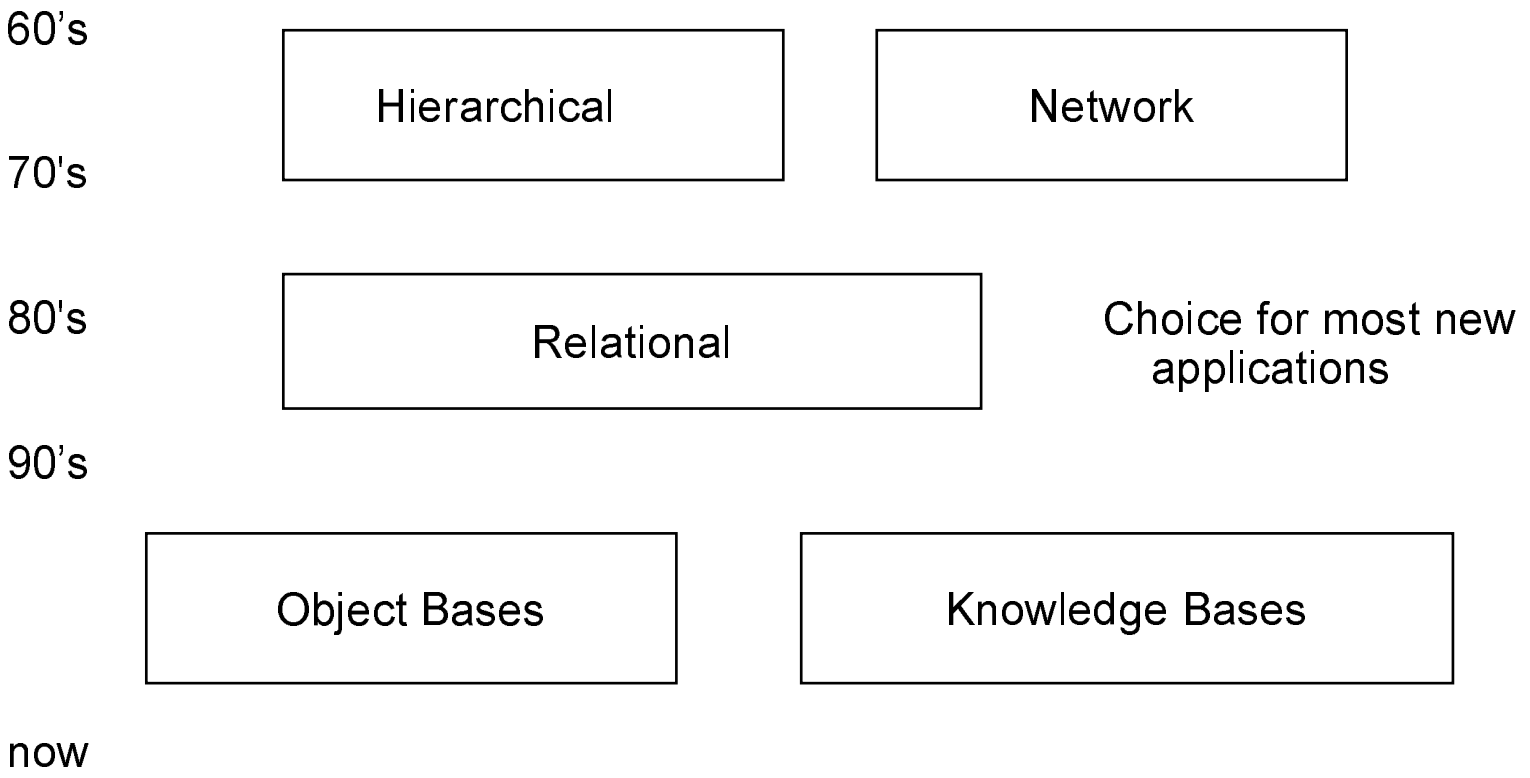


OO Database Model

- Pros
  - Add semantic content
  - Visual presentation includes semantic content
  - Database integrity
  - Both structural and data independence
- Cons
  - Lack of OODM standards
  - Complex navigational data access
  - High system overhead slows transaction

The Development of Data Model




60’s
Hierarchical
Network
70's
80's
Relational
Choice for most new applications
90’s
Object Bases
Knowledge Bases
now

Wrap-up: The Evolution of Data Models

- Common characteristics required for data models
  - Some degree of conceptual simplicity
  - Represent the real word as close as possible
  - Representation of mini-word behavior must be in compliance with the consistency and integrity of any data model

- Database models and the internet
  - Flexible, efficient, and secure internet access
  - Support for complex data types and relationships
  - Seamless interfacing with multiple data sources and structures
  - Simplicity of conceptual database model
  - An abundance of available database tools
  - A powerful DBMS makes DBA's job easier