Overview of Data Management

School of Computer Science
University of Waterloo

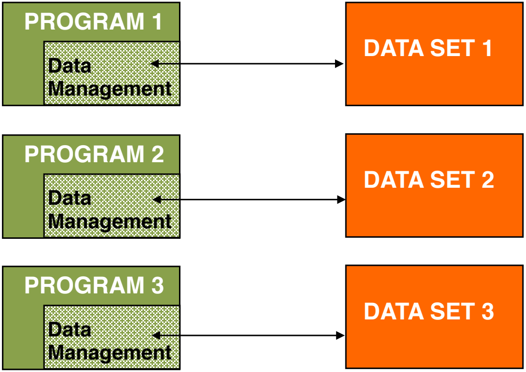
Databases CS348

What is "Data"

- ANSI definition of data:
  1. A representation of facts, concepts, or instructions in a formalized manner suitable for communication, interpretation, or processing by humans or by automatic means.
  2. Any representation such as characters or analog quantities to which meaning is or might be assigned. Generally, we perform operations on data or data items to supply some information about an entity.
- Volatile vs persistent data
  - Our concern is primarily with persistent data

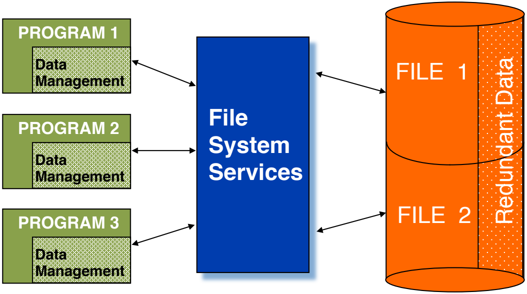
Early Data Management – Ancient History

- Data stored on magnetic tapes.
- One data set per program. High data redundancy

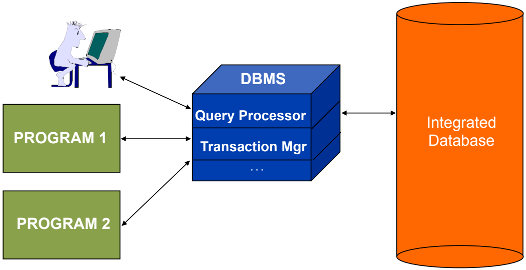


File Processing – More Recent History

- Data are stored in files located on disk drives with a file system interface between programs and files.
- Various access methods exist (e.g., sequential, indexed, random).
- One file used by to one or more programs.



Database Approach



What is a Database

Definition (Database)

A *large* and *persistent* collection of factual data and metadata organized in a way that facilitates efficient *retrieval* and *revision*.

Example of factual data: *John's age is 42.*

Example of metadata: *There is a concept of an employee that has a name and an age.*

What is a Data Model

Definition (Data Model)

A *data model* determines the nature of the metadata and how retrieval and revision is expressed.

Examples of databases:

- a file cabinet
- a library system
- an inventory control system

Definition (Database Management System (DBMS))

A program (or set of programs) that implements a data model.

Database Management System (DBMS)

Idea

Abstract common functions and create a uniform well defined interface for applications that require a database.

1. Supports an underlying data model
   (all data stored and manipulated in a well defined way)
2. Access control
   (various data can be accessed or revised only by authorized people)
3. Concurrency control
   (multiple concurrent applications can access data)
4. Database recovery
   (reliability; nothing gets accidentally lost)
5. Database maintenance (e.g., revising metadata)

Schema and Instance

Definition (Schema)

A *database schema* is a collection of metadata conforming to an underlying data model.

Definition (Instance)

A *database instance* is a collection of factual data as defined by a given database schema.

A schema can (and typically does) have many possible database instances.

Example – A Relational Database

- Schema
  - EMP(ENO, ENAME, TITLE)
  - PROJ(PNO, PNAME, BUDGET)
  - WORKS(ENO, PNO, RESP, DUR)
- Instance

EMP

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E7	R. Davis	Mech. Eng.
E8	J. Jones	Syst. Anal.

PROJ

PNO	PNAME	BUDGET
P1	Instrumentation	150000
P2	Database Develop.	135000
P3	CAD/CAM	250000
P4	Maintenance	310000

WORKS

ENO	PNO	RESP	DUR
E1	P1	Manager	12
E2	P1	Analyst	24
E2	P2	Analyst	6
E3	P3	Consultant	10
E3	P4	Engineer	48
E4	P2	Programmer	18
E5	P2	Manager	24
E6	P4	Manager	48
E7	P3	Engineer	36
E8	P3	Manager	40

Apps that use a DBMS

Longstanding

- inventory control
- payroll
- banking and financial systems
- reservation systems

More recent

- computer aided design (CAD)
- software development (CASE, SDE/SSE)
- telecommunication systems
- e-commerce
- dynamic/personalized web content

Brief History of Data Management: 1970s

- Edgar Codd proposes relational data model (1970)
  - firm mathematical foundation → *declarative* queries
- *Charles Bachman wins ACM Turing award (1973)*
  - "The Programmer as Navigator"
- Peter Chen proposes E-R model (1976)
- Transaction concepts (Jim Gray and others)
- IBM's System R and UC Berkeley's Ingres systems demonstrate feasibility of relational DBMS (late 1970s)

Brief History of Data Management: 1980s

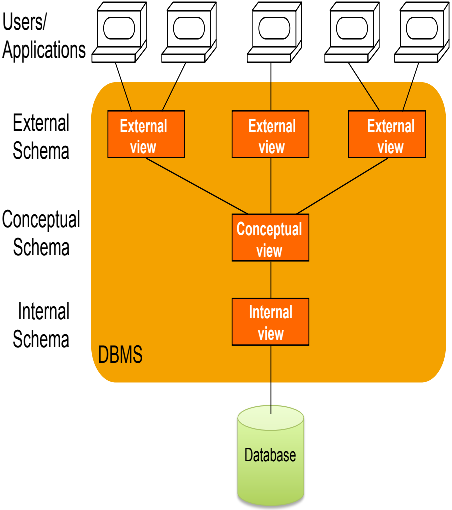
- Development of commercial relational technology
  - IBM DB2, Oracle, Informix, Sybase
- *Edgar Codd wins ACM Turing award (1981)*
- SQL standardization efforts through ANSI and ISO
- Object-oriented DBMSs (late 1980's to middle of 1990's)
  - persistent objects
  - object id's, methods, inheritence
  - navigational interface reminicent of hierarchical model

Brief History of Data Management: 1990s-Present

- Continued expansion of SQL and system capabilities
- New application areas:
  - the Internet
  - On-Line Analytic Processing (OLAP)
  - data warehousing
  - embedded systems
  - multimedia
  - XML
  - data streams
- *Jim Gray wins ACM Turing award (1998)*
- Relational DBMSs incorporate objects (late 1990s)
- Many new players in the DB industry (2000+)
- *Michael Stonebraker wins ACM Turing award (2014)*

Three Level Schema Architecture

1. External schema (view): what the application programs and user see. May differ for different users of the same database.
2. Conceptual schema: description of the logical structure of *all* data in the database.
3. Physical schema: description of physical aspects (selection of files, devices, storage algorithms, etc.)



Data Independence

Idea

Applications do not access data directly but, rather through an abstract data model *provided by the DBMS.*

Two kinds of data independence:

Physical: applications immune to changes in storage structures

Logical: modularity:

The `WAREHOUSE` table is not accessed by the payroll app; the `EMPLOYEE` table is not accessed by the inventory control app.

Note

One of the most important reasons to use a DBMS!

Transactions

When multiple applications access the same data, undesirable results occur.

Example:

```
withdraw(AC,1000)         withdraw(AC,500)
  Bal := getbal(AC)
                            Bal := getbal(AC)
  if (Bal>1000)             if (Bal>500)
    <give-money>              <give-money>
    setbal(AC,Bal-1000)

                              setbal(AC,Bal-500)
```

Idea

Every application may think it is the sole application accessing the data. The DBMS should guarantee correct execution.

Transactions (cont'd)

Definition (Transaction)

An application-specified atomic and durable unit of work.

Properties of transactions ensured by the DBMS:

Atomic: a transaction occurs entirely, or not at all

Consistency: each transaction preserves the consistency of the database

Isolated: concurrent transactions do not interfere with each other

Durable: once completed, a transaction's changes are permanent

Interfacing to the DBMS

Data Definition Language (DDL): for specifying schemas

- may have different DDLs for external schema, conceptual schema, physical schema

Data Manipulation Language (DML): for specifying retrieval and revision requests

- **navigational** (procedural)
- **non-navigational** (declarative)

Types of Database Users

End user:

- Accesses the database indirectly through forms or other query-generating applications, or
- Generates ad-hoc queries using the DML.

Application developer:

- Designs and implements applications that access the database.

Database administrator (DBA):

- Manages conceptual schema.
- Assists with application view integration.
- Monitors and tunes DBMS performance.
- Defines internal schema.
- Loads and reformats database.
- Is responsible for security and reliability.

Summary

Using a DBMS to manage data helps:

- to remove common code from applications
- to provide uniform access to data
- to guarantee data integrity
- to manage concurrent access
- to protect against system failure
- to set access policies for data