CSC 310
Database Theory and Implementation

Blase B. Cindric
University of Mount Union
General Architecture of a Database System

Definitions

*Database (DB) →*

an integrated collection of related data

*Database Management System (DBMS) →*

a collection of programs that control the database

*Database System (DBS) →*

consists of DB + DBMS + Application programs
More Terminology

Integrated collection of related data

*integrated* → all data stored in a uniform manner on secondary storage

*related data* → data describing differing things, people, concepts, along with the connections between these things
Example: Information about College Registration

- Students
- Courses
- Faculty Members

- Students taking Courses
- Faculty teaching Courses
- Faculty advising Students

All this can be organized in a single DB
A Simplified DBS Organization

User

DBS

.Application Programs

Query and Transaction Processing

Management of Stored Data

DBMS

Catalog Meta-Data

Database

CSC 310 DB Theory & Impl.
Advantages & Disadvantages of DBMSs vs. File Systems (Sec 1.6)

Advantages:

- **Reduced Data Redundancy**
  - duplicate copies of data values not stored unless necessary
- **Data Consistency**
  - whenever there is duplication, ensures that all copies are consistent
- **Integration of Data**
  - more information obtained from same amount of data when it is integrated in a DB
- **Data Sharing within an organization**
  - any authorized user can access the entire database
- **Integrity Enforcement**
  - data must satisfy integrity constraints to be stored in the DB
    - ex.: annual income on W-2 form must be $\geq 0$
- **Improved Security of Data**
  - supports access to data only by authorized users
    - different levels of access for different classes of users
Advantages & Disadvantages of DBMSs vs. File Systems (Sec 1.6)

Advantages (continued):

Enforcement of Organizational Standards
- departmental, national, international data formats, rules, etc.

Economy of Scale
- cost savings with one integrated database system vs. multiple file systems with differing formats, languages, etc.

Improved Data Accessibility and Responsiveness
- special DB languages allow *ad hoc* queries to gather data from different sources, all from the integrated DB

Concurrency Control
- many users are allowed to access the same data at the same time; the system ensures that conflicting requests are handled

Data Backup and Recovery
- data are not lost when the system fails for whatever reason
Advantages & Disadvantages of DBMSs vs. File Systems (Sec 1.6)

Disadvantages of DBMSs:

More Complexity
users and designers must understand DBMS to make full use of all system capabilities (to maximize investment in the system)

Cost
large, mainframe-based multi-user DBMSs can cost upwards of $1,000,000 or more, plus annual maintenance costs approaching 6 figures

Possible additional hardware costs
secondary storage may need to be increased vs. file system needs to store catalog meta-data

Cost of Conversion
up-front investment will be worthwhile in the long run, when advantages of DBMS are revealed over time

Higher Impact of Failure
if system goes down, no user can perform work; in a file system, only the application that goes down is affected

Experience has proven that the advantages of DBMSs far outweigh the disadvantages, especially as the size of the organizational data collection increase
Database Design / Data Modeling

Problem

Portion of Real World of Interest

DBMS

Data Model

Schema

This is the activity called Modeling/Designing CSC 310 DB Theory & Impl.
**Database Design**

→ specifying the schema of a database in a given data model

**Schema of a database**

→ the structure of a database that:
  - captures data types, relationships and constraints on the data;
  - is independent of any application program; and
  - changes infrequently

**Instance of a database**

→ the actual data in the DB at a particular instant in time (the DB contents)
More Terms from Database Design / Data Modeling

Data Model

→ a method for organizing a DB that consists of:
  • a set of primitives for defining the structure of a DB; and
  • a set of operations for specifying retrievals and updates on a database
Categories of Database Design

**Conceptual Database Design**

- an abstract but complete description of the database
  - thinking about data to be stored (the Miniworld)
  - identifying objects/entities and relationships among them
  - expressing them in a pseudo-formal notation (conceptual data model)

example: E-R model

**Logical Database Design**

- transforming the initial conceptual design into a formal schema expressed in an implementation data model

example: Hierarchical, Network, Relational, Object-Oriented
Categories of Database Design

Physical Database Design

specifying access methods and the internal storage organization of objects of the implementation schema

often called the *internal* or *physical data model*

example: $B^+$ trees, 2-3-4 trees, limit-modulated graphs
Three-Schema Architecture

EXTERNAL LEVEL

CONCEPTUAL LEVEL

INTERNAL LEVEL

End Users

External View 1

External View n

Conceptual Schema

Internal Schema

Stored DB

CSC 310 DB Theory & Impl.
Definitions from the Three Schema Architecture

*Internal Schema*

describes the physical storage structure of the DB

*Conceptual Schema*

describes the structure of the whole DB in terms of the data model used (hiding the details of the physical storage structure)

*External Schema*

describes the DB structure as seen from the point of view of a user
typically limited to a subset of the overall conceptual schema
based on the user’s need-to-know
Database Languages (Sec. 2.2)

Database users are concerned with obtaining results, and are not necessarily programmers as such, DB command languages are designed to enable users to describe what results are to be generated by the command, but not to specify how to obtain those results

- as opposed to traditional programming languages (C++, Java, C#, etc.), where programming is defining how to get results from a DB
Database Language Example

Retrieve a list of all students from Wyoming, sorted in alphabetical order by last name

- C++: must have multiple loops to search and sort data
- SQL:

  SELECT * FROM Students
  WHERE State = "WY"
  ORDER BY LastName;
4GLs

Database Languages are often called *Fourth-Generation Languages*, or 4GLs

1GL: Machine Language for specific CPU
2GL: Assembly Language for specific CPU
3GL: High-Level Programming Language (COBOL, Fortran, C++, …)
4GL: Database Languages (SQL, QBE, RPL, …)
A DB Language consists of two major parts:

- **Data Definition Language (DDL)**
  specifies DB structure, data types, security settings, etc.
- **Data Manipulation Language (DML)**
  used for retrieving information and updating database content
Data Definition Language
(Sec. 2.2.1)

DDL commands are used to define the DB schema and make modifications to an already existing schema.

DB schema is stored in the Catalog contains Meta-Data --- information about the data in the DB.

Catalog contains DB table definitions, view defs., usage statistics, ....
Data Definition Language

Example: for Students at a University,

- # of individual values stored for each student
- max length of variable-sized values (text strings)
- size of numeric values (significant digits, decimal places)
- uniquely-identifying values (Student #, SS#)
- integrity constraints (0.0 < GPA < 4.0)
- user groups with authorized access to Students (read, insert new, delete, update)
Data Manipulation Language
(Sec. 2.2.2)

DML commands provide for storage and retrieval of data to/from the DB

There are two subsets of DML:

1) Database Maintenance commands
   for storage of data
   insert new data, delete unneeded data, update

2) Database Query commands
   for retrieval of data
   typically non-procedural (concerned with what is to be retrieved, not how it is to be done)
   4GLs provide search, sort, and control-break logic as built-in operations
Database Languages

In this course, we will focus on the most popular DB language for Relational DBMSs (the data model most widely used in the world today):

SQL (Structured Query Language)
Essential Functions of a DBMS (Sec. 2.4)

To be considered competitive with other products in the market, a modern DBMS must provide the following ten services:

1) Data storage, retrieval and update
   a fully-functional DML is needed

2) Catalog that is modifiable by users

3) Transaction support
   a DB transaction is a sequence of individual DB operations that represent one real-world operation
   ex. ATM withdrawal: involves a balance lookup and possibly a modification of balance
Essential Functions of a DBMS (continued)

4) Mechanism to control *concurrent DB access* by multiple users
   must avoid conflicts (two simultaneous updates to same DB value)

5) Recovery Services
   backups, roll-backs, transaction logs

6) User Authentication
   allowing only authorized users to access the DB
Essential Functions of a DBMS (continued)

7) Support for Data Communication

8) Integrity Enforcement
   two types:
   • application-specific
   • data-model-required
Essential Functions of a DBMS (continued)

9) Support of Data Independence
   physical details of storage structure are hidden from users and programs

10) Utility Services
    import/export, usage statistics, memory management, ....