Databases and Database Users

406.426 Design & Analysis of Database Systems

Jonghun Park

jonghun@snu.ac.kr

Dept. of Industrial Engineering

Seoul National University

outline

- types of databases and database applications
- basic definitions
- typical DBMS functionality
- example of a database (UNIVERSITY)
- main characteristics of the database approach
- database users
- advantages of using the database approach
- when not to use databases

types of databases and database applications

- traditional databases: numeric and textual databases
- multimedia databases
- geographic Information Systems (GIS)
- data warehouses
- online analytical processing (OLAP)
- real-time and active databases
- web databases
- …

databases are everywhere

basic definitions

- data: known **facts** that can be recorded and have implicit meaning
- database (DB): a collection of **related** data
- implicit properties of a DB
  - represents some **aspect** of the real world: miniworld, UoD
  - is a **logically coherent** collection of data with some inherent meaning
  - is designed, built, and populated with data for a specific **purpose**

basic definitions

- **database management system (DBMS)**
  - a **collection of programs** that enables users to create and maintain a database
  - a general-purpose **software system** that facilitates the processes of **defining**, **constructing**, **manipulating**, **protecting**, **maintaining**, and **sharing** databases among various users and applications

basic definitions

- **database system**: database + DBMS

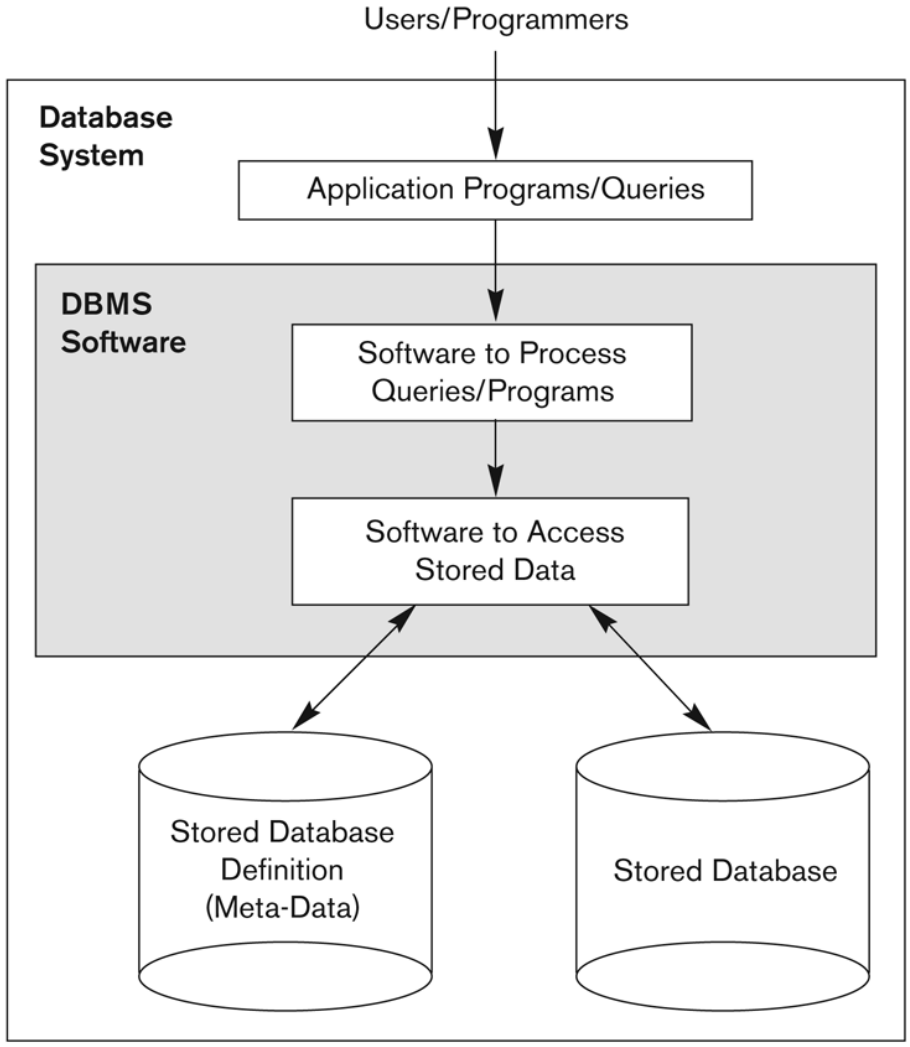

Figure 1.1
A simplified database system environment.

example of a DB

- mini-world for the example:
  - part of a UNIVERSITY environment
- some mini-world **entities**:
  - STUDENTs
  - COURSEs
  - SECTIONs (of COURSEs)
  - (academic) DEPARTMENTs
  - INSTRUCTORs

example of a DB

- Some mini-world relationships:
  - SECTIONs are of specific COURSEs
  - STUDENTs take SECTIONs
  - COURSEs have prerequisite COURSEs
  - INSTRUCTORs teach SECTIONs
  - COURSEs are offered by DEPARTMENTs
  - STUDENTs major in DEPARTMENTs
- the above entities and relationships are typically expressed in a conceptual data model, such as the **ENTITY-RELATIONSHIP data model** (Chapters 3, 4)

example of a simple database

define
construct
manipulate
…

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	04	King
92	CS1310	Fall	04	Anderson
102	CS3320	Spring	05	Knuth
112	MATH2410	Fall	05	Chang
119	CS1310	Fall	05	Anderson
135	CS3380	Fall	05	Stone

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Figure 1.2
A database that stores student and course information.

characteristics of the DB approach

- traditional file processing
  - each user defines and implements the **files needed for a specific software application** as part of programming the application
  - each user maintains separate files and programs
- DB approach
  - a **single repository of data** is maintained that is defined once and then is accessed by various users
- main characteristics of the DB approach
  - **self-describing** nature of a database system
  - **insulation** between programs and data, and data abstraction
  - support of **multiple views** of the data
  - **sharing** of data and multi-user transaction processing

self-describing nature of a database system

- catalog (aka metadata)
  - contains information such as the **structure** of each file, the **type** and storage format of each data item, and various **constraints** on the data

RELATIONS

Relation_name	No_of_columns
STUDENT	4
COURSE	4
SECTION	5
GRADE_REPORT	3
PREREQUISITE	2

COLUMNS

Column_name	Data_type	Belongs_to_relation
Name	Character (30)	STUDENT
Student_number	Character (4)	STUDENT
Class	Integer (1)	STUDENT
Major	Major_type	STUDENT
Course_name	Character (10)	COURSE
Course_number	XXXXNNNN	COURSE
….	….	…..
….	….	…..
….	….	…..
Prerequisite_number	XXXXNNNN	PREREQUISITE

Note: Major_type is defined as an enumerared type with all known majors. XXXXNNNN is used to define a type with four alpha characters followed by four digits

Figure 1.3
An example of a database catalog for the database in Figure 1.2.

insulation between programs and data

- in traditional file processing, the structure of data files is **embedded** in the application programs
  - need for **program-data independence**, **program-operation independence**
- **data abstraction**: the characteristic that allows the independence
- **conceptual representation** of data
  - does not include many of the details of how the data is **stored** or how the operations are **implemented**
- **data model**
  - a type of data abstraction that is used to provide the **conceptual representation**
  - hides the storage and implementation details

support of multiple views of the data

- view
  - may be a **subset** of the DB or may contain **virtual data** that is derived from the database files but is **not explicitly stored**

(a)

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

(b)

PREREQUISITES	CourseName	CourseNumber	Prerequisites
	Database	CS3380	CS3320
			MATH2410
	Data Structures	CS3320	CS1310

sharing of data and multi-user TP

- a multi-user DBMS must allow **multiple users** to access the DB at the **same time**
- concurrency control
  - to ensure that several users trying to update the same data do so in a controlled manner so that the **result of the update is correct**
  - example from an OLTP application: concurrent requests for **airline reservation**
- transaction
  - an executing program or process that includes **one or more database accesses**, such as reading or updating of database records
  - ACID properties

DB users

- database administrators:
  - responsible for **authorizing** access to the database, for **coordinating** and **monitoring** its use, **acquiring** software, and hardware resources, **controlling** its use and **monitoring** efficiency of operations
- database designers:
  - responsible for **identifying the data** to be stored in the DB and for choosing appropriate **structures** to represent and store the data
  - interact with each potential group of users and **develop views** of the database that meet the data and processing requirements of these groups
- end-users
  - use the data for queries, reports and some of them actually update the database content
- software analysts
  - determine the **requirements** of end users, especially naïve and parametric end users, and develop specifications for canned transactions that meet these requirements
- application programmers
  - implements the specifications as programs

categories of end-users

- casual: access database occasionally when needed
- naïve or parametric:
  - make up a **large portion** of the end-user population
  - use standard types of queries and updates in the form of "canned transactions" against the database
  - examples: bank-tellers
- sophisticated:
  - include business analysts, scientists, engineers, others
  - implement their applications to meet their complex requirements
- stand-alone:
  - maintain personal databases using ready-to-use packaged applications
  - example: a tax program user

workers behind the scene

- typically do not use the DB for their own purposes
- DBMS system designers and implementers
  - design and implement the DBMS modules and interfaces as a software package
- tool developers
  - design and implement the software packages that **facilitate database system design and use** and that help improve **performance**
  - e.g., DB design, performance monitoring, graphical interfaces,…
- operators and maintenance personnel
  - responsible for the actual running and maintenance of the hardware and software environment for the DB system

advantages of using the database approach

- controlling **redundancy**
- restricting unauthorized access: **security**, **authorization**
- providing **persistent storage** for program objects
- providing storage structures for efficient **query processing**: indexes, tree data structures, hash structures
- providing **backup** and **recovery**
- providing multiple **user interfaces**: query languages, APIs, forms, menu-driven interfaces, NL interfaces, GUIs, Web…
- representing and managing **complex relationships** among data
- enforcing **integrity constraints** that hold for the data
- permitting **inference** and **actions** using rules: deductive database, active database (event-driven)
- additional implications

controlling redundancy

- the problems with the redundancy in **storing the same data multiple times**
  - the need to perform a single logical **update** multiple times
  - waste of storage space
  - inconsistency
- ideally we should have a DB design that stores each logical data item in only one place in the DB
- “controlled” redundancy in the DB approach
  - for improving the **performance** of queries
  - however, DBMS should have the capability to control the redundancy

example of controlled redundancy

redundant storage of StudentName and CourseNumber

(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

(b)

GRADE_REPORT	StudentNumber	StudentName	SectionIdentifier	CourseNumber	Grade
	17	Brown	112	MATH2410	B

an example of inconsistent record

additional implications

- potential for enforcing standards
- reduced application development time
- flexibility to change data structures
- availability of up-to-date information
- economies of scale
  - reduces the amount of wasteful overlap between activities of data processing personnel in different departments

historical development of DB technology

- early database Applications:
  - hierarchical and network models were introduced in mid 1960's and dominated during the seventies
  - quite difficult to reorganize the DB when changes were made to the requirements of the application
  - a bulk of the worldwide database processing still occurs using these models
- **hierarchical model**
  - implemented in a joint effort by IBM and North American Rockwell around 1965
  - resulted in the IMS family of systems
  - the most popular model
- **network model**
  - the first one to be implemented by Honeywell in 1964-65 (IDS System)
  - adopted heavily due to the support by CODASYL (CODASYL - DBTG report of 1971)
  - later implemented in a large variety of systems - IDMS (Cullinet - now CA), DMS 1100 (Unisys), IMAGE (H.P.), VAX -DBMS (Digital Equipment Corp.)

historical development of DB technology

- relational model based systems
  - the model that was originally introduced in 1970 by **E.F. Codd** (IBM) was heavily researched and experimented within IBM and the universities
  - relational DBMS products emerged in the 1980's
  - originally proposed to **separate the physical storage of data from its conceptual representation** and to provide a mathematical foundation for DBs
  - introduced **high-level query languages**
  - now in several commercial products (DB2, ORACLE, SQL Server, SYBASE, INFORMIX)

historical development of DB technology

- object-oriented applications
  - OODBMSs were introduced in late 1980's and early 1990's to cater to the need of **complex data processing** in CAD and other applications
  - the **complexity** of the model and the **lack of an early standard** contributed to their limited use
  - one set comprises models of persistent O-O Programming Languages such as C++ (e.g., in OBJECTSTORE or VERSANT), and Smalltalk (e.g., in GEMSTONE)
  - additionally, systems like O2, ORION (at MCC - then ITASCA), IRIS (at HP- used in Open OODB)
- object-relational models
  - started with Informix Universal Server
  - exemplified in the latest versions of Oracle-11g, DB2, and SQL Server etc. systems

historical development of DB technology

- interchanging data on the web
  - web contains data in HTML with links among pages
  - **XML** is considered to be the primary standard for interchanging data among various types of DBs and Web pages
  - XML DBs

extending database capabilities

- **new functionality** is being added to DBMSs in the following areas:
  - scientific applicationswith large amounts of data storage
  - image storage and retrieval
  - audio and video data management
  - data mining applications
  - spatial data management applications such as GIS
  - time series and historical data management
- the relational systems were not very suitable for these
- the above gives rise to new R & D in incorporating new data types, complex data structures, new operations, and storage and indexing schemes in database systems
  - e.g., VLDB conference

when not to use a DBMS

- main inhibitors of using a DBMS
  - high initial **investment** in hardware, software, and training
  - the **generality** that a DBMS provides for defining and processing data
  - **overhead** for providing security, concurrency control, recovery, and integrity functions
- when a DBMS may be unnecessary:
  - if the database and applications are simple, well defined, and not expected to change
  - if there are stringent **real-time requirements** that may not be met because of DBMS overhead
  - if access to data by multiple users is not required

databases vs. information retrieval

- database technology
  - applies to **structured and formatted data** that arises in routine applications in government, business, and industry
  - heavily used in manufacturing, retail, banking, insurance, finance, and health care industries
- information retrieval (IR) technology
  - deals with books, manuscripts, and various forms of library-based articles
  - concerned with **searching for material** based on the keywords, and also with the problems dealing with document processing, automatic text categorization, and so on
- web IR
  - billions of web pages containing images, text, and dynamic objects
  - needs IR + DB approaches

Oracle

- #1 DBMS company in the world
- #2 biggest software company in the world
- expensive and requires high maintenance cost charges
- why Oracle?
  - DBs (almost) never shrink
  - DBs contain mission critical information
  - etc.