COS221
L01 - Overview to Databases
(Chapter 1 - Editions 6 and 7)

Linda Marshall

22 February 2023

Databases and Database Users

Why are databases needed?

Database types

- Traditional
- Multimedia
- Geographic
- Data warehouse and Online Analytical Processing (OLAP)
- Real-time and Active
- ...

Definitions

1. A *database* is a collection of data.
2. *Data* is recorded known facts with implicit meaning.
3. *Database Management System (DBMS)* is a collection of programs used to create and maintain a database.

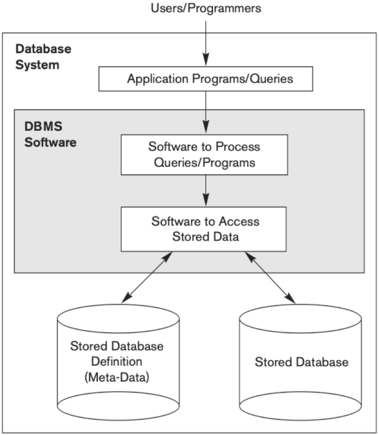
Definitions - Database

- A database comprises of *files* which stores *data records* comprising of *data elements*.
- implicit properties of a data base:
  - representation of an aspect of the real world (mini-world)
  - logically coherent data with implicit meaning
  - designed, built and populated with related data

Definitions - Database Management Systems (DBMS)

- Defines datatypes, structures and constraints of the data in the database
- Provides a construction process to store data on storage media
  - An *application program* sends queries or requests to the DBMS
  - A *query* is used to retrieve data
  - A*transaction* reads and writes data to the DBMS
- Querying and updating the database to reflect changes in the mini world
- Generating report results in the *manipulation* of the database
- By *sharing* the database, multiple users can access and manipulate the database simultaneously
- *Protects* against malfunction (hardware and software) and malicious access
- *Maintains* the system by allowing it to evolve over years

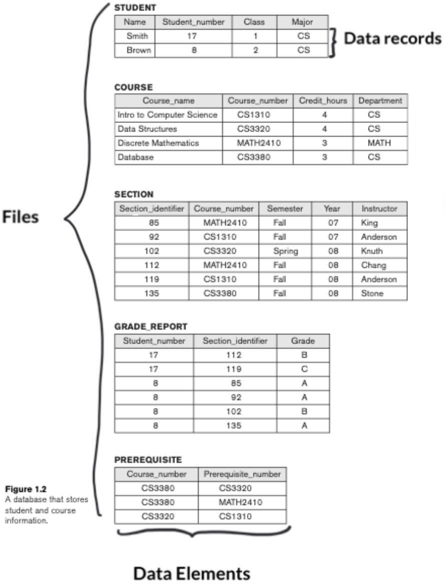
Definitions - Database Management Systems (DBMS)


Users/Programmers
Database System
Application Programs/Queries
DBMS Software
Software to Process Queries/Programs
Software to Access Stored Data
Stored Database Definition (Meta-Data)
Stored Database

Database + DBMS = Database System

Figure 1.1
A simplified database system environment.

An Example



Files
Data records
Data Elements

STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

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	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

GRADE_REPORT

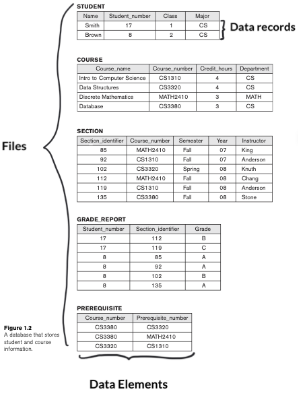
Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Figure 1.2
A database that stores student and course information.

An Example


Files
Data records
Data Elements

STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

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	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	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

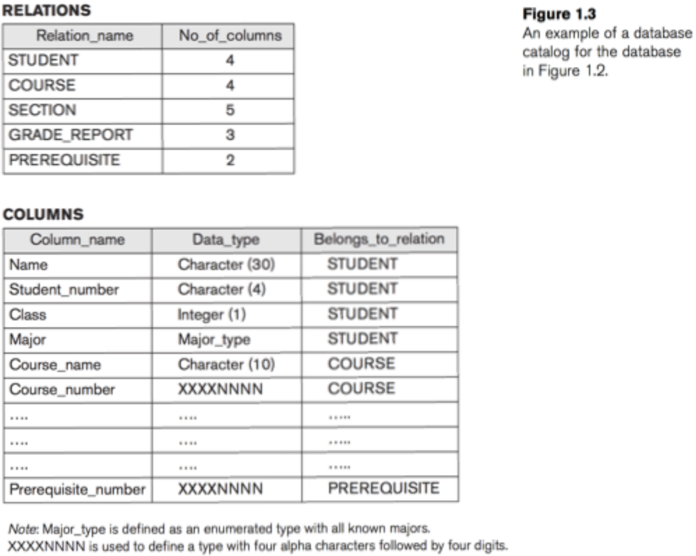
PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Figure 1.2
A database that stores student and course information.

- Each *data element* has a *data type* associated with it
- *Relationships* exist between *data records*

An Example - Meta-data



RELATIONS

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

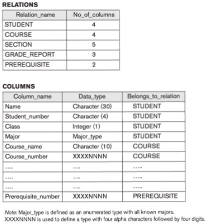
COLUMNS

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 enumerated 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.

An Example - Meta-data



RELATIONS

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

COLUMNS

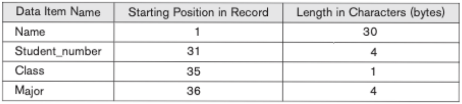
Column_name	Data_type	Belongs_to_relation
Name	Character (30)	STUDENT
Student_number	Character (4)	STUDENT
Class	Integer (1)	STUDENT
Major	Major_type	STUDENT
Course_name	Character (10)	COURSE
Course_number	XXXXNNNN	COURSE
....	....	.....
....	....	.....
....	....	.....
Prerequisite_number	XXXXNNNN	PREREQUISITE

Note: Major_type is defined as an enumerated 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.

- *Meta-data* is stored in the *catalog*
- Using a *catalog* promotes *program-data independence*
- *Program-data independence* is as a result of *data abstraction*
- A DBMS provides users with a *conceptual representation*

An Example - Meta-data

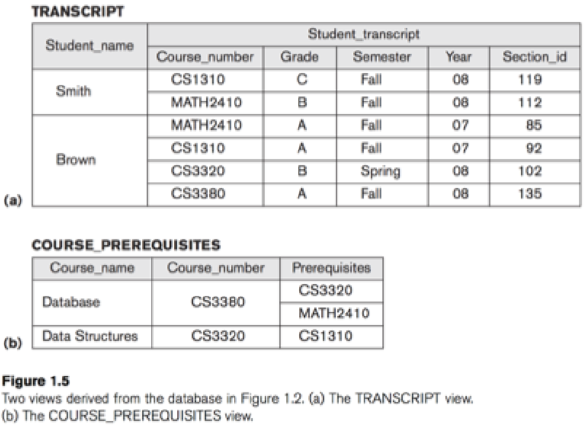


Data Item Name	Starting Position in Record	Length in Characters (bytes)
Name	1	30
Student_number	31	4
Class	35	1
Major	36	4

Figure 1.4
Internal storage format for a STUDENT record, based on the database catalog in Figure 1.3.

- A data model is a data abstraction that is used to provide the conceptual representation and hides the internal representation.

An Example - Views



TRANSCRIPT

(a)

Student_name	Student_transcript				
	Course_number	Grade	Semester	Year	Section_id
Smith	CS1310	C	Fall	08	119
	MATH2410	B	Fall	08	112
Brown	MATH2410	A	Fall	07	85
	CS1310	A	Fall	07	92
	CS3320	B	Spring	08	102
	CS3380	A	Fall	08	135

COURSE_PREREQUISITES

(b)

Course_name	Course_number	Prerequisites
Database	CS3380	CS3320
		MATH2410
Data Structures	CS3320	CS1310

Figure 1.5
Two views derived from the database in Figure 1.2. (a) The TRANSCRIPT view. (b) The COURSE_PREREQUISITES view.

- Multiple views can be derived from the database.
- A view may contain data derived from the database but not explicitly stored. This is referred to as *virtual data*.

Multiuser support

- DBMS must include concurrency control, especially for OnLine Transaction Processing (OLTP) type applications, e.g. aircraft ticket reservations
- The DBMS enforces the following properties on transactions
  - *isolation* - all transactions appear to execute in isolation
  - *atomicity* - all transactions execute or none do

Actors on the Scene

- Database Administrator
- Database Designers
  - Requirements specification and analysis
  - Conceptual design (e.g. ER diagrams)
  - Logical design (e.g. Relational model)
  - Physical design (implementation of the design in a DBMS)
- End Users
  - casual
  - naive/parametric
  - sophisticated
  - standalone
- Systems Analysts and application programmers - use the database

Workers behind the Scene

- DBMS designers and implementers
- Tool developers
- Operators and maintenance - typically do not use the database

Advantages of using a DBMS

- Controlling redundancy - Normalisation, denormalisation and controlled redundancy.
- Restricting unauthorised access - Security and authorisation subsystem. Access to privileged software.
- Persistent storage of widgets
- Providing storage structures and search techniques for query processing - Indexing, buffering, caching ... optimisation of query processing.
- Backup and Recovery
- Multiple user interfaces
- Representation of complex relationships
- Enforcement of integrity constraints - Referential integrity, key/uniqueness, business rules and semantics.

Advantages of using a DBMS, *cont.*

- Permitting inferencing and actions using rules - Deductive and Active databases make use of declarative/complex rules to extract information. Triggers are used to activate rules
- Enforcing standards
- Reduction in application development time
- Flexibility
- Availability of up-to-date information
- Economies of scale