Database Management Systems

Vijaya Saradhi

IIT Guwahati

Fri, 10th Jan 2020

File Systems

Limitations

- We do not have 500 GB of main memory (earlier days)
- 32-bit computer system can refer directly to only 4GB of data
- Write special purpose programs for every question
- Protect data from inconsistent changes
- Protect data from unauthorized access
- Protect data from failures
- Restore data in case of disk failures (what is the mechanism?)
- Issues in making copies of data
- Issues in updating data

File Systems

Limitations

- Write special purpose programs for every question
  - Lengthy development time
  - Difficulty in getting the answers quickly (some one has to develop these programs)
  - Maintaining data AND programs becomes complex task (earlier we have only data)
  - Flexible formatting is not possible

Advantages of DBMS

Advantages

- Availability data is can be accessed at any time
- Data independence Application programs should not be exposed to details of data representation
- Efficient data access effective storage and efficent retrival
- Data integrity and security always accessed through DBMS which enforces constraints
- Data administration centralizing the administration of data leads to significant improvements
- Concurrent access
- Crash Recovery
- Reduce application development time

DBMS

What it contains?

- A DBMS is a collection of programs that enables users to create and maintain a database
- DBMS is a general-purpose software system
- It facilitates the process of
  - Defining
  - Constructing
  - Manipulating and
  - Sharing database among various users and applications

Defining

Elements

Defining a database involves specifying

- the data types
- data structures
- constraints on the data to be stored in the database
- Database descriptive information is also stored by the DBMS
- The description is in the form of database catalog which is the meta-data

Constructing, Manipulating and Sharing

Elements

- Constructing the database is the process of storing the data on some storage medium that is controlled by the DBMS
- Manipulating a database includes functions such as
  - Querying the database to retrieve specific data
  - Updating the database to reflect changes in the mini-world
  - Generating reports from data
- Sharing a database allows multiple users and programs to access the database simultaneously

An Example

University Database

- Storing information about students, courses, & grades
- Database structure and a few samples of data for such a database

An Example

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

An Example

University Database

- The database is organized in five files
- Each of which stores data records of same type
- STUDENT file stores records of all the students
- COURSE file stores records of all the courses
- ...

An Example

Define

- To *define* this database, structure of records of each file must be specified
- Specify different *types of data elements* to be stored for each record
- STUDENT record includes: Name, Student_number, Class and Major
- A similar description should be done for all the files
- Descriptions specify domain example: grade must be from the set {'A', 'B', 'C', 'D', 'F', 'I'}

An Example

Construct

- To *construct* this database, we store data to represent each student, course, section, grade report, and prerequisite
- Each of these must be in different files
- Records in various files may be related
- Example: Record Smit in Student file is related to two records in Grade_Report file
- Similarly, each record in Prerequisite file relates to two course records
  - One representing course
  - Other representing prerequisite
- Such relations are prevalent in relational databases

An Example

Manipulate - Querying

- Involve querying and updating
- Example: Retrieve the transcript of 'Smith'
- List names of students who took database course in fall 2008 and their grades in that section
- List the pre-requisites of database course

Manipulate - Updating

- Change the class of 'Smith' to sophomore
- Create a new section for database course for this semester
- Enter a grade of 'A' for 'Smith' in the database section of last semester

Characteristics

File Based Approach

- In file processing, each user defines and implements the files needed for a specific application
- The database definition is a part of programming
- Example: two distinct users
  - Academic section - uses two files students and grades
  - Accounts section - uses two files students and fee payments
- Same information is stored in different places; maintained by different people; programs to manipulate these files
- Leads to redundancy in defining and storing data
- Duplicated efforts to maintain common up-to-date data

Characteristics

Database Approach

- A single repository maintains data
- Data is defined once
- Accessed by various users
- Data elements names (labels) are defined once
- These names are used by every one who interacts with the database

Characteristics

Differentiating

- 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

Meta-data

- Contains complete definition or description of database
- Database structure
- Database constraints
- Stored in DBMS catalog or meta-data
- Meta-data is used by DBMS software and users
- DBMS is not written for a specific database application
- DBMS refers to meta-data to infer the structure of files
- Must work with any number of database applications

Self-Describing Nature

Meta-data - Traditional Applications

- Data definition is part of the application programs

```
int student;
char name[50];
float cpi;
```

- The type and storage structures are specified in the programming
- On contrary data stored in the files do not have these definitions
- Different application can potentially interpret this data differently as

```
char student[9];
char name[50];
char cpi[5];
```

- Even class declaration and associated objects carry this limitation
- Persistence definition is needed

Catalog/Meta-data

Relations meta-data

RELATIONS

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

Catalog/Meta-data

Columns meta-data

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
....	....	.....
....	....	.....

Insulation

Data Abstraction

- Traditional file processing, structure of data file is embedded in the application programs
- Changes to file structure require changing application programs
- DBMS stores structure of data files in DBMS catalog
- This is program-data independence

Internal File Structure

Low-level Details

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

Multiple Views

Views

- Multiple users interact with the data
- Each require a distinct perspective of view
- A view may be a subset of the database
- It may contain virtual data that is derived from the database

Multiple Views

Views

TRANSCRIPT

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

Multiple Views

Views

COURSE_PREREQUISITES

Course_name	Course_number	Prerequisites
Database	CS3380	CS3320
		MATH2410
Data Structures	CS3320	CS1310

Sharing of Data & Multiuser Transaction Processing

Sharing

- Allow multiple users to access database at the same time
- DBMS must include concurrency control software
- Allows users to access and update data in a controlled manner
- Example:
  - Several reservation agents try to assign a seat on an airline flight
  - DBMS should ensure that each seat can be accessed by only one agent at a time
- This type of applications well known as online transaction processing (OLTP) applications
- Concurrent transactions operate correctly and efficiently

Transaction

Definition

- This concept became central to many database applications
- Transaction is an executing program or executing process
- It includes one or more database accesses reading or updating
- Each transaction is supposed execute a logically correct database access
- Must enforce several transaction properties
  - Atomicity either all database opreations in a trasaction are executed or none are executed
  - Consistency Resulting database state must obey the constraints on the data
  - Isolation Ensures each transaction appears to execute in isolation from other transactions
  - Durable The updates must lead to persist

Various Users

Actors

- Large database involve hundreds of users
- Responsible for design, use and maintenance of large database
  - Database Administrators
  - Database Designers
  - End Users

Administrator

The Chief

- To oversee and manage resources
- authorizing access
- coordinate and monitor the use
- acquire software and hardware as needed

Designers

Specific Role

- Responsible for identifying data to be stored in the database
- Choosing appropriate structures to represent and store the data
- These steps comes before database implementation and use
- Understands users requirements and create a design that meet the requirements
- Also responsible for developing views of the database

End Users

Variety of Users

- Casual end users
- Naive users
- Sophisticated users
- Standalone users

Input/Output

Options

- GUI based
- Terminal based
- Batch mode
- Voice driven
- ...