Database Management Systems

Vijaya Saradhi

IIT Guwahati

Mon, 13th Jan 2020

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

Restricting unauthorized access

Restricted access

- In multiple users use, most users will not be authorized to access all information
- Example: Financial data, grades data, employee data, student data etc.
- Only authorized users permitted to retrieve; some are allowed to retrieve and update etc.
- Access operations - retrieval or update must also be controlled
- The responsibility of security and authorization subsystem

Providing storage structures for efficient query processing

Efficient Query Processing

- DBMS must provide capabilities for efficiently executing queries and updates
- As data is stored on disk, DBMS must provide specialized data structures and search techniques
- They should speed up disk search
- Auxiliary files called indexes are used for this purpose
- Indexes are tree data structures or hash data structures
- Records must be copied from disk to main memory during a query
- DBMS must have buffering or caching module
- OS performs data-to-memory buffering
- DBMS do this on their own

Providing backup and recovery

Backup & Recovery

- DBMS must provide facilities for recovering from hardware or software failures
- The backup and recovery subsystem is responsible for this
- Examples: Network failures, system failures, hardware failures, power failures etc.
- Recovery subsystem is responsible for restoring to the state it was in before transaction started executing

DBMS Interfaces

Multiple user interfaces

- Forms-based interfaces (GUI based)
- Menu-based interfaces (Web based)
- Natural language interfaces
- Speech input and output
- Terminal based
- Batch mode
- External connectivity
- Interfaces for DBA

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

DBMS Architecture

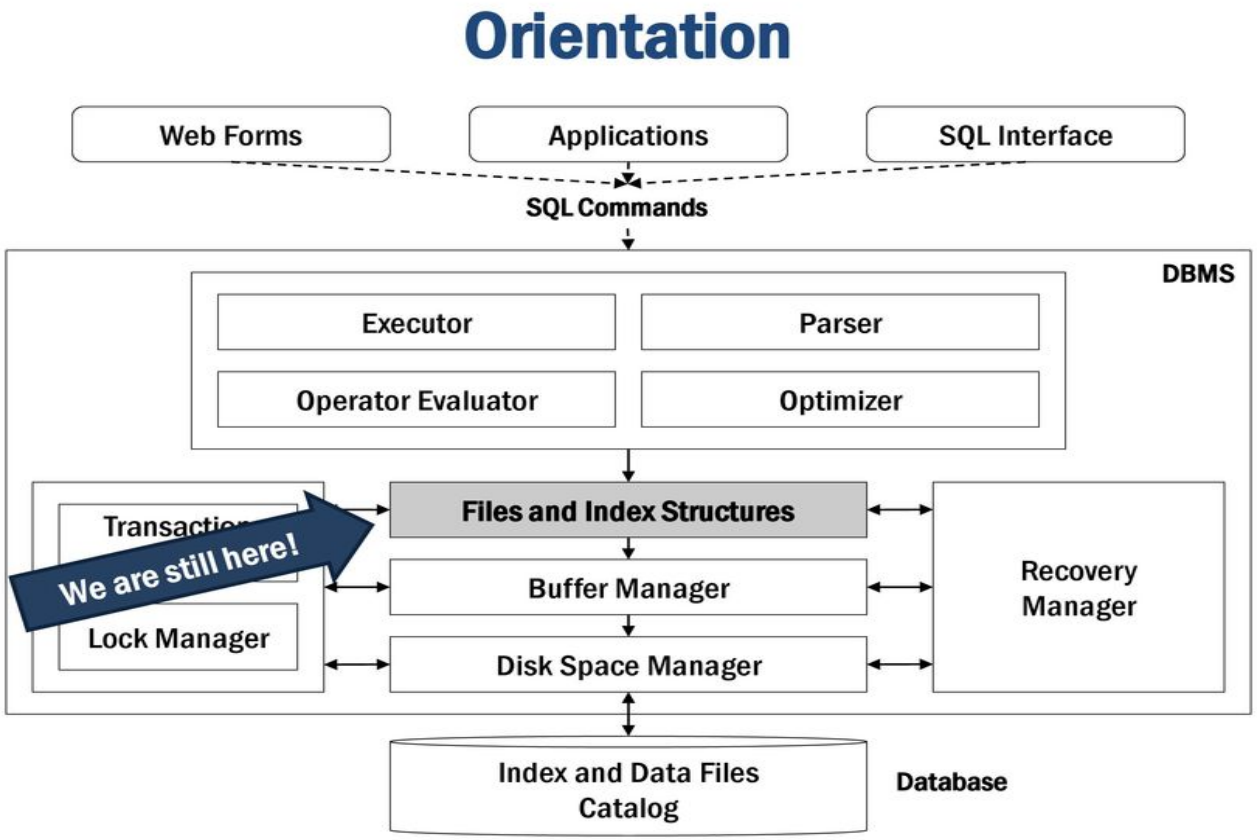


Figure Credit: Raghu Ramakrishnan and Johannes Gehrke: "Database Management Systems", McGraw-Hill, 2003.

Advantages

- Redundancy: storing same data multiple times leads to problems

GRADE_REPORT

Student_number	Student_name	Section_identifier	Course_number	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

- One update needs to be performed multiple times - duplication of effort
- Wastage of storage space; for large databases this is a concern
- Data may become inconsistent due to redundancy: when update is

Advantages

Continued

- We should ideally have each logical data item (name, student identifier, etc) stored in only one place
- This is data normalization
- It ensures consistency and saves storage space
- Some times it is necessary to use controlled redundancy
- This is to improve the performance of queries
- Example: Student_name and course_number are stored in grade report

Controlled Redundancy

Stud. No	Stud. Name	Sec. Id	Course No.	Grade
17	Smit	112	MATH2410	B
17	Smit	119	CS1310	C
8	Brown	85	MATH2410	A
8	Brown	82	CS1310	A
8	Brown	102	CS3320	B
8	Brown	105	CS33280	A

Controlled Redundancy

Reasons

- Whenever we retrieve a record, both student name and course number are retrieved
- By placing them together time for searching student name when retrieving grade report record is saved
- This will speed up the query time
- This is known as denormalization

Controlled Redundancy

DBMS overheads

- Should have the capability to control this redundancy
- Prohibit inconsistencies among files
- This should be performed automatically
- Check student_name, stduent_number values with that in STUDENT table
- Similarly check section_id and Course number with that in SECTION file records
- Such checks can be specified during database design
- Are automatically enforced

Enforcing Integrity Constraints

ICs

- Each database application has certain integrity constraints
- Every student has unique roll number
- Unique mobile number
- Unique address
- In a semester student should not register for more than 40 credits
- B Tech student should register for a minimum of 290 credits for award of degree

Active Databases

Active Database

- Writing triggers to perform certain actions
- Writing events to perform time based events
- Writing involved procedures to enforce rules stored procedures

Additional Benefits

Benefits

- Enforcing standards
  - Facilitates communication and cooperation among vairous departments, projects and users
  - Names, data elements, display formats, report structures, terminology
- Reduced application development time
- Flexibility
- Availability of up-to-date information

Data Model

Definition

Collection of high level data description constructs that hide many low-level storage details

Data Model

Categories

- Conceptual data models: provide concepts that are close to the way users perceive data
- Low-level/Physical data models: details how data is stored on computer storage media
- Representational data models

Conceptual Data Model

About

- They use concepts such as entities, attributes and relationships
- An entity represents a real-world object or a concept. That has physical existence or conceptual thing
- An attribute represents some property that describes an entity
- A relationship among two or more entities represents an association among entities
- A popular high-level conceptual data model is Entity-Relationship model

Physical data models

About

- Describe how data is stored as files
- Specifies record formats, record ordering, access paths
- access path is a structure that makes the search for a record(s) efficient
- An index is an example of an access path

Schemas, Instances, & Database State

About

- Description of database is different from database itself
- The description of a database is called database schema
- Schema is specified during database design
- Schema's do not change frequently as opposed to data

Schema - Example

STUDENT

Name	Student_number	Class	Major

COURSE

Course_name	Course_number	Credit_hours	Department

PREREQUISITE

Course_number	Prerequisite_number

SECTION

Section_identifier	Course_number	Semester	Year	Instructor

GRADE_REPORT

Student_number	Section_identifier	Grade

Schemas

About

- Displays some aspects of a scheme
- Data types, relationship between schemas are not shown
- Constraints are not represented in this scheme
- Not all constraints can be represented in the schema

Schemas and databases

About

- When we define a new database, we specify its database schema only to the DBMS
- Immediately after this database state is empty
- We get to initial state when database is populated with some data.
- Every update operation leads to a different database state
- At any point database has a *current state*
- DBMS is responsible for ensuring that every state is a valid state

Database - at a particular moment of time

database state or snapshot or set of occurances or instances

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

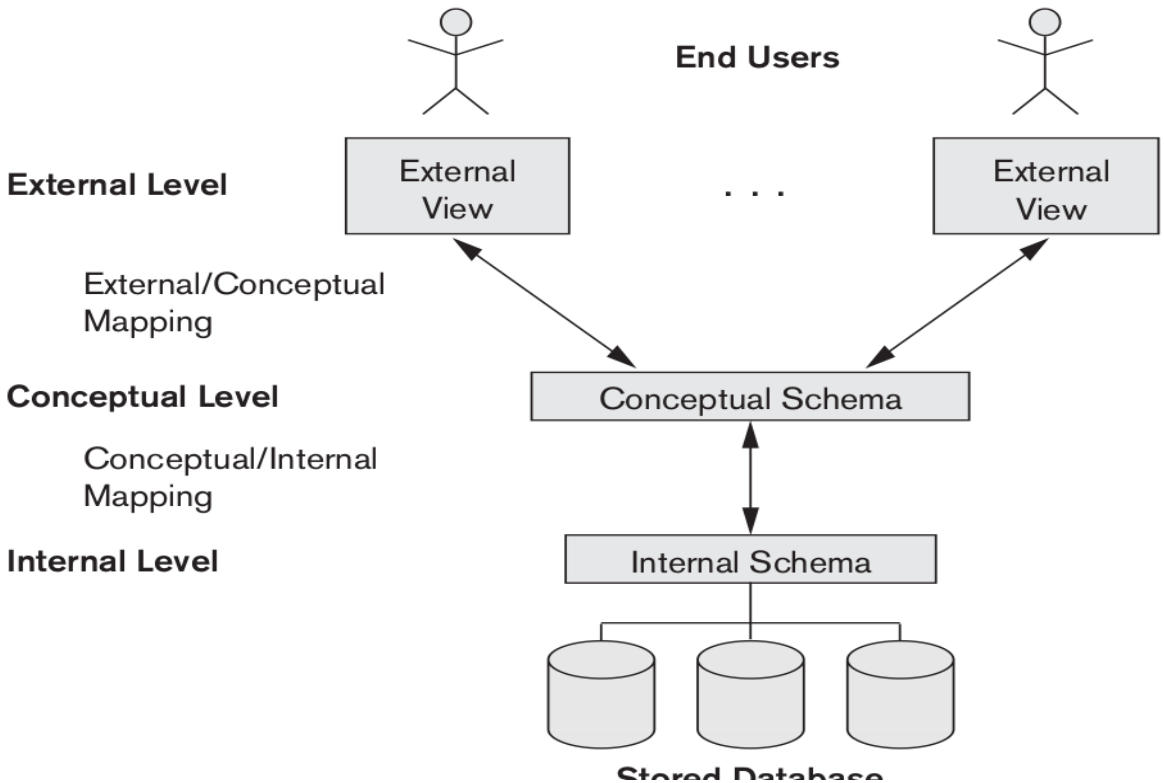
Course_number	Prerequisite_number

Database State

- When a new database is defined state of database is empty
- When database is populated with initial data, initial state is then present
- Every modification operation on the database yields a new state - current state
- Database is in valid state when the current state satisfies structure and constraints

Three Schema Architecture

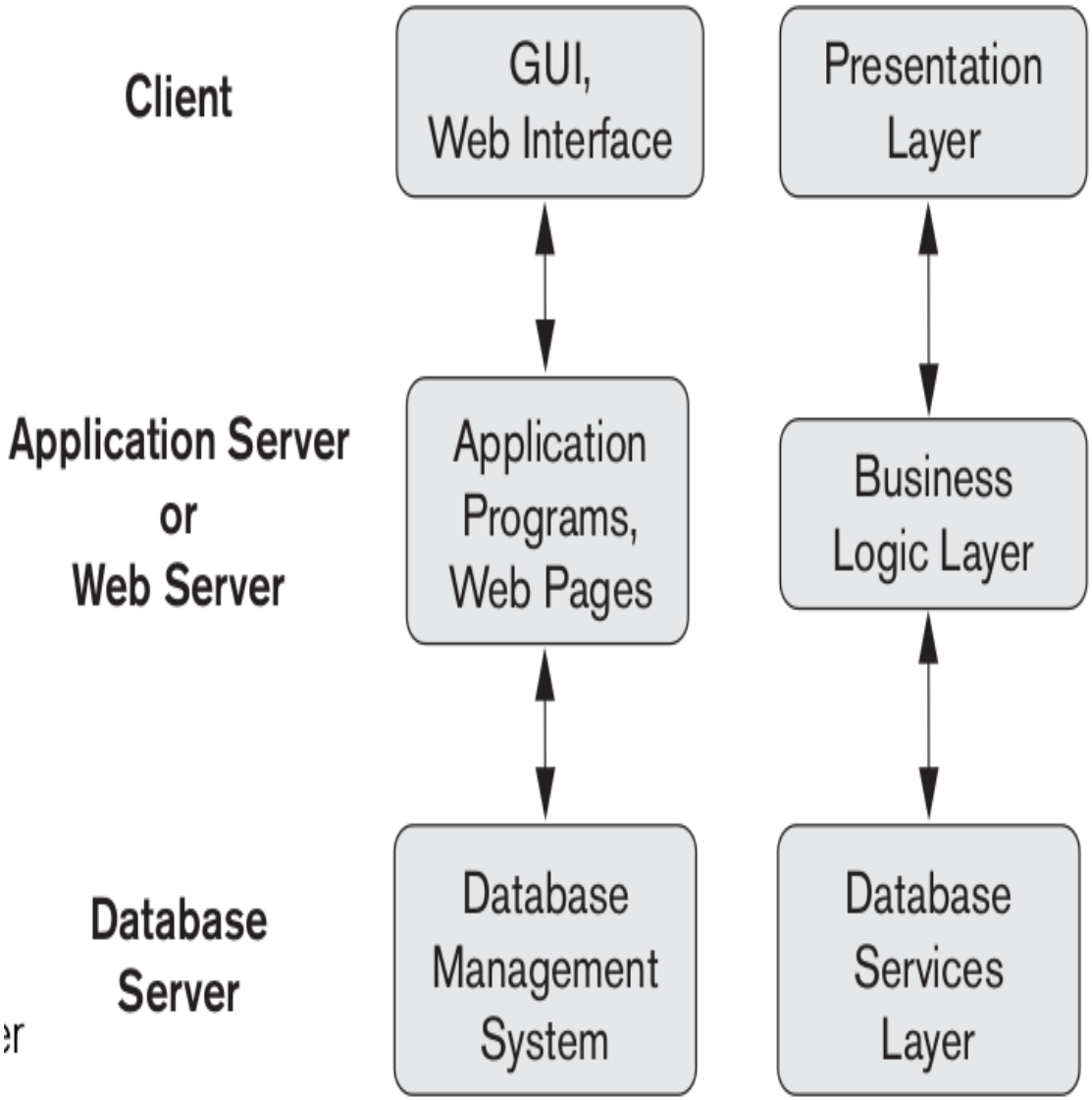
- Proposed to realize the three characteristics;
  - Self-describing nature
  - Insulation between programs and data; data abstraction
  - Support to multiple views of data



Three Schema Architecture

- Internal Schema (Physical Schema) complete details of physical storage structures along with access path are described
  - Store all relations in unsorted/sorted files of records
  - Create indexes on specified columns for access path mechanism
- Conceptual Schema Describes structure of the database
  - Describes entities, data types, relationships, user operations and constraints
  - Physical storage details are not part of the conceptual schema description
- External Schema Describes part of the database (views)
  - That is of interest to a particular group of users
  - Hides all the other database that are not relevant

3-tier


Client
Application Server
or
Web Server
Database
Server
GUI,
Web Interface
Application
Programs,
Web Pages
Database
Management
System
Presentation
Layer
Business
Logic Layer
Database
Services
Layer

DBMS Languages

- Data Definition Language
- Data Manipulation Language
- View Definition Language