CS430: Database Systems

Dr. Chetan Jaiswal,
Department of Computer Science,
Truman State University
cjaiswal@truman.edu

Chapter 1

- Introduction
- An Example
- Characteristics of the Database Approach
- Actors on the Scene
- Workers behind the Scene
- Advantages of Using the DBMS Approach
- A Brief History of Database Applications
- When Not to Use a DBMS

Introduction

- **Database**
  - Collection of related data
  - Known facts that can be recorded and that have implicit meaning
  - Miniworld or universe of discourse (UoD)
  - Represents some aspect of the real world
  - Logically coherent collection of data with inherent meaning
  - Built for a specific purpose

Introduction

- **Example of a large commercial database**
  - **Amazon.com (42TB), AT&T (323TB), Youtube (Unknown >> 500TB), NERSC (National Energy Research Scientific Computing Center, 2.85PB)…**
- **Database management system (DBMS)**
  - **Collection of programs**
  - **Enables users to create and maintain a database**

https://www.scribd.com/document/83087623/The-Top-Ten-Largest-Databases-in-the-World

Introduction

- **Meta-data**
  - Database definition or descriptive information
  - Stored by the DBMS in the form of a database catalog or dictionary
- **Manipulating a database**
  - Query and update the database miniworld
  - Generate reports

Introduction

- **Transaction**
  - May cause some data to be read and some data to be written into the database
- **Protection includes**
  - System protection
  - Security protection
- **Maintain the database system**
  - Allow the system to evolve as requirements change over time

An Example

- **UNIVERSITY database**
  - Information concerning students, courses, and grades in a university environment
- **Data records**
  - Student
  - Course
  - Section
  - Grade-report
  - Prerequisite

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

Figure 1.2
A database that stores student and course information.

An Example

- Specify structure of records by specifying data type for each data element
  - String of alphabetic characters
  - Integer
  - Character
  - Date
  - Time
  - Etc.

An Example of a Catalog

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

- Construct UNIVERSITY database
  - Store data to represent each student, course, section, grade report, and prerequisite as a record in appropriate file
- Relationships among the records
- Manipulation involves querying and updating

An Example

- **Update examples**
  - Change the class of ‘Smith’ to sophomore
  - Create a new section for the ‘Database’ course for this semester
  - Enter a grade of ‘A’ for ‘Smith’ in the ‘Database’ section of last semester

Self-Describing Nature of Database systems

- Database system contains complete definition of structure and constraints
- Meta-data
  - Describes structure of the database
- Database catalog used by
  - DBMS software
  - Database users who need information about database structure

Insulation Between Programs and Data

- **Program-data independence**
  - Structure of data files is stored in DBMS catalog separately from access programs
- **Program-operation independence**
  - Interface includes operation name and data types of its arguments
  - Implementation can be changed without affecting the interface

Data Abstraction

- **Data abstraction**
  - Allows program-data independence and program-operation independence
- **Conceptual representation of data**
  - Does not include details of how data is stored or how operations are implemented
- **Data model**
  - Type of data abstraction used to provide conceptual representation

Support of Multiple Views of the Data

- **View**
  - Subset of the database
  - Contains *virtual data* derived from the database files but is not explicitly stored
- **Multiuser DBMS**
  - Users have a variety of distinct applications
  - Must provide facilities for defining multiple views

Data Sharing and Transaction Processing

- Allow multiple users to access the database at the same time
- Concurrency control software
  - Ensure that several users' updates to the same data preserves database consistency
- Online transaction processing (OLTP) application

Transaction Processing

- **Transaction**
  - **Central to many database applications**
  - **Executing program or process that includes one or more database**
  - ***Atomicity, Consistency, Isolation, Durability (ACID)* properties**

Actors on the Scene

- Database administrators (DBA) are responsible for
  - Authorizing access to the database
  - Coordinating and monitoring its use
  - Acquiring software and hardware resources
- Database designers are responsible for
  - Identifying the data to be stored
  - Choosing appropriate structures to represent and store this data

Actors on the Scene

- **End users**
  - People whose jobs require access to the database: Casual end users, Naive or parametric end users, Sophisticated end users, Standalone users
- **System analysts**
  - Determine requirements of end users
- **Application programmers**
  - Implement these specifications as programs

Workers behind the Scene

- DBMS system designers and implementers
  - Design and implement the DBMS modules and interfaces as a software package
- Tool developers
  - Design and implement tools
- Operators and maintenance personnel
  - Responsible for running and maintenance of hardware and software environment for database system

Advantages of Using the DBMS

- Controlling redundancy
  - Sometimes necessary to use controlled redundancy to improve the performance of queries
- Restricting unauthorized access
  - Security and authorization subsystem
  - Privileged software

Advantages of Using the DBMS

- Providing storage structures and search techniques for efficient query processing
  - Indexes
  - Buffering and caching
  - Query processing and optimization

Advantages of Using the DBMS

- Providing backup and recovery
  - Backup and recovery subsystem of the DBMS is responsible for recovery
- Providing multiple user interfaces
  - Graphical user interfaces (GUIs)
- Representing complex relationships among data
  - May include numerous varieties of data that are interrelated in many ways

Advantages of Using the DBMS

- **Enforcing integrity constraints**
  - **Referential integrity constraint: Every section record must be related to a course record**
  - **Key or uniqueness constraint: Every course record must have a unique value for Course_number**
  - **Business rules**
  - **Inherent rules of the data model**

Advantages of Using the DBMS

- Additional implications of using the database approach
  - Reduced application development time
  - Flexibility
  - Availability of up-to-date information
  - Economies of scale

When Not to Use a DBMS

- More desirable to use regular files for
  - Simple, well-defined database applications not expected to change at all
  - Stringent, real-time requirements that may not be met because of DBMS overhead
  - Embedded systems with limited storage capacity
  - No multiple-user access to data

Summary

- Database
  - Collection of related data (recorded facts)
- DBMS
  - Generalized software package for implementing and maintaining a computerized database
- Several categories of database users
- Database applications have evolved
  - Current trends: IR (Information Retrieval), Web, Statistical Analysis

DBMS Architecture

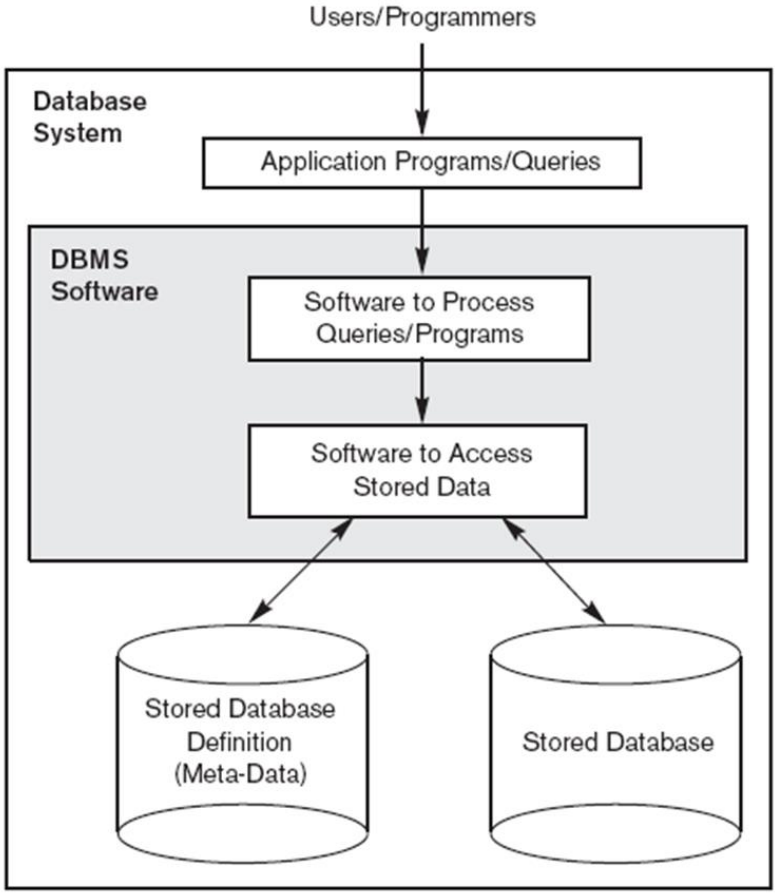


Figure 1.1
A simplified database system environment.

Data Model Categories

- Entity
  - Represents a real-world object or concept
- Attributes
  - Represents some property of interest
  - Further describes an entity
- Relationship among two or more entities
  - Represents an association among the entities
  - Entity-Relationship model
- Database Schema
  - Description of a database

Data Model Categories

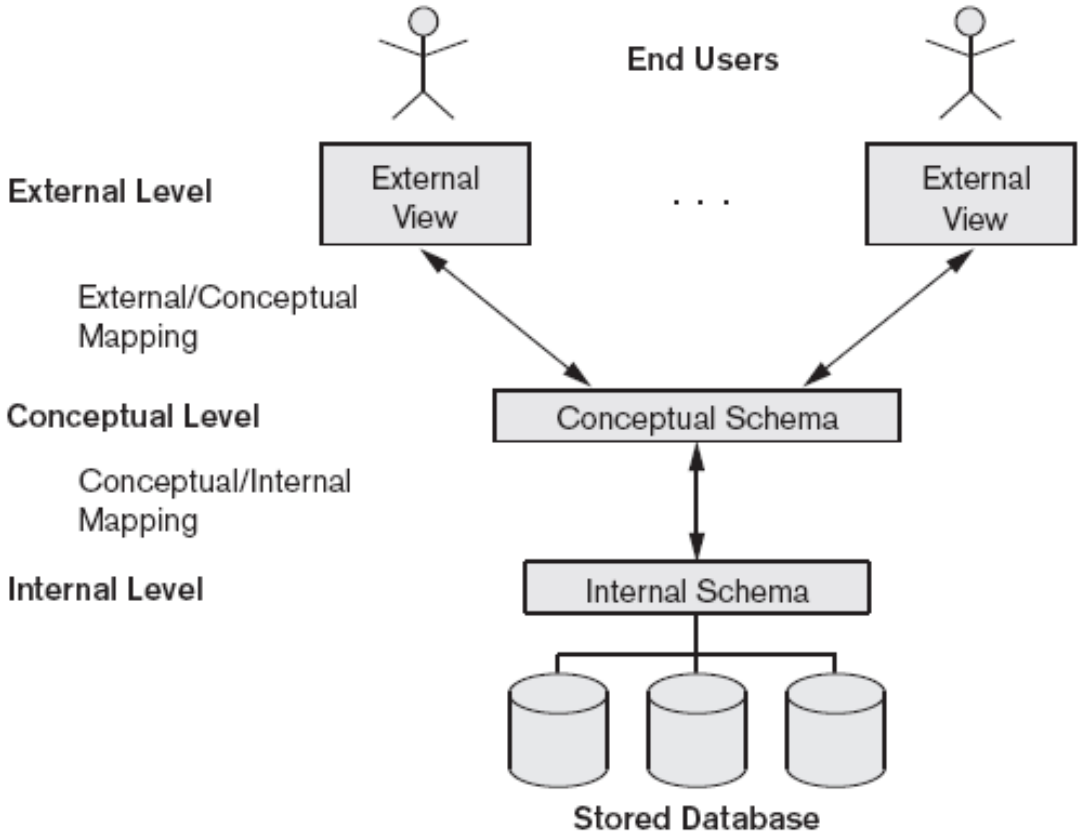
- **Relational data model**
  - Used most frequently in traditional commercial DBMSs
- **NoSQL data model: Access based storage or requirement based storage**
  - Column: Accumulo, Cassandra, Druid, HBase, Vertica
  - Document: Clusterpoint, Apache CouchDB, Couchbase, DocumentDB, HyperDex, Lotus Notes, MarkLogic, MongoDB, OrientDB, Qizx
  - Key-value: CouchDB, Oracle NoSQL Database, Dynamo, FoundationDB, HyperDex, MemcacheDB, Redis, Riak, FairCom c-treeACE, Aerospike, OrientDB, MUMPS
  - Graph: Allegro, Neo4J, InfiniteGraph, OrientDB, Virtuoso, Stardog
  - Multi-model: OrientDB, FoundationDB, ArangoDB, Alchemy Database, CortexDB

Three-Schema Architecture

- **Internal level**
  - Describes physical storage structure of the database
- **Conceptual level**
  - Describes structure of the whole database for a community of users (includes entities, data types, relationships, operations, constraints etc.)
- **External** or **view level**
  - Describes part of the database that a particular user group is interested in

Three-Schema Architecture

Figure 2.2
The three-schema architecture.



Data Independence

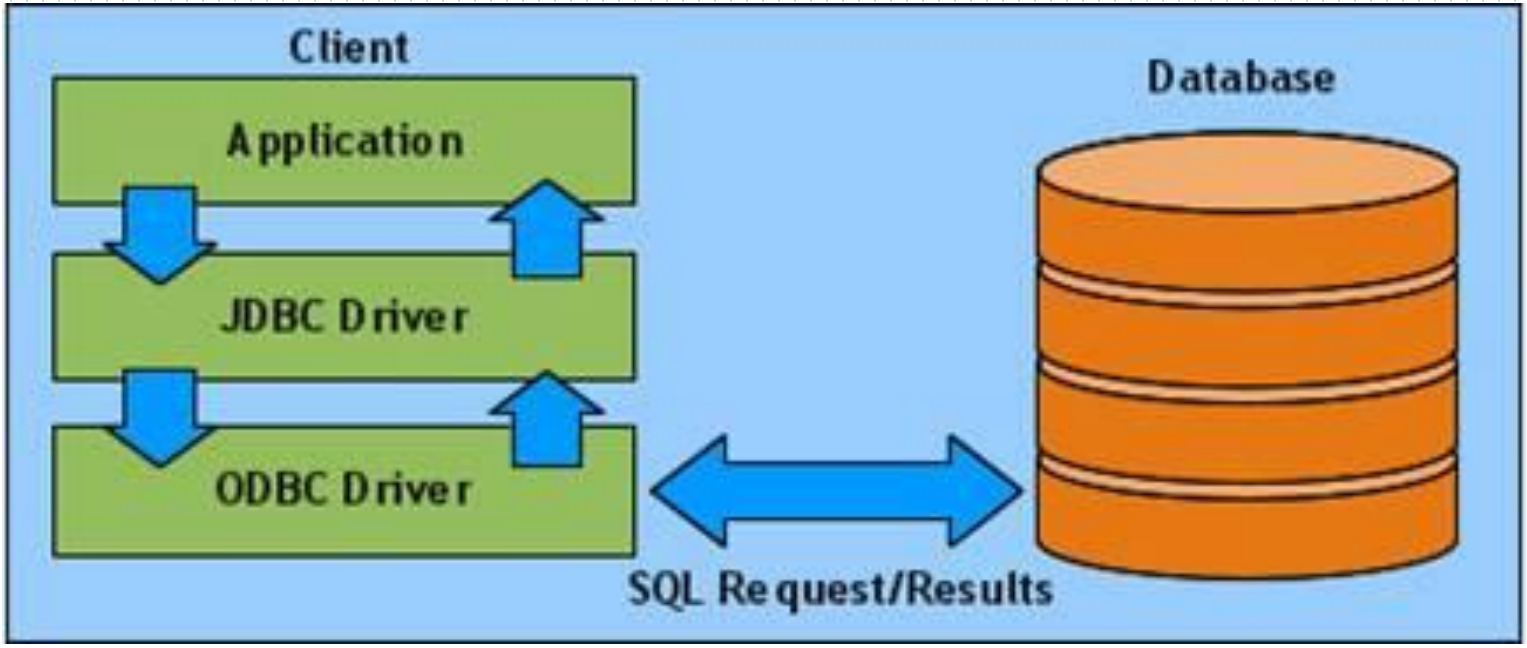
- Capacity to change the schema at one level of a database system
  - Without having to change the schema at the next higher level
- Types
  - Logical
  - Physical

DBMS Languages

- Data definition language (DDL)
- Storage definition language (SDL)
- View definition language (VDL)
- Data manipulation language (DML)
- Data control language (DCL)

Database System Architecture

JDBC-ODBC Bridge



XML…

- Stores data in between tags.
- Human readable and machine readable.
- Universal database.
- Data stored in form of files, no management system.
- Easy and light weight