Introduction to Database Management System

Topics before mid-1

- **Databases and Database Users**
- **Database System Concepts and Architecture**
- **The Relational Data Model and Relational Database Constraints**
- **Basic SQL**
- **More SQL: Complex Queries, Triggers, Views, and Schema Modification**
- MID -1

Topics after mid-1

- **Data Modeling Using the Entity– Relationship (ER) Model**
- **The Relational Algebra and Relational Calculus**
- **Basics of Functional Dependencies and Normalization for Relational Databases**
- **Mid-II**

- Topics after mid-II

- **Introduction to Transaction Processing Concepts and Theory**
- **Concurrency Control Techniques**
- **Database Recovery Techniques**
- **NOSQL Databases and Big Data Storage Systems**
- **Final Exam**

ASSESSMENT

- 4 Assignments 10%
- Project 10%
- Mid Exam 30%
- Final Exam 50%

- References

- ‘Fundamentals of database systems’ by Ramez Elmasri and Shamkant B. Navathe
- ‘Database Systems: A practical approach to design, implementation and management’ by Connolly and Begg
- `A first course in database systems’ by Ullman and Widom
- Other text book : ‘Database Systems’ by CJ Date

Learning Outcomes

- *Student will be able to understand the conceptual design of database*
- *Student will be able to describe the important of good database design*
- *Student will be able to apply database design for a real business*

Why Study Databases?

- Databases are useful
  - Many computing applications deal with large amounts of information
  - Database systems give a set of tools for storing, searching and managing this information
- Databases in CS
  - Databases are a ‘core topic’ in computer science
  - Basic concepts and skills with database systems are part of the skill set you will be assumed to have as a CS graduate

Data versus Information

Data

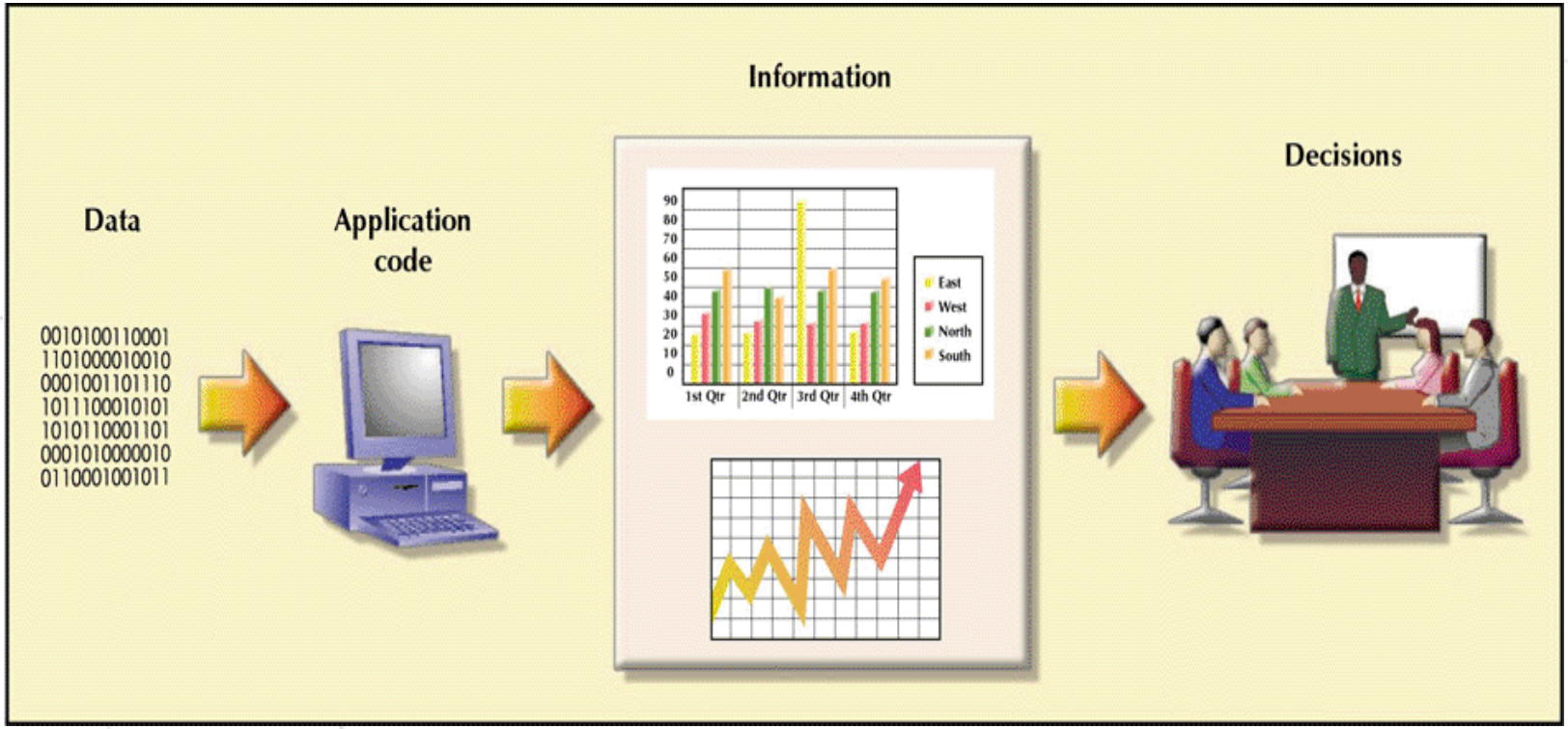
- Raw facts
  - Have not yet been processed to reveal their meaning to the end user
  - e.g. numbers, characters, graphics, images, audio, video
- Building blocks of information
- **Data management**
  - Generation, storage, and retrieval of data
- Data is what you store in *database.*

ID	Name	Address	City	State
1	Andrew	SW 16th Ave	Orlando	FL
2	Robert	23rd Terrace	Boston	MA
3	Bob	52nd Avenue	Bronx	NY
4	Lee Hung	52nd Avenue	Albany	NY
5	Richard	999 Palm Bay	Detroit	MI
6	Tony	15th Avenue	Austin	TX

Information

- Produced by processing raw data to reveal its meaning
- Requires context
  - e.g. Total sale for month June, Total students at Bahria Karachi campus
- Bedrock of **knowledge**
- Should be accurate, relevant, and timely to enable good decision making
- Information is what you **retrieve** from a database.

Information –
Pie Chart, Bar Graph
Table


Data
0010100110001
1101000010010
0001001101110
1011100010101
1010110001101
0001010000010
0110001001011
Application code
Information
90
80
70
60
50
40
30
20
10
0
1st Qtr
2nd Qtr
3rd Qtr
4th Qtr
East
West
North
South
Decisions

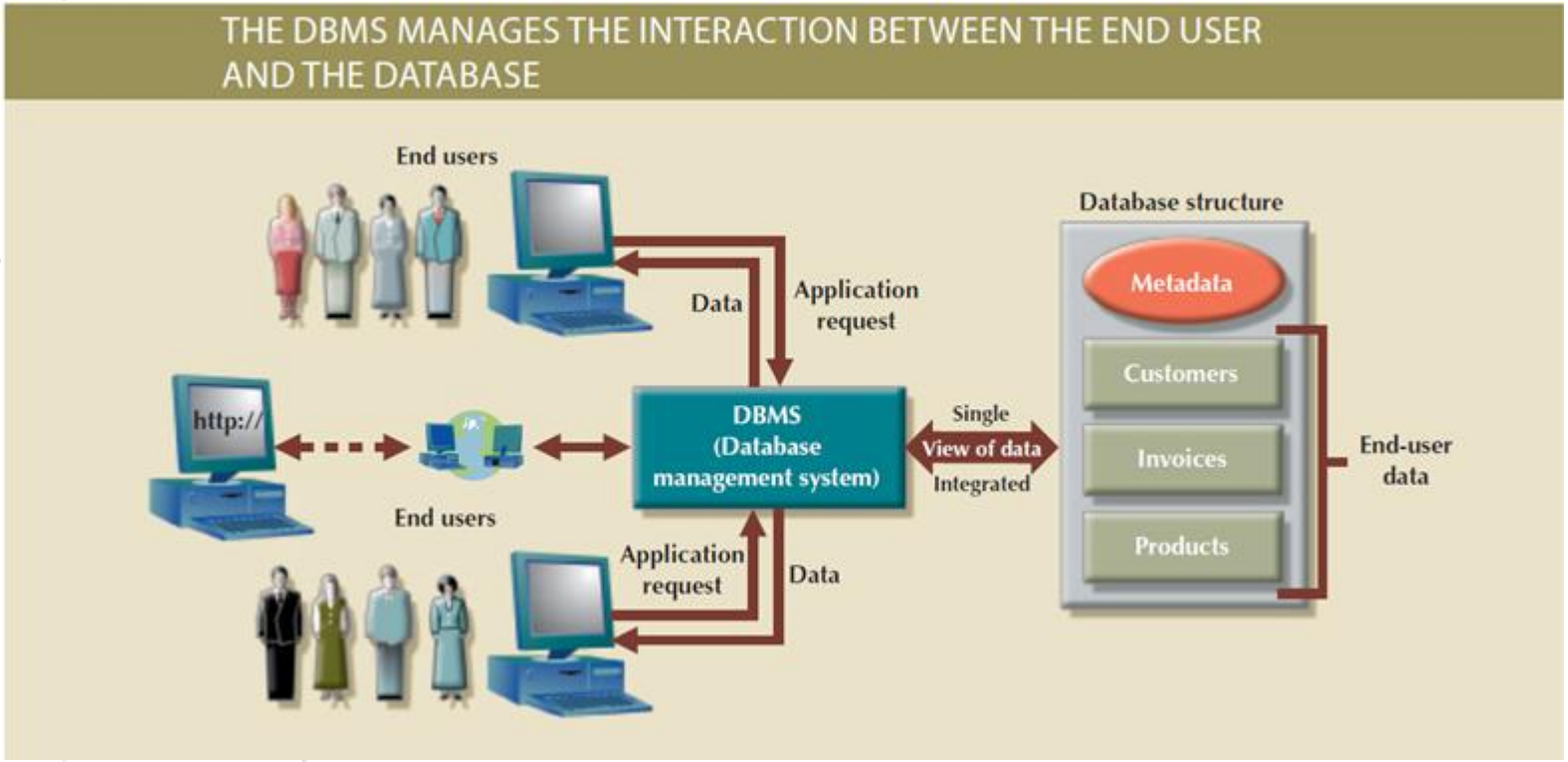
Introducing the Database

- Shared, integrated computer structure that stores a collection of:
  - End-user data - Raw facts of interest to end user
  - **Metadata**: Data about data, which the end-user data are integrated and managed
    - Describe data characteristics and relationships
- **Database management system (DBMS)**
  - Collection of programs
  - Manages the database structure
  - Controls access to data stored in the database

Name	Data Type	Size	Description
ID	Number	4	Unique Student ID
Name	Text	50	Student name
Address	Text	200	Mailing Address
City	Text	50	Residential City
State	Text	2	Residential State
ZIP	Number	5	ZIP Code

Metadata

The DBMS Manages the Interaction between the End User and the Database



Basic Definitions

- **Database:**
  - A collection of related data.
- **Data:**
  - Known facts that can be recorded and have an implicit meaning.
- **Mini-world:**
  - Some part of the real world about which data is stored in a database. For example, student grades and transcripts at a university.
- **Database Management System (DBMS):**
  - A software package/ system to facilitate the creation and maintenance of a computerized database.
- **Database System:**
  - The DBMS software together with the data itself. Sometimes, the applications are also included.

Types of Databases and Database Applications

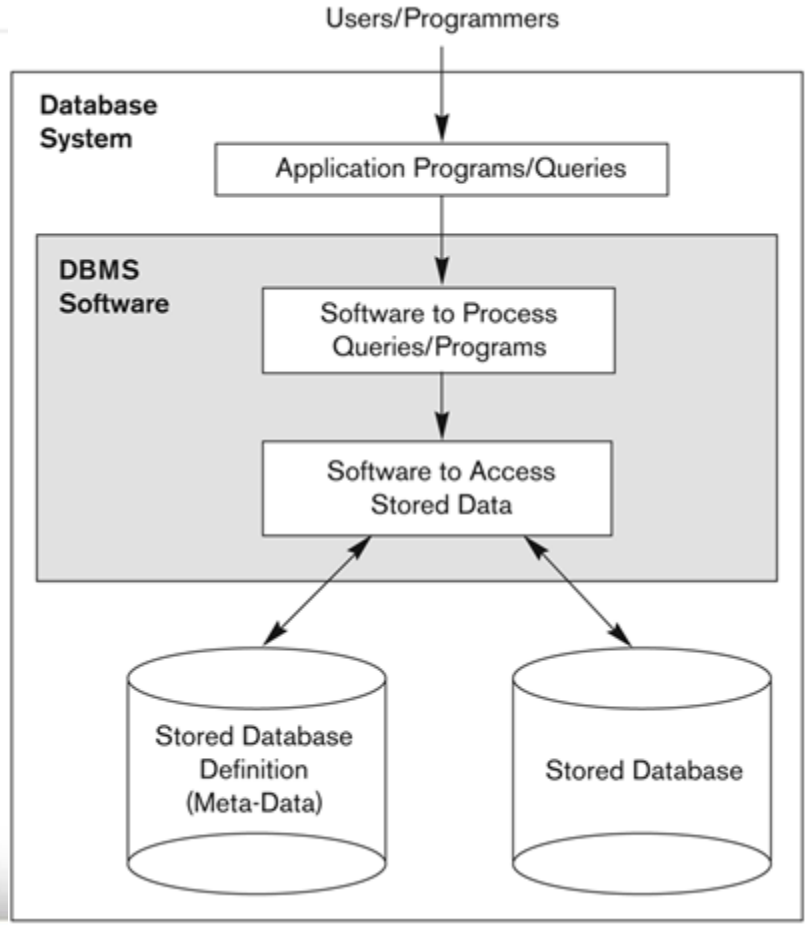
- Traditional Applications:
  - Numeric and Textual Databases
- More Recent Applications:
  - Multimedia Databases
  - Geographic Information Systems (GIS)
  - Biological and Genome Databases
  - Data Warehouses
  - Mobile databases
  - Real-time and Active Databases

Impact of Databases and Database Technology

- Businesses: Banking, Insurance, Retail, Transportation, Healthcare, Manufacturing
- Service Industries: Financial, Real-estate, Legal, Electronic Commerce, Small businesses
- Education : Resources for content and Delivery
- More recently: Social Networks, Environmental and Scientific Applications, Medicine and Genetics
- Personalized Applications: based on smart mobile devices


1 Personal Cloud Storage
Photos
Videos
Documents
2 Social Media
Posts
User profiles
Interests
3 Education
Student records
Online learning
Academic data
4 Healthcare
Patient details
Medical history
Diagnostic test records
5 Human Resource Management
Employee information
Salary details
Reimbursements
6 Aviation Industry
Flight tickets
Seat availability
Flight schedules
7 eCommerce
Online shopping
Tracking shipments
Order history
8 Hospitality Industry
Manage reservations
Room availability
Inventory management
9 Online Streaming
Watchlists
Playlists
Viewing history
10 Finance Sector
Transaction history
Stock prices
Fraud detection
11 Online Gaming
Player progress
In-app purchases
Group invitations
12 Weather
Temperature and humidity
Wind speed
Predict weather patterns

Simplified Database System Environment



Typical DBMS Functionality

- *Define* a particular database in terms of its data types, structures, and constraints
- *Construct* or Load the initial database contents on a secondary storage medium
- *Manipulating* the database:
  - Retrieval: Querying, generating reports
  - Modification: Insertions, deletions and updates to its content
  - Accessing the database through Web applications
- *Processing* and *Sharing* by a set of concurrent users and application programs – yet, keeping all data valid and consistent

Example of a Database (with a Conceptual Data Model)

- **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 Database (with a Conceptual Data Model) (cont.)

- **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

Example of a simple database

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

Application Activities Against a Database

- Applications interact with a database by generating
- Queries: that access different parts of data and formulate the result of a request
- Transactions: that may read some data and “update” certain values or generate new data and store that in the database
- Applications must not allow unauthorized users to access data
- Applications must keep up with changing user requirements against the database

Additional DBMS Functionality

- DBMS may additionally provide:
  - Protection or Security measures to prevent unauthorized access
  - “Active” processing to take internal actions on data
  - Presentation and Visualization of data
  - Maintenance of the database and associated programs over the lifetime of the database application
    - Called database, software, and system maintenance

Main Characteristics of the Database Approach

- **Self-describing nature of a database system:**
  - A DBMS **catalog** stores the description of a particular database (e.g. data structures, types, and constraints)
  - The description is called **meta-data***.
  - This allows the DBMS software to work with different database applications.
- **Insulation between programs and data:**
  - Called **program-data independence**.
  - Allows changing data structures and storage organization without having to change the DBMS access programs.

---

* Some newer systems such as a few NOSQL systems need no meta-data: they store the data definition within its structure making it self describing

Example of a simplified database 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 enumerared type with all known majors. XXXXNNNN is used to define a type with four alpha characters followed by four digits

Main Characteristics of the Database Approach (cont.)

- **Data Abstraction:**
  - A **data model** is used to hide storage details and present the users with a conceptual view of the database.
  - Programs refer to the data model constructs rather than data storage details
- **Support of multiple views of the data:**
  - Each user may see a different view of the database, which describes **only** the data of interest to that user.

Main Characteristics of the Database Approach (cont.)

- **Sharing of data and multi-user transaction processing:**
  - Allowing a set of **concurrent users** to retrieve from and to update the database.
  - *Concurrency control* within the DBMS guarantees that each **transaction** is correctly executed or aborted
  - *Recovery* subsystem ensures each completed transaction has its effect permanently recorded in the database
  - **OLTP** (Online Transaction Processing) is a major part of database applications. This allows hundreds of concurrent transactions to execute per second.

Database Users

- Users may be divided into
  - Those who actually use and control the database content, and those who design, develop and maintain database applications (called “Actors on the Scene”), and
  - Those who design and develop the DBMS software and related tools, and the computer systems operators (called “Workers Behind the Scene”).

Database Users – Actors on the Scene

- Actors on the scene
  - **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 to define the content, the structure, the constraints, and functions or transactions against the database. They must communicate with the end-users and understand their needs.

Database End Users

- Actors on the scene (continued)
  - **End-users:** They use the data for queries, reports and some of them update the database content. End-users can be categorized into:
    - **Casual**: access database occasionally when needed
    - **Naïve** or Parametric: they make up a large section of the end-user population.
      - They use previously well-defined functions in the form of "canned transactions" against the database.
      - Users of Mobile Apps mostly fall in this category
      - Bank-tellers or reservation clerks are parametric users who do this activity for an entire shift of operations.
      - Social Media Users post and read information from websites

Database End Users (cont.)

- **Sophisticated:**
  - These include business analysts, scientists, engineers, others thoroughly familiar with the system capabilities.
  - Many use tools in the form of software packages that work closely with the stored database.
- **Stand-alone:**
  - Mostly maintain personal databases using ready-to-use packaged applications.
  - An example is the user of a tax program that creates its own internal database.
  - Another example is a user that maintains a database of personal photos and videos.

Database Users – Actors on the Scene (cont.)

- **System Analysts and Application Developers**

  This category currently accounts for a very large proportion of the IT work force.
- **System Analysts**: They understand the user requirements of naïve and sophisticated users and design applications including canned transactions to meet those requirements.
- **Application Programmers:** Implement the specifications developed by analysts and test and debug them before deployment.
- **Business Analysts**: There is an increasing need for such people who can analyze vast amounts of business data and real-time data ("Big Data") for better decision making related to planning, advertising, marketing etc.

Database Users – Actors behind the Scene

- **System Designers and Implementors:** Design and implement DBMS packages in the form of modules and interfaces and test and debug them. The DBMS must interface with applications, language compilers, operating system components, etc.
- **Tool Developers**: Design and implement software systems called tools for modeling and designing databases, performance monitoring, prototyping, test data generation, user interface creation, simulation etc. that facilitate building of applications and allow using database effectively.
- **Operators and Maintenance Personnel**: They manage the actual running and maintenance of the database system hardware and software environment.

Advantages of Using the Database Approach

- Controlling redundancy in data storage and in development and maintenance efforts.
  - Sharing of data among multiple users.
- Restricting unauthorized access to data. Only the DBA staff uses privileged commands and facilities.
- Providing persistent storage for program Objects
- Providing Storage Structures (e.g. indexes) for efficient Query Processing.

Advantages of Using the Database Approach (cont.)

- Providing optimization of queries for efficient processing.
- Providing backup and recovery services.
- Providing multiple interfaces to different classes of users.
- Representing complex relationships among data.
- Enforcing integrity constraints on the database.
- Drawing inferences and actions from the stored data using deductive and active rules and triggers.

Additional Implications of Using the Database Approach

- Potential for enforcing standards:
  - This is very crucial for the success of database applications in large organizations. **Standards** refer to data item names, display formats, screens, report structures, meta-data (description of data), Web page layouts, etc.
- Reduced application development time:
  - Incremental time to add each new application is reduced.

Additional Implications of Using the Database Approach (cont.)

- Flexibility to change data structures:
  - Database structure may evolve as new requirements are defined.
- Availability of current information:
  - Extremely important for on-line transaction systems such as shopping, airline, hotel, car reservations.
- Economies of scale:
  - Wasteful overlap of resources and personnel can be avoided by consolidating data and applications across departments.

When not to use a DBMS

- Main inhibitors (costs) of using a DBMS:
  - High initial investment and possible need for additional hardware.
  - Overhead for providing generality, 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 access to data by multiple users is not required.
- When a DBMS may be infeasible:
  - In embedded systems where a general purpose DBMS may not fit in available storage

When not to use a DBMS

- When no DBMS may suffice:
  - If there are strict real-time requirements that may not be met because of DBMS overhead
  - If the database system is not able to handle the complexity of data because of modeling limitations (e.g., in complex genome and protein databases)
  - If the database users need special operations not supported by the DBMS (e.g., GIS and location based services).

Recent Developments

- Social Networks started capturing a lot of information about people and about communications among people-posts, tweets, photos, videos in systems such as:

- Facebook
- Twitter
- Linked-In

- All of the above constitutes data
- Search Engines- Google, Bing, Yahoo : collect their own repository of web pages for searching purposes

Recent Developments (cont.)

- New Technologies are emerging from the so-called non-database software vendors to manage vast amounts of data generated on the web:

- Big Data storage systems involving large clusters of distributed computers
  - NOSQL (Not Only SQL) systems
- A large amount of data now resides on the “cloud” which means it is in huge data centers using thousands of machines.

Extending Database Capabilities

- Emergence of Big Data Technologies and NOSQL databases
  - New data storage, management and analysis technology was necessary to deal with the onslaught of data in petabytes a day (10**15 bytes or 1000 terabytes) in some applications – this started being commonly called as “Big Data”.
  - Hadoop (which originated from Yahoo) and Mapreduce Programming approach to distributed data processing (which originated from Google) as well as the Google file system have given rise to Big Data technologies (Chapter 25). Further enhancements are taking place in the form of Spark based technology.
  - NOSQL (Not Only SQL- where SQL is the de facto standard language for relational DBMSs) systems have been designed for rapid search and retrieval from documents, processing of huge graphs occurring on social networks, and other forms of unstructured data with flexible models of transaction processing (Chapter 24).