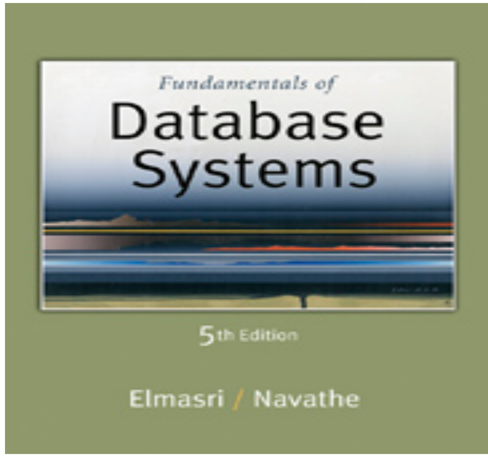
CSE 3330/5330

Database Systems 1



Fundamentals of
Database Systems

5th Edition

Elmasri / Navathe

PEARSON
Addison Wesley

Nadra Guizani

- Computer Engineering PhD – Purdue University
- nadra.guizani@uta.edu
- Taught – CS1, CS2, Machine Learning, Software Development w/ JAVA, Circuits
- Office Hours: M/W 5 pm – 6 pm ; Tuesday/Thursday 11 am – 1 pm
- Everything is posted and updated on **Canvas**
- **Textbook: *Fundamentals of Database Systems, Seventh Edition*** by Elmasri/Navathe

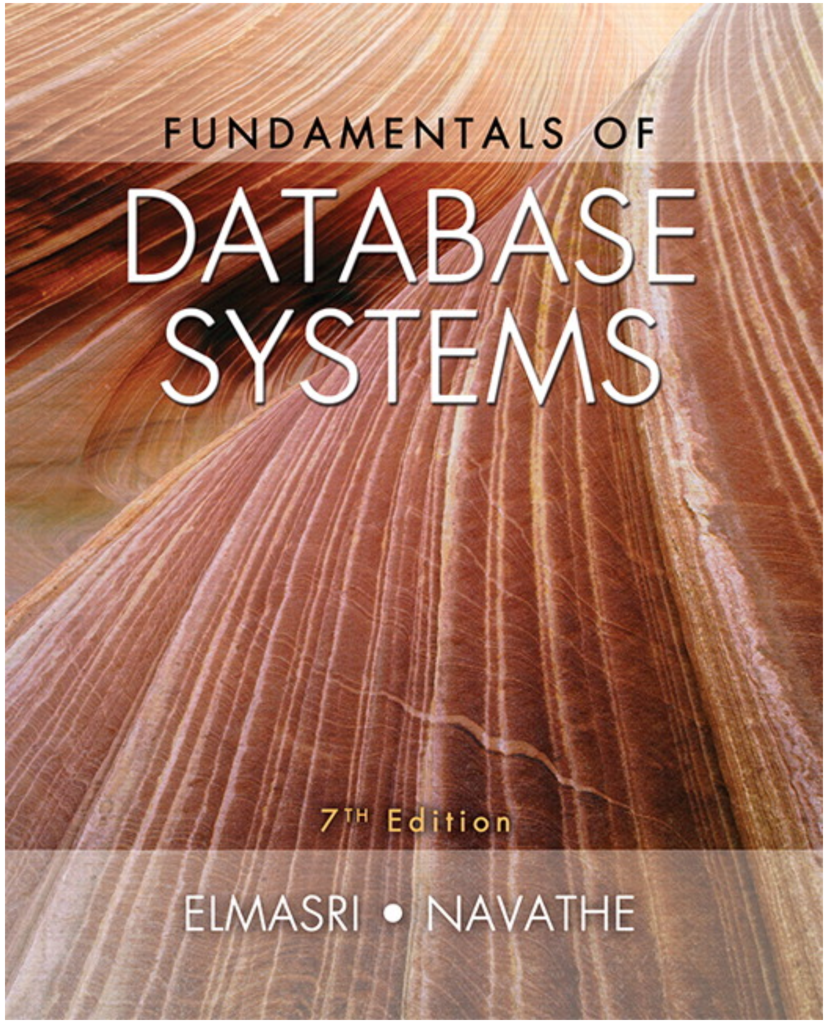
- **Grader: Nasim Shirvani Mahdavi**

- **Grader Email: nasim.shirvanimahdavi2@mavs.uta.edu**

- **GTA Office and Office Hours:** Mondays from 9 am to 11 am

- In addition to the tests, two projects will be given. The final grade will be calculated based on the two tests (50% of grade), projects (50% of grade).

- Projects will require: JAVA programming using JDBC, **or** C/C++/C# programming with ODBC **or** Python programming **or** other programming languages upon approval, and the use of database systems such as MariaDB **or** MySQL **or** PostGRES or Oracle.


FUNDAMENTALS OF
DATABASE
SYSTEMS
7TH Edition
ELMASRI • NAVATHE

CHAPTER 1

Databases and Database Users

OUTLINE

- Types of Databases and Database Applications
- Basic Definitions
- Typical DBMS Functionality
- Example of a Database (UNIVERSITY)
- Main Characteristics of the Database Approach
- Types of Database Users
- Advantages of Using the Database Approach
- Historical Development of Database Technology
- Extending Database Capabilities
- When Not to Use Databases

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
- First part of book focuses on traditional applications
- *A number of recent applications are described later in the book (for example, Chapters 24,25,26,27,28,29)*

Recent Developments (1)

- 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 (2)

- 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 (Chapter 25)
- **NOSQL** (Non-SQL, Not Only SQL) systems (Chapter 24)
- A large amount of data now resides on the “cloud” which means it is in huge data centers using thousands of machines.

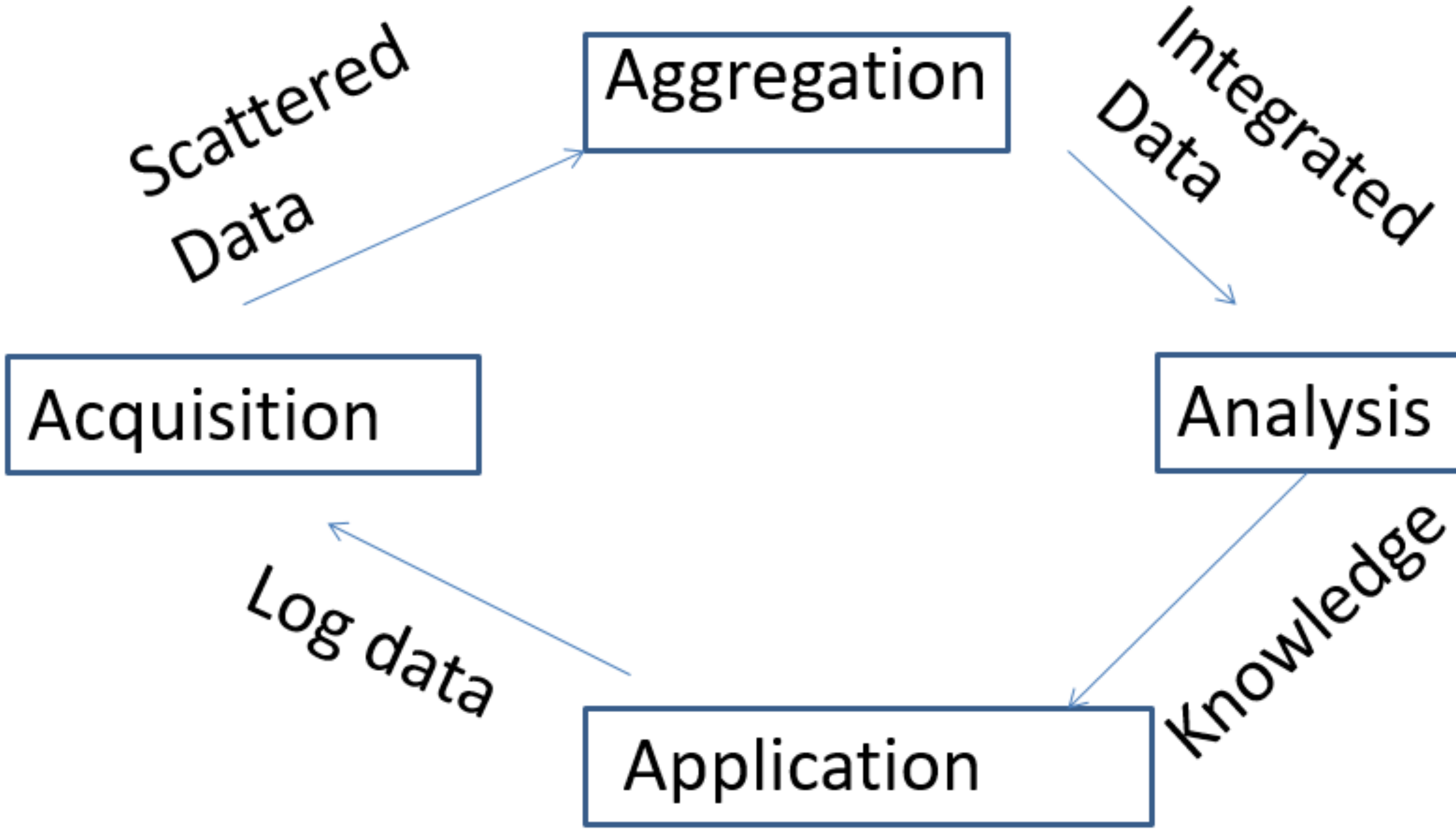
What is “big data”?

- "Big Data are high-volume, high-velocity, and/or high-variety information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization” (Gartner 2012)
- Bottom line: Any data that exceeds our current capability of processing can be regarded as “big”
  - Complicated (intelligent) analysis of data may make a small data “appear” to be “big”

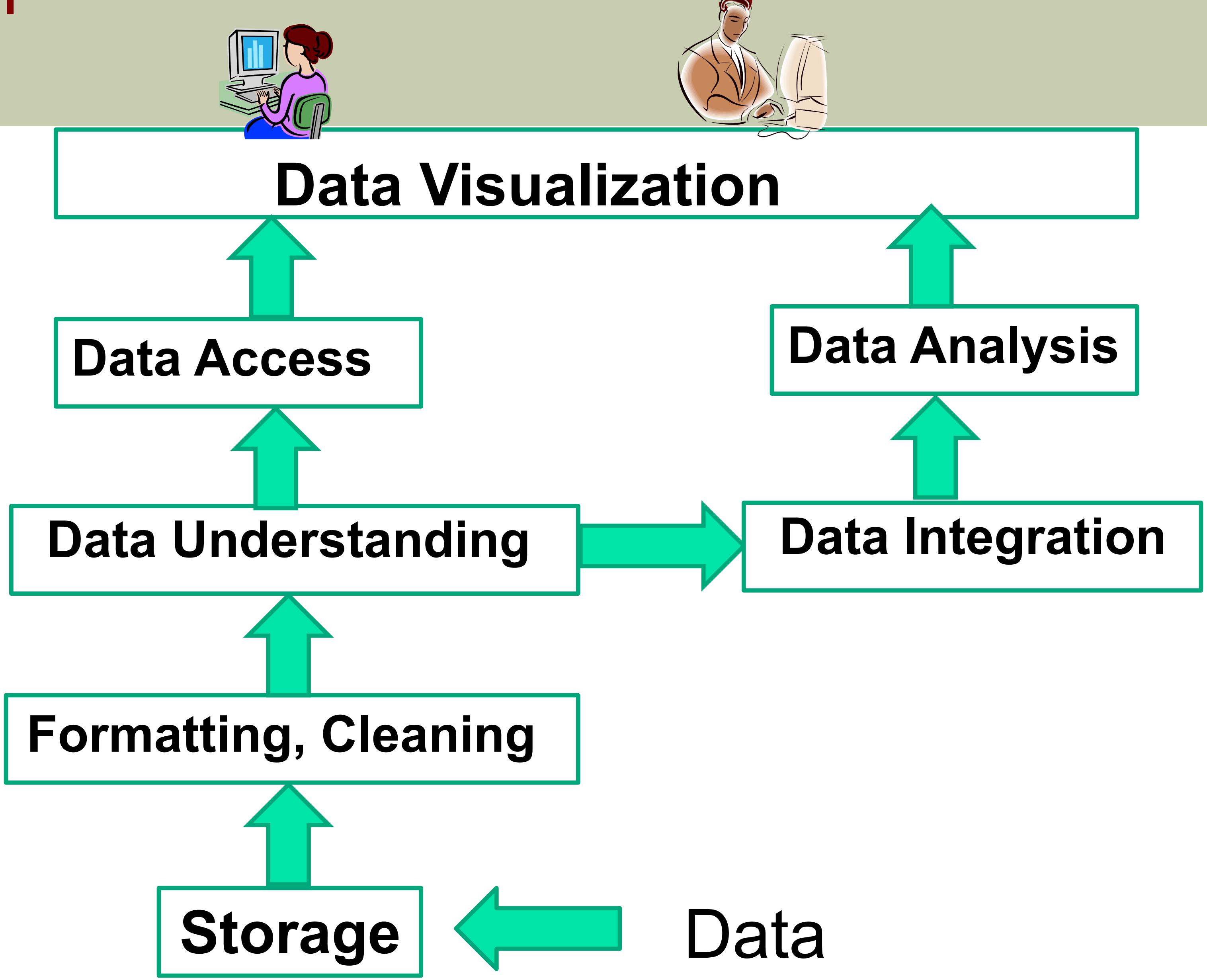
Why is “big data” a “big deal”?

- Government
- Private Sector
  - **Walmart handles more than 1 million customer transactions every hour, which is imported into databases estimated to contain more than 2.5 petabytes of data**
  - **Facebook handles 40 billion photos from its user base**
  - Falcon Credit Card Fraud Detection System protects 2.1 billion active accounts world-wide
- Science
  - Large Synoptic Survey Telescope will generate 140 Terabyte of data every 5 days
  - Biomedical computation like decoding human Genome and personalized medicine

Lifecycle of Data: 4 "A"s



Computational View of Big Data



Data Visualization

Data Access

Data Analysis

Data Understanding

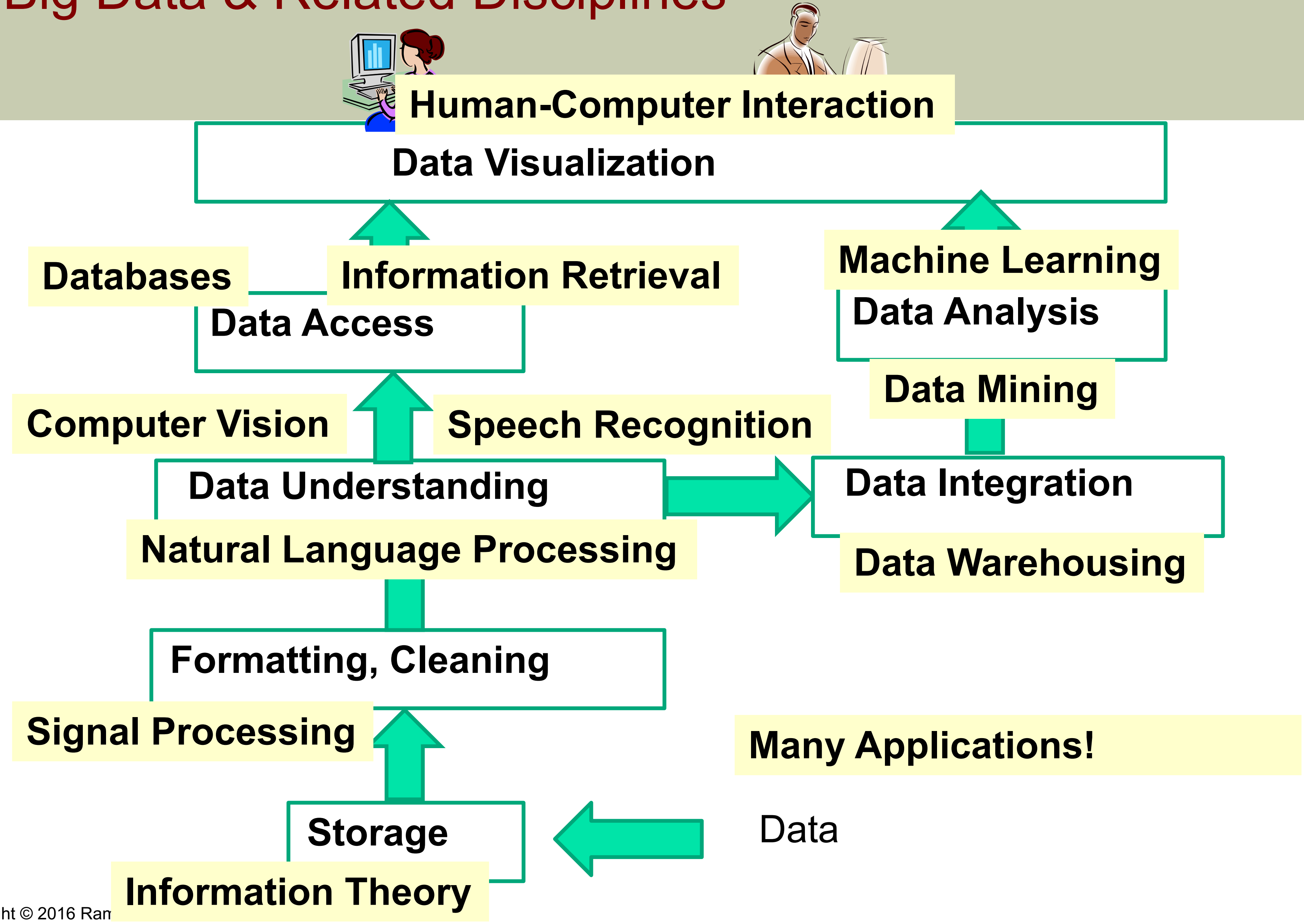
Data Integration

Formatting, Cleaning

Storage

Data

Big Data & Related Disciplines



Human-Computer Interaction

Data Visualization

Databases

Information Retrieval

Machine Learning

Data Access

Data Analysis

Data Mining

Computer Vision

Speech Recognition

Data Understanding

Data Integration

Natural Language Processing

Data Warehousing

Formatting, Cleaning

Signal Processing

Many Applications!

Storage

Data

Information Theory

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.

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

Simplified database system environment

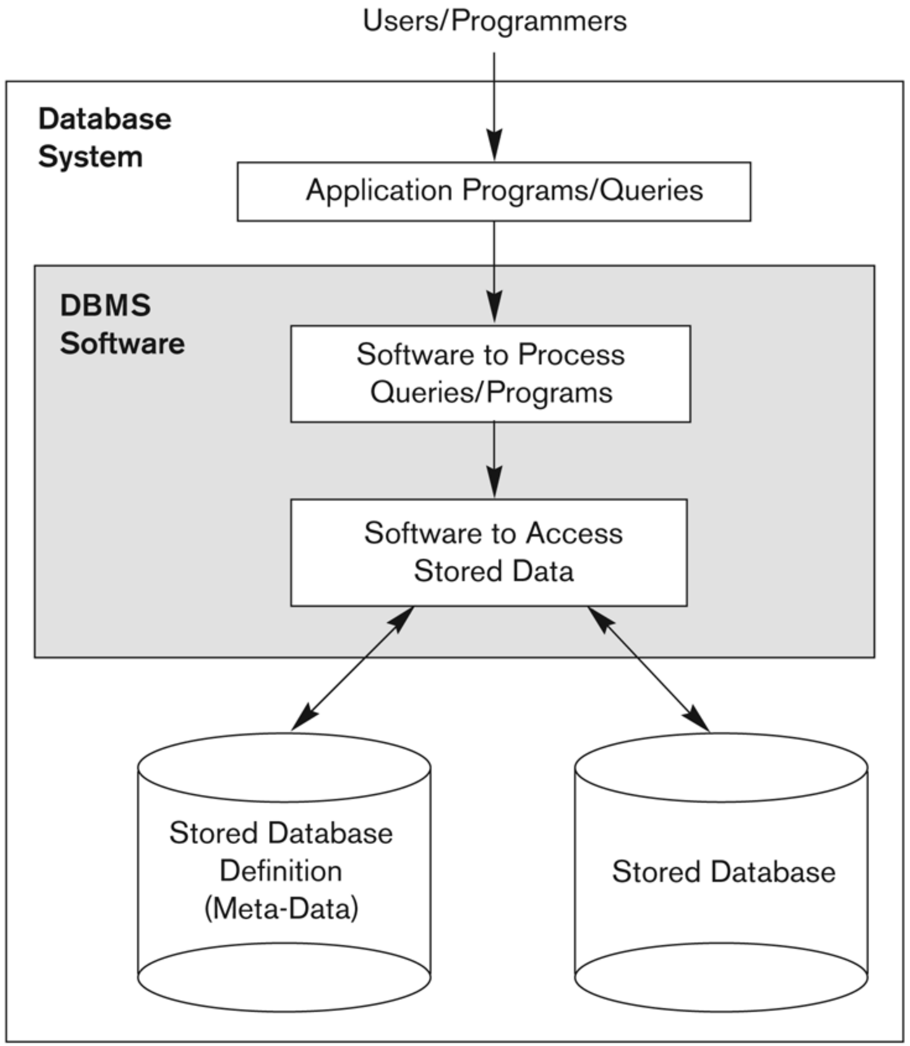

Users/Programmers
Database System
Application Programs/Queries
DBMS Software
Software to Process Queries/Programs
Software to Access Stored Data
Stored Database Definition (Meta-Data)
Stored Database

Figure 1.1
A simplified database system environment.

Basic Definitions

- **Database:**
  - A collection of related data items within a specific business process or problem setting (Has a target group of users and applications)
- **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.

Simplified database system environment

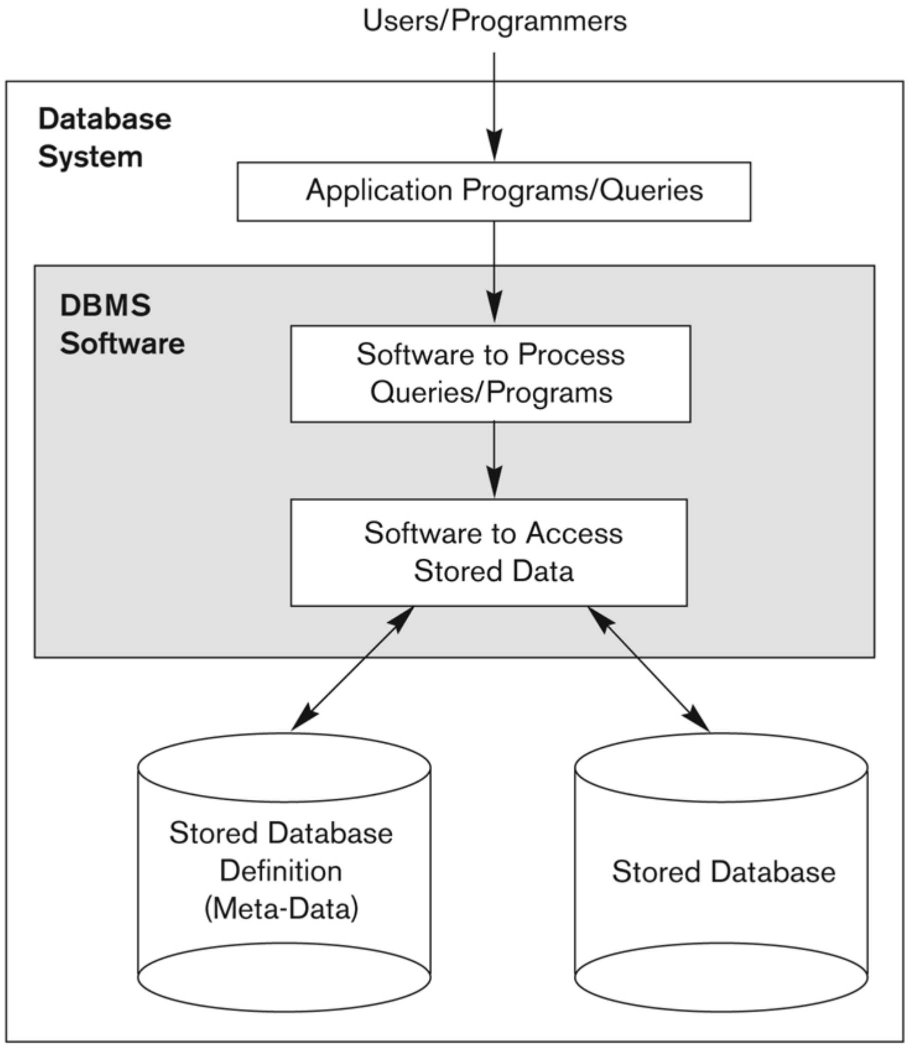

Users/Programmers
Database System
Application Programs/Queries
DBMS Software
Software to Process Queries/Programs
Software to Access Stored Data
Stored Database Definition (Meta-Data)
Stored Database

Figure 1.1
A simplified database system environment.

Basic Definitions

- Design of a new application for an existing database or design of a brand new database starts off with a phase called **requirements specification and analysis.**
- These requirements are documented in detail and transformed into a **conceptual design** that can be represented and manipulated using some computerized tools so that it can be easily maintained, modified, and transformed into a database implementation (ERD).
- The design is then translated to a **logical design** that can be expressed in a data model implemented in a commercial DBMS.
- The final stage is **physical design**, during which further specifications are provided for storing and accessing the database. The database design is implemented, populated with actual data, and continuously maintained to reflect the state of the miniworld.

Basic Definitions

SQL (/ˌɛsˌkjuːˈɛl/), *”SeQueL"* (/ˈsiːkwəl/)

Structured Query Language

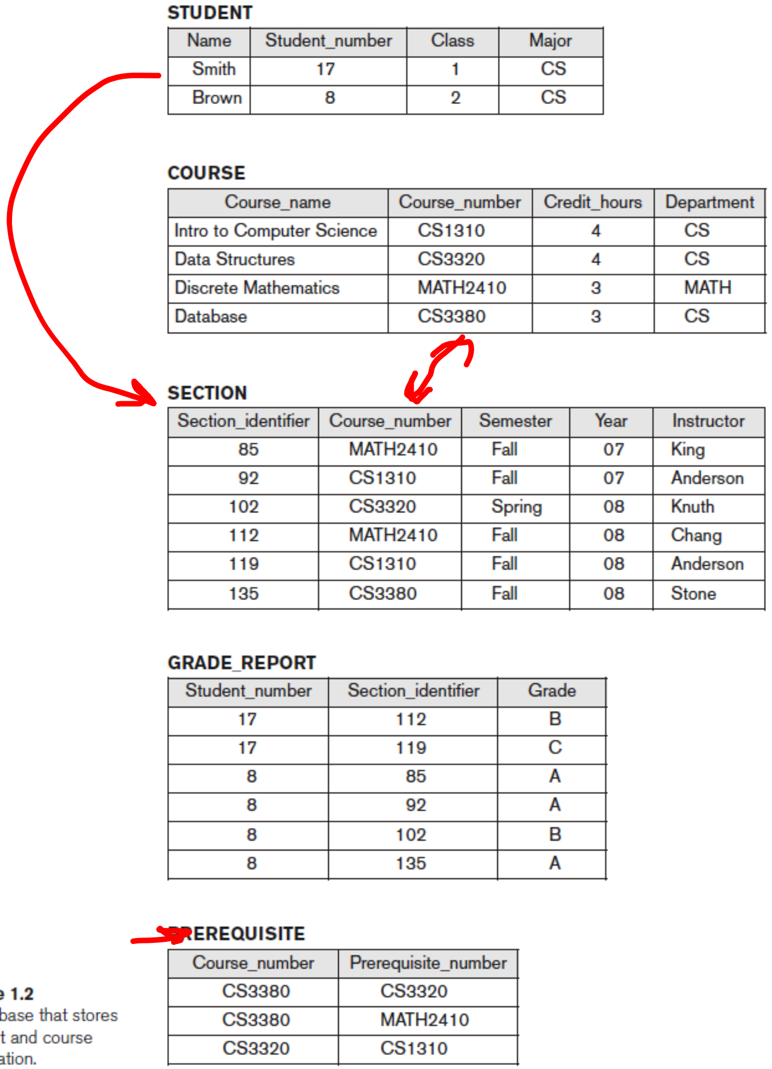
Source: https://en.wikipedia.org/wiki/SQL

SEQUEL
/ˈsi:kwəl/
(Structured English QUEry Language)
S-Q-L or SEQUEL?
SQL
/ˈɛs kjuː ˈɛl/
Khan Academy
(Structured Query Language)

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 Simple Database



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.

Example of a Simple Database (Cont)

- A **Relation** is a Table

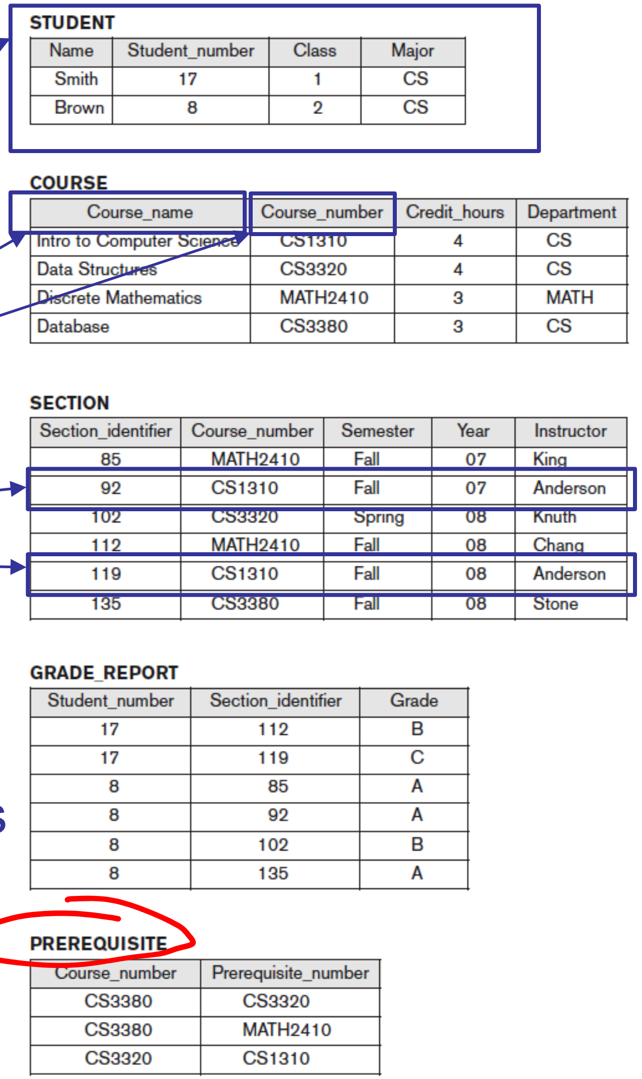
  "STUDENT is a table"

  Attributes(column headers)

  Tuples (rows)

- Relationships:

  COURSEs **have** prerequisite COURSEs



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.

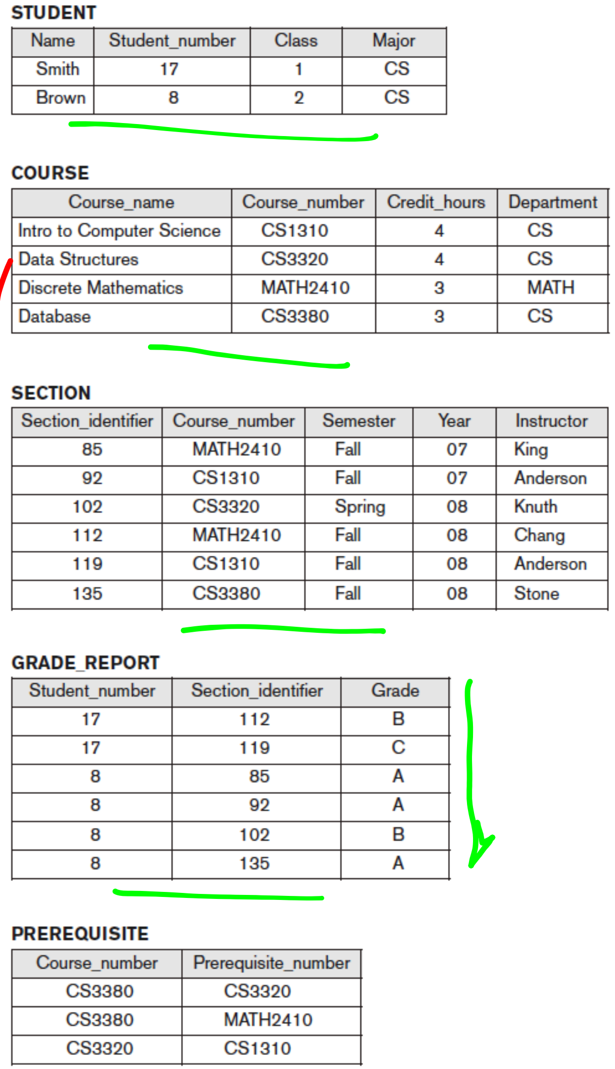
Example of a Database (with a Conceptual Data Model)

- **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 and Queries

SQL Examples:

```
Select     Name
From       STUDENT


Select     Grade
From       GRADE_REPORT


Select     Name, Grade
From       STUDENT,GRADE_REPORT
Where      STUDENT.Student_number = GRADE_REPORT.Student_number


Select     Name, Grade
From       STUDENT,GRADE_REPORT
Where      STUDENT.Student_number = GRADE_REPORT.Student_number
           and STUDENT.Name = 'Smith'
```



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.

Simplified database system environment

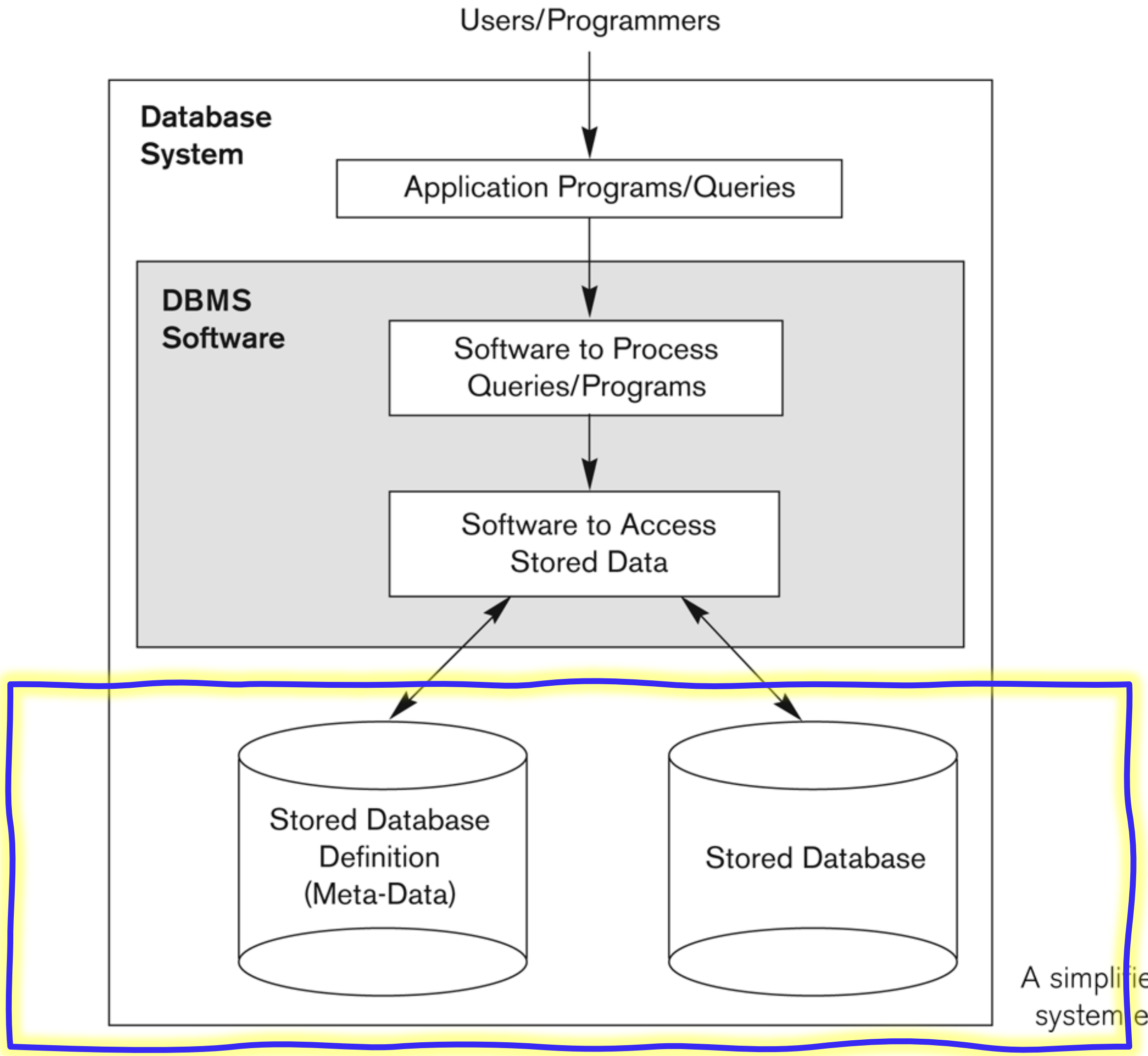
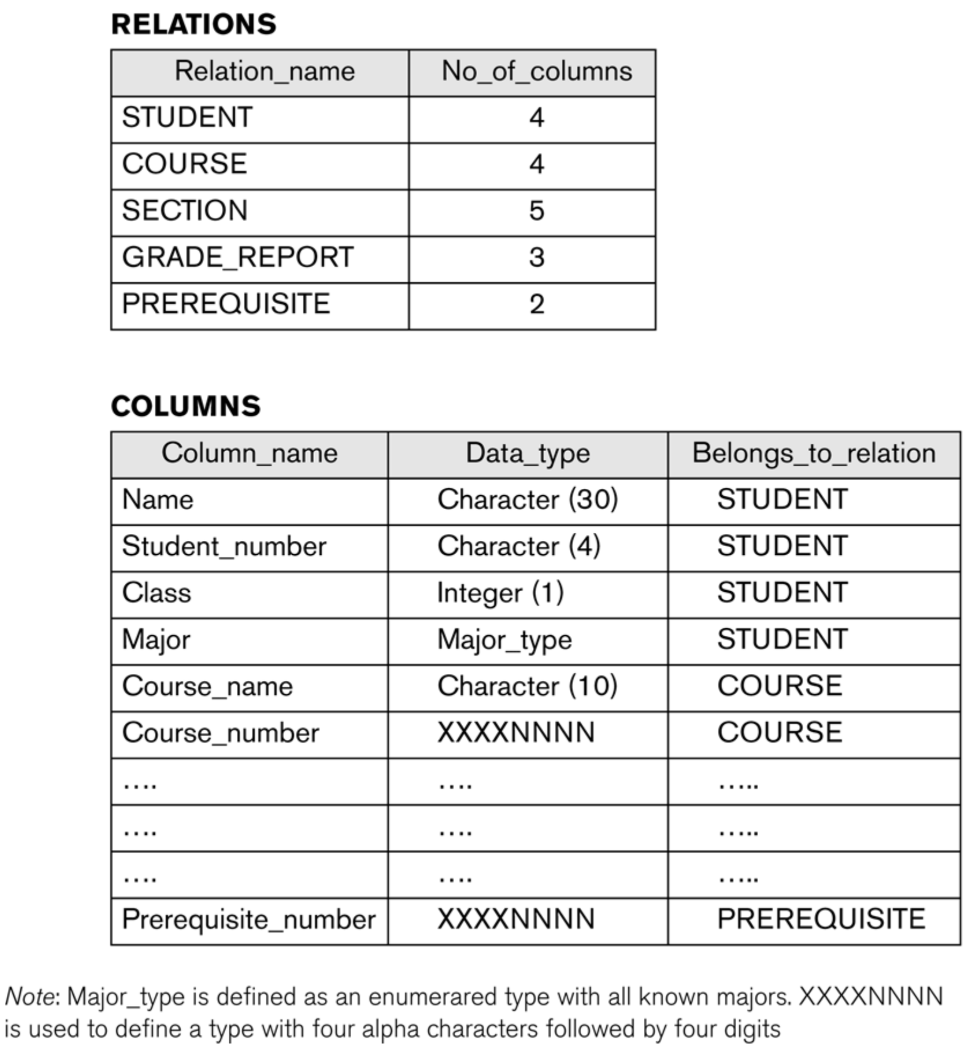

Users/Programmers
Database System
Application Programs/Queries
DBMS Software
Software to Process Queries/Programs
Software to Access Stored Data
Stored Database Definition (Meta-Data)
Stored Database

Figure 1.1
A simplified database system environment.

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

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

Figure 1.3
An example of a database catalog for the database in Figure 1.2.

Main Characteristics of the Database Approach (continued)

- **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.

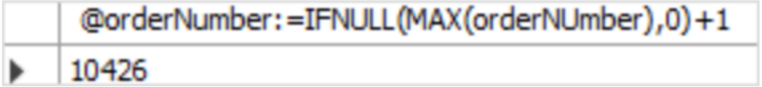
Main Characteristics of the Database Approach (continued)

- **Sharing of data and multi-user transaction processing:**
  - **Transaction** is an executing program or process that includes one or more database accesses, such as reading or updating of database records.
  - The **isolation** property ensures that each transaction appears to execute in isolation from other transactions, even though hundreds of transactions may be executing concurrently.
  - The **atomicity** property ensures that either all the database operations in a transaction are executed or none are.
  - ***Recovery* subsystem** ensures each completed transaction has its effect permanently recorded in the database

Transaction Example

```
-- 1. start a new transaction
START TRANSACTION;

-- 2. Get the latest order number
SELECT
    @orderNumber:=MAX(orderNUmber)+1
FROM
    orders;

-- 3. insert a new order for customer 145
INSERT INTO orders(orderNumber,
                   orderDate,
                   requiredDate,
                   shippedDate,
                   status,
                   customerNumber)
VALUES(@orderNumber,
       '2005-05-31',
       '2005-06-10',
       '2005-06-11',
       'In Process',
        145);

-- 4. Insert order line items
INSERT INTO orderdetails(orderNumber,
                         productCode,
                         quantityOrdered,
                         priceEach,
                         orderLineNumber)
VALUES(@orderNumber,'S18_1749', 30, '136', 1),
      (@orderNumber,'S18_2248', 50, '55.09', 2);

-- 5. commit changes
COMMIT;
```

Source: http://www.mysqltutorial.org/mysql-transaction.aspx

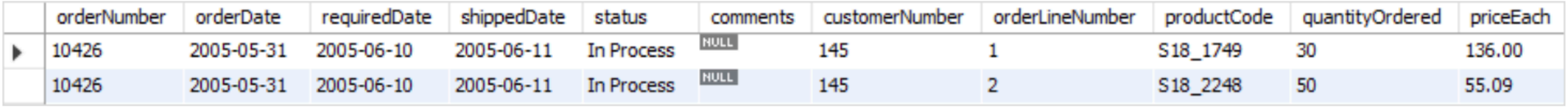


```
@orderNumber:=IFNULL(MAX(orderNUmber),0)+1
10426
```

Query Check

Source: http://www.mysqltutorial.org/mysql-transaction.aspx

Here is the output:



	orderNumber	orderDate	requiredDate	shippedDate	status	comments	customerNumber	orderLineNumber	productCode	quantityOrdered	priceEach
▶	10426	2005-05-31	2005-06-10	2005-06-11	In Process	NULL	145	1	S18_1749	30	136.00
	10426	2005-05-31	2005-06-10	2005-06-11	In Process	NULL	145	2	S18_2248	50	55.09

Main Characteristics of the Database Approach (continued)

- **Sharing of data and multi-user transaction processing:**
  - Allowing a set of **concurrent users** to retrieve from and to update the database. **Concurrency control** software guarantees if several users trying to update the same data that each update will correctly executed or aborted.
  - **OLTP** (Online Transaction Processing) is a major part of database applications. This allows hundreds of concurrent transactions to execute per second.

Simplified database system environment

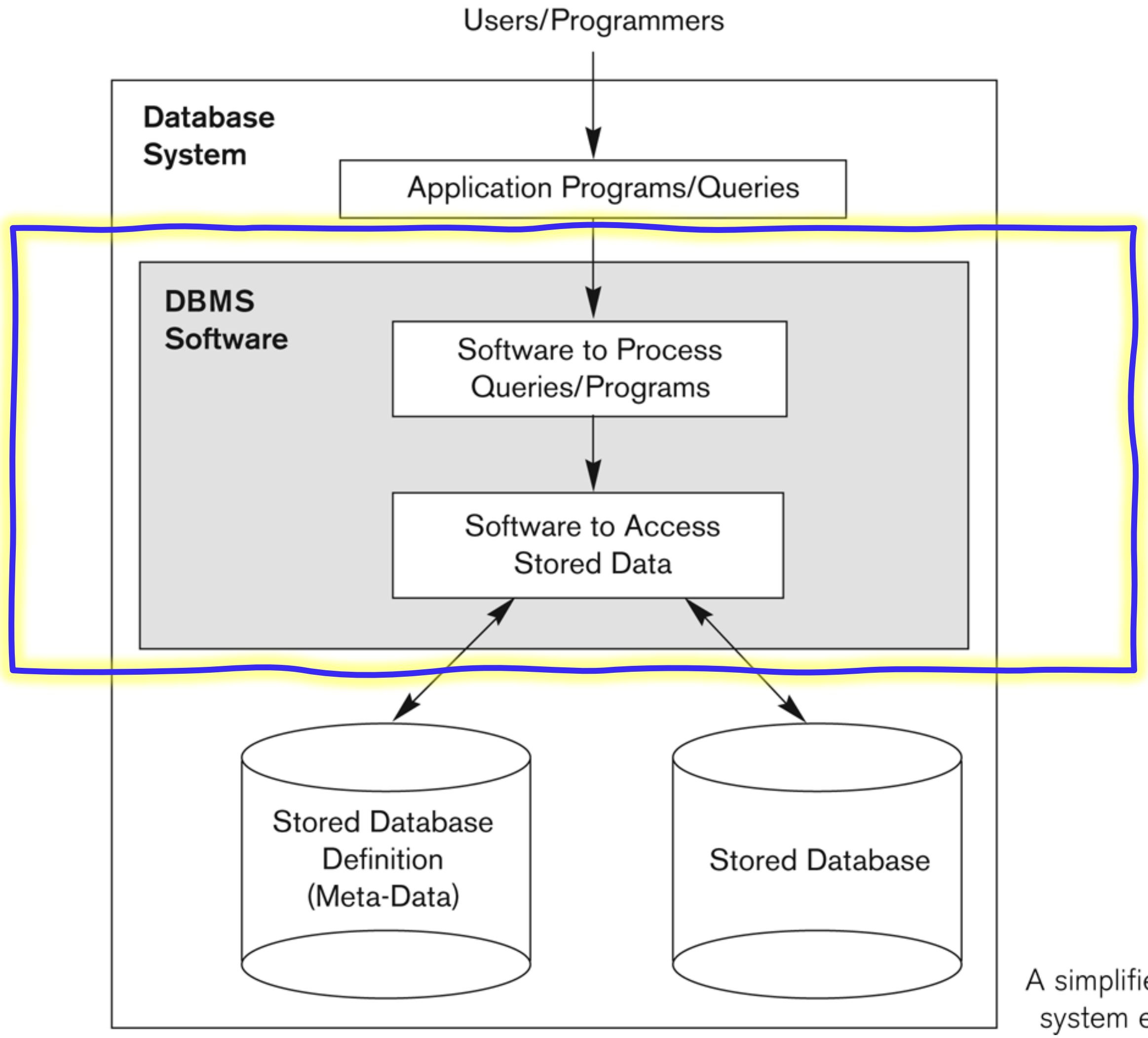

Users/Programmers
Database System
Application Programs/Queries
DBMS Software
Software to Process Queries/Programs
Software to Access Stored Data
Stored Database Definition (Meta-Data)
Stored Database

Figure 1.1
A simplified database system environment.

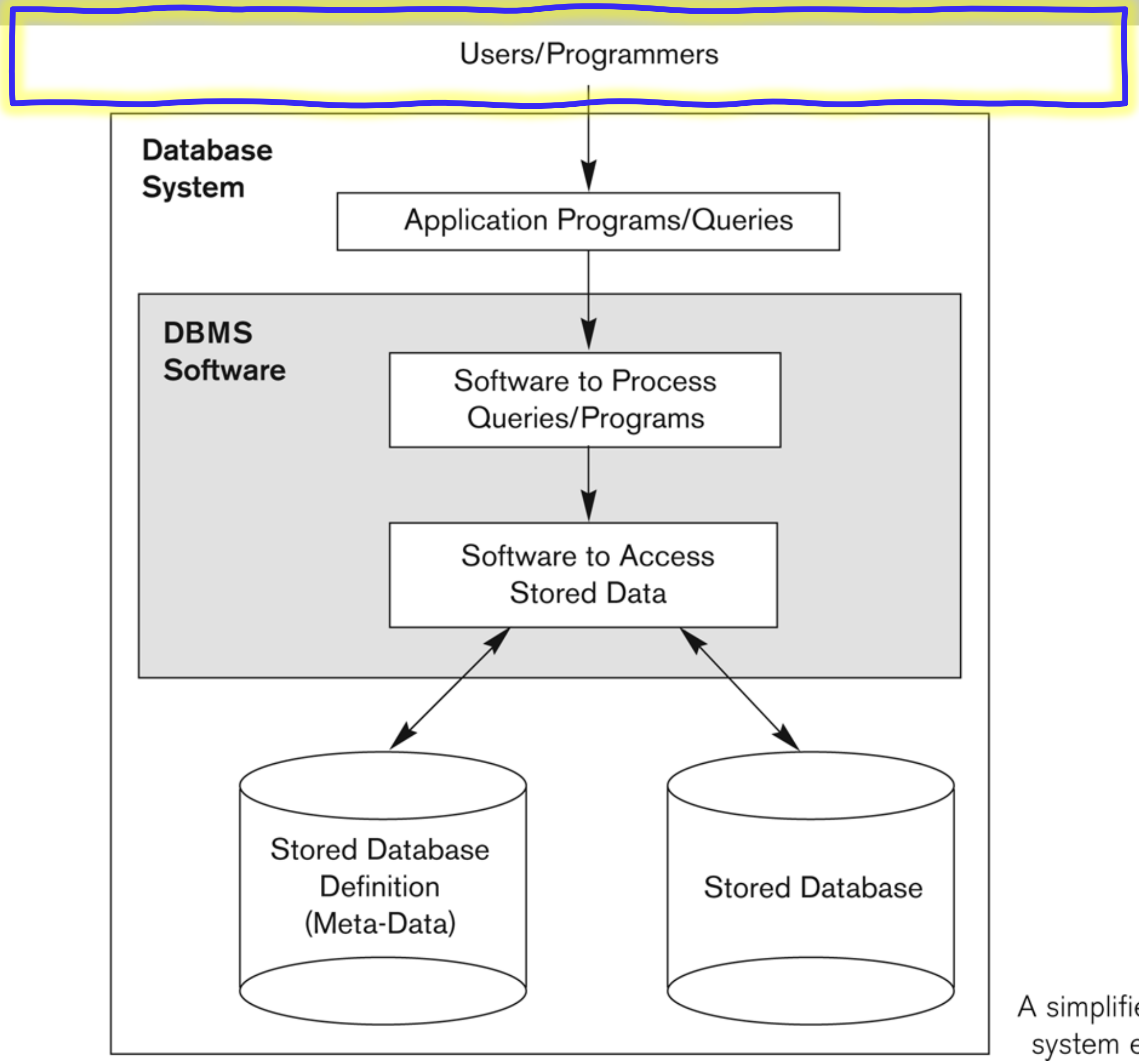
What a DBMS Facilitates

- ***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

Other DBMS Functionalities

- DBMS may additionally provide:
  - **Protection** against hardware or software malfunction (or **crashes**) and
  - Security **protection** against unauthorized or **malicious** access
  - Presentation and **visualization** of data
  - **Maintenance** of the database and associated programs over the lifetime of the database application

Simplified database system environment



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

Figure 1.1
A simplified database system environment.

Database System 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 System 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 System End Users (continued)

- 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
      - Occasionally access the database, but they may need different information each time.
      - They use a sophisticated database query interface to specify their requests and are typically middle - or high-level managers or other occasional browsers.

Database System End Users

- Actors on the scene (continued)
  - **End-users:**
    - **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 System End Users (continued)

- Actors on the scene (continued)
  - **End-users:**
    - **Sophisticated:**
      - These include business analysts, scientists, engineers, others thoroughly familiar with the system capabilities capable to implement their own applications.
      - Many use tools in the form of software packages that work closely with the stored database.

Database System End Users (continued)

- Actors on the scene (continued)
  - **End-users:**
    - **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 System Users – Actors on the Scene (continued)

- **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 (software developers/engineers).

Such analysts and programmers—commonly referred to as **software developers** or **software engineers**—should be familiar with the full range of capabilities provided by the DBMS to accomplish their tasks

Database System Users – Workers behind the Scene

- **System Designers and Implementors:** Design and implement DBMS **packages** in the form of modules and interfaces and test and debug them. A module could be to implement a catalog, processing query language, accessing and buffering data, controlling concurrency, and handling data recovery and security. 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.

Examples:

MS SQL Profiler: (https://www.youtube.com/watch?v=5fLsrRAtTJA)

SAP Crystal Reports: (https://www.youtube.com/watch?v=oohyamfsgkE)

Database System Users – Workers behind the Scene

- **Operators and Maintenance Personnel:** They manage the actual running and maintenance of the database system hardware and software environment.