Database Systems (CS2005)

Week – 01

Instructor: **Basit Ali**

Course Outline

Uploaded on GCR

GCR code: **svvabqix**

Book

FUNDAMENTALS OF Database Systems
SEVENTH EDITION

By

Ramez Elmasri & Shamkant B. Navathe

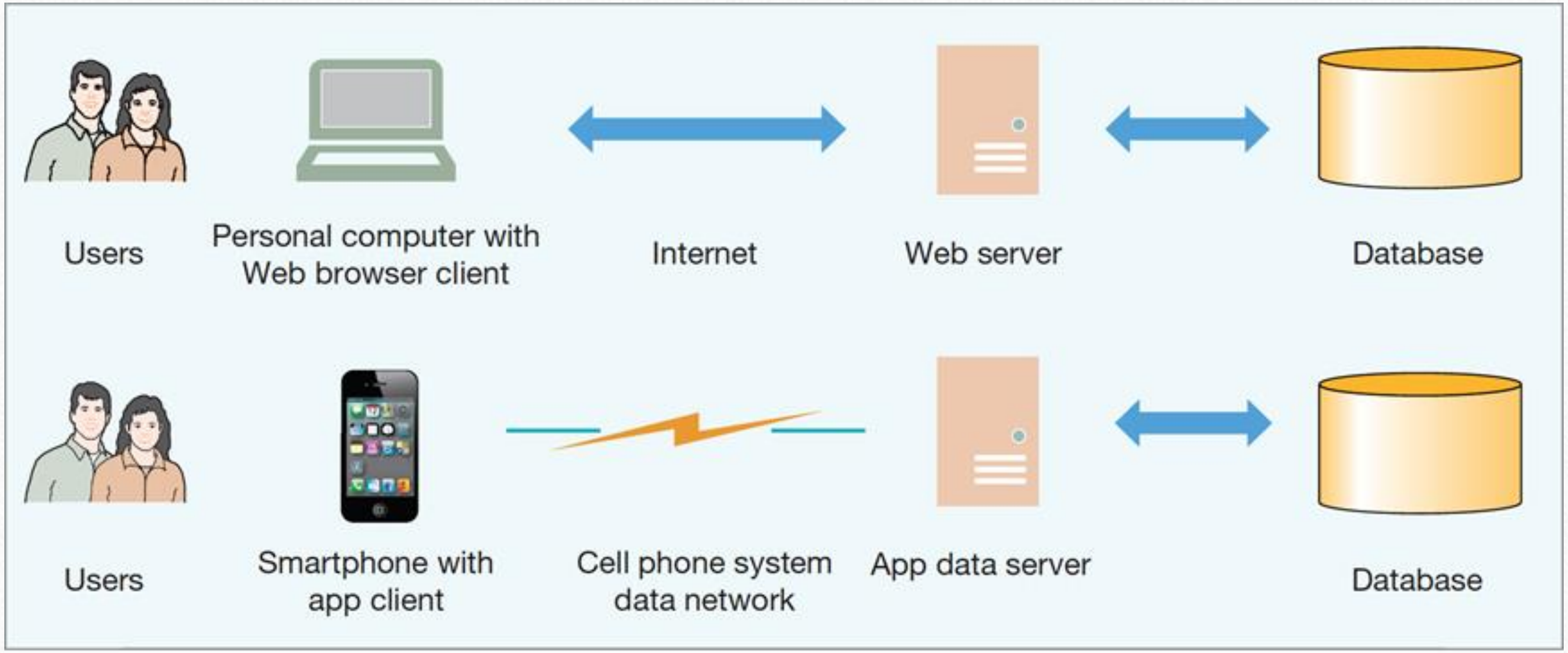
Assessments / Grading

Mid-1	15
Mid-2	15
Assignments	10
Project	10
Final Exam	50

Chapter #1

Databases and Database Users

Users Are Dependent on Databases



Purpose of a Database

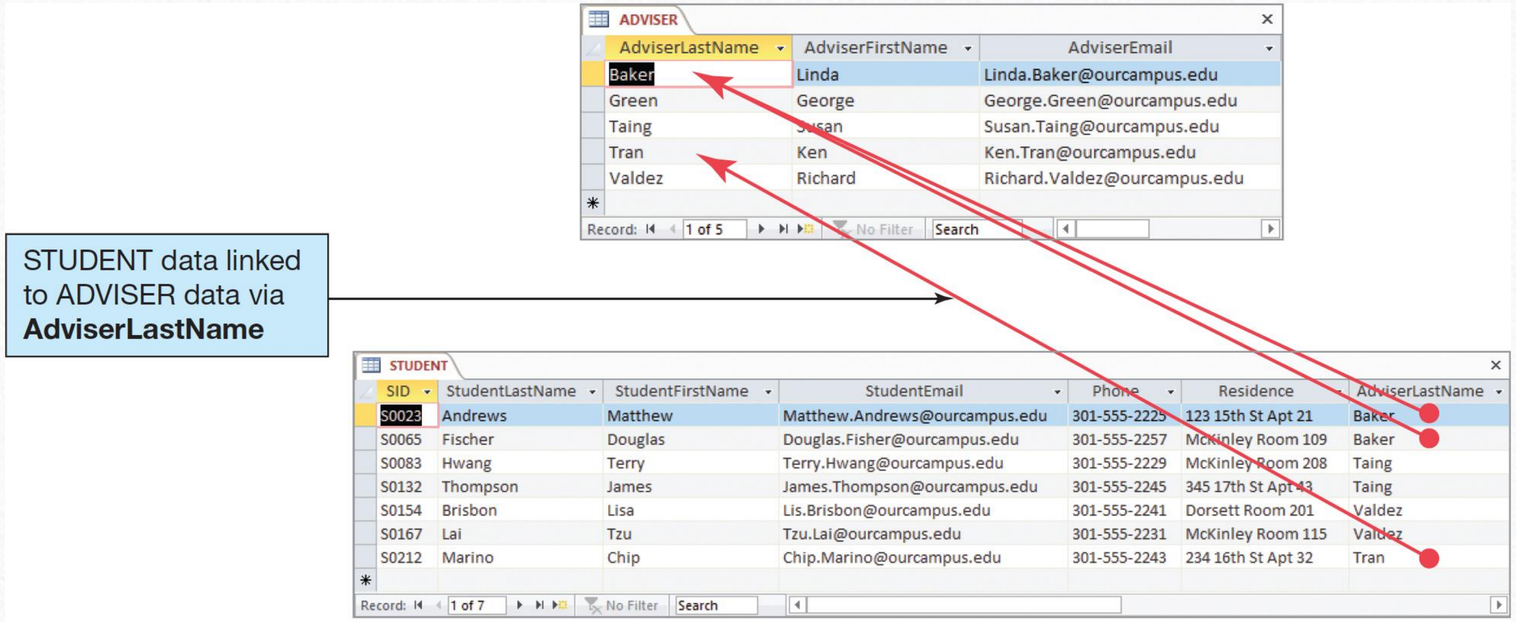
- The purpose of a database is to keep track of things.
- Unlike a list or spreadsheet, a database may store more complicated information than a simple list.

Relational Database Tables

- A relational database stores data in tables.
- Each table holds data about one single theme.
- In a relational database, tables are joined together using the value of the data.
- Relational databases minimize data redundancy, preserve complex relationships, and allow for partial data.

Relational Database Example

The Adviser and Student Tables



AdviserLastName	AdviserFirstName	AdviserEmail
Baker	Linda	Linda.Baker@ourcampus.edu
Green	George	George.Green@ourcampus.edu
Taing	Susan	Susan.Taing@ourcampus.edu
Tran	Ken	Ken.Tran@ourcampus.edu
Valdez	Richard	Richard.Valdez@ourcampus.edu

Record: 1 of 5

SID	StudentLastName	StudentFirstName	StudentEmail	Phone	Residence	AdviserLastName
S0023	Andrews	Matthew	Matthew.Andrews@ourcampus.edu	301-555-2225	123 15th St Apt 21	Baker
S0065	Fischer	Douglas	Douglas.Fisher@ourcampus.edu	301-555-2257	McKinley Room 109	Baker
S0083	Hwang	Terry	Terry.Hwang@ourcampus.edu	301-555-2229	McKinley Room 208	Taing
S0132	Thompson	James	James.Thompson@ourcampus.edu	301-555-2245	345 17th St Apt 43	Taing
S0154	Brisbon	Lisa	Lis.Brisbon@ourcampus.edu	301-555-2241	Dorsett Room 201	Valdez
S0167	Lai	Tzu	Tzu.Lai@ourcampus.edu	301-555-2231	McKinley Room 115	Valdez
S0212	Marino	Chip	Chip.Marino@ourcampus.edu	301-555-2243	234 16th St Apt 32	Tran

Record: 1 of 7

Access 2016, Windows 10, Microsoft Corporation.

The Department, Adviser, and Student Tables

Can insert DEPARTMENT data as needed—no ADVISER or STUDENT data required

DEPARTMENT

DeartmentName	DepartmentPhone	AdminLastName	AdminFirstName	AdminEmail
Accounting	301-557-1011	Smith	Shawna	Shawna.Smith@ourcampus.edu
Biology	301-557-1021	Kelly	Chris	Chris.Kelly@ourcampus.edu
Chemistry	301-557-1031	Chaplin	Robin	Robin.Chaplin@ourcampus.edu
InfoSystems	301-557-1041	Rogers	Aaron	Aaron.Rogers@ourcampus.edu

Record: 1 of 4 No Filter Search

ADVISER

AdviserLastName	AdviserFirstName	AdviserEmail	Department
Baker	Linda	Linda.Baker@ourcampus.edu	Accounting
Green	George	George.Green@ourcampus.edu	Biology
Taing	Susan	Sue.Taing@ourcampus.edu	Accounting
Tran	Ken	Ken.Tran@ourcampus.edu	InfoSystems
Valdez	Richard	Richard.Valdez@ourcampus.edu	Chemistry
Yeats	Bill	Bill.Yeats@ourcampus.edu	InfoSystems

Record: 1 of 6 No Filter Search

Can change STUDENT Adviser name as needed—new value is linked to its own data

Can delete STUDENT data as needed—no DEPARTMENT or ADVISER data lost

STUDENT

SID	StudentLastName	StudentFirstName	StudentEmail	Phone	Residence	AdviserLastName
S0023	Andrews	Matthew	Matthew.Andrews@ourcampus.edu	301-555-2225	123 15th St Apt 21	Baker
S0065	Fischer	Douglas	Douglas.Fisher@ourcampus.edu	301-555-2257	McKinley Room 109	Baker
S0083	Hwang	Terry	Terry.Hwang@ourcampus.edu	301-555-2229	McKinley Room 208	Taing
S0132	Thompson	James	James.Thompson@ourcampus.edu	301-555-2245	345 17th St Apt 43	Taing
S0154	Brisbon	Lisa	Lis.Brisbon@ourcampus.edu	301-555-2241	Dorsett Room 201	Valdez
S0167	Lai	Tzu	Tzu.Lai@ourcampus.edu	301-555-2231	McKinley Room 115	Valdez
S0212	Marino	Chip	Chip.Marino@ourcampus.edu	301-555-2243	234 16th St Apt 32	Tran

Record: 7 of 7 No Filter Search

Access 2016, Windows 10, Microsoft Corporation.

The Art Course Database Tables

CUSTOMER

CustomerNumber	CustomerLastName	CustomerFirstName	Phone
1	Johnson	Ariel	206-567-1234
2	Green	Robin	425-678-8765
3	Jackson	Charles	360-789-3456
4	Pearson	Jeffery	206-567-2345
5	Sears	Miguel	360-789-4567
6	Kyle	Leah	425-678-7654
7	Myers	Lynda	360-789-5678
(New)			

Record: 1 of 7 No Filter Search

Can change COURSE CourseDate without problems

COURSE

CourseNumber	Course	CourseDate	Fee
1	Adv Pastels	10/1/2017	$500.00
2	Beg Oils	9/15/2017	$350.00
3	Int Pastels	3/15/2017	$350.00
4	Beg Oils	10/15/2017	$350.00
5	Adv Pastels	11/15/2017	$500.00
(New)			$0.00

Record: 1 of 5 No Filter Search

Can insert new COURSE data as needed

ENROLLMENT

CustomerNumber	CourseNumber	AmountPaid
1	1	$250.00
1	3	$350.00
2	2	$350.00
3	1	$500.00
4	1	$500.00
5	2	$350.00
6	5	$250.00
7	4	$0.00
0	0	$0.00

Record: 1 of 8 No Filter Search

Can delete ENROLLMENT rows as needed—no adverse consequences

Relational Tables are Processed by Structured Query Language (SQL)

- **Structured Query Language** (SQL) is an international standard for creating, processing, and querying databases and their tables.
- Many databases use SQL to retrieve, format, report, insert, delete, and/or modify data for users.

SQL Example (1 of 2)

- We can use SQL to combine the data in the three tables in the Art Course Database to recreate the original list structure of the data that was in the spreadsheet.
- We do this by using an SQL SELECT statement as shown on the next slide.

SQL Example (2 of 2)

```
SELECT    CUSTOMER.CustomerLastName,
          CUSTOMER.CustomerFirstName, CUSTOMER.Phone,
          COURSE.CourseDate, ENROLLMENT.AmountPaid,
          COURSE.Course, COURSE.Fee
FROM      CUSTOMER, ENROLLMENT, COURSE
WHERE     CUSTOMER.CustomerNumber = ENROLLMENT.
          CustomerNumber
   AND    COURSE.CourseNumber = ENROLLMENT.CourseNumber;
```

- The results of this SQL code can be seen on the next page.

SQL Example Results for the Art Course Database

Results of the SQL Query to Recreate the Art Course List Data

Art Course List

CustomerLastName	CustomerFirstName	Phone	CourseDate	AmountPaid	Course	Fee
Johnson	Ariel	206-567-1234	10/1/2017	$250.00	Adv Pastels	$500.00
Johnson	Ariel	206-567-1234	3/15/2017	$350.00	Int Pastels	$350.00
Green	Robin	425-678-8765	9/15/2017	$350.00	Beg Oils	$350.00
Jackson	Charles	360-789-3456	10/1/2017	$500.00	Adv Pastels	$500.00
Pearson	Jeffery	206-567-2345	10/1/2017	$500.00	Adv Pastels	$500.00
Sears	Miguel	360-789-4567	9/15/2017	$350.00	Beg Oils	$350.00
Kyle	Leah	425-678-7654	11/15/2017	$250.00	Adv Pastels	$500.00
Myers	Lynda	360-789-5678	10/15/2017	$0.00	Beg Oils	$350.00

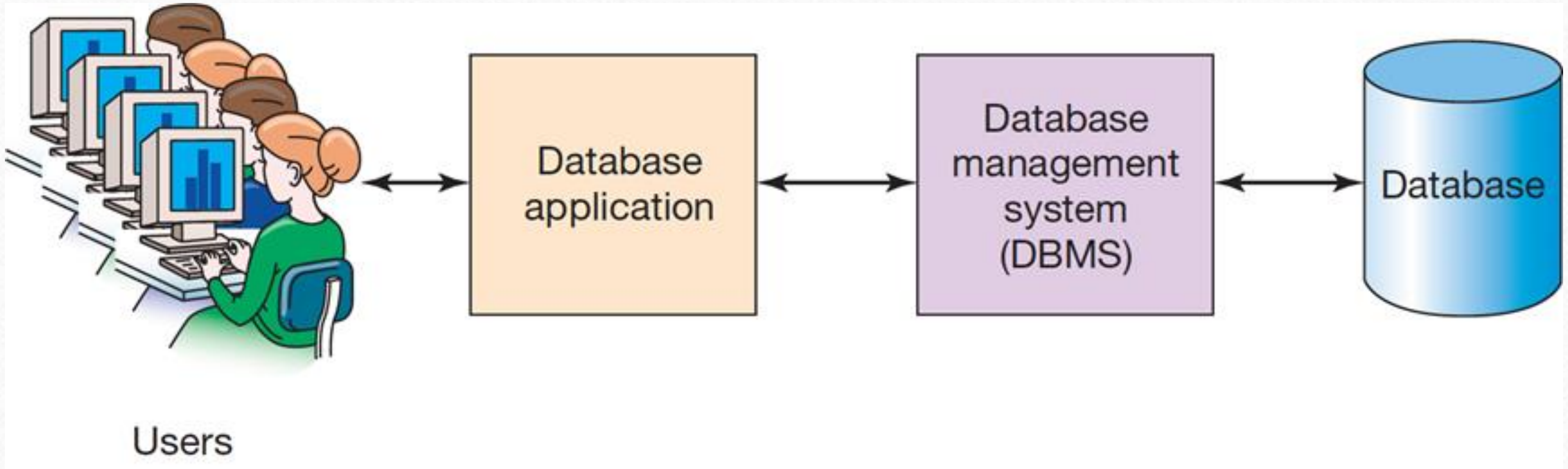
Record: 1 of 8 No Filter Search

Access 2016, Windows 10, Microsoft Corporation.

What is a Database System?

- A **database system** is comprised of four components:
  - Users
  - Database Application
  - Database Management System (DBMS)
  - Database

Components of a Database System



Users

- A **user** of a database system will:
  - Use a database application to track things
  - Use forms to enter, read, delete, and query data
  - Produce reports

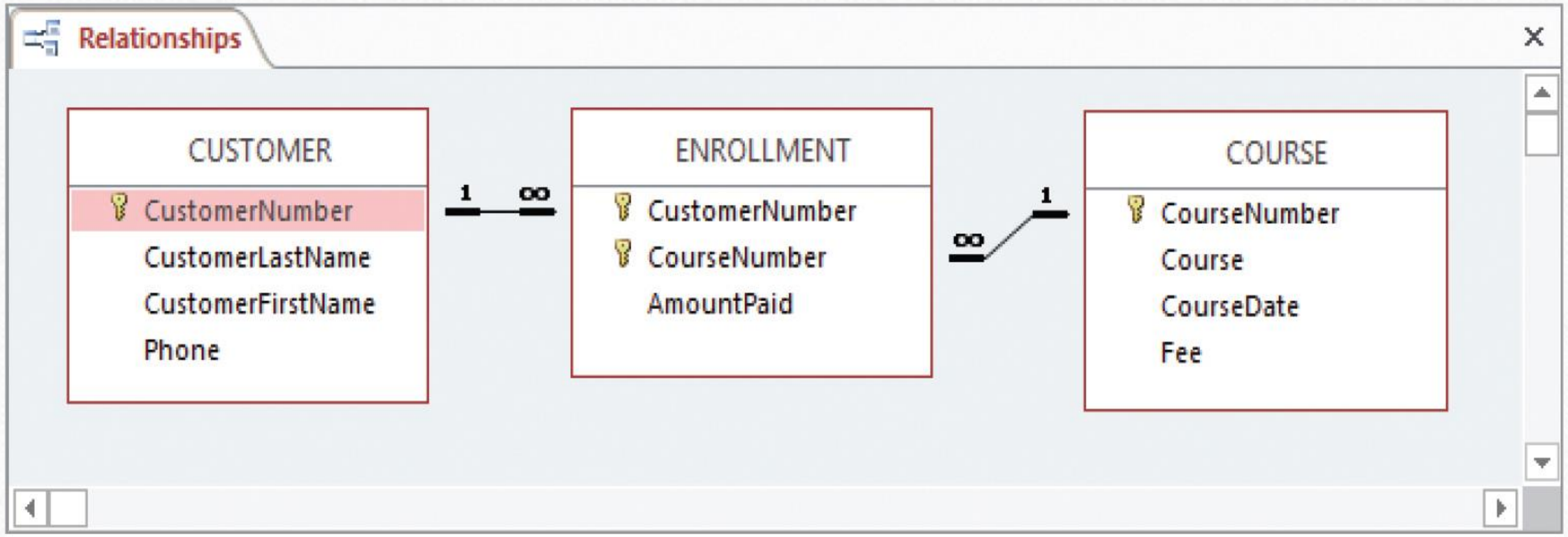
The Database

- A **database** is a self-describing collection of related tables.
  - The database itself contains the definition of its structure.
  - **Metadata** is data describing the structure of the database data.

- Tables within a relational database are related to each other.

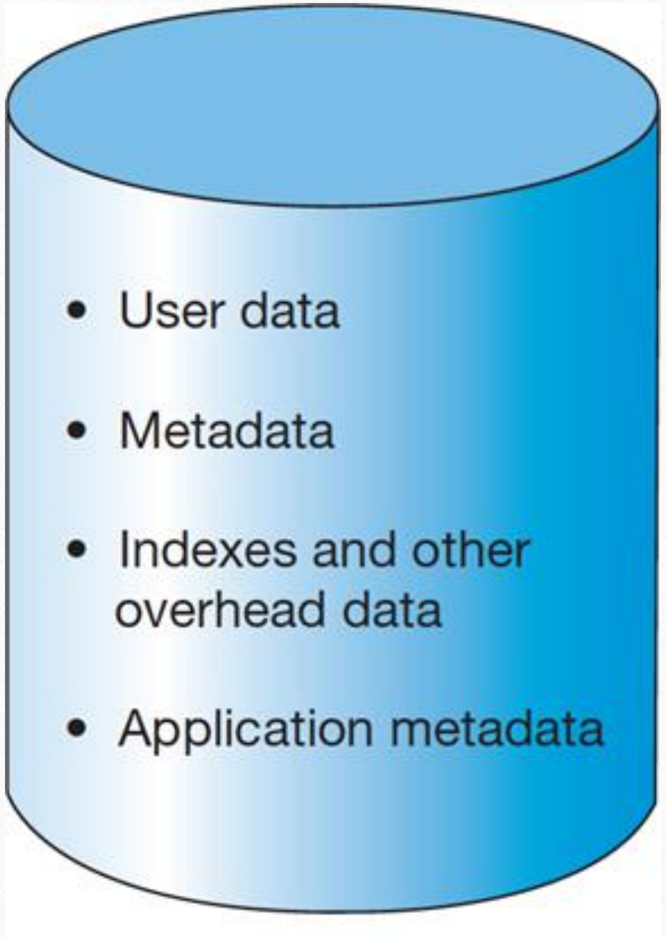
Example Database Metadata: A Relational Diagram

Example Metadata: A Relationship Diagram for the Art Course Tables



Access 2016, Windows 10, Microsoft Corporation.

Database Contents



Database Management System (DBMS)

- A database management system (DBMS) serves as an intermediary between database applications and the database.
- The DBMS manages and controls database activities.
- The DBMS creates, processes, and administers the databases it controls.

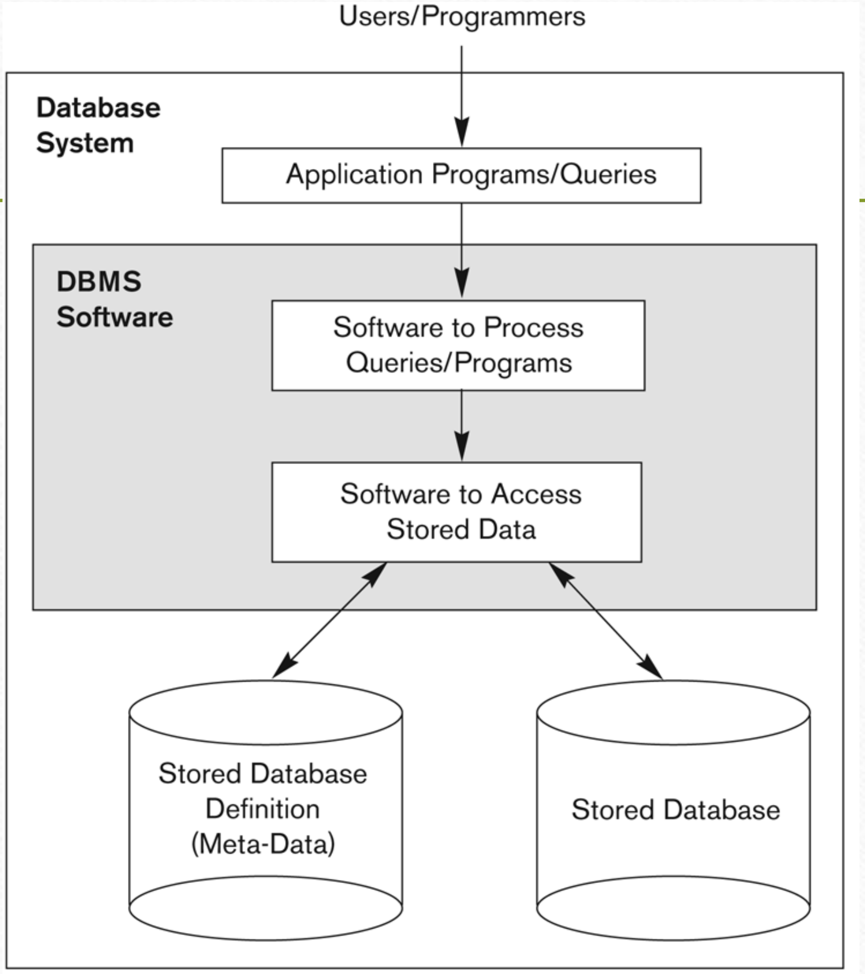
Functions of a DBMS

- Create database
- Create tables
- Create supporting structures (e.g., indexes)
- Read database data
- Modify (insert, update, or delete) database data
- Maintain database structures
- Enforce rules
- Control concurrency
- Provide security
- Perform backup and recovery

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



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

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

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)

- **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

- Note: The above entities and relationships are typically expressed in a conceptual data model, such as the ENTITY-RELATIONSHIP data model (see Chapters 3, 4)

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

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 enumerated 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 (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:**
  - 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 (continued)

- **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 (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.
  - 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
  - E.g., Object-oriented DBMSs make program objects persistent– see Chapter 12.
- Providing Storage Structures (e.g. indexes) for efficient Query Processing – see Chapter 17.

Advantages of Using the Database Approach (continued)

- 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 (continued)

- 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.

Historical Development of Database Technology

- Early Database Applications:
  - The Hierarchical and Network Models were introduced in mid 1960s and dominated during the seventies.
  - A bulk of the worldwide database processing still occurs using these models, particularly, the hierarchical model using IBM's IMS system.
- Relational Model based Systems:
  - Relational model was originally introduced in 1970, was heavily researched and experimented within IBM Research and several universities.
  - Relational DBMS Products emerged in the early 1980s.

Historical Development of Database Technology (continued)

- Object-oriented and emerging applications:
  - Object-Oriented Database Management Systems (OODBMSs) were introduced in late 1980s and early 1990s to cater to the need of complex data processing in CAD and other applications.
    - Their use has not taken off much.
  - Many relational DBMSs have incorporated object database concepts, leading to a new category called *object-relationa*l DBMSs (ORDBMSs)
  - *Extended relational* systems add further capabilities (e.g. for multimedia data, text, XML, and other data types)

Historical Development of Database Technology (continued)

- Data on the Web and E-commerce Applications:
  - Web contains data in HTML (Hypertext markup language) with links among pages.
  - This has given rise to a new set of applications and E-commerce is using new standards like XML (eXtended Markup Language). (see Ch. 13).
  - Script programming languages such as PHP and JavaScript allow generation of dynamic Web pages that are partially generated from a database (see Ch. 11).
    - Also allow database updates through Web pages

Extending Database Capabilities (1)

- New functionality is being added to DBMSs in the following areas:
  - Scientific Applications – Physics, Chemistry, Biology - Genetics
  - Earth and Atmospheric Sciences and Astronomy
  - XML (eXtensible Markup Language)
  - Image Storage and Management
  - Audio and Video Data Management
  - Data Warehousing and Data Mining – a very major area for future development using new technologies (see Chapters 28-29)
  - Spatial Data Management and Location Based Services
  - Time Series and Historical Data Management
- The above gives rise to *new research and development* in incorporating new data types, complex data structures, new operations and storage and indexing schemes in database systems.

Extending Database Capabilities (2)

- Background since the advent of the 21^{st} Century:
  - First decade of the 21^{st} century has seen tremendous growth in user generated data and automatically collected data from applications and search engines.
  - Social Media platforms such as Facebook and Twitter are generating millions of transactions a day and businesses are interested to tap into this data to "understand" the users
  - Cloud Storage and Backup is making unlimited amount of storage available to users and applications

Extending Database Capabilities (3)

- 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).

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 stringent real-time requirements that may not be met because of DBMS overhead (e.g., telephone switching systems)
  - 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).