CS 5200 Database Management Systems

Fall 2022

Introductory Material
Lecture 0
September 9, 2022

Prof. Scott Valcourt

s.valcourt@northeastern.edu

603-380-2860 (cell)

Portland, ME

The Roux Institute
at Northeastern University

Scott Valcourt

- Roux Institute, Portland, ME
- s.valcourt@northeastern.edu
- PhD Engineering: Systems Design & Cognate in College Teaching
- MS: Computer Science
- BA: Computer Science with Mathematics Emphasis
- Cyberinfrastructure, Broadband Communications, Systems, Telehealth, CS Education

Scott Valcourt

- Served as the second director of the UNH InterOperability Lab (IOL)
  - I killed most of the dead networking technologies
- Served as the PI on NH's NTIA BTOP ARRA-funded network construction project that created NetworkNH
- Co-Founder of the UNH Telehealth Practice Center

- I have been a Scout for nearly 44 years!
- I am a native Mainer!

Getting to Know You

- Circulate through the room and offer:
  - Your name
  - Your hometown
  - Your undergraduate degree and from which institution
  - What attracts you to the Roux Institute
  - Something you would enjoy doing if you were not here right now

Class Norms

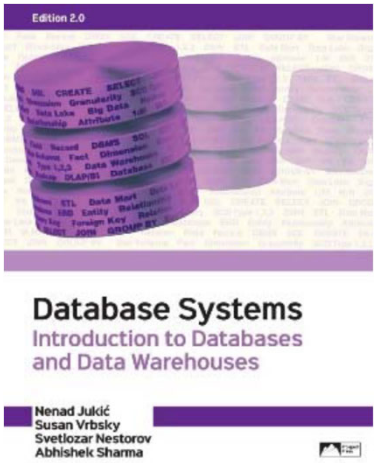
- Engagement
  - Can we agree to leave our cameras on?
  - Can we agree to leave our microphones off unless speaking?
  - Can we agree to be sure that our screen name is our own?
- Interaction
  - Chat a question anytime
  - Raise Hand to ask a question and open microphone when called
  - Attempt to follow a class discussion back-off algorithm
- Process
  - Friday Class Cadence
- Class Recording
  - Can we agree to allow for the recording of the class, removing video, yet maintaining audio questions and discussion?

Section Overview

- Syllabus Review
- Calendar Review
- Housekeeping Mechanics
- General Overview of the Course
- The Web
- Web Applications
- REST

Syllabus and Calendar Review

- Introduction to Databases and Data Warehouses
  - Nenad Jukić, Susan Vrbsky, Svetlozar Nestorov, Abhishek Sharma, 2021
  - Edition 2.0
  - Prospect Press, ISBN 978-1-943153-68-8
  - https://www.prospectpressvt.com/textbooks/jukicdatabase-systems-introduction-to-databases-and-data-warehouses-2-0



- See the syllabus PDF document on Canvas
- As well as the course calendar PDF document on Canvas

Syllabus and Calendar Review

- Hands-on, examples and problem solving in the class
- Bring a laptop/remote connectivity device
- We will use Open Source software that operates on every Operating System
- If there is something that you hear about that sounds interesting, bring it up and we'll discuss

Reaching Out

- Office Hours
  - Prof. Valcourt:
    - Mondays 10:00am-11:00am ET
    - By Appointment

- Email
  - Prof. Valcourt: s.valcourt@northeastern.edu

- Cell Phone and Text
  - (603) 380-2860

Grades

- Homework Assignments 50%
- Quizzes 10%
- Project (Parts A and B) (10% each) 20%
- Final Exam 20%

- Lateness Policy and Graduate Studies

CS5200 Course Topics

At the end of CS 5200, a student should be able to do the following:

1. Explain the principles and characteristics of good relational database design.
2. Design entity relationship models for a business problem domain verified by the rules of normalization (through third normalized form).
3. Build simple to moderately complex data models that solve business problems.
4. Compare and contrast the data definition language (DDL), data manipulation language (DML) and data control language (DCL) components of Structured Query Language (SQL).
5. Write simple to moderately complex SQL queries in a multiple table environment.
6. Write SQL to create tables and indexes, insert rows, delete rows, update rows, drop tables and indexes and alter the database.
7. Understand the concept of database transactions and demonstrate the proper use of commits and rollbacks.
8. Create and use SQL scripts.
9. Have an awareness of contemporary issues in database design and development.
10. Install, configure, manage and maintain a database system.

Resources

- MariaDB Server 10.6.9 (latest long-term release)
  - https://mariadb.org/download/
  - https://mariadb.com/kb/en/mariadb-server-10-6-9/
  - https://mariadb.com/kb/en/installing-mariadb-on-macos-using-homebrew/
  - https://mariadb.com/resources/blog/installing-mariadb-10-1-16-on-mac-os-x-with-homebrew/
- Khoury GitHub
  - https://northeastern.service-now.com/kb_view.do?sysparm_article=KB0012152
- Putty for Windows
  - https://www.chiark.greenend.org.uk/~sgtatham/putty/

Final Planning Thoughts

- Any other norms or procedures we want to identify and agree to in our learning environment?

CS 5200 Database Management Systems

Fall 2022

Database Management Systems: An Architectural View
Lecture 1
September 9, 2022

Prof. Scott Valcourt

s.valcourt@northeastern.edu

603-380-2860 (cell)

Portland, ME

The Roux Institute
at Northeastern University

Initial Terminology

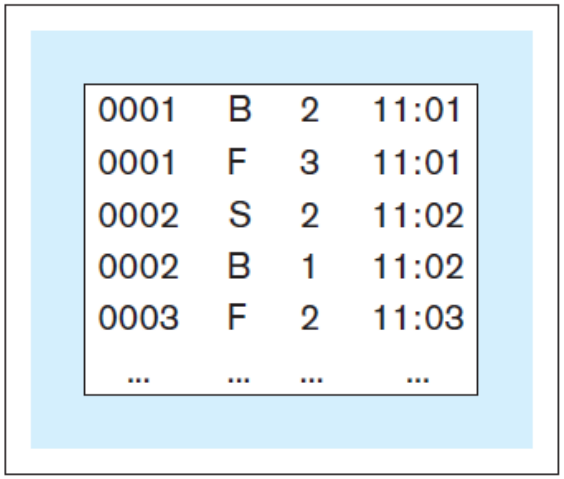
- **Data** - facts that are recorded and can be accessed
  - Data formats – text, numbers, figures, graphics, images, audio/video recordings and more
  - Data is recorded and kept because it is considered to be of use to an intended user
- **Information** - refers to the data that is accessed by a user for some particular purpose
  - Typically, getting the needed information from a collection of data requires performing an activity, such as searching through, processing, or manipulating the data in some form or fashion

Initial Terminology

- **Metadata** - data that describes the structure and the properties of the data
  - Metadata is essential for the proper understanding and use of the data

Initial Terminology

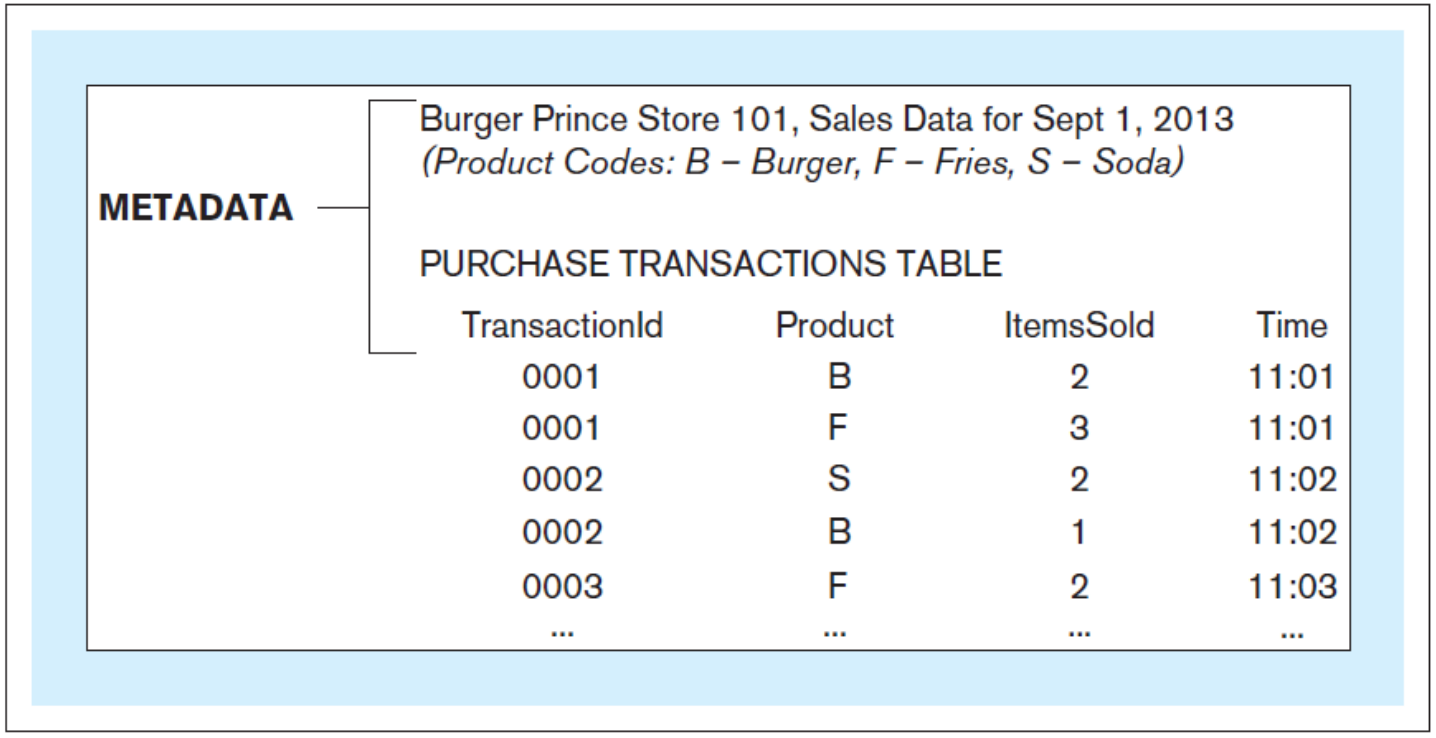
Data without metadata - example



0001	B	2	11:01
0001	F	3	11:01
0002	S	2	11:02
0002	B	1	11:02
0003	F	2	11:03
...	...	...	...

Initial Terminology

Data with metadata - example



Initial Terminology

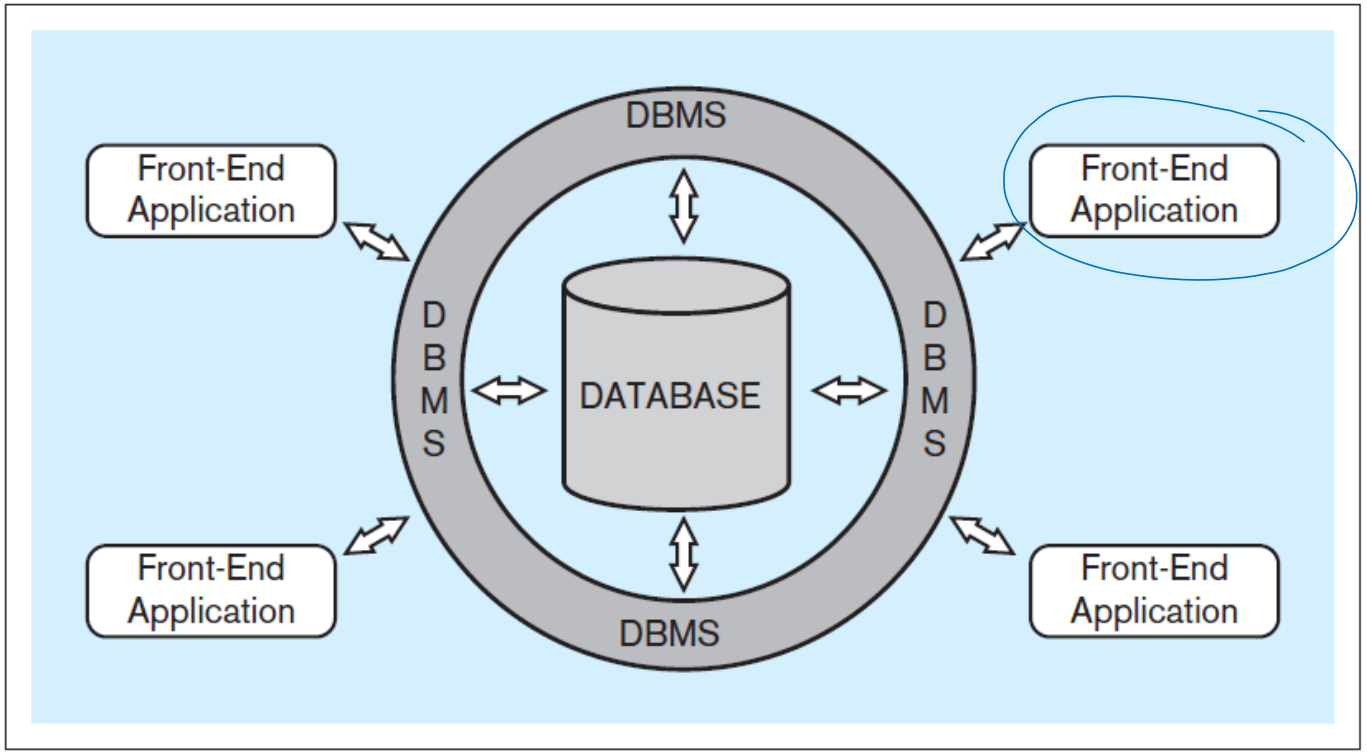
- **Database** - structured collection of related data stored on a computer medium
  - Organizes the data in a way that facilitates efficient access to the information captured in the data
- **Database metadata** – represents the structure of the database
  - Database content that is not the data itself (data about the data)
  - Contains:
    - Names of data structures
    - Data types
    - Data descriptions
    - Other information describing the characteristics of the data

Initial Terminology

- **Database management system (DBMS)** - software used for:
  - Creation of databases
  - Insertion, storage, retrieval, update, and deletion of the data in the database
  - Maintenance of databases
- **Database system** - computer-based system whose purpose is to enable an efficient interaction between the users and the information captured in a database

Initial Terminology

Typical database system architecture

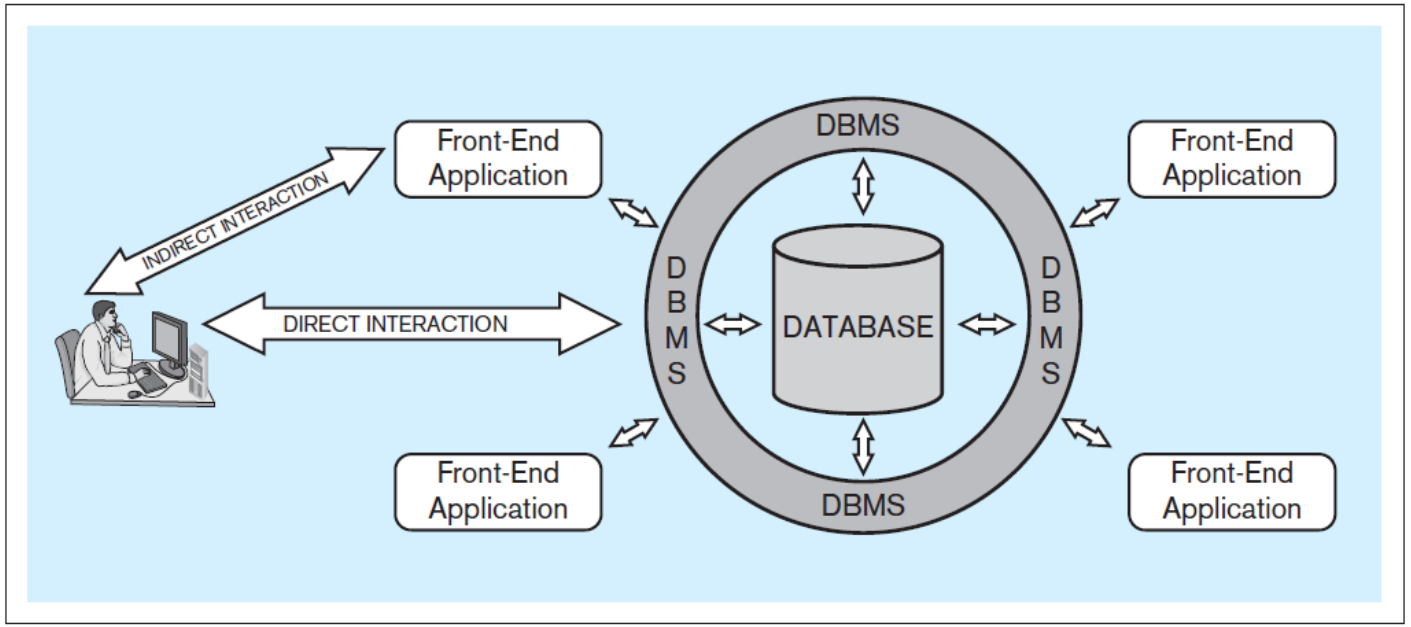


Initial Terminology

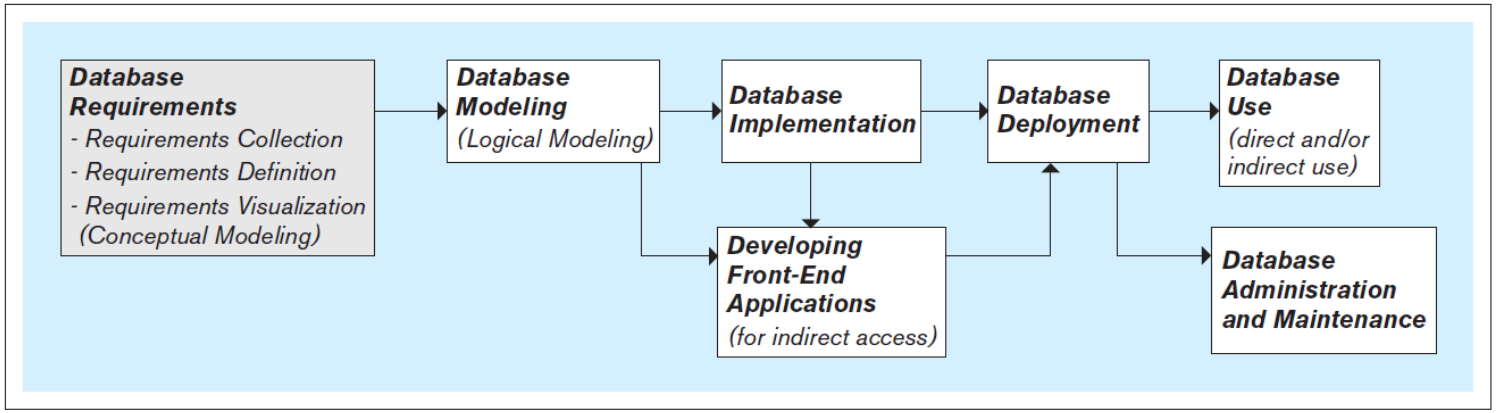
- **Front-end applications** - provide a mechanism for easy interaction between the users and the DBMS
- **End-users** (**business-users**) - users using a database system to support their tasks and processes
- **Indirect interaction** - end-user communicating with the database through front-end applications
- **Direct interaction** - end-user communicating with the database directly through DBMS

Initial Terminology

Typical database system architecture



Database Systems Development Steps



Database Systems Development Steps

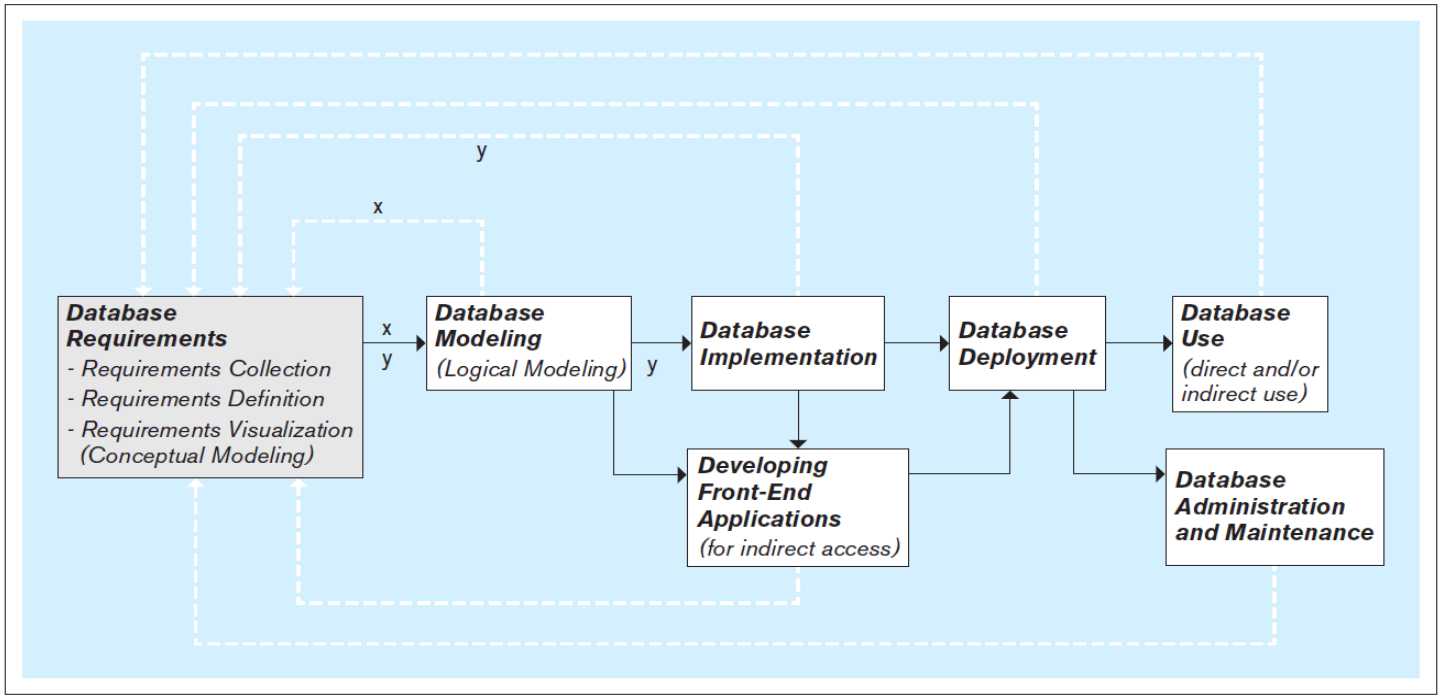
- **Requirements collection, definition, and visualization** - results in the requirements specifying which data the future database system will hold and in what fashion, and what the capabilities and functionalities of the database system will be
  - The **collected** requirements should be clearly **defined** and stated in a written document, and then **visualized**

Database Systems Development Steps

- **Requirements collection, definition, and visualization**
  - **Conceptual database model** – a visualization of requirements by using a conceptual data modeling technique (such as entity-relationship [ER] modeling)

Database Systems Development Steps

Iterative nature of the database requirements collection, definition, and visualization process



Database Systems Development Steps

- **Database modeling** (**logical database modeling**) - creation of the database model that is implementable by the DBMS software
  - *Logical database modeling* follows *conceptual database modeling*

Database Systems Development Steps

- **Database implementation** - using a DBMS to implement the database model as an actual database
  - Most modern databases are implemented using a relational DBMS (RDBMS) software

Database Systems Development Steps

- **Developing front-end applications** - designing and creating applications for indirect use by the end-users
  - Front-end applications are based on the database model and the requirements specifying the front-end functionalities
  - Front-end applications contain interfaces (such as forms and reports) accessible via a navigation mechanism (such as a menu)

Database Systems Development Steps

- **Database deployment -** releasing the database system for use by the end users

Database Systems Development Steps

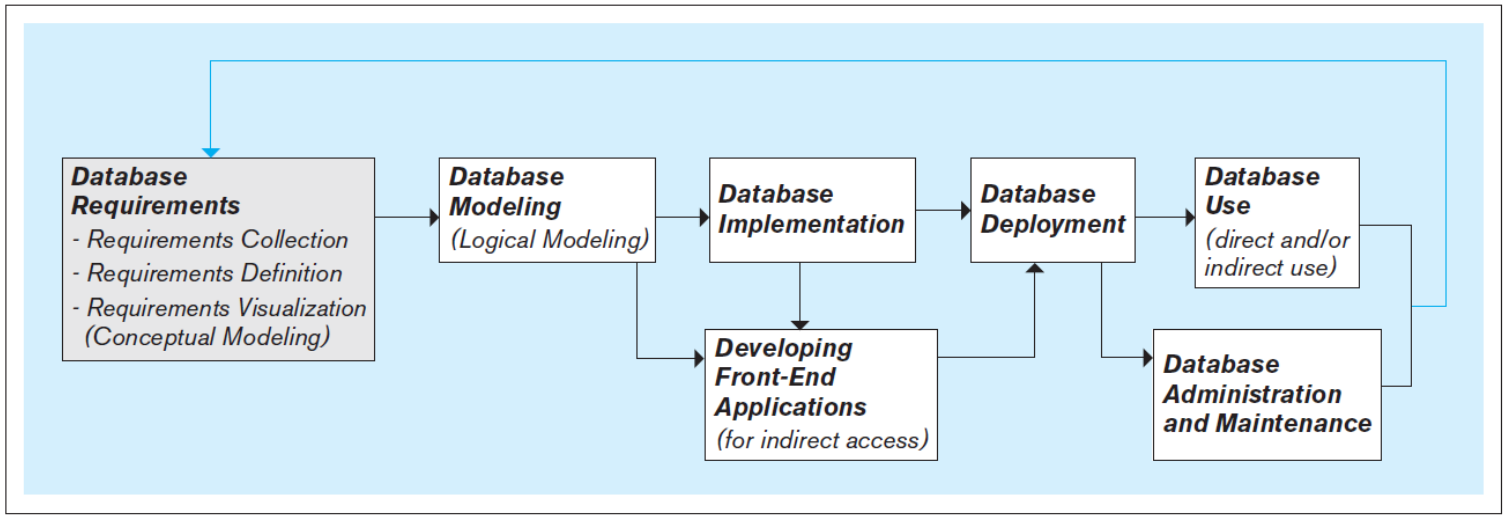
- **Database use -** the insertion, modification, deletion and retrieval of the data in the database system

Database Systems Development Steps

- **Database administration and maintenance -** performing activities that support the database end user, including dealing with technical issues, such as:
  - Providing security for the information contained in the database
  - Ensuring sufficient hard-drive space for the database content
  - Implementing the backup and recovery procedures

The Next Version of the Database

- Follows the same development steps as the initial version



Database Scope

- Databases can vary in their scope from small single-user (personal) databases to large enterprise databases that can be used by thousands of end-users
- Regardless of their scope, all databases go through the same fundamental development steps (*requirements, modeling, implementation, deployment, use,* etc.)

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

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

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:**
  - 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 (DBA):**
    - 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
  - Providing Storage Structures (e.g. indexes) for efficient Query Processing

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.

Operational versus Analytical Databases

- **Operational information (transactional information) -** the information collected and used in support of day-to-day operational needs in businesses and other organizations
- **Operational database** - collects and presents operational information in support of daily operational procedures and processes
- **Analytical information** - the information collected and used in support of analytical tasks
  - Analytical information is based on operational (transactional) information
- **Analytical database** - collects and presents analytical information in support of analytical tasks

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).
  - Script programming languages such as PHP and JavaScript allow generation of dynamic Web pages that are partially generated from a database
    - Also allow database updates through Web pages

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
- 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 (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 21st Century:
  - First decade of the 21st 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. 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.

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).

Book Topics Overview

- Issues related to the development and use of operational databases are covered in Chapters 2-6
- Issues related to the development and use of analytical databases are covered in Chapters 7-10
- Chapter 11 provides an overview of some of the basic functionalities of relational DBMS packages and illustrate how those functionalities are used for administration and maintenance of both operational and analytical databases
- Appendix materials are special topic related; we will cover most of these topics this semester

Homework Assignment

- Read Chapter 1 from the Jukić text
- Install MariaDB on your local computing device
- Create a user account that has Global Privileges to use MariaDB (don't forget your root login password, though)
- Install Chinook database into MariaDB
- Complete Homework Assignment 1 (PDF on Canvas)

What's Next

- Software Installation
  - MariaDB
  - Phpadmin/HeidiSQL
  - Chinook Database
- Quiz 1
- Homework Assignment 1
- Read Chapter 1 from the textbook