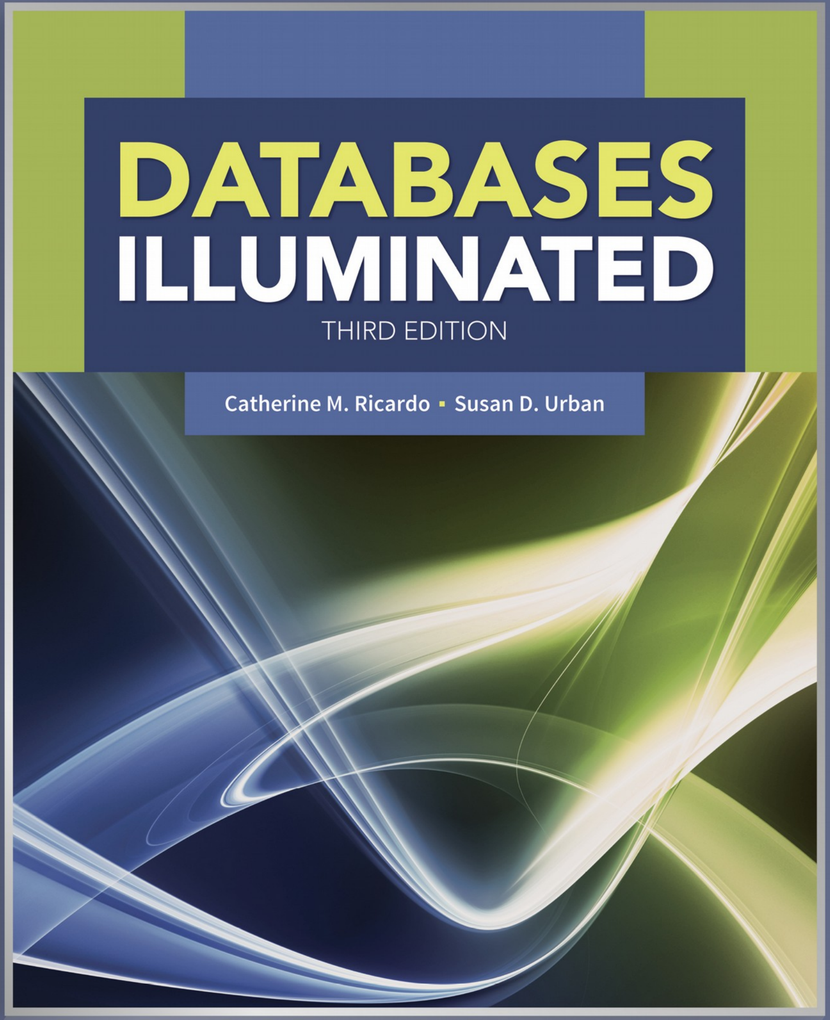
Databases Illuminated

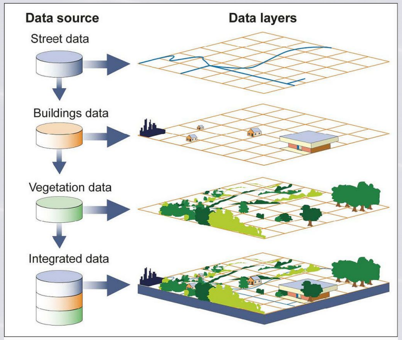
Chapter One

Introductory Database Concepts



Uses of Databases

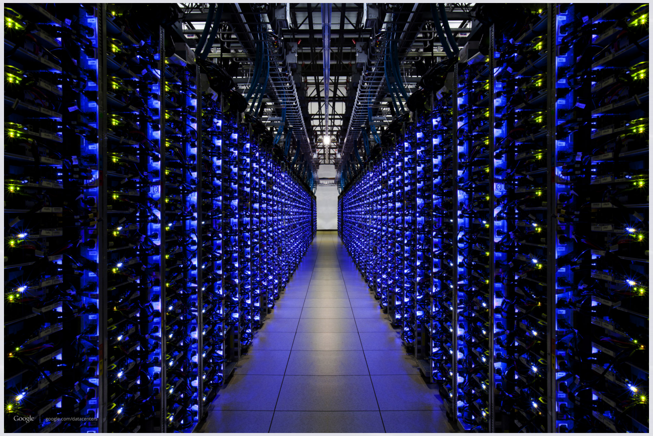
- Databases used in large and small organizations. Examples:
  - Consumer websites
  - Search engines
  - Travel reservations
  - Online banking
  - Health care
  - Libraries
  - Advanced applications
    - Geographic Information Systems (GIS) (Maptitude) (GIS-Youtube)
    - Software development
    - Scientific research (Google Scholar) (Online DBs)
    - Decision Support Systems (DSS) (Clinical DSS)
    - Customer Relations Management (CRM-Explained) (CRM)



Source: GAO.

Big Data

- Capturing, organizing, and analyzing massive amounts of data
  - Data warehouses
  - Data mining (Intro-DM)
  - Internet sources
  - Ubiquitous devices
    - Ubiq-Comp
    - MagicMirror



Google Data Center (GOOGLE)

(Facebook)

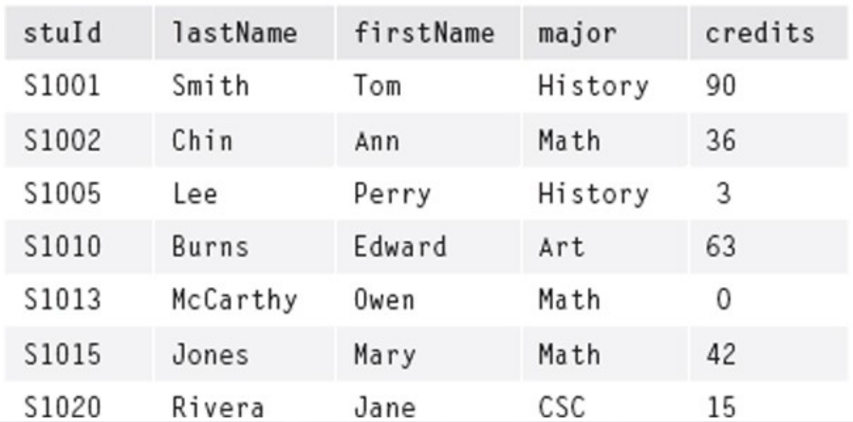
Apple Data Center

Basic Definitions

- **Database:**
  - A collection of related data.
  - Can be of any size and complexity (Addr. book vs library vs FB)
- **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.

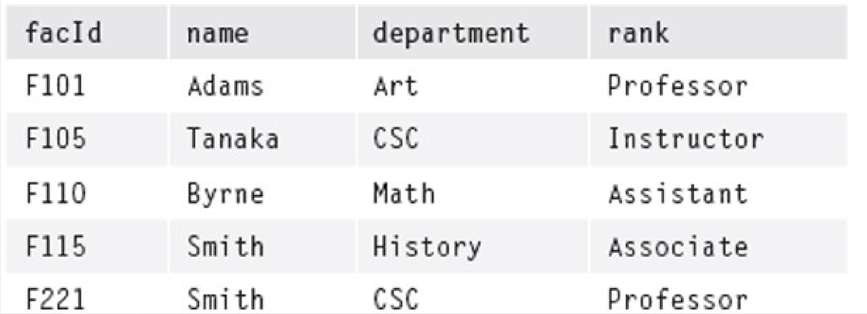
A Sample Database

- Simple `University` database
- Keeps information about
  - Students
  - Faculty
  - Classes-links faculty to their classes
  - Enrollment-links students to their classes
- Example uses LibreOffice Base (LibreOffice) (Tutorial)
- Data represented as **tables**
- Each row of `Student` table represents one student, of `Faculty` one faculty member, of `Class` one class
- Each row of `Enroll` represents **relationship** between one student and one class



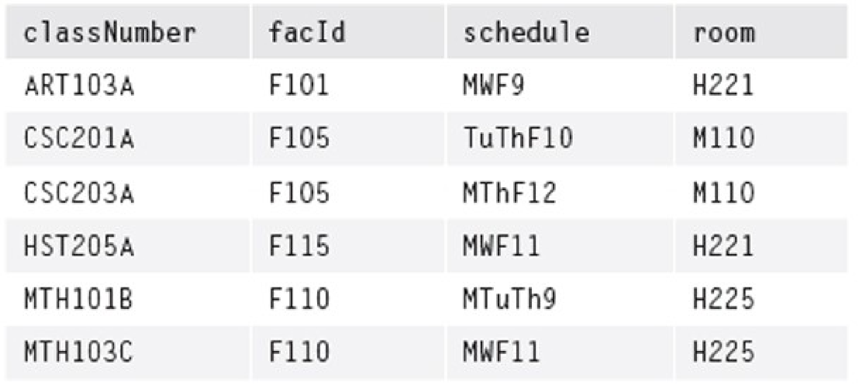
stuId	lastName	firstName	major	credits
S1001	Smith	Tom	History	90
S1002	Chin	Ann	Math	36
S1005	Lee	Perry	History	3
S1010	Burns	Edward	Art	63
S1013	McCarthy	Owen	Math	0
S1015	Jones	Mary	Math	42
S1020	Rivera	Jane	CSC	15

Figure 1.1A The `Student` Table



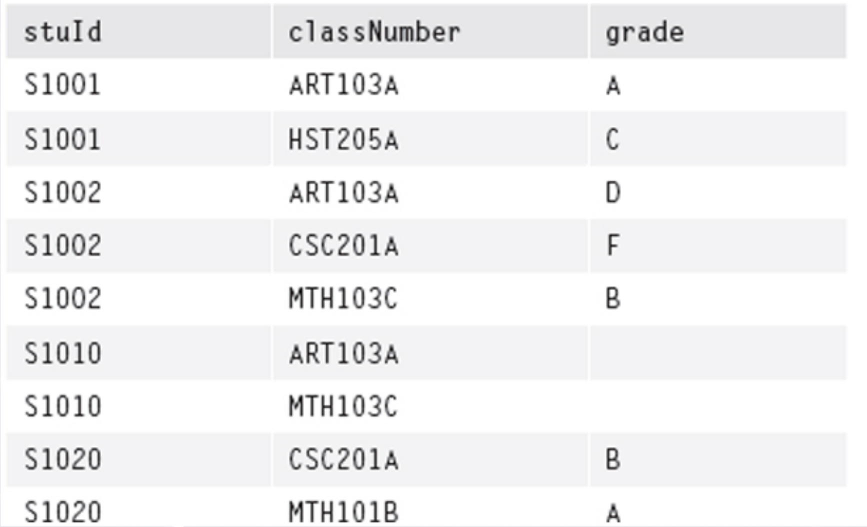
facId	name	department	rank
F101	Adams	Art	Professor
F105	Tanaka	CSC	Instructor
F110	Byrne	Math	Assistant
F115	Smith	History	Associate
F221	Smith	CSC	Professor

Figure 1.1B The `Faculty` Table



classNumber	facId	schedule	room
ART103A	F101	MWF9	H221
CSC201A	F105	TuThF10	M110
CSC203A	F105	MThF12	M110
HST205A	F115	MWF11	H221
MTH101B	F110	MTuTh9	H225
MTH103C	F110	MWF11	H225

Figure 1.1C The `Class` Table



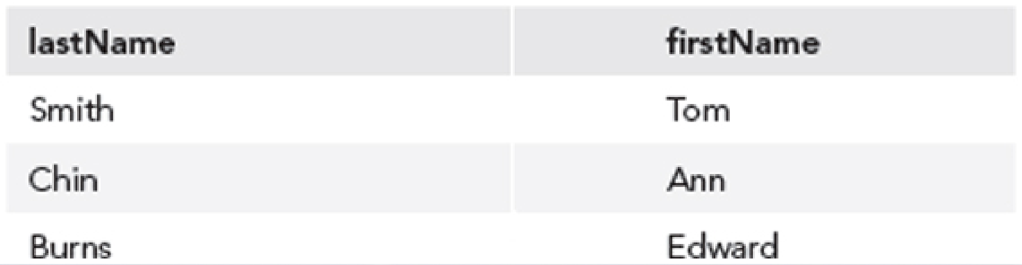
stuId	classNumber	grade
S1001	ART103A	A
S1001	HST205A	C
S1002	ART103A	D
S1002	CSC201A	F
S1002	MTH103C	B
S1010	ART103A	
S1010	MTH103C	
S1020	CSC201A	B
S1020	MTH101B	A

Figure 1.1D The `Enroll` Table

Query Tool

- LibreOffice Base has a simple tool for forming and executing queries
- Query: Find the names of all students enrolled in ART103A
- Need to use `Enroll` table and `Student` table, since `Enroll` does not have names
- Query result:

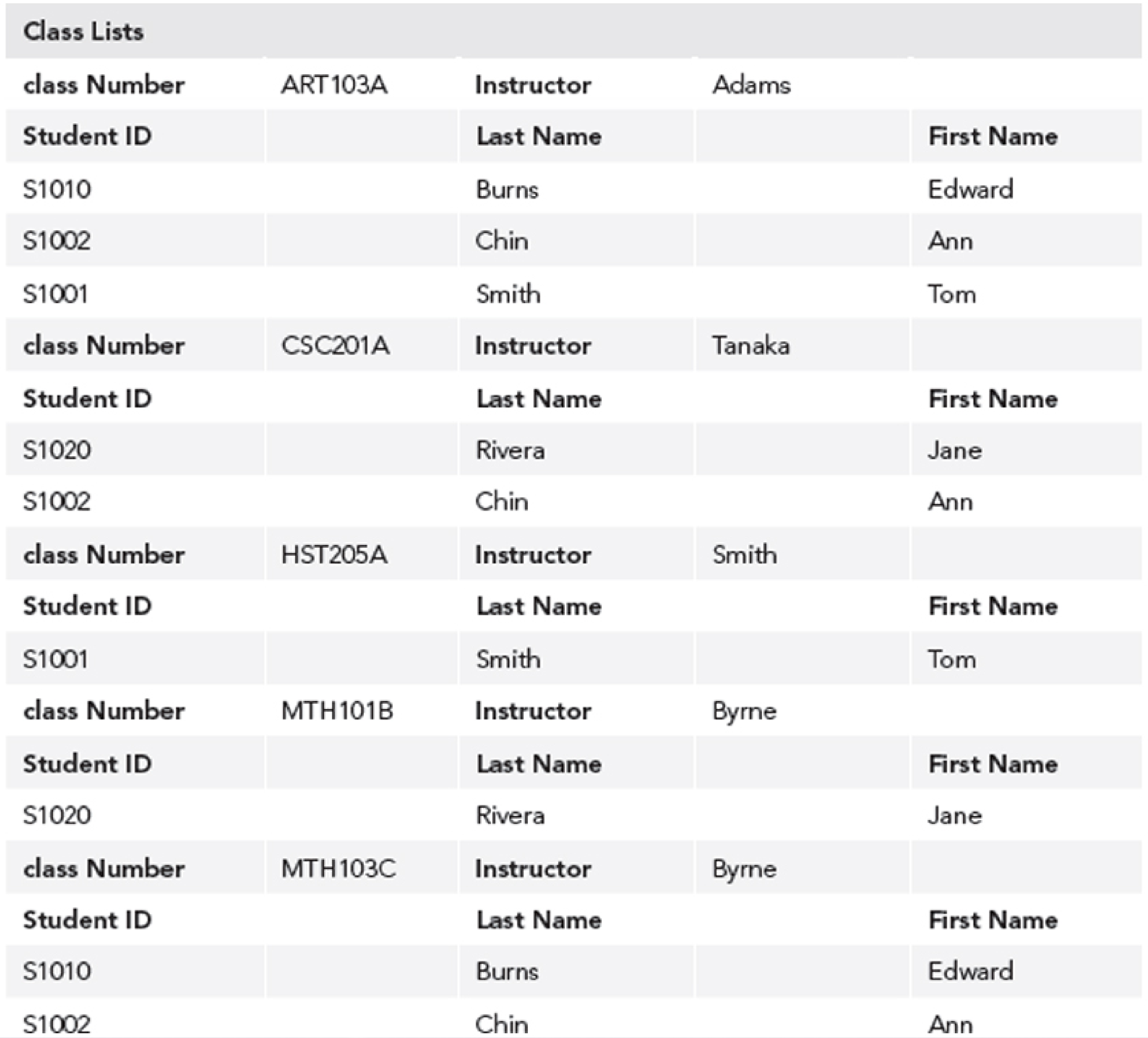
Figure 1.2 Results of Query "Find names of all students enrolled in ART103A"



lastName	firstName
Smith	Tom
Chin	Ann
Burns	Edward

Reporting Tool

- LibreOffice Base has a report generator
- **Example:** Print a report showing each class number, the ID and name of the faculty member teaching the class, and the IDs and names of all the students in that class

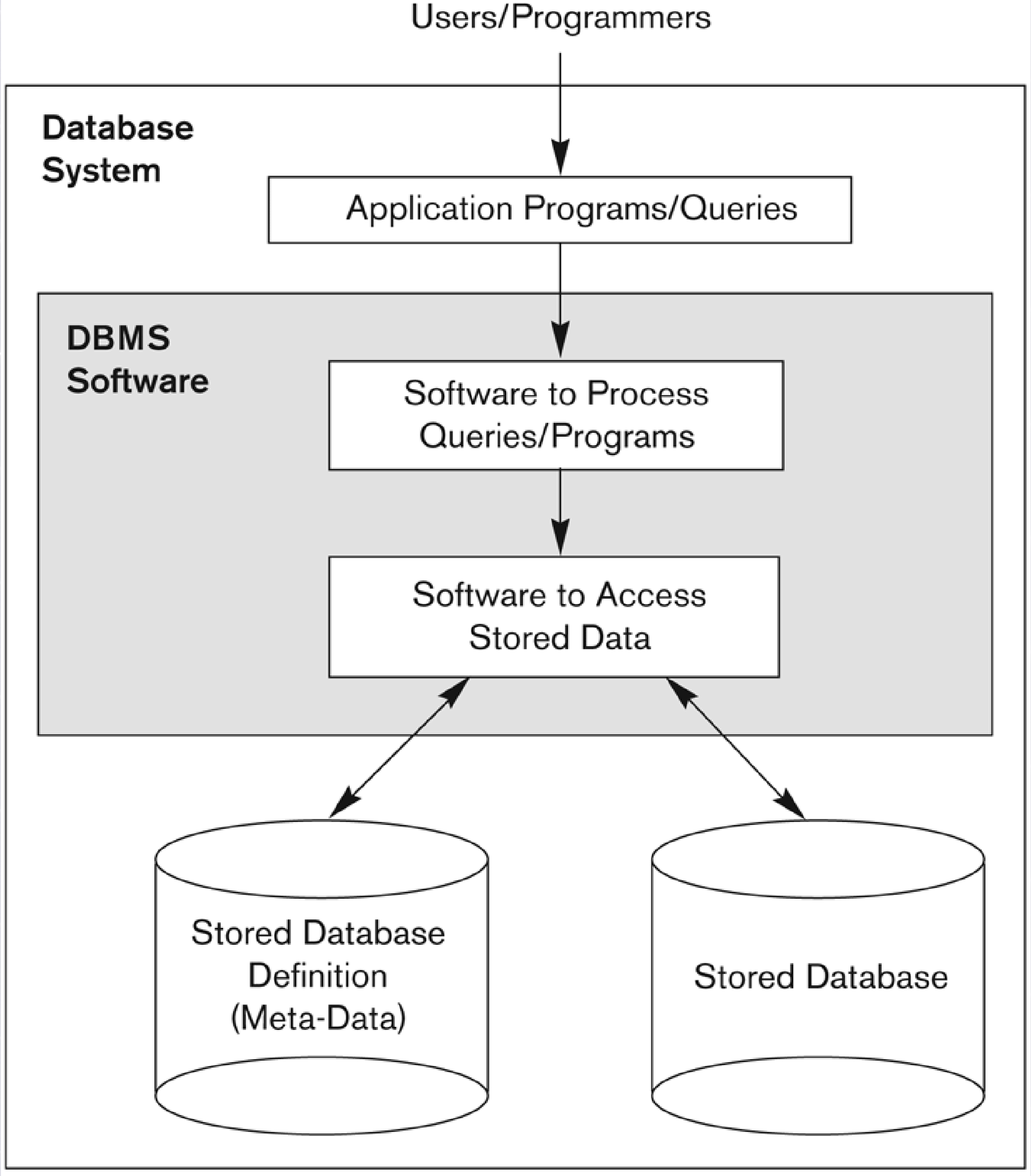


Class Lists				
class Number	ART103A	**Instructor**	Adams	
Student ID		**Last Name**		**First Name**
S1010		Burns		Edward
S1002		Chin		Ann
S1001		Smith		Tom
class Number	CSC201A	**Instructor**	Tanaka	
Student ID		**Last Name**		**First Name**
S1020		Rivera		Jane
S1002		Chin		Ann
class Number	HST205A	**Instructor**	Smith	
Student ID		**Last Name**		**First Name**
S1001		Smith		Tom
class Number	MTH101B	**Instructor**	Byrne	
Student ID		**Last Name**		**First Name**
S1020		Rivera		Jane
class Number	MTH103C	**Instructor**	Byrne	
Student ID		**Last Name**		**First Name**
S1010		Burns		Edward
S1002		Chin		Ann

Figure 1.3 `Class Lists` Report

The Integrated Database Environment

- **Database**
  - Large repository of data
  - Shared resource, used by many departments and applications of an enterprise
  - Contains several different record types
  - Contains ***metadata*** -"knows" about structure and relationships in data
  - Managed by database administrator - DBA
- **DBMS**, Database Management System (DBMS)
  - Controls access to database
  - Has facilities to
    - Set up database structure
    - Load the data
    - Retrieve requested data and format it for users
    - Hide sensitive data
    - Accept and perform updates
    - Handle concurrency
    - Perform backup and recovery … and many other functions…
- **Users**
- **Applications**


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

The Integrated Database Environment

Example of Integrated Database Environment

- `University` database
- DBMS - may be Access, Oracle, DB2,…(List)
- Users may be individuals on workstations (interactive users) or application programs
- Both users and applications go through DBMS
- Applications produce standard output, such as reports

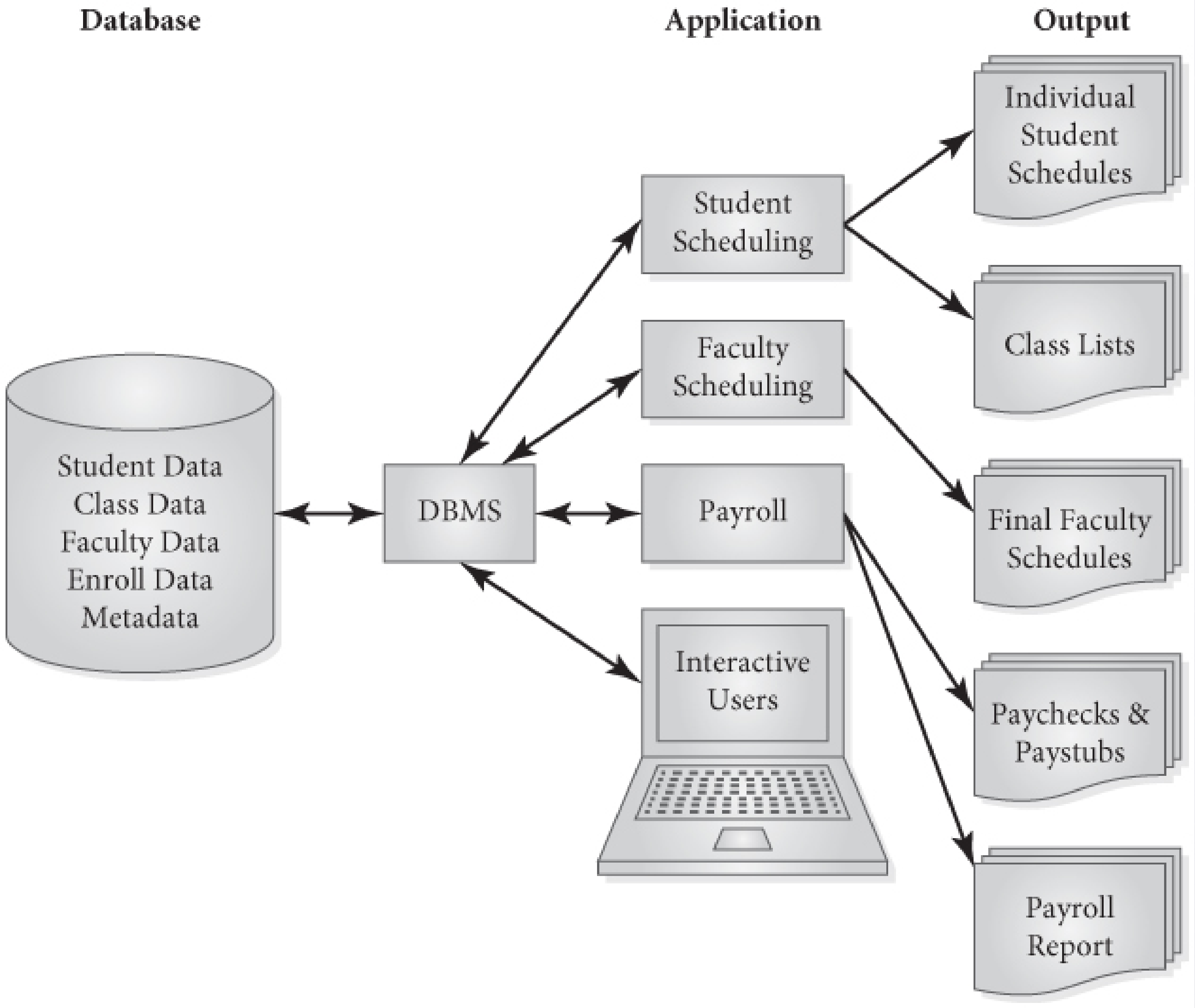


Figure 1.4 The Integrated Database Environment

People in Integrated Database Environment

- End users – see a “view” of data
  - **Casual users** use query language
  - **Naïve users** use programs
  - **Secondary users** use database output
- Applications programmers – write programs for other users
- Database administrator (DBA) – designs, creates, maintains the database

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.

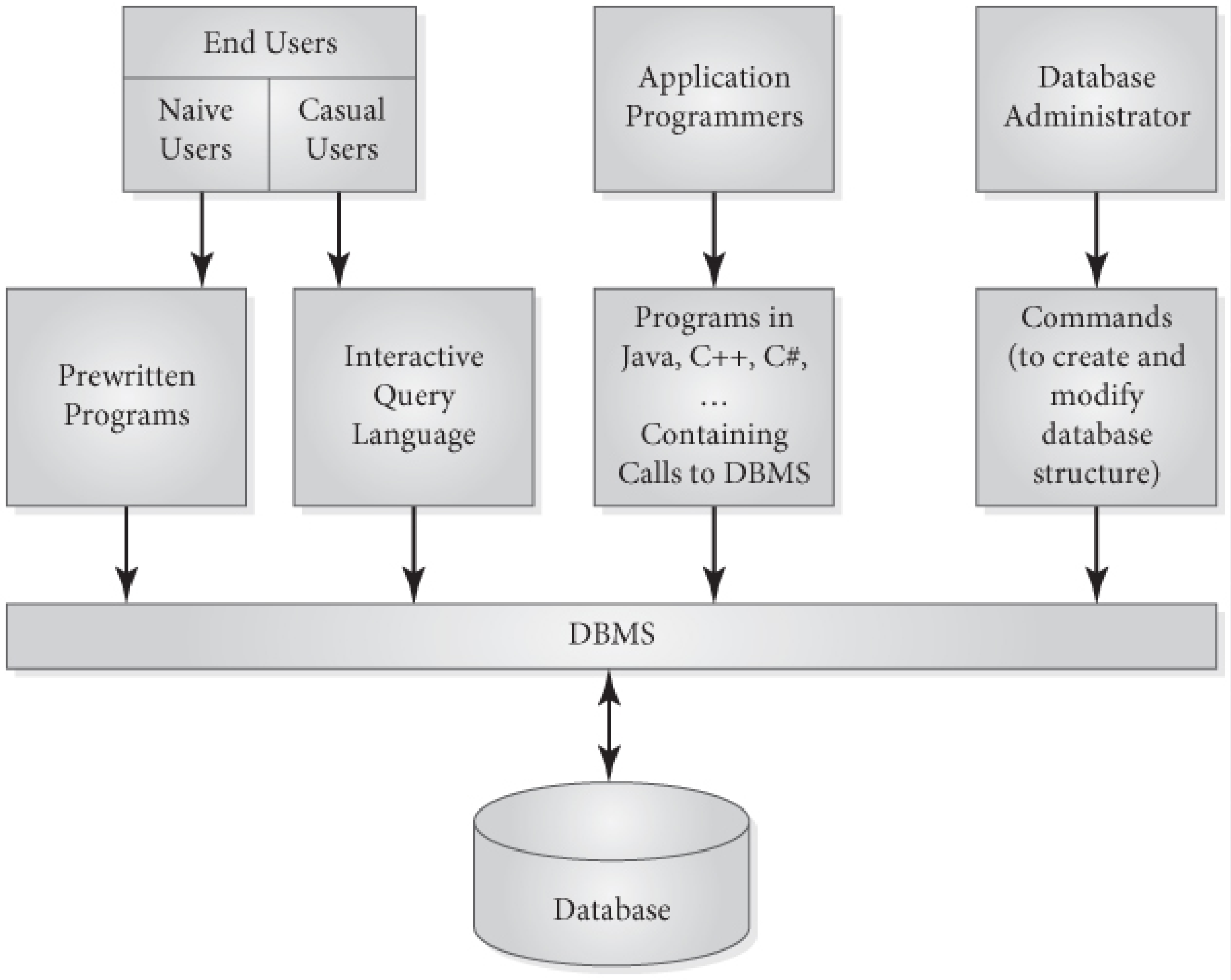

End Users
Naive Users
Casual Users
Application Programmers
Database Administrator
Prewritten Programs
Interactive Query Language
Programs in Java, C++, C#, … Containing Calls to DBMS
Commands (to create and modify database structure)
DBMS
Database

Figure 1.5 Roles in the Database Environment

Advantages of Integrated Databases

- Compared with file systems, which create “islands of information”, database can provide (FlatFile)
  - Sharing of data throughout the enterprise
  - Control of redundancy
  - Data consistency
  - Improved data standards
  - Better data security, perhaps using encryption
  - Improved data integrity
  - Balancing of conflicting requirements
  - Faster development of new applications
  - Better data accessibility
  - Economy of scale
  - More control of concurrency
  - Better backup and recovery procedures

Brief History of Information Systems -1 (IS) (Hist.1940-)

- Early human records-clay tablets, hieroglyphics, cave paintings, paper records-family histories, treaties, inventories
- Hollerith used **punched cards** in 1890 US census (Jacquard Loom) (PC)(PC-Prog.)
- **Punched paper tape** introduced in 1940s
- **Magnetic tape** introduced about 1950-used in UNIVAC I
- Cards, paper tape, magnetic tape are **sequential access devices**
- Used in sequential processing applications such as payroll
- **(History)**

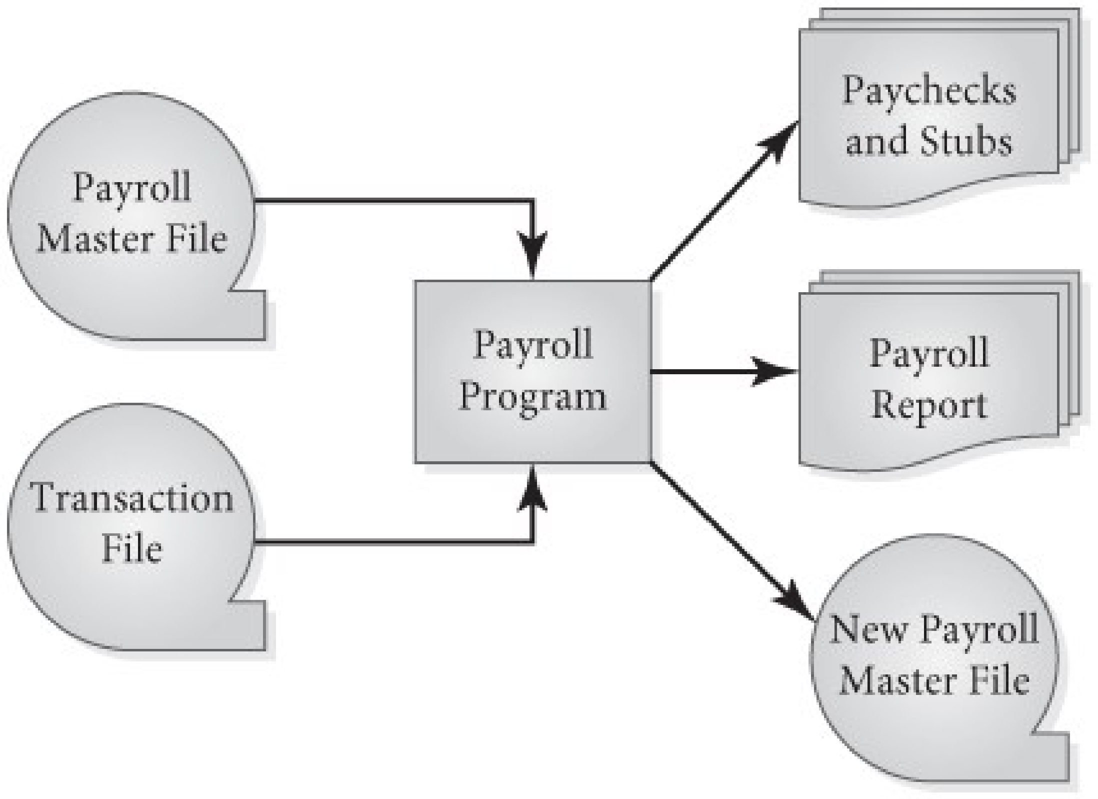


Figure 1.6 A Sequential File Processing System

- **Batch processing** uses master file and transaction file as input; produces new master file as output

Brief History of Information Systems - 2

- Magnetic disk introduced in 1950s - **direct access device**
- Programming languages COBOL (COBOL) PL/1 (PL/1) in 1960s.
- Early database models developed (Data Models: Definition) (Data Model:Types) (Data Models:Evolution) (File Systems)
  - Hierarchical model (Hiararchical DB)
    - IBM **IMS** developed for Apollo moon landing project
    - IMS product released in 1968
    - Most popular pre-relational DBMS; still in use
    - SABRE airline reservation system used IMS
  - Network model (Network DB)
    - GE **IDS** developed by Charles Bachman in early 1960s
    - CODASYL DBTG proposed standards published in 1971
    - ANSI rejected proposal
    - New standards published in 1973, 1978, 1981 and 1984
    - Provided standard terminology, notion of layered database architecture

Brief History of Information Systems-3

- **Relational model** (Relational DB) (History)
  - Proposed by E.F. Codd in 1970 paper, *A Relational Model of Data for Large Shared Data Banks*
  - Strong theoretical foundation
  - **System R**, late 1970s
    - IBM's prototype relational system
    - Introduced SQL, now standard language
  - Peterlee Relational Test Vehicle, IBM UK Scientific Laboratory
  - **INGRES** at UC, Berkeley-Wong & Stonebraker
    - Postgres and PostgresSQL
  - **ORACLE** used some System R results; Ellison, Miner, & Oakes
  - Early microcomputer-based relational DBMSs :dBase, R:Base, Foxpro, Paradox
  - MS Access, Oracle, DB2, MySQL, PostGreSQL, MS SQL Server most popular DBMSs

Brief History of Information Systems-4

- **Entity Relationship** model – Peter Chen, 1976 (ER: Image)
  - Semantic model – tries to capture meaning, used mostly for design
- **Object-oriented model** - Introduced in 1990s (Object DB)
  - Can handle complex data
  - **UML** by Booch, Jacobson, & Rumbaugh-modeling language for software engineering-used for modeling OO databases (Wiki) (UML)
  - Object-oriented programming languages extended
  - Caché, ObjectStore, Gemstone, Versant, Objectivity
- **Object-relational** model (Object-Relational DB)
  - Object-oriented capabilities in relational databases

Brief History of Information Systems-5

- **Data warehouses** - developed in 1990s
  - *Building the Data Warehouse* by **Bill Inmon;** *The Data Warehouse Toolkit* By Ralph Kimball
  - Take data from many sources
  - May store historical data
  - Used for **data mining**, finding trends in data (Wiki)
- Internet provides access to vast network of databases
  - E-commerce
  - XML standard for data exchange (XML)
  - Semi-structured data model

Big Data

- The 5 Vs of big data
  - **Volume**
    - huge amount of data from wide array of sources
  - **Velocity**
    - data is generated at great speed
    - speed needed for organizing, storing, processing
  - **Variety**
    - source data has many different formats
  - **Veracity**
    - must ensure correctness of data
  - **Value**
    - competitive advantage

Big Data Technologies

- **Hadoop** - Batch-oriented retrieval of large amounts of data (Hadoop) (Hadoop) (Intro.)
  - Hadoop Distributed File System (HDFS)
    - fault tolerance by dividing large files into blocks
    - blocks replicated in clusters of nodes and racks
  - MapReduce parallel programming model
    - map step: filters/ transforms input data
    - reduce step: performs calculations or aggregation
- **NoSQL**- For real-time queries and row-level inserts, updates, deletes (NoSQL)
  - horizontal scaling with replication and distribution over servers
  - flexible schema, weaker concurrency model, simple interface, parallel processing
  - types: key-value pairs, column-oriented, documented-oriented, graph-oriented
- **NewSQL (NewSQL)**
  - stronger data consistency model
  - new distributed architectures
  - cloud computing