CS 309A- Database Management Systems

What is a Database System?

- Database: A very large collection of *related* data
- Models a real world enterprise:

Why Study Databases

- Wide-range of database applications:
  - Banking: transactions
  - Airlines: reservations, schedules
  - Colleges: registration, grades
  - Sales: customers, products, purchases
  - Online retailers: order tracking, customized recommendations
  - Manufacturing: production, inventory, orders, supply chain
  - Human resources: employee records, salaries, tax deductions

University Database Example

- Application program examples
  - Add new students, instructors, and courses
  - Register students for courses, and generate class rosters
  - Assign grades to students, compute grade point averages (GPA) and generate transcripts
- In the early days, database applications were built directly on top of file systems

Why Databases?

- Why not store everything in flat files:

  i.e., use the file system of the OS, cheap/simple…

  Name, Course, Grade

  John Smith, CS113, B

  Mike Stonebraker, CS205, A

  Jim Gray, CS405, A

  John Smith, CS315, B+

  ……

This is how things were in the "Bad Old Days"

Problem 1

- Data redundancy and inconsistency
  - Multiple file formats,
  - duplication of information in different files

Name, Course, Email, Grade	***Name, Email, Course, Grade***
John Smith, CS113, js@etown.edu, B	Mike Stonebraker, ms@etown.edu, CS234, A
Jim Gray, CS560, jg@etown.edu, A	J. Smith, js@etown.edu, CS560, B+
John Smith, CS560, js@etown.edu, B+	

Why is this a problem?

- Wasted space
- Potential inconsistencies

  (e.g., multiple formats, John Smith vs J. Smith)

Problem 2

- Data retrieval:
  - Find the students who took CS113
  - Find the students with GPA > 3.5

  For every query we need to write a program!

- We need the retrieval to be:
  - Easy to write
  - Execute efficiently

Problem 3

- Data Integrity
  - No support for sharing:
    - Prevent simultaneous modifications
  - No coping mechanisms for system crashes
  - No means of Preventing Data Entry Errors (checks must be hard-coded in the programs)
  - Security problems: hard to provide user access to some, but not all, data

- Database systems offer solutions to all the above problems

Problem 4

- Long-lived data → Evolution
- What happens if I need to change my mind about how the data is stored?
  - Access patterns change
- Don’t want to have to re-write all my applications.

- Solution: Data independence!

Database

- A **database** is a shared, integrated computer *structure*.
- The data stored in a database includes:
  - End-user data: raw facts of interest to the end user
  - Metadata: data about data

- The **metadata**
  - describe the data characteristics and relationships in data.
  - present a more complete picture of the data in the database.

End-user data vs. Metadata

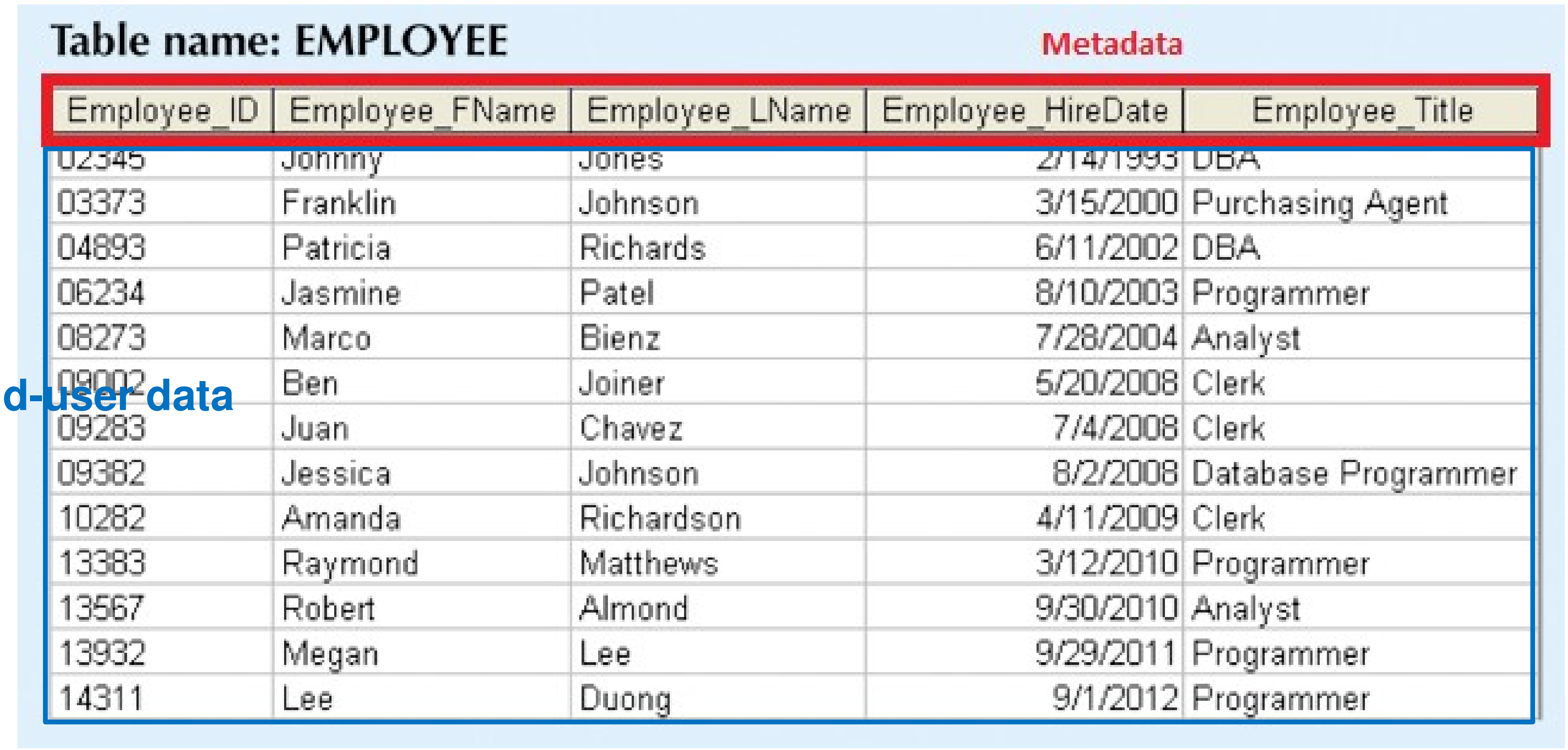

Metadata
End-user data

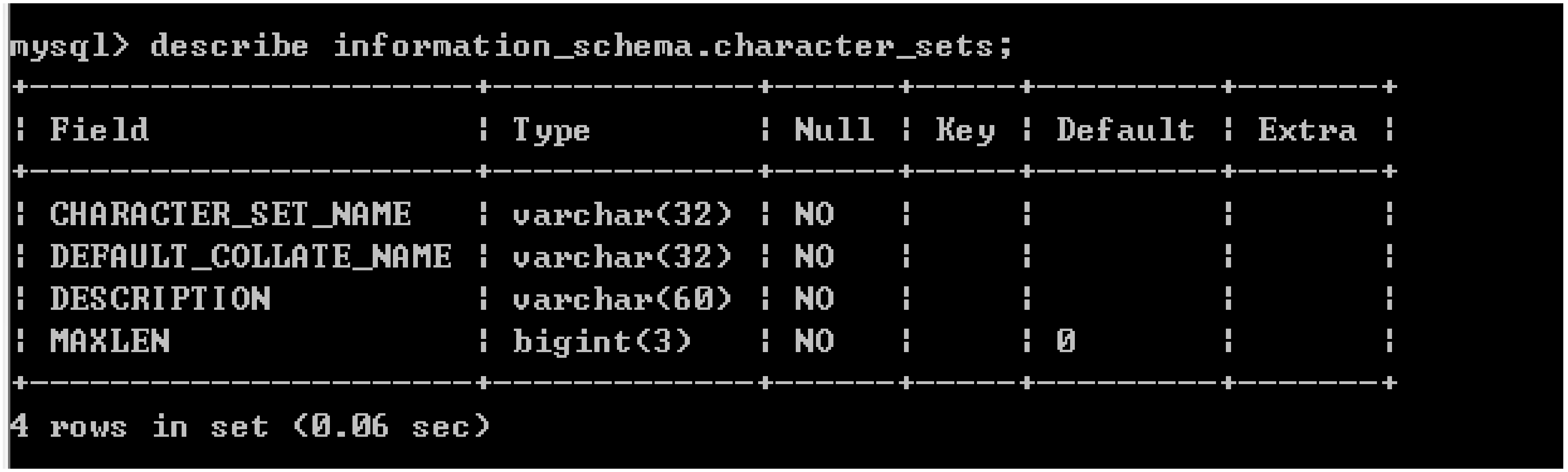
Table name: EMPLOYEE

Employee_ID	Employee_FName	Employee_LName	Employee_HireDate	Employee_Title
02345	Johnny	Jones	2/14/1993	DBA
03373	Franklin	Johnson	3/15/2000	Purchasing Agent
04893	Patricia	Richards	6/11/2002	DBA
06234	Jasmine	Patel	8/10/2003	Programmer
08273	Marco	Bienz	7/28/2004	Analyst
09002	Ben	Joiner	5/20/2008	Clerk
09283	Juan	Chavez	7/4/2008	Clerk
09382	Jessica	Johnson	8/2/2008	Database Programmer
10282	Amanda	Richardson	4/11/2009	Clerk
13383	Raymond	Matthews	3/12/2010	Programmer
13567	Robert	Almond	9/30/2010	Analyst
13932	Megan	Lee	9/29/2011	Programmer
14311	Lee	Duong	9/1/2012	Programmer

Metadata

- Data characteristics
  - name of data element
  - Data types (numeric, dates, or text)
  - Empty or not
- Relationships
  - important component of database design
  - often defined by their environment, e.g., EMPLOYEE and JOB

An example of metadata



```
mysql> describe information_schema.character_sets;
+----------------------+-------------+------+-----+---------+-------+
| Field                | Type        | Null | Key | Default | Extra |
+----------------------+-------------+------+-----+---------+-------+
| CHARACTER_SET_NAME   | varchar(32) | NO   |     |         |       |
| DEFAULT_COLLATE_NAME | varchar(32) | NO   |     |         |       |
| DESCRIPTION          | varchar(60) | NO   |     |         |       |
| MAXLEN               | bigint(3)   | NO   |     | 0       |       |
+----------------------+-------------+------+-----+---------+-------+
4 rows in set (0.06 sec)
```

Database Management System (DBMS)

- a collection of *programs*
- Manages the database structure
- Controls access to the data stored in the database

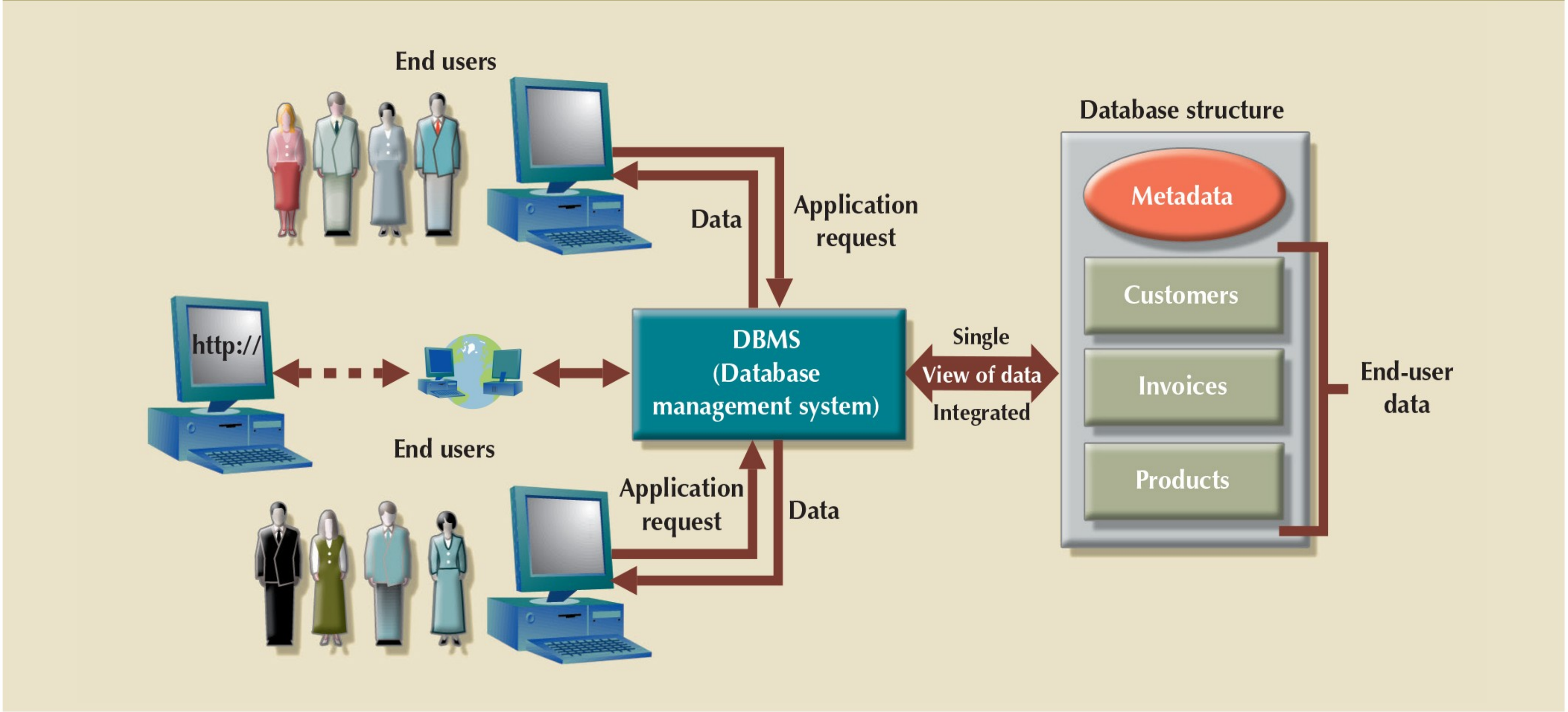
Role of the DBMS

- DBMS is the ***intermediary*** between the user and the database
  - Database structure stored as file collection
  - Can only access files through the DBMS

- DBMS enables data to be *shared*
- DBMS *integrates* many users' views of the data

Role of the DBMS

FIGURE 1.3 THE DBMS MANAGES THE INTERACTION BETWEEN THE END USER AND THE DATABASE



Advantages of the DBMS

- Better data integration and less data inconsistency
  - Data inconsistency: Different versions of the same data appear in different places
- Increased end-user productivity
- Improved:
  - Data sharing
  - Data security
  - Data access
  - Decision making
- Data quality: Accuracy, validity, and timeliness of data

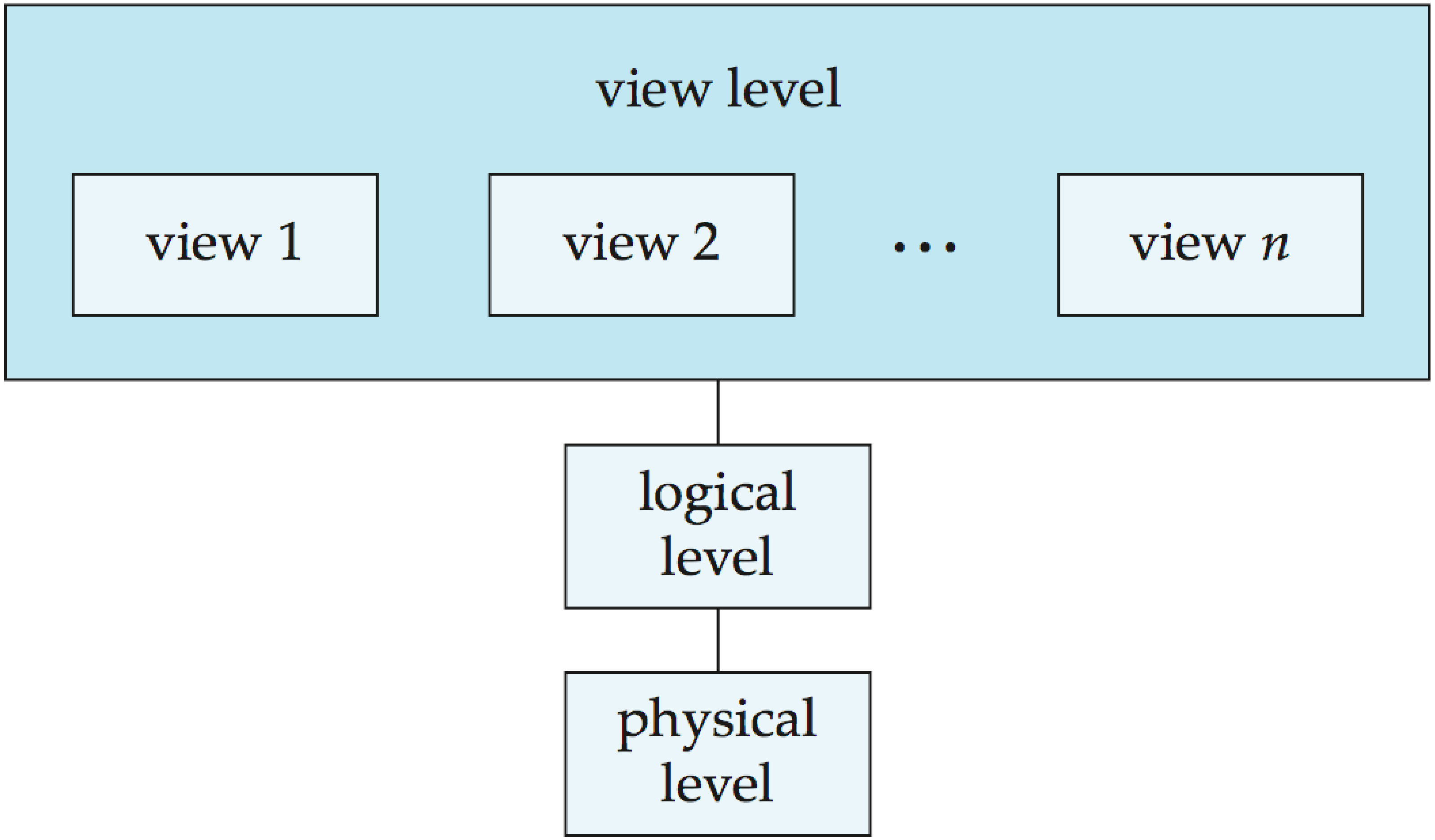
Levels of Abstraction

- **Physical level:** describes how a record (e.g., instructor) is stored.
- **Logical level:** describes data stored in database, and the relationships among the data.

```
type instructor = record
        ID : string;
        name : string;
        dept_name : string;
        salary : integer;
            end;
```

- **View level:** application programs hide details of data types. Views can also hide information (such as an employee's salary) for security purposes.

An architecture for a database system



Instances and Schemas

- Similar to types and variables in programming languages
- Logical Schema – the overall logical structure of the database
  - Example: The database consists of information about a set of customers and accounts in a bank and the relationship between them
    - Analogous to type information of a variable in a program
- Physical schema– the overall physical structure of the database
- Instance – the actual content of the database at a particular point in time
  - Analogous to the value of a variable

Physical Data Independence

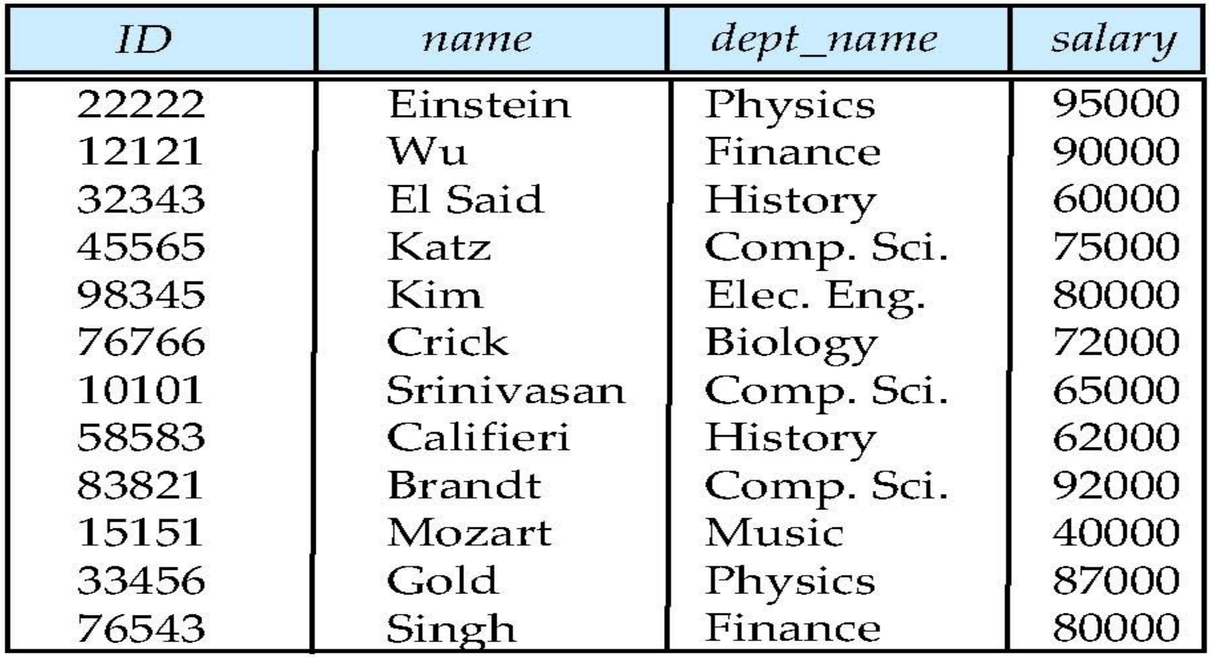
- the ability to modify the physical schema without changing the logical schema
  - Applications depend on the logical schema
  - In general, the interfaces between the various levels and components should be well defined so that changes in some parts do not seriously influence others.

Data Models

- A collection of tools for describing
  - Data
  - Data relationships
  - Data semantics
  - Data constraints
- Relational model (most widely used)
- Entity-Relationship data model (mainly for database design)
- Object-based data models (Object-oriented and Object-relational)
- Semistructured data model (XML)

Relational Model

- All the data is stored in various tables.
- Example of tabular data in the relational model



ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

A Sample Relational Database

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

dept_name	*building*	*budget*
Comp. Sci.	Taylor	100000
Biology	Watson	90000
Elec. Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

(b) The *department* table

Data Definition Language (DDL)

- Specification notation for defining the database schema

  Example: **create table** *instructor* (

  ```
  ID          char(5),
  name        varchar(20),
  dept_name   varchar(20),
  salary      numeric(8,2))
  ```

- DDL compiler generates a set of table templates stored in a Data Dictionary
- Data dictionary contains metadata (i.e., data about data)
  - Database schema
  - Integrity constraints
    - Primary key (ID uniquely identifies instructors)
  - Authorization
    - Who can access what

Data Manipulation Language (DML)

- Language for accessing and manipulating the data organized by the appropriate data model
  - **Procedural** – what and how
  - **Declarative (nonprocedural)** – what not how
- Query language: the portion of a DML that involves information retrieval
- Often use the term query language and DML synonymously

Query language

- **Pure (formal relational)**– used for proving properties about computational power and for optimization
  - Relational Algebra
  - Tuple relational calculus
  - Domain relational calculus

- **Commercial** – used in commercial systems
  - SQL(Structured Query Language) is the most widely used commercial language

SQL

- Nonprocedural
- Takes several tables(possibly only one) and always return a single table
- To be able to compute complex functions, SQL is usually embedded in some higher-level language
- Application programs generally access databases through one of
  - Language extensions to allow embedded SQL
  - Application program interface (e.g., ODBC/JDBC) which allow SQL queries to be sent to a database

Database Design

- The process of designing the general structure of the database:
  - Logical Design – Deciding on the database schema. Database design requires that we find a “good” collection of relation schemas.
    - Business decision – What attributes should we record in the database?
    - Computer Science decision – What relation schemas should we have and how should the attributes be distributed among the various relation schemas?
  - Physical Design – Deciding on the physical layout of the database

Discussion

- database design for a University organization
- Interviews with database users
- Designer’s own analysis
- Major characteristics of the university

Design Approaches

- Need to come up with a methodology to ensure that each of the relations in the database is “good”
- Two ways of doing so:
  - Entity Relationship Model
    - Models an enterprise as a collection of *entities* and *relationships*
    - Represented diagrammatically by an *entity-relationship diagram:*
  - Normalization Theory
    - Formalize what designs are bad, and test for them

Object-Relational Data Models

- Relational model: flat, “atomic” values
- Object Relational Data Models
  - Extend the relational data model by including object orientation and constructs to deal with added data types.
  - Allow attributes of tuples to have complex types, including non-atomic values such as nested relations.
  - Preserve relational foundations, in particular the declarative access to data, while extending modeling power.
  - Provide upward compatibility with existing relational languages.

XML: Extensible Markup Language

- Defined by the WWW Consortium (W3C)
- Originally intended as a document markup language not a database language
- The ability to specify new tags, and to create nested tag structures made XML a great way to exchange **data**, not just documents
- XML has become the basis for all new generation data interchange formats.
- A wide variety of tools is available for parsing, browsing and querying XML documents/data

Database classification: types of Databases

- Databases can be classified according to:
  - Number of users
  - Database location(s)
  - Type of data stored
  - Intended data usage
  - Degree to which the data are structured

Number of users supported

- **Single-user database** supports only one user at a time
  - Desktop database: single-user; runs on PC

- **Multiuser database** supports multiple users at the same time
  - Workgroup and enterprise databases

The type of data stored

- **General-purpose databases** contain a wide variety of data used in multiple disciplines.
  - A census database contains general demographical data.
  - The ProQuest database contains newspaper, magazine, and journal articles for a variety of topics.

- **Discipline-specific databases** contain specific subject area data.
  - ComputStat and CRSP databases store financial data.
  - Geographic information system (GIS) databases store geospatial and related data.
  - Medical databases store confidential medical history data.

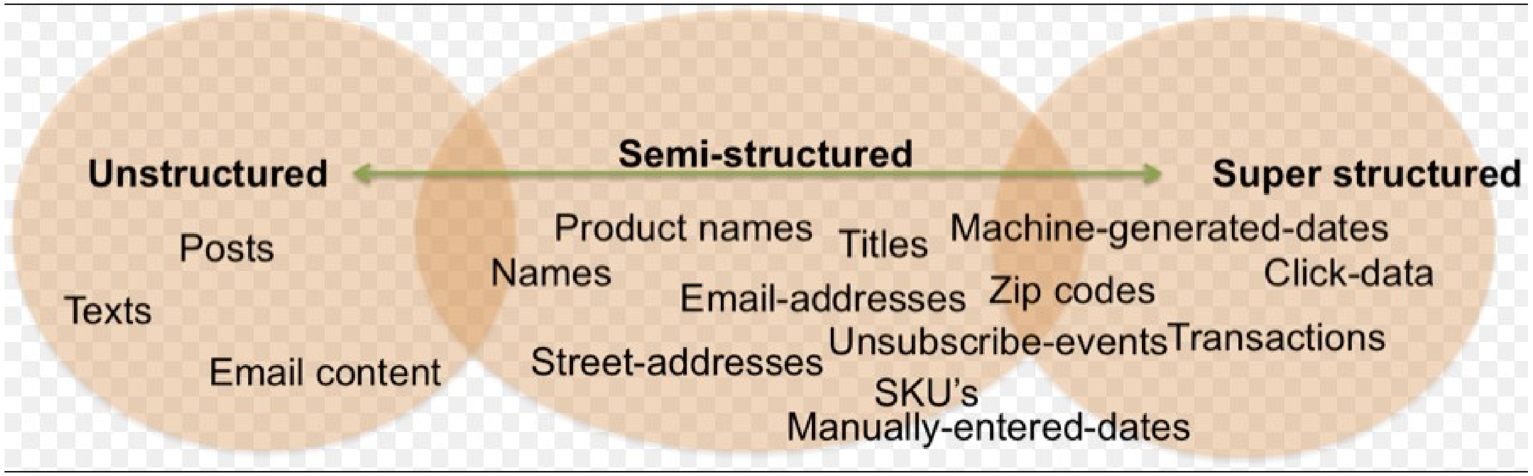
Intended data usage

- **Operational database** supports a company’s day-to-day operations
  - Transactional or production database

- **Analytical databases** supports for tactical or strategic decision making
  - Data warehouse: data stored is used for tactical or strategic decisions
  - Online analytical processing(OLAP) front end: a set of tools that work together to provide an advanced data analysis environment for retrieving, processing, and modeling data from the data warehouse.

Degree the data are structured

- **Unstructured data** exist in their original state
- **Structured data** result from formatting
  - Structure applied based on type of processing to be performed
- **Semistructured data** have been processed to some extent



Databases provided by various DBMS vendors

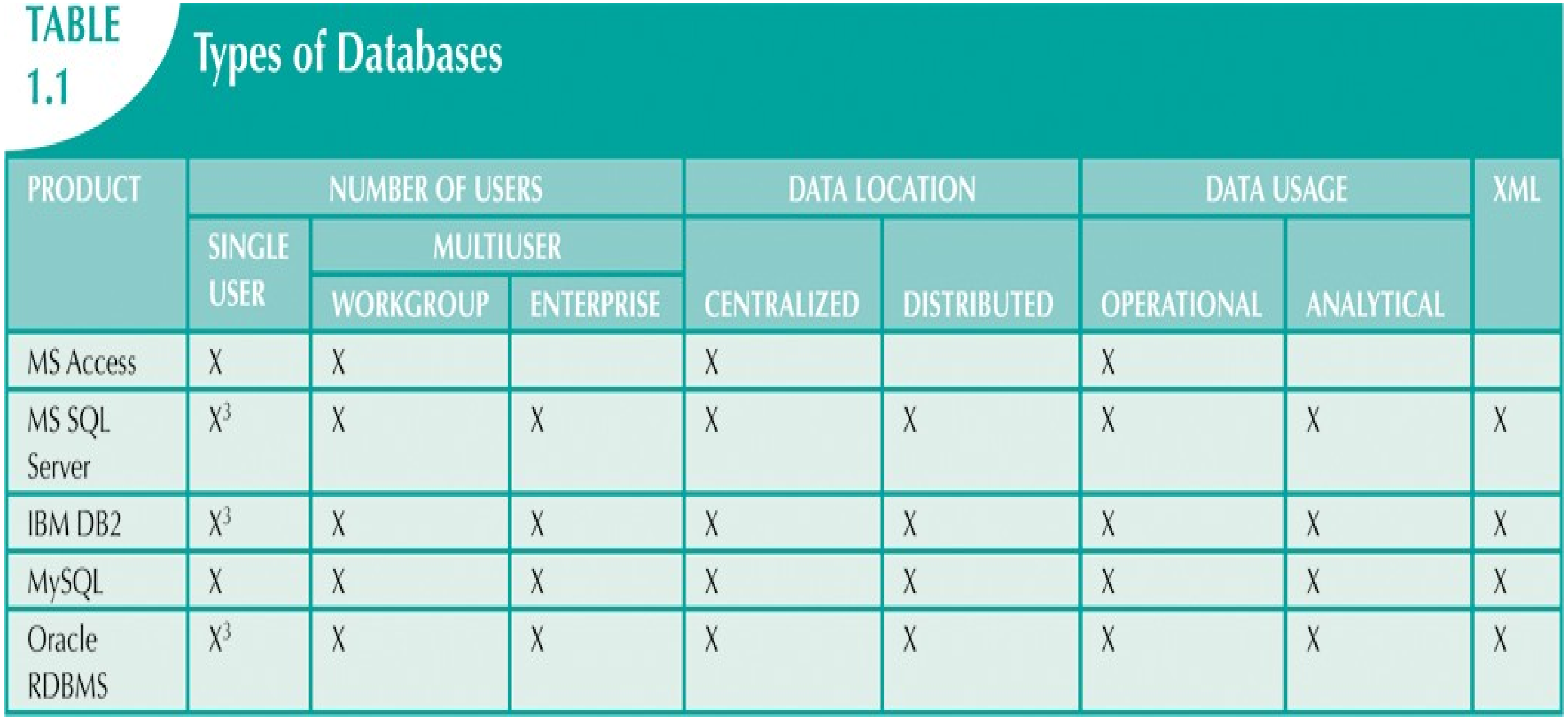


TABLE 1.1 Types of Databases

PRODUCT	NUMBER OF USERS			DATA LOCATION		DATA USAGE		XML
	SINGLE USER	MULTIUSER						
		WORKGROUP	ENTERPRISE	CENTRALIZED	DISTRIBUTED	OPERATIONAL	ANALYTICAL	
MS Access	X	X		X		X		
MS SQL Server	X[3]	X	X	X	X	X	X	X
IBM DB2	X[3]	X	X	X	X	X	X	X
MySQL	X	X	X	X	X	X	X	X
Oracle RDBMS	X[3]	X	X	X	X	X	X	X

History of Database Systems

- 1950s and early 1960s:
  - Data processing using magnetic tapes for storage
    - Tapes provided only sequential access
  - Punched cards for input
- Late 1960s and 1970s:
  - Hard disks allowed direct access to data
  - Network and hierarchical data models in widespread use
  - Ted Codd defines the relational data model
    - won the ACM Turing Award for this work
  - High-performance (for the era) transaction processing

History of Database Systems

- 1980s:
  - Research relational prototypes evolve into commercial systems
    - SQL becomes industrial standard
  - Parallel and distributed database systems
  - Object-oriented database systems
- 1990s:
  - Large decision support and data-mining applications
  - Large multi-terabyte data warehouses
  - Emergence of Web commerce

History of Database Systems

- Early 2000s:
  - XML and XQuery standards
  - Automated database administration
- Later 2000s:
  - Giant data storage systems
    - Google BigTable
    - Amazon Database on AWS

Thank you & Questions