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

John Smith, CS113, js@etown.edu, B

Jim Gray, CS560, jg@etown.edu, A

John Smith, CS560, js@etown.edu, B+

Name, Email, Course, Grade

Mike Stonebraker, ms@etown.edu, CS234, A

J. Smith, js@etown.edu, CS560, 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

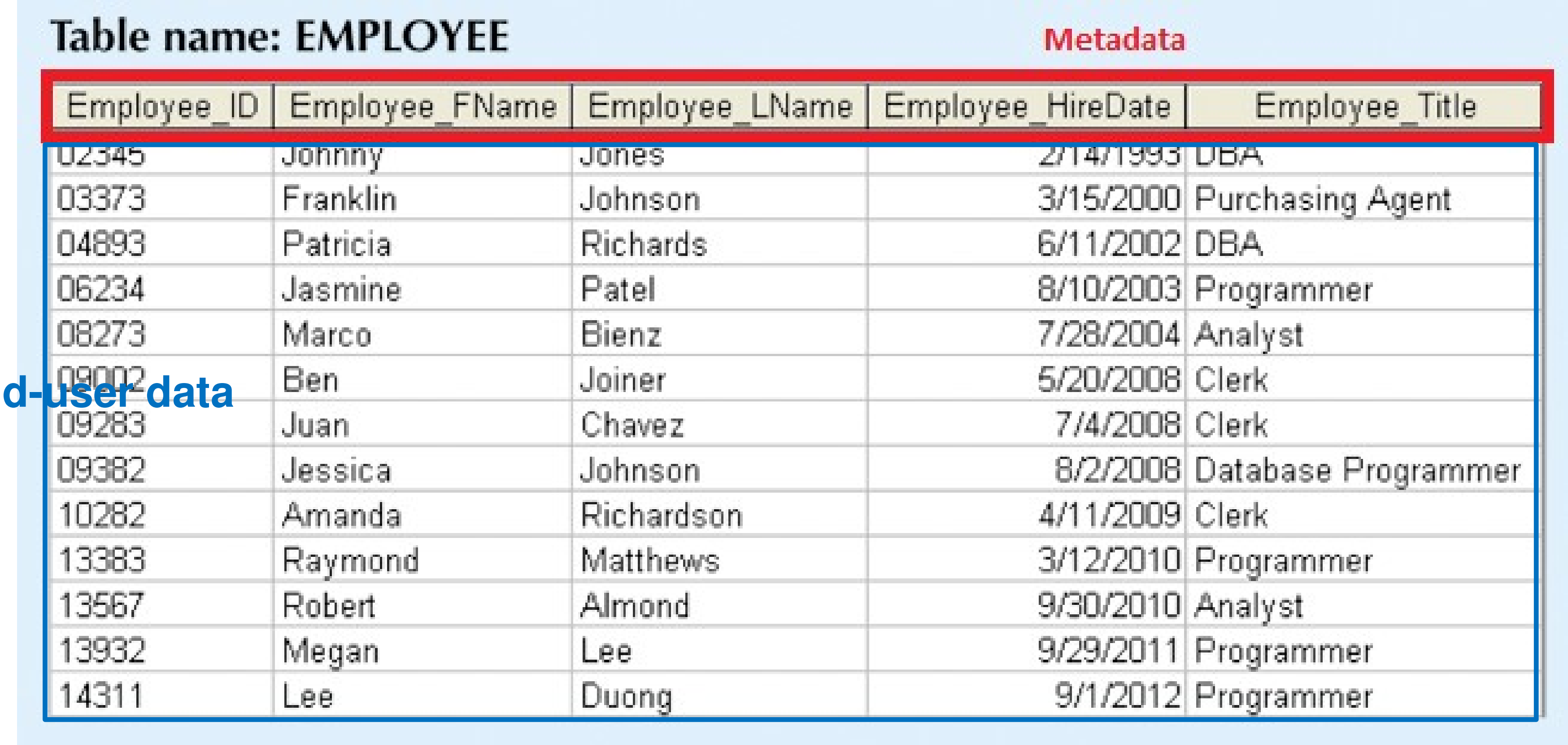


Table name: EMPLOYEE

Metadata

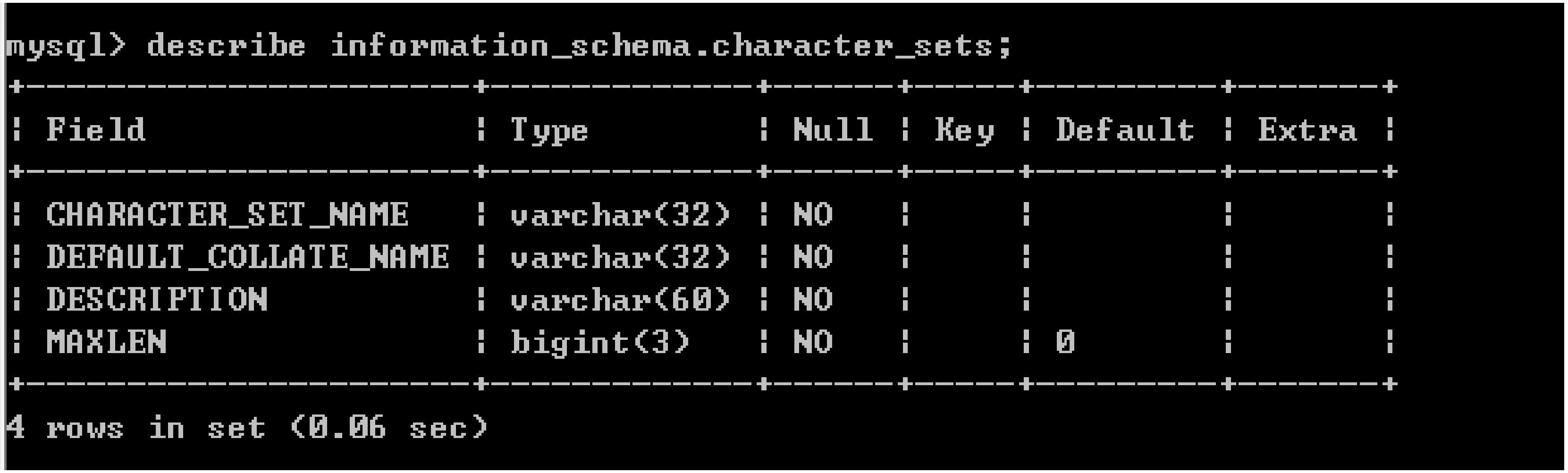
End-user data

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

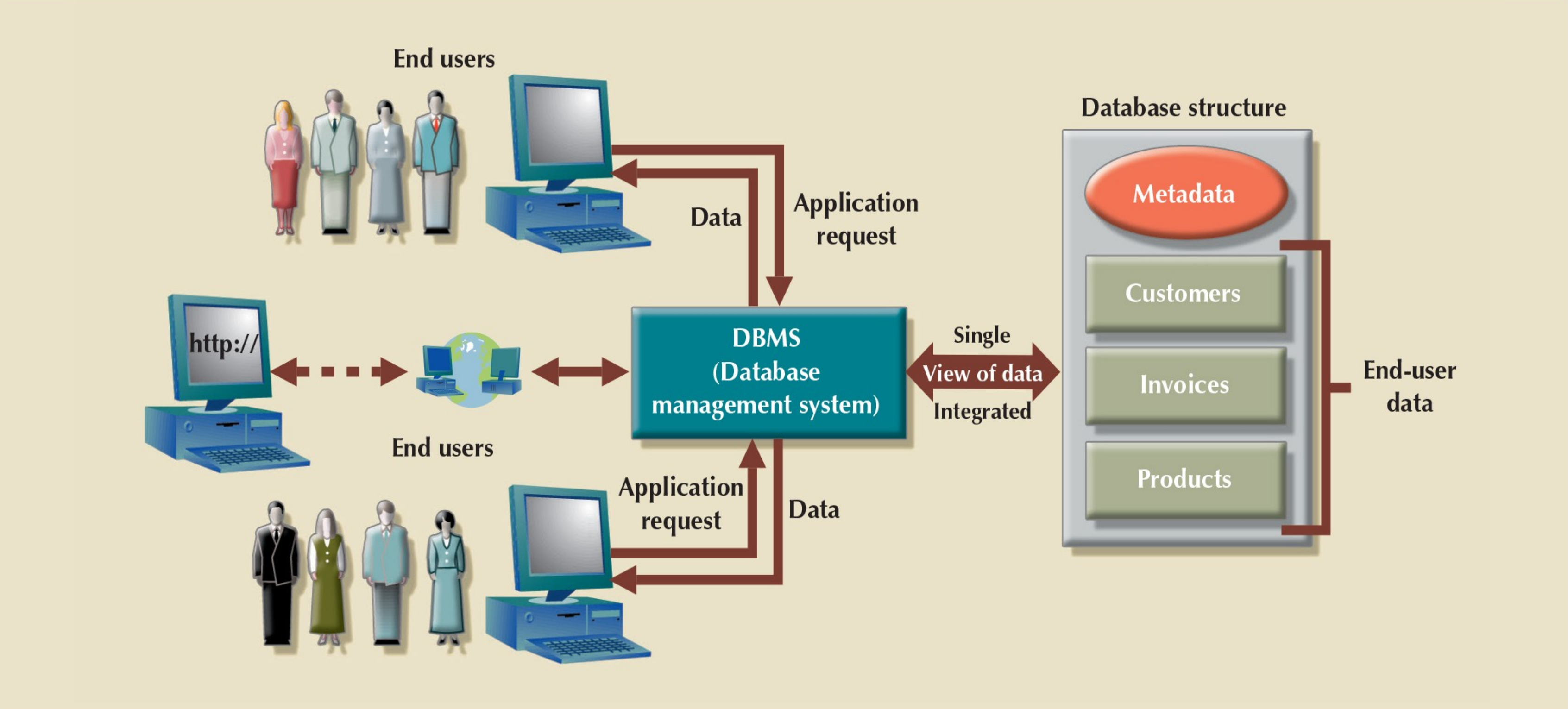
Role of the DBMS

- 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

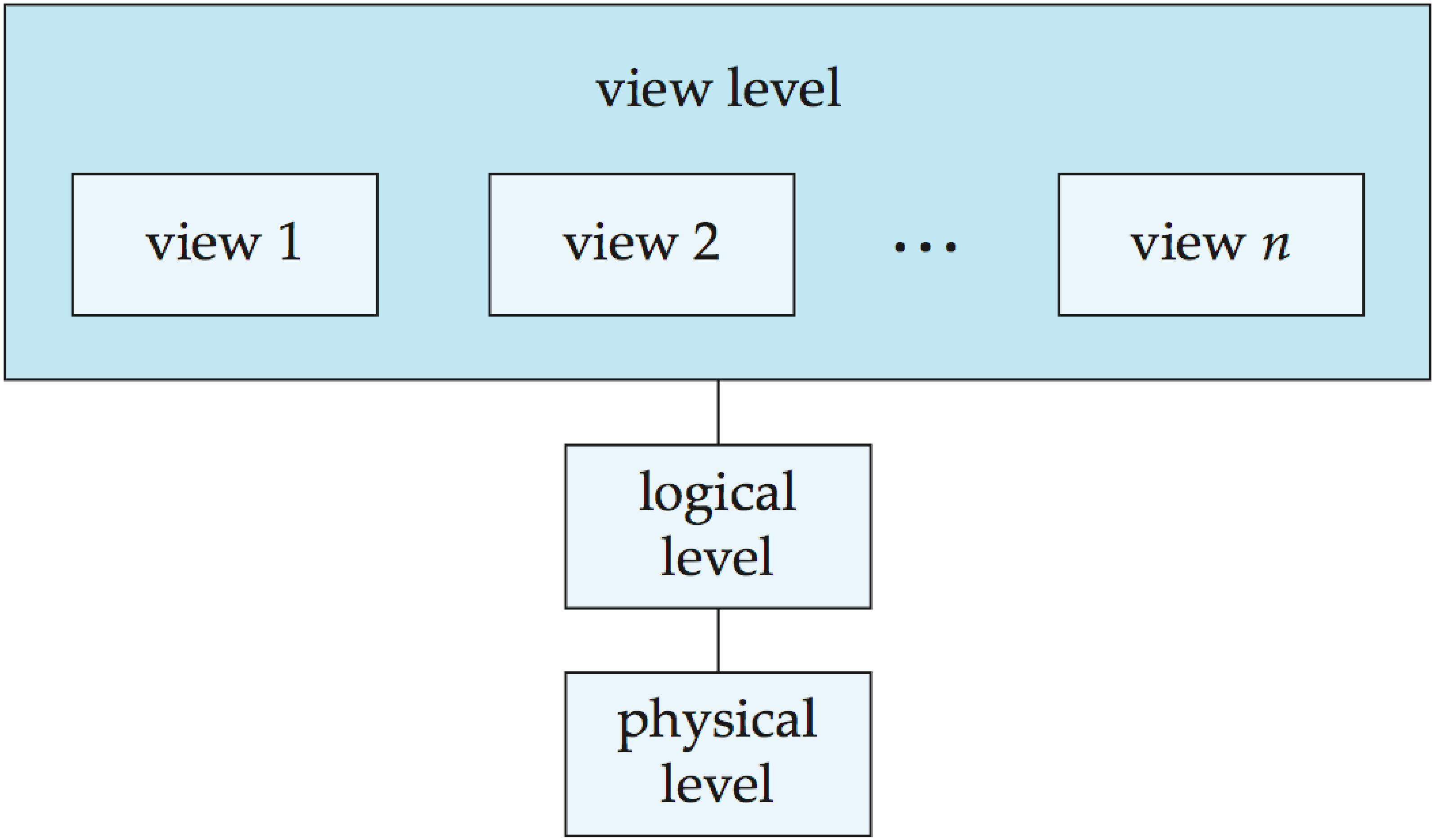
Levels of Abstraction

- **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

Thank you & Questions