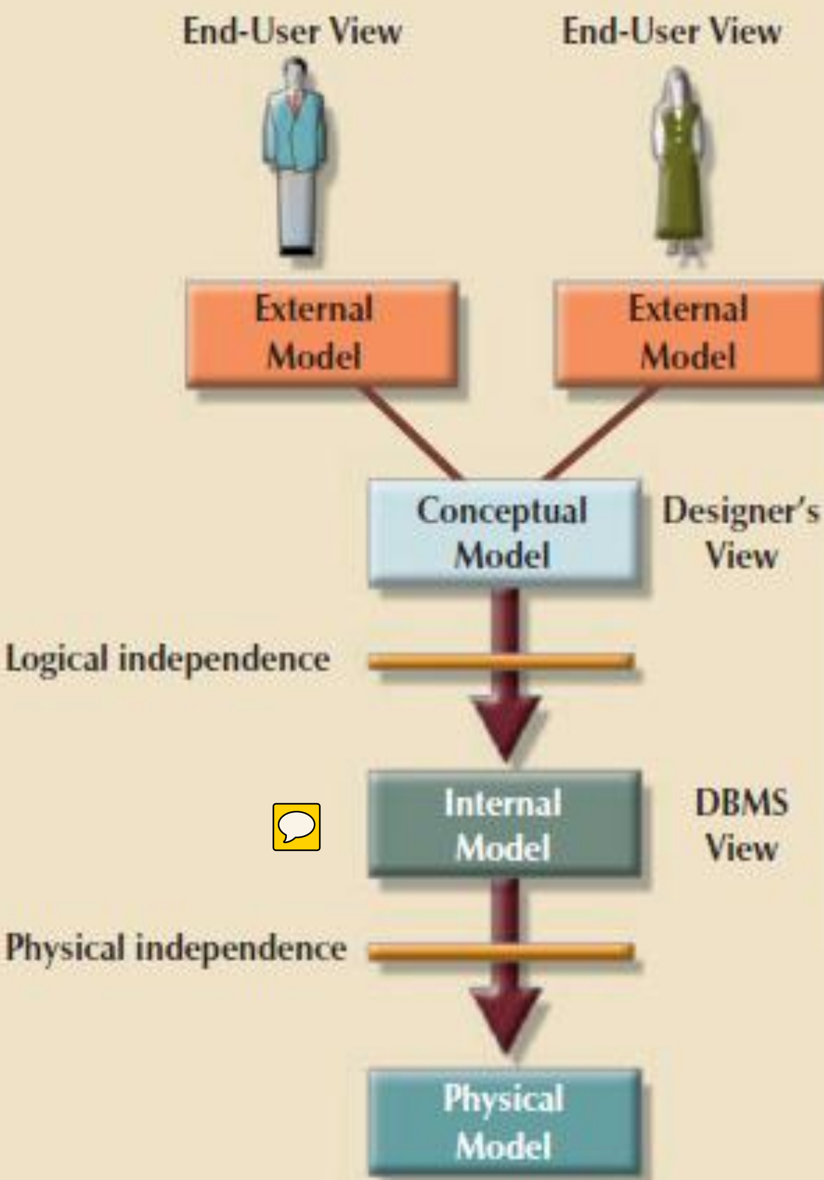
Week No. 7
Database Management System

FIGURE 2.8 Data abstraction levels



Degree of Abstraction		Characteristics
High	ER	Hardware-independent Software-independent
	Object-Oriented	
Medium	Relational	Hardware-independent Software-dependent
	Network	Hardware-dependent Software-dependent
Low	Hierarchical	

Data Abstraction levels:

Software independence:

The model does not depend on the DBMS software used to implement the model.

Hardware independence:

The model does not depend on the hardware used in the implementation of the model.

The External Model

- The external model is the **end users' view of the data environment**. The term end users refers to people who use the application programs to manipulate the data and generate information. **End users usually operate in an environment in which an application has a specific business unit focus.**

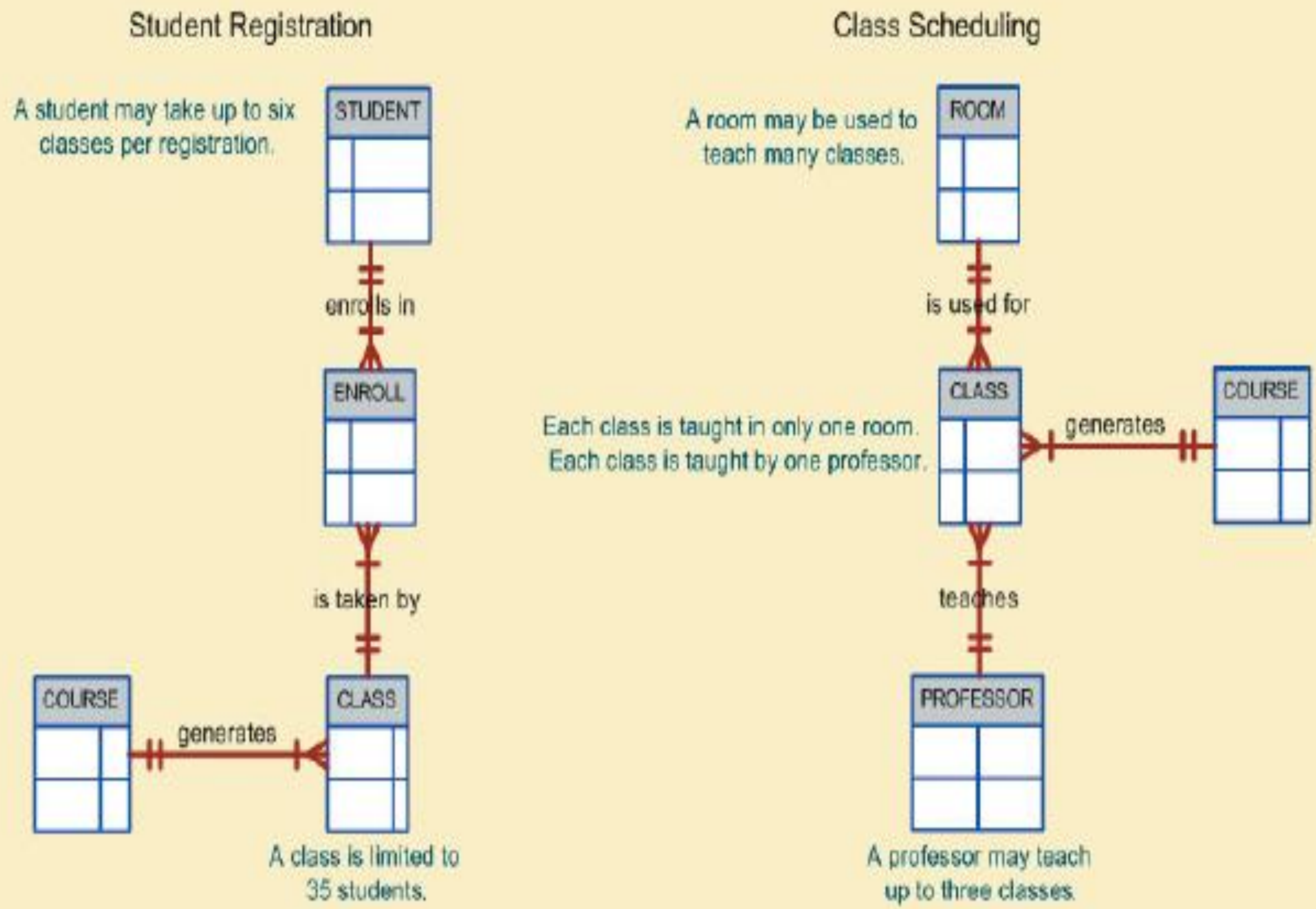
The External Model

- Companies are generally divided into **several business units**, such as sales, finance, and marketing. Each business unit is **subject to specific constraints and requirements**, and each one uses a data subset of the overall data in the organization. Therefore, **end users working within those business units view their data subsets as separate from or external to other units within the organization**.

The External Model

- Because data is being modeled, **ER diagrams** will be used to represent the **external views**. A specific representation of an external view is known as an **external schema.**

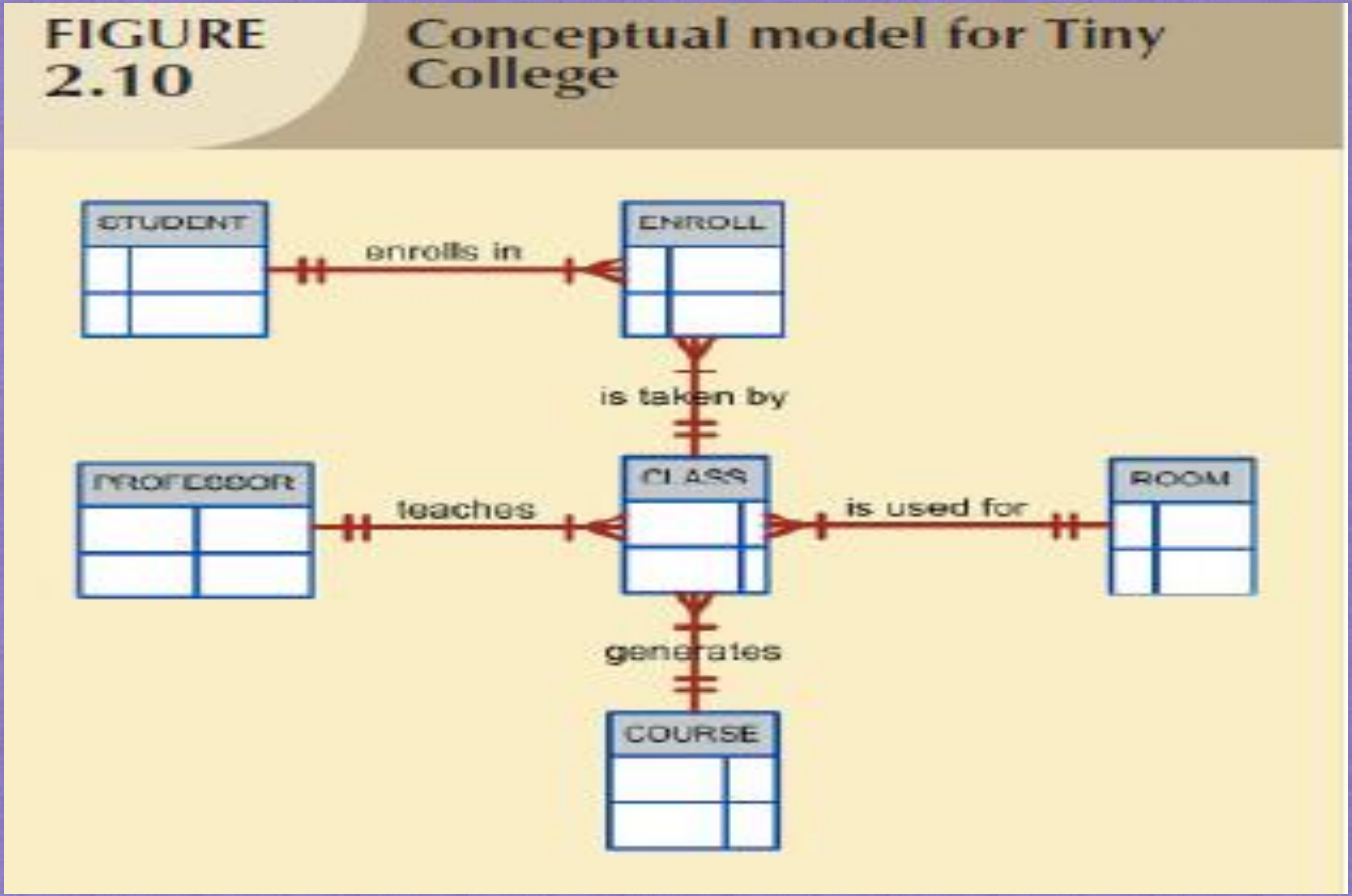
FIGURE 2.9 External models for Tiny College



The Conceptual Model

- The conceptual model represents **a global view of the entire database** as viewed by the entire organization. That is, the conceptual model **integrates all external views (entities, relationships, constraints, and processes)** into a single global view of the entire data in the enterprise.
- Also known as **a conceptual schema**, it is the basis for the identification and high-level description of the main data objects (avoiding any database model-specific details).

The Conceptual Model

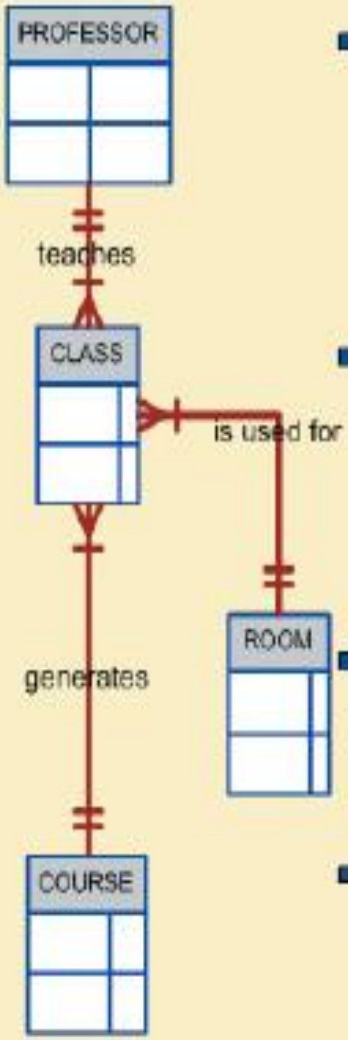


The Internal Model

- The internal model is the representation of the database as **“*seen*” by the DBMS**.
- In other words, **the internal model requires the designer to match the conceptual model’s characteristics and constraints to those of the selected implementation model**.
- An **internal schema** depicts a specific representation of an internal model, using the database constructs supported by the chosen database.

FIGURE 2.11 An internal model for Tiny College

CONCEPTUAL MODEL



INTERNAL MODEL

```
Create Table PROFESSOR(
 PROF_ID          NUMBER PRIMARY KEY,
 PROF_LNAME       CHAR(15),
 PROF_INITIAL     CHAR(1),
 PROF_FNAME       CHAR(15),
 .........);

Create Table CLASS(
 CLASS_ID         NUMBER PRIMARY KEY,
 CRS_ID           CHAR(8)  REFERENCES COURSE,
 PROF_ID          NUMBER REFERENCES PROFESSOR,
 ROOM_ID          CHAR(8)  REFERENCES ROOM,
 .........);

Create Table ROOM(
 ROOM_ID          CHAR(8) PRIMARY KEY,
 ROOM_TYPE        CHAR(3),
 .........);

Create Table COURSE(
 CRS_ID           CHAR(8) PRIMARY KEY,
 CRS_NAME         CHAR(25),
 CRS_CREDITS      NUMBER,
 .........);
```

The Internal Model

- When you can change the internal model without affecting the conceptual model, you have **logical independence.**
- However, the internal model is still **hardware-independent** because it is unaffected by the choice of the computer on which the software is installed. Therefore, a change in storage devices or even a change in operating systems will not affect the internal model.

The Physical Model

- The physical model operates at the **lowest level of abstraction, describing the way data are saved on storage media such as disks or tapes**. The physical model requires the definition of both the **physical storage devices and the (physical) access methods** required to reach the data within those storage devices, making it both **software- and hardware dependent**.

The Physical Model

- The storage structures used are dependent on the software (the DBMS and the operating system) and on the type of storage devices that the computer can handle. The precision required in the physical model's definition demands that database designers who work at this level have **a detailed knowledge of the hardware and software used to implement the database design.**
- **When you can change the physical model without affecting the internal model, you have physical independence.** Therefore, a change in storage devices or methods and even a change in operating system will not affect the internal model.

A summary of the levels of data abstraction is given in Table 2.4.

TABLE 2.4 Levels of Data Abstraction

MODEL	DEGREE OF ABSTRACTION	FOCUS	INDEPENDENT OF
External	High	End-user views	Hardware and software
Conceptual	↑	Global view of data (database model independent)	Hardware and software
Internal	↓	Specific database model	Hardware
Physical	Low	Storage and access methods	Neither hardware nor software

End of Lecture

Any Questions... ?

Chapter 3

The Relational Database Model

In this chapter, you will learn:

- **Basic components of the relational database model**
  - Entities and their attributes
  - Relationships among entities
- **Relational algebra**
- **Relationship in relational database**
- **Data redundancy**

Predicate logic and Set Theory

- The relational model, introduced by E. F. Codd in 1970, is based on predicate logic and set theory.
- **Predicate logic:** provides a framework in which an assertion (statement of fact) can be verified as either true or false.
- **Set theory:** a mathematical science that deals with sets, or groups of things, and is used as the basis for data manipulation in the relational model.

Basic Definition

- **Entities and Attributes**
  - Entity is a person, place, event, or thing about which data is collected
  - Attributes are characteristics of the entity
- **Tables**
  - Holds related entities or entity set
  - Also called relations
  - Comprised of rows and columns

Table Characteristics

- Two-dimensional structure with rows and columns
- Rows (tuples) represent single entity
- Columns represent attributes
- Row/column intersection represents single value
- Tables must have an attribute to uniquely identify each row
- Column values all have same data format
- Each column has range of values called attribute domain
- Order of the rows and columns is immaterial to the DBMS

Example Tables

Database name: Ch03_TinyCollege
Table name: STUDENT

STU_NUM	STU_LNAME	STU_FNAME	STU_INIT	STU_DOB	STU_HRS	STU_CLASS
321452	Bowser	William	C	12-Feb-1975	42	So
324257	Smithson	Anne	K	15-Nov-1981	81	Jr
324258	Brewer	Juliette		23-Aug-1969	36	So
324269	Oblonski	Walter	H	16-Sep-1976	66	Jr
324273	Smith	John	D	30-Dec-1958	102	Sr
324274	Katinga	Raphael	P	21-Oct-1979	114	Sr
324291	Robertson	Gerald	T	08-Apr-1973	120	Sr
324299	Smith	John	B	30-Nov-1986	15	Fr

STUDENT table, continued

STU_GPA	STU_TRANSFER	DEPT_CODE	STU_PHONE	PROF_NUM
2.84	No	BIOL	2134	205
3.27	Yes	CIS	2256	222
2.26	Yes	ACCT	2256	228
3.09	No	CIS	2114	222
2.11	Yes	ENGL	2231	199
3.15	No	ACCT	2267	228
3.87	No	EDU	2267	311
2.92	No	ACCT	2315	230

STU_HRS = Credit hours earned
STU_CLASS = Student classification
STU_DOB = Student date of birth
STU_GPA = Grade point average
STU_PHONE = 4-digit campus phone extension
PROF_NUM = Number of the professor who is the student's advisor

Example Tables

- The STUDENT table is perceived to be a two-dimensional structure composed of eight rows (tuples) and twelve columns (attributes).
- Each row in the STUDENT table describes a single entity occurrence within the entity set. (The entity set is represented by the STUDENT table.) Note that the row (entity or record) defined by STU_NUM = 321452 defines the characteristics (attributes or fields) of a student named William C. Bowser. For example, row 4 in Figure describes a student named Walter H. Oblonski. Similarly, row 3 describes a student named Juliette Brewer. Given the table contents, the STUDENT entity set includes eight distinct entities (rows), or students.

Example Tables

Database name: Ch03_TinyCollege
Table name: STUDENT

STU_NUM	STU_LNAME	STU_FNAME	STU_INIT	STU_DOB	STU_HRS	STU_CLASS	STU_GPA	STU_TRANSFER	DEPT_CODE	STU_PHONE	PROF_NUM
321452	Bowser	William	C	12-Feb-1975	42	So	2.84	No	BIOL	2134	205
324257	Smithson	Anne	K	15-Nov-1981	81	Jr	3.27	Yes	CIS	2256	222
324258	Brewer	Juliette		23-Aug-1969	36	So	2.26	Yes	ACCT	2256	228
324269	Oblonski	Walter	H	16-Sep-1976	66	Jr	3.09	No	CIS	2114	222
324273	Smith	John	D	30-Dec-1958	102	Sr	2.11	Yes	ENGL	2231	199
324274	Katinga	Raphael	P	21-Oct-1979	114	Sr	3.15	No	ACCT	2267	228
324291	Robertson	Gerald	T	08-Apr-1973	120	Sr	3.87	No	EDU	2267	311
324299	Smith	John	B	30-Nov-1986	15	Fr	2.92	No	ACCT	2315	230

STUDENT table, continued

STU_HRS = Credit hours earned
STU_CLASS = Student classification
STU_DOB = Student date of birth
STU_GPA = Grade point average
STU_PHONE = 4-digit campus phone extension
PROF_NUM = Number of the professor who is the student's advisor

Example Tables

- Each column represents an attribute, and each column has a distinct name.
- All of the values in a column match the attribute's characteristics. For example, the grade point average (STU_GPA) column contains only STU_GPA entries for each of the table rows. Data must be classified according to their format and function. Although various DBMSs can support different data types, most support at least the following:
- a. Numeric. Numeric data are data on which you can perform meaningful arithmetic procedures. For example, STU_HRS and STU_GPA in Figure are numeric attributes. On the other hand, STU_PHONE is not a numeric attribute because adding or subtracting phone numbers does not yield an arithmetically meaningful result.
- b. Character. Character data, also known as text data or string data, can contain any character or symbol not intended for mathematical manipulation. In Figure, for example, STU_LNAME, STU_FNAME,STU_INIT, STU_CLASS, and STU_PHONE are character attributes.

Example Tables

Database name: Ch03_TinyCollege
Table name: STUDENT

STU_NUM	STU_LNAME	STU_FNAME	STU_INIT	STU_DOB	STU_HRS	STU_CLASS
321452	Bowser	William	C	12-Feb-1975	42	So
324257	Smithson	Anne	K	15-Nov-1981	81	Jr
324258	Brewer	Juliette		23-Aug-1969	36	So
324269	Oblonski	Walter	H	16-Sep-1976	66	Jr
324273	Smith	John	D	30-Dec-1958	102	Sr
324274	Katinga	Raphael	P	21-Oct-1979	114	Sr
324291	Robertson	Gerald	T	08-Apr-1973	120	Sr
324299	Smith	John	B	30-Nov-1986	15	Fr

STUDENT table, continued

STU_GPA	STU_TRANSFER	DEPT_CODE	STU_PHONE	PROF_NUM
2.84	No	BIOL	2134	205
3.27	Yes	CIS	2256	222
2.26	Yes	ACCT	2256	228
3.09	No	CIS	2114	222
2.11	Yes	ENGL	2231	199
3.15	No	ACCT	2267	228
3.87	No	EDU	2267	311
2.92	No	ACCT	2315	230

STU_HRS = Credit hours earned
STU_CLASS = Student classification
STU_DOB = Student date of birth
STU_GPA = Grade point average
STU_PHONE = 4-digit campus phone extension
PROF_NUM = Number of the professor who is the student's advisor

Example Tables

- c. Date. Date attributes contain calendar dates stored in a special format known as the Julian date format. Although the physical storage of the Julian date is immaterial to the user and designer, the Julian date format allows you to perform a special kind of arithmetic known as Julian date arithmetic. Using Julian date arithmetic, you can determine the number of days that have elapsed between two dates, such as 12-May-1999 and 20-Mar-2008, by simply subtracting 12-May-1999 from 20-Mar-2008. In Figure, STU_DOB can properly be classified as a date attribute. Most relational database software packages support Julian date formats. While the database's internal date format is likely to be Julian, many different presentation formats are available. For example, in Figure, you could show Mr. Bowser's date of birth (STU_DOB) as 2/12/75. Most relational DBMSs allow you to define your own date presentation format. For instance, Access and Oracle users might specify the "dd-mmm-yyyy" date format to show the first STU_DOB value in Figure as 12-Feb-1975. (As you can tell by examining the STU_DOB values in Figure, the "dd-mmm-yyyy" format was selected to present the output.)

Example Tables

Database name: Ch03_TinyCollege
Table name: STUDENT

STU_NUM	STU_LNAME	STU_FNAME	STU_INIT	STU_DOB	STU_HRS	STU_CLASS
321452	Bowser	William	C	12-Feb-1975	42	So
324257	Smithson	Anne	K	15-Nov-1981	81	Jr
324258	Brewer	Juliette		23-Aug-1969	36	So
324269	Oblonski	Walter	H	16-Sep-1976	66	Jr
324273	Smith	John	D	30-Dec-1958	102	Sr
324274	Katinga	Raphael	P	21-Oct-1979	114	Sr
324291	Robertson	Gerald	T	08-Apr-1973	120	Sr
324299	Smith	John	B	30-Nov-1986	15	Fr

STUDENT table, continued

STU_GPA	STU_TRANSFER	DEPT_CODE	STU_PHONE	PROF_NUM
2.84	No	BIOL	2134	205
3.27	Yes	CIS	2256	222
2.26	Yes	ACCT	2256	228
3.09	No	CIS	2114	222
2.11	Yes	ENGL	2231	199
3.15	No	ACCT	2267	228
3.87	No	EDU	2267	311
2.92	No	ACCT	2315	230

STU_HRS = Credit hours earned
STU_CLASS = Student classification
STU_DOB = Student date of birth
STU_GPA = Grade point average
STU_PHONE = 4-digit campus phone extension
PROF_NUM = Number of the professor who is the student's advisor

Example Tables

- d. Logical. Logical data can have only a true or false (yes or no) condition. For example, is a student a junior college transfer? In Figure, the STU_TRANSFER attribute uses a logical data format. Most, but not all, relational database software packages support the logical data format. (Microsoft Access uses the label “Yes/No data type” to indicate a logical data type.)
- The column’s range of permissible values is known as its domain. Because the STU_GPA values are limited to the range 0–4, inclusive, the domain is [0,4].
- The order of rows and columns is immaterial to the user.
- Each table must have a primary key. In general terms, the primary key (PK) is an attribute (or a combination of attributes) that uniquely identifies any given row. In this case, STU_NUM (the student number) is the primary key. Using the data presented in Figure, observe that a student’s last name (STU_LNAME) would not be a good primary key because it is possible to find several students whose last name is Smith. Even the combination of the last name and first name (STU_FNAME) would not be an appropriate primary key because, as Figure shows, it is quite possible to find more than one student named John Smith.

Terminology for Relational Database

Table-Oriented	Set-oriented	Record-Oriented
Table	Relation	Record type
Row	Tuple	Record
Column	Attribute	Field

Tables are labeled files. Technically speaking, this substitution of terms is not always appropriate; the database table is a logical rather than a physical concept, and the terms file, record, and field describe physical concepts.

End of the Lecture