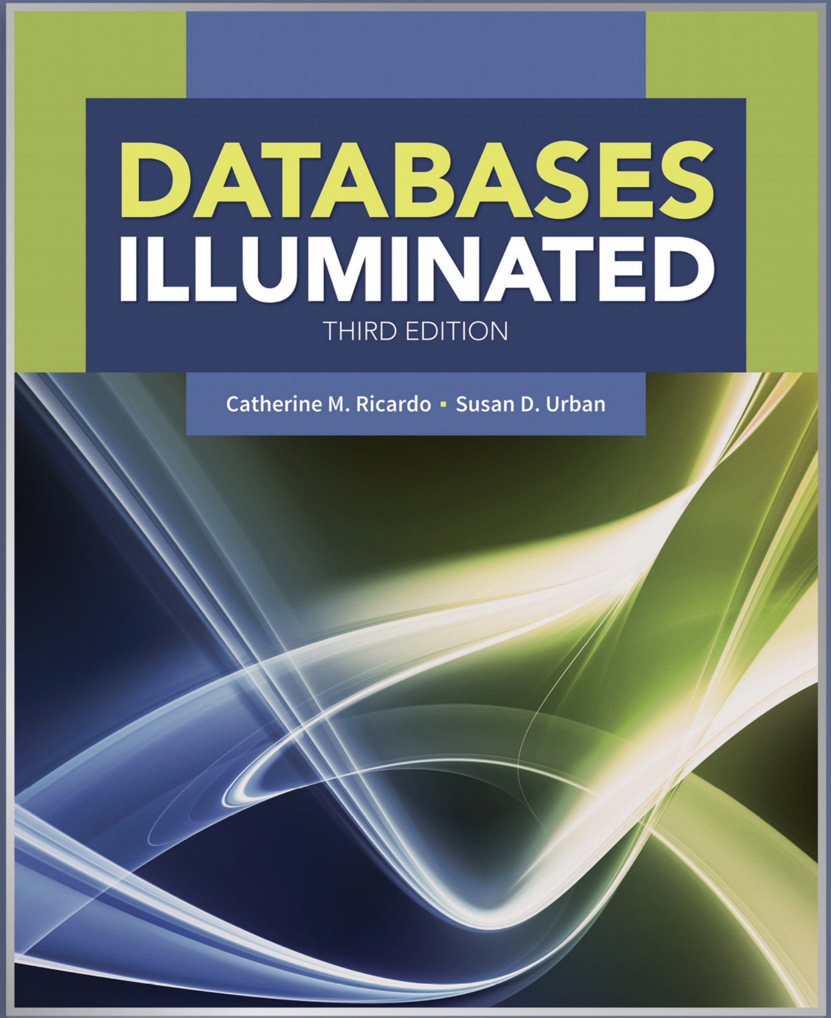
Databases Illuminated

Chapter 2
Database Planning and Database Architecture



Data as a Resource

- **Resource**: An asset that has value and incurs cost
- Resources in an organization include capital equipment, financial assets, personnel **and data**
- **Database** is a resource because
  - Operational data has value
  - Database incurs cost
  - Professionally managed by DBA

Characteristics of Data

- Data vs. information
  - Data: raw facts
    - Example: printout of tables as they are stored, without headings saying what they mean
  - Information: processed data, useful for decision-making
    - Example: formatted report using database

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

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

Four Levels of Data

1. Real world
   - **Enterprise** in its environment
   - **Mini-world**, part of the world represented in the database
2. Conceptual Model
   - Entities, entity sets, attributes, relationships
   - Often represented as E-R (ER: Image) (ER Model), EE-R (Wiki) (EER Model), or UML (UML) diagram (UML Model)
3. Logical model of database-intension
   - **Metadata**, data about data
   - Record types, data item types, data aggregates
   - Schema - stored in system catalog
4. Data occurrences-extension
   - Database itself
   - Data instances
   - Files (Data Models)

FIGURE 2.1
Four Levels of Discussing Data

Realm	Objects	Examples
Real World Mini-World	Enterprise	Corporation University Bank
Conceptual Model	Entity Attribute Relationship	Research Department LastName HasAccount
Logical Model	Record Type Data Item Type Data Aggregate	Dept Record Type StuId Address
Data Occurrences	Records Files Database	Dept 12 Record Student File University DB

Data Sublanguages

- DBMS uses a data sublanguage, with at least two parts (DDL-DML)
  - **Data definition language** (**DDL**) - used to define the database (ex: CREATE, DROP, ALTER)
  - **Data manipulation language** (**DML**) - used to process the database (ex: INSERT, UPDATE, DELETE, SELECT)
  - Data sublanguage may be embedded in a host language-general programming language, such as C, C++, C#, Java,…

Characteristics of a Conceptual Database Model

- Faithfully mirrors the operations of the organization
- Flexible enough to allow changes as new information needs arise
- Supports many different user views
- Independent of physical implementation
- Does not depend on the model used by a particular database management system

Stages in Database Design (DataModels)

- Analyze user environment
- Develop conceptual data model
- Choose a DBMS
- Develop logical model, by mapping conceptual model to DBMS
- Develop physical model
- Evaluate physical model
- Perform tuning, if indicated
- Implement physical model

Note loops in Figure 2.2

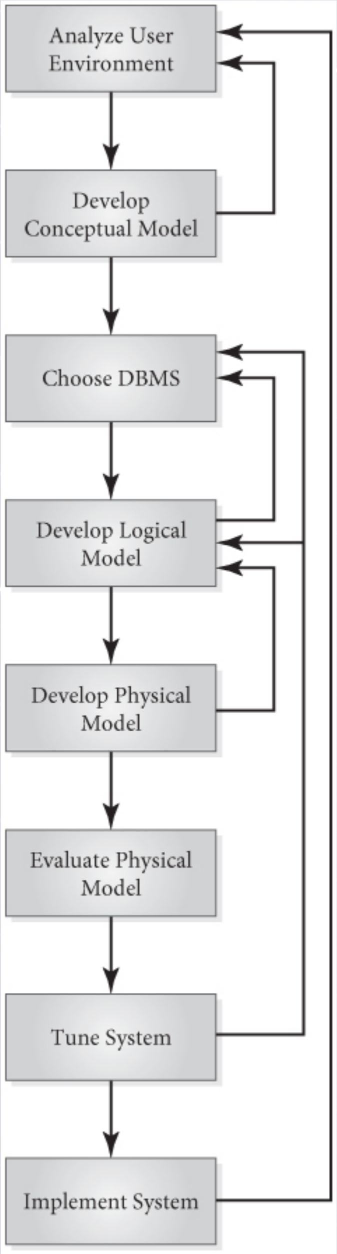


Figure 2.2 Steps in Staged Database Design

Design Tools

- Data dictionary
  - May be freestanding
  - System catalog
- Project management software
  - Graphs, charts, document control, communication
- Diagramming tools
  - E-R, UML diagrams
- CASE (Computer-Aided Software Engineering) tools (wiki)
  - System analysis, project management, design

Data Dictionary (DataDict.)

- Contains **metadata**
- Can be **integrated** (system catalog, part of DBMS) or **free-standing**
- Both types are useful for
  - Collecting information about data in central location
  - Securing agreement on meanings of items
  - Communicating with users
  - Identifying inconsistencies – synonyms and homonyms
  - Keeping track of changes to DB structure
  - Determining impact of changes to DB structure
  - Identifying sources of/responsibility for items
  - Recording external/logical/physical models & mappings
  - Recording access control information
  - Providing audit information
- System Catalog also provides audit information

Project Management Software

- Tools to plan and manage projects, especially with many people
- Includes several types of charts and graphs
  - GANTT chart- See Figure 2.11 (Excel-PowerPoint) (Excel-Video)
  - PERT chart (Excel-Video)
  - ***GANTT chart is a bar chart and PERT chart is a flow chart***.
- User specifies
  - Scope and objectives
  - Major tasks and phases
  - Task dependencies
  - Resources, including personnel
  - Timelines
- Software can
  - Generate calendars
  - Produce graphs with different views of project
  - Provide means of communication for staff
  - Ex: Microsoft Project, ProjectLibre (Tutorial)

ID	Task Name	Resource Names	Start	Finish	Duration
1	**Define specifications**		**1/5/2004**	**1/8/2004**	**4d**
2	Interview users	Adam	1/5/2004	1/6/2004	2d
3	Identify transactions, reports needed	Adam, Beth	1/7/2004	1/7/2004	1d
4	Begin data dict, x-ref table	Beth,Colin	1/8/2004	1/8/2004	1d
5	**Create E-R diagram**		**1/9/2004**	**1/13/2004**	**3d**
6	Identify entities,attributes,relationships	Adam	1/9/2004	1/9/2004	4h
7	Identify cardinality,participation constraints	Adam, Beth	1/12/2004	1/12/2004	4h
8	Draw diagram	Adam, Colin	1/13/2004	1/13/2004	1d
9	Map E-R to relational model	Beth	1/14/2004	1/14/2004	1d
10	Normalize relational model	Adam	1/15/2004	1/15/2004	1d
11	Create relational database	Colin	1/16/2004	1/19/2004	2d
12	Create EER diagram	Adam, Beth, Colin	1/19/2004	1/20/2004	2d
13	Map EER model to object-relational model	Adam	1/21/2004	1/21/2004	1d
14	Create object-relational database	Beth	1/22/2004	1/23/2004	2d
15	Create UML diagram	Beth	1/23/2004	1/26/2004	2d
16	Map UML diagram to object-oriented model	Colin	1/27/2004	1/27/2004	1d
17	Create object-oriented database	Adam	1/28/2004	1/29/2004	2d

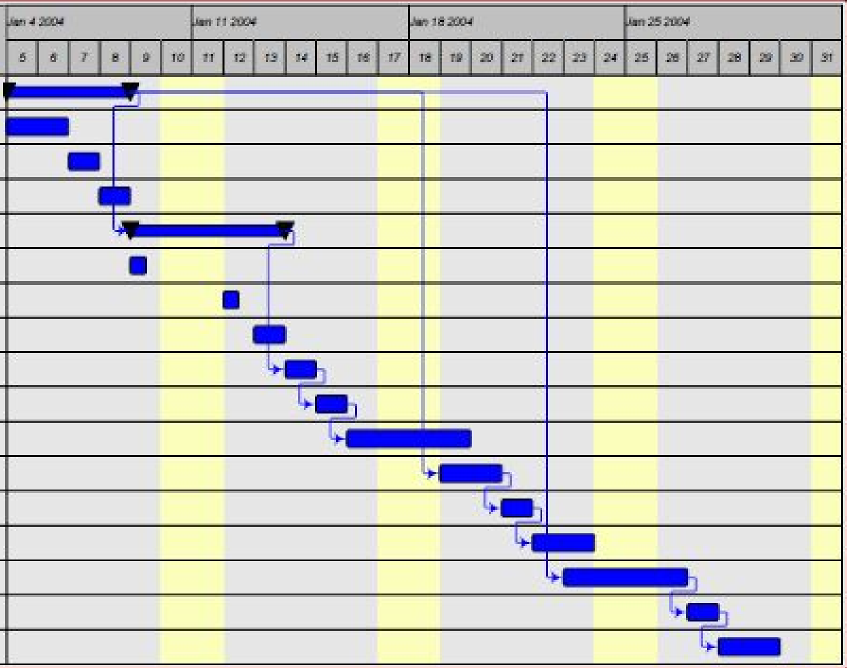
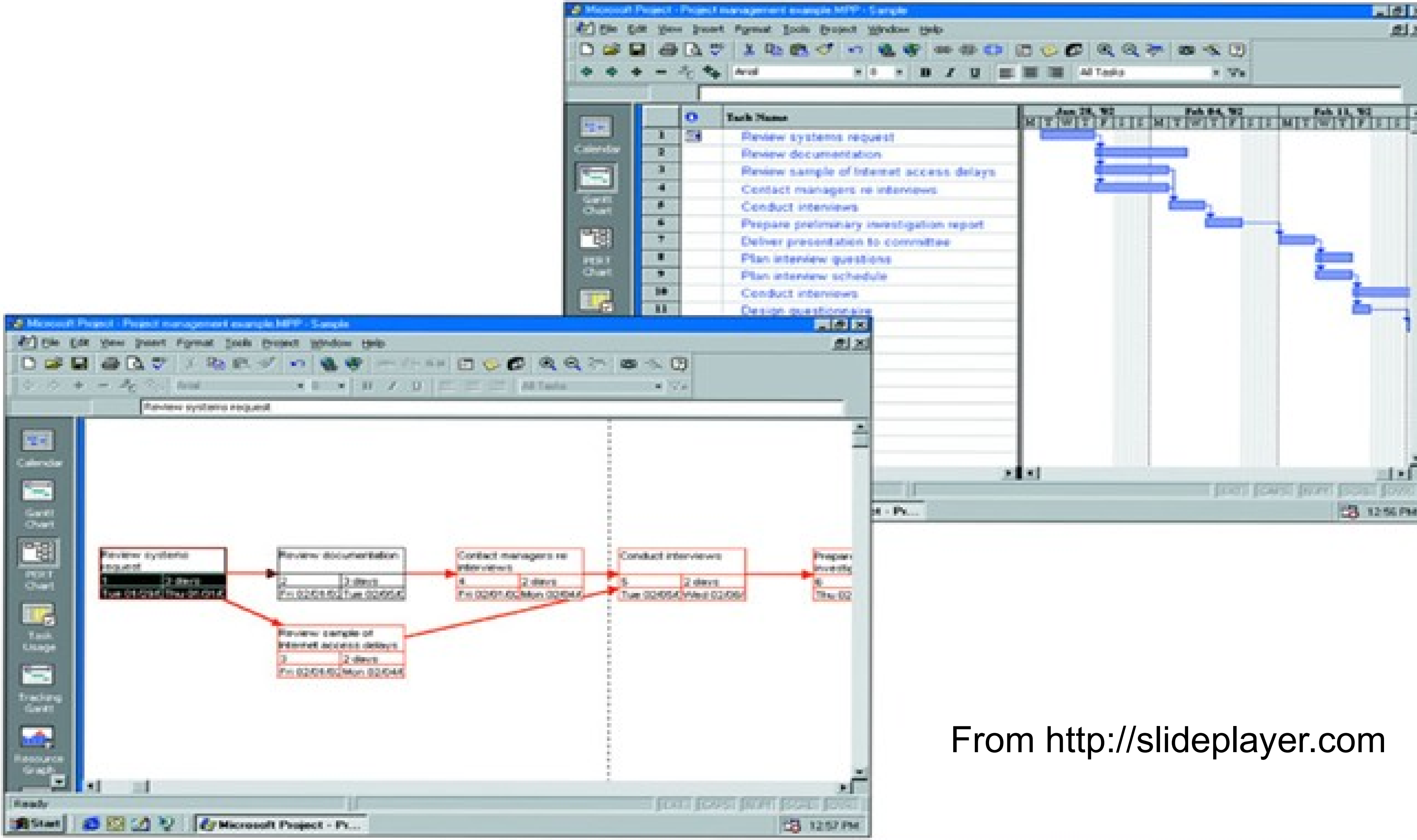


Figure 2.11: Gantt Chart for The Art Gallery Database Project



From http://slideplayer.com

FIGURE TK 3-12 The Microsoft Project screen at the top of the figure shows an example of a systems development project in the form of a GANTT chart. The same information also is displayed in the form of a PERT/CPM chart, as shown in the bottom screen.

Database Administrator Skills

- DBA must be
  - Technically competent
  - Good manager
  - Have excellent interpersonal and communication skills
- Has primary responsibility for
  - planning,
  - designing, developing and
  - managing the operating database
- Database designer may do conceptual and logical design; DBA does physical design, implementation, develops, manages system

Planning and Design Stage

- Preliminary planning
- Identifying user requirements
- Developing and maintaining the data dictionary
- Designing the conceptual model- may use E-R or UML diagram
- Choosing a DBMS
- Developing the logical model-writes schema
- Developing the physical model

Development Phase

- Creating and loading the database using DDL
- Developing user views
- Writing and maintaining documentation
- Developing and enforcing data standards
- Developing and enforcing application program standards
- Developing operating procedures
- Doing user training

Database Management Phase

- Ensuring database security
- Monitoring performance
- Tuning and reorganizing
- Keeping current on database improvements

Three-level Database Architecture

- CODASYL DBTG and ANSI/X3/SPARC - database architecture at 3 levels of abstraction – **external, logical, internal** - written description called a **schema**
- Rationale for separation of external and internal levels
  - Different users need different views of same data
  - Users data needs may change over time
  - Hides complexity of database storage structures
  - Can change logical structure without affecting all users
  - Can change data and file structures without affecting overall logical structure or users' views
  - Database structure unaffected by changes to the physical aspects of storage

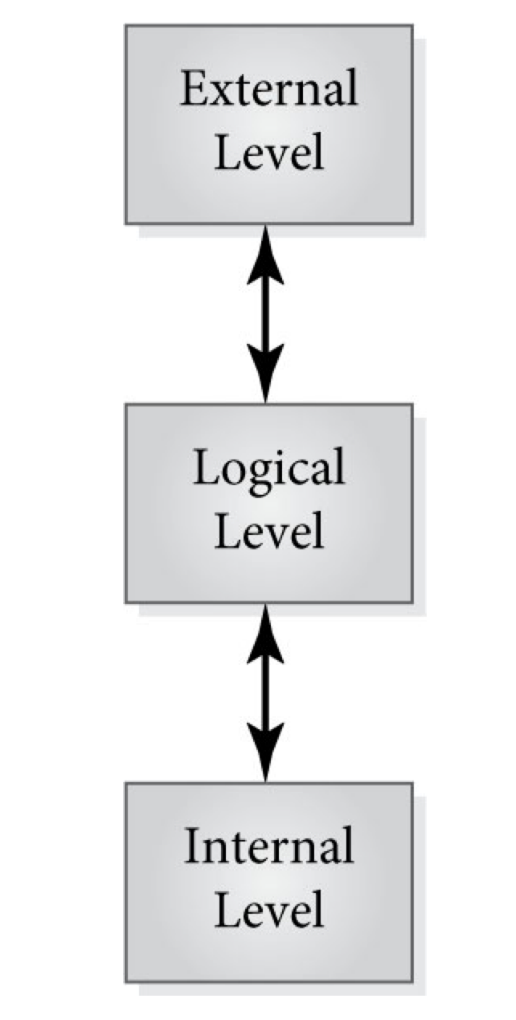
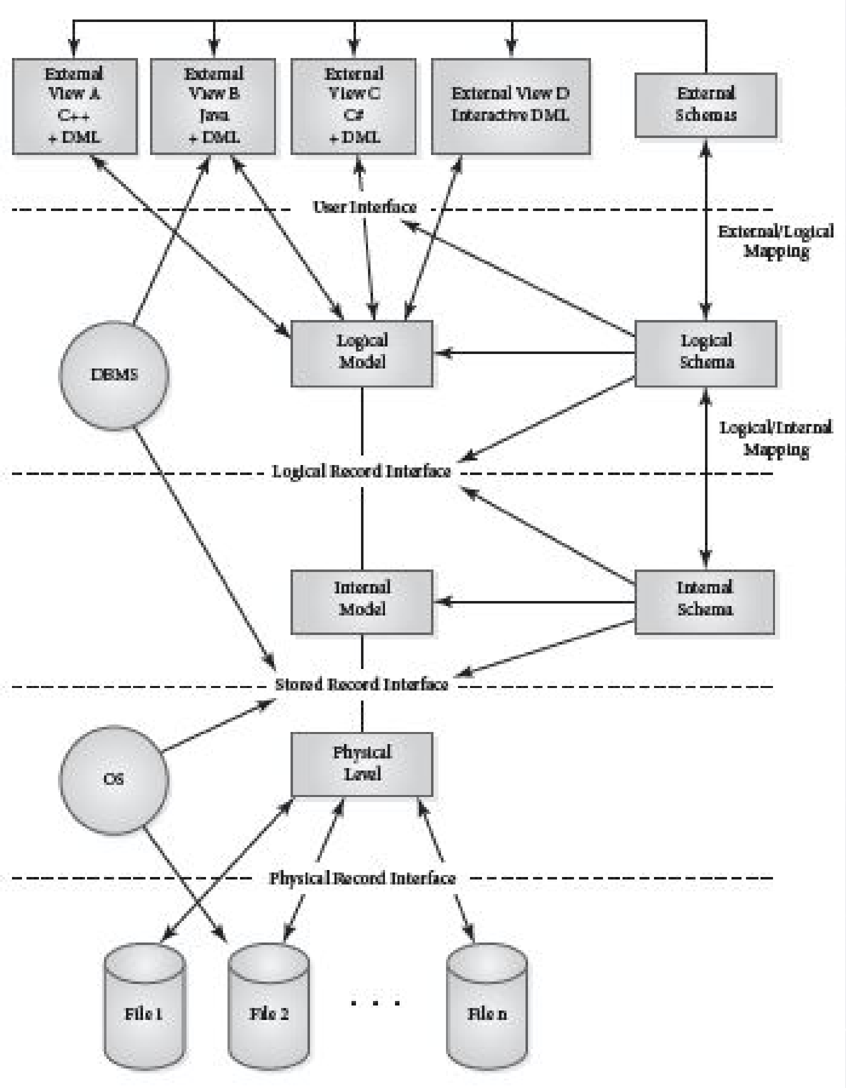


Figure 2.4 Three-Level Database Architecture



External Level

- Consists of many user models or **views**
- Has external records - records seen by users
- May include calculated or virtual data
- Described in external schemas (sub-schemas)
- Used to create user interface

Logical Level

- Entire information structure of database
- “community view” as seen by DBA
- Collection of logical records
- Derived from conceptual model
- All entities, attributes, relationships represented
- Includes all record types, data item types, relationships, constraints, semantic information, security and integrity information
- Relatively constant over time
- Described in logical schema
- Used to create logical record interface

Internal Level

- Implementation level
- Includes data structures, file organizations used by DBMS
- Depends on DBMS used
- Described in internal schema
- Used to create stored record interface with operating system
- Operating system creates physical files and physical record interface, below DB

FIGURE 2.6
Differences in External, Logical, Stored, and Physical Records

External Employee Record:

employeeName	empNumber	dept
JACK JONES	E101	Marketing

Logical Employee Record:

empId	lastName	firstName	dept	salary
E101	Jones	Jack	12	55000

Stored Employee Record:

empId	lastName	firstName	dept	salary	forward pointer	backward pointer

E101bbbbbbJonesbbbbbbJackbbbbbb12bbbbbb55000bbbb10101bbbbbb10001

Physical Record:

Block header	rec of E90	rec of E95	rec of E101	rec of E125

FIGURE 2.5
Retrieving Record of E101 for User A

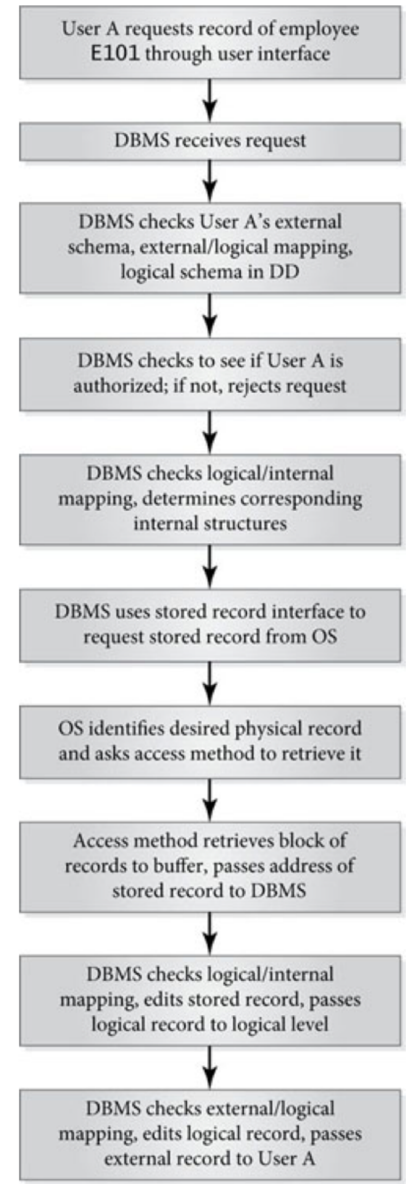
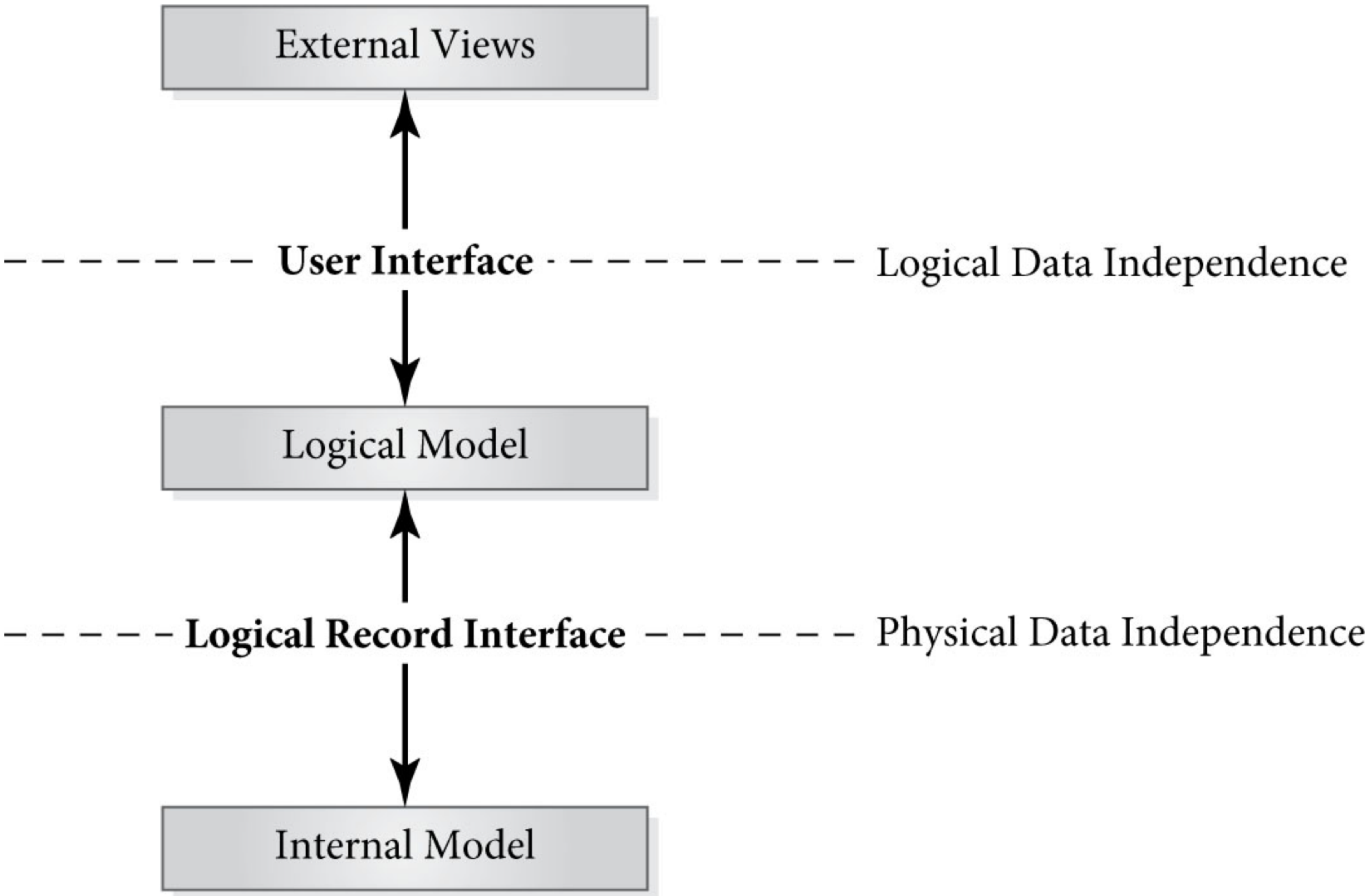


FIGURE 2.6
Differences in External, Logical, Stored, and Physical Records

Data Independence

- Logical data independence
  - Immunity of external models to changes in the logical model
  - Occurs at user interface level
- Physical data independence
  - Immunity of logical model to changes in internal model
  - Occurs at logical interface level

FIGURE 2.7
Logical and Physical Data Independence



Data Models

- Collection of tools for describing structure of database
- Often includes a type of diagram and specialized vocabulary
- Description of the data, relationships in data, constraints on data, some data meanings
- Most permanent part in database architecture
- Corresponds to conceptual level or logical level
- **Intension** or scheme of the database
- May change with schema evolution

Entity-Relationship Model

- A semantic model, captures meanings
- Conceptual level model
- Proposed by Peter Chen in 1970s
- **Entities**: real-world objects about which we collect data
- **Attributes**: describe the entities
- **Relationships**: associations among entities
- **Entity set**: set of entities of the same type
- **Relationship set**: set of relationships of same type
- Relationships sets may have descriptive attributes
- Represented by E-R diagrams

FIGURE 2.8
Basic Symbols for E-R Diagrams

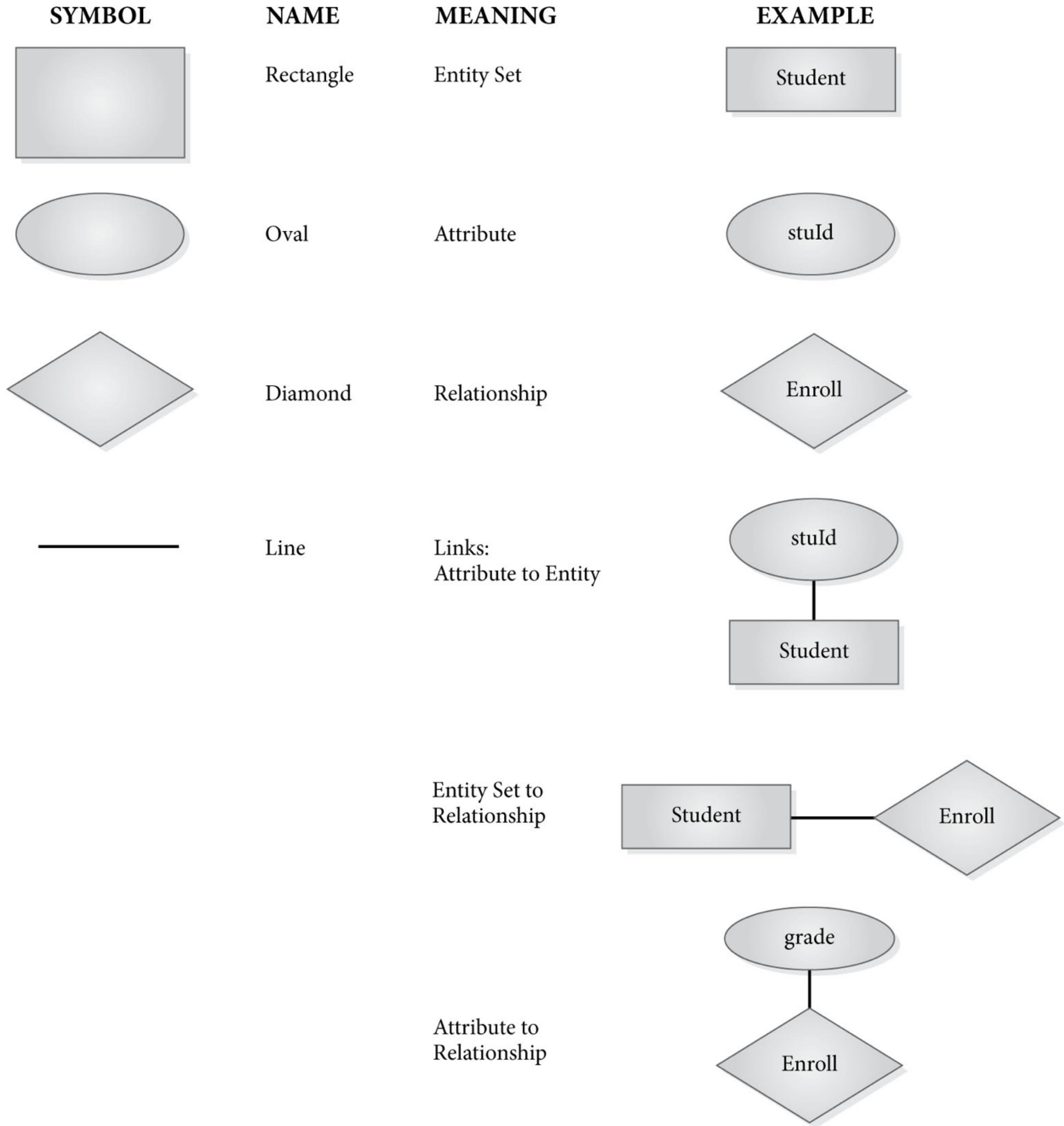
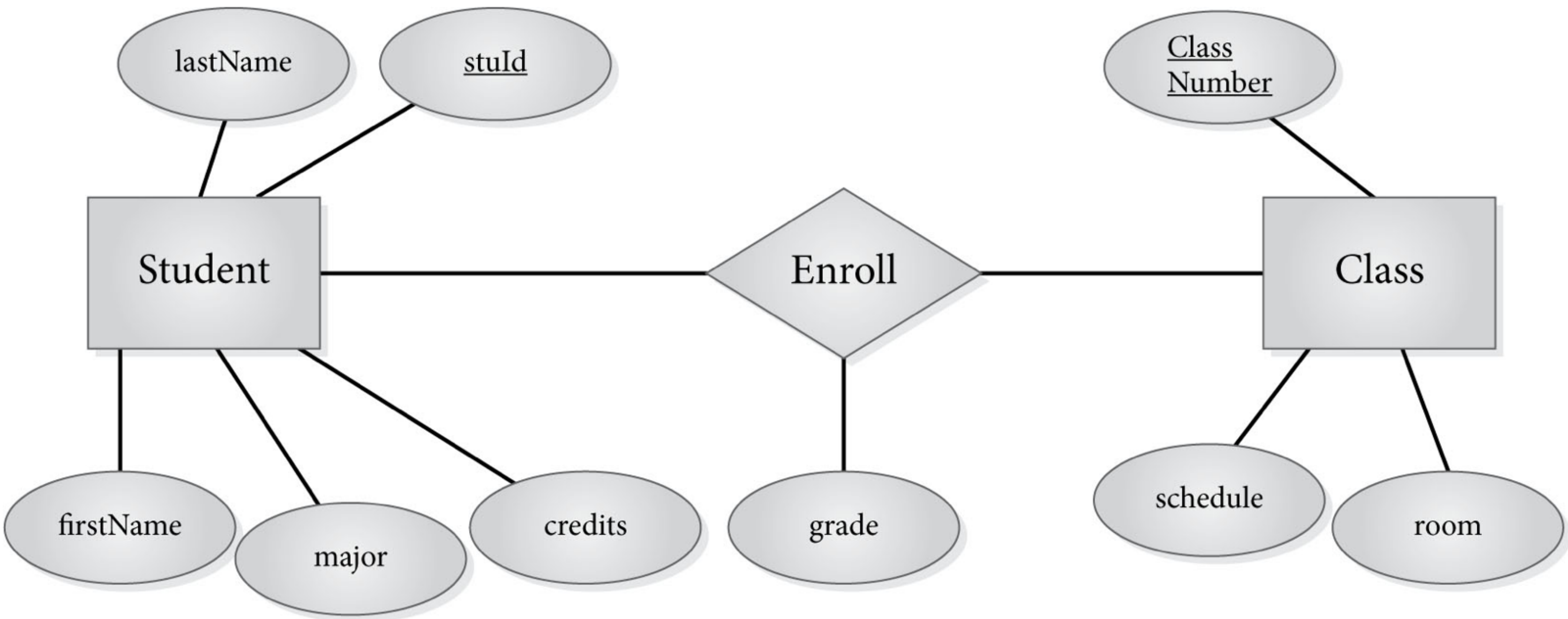


FIGURE 2.9

Simplified E-R Diagram



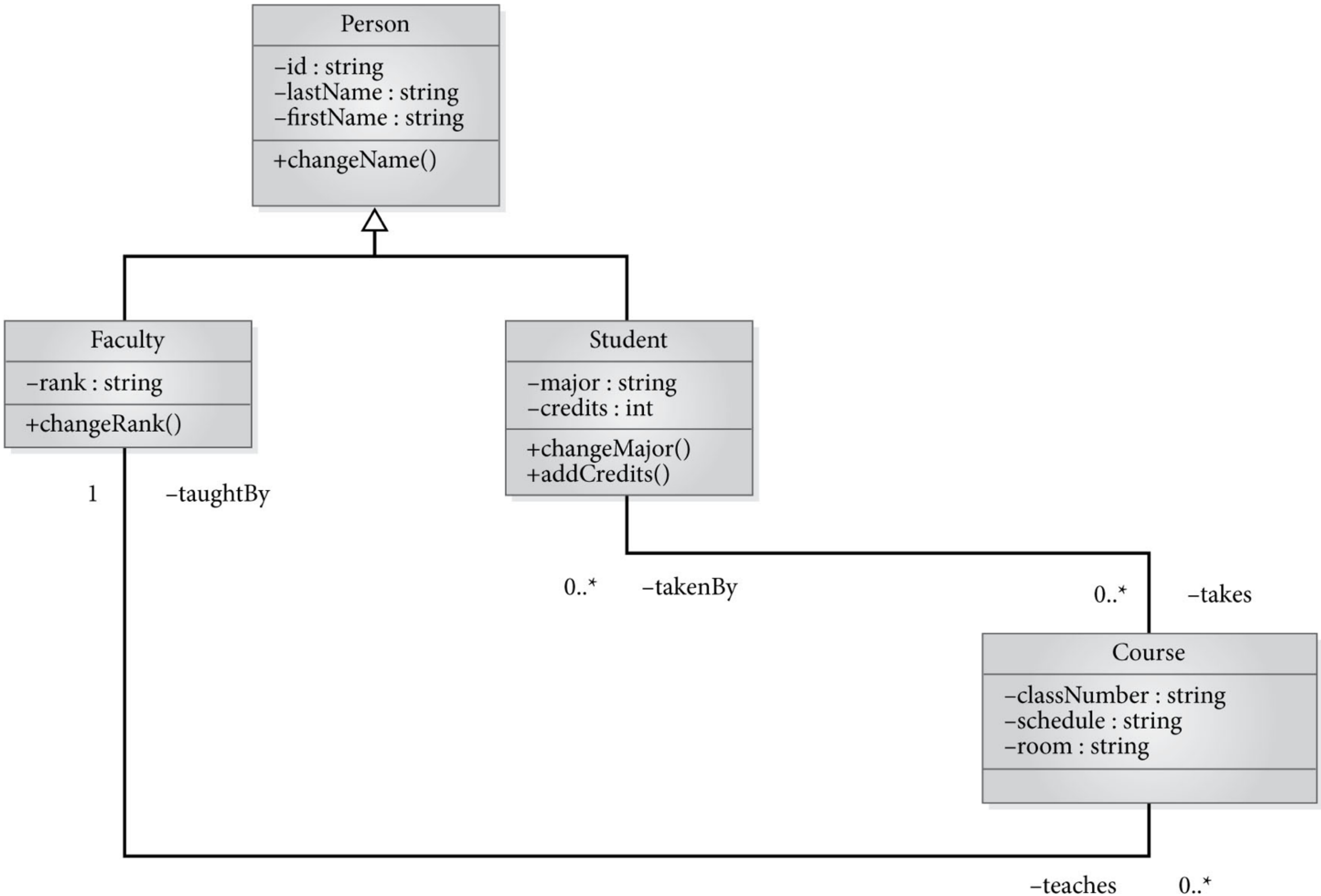
Relational Model (Relational DB)

- Record-based model
- Logical-level model
- Proposed by E.F. Codd
- Based on mathematical relations
- Uses **relations**, represented as tables
- Columns of tables represent attributes
- Tables represent relationships as well as entities
- Successor to earlier record-based models—**network** (Network DB) and **hierarchical** (Hiararchical DB)

Object-oriented Model

- Includes **encapsulation**, **inheritance**
- Objects have both **state**(attributes) and **behavior**(methods)
- Designer defines **classes** with attributes, methods, and relationships
- Class constructor method creates object instances
- Each object has a unique object ID
- Classes can be grouped into class hierarchies
- Database objects have **persistence**
- Both conceptual-level and logical-level model
- UML class diagrams often used

FIGURE 2.10
A Class Diagram



Object-relational model

(Object-Relational DB)

- Adds to relational model
  - new complex datatypes
  - objects with attributes and methods
  - inheritance
- SQL extended to handle objects

Large-Scale Data Models

(Data Warehouse) (Data Warehouse) (Data Warehouse-Full)

- star schema (Youtube)
  - Central fact table-observed data
  - Dimension tables-data about attributes
- data cube
  - Multidimensional arrays, like stacked tables
  - Supports pivoting, rollup, drilldown of data
- semi-structured
  - Collection of nodes, each with data, with different schemas
  - Node contains a description of its own contents
- key-value pairs (Youtube)
  - Schema-less; uses associative array of <key,value> pairs
- big table systems (Youtube) (Youtube)
  - Map indexed by a row key, column key, and timestamp
- graph-oriented systems (Youtube)
  - Graph consists of nodes, properties, and edges