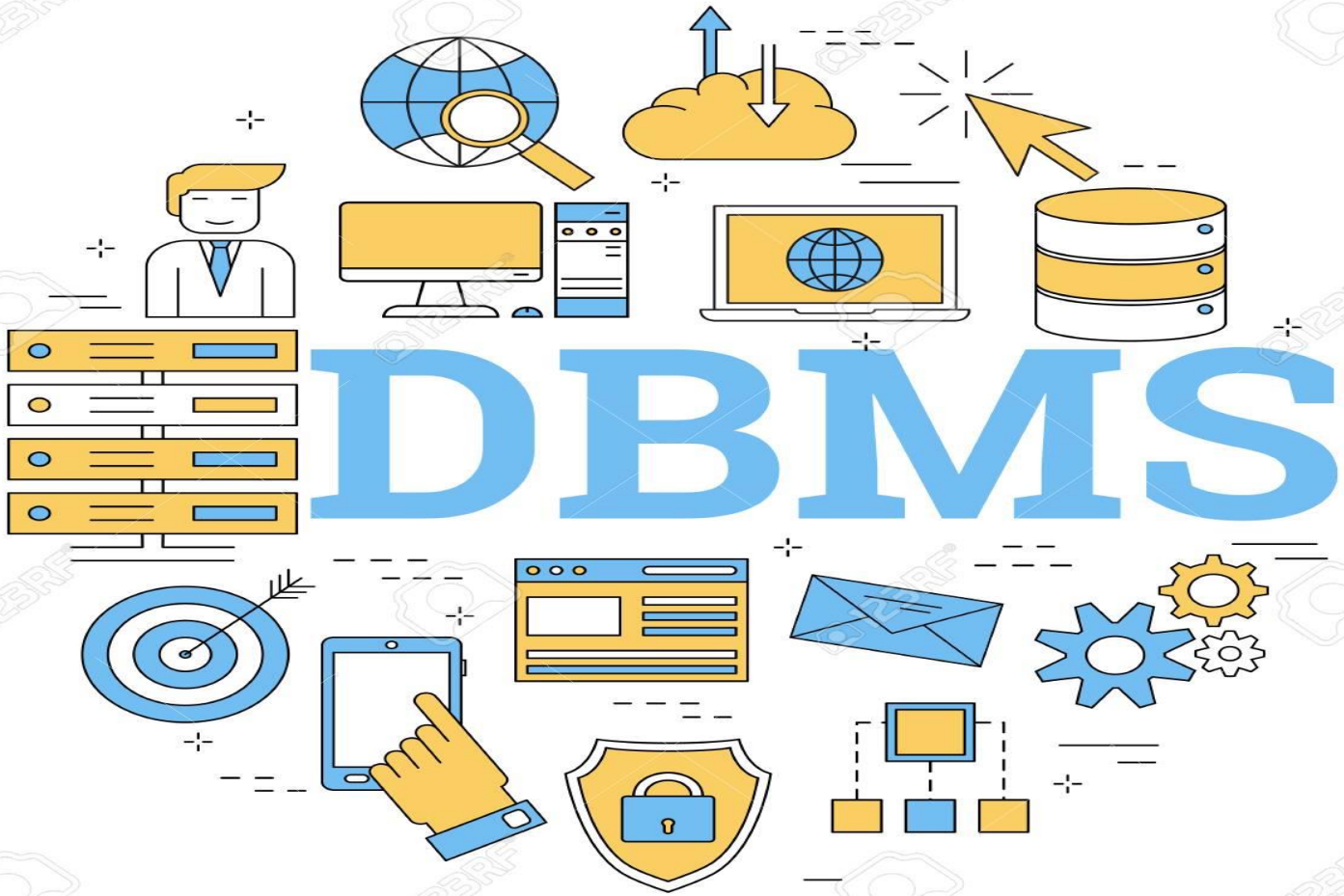

DBMS

Text Books:

Abraham Silberschatz, Henry F. Korth and S. Sudarshan, “Database System Concepts”, McGraw Hill Education, 6th Edition,2013

Reference Books:

R. Elmasri and S. Navathe, “Fundamentals of Database Systems.”, Pearson Education, 7th Edition, 2017

CHAPTER 1

The Need for Databases

Data Models

DATABASE MANAGEMENT SYSTEM (DBMS)

DBMS contains information about a particular enterprise

- Collection of interrelated data
- Set of programs to access the data
- An environment that is both *convenient* and *efficient* to use

Database Applications:

- Banking: transactions
- Airlines: reservations, schedules
- Universities: 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

Databases can be very large.

Databases touch all aspects of our lives

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

DRAWBACKS OF USING FILE SYSTEMS TO STORE DATA

Data redundancy and inconsistency

- Multiple file formats, duplication of information in different files

Difficulty in accessing data

- Need to write a new program to carry out each new task

Data isolation

- Multiple files and formats

Integrity problems

- Integrity constraints (e.g., account balance > 0) become "buried" in program code rather than being stated explicitly
- Hard to add new constraints or change existing ones

DRAWBACKS OF USING FILE SYSTEMS TO STORE DATA (CONT.)

Atomicity of updates

- Failures may leave database in an inconsistent state with partial updates carried out
- Example: Transfer of funds from one account to another should either complete or not happen at all

Concurrent access by multiple users

- Concurrent access needed for performance
- Uncontrolled concurrent accesses can lead to inconsistencies
  - Example: Two people reading a balance (say 100) and updating it by withdrawing money (say 50 each) at the same time

Security problems

- Hard to provide user access to some, but not all, data

Database systems offer solutions to all the above problems

DATA MODELS

Hierarchical

Network

Relational

Entity relationship

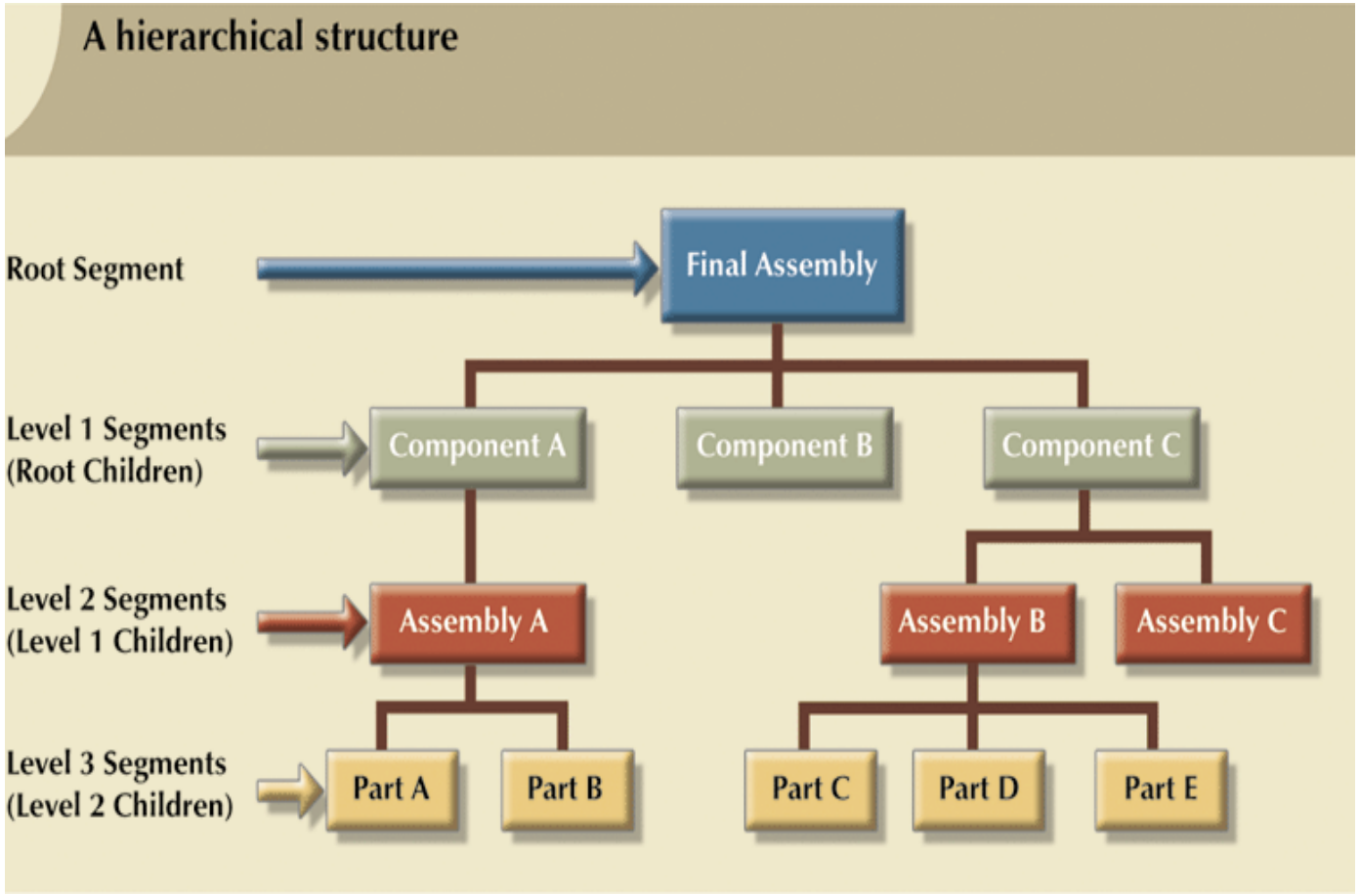
Object oriented (OO)

THE HIERARCHICAL MODEL

Developed in the 1960s to manage large amounts of data for complex manufacturing projects

Basic logical structure is represented by an upside-down “tree”

THE HIERARCHICAL MODEL (CONTINUED)



THE HIERARCHICAL MODEL (CONTINUED)

The hierarchical structure contains levels, or segments

Depicts a set of one-to-many (1:M) relationships between a parent and its children segments

- Each parent can have many children
- Each child has only one parent

THE HIERARCHICAL MODEL (CONTINUED)

Advantages

- Many of the hierarchical data model's features formed the foundation for current data models
- Its database application advantages are replicated, albeit in a different form, in current database environments
- Generated a large installed (mainframe) base, created a pool of programmers who developed numerous tried-and-true business applications

THE HIERARCHICAL MODEL (CONTINUED)

Disadvantages

- Complex to implement
- Difficult to manage
- Implementation limitations
- Lack of standards

THE NETWORK MODEL

- It was in 1971 that the Conference on Data System Languages or CODASYL officially or formally defined the Network model. The network databases arrange its data as a directed graph

Created to

- Represent complex data relationships more effectively
- Improve database performance
- Impose a database standard

THE NETWORK MODEL (CONTINUED)

Schema Data Definition Language (DDL)

- Enables database administrator to define schema components

Subschema DDL

- Allows application programs to define database components that will be used

DML

- Works with the data in the database

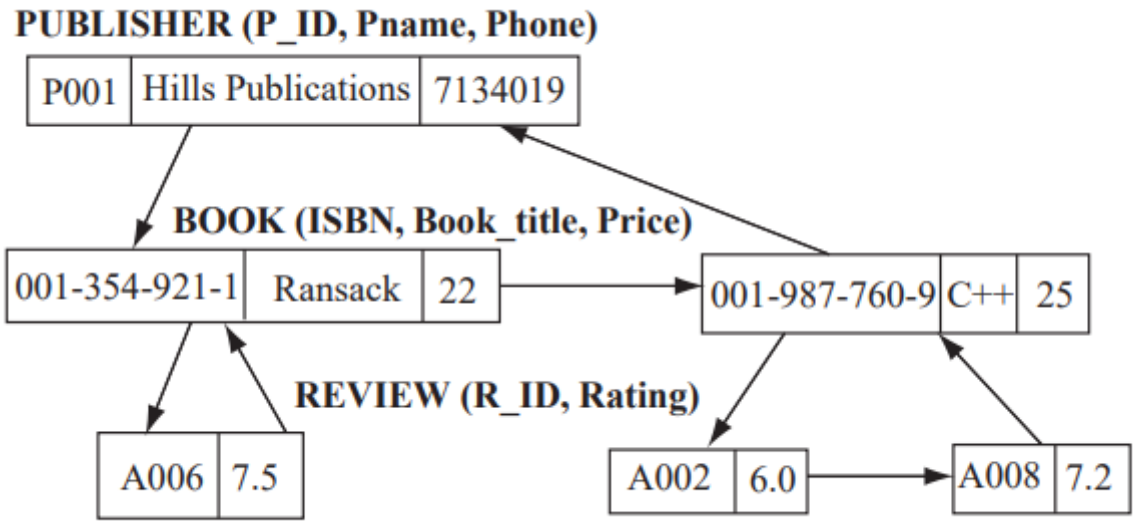
THE NETWORK MODEL (CONTINUED)

Resembles hierarchical model

The main **advantage** of network data model is that a parent node can have many child nodes and a child can also have many parent nodes. Thus, the network model permits the modeling of manyto-many relationships in data.

The main **limitation** of the network data model is that it can be quite complicated to maintain all the links and a single broken link can lead to problems in the database.

THE NETWORK MODEL (CONTINUED)



THE NETWORK MODEL (CONTINUED)

Advantages

Multi-parent support.

Somewhat same simplicity as the hierarchical model.

More useful than the hierarchical data model.

Deals with even larger amounts of information than the hierarchical model.

Promotes data integrity.

Data independence.

Improved data access.

THE NETWORK MODEL (CONTINUED)

Disadvantages

- Too cumbersome
- Data relationships must be predefined.
- Much more complex than the hierarchical data model.
- Users are still required to know the physical representation of the database
- Information can be related in various and complicated ways.

THE RELATIONAL MODEL

Developed by Codd (IBM) in 1970

Conceptually simple

Computers lacked power to implement the relational model

Today, microcomputers can run sophisticated relational database software

THE RELATIONAL MODEL (CONTINUED)

Relational Database Management System (RDBMS)

Performs same basic functions provided by hierarchical and network DBMS systems, in addition to a host of other functions

Most important advantage of the RDBMS is its ability to hide the complexities of the relational model from the user

THE RELATIONAL MODEL (CONTINUED)

Table (relations)

- Matrix consisting of a series of row/column intersections
- Related to each other through sharing a common entity characteristic

Relational diagram

- Representation of relational database's entities, attributes within those entities, and relationships between those entities

THE RELATIONAL MODEL (CONTINUED)

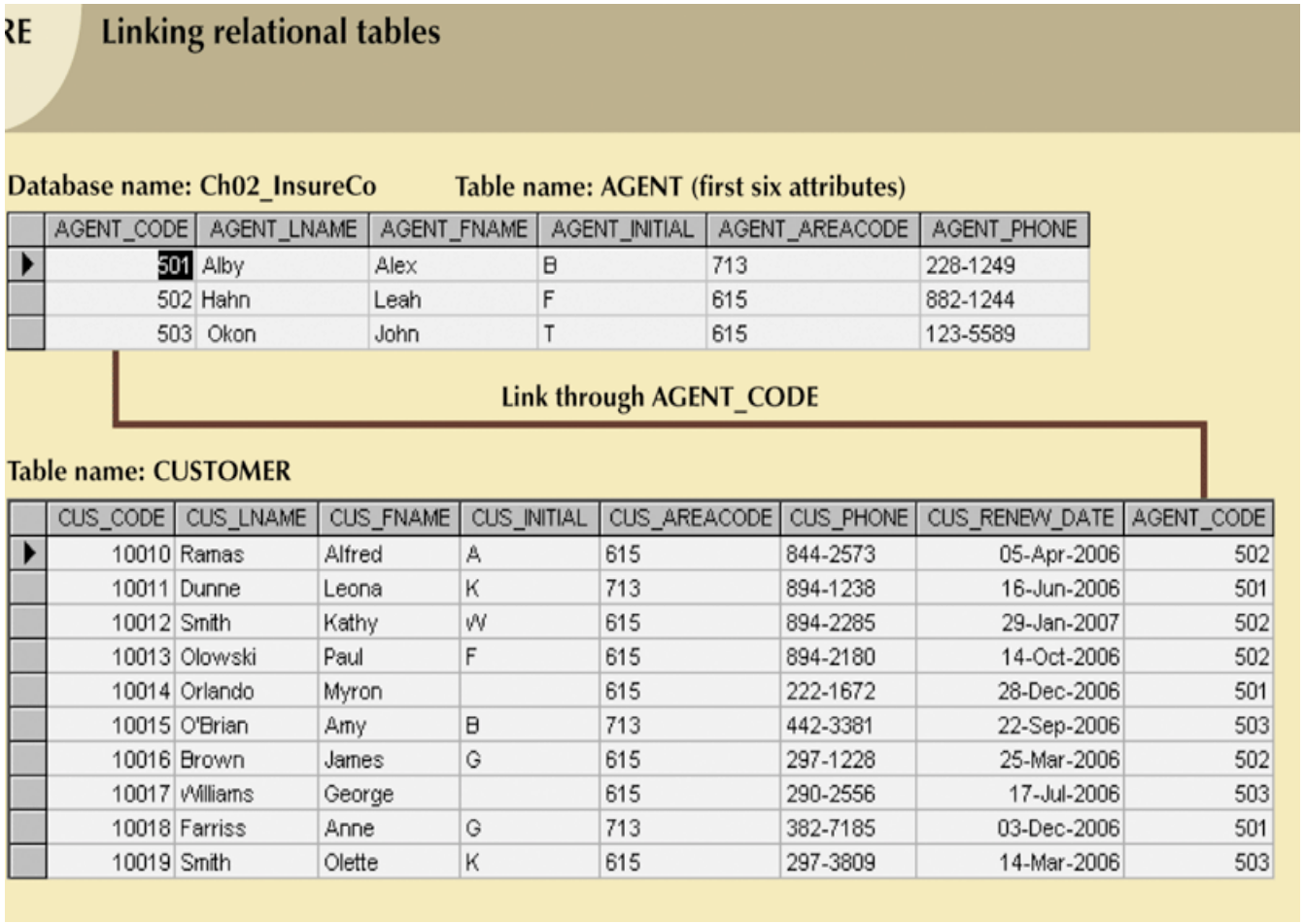
Relational Table

- Stores a collection of related entities
  - Resembles a file

Relational table is purely logical structure

- How data are physically stored in the database is of no concern to the user or the designer
- This property became the source of a real database revolution

THE RELATIONAL MODEL (CONTINUED)



RE Linking relational tables

Database name: Ch02_InsureCo Table name: AGENT (first six attributes)

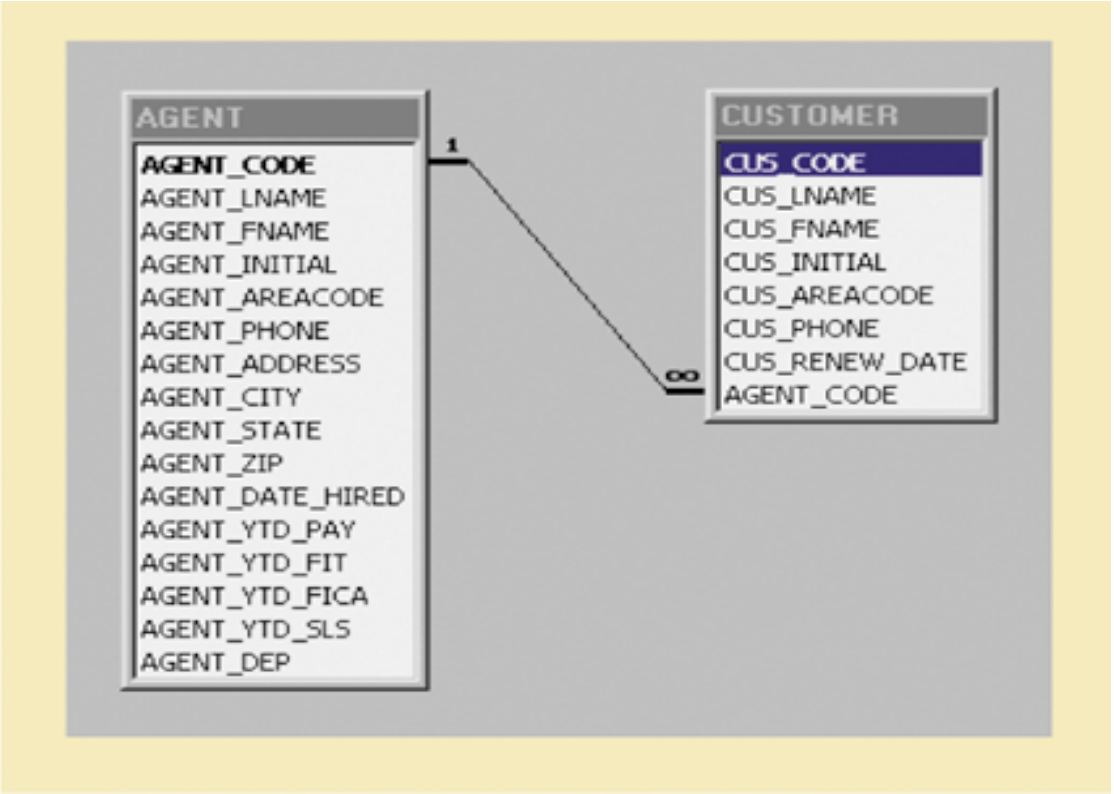
AGENT_CODE	AGENT_LNAME	AGENT_FNAME	AGENT_INITIAL	AGENT_AREACODE	AGENT_PHONE
501	Alby	Alex	B	713	228-1249
502	Hahn	Leah	F	615	882-1244
503	Okon	John	T	615	123-5589

Link through AGENT_CODE

Table name: CUSTOMER

CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_AREACODE	CUS_PHONE	CUS_RENEW_DATE	AGENT_CODE
10010	Ramas	Alfred	A	615	844-2573	05-Apr-2006	502
10011	Dunne	Leona	K	713	894-1238	16-Jun-2006	501
10012	Smith	Kathy	W	615	894-2285	29-Jan-2007	502
10013	Olowski	Paul	F	615	894-2180	14-Oct-2006	502
10014	Orlando	Myron		615	222-1672	28-Dec-2006	501
10015	O'Brian	Amy	B	713	442-3381	22-Sep-2006	503
10016	Brown	James	G	615	297-1228	25-Mar-2006	502
10017	Williams	George		615	290-2556	17-Jul-2006	503
10018	Farriss	Anne	G	713	382-7185	03-Dec-2006	501
10019	Smith	Olette	K	615	297-3809	14-Mar-2006	503

THE RELATIONAL MODEL (CONTINUED)



THE RELATIONAL MODEL (CONTINUED)

Rise to dominance due in part to its powerful and flexible query language

Structured Query Language (SQL) allows the user to specify what must be done without specifying how it must be done

SQL-based relational database application involves:

- User interface
- A set of tables stored in the database

RELATIONAL MODEL

All the data is stored in various tables.

Example of tabu

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

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

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

THE ENTITY RELATIONSHIP MODEL

Widely accepted and adapted **graphical tool** for data modeling

Introduced by Chen in 1976

Graphical representation of entities and their relationships in a database structure

THE ENTITY RELATIONSHIP MODEL (CONTINUED)

Entity relationship diagram (ERD)

- Uses graphic representations to model database components
- Entity is mapped to a relational table

Entity instance (or occurrence) is row in table

Entity set is collection of like entities

Connectivity labels types of relationships

- Diamond connected to related entities through a relationship line

THE ENTITY RELATIONSHIP MODEL (CONTINUED)

A One-to-Many (1:M) Relationship: a PAINTER can paint many PAINTINGs;
each PAINTING is painted by one PAINTER.

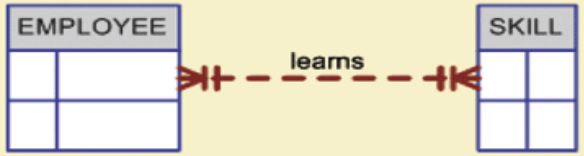
A Many-to-Many (M:N) Relationship: an EMPLOYEE can learn many SKILLs;
each SKILL can be learned by many EMPLOYEEs.

A One-to-One (1:1) Relationship: an EMPLOYEE manages one STORE;
each STORE is managed by one EMPLOYEE.

THE ENTITY RELATIONSHIP MODEL (CONTINUED)

A One-to-Many (1:M) Relationship: a PAINTER can paint many PAINTINGs;
each PAINTING is painted by one PAINTER.

A Many-to-Many (M:N) Relationship: an EMPLOYEE can learn many SKILLs;
each SKILL can be learned by many EMPLOYEEs.



A One-to-One (1:1) Relationship: an EMPLOYEE manages one STORE;
each STORE is managed by one EMPLOYEE.

THE OBJECT ORIENTED MODEL

Modeled both data and their relationships in a single structure known as an object

Object-oriented data model (OODM) is the basis for the object-oriented database management system (OODBMS)

OODM is said to be a semantic data model

THE OBJECT ORIENTED MODEL (CONTINUED)

Object described by its factual content

- Like relational model's entity

Includes information about relationships between facts within object, and relationships with other objects

- Unlike relational model's entity

Subsequent OODM development allowed an object to also contain all operations

Object becomes basic building block for autonomous structures

THE OBJECT ORIENTED MODEL (CONTINUED)

Object is an abstraction of a real-world entity

Attributes describe the properties of an object

Objects that share similar characteristics are grouped in classes

Classes are organized in a class hierarchy

Inheritance is the ability of an object within the class hierarchy to inherit the attributes and methods of classes above it

END