Designing Databases

13

Systems Analysis and Design, 7e
Kendall & Kendall

Learning Objectives

- Understand database concepts
- Use normalization to efficiently store data in a database
- Use databases for presenting data
- Understand the concept of data warehouses
- Comprehend the usefulness of publishing databases to the Web

Data Storage

- The data must be available when the user wants to use them
- The data must be accurate and consistent
- Efficient storage of data as well as efficient updating and retrieval
- It is necessary that information retrieval be purposeful

Data Storage (Continued)

- There are two approaches to the storage of data in a computer-based system:
  - Store the data in individual files, each unique to a particular application
  - Build a database
    - A database is a formally defined and centrally controlled store of data intended for use in many different applications

Major Topics

- Databases
- Normalization
- Key design
- Using the database
- Data warehouses
- Data mining

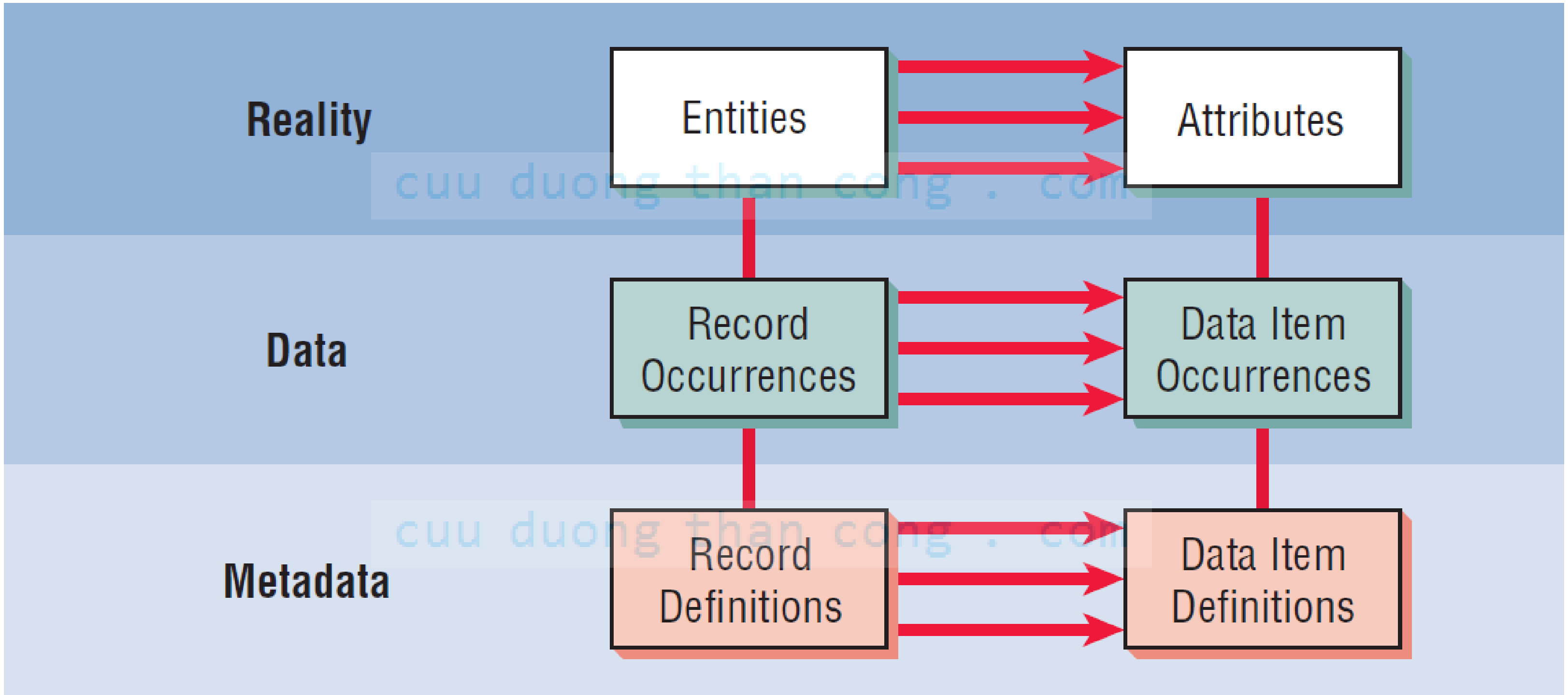
Databases

- Effectiveness objectives of the database:
  - Ensuring that data can be shared among users for a variety of applications
  - Maintaining data that are both accurate and consistent
  - Ensuring data required for current and future applications will be readily available
  - Allowing the database to evolve as the needs of the users grow
  - Allowing users to construct their personal view of the data without concern for the way the data are physically stored

Reality, Data, and Metadata

- Reality
  - The real world
- Data
  - Collected about people, places, or events in reality and eventually stored in a file or database
- Metadata
  - Information that describes data

Figure 13.1 Reality, data, and metadata



Entities

- Any object or event about which someone chooses to collect data
- May be a person, place or thing
- May be an event or unit of time

Entity Subtype

- An entity subtype is a special one-to-one relationship used to represent additional attributes, which may not be present on every record of the first entity
- This eliminates null fields stored on database tables
- For example, students who have internships. The STUDENT MASTER should not have to contain information about internships for each student

Relationships

- Relationships
  - One-to-one
  - One-to-many
  - Many-to-many
- A single vertical line represents one
- A crow's foot represents many

Figure 13.2 Entity-relationship (E-R) diagrams can show one-to-one, one-to-many, or many-to-many associations

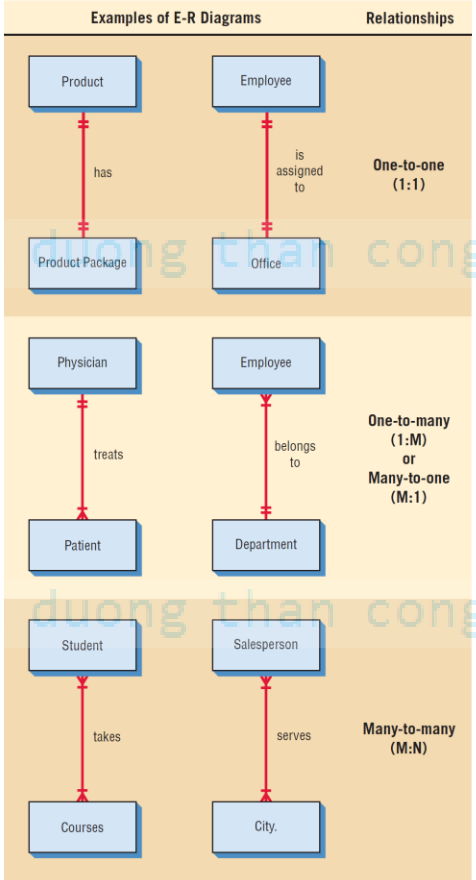


Figure 13.3 The entity-relationship symbols and their meanings

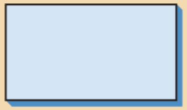
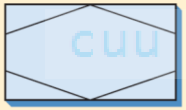
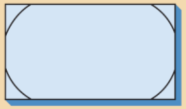
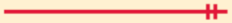
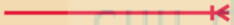
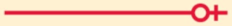
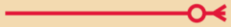
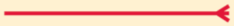
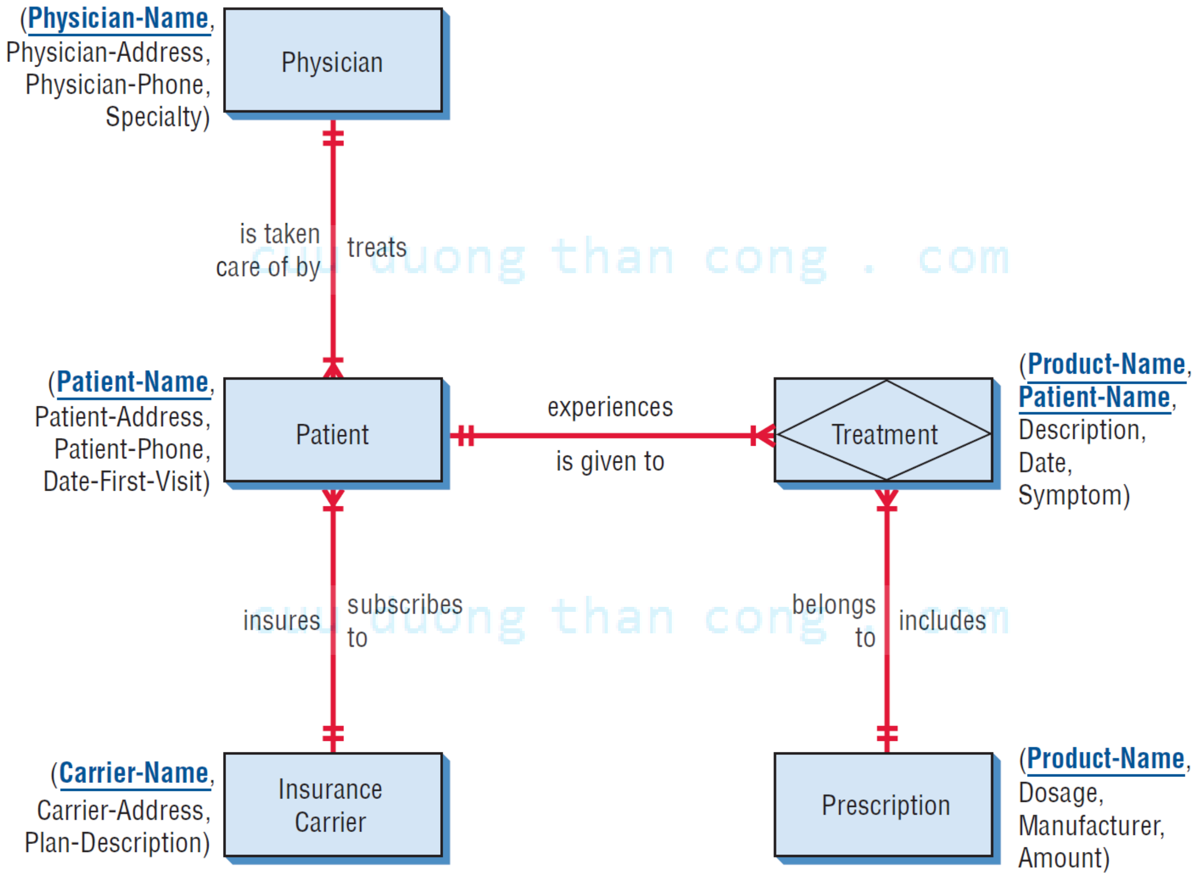
Symbol	Official Explanation	What It Really Means
	Entity	A class of persons, places, or things
	Associative entity	Used to join two entities
	Attributive entity	Used for repeating groups
	To 1 relationship	Exactly one
	To many relationship	One or more
	To 0 or 1 relationship	Only zero or one
	To 0 or more relationship	Can be zero, one, or more
	To more than 1 relationship	Greater than one

Figure 13.4 The entity-relationship diagram for patient treatment. Attributes can be listed alongside the entities. In each case, the key is underlined



Attributes, Records, and Keys

- Attributes represent some characteristic of an entity
- Records are a collection of data items that have something in common with the entity described
- Keys are data items in a record used to identify the record

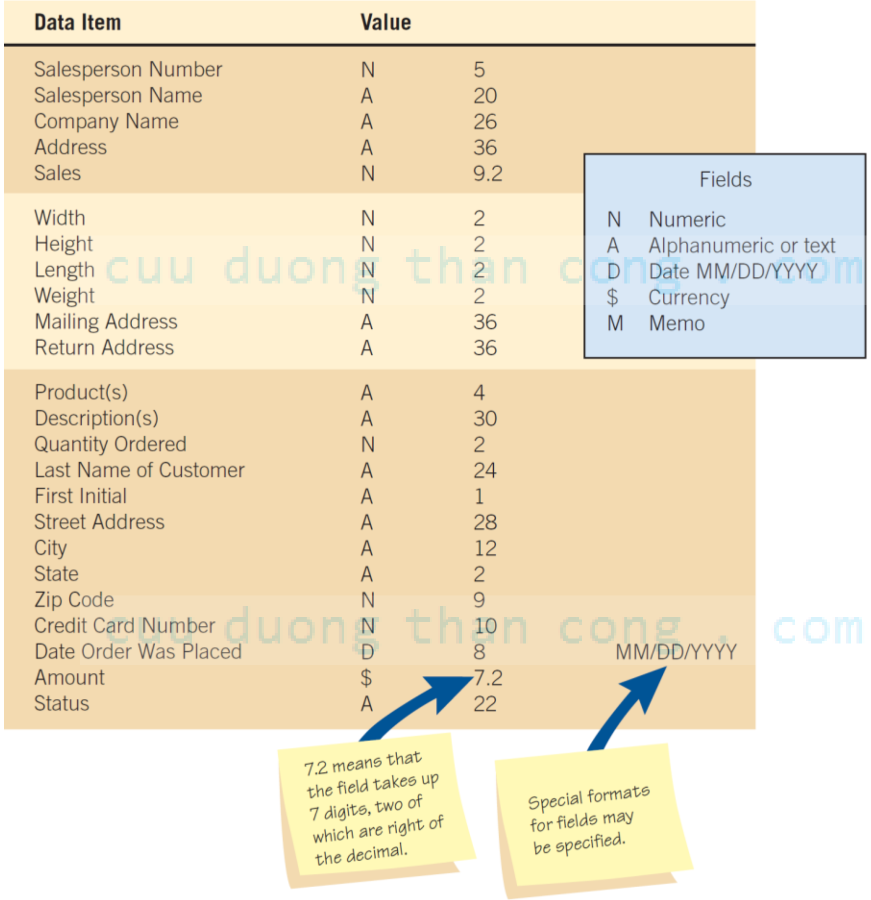
Key Types

- Key types are:
  - Primary key – unique attribute for the record
  - Candidate key – an attribute or collection of attributes, that can serve as a primary key
  - Secondary key, a key which may not be unique, used to select a group of records
  - Composite key, a combination of two or more attributes representing the key

Metadata

- Data about the data in the file or database
- Describe the name given and the length assigned each data item
- Also describe the length and composition of each of the records

Figure 13.7 Metadata includes a description of what the value of each data item looks like



Data Item	Value		
Salesperson Number	N	5	
Salesperson Name	A	20	
Company Name	A	26	
Address	A	36	
Sales	N	9.2	
Width	N	2	
Height	N	2	
Length	N	2	
Weight	N	2	
Mailing Address	A	36	
Return Address	A	36	
Product(s)	A	4	
Description(s)	A	30	
Quantity Ordered	N	2	
Last Name of Customer	A	24	
First Initial	A	1	
Street Address	A	28	
City	A	12	
State	A	2	
Zip Code	N	9	
Credit Card Number	N	10	
Date Order Was Placed	D	8	MM/DD/YYYY
Amount	$	7.2	
Status	A	22	

Fields

N	Numeric
A	Alphanumeric or text
D	Date MM/DD/YYYY
$	Currency
M	Memo

7.2 means that the field takes up 7 digits, two of which are right of the decimal.

Special formats for fields may be specified.

Files

- A file contains groups of records used to provide information for operations, planning, management, and decision making
- Files can be used for storing data for an indefinite period of time, or they can be used to store data temporarily for a specific purpose

File Types

- Master file
- Table file
- Transaction file
- Report file

Master and Table Files

- Master files
  - Contain records for a group of entities
  - Contain all information about a data entity
- Table files
  - Contains data used to calculate more data or performance measures
  - Usually read-only by a program

Transaction and Report Files

- Transaction records
  - Used to enter changes that update the master file and produce reports
- Report files
  - Used when it is necessary to print a report when no printer is available
  - Useful because users can take files to other computer systems and output to specialty devices

File Organization

- Sequential organization
- Linked lists
- Hashed file organization

Relational Databases

- A database is intended to be shared by many users
- There are three structures for storing database files:
  - Relational database structures
  - Hierarchical database structures
  - Network database structures

Figure 13.10 Database design includes synthesizing user reports, user views and logical and physical designs

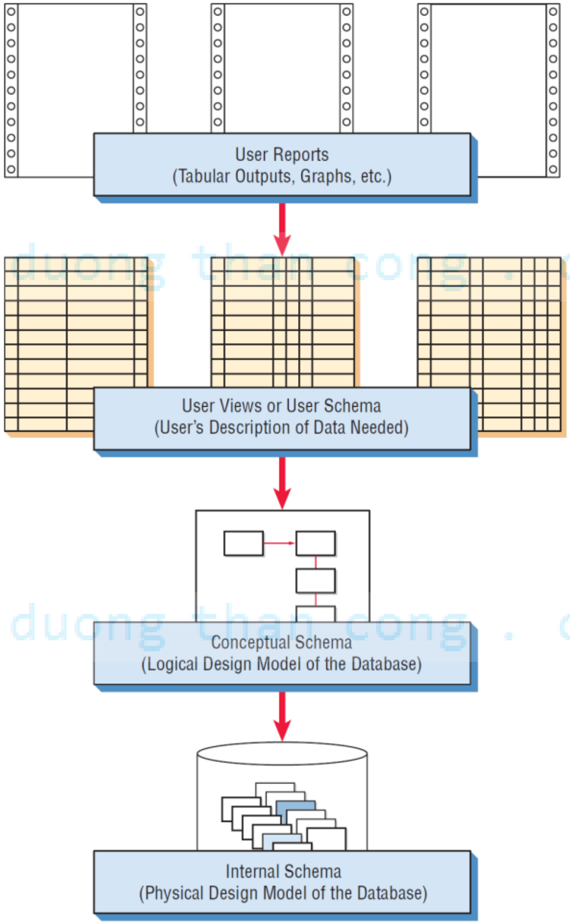
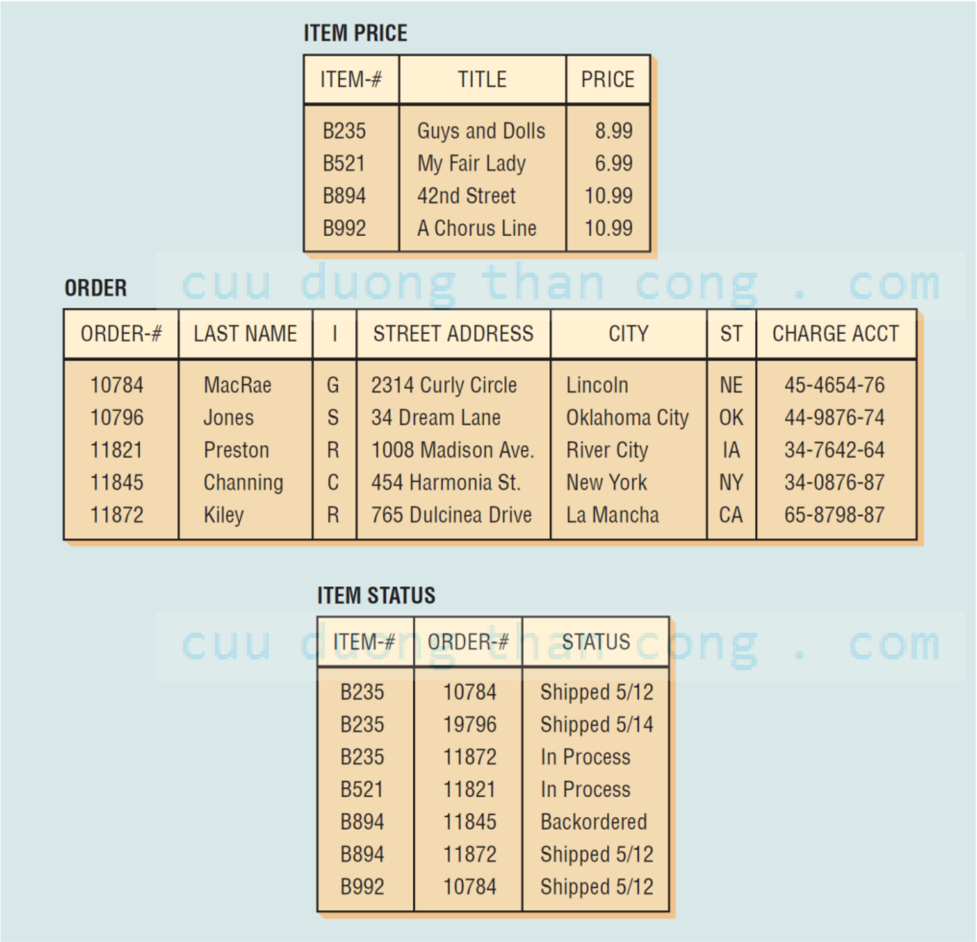

User Reports
(Tabular Outputs, Graphs, etc.)
User Views or User Schema
(User's Description of Data Needed)
Conceptual Schema
(Logical Design Model of the Database)
Internal Schema
(Physical Design Model of the Database)

Figure 13.11 In a relational data structure, data are stored in many tables



ITEM PRICE

ITEM-#	TITLE	PRICE
B235	Guys and Dolls	8.99
B521	My Fair Lady	6.99
B894	42nd Street	10.99
B992	A Chorus Line	10.99

ORDER

ORDER-#	LAST NAME	I	STREET ADDRESS	CITY	ST	CHARGE ACCT
10784	MacRae	G	2314 Curly Circle	Lincoln	NE	45-4654-76
10796	Jones	S	34 Dream Lane	Oklahoma City	OK	44-9876-74
11821	Preston	R	1008 Madison Ave.	River City	IA	34-7642-64
11845	Channing	C	454 Harmonia St.	New York	NY	34-0876-87
11872	Kiley	R	765 Dulcinea Drive	La Mancha	CA	65-8798-87

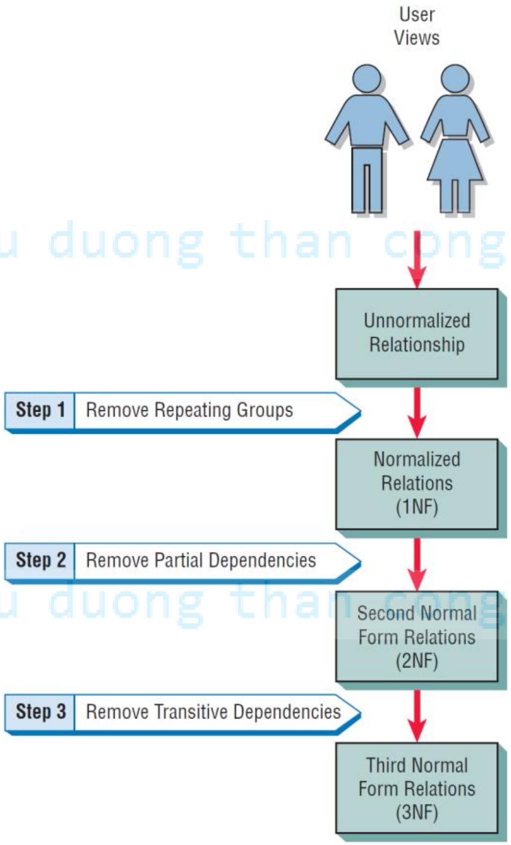
ITEM STATUS

ITEM-#	ORDER-#	STATUS
B235	10784	Shipped 5/12
B235	19796	Shipped 5/14
B235	11872	In Process
B521	11821	In Process
B894	11845	Backordered
B894	11872	Shipped 5/12
B992	10784	Shipped 5/12

Normalization

- Normalization is the transformation of complex user views and data stores to a set of smaller, stable, and easily maintainable data structures
- The main objective of the normalization process is to simplify all the complex data items that are often found in user views

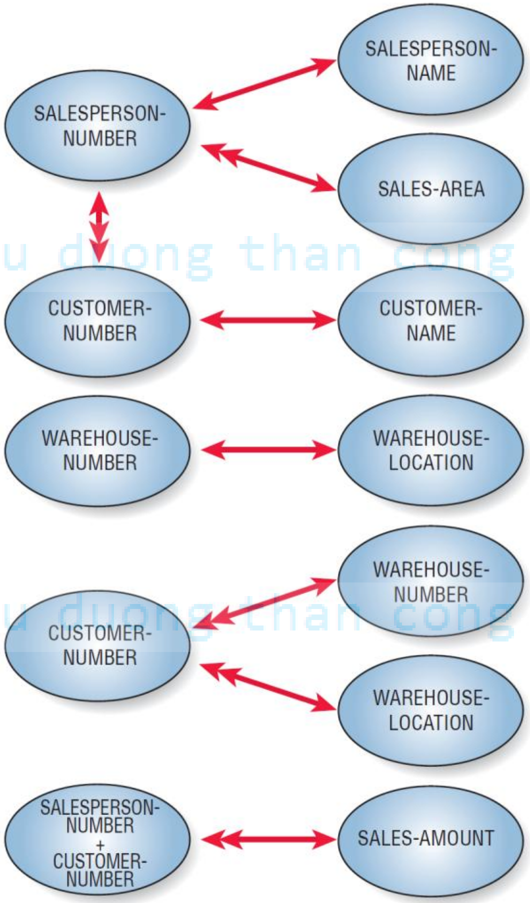
Figure 13.12 Normalization of a relation is accomplished in three major steps


User Views
Unnormalized Relationship
Step 1 Remove Repeating Groups
Normalized Relations (1NF)
Step 2 Remove Partial Dependencies
Second Normal Form Relations (2NF)
Step 3 Remove Transitive Dependencies
Third Normal Form Relations (3NF)

Data Model Diagrams

- Shows data associations of data elements
- Each entity is enclosed in an ellipse
- Arrows are used to show the relationships

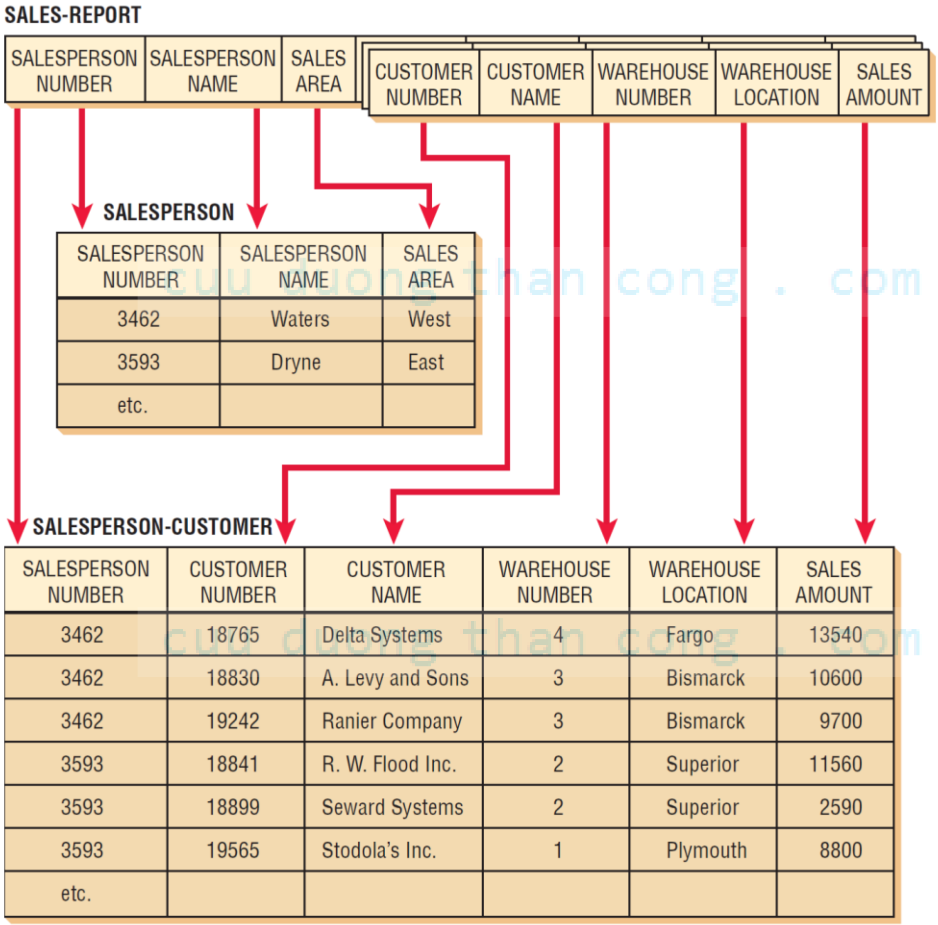
Figure 13.15 Drawing data model diagrams for data associations sometimes helps analysts appreciate the complexity of data storage



First Normal Form (1NF)

- Remove repeating groups
- The primary key with repeating group attributes are moved into a new table
- When a relation contains no repeating groups, it is in first normal form

Figure 13.18 The Original unnormalized relation SALES-REPORT is separated into two relations, SALESPERSON (3NF) and SALESPERSON-CUSTOMER (1NF)



SALES-REPORT

SALESPERSON NUMBER	SALESPERSON NAME	SALES AREA	CUSTOMER NUMBER	CUSTOMER NAME	WAREHOUSE NUMBER	WAREHOUSE LOCATION	SALES AMOUNT

SALESPERSON

SALESPERSON NUMBER	SALESPERSON NAME	SALES AREA
3462	Waters	West
3593	Dryne	East
etc.		

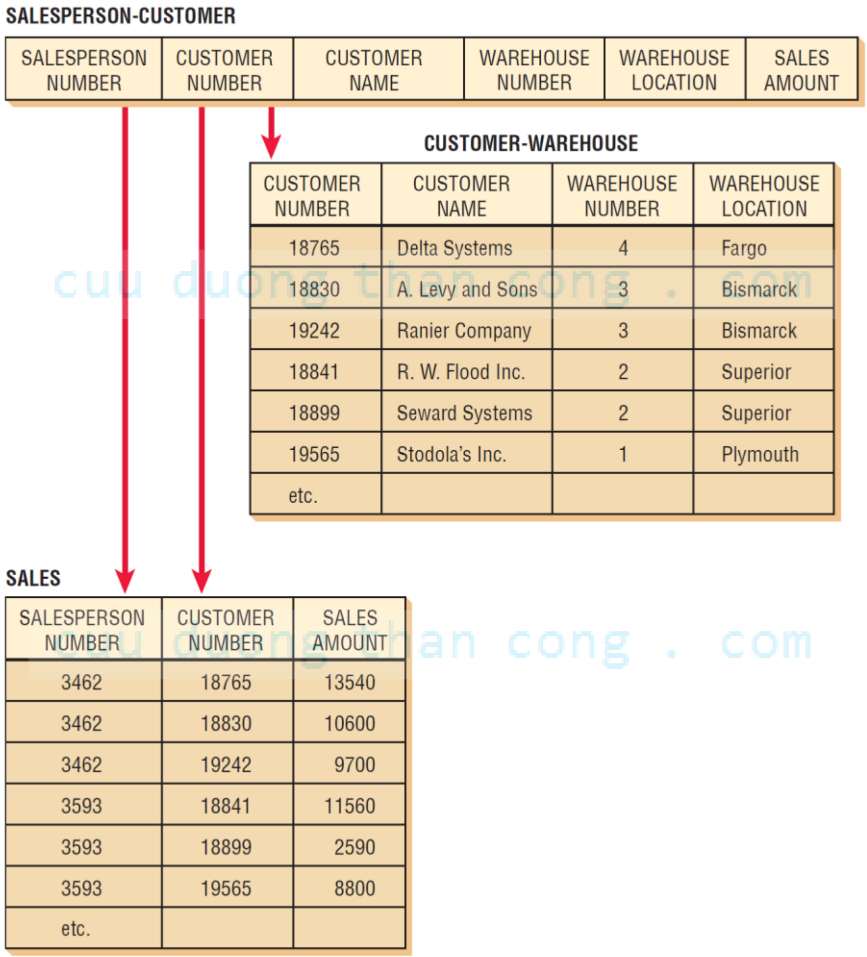
SALESPERSON-CUSTOMER

SALESPERSON NUMBER	CUSTOMER NUMBER	CUSTOMER NAME	WAREHOUSE NUMBER	WAREHOUSE LOCATION	SALES AMOUNT
3462	18765	Delta Systems	4	Fargo	13540
3462	18830	A. Levy and Sons	3	Bismarck	10600
3462	19242	Ranier Company	3	Bismarck	9700
3593	18841	R. W. Flood Inc.	2	Superior	11560
3593	18899	Seward Systems	2	Superior	2590
3593	19565	Stodola's Inc.	1	Plymouth	8800
etc.					

Second Normal Form (2NF)

- Remove any partially dependent attributes and place them in another relation
- A partial dependency is when the data are dependent on a part of a primary key
- A relation is created for the data that are only dependent on part of the key and another for data that are dependent on both parts

Figure 13.20 The relation SALESPERSON-CUSTOMER is separated into a relation called CUSTOMER-WAREHOUSE (2NF) and a relation called SALES (1NF)



SALESPERSON-CUSTOMER

SALESPERSON NUMBER	CUSTOMER NUMBER	CUSTOMER NAME	WAREHOUSE NUMBER	WAREHOUSE LOCATION	SALES AMOUNT

CUSTOMER-WAREHOUSE

CUSTOMER NUMBER	CUSTOMER NAME	WAREHOUSE NUMBER	WAREHOUSE LOCATION
18765	Delta Systems	4	Fargo
18830	A. Levy and Sons	3	Bismarck
19242	Ranier Company	3	Bismarck
18841	R. W. Flood Inc.	2	Superior
18899	Seward Systems	2	Superior
19565	Stodola's Inc.	1	Plymouth
etc.			

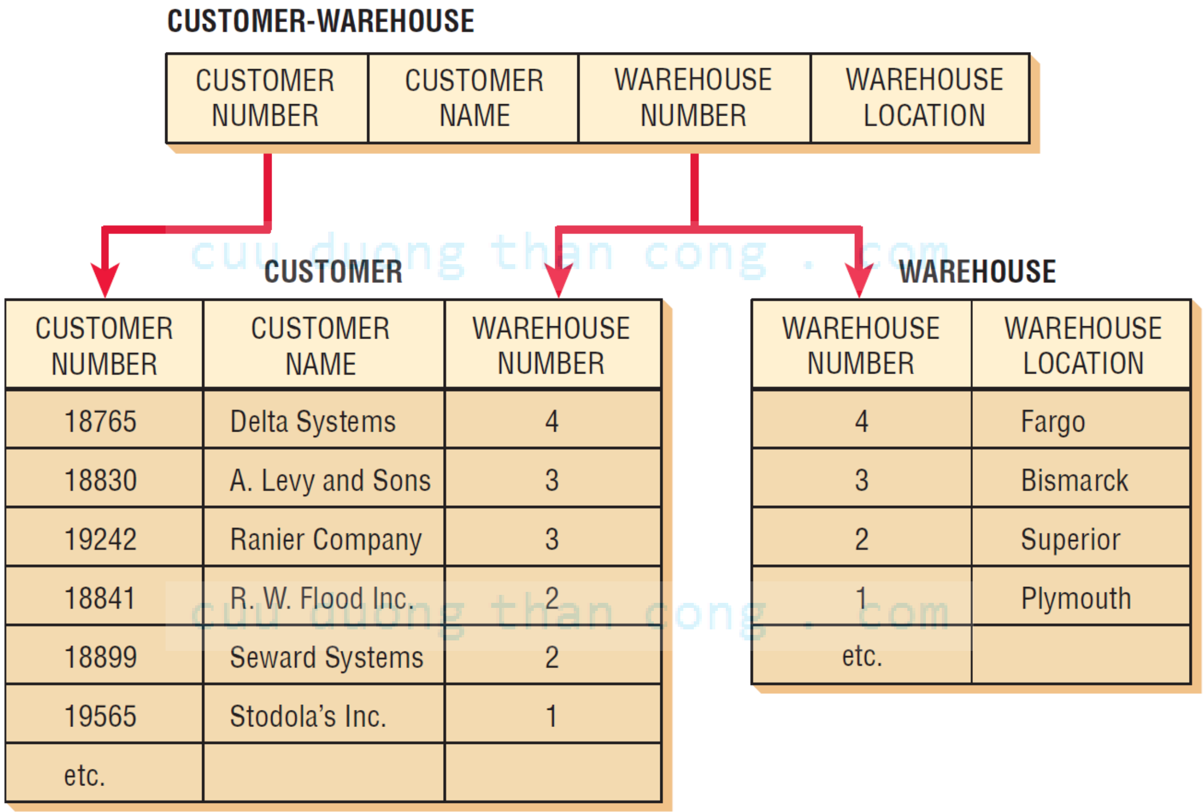
SALES

SALESPERSON NUMBER	CUSTOMER NUMBER	SALES AMOUNT
3462	18765	13540
3462	18830	10600
3462	19242	9700
3593	18841	11560
3593	18899	2590
3593	19565	8800
etc.		

Third Normal Form (3NF)

- Must be in 2NF
- Remove any transitive dependencies
- A transitive dependency is when nonkey attributes are dependent not only on the primary key, but also on a nonkey attribute

Figure 13.22 The relation CUSTOMER-WAREHOUSE is separated into two relations called CUSTOMER (1NF) and WAREHOUSE (1NF)



CUSTOMER-WAREHOUSE

CUSTOMER NUMBER	CUSTOMER NAME	WAREHOUSE NUMBER	WAREHOUSE LOCATION

CUSTOMER

CUSTOMER NUMBER	CUSTOMER NAME	WAREHOUSE NUMBER
18765	Delta Systems	4
18830	A. Levy and Sons	3
19242	Ranier Company	3
18841	R. W. Flood Inc.	2
18899	Seward Systems	2
19565	Stodola's Inc.	1
etc.		

WAREHOUSE

WAREHOUSE NUMBER	WAREHOUSE LOCATION
4	Fargo
3	Bismarck
2	Superior
1	Plymouth
etc.	

Using the Entity-relationship Diagram to Determine Record Keys

- When the relationship is one-to-many, the primary key of the file at the one end of the relationship should be contained as a foreign key on the file at the many end of the relationship
- A many-to-many relationship should be divided into two one-to-many relationships with an associative entity in the middle

Guidelines for Master File/Database Relation Design

- Each separate data entity should create a master database table
- A specific data field should exist on one master table
- Each master table or database relation should have programs to create, read, update, and delete the records

Integrity Constraints

- Entity integrity
- Referential integrity
- Domain integrity

Entity Integrity

- The primary key cannot have a null value
- If the primary key is a composite key, none of the fields in the key can contain a null value

Referential Integrity

- Referential integrity governs the nature of records in a one-to-many relationship
- Referential integrity means that all foreign keys in the many table (the child table) must have a matching record in the parent table

Referential Integrity (Continued)

Referential integrity implications:

- You cannot add a record in the child (many) table without a matching record in the parent table
- You cannot change a primary key that has matching child table records
- You cannot delete a record that has child records

Referential Integrity (Continued)

- Implemented in two ways:
  - A restricted database updates or deletes a key only if there are no matching child records
  - A cascaded database will delete or update all child records when a parent record is deleted or changed

Domain Integrity

- Domain integrity rules are used to validate the data
- Domain integrity has two forms:
  - Check constraints, which are defined at the table level
  - Rules, which are defined as separate objects and can be used within a number of fields

Anomalies

- Data redundancy
- Insert anomaly
- Deletion anomaly
- Update anomaly

Data Redundancy

- When the same data is stored in more than one place in the database
- Solved by creating tables that are in third normal form

Insert Anomaly

- Occurs when the entire primary key is not known and the database cannot insert a new record, which would violate entity integrity
- Can be avoided by using a sequence number for the primary key

Deletion Anomaly

- Happens when a record is deleted that results in the loss of other related data

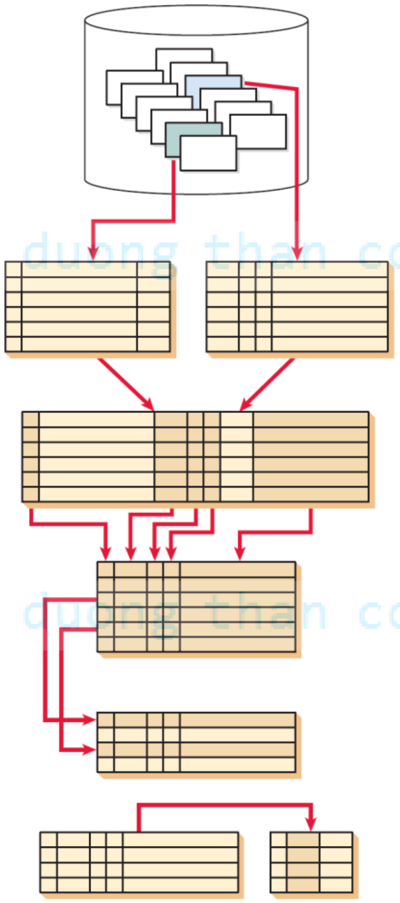
Update Anomaly

- When a change to one attribute value causes the database to either contain inconsistent data or causes multiple records to need changing
- May be prevented by making sure tables are in third normal form

Retrieving and Presenting Database Data

- Choose a relation from the database
- Join two relations together
- Project columns from the relation
- Select rows from the relation
- Derive new attributes
- Index or sort rows
- Calculate totals and performance measures
- Present data

Figure 13.28 Data are retrieved and presented in eight distinct steps



Choose a relation(s) from the database.

Join the relations together.

Project columns from the relation.

Select rows from the relation.

Derive new attributes.

Join Two Relations Together

- Takes many 3NF relations and combines them to make a more useful relation
- Join types:
  - Inner join
  - Outer join
    - Left outer join
    - Right outer join
    - Full outer join
  - Self-join

Denormalization

- Denormalization is the process of taking the logical data model and transforming it into an efficient physical model

Data Warehouses

- Used to organize information for quick and effective queries

Data Warehouses and Database Differences

- In the data warehouse, data are organized around major subjects
- Data in the warehouse are stored as summarized rather than detailed raw data
- Data in the data warehouse cover a much longer time frame than in a traditional transaction-oriented database
- Data warehouses are organized for fast queries

Data Warehouses and Database Differences (Continued)

- Data warehouses are usually optimized for answering complex queries, known as OLAP
- Data warehouses allow for easy access via data-mining software
- Data warehouses include multiple databases that have been processed so that data are uniformly defined
- Data warehouses usually include data from outside sources

Online Analytic Processing

- Online analytic processing (OLAP) is meant to answer decision makers’ complex questions by defining a multidimensional database

Data Mining

- Data mining, or knowledge data discovery (KDD), is the process of identifying patterns that a human is unable to detect

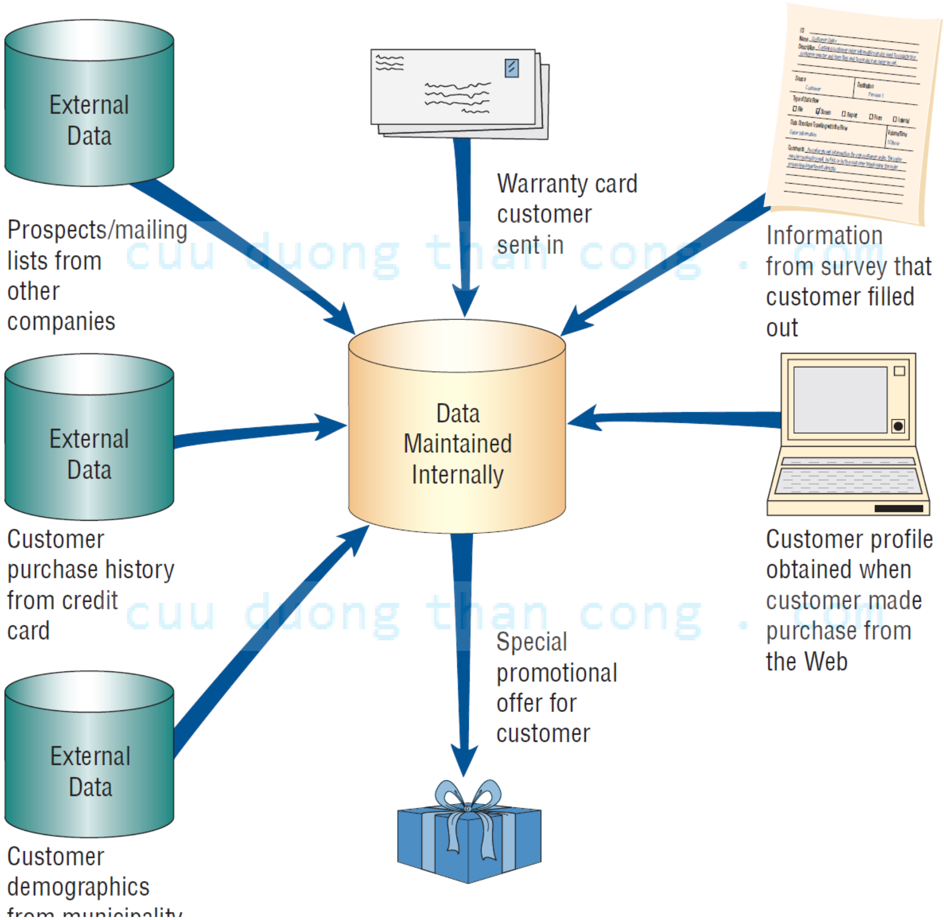
Data-Mining Decision Aids

- Siftware
  - Statistical analysis
  - Decision trees
  - Neural networks
  - Intelligent agents
  - Fuzzy logic
  - Data visualization

Data-Mining Patterns

- Associations, patterns that occur together
- Sequences, patterns of actions that take place over a period of time
- Clustering, patterns that develop among groups of people
- Trends, the patterns that are noticed over a period of time

Figure 13.37 Data mining collects personal information about customers in an effort to be more specific in interpreting and anticipating their preferences


External Data
Prospects/mailing lists from other companies
Warranty card customer sent in
Information from survey that customer filled out
External Data
Customer purchase history from credit card
Data Maintained Internally
Customer profile obtained when customer made purchase from the Web
External Data
Customer demographics from municipality
Special promotional offer for customer

Data-Mining Problems

- Costs may be too high to justify
- Has to be coordinated
- Ethical aspects

Summary

- Storing data
  - Individual files
  - Database
- Reality, data, metadata
- Conventional files
  - Type
  - Organization
- Database
  - Relational
  - Hierarchical
  - Network

Summary (Continued)

- E-R diagrams
- Normalization
  - First normal form
  - Second normal form
  - Third normal form
- Denormalization
- Data warehouse
- Data mining