Chapter 4: Design

A. Designing Databases

1. Introduction

File and database design occurs in two steps.

Develop a logical database model, which describes data using notation that corresponds to a data organization used by a database management system.

- Relational database model.

Prescribe the technical specifications for computer files and databases in which to store the data.

- Physical database design provides specifications.

Logical and physical database design in parallel with other system design steps.

Database design occurs in three stages:

a. Conceptual Design
- Entities and their relationship are identified by analyzing input and output of the system

b. Logical Design
- Attributes are identified
- Primary keys are defines
- ER diagram is used for logical representation

c. Physical Design
- Convert ER diagram to database tables
- Foreign keys are defined
- Normalization

2. Database Design

Features	Conceptual Design	Logical Design	Physical Design
Entity Names	√	√	
Entity Relationships	√	√	
Attributes		√	√
Primary Keys		√	√
Foreign Keys			√
Table Names			√
Normalization			√

Generic steps of database design are:

- Discover and organize entities and their attributes
- Define primary keys
- Define the relationships between different entities
- Create ER diagram
- Convert ER diagram to tables using sample data using sample data
- Normalize the tables

Process of Database Design

Physical Design

Based upon results of logical database design

Key decisions

- Choosing the storage format (called data type) for each attribute from the logical database model; the format is chosen to minimize storage space and to maximize data quality. Data type involves choosing length, coding scheme, number of decimal places, minimum and maximum values, and potentially many other parameters for each attribute.
- Grouping attributes from the logical database model into physical records (in general, this is called selecting a stored record, or data, structure).
- Arranging related records in secondary memory (hard disks and magnetic tapes) so that individual records and groups of records can be stored, retrieved, and updated rapidly (called file organization). You should also consider protecting data and recovering data after errors are found.
- Selecting media and structures for storing data to make access more efficient. The choice of media affects the utility of different file organizations. The primary structure used today to make access to data more rapid is key indexes on unique and no unique keys.

3. Relational Database Model

Data represented as a set of related tables or relations.

Relation is two-dimensional table of data. Each relation consists of a set of named columns and an arbitrary number of unnamed rows.

Relations have several properties that distinguish them from non-relational tables:

- Entries in cells are simple. An entry at the intersection of each row and column has a single value.
- Entries in a given column are from the same set of values.
- Each row is unique. Uniqueness is guaranteed because the relation has a non-empty primary key value.
- The sequence of columns can be interchanged without changing the meaning or use of the relation.
- The rows may be interchanged or stored in any sequences.

Well-structured relation

A relation that contains a minimum amount of redundancy and that allows users to insert, modify, and delete the rows without error or inconsistencies; also known as a table.

4. Normalization

We have presented an intuitive discussion of well-structured relations; however, we need rules and a process for designing them.

Normalization is a process for converting complex data structures into simple, stable data structures.

We have presented an intuitive discussion of well-structured relations; however, we need rules and a process for designing them. Normalization is a process for con verting complex data structures into simple, stable data structures

Rules of Normalization

Normalization is based on well-accepted principles and rules. There are many normalization rules, more than can be covered (for a more complete coverage).

a. First Normal Form (1NF)

- A relation will be 1NF if it contains an atomic value.
- It states that an attribute of a table cannot hold multiple values. It must hold only single-valued attribute.
- First normal form disallows the multi-valued attribute, composite attribute, and their combinations.

Example: Relation EMPLOYEE is not in 1NF because of multi-valued attribute EMP_PHONE.

EMP_ID	EMP_NAME	EMP_PHONE	EMP_STATE
14	John	7272826385, 9064738238	UP
20	Harry	8574783832	Bihar
12	Sam	7390372389, 8589830302	Punjab

The decomposition of the EMPLOYEE table into 1NF has been shown below:

EMP_ID	EMP_NAME	EMP_PHONE	EMP_STATE
14	John	7272826385	UP
14	John	9064738238	UP
20	Harry	8574783832	Bihar
12	Sam	7390372389	Punjab
12	Sam	8589830302	Punjab

b. Second Normal Form (2NF)

- In the 2NF, relational must be in 1NF.
- In the second normal form, all non-key attributes are fully functional dependent on the primary key

Example: Let's assume, a school can store the data of teachers and the subjects they teach. In a school, a teacher can teach more than one subject.

TEACHER table

TEACHER_ID	SUBJECT	TEACHER_AGE
25	Chemistry	30
25	Biology	30
47	English	35
83	Math	38
83	Computer	38

In the given table, non-prime attribute TEACHER_AGE is dependent on TEACHER_ID which is a proper subset of a candidate key. That's why it violates the rule for 2NF.

To convert the given table into 2NF, we decompose it into two tables:

TEACHER_DETAIL table:

TEACHER_ID	TEACHER_AGE
25	30
47	35
83	38

TEACHER_SUBJECT table:

TEACHER_ID	SUBJECT
25	Chemistry
25	Biology
47	English
83	Math

c. Third Normal Form (3NF)

- A relation will be in 3NF if it is in 2NF and not contain any transitive partial dependency.
- 3NF is used to reduce the data duplication. It is also used to achieve the data integrity.
- If there is no transitive dependency for non-prime attributes, then the relation must be in third normal form.

A relation is in third normal form if it holds at least one of the following conditions for every non-trivial function dependency X → Y.

1. X is a super key.
2. Y is a prime attribute, i.e., each element of Y is part of some candidate key.

Example:

EMPLOYEE_DETAIL table:

EMP_ID	EMP_NAME	EMP_ZIP	EMP_STATE	EMP_CITY
222	Harry	201010	UP	Noida
333	Stephan	02228	US	Boston
444	Lan	60007	US	Chicago
555	Katharine	06389	UK	Norwich
666	John	462007	MP	Bhopal

Super key in the table above:

1. {EMP_ID}, {EMP_ID, EMP_NAME}, {EMP_ID, EMP_NAME, EMP_ZIP}....so on

Candidate key: {EMP_ID}

Non-prime attributes: In the given table, all attributes except EMP_ID are non-prime.

Here, EMP_STATE & EMP_CITY dependent on EMP_ZIP and EMP_ZIP dependent on EMP_ID. The non-prime attributes (EMP_STATE, EMP_CITY) transitively dependent on super key (EMP_ID). It violates the rule of third normal form.

That's why we need to move the EMP_CITY and EMP_STATE to the new <EMPLOYEE_ZIP> table, with EMP_ZIP as a Primary key.

EMPLOYEE table:

EMP_ID	EMP_NAME	EMP_ZIP
222	Harry	201010
333	Stephan	02228
444	Lan	60007
555	Katharine	06389
666	John	462007

EMPLOYEE_ZIP table:

EMP_ZIP	EMP_STATE	EMP_CITY
201010	UP	Noida
02228	US	Boston
60007	US	Chicago
06389	UK	Norwich
462007	MP	Bhopal

d. Boyce Codd normal form (BCNF)

- BCNF is the advance version of 3NF. It is stricter than 3NF.
- A table is in BCNF if every functional dependency X → Y, X is the super key of the table.
- For BCNF, the table should be in 3NF, and for every FD, LHS is super key.

Example: Let's assume there is a company where employees work in more than one department.

EMPLOYEE table:

EMP_ID	EMP_COUNTRY	EMP_DEPT	DEPT_TYPE	EMP_DEPT_NO
264	India	Designing	D394	283
264	India	Testing	D394	300
364	UK	Stores	D283	232
364	UK	Developing	D283	549

In the above table Functional dependencies are as follows:

1. EMP_ID → EMP_COUNTRY
2. EMP_DEPT → {DEPT_TYPE, EMP_DEPT_NO}

Candidate key: {EMP-ID, EMP-DEPT}

The table is not in BCNF because neither EMP_DEPT nor EMP_ID alone are keys.

To convert the given table into BCNF, we decompose it into three tables:

EMP_COUNTRY table:

EMP_ID	EMP_COUNTRY
264	India
264	India

EMP_DEPT table:

EMP_DEPT	DEPT_TYPE	EMP_DEPT_NO
Designing	D394	283
Testing	D394	300
Stores	D283	232
Developing	D283	549

EMP_DEPT_MAPPING table:

EMP_ID	EMP_DEPT
D394	283
D394	300
D283	232
D283	549

Functional dependencies:

1. EMP_ID → EMP_COUNTRY
2. EMP_DEPT → {DEPT_TYPE, EMP_DEPT_NO}

Candidate keys:

For the first table: EMP_ID
For the second table: EMP_DEPT
For the third table: {EMP_ID, EMP_DEPT}

Now, this is in BCNF because left side part of both the functional dependencies is a key.

e. Fourth normal form (4NF)

- A relation will be in 4NF if it is in Boyce Codd normal form and has no multi-valued dependency.
- For a dependency A → B, if for a single value of A, multiple values of B exists, then the relation will be a multi-valued dependency.

Example

STUDENT

STU_ID	COURSE	HOBBY
21	Computer	Dancing
21	Math	Singing
34	Chemistry	Dancing
74	Biology	Cricket
59	Physics	Hockey

The given STUDENT table is in 3NF, but the COURSE and HOBBY are two independent entity. Hence, there is no relationship between COURSE and HOBBY.

In the STUDENT relation, a student with STU_ID, **21** contains two courses, **Computer** and **Math** and two hobbies, **Dancing** and **Singing**. So there is a Multi-valued dependency on STU_ID, which leads to unnecessary repetition of data.

So to make the above table into 4NF, we can decompose it into two tables:

STUDENT_COURSE

STU_ID	COURSE
21	Computer
21	Math
34	Chemistry
74	Biology
59	Physics

STUDENT_HOBBY

STU_ID	HOBBY
21	Dancing
21	Singing
34	Dancing
74	Cricket
59	Hockey

f. Fifth normal form (5NF)

- A relation is in 5NF if it is in 4NF and not contains any join dependency and joining should be lossless.
- 5NF is satisfied when all the tables are broken into as many tables as possible in order to avoid redundancy.
- 5NF is also known as Project-join normal form (PJ/NF).

Example

SUBJECT	LECTURER	SEMESTER
Computer	Anshika	Semester 1
Computer	John	Semester 1
Math	John	Semester 1
Math	Akash	Semester 2
Chemistry	Praveen	Semester 1

In the above table, John takes both Computer and Math class for Semester 1 but he doesn't take Math class for Semester 2. In this case, combination of all these fields required to identify a valid data.

Suppose we add a new Semester as Semester 3 but do not know about the subject and who will be taking that subject so we leave Lecturer and Subject as NULL. But all three columns together acts as a primary key, so we can't leave other two columns blank.

So to make the above table into 5NF, we can decompose it into three relations P1, P2 & P3:

P1

SEMESTER	SUBJECT
Semester 1	Computer
Semester 1	Math
Semester 1	Chemistry
Semester 2	Math

P2

SUBJECT	LECTURER
Computer	Anshika
Computer	John
Math	John
Math	Akash
Chemistry	Praveen

P3

SEMSTER	LECTURER
Semester 1	Anshika
Semester 1	John
Semester 1	John
Semester 2	Akash
Semester 1	Praveen

5. Transforming E-R Diagrams into Relations

E-R Structure	Relational Representation
Regular entity	Create a relation with primary key and nonkey attributes.
Weak entity	Create a relation with a composite primary key (which includes the primary key of the entity on which this weak entity depends) and nonkey attributes.
Binary or unary 1:1 relationship	Place the primary key of either entity in the relation for the other entity or do this for both entities.
Binary 1:*N* relationship	Place the primary key of the entity on the one side of the relationship as a foreign key in the relation for the entity on the many side.
Binary or unary *M*:*N* relationship or associative entity	Create a relation with a composite primary key using the primary keys of the related entities, plus any nonkey attributes associative entity of the relationship or associative entity.
Binary or unary *M*:*N* relationship or associative entity with additional key(s)	Create a relation with a composite primary key using the primary keys of the related entities and additional primary key attributes associated with the relationship or associative entity, plus any nonkey attributes of the relationship or associative entity.
Binary or unary *M*:*N* relationship or associative entity with its own key	Create a relation with the primary key associated with the relationship or associative entity, plus any nonkey attributes of the relationship or associative entity and the primary keys of the related entities (as foreign key attributes).
Supertype/subtype	Create a relation for the superclass, which contains the primary relationship key and all nonkey attributes in common with all subclasses, plus create a separate relation for each subclass with the same primary key (with the same or local name) but with only the nonkey attributes related to that subclass.

Figure 5-8: Mapping a regular entity

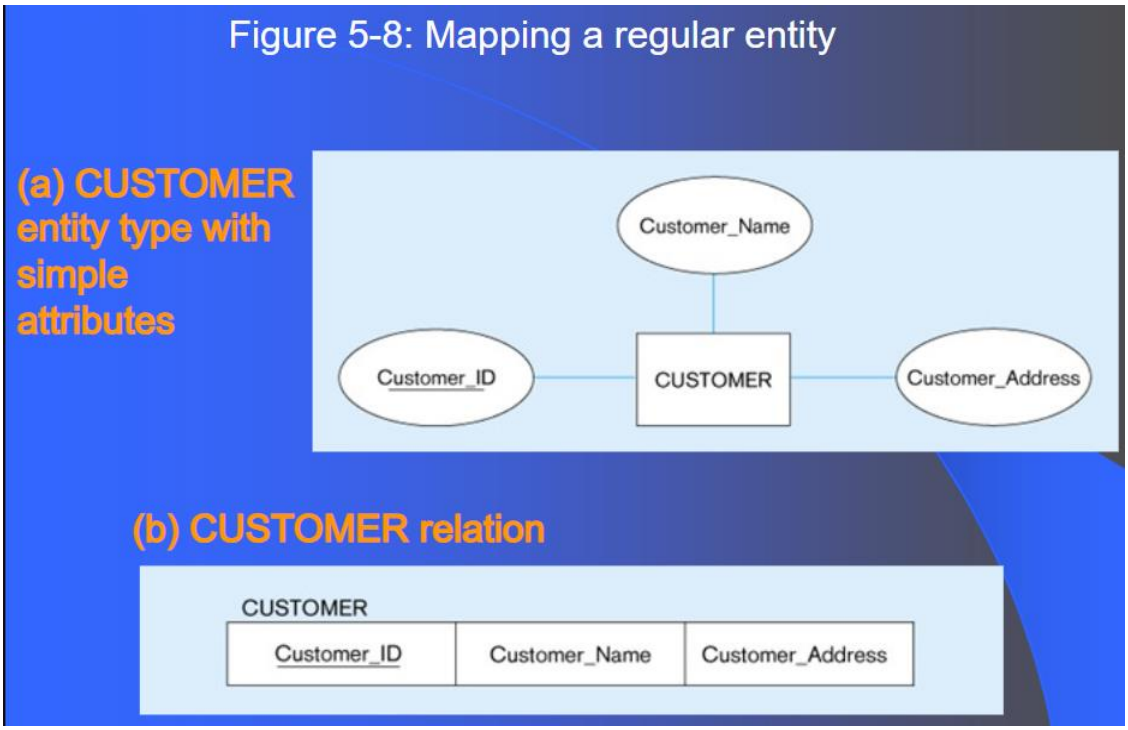


Figure 5-9: Mapping a composite attribute

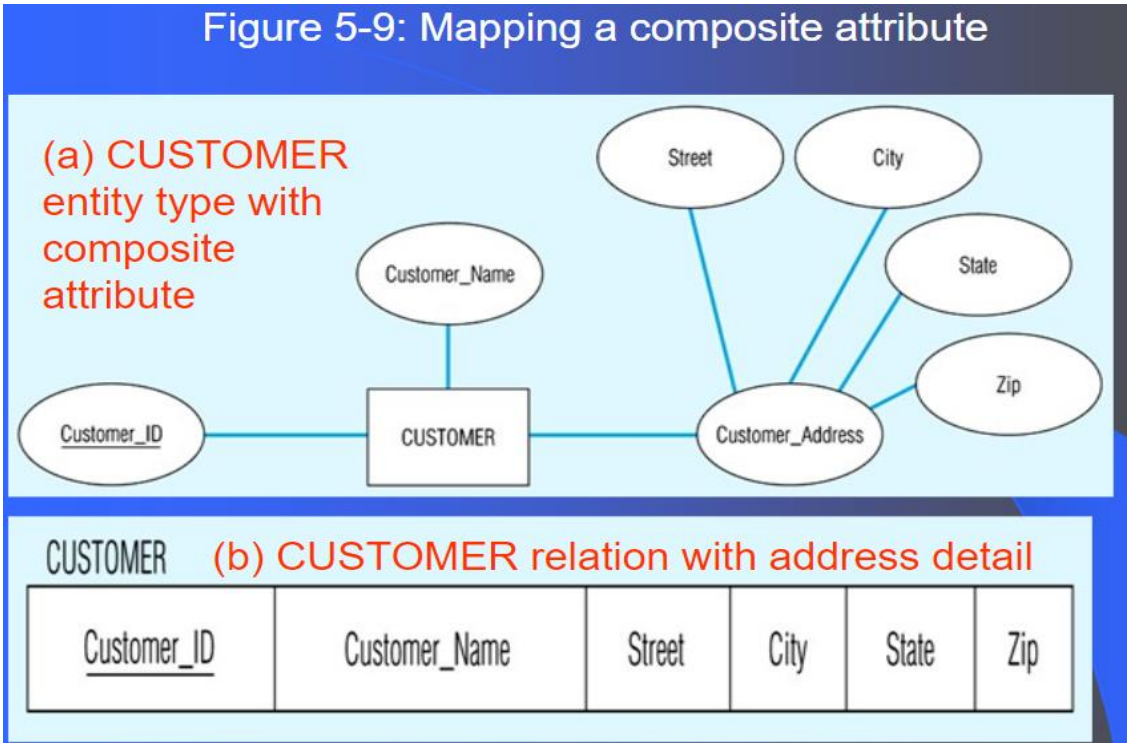


Figure 5-10: Mapping a multivalued attribute

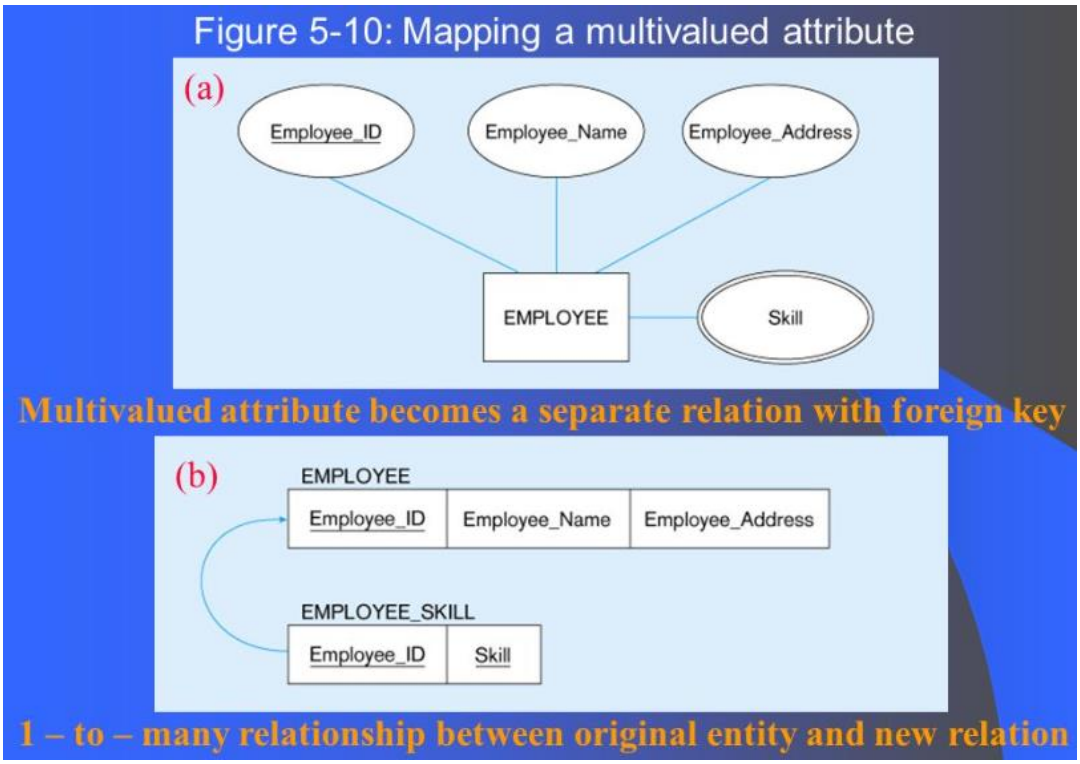


Figure 5-11: Example of mapping a weak entity

(a) Weak entity DEPENDENT

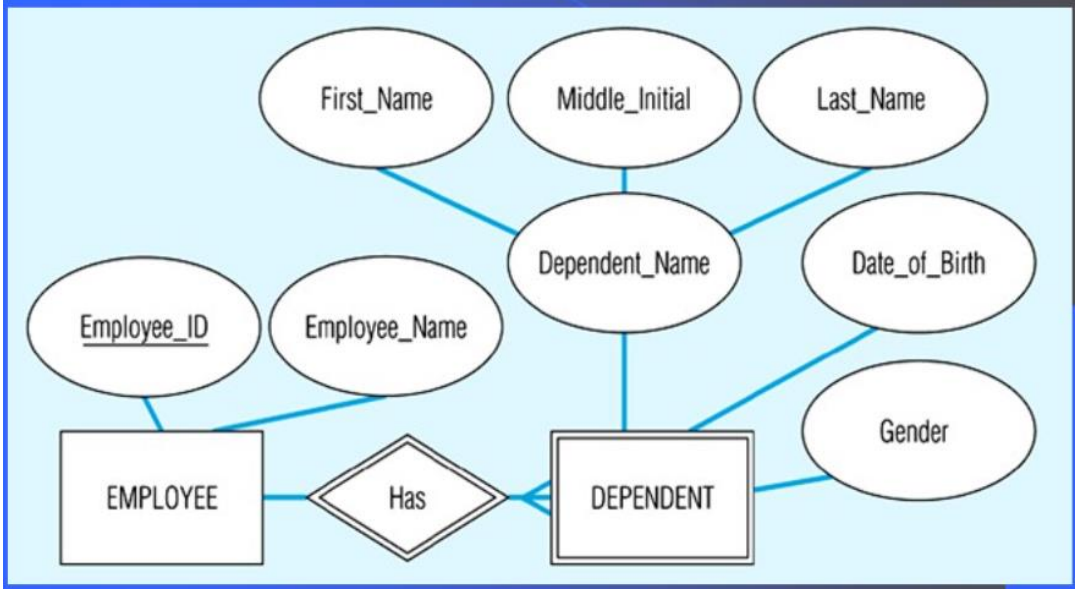


Figure 5-11(b) Relations resulting from weak entity

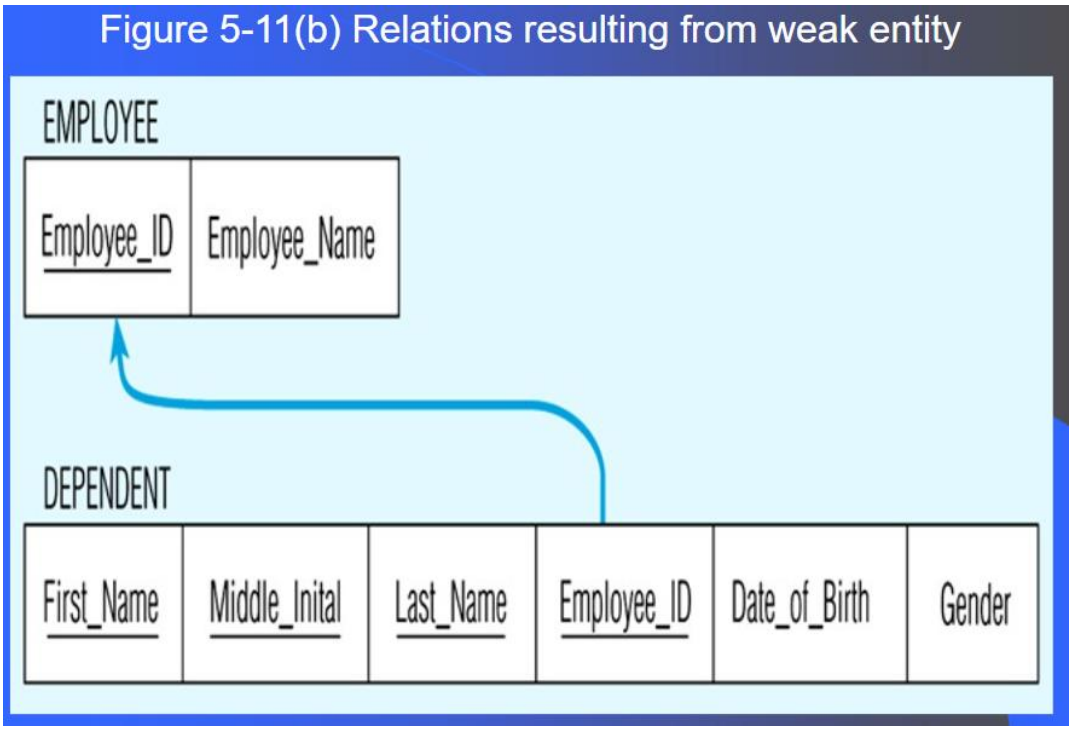


Figure 5-12: Example of mapping a 1:M relationship

(a) Relationship between customers and orders

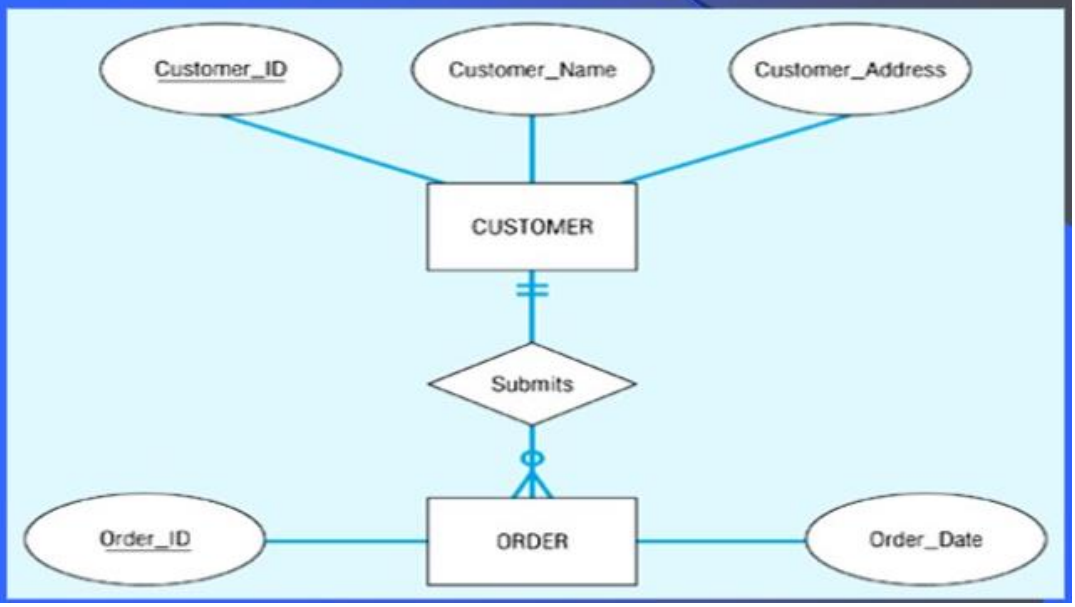


Figure 5-12(b) Mapping the relationship

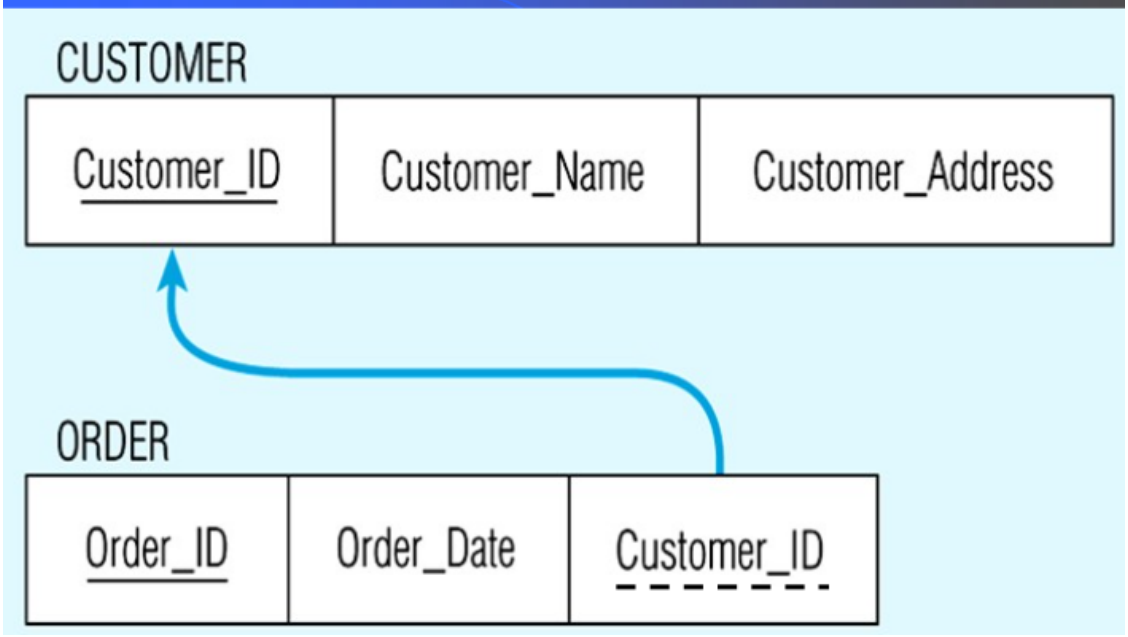


Figure 5-14: Mapping a binary 1:1 relationship

(a) Binary 1:1 relationship

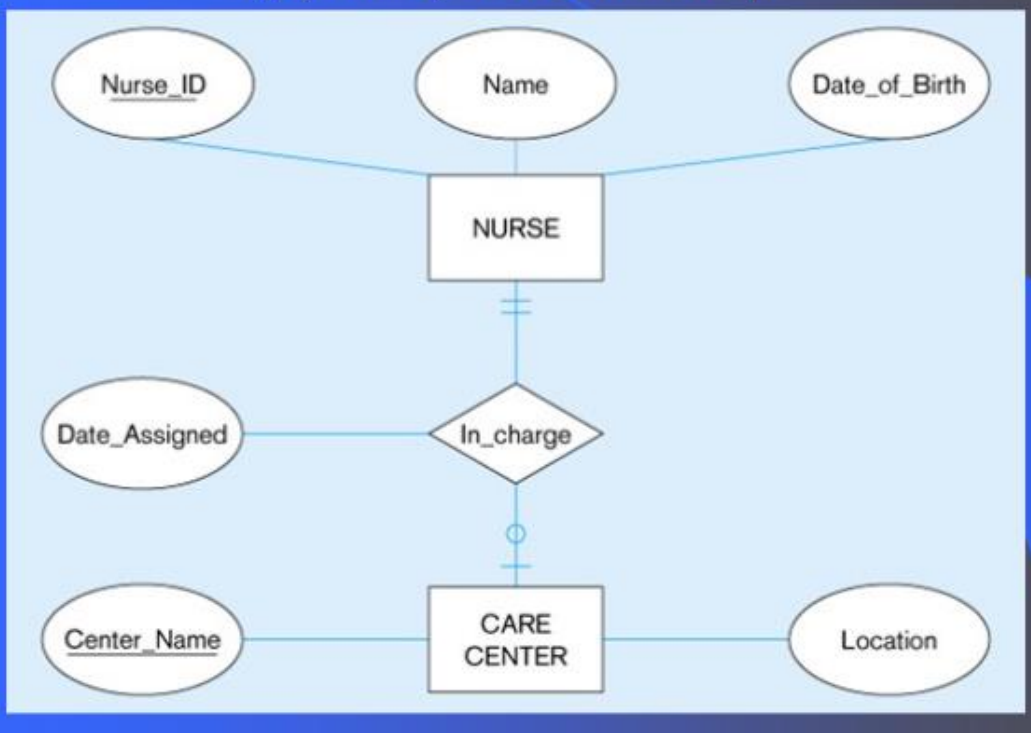


Figure 5-14(b) Resulting relations

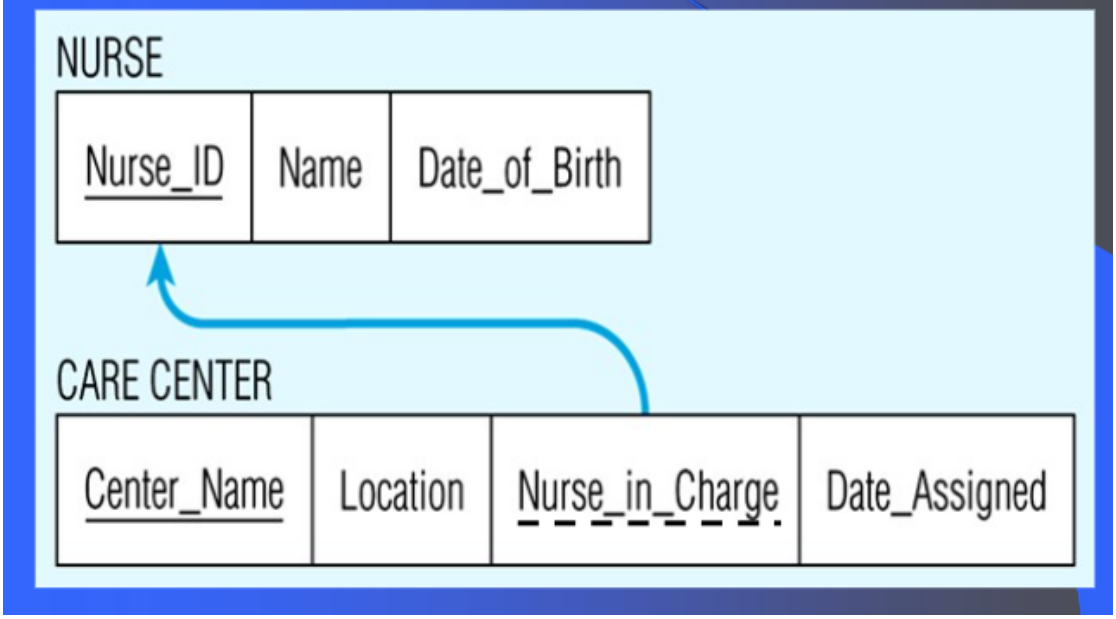


Figure 5-13: Example of mapping an M:N relationship

(a) ER diagram (M:N)

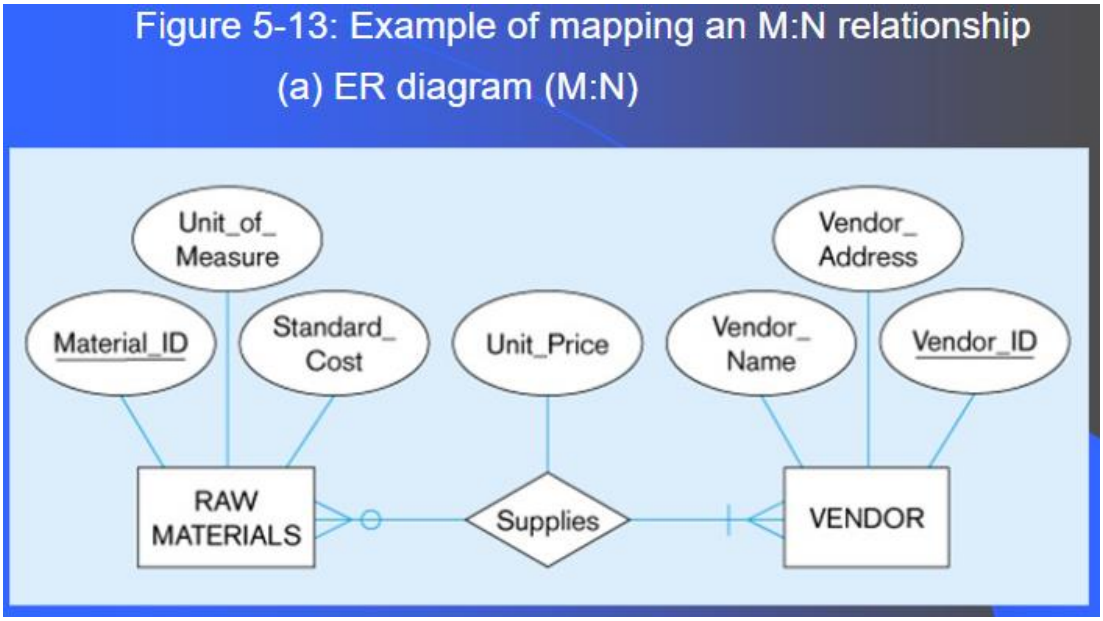


Figure 5-13(b) Three resulting relations

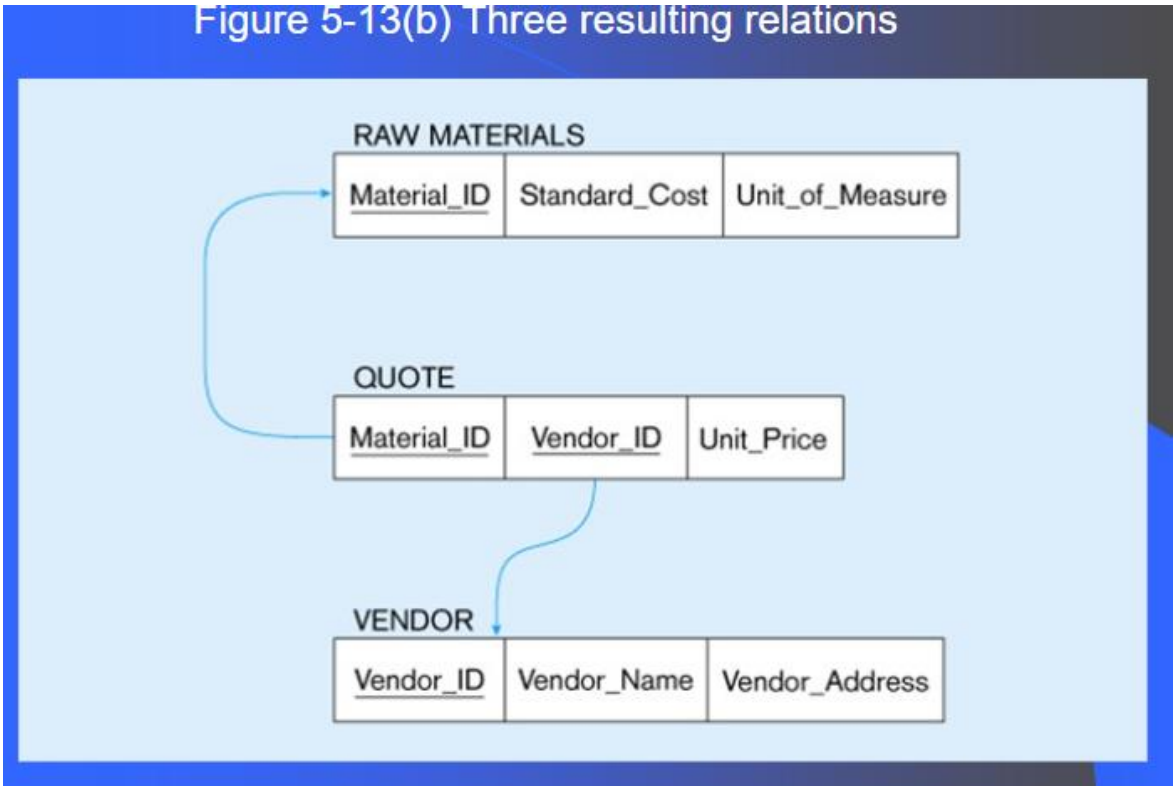


Figure 5-15: Mapping an associative entity

(a) Associative entity

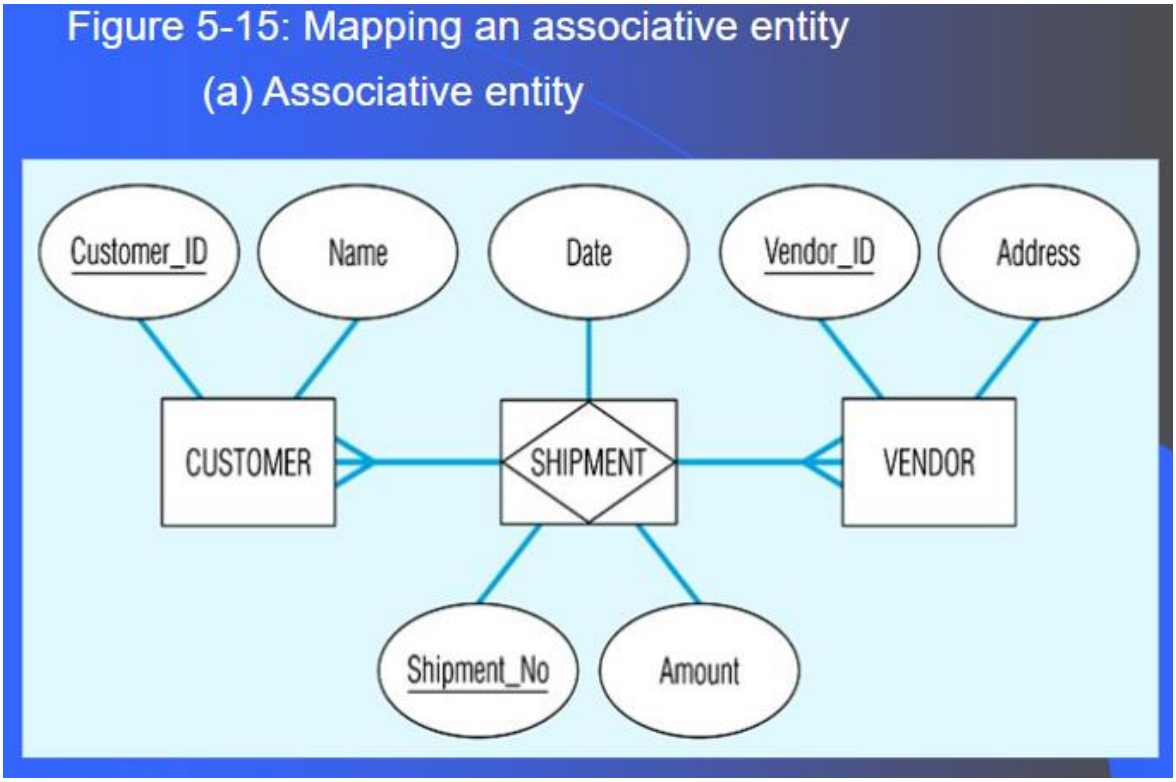


Figure 5-15(b) Three resulting relations

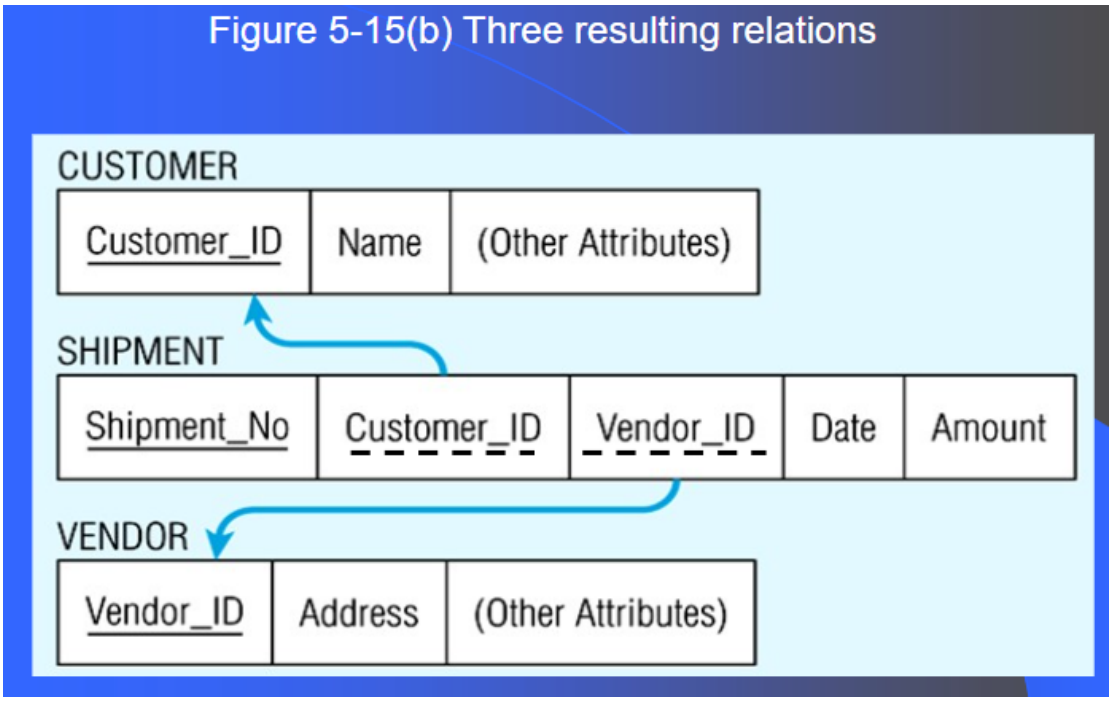


Figure 5-17: Mapping a unary 1:N relationship

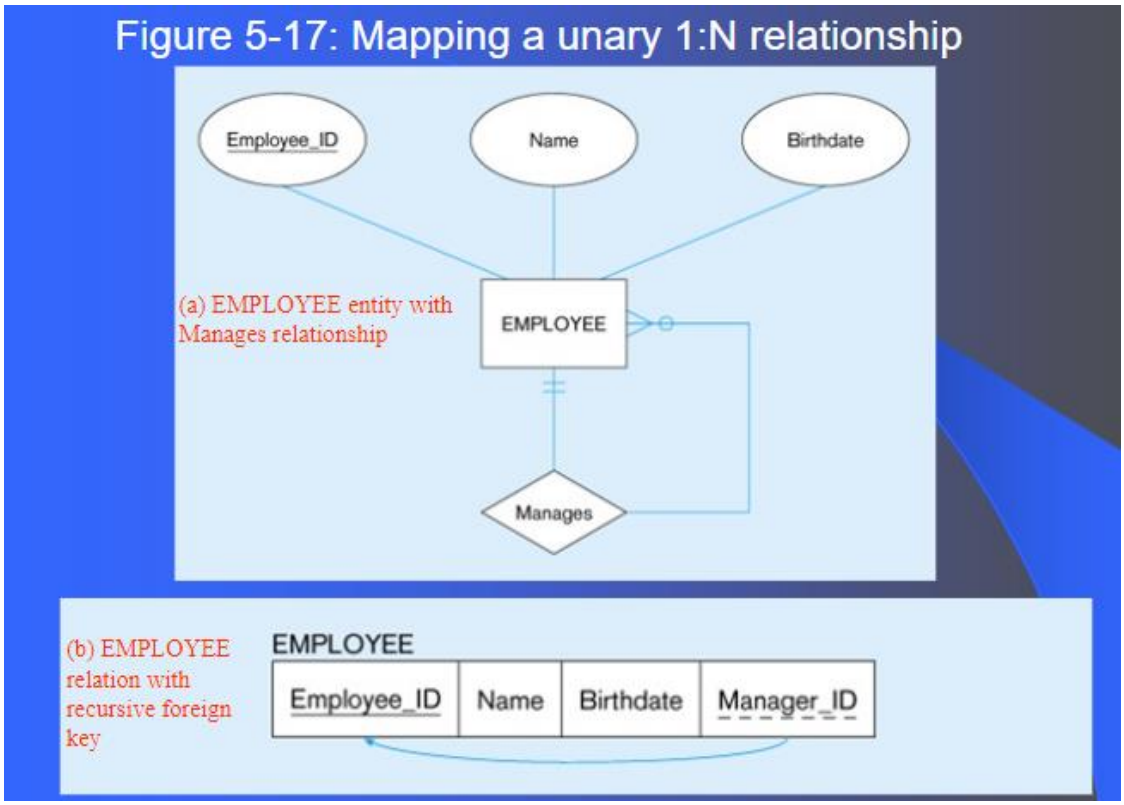
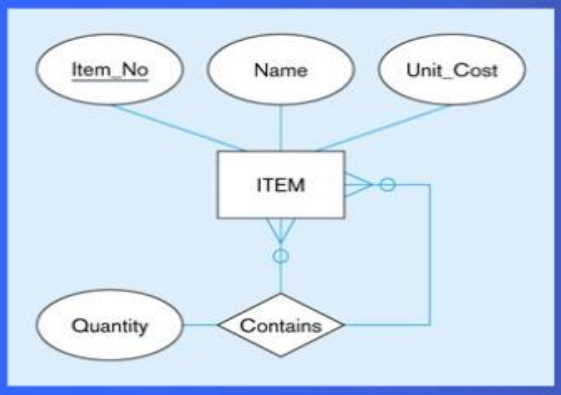


Figure 5-18: Mapping a unary M:N relationship



(a) Bill-of-materials relationships (M:N)

(b) ITEM and COMPONENT relations

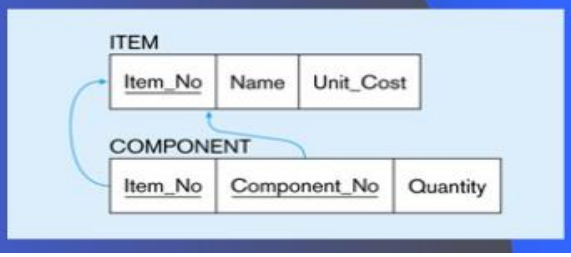


Figure 5-19: Mapping a ternary relationship

(a) Ternary relationship with associative entity

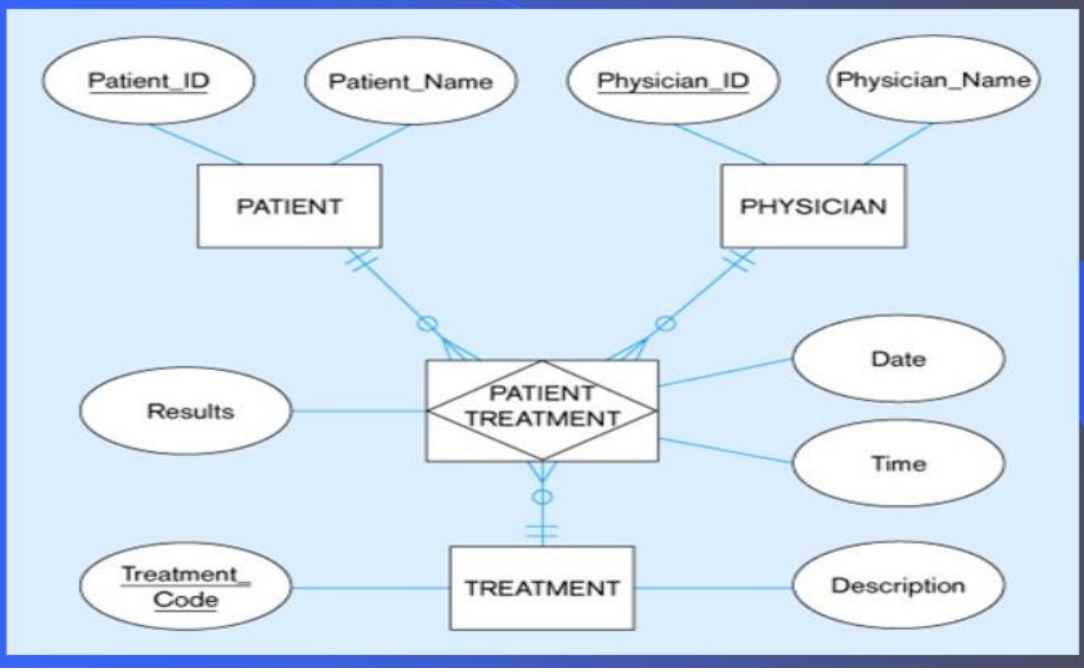


Figure 5-19(b) Mapping the ternary relationship

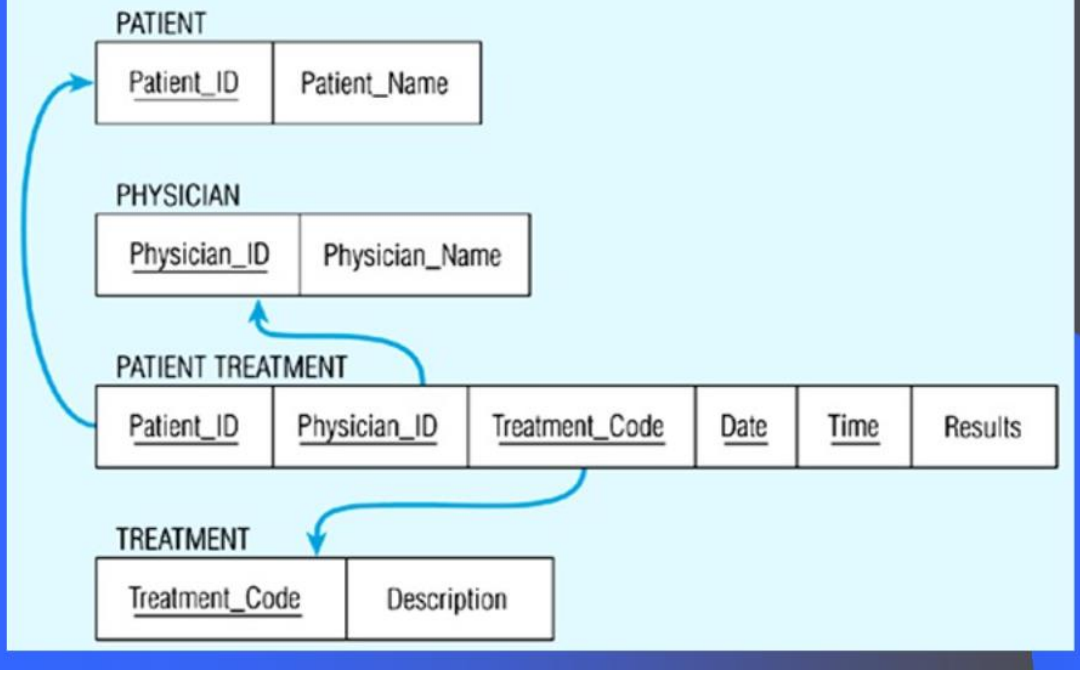
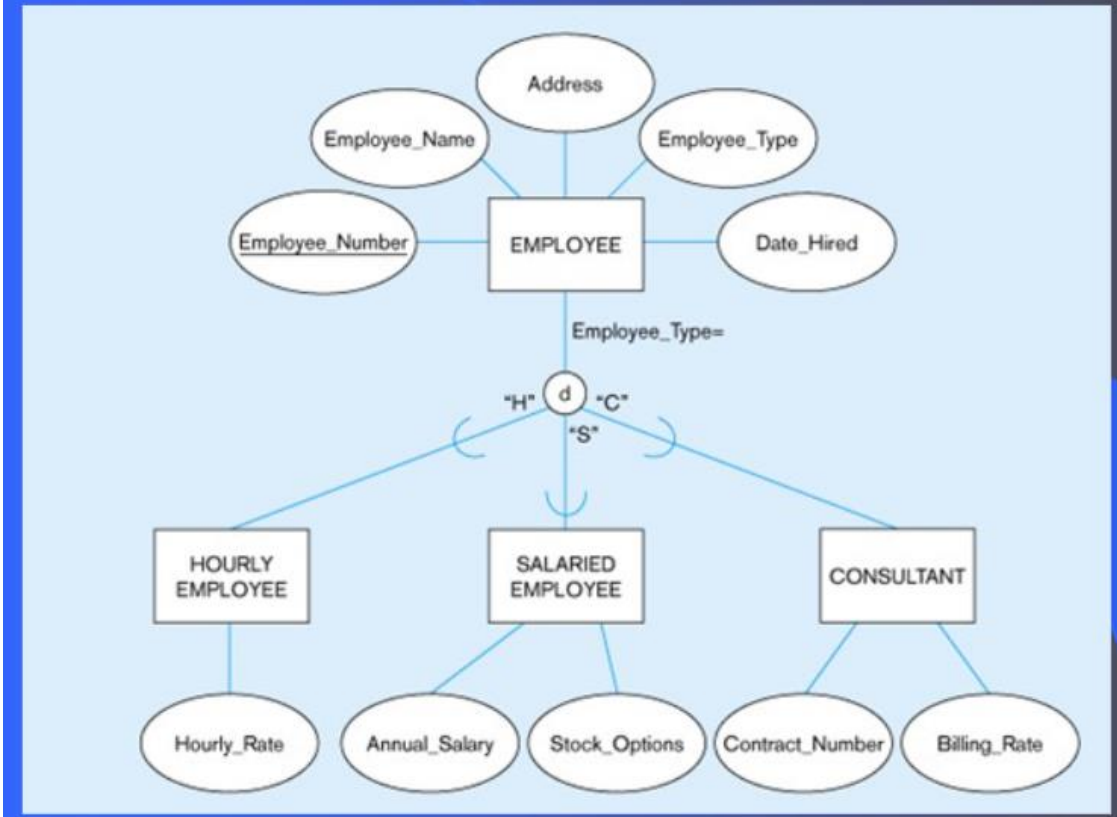
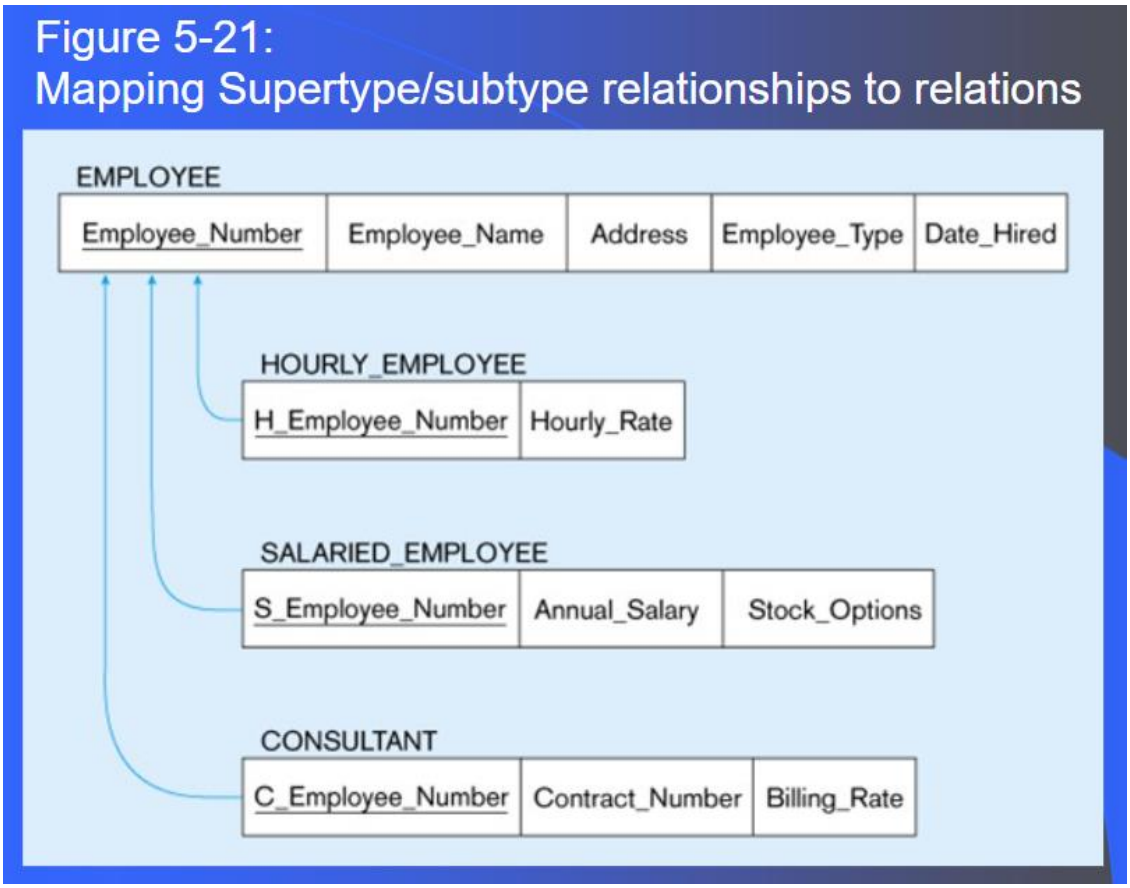


Figure 5-20: Supertype/subtype relationships





6. Physical File and Database Design

Designing physical files and databases requires certain information that should have been collected and produced during prior SDLC phases.

This information includes the following:

Normalized relations, including volume estimates

- Definitions of each attribute
- Descriptions of where and when data are used: entered, retrieved, deleted, and updated (including frequencies)
- Expectations or requirements for response time and data integrity
- Descriptions of the technologies used for implementing the files and database so that the range of required decisions and choices for each is known

Normalized relations are, of course, the result of logical database design. Statistics on the number of rows in each table as well as the other information listed above may have been collected during requirements determination in systems analysis. If not, these items need to be discovered to proceed with database design.

We take a bottom-up approach to reviewing physical file and database design. Thus, we begin the physical design phase by addressing the design of physical fields for each attribute in a logical data model.

7. Designing Fields

A field is the smallest unit of application data recognized by system software, such as a programming language or database management system. An attribute from a logical database model may be represented by several fields.

For example, a student name attribute in a normalized student relation might be represented as three fields: last name, first name, and middle initial. In general, you will represent each attribute from each normalized relation as one or more fields.

The basic decisions you must make in specifying each field concern the type of data (or storage type) used to represent the field and data integrity controls for the field.

Choosing Data types:

A data type is a coding scheme recognized by system software for representing organizational data. The bit pattern of the coding scheme is usually immaterial to you, but the space to store data and the speed required to access data are of consequence in the physical file and database design. The specific file or database management software you use with your system will dictate which choices are available to you.

Selecting a data type balances four objectives that will vary in degree of impor tance depending on the application:

- Minimize storage space
- Represent all possible values of the field
- Improve data integrity for the field
- Support all data manipulations desired on the field

Calculated Fields: It is common for an attribute to be mathematically related to other data. A field that can be derived from other database fields is called a calculated field (or a computed field or a derived field). If you specify a field as calculated, you would then usually be prompted to enter the formula for the calculation; the formula can involve other fields from the same record and possibly fields from records in related files. The database technology will either store the calculated value or compute it when requested.

Coding and Compression Techniques: Some attributes have very few values from a large range of possible values. **For example,** suppose that each product from PVF has a finish attribute, with possible values of Birch, Walnut, Oak, and so forth. To store this attribute as text might require 12, 15, or even 20 bytes to represent the longest finish value. Suppose that even a liberal estimate is that PVF will never have more than 25 finishes. Thus, a single alphabetic or alphanumeric character would be more than sufficient. We not only reduce storage space but also increase integrity (by restricting input to only a few values. Codes also have disadvantages. If used in system inputs and outputs, they can be more difficult for users to remember, and programs must be written to decode fields if codes will not be displayed.

8. Designing Physical tables:

A relational database is a set of related tables (tables are related by foreign keys referencing primary keys). In logical database design, you grouped into a relation those attributes that concern some unifying, normalized business concept, such as a customer, product, or employee.

Physical table is a named set of rows and columns that specifies the fields in each row of the table. The design of a physical table has two goals: efficient use of secondary storage and data processing speed.

The efficient use of secondary storage (disk space) relates to how data are loaded on disks. Disks are physically divided into units (called pages) that can be read or written in one machine operation. Space is used efficiently when the physical length of a table row divides close to evenly into the length of the storage unit. For many information systems, this even division is very difficult to achieve because it depends on factors, such as operating system parameters, outside the control of each database.

A second and often more important consideration when selecting a physical table design is efficient data processing. Data are most efficiently processed when they are stored close to one another in secondary memory, thus minimizing the number of input/output (I/O) operations that must be performed. Typically, the data in one physical table (all the rows and fields in those rows) are stored close together on disk.

B. Designing Forms and Reports

1. Introduction

Form: A business document that contains some organizational related data and may include some areas where additional data are to be filled in. An instance of a form is typically based on one database record.

Report: A business document that contains only predefined data. A passive document for reading or viewing data. Typically contains data from many databases records or transactions.

Common Types of Business Reports

Report Name	Description
Scheduled Reports	Reports produced at predefined intervals—daily, weekly, or monthly—to support the routine informational needs of an organization.
Key-Indicator Reports	Reports that provide a summary of critical information on a recurring basis.
Exception Reports	Reports that highlight data that are out of the normal operating range.
Drill-Down Reports	Reports that provide details behind the summary values on a key-indicator or exception report.
Ad-hoc Reports	Unplanned information requests in which information is gathered to support a nonroutine decision.

2. Designing Forms and Reports

- ✓ User-focused activity
- ✓ Follows a prototyping approach

Requirements determination:

- Who will use the form or report?
- What is the purpose of the form or report?
- When is the report needed or used?
- Where does the form or report need to be delivered and used?
- How many people need to use or view the form or report?

Prototyping

- Initial prototype is designed from requirements.
- Users review prototype design and either accept the design or request changes.
- If changes are requested, the construction-evaluation-refinement cycle is repeated until the design is accepted.

A coding sheet is an "old" tool for designing forms and reports, usually associated with text-based forms and reports for mainframe applications.

Visual Basic and other development tools provide computer-aided GUI form and report generation.

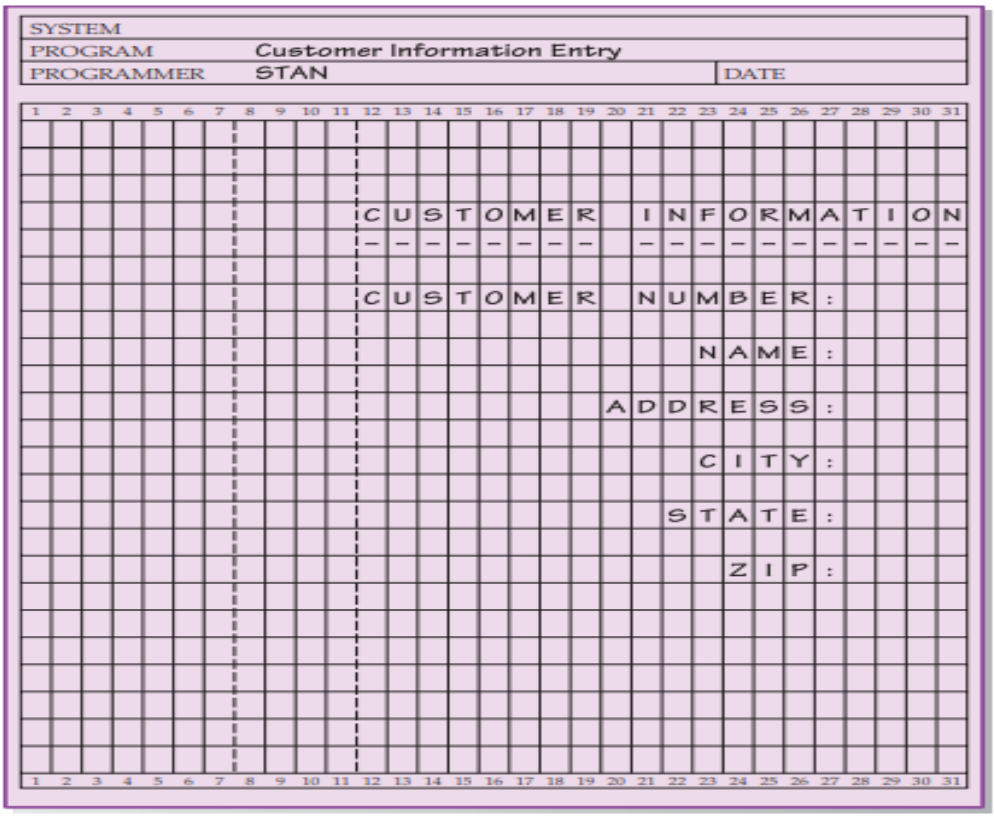
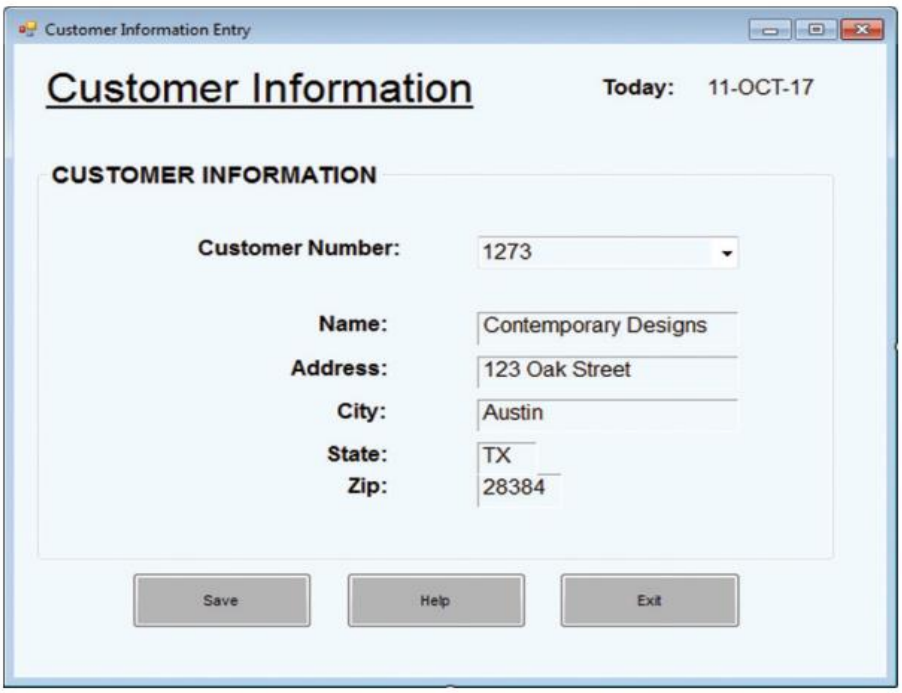


FIGURE 10-2
The layout of a data input form using a coding sheet

FIGURE 10-3B
A data input screen designed in Microsoft's Visual Basic.NET
(*Source:* Microsoft Corporation.)



Deliverables and outcomes
Design specifications have three sections:

- **Narrative overview**: characterizes users, tasks, system, and environmental factors
- **Sample design:** image of the form (from coding sheet or form building development tool)
- **Testing and usability assessment:** measuring test/usability results (consistency, sufficiency, accuracy, etc.)

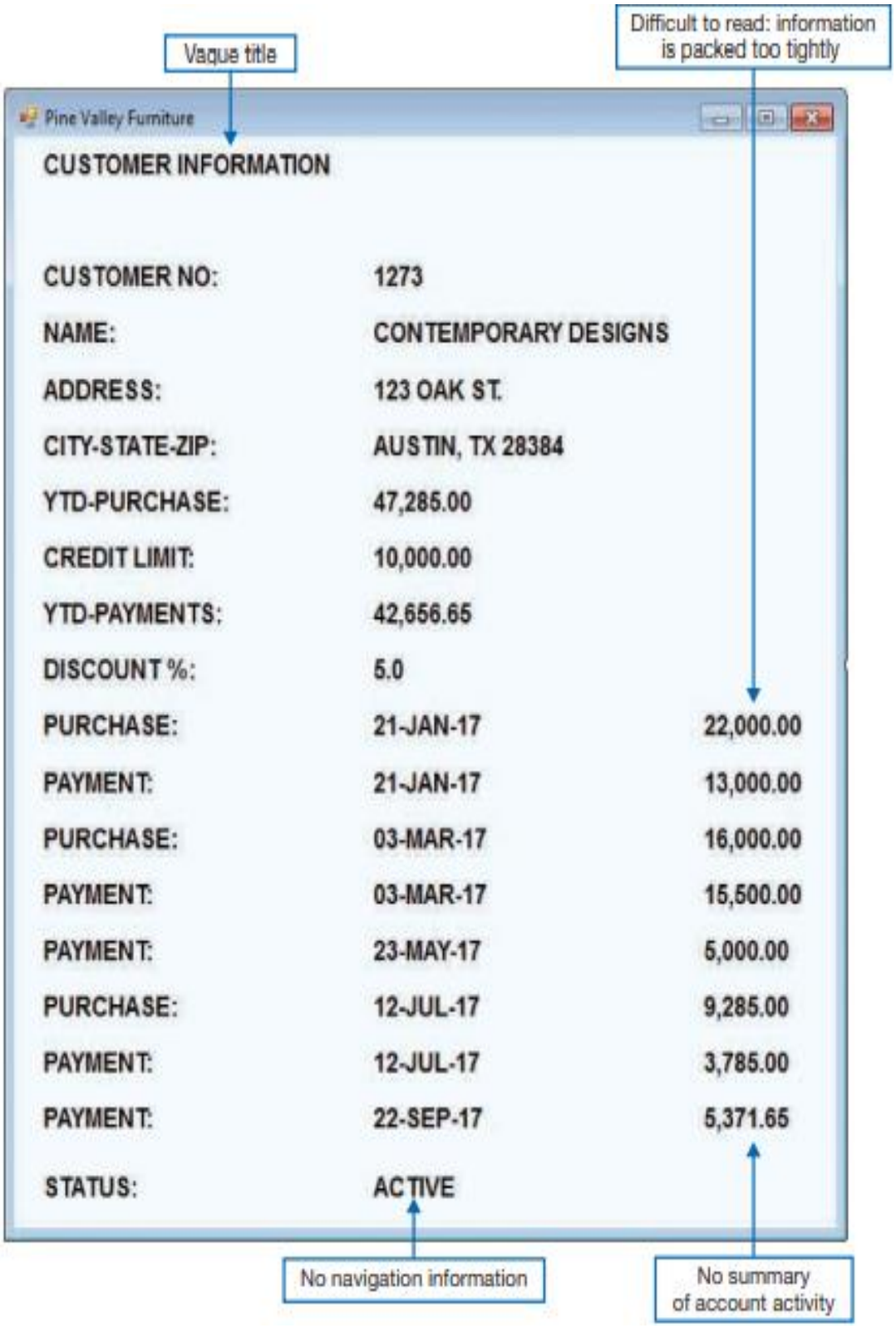
3. Formatting Forms and Reports

Guidelines:

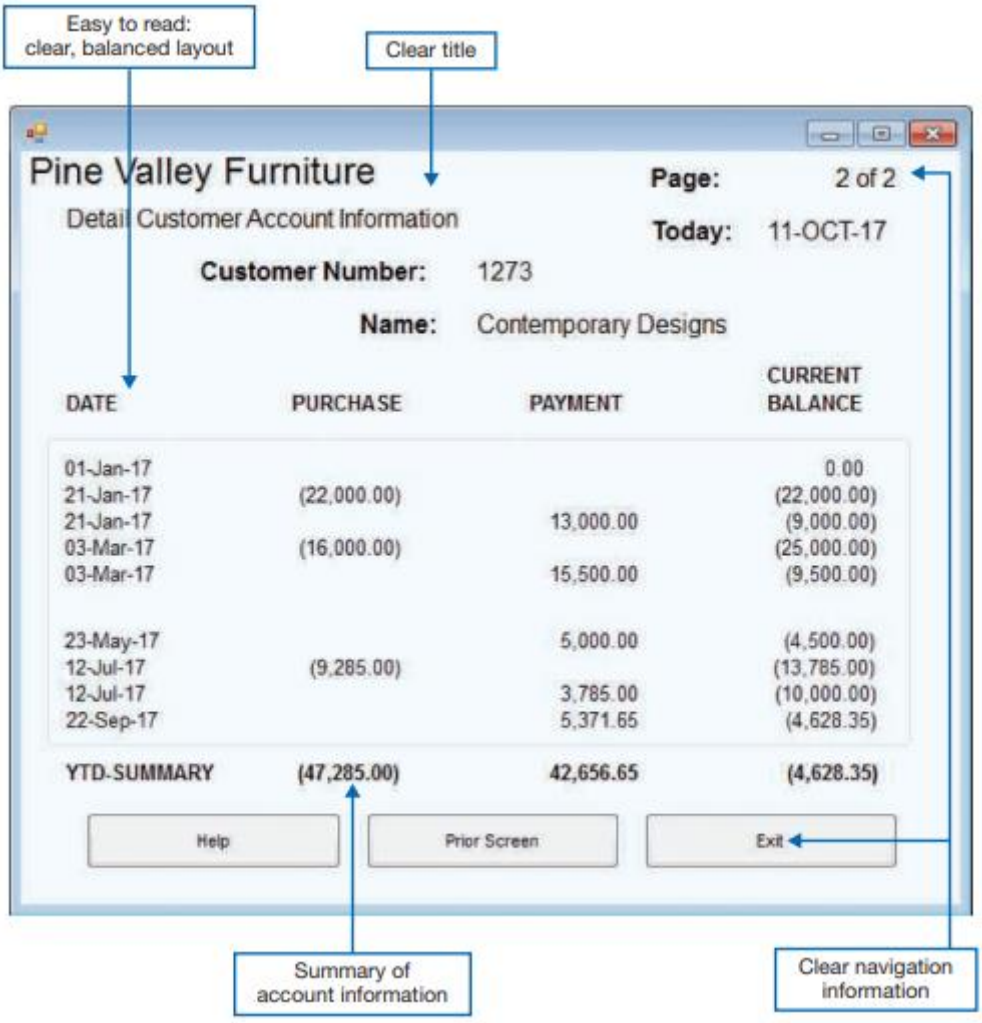
- **Meaningful titles:** clear, specific, version information, current date
- **Meaningful information:** include only necessary information
- **Balanced layout:** adequate spacing, margins, and clear labels
- **Easy navigation system:** show how to move forward and backward, and where you are currently

General Formatting Guidelines

Poorly Designed form



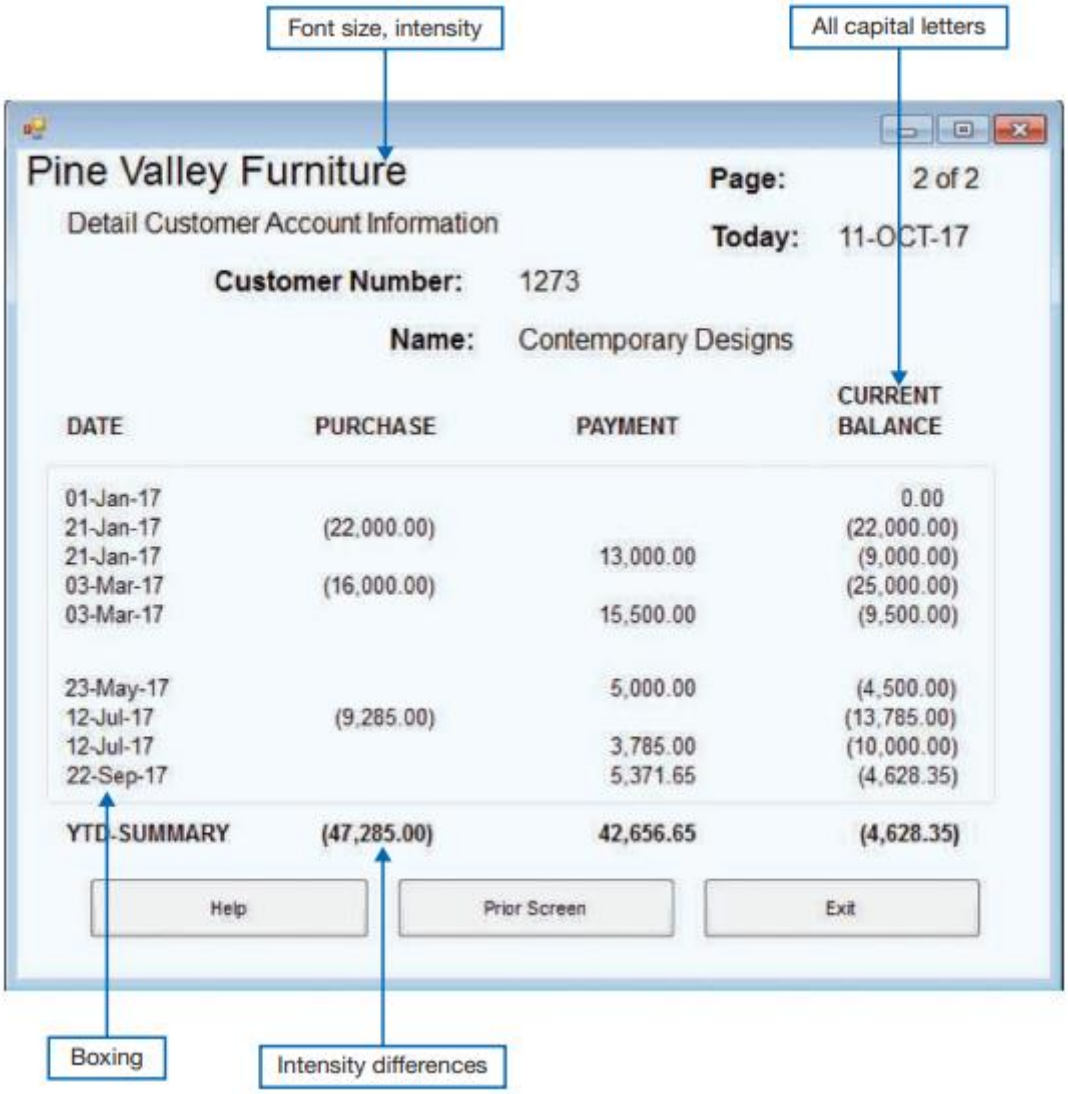
Improved design of Form



Highlighting Information

There are several situations when highlighting can be a valuable technique for conveying special information:

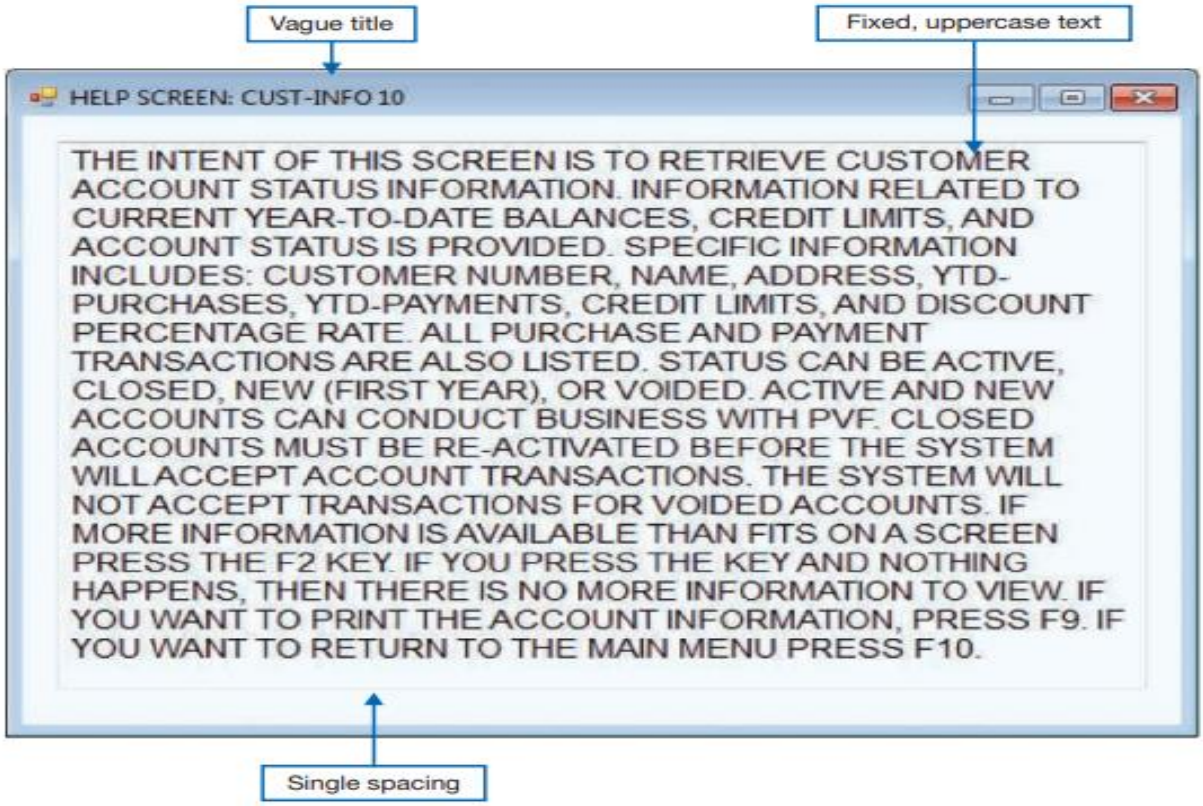
- Notifying users of errors in data entry or processing
- Providing warnings to users regarding possible problems such as unusual data values or an unavailable device
- Drawing attention to keywords, commands, high-priority messages, and data that have changed or gone outside normal operating ranges



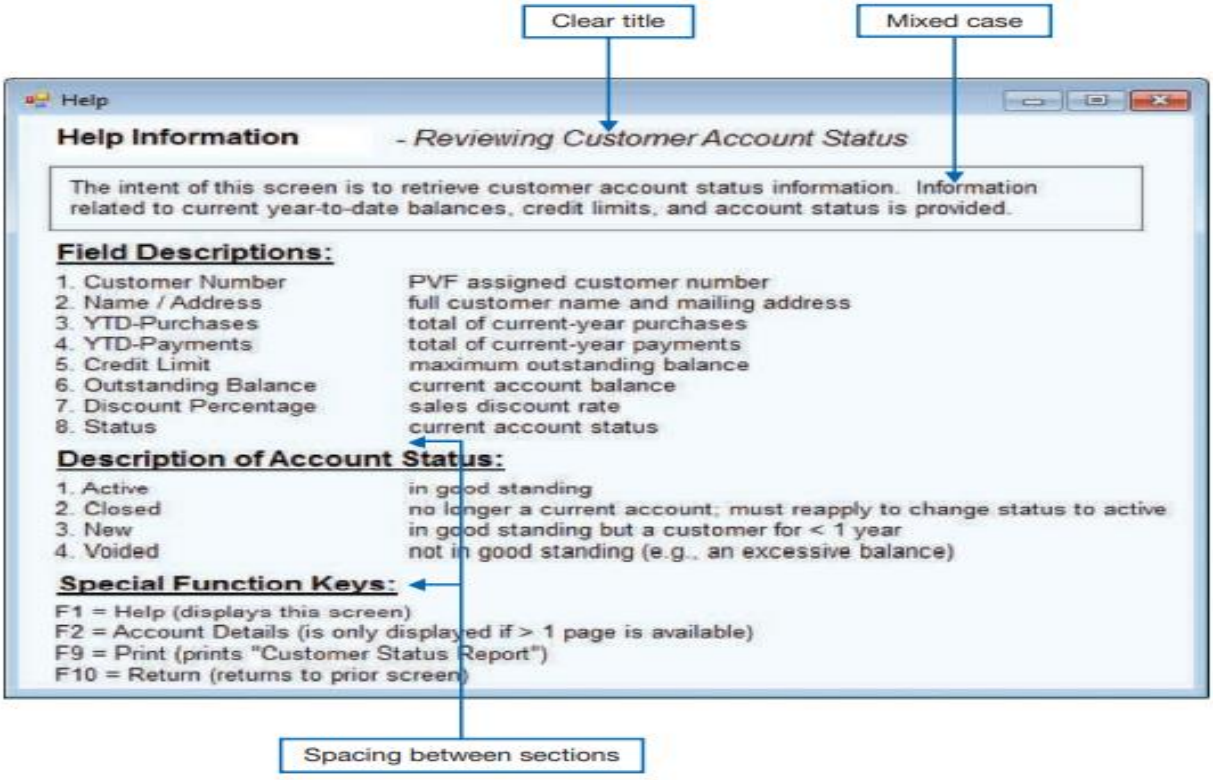
Guidelines for Displaying Text

Case	Display text in mixed uppercase and lowercase and use conventional punctuation.
Spacing	Use double spacing if space permits. If not, place a blank line between paragraphs.
Justification	Left-justify text and leave a ragged-right margin.
Hyphenation	Do not hyphenate words between lines.
Abbreviations	Use abbreviations and acronyms only when they are widely understood by users and are significantly shorter than the full text.

Poor Design for displaying Text



Improved design for displaying Text



Color vs. No Color

Benefits - Color:

- Soothes or strikes the eye.
- Accents an uninteresting display.
- Facilitates subtle discriminations in complex displays.
- Emphasizes the logical organization of information.
- Draws attention to warnings.
- Evokes more emotional reactions.

Problems from Using Color

- Color pairings may wash out or cause problems for some users.
- Resolution may degrade with different displays.
- Color fidelity may degrade on different displays.
- Printing or conversion to other media may not easily translate.

Designing Tables and Lists

Labels

- All columns and rows should have meaningful labels.
- Labels should be separated from other information by using highlighting.
- Redisplay labels when the data extend beyond a single screen or page.

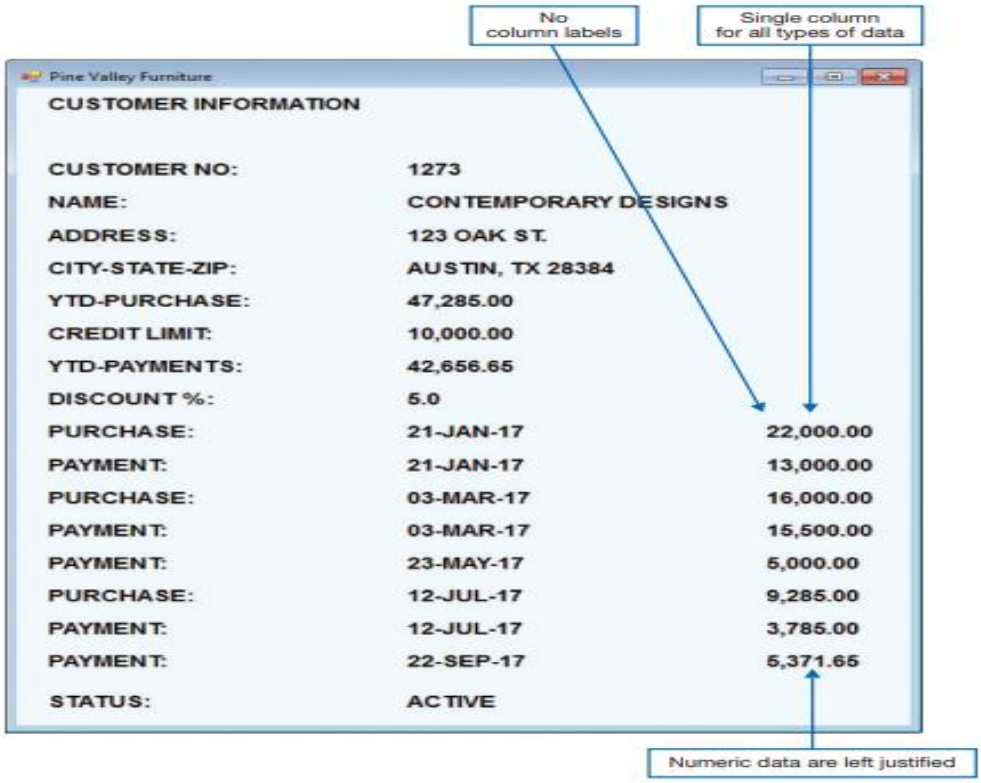
Formatting columns, rows and text:

- Sort in a meaningful order.
- Place a blank line between every five rows in long columns.
- Similar information displayed in multiple columns should be sorted vertically.
- Columns should have at least two spaces between them.
- Allow white space on printed reports for user to write notes.
- Use a single typeface, except for emphasis.
- Use same family of typefaces within and across displays and reports.
- Avoid overly fancy fonts.

Formatting numeric, textual and alphanumeric data:

- Right justify numeric data and align columns by decimal points or other delimiter.
- Left justify textual data. Use short line length, usually 30 to 40 characters per line.
- Break long sequences of alphanumeric data into small groups of three to four characters each.

Poorly Design form



Improved Design form

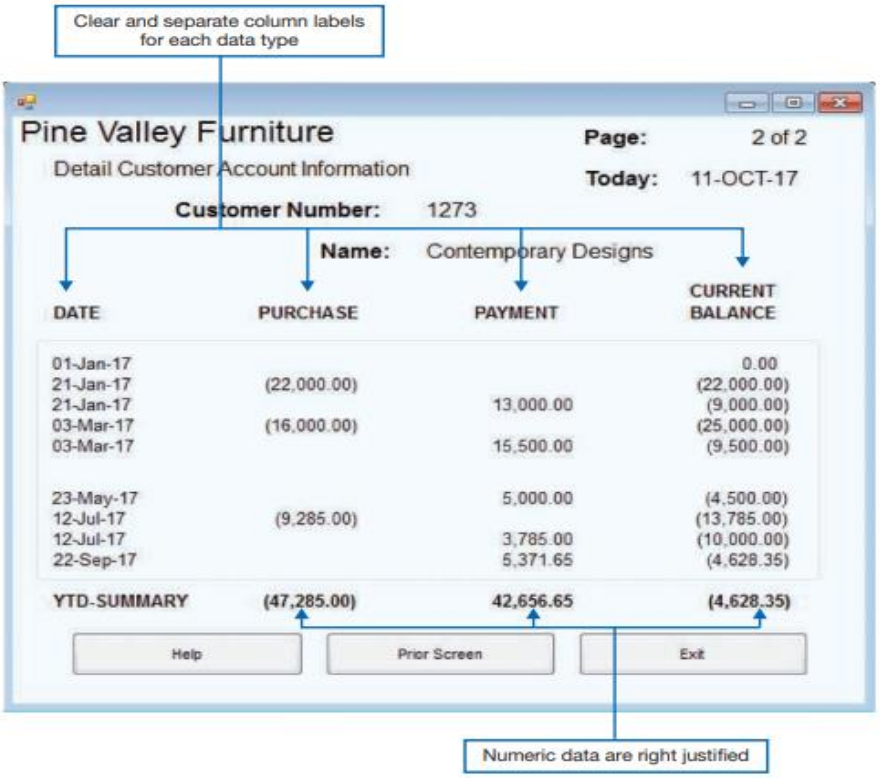
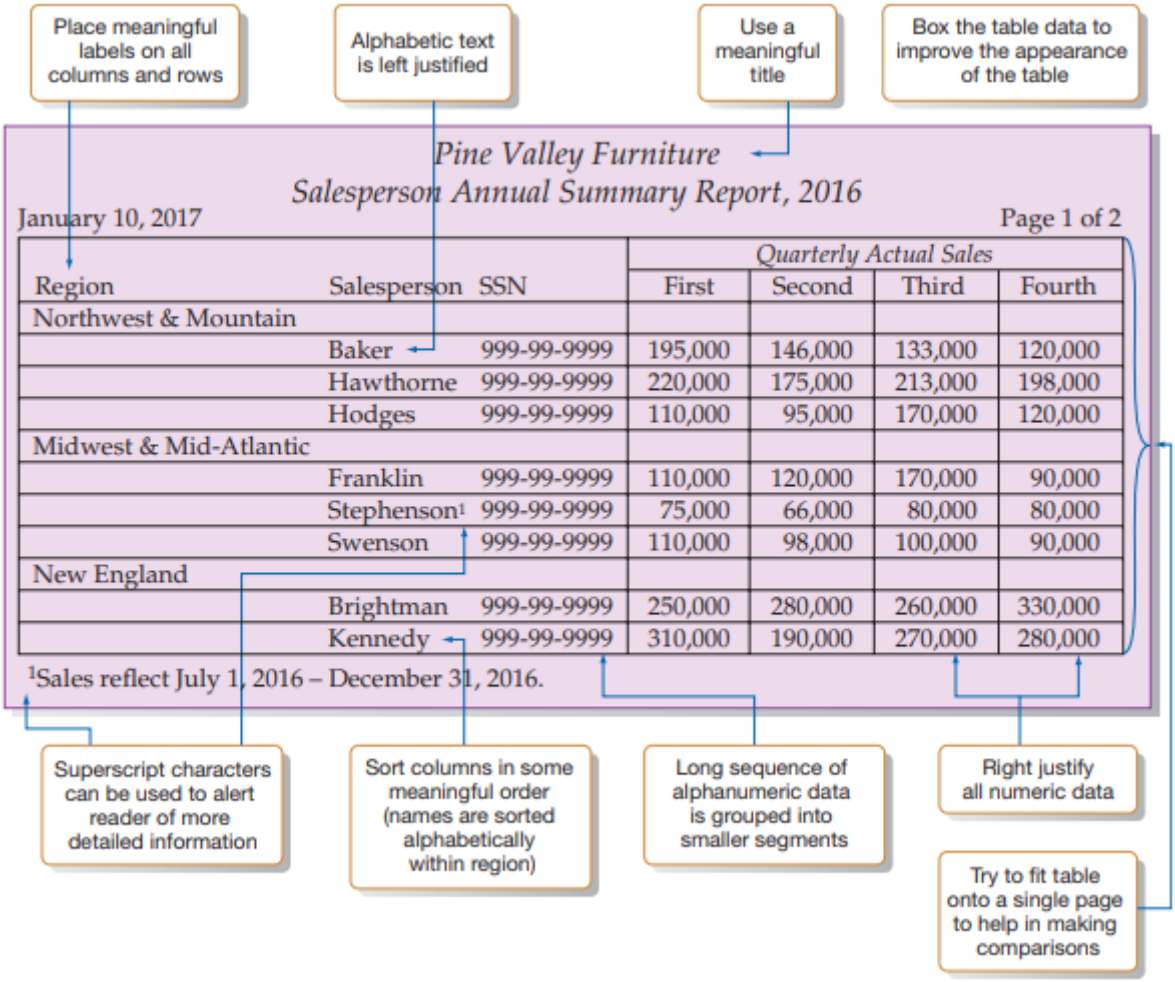


Table Vs Graph

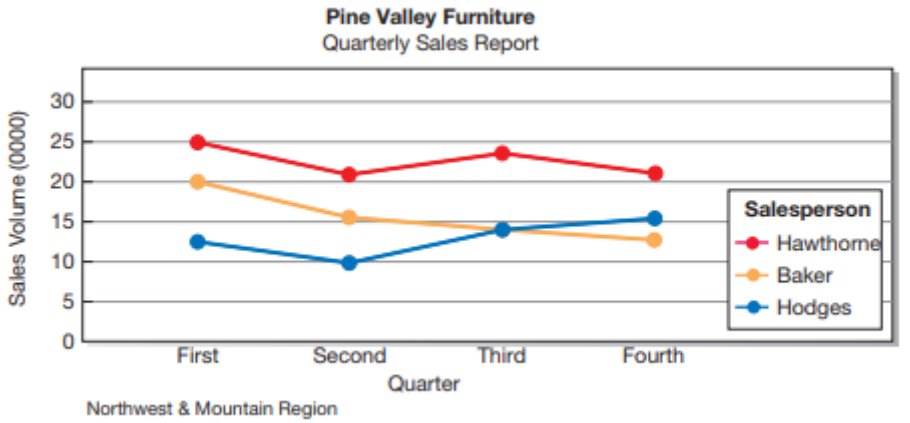
Use tables for reading individual data values
Use graphs for:

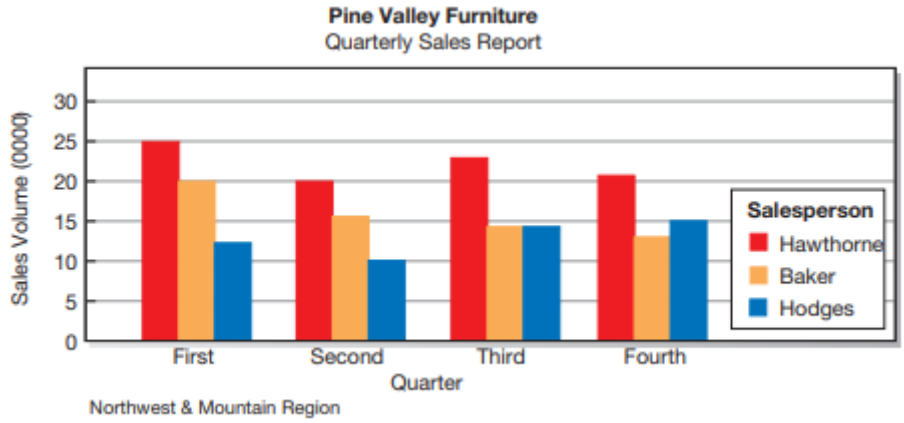
- Providing quick summary
- Displaying trends over time
- Comparing points and patterns of variables
- Forecasting activity
- Simple reporting of vast quantities of information

Tables



Graphs





4. Assessing Usability

Overall evaluation of how a system performs in supporting a particular user for a particular task.
Objective for designing forms, reports and all human-computer interactions is usability.
There are three **characteristics:**

- **Speed** - Can you complete a task efficiently?
- **Accuracy** - Does the output provide what you expect?
- **Satisfaction** - Do you like using the output?

Guidelines for Maximizing Usability

Usability Factor	Guidelines for Achievement of Usability
Consistency	Consistent use of terminology, abbreviations, formatting, titles, and navigation within and across outputs. Consistent response time each time a function is performed.
Organization	Formatting should be designed with an understanding of the task being performed and the intended user. Text and data should be aligned and sorted for efficient navigation and entry. Entry of data should be avoided where possible (e.g., computing rather than entering totals).
Clarity	Outputs should be self-explanatory and not require users to remember information from prior outputs in order to complete tasks. Labels should be extensively used, and all scales and units of measure should be clearly indicated.
Format	Information format should be consistent between entry and display. Format should distinguish each piece of data and highlight, not bury, important data. Special symbols, such as decimal places, dollar signs, and ± signs, should be used as appropriate.
Flexibility	Information should be viewed and retrieved in a manner most convenient to the user. For example, users should be given options for the sequence in which to enter or view data and for use of shortcut keystrokes, and the system should remember where the user stopped during the last use of the system.

Characteristics for Consideration When Designing Forms and Reports

Characteristic	Consideration for Form and Report Design
User	Issues related to experience, skills, motivation, education, and personality should be considered.
Task	Tasks differ in amount of information that must be obtained from or provided to the user. Task demands such as time pressure, cost of errors, and work duration (fatigue) will influence usability.
System	The platform on which the system is constructed will influence interaction styles and devices.
Environment	Social issues such as the users' status and role should be considered in addition to environmental concerns such as lighting, sound, task interruptions, temperature, and humidity. The creation of usable forms and reports may necessitate changes in the users' physical work facilities.

Common errors when designing the layout of Web Pages

Error	Recommendation
Nonstandard Use of GUI Widgets	Make sure that when using standard design items, they behave in accordance with major interface design standards. For example, the rules for radio buttons state that they are used to select one item among a set of items, that is, not confirmed until "OK'ed" by a user. In many websites selecting radio buttons is used as both *selection* and *action*.
Anything That Looks Like Advertising	Because research on web traffic has shown that many users have learned to stop paying attention to web advertisements, make sure that you avoid designing any legitimate information in a manner that resembles advertising (e.g., banners, animations, pop-ups).
Bleeding-Edge Technology	Make sure that users don't need the latest browsers or plug-ins to view your site.
Scrolling Test and Looping Animations	Avoid scrolling text and animations because they are both hard to read and users often equate such content with advertising.
Nonstandard Link Colors	Avoid using nonstandard colors to show links and for showing links that users have already used; nonstandard colors will confuse the user and reduce ease of use.
Outdated Information	Make sure your site is continuously updated so that users "feel" that the site is regularly maintained and updated. Outdated content is a sure way to lose credibility.
Slow Download Times	Avoid using large images, lots of images, unnecessary animations, or other time-consuming content that will slow the downloading time of a page.
Fixed-Formatted Text	Avoid fixed-formatted text that requires users to scroll horizontally to view content or links
Displaying Long Lists as Long Pages	Avoid requiring users to scroll down a page to view information, especially navigational controls. Manage information by showing only *N* items at a time, using multiple pages, or by using a scrolling container within the window.

Good Web Design Practices

Lightweight Graphics: small images to quick image download

Forms and Data Integrity

Template-based HTML:

- Templates to display and process common attributes of higher-level, more abstract items
- Creates an interface that is very easy to maintain

C. Designing Interfaces and Dialogues

1. Introduction

Designing Interfaces and Dialogues

- Interface design focuses on how information is provided to and from users
- Dialog design focuses on sequencing of interface display
- The design of interface and dialogues is the process of defining the manner in which human and computers
- Exchange information
- Similar to designing f&r, the process of designing i&d is user focused activity (prototyping methodology of iterative collecting information, constructing, assessing usability, and making refinements)
- To design usable i&d you must answer the same who, what, when, where and how
- Design specification had 3 section, narrative overview, sample design and testing & usability assessment
- Focus on how information is provided to and captured from users
- Dialogues are analogous to a conversation between two people
- A good human-computer interface provides a unifying structure for finding, viewing and invoking the different components of a system

The Process of Designing Interfaces and Dialogues

- User-focused activity
- Parallels form and report design process
- Employs prototyping methodology:
    - Collect information
    - Construct prototype
    - Assess usability
    - Make refinements

Design Specification:

- Narrative overview
- Sample design
- Testing and usability assessment

Design Specification

1. Narrative Overview
   a. Interface/Dialogue Name
   b. User Characteristics
   c. Task Characteristics
   d. System Characteristics
   e. Environmental Characteristics

2. Interface/Dialogue Designs
   a. Form/Report Designs
   b. Dialogue Sequence Diagram(s) and Narrative Description

3. Testing and Usability Assessment
   a. Testing Objectives
   b. Testing Procedures
   c. Testing Results
      i) Time to Learn
      ii) Speed of Performance
      iii) Rate of Errors
      iv) Retention over Time
      v) User Satisfaction and Other Perceptions

2. Interaction Methods and Devices

Methods of Interacting

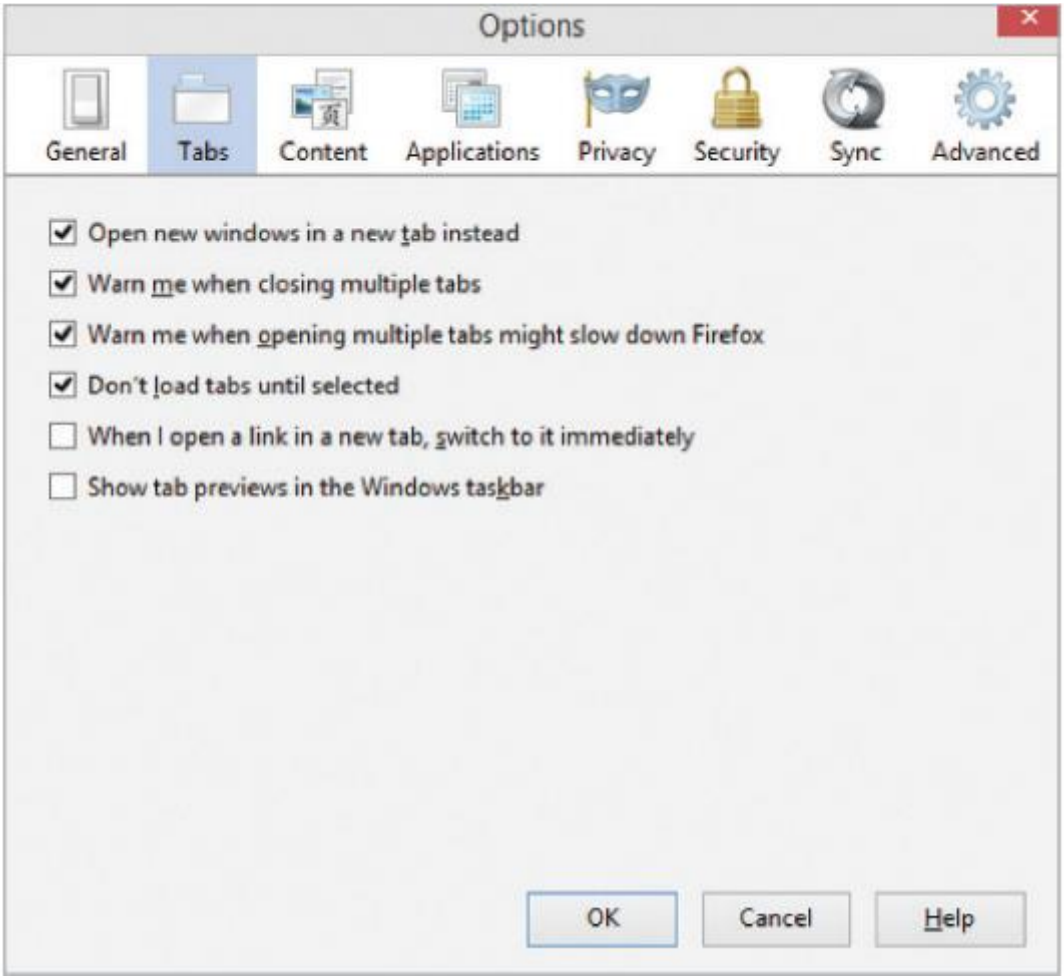
Interface: The method by which a user interacts with the information system

a. Command Line Interaction

- Users enter explicit statements into a system to invoke operations
- Example from MS DOS:
  - COPY C:PAPER.DOC A:PAPER.DOC
  - This copies a file from the C: drive to the A: drive
- Includes keyboard shortcuts and function keys
- Experienced users and for rapid interaction with a system
- User interface standards

b. Menu Interaction

- A list of system options is provided and specific command is invoked by user selection of a menu option
- Two common menu types:
    - Pop-up: menu placed near current cursor position and list of commands or possible values
    - Drop-down: access point to menu placed at top line of display, menu drops down when access point clicked

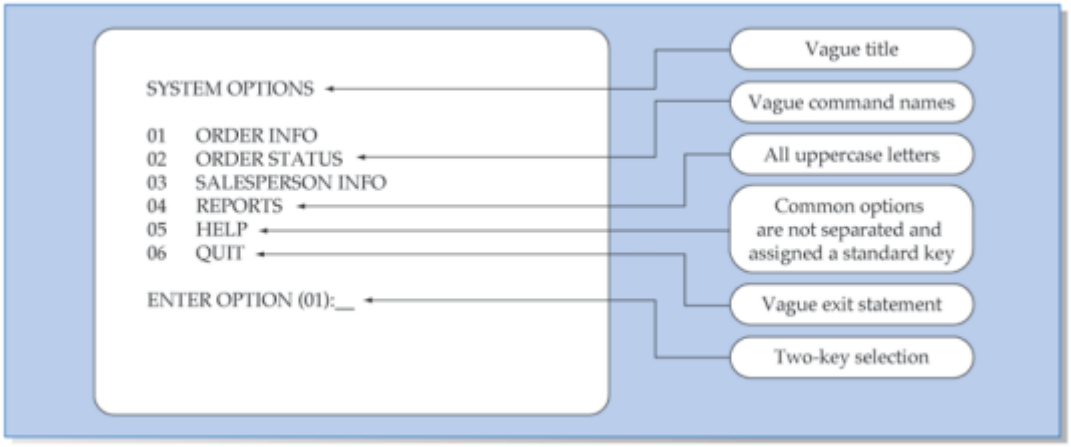


Guidelines for Menu Design

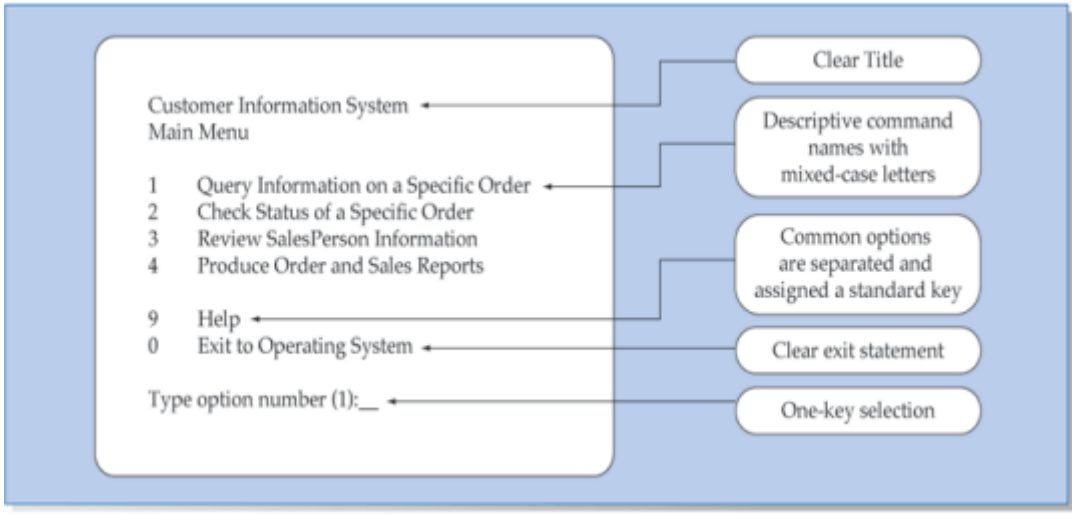
- Wording: meaningful titles, clear command verbs, mixed upper/lower case
    - Quit → prior menu or exit program?
- Organization: consistent organizing principle
    - Related options grouped together
    - Same option should have the same wording
- Length: all choices fit within screen length
    - Use submenus to break up exceedingly long menus
    -

- Selection: consistent, clear and easy selection methods
    - How to select and the consequences of each option - will another menu appear?
- Highlighting: only for selected options (check mark) or unavailable options (dimmed text)
- Use menu building tools

Poor Menu

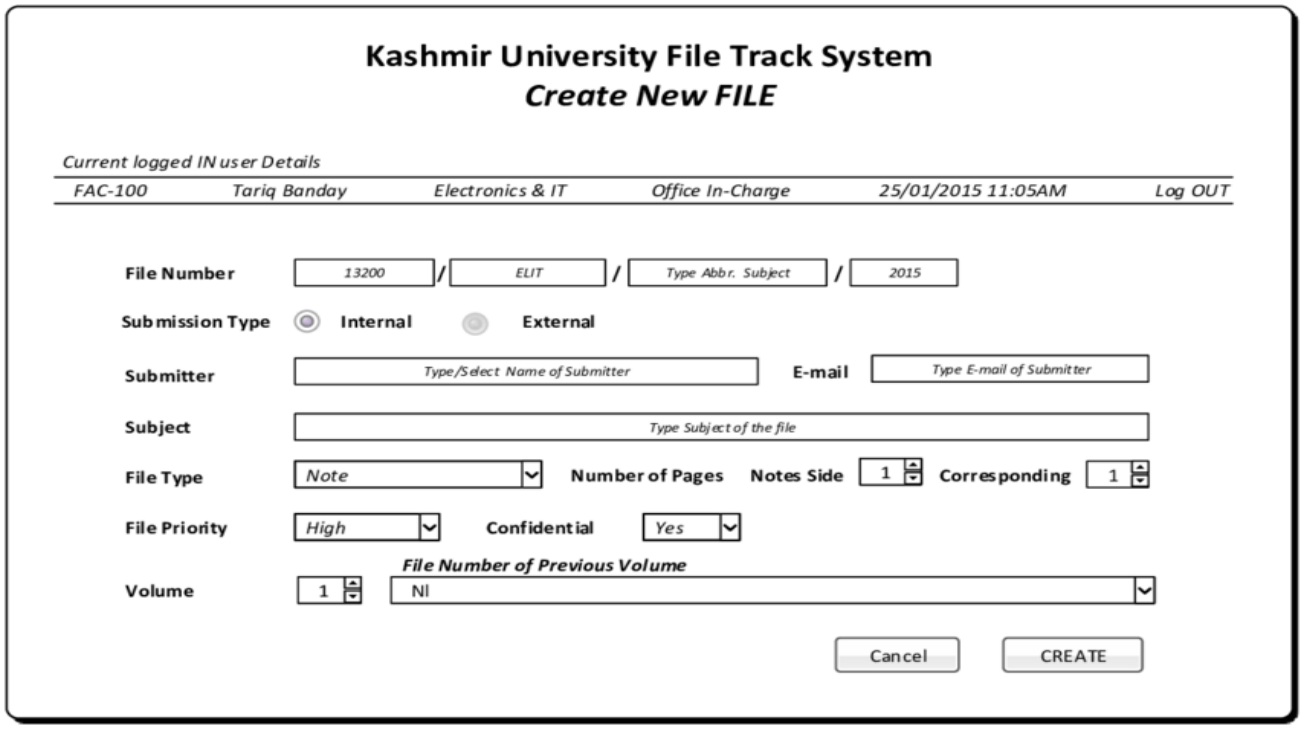


Good Menu



c. Form Interaction

- Allows users to fill in the blanks when working with a system
- Measures of an effective design:
    - Self-explanatory title and field headings
    - Fields organized into logical groupings
    - Distinctive boundaries
    - Default values
    - Displays appropriate field lengths
    - Minimizes the need to scroll windows



d. Object Interaction

- Symbols are used to represent commands or functions.
- Icons:
    - Graphic symbols that look like the processing option they are meant to represent
    - Use little screen space
    - Can be easily understood by users

e. Natural Language Interaction

- Inputs to and outputs from system are in a conventional speaking language like English
- Based on research in artificial intelligence
- Current implementations are tedious and difficult to work with, not as viable as other interaction methods

Hardware options for System Interaction

Common Devices for Interacting with an Information System

Device	Description and Primary Characteristics or Usage
Keyboard	Users push an array of small buttons that represent symbols that are then translated into words and commands. Keyboards are widely understood and provide considerable flexibility for interaction.
Mouse	A small plastic box that users push across a flat surface and whose movements are translated into cursor movement on a computer display. Buttons on the mouse tell the system when an item is selected. A mouse works well on flat desks but may not be practical in dirty or busy environments, such as a shop floor or check-out area in a retail store. Newer pen-based mice provide the user with more of the feel of a writing implement.
Joystick	A small vertical lever mounted on a base that steers the cursor on a computer display. Provides similar functionality to a mouse.
Trackball	A sphere mounted on a fixed base that steers the cursor on a computer display. A suitable replacement for a mouse when work space for a mouse is not available.
Touch Screen	Selections are made by touching a computer display. This works well in dirty environments or for users with limited dexterity or expertise.
Light Pen	Selections are made by pressing a pen-like device against the screen. A light pen works well when the user needs to have a more direct interaction with the contents of the screen.
Graphics Tablet	Moving a pen-like device across a flat tablet steers the cursor on a computer display. Selections are made by pressing a button or by pressing the pen against the tablet. This device works well for drawing and graphical applications.
Voice	Spoken words are captured and translated by the computer into text and commands. This is most appropriate for users with physical challenges or when hands need to be free to do other tasks while interacting with the application.

Summary of Interaction Device Usability Problems

	Problem						
Device	Visual Blocking	User Fatigue	Movement Scaling	Durability	Adequate Feedback	Speed	Pointing Accuracy
Keyboard	☐	☐	■	☐	■	■	☐
Mouse	☐	☐	■	☐	■	☐	☐
Joystick	☐	☐	■	☐	■	☐	■
Trackball	☐	☐	■	■	■	☐	☐
Touch Screen	■	■	☐	■	☐	☐	■
Light Pen	■	■	☐	☐	☐	☐	■
Graphics Tablet	☐	☐	■	☐	■	☐	☐
Voice	☐	☐	■	☐	■	☐	■

Key:

☐ = little or no usability problems

■ = potentially high usability problems for some applications

- Visual Blocking = extent to which device blocks display when using
- User Fatigue = potential for fatigue over long use
- Movement Scaling = extent to which device movement translates to equivalent screen movement

- Durability = lack of durability or need for maintenance (e.g., cleaning) over extended use
- Adequate Feedback = extent to which device provides adequate feedback for each operation
- Speed = cursor movement speed
- Pointing Accuracy = ability to precisely direct cursor

Summary of General Conclusions from experimental Comparisons of Input Devices in Relation to Specific Task activities

Task	Most Accurate	Shortest Positioning	Most Preferred
Target Selection	trackball, graphics tablet, mouse, joystick	touch screen, light pen, mouse, graphics tablet, trackball	touch screen, light pen
Text Selection	mouse	mouse	—
Data Entry	light pen	light pen	—
Cursor Positioning	—	light pen	—
Text Correction	light pen, cursor keys	light pen	light pen
Menu Selection	touch screen	—	keyboard, touch screen

Key:

- Target Selection = moving the cursor to select a figure or item
- Text Selection = moving the cursor to select a block of text
- Data Entry = entering information of any type into a system
- Cursor Positioning = moving the cursor to a specific position
- Text Correction = moving the cursor to a location to make a text correction
- Menu Selection = activating a menu item

— = no clear conclusion from the research

3. Designing Interfaces and Dialogues in Graphical Environments

Designing Interfaces

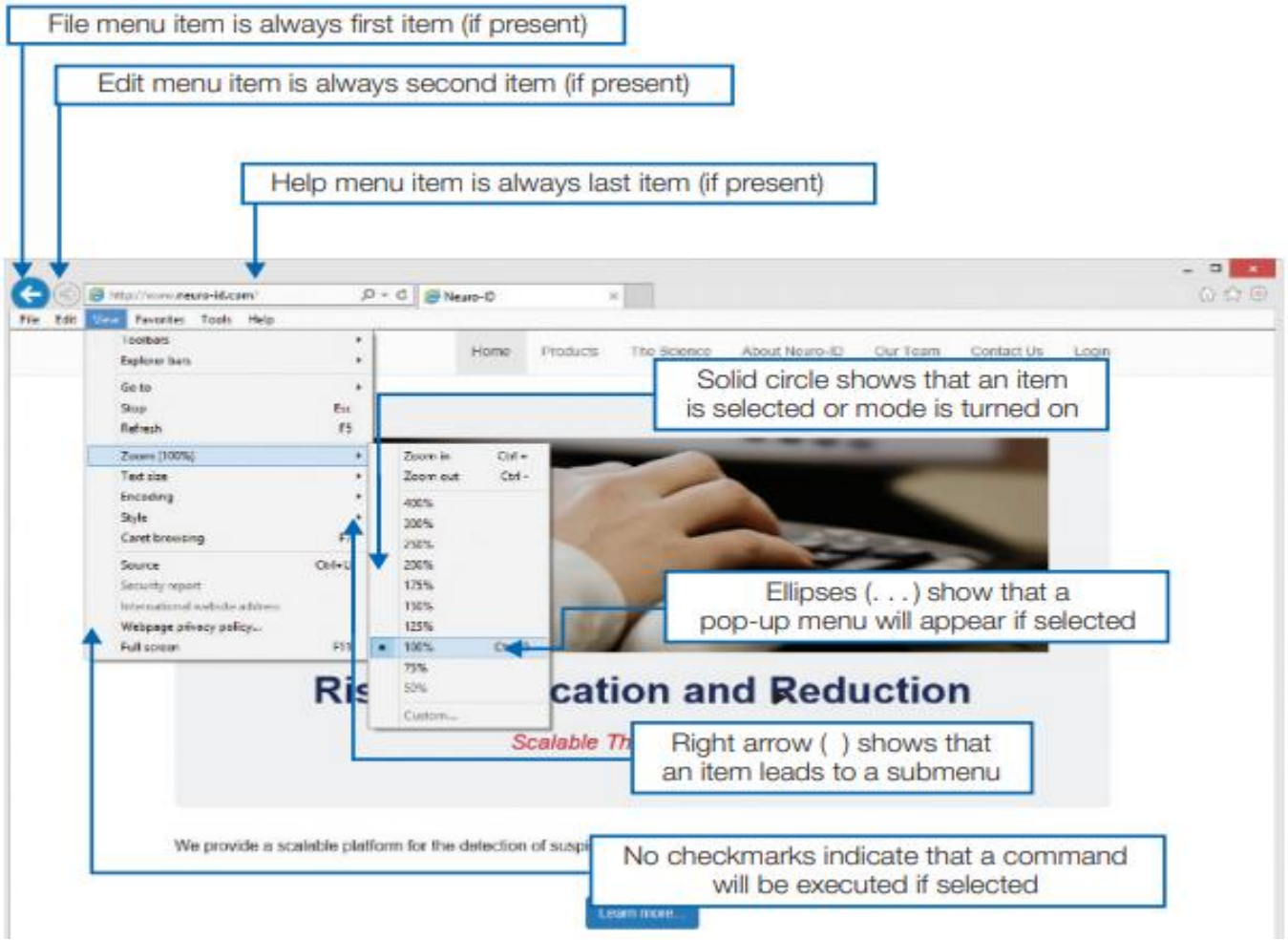
In most discussions of GUI programming, two rules repeatedly emerge as composing the first step to becoming an effective GUI designer:

1 Become an expert user of the GUI environment.
2 Understand the available resources and how they can be used.

Common Properties of Windows and Forms in a GUI environment That Can be active or Inactive

Property	Explanation
Modality	Requires users to resolve the request for information before proceeding (e.g., need to cancel or save before closing a window)
Resizable	Allows users to resize a window or form (e.g., to make room to see other windows that are also on the screen)
Movable	Allows users to move a window or form (e.g., to allow another window to be seen)
Maximize	Allows users to expand a window or form to a full-size screen (e.g., to avoid distraction from other active windows or forms)
Minimize	Allows users to shrink a window or form to an icon (e.g., to get the window out of the way while working on other active windows)
System Menu	Allows a window or form to also have a system menu to directly access system-level functions (e.g., to save or copy data)

GUI Design Interface



Designing Dialogues

- Dialogue: Sequence in which information is displayed to and obtained from a user
- Primary design guideline is consistency in sequence of actions, keystrokes and terminology
- Three step process
    1. Design dialogue sequence
    2. Build a prototype
    3. Assess usability

Dialogue Design Issues

- Goal is to establish the sequence of displays that users will encounter when working with system
- Ability of some GUI environments to jump from application to application or screen to screen makes sequencing a challenge
- One approach is to make users always resolve requests for information before proceeding
- Dialogue diagramming helps analysts better manage the complexity of designing graphical interfaces

Common errors when designing the Interface and Dialogues of Websites

Error	Description
Opening New Browser Window	Avoid opening a new browser window when a user clicks on a link unless it is clearly marked that a new window will be opened; users may not see that a new window has been opened, which will complicate navigation, especially moving backward.
Breaking or Slowing Down the Back Button	Make sure users can use the back button to return to prior pages. Avoid opening new browser windows, using an immediate redirect where, when users click the back button, they are pushed forward to an undesired location, or prevent caching such that each click of the back button requires a new trip to the server.
Complex URLs	Avoid overly long and complex URLs because it makes it more difficult for users to understand where they are and can cause problems if users want to e-mail page locations to colleagues.
Orphan Pages	Avoid having pages with no "parent" that can be reached by using a back button; requires users to "hack" the end of the URL to get back to some other prior page.
Scrolling Navigation Pages	Avoid placing navigational links below where a page opens because many users may miss these important options that are below the opening window.
Lack of Navigation Support	Make sure your pages conform to users' expectations by providing commonly used icon links such as a site logo at the top or other major elements. Also place these elements on pages in a consistent manner.
Hidden Links	Make sure you leave a border around images that are links, don't change link colors from normal defaults, and avoid embedding links within long blocks of text.
Links That Don't Provide Enough Information	Avoid not turning off link-marking borders so that links clearly show which links users have clicked and which they have not. Make sure users know which links are internal anchor points versus external links, and indicate if a link brings up a separate browser window from those that do not. Finally, make sure link images and text provide enough information to users so that they understand the meaning of the link.
Buttons That Provide No Click Feedback	Avoid using image buttons that don't clearly change when being clicked; use web GUI toolkit buttons, HTML form-submit buttons, or simple textual links.