UNIT 4
Design

LH- 12 HRS

Er. Rolisha Sthapit

SYSTEM ANALYSIS AND DESIGN (SAD)

Contents

4.1 Designing Databases

4.1.1Introduction, 4.1.2Database Design, 4.1.3Relational Database Model, 4.1.4Normalization, 4.1.5Transforming E R Diagrams Into Relations, 4.1.6Merging Relations, 4.1.7Physical File And Database Design, 4.1.8Designing Fields, 4.1.9Designing Physical Tables

4.2 Designing Forms And Reports

4.2.1 Introduction, 4.2.2Designing Forms And Reports, 4.2.3Formatting Forms And Reports, 4.2.4Assessing Usability

4.3 Designing Interfaces and Dialogues

4.3.1 Introduction, 4.3.2 Designing Interfaces And Dialogues, 4.3.3 Interaction Methods And Devices, 4.3.4 Designing Interfaces And Dialogues In Graphical Environments

4.1 Designing Databases

4.1.1Introduction,

4.1.2Database Design,

4.1.3Relational Database Model,

4.1.4Normalization,

4.1.5Transforming E R Diagrams Into Relations,

4.1.6Merging Relations,

4.1.7Physical File And Database Design,

4.1.8Designing Fields,

4.1.9Designing Physical Tables

4.1 Designing Database

4.1.1 Introduction

- System development is systematic process which includes phases such as planning, analysis, design, deployment and maintenance. Design is a process of planning a new business system or replacing an existing system by defining in components or modules to satisfy the specific requirements.
- Before planning, we need to understand the old system thoroughly and determine how computer can best be used in order to operate efficiently.
- It is the phase that bridges the gap between problems domain and the existing system in a manageable way. This phase focuses on the solutions domain, i.e. “how to implement?”. It is the phase where the SRS document is converted into a format that can be implemented and decided how the system will operate.

Database design has five purposes:

1. Structure the data in stable structures, called normalized tables, that are not likely to change over time and that have minimal redundancy.

2. Develop a logical database design that reflects the actual data requirements that exist in the forms (hard copy and computer displays) and reports of an information system. This is why database design is often done in parallel with the design of the human interface of an information system.

3. Develop a logical database design from which we can do physical database design. Because most information systems today use relational database management systems, logical database design usually uses a relational database model, which represents data in simple tables with common columns to link related tables.

4. Translate a relational database model into a technical file and database design that balances several performance factors.

5. Choose data storage technologies (such as Read/ Write DVD or optical disc) that will efficiently, accurately and securely process database activities.

4.1.2Database Design

- It is the organization of data according to a database model. The designer determines what data must be stored and how the data elements interrelate.
- Database design is a collection of processes that facilitate the designing, development, implementation and maintenance of enterprise data management systems. It helps produce database systems:

✓ That meets the requirements of the users

✓ Have high performance.

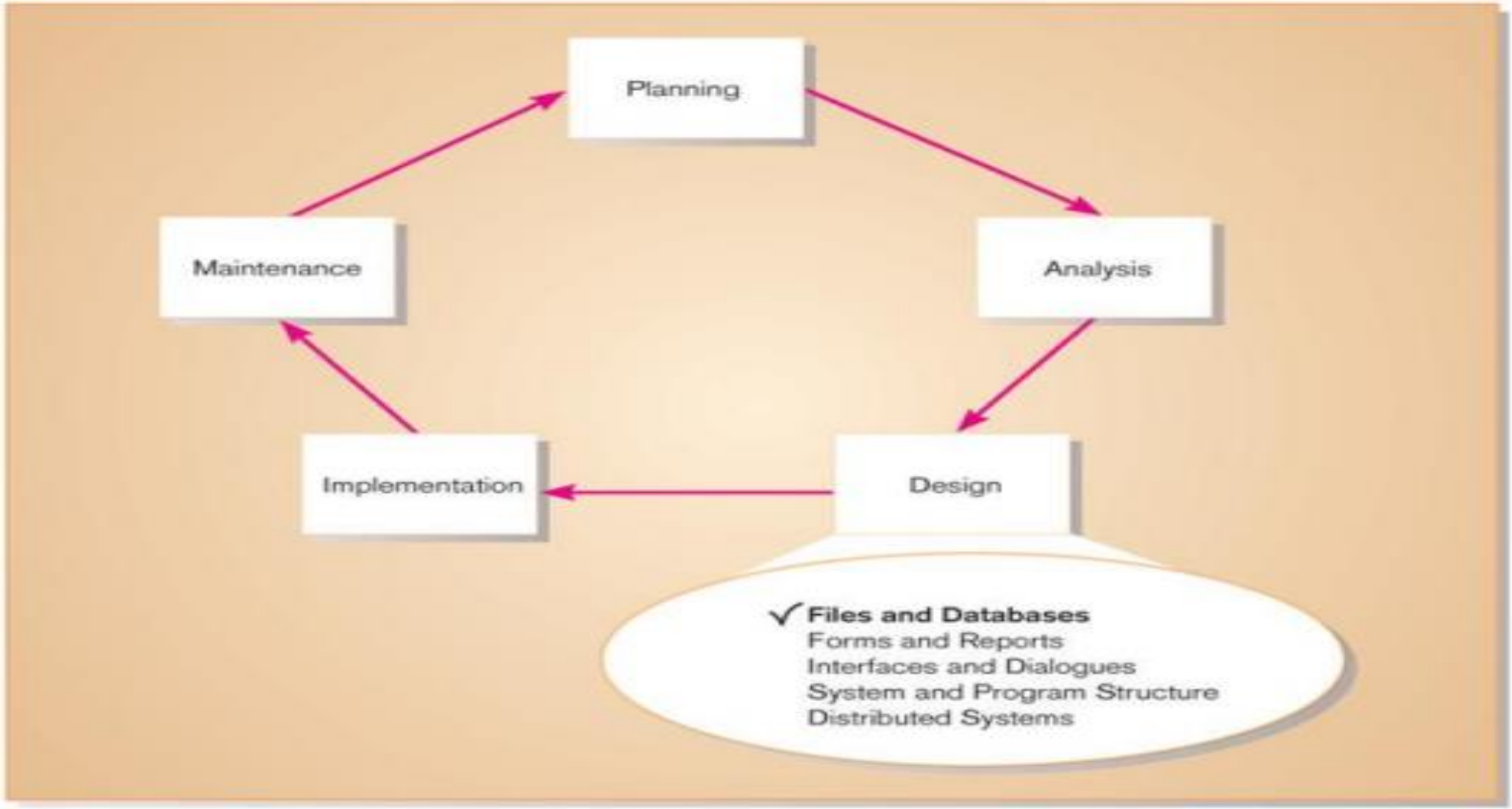
✓ The main objectives of database designing are to produce logical and physical designs models of the proposed database system.

✓ The logical model concentrates on the data requirements and the data to be stored independent of physical consideration.

✓ The physical data design model involves translating the logical design of the database onto physical media using hardware resources and software systems such as database management systems (DBMS).

- **Planning – Analysis – Design (Databases: Forms and reports, Dialogues & interfaces, Finalizing design specification) – Implementation – Maintenance**

Figure 10-1 Systems development life cycle with design phase highlighted



The Process of Database Design

- In logical database design, we use a process called normalization, which is a way to build a data model that has the properties of simplicity, non-redundancy, and minimal maintenance.
- Enterprise wide data model (ER with only attributes)
- Conceptual data model (ER with only attributes for specific project)

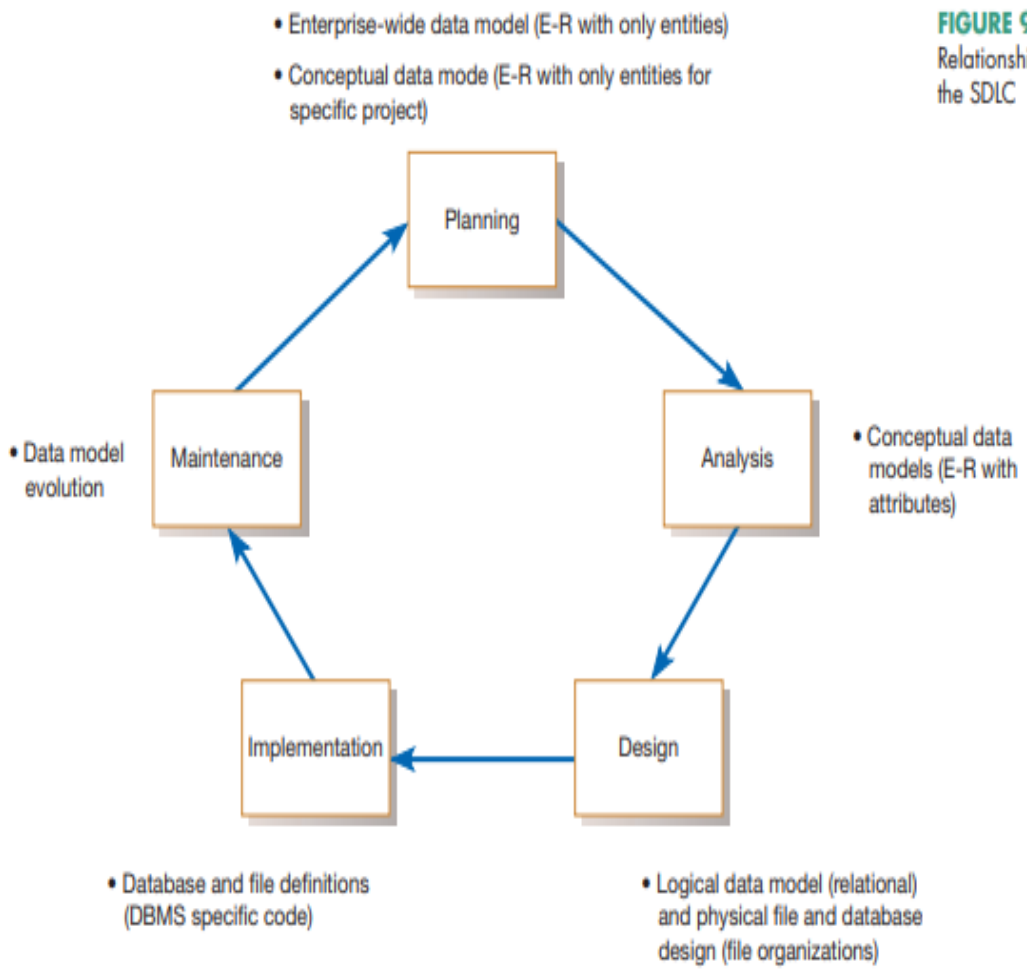


FIGURE 9-2
Relationship between data modeling and the SDLC

- In most situations, many physical database design decisions are implicit or eliminated when we choose the data management technologies to use with the application. We concentrate on those decisions that will make most frequently and use Oracle to illustrate the range of physical database design parameters you must manage.
- There are four key steps in logical database modeling and design:

1. Develop a logical data model for each known user interface (from and report) for the application using normalization principles.

2. Combine normalized data requirements from all user interfaces into one consolidated logical database model; this step is called view integration.

3. Translate the conceptual E-R data model for the application or enterprise, developed without explicit consideration of specific user interfaces, into normalized data requirements.

4. Compare the consolidated logical database design with the translated E-R model and produce, through view integration, one final logical database model for the application.

Objectives of Database

The general theme behind a database is to handle information as an integrated whole. There is none of the artificiality that is normally embedded in separate file or applications. A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make information access easy, quick, inexpensive and flexible for the user. In database design, several specific objectives are considered:

1. Control Redundancy: redundant data occupies space and, therefore, is wasteful often give conflicting information too. A unique aspect of database design is storing data only once, which controls redundancy and improves system performance.

2. Ease of learning and use: A major feature of a user-friendly database package is how easy it is to learn and use. Related to this point is that a database can be modified without interfering with established ways of using the data.

3. Data independence: An important database objective is changing hardware and storage procedures or adding new data without having to rewrite application programs. The database should be “tunable” to improve performance without rewriting programs.

4. More information at low cost: Using, storing and modifying data at low cost are important. Although hardware prices are falling, software and programming costs are on the rise. This means that programming and software enhancements should be kept simple and easy to update.

5. Accuracy and integrity: The accuracy of a database ensures that data quality and content remain constant. Integrity controls detect data inaccuracies where they occur.

6. Recovery from failure: With multi-user access to a database, the system must recover quickly after it is down with no loss of transactions. This objective also helps maintain data accuracy and integrity.

7. Privacy and Security: For data to remain private, security measures must be taken to prevent unauthorized access. Database security means that data are protected from various forms of destruction; users must be positively identified and their actions monitored.

8. Performance: This objective emphasizes response time to inquiries suitable to the use of the data. How satisfactory the response time is depends on the nature of the user-database dialogue. For example, inquiries regarding airline seat availability should in handled in a few seconds. On the other extreme, inquiries regarding the total sale of a product over the past two weeks may be handled satisfactorily in 50 seconds.

Data Modeling

- Data modeling is the process of creating a data model for the data to be stored in a database. The data model is a conceptual representation of
  - Data objects
  - The associations between different data objects
  - The rules
- Data modeling helps in the visual representation of data and enforces business rules, regulatory compliances, and government policies on the data. Data model ensures consistency in naming conventions, default values, semantics, and security while ensuring quality of the data. Data model emphasizes on what data is needed and how it should be organized instead of what operations need to be performed on the data. Data model is like architect's building plan which helps to build a conceptual model and set the relationship between data items.

Uses of Data Model

The primary goals of using data model are:

- Ensures that all data objects required by the database are accurately represented.
- Omission of data will lead to creation of faulty reports and produce incorrect results.
- A data model helps design the database at the conceptual, physical and logical levels.
- Data model structure helps to define the relational tables, primary and foreign keys and stored procedures.
- It provides a clear picture of the base data and can be used by database developers to create a physical database.
- It is also helpful to identify missing and redundant data.
- Though the initial creation of data model is labor and time consuming, in the long run, it makes user IT infrastructure upgrade and maintenance cheaper and faster.

Types of Data Models

There are mainly three types of data models:

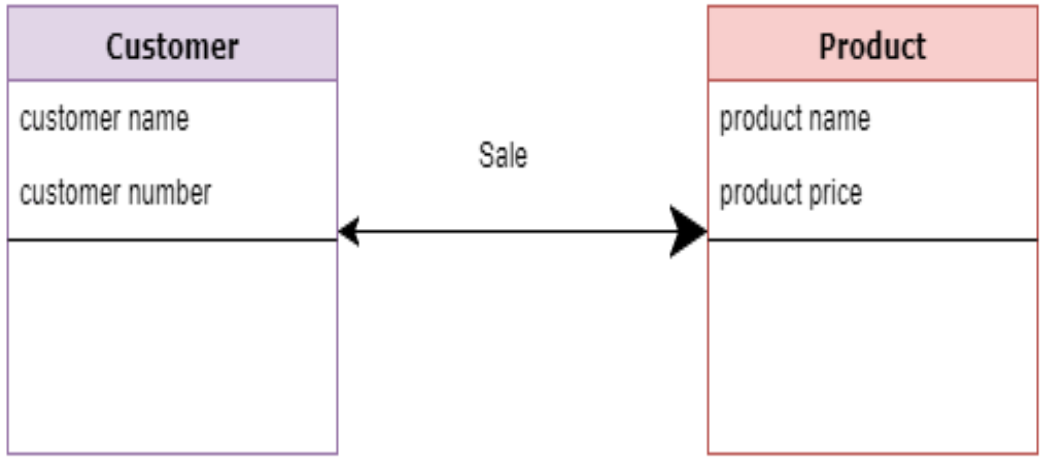
- Conceptual Model
- Logical Data Model
- Physical Data Model

Conceptual Model

- The data model defines what the system contains.
- This model is typically created by business stakeholders and Data Architects.
- The purpose is to organize scope and define business concepts and rules.
- The main aim of this model is to establish the entities, their attributes, and their relationships.
- In this Data Modeling level, there is hardly and detail available of the actual database structure. The 3 basic tenants of data model are:
- **Entity**: It is real-world thing.
- **Attribute**: It is characteristics or properties of an entity.
- **Relationship:** It is dependency or association between two entities.
- Example: Customer and Product are two entities. Customer number and name are attributes of the Customer entity. Product name and price are attributes of product entity. Sale is the relationship between the customer and product.

Characteristics of Conceptual Data Model

- Offers organization-wide coverage of the business concepts.
- This type of Data Models are designed and developed for a business audience.
- The conceptual model is developed independently of hardware specifications like data storage capacity, location or software specifications like DBMS vendor and technology.
- The focus is to represent data as a user will see it in the “real world”.



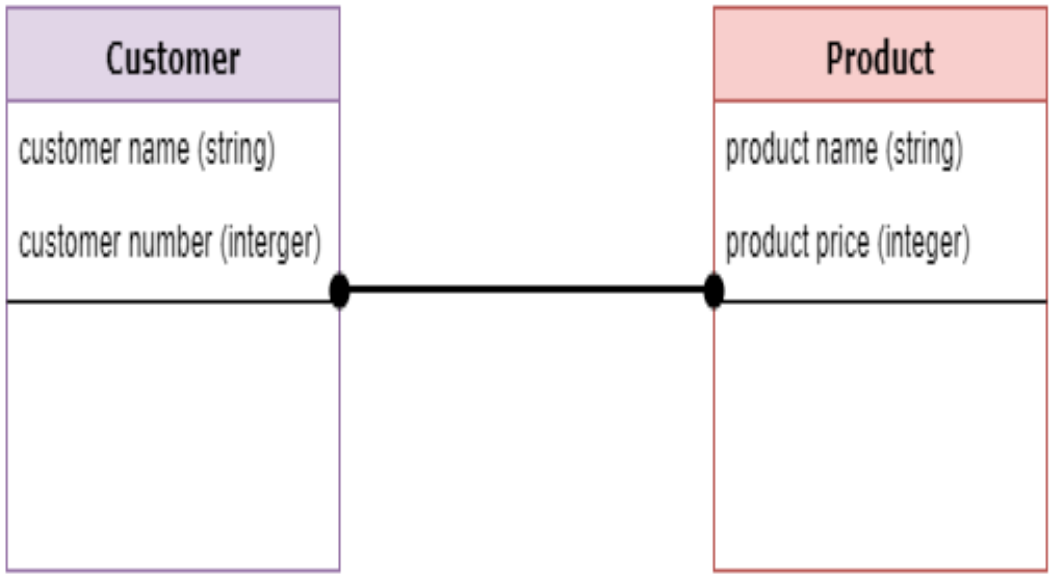
Conceptual Data Model

Logical Data Model

- **Logical Data Model** defines **how** the system should be implemented regardless of the DBMS. This model is typically created by Data Architects and Business Analysts.
- The purpose is to developed technical map of rules and data structures. Logical data models add further information to the conceptual model elements.
- The advantages of the Logical data model is to provide a foundation to form the base for the Physical model.
- At this data modeling level, no primary or secondary key is defined.
- At this data modeling level, you need to verify and adjust the connector details that were set earlier for relationships.

Characteristics of a logical data model

- Describes data needs for a single project.
- Designed and developed independently from the DBMS.
- Data attributes will have data types with exact precisions and length.
- Normalization processes to the model is applied typically till 3NF.



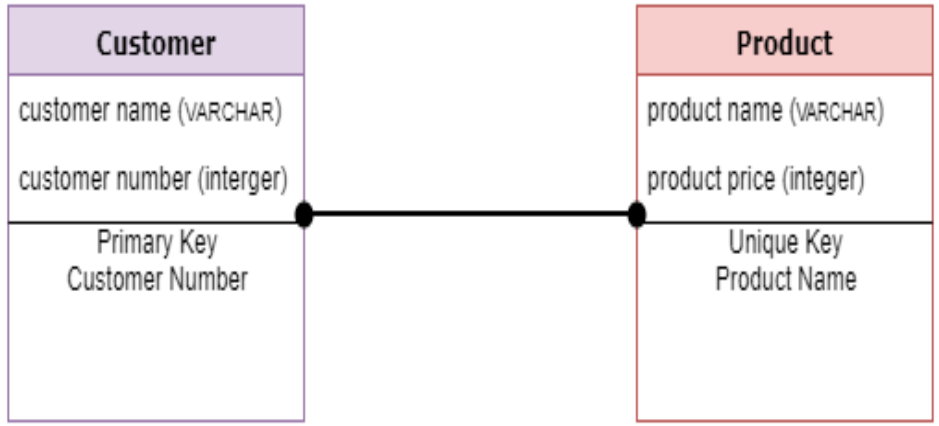
Logical Data Model

Physical Data Model

- **Physical Data Model** describes **how** the system will be implemented using a specific DBMS. This model is typically created by DBA and developers.
- The purpose is actual implementation of the database.
- A physical data model describes the database specific implementation of the data model.
- It offers an abstraction of the database and helps generate schema. This is because of the richness of meta-data offered by a Physical Data Model.
- This type of data model also helps to visualize database structure.
- It helps to model database columns keys, constraints, indexes, triggers, and other RDBMS Model.

Characteristics of a Physical data model

- Describes data needs for a single project integrated with other physical data models based on project scope.
- It contains relationships between tables that which addresses cardinality and null ability of the relationships.
- Developed for a specific version of a DBMS, location, data storage or technology to be used in the project.
- Columns should have exact data types, lengths assigned and default values.
- Primary and Foreign keys, views, indexes, access profiles, and authorizations, etc. are defined.



Physical Data Model

Advantages of Data Model

- The main goal of a designing data model is to make certain that data objects offered by the functional team are represented accurately.
- The data model should be detailed enough to be used for building the physical database.
- The information in the data model can be used for defining the relationship between tables, primary and foreign keys, and stored procedures.
- Data Model helps business to communicate the within and across organizations.
- Data model helps to documents data mappings in ETL process.
- Help to recognize correct sources of data to populate the model.

ETL: Extract, Transform and Load

Disadvantages of Data Model

- To develop data model one should know physical data stored characteristics.
- This is a navigational system produces complex application development, management. Thus, it requires knowledge of the biographical truth.
- Even smaller changes made in structure require modification in the entire application.
- There is no set data manipulation language in DBMS.

4.1.3Relational Database Model

- The vast majority of information systems today use the relational database model. The **relational database model** represents data in the form of related tables, or relations.
- A relation is a named, two-dimensional table of data. Each relation (or table) consists of a set of named columns and an arbitrary number of unnamed rows.
- Each column in a relation corresponds to an attribute of that relation.
- Each row of a relation corresponds to a record that contains data values for an entity.
- In relational model, the data and relationships are represented by collection of inter-related tables. Each table is a group of columns and rows, where column represents attribute of an entity and rows represents records.
- Relational data model represents the logical view of how data is stored in the relational databases.
- There exist some concepts related to this, which includes the following terms:
- **Table:** In relational data model, data is stored in the table. The table consists of a number of rows and columns. Thus, table is used because it can represent the data in the simplest form possible making data retrieval very fast.
- **Attribute:** Any relation have defined properties that are called an attributes. They are also called fields and columns.

- **Tuple:** Rows of table represents the tuple which contains the data records.
- **Domain:** Domain is a set of values which is indivisible i.e. value for each attribute present in the table contains some specific domain in which the value needs to lie. For example: The value of date of birth must be greater than zero. As, it cannot be negative. This is called domain of an attribute.
- **Relation:** A relation in relational data model represents the respective attributes and the correlation among them. Relations have several properties that distinguish them from non-relational tables:

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

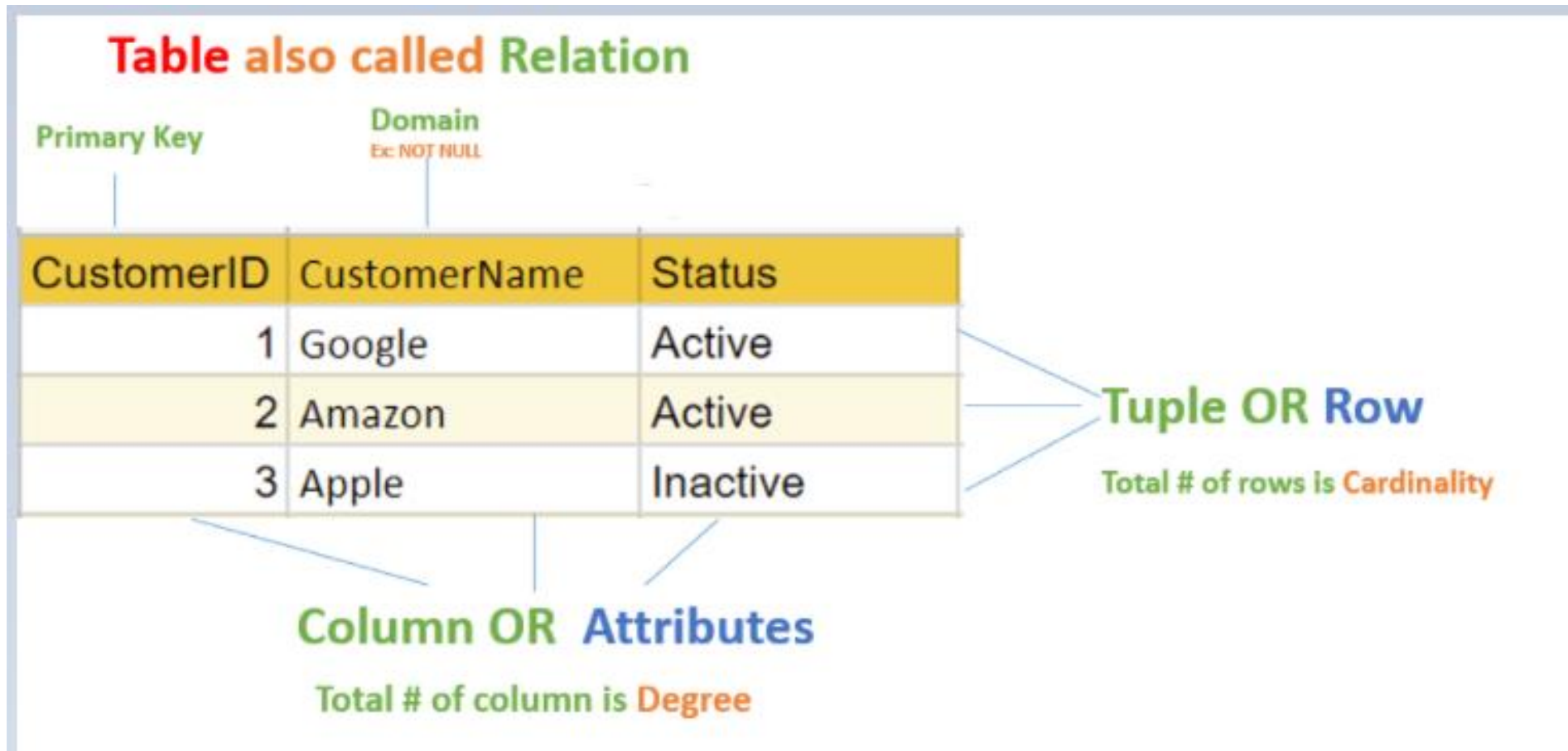


Table also called Relation
Primary Key
Domain
Ex: NOT NULL
CustomerID
CustomerName
Status
1
Google
Active
2
Amazon
Active
3
Apple
Inactive
Tuple OR Row
Total # of rows is Cardinality
Column OR Attributes
Total # of column is Degree

4.1.4Normalization

- **Normalization** is the process of organizing data into a related table; it also eliminates redundancy and increases the integrity which improves performance of the query.
- To normalize the database, we divide the database into tables and establish relationships between the tables.
- Database normalization can essentially be defined as the practice of optimizing table structures.
- Optimization is accomplished as a result of a through investigation of the various pieces of data that will be stored within the database, in particular concentrating upon how this data is interrelated.

- **Normalization avoids**
- **Duplication of data:** The same data is listed in multiple lines of the database.
- **Insert Anomaly/Inconsistency:** A record about an entity cannot be inserted into the table without first inserting information about another entity – Cannot enter a customer without a sales order.
- **Delete Anomaly:** A record cannot be deleted without deleting a record about a related entity. Cannot delete a sales order without deleting all of the customer's information.
- **Update Anomaly:** Cannot update information without changing information in many places. To update customer information, it must be updated for each sales order the customer has placed.

De-Normalization

- **De-Normalization** is the process of adding redundant data to speed up complex queries involving multiple tables JOINS. One might just go to a lower form of Normalization to achieve De-normalization and better performance.
- Data is included in one table from another in order to eliminate the second table which reduces the number of JOINS in a query and thus achieves performance.

Problems without Normalization

- If a table is not properly normalized and has data redundancy then it will not only eat up extra memory space but will also make it difficult to handle and update the database, without facing data loss. Insertion, updating and deletion anomalies are very frequent if database is not normalized.
- To understand these anomalies let us take an example of a student table.
- **Student**

Sid	Sname	Branch	HOD	Office Phone
100	Binod	Birtamode, Branch	Mr. Bikash	5401372
101	Anmol	Birtamode, Branch	Mr. Bikash	5401372
102	Ashana	Birtamode, Branch	Mr. Bikash	5401372
103	Menuka	Birtamode, Branch	Mr. Bikash	5401372
104	Umesh	New Road, Branch	Mr. Hari	5202343
105	Ramesh	New Road, Branch	Mr. Hari	5202343

- In the table above, we have data of six Computer Science students. As we can see, data for the fields branch, HOD and Office_Phone is repeated for the students who are in the same branch in the college, this id Data redundancy.

Insertion Anomaly

- Suppose for a new admission, until and unless a student opts/determines/decides for a branch, data of the student cannot be inserted, or else we will have to set the branch information as NULL. Also, if we have to insert data of 100 students of same branch, then the branch information will be repeated for all those 100 students. These scenarios are nothing but insertion anomalies.

Update Anomaly

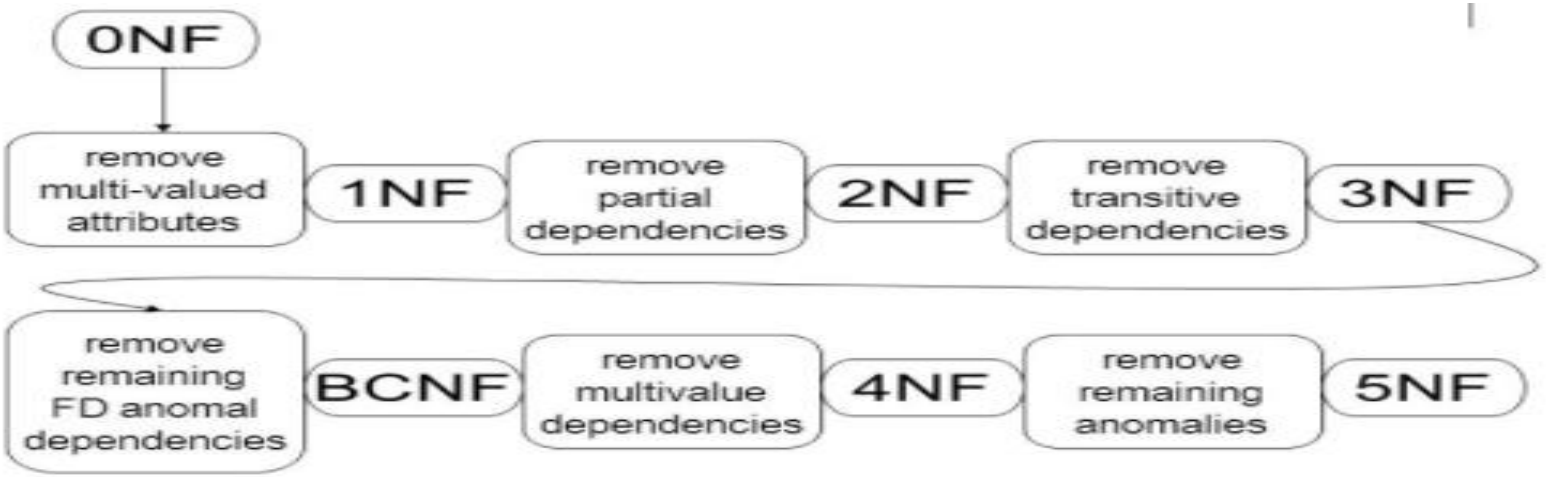
- When if Mr. Hari leaves the college? Or is no longer the HOD of computer science department? In that case all the student records will have to be updated, and if by mistake we miss and record, it will lead to data inconsistency. This is Updating anomaly.

Deletion Anomaly

- In our Student table, two different information are kept together, Student information and Branch information. Hence, at the end of the academic year, if student records are deleted, we will also lose the branch information. This is deletion anomaly.

Types of Normalization

- Normalization is a Six stage process – After the first stage, the data is said to be in first normal form, after the second, it is in second normal form, after the third, it is in third normal form and so on.
- Un-normalized (UDF) (Remove repeating Groups) – First Normal Form (1NF) (Remove partial Dependencies) – Second Normal Form (2NF) (Remove transitive Dependencies) – Third Normal Form (3NF) (Remove remaining Functional dependency anomaly) – Boyce Code Normal Form (BCNF) (Remove multi-valued Dependency) – Fourth Normal Form (4NF) (Remove remaining Anomalies) –Fifth Normal Form (5NF)



FD- Functional Dependency

First Normal Form (1NF)

In 1st NF:

- The table cells must be of single value.
- Eliminate repeating groups in individual tables.
- Create a separate table for each set of related data.
- Identify each set of related data with a primary key.

Definition: An entity is in the first normal form if it contains no repeating groups. In relational terms, a table is in the first normal form if it contains no repeating columns. Repeating columns make your data less flexible, waste disk space, and make it more difficult to search for data.

Order	Customer	Contact Person	Total
1	Rishab	Manish	134.23
2	Preeti	Roshan	521.24
3	Rishab	Manish	1042.42
4	Rishab	Manish	928.53

The above relations satisfies the properties of a relation and is said to be in first normal form (or 1NF). Conceptually it is convenient to have all the information in one relation since it is then likely to be easier to query the database.

Second Normal Form (2NF)

In 2NF:

- Remove partial dependencies.
- Functional dependency: The value of one attribute in a table is determined entirely by the value of another.
- Partial Dependency: A type of functional dependency where an attribute is functionally dependent on only part of the primary key (primary key must be a composite key).
- Create separate table with the functionally dependent data and the part of the key on which it depends. Tables created at this step will usually contain descriptions of resources.

Definition: A relation is in 2NF if it is in 1NF and every non-key attribute is fully dependent on each candidate key of the relation.

Example:

The following relation is not in Second Normal Form (2NF):

Order	Customer	Contact Person	Total
1	Rishab	Manish	134.23
2	Preeti	Rohan	521.24
3	Rishabh	Manish	1042.42
4	Rishabh	Manish	928.53

- In the table above, the order number serves as the primary key. Notice that the customer and total amount are dependent upon the order number – this data is specific to each order. However, the contact person is dependent upon the customer. An alternative way to accomplish this would be to create two tables:

Customer	Contact Person
Rishabh	Manish
Preeti	Rohan

Order	Customer	Total
1	Rishabh	134.23
2	Preeti	521.24
3	Rishabh	1042.42
4	Rishabh	928.53

- The creation of two separate tables eliminates the dependency problem. In the second table, contact person is dependent upon the primary key – customer name.
- The first table only includes the information unique to each order.
- Someone interested in the contact person for each order could obtain this information by performing a join operation.

Third Normal Form (3NF)

In 3rd NF:

- Remove transitive dependencies.
- Transitive Dependency: A type of functional dependency where an attribute is functionally dependent on an attribute other than the primary key. Thus its value is only indirectly determined by the primary key.
- Create a separate table containing the attribute and the fields that are functionally dependent on it. Tables created at this step will usually contain descriptions of either resources or agents. Keep a copy of the key attribute in the original file.

Definition:

- A relation is in third normal form, if it is in 2NF and every non-key attribute of the relation is non-transitively dependent on each candidate key of the relation.

- Example

Company	City	State	Zip
Sanima Ltd.	Kathmandu	State 3	10169
Shivam Ltd.	Birjung	State 2	33196
API Company Ltd.	Darchula	State 7	21046

- The above table is not in the 3NF.
- In this example, the city and state are dependent upon the ZIP code. To place this table in 3NF, two separate tables would be created – one containing the company name and ZIP code and the other containing city, state, ZIP code pairings.

Company	ZIP
Sanima Ltd.	10169
Shivam Ltd.	33196
API Company Ltd.	21046

City	State	ZIP
Kathmandu	State 3	10169
Birgunj	State 2	33196
Darchula	State 7	21046

Database designers should always keep in mind the tradeoffs between higher level normal forms and the resource issues that complexity created.

Boyce-Codd Normal Form (BCNF)

A relation is in Boyce-Codd Normal Form (BCNF) if every determinant is a candidate key.

In BCNF:

- When a relation has more than one candidate key, anomalies may result even though the relation is in 3NF.
- 3NF does not deal satisfactorily with the case of a relation with overlapping candidate keys. i.e. composite candidate keys with at least one attribute in common.
- BCNF is based on the concept of a determinant.
- A determinant is any attribute (simple or composite) on which some other attribute is fully functionally dependent.
- A relation is in BCNF is, and only if, every determinant is a candidate key.

- **Example: Client Interview**

Client No	Interview Date	Interview Time	Staff No	Room No
CR76	13-May-11	10:30	SG5	G101
CR76	13-May-11	12:00	SG5	G101
CR74	13-May-11	12:00	SG37	G102
CR56	02-July-11	10:30	SG5	G102

- Functional dependency 1(FD1): Client No, Interview Date – Interview Time, Staff No, Room No(Primary Key)
- FD2: Staff No, Interview Date, Interview Time – Client No (Candidate key)
- FD3: Room No, Interview Date, Interview Time – Client No, Staff No (CK)
- FD4: Staff No, Interview Date – Room No (Not a Candidate Key (CK))
- As a consequence, the client Interview relation may suffer from update anomalies. To transform the Client Interview relation to BCNF, we must remove the violating functional dependency by creating two new relations called Interview and Staff Room as shown below,
- **Interview (Client No, Interview Date, Interview Time, Staff No)**
- **Staff Room (Staff No, Interview Date, Room No)**

Interview

Client No	Interview Date	Interview Time	Staff No
CR76	13-May-11	10:30	SG5
CR76	13-May-11	12:00	SG5
CR74	13-May-11	12:00	SG37
CR56	02-July-11	10:30	SG5

Staffroom

Staff No	Interview Date	Room No
SG5	13-May-11	G101
SG37	13-May-11	G102
SG5	02-July-11	G102

- **Fourth Normal Form (4NF)**

A table in fourth normal form (4NF) if and only if it is in BCNF and contain no more than one multi-valued dependency.

An entity is in Fourth Normal Form (4NF) when it meets the requirement of being in Third Normal Form (3NF) and additionally:

- Has no multiple set of multi-valued dependencies. In other words, 4NF states that no entity can have more than a single one-to-many relationship within an entity if the one-to-many attributes are independent of each other.
- Fourth Normal Form applies to situations involving many-to-many relationships.
- In relational databases, many-to-many relationships are expressed through cross-reference tables.

Definition: A table is in fourth normal form (4NF) if and only if it is in BCNF and contains no more than one multi-valued dependency.

- For example:

Employee	Skills	Hobbies
1	Programming	Golf
1	Programming	Bowling
1	Analysis	Golf
1	Analysis	Bowling
2	Analysis	Golf
2	Analysis	Gardening
2	Management	Golf
2	Management	Gardening

- Employee Table

- In a table, it is difficult to maintain since adding a new hobby required multiple new rows corresponding to each skill. This problem is created by the pair of multi-valued dependencies Employee – Skills and Employee – Hobbies. A much better alternative would be to decompose INFO into two relations:

Employee	Skills
1	Programming
1	Analysis
2	Analysis
2	Management

Hobbies

Employee	Hobbies
1	Golf
1	Bowling
2	Golf
2	Gardening

Fifth Normal Form (5NF)

In 5th Normal Form:

- A relation that has a join dependency cannot be decomposed by a projection into other relations without spurious results.
- A relation is in 5NF when its information content cannot be reconstructed from several smaller relations i.e. from relations having fewer attributes than the original relation.

Definition: A table is in fifth normal form (5NF) or Project-Join Normal Form (PJNF) if it is in 4NF and it cannot have a lossless decomposition into any number of smaller tables.

- Fifth normal form, also known as join-projection normal form (JPNF), states that no non-trivial join dependencies exist. 5NF states that any fact should be able to be reconstructed without any anomalous results in any case, regardless of the number of tables being joined. A 5NF table should have only candidate keys and its primary key should consist of only a single column.

Example:

A buying table, used to track buyers, what they buy, and from whom they buy. Take the following sample data:

Buyer	Vendor	Item
Kamala	Alphabet House	Jeans
Abin	Alphabet House	Jeans
Kamala	Radhika Sarees	Saree
Abin	Radhika Sarees	Saree
Kamala	Radhika Sarees	Suit

- The problem with the above table structure is that if Claiborne starts to sell Jeans then how many records must you create to record this fact? The problem is there are pair wise cyclical dependencies in the primary key. That is, in order to determine the item you must know the buyer and vendor, and to determine the vendor, you must know the buyer and the item, and finally to know the buyer you must know the vendor and the item. And the solution is to break this one table into three tables; Buyer-Vendor, Buyer-Item, and Vendor-Item.

Buyer-Vendor

Buyer	Vendor
Kamala	Alphabet House
Abin	Alphabet House
Kamala	Radhika Sarees
Abin	Radhika Sarees

Buyer-Item

Buyer	Item
Kamala	Jeans
Abin	Jeans
Kamala	Saree
Abin	Saree
Kamala	Suit

Vendor-Item

Vendor	Item
Alphabet House	Jeans
Radhika Sarees	Saree
Radhika Sarees	Suit

The tables given above are in the 5NF.

4.1.5Transforming E R Diagrams Into Relations

- Transforming an E-R diagram into normalized relations and then merging all the relations into one final, consolidated set of relations can be accomplished in four steps. These steps are summarized briefly here, and then steps, 1, 2, 3 and 4 are discussed in detail in the remainder of this chapter.

1. Represent entities:

- Each entity type in the E-R diagram becomes a relation. The identifier of the entity type becomes the primary key of the relation, and other attributes of the entity type become non-primary key attributes of the relation.

2. Represent relationships:

- Each relationship in an E-R diagram must be represented in the relational database design. How we represent a relationship depends on its nature. For example, in some cases we represent a relationship by making the primary key of one relation a foreign key of another relation. In another cases, we create a separate relation to represent a relationship.

3. Normalize the relations:

- The relations created in steps 1 and 2 may have unnecessary redundancy. So we need to normalize these relations to make them well structured.

4. Merge the relations:

- So far in database design we have created various relations from both a bottom-up normalization of user views and from transforming one or more E-R diagrams into sets of relations. Across these different sets of relations, there may be redundant relations (two or more relations that describe the same entity type) that must be merged and renormalized to remove the redundancy.

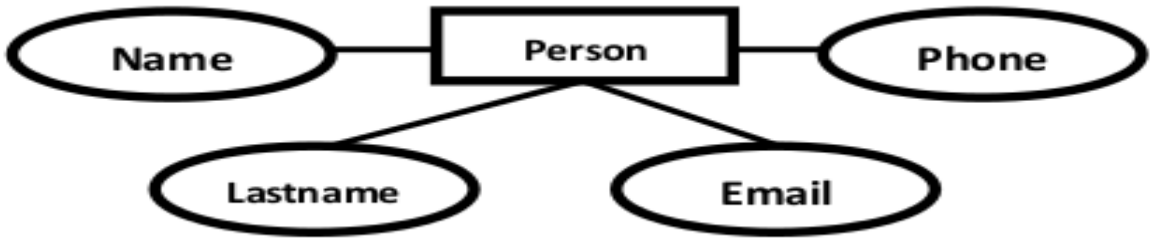
Transforming Entities and Relationships E-R Diagrams into Relations

- The ER Model is intended as a description of real-world entities. Although it is constructed in such a way as to allow easy translation to the relational schema model, this is not an entirely trivial process. The ER diagram represents the conceptual level of database design meanwhile the relational schema is the logical level for the database design. We will be following the simple rules:

1. Entities and Simple Attributes:

An entity type within ER diagram is turned into a table. You may preferably keep the same name for the entity or give it a sensible name but avoid DBMS reserved words as well as avoid the use of special characters. Each attribute turns into a column (attribute) in the table. The key attribute of the entity is the primary key of the table which is usually underlined. It can be composite if required but can never be null.

- Taking the simple ER diagram:



- The initial relational schema is expressed in the following format writing the table names with the attributes list inside a parentheses as shown below:

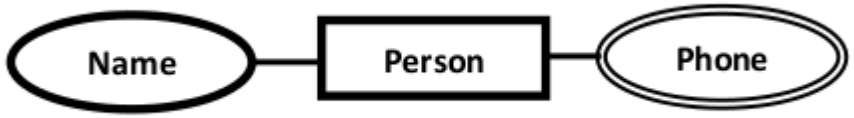
Persons(personid , name, lastname, email)

Persons and Phones are Tables. name, lastname, are Table Columns (Attributes).

personid is the primary key for the table : Person

2. Multi-Valued Attributes

- A multi-valued attribute is usually represented with a double-line oval.



- If you have a multi-valued attribute, take the attribute and turn it into a new entity or table of its own. Then make a 1:N relationship between the new entity and the existing one. In simple words.

1. Create a table for the attribute.
2. Add the primary (id) column of the parent entity as a foreign key within the new table as shown below:

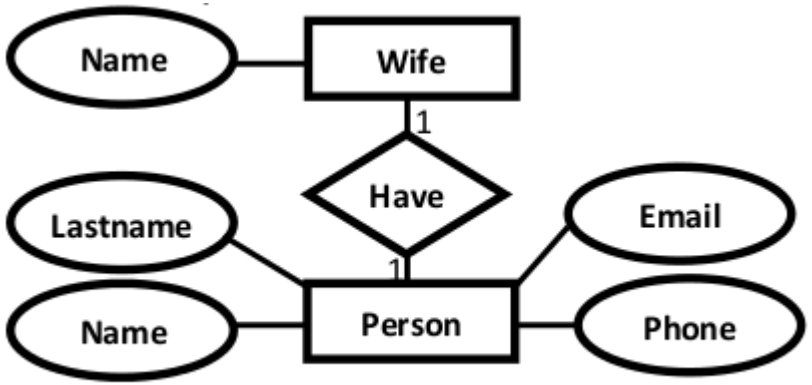
Persons(personid , name, lastname, email)

Phones (phoneid , personid, phone)

personid within the table Phones is a foreign key referring to the personid of Persons

3. 1:1 Relationships:

- To keep it simple and even for better performances at data retrieval, I would personally recommend using attributes to represent such relationship. For instance, let us consider the case where the Person has or optionally has one wife. You can place the primary key of the wife within the table of the Persons which we call in this case Foreign key as shown below:



Persons(personid , name, lastname, email , ***wifeid***)

Wife (wifeid , name)

Or vice versa to put the **personid** as a foreign key within the Wife table as shown below:

Persons(personid , name, lastname, email)

Wife (wifeid , name , ***personid***)

For cases when the Person is not married i.e. has no wifeID, the attribute can set to NULL

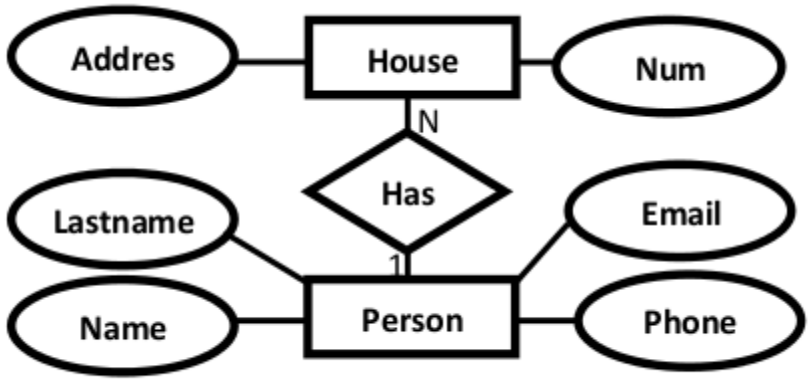
4. 1:N Relationships

This is the tricky part ! For simplicity, use attributes in the same way as 1:1 relationship but we have only one choice as opposed to two choices. For instance, the Person can have a **House** from zero to many , but a **House** can have only one **Person**. To represent such relationship the **personid** as the Parent node must be placed within the Child table as a foreign key but **not the other way around as shown next:**

It should convert to :

Persons(personid , name, lastname, email)

House (houseid , num , address, personid)

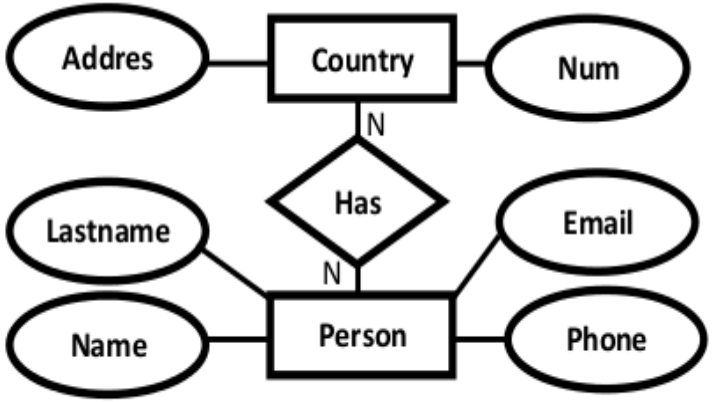


5. N:N Relationships

- We normally use tables to express such type of relationship. This is the same for N − ary relationship of ER diagrams. For instance, The Person can live or work in many countries. Also, a country can have many people. To express this relationship within a relational schema we use a separate table as shown below:

It should convert into :

- Persons(<u>personid</u> , name, lastname, email)
- Countries (<u>countryid</u> , name, code)
- HasRelat (<u>hasrelatid</u> , **personid , countryid**)

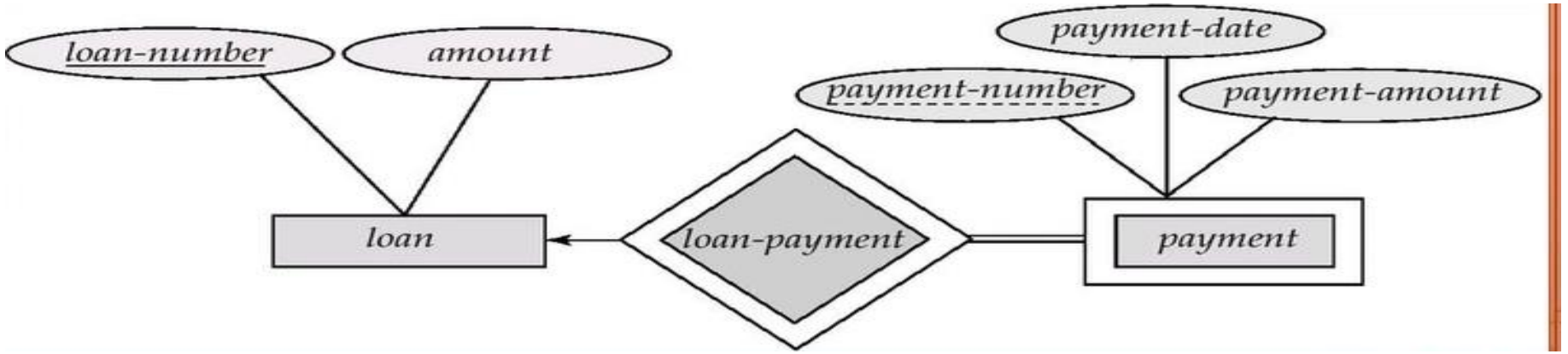


Mapping Weak Entity Sets to ER

- A weak entity set does not have its own primary key and always participates in one-to-many relationship with owner entity set and has total participation. For a weak entity set create a relation that contains all simple attributes (or simple components of composite attributes). In addition, relation for weak entity set contains primary key of the owner entity set as a foreign key and its primary key is formed by combining partial key (discriminator) and primary key of the owner entity set.
- Mapping Process:

1. Create table for weak entity set.
2. Add all its attributes to table as field.
3. Add the primary key of identifying entity set.
4. Declare all foreign key constraints.

- Example: Lets take a weak entity as Dependents as shown in ER diagram below:



- Here to draw table of weak entity set ‘Payment’ we simply set all their attributes i.e payment-number, payment-date, payment-amount and also set primary key of Loan table to the Payment table as below:

Loan

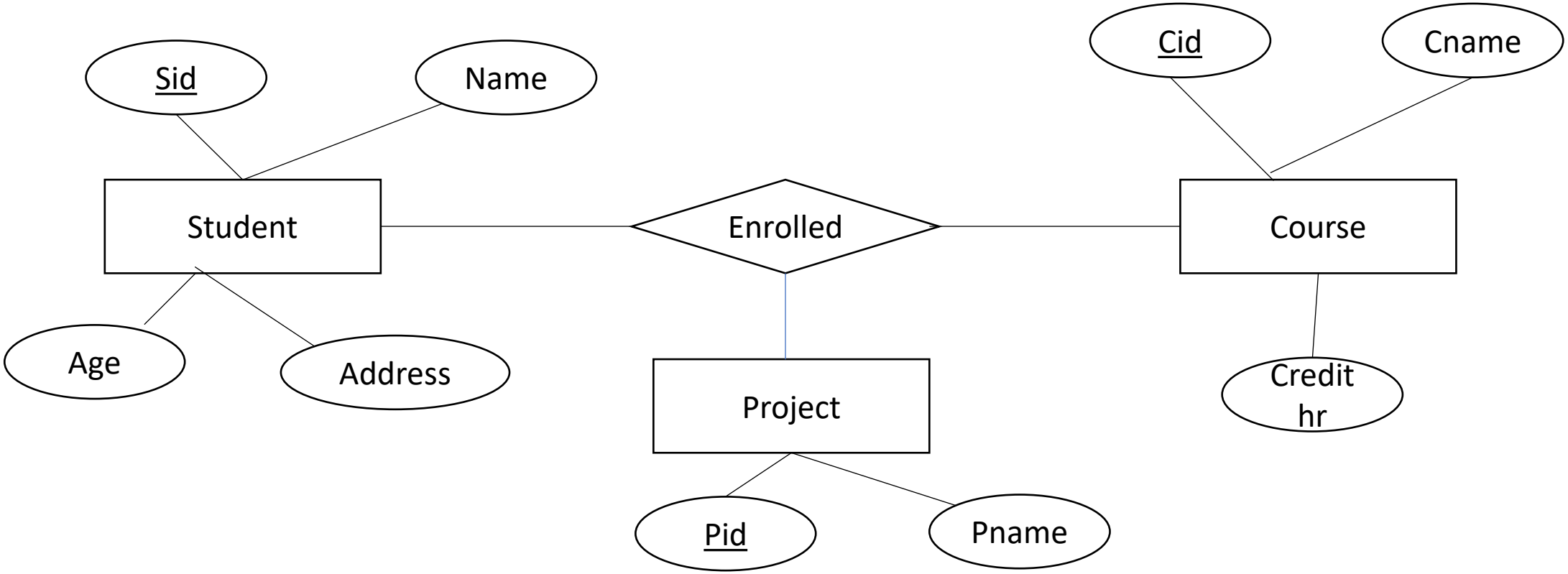
loan-number	amount

Payment

loan-number	Payment-number	Payment-date	Payment-amount

Mapping of N-ary relationship types to ER

- For each n-ary relationship set for n>2, a new relation is created. Primary keys of all participating entity sets are included in the relation as foreign key attributes. Besides this all simple attributes of the n-ary relationship set (or simple components or composite attributes are includes as attributes of the relation.



- Here we create separate table for each entities and also create a single table for relationship Enrolled that contains the primary keys of each of the entities associated with this relationship.

Student

Sid	Name	Age	Address

Course

Cid	Cname	Credithr

Project

Pid	Pname

Enrolled

Sid	Cid	Pid

4.1.6Merging Relations

- As part of the logical database design, normalized relations likely have been created from a number of separate E-R diagrams and various user interfaces. Some of the relations may be redundant – they may refer to the same entities. If so, you should merge those relations to remove the redundancy. This section describes merging relations, or view integration, which is the last step in logical database design and prior to physical file and database design.

An example of Merging Relations

Suppose that modeling a user interface or transforming an E-R diagram results in the following 3NF relations:

Employee1 (Emp_ID, Name, Address, Phone)

Modeling a second user interface might result in the following relations:

Employee2 (Emp_ID, Name, Address, Jobcode, Number_of_Years)

Because these two relations have the same primary key (Emp_ID) and describe the same entity, they should be merged into one relation. The result of merging the relations in the following relation:

Employee (Emp_ID, Name, Address, Phone, Jobcode, Number_of_Years)

Notice that an attribute that appears in both relations (such as Name in this example) appears only once in the merged relation.

4.1.7Physical 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 determinations in systems analysis. 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.

4.1.8Designing 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 is a normalized student relation might be represented as three fields: last name, first name, and middle name.

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.
- For example, Table 9-2 lists the most commonly used data types available in Oracle 10g. Selecting a data type balances four objectives that will vary in degree of importance depending on the application:

1. Minimize storage space
2. Represent all possible values of the field
3. Improve data integrity for the field
4. Support all data manipulations desired on the field

TABLE 9-2 Commonly Used Data Types in Oracle 10g

Data Type	Description
VARCHAR2	Variable-length character data with a maximum length of 4000 characters; you must enter a maximum field length (e.g., VARCHAR2(30) for a field with a maximum length of 30 characters). A value less than 30 characters will consume only the required space.
CHAR	Fixed-length character data with a maximum length of 255 characters; default length is 1 character (e.g., CHAR(5) for a field with a fixed length of five characters, capable of holding a value from 0 to 5 characters long).
LONG	Capable of storing up to two gigabytes of one variable-length character data field (e.g., to hold a medical instruction or a customer comment).
NUMBER	Positive and negative numbers in the range 10^{-130} to 10^{126}; can specify the precision (total number of digits to the left and right of the decimal point) and the scale (the number of digits to the right of the decimal point) (e.g., NUMBER(5) specifies an integer field with a maximum of 5 digits and NUMBER(5, 2) specifies a field with no more than five digits and exactly two digits to the right of the decimal point).
DATE	Any date from January 1, 4712 BC to December 31, 4712 AD; date stores the century, year, month, day, hour, minute, and second.
BLOB	Binary large object, capable of storing up to four gigabytes of binary data (e.g., a photograph or sound clip).

- **Calculated Fields:** it is common for an attribute to be mathematically related to other data. For example, an invoice may include a total due field, which represents the sum of the amount due on each item on the invoice. A field that can be derived from other database fields is called a calculated field (or a computed field or a derived field). Recall that a functional dependency between attributes does not imply a calculated field. Some database technologies allow you to explicitly define calculated fields along with other raw data fields. 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.

4.1.9Designing Physical Tables

- A relational database is a set of related table (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.
- In contrast, a physical table is a named set of rows and columns that specifies the fields in each row of the table.
- A physical table may or may not correspond to one relation. Whereas normalized relations possess properties of well-structured relations, the design of a physical table has two goals different from those of normalization: efficient use of secondary storage and data processing speed.

- The efficient use of secondary storage (disk space) related 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 I/O operations that must be performed. Typically, the data in one physical table are stored close together on disk. De- normalization is the process of splitting or combining normalized relations into physical tables based on affinity (attraction / similarity) of use of rows and fields.

4.2 Designing Forms And Reports

4.2.1 Introduction,

4.2.2Designing Forms And Reports,

4.2.3Formatting Forms And Reports,

4.2.4Assessing Usability

4.2 Designing Forms And Reports

4.2.1 Introduction

- The forms are used to present or collect information on a single item, such as a customer, product, or event.
- Forms can be used for both input and output.
- Reports on the other hand are used to convey information on a collection of items.
- Form and report design is a key ingredient for successful systems. Because users often equate the quality of a system its input and output methods, you can see that the design process for forms and reports is an especially important activity.
- And because information can be collected and formatted in many ways, gaining an understanding of design dos and don’ts and the tradeoffs between various formatting options is useful for all system analysts.
- Both forms and reports are the product of input and output design and are business document consisting of specified data. The main difference is that forms provide fields for data input but reports are purely used for reading.

4.2.2Designing Forms And Reports

- Forms are used to present or collect information on a single item such as a customer, product or event. Forms can be used for both input and output.
- Reports, on the other hand, are used to convey information on a collection of items.
- Forms and report design is a key ingredient for successful systems. As users often equate the quality of a system to the quality of its input and output methods, the design process for forms and reports is an especially important activity.
- Forms and reports are identified during requirements structuring. The kinds of forms and reports the system will handle are established as a part of the design strategy formed at the end of the analysis phase of the system development process.
- Forms and reports are integrally related to various diagrams developed during requirements structuring. For example, every input form will be associated with a data flow entering a process on a DFD, and every output form or report will be a data flow produced by a process on a DFD.
- Further, the data on all forms and reports must consists of data elements on data stores and on the E- R data model for the applications, or must be computed from these data elements.

Form

A business document that contains some predefined 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; it is a passive document used solely for reading or viewing. A report typically contains data from many unrelated records or transactions.

The process of Designing Forms and Reports

Designing forms and reports is a user-focused activity that typically follows a prototyping approach. User-centered design refers to a design approach that involves an understanding of the target audience, their tasks and goals, information

needs, experience levels, and so on. First, you must gain an understanding of the intended user and task objectives by collecting initial requirements during requirements determination. During this process, several questions must be answered. These questions attempt to answer the “who, what, when, where, and how” related to the creation of all forms and reports are given below:

- Who will use the form and report?
- What is the purpose of the form or report?
- When is the form or 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 and report?

- Gaining an understanding of these questions is a required first step in the creation of any form or report. After collecting the initial requirements, you structure and refine this information into an initial prototype. Structuring and refining the requirement are completely independent of the users, although you may need to occasionally contact users in order to clarify some issues overlooked during analysis. Finally, you ask users to review and evaluate the prototype. After reviewing the prototype, users may accept the design or request that changes be made. If changes are needed, you will repeat the construction- evaluate-refinement cycle until the design is accepted. Usually, several iterations of this cycle occur during the design of a single form or report.
- The initial prototype may be constructed in numerous environments. The obvious choice is to use CASE tool or the standard development tools used within your organization. Often, initial prototypes are simply mock screens that are not working modules or systems. Mock screens can be produced from a word processor; computer graphics design package, or electronic spreadsheet.

Deliverables and Outcomes

Design specifications are the major deliverables and are inputs to the system implementation phase. Design specifications have three sections:

1. Narrative overview: This section contains general overview of the characteristics of the target users, tasks, system, and environmental factors in which the form or report will be used. The purpose is to explain to those who will actually develop the final form, why this form exists, and how it will be used so that they can make the appropriate decisions.

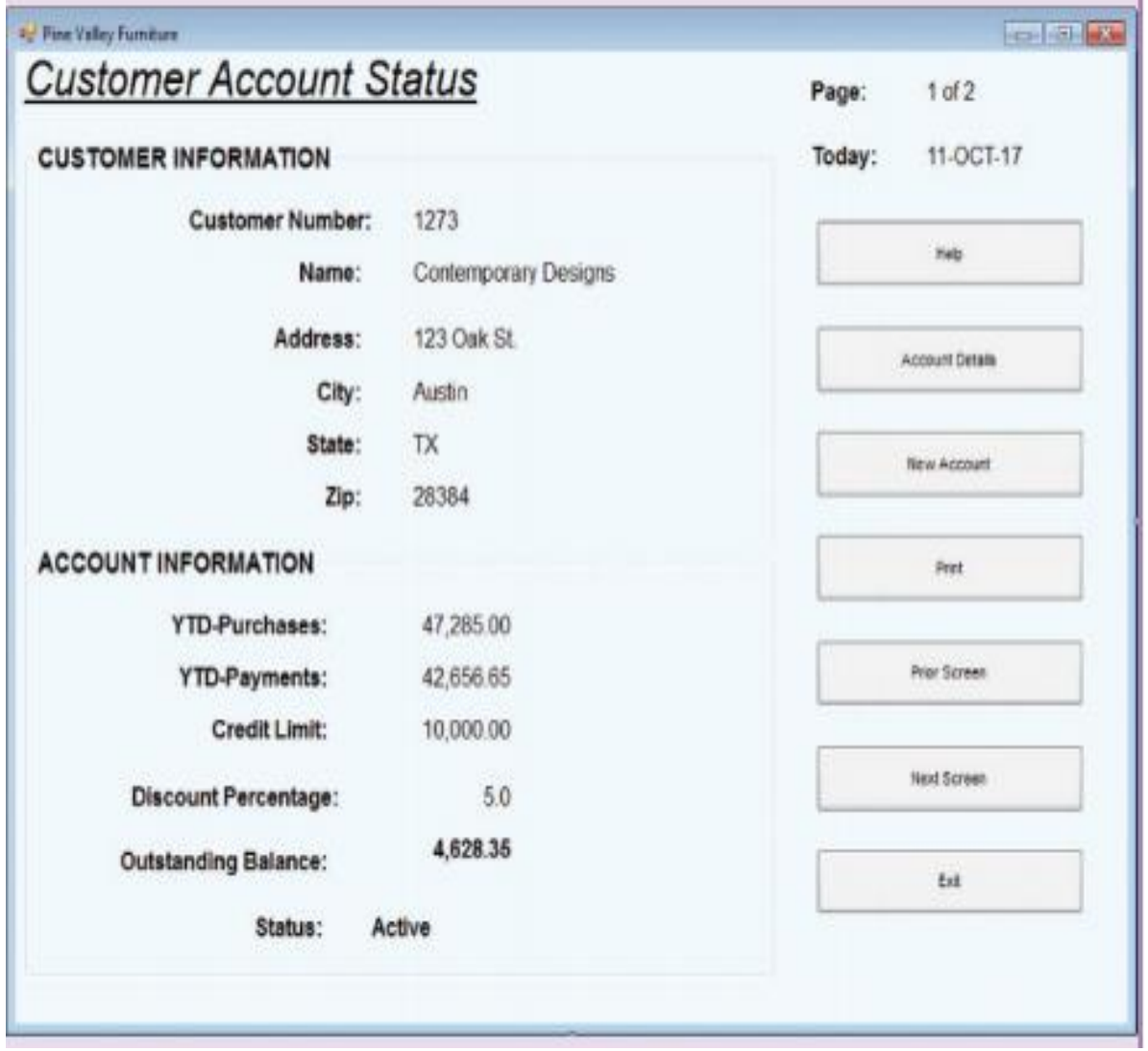
2. Sample Design: This section provides a sample design of the form. This design may be hand drawn using a coding sheet although, in most instances, it is developed using CASE or standard development tool. Using actual development tools allows the design to be more thoroughly tested and assessed.

3. Testing and usability assessment: This section provides all testing and usability assessment information. Assessing usability depends on speed, accuracy, and satisfaction.

(a) Narrative overview

Form:	Customer Account Status
Users:	Customer account representatives within corporate offices
Tasks:	Assess customer account information: address, account balance, year-to-date purchases and payments, credit limit, discount percentage, and account status
System:	Novell Network, Microsoft Windows
Environment:	Standard office environment

(b) Sample design



(c) Testing and usability assessment

User Rated Perceptions (average 14 users):

consistency [1 = consistent to 7 = inconsistent]:	1.52
sufficiency [1 = sufficient to 7 = insufficiency]:	1.43
accuracy [1 = accurate to 7 = inaccurate]:	1.67

. . .

4.2.3Formatting Forms And Reports

- A wide variety of information can be provided to users of information systems and, as technology continues to evolve, a greater variety of data types will be used. There are numerous guidelines for formatting information.

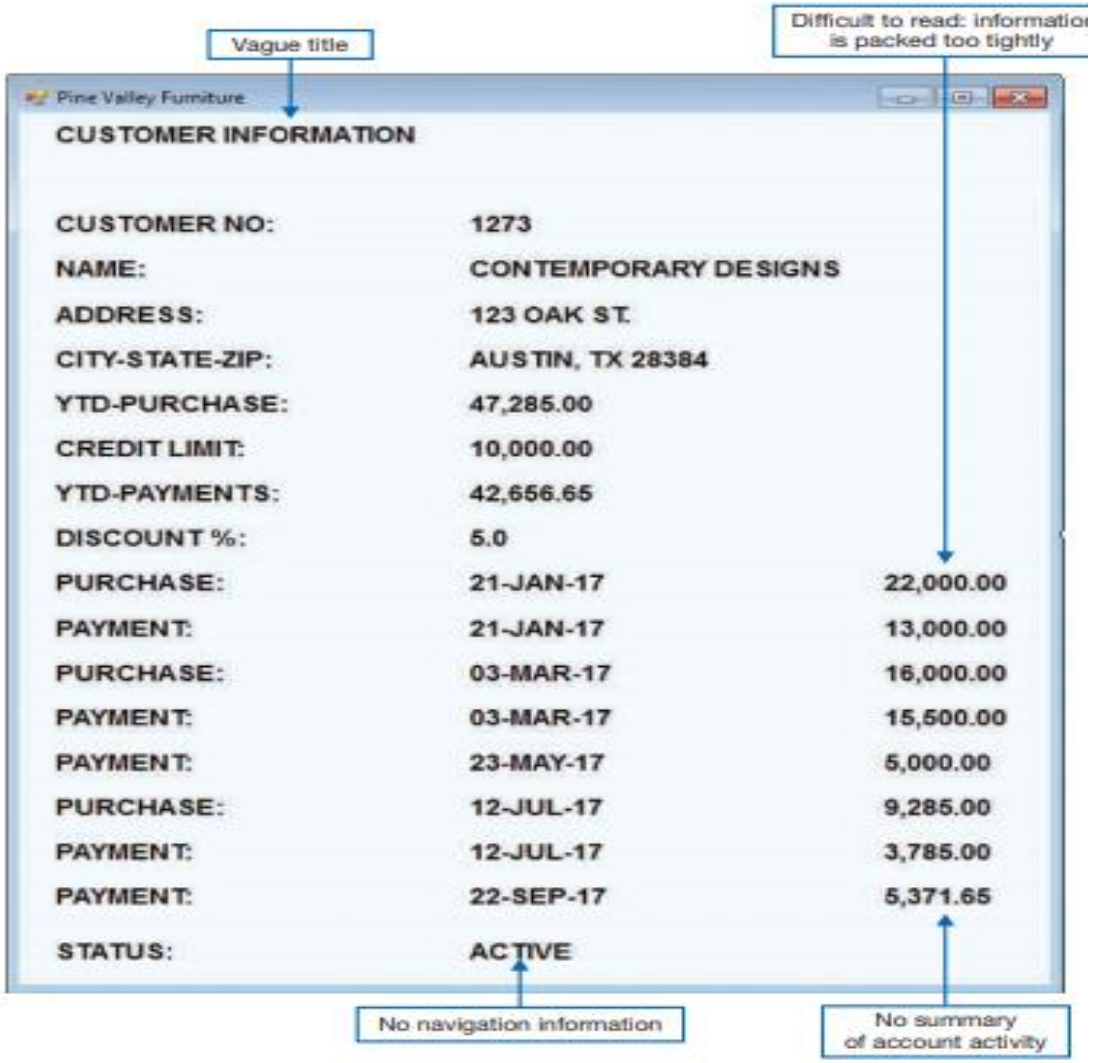
General Formatting Guidelines

- Over the past several years, industry and academic researchers have spent considerable effort investigating how information formatting influences individual task, performance, and perceptions of usability. Through this work, several guidelines for formatting information have emerged as given below:

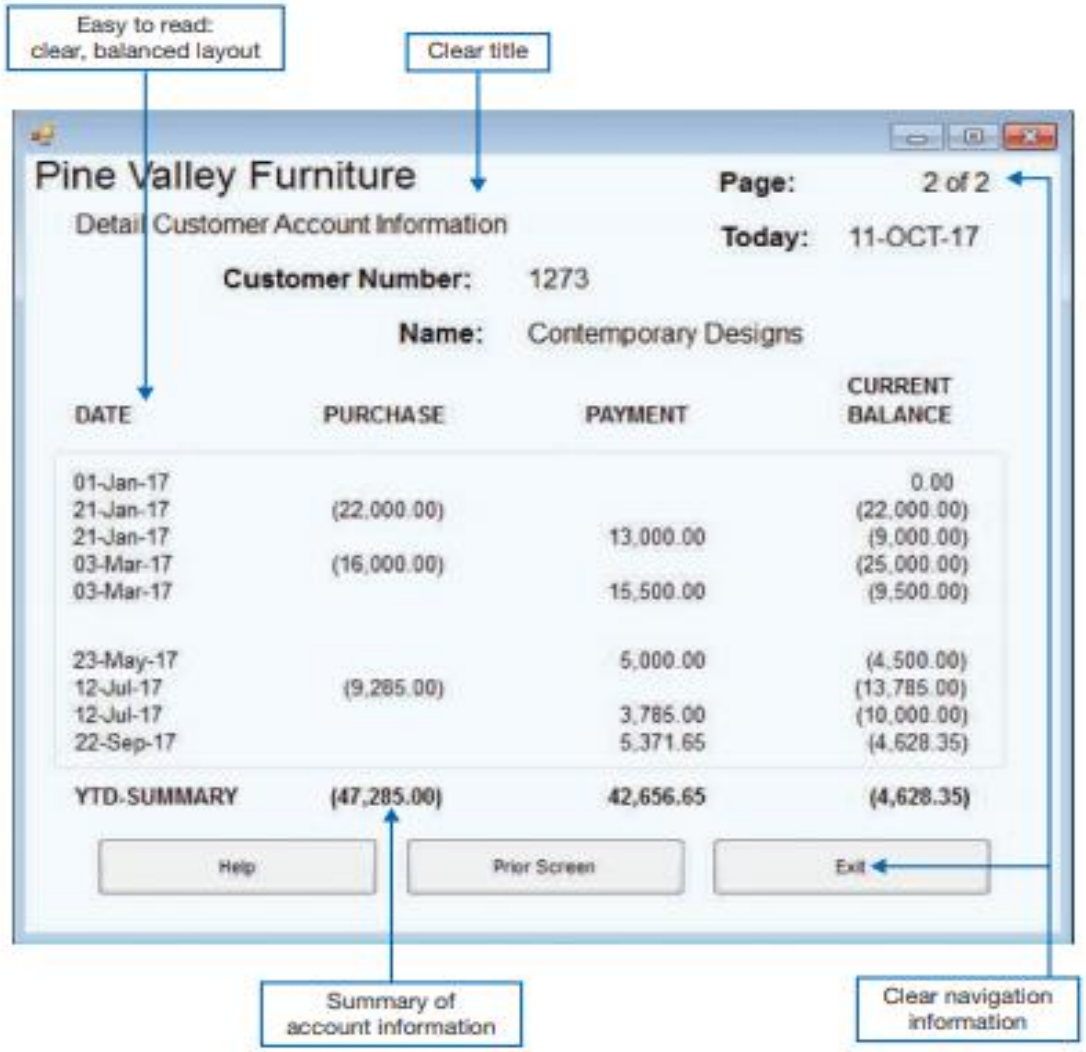
TABLE 10-3 General Guidelines for the Design of Forms and Reports

Meaningful Titles:
Clear and specific titles describing content and use of form or report
Revision date or code to distinguish a form or report from prior versions
Current date, which identifies when the form or report was generated
Valid date, which identifies on what date (or time) the data in the form or report were accurate
Meaningful Information:
Only needed information should be displayed
Information should be provided in a manner that is usable without modification
Balance the Layout:
Information should be balanced on the screen or page
Adequate spacing and margins should be used
All data and entry fields should be clearly labeled
Design an Easy Navigation System:
Clearly show how to move forward and backward
Clearly show where you are (e.g., page 1 of 3)
Notify user when on the last page of a multipaged sequence

Contrasting customer information forms



Improved design for form



Highlighting Forms

TABLE 10-4 Methods of Highlighting

Blinking and audible tones
Color differences
Intensity differences
Size differences
Font differences
Reverse video
Boxing
Underlining
All capital letters
Offsetting the position of nonstandard 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

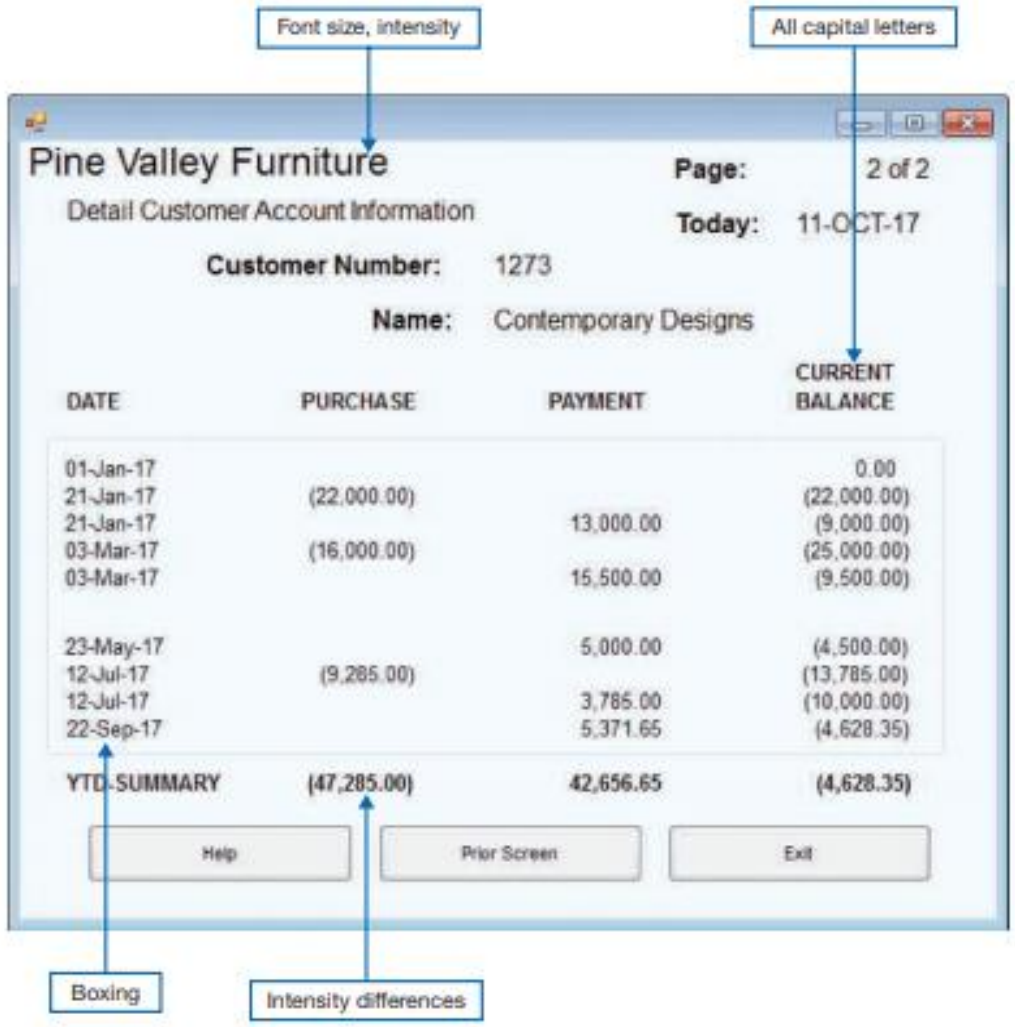


TABLE 10-5 Benefits and Problems from Using Color

Benefits from Using 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 (e.g., color blindness).
Resolution may degrade with different displays.
Color fidelity may degrade on different displays.
Printing or conversion to other media may not easily translate.

(*Source:* Based on Shneiderman et al., 2009; Benbasat, Dexter, and Todd, 1986.)

TABLE 10-6 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.

TABLE 10-7 General Guidelines for Displaying Tables and Lists

Use Meaningful 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 (e.g., ascending, descending, or alphabetic).
Place a blank line between every five rows in long columns.
Similar information displayed in multiple columns should be sorted vertically (i.e., read from top to bottom, not left to right).
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 30–40 characters per line (this is what newspapers use, and it is easier to speed-read).
Break long sequences of *alphanumeric data* into small groups of three to four characters each.

TABLE 10-8 Guidelines for Selecting Tables versus Graphs

Use Tables For:

Reading individual data values

Use Graphs For:

Providing a quick summary of data
Detecting trends over time
Comparing points and patterns of different variables
Forecasting activities
Reporting vast amounts of information when relatively simple impressions are to be drawn

4.2.4Assessing Usability

- There are many factors to consider when you design forms and reports. The objective for designing forms, reports, and all human-computer interactions is usability. Usability typically refers to the following three characteristics:

1. **Speed:** Can you complete a task efficiently?
2. **Accuracy:** Does the system provide what you expect?
3. **Satisfaction:** Do you like using the system?

- In other words, usability means that your designs should assist, not hinder, user performance. Thus, usability refers to an overall evaluation of how a system performs in supporting a particular user for a particular task. In the remainder of this section, we describe numerous factors that influence usability and several techniques for assessing the usability of a design.

General Design Guidelines for Usability of Forms and Reports

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 key strokes, and the system should remember where the user stopped

4.3 Designing Interfaces and Dialogues

4.3.1 Introduction,

4.3.2 Designing Interfaces And Dialogues,

4.3.3 Interaction Methods And Devices,

4.3.4 Designing Interfaces And Dialogues In Graphical Environments

4.3 Designing Interfaces and Dialogues
4.3.1 Introduction

Interface design focuses on how information is provided to and captured from users; dialogue design focuses on how information is provided to and captured from users; dialogue design focuses on the sequencing of interface displays. Dialogues are analogous to a conversation between two people. The grammatical rules followed by each person during a conversation are analogous to the interface. Thus the design of the interfaces and dialogues is the process of defining the manner in which humans and computers exchange information. A good human computer interface provides a uniform structure for finding, viewing, and invoking the different components of a system.

Measures of Usability

- User-friendliness is a term often used, and misused, to describe system usability. Although the term is widely used, it is too vague from a design standpoint to provide adequate information because it means different things to different people. Consequently, most development groups use several methods for assessing usability, including the following considerations:

Learnability, Efficiency, Error rate, Memorability, Satisfaction and aesthetics

- **Learnability:** How difficult is it for a user to perform a task for the first time?
- **Efficiency:** How quickly can users perform tasks once they know how to perform them?
- **Error rate:** How many errors might a user encounter, and how easy it is to recover from those errors?
- **Memorability**: How easy it is to remember how to accomplish a task when revisiting the system after some period of time?
- **Satisfaction and aesthetics:** How enjoyable is the system's visual appeal and how enjoyable is the system to use?

In assessing usability, you can collect information by observation, interviews, keystroke capturing, and questionnaires. Time to learn reflects how long it takes the average system user to become proficient using the system. Equally important is the extent to which users remember how to use inputs and outputs over time.

Characteristics for considering when designing forms and reports

Characteristics	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.

4.2.2Designing Interfaces and Dialogues

Similar to designing forms and reports, the process of designing interfaces and dialogues is a user-focused activity. This means that you follow a prototyping methodology of iteratively collecting information, constructing a prototype, assessing usability, and making refinements. To design usable interfaces and dialogues, you must answer the same who, what, when, where and how questions used to guide the design of forms and reports. Thus, this process parallels that of designing forms and reports.

Deliverables and outcomes

The deliverable and outcome from system interface and dialogue design is the creation of a design specification. This specification is also similar to the specification produced for form and report designs – with one exception:

1. Narrative overview
2. Sample design
3. Testing and usability assessment

For interface and dialogue designs, one additional subsection is included: a section outlining the dialogue sequence – the ways a user can move from one display to another.

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

4.3.3 Interaction Methods And Devices

The human-computer interface defines the ways in which users interact with an information system. All human-computer interfaces must have an interaction style and use some hardware devices for supporting this interaction.

Method of Interacting

When designing the user interface, the most fundamental decision you make relates to the methods used to interact with the system. Given that there are numerous approaches for designing the interaction, we briefly provide a review of those most commonly used. Our review will examine the basics of five widely used styles: command language, menu, form, object and natural language. We will also describe several devices for interacting, focusing primarily on their usability for various interaction activities.

- **Command Language Interacting**
- In command language interaction, the user enters explicit statements to invoke operations within a system. This type of interaction requires users to remember command syntax and semantics. For example, to rename a copy of a file called “file.doc” in the current directory as “newfile.doc” at the command prompt within Linux, you would type: $ cp file.doc newfile.doc
- Command language interaction places a substantial burden on the user to remember names, syntax, and operations. Most newer or large-scale systems no longer rely entirely on a command language interface. Yet command languages are good for experienced users, for systems with a limited command set, and for rapid interaction with the system.

Menu Interaction

- A menu is simply a list of options; when an option is selected by the user, a specific command is invoked or another menu is activated. Menus have become the most widely used interface methods because the user only needs to understand simple signposts and route operations to effectively navigate through a system.
- Menus can differ significantly in their design and complexity. The variation of their design is most often related to the capabilities of the development environment, the skills of the developer, and the size and complexity of the system. For smaller and less complex systems with limited system options, you may use a single menu or a linear sequence of menus. A single menu has obvious advantages over a command language but may provide little guidance beyond invoking the command.
- For large and more complex systems, you can use menu hierarchies to provide navigation between menus. These hierarchies can be simple tree structures or variations wherein children menus have multiple parent menus. Some of these hierarchies may allow multilevel traversal variations as to how menus are arranged can greatly influence the usability of a system.

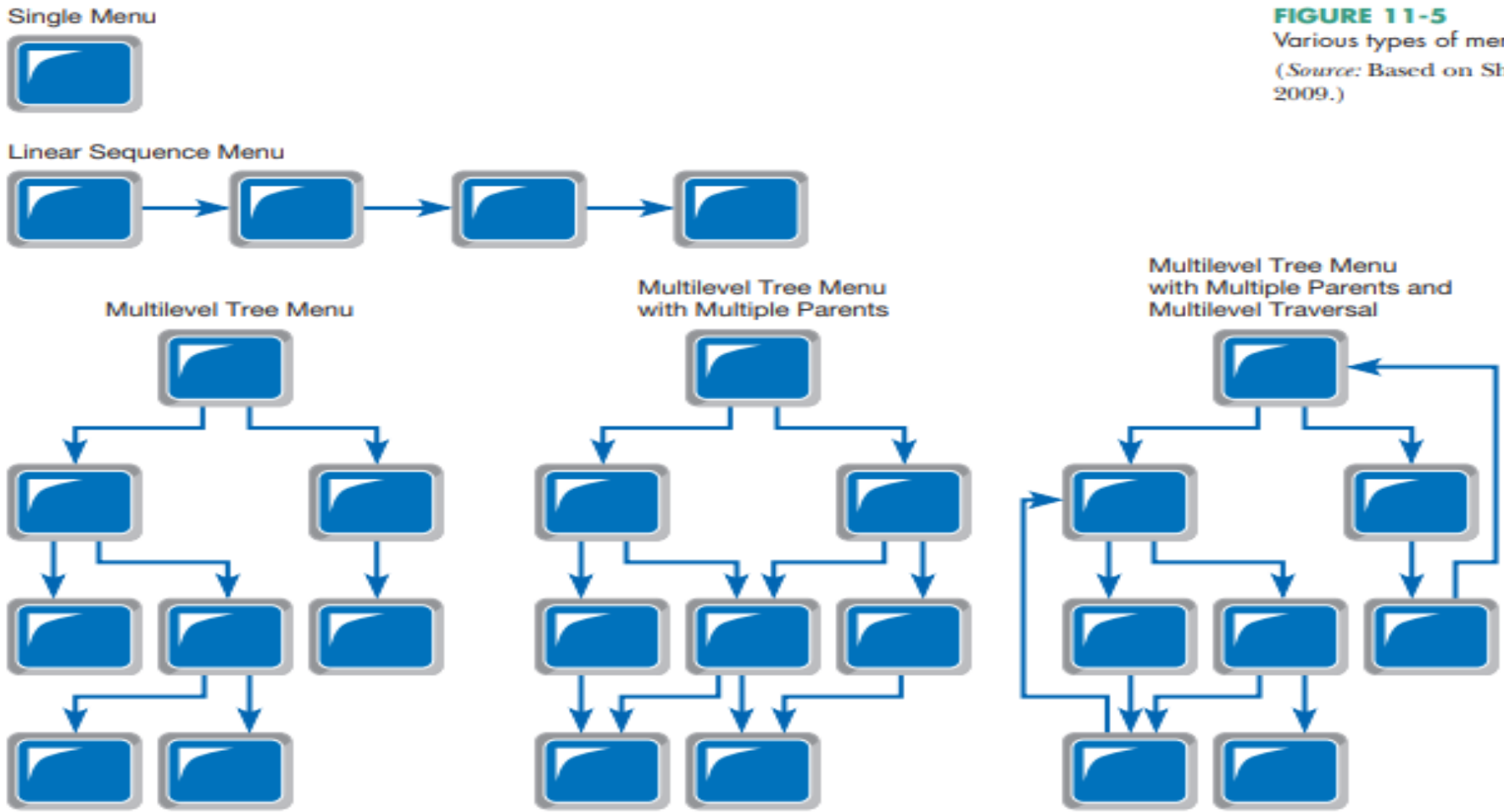


FIGURE 11-5
Various types of menu configurations
(*Source:* Based on Shneiderman et al., 2009.)

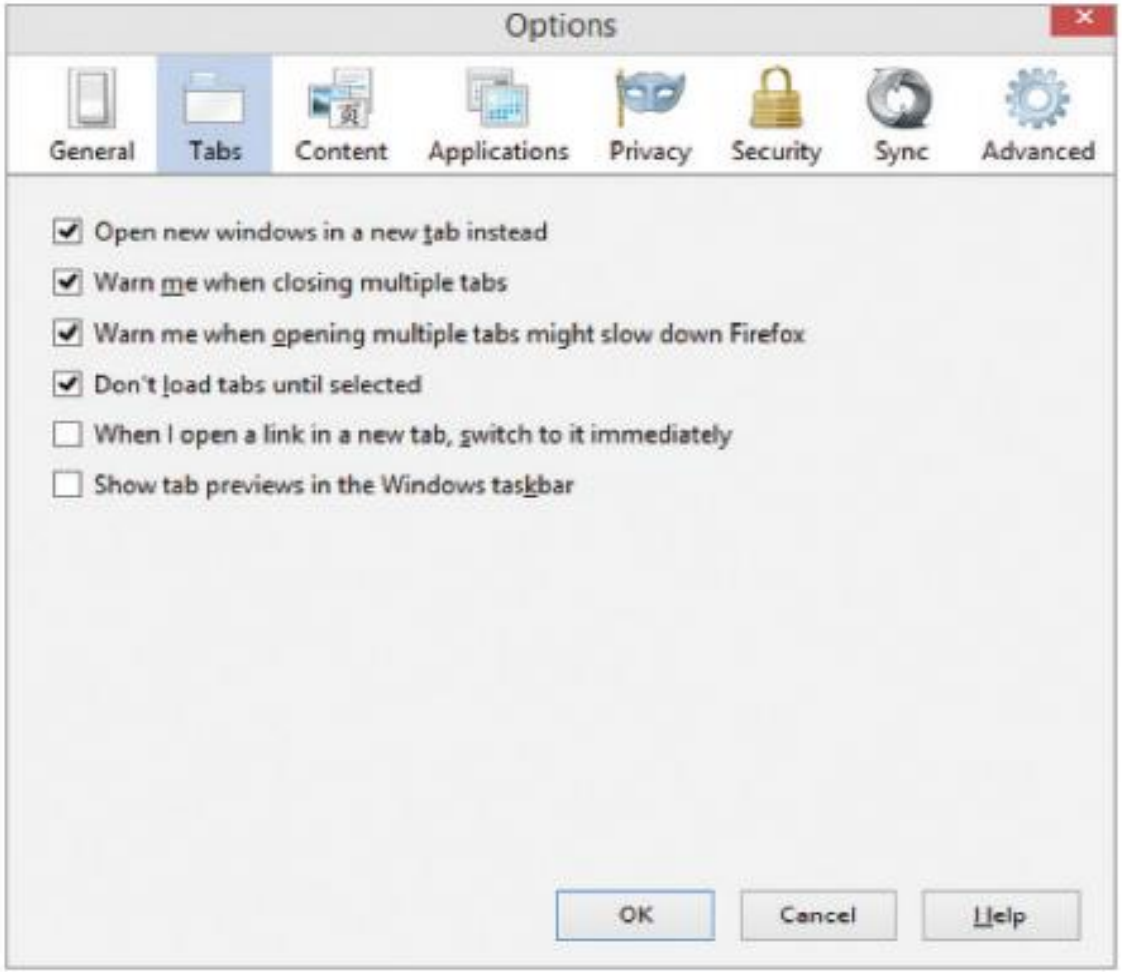
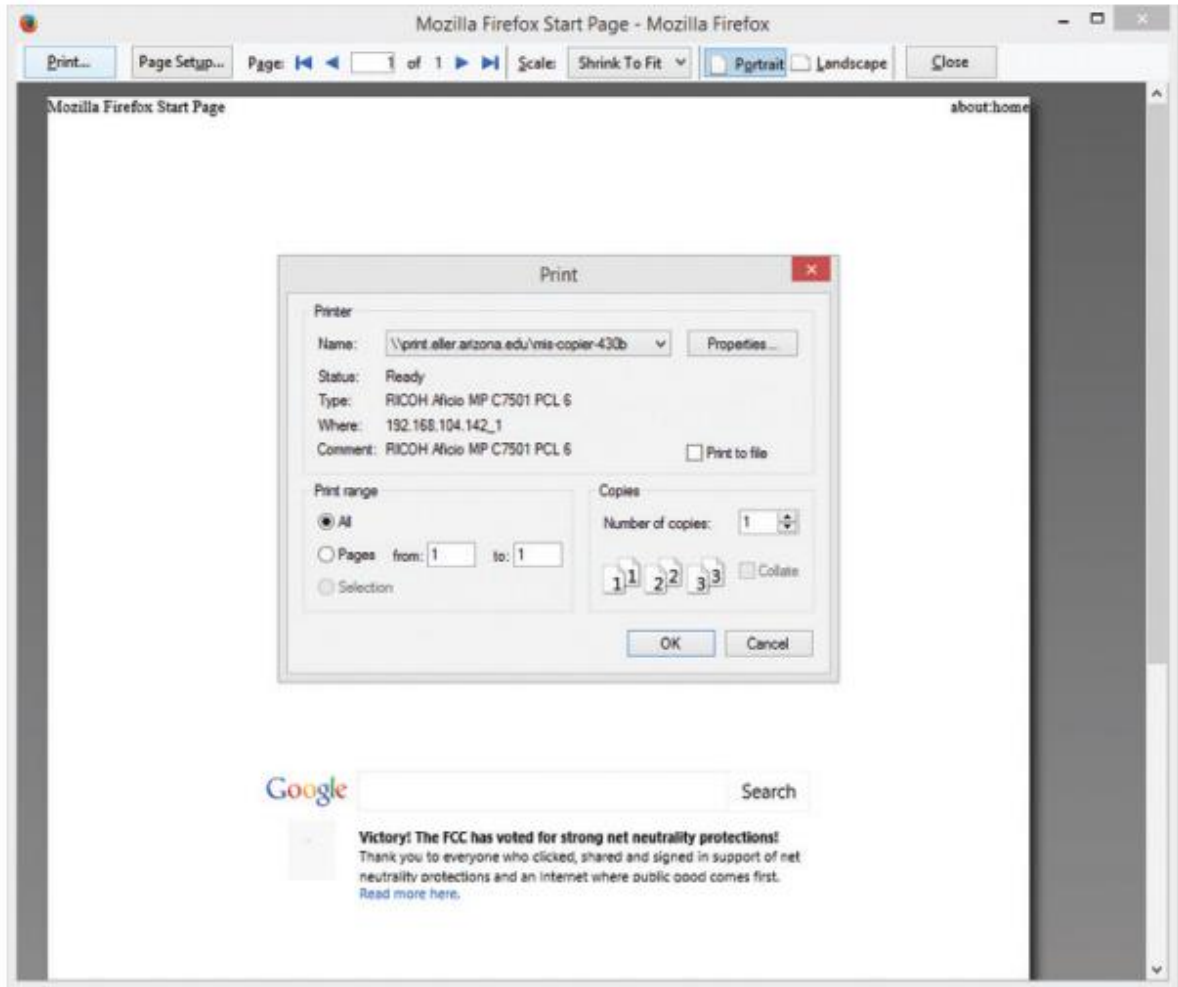


FIGURE 11-4
Single-level menu on the "Tabs" settings under the "Options" menu in the Firefox web browser

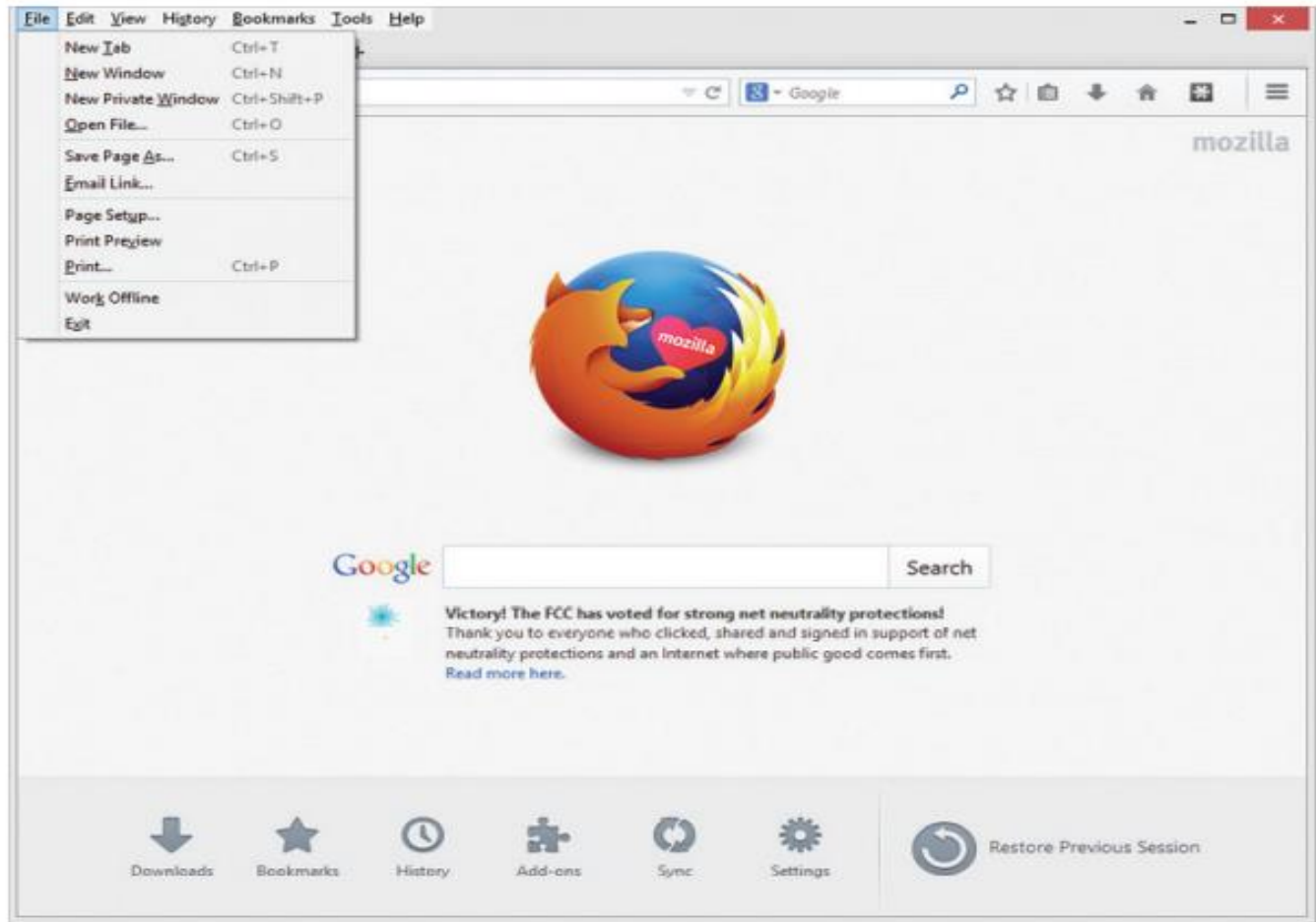
(*Source:* Mozilla Firefox.)

- There are two common methods for positioning menus. With a **pop-up menu** (also called a dialogue box), menus are displayed near the current cursor position so users don’t have to move the position or their eyes to view system options.
- With a **drop-down menu**, menus drop down from the top line of the display. Drop-down menus have become very popular in recent years because they provide consistency in menu location and operation among applications and efficiently use display space.

FIGURE 11-6
Menus from Mozilla Firefox
(a) Pop-up menu
(*Source:* Mozilla Firefox.)



(b) Drop-down menu

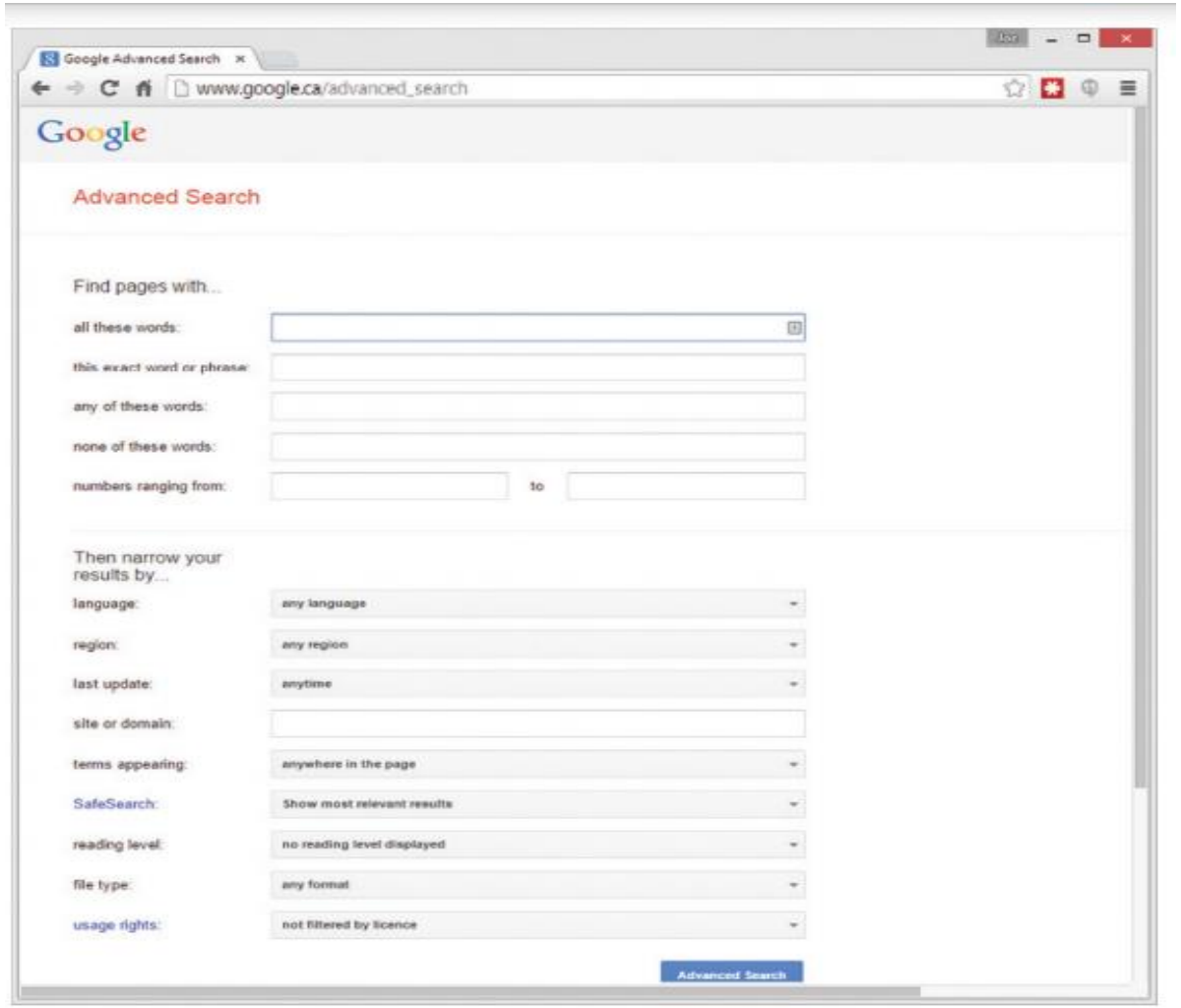


Guidelines for Menu Design

Working	- Each Menu should have a meaningful title. - Command verbs should clearly and specifically describe operations. - Menu items should be displayed in mixed uppercase and lowercase letters and have a clear, unambiguous interpretation.
Organization	- A consistent organizing principle should be used that relates to the tasks the intended users perform, for example, related options should be grouped together, and the same option should have the same wording and codes each time it appears.
Length	- The number of menu choices should not exceed the length of the screen. - Submenus should be used to break up exceedingly long menus.
Selection	- Selection and entry methods should be consistent and reflect the size of the application and sophisticated of the users. - How the user is to select each option and the consequences of each option should be clear (eg: whether another menu will appear)
Highlighting	- Highlighting should be minimized and used only to convey selected options(e.g. a check mark) or unavailable options(e.g. dimmed text)

Form Interaction

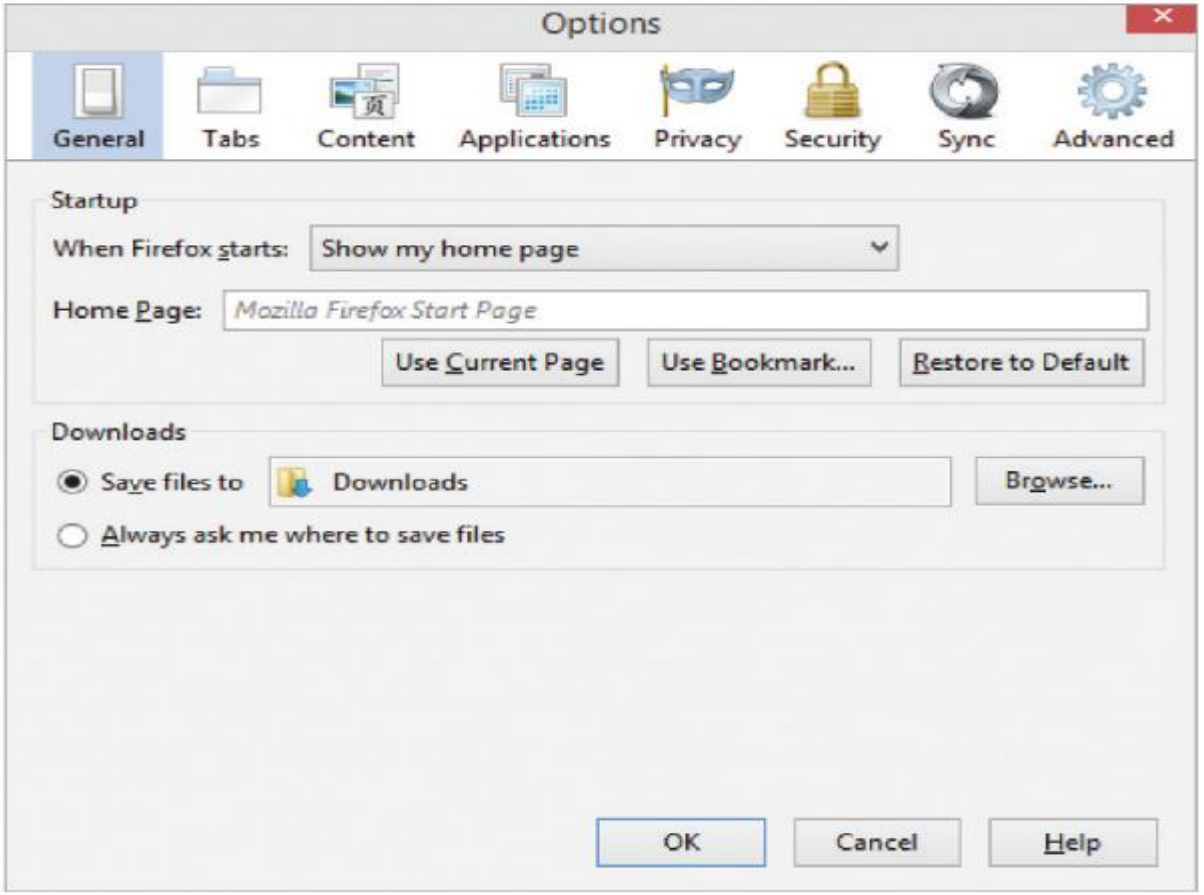
- The premise of form interaction is to allow users to fill in the blanks when working with a system. Form interaction is effective for both the input and presentation of information. An effectively designed form includes a self-explanatory title and fields beadings, has fields organized into logical groupings with distinctive boundaries, provides default values when practical, displays data in appropriate field lengths, and minimizes the need to scroll windows.
- Form interaction is the most commonly used method for data entry and retrieval in business-based systems. Using interactive forms, organizations can easily provide all types of information to web surfers.



Form interaction
A highly intuitive human–computer interaction method whereby data fields are formatted in a manner similar to paper-based forms.

Object-based Interaction

- The most common method for implementing object based interaction is through the use of icons. Icons are small graphical symbols that look like the processing option they are meant to represent. Users select operations by pointing to the appropriate icon with the some type of pointing device.
- The primary advantages to icons are that they take up little screen space and can be quickly understood by most users. An icon may also look like a button that, when selected or depressed, causes the system to take an action relevant to that form, such as cancel, save, edit a record, or ask for help.



Object-based interaction
A human–computer interaction method in which symbols are used to represent commands or functions.

Natural Language Interaction

- One branch of artificial intelligence research studies techniques for allowing systems to accept inputs and produce outputs in a conventional language such as English.
- This method of interaction is referred to as natural language interaction. Presently, natural language interaction is not as viable an interaction style as the other methods presented.
- Current implementations can be tedious, frustrating, and time consuming for the user and are often built to accept input in narrowly constrained domains (e.g. database queries).
- Natural Language interaction is being applied with both keyboard and voice entry systems.

Natural language interaction
A human–computer interaction method whereby inputs to and outputs from a computer-based application are in a conventional spoken language such as English.

TABLE 11-2 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.

Designing Interfaces

- User interface is the front-end application view to which user interacts in order to use the software.
- User can manipulate and control the software as well as hardware by means of user interface.
- Today, user interface is found at almost everyplace where digital technology exists, right from computers, mobile phones, cars, music players, airplanes, ships etc.
- User interface is part of software and is designed such a way that it is expected to provide the user insight of the software.
- UI provides fundamental platform for human-computer interaction.
- UI can be graphical, text-based, audio-video based, depending upon the underlying hardware and software combination.

- UI can be hardware or software or a combination of both. The software becomes more popular if its user interface is:

❑ Attractive

❑ Simple to use

❑ Responsive in short time

❑ Clear to understand

❑ Consistent on all interfacing screens

Designing Layouts

- To ease user training and data recording, you should use standard formats for computer-based forms and reports similar to those used on paper-based forms and reports for recording or reporting information. This form has several general areas common to most forms:

- ❑ Header information
- ❑ Sequence and time-related information
- ❑ Instruction or formatting information
- ❑ Body or data details
- ❑ Totals or data summary
- ❑ Authorization or signatures
- ❑ Comments

- In many organizations, data are often first recorded on paper-based forms and then later recorded within an application systems. When designing layouts to record or information on paper-based forms, you should try to make both as similar as possible. Additionally, data entry displays should be consistently formatted across applications to speed data entry and reduce errors.

- Another concern when designing the layout of computer-based forms is the design of between-field navigation. Because you can control the sequence for users to move between fields, standard screen navigation should flow from left to right and top to bottom just as when you work on paper-based forms.
- When designing the navigation procedures within your system, flexibility and consistency are primary concerns. Users should be able to freely move forward and backward or to any desired data entry fields. Users should be able to navigate each form in the same way or in as similar a manner as possible. Additionally, data should not usually be permanently saved by the system until the user makes an explicit request to do so. This allows the user to abandon a data entry screen, back up, or move forward without adversely affecting the contents of the permanent data.

FIGURE 11-12
Computer-based form reporting customer sales activity (Pine Valley Furniture)
(*Source:* Microsoft Corporation.)

Designing Dialogues

- The process of designing the overall sequences that users follow to interact with an information system is called dialogue design. A dialogue is the sequence on which information is displayed to and obtained from a user. As the designer, our role is to select the most appropriate interaction methods and devices to define the conditions under which information is displayed to and obtained from users. The dialogue design process consists of three major steps:
  - Designing the dialogue sequence
  - Building a prototype
  - Assessing usability
- For a dialogue to have high usability, it must be consistent in form, function and style. All other rules regarding dialogue design are mitigated by the consistency guideline. For example, the effectiveness of how well errors are handled or feedback is provided will be significantly influenced by consistency in design. If the system does not consistently handle errors, the user will often be at a loss as to why certain things happen.

Guidelines for the Design of Human-Computer Dialogues

Guideline	Explanation
Consistency	Dialogues should be consistent in sequence of actions, keystrokes, and terminology (e.g. the same labels should be used for the same operations on all screens, and the location of the same information should be the same on all displays.)
Shortcuts and Sequence	Allow advanced users to take shortcuts using special keys (e.g. CTRL + C to copy highlighted text). A natural sequence of steps should be followed (e.g. enter first name before last name, if appropriate).
Feedback	Feedback should be provided for every user action (e.g. confirm that a record has been added, rather than simply putting another blank form on the screen).
Closure	Dialogues should be logically grouped and have a beginning, middle, and end (e.g. the last in the sequence of screens should indicate that there are no more screens.)
Error Handling	All errors should be detected and reported; suggestions on how to proceed should be made (e.g. suggest why such errors occur and what user can do to correct the error). Synonyms for certain responses should be accepted (e.g. accept either " t"," T" , or " TRUE").
Reversal	Dialogues should, when possible, allow the user to reverse actions (e.g. undo a deletion); data should not be deleted without confirmation (e.g. display all the data for a record the user has indicated is to be deleted).

Guideline	Explanation
Control	Dialogues should make the user (especially an experienced user) feel in control of the system (e.g. provide a consistent response time at a pace acceptable to the user).
Base	It should be a simple process for users to enter information and navigate between screens (e.g: provide means to provide forward, backward, and to specific screens, such as first and last.)

Designing the Dialogue Sequence

- Your first step in dialogue design is to define the sequence. In other words, you must first gain an understanding of how users might interact with the system. This means that you must have a clear understanding of user, task, technological, and environmental characteristics when designing dialogues. Suppose that the marketing manager at Pine Valley Furniture (PVF) wants sales and marketing personnel to be able to review the year-to-date transection activity for any PVF customer. After talking with the manger, you both agree that a typical dialogue between a user and the Customer Information System for obtaining this information might proceed as follows:

- ❑ Request to view individual customer information
- ❑ Specify the customer of interest
- ❑ Select the year-to-date transaction summary display
- ❑ Review customer information
- ❑ Leave system

Dialogue diagramming
A formal method for designing and representing human–computer dialogues using box and line diagrams.

- As a designer, once you understand how a user wishes to use a system, you can then transform these activities into a formal dialogue specification.

A formal method for designing and representing dialogues is dialogue diagramming. Dialogue diagrams have only one symbol, a box with three sections; each box represents one display (which might be a full screen or a specific form or window) within a dialogue. The three sections of the box are used as follows:

1. Top: Contains a unique display reference number used by other displays for referencing it.
2. Middle: Contains the name or description of the display.
3. Bottom: Contains display reference numbers that can be accessed from the current display

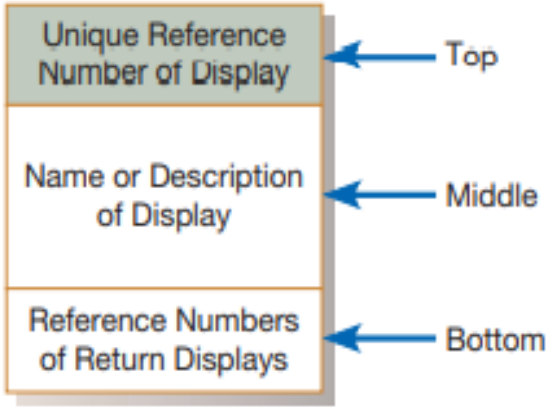
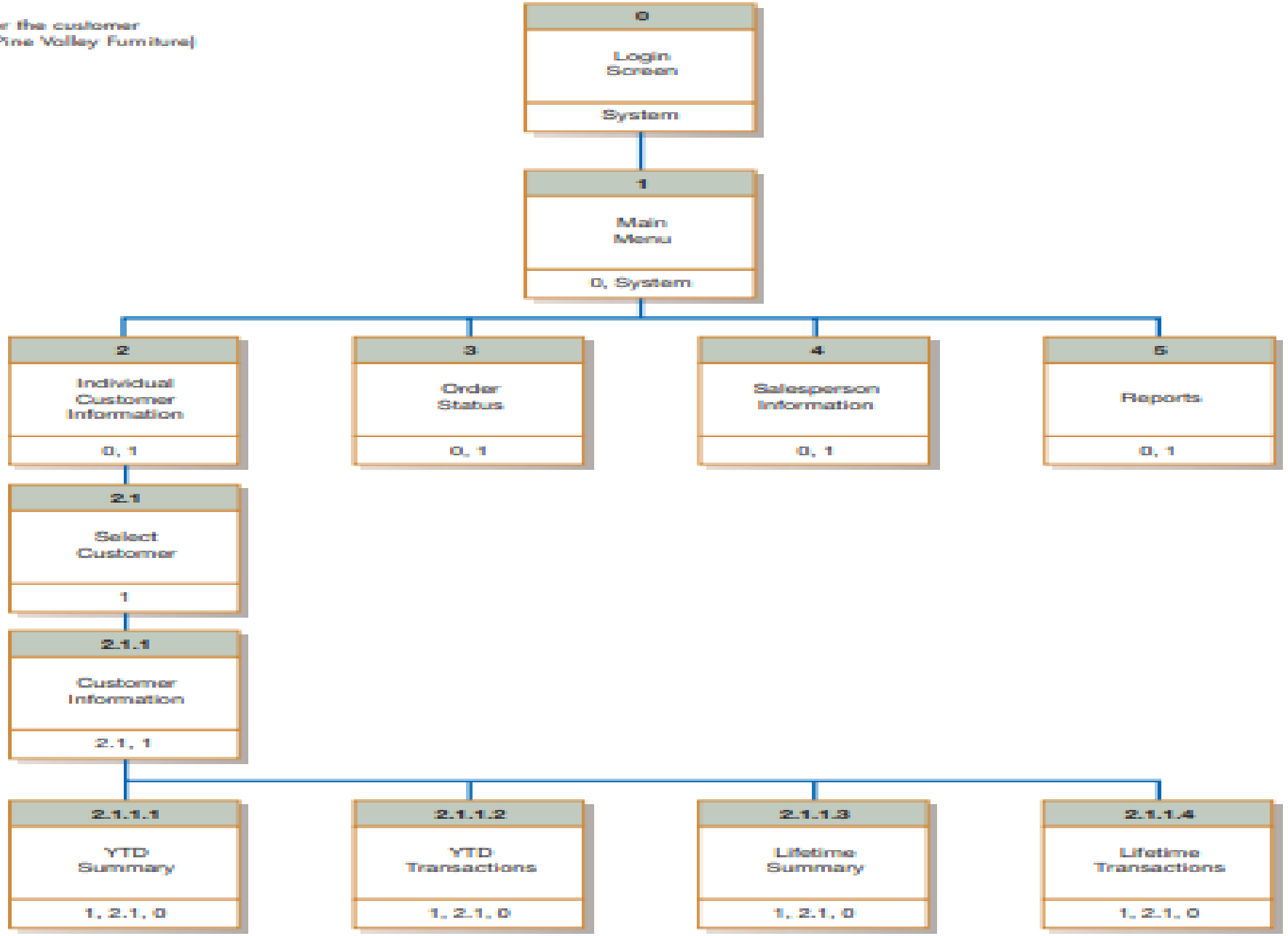


FIGURE 11-19
Dialogue diagram for the customer information system (Pine Valley Furniture)



4.3.4 Designing Interfaces And Dialogues In Graphical Environments

- Graphical User Interface (GUI) environments have become the de facto standard for human-computer interactions. Although all of the interface and dialogue design guidelines presented previously apply to designing GUIs, additional issues that are unique to these environments must be considered. Here, we briefly discuss some of these issues.

Graphical Interface Design Issues

- When designing GUIs for an operating environment such as Microsoft Windows or the Apple OSX, numerous factors must be considered. Some factors are common to all GUI environments, whereas others are specific to a single environment. We will not, however, discuss the subtleties (refinements/sensitivities) and details of any single environment. We will not however, discuss the subtleties and details of any single environment. Instead, our discussion will focus on a few general truths that experienced designers mention as critical to the design of usable GUIs.

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

❑ Become an expert user of the GUI environment.

❑ Understand the available resources and how they can be used.

- The first step should be an obvious one. The greatest strength of designing within a standard operating environment is that standards for the behavior of most system operations have already been defined.
- For example, how you cut and paste, set up your default printer, design, design menus, or assign commands to functions have been standardized both within and across applications.

- This allows experienced users of one GUI-based application to easily learn a new application. Thus, in order to design effective interfaces in such environments, you must first understand how other applications have been designed so that you will adopt the established standards for “look and feel.”
- Failure to adopt the standard conventions in a given environment will result in a system that will likely frustrate and confuse users.

FIGURE 11-21
Highlighting GUI design standards
(*Source:* Screenshot from Neuro-ID, Inc. Used with permission.)

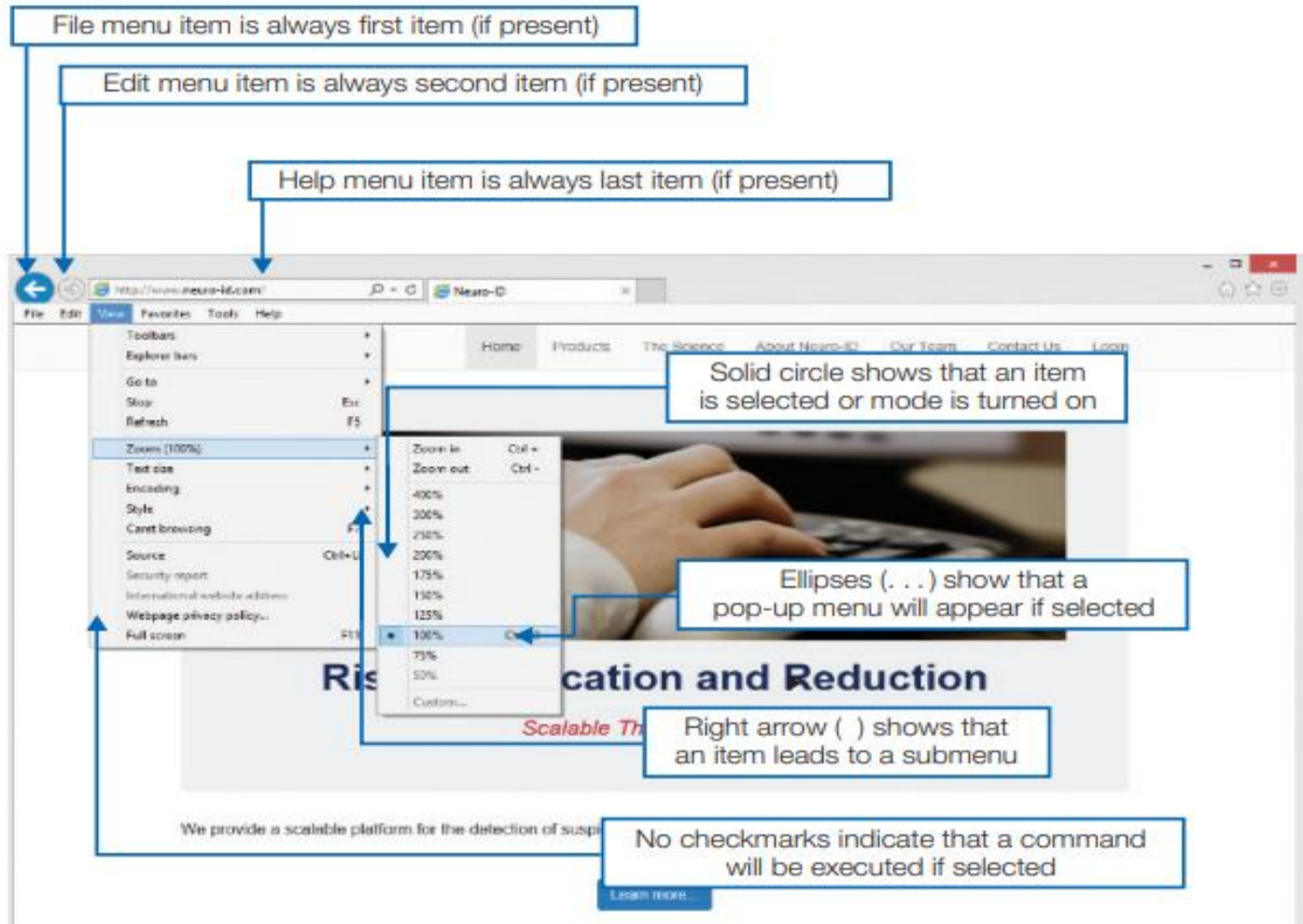


TABLE 11-14 Common Properties of Windows and Forms in a GUI Environment That Can Be Active or Inactive

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)

(Source: Based on Wagner, 1994.)

Dialogue Design Issues in a Graphical Environment

- When designing a dialogue, your goal is to establish the sequence of displays (full screens or windows) that users will encounter when working with the system. This process can be a bit more challenging due to the GUI's ability to suspend activities (without resolving a request for information or exiting the application altogether) and switch to another application or task. For example, within Microsoft Word, the spell-checker executes independently from the general word processor.
- This means that you can easily jump between the spell-checker and word processor without exiting either one. Conversely, when selecting the print operation, you must either initiate printing or abort the request before returning to the word processor. T
- hus, Windows-type environments allow you to create forms that either require the user to resolve a request before proceeding (print example) or selectively choose to resolve a request before proceeding (the spell-checker). Creating dialogues that allow the user to jump from application to application or from module to module within a given application requires that you carefully think through the design of dialogues.

- One easy way to deal with the complexity of designing advanced GUIs is to require users to always resolve all requests for information before proceeding. For such designs, the dialogue diagramming technique is an adequate design tool. This, however, would make the system operate in a manner similar to a traditional non-GUI environment where the sequencing of displays is tightly controlled.
- The drawback to such an approach would be the failure to capitalize on the task-switching capabilities of these environments. Consequently, designing dialogues in environments where the sequence between displays cannot be predetermined offers significant challenges to the designer. Using tools such as dialogue diagramming helps analysts to better manage the complexity of designing graphical interfaces.

Database Normalization With Examples

- Database Normalization Example can be easily understood with the help of a case study. Assume, a video library maintains a database of movies rented out. Without any normalization in database, all information is stored in one table as shown below. Let's understand Normalization database with normalization example with solution:

FULL NAMES	PHYSICAL ADDRESS	MOVIES RENTED	SALUTATION
Janet Jones	First Street Plot No 4	Pirates of the Caribbean, Clash of the Titans	Ms.
Robert Phil	3rd Street 34	Forgetting Sarah Marshal, Daddy's Little Girls	Mr.
Robert Phil	5th Avenue	Clash of the Titans	Mr.

1NF (First Normal Form) Rules

- Each table cell should contain a single value.
- Each record needs to be unique.
- The above table in 1NF-

1NF Example

FULL NAMES	PHYSICAL ADDRESS	MOVIES RENTED	SALUTATION
Janet Jones	First Street Plot No 4	Pirates of the Caribbean	Ms.
Janet Jones	First Street Plot No 4	Clash of the Titans	Ms.
Robert Phil	3rd Street 34	Forgetting Sarah Marshal	Mr.
Robert Phil	3rd Street 34	Daddy's Little Girls	Mr.
Robert Phil	5th Avenue	Clash of the Titans	Mr.

Example of 1NF in DBMS

What is a KEY in SQL?

A **KEY in SQL** is a value used to identify records in a table uniquely. An SQL KEY is a single column or combination of multiple columns used to uniquely identify rows or tuples in the table. SQL Key is used to identify duplicate information, and it also helps establish a relationship between multiple tables in the database.

- Note: Columns in a table that are NOT used to identify a record uniquely are called non-key columns.

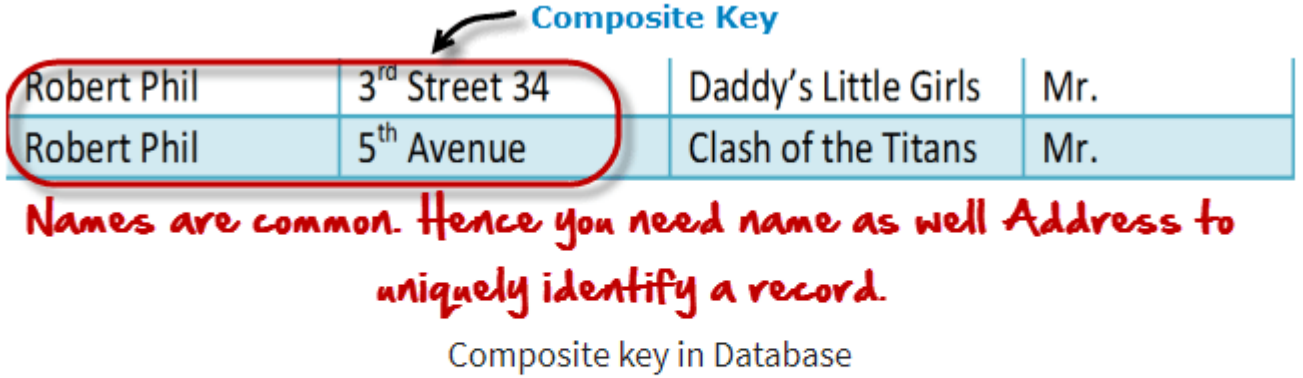
What is a Primary Key?

- A primary is a single column value used to identify a database record uniquely.
- It has following attributes
- A primary key cannot be NULL
- A primary key value must be unique

- The primary key values should rarely be changed
- The primary key must be given a value when a new record is inserted.

What is Composite Key?

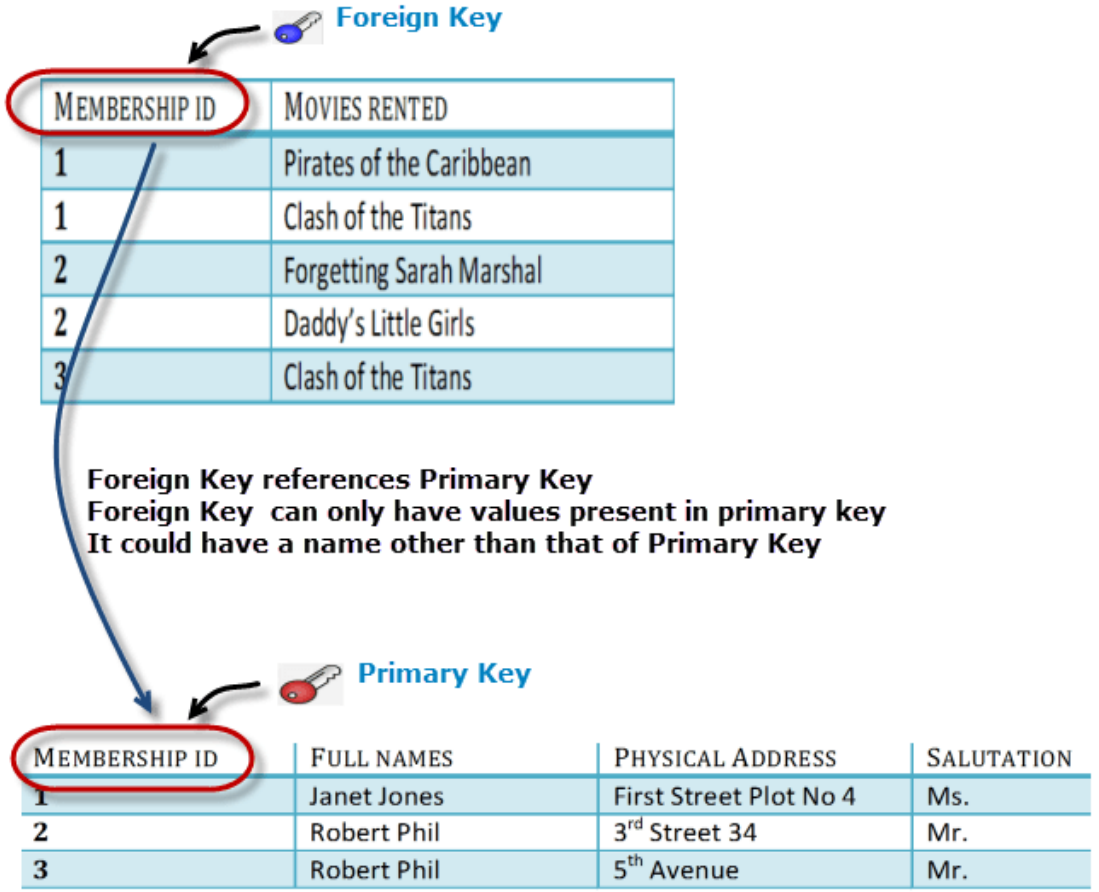
- A composite key is a primary key composed of multiple columns used to identify a record uniquely
- In our database, we have two people with the same name Robert Phil, but they live in different places.



Composite key in Database

Hence, we require both Full Name and Address to identify a record uniquely. That is a composite key.

- Database - Foreign Key
- In Table 2, Membership_ID is the Foreign Key
- Foreign Key references the primary key of another Table! It helps connect your Tables
- A foreign key can have a different name from its primary key
- It ensures rows in one table have corresponding rows in another
- Unlike the Primary key, they do not have to be unique. Most often they aren't
- Foreign keys can be null even though primary keys can not



2NF (Second Normal Form) Rules

- Rule 1- Be in 1NF
- Rule 2- Single Column Primary Key that does not functionally dependant on any subset of candidate key relation
- It is clear that we can't move forward to make our simple database in 2nd Normalization form unless we partition the table above.

MEMBERSHIP ID	FULL NAMES	PHYSICAL ADDRESS	SALUTATION
1	Janet Jones	First Street Plot No 4	Ms.
2	Robert Phil	3rd Street 34	Mr.
3	Robert Phil	5th Avenue	Mr.

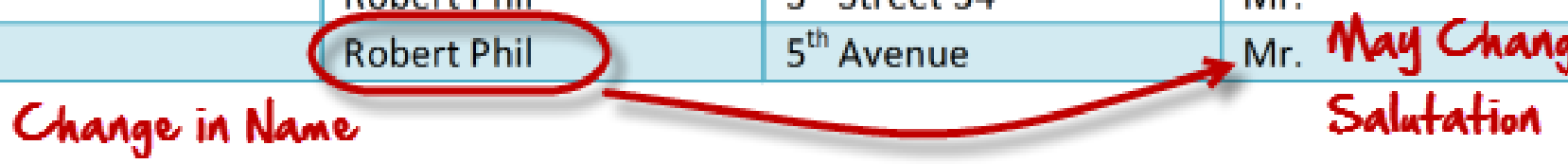
MEMBERSHIP ID	MOVIES RENTED
1	Pirates of the Caribbean
1	Clash of the Titans
2	Forgetting Sarah Marshal
2	Daddy's Little Girls
3	Clash of the Titans

We have divided our 1NF table into two tables viz. Table 1 and Table2. Table 1 contains member information. Table 2 contains information on movies rented.

We have introduced a new column called Membership_id which is the primary key for table 1. Records can be uniquely identified in Table 1 using membership id

- What are transitive functional dependencies?
- A transitive functional dependency is when changing a non-key column, might cause any of the other non-key columns to change
- Consider the table 1. Changing the non-key column Full Name may change Salutation.

MEMBERSHIP ID	FULL NAMES	PHYSICAL ADDRESS	SALUTATION
1	Janet Jones	First Street Plot No 4	Ms.
2	Robert Phil	3rd Street 34	Mr.
3	Robert Phil	5th Avenue	Mr.



3NF (Third Normal Form) Rules

- Rule 1- Be in 2NF
- Rule 2- Has no transitive functional dependencies
- To move our 2NF table into 3NF, we again need to again divide our table.

We have again divided our tables and created a new table which stores Salutations.
There are no transitive functional dependencies, and hence our table is in 3NF
In Table 3 Salutation ID is primary key, and in Table 1 Salutation ID is foreign to primary key in Table 3

Membership ID	Full names	Physical Address	Salutation ID
1	Janet Jones	First Street Plot No 4	2
2	Robert Phil	3rd Street 34	1
3	Robert Phil	5th Avenue	1

Membership ID	Movies rented
1	Pirates of the Caribbean
1	Clash of the Titans
2	Forgetting Sarah Marshal
2	Daddy's Little Girls
3	Clash of the Titans

Salutation ID	Salutation
1	Mr.
2	Ms.
3	Mrs.
4	Dr.

- **BCNF (Boyce-Codd Normal Form)**
- Even when a database is in 3rd Normal Form, still there would be anomalies resulted if it has more than one **Candidate** Key.
- Sometimes is BCNF is also referred as **3.5 Normal Form.**
- **4NF (Fourth Normal Form) Rules**
- If no database table instance contains two or more, independent and multivalued data describing the relevant entity, then it is in 4th Normal Form.
- **5NF (Fifth Normal Form) Rules**
- A table is in 5th Normal Form only if it is in 4NF and it cannot be decomposed into any number of smaller tables without loss of data.
- **6NF (Sixth Normal Form) Proposed**
- 6th Normal Form is not standardized, yet however, it is being discussed by database experts for some time. Hopefully, we would have a clear & standardized definition for 6th Normal Form in the near future...
- Reference:https://www.guru99.com/database-normalization.html

Electronic Commerce Applications: Designing Forms and reports for Pine Valley Furniture's Webstore

- Designing the forms and reports for an Internet-based electronic commerce application is a central and critical design activity. Because this is where a customer will interact with a company, much care must be put into its design. Like the process followed when designing the forms and reports for other types of systems, a prototyping design process is most appropriate.
- Although the techniques and technology for building Internet sites are rapidly evolving, several general design guidelines have emerged. In this section, we examine some of these as they apply to the design of PVF's WebStore.
- When designing forms and reports, there are several errors that are specific to website design. It is unfortunately beyond the scope of this book to critically examine all possible design problems within contemporary websites. Here, we will simply summarize those errors that commonly occur and that are particularly detrimental to the user's experience.

TABLE 10-11 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.

Designing Forms and Reports at Pine Valley Furniture

- When Jim Woo and the PVF development team focused on designing the forms and reports (i.e., the “pages”) for the WebStore, they first reviewed many popular electronic commerce websites. From this review, they established the following design guidelines:
- Use lightweight graphics.
- Establish forms and data integrity rules.
- Use stylesheet-based HTML.
- In order to ensure that all team members understood what was meant by each guideline, Jim organized a design briefing to explain how each guideline would be incorporated into the WebStore interface design.

Lightweight Graphics

Lightweight graphics
Small, simple images that allow a web page to be displayed more quickly.

In addition to easy menu and page navigation, the PVF development team wants a system where web pages load quickly. A technique that can assist in making pages load quickly is the use of lightweight graphics. **Lightweight graphics** are small, simple images that allow a page to load as quickly as possible. "Using lightweight graphics allows pages to load quickly and helps users to reach their final location in the site—hopefully the point of purchase area—as quickly as possible. Large color images will only be used for displaying detailed product pictures that customers explicitly request to view," explained Jim. Experienced web designers have found that customers are not willing to wait at each hop of navigation for a page to load, just so they have to click and wait again. The quick feedback that a website with lightweight graphics can provide will help to keep customers at the WebStore longer.

Forms and Data Integrity Rules

Because the goal of the WebStore is to have users place orders for products, all forms that request information should be clearly labeled and provide adequate room for input. If a specific field requires a specific input format such as a date of birth or phone number, it must provide a clear example for the user so that data errors can be reduced. Additionally, the site must clearly designate which fields are optional, which are required, and which have a range of values.

Jim emphasized, "All of this seems to be overkill, but it makes processing the data much simpler. Our site will check all data before submitting it to the server for processing. This will allow us to provide quicker feedback to the user on any data entry error and eliminate the possibility of writing erroneous data into the permanent database. Additionally, we want to provide a disclaimer to reassure our customers that the data will be used only for processing orders, that it will never be sold to marketers, and that it will be kept strictly confidential."

Stylesheet-Based HTML

When Jim talked with the consultants about the WebStore during the analysis phase, they emphasized the advantages of using **stylesheet-based HTML** He was told that when displaying individual products, it would be very advantageous to try to have a few "stylesheets" that could be used where appropriate in order to make sure that all pages within the WebStore had the same look and feel. Stylesheets describe how information will be presented (i.e., its style). Thus, stylesheet-based HTML design allows the content of a web page to remain separate from the way it is formatted. By separating the content from its formatting information, it is much easier to update the look and feel of the website and make sure that all pages have a similar appearance. Jim explained, "We need to look for ways to make the website consistent and easy to update. By using stylesheets, we not only get all the pages looking the same, we can also update the look of the website by making changes in a few stylesheets, rather than in hundreds of web pages. For example, a desk and a filing cabinet are two completely different products. Yet, stylesheets guarantee that their separate pages will be formatted and look the same way."

Stylesheet-Based HTML
A web design approach that separates content from the way in which it is formatted and presented, making ongoing maintenance easier and site-wide consistency much higher.

Electronic Commerce Applications: Designing Interfaces and Dialogues for Pine Valley Furniture's Webstore

- Designing the human interface for an Internet-based electronic commerce application is a central and critical design activity. Because this is where a customer will interact with a company, much care must be put into its design.
- Like the process followed when designing the interface for other types of systems, a prototyping design process is most appropriate when designing the human interface for an Internet electronic commerce system.
- Although the techniques and technology for building the human interface for Internet sites is rapidly evolving, several general design guidelines have emerged. In this section, we examine some of these as they apply to the design of PVF's WebStore.

General Guidelines

Over the years, interaction standards have emerged for virtually all of the commonly used desktop computing environments such as Windows or OSX. However, some interface design experts believe that the growth of the web has resulted in a big step backward for interface design. One problem, as discussed in Chapter 10, is that countless nonprofessional developers are designing commercial web applications. In addition to this, there are four other important contributing factors (Johnson, 2007):

1. The web's single "click-to-act" method of loading static hypertext documents (i.e., most buttons on the web do not provide click feedback)
2. Limited capabilities of most web browsers to support finely grained user interactivity
3. Limited agreed-upon standards for encoding web content and control mechanisms
4. Lack of maturity of web scripting and programming languages as well as limitations in commonly used web GUI component libraries

TABLE 11-15 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.

Designing Interfaces and Dialogues at Pine Valley Furniture

To establish design guidelines for the human–computer interface, Jim Woo and the PVF development team again reviewed many popular electronic commerce websites. The key feature they wanted to incorporate into the design was an interface with "menu-driven navigation with cookie crumbs." In order to ensure that all team members understood what was meant by this guideline, Jim organized a design briefing to explain how this feature would be incorporated into the WebStore interface.

Menu-Driven Navigation with Cookie Crumbs

After reviewing several sites, the team concluded that menus should stay in the exact same place throughout the entire site. They concluded that placing a menu in the same location on every page will help customers to more quickly become familiar with the site and therefore more rapidly navigate it. Experienced web developers know that the quicker customers can reach a specific destination at a site, the quicker they can purchase the product they are looking for or get the information they seek. Jim emphasized this point by stating, "These details may seem silly, but the second a user finds themselves 'lost' in our site, they're gone. One mouse click and they're no longer shopping at Pine Valley, but at one of our competitor's sites."

A second design feature, and one that is being used on many electronic commerce sites, is cookie crumbs (see Figure 11-22). **Cookie crumbs** are "tabs" or sequenced links on a web page that show a user where he or she is within a site and where he or she has been. These tabs or sequenced link are hypertext links that can be used to quickly move backward in the site. For example, suppose that a site is four levels deep, with the top level called "Log in," the second called "Shipping & Payment," the third called "Review Order," and the fourth called "Confirmation." As the user moves deeper into the site, a tab or sequenced link is displayed across the top of the page showing the user where he or she is, giving the user the ability to quickly jump backward one or more levels. In other words, when first entering the store, a tab or link will be displayed at the top (or some other standard place) of the screen with the word "Log in." After moving down a level, two links will be displayed, "Log in" and "Shipping & Payment." After providing the shipping and payment

Cookie crumbs
The technique of placing "tabs" or sequenced links on a web page that show a user where he or she is within a site and where he or she has been.

information on the second level, a third level is displayed where a user can review the order information. When this level is displayed, a third link is provided with the label "Review Order." Finally, if the customer decides to place an order and selects this option, a fourth-level screen is displayed and a fourth link is displayed with the label "Confirmation." In summary.

1. Level 1: Log in
2. Level 2: Log in → Shipping & Payment
3. Level 3: Log in → Shipping & Payment → Review Order
4. Level 4: Log in → Shipping & Payment → Review Order → Confirmation

By using cookie crumbs, users know exactly how far they have wandered from "home." If each tab is a link, users can quickly jump back to a broader part of the store should they not find exactly what they are looking for. Cookie crumbs serve two important purposes: First, they allow users to navigate to a point previously visited and will ensure that they are not lost. Second, they clearly show users where they have been and how far they have gone from home.

FIGURE 11-22
Cookie crumbs help users know where they are within a website