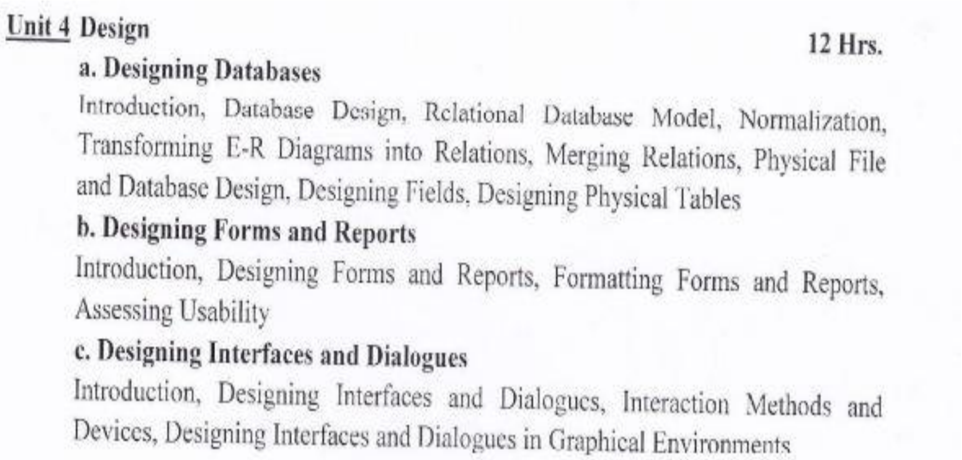
SYSTEM ANALYSIS AND DESIGN(SAD)

Chapter 3: **Design**



Unit 4 Design **12 Hrs.**

a. Designing Databases

Introduction, Database Design, Relational Database Model, Normalization, Transforming E-R Diagrams into Relations, Merging Relations, Physical File and Database Design, Designing Fields, Designing Physical Tables

b. Designing Forms and Reports

Introduction, Designing Forms and Reports, Formatting Forms and Reports, Assessing Usability

c. Designing Interfaces and Dialogues

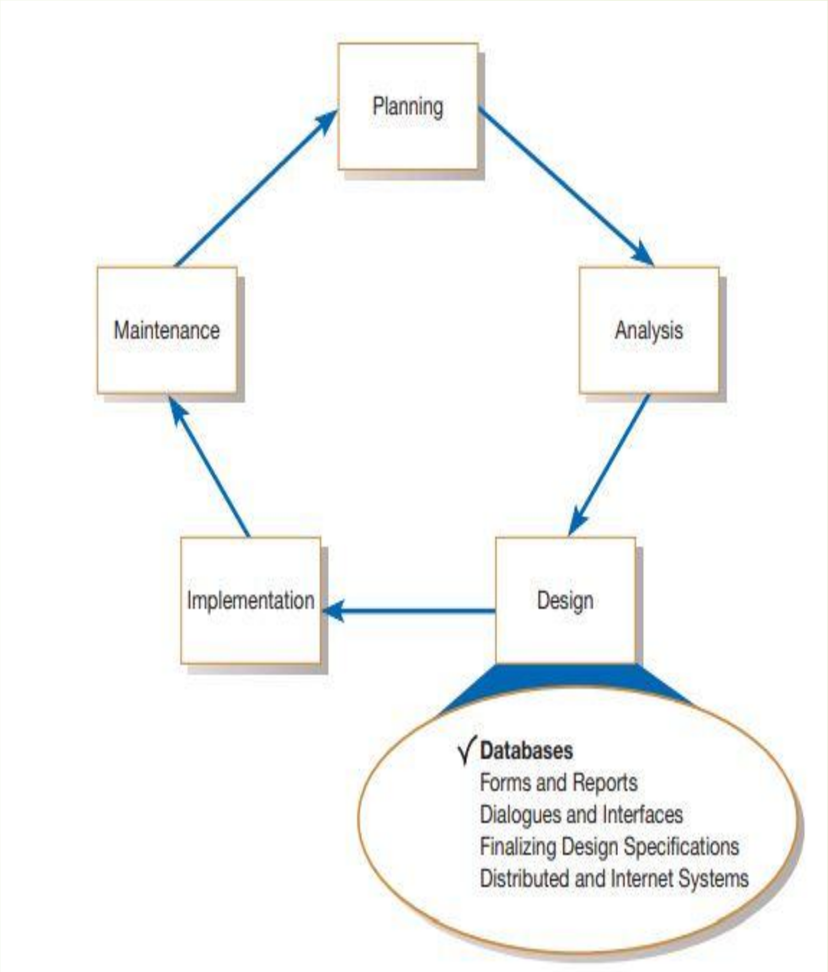
Introduction, Designing Interfaces and Dialogues, Interaction Methods and Devices, Designing Interfaces and Dialogues in Graphical Environments

Compiled By : Er. Omkar Nath Gupta

Designing Databases

Introduction

- 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.
- 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.(SRS=Software Requirements Specification)
- 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).

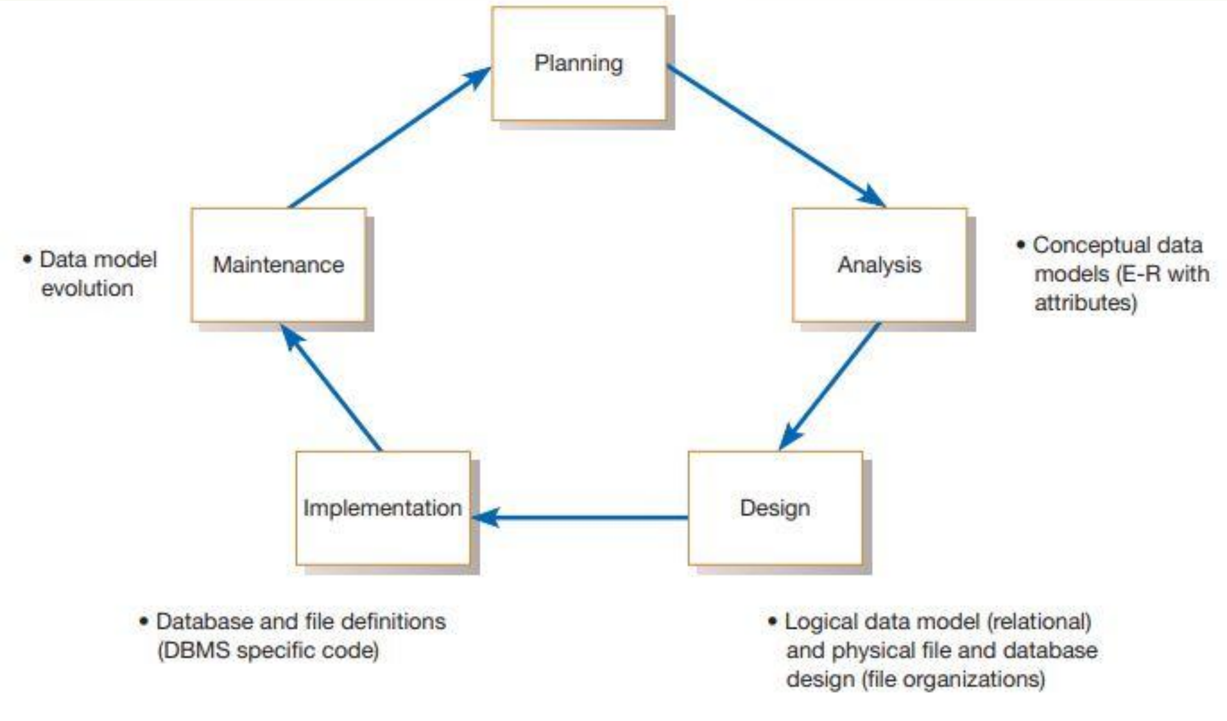


The Process of Database Design

- 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)



Planning

Analysis

- Conceptual data models (E-R with attributes)

Design

- Logical data model (relational) and physical file and database design (file organizations)

Implementation

- Database and file definitions (DBMS specific code)

Maintenance

- Data model evolution

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

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 new new data having to 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.

Relational Database Model:

- The relational database model represents data in the form of related tables, or relations. Each relation (or table) consists of a set of columns, called attributes and an arbitrary number of rows called records. Relational data model represents the logical databases 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.

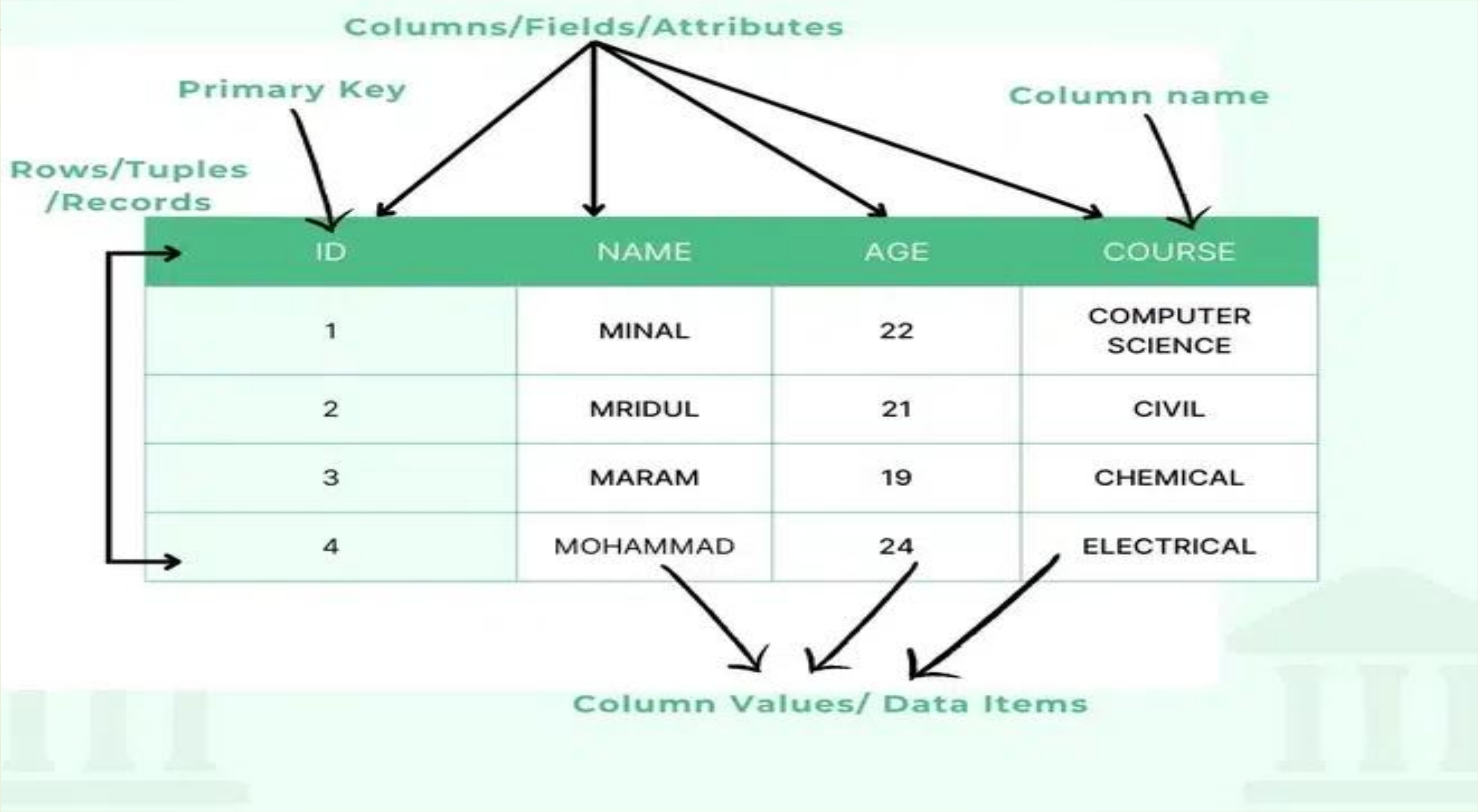
Attribute: Any relation have defined properties that are called an attributes. They are also called fields and columns.

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.

Example:



ID	NAME	AGE	COURSE
1	MINAL	22	COMPUTER SCIENCE
2	MRIDUL	21	CIVIL
3	MARAM	19	CHEMICAL
4	MOHAMMAD	24	ELECTRICAL

Characteristics/Properties of Relations:

→Each relation in a database must have a unique name.

→ A relation must not have two attributes with same name. Each attribute must have a distinct name.

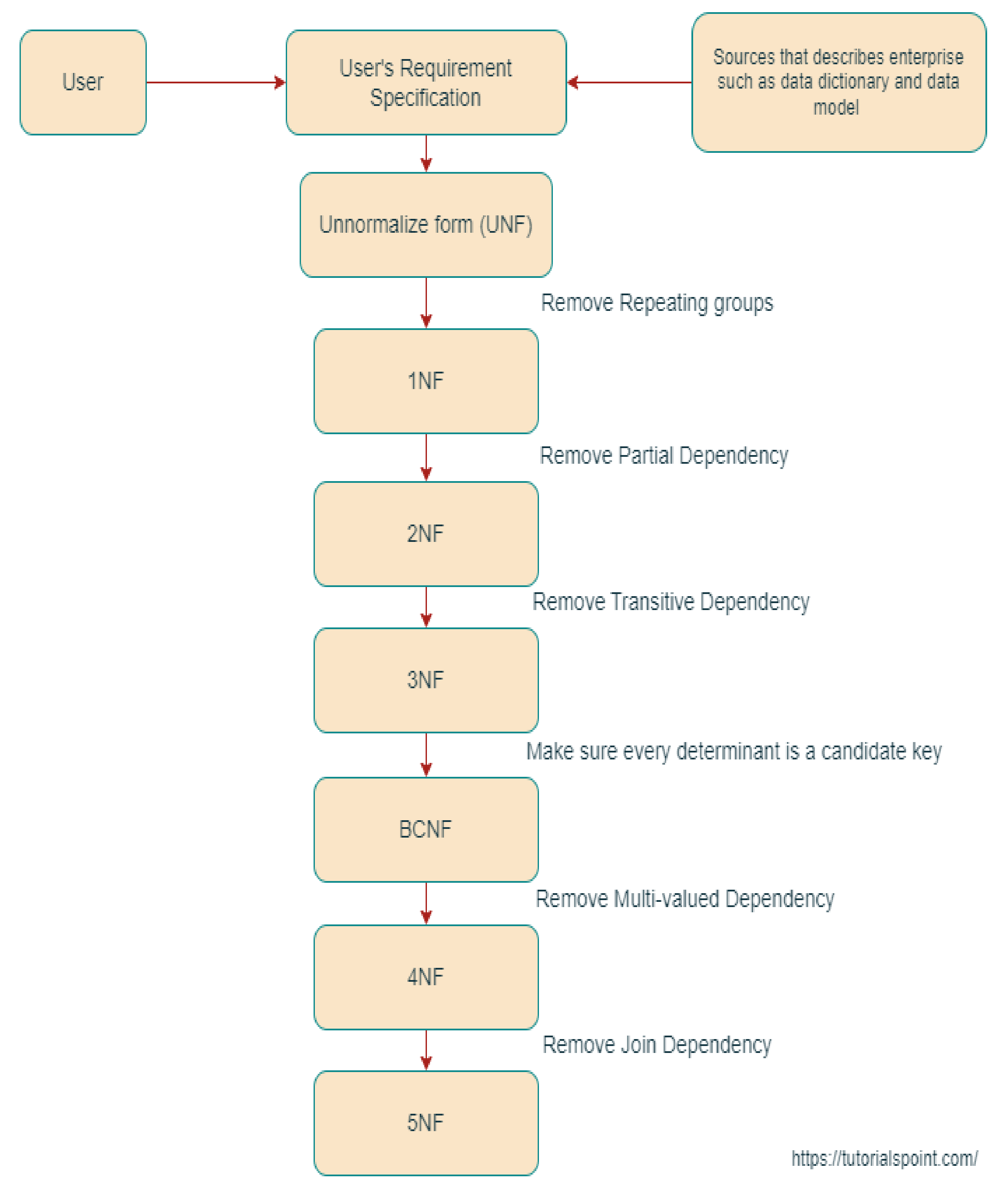
→Duplicate tuples must not be present in relation.

→ Each tuple must have exactly one data value for an attribute.

→Tuples and attributes of relation do not have to follow a significant order.

Normalization:

- Normalization is the process of minimizing redundancy from a relation or set of relations. Redundancy in relation may cause insertion, deletion and updation anomalies. It can be considered as a "filtering" or "purification" process to make the design have successively better quality.
- ***Normal Forms:*** The normal form of a relation refers to the highest normal form condition that it meets, and hence indicates the degree to which it has been normalized.
- ***De-Normalization:*** It is the process of adding redundant data to Speed up complex queries involving multiple table JOINS.



User

User's Requirement Specification

Sources that describes enterprise such as data dictionary and data model

Unnormalize form (UNF)

Remove Repeating groups

1NF

Remove Partial Dependency

2NF

Remove Transitive Dependency

3NF

Make sure every determinant is a candidate key

BCNF

Remove Multi-valued Dependency

4NF

Remove Join Dependency

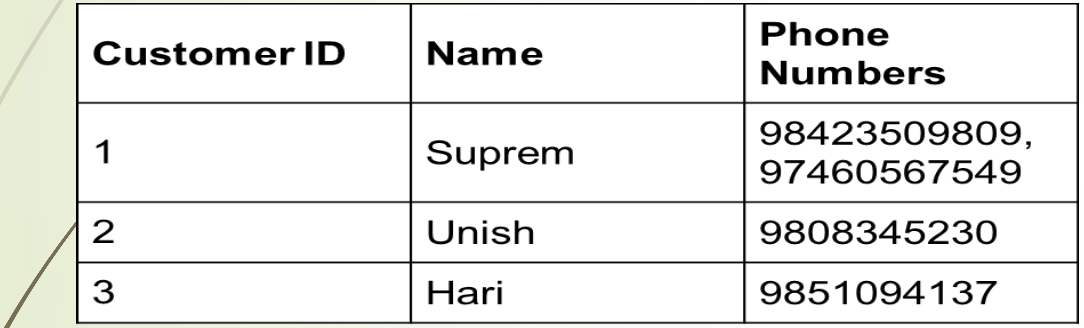
5NF

https://tutorialspoint.com/

First Normal Form (1NF)

- 1NF requires that each column in a table contains atomic values and that each row is uniquely identified. This means that a table cannot have repeating groups or arrays as columns, and each row must have a unique primary key.
- A table is in 1NF if each column contains atomic values and each row is uniquely identified.

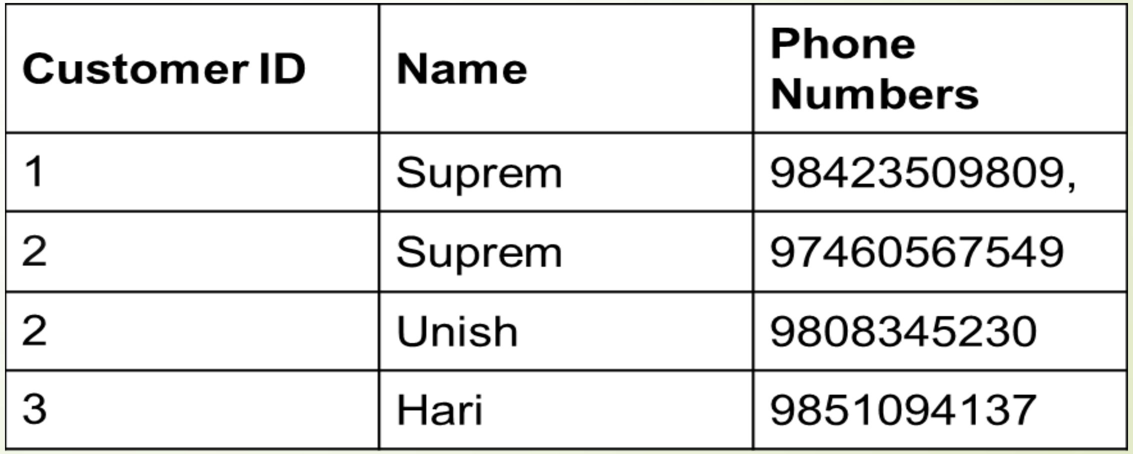
For example, a table that lists customers and their phone numbers



Customer ID	Name	Phone Numbers
1	Suprem	98423509809, 97460567549
2	Unish	9808345230
3	Hari	9851094137

This violates 1NF because the Phone Numbers column contains repeating groups.

To normalize this table to 1NF, we can split the Phone Numbers column into separate rows and add a separate primary key column ?



Customer ID	Name	Phone Numbers
1	Suprem	98423509809,
2	Suprem	97460567549
2	Unish	9808345230
3	Hari	9851094137

Second Normal Form (2NF)

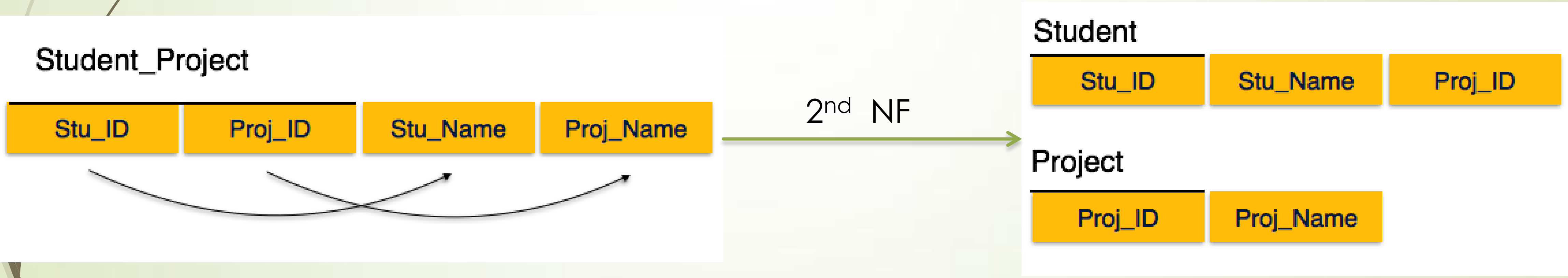
- 2NF builds on 1NF by requiring that each non-primary key column in a table is fully functionally dependent on the primary key. This means that a table should not have partial dependencies, where a non-primary key column depends on only part of the primary key.
- A table is in 2NF if each non-primary key column is fully functionally dependent on the primary key.

Partial dependency: if proper subset of candidate key determines non-prime attribute.

Non-prime attribute : Attributes which are not part of any Candidate key.

Candidate key: It is a key whose proper subset is not a key.

Prime-attribute: The attributes which are making candidate keys are prime attribute.



Student_Project

Stu_ID	Proj_ID	Stu_Name	Proj_Name

2nd NF

Student

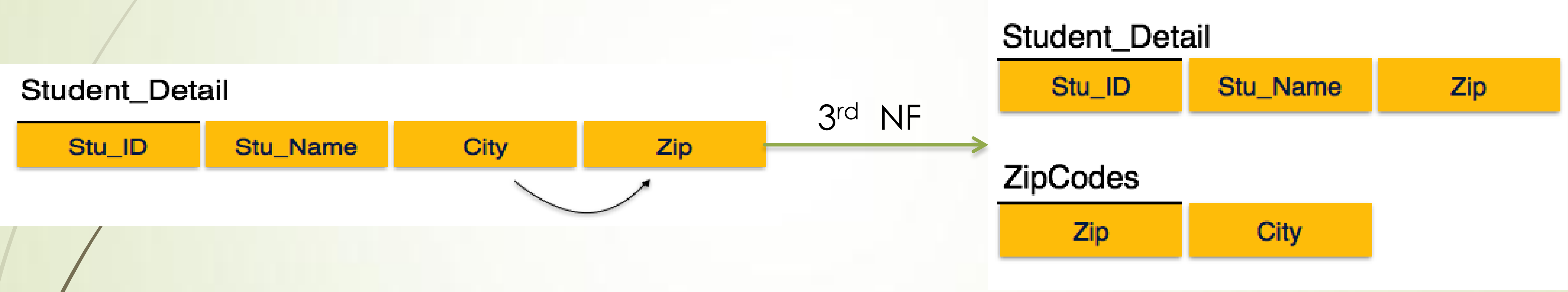
Stu_ID	Stu_Name	Proj_ID

Project

Proj_ID	Proj_Name

Third Normal Form

- For a relation to be in Third Normal Form, it must be in Second Normal form and the following must satisfy −
- No non-prime attribute is transitively dependent on prime key attribute.
- For any non-trivial functional dependency, X → A, then either X is a superkey or, A is prime attribute.



Student_Detail

Stu_ID	Stu_Name	City	Zip

Student_Detail

Stu_ID	Stu_Name	Zip

ZipCodes

Zip	City

Boyce-Codd Normal Form (BCNF)

- Boyce-Codd Normal Form or BCNF is an extension to the third normal form, and is also known as 3.5 Normal Form.
- For a table to satisfy the Boyce-Codd Normal Form, it should satisfy the following two conditions:
- It should be in the **Third Normal Form**.
- And, for any dependency A → B, A should be a **super key**.
- The second point sounds a bit tricky, right? In simple words, it means, that for a dependency A → B, A cannot be a **non-prime attribute**, if B is a **prime attribute**.

student_id	subject	professor
101	Java	P.Java
101	C++	P.Cpp
102	Java	P.Java2
103	C#	P.Chash
104	Java	P.Java

- One student can enrol for multiple subjects. For example, student with **student_id** 101, has opted for subjects - Java & C++
- For each subject, a professor is assigned to the student.
- And, there can be multiple professors teaching one subject like we have for Java.

student_id	p_id
101	1
101	2

p_id	professor	subject
1	P.Java	Java
2	P.Cpp	C++

student_id, subject form primary key, which means subject column is a **prime attribute**.

But, there is one more dependency, professor → subject.

And while subject is a prime attribute, professor is a **non-prime attribute**, which is not allowed by BCNF.

Transforming 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:

I. ***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.

II. ***Represent relationships:*** Each relationship in an E-R diagram must be represented on the relational database design. How we represent relationship depends on its nature.

III. ***Normalize the relations:*** The relations created in steps I & and II may have unnecessary redundancy. So we need to normalize these relations to make them well structured.

IV. ***Merge the relations:*** Across 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.

Merging Relations:

- Some of the relations may be redundant,we should merge those relations to remove the redundancy.

Example:

Suppose modeling a user interface or transforming an E-R diagram results in 3Nf relation.

Employee 1 (Emp-ID, Name, Address, Phone)

Modeling a second user interface might result in following relation:

Employee 2 (Emp_ID, Name, Address, Jobcode, Experience)

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

Employee (Emp-ID, Name, Address, Phone, Jobcode, Experience))

Physical file and Database design:

- Designing physical files and databases requires certain Information that should have been collected during prior SDIC phases. The information includes the following:
- Normalized relations, including volume estimates
- Definitions of each attribute.
- Descriptions of where and when data are used.
- Requirements for response and data integrity.
- Descriptions of the technologies used for implementing the files and database so that range of required decision and choices for each is known.

Designing Fields:

- Designing Fields in database management involves understanding the structure and characteristics of data elements.
- A field is the smallest unit of application data recognized by system software, such as a programming language or a database management system (DBMS). It represents an attribute from a logical database model and can be represented by several fields.
- A calculated field is derived from other database fields. Some database technologies allow explicit definition of calculated fields alongside raw data fields. The database technology will either store the calculated value or compute it when requested.

Designing Physical Tables:

- A physical table is a named set of rows and columns that specifies fields in each row of the table. The design of a physical table, has two goals, efficient use of secondary storage and data processing speed.
- The efficient use of secondary storage relates to how data are loaded on disks. The efficiency of secondary storage depends on factors such as operating system parameters, outside the control of each database.
- Data are most efficiently processed when they are stored close to one another on Secondary memory, thus, minimizing the no. of input/output (1/0) operations that must be performed.

Designing Forms and Reports:

- The forms are used to collect or present 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. Both forms and reports are the product of input and output design. The main difference is that forms provide fields for data input but reports are purely used for reading.

Process of designing forms and reports:

- Designing forms and reports is a user-focused activity that typically follows prototyping approach. First we must gain an understanding of the intended user and task objectives by collecting initial requirements. During this process, Several questions to "who, what, when, where and must be answered .

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

Graining and understanding of these questions a required first step in creation of any form or report. After collecting initial requirements, we structure and refine these information into an initial prototype. Now, we ask users to review and evaluate the prototype. After reviewing the prototype, users may accept the design or request some changes. If changes are needed we will refine according to user requirements until the design is accepted.

Deliverables and Outcomes:

- **Narrative overview:** It contains overview of characteristics of target users, tasks, system etc. in which form f or report will be used.
- **Sample design:** It provides sample design which may be hand drawn using a coding sheet although, in most instances it is developed using CASE tools.
- **Testing and usability assessment:** This section provides all testing and usability assessment information. Assessing usability depends on speed, accuracy, and satisfaction.

Formatting forms and reports:

1. **General Formatting Guidelines:** Proper formatting of forms and reports is essential. Following are the general guidelines for the design of forms and reports

- ✓ ***Meaningful titles:*** The form or report should contain title that is clear and specific. It should clearly describe the content and use of form or report.
- ✓ ***Meaningful information:*** Only the information that is relevant and needed by the user should be displayed on the form or report.
- ✓ ***Balanced Layout:*** The information should be balanced on the Screen or page page, ie, the display should not be too crowded and, not to spread out.
- ✓ ***Easy Navigation:*** It should be possible for the user to easily move forward and backward, through the contents of form or report.

2. **Guidelines for displaying contents:** the way the form or a report appears to the human eye has a lot of impact on the user so, we follow some guidelines for better display of contents:

- ✓ ***Highlighting information:*** Highlighting the information will enhance the appearance of the output. Highlighting of information can be carried out using different methods such as color difference, intensity difference, underlining, font and size differences etc.
- ✓ ***Using Color:*** Use of appropriate colors while designing has several advantages like strikes the eye, draws attention , use of colors in graphs and charts helps in better understanding etc.
- ✓ ***Displaying text:*** We should use appropriate punctuation wherever required. The text should be properly spaced and there should be blank line between paragraphs.
- ✓ ***Designing tables and lists:*** We should use meaningful labels to all columns and rows and separate labels from other information by using highlighting. Displayed data should be sorted in meaningful order.

Assessing Usability

- Usability is a measure of how well a specific user in a specific context can use a product/design to achieve a defined goal effectively, efficiently and satisfactorily. Designers usually measure a design's usability throughout the development process—from wireframes to the final deliverable—to ensure maximum usability.

Unability typically refers to the following three characteristics:

i. **Speed:** Can you complete a task efficiently?

ii. **Accuracy:** Does the system provide what you expect?

iii. **Satisfaction;** Do you like using the system?

General design Guidelines for usability of forms and reports:

Usability Factor	**Guideline for achievement of usability**
Consistency	Consistent use of terminology, formatting titles and navigation within and across outputs,
Organization	Text and data should be aligned and sorted for efficient navigation and entry.
Clarity	Outputs should be clear to the user and units of measure should be clearly indicated.
Format	Information format should be consistent between entry and display
Flexibility	Information should be viewed and retrieved in a manner most convenient to the user.

Designing Interfaces and Dialogues:

Interface design focuses on how information is provided to and captured from users. Dialogue design focuses on the Sequencing of interface displays. The design of interfaces and dialogues is the process of defining the manner in which humans and computers exchange information.

Measures of Usability:

Learnability: How difficult is it for a user to perform task for the first time?

Efficiency: How quickly can users perform tasks?

Error rate: How many errors might a user encounter , and how easy it is to recover from those errors?

Memorability: How easy is It to remember how to accomplish a task?

Satisfaction: How enjoyable is the system to use?

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.

Methods of Interacting

Command language interaction: This type of interaction requires users to remember command syntax and semantics. The user enters explicit statements to invoke operations within system.

Menu interaction: A menu is an simply a list of options. When option is selected by the user, a specific command is Invoked. Menus are the most widely used interfaces for Interaction.

Form Interaction:It allows users to fill in the blanks when working with a system. Form Interaction is important and effective for both the input and presentation.

Object-based interaction: Using icons is the most common method for implementing object-based Interaction. Icons are the graphic symbols. Icons take little space and can be quickly understood by users.

Natural language interaction: Natural language interaction is being applied with both keyboard and voice entry systems.

Designing Interfaces:

- User interface is the front-end application view to which user interacts in order to use the software. The Software becomes more popular if it's user interface is attractive, simple to use, responsive etc. It consists of following attributes:

i. ***Designing Layouts:*** For designing layouts standard formats similar to paper-based forms and reports should be used. Screen navigation on data entry Screens should be left-to-right, top-to-bottom as on paper forms. Flexibility and consistency are primary design goals.

ii. ***Structuring data entry:***

- ✓ Never require data that are already on-line or that can be computed.
- ✓ Always provide default values when appropriate
- ✓ Make clear the type of data units requested for entry.
- ✓ Use character replacement when appropriate.

iii. ***Controlling data input:***

- ✓ One objective of interface design is to reduce data entry errors,
- ✓ Role of systems analyst is to anticipate user errors and design features into the system's interfaces to avoid, detect and Correct data entry mistakes.

iv. ***Providing Feedback:***

- ✓ Interface design should provide status information as feedback like keeping users information of what is going on the system.
- ✓ It should provide error and warning messages.

v. ***Providing Help:***

- ✓ We should try to place ourselves in user's place when designing help.
- ✓ Help messages should be short and to the point.

Designing Dialogues:

The process of designing the overall sequences that users follow to interact with an Information system is called dialogue design. The dialogue design process consists of three major steps:

→Designing the dialogue sequence

→ Building a prototype

→Assessing usability.

In designing the dialogue sequence, we must first gain the understanding of how users might interact with the system. Then we build the prototype of the system meeting user's requirements. For a dialogue, to have high usability, It must be consistent in form, function and style.

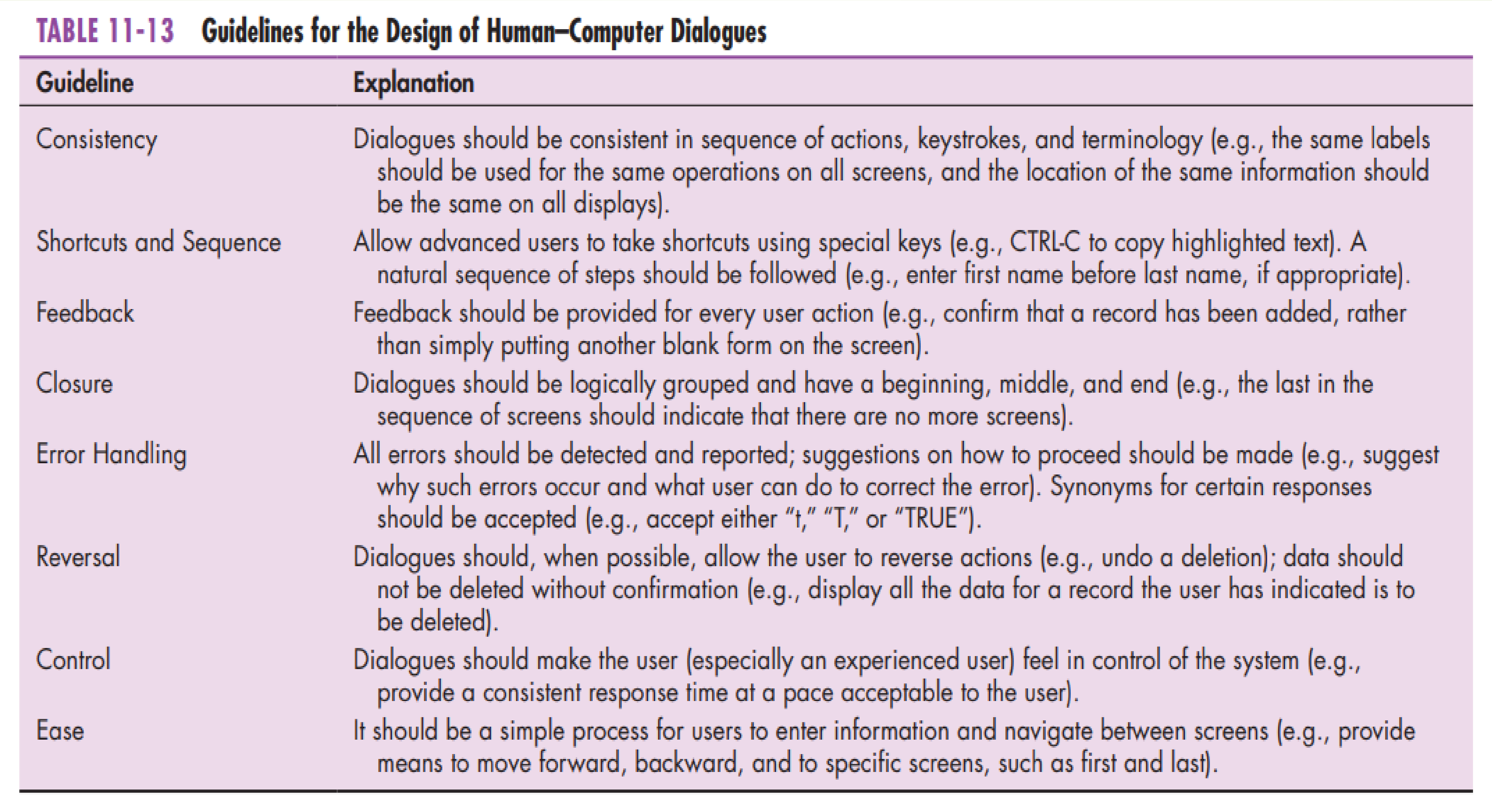


TABLE 11-13 Guidelines for the Design of Human–Computer Dialogues

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).
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).
Ease	It should be a simple process for users to enter information and navigate between screens (e.g., provide means to move forward, backward, and to specific screens, such as first and last).

Designing Interfaces and Dialogues in Graphical Environments:

Graphical user interface (GUI) environments have become the de facto for human computer interaction. When designing graphical interfaces we encounter different issues, so numerous factors must be considered. Some factors are common to all GUI environments, whereas others are Specific to the single environment. In most discussions of GUI programming two rules repeatedly emerge as composing the first step to become an effective GUI designer:

→Become an expert of the GUI environment.

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