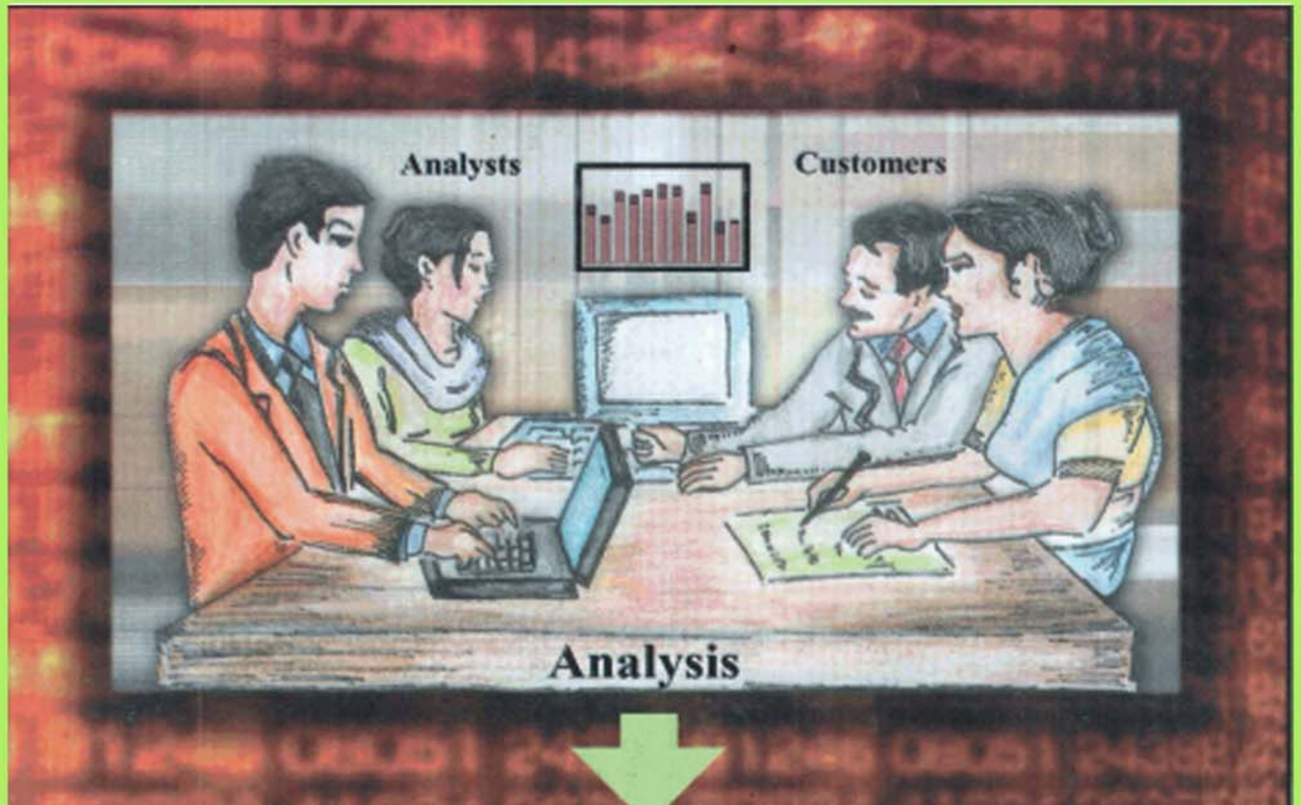
Indira Gandhi National Open University
School of Computer and Information Sciences

MCS-014
SYSTEM ANALYSIS
AND DESIGN



MORE DESIGN ISSUES AND CASE TOOLS

3

Indira Gandhi
National Open University
School of Computer and
Information Sciences

MCS-014
SYSTEMS ANALYSIS AND DESIGN

Block

3

MORE DESIGN ISSUES AND CASE TOOLS

Acknowledgements

To all the faculty of SOCIS, IGNOU for their comments on the course material;
to Shri Vikas Kumar for help in finalizing the CRC.

June, 2004

©Indira Gandhi National Open University, 2004

ISBN-81-266-1300-9

BLOCK INTRODUCTION

This block is on the issues related to design and CASE tools.

If an organization whose processes are not computerized is considered then it is often found that the staff at managerial level request for reports or summary of information rather than going through bulky files. Staff who is in charge of making report studies the files , understands them thoroughly and makes a summary of information in them. But, this process is tedious and most of the software now comes with report generators. To avoid the problem of manual preparation of reports, when the processes of the company are being computerized, a provision for reports generation will be made in the software that is being developed for the company. The same is the case with the user interface of the software. Often, forms are needed through which data is accepted by the package for further processing. So, design of forms must be user friendly and design of reports should be friendly to the managers. Both design of forms and reports are covered in this block.

During the system study, Software Engineers often come across registers which consist of fields (Of course, it is not uncommon to find redundant fields in them). The logical design of the database to be developed will not be mapped from the fields in the registers. Rather, the fields are analyzed, various rules called Normal forms are applied so that the logical design arrived at does not consist of redundancy and integrity is not violated. After the logical design is made, Physical data base has to be done which is covered in this block.

Now a days, it is common to find CASE tools for every phase of SDLC. CASE tools make the life of analysts, programmers etc. easier. But, it is very important to chose the right tool for the respective task. CASE tools that are used for systems development are covered in this block.

This block consists of 3 units and is organized as follows:

Unit-8 deals with the design of forms and report. In this unit, importance of forms, reports and differences between them are discussed. Also, criteria for designing forms and reports along with deliverables and outcomes are covered in this unit.

Unit-9 deals with the design of physical files and databases. In this unit, the process of designing database fields, physical records and files is also discussed. This unit also includes a case study.

Unit-10 focuses on CASE tools for development of systems. This unit covers the role of CASE tools along with the advantages and disadvantages of their use. Various components of CASE are discussed. This unit also discusses the issues related with visual CASE tools.

UNIT 8 FORMS AND REPORTS DESIGN

Structure **Page Nos.**

8.0 INTRODUCTION

This unit deals with the interface of software with users. Usually, the interface is through forms and reports that are associated with the system. In this unit, we will study different aspects of designing Forms and Reports, as these are the key ingredients for successful systems. As the quality of a system greatly depends upon the quality of inputs and outputs, the process of designing forms and reports is very important.

The logical phase within the system development life cycle (SDLC) deals with the issues related to the design of system inputs and outputs (forms and reports) as shown in figure 8.1. Forms are used to collect data for the system and reports to deliver information to users. With forms, data can be entered into the database. With data entered in the database, it is possible to use a query language so as to generate reports about the data. In this unit, we shall also look into the deliverables produced during the process of designing forms and reports. Formatting of forms and reports is also discussed as this serves as the building block for designing.

Forms and reports should be well conceived and attractive in design. In order to achieve this goal, we shall look into different criteria that are to be followed while designing forms and reports.

8.1 OBJECTIVES

After going through this unit, you should be able to:

- define Forms & Reports and their importance in real life;
- list the process of designing Forms & Reports ;
- know about Internal information, External information, Turnaround documents and differentiate between them;
- apply the general guidelines for formatting Forms and Reports; and
- specify different criteria for designing Forms and Reports.

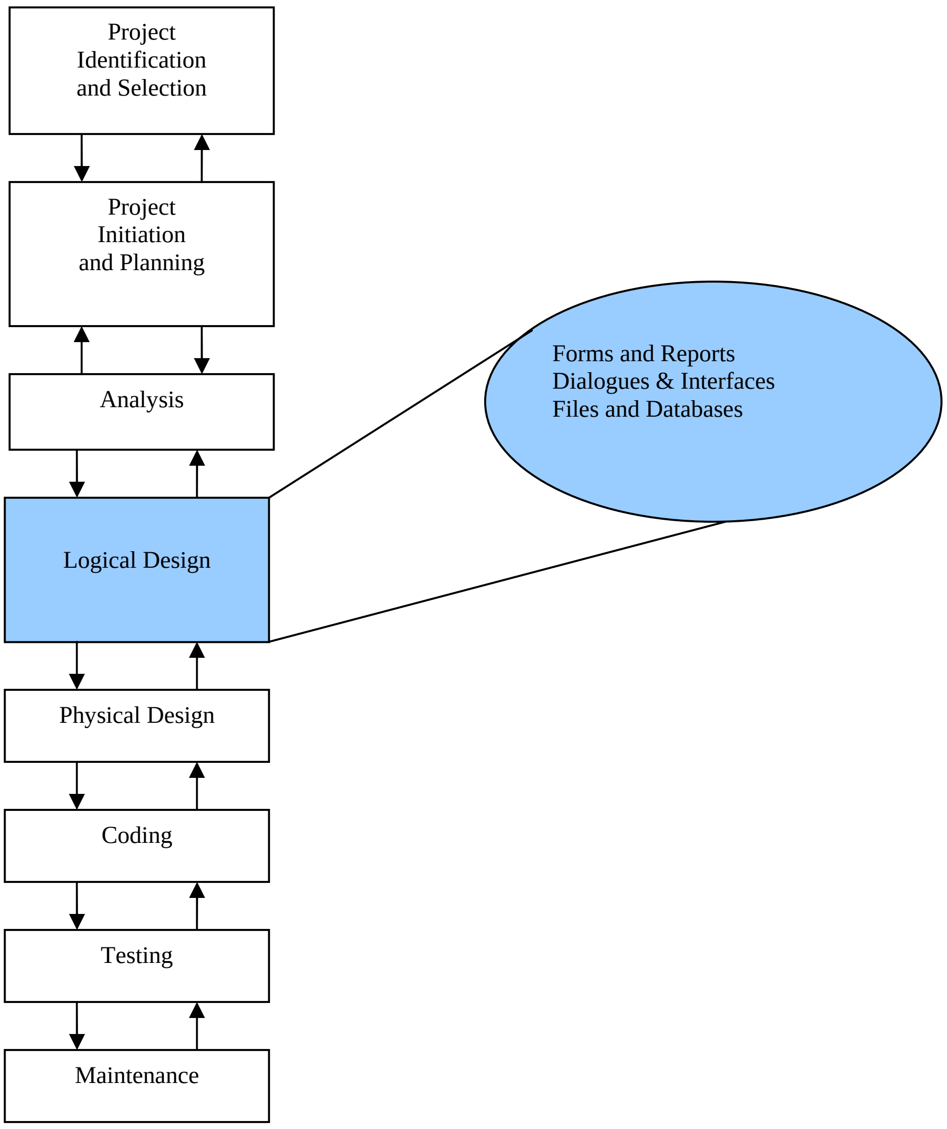


Figure 8.1: Systems Development Life Cycle with Logical Design Phase Highlighted

8.2 FORMS

Like a form on paper that is used to fill out information with a pen or pencil, a Form in computer terminology identifies the data we want to collect. It also allows us to enter data into the database, display it for review and also print it for distribution. However, an electronic form has several important advantages over standard paper forms. These have the advantage of using a computer database and are more versatile and powerful than paper forms.

Examples of forms are Business forms, Electronic spread sheet, ATM transaction layout, etc.

Figure 8.2 shows a simple form that is used to collect employee details.

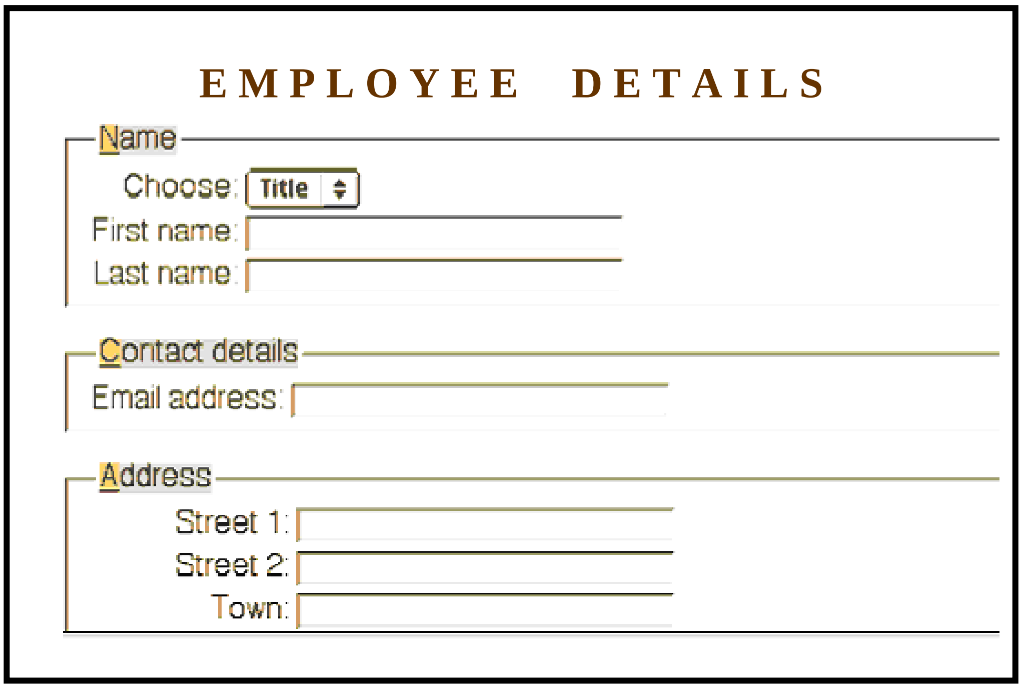


Figure 8.2: A Simple Form

8.2.1 Importance of Forms

The following are various advantages of Forms:

- A form provides an easy way to view data.
- Using forms, data can be entered easily. This saves time and prevents typographical errors.
- Forms present data in an attractive format with special fonts and other graphical effects such as colour and shading.
- Forms offer the most convenient layout for entering, changing and viewing records present in the database.
- An entry field in a form can present a list of valid values from which users can pick to fill out the field easily.

8.3 REPORTS

Analysing and presenting data are just as important as entering and sorting these out. Computer systems use reporting and query applications to retrieve the data that are available in the database and present it in a way that provides useful information, drives decision-making and supports business projects. A report presents data as meaningful information, which can be used and distributed.

A report is the information that is organized and formatted to fit the required specification. It is a passive document that contains only predefined data and is used solely for viewing and reading. Reports can be printed on paper, or these may be transferred to a computer file, a visual display screen, etc. Reports are the most visible component of a working information system and hence they often form the basis for the users and management's final assessment of the systems value.

Examples of reports are: invoices, weekly sales summaries, mailing labels, pie chart, etc.

Figure 8.3 shows a simple report that displays the residence telephone numbers of all the employees in the organization.

EMPLOYEE RESIDENCE PHONE LIST

S.No	LAST NAME	FIRST NAME	DESIGNATION	PHONE NUMBER
1	Verma	Ajay	Regional Manager	6522081
2	Gupta	Vinay	Branch Manager	6478017
3	Michael	Nancy	H.R Manager	6152430
4	Singh	Amar	Sales Executive	5769081

Figure 8.3: A Simple Report

8.3.1 Importance of Reports

The following are various advantages of Reports:

- We can organize and present data in groups.
- We can calculate running totals, group totals, grand totals, percentage of totals, etc.
- Within the body of Reports, we can include sub-forms, sub-reports and graphs.
- We can present data in an attractive format with pictures, special fonts and lines.
- We can create a design for a report and save it so that we can use it over and over again.

8.4 DIFFERENCES BETWEEN FORMS AND REPORTS

The following are some differences between Forms and Reports:

- Forms can be used for both input and output. Reports, on the other hand, are used for output, i.e., to convey information on a collection of items.
- Typically, forms contain data from only one record, or are at least based on one record such as data about one student, one customer, etc. A report, on the other hand is only for reading and viewing. So, it often contains data about multiple unrelated records in a computer file or database.
- Although we can also print forms and datasheets, reports give more control over how data are displayed and show greater flexibility in presenting summary information.

8.5 PROCESS OF DESIGNING FORMS AND REPORTS

Good quality business processes deliver the right information to the right people in the right format and at the right time. The design of forms and reports concentrates on this goal.

Designing of forms and reports is a user-focused activity that typically follows a prototyping approach. Before designing a form or a report, we should have a clear idea so as to what is the aim of the form or report and what information is to be collected from the user.

There are some useful questions related to the creation of all forms and reports, such as "who, what, when, where and how" which must be answered in order to design effective forms and reports.

WHO	Understanding who the actual users are, their skills and abilities, their education level, business background, etc., will greatly enhance the ability to create effective design.
WHAT	We need to have a clear understanding of what is the purpose of the form or report and what task will the users be performing and what information is required so as to successfully complete the given task.
WHEN	Knowing when exactly the form or report is needed and used will help to set up time limits so that the form or report can be made available to the users within that time frame.
WHERE	Where will the users be using this form or report (i.e., will the users have access to on-line systems or will they be using them in the field)?
HOW	How many people will be using this form or report, i.e., if the form or report is to be used by a single person, then it will be simple in design but if a large number of people are going to use it, then the design will have to go through a more extensive requirements collection and usability assessment process.

After having answered all the above questions, we would have collected all the initial requirements. The next step is to refine this information into an initial prototype. Structuring and refining the requirements are completed without interacting with the end users, although we may need to occasionally contact users in order to clarify some issues that might have been overlooked during analysis.

Once the initial prototype is ready, we should ask the users to review and evaluate the prototype. After the review, the design may be accepted by the users. Or at times the users may ask for certain changes to be made. In case changes are to be made then the construction-evaluation-refinement cycle will have to be repeated until the design is accepted.

The next step in the Design process is to Design, Validate and Test the outputs using some combination of the following tools:

a)	Layout tools	(Ex.: Hand sketches, printer/display layout charts or CASE)
b)	Prototyping tools	(Ex.: Spreadsheet, PC, DBMS, 4GL)
c)	Code generating tools	(Ex.: Report writer)

The initial prototype may be constructed in numerous environments. For example, a CASE tool or the standard development tools that are used within the organization be used. Usually, initial prototype are mock screens that can be produced using word processor, computer graphics design package, or electronic spreadsheet. Mock screens are not the working modules or systems.

Tools for designing forms and reports are rapidly evolving and now a days Online graphical tools for designing forms and reports are very much in use in most professional development organizations.

Check Your Progress 1

1. Define Form and Report. Also, differentiate between them.
   ……………………………………………………………………………………
   ……………………………………………………………………………………
   ……………………………………………………………………………………
2. List the advantages of using Forms and Reports.
   ……………………………………………………………………………………
   ……………………………………………………………………………………
   ……………………………………………………………………………………
3. Describe the process of designing Forms and Reports.
   ……………………………………………………………………………………
   ……………………………………………………………………………………
   ……………………………………………………………………………………
4. State True or False for the following:

   a) A form is a type of layout that typically lists and summarizes data from several unrelated data base records.
   b) A report is a type of output that typically lists data associated with one data base record.
   c) Within reports, we can include sub-forms and sub-reports.

8.6 DELIVERABLES AND OUTCOMES

Each phase in the System Development Life Cycle (SDLC) helps in the construction of the system. As we move from one phase to another, each phase produces some deliverables (measurable result or output of a process) that will be used in the later phases or activity. While designing forms and reports, design specifications are the major deliverables and these serve as inputs to the system implementation phase.

8.7 DESIGN SPECIFICATIONS

Design specifications which are major deliverables while designing forms and reports have the following three sections:

- Narrative overview
- Sample design
- Testing and usability assessment.

8.7.1 Narrative Overview

This contains the general overview of the characteristics of actual users of the form or report, task, the system that will be used and the environment factors in which the form or report will be used. The main purpose of Narrative overview is to provide information in detail to the people who will develop the final form or report, regarding the main aim of the form, who the actual users of the form will be, and how it will be used, so that they can make appropriate implementation decisions.

8.7.2 Sample Design

The sample design of the form may be hand-drawn using a coding sheet or it may be developed using CASE or standard development tools. If the sample design is done using actual development tools, then the form can be thoroughly tested and assessed.

8.7.3 Testing and Usability Assessment

This section provides information required for testing and usability assessment. While testing, it is important to use realistic or reasonable data and demonstrate all controls.

8.8 TYPES OF INFORMATION

The main purpose of any form or report is to convey certain information to the user. Information can be classified according to their distribution inside or outside the organization and the people who read and use them as follows:

- Internal information,
- External information, and
- Turnaround information.

8.8.1 Internal Information

Internal information is the information that is collected, generated or consumed within an organization. That is, this information is intended only for the internal system owners and system users within an organization.

Internal information either supports day to day business operations or management monitoring and decision making, e.g., Detailed summary or exception information printed on hard copy reports for internal business use. Internal information can also consists of simple informational reports summarizing daily activities within the organization.

The following are three sub classes of internal information:

- **Detailed reports:** These reports present information with little or no filtering or restrictions.

  Ex.: Detailed listing of all customer accounts, orders or products in inventory. The example in figure 8.4 shows a listing of all purchase orders that were generated on a particular date. The P.O. No. Indicates the product order number.

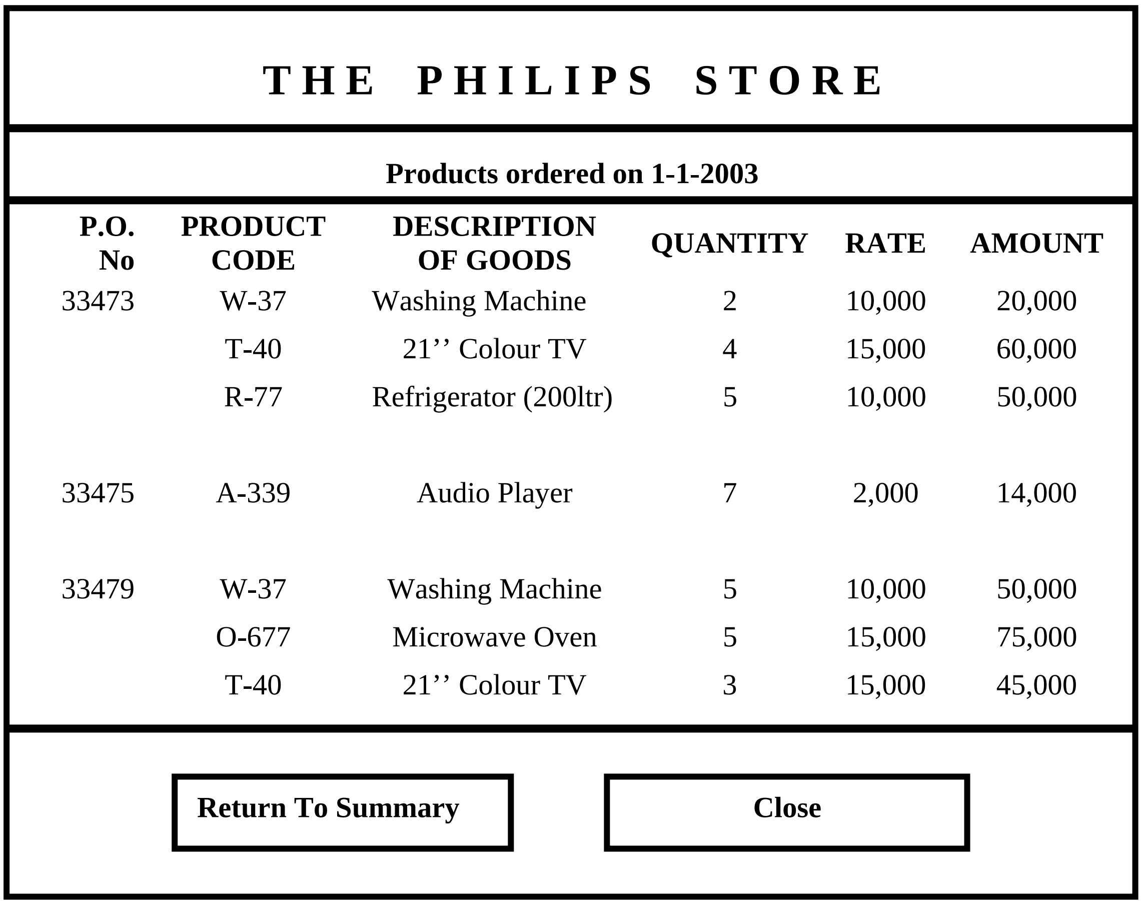
- **Summary reports:** These reports categorize information for managers who do not want to wade through details. The data in summary reports are categorized and summarized to indicate trends and potential problems. We can also include charts and graphs in the summary report so that it clearly summarizes trends at a glance.

  Ex.: Report that summarizes the months and years total sales by product types and category.

  The example in figure 8.5 summarizes the Sales details for a given month by product type and category.

- **Exception reports:** These reports filter data before they are presented to the manager as information, i.e., these reports include exceptions to some conditions or standards.

Ex.: A report that identifies items, which are low in stock.
The example in figure 8.6 depicts the identification of dealers who have to pay the dues.



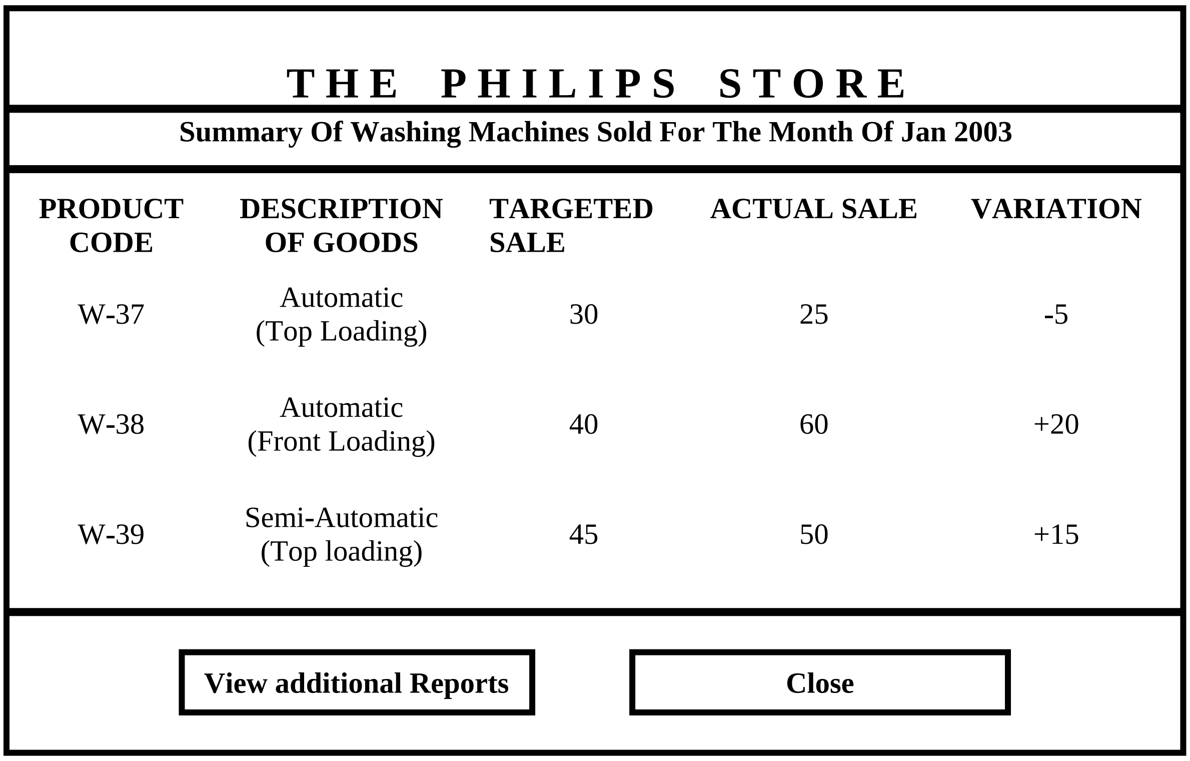
THE PHILIPS STORE

Products ordered on 1-1-2003

P.O. No	PRODUCT CODE	DESCRIPTION OF GOODS	QUANTITY	RATE	AMOUNT
33473	W-37	Washing Machine	2	10,000	20,000
	T-40	21’’ Colour TV	4	15,000	60,000
	R-77	Refrigerator (200ltr)	5	10,000	50,000
33475	A-339	Audio Player	7	2,000	14,000
33479	W-37	Washing Machine	5	10,000	50,000
	O-677	Microwave Oven	5	15,000	75,000
	T-40	21’’ Colour TV	3	15,000	45,000

Return To Summary
Close

Figure 8.4: A Detailed Report



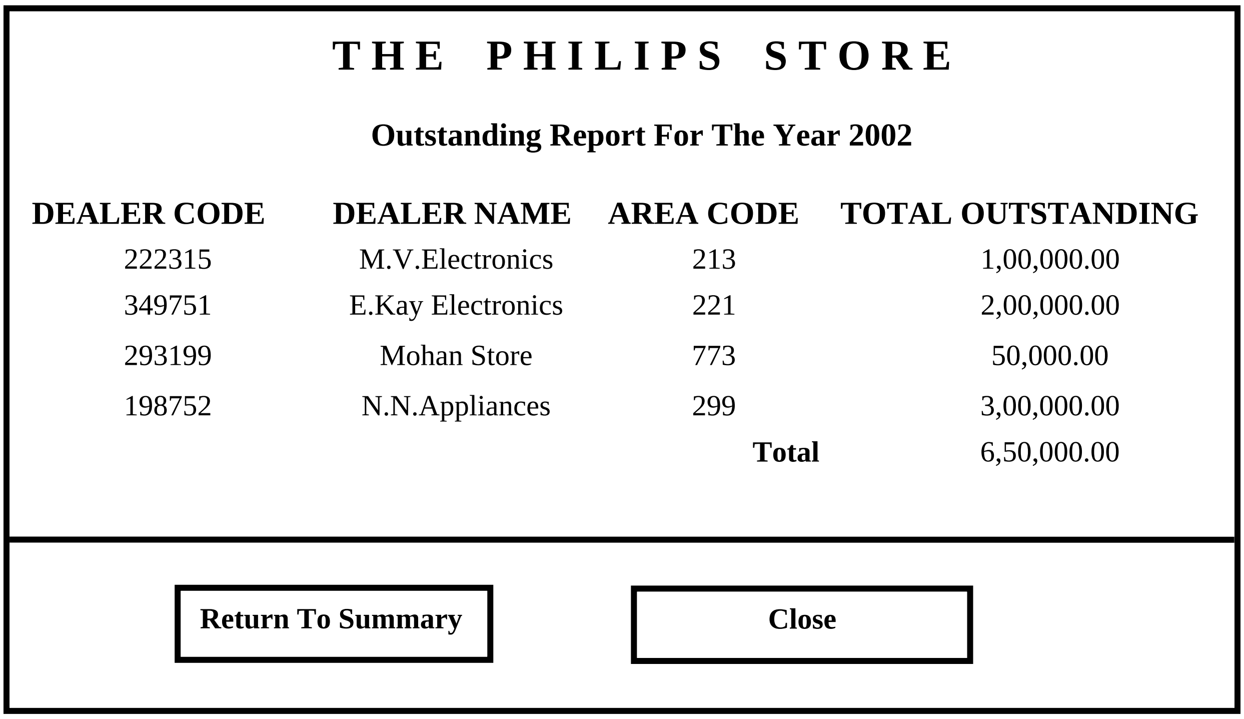
THE PHILIPS STORE

Summary Of Washing Machines Sold For The Month Of Jan 2003

PRODUCT CODE	DESCRIPTION OF GOODS	TARGETED SALE	ACTUAL SALE	VARIATION
W-37	Automatic (Top Loading)	30	25	-5
W-38	Automatic (Front Loading)	40	60	+20
W-39	Semi-Automatic (Top loading)	45	50	+15

View additional Reports
Close

Figure 8.5: A Summary Report



T H E   P H I L I P S   S T O R E

Outstanding Report For The Year 2002

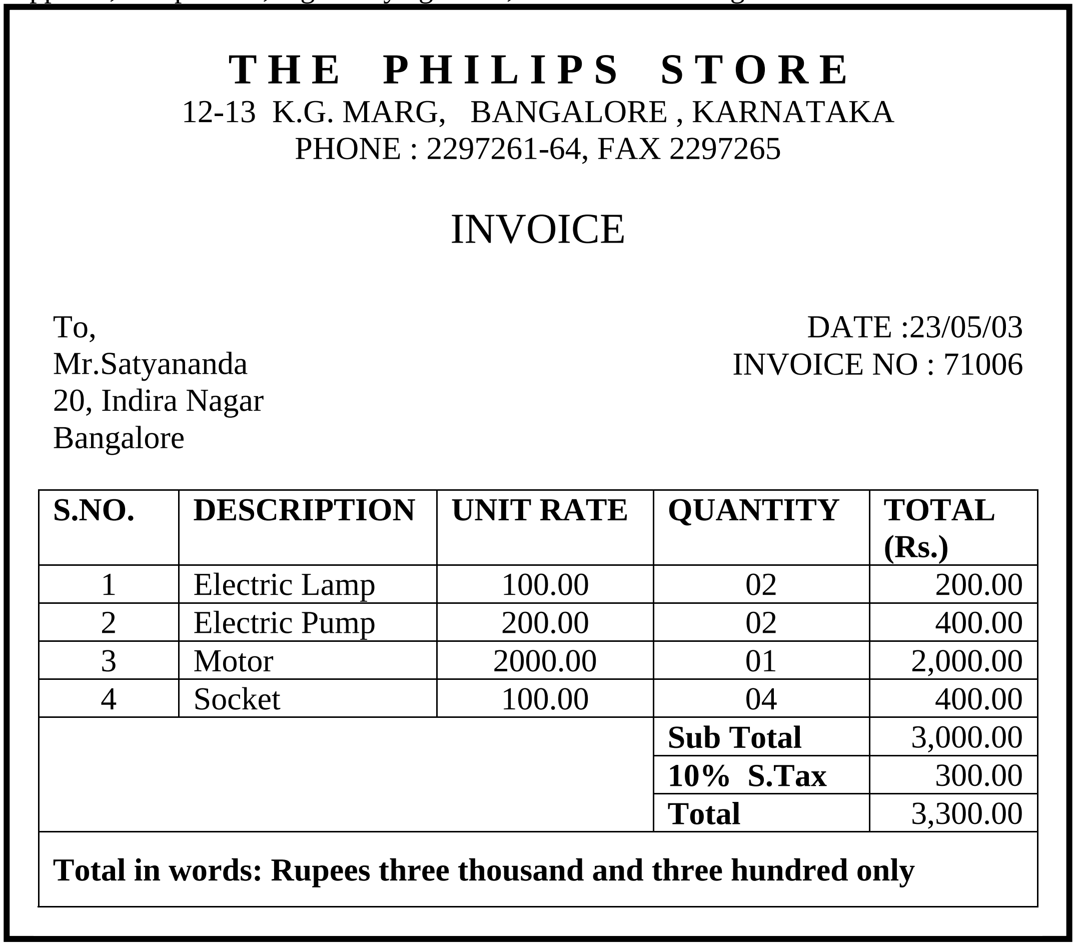
DEALER CODE	DEALER NAME	AREA CODE	TOTAL OUTSTANDING
222315	M.V.Electronics	213	1,00,000.00
349751	E.Kay Electronics	221	2,00,000.00
293199	Mohan Store	773	50,000.00
198752	N.N.Appliances	299	3,00,000.00
		Total	6,50,000.00

Return To Summary   **Close**

Figure 8.6: Exception Report

8.8.2 External Information

External information refers to the information collected from or created for customers, suppliers, competitors, regulatory agencies, etc. outside the organization.



T H E   P H I L I P S   S T O R E
12-13 K.G. MARG, BANGALORE , KARNATAKA
PHONE : 2297261-64, FAX 2297265

INVOICE

To,
Mr.Satyananda
20, Indira Nagar
Bangalore

DATE :23/05/03
INVOICE NO : 71006

S.NO.	DESCRIPTION	UNIT RATE	QUANTITY	TOTAL (Rs.)
1	Electric Lamp	100.00	02	200.00
2	Electric Pump	200.00	02	400.00
3	Motor	2000.00	01	2,000.00
4	Socket	100.00	04	400.00
			Sub Total	3,000.00
			10% S.Tax	300.00
			Total	3,300.00

Total in words: Rupees three thousand and three hundred only

Figure 8.7: An example of External Information

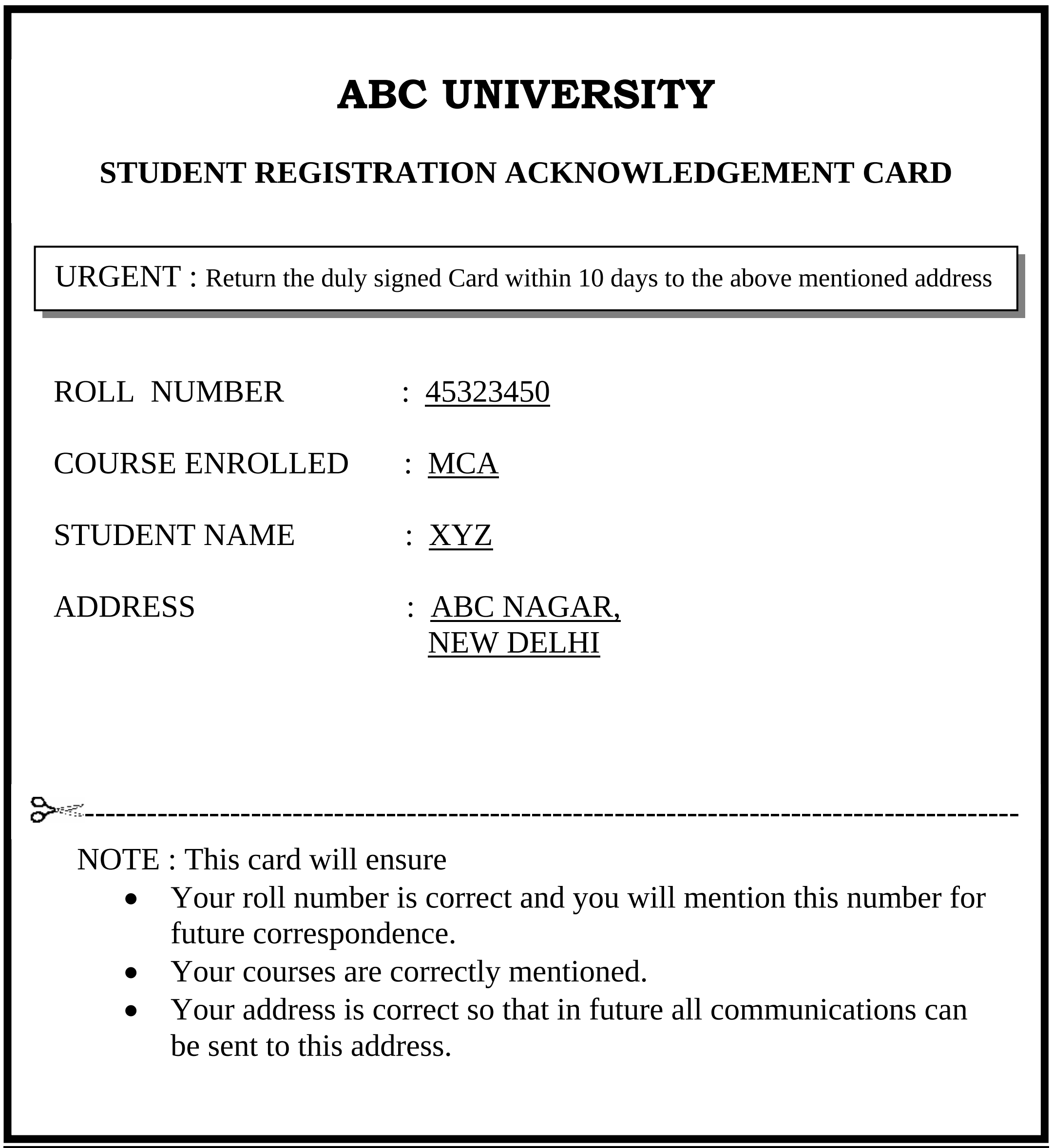
Examples of external information are: Invoices account statements, Product documentation, Purchase orders, Mailing labels, etc.

Figure 8.7 shows an invoice of a store that sells products manufactured by the Philips Company.

8.8.3 Turnaround Documents

There will be several instances where the information output is again used as input to obtain new information. The document that consists of such information is called Turnaround document. These begin as external information delivered to an external customer as an output, but ultimately return (in part or in whole) as internal information to provide new information as an input to an information system. For example, warranty card or acknowledge slip. This form has pre-printed system generated information that a customer must verify and return to the organization. The customer also adds new information (Ex.: Name, Address, etc.), which serves as input to the customer tracking system.

Figure 8.8 shows a simple acknowledgement card which is sent by a university to the student and the student in turn checks the information and sends a part of the document back to the university.



ABC UNIVERSITY

STUDENT REGISTRATION ACKNOWLEDGEMENT CARD

URGENT : Return the duly signed Card within 10 days to the above mentioned address

ROLL NUMBER	: 45323450
COURSE ENROLLED	: MCA
STUDENT NAME	: XYZ
ADDRESS	: ABC NAGAR, NEW DELHI

✂---

NOTE : This card will ensure

- Your roll number is correct and you will mention this number for future correspondence.
- Your courses are correctly mentioned.
- Your address is correct so that in future all communications can be sent to this address.

Figure 8.8: Turnaround Document

8.9 GENERAL FORMATTING GUIDELINES

Proper formatting of forms and reports is very much essential. But, unfortunately, a definitive set of rules for delivering every type of information to users is yet to be defined and these rules are continuously evolving along with the rapid changes in technology. However, certain guidelines are available which are to be considered while formatting information. One of the most important thing to keep in mind while designing usable forms and reports is that there should be active interaction with the users. If the appearance of forms or reports is awkward to use or confusing, the users will be dissatisfied even if the rest of the system performs well. Formatting of forms and reports influences individual task performance and perceptions of usability.

The following are general guidelines for the design of Forms and Reports, which make the Form, or Report more acceptable:

- Meaningful titles,
- Meaningful information,
- Balanced layout, and
- Easy navigation.

8.9.1 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. It should also include the date on which the form or report was generated. Page heading formats should be consistent throughout the system. For example, the date should always appear in the same place. Column headings should clearly indicate the contents of the columns and should be separated from the body using extra blank lines, horizontal rule etc.

8.9.2 Meaningful Information

Only the information that is relevant and needed by the user should be displayed on the form or report. Information should be provided in such a manner that the user could use it without any modification. All the information irrelevant to the intent of the form or report should not be displayed.

8.9.3 Balanced Layout

The information should be balanced on the screen or page, i.e., the display should not be too crowded and not too spread out. When deciding where to put individual fields on the form or report, we should see that the form or report is easy to understand and to use. The most important information should be placed where they are easiest to find (generally, at the beginning). The different fields should be separated by means of extra spaces whenever possible so that a subsequent field expansion will not necessarily force to redesign the entire layout. All related information should be grouped together wherever possible. For example, name, street address and city/state/pin code can be grouped together. Appropriate line spacing greatly enhances the readability of a form or report. We should insert extra blank lines to indicate where headers end and the body of information begins, where one multi-line detail ends and the next begins, where one group of items end and another begins, etc.

8.9.4 Easy Navigation

It should be possible for the user to easily move forward and backward through the contents of form or report. At any instance, it should be possible for the user to know where exactly s/he is (e.g., Page 2 of 3). The user should be notified when s/he is on the last page of a multi-page sequence. The user must be able to exit or quit the report or form easily.

Check Your Progress 2

1. What do you mean by internal information, external information and turnaround document?

   ……………………………………………………………………………………

   ……………………………………………………………………………………

   ……………………………………………………………………………………

2. List the general formatting guidelines.

   ……………………………………………………………………………………

   ……………………………………………………………………………………

   ……………………………………………………………………………………

3. What are the different sections of design specifications?

   ……………………………………………………………………………………

   ……………………………………………………………………………………

   ……………………………………………………………………………………

4. State True or False for the following:

   a) Balance is a way to draw attention to or away from data in output.
   b) Turnaround documents are produced for people outside the organization, but ultimately return as input to the internal system.
   c) Narrative overview contains the general guidelines for formatting forms and reports.

8.10 GUIDELINES FOR DISPLAYING CONTENTS

The way the form or a report appears to the human eye has a lot of impact on the user and by following specific guidelines for highlighting information such as using colour to display text, and presenting numeric tables and lists, we can make the form or report more presentable.

8.10.1 Highlighting Information

Highlighting the information will enhance the appearance of the output. However, highlighting should be used sparingly to draw the user to or away from certain information and to group together related information. Highlighting of information can be carried out using different methods such as using blinking and audible tones, colour difference, intensity difference, font and size differences, underlining, etc. Highlighting methods should be selected and consistently used based upon the level of importance of the emphasized information.

Highlighting will be very useful in situations such as the following:

- Notifying users of errors in data entry or processing;
- Drawing attention to high priority messages; and
- Providing warnings to users in situations like unusual data values or an unavailable device.

8.10.2 Using Colour

Colour influences the usability of the system. Use of appropriate colours while designing has several advantages which are given below:

- Soothes or strikes the eye;
- Emphasizes the logical organization of information;
- Draws attention to warnings;

- Evokes more emotional reactions; and
- Use of colours in graphs and charts helps in better understanding.

The following are some disadvantages of using colours:

- Resolution may degrade with different displays;
- Colour fidelity may degrade on different displays; and
- Printing or conversion to other media may not be possible easily.

8.10.3 Displaying Text

Now a days, as the text based applications such as e-mail, bulletin boards , chatting, etc., are being widely used, textual output is becoming increasingly important. Some of the guidelines to be followed in order to display text are given below:

We should use appropriate punctuation wherever required. The text should be properly spaced and there should be blank line between paragraphs. We should not hyphenate words between lines or use obscure abbreviations and acronyms while displaying textual information.

8.10.4 Designing Tables and Lists

The content and meaning of tables and lists are significantly derived from the format of the information. There are a few simple guidelines that are to be followed while designing tables and lists. Use meaningful labels to all columns and rows and separate labels from other information by using highlighting. The data displayed should be sorted in a meaningful order (like ascending or descending or alphabetic). Columns should be separated by at least two spaces between them. We should make use of single typeface except for emphasis.

Numeric, textual and alphanumeric data should be properly formatted. For example, numeric data should be right justified and columns should be aligned using decimal points or other delimiter.

Textual data should be left justified and line length should be somewhere between 30-40 characters per line. If there are long sequences of alphanumeric data, then they should be broken into small groups of three to four characters each.

8.11 CRITERIA FOR FORM DESIGN

Forms should be well conceived and attractive in design. We can achieve this goal if we design a form that satisfies the following criteria:

- Organization,
- Consistency,
- Completeness,
- Flexible entry, and
- Economy.

8.11.1 Organization

The different parts of a form must be arranged in a proper order with visual separation between the parts. Balancing of different information on the form should be done according to the sequence of entry, frequency of use, function and significance of that particular data. The first data available, the most important data and the data that is going to be used most frequently should always be placed in the beginning of the form. If there are groups of data of the like information, they should be placed together just as Name+Address+Phone Number. Grouping of information will help the user to understand which section of the form they are completing.

8.11.2 Consistency

Forms designed should be internally consistent. They must also be consistent with related forms and with other forms in the organization. If the forms are consistent, then it will be easy for the users to learn how to fill them. Consistent forms reduce errors and data capture costs.

8.11.3 Completeness

The form should gather all the necessary data at the source so that there is no need to transcribe data to other forms. This reduces the major source of errors.

8.11.4 Flexible Entry

It should be possible to enter data by hand or with a typewriter. In most cases, both kinds of entries occur.

8.11.5 Economy

The total cost of design, printing, data entry, etc., must be minimized. Most of the times, it is required to increase one cost to reduce another. Usually, handling costs are much more than the cost of designing and printing. Having spent more resources on design and printing often reduces the cost of data capture and keying.

8.12 CRITERIA FOR REPORT DESIGN

Reports convey information from one of more computer files to the user. They perform this task satisfactorily only when they present information to the user accurately and in small portions.

Several criteria that should be considered in order to produce good reports are given below:

- Relevance,
- Accuracy,
- Clarity,
- Timeliness, and
- Cost.

8.12.1 Relevance

Only the information that is relevant to the purpose of the report should be present in the report. This is a selection process, i.e., all the relevant information should be included and all the irrelevant information or data should be excluded. Only required information should be printed or displayed. In on-line reports, we should use information hiding and provide methods to expand and contract levels of information details.

8.12.2 Accuracy

The data that appears on the report should be accurately recorded, accurately transmitted and accurately transformed into summary data. Accuracy is very important because if the data are inaccurate, then the main purpose of the report which is to provide accurate information to the user will not be accomplished. Incomplete data are also inaccurate.

8.12.3 Clarity

The information that is present on the report should be clear and understandable. The information present should be balanced on the report, the display should not be too crowded and not too spread out. Sufficient margins and spacing throughout the output will enhance readability. Desired information must be easy to locate. Comparisons, ratios, percentages, exception flags and graphs should be used where necessary.

8.12.4 Timeliness

Reports must be prepared and ready for use in time. Most reports provide information, which is used to make decisions. Hence, this information must reach the recipients while the information is pertinent to transactions or decisions. Information is of very little use if it arrives after the decisions are made.

8.12.5 Cost

Every report has two costs. First is the cost of preparation, which consists of analysis, design, computation and distribution. Second is the cost of reading the report and locating germane parts of it. Often the cost of reading the report is forgotten during the calculation of costs. The reading cost can be significantly reduced only if the appropriate information is presented clearly on the report. The total cost should always be less than the expected benefits. Only then the report should be prepared.

Check Your Progress 3

1. When can highlighting be used to convey special information to users?
   ……………………………………………………………………………………
   ……………………………………………………………………………………
   ……………………………………………………………………………………

2. What are the advantages and disadvantages of using colours when designing system outputs?
   ……………………………………………………………………………………
   ……………………………………………………………………………………
   ……………………………………………………………………………………

3. Specify the criteria used to identify a good report design.
   ……………………………………………………………………………………
   ……………………………………………………………………………………
   ……………………………………………………………………………………

4. Specify the criteria used to identify a good form design.
   ……………………………………………………………………………………
   ……………………………………………………………………………………
   ……………………………………………………………………………………

5. Describe how numeric, textual and alphanumeric data should be formatted in a table or a list.
   ……………………………………………………………………………………
   ……………………………………………………………………………………
   ……………………………………………………………………………………

8.13 SUMMARY

As we have seen, the main aim of this unit is to help the reader in selecting appropriate formats for conveying information on Forms and Reports. Here, we have seen that forms and reports are instruments of communication between people and computers.

In this unit, we have seen that Forms are used to collect or present information and Reports to convey information on a collection of items. Also, we have seen that information can take any of the three general forms:

Internal information is the information, which is used only within the organization.

External information is the information, which is used for distribution outside the organization.

Turnaround documents are documents that are produced for outside parties, but ultimately return (in part or in whole) as an input to internal system.

There are many ways in which information can be collected and formatted. Also, specific guidelines should be followed, as these guidelines will help in creating professional usable systems. Here, we have presented a variety of guidelines covering use of titles, layout of fields, highlighting data, use of colours, navigation between pages, formatting text and numeric data.

We have seen that before we start the designing of forms or reports, we should collect answers for questions like who, what, when, where, and how. Answers to these questions will help us to have a clear idea of what actually is required from the form or report to be designed. After getting answers to above questions, the initial prototype can be created.

A well designed form should satisfy the criteria like Organization, Consistency, Completeness, Flexible entry, Economy and a well design report should satisfy the criteria like relevance, accuracy, clarity, timeliness, cost, etc.

8.14 SOLUTIONS/ ANSWERS

Check Your Progress 1

1. In computer terminology, a Form identifies the data we want to collect. A report is the information that is organized and formatted to fit the required specification. The difference between Form and Report is that a form is used to collect or present information and reports are used to convey information on a collection of items.

2. Some of the advantages of using the forms are given below:

   - Forms provide an easy way to view data.
   - Using forms, data can be entered efficiently.
   - We can present data in attractive format using forms.
   - Forms offer convenient layout for entering, changing and viewing records present in the database.

   The advantages of reports are:

   - Reports organize and present data in groups.
   - With reports, we can calculate running totals, group totals, grand totals etc.
   - We can present data in attractive format.

- Within reports we can include sub-forms, sub-reports and graphs.

3. While designing Forms and Reports, the following steps have to be taken:

   a) The first is to collect all relevant information regarding the form or Report by asking questions like who, what, when, where and how.
   b) Refine the above information into an initial prototype.
   c) Review and evaluate the prototype.
   d) Design, validate and test the outputs using some combination of prototyping and code generation tools.

4. State True or false

   a) False
   b) False
   c) True

Check Your Progress 2

1. Internal information – For use within the organization
   External information – For distribution outside the organization
   Turnaround documents – Produced for outside parties but are again returned back as input to the Internal system.

2. The general formatting guidelines are:

   - Meaningful titles,
   - Meaningful information,
   - Balanced layout, and
   - Easy navigation.

3. The different sections of design specification are:

   - Narrative overview
   - Sample design
   - Testing and usability assessment

4. State True or False

   a) False
   b) True
   c) False

Check Your Progress 3

1. Special information such as the following should be conveyed using Highlighting to users:

   - Notifying the users in case of some errors,
   - Drawing attention to high priority messages, and
   - Provide any warning messages to users.

2. Advantages of using colours:

   - Soothes or strikes the eye,
   - Emphasizes the logical organization of information, and
   - Draws attention to warnings.

Disadvantages of using colours:

- Resolution may degrade with different displays
- Colour fidelity may degrade on different displays
- Printing or conversion to other media may not easily translate.

3. Criteria used to identify good report design are:
   - Relevance,
   - Accuracy,
   - Clarity,
   - Timeliness, and
   - Cost.

4. Criteria used to identify good form design are:
   - Organization,
   - Consistency,
   - Completeness,
   - Flexible entry, and
   - Economy.

5. Numeric data should be right justified and columns should be aligned using decimal point or delimiter. Textual data should be left justified and line length should be somewhere between 30 – 40 characters per line.

8.15 FURTHER READINGS

Jeffrey A.Hoffer, Joey F.George and Joseph S.Valacich; *Modern Systems Analysis and Design*; Pearson Education; Third Edition; 2002.

Jeffrey L. Whitten, Lonnie D. Bentley, Kevin C. Dittman; *Systems Analysis and Design Methods*; Tata McGraw Hill; Fifth Edition;2001.

Alan Dennis, Barbara Haley Wixom; *Systems Analysis and Design;*John Wiley & Sons;2002.

K.C.Laudon and J.P.Laudon; *Management Information Systems*; Pearson Education; Seventh Edition.

John W. Satzinger, Stephen D. Burd, Robert B. Jackson; *Systems Analysis and Design in a changing world*; Course Technology Inc.; Third Edition;2004.

Reference Websites

http://www.formdesk.com
http://www.functionx.com

UNIT 9 PHYSICAL FILE DESIGN AND DATA BASE DESIGN

Structure **Page Nos.**

9.0 INTRODUCTION

Once the logical system design has been finalized in consultation with the user, the details of physical design of the system can start. Physical database design involves actual implementation of the logical database in the DBMS. The requirements of physical design of the system require the logical design of the system. The database design involves three levels of design concepts namely Conceptual, Logical and Physical schemas. Conceptual model produces a data model which accounts for the relevant entities and relationships within the target application domain. Logical model ensures via normalization procedures and the definition of integrity rules that the stored database will be non-redundant and properly connected. Physical model specifies how database records are stored, accessed and related to ensure adequate performance. A good database design helps efficient storage and retrieval of data.

9.1 OBJECTIVES

After going through this unit, you should be able to understand the:

- advantage of databases over files;
- difference between logical and physical design;
- rules for good database design practices;
- define the concepts of fields, records and database;
- how to design the fields and records in a database table;
- understand various constrains enforced during database design;
- typical database design concepts; and
- explain the concepts of records, record types, and files, as well as the different techniques for placing file records on disk.

9.2 INTRODUCTION TO DATABASE DESIGN

Over a period of time, massive advancements have taken place in the field of databases. Storage of data and retrieval is an integral part of any information system.

9.2.1 Flat Files vs. Database

Traditionally, data are being kept in flat files. Consider an example of flat file which stores pin code of customer's address. Any change in the size of the field will enforce changes in the entire program which uses the data related to customers' address. If the file is being used by more than one application, the problem can become even worse.

9.2.2 Steps in Database Design

Analysis is the process of creating a conceptual data model independent of the target database technology. The typical result is an ER model.

Design is the process of creating a logical data model. This step is dependent on the target technology (relational, hierarchical, or network), but not on the specific database implementation (such as Oracle, MS SQL, DB2 for OS/390 or DB2 Universal Database).

Implementation is the process of creating a physical model or schema for one specific database system, such as Oracle, MS SQL, DB2. The result is an optimized physical design.

9.2.3 E-R Model to Database Design

The process of database design process involves the task of converting logical model to working physical model. The logical model can be arrived at by the application of normal forms. Normal forms are nothing but a set of rules applied sequentially starting from the basic table to the table resultant due to the application of a normal form. As of today, there are a total of five normal forms. The first normal form is applied to the basic table. The resultant table is given as input to the second normal form and so on. It is not mandatory to continue this process until the fifth normal form. Usually, the process is continued until third normal form. Fourth and fifth normal forms are applied only when there are multivalue dependencies and join dependencies respectively. We are not introducing you to the normalization theory in this course. It will be dealt in the course related to databases in detail. Physical database design starts from a given relational model. That is, from the definition of a set of tables and their respective columns. The objective of physical database design is to fulfil the performance requirements of a set of applications by optimizing the use of the DBMS. Key areas include: optimizing the index configuration, data placement and storage allocation.

Entities with one-to-one relationship should be merged into a single entity.

A table models each entity with a primary key and non-key attributes, some of whom may be foreign key(s).

One-to-many relationships are modeled by a foreign key attribute in the table representing the entity on the "many" side of the relationship.

Many-to-one relationship between two entities is modeled by a third table that has foreign keys that refer to the entities. These foreign keys should be included in the primary key of the relationship table, if appropriate.

Many commercially available tools can automate the process of converting a E-R model to a database schema. Figure 9.1 depicts various steps involved in the design of databases.

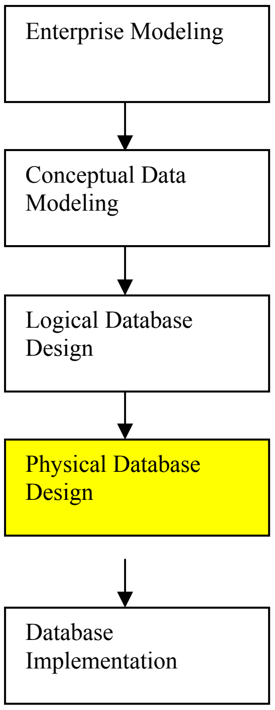


Figure 9.1: Steps in Database Design

9.2.4 Inputs to Physical Database Design

1. Logical structure of Database (Normalized relations).
2. Definition of attributes – data type, integrity control, error handling.
3. Choice of RDBMS
   a. Hierarchical
   b. Network
   c. Relational (DB2, MySQL)
   d. Object relational (Oracle 8i/9i)
4. Estimation of database size growth rate and frequency of usage.
5. Requirements for backup, recovery, response time and retention time.

9.2.5 Guidelines for Database Design

The following are various guidelines for Database Design :

- ensure that the data stored in the files (database tables) are atomic. Data stored in the atomic form can be combined later to generate data in specific form;
- every table must have a primary key which identifies each record in the table distinctly. Descriptive and meaningful name is to be used while naming a field in the table (For example, use *product_id* instead of *ID*);
- use single column primary key whenever possible. As most of the join operations are made on primary key and composite primary keys make the operation slower;
- use numeric key whenever possible;
- use primary key name as foreign key for better readability;
- avoid allowing null values to go into the columns that have discrete range of possible values; and
- avoid multiple tables with similar structure when one table is sufficient.

Figure 9.2 depicts various records and fields in a file which consist of details of various employees in a hospital.

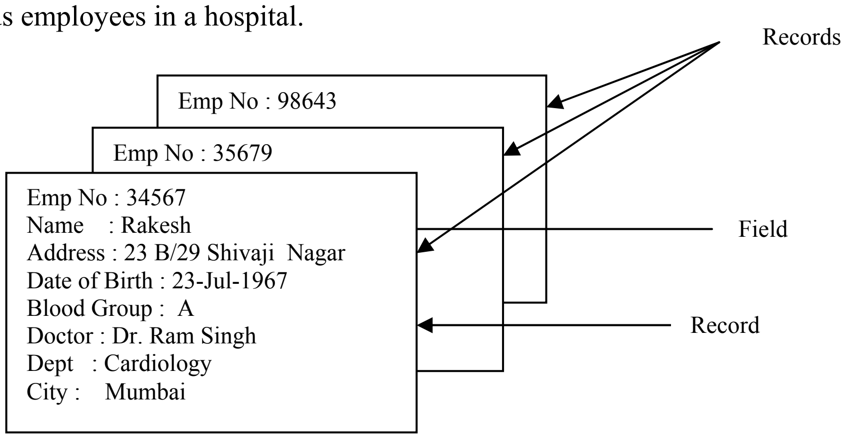


Figure 9.2: Records and Fields

Check Your Progress 1

1. The process of Database Design involves the task of converting logical model to ____________.
2. _______ types of relationship should be merged to a single entity.
3. Commercially available tools can automate the process of converting a ________to a database schema.

9.3 DESIGN OF DATABASE FIELDS

Attributes in E-R model are known as fields in physical data model. A field is the smallest unit of data that is stored and manipulated. Fields are used in conventional files as well as in databases. It is the implementation of attributes and can be termed as smallest unit of meaningful data.

9.3.1 Types of Fields

The following are various types of fields in databases:

Primary key: Any conventional file system or database stores two kind of fields namely descriptive fields and primary key. Descriptive fields comprise the customer names, inventory numbers, item descriptions, and so on, which are used by the application. Keys refer to the primary and foreign key that are used to find database records and relate them to one another.

Keys are fundamental to the concept of relational databases because they enable tables in the database to be related with each other.

A table must have a primary key i.e. an attribute or combination of attributes that are guaranteed to be unique and not null. It is sometimes helpful to introduce a surrogate field to act as a key. This could be a table attribute, which has no business meaning, but simply added to serve as a unique identifier for each record in the table. This is sometimes referred to as *plumbing*.

The requirements for a primary key are very hard. It must conform to the following rules:

- They should exist.
- Be unique in the table.
- The values must not change or become null during the life of each entity instance.
- It must have a not-null value for each instance of the entity.

Surrogate keys are often required because sometimes, real business data does not fulfil the requirement of a primary key. Furthermore, the surrogate key is typically a single field (not a composite key), which simplifies the database schema, particularly when the key is used in other tables as a foreign key.

Most of modern RDBMS are tuned in for queries on integers, so it is advisable to use this datatype as a primary key. Many RDBMS provide a special serial number or sequence number of integer type, which generate a sequence of unique integers as a row is inserted into the table. Declaring a column to be of this type guarantees that a unique key is generated for each inserted row.

Secondary key: Also known as *Alternate key*. This is a field or collection of fields in the table which can be used as primary key in addition to the already existing primary key.

Foreign keys are table attributes, the values of which are the same as those of primary keys of another table. It is often desirable to label foreign key columns explicitly. For instance, by adopting a naming convention. Existence of foreign key enforces the referential integrity constrains (discussed later in this Unit). A referential integrity constraint (references) should be declared as part of the CREATE statement in a DBMS while creating the table.

Descriptive fields: Attributes that are not used as key but store business data.

9.3.2 Rules for Naming Tables and Fields

Names for all database elements should be:

- Unique
- Meaningful
- Short

Restrictions for naming tables:

- Use no acronyms or abbreviations. Should be descriptive to convey meaning.
- Should not imply more than one subject

Restriction for naming fields:

- No acronyms
- Use abbreviations only if clear and meaningful
- Should not imply more than one subject
- Should be singular.

While designing database fields, it is required to set the properties of the fields.

Name: A name is used to refer the attribute in the DBMS that uniquely labels the field. The name of the attribute in the logical data model and the name of the field in the physical data model must be same. For example, *student name* in a *student* table.

Data type: Type of data the field is expected to store. This could be numeric, alphanumeric etc. The data type, supported by various RDBMS varies to a great extent. For example, student_name CHAR(25), indicates that the name of the student is of character data type, 25 indicates the maximum size of the data that can be stored in the field. The data type selected should ensure the following:

- it involves minimum usage of memory and represents all possible values
- supports all types of data manipulation that is expected from the business transaction.

Size: It indicates the size of the database fields. Many RDBMS support sizes that are variable. For example, VARCHAR data type in Oracle.

Null or not Null: specifies whether the field will accept null value. Not null constrains applied in DBMS ensure that null values are not entered to the respective fields. A null value is a special value distinct from 0 or blank. A null value indicates that the value is either missing or unassigned yet. We may specify that customer_name in a customer table to be not null. When a field is declared a primary key DBMS automatically ensures that the field in not null.

Domain: It indicates the range of values that are accepted by the fields. For example: Basic_Pay in a *employee* table can assume any value between the lowest basic_pay and highest basic_pay existing in the company. In such cases, the value of the field can be restricted to the one between the highest and lowest value to avoid entry of non-existing basic_pay.

Default value: It refers to the value that is stored by default in the field. For example, ship_date in a invoice is most of the time same as invoice_date (current date). When a default value is assigned to a field, it reduces a lot of data entry time and reduces the chances of error.

Referential integrity: It refers to a set of rules that avoid data inconsistency and quality problems. Referential integrity ensures that a foreign key value cannot be entered unless it matches a primary key value in another table. RDBMS automatically enforces the referential integrity once the database designer identifies and implements primary and foreign key relationship.

- It prevents orphaned records. i.e. when a row containing a foreign key is created, the referential integrity constrains enforced vide the RDBMS ensure that the same value also exists as a primary key in the related table.
- When a row is deleted, it should be ensured that no foreign key in related tables is the same value as primary key of the deleted row.

Figure 9.3 depicts primary and foreign keys for two tables namely *customer* and *order*.

Customer_id	**Customer_name**	**Customer_city**	**Customer_phone**
5466	John	New Delhi	2345678
5678	David	Mumbai	2567890

Table **Customer**

Order_no	**Order_date**	***Customer_id***	**Amount**
123456	12/3/2004	*5466*	Rs 345
345678	11/3/2003	*5678*	Rs. 567

Table **Order**

Figure 9.3: Primary key and Foreign key relationship

Customer_id is the primary key in **customer** table and is a foreign key in **order** table. This referential integrity constraints ensure that the value *customer_id* in **order** table must exist in the **customer** table. The primary key is shown in bold and the foreign key in intalics.

Check Your Progress 2

1. Attributes in E-R model are known as _______ in physical model of data.
2. __________ are often required because real business data sometimes does not fulfil the requirement of existence of a primary key.

3. A set of rules that avoid data inconsistency and quality problems is called _____________.

9.4 DESIGN OF PHYSICAL RECORDS

A Record is a collection of fields. Records are common to both databases and files. Records are collection of fields in a predefined format. The design of physical record involves putting the collection of fields in a single logical unit so that the fields are stored in adjacent locations for better storage and retrieval.

The main objective of the design of physical records is to store and retrieve them efficiently. Also, the fields should be stored in adjacent locations in such a way that the storage is used efficiently and speed of data processing is appropriate.

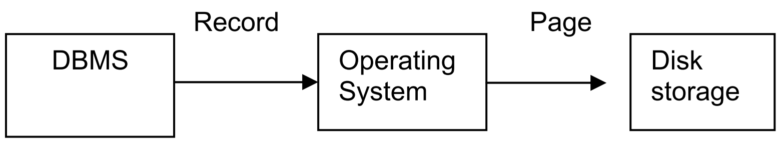


Figure 9.4: Process of storing a record physically

Physical pages or blocks are units of information moved between disk and memory buffers. They hold not only records, or table entries, but other information such as the amount of free space currently available in the block, the starting position of each record, etc. Blocks of data (pages) are normally read or written by the operating system. Page is referred to as the amount of data written in one I/O operation of operating system.
Blocking factor refers to the number of physical records per page. If a record size is 1340 bytes and the page size is 2048 bytes, then 708 bytes are wasted if DBMS does not allow physical records to span different pages. Selecting a block size involves a trade-off. In principle, the larger the block size, the fewer read-write operations need be performed to access a file by the operating system and therefore the more efficient is the processing. However, it requires a correspondingly large allocation of buffer space in memory. Since this is limited (and perhaps shared by many users), there is in practice, an upper bound. Moreover, large block sizes are primarily advantageous for sequential access.
Denormalization is the process of transforming normalized relations into unnormalized physical record specifications. The motivation behind de-normalization is poor performance of normalized table. The following may be of use for denormalization.

- Combine two entities with one-to-one relationship to one entity. This avoids the cost of joining two tables when the data are required from both the tables.
- Another form of de-normalization is to repeat the non key attribute (field) of one table in another table to facilitate the execution of query faster. However, it depends on the application at hand. Figure 9.5 depicts denormalization for optimized query processing.

Order_no	**Order_date**	***Customer_id***	**Amount**	**Customer_name**
123456	12/3/2004	*5466*	Rs 345	John
345678	11/3/2003	*8909*	Rs. 567	David

Figure 9.5: Denormalizations for Optimized Query Processing

In a particular application it is seen that queries about order also require the cutomer_name. In case of normalized table, this would always require joining **Customer** table and **order** table each time the query is processed. We have modified the **order** table by adding back the customer_name from the **customer** table in **order** table. Now all queries will require only the order table as all relevant information are available in this table.

Activities to enhance performance

1. Combining tables to avoid joins
2. Horizontal partitioning refers to placing different rows of a table into separate files. For example, in an **order** table order pertaining to different regions can be kept in a separate table for efficient retrieval of records.
3. Vertical partitioning refers to placing different columns of a table into separate files by repeating the primary key in each of the files.
4. Record partitioning refers to a combination of both horizontal and vertical partitioning as in distributed database processing
5. Data replication refers to the same data being stored in multiple places in the database.

Figures 9.6 and 9.7 depict table's structure in its original and improved versions.

Process of Denormalization of Tables

- Select the dominant process based on frequency of execution and frequency of data access;
- Define the join table for dominant process;
- Evaluate the cost of query, updates and storage for the schema. Consider possibility of renormalization to avoid table joins.

Fixed length records and variable length records

In fixed length record format, the length of record is always the same. The fixed length records are easier to design and implement but are more wasteful than variable length record formats when comes to utilization of space.

In variable length record format, the records are of variable length. To identify the end of a record, variable length records use end of record marker. It can be a special character.

Name	Date of Birth	Address	Phone
Rakesh	23-JUL-1967	12/B Patel Nagar, New Delhi - 110023	2234567
Ahbinav	8-NOV-1970	1201 Erose Appt., New Delhi - 110001	2236780
Roy	1-APR-1954	M-1034 Vanasthalipuram, Hyderabad-500001	2438723
Venkat	5-OCT-1969	24/A Gandhinagar, Vijayawada-520003	2658913
Ajay	5-AUG-1975	56 Canal Road, Patna- 600005	2753219
Sachin	8-SEP-1969	234 Mount Road, Varanasi-546987	2658703

Figure 9.6: Structure of **Students** table

Check Your Progress 3

1. The number of physical records per page is called __________.
2. Placing different rows of a table into separate files is called ___________.
3. The process of transforming normalized relations into un normalized table is called _____________.

Name	Date of Birth	Address	City	Pin Code	Phone
Rakesh	23-JUL-1967	12/B Patel Nagar	New Delhi	110023	2234567
Ahbinav	8-NOV-2001	1201 Erose Appt	New Delhi	110001	2236780
Roy	1-APR-1954	M-1034 Vanasthalipuram	Hyderabad	500001	2438723
Venkat	5-OCT-1969	24/A Gandhinagar	Vijayawada	520003	2658913
Ajay	5-AUG-1975	56 Canal Road	Patna	600005	2753219
Sachin	8-SEP-1969	234 Mount Road	Varanasi	546987	2658703

Figure 9.7: Improved Table Structure of Figure 9.6

9.5 DESIGN OF PHYSICAL FILES

Files are collection of logically connected records. In RDBMS, files are called tables. However, the way the files are stored in memory depend on the operating system. Many operating systems allow files to be split into pieces but the same is transparent to the user.

9.5.1 Types of Files

A **Master file** is a permanent file of all the data needed by a business. Some of the fields may be regularly updated from transactions. For example, a file consisting of information about customers.

A **Transaction file** is a temporary file of all the transactions (items bought, sold, deleted etc.) that have taken place in a given period. It stores data related to day to day business transactions. For example, data related to daily sales activity. The contents of these files are used to update the master file. Daily sales are used to update balance in the customer file.

An **Archive file** is a file of data in permanent storage (usually, the data is stored for legal reasons or to perform trend analysis). Archive files will contain all information about the past dealings of the business and would normally be stored on a different site to facilitate recovery in case of a disaster such as fire.

An **Audit file** is a file that does not store business data but data related to transaction log. For example, data and time of access, modification etc. of data , values of fields before and after modification etc.

A **Work file** is file temporarily created to hold intermediate result of the data processing. For example, a sorted file of list of customers.

9.5.2 File Organization

File organization is the physical organization of records on the disk. There are different types of file organizations depending on the organization of records in the disk and other secondary storage.

Before deciding on a specific file organization, we should ensure that its application leads to the following:

- Fast retrieval of records

- Reduce disk access time
- Efficient use of disk spaces.

Serial file organization

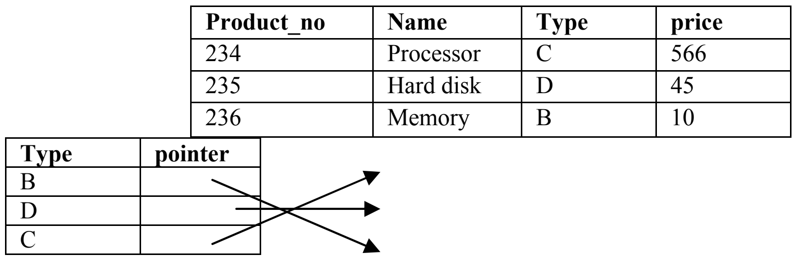
A serial file is created by placing the record as it is created. It leaves no gap between the records that are stored on the disk. The utilization of space called packing density approaches 100 percent in this case. Examples of serial files are print file, dump file, log files, and transaction files. These files are created once and are not used for addition or deletion or any kind of record searching operation.

Sequential file organization

In this organization, the records are physically ordered by primary key. To locate a particular record, the program starts searching from the beginning of the file till the matching primary key is found. Alphabetic list of customers is a common example of sequential file organization. Deletion of record may cause wastage of space and adding a new record requires rewriting of the file. This type of file organization is suitable for master files and is not used where fast response time is required.

Indexed sequential file organization

In this organization, records are not physically ordered. Index is created to facilitate searching of records. Index Records give physical location of each data record. Indexes are separate files with a link to the main file. This type of file organization is used where faster response time is required. Figure 9.8 depicts Indexed sequential file organization.



Product_no	Name	Type	price
234	Processor	C	566
235	Hard disk	D	45
236	Memory	B	10

Type	pointer
B	
D	
C	

Figure 9.8: Indexed Sequential File Organization

Hashed file organization

In this organization, records are physically ordered according to hashing algorithm. The address where each record is stored is determined using hashing algorithms. Figure 9.9 depicts an example of a Hashed file organization.
The following is a typical hashing algorithm :

1. Uses a field in record called the hash field (generally the key filed).
2. Divides by prime number known as hash function.
3. Produces record address called the hash address.

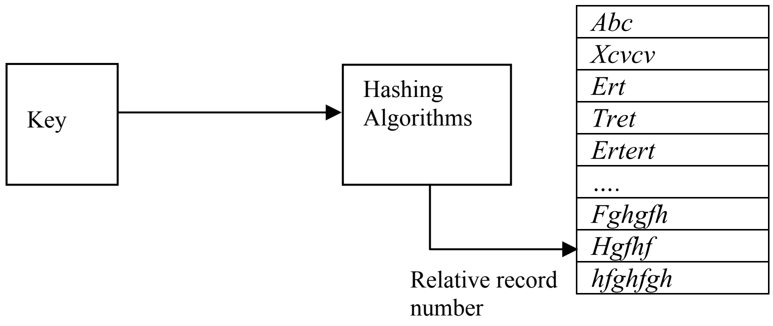


Figure 9.9 : Hashed File Organization

Check Your Progress 4

1. The Records are physically ordered by primary Key in __________ file organization.
2. ________ type of file organization is suitable for master files.

9.6 DESIGN OF DATABASE

Database design is similar to the pillars of a building. Any negligence, errors in Database design may lead to degraded performance of the software. In some cases such as real time applications, it may also lead to disasters.

The following are various steps in Database design :

- Selection of database architecture
- Designing database schema
- Selecting indexes
- Estimating capacity of the database.

Selection of database architecture

Selecting database architecture is one of the most challenging parts of database design for any information system**.** Before deciding on the target DBMS where the database is to be implemented, few considerations are required.

Hierarchical database structure is a kind of database management system that links records in tree data structure such that each record has only one owner. For example an order is owned by only one customer. It resembles a tree structure.

Network database structure is more flexible structure than Hierarchical model as well as relational model, but not preferred due to the high processing time. A neural network is an example of network database.

Relational database structure is most commonly used database model. The data is modeled as a mathematical relation. Reference key joins different tables together. Indexes provide rapid access to specific record in the database. For example, DB2, MySQL, Oracle are some RDBMS.

Object Oriented Database management system is based on object oriented paradigm. In object oriented database, data is stored as objects and can be interpreted only by using the methods specified by its class.

A blue print of the database is a physical model. A schema defines the database in terms of tables, keys, indexes and integrity rules. A **relational schema** consists of a relation, name of the attributes in the relations and restrictions on the relations called integrity constraints.

A **database schema** is a set of relation schemas. Changes to a schema or database schema are expensive. So, careful thought must be given to design of a database schema. The following are some guidelines for the design of a database schema.

- Each entity should be implemented as a database table.
- Each attribute should be implemented as a field.
- Each table must have a primary key and an index based on the key.
- Each table may have zero or more secondary keys.
- Appropriate foreign keys.

The following is an example of a database schema:

employee(emp_name, birth_date, social_security_no, dept_name),
department(dept_name, function).
Figure 9.10 depicts the database schema of the above.

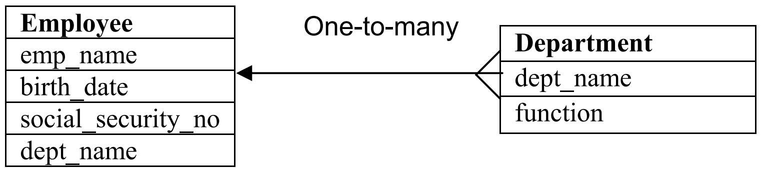


Figure 9.10: A Database Schema

A database schema defines a database in terms of tables, keys, indexes and constraints.

The following are various objects of a schema :

- Fields
- Records
- Tables
- Database

Selecting Indexes: During the process of file and database design one must choose index based on a single field (usually the primary key) or multiple fields. While selecting index, one must keep in mind the performance issues vis-à-vis the issue of inserting and deleting records. While indexes can be used generously on tables primarily used for query purpose with rare necessity to update records, they should be used judiciously in tables that support transaction processing which involve large insertion, updation and deletion operations. The amount of time needed to maintain the indexes in database tables increases with the number of rows stored.

When an index is created on a table, a separate storage area is allocated to store the index structure. A database table can have one or more indexes associated with it.

Figure 9.11: An index created on the EMP_ID field (one.jpg)

For example, consider an **employee** record. Figure 9.11 depicts an index created on the **EMP_ID** field of the **employee** table. The index table contains a sorted list of the employee ID values**.** Indexes may significantly improve the performance of SQL queries. It may not be noticed with small tables but it can be quite significant for larger tables. However, there are disadvantages to having too many indexes on table. Indexes can slow down the speed of some inserts, updates, and deletes when the

DBMS has to maintain the indexes as well as the database tables. Also, indexes take additional disk space as they are stored separately.

Example

In an **employee** database table, one will retrieve records based on employee name, department, or hire date. One may create three indexes-one on the DEPT field, one on the LAST_NAME field, and one on the HIRE_DATE field. The indexes are created on fields that appear in **where** clause of select statements.

If the boolean operator in the where conditions is AND (for example, CITY = 'Mumbai' AND STATE = 'MH'), then a concatenated index on the CITY and STATE fields will help. This index is also useful for retrieving records based on the CITY field.

If the boolean operator in the where condition is OR (for example, DEPT = 'EDP' OR HIRE_DATE > '01/30/03'), an index does not help performance. Therefore, index file need not be created.

Estimating capacity of the database: Database administrator needs to calculate the amount of disk space required for the database.

1. In a given table, calculate the record size by adding the sizes of various fields in it.
2. Also, the number of records that will be present in each table at a particular period of time should be forecast.

   Table size = record size * number of records in the table.

Database size is sum of sizes of all tables in that database. As rule of thumb, add a factor of 50% for indexes and other overheads to get the expected database size. While designing a database, future growth of database should also be kept in mind. Most of the business databases have a liner growth trend. Figure 9.12 depicts the growth of a database in relation to time.

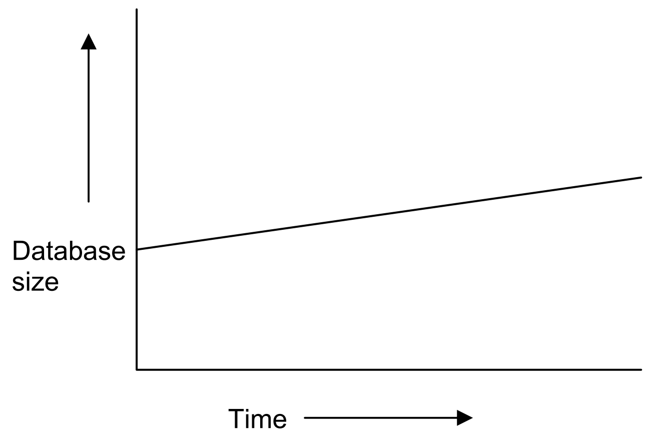


Figure 9.12 : Growth of Database

Check Your Progress 5

1. ________ is a blue print of the database.
2. A neural network is an example of __________.
3. Indexes take additional disk space? Yes/No.

9.7 CASE STUDY

The physical database design is the process of transforming a logical data model into an actual working physical database. A logical data model is required before designing a physical database. We will assume that the logical data model is complete, though. We will consider physical database design targeting a relational DBMS.

The very first step is to create an initial physical data model by transforming the logical data model into a physical implementation based on the target DBMS to be used for deployment. In reality, the physical data model depends largely on the target DBMS to be used for implementing the database. Therefore, to successfully create a physical database design requires a good working knowledge of the features of the DBMS including:

- knowledge of the database objects supported by DBMS, the physical structures and files required to support those objects;
- details regarding how the target DBMS supports indexing, referential integrity, constraints, data types, and other features that augment the functionality of database objects;
- knowledge of the DBMS configuration parameters and limitations; and
- data definition language (DDL) to translate the physical design into actual database objects.

We can create an effective and efficient database from a logical data model,i.e. E-R Diagram. The first step in transforming a logical data model into a physical model is to perform a simple translation from logical terms to physical objects. It may be noted that this simple transformation will not result in a complete and correct physical database design – it is simply the first step and can act as a template to further refining the database design. The transformation consists of the following steps:

- transforming entities in ER diagram into database tables;
- transforming attributes into table columns; and
- transforming domains into data types and constraints to primary key and foreign key relationship.

Deciding a primary key is an integral part of the physical design of entities and attributes. A primary key should be assigned for every entity in the logical data model. One should try to use the primary key as selected in the logical data model. However, if suitable attribute is not available, multiple attributes can be used as primary key. It may be required to choose a primary key other than the attributes in logical design by inserting a column that does not store business data (surrogate key) for physical implementation.

In a physical database, each table column must be assigned a data type supported by the DBMS. Certain data types require a maximum length to be specified, e.g.,a character data type could be specified as CHAR(25), indicating that up to 25 characters can be stored in the column.

Figure 9.13 depicts the E-R diagram of an **employee** database.

After following normal steps of transformation described above, further refinement of relations is possible before implanting on the target DBMS. One such example is shown below.

In the above ER diagram, Department being a multivalued attribute, we will convert this multivalued Attribute into relation namely Department.

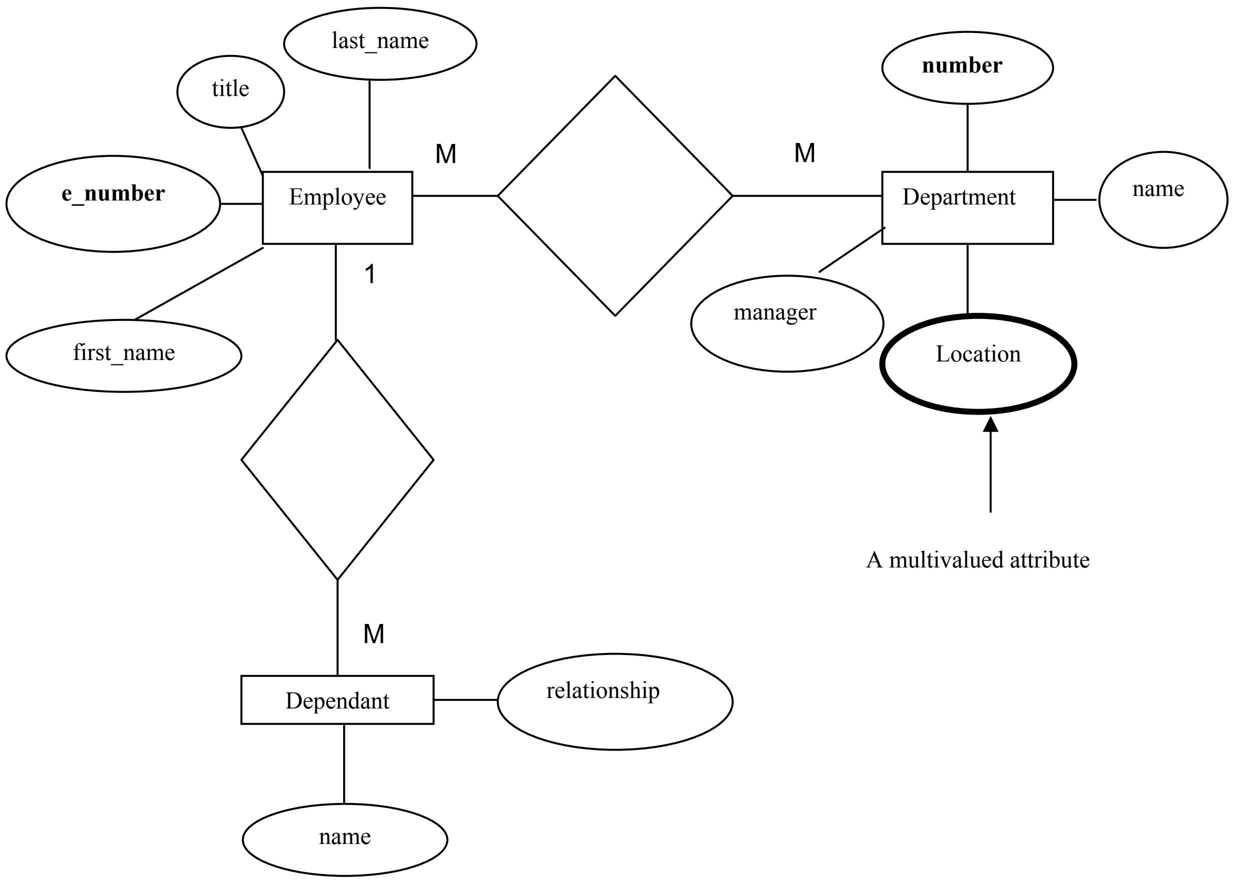


Figure 9.13: E-R Diagram for Employee database

e_number	Title	Last_name	First_name	Supervisor_emp_no

EMPLOYEE

number	Name	Manager

DEPARTMENT

Dept_no	Location

LOCATION

Emp_number	**Name**	relationship

DEPENDENT

Figure 9.14: A Relational Database Schema

Figure 9.14 shows the nearest relational database schema for the E-R diagram depicted in Figure 9.13. The aim of database design should be to create a database plan to fit present and future objectives. A logical data model should be used as the blueprint for designing and creating a physical database. But, the physical database cannot be created properly with a simple logical to physical mapping. Physical database design decisions need to be made by the Data Base Administrator before implementing physical database structures to the target DBMS. Sometimes, this may require deviation from the logical data model.

9.8 SUMMARY

Traditionally, file is being used for keeping data. Use of DBMS has been the standard to store data for today's information systems due to their various advantages. Any physical database design involves steps like choice of target DBMS on which the database is going to be implemented. Relational database is mostly being used unless the application has specific requirements. The physical database design process can be considered as a mapping from logical model to physical working database, which involves design of fields, design of records and finally design of the database. While transforming the logical model to physical model, many implementation issues related to the information system and target DBMS are to be addressed. Database volume estimation is an important part of database design. The present size and future growth of database is to be estimated before implementing the database.

9.9 SOLUTIONS/ ANSWERS

Check Your Progress 1

1. Working physical model
2. One-to-one
3. E-R Model

Check Your Progress 2

1. Field
2. Surrogate keys
3. Referential integrity rules

Check Your Progress 3

1. Blocking factor
2. Horizontal partitioning
3. Denormalization

Check Your Progress 4

1. Sequential
2. Sequential

Check Your Progress 5

1. Database Schema
2. Network database
3. Yes

9. 10 FURTHER READINGS

Jeffrey L. Whitten, Lonnie D. Bentley, Kevin C. Dittman; *Systems Analysis and Design Methods*; Tata McGraw Hill; Fifth Edition;2001.

Joey George, J Hoffer and Joseph Valacich; Pearson Education, *Modern System Analysis and Design*;2001.

Reference Websites

http://seastorm.ncl.ac.uk/itti/create.html#create
http://www.rspa.com

UNIT 10 CASE TOOLS FOR SYSTEMS DEVELOPMENT

Structure **Page Nos.**

10.0 INTRODUCTION

Revolutionary changes are occurring in the traditional process of software system development. The normal SDLC process is often seen as inflexible and time consuming and expensive. Keeping in view of these limitations of SDLC process, the Computer Aided Software Engineering process has emerged to help organizations to develop systems and software. CASE involves using software packages called CASE tools to perform and automate many activities of system development life cycle. The CASE tools are helpful in business planning, project management, user interface design, database design and programming. Use of CASE tools makes computer-aided software development possible. In today’s scenario where quick product delivery could be challenging task for the software vendor, CASE tools have come as helping hand to assist organization manage the software development process and automate certain processes of these activities. It is noteworthy that some capabilities of CASE are found in almost every modern software development tool. Some automatically design the user interface, few can auto generate codes, etc. It is difficult to imagine the life of a programmer without CASE tools.

10.1 OBJECTIVES

After going through this unit, you should be able to:

- know the role of CASE tools and their use by the organizations;
- know various components of CASE tools;
- explain Visual CASE tools and know about some such commercial tools;
- describe the sophisticated CASE tools; and
- describe Object Oriented CASE tools with their utility in development of software.

10.2 USE OF CASE TOOLS BY ORGANIZATIONS

CASE stands for **C**omputer **A**ided **S**oftware **E**ngineering

CASE is the use of computer-based support in software engineering process. The support could be of any type like managerial, technical or administrative on any part of the software development process. All the software that helps in the process of software engineering can be termed as CASE tools.

10.2.1 Definition of CASE Tools

A CASE tool is a computer-based product aimed at supporting one or more software engineering activities within a software development process.

Although, ideally a CASE tool should support all the activities of software engineering process starts from requirements analysis to designing, coding, testing, implementation and documentation, in reality, CASE tools often support one activity or at least a group of related activities.

As software development activities become more complex and relatively unmanageable, there has been an awareness of the need for automated tools to help the software developer to accomplish this task. Initially, the focus was primarily on program support tools such as design of translators, compilers, assemblers, macro processors, and other tools. As computers became more powerful and the software that ran on them has grown larger and more complex, the support tools began to expand further. Some capabilities of CASE tools are also found in the common application development software.

Large-scale use of computers has necessitated its maximum and efficient use and development of software for various activities of any organization. A software development effort can be viewed as a significant effort to design appropriate solutions, test and implement the solutions and finally documenting the solutions. In view of this, a wide range of support tools began to emerge to help the development team.

10.2.2 Use of CASE tools by organizations

The following are some of the ways in which CASE tools are used :

- **To facilitate single design methodology:** CASE tools help organization to standardize the development process. It also facilitates coordinated development. Integration becomes easy as common methodology is adopted.
- **Rapid Application Development:** Organizations use CASE tools to improve the speed and quality of system development.
- **Testing:** CASE tools help to ease and improve testing process through automated checking and simplify program maintenance.
- **Documentation:** In traditional software development process, the quality of documentation at various stages depend on the individual. CASE tools improve the quality and uniformity of documentation at various stages of SDLC. It also ensures the completeness of the documentation.
- **Project Management:** It improves project management activity and to some extent automates various activities involved in project management.
- **Productivity and reduction of cost:** Use of CASE tools makes the software easy to maintain and hence reduce the maintenance costs. Automation of various activities of system development and management processes increases productivity of the development team.

10.2.3 Role of CASE Tools

CASE tools play a major role in the following activities:

- Project management
- Data dictionary
- Code generation
- User interface design
- Schema generation
- Creation of meta-data for data warehouse
- Reverse engineering
- Re-engineering
- Document generation
- Version control
- OO analysis and design
- Software testing
- Data modeling
- Project scheduling
- Cost estimation

CASE technology has resulted in significant improvements in quality and productivity. An ideal CASE tool should support all facets of system development like analysis, design, implementation, testing and maintenance. All aspects of software engineering process are not supported by today's CASE tool. Most of the CASE tools provide good support for data modeling, object oriented design and programming. Also, they moderately support testing and maintenance.

10.2.4 Advantages of CASE Tools

The following are some advantages of CASE tools:

- **Integrated development environment:** CASE tools provide unique user interface for the developer and analyst, automate time consuming and tedious activities like code generation.
- **Guidance in development:** It provides common platform for all the developers and helps methodical system development.
- **Consistency between the model and documentation:** Documentation is generated out of the model automatically leading to consistency between the model and documentation.

10.2.5 Disadvantages of CASE Tools

The following are some of the disadvantages of CASE tools:

- Complex functionality
- Many project management problems are not amenable to automation. Hence, CASE tools can't be used in such cases.

Check Your Progress 1

1. Systems analysts use automated software tools called _______ to develop information systems.
2. Organizations use CASE tools to increase the _______ and ________ of Systems development.
3. If a particular software development activity is not amenable to ________, then CASE tools cannot be used.

10.3 COMPONENTS OF CASE

The activity that can be automated, whether partially or fully, depends on the CASE tools that are used. Most of the CASE tools generate a working model or prototype, which makes development process faster and easier.

10.3.1 Types of CASE Tools

The following are various types of CASE tools:

- **Planning and management tools:** Begin the development process with information planning and project management.
- **Analysis tools:** These tools ensure that business requirements are correctly captured during the analysis phase early in the development process. Analysis tools are used to check for incomplete, inconsistent or incorrect specifications.
- **Design toolset:** It provides detailed specification of the system.
- **Information integrator:** It integrates system specifications and checks them for consistency and completeness. It also records them in the CASE repository.
- **Code generator:** It automatically generates code specific to a language based on the system specification.
- **Database design toolset:** It suggests database design and generates system control information.
- **User interface generator:** It generates user interface based on system specification.
- **Report generator:** It generates reports based on specification

Figure 10.1 depicts various components of a typical CASE tool.

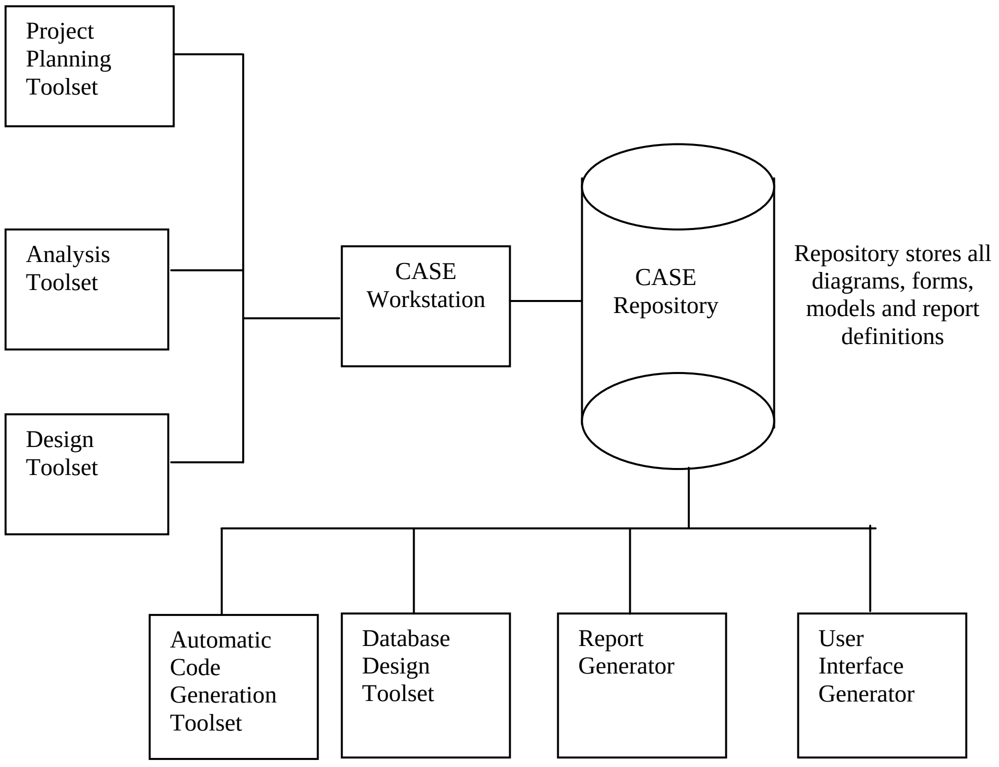


Figure 10.1: Components of a typical CASE tool

All case tools are based on prototyping which is particularly useful when the user requirements are difficult to define. Large systems use traditional SDLC approach but part of the system can be prototyped. The prototype is then repeatedly refined till it becomes acceptable.

10.3.2 Classification of CASE Tools

Figure 10.2 depicts various types of CASE tools.

Although CASE tools can be classified depending on the functionalities, these can be broadly classified into five generic categories:

- **Development tools:** These tools are interactive in nature. They are used for design support and code generation.
- **Front-end tools:** They support activities early in the life cycle of a software development process (planning, analysis and design). Examples are data flow diagram, data structure diagram, ER diagram, prototyping tools, etc.
- **Back-end tools:** They support activities later in the life cycle of a software development process (Implementation and maintenance). Examples are program flow chart, program editor, debugger, code generators etc.
- **Horizontal tools:** These tools are not specific to a particular life cycle step but are common across a number of life cycle steps e.g., Documentation tool.
- **Vertical tools:** These tools are specific to a life cycle.

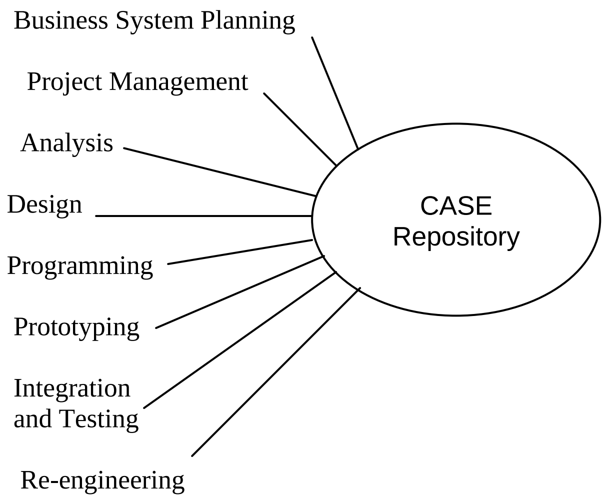


Figure 10.2: Types of CASE Tools

10.3.3 Reverse and Forward Engineering

We will discuss two important concept related to CASE tools namely, Forward Engineering and Reverse Engineering. Figure 10.3 depicts both Forward and Reverse Engineering.
Reverse engineering is the process of recreation of model based on existing code. First, the existing code is scanned to generate the model. Then the model can be fine tuned in accordance with requirements. Reverse engineering allows developers to create model for old systems, which were never modeled. It analyses existing software with purpose of understanding its design and specification. Reverse engineering tools read program source code and create graphical and textual representation of design.

Forward engineering is the process of generation of skeleton code out of the models. First step is to create the model for a system, then generate the relevant code for the model and then allow modification of this code in tune with the requirements.
Re-engineering means “restructuring and rewriting the legacy system or part of it without changing its original functionality”. Re-engineering efforts make the software up-to-date to current technology and hence easy to maintain. The new system becomes restructured and re-documented. Re-engineering tools read program source code and interactively change and existing system to improve quality performance or maintainability.

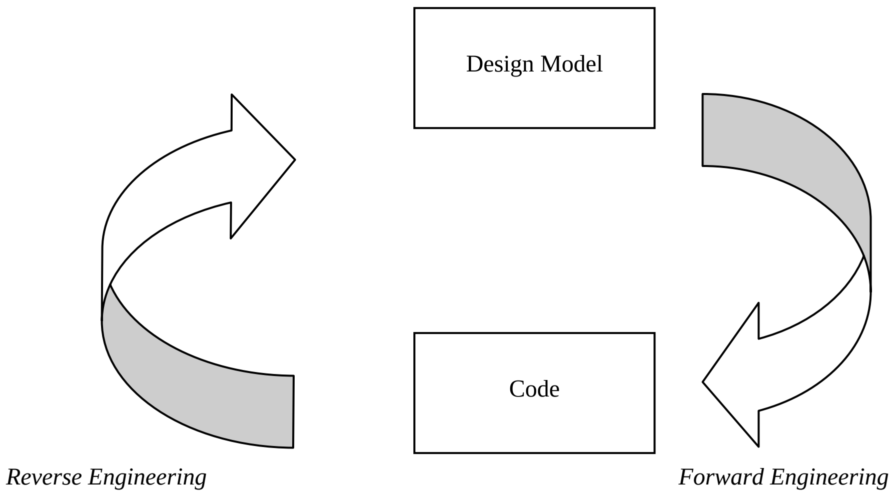


Figure 10.3: Reverse Engineering and Forward Engineering

Check your progress 2

1. After Re-engineering, the new system becomes ___________ and ___________.
2. __________ is the process of generation of skeleton code out of the design models.
3. All case tools are based on __________, which is particularly useful when the user requirements are difficult to define.

10.4 VISUAL AND EMERGING CASE TOOLS

Visual CASE tools enable user to quickly create user interface and related skeleton code.

10.4.1 Traditional systems development and CASE based systems development

The following are the features of Traditional systems development:

- Emphasis on program coding, testing and documentation
- Specifications are to be written by the analyst and are paper-based
- Coding is manual and tedious
- Documentation is written by the programmer or the analyst and are often done after completion of the process
- Software testing process follows the traditional approach
- Manual maintenance of code and documentation.

The following are the features of CASE-based systems development:

- Emphasis is on analysis and design
- Rapid and interactive prototyping of models
- Automated code generation
- Automated documentation generation
- Automated design checking
- Maintain design specifications.

10.4.2 CASE environment

The earlier generation of CASE tool developers concentrated to a large extent on the automation of isolated tasks of system development process, such as document production, version control of source code, and design method support. Need of integrating these tools to support a common development environment was felt to support the activity effectively. A typical CASE environment consists of a number of CASE tools and related components for supporting most or all phases of system development life cycle that operates on a common hardware and software platform. CASE environment is not just a random amalgamation of CASE tools, it provides proper interaction between the CASE tools. One should concentrate less on which components should be chosen, and much more on how the selected components can be made to work together effectively. Figure 10.4 depicts typical CASE environment.

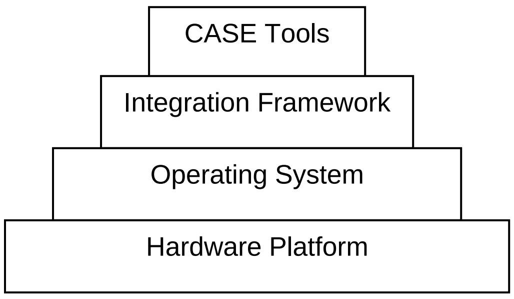


Figure 10.4 : CASE environment

The key to CASE tools is integration. Tools are more effective if they are integrated to work together. Integration could be data integration, user interface integration or activity integration.

Examples of visual CASE Tools

- Oracle 2000 Designer
- Evergreen EasyCase
- Aonix: Software Through Pictures
- Popkin System Architect
- Cadre Teamwork
- ColdFusion
- Rational Rose
- Visual CASE
- Enterprise Architect.

Rational Rose is one of the most widely used CASE tools by the software community. The teams responsible for software development are finding that modeling using CASE tools are becoming increasingly important for the software development process. Software developed using model driven technique such as Rational Rose offer a range of products to suit different activities of software development process.

UML modeling: The Unified Modeling Language, or UML is mostly a graphical modeling language that is used to express designs. It is a standardized language in which artifacts and components of a software system can be specified. It is important to understand that UML simply describes a notation and not a process. It does not put forth a single method or process of design, but rather is a standardized tool that can be used in a design process.

The following are some of the tasks that can be performed by using CASE tools:

- UML modeling
- Code generation/construction for Visual C++, Visual Basic, C++, Ada, JAVA
- Database design
- Fully executable codes for C, C++,etc. across platforms
- Component testing.

Another CASE tool is Enterprise Architect (Parx System). This is an object oriented CASE tool for the entire software development life cycle. Its features are:

- Business process modeling
- Forward and reverse engineering
- Automation interface
- Support for C++, Java, VB, VB.Net, Delphi
- Project estimation tool
- Testing tool
- User interface design
- Requirement gathering
- Components model
- Deployment model

10.4.3 Emerging CASE Tools

Initially CASE tools were not very sophisticated in terms of the process they support. Now, integrated CASE tools have emerged to support the entire gamut of system engineering and software development process.

Integrated CASE (I-CASE)

- It offers automated systems development environment that provides numerous tools to create diagrams, forms and reports.
- All tools share one common user interface.
- User has the feeling of working on one tool.
- Provide analysis, reporting and code generation facilities.
- Seamlessly shares and integrate data across and between tools.
- Repository is a central place to store information that is to be shared between various tools.

Check Your Progress 3

1. Oracle Designer 2000 is a _________ CASE tool
2. Enterprise Architect is an object oriented CASE tool which can be used during ______ phases of life cycle
3. All integrated CASE tools usually share _________ user interface

10.4.4 Object Oriented CASE Tools

Object oriented CASE tools are similar to other CASE tools. These differ from others only in terms of their capability to create class diagrams and text specifications for reports. Object oriented CASE tools support an O-O methodology such as *Rumbaugh's Object Management Technique (OMT).*

Examples of object oriented CASE tools

A large number of object oriented CASE tools are available in the market. These include: *Paradigm Plus* from Protosoft, *Rational Rose* from Rational and *WithClass* from MicroGold Software. Our discussion here will describe object oriented CASE tools in general without making references to a specific product.

The following are some of the features found across most of the Object Oriented CASE tools:

1. Create graphics, such as class diagrams, message diagrams, state diagrams etc.
2. Create text specifications such as system specification, class specification and relationship specification.
3. Generate source code.
4. Repository of models.

Differences between various types of object oriented CASE tools

All of these object oriented CASE tools are similar in terms of their capabilities to create class diagrams, code generation. However, they may differ in terms of their extendibility, number of supported operating systems and additional features and capabilities, such as support for different methodologies and computer language code generation (C++, Ada, Java etc). Most provide capability to automatically generate code of C++ and other languages from a class diagram and class specifications. Some provide the capability to generate class diagrams from code (reverse engineering). Figure 10.5 depicts drawing and text tools.

An object oriented CASE tool has the capabilities of drawing class diagrams and state diagrams. Specification tools create text reports.

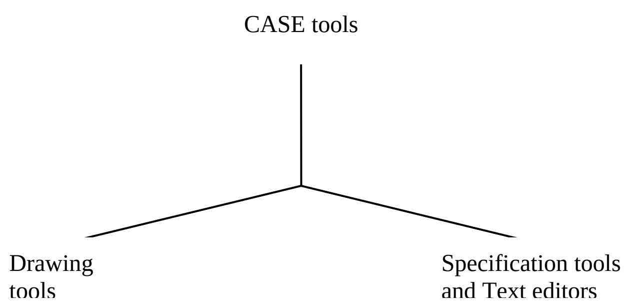


Figure 10.5: Drawing and text tools

Figure 10.6 depicts major outputs of object oriented CASE tools.

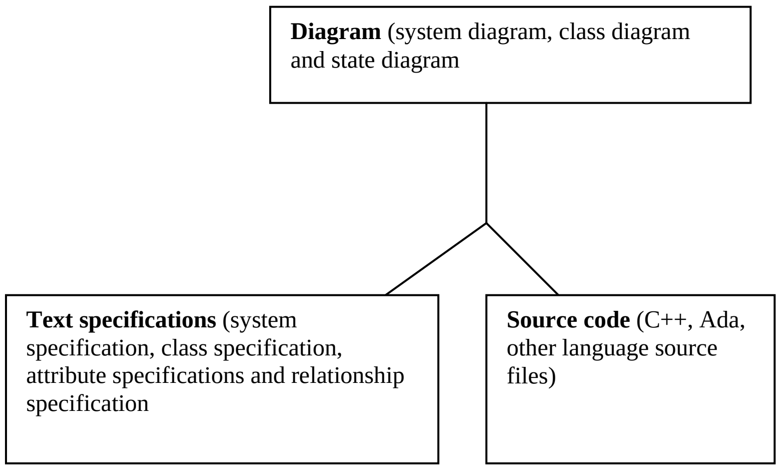


Diagram (system diagram, class diagram and state diagram
Text specifications (system specification, class specification, attribute specifications and relationship specification
Source code (C++, Ada, other language source files)

Figure 10.6: Major Outputs of Object Oriented CASE Tools

The class diagram is core to object-oriented design. It describes the types of objects in the system and the static relationships between them. The core element of the class diagram is the class. In an object-oriented system, classes are used to represent entities within the system. Entities are often related to real world objects.

Figure 10.7 depicts the class diagram for a typical Banking application.

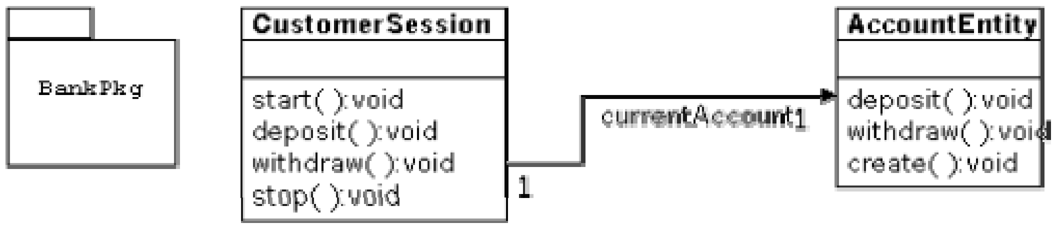


Figure 10.7 : A class Diagram for a typical Banking Application

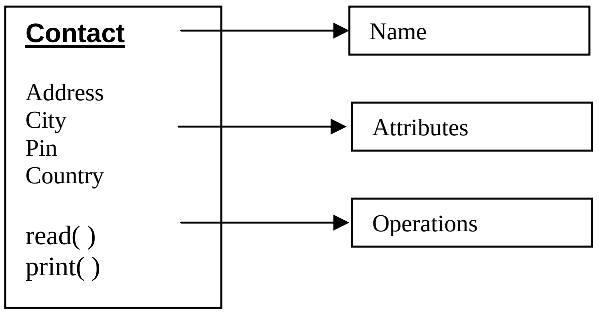


Figure 10.8: An Example of a Simple Class *Contact*

10.4.5 Creating documentation and reports using object oriented CASE tools

A system requirement describes a condition or capability which a system must conform to, either directly from the user need or derived from or stated in a contract, standard, specification, or other formally imposed document.

Most of the object oriented CASE tools assist in documenting as well as in object oriented analysis and design. Object oriented CASE tools have capability to import graphics from other tools. The system requirement is given shape in sketches with pencil and a paper. Then, it is used by the drawing tools and text editor available to create system diagram, class diagram and other diagrams. Document processor does it all to create a model document from these models. Figure 10.9 depicts the process to create a model document.

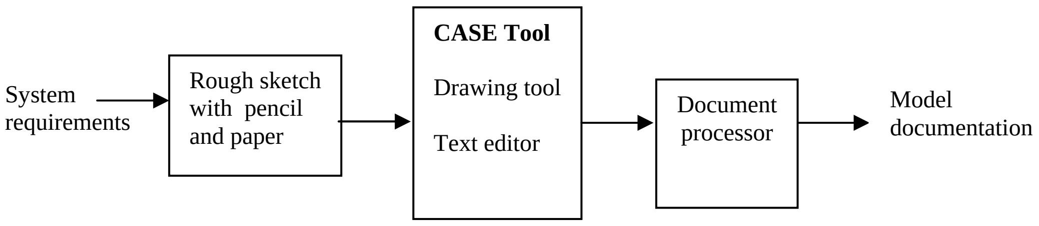


Figure 10.9 : Process to create a model document

10.4.6 Creating an executable prototype using object oriented CASE tools

Most of the object oriented CASE tools greatly assist in creating executable prototypes based on design specifications. The diagram below shows how CASE tools are used to create executable codes. System requirements are prepared in text and diagrams by pencil and paper. Design specifications like class specifications, system diagrams and various text specifications are then generated using tools available in CASE. From this, design specification codes are generated using code generation tools. Most of the CASE tools support generation of C++ code whereas some may support other languages as well. The source code generated might require updation with formulas, expression, etc. CASE tools also create updated diagrams based on updated source code. Figure 10.10 depicts the process to create an executable prototype using a CASE tool.

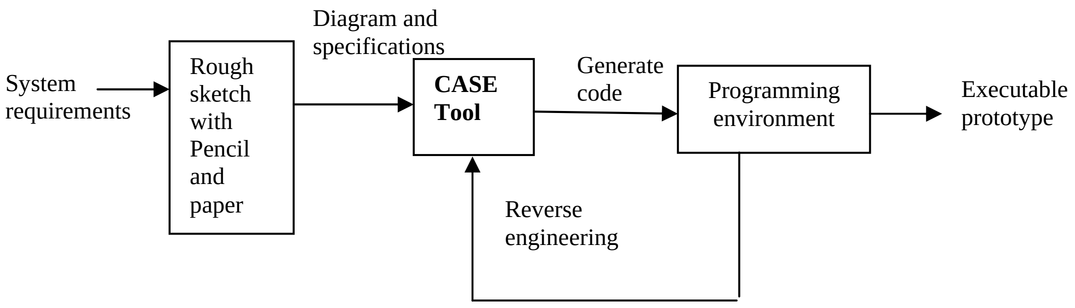


Figure 10.10 : Process to create an executable prototype using a CASE tool

10.4.7 Sequence Diagrams

UML provides a graphical means of depicting object interactions over time in Sequence Diagrams. These typically show a user or actor, the objects and components they interact with in the execution of a use case. One sequence diagram typically represents a single Use Case scenario or flow of events.

Sequence diagrams are an excellent way to document usage scenarios and to both capture required objects early in analysis and to verify object usage later in design. Sequence diagrams show the flow of messages from one object to another, and as such correspond to the methods and events supported by a class/object.

Figure 10.11 shows an example of a sequence diagram, with the user or actor on the left initiating a flow of events and messages that correspond to the Use Case scenario. The messages that pass between objects will become class operations in the final model.

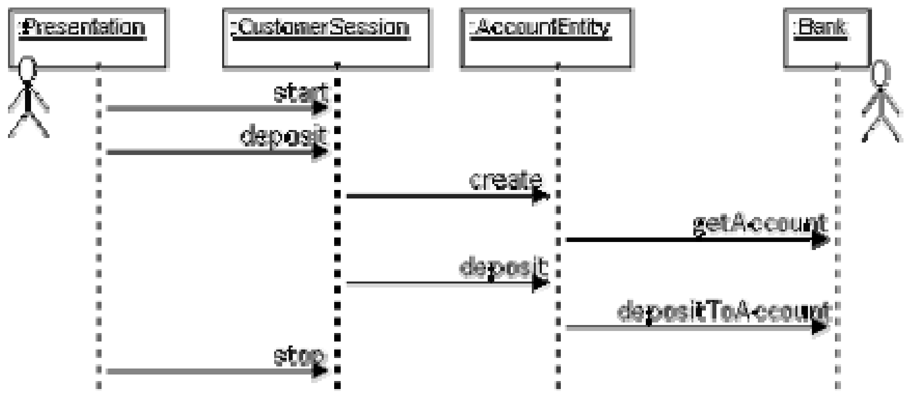


Figure 10.11: A typical Sequence Diagram for deposit into Account

Check Your Progress 4

1. The process of generating class diagram from code is called ______
2. _______________ are an excellent way to document usage scenarios and to both capture required objects early in analysis and to verify object usage later in design.

10.5 SUMMARY

Changes are occurring in traditional way of system development. Organizations use CASE tools to improve integration and development activities. The tools also help standardize the development process. Computer Aided Software Engineering (CASE) automates part of system development process. A CASE tool provides iterative and interactive tools for various activities like user interface design and code generation. Modern CASE tools are aimed at supporting the entire activity of system development. Hence, CASE tools, known as I-CASE (Integrated CASE) has evolved. CASE tools are classified as front-end tools or back-end tools depending on which part of SDLC process they automate. Front-end tools automate initial part of SDLC process whereas back-end tools automate process that come later in SDLC life cycle.

10.6 SOLUTIONS/ ANSWERS

Check Your Progress 1

1. CASE tools
2. Speed, quality
3. Automation

Check Your Progress 2

1. Restructured, Redocumented
2. Forward Engineering
3. Prototyping

Check Your Progress 3

1. Visual
2. All
3. Common

Check Your Progress 4

1. Reverse engineering
2. Sequence diagrams

10.7 FURTHER READINGS

Joey George, J. Hoffer and Joseph Valacich; *Modern Systems Analysis and Design, Third Edition, 2001*, Pearson Education.

James A. O’Brien; *Introduction to Information Systems, An End user/Enterprise Perspective*; Mc Graw Hill Edition; 1995

Reference Websites

http://www.sei.cmu.edu/legacy/case/case_whatis.html
http://www.rational.com/product

SOCIAS-IGNOU/P.O.10.5T/MAY, 2004

ISBN-81-266-1230-4