

COMPUTER STUDIES

રાજ્ય સરકારની વિનામૂલ્યે યોજના હેઠળનું પુસ્તક



Standard
11

ગુજરાત રાજ્યના શિક્ષણવિભાગના પગા-કમાંક
મશબ/1213/110/ઇ, તા. 18-3-2013-થી મંજૂર

COMPUTER STUDIES

Standard 11



PLEDGE

India is my country.
All Indians are my brothers and sisters.
I love my country and I am proud of its rich and varied heritage.
I shall always strive to be worthy of it.
I shall respect my parents, teachers and all my elders and treat everyone with courtesy.
I pledge my devotion to my country and its people.
My happiness lies in their well-being and prosperity.

રાજ્ય સરકારની વિનામૂલ્યે યોજના હેઠળનું પુસ્તક



Gujarat State Board of School Textbooks
'Vidyayan', Sector 10-A, Gandhinagar-382010

© Gujarat State Board of School Textbooks, Gandhinagar
Copyright of this book is reserved by Gujarat State Board of School Textbooks.
No reproduction of this book in whole or in part, in any form is permitted without written permission of the Director, Gujarat State Board of School Textbooks.

Subject Adviser

Prof. R. P. Soni

Authors

Dr Harshal Arolkar (Convener)
Dr Sonalben Jain
Dr Nilesh Modi
Shri Triptiben Dodiya

Reviewers

Dr Kuntal Patel
Shri Jyotika R. Doshi
Shri Girish S. Brahmbhatt
Shri Saket A. Dave
Shri Ankita S. Dwivedi
Shri Bimal K. Raval
Shri Mayuri I. Shah
Shri R. L. Gilatar
Shri Rajnikant P. Pandya
Shri Pankaj R. Shukla
Shri Nishita N. Gandhi
Shri Rajshree N. Padia
Shri Dimple U. Shukla

Co-ordinator

Shri Ashish H. Borisagar
(Academic Secretary)

Preparation and Planning

Dr. Kamlesh N. Parmar
(Dy. Director : Academic)

Lay-out and Planning

Shri Manish H. Badheka
(Dy. Director : Production)

Distribution-Planning

Shri Harshad H. Chaudhari
(Deputy Director : Administration-Distribution)

PREFACE

The Gujarat State Secondary and Higher Secondary Education Board has prepared new syllabi based on the open source operating system and compatible open source software tools for various topics of Computer Studies. These syllabi are sanctioned by the Government of Gujarat.

It is a matter of pleasure for the Gujarat State Board of School Textbooks to place this textbook of **Computer Studies** before the students of **Standard 11** prepared according to the new syllabus.

Before publishing the textbook, its manuscript has been fully reviewed by experts and teachers teaching at this level. Carrying out suggestions given by teachers and experts, we have made necessary changes in the manuscript and then have published the textbook.

The board has taken special care to ensure that this textbook is interesting, useful and free from errors. However, we welcome suggestions to enhance the quality of the textbook.

Vinaygiri Gosai

Director
Date : 3-1-2025

Mukesh Kumar (IAS)

Executive President
Gandhinagar

First Edition : 2013, Reprint : 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025

Published by : Vinaygiri Gosai, Director, on behalf of Gujarat State Board of School Textbooks, 'Vidhyayan', Sector 10-A, Gandhinagar.

Printed by :

FUNDAMENTAL DUTIES

It shall be the duty of every citizen of India :*

- (a) to abide by the Constitution and respect its ideals and institutions, the National Flag and the National Anthem;
- (b) to cherish and follow the noble ideals which inspired our national struggle for freedom;
- (c) to uphold and protect the sovereignty, unity and integrity of India;
- (d) to defend the country and render national service when called upon to do so;
- (e) to promote harmony and the spirit of common brotherhood amongst all the people of India transcending religious, linguistic and regional or sectional diversities; to renounce practices derogatory to the dignity of women;
- (f) to value and preserve the rich heritage of our composite culture;
- (g) to protect and improve the natural environment including forests, lakes, rivers and wild life, and to have compassion for living creatures;
- (h) to develop the scientific temper, humanism and the spirit of inquiry and reform;
- (i) to safeguard public property and to abjure violence;
- (j) to strive towards excellence in all spheres of individual and collective activity so that the nation constantly rises to higher levels of endeavour and achievement;
- (k) to provide opportunities for education by the parent or the guardian to his child or ward between age of 6 to 14 years, as the case may be.

* Constitution of India : Section 51-A

CONTENTS

1. Introduction to Multimedia	1
2. Animation Tool : Synfig	20
3. Creating Animation Using Synfig	38
4. Introduction to Layers	57
5. Using Pictures in Synfig	73
6. Basic Ubuntu Linux Commands	99
7. Vim Editor and Basic Scripting	137
8. Advanced Scripting	158
9. Introduction to Database Management System	184
10. Working with Tables	208
11. Retrieving Data Using Queries	233
12. Forms and Reports	263
13. Current Trends and Technologies	286
● Appendix	313



About This Textbook...

Dear Teachers,

With a mission to spread computer literacy on a fast track, the Gujarat Government has provided latest computer equipments to more than 6000 aided schools under the ICT@School program. As a new policy initiative all the schools are given the Ubuntu (a variant of Linux) Operating System and other Open Source software packages so that schools can use and exchange the software freely without botheration of the licensing issues. Since earlier text books were largely based on proprietary software, there was a need to rewrite the text books based on new syllabus. This was also necessary in view of the fact that the 8th standard has been transferred to primary section. Therefore, new content has been provided for 9th to 12th standard in a phased manner based on the Open Source Operating System and compatible Open Source software tools for various topics of computer studies.

This text book for 11th standard is the third in series for the subject of ‘Computer Studies’. In this text book, the aim is to provide elementary knowledge of multimedia, introduction to animation using Open Source multimedia tool on Linux known as Synfig, discuss essential Ubuntu Linux commands for various operations, introduction to Shell scripting, data base operations on Open Office Data Base Management System tool known as Base, and an overview of certain emerging technologies which may gain ground in the days to come.

We hope the coverage will be useful to the students to get insight into computer applications and you will enjoy teaching and conducting practical using open source software tools.

Dear Students,

We assume that you are familiar with Open Source Operating System Ubuntu Linux and Open Office components. In this text book you will be learning fundamentals of multimedia, basic Linux commands for file and general operations, working with editors and Shell scripting for elementary administrative tasks. You will also learn how to create database, write queries for information retrieval and generate forms and reports. Finally, few emerging technologies and trends will be discussed so that you would become aware of the information technology which is likely to take in close future.

The chapters 1 to 5 cover introduction to multimedia, features of multimedia tool Synfig, creating animation using Synfig, advanced concepts of animation using layers and what can be achieved by superimposing layers, how to handle pictures and images in Synfig tool. Chapters 6 introduces basic Linux commands for general purpose operations, file and directory manipulations, i/o redirection and pipe as well as for sorting of records. Chapter 7 covers the Vim editor for creating and editing files followed by introduction to shell scripting. In chapter 8 several examples of Shell scripting have been presented so that you can develop confidence and write scripts for elementary administrative tasks.

Chapter 9 and 10 introduce the Open Office tool Base for creating database (tables) and populating the created tables with actual data. Chapters 11 and 12 discuss writing queries for data retrieval and creation of forms as well as reports for output presentation. The last chapter 13 deals with few emerging technologies and trends such as Mobile Computing, Cloud Computing, RFID, Biometrics, and Green Computing among others. We are sure these will make you reading and learning of current and future trends in computing interesting.

It is sure that if you carefully study the text and practice the laboratory exercises, you will develop reasonable confidence in working with Synfig animation tool, writing Shell scripts, creating data base and generating forms, reports and queries for information retrieval.



Introduction to Multimedia



Communication is an integral part of our life. We use various means of communication like radio, newspaper, television, theatre, movies, internet and others. All these medium help us gain knowledge or educate us. A single medium of information is just not enough to convey meaningful messages to variety of audiences. For example while talking on telephone to a friend, we can hear his voice but cannot see his facial expressions. When you write letter to a friend describing him about your trip to Kashmir, only the text can be read. You cannot hear the voice of the person. If you send him a picture along with the letter, he can imagine the fun you had during your trip. However, if you send a video clip, he could visualize more about the fun you had. As you can see, the more mediums of information you use the impact of the communication increases. Multimedia is all about adding various effects to make communication effective. In this chapter we will learn about multimedia, basic elements of multimedia and applications of multimedia.

Multimedia

The term multimedia has been coined from two terms : multiple and media. Hence multimedia means usage of multiple medias to communicate. In other words, multimedia is a combination of different media elements like text, audio, graphics, video and animation. The combination of all the media elements makes the communication more structured and understandable to the user. Multimedia is now-a-days used in every field like education, theatre, advertising, fashion, gaming to name a few. Figure 1.1 shows different components of multimedia.

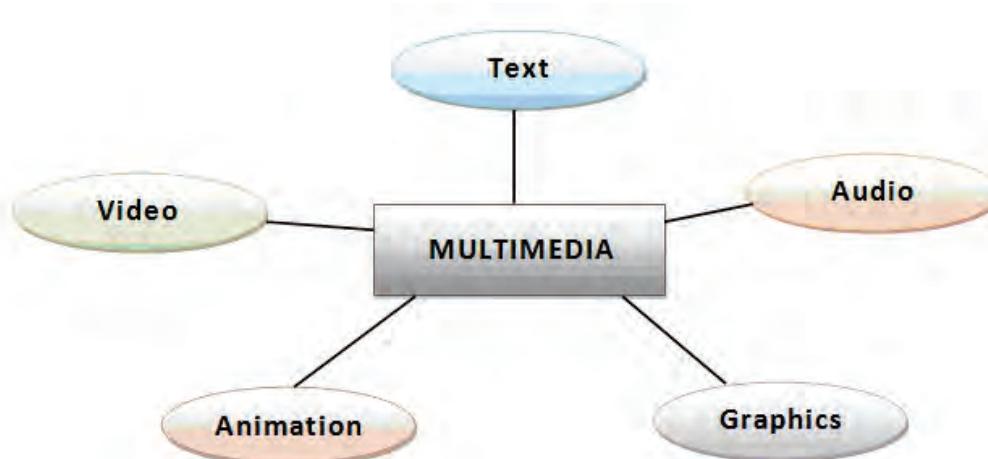


Figure 1.1 : Components of Multimedia

Computers play a vital role in the development of these media elements. A computer capable of handling text, graphics, audio, video and animation is sometimes called multimedia computer. Figure 1.2 shows use of the various elements of multimedia.



Figure 1.2 : Use of elements of Multimedia

Elements of multimedia

We defined multimedia as combination of various media elements like text, audio, graphics, video and animation. Let us now describe each element.

Text

Including text in multimedia is the basic step towards development of multimedia presentation. Text is used to communicate information to the user. Proper use of text and words in multimedia presentation will help to communicate the idea and message to the user. Text can be of any type, a word, a single line, or a paragraph. The textual data for multimedia can be developed using any text editor. However to give special effects we need graphics software. We can even use word processing software like OpenOffice Writer or MS-Word to create textual data. The text can have different type, size, color and style to suit the professional requirement of the multimedia software.

Observe that the sample shown in figure 1.3 does not look the same. This is because they have a different typeface, font and size.

GUJARAT
GUJARAT
GUJARAT

Figure 1.3 : Sample Text

A typeface is family of characters or letters which have similar look. For example : Times, Arial, Courier and others. Typefaces have different style and size known as Font. Styles are Italic, Bold, Underline and others. Size is the distance between the top of the letter to the bottom point. It is expressed in points for example 12 point, 20 point and so on. For example, if we have a font of 'Times new roman 14 point', then Times new roman is style and 14 point is size.

Typeface is categorized into : Serif and Sans Serif. Serif is a little decoration at the end of the character. Times, Century, Bookman are some examples of serif fonts. Serif fonts are easy to read on a printed page as they guide the eye along the line of text. So they are usually used when there is a lot of text to be read. On the other hand, as the name suggests, Sans Serif (sans in French means ‘without’) are without the decoration. Arial, Verdana and Helvetica are some examples of Sans Serif. They are usually used for headlines and titles. But on computer screen due to low resolution, serif fonts are difficult to read than the sans serif. Figure 1.4 shows the sample of typeface.

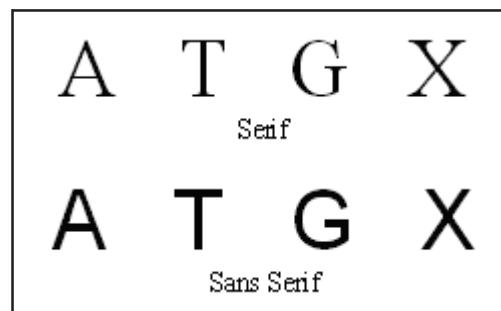


Figure 1.4 : Sample Typeface

Text in multimedia is different from the text used in traditional pen and paper method. In traditional method we convey the information using only text while in multimedia we have other elements along with text. And these elements work together to convey meaningful information. Thus we can say that the information is in the form of text, sound, pictures etc. It should be presented to the user in such a way that he/she is not confused. Thus the text in multimedia should be short, to the point and attractive to get the attention of the user.

Assume that, we want to make a presentation on “Gujarat Tourism” using OpenOffice Impress. First, we need to create a main page with the title “Gujarat Tourism”. We can simply type the text but to make it more attractive we can use Fontwork. The steps to create the text are as mentioned :

- Open a blank presentation in OpenOffice Impress.
- Click on View → Toolbars → Drawing. This will open the drawing toolbox.
- On the *Drawing* toolbar click on the *Fontwork Gallery* icon  A *Fonwork Gallery* dialog box as shown in the figure 1.5 will open.

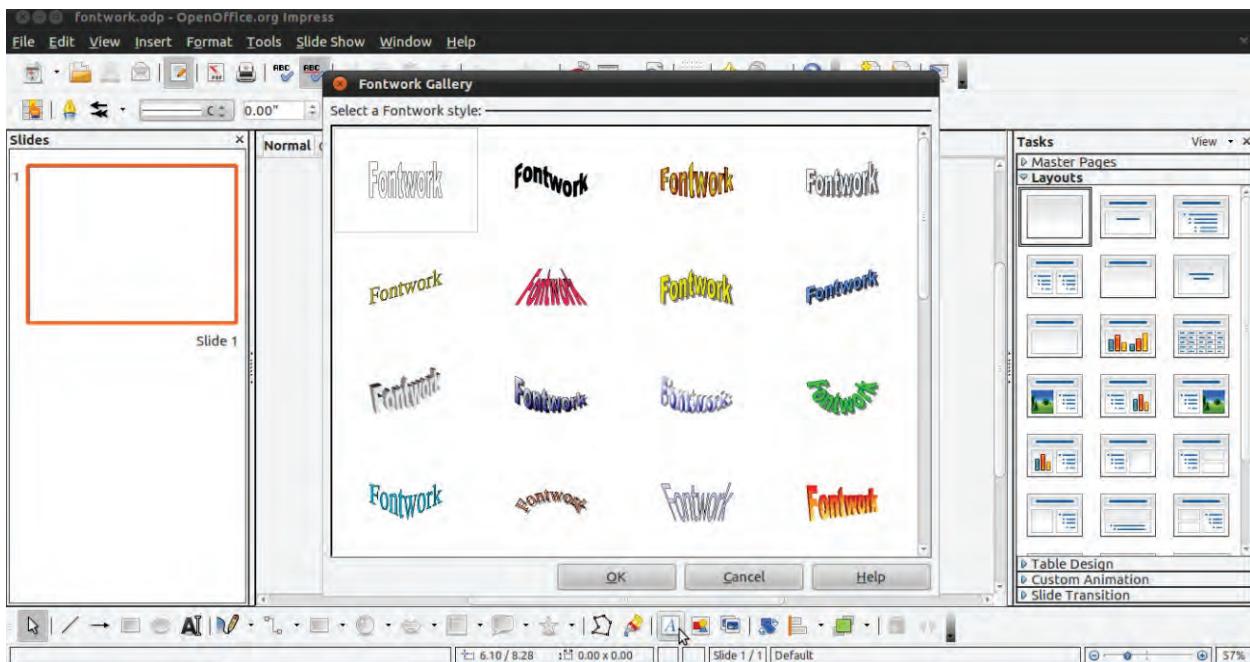


Figure 1.5 : Fontwork Gallery dialog box

- In the *Fontwork Gallery* dialog box, select any style that you like and click on the OK button. The Fontwork object is inserted into the document as shown in figure 1.6.
- Double click on the object to enter into the text edit mode.

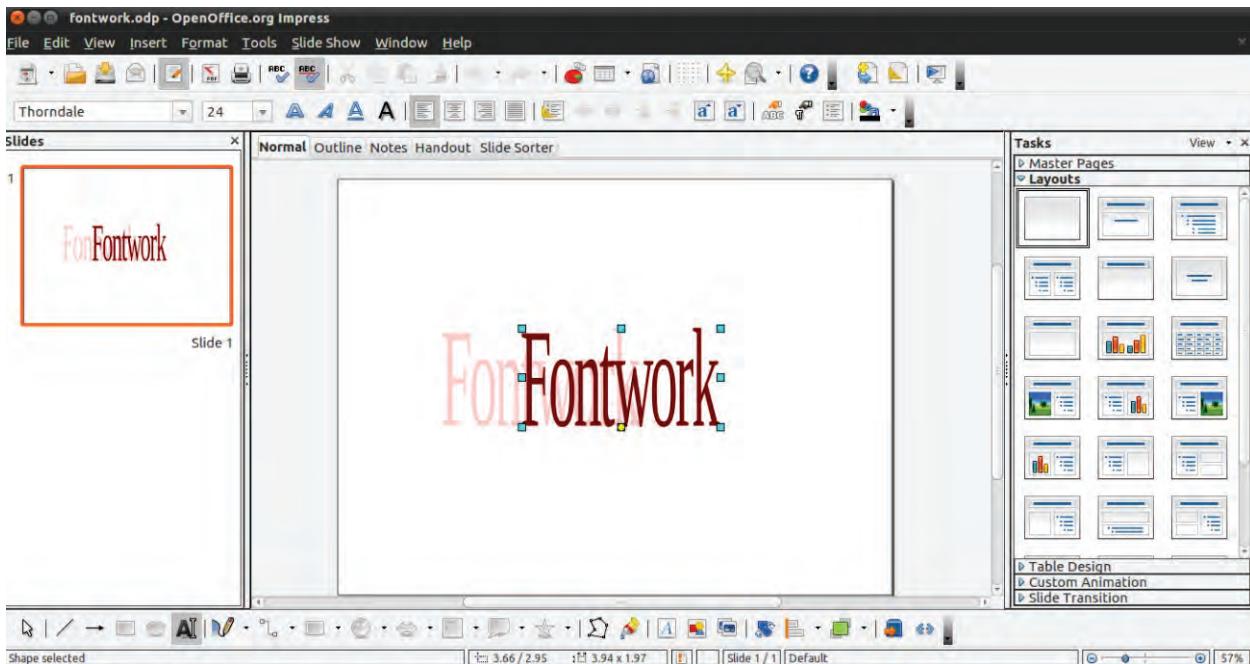


Figure 1.6 : Edit the fontwork text dialog box

- Replace the default “Fontwork” text with text “Gujarat Tourism”.
- Press ESC key to exit the text edit mode.

- You can now see that the text “Gujarat Tourism” appears on the slide as shown in the figure 1.7.
- Save the file for further use.

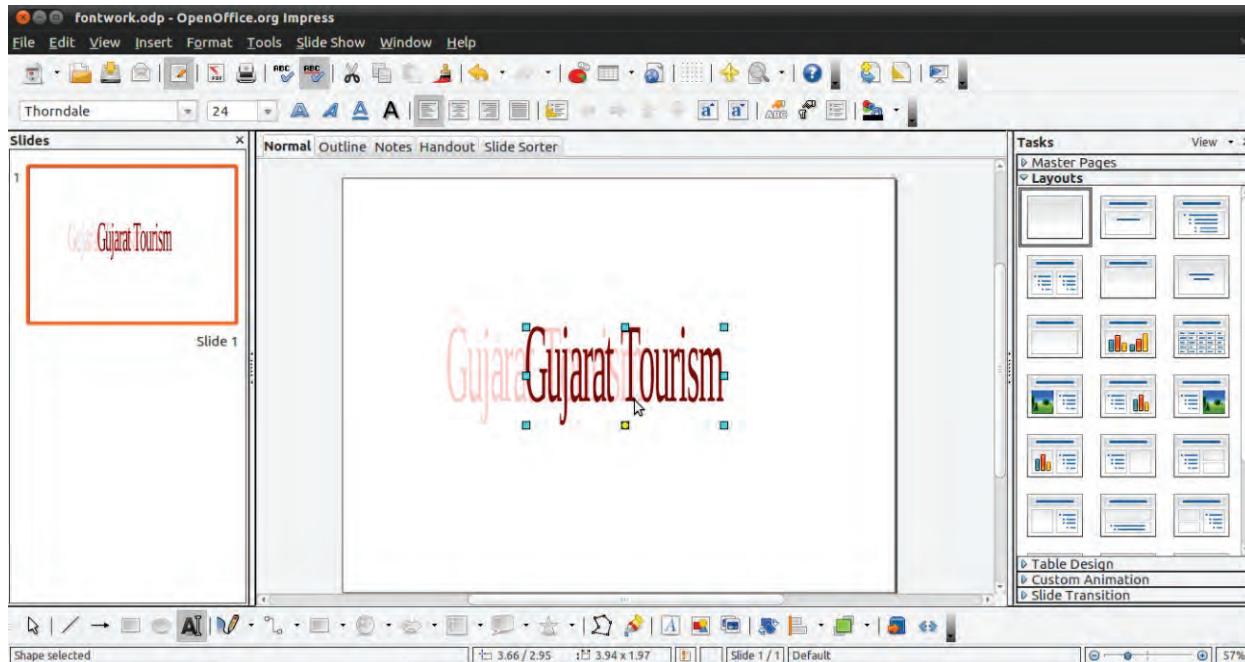


Figure 1.7 : Final slide with the text

Audio

Sound is the most important element of multimedia. We can define sound as a meaningful speech in any language. The term sound is used in the analog form, and the digitized form of the sound is called audio.

Text and images without any sound are helpful when we are presenting in front of the audience. But in the absence of the presenter, sound plays an important role. The best example is when we are viewing an educational CD. The user can enable or disable the sound option. But the learning process is more enjoyable when it is played with sound. To make such a kind of multimedia presentation, we need to first record the voice and play it according to the animation in the presentation. Thus when a user sees the presentation he/she is able to understand it easily by seeing the animation along with the voice.

Now-a-days, computers support multimedia applications. They come along with required hardware and software needed for multimedia applications. We get inbuilt sound card, speaker, microphone, and web camera which are required for multimedia applications. The operating system provides application that can be directly used for capturing audio and video.

Let us have a look at an audio recording application provided in Ubuntu Linux. Be sure you have microphone and speaker connected to the computer. Let us record our voice using the application by following the steps given :

- Select Applications → Sound and video → Sound recorder. This opens a *Sound Recorder* application as shown in figure 1.8.
- Choose File → New.
- Use the Record as drop-down list to select one of the following recording options :
 - CD quality, Lossless • CD quality, Lossy • Voice

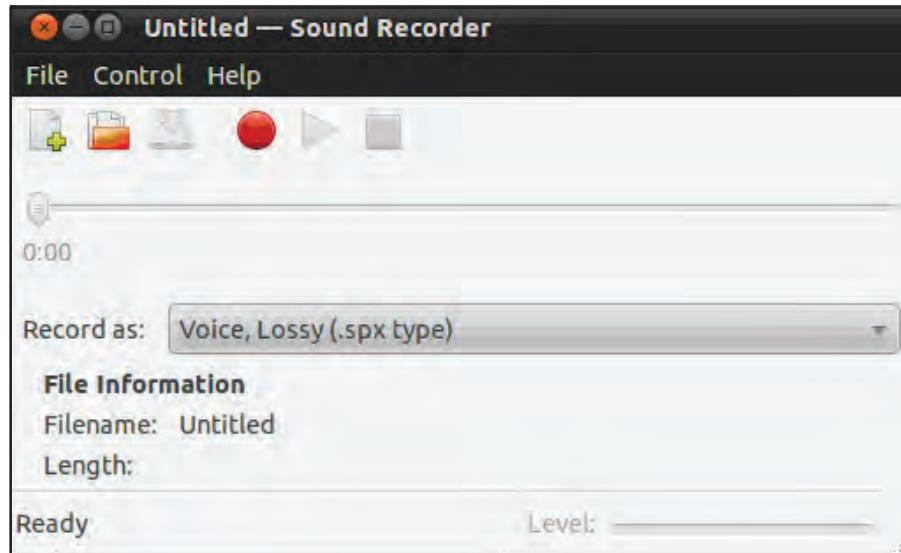


Figure 1.8 : Sound Recorder

- To start recording, choose Control → Record or click on record button  as in shown figure 1.8.
- Now start recording the voice. For example, describe a tourist place of Gujarat for our multimedia presentation on “Gujarat Tourism”. Figure 1.9 shows the audio recording process.

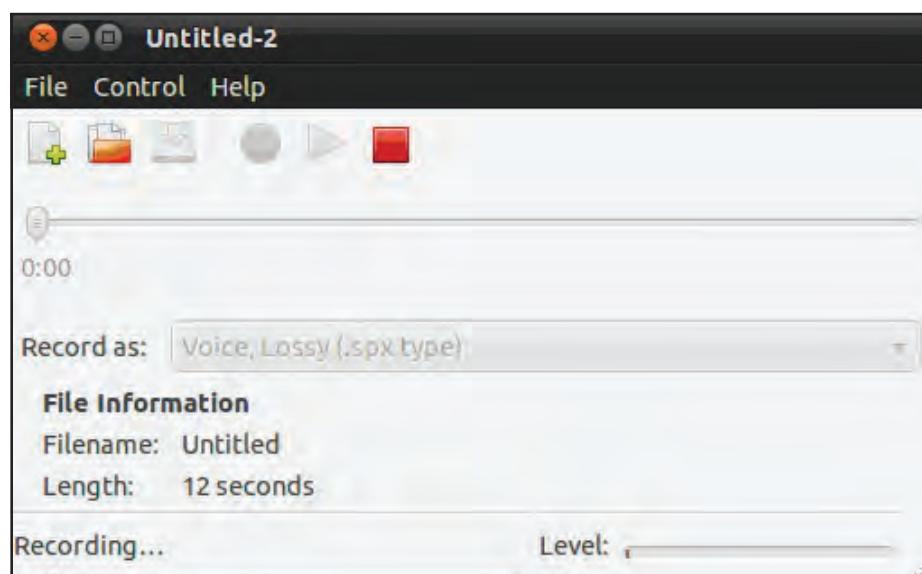


Figure 1.9 : Audio Recording

- To stop recording, choose Control → Stop or click on Stop button ().
- To play back the recording, choose Control → Play or click on Play button ().
- To run the audio mixer, choose File → Open Volume Control.
- To save the recording, choose File → Save As, and type a name for the sound file.

We can also make changes to this sound file like, delete a part of it, change the playback speed, change the playback volume and add echo etc. Multimedia elements like sounds or videos are stored in media files. To discover the type of media file we should look at the file extension. The commonly used audio formats are described in table 1.1.

File Extension	Type of File	Description
.mid, .midi	MIDI File	MIDI (Musical instrument digital interface) files contain music data.
.rm, .ram	Real Audio File	.ram (real audio metadata) is combination of audio and video.
.wav	Wave File	Waveform audio file.
.wma	Windows Media Audio File	Audio file compressed with windows media compression.
.mp3, .mpga	MP3 Audio File	Compressed audio format.

Table 1.1 : Commonly used Audio File Extension

.wav is the most popular uncompressed sound format on the Internet, and it is supported by all popular browsers. The .mp3 is the new compressed format for recorded music. So if you want to use the recorded music, .mp3 is the best choice.

Image

Generally multimedia presentations are graphics/image based. Information communicated through images is easier to remember and understand. Images used in a multimedia application can be photographs converted into digital form with the help of scanners or generated on the computer. Computers store the pictures in the form of pixels maps also known as bitmap or raster images. A Pixel (short for picture element) is the small dot on the screen. A map is a two dimensional matrix of these dots. Thus, a bitmap is a simple matrix of tiny dots which forms an image on the computer screen. Each pixel contains values representing a color. When an image is sent to be displayed on the screen, the picture data is converted to pixels.

Another form of image generated by computers is vector graphics. They store the images in the form of mathematical equations. On the screen both bitmap and vector graphics looks the same. But when you enlarge the bitmap image, it will blur and the pixels will get enlarged as shown in figure 1.10. The vector images on the other hand looks the same when enlarged. This is because,

the data is stored in the mathematical form and the values are recalculated when displayed on the screen.



Bitmap image



Enlarged bitmap image showing pixels

Figure 1.10 : Bitmap Image

Both the types of images are stored in different file formats as mentioned in table 1.2. Typically, these files are compressed to save memory space. Bitmap image file formats like gif, jpeg and png use compression within the file itself. The commonly used image formats are described in table 1.2.

File Extension	Type of File	Description
.bmp	Bitmap Image	Uncompressed image file used to store bitmap digital images.
.gif	Graphical Interchange Format File	Common for web graphics with small images and images with text. Uses limited number of colors.
.png	Portable Network Graphic	Used to store graphics for web images and supports upto 32-bit colors. It was an improvement over gif.
.jpeg / .jpg	Joint Photographic Expert Group Image File	Common image format used by digital cameras. Supports upto 24-bit colors.
.psd	Photoshop Document	Image file created by Adobe Photoshop.
.tif	Tagged Image File Format	Highly flexible and platform independent format which is widely used today.

Table 1.2 : Commonly used Image File Extension

Using inbuilt images is a common practice while preparing a presentation. For example, in Open office Writer, we can insert a picture by going to Insert → Picture. The pictures used can be a

file already present in the computer or we can scan the picture using a scanner and convert it into digital format to be used further by the computer. Figure 1.11 shows a document with an image inserted using OpenOffice Writer.

The image used in figure 1.11 was already available to us. We may use any other image of our liking. We can also make changes to our images by using programs like Adobe Photoshop, GIMP, MS-Paint and others.

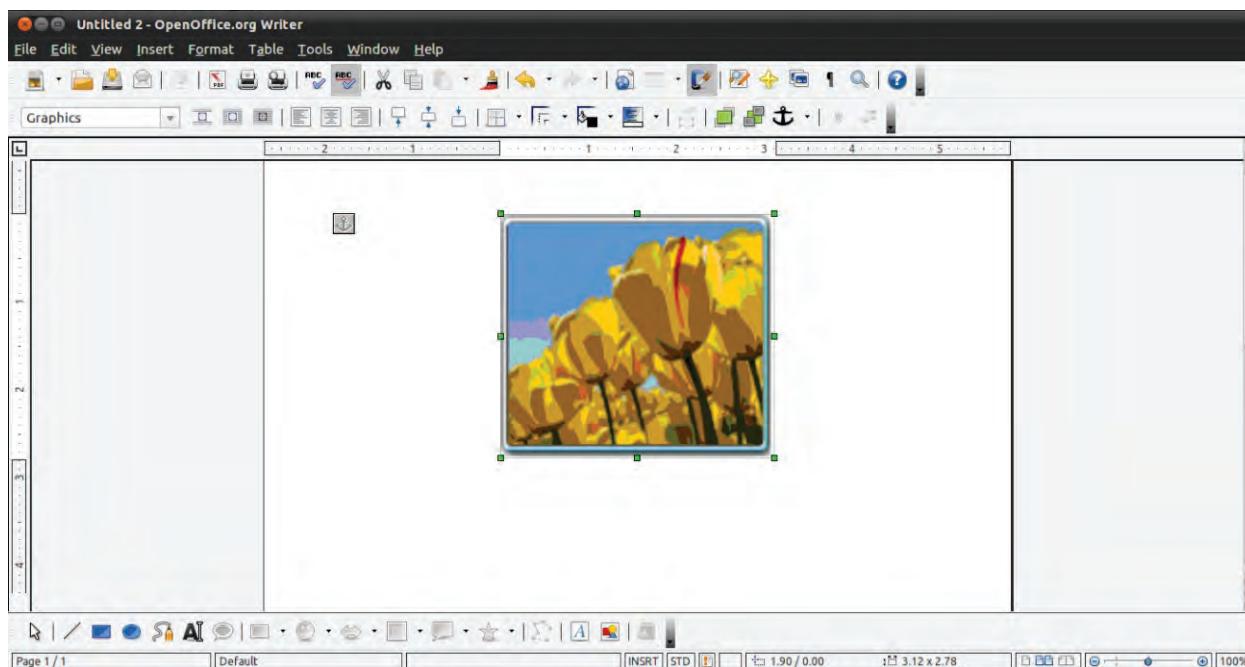


Figure 1.11 : Document with Image

Video and Animation

Animation can add great power to any multimedia application. Any static presentation becomes lively by adding a video or animation. Let us first differentiate between animation and video. The term video refers to the sequence of natural scenes captured using analog or digital video capturing device. This device can be a web camera, digital camera or even mobile phones.

Animation is a visual change over time. The digital images are played one after the other to create a moving effect. We can say that, animation is created from drawn pictures and video is created using real time visuals. Carefully planned and well-executed video clips or animations can make a great difference in a multimedia application. Some examples of animation are movies like Kung Fu Panda and Smurfs.

Animations can be classified as two-dimensional (2D) or three dimensional (3D). In 2D animation the visual change occurs on the x and y axis of the screen. This type of animation is simple. Some examples of 2D animation software are Macromedia Flash, Synfig studio and Pencil. In 3D animation the visual change occurs along three axis namely x, y and z. This type of animation gives almost

a realistic view of the image as can be seen by the human eye. Some examples of 3D animation software are Maya, Blender and 3D Max. Figure 1.12 shows the view of an object in 2D animation software and figure 1.13 shows the view of an object in 3D animation software. We will learn how to create animation in later chapters.

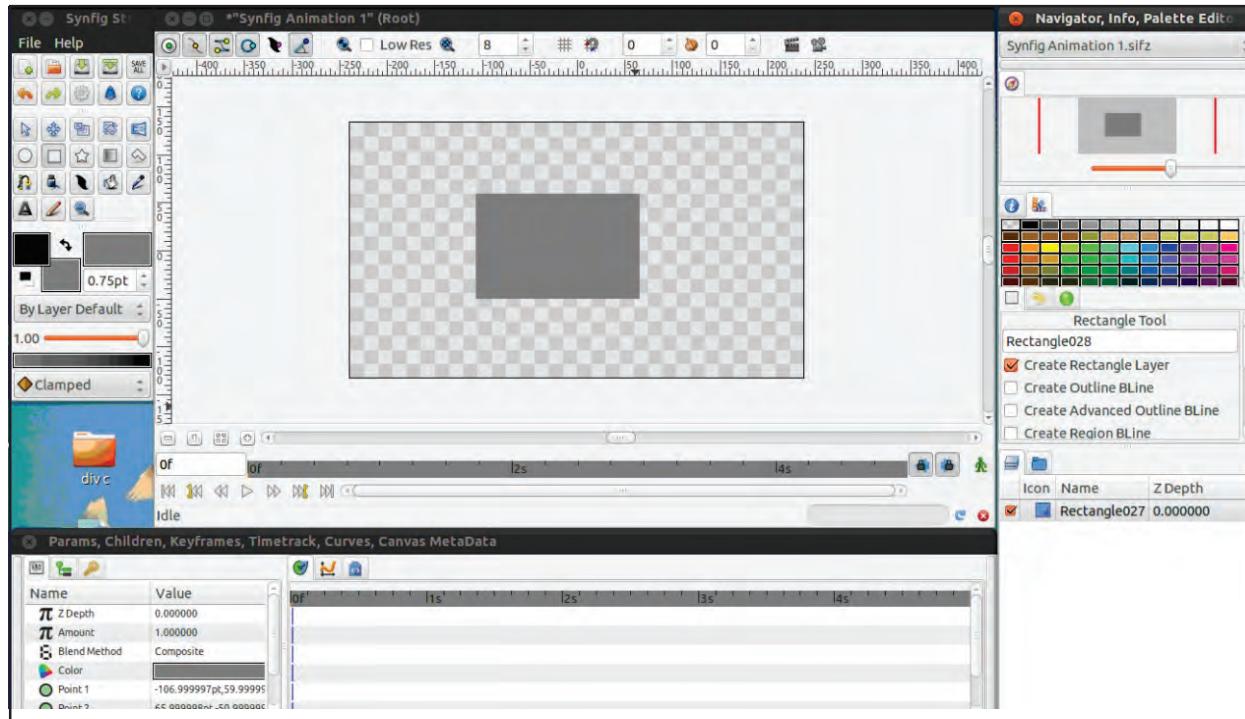


Figure 1.12 : 2D view of an object

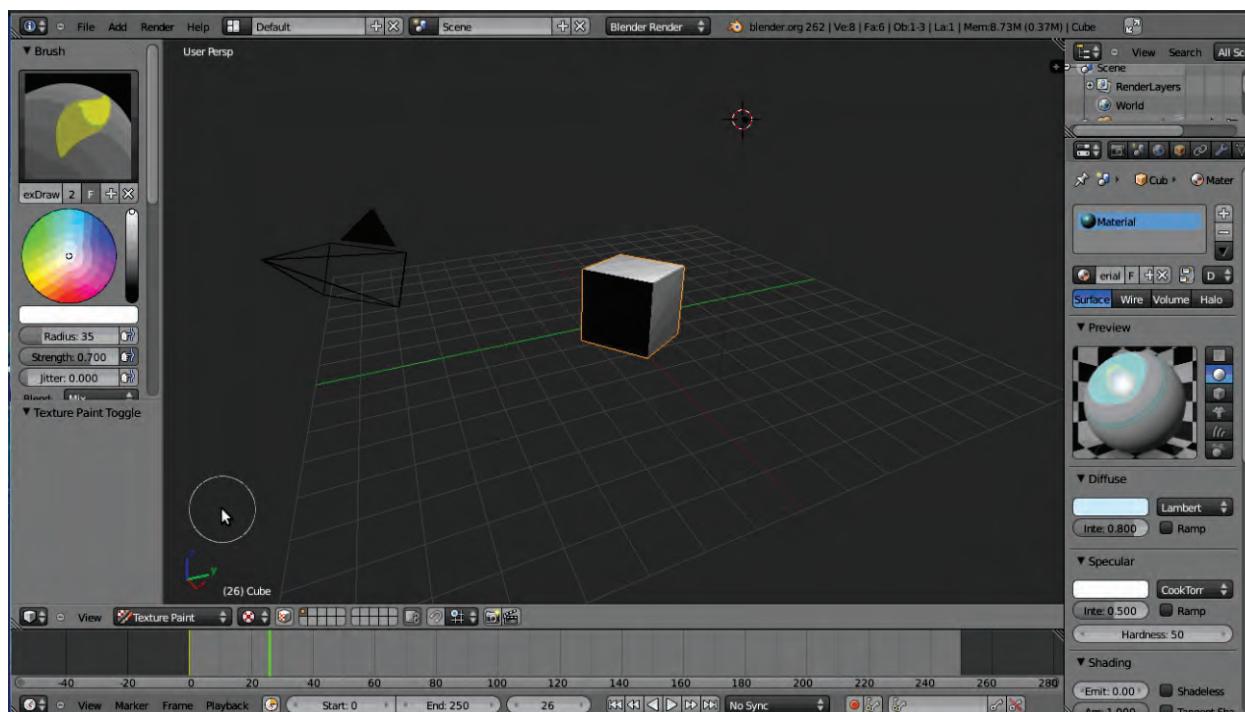


Figure 1.13 : 3D view of an object

Let us now see how to record a video using VLC media player. It is essential that you have VLC Media Player installed in your computer, if not then download it from Ubuntu Software Center. It is a free open source multimedia player. Be sure you have webcam connected to the computer. Let us start recording a video by following the steps given :

- Open VLC media player.
- Select Media → Open Capturing Device. This will take you to the Open Media dialog as shown in figure 1.14. Select the video device name and audio device name. Figure 1.14 shows the selected options.

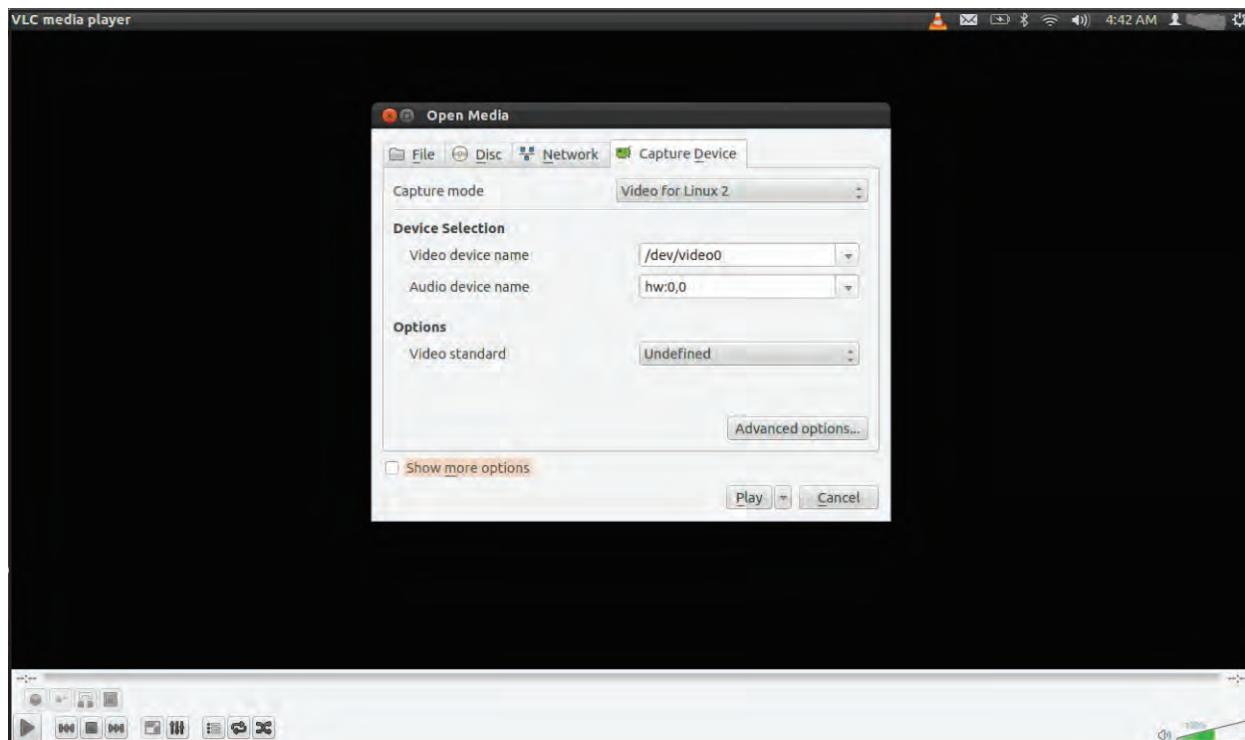


Figure 1.14 : Open Media dialog box

Note :

The device name and the audio device name may vary as per the settings of reader's machine. In case you have multiple devices you can choose the one that you would like to use.

- From the Play drop down menu visible in the *Open Media* dialog box of figure 1.14, select Convert option. This will open the *Convert* dialog box as shown in figure 1.15. In the textbox after *Destination file* label, select the destination and filename. Check 'Display the output' checkbox to see what you are recording.
- Click on the Start button.

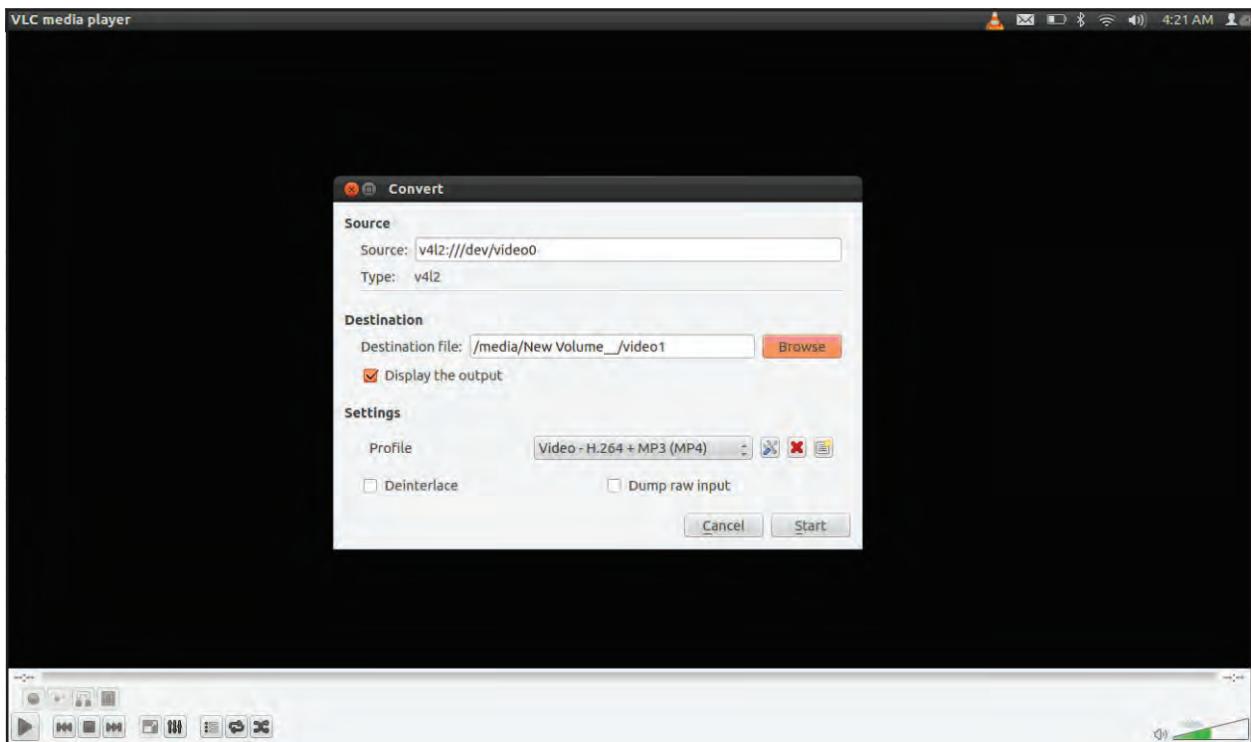


Figure 1.15 : Save the Video file

Figure 1.16 shows the video recording in progress. Click on the Stop button to stop the recording.

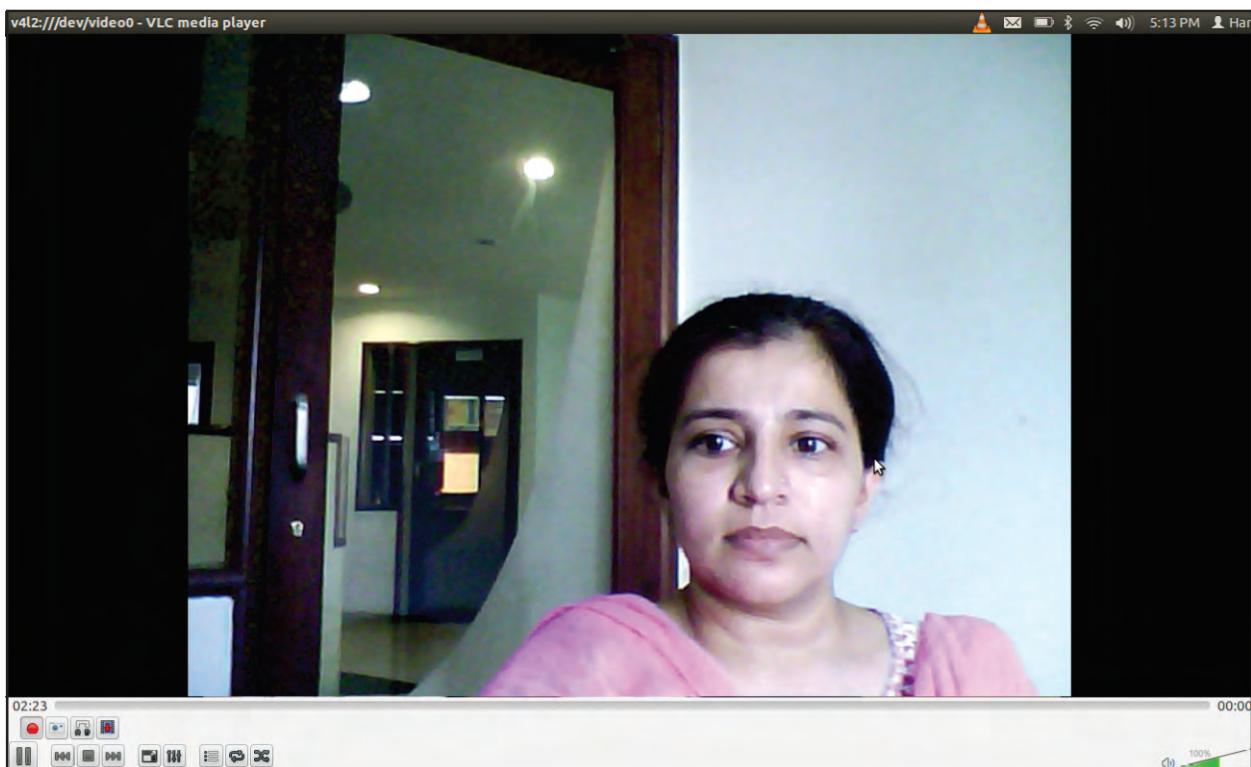


Figure 1.16 : Video recording in progress

Later the recorded video can be attached to the multimedia application. You can also try out another open source video recording software called "Cheese Webcam Booth" available in Ubuntu Software Center. The commonly used video formats are shown in table 1.3.

File Extension	Type of File	Description
.avi	Audio Video Interleave File	Developed by Microsoft to play videos in windows environment.
.wmv	Windows Media Format File	.wmv (windows media video) is a compressed video format developed by Microsoft for internet streaming applications.
.mpg, .mpeg	Moving Pictures Expert Group File	Popular video format used for creating movies distributed over internet.
.mov	Quick Time Format File	Developed by Apple. Files are compressed and are easier to download from internet.
.rm, .ram	Real Video Format File	.ram (real audio metadata) are combination of audio and video.
.swf, .flv	Adobe Flash Shockwave File	Animated file containing text and images. Created by using Adobe Flash software.
.mp4	MPEG4 Video File	Commonly used for sharing video files on the internet.

Table 1.3 : Commonly used Video File Extension

Today mp4 is most widely used format for the Internet video. It is supported by Flash players, YouTube as well as HTML5.

Interactive objects

For a multimedia presentation to be effective it should have maximum interactivity. At times the user may want to skip a portion of the presentation or want to again go through that same portion. The user is given buttons or hyperlinks for navigation in such cases. When these interactive objects are clicked, the flow of the presentation changes as per the user's choice.

Earlier the Internet browsers supported only one media element : text. Text was also limited to single font and color. All the media elements described earlier are now supported by latest Internet browsers in different ways. Some media elements can be handled inline and some require plug-in. Inline media objects are specified inside the program using a specific <inlinemediaobject> tag. Plug-in are extra helper programs that can be easily installed and used as a part of the web browser. The extent to which a particular media element is employed is determined by the nature

and scope of the project. For example, multimedia project on “Gujarat Tourism” may use more images, video clips and text about various places of Gujarat. We will learn more about these elements in later chapters.

Classification of Multimedia

Generally multimedia is classified into two namely : Interactive and Non-Interactive multimedia.

In Interactive multimedia, the sequence and timing of media elements can be controlled by the user. For example, a multimedia CD on “Learn Musical Instruments”, the user can select on the different musical instruments he/she wants to play. User can also select his favourite song which he wants to play using that instrument. Most of the CD-ROM titles and games available in the market are interactive in nature. Generally, interactivity increases the user participation and he/she enjoys it more. The degree of interactivity depends upon the field of the multimedia project.

In traditional mass media like television, radio or newspaper the communication is one-way. In other words we can say that the process originates from the source and is delivered to the mass audience. For example, a movie has a beginning and an end irrespective of the user watching it on the television or not. Though these technologies also use video, audio, text and graphics but in an inflexible way. While in multimedia the user does not remain passive but can control the elements. Thus, the difference between mass media and multimedia is that there is a shift from audience to user and the communication becomes two-way instead of one-way.

In Non-Interactive multimedia the user simply watches the media as it plays from beginning to the end. He/she has no control over the flow. For example : a corporate presentation or a multimedia demo.

Another classification of multimedia is based on its applications : Entertainment or Edutainment.

All types of games, movies are examples of Entertainment multimedia wherein the purpose is to entertain the user. Whereas the multimedia titles that educate the user are classified under edutainment multimedia. All types of educational CD's are example of edutainment multimedia.

Usage of Multimedia

Multimedia has become an integral part of our life. Multimedia finds its application in various areas including, but not limited to, advertisements, art, education, entertainment, engineering, medicine, mathematics, business and scientific research. Let us discuss some of the usage of multimedia in various fields :

Education and Training

In the area of education, multimedia has a great importance. We have been using presentations consisting of charts, tables and other objects to impart knowledge since a long time. But now-a-days the classroom education is not limited to the earlier conventional method rather it needs audio and visual media. Multimedia is used to produce computer based training courses commonly known as CBT. A CBT lets the user go through a series of presentations, animations, text about a particular

topic along with examples. Today CBT are used for almost all the age groups right from kindergarten to post graduates. For example, edutainment CD like Nursery Rhymes or Learn Computers. Edutainment is an informal term used to describe combining education with entertainment, especially multimedia entertainment.

Multimedia is used extensively in training programs. Medical trainers and doctors can practice surgery methods via simulation before the actual surgery. Mechanics can learn to repair the engines or a salesperson can learn about the product details.

Some of the uses of multimedia in education and training are as mentioned below :

- Recording or broadcasting lectures.
- Using video conferencing we can hear an expert speaker from a distant location.
- Demonstrating surgeries or other techniques that learners may not otherwise have the opportunity to see and later put them in practice.
- Record student's performance to enable feedback.

Advertisements

In the field of advertising multimedia plays a vital role. Today, advertising is the major source of launching and promoting a product by the manufacturing companies in the market. Exciting animations, effects and slogans can make an advertisement popular. This can lead to promotion of the product. We see so many advertisements today and some of them catch our attention and we ultimately end up buying that product. For example, Jujus and pug dog in Vodafone advertisement were used to promote the various plans and packages of the company.

Entertainment

Multimedia is heavily used in the entertainment industry to develop special effects in animations and movies. Movies like Ice age, Jurassic park, Avatar will always be remembered for their special effects and animations.

Multimedia games are very popular among children and a variety of these games are available either as CD-ROM's or online. With availability of lot of gaming software programs for individuals or groups, virtual gaming has become a reality today. With usage of such software two players in different countries can play a game sitting on their computers.

Journalism

Multimedia is used a lot in the field of journalism. There are many magazines and newspaper that are published periodically. Today not only we see the text in the newspaper, but can also see the photographs. E-newspaper and E-magazines are also available online where we can see the videos related to particular news.

Stages in Multimedia Production

Any multimedia presentation must be carefully planned and designed. As we studied that multimedia is applicable to almost all the areas, a plan should be according to the field selected. For example,

an application made for entertainment may not work with an educational application and vice versa. Irrespective of the field, the stages in multimedia production remain the same. Let us discuss the stages in multimedia production.

Research and Analysis

During this stage we need to find out about the audience, their skills, needs and qualification. We should also gather as much information as possible about the content to be presented.

Scripting (or Flowcharting)

Scripting or flowcharting means deciding the flow of the multimedia project. This is done by making a flowchart to show the main menu and the branching when a user selects a particular option. For example, if we were designing educational multimedia project on Science for Standard XI, the home page would display all the chapter names. Selecting a particular chapter would display the explanation, exercises, test and other aspects related to the chapter. Thus we can draw a flowchart starting with our main menu and then the subpages that are linked with each other.

Note : The term scripting here does not refer to action scripting or java scripting.

Storyboarding

During this stage the actual visualization of the project takes place. The designer decides how each screen should look like, which media elements are to be used and where to place them on the screen. The storyboard in our earlier example, will contain the design of the home page, where we will place the buttons, how will they look like, what happens when we click this button and other such required elements.

Collection of media elements and construction

After the storyboarding, the designer is ready with the prototype and starts creating the graphics and other media elements to be used in the project. For example, characters to be used are created, sounds and videos are recorded and animations are created.

Programming

After the collection and construction of media elements, they are combined together into a final product using software packages like Macromedia Flash, Synfig, Hypercard to name a few.

Testing

This is the final stage of any multimedia project. In this stage we check if all the media elements are working as per the requirements or not. Also, whether the audience targeted will find the design and the content attractive or not.

If we work as per the stages mentioned here then most of the applications that we may develop will turn out to be good.

Summary

In this chapter we discussed multimedia and its components text, audio, graphics, video and animation. Each of the components plays a vital role in the development of the multimedia project. We introduced each component and discussed it in detail. Multimedia is classified into Interactive, Non-Interactive, Entertainment and Edutainment multimedia. Today the usage of multimedia can be found in almost all the fields like education and training, advertisements, entertainment, journalism to name a few. In the systematic and fruitful development of a multimedia application the stages play an important role.

EXERCISE

(6) Which of the following terms represent the distance between the top of the letter to the bottom point?

- (a) size
- (b) style
- (c) length
- (d) width

(7) Which of the following terms represent typefaces that have different styles and sizes?

- (a) font
- (b) text
- (c) letter
- (d) character

(8) Macromedia Flash, Synfig studio and Pencil are examples of which of the following animation types?

- (a) 3D
- (b) 2D
- (c) 4D
- (d) 5D

(9) Which of the following terms represent a small dot on the computer screen?

- (a) pixel
- (b) point
- (c) cursor
- (d) cell

(10) Which of the following terms represent a classification of images?

- (a) vector, raster
- (b) bitmap, raster
- (c) picture, clipart
- (d) graphics, picture

(11) Which of the following terms represent a visual change over time?

- (a) text
- (b) audio
- (c) animation
- (d) graphics

(12) Which of the following terms represent a two dimensional matrix of dots?

- (a) screen
- (b) map
- (c) cell
- (d) array

(13) Which of the following terms is full form of MIDI?

- (a) musical instrument data interface
- (b) musical instrument digital interface
- (c) musical image digital interface
- (d) musical instrument digital image

(14) Which of the following terms represent the most common image format used by digital cameras?

- (a) .gif
- (b) .jpeg
- (c) .tif
- (d) .bmp

(15) Which of the following is used to store the images are stored in form of mathematical equations?

- (a) vector
- (b) raster
- (c) bitmap
- (d) picture

(16) Which of the following devices help us to convert photographs into a digital format?

- (a) scanners
- (b) modem
- (c) printer
- (d) monitor

- (17)** Which of the following terms represent PNG?
- (a) programmable network graphic (b) photo network graphic
(c) portable new graph (d) portable network graphic
- (18)** Which of the following terms represent a helper programs that can be easily installed and used as a part of the web browser?
- (a) plug-in (b) text
(c) video (d) audio
- (19)** In which of the following terms can Multimedia be classified?
- (a) intelligent and non-intelligent (b) interactive and non-Interactive multimedia.
(c) intuitive and non-intuitive (d) informative and non-informative
- (20)** In which of the following animation types does the visual change occur in the x and y axis?
- (a) 2D (b) 3D
(c) 4D (d) 5D

LABORATORY EXERCISES

- 1.** Create a title text using OpenOffice Impress or Writer for the text “Gujarat”.
- 2.** Record your voice using sound recorder describing about “Adalaj ni Vav”.
- 3.** Record a video describing “What is Multimedia”.





Animation Tool : Synfig



In previous chapter, we discussed about multimedia and the basic building blocks of multimedia. To create a multimedia presentation using these building blocks we need application programs known as multimedia authoring tools. These authoring tools provide the important framework to organize and edit the multimedia elements like graphics, sound, video and animation.

Generally, the authoring tools are classified into:

- Card or Page based tools
- Icon and Event based tools
- Time based tools

In Card or Page based authoring tools the elements are organized as pages of a book or pile of cards. The authoring tool links these pages into an organized sequence. These types of tool are useful when the different elements can be viewed individually like the pages of a book. Some examples of card based authoring tools are Hypercard and Multimedia Toolbook.

In Icon and Event based authoring tools the elements are organized as objects in a structural framework or process. Here you need to build the flowchart of the events or tasks and then add the elements as per the structure. Some examples of icon and event based authoring tools are Authorware and IconAuthor.

In Time based authoring tools the elements are organized along a timeline. They are useful when you want to give a message that has a beginning and an end. The elements are displayed as per the time or location of the events. Time based authoring tools are the most popular amongst all. Some examples of time based authoring tools are Synfig, Pencil, Flash and Director.

There are many multimedia tools available in the market. Some of these tools are proprietary while others are open source. Some of the open source animation tools easily available on internet are Pencil and Synfig.

Pencil

Pencil is a 2D animation software that lets you create traditional hand-drawn animation using both bitmap and vector graphics. It can be freely downloaded from <http://www.pencil-animation.org>. It is available on various operating systems like Linux, Windows and MacOS X.

Figure 2.1 shows the user interface of pencil software. The interface looks quite similar to other animation or drawing softwares. In pencil animation software, you can create good animation if you are good at free hand drawing instead of object drawing.

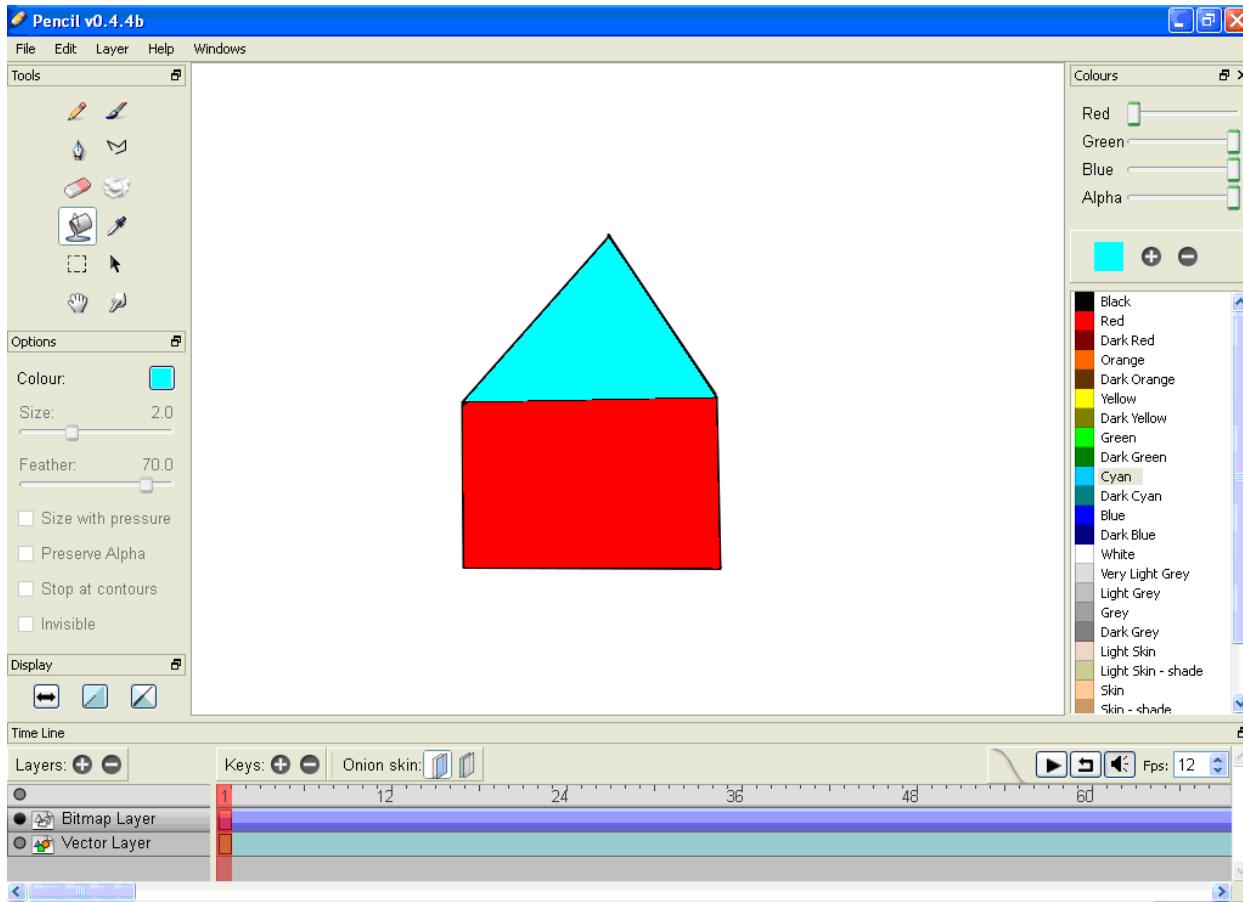


Figure 2.1 : User interface of Pencil animation software

For object drawing and animation, let us learn about an open source animation tool called Synfig Studio.

Synfig Studio

Synfig Studio is a time based multimedia authoring tool that falls in the category of open source. It is available free of cost on the internet and can be downloaded from <http://www.synfig.org>. Synfig Studio (or simply Synfig) is 2D vector animation software released in 2005 under the GNU GPL (General Public License). It is designed to produce film-quality animation with fewer people and resources. It is available on various operating systems like Linux, Windows and MacOS X. The first version of Synfig was 0.61.05 and new versions have been regularly launched with additional features till date. The latest version of Synfig is 0.63.05. Since it is not available as a standard package in Ubuntu Linux, we need to install it first. Once installed, it will be available under the Graphics option of Applications menu.

Synfig stores animations in XML (extensible markup language) file format, often compressed using compression software gzip which stands for GNU zip. These files use the filename extension .sif in uncompressed form or .sifz in compressed form.

Getting Started

To start Synfig, in the menu bar click Applications → Graphics → Synfig. When you start Synfig Studio you must have observed that it does not start in one window, but opens a number of individual

windows on the desktop. We can drag the window to change the size. If you want to reset the Synfig studio window to default window arrangement as shown in the figure 2.2 then go to Toolbox → File menu → Panels → Reset Windows to Original layout.

Let us first get familiar with the user interface of Synfig. In figure 2.2 you can see the Synfig studio window layout.

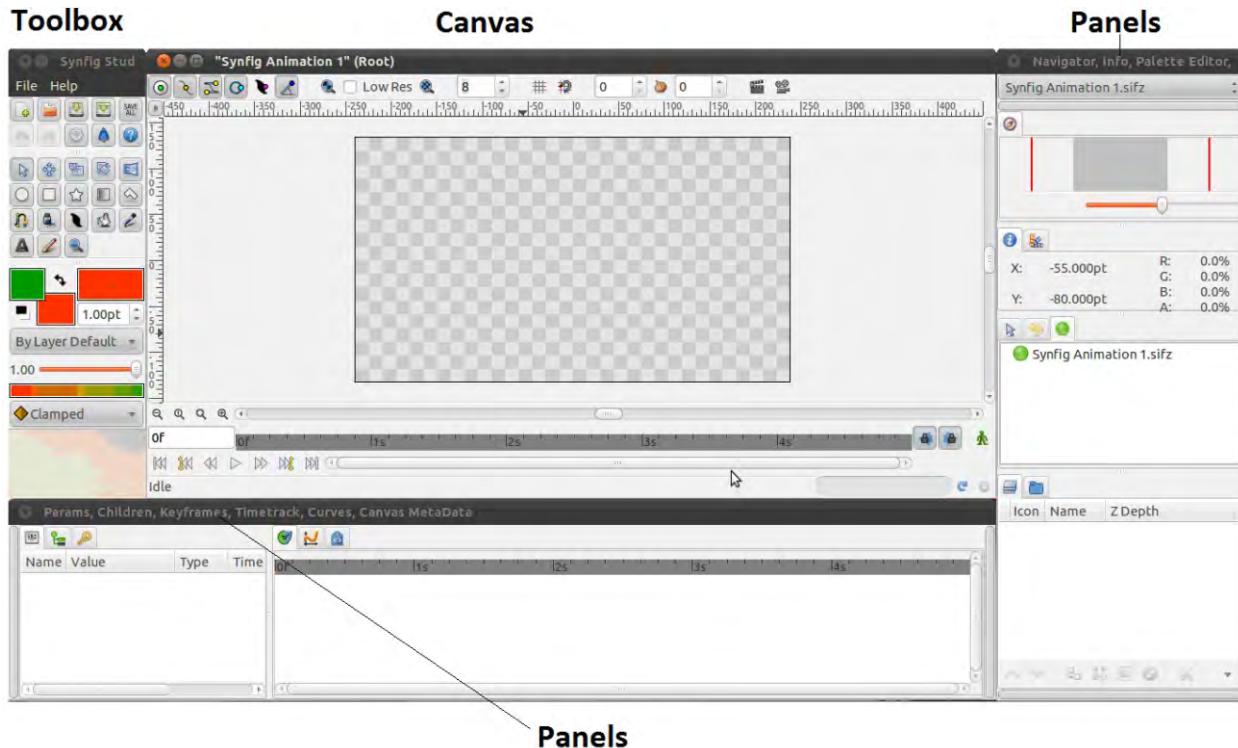


Figure 2.2 : Synfig user interface

As seen in the figure 2.2 the components in the user interface are: toolbox, canvas and panels. Let us discuss the role of each of them.

Toolbox

Toolbox is the main window which contains system menu and tools to create and edit your artwork as shown in figure 2.3. Closing the toolbox exits the application. Even if you have several projects open there will only be one Toolbox for all.

Using the toolbox we can create a new file, open an existing file, save the file and setup properties. The toolbox also contains the general tools like circle, rectangle, transform and others, for creating and editing an object. The toolbox window is separated into three areas or palette:

- The upper palette contains buttons for standard file operations like create a new file, open a file, save, save all files, undo and redo as well as access to the settings dialog and the help system.
- In the middle palette we have the tools like circle, rectangle, transform and others for creating and manipulating an object. We will learn more about each tool later in this chapter.

- The lowest palette contains the default settings for new layers like :
 - Foreground and background color :** it creates the fill and the outline colors for the object. When we create a layer without outline then it has only fill color. We can change the color by clicking on fill or outline color.
 - Brush size :** sets the size of the line for new outline layers.
 - Blend method :** sets the blend method which will be discussed in the later chapters. It is set to “By layer default”.
 - Opacity :** it controls the layers visibility. 0 means the layer is invisible and 1 means the layer is visible.
 - Gradient :** used by Gradient layers as their fill. It sets to a default gradient between the foreground color and background color.
 - Interpolation :** each waypoint has an interpolation setting which determines the manner in which the parameter changes.

Canvas

When you open a new file or create a new project you will see the canvas window. It can be seen at the center of the user interface as shown in figure 2.4. This is the place where you display your artwork and create animation. Whenever we start Synfig studio a new canvas window appears.

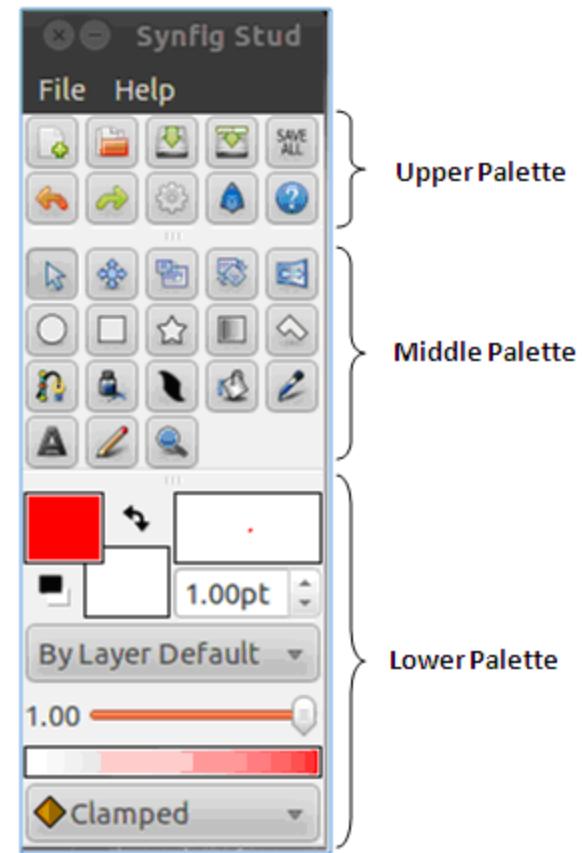


Figure 2.3 : Toolbox

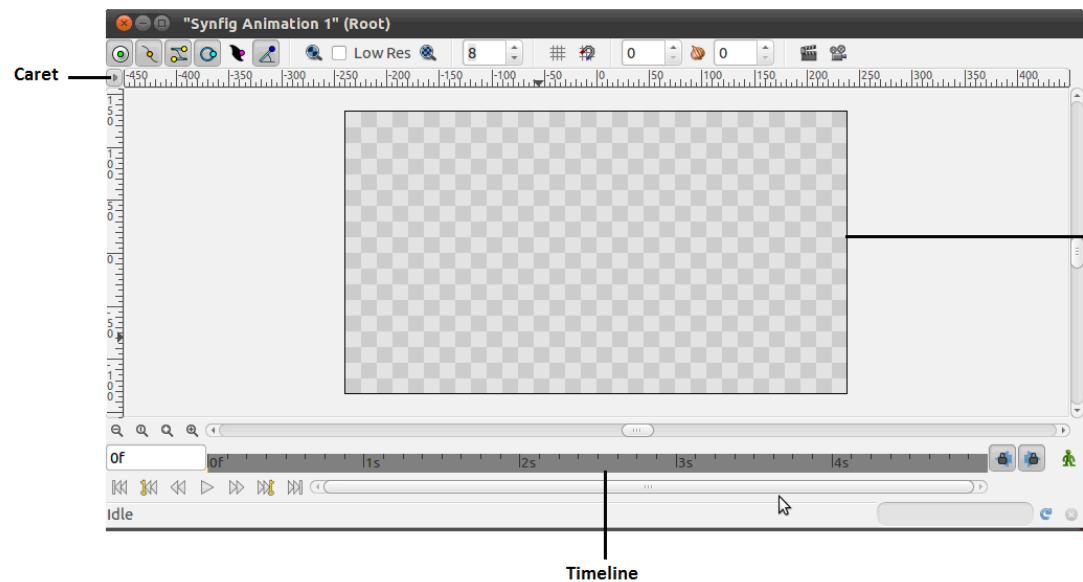


Figure 2.4 : Canvas Window

The area with the grey check-board pattern is the working area in which you can create elements/layers and manipulate them.

In the upper left corner of the canvas window you can see caret ▶, a little black triangle. Clicking on this button, the canvas window menu will pop up which allows access to most of the Synfig features. Generally, in most of the applications you find the set of menus at the top of the screen or at the top of the drawing window.

You can also see the timeline which appears only when you have non-zero duration in the settings dialog of your project. To the left you can see the number of the current frame and on the right side two buttons to switch the animation status and to lock/unlock the keyframes.

Panels

Synfig has different type of panels like layers, parameter, history etc. Panels contain tools and information about certain elements of your project. Some panels will allow you to modify these elements.

The two windows on the bottom and to the right are customizable dock dialogs. Each dock dialog contains a set of panels which are arranged horizontally or vertically. Some panels share the same space inside the dock dialog and we can switch between them by clicking on their tabs. We can rearrange the contents of dock dialogs by dragging the panel tab to where we want it. If accidentally we close a panel go to the Toolbox, select File → Panels and click on the name of the panel you need. Then drag the panel into the dock dialog where you want to place it. Figure 2.5 and 2.6 shows the panels arranged vertically and horizontally.

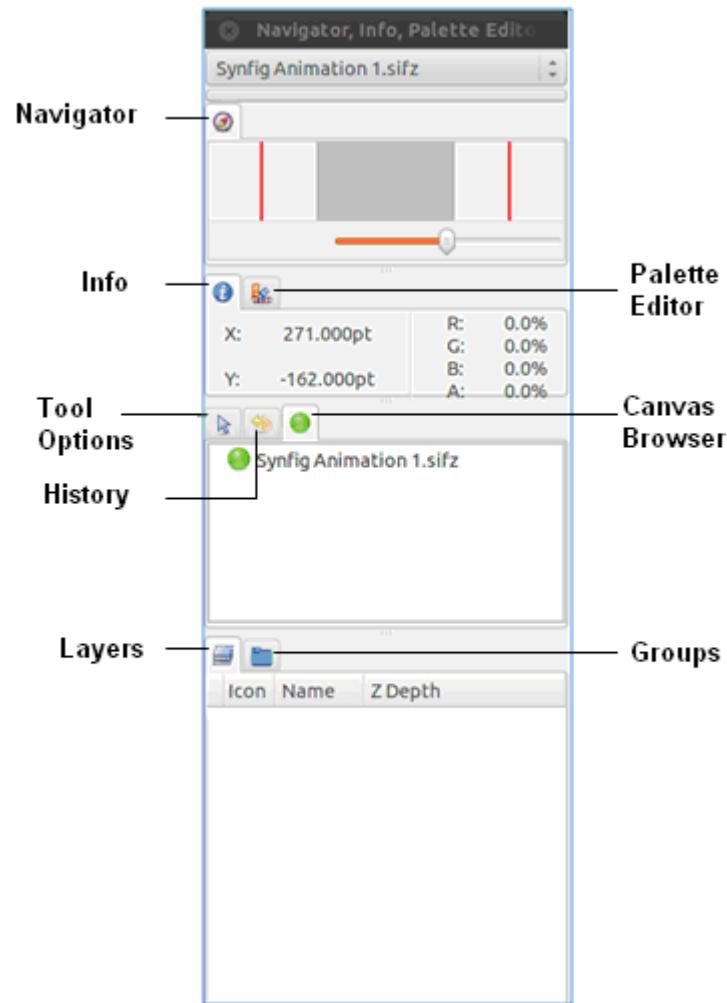


Figure 2.5 : Panels arranged vertically

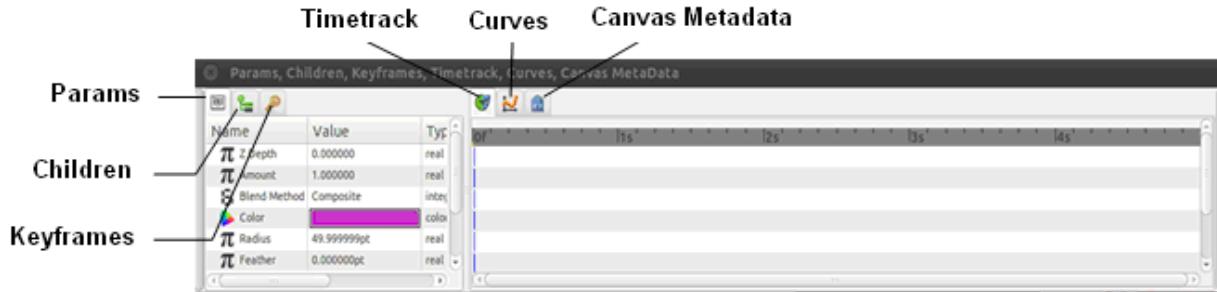


Figure 2.6 : Panels arranged horizontally

Let us discuss the function of some of the important panels:

Layers Panel

It displays the layers of your working canvas. It also allows you to manipulate these layers.

Params Panel

It shows the currently selected layer parameters. When multiple layers are selected, only the parameters that are common in the selected layers are displayed.

Tool Options Panel

It shows options that are specific to the currently selected tool.

Navigator

It shows a thumbnail image of what the currently selected canvas looks like. We can also zoom in and move the focus around.

History Panel

It keeps track of all the actions that are done while editing the file. The check box column is used to disable specific actions without going through the entire action list. So if we want to redo or undo an action or group of actions just click on the check box.

There are many other panels and to know about each simply hold mouse over its icon and a tooltip will pop up describing its function.

Creating new file

Now let us see how to create a new file and set its properties. To create a new file, perform the following steps:

- Select Toolbox → New file. Alternatively, when you start Synfig a new file is automatically created.

Click on the Caret → Edit → Properties. This will open a dialog box as shown in figure 2.7.

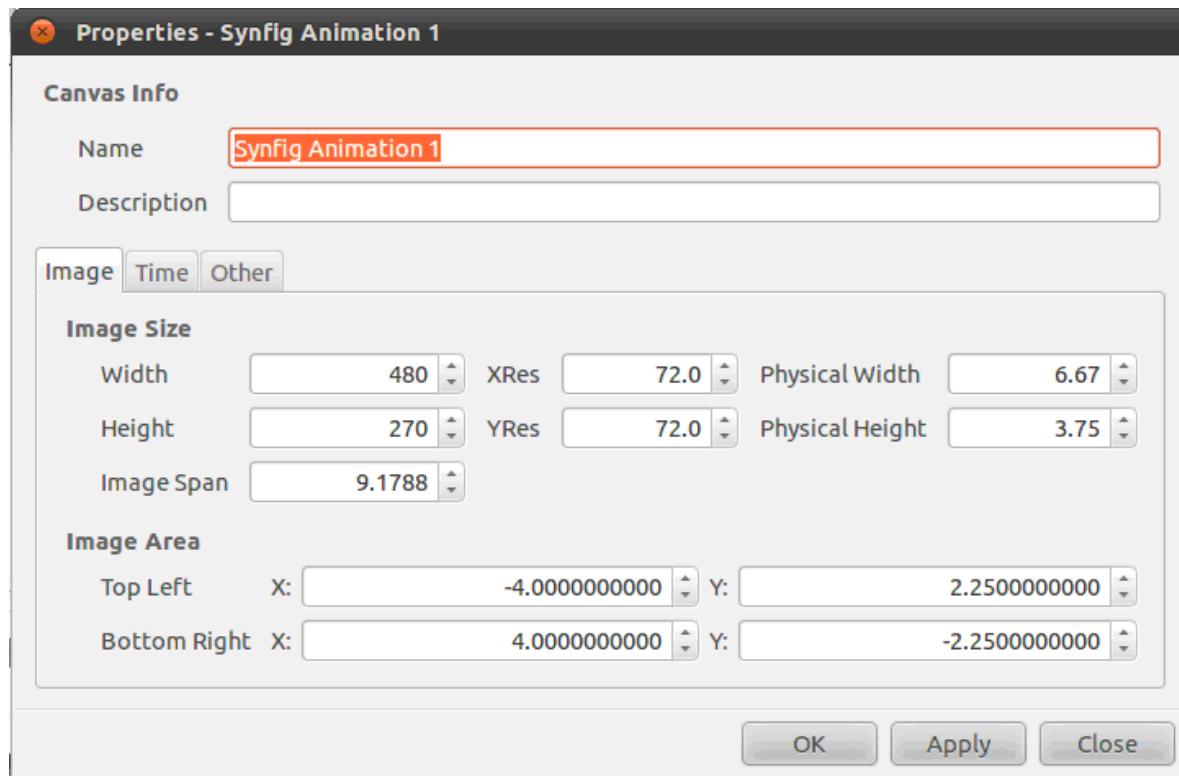


Figure 2.7 : File properties dialog box

You can set the name of the file and give a description to your animation. In the window, you can see three tabs namely *Image*, *Time* and *Other*. The *Image* tab is used to set the image size and image area as shown in figure 2.7. The *Time* tab is used for setting the start time, end time and frames per second. The default settings are as shown in figure 2.8. We will learn more about these properties later.

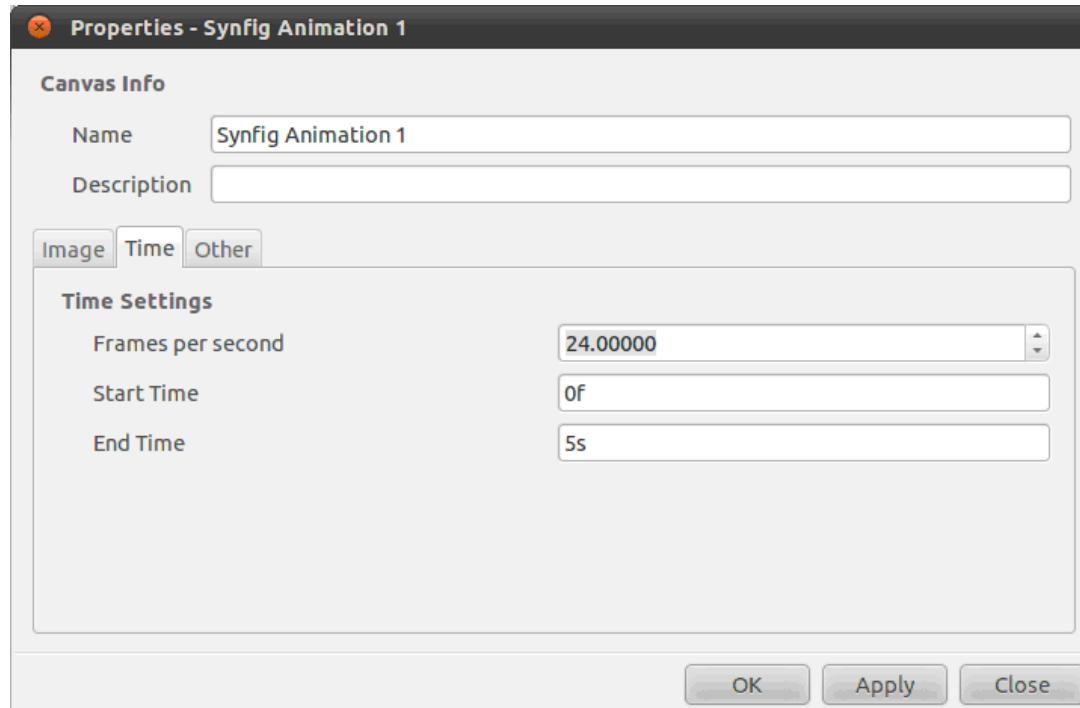


Figure 2.8 : File property dialog box with time tab selected

- Now click OK button and the changes will be applied.
- After creating the file and setting properties we can now start creating our artwork and animation.

Let us now see different tools used to create and manipulate an object.

Tools

Tools help you to create the artwork for your multimedia project. Synfig provides various tools to create and manipulate an object. Figure 2.9 shows these tools, each tool has its own set of options and parameters. Tools are visible in the tool options panel. And parameters are visible in the parameter panel. When you select a particular tool, the tool options panel displays the options that are available for that tool.

Let us first discuss the tools used to draw an object.

Circle Tool

The Circle tool is used to create new circle layer. Circle layers are used to represent circles. A circle has two important parameters: centre and radius. You can see the parameters in the parameter panel. We can change any of the parameter after drawing the object.

To draw a circle select the circle tool, Click in the working area where you want the center of the circle to be and then drag to set the radius as shown in figure 2.10.

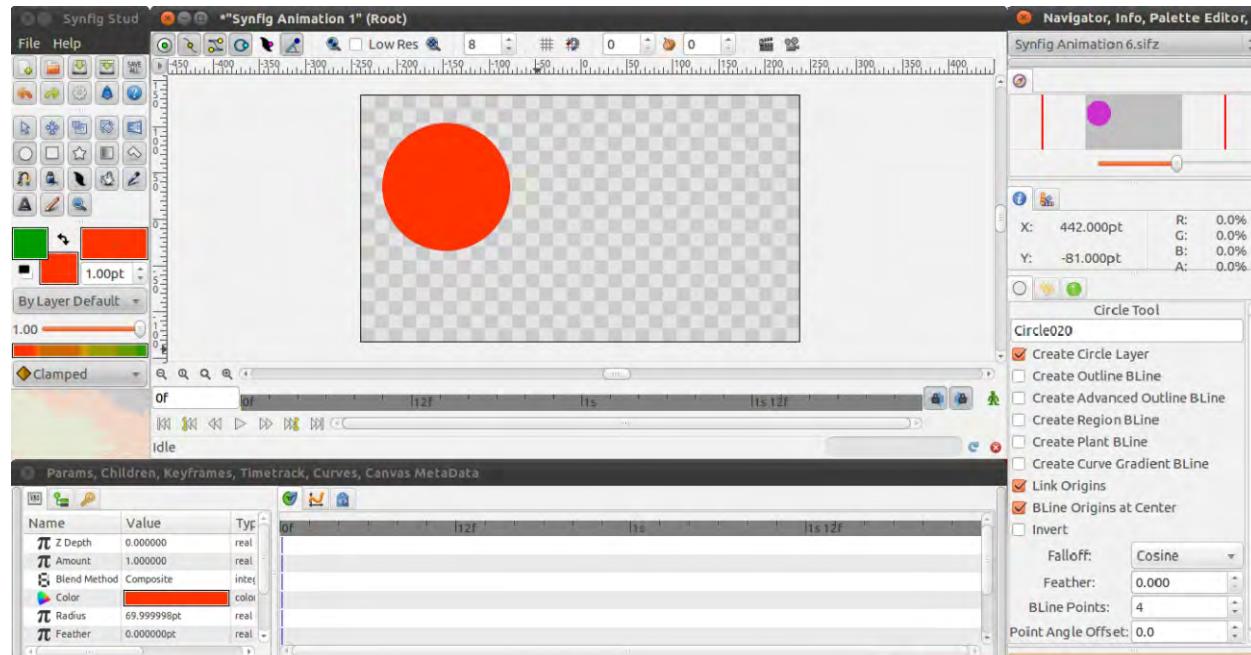


Figure 2.10 : Use of Circle tool



Figure 2.9 : Tools to create and manipulate objects

Using the tool options panel you can also change the name of the circle layer, create additional layers using the checkboxes like create outline layer which is used to create an outline around the circle. In the figure 2.10, only the *Create Circle Layer* is selected. If we select the *Create Outline Bline* option then the circle object will be created with an outline.

If you want to edit the circle, say for example you want to increase the radius of the circle or move the circle then click on the duck as shown in the figure 2.11. In case the ducks are not visible then click on the transform tool to see the ducks. You may be wondering what a duck is?

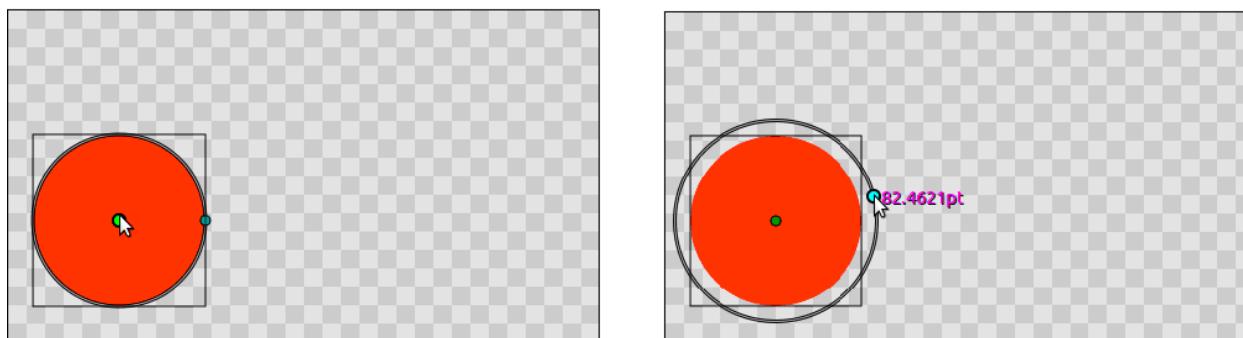


Figure 2.11 : Ducks for circle

A duck is a handle that is used to control some parameter of a layer. For example, a circle will have two ducks: one to control the position of its centre, and second to control its radius. In figure 2.11, you can see the green and blue colored ducks. Ducks come in different color and they have specific meaning. The color of the duck signifies a change in the following aspects of the object:

- Green – position of the object
- Blue – radius of the circle
- Orange – vertices
- Yellow – curves
- Dark blue – to change the angle in star

In figure 2.11 the green duck is used to change the position of the circle and the blue color duck is to change the radius of the circle. Ducks are displayed in the work area for all the currently selected layers. You have to drag the ducks to change the parameters. You can even use the cursor keys for more precise control. If you want to constrain the movement to be either horizontal or vertical then hold down the shift key while dragging.

Sometimes, one type of duck interferes or gets into the way of another duck. You can turn the ducks on/off by selecting the option present on the top of the canvas window as shown in figure 2.12.



Figure 2.12 : Turn on/off various ducks

Rectangle Tool

We can use the rectangle tool in the same way we used the circle tool. It is used to create a rectangle layer. To draw a rectangle, select the rectangle tool, click where you want any corner of the rectangle to be, and then drag to the opposite corner as shown in figure 2.13.

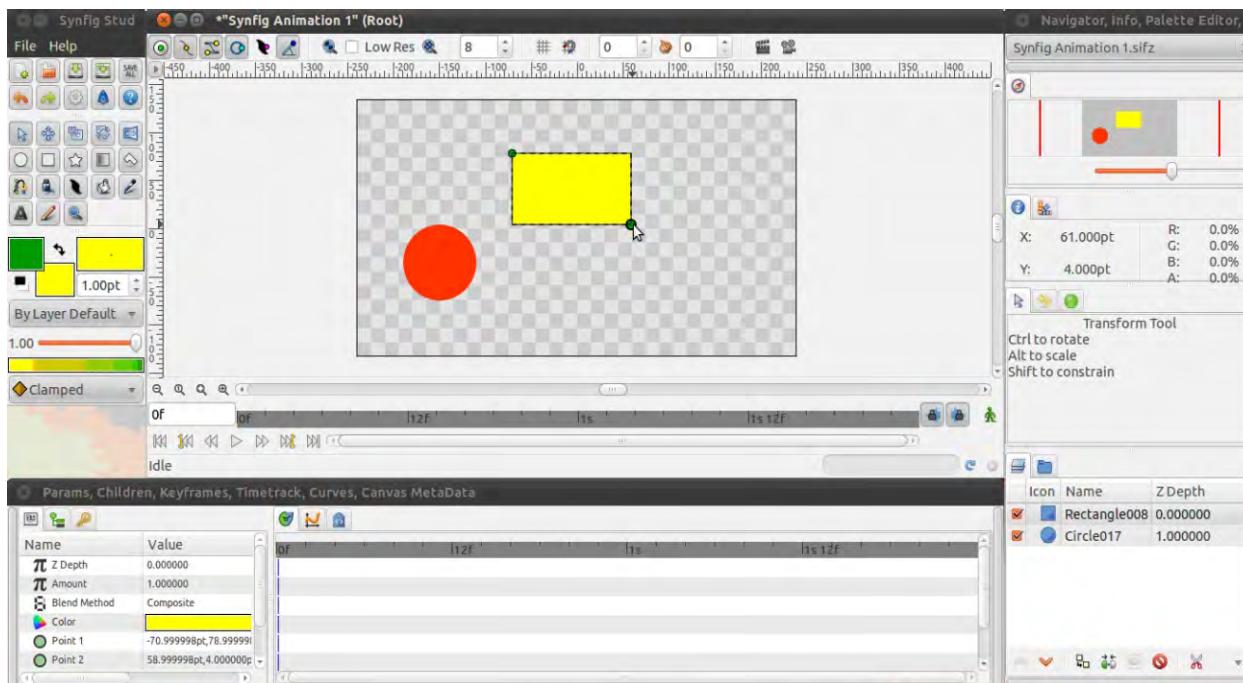


Figure 2.13 : Use of Rectangle tool

You can also see the tool options available in the tool options panel. Also the parameters are displayed in the parameter panel. As we have two objects on the working area, the parameter panel displays the parameters that are common to circle and rectangle. This way we can change a single parameter to make changes in all the objects. If we want to see the parameters of only the rectangle object; then in the layers panel we can select rectangle layer which will now display the parameters of rectangle only.

We can make the changes to the rectangle by selecting the duck and then dragging it. Figure 2.14 shows the ducks for the rectangle tool.

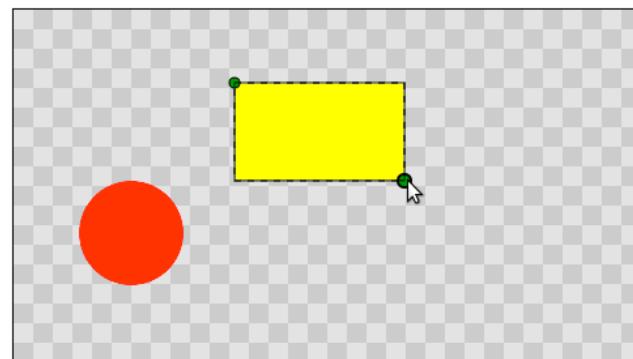


Figure 2.14 : Ducks for Rectangle tool

Star and Polygon Tool

Likewise, we can use the star and polygon tool to create the star layer and polygon layer. The only change that you can see is while editing them. The ducks for each shape is different and change accordingly. Figure 2.15 (a) shows use of star tool and ducks to edit the star. Make the changes to see the difference. Figure 2.15 (b) shows how to draw polygon with polygon tool. A polygon can be created using any number of points and then click on the first point to close it.

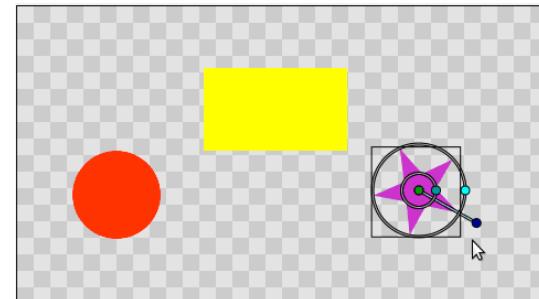


Figure 2.15(a) : Use of Star tool

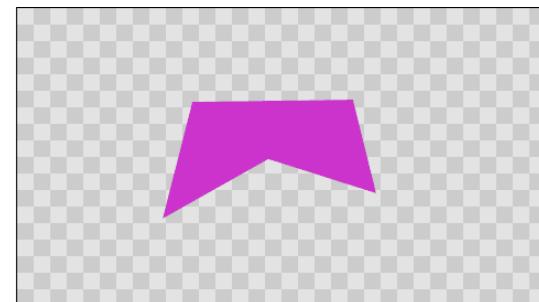
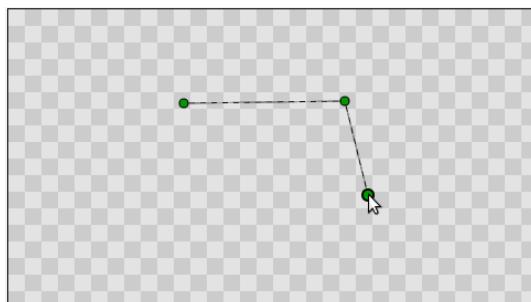


Figure 2.15(b) : Use of Polygon tool

BLine Tool

Bline tool is used to make objects with any number of points and curves. Click on the Bline tool and start inserting the points. In figure 2.16 (a) you can see a shape drawn using Bline tool. To stop drawing the object, right click on the last point. After drawing, the object can be edited using the ducks. Figure 2.16 (b) shows the ducks of the shape drawn using Bline tool.

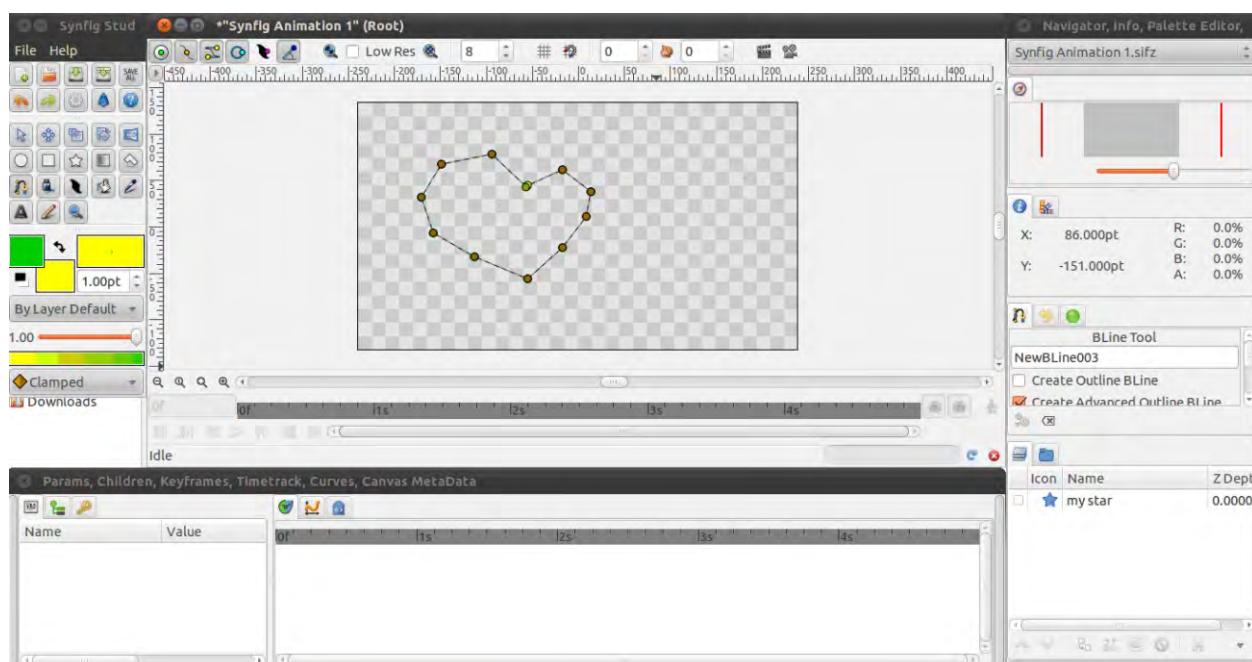


Figure 2.16(a) : BLine tool

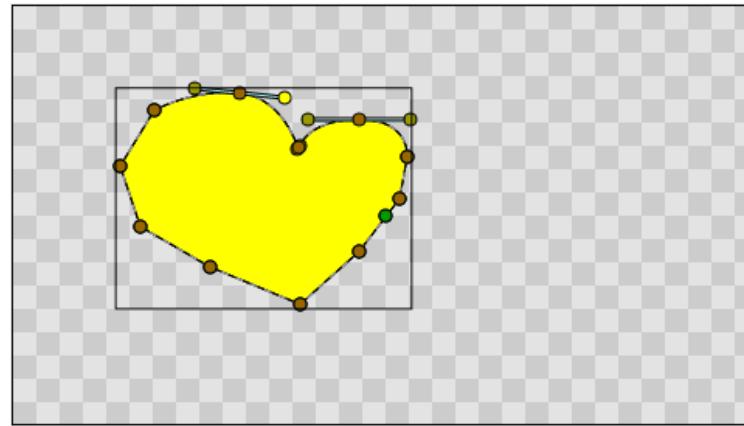


Figure 2.16(b) : Drawn shape with the ducks

After discussing about the object drawing tools, now let us discuss some other tools.

Draw Tool

You must be familiar with the draw tool as it is the most common tool used for drawing. You can draw with the help of mouse. Figure 2.17 shows the use of draw tool.

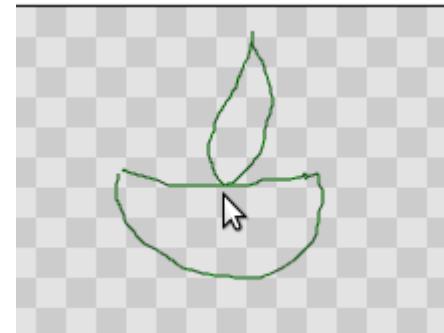


Figure 2.17: Draw tool

Fill Tool

The Fill tool changes the color of objects. To fill a color in the object, select the color from the color palette. Select the Fill tool button and then click on the object. This tool has an effect on the following layers:

- Circle Layer
- Rectangle Layer
- Polygon Layer
- Star Layer
- Region Layer
- Outline Layer
- Checkboard Layer

The other layers are not affected by this tool because they use gradient to fill the visual area.

Eyedrop Tool

The Eyedrop tool allows you to select colors from the Eyedrop tool. Select the Eyedrop tool, and then click in the work area to set the default foreground color in the Eyedrop tool to be the color you are currently pointing at.

Gradient Tool

The Gradient tool is used to create smooth transitions between two or more colors in an object. When you select the Gradient tool, the tool options panel will show the options for the Gradient tool as in figure 2.18.

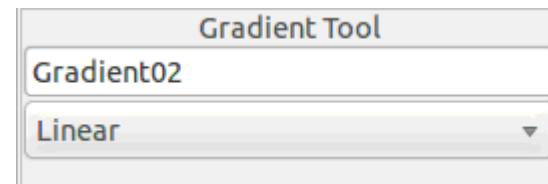


Figure 2.18 : Gradient tool options panel

It allows you to set a name for the layer and choose the type of gradient.

There are 4 types of gradients namely linear, radial, conical and spiral.

Linear - This produces a simple transition in a straight line. Set the foreground and background colors. Click where you want the gradient to begin, and drag to where you want the transition to end. The gradient will be created perpendicular to the line you drag out. You can edit the gradient by moving the endpoint in any direction using the normal tool. Figure 2.19 shows the effect of linear gradient.

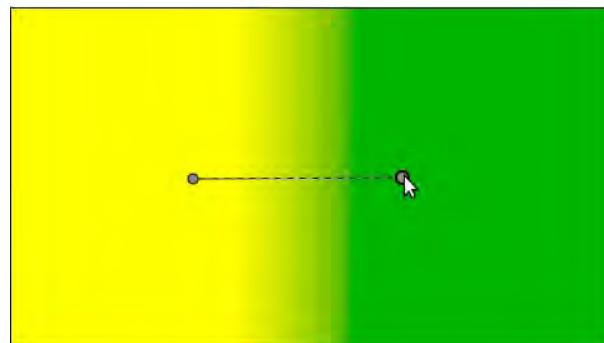


Figure 2.19 : Linear gradient

Radial - This produces circular colors with the transition being at the center of those circles. Click where you want the center of the circles to be, and drag to set the radius of the transition. Figure 2.20 shows the effect of radial gradient.

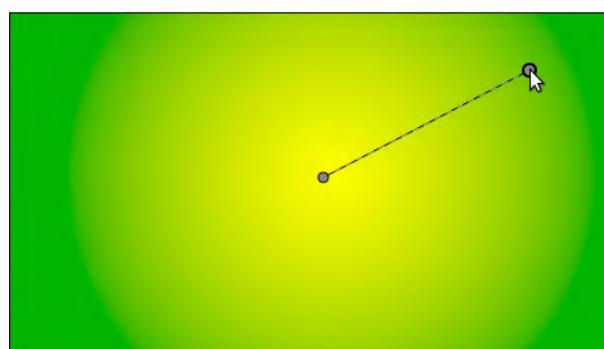


Figure 2.20 : Radial gradient

Conical - This has the appearance of looking down on a tip of a cone. The gradient is along the circular arc of the center and goes in all directions. Click to set the center, and drag to indicate the direction in which the foreground and background colors should go. The center endpoint adjusts the center of the gradient and the other endpoint adjusts the direction of the gradient. Figure 2.21 shows the effect of conical gradient.

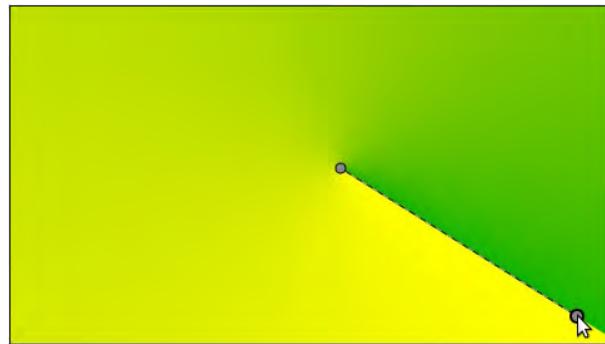


Figure 2.21 : Conical gradient

Spiral - This produces a spiral gradient. Click to set the center of the spiral and drag to set the ‘tightness’. Figure 2.22 shows the effect of spiral gradient.

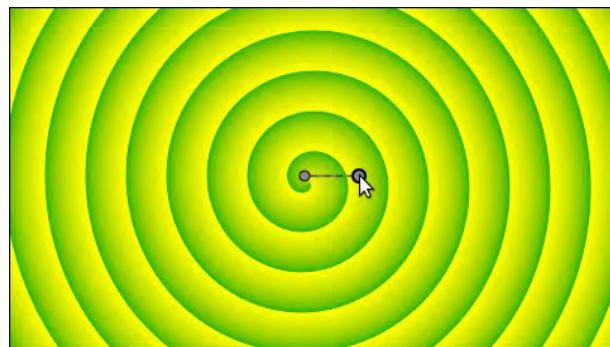


Figure 2.22 : Spiral gradient

Text tool

Text tool allows you to enter text by creating a text layer. Select the text tool icon from the toolbox, and then click in the canvas window where you want the text layer to be created. A dialog box will appear, allowing you to enter the text you want in the created layer as shown in figure 2.23 (a).

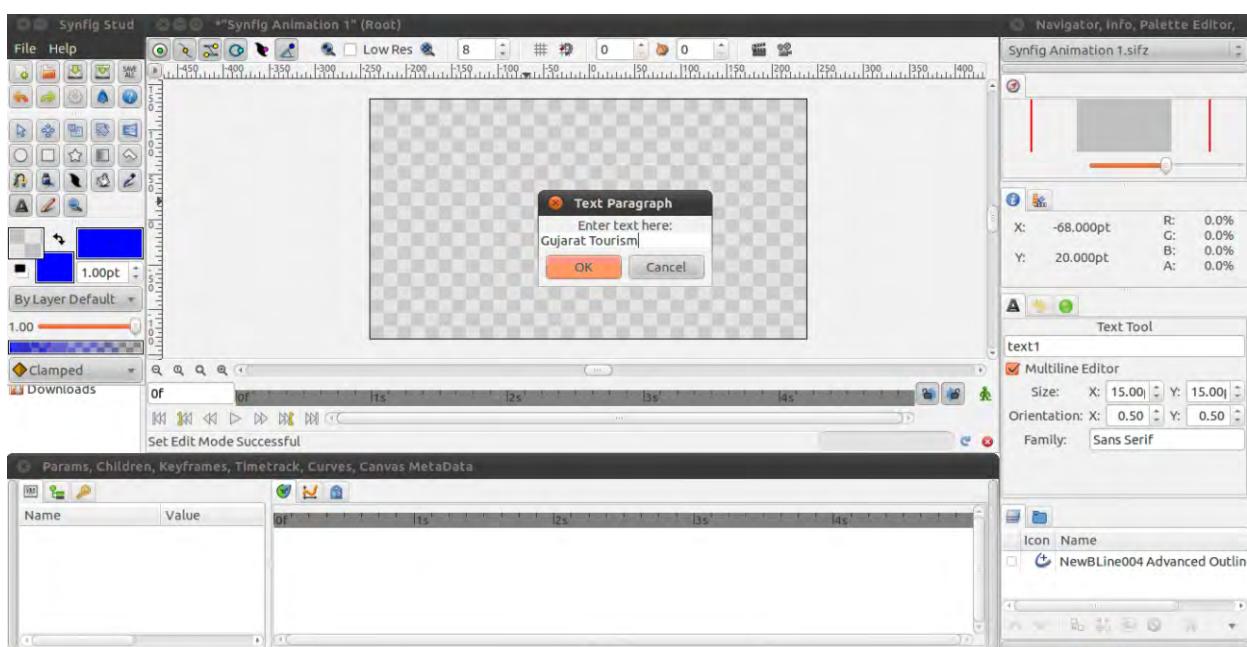


Figure 2.23(a) : Text layer dialog box

Figure 2.23 (b) shows a new text layer created along with its ducks. Using the parameter panel we can change the entered text later on according to our need.



Figure 2.23(b) : Text with ducks

The tool options panel will allow us to:

- Select the name for the new layer; this number will increment as we add new layer.
- Select single or multi line editor for entering the text. If single line is selected then click the enter key to submit the value while in multi-line editor, click the OK button to submit the value.
- Select the horizontal and vertical size of the text.
- Select the ‘orientation’ of the text. By default (0.5, 0.5) is selected which means that the text will be centered around the point you clicked on.
- Select the font family.

Zoom Tool

The Zoom tool is used to get a closer or far away view of the objects in the working area. The zooming does not affect the output results. It is used to just view the objects from close or far range.

Width Tool

The Width tool helps to increase or decrease the width of a line. It acts like the way you use pencil to make a line thicker or eraser to make it thinner. Select the outline layer to change the width on particular segment of a line, and then move mouse over that line, press the left mouse button and move cursor back and forth along the line. The width of outline will be increased at the places where you moved the cursor. To decrease the width hold “CTRL” key while moving the cursor. This tool is used to fine tune the line width.

Transform Tool

Transform tool allows you to select objects and to move, rotate or scale the ducks. Select the object on the canvas window and click the transform tool. If the object cannot get

selected as it is obscured by another layer then you can select its layer in the layer panel. This will display the ducks of the object. Select the duck and make the change according to your need.

Smooth Move Tool

Smooth move tool works just like transform tool, except when multiple ducks are selected. Select multiple ducks by pressing CTRL+a or dragging a rectangle around the ducks. Now when you drag one of the selected duck, the other selected ducks will also move along with it.

Scale Tool

Just as we discussed smooth move tool, here also when we select multiple ducks, and drag one of the selected ducks the other selected ducks will scale relative to the centre of the selected group.

Rotate Tool

As in smooth move and scale tool, when we select multiple ducks, and drag one of the selected ducks the other selected ducks will move rotating about the centre of the selected group.

Summary

In this chapter we learned about the authoring tools and its types. We have become familiar with the Synfig user interface and also learned to create a new file and set its properties. We discussed the purpose of each panel. The use of each tools along with the tool options were discussed. We can use these tools to create and manipulate the objects and then animate later on. So let us now learn to animate these objects. Having learnt functionalities of various tools available in Synfig, we can now proceed to create a multimedia animation using these tools.

EXERCISE

- 1.** What are authoring tools? List different types of authoring tools.
- 2.** What is the purpose of gradient tool? State the use of four types of gradient learned in this chapter.
- 3.** Briefly explain the three palettes of toolbox window.
- 4.** List the important panels and their function.
- 8. Choose the most appropriate option from those given below :**
 - (1)** The elements are organized as pages of a book or pile of cards in which of the following forms?

(a) card or page based tools	(b) icon and event based tools
(c) time based tools	(d) animation tools

LABORATORY EXERCISES

1. Draw a circle, rectangle and star on the canvas. Select different color for each object. Also insert text as per the object.
 2. Using Bline tool draw a candle.
 3. Using appropriate tools draw a flower and a candle.





Creating Animation Using Synfig



In previous chapter we discussed about the Synfig Interface. We also saw how to use different tools to create and manipulate the basic artwork. Let us now learn how to create an animation. As we have discussed earlier, animation is a visual change over time. In other words, it is the sequence of images displayed one after the other in a given time frame. Let us discuss some basics of animation.

Animation

Animation is a visual change over time wherein the digital images are played one after the other to create a moving effect. The images used in animation can be first created and then displayed in a sequence. This type of animation is known as frame by frame animation. Frames are defined as the length of time in the document. Each frame consists of some objects to be displayed over a period of time. Hence in frame by frame animation, each small change in the image is created and then displayed in sequence. This method is slow and tedious. It also requires lot of time and resource.

The other simple way to create animation is through keyframe. A keyframe is a frame in which we define a change to an object's properties. Using keyframes you need to specify the first position and the last position of the object. The in-between frames are automatically understood and need not be specified. For example, say we want to show a circle moving from left to right. Here you need to create the first keyframe on the left and include the last keyframe on the right. The series of frames in between are drawn automatically. This effect is known as tweening. Tweening is an action which calculates the number of frames between keyframes and path of the action. This type of animation is known as cel animation and the computers generally employ this type animation. The real difference between varieties of animation software is how much to be drawn by the user and how much is automatically generated by the computer.

Another type of computer animation is known as kinematics. It is the study of movement and motion of objects that have joints such as walking man or running leopard. In some animations we can use an effect known as morphing, wherein the image transforms into another. Say for example you must have seen that a person's face transforms into another face and so on. Many products are available in the market that supports this type of effect.

Creating Animation

Now let us create our first animation of a circle moving from left to right. Follow the steps given to create this animation:

- Start Synfig (or Select Application → Graphics → Synfig), this will automatically open a new file for you.
- Click the Caret → Edit → Properties. This opens the canvas properties dialog box as shown in the figure 3.1.

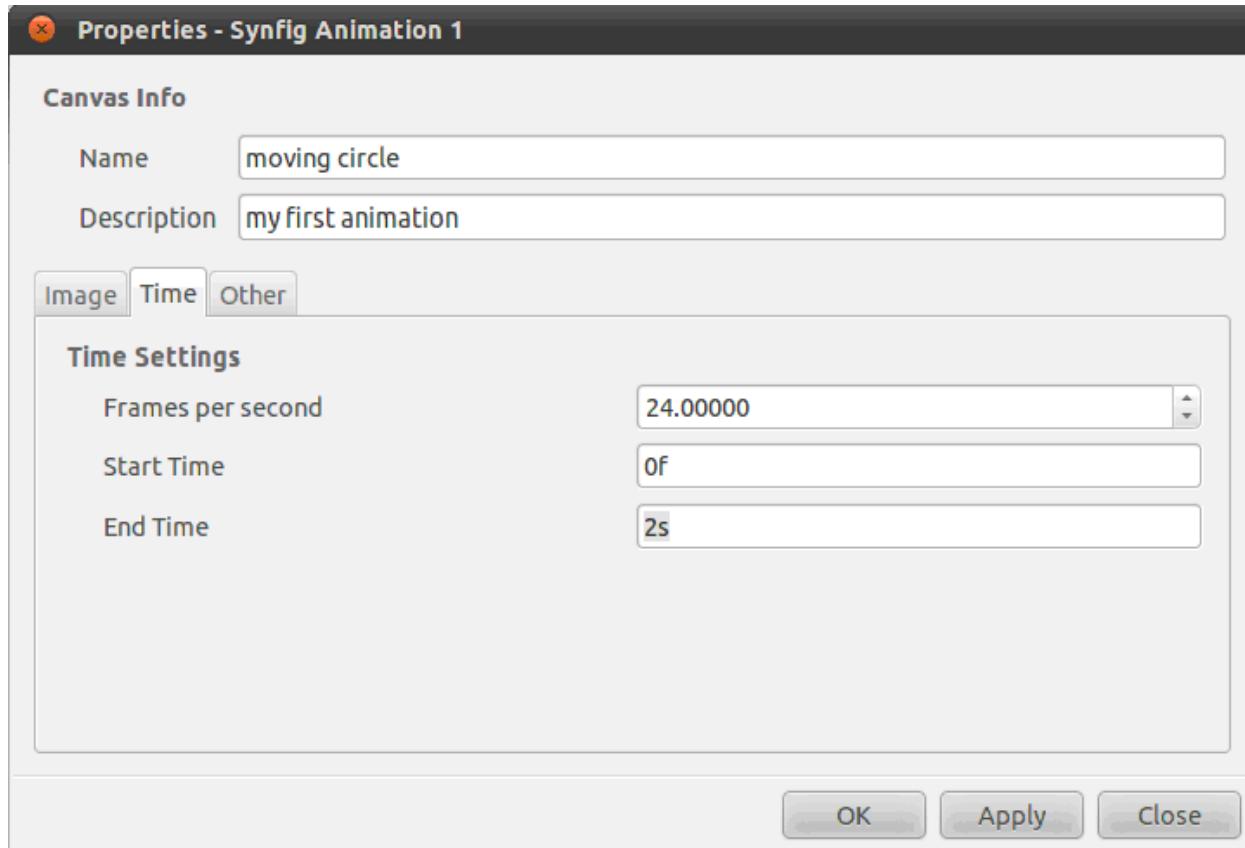


Figure 3.1 : Canvas property dialog box

- Give a name and description to your first animation. As shown in figure 3.1 we have given the name to our animation as “moving circle” and description “my first animation”. Now click on Time tab and change the End time to 2s. It means that our animation will be of 2 seconds duration. By default the End time is set as 5s.
- Click Ok. This will close the canvas properties dialog box and the changes we made can be seen as the file name has changed and the timeline is now 2s long.
- Now we need to draw a circle. Select the color of the circle by clicking on foreground color or using the palette. We can also change the color later by using the parameter panel. Currently we choose the fill color as yellow. Then select the circle tool and create a circle as shown in figure 3.2. You can use the transform tool to make changes in the circle drawn.

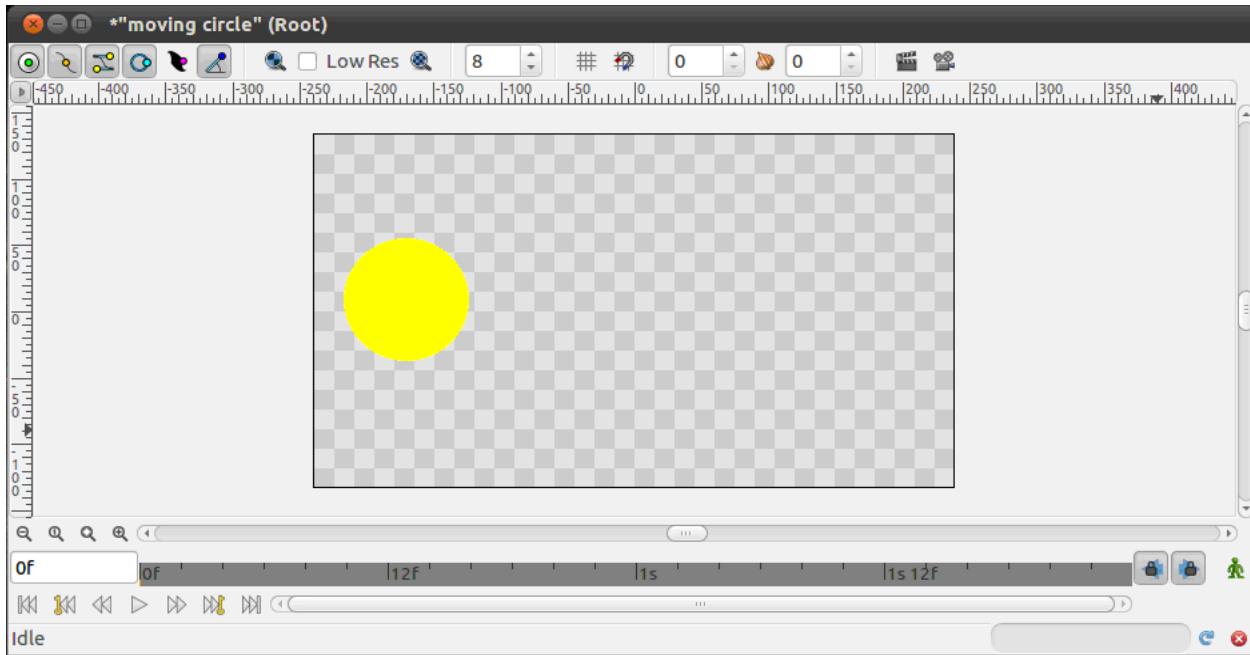


Figure 3.2 : Use of circle tool

Now we are ready with the object that is to be animated. Before starting with animation let us understand two important things: Timebar and Animate editing mode.

Timebar

As shown in figure 3.3 click on the timebar, you can see the orange indicator showing the position in time. Clicking at various points, you can see values “0f, 1s, 1s 10f” etc. in the entry field. You can also type the position in the entry field and press Enter, the indicator will take you to that position in timeslider. These values indicate a specific point on the timeline in terms of seconds (s) and frames (f). By default each second is divided into 24 frames. As you can see in figure 3.1 frames per second (fps) is set as 24. The frame starts at 0 and goes up to 24. For example, when one second and thirteen frames have passed, the entry box displays “1s 13f”.



Figure 3.3 : Timebar with the indicator at 0f, 1s and 1s 13f

Animate editing mode

On the right side of the timebar shown in the figure 3.3, you can see a green button () . It will be visible if you have a non-zero end time. Clicking on this button, turns it into red button () and the canvas displays a red outline as in figure 3.4. This means that now the animate editing mode is active and any changes made will have an effect on the animation in the timeslider.

In animate mode, whenever a parameter of an object is changed, a waypoint is created to remember the change in the position and the time on the timeslider at which the change happened.

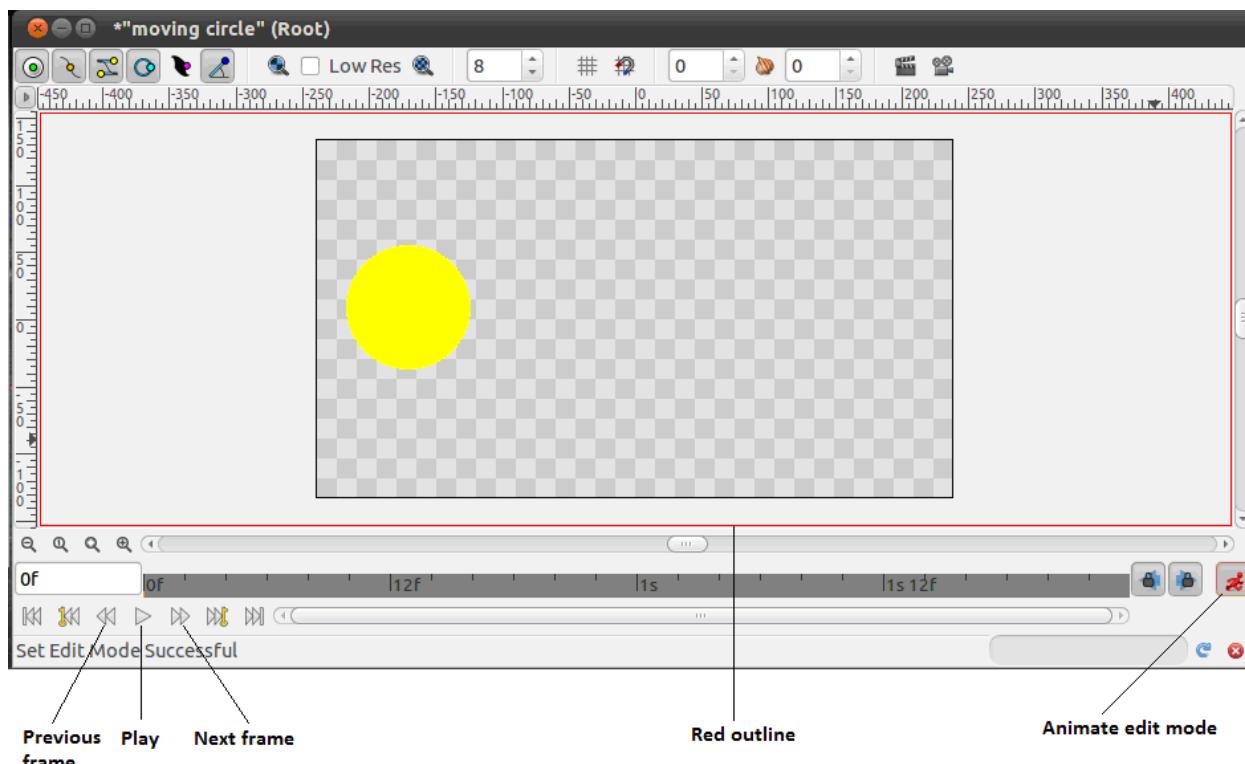


Figure 3.4 : Animate editing mode

When not in animate editing mode, changes to a parameter are applied throughout the entire timeline of the animation.

- Let us now start creating our animation. Click on Animate editing mode.
- You can see “0f” in timeline entry field. Move the circle slightly so that it records the location for the first frame. You can also move the circle to a new location on the working area as the starting point.
- Now click at the end of the timeline. This will show the ‘2s’ mark in the entry field. You can even type ‘2s’ in the entry field to reach the end of timeline. Grab the circle

and hold the shift key to move it in a straight line to the right of the working area as shown in figure 3.5.

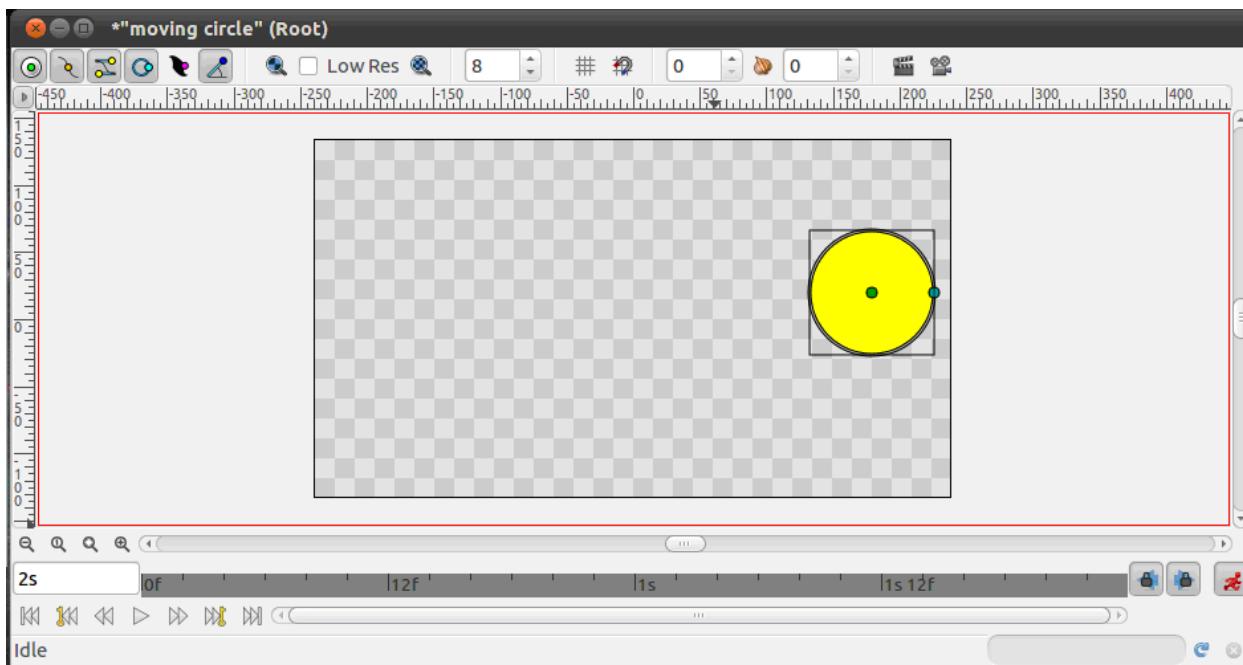


Figure 3.5 : Moving the circle to the right

- Click on any position on the timeslider and you will see the circle on a new position. We had specified only the beginning and the end locations of our circle, but Synfig has automatically created the in-between images or frames. As we discussed earlier, this process is known as tweening. These frames when displayed together will give an effect of moving circle.
- Click on the play key as shown in figure 3.4 to see how our animation will look. You can also click on the previous frame and next frames buttons to see frame by frame animation.
- Finally we need to render our work. Rendering means the computer uses various algorithms to apply the final effects you have specified on the objects created.
- Before rendering, click on the  button to leave the animate editing mode.
- Click on Caret → File → Save. Save the file by the name “moving-circle.sifz”.
- Click on Caret à File à Render. Change the filename to “moving-circle.gif” and the same location where you saved “moving-circle.sifz”. Select target format as “gif” instead of “Auto”, then click Render as shown in figure 3.6 (a). It will take few seconds for the file to render. You can see the message “file rendered successfully” on window status bar located on the bottom of the window.

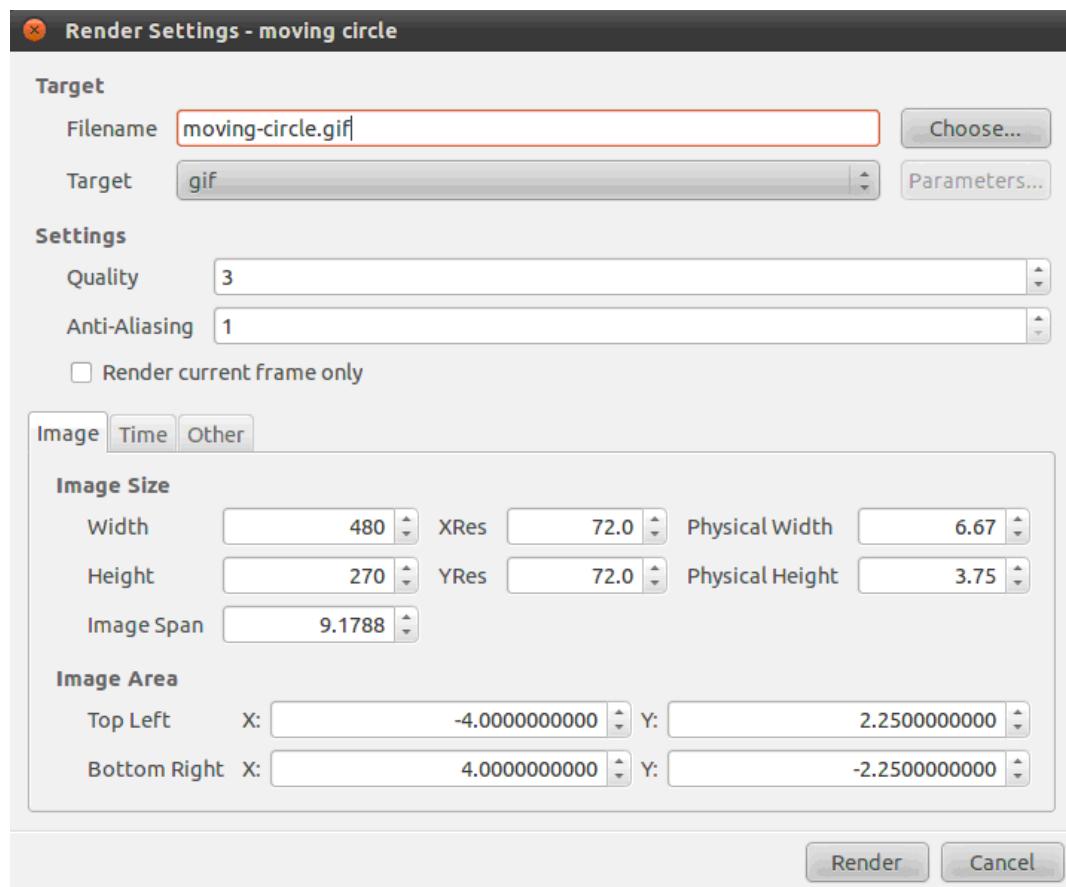


Figure 3.6(a) : Render the file

- Open moving-circle.gif file in a web browser or image viewer to see your animation. Figure 3.6 (b) shows the output as visible in the web browser.

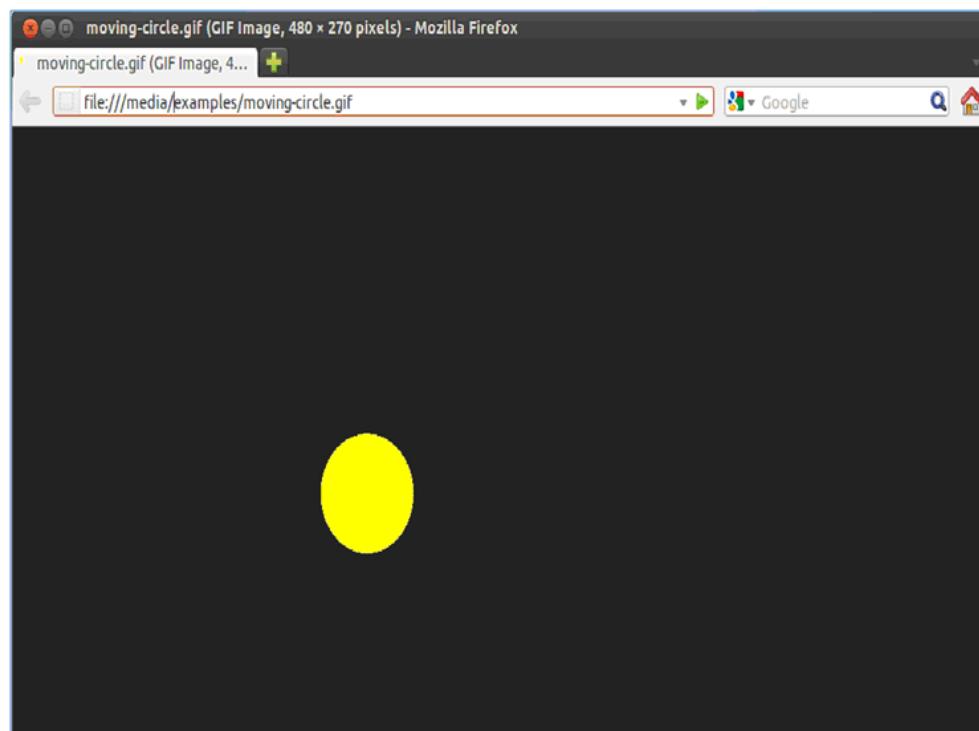


Figure 3.6(b) : Output displayed in web browser

Good. So, you have created your first animation.

Animation using keyframe

Let us create an animation of moving the circle from left to right and then left again using keyframes. For this we need to create following three steps:

- First the circle is on the left.
- Second, the circle is on the right.
- And third, the circle is back on the left.

Follow the steps given to create this animation:

- Start Synfig and open a new file.
- Click the Caret → Edit → Properties. Give a name and description to your animation. As shown in figure 3.7 we have given the name to our animation as “moving circle2” and description “moving circle using keyframes”. Now click on Time tab and change the End time to 2s. Click OK to make the changes.

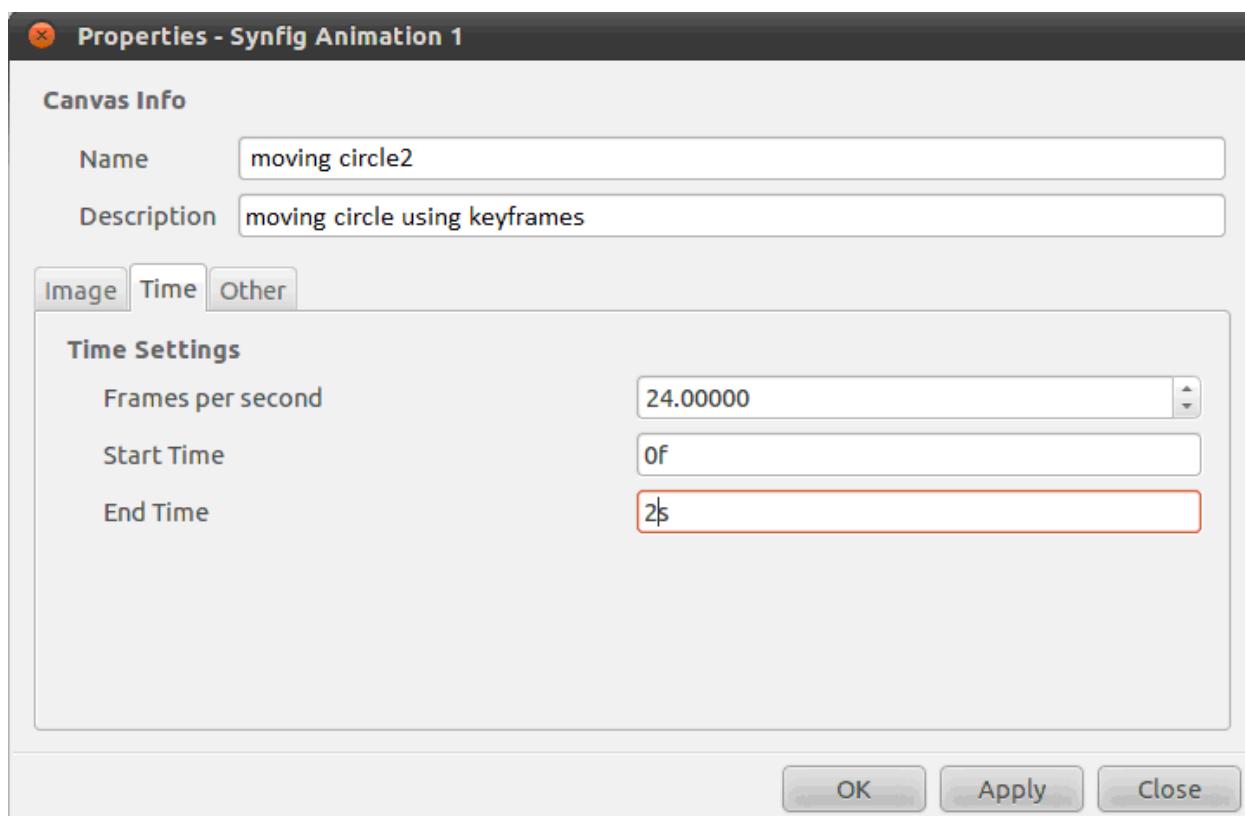


Figure 3.7 : Canvas property dialog box

- Now let us draw the background for our animation. Select the rectangle tool and create a simple black rectangle as shown in the figure 3.8.

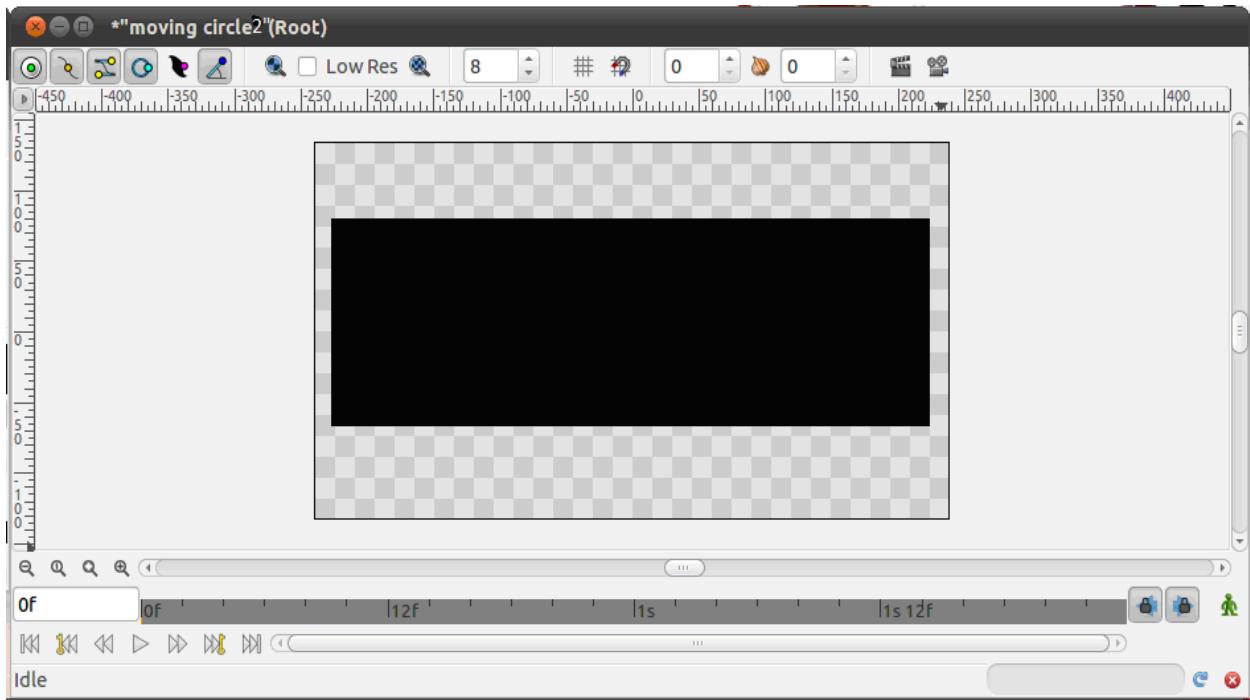


Figure 3.8 : Rectangle drawn using rectangle tool

- Now we need to draw a circle. First change the fill color to green. Then select the circle tool and create a circle as shown in figure 3.9. You can use the transform tool to make changes in the circle drawn.

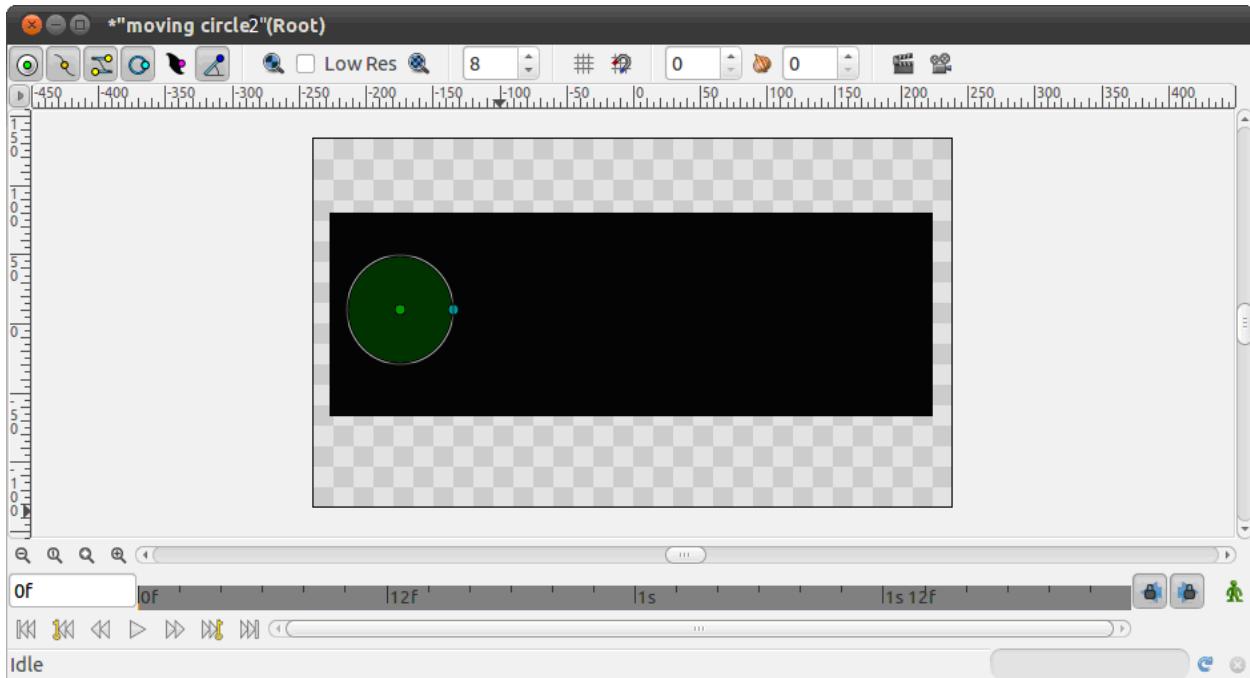


Figure 3.9 : Circle drawn using circle tool

- As the object is ready let us now start creating our animation by clicking on Animate editing mode.

- Go to parameter panel and click on  to open keyframe panel to add keyframes. Earlier we discussed the three steps; these will now become our keyframes. Click the small button  to add a new keyframe as shown in figure 3.10. This will make a new entry in the list displaying “0f, 0f, (JMP)”.

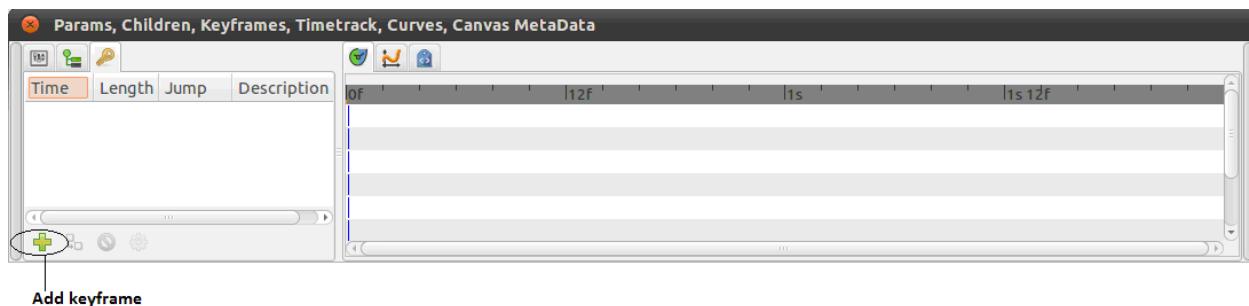


Figure 3.10 : Keyframe panel

- Go to the “1s” mark in the time slider. Click on  to open keyframe panel to add keyframes. Again go to “2s” in the timeslider and add another keyframe. So now you have three keyframes in the list as shown in figure 3.11.

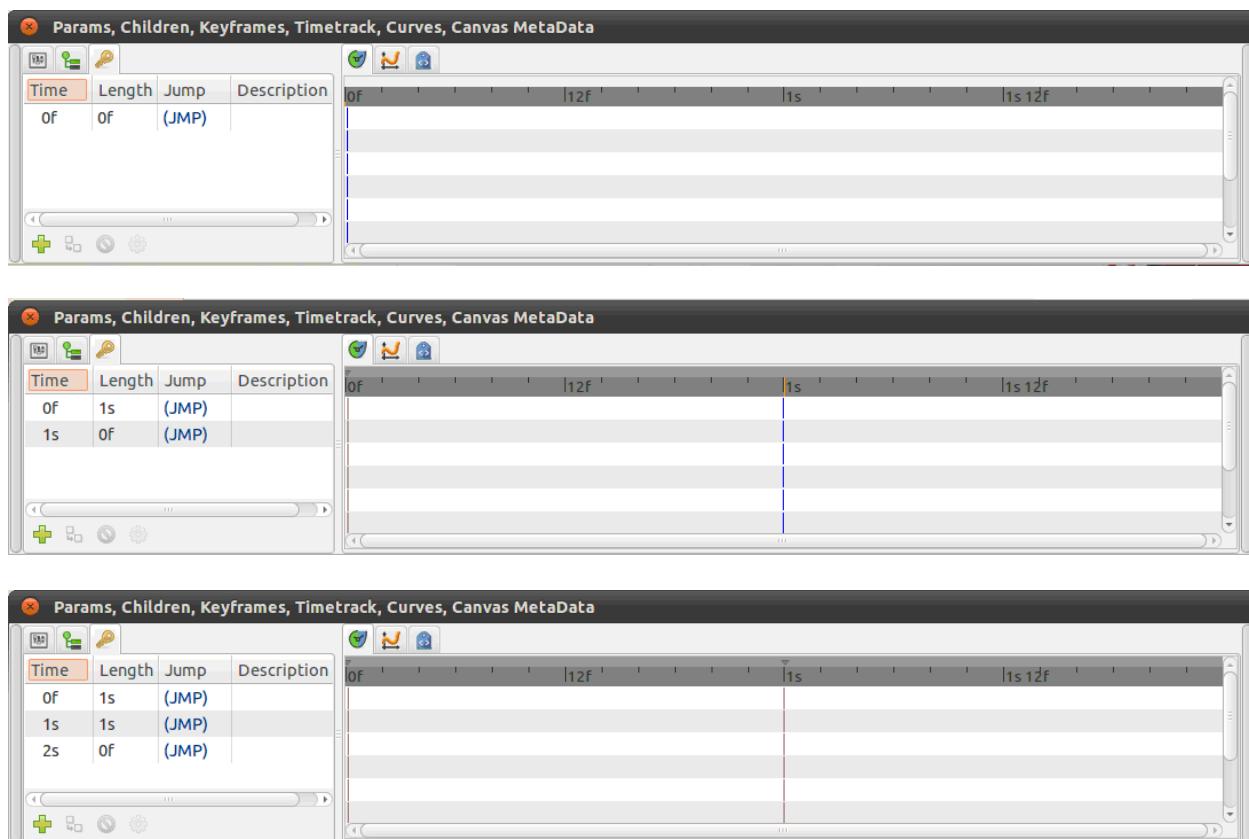


Figure 3.11 : Three keyframes at 0f, 1s and 2s

Let us understand the keyframe panel. It displays “Time” which indicates the start time, “Length” which indicates duration, “Jump” which are links used to jump to correct time and “Description”.

- Here, in all the three keyframes the circle is on the left. But as discussed in our steps the second position of the circle is on the right. So now let us edit the circle at second keyframe i.e at 1s mark. Click on the “(JMP)” of 1s mark and move the circle from left to right as shown in figure 3.12 (a). So finally, we have our circle at 0f on the left, then at 1s we have it on right and on 2s we have it on left again. On the Timetrack panel you can see three orange diamonds known as Waypoints as shown in figure 3.12 (b). They represent the time at which the parameter (like color or location) of the object changes.
- Click on any position on the timeslider and you will see the circle on a new position. We specified only three locations of our circle, but Synfig has automatically created the in-between images or frames. These frames when displayed together will give an effect of moving circle.

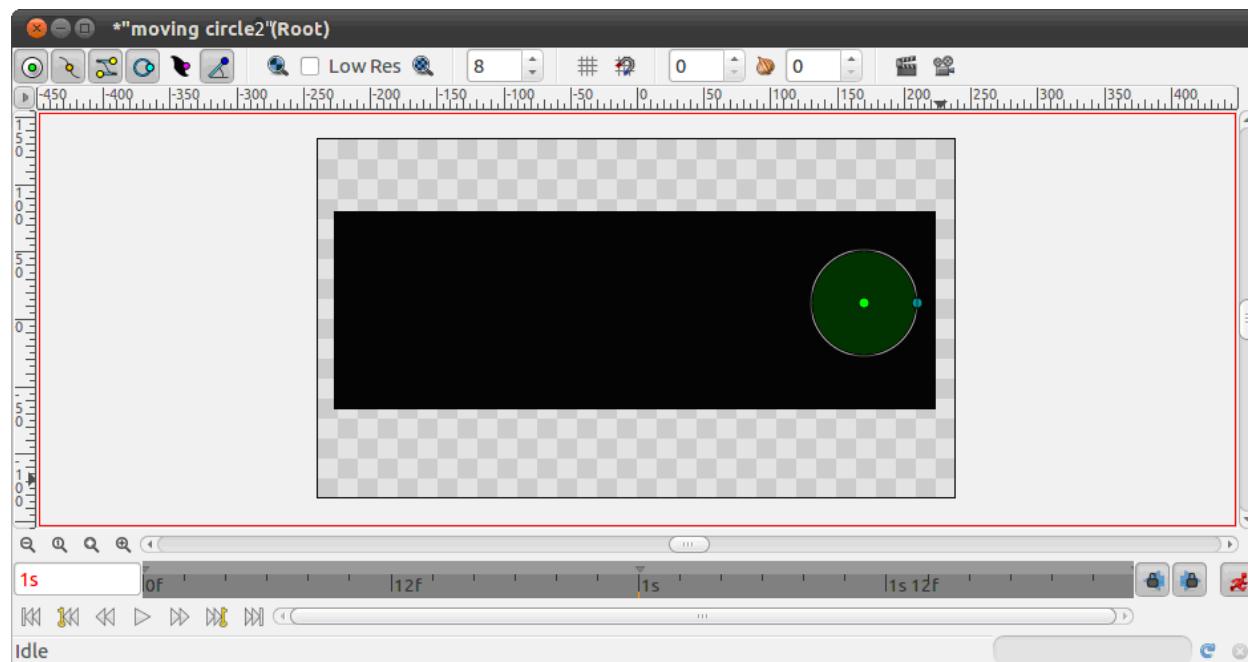


Figure 3.12(a) : Moving circle at 1s to its right

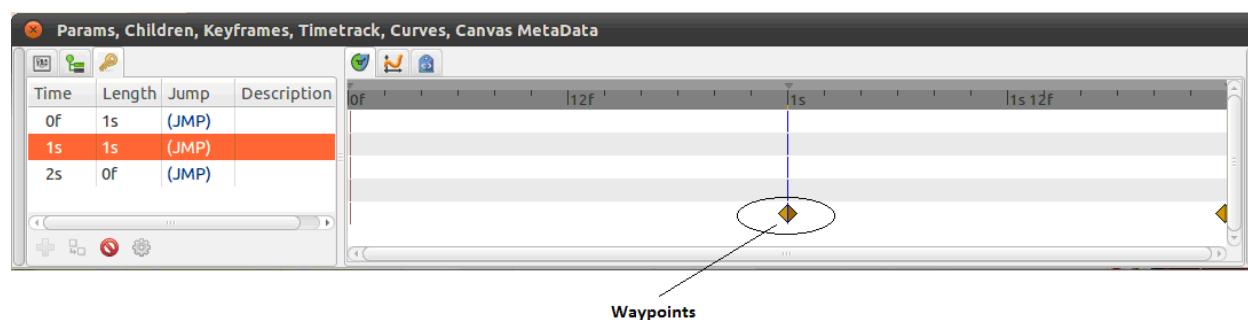


Figure 3.12(b) : Waypoints

- Now to see the animation we need to render our work.
- Before rendering, click on the  button to leave the animate editing mode.
- Click on Caret → File → Save. Save the file by the name “moving-circle2.sifz”.
- Click on Caret → File → Render. Change the filename to “moving-circle2.gif” and the same location where you saved “moving-circle2.sifz”. Select target format as “gif” instead of “Auto”, then click Render as shown in figure 3.13 (a). You can see the message “file rendered successfully” on window status bar located on the bottom of the window.

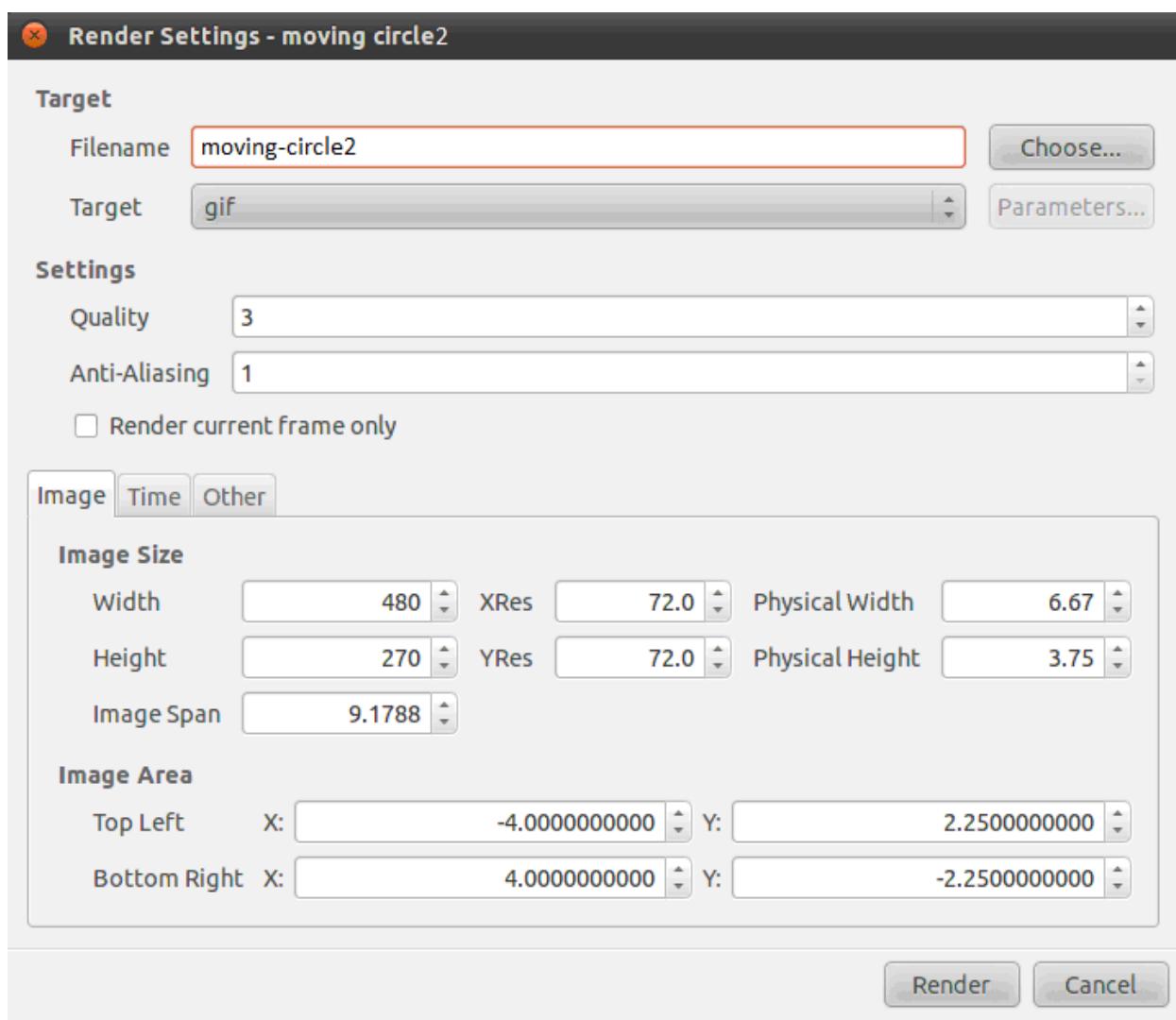


Figure 3.13(a) : Render the file

- Open moving-circle.gif file in web browser to see your animation. Figure 3.13 (b) shows the output.

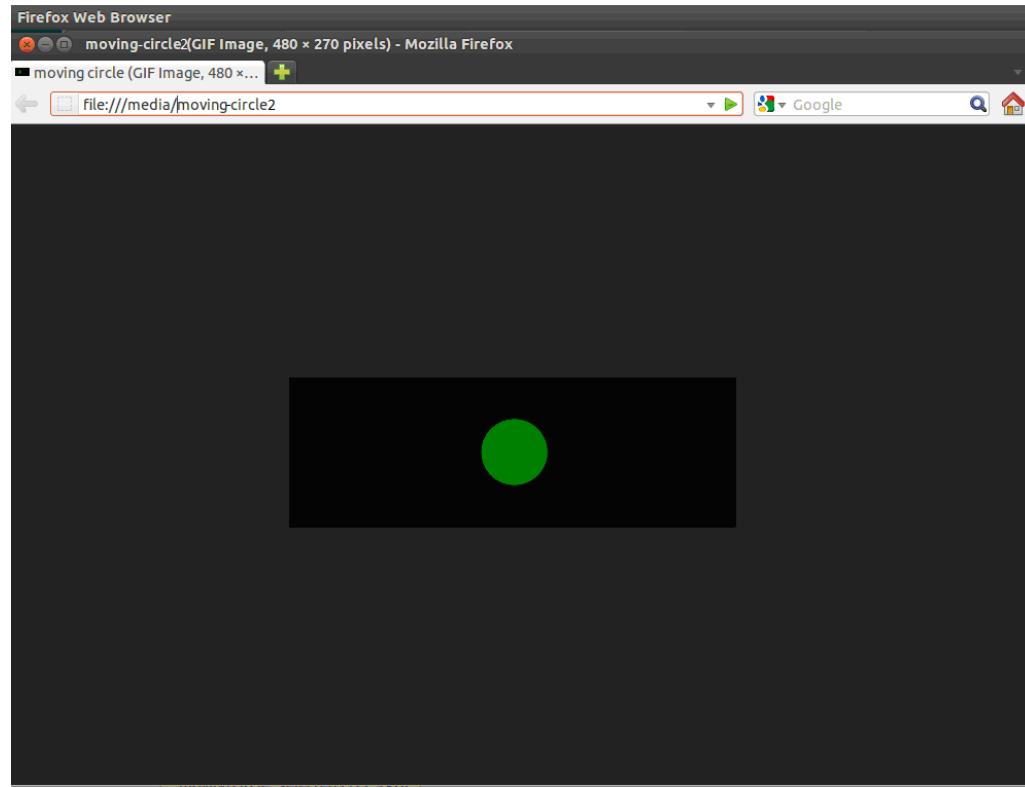


Figure 3.13(b) : Output displayed in web browser

Using Time Loop

If you want your animation to repeat again use the time loop. Let us take an example of a bouncing ball to understand Time loop. We will also see how to duplicate a keyframe which saves time in positioning the object in the animation all over again. In the example we will bounce the ball on the wall. The animation will be created twice using duplicate keyframes concept. And then use the time loop to show the bouncing continue till the end time of the animation.

- Create a new file.
- Select the Circle tool and in the tool options select only the region and the outline layer as in figure 3.14. Draw a circle on the canvas as shown in figure 3.15.

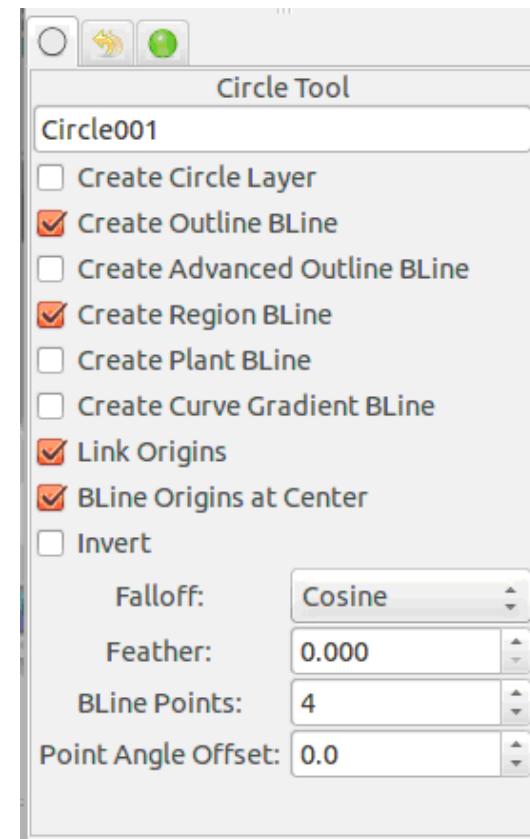


Figure 3.14: Circle tool options

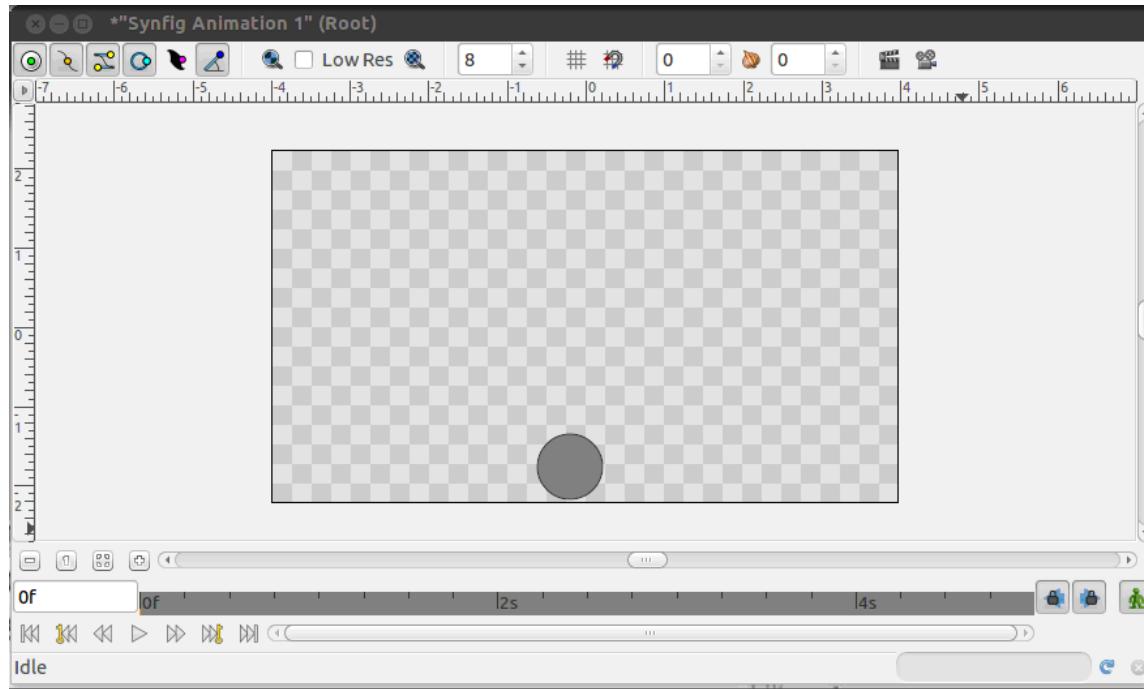


Figure 3.15 : Ball drawn using Circle tool

- Turn on the animate mode. In the timeline, record the position at “0f” by moving the ball slightly from its position. Now select the keyframe panel and add a keyframe by pressing at the bottom of the keyframe panel. Figure 3.16 shows the keyframe added at “0f”.

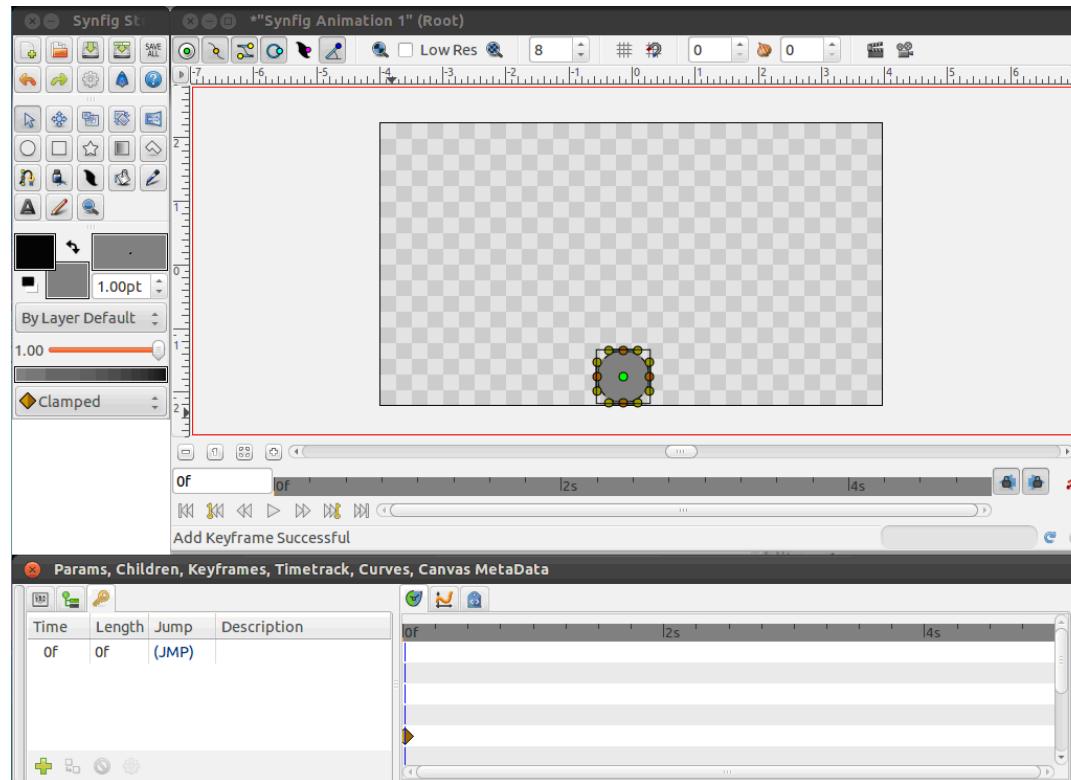


Figure 3.16 : Keyframe at 0f

- Now in the timeline come to 6f and drag the ball to a new position. Add a new keyframe at “6f” as shown in figure 3.17.

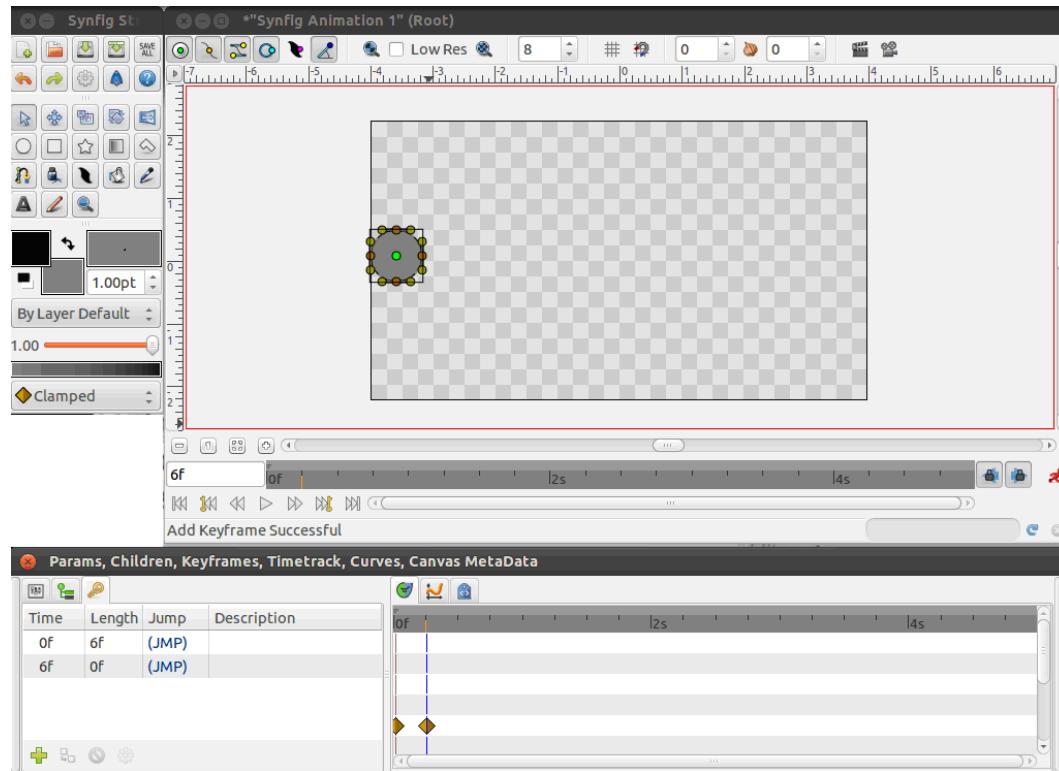


Figure 3.17 : Keyframe added at 6f

- Similarly, go to “12f”, “18f” and drag the ball to the location as shown in figure 3.18 and 3.19. Add a keyframe at both the timing location.

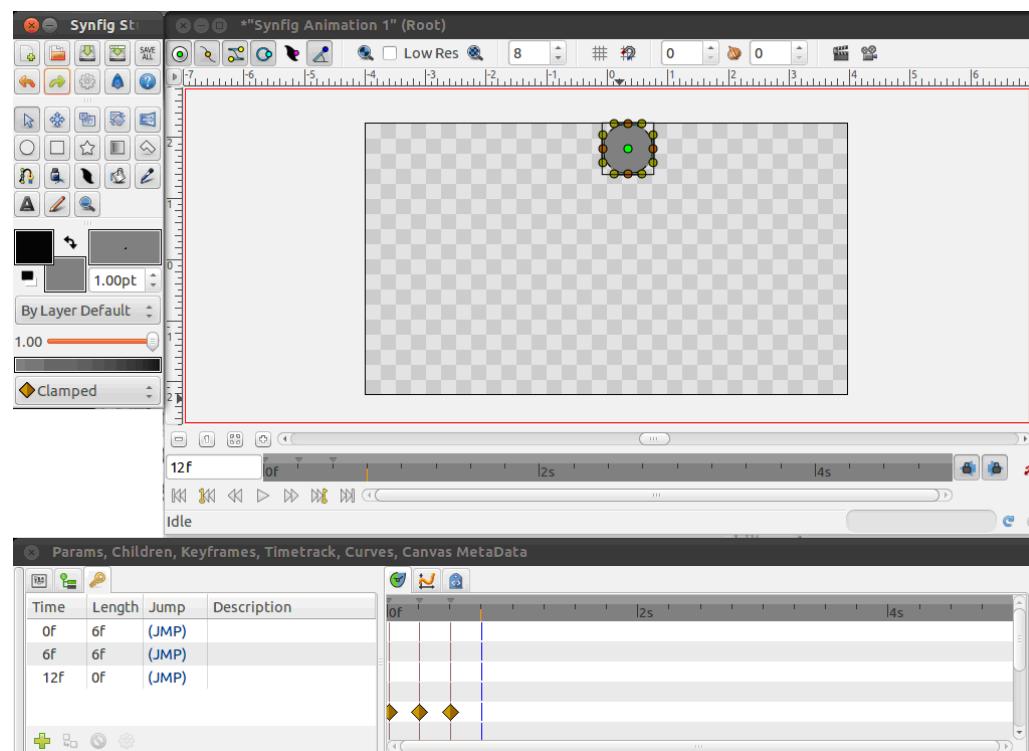


Figure 3.18 : Keyframe added at 12f

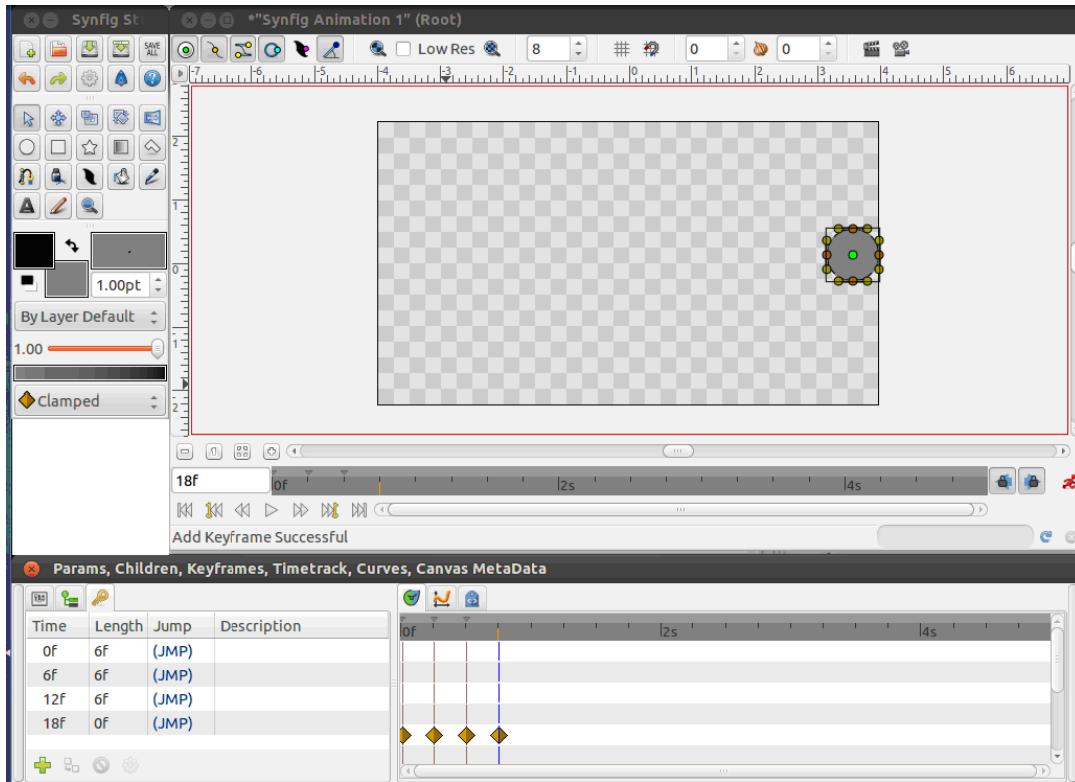


Figure 3.19 : Keyframe added at 18f

- Now come back to “1s” in the timeline, here we need the ball to come to its original position which we had at “0f”. So we need to duplicate the keyframe at “0f”. Select the “0f” keyframe from the keyframe panel and press duplicate keyframe as shown in figure 3.20. This will add a new duplicate keyframe at “1s” as shown in figure 3.21.

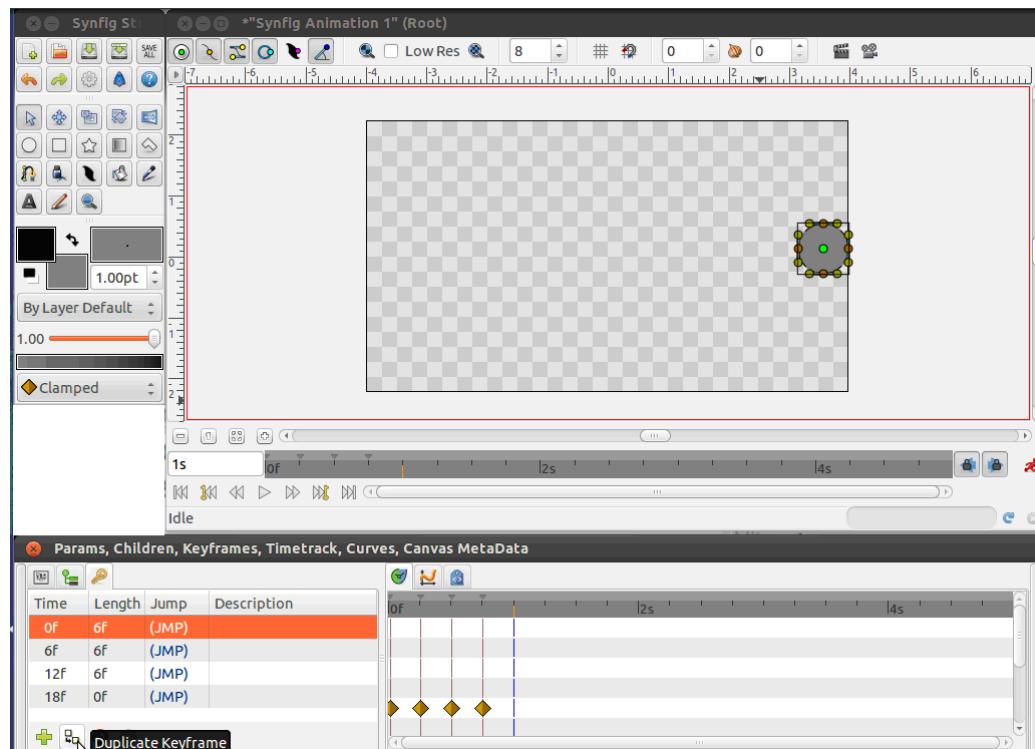


Figure 3.20 : Add a duplicate keyframe at 1s

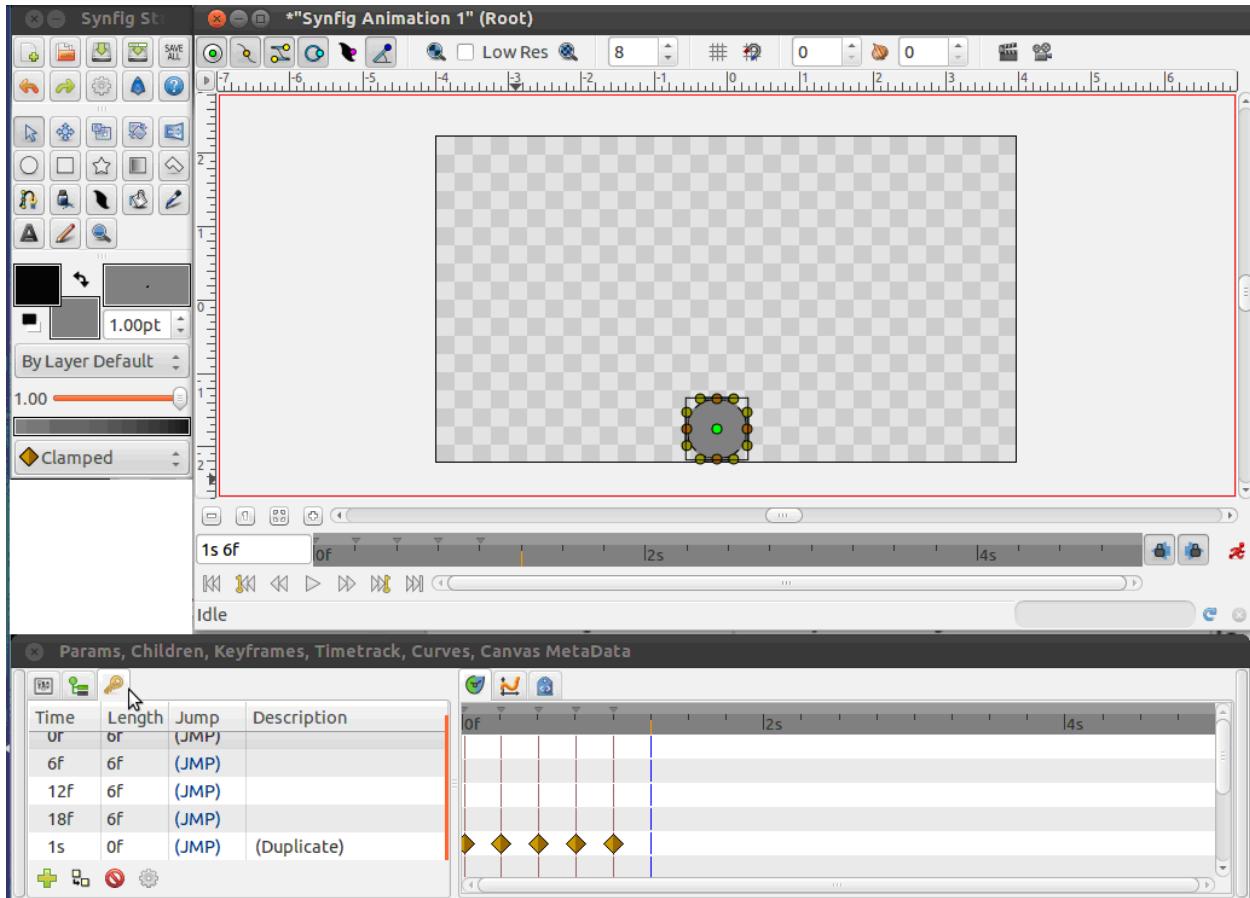


Figure 3.21 : Panel showing duplicate keyframe added

- Likewise to animate the ball twice we need to create the duplicate keyframes at 1s6f, 1s12f, 1s18f and 2s. Figure 3.22 shows the duplicate keyframes added.

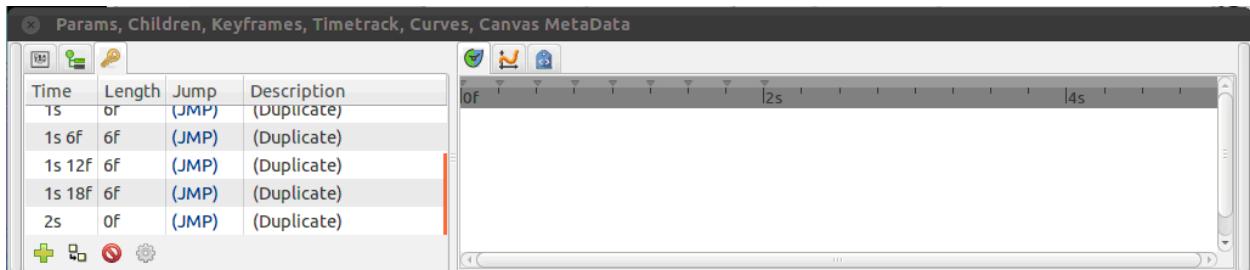


Figure 3.22 : Duplicate keyframes added

- Now preview the animation using File → Preview. You will see the ball bouncing twice till 2 seconds and then stop. We want the animation to go on till the end time. In our case the end time is 5 seconds.

We will insert a new time loop layer above the circle layer.

- Select topmost circle outline layer → Right click → Add new layer → Other Time loop. This will add a new layer as shown in figure 3.23.

	Icon	Name	Z Depth
<input checked="" type="checkbox"/>		Time Loop	0.000000
<input checked="" type="checkbox"/>		Circle001 Outline	1.000000
<input checked="" type="checkbox"/>		Circle001 Region	2.000000

Figure 3.23 : Inserting time loop layer

Params, Children, Keyframes, Timetrack, Curve		
  		
Name	Value	Type
 Z Depth	0.000000	real
 Link Time	0f	time
 Local Time	0f	time
 Duration	1s	time
 Only For Positive Duration	<input type="checkbox"/>	bool
 Symmetrical	<input checked="" type="checkbox"/>	bool

Figure 3.24 : Duration parameter in the Time loop

- The time loop parameters in figure 3.24 shows the duration which denotes the frames or seconds that are looped. We can change it to 2s as our animation is playing twice till 2s.
- Preview the animation and you will see that the animation goes on playing till the end time. If you change the endtime to “10s” in the file properties window then the animation will go on playing till 10s.
- Save and render the file. Figure 3.25 shows the output on web browser.

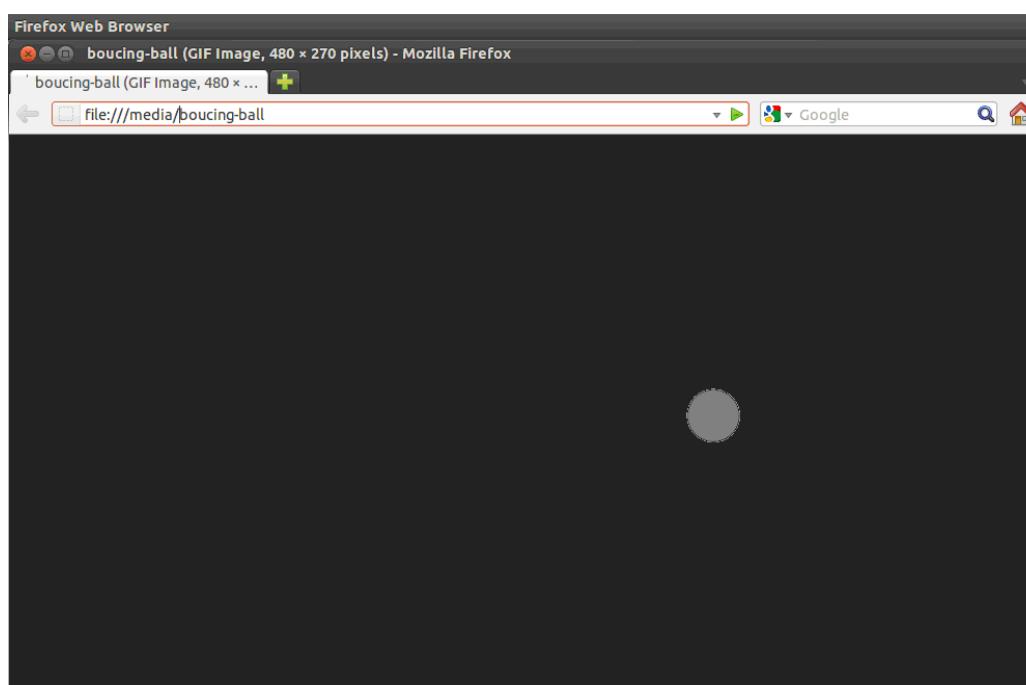


Figure 3.25 : Output displayed in web browser

Summary

In this chapter we learned how to create animation. Animate edit mode helps in recording the changes made to the object's position at different time using the timebar. When not in animate edit mode, any changes made to the object or its parameters will not have an effect on the animation. The concept of keyframe and waypoint was discussed. We also learned how to repeat an animation till the end time using the time loop layer. In the next chapter we will learn more about the concept of layers and using it to give effects to the objects.

EXERCISE

LABORATORY EXERCISES

1. Create an animation showing the star moving from top to bottom.
 2. Create an animation showing a ball bouncing on the floor. Use time loop to continue the animation till the end time.
 3. Create an animation showing a circle moving from right to left and at the same time star moving from left to right on the canvas.
 4. Create an animation showing three balls bouncing at different speeds.



Introduction to Layers



A beautiful sight of a rising sun is always soothing to eyes. If this sunrise is observed in mountains, the sight becomes more pleasurable. This scene consists of many objects like mountains, sun, birds, trees etc. Assume that we want to create such an animation. Each of these objects when drawn on the canvas will be either above or below another object. For example, sun rises from behind the mountains, so the object ‘sun’ should be placed behind the object ‘mountain’. Similarly, if the mountain is behind the hut, the object ‘hut’ will be placed above the object ‘mountain’. Likewise we can place the other objects in our example like the trees and birds above or below the other objects. In this chapter we will learn about a concept named “layers” that will help us to create the scene mentioned here.

Layers

A “layer” in Synfig refers to different levels on which we can place our drawing objects or pictures, stacked above and below each other. Top layers will hide bottom layers. We can think of layers as sheets of paper which are cut into different shapes. On one layer we have a paper-sheet shaped like a star. On second layer we have a paper-sheet shaped like a square. The square might overlap and hide some part of the star.

Use of layers gives us the freedom to work with each object individually. When modified it does not disturb the other objects either above or below it. Layers can also be rearranged, allowing us to change the order in which objects are displayed, thus changing the appearance of our scene.

In Synfig, each individual element of the canvas is broken down into layers. For example, when we had drawn a circle it appeared on a different layer. The layers in Synfig are different from other animation software in the following two aspects:

1. Every object, element, and effect have their own layer. We do not have a layer with multiple effects. In other words a layer represents a single primitive such as object, outline or image.
2. The upper layer can change the behavior of the layers below it. It can distort or modify the layers beneath. This is done by using effect or filter layers.

There are different types of layers such as gradient, blur, distortion, filter and others.

Each layer has its own set of parameters. We can see the parameters of a layer by first selecting the layer from the layers panel and then looking into the params panel. Let us understand layers

concept by taking an example. Assume that, we want to give a gradient effect to a circle layer. Perform the steps mentioned to achieve gradient effect.

- Create a new file.
- Create two circles of different sizes using the circle tool. Keep the color of the circle as white so as to see the effect of gradient. Figure 4.1 shows these two circles.

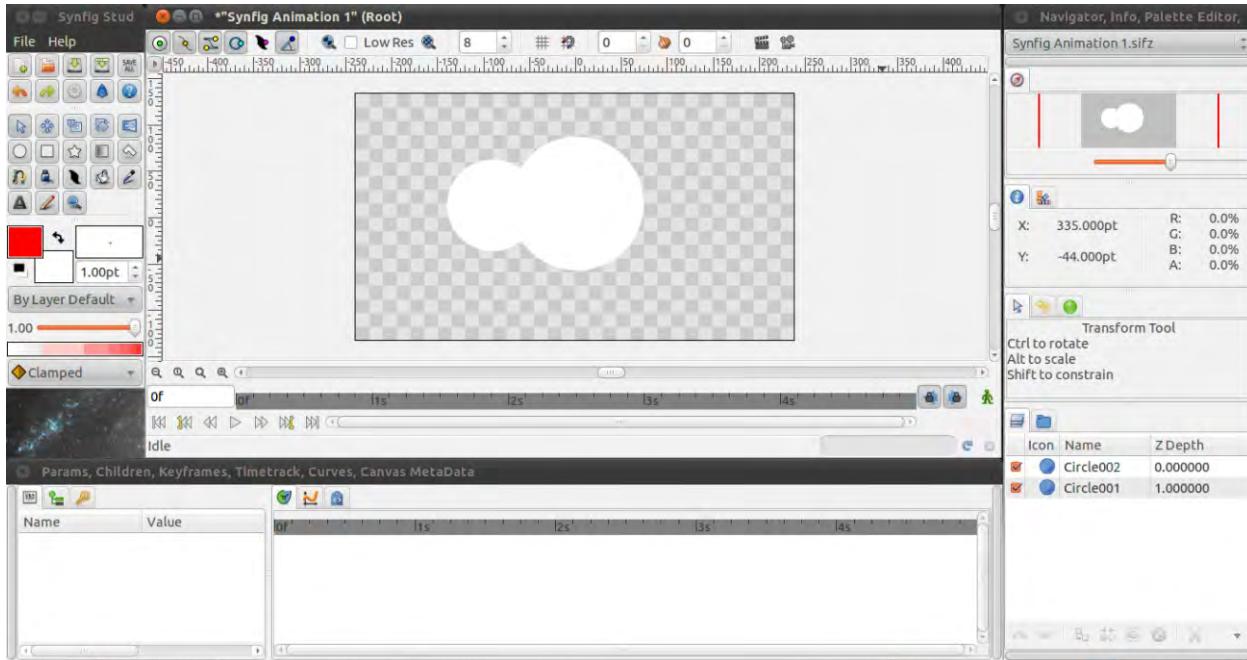


Figure 4.1 : Two circles of different size

- Select the gradient tool from toolbox. Select the outline color of your choice. Here we have selected the outline color as red. Now drag the gradient over the area as shown in the figure 4.2.

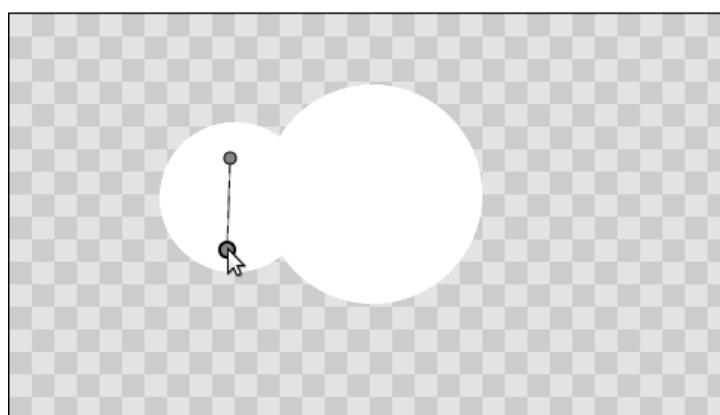


Figure 4.2 : Add gradient

- You see the entire canvas is covered by the gradient as in figure 4.3. This can be useful if you want to use the gradient as the background layer. But we want the gradient to be applied only onto the objects.

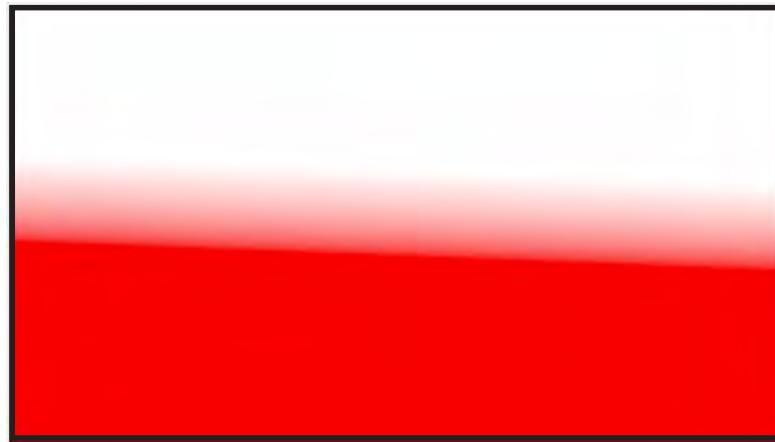


Figure 4.3 : Effect of Gradient on the canvas

- Select the gradient layer and go to the params panel. You can see “blend method” parameter. Blend mode defines how the image is layered on everything below it. By default, the setting is “Composite”, which simply displays the content of the layer. In the blend method parameter change the setting to “straight onto” as shown in figure 4.4. Now you can see that the gradient is applied to the object layers below it. But we want the gradient to be applied only to a single object and not all the objects below the gradient layer. For this we will be using encapsulation.

Encapsulation means grouping things together. If we want to apply gradient to a particular object or a group of objects then we need to encapsulate them into a layer. Thus, using encapsulation we can apply the gradient to one layer without applying it to the layers beneath it.

So, in our example we need to select the gradient and the object (to which the gradient is to be applied) and encapsulate them together.

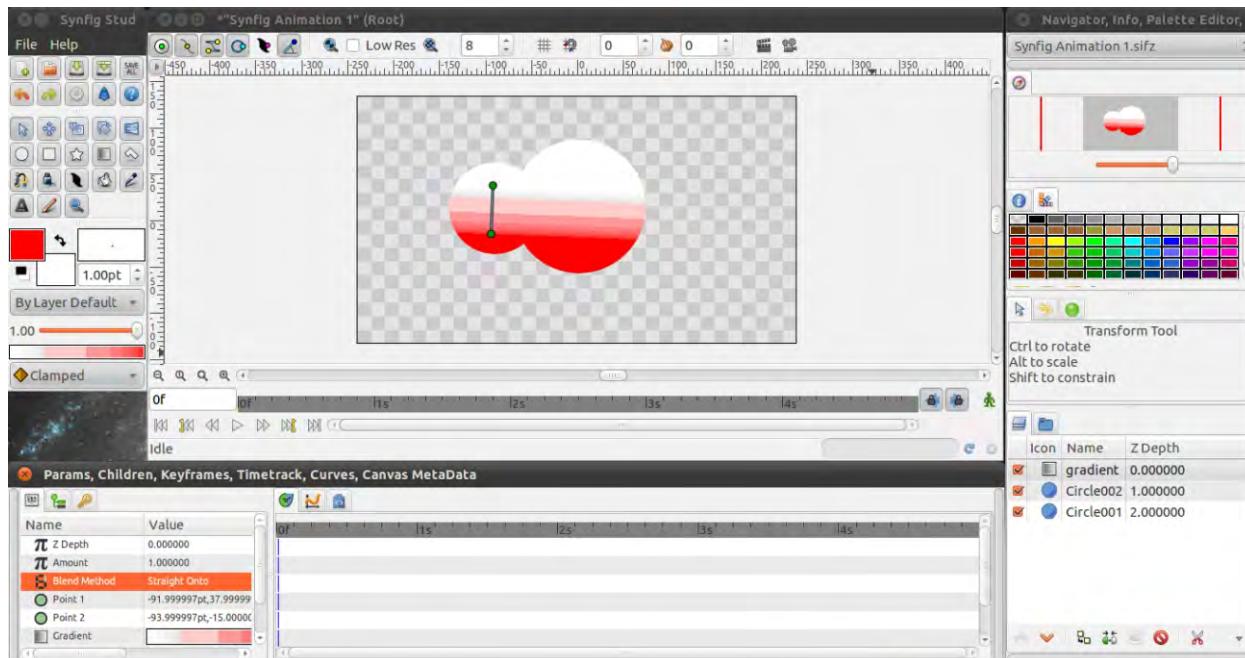


Figure 4.4 : Use of “straight onto” blend method

- Select the gradient layer and the circle layer (press and hold the CTRL key to select two layers) on which we want to apply the gradient. Right click → Encapsulate. This will create a new layer named “inline canvas” as shown in figure 4.5. You can see a small triangle near the inline canvas. Click on this and the inline canvas will expand to show you the two layers: gradient and circle as given in figure 4.6. Figure 4.7 shows the canvas with gradient applied only to the smaller circle.

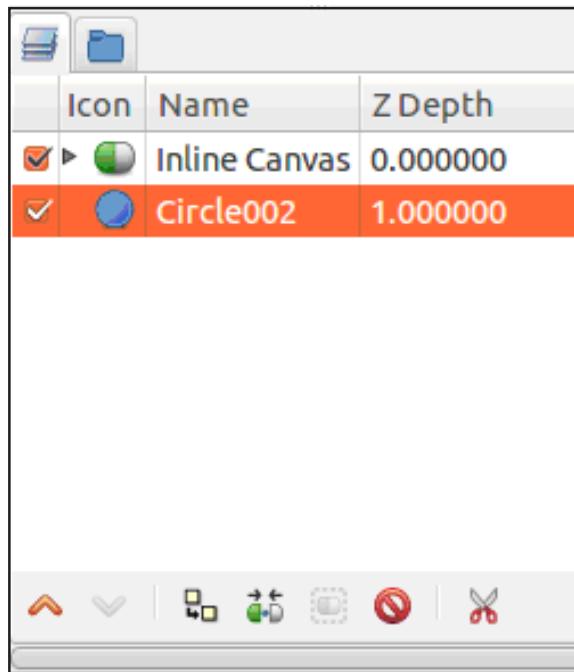


Figure 4.5 : Inline canvas layer created

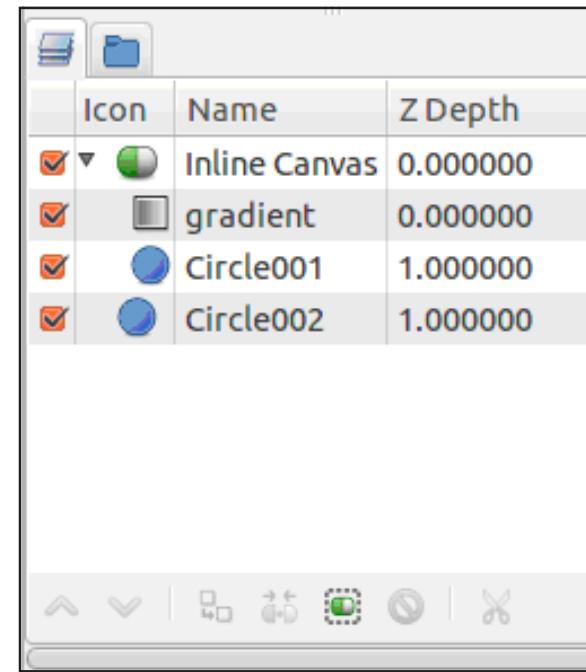


Figure 4.6 : Expanded Inline canvas layer

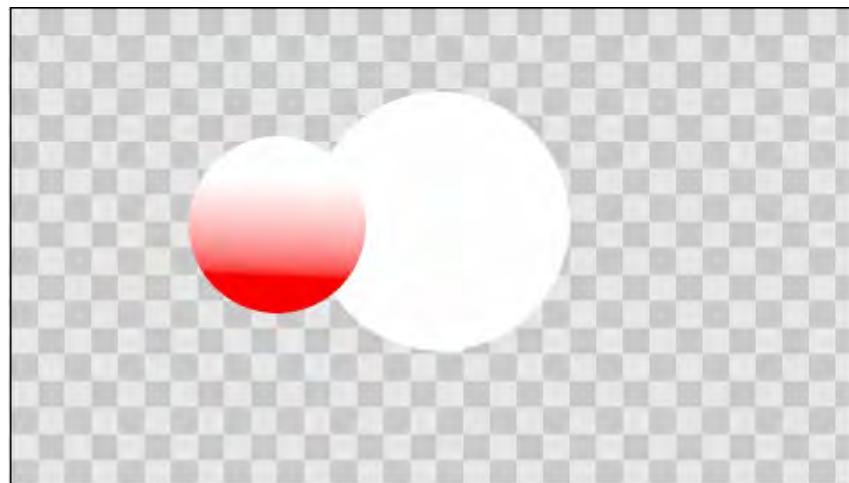


Figure 4.7 : Canvas showing the gradient effect after encapsulation

Thus we have seen how the gradient is applied to the objects using encapsulation. To make it more clear let us add one more circle using the circle tool as shown in figure 4.8.

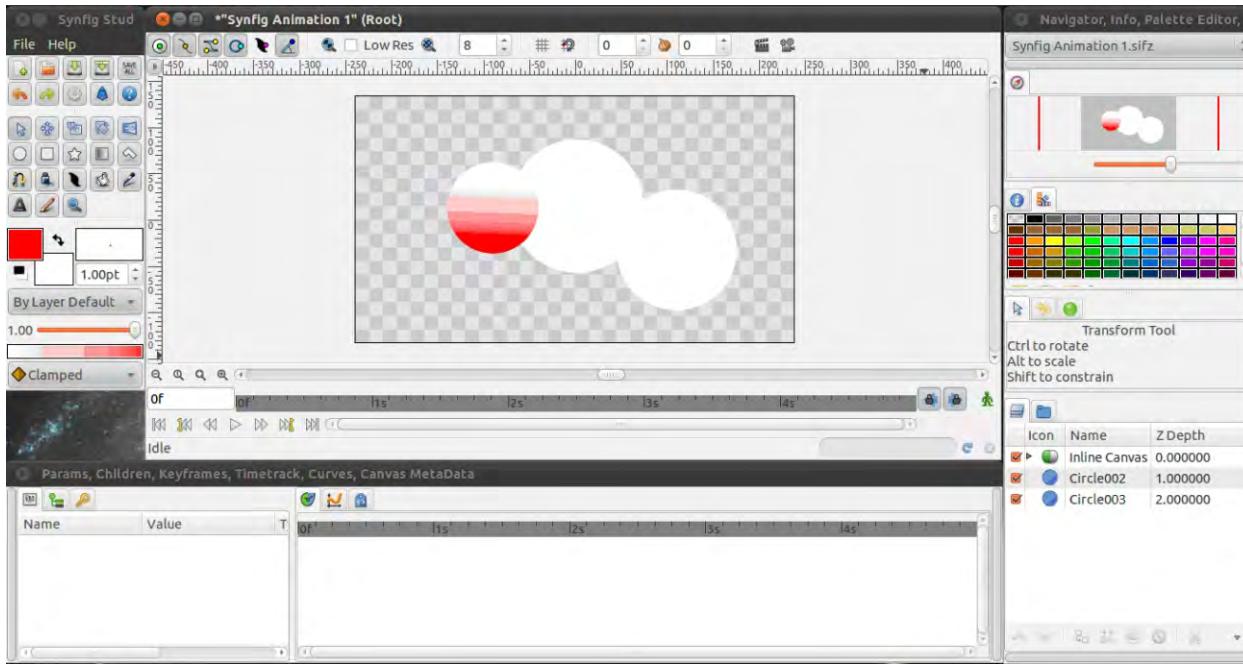


Figure 4.8 : Create a new circle layer

Now drag this new circle layer into the inline canvas layer. We place the circle layer just below the gradient layer and the effect will be as shown in figure 4.9. Here you can see the gradient is applied to all the objects inside the encapsulated area.

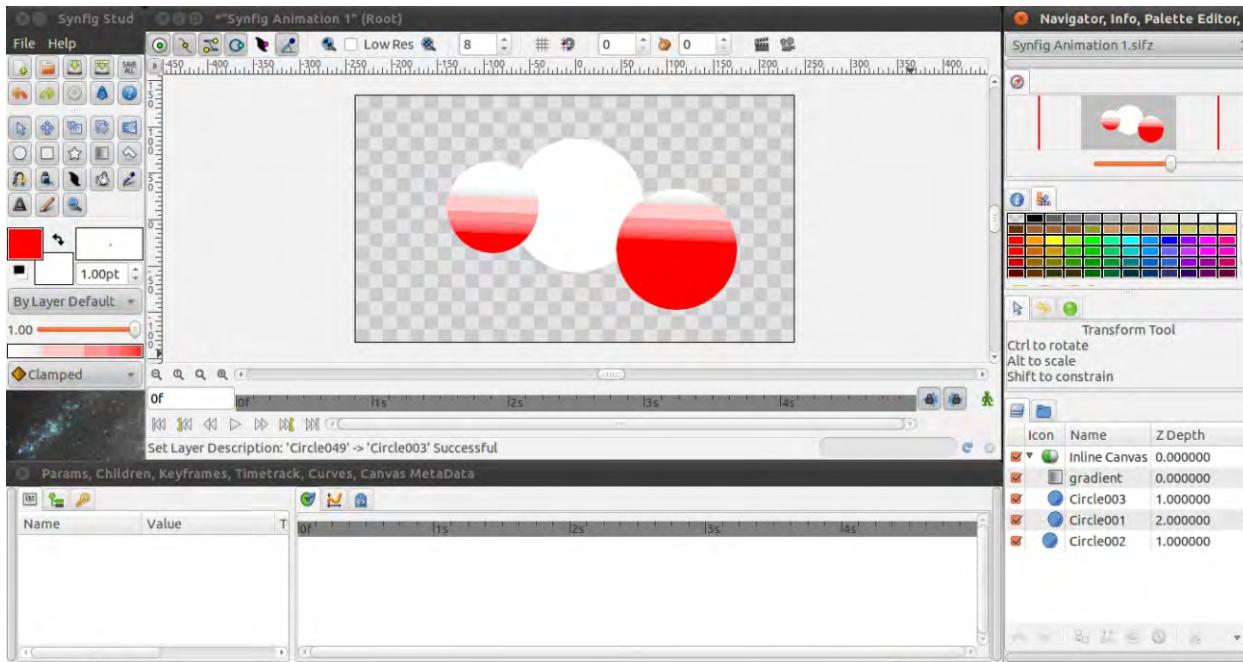


Figure 4.9 : New circle layer placed below the gradient layer

If we reposition the gradient layer using the lower layer button as shown in figure 4.10 (a), the gradient will now be applicable to only one of the circle layer. In other words, we can say that the gradient layer has an effect on only the layers that are below it in the encapsulated area. Figure 4.10 (b) shows the effect of this repositioning.

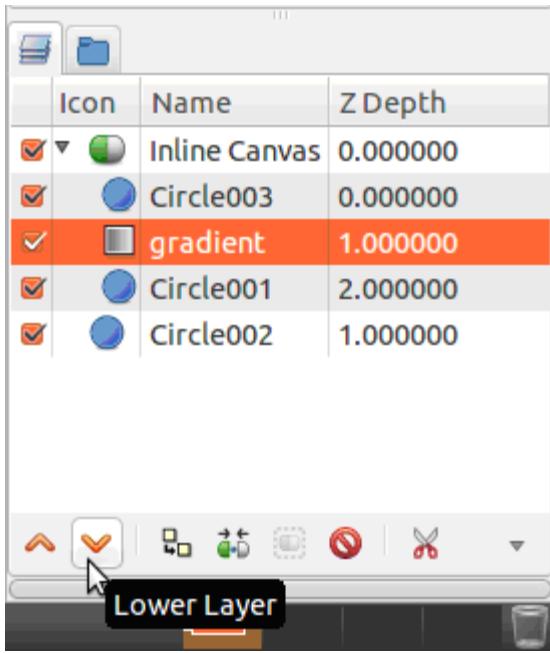


Figure 4.10(a) :

Reposition gradient layer

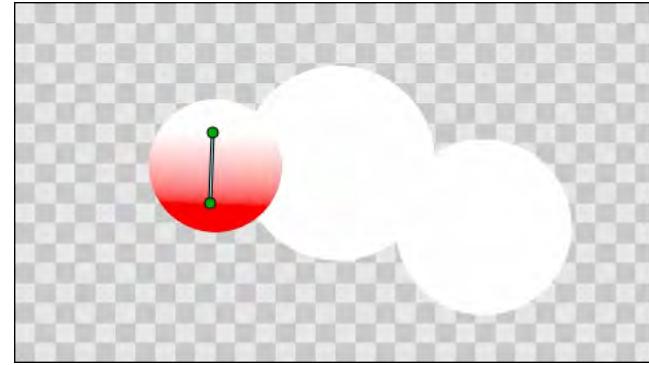


Figure 4.10(b) :

Effect of repositioning gradient layer

The important point to remember here is that a layer can only modify the data that it gets from the layer directly below it. For example if we used a Blur Layer on top of the layers inside the inline canvas, it would just blur the inline canvas layers and not every layer that lies under the inline canvas.

Rotation on a specified path

Using the concept of encapsulation, let us discuss another example of a ball rotating around a particular path. In our animation example of the circle moving from left to right described in previous chapter, we defined the starting point and the ending point. Here, Synfig automatically assumes the path to be straight. But in our example we want to move the object as per our defined path. Follow the steps mentioned to create user defined path.

- Create a new file.
- Let us first create our path for rotation using the Bline tool. Select the Bline tool and draw an oval shaped path as shown in the figure 4.11. In the Tool options panel select only the *Create Outline Bline* option.

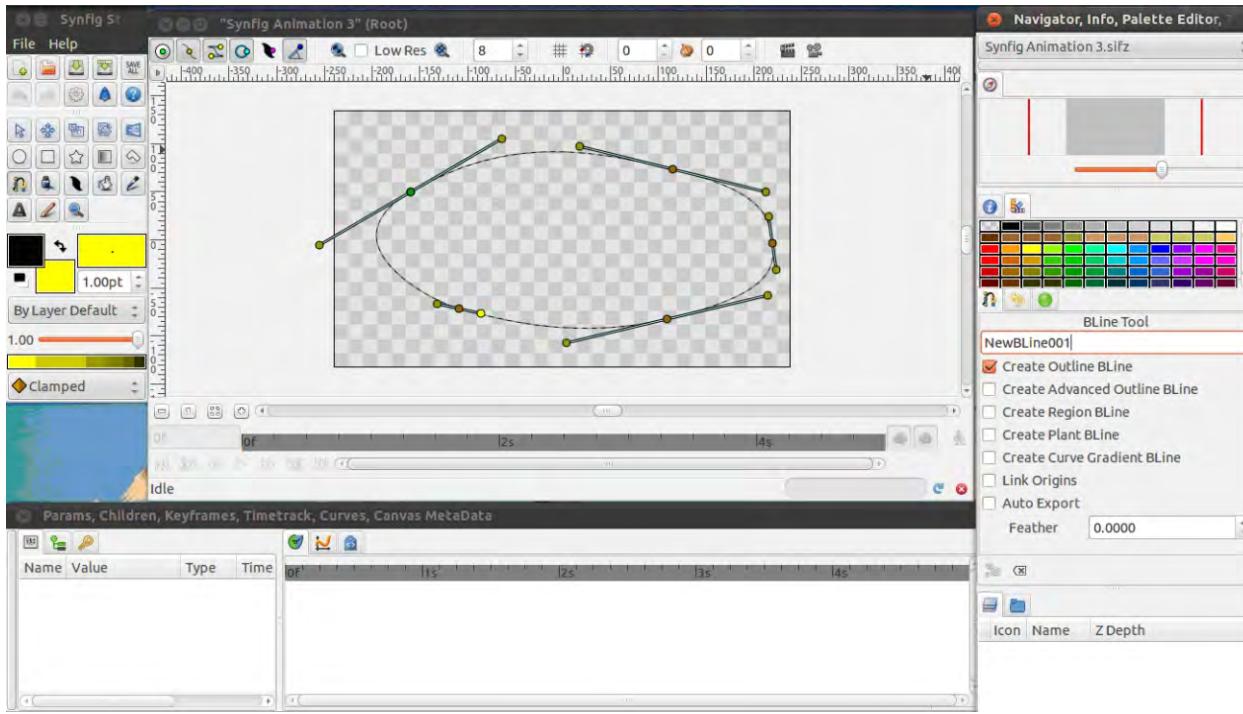


Figure 4.11 : Creation of path using Bline tool

- Select the transform tool to finish the Bline. (Pressing Esc will clear the Bline).
- Now select the circle tool. Using the palette editor select a color of your choice and draw a circle on the canvas as shown in figure 4.12.

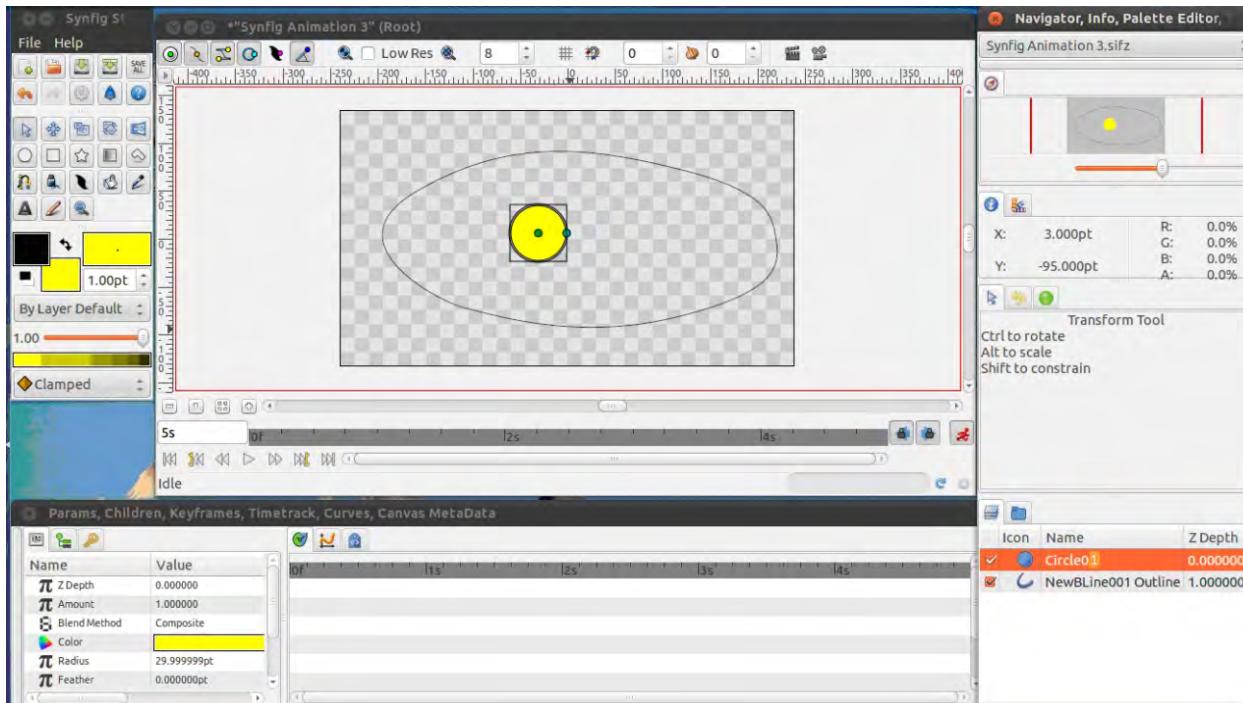


Figure 4.12 : Create the circle

- Let us add a new rotate layer above the circle layer. In the layers panel select the circle layer → Right click → New layer → Transform → Rotate. This will add a new layer named rotate above the circle layer as shown in figure 4.13 (a) and 4.13 (b).

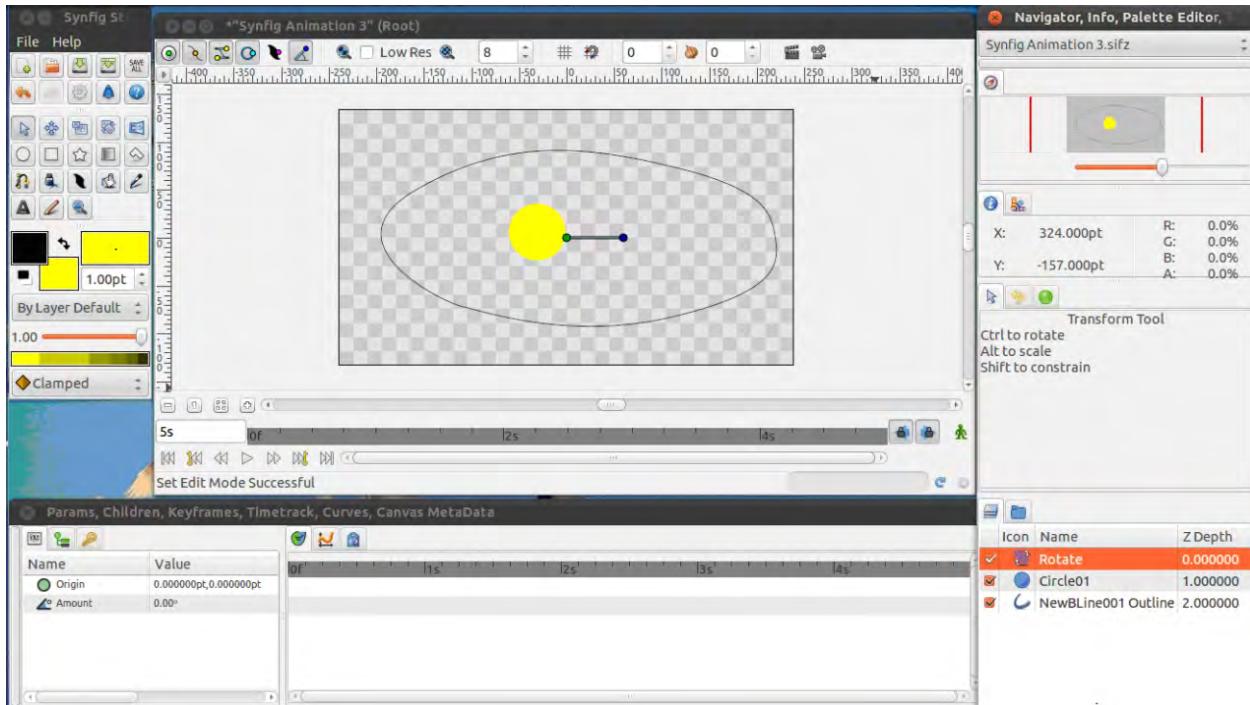


Figure 4.13(a) : New rotate layer above the circle layer

Icon	Name	Z Depth
✓	Rotate	0.000000
✓	Circle01	1.000000
✓	NewBLine001 Outline	2.000000

Figure 4.13(b) : Layers panel showing rotate layer above the circle layer

As we need to rotate the circle, we have added a rotate layer above the circle layer.

- Now we will encapsulate the circle layer and rotate layer. To select both the layers, click on the rotate layer in the layers panel, press shift and select circle layer.
- Right click and select encapsulate. Thus the rotate and the circle layer are encapsulated into a layer

Icon	Name	Z Depth
✓ ▾	Inline Canvas	0.000000
✓	Rotate	0.000000
✓	Circle01	1.000000
✓	NewBLine001 Outline	1.000000

Figure 4.14 : Encapsulate rotate and circle layer

named inline canvas. You can change the name of the encapsulated layer. Figure 4.14 shows the encapsulated layer ‘Inline Canvas’.

- Select the rotate and the circle layer and bring their positioning duck (green duck) near to one another as shown in figure 4.15.

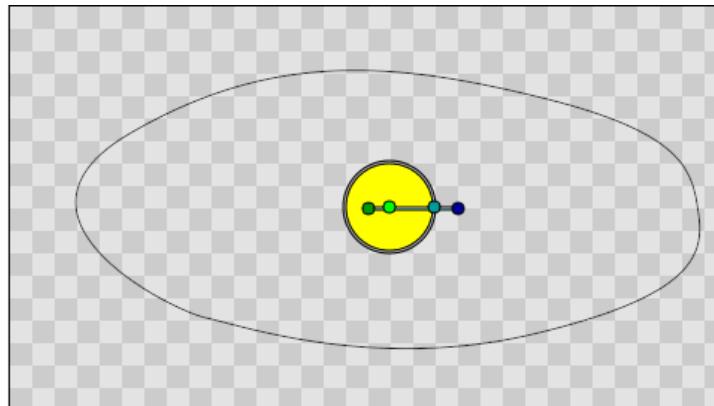


Figure 4.15 : Dragging the green ducks near to one another

- Select the inline canvas layer and select its positioning duck (green duck). Hold the CTRL key and choose the rotate layer. Continue to hold the CTRL key and select the blue duck on the rotation layer. Continue to hold the CTRL key and select the Bline layer. Now right click on the Bline in the canvas where you want the circle to start from. Select the option ‘Link to Bline’.
- Link to Bline feature is used to attach a vertex of an object to the vertex of a Bline. The vertex gets stuck to the Bline vertex and can be moved along with it. Figure 4.16 shows the effect of using link to bline option.

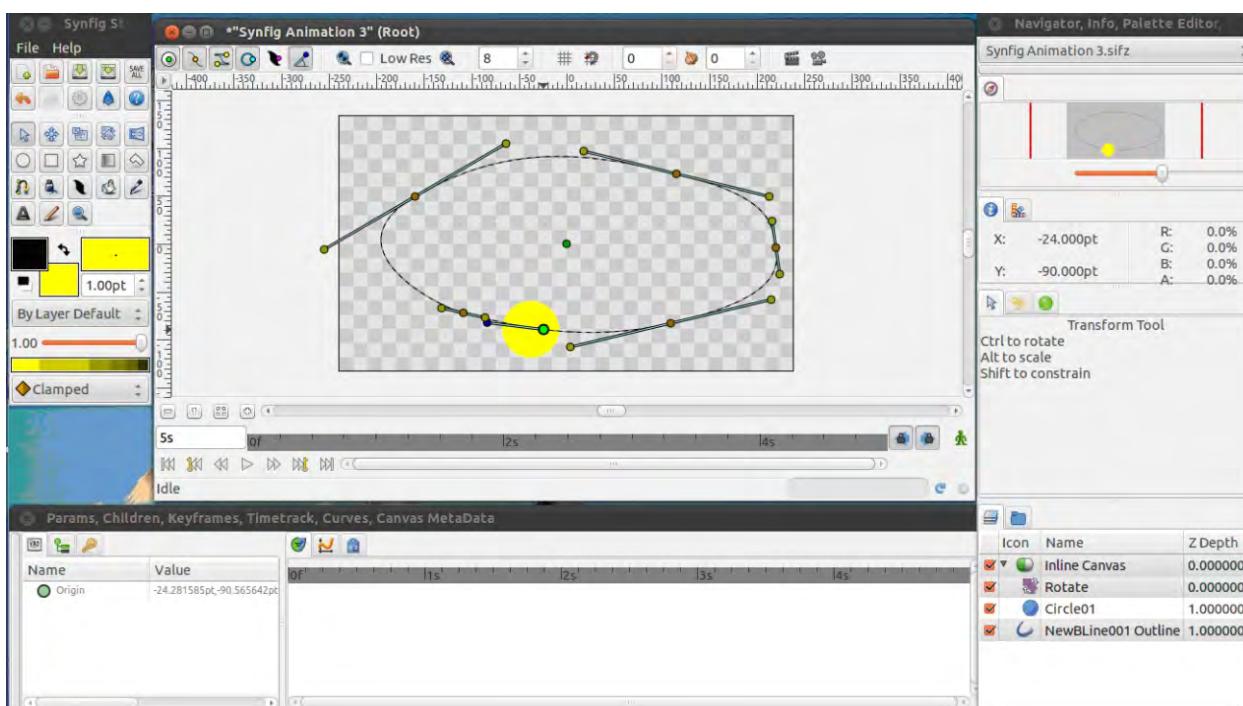


Figure 4.16 : Link to Bline option selected

- Click on the small triangle near the inline canvas layer to close it. Select the inline canvas layer as shown in figure 4.17.

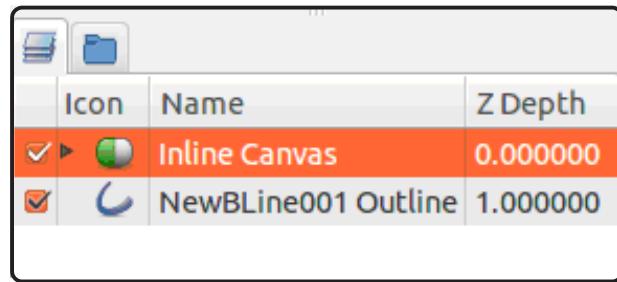


Figure 4.17 : Select Inline canvas layer

- Turn on the animation editing mode.
- In figure 4.18 you can see the red line around the canvas indicating that the animation mode is on. Drag the circle a little bit using the positioning duck (green) to record the starting position on the “0f” mark in the timeline as in figure 4.18. You can see a waypoint is added at “0f” mark.

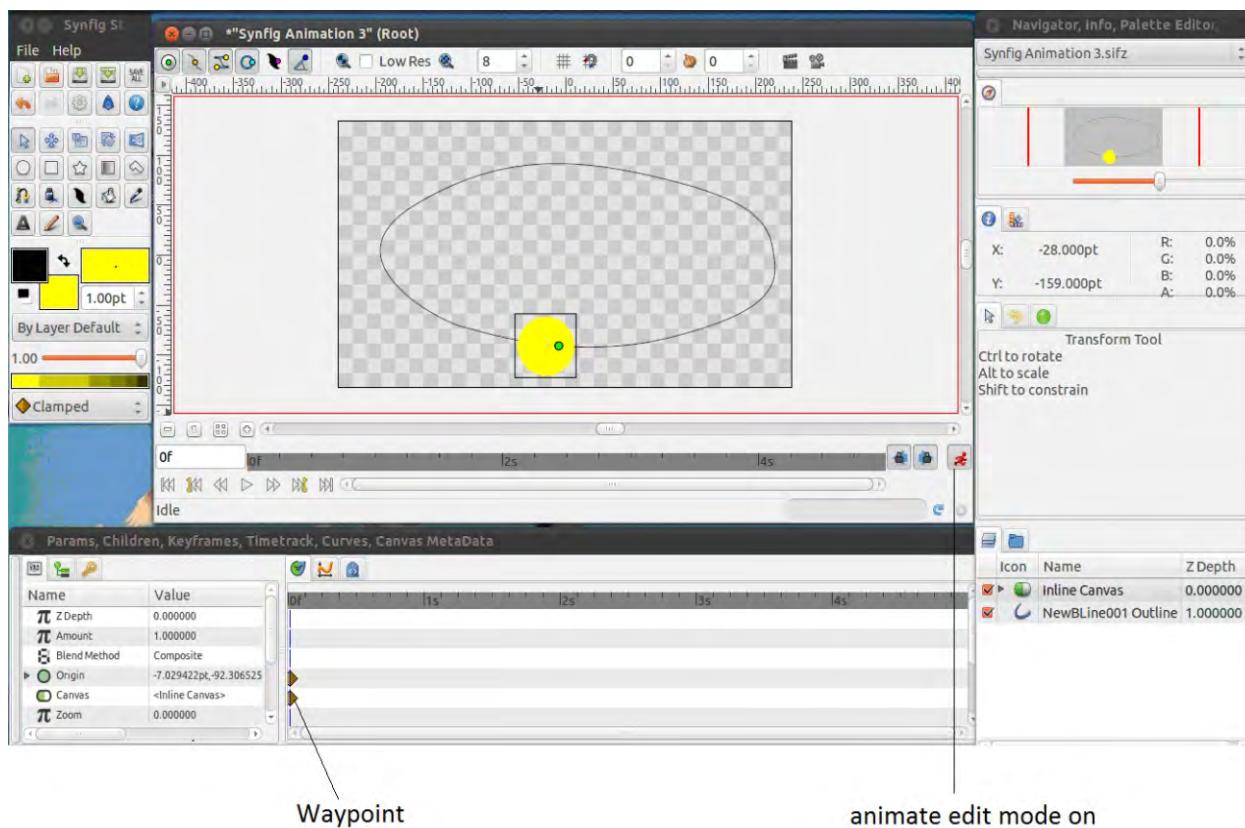


Figure 4.18 : Recording the position at 0s in the timeline

- Now click on 5s mark in the timeline and drag the circle in anticlockwise direction using the positioning duck (green duck) till you reach the ending point of the bline.

Figure 4.19 (a), (b) (c), (d), (e) and (f) shows the circle dragged from starting point to ending point. The timeline in each image shows the position of the circle at that particular time.

Note : Do not drag the circle directly to the end point by moving in clockwise direction. This will not record properly the position of the circle at different timing locations.

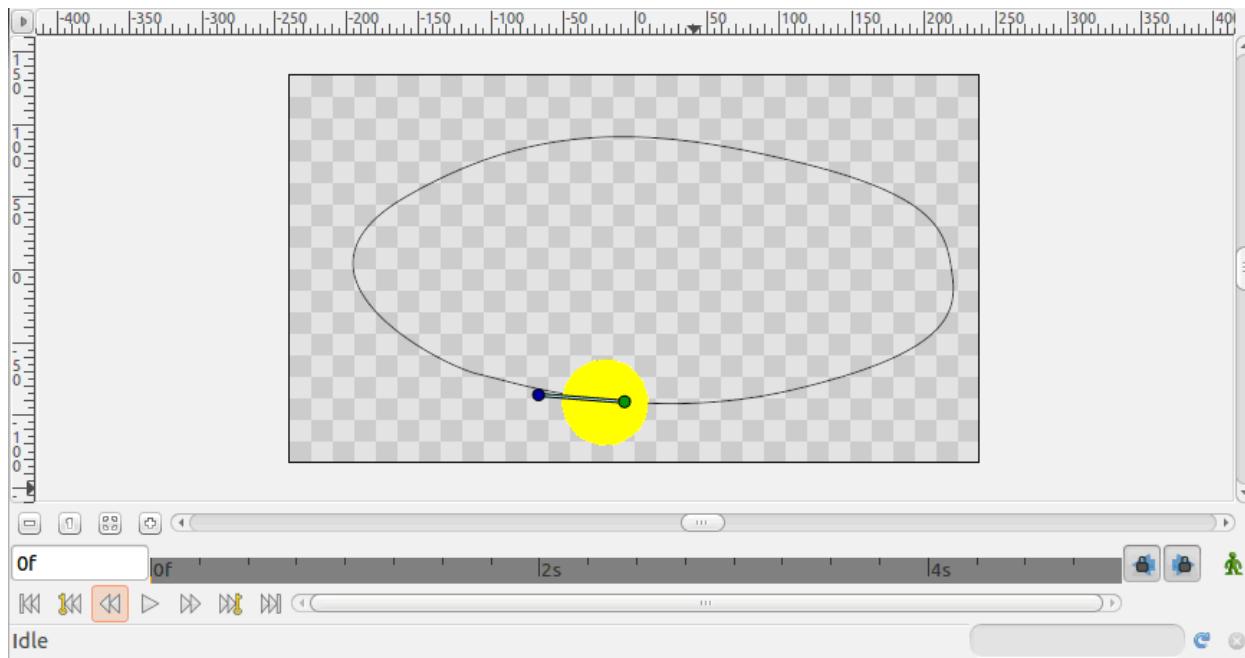


Figure 4.19(a) : Position of circle at 0f

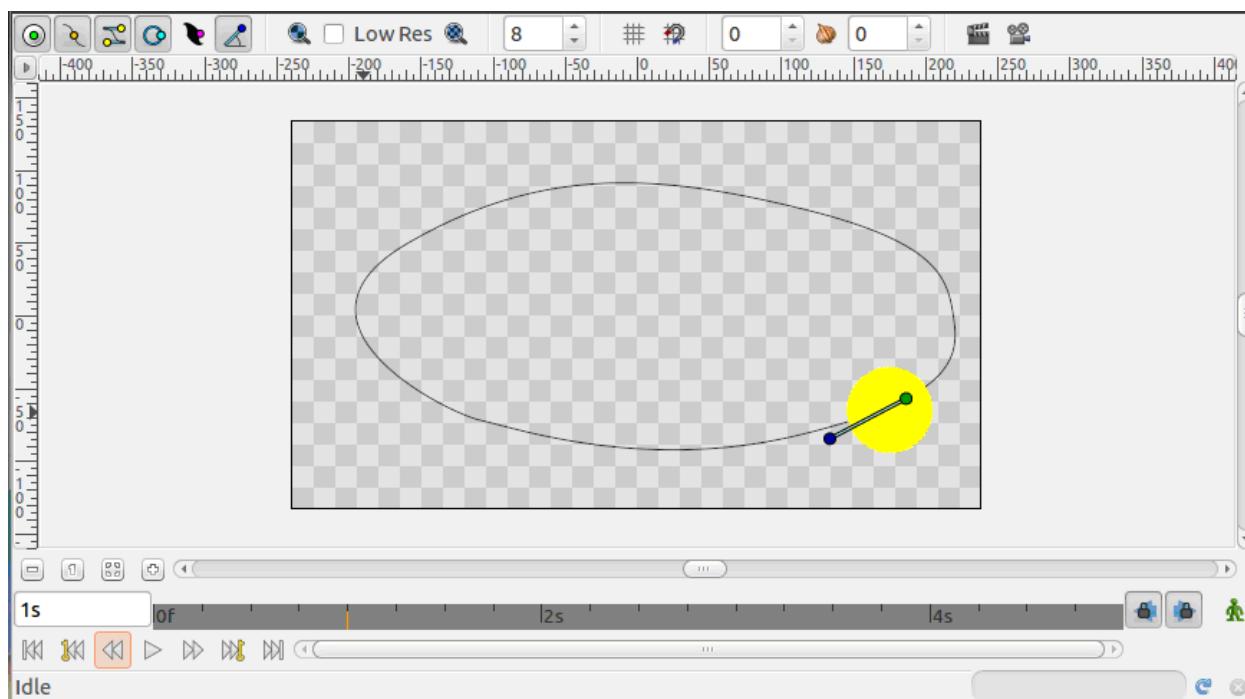


Figure 4.19(b) : Position of circle at 1s

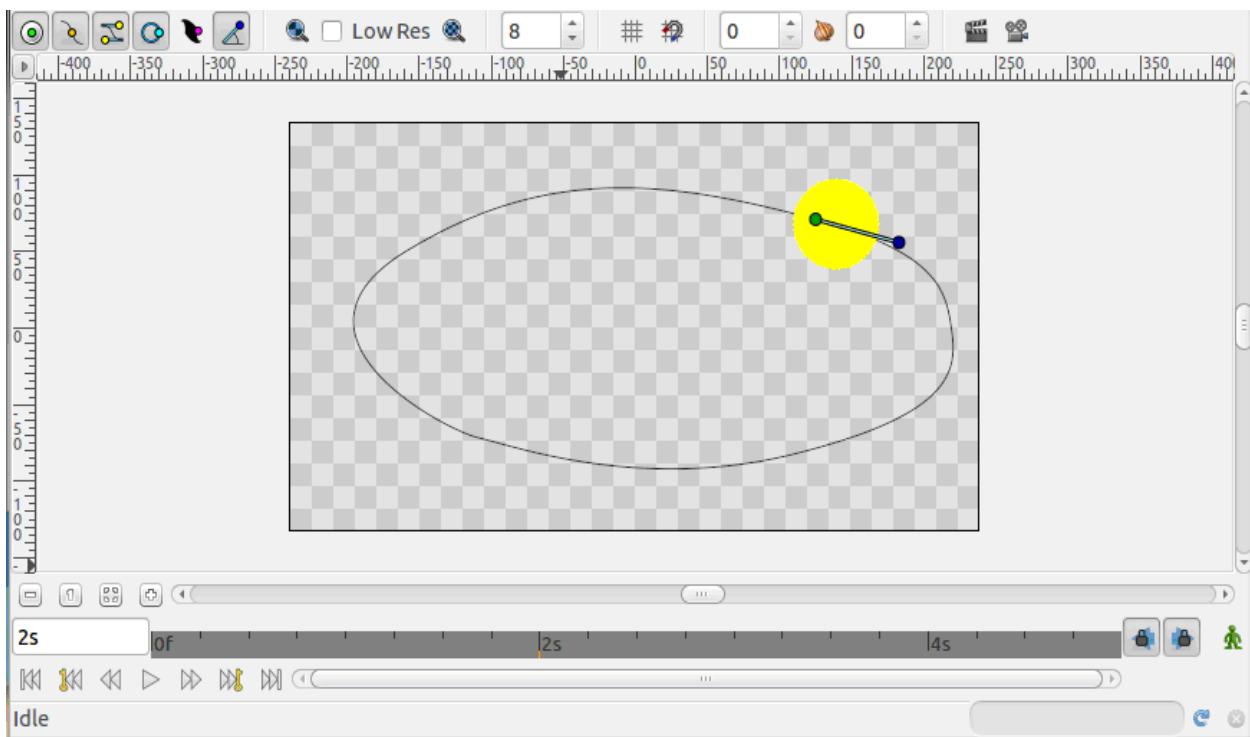


Figure 4.19(c) : Position of circle at 2s

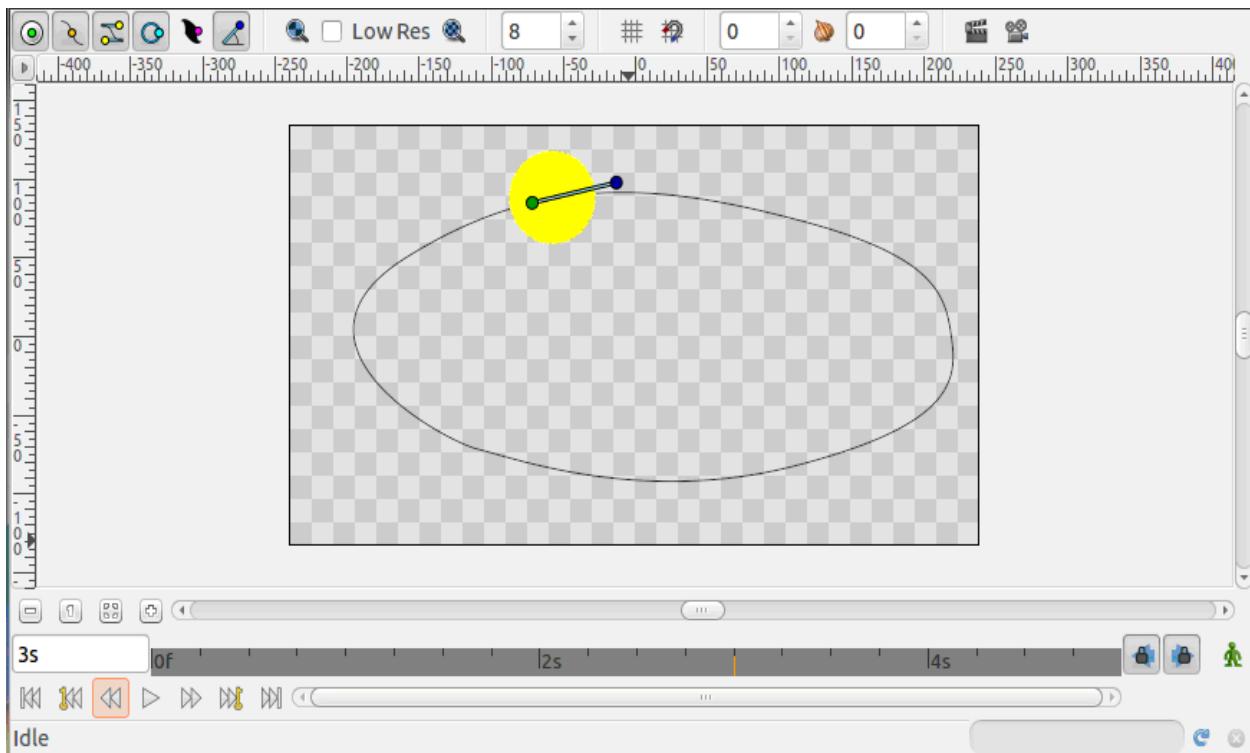


Figure 4.19(d) : Position of circle at 3s

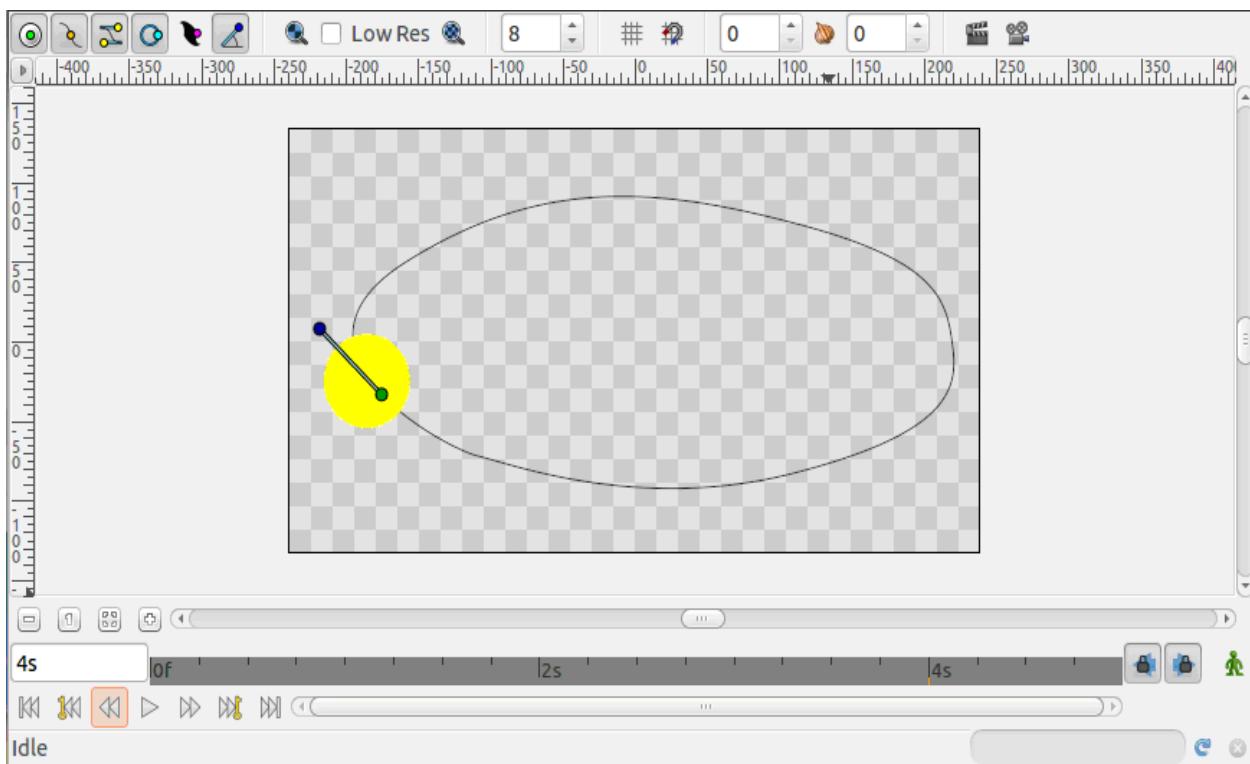


Figure 4.19(e) : Position of circle at 4s

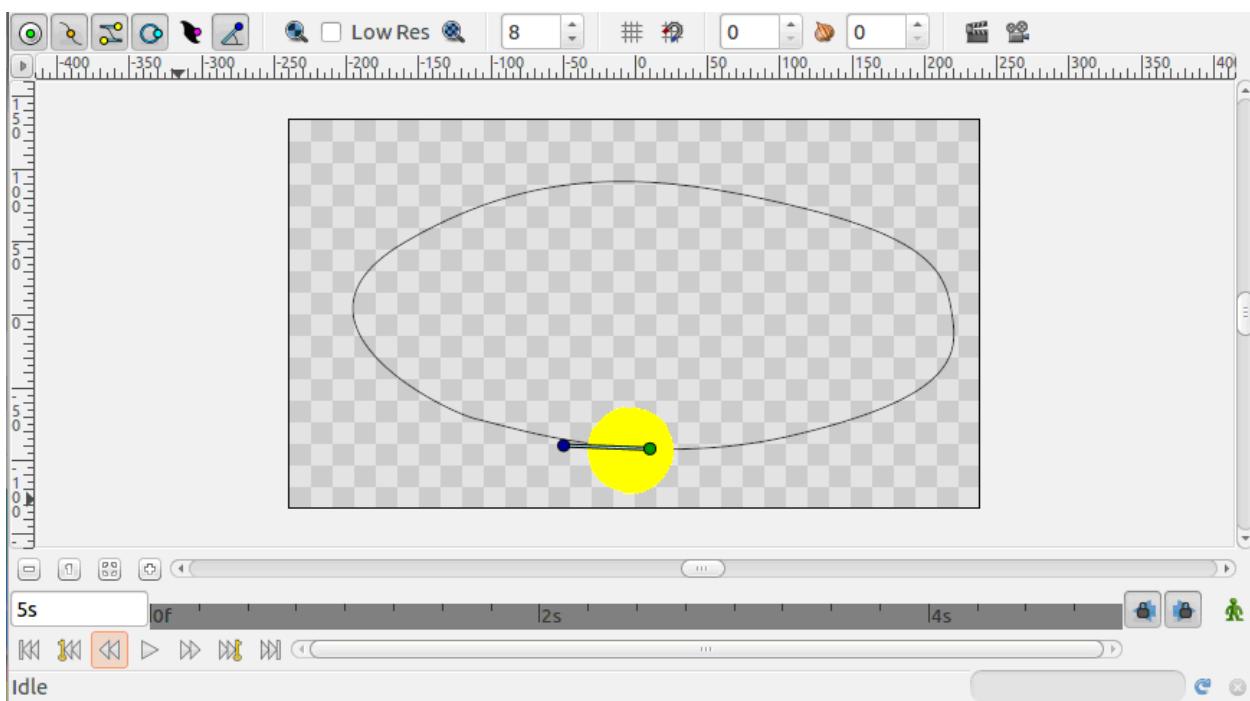


Figure 4.19(f) : Position of circle at 5s

- Click on play to preview the animation. You can see the circle moving in anticlockwise direction.
- Save and render the file. Figure 4.20 show the output as viewed using a web browser.

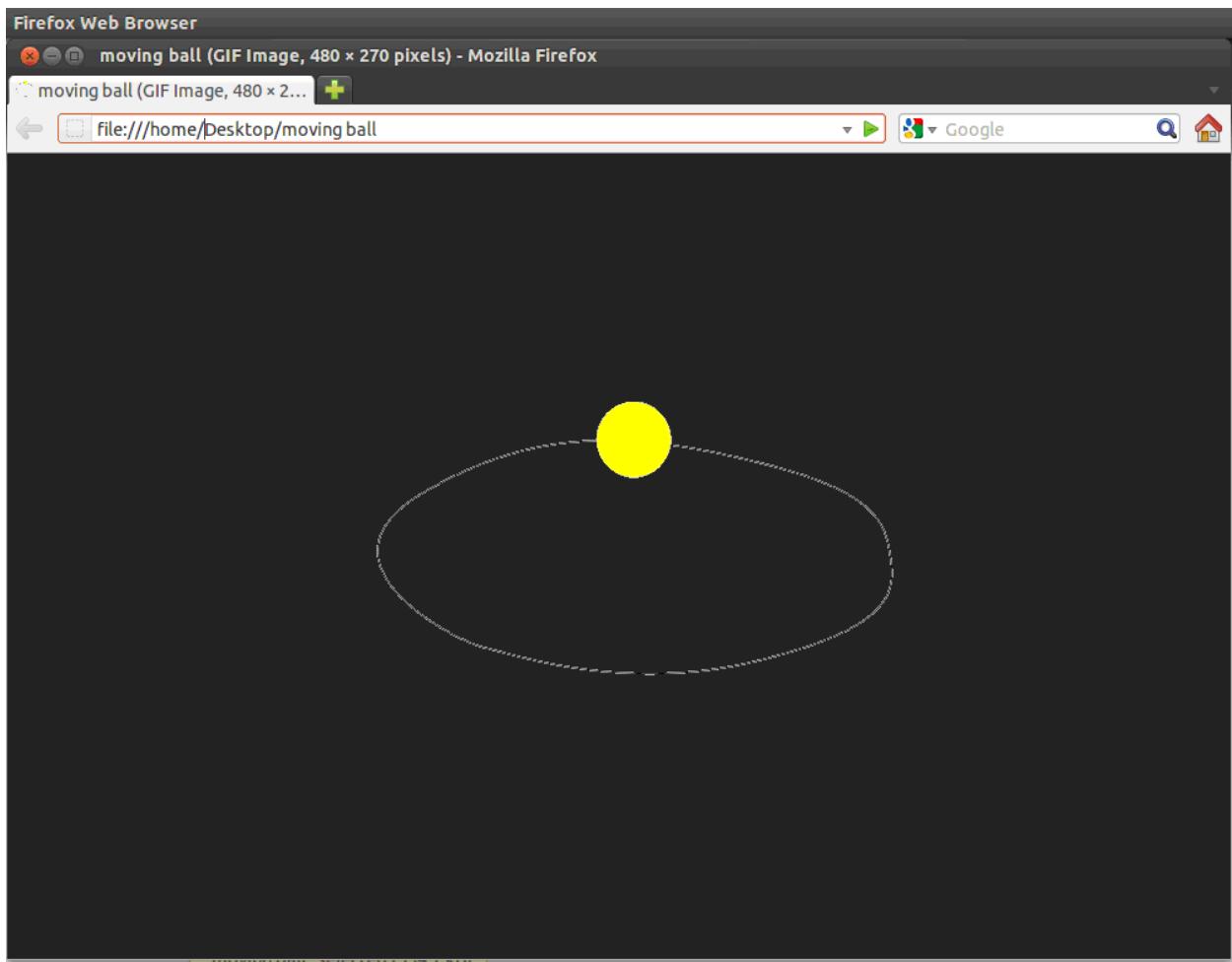


Figure 4.20 : Output displayed in web browser

Summary

In this chapter we introduced the concept of layers and how we can place our objects above or below another objects. Layers give us the freedom to work with each object individually. We introduced an important concept called “Encapsulation”. Using encapsulation, we can give effects to various objects without changing other objects in the animation. We also discussed how to animate objects on a user defined path.

EXERCISE

1. Explain Layers. Give an example.
2. What is Encapsulation? Why do we need it?
3. Imagine a street view with objects such as building, car, sun, traffic light and person. State which layer will be above and below?
4. What is the purpose of Link to Bline option?

5. Choose the most appropriate option from those given below :

(11) Which of the following terms represent the different levels on which we place our objects stacked above and below each other?

- (a) layer
 - (b) frame
 - (c) parameter
 - (d) panels

(12) Gradient, blur, distortion and filter are types of which of the following components?

- (a) frames
 - (b) layer
 - (c) keyframes
 - (d) panels

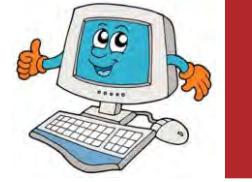
LABORATORY EXERCISES

1. Draw a rectangle, circle and square overlapping each other. Select different tool options and different color for each object. Also give a background color to the canvas.
 2. Create a star and a circle layer. Apply gradient on the star layer. Create a square and apply blur effect on it. Now create a rectangle and also apply the blur effect. Note: the objects should overlap each other.
 3. On the canvas using gradient show the sky and ground. Use different colored gradient for both. For sky use the shades of blue color as gradient and for ground use the shades of green color as gradient.
 4. Draw two circles overlapping each other and apply blur effect to one of them.
 5. Create an animation showing a star moving on the user defined path.
 6. Create an animation showing a star rotating while moving from left to right.
 7. Create an animation to show a kite flying in the sky.



5

Using Pictures in Synfig



Pictures when used in a presentation or animation improve the quality of reception. In Synfig, we can insert images to give a different look to our animation. We can also work with these images and adjust them as per our requirement just the same way as we work with the objects drawn using the tools. In this chapter, we will see how to work with pictures.

Inserting Image

Let us try to create a layer that contains image in it. To import an image onto a canvas perform the steps mentioned.

- Create a new file.
- Select File → Import. Alternatively, you can also press CTRL + i. This combination is the shortcut key to import image. This will open a dialog box showing the folders from where you want to import the image as shown in figure 5.1. Select the image that you would like to import and press open. The image will be imported on the canvas. Figure 5.2 shows the image imported to the canvas by us. Note that our screen may vary depending on the image that you choose to import.

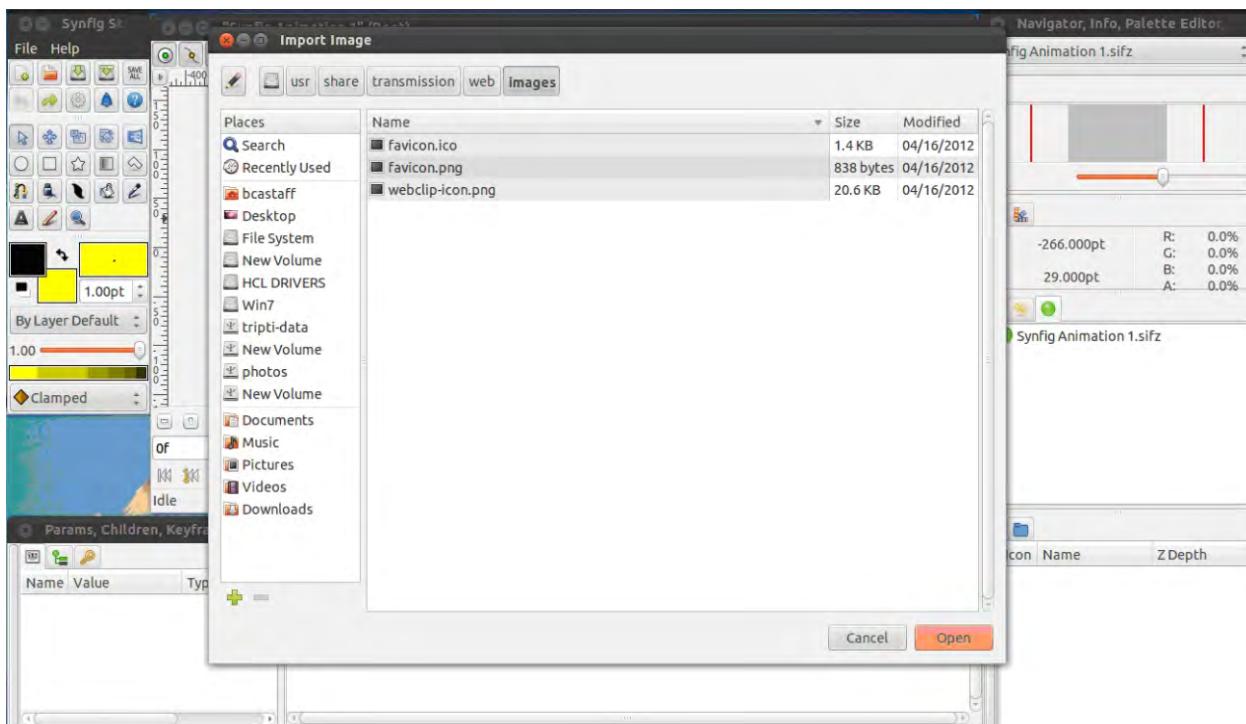


Figure 5.1 : Import image dialog box

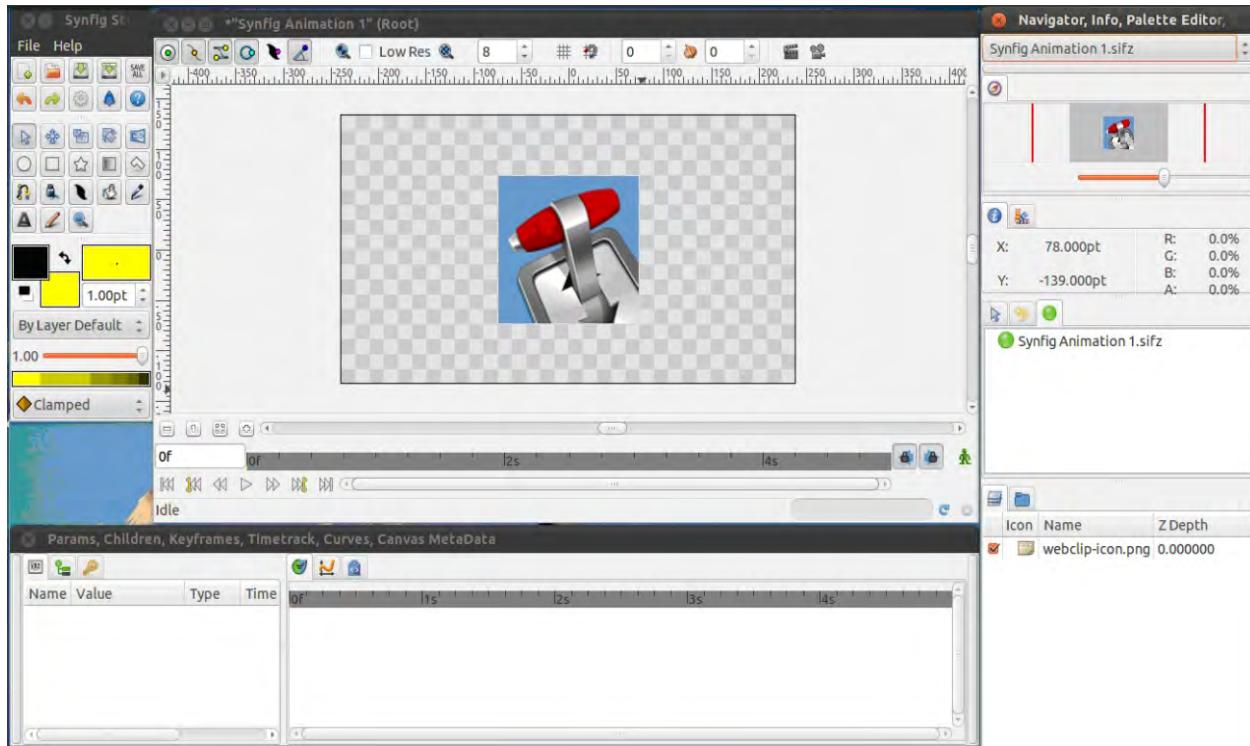


Figure 5.2 : Image inserted on the canvas

- To resize the imported image, select the image layer in the layers panel. As shown in figure 5.3 you can see two green points on the image. These can be now used to change the size of the image. Figure 5.4 shows the resized image when the green dots are moved.

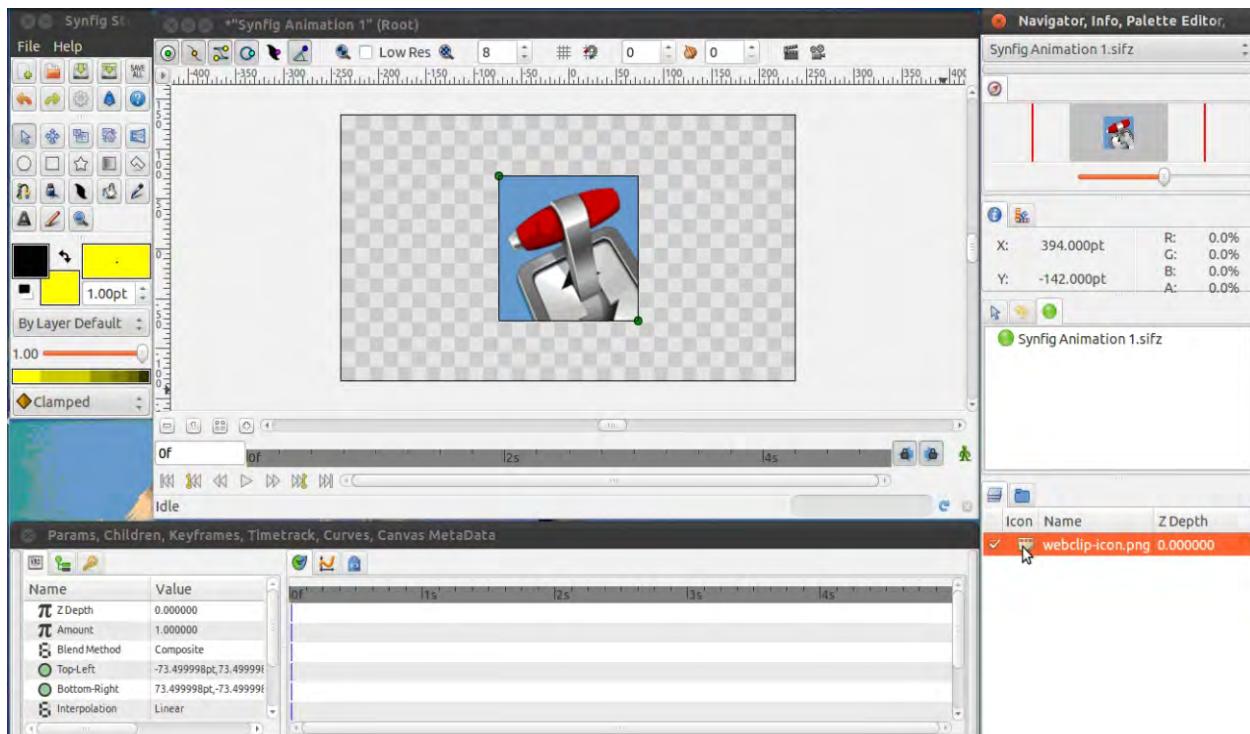


Figure 5.3 : Green points to resize the image



Figure 5.4 : Image resized using green points

- Observe that whenever we resize, the image tends to get distorted. This happens as we have not maintained the aspect ratio. If we want to resize the image and keep its aspect ratio we need to encapsulate the layer.
- Right click on the image layer and select encapsulate. This will add an inline canvas layer. Open up the encapsulated layer using the small triangle and you can see the image layer inside the inline canvas as shown in figure 5.5.

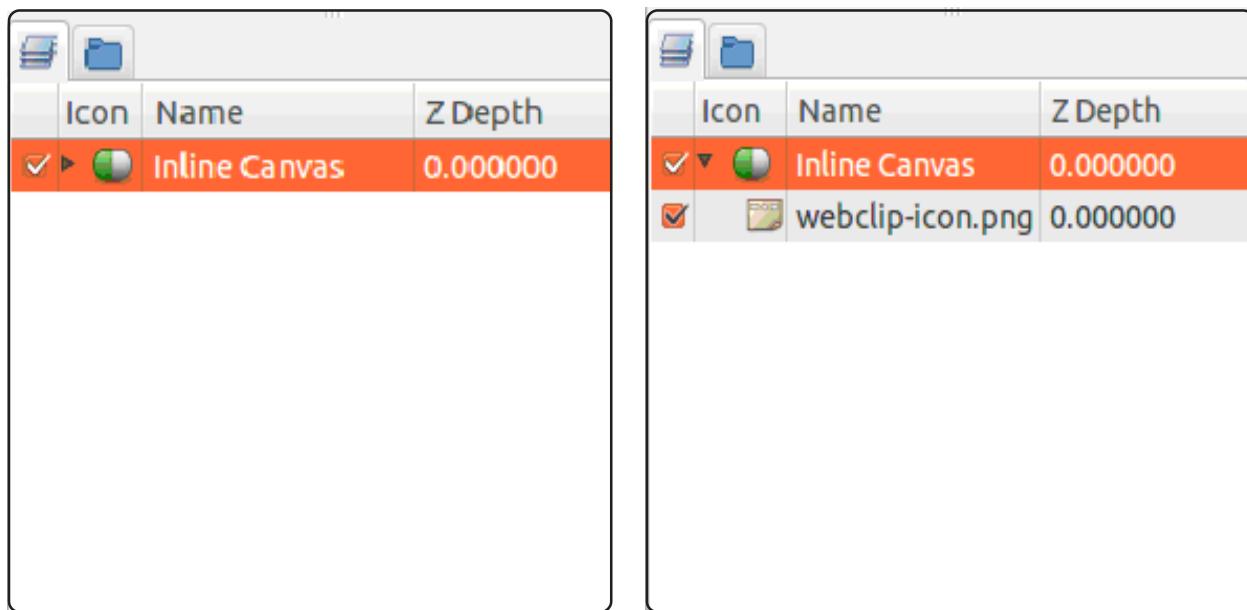


Figure 5.5 : Encapsulated image layer

- Now we need to add a new scale layer above the image layer. Select the image layer → Right click → New Layer → Transform → Scale. You can see a scale layer is added to top of the image layer as shown in figure 5.6.

	Icon	Name	Z Depth
<input checked="" type="checkbox"/>		Inline Canvas	0.000000
<input checked="" type="checkbox"/>		Scale	0.000000
<input checked="" type="checkbox"/>		webclip-icon.png	1.000000

Figure 5.6 : Scale layer added above image layer

- Select the scale layer and in the parameters panel change the amount setting from 0 to -1 as shown in figure 5.7. In the canvas you see the change in the image size as shown in figure 5.8.

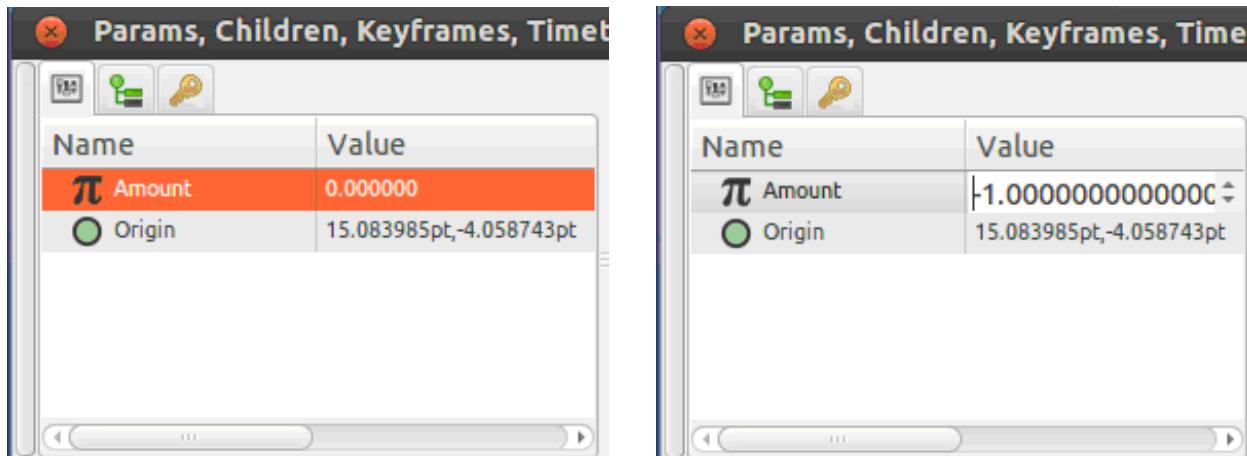


Figure 5.7 : Scale layer amount setting changed from 0 to -1

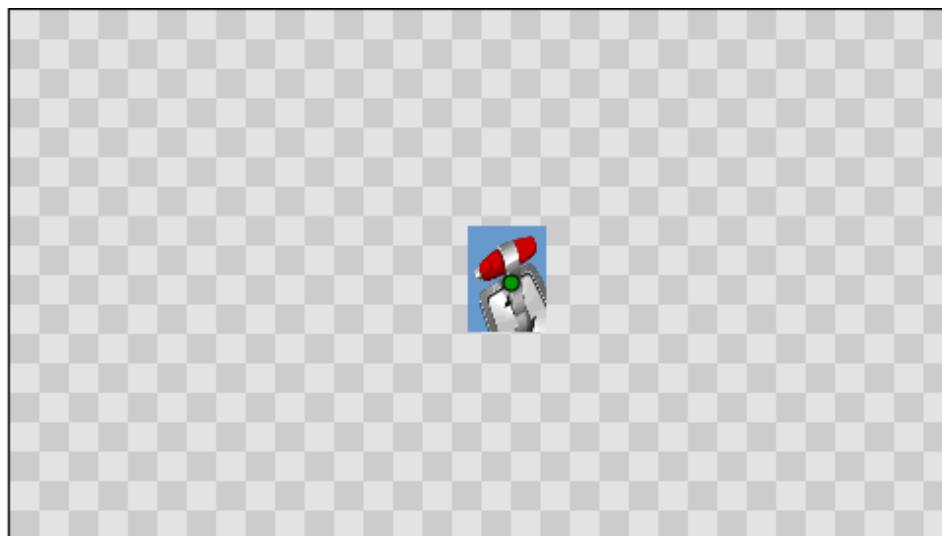


Figure 5.8 : Scaled image

- Now if you want to rotate the image then add a new rotate layer above the scale layer. Select the scale layer → Right click → New Layer → Transform → Rotate. You can see a rotate layer is added to top of the scale and image layer as shown in figure 5.9.

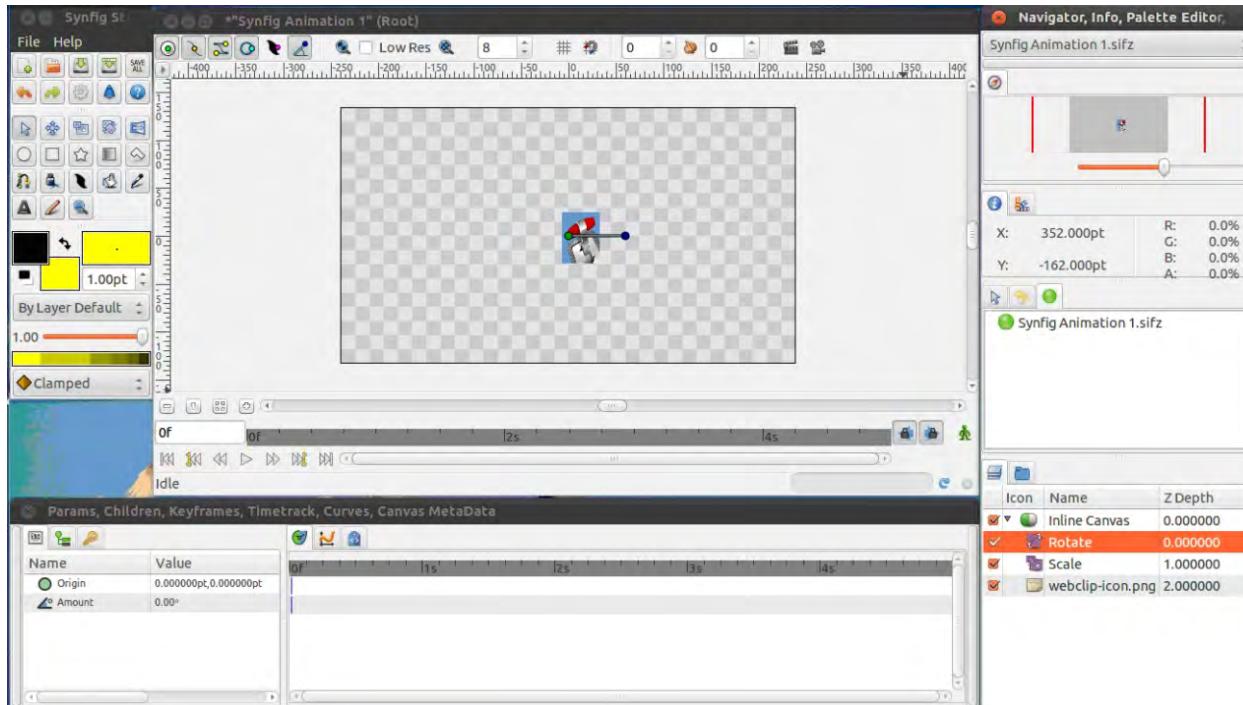


Figure 5.9 : Rotate layer added above scale layer

- By using the rotation duck (blue duck) you can rotate the image as you want. Figure 5.10 shows the rotated image.

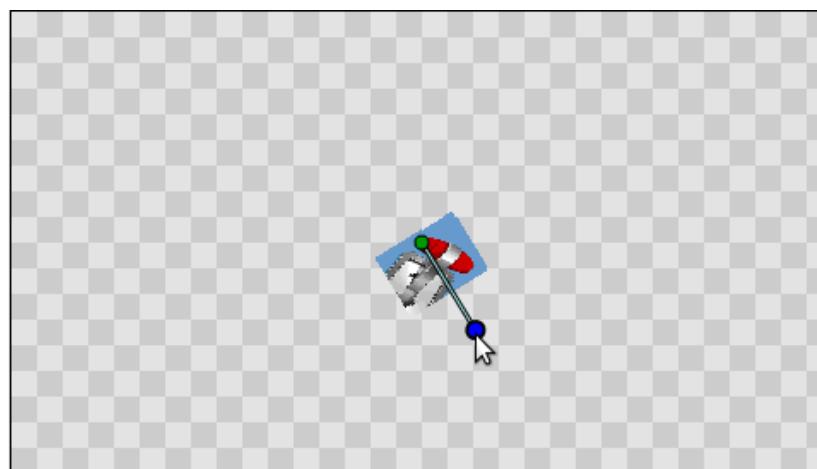


Figure 5.10 : Image rotated using the duck

Observe that once an image has been inserted we can scale, rotate and perform different operations on it. Images when used always enhance the animation.

Masking

Suppose, in your animation you want to show the movement of an object say clouds moving through the window. If the window is drawn using Synfig then you can place the object behind the window and can move the cloud object. But if the window is an image then it becomes difficult to animate the cloud.

Masking is a feature by which you can hide or reveal areas of a layer. Basically, the shape used as a mask acts like a window with the help of which you can see the objects beneath it. Alternatively, it can be used as patch to hide the objects beneath it.

Let us understand by an example and try to show a cloud moving through the window using the concept called masking.

- Create a new file.
- Now import an image on the canvas using **CTRL + i**. Select the picture. And resize it as per the requirement. Figure 5.11 shows the imported image. Note that your screen may look different based on the image you select. You can see the image layer added in the layers panel.

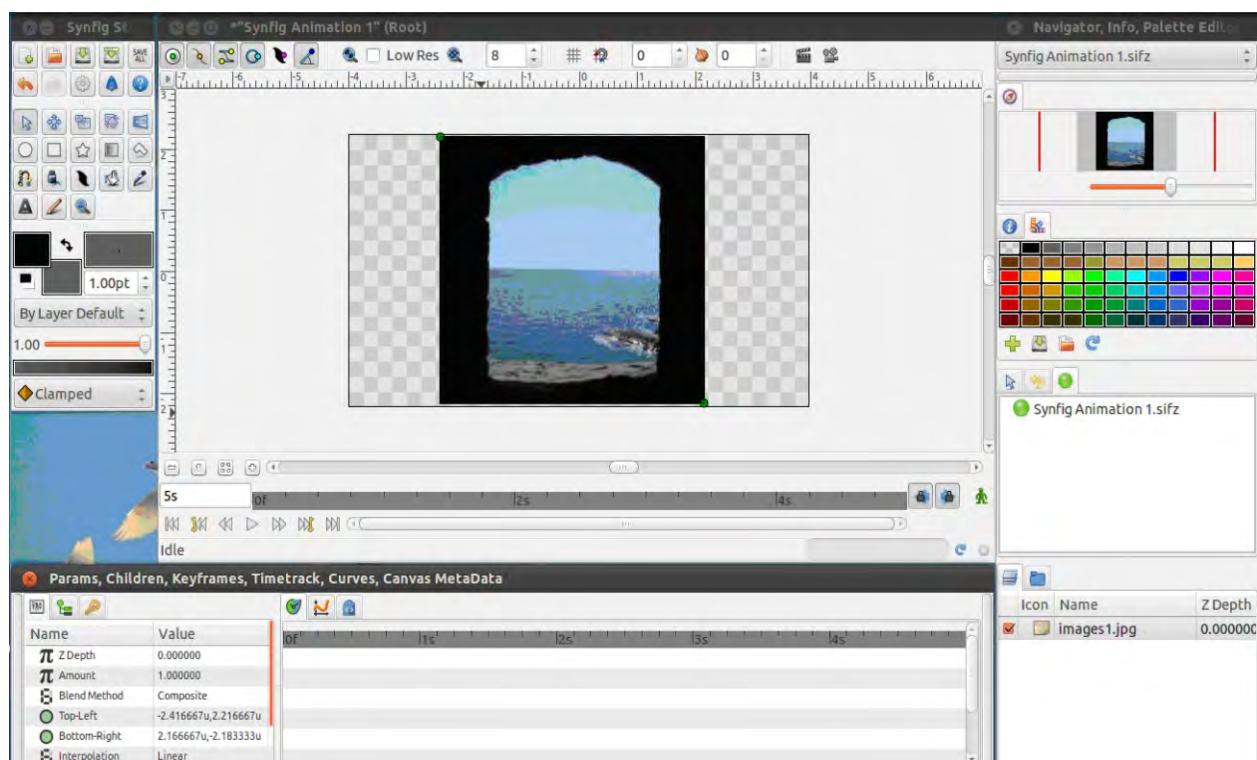


Figure 5.11 : Image imported on canvas

- Now using Bline tool draw a cloud on the left side of the canvas as shown in figure 5.12. You can see the Bline layer on top of the image layer.

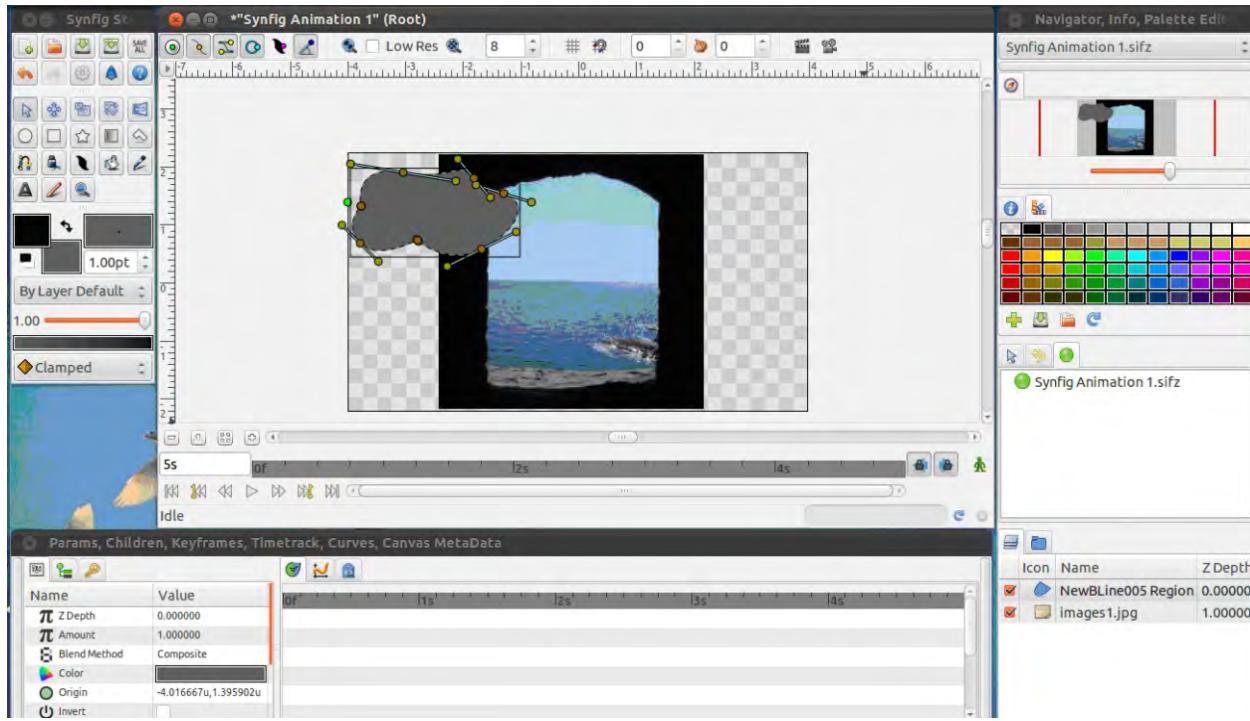


Figure 5.12 : Cloud drawn using Bline tool

- Now turn on the animate edit mode.
- We will create a small animation of cloud moving from left to right of the canvas. Move the cloud a little bit to record the position at 0f in the timeline. Now click on 5s mark on the timeline and drag the cloud towards the right. Figure 5.13 (a) and (b) shows the status of animation at “0f” and “5s” respectively.

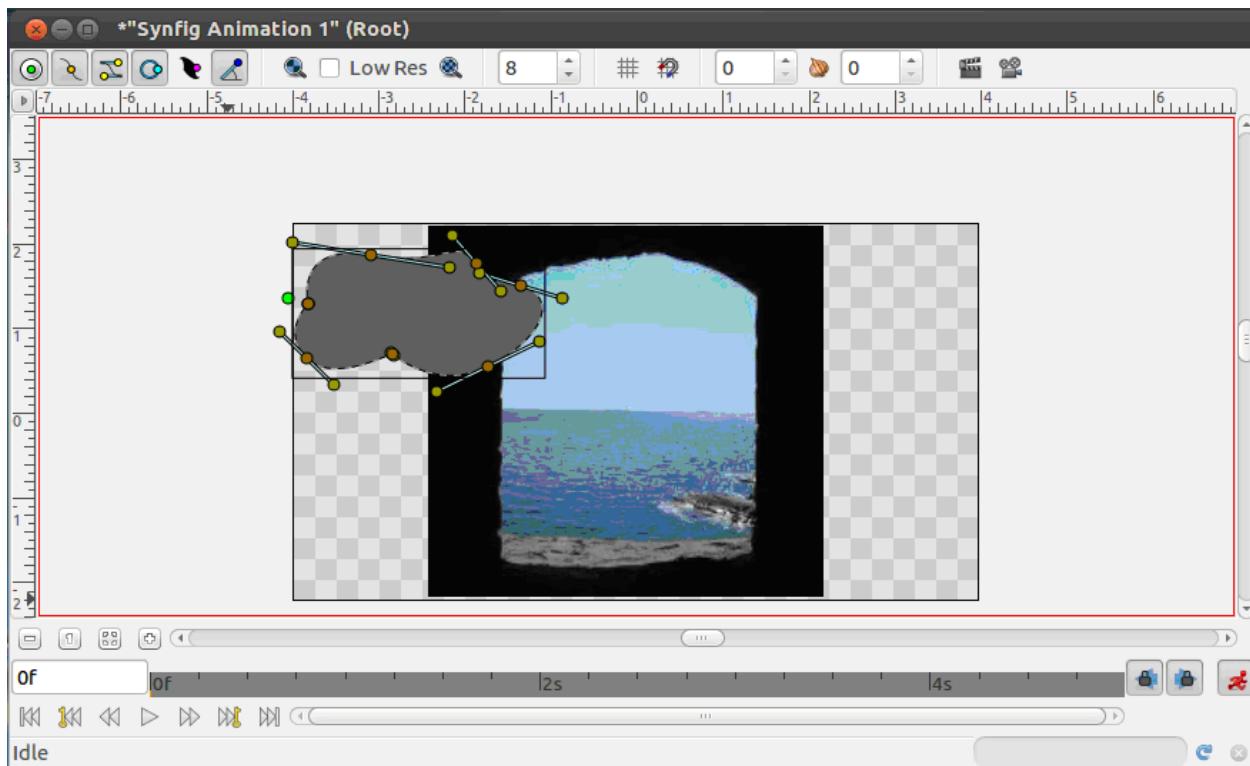


Figure 5.13(a) : Animation of cloud at 0f

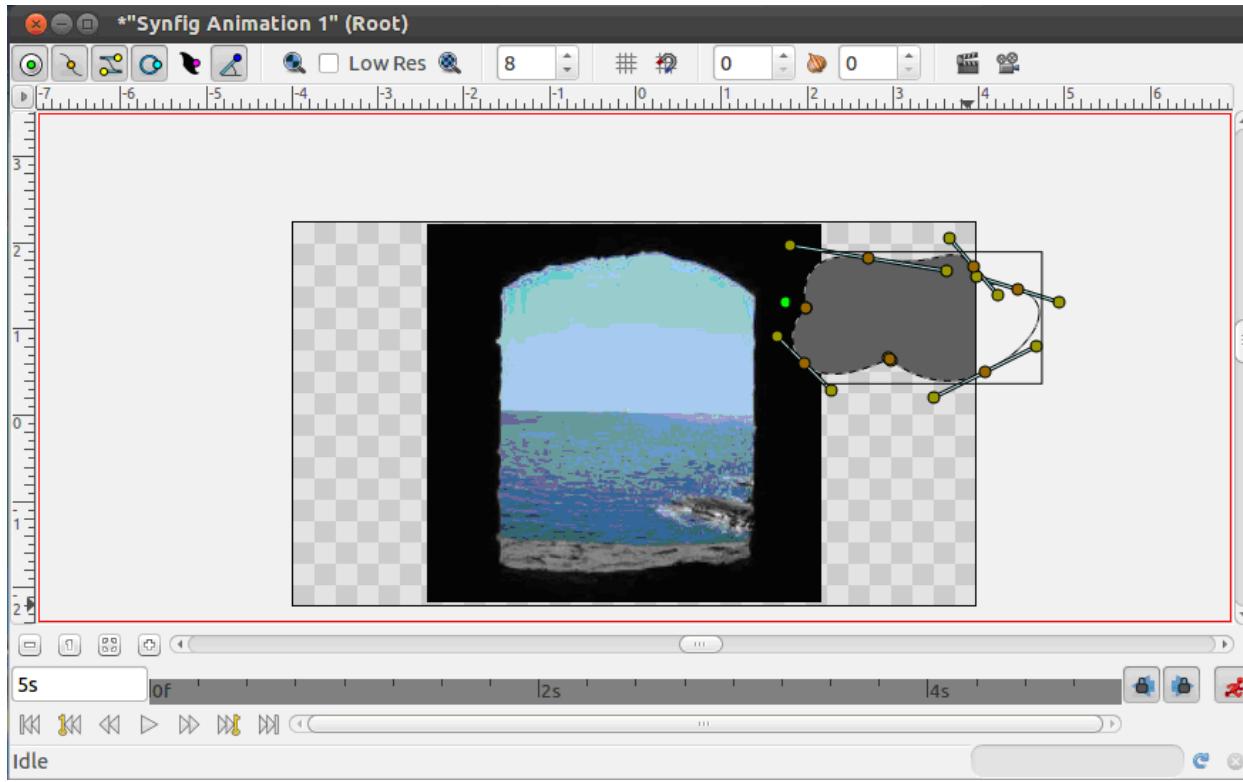


Figure 5.13(b) : Animation of cloud at 5s

- Turn off the animate mode and click on the play button to see the animation.
- You can see the cloud moving in front of image, as the cloud layer (or NewBLine005 layer) is above the image layer shown in figure 5.14(a).
- To change the animation, in the layers panel, drag the cloud layer below the image layer as shown in figure 5.14(b). Click on the play button to see the animation. You can see the difference in the animation. Now the cloud is moving from behind the image.

Icon	Name	Z Depth
	NewBLine005 Region	0.000000
	images1.jpg	1.000000

Figure 5.14(a) : Actual position of layers

Icon	Name	Z Depth
	images1.jpg	0.000000
	NewBLine005 Region	1.000000

Figure 5.14(b) : Modified position of layers

But we want the cloud to move through the window image only. We will do this by using masking. Masking is used to hide or reveal an object.

Let us first use masking for hiding the objects. Here we want the cloud object to be hidden at both the ends of the image.

- First, we need to create a mask shape to hide the cloud on both the sides of the canvas. We will use the Bline tool to create the mask shape. Select the Bline tool and from the tool options select only ‘Create Region Bline’ option. Draw a rectangular shaped Bline as shown in figure 5.15. To end the Bline, at the last vertex right click and select ‘Loop Bline’.

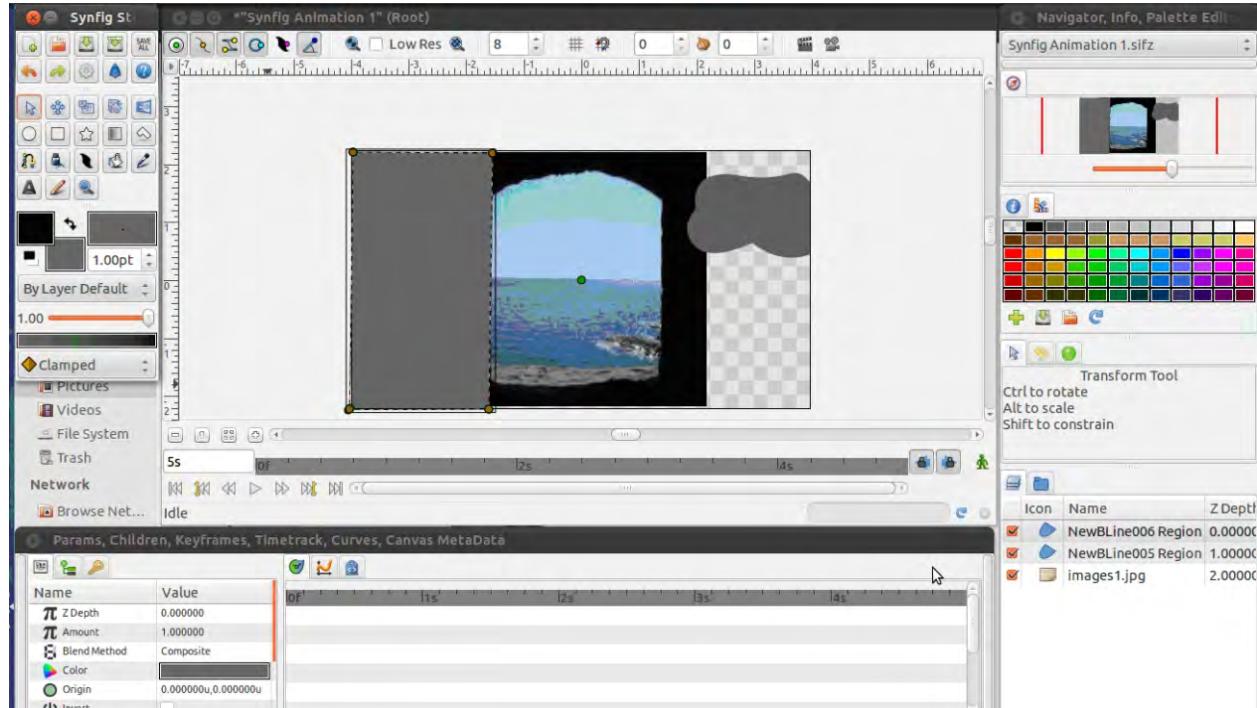


Figure 5.15 : Mask shape created using Bline tool

- Place the mask shape above the elements you want to mask. In our example we will place the mask shape above the cloud layer. Also, as we want to apply mask only on the cloud layer, we will encapsulate the mask shape and the cloud layer. Figure 5.16 shows the encapsulated layer ‘inline canvas’ and also shows the contents of inline canvas layer. We have renamed the layers NewBline005 as cloud layer and NewBline006 as mask layer. Click on the layer name which will allow you to change the name.

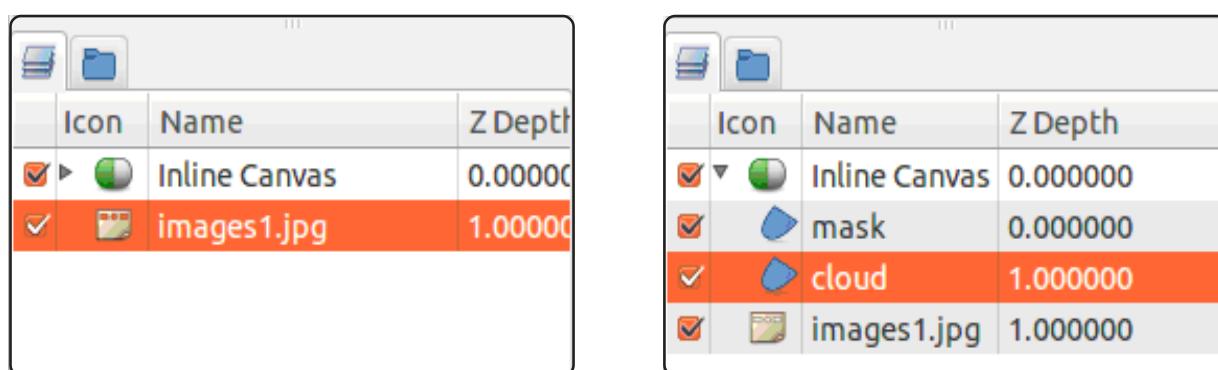


Figure 5.16 : Mask shape and cloud layer encapsulated

- Select the mask layer from the layers panel and in the parameters panel set the blend method to ‘Alpha over’. Figure 5.17 shows the blend method changed to ‘Alpha Over’.

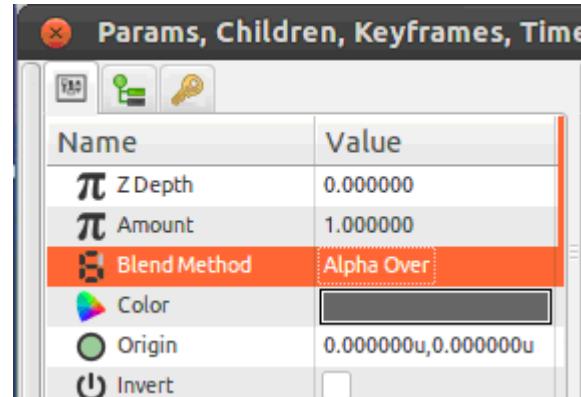


Figure 5.17 : Change the Blend method to ‘Alpha over’

- Play the animation and you will see that the cloud is now not visible on the left side of the canvas.
- We need to do the same thing on the right side of the canvas also. So draw a rectangular mask shape using Bline tool on the right side of the canvas as we did earlier on the left side. Figure 5.18 shows the mask shape on the right side of the canvas.

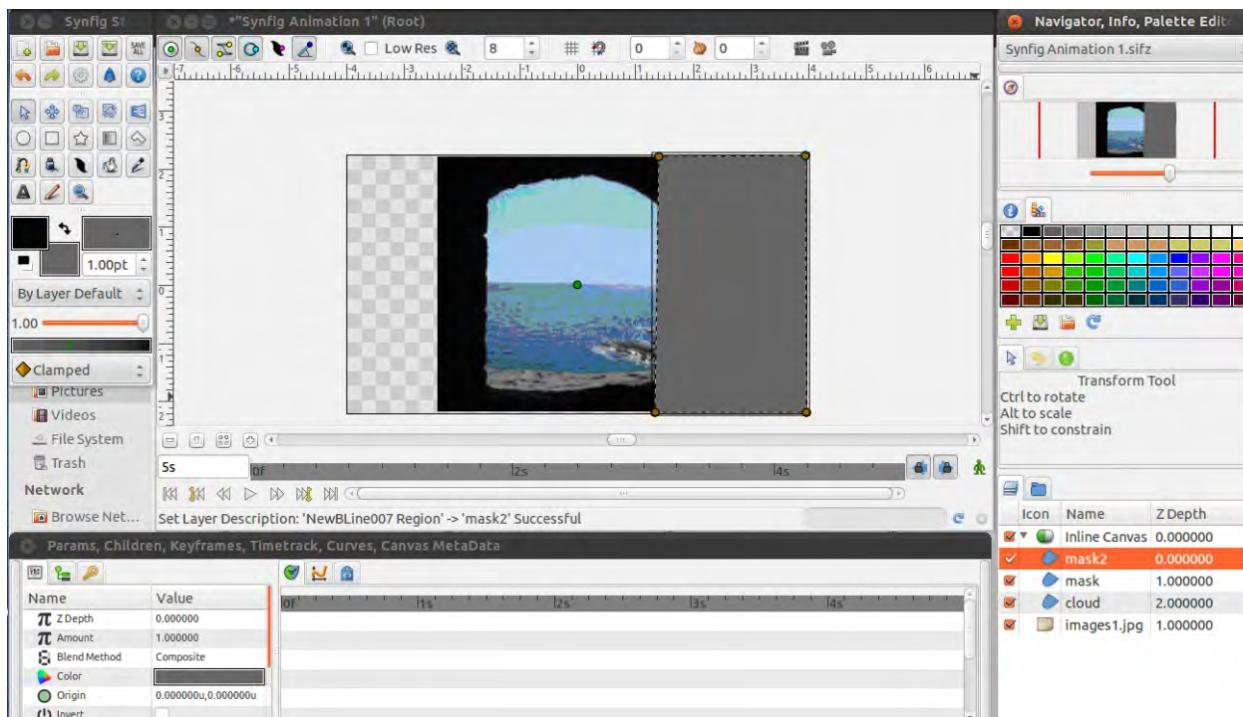


Figure 5.18 : Mask shape created on the right side of the canvas

- Rename the layer as mask2 and drag the mask2 shape layer into the encapsulated layer above the mask layer as shown in figure 5.19.

Icon	Name	Z Depth
✓	Inline Canvas	0.000000
✓	mask2	0.000000
✓	mask	1.000000
✓	cloud	2.000000
✓	images1.jpg	1.000000

Figure 5.19 : Repositioning mask2 layer

- Select the mask2 layer from the layers panel and in the parameters panel set its blend method to ‘Alpha over’.
- Play the animation and you can see that now the cloud moves through the window image.
- Save and render the file. Figure 5.20 shows the output on web browser.

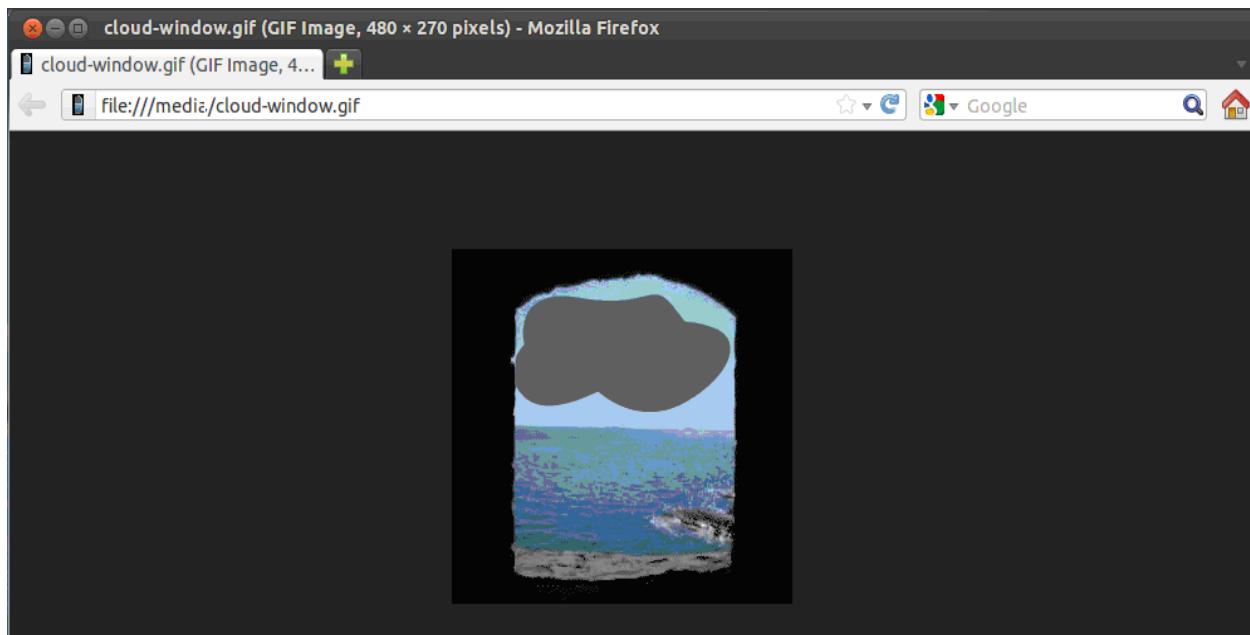


Figure 5.20 : Output displayed in web browser

Thus, using masking we can hide the objects. Masking can also be used to reveal the objects. Let us say, in the image we want to show the sun. Only some part of the sun is visible in the image. Here we use masking to reveal the objects.

- Create a new file.
- Import an image on the canvas using CTRL + i. Select the picture. Figure 5.21 shows the imported image and the image layer added in the layers panel.

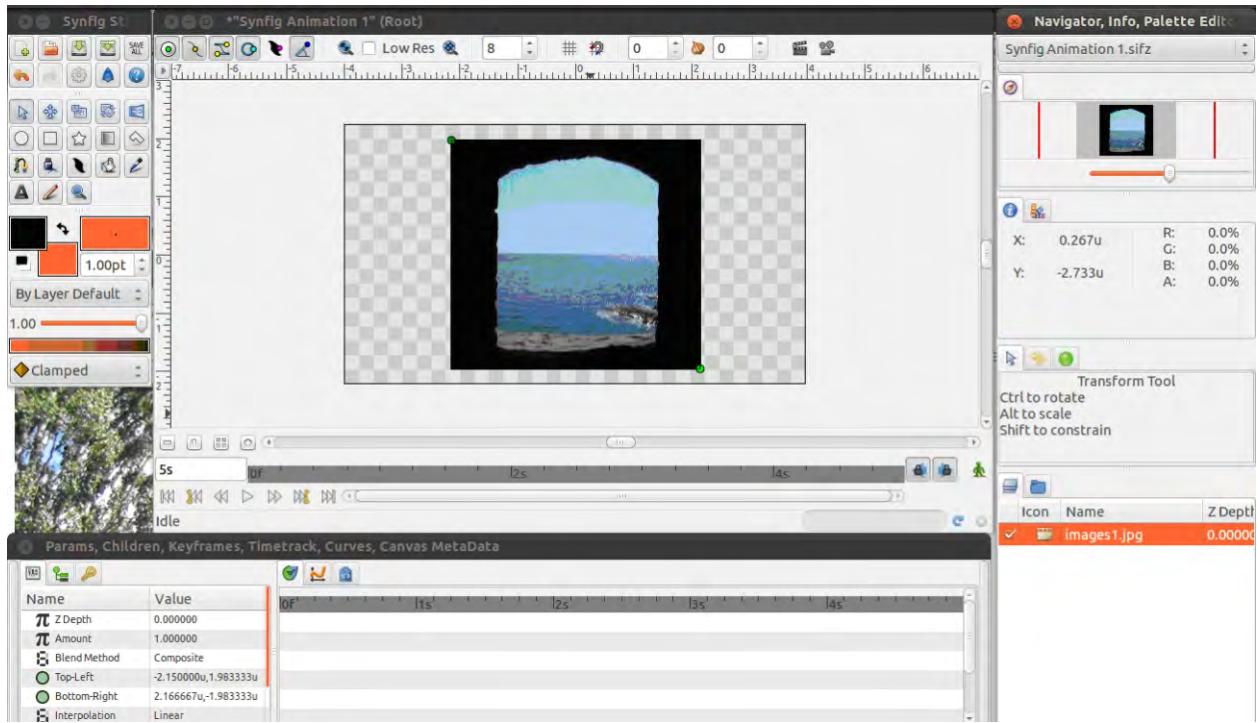


Figure 5.21 : Import the image

- Select circle tool and the color from the palette. Draw an orange coloured circle as shown in the figure 5.22. Rename the layer as ‘sun’.

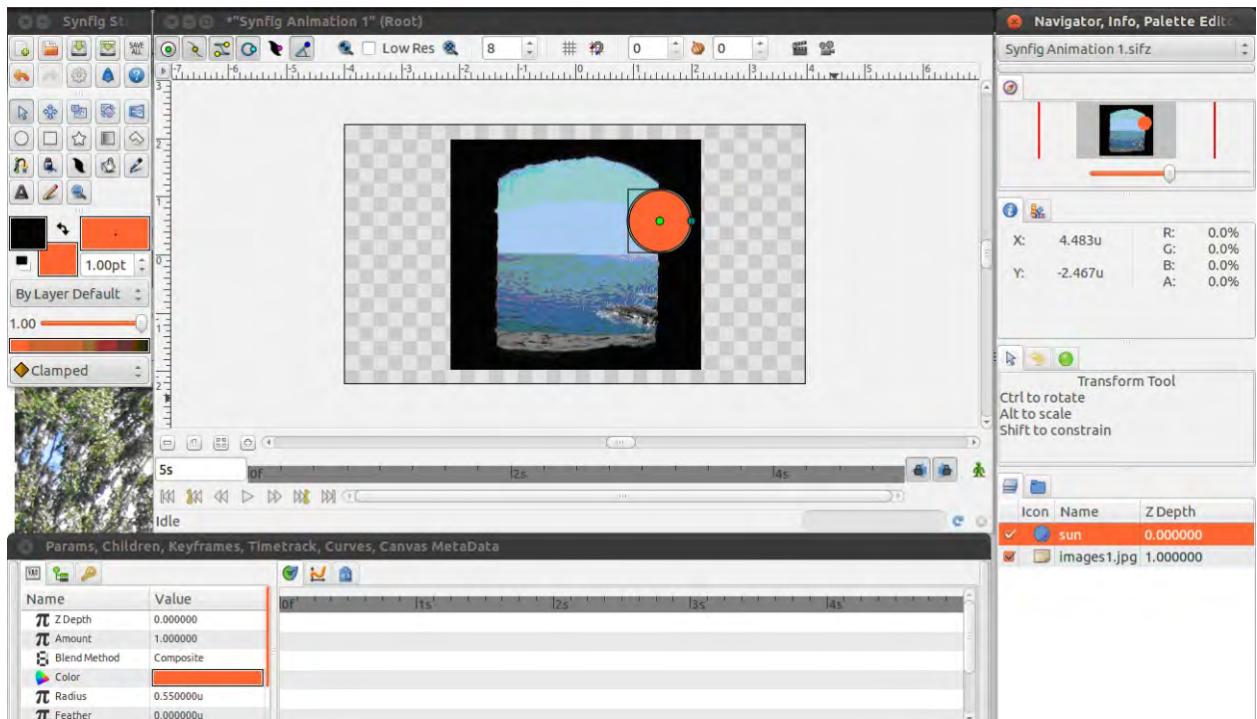


Figure 5.22 : Draw a circle

- Draw a mask shape using Bline tool. Here we need to draw the mask on the part of the sun that will be revealed. Figure 5.23 shows the mask. Rename the layer as ‘mask’ as shown in figure 5.24.

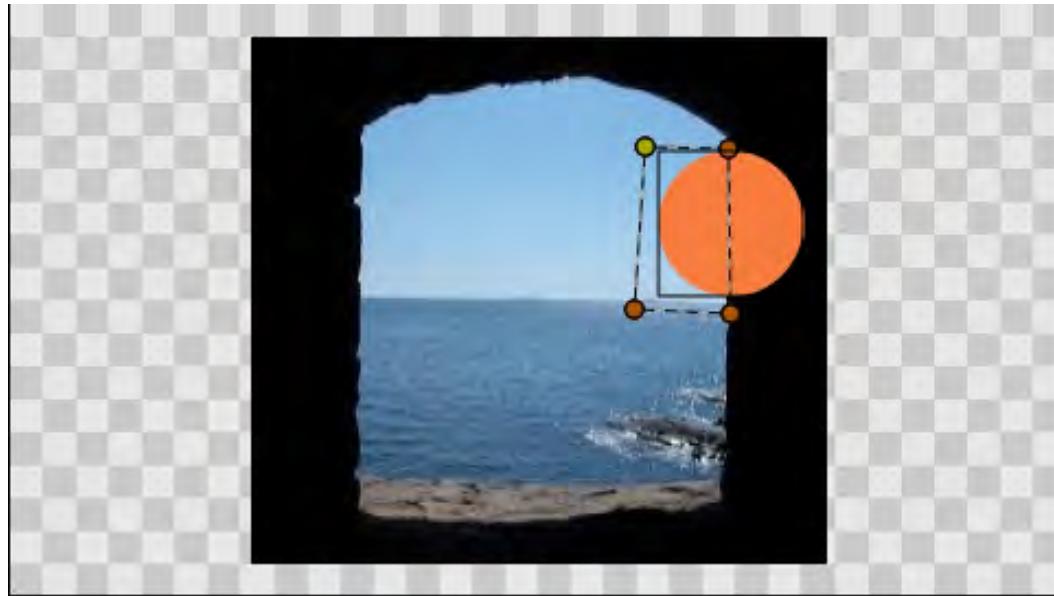


Figure 5.23 : Draw a mask shape

Icon	Name	Z Depth
mask	mask	0.000000
sun	sun	1.000000
image	images1.jpg	2.000000

Figure 5.24 : Renaming the new Bline layer

- Select the mask layer and in the parameters panel select the ‘invert’ option as shown in figure 5.25. Figure 5.26 shows the effect on the canvas after selecting invert option.

Params, Children, Keyframes, Timetrack, Curve		
BlendMethod	Composite	integer
Color	<input type="color" value="#FFFF00"/>	color
Origin	0.000000u,0.000000u	vector
Invert	<input checked="" type="checkbox"/>	bool
Antialiasing	<input checked="" type="checkbox"/>	bool
Feather	0.000000u	real

Figure 5.25 : Invert option selected in parameters panel

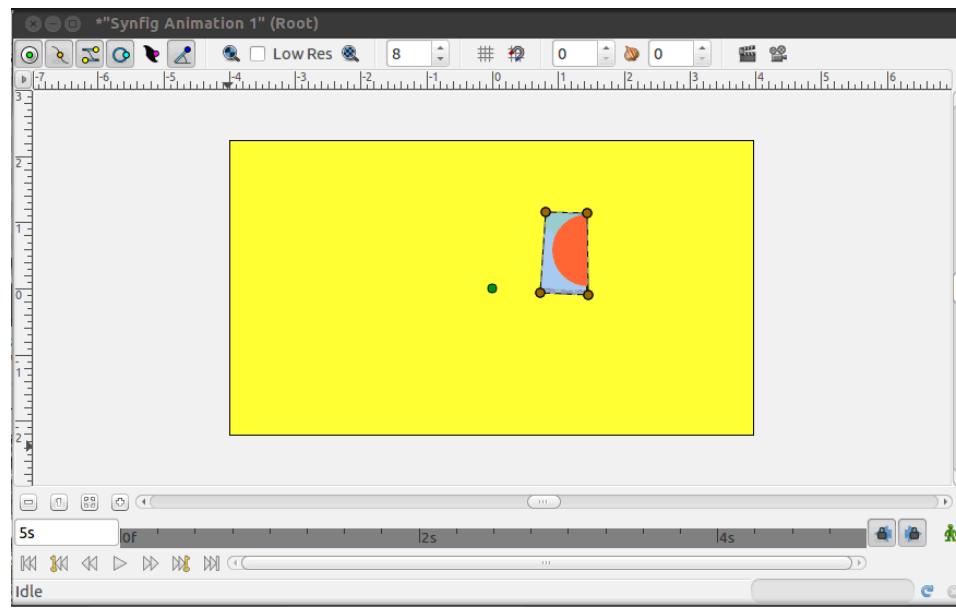


Figure 5.26 : Effect on the canvas after selecting invert option

- Change the blend method to ‘Alpha over’ as shown in the figure 5.27. The canvas will now appear as shown in figure 5.28.

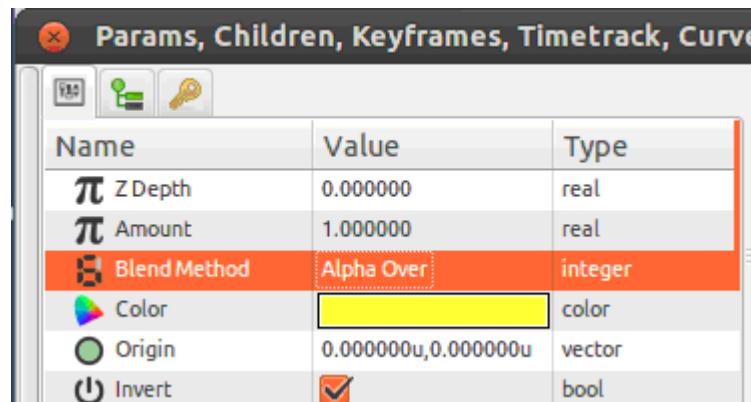


Figure 5.27 : Change the blend method to Alpha over

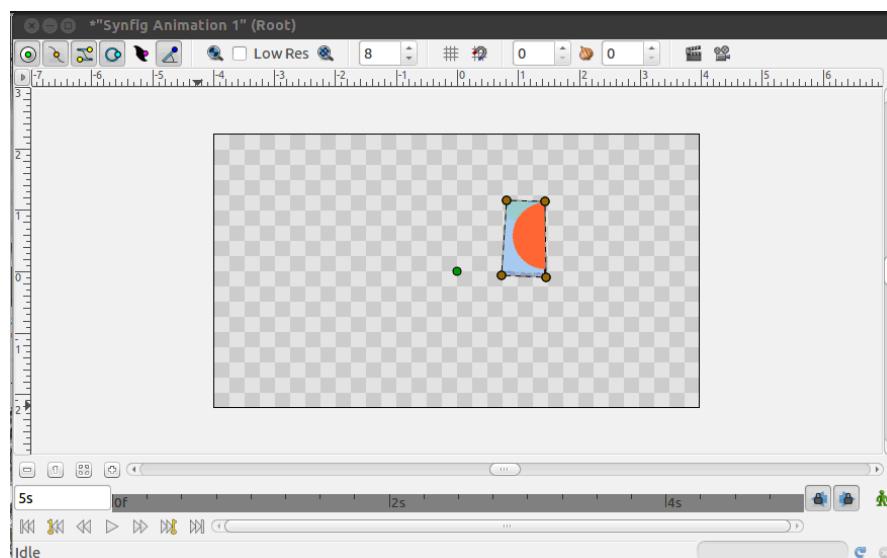


Figure 5.28 : Effect on the canvas after changing blend method

- As the image layer is placed below the mask layer, only the objects that are below the mask layer are visible. But we want the mask effect on only the sun layer and not the image. So we need to encapsulate both of them as shown in figure 5.29.

Icon	Name	Z Depth
✓ ▾	Inline Canvas	0.000000
✓	mask	0.000000
✓	sun	1.000000
✓	images1.jpg	1.000000

Figure 5.2 9: Encapsulate the mask and sun layers

- We can see the effect of masking in figure 5.30. We can put any number of objects below the mask layer which will be revealed. The objects on which we do not want to have masking effect should be placed out of the encapsulated layer.

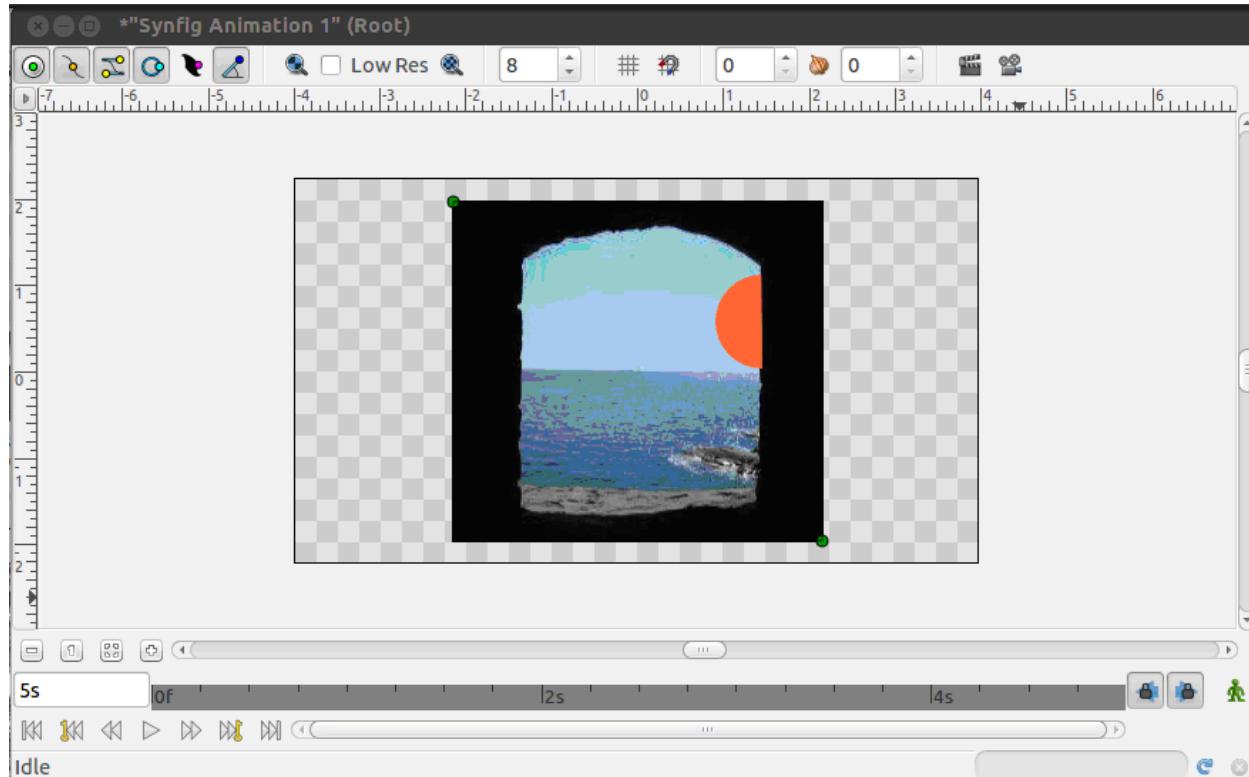


Figure 5.30 : Effect of masking to reveal an object

Create slideshow using Synfig

Assume that you want to create a slideshow on “Gujarat Tourism” wherein you want to show the viewer glimpse of Gujarat. To show the glimpse we will need to display images one after the other. Also at a time, only one image is visible. As we need to work with individual image they should be placed on different layers. To handle the visibility of the images we will use the parameters ‘blend method’ and ‘amount’ property of each layer. We have earlier used the blend method property which

defines the layering of the image on everything below it. The amount property is used for the visibility of the image just like the alpha value. Amount value of 1 signifies the layer is fully visible and 0 means the layer is fully transparent. Follow the steps given to create the slideshow:

- Create a new file.
- Press CTRL + i to import the image. If you do not have images then download some from the Internet. Import the images one by one on the canvas. Place each image in a different layer. Figure 5.31(a), (b), (c), (d), (e) and (f) shows the six imported images in the layers panel. Resize the images to the size of the canvas.

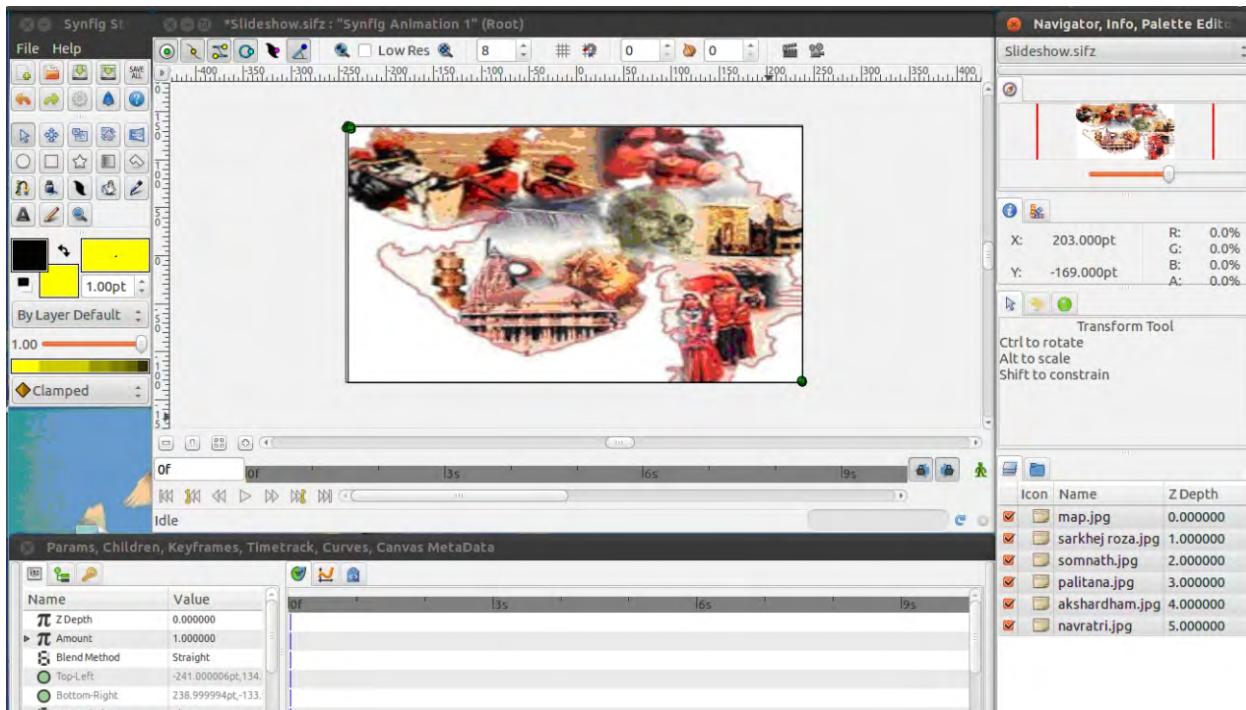


Figure 5.31(a) : Imported Gujarat Map image

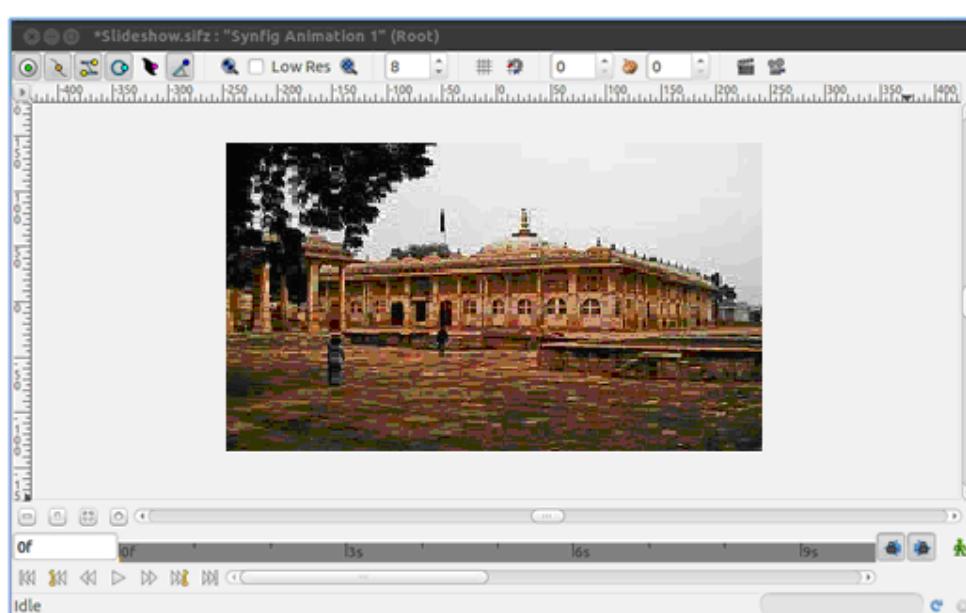


Figure 5.31(b) : Imported Sarkhej Roza image

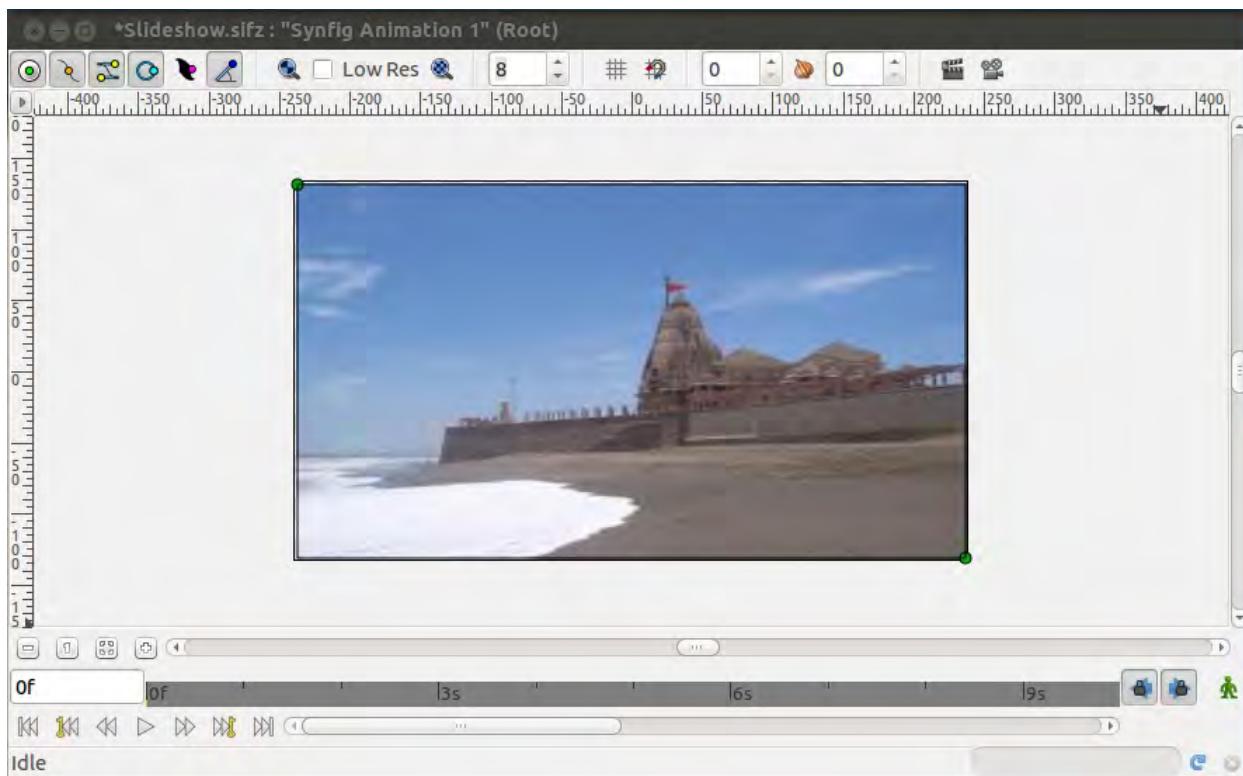


Figure 5.31(c) : Imported Somnath image

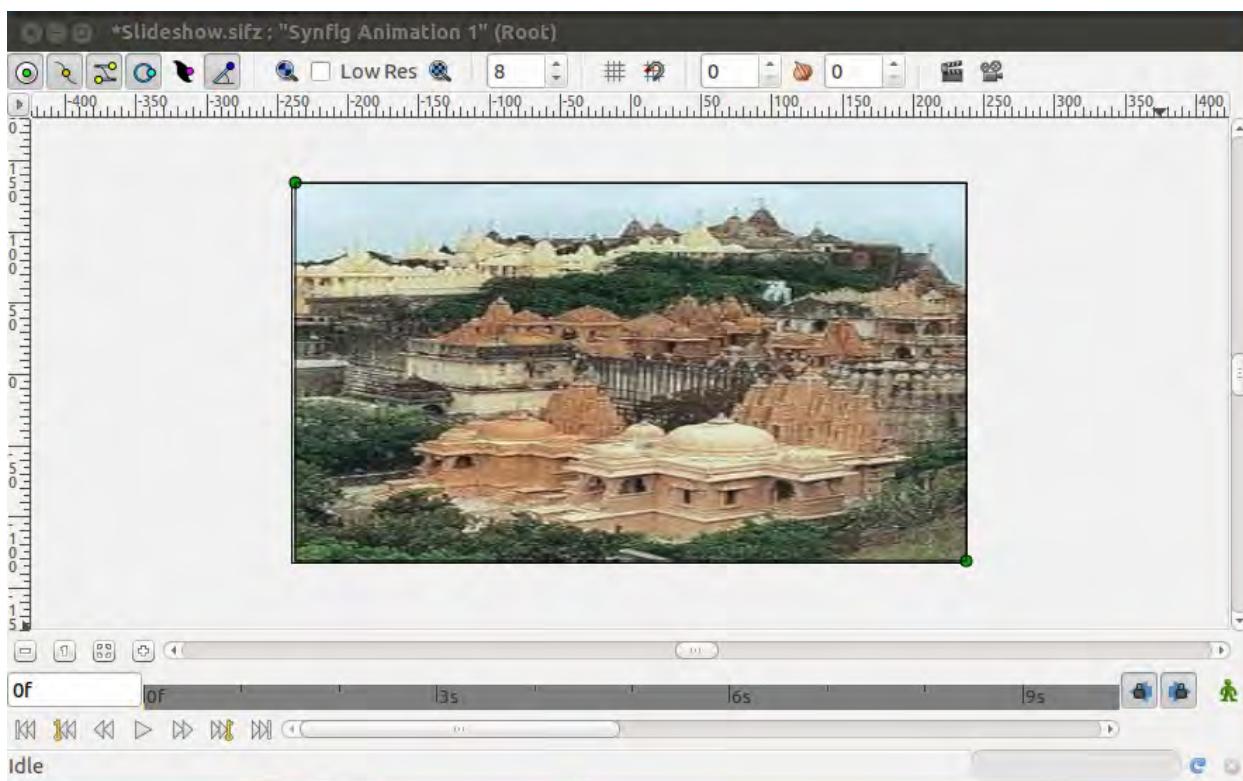


Figure 5.31(d) : Imported Palitana image

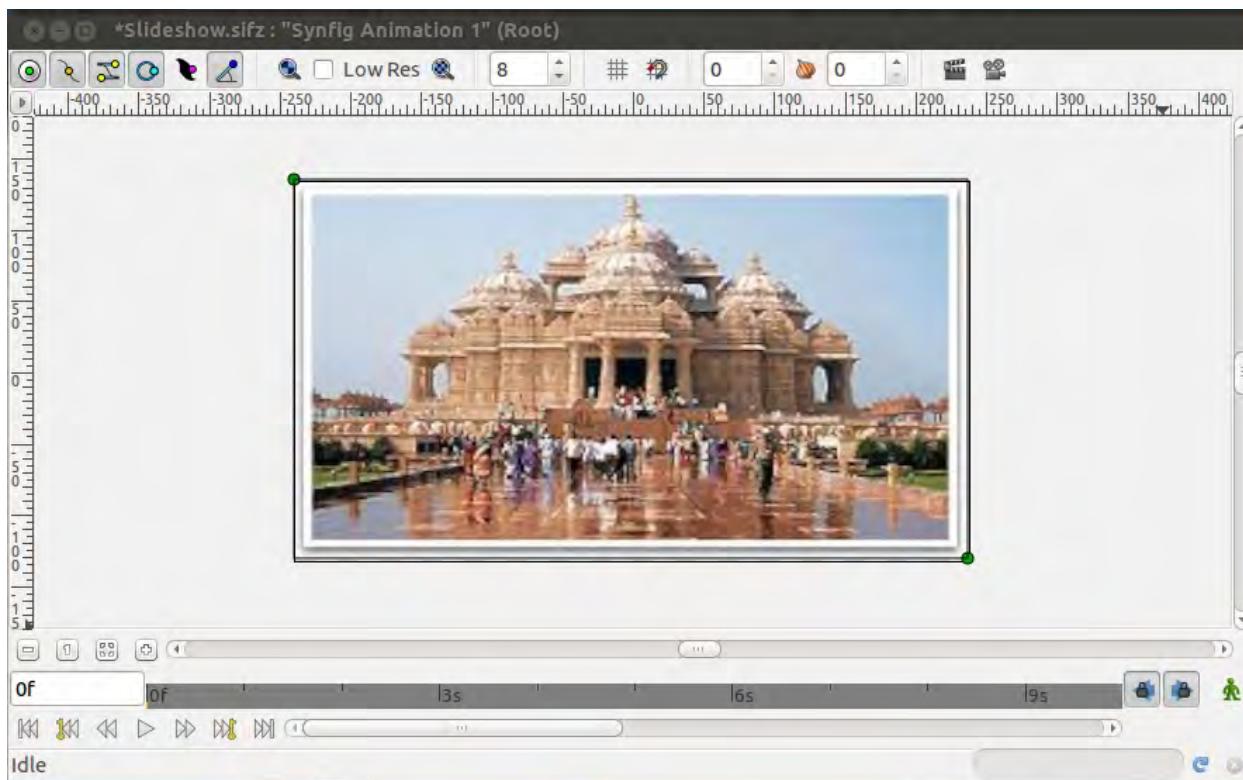


Figure 5.31(e) : Imported Akshardham image

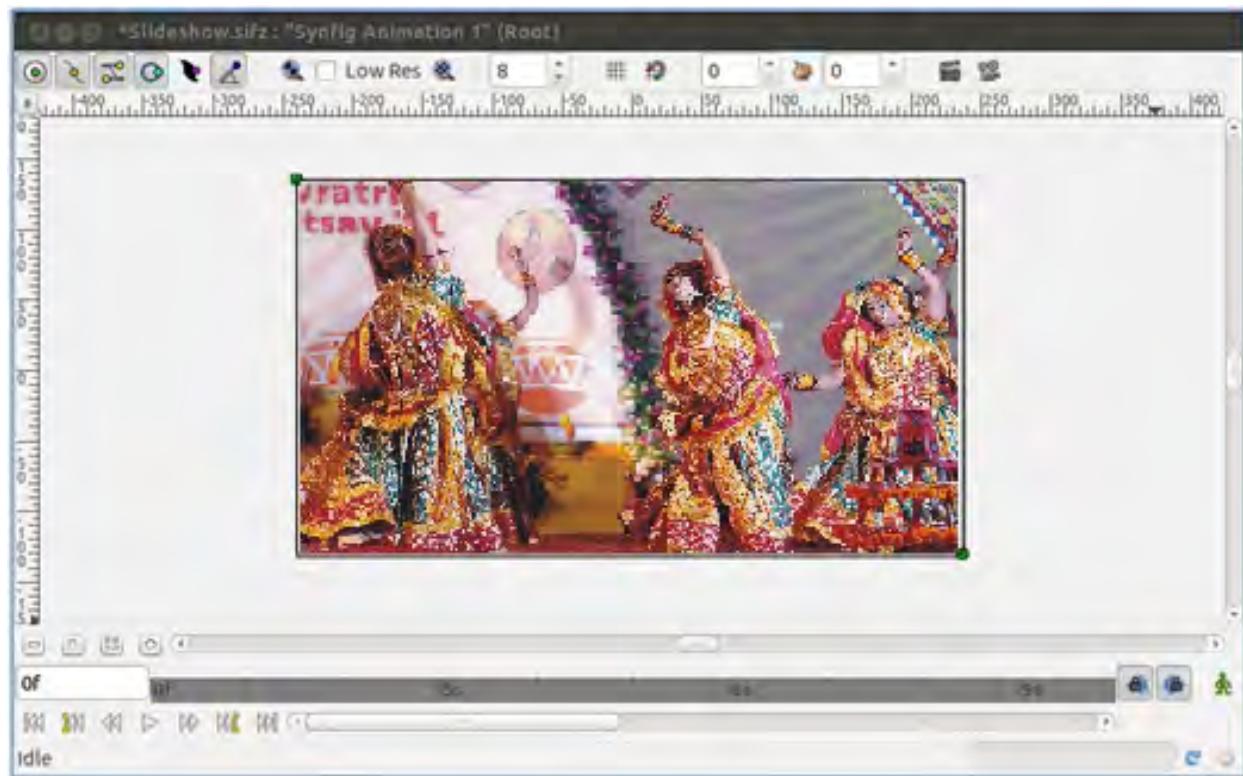


Figure 5.31(f): Imported Navratri image

- Press shift key and select all the images in the layers panel. Go to parameters panel and change the blend method to ‘straight’ for all the images as shown in figure 5.32.

Icon	Name	Z Depth
✓	map.jpg	0.000000
✓	sarkhej roza.jpg	1.000000
✓	somnath.jpg	2.000000
✓	palitana.jpg	3.000000
✓	akshardham.jpg	4.000000
✓	navratri.jpg	5.000000

Name	Value	Type
π Z Depth	0.000000	real
π Amount	1.000000	real
Blend Method	Straight	integer
Top-Left	-3.966667u,2.216667u	vector
Bottom-Right	4.000000u,-2.216667u	vector
Interpolation	Linear	integer (Static)

Figure 5.32 : Select the images and change blend method to straight

- After setting the blend method property we now need to work with the amount parameter. But like blend method, the amount parameter cannot be set collectively for all the layers. We need to work with amount property for each layer individually. Here we need to convert the amount property to ‘Timed swap’.
- Select the topmost map layer as shown in figure 5.33 and in the parameters panel select the amount parameter and right click. Right clicking will open a context menu. Select Convert → Timed swap. Convert specifies that the parameter is controlled automatically in different ways. Each parameter can be converted and contains different options. Figure 5.34 shows the amount property Timed swap. This will add new sub parameters: Before, After, Time and Length to the amount property as shown in figure 5.35. These options specify the visibility of the image, swapping it from ‘before’ to ‘after’ parameter using ‘length’ and finishing the swap at the specified ‘time’.

Icon	Name	Z Depth
✓	map.jpg	0.000000
✓	sarkhej roza.jpg	1.000000
✓	somnath.jpg	2.000000
✓	palitana.jpg	3.000000
✓	akshardham.jpg	4.000000
✓	navratri.jpg	5.000000

Figure 5.33 : Select the topmost image layer

Name	Value	Type
π Z Depth	0.000000	real
π Amount	1.000000	real (Timed Swap)
Blend Method	Straight	integer
Top-Left	-3.966667u,2.216667u	vector
Bottom-Right	4.000000u,-2.216667u	vector
Interpolation	Linear	integer (Static)

Figure 5.34 : Change the amount parameter to Timed swap

Params, Children, Keyframes, Timetrack, Curve		
Name	Value	Type
π ZDepth	0.000000	real
π Amount	1.000000	real (Timed Swap)
π Before	1.000000	real
π After	1.000000	real
Time	2s	time
Length	1s	time

Figure 5.35 : Newly added parameters to the amount parameter

- Set the parameter ‘before’ to 1.0 and ‘after’ to 0.0. The parameter before set to 1.0 indicates the layer will be visible before the ‘time’ specified and after set to 0.0 indicates that the layer will be invisible after the specified ‘time’. If you want the images to display for 5 seconds using 1 second transition between each image then change the length to ‘1s’ and time to ‘5s’. After 5 seconds the image will be invisible. Figure 5.36 shows the settings applied to amount parameter.

Params, Children, Keyframes, Timetrack, Curve		
Name	Value	Type
π ZDepth	0.000000	real
π Amount	1.000000	real (Timed Swap)
π Before	1.000000	real
π After	-0.000000	real
Time	5s	time
Length	1s	time

Figure 5.36 : Map layer Sub parameters settings

- Select the next image layer. Make it timed swap just like we did in earlier layer. Set the parameter ‘before’ to 1.0, ‘after’ to 0.0 and length to 1s. Set time to 10s. The swapping will be done at 10s. And image will be invisible after 10s. Likewise for the next image

Params, Children, Keyframes, Timetrack, Curve		
Name	Value	Type
π ZDepth	0.000000	real
π Amount	1.000000	real (Timed Swap)
π Before	1.000000	real
π After	-0.000000	real
Time	10s	time
Length	1s	time

Figure 5.37(a) :
Time setting for Sarkhej Roza layer

Params, Children, Keyframes, Timetrack, Curve		
Name	Value	Type
π ZDepth	0.000000	real
π Amount	1.000000	real (Timed Swap)
π Before	1.000000	real
π After	-0.000000	real
Time	15s	time
Length	1s	time

Figure 5.37(b) :
Time setting for Somnath layer

Name	Value	Type
π Z Depth	0.000000	real
π Amount	1.000000	real (TimedSw)
π Before	1.000000	real
π After	-0.000000	real
<input checked="" type="radio"/> Time	20s	time
<input checked="" type="radio"/> Length	1s	time

Figure 5.37(c) :
Time setting for Palitana layer

Name	Value	Type
π Z Depth	0.000000	real
π Amount	1.000000	real (TimedSw)
π Before	1.000000	real
π After	-0.000000	real
<input checked="" type="radio"/> Time	25s	time
<input checked="" type="radio"/> Length	1s	time

Figure 5.37(d) :
Time setting for Akshardham layer

layers change the time to 15s, 20s, 25s and so on. Figure 5.37 (a), (b), (c), (d) and (e) shows the settings for all the image layers.

Name	Value	Type
π Z Depth	0.000000	real
π Amount	1.000000	real (TimedSw)
π Before	1.000000	real
π After	-0.000000	real
<input checked="" type="radio"/> Time	30s	time
<input checked="" type="radio"/> Length	1s	time

Figure 5.37(e) : Time setting for Navratri layer

- Click on Caret → Edit → Properties. Select the time tab and set the end time equal to the last image layers time value. In our case the last image layer time is 30s. You can set the end time a little more than 30s if you want a blank screen in the end. Figure 5.38 shows the end time setting.

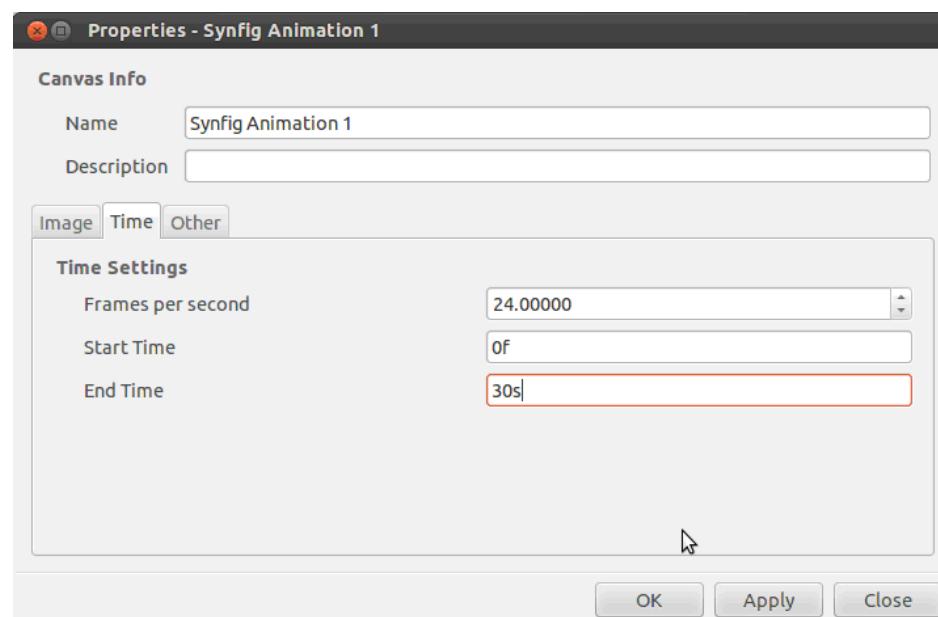


Figure 5.38 : End time set to 30s

- Click the play button to preview the presentation.
- Save and render the file. Figure 5.39 shows the output in web browser.

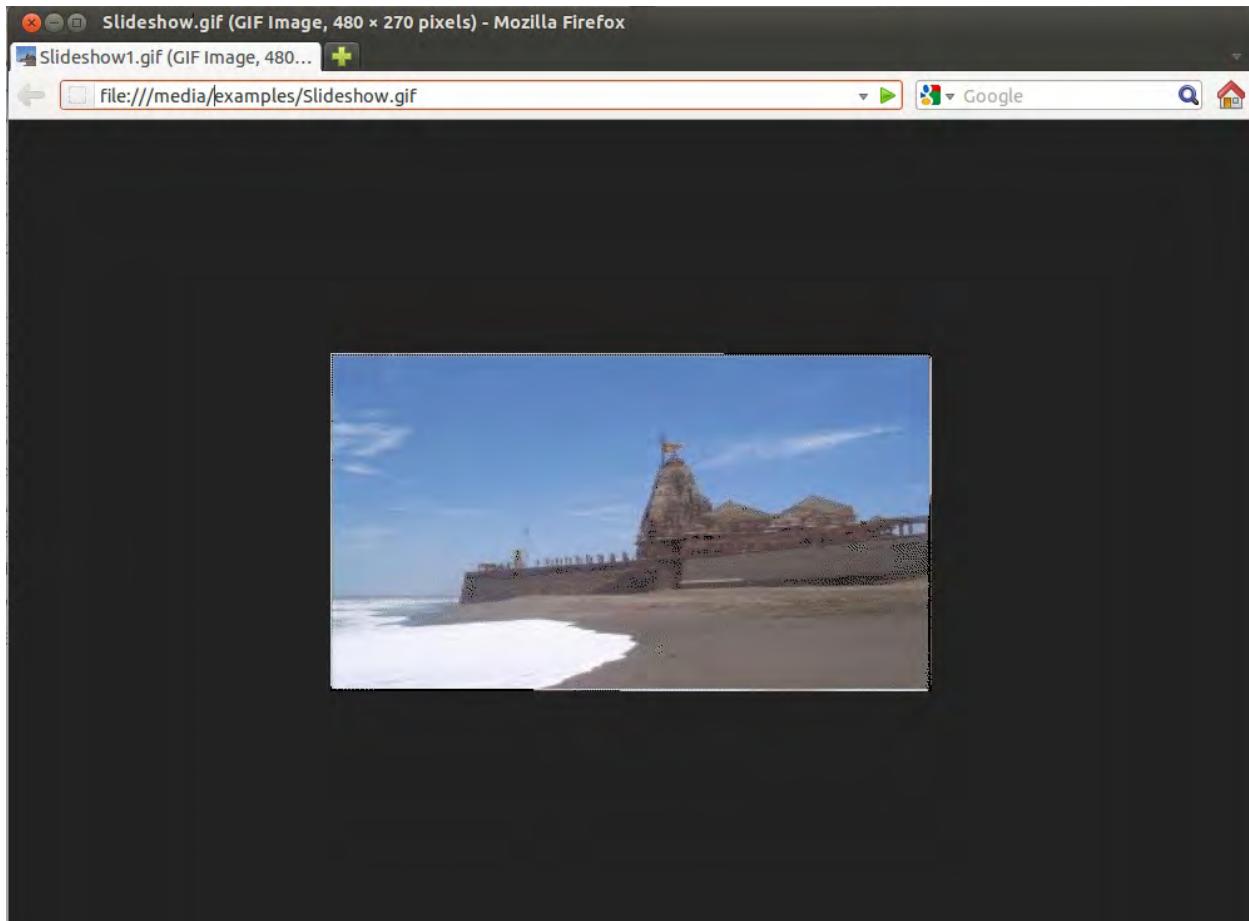


Figure 5.39 : Output displayed in web browser

If you want to place a background in the slideshow then encapsulate all the image layers. Place the background as the bottommost layer. The blend and amount parameters will be applied only in the encapsulated layer. We can also add text to the images and display the text during the slideshow.

Interactivity

Sometimes in animation, you need to move according to the choice of the user. Also the user might want to skip a particular portion of the animation and move further. These actions used for interactivity in animation software are known as action scripts. Action scripts are used to add flexibility to the animation programs. The term action script signifies the use of scripting language to add flexibility to the actions performed during the animation. It helps the user to either play the animation on a particular action or move as per his choice during the animation.

As of now, Synfig studio does not support action scripting directly in the animation software. But we can use javascript or java to add interactivity to our animation. It may not be possible here to discuss in detail the use of javascript or java to our animation. But let us discuss an example wherein we have used javascript to add interactivity to our animation.

Assume that you want to play the animation only when the user takes the mouse over the image. To achieve this, first we take two images. One of the images is the static image. In our example the static image is static-ball.gif. The second image is the animated image. In our example the animated image is animated-ball.gif. The animated image is the rendered file from Synfig in a format like gif, jpeg or bmp. The listed options are available when we render the files in Synfig as discussed in the earlier chapters. When the user opens the web browser, static image will be displayed. After he/she takes the mouse over the image, the animated image will be displayed using the javascript code. When the user moves the mouse away from the image the animation will stop. Follow the steps as mentioned:

- Create a new file in Gedit and write the code given:

```
<html>
<head>
<title>Rollover with a Mouse Events</title>
<script type="text/javascript">
if( document.images ){
    var image1 = new Image(); // Preload an image
    image1.src = "static-ball.gif";
    var image2 = new Image(); // Preload second image
    image2.src = "animated-ball.gif";
}
</script>
</head>
<body>
<a href="#" onMouseover="document.myImage.src=image2.src" onMouseout="document.myImage.src=image1.src">

</a>
</body>
</html>
```

- Save the file with extension .html.
- Open the file in a web browser to see the output. Figure 5.40 shows the output when the page loads. Taking the mouse over the image will animate it. Figure 5.41 shows the animated image.

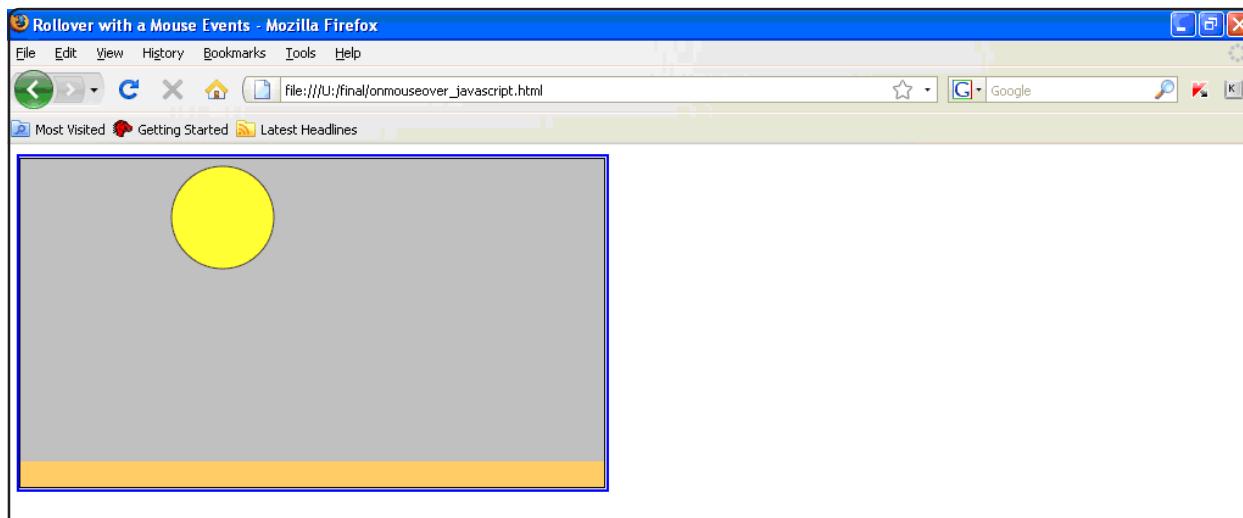


Figure 5.40 : Image displayed when the page loads.

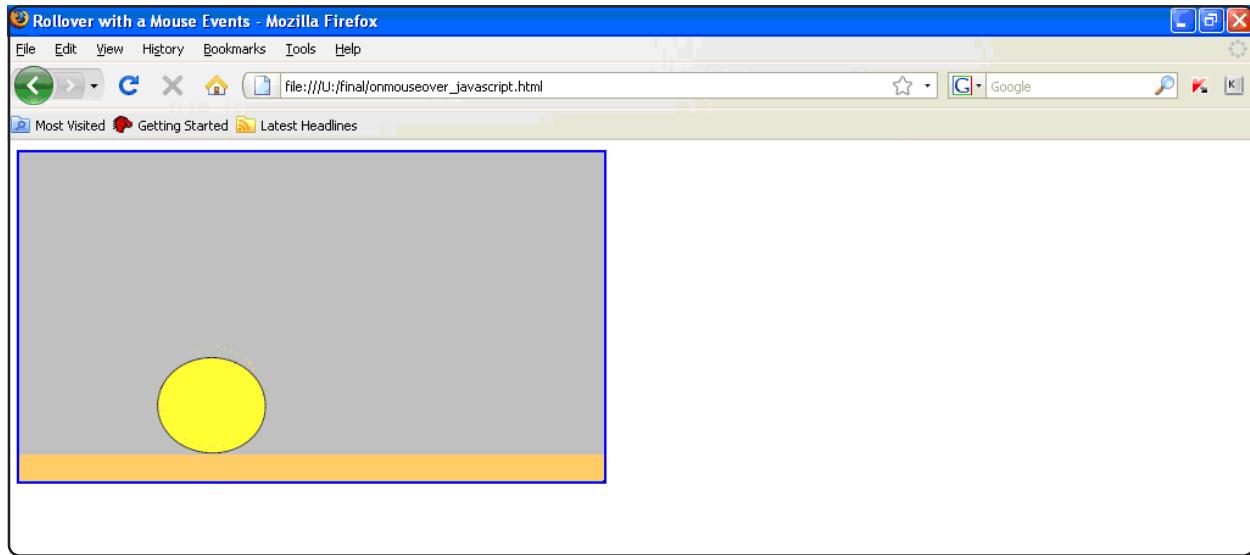


Figure 5.41 : Animated image displayed when the mouse is over the image.

Thus, using javascript we added interactivity to the animation.

Portability of files

The animation files created using particular animation software can be converted into Adobe Flash files (SWF files) for further manipulation. SWFTOOLS is an open source software tool suite which provides the conversion of various file formats into swf files. SWF is an Adobe flash file format used for vector graphics animation. After converting the files into swf format they can be further manipulated or created using Adobe Flash animation software. The tool includes programs for reading SWF files, combining them, and creating them from other content like images, sound or video file. Table 5.1 shows some of the programs used for conversion of files into swf format.

Program	Description
GIF2SWF	Converts gif files into swf files
PNG2SWF	Converts png files into swf files
JPEG2SWF	Converts jpeg files into swf files
PDF2SWF	Converts pdf files into swf files
WAV2SWF	Converts wav audio files into swf files
AVI2SWF	Converts avi animation files into swf files
FONT2SWF	Converts font files to swf files
SWFExtract	Extracts images, sounds and movie clips from swf files

Table 5.1 : List of programs supported in SWFTOOLS

Using the programs listed in the table 5.1, we can also convert the gif or jpeg files created using Synfig to Adobe Flash files (swf) for further manipulation.

Summary

In this chapter we learned how to insert pictures and work with them. Pictures like any other object created in Synfig can be rotated, transformed or scaled. We also discussed to hide or reveal an object using the concept of masking. We can also create a slideshow of images by working with the blend method and timed swap parameter. Adding interactivity to the animation can be provided by using javascript or java.

EXERCISE

LABORATORIES EXERCISES

1. Create a slideshow on the theme of “Khushboo Gujarat ki”.
 2. Create a slideshow on the theme of “Festivals of India”. Use text to display name of each festival.
 3. Insert an image showing scenery with sky and mountains. Using masking show the sunrise in the image.



6

Basic Ubuntu Linux Commands



In standard 9 we have studied in depth the working of an operating system called Ubuntu. During the study we learned that once a user logs into the computer system having Ubuntu Linux, he/she can interact with the computer using command line interface or graphical interface. Both these interfaces are important and have their own uses. In this chapter we will learn how to use the command line interface in detail. The command line interface allows us to access the real power of Linux with greater efficiency. It is the most influential parts of any Ubuntu system.

Starting Up the Terminal

To open a command-line console in a graphical interface, a window named terminal window is provided in Linux. To open a terminal window, click on Applications à Accessories à Terminal alternatively you can use CTRL+ ALT + t keys together. A terminal window similar to the one shown in figure 6.1 will be seen on the screen. Note that the look of your actual window may be different from the one shown in figure 6.1 as the user might be different.



Figure 6.1 : Terminal Window

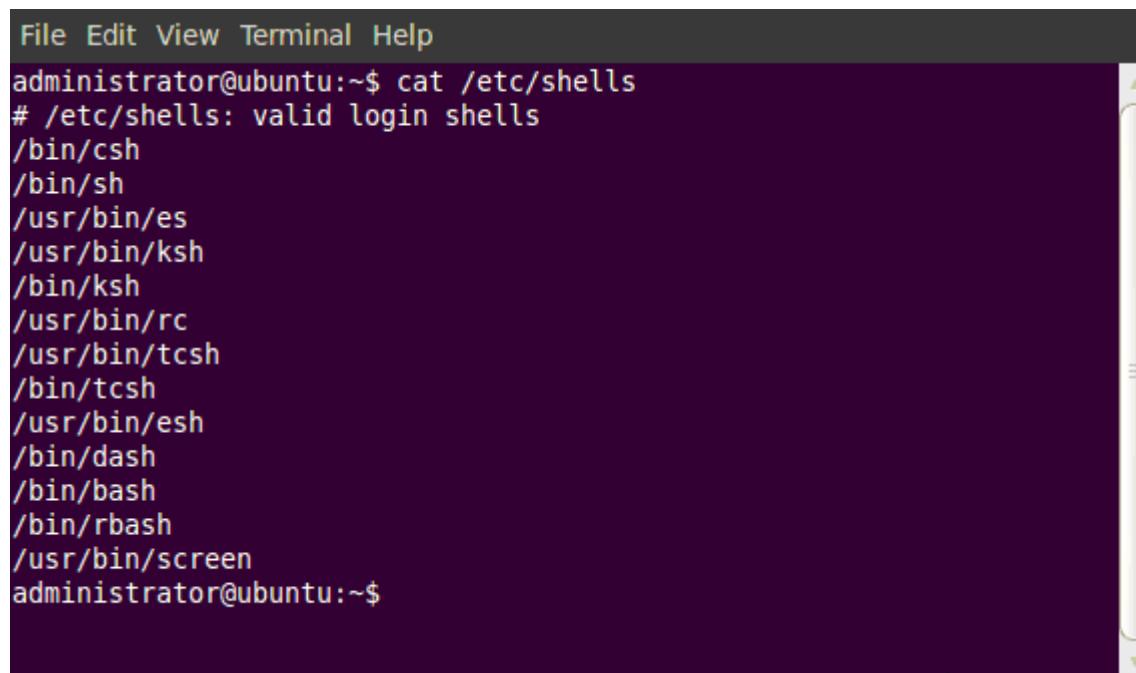
Once the window is clearly visible, you will see a blinking cursor preceded by some letters, and perhaps numbers and symbols, ending with a \$. The first word in that string of characters is username, followed by the symbol @. The symbol @ is followed by the name of the computer that is being used. Finally you will find a colon and the name of the directory you are working in (Generally you start working in your home directory, which is represented by a ~ symbol).

The command prompt indicates that the interface is ready to interact with the user in the form of commands. A command is basically a program that accomplishes certain task. Once the prompt is displayed we can issue commands as described in this chapter. Before discussing various commands, let us revise our knowledge about the term shell. A shell is the command-line interface. Shell is a user program or an environment provided for user interaction. It is a command language interpreter that accepts or issues commands, understands it, interacts with the kernel to execute it and displays results as per given instruction. Numerous shells are available to work on the Ubuntu Linux systems, but the shells available on a particular system may vary.

Some popular shells provided with Linux are Bourne shell (sh), C shell (csh and tcsh), Korn shell (ksh) and bash (sh) shell. Bourne shell with sh as its acronym is the earliest Unix shell used as command line interface. Bourne shell provides basic mechanisms for shell script programming, which allows us to write a program based solely on commands. C shell identified as csh is another shell commonly found on Linux systems. Shell programming can be done using C programming syntax in this shell. The newer version of csh is tcsh. It provides additional shell script programming features to address the limitations of csh. The Korn shell or ksh was developed to combine the features of both sh and csh. Bash shell is a newer version of Bourne shell. Thus it contains same syntax and functions as sh. Nowadays bash is considered standard shell for Linux systems and is thus commonly used and available on all Linux operating systems.

Listing the shells available on the system

To find all available shells in your system you can use the cat command. Type the command as shown in figure 6.2 on the command prompt. You will get list of all the available shells in your computer system. The list of shell that you get as an output may be different from the one shown in figure 6.2 depending on your system configurations. The cat command is discussed later in this chapter.



```
File Edit View Terminal Help
administrator@ubuntu:~$ cat /etc/shells
# /etc/shells: valid login shells
/bin/csh
/bin/sh
/usr/bin/es
/usr/bin/ksh
/bin/ksh
/usr/bin/rc
/usr/bin/tcsh
/bin/tcsh
/usr/bin/esh
/bin/dash
/bin/bash
/bin/rbash
/usr/bin/screen
administrator@ubuntu:~$
```

Figure 6.2 : Different available shells

Determining the default shell

Each Ubuntu Linux account is configured with a certain shell as its default command line interface. Each time you log on, this default shell is utilized for working within the system. Linux operating system comes with certain variables that contain current environment settings as its values and thus are known as environment variables.

The value of default shell is stored in one such environment variable named SHELL. Thus by displaying the value of the variable SHELL, we can come to know which our default shell is. To display the value of any variable, *echo* command can be used. Type echo \$SHELL on the command prompt and press Enter key. Default shell will be displayed on the screen as shown in figure 6.3. Note that Linux commands are case sensitive hence SHELL, Shell and shell are not same.

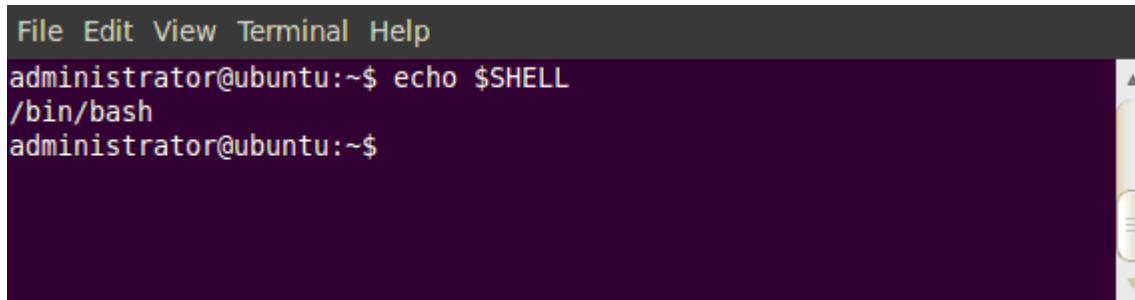
A screenshot of a terminal window titled "Terminal". The window has a dark background and a light-colored text area. At the top, there is a menu bar with "File", "Edit", "View", "Terminal", and "Help". Below the menu bar, the terminal prompt shows "administrator@ubuntu:~\$". The user then types "echo \$SHELL" and presses Enter. The output " /bin/bash" is displayed in the terminal. The terminal window has scroll bars on the right side.

Figure 6.3 : Default Shell

Changing the current shell

As discussed earlier we have different shells available with Linux. To change your default shell, type the name of the shell you want to use on the command line. For example, if you want to use the C shell (provided it is available on the system), type csh at a command prompt. Then the command prompt will provide a csh interface.

Note :

A shell change is temporary and will last only as long as you are logged on that command line.

To return to default shell, type exit or press CTRL + d at the command prompt of the new shell.

Command Syntax

The syntax of Linux commands is uniform. It consists of three parts, in the order specified below :

- **Name :** It is the name of the command, for example ls, echo etc.
- **Options :** It is possible to alter the behavior of the commands by specifying additional options. A command may have zero or more options. Options when present starts with a hyphen symbol (-) and are usually a single letter or a digit. Some commands may have options with double hyphen and/or sequence of letters or digits. Depending on the command, the number and meaning of the options will vary.
- **Arguments :** Along with options user can also provide arguments. A command may take zero or more arguments to do its work. The number and expected meaning of the arguments vary from command to command. Some commands may take no arguments; others may take an exact number, while other commands may take any number of arguments.

Linux commands can be classified as internal or external based on whether its binary file exists or not. The commands that have a binary file explicitly stored in either /sbin, /usr/sbin, /usr/bin, /bin, or /usr/local/bin directories are called external commands. They are generally executed by the kernel and will generate a process id at the time of execution. Most of the commands that we use in

Linux are external commands. On the other hand the commands directly executed by the shell are called internal commands. Internal commands do not generate a new process.

To know whether a command is internal or external we can use the *type* command. The syntax of type command is shown below :

\$type command

For example if we execute a command

\$type info

we will get the output as shown below :

info is /usr/bin/info

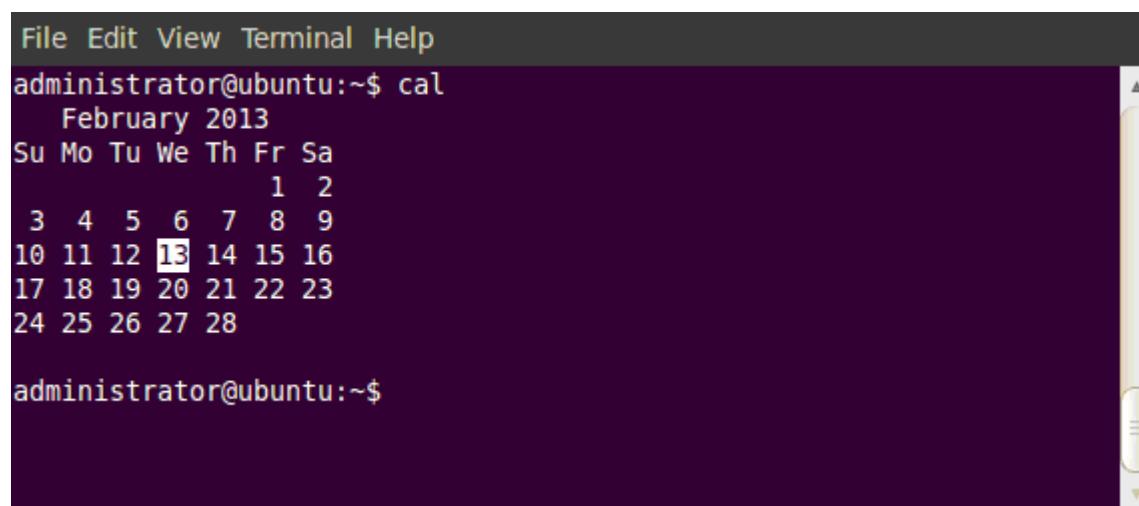
As can be observed this refers to a binary file *info* stored in /usr/bin/, this indicates that info is an external command.

Issuing General Purpose Commands

As you are now familiar with the syntax of commands, let us see how to issue commands through command line interface. The best way to start with learning Linux commands is to try working with some general purpose Linux commands. You can issue a command by typing a command name followed by necessary options and arguments. Other way is you can type in first few letters of a command, press the tab key and the shell automatically provides the remaining information. For example, to display a calendar, type ca on the prompt and press tab key. Linux shell will automatically display list of all commands starting with alphabets ca including the calendar command *cal* on the screen. If you get more than one command in the list, then type the desired command on the prompt and press Enter key to execute the command.

Calendar (*cal*)

The *cal* command is used to display a calendar of any specific month or entire year. The default output of *cal* command is calendar of the current month. See figure 6.4.



The screenshot shows a terminal window with a dark background. At the top, there is a menu bar with 'File', 'Edit', 'View', 'Terminal', and 'Help'. Below the menu, the terminal prompt 'administrator@ubuntu:~\$' is visible. The main area of the terminal displays the output of the 'cal' command for February 2013. The output shows the days of the week (Su Mo Tu We Th Fr Sa) and the dates of the month. The date '13' is highlighted with a red box. The terminal window has scroll bars on the right side.

```
File Edit View Terminal Help
administrator@ubuntu:~$ cal
      February 2013
Su Mo Tu We Th Fr Sa
      1  2
  3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28

administrator@ubuntu:~$
```

Figure 6.4 : Output of cal command

We can change the calendar as per our requirement, for example to display the calendar of January, 2013 type the following command on the prompt and press Enter key.

\$cal 01 2013

The output of this command will be similar to the one shown in figure 6.4 except that the month would be January 2013. Similarly if we want to display the calendar of the entire year 2013, we simply have to type cal followed by the year as shown below :

\$cal 2013

Note that the calendar of entire year may not be displayed on the entire monitor screen; hence we will have to use a pipe operator as shown below :

\$cal 2013 | more

In the above command we have concatenated two commands, here *more* is also a command which takes input from the *cal* command. The pipe (|) symbol used in between the two commands is discussed in detail later in this chapter.

Date (*date*)

Another utility command is *date*; it is used to display the system date.

\$date

The output of the command is shown in figure 6.5.

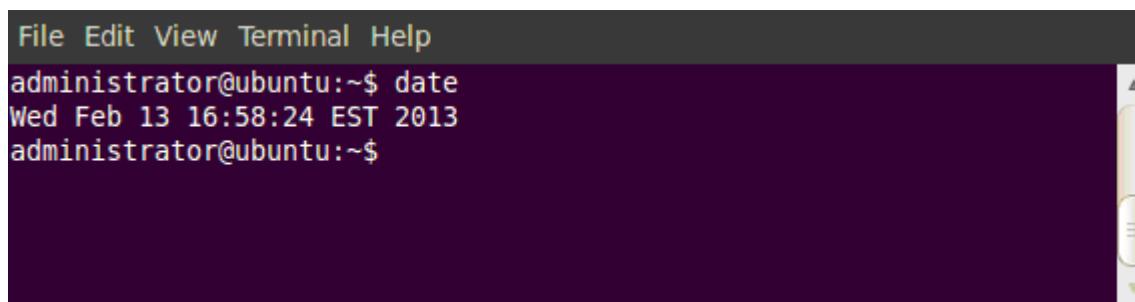
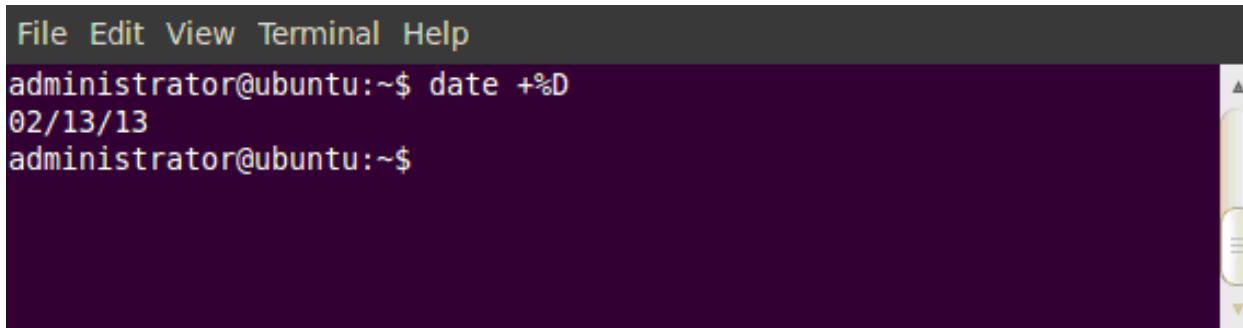
A screenshot of a terminal window titled 'Terminal'. The window has a dark background and a light-colored title bar. The title bar contains the word 'Terminal' and the path 'administrator@ubuntu:~\$'. The main area of the terminal shows the command 'date' being run and its output: 'Wed Feb 13 16:58:24 EST 2013'. The terminal window has scroll bars on the right side.

Figure 6.5 : Output of date command

Observe that the output displays both date as well as time. The *date* command can also be used with suitable format specification as arguments. Each format is preceded by + symbol, followed by % operator and a single character describing the format. For example, to display only the current date in mm/dd/yy format use the command shown below :

\$date +%D

Figure 6.6 shows the output of the command.



```
File Edit View Terminal Help
administrator@ubuntu:~$ date +%D
02/13/13
administrator@ubuntu:~$
```

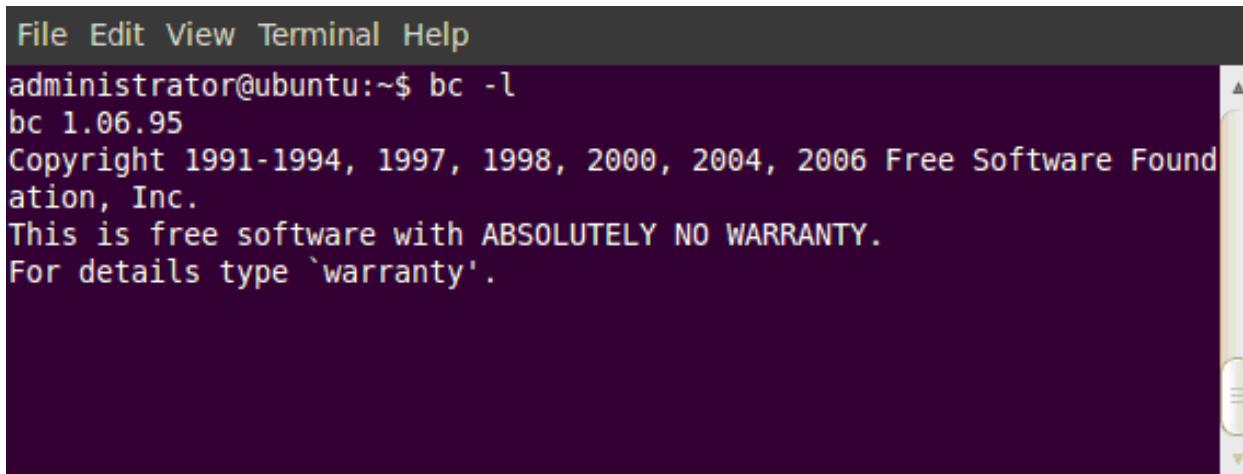
Figure 6.6 : Output of formatted date command

The command line calculator (*bc*)

The *bc* command in Linux is a command line calculator. In addition to performing simple mathematical functions, it can also perform conversions between different number systems, as well as allows us to use some scientific functions. To work with this command use the syntax given below :

```
$bc -l
```

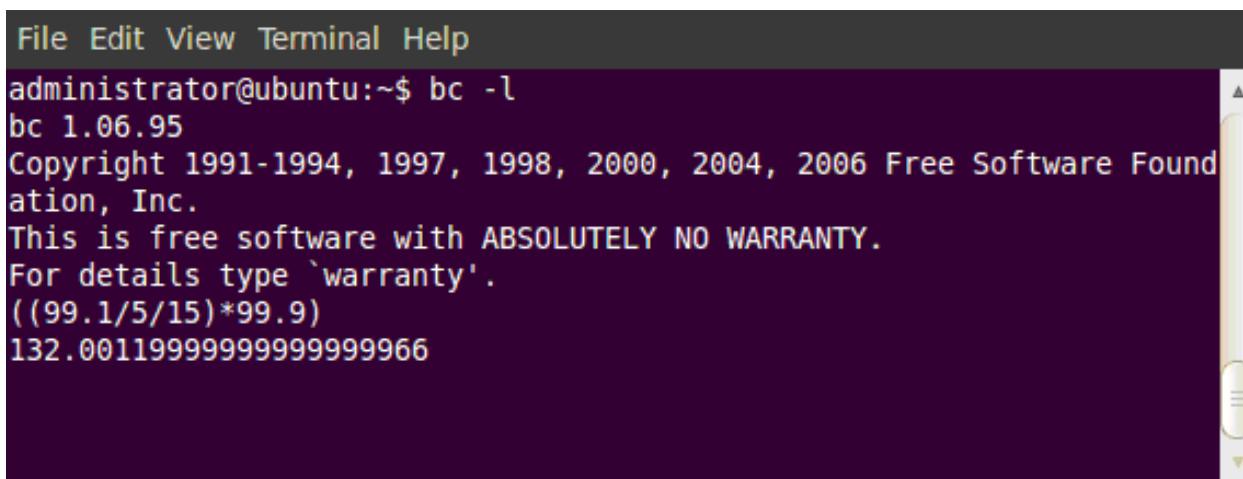
A screen similar to the one shown in figure 6.7 will be displayed. Notice that the dollar prompt is not visible on the screen; this indicates that the *bc* command is now ready to take input from you. The *-l* switch is used to include the standard math library.



```
File Edit View Terminal Help
administrator@ubuntu:~$ bc -l
bc 1.06.95
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006 Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
```

Figure 6.7 : Initiation of bc command

Now, just type the formula that you want to evaluate at the blinking cursor and the press Enter key. You can type a simple expression like $5 * 5$, or you may type a complex expression with grouped operators. Let us type the expression $((99.1 / 5.15) * 99.9)$, now press the Enter key. The command will display the output in the next line as shown in figure 6.8.



The screenshot shows a terminal window with a dark background. At the top, there's a menu bar with "File", "Edit", "View", "Terminal", and "Help". Below the menu, the terminal prompt is "administrator@ubuntu:~\$". The user then types "bc -l" and presses Enter. The terminal displays the version "bc 1.06.95" and copyright information from 1991-1994, 1997, 1998, 2000, 2004, 2006 Free Software Foundation, Inc. It also states that it is free software with ABSOLUTELY NO WARRANTY. For details type 'warranty'. The user then enters the expression "((99.1/5/15)*99.9)" and gets the output "132.0011999999999999966".

Figure 6.8 : Output of bc command

Observe that we still are not able to see the prompt; this simply means that we can continue working in command line calculator mode.

In addition to the normal mathematical functions like addition, subtraction, multiplication, division, modulus, and exponents, we can also use trigonometric or logarithmic functions like sine, cosine, arctangent, and log. For example, if you need to find the natural logarithm of value 2013, use the command **l(2013)** and you will get output **7.60738142563979148420**.

The *ibase* function allows us to set the numbering system that we want to use for input. Similarly the *obase* function allows us to set what numbering system to use for output. Let us try to convert numbers from decimal number system to hexadecimal number system. First, we need to set *obase* as shown below :

obase = 16

Now, type the number you want to convert to hexadecimal as shown in the example and press the Enter key.

256

100

Here 100 is the hexadecimal equivalent of decimal number 256.

Similarly if you want to convert this result to binary number system then just change the obase again as shown here.

obase=2

Now type **100** and the output will be **1100100**, observe that this is binary equivalent of decimal 100 and not 256. The reason for this is very simple, we have not changed ibase, and hence all entries are considered to be decimal entries.

To convert hexadecimal value 100 to its binary equivalent, set ibase and obase as shown below :

ibase=16

obase=2

Now type a hexadecimal number that you need to convert to binary. For example, type **100** and you will get **100000000** as a result.

To return back to decimal mode set ibase to 10. Execute the following to find out the square root of a number using sqrt function available in math library.

sqrt(256)

16.0000000000000000000000

As compared to graphical calculator the command line bc calculator is more faster and flexible. To return to the Linux command prompt, press CTRL+ d.

Displaying a message (*echo*)

We need to display message very frequently when using command prompt. The *echo* command is used to display a message on the terminal. For example, type the following command and press the Enter key. The string written after echo will be displayed on your monitor screen.

\$echo Hi, I am learning Ubuntu Linux

Hi, I am learning Ubuntu Linux

It is also possible to enclose the string within double quotes. The output will not contain the double quotes. The echo command can also be used to display values of variable. For example, define a variable named cost and assign it value 10 as shown below :

\$cost=10

Once you press the Enter key you will be returned to prompt. Now type the command given below :

\$echo The cost of product is Rs. \$cost

The cost of product is Rs. 10

To display the value of cost on the screen it is passed to the echo command. Notice that in the string we have written cost twice. The one that is prefixed with \$ symbol represents a variable, while the other is a normal string. When the echo command finds any string prefixed with a \$ symbol will consider it to be a variable. It will then try to print the value of variable.

The echo command can be used along with other commands to give meaningful output. For example we may combine the echo and the date command in the manner shown below :

\$echo Current time is ‘ date +%T ‘

Current time is 14 :55 :04

Observe that the command to be executed is placed within back quotes (quotes available on key with ~ sign), thus first the date command will be executed and then its result will be displayed using echo command.

Changing password (*passwd*)

A user needs to change password very often due to various reasons. The *passwd* command helps us perform this operation. It is used to change the password of the current login account by default. The following command allows you to change the password.

\$passwd

Once you press the Enter key a message similar to the one below will be shown, along with the blinking cursor.

Changing password for administrator

(current) UNIX password :

Type your current password and press the Enter key, Linux will check whether you have entered a valid password, and if you have then it will prompt for the new password. You will be asked to enter new password and retype it again as shown below :

Enter new UNIX password :

Retype new UNIX password :

passwd : password updated successfully

If you have typed the new password correctly and it does not conflict with any guidelines decided for password, your new password will be registered by the system. In case of any problem you may get an error message. It is also possible to change the password of other user of the system by specifying the username after the passwd command. For example, if we have a user named harshal, to change its password we may type the following command.

\$passwd harshal

In case of genuine user you will be allowed to change its password.

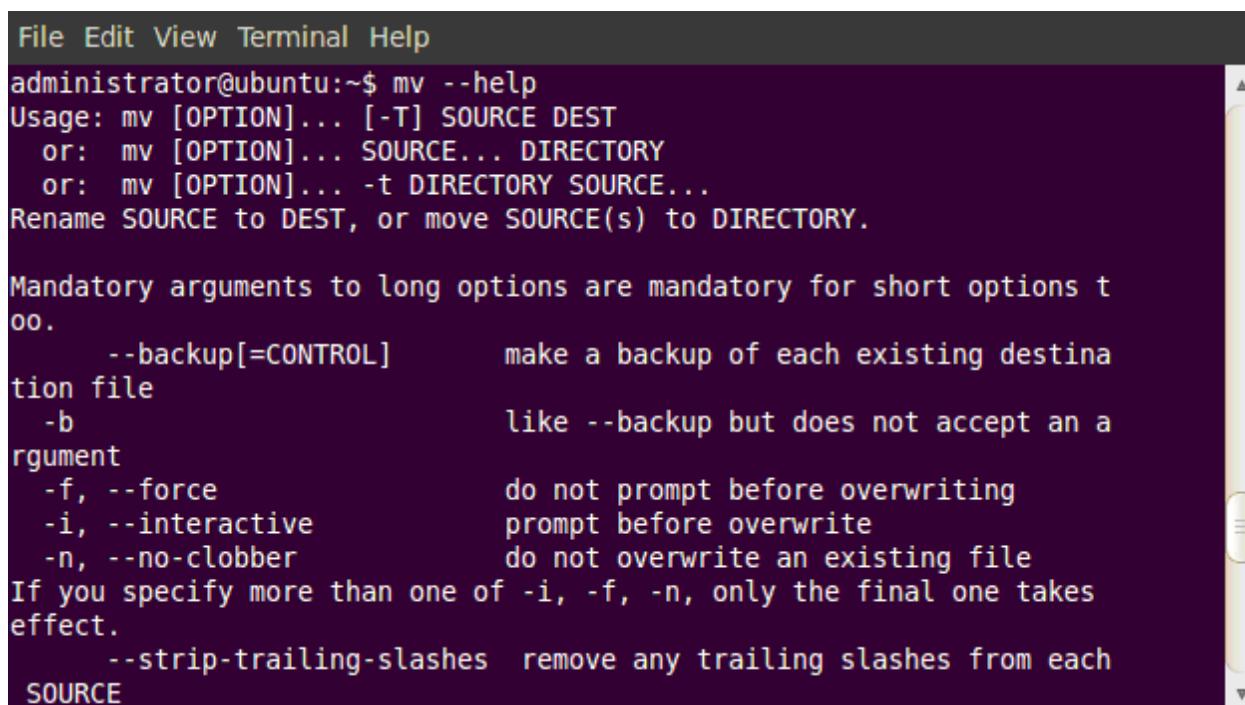
Clearing the Screen (*clear*)

While working on command prompt you must have observed that the screen often gets full. At times it also becomes difficult to see the output clearly, we have one simple solution for this problem, use the *clear* command to remove data on the screen.

\$clear

Getting Help on the Linux Commands

Before looking at any other commands first let us learn how to get help when using commands on Linux platform. Linux provides two inbuilt commands namely *help* and *man* to assist the user while working on the command line interface. All the commands that we use in Linux supports the *-h* (or *-help*) option. This option generates a small description of how to use the command. Figure 6.9 shows the use of help command.



The screenshot shows a terminal window with a dark background and light-colored text. The title bar includes 'File Edit View Terminal Help'. The main area displays the output of the 'mv --help' command. It starts with usage information for moving files between sources and destinations. It then details various options: '--backup[=CONTROL]' for making backups, '-b' for a backup-like behavior without a file argument, '-f, --force' for overwriting files without prompting, '-i, --interactive' for prompting before overwriting, '-n, --no-clobber' for not overwriting existing files, and '--strip-trailing-slashes' for removing trailing slashes from source paths. A note states that if multiple options like -i, -f, and -n are used, only the last one takes effect.

```
File Edit View Terminal Help
administrator@ubuntu:~$ mv --help
Usage: mv [OPTION]... [-T] SOURCE DEST
      or: mv [OPTION]... SOURCE... DIRECTORY
      or: mv [OPTION]... -t DIRECTORY SOURCE...
Rename SOURCE to DEST, or move SOURCE(s) to DIRECTORY.

Mandatory arguments to long options are mandatory for short options too.
      --backup[=CONTROL]      make a backup of each existing destination file
      -b                      like --backup but does not accept an argument
      -f, --force              do not prompt before overwriting
      -i, --interactive        prompt before overwrite
      -n, --no-clobber         do not overwrite an existing file
If you specify more than one of -i, -f, -n, only the final one takes effect.
      --strip-trailing-slashes remove any trailing slashes from each SOURCE
```

Figure 6.9 : Use of help command

Observe that the command used to display the help in figure 6.9 is *mv - - help*. The alternate mechanism to get help on the command is to use Linux online manuals. The *man* command activates a manual corresponding to a specific command that we need to look at. For example, the command *man mv* will show us the manual for the move command. Figure 6.10 shows the output of *man* command.

Observe that *man* command gives us exhaustive information of a command and may run into multiple screens. It generally displays one page at a time, as we press the Enter key the contents scroll down. To come out of the manual screen type alphabet 'q', this will take you back to the command prompt.

```
File Edit View Terminal Help
MV(1) User Commands MV(1)

NAME
mv - move (rename) files

SYNOPSIS
mv [OPTION]... [-T] SOURCE DEST
mv [OPTION]... SOURCE... DIRECTORY
mv [OPTION]... -t DIRECTORY SOURCE...

DESCRIPTION
Rename SOURCE to DEST, or move SOURCE(s) to DIRECTORY.

Mandatory arguments to long options are mandatory for short options
too.

--backup[=CONTROL]
make a backup of each existing destination file
Manual page mv(1) line 1
```

Figure 6.10 : Use of man command

In case we want only small description of a command then we may use the *whatis* command. It gives us one line explanation of the command, but omits any additional information about options. Figure 6.11 shows the sample output of *whatis* command.

```
File Edit View Terminal Help
administrator@ubuntu:~$ whatis mv
mv (1)           - move (rename) files
administrator@ubuntu:~$
```

Figure 6.11 : Sample output of whatis command

Many times it may happen that we may not know which command to look for exactly. In such situations we may use the *apropos* command. The syntax of the command is mentioned below :

\$apropos string

When we execute this command we will get a list of all the commands that has the string as the part of the command or command description. For example if we type *apropos copy* on the command prompt and try to execute it, then we may get screen full of commands that have copy as a string within the command or its description. A user must take caution while using this command. Sample output of *apropos* command is shown in figure 6.12.

```

File Edit View Terminal Help
administrator@ubuntu:~$ apropos copy
bcopy (3)           - copy byte sequence
bf_copy (1)          - shell script to copy a bogofilter working...
bf_copy-bdb (1)      - shell script to copy a bogofilter working...
copysign (3)         - copy sign of a number
copysignf (3)        - copy sign of a number
copysignl (3)        - copy sign of a number
cp (1)               - copy files and directories
cpgr (8)              - copy with locking the given file to the p...
cpio (1)              - copy files to and from archives
cppw (8)              - copy with locking the given file to the p...
dd (1)               - convert and copy a file

```

Figure 6.12 : Sample output of apropos command

Working with Directories

In Linux a directory is a special type of file that is used to store files and other directories. Here ‘ / ‘ symbol represents the root directory. All other directories come under root directory. Let us learn how to work with directories using terminal window.

Home directory

When a user logs on to the system, Linux automatically places the user in the directory called the home directory. It is created by the system at the time when a user account is created and generally will have a path /home/username. Here the username refers to the login name. For example if the username is harshal, then the home directory will be /home/harshal. It is possible to change this path if needed. The default working directory path is stored in system variable named HOME. We can cross check the directory by using the echo command.

\$echo \$HOME

/home/harshal

Note that the path displayed using this command is an absolute path name. Absolute path name is a sequence of directory names separated by / (slashes). An absolute path name shows a location in reference to the root directory. The first slash (/) is synonymous to root directory while the other slashes act as delimiters to other directory names. Thus the directory harshal is located within directory home which further is located in the root directory. Similarly the directory **/home/administrator** refers to home directory of username administrator.

It is possible to change the default path of the home directory. Suppose that you are able to see the output as shown below :

/home/its/ug1/svics

Here *svics* (home directory of user *svics*) is within the sub directory *ug1* (a directory that represents a sub group), which further is within a directory *its* (a directory that represents a group). Directory home and ‘/’ have their default meaning.

Present Working Directory (*pwd*)

After we log into a system we can move around from one directory to another. But at any given point of time we will be located only in one directory. The directory where we are located at that moment is known as *current directory* or *present working directory*. To know the current directory that we are working in we can use the *pwd* command.

\$pwd

Creating a Directory (*mkdir*)

A directory in Linux can be created using the *mkdir* command. The command takes the name of the directory to be created as its argument. Let us create a directory named *subject*.

\$mkdir subject

The power of command line over GUI lies in its flexibility. If we create a directory using GUI we will be able to create one directory at a time, while it is possible to create multiple directories using a single *mkdir* command. The following command syntax illustrates the same.

\$mkdir animals birds vehicles plants

The command when executed will create four directories named animals, birds, vehicles and plants in the current directory.

Change Directory (*cd*)

In the case when we need to store any data within a directory, first we need to make it our current directory. We can change (go within) a specific directory using the *cd* command. Let us try to create a directory named *math*, *science* and *economics* within a directory *subject*. To create these directories first we need to be in the *subject* directory. The command sequence shown below allows us to perform the said operation.

\$cd subject

\$pwd

/home/administrator/subject

\$mkdir math science economics

Observe that the user name here is administrator. To again come back to the *administrator* directory, simply type the command below :

\$cd ..

In the above command double dots (..) refer to the parent directory. Note that there should be one space between the cd command and the double dot.

Assume that you are in some internal directory that has path /home/administrator/subject/economics and you need to come back to the users home directory then again the cd command comes in handy. To come back to home directory we can issue the cd command twice as shown below :

\$cd ..

\$cd ..

The command sequence here will take us back one level at time. If we are say M levels down within the home directory, then we will have to execute the cd command M times. An alternative approach is to use a single cd command as shown below :

\$cd ../../

Some example usage of the cd command along with its description is given in table 6.1.

Command Issued	Action Performed
cd ~/Desktop	Changes directory to /home/username/Desktop, from any current path. Here the symbol ~ refers to home directory of the user.
cd /	Changes directory to the root directory from any current path.
cd	Changes directory to the home directory from any current path.
cd -	Changes directory to the previously changed directory.
cd /var/www	Changes directory directly to the www sub-directory with directory var. It is useful when we know the path explicitly.

Table 6.1 : Sample cd commands

Remove Directory (*rmdir*)

An empty directory can be deleted by using the command *rmdir*.

\$rmdir science

Here science is a directory name, and it will be removed using the above command only if it is empty. In case it is not empty we will get a message ‘*rmdir : failed to remove ‘science’ : Directory not empty*’. In this case we have to first delete all the contents within the directory and then reissue this command. Note that it is also possible to delete multiple empty directories in the same way we created them.

To delete a non-empty directory with all its contents we can use the *rm* command as shown below :

\$rm -r science

The *rm* command is discussed in detail in the later part of chapter.

Naming Conventions in Linux

We have created directories with different names using the *mkdir* command. While creating any object like a directory or a file in Linux we have to follow certain rules. On most of the Linux systems today, a name (of directory or file) can consist of up to 255 characters. Unlike Windows OS, names in Linux can practically consist of any ASCII character except for the slash (/) and the NULL character. Any other control characters or nonprintable characters are permitted. Examples of some valid names are .name, ^myname^-++, -{}(), test\$#, xy.ab.ef etc.

However, it is recommended that a name should be relevant and contain only alphabetic characters, numerals, period (.), hyphen (-) and underscore (_). Linux strictly adheres to case sensitivity, thus math, Math and MATH are three different names. It is possible for the directories with these names to coexist at same level. If these names refer to a file then they can again coexist in the same directory.

Working with Files

Directory generally works as a container. The data is normally stored in files, which further may be stored in directory for proper arrangement and easy access. Text editors like nano, pico, vi or vim, ed and others are generally best suited for creating text file. However, many times the user needs to create a file quickly. The *cat* command comes in handy in such scenario. It is mainly used to display the contents of a small file on the terminal. But it can also be used to create a file, concatenate two files and append contents into a file.

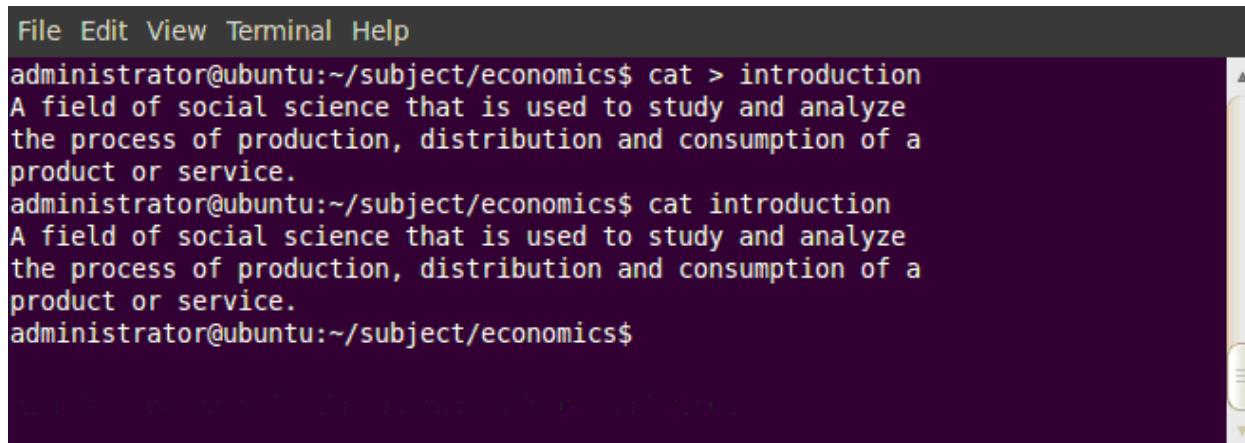
Create a file using *cat* command

Let us create a file named *introduction* within the directory *economics*. To create this file you will have to first make the directory *economics* as your current directory. You can use the *cd* command here. Now type the command *cat* followed by a greater than (>) symbol and name of the file as shown below :

\$cat > introduction

When we execute this command the cursor will be positioned in the next line, waiting for us to type the contents of the file. Type the text that you want to store in file and press *CTRL + d*. This will take you back to the command prompt. The combination *CTRL + d* in Linux indicate the end of file character. The greater than (>) symbol used in the above command is known as

redirection operator. It is used to instruct the shell that a redirection is required i.e., input should go to the specified file. The cat command when used without the greater than (>) symbol, displays the contents of the filename specified in argument. Figure 6.13 shows the process of creating and then displaying a file.



The screenshot shows a terminal window with a dark background and light-colored text. At the top, there's a menu bar with 'File', 'Edit', 'View', 'Terminal', and 'Help'. Below the menu, the terminal prompt is 'administrator@ubuntu:~/subject/economics\$'. The user runs the command 'cat > introduction', which creates a new file named 'introduction' containing the text 'A field of social science that is used to study and analyze the process of production, distribution and consumption of a product or service.'. After this, the user runs 'cat introduction', which displays the same text. Finally, the user exits the terminal with 'administrator@ubuntu:~/subject/economics\$'.

Figure 6.13 : Creating and displaying file using cat command

Appending contents using cat command

Assume that you have an existing file and you want to add some more content in the file. The cat command can be used again here with one simple change. The redirection operator used previously is to be replaced by append output (>>) redirection operator. The command to append data in the *introduction* file is shown below :

\$cat >> introduction

An alternate definition states that Economics is a science which studies human behaviour as a relationship between ends and scarce means which have alternative uses.

[CTRL+d]

Note that if the file already exists and we use the command *cat > filename* then, the existing contents will be overwritten with the new one. So it is necessary to be careful while opening a file that already has some contents.

Concatenating multiple files using cat command

The cat command can also be used to concatenate the contents of multiple files and store it in another file. The syntax of using the concatenation is shown below :

\$cat file1 file2 > file3

The above command will create *file3* that contains the text of both the files, namely *file1* and *file2*. The new file created will have contents based on the sequence of filenames. Here the initial contents will be of *file1*, after which the contents of *file2* will be appended.

Deleting a File (*rm*)

The *rm* command is used to delete/remove one or more files. For example to delete the file *introduction*, execute the following command :

\$rm introduction

We can also delete multiple files using a single rm command. For instance the command

\$rm file1 file2 file3

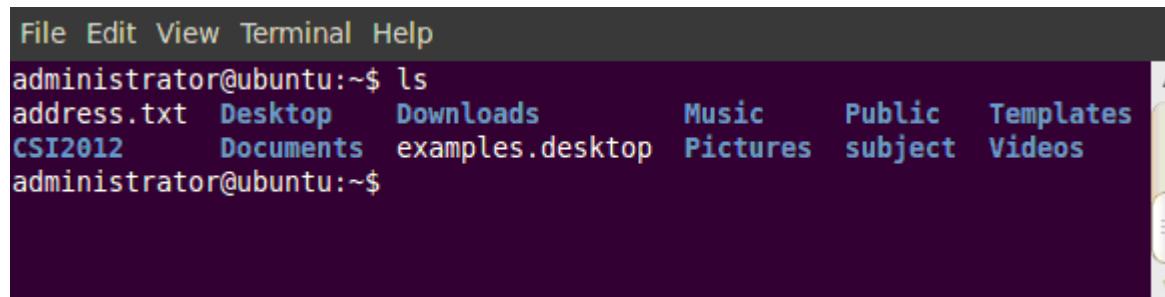
will delete all the three files supplied to it as argument. Table 6.2 gives some options that can be used along with the rm command.

Option	Usage
<code>rm -i Filename</code>	Deletes file using interactive mode. The user will be asked to verify the delete operation.
<code>rm -r Directoryname</code>	Deletes directory and along with all its contents.
<code>rm -r *</code>	Deletes all contents (file and/or directory) within the current directory. Here the symbol * is known as wildcard character. This is a very dangerous command as it will delete all the files and directories within the current directory, hence such command should be used only if you certain about the action and the result.
<code>rm -rf *</code>	Same as <code>rm -r</code> , but also deletes the contents even if it is write protected.

Table 6.2: Some options of rm command

Moving around the File system

So far we have learned how to create and delete a directory or file. Now let us see how to view the contents that are part of our file system. The *ls* command gives us the list of the contents in the current or a specified directory. The *ls* command can be used with different options to change the output. Let us begin with the plain and simple *ls* command without any options. Figure 6.14 shows the output of simple *ls* command. Note that the output on your computer may vary.



A screenshot of a terminal window on a Linux system. The window has a dark background with light-colored text. At the top, there is a menu bar with options: File, Edit, View, Terminal, and Help. Below the menu, the terminal prompt is shown as "administrator@ubuntu:~\$". The command "ls" is entered, and the output lists several files and directories: "address.txt", "Desktop", "Downloads", "Music", "Public", "Templates", "CSI2012", "Documents", "examples.desktop", "Pictures", "subject", and "Videos". The "CSI2012" directory is highlighted in blue. The prompt "administrator@ubuntu:~\$" appears again at the bottom of the list.

Figure 6.14 : Output of the ls command

Let us now create a file named `.introduction` and then check whether we are able to list it or not. Type the command shown below to create the file.

```
$cat > .introduction
```

Learning Ubuntu Linux is fun....

[CTRL + d]

Now use the ls command again to list the contents of a file. You must have observed that the file recently created is not visible on the screen. Note that in Linux any filename that is preceded by a ‘.’ is treated as a hidden file. To list hidden files in the current directory we need to use -a option of the ls command. Figure 6.15 shows how to list hidden files.

```
File Edit View Terminal Help
administrator@ubuntu:~$ ls -a
. examples.desktop Pictures
.. .fontconfig .pki
address.txt .gconf .printer-groups.xml
.adobe .gconfd .profile
.aptitude .gksu.lock Public
.avast .gnome2 .pulse
.bash_history .gnome2_private .pulse-cookie
.bash_logout .gstreamer-0.10 .recently-used.xbel
.bashrc .gtk-bookmarks .Skype
.cache .gvfs .ssh
.compiz .ICEauthority .subject
.config .icons .sudo_as_admin_successful
CSI2012 .introduction .swp
 dbus .java Templates
.debtags .local .themes
Desktop .macromedia .thumbnails
.dmrc .mozilla .update-notifier
Documents Music Videos
Downloads .nautilus .xsession-errors
.esd_auth .openoffice.org .xsession-errors.old
.evolution .padminrc
administrator@ubuntu:~$
```

Figure 6.15: Listing hidden files

Observe the difference in the output of figure 6.14 and figure 6.15. Note that figure 6.15 shows more number of files than shown in figure 6.14. The two entries ‘.’ and ‘..’ visible in figure 6.15 are of importance, these two entries are automatically created in the directory whenever the directory is created. Table 6.3 gives some options that can be used along with the ls command.

Option	Usage
ls ~	Lists the files that are in user's home directory.
ls [svics]*	Lists all the files in which the first character of the filename matches with any of the given alphabets within the square brackets. The remaining part of filename can contain any valid ASCII character.
ls [n-s][5-7]??	Lists all files with 4 character filename. With the condition that the first character is in the range n to s, second character is in the range of 5 to 7, whereas the third and fourth characters are any valid ASCII character.
ls -r	List the files by sorting them in reverse order.
ls -t	List the files by sorting them based on their modification time.
ls -F	List the files and mark all executable files with * and directories with / symbol.
ls -l	List one file per line.

Table 6.3: Some options of ls command

Pattern Matching – The wildcards

In the above discussion you have already seen the usage of characters asterisk (*) and question mark (?). These characters are known as wildcard characters used for matching a pattern as required by the user. Table 6.4 summarizes the working of wildcards used by shell.

Wildcard	Pattern to be matched
*	Any number of characters including none
?	A single character
[abc]	A single character – either a, b or c (user can use other characters also).
[!abc]	A single character <i>other than</i> a, b or c (user can use other characters also).
[p-s]	A single character within the ASCII range of the characters p to s (user can use other characters also).
[!p-s]	A single character that is not within the ASCII range of the characters p to s (user can use other characters also).

Table 6.4: The wildcard characters

Manipulating Files and Directories

In the previous section we learned how to create and delete a file or a directory. Let us now see how to perform operations like copy, move, and assign permission on them.

Copying a file (*cp*)

Very often we need to create a replica of the data that we have generated, the *cp* command copies a file or group of files specified as an argument to it. It creates an exact replica of a file on the disk at the location specified by the user. The *cp* command needs at least two arguments. The first argument refers to a source file while the second argument refers to a destination file. Let us create a copy of file *introduction* using the following command :

\$cp introduction new_introduction

After execution of the above command, an exact copy of file *introduction* will be created with the name *new_introduction*. If a file with the name *new_introduction* already exists, it will simply be overwritten without any warning from the system. In case no such file exists a new file will be first created and then the contents of the file *introduction* will be copied in it.

The *cp* command can also be used to copy more than one file into a specified directory. For instance, the following command :

\$cp file1 file2 my_dir

will copy two files named *file1* and *file2* in a directory named *my_dir*. It is necessary that the directory *my_dir* already exists, or else we will get an error message. Table 6.5 gives some example usage of the *cp* command.

Command	Description
<i>cp /vol/examples/tutorial/science.txt .</i>	Copies the file <i>science.txt</i> to current directory. The dot (.) at the end refers to the current directory.
<i>cp chap01 progs/unit1</i>	A file named <i>chap01</i> is copied within the directory <i>progs</i> with name <i>unit1</i> . (no directory with name <i>unit1</i> should exist in <i>progs</i> directory)
<i>cp chap01 progs</i>	A file named <i>chap01</i> is copied within the directory <i>progs</i> with same name. (because <i>progs</i> is a directory).
<i>cp -r progs newprogs</i>	The directory named <i>progs</i> along with all its contents is copied and stored as a directory <i>newprogs</i> .

Table 6.5 : Sample cp commands

Renaming files and/or moving files (*mv*)

Changing the name of a file or directory is another operation that user performs regularly. The *mv* command is used for renaming a file or directory. For example, to rename the file *introduction* to *introduction.txt*, execute the following command :

\$mv introduction introduction.txt

The command will rename the file and store it at the same location as that of the file *introduction*. Thus no additional space is consumed on disk during renaming.

The *mv* command can also be used to move a file or group of files to a different directory. For example, the command

\$mv file1 file2 my_dir

will move the files named *file1* and *file2* to the directory named *my_dir*.

This command is also used to rename a directory. For instance, the execution of the following command :

\$mv math mathematics

will rename an existing directory *math* to *mathematics*.

Paging output (*more*)

The *more* command is used to view one page of content on the screen at a time. For instance if the file *introduction.txt* contains text that can not fit in a single screen, then reading its contents would become difficult. The *more* command when used displays the contents of file one page at a time. To view the next page we may press any key. Generally we may press character ‘b’ to view previous page and character ‘f’ to view next page. The sample usage of the command is shown below :

\$more introduction.txt

Compare two files (*cmp*)

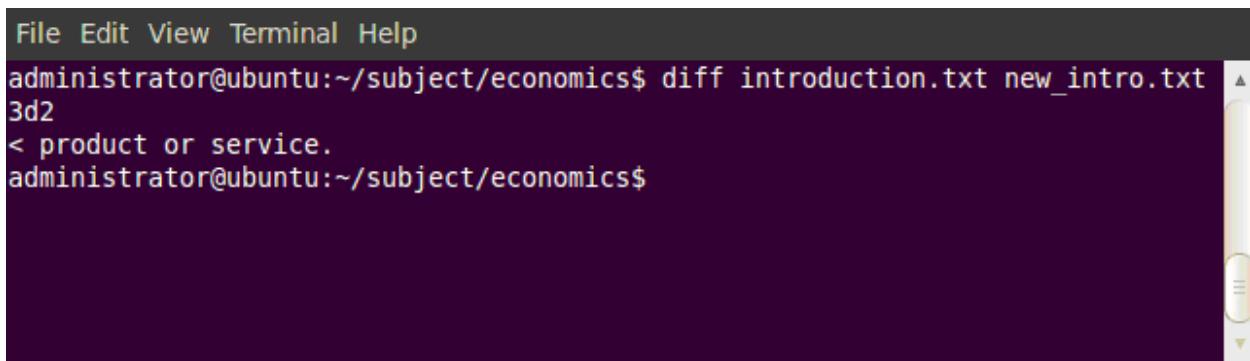
The *cmp* command compares two files of any type and writes the results to the standard output. If the two files compared differ in contents then the byte and line number at which the first dissimilarity occurred is reported. In case there is no difference between the contents of the files we simply see the command prompt again. The sample usage of the command is shown below :

\$cmp introduction introduction.txt

Difference (*Diff*)

An extension of the *cmp* command is the *diff* command. The *diff* command compares two files and displays the contents of both files indicating where the difference lies. To understand the working of the *diff* command we have created a copy of the file *introduction.txt* and named it *new_intro.txt*. We have also removed some lines from the new file. Figure 6.16 shows the output of the command shown below :

\$diff introduction.txt new_intro.txt



The screenshot shows a terminal window with a dark background. At the top, there is a menu bar with options: File, Edit, View, Terminal, and Help. Below the menu, the terminal prompt is "administrator@ubuntu:~/subject/economics\$". The command entered is "diff introduction.txt new_intro.txt". The output of the command is displayed, showing the difference between the two files. The output includes the line count (3d2), followed by a line starting with a less than sign (<) indicating that the line is present in introduction.txt but not in new_intro.txt. The line is "< product or service.".

Figure 6.16 : Output of diff command

In figure 6.16 the lines beginning with a < indicates that file introduction.txt contains the text shown but file new_intro.txt does not contain the line. Any changes in the file new_intro.txt would be shown with the lines beginning with a > sign.

Counting lines, words and characters in a file (wc)

The **wc** command is used to count the number of lines, words, and characters in the specified file or files. The **wc** command can be used along with three options -l, -w and -c for counting lines, words and characters respectively. For instance, execute the following command to count the number of lines in the file introduction.txt.

\$wc -l introduction.txt

4 introduction.txt

Similarly the commands **wc -w introduction.txt** and **wc -c introduction.txt** will give us the count of number of words and characters in the file. To get all the information together we can use the command as shown below :

\$wc -l -w -c introduction.txt

4 49 307 introduction.txt

File permissions

In the earlier section, we have seen options which can be used with ls command. The ls command has several other options also. For example the following command when executed may result in the output similar to the one shown here.

\$ls -l

total 6

-rw-r--r-- 1 administrator administrator 313 2013-02-15 18 :04 about_Gandhiji.txt

-rw-r--r-- 1 administrator administrator 444 2013-02-15 18 :19 introduction.txt

-rw-r--r-- 1 administrator administrator 401 2013-02-20 16 :43 address.txt

drwxr-xr-x 1 administrator administrator 4096 2013-02-21 18 :15 backup

-rw-r--r-- 1 administrator administrator 144 2013-02-13 18 :49 city.txt

-rw-r--r-- 1 administrator administrator 226 2013-02-20 14 :11 script10.sh

Observe the output; it gives us a clear idea about an object in our file system. An object can be categorized as a regular file, a directory or a process. It shows us the owner of the object, size of the object, date and time on which the object was created along with the name of the object. Let us try to understand the file permission in detail. Figure 6.17 shows the relation of different permissions w.r.t. owner, group and other users.

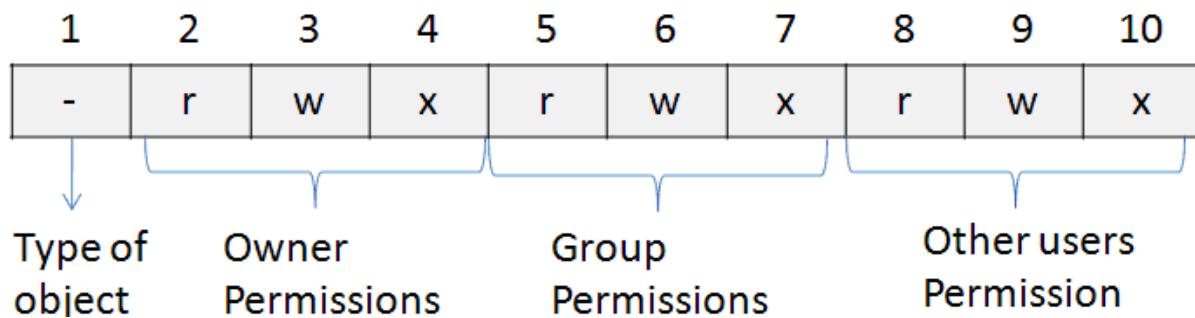


Figure 6.17 : File permissions

As can be observed in figure 6.17 the first column refers to type of object. The character ‘-’ in the first column refers to a file, character ‘d’ refers to a directory and character ‘p’ refers to a process. The next nine characters tell the system what access is permitted for this object; hence the name “permissions”. An object in Linux has three permissions namely, read (r), write (w) and execute (x).

The set of three characters after the file type shown in column 2 to 4 tells which permission the owner of the file has. Owner of the file is the user who created the file (administrator in our case). The character **r** in the first position means you are permitted to read the file. A **w** in the second position means you may write in the file. This includes the ability to delete a file. An **x** in the third position means you may execute the file. A hyphen ‘-’ in any position means that you don’t have that particular permission. As you can in the output of the `ls -l` command, administrator the user who owns the file can read and write files like `about_Gandhiji.txt`, `introduction.txt` and `address.txt`. The characters in column 5 to 7 denotes permission given to the group, to which user belongs. Similarly the characters in column 8 to 10 denote permission given to any other user or group. Generally the term others refer to the users of the system that do not belong to group to which the owner of the file belongs. A user if wishes can change the permission of an object that he/she owns.

The file permissions can also be given as numeric representation. We use octal (base of 8) number system for representing permissions as numeric values. Every octal digit combines read, write and execute permissions together. For example, in permission 644, “6” refers to the rights of the owner, “4” refers to rights of the group and “4” refers to rights of others. The permission 0644 when assigned is interpreted as read and write permission to owner, only read permission to group and others. Table 6.6 shows the interpretation of the octal numbers when used as permission.

Permission in text mode	Permission in octal mode	Meaning
---	0	No permission assigned
--x	1	Only execute access is allowed
-w-	2	Only write access is allowed
-wx	3	Write and execute access are allowed
r--	4	Only read access is allowed
r-x	5	Read and execute access are allowed
rw-	6	Read and write access are allowed
rwx	7	Everything is allowed

Table 6.6 : Octal numbers and permission

Changing Permissions (*chmod*)

To change the permission we use the *chmod* command. The operation of changing the permission is also known as change mode operation. For instance, in the above example we have seen that user (owner) has read and write permissions on the file. To make the file read only file the following command can be used :

\$chmod ugo-w introduction.txt

The character ‘u’ in the above chmod command stands for user, ‘g’ for group and ‘o’ for other. After executing the command if we again list the file, then the output will be similar to the one shown below.

\$ls -l introduction.txt

-r--r--r-- 1 administrator administrator 307 2013-02-11 14 :19 introduction.txt

Please note that write operation will not be permitted on this file. Additionally it also prevents user from deleting the file intentionally or unintentionally.

To assign write and execute permission to the owner of a file, execute the following command.

\$chmod u+wx script10.sh

\$ls -l script10.sh

-rwxr--r-- 1 administrator administrator 226 2013-02-20 16 :05 script10.sh

Here the file script10.sh is known as a script file. We are going to learn about shell scripting in the next chapters.

Table 6.7 shows some abbreviations and its meaning when used with the chmod command.

Category	Operation	Permission
u-user	+ assign permission	r- read permission
g-group	- remove permission	w - write permission
o-other	= assign absolute permission	x - execute permission
a-all		

Table 6.7 : Abbreviation used by chmod

I/O Redirection

A user interacts with the Operating System using a standard input device (keyboard). The Operating system displays the output on standard output device (monitor). Thus if any command is executed its input will be taken from the keyboard and output will be displayed on the monitor.

Sometimes it is useful to redirect the input or output to a file or a printer. Linux provides redirection symbols to change the standard input flow. The greater than symbol ‘ > ‘ implies redirection of output. It instructs the OS to put the output in the destination (file) specified by the user instead of displaying on the monitor screen. Similarly the less than symbol ‘ < ‘ implies redirection of input, it instructs the OS to accept the input from the specified source (file) instead of keyboard.

Assume that we issue a command **wc -l < introduction.txt**, here we are instructing OS to accept the input from a file named introduction.txt instead of the keyboard. Similarly the command **ls > list.txt** when executed will transfer the output of ls command to a file named list.txt instead of the monitor. When output redirection is used the output will not be displayed on the monitor hence to see the output we will have to use the command **cat list.txt**.

Piping

Redirection facility discussed above helps in associating the Linux commands to files. Many times we need to use multiple commands to perform a single operation. The piping facility of Linux helps in such cases. The pipe symbol (|) is used to provide the output of one command as an input to another command. The process of converting output of one command into input of another command is known as piping. Let us see an example of piping.

\$ls | wc -l

When we execute the above command, the output of *ls* command becomes the input to the *wc* command. Thus we will get information of total number of files in a current directory. Real power of pipe facility can be availed when we use it along with filters. Filters have been discussed in the next section.

Filters

Filters are commands that accept data from the standard input, process or manipulate it and then write the results to the standard output. Various filters like head, tail, cut, paste, sort and uniq are available in Ubuntu Linux. Let us see the working of these filters.

Displaying lines from top of the file (*head*)

The head command is used to display the required number of lines at the beginning of the file based on user's requirement. When used without any option it displays first 10 lines of the file. To display the lines as per users requirement we need to pass an argument to the head command. For example to display first 2 lines of the file *introduction.txt*, execute the following command :

```
$head -2 introduction.txt
```

Displaying lines from bottom of the file (*tail*)

The tail command works exactly opposite of the head command. It displays specified number of lines from the end of the file. To display last 2 lines of the file *introduction.txt*, execute the following command :

```
$tail -2 introduction.txt
```

We can use the tail command to display lines from n^{th} line within the files. For example if we execute the following command :

```
$tail +5 introduction.txt
```

 will display lines from 5th line onwards from the file *introduction.txt*.

Slicing a file vertically (*cut*)

The head and tail command discussed in the above sections are used to slice the file horizontally. We can slice the data within the file vertically using the *cut* command. The cut command gives exact and precise outputs if the file has specific delimiters. Let us create one such delimited file to understand the working of the cut command. Create a file named *address.txt* using cat command that stores the data as shown :

```
$cat address.txt
```

```
20013, Vaidehi, Sanjay, Shah, Sector-23, GH-6, Gandhinagar, 382023  
20014, Dhrumil, Ajay, Patel, Yesh Enclave, Mota Bazar, Vidyanagar, 388120  
20015, Harshit, Amit, Jain, 58, Jaldeep I, Ahmedabad, 380058  
20016, Abdul, Shamsher, Khan, Khan Villa, M G Road, Nadiad, 388011  
20017, Nirav, Jose, Mackwan, Jose House, M G Road, Nadiad, 388011  
20018, Vidita, Harshal, Arolkar, 17, Jaldeep I, Ahmedabad, 380058
```

Let us see how to use the cut command along with its various options.

Cutting Characters (-c)

To extract specific characters from each line of the file, cut command with -c option is used. For instance, to extract the roll numbers and first names from the file *address.txt* execute the following command :

```
$cut -c 1-15 address.txt
```

20013, Vaidehi,
20014, Dhrumil,
20015, Harshit,
20016, Abdul, S
20017, Nirav, J
20018, Vidita,

Though the output looks fine to certain extent, it is not exactly the same as we would have expected. Look at the data of Abdul and Nirav it has additional characters. The **-c** option is useful for fixed length fields, we had problem in the output as the first names were not stored using a fixed length.

Cutting fields (-f)

To overcome the problem mentioned in the **-c** option we may use a delimiter. The cut command can treat values separated by the delimiter as separate field values. Observe that we have used comma ‘,’ as a delimiter in the address.txt file. Thus, to extract roll number and first name only we may execute the following command :

```
$cut -d "," -f 1,2 address.txt
```

20013, Vaidehi
20014, Dhrumil
20015, Harshit
20016, Abdul
20017, Nirav
20018, Vidita

Observe that we have got the desired output now. In this command the **-d** is option used to specify delimiter appearing in a file (comma in our case) and **-f** option is used to specify field numbers to be displayed (roll number (1) and first name (2) in our case).

It is also possible to slice the file vertically from fields in between. For example assume that we want to display the first name, city and pin-code then we need to cut field numbered 2 and 7 onwards. The said operation can be performed by executing the following command :

```
$cut -d "," -f 2,7- address.txt
```

Vaidehi, Gandhinagar, 382023
Dhrumil, Vidyanaagar, 388120
Harshit, Ahmedabad, 380058
Abdul, Nadiad, 388011
Nirav, Nadiad, 388011
Vidita, Ahmedabad, 380058

Here 7- in the command imply that we need to display all fields after field number seven (including seven) from the file address.txt. It is also possible to redirect the output to a file. For example if we execute the following command :

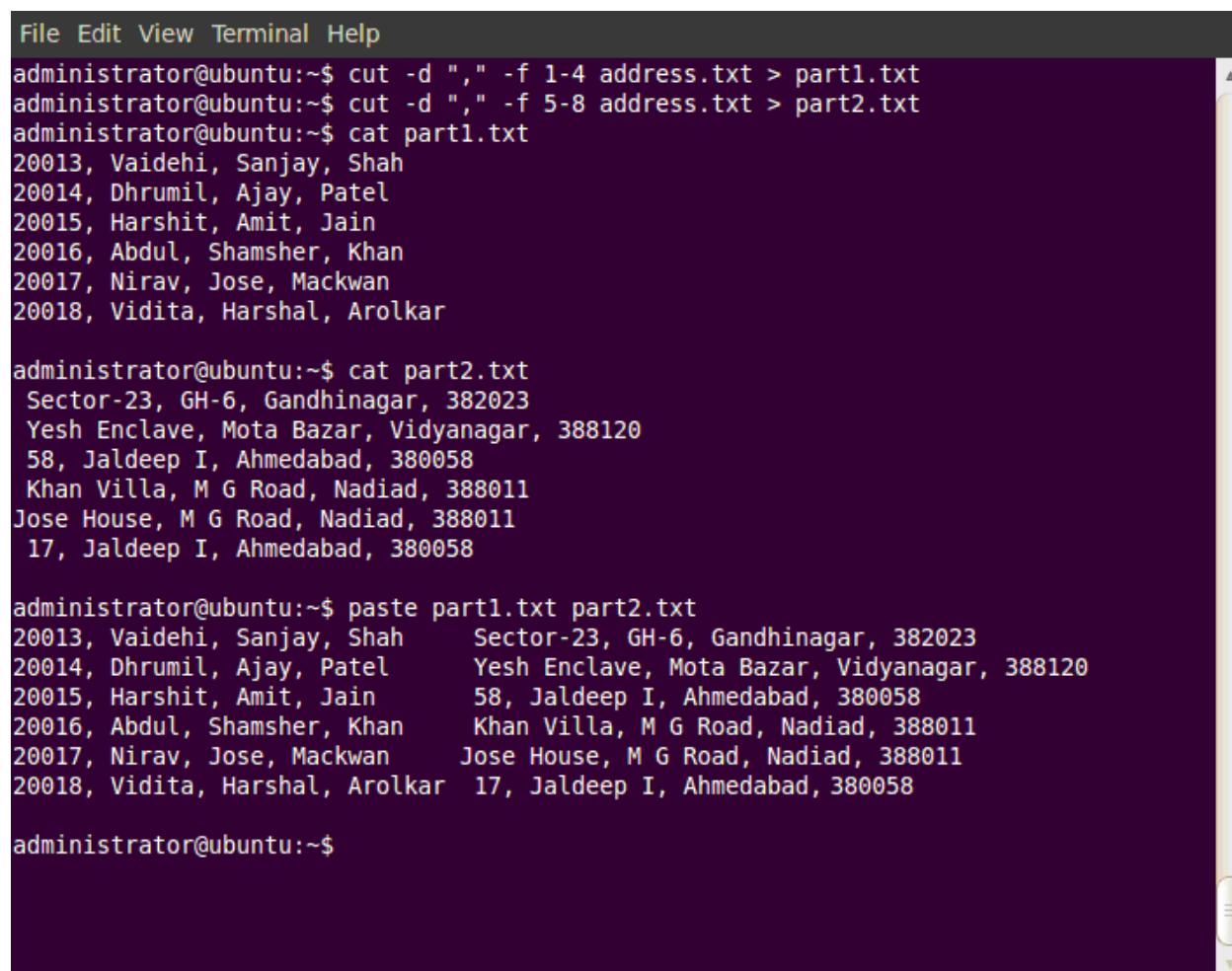
```
$cut -d "," -f 2,7- address.txt > output_cut.txt
```

the output instead of being displayed on monitor will be transferred to the file output_cut.txt.

Joining Contents (*paste*)

Two files can be pasted together using the *paste* command. For the *paste* command to work properly we need to ensure that both the files have exactly the same number of lines. If the number of lines is not same then the command may not result into expected output as it pastes from top of the files.

We will first create two different files named part1.txt and part2.txt using cut command and then join them using the *paste* command. Figure 6.18 shows the process of performing this operation.



The screenshot shows a terminal window with a dark background and light-colored text. At the top, there's a menu bar with options: File, Edit, View, Terminal, and Help. Below the menu, the terminal prompt is "administrator@ubuntu:~\$". The user runs three commands to create two files:

```
administrator@ubuntu:~$ cut -d "," -f 1-4 address.txt > part1.txt
administrator@ubuntu:~$ cut -d "," -f 5-8 address.txt > part2.txt
administrator@ubuntu:~$ cat part1.txt
```

The output of the third command is:

```
20013, Vaidehi, Sanjay, Shah
20014, Dhrumil, Ajay, Patel
20015, Harshit, Amit, Jain
20016, Abdul, Shamsher, Khan
20017, Nirav, Jose, Mackwan
20018, Vidita, Harshal, Arolkar
```

Then, the user runs a command to join the files:

```
administrator@ubuntu:~$ cat part2.txt
```

The output of the fourth command is:

```
Sector-23, GH-6, Gandhinagar, 382023
Yesh Enclave, Mota Bazar, Vidyanaagar, 388120
58, Jaldeep I, Ahmedabad, 380058
Khan Villa, M G Road, Nadiad, 388011
Jose House, M G Road, Nadiad, 388011
17, Jaldeep I, Ahmedabad, 380058
```

Finally, the user runs:

```
administrator@ubuntu:~$ paste part1.txt part2.txt
```

The output of the fifth command is:

20013, Vaidehi, Sanjay, Shah	Sector-23, GH-6, Gandhinagar, 382023
20014, Dhrumil, Ajay, Patel	Yesh Enclave, Mota Bazar, Vidyanaagar, 388120
20015, Harshit, Amit, Jain	58, Jaldeep I, Ahmedabad, 380058
20016, Abdul, Shamsher, Khan	Khan Villa, M G Road, Nadiad, 388011
20017, Nirav, Jose, Mackwan	Jose House, M G Road, Nadiad, 388011
20018, Vidita, Harshal, Arolkar	17, Jaldeep I, Ahmedabad, 380058

Administrator@ubuntu:~\$

Figure 6.18 : Example of *paste* command

Ordering Output (*sort*)

The *sort* command is used to order the data stored within a file in ascending or descending sequence at the time of display. Like the *cut* command, it also identifies fields and can sort on specified fields. When the *sort* command is used without any options, it sorts the file based on entire line. It reorders the lines based on ASCII sequence. The sorting is first applied on white spaces, followed by numerals, uppercase letters and finally lowercase letters.

Let us try to arrange the file address.txt in descending order of roll numbers (you must have observed that it is already arranged in ascending order). To display contents of file sorted in reverse order, execute the following command :

```
$sort -r address.txt
```

```
20018, Vedita, Harshal, Arolkar, 17, Jaldeep I, Ahmedabad, 380058  
20017, Nirav, Jose, Mackwan, Jose House, M G Road, Nadiad, 388011  
20016, Abdul, Shamsher, Khan, Khan Villa, M G Road, Nadiad, 388011  
20015, Harshit, Amit, Jain, 58, Jaldeep I, Ahmedabad, 380058  
20014, Dhrumil, Ajay, Patel, Yesh Enclave, Mota Bazar, Vidyanagar, 388120  
20013, Vaidehi, Sanjay, Shah, Sector-23, GH-6, Gandhinagar, 382023
```

Note :

The execution of *sort* command does not modify the actual file. The records in the actual file will remain in the same position. The sort order is applied only at the time of displaying the output.

The *sort* command is mostly used in conjunction with other commands. For example, we can use it with the *cut* command to order the output of the *cut* command. Execute the following command to see the output of both the *cut* and *sort* commands when combined.

```
$cut -d "," -f 2-4 address.txt | sort  
Abdul, Shamsher, Khan  
Dhrumil, Ajay, Patel  
Harshit, Amit, Jain  
Nirav, Jose, Mackwan  
Vaidehi, Sanjay, Shah  
Vedita, Harshal, Arolkar
```

The *cut* command in the above example extracts second, third and fourth column from the file address.txt. The extracted output is then given as input to the *sort* command. The *sort* command then sorts the contents and displays it on the screen.

Character Conversion (*tr*)

The command used as filters work with a line or column. The *tr* (translate) command allows us to work with individual characters within a line. It is used to translate (convert) strings or patterns from one set of characters to another.

Working with the address.txt file you must have observed that it contains “,” as delimiter. Assume that we do not want to show the delimiter at the time of the display, instead we would like to show a blank space. The *tr* command allows us to perform this operation. Execute the command shown below :

```
$cat address.txt | tr -s ',' ''
```

20013 Vaidehi Sanjay Shah Sector-23 GH-6 Gandhinagar 382023

20014 Dhrumil Ajay Patel Yesh Enclave Mota Bazar Vidyanagar 388120

20015 Harshit Amit Jain 58 Jaldeep I Ahmedabad 380058

20016 Abdul Shamsher Khan Khan Villa M G Road Nadiad 388011

20017 Nirav Jose Mackwan Jose House M G Road Nadiad 388011

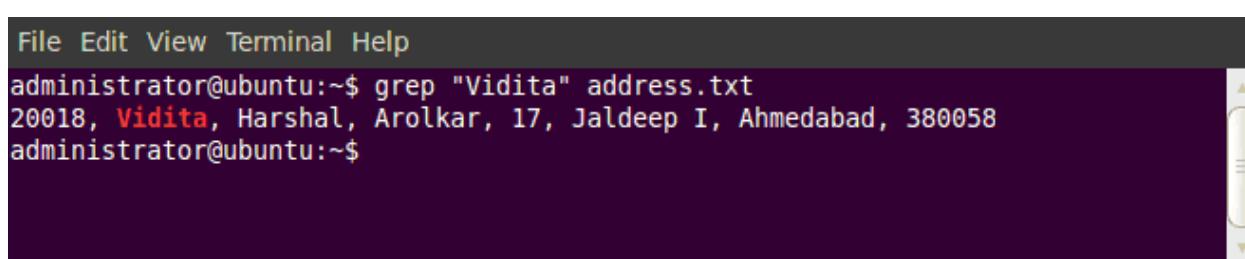
20018 Vedita Harshal Arolkar 17 Jaldeep I Ahmedabad 380058

The translation is only applicable at the display it will not permanently replace the actual delimiter. The *-s* option squeezes the additional space visible in the actual file. In case we need to save the translations visible we can redirect the output to a new file.

Pattern matching (*grep*)

The find operation is one of the most widely used operation in GUI applications. We must have used the CTRL + f keys to find a keyword within files. The *grep* command performs similar operation from the command line interface. The command is based on a fundamental idea of search globally for a regular expression and display lines where instances are found (g/re/p).

Let us make use of the *grep* command to find a name within the file address.txt and display its record. Figure 6.19 shows the working of the *grep* command.



A screenshot of a terminal window on an Ubuntu system. The window has a dark background with light-colored text. At the top, there is a menu bar with options: File, Edit, View, Terminal, and Help. Below the menu, the terminal prompt shows "administrator@ubuntu:~\$". The user then enters the command "grep "Vedita" address.txt". The terminal displays the result: "20018, Vedita, Harshal, Arolkar, 17, Jaldeep I, Ahmedabad, 380058". Finally, the prompt "administrator@ubuntu:~\$" appears again. On the right side of the terminal window, there is a vertical scroll bar.

Figure 6.19 : Working of grep command

Observe that the string that we are looking for is shown in red colour, also we have enclosed the keyword in double quote. It is not compulsory to enclose the keyword in double quotes hence the command **grep Vidita address.txt** will also give same output. The string used in the grep command is case sensitive hence the strings “Vidita” and “vidita” are different. We can use different options along with grep command that will help us refine our search in a better way. Table 6.8 lists the options and their usage.

Option	Usage
-c	Return only the number of matches, without quoting the text
-i	Ignore case while searching
-l	Return only file names containing a match, without quoting the text.
-n	Return the line number of matched text, as well as the text itself.
-v	Returns all the lines that do not match the text.
-w	Return lines which display only whole words
-o	Shows only the matched string

Table 6.8: Options of grep command

One very powerful feature of grep command is to use a regular expression as a keyword. Say for example we want details about the persons who are staying in society that has ‘Jal’ as its starting and ‘I’ as its end, then regular expression can be used. Let us execute the command shown here :

```
$grep "Jal.*I" address.txt
```

The output of this command is shown in figure 6.20. Here “Jal.*I” is a regular expression.

```
File Edit View Terminal Help
administrator@ubuntu:~$ grep "Jal.*I" address.txt
20015, Harshit, Amit, Jain, 58, Jaldeep I, Ahmedabad, 380058
20018, Vidita, Harshal, Arolkar, 17, Jaldeep I, Ahmedabad, 380058
administrator@ubuntu:~$
```

Figure 6.20 : Using regular expressions in grep command

A regular expression is normally followed by one of several repetition operators shown in table 6.9.

Repetition Operator	Meaning
?	The preceding item is optional and matched at most once.
*	The preceding item will be matched zero or more times.
+	The preceding item will be matched one or more times.
{n}	The preceding item is matched exactly n number of times.
{n,}	The preceding item is matched n or more number of times.
{,m}	The preceding item is matched at most m number of times.
{n,m}	The preceding item is matched at least n number of times, but not more than m number of times.

Table 6.9 : Repetition operator

Searching a file or Directory (*find*)

Many times we forget the location of the file or a directory that we have created. The *find* command helps us look for such forgotten objects. The *find* command looks for the search criteria (file or directory or both) that you have specified starting from the directory you specify within all its subdirectories. We can also search for the object based on its name, owner, group, type, permissions, date, and other criteria. Note that the *find* command when used without any other arguments displays the pathname of all the files and directories in the present directory and all its subdirectories.

Assume that we want to look for the location of file *introduction.txt* that we had created earlier, the command would be

```
$find -name introduction.txt
```

If the file exists then its path will be given as an output. Otherwise we may either get an error or a prompt will be visible if no such file exists. Note that we know the name of the file here, what if we only remembered first few characters of the file name. The wildcard characters can be used in such cases. The example shown below helps in finding all files that start with string “intro”.

```
$find -name intro*
./subject/economics/my1_dir/intro1
./subject/economics/my1_dir/intro
./subject/economics/introduction.txt
./subject/economics/introduction
./subject/economics/my_dir/intro1
./subject/economics/my_dir/intro
./subject/economics/intro1
```

The output of this command may vary on your screens depending on the number of files that you have which start with string “intro”. Also in Linux we may not assign a file extension, hence we may not be able to know whether intro1, intro and introduction in the above outputs are files or directories. We may refine the search if needed by using the *type* option as shown below :

\$find -name intro* -type f

Table 6.10 shows some example of find and its expected output description.

Command	Description
find / -type d	Search all directory and sub directory available on root only.
find . -mtime -1	Search objects modified within the past 24 hours.
find . -mtime +1	Search objects modified more than 48 hours ago.
find ./dir1 ./dir2 -name script.sh	Search directories “./dir1” and “./dir2” for a file “script.sh”.
find -size 0 -delete	Search for files of zero byte and delete them from the disk.
find -executable	Search for the executable file in current directory.
find /home -user jagat	Search for the object whose owner is jagat within the home directory and its sub directory.
find . -perm 664	Search for object that has read and write authorization for their owner, and group but which other users can only read.

Table 6.10: Example of find command

Running Commands as the Superuser

When you log in to your computer, the account you use is a regular user account. This account has a limited right. The security model of Ubuntu generally allows you to work as a normal user. Not providing administrative rights prevents any accidental changes or installation of malicious programs that may disturb functioning of the system. But many times the user may need to have administrative privileges. The administrative privileges are available to only a user known as superuser. To use the superuser account when using the terminal, we need to add *sudo* as a prefix to the commands that we want to execute. For example, execute the following command to install a new program called skype from command line.

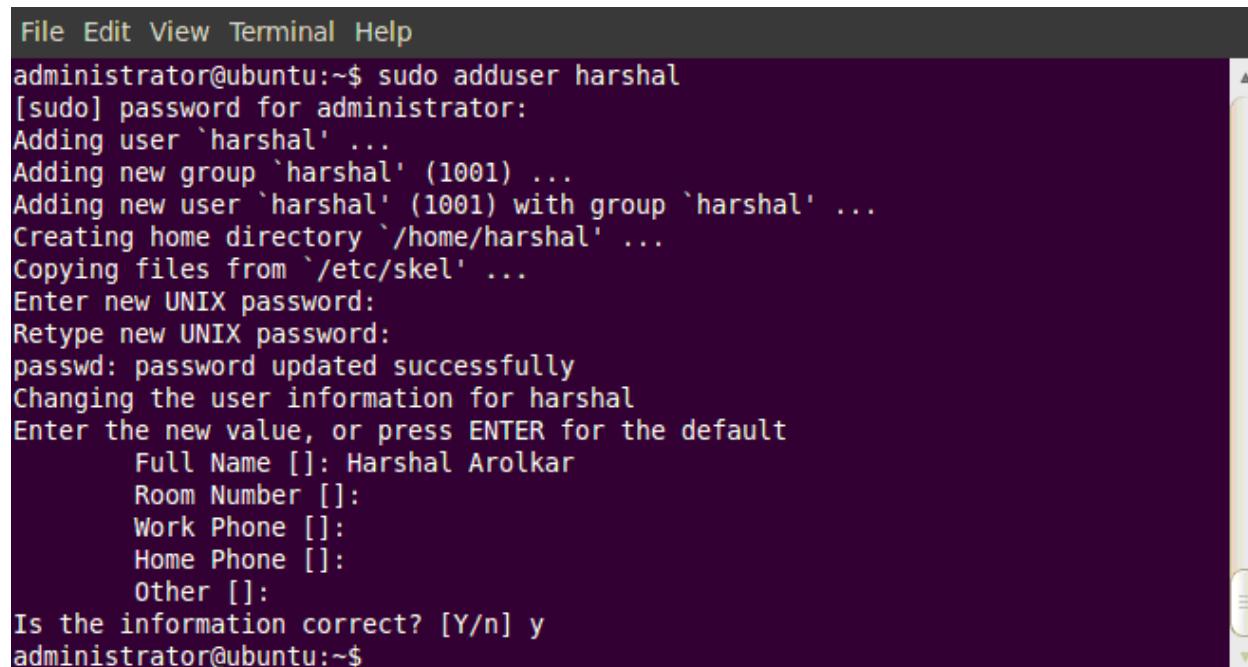
\$sudo apt-get install skype

When you execute the command it will ask for password, provide the password of superuser (generally it will be different from the normal user). This password is the password of the first user that you

added when you installed Ubuntu Linux on the computer. Once the user is authenticated as sudo by means of the terminal, the software will start installing. Once the installation is over we can start using the software.

The super user can also perform the operations of adding, deleting or updating a user, group or object in the system. Some of the commands listed below are used for such purposes.

adduser : The *adduser* command creates a new user on the system. Figure 6.20 shows the process of creating a user. The command when executed will ask for password and some additional details as shown in figure 6.21. Once all the details are provided a new user along with its home directory will be created in the system.



The screenshot shows a terminal window with a dark background and light-colored text. At the top, there's a menu bar with 'File', 'Edit', 'View', 'Terminal', and 'Help'. Below the menu, the terminal prompt is 'administrator@ubuntu:~\$'. The user runs the command 'sudo adduser harshal'. The terminal then prompts for a password, asking '[sudo] password for administrator:'. It continues to add the user 'harshal' to a group 'harshal' (ID 1001), creates a home directory '/home/harshal', and copies files from '/etc/skel'. It then asks for a new UNIX password, retype it, and updates the password successfully. It changes the user information for 'harshal' and prompts for optional details like Full Name, Room Number, Work Phone, Home Phone, and Other. Finally, it asks if the information is correct, with 'Y/n' as the response, and ends with the prompt 'administrator@ubuntu:~\$'.

```
File Edit View Terminal Help
administrator@ubuntu:~$ sudo adduser harshal
[sudo] password for administrator:
Adding user `harshal' ...
Adding new group `harshal' (1001) ...
Adding new user `harshal' (1001) with group `harshal' ...
Creating home directory `/home/harshal' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for harshal
Enter the new value, or press ENTER for the default
    Full Name []: Harshal Arolkar
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
administrator@ubuntu:~$
```

Figure 6.21 : Adding a user

passwd : The *passwd* command when executed as a super user do, allows us to change the password of any valid user of the system.

who : The *who* command when executed displays the list of all the users that are presently logged into the machine.

addgroup : The *addgroup* command adds a new group. The users are normally divided into groups so that they can be better controlled.

deluser : The *deluser* command is used to delete a user from the system. Note that we need to explicitly remove the user's files and home directory, by using the -remove -home option.

delgroup : The *delgroup* command deletes a group from the system. To perform this operation we must first make sure that no user is associated with the group that we are going to delete.

Summary

In this chapter we learned how to use the Ubuntu Linux command line interface. The CLI (Command Line Interface) when used allows us to perform all operations that we perform using the GUI in efficient and fast manner. We saw how to initiate the CLI using the Linux terminal. We also learned how to create, rename and delete a file or a directory, find out the directory that we are working in, change the directory if required. Later we saw how to create a copy of the file as well as directory. An access right is one of the ways to make sure that our data is not misused; we saw how to assign or change access rights. We saw how to increase the effectiveness of commands by joining multiple commands using the pipes. Further we looked at some features like counting the words or lines within the file, slicing the file horizontally and vertically, joining the file, searching for a file, directory or a string pattern within the files, arranging the display in ascending or descending order. Finally we saw how normal user can perform administrative tasks of installing new software, adding or deleting a user or group, checking who is using the system or change the password of some user.

EXERCISE

(12) Which of the following syntax is correct to assign a read permission on a user file?

- (a) chmod r filename
- (b) chmod u+r filename
- (c) chmod filename r
- (d) chmod filename u+r

(13) Which of the following refers to the minimum arguments of cp command?

- (a) One
- (b) Two
- (c) Three
- (d) None

(14) The mv command in Linux is used for which of the following purpose?

- (a) To rename a directory
- (b) To move a file
- (c) To copy a file
- (d) All of the above

(15) Which of the following command is used to view one page content on the screen at a time?

- (a) More
- (b) more
- (c) PAGE
- (d) page

Laboratory Exercises

1. Perform the following using Linux commands :

- (a) Print the calendar of December 2012.
- (b) Execute the command which displays login name, the name of your terminal and date and time since user logged in.
- (c) List all files starting with character ‘n’ or ‘N’.
- (d) Display the current working directory.
- (e) Prepare two files named class11_A.txt and class11_B.txt containing details of students of eleventh standard. The file should contain names of the students. Now merge these two files in a single file and name it class11_.txt. (Use cat command).
- (f) Hide the three files created in question ‘e’.
- (g) List only directories.
- (h) Get help on the use of the cat command.
- (i) List all files whose fourth character is ‘g’ and sixth character is digit.
- (j) Using terminal calculator perform the following operation :
 - (1) Calculate $2500/7$
 - (2) Convert decimal 50 to its binary equivalent
 - (3) Convert decimal 25 to its hexadecimal equivalent
 - (4) Find square root of 36
 - (5) Convert hexadecimal 25 to its decimal equivalent

2. Show the output of following Linux commands :

- (a) cat f1 >> f2
- (b) echo \$SHELL
- (c) mkdir d1 d2 d3
- (d) ls s*t
- (e) ls [a-f]*
- (f) ls -R
- (g) ls -xR
- (h) cp f1 f2
- (i) ls | sort
- (j) ls | tr -s " " | cut -d " " -f 5 | sort
- (k) ls -l | grep -c "address.txt"
- (l) grep "Harshit Jain" address.txt
- (m) chmod u-w address.txt
- (n) wc -l address.txt > totalstudents



Vim Editor and Basic Scripting



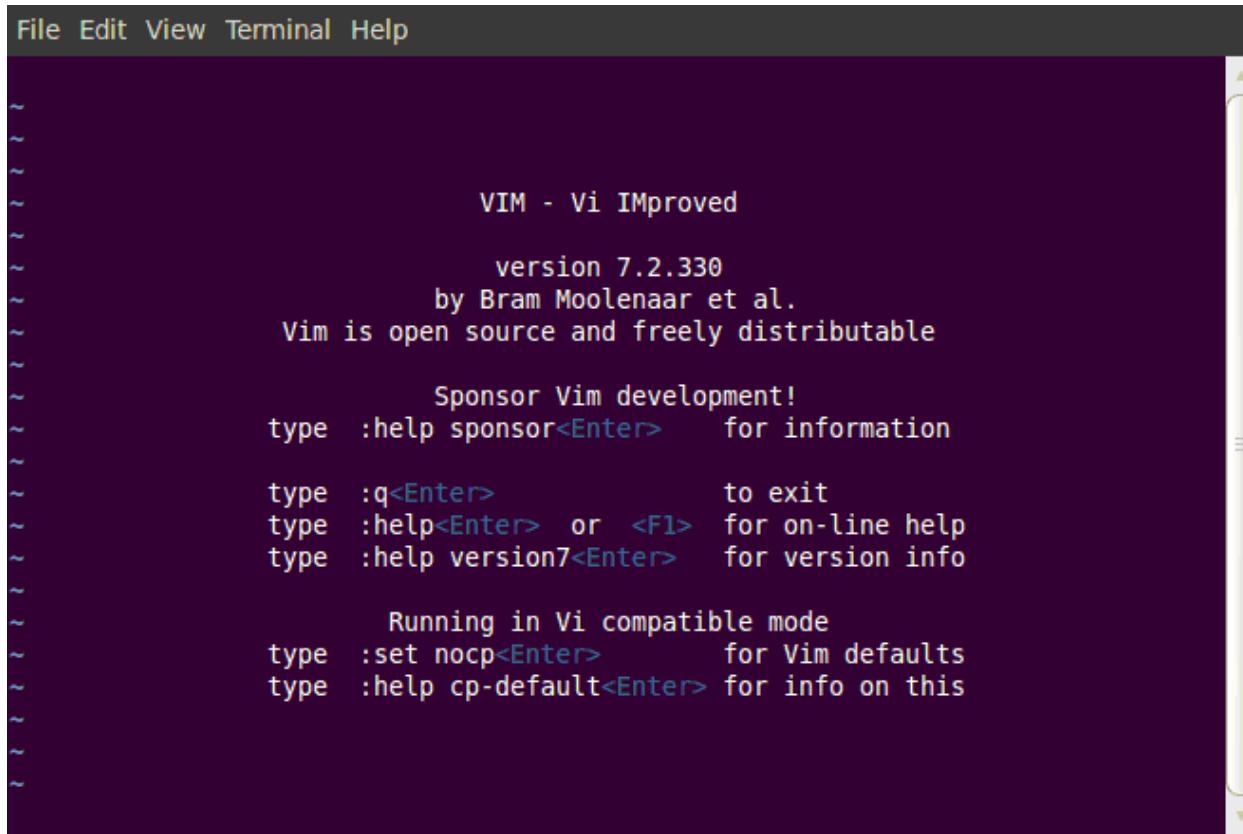
In the previous chapter, we have discussed commands that can be used to work with Ubuntu Linux. The commands were executed one at a time. Though we could execute multiple commands by using the pipes, the process would become tedious as number of commands increases. A better way of executing multiple commands at one go is to type the sequence of commands in a text file. Then give this file to Linux shell for execution. The Linux shell will execute all the commands available within the text file in the specified sequence. This text file is known as shell script. A shell script can be defined as series of commands written in a plain text file. The shell scripts are commonly used by the users to perform routine individual tasks and system administration. In this chapter we will look at an editor that assists us in writing the shell script along with some sample shell scripts.

Working with Vim Editor

We have learned how to create a file using the cat command. The cat command although it allows us to create a file is not a good option to use when creating a shell script. We need a good text editor to perform such operations. Text editors like nano, pico, vi or Vim, ed and others are generally best suited for creating text file. Gedit is a graphical editor available with GNOME desktop environment. Kwrite is a graphical editor available with KDE desktop environment. We will use the Vim editor which is a visual display editor to write the shell script. This editor is available with almost all Unix and Linux flavors.

The Vim (Vi Improved) is a text editor written by Bram Moolenaar and first released publicly in 1991. It is an enhanced version of the vi editor distributed with most UNIX systems. Vim is a highly configurable text editor built to enable efficient text editing. The Vim editor can be used from both a command line interface and as a standalone application in a graphical user interface.

To work with the Vim editor we will have to initiate it first. Open a new Terminal Window. We can open the Vim editor using two ways, first type **vi** at the prompt and press Enter key or type **vi** followed by a file name and press Enter key. Figure 7.1 shows the Vim editor interface when we don't specify a file name.



The screenshot shows the Vim editor interface with a dark background. At the top, there is a menu bar with options: File, Edit, View, Terminal, and Help. Below the menu, the screen displays the Vim startup message:

```
VIM - Vi IMproved
version 7.2.330
by Bram Moolenaar et al.
Vim is open source and freely distributable

Sponsor Vim development!
type :help sponsor<Enter> for information

type :q<Enter> to exit
type :help<Enter> or <F1> for on-line help
type :help version7<Enter> for version info

Running in Vi compatible mode
type :set nocp<Enter> for Vim defaults
type :help cp-default<Enter> for info on this
```

Figure 7.1 : Vim editor interface

It is a good option to start the Vim editor by specifying a file name. Type the command given below:

```
$vi first_vim_file
```

and press Enter key, the command when executed will open the editor as shown in figure 7.2.



The screenshot shows the Vim editor interface with a dark background. At the top, there is a menu bar with options: File, Edit, View, Terminal, and Help. Below the menu, the screen displays the message for a new file:

```
"first_vim_file" [New File]
```

Figure 7.2 : Creating file using Vim editor

Observe that the screen in figure 7.2 is filled with tildes (~) on the left side of the screen. The tilde (~) symbol indicates that the lines are yet to be used by the editor. Notice that the cursor appears in the top left corner of the screen and some text is visible on the last line. The last line is known as command line and it displays the name of the file along with the information about the total number of lines and columns within the file.

Vim Modes

The Vim editor functions in three different modes namely, a command mode, an insert mode and a last line mode.

The command mode

When we first start editing a file using the Vim editor, the editor will be opened in a command mode. We can issue many commands that allow us to insert, append, delete text, or search and navigate within our file. Note that when using command mode we can't insert text immediately. We first need to issue an insert (i), append (a), or open (o) command to insert the text in the file.

An extension to the command mode is a visual mode. Visual mode is a flexible and easy way to select a piece of text from the file. It is the only way to select a block of text that needs to be modified. Table 7.1 shows us the characters that assist us in the visual mode.

Command	Usage
v	Switch to the visual mode (allows us to manipulate characters)
V	Switch to the visual mode (allows us to manipulate lines)
CTRL + v	Switch to the block-visual mode (allows us to manipulate rectangular blocks of text)

Table 7.1 : Characters that are used in visual mode

The insert mode

When we issue an insert, append, or open command, we will be in the insert mode. The current text editors show us the current mode of operation. Once in an insert mode we can type text into our file or navigate within the file.

We can toggle between the command mode and the insert mode by pressing the ESC key. This operation will be performed often when using the Vim editor. Table 7.2 shows us the characters that assist us in the insert mode.

Command	Usage
a	To insert text after the current cursor position.
i	To insert text before the current cursor position.
A	To append text at the end of the current line.
I	To insert text from the beginning of a line.
O	To insert in a new line above the current cursor position.
o	To insert in a new line below the current cursor position.

Table 7.2: Characters that are used in insert mode

The last line mode

The last line mode normally is used to perform operations like quitting the Vim session or saving a file. To go to the last line mode we first need to be in the command mode. From command mode we can go to last line mode by pressing the colon (:) key. After pressing this key, we will see a colon character at the beginning of the last line of our editor window with a cursor blinking near it. This indicates that the editor is ready for accepting a “last line command”.

It is possible to toggle back to the command mode from the last line mode by pressing the ESC key twice or by pressing the [Backspace] key until the initial “:” character is gone along with all the characters that we had typed or by simply pressing the ENTER key.

Creating a file in Vim

Let us now learn how to create a simple text file using the Vim editor. Execute the command given below to open the editor interface.

\$vi about_Gandhiji

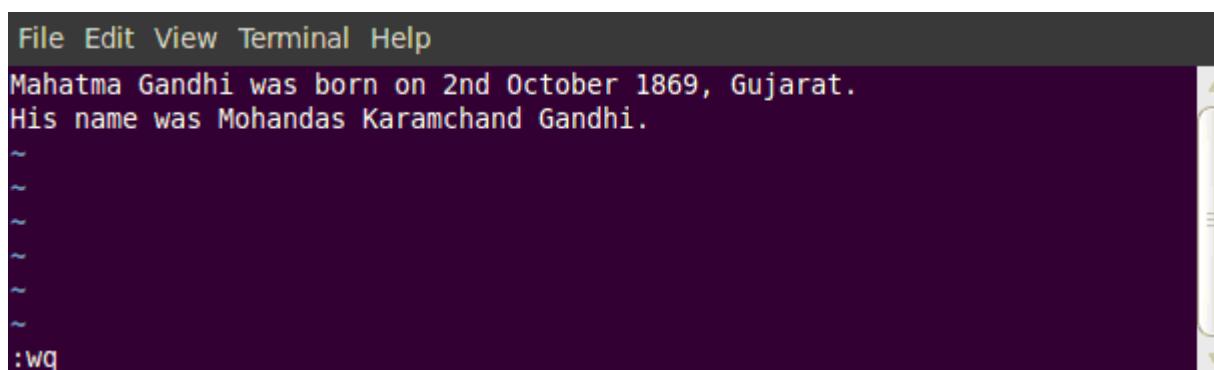
To enter text within the file the editor needs to be in the insert mode. By default, the editor will start in the command mode. As seen in table 7.2 there are several commands that put the editor into the insert mode. The most commonly used commands to get into insert mode are ‘a’ and ‘i’.

Press ‘i’ and the editor will now be in the insert mode. Now type the contents as given in the box below.

```
Mahatma Gandhi was born on 2nd October 1869 in Porbandar, Gujarat.  
His name was Mohandas Karamchand Gandhi.
```

Saving the file

Once we have written the contents shown we need to save this file. To save the file we need to switch to the last line mode from the insert mode. Press the ESC key and type colon (:), you will notice that colon is displayed in the bottom of the screen. Type **wq** (write and quit) as shown in figure 7.3 and press the Enter key. The Vim editor will now be closed and we will be able to see the shell prompt. To see the contents of the file we can use the cat command.



```
File Edit View Terminal Help  
Mahatma Gandhi was born on 2nd October 1869, Gujarat.  
His name was Mohandas Karamchand Gandhi.  
~  
~  
~  
~  
~  
~  
:wq
```

Figure 7.3: Saving a file and quitting the editor

There are several other commands available to save a file depending on the current status and usage. Table 7.3 shows commands that can be used in the last line mode along with their usage.

Command	Usage
.w	To save file and remain in editing mode
:wq	To save file and quit editing mode
x	To save file and quit editing mode (same as above)
:q	To quit editing mode when no changes are made
:q!	To quit editing mode without saving changes made in the file
:saveas FILENAME	To save existing file with new name and continue editing it under the new file name.

Table 7.3 : The last line mode commands to save the file

Note that if we open the Vim editor without typing file name initially the text will be directly stored in the system buffer (main memory). To transfer the contents from the buffer to a hard disk we need to type the file name along with the *wq* command.

Moving around in the document

You must have observed when creating the file about Gandhiji the arrow keys (by any chance if you used them) were not working as per our expectations. Normally the arrow keys are used to move the cursor in up, down, left and right directions. In insert mode we cannot do anything except for typing the text. When using the Vim editor we need to use some special keystrokes to move within the document after going to the command mode. Once in command mode we can also use the arrow keys to move within the document. Table 7.4 lists the keystrokes that are used to navigate within the documents.

Command	Usage
h	Moves cursor left
l	Moves cursor right
j	Moves cursor down
k	Moves cursor up
Spacebar	Move cursor right one space

-/+ Keys	Move cursor down/up in first column
CTRL + d	Scroll down one half of a page
CTRL + u	Scroll up one half of a page
CTRL + f	Scroll forward one page
CTRL + b	Scroll back one page
M	Move the cursor middle of the page
H	Move the cursor to top of the page
L	Move the cursor to bottom of page
\$	Move the cursor to end of line
)	Move the cursor to beginning of next sentence
(Move the cursor to beginning of current sentence
G	Move the cursor to end of file
W	Move the cursor one word at a time
Nw	Move the cursor ahead by N number of words
B	Move the cursor back a word at a time.
b	Move the cursor back a word at a time.
Nb	Move the cursor back by N number of words
e	Move the cursor to end of word
gg	Move to first line of file
0	Move to the beginning of the line.

Table 7.4 : Keys to navigate in file

Try and work on these keystrokes to become familiar with them.

Editing the Document

Editing the document is one of the most common operations that a user would perform once the document is created. It is possible to insert or delete any data at a specific position as per our

needs. We can also replace contents or change the case of individual characters if required. A user needs to toggle between the command and the insert mode when we edit a document. The characters in table 7.2 showed us how to initiate different insert options. Let us try to add the contents given in the box below to the file about_Gandhiji.

His mother, Putlibai, was a very religious lady and used to tell him stories from the scriptures and mythology.

Little Gandhi grew up to be an honest and a decent student. At the age of 13 he was married to Kasturba.

To edit the document we need to again open it using the vi command, hence execute the command **vi about_Gandhiji** again. You will observe that the blinking cursor is visible on the first character at top left. In normal cases we would have used the ‘I’ option to enter the insert mode, but we need to append the contents at the end.

Type G and you will see that the cursor gets positioned at last line. The position of the cursor will depend upon how the file was initially saved. Typing G may place the cursor in a new line after the last line (case when Enter key was pressed before saving the file) or the cursor will be placed on the first character of the last line (case when Enter key was not pressed before saving the file).

If the cursor is placed at new line we press ESC and then ‘I’ and start typing the contents. In case we are at the first character of the last line, then press ‘o’. This action will take you to a new line. Now start typing the contents, in case of any errors in typing we may use the Backspace key and go back one cursor position at a time to correct the error. Once the editing is over press ESC key and type :wq to go the last line mode, save the file and end the Vim session. Figure 7.4 shows the look of the editor after the new contents have been added.

```
File Edit View Terminal Help
Mahatma Gandhi was born on 2nd October 1869, Gujarat.
His name was Mohandas Karamchand Gandhi.
His mother, Putlibai was a very religious lady and used to tell him stories from the scriptures and mythology.
Little Gandhi grew up to be an honest and a decent student. At the age of 13 he was married to Kasturba.
~
~
:wq
```

Figure 7.4 : Appending data in existing file

We may use the commands shown in table 7.5 to perform various editing operations on any document.

Command	Usage
u	Undo last change
U	Undo all changes to entire line
dd	Delete single line
Ndd	Delete N number of lines
D	Delete contents of line after cursor
C	Delete contents of line after cursor and insert new text. Press ESC key to end the insertion
dw	Delete one word
Ndw	Delete N number of words
cw	Change word
x	Delete the character under the cursor.
X	Delete the character before the cursor (Backspace).
r	Replace single character
R	Overwrite characters from cursor onward
s	Substitute one character under cursor and continue to insert
S	Substitute entire line and begin to insert at beginning of line
~	Change case of individual character
.	Repeat last command action.

Table 7.5 : Commands to perform editing

In addition to the commands given in table 7.5 the Vim editor also allows us to copy text from our file into temporary buffers and vice-versa. Each buffer acts like temporary memory, more commonly known as “clipboard”. Table 7.6 lists some of the commands that are used for capturing and pasting data.

Command	Usage
yy	Copy single line (defined by current cursor position) into the buffer
Nyy	Copy N lines from current cursor position into the buffer
p	Place (paste) contents of buffer after current line defined by current cursor position.

Table 7.6 : Commands to capture and paste

Searching and replacing text

Searching for content and replacing it is another common operation performed by users. The Vim editor allows us to use special commands to search text or a regular expression within the file. We can also substitute a word in place of another using command. Table 7.7 lists various commands that are useful for performing search or replace operation within a file.

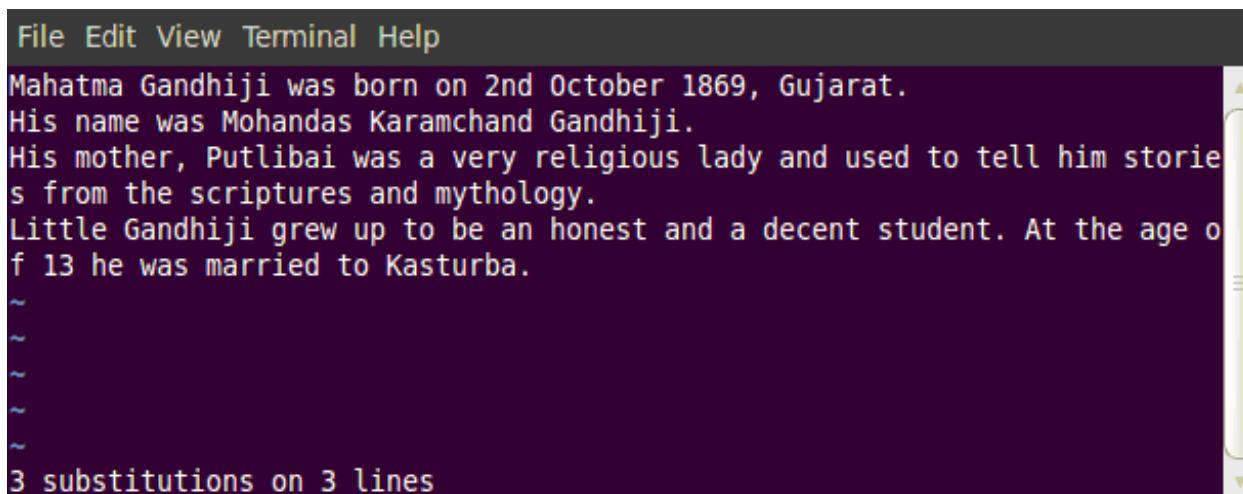
Command	Usage
/	Search for text in a forward direction
?	Search for text in a backwards direction
n	Search again in the same direction
SHIFT + n	Search again in the opposite direction
f	Press <i>f</i> and type the character to be searched. The cursor will move to that character on the current line.
SHIFT + f	Similar to <i>f</i> but searches in backward direction
t	Similar to <i>f</i> except that it moves the cursor one character before the specified character.
SHIFT + t	Similar to <i>t</i> but searches in backward direction
:s/old_string/new_string	Substitutes <i>new_string</i> for the first occurrence of the <i>old_string</i> in the current line
:s/old_string/new_string/g	Substitutes <i>new_string</i> for all the occurrences of the <i>old_string</i> in the current line
:%s/old_string/new_string/g	Substitutes <i>new_string</i> for all the occurrences of the <i>old_string</i> in the whole file
:%s/old_string/new_string/gc	Substitutes <i>new_string</i> for all the occurrences of the <i>old_string</i> in the file, but asks for confirmation before substituting the <i>new_string</i>

Table 7.7 : Commands to perform search and replace operation

Let us try to use some of these commands in the file *about_Gandhiji*. Assume that we wanted to replace all the occurrence of the word “Gandhi” with word “Gandhiji”. To perform this operation we need to first open the file using the command *vi about_Gandhiji*, then go to the last line mode by pressing the *ESC* key and execute the command given below:

```
:%s/Gandhi/Gandhiji/g
```

Here %s indicates we are trying to replace a string. The term “Gandhi” refers to the old string that is to be replaced while the term “Gandhiji” refers to the new string. The option “g” indicates that we have to substitute all the occurrences of the term “Gandhi” with the term “Gandhiji” in the whole file. The output of this command is shown in figure 7.5. Observe that it also shows how many occurrences have been replaced. We need to save the file if we need this change to be reflected in it. If we quit without saving then the changes will not be reflected in the file.



The screenshot shows a Vim editor window with a dark background. The menu bar at the top includes 'File', 'Edit', 'View', 'Terminal', and 'Help'. The main text area contains the following text:

```
Mahatma Gandhiji was born on 2nd October 1869, Gujarat.  
His name was Mohandas Karamchand Gandhiji.  
His mother, Putlibai was a very religious lady and used to tell him stories from the scriptures and mythology.  
Little Gandhiji grew up to be an honest and a decent student. At the age of 13 he was married to Kasturba.  
~  
~  
~  
~  
~  
3 substitutions on 3 lines
```

A vertical scroll bar is visible on the right side of the editor window.

Figure 7.5 : Search and replace operation

Executing Linux commands through Vim

It is also possible to execute the Linux commands from within the Vim editor. To execute any Linux command we need to type the exclamation (!) symbol before the command.

For example if we want to see the current working directory then perform the following steps :

- Open the Vim editor by typing vi on the command prompt.
- Go to the last line mode by pressing ‘ESC :’
- Type !pwd
- Press Enter key.

We will be able to see the current working directory. Figure 7.6 shows the operation and its output.

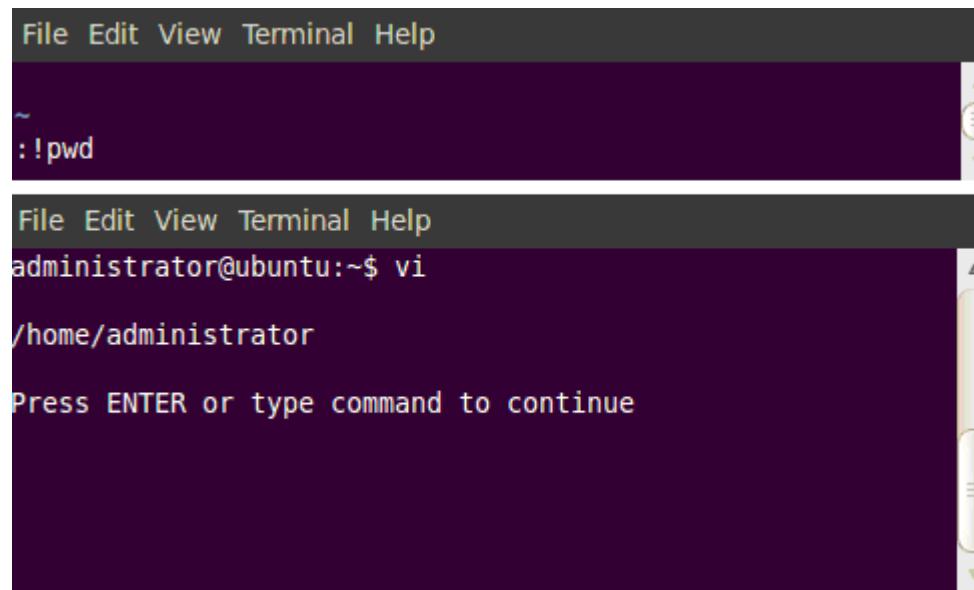


Figure 7.6 : Linux command in Vim editor

Similarly if we want to add the current date in a new line from the current cursor position within a file we may execute the following steps:

- Open the file using Vim editor.
- Go to the last line mode by pressing ‘ESC :’
- Type r !date
- Press Enter key.

The r option allows us to insert data in the file or buffer.

Shell Script

We saw how a command can be executed from a command prompt as well as using the Vim editor. Both these mechanisms allow us to execute commands one at a time. A shell script allows us to execute more than one command at one go in a better way. Thus instead of spending time in typing the commands on the prompt every time a task needs to be performed, we can create a shell script and execute the given set of commands by typing a single line command.

Shell script can be defined as “Set of commands written in plain text file that performs a designated task in a controlled order.” Shell scripts can be designed to be interactive; such a script may accept input from the user and perform different tasks based on the input provided.

Creating and Executing a Shell Script

We can create the shell script using any text editor. As we have learned to work with the Vim editor we will create scripts in it. Let us create a small shell script that welcomes a user. Type the script shown in the box below in Vim editor and save it with the name *script1.sh*. The extension

“sh” is basically used to specify that the file is a shell script. Note that it is not mandatory to create a file with an extension “sh”, a file without extension can also be used as a shell script. But it is always a good practice to give extension as it will help us differentiate between normal files and shell script files.

```
#Script 1: Script to welcome the user who has logged into the system  
clear  
echo Hello  
who am i  
echo Welcome to Ubuntu Linux
```

Observe the first line in the script; it begins with symbol ‘#’. Any line preceded by the ‘#’ symbol is considered as a comment. The comments when part of the script are not executed; they are messages that help user understand the usage or meaning of the script. The second line is a command that clears the screen contents before giving the output of the script. The third line displays a message “Hello”, the fourth line executes a command “*who am i*” that gives us the name and some additional details of the user currently logged into the system. The last line again displays a message “Welcome to Ubuntu Linux”.

To execute the script we need to use *sh* or *bash* command. If the script is stored in current directory then type the command mentioned below:

\$bash script1.sh or \$sh script1.sh

Sometimes we might come across issues related to file privileges. For a script to be executed it needs to have execute permission explicitly set. By any chance if such a problem occurs we will have to use the *chmod* command to set the desired privileges. For Example issuing the command *chmod +x script1.sh* will make the file executable. If everything goes fine we will get the output similar to the one shown in figure 7.7.

```
File Edit View Terminal Help  
Hello  
administrator pts/0 2013-02-18 10:06 (:0.0)  
Welcome to Ubuntu Linux  
administrator@ubuntu:~$
```

Figure 7.7 : Output of Script 1

Observe that in figure 7.7 we are getting some additional contents along with the user name. Let us use our knowledge of filters and try to remove the additional contents. The modified script is shown below:

```
#Script 2: Modified script to welcome the user who has logged into the system
clear
echo Hello
echo "`who am i | cut -d " " -f 1`"
echo Welcome to Ubuntu Linux
```

Observe that we have used the filter *cut* along with the command *who am i*. We further have joined the two commands using the pipe. To make sure that the contents within the double quotes after the echo command are not treated as message we enclose them in back quotes (` `). The back quotes are printed on the key with ~ sign on the keyboard. Type the modified script using Vim editor and save it as *script2.sh*. Execute this script and you will observe that we are only able to see the contents that we want. Figure 7.8 shows the output of the modified script.

```
File Edit View Terminal Help
Hello
administrator
Welcome to Ubuntu Linux
administrator@ubuntu:~$
```

Figure 7.8 : Output of Script 2

Let us further modify the script to display current date and time. The script is given in the box below .

```
#Script 3: Script to welcome the user and display login date and time
clear
echo Hello
echo "`who am i | cut -d " " -f 1`"
echo Welcome to Ubuntu Linux
echo The current date and time is
date
```

Create a file named *script3.sh* and type the contents of the script 3 in it. Execute it and observe the output. Once we are comfortable using the shell scripts we will find them very helpful in performing repetitive tasks.

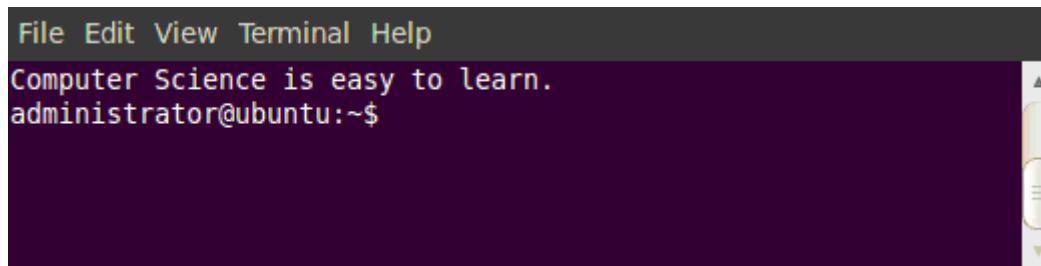
Shell Script Variables

The process of shell scripting is almost similar to the process of writing programs in a higher level language. One of the most common features of higher level programming is provision of variables.

As the name indicates variables are entities wherein we can store or edit a value. The value stored in the variable can also be reused or changed as per users need. Shell script variables like any other programming language variables are integral part of shell scripting. A variable when used in a shell script allows us to assign a value to it or accept its value from the user. We can also display the value assigned to this on the screen using echo command. Let us write a small script that shows the use of variable. The code of the script is given in the box below:

```
# Script 4: Shell script to show use of variables  
clear  
subject="Computer Science"  
echo $subject is easy to learn.
```

Type the contents of the script 4 in a file named *script4.sh*. Let us try to understand the script just saved. Similar to all the other scripts the first line indicates a comment. The second line clears the screen of any previous contents. In the third statement we have defined a variable called *subject* and assigned it value “Computer Science” using a simple assignment operator. As the string contains white space in between two words we need to enclose it within double quotes. The fourth line displays the message on the screen. The ‘\$’ symbol preceding the variable name *subject* instructs the shell to extract the value stored or assigned in the variable. Figure 7.9 displays the output of the script when it is executed.



```
File Edit View Terminal Help  
Computer Science is easy to learn.  
administrator@ubuntu:~$
```

Figure 7.9 : Output of Script 4

A user needs to take care that there should not be any space on either side of the equal to (=) symbol at the time of assigning a value. If due to some reasons a space occurs at either side, the shell will interpret the string after the space as a command. This may give unexpected outputs. The statement *subject=“Computer Science”* first creates a variable named *subject* and then assigns it a value “Computer Science”. If we reuse this variable again, the old value stored in it will be overwritten. Let us try to understand the last statement by writing a simple script given in the box:

```
# Script 5: Shell script to show use of variables  
clear  
subject="Computer Science"  
echo $subject is easy to learn.  
subject="Economics"  
echo $subject is easy to learn.
```

Save the script as script5.sh and observe its output after executing it.

We need to follow the rules mentioned below when defining a variable in a shell script.

- A variable name can consist of alphabets, digits or an underscore (_).
- No special character other than underscore allowed as part of variable name.
- The first character of a variable name must either be an alphabet or an underscore.

Note :

If the shell is unable to understand a word as a variable it will interpret it as a Linux command.

User Interaction and Shell Script

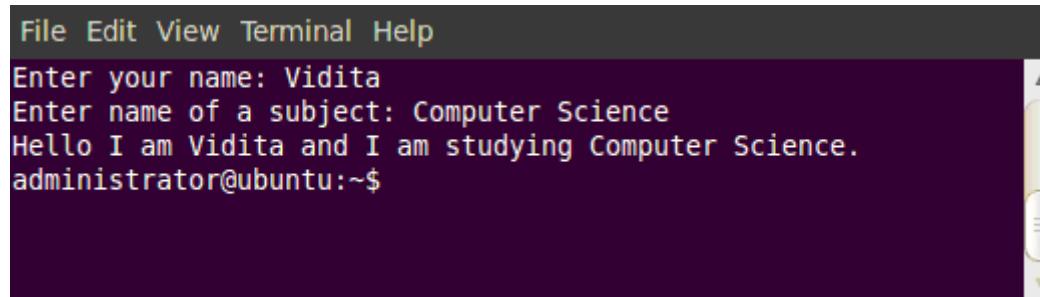
You must have observed in the previous example that we have assigned a value to the variable within the script itself. A variable when used in such a manner does not have much significance. A variable is generally used so that it can be further used in some operations with change in its value if needed. This property can be achieved only if we are able to accept the value of the variable from the user. It is possible to assign value to variables defined in the shell script using the *read* command. The *read* command expects the user to key in the data on the standard input device, it then takes all the contents that we type and stores it in the variable name supplied to it as an argument. Let us rewrite a script 5 to accept the subject names from the user. Save the code in the box with a file name *script6.sh*.

```
# Script 6: Shell script to accept value of variable from user
clear
echo -n "Enter your name: "
read name
echo -n "Enter name of a subject: "
read subject
echo Hello I am $name and I am studying $subject.
```

When we execute the script, the first executable statement will first clear the screen contents. Then it will display a message “Enter your name:”, the next statement waits for the user to enter its name. Pressing of Enter key indicates end of entry, so be careful that it is pressed only after the name has been typed. In case we press Enter key without typing anything the script will assign NULL value to the variable and move to next command. The third and fourth

statement also does the same task of displaying message and waiting for the user to key in a value for subject. The last statement displays the values of both the variables along with an appropriate message.

Observe that we have used -n option along with the echo command. This option instructs the echo command not to print a new line after the message is displayed. The echo command by default inserts a new line after displaying the message passed to it as an argument. Figure 7.10 shows output of the script.



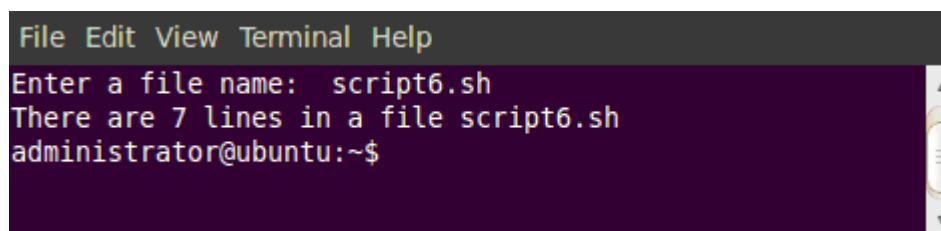
The screenshot shows a terminal window with a dark background and light-colored text. At the top, there's a menu bar with 'File', 'Edit', 'View', 'Terminal', and 'Help'. Below the menu, the terminal prompt is 'administrator@ubuntu:~\$'. The user has entered three commands: 'Enter your name: Vidita', 'Enter name of a subject: Computer Science', and 'Hello I am Vidita and I am studying Computer Science.'. The output of the third command is followed by a new line character, which is visible as a small white space in the terminal window.

Figure 7.10 : Output of Script 6

As we are accepting the values of both the variables from the user, the output will change according to what user enters every time we execute the script. Let us write one more script that accepts a file name from the user and display the total number of lines in that file. The code of the script is shown herewith save it as *script7.sh*.

```
#Script 7: Shell script to display total number of lines in a file
clear
echo -n "Enter a file name: "
read fname
echo "There are `cat $fname | wc -l` lines in a file $fname"
```

Figure 7.11 shows the output of the script when it is executed.



The screenshot shows a terminal window with a dark background and light-colored text. At the top, there's a menu bar with 'File', 'Edit', 'View', 'Terminal', and 'Help'. Below the menu, the terminal prompt is 'administrator@ubuntu:~\$'. The user has entered 'Enter a file name: script6.sh' and the script has responded with 'There are 7 lines in a file script6.sh'. The output of the script is followed by a new line character, which is visible as a small white space in the terminal window.

Figure 7.11 : Output of Script 7

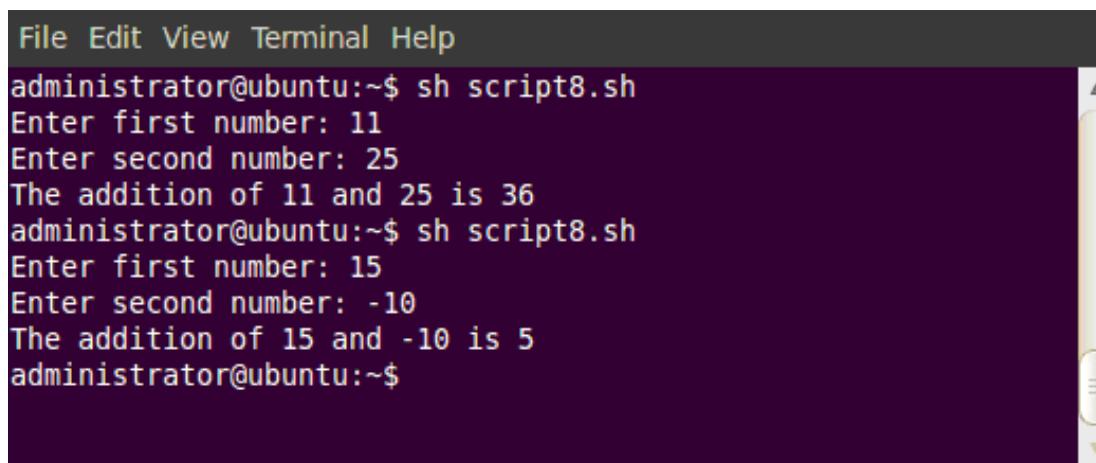
Shell Arithmetic

In the previous section we saw how to define a variable, assign value to it and how to retrieve the stored value. The value assigned so far were all strings (set of alphabet, digit or special characters). We can also assign only numeric values to the variable and perform operation with them. Let us write a shell script that accept two numbers and perform addition of these numbers. The code of the script is shown in the box below:

```
# Script 8: Script to add two numbers
echo -n "Enter first number: "
read num1
echo -n "Enter second number: "
read num2
sum=`expr $num1 + $num2`
echo "The addition of $num1 and $num2 is $sum"
```

Type the script and save it as *script8.sh*. In this script the term *expr* means expression, the contents written after *expr* are assumed to be operands and operators of an expression. Note that there should be one space between operator (+) and operands (\$num1, \$num2). Additionally, there should be no space before or after the assignment operator (=).

Figure 7.12 shows the different output of the script when it is executed twice. We are able to see the different outputs in one screen as we have not used the clear command in this script.



```
File Edit View Terminal Help
administrator@ubuntu:~$ sh script8.sh
Enter first number: 11
Enter second number: 25
The addition of 11 and 25 is 36
administrator@ubuntu:~$ sh script8.sh
Enter first number: 15
Enter second number: -10
The addition of 15 and -10 is 5
administrator@ubuntu:~$
```

Figure 7.12 : Output of Script 8

We can also perform subtraction, multiplication, division and modular division by using the -, *, / and % operators respectively. The expressions are evaluated as per the general norms of mathematics.

In case of tie between operators of same priority, preference is given to the operator which occurs first. To force one operation to be performed earlier than the other, we can enclose the operation in parenthesis.

For example, in the expression `$num1 * (\$num2 + $num3) / $num4`, the operation `$num2 + $num3` will be evaluated first as it is enclosed within parentheses. Observe that we have preceded the '*' symbol as well as the left and right parentheses by a back slash character (\).

Note :

We need to prefix the multiplication (*) symbol with backslash (\) character when finding product of two numbers. Otherwise the shell will treat the (*) symbol as a wildcard character.

Let us create one more script that will accept a birth year from the user and display users current age in years. The code of the script is shown in the box below.

```
# Script 9: Script to calculate age of user in years
echo -n "Enter year of your birth: "
read byear
cyear=`date | tr -s ' ' | cut -d " " -f 6`
age=`expr $cyear - $byear`
echo "You are $age years old as of today."
```

Observe that in script 9 to make sure that all the multiple spaces in output of date command are squeezed to single space we have used the *tr* command with *-s* option. As we want only the year value which appears in the 6th column of the output when the date command is executed we have used the *cut* filter. Figure 7.13 shows the different output of the script when it is executed.

```
File Edit View Terminal Help
administrator@ubuntu:~$ sh script9.sh
Enter year of your birth: 1974
You are 39 years old as of today.
administrator@ubuntu:~$ sh script9.sh
Enter year of your birth: 2001
You are 12 years old as of today.
administrator@ubuntu:~$
```

Figure 7.13 : Output of Script 9

Use of Shell Scripts

The shell script is a very powerful tool of Linux. It has almost all the capabilities of any higher level programming language. Once familiar with it we can perform and automate many tasks easily. Generally the repetitive task should be done using the shell scripts. Some uses of shell scripts are mentioned below:

- Create a new command using multiple set of commands.
- For automating many aspects of computer maintenance, for example create 1000 user accounts; delete all size 0 files, installation of new software etc.
- Data backup

As such there are no limitations on its usage; a user may use it for any purpose that he wants.

Summary

In this chapter we learned how to use the Vim test editor provided with Ubuntu Linux. The text editor is very powerful; it allows creating, updating and deleting contents of a file. Further we can search for required contents within a file. We also learned how to create a simple text file as well as a shell script using this editor. The shell script is a text file that contains sequence of commands that can be executed by simply using the shell scripts file name. Finally we saw how a shell script can be made equivalent to a high level program by making use of a variable and then using it in an expression.

EXERCISE

1. Explain different modes available in Vim editor.
2. List and explain the working of different save options of Vim editor.
3. Explain the difference between using dd and 2dd command.
4. What is a shell script?
5. List at least three uses of shell script.
6. **Choose the most appropriate option from those given below :**
 - (1) In how many modes Vim editor works?
 - (a) One
 - (b) Two
 - (c) Three
 - (d) Four
 - (2) Which of the following statement is true for Gedit?
 - (a) It is a Command line editor.
 - (b) It is a Graphical editor.
 - (c) It is not an editor.
 - (d) It is available with KDE Desktop environment.

- (3) :wq in Vim editor is used for which of the following activities?
- (a) To save file and remain in editing mode
 - (b) To save file and quit editing mode
 - (c) To quite editing mode without saving changes made in the file
 - (d) All of the above
- (4) Which of following keys is not used to go into insert mode of the Vim editor?
- (a) o
 - (b) i
 - (c) a
 - (d) cw
- (5) Which of the following keys are used to delete a line?
- (a) ce
 - (b) ge
 - (c) dd
 - (d) d\$
- (6) Which of the following statements is used to search for phrase in the file?
- (a) :set is
 - (b) :help cmd
 - (c) :!cmd
 - (d) /phrase<ENTER>
- (7) Which of the following syntax is used to substitute all occurrences of phrase1 with phrase2 in the entire file without asking for user confirmation?
- (a) :%s/phrase1/phrase2/g
 - (b) :%s/phrase1/phrase2/gc
 - (c) :s/phrase1/phrase2/g
 - (d) :s/phrase1/phrase2/gc
- (8) Which of the following character is used for commenting a line in a shell script?
- (a) *
 - (b) %
 - (c) \$
 - (d) #
- (9) Which of the following symbol instructs a shell script to extract the value of a variable?
- (a) *
 - (b) %
 - (c) \$
 - (d) #

Laboratory Exercises

Write a script to perform following operations :

- (a) To display the date and time in the given format:
“Today is February 15, 2013 and current time is 12:10:23”

- (b) To display the login details of current user in the following format:
- Name of the user:
- Login date:
- Login time:
- (c) To display the date, time, username and current directory.
- (d) To accept a string and a filename from the user. Search all occurrences of the string inside a given file.
- (e) To accept a file name from the user and count number of lines in it.
- (f) To accept two file names from the user and creates a new file containing the contents of both the files provided as input.
- To accept two file names from the user and compare them.



8 Advanced Scripting



In chapter 7 we learned how to use the Vim editor and also saw how to write the basic shell scripts. We mentioned that the shell scripts have features similar to a higher level language. Are we then learning a new language? No, we are not learning any new language at all. We are learning one of the best feature that an open source OS provides. The shell scripts are used for routine system administration tasks. They are the best tools an administrator can get to easily monitor and control his systems even if he is at remote location. The shell scripts designed so far were sequential in nature; the commands were executed in the same order in which they appeared in the script. While performing administrative tasks, we may need to perform execution of some statements repeatedly. We may also need to skip execution of some statements based on predefined conditions. In this chapter we will see some scripts related to system administration and discuss how to use decision statements and looping constructs in shell script.

Finding Process Id

In Linux all programs (executables stored on hard disk) are executed as processes (a program loaded into memory and running). Each process when started has a unique number associated with it known as process id (PID). We can perform operations like view or stop a process. To see the processes associated with the current shell we can issue the *ps* command without any parameters. We can view the process of all the users by using the *ps -ef* command. Figure 8.1 shows the processes running on our system.

```
File Edit View Terminal Help
administrator@ubuntu:~$ ps -ef
UID      PID  PPID  C STIME TTY          TIME CMD
root      1      0  0 15:14 ?        00:00:00 /sbin/init
root      2      0  0 15:14 ?        00:00:00 [kthreadd]
root      3      2  0 15:14 ?        00:00:00 [migration/0]
root      4      2  0 15:14 ?        00:00:00 [ksoftirqd/0]
root      5      2  0 15:14 ?        00:00:00 [watchdog/0]
root      6      2  0 15:14 ?        00:00:00 [migration/1]
root      7      2  0 15:14 ?        00:00:00 [ksoftirqd/1]
root      8      2  0 15:14 ?        00:00:00 [watchdog/1]
root      9      2  0 15:14 ?        00:00:00 [events/0]
root     10      2  0 15:14 ?        00:00:00 [events/1]
```

Figure 8.1 : List of processes

Table 8.1 gives the meaning of some of the columns listed in figure 8.1.

Column Name	Description
UID	Name or number of the user who owns the process.
PID	A unique numeric process identifier assigned to each process.
PPID	Identifies the parent process id, the process that created the current process.
STIME	The start time for the current process.
TTY	Identifies the terminal that controls the current process.
TIME	Identifies the amount of CPU time accumulated by the current process.
CMD	Identifies the command used to invoke the process.

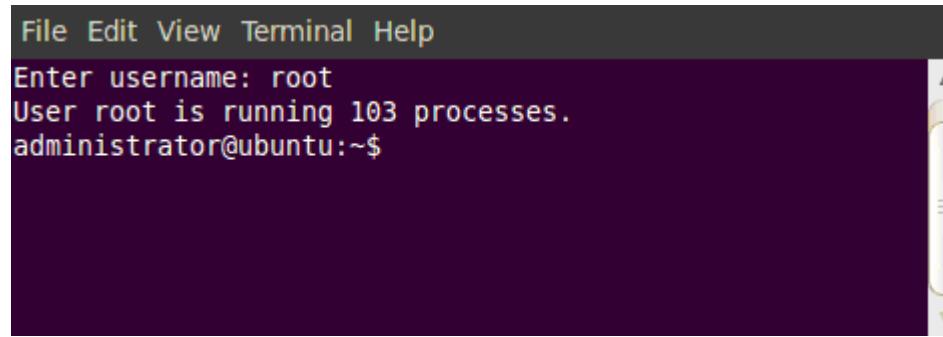
Table 8.1 : Explanation of columns displayed in ps –ef command

Many times an administrator needs to find how many processes a particular user is executing. Let us write a script that helps administrator find number of process run by a particular user.

```
#Script 10: Script to find out how many processes a user is running.

clear
echo -n "Enter username: "
read usrname
cnt=`ps -ef | cut -d " " -f 1 | grep -o $usrname | wc -w`
echo "User $usrname is running $cnt processes."
```

Save the script as *script10.sh*. Let us try to understand the working of this script. The first command clears the content on the screen. Then a message is displayed for the user to enter a user name. The read command then assigns the string read from the keyboard to variable *usrname*. Then we have combined four commands namely ps, cut, grep and wc using pipe. The ps -ef command displays list of processes being run by all the users of the system. Its output is then given to the cut command. The cut command extracts the first field (username) from this output. The extracted list of first field is then given to the grep command. The grep command finds all the users that match with the value that is extracted from variable *usrname*. This matched list is then given to the wc command that counts the occurrence of the given word (username). Finally this word count is assigned to variable *cnt*. The last command then displays the actual output needed. Figure 8.2 shows the sample output of the script.



```
File Edit View Terminal Help
Enter username: root
User root is running 103 processes.
administrator@ubuntu:~$
```

Figure 8.2 : Output of Script 10

As mentioned earlier we may remove a process and release some memory space if so required. To remove the process from memory we use the *kill* command. For example if we issue a command

Skill -9 101

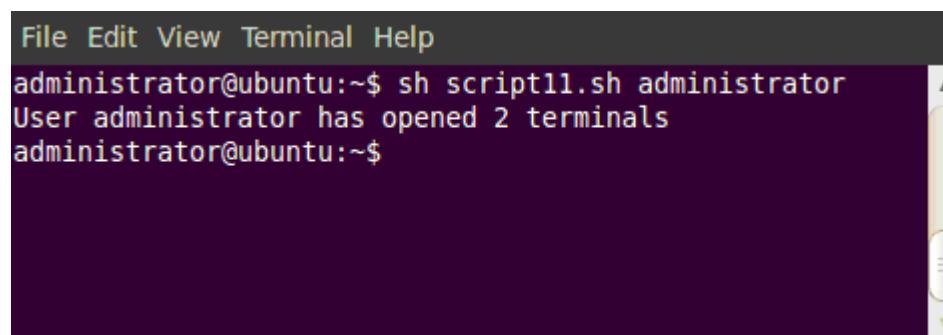
Then the process with PID=101 will be forcibly removed from the memory. Let us have a look at script similar to *script10.sh* that accepts user name as a command line argument and tells us how many terminals that user is using. The code of the script is given in the box below:

```
#Script 11: Script to find out how many terminals a user has opened.
cnt=`who | cut -d " " -f 1 | grep -o $1 | wc -w`
echo "User $1 has opened $cnt terminals"
```

Save the script as *script11.sh*. Observe that the script we created in the previous example used variables. In this script we have made use of a command line argument. The entity \$1 here refers to a command line argument. To execute this script type the command as mentioned below:

\$sh script11.sh administrator

You must have observed that the script is executed in the similar manner as we have executed the previous script. But here we have specified additional value “administrator” (readers may specify any name of their choice). Linux stores the values provided through command line in dollar variables, named \$1, \$2, \$3 and so on. First argument will be stored in \$1, second in \$2, third in \$3 and so on till \$9. These arguments are known as command line arguments. The output of the script is shown in figure 8.3.



```
File Edit View Terminal Help
administrator@ubuntu:~$ sh script11.sh administrator
User administrator has opened 2 terminals
administrator@ubuntu:~$
```

Figure 8.3 : Output of Script 11

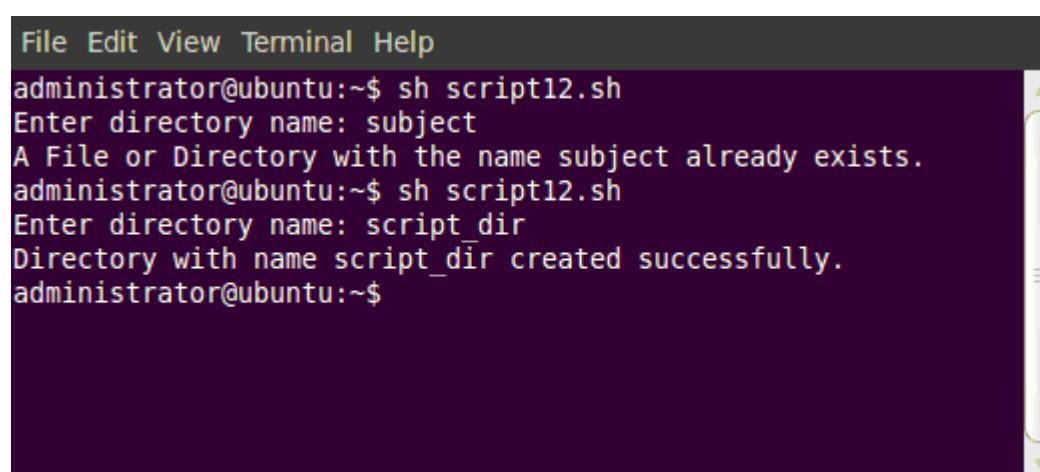
Let us try to understand the working of this script. In the first statement after comment we have combined four commands namely who, cut, grep and wc using pipe. The *who* command displays list of all users that have logged into the system. Its output is then given to the cut command. The cut command extracts the first field from this output. The extracted list of first field is then given to the grep command. The grep command then finds out all the users that match with the entered command line argument value (\$1 = administrator). This matched list is then given to the wc command that counts the occurrence of the given word (username). Finally this word count is assigned to variable cnt. The last command then displays the actual output needed.

Decision Making Tasks

Let us say an administrator wants to create a directory, he can do it using an mkdir command. But if he uses a script for creating a directory he can generate appropriate messages also. Let us write a script that allows an administrator to create a directory.

```
#Script 12: Script to create a directory with appropriate message.  
echo -n "Enter directory name: "  
read mydir  
if [ -d $mydir -o -f $mydir ]  
then  
    echo "A File or Directory with the name $mydir already exists".  
    exit 0  
fi  
mkdir $mydir  
echo "Directory with name $mydir created successfully."
```

Save the script as *script12.sh*. Observe that in this script we have used an if-then-fi construct. This construct in shell scripts allows us to perform decision making. The *if* statement of Linux is concerned with the exit status of a test expression. The exit status indicates whether the command was successfully executed or not. The exit status of command is 0 if it has been executed successfully; otherwise it is set to 1. Figure 8.4 shows the output of the script.



```
File Edit View Terminal Help  
administrator@ubuntu:~$ sh script12.sh  
Enter directory name: subject  
A File or Directory with the name subject already exists.  
administrator@ubuntu:~$ sh script12.sh  
Enter directory name: script_dir  
Directory with name script_dir created successfully.  
administrator@ubuntu:~$
```

Figure 8.4 : Output of Script 12

Observe the output of figure 8.4 carefully. In one case we get message indicating that the directory already exists and in second case we are able to create a directory with the specified name. Note that the condition in the above script is enclosed in a square bracket. There should be one space after opening square bracket and one before closing square bracket. If the condition is evaluated to true then statements typed inside *then* block will be executed otherwise not. The end of the *if* statement is indicated by *fi* statement. Also note that the *then* keyword should be typed below *if* statement else we will get error. The -d, -f -o options used in the script will be discussed later in the chapter.

We can use the following four decision making instructions while creating a shell script in Linux:

if-then-fi

if-then-else-fi

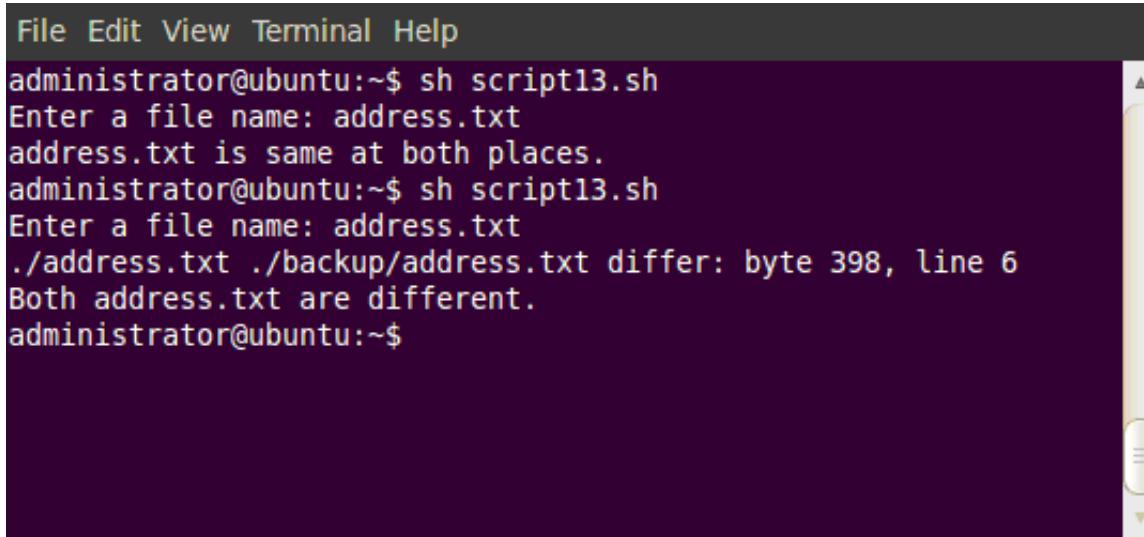
if-then-elif-then-else-fi

case-esac

It is a normal practice to copy a file and keep in it another directory (maybe for the purpose of backup). The user many times gets confused whether both the files are same or different. Let us write a script that helps user compares such files. The script when executed compares both the files using the *cmp* command. Based on the output of the *cmp* command it then displays appropriate messages.

```
#Script 13: Script to compare files.  
echo -n "Enter a file name: "  
read fname  
if cmp ./fname ./backup/fname  
then  
    echo "$fname is same at both places."  
else  
    echo "Both $fname are different."  
fi
```

Save the script as *script13.sh*. Here we first accept a file name from the user. To keep the script simple as of now we have used relative paths for directory (user can convert it absolute path). We have also assumed that the file names at both the locations are same. Figure 8.5 shows the output of the script.



```
File Edit View Terminal Help
administrator@ubuntu:~$ sh script13.sh
Enter a file name: address.txt
address.txt is same at both places.
administrator@ubuntu:~$ sh script13.sh
Enter a file name: address.txt
./address.txt ./backup/address.txt differ: byte 398, line 6
Both address.txt are different.
administrator@ubuntu:~$
```

Figure 8.5 : Output of Script 13

Here we have executed the script twice. In the first run both the files contents are same hence we get the message that both files are same. Before second run we have modified the file in the current directory hence we are getting the message that files are different.

In previous chapter we have written a small script to welcome the user that has logged in the system. Let us modify it further so that we display a proper welcome message (Good morning, Good afternoon or Good evening depending on the time the user has logged in).

```
#Script 14: Script to display welcome message to the user.

clear
hour=` date +"%H"`
username=`who am i | cut -d " " -f 1`
if [ $hour -ge 0 -a $hour -lt 12 ]
then
    echo "Good Morning $username, Welcome to Ubuntu Linux Session."
elif [ $hour -ge 12 -a $hour -lt 18 ]
then
    echo "Good Afternoon $username, Welcome to Ubuntu Linux Session."
else
    echo "Good Evening $username, Welcome to Ubuntu Linux Session."
fi
```

Figure 8.6 shows the output of the script. The output will vary depending on when the user has logged in.

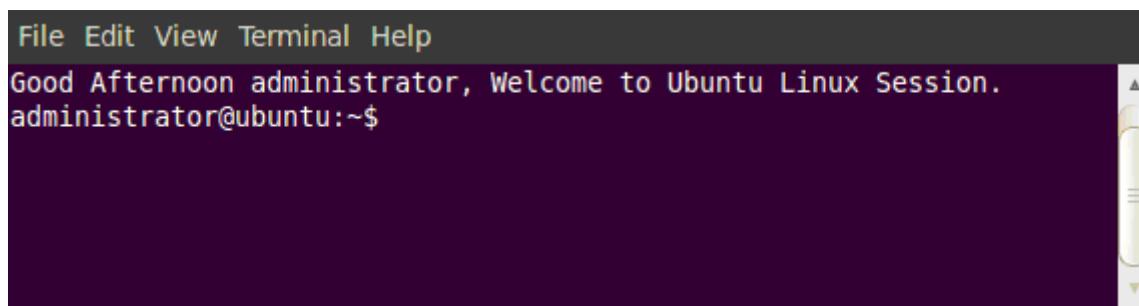


Figure 8.6 : Output of Script 14

The test command

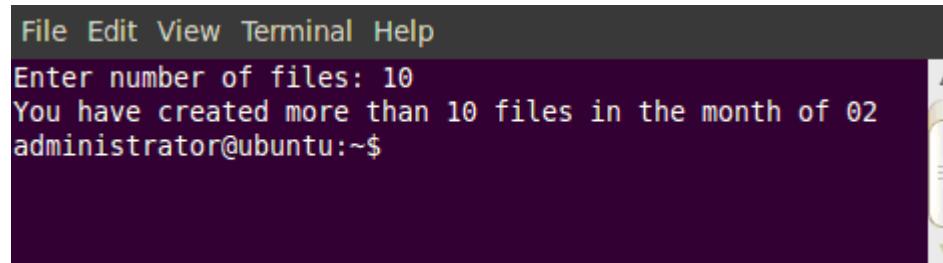
It is possible to use different forms of if statements. Linux also provides test command which can be used in place of square brackets used in previous scripts. Let us write script to check whether a user has created more than some specified files in a given month or not.

```
#Script 15: Script to see whether user has created more than specified files in a month.  
clear  
cnt=`ls -l | grep -c [-]"$1"  
echo -n "Enter number of files: "  
read nfile  
if test $cnt -gt $myfile  
then  
    echo "You have created more than $myfile files in the month of $1"  
else  
    echo "You have not created more than $myfile files in the month of $1"  
fi
```

Let us try to understand the script. Here we have defined a variable named *cnt*. This variable is assigned the total count of the files created in a specified month. To find out the number of files we have used two commands namely *ls* and *grep*. The *ls -l* command is used to display details of files and directories. Its output is then given to the *grep* command that matches regular expression *[-]"\$1"*. The month is specified as two digit numeric value and accepted through command line argument assigned to *\$1*. Then we have defined a variable *myfile* that stores the value of number of files that we want to compare with. The *-gt* option in the *test* indicated greater than comparison. Here we are checking whether the value of *cnt* is greater than the value of *myfile* or not. If the value of *cnt* is greater than the value of *myfile* we print the message “You have created more than *\$myfile* files in the month of *\$1*”, where *\$myfile* and *\$1* are replaced with appropriate values. Otherwise we print message “You have not created more than *\$myfile* files in the month of *\$1*”.

Figure 8.7 shows the output of the script when we issue a command shown below on the command prompt.

```
$sh script15.sh 02
```



```
File Edit View Terminal Help
Enter number of files: 10
You have created more than 10 files in the month of 02
administrator@ubuntu:~$
```

Figure 8.7 : Output of Script 15

The if statement can work with numerical values, strings and files. In turn the tests performed are known as numerical test, string test and files test respectively. Observe that we have used options like -d, -f, -o, -a, -ge, -lt and -gt in the scripts created so far. All these options allow us to perform various types of condition matching.

Relational Operators

The numerical test is performed using relational operators. The options -ge, -lt and -gt refers to relational operators. These operators are used to compare values of two numeric operands. Table 8.2 lists the relational operators that can be used in shell scripts along with their usage.

Operator	Usage
-gt	greater than
-lt	less than
-ge	greater than or equal to
-le	less than or equal to
-ne	not equal to
-eq	equal to

Table 8.2 : Relational operators

Logical Operators

For taking precise and appropriate decisions many times a user needs to combine one or more conditions. To combine conditions we make use of logical operators. Table 8.3 lists the logical operators along with their usage.

Operator	Usage	Minimum conditions that can be combined	Output
-a	AND	Two	True if both conditions are true, false otherwise
-o	OR	Two	True if any one condition is true, false only if both conditions are false
!	NOT	One	Converts true to false and vice versa

Table 8.3 : Logical Operators

File Operators

It is also possible to use *if* statement to check the status of file or a directory. Similar to the relational operators we have file operators that allows us to check the status of a file. These operators are used as a condition within the *if* statement. By using file operators we can come to know whether a specified name is an ordinary file or a directory. We can also find out the status of file permissions using them. Table 8.4 lists usage of these options.

Condition Tested	Output
-s name	True if a file with the specified name exists and has size greater than 0.
-f name	True if a file with the specified name exists and is not a directory.
-d name	True if a directory with the specified name exists.
-r name	True if a file with the specified name exists and the user has read permission on it.
-w name	True if a file with the specified name exists and the user has write permission on it.
-x name	True if a file with the specified name exists and user has execute permission on it.

Table 8.4 : File test conditions

Many times administrator needs to find whether a specified file has size equal to zero or not. He can then perform maintenance operations like delete the file in case its size is zero. He may additionally need to find whether write permissions on the file is set or not. Let us write a script that allows administrator to check the file size and know what permissions are allocated to the file.

```

#Script 16: Script to check file size.

echo -n "Enter a file name: "

read fname

if [ -s $fname -a -w $fname ]
then
    echo $fname has size greater than 0 and user has write permission on it.
else
    echo $fname has size 0 or user does not have write permission on it.
fi

```

Save the script as *script16.sh*. Here the statement **if [-s \$fname -a -w \$fname]** has multiple conditions. The result of the *if* statement is evaluated when both the conditions give us some output. Table 8.5 lists the value that can be generated as output when the if statement is evaluated and figure 8.8 shows different output of the script.

-s \$fname	Reason	-w \$fname	Reason	if [-s \$fname -a -w \$fname]
False	File size = 0 or File does not exists	False	Write permission not set	False
False	File size = 0 or File does not exists	True	Write permission set	False
True	File size > 0	False	Write permission not set	False
True	File size > 0	True	Write permission set	True

Table 8.5 : Outputs of if [-s \$fname -a -w \$fname]

```

File Edit View Terminal Help
administrator@ubuntu:~$ sh script16.sh
Enter a file name: script5.sh
script5.sh has size 0 or user does not have write permission on it.
administrator@ubuntu:~$ sh script16.sh
Enter a file name: script10.sh
script10.sh has size greater than 0 and user has write permission on it.
administrator@ubuntu:~$ 

```

Figure 8.8 : Output of Script 16

The if-then-fi and if-then-else-fi statements used so far allow us to test limited set of conditions. In case a user needs to perform more number of tests these statements may not be of much help. In such cases we may use the if-then-elif-then-else-fi or the case statement.

Let us write a script that accepts three files from user and displays the file which has maximum size.

#Script 17: Script to find the file with the maximum size.

```
clear

echo -n "Enter name of first file: "
read fname1

echo -n "Enter name of second file: "
read fname2

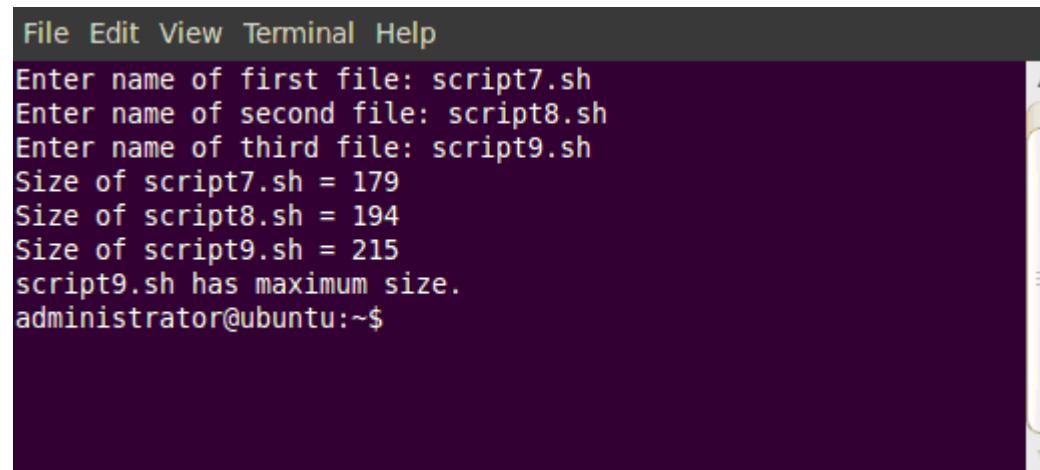
echo -n "Enter name of third file: "
read fname3

fsize1=`wc -c $fname1 | cut -d " " -f 1`
fsize2=`wc -c $fname2 | cut -d " " -f 1`
fsize3=`wc -c $fname3 | cut -d " " -f 1`

echo Size of $fname1 = $fsize1
echo Size of $fname2 = $fsize2
echo Size of $fname3 = $fsize3

if [ $fsize1 -eq $fsize2 -a $fsize1 -eq $fsize3 ]
then
    echo "All files have same size"
elif [ $fsize1 -gt $fsize2 -a $fsize1 -gt $fsize3 ]
then
    echo "$fname1 has maximum size."
elif [ $fsize2 -gt $fsize1 -a $fsize2 -gt $fsize3 ]
then
    echo "$fname2 has maximum size."
else
    echo "$fname3 has maximum size."
fi
```

Save the script as *script17.sh*. The six statements after the clear command are used to accept the file names from the user. The next three statements calculate size of the files, later these sizes are displayed to the user. Finally using the if condition, the script finds out the file that has maximum size. Figure 8.9 shows the output of the script.



```
File Edit View Terminal Help
Enter name of first file: script7.sh
Enter name of second file: script8.sh
Enter name of third file: script9.sh
Size of script7.sh = 179
Size of script8.sh = 194
Size of script9.sh = 215
script9.sh has maximum size.
administrator@ubuntu:~$
```

Figure 8.9 : Output of Script 17

The case statement

The if-then-elif-then-else-fi statement looks clumsy as number of comparison grows. The alternate option for checking such conditions is to use a case statement. Let us write a script that allows us to accept a choice from the user and perform different file operations based on the entered choice.

```
# Script 18: Script to perform various file and directory operations.

echo "1 - Display Current Dir "
echo "2 - Make Dir "
echo "3 - Copy a file "
echo "4 - Rename a file "
echo "5 - Delete a file "
echo "0 - Exit "

echo -n "Enter your choice [0-5] : "
read choice

case $choice in
 1)
    echo $PWD
    ;;
 2)
    echo "Creating new directory"
    ;;
 3)
    echo "Copying file"
    ;;
 4)
    echo "Renaming file"
    ;;
 5)
    echo "Deleting file"
    ;;
 0)
    exit
    ;;
esac
```

```
2)
echo -n "Enter name of the directory to be created: "
read dname
if [ -d $dname ]
then
echo "Directory with the name $dname already exists."
exit 0
else
mkdir $dname
echo "Directory $dname created successfully."
fi
;;
3)
echo -n "Enter source file name : "
read sfile
echo -n "Enter destination file name : "
read dfile
cp -u $sfile $dfile
;;
4)
echo -n "Enter old file name : "
read oldf
echo -n "Enter new file name : "
read newf
mv $oldf $newf
;;
5)
echo -n "Enter file name to delete : "
read fname
rm $fname
;;
```

```

0)
    exit 0
;;
*)
    echo "Incorrect choice exiting script."
esac

```

Save the script as *script18.sh*. Observe that for each operation that we need to perform we have written different section. When a user enters a numeric value between 0 and 5, it is assigned to the variable *choice*. The case statement extracts the value of variable *choice*, the control is transferred to the section with a matching value specified before the closing round brackets. All the statements written within that section are executed till two semicolons (;;) are encountered. Once these semicolons are encountered the control is transferred to the line after the end of the case statement. The end of case statement is specified by esac keyword. The shell then starts executing statements written after the end of case statement.

If user enters any value that does not match any of the case value specified, then the control is transferred to the section that has asterisk (*) as its value. If specified, this section allows a user to exit the script or perform additional processing after displaying an appropriate message. Figure 8.10 shows us different output of script 18.

```

File Edit View Terminal Help
administrator@ubuntu:~$ sh script18.sh
1 - Display Current Dir
2 - Make Dir
3 - Copy a file
4 - Rename a file
5 - Delete a file
0 - Exit
Enter your choice [0-5] : 1
/home/administrator
administrator@ubuntu:~$ sh script18.sh
1 - Display Current Dir
2 - Make Dir
3 - Copy a file
4 - Rename a file
5 - Delete a file
0 - Exit
Enter your choice [0-5] : 2
Enter name of the directory to be created: LinuxScript
Directory with the name LinuxScript already exists.
administrator@ubuntu:~$

```

Figure 8.10 : Output of Script 18

The syntax of case statement is:

```
case variable_name in
    value1)
        Command1
        Command 2
        ....
        ;;
    value 2)
        Command 1
        Command 2
        ....
        ;;
        *)
    Command 1
    Command 2
    ....
    ;;
esac
```

Note :

We can assign numeric, character or string values to the variable that accepts the choice. In case we assign string values then within the case it should be enclosed between single quotes. For example if we accept string *abc* then within the case statement it should be mentioned as ‘abc’.

Handling Repetition

Cleaning of disk space is a normal operation that the administrator needs to perform. Let us write a simple script that assists the administrator in finding zero sized file and delete it. The script to perform the operation is given below:

```
#Script 19: Script to delete zero sized files.

echo -n "Enter directory name : "
read dname
if [ ! -d $dname ]
then
    echo Directory $dname does not exist.
```

```

else
ctr=0
for i in `find "$dname/" -type f -size 0c`
do
rm $i
echo File $i" : deleted"
ctr=`expr $ctr + 1`
done
if [ $ctr -gt 0 ]
then
echo "$ctr zero sized files have been deleted."
else
echo "No zero sized files present in directory."
fi
fi

```

Observe that in this script we have used a statement *for i in ‘find “\$dname/” -type f -size 0c’*. This statement is used to repeat some actions again and again. Figure 8.11 shows the output of script 19.

```

File Edit View Terminal Help
Enter directory name : LinuxScript
File LinuxScript/test : deleted
File LinuxScript/test1 : deleted
File LinuxScript/test2 : deleted
3 zero sized files have been deleted.
administrator@ubuntu:~$

```

Figure 8.11 : Output of Script 19

While writing scripts for certain tasks we may require performing an action multiple times. The process of repeating the same commands number of times is known as looping. Linux provides three keywords namely *for*, *while* and *until* that can be used to perform repetitive actions.

In script 19 we have used *for* statement. The *for* loop allows us to specify a list of values in its statement. The loop is then executed for each value mentioned within the list. The general syntax of *for* statement is shown below:

```

for control-variable in value1, value2, value3.....
do
    command 1
    command 2
    command 3
done

```

Another activity that administrator regularly performs is taking backup of files. Let us say he needs to take backup of particular type of files. In such a case, taking backup of one file at a time does not make sense. Creating an exact copy at another location will also waste storage space. In such cases an administrator can use a script that first creates a backup directory in the folder where the files are located. Then the files which needs backup are copied into it. The directory is then compressed and finally moved to a new location. The script written below performs this action.

```

#Script 20: Script to backup and compress desired files from current location.

clear
dat=`date +"%d_%m_%Y"'
bdir=backup_$dat
if [ ! -d $bdir ]
then
    mkdir $bdir
else
    echo "Directory with name $bdir already exist."
    exit 0
fi
echo -n "Enter the extension of the files to backup: "
read fextn
ctr=0
for i in `ls -1 *.$fextn`
do
    cp $i ./$bdir
    ctr=`expr $ctr + 1`
done

```

```

if [ $ctr -gt 0 ]
then
tar -czf $bdir.tar $bdir
cd $bdir
rm -r *.*
cd ..
rmdir $bdir
echo "All files with extension .$fextn stored in $bdir.tar"
else
rmdir $bdir
echo "No files with the extension found."
fi

```

Save the script as *script20.sh*. Let us understand how the script works. Initially we have defined two variables namely *dat* and *bdir*. The *dat* variable is assigned the value of current date in the specified format. For example if the current date is 21 February 2013, then variable *dat* will be assigned value *21_02_2013*. The variable *bdir* is then assigned value *backup_21_02_2013*. Then we check whether such a directory exists or not. If it does not exist we create this directory otherwise we exit with the message saying that the directory already exists. If we create a directory then we ask the user to enter a file extension. The script looks for the files with specified extension in the current directory and if found copies them in the backup directory. Once all files are copied, the backup directory is compressed (packed) using the tar command. The *tar -czf \$bdir.tar \$bdir* statement performs this operation. Here we create a tar file named *backup_currentdate.tar*. Then we empty the contents of the backup directory and delete it. In case we do not find any files with the extension specified we display appropriate message. The administrator if he wants now can move the compressed tar file to the location he desires. We can uncompress the tar file by using the command *tar -xvf filename*.

Repetition: while statement

We can also use the *while* statement for looping. It repeats the set of commands specified between keywords *do* and *done* statements as long as the condition specified as an expression is true. Let us write a script that allows administrator to remove a specified number of files from a directory.

#Script 21: Script to delete specified number of files from a directory.

```

clear
echo -n "Enter the name of directory from where you want to delete: "

```

```

read dname
if [ -d $dname ]
then
cd $dname
echo -n "Enter the number of files you want to delete: "
read fdel
ctr=1
while [ $ctr -le $fdel ]
do
echo -n "Enter the name of the file to be deleted: "
read fname
if [ -f $fname ]
then
rm $fname
echo "$fname deleted successfully."
else
echo "File with name $fname not found."
fi
ctr=`expr $ctr + 1`
done
else
echo "Directory $dname does not exist."
fi
cd ..

```

Save the script as *script21.sh*. Let us understand how the script works. Initially the user is prompted to enter a directory name. The *dname* variable is assigned this value. Then we check whether such a directory exists or not. If it exists we change to that directory and ask user the number of files he wants to delete. Then we start a while loop that finds the files to be deleted. If the file is found we delete it else we display a message indicating file not found. The loop is continued till the value of variable *ctr* is less than or equal to the number of files specified by the user. Once the operation is over we go back to the parent directory. The syntax of while loop is shown below:

```
while [ test_condition ]
do
    command or set of commands
done
```

Repetition: until statement

Another method to execute repetitive statements is to make use of the *until* statement. The until loop is similar to the while loop. However, the *until* loop executes till the condition is false and the while loop executes till the condition is true.

So far, we have seen how we can use decision-making and looping constructs to write shell scripts. Script 19 is an example of shell script which uses some of the constructs discussed above. It is a menu driven script demonstrating until-loop, to display list of files in a current directory, changing password, displaying current date and time and searching a word from a file.

#Script 22: Script to perform operations till user decides to exit.

```
choice=y
until [ $choice = n ]
do
    clear
    echo "....."
    echo "      Choose an option from menu given below      "
    echo "....."
    echo "a: List of files and directories in a current directory."
    echo "b: Display current working directory"
    echo "c: Display current date and time"
    echo "d: Searching a word from file"
    echo "e: Exit"
    echo "  "
    echo "....."
    echo -n "Enter your choice [a-e]: "
    read ch
    case $ch in
        a)
```

```
ls -l
;;
b)
echo "You are working in `pwd`"
;;
c)
echo "Current date and time is `date`"
;;
d)
echo -n "Enter the word to be searched: "
read word
echo -n "Enter the file name from which the word is to be searched: "
read file
if [ -f $file ]
then
grep $word $file
else
echo -n "File with name $file does not exist."
fi
;;
e)
exit
;;
*)
echo "Incorrect choice, try again."
;;
esac
echo -n "Do you want to continue? : "
read choice
done
```

Save the script as *script22.sh*. When user executes the script he will be shown a menu and asked to enter a choice. Depending on the choice entered an action from the case will be executed. Enter different choice each time and see the output. The script will keep on executing till user enters *e* as a choice in which case the exit statement within the case is executed or he enters *n* when the question “Do you want to continue?” is asked. Figure 8.12 shows the output of script 22.

```
File Edit View Terminal Help
Choose an option from menu given below
a: List of files and directories in a current directory.
b: Display current working directory
c: Display current date and time
d: Searching a word from file
e: Exit

Enter your choice: [a-e] c
Current date and time is is Fri Feb 22 14:36:21 EST 2013
Do you want to continue? : n
administrator@ubuntu:~$
```

Figure 8.12 : Output of Script 22

Functions in script

Linux shell script also provides us the feature of creating functions. Functions are small subscripts within a shell script. They are used make the scripting more modular. Using functions we can improve the overall readability of the script. The function used in shell script do not return a value, they return a status code. Let us see one script that assists the user in finding out how many files were created on current date or when a particular file was last modified.

```
#Script 23: Script to show use of function.

file_today(){
    cur_date=`date +%Y-%m-%d`
    cnt=`ls -l tr | grep "$cur_date" | wc -l`
    echo "Current date is : $cur_date"
    echo "No. of files created today : $cnt"
}
```

```

modified_today(){
if [ -f "$1" ]
then
stat -c %y "$1"
else
echo """$1" does not exist"
fi
}
choice=y
until [ $choice = n ]
do
clear
echo "....."
echo "      Choose an option from menu given below      "
echo "....."
echo "a: List of files created today."
echo "b: Display last file modification date."
echo "c: Exit"
echo "  "
echo "....."
echo -n "Enter your choice [a-c]: "
read ch
case $ch in
a)
file_today
;;
b)
echo -n "Enter a file name: "
read fname
modified_today $fname
;;

```

```

c)
exit
;;
*)
echo "Incorrect choice, try again."
;;
esac
echo -n "Do you want to continue? : "
read choice
done

```

Save the script as *script23.sh*. Observe that in script 23 we have used two functions namely `file_today()` and `modified_today()`. The opening and closing parenthesis after a variable name indicates that it is a function. When user enters *a*, function `file_today()` that contains code for finding the files created on a current date is called and executed. Similarly when user enters *b* he is prompted to enter a file name. This name is then passed to function `modified_today()` that checks if the files exists or not. If the file exists its last modification date is displayed otherwise appropriate message is displayed. Figure 8.13 shows the output of script 23.

```

File Edit View Terminal Help
.....
Choose an option from menu given below
.....
a: List of files created today.
b: Display last file modification date.
c: Exit

.....
Enter your choice [a-c]: a
Current date is : 2013-02-22
No. of files created today : 12
Do you want to continue? : n
administrator@ubuntu:~$
```

Figure 8.13 : Output of Script 23

Summary

In this chapter we have seen how a shell script can be used for several tasks of system administration. We saw how decision making and looping constructs can be used in shell scripts. We also saw how we can write a shell script in the form of functions. The shell script thus offers the facility to combine the power of various inbuilt commands. This makes it almost equivalent to a higher level programming language.

EXERCISE

Laboratory Exercises

Write a shell script to perform the following operations:

- (a) To accept two file names from user. The script should check whether the two file's contents are same or not. If they are same then second file should be deleted.
- (b) To count and report the number of entries present in each subdirectory mentioned in the path, which is supplied as a command-line argument.
- (c) To list name and size of all files in a directory whose size is exceeding 1000 bytes (directory name is to be supplied as an argument to the shell script).
- (d) To rename a file.
- (e) To convert all file contents to lower case or upper case as specified by user.
- (f) To find out available shells in your system and in which shell are you working.
- (g) To find out the file that has minimum size from the current directory.





In current scenario with the advent and reach of digital technologies, almost all data is stored and managed electronically. Day to day transactions in small and medium enterprises are recorded in systems termed as information systems. Users of information system are growing multifold. Users expect flexibility in accessing data irrespective of the places they are at and irrespective of where the data is stored. Organizations now being global, information systems are not now limited to storage and maintenance of enterprise transactions which were understood as replacement of registers/files maintained in organization. Search engines, reservation systems, online banking, online tax filing, stock trading are examples of systems used by millions of users simultaneously. Today usage of information systems is becoming inevitable. All of us directly or indirectly act as users of one or other information systems. Doctors, engineers, shopping store owners, broking agents, educators, and students, none of the entities have remained untouched. Information systems today has also become a strong helping hand in taking decisions. To take best possible decision, everyone needs correct, precise and relevant information on time. It is obvious that the perfect decision depends on the accuracy of information available. Information is the output generated through processing of raw data. So, let us first understand the two important terms i.e., Data and Information.

Data and Information

Many a times the word Data and Information are used interchangeably. So it is necessary to have a clear idea of both the words. Let us first understand Data and Information by their possible definitions.

“All kinds of facts, figures and details related to people, places, things or events is known as data.” Data may be in any form. It may be written, oral, computerized or non-computerized. Data in its original form may not be of much use. Data must be processed in a proper way to generate the useful and meaningful information. Looking to the importance of data for decision-making process, many business houses consider it as important asset of the business.

“Information on the other hand can be described as the required result obtained from processing of the data.” This means that, the Information is nothing but “The Processed Data”. Information is important because it forms the foundation for Decision Making. Taking timely decisions, based on correct and relevant information may be beneficial for us. On the other side, taking decision based on incomplete or incorrect information may badly affect the business.

Let us clear the concept of data and information with an example. Suppose you and your team in a school decide to send birthday wish to each student of the school. This idea requires collection of birth dates like 10th September 1998. The dates collected act as data in this case. If today is 10th September, you will find out names of the boys and girls whose birthday falls on this date. The set of names in your list now acts as information.

Database and Data Model

Having learnt about data and information, let us now have a look at database. We all use one or more type of databases in our day-to-day life. We keep a diary containing address and phone

numbers of our relatives and friends at home. Same way your school maintains details of all the students enrolled in the school. An attendance record of every student has to be maintained by school. Indian government is issuing unique identity numbers to each citizen under Adhar project. Thus a detail of each person has to be maintained. Passport office maintains details of the issued passports. Doctors maintain medical history of each patient. Railway department maintains details of passengers. All these are examples of records.

As per the common definition of the database “*Database is a collection of related data items stored in an organized manner*”. As discussed earlier, data is stored because data is to be used. Thus the word *related* and *organized* is very important in definition of database. You may compare storing data to arranging things in your room. You might be receiving instructions from your parents and teachers to keep your belongings at a proper and pre-decided place. You may fix a place of school bag, stationary, books etc. You may label the compartments in the book shelf. For example, science, general knowledge etc. If your father gifts you a book on human body, you may decide to keep it in science section. Next time when you want to read the book, it would be very easy for you to find out the same book. Logical arrangement of things always makes searching easier as and when required.

Similar to your books, clothes in wardrobe also require logical arrangement. However, you might have noticed that structure of wardrobe and book shelf is normally different. The place designated to store jewellery or kitchen utensils has different structure. Above examples indicate, that depending on the things to be stored, structure of shelves is to be designed.

Similarly, databases are designed according to certain rules. The underlying structure of database is known as data model. Data model describes way of storing and retrieving the data. There are different data models like hierarchical data model, network data model and relational data model. In this text, we will be discussing relational model. Different Database Management Systems are available in the market, which are based on one of the above mentioned data models. Let us now understand what Database Management Systems are.

Database Management Systems

To work with computerized database, one must first define and create a database. However, just creating database is not the only task. In order to keep the database ready with latest up-to-date data, following basic tasks are expected to be performed regularly with as much ease as possible.

- Adding data into the database as and when transaction takes place.
- Editing the content of the database with the corrections required as and when noticed.
- Removing unnecessary data from the database, if any.
- Organizing the database in a proper manner.
- Retrieval of data as per user requirement.
- Securing data stored in database.
- Maintaining backup of database.
- Allowing multiple users to access database simultaneously.
- Allowing data to be stored in multiple languages.
- Allowing data to be stored at multiple places.

All tasks related to database handling is known as database management. Naturally, we need special software to do this kind of work. Such software are known as Database Management Systems

(DBMS). DBMS can be defined as system designed for efficient storage, maintenance and retrieval of data. In short the DBMS software is what makes database practical and more usable. In today's world a database itself is useless if there is no DBMS associated with it for accessing its data. Oracle, Ingress, DB2, SQL server, MYSQL, MS Access, Base etc. are the names of some popular DBMS software available in the market. In this book, we will discuss about a Base Database called as OpenOffice Base. From now onwards we call it as Base. As we are aware of terms data, information, database and database management system, let us proceed further and get acquainted with Base, an open source database available as part of Open Office suite. Base is based on a Relational data model.

Introduction to Base

Base is Relational Database Management System (RDBMS) as it follows relational model. As discussed earlier, the manner in which information is organized can have a profound effect on how easy and efficient it is to access and manage. Perhaps the simplest but most versatile way to organize information is to store it in tables. The relational model is centered on this idea: the organization of data into collections of two-dimensional tables called "relations." Designed by E.F. Codd, relational model is based on the theory of sets and relations of mathematics. A DBMS that is based on relational model is called as RDBMS. Relational model is the most successful data model. One of the reasons behind the success of relational model is its simplicity. It is easy to understand the data relationship and easy to manipulate.

Base is a collection of related data objects known as Tables, Queries, Reports and Application Modules. Let us now understand objects that can be created using Base in brief. Each of these objects are discussed in detail in later chapters.

Table : Being RDBMS, Table is the basic unit for storing data in database designed using Base. Tables are organized in the form of columns and rows. To decide what information would be stored in a table and what would be the column titles, database designer needs to first decide entities in database. Entities are real world objects about which information is to be stored in database. For example, while creating a database design for Employee Attendance System, the first entity which comes to our mind is Employee. Details of each employee need to be stored in database. These details are stored in the form of rows. Each entity generally has collection of attributes associated to it. For example, each employee would have attributes like first name, last name, address, date of birth, contact number etc. Attributes of an entity are represented in the form of columns. Figure 9.1 shows how data is represented in relational model and what are the terms used to refer to various components of a table.

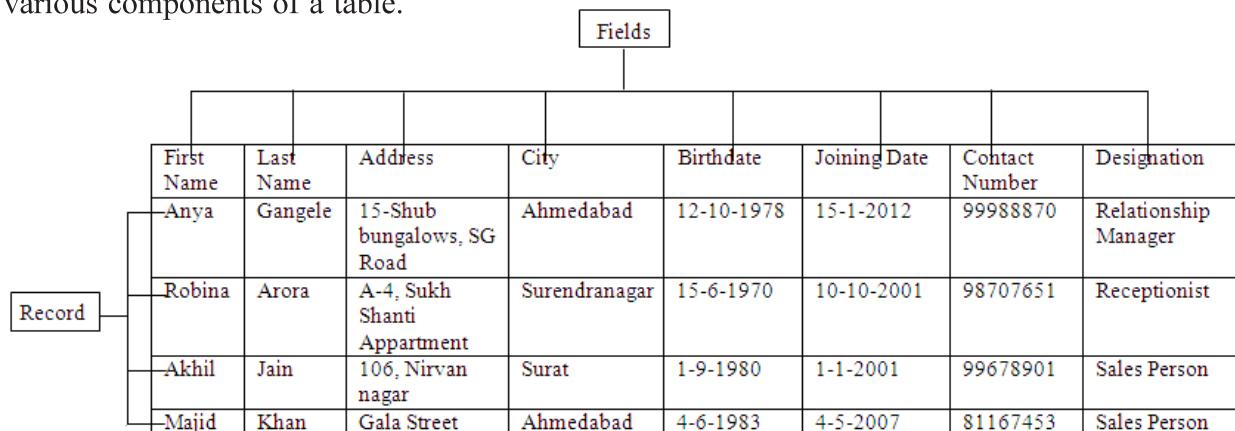


Figure 9.1 : Employee table

As can be seen in figure 9.1, attributes of an entity are also termed as fields. Thus in employee table, First Name, Last Name, Address, City, Birthdate, Joining Date, Contact Number and Designation are fields. Collection of all these fields forms a record. Thus << ‘Anya’, ‘Gangele’, ‘15-Shub bungalows, SG Road’, ‘Ahmedabad’, ‘12-10-1978’, ‘99988870’, ‘15-1-2012’, ‘Relationship Manager’>> is a record.

Forms : We need to enter new records into the table, edit existing records, and view the records available with the table. Form is an object which allows entering the data in the table and editing or deleting existing data in the table. It consists of format, style and widgets like radio button, list boxes that provides easy and systematic way to populate table with data.

Queries : A question asked within the database environment is known as Query. For example, how many Android based mobiles were sold in July in current year? Query displays subset of data contained in various tables of a database.

Reports : The output of Query in Base is displayed in the form of rows and columns. As we discussed earlier, the available data is to be processed for required information and an output comprised of information is to be generated. Obviously, user expects more formal and attractive output. Base has a facility to prepare effective professional report. The presentation of information in an organized and readable format as per the user requirement is known as Reports. Various complex reports could be generated that can help in taking decisions by management in organizations.

Creating a Sample Database for an Application

During the entire journey of learning Base, we will move towards creation of a small application for a hypothetical Modern Electronic Store. This store sells devices like Mobiles, Camera, Handy Cams, Laptops, Netbooks, ipads, Smartphones and accessories related to them. Products are procured from various suppliers. Store has physical shop as well as web shop. Thus customers can either purchase items walking in the shop or place orders online. Sales Persons are paid special incentives for converting leads to customers. Let us name the application we are aiming for as Store Management System. Let us design a sample database for Store Management System. In later chapters, we will design Forms and Reports required for this system. Following steps help us to design the required database.

Step 1 :

Decide about whom data is to be stored? In our example, we will store data pertaining to products, suppliers, employees and customers. All of these are known as *Entities*. As discussed earlier, entities can be defined as any person, thing, place or an object about which we store the data in the database. For each entity a separate table should be designed. Thus we will design four tables named Product, Supplier, Employee and Customer.

Step 2 :

Decide what attributes of each entity is to be stored. For example, what attributes of product, supplier, employee and customer should be stored? These attributes will become fields of a Table. *Attributes (Fields) can be defined as characteristics of an entity.* We will store details of suppliers, employees, customers and products as shown in Table 9.1, Table 9.2, Table 9.3 and Table 9.4.

Supplier	
Attributes	Description
Supplier Name	Name of supplier, it could be name of company like Samsung India
Address Line 1	Address from where products are dispatched
Address Line 2	Street and/or area
City	City from where products are dispatched
Pincode	Pincode of area within a city
Contact Person Name	Person to be contacted
Email Id	E-mail address of contact person
Contact Number	Contact number of office

Table 9.1 : Attributes for Supplier entity

Employee	
Attributes	Description
Firstname	Name of the Employee
Lastname	Surname of the Employee
Address Line 1	Address of the Employee
Address Line 2	Street and/or area
City	City
Pincode	Pincode of area of above city
Email Id	E-mail address of Employee
Mobile Number	Mobile number of Employee
Birthdate	Birthdate of Employee
Joiningdate	Joining date of Employee
Designation	Current designation of an employee

Table 9.2 : Attributes for Employee entity

Customer	
Attributes	Description
Firstname	Name of the Employee
Customer First Name	Name of customer
Customer Last Name	Surname of customer
Address Line 1	Address of customer
Address Line 2	Street or area
City	City
Pincode	Pincode of area of above city
Email Id	E-mail address of customer
Mobile number	Mobile number of customer
CardHolder	Whether customer is member of Modern Electronic Store

Table 9.3 : Attributes for Customer entity

Product	
Attributes	Description
Product Category	Category of Product like ipad, mobile, smartphone etc.
Model Name	Name of the Model like Samsung Galaxy III
Supplier Name	Name of the Supplier
Reorder Level	Stock level that triggers purchase order
Selling Price	Current selling price of the product
Quantity on Hand	Stock available
Product Image	Image of Product
Camera Pixel Size	Mega pixels of Camera
Processor	Processor name
OS Support	Name of Operating Systems supported by model
Touch Screen	Touch Screen Feature available or not
Wi-fi Enabled	Whether the phone is wi-fi enabled
Memory Slot	Capacity of Initial slot of memory
FM Support	Whether FM support is available or not
Display Size	Display size in inches
HDD Capacity	Hard disk capacity
Weight	Weight of the item

Table 9.4 : Attributes for Product entity

Logical Names and Physical Names of Attributes

When thinking about the name for attributes it is useful to differentiate between a logical name of the attribute and the physical name used in the database application. For example, “First Name” is the name of the fields that stores values of the first name like “Sonal” or “Minal”. “First Name” describes a data entity we are working with. It does not matter if “First Name” is alternatively written “FIRST NAME” or “firstName” or “first_name” or even “FName” as long as we conceptually understand that it refers to the first name.

When we think about conceptual or logical relationships, we need clear and descriptive name. Logical names are used at the time of designing the data model. Unfortunately Logical names do not translate so simply into the names that database software will allow us to use. For example, Base allows us to name a field as “First Name” but other database applications would reject it and ask us to use “first_name” instead. This restriction is due to the way particular database software has been designed. The name actually used in the internal structure of our tables is called a “physical” name, as opposed to the logical name discussed above. This physical name will always have to conform to the conventions imposed by the software that we are using.

Following the conventions mentioned below to name the attributes will facilitate porting of database design to other database software (Base in our case).

1. Start all attribute names with a letter.
2. For subsequent characters use either letters, numbers or the underscore character.
3. Do not use a space between words; instead separate them with the underscore.
4. Do not use special characters except underscore.
5. Use abbreviations, if needed, to keep the length of attribute names short.

We will use camel back notations in database design for our sample application. This means each word in field name starts with capital letter. For example, FirstName, LastName, AddressLine1 etc. Figure 9.6 and 9.7 shown later in the chapter shows field names that would be used for our sample database for Modern Electronic Store.

Step 3 :

Decide what kind of data we want to enter into each attribute. Depending on it we need to decide data type of each field while designing the table. It is compulsory to define the data type of each field. *Data type is a way to define storage structure of the field.* Computers have different ways of storing data. In general, they trade memory or speed for accuracy. Computations that require more accuracy tend to be slower and use more memory. When building tables with Base, we would be presented with a drop-down menu of options asking us to choose the kind of attributes we want to store. The choice we make here will affect the performance of the database.

Figure 9.2, 9.3, 9.4 and 9.5 shows sample records we would like to insert in the tables discussed. Later we will discuss data types provided by Base. Then we will decide which data type we should designate for the attributes of our sample tables.

Supplier Name	Address Line 1	Address Line 2	City	Pincode	Contact Person Name	Email Id	Contact Number
Samsung Corporation	27, Maker Chambers	Nariman point	Mumbai	400021	Mr.Mahesh Wadkar	purchase@samsung.com	66006600
Hewlett-Packard India Sales Pvt. Ltd	24, Salarpuria Areana	Adugodi, Hosur Road	Bangalore	560030	Mr.Venkatesh Rao	placeorder@hp.com	26543123
Sony India	A-31, Mohan Co-operative Indisurial Estate	Mahura Road	Delhi	110001	Mr.Habib Ansari	purchase@sony.com	33344441
Nokia India	Salarpuria Hallmark	Outer Ring Road	Bangalore	560089	Ms.Rita Kulkarni	po@nokia.com	89674210

Figure 9.2 : Sample records of Supplier

Product Category	Model Name	Supplier Name	Reorder Level	Selling Price	Quantity on Hand	Product Image	Camera Pixel Size	Processor	OS Support	Touch Screen	Wi-fi Enabled	Memory Slot (in GB)	FM Support	Display Size (in inches)	HDD Capacity (in GB)	Weight (in gms)
Smartphone	Samsung Galaxy SIII	Samsung Corporation	20	42500	43	/images/sgIII.jpg	8	1.4Ghz Quad Core Processor	Android 4.0	yes	yes	16	yes	4.8		133
Laptop	hp ENVY Sleekbook 6t-1000	Hewlett-Packard India Sales Pvt. Ltd	15	35000	56	/images/hp6t.jpg		Intel Core i5	Windows		yes	16		15.6	500	4.75
Laptop	hp ENVY Sleekbook 6t-1000-L	Hewlett-Packard India Sales Pvt. Ltd	15	30000	56	/images/hp6t.jpg		Intel Core i5	Linux		yes	16		15.6	500	4.75
Mobile	Nokia Lumia 710	Nokia India	30	15499	32	/images/lumina710.jpg	5	Qualcomm MSM8255 (WCDMA)	Windows Phone 7.5	yes	yes	8	yes	3.7		125.5
Camera	DSC-S3000/S	Sony India	35	4490	38	/images/dscs3000.tif	10.1							6.14		11.9

Figure 9.3 : Sample records of Product

First Name	Last Name	Address Line 1	Address Line 2	City	Pincode	Email Id	Mobile Number	Birthdate	Joiningdate	Designation
Anya	Gangele	15-Shub Bungalows	SG Road	Ahmedabad	380067	anya.gangele@gmail.com	99988870	12/12/1981	1/6/2011	Relationship Manager
Robina	Arora	A-4, Sukh Shanti Appartment	Thaltej	Surendranagar	363002	robina.arora@yahoo.com	98707651	6/7/1984	2/7/2006	Receptionist
Akhil	Jain	106, Nirvan Nagar	Army Road	Surat	395001	akhil.jain@yahoo.in	99678901	8/10/1983	1/1/2000	Sales Person
Majid	Khan	B-12, Nishant Appts.	Satellite	Ahmedabad	380058	majid@rediffmail.com	81167453	7/11/1985	1/5/2001	Sales Person

Figure 9.4 : Sample records of Employee

Customer Fname	Customer Lname	Address Line1	Address Line 2	City	Pincode	Email Id	Mobile Number	Card Holder
Puja	Sethia	17,Akash	Jodhpur Char Rasta	Ahmedabad	380056	puja.sethia@yahoo.com	999886662	Yes
Sakina	Nagingar	T-37, Abhimanyu Nagar	Curry Road	Mehsana	384001	sakina82@gmail.com	8111203450	No
Joseph	Thomas	BG-1, Sunstar Complex	Sola Road	Patan	362268	joseph@yahoo.com	9980054321	Yes
Prerna	Agrawal	Smita Bunglows	PNT Road	Ahmedabad	380089	prerna@gmail.com	9004532189	No
Harshit	Shah	17, Jaldeep-1	Bopal	Ahmedabad	380058	harshit@yahoo.com	9012354310	Yes

Figure 9.5 : Sample records of Customer

Data types available in Base

Data types available in Base can be divided into three categories, namely, Alphanumeric type, Calendar type and Binary type.

Alphanumeric Type

When we wish to enter a data that consists of letter, number as well as special character then we can select one of the data types shown in Table 9.5. For example Addresses may contain

alphabets, special characters as well as numbers, thus the appropriate data type assigned to addresses would be one of the alphanumeric type. We can select Text (Varchar) as data type for Address field. The number of bytes to be allocated to Var Char data type depends on the number of characters entered by the user. For example, if address of Harshit contains 100 characters, 100 bytes would be reserved and if it contains 50 characters only 50 bytes would be reserved.

Name	Data type	Max Length	Description
Memo	Long Var Char	2 GB	Stores up to the max length or number indicated by user. It accepts any UTF 8 Character
Text (fix)	Char	2GB	Stores exactly the length specified by user. Pads with trailing spaces for shorter strings. Accepts any UTF 8 Character.
Text	Var Char	2GB	Stores up to the specified length. No padding (Same as long var char)
Text	Var Char Ignore Case	2GB	Stores up the specified length. Comparisons are not case sensitive. It stores capitals as you type them.

Table 9.5 : Alphanumeric Type Attributes : Used for storing alphanumeric characters

Character data type has different nature. It is used when we are aware of exact number of characters to be entered in the field. For example we can use it in case of license number, passport number as they have fixed number of characters. The character data type though has a limitation on number of characters that can be used.

When we want to store some descriptive data, which may have more than 255 characters, then the Memo data type will be useful. This data type allows us to store any text data up to 64,000 characters. Text (Var Char) Ignore case is efficient when the fields are used for searching and it is unsure that whether user will enter searching text in lower case or upper case. For example, First Name or Last name can be stored in Text (Var Char) Ignore case.

Note that computers have different ways of storing alphanumeric characters. For example there is the ASCII code that needs only one byte to store a character. Unfortunately, this limits the number of possible characters that you can use to only 256. Although enough for many applications, it falls short if you want to have access to expanded character sets like Gujarati or Hindi characters. Standards that allow for larger numbers of characters have been developed, like Unicode that use more bytes per character.

Base will store alphanumeric characters using UTF-8, which is a code that is compatible with both ASCII and Unicode. Base will use one or more bytes for each character according to internal calculations. When Base asks you the length for a particular field, e.g. for Surname of Employee, it is not asking the number of Bytes you want to allocate but the number of characters you want to store. How many bytes are actually used is fixed by the software.

Numeric Type

This data type is used to store numerical information like marks of a student, salary of an employee, quantity on hand or balance in the bank account of a person.

We have already seen that in alphanumeric type, the number of bytes actually used is fixed by the software. This is not the case when you store the value for a number. Different ways of storing numbers will require more or less bytes. Numeric data type attributes are described by the number of bytes they use and whether they are signed or unsigned. These two factors determine the range of possible values they can hold. Base offers several types of numeric data attributes, both signed and unsigned that use different amount of bytes.

At the least memory consuming side of number storage, we have the Boolean numbers. A Boolean number is in fact uses just one bit, and we use it to store YES/NO type of data, like answer to the question ‘whether the mobile is Wi-Fi enabled?’, At the other end there are attributes called ‘floating point numbers’ that allow us to store numbers that have decimal places like 1.715249. They are the most memory consuming numbers but the only ones that can perform divisions with good accuracy. Tables 9.6 list different numeric data types along with the number of bits/bytes it uses and its range.

Name	Data type	No. of Bits/Bytes	Signed	Range
Boolean	Yes/No	1 Bit	–	0 – 1
Tinyint	Tiny Integer	1 Byte	No	0 – 255
Smallint	Small Integer	2 Bytes	Yes	-32768 to 32768
Integer	Integer	4 Bytes	Yes	-2.14×10^9 to 2.14×10^9
Bigint	Big integer	8 Bytes	Yes	-2.3×10^{18} to 2.3×10^{18}
Numeric	Number	No limit	Yes	Unlimited
Decimal	Decimal	No limit	Yes	Unlimited
Real	Real	4 Bytes	Yes	5×10^{-324} to 1.79×10^{308}

Table 9.6 : Numeric Type Attributes : Used for storing numeric values

Calendar Type

Another important data type is the Calendar. They are used to store calendar information like year, month, day, hour, minute, second and fraction of a second. Date allows storing year, month and day as it is stored in the system through which data is entered. The same is true for the Time type attribute, which stores the time of the day: hour, minute and second. Finally, some procedures might demand recording of both the time and day of an event. Timestamp has been designed for recording all information at once. In our example database, the fields Joining date, Birth date etc. must be assigned Date data type. Table 9.7 list the name and description of the Calendar data type.

Name	Description
Date	Stores month, day and year information
Time	Stores hour, minute and second information
Timestamp	Stores date and time information

Binary Type

The Binary type attributes allow us to store any information that comes as a long string of zeros and ones. Digitized images and Sounds use this format. They are distinguished by the computer because the initial zeros and ones identify the kind of file they represent (a JPEG image or an MP3 file, etc.). However, Base will make no attempt to identify the kind of file you have stored. This is to say that it won't care if the file is an MP3 or a TIFF and it will happily store it. This in effect means that you could use a Base database to store, for example, photos of the products or employees, or sound snippets or voice messages. Table 9.8 lists the details of the binary data types.

Name	Data type	Max Length	Description
Image	Long Var Binary	2GB	Stores any array of bytes (images, sounds, etc).
Binary	Var Binary	2GB	Stores any array of bytes.
Binary (fix)	Binary	2GB	Stores any array of bytes.

Table 9.8 : Binary Type attributes: Used for storing images and sounds

Now let us assign data types as per our requirement of input data. Figure 9.6 and figure 9.7 shows data types assigned to each field of Supplier, Product, Employee and Customer table.

Supplier		Employee	
Field Name	Data type	Field Name	Data type
SupplierName	Text [VARCHAR_IGNORECASE]	FirstName	Text [VARCHAR_IGNORECASE]
AddressLine1	Text [VARCHAR_IGNORECASE]	LastName	Text [VARCHAR_IGNORECASE]
AddressLine2	Text [VARCHAR_IGNORECASE]	AddressLine1	Text [VARCHAR_IGNORECASE]
City	Text [VARCHAR_IGNORECASE]	AddressLine2	Text [VARCHAR_IGNORECASE]
Pincode	Text fix [CHAR]	City	Text [VARCHAR_IGNORECASE]
ContactPersonName	Text [VARCHAR_IGNORECASE]	Pincode	Text fix [CHAR]
EmailId	Text [VARCHAR]	EmailId	Text [VARCHAR]
ContactNumber	Integer	MobileNumber	Text [VARCHAR]
		BirthDate	Date
		JoiningDate	Date
		Designation	Text [VARCHAR_IGNORECASE]

Figure 9.6 : Data types assigned to fields of Supplier and Employee table

Product		Customer	
Field Name	Data type	Field Name	Data type
ProductCategory	Text [VARCHAR_IGNORECASE]	CustomerFname	Text [VARCHAR_IGNORECASE]
ModelName	Text [VARCHAR_IGNORECASE]	CustomerLname	Text [VARCHAR_IGNORECASE]
SupplierName	Text [VARCHAR_IGNORECASE]	AddressLine1	Text [VARCHAR_IGNORECASE]
ReorderLevel	SmallInt	AddressLine2	Text [VARCHAR_IGNORECASE]
SellingPrice	Integer	City	Text [VARCHAR_IGNORECASE]
QuantityonHand	SmallInt	Pincode	Text [VARCHAR]
ProductImage	Text [VARCHAR]	EmailId	Text [VARCHAR_IGNORECASE]
CameraPixelSize	Integer	MobileNumber	Text [VARCHAR]
Processor	Text [VARCHAR_IGNORECASE]	Card Holder	Boolean
OSSupport	Text [VARCHAR_IGNORECASE]		
TouchScreen	Boolean		
Wi-fiEnabled	Boolean		
MemorySlotInGB	SmallInt		
FMSupport	Boolean		
DisplaySizeInInches	SmallInt		
HDDCapacity	SmallInt		
WeightInGrams	SmallInt		

Figure 9.7 : Data types assigned to fields of Product and Customer table

Note : When we know that field will contain only numbers but, we don't expect to perform any calculations or sorting with the values in the field, it is better to use the Text data type. Hence Pincode and Mobile number have been assigned Text data type.

Here you may think that we can use text data type for the field DisplaySizeInInches defined in Product table in our sample database. Base will allow to insert data like 1, 12, 457 etc. But when we perform sorting (discussed in later chapter) on the records according to Display Size, we will get the result in the reverse numeric order: 457, 12, 1. The reason is when Base sorts numbers in a Text field, it reads the numbers from left to right instead of right to left. Thus while sorting Base reads the right most digit (7 in case of 457) and sorts accordingly.

Step 4 :

Decide a key field of the table designed. The field that has unique importance in the table is known as Key Field.

What is the purpose of model name given to each product manufactured by a company? It helps us distinguish a product from other. You may visit Modern Electronic Store and can directly ask him to show you Samsung Galaxy III. You will not find any confusion on shop keeper's face. On the other hand the Employee table may contain more than one employee with same first names. How could they be identified? We may query the database to return details of Anya, but database may return with multiple records for Anya. We may get confused seeing the result thinking which Anya's details we were looking for? Thus each table should contain at least one field which can identify each record of a table uniquely. For example, a unique employee code may be given to each employee. Hence we may need to add this new field as an attribute in the Employee table.

In terms of database, a field or combination of fields capable of identifying each record uniquely is known as Primary key. You may think even driving license number or passport number can serve as Primary key but then there would be multiple records with null value. It is not always necessary that an employee when recruited has driving license or passport. Have a look at the sample records of Product table shown in figure 9.3 again. You must have observed that we have not entered any value in HDD Capacity of Smart phone. There can be two reasons for this, either we do not know HDD Capacity of Smartphone or that particular attribute may or may not contain any value. Such type of values is known as Null value. A null value means unknown or missing value. *Primary Key values cannot contain Duplicate values or Null values.*

Figure 9.6 and 9.7 shows four tables Supplier, Product, Employee and Customer. As can be seen the Supplier table does not have any field that contains a unique value. Thus we shall need to add an extra field **Scode** in the Supplier table so that each supplier record can be uniquely identified. We also need to decide the structure of Scode. Let us assume it to be a field that has four characters. The first character has to be alphabet 'S' and the other three characters would represent a number. Thus suppliers of the Modern Electronic Store could be identified as S001, S002, S003 and so on. Similarly, an attribute **Pcode** should be added in Product table, **Ecode** in Employee table and **Ccode** in Customer table.

You may observe that Product table consists of field Model Name which can also act as Primary key in the Product table. However, you may notice that model name consists of variable characters with approximately more than 15 characters for each Product. If we designate only 8 characters for the product code, we can identify 9999999 products uniquely (first character would be P). Primary key of a table is also used to establish relationships with multiple tables in the database. We will discuss how to establish relation between tables in next chapter.

Defining a primary key is must for each table. A table though cannot have more than one primary key. However combination of more than one field can serve as primary key. A primary key so designed is known as Composite primary key. Before discussing the design of other tables to be used in our sample application, let us first design the table already discussed using Base.

Opening Base

Let us first start Base. Select Applications → Office → OpenOffice.org 3.2 Base. When you start the Base software, you will see some options available on the Base opening screen as shown in figure 9.8.



Figure 9.8 : Opening Screen of Base

If we want to work with any of the database created earlier then we can open such databases by choosing an option *Open an existing database file*. Here, in our case, we have not created any database so far. Hence we will choose the option *Create a new database*. By doing so, Base will display a screen as shown in figure 9.9.

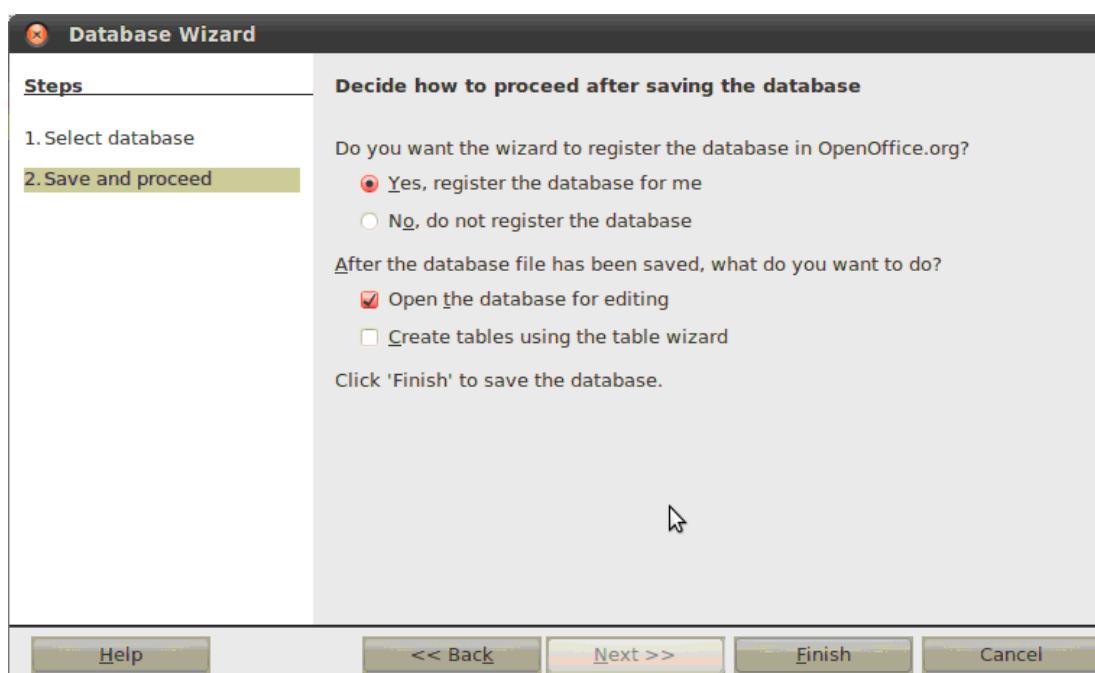


Figure 9.9 : Opening the Database

In figure 9.9, you may observe that we get an option to register our database in OpenOffice.org web site. We may select this option in case we want to make the database public. Normally we would not make the database public, hence select the second option “*No, do not register the database*”. Further Base gives us two options to start working with database. We can either create tables using Design view or can use Wizard. By default, *Open the database for editing* option would be checked. If we want to use wizard, Select the check box with option *Create tables using the table wizard*. Let us create the tables using wizard. Select the option *Create tables using the table wizard* and click on Finish button. The Save As dialog box shown in figure 9.10 will now open.

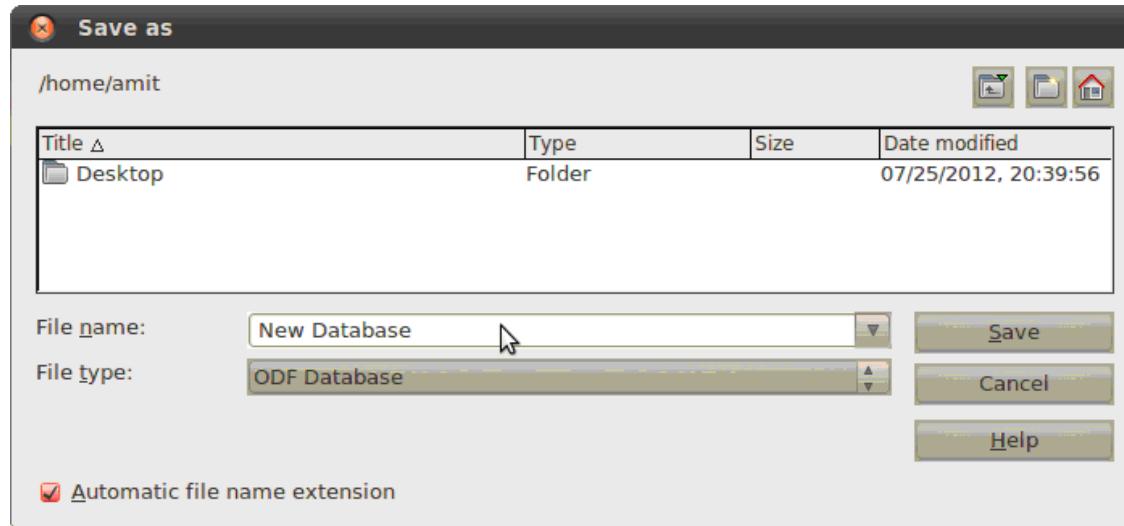


Figure 9.10 : Naming Database File

In textbox with label *File name:* where text *New Database* is visible type ModernElectronicStore and click Save button. Choose appropriate location to save the file by clicking the drop down arrow. Base will automatically assign .odb extension to the database file. Click Save Button and Table Wizard as shown in figure 9.11 will be displayed.

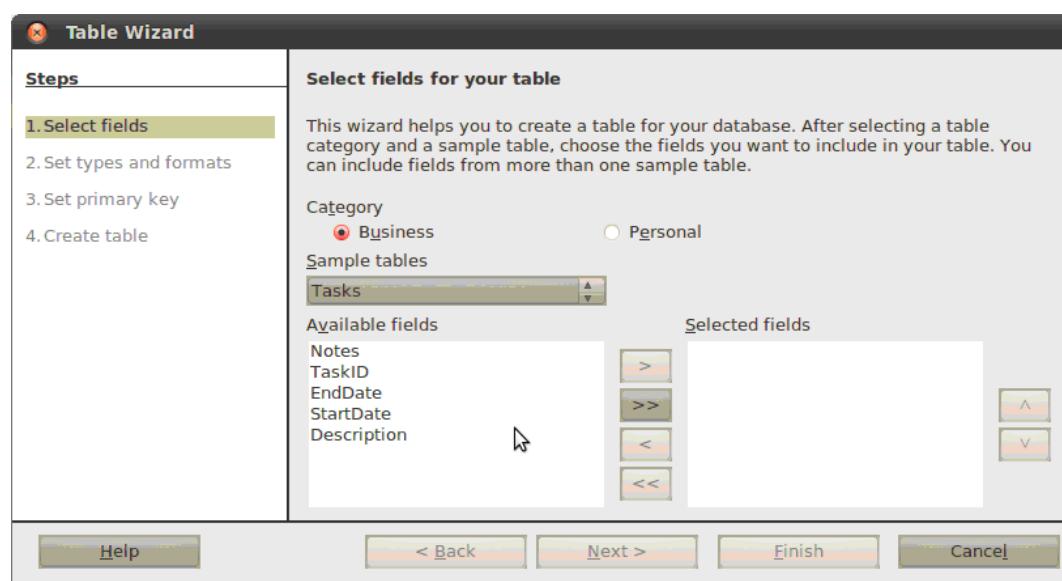


Figure 9.11 : Table creation using Wizard

We will first use Table Wizard of Base software which consists of readymade tables divided into two categories, viz., Business and Personal.

Business Category consists of sample tables like Tasks, Assets, Contacts, Expenses, Invoices etc. Personal Category consists of tables like Photographs, DVDCollection, Recipes, Plants etc. Select Business category and click on drop down list available under *Sample tables* list. Observe the list and you will find that all the four tables we intend to create are available in the list. Select Supplier. List of fields as shown in figure 9.12 will be populated.

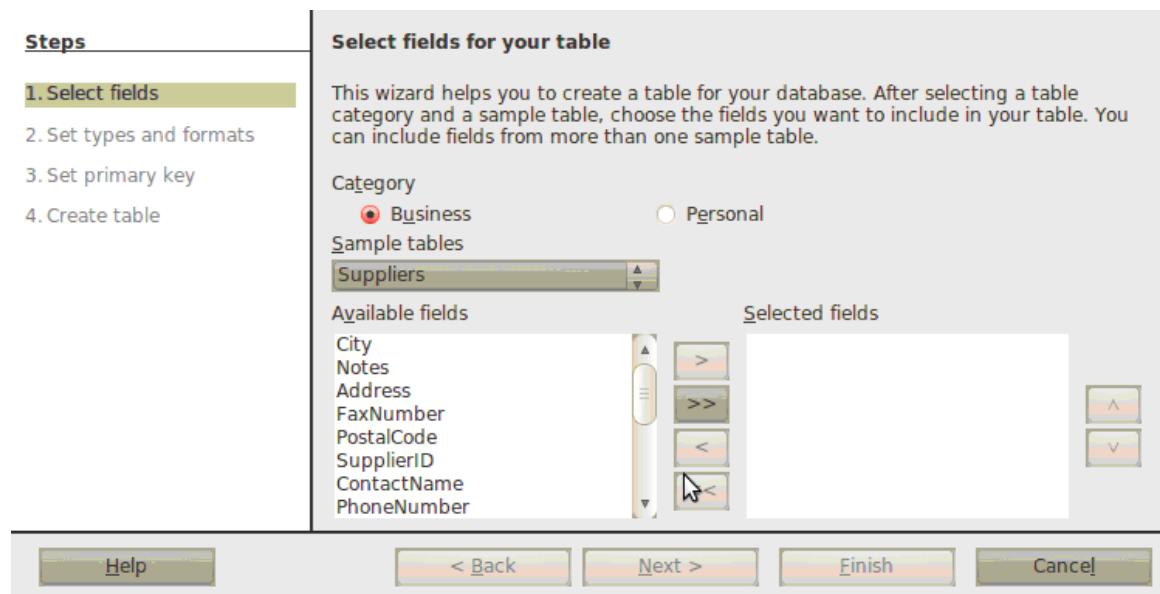


Figure 9.12 : Creating Suppliers table

Click on the  button to select all the fields that are visible under *Available fields* list. Alternatively you can hold control key and select only the desired fields one by one. Once you are done, click on  button. Time being we will click on  button so that all the fields are part of our table.

Click on Finish button, the Suppliers table will be opened in Data View. Opening table in Data View allows user to enter records in the table. Figure 9.13 shows data view of Supplier table.

The screenshot shows a window titled "Suppliers - ModernElectronicStore - OpenOffice.org Table Data View". The menu bar includes File, Edit, View, Insert, Tools, Window, and Help. Below the menu is a toolbar with icons for new, open, save, print, and various filters. The main area displays a table with columns: ID, City, Notes, Address, FaxNumber, PostalCode, SupplierID, and ContactName. The first row contains data: ID is empty, City is "London", Notes is empty, Address is "10 Downing Street", FaxNumber is empty, PostalCode is "SW1A 2AA", SupplierID is empty, and ContactName is empty. The second row is empty. At the bottom, there is a navigation bar with buttons for Record, Page, and other database operations.

Figure 9.13 : Supplier Table in Data View

Base opens each table in a separate window. Close the Suppliers Data View window. Open ModernElectronicStore.odb window and you will find main Database window as shown in figure 9.14.

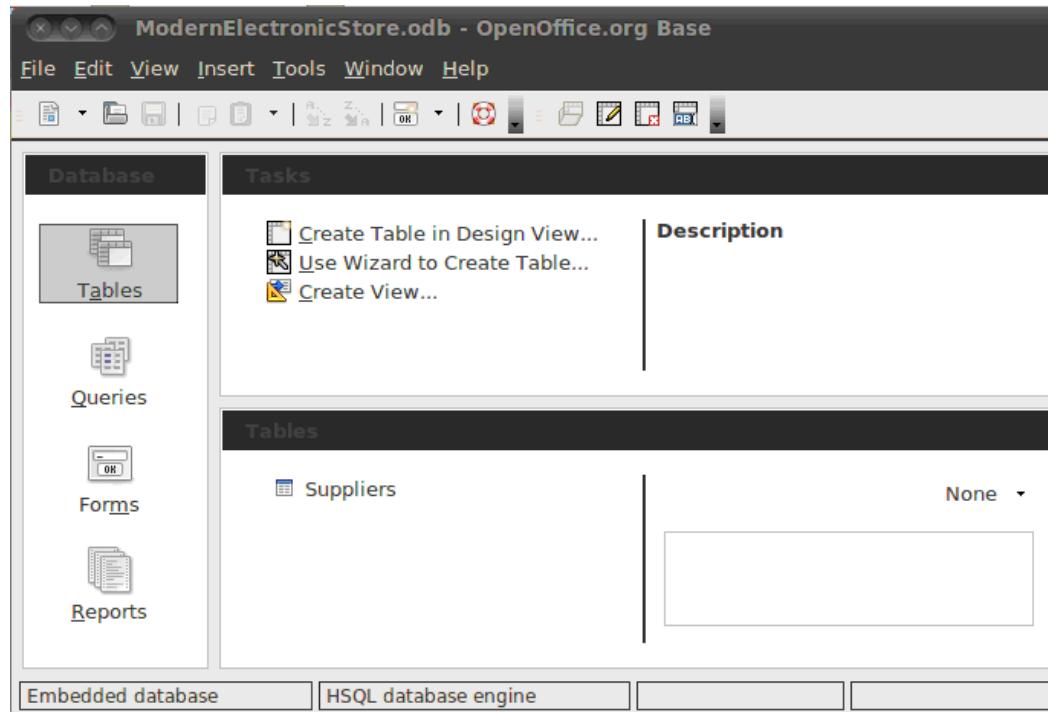


Figure 9.14 : Database Window

In the left pane, you can see Database Objects, viz., Tables, Queries, Forms and Reports. The Tables object icon would be a default selection and you can see that Suppliers table is visible under the label *Tables*. Let us view details of fields of Suppliers table. Right click on Suppliers table icon. Choose Edit option from the available options (see figure 9.15).

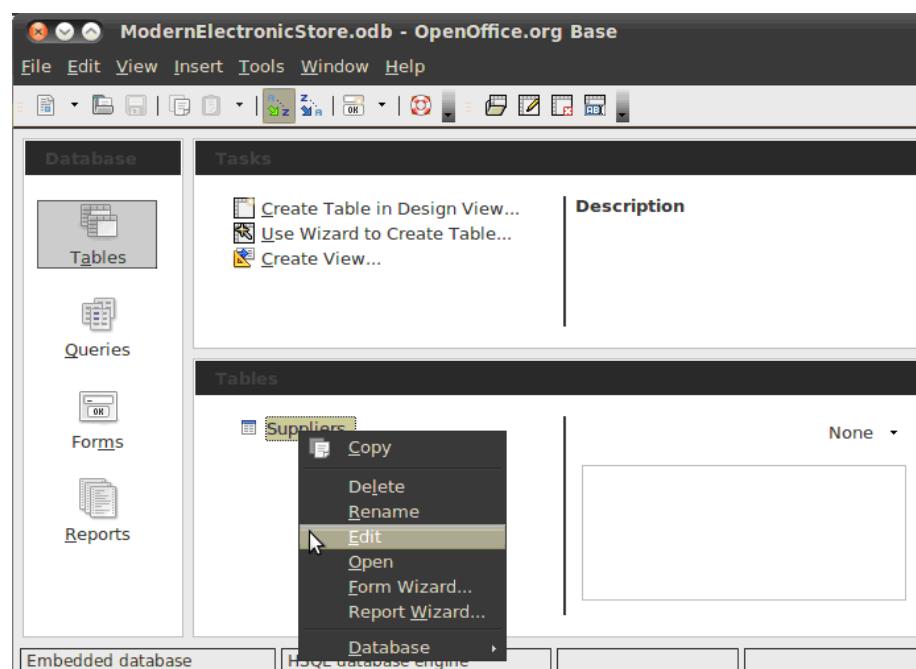


Figure 9.15 : Editing Suppliers Table

Suppliers table is now opened in Design View as shown in figure 9.16. We can now edit, update or delete the fields of the table.

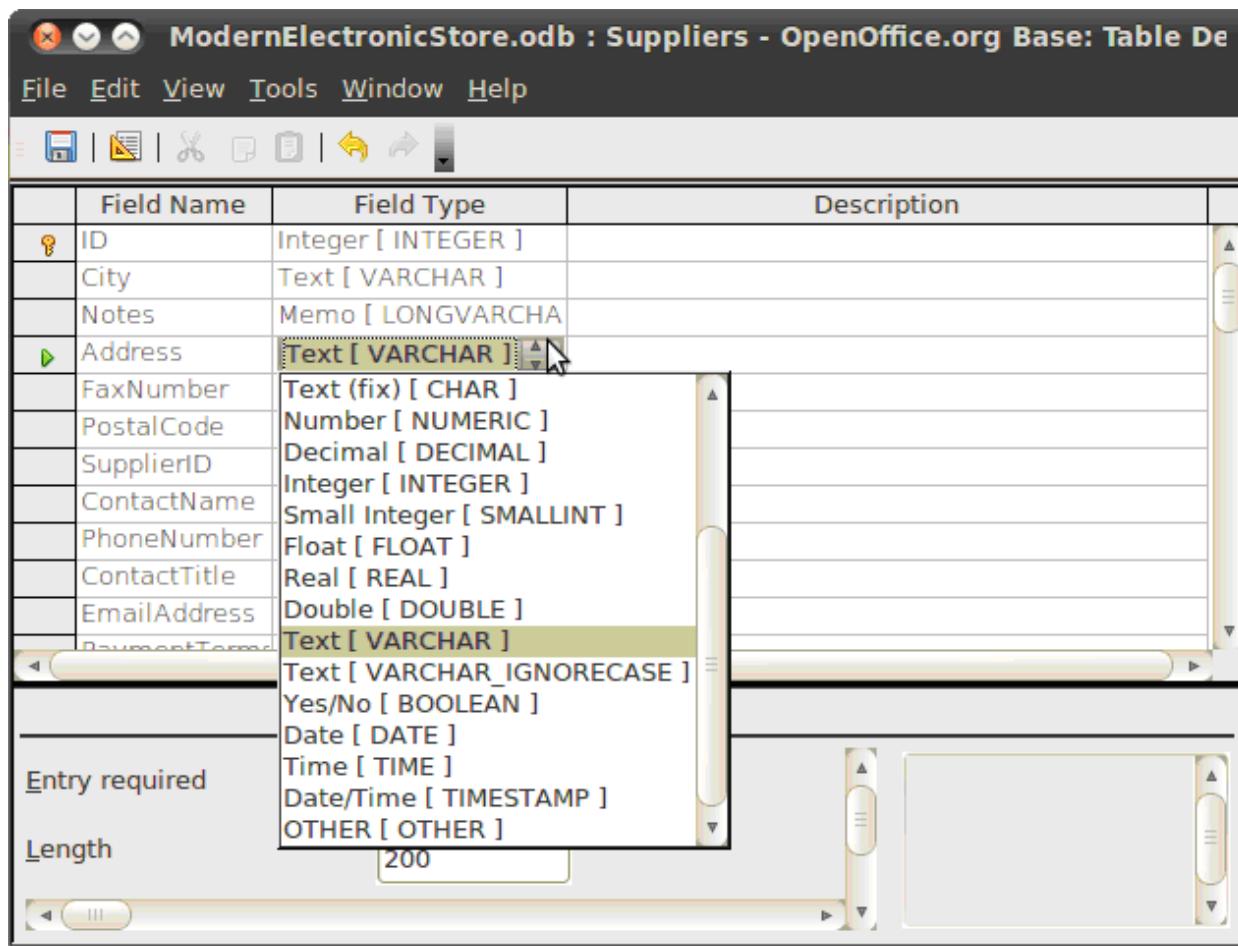


Figure 9.16 : Choosing Data type for a field

In this view defining each field is associated with four terms as stated below:

1. Field Name
2. **Field Type :** It allows us to assign a data type to each field.
3. **Description :** It is good practice to enter description for each of the field, though it is not compulsory to write description for each field of the table. The description helps the user to understand what the purpose of each field is.
4. **Field Properties :** It is used to control and validate the data that is to be entered. For each data type selected, a list of field properties related to it is displayed. It also has some default selection. We can change all the field properties as per our requirement. Various field properties will be discussed in detail in later chapter.

Earlier in this chapter we have discussed that we will be using Text[VARCHAR_IGNORECASE] data type for Address field. Click on the drop down arrow in front of Address field in *Field Type*

column, data types as shown in figure 9.16 will be displayed. Choose Text [VARCHAR_IGNORECASE] data type. You can also rename the Address field to AddressLine1 and update its data type. The Products, Employees and Customers table can be created and modified as per our requirement in the similar manner.

Deleting Objects

To delete any object created in Base, right-click the object icon and click on Delete option. Delete all the tables including the Suppliers table created through the wizard. We will recreate the required tables as per the specifications given in figure 9.6 and 9.7 using Design View.

Note : While creating the tables in Design view we will add Scode, Pcode, Ecode and Ccode field in the Supplier, Product, Employee and Customer table respectively.

Creating Table using Design View

You might have observed that when we use wizard to create a table we need to make certain changes to the pre-designed tables so that it can be readily used. In real scenario the tables provided by Base would not suffice. **Hence creating tables using Design View is more beneficial.**

Let us create table using Design View. In main Database window, under Tasks pane, two options for creation of table, *Create Table in Design View....* and *Use Wizard to Create Table...* as shown in figure 9.17 will be available. Click on the option *Create Table in Design View* and Table Design Window will be opened.



Figure 9.17 : Tasks Pane in Database Window

Table Design View as shown in figure 9.18 is grid structure with three columns, Field Name, Field Type and Description and Field Properties pane at the bottom half of the window. Create a Supplier table discussed figure 9.6. Type name of the field and select data type corresponding to each field also make sure to add Scode as the first field.

	Field Name	Field Type	Description
	Scode	Text [VARCHAR]	
	SupplierName	Text [VARCHAR_IGNORECASE]	
	AddressLine1	Text [VARCHAR_IGNORECASE]	
	AddressLine2	Text [VARCHAR_IGNORECASE]	
	City	Text [VARCHAR_IGNORECASE]	
	Pincode	Text (fix) [CHAR]	
	ContactPersonName	Text [VARCHAR_IGNORECASE]	
	EmailId	Text [VARCHAR]	
▶	ContactNumber	Integer [INTEGER]	

Figure 9.18 : Table Design View Window

Set Primary key

To designate a field as Primary key, right click on the Scode field to open a sub-menu as shown in figure 9.19. Select the *Primary Key* option and a symbol which looks like a key (🔑) will be displayed in first column against the chosen field.

	Field Name	Field Type	Description
🔑	Scode	Text [VARCHAR]	
	Cut	Text [VARCHAR_IGNORECASE]	
	Copy	Text [VARCHAR_IGNORECASE]	
	Delete	Text [VARCHAR_IGNORECASE]	
	Insert Rows	Text [VARCHAR_IGNORECASE]	
✓	Primary Key	Text (fix) [CHAR]	
	ContactPersonName	Text [VARCHAR_IGNORECASE]	
	EmailId	Text [VARCHAR]	
	ContactNumber	Integer [INTEGER]	

Figure 9.19 : Designating Primary key of a Table

Save Table

Having created the table structure, it's now time to save the table. Click on Save button (--) visible in figure 9.19. The Save dialog box as seen in figure 9.20 will be displayed. Type the name of the table and click on OK button.

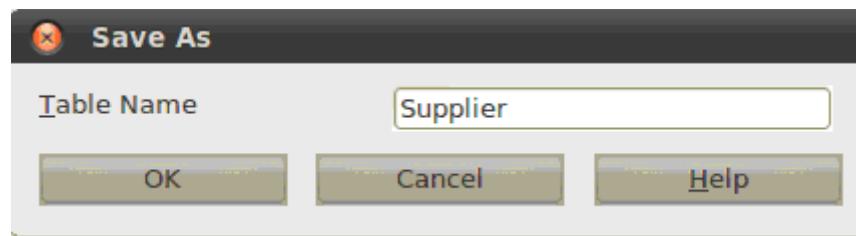


Figure 9.20 : Saving Table

Similarly create the tables Product, Employee and Customer. The tables created by us till now are known as **Master Tables**. These tables store data that has information of transactions like, customer purchasing a mobile phone from Modern Electronic Store or details of purchase made by Mobile Electronic Store from its supplier Sony India. It is suitable to design separate tables to store transactions and relate them with master tables. We will discuss transaction tables in later chapter.

Summary

In this chapter we have discussed about importance of data, information, database and Database Management System (DBMS). A raw fact generally refers to data, data when processed becomes information, and properly arranged information is stored in database. The DBMS includes different objects like table, query, report and form. We have also discussed the steps required to design the database. Further we saw the usage of OpenOffice Base software to create tables making use of Wizard and Design view. In the next chapter we will learn how to relate tables and perform operations on table records using Base.

EXERCISE

1. Explain the terms Data and Information with example.
2. Explain the need of information.
3. Define the terms database, data model and Database Management Systems.
4. Define Table, Record and Field with example and diagram.
5. What are logical field names and physical field names?
6. Write down instructions to be followed while deciding a field name.
7. Write down the steps to design the database.
8. What is the difference between Text (fix)[CHAR] and Text[VARCHAR]? Explain giving suitable example.

- 9.** Define Key field.
 - 10.** What is primary key? Explain giving a suitable example?
 - 11.** What are the properties of primary key?
 - 12.** What is Data type? List and explain the data types available in Base.
 - 13.** When should one use memo data type?
 - 14.** List the objects available in Base.
 - 15.** What is Null value? Explain giving a suitable example.
 - 16.** Choose the most appropriate option from those given below :
- (1) DBMS Stands for
- (a) Database Management System (b) Database Migration System
(c) Data Management System (d) Direct Base Management System
- (2) Which of following is a term used for “Raw facts or figures”?
- (a) Information (b) Variable
(c) Data (d) Field
- (3) Which of the following best describes customer in Database?
- (a) Relationship (b) Attribute
(c) Entity (d) Data
- (4) Processed data is known as
- (a) Fact (b) Prepared data
(c) Information (d) Decision
- (5) Which one of the following is not a database?
- (a) MySQL (b) Base
(c) SQL Server (d) SpreadSheet
- (6) Which of the following is not an Open source DBMS?
- (a) MySQL (b) Ingress
(c) Base (d) Oracle
- (7) The alphabet R in RDBMS stands for which of the following?
- (a) Rotational (b) Relational
(c) Random (d) Ring
- (8) What is the name of the database available as part of Open Office Suite?
- (a) OfficeDB (b) Base
(c) OpenDB (d) Access
- (9) Which of the following is the default extension of Base Database file?
- (a) .bdf (b) .odf
(c) .odb (d) .dbf

LABORATORY EXERCISES

1. Figure 9.21 shows the fields and their description that can be used in Student, Teacher and Subject tables. Decide data types for each field and create the tables using Base. Enter at least five records in it.

Student	
Field Name	Description
Gmo	General Register Number
Firstname	Name of the Student
Surname	Surname of the Student
Address	Address of the Student
City	City
Pincode	Pincode
Birthdate	Date of Birth
Gender	Male or Female
Standard	Studying in which standard
Join_date	Date of Joining School
Leaving_date	Date of Leaving School

Teacher	
Field Name	Description
Firstname	Name of the Teacher
Surname	Surname of the Teacher
Address	Address of the Teacher
City	City
Pincode	Pincode
Phone_no	Phone number of teacher
Email_id	E-mail id of teacher
Mobile_no	Mobile number of teacher

Subject	
Field Name	Description
Sub_Name	Name of the Subject
Details	Description of the subject

Figure 9.21 : Tables for School Management System

2. Create the transaction tables shown in figure 9.22 in Base. Enter at least five records in it.

Standard_Subj : To store details about subjects taught in each standard	Subj_Teacher: To store details about which teacher is teaching the subject in each standard																																	
<table border="1"> <thead> <tr> <th>Field Name</th> <th>Data Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Standard</td> <td>Number</td> <td>Which Standard</td> </tr> <tr> <td>Scode</td> <td>Text</td> <td>Which Subject</td> </tr> </tbody> </table>	Field Name	Data Type	Description	Standard	Number	Which Standard	Scode	Text	Which Subject	<table border="1"> <thead> <tr> <th>Field Name</th> <th>Data Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Year</td> <td>Number</td> <td>Current Year</td> </tr> <tr> <td>Standard</td> <td>Number</td> <td>In which Standard</td> </tr> <tr> <td>Scode</td> <td>Text</td> <td>Subject Code</td> </tr> <tr> <td>Tcode</td> <td>Text</td> <td>Teacher Code</td> </tr> </tbody> </table>	Field Name	Data Type	Description	Year	Number	Current Year	Standard	Number	In which Standard	Scode	Text	Subject Code	Tcode	Text	Teacher Code									
Field Name	Data Type	Description																																
Standard	Number	Which Standard																																
Scode	Text	Which Subject																																
Field Name	Data Type	Description																																
Year	Number	Current Year																																
Standard	Number	In which Standard																																
Scode	Text	Subject Code																																
Tcode	Text	Teacher Code																																
Attendance: To store daily attendance of the Students	Result: To store result details of each student																																	
<table border="1"> <thead> <tr> <th>Field Name</th> <th>Data Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Gmo</td> <td>Number</td> <td>General Register of a Student</td> </tr> <tr> <td>Date</td> <td>Date/Time</td> <td>On which Date</td> </tr> <tr> <td>Pr_ab</td> <td>Yes/No</td> <td>Present or Absent</td> </tr> </tbody> </table>	Field Name	Data Type	Description	Gmo	Number	General Register of a Student	Date	Date/Time	On which Date	Pr_ab	Yes/No	Present or Absent	<table border="1"> <thead> <tr> <th>Field Name</th> <th>Data Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Gmo</td> <td>Number</td> <td>General Register No.</td> </tr> <tr> <td>Scode</td> <td>Text</td> <td>Of which Subject</td> </tr> <tr> <td>Month</td> <td>Text</td> <td>In which Month</td> </tr> <tr> <td>Year</td> <td>Number</td> <td>Which Year</td> </tr> <tr> <td>Term</td> <td>Number</td> <td>Name of the Term</td> </tr> <tr> <td>Marks</td> <td>Number</td> <td>Marks scored</td> </tr> </tbody> </table>	Field Name	Data Type	Description	Gmo	Number	General Register No.	Scode	Text	Of which Subject	Month	Text	In which Month	Year	Number	Which Year	Term	Number	Name of the Term	Marks	Number	Marks scored
Field Name	Data Type	Description																																
Gmo	Number	General Register of a Student																																
Date	Date/Time	On which Date																																
Pr_ab	Yes/No	Present or Absent																																
Field Name	Data Type	Description																																
Gmo	Number	General Register No.																																
Scode	Text	Of which Subject																																
Month	Text	In which Month																																
Year	Number	Which Year																																
Term	Number	Name of the Term																																
Marks	Number	Marks scored																																
Fees: To store details of Fee amount paid by students	Sports: To keep record of Interest of students in Sports																																	
<table border="1"> <thead> <tr> <th>Field Name</th> <th>Data Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Gmo</td> <td>Number</td> <td>General Register Number</td> </tr> <tr> <td>Date</td> <td>Date/Time</td> <td>On which date</td> </tr> <tr> <td>Tuition_fee</td> <td>Currency</td> <td>Tuition fee paid by the student</td> </tr> <tr> <td>Term_fee</td> <td>Currency</td> <td>Term fee paid by the student</td> </tr> </tbody> </table>	Field Name	Data Type	Description	Gmo	Number	General Register Number	Date	Date/Time	On which date	Tuition_fee	Currency	Tuition fee paid by the student	Term_fee	Currency	Term fee paid by the student	<table border="1"> <thead> <tr> <th>Field Name</th> <th>Data Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Gmo</td> <td>Number</td> <td>General Register No.</td> </tr> <tr> <td>Interest</td> <td>Text</td> <td>Interested in which activity</td> </tr> <tr> <td>Achievement</td> <td>Memo</td> <td>Any achievements?</td> </tr> </tbody> </table>	Field Name	Data Type	Description	Gmo	Number	General Register No.	Interest	Text	Interested in which activity	Achievement	Memo	Any achievements?						
Field Name	Data Type	Description																																
Gmo	Number	General Register Number																																
Date	Date/Time	On which date																																
Tuition_fee	Currency	Tuition fee paid by the student																																
Term_fee	Currency	Term fee paid by the student																																
Field Name	Data Type	Description																																
Gmo	Number	General Register No.																																
Interest	Text	Interested in which activity																																
Achievement	Memo	Any achievements?																																
Cultural: To keep record of Interest of students in Cultural Activities																																		
<table border="1"> <thead> <tr> <th>Field Name</th> <th>Data Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Gmo</td> <td>Number</td> <td>General Register Number</td> </tr> <tr> <td>Interest</td> <td>Text</td> <td>Interested in which activity</td> </tr> <tr> <td>Achievement</td> <td>Memo</td> <td>Any achievements?</td> </tr> </tbody> </table>	Field Name	Data Type	Description	Gmo	Number	General Register Number	Interest	Text	Interested in which activity	Achievement	Memo	Any achievements?																						
Field Name	Data Type	Description																																
Gmo	Number	General Register Number																																
Interest	Text	Interested in which activity																																
Achievement	Memo	Any achievements?																																

Figure 9.22 : Tables for School Management System

10 Working with Tables



As discussed in the previous chapter, we all have to take many decisions daily and for that we need one or the other type of information. To get the correct information at right time, we have to prepare database and perform various operations like inserting data, editing the previously entered data, delete unnecessary and wrong data or arrange the data as per requirement. In previous chapter, we have already discussed how to create database and various tables in Base. Now, let us learn how to do various operations on the tables created by us.

Inserting data in the table

To insert records into the table, we must open the table first. To open the table, double click on the icon that has the required table name in Tables Pane of Database Window. Alternatively right click on the desired table and select the *Open* option from the sub menu visible.

Let us open the Supplier table we had created using the Design View. When we open the Supplier table, its structure will be displayed in the Datasheet View as shown in figure 10.1.

The screenshot shows a software interface for managing a database table named 'Supplier'. The window title is 'Supplier'. The menu bar includes 'File', 'Edit', 'View', 'Insert', 'Tools', 'Window', and 'Help'. Below the menu is a toolbar with various icons. The main area displays a table structure with columns: Scode, SupplierName, AddressLine1, AddressLine2, City, Pincode, ContactPersonName, EmailId, and ContactNumber. A single row is visible, representing a record. The first cell contains a green arrow icon pointing right, which is the 'Record Selector Icon'. The status bar at the bottom indicates 'Record 1 of 1'.

Scode	SupplierName	AddressLine1	AddressLine2	City	Pincode	ContactPersonName	EmailId	ContactNumber

Figure 10.1 : Initial data entry screen

Here, we can see that the screen displays the field names in horizontal line. The line consisting of field names is known as Title line. We can see that it can show only some limited fields that can fit on the screen. To see the remaining fields, we have to use the scroll bar given at the bottom of the screen. Below the Title Line, there is a row consisting of empty boxes. The job of filling all these empty boxes with appropriate data is known as Data Entry. In front of the first field there is a small box that contains a pointing arrow sign. This box is known as ‘Record Selector Box’ and the sign in it is referred as ‘Record Selector Icon’. Some of the Record Selector Icons and their functions are as mentioned below:

The black pointing arrow icon is known as the **record pointer** that indicates a current record (current row) of the table at any given moment of time. This means that if we start typing, changes in the field value will appear in a current record.

The green icon with flash is the end of the table mark. It is displayed in the next to the last record in the table. To add new record in Datasheet View, scroll to the last row, then click. By

doing so, our cursor will be positioned in the field and the icon will be changed to the black pointing arrow (➡).

The pencil icon (>Edit) is an Edit Icon. When we click on any of the field to edit the data, the black pointing arrow icon will be replaced with this pencil like icon. This icon will be displayed, when we have made changes to a record that have not yet been saved. If we move the cursor to another record, then correction will be saved, and if we press ESC key then the correction will be cancelled and the original contents will be restored.

Observe the bar at the lower left portion of figure 10.2, there is a word ‘Record’, then a numeric value ‘4’ in the box and at the end word ‘of 4’. This bar is known as **Navigation Bar**. It contains some navigation buttons to scroll the records vertically. It also lets us know the current position of the record pointer. The meaning of ‘Record 4 of 4’ is that we have total 4 records in our table and at present our cursor is on the fourth record.

Let us fill various fields with the relevant data discussed in previous chapter. The data entry in Base is controlled and restricted based on the data type and field property that we have defined for each field. Observe that after entering data in the last field of the current row, the cursor automatically comes to the first field *Scode* of the next row.

In case of the Boolean type field a square box will be shown in the field. If we click on it, it will be marked with a tick sign. If the box is marked with tick mark sign it means that the value of the field is true, otherwise false. After entering some records our screen may look somewhat as shown in figure 10.2.

Scode	SupplierName	AddressLine1	AddressLine2	City	PinCode	ContactPersonName	EmailID	ContactNumber
S01	Samsung Corporation	27, Maker Chambers	Nariman Point	Mumbai	400021	Mr. Mahesh Wadkar	purchase@sam	66006600
S02	Hewlett-Packard India	24, Salarpuria Arena	Adugodi, Hosur Rd	Hosur	560030	Mr.Venkatesh Rao	placeorder@hp	26543123
S03	Sony India	A-31, Mohan Co-oper	Mathura Road	Mathura	110044	Mr.Habib Ansari	purchase@sony	33344441
S04	Nokia India	Salarpuria Hallmark	Outer Ring Road	Gurgaon	560089	Ms.Rita Kulkarni	po@nokia.com	89674210

Record 4 of 4

Current Record Total Records

New Record
Last Record
Next Record
Previous Record
First Record

Figure 10.2 : Datasheet view after inserting records

Editing Records in the Table

The data once entered may have to be edited for one of the following two reasons:

1. The data entered during data entry is incorrect.
2. After correct data entry, there is a change in the value of the data; in this case we have to edit our record. For example, assume that we have entered the correct address of

a particular Supplier. After sometime the supplier shifts his company location. In this case though the address of that Supplier was entered correctly, we will have to edit it due to the change in location.

The process of correcting the data entered previously is known as **Editing**. To do edit operation, we have to open the table and simply place the cursor on the field value that we want to edit and make the desired correction.

Deleting Records from the Table

To keep our database precise and meaningful, it is very much necessary to delete unnecessary or incorrect records from the table. Doing this will provide a clear picture and at the same time it will free some disk space. To delete any record from the table, open the table and select the unwanted record. After selecting the record, we can delete it using following two ways:

1. By pressing DEL key from the keyboard or selecting *Delete* option from Edit menu.
2. By right clicking the selected record and choosing *Delete Record* option from the sub menu.

When you delete the record, Base will display a dialog box with a warning message as shown in figure 10.3.

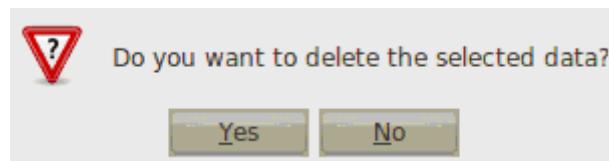


Figure 10.3 : Record Deletion Alert Box

From this dialog box, if we click on the Yes button, then the selected record will be deleted from the table permanently. But if we click on No button then the record will not be deleted from the table and it will be redisplayed in the view.

Sorting Data in the Table

We create and maintain the database, because our final aim is to find the information as and when needed. Gradually the records in table will increase hence to get the information easily and speedily, it is desirable that the data in the table is arranged in some particular order. A Product table may contain thousands of records related to different products. Assume that we want to find the details of a particular product along with its price. Now in this case, if the product table is arranged in order of field Pcode i.e product code, it becomes very easy for us to find it. But, suppose we do not know the product code, then....!! Definitely, we can't find the record of that product easily and speedily. But if we have arranged the Product table in order of PCode and SellingPrice fields then it will be easy to get the desired record. So, let us see how to sort the table. Table can be sorted in one of the two ways as mentioned below:

1. Select on the field that we want to sort and then click on either Sort Ascending button  or Sort Descending button  from the toolbar, as per your needs.
2. Alternatively click on Sort button  and a dialog-box as shown in the figure 10.4 will be displayed.

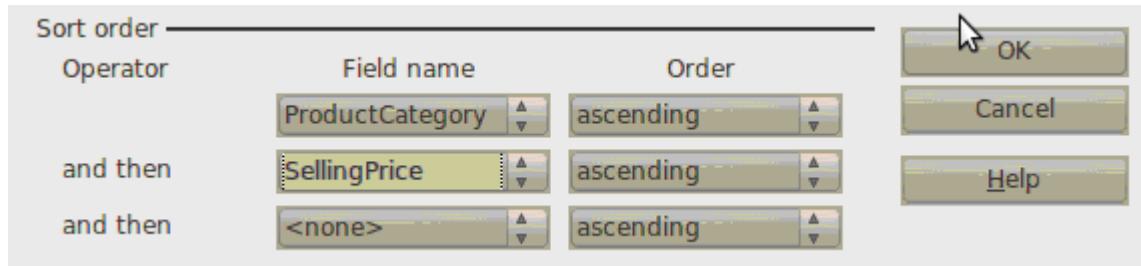


Figure 10.4 : Choice of Sort Order

Select appropriate field value under the *Field name* dropdown box and choose required order of sorting under the *Order* dropdown box. In our case, the Product table will be sorted based on ProductCategory in ascending order first and then the data will again be sorted in ascending order of the SellingPrice field. For example all Laptops would precede all Mobiles. Within Laptops and Mobiles, the records would be arranged according to their selling price with product with lesser price listed first.

Controlling Data Redundancy

By now, you might have become quite familiar with Base. So let us continue our discussion of designing database for our sample application. We have already designed four tables, Supplier, Product, Employee and Customer. Now let us have a relook at the table design from the point of view of data redundancy. *Data redundancy is a term used for unnecessary duplication or repetition of data.*

We have seen that the Customer table has a field called City. Modern Electronic Store would have many customers that belong to the same city. Thus if 100 customers belong to city called ‘Ahmedabad’, then for each of these customers the string ‘Ahmedabad’ would be repeated. Data redundancy thus leads to wastage of storage space and is inefficient for several reasons. For example, a change in redundant data requires changes at multiple places in a database.

To eliminate redundant data from the database, we must take special care to organize the data stored in the tables. Database designers attempt to eliminate it as far as possible by using a technique called data normalization. *Normalization is a process that suggests decomposition of single table into multiple tables, thus creating a parent-child relationship.* Thus in case of Customer table in our sample database, a new table can be created to store information about

city. Let us name this table as City. We are aware that each area within a city has a distinct pincode associated with it. A pincode can be used to uniquely identify city and area within city. Thus pincode can be used as a primary key to identify each record in the City table. Table 10.1 shows the structure of City table and its sample records.

City		
Pincode	Area	CityName
380058	Bopal	Ahmedabad
380006	Law Garden	Ahmedabad
363421	Raska	Limbdi

Table 10.1 : City table

We now need to make changes in the Customer table. We can remove the City field from the Customer table. When the data of a new customer is inserted in the Customer table, we will use the pincode field to get the information regarding customer's city and area.

Similarly, the City field should be removed from Supplier and Employee tables. To maintain complete postal address information, we can also add the fields State and Country to which the city belongs. One way to maintain this information is as shown in Table 10.2.

Pincode	Area	CityName	StateName	CountryName
380058	Bopal	Ahmedabad	Gujarat	India
380006	Law Garden	Ahmedabad	Gujarat	India
363421	Raska	Limbdi	Gujarat	India

Table 10.2 : City table with State and Country

Now if we are given a pincode, the information pertaining to the location of customer can be searched from City table. But you can observe that above designed city table contains same flaw called data redundancy. Same city and state gets repeated multiple times. To avoid data redundancy, again City table has to be decomposed into multiple tables. In place of the field StateName, a field StateID should be stored and in place of the field CountryName, a field CountryID should be stored. A new table that stores information about the State and Country should be created. Figure 10.5 shows structure and relation of these tables.

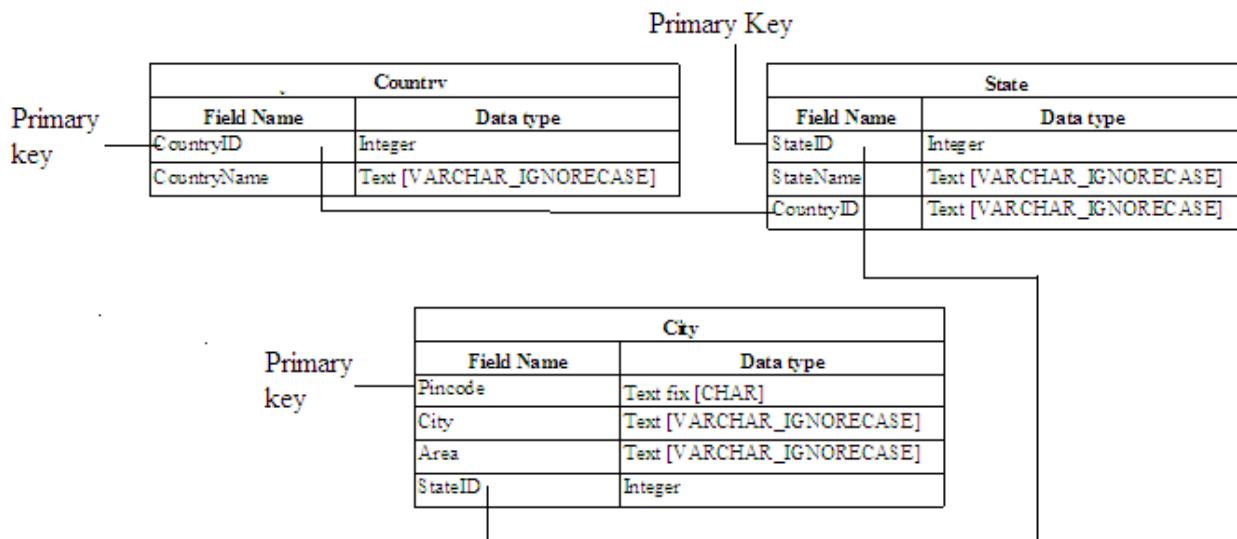


Figure 10.5 : Country, State and City Tables

Here we have assigned CountryID, StateID and Pincode fields as Primary keys of Country, State and City tables respectively. As seen in figure 10.5 now city, state and country tables are related with each other. Also as the pincode would be one of the fields in Customer, Employee and Supplier tables, these tables are related with the City table. **The table which contains primary key is known as master table.** As discussed earlier, two tables are related to each other using a common field. In above example, the State and Country table is related using CountryID field. The field CountryID acts as primary key in the Country table. The field CountryID in the State table thus borrows the values from the field CountryID in the Country table. The CountryID field in the State table is known as Foreign key.

Foreign key can be defined as field or combination of fields whose values are borrowed from another table. When it is combination of more than one field, it is known as composite foreign key. When a master table has composite primary key, tables related with it will hold composite foreign keys.

Note : Foreign key field cannot contain a value that does not exist in its related primary key field.

Now if you look at Product table, product category “Mobile” or “Camera” would be stored multiple times. Thus a separate table named ProductCategory should be created. Each product category now could be assigned a unique number. This unique number can then be used to identify the category of a product. For example, Mobile could be assigned category code 1. Thus when a record for new mobile is inserted in the Product table, a value ‘1’ is to be stored in the ProductCategory field. Note that the data type of the ProductCategory field in the Product table and CategoryCode field in the ProductCategory table should be same.

Similarly, the field SupplierName in the Product table can be replaced by the field Scode that refers to supplier code. Table 10.3 lists the primary key and foreign keys of each table after modification suggested so far.

Table name	Primary Key	Foreign Key
Country	CountryID	-
State	StateID	CountryID (referencing Country table)
City	CityID	StateID (referencing State table)
Supplier	Scode	Pincode (referencing City table)
Customer	Ccode	Pincode (referencing City table)
Employer	Ecode	Pincode (referencing City table)
ProductCategory	CategoryCode	-
Product	Pcode	CategoryCode (referencing ProductCategory table) Scode (referencing Supplier table)

Table 10.3 : Primary and Foreign keys of each table discussed in Sample database

Let us now design other tables to store data regarding transactions like purchasing of a product by the customer. We can call it as purchase order placed by customer. A customer can either place order online or can walk into the store and purchase a product in Modern Electronic Store. An order placed by customer indicates that information related to customer, employee and product involved in the transaction has to be maintained. We cannot miss out date of order placement. Again quantity of each product is equally important. A customer may purchase multiple piece of same model. Table 10.4 shows fields of the Order table.

Order	
Field Name	Data type
OrderID	Integer
OrderDate	Date
Ccode	Text fix [CHAR]
Ecode	Text fix [CHAR]
Pcode	Text fix [CHAR]
Quantity	Integer
SalePrice	Decimal

Table 10.4 : Order Table

Here OrderID is designated as a primary key. Additional field is necessary in this case as no other field or their combination can uniquely determine records. All fields of Order table may have duplicates. Try to insert a record <1, 23-June-2012, C01, E01, P00000001, 2>. Now assume that customer has purchased two different products, P00000001 and P00000002. Then a new record that needs to be inserted would be <1,23-June-2012,C01,E01, P00000002, 1> .

You can observe that details like order date, employee code and customer code are repeated. If a customer purchases 10 products at a time, 10 different records are to be inserted with too much of data duplications. OrderID cannot act as primary key in that case. A combination of OrderID and Pcode is to be designated as Primary key. Solution to the discussed problem is to decompose the Order table into two tables, Order and OrderDetail. Thus fields OrderDate, Ccode and Ecode would be stored only once with OrderID uniquely identifying them in the Order table. The OrderDetail table would then contain fields OrderID, Pcode, Quantity and SalePrice. Values of OrderID will be borrowed from Order table and thus it will be foreign key field. The OrderDetail table thus contains the details which are being repeated. Table 10.5 shows sample records of OrderDetail table.

OrderDetail			
OrderID	Pcode	Quantity	SalePrice
O00001	P00000001	2	35000
O00001	P00000002	1	20000
O00002	P00000009	1	43000

Table 10.5 : Sample records for OrderDetail

Now, what should be the primary key of the OrderDetail table? As discussed earlier, field designated as a primary key cannot have duplicate values. In OrderDetail table as shown in sample records, all the fields will have duplicate values. In such situations we need to find combination of fields that will have unique values. In the OrderDetail table, combination of OrderID and PCode definitely would be unique. The product code would not be repeated in a same order. Combination of two fields can be designated as a primary key. Combination of more than one field when designated, as a primary key, is also known as *Composite Primary Key*. Figure 10.6 shows structure of both the Order and OrderDetail tables.

Order		OrderDetail	
To store details of orders		To store details of products purchased by customer	
Field Name	Data type	Field Name	Data type
OrderID	Integer	OrderID	Text fix [CHAR]
OrderDate	Date	Pcode	Text fix [CHAR]
Ccode	Text fix [CHAR]	Quantity	Integer
Ecode	Text fix [CHAR]	SalePrice	Decimal

Figure 10.6 : Structure of tables Order and OrderDetail

Before proceeding further let us see how can we create composite primary key.

Composite Primary Key

To create composite primary key in OrderDetail table,

- Select row containing OrderID field.
- Press CTRL key and Select Pcode row. Both the rows will be simultaneously selected. See figure 10.7.
- Right click in selected area. A popup menu with option *Primary Key* will be displayed.

Click on *Primary Key* option and key symbol will be displayed in the left of the two selected rows.

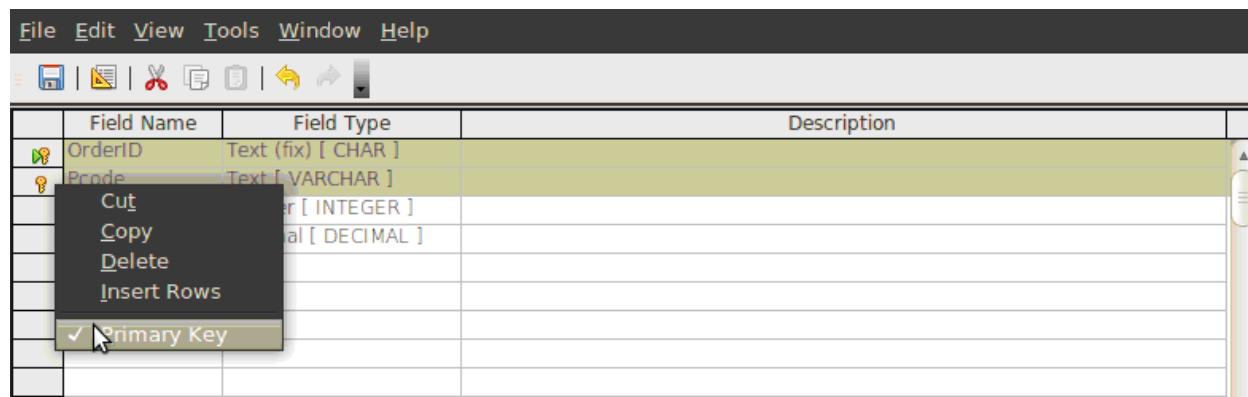


Figure 10.7 : Defining Composite Primary Key

Some more tables like OrderPayment and OrderShipment are also to be designed. Figure 10.8 shows the structure of both these tables.

OrderShipment	
To record shipping address of a customer who has placed order online	
Field Name	Data type
OrderID	Integer
ShippingAddress1	Text [VARCHAR_IGNORECASE]
ShippingAddress2	Text [VARCHAR_IGNORECASE]
Pincode	Text fix [CHAR]
ShipingDate	Date
ShippingStatus	Boolean

OrderPayment	
To store details of payment made by customer	
Field Name	Data type
OrderID	Integer
PaymentID	Integer
PaymentDate	Date
PaymentAmount	Decimal
PaymentMethod	Text [VARCHAR_IGNORECASE]
ChequeDDNumber	Text [VARCHAR_IGNORECASE]
BankName	Text [VARCHAR_IGNORECASE]
BankBranch	Text [VARCHAR_IGNORECASE]

Figure 10.8 : OrderPayment and OrderShipment tables

The OrderShipment table would contain address where the products are to be delivered. The ShippingStatus field is used to maintain the status of the product delivery. Initially its value would be set to 'N'. Later as the product is received by customer; it has to be set to 'Y'.

The OrderPayment table would contain details of payment received from customer. A customer may be given option of payment in installments and thus in the OrderPayment table OrderID and PaymentID are designated as composite primary key. For example, for OrderID “1” PaymentID could be 1, 2, 3, 4... and so on. For OrderID “2”, PaymentID would again start from 1. Payment method could be cash, cheque, credit or debit card. The PaymentMethod field would contain one of these four options. If the payment is made by cheque, then the cheque number and name of issuing bank would also be recorded in the OrderPayment table.

Make sure that you now have the Supplier, Employee, Customer, Product, Country, State, City, ProductCategory, Order, OrderDetail, OrderShipment, OrderPayment tables with required changes in your database. In case all these tables are not present create them using either the Wizard or Design View.

Creating Relationships between Tables in Base

By now, you might have designed the tables we have discussed. Assume that a record of a new customer from Goregaon, Mumbai, Maharashtra is to be entered in the Customer table. Till now Maharashtra is not entered in the State table, neither the Goregaon area of Mumbai city has been entered in the City table. Insert a record of a customer with pincode value 452001. Here the value 452001 is the pincode of Goregaon area, in Mumbai city, in the State of Maharashtra. Base will accept the record entered by you. But think logically, should this record entry be allowed? No we should not allow such entry to be done in the Customer table. The reason is very simple; there is no corresponding record in the City table that that be related with this entry. To enforce such restriction, we should use the *Relationship* feature of Base. Establishing relationship between tables will restrict the user from entering garbage data in the referenced fields.

Let us learn how to establish the appropriate relationship among the different tables of the database that we have created using Base.

- Click *Relationship...* option from the Tools menu.

Then select *Insert → Add Tables*. Alternatively you can click on the Add Tables button () seen in figure 10.9. Add Tables dialog box with list of tables created in the active database will be displayed.

- Select the Country table icon and click on Add button. You will find that selected table along with all its field is displayed in the background in Relationship window.
- Similarly, select the State table. The Relationship window will look as shown in figure 10.9.

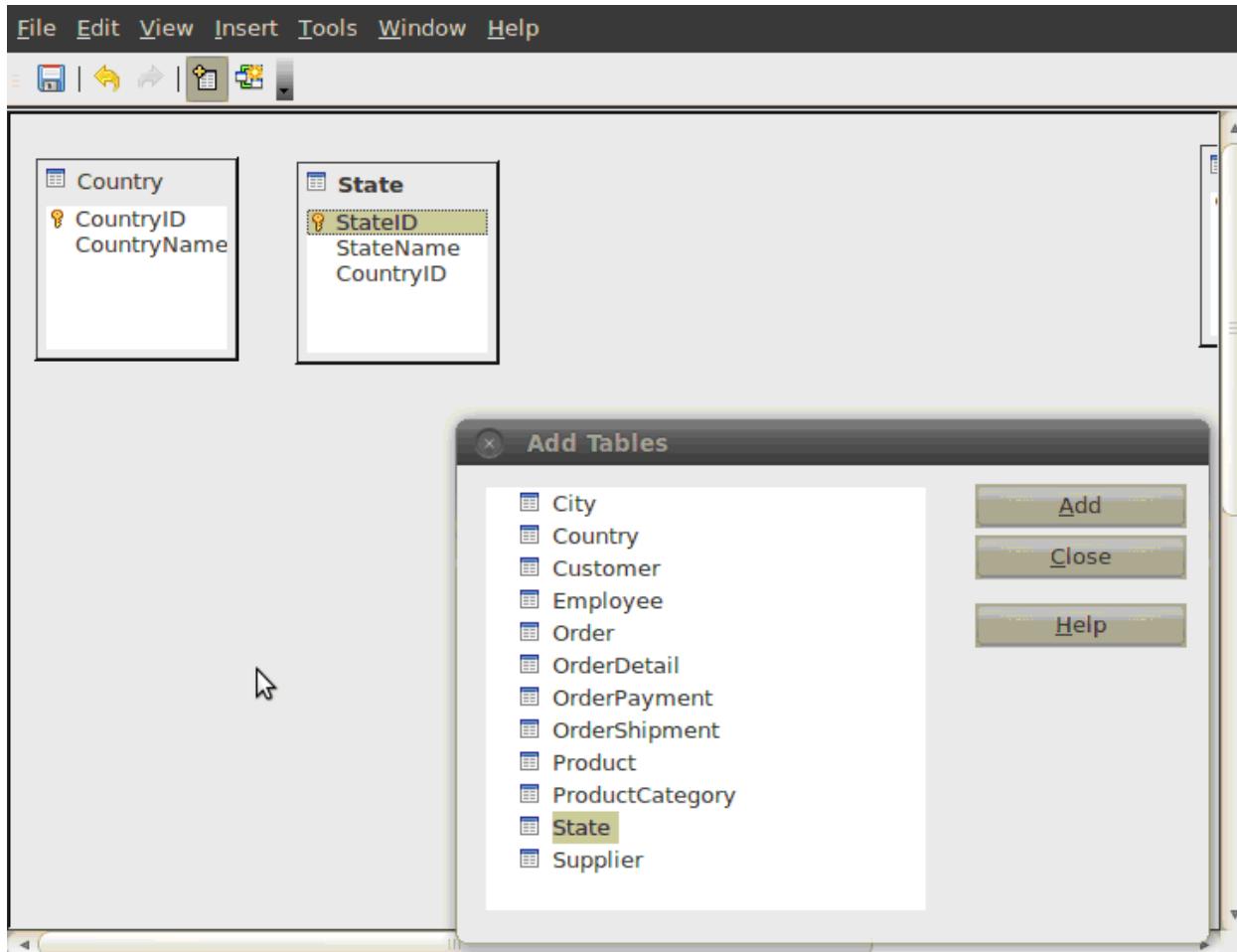


Figure 10.9 : Relationships dialog box

- To create a relationship, just use ‘drag and drop’ operation. Click on the CountryID field (primary key) of the Country table.
- Drag the CountryID field of the Country table and drop it on the CountryID field (foreign key) of the State table.
- A line connecting both the fields with labels *I* and *n* will be displayed as seen in figure 10.10. Notice that label text *I* is displayed on the primary key side and label text *n* is displayed on foreign key side. This indicates that the CountryID field is a primary key and will hold unique values, and each unique value stored in primary key field may be repeated *n* number of times in foreign key. This type of relationship where one value of referred table is associated with multiple values in referencing table is called *One-to-Many* relationship. Figure 10.11 shows some sample records stored in One-to-Many relationship. Types of relationships are also discussed in detail later in this chapter.

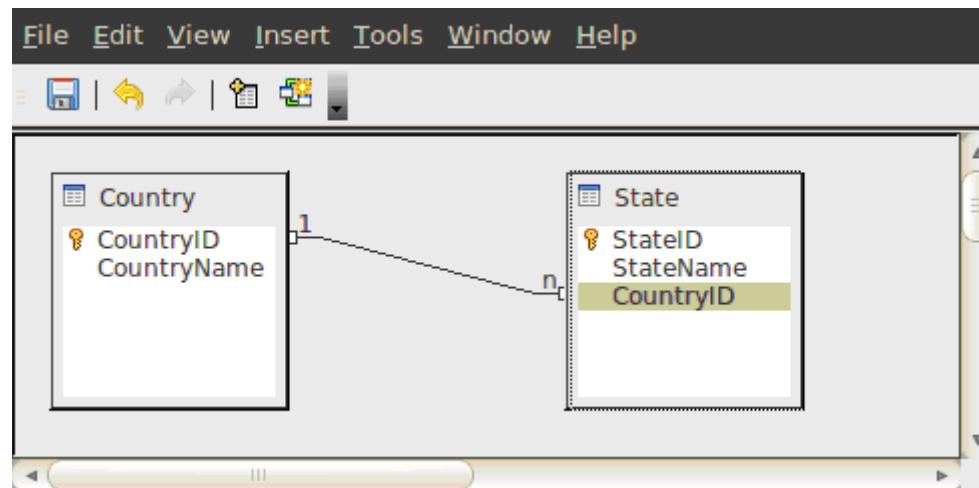


Figure 10.10 : One-to-Many relationship between Country and State

Figure 10.10 shows the relationship between the tables Country and State. Both the table shown in figure 10.10 can be moved to any desired location by clicking their title bar and then dragging it. Here the CountryID field of the Country table is known as *referenced field* while the CountryID field of the State table is known as *referencing field* (see figure 10.11).

Referenced field

Referencing field

Stored Once

Repeated Multiple Times

CountryID	CountryName
1	India
2	China
3	Japan

StateID	StateName	CountryID
1	Gujarat	1
2	MadhyaPradesh	1
3	Rajasthan	1

Figure 10.11: Records indicating One-to-Many Relationship

Note : To create a relationship between two tables, the data types of *referencing field* and *referenced field* must be same. In case data type of both the fields do not match, error as shown in figure 10.12 would be displayed while creating relationship.

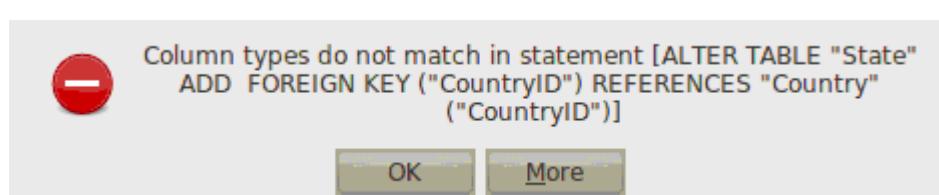


Figure 10.12 : Column data type mismatch error in creating relationship

Referential Integrity

Once the relationship between two tables is created, double-click the line depicting relationship. Relations dialog box as shown in figure 10.13 will be displayed. Till now, we are very clear with the statement that the Country table and the State table are related with each other. Thus a user

can enter data of only those country's details in the State table which have been already entered in the Country table. As seen in figure 10.11, first India was inserted in the Country table. Later three states of India were inserted in the State table. Record pertaining to India in the Country table can be considered as *master or parent* record, while records pertaining to India in the State table can be considered as *transaction or child* records.

But what if now user deletes a parent record from the Country table? What about its related record in the other tables? Say for example, we have entered say ten records in the State table that have used CountryID as 1. What should be done with records of the Country table with CountryID as 1 in the State table, if the user decides to delete or update this record from the master table? There must not be any entry in the State table without a related record in the Country table. This concept is known as Referential Integrity. **Referential Integrity** principle can be stated as, *No unmatched foreign key values should exist in the database.*

The database designer shall choose and set one of the four options shown in figure 10.13 depending on the transaction requirement of the company. These options allow us to maintain referential integrity in database while performing an update or delete operation.

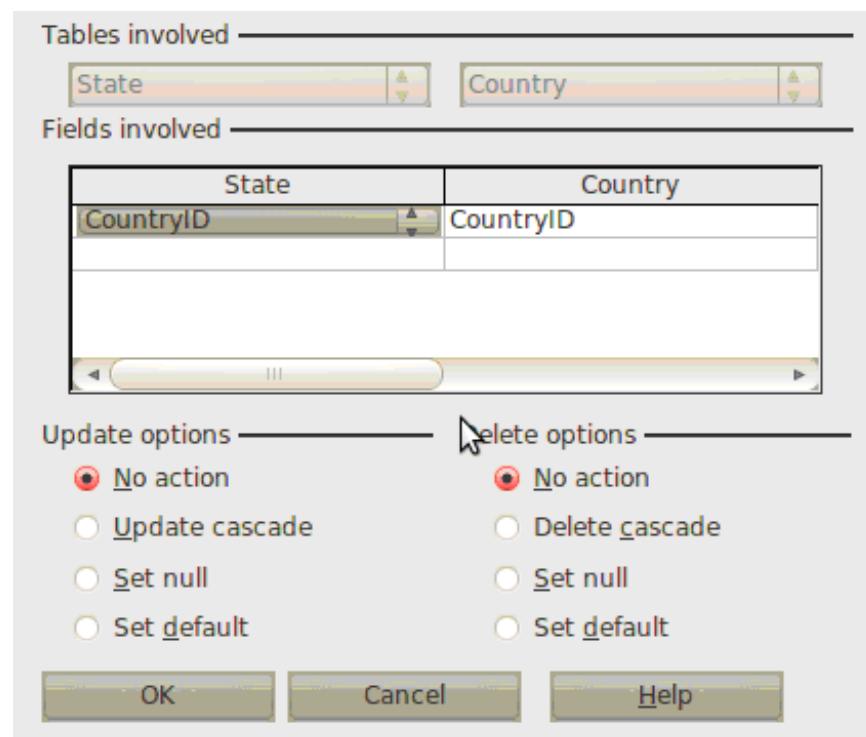


Figure 10.13 : Enforcing referential integrity

No action: This option states that a user should not be allowed to delete or update any record if its related record exists in some other table. Select *No Action* option and then try to delete the record containing India's details. Base will confirm whether you want to delete records or not and then will display message as shown in figure 10.14.



Figure 10.14 : Enforcing Referential Integrity using No Action

Update cascade: This option states that if user is allowed to delete or update referenced values, all the related records should be automatically deleted or updated.

Set null: This option states if user deletes or updates the referenced field, all the related records will hold null value in the related field.

You can similarly create the relationship between the other tables of the Modern Electronic Store database. You can add other tables by clicking on *Add Tables* icon in tool bar as shown in figure 10.15. *Add Tables* dialog box as shown previously in figure 10.9 will be displayed.

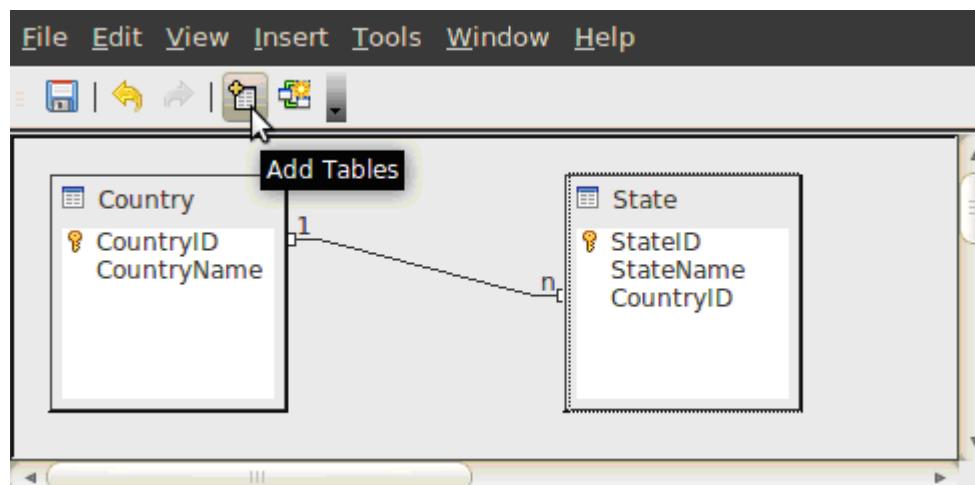


Figure 10.15 : Adding Tables in Relationship window

Once you create the relationship between all tables of our sample database, relationship window will look similar to the one shown in figure 10.16.

Now as it is necessary to save the relationship we have recently established, we must save it by clicking on the Save button on the tool bar. In case we forget to save, then to remind us, an alert message will be shown on the screen. At times, changes made in relationships are not reflected in database after saving also. If such a thing happens then close the database and reopen it. You will find that modifications in relationships are implemented and visible.

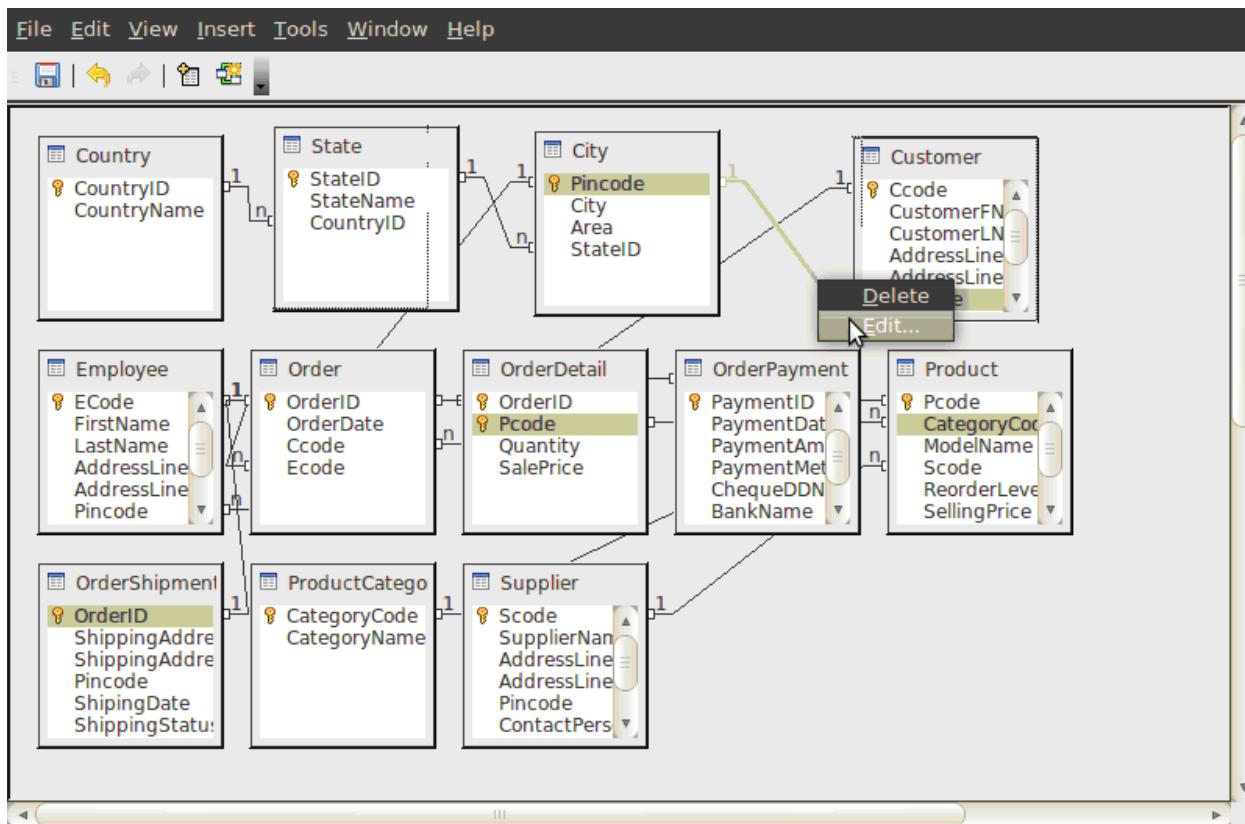


Figure 10.16 : Relationship of all tables in Database

Editing Relationships

We can delete or edit the relationship established between any two tables. For that, we just have to open the relationship screen and then click on the desired relationship line drawn between two tables. By doing so, the line will get thickened. After this if we right click, a popup menu will be visible as shown in figure 10.16.

Here, we can select the necessary option depending upon our requirement. If we select the Edit option from the popup menu then a Relationship dialog box will be shown where we can edit the relationship attributes as per our requirement. Alternatively we may select Delete option and remove the relationship. So now we know how to create, edit or delete various types of relationships between any two tables of the database. Create relationships between all the tables if not already created.

Now, let us discuss types of relationships in detail. Conceptually there can be three major types of relationship between any tables of the database.

- One to One
- One to Many
- Many to Many

Type of relationship among the data of two tables is defined based on how many corresponding records can be there in second table corresponding to the first table. Let us understand each of them properly.

One to One Relationship : As its name suggests, One to One Relationship indicates that one specific record of a particular table has one and only one corresponding record in the other table of the database. Do not get surprised, as it's not new to us and very common in our day-to-day life examples. Think of a theatre. How many people can seat in a chair? Naturally, One. Then how many seats can be occupied by one person? Definitely, One. So, we can say that the relationship between Chair and Person is One-to-One Relationship. You can think of relationship between your class and class teacher or library and library card issued to a student. You will definitely agree that, any class in your school will have one and only one class teacher. At the same time any teacher is assigned one and only one class to look after. So we can say that, the Class and Class Teacher is an example of One-to-One relationship.

Observe relationships created in our sample database. The field OrderID in the Order table is related to the field OrderID in the OrderShipment table. Now only one record corresponding to each order for example, 1 would be found in OrderShipment table. It indicates that One-to-One relationship exists between the Order and OrderShipment tables. You may think then why we have not added the fields of OrderShipment to the Order table. In ModernElectronicDatabase case, the OrderShipment details are not to be populated for every order. The OrderShipment details are to be recorded only when customer places order online or expects the delivery of products at home or some other address when he has physically walked into the store and done the purchasing. Thus if the Order table is not decomposed, Null values will be populated in all the OrderShipment fields. Hence, it is better to decompose tables and maintain One-to-One relationship in such cases.

One-to-Many Relationship: The most common type of relationship between the tables is One-to-Many relationship. In this type of relationship one specific record of a particular table may have many corresponding records in the other related table of the database. One customer can purchase multiple products. Similarly, one city belongs to single state, but one state can have multiple cities. Imagine a class teacher scenario in your school, there can be one and only one class teacher to any class. But any class teacher definitely have many student enrolled with him. That's why the relationship between Class Teacher and Student is One-to-Many.

Many-to-Many Relationship: The Many-to-Many relationship occurs in the case of some specific record, which has to be stored more than once in both the tables. In our example of ModernElectronicStore database, we have maintained information of employees. Till now we have assumed that an employee will hold a single designation. At times in an organization an employee may hold multiple designations. Thus in that case we can state that an employee may have multiple designations and a single designation may belong to multiple employees. Separate table that stores employee designation has to be maintained in that case. Thus there exists Many-to-Many relationship between these two entities. Figure 10.17 shows a sample many-to-many relationship scenario.

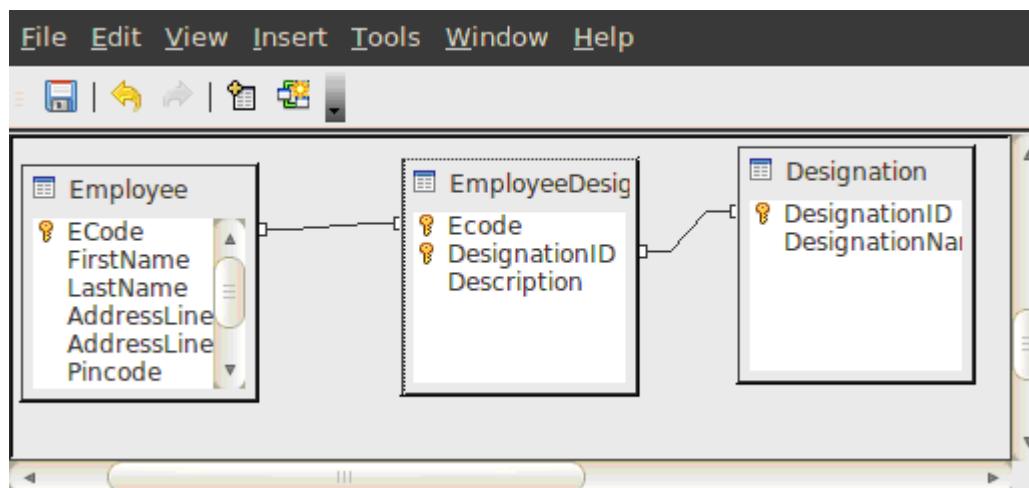


Figure 10.17 : Many-to-Many relationship between Employee and Designation

But you can see in figure 10.17, we don't have any direct relationship between the tables Employee and Designation. This is because in database Many-to-Many relationships are split into two One-to-Many relationships by creating a third table. In our example we have created a new table named EmployeeDesignation, which has One-to-Many relationship with the Employee table and Designation table both. Thus EmployeeDesignation table acts as a junction table between the Employee and the Designation table.

Till now we have created tables, decided primary key fields and foreign key fields and related tables with each other. Now let us discuss about field properties.

Field Properties

As discussed in the earlier chapter, before working with the data we have to create a well defined structure of tables. To define a field for any table we have to perform the following steps:

- Specify the *Field Name*
- Choose a proper *Data type* for the field
- Write *Description* of the field. (Optional)

In the field of information technology, the *data* is the raw material used to produce finished goods known as *Information*. It is a universal truth that 'We can't expect getting best without giving better'. In other words, we can say that the quality of the output obviously depends upon the quality of input. If we can control and restrict the values being stored at the time of data entry then, it can be the best practice to control the input data, which can generate the correct information.

What if a user leaves the Name field of the Customer table blank? Even well experienced data entry operators many times enter incorrect data by mistake. To prevent such mistakes, Base provides a special facility called '*Field Level Validation*'. It is nothing but putting some possible rules and checkpoints for data to be entered for each field. This further restricts the values entered in a particular field. Data validation is the important aspect of data processing. You must note that, if the data entered in various tables is correct and accurate then and then only the database will become a valuable asset to your organization.

Base has very important and interesting feature called *Field Properties*. **Field properties** determine how the values in the field are stored and displayed. Each type of field has a particular set of properties. For example, in what format will the user enter the data (dd-mm-yy or mm-dd-yy) in the OrderDate field, how should it be displayed while printing or viewing the data and what if user enters invalid date like 12/14/12 for a date format dd-mm-yy.

Field Properties can be decided at the time of designing the structure of the table. Select the field in Table Design Window, Base provides a *Field Properties* box as shown in figure 10.18 with appropriate default values according to the corresponding selected data type.

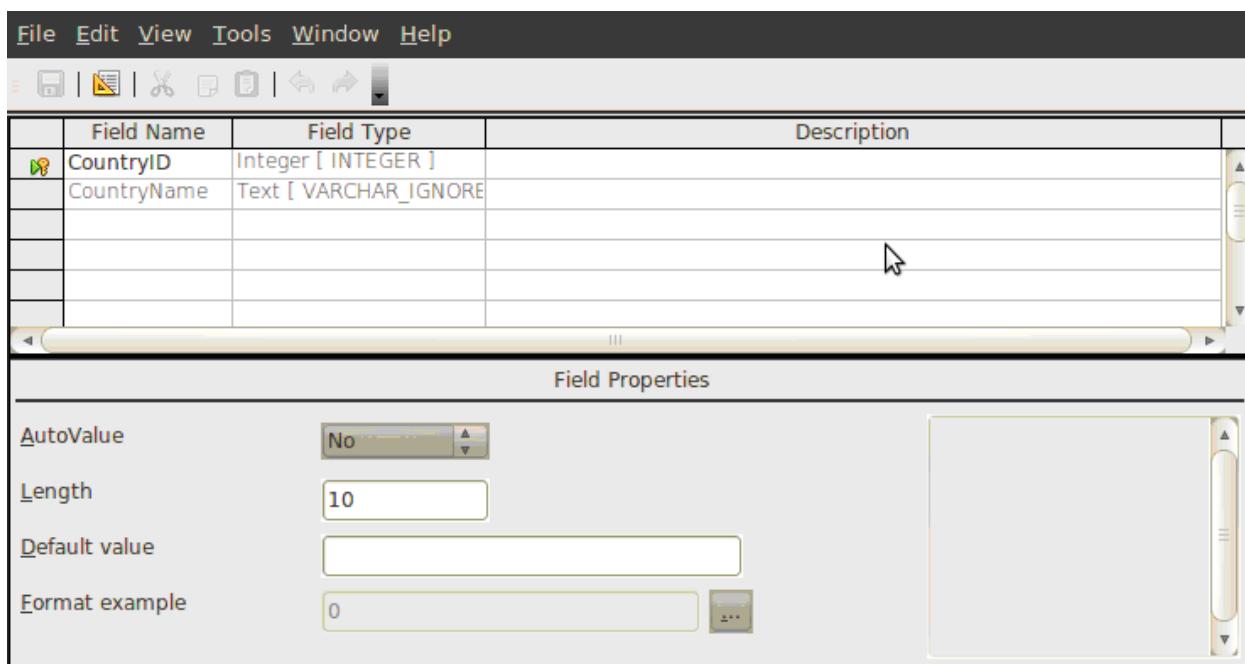


Figure 10.18 : Field Property dialog box

The various *Field Properties* work as field level validation for the table. *Field Properties* displayed on the screen may vary depending upon the data type that we have selected for the particular field. Let us discuss some of the common field properties in detail:

AutoValue: As can be seen in figure 10.18, *AutoValue* property has drop down box (combo box) next to it. Click on the drop down box and you will find two values, “Yes” and “No” within it. If the value of an *AutoValue* is set to “Yes” for a numeric field, Base can automatically increment the value for each new record. This field is particularly used for numeric fields designated as primary key. For example, the Country table consists of a field CountryID as primary key. The data type of the CountryID field is *integer*. The values expected for this field are 1, 2, 3... and so on. Thus *AutoValue* can be set to “Yes” in this case.

Entry required: Out of all the fields in the database there may be some important field, which

must not be left empty. Now, it may happen that the data entry operator forgets to enter the required value in important fields. How about, if we do not allow the user to leave such important field as blank at the time of data entry itself? We can achieve this requirement by setting the field property *Entry required*. The setting of this property can either be ‘Yes’ or ‘No’ (see figure 10.19).

This property determines whether the user must enter a value for a particular field to complete a record entry. We should set this property to ‘Yes’ if we do not want to allow the user to skip entry in any important field of our table. For example, we must set this property type for the fields that store value related to name of a person.

Note : We do not need to set the value of *Entry required* property to *Yes* for the field declared as *primary key*.

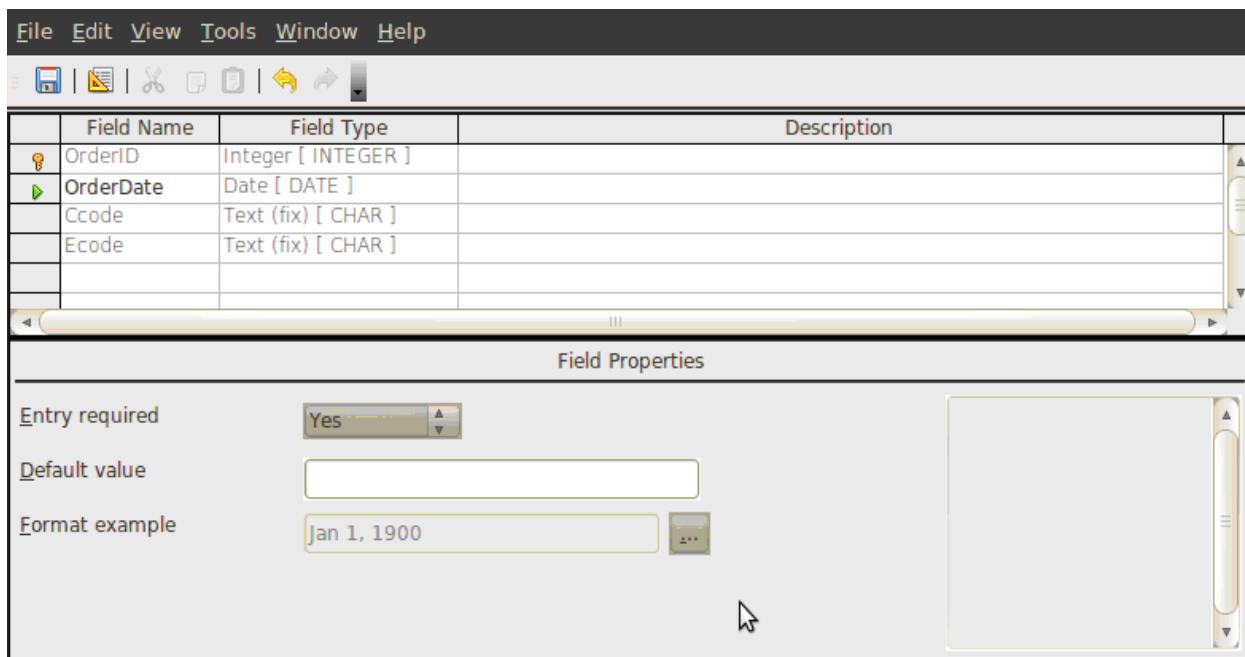


Figure 10.19 : Entry required Property set to Yes

Length: Whenever we select the *Text* data type for a particular field, this property automatically takes some predefined value e.g. 50 for *Text* data type. We can specify field size as per our requirement. Base automatically assigns some predefined field size to the various *Numeric*, *Date/Time*, *Yes/No* and *Memo* data types. That’s why in such cases this property will be disabled on the screen.

Default value: Many times it may happen, that we want to store some predefined default value automatically in a particular field at the time of entering a new record in the table. Suppose that, in the OrderDetail table we have a field named Quantity. Now, we may want to store value ‘1’

as a default value for this field. Select the Quantity field, type ‘1’ in the text box next the *Default value* label as shown in figure 10.20. Once we set this property for a field, the specified default value will automatically be displayed when we add a new record to the table. A user can change the value if required at the time of data entry.

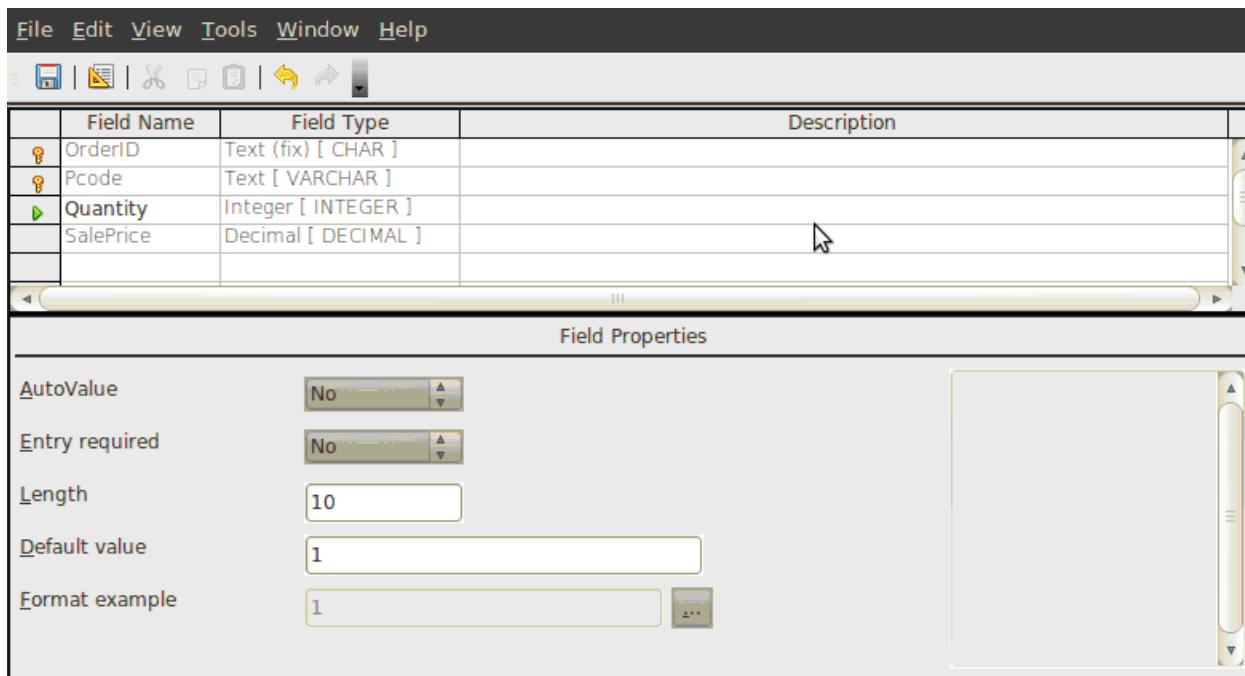


Figure 10.20 : Specifying Default value for a field

Format: This property specifies the format for displaying and printing the data that we enter in the field. It has no effect on the way the data is stored and it doesn’t check for invalid entries. The Format property uses different settings for different data types. Base provides some predefined formats for *Number*, *Date/Time*, and *Yes/No* data types.

For example, to change format of the field OrderDate in the Order table,

- Open the Order table in Design view.
- Select the OrderDate field and click on button at the end of Format example label in Field Properties pane. A Field Format dialog box as shown in figure 10.21 will open.
- Select Date option from the list under the heading Category.
- Select Dec 31, 1999 option from the list under the heading Format.
- Click on OK button.

Now open the table and enter valid date in any format. You will find that Base automatically converts all the entry in this field in the format that shows first three characters of month, a space then two digit date and two digit year values separated by a comma.

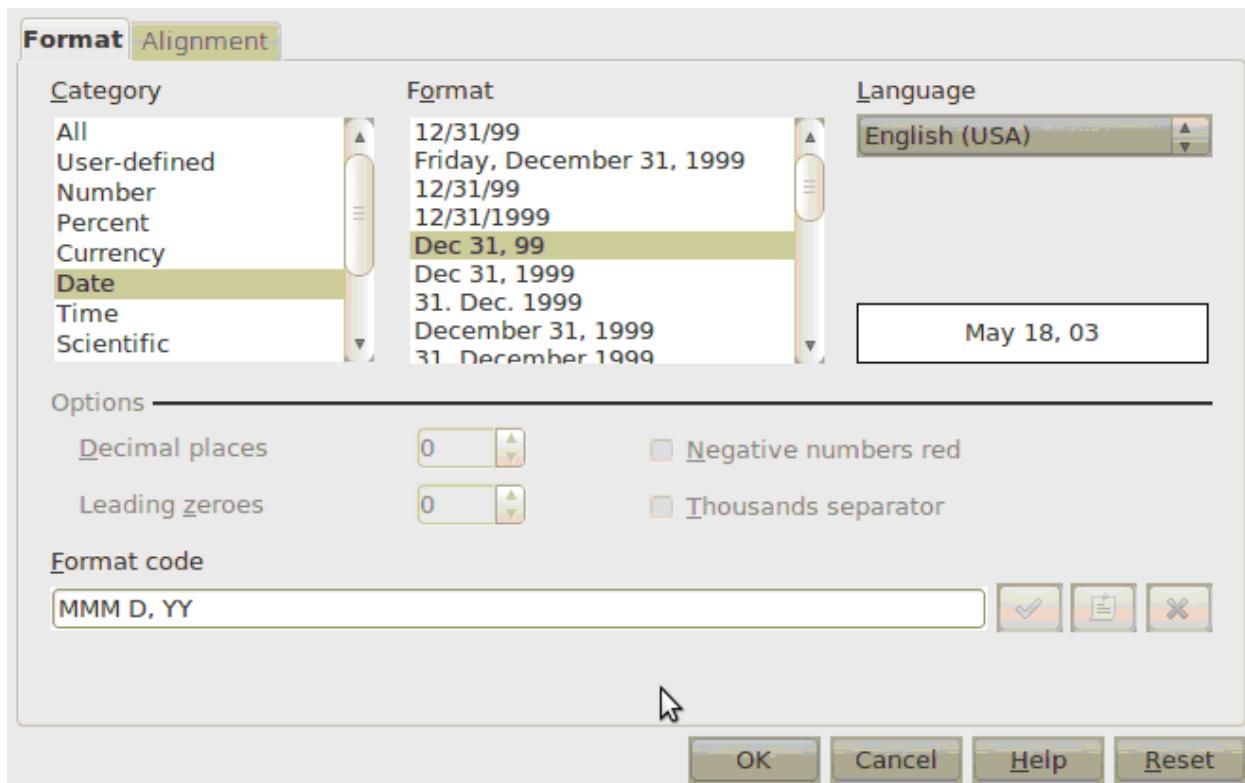


Figure 10.21 : Predefined formats

Summary

In this chapter we have discussed the operations that can be performed on the table. We learnt how to create relationship between the tables and types of relationship that can exist between the tables. Finally we learnt how to control data entry and restrict the user from entering wrong data in the table using Base feature called Field Properties. We need to populate designed database for Modern Electronic Database with meaningful records and in the next chapter we will see how the inserted data helps us generate information using Queries.

EXERCISE

1. Explain how we can perform insertion operation in a table.
2. What do you mean by relations in Base?
3. Explain types of relationships giving suitable example.
4. Explain the concept of foreign key giving suitable example.
5. What do we mean by Composite primary key? Explain giving suitable example.
6. Explain data redundancy giving suitable example.
7. State referential integrity rule.
8. What does required field property signify?
9. Explain the importance of Default value in database.

(10) Which of the following statement is true for Default field property in Base?

- (a) Default value can be numeric value only.
- (b) Default value once set can be changed later.
- (c) Default value cannot be greater than 500.
- (d) Default value should be greater than value set in Length.

(11) Which of the following property is equivalent to NOT NULL?

- | | |
|--------------|-------------|
| (a) Length | (b) Default |
| (c) Required | (d) Format |

LABORATORY EXERCISES

- 1.** The structure of the table is given in the form of Tablename(Attribute1, Attribute2,.....). Perform the following five operations for each database using Base.
- (a) Create tables with the given names and attributes.
 - (b) Decide appropriate data type for each attribute
 - (c) Identify primary key and foreign key in each of the database.
 - (d) Establish relationships amongst the tables created within a database.
 - (e) Insert at least five appropriate records in each table.

A.	Movie(MovieId, MovieName, DateOfRelease) Screen(ScreenId, Location, MaxCapacity) Current(MovieId, ScreenId, DateOfArrival, DateOfClosure)
B.	Customer(CustomerID, CustomerName, Address, City, BirthDate, ContactNo) Account (AccountNo, CustomerId, AccountType, AccountBalance) Transaction(TransId, AccNo, TransDate, TransType, TransAmount)
C.	Book(BookId, BookTitle, Description, BookAuthor, Status) Student(StudId, StudName, Address,City, BirthDate,ContactNo) Book_Issue(TransId, BookId, StudId, IssueDate, ReturnDate)
D.	Employee(EmpId, EmpName, Address, BirthDate, ContactNo, ManagerId) Department(DeptId, DeptName) Employee_Department(EmpId, DeptId, Salary)
E.	Flight(FlightId, CompanyName, FlightFrom, FlightTo, FlightFare, Capacity) Passenger(PassengerId, Name, Address, City, BirthDate, Gender, ContactNo) Flight_Scheduled(Transid, FlightId, DepartureDate) Flight_Passenger(Transid, PassengerId)

F.	Train(TrainId, TrainName, TrainFrom, TrainTo, DepartureTime, ArrivalTime) Train_fare(Trainid, Class, Fare) Passenger(PassengerId, Name, Address, City, BirthDate, ContactNo, Email_id) Train_Passenger(TrainId, PassengerId)
G.	Vehicle(VehicleId, Name, Type, Price, Description) Customer(CustomerId, CustomerName, Address, BirthDate, ContactNo) Vehicle_Customer(VehicleId, CustomerId, PurchaseDate, DeliveryDate)
H.	Product(ProductId, ProductName, Quantity, ProductPrice) Salesman(SCode, SName, SAddress, BirthDate, ContactNo) SalesOrder(SCode, ProductId, QtySold)
I.	Customer(CustomerId, CustName, Gender, CustAddr, CustCity, EmailID, ContactNo) Magazine(MagazineId, MagazineName, UnitRate, Publication) Subscription(CustomerId, MagazineId, StartDate, EndDate)
J.	Employee(EmpCode, EmpName, EmpAddress, EmpCity, EmpSalary, EmpJobName) Project(ProjCode, ProjName, StartDate, ProjPrice) WorksFor(ProjCode, EmpCode, HoursWorked)

2. Identify primary key and foreign key from the sample tables shown in figure 10.22 and 10.23.

Student	
Field Name	Description
Gmo	General Register Number
Firstname	Name of the Student
Surname	Surname of the Student
Address	Address of the Student
City	City
Pincode	Pincode
Birthdate	Date of Birth
Gender	Male or Female
Standard	Studying in which standard
Join_date	Date of Joining School
Leaving_date	Date of Leaving School

Teacher	
Field Name	Description
Firstname	Name of the Teacher
Surname	Surname of the Teacher
Address	Address of the Teacher
City	City
Pincode	Pincode
Phone_no	Phone number of teacher
Email_id	E-mail id of teacher
Mobile_no	Mobile number of teacher

Subject	
Field Name	Description
Sub_Name	Name of the Subject
Details	Description of the subject

Figure 10.22 : Master tables of Student Management System

Standard_Subj : To store details about subjects taught in each standard	Subj_Teacher : To store details about which teacher is teaching the subject in each standard																																	
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Field Name</th><th>Data Type</th><th>Description</th></tr> </thead> <tbody> <tr> <td>Standard</td><td>Number</td><td>Which Standard</td></tr> <tr> <td>Scode</td><td>Text</td><td>Which Subject</td></tr> </tbody> </table>	Field Name	Data Type	Description	Standard	Number	Which Standard	Scode	Text	Which Subject	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Field Name</th><th>Data Type</th><th>Description</th></tr> </thead> <tbody> <tr> <td>Year</td><td>Number</td><td>Current Year</td></tr> <tr> <td>Standard</td><td>Number</td><td>In which Standard</td></tr> <tr> <td>Scode</td><td>Text</td><td>Subject Code</td></tr> <tr> <td>Tcode</td><td>Text</td><td>Teacher Code</td></tr> </tbody> </table>	Field Name	Data Type	Description	Year	Number	Current Year	Standard	Number	In which Standard	Scode	Text	Subject Code	Tcode	Text	Teacher Code									
Field Name	Data Type	Description																																
Standard	Number	Which Standard																																
Scode	Text	Which Subject																																
Field Name	Data Type	Description																																
Year	Number	Current Year																																
Standard	Number	In which Standard																																
Scode	Text	Subject Code																																
Tcode	Text	Teacher Code																																
Attendance : To store daily attendance of the Students	Result : To store result details of each student																																	
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Field Name</th><th>Data Type</th><th>Description</th></tr> </thead> <tbody> <tr> <td>Gmo</td><td>Number</td><td>General Register of a Student</td></tr> <tr> <td>Date</td><td>Date/Time</td><td>On which Date</td></tr> <tr> <td>Pr_ab</td><td>Yes/No</td><td>Present or Absent</td></tr> </tbody> </table>	Field Name	Data Type	Description	Gmo	Number	General Register of a Student	Date	Date/Time	On which Date	Pr_ab	Yes/No	Present or Absent	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Field Name</th><th>Data Type</th><th>Description</th></tr> </thead> <tbody> <tr> <td>Gmo</td><td>Number</td><td>General Register No.</td></tr> <tr> <td>Scode</td><td>Text</td><td>Of which Subject</td></tr> <tr> <td>Month</td><td>Text</td><td>In which Month</td></tr> <tr> <td>Year</td><td>Number</td><td>Which Year</td></tr> <tr> <td>Term</td><td>Number</td><td>Name of the Term</td></tr> <tr> <td>Marks</td><td>Number</td><td>Marks scored</td></tr> </tbody> </table>	Field Name	Data Type	Description	Gmo	Number	General Register No.	Scode	Text	Of which Subject	Month	Text	In which Month	Year	Number	Which Year	Term	Number	Name of the Term	Marks	Number	Marks scored
Field Name	Data Type	Description																																
Gmo	Number	General Register of a Student																																
Date	Date/Time	On which Date																																
Pr_ab	Yes/No	Present or Absent																																
Field Name	Data Type	Description																																
Gmo	Number	General Register No.																																
Scode	Text	Of which Subject																																
Month	Text	In which Month																																
Year	Number	Which Year																																
Term	Number	Name of the Term																																
Marks	Number	Marks scored																																
Fees : To store details of Fee amount paid by students	Sports : To keep record of Interest of students in Sports																																	
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Field Name</th><th>Data Type</th><th>Description</th></tr> </thead> <tbody> <tr> <td>Gmo</td><td>Number</td><td>General Register Number</td></tr> <tr> <td>Date</td><td>Date/Time</td><td>On which date</td></tr> <tr> <td>Tuition_fee</td><td>Currency</td><td>Tuition fee paid by the student</td></tr> <tr> <td>Term_fee</td><td>Currency</td><td>Term fee paid by the student</td></tr> </tbody> </table>	Field Name	Data Type	Description	Gmo	Number	General Register Number	Date	Date/Time	On which date	Tuition_fee	Currency	Tuition fee paid by the student	Term_fee	Currency	Term fee paid by the student	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Field Name</th><th>Data Type</th><th>Description</th></tr> </thead> <tbody> <tr> <td>Gmo</td><td>Number</td><td>General Register No.</td></tr> <tr> <td>Interest</td><td>Text</td><td>Interested in which activity</td></tr> <tr> <td>Achievement</td><td>Memo</td><td>Any achievements?</td></tr> </tbody> </table>	Field Name	Data Type	Description	Gmo	Number	General Register No.	Interest	Text	Interested in which activity	Achievement	Memo	Any achievements?						
Field Name	Data Type	Description																																
Gmo	Number	General Register Number																																
Date	Date/Time	On which date																																
Tuition_fee	Currency	Tuition fee paid by the student																																
Term_fee	Currency	Term fee paid by the student																																
Field Name	Data Type	Description																																
Gmo	Number	General Register No.																																
Interest	Text	Interested in which activity																																
Achievement	Memo	Any achievements?																																
Cultural : To keep record of Interest of students in Cultural Activities																																		
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Field Name</th><th>Data Type</th><th>Description</th></tr> </thead> <tbody> <tr> <td>Gmo</td><td>Number</td><td>General Register Number</td></tr> <tr> <td>Interest</td><td>Text</td><td>Interested in which activity</td></tr> <tr> <td>Achievement</td><td>Memo</td><td>Any achievements?</td></tr> </tbody> </table>	Field Name	Data Type	Description	Gmo	Number	General Register Number	Interest	Text	Interested in which activity	Achievement	Memo	Any achievements?																						
Field Name	Data Type	Description																																
Gmo	Number	General Register Number																																
Interest	Text	Interested in which activity																																
Achievement	Memo	Any achievements?																																

Figure 10.23 : Transaction tables of Student Management System

Retrieving Data Using Queries



In the previous chapter, we have mentioned that database is a huge collection of data. Now, think of a situation where we want to retrieve information from a database. Now suppose we want to send information about Independence Day offers to our customers, we would need list of our customers along with their addresses. Assuming that this information is stored in a table named Customer, the details can be obtained by opening this table. Query is the feature of Base, which can be used to retrieve specific set of information from database. The user can retrieve data according to his choice, criteria and format irrespective of how it is stored in the database. We can ask questions like “Give me details of customer who has purchased products worth more than one lakh rupees in the current year” or “Give me details of products that have not been sold even once”. One of the key reasons to use Database Management System is its ability to design, save and use the queries as and when required.

Defining Query

Query basically means asking question, doing inquiry or performing analysis. In Base, to query is to ask a question about the information in the database. Through a query one can tell Base to display exactly which fields and records a person would like to view from the database. It is a set of rules for fetching information from a table, or from several tables at once. The result of a query is itself in the form of a table. It consists of a set of records, organized in rows (one per record) and columns (one per field).

To create a query, open a database and click on the icon labeled as Queries in the left hand pane. The query window is organized like the other windows in Base. Different ways to create a query will be shown in the Tasks pane displayed at the top. If you have already created some queries then it will be listed under the pane titled Queries. Figure 11.1 shows the window when Queries icon is selected.

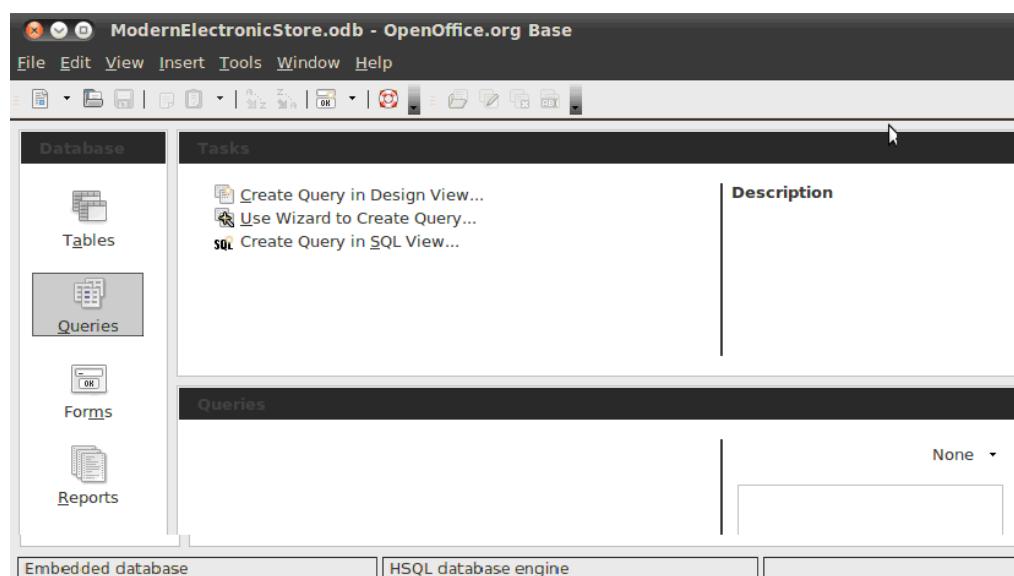


Figure 11.1 : Queries Window

As you can see in figure 11.1 under the Tasks pane, Base offers three different ways to create a query.

- Create Query in Design View...: This view provides a minimal amount of guidance.
- Use Wizard to Create Query... : Wizard guides us through the process of query creation.
- Create Query in SQL View... : It provides no guidance at all. Users need to have knowledge about fourth generation computer language called Structured Query Language.

If you are database expert, you might prefer Design or SQL View. However, Base's Query Wizard has the advantage of helping us organize our thoughts while requiring no previous knowledge.

Let us discuss each of these options one by one. We will create a query that gives us list of customer names and addresses.

Query Creation Using Wizard

- Double click the option *Use Wizard to Create Query....* A Query Wizard dialog box as shown in figure 11.2 will be displayed. Notice the left side of dialog box. Eight steps to create a query have been given. Only first step, Field selection is compulsory. It helps us in identifying the fields that are to be displayed in our output. Other steps allow us to format the output and can be skipped if not required.
- The first step in creating the query is to select the table and the set of fields in that table from which information is to be retrieved. We may select all or some of the fields visible in the *Available fields* list box. You can use the left and right arrow buttons to move the fields from *Available fields* list box to *Field in the Query:* list box. The fields that you finally select to be part of your query will be listed under *Field in the Query:* list box. These fields can then be arranged in the order required using the up and down buttons. Once the fields are finalized click on Next button. Observe that in figure 11.2 we have selected table Customer and are able to see all the fields related to it under *Available fields* list box.

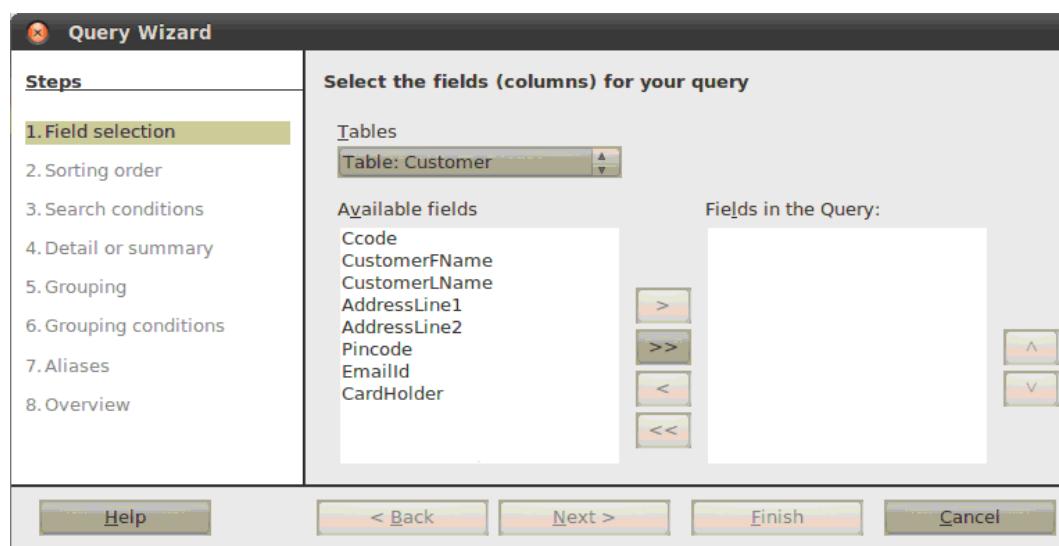


Figure 11.2 : Selection of table and fields for query

- The second step is to mention the sort order in which output of the query will be displayed. It allows you to select up to four fields for deciding the sorting order of the output. For example, we might want to display the query result sorted in order of first name initially, and then by last name of the customer. Once the sorting order is decided click on Next button. Figure 11.3 shows how to select the sorting order of fields.

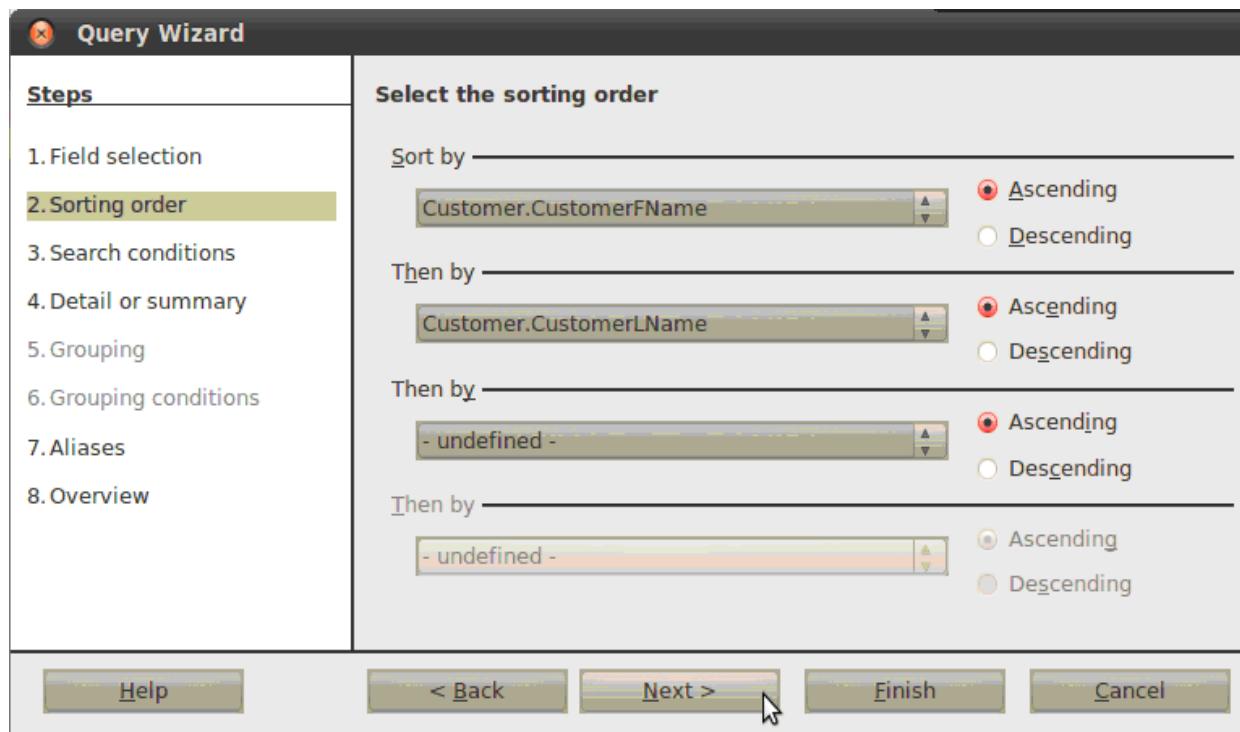


Figure 11.3 : Applying sorting on field

- In the third step of the wizard, we actually set up the query. Here we have to select appropriate values for the *Field*, the *Condition*, and the *Value* parameters. We can define maximum three search conditions in one query. In the case of the address query, if you want list of customers with “Shah” as their last name, the criteria would be simple: You would select the CustomerLName field from the drop down under the label *Fields*, then select *is equal to* from the drop down under the *Condition* label and finally in the text box under the label *Value*, type *Shah*. Observe that we have two options in this step. *Match all of the following* and *Match any of the following*. Since we have only one condition, we wouldn’t need to change the default setting. If multiple search conditions are to be set, like if we are looking for customers with last name as Shah or Patel then, we need to choose *Match any of the following*. Once the search conditions are finalized click on Next button. Figure 11.4 shows the search condition settings for listing of all customers with last name as Shah.

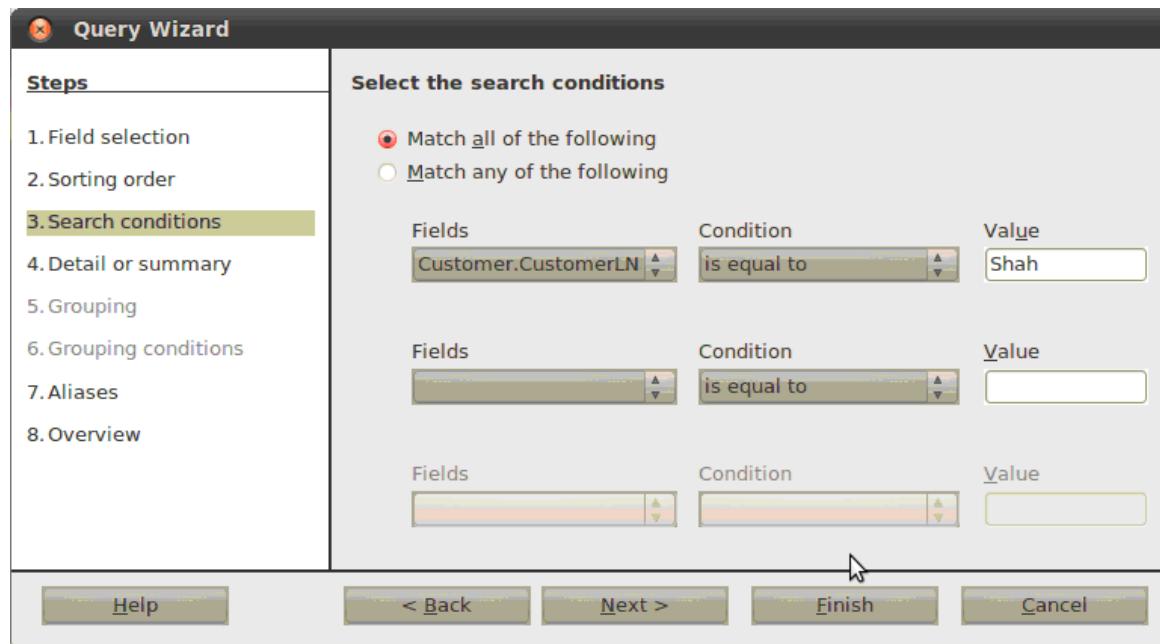


Figure 11.4 : Applying search condition on field

- At this point Base skips three steps in its standard wizard and jumps to the seventh step. In our case, the selected Customer table does not contain any numeric field and so steps including options to summarize or perform numerical calculations are skipped.
- In the seventh step, Base expects aliases for selected field names. The purpose of this step is to make the query wizard display the field names in human readable form. It allows us to add small touches to the field name such as spaces between words and writing full forms of short field names like First Name for CustomerFName. This step is also optional, we may add aliases if required only. If we do not add aliases then field names of a table will be displayed as it is in a query. Once all aliases are decided click on Next button. Figure 11.5 shows how to add aliases.

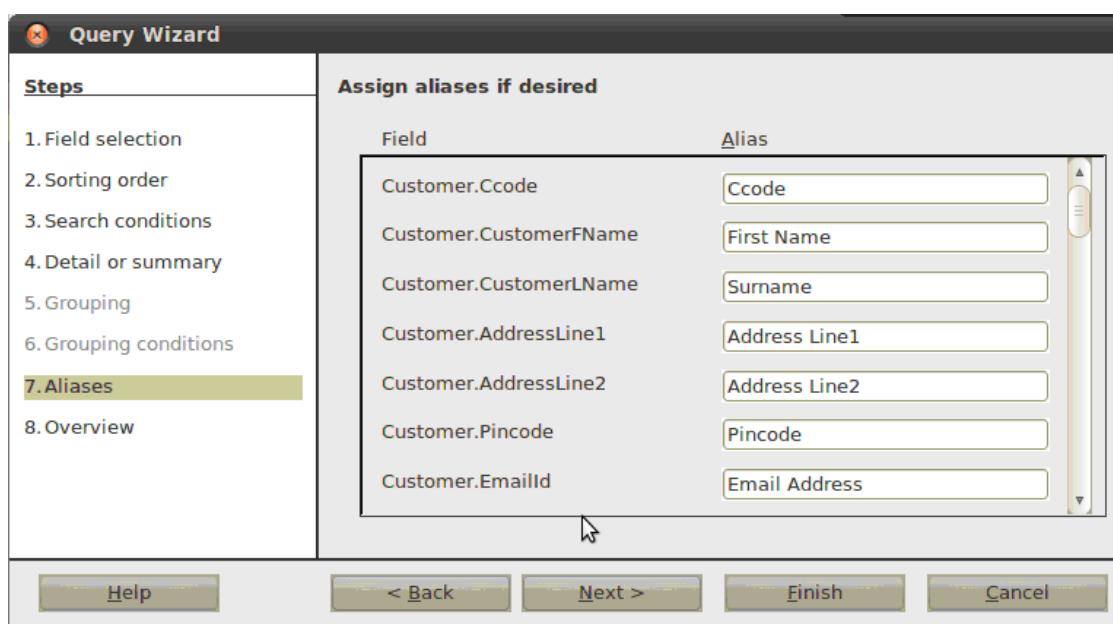


Figure 11.5 : Adding Aliases

- Finally, in the eighth step, we are given an overview of all the steps performed till now. Figure 11.6 gives us the overview of query recently created. Assign desired name to the query by typing it in text box labeled *Name of the query*, in our case we have named it as CustomerList. Take a moment to look over what we have done. In case changes are to be made, use the Back button to make the changes desired. Once the query is created, the only way to make changes is through the Design View. Two options under the question *How do you want to proceed after creating a query?* can be noticed. We can either view the query result immediately by choosing *Display Query* option, or else open it in Design View using *Modify Query* option. Design view can be used to insert additional features in query that may be beyond the capacity of the wizard.

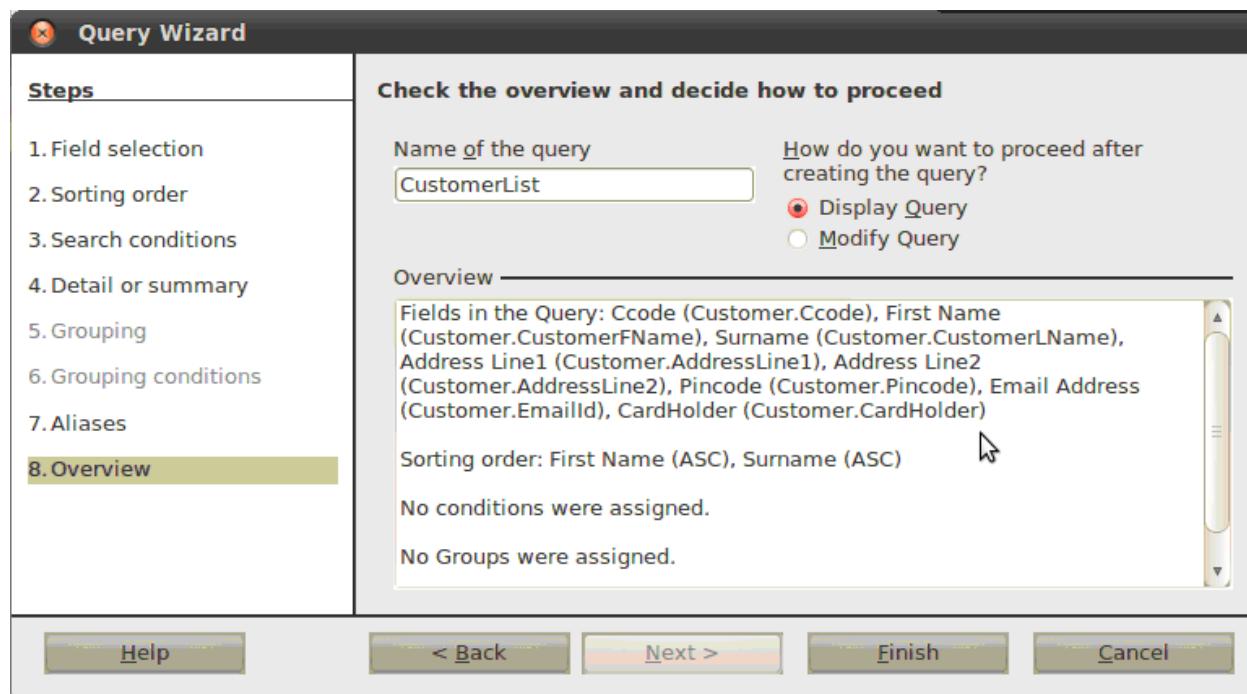


Figure 11.6 : Overview of created query

- For example, in the discussed example, where we are expecting list of customer names and addresses, we expect city name, state name and country name as well. The customer table we selected consists of only Pincode field. Details related to the Pincode field are available in other tables namely City, State and Country. However Wizard used to create a query provided us option of selecting a single table. Thus Design Views are used to create complicated queries.
- Click on Finish button and you will find that query result of the query is displayed in Data Sheet view as shown in figure 11.7.

	Ccode	First Name	Surname	Address Line1	Address Line2	Pincode	Email Address	CardHolder
▶	C05	Harshit	Shah	17, Jaldeep-1	Bopal	380058	harshit@yahoo.com	<input checked="" type="checkbox"/>
○								

Record 1 of 1

Figure 11.7 : Result of Query

Let us now calculate the amount received so far for each order. We will use query wizard once again to explore working with numeric fields. Make sure that the Queries icon is selected.

- Double click on *Use Wizard to Create Query...* option
- In step 1 of query wizard select OrderPayment table. You might have noticed that in the drop down list, the CustomerList query created recently is also included along with all the other tables. Note that Base allows us to make use of a query already created for creating another query.
- Select the OrderID and PaymentAmount fields and click on the button with greater than symbol (>) to move these fields to *Fields in the Query:* list box.
- Click on Next button.
- In step 2 select the OrderID field from the drop down box labeled as *Sort by*.
- Click on the Next button.
- As we do not want to filter records, no search criteria are to be mentioned. Hence simply skip the step 3 by clicking on the Next button. We will now see a dialog box of step 4 as shown in figure 11.8.

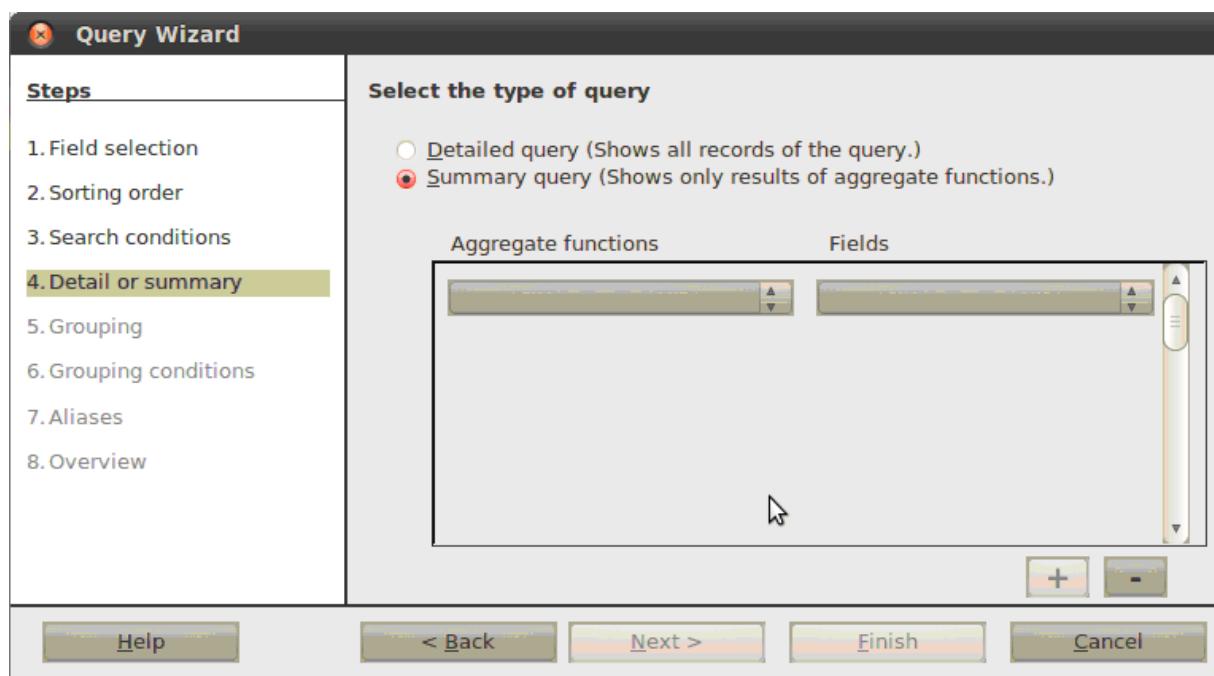


Figure 11.8 : Summarizing the result

- Select the *Summary query (Show only results of aggregate functions)* options as shown in figure 11.8.
- Click on the drop down box shown under *Aggregate functions* label. Select *get the sum of* option; similarly select the OrderPayment.PaymentAmount field from the *Fields* drop down box as we want to perform sum on the PaymentAmount field. This operation is shown in figure 11.9.

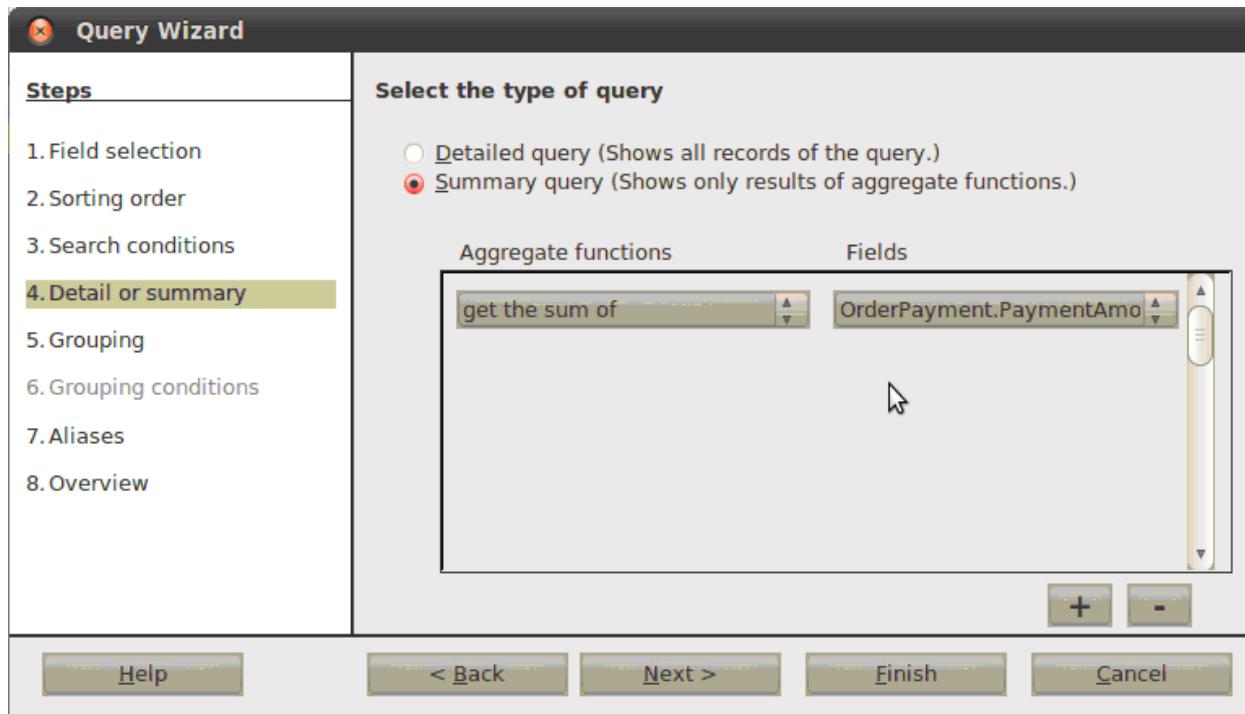


Figure 11.9 : Usage of Aggregate function

- Click on the Next button. This will lead you to step 5. Here we have to specify the fields for which groups are to be created. We may choose to perform the sum of payment amounts of all records. But it will give us total amount received from all the customers so far by Modern Electronic Store. What we want is the details of payment received for each order. Thus if manually this operation is to be performed, records are to be grouped as per the OrderID field first. This means that we need to arrange the records having OrderID value as '1' in one group, all records having OrderID value as '2' in another group and so on. Then for each of this group, the payment amount is to be added. As a result, one record pertaining to each OrderID will appear in the query result. Thus in this step as seen in figure 11.10 the OrderID field is to be mentioned in *Group by* list box.

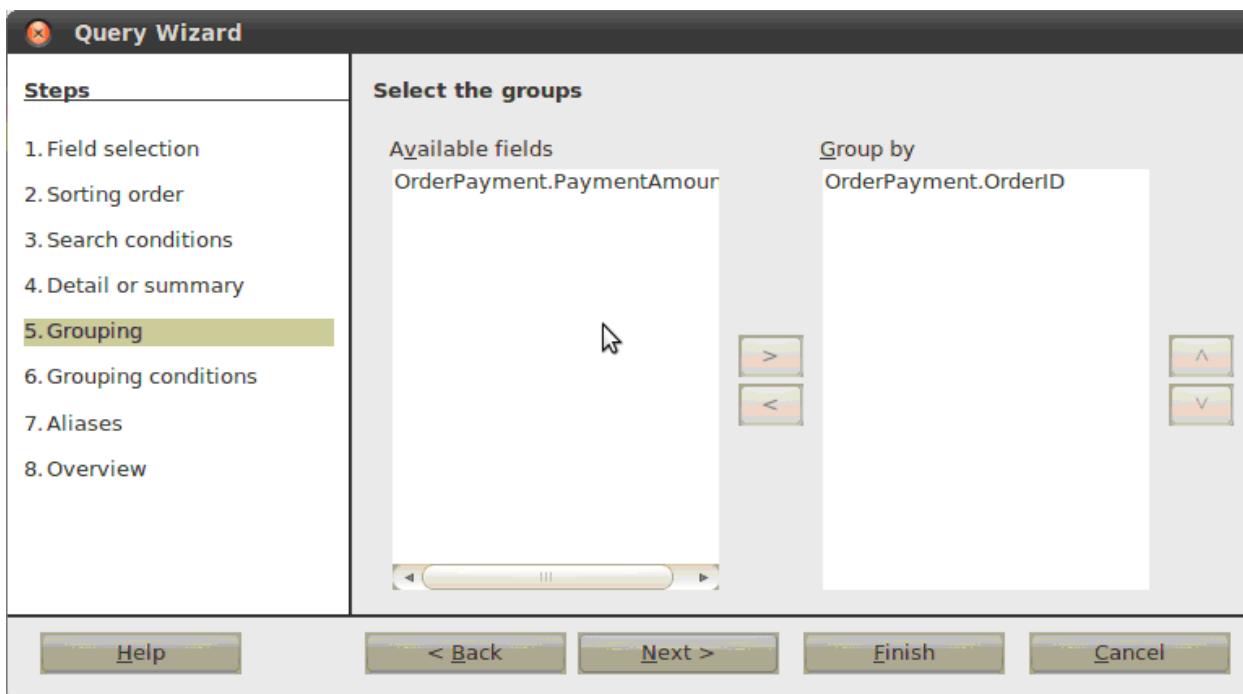


Figure 11.10 : Applying Group by on field(s)

- Click on the Next button.
- In step 6, we can specify some grouping conditions to further filter the output if required. Assume that we want to retrieve only those records where payment amount is greater than Rs. 10000. We are not applying any additional filters here, so click on Next button.
- In step 7 mention Aliases if needed and click on Next button.
- In step 8 assign it a name `Query_OrderPayment` and select the *Modify Query* option. This option is recommended here because, if directly the query is executed, Base will show only one field, `PaymentAmount` corresponding to each order. However, we expect to see the `OrderID` along with each `PaymentAmount` field.
- Click on the Finish button. You will find that the query opens in a Design View as shown in figure 11.11.

The screenshot shows the 'Design View of Query' window. At the top, there's a menu bar with 'File', 'Edit', 'View', 'Insert', 'Tools', 'Window', and 'Help'. Below the menu is a toolbar with various icons. On the left, a tree view shows the 'OrderPayment' table with its fields: OrderID, PaymentID, PaymentDate, PaymentAmount, PaymentMethod, ChequeDDNumber, BankName, and BankBranch. The main area contains a table with columns for Field, Alias, Table, Sort, Visible, Function, Criterion, and Or. The 'Field' column lists 'PaymentAmount' and 'OrderID'. The 'Alias' column has 'PaymentAmount' and '' respectively. The 'Table' column has 'OrderPayment' and 'OrderPayment' respectively. The 'Sort' column has 'ascending'. The 'Visible' column has checkboxes for 'PaymentAmount' (checked) and 'OrderID' (unchecked). The 'Function' column has 'Sum' and 'Group'. The 'Criterion' and 'Or' columns are empty.

Field	PaymentAmount	OrderID					
Alias	PaymentAmount						
Table	OrderPayment	OrderPayment					
Sort		ascending					
Visible	<input checked="" type="checkbox"/>	<input type="checkbox"/>					
Function	Sum	Group					
Criterion							
Or							

Figure 11.11 : Design View of Query

- Select the check box available under the OrderID field as shown in figure 11.12.

This screenshot is identical to Figure 11.11, but the 'Visible' column for the 'OrderID' field now has a checked checkbox, indicating it is selected for display.

Field	PaymentAmount	OrderID					
Alias	PaymentAmount						
Table	OrderPayment	OrderPayment					
Sort		ascending					
Visible	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Function	Sum	Group					
Criterion							
Or							

Figure 11.12 : Selecting field(s) for display

Click on the *Run Query* button () to execute the query. You will find that the result similar to the one shown in figure 11.13 is displayed.

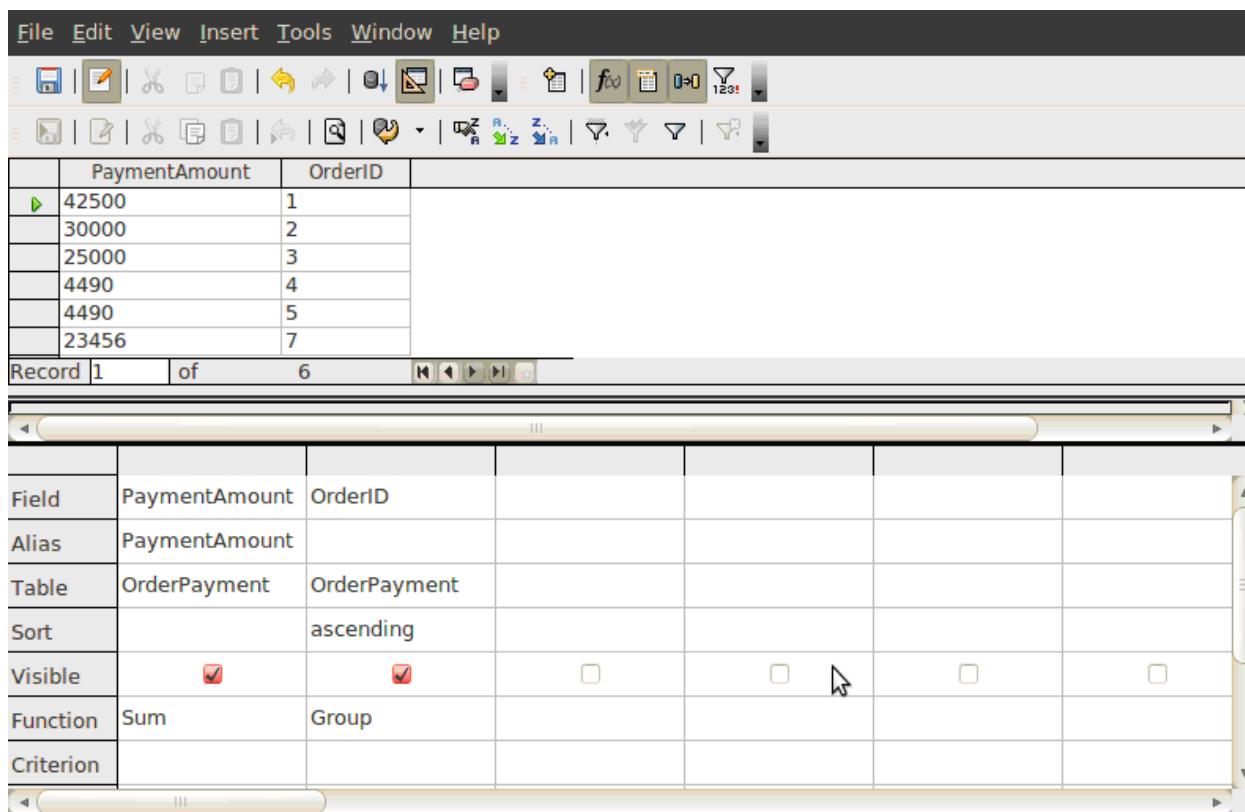


Figure 11.13 : Result of aggregation

Creation of Query Using Design View

Let us now discuss usage of Design View to create a query. While creating a query to list the customer addresses, we mentioned that it is not possible to get the details from the City, State and Country table. This is due to the limitation of the wizard as it allows us to query on a single table only. So, we will now rewrite the same query using Design View.

- Click on the *Queries* icon in the Database Window.
- Double click on the *Create Query in Design View...* option in Tasks pane.
- We will see *Add Table or Query* dialog box as shown in figure 11.14.

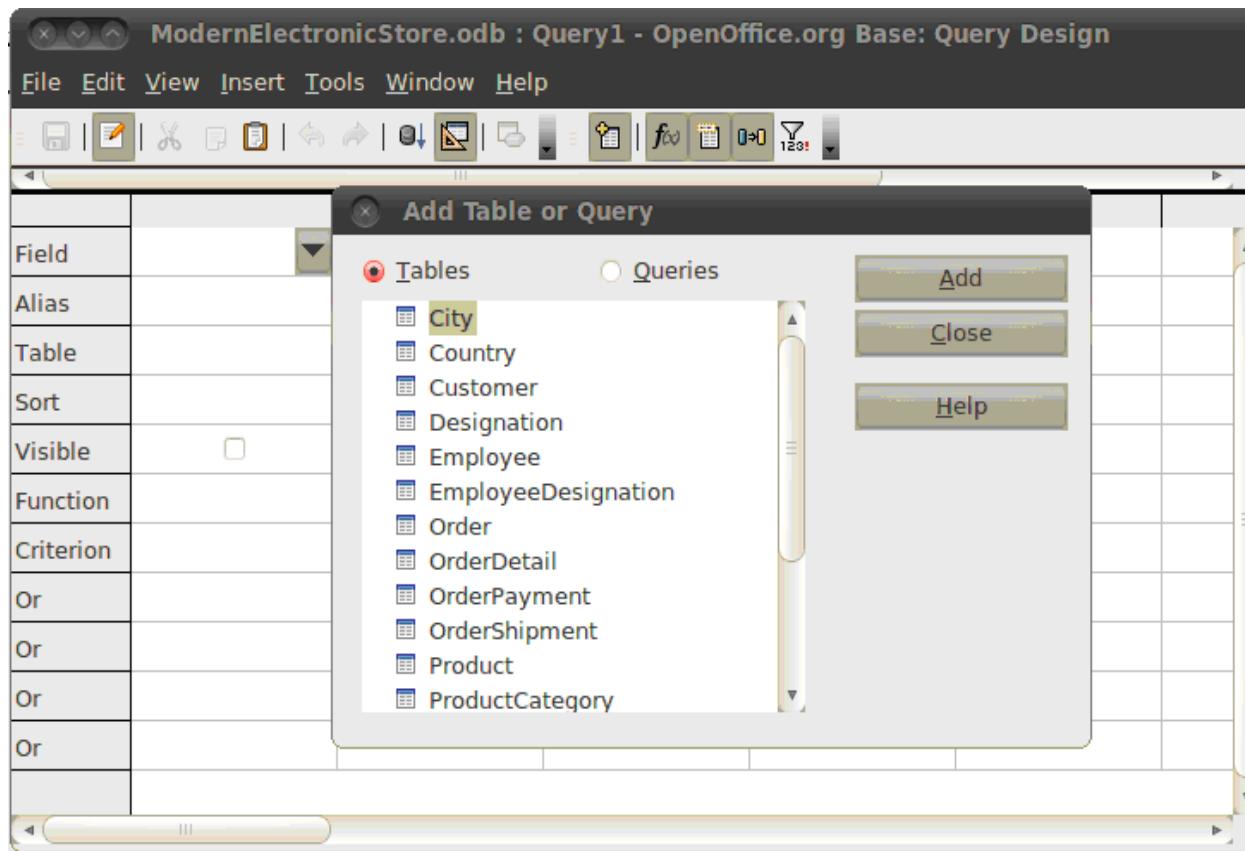


Figure 11.14 : Add Query or Table dialog box

- Select the Customer table and click on Add button.
- Similarly select City, State and Country table. You will now find four tables in Table pane as shown in figure 11.15. Base also displays the relationship, which we have earlier defined between the tables.
- Click on the Close button. If you want to add some more tables, the you can open the *Add Table or Query* dialog box again by clicking on Add Table or Query button (⊕) on Query design toolbar.
- Double click on CustomerFName, CustomerLName, AddressLine1, AddressLine2 fields from the Customer table. Similarly select City field from City table, StateName field from State table, Pincode field from City table and CountryName field from Country table. The field names along with their respective table names will be displayed in grid as shown in figure 11.15.

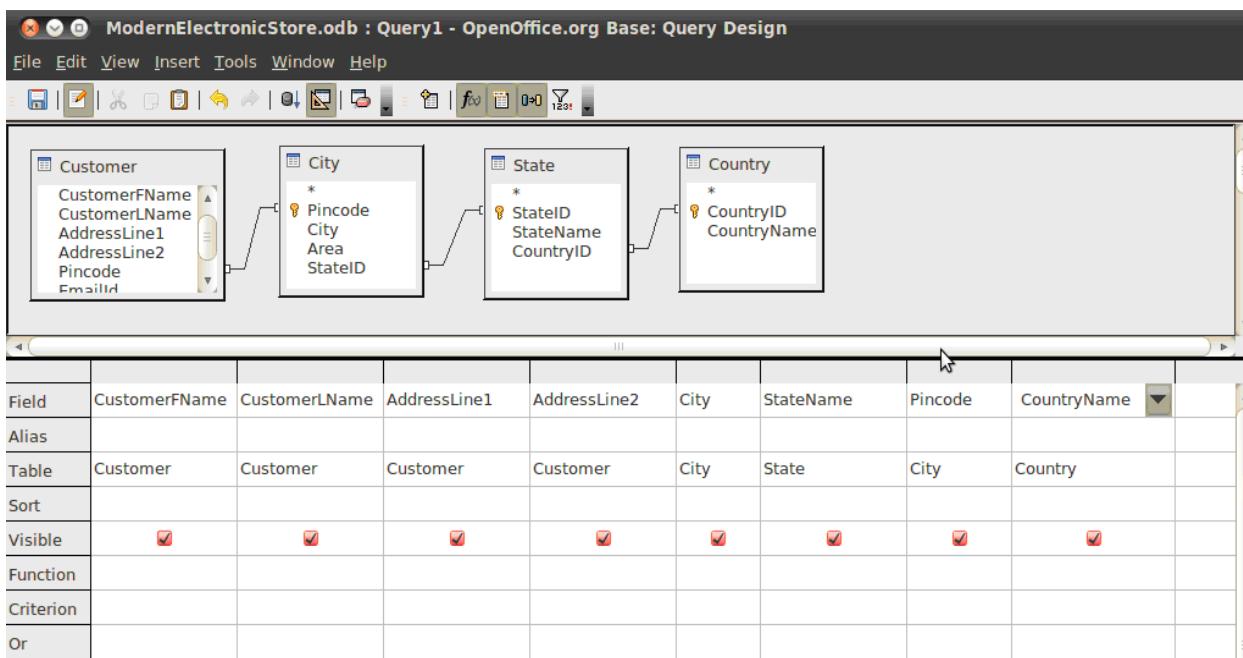


Figure 11.15 : Selection of fields

- Observe that we are also able to see some record (row) headings like Alias, Sort, Visible, Function, Criterion and Or. You might have noticed that by default visible option for each field is set to true. It indicates that all selected fields will be displayed in the output.
- Alias can be used for displaying meaningful names for the fields. For example, in place of CustomerFname, we would prefer to use *Name of Customer* as column title in the query result. Type Name of Customer in the text box visible after the row heading *Alias* under the CustomerFName column.
- To display customer records in alphabetical order of his/her names, select *ascending* from drop down box visible after the row heading *Sort* under the CustomerFName column. Similarly, select *ascending* in the *Sort* option under the CustomerLName column.
- Click on the Run Query button () on the Query Design toolbar. Query result similar to the one shown in figure 11.16 would be displayed.
- To save a query for later use, select the Save option from the File menu. Alternatively, click on the Close button and Base will display a Save dialog box.
- Type desired name, for example *CustomerAddresses* and click on OK button.

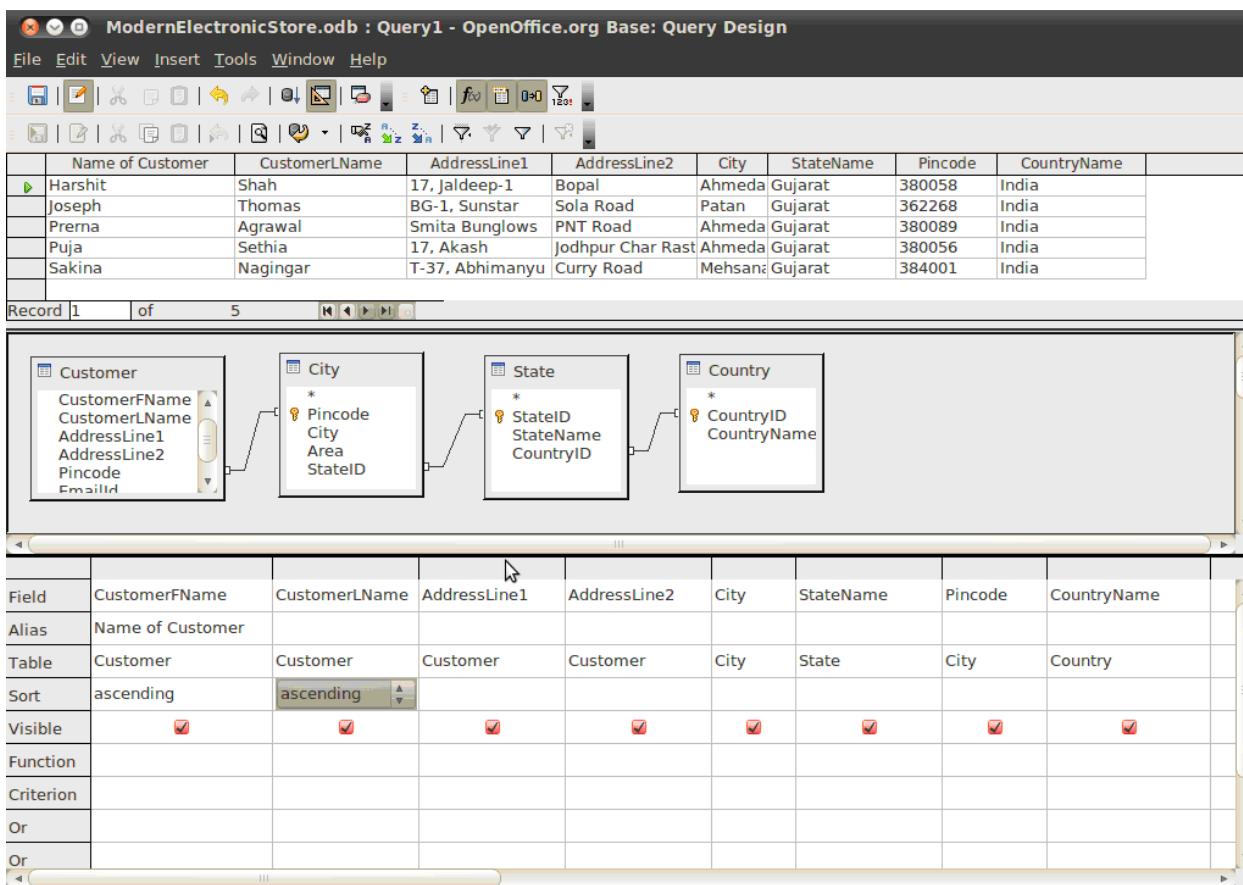


Figure 11.16 : Query output

Close the query window once you have observed the output.

Editing a Query

After creating a query, one may like to change a query. For example, in the query created above, we would like to add *Surname* as an alias in CustomerLName column.

To make this change perform the following steps:

- Click on *Queries* icon. Right click on the query CustomerAddresses, from the popup menu choose *Edit* option. This will display the query in Design View.
- Type *Surname* in the text box visible after row heading *Alias* under the CustomerLName column.
- Run Query.

Applying Criteria

We have seen that we can write a query, which displays selected fields of a table. Now suppose, instead of viewing all records we wish to view the details of customers residing in the city of Ahmedabad. This means we want Base to display a subset of selected records. To do this, we can specify a criterion that limits the records to only those where the City field contains “Ahmedabad” as a value.

Using Single Field

- Right click on the CustomerAddresses Query.
- Click Edit option to open the query in Design View.
- In the *Criterion* cell of the City field type “Ahmedabad” as shown in figure 11.17. Note that text must be enclosed within a quotation (?) ’) delimiter; date must be enclosed in the hash (#) delimiter while the number literals do not need any delimiters. If we miss to put delimiters, Base will not report any error; instead will apply delimiters on its own.
- Save and Run the query and you will find the desired result.

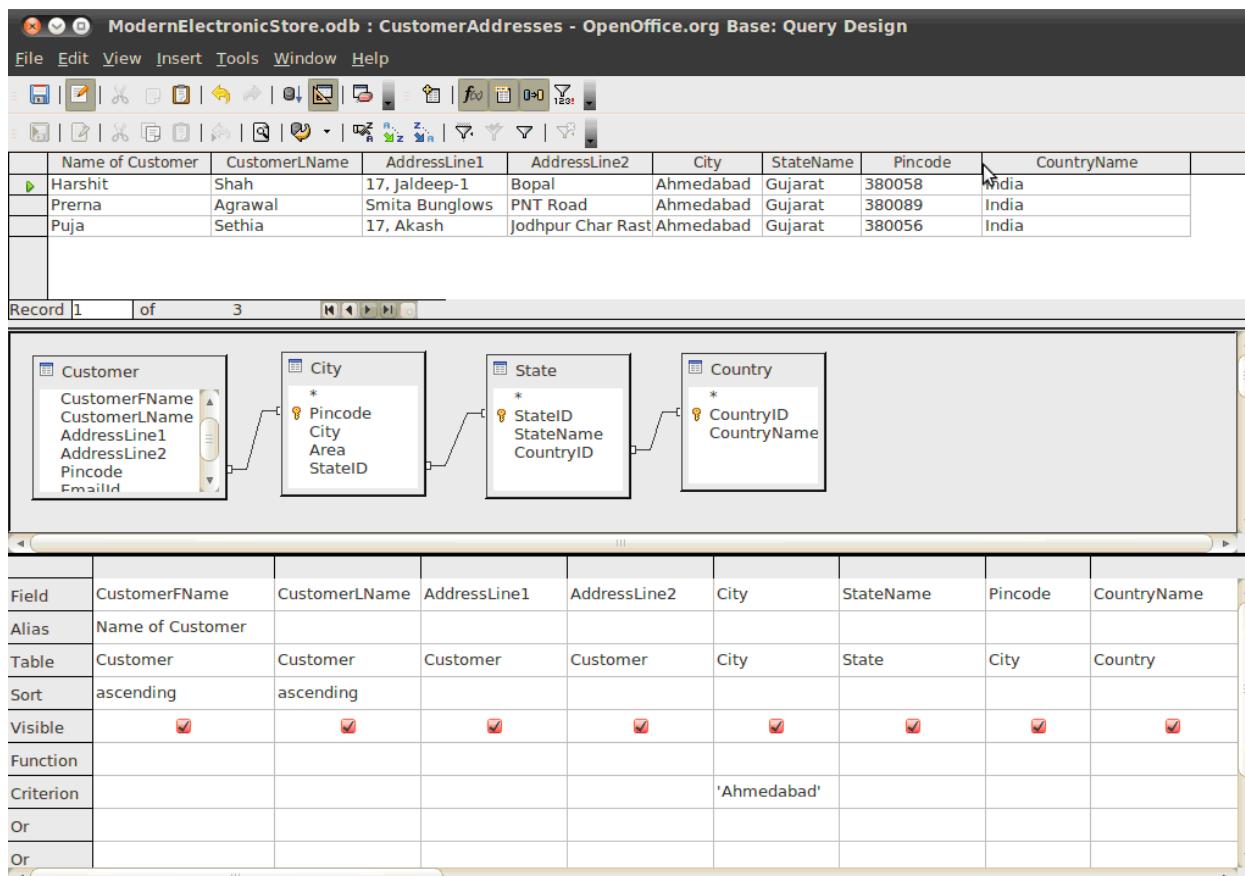


Figure 11.17 : Setting Criteria

Apart from the constant values used as shown in figure 11.17, Base also allows us to design expression for defining criteria using different types of operators. The operator list is as shown in table 11.1.

Operator	Symbols
Comparison	=, >, <, >=, <=, <>
Logical	And, Or, Not
Special	Like, Is, Between, In

Table 11.1 : Operators used in Base

Suppose we want to display the list of employees who joined after 1st June 2011. Then create a new query in Design View. Add the Employee table. Then select fields FirstName, LastName and JoiningDate. Type “> #01/06/2011#” in the JoiningDate fields *Criterion* cell. De-select the check box displayed in the *Visible* cell under the JoiningDate column. Now Save and Run the query. The output will be similar to the one shown in figure 11.18. Observe that the JoiningDate field is not displayed.

Note : When you do not want to display any field used in the query in the output, clear the check mark shown in the *Visible* property of that field.

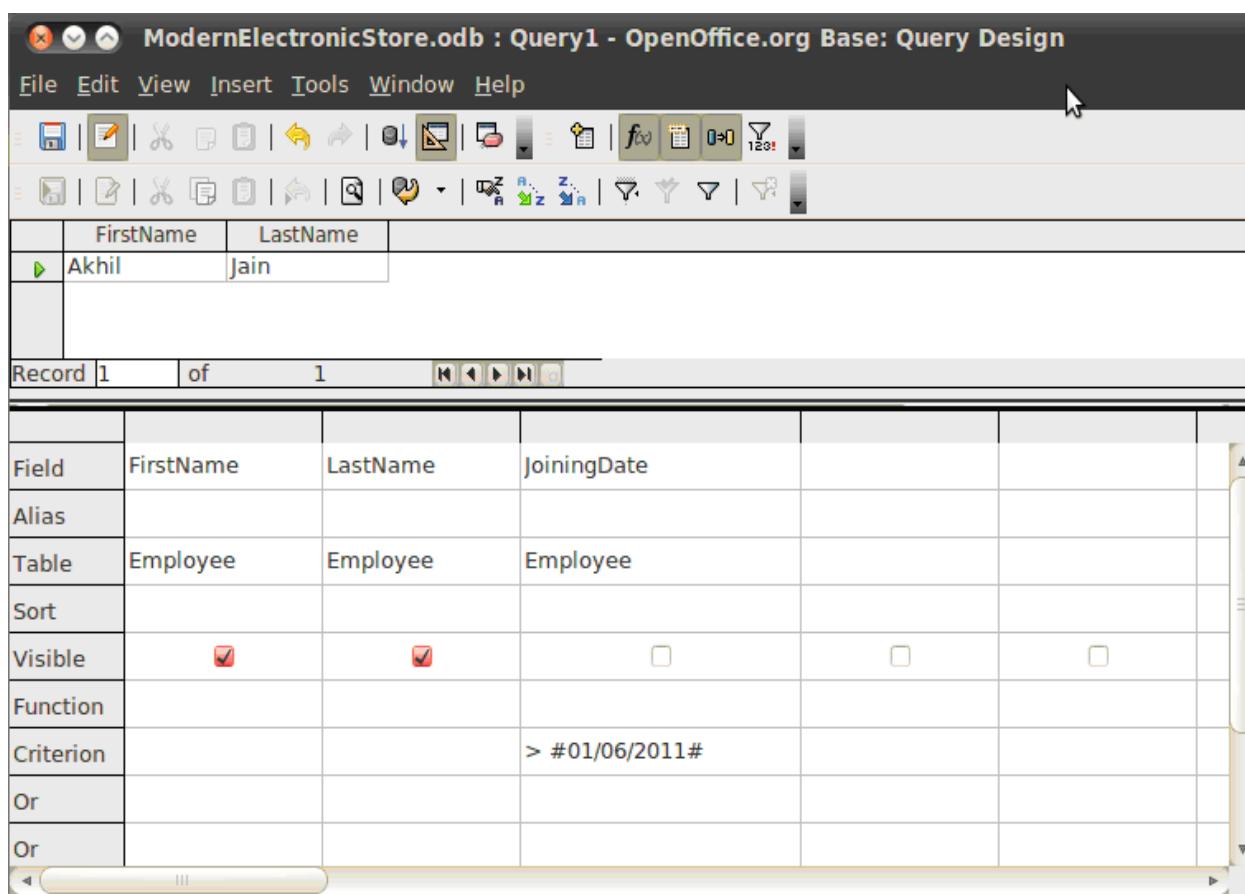


Figure 11.18 : Applying Criteria in Date Field

Similarly to display employees who joined between 1st June 2005 and 1st Nov 2012, The *Criterion* in the JoiningDate field can be set as “ >= # 1/6/2005 # And <= # 11/1/2012 #”. Base also offers *Between* operator to specify the same criteria as shown in figure 11.19.

ModernElectronicStore.odb : Query1 - OpenOffice.org Base: Query Design

File Edit View Insert Tools Window Help

The screenshot shows the OpenOffice.org Base Query Design interface. At the top, the title bar reads "ModernElectronicStore.odb : Query1 - OpenOffice.org Base: Query Design". Below the title bar is a menu bar with File, Edit, View, Insert, Tools, Window, and Help. Underneath the menu bar is a toolbar with various icons for database operations. The main area displays a table with three records:

	FirstName	LastName	JoiningDate
▶	Anya	Gangele	01/05/11
	Robina	Arora	02/06/06
	Akhil	Jain	12/31/11

Below the table is a status bar showing "Record 1 of 3". Underneath the table is a query builder panel with the following fields:

- Field:** FirstName, LastName, JoiningDate
- Alias:** (empty)
- Table:** Employee, Employee, Employee
- Sort:** (empty)
- Visible:** checked for FirstName, LastName, JoiningDate; unchecked for others
- Function:** (empty)
- Criterion:** Between #1/6/2005 # And # 11/1/2012 #
- Or:** (empty)

Figure 11.19 : Using the Between Operator

Now, suppose that we want to send discount coupons to customers who live in the city of Ahmedabad and Patan. We need a list of customers residing either in the city of Ahmedabad or Patan. Then create a new query in Design View. Add the Customer, City and State tables. Then select CustomerFName, CustomerLName, AddressLine1, AddressLine2, City, StateName and CardHolder fields from the tables as shown in figure 11.20. Type the criterion as can be seen in the City field. Now Save and Run the query. The output will be similar to the one shown in figure 11.20.

File Edit View Insert Tools Window Help

The screenshot shows the OpenOffice.org Base Query Design interface. At the top, the title bar reads "ModernElectronicStore.odb : Query1 - OpenOffice.org Base: Query Design". Below the title bar is a menu bar with File, Edit, View, Insert, Tools, Window, and Help. Underneath the menu bar is a toolbar with various icons for database operations. The main area displays a table with four records:

	CustomerFName	CustomerLName	AddressLine1	AddressLine2	City	StateName	CardHolder
▶	Puja	Sethia	17, Akash	Jodhpur Char Rast	Ahmeda	Gujarat	<input type="checkbox"/>
	Joseph	Thomas	BG-1, Sunstar	Sola Road	Patan	Gujarat	<input type="checkbox"/>
	Prerna	Agrawal	Smita Bunglows	PNT Road	Ahmeda	Gujarat	<input type="checkbox"/>
	Harshit	Shah	17, Jaldeep-1	Bopal	Ahmeda	Gujarat	<input checked="" type="checkbox"/>

Below the table is a status bar showing "Record 1 of 4". Underneath the table is a query builder panel with the following fields:

- Field:** CustomerLName, AddressLine1, AddressLine2, City, StateName, CardHolder
- Alias:** (empty)
- Table:** Customer, Customer, Customer, City, State, Customer
- Sort:** (empty)
- Visible:** checked for CustomerLName, AddressLine1, AddressLine2, City, StateName, CardHolder; unchecked for others
- Function:** (empty)
- Criterion:** 'Ahmedabad'
- Or:** 'Patan'
- Or:** (empty)

Figure 11.20 : Using Multiple Criteria on single field

Criteria for the above query can also be specified using the *IN* operator. Type *IN* ('Ahmedabad'; 'Patan') in the *Criterion* row of the City field and you will get the same result. You can use *NOT IN* ('Ahmedabad'; 'Patan') in *Criterion* row of the City field to retrieve records of customers from the all other places except Ahmedabad or Patan.

Using Multiple Fields

Now suppose we want to send discount coupons only to card holders of Modern Electronic Store residing in Ahmedabad or Patan. For solving this problem, we need to use the *AND* operator which specifies that City should be (either Ahmedabad or Patan) and (CardHolder field should not be empty). Here, we need to apply *OR* criteria within field and *AND* criteria between fields. Type the criteria as shown in the figure 11.21. Here the value NULL (No Check mark in CardHolder field) signifies that field is empty.

The screenshot shows the OpenOffice.org Base Query Design window. At the top, the title bar reads "ModernElectronicStore.odb : and_query - OpenOffice.org Base: Query Design". Below the title bar is a toolbar with various icons for database operations. The main area displays a table of customer data:

	CustomerFName	CustomerLName	AddressLine1	AddressLine2	City	StateName	CardHolder
Puja	Sethia	17, Akash	Jodhpur Char Rast	Ahmeda	Gujarat	<input type="checkbox"/>	
Joseph	Thomas	BG-1, Sunstar	Sola Road	Patan	Gujarat	<input type="checkbox"/>	
Preerna	Agrawal	Smita Bunglows	PNT Road	Ahmeda	Gujarat	<input type="checkbox"/>	
Harshit	Shah	17, Jaldeep-1	Bopal	Ahmeda	Gujarat	<input checked="" type="checkbox"/>	

Below the table, the status bar shows "Record 1 of 4".

Below the table is the query builder interface, which includes a toolbar and a detailed configuration pane:

Field	CustomerLName	AddressLine1	AddressLine2	City	StateName	CardHolder
Alias						
Table	Customer	Customer	Customer	City	State	Customer
Sort						
Visible	<input checked="" type="checkbox"/>					
Function						
Criterion				IN ('Ahmedabad'; 'Patan')		Is Not Empty
Or						
Or						

Figure 11.21 : Using Criteria on multiple fields

Now, if you want to view the list of customers who can belong to either Ahmedabad or Patan or has Membership Card. Then type *IN* ('Ahmedabad', 'Patan') in the *Criterion* row of the City column. Also type *Is Not Empty* in the *Or* row of the CardHolder column (See figure 11.22).

Note the difference between previous query and this query. In the first case we wrote both the expressions in same row while applying the AND condition between two fields. While in this

query, the OR condition between two fields is written in separate rows. Observe that in the result set as shown in figure 11.22, a new record of a customer from Mehsana is also listed as the customer is also a membership card holder.

The screenshot shows the 'Query Design' window for 'ModernElectronicStore.odb'. The title bar says 'ModernoElectronicsStore.odb : or_query - OpenOffice.org Base: Query Design'. The menu bar includes File, Edit, View, Insert, Tools, Window, Help. The toolbar has various icons for creating, modifying, and deleting queries. The main area displays a table with columns: CustomerFName, CustomerLName, AddressLine1, AddressLine2, City, StateName, CardHolder. The data shows records for Puja, Sakina, Joseph, Preerna, and Harshit. Below the table is a status bar showing 'Record 1 of 5'. The bottom half of the window is the 'Query Definition' pane, which lists fields, aliases, tables, sort criteria, visibility, functions, and a criterion section. In the 'Criterion' section, there are two rows under 'Or': 'IN ('Ahmedabad'; 'Patan')' and 'Is Not Empty'.

Figure 11.22 : Applying OR Criteria in multiple fields

Using Wild Cards

Suppose we want to see the list of products with their model names starting with character set *hp*. Then create a new query using table Product. Select fields Pcode, ModelName, SellingPrice and OSSupport. Set the criterion as shown in figure 11.23.

The screenshot shows the 'Query Design' window for 'ModernElectronicStore.odb'. The title bar says 'ModernoElectronicsStore.odb : wildcardcharacter - OpenOffice.org Base: Query Design'. The menu bar and toolbar are similar to Figure 11.22. The main area displays a table with columns: Pcode, ModelName, SellingPrice, OSSupport. The data shows two records for 'hp ENVY Sleekbook 6t 1000'. Below the table is a status bar showing 'Record 1 of 2'. The bottom half of the window is the 'Query Definition' pane. In the 'Criterion' section, there is one row: 'LIKE 'hp*''.

Figure 11.23 : Wild Cards

The asterisk symbol (*) used in expression in *Criterion* cell of ModelName field in figure 11.23 is known as wild card character. A wild card is a symbol that represents any character or combination of characters. Thus ‘hp*’ represents a word whose first two alphabets are hp, followed by any group of characters. Similarly, the criterion ‘Like *hp’ will display products with names ending with alphabets ‘hp’ and ‘Like hp*hp’ will display products with names starting and ending with alphabets ‘hp’. Note that you must include the Like operator with the wild card characters.

Performing Calculations

Calculations within a query in Base can be performed using one of the following ways:

- Custom calculation
- Predefined calculation

Custom calculation: It includes performing numeric, date, and text calculations on each record using data from one or more fields. Calculations include operations like add, multiply, subtract or divide the values in two different fields. To perform custom calculation we need to add an extra field known as calculated field.

Let us discuss it with an example. Look at the OrderDetail table. If we want to know total amount pertaining to each product purchased by a customer, we can calculate it. The value of Amount is equal to the Quantity multiplied by SalePrice. The new field that stores information about Amount is known as calculated field.

Perform the following steps to find out total amount paid by each customer in each month.

- Create a new query using Design View.
- Select the OrderDetail table from *Add Table or Query* dialog box.
- Double click on the OrderID field to include it in design grid.
- Type Quantity * SalePrice in the *Field* row of the second column in the query design grid. You may not be able to see the entire entry because the *Field* row is not large enough.
- Right click the *Field* row in the second column in the design grid and then select *Column Width* from the popup menu displayed as shown in figure 11.24 would be displayed. Specify 4.20 cm as width to make text in the *Field* row visible.
- In the *Alias* row under the second column, type *Amount*. (See figure 11.24)

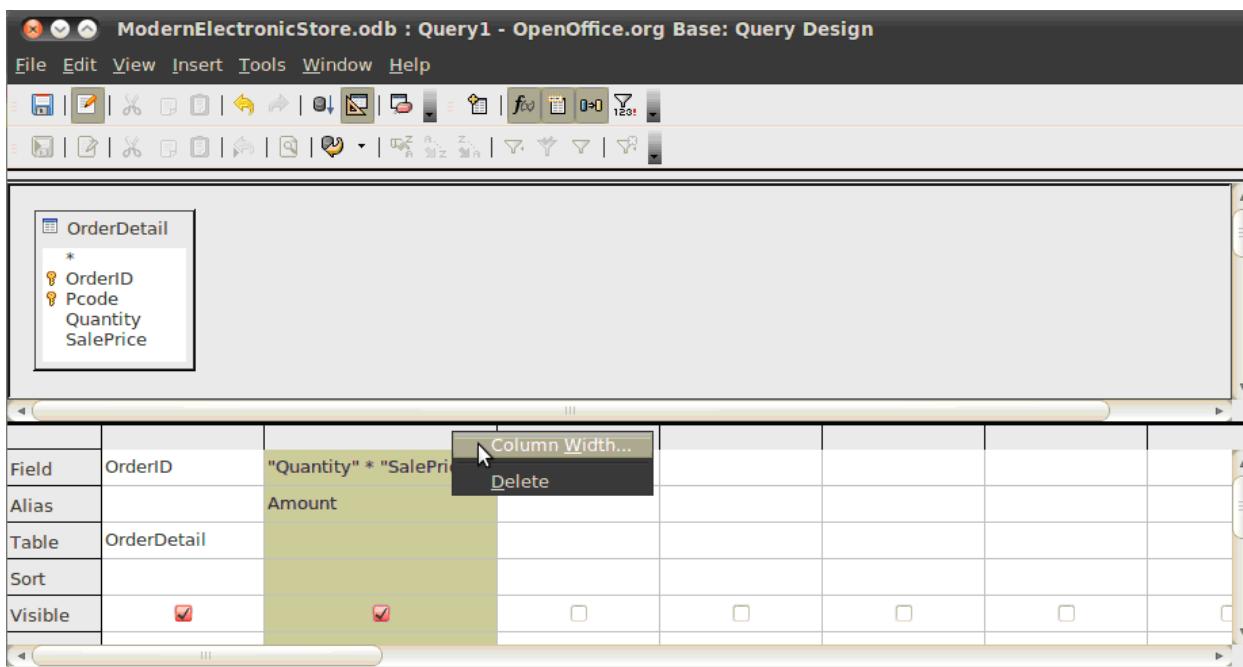


Figure 11.24 : Using Calculated Fields

Predefined calculations (Summarizing the data): We have seen how to perform calculation on fields. Many times we need to perform calculations on group of records rather than on fields. For example, finding the total number of products, or computing average amount spent by each customer, we need to perform summary calculation. Base provides some predefined calculations to compute sum, average, count, minimum, maximum, standard deviation, or variance on group of records. These calculations differ from calculated fields as they are applied on multiple records within a table resulting in a single value. Let us design a query for finding total number of customers of Modern Electronic Store by performing the following steps:

- Click on *Queries* icon and select *Create Query in Design View...*
- Select the Customer table from *Add Table or Query* dialog box.
- Double click on the *Ccode* field.
- Type “Total Customers” in *Alias* row.
- In a row with a label *Function* in query design grid, open the drop down menu.
- Select the *Count* function from the list of aggregate functions as shown in figure 11.25.
- Run the query and you will get the total number of customers of Modern Electronic Store.

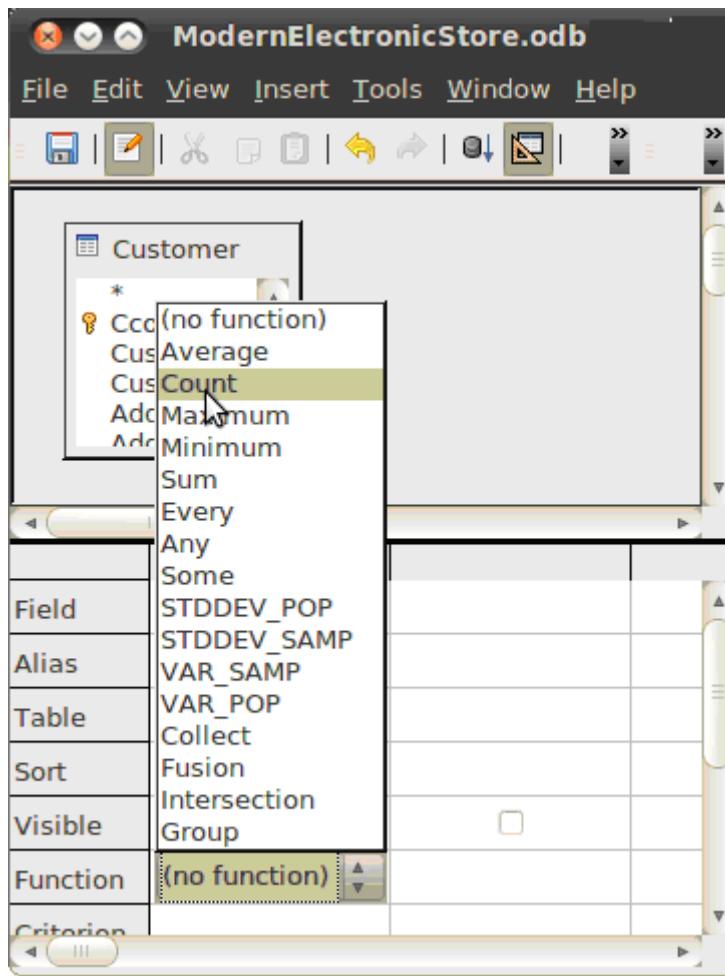


Figure 11.25 : Using Aggregate Function

Remember that the results of a calculation in a field are only for display purpose. These results aren't actually stored in the underlying table. Instead, Base performs the calculation each time you run the query. Thus a result generated from such queries is always based on the most current data available in the database.

Grouping the Data

Earlier, we have seen how to retrieve the list of customers residing in the city of Ahmedabad or Patan. Here we provided customer details, which proved useful in sending discount coupons. Now the manager has another query. "Can I have total number of customers residing in each of these cities". Total should be corresponding to each city mentioned in the Customer table. You might think that yes, we can apply total on the CustomerFname field which will give you count. Try it and view the result. You will find that you are wrong!!!. The count you compute as a result is total number of customers residing in Ahmedabad or Patan. Fine, you can say that I will apply the count function on the City field. Try it and again you will end up with incorrect result.

Take pen and paper. Now start thinking how it is possible. Identify the cities of customers. It may be Ahmedabad, Mehsana or Patan etc. Make a group of customers in Ahmedabad, customers in Patan, and similarly for all cities. Now apply total on each group and you will get the result, which

your manager requires. You might think that it's really complex. Instead it is very easy when you do it with Base. Perform the following steps:

- Open a new query in Design View.
- Add the Customer table and the City table from *Add Table or Query* dialog box.
- Double click on the City field from the City table. Similarly select the Ccode field from the Customer table.
- In the *Function* row of the City field, select *Group By* as seen in figure 11.26.
- In the *Function row* of Ccode field, select *Count*.
- Run the query and desired result will be displayed.

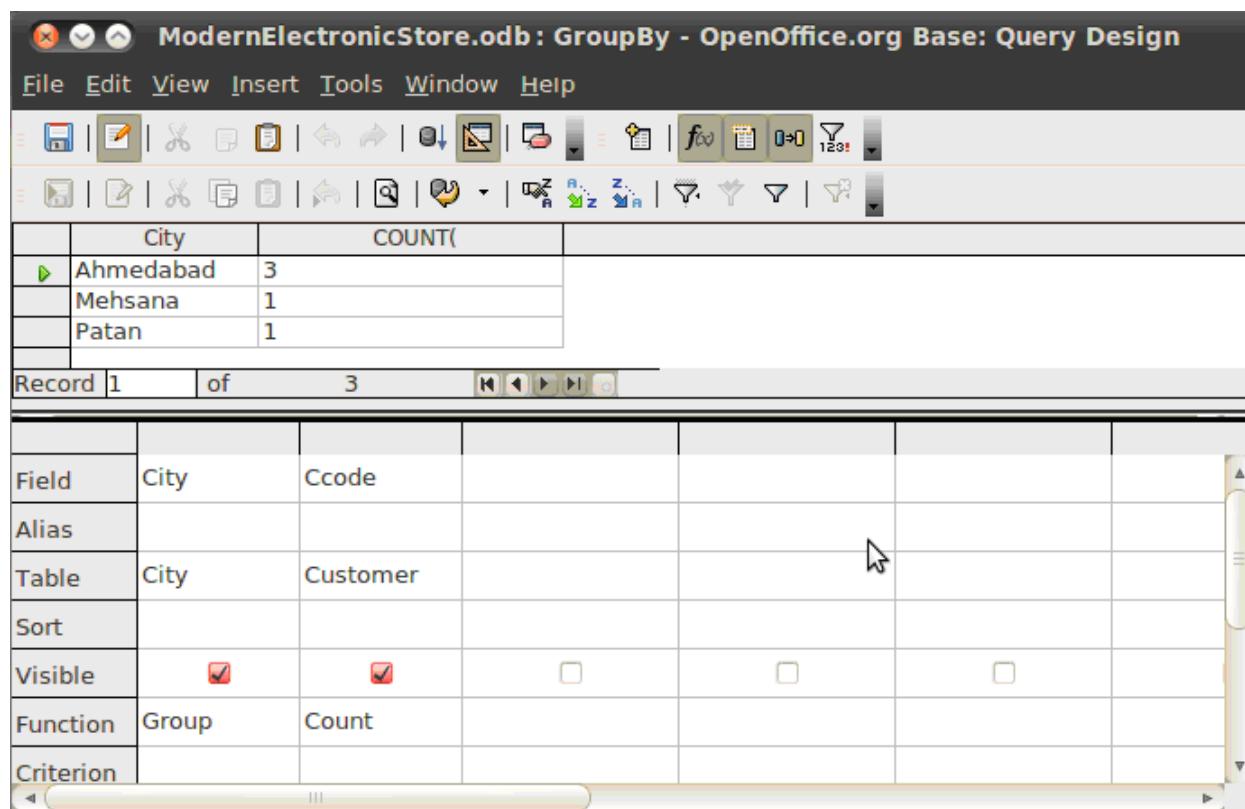


Figure 11.26 : Grouping the Records to summarize result

Parameter Query: Parameter queries are designed to accept values from the user at run time. Till now the queries that we created used fixed criterion. The criterion once defined will not be changed for every execution of the query. The output of the query may though vary depending on the current data in the table. Generally when we run a parameter query it will display a dialog box asking the user to enter the values of the parameter. These values are then assigned as criterion values for retrieving the data.

Let us design a query to display the detail of laptops available at Modern Electronic Store. Following steps when used will give us the desired result.

- Open new query in Design View.
- Add the Product and ProductCategory table.
- Double click on * visible in the Product table. It will add all the fields of the Product table in the query.
- Double click on the CategoryName field from the Category table.
- Type *Laptop* in *Criteria* cell of the CategoryName field.
- Save the query with name *DetailsOfLaptops*.

Now, suppose you are asked to retrieve the details of smart phones. You need to design a new query with *Smart Phone* as criteria in the CategoryName field and save it as *DetailsofSmartphones*. What if you are asked the same question for all the product categories? You need multiple queries!!!.

Base has one more interesting and very useful feature to help us in this type of situations. We can create a parameter query. To create a Parameter Query, the design of the query will remain same, but we need to enter the parameter, rather than specifying the actual value in the *Criterion* cell. When we run the query, Base will display a dialog box that will prompt us to enter the value of the parameter specified. Perform the following steps to create a parameter query for listing out the different product categories.

- Open a new query in Design View.
- Add the Product and ProductCategory tables.
- Double click on Pcode from the Product table.
- Double click on the CategoryName field from the Category table.
- Type *:CategoryName* in the *Criterion* cell of the CategoryName. The query will look as shown in figure 11.27. Note that the criterion parameter must be preceded by a colon symbol (:).
- Click on the Run button to view query results. Base will display dialog box as shown in figure 11.28.
- Type *Laptop* in the text box under label *Value* and click on OK button. You will get the list of laptops.

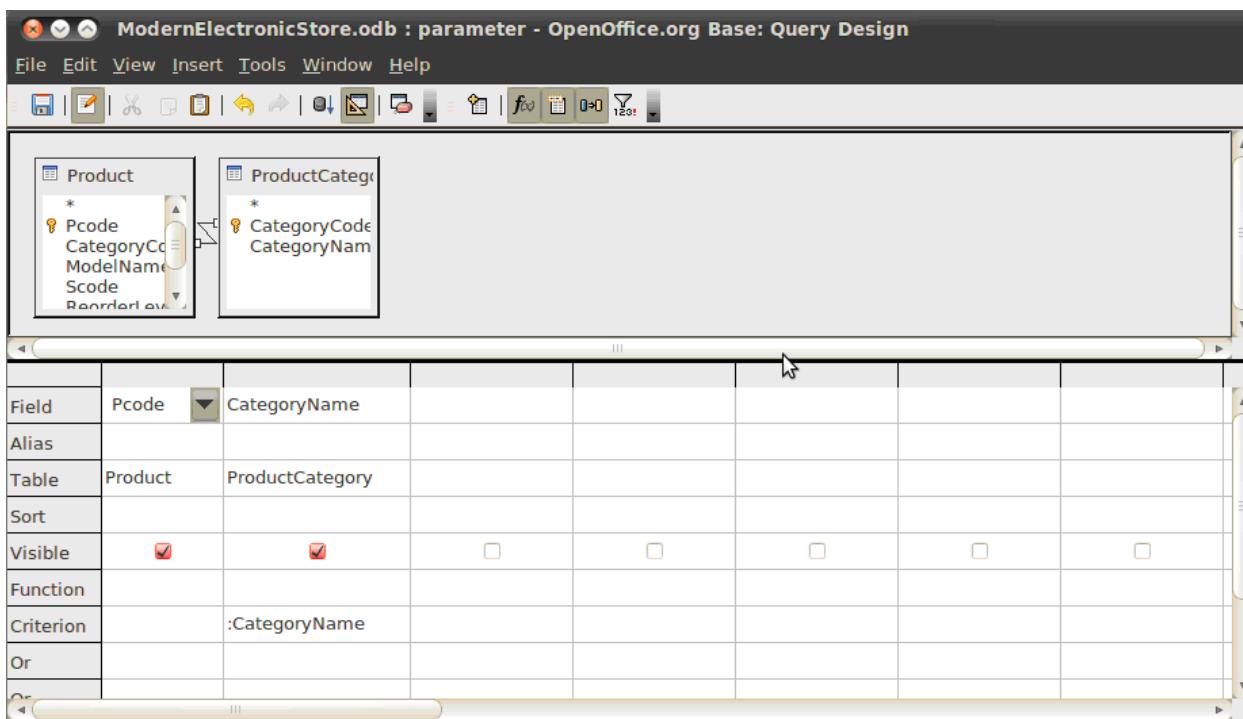


Figure 11.27 : Parameterized Query



Figure 11.28 : Parameter value

Try to execute this query again with different values of the product categories and observe the output.

Structured Query Language

By now we have seen that there is always more than one way to do the same task in Base. For example, *Create Table in Design View...* and *Use Wizard to Create Table...* both options allow us to create table of a database. The difference is that the wizard makes the task easier while Design View gives more flexibility. We still have a third option for creating table; the SQL commands, which gives us the most flexibility and control.

SQL stands for Structured Query Language. It is a standard language used to query a relational database. The SQL queries are in the form of statements. In earlier chapters we have seen how to create a table, insert a data into it, edit and delete the data in the tables. All these operations can also be done using SQL statements. Let us create a new table using the SQL statement.

Click on the Tools options in the menu bar and then select the *SQL...* option. The *Execute SQL Statement* dialog box will get opened with a cursor blinking in the text box under the label *Command to Execute*. We can type the instructions to perform different operations related to table or a query here. Let us try to create a table named Scheme that has four fields SchemeID, StartDate, EndDate and Description. Pay attention to quote signs, capitalization and syntax to avoid errors in execution of queries. Type the statement shown below in the text box under the label *Command to Execute*. (See figure 11.29)

```
CREATE TABLE "Scheme"  
("SchemeID" INTEGER NOT NULL PRIMARY KEY,  
"StartDate" DATE,  
"EndDate" DATE,  
"Description" VARCHAR(25));
```

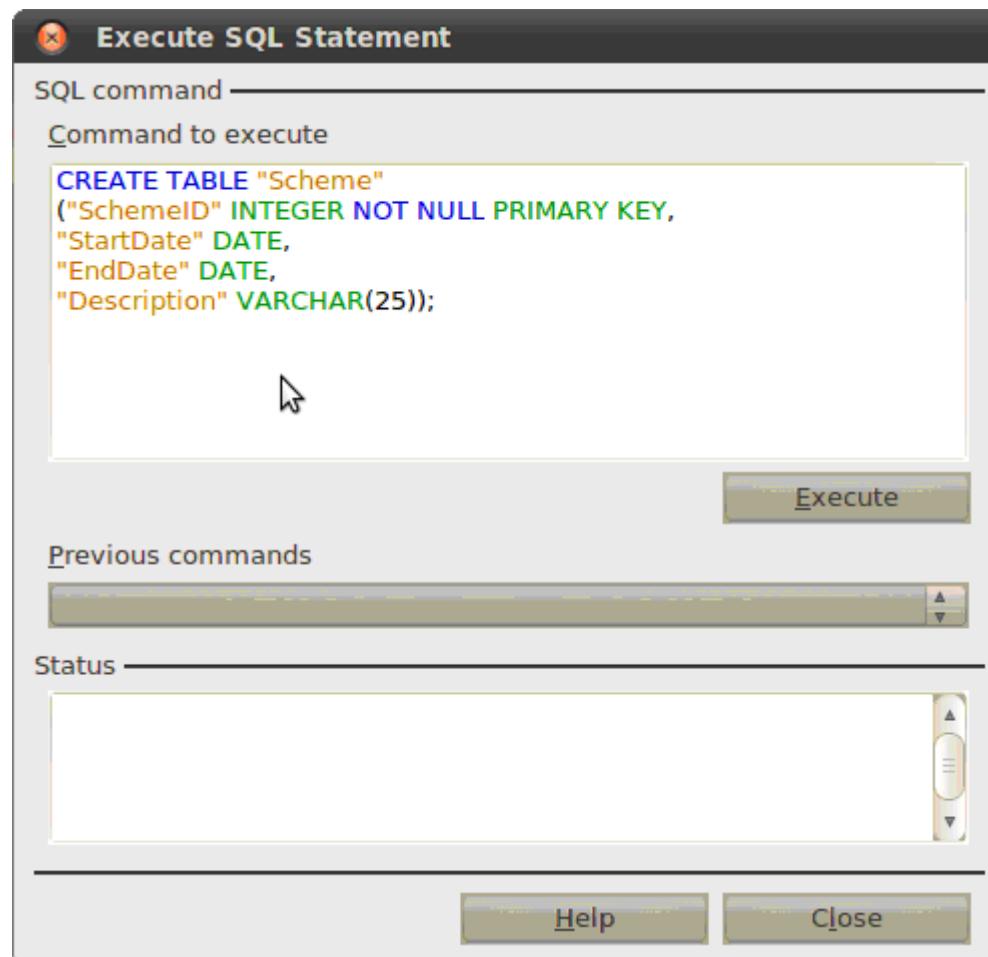


Figure 11.29 : SQL Command Window

When you enter the instructions, the *Execute* button gets highlighted. When you are finished with writing the query, click on it. This will run the written SQL command and the table would be created. After a few seconds, the window will inform that the instructions have been executed. Other than that there will be no visible output on your screen. If you now go to the *View* menu and click on *Refresh Tables*, a complete list of the tables we have created will appear in the lower section of your screen including the new table recently created.

At times we may need to remove the table from a database. The **DROP TABLE** statement is used to remove a table. For example, if we want to drop the Scheme table created recently, then in the text box under the label *Command to Execute* write the following statement:

DROP TABLE Scheme IF EXISTS;

Now click on the Execute button and you will see that the Scheme table is removed from the database.

We can use SQL statement to retrieve information from the table. But we need to use different Window for this purpose. To open this Window click Queries icon. Three options will appear in the top panel under the label *Tasks*. Select the *Create Query in SQL view...* option. This will open a dialog box with a blinking cursor as shown in figure 11.30. We can now type the required SQL statement here to retrieve the information. The SQL statements to retrieve information start with **SELECT** keyword and are also known as SQL Queries. Figure 11.30 shows an example of SQL query.



Figure 11.30 : SQL Select Query Window

Let us try to get the output of the query **SELECT * from Employee** visible in figure 11.30.

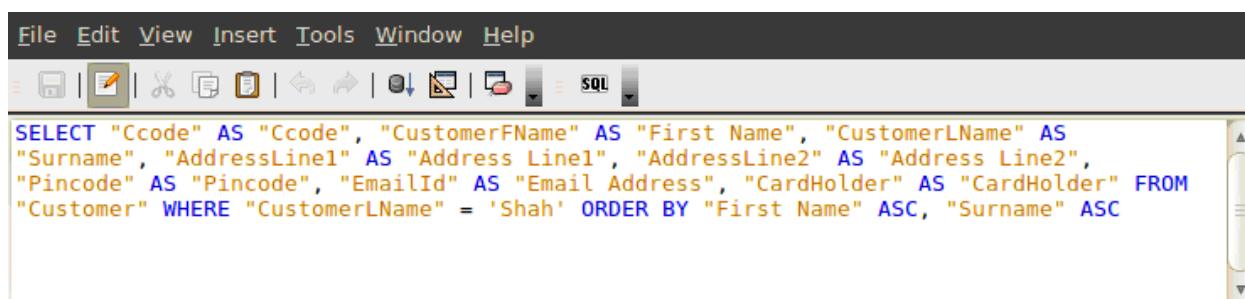
Click on Run Query button on the function bar at the top to view the result. You will find that complete detail of all the employees stored in the Employee table is displayed. To save the query, click on the Save button. Name the query as *EmployeeDetails* and close the Window.

Let us try to create some more SQL queries. Open new query in Design View. Select the *Create Query in SQL view...* option. Type the following statement:

SELECT FirstName, LastName from Employee;

Click on the Run Query button on the function bar at the top to view the result. Observe that the result now displays data for only two fields namely FirstName and LastName for all the employees.

Notice that the query created using SQL statement is also available under the Queries tab similar to queries created using Design View. Right click on any query and you will find an option *Edit in SQL View*. Click it to edit the query using the SQL statement. In fact an SQL statement for each query created as part of this chapter either, using wizard or design view can be viewed. For example, right click on the query CustomerList select *Edit in SQL View* and you will find query statement as seen in figure 11.31.



The screenshot shows the Microsoft Access application interface. The menu bar includes File, Edit, View, Insert, Tools, Window, and Help. The toolbar contains various icons for file operations like Open, Save, Print, and Database. A dropdown menu is open next to the 'SQL' icon. The main window displays the SQL code for the CustomerList query:

```
SELECT "Ccode" AS "Ccode", "CustomerFName" AS "First Name", "CustomerLName" AS "Surname", "AddressLine1" AS "Address Line1", "AddressLine2" AS "Address Line2", "Pincode" AS "Pincode", "EmailId" AS "Email Address", "CardHolder" AS "CardHolder" FROM "Customer" WHERE "CustomerLName" = 'Shah' ORDER BY "First Name" ASC, "Surname" ASC
```

Figure 11.31 : SQL statement for Query designed using Design view

The words Ccode, First Name, Surname, Address Line1, Address Line2, Pincode, Email Address and CardHolder visible after the keyword AS are aliases. While the word Customer after keyword FROM is a table name. SQL statement in figure 11.31 also includes keyword WHERE that is used to retrieve the records based on criteria. The keyword ORDER BY is used to indicate that output needs to be sorted on field CustomerFName (represented by an alias First Name). The keyword ASC further mentions that data should be sorted in ascending order of first names.

Summary

Storing data of business is one aspect of DBMS. The important aspect is accurate retrieval of the same as and when needed. In this chapter we have discussed about how to retrieve information using queries. We learned how to create a query using a wizard, a query design view and SQL commands. We created simple queries as well as parameter query. A simple query does not ask for any input at run time, while a parameter query does ask for input at run time. Also the result of the parameter query can differ at every execution as the parameters entered might be different. In the next chapter we will use the queries that we have created in this chapter to create reports.

EXERCISE

1. What are Queries in Base? Why do we design Queries?
2. Explain the use of Criterion in a query.

LABORATORY EXERCISES

- 1.** Solve the following queries using the Student database created by us as part of exercise of chapter 9 and Chapter 10.

 - (a) List details of all the students studying in tenth standard.
 - (b) List names and address of students who have left the school.
 - (c) List teachers belonging to the city of Ahmedabad or Surat.
 - (d) List total number of subjects taught in the school.
 - (e) List total presence of student with Grno 10 in January 2012.
 - (f) How many subjects does Mr. Akhil Mehta teach in the school? Display it along with the standard that he teaches in.
 - (g) Calculate Percentage of Student with Grno 1 in October, 2011 in first term.
 - (h) Display the result of first term test conducted during October 2011 in the subject of Maths for each student.
 - (i) Create a parameterized query to accept city as a parameter and display students belonging to that city.

2. Design the following tables. Insert ten appropriate records in each table.

Student(StudentId , Name , Branch , Institute)

Exam (CourseNo , CourseName , DateofExam)

Appeared(StudentId , CourseNo)

Solve the following queries :

- (a) List the details of exam conducted for course number 8 or 12.
 - (b) List the student id, his name and the course in which he appeared for exam.
 - (c) List the name of all the students who study in “Satyam” institute.
 - (d) Find total number of student registered in the course number 4.
 - (e) List the course name, date on which exam was conducted and names of the all the students who appeared in that exam.
 - (f) List the course number and name of the course whose exam is to be held on 12/2/2012.
 - (g) List the details of the exam whose course number is 8 or 10 and date of exam is 2/2/2012.
 - (h) List the branch of student whose name starts with alphabet A.
 - (i) Delete all the records of ABC Institute.
3. Design the following tables. Insert ten appropriate records in each table.
EMPLOYEE (EmpId , EmpName, Salary, Gender , Department, JoiningDate)
- Solve the following Queries :**
- (a) List the details of the employee whose name starts with alphabet D.
 - (b) List the details of the employee whose salary is between Rs. 1000 and Rs. 3000.
 - (c) List the details of all the male employees.
 - (d) List the details of the employee who are in marketing department.
 - (e) Find Average salary distributed in the company.
 - (f) List the details of the employee whose salary is greater than Rs. 5000.
 - (g) List the details of the employee whose joining date is before 01/01/2012.
 - (h) List the details of the employee who belongs to either marketing or finance department.
 - (i) List the details of the employee who are not working in the purchase department.
 - (j) List the details of the employee who have joined after 10/09/2011 and working in finance department.





In the earlier chapters, we have discussed how to design the database, perform operations on the table using Table Datasheet View and view information by using Query Datasheet View. The Datasheet View is in the form of row and columns. When we entered data in the Product table, we entered it using spreadsheet style. This black and white format of entering and viewing data is sometimes unpleasant and boring to the user. While designing the Customer table we have used small names for defining field like Ccode which stands for customer code. Sometimes, these types of fields are not self explanatory. In this chapter, we will discuss an alternative way to enter and display the data in the database known as Forms and Reports. We will discuss how to display output in the formatted manner using reports. As explained in previous chapters, Forms and Reports similar to Tables and Queries are objects displayed in left pane of the Database Window. Let us discuss each of them one by one.

Forms

Recall when you took admission in school, you might have entered your data in the admission form designed with fill in the blanks style to enter the data. Base also offers similar alternative to view and work with the data in the tables and it is known as Form. A Form resembles the pen and paper style forms. In the language of databases, a form is a front end for data entry and editing. Forms can be designed using various styles, colors, along with heading, name and logo. Designing forms is really interesting. There are two ways to create a Form.

1. Using Wizard
2. Using Design View

The simplest way is to use the Wizard. Let us continue our journey of learning forms in Base by creating a form using Wizard.

Creating form using Wizard

There are two ways to initiate Form Wizard. The first way is to right click the table for which form is to be created. The second way is to click on Forms icon and choose *Use Wizard to create a form* option from the *Tasks* pane. Figure 12.1 shows how to use the first option. Follow the mentioned steps to create a simple form using wizard.

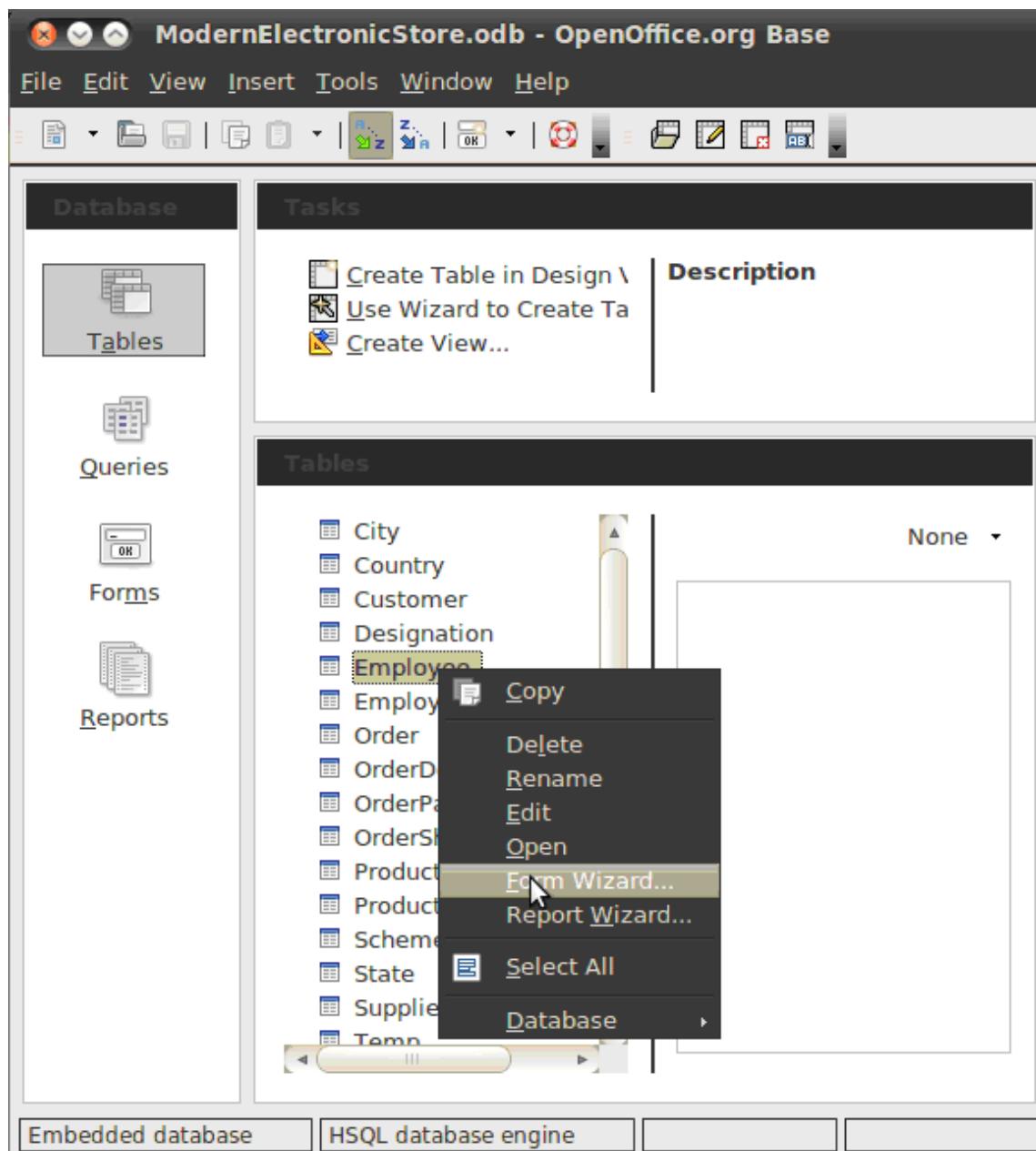


Figure 12.1 : Creating a Form Using Wizard

- Right click on the Employee table.
- From the pop up menu select *Form Wizard...* option. (See figure 12.1)
- The wizard will open two dialog boxes namely; Form Design and Form Wizard together as shown in figure 12.2. For time being we will only look at the Form Wizard dialog box. Also observe that all the fields of the Employee table are listed under *Available Fields* list box of Form Wizard dialog box.
- Click on the right double arrow (>>) to move all of these fields to the *Fields in the form list*.
- Click on the Next button.

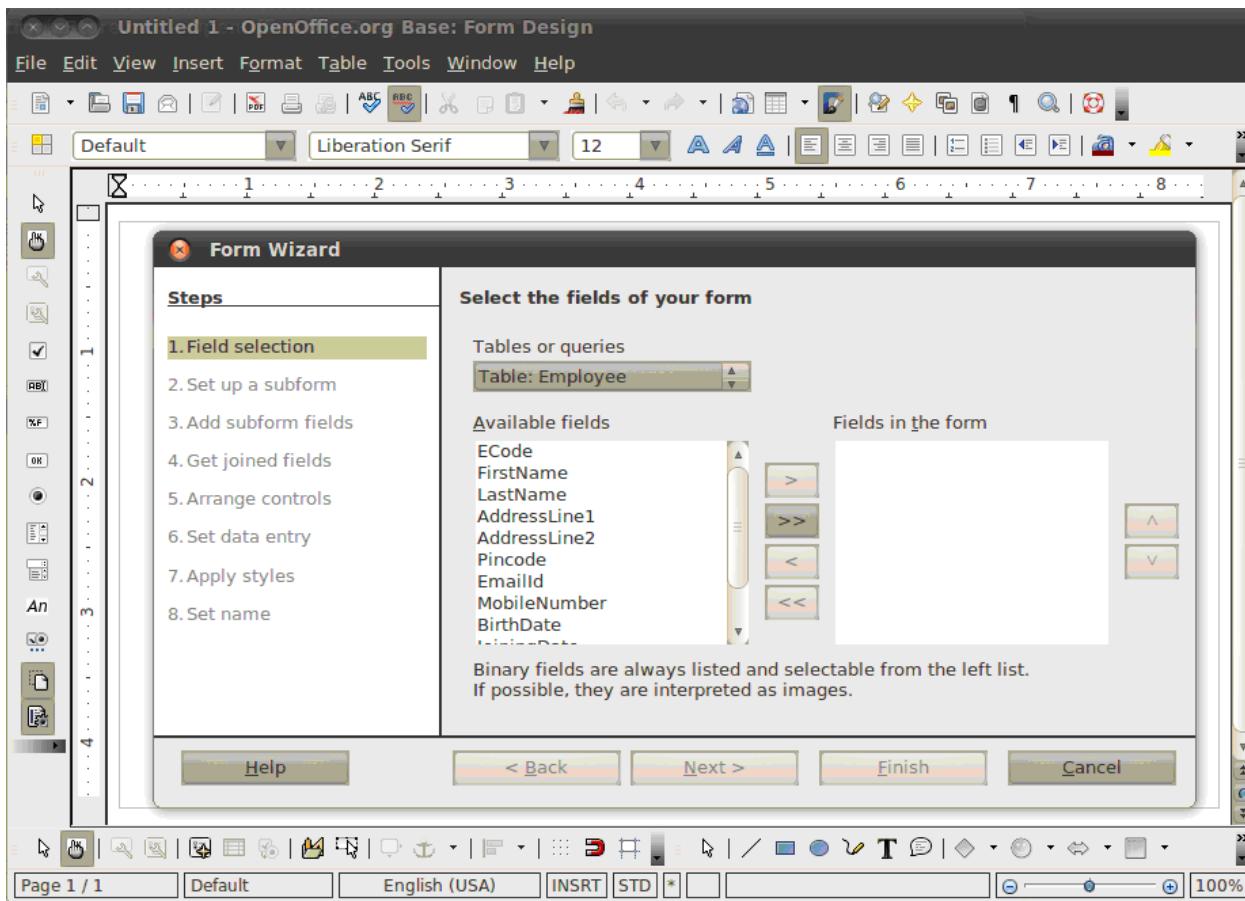


Figure 12.2 : Field Selection for Form creation

- This step consists of setting up a subform. Subforms are discussed later in this chapter. For creating a simple form, click on Next button and step 5 of form wizard as shown in figure 12.3 would be displayed.

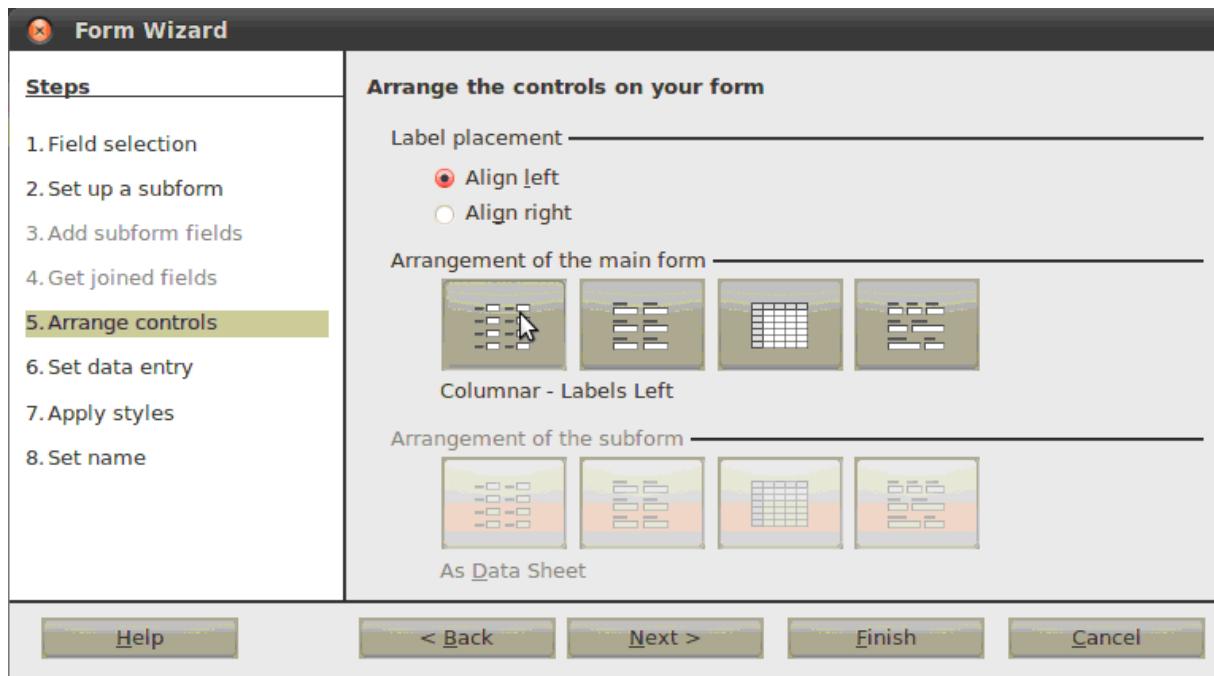


Figure 12.3 : Arrangements of Controls

- This step consists of aligning various controls visible on the form. A control consists of two parts: a label and a field. This step when performed determines how the label and field will be visible on the screen. The labels can be aligned on the left side or the right side. In our case Align left option has been selected. The field corresponding to a label can be arranged in four ways namely Columnar – Labels Left, Columnar - Labels on Top, As Data Sheet, and In Blocks - Labels Above. Under the heading Arrangement of the main form, Click on the Columnar - Labels Left icon.

Click on the Next button, step 6 of form wizard would be displayed as shown in figure 12.4.

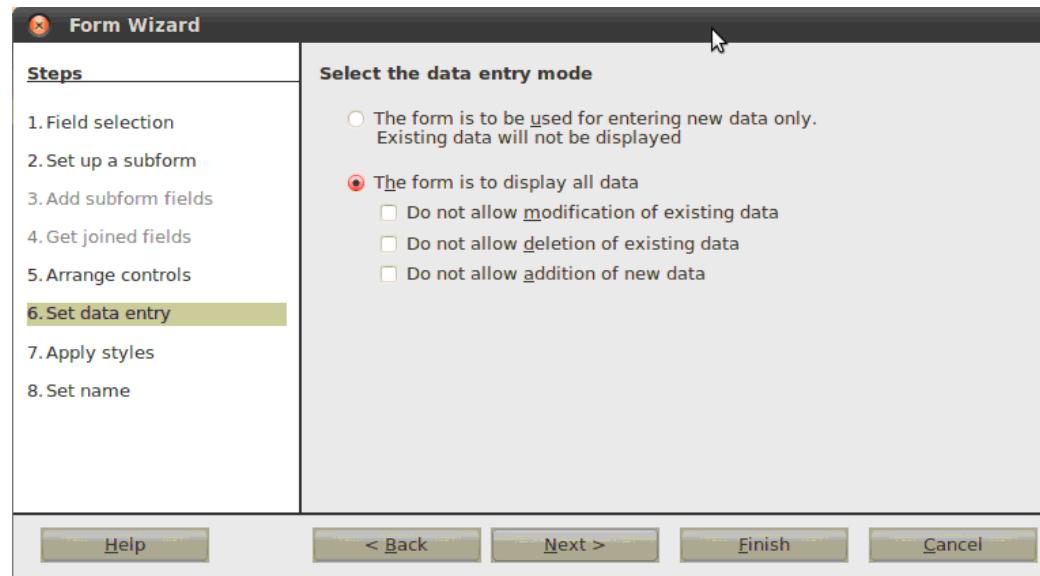


Figure 12.4 : Data entry control settings

- Unless you have a need for any of these entries to be checked, accept the default settings. Click on the Next button and step 7 of form wizard would be displayed as shown in figure 12.5.

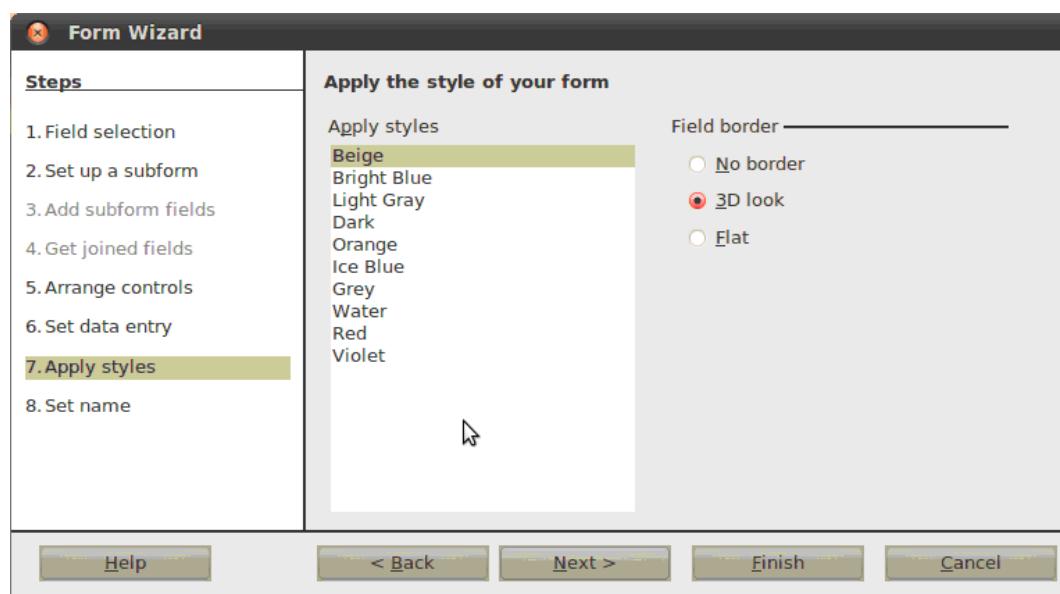


Figure 12.5 : Applying Styles

- Select the color you want in the *Apply styles* list. Figure 12.5 shows that Beige color is selected from *Apply styles* list, while 3D look is selected in *Field border* options. You might experiment with the different possible settings. Click on the Next button to display step 8 of form wizard. The screen of step 8 would look similar to the one shown in figure 12.6.

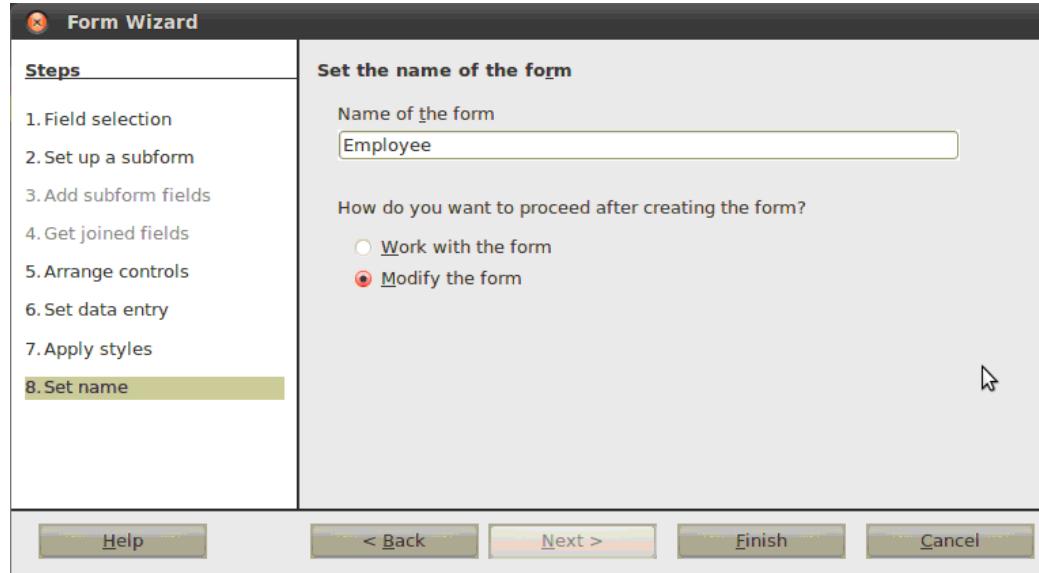


Figure 12.6 : Naming form

- Enter the name for the form. In this case, it is Employee. Click *Modify the form* radio button. Click on the Finish button. The form now opens in the Edit mode as shown in figure 12.7.

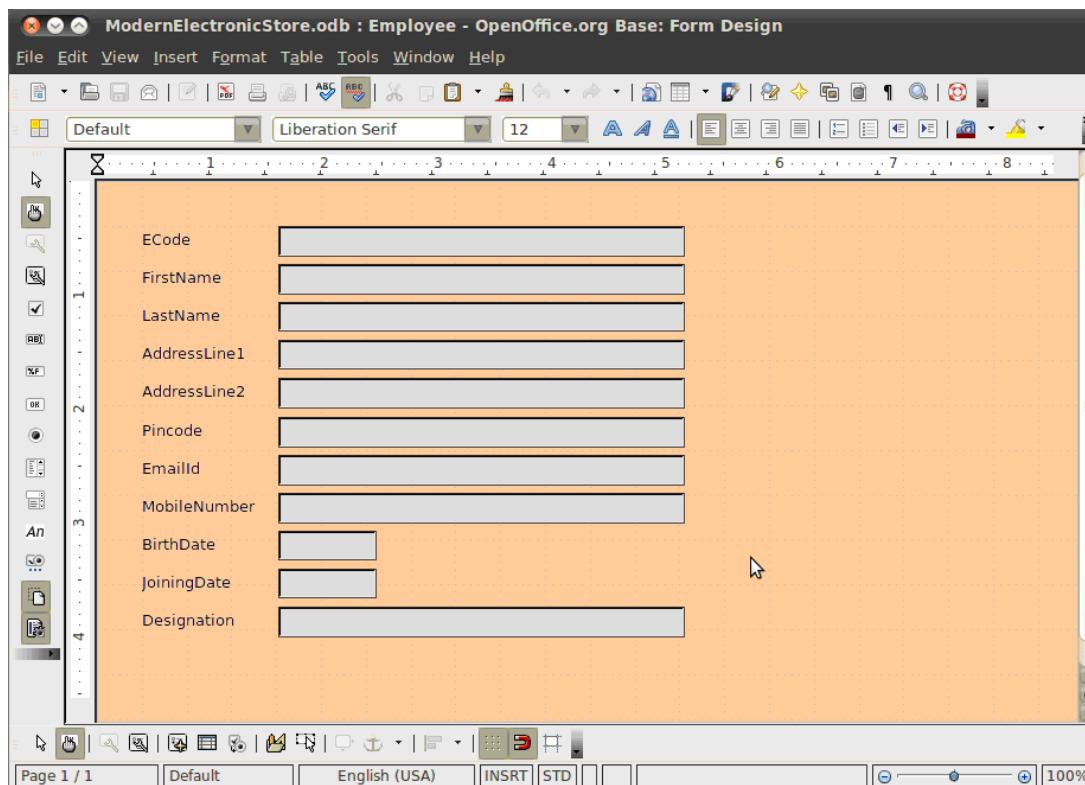


Figure 12.7: Form in modify mode

Modifying Form

The form once created can be easily modified by moving the controls on the form as per our liking. We can move the controls to different places in the form. Background picture can also be changed. We can modify the label for the fields such as the label Ecode can be changed to Employee Code. It is also possible to change a text box to a list box. Let us try to modify the form created using wizard.

Changing Label Text

Let us change label ECode to Employee Code by performing the following steps:

- Press CTRL + Click to select the label containing ECode as text.
- Now right click on the label and from the popup menu select *Control...* option. A *Properties: Label Field* dialog box as shown in figure 12.8 will be opened.
- Type Employee Code in the text box after Label..... (See figure 12.8).
- Close the *Properties: Label Field* dialog box and you will observe that the label ECode visible in the Form Design changes to Employee Code.

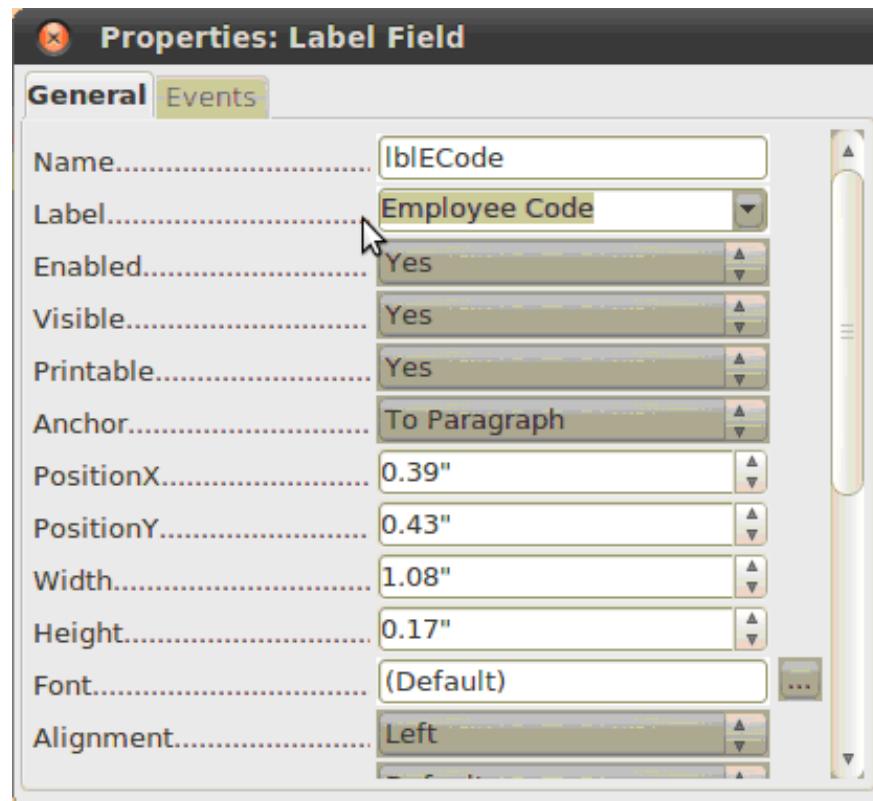


Figure 12.8 : Properties Dialog box

As can be seen in figure 12.8, the *Properties: Label Field* dialog box consists of many properties related to this control. Features like Height, Width, Text Alignment, and Visibility of a control can also be set using it.

Moving Control

The controls in the main form consist of a label and its field. Clicking a label or field selects the entire control. A border appears around the control with eight green handles as can be seen in figure 12.9. You can then drag and drop it anywhere you want within the form.

Many times we may need to modify only label. Hence only one component, which is label, needs to be chosen. To choose either a label or a text field, press CTRL key and click on the label or a text field. Figure 12.10 shows the text field selection.

Changing Control size

Let us increase the width of the Employee Code field.

- Press CTRL and click on the Employee Code field text box to select it.
- Move the cursor over the middle of the green handle on the right side. It should change to a single arrow as seen in figure 12.11.
- Hold the left mouse button down as you drag the cursor to the right until the width is desired width. You can see the changing size at bottom of the screen (see figure 12.12).



Figure 12.9 : A Selected control



Figure 12.10 :
Selecting a field of control



Figure 12.11 :
Changing Control size

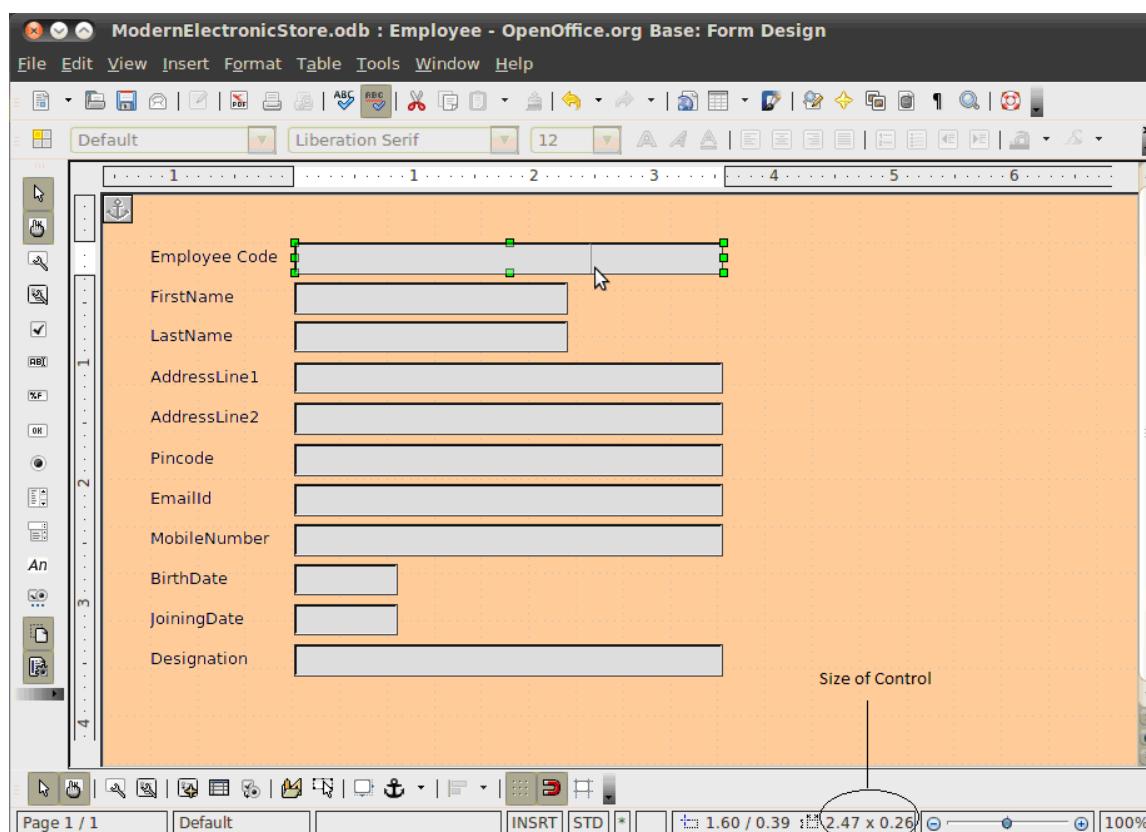


Figure 12.12 : Changing Size of Control

- You can double click the position and size area to open the *Position and Size* dialog box as shown in figure 12.13.

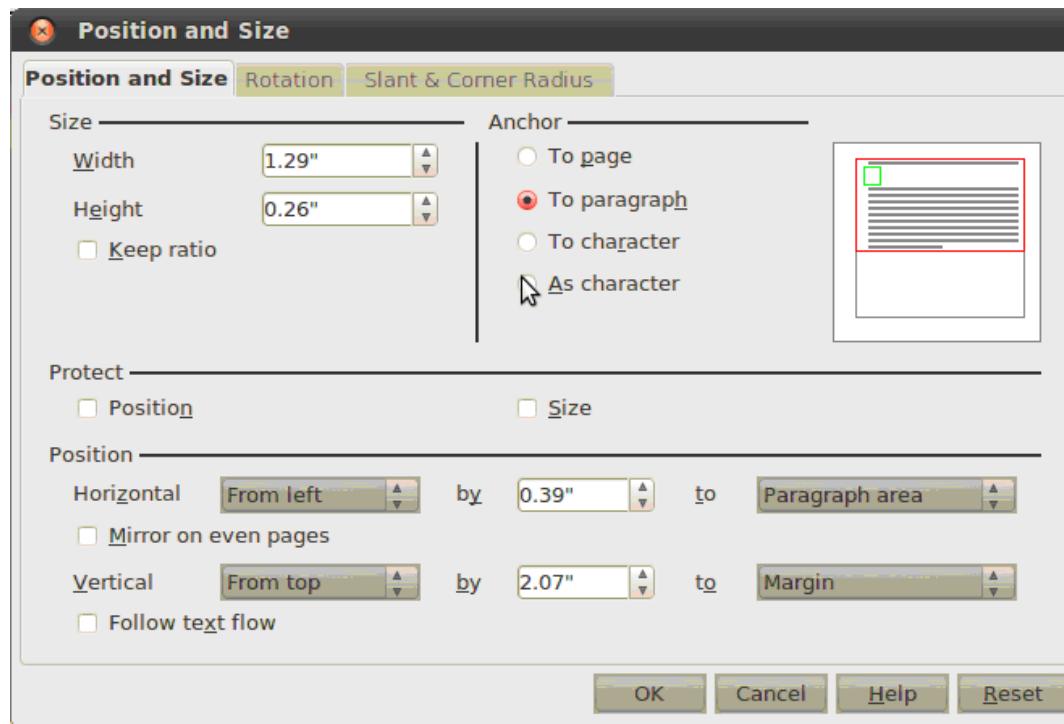


Figure 12.13 : Position and Size

Creating Help Text

Have you ever rested a mouse pointer for a while on buttons in toolbar? Try it. You will find a yellow box showing the text that mentions the purpose of the button. This yellow box is known

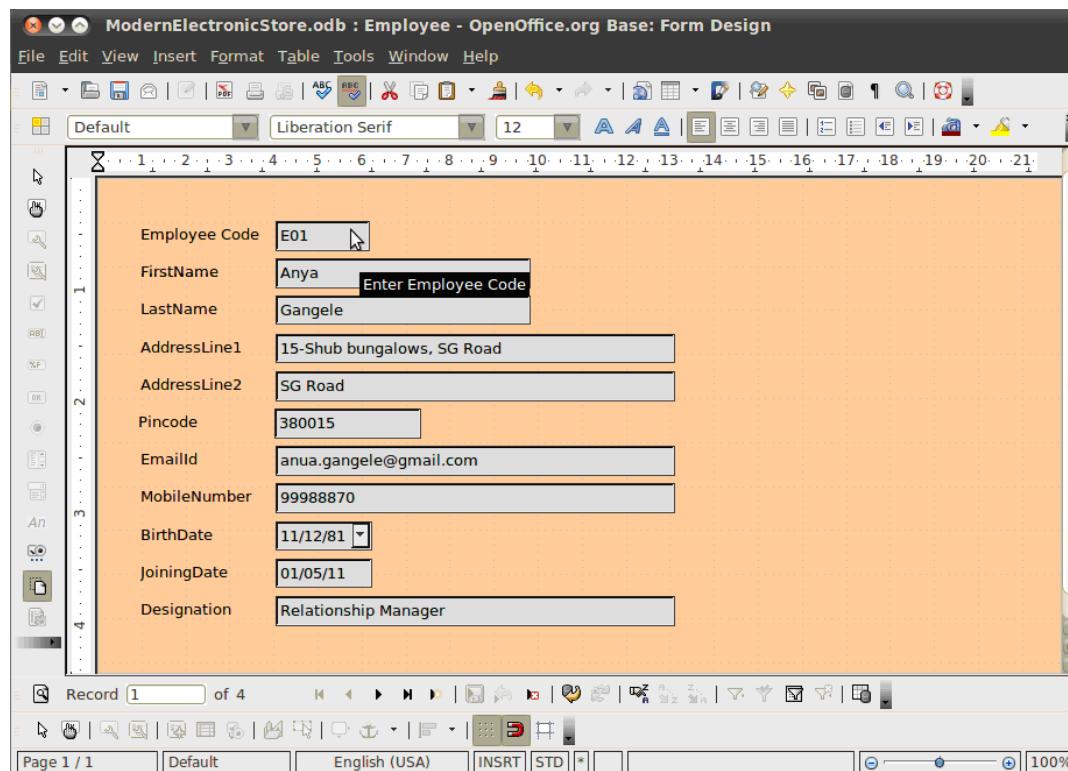


Figure 12.14 : Help Text

as *tool tip text*. Similarly, controls placed on data entry forms in Base can be tagged with help text as can be seen in figure 12.14.

Let us attach a tool tip *Enter Employee Code*, on the text box control associated with Employee Code label. Press CTRL and click on the text box to select the control. Right click the control and choose *Control...* option. The *Properties: Text Box* dialog box as seen in figure 12.15 would be displayed. Scroll down till you see a label *Help text*. Now enter the text “Enter Employee Code” as shown in figure 12.15.

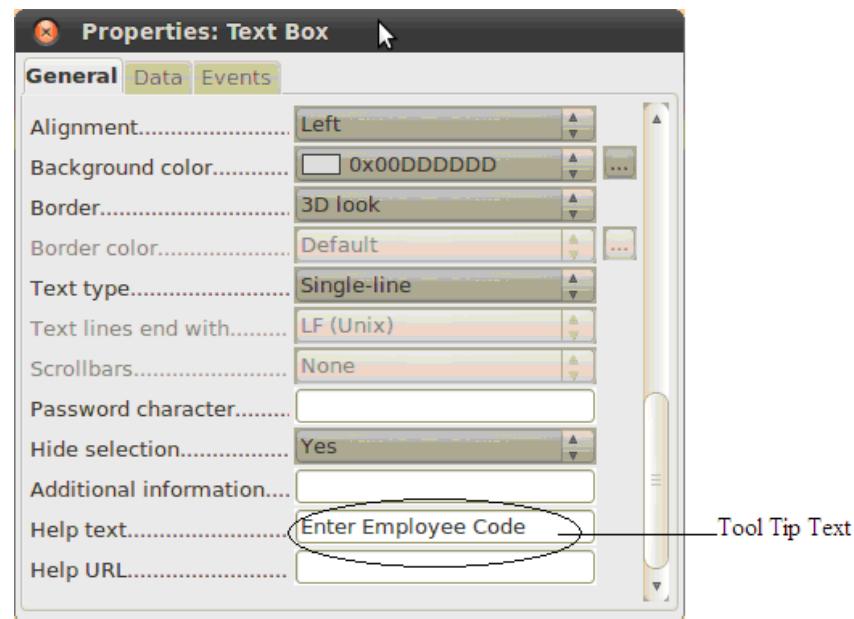


Figure 12.15 : Setting Help text

Change the Date field

Perform the following steps to change the date field to calendar control as seen in figure 12.16.

 A screenshot of the OpenOffice.org Base 'Form Design' window. The form contains fields for Employee Code (E01), FirstName (Anya), LastName (Gangele), AddressLine1 (15-Shub bungalows, SG Road), AddressLine2 (SG Road), Pincode (380015), EmailId (anua.gangele@gmail.com), MobileNumber (99988870), BirthDate (11/12/81), JoiningDate (December 1981 calendar), and Designation. The BirthDate field is currently active, showing a calendar control. The toolbar at the top includes icons for file operations, zoom, and various tools. The bottom features a navigation bar with 'Page 1 / 1', 'Default', and other form-related buttons.

Figure 12.16 : Changing Date field

- Select the BirthDate field text box by clicking on it with CTRL key pressed.
- Move the cursor over the middle of the green handle on the right side. It should change to a single arrow.
- Hold the left mouse button down as you drag the cursor to the right until the length is as per requirement.
- Release the mouse button.
- Right click on the text box and choose *Control...* option. Alternatively you can click on the *Control* icon in the Form Controls toolbar as shown in figure 12.17. This toolbar will be placed vertically on the left side of the form (See figure 12.2 or 12.3). A *Properties: Date Field* dialog box will open.

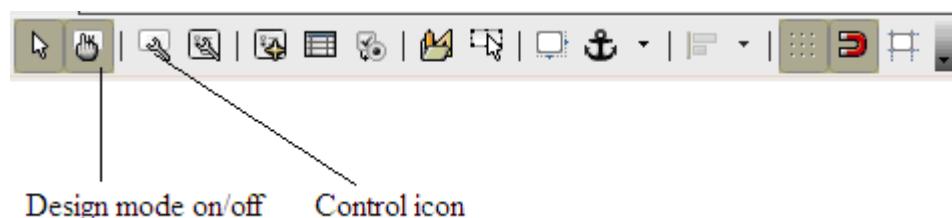


Figure 12.17 : Form Controls toolbar

- Scroll down to the *Date format.....* property. This is a drop down list with *Standard (short)* as the default setting.
- Click on the default *Standard (short)* to open the list. Select the *Standard (long)* option from the list.
- Scroll down to the *Dropdown.....* property. Its default setting is No. Change it to Yes.
- Close the dialog box. You will observe that the text box has now turned into a drop down.

Note : We can also click on the *Design Mode On/Off* icon (the second icon from the left in figure 12.17). This will toggle the form design view to form view and vice-versa.

Changing Background

The background for a form can be a color, or a graphic (picture). You can use any of the colors in the Color Table available in Tools-Options-OpenOffice.org-Colors. You can also use a graphic file as the background. Perform the following steps to change background of form.

- Right click on the blank space in a form. Sub menu as shown in figure 12.18 would appear.

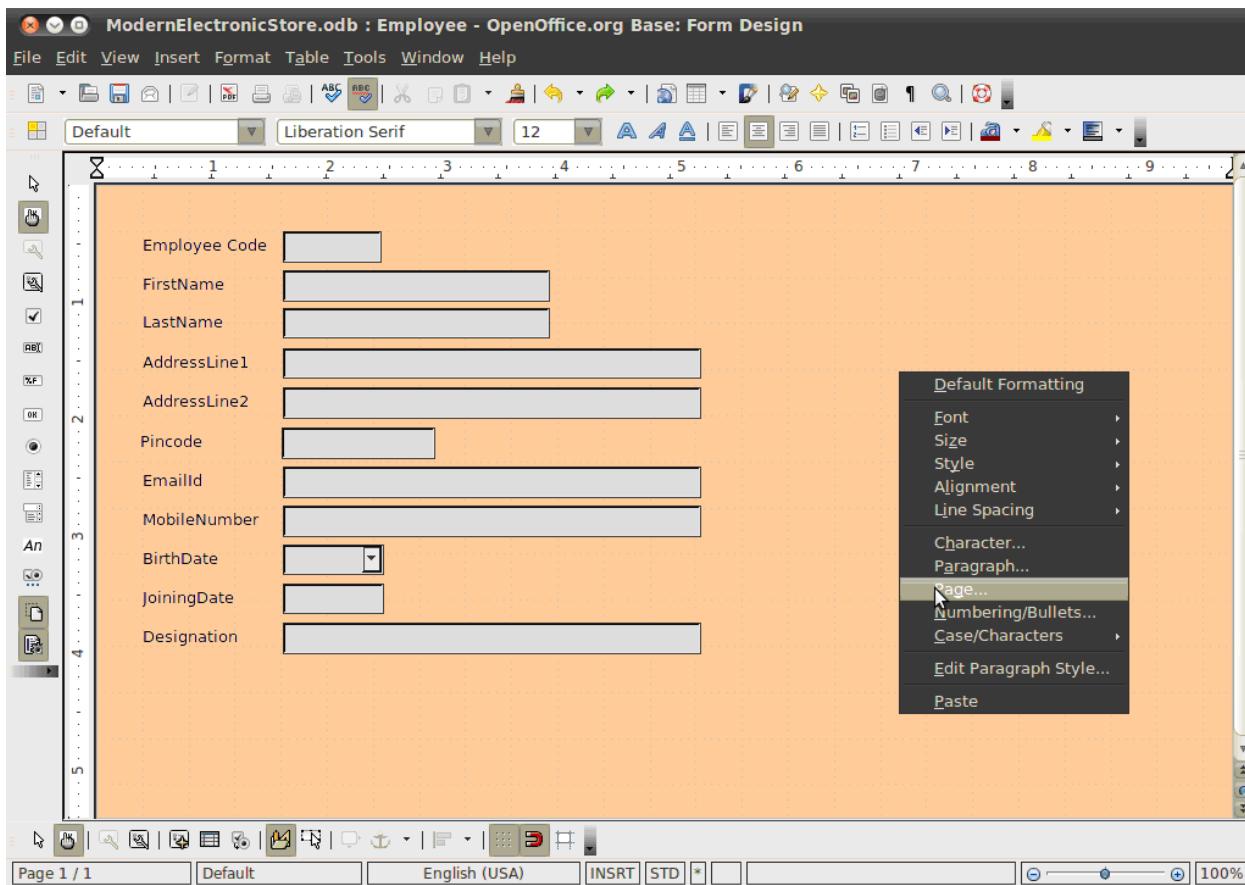


Figure 12.18 : Form Sub-menu

- Choose *Page...* option. A *Page Style* dialog box as shown in figure 12.19 would appear.

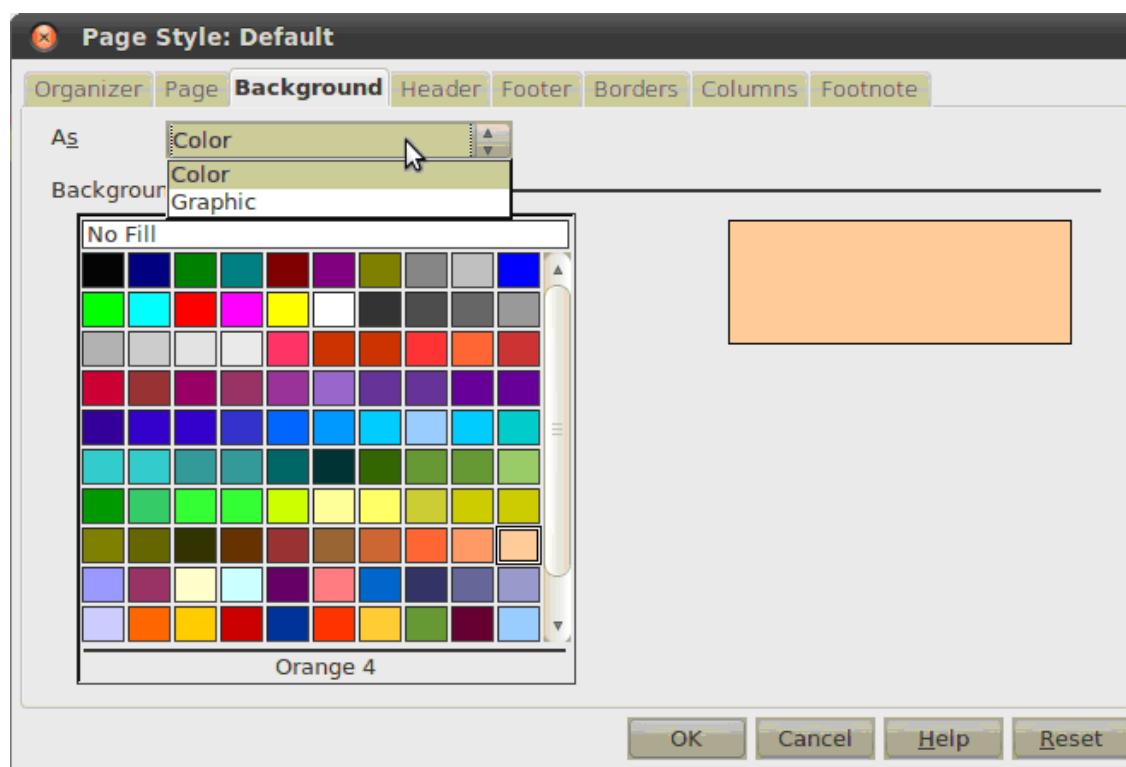


Figure 12.19 : Page Style dialog box

- Choose the desired color and click on OK button. The background of the form will be now changed to the selected color.

Searching a Record using Form

Open the Employee form created by double clicking the Employee icon under Forms tab. The screen will look similar to the one shown in figure 12.20.

Employee Code	E01
FirstName	Anya
LastName	Gangele
AddressLine1	15-Shub bungalow, SG Road
AddressLine2	SG Road
Pincode	380015
EmailId	anua.gangele@gmail.com
MobileNumber	99988870
BirthDate	12/11/81
JoiningDate	01/05/11
Designation	Relationship Manager

Figure 12.20 : Employee form in Data Entry mode

Notice that in figure 12.20 when the Form is opened, the first record is displayed on the screen. You can view only one record at a time in the Form view. To see other records, use the Navigation buttons visible at the bottom of the form. (See figure 12.20).

At times we may need to find a specific record. Perform the following steps to find a particular record:

Select Find Record option on the form navigation bar as shown in figure 12.21.

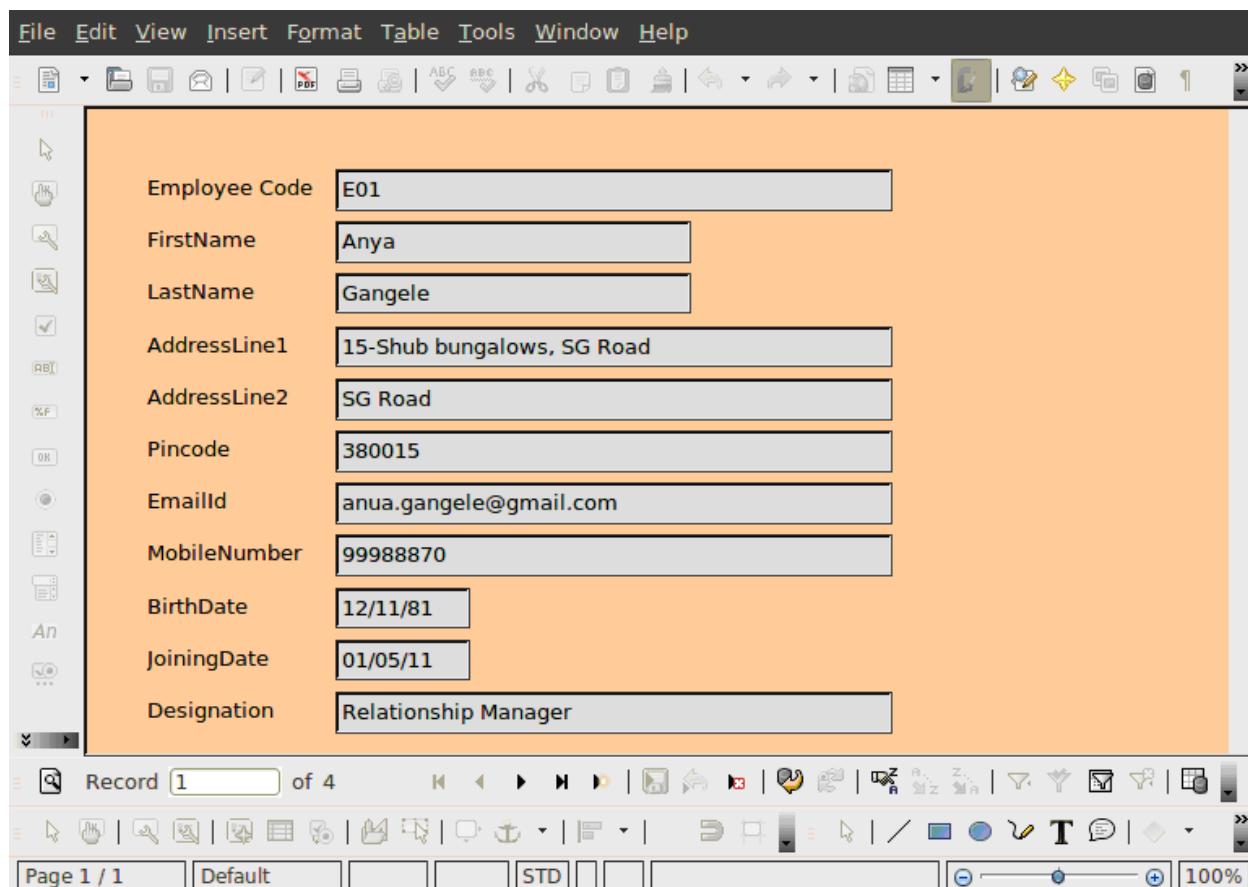


Figure 12.21 : Find Record option

A *Record Search* dialog box as seen in figure 12.22 will appear on the screen.

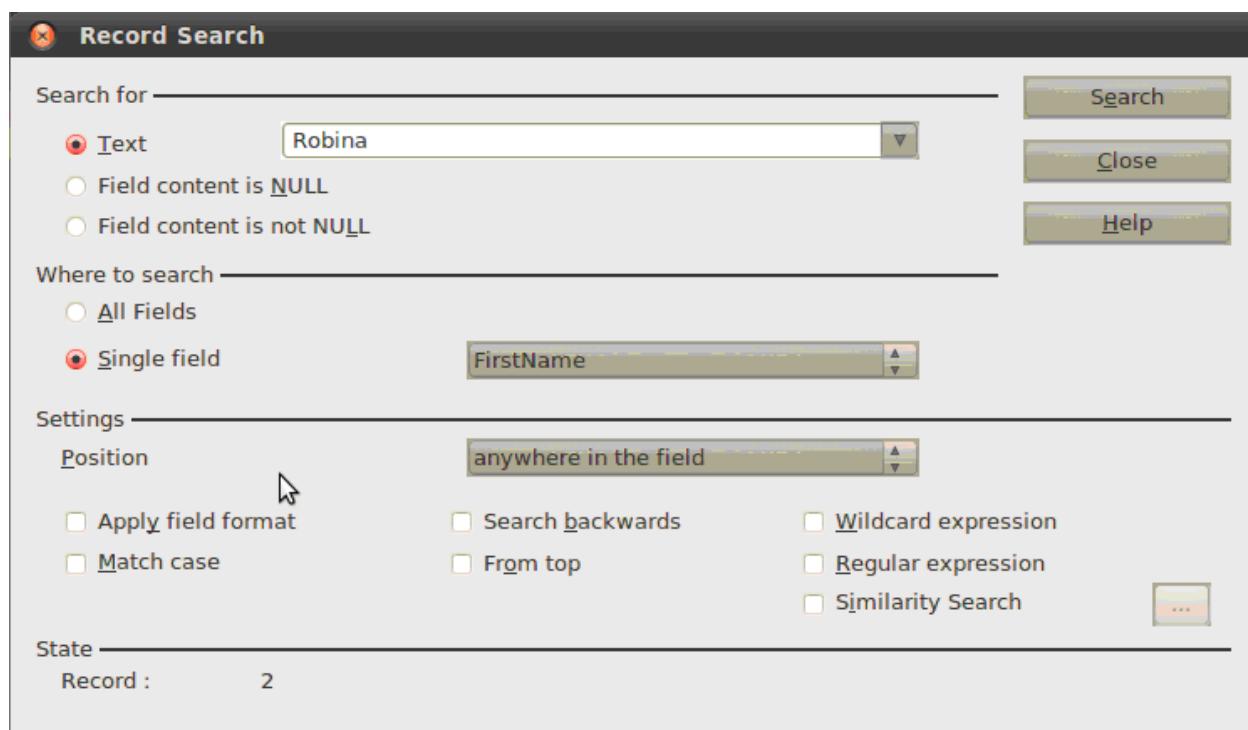


Figure 12.22 : Record Search dialog box

- Select the *Single Field* option under the heading *Where to search*.
- Select the FirstName field name in drop down box next to *Single field*. (See figure 12.22). This will populate the drop down box next to *Text* option.
- Select *Robina* from the drop down next to *Text option*. Observe that the radio button in front of Text label is selected.
- Click on Search button.
- Click on the Close button, the entries in the form will now be filled up with the record pertaining to Robina.

Inserting and Deleting a Record using Form View

To insert a new record in the table using the form view perform following steps:

- Open the form related to the table that you want to enter the data in Form View.
- Click on the New Record button  on the navigation bar.
- Type the data in blank boxes visible.
- Close the Form.
- Click on the Tables icon in the Database Window.
- Open the table and you will find that the record inserted by you is now stored in the table.
- You can also delete a record through a form using the Delete Record () button on the Form View toolbar. Clicking on this button will delete current record displayed in the form. The next record will be displayed automatically. This operation will also remove the record from the corresponding table.

Reports

After learning how to create the forms, you must now be sure that using forms to enter data in a table is much easier than using its Data Sheet View. We have seen that it is possible to extract any information in Base using a query. The output obtained from a query though presented in a tabular format is not properly aligned. Report provides a way to present the information retrieved in an attractive, arranged and decisive manner. One of the purposes of generating a report is to make a hard copy of the output. Hence the layout of report is generally designed by keeping in mind the look of the hard copy required. We can create a report based on a query, a table or combination of both.

Reports created in this chapter are based on either a single table or a query. If you want to use fields from different tables, it is advised that you create a query that combines these fields as an output. Then create a report required using this query.

For example, in the earlier chapter, we created a query by joining four tables, Customer, City, State and Country to display the list of customer names along with their addresses. We named this query

as CustomerList. Let us create a report based on the same query. Report wizard can also be opened using the following two ways.

- Right click on the query or table for which the report has to be created and selecting the *Report Wizard...* option from the popup menu.
- Alternatively click the *Reports* icon in the Database Window and choose *Use Wizard to Create Report...* option.
- When we use either of the options mentioned above, two dialog boxes as shown in figure 12.23 will be opened. At present we will concentrate on the Report Wizard dialog box only.
- The first step is to select the query CustomerList from *Tables or queries* drop down list in *Report Wizard* dialog box. Use the >> button to move all these fields from the *Available fields* list to the *Fields in report* list.
- Click on the Next button.

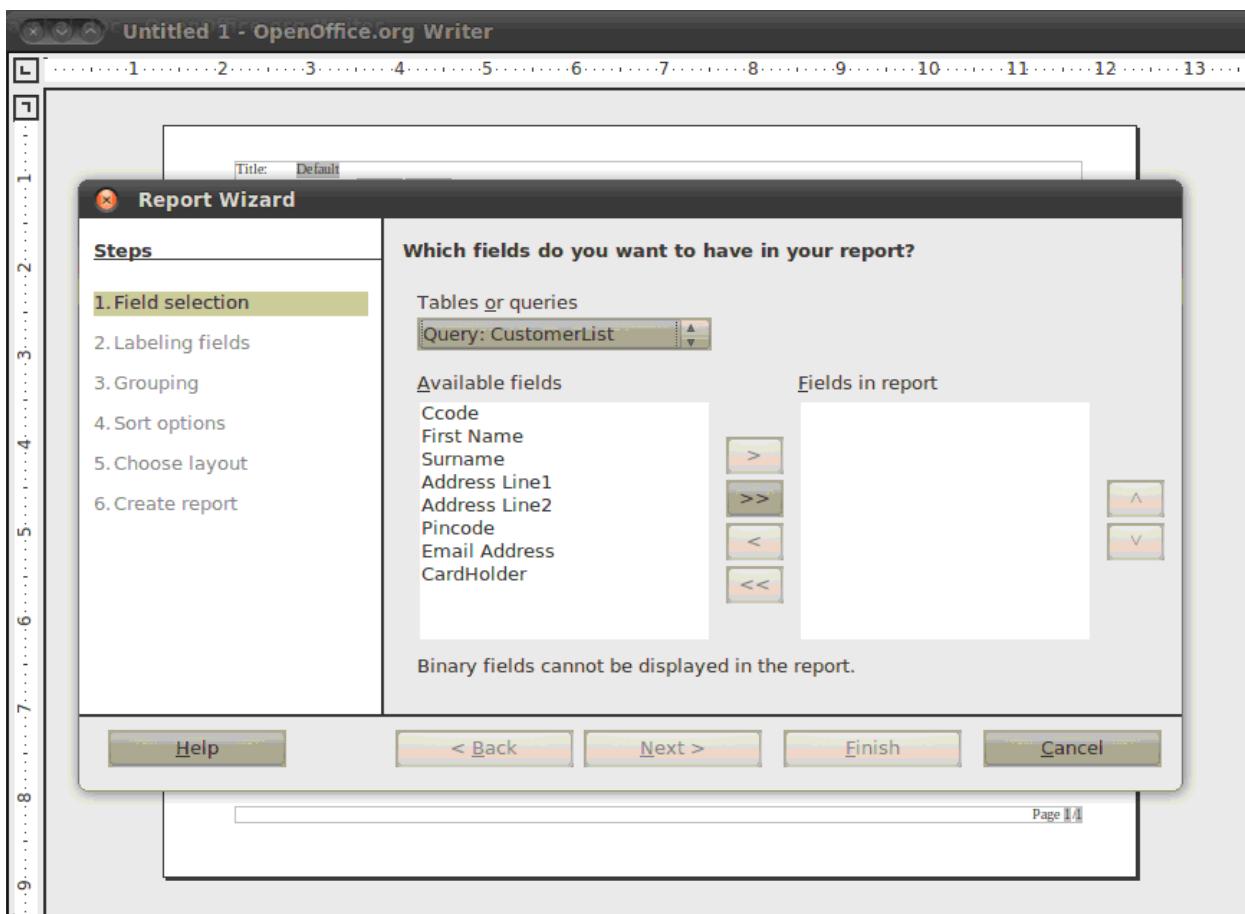


Figure 12.23 : Step 1 of Report wizard

- In second step give appropriate labels to each field as per your requirement. Sample labels are as shown in figure 12.24.

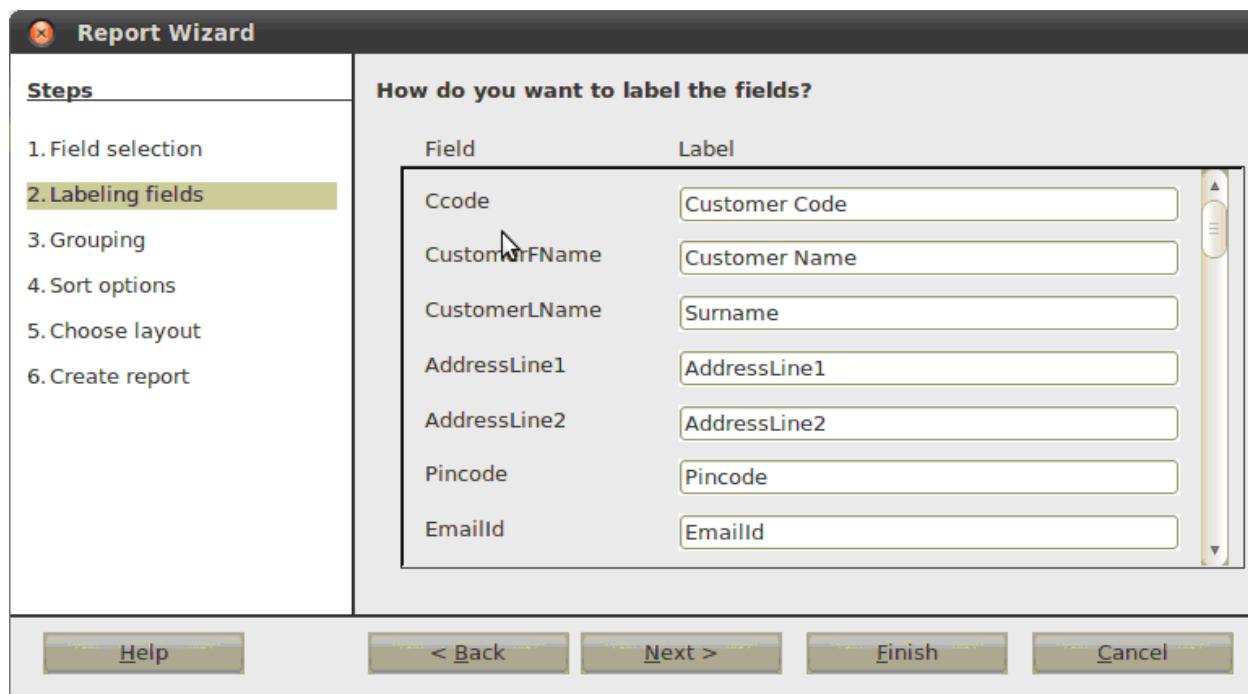


Figure 12.24 : Step 2 of Report Wizard: Labeling fields

- Click on the Next button.
- Third step of wizard is grouping the fields. Since we do not want to group any field in this report, click on the Next button. Step 4 of the Report Wizard as shown in figure 12.25 will be displayed.

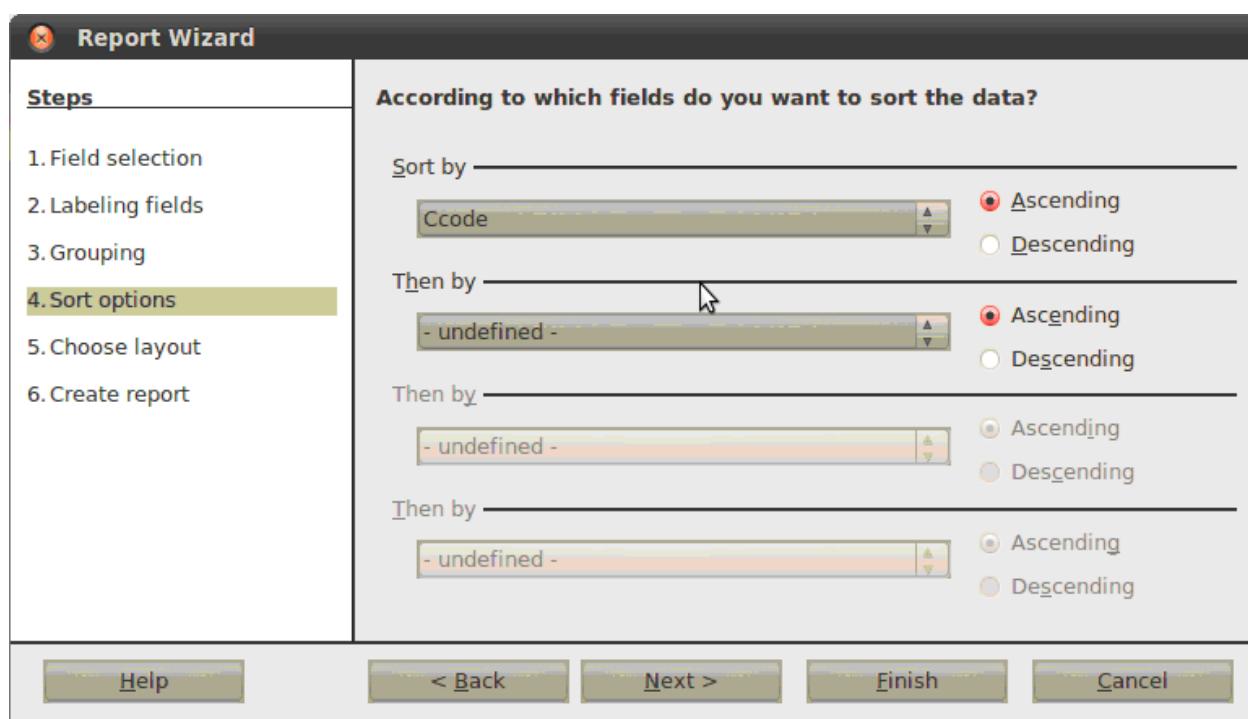


Figure 12.25 : Step 4 of Report Wizard: Mention Sort options

- In fourth step we can mention sort options. Let us sort the records based on the field Ccode in Ascending order. Click on the Next button to display next step as shown in figure 12.26.
- The fifth step of Report wizard gives us choice of various layouts. Layouts include color combinations as well as positioning of field and alignments of texts. You can try each of them one by one. We will be using the default settings for the layout (See figure 12.26).
- Click on the Next button.

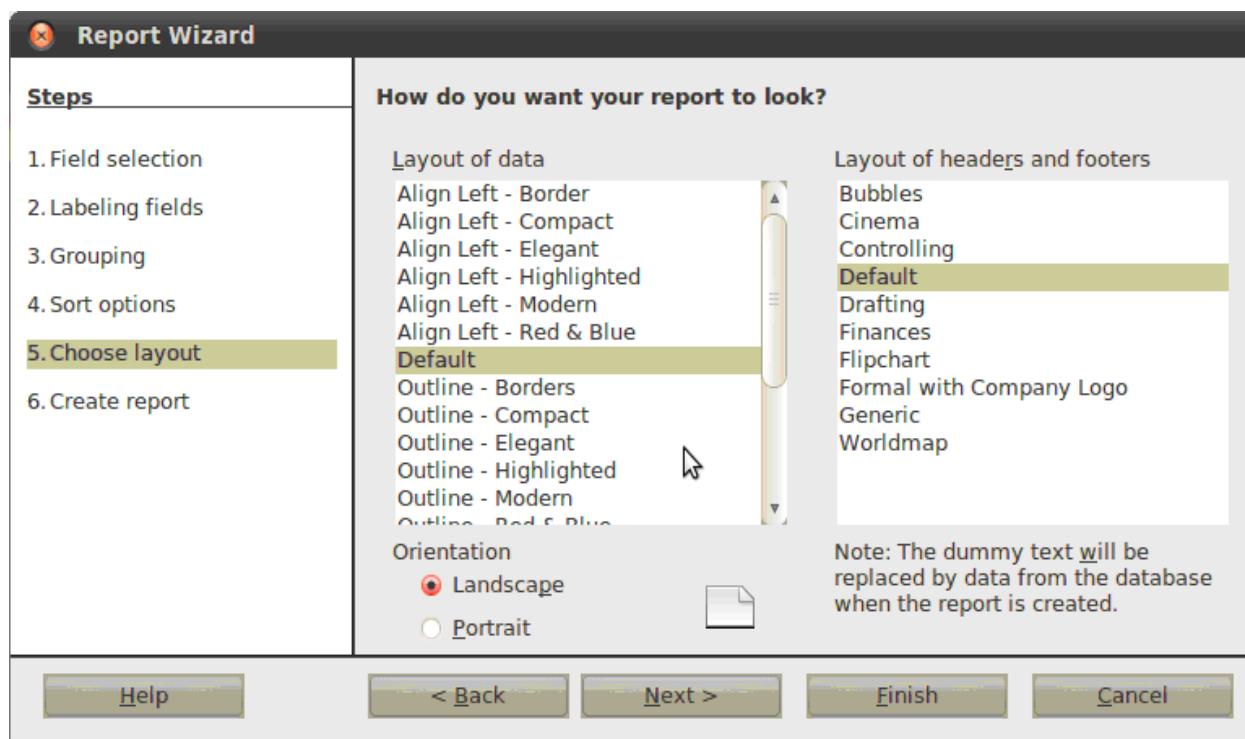


Figure 12.26 : Step 5 of Report Wizard: Choose Layout

- In step six you will be asked to give the title of the report. Type *CustomerList* in the text box under label *Title of report* as can be seen in figure 12.27. Observe that under the label *What kind of report do you want to create?* you have two options. The option chosen here will have an effect on the output of the report. If we choose the *Static report* option, then the data in the output of report will remain same every time we try to open the report. While choosing the *Dynamic report* option will allow us to see the effect of any changes made in the table, every time we open the report.

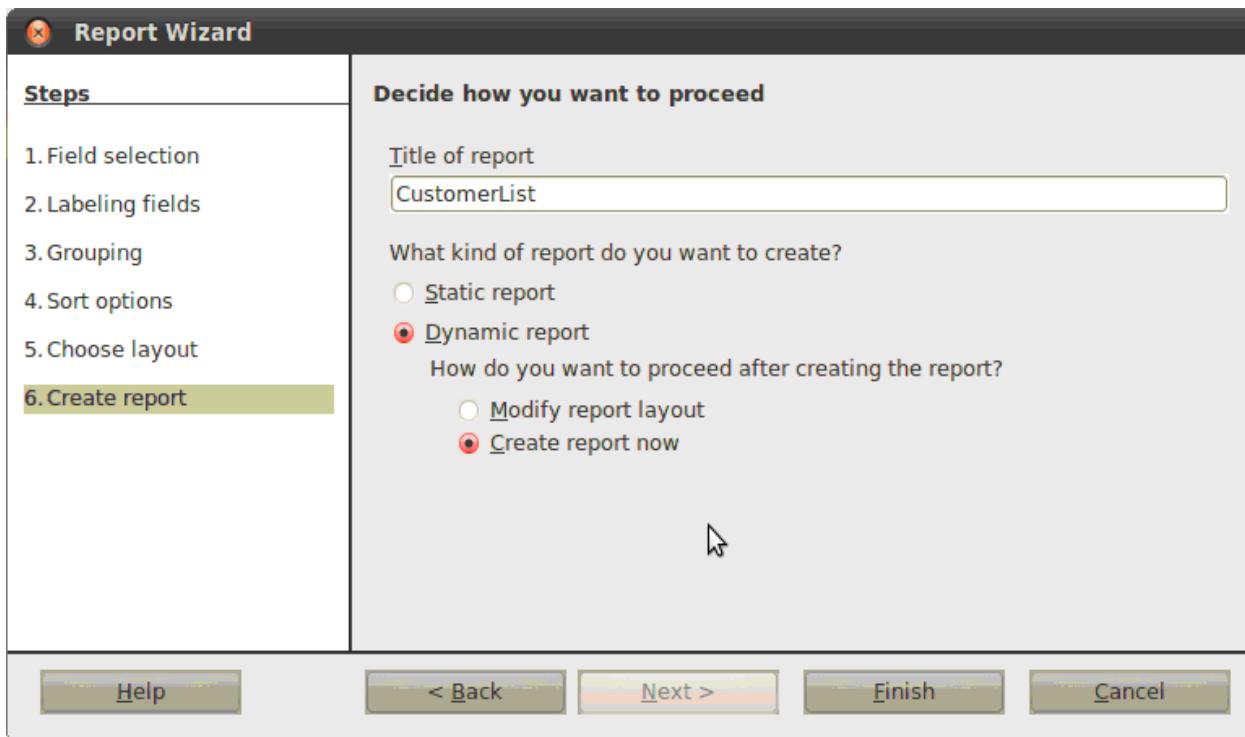


Figure 12.27 : Step 5 of Report creation: Mention Title of report

Note :

The objects created in Base are dependent on each other. This means that if a query Q is created based on a table T, then any changes made to the table T will be reflected in the query Q and vice versa. Similarly if we create a report R based on this query Q, then any change in query Q should be reflected in report R. Albeit changes in report are not possible. Choosing the *Static report* option will permit the changes to be reflected in the report.

- Depending on what you require, choose one of the options mentioned in step six. In our case we have selected *Dynamic report*. Select the *Create report now* option.
- Click on Finish button and you will see a report similar to the shown in figure 12.28.

The screenshot shows the OpenOffice.org Base Report Design interface. At the top, the title bar reads "ModernElectronicStore.odb : CustomerList (read-only) - OpenOffice.org Base: Report Design". The menu bar includes File, Edit, View, Insert, Format, Table, Tools, Window, and Help. The toolbar contains various icons for file operations like Open, Save, Print, and Database. The main area displays a report with a header section containing "Author: Amit Jain" and "Date: 8/17/12". Below this is a table with 11 columns: Customer Code, Customer Name, Sur-name, Ad-dress-Line1, Ad-dress-Line2, Pincode, EmailId, Card-Holder, City, State-Name, and CountryName. The table has five rows of data. Row 1: C01, Puja, Sethia, 17, Akash, Jodhpur, 380056, puja.sethia@yahoo.com, , Ahmedabad, Gujarat, India. Row 2: C02, Sakina, Nagingar, T-37, Abhima nyu BG-1, Curry Road, 384001, sakina82@gmail.com, , Mehsana, Gujarat, India. Row 3: C03, Joseph, Thomas, BG-1, Sunstar, Sola Road, 362268, joseph@yahoo.com, , Patan, Gujarat, India. Row 4: C04, Preerna, Agrawal, Smita Bunglow No 17, Jaldeep -1, PNT Road, 380089, preerna@gmail.com, , Ahmedabad, Gujarat, India. Row 5: C05, Harshit, Shah, Bopal, 380058, harshit@yahoo.com, , Ahmedabad, Gujarat, India. The bottom of the screen shows the page navigation bar with "Page 1 / 1", "First Page", "STD", and "100%".

Customer Code	Customer Name	Sur-name	Ad-dress-Line1	Ad-dress-Line2	Pincode	EmailId	Card-Holder	City	State-Name	CountryName
C01	Puja	Sethia	17, Akash	Jodhpur Char Rasta	380056	puja.sethia@yahoo.com	<input type="checkbox"/>	Ahmedabad	Gujarat	India
C02	Sakina	Nagingar	T-37, Abhima nyu BG-1,	Curry Road	384001	sakina82@gmail.com	<input checked="" type="checkbox"/>	Mehsana	Gujarat	India
C03	Joseph	Thomas	BG-1, Sunstar	Sola Road	362268	joseph@yahoo.com	<input type="checkbox"/>	Patan	Gujarat	India
C04	Preerna	Agrawal	Smita Bunglow No 17, Jaldeep -1	PNT Road	380089	preerna@gmail.com	<input type="checkbox"/>	Ahmedabad	Gujarat	India
C05	Harshit	Shah		Bopal	380058	harshit@yahoo.com	<input checked="" type="checkbox"/>	Ahmedabad	Gujarat	India

Figure 12.28 : Report containing Customer Addresses

Note : The author name and date will depend on the machine name and day on which you are creating/opening the report. The data contents will be according to the entry that you have done in the table.

Let us now create report that shows the orders placed by each customer between two dates. First we will create a parameterized query so we can create report containing records between any two desired dates. The parameterized query will make our report dynamic, as whenever we try to execute the report we will be asked to enter two dates. The report will then be generated for the data between these two specified dates.

- Open a query in Design View. Add Customer, Order and OrderDetail tables.
- Double click on the Ccode, CustomerFname, CustomerLname, OrderID and OrderDate fields from respective tables.
- In the *Criterion* cell of the OrderDate field, type *BETWEEN :date1 AND :date2* as seen in figure 12.29.

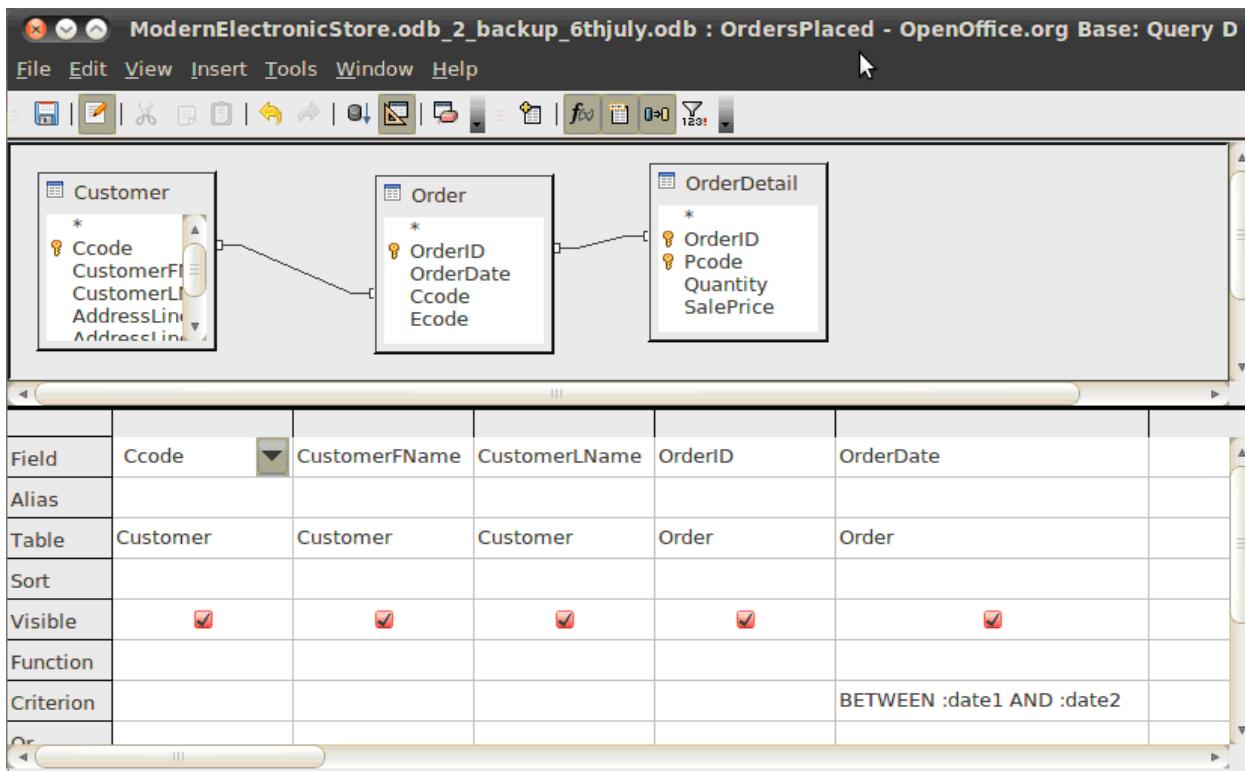


Figure 12.29 : Parameterized query

- Save the query with name *OrdersPlaced*.
- Run the query to view the output to make sure that it works. Two parameters would be expected as an input. (See figure 12.30)

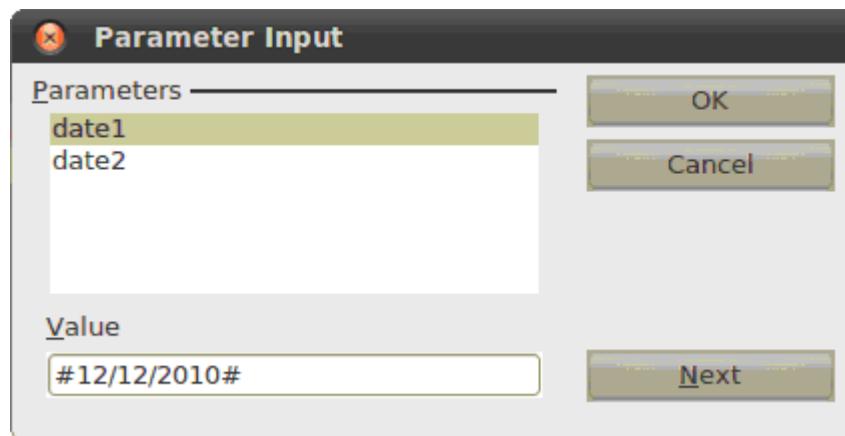


Figure 12.30 : Parameter Input to a query

- Close the query.
- Right click on the query and from the pop up menu select *Report Wizard...* option.
- Use >> to move both fields from the *Available Fields* to the *Fields in report* list. Click Next.
- Give appropriate aliases to field names. Click Next.

- In the Grouping step of the report wizard, Click on the Ccode field to highlight it. Use > button to move the field to the Groupings list. Similarly move the fields CustomerFname and CustomerLname to the Groupings list. Click Next.
- You will observe that sorting has been automatically applied on the Ccode, CustomerFName and CustomerLName fields. Click Next.
- Choose a layout. Click Next.
- Use the suggested name for report; here it will be same as the query.
- Select *Dynamic report* option.
- Click Finish and you will find that input parameter dialog box gets opened.
- Give two parameters for date1 and date2. A report that shows orders placed between two dates would now be displayed as shown in figure 12.31.

The screenshot shows a report viewer window with a toolbar at the top and bottom. The report content is organized into sections by customer code (C01, C02, C03). Each section contains the customer's name and a table of their orders with OrderID and Order Date.

Customer code C01	
Customer Name	Puja Sethia
<i>OrderID</i>	<i>Order Date</i>
1	12/31/10
2	01/20/12

Customer code C02	
Customer Name	Sakina Nagingar
<i>OrderID</i>	<i>Order Date</i>
3	02/14/12

Customer code C03	
Customer Name	Joseph Thomas
Surname	
<i>OrderID</i>	<i>Order Date</i>
4	02/19/12

Figure 12.31 : Parameterized Report

Once the report has been created we can make a hard copy or a soft copy of the same. To make a hard copy we can print the report by making use of print button () on standard toolbar. Alternatively to save it as a soft copy click on PDF button () on standard toolbar, this will save the report in a PDF format.

Summary

For any application, forms and reports are the objects with which end user interacts. In this chapter we have discussed professional way to enter data in the table using forms. We have also discussed creation of reports using table as well as query.

EXERCISE

LABORATORY EXERCISES

- 1.** Create forms for all the tables created for School Management System in Chapter 9 and Chapter 10.
- 2.** Create the following reports :
 - Mark sheet of a student for particular term and year.
 - Fees collected by school in each month.
 - Total number of students studying in each standard.
 - For each standard, list the subjects and teacher who is teaching the subject.
 - Total presence of each student every month.





Today technology has become integral part of human lives. No aspect of human lives has remained untouched from it. The pace with which the technology changes, is unprecedented. Every few months we hear about a new technology coming into market. In this chapter we will look at some emerging technologies in the field of computing, entity recognition, digital photography, and data storage.

Emerging Trends in Computing

The activity that we perform with the use of a computer is generally termed as computing. Though the activities may vary from user to user, the usage of an electronic device capable of processing, storing and communicating with other devices becomes core of this activity.

We all at some point of time must have come across the categories of computers. The categories include supercomputers, mainframe computers, minicomputers, microcomputers and mobile computers. The first four categories required good amount of space for keeping the machine operational. The mobile computers today have revolutionized the use of computers and require minimal space.

The major impact of mobility came with the availability of laptop computers and wireless communication systems. Though smaller mobile computing devices, like Personal Digital Assistant (PDA) did exist. Its impact was not felt to a great extent due to high cost and limited functionality. With availability of communication technologies like Bluetooth, Wi-Fi, GPS etc. and Cellular data services like Global System for Mobile Communications (GSM), Code Division Multiple Access (CDMA), General Packet Radio Service (GPRS) and now 3rd Generation (3G) the scenario has changed completely.

This technology combined with nano technology has given rise to miniature devices. Smart phones and tablets are two such great inventions of this era. In this section we will discuss about two such path-breaking computing trends namely mobile computing and cloud computing. Both these computing trends have changed the computer usage scenario.

Mobile Computing

Smart mobile computing devices are indeed one of the game changing products of recent times. The typical mobile computing device has three primary components: software, communication, and hardware. The mobile devices today come with an inbuilt operating systems and set of applications. Some examples of mobile OS are Android, iOS, Symbian, Windows 8 etc. The inbuilt applications allow us to download games, interactive programs, content of newspapers, books, magazines etc.

The term communication in the above context refers to capabilities such as Internet connectivity and phone usage.

The mobile devices are becoming omnipresent and are reaching remotest areas of the world. With limitless capabilities that these mobile devices offer to the large extent of population, it has now become a necessary tool of our time. The main benefit that mobile devices have given us is that we can do things without being confined to one place. Previous to availability of this technology person had to be present at a specific place. However, today, with availability of various kinds of mobile computing gadgets, one can practically accomplish assigned tasks from any place in the world.

The mobile computing trend has become a great success due to supporting technologies like Wi-Fi and 3G (wide-area telecommunications network). These technologies allow us to be connected to the rest of the world wherever we might be. They allow an access to uninterrupted connection to the World Wide Web, and thus to a large pool of information. Some of the standard features that almost all mobile networks today support are Short Message Service (SMS), General Packet Radio Service (GPRS), Multimedia Messaging Service (MMS), Bluetooth and Wireless Application Protocol (WAP). These services combined give huge set of applications that can be used by users. Let us have a brief look at some of these technologies.

Wi-Fi

Wi-Fi is a wireless networking protocol that allows a Wi-Fi enabled devices to communicate without use of physical cables. It is an industry term used to represent wireless local area network (LAN) protocol based on the 802.11 IEEE network standards. Today it is the most popular means of communication used for wireless data transfer within a fixed location. This technology has formed the basis of creating personal or adhoc networks. Over a period of time, different standards of Wi-Fi, like 802.11a, 802.11b, 802.11g, and 802.11n have been evolved. All the four variants are almost similar except for the data rate allowed and distance the signal can travel. 802.11a gives the lowest data rate (6 Mbits/s to 54 Mbits/s) and has an outdoor range of 120 meters. On the other hand 802.11n gives the highest data rate (15 Mbits/s to 150 Mbits/s) and has an outdoor range of 250 meters. Almost all the mobile phones that you buy today are Wi-Fi enabled.

General Packet Radio Service (GPRS)

As the use of mobile devices increased, so did the user's expectation. GPRS became a cutting edge technology in this thrust of expectation. It took the mobile usage experience to next level by providing a packet switched technology that enabled data communications. This technology is used to allow the mobile phone connect to the Internet. Basically, any network connection that is not voice or text messaging uses a data connection like GPRS. It offers a tenfold increase in data speed over the circuit switched technologies. Theoretically the defined speed is up to 115kbit/s, however, practically the speeds are around 30-40 Kbps.

Using this technology a subscriber can remain connected Internet always.

3G and 4G

The terms 3G and 4G refer to third and fourth generation mobile communication standards. The 1G service was based on analog signaling and reached the data speed of 19.2 Kbps. As the popularity of mobile devices grew so did its customers. The growth led to digital networking and the service came to be known as 2G. It had increased network capacity and had data speed of approximately 1.2 Mbps.

Currently we are using 3G and 4G networking. The 3G service offers faster data transfers, up to speeds as high as 3 Mbps. It offers powerful multimedia services that allow organization to develop applications such as online billing systems and video conferencing. It also provides platform for many popular mobile phone games full of graphics and animation. Using this technology it is also possible to develop applications for mobile TV, instant messaging and video chatting. The only factor that might affect the usage is its subscription cost.

The 4G networks offer advanced mobile services such as video and movie streaming at much higher speed as compared to 3G. Theoretically 4G services can offer download speeds up to 100Mbps and upload speeds of 50Mbps. It has far expanded coverage and hence, offers more or less constant connectivity. The 4G networks are in its infancy stage yet, but its coverage may expand rapidly. The 3G and 4G networking both have a great deal to offer in terms of speed and quality of data communication.

Bluetooth

The mobile phone usage gave rise to the need of sharing data from one mobile to another. One could use SMS to exchange data, but it involved cost. A wireless technology called Bluetooth came as a boon for exchanging data over short distances. It used the short wavelength radio transmissions in the ISM (Industrial, Scientific and Medical) band for communication. This technology helps us create Personal Area Networks (PANs) with high level of security.



Figure 13.1 : Bluetooth Logo

The idea of Bluetooth was initially conceived by a company named Ericsson and later adopted by almost all other companies. Today it has become a standard for a small, cheap radio chip to be plugged into computers, printers, mobile phones and other electronic devices. A Bluetooth chip transmits data at a special frequency to a receiver Bluetooth chip. The receiver chip then gives this information to the computer, mobile phone or any other electronic device that it is associated with. You all must have seen a Bluetooth enabled mobile phone that can pair with mobile phone headset. Figure 13.1 shows the logo of Bluetooth, any device having this logo indicates that it is Bluetooth enabled.

Smartphones and Tablets

Cellular mobile phone has changed the dimensions of connectivity. Normal phones allow us to talk and send text messages. Another term that has become very common today is smartphone. But what exactly is a smartphone? Have you ever wondered how it is different from a mobile cell phone? How it has suddenly become smart? Well a smartphone is a device that lets you make telephone calls, and also provides added features like Personal Digital Assistant (PDA), send and receive E-mails, edit Office documents, use mobile Apps, play games, tune in to radio, stream movies, edit photos, get driving directions via GPS, instant messaging, provides Wi-Fi as well as Bluetooth support and create a playlist of digital tunes. Please note that the features available are non exhaustive and all cannot be mentioned here. Additionally the smartphone also has an Operating System, and a touch screen mechanism. If you look at the features, you will find that they are almost the features of a computer. Well that simply means that we are carrying a pocket computer along with us every time we carry a smart phone. The only drawback that becomes visible at the moment is the small screen size. Figure 13.2 shows a look of smartphone and tablet available in the market today.



Figure 13.2 : Smart Phones and Tablet

A tablet is a portable and mobile computing device. It has a larger touch screen and may have built-in phone facility. It generally uses onscreen virtual keyboard, a passive stylus pen, or a digital pen. It can perform all the operations that can be performed by a mobile phone as well as a computer.

The technology surrounding smartphones and tablets is constantly changing. The screen size, as well as the storage capacities is changing very fast. Hence what we call a smartphone today may change by next month, or next year or later and may become super smartphone!

Satellite Phones

We have already mentioned that the cellular mobile technology has revolutionized the connectivity aspect. But with all the technology in place, at times we still face a problem, problem of *network not available*. The cellular mobile phones have limitation that they need to be in the range of cellular towers. At many places it is still very difficult to erect such towers. Imagine the oil exploration companies working in high sea; we don't have a provision of erecting a cellular tower in the middle of the sea. What happens in such scenarios? How are we able to connect or transmit data in such cases? The solution to this problem is satellite communication technology and satellite phones.

A satellite phone more commonly known as *satphone* is one type of mobile phone. Instead of connecting to the terrestrial cellular towers it connects to the communication satellites orbiting around the earth. Satphones provide functionality like voice calls, text messaging service and low-bandwidth internet access, similar to the terrestrial cellular networks.

The coverage of satphones may include the entire Earth, or only specific region depending on its architecture. Generally the usage of satphones is controlled by the Government, in India we need to take special permission from the Government of India to use satellite phone. Inmarsat is the oldest satellite phone operator. It started its service by providing large fixed installations for ships. Thuraya is another leading satellite phone service provider. Its powerful satellites enable two third of the world's population to communicate using satellite phones. It delivers satellite communication solutions across Europe, Africa, the Middle East, Asia and Australia. Figure 13.3 shows a glimpse of satellite phone.



Figure 13.3 : Satellite Phone

As can be observed in figure 13.3 the satellite phones currently available in the market are almost similar to the regular mobile phones. Satphones are very popular when expeditions happen in remote areas where terrestrial cellular service is not available. With the use of satphones we get uninterrupted connectivity though at a higher cost.

Cloud Computing

The next big thing available to the users today is a technology called cloud computing. Cloud computing allows users to have complete access of their data or resource from any part of the world. The cloud computing facility is available for both personal and official usage. The first type of usage is called Personal Cloud Computing. A typical example of personal cloud computing scenario is Google drive. Google provides us a free drive space of 5 GB shared between Google Drive and Google+ Photos so that you can keep your files, email, and photos accessible from any device, anywhere. This option is always better than carrying a pen drive. Figure 13.4 shows a typical GUI of Google drive.

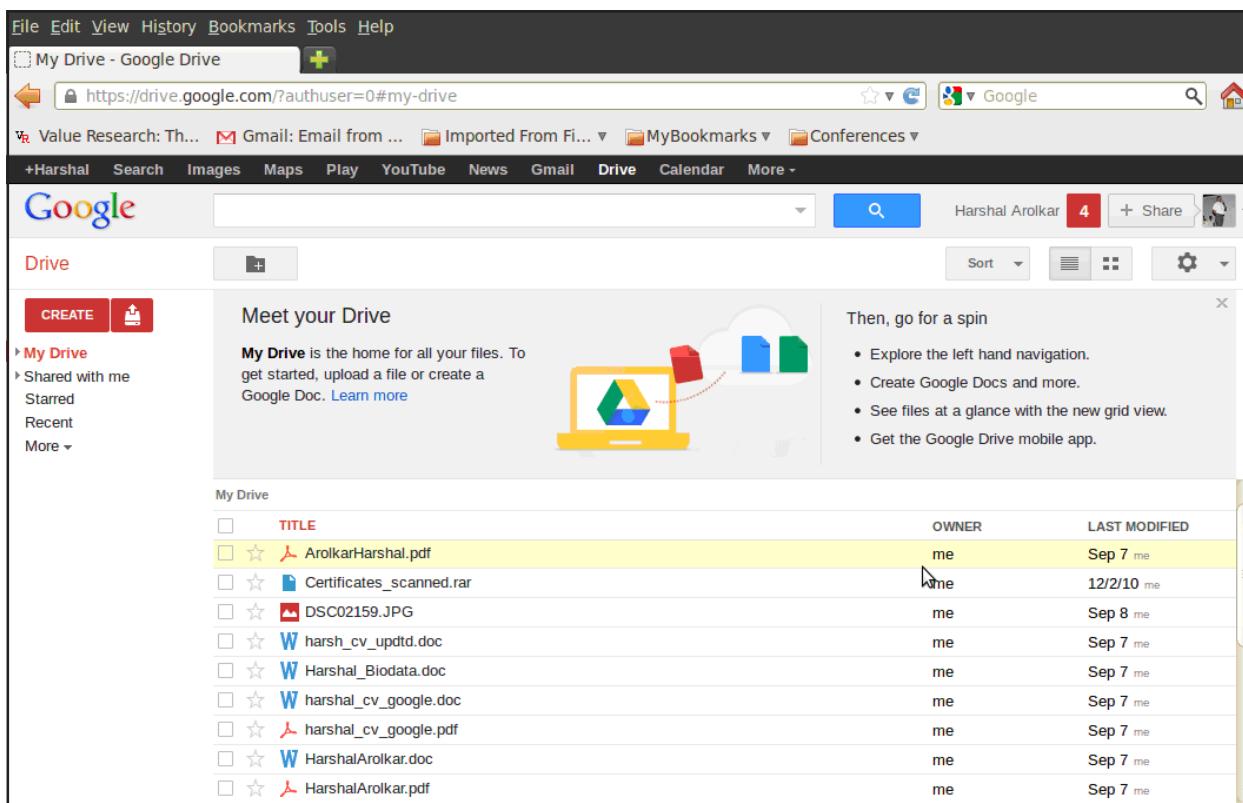


Figure 13.4 : GUI of Google drive

One of the many benefits of the personal cloud is the feature of automatic synchronization. This feature comes in handy when user wants to keep copies at multiple places. Say for example, if a user searches for video on web and purchases this video using mobile device, then this video file not only downloads to their mobile device but also downloads automatically to all the devices that are linked with the cloud account. It is possible to sync the cloud account with multiple devices like user home desktop, office laptop and the mobile phone.

Ubuntu One is the personal cloud platform that allows us to keep all our digital data together. We can use the content, our way, wherever we are. With Ubuntu One suite of cloud services that includes storage, sync, sharing and streaming, we have immediate access to our music collection, favorite photos, videos, important documents and more, at any time and from any device. It was launched in May 2009 to provide sustainable revenue for Ubuntu, the free operating system, which is commercially backed by Canonical. To create your account on Ubuntu One, visit <https://one.ubuntu.com/>. Figure 13.5 shows the home page of Ubuntu One.

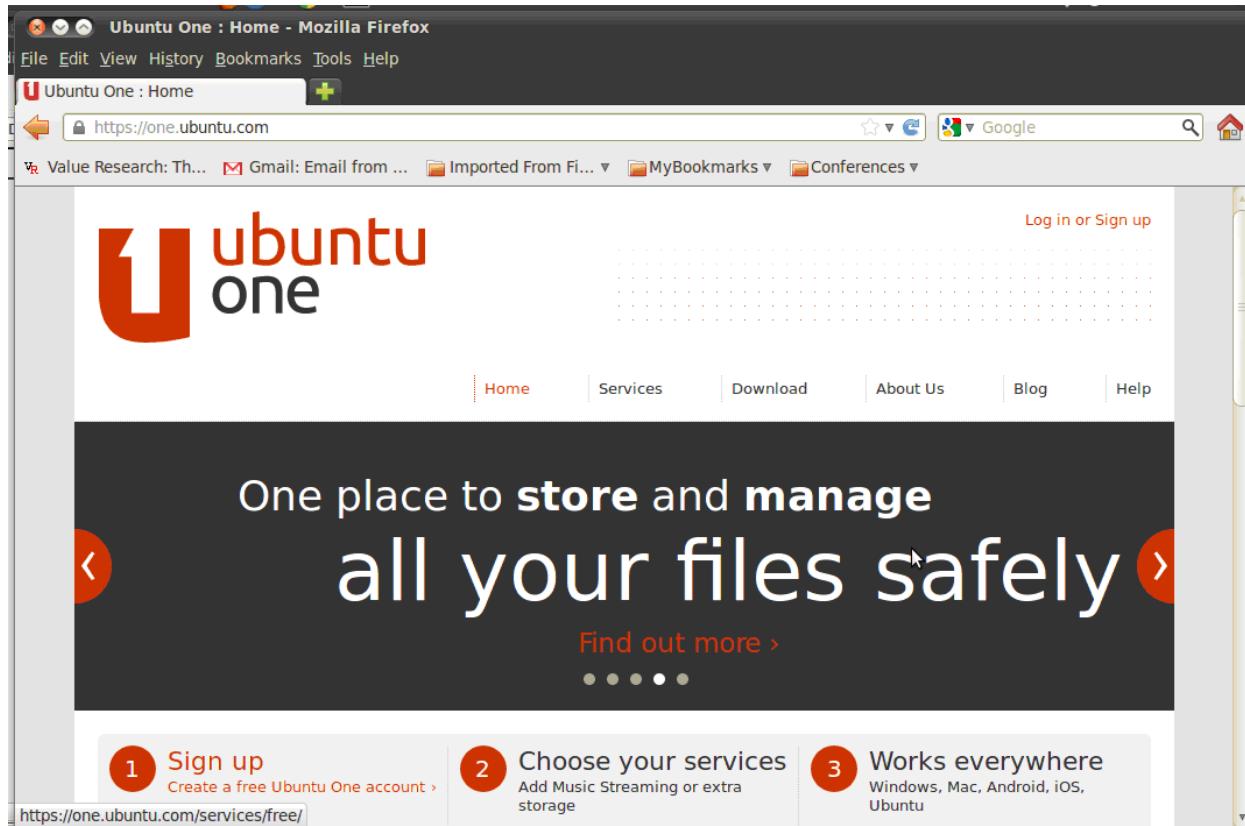


Figure 13.5 : Home page of Ubuntu One

Businesses have also benefitted greatly from the model of cloud computing. For businesses the cloud computing scenario is available in different models namely Software as a Service (SaaS), Infrastructure as a Service (IaaS), and Platform as a Service (PaaS). Depending on the requirement of the business organization one may choose any of these models. Let us see the basic concept of each of these models in brief.

Software as a Service (SaaS) : A business organization instead of developing its own software application generally uses the software by making payment to the cloud service provider. The service provider hosts the software at its web server, so that the company does not need to install it. The service provider will also manage the software on behalf of the company. This model relieves the business organization from buying, installing and maintaining software or hardware. All they need is an Internet connection that allows them to connect to the software and use it. SaaS examples include customer relationship management as a service. Salesforce.com is one of the most used cloud service for customer relationship management (CRM) software. Figure 13.6 shows the home page of <http://www.salesforce.com/in/>.

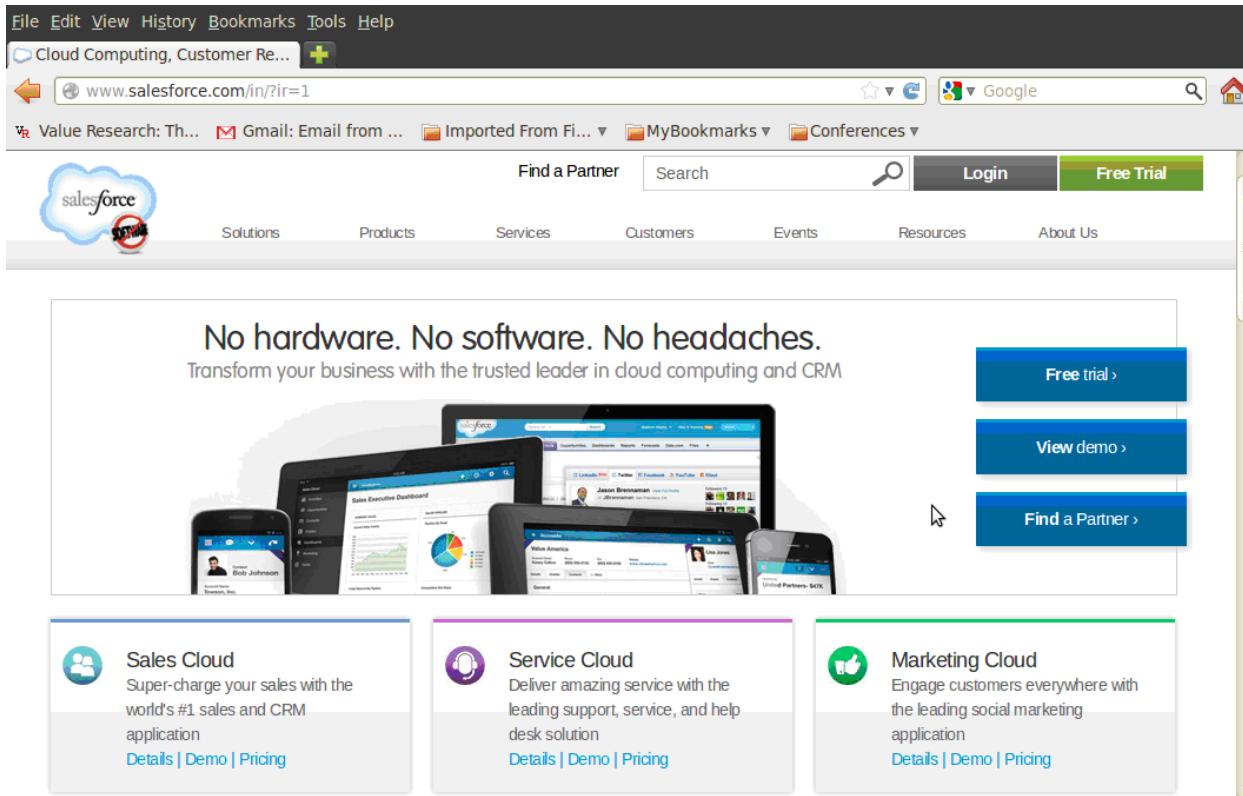


Figure 13.6 : Homepage of <http://www.salesforce.com/in/>

Infrastructure as a Service (IaaS) : A business organization may hire the hardware infrastructure as a service from the cloud service provider instead of establishing its own physical resources. The hardware at cloud is used to support different operations such as computation, storage, printing, backup, and networking components. The service provider owns all the equipments and is responsible for housing, running and maintaining it. Some of the IaaS providers are Windows Azure Virtual Machines, Google Compute Engine, and HP Cloud.

Platform as a Service (PaaS) : A business organization here is generally a software company that creates the software using tools and/or libraries from the service provider. It also controls software deployment and configuration settings. The service provider provides the networks, servers, storage and other services that are required for developing software. This model facilitates the deployment of software applications without the cost and complexity of buying and managing the underlying hardware and software and provisioning hosting capabilities. Many companies have ventured into the domain of cloud computing. Figure 13.7 shows a glimpse of some of these companies.

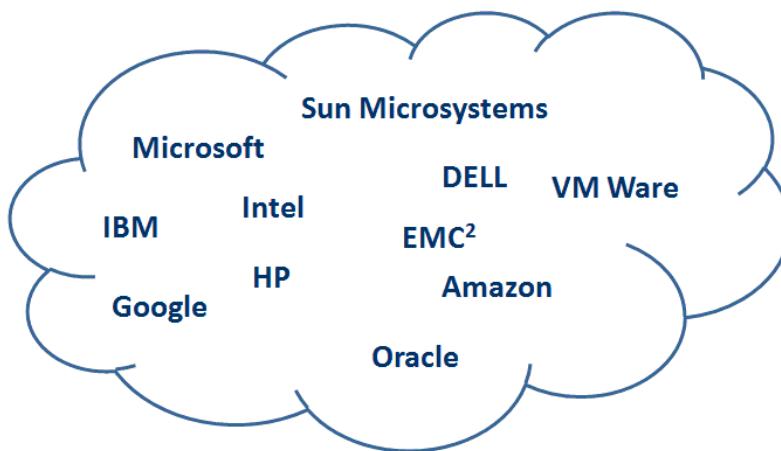


Figure 13.7 : Companies providing cloud services

The cloud computing scenario can be distinguished from normal scenarios by its key characteristics such as it is usually off site, accessed over the internet, minimal IT knowledge is required to access the service, available on all platforms from desktop to mobile phones. Today cloud computing facility is extremely useful for a consumer and for the business for it saves the establishment costs.

Emerging trends in Object/Human Recognition

Today we are living in the world of ubiquitous computing, wherein we are surrounded by computers and sensing devices throughout the physical environment. Though present these computers are effectively invisible to the user. Such ubiquity has been made possible due to availability of cloud computing platforms. One of the major problems in ubiquitous computing domain is the identification of the objects like a specific computer or sensor. In this section we will learn about some techniques like Radiofrequency identification (RFID), Biometrics, Global Positioning System (GPS) and speech recognition that are used for identifying objects.

Radio Frequency identification (RFID)

One of the most widely used technology for identification and tracking of objects used in the market is radio frequency identification system. It consist of three parts; an antenna, a transceiver and a transponder.

The antenna is used to generate radio frequency signals within a relatively short range. The range can vary from 10 centimeters to 200 meters. The antenna may be handheld or fixed and virtually can be mounted on any surface from a computer to a wall.

The transceiver reads the radio frequency signal and transfers the information to a processing device generally a radio frequency reader. It simply works as a decoder to interpret the data that is stored in the transponder.

The transponder is the RFID tag that is attached to the object. The tag is a programmed microchip and is capable of storing information of around 2,000 bytes or less. Further, the tags can be passive or active. The passive tags do not require battery for working while the active tags come along with their own battery.

The transmission of data from RFID tag happens only when the tag passes through the field of the scanning antenna. The tag then detects the activation signal from the antenna, this activates the RFID tag, and it transmits the information on its microchip to be picked up by the scanning antenna. The RFID systems are used in many applications like tracking parts for just-in-time manufacturing, tracking purchase of customers in shopping malls etc. Figure 13.8 shows a brief idea of the implementation of RFID systems.

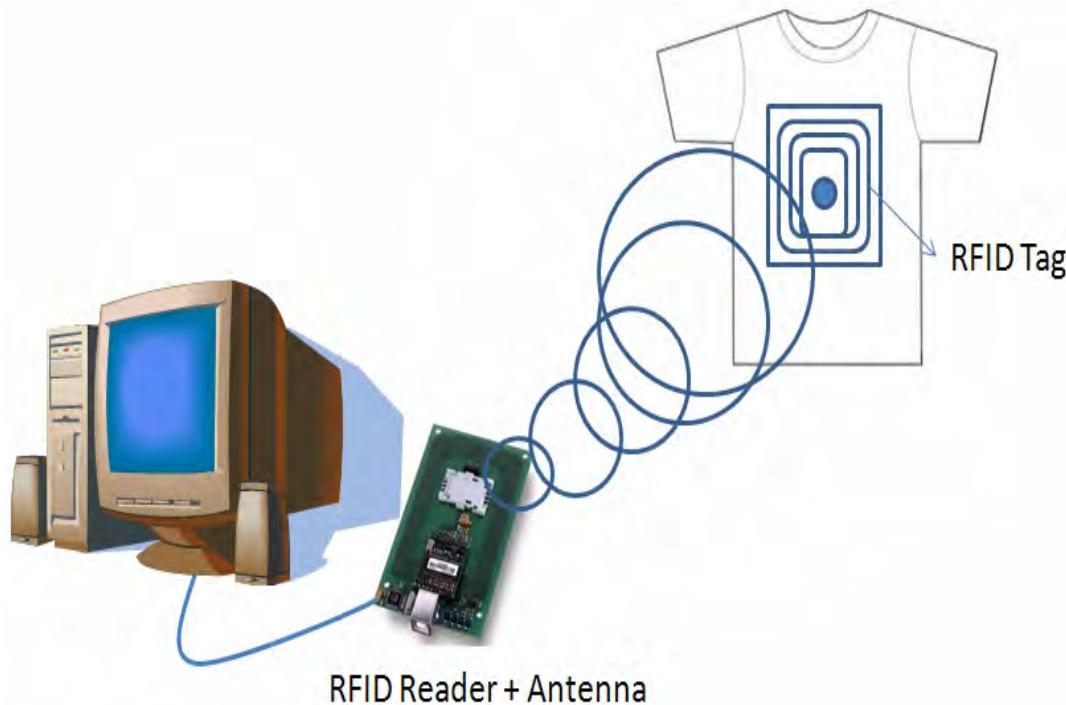


Figure 13.8 : Working of RFID system

RFID technology is being used in many industries for different purposes. In 2008 Beijing Summer Olympic Games, RFID technology was used for high tech ticketing and food tracking. RFID embedded tickets in Beijing games were intended to battle counterfeiting of tickets, to provide speedy entry for visitors at venue gates and to guard against unauthorized access to specific areas of the Olympics compound.

It also helped security personnel monitor the hotels, venues, manufacturers, distribution centers, hospitals and provided support for food safety tracking system in Olympics. The RFID system used in food tracking system monitored production, processing and transport of food packets to different venues.

Animal tracking has been one of the earliest applications of RFID. The livestock are considered to be assets for an owner. As they are constantly on the move it becomes difficult to manage large herds. RFID technology is used to track such constantly moving herds. The RFID tags are attached onto a livestock (in form of earring, neck belt etc) or implanted inside livestock as radio signals do not harm the body. Once installed these RFID tags enables the recording and tracking of individual animals.

We can also create a livestock database that keeps track of the animal's origin, parentage and breed using these unique tags. The RFID usage also helps monitor health of the animals, allows us to keep individual medication and vaccination records, and prevent the spread of disease by tracking and segregating infected livestock at early stages. With lots of proven benefits accruing from the use of RFID technology, it is now becoming a regular part of modern farm management.

RFID technology is also used in libraries nowadays. It is used in library for tracking the movement of books and theft detection. The RFID based library system, consist of smart RFID labels used instead of normal bar codes, the RFID readers (fixed or hand held) and library management software. Using such a system allows libraries to manage their day to day activities more effectively.

To start with we need paper thin flexible smart labels, the size of which may vary as per the need of library but in most cases a label of approximately 2"X2" size will suffice. This label is placed on the inside cover of each book in a library such that it is not obstructed during opening or closing the book. The label has a small antenna and a tiny chip that stores information about the book including a unique Accession number to identify each entity. The information contained on microchips in the labels can be read from any place within the library using RFID scanner/reader. We may additionally require a wireless communication system to transfer details of the label into library information system.

The technology when used facilitates easier and faster issue and return mechanism of books. The use of this technology also reduces the amount of time required to perform circulation operations as the information from RFID tags can be read much faster than from barcodes. It is also possible to read details of multiple books from one rack at a time without removing them. Thus a hand held reader can be moved rapidly across a shelf of books to read all of the unique identification information. This further helps in maintaining proper inventory and also allows us to check if the items are out of order.

Most of us at some point of time must have gone to supermarkets (big shopping complexes). Many people visit such place for buying products. You must have observed that at the exit gates of such supermarkets you are able to hear a strong beep sound. This is a security mechanism used to stop theft of products. The products contain an RFID tag which is generally removed by the person who generates the bill. In case he/she forgets to remove the tag or someone tries to take of the things forcibly the alarm is raised.

The product purchasing process in supermarkets is often very time consuming, it comprises of the time spent in corridors, time spent to search the product location and finally the time spent in the checkout queues. The RFID technology is now also being used to reduce the time a customer spends to buy a product. The supermarkets now have shopping cart that has a touch-screen monitor, a client card automatic reader, a positioning transmitter and a product reader.

When the customer arrives at the supermarket he takes one such cart and logs into the markets system using his client card. The system after welcoming the customer displays the list of the items that he/she had last purchased. A customer may add more items or remove some items from this list. Then With the help of the cart's navigation system, the customer is directed to the shelves or places where the products in list can be found. As soon as the customer places the products in the shopping cart, the system starts calculating the amount that needs to be paid by the customer. Once all the purchase is made the customer heads towards the exit. As all the process of calculating the bill is over he simply needs to pay the amount and move out of the complex. Many applications of RFID have come into existence today, but discussing all is out of scope of this book.

Biometrics

With the increased use of technology it has become necessary to control the access to information. Many methods like *access rights*, *password protection*, *encryption* etc. have been used to control the access. However, for better access control a new technique known as biometrics is now being extensively used in many applications. Biometrics (or biometric authentication) generally involves identification of humans based on their characteristics or behaviour. These human body characteristics include fingerprints, eye retinas and irises, voice patterns, facial patterns, signature and hand measurements. A biometric system generally is made of five components.

1. A sensor that is used to collect the data,
2. Set of signal processing algorithms that performs quality control activities and develops the biometric pattern,
3. Data storage that keeps information to compare the new biometrics patterns with,
4. A matching algorithm that does the actual comparison and
5. A decision process (either automated or human-assisted) that uses the results from the matching algorithm to make decisions.

Today biometric technology is found in many aspects of our daily life. We will find it in applications related to payment of phone bills to accessing personal computers. This section discusses about fingerprint, iris and voice recognition techniques of biometrics used in identification and authentication of users and access control.

Fingerprint Recognition

Fingerprint has been used as an alternate to sign for more than a century now. Generally a person who does not know how to write, when he/she needs to sign a legal document puts his/her thumb on the ink pad and creates its impression on the document. It is an authentic mode of identifying a person.

The modern fingerprint recognition mechanism uses the impressions made by the unique, minute, ridge formations or patterns found on the fingertips. It has been observed by researchers that though two fingerprint patterns are similar, two fingerprints do not have identical individual ridge characteristics. Also the patterns once formed do not get distorted except under some unusual circumstances like accident.

Figure 13.9 shows example of fingerprint features and a fingerprint reader used to mark attendance. Most of the laptops available in the market today provide fingerprint recognition as mode of authentication.



Figure 13.9 : Features of fingerprint and Fingerprint reader

Table 13.1 lists some of the government, forensic and commercial applications of finger print recognition.

Government	Forensic	Commercial
<ul style="list-style-type: none"> Identification cards like AADHAR, Ration Card, BPL Cards. Driver's License Welfare Disbursement Passport Control, etc. 	<ul style="list-style-type: none"> Criminal Investigation Corpse Identification Criminal Record Keeping Terrorist Identification, etc. 	<ul style="list-style-type: none"> Computer Network Logon Electronic Data Security Smarts Cards, like ATM Cards, Debit Card Facility Access Control Personal Digital Assistant Distance Learning, etc.

Table 13.1 : Some application areas of fingerprint recognition

Iris Recognition

Most of the readers who are reading this book must have gone through the process of acquiring AADHAR card. If you remember the process then at one point of time you were

asked to look into a device that captured your iris information. The iris is a muscle within the eye that regulates the size of the pupil, and controls the amount of light entering the eye. It is the colored portion of the eye with coloring based on the amount of melatonin pigment within the muscle.

The iris generally does not change during the person's lifetime. The characteristics of iris do not change even after surgeries and furthermore it has 266 unique characteristics (discussing them is out of the scope of this book). The iris recognition systems used in the market uses up to 173 characteristics. The primary visible characteristic of the iris is the trabecular meshwork, tissue which gives the appearance of dividing the iris radially.

An iris recognition system requires a small high resolution camera to capture picture of the iris. It generally captures a black and white image. It relies heavily on infrared imaging to take such a picture. The application that provides physical access to users requires a user to stand within 3 to 10 inches from the camera. It then centers the iris in a mirror within an area of 1 inch square directly in front of the camera. We also get such systems in laptops nowadays to control logical access to computers. It uses a desktop camera and works well within the distance of about 18 inches to capture the iris image. Some of the major applications of iris recognition today are mentioned below:

- Access to bank accounts at cash machines
- Anti-theft devices
- Authenticated access to buildings and homes
- Automated international border crossing (used as a substitute of passport to stop illegitimate entry into a country by unauthenticated person)
- Automobile ignition and unlocking
- Aviation security
- Biometric-Key Cryptography
- Cell phone and other wireless device based authentication
- Controlling access to restricted areas
- Creation of a “watch list” database used by Government to keep track of terrorists
- Database access and computer login

Voice/Speech Recognition

Voice or speech recognition is a biometric technique that uses an individual's voice as input to control the activity that can be done by him/her. It relies on features influenced by both the biological structure of an individual's vocal tract and the behavioral characteristics of the individual. Although we call it voice/speech recognition the correct term is voice/speech verification.

The technology is being used by banks to verify identity of the caller when making telephone transactions and performing account maintenance. It is also used extensively by mobile phone users to make a call directly by speaking out the name of the person whom call needs to be made. In forensic science it is used to determine the truthfulness of responses a witness gives as part of an investigation.

Similar to other biometric systems to work with voice recognition systems you need to first train the system. In the training period the voice sample is first converted from an analog format to a digital format. The features of the individual's voice are extracted, and then stored as a pattern for further use. The environmental factors like sound in the surrounding, problems in the recording device etc do play a role in efficient working of the system. Some of the application areas of voice/speech recognition are as mentioned below:

Playing back simple information : Used as forensics to identify the voice of a person.

Call steering : Routing of regular callers to appropriate departments or person based on their voice match. It is used in customer service segment.

Automated identification of caller : It is used in customer service segment to identify the caller.

Removing Interactive Voice Response (IVR) menus : It is an improvisation in IVR and automated call handling systems that are often unpopular with customers. This technique removes the push button menus and replaces it with direct call steering.

Voice enabled operations : Enables the user to dial a number or perform an operation on mobile phones by issuing a voice command. Such applications are very useful for blind or partially blind users. We can also open an application window, type a letter or shut down the computer by giving voice commands. Most operating systems today support such facilities.

Global Positioning System (GPS)

The Global Positioning System (GPS) is one of the widely used systems for navigation applications. It is a system created by the U.S. Department of Defense. The system consists of a network of 24 satellites placed into orbit above earth. The GPS was originally intended for military applications, but in the 1980s, the government made this system available for civilian use. It is a very robust system that works in any weather conditions, anywhere in the world, 24 hours a day. As of now there are no subscription fees or setup charges to use GPS. Any device having a GPS receiver can use this system. Almost all the smart phones today come with inbuilt GPS receiver. Alternatively one can download apps that use GPS and help in navigation.

The GPS technology has been found useful in many application areas such as, fleet tracking, cellular telephony, disaster relief and emergency services, aircraft tracking, robotics etc. To pin point the exact location of an object the GPS satellites circle the earth twice a day in a very precise orbit and transmit signal information to earth. The GPS receiver take this information and uses a process called triangulation to calculate the user's exact location. The GPS receiver compares the time a signal was transmitted by a satellite with the time it was received. The time difference tells the GPS receiver how far away the satellite is. This process is repeated for two to three more satellites and the receiver then can determine the user's exact position and display it on the any electronic map available with the user. To calculate only latitude and longitude the GPS receiver uses at least three satellites. In case of additional information, such as speed, behavior, track, trip distance, distance to destination, sunrise and sunset time and more is needed the GPS receiver needs to be connected with four or more satellites. Some interesting applications of the GPS technology are mentioned below:

Agriculture: It is used for measuring area, preparation of digital agricultural maps, and preparation of digital land ownership maps.

Disaster management : It is used in search and rescue operation, for example lost hikers and adventurers, person buried under debris or under water can be found if they have a cell phone or GPS system that is on.

Military: It is used in different aircrafts like fighter planes, aerial tankers, bombers etc. for accurate identification of position of targets as well as the aircrafts. A navy ship also uses it for the same purpose. It is also used on satellites to obtain highly accurate orbit data and to control spacecraft orientation.

Tracking Deliveries and Shipments: Most courier services use GPS tracking system to monitor delivery of goods to customer. Likewise transportation system uses it to monitor their transport system. A company using this technology can see if its vehicle has been delayed and suggest a faster route, alternatively it can also find out the current location of its vehicle.

Recovery of Stolen Property: Security services can determine the exact location of the stolen vehicle or a device such as mobile phone or a laptop equipped with GPS and can find the vehicle or device very easily.

Emerging Trends in Storage devices

You must have come across client server applications where one machine works as a resource provider while the other works as a resource user. File server is one such application; it is used to provide resources to all the client machines connected to it. Generally a file server is a machine that does login authentication of the users and based on the access control policy set by the administrator allows restricted access to a file resource. The access to such a resource is slow as the processor of the server gets loaded with multiple activities. With availability of high end machines with faster

processing capacities the scenario of data storage has also changed. Today we make use of USB supported external hard drive to create additional storage for our data or simply to take a backup of our data. Also the servers come with attached storage with the capacity of Terabytes. But managing such external devices is always cumbersome. This section discusses about some recent trends in data storage devices.

Network Attached Storage (NAS)

Networks have been used from decades to expand the capacity of computing in an organization. We know that it is possible to give prints on a network printer (a printer that is connected to a network as a device, rather than being connected to a computer). Such a network printer has its own identity (unique IP address) that makes it work faster than a printer shared on a computer. Network Attached Storage (NAS) is a similar technique; it is hard disk storage mechanism that is assigned its own network address rather than being attached to an individual's computer.

The NAS device may consist of a single Terabyte hard disk or array of such hard disks. It is attached to a local area network and assigned a unique IP address. When the user requests for a specific file, the file requests are mapped by the main server that is used to authenticate the user, to the NAS file server. By removing storage access and its management from the authentication server, both application programming and process of sending and receiving files become faster as they are now not competing for the same processor resources.

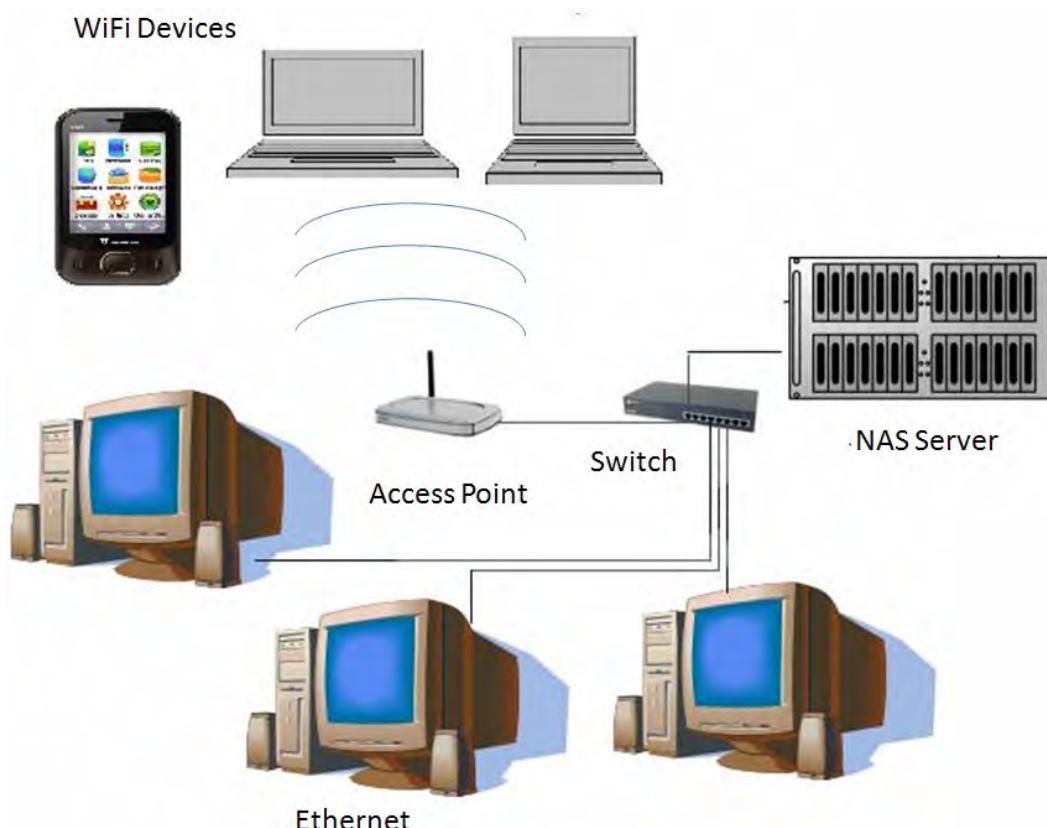


Figure 13.10 : NAS Architecture

With a NAS system in place, we can quickly and simply store and share our data such as music, videos, images and other files from one convenient location. Also the access to these resources can be obtained by heterogeneous clients like, a desktop, laptop, digital camera or a smart phone. Figure 13.10 shows a typical implementation of NAS in an organization.

Storage Area Network (SAN)

A storage area network (SAN) as the name suggests is a dedicated network for storage. It provides access to consolidated, block level data storage. The basic idea of SAN is to allow an application server to have uniform access to different storage devices such as hard disk arrays, tape libraries, and optical storage devices. Due to this uniform access feature these heterogeneous devices appear like locally attached devices to the operating system. Figure 13.11 shows a typical implementation of SAN in an organization.

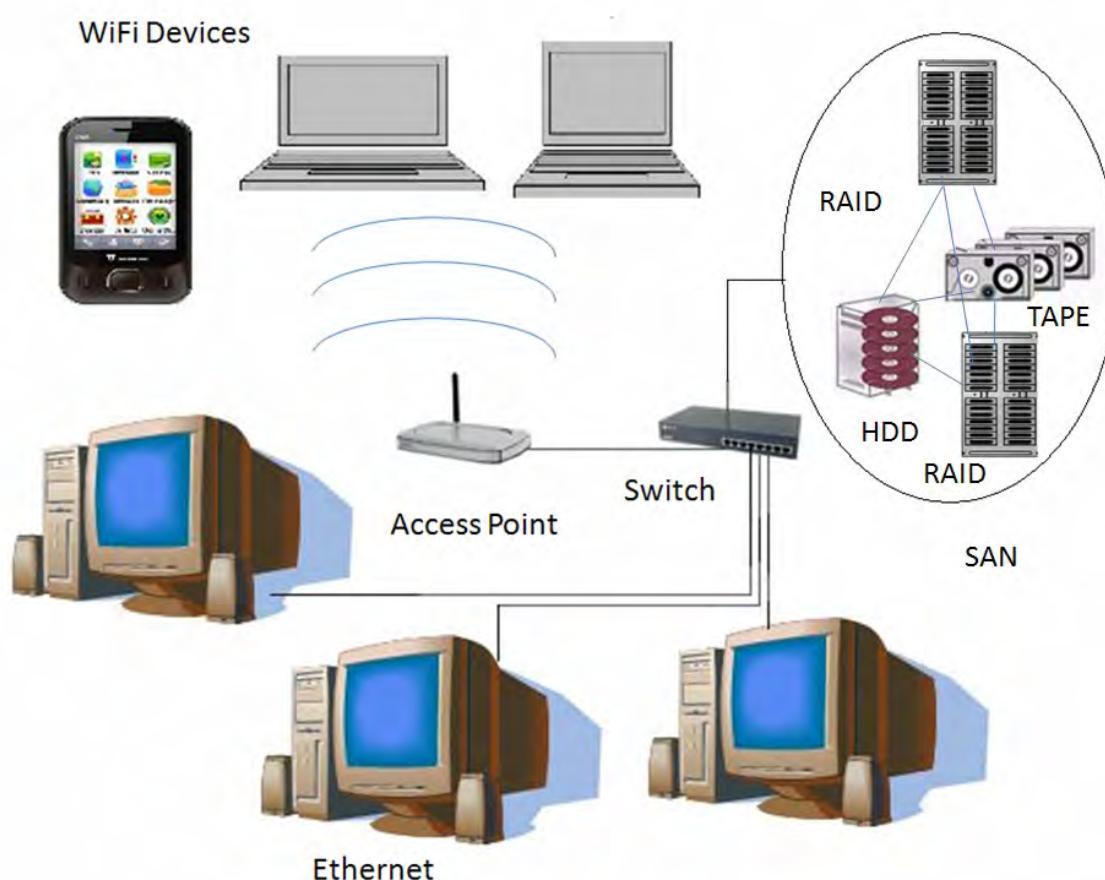


Figure 13.11 : SAN architecture

Observe that SAN is a type of local area network that is designed to handle large data transfers. It supports operations like data storage, retrieval and replication. To take the benefit of this technology we need high end servers, multiple disk arrays and fiber channel interconnection mechanism for communication.

Online Storage options

We have learnt about Google Drive in the cloud computing section. With the advent of cloud computing facility we have also got lot of online storage options. The online storage options can be used for both personal as well as business use. Online storage options are helpful especially when we want to keep backups and require universal access to our data. The beauty of this option is that we can access the data from anywhere in the world, from any device. The only requirement is good internet connectivity. Some of the free online storage options available are ADrive, Microsoft Live Sky-Drive, 4Shared, Mozy, Dropbox, Box, MediaFire, DivShare etc. They offer different services in online storage and provide us with different capacity of storage space. Similar to free options we also have paid options. Such service provider guarantees more security and service parameters as compared to the free service providers. IBackup and Amazon S3 are example of paid storage options. The online storage options in a way can be considered cloud services.

Computer Controlled Devices

The computers are becoming smarter and intelligent day by day. We are quite fascinated by this newer capability of the computers. No wonder if we say that sometimes it becomes really difficult for us to know whether the person on the other side of our telephone conversation is a human or a recorded voice in the machine!

We have heard about automations in organization, wherein machines replaced activities that were performed manually. For example instead of filling the cold drinks bottle manually we now have bottling plants. Machines controlled by programs are used here to fill the bottles of different shapes and sizes. With the advances in computer hardware and software technology, one field that has also grown is the field of robotics. The study of robotics is to mainly understand how effectively a robot can work with the use of sensors and actuators in dynamic environments. Today robots are highly automated mechanical and electronic manipulators controlled by computers.

Some of the application areas of robotics are intelligent home, military applications, farming and animal husbandry, vehicle industry, hospitals, outer space expeditions, disaster management and entertainment. Let us see in brief how a robot is used in these applications.

Intelligent Home : Today automated systems can monitor home security and other indoor activities. Doors and windows of a house can be opened automatically based on certain activity or commands. Electronic appliances such as light, fans, heater, air condition etc. can be pre-programmed to start, stop or change its setting in between the operation. This technology assists the occupants irrespective of their state of mobility. This technology can be used by persons who can move easily within the house or by a person who may be bedridden with an equal ease.

Military Applications : Surveillance is a key to safeguard from activities happening in and around your surroundings. If you remember the scene from movie 3 idiots, the object that Amir

Khan rectifies was also intended for surveillance. Airborne robot drones are used for surveillance in today's modern army. An airborne drone is a small flying machine with the shape of bird or any other object fitted with very high definition cameras, sensors and communication as well as networking tools. It is used to collect vast amount of data of any object of interest from an individual to a country.

Farming and Animal Husbandry : Agriculture is one such field where lot of robotics have been tried and tested. To name a few, automatic water allocation based on soil conditions, automatic harvesting of crop, milking of cows etc.

Vehicle Industry : Vehicle industry uses most of the automation tools available. It uses robotic arms that are able to perform multiple tasks during the car manufacturing process. The tasks that these robots can perform but not limited to are welding, cutting, lifting, sorting and bending. Vehicles today come with additional processors that make these vehicles smart compared to their previous versions. Navigation systems based on GPS are not new in vehicles today. It is being used for getting directions, looking for a place, making sure that the path taken is correct and many more things.

Google has succeeded in introducing the Google driverless car. The project supported by Google is developing technology for driverless cars. The U.S. state of Nevada passed a law on June 29th, 2011 permitting the operation of driverless cars in Nevada and California. The Nevada law came into effect on March 1, 2012, and the Nevada Department of Motor Vehicles issued the first license for a self-driven car in May 2012. The license was issued to a Toyota Prius modified with Google's experimental driverless technology. Figure 13.12 shows driverless Google car.

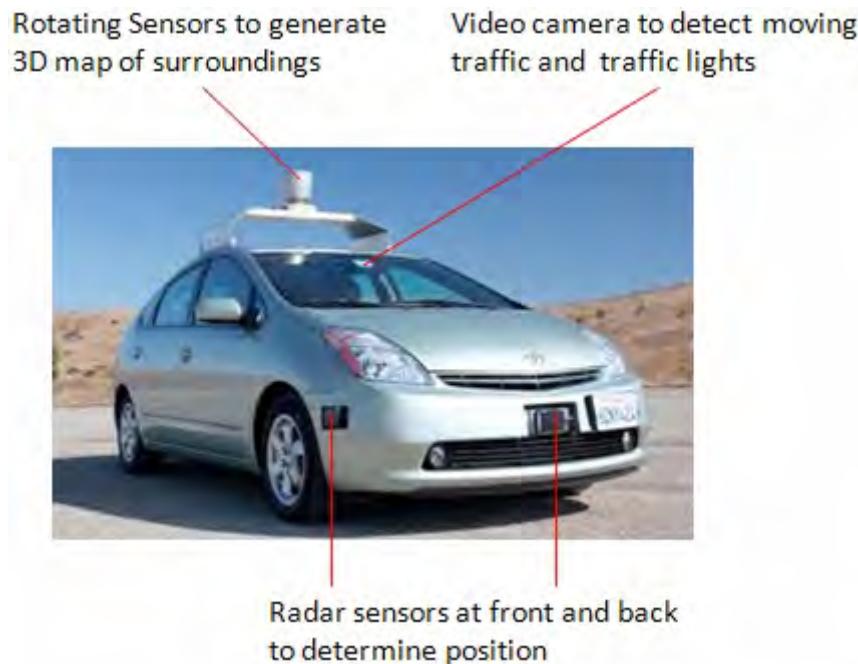


Figure 13.12 : Google driverless car

To start with the driverless project team has equipped ten vehicles, consisting of six Toyota Prius, an Audi TT, and three Lexus RX450h with required technologies. The system is purely based on sensors and navigation map stored in the car. The car drives at the speed limit stored on its maps. It maintains its distance from other vehicles using its system of sensors. It also has an override mechanism that allows a human driver to take control of the car. In the testing phase two people were seated in the car one driver and another engineer. The driver was supposed to take control of the car in case of any mishaps, while the engineer was supposed to correct any software malfunction if it happened.

Hospitals : Health care is another area where intrusion of robotics is saving lives and assisting people to improve their lives. Researchers have developed, robotic suit that enable nurses to lift patients without damaging their backs, a robotic limb linked to tiny sensor in brain called BrainGate that works on thoughts of person rather than any physical action as well as a robotic arm that can perform precision surgery.

Outer space expedition and disaster management : Robots can visit environments that are harmful to humans like volcanoes, deep oceans, moon etc. This makes them a good choice in terms of rescue operations in case of earth quakes and other natural calamities.

You must have heard about India's moon mission called Chandrayaan - 1. It was one of the unmanned moon mission launched by the Indian Space Research Organisation (ISRO) in October 2008, and operated until August 2009. A NASA instrument onboard Chandrayaan - 1, revealed water molecules in amounts greater than predicted. Water on the moon is just one of the many crucial moon findings of this mission.

Entertainment : Entertainment industry has also not remained untouched of the robotics trend. Interactive robots that exhibit behaviors and learning ability have been made by companies like Sony and Honda. The robot manufactured by Sony called QRIO (Quest for cuRIOsity), originally named SONY Dream Robot is humanoid entertainment robot that can move freely, play with a golf ball and can respond to verbal instruction. While the robot manufactured by Honda called HEARBO can pick up, distinguish, and analyze multiple simultaneous sound sources without difficulty. It can easily differentiate between sounds made by children playing on one side of the room, from a doorbell ringing on the other. Figure 13.13 shows a QRIO saluting the audience.

One of the fields that helped computer controlled devices usage grow significantly is Artificial Intelligence (AI). Human being are gifted with intelligence, it gives us the ability to acquire knowledge and skills. It also allows us to apply the knowledge or skill acquired to solve problems, irrespective of whether they are trivial or complex. The field of Artificial Intelligence intends to provide a computer the ability to acquire knowledge and use it in real time scenarios.



Figure 13.13 : QRIO saluting the audience

A simple example of adding AI to a computer is a chess program that stores the step it has taken against a particular move of all opponents and its outcome. Next time when some opponent takes the same step, the computer analyses all its previous moves and decides to choose a move that may help it win.

Including AI in computer systems makes it think rationally like human. It also allows the systems to act rationally like humans. Scientists and engineers are making all their efforts to make computers intelligent enough to compete with humans.

Emerging Trends in Digital Photography

The art of taking photographs or photography has been around for many years. It started in 1840s, when William Henry Fox Talbot, combined light, paper, a few chemicals and a wooden box to produce the first photographic print. This process of producing a photograph has been revised again and again. The major change came with the technological advancements and innovations such as the digital cameras. It allowed photography to advance from a chemical process to digital one that involves bytes of information.

Digital photography is similar to conventional film photography. Digital cameras also use light to create images. The only difference is that instead of using the film to store the image, digital cameras capture images as matrix of pixels. The image becomes sharper with more number of pixels. This is the reason we see 10 megapixels, 12 megapixels etc while looking at the specification of digital camera.

The scenario of digital photography using digital cameras has recently changed with the advent of mobile phones with inbuilt cameras. We get 2 megapixel cameras in a basic model of mobile phone.

In year 2012 Nokia came out with 41 megapixel camera. It promises 7728×5354 resolution image. The whole idea of taking a photo and sharing it with our friends in real time on social networks is very fascinating. The entry level digital cameras are facing lot of competition from smart phones due to their capacity of taking photos as well as sharing capacity. To tackle the issue mentioned Samsung has come up with an Android based camera called GALAXY. Figure 13.14 shows the front and the back view of Samsung Galaxy camera.



Figure 13.14 : Front and the back view of Samsung Galaxy camera

Just imagine a camera with an Operating System. Samsung claims that the camera is the only truly connected camera in the world as it support 3G and Wi-Fi. We can connect, upload, share, and even post on our favorite social media. It also has auto cloud back up feature that automatically saves our precious photos into the cloud the instant we take them. To be called as a smart camera it also listens to our voice. The camera is able to perform operations based on our voice command.

Internet Enabled Television

The simple phones have now become smart phones thanks to Internet technology. The televisions have also gone through a drastic technological change from CRT to LED. Today we get flat smart televisions that can be wall mounted and are Wi-Fi enabled. Such televisions have ability to connect directly to the World Wide Web and display content such as YouTube videos, weather reports and streaming movies or television shows. We do not require any computer or outside equipment to display these contents.

The technology is still in its infancy and hence the Internet contents that can be viewed by users vary from manufacturer to manufacturer. For instance, some Internet-enabled TVs are capable of streaming data from YouTube, Amazon Video-On-Demand, Picasa Web Albums, Bloomberg News and a weather channel. Other Internet-enabled TVs additionally display material from eBay and Twitter. Further some Internet-enabled televisions allow us access to Netflix's vast library of films and TV shows. The Internet enabled televisions also come with embedded OS and you can connect the television to the Internet using wireless medium or wired Ethernet connection.

Green Computing

Whenever we use a computing device we tend to use a resource that might cause problems to environment. Use of technology always has two sides of coin, one that improves the quality of life and the other that might affect the environment. Global warming is one such side effect of excessive use of technology. Green computing is the term that signifies efficient use of computing resources.

The main goal of green computing initiative is to minimize the environmental impact and maximize economic viability when using a computing resource. ENERGY STAR program launched in 1992 was the first step towards green computing. The program awarded labels to computing products that minimized the use of energy and maximized the efficiency. The labels could be obtained for products like computer monitors, television sets and temperature control devices like refrigerators, air conditioners, and other such similar items.

The sleep mode function of computer monitors was one of the first results of green computing initiative. This mode is now available in many other devices; it places the electronic device on standby mode if no user activity is detected during a pre set period of time. Virtualization of servers is the current trend in green computing. Rather than using multiple machines as servers, companies today use one high end server with virtualization and create multiple servers from it.

Even if no inbuilt features are provided, we can still use some green computing practices. We can turn off the monitor when it's not in use or use LCD or LED monitors instead of the traditional CRT monitors.

Summary

In this chapter we learnt about some emerging trends in computing technologies. We saw two computing domains called mobile computing and cloud computing. We learnt about different architectures of cloud computing namely SaaS, PaaS and IaaS. We also had a look at technologies used to identify a human or an object like RFID, GPS and Biometrics. In Biometrics we learnt about technologies like finger print recognition, Iris recognition and voice recognition. Then we had a look at emerging trends in data storage, wherein we discussed two data storage architecture; namely NAS and SAN along with different online storage options available. Robotics and computer controlled devices are a norm today; we had a look at some such technologies. Finally we had a look at some technology convergence in form of digital cameras and Internet enabled televisions. The last topic green computing addressed the issue of global warming and how it can be tackled using small steps.

EXERCISE

1. List the technologies that are most common to mobile computing.
2. Differentiate between personal cloud applications and business cloud applications.
3. Explain in brief the working of RFID.
4. What is the significance of Biometric techniques in object identification?
5. Identify at least three differences between finger print recognition and iris recognition.
6. Explain in brief the term SaaS.
7. Explain in brief the term PaaS.
8. Differentiate between NAS and SAN.
9. How is an online storage beneficial to an individual?
10. List some application area where computer controlled devices are used.
11. **Choose the most appropriate option from those given below :**
 - (1) The term Cloud in cloud computing represents which of the following?
 - (a) Wireless
 - (b) Hard drives
 - (c) People
 - (d) Internet
 - (2) Google Drive is an example of which of the following cloud types?
 - (a) Private Cloud
 - (b) Public Cloud
 - (c) Personal Cloud
 - (d) Business Cloud
 - (3) Which of the following isn't an advantage of cloud?
 - (a) No worries about running out of storage
 - (b) Easier to maintain a cloud network
 - (c) Immediate access to computing resources
 - (d) Paying only for what you use
 - (4) Which of the following is not a model of cloud computing?
 - (a) PaaS
 - (b) IaaS
 - (c) BaaS
 - (d) SaaS
 - (5) Which of the following is correct full form of SaaS?
 - (a) Software alternative accounting standard
 - (b) Short alert activation supplement
 - (c) Software as a service
 - (d) Service as a software

Laboratory Exercises

1. Use Google drive to upload some files and explore different operation that you can perform.
2. Open an account in ADrive and compare its feature with Google drive.
3. Create an account on Ubuntu One and explore its features.
4. Try to explore the free trial of salesforce.com
5. Explore your environment and write a report on how RFID is being used at different places.

Install VEDICS (Voice Enabled Desktop Interaction and Control System) and see how the computer works on voice commands.



Appendix

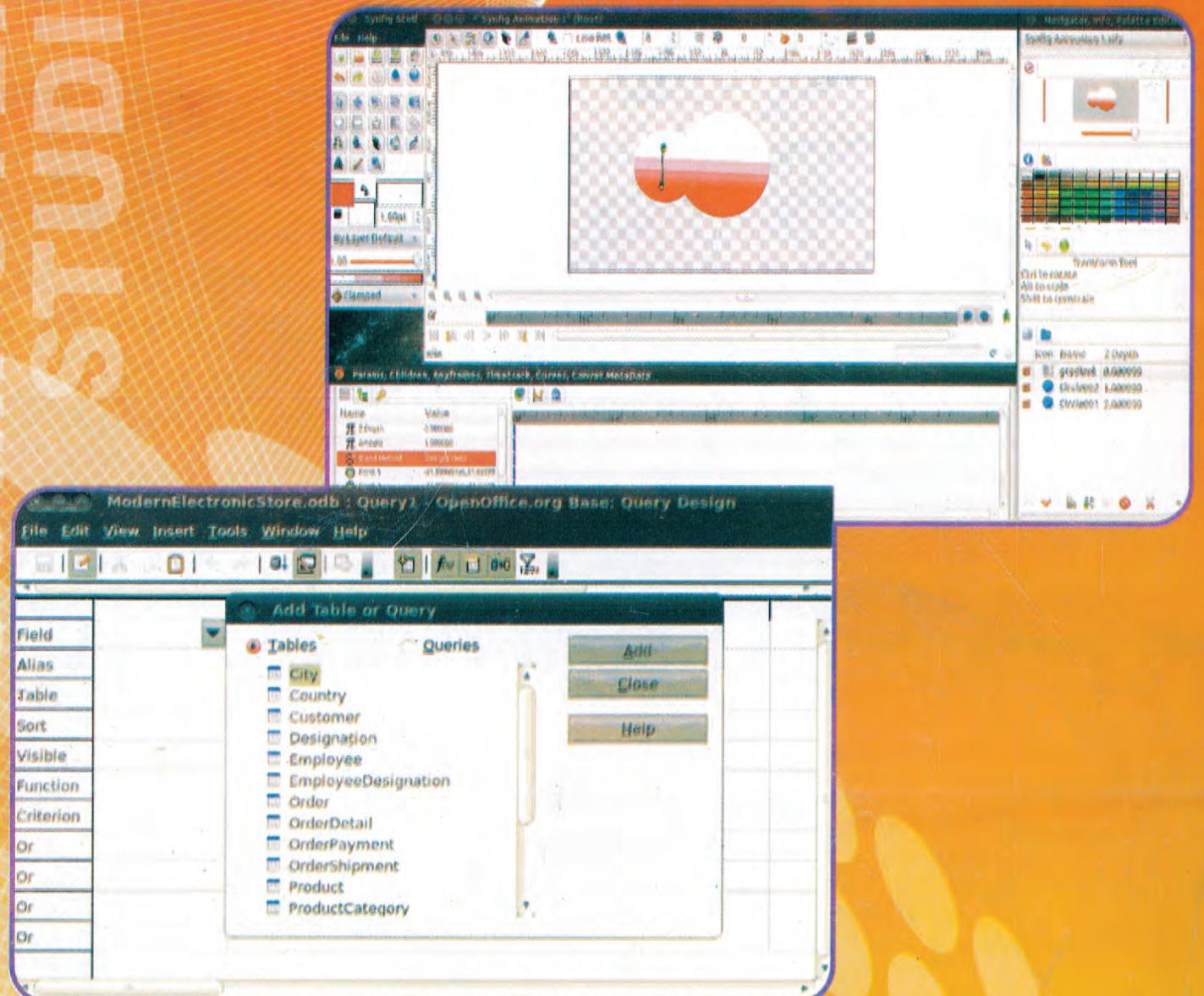


Basic Linux Commands covered in the textbook

Command	Usage
type	To check whether a command is internal or external
cal	To display calendar as per users requirement
date	To display current date and system time
bc	To initiate command line calculator
echo	To display message on terminal
passwd	To change the password of the current user
clear	To clear the content on the screen
man	To display the manual of a specified command
whatis	To display small description of a specified command
apropos	To display list of all commands with a specified string
pwd	To display the path of current working directory
mkdir	To create a directory
cd	To change a directory
rmdir	To remove a directory
cat	To create a file or to display contents of file
rm	To delete/remove a file
ls	To see the list of contents in a given directory
cp	To copy a file
mv	To rename a file or directory or to change location of file

more	To show one page of output on screen at a time
cmp	To compare two files
diff	To compare two files and display the contents where the difference lies.
wc	To count characters, words or number of lines
chmod	To change the permissions of file or directory
head	To display lines from the beginning of the file
tail	To display lines from the end of the file
cut	To cut the file vertically
paste	To paste contents of two files together
sort	To arrange the files in ascending or descending order
tr	To translate (convert) strings or patterns from one set of characters to another
grep	To search for strings or patterns within a file
find	To search for file/directory
sudo	To run command as a super user
adduser	To add new user to system
who	To display information of all users currently logged in the system
addgroup	To add user group to the system
deluser	To delete a user from the system
delgroup	To delete a user group form the system
sh	To execute specified shell script
bash	To execute specified shell script
who am i	To display information of the user currently logged in the system
ps	To see the processes associated with the current shell
kill	To kill (remove from main memory) the process with specified id.
tar	To compress (pack) a file/directory





ગુજરાત રાજ્ય શાસક પાઠ્યપુસ્તક મંડળ
ગાંધીનગર