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?

6. List at least three uses of shell script.

7. **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.

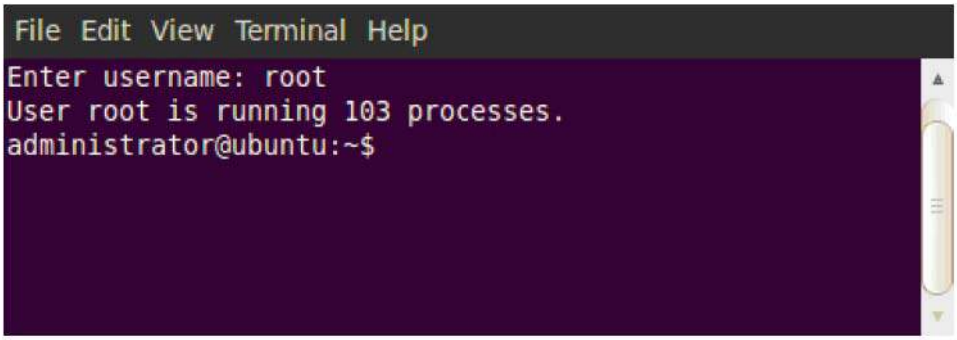


Figure 8.2 : Output of Script 10

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

$kill -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.

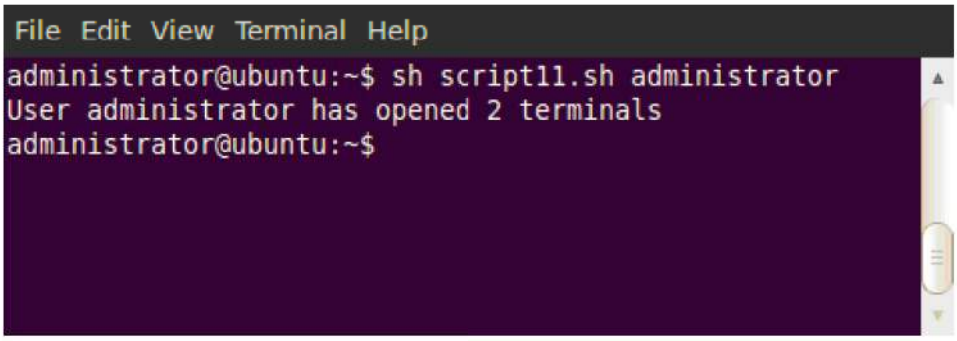


Figure 8.3 : Output of Script 11

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.

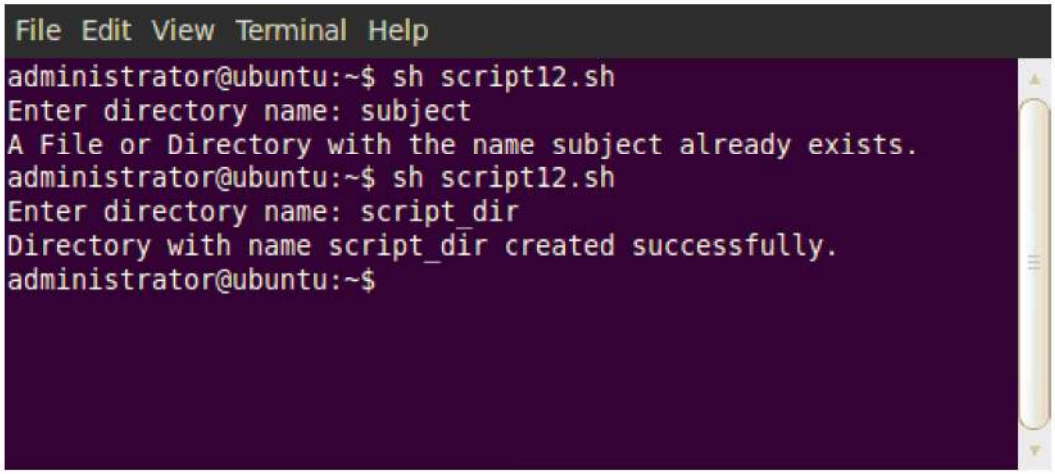


Figure 8.4 : Output of Script 12

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 absolute paths for directory (user can convert it relative path). We have also assumed that the file names at both the locations are same. Figure 8.5 shows the output of the script.

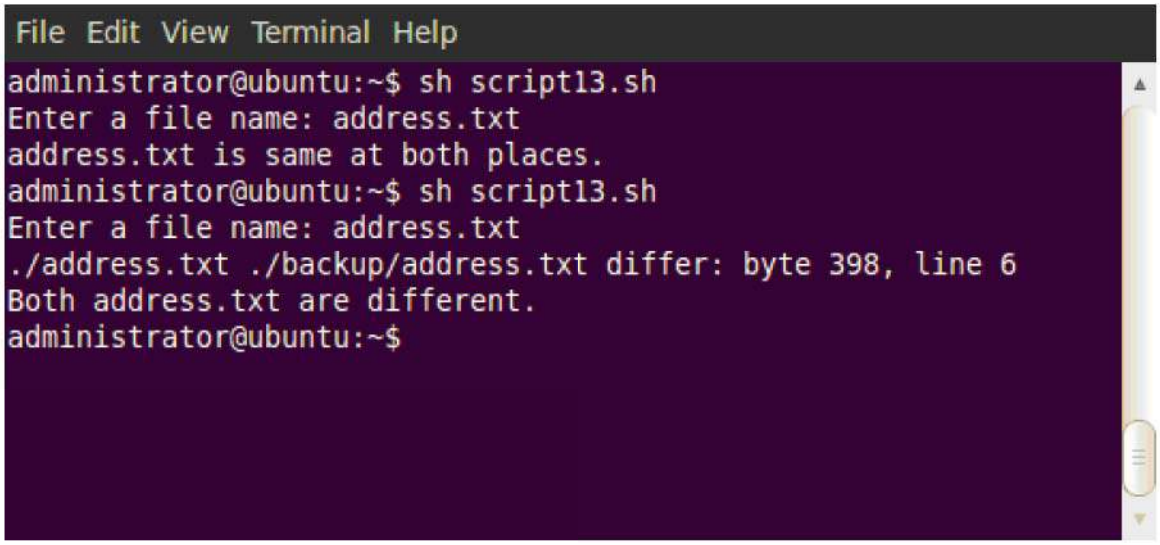


Figure 8.5 : Output of Script 13

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

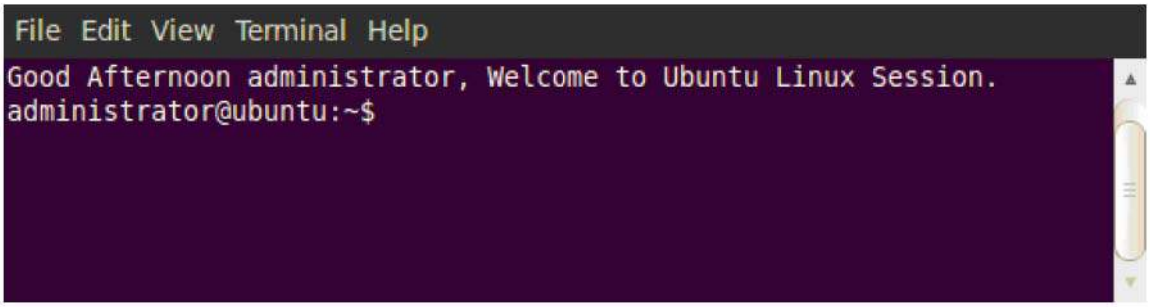


Figure 8.6 : Output of Script 14

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 $nfile
then
   echo "You have created more than $nfile files in the month of $1"
else
 echo "You have not created more than $nfile 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 *nfile* 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 nfile or not. If the value of cnt is greater than the value of nfile we print the message "You have created more than $nfile files in the month of $1", where $nfile and $1 are replaced with appropriate values. Otherwise we print message "You have not created more than $nfile 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

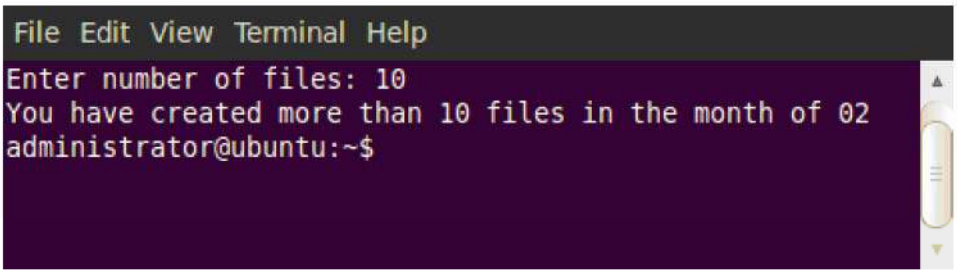


Figure 8.7 : Output of Script 15

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.

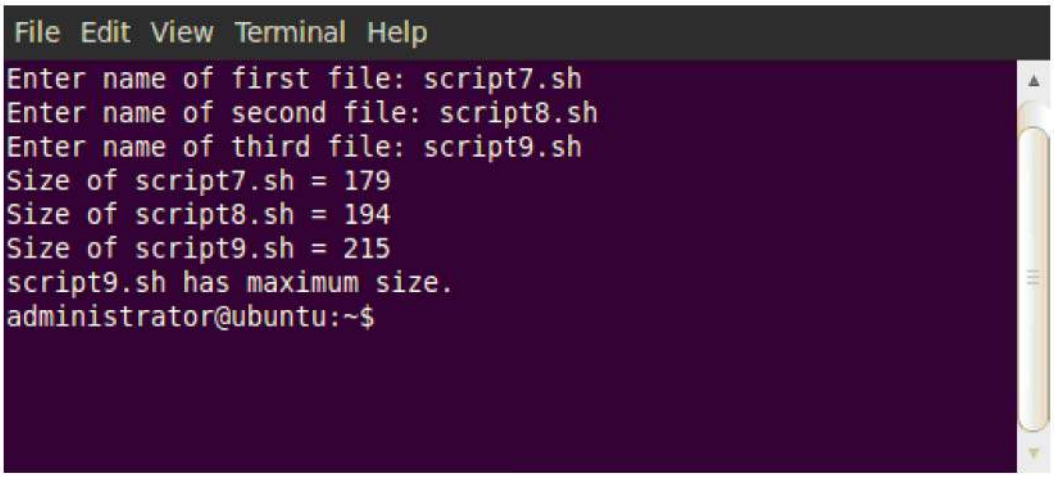


Figure 8.9 : Output of Script 17

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 -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.

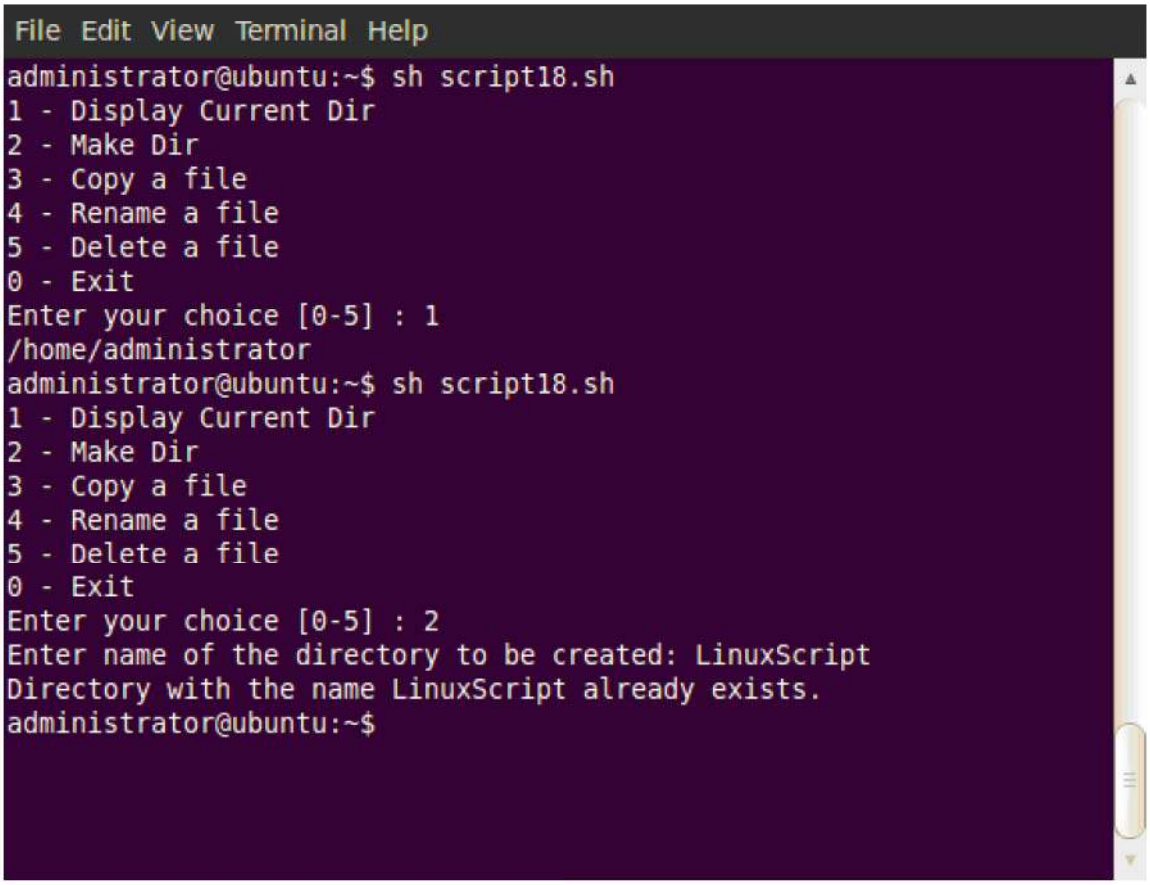


Figure 8.10 : Output of Script 18

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.

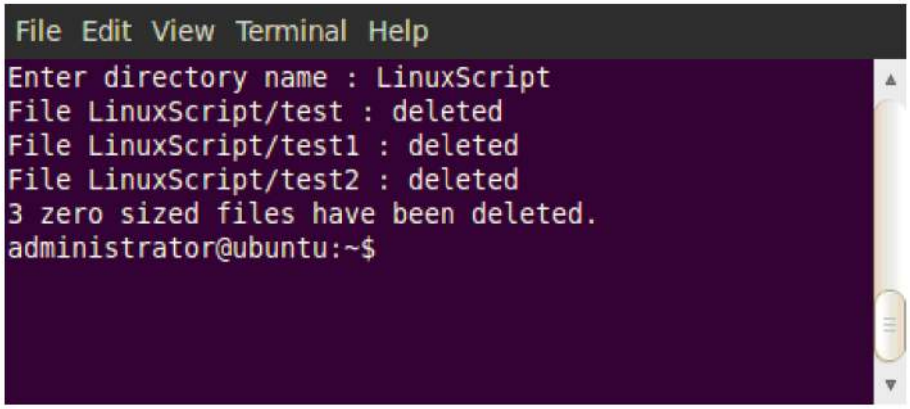


Figure 8.11 : Output of Script 19

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 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.

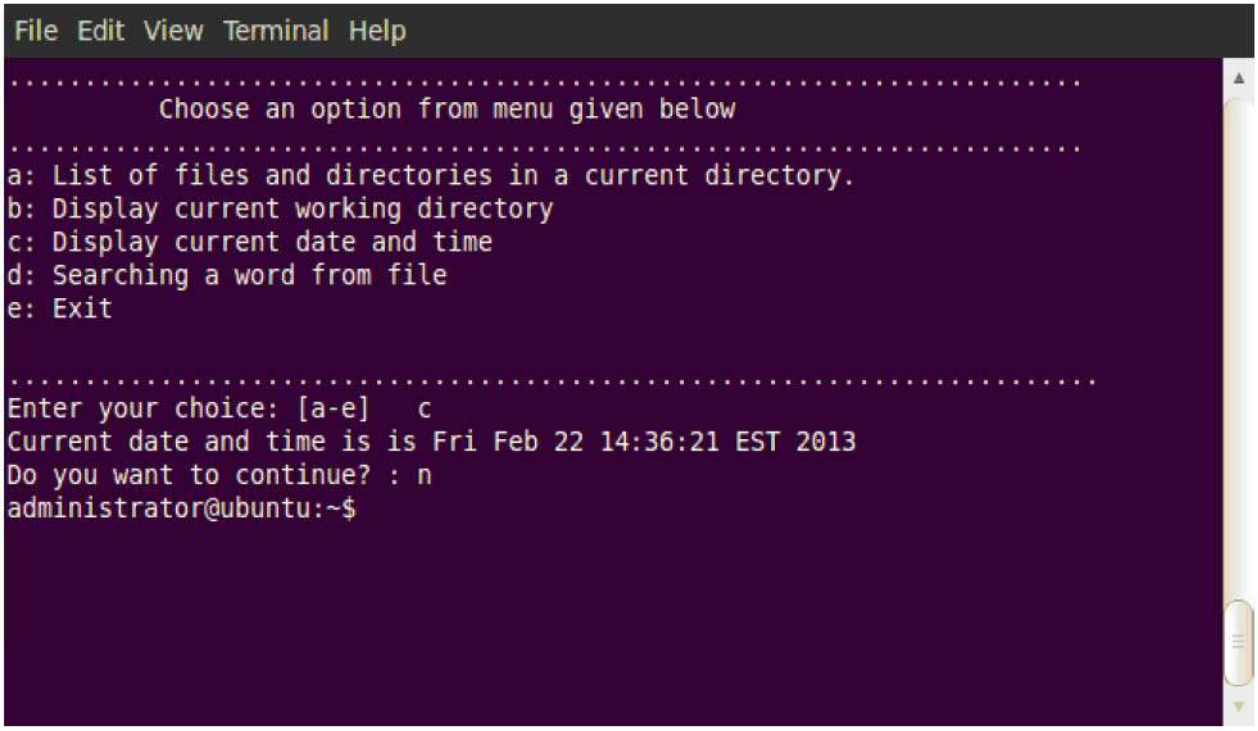


Figure 8.12 : Output of Script 22

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.

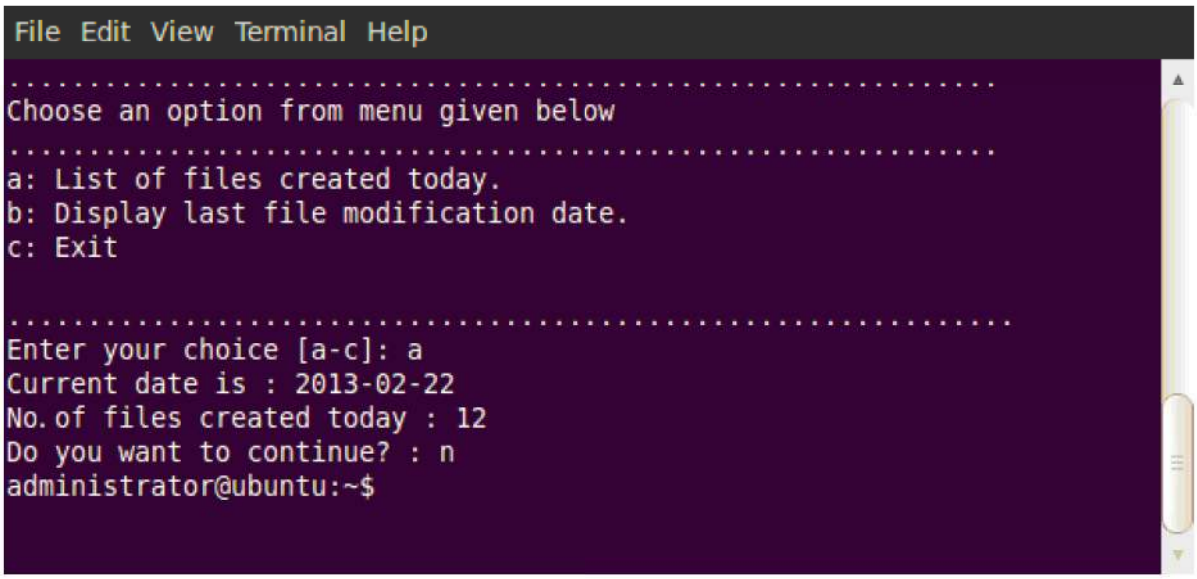


Figure 8.13 : Output of Script 23

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

1. Explain conditional execution in shell script with proper example.
2. Explain case statement of shell script with the option of pattern list.
3. Explain the use of while loop.
4. Explain the use of until loop.
5. **Choose the most appropriate option from those given below :**

(1) Which of the following command is used to set the file permission as executable?

(a) grep (b) chmod

(c) ls (d) x

(2) Which of the following symbol is used to break the flow of control in the case statement?

(a) ** (b) ;;

(c) ++ (d) >>

(3) Which of the following keyword specifies the end of the case statement?

(a) end-case (b) end case

(c) esac (d) stop-case

(4) Which loop iterates till the condition evaluates to true?

(a) while (b) until

(c) for (d) case

(5) Which of the following allows us to specify a list of values in its statement?

(a) while (b) until

(c) for (d) if

(6) In case structure, which of the following character denotes default case?

(a) * (b) +

(c) d (d) All of the above

(7) Which of the following statement is true for testing whether the file is read only or not?

(a) test –read filename (b) check –read filename

(c) test –r filename (d) check –r filename

(8) Which of the following indicate the end of if condition in shell script?

(a) end-if (b) fi

(c) } (closing curly bracket) (d) It does not have any end statement

(9) Which of the following operator is used for less than comparison in Linux?

(a) < (b) lessthan

(c) lt (d) -lt

(10) Which of the following can be used in place of square brackets used in if condition?

(a) Curly braces (b) test command

(c) check command (d) All of the above

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.

9 Introduction to Database Management System

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

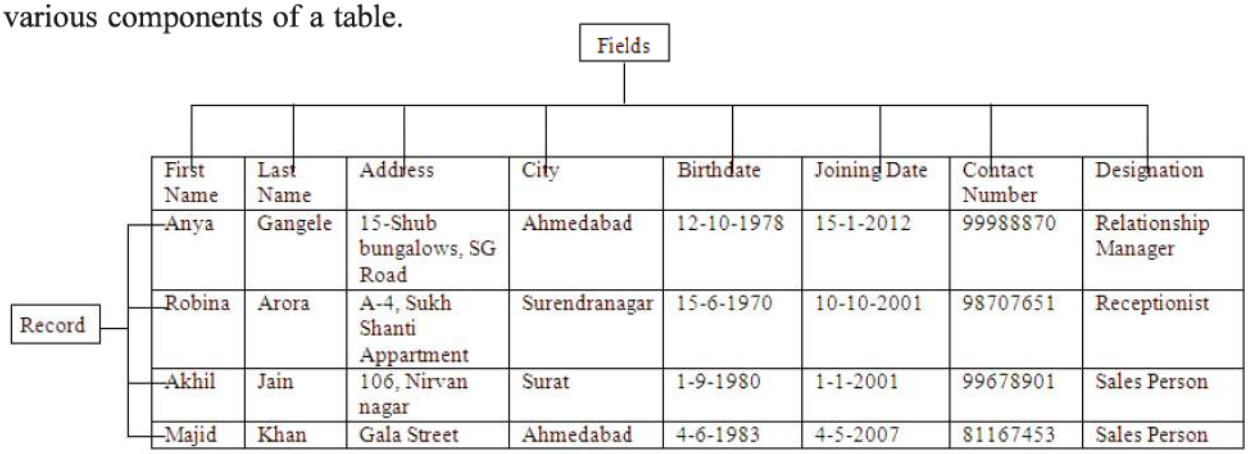
(DBMS). DBMS can be defined as system designed for efficient storage, maintenance and retrieval of data. In short the DBMS software is what makes database practical and more usable. In today's world a database itself is useless if there is no DBMS associated with it for accessing its data. Oracle, Ingress, 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.



First Name	Last Name	Address	City	Birthdate	Joining Date	Contact Number	Designation
Anya	Gangele	15-Shub bungalows, SG Road	Ahmedabad	12-10-1978	15-1-2012	99988870	Relationship Manager
Robina	Arora	A-4, Sukh Shanti Appartment	Surendranagar	15-6-1970	10-10-2001	98707651	Receptionist
Akhil	Jain	106, Nirvan nagar	Surat	1-9-1980	1-1-2001	99678901	Sales Person
Majid	Khan	Gala Street	Ahmedabad	4-6-1983	4-5-2007	81167453	Sales Person

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 confirm 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	CameraPixel 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/sdscs3000.tiff	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	9998886662	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	$5 \times 10^{(-324)}$ to $1.79 \times 10^{(308)}$

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	
Field Name	**Data type**
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

Employee	
Field Name	**Data type**
FirstName	Text [VARCHAR_IGNORECASE]
LastName	Text [VARCHAR_IGNORECASE]
AddressLine1	Text [VARCHAR_IGNORECASE]
AddressLine2	Text [VARCHAR_IGNORECASE]
City	Text [VARCHAR_IGNORECASE]
Pincode	Text fix [CHAR]
EmailId	Text [VARCHAR]
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	
Field Name	**Data type**
ProductCategory	Text [VARCHAR_IGNORECASE]
ModelName	Text [VARCHAR_IGNORECASE]
SupplierName	Text [VARCHAR_IGNORECASE]
ReorderLevel	SmallInt
SellingPrice	Integer
QuantityonHand	SmallInt
ProductImage	Text [VARCHAR]
CameraPixelSize	Integer
Processor	Text [VARCHAR_IGNORECASE]
OSSupport	Text [VARCHAR_IGNORECASE]
TouchScreen	Boolean
Wi-fiEnabled	Boolean
MemorySlotInGB	SmallInt
FMSupport	Boolean
DisplaySizeInInches	SmallInt
HDDCapacity	SmallInt
WeightInGrams	SmallInt

Customer	
Field Name	**Data type**
CustomerFname	Text [VARCHAR_IGNORECASE]
CustomerLname	Text [VARCHAR_IGNORECASE]
AddressLine1	Text [VARCHAR_IGNORECASE]
AddressLine2	Text [VARCHAR_IGNORECASE]
City	Text [VARCHAR_IGNORECASE]
Pincode	Text [VARCHAR]
EmailId	Text [VARCHAR_IGNORECASE]
MobileNumber	Text [VARCHAR]
Card Holder	Boolean

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.

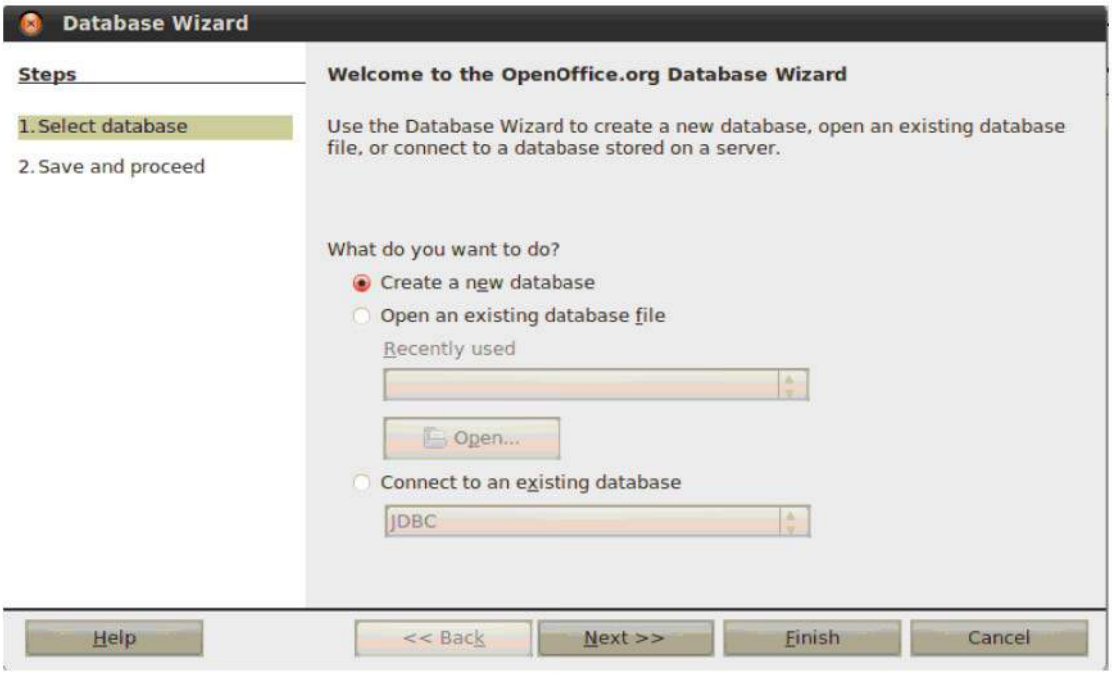


Figure 9.8 : Opening Screen of Base

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.

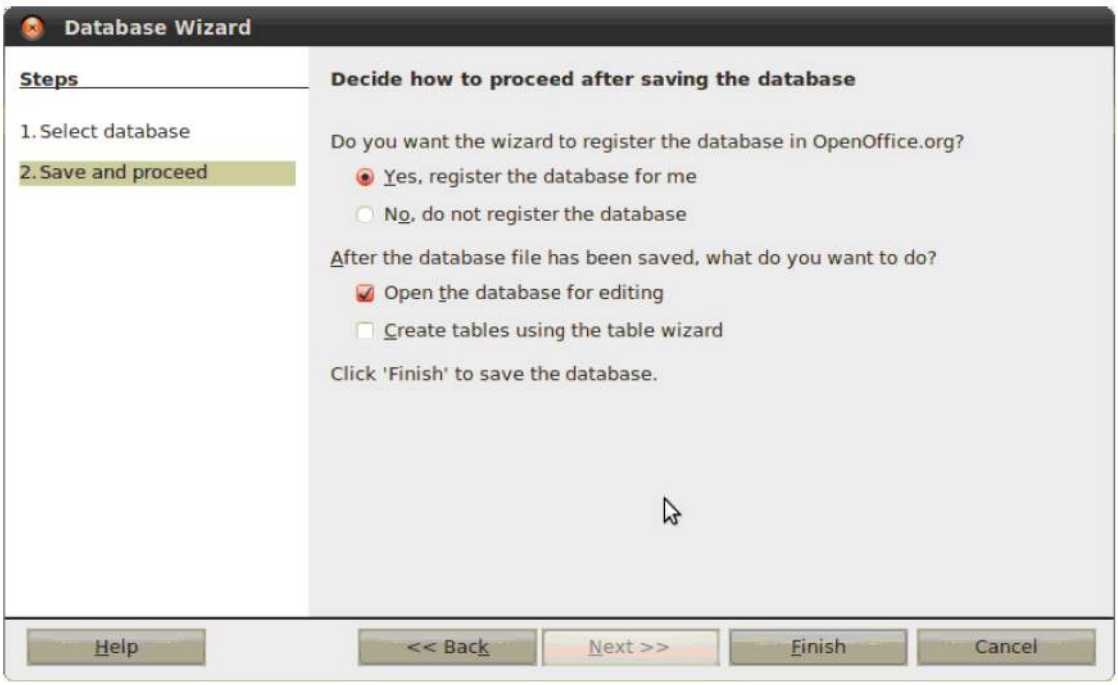


Figure 9.9 : Opening the Database

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.

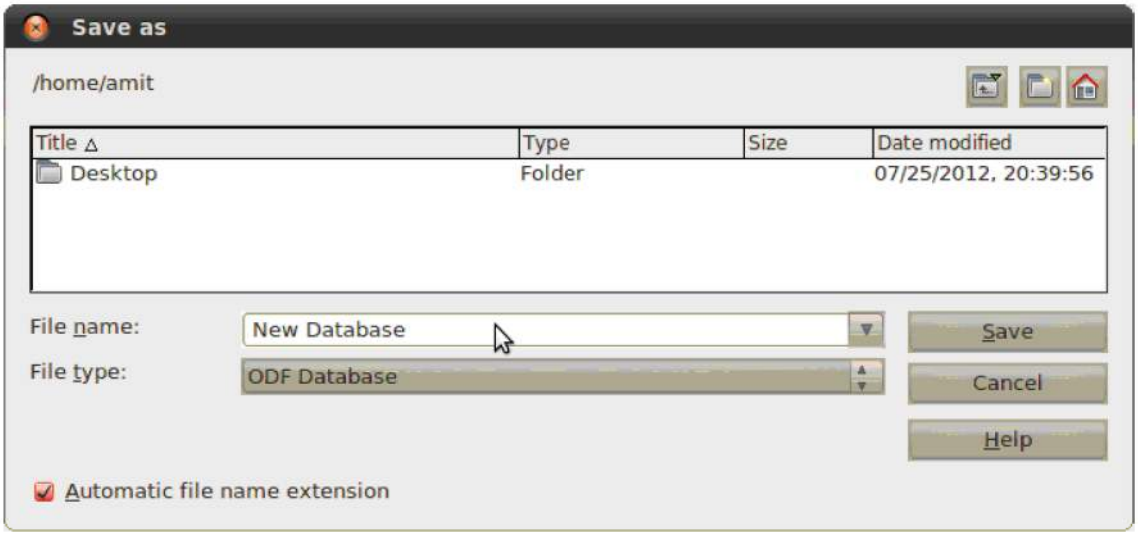


Figure 9.10 : Naming Database File

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.

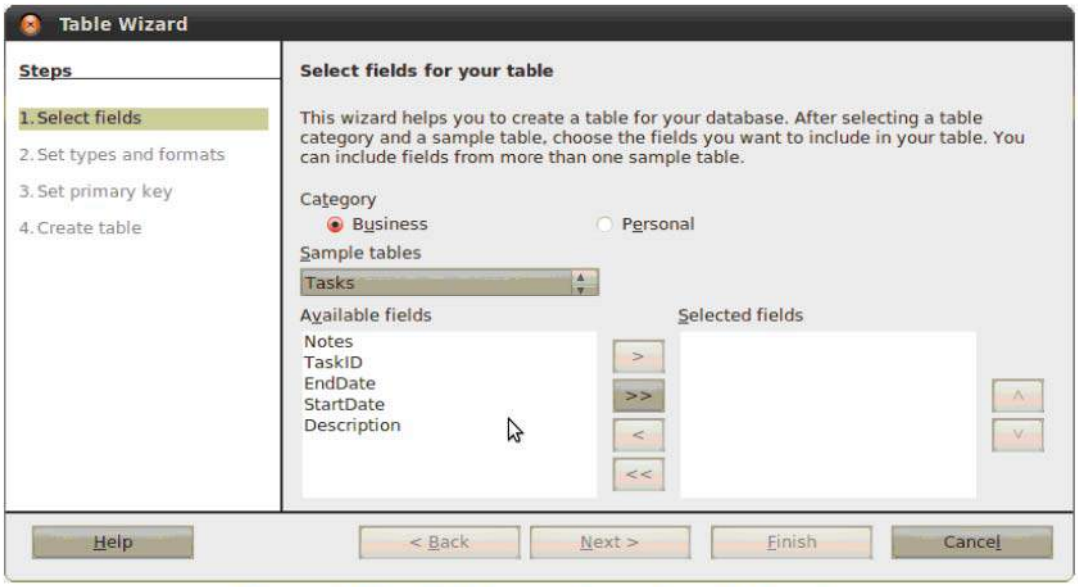


Figure 9.11 : Table creation using Wizard

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.

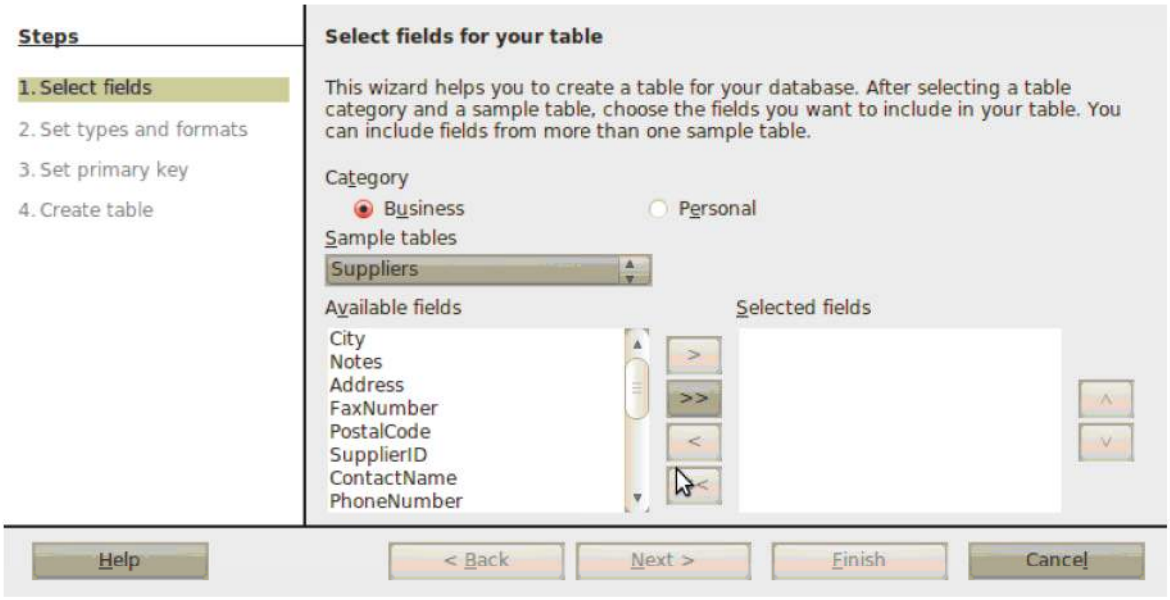


Figure 9.12 : Creating Suppliers table

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.

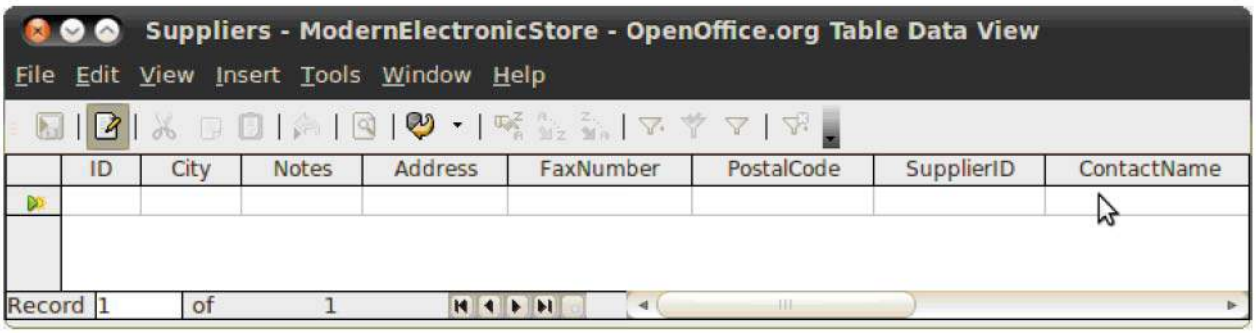


Figure 9.13 : Supplier Table in Data View

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.

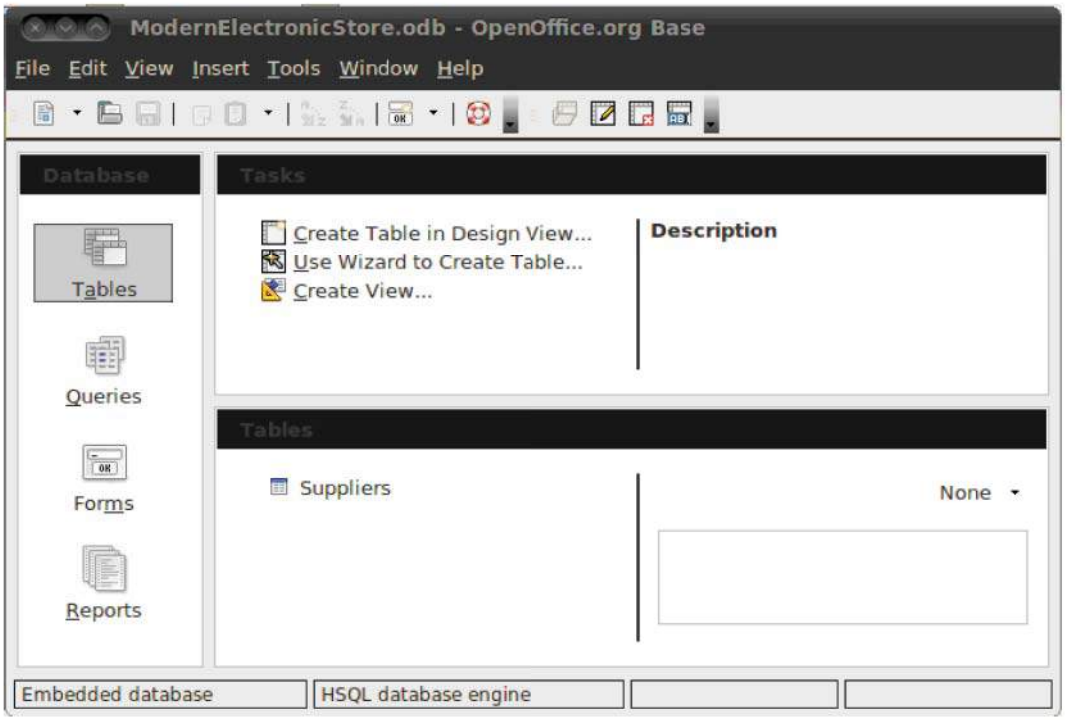


Figure 9.14 : Database Window

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).

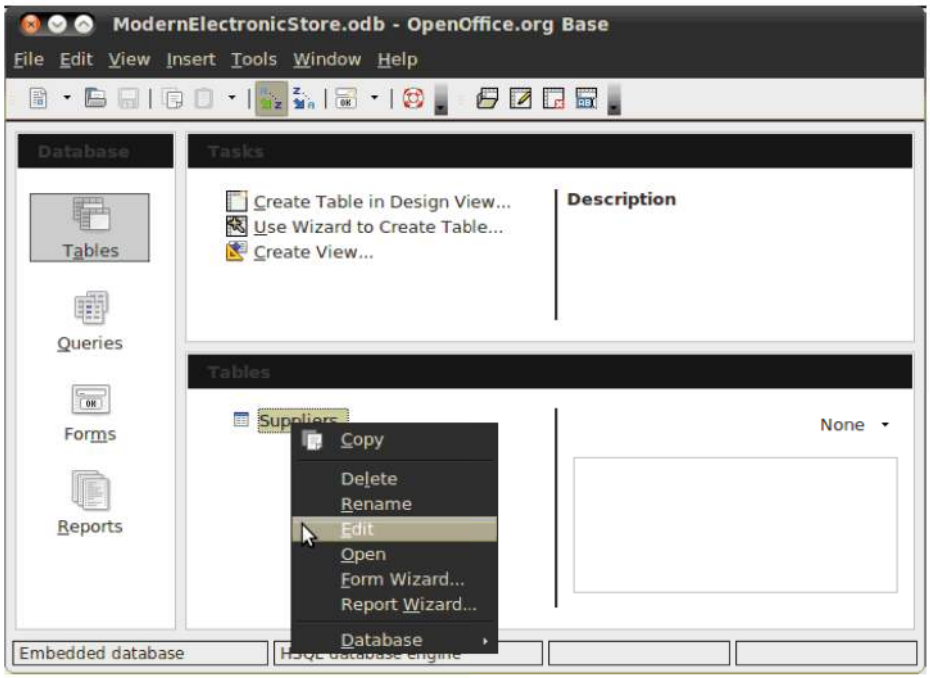


Figure 9.15 : Editing Suppliers Table

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.

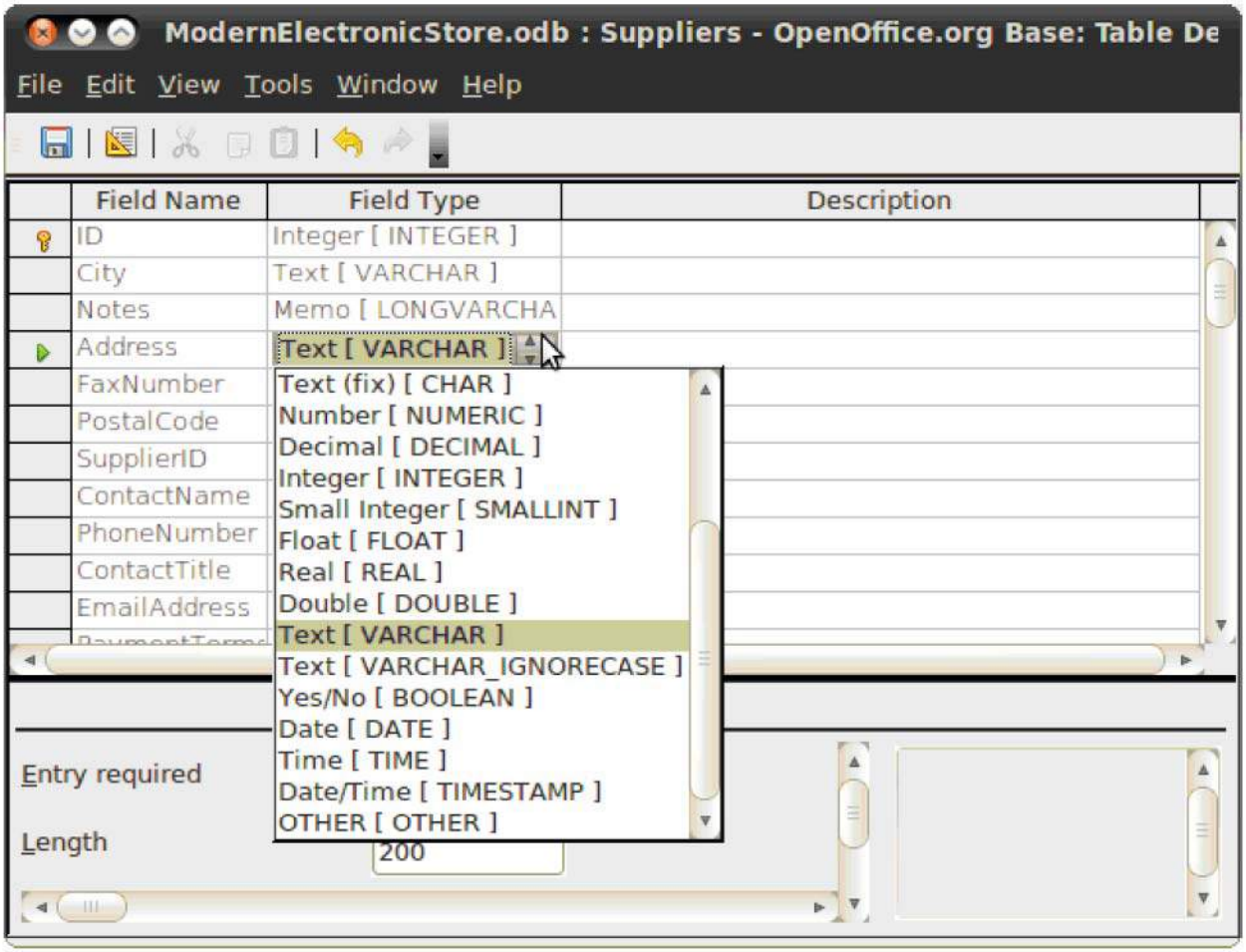


Figure 9.16 : Choosing Data type for a field

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.

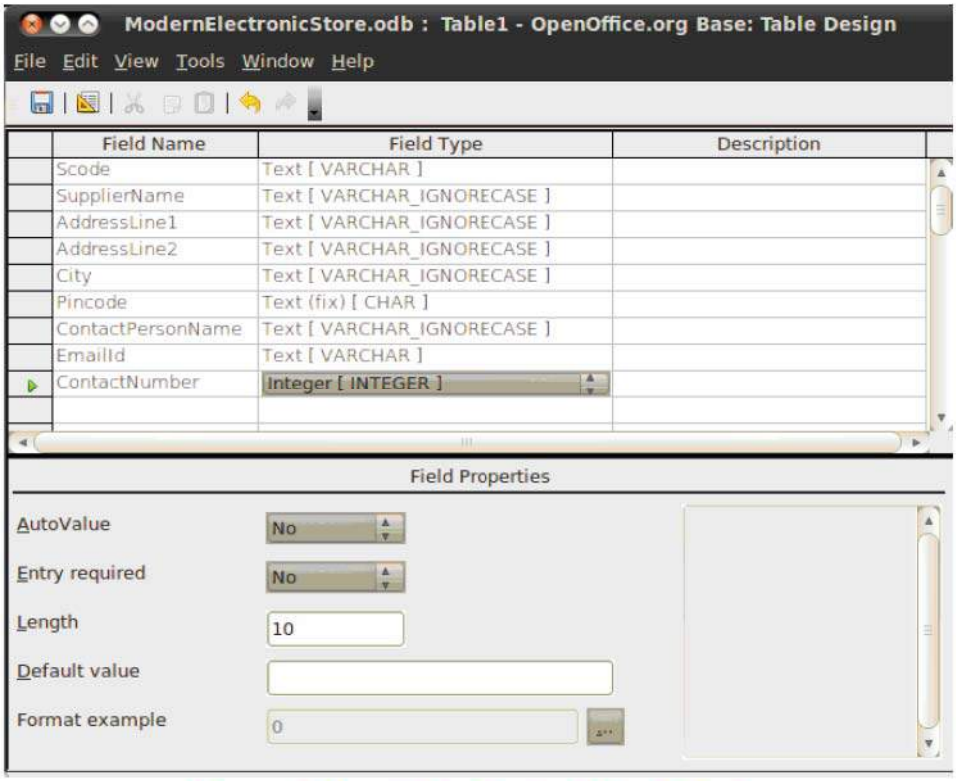


Figure 9.18 : Table Design View Window

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.

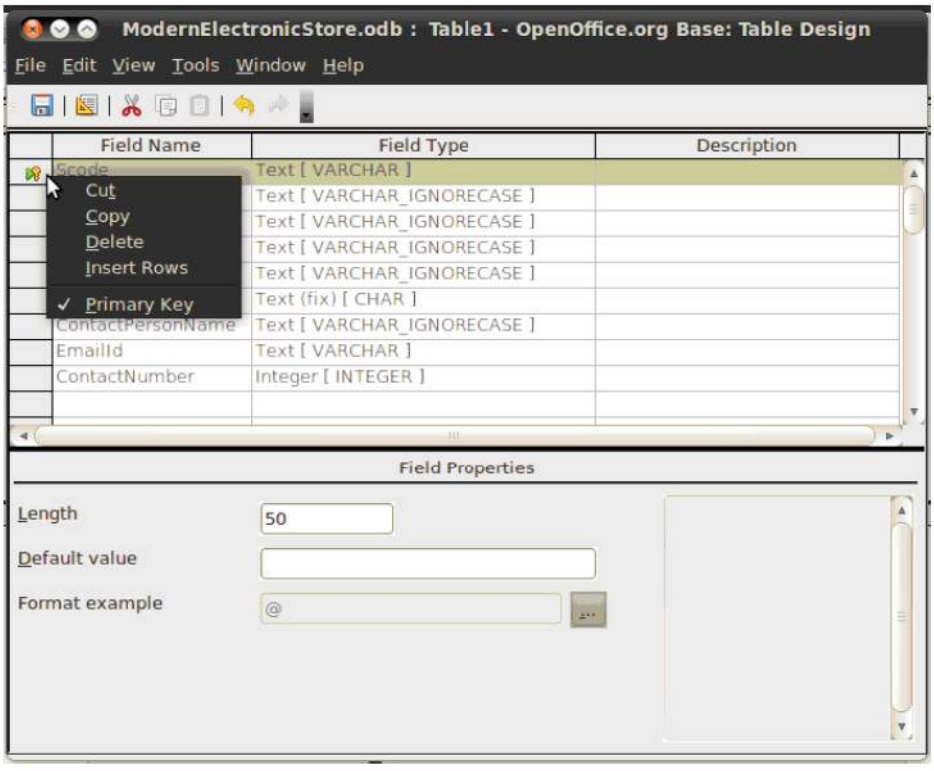


Figure 9.19 : Designating Primary key of a Table

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

(10) Which of the following is not an object maintained by Base?

(a) Tables (b) Queries

(c) Charts (d) Forms

(11) Which data type cannot be used to store numbers?

(a) Decimal (b) Integer

(c) Text (d) Date

(12) Which data type is used to store image in Base database?

(a) Binary (b) Photo

(c) Long (d) Huge

(13) Which of the following is false statement?

(a) Primary key cannot contain null values.

(b) Primary key cannot contain duplicate values.

(c) Primary key can be combination of more than one field.

(d) Primary key is always numeric field.

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**
Grno	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

Field Name	Data Type	Description
Standard	Number	Which Standard
Scode	Text	Which Subject

Attendance: To store daily attendance of the Students

Field Name	Data Type	Description
Grno	Number	General Register of a Student
Date	Date/Time	On which Date
Pr_ab	Yes/No	Present or Absent

Fees: To store details of Fee amount paid by students

Field Name	Data Type	Description
Grno	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

Cultural: To keep record of Interest of students in Cultural Activities

Field Name	Data Type	Description
Grno	Number	General Register Number
Interest	Text	Interested in which activity
Achievement	Memo	Any achievements?

Subj_Teacher: To store details about which teacher is teaching the subject in each standard

Field Name	Data Type	Description
Year	Number	Current Year
Standard	Number	In which Standard
Scode	Text	Subject Code
Tcode	Text	Teacher Code

Result: To store result details of each student

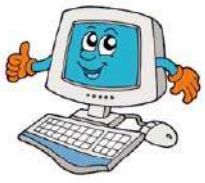
Field Name	Data Type	Description
Grno	Number	General Register No.
Scode	Text	Of which Subject
Month	Text	In which Month
Year	Number	Which Year
Term	Number	Name of the Term
Marks	Number	Marks scored

Sports: To keep record of Interest of students in Sports

Field Name	Data Type	Description
Grno	Number	General Register No.
Interest	Text	Interested in which game
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.

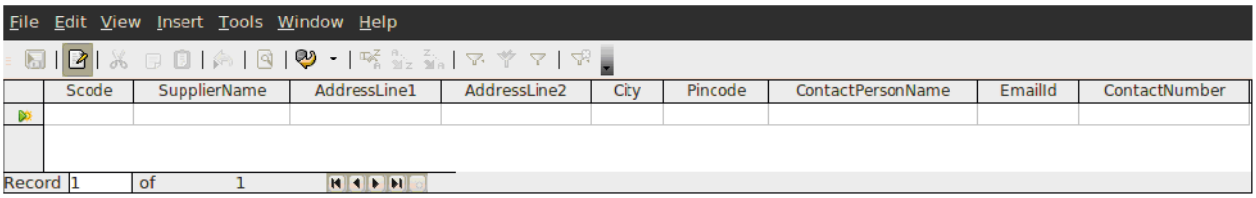


Figure 10.1 : Initial data entry screen

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 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.

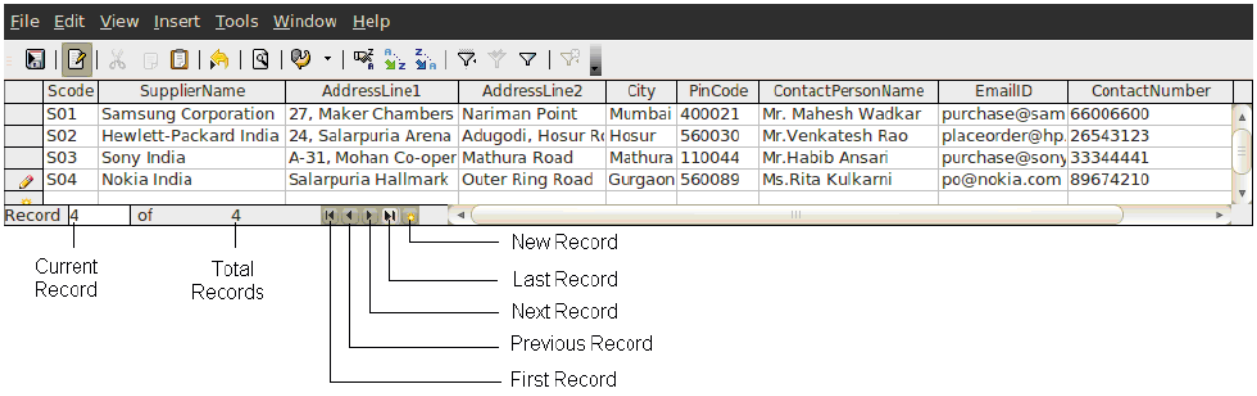


Figure 10.2 : Datasheet view after inserting records

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 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.

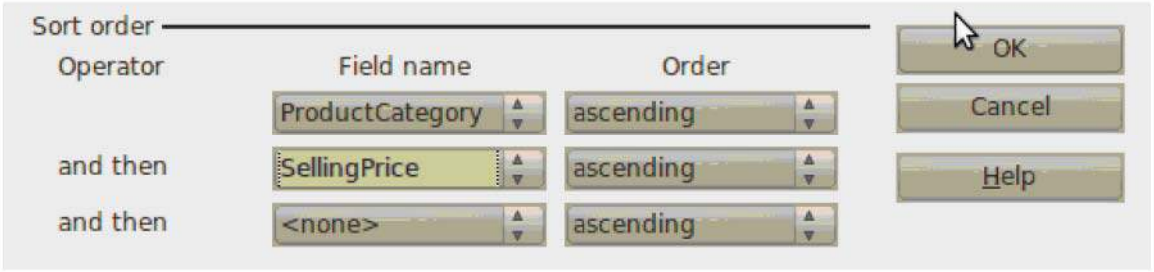


Figure 10.4 : Choice of Sort Order

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.

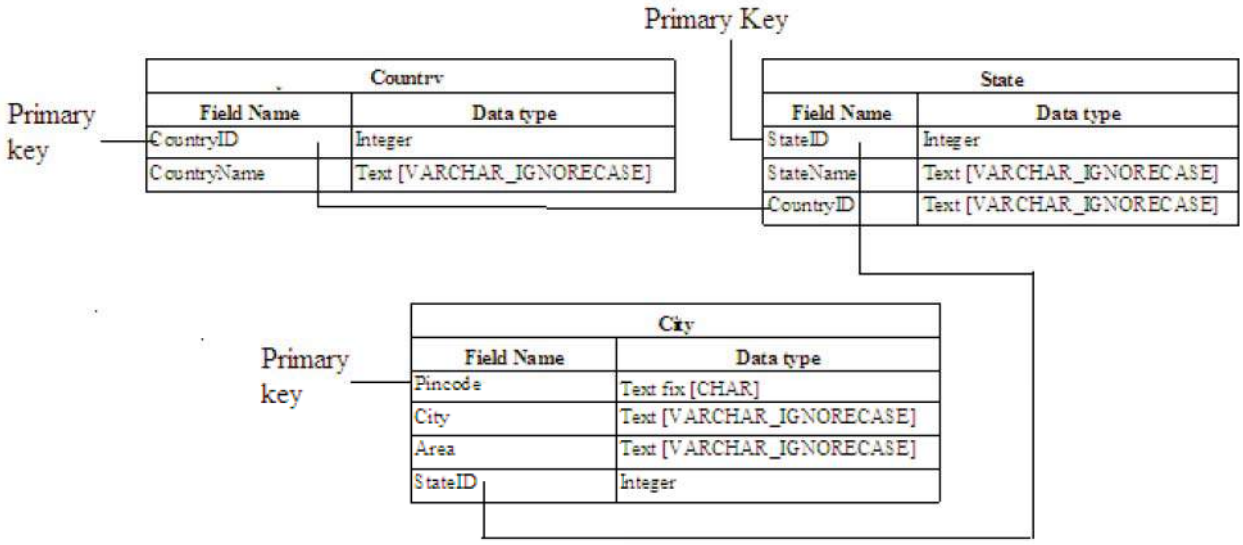


Figure 10.5 : Country, State and City Tables

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 that 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	
To store details of orders	
Field Name	**Data type**
OrderID	Integer
OrderDate	Date
Ccode	Text fix [CHAR]
Ecode	Text fix [CHAR]

OrderDetail	
To store details of products purchased by customer	
Field Name	**Data type**
OrderID	Text fix [CHAR]
Pcode	Text fix [CHAR]
Quantity	Integer
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.

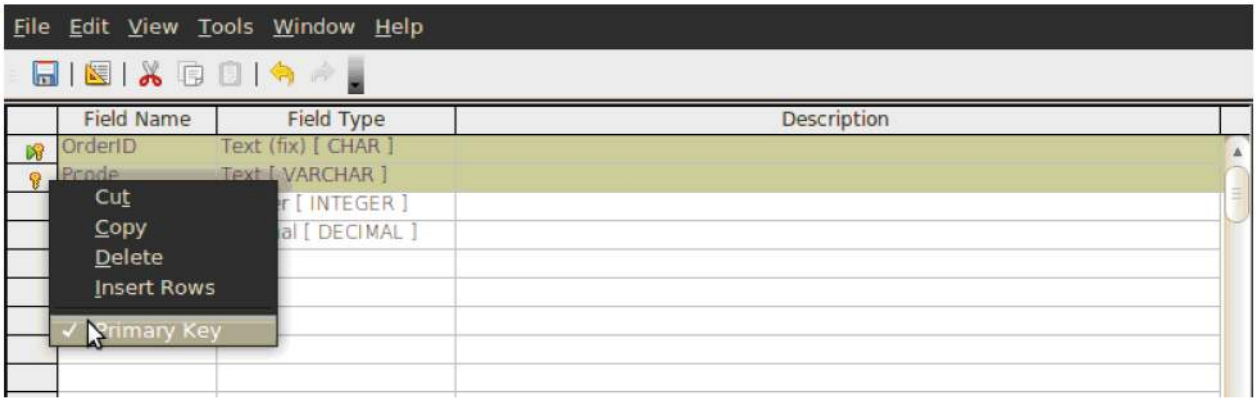


Figure 10.7 : Defining Composite Primary Key

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.

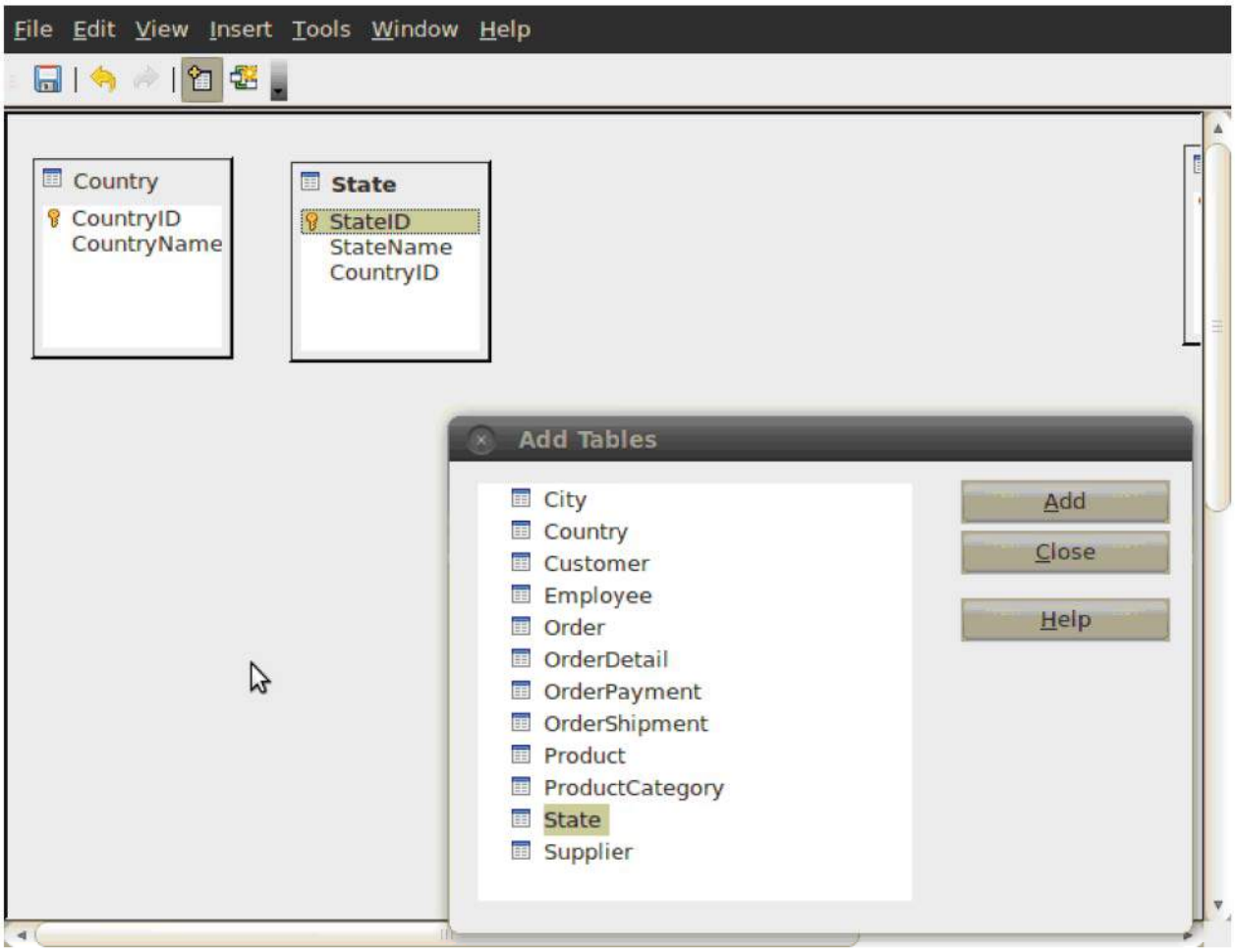


Figure 10.9 : Relationships dialog box

- 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 ***1*** and ***n*** will be displayed as seen in figure 10.10. Notice that label text *1* 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.

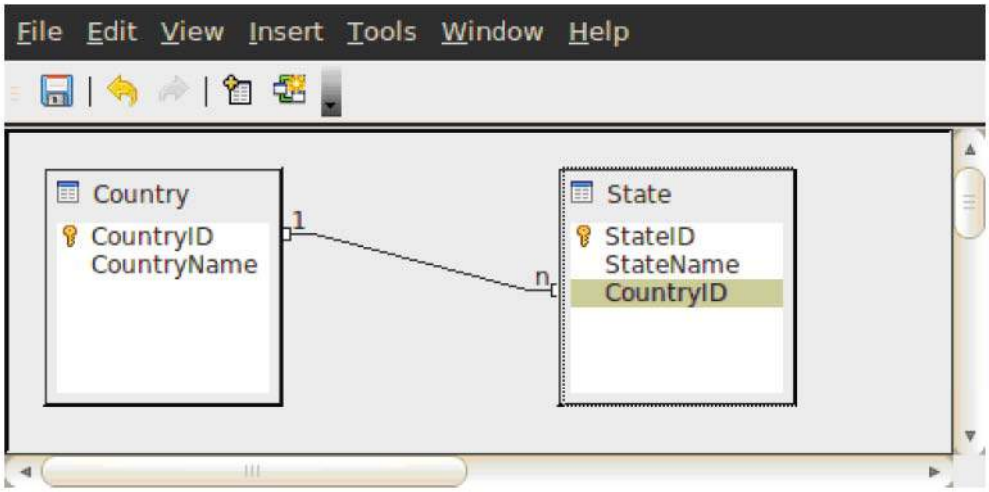


Figure 10.10 : One-to-Many relationship between Country and State

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).

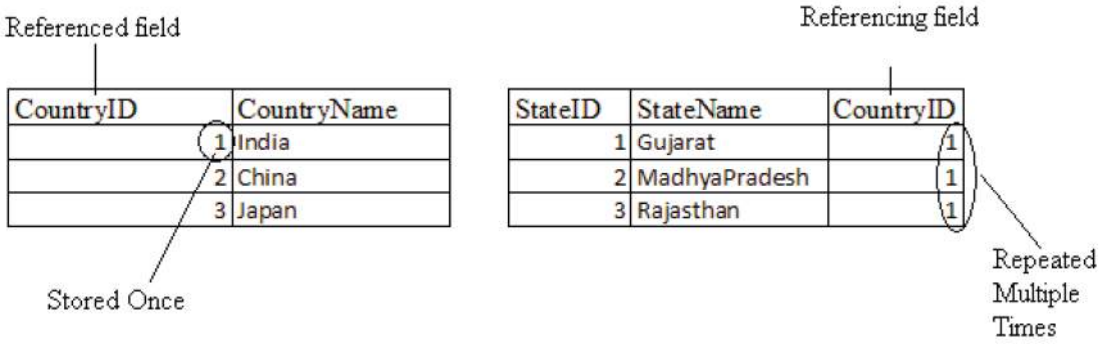


Figure 10.11: Records indicating One-to-Many Relationship

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.

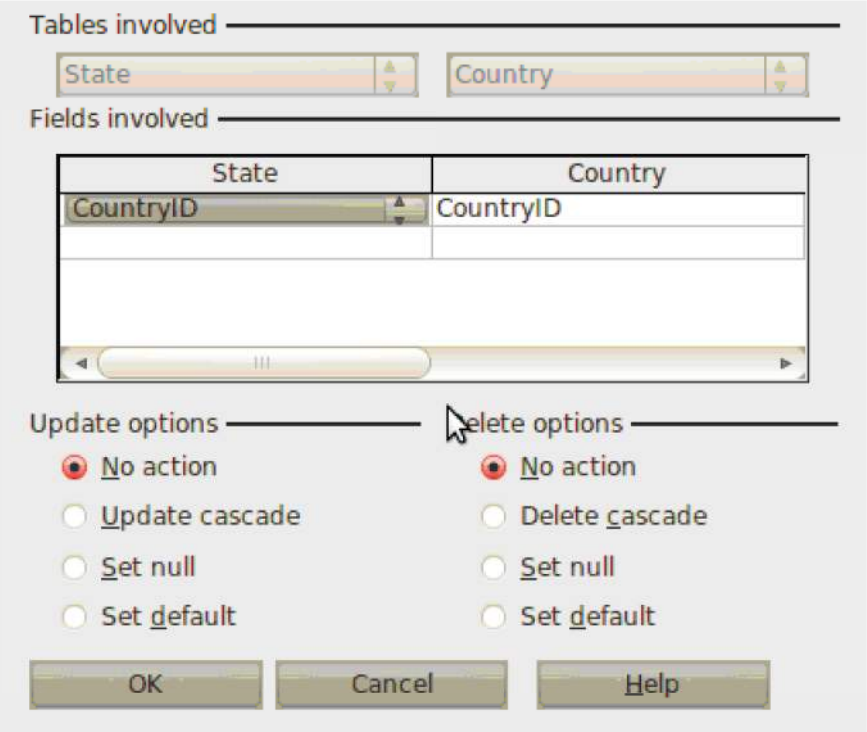


Figure 10.13 : Enforcing referential integrity

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.

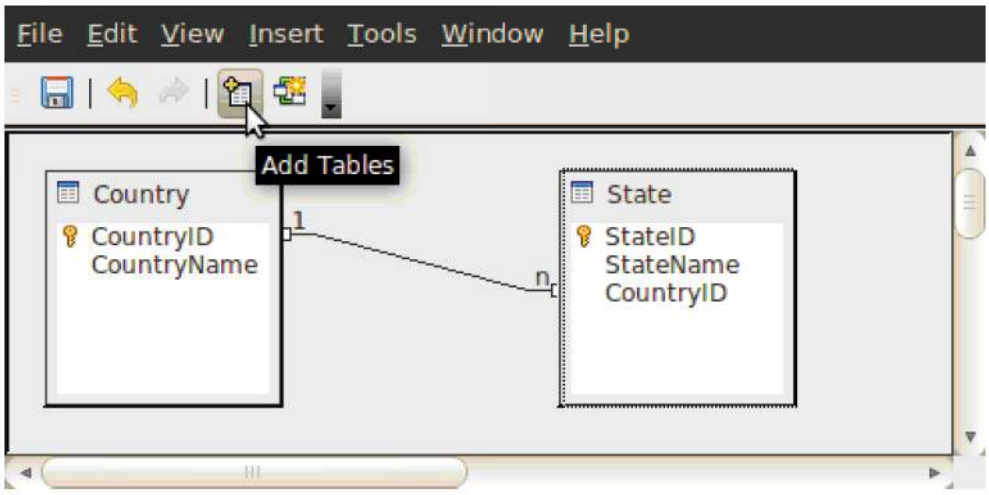


Figure 10.15 : Adding Tables in Relationship window

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.

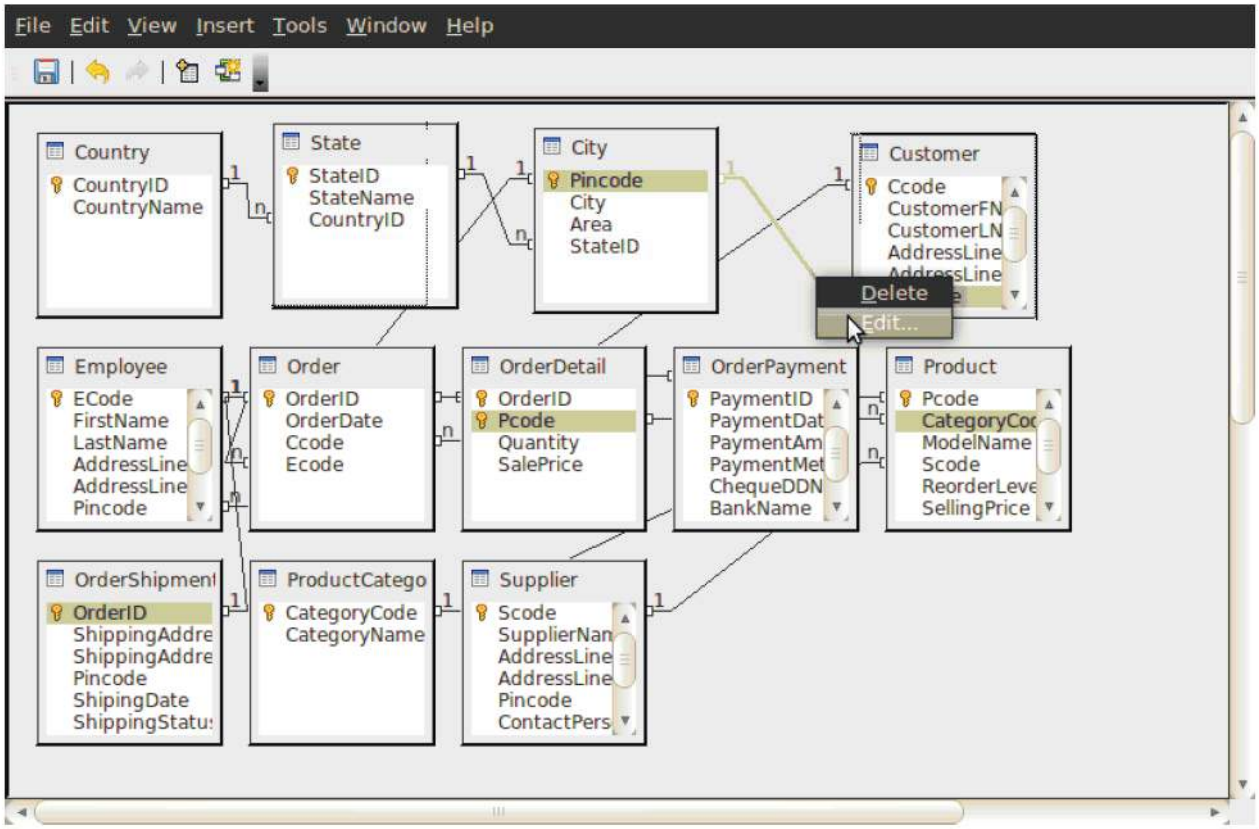


Figure 10.16 : Relationship of all tables in Database

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 ModenElectronicDatabase 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.

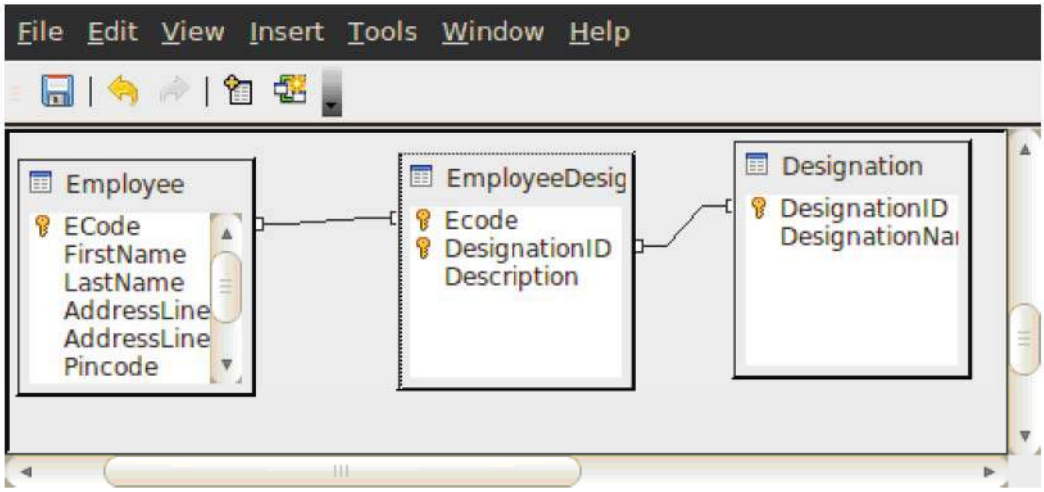


Figure 10.17 : Many-to-Many relationship between Employee and Designation

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.

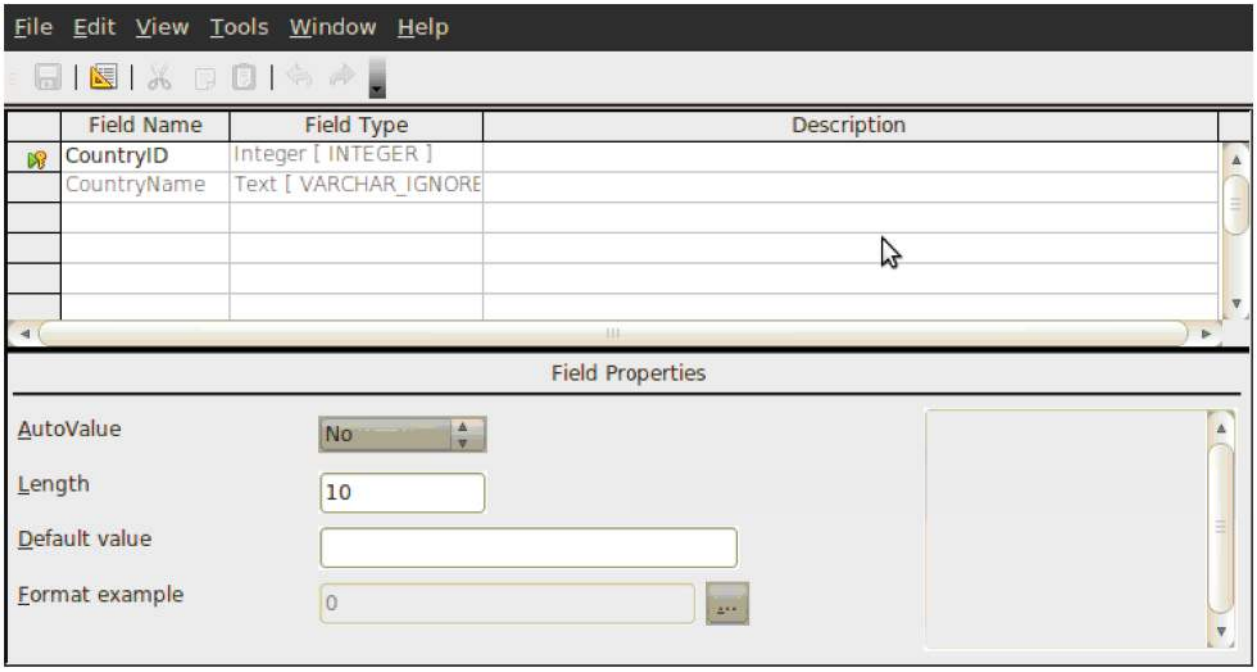


Figure 10.18 : Field Property dialog box

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.*

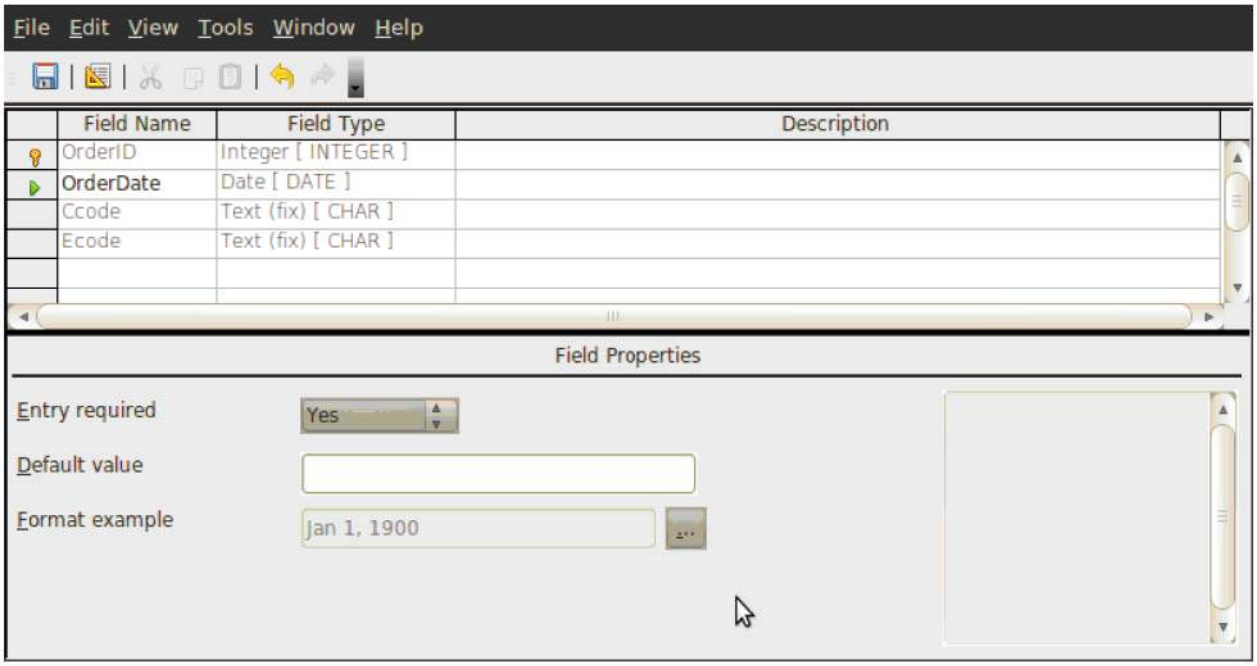


Figure 10.19 : Entry required Property set to Yes

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.

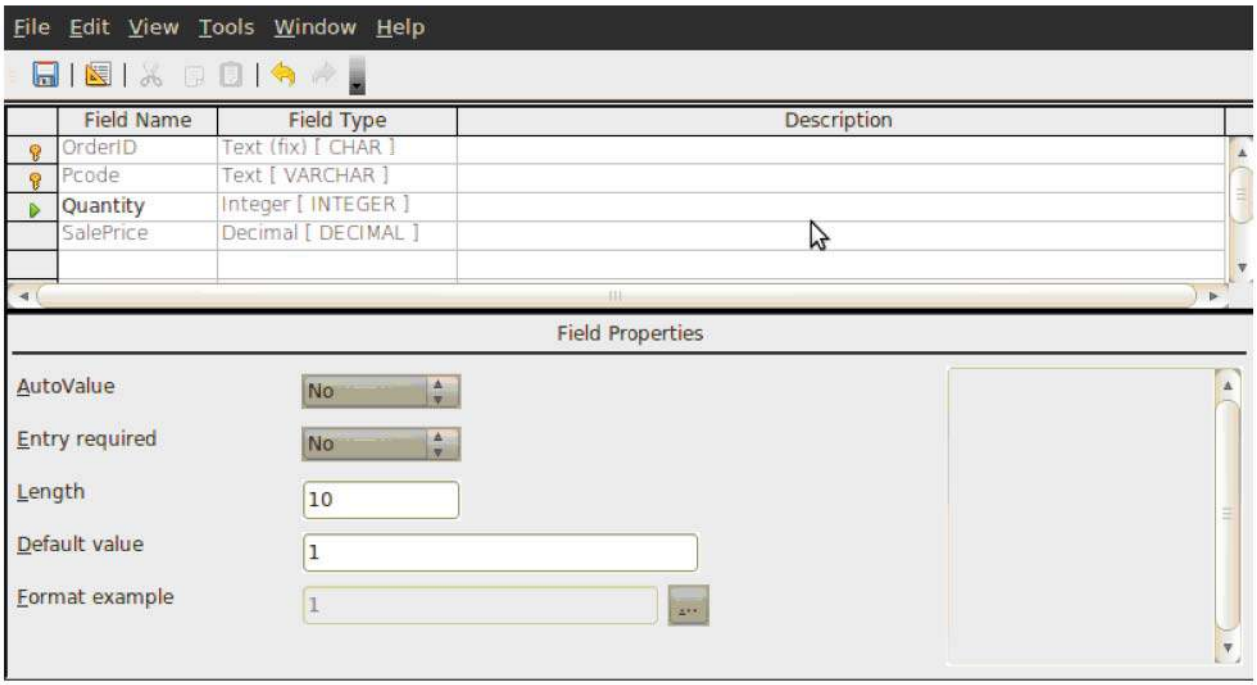


Figure 10.20 : Specifying Default value for a field

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.

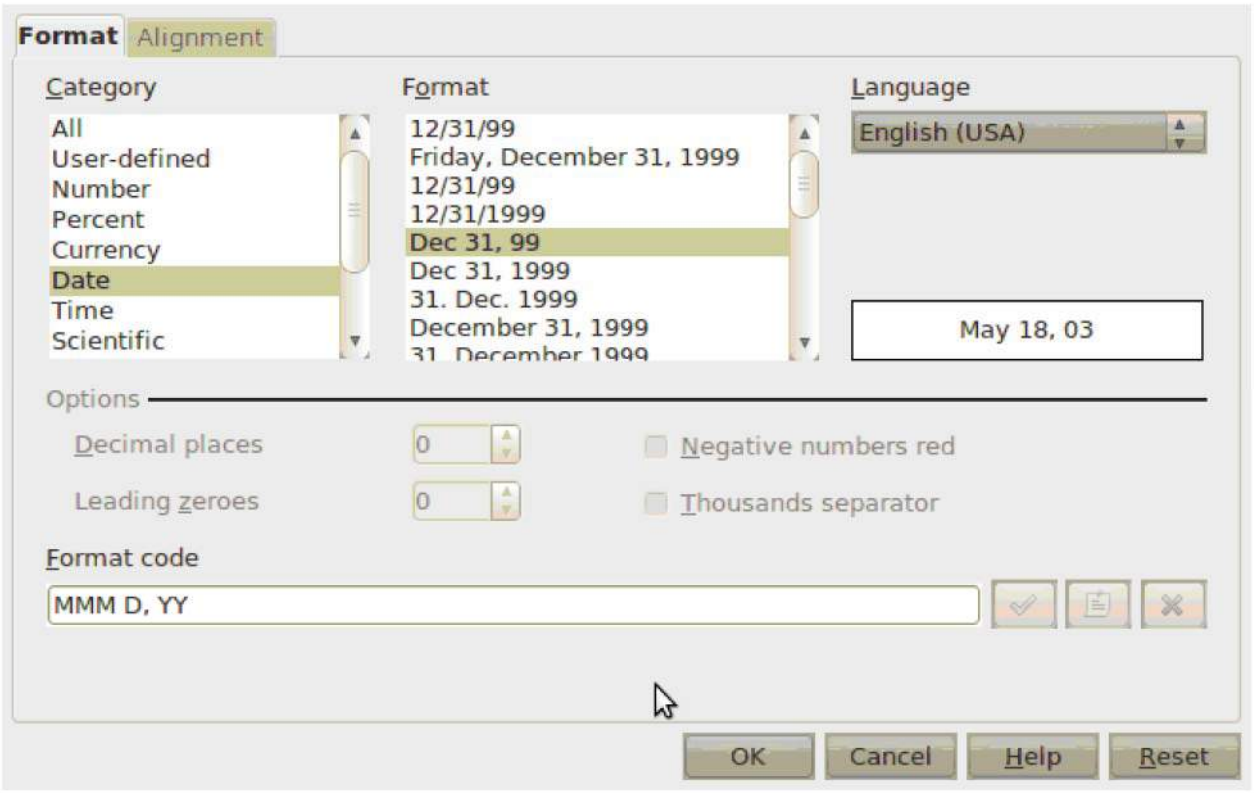


Figure 10.21 : Predefined formats

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. Explain Autovalue property.

11. What is use of Entry required property?

12. What is the use of Format property?

13. **Choose the most appropriate option from those given below :**

(1) Which of the following operations cannot be performed on a record in database?

(a) Insert (b) Delete

(c) Hide (d) Update

(2) Which of the following represents maximum levels of sorting provided by Base Wizard?

(a) 3 (b) 4

(c) 5 (d) 6

(3) Which of the following refers to data redundancy?

(a) Deletion of data (b) Repetition of data

(c) Decomposition of tables (d) Relationships of tables

(4) Given a relation, TeacherSubject (Tcode, Scode, Standard), which of the following fields can be an appropriate foreign key field?

(a) Tcode (b) Scode

(c) Tcode,Standard (d) Tcode,Scode

(5) Which of the following statement is incorrect?

(a) Every table must have a primary key.

b. Primary key can be combination of fields.

(c) Data type of Primary key and Foreign key should be same.

d. Names of Primary key and Foreign key should be same.

(6) Which of the following type of relationship exists between Student and Teacher in a school?

(a) One to One (b) One to Many

(c) Many to Many (d) No relationship exists

(7) Which of the following type of relationship exists between Student and Class Teacher in a school?

(a) One to One (b) One to Many

(c) Many to Many (d) No relationship exists

(8) Which of the following options are possible to implement referential integrity?

(a) No Action (b) Set Null

(c) Delete table (d) Update Cascade

(9) AutoValue property can be set for field with which of the following datatype?

(a) Text (b) Image

(c) Integer (d) Boolean

(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.

(c) 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
Grno	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

Field Name	Data Type	Description
Standard	Number	Which Standard
Scode	Text	Which Subject

Attendance: To store daily attendance of the Students

Field Name	Data Type	Description
Grno	Number	General Register of a Student
Date	Date/Time	On which Date
Pr_ab	Yes/No	Present or Absent

Fees: To store details of Fee amount paid by students

Field Name	Data Type	Description
Grno	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

Subj_Teacher: To store details about which teacher is teaching the subject in each standard

Field Name	Data Type	Description
Year	Number	Current Year
Standard	Number	In which Standard
Scode	Text	Subject Code
Tcode	Text	Teacher Code

Result: To store result details of each student

Field Name	Data Type	Description
Grno	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

Cultural: To keep record of Interest of students in Cultural Activities

Field Name	Data Type	Description
Grno	Number	General Register Number
Interest	Text	Interested in which activity
Achievement	Memo	Any achievements?

Sports: To keep record of Interest of students in Sports

Field Name	Data Type	Description
Grno	Number	General Register No.
Interest	Text	Interested in which game
Achievement	Memo	Any achievements?

Figure 10.23 : Transaction tables of Student Management System

11 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.

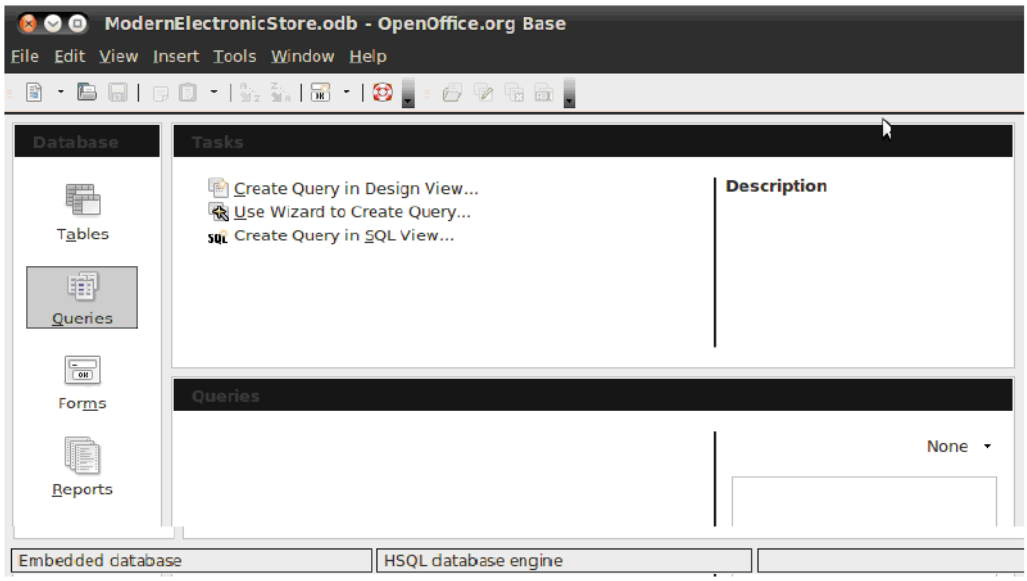


Figure 11.1 : Queries Window

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.

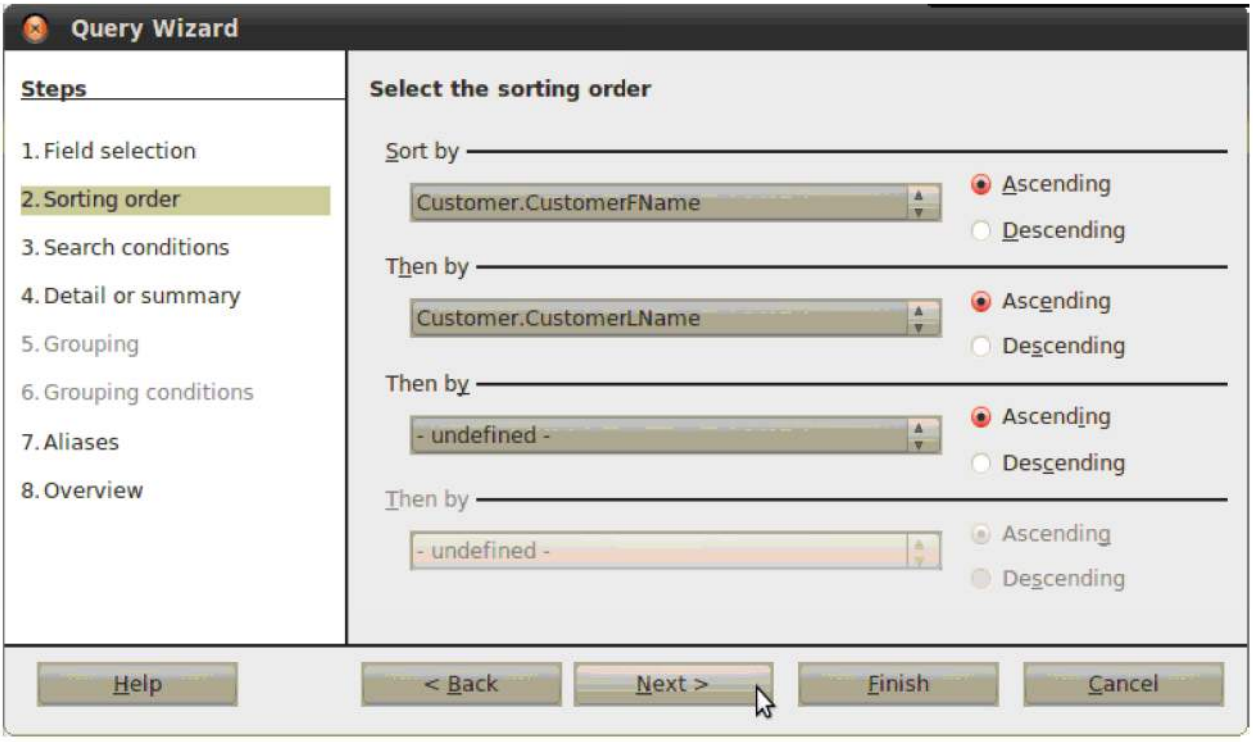


Figure 11.3 : Applying sorting on field

- 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.

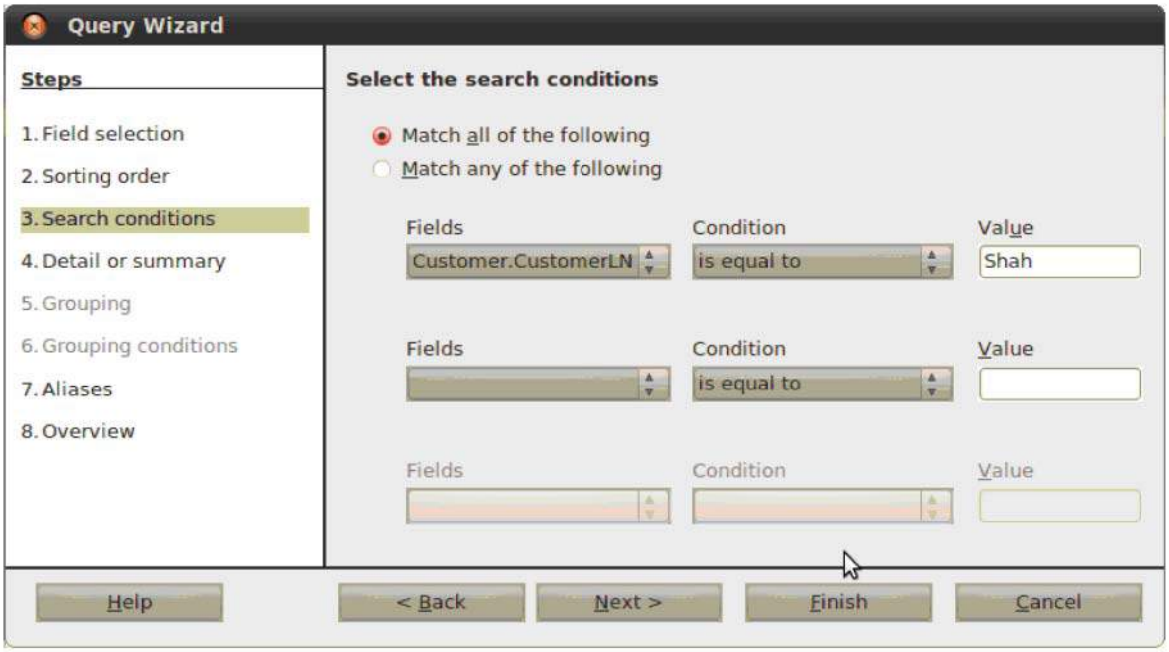


Figure 11.4 : Applying search condition on field

- 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.

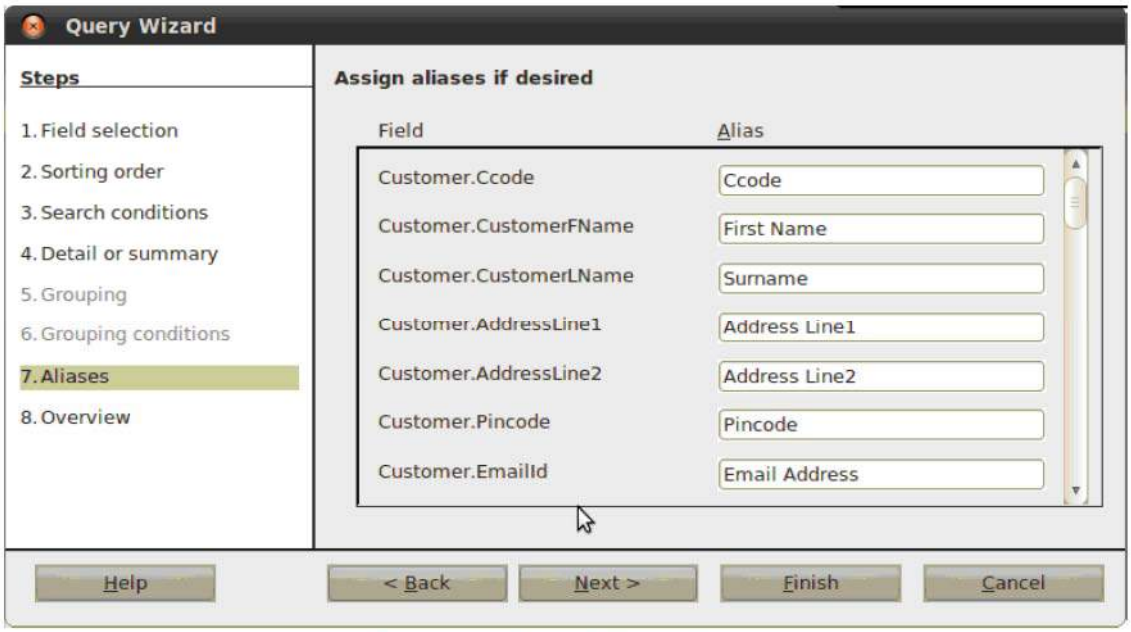


Figure 11.5 : Adding Aliases

- 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.

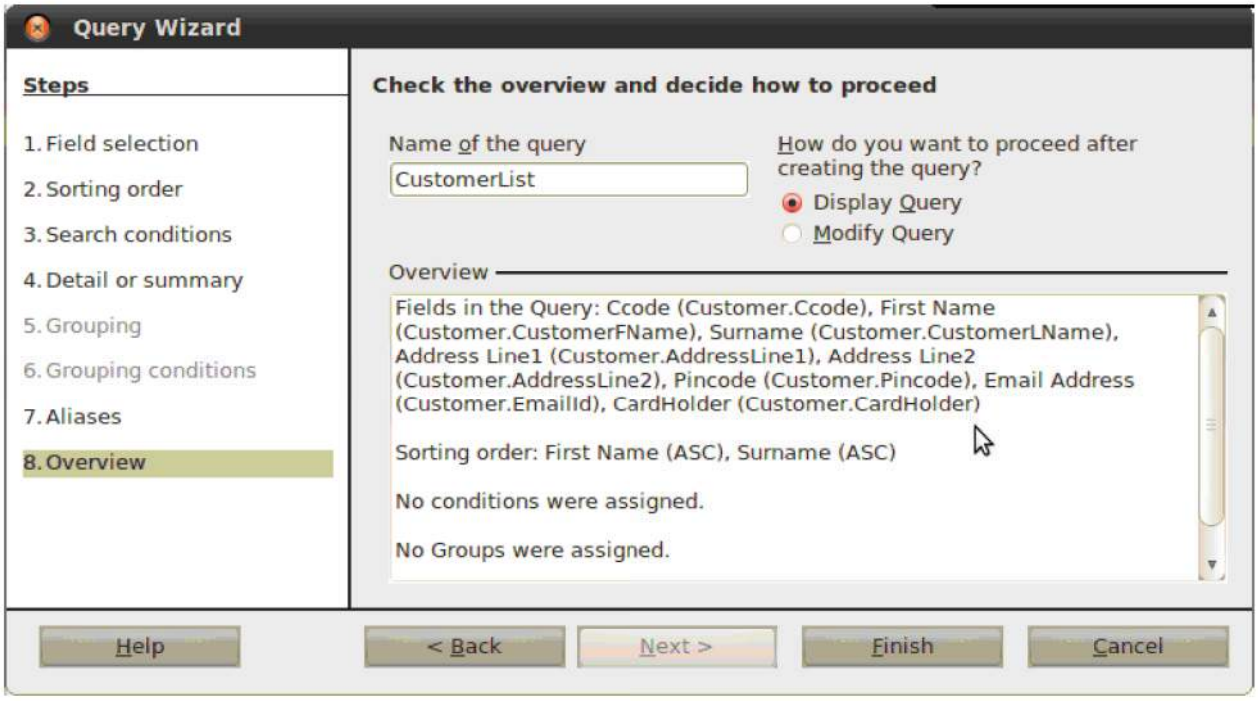


Figure 11.6 : Overview of created query

- 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.

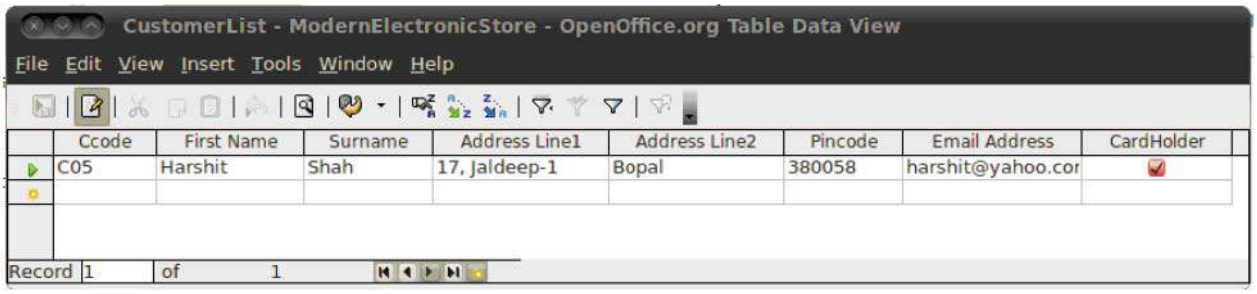


Figure 11.7 : Result of Query

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.

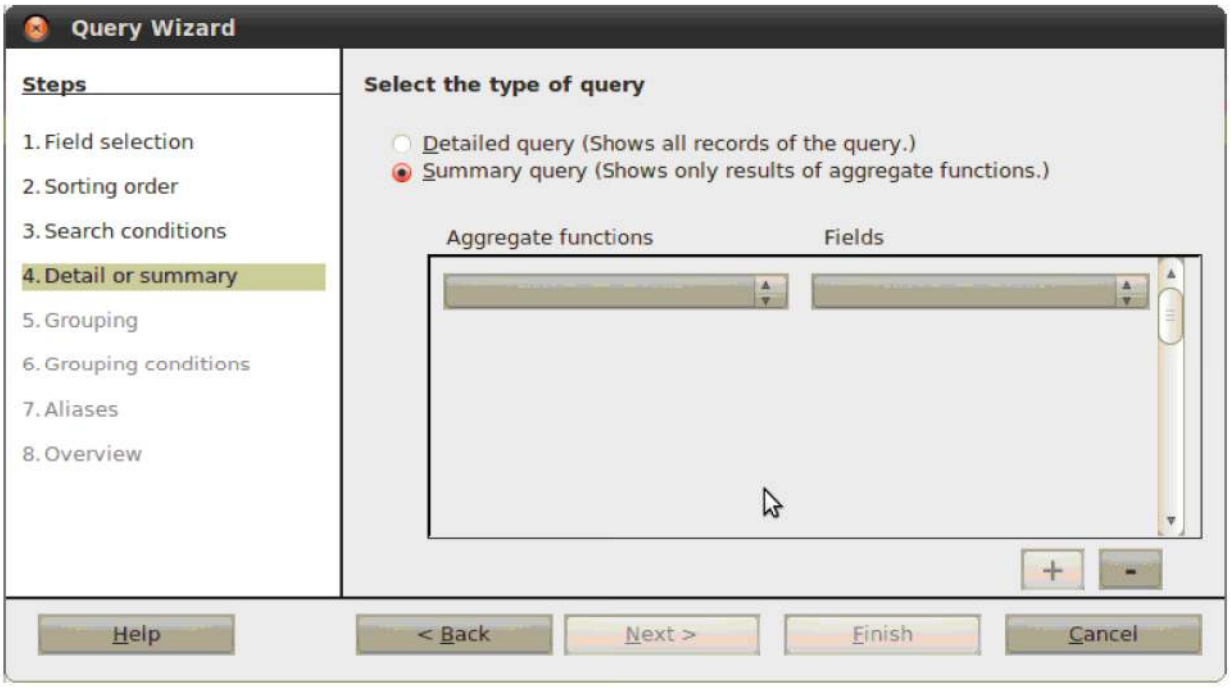


Figure 11.8 : Summarizing the result

- 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.

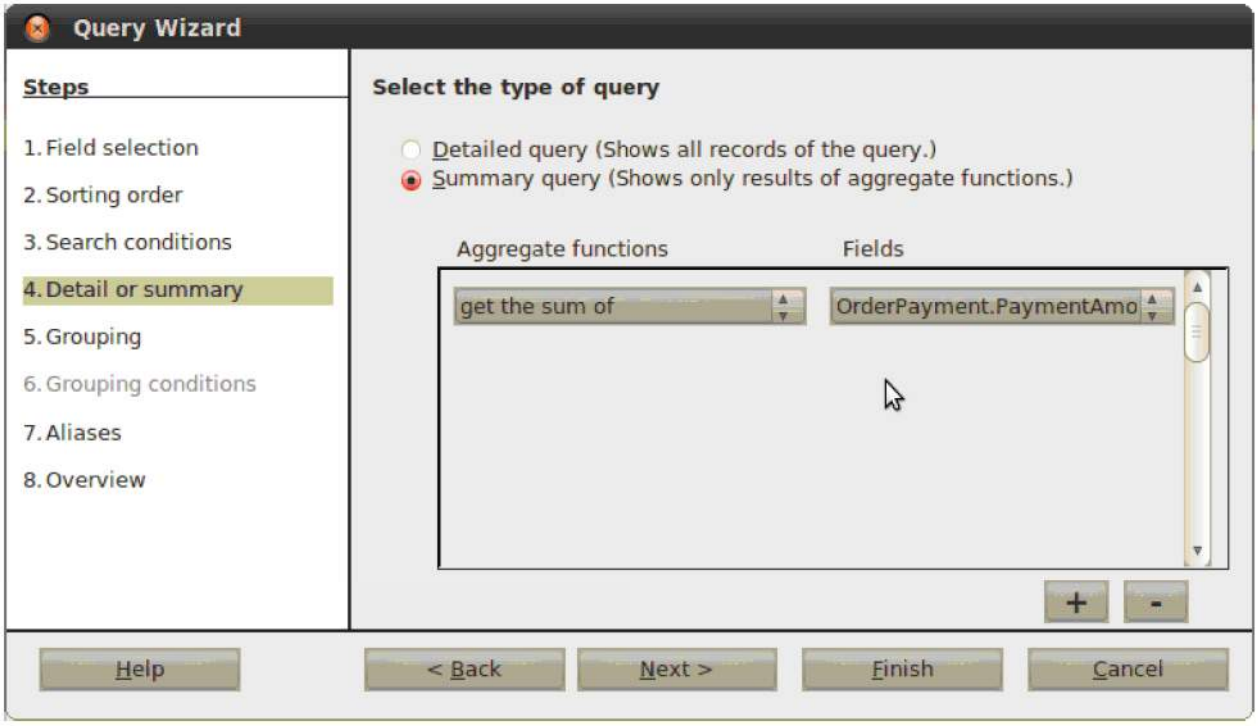


Figure 11.9 : Usage of Aggregate function

- 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.

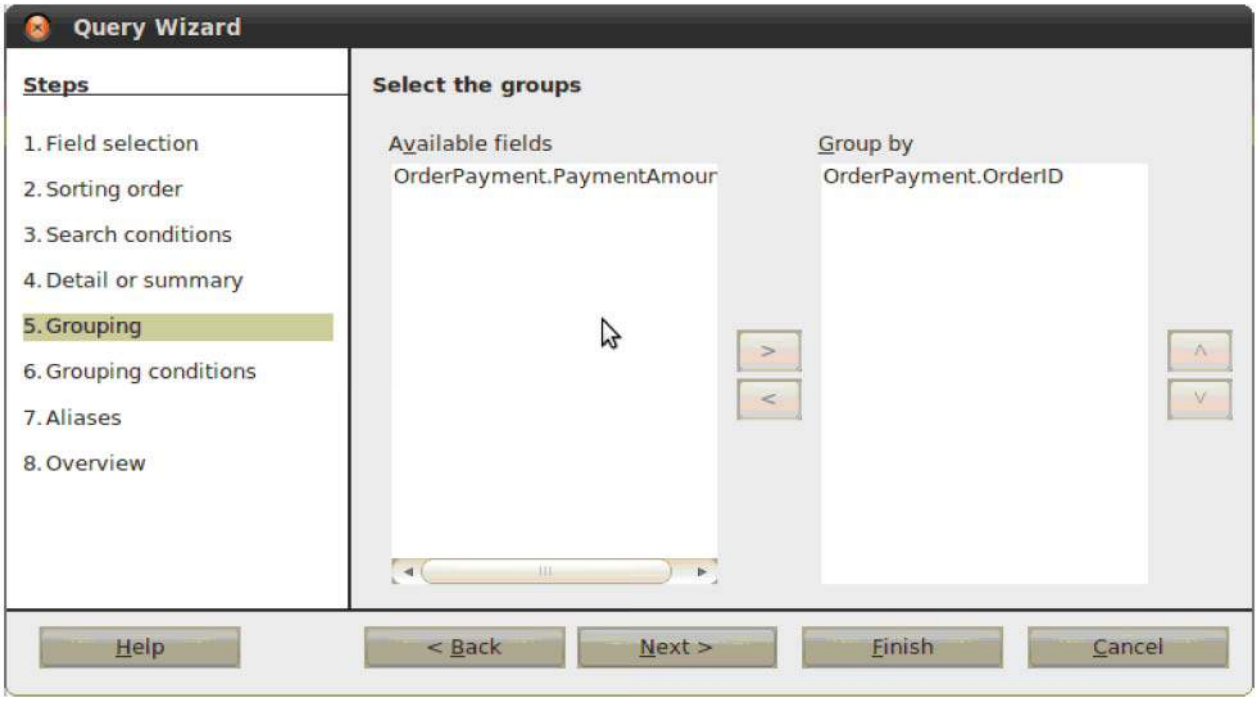


Figure 11.10 : Applying Group by on field(s)

- 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.

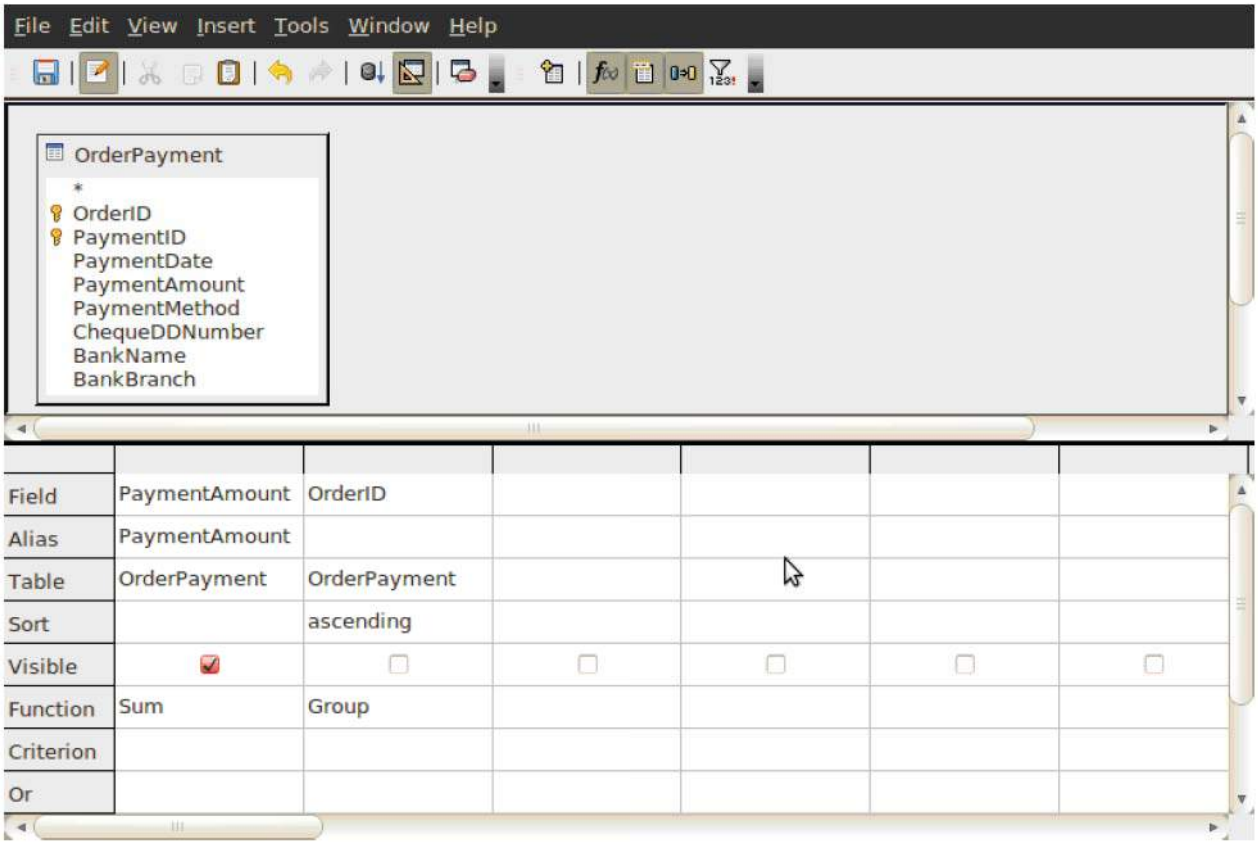


Figure 11.11 : Design View of Query

- Select the check box available under the OrderID field as shown in figure 11.12.

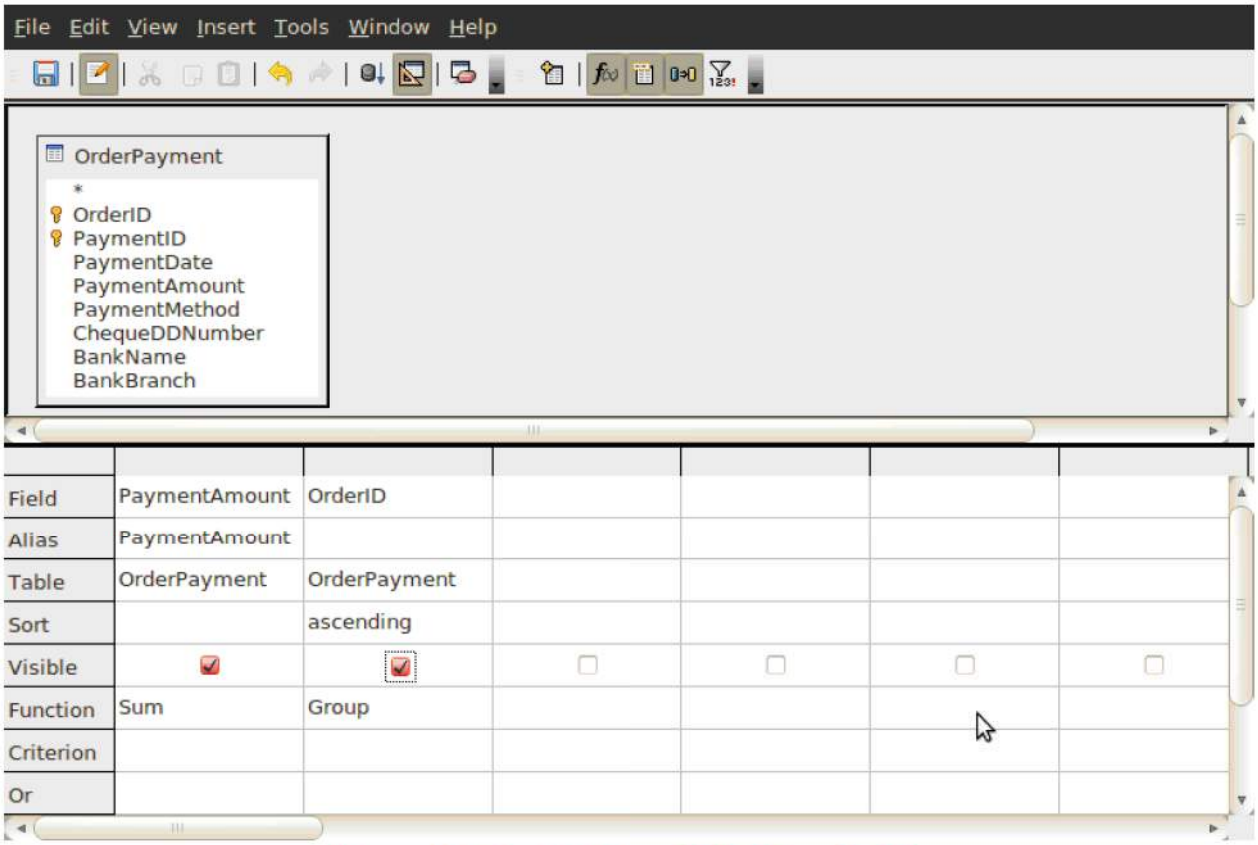


Figure 11.12 : Selecting field(s) for display

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.

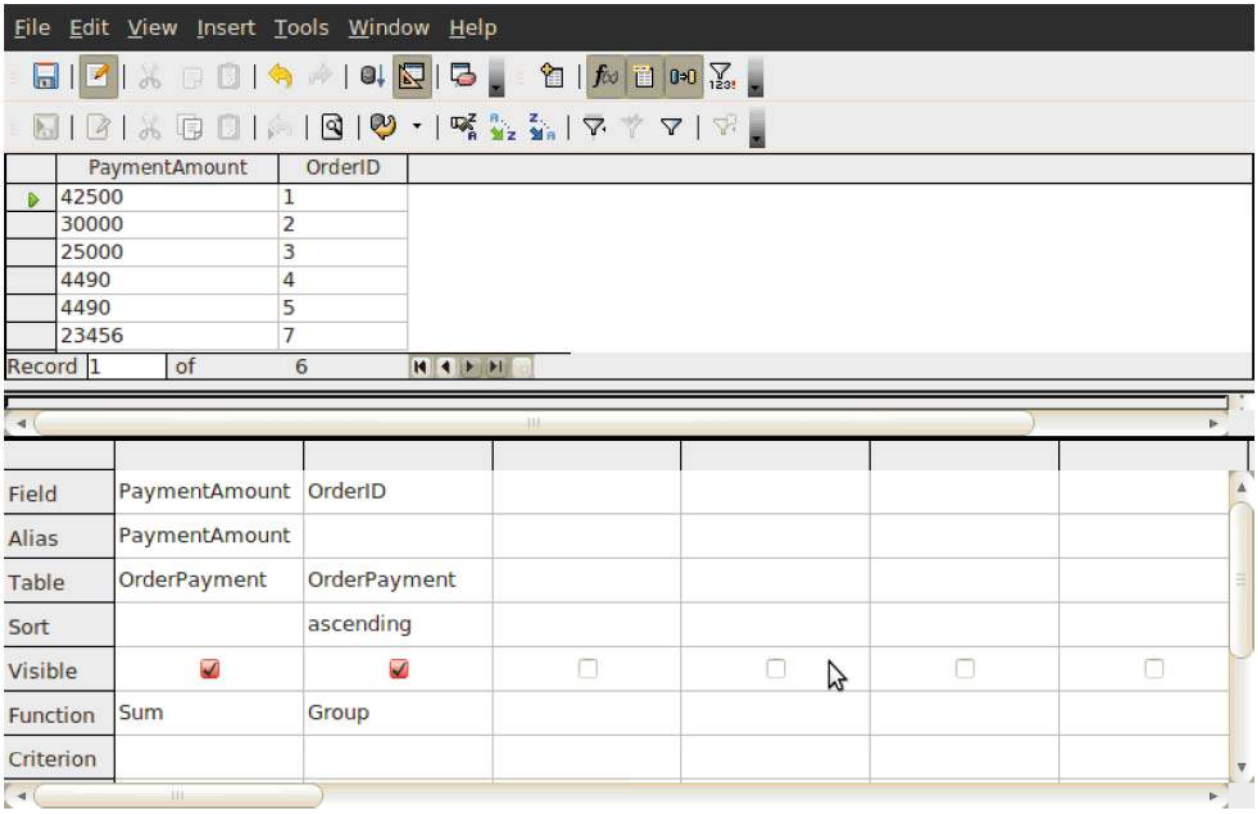


Figure 11.13 : Result of aggregation

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.

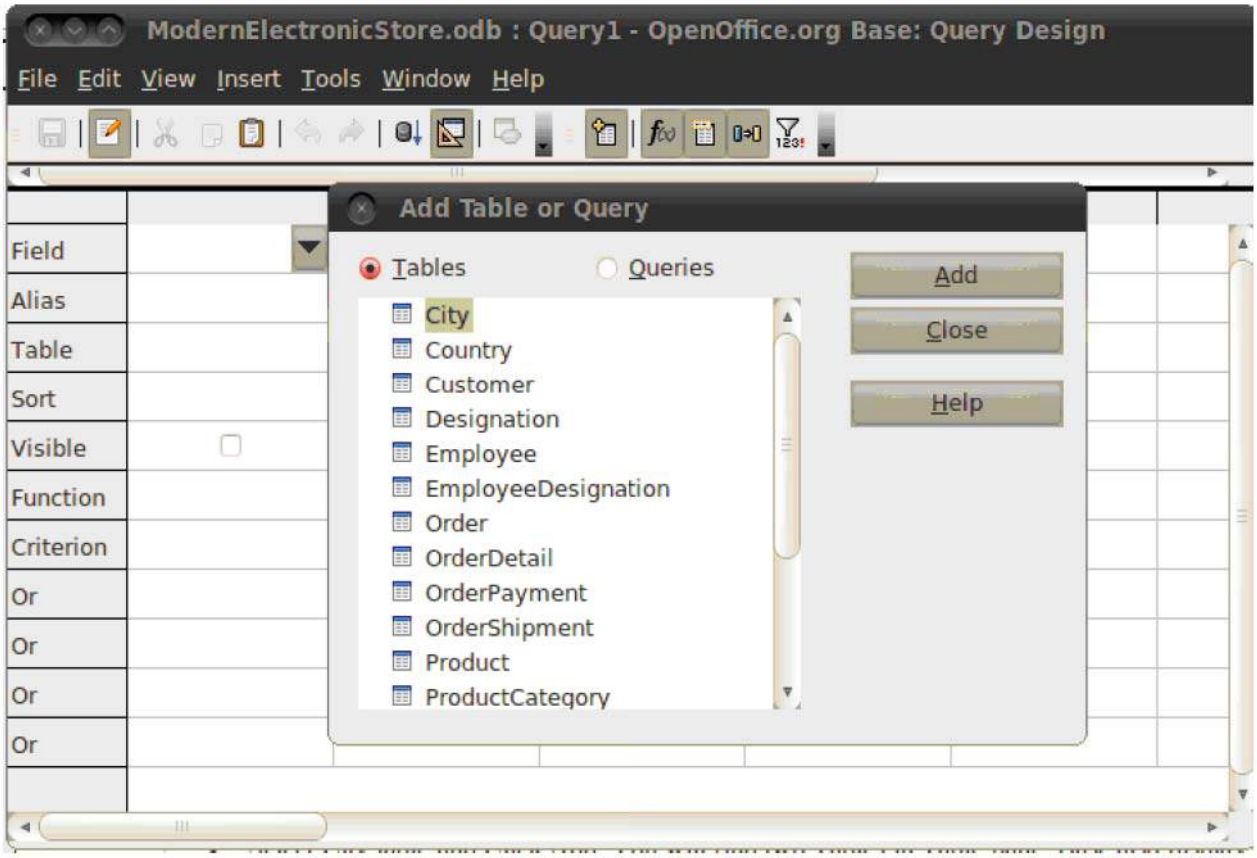


Figure 11.14 : Add Query or Table dialog box

- 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.

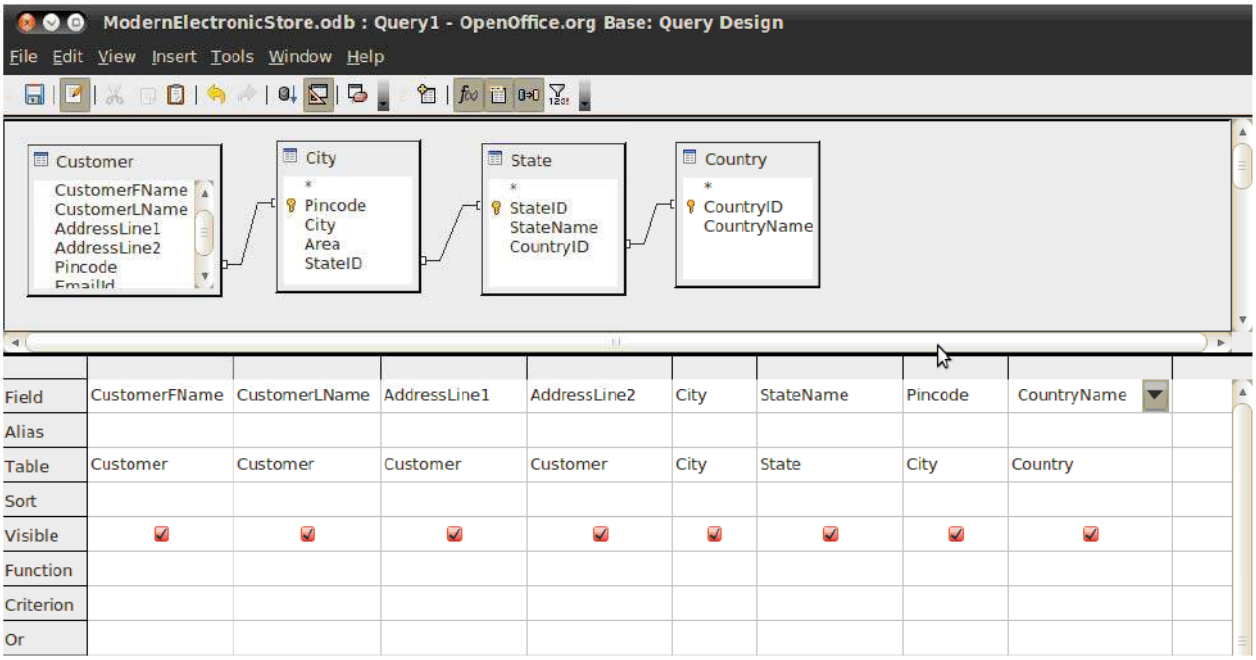


Figure 11.15 : Selection of fields

- 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 a*scending* from drop down box visible after the row heading *Sort* under the CustomerFName column. Similarly, select a*scending* 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.

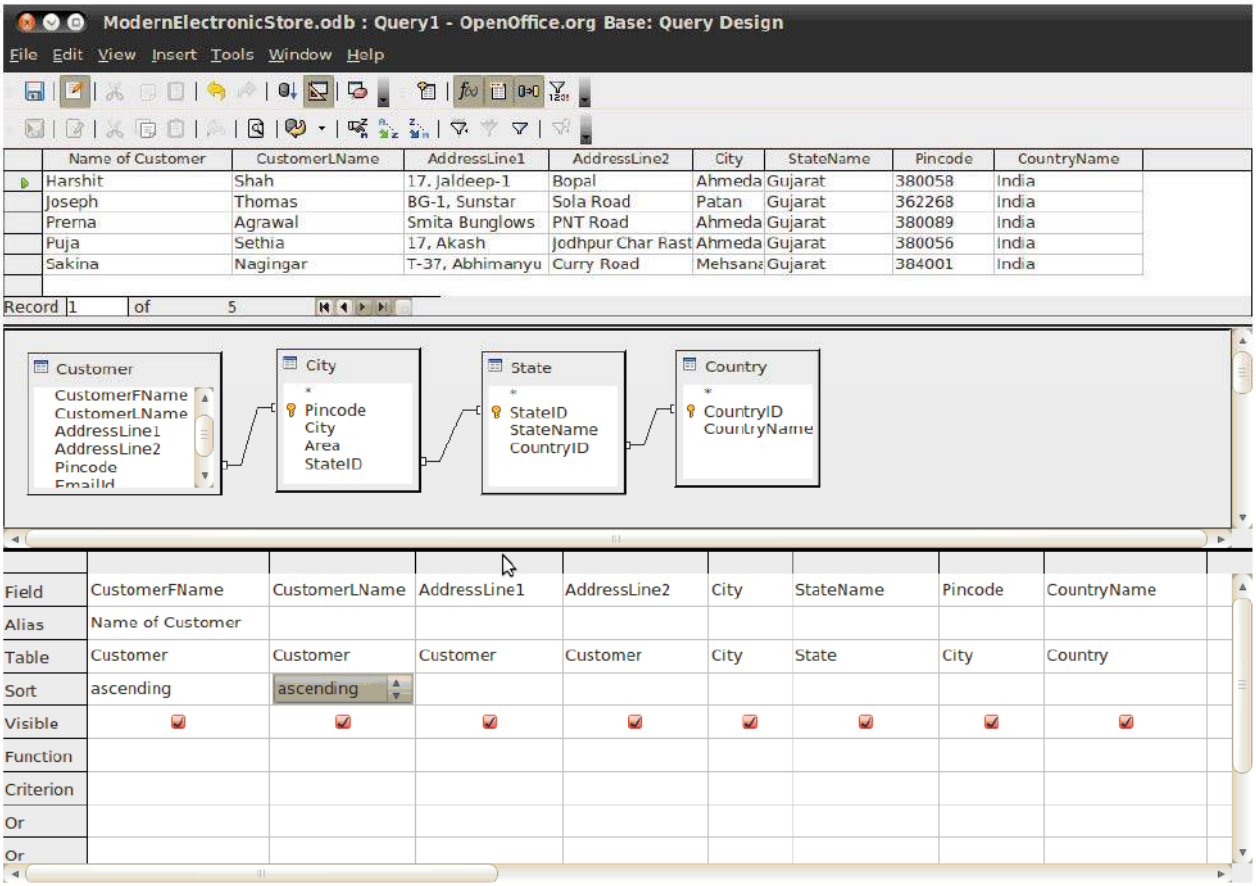


Figure 11.16 : Query output

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.

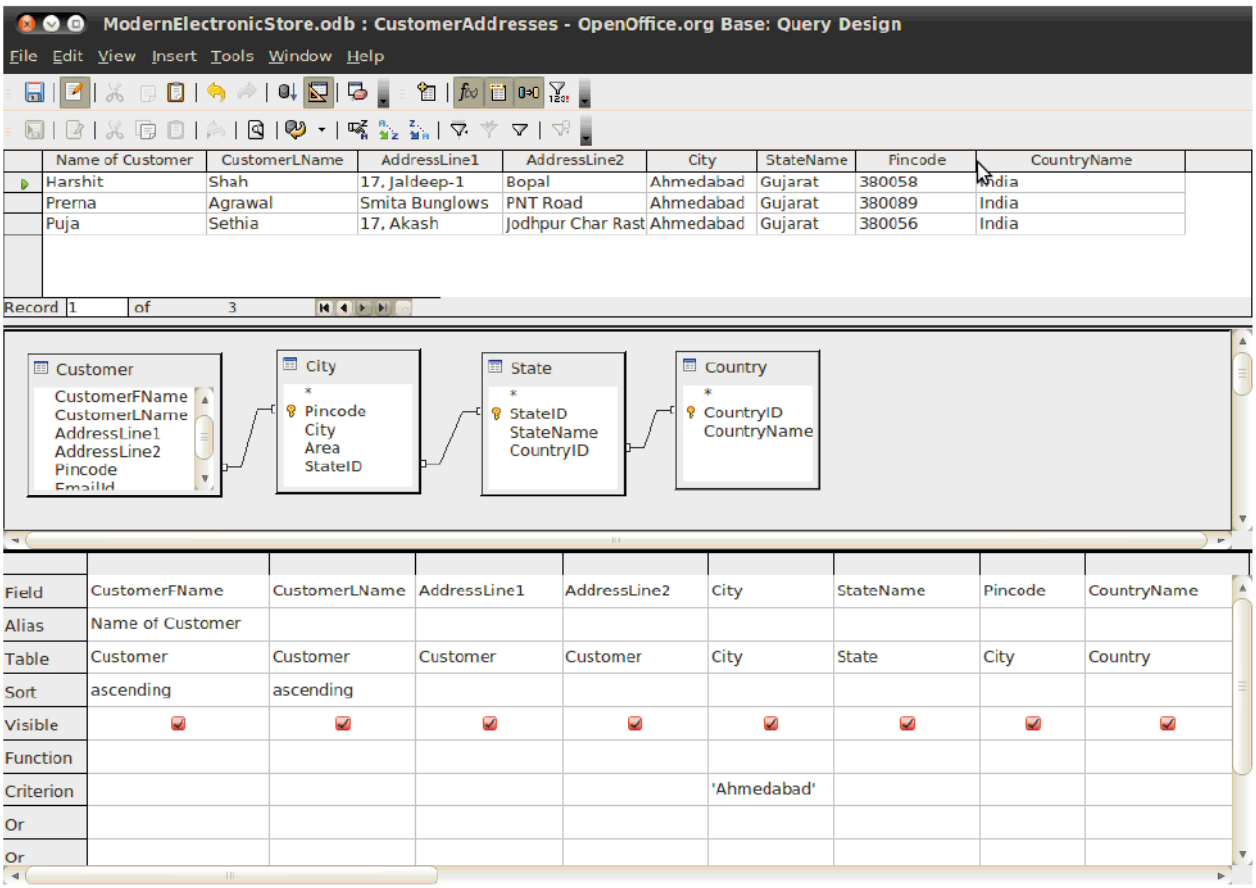


Figure 11.17 : Setting Criteria

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.

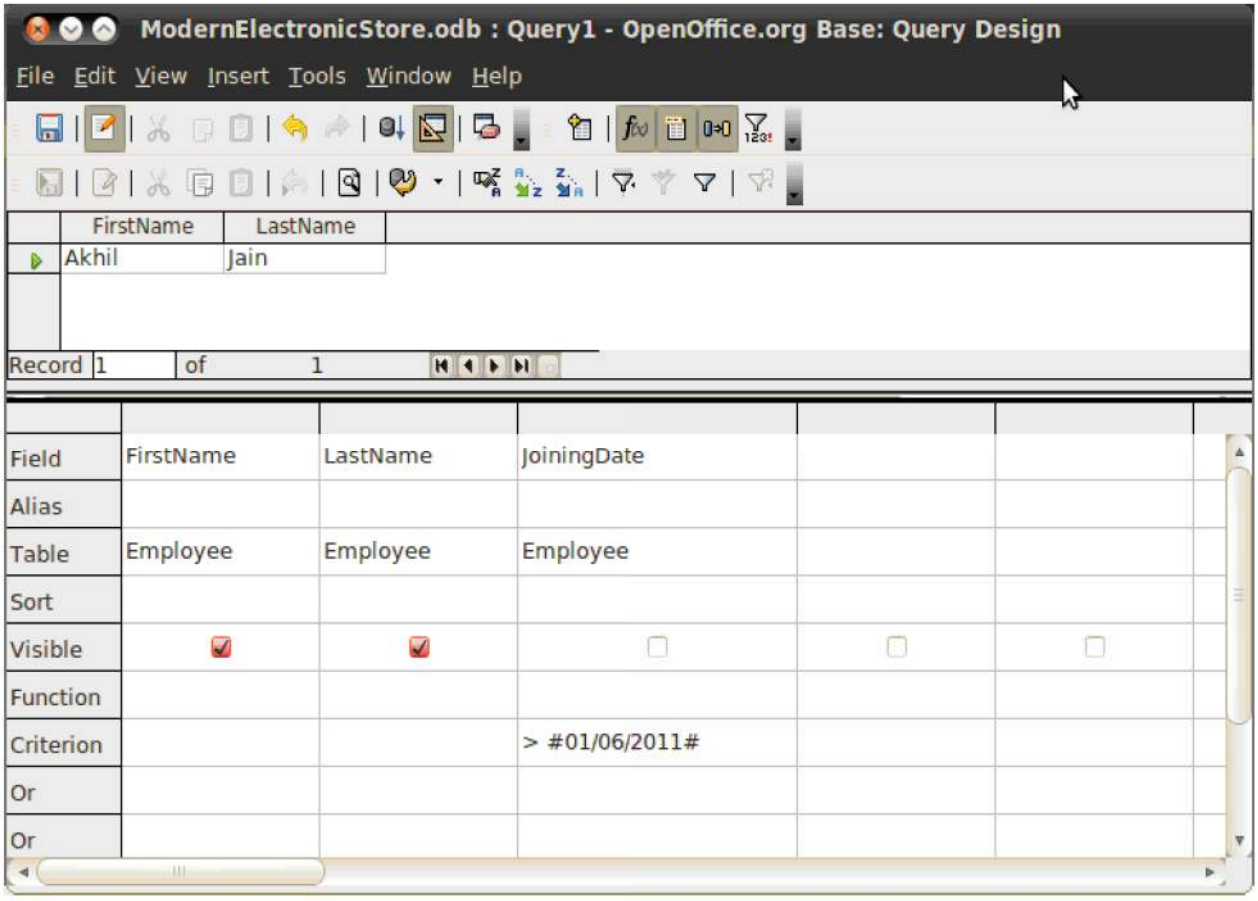


Figure 11.18 : Applying Criteria in Date Field

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.

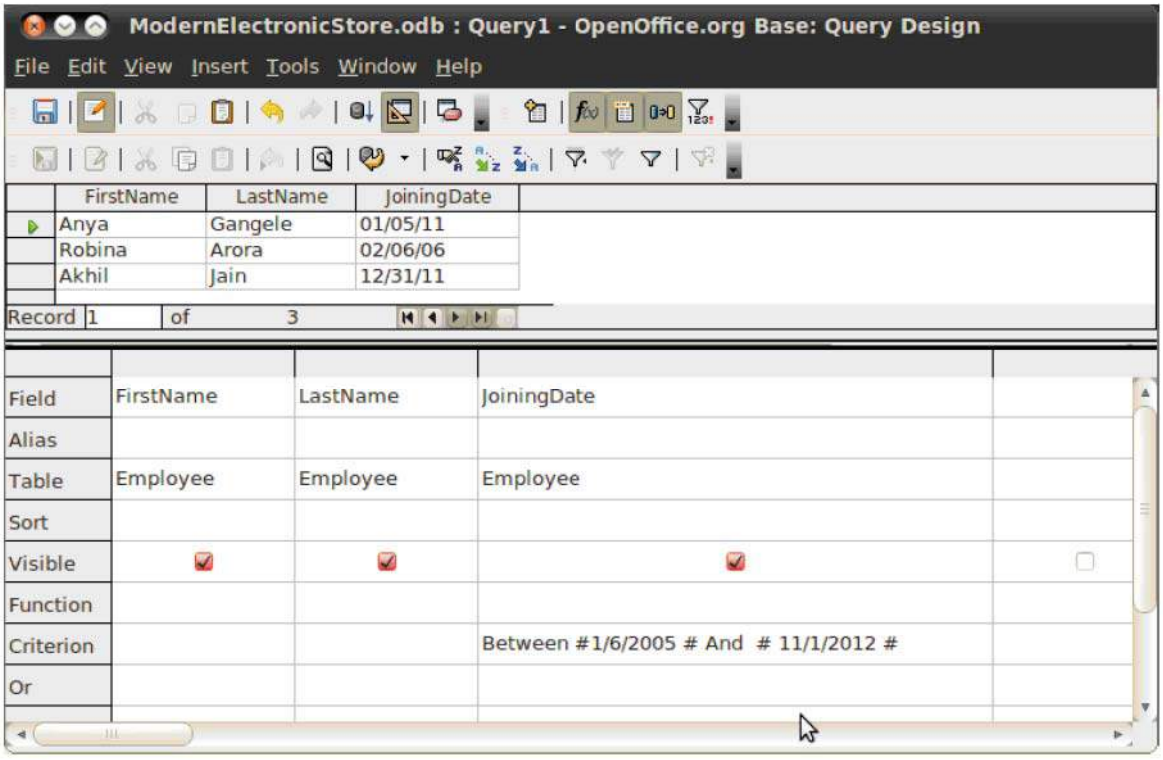


Figure 11.19 : Using the Between Operator

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.

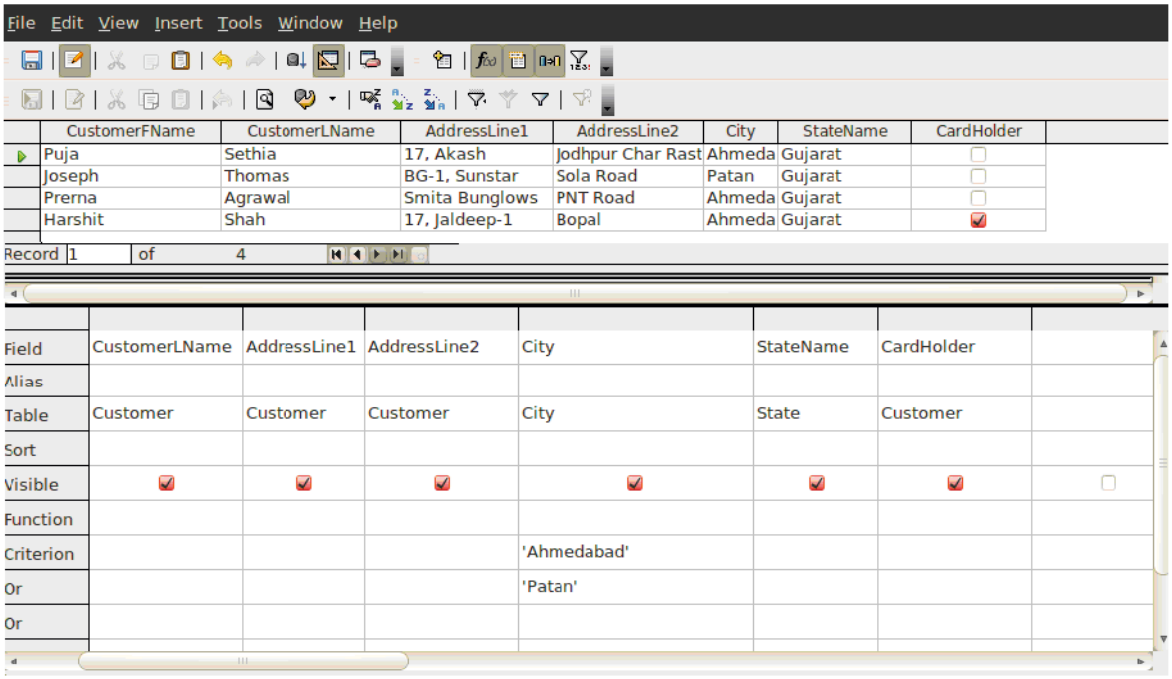


Figure 11.20 : Using Multiple Criteria on single field

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.

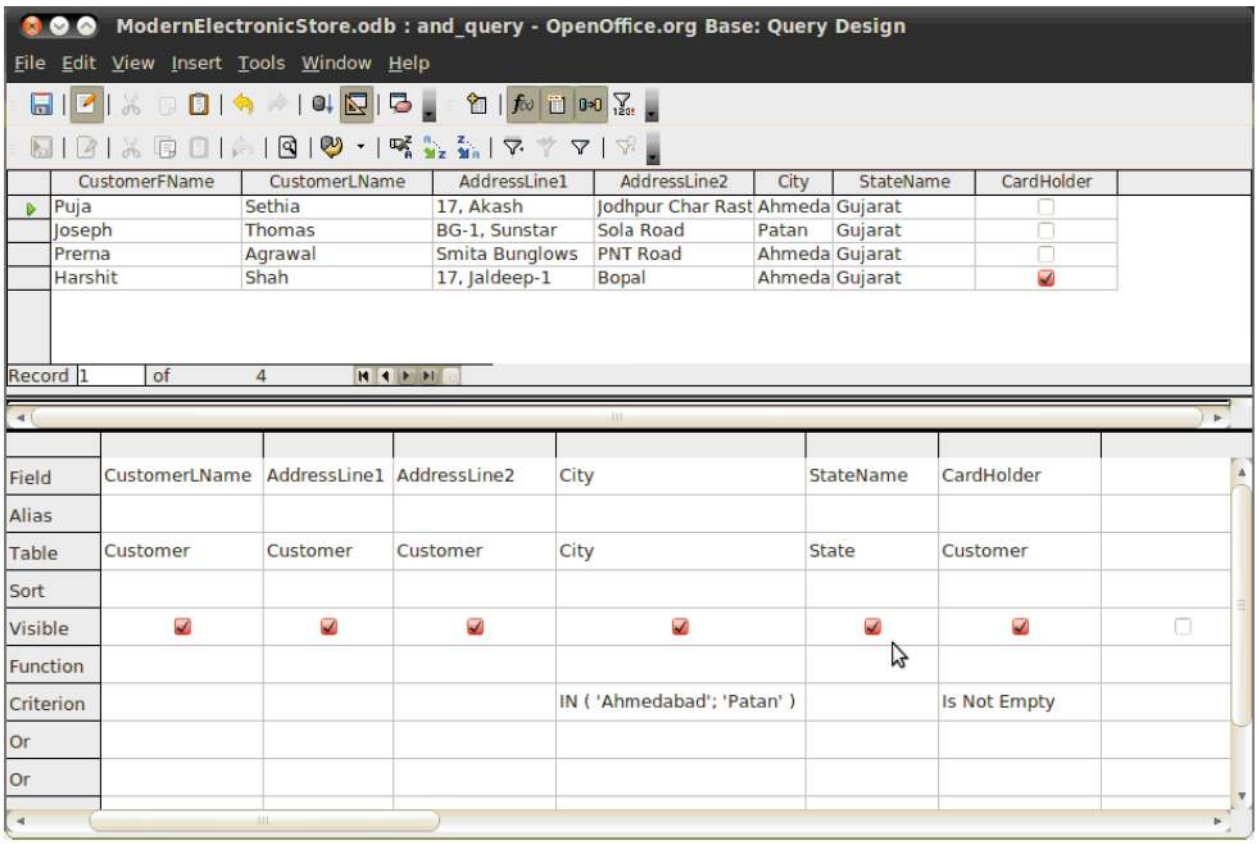


Figure 11.21 : Using Criteria on multiple fields

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.

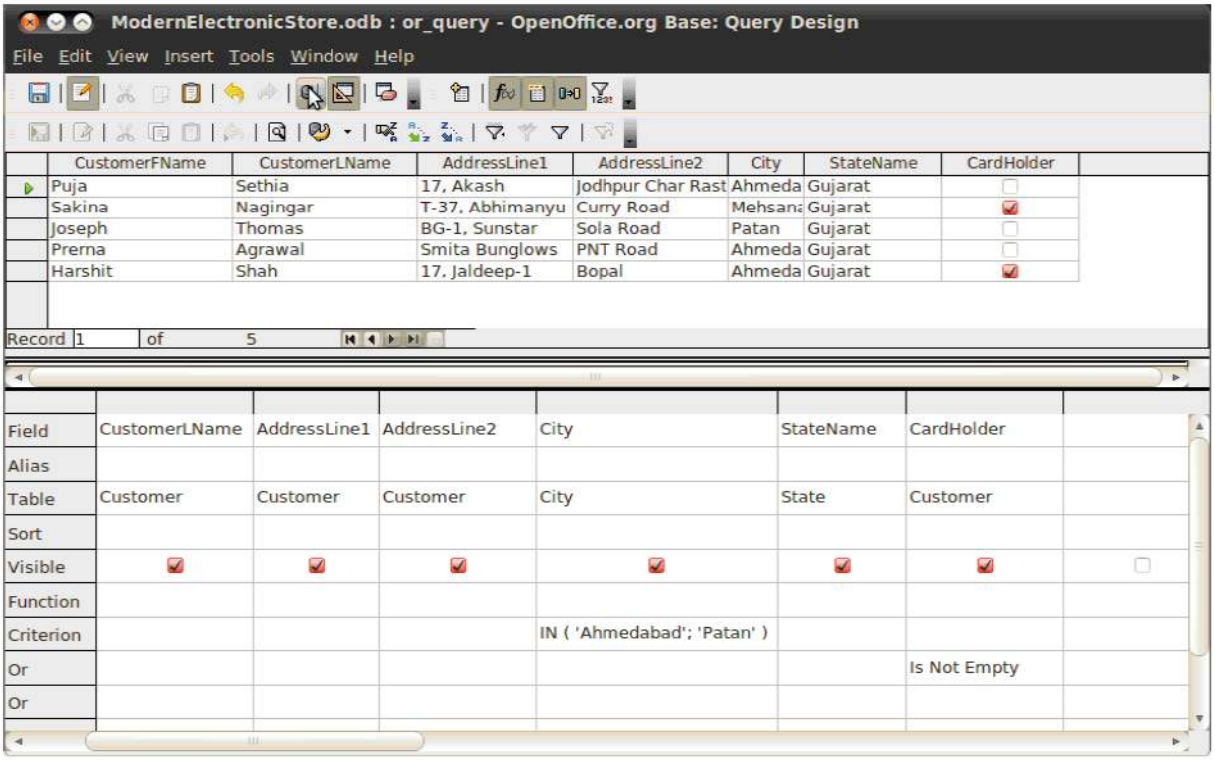


Figure 11.22 : Applying OR Criteria in multiple fields

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.

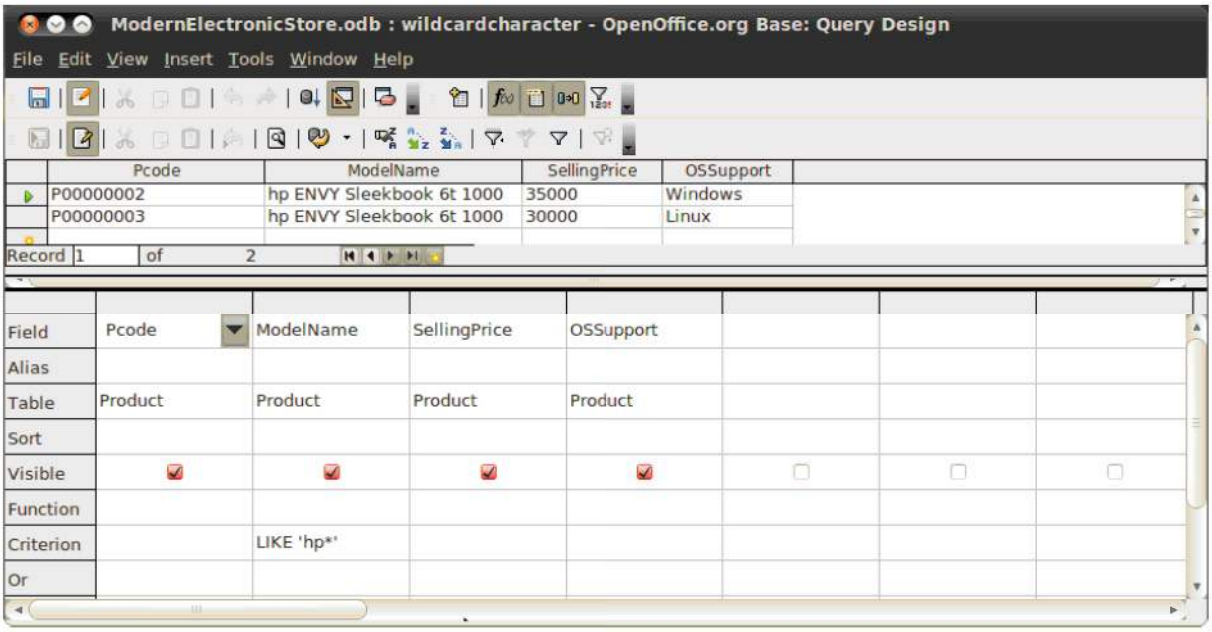


Figure 11.23 : Wild Cards

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)

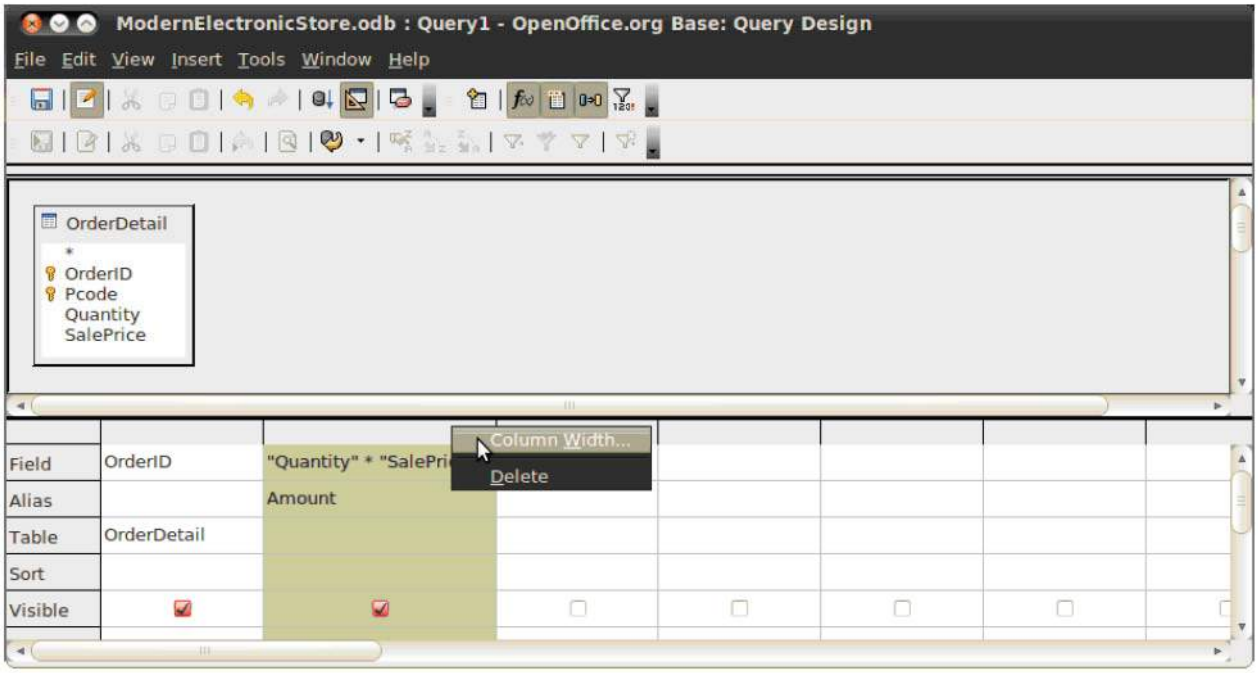


Figure 11.24 : Using Calculated Fields

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 spend 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.

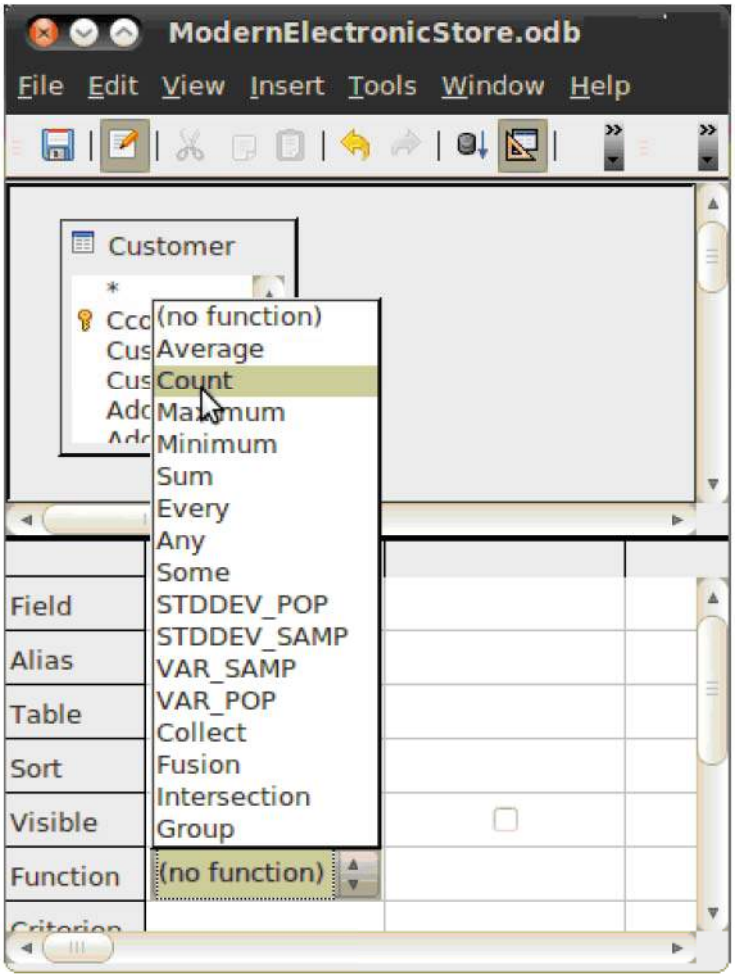


Figure 11.25 : Using Aggregate Function

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.

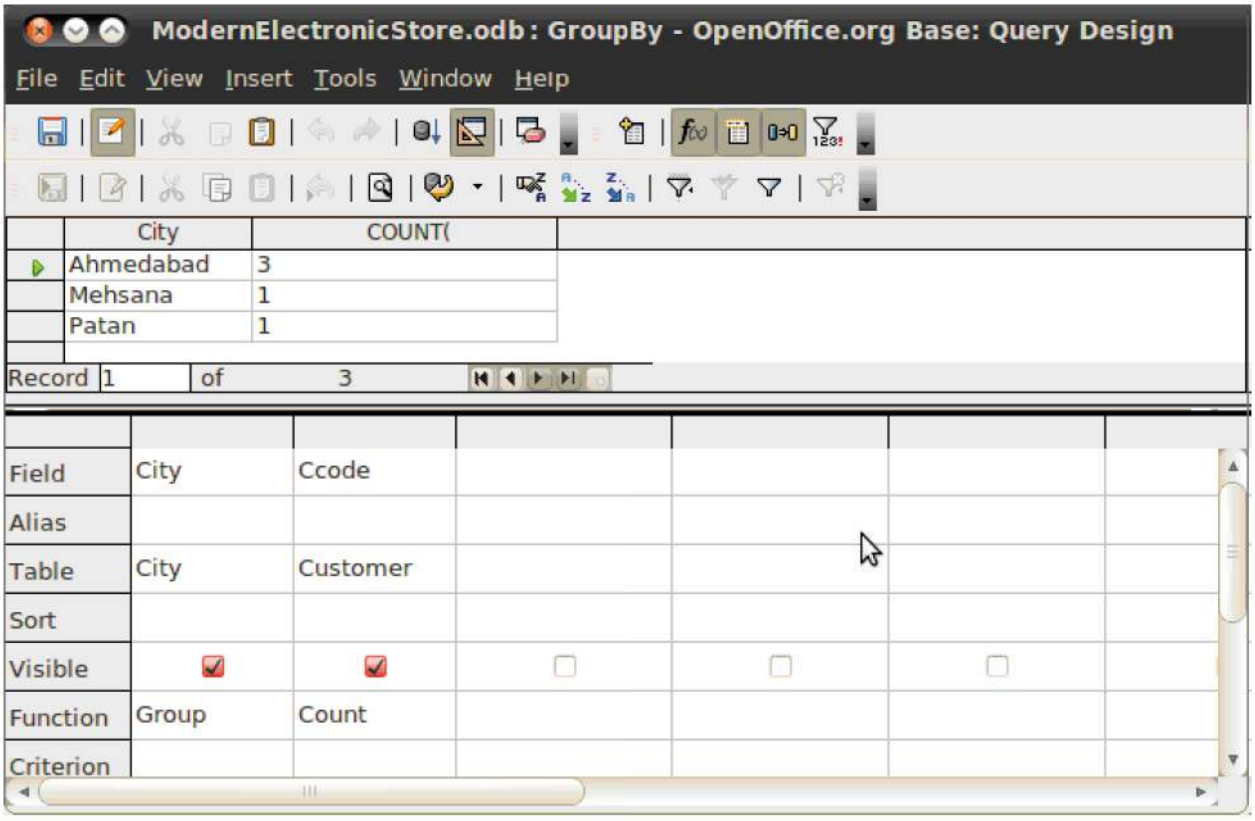


Figure 11.26 : Grouping the Records to summarize result

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 use 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 ProductCatgory 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.

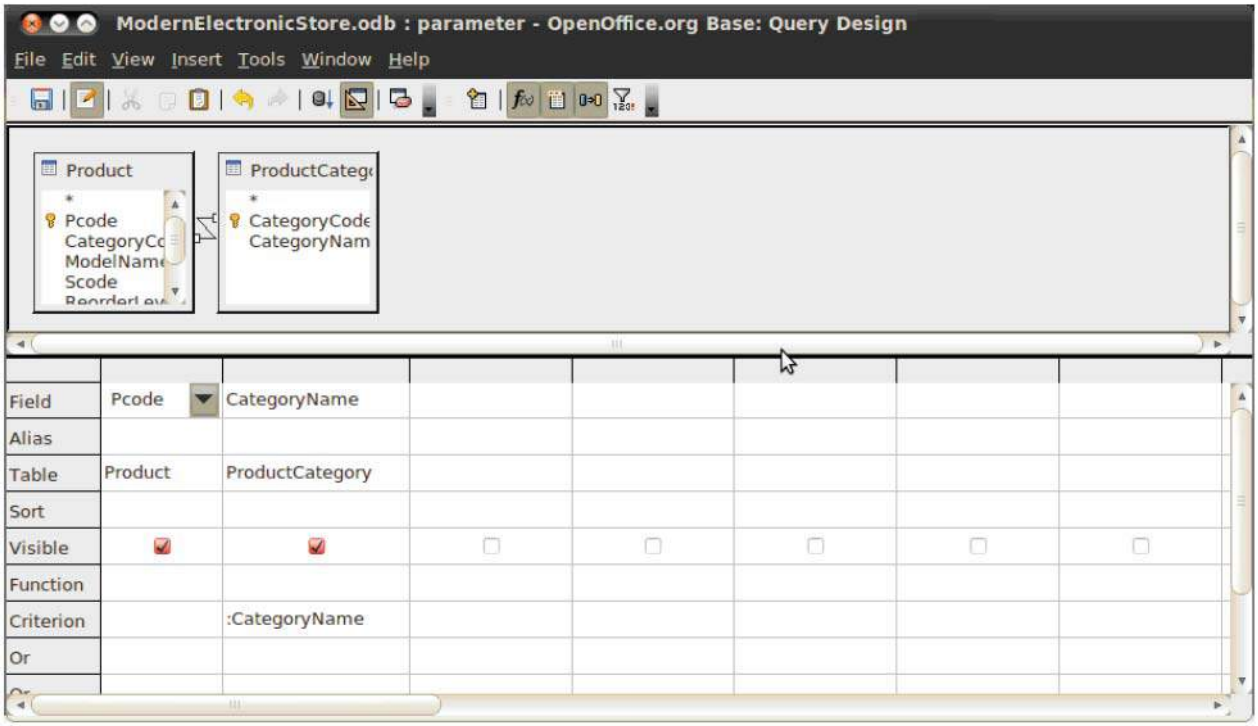


Figure 11.27 : Parameterized Query

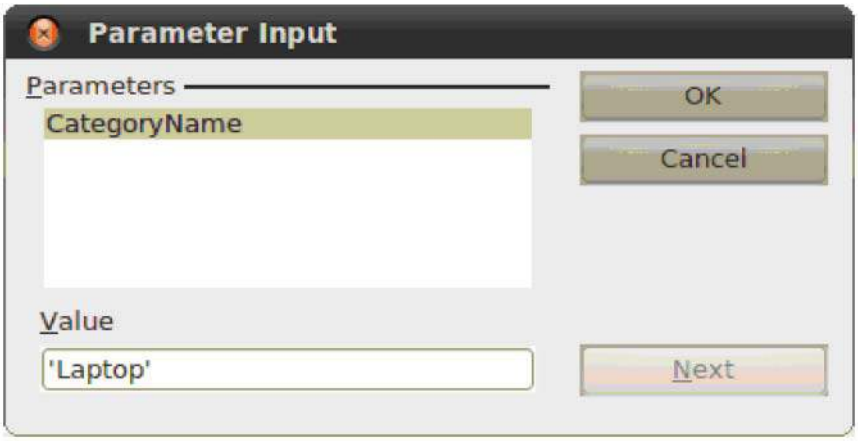


Figure 11.28 : Parameter value

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));

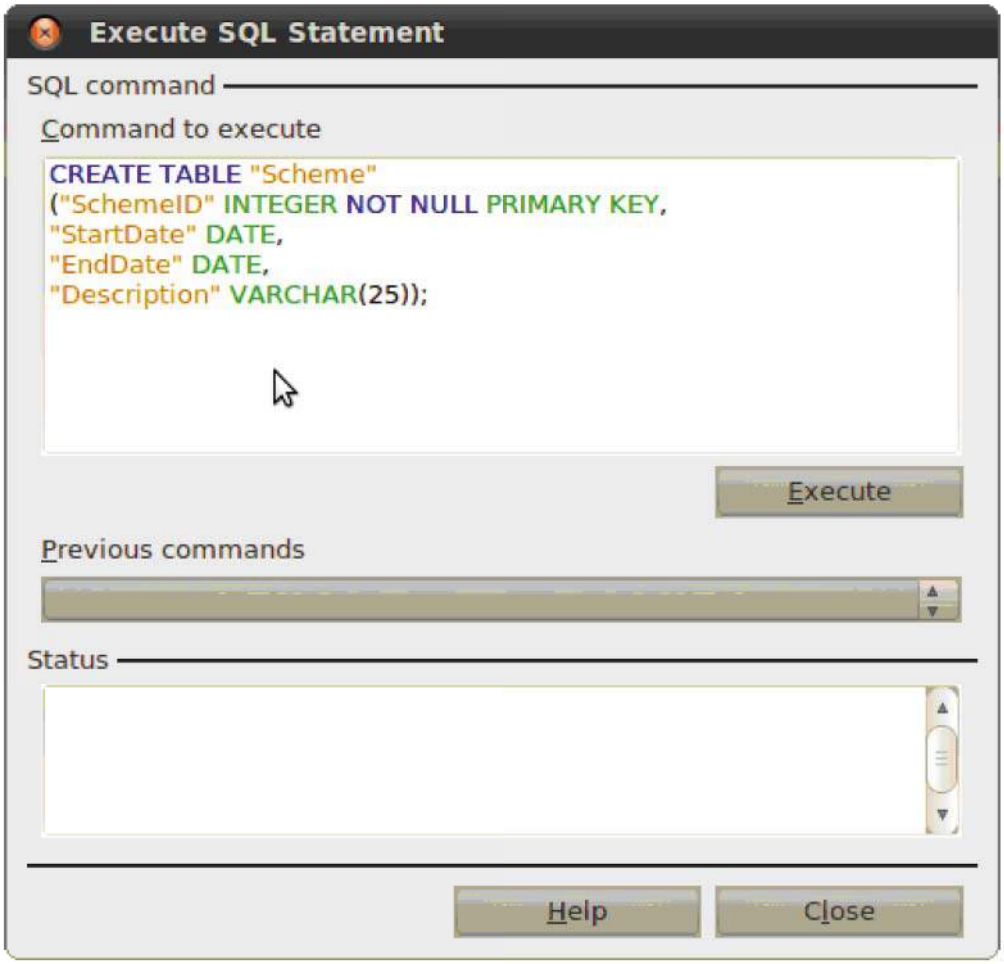


Figure 11.29 : SQL Command Window

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.

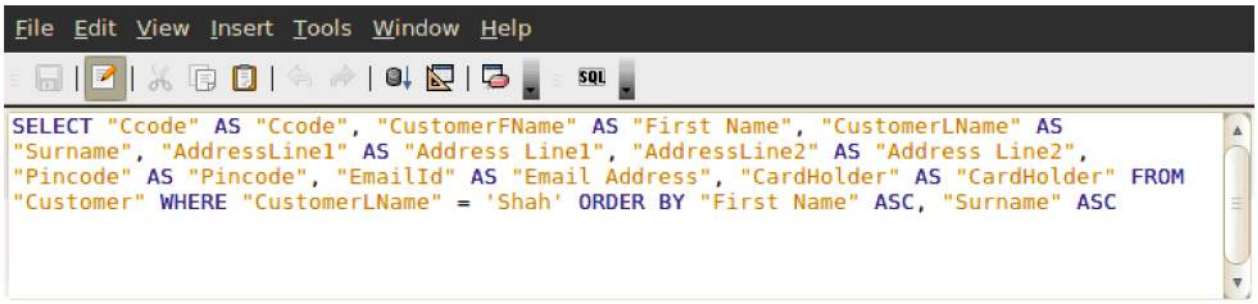


Figure 11.31 : SQL statement for Query designed using Design view

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.

3. What are wild card characters? Explain in detail each of them.

4. What are aggregate functions? What are they used for?

5. Name and explain the most commonly used aggregate functions.

6. Define Calculated Field. Explain it giving suitable example.

7. Explain grouping giving a suitable example.

8. What are parameterized query?

9. What does the value Empty mean? Explain giving a suitable example.

10. **Choose the most appropriate option from those given below :**

(1) The result of a select query is generally represented as which of the following components?

(a) Table (b) Query

(c) Constraint (d) Relation

(2) Which of the following feature of Base is used to retrieve specific information from the database?

(a) Table (b) Query

(c) Form (d) Report

(3) Which of the following is not an aggregate function?

(a) Square root (b) Sum

(c) Min (d) Max

(4) Which of the following is used to group the result of a query?

(a) Order by (b) Group by

(c) Arrange by (d) Set of

(5) Which of the following statement is false?

(a) Query can be stored as an object in database and reused.

(b) Query is written to be used only once.

(c) Query cannot be created using wizard.

(d) Query cannot be written on a query.

(6) What is an alias?

(a) Alias is creation of duplicate query.

(b) Alias is used to give meaningful name to fields selected in a query.

(c) Alias is used to print query.

(d) Alias is nowhere related to query.

(7) Which delimiter is used to surround the text in a query criterion?

(a) " (double quote) (b) ' (single quote)

(c) $ (dollar) (d) # (hash)

(8) Which of the following operator is used with wild card character?

(a) Like (b) is

(c) equal (d) =

(9) Which of the following operator is used as wild card character in Base?

(a) & (b) +

(c) - (d) *

(10) Which query, when run, displays a dialog box asking to enter the value to match the criteria for retrieving the data?

(a) Select Query (b) Insert Query

(c) Parameter Query (d) Update query

(11) The Parameter name in parameter query is preceded by which symbol?

(a) Comma (b) Colon

(c) Question mark (d) Explanation mark

(12) SQL stands for?

(a) Simple Query Language

(b) Structured Query Language

(c) Simple Question for large databases

(d) Structured queries for large databases

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.

12 Forms and Reports

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.

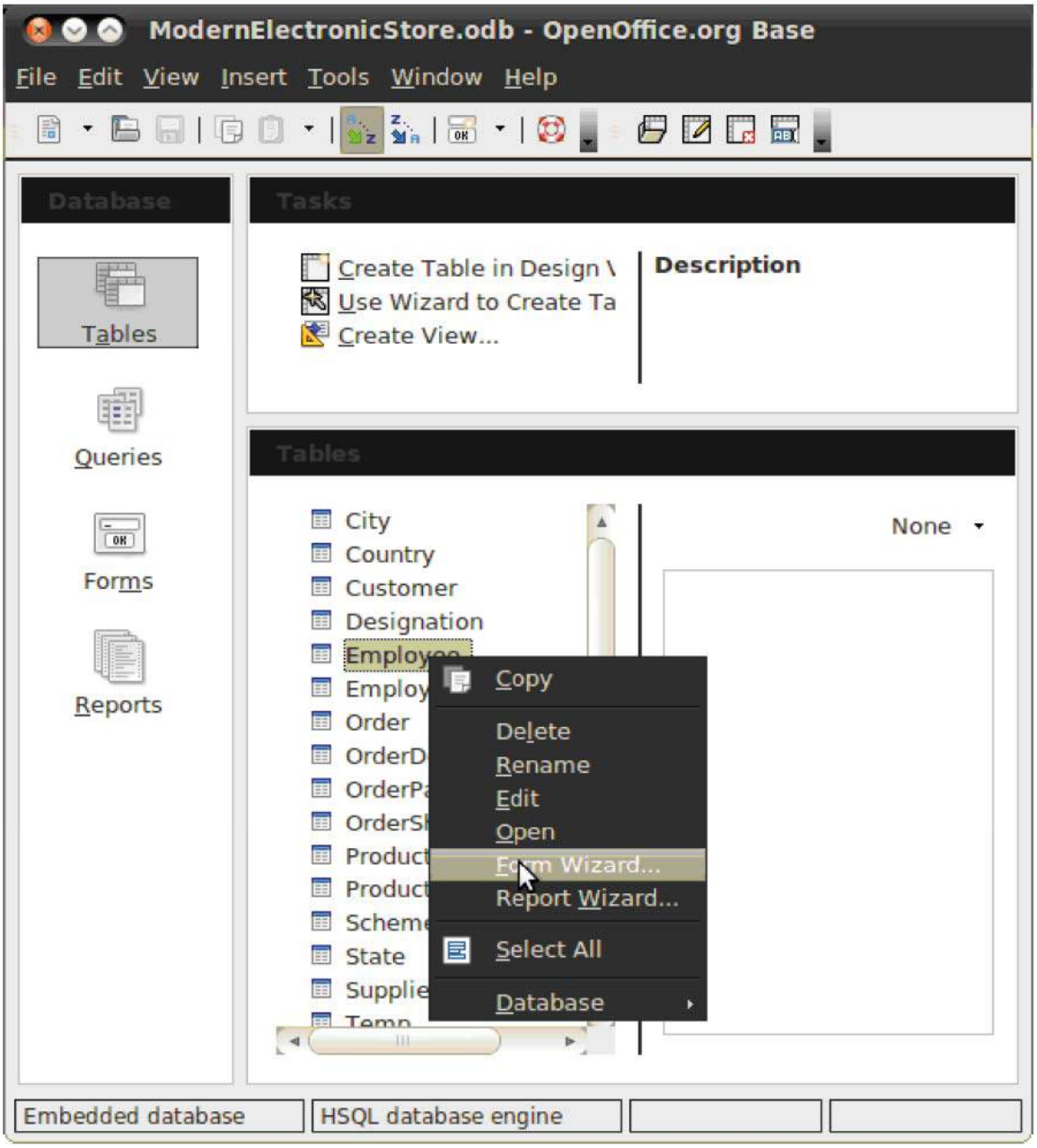


Figure 12.1 : Creating a Form Using Wizard

- 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.

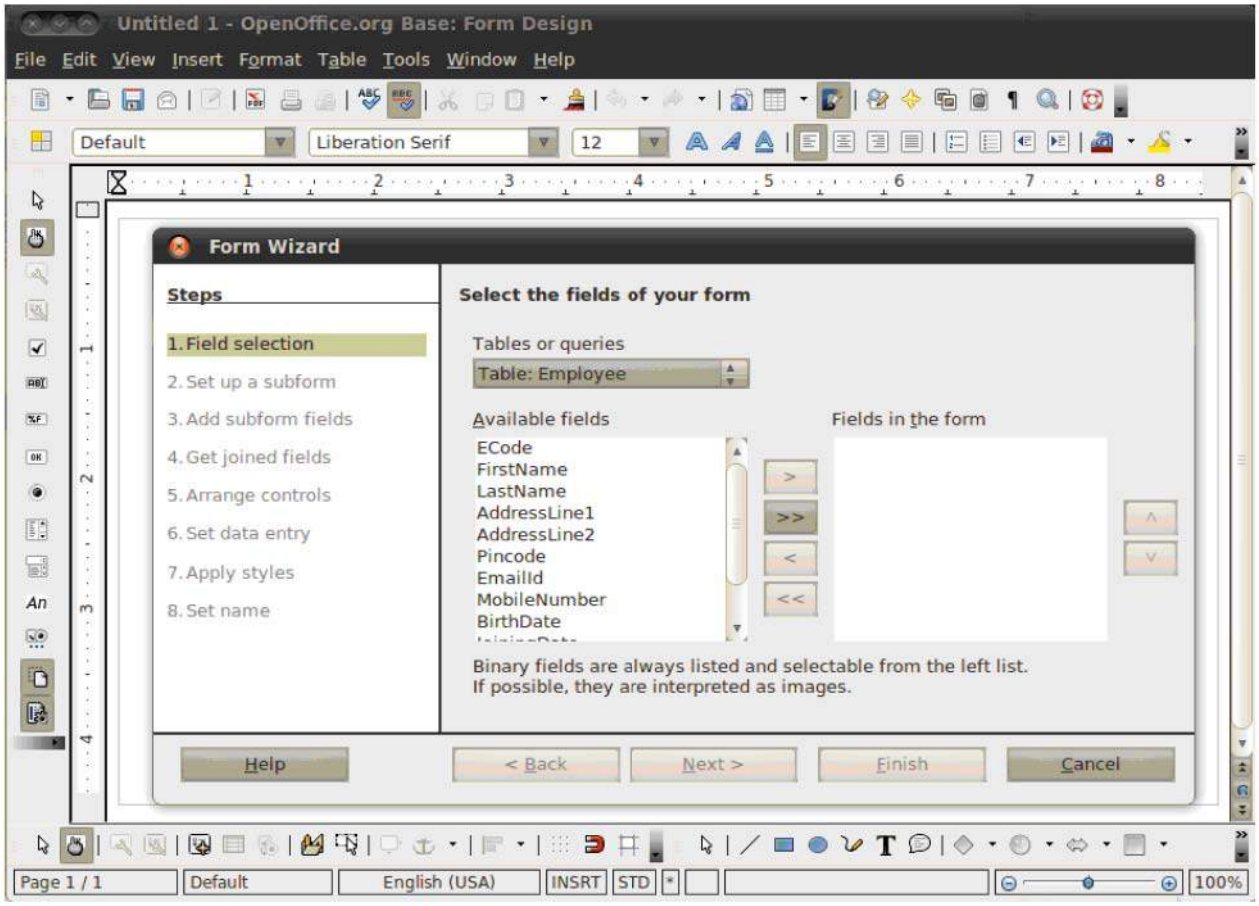


Figure 12.2 : Field Selection for Form creation

- 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.

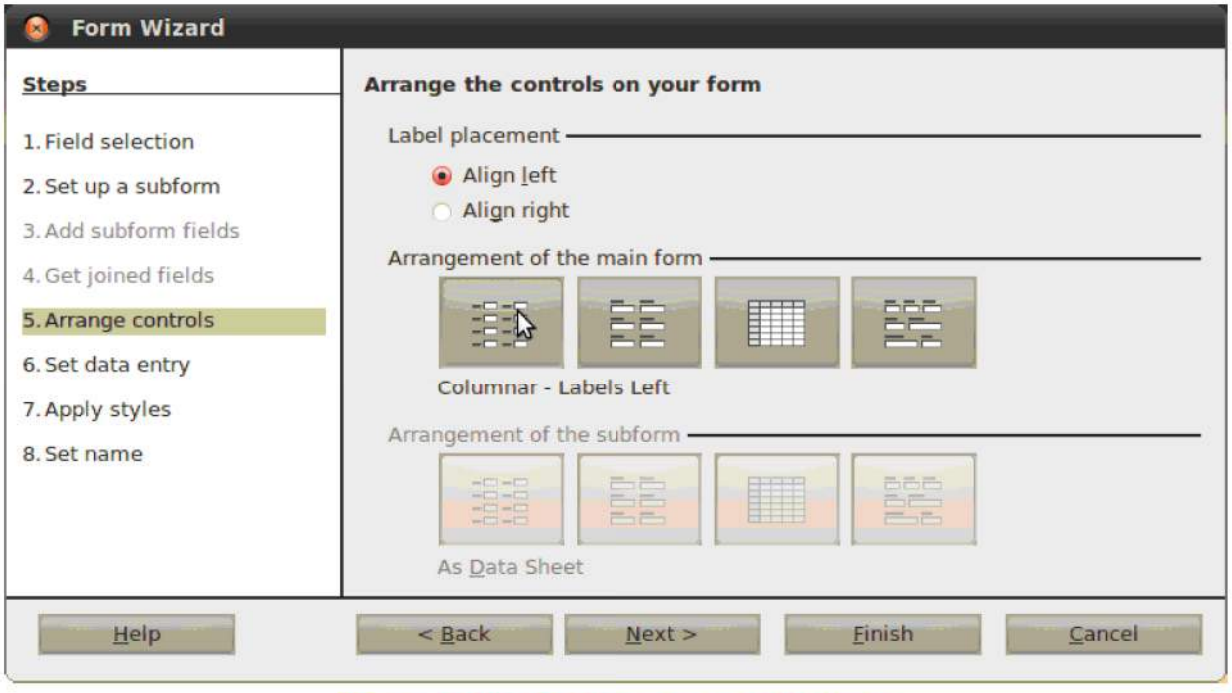


Figure 12.3 : Arrangements of Controls

- 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.

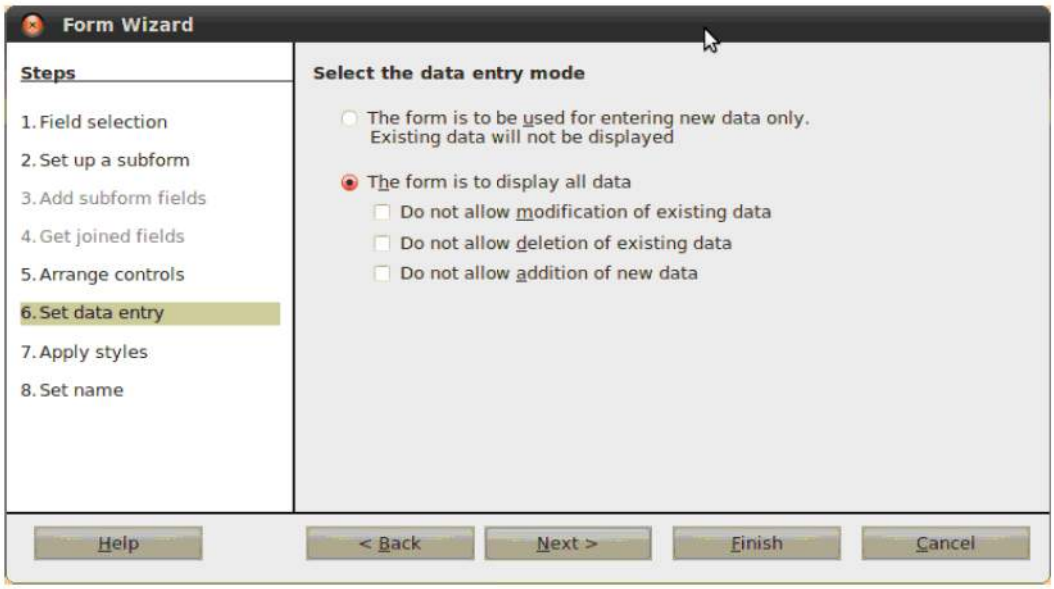


Figure 12.4 : Data entry control settings

- 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.

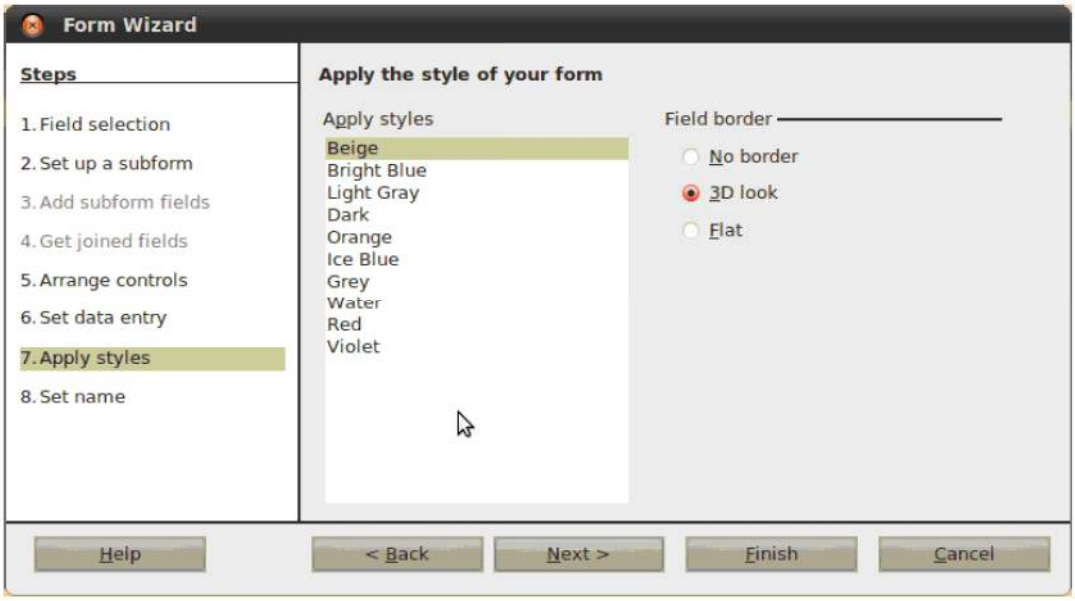


Figure 12.5 : Applying Styles

- 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.

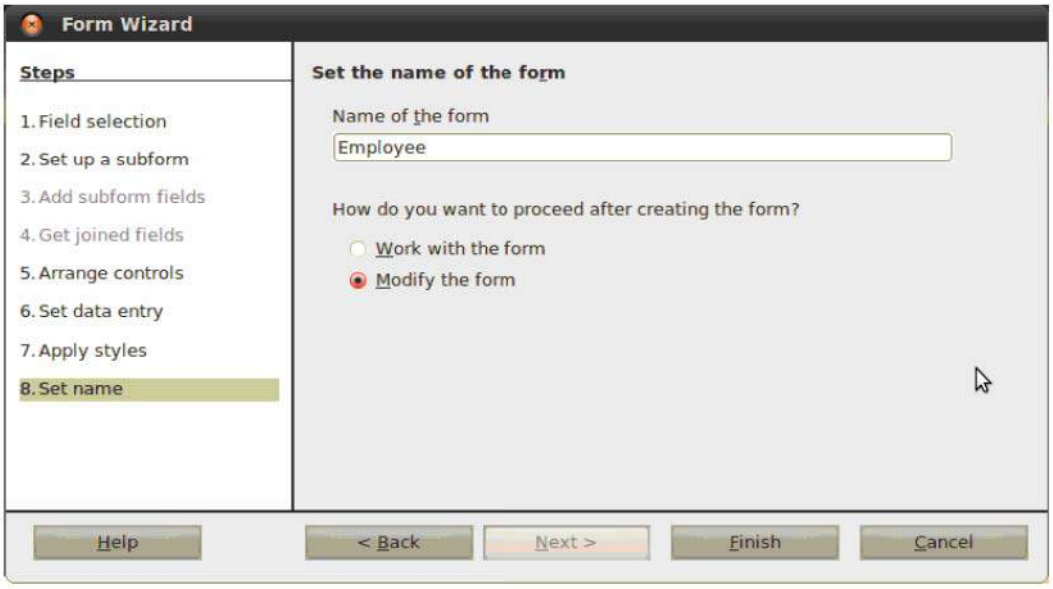


Figure 12.6 : Naming form

- 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.

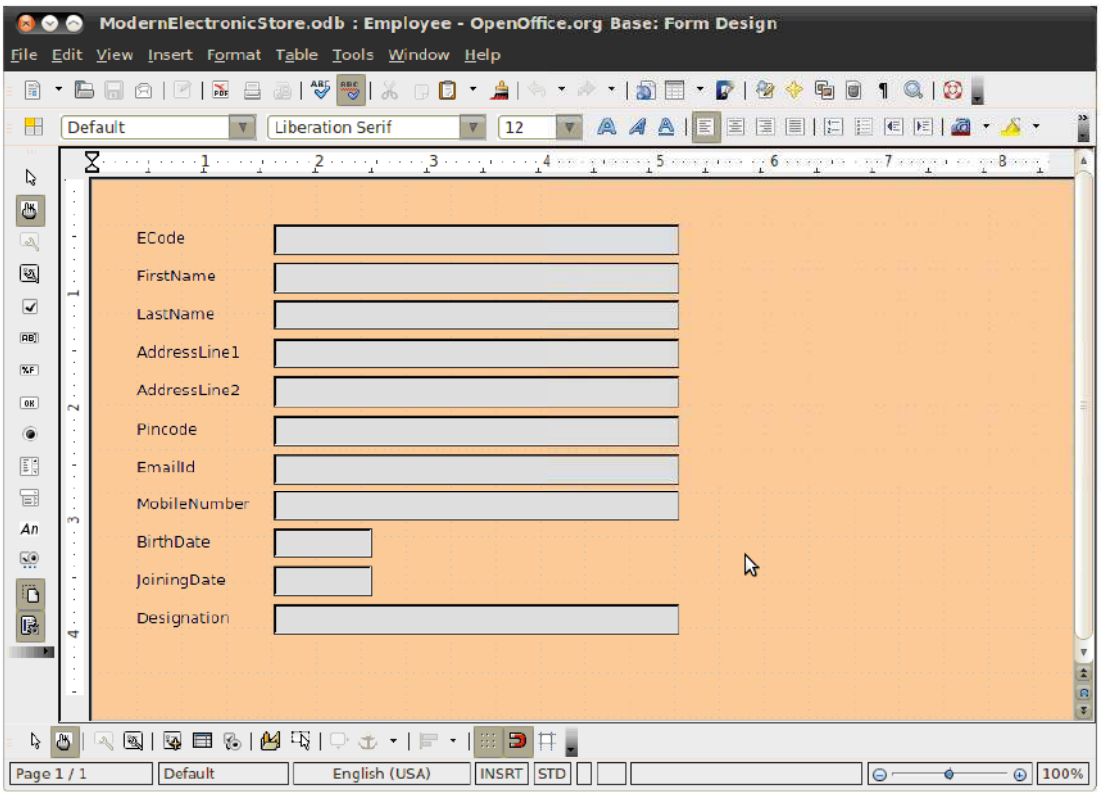


Figure 12.7: Form in modify mode

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 C*ontrol...* 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.

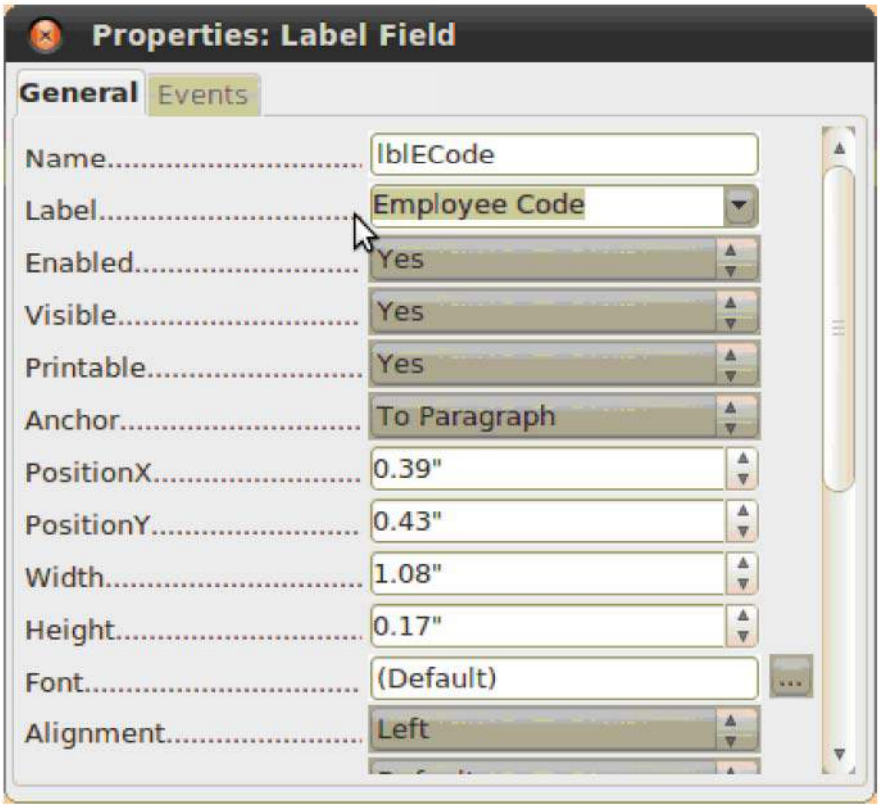


Figure 12.8 : Properties Dialog box

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.

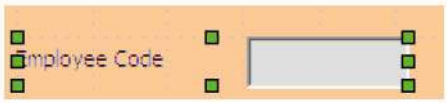


Figure 12.9 : A Selected control

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.

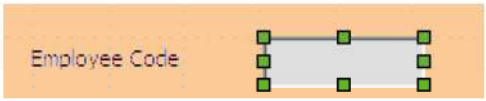


Figure 12.10 : Selecting a field of control

Changing Control size

Let us increase the width of the Employee Code field.

Figure 12.11 : Changing Control size

- 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).

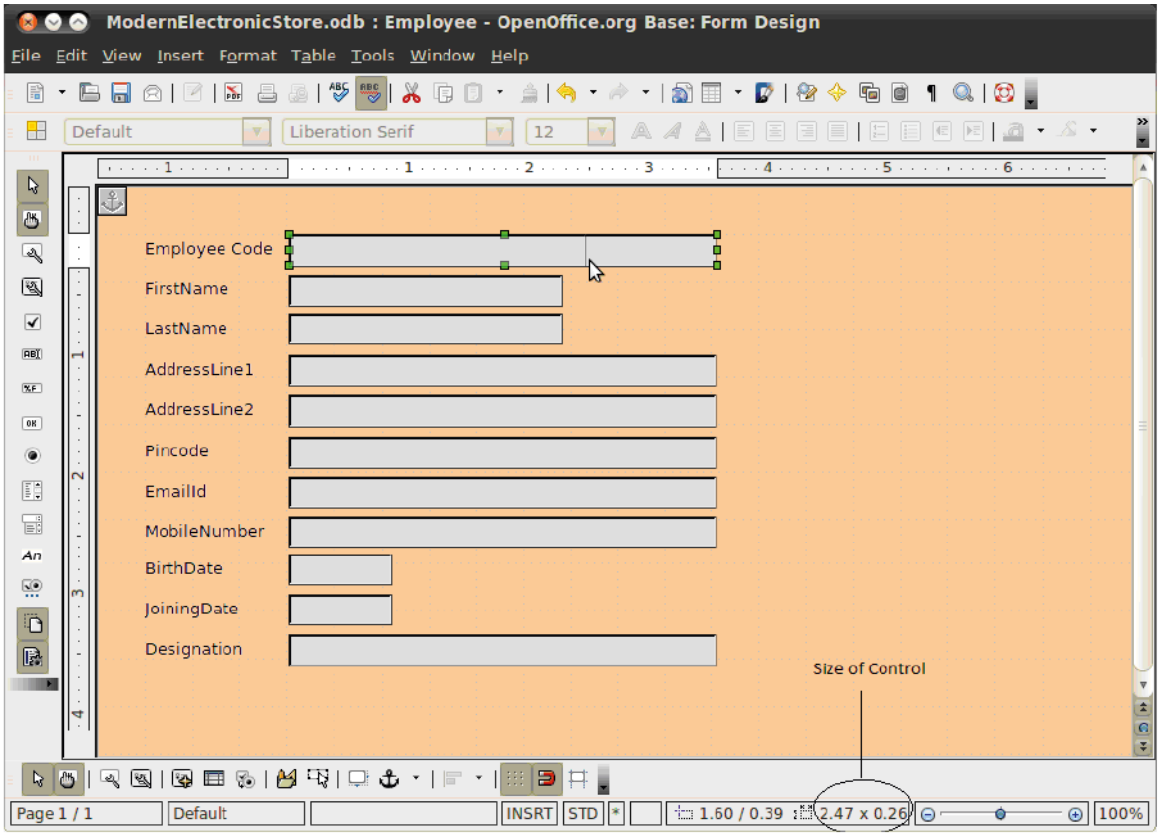


Figure 12.12 : Changing Size of Control

- You can double click the position and size area to open the *Position and Size* dialog box as shown in figure 12.13.

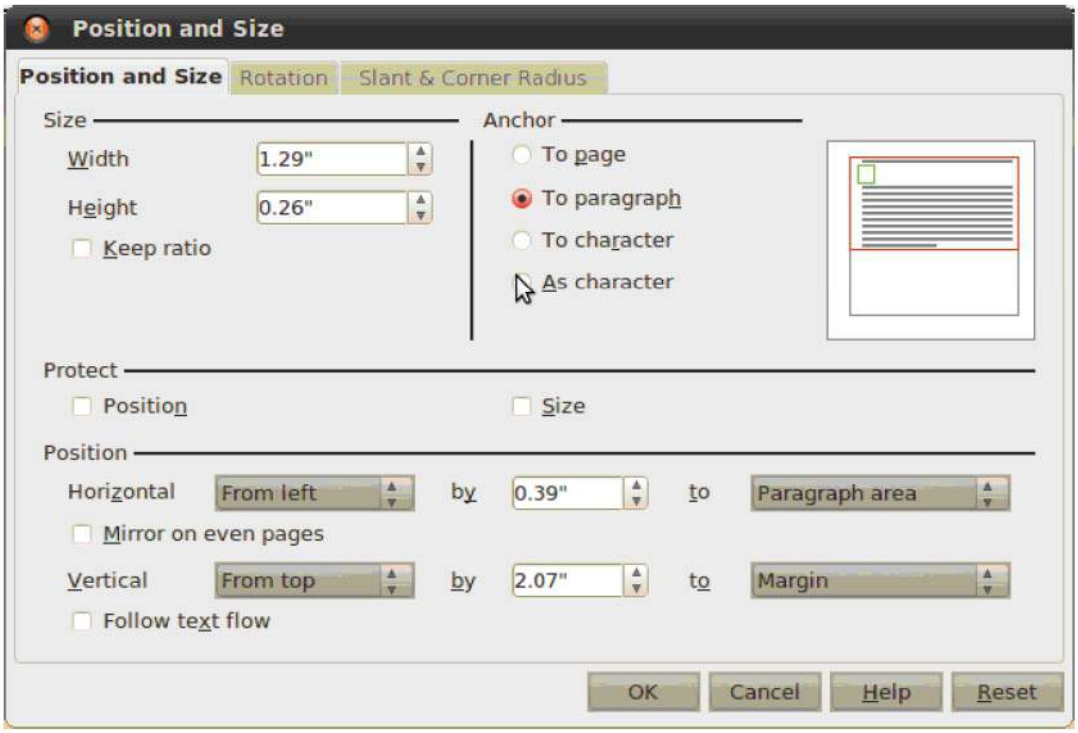


Figure 12.13 : Position and Size

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

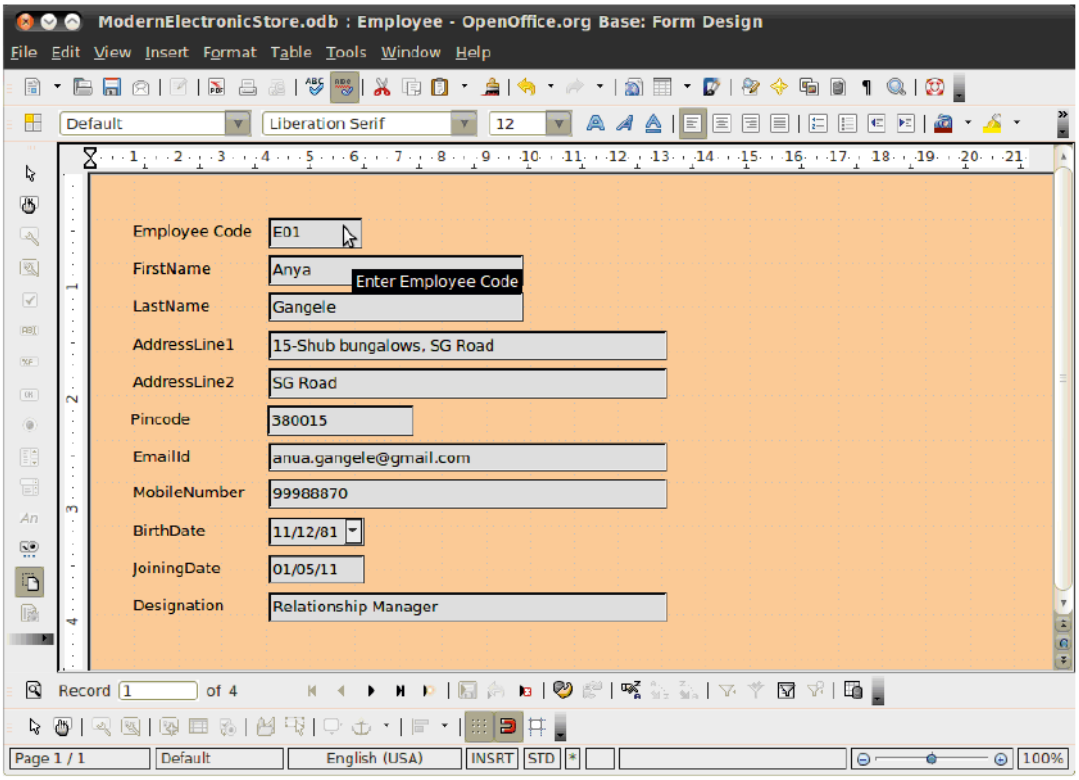


Figure 12.14 : Help Text

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.

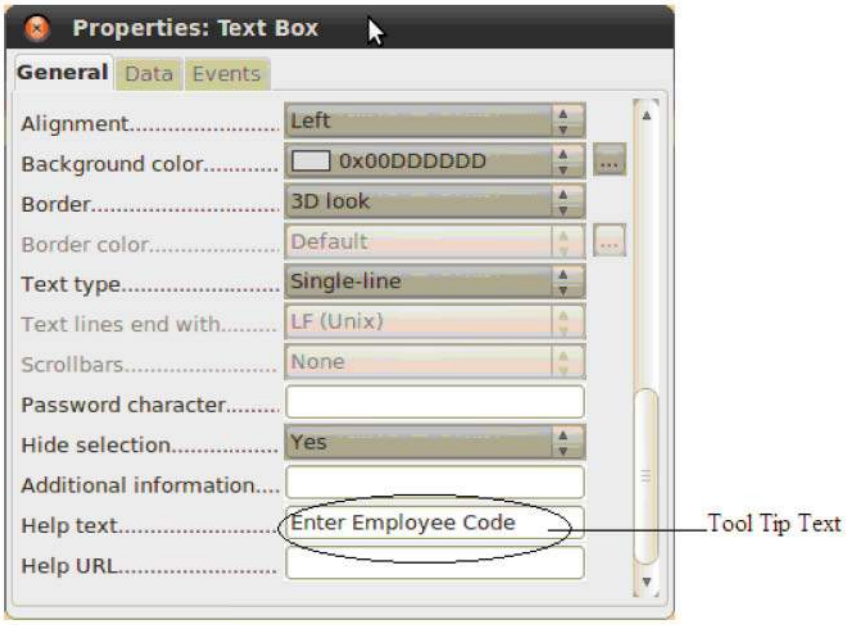


Figure 12.15 : Setting Help text

Change the Date field

Perform the following steps to change the date field to calendar control as seen in figure 12.16.

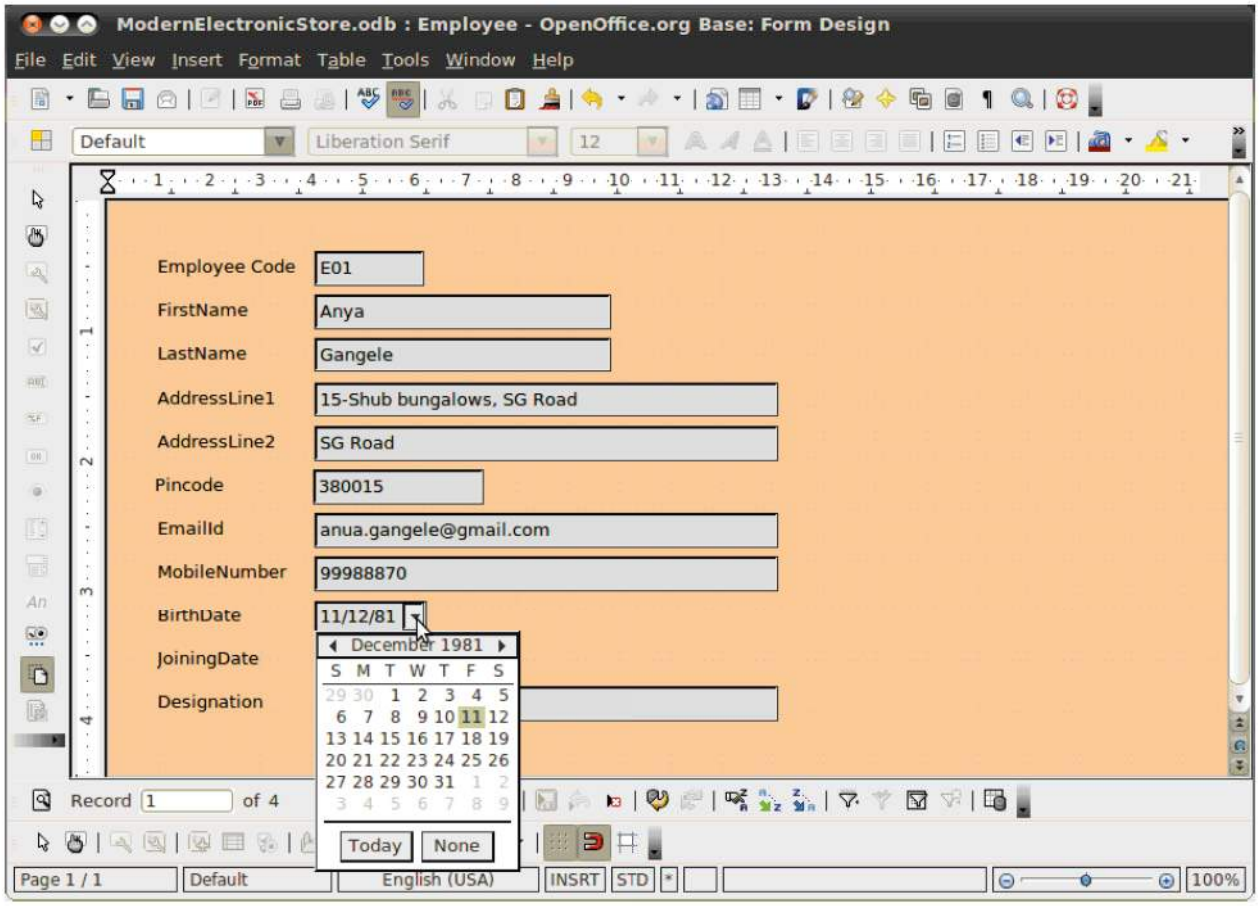


Figure 12.16 : Changing Date field

- 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.

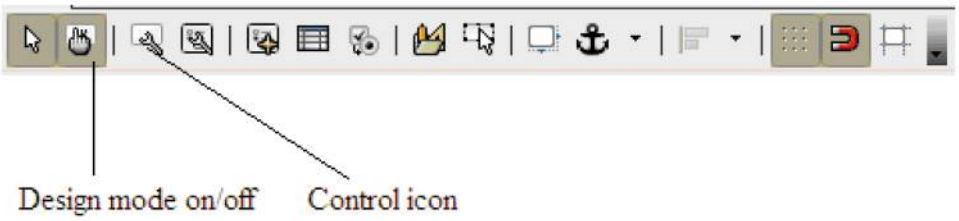


Figure 12.17 : Form Controls toolbar

- 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.

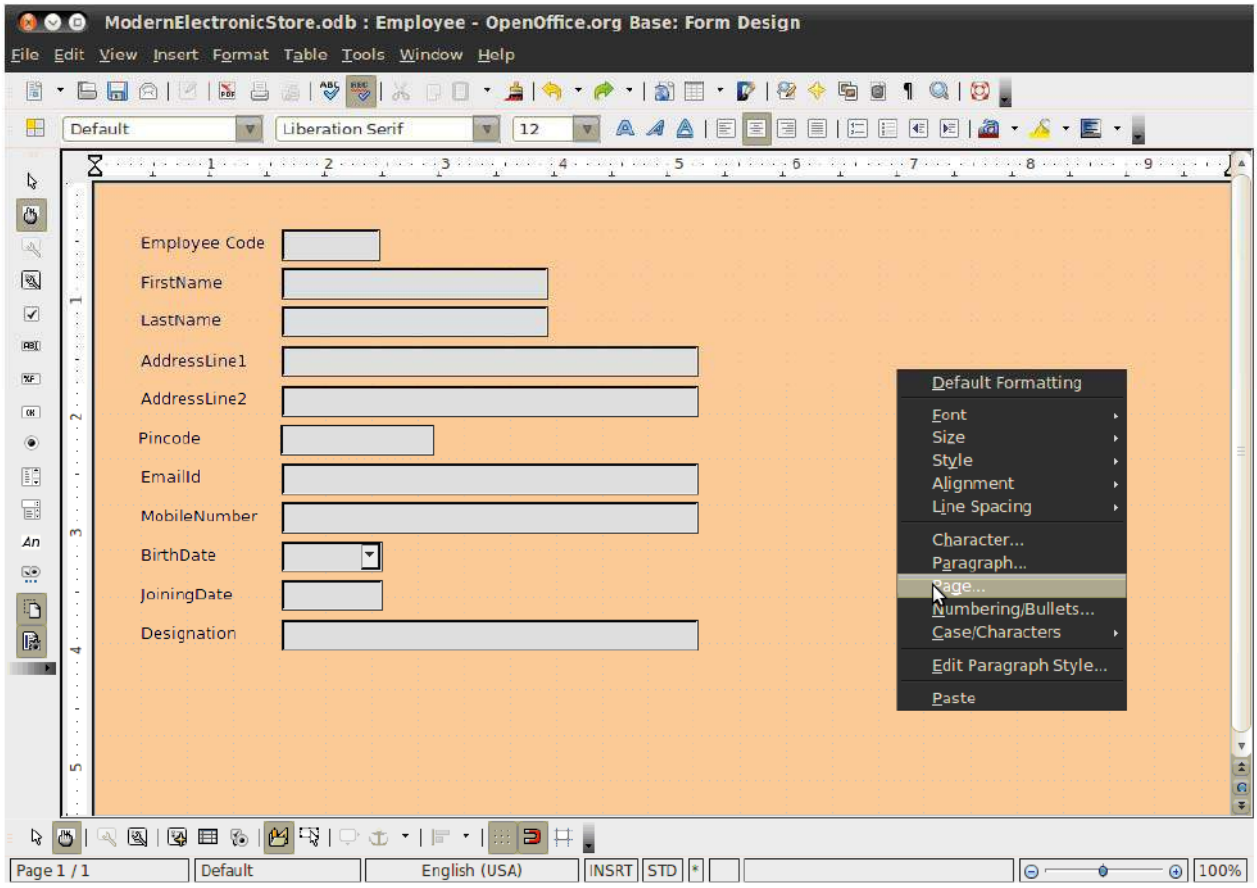


Figure 12.18 : Form Sub-menu

- Choose *Page...* option. A *Page Style* dialog box as shown in figure 12.19 would appear.

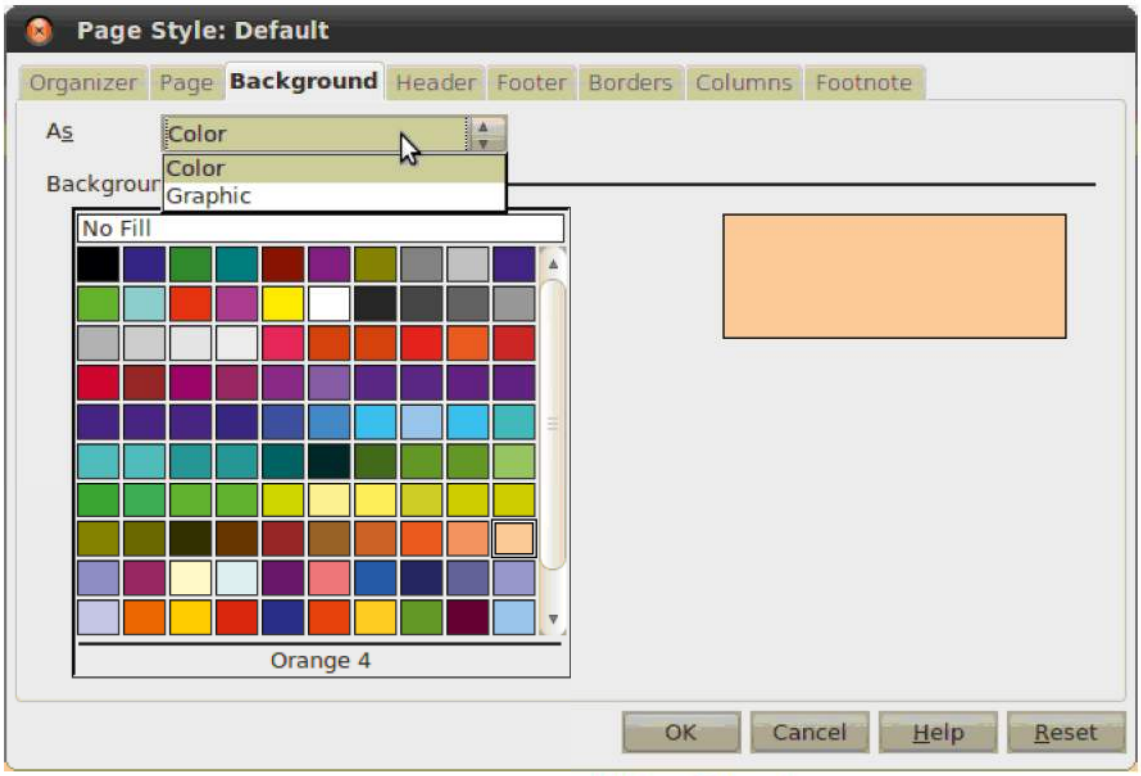


Figure 12.19 : Page Style dialog box

- 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.

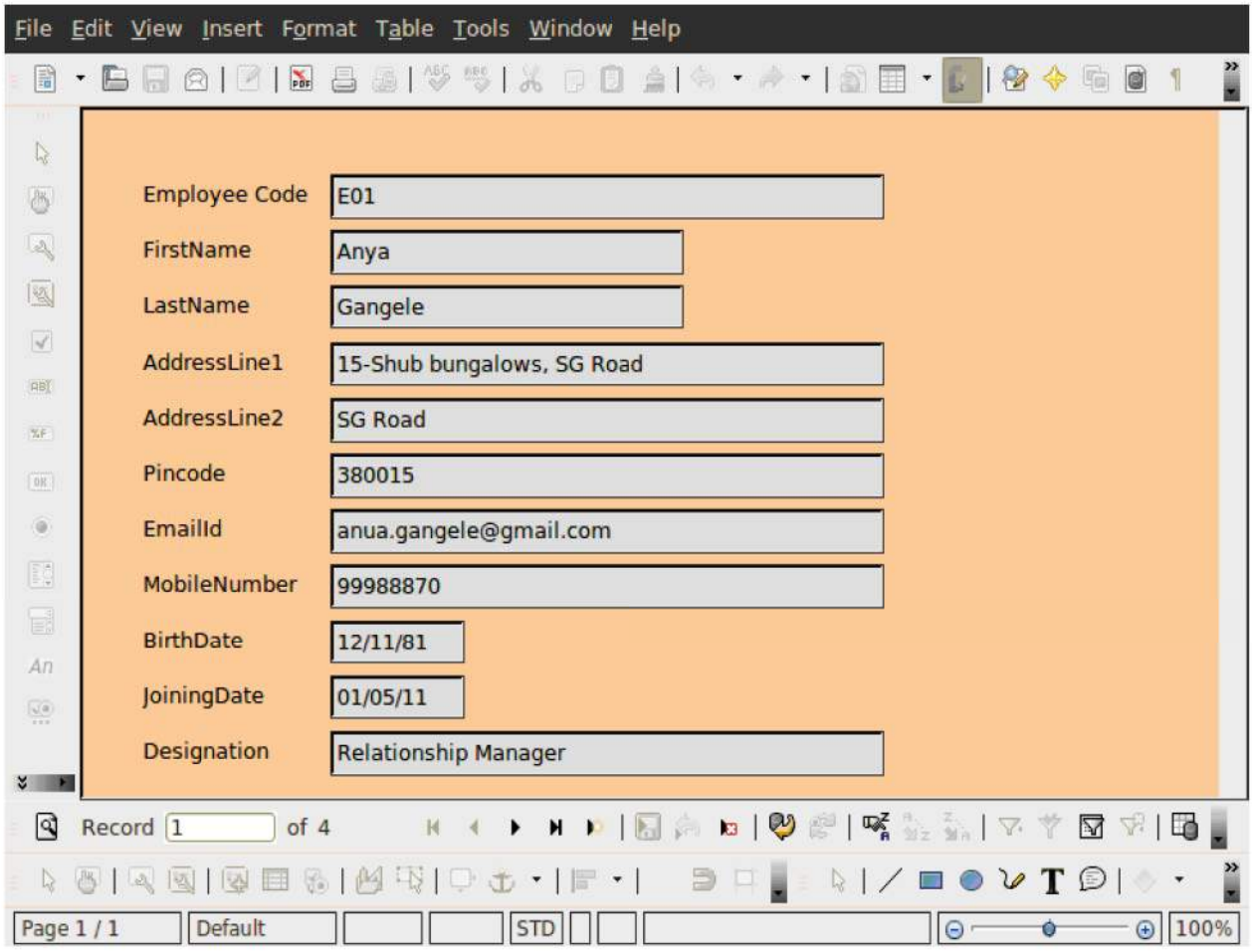


Figure 12.20 : Employee form in Data Entry mode

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.

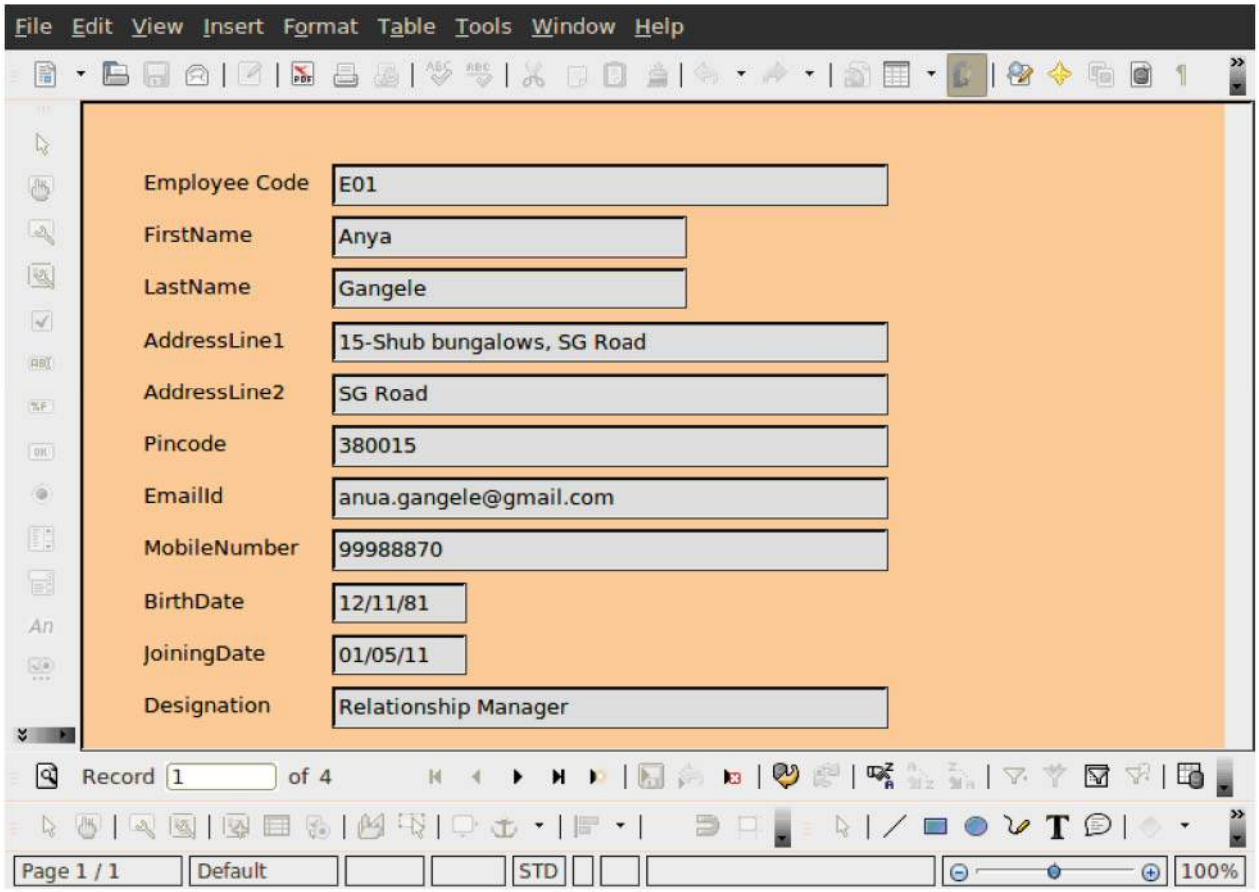


Figure 12.21 : Find Record option

A *Record Search* dialog box as seen in figure 12.22 will appear on the screen.

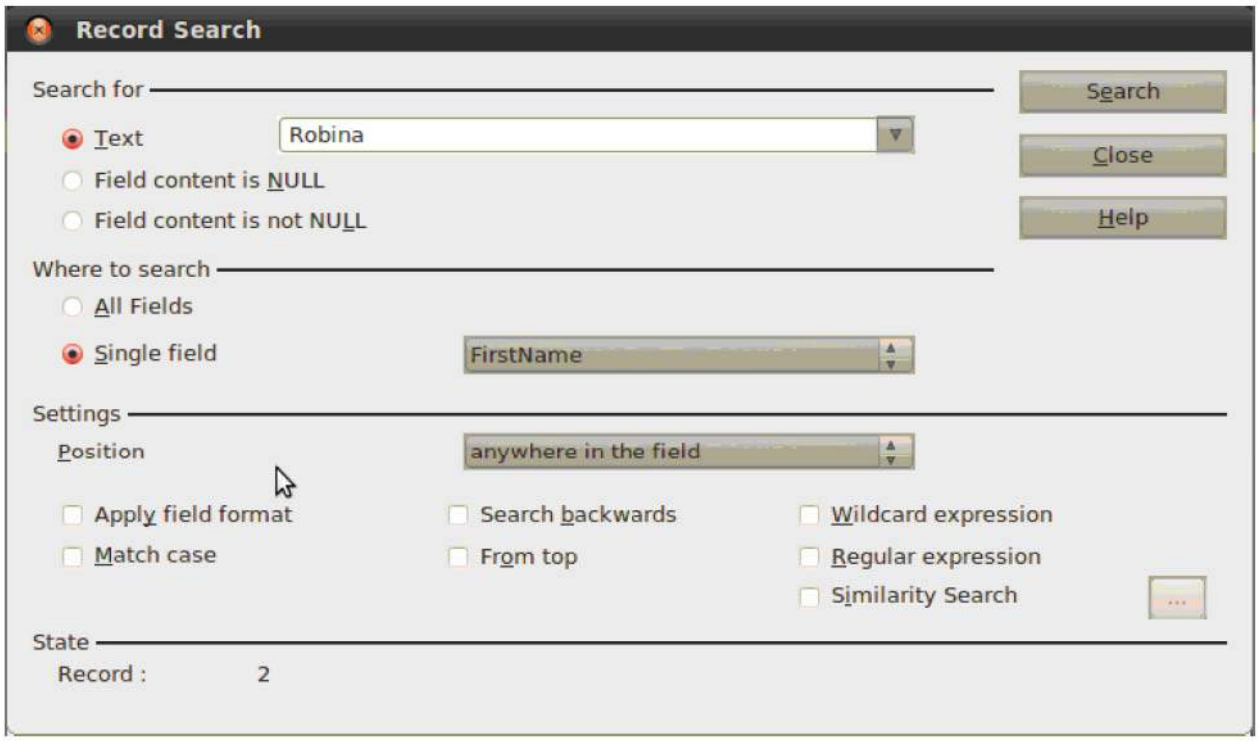


Figure 12.22 : Record Search dialog box

- 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.

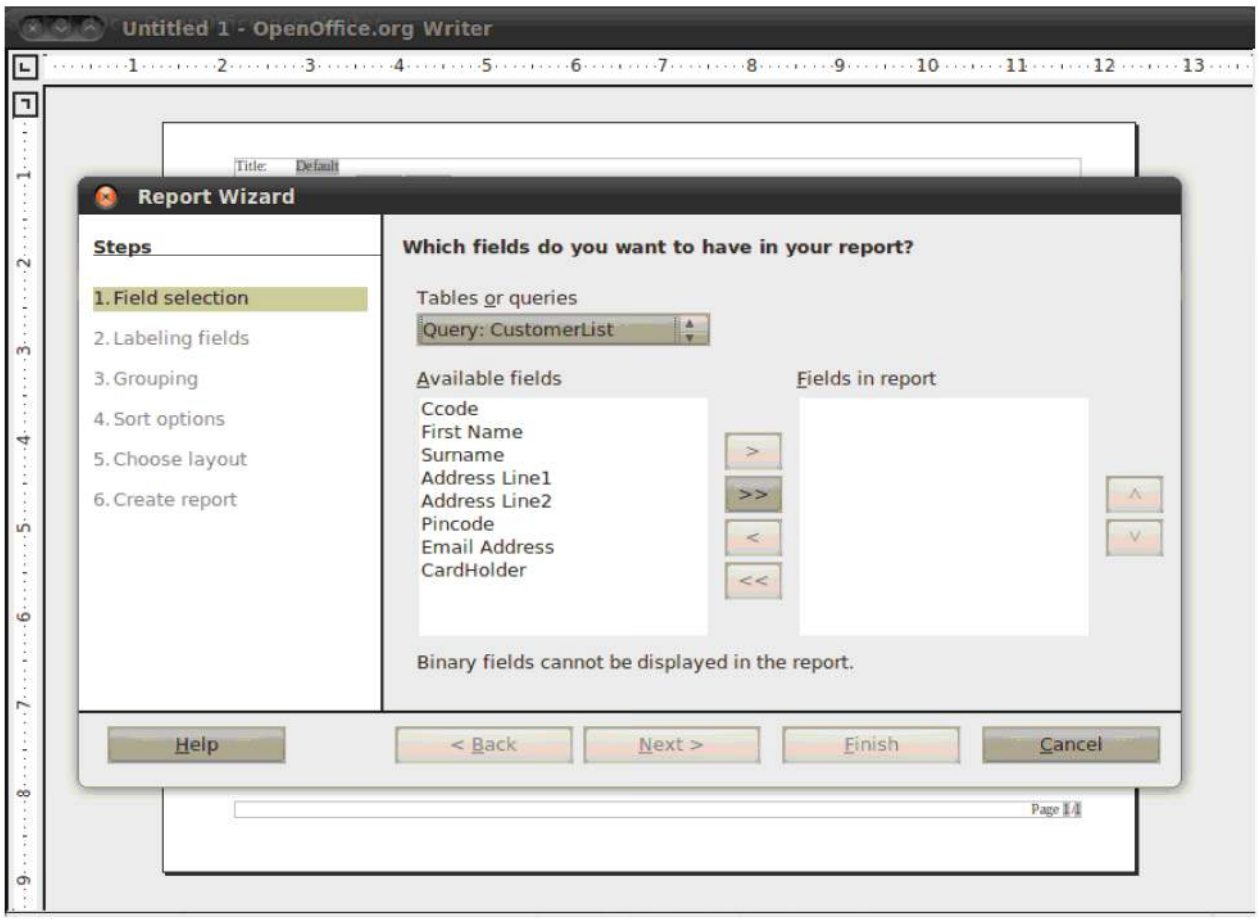


Figure 12.23 : Step 1 of Report wizard

- In second step give appropriate labels to each field *as* per your requirement. Sample labels are as shown in figure 12.24.

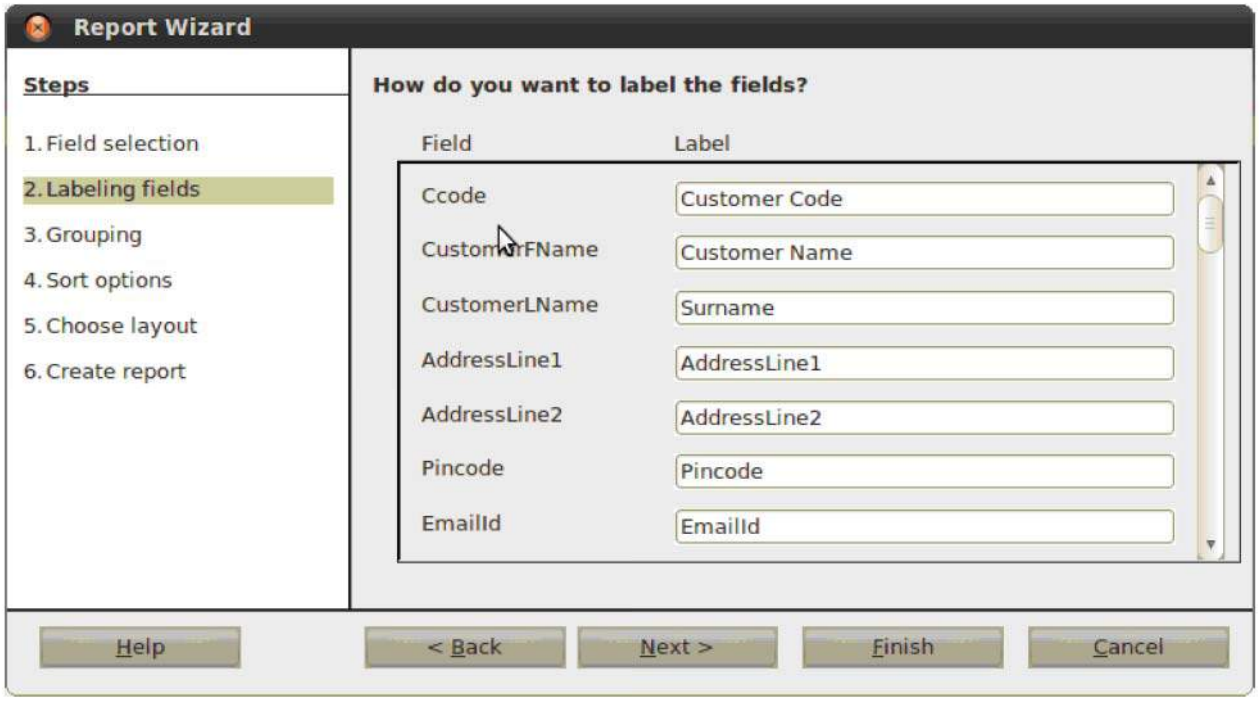


Figure 12.24 : Step 2 of Report Wizard: Labeling fields

- 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.

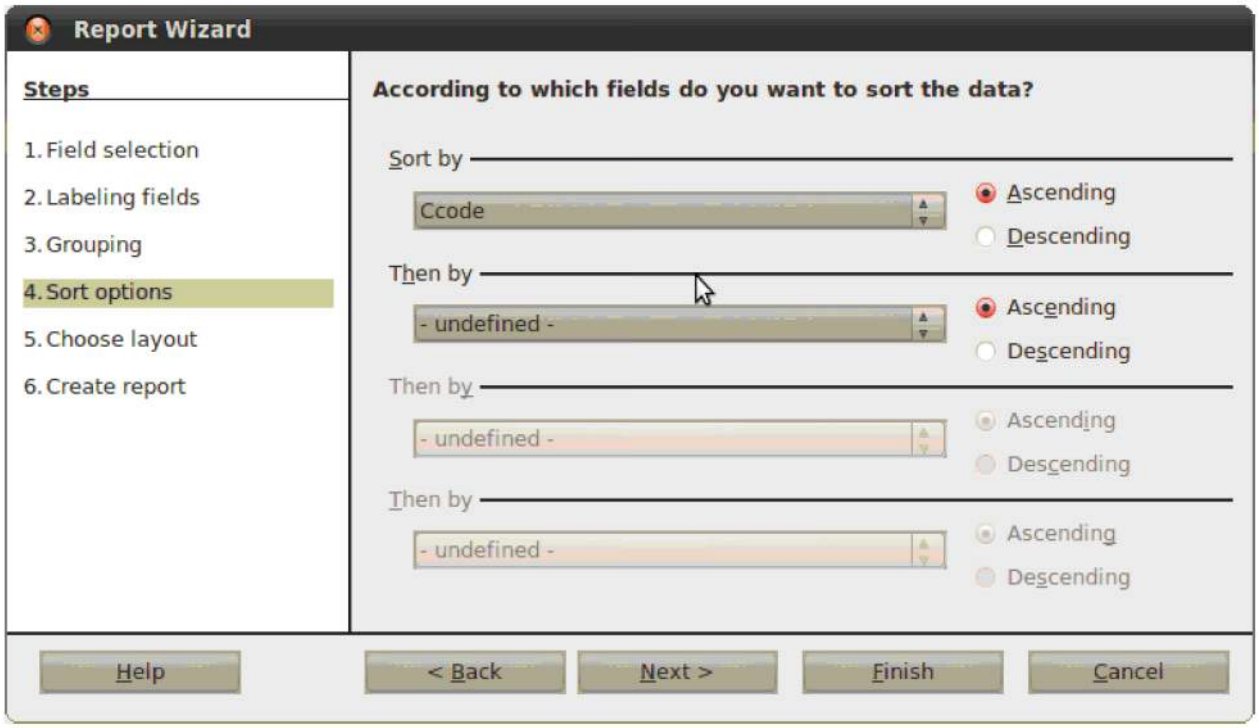


Figure 12.25 : Step 4 of Report Wizard: Mention Sort options

- 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.

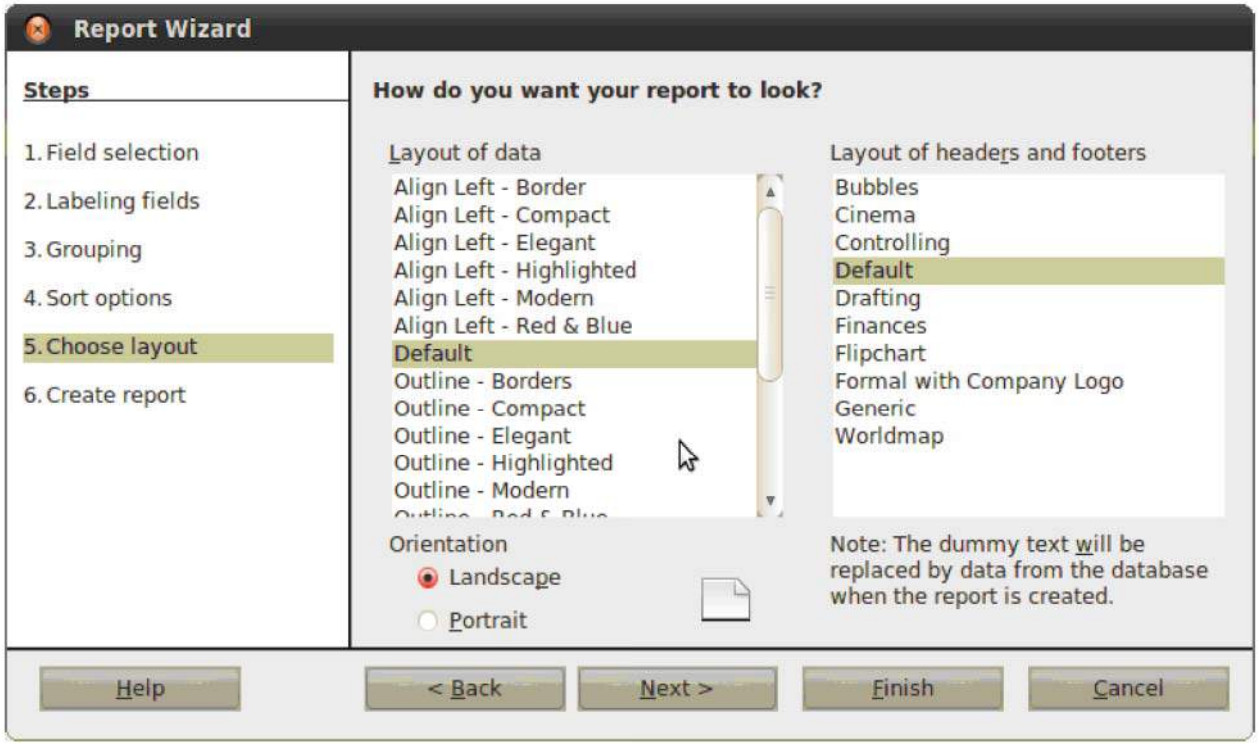


Figure 12.26 : Step 5 of Report Wizard: Choose Layout

- 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.

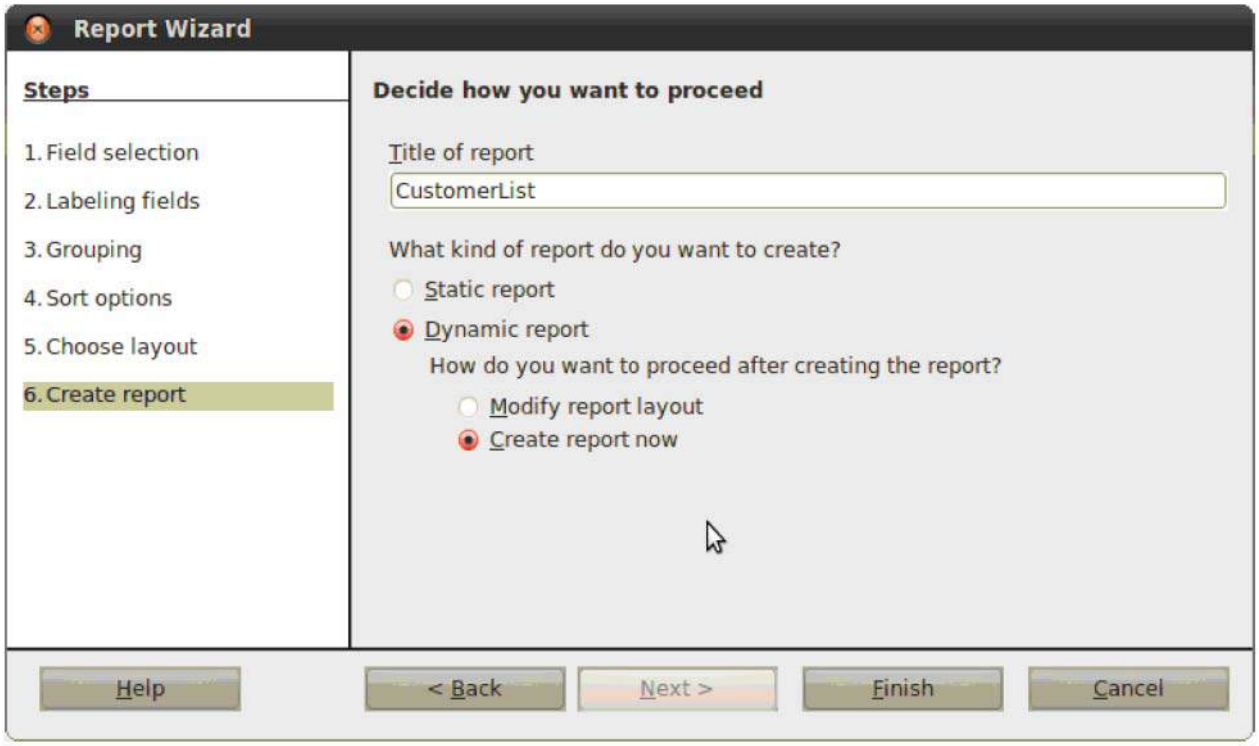
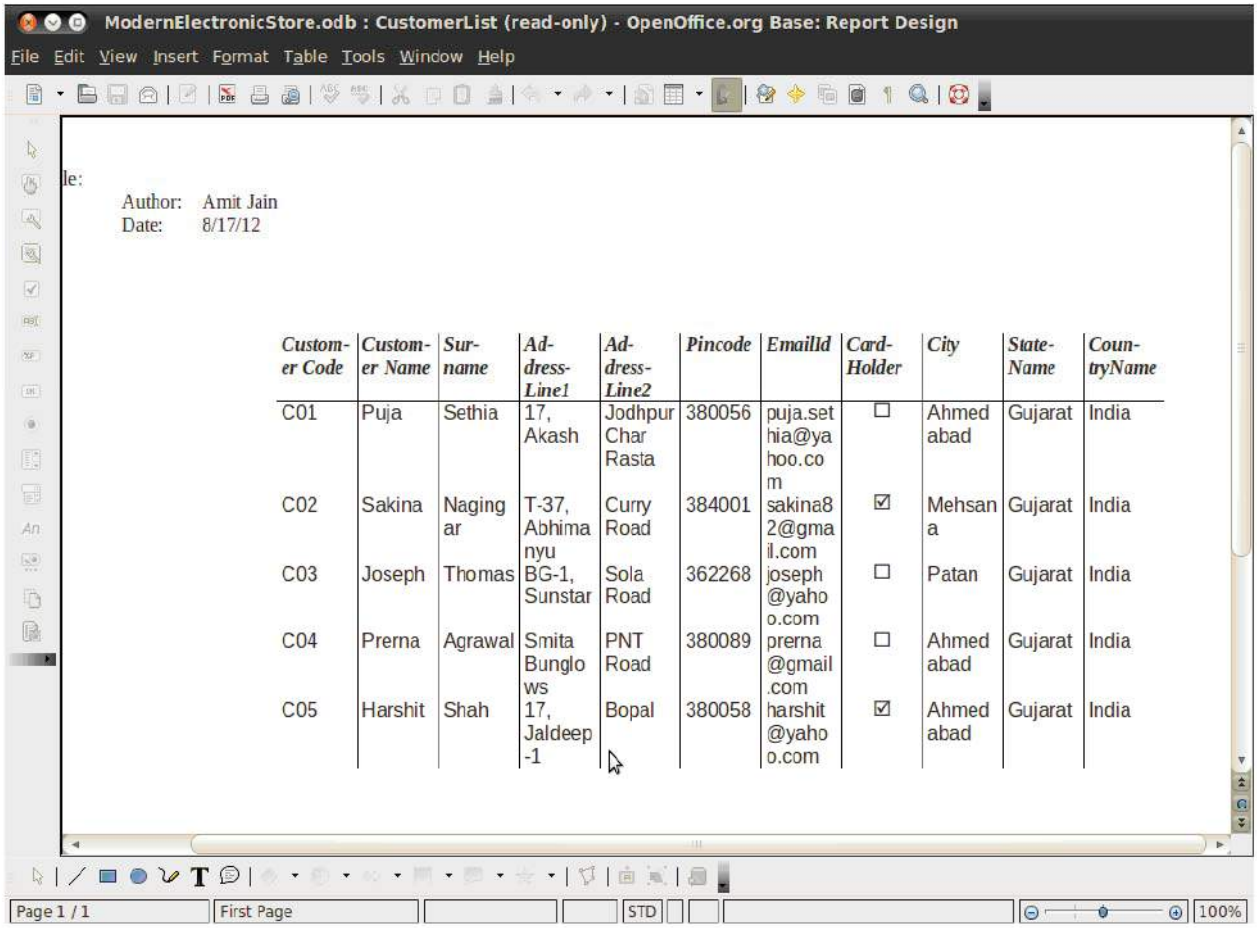


Figure 12.27 : Step 5 of Report creation: Mention Title of report

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.



Customer Code	Customer Name	Surname	Address-Line1	Address-Line2	Pincode	EmailId	Card-Holder	City	State-Name	CountryName
C01	Puja	Sethia	17, Akash	Jodhpur Char Rasta	380056	puja.sethia@yahoo.com	☐	Ahmedabad	Gujarat	India
C02	Sakina	Nagingar	T-37, Abhimanyu	Curry Road	384001	sakina82@gmail.com	☑	Mehsana	Gujarat	India
C03	Joseph	Thomas	BG-1, Sunstar	Sola Road	362268	joseph@yahoo.com	☐	Patan	Gujarat	India
C04	Prerna	Agrawal	Smita Bunglows	PNT Road	380089	prerna@gmail.com	☐	Ahmedabad	Gujarat	India
C05	Harshit	Shah	17, Jaldeep-1	Bopal	380058	harshit@yahoo.com	☑	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.

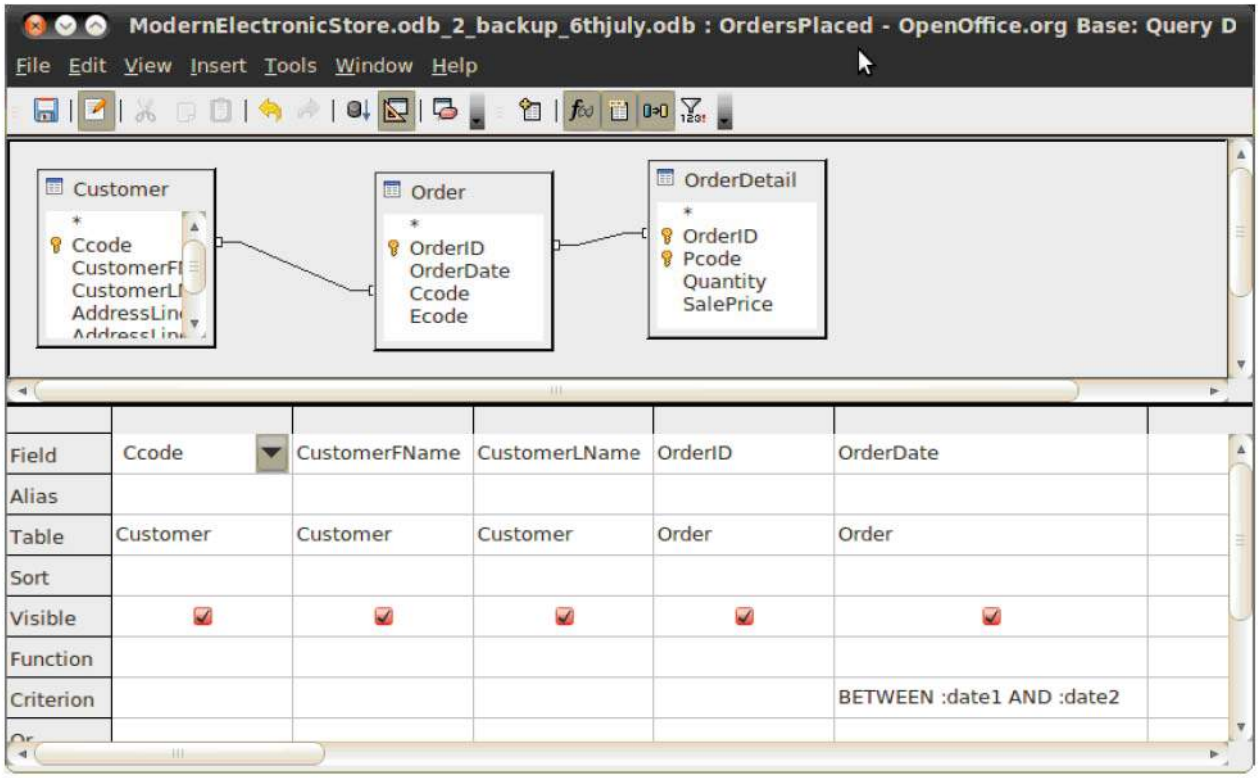


Figure 12.29 : Parameterized query

- 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.

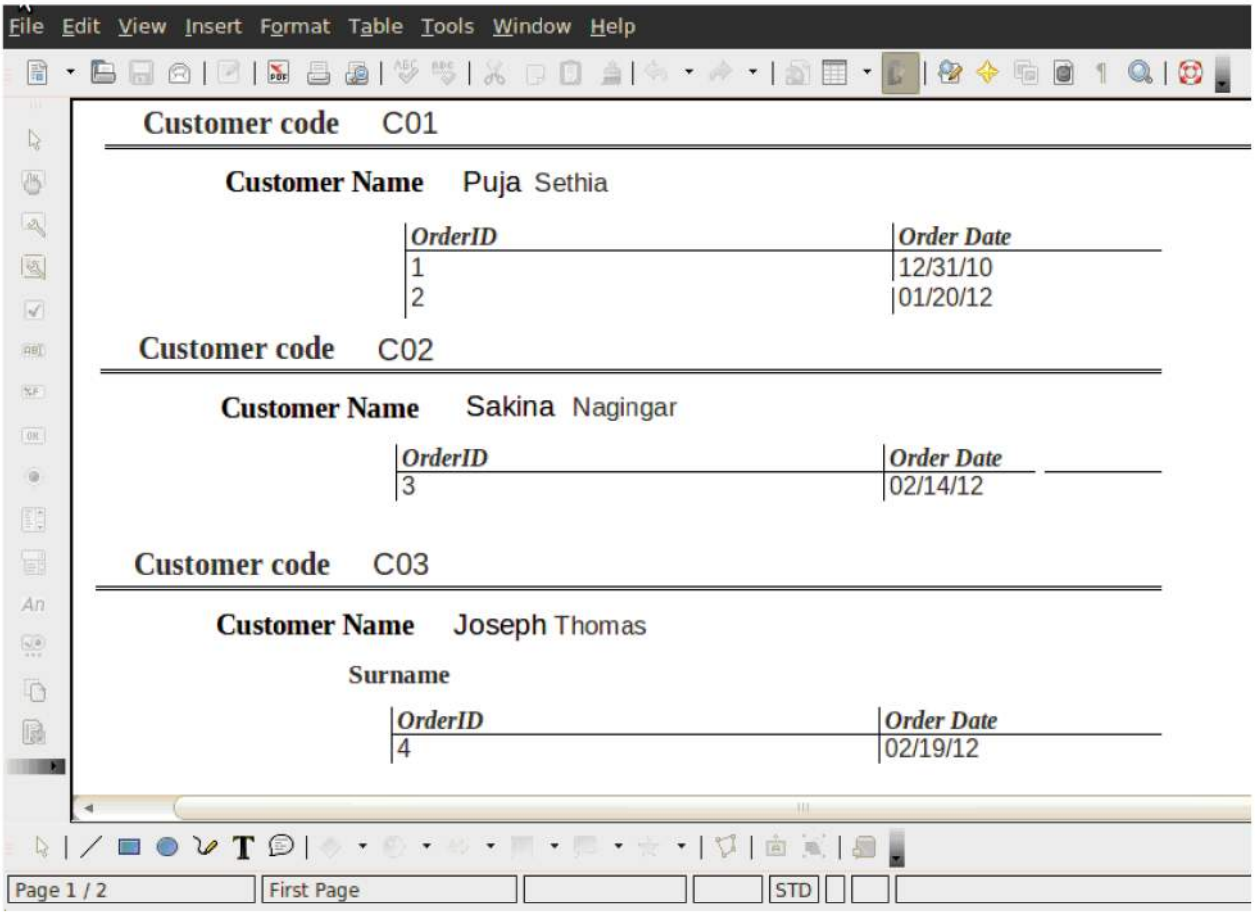


Figure 12.31 : Parameterized Report

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

1. What are Forms? Why should one design forms ?
2. What are the navigation buttons on the forms used for ?
3. List the default Layouts and Styles provided by Base while creating a Form.
4. What are reports ? Write down the usage of reports.
5. What is the difference between forms and reports ?
6. List the default Layouts and Styles provided by Base while creating a Report.

8. **Choose the most appropriate option from those given below :**

(1) For what main purpose a form is designed?

(a) To display data (b) To enter data

(c) To create tables (d) To view query result

(2) For what main purpose a report is designed?

(a) To view information in professional manner

(b) To enter data

(c) To create tables

(d) To get statistics of usage of records

(3) Form wizard allows creating data entry forms for which of the following?

(a) Report (b) Table

(c) Query (d) Both b and c

(4) Which option is used to select a field?

(a) CTRL + Click (b) Click

(c) Double Click (d) Right Click

(5) Help Text created by user can be generally associated with which of the following object?

(a) Table (b) Form

(c) Control Field on Form (d) Report

(6) Report wizard is used to create report from how many tables at a time?

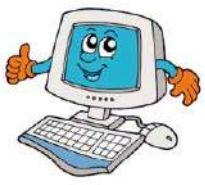
(a) One (b) Two

(c) Three (d) More than three

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.

13 Current Trends and Technologies



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 is 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.

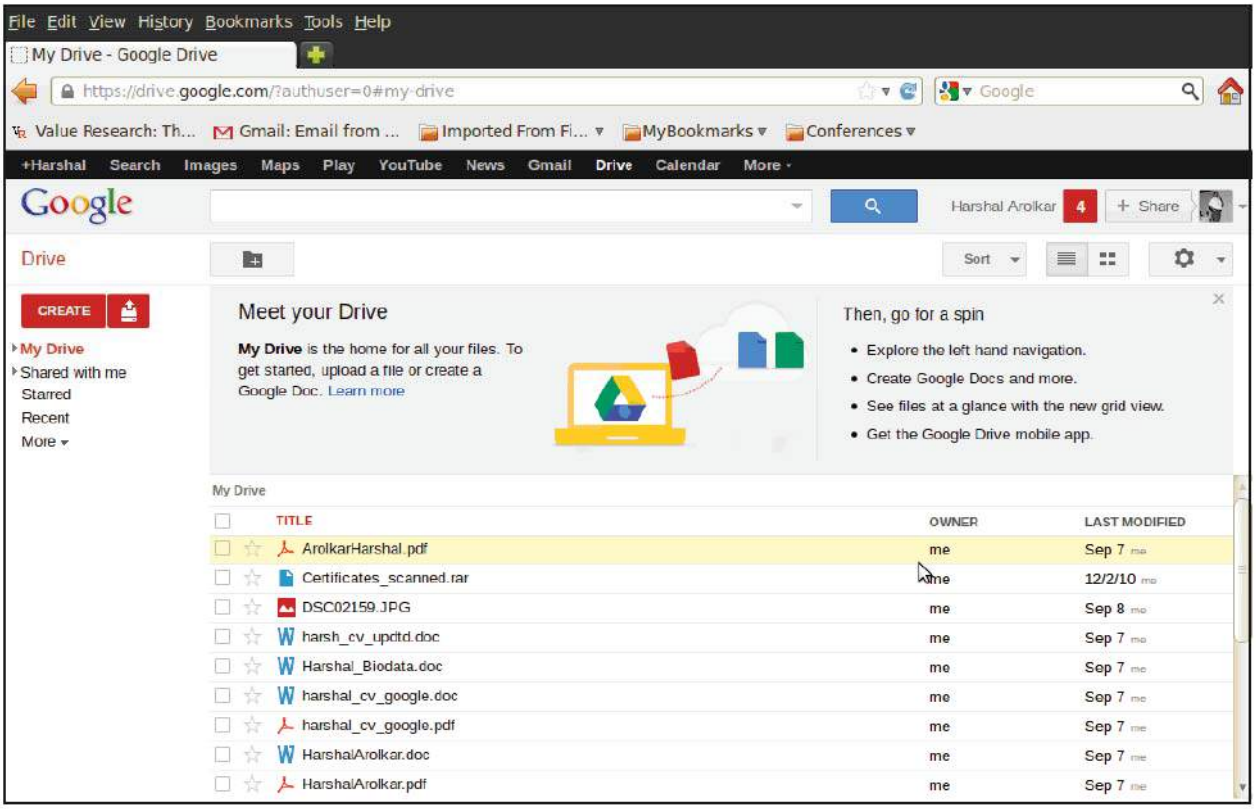


Figure 13.4 : GUI of Google drive

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.

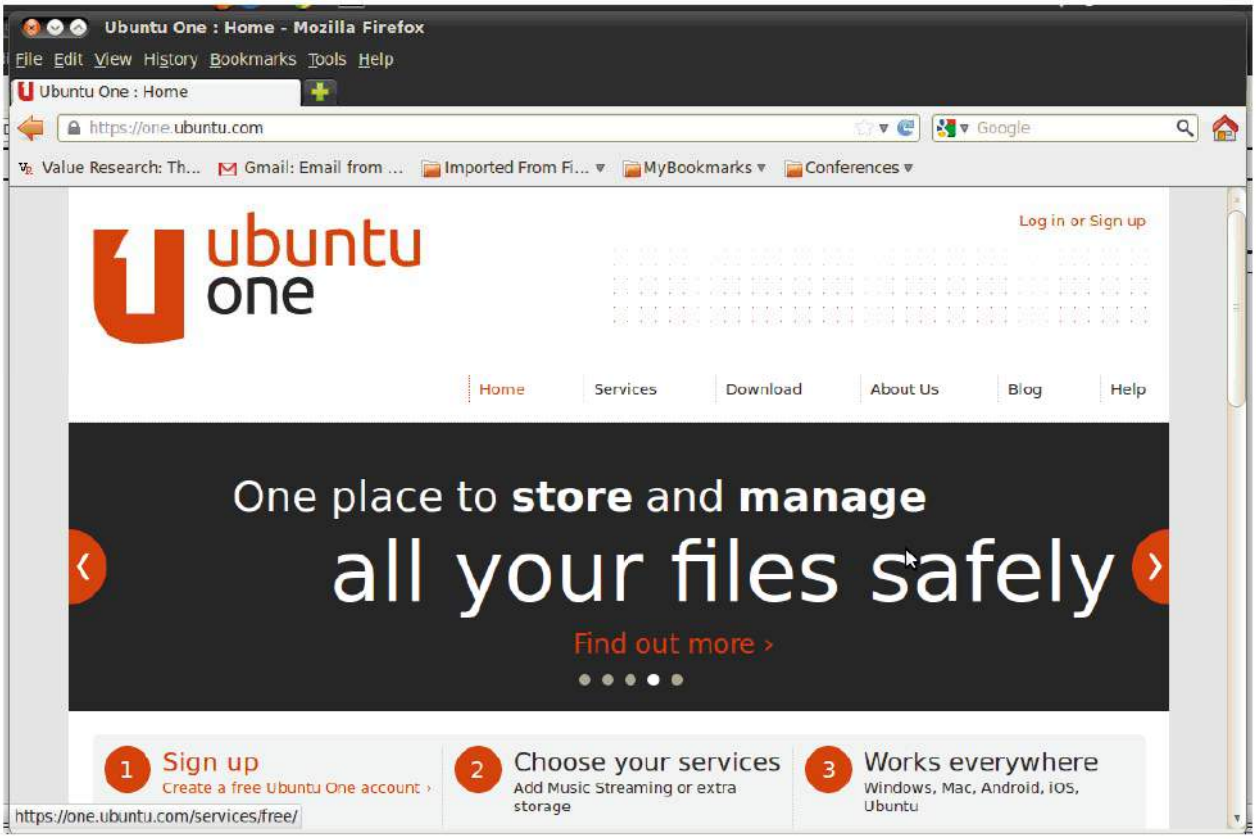


Figure 13.5 : Home page of Ubuntu One

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/.

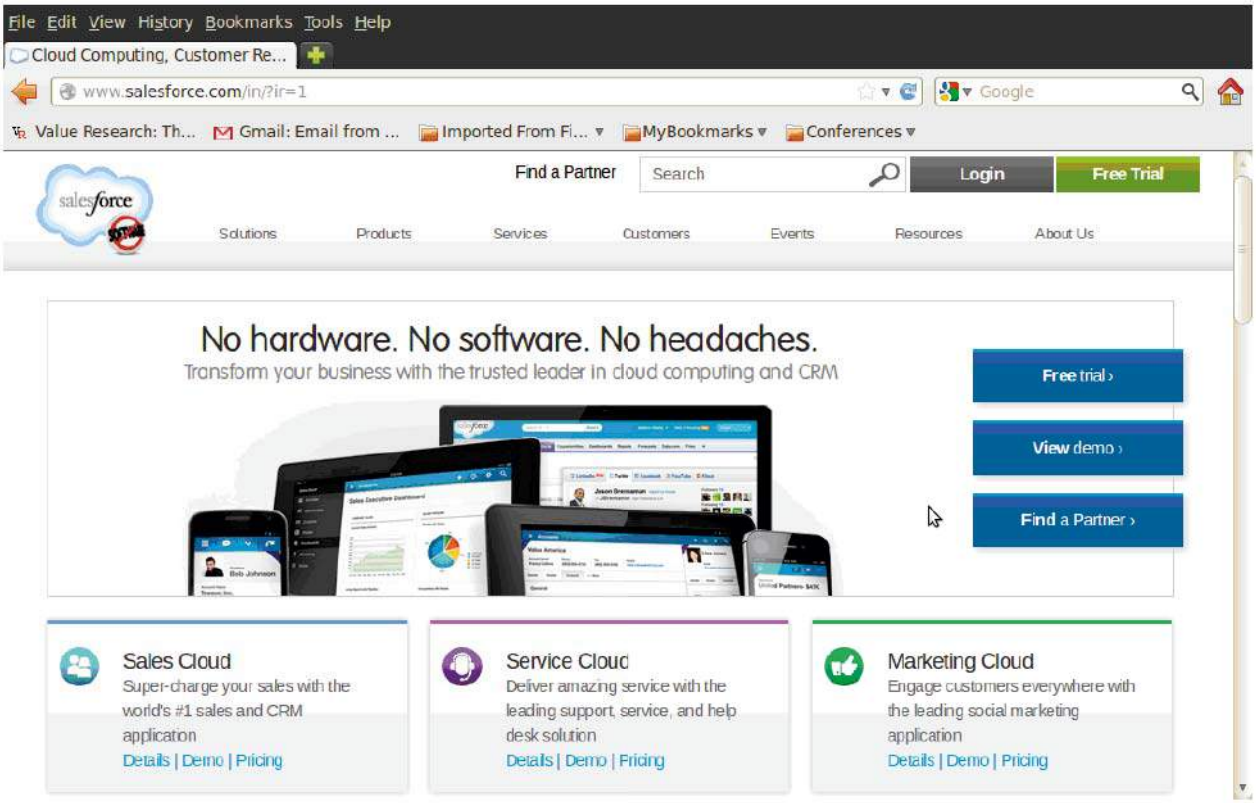


Figure 13.6 : Homepage of http://www.salesforce.com/in/

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.

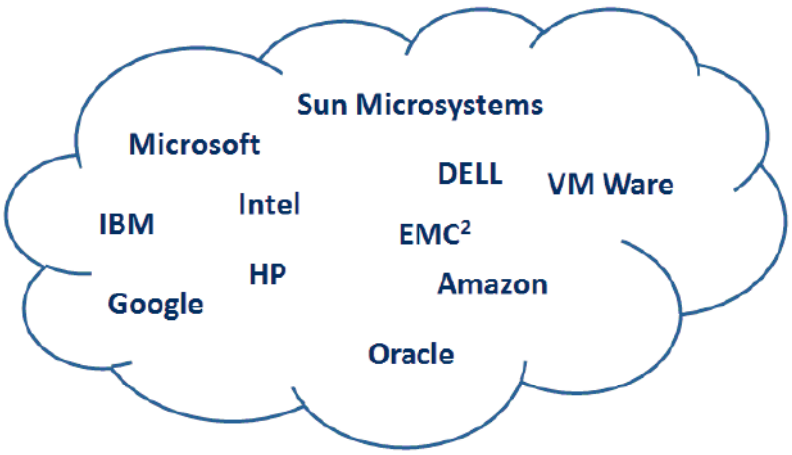


Figure 13.7 : Companies providing cloud services

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.

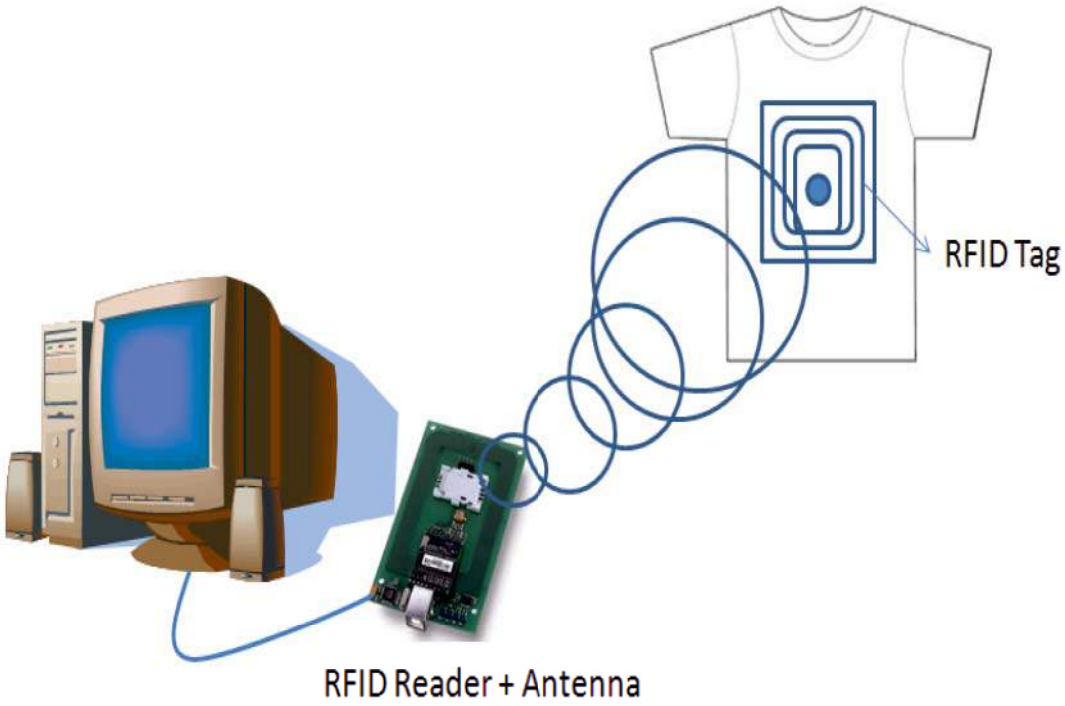


Figure 13.8 : Working of RFID system

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 wary 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 is 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 sent 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.

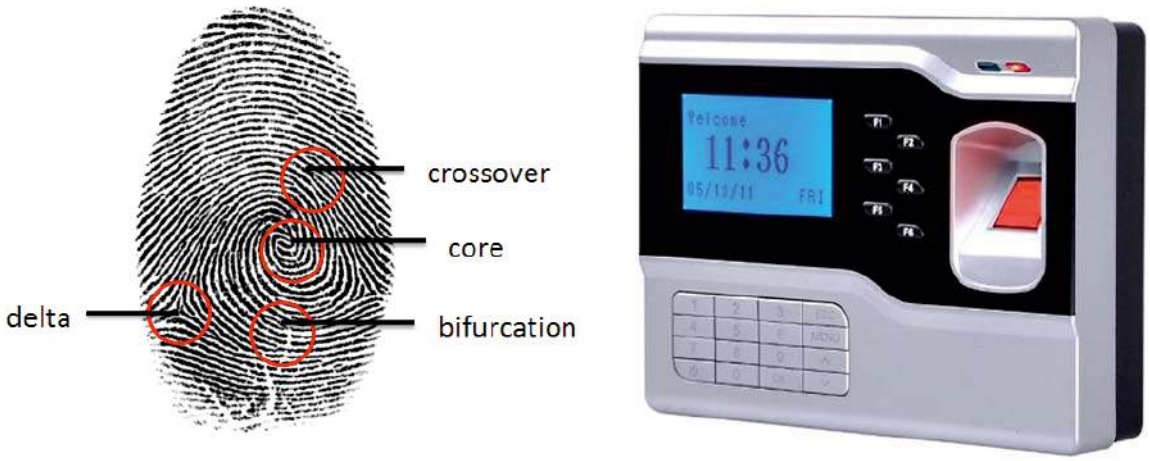


Figure 13.9 : Features of fingerprint and Fingerprint reader

Table 13.1 lists some of the government, forensic and commercial applications of finger print recognition.

Government	Forensic	Commercial
• Identification cards like AADHAR, Ration Card, BPL Cards. • Driver's License • Welfare Disbursement • Passport Control, etc.	• Criminal Investigation • Corpse Identification • Criminal Record Keeping • Terrorist Identification, etc.	• 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.

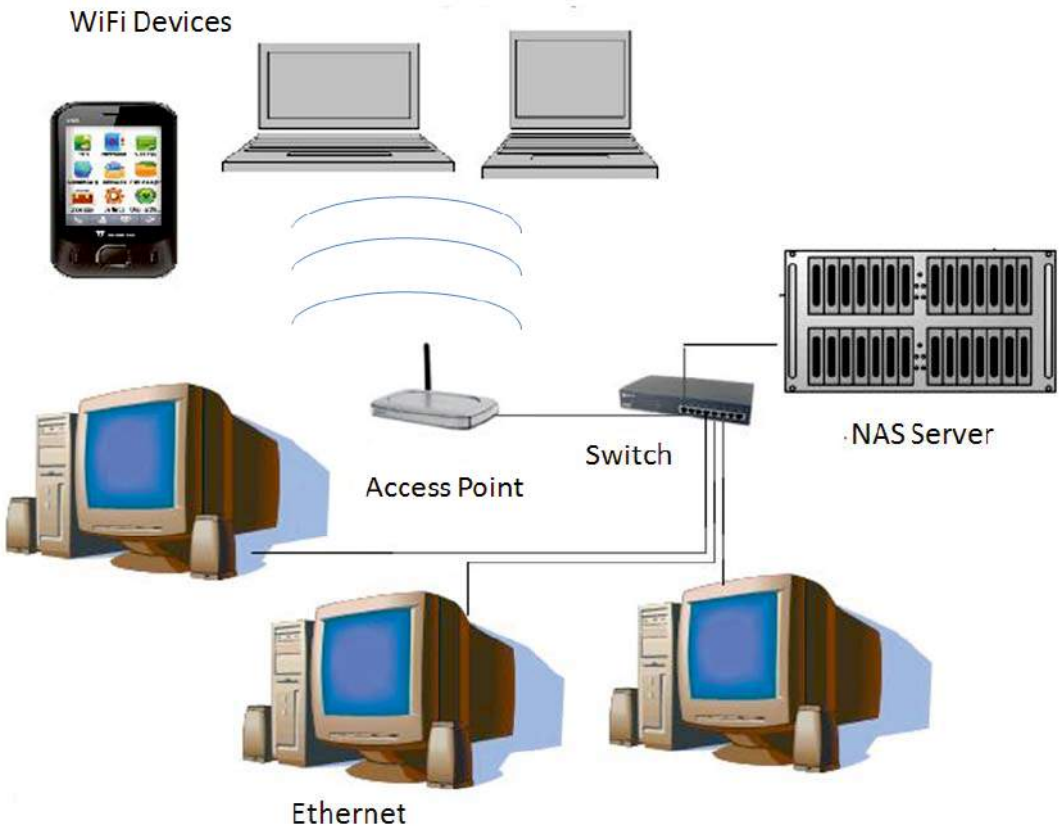


Figure 13.10 : NAS Architecture

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.

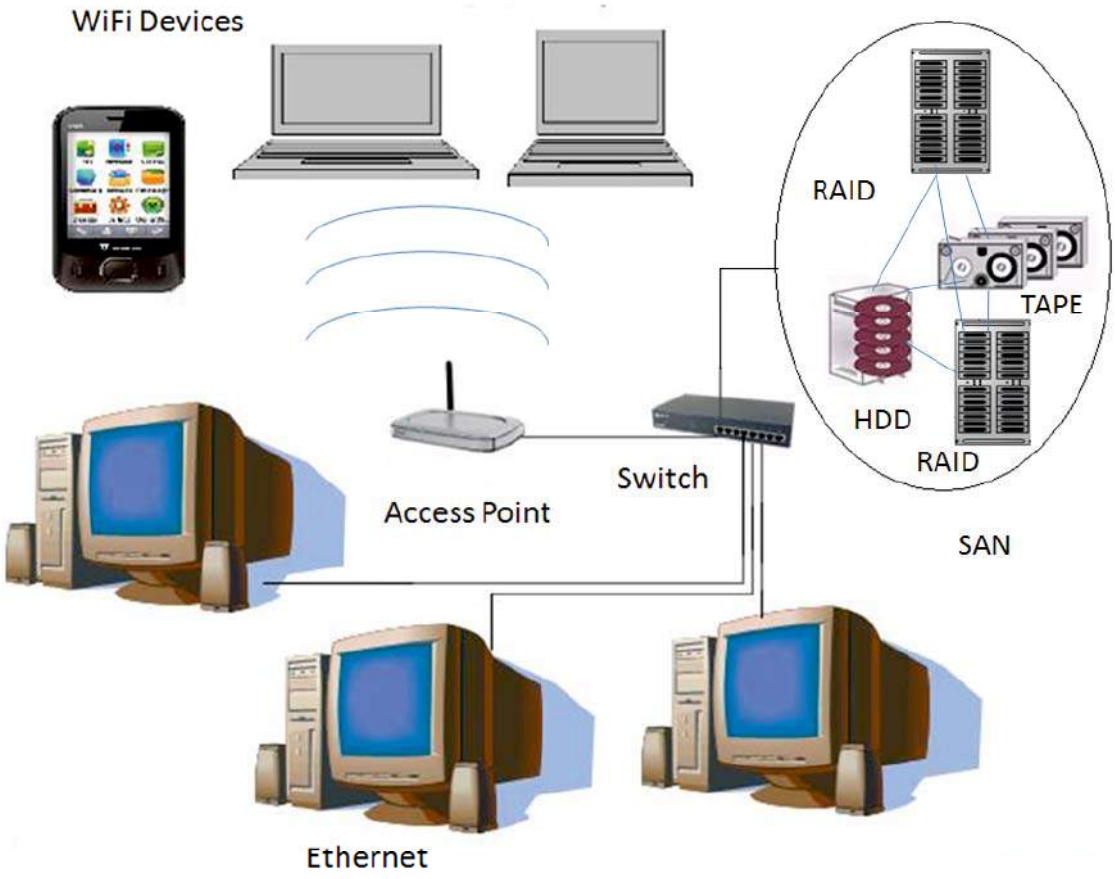


Figure 13.11 : SAN architecture

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.

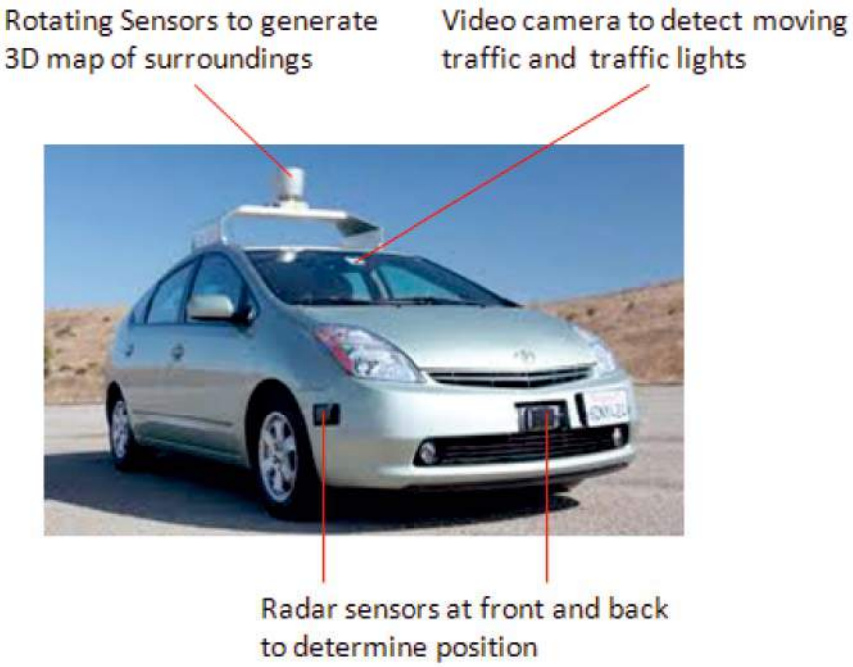


Figure 13.12 : Google driverless car

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 affect 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

(6) Which of the following represents the maximum storage capacity of the microchip used in RFID?

(a) 10 bytes
(b) 2000 bytes
(c) 1000 bytes
(d) 500 bytes

(7) How many satellites does the GPS receiver require to calculate only latitude and longitude?

(a) One
(b) Two
(c) Three
(d) Four

(8) Which of the following refers to a storage device that is connected directly to a network?

(a) network attached storage
(b) storage area network
(c) direct attached storage
(d) RAID

(9) Which of the following refers to a network of storage devices that can be accessed over a network?

(a) direct attached storage
(b) RAID
(c) network attached storage
(d) storage area network

(10) Which of the following terms refer to the study that finds how effectively a machine can work with the use of sensors and actuators in dynamic environments?

(a) Kinetics
(b) Genetics
(c) Robotics
(d) Biometric

(11) Which of the following refers to a program launched in 1992 as a step towards green computing?

(a) STAR ENERGY
(b) ENERGY STAR
(c) ENERGY SAVE
(d) SAVE ENERGY

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