COMPUTERISED ACCOUNTING

PLUS TWO

(FOSS Tools for Accounting)

PART I - DRAFT Version 1.0

GENERAL EDUCATION DEPARTMENT
GOVERNMENT OF KERALA

Prepared by :

IT@School Project Poojappura, Thiruvananthapuram-12,
Website : www.itschool.gov.in, email : contact@itschool.gov.in
Phone : 0471-2529800, Fax: 0471-2529810

For any queries, please contact the above Office.

PREFACE

Kerala has become the World's first State with the largest deployment of Free and Open Source Software in Education Sector, as part of the ICT enabled education being implemented by IT@School in High School section in the State. In August 2008, Government had instructed all institutions under General Education Department to strictly use Free Software alone in all future teaching and training activities and a Government Order in this regard is in effect. Similarly, the Information Technology Department has also released a Circular in July 2016, directing all departments to use FOSS. Adhering to these directions, the Computer Science and Humanities sections in the Higher Secondary had shifted to Free Software. But at the same time, in the Commerce section in Higher Secondary (Computerised Accounting System), there were using MS Excel, MS Access and Tally package which are proprietary in nature.

The 48th meeting of the State Curriculum Steering Committee held on 15.02.2017, has decided to adopt Free & Open Source Software (FOSS) for entire activities in Higher Secondary Sections. The shift to Free Software in Higher Secondary is planned to be implemented without making any changes in the critical structure of the present syllabus, but by incorporating changes in the software which is being used. Accordingly, the revised textbook would include Free Software applications such as LibreOffice Calc, LibreOffice Base and GNUKhata which would replace the proprietary applications. IT@School would implement mechanisms for easy classroom transactions of chapters including customisation of applications, teacher training, video tutorials etc. This handbook is an attempt to help the higher secondary teachers and students in easy classroom transactions.

K.ANVAR SADATH
Executive Director
IT@School Project

TERM 1

CONTENTS

2.SPREADSHEET

4.Graphs and Charts for Business Data

5. Accounting Software Package - GNUKhata

6. Database Management System for Accounting

Chapter 1

SPREADSHEET

Spreadsheet application is a computer program that allows us to record, calculate and compare numerical or financial data. A *spreadsheet* is a configuration of rows and columns. Rows are horizontal vectors while columns are vertical vectors. A spreadsheet is also known as a worksheet. LibreOffice Calc, MS Office Excel, Open Office Spreadsheet, etc. are examples of Spreadsheet software.

BASIC CONCEPTS OF SPREADSHEET

A file in spreadsheet is known as a "Workbook". A workbook is a collection of a number of "Worksheets". At a time, only one worksheet can be made as active worksheet and that worksheet is available to a user for carrying out operations. Worksheet names will be shown in the "Sheet Tab" at the bottom left of the window. Additional sheets can be added and its name can be changed, if required.

Features of Spreadsheet

Major features of Spreadsheets are:

- It provides functions to create formulae to perform calculations on data.
- Data can be presented in charts or diagram for easy analysis (eg. Bar diagram, pie chart etc.)
- Can be used to store, arrange and filter data.
- Conditional formatting can be used for Criteria based separation of data.
- Data validation feature can be used to get the exact data.

LibreOffice Calc

LibreOffice Calc is a spreadsheet application that you can use to calculate, analyse, and manage data. It includes in LibreOffice Package, which is Free and Open Source software under the General Public Licence (GPL). LibreOffice Calc is available for a variety of platforms, including Linux, OS X, Microsoft Windows, and FreeBSD.

*Free and open-source software (**FOSS**) is the computer software that can be classified as both free software and open-source software. That is, anyone is freely licensed to use, copy, study, and change the software. The source code is openly shared so that people are encouraged to voluntarily improve the design of the software. This is in contrast to proprietary software,(like Windows) where the software is under restrictive copyright and the source code is usually hidden from the users. The benefit of using FOSS can include decreasing software costs, increasing security and stability (especially in regard to Malware), protecting privacy and giving users more control over their own hardware. Free, open-source operating systems such as Linux are widely utilized today, powering millions of servers, desktops, smartphones (eg. Android) and other devices.*

Basic Features of LibreOffice Calc

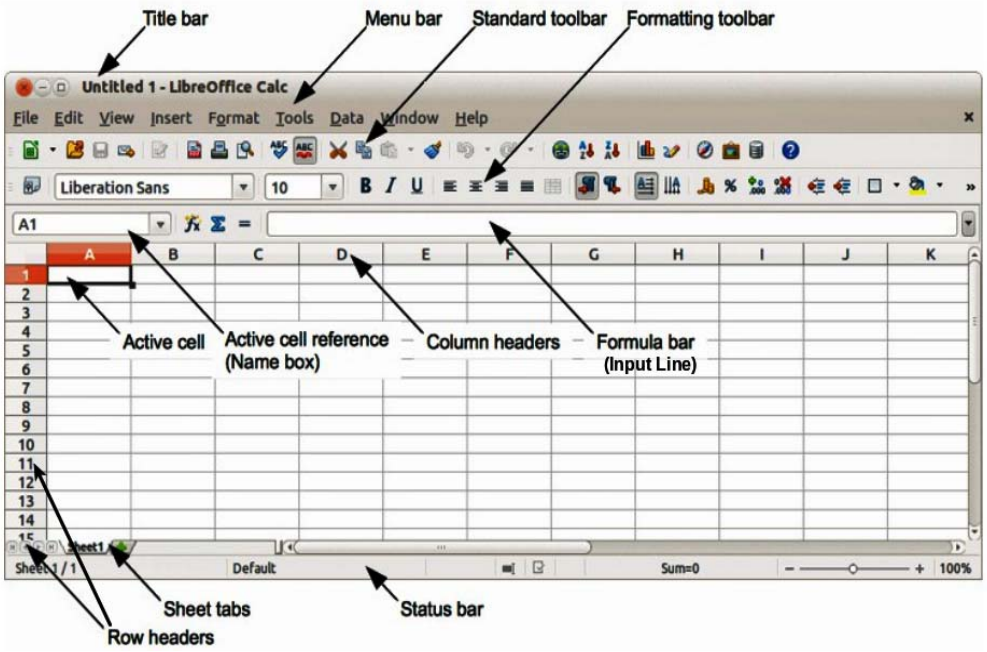


Figure 2.1: LibreOffice Calc window

Rows and Columns

The Worksheet in LibreOffice Calc contains Rows and Columns in Table format. Rows are named numerically (1,2,.....) from top to bottom while Columns are referred by alpha characters (A,B,C,D,....) from left to right (refer Figure 2.2).

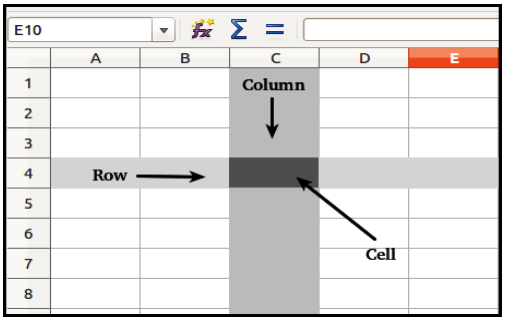


Figure 2.2 Row, Column and Cell

Inserting Rows and Columns

We can add or delete Rows and columns in a Spreadsheet. To add column, click at the column header (right click on the mouse), there we get an option to add column (see figure 2.3). Likewise we can add row (see figure 2.4). To delete column, click at the column header (right click on the mouse), there we get an option to delete column. Likewise we can remove a row.

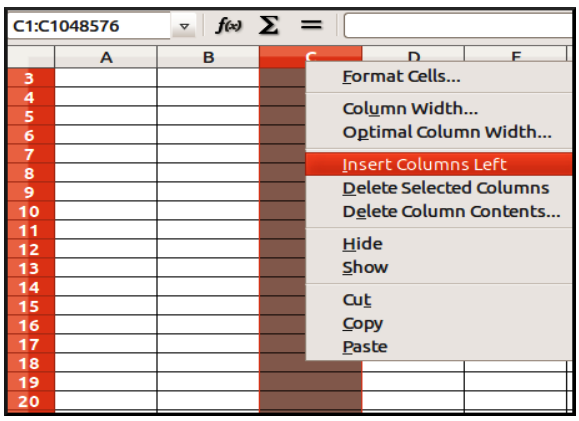


Figure 2.3 To add Column

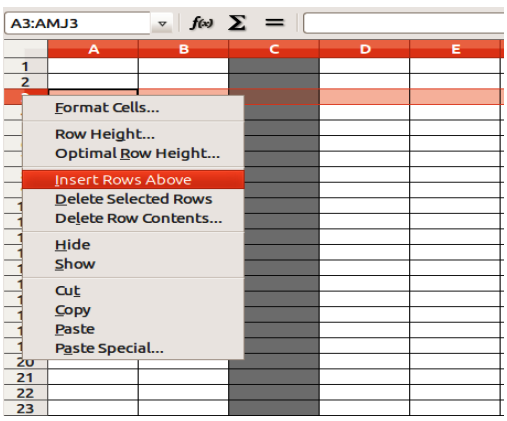


Figure 2.4 To add Rows

Cell

In spreadsheet, a value or function or an arithmetic expression is recorded in cells. The intersection of a Row and a Column is called a cell (Figure 2.2).

A cell is identified by a combination of a letter (column header) and a number (row header). For example, the first cell of a worksheet is identified as A1 (as it shown in Figure 2.1) at **column A** and **row 1**. As in the case, the cell having address as G8 is in the G column and 8th row. Each cell thus has a unique identification called as cell address.

Ranges

Range is a group of adjacent cells that forms a rectangular area. A range may contain just a single cell, or many cells. A range is specified by giving the address for first cell in the range and the last cell in the range. For example the range starting from B1 to B5 is written as B1:B5, where colon (:) is the range operator.

Naming Cells and Ranges

Naming ranges in Calc will save time for writing complex formulas. The name can be used in place of cell range whenever reference it e.g. in D4 we have =SUM(B2:F2) (Figure 2. 5)

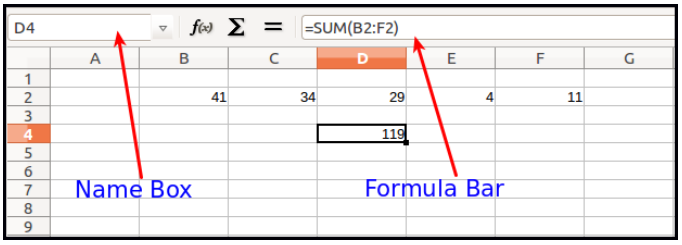


Figure 2.5 Example of Sum Function

The cell referenced in the function B2:F2 can be replaced with a descriptive name say Numbers (name range) which is easier to remember and in D4 it will be = SUM (Numbers).

Behind the Numbers Calc is hiding cell references, we will see how it works now. The steps are for defining Name Ranges are as follows:

1. Select the cell(s) which are to be named (Ex: B2:F2).
2. Select **Define Range** from the **Data** Menu
3. This will display a dialogue box as " Define Database Range" shown in Figure 2.6. It will provide a place to enter "Name" in which type "Numbers" which will represent cell ranges $B$2:$F$2 as shown in be "Range".
4. Click OK on the dialogue box which returns to the spreadsheet. Notice that the Name Box having our heading "Numbers".

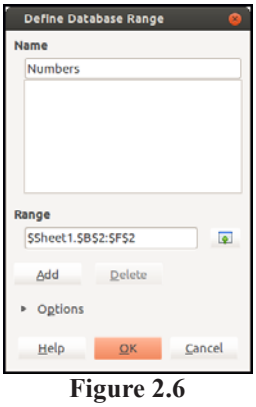


Figure 2.6
Define Data Range

The D4 will be having =SUM(Numbers) And will display the result (Figure 2.7).

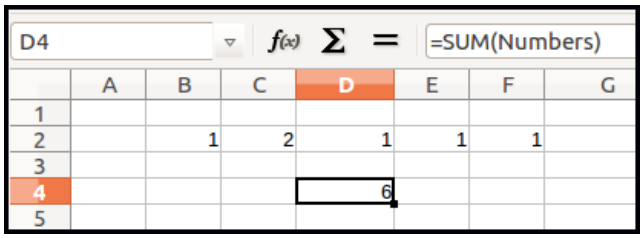


Figure 2.7 Example of Sum of Data Range

Now we will use a summation of numbers using condition in the cell D4. Type the formula = SUMIF (Numbers,"<6") and the answer will be 15 (for the Numbers less than 6 in the named range B2:F2) (Figure 2.8).

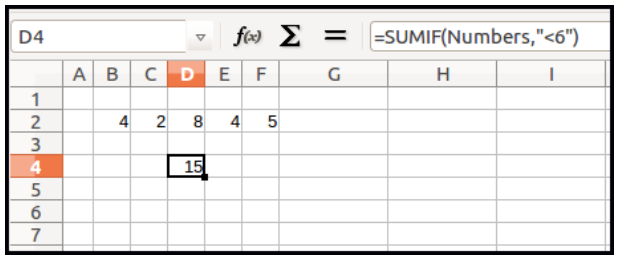


Figure 2.8 Example of Average of Data Range

The named range can be used with other Functions such as AVERAGE() (as shown in Figure 2.9).

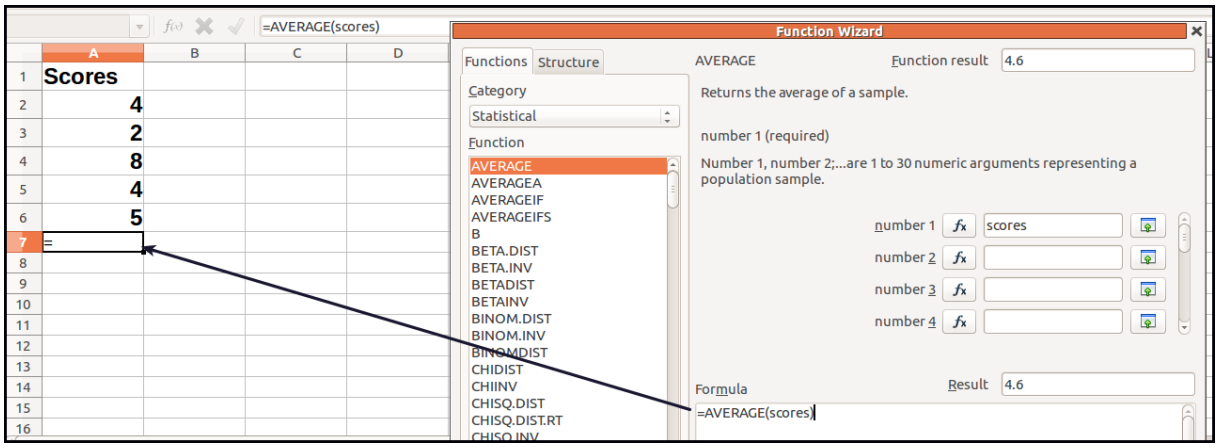


Figure 2.9 Example of Average of Data Range

Cell Reference

A cell reference identifies the location of a cell or group of cells in the spreadsheet also referred as a cell address. Cell references are used in formulas, functions, charts, other LibreOffice Calc commands and also refer to a group or range of cells. Ranges are identified by the cell references of the cells in the upper left (cell A1) and lower right (cell E2) corners in . The ranges are identified using colon (:) e.g. A1: E2 which tells LibreOffice Calc to include all the cells between these start and end points.

Relative Cell Reference: By default cell reference is relative; which means that as a formula or function is copied and pasted to other cells, the cell references in the formula or function change to reflect the new location.

Absolute cell reference: The other cell reference is absolute cell reference which consists of the column letter and row number surrounded by dollar ($) signs e.g. $C$4. An absolute cell reference is used when we want a cell reference to stay fixed on specific cell, which means that when a formula or function is copied and pasted to other cells, the cell references in the formula or function do not change.

Mixed cell reference: It is a combination of relative and absolute Cell references that holds either row or column constant when the formula or function is copied to another location e.g., $C4 or C$4.

Spreadsheet Navigation

We can move around a worksheet through four arrow keys (i.e. left, right, up, down). The mouse can also be used for navigation in Sheet except data entry. Some of the important operations and common navigations can be performed by using key strokes as shown in Table 2.1. Pressing a key is called key stroke but to fulfill one command for operation in the Sheet some time we require pressing two keys together to get one key stroke.

Movement	Key Stroke (Press key)
One cell down	Down arrow key or Enter key
One cell up	Up arrow key or Shift + Enter key
One cell right	Right arrow key or Tab key
One cell left	Left arrow key or Shift + Tab key

Table 2.1 Key stroke

The other navigational and operational strokes are used for faster cursor movement than one cell at a time with cluster of filled cells. Cluster of filled cells implies a set of consecutive cells in a row or in a column having some data.

Movement	Key Stroke (Press key)
Top of Sheet (cell A1)	***Ctrl + Home***
The cell at the intersection of the last row and last column containing data	***Ctrl + End***
Moves the cursor to the right edge of the current data range. If the column to the right of the cell that contains the cursor is empty, the cursor moves to the next column to the right that contains data.	***Ctrl + Right arrow***

Moves the cursor to the bottom edge of the current data range. If the row below the cell that contains the cursor is empty, the cursor moves down to the next row that contains data.	***Ctrl + Down arrow***
Beginning of the Row	***Home key*** or ***Ctrl + Left arrow***
Moves the cursor to the left edge of the current data range. If the column to the left of the cell that contains the cursor is empty, the cursor moves to the next column to the left that contains data.	***Ctrl + Left arrow***
Moves the cursor to the top edge of the current data range. If the row above the cell that contains the cursor is empty, the cursor moves up to the next row that contains data.	***Ctrl + Up arrow***
Moving consecutively to the last filled cells in a column	***End key***

Table 2.2 Keys for Navigation

Labels

A text or a special character will be treated as labels for rows or columns or descriptive information. Labels cannot be treated mathematically-multiplied, subtracted, etc. For example Name of Employees, Date of birth.

Formulas

The formula means a mathematical calculation on a set of cells. Formulas must start with an ' = ' sign (equal to sign), e.g. If the cell E3 have formula = D1+E1 which gives the sum of numbers in cell D1 and E1 in E3.

A spreadsheet without any formulas is a collection of data which are arranged in rows and columns (a database) like a calendar, timetable or simple list, etc. There is a input Line (Formula bar) on Libreoffice Calc toolbar and by using function wizard available in it we can insert any functions in spreadsheet.

When a cell contains a formula, it often contains references to other cells. Such a cell reference is a type of variable. Its value is the value of the referenced cell or some derivation of it. If that cell in turn references other cells, the value depends on the values of those.

By convention, the left hand side of equal to sign in a formula is normally considered as that cell itself

A formula identifies the calculation needed to place the result in the cell it is contained within. The cell E3 containing a formula, therefore it has two display components; the formula itself and the resulting value. The formula is shown only when the cell is selected by "clicking" the mouse over a particular cell; otherwise it contains the result of the calculation.

The arithmetic operations and complex nested conditional (what-if scenario) operations can be performed by spreadsheets which follow order of mathematical (expression) operations rules.

Order of mathematical operations (expressions)

Computer math uses of Algebra. Any operation[s] contained in brackets will be carried out first followed by any exponents.

After that, Calc considers division or multiplication operations to be of equal importance, and carries out these operations in the order they occur left to right in the equation.

The same goes for the next two operations – addition and substraction. They are considered equal in the order of operations. Whichever one appears first in an equation, either addition or subtraction is the operation carried out first.

Three easy ways to remember the order of operation is to use the acronym:

GEMS	**PEMDAS**		**BEMDAS**
( ) Grouping	( ) parenthesis	Please	( ) Brackets
^ Exponents	^ exponents	Excuse	^ Exponents
* Multiplication :	* multiply	My	* Multiplication
/ or Division :	/ divide	Dear	/ Division
- Subtraction :	+ add	Aunt	+ Addition
+ or Addition :	- subtraction	Sally	- Subtraction

Check yourself

1. What is Spreadsheet ?
2. Mention the purpose of Spreadsheet.
3. What are the features of spreadsheet?
4. List out the Spreadsheet software available.
5. What is FOSS? cite Examples
6. LibreOffice Calc is available in any other Operating System other than Linux?
7. What is worksheet?
8. Let us do Procedure to Add or delete a Column
9. Let us do procedure to add or delete a Raw
10. Unique Identification of a cell is called .....
11. Name box displays ——————————
12. What is range?

13. What is cell Reference?
14. What do you mean by Relative cell reference?
15. What do you mean by Absolute cell reference?
16. What do you mean by Mixed cell reference?
17. Let us do Practice With the Key Stoke
18. Which is the Key Stroke to Move to Cell A1 in LibreOffice Calc?
19. The intersection of a Row and a Column is called.....
20. What do you mean by Label? Cite Examples.
21. .....must start with an ' = ' sign in LibreOffice Calc (equal to sign)
22. What is Formula? Cite examples.
23. Input Line is also known as .....

FUNCTIONS

A function is a special key word which can be entered into a cell in order to perform and process the data which is appended within brackets. There is a function button ***f(x)*** on the formula bar. When we click on it, function offers assistance through function wizard. Alternatively we can enter the function directly into the formula bar.

A function is a built in set of formulas which starts with an 'equal to sign'(=) such as =FunctionName(Data). The data or argument includes a range of cells.

SUM(), AVERAGE() and COUNT() are common functions and relatively easy to understand. They each apply to a range of cells containing numbers (or blank but not text) and return either the arithmetic total of the numbers, the average mean value or the quantity of values in the range.SUM (), AVERAGE () and COUNT () are common functions and relatively easy to understand. They each apply to a range of cells containing numbers (or blank but not text) and return either the arithmetic total of the numbers, the average mean value or the quantity of values in the range.

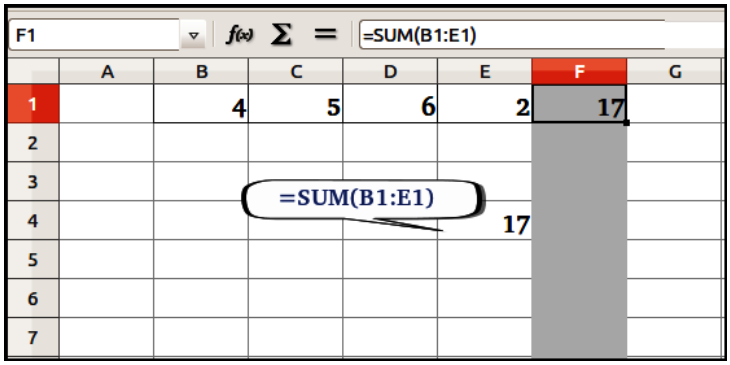


Figure 2.10 Function

For Example: The **SUM (Σ)** function is the most basic and one of the common user functions. It is used to get the addition of various numbers or the contents of various cells. On the Formula bar (figure 2.10) Sum button **(Σ)** can be use directly for summation of values from cells. Once we click the Sum button at cell F1, the function adds the contents of cell range B1 to E1

and displays the answer that we want to get the sum of. If we want answer in the cell F4 (Figure 2.10) use the mouse to click in the cell F4 and click on sum (Σ) button then from keyboard type range of the cells B1:E1; the answer 17 will appear in cell F4; or we can write directly the complete function = SUM (B1: E1) appears in the formula bar above the worksheet. The Sum function also includes other series based functions such as AVERAGE, MIN, MAX and COUNT.

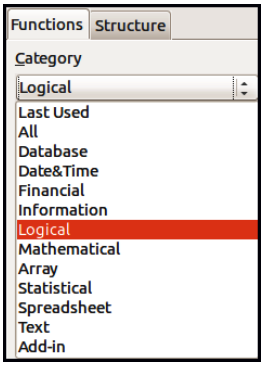


Figure 2.11 Functions

There are twelve different categories of functions available in LibreOffice Calc, which are classified as per the usage as shown in Figure 2.11. e.g. The Database, Date and Time, Financial, Information, Logical, Mathematical, Lookup and References and Text functions are useful in Computerised Accounting and will be explained later subsequently.

IF Function

In continuation to our need of what-if-scenario now we will learn about an important logical function IF Function. This function can be evoked from formula bar. This function returns one value if a specified condition evaluated to TRUE and another value if it evaluates to FALSE. We will learn more about the usage of functions in the business applications subsequently; there are a large selection of IF functions available.

Syntax : **IF (logical_test, value_IF_true, value_IF_false)**

where,

- logical_test : the value or expression that is determined to be true or false; this requires the usage of a logical operator. A logical operator is one used to perform a comparison between two values and produce a result of true or false (there is no middle result: something is not half true or half false or "Don't Know"; either it is true or it is false). For example, A1 < 20 could be used as a logical test, where symbol "<" is a logical operator "less than". (There are many more logical operators such as =, <=, <>, >, >= etc.)
- value_if_true : The value returned if the test is determined to be true. This value can be a value, text, or expression, formula, etc. or it can be return the value of another cell.
- value_if_false : The value returned, if the test is determined to be false. This value can be a value, text, or expression, formula, etc. or it can return the value of another cell.

Example,

i. = IF(A1 < 20, "Yes", "No")

this function will return "Yes" if cell A1 < 20 and "No" for anything else.

ii. = IF (C2 > B2, (C2+D2)/2, (B2+D2)/2)

this function will compare both cell C2 > B2 and will calculate and return

(C2+D2)/2 if it is true else it will calculate and return (B2+D2)/2.

iii. Example : Let us calculate the amount of saving (Cell address ”value”) on the basis of percentage value (Cell address “saving”) shown in Figure 2.12

F4 | =IF(E4>10000,0.1,0.05)

	A	B	C	D	E	F	G
1	Saving Percentage and Amount						
2							
3	Sr. No.	Name	Income	Expences	Total	Saving	Value
4	1	Ramesh	12,345.00	3,456.00	8,889.00	5.00%	444.45
5	2	Suzy	23,456.00	4,567.00	18,889.00	10.00%	1888.9
6	3	Ashok	12,340.00	5,678.00	6,662.00	5.00%	333.1
7	4	Mitesh	23,450.00	12,340.00	11,110.00	10.00%	1111
8	5	Bhadrakant	17,000.00	5,678.00	11,322.00	10.00%	1132.2
9	6	Khyati	9,735.00	2,345.00	7,390.00	5.00%	369.5
10	7	Nirav	13,467.00	4,567.00	8,900.00	5.00%	445
11	8	Priyanka	12,342.00	5,678.00	6,664.00	5.00%	333.2
12							

Figure 2.12 Example of IF function

Creating IF function using the Formula Tab and dialogue box.

1. Select the cell F4 (Figure 2.12) where the function is to be introduce
2. Click on *f*(*x*) in the formula bar and select logical from category of function.

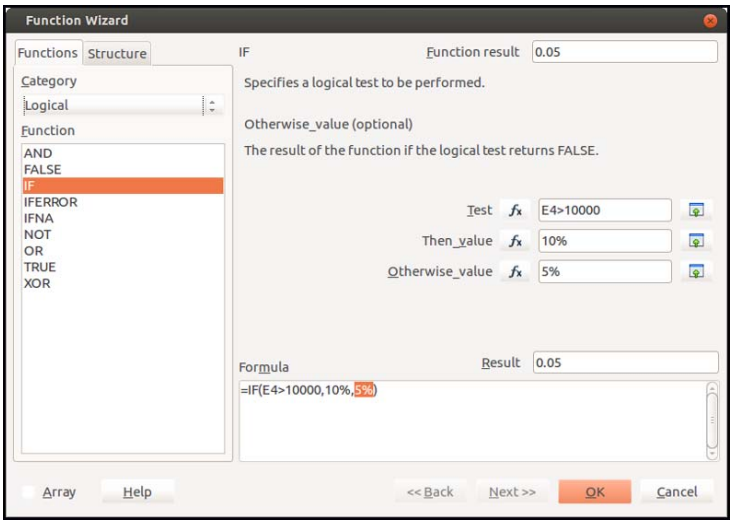


Figure 2.13 IF function wizard

3. Select IF function which will provide Function Arguments dialogue box (Figure 2.13).
4. Type an appropriate condition in the logical_test box (e.g. E4 > 10000)
5. In the **Then_Value** box type the require value (e.g. 10%) if the logical test condition is met.
6. In the **Otherwise_Value** box type the value (e.g. 5%) if the logical test condition is NOT met.

7. Click OK, the answer for the condition will be displayed (in cell F4 it will be 5%). Copy the function from F4 to all other cells F5:F11.

In the Formula Box the function will be displayed as

=IF (F4>10000, 10%, 5%)

This is simple use of IF function. IF can also we used to look at different types of functions.

Example **= IF (AVERAGE (A2:A6) > 10, SUM (B2:B6), 0)**

This function will be able to look at the average of cells A2 to A6 and if the average is higher than 10 it will sum the value of the cells B2 to B6, if the average is equal to or less than 10 it will return to 0.

The nested IFs can be used to look for several conditions. In some cases, we need to check more than one condition. In other words, check the first condition; if that condition is false, check another condition. If a nested function is used as an argument it must return the same type of value that the argument uses. For example, if the argument returns a TRUE or FALSE value, then the nested function must return a TRUE or FALSE otherwise Calc will display an error message #Value! in the cell.

This way we can check as many conditions as we need to. The truthfulness of each condition would lead to its own statement. If none of the conditions is true, then it executes the last statement. To implement this scenario include an IF() function inside of another. Such as :

= IF (logical_test, value_IF_true, value_IF_false) simple IF statement.

Let us substitute other IFs

IF (logical_test, IF (logical_test, IF (logical_test, value_IF_true, value_IF_false), value_IF_false), value_IF_false)

e.g. Suppose E2 cell contains marks of a test and cell F2 will have result based on following nested IF () condition.

= IF (E2<96, IF (E2<91, IF (E2<55,"Fail","C Grade"), "B Grade"), "A Grade")

Other Useful functions

The LibreOffice Calc provides twelve category of functions (Figure 2.11). in which input data can be worked as arguments and result available from the function will be output information. In business applications the input of data usually contains dates (date of invoice preparation, date of payment, payment received date, or due date etc.), rate of interest, tax percentage, delays in payment, accumulated interest, depreciation, future value, net present value, etc.

DATE & TIME FUNCTIONS

Date and Time functions are used for inserting, editing and manipulating date and time.

LibreOffice internally handles a date/time value as a numerical value. If you assign the numbering format "Number" to a date or time value, it is converted to a number. For example, 01/01/2000 12:00 PM, converts to 36526.5. The value preceding the decimal point corresponds to the date; the value following the decimal point corresponds to the time. This number can be converted in to date format by changing the format as date or time

To set the default date format – go to Tools → Options → LibreOffice Calc → Calculate (Figure 2.14)

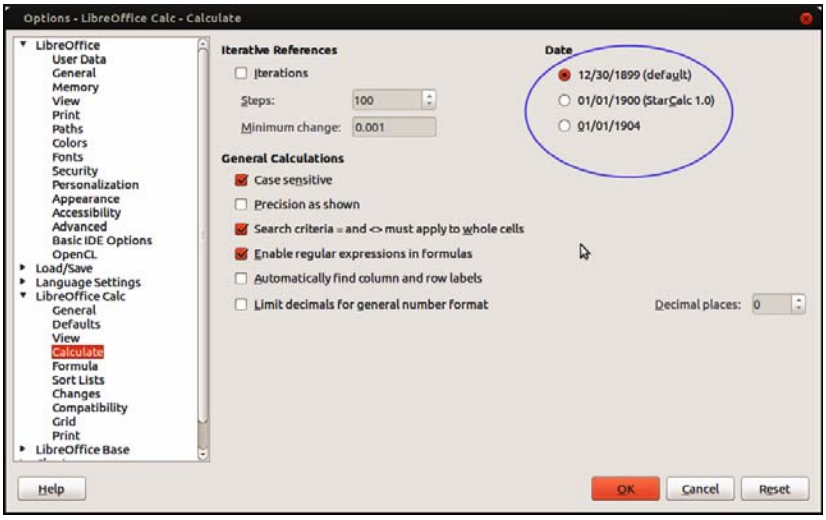


Figure 2.14 Setting default date format

When entering dates , " / " or " - " may be interpreted as arithmetic operators . So date values used for LibreOffice calc's function must be entered in double quotes. For Example "01/01/2017".

The most commonly used Date & Time functions are -

1) TODAY ()

Returns the current system date. The value is updated when you reopen the document or modify the values of the document.

Syntax: **=TODAY()**

TODAY is a function without arguments.(Figure 2.15)

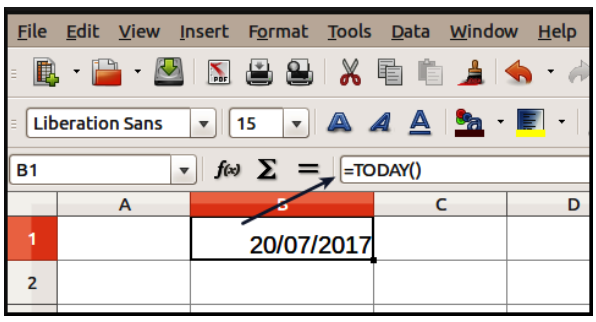


Figure 2.15 Example of TODAY function

2) NOW()

Returns the system date and time. The value is updated when you recalculate the document or each time a cell value is modified.

Syntax :- **=NOW()**

NOW is a function without arguments.

For example - **=NOW()** Returns the system date and time in B1 (Figure 2.16)

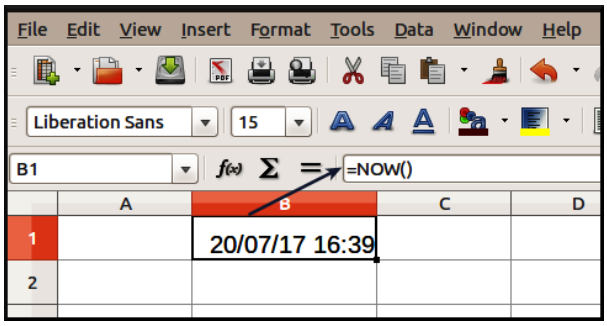


Figure 2.16 Example of NOW function

3) YEAR()

Returns the year for the given date value as a number .

Syntax : **=YEAR(Date value) or =YEAR("Date")**

Example : =YEAR("01/05/2016) returns 2016 as shown in Figure 2.17

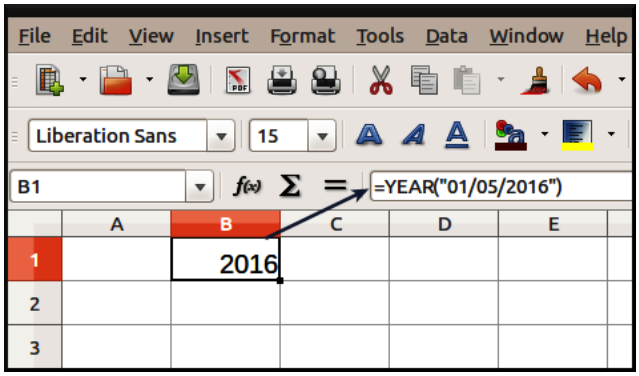


Figure 2.17 Example of YEAR function

4) MONTH()

Returns the month for the given date value. The month is returned as an integer between 1 and 12.

Syntax : **=MONTH("date") or MONTH(date value)**

Examples : =MONTH("05/08/2016") returns the result- 8 as shown in Figure 2.18

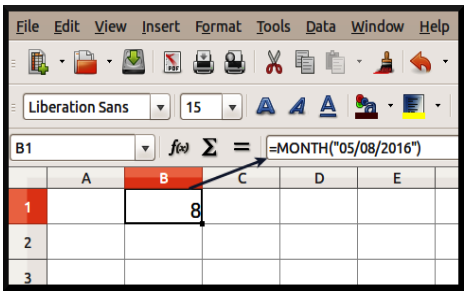


Figure 2.18 Example of MONTH function

5) DAY()

Returns the day of given date value. The day is returned as an integer between 1 and 31.

Syntax : **=DAY(Number)**

Examples : =DAY("25/05/2016") return 25 as shown in Figure 2.19

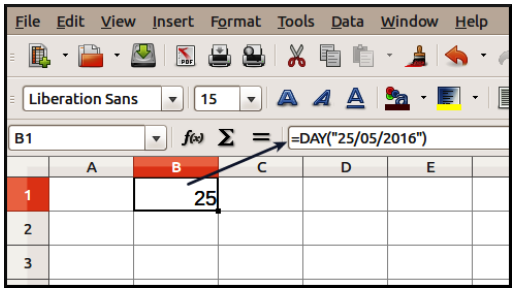


Figure 2.19 Example of DAY function

6) DATEVALUE()

The function "DATEVALUE" converts the given dates in to the corresponding internal date number. The internal date number is returned as a number. The result may change depending on system date settings. By default 31/12/1899 has the value as 1.

Syntax :- **=DATEVALUE("Date")**

'Date' is a valid date expression and must be entered with quotation marks.

Example : **=DATEVALUE("19/06/2017")** – Display 42905 as shown in Figure 2.20

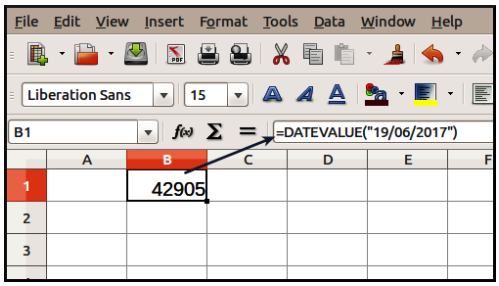


Figure 2.20 Example of DATAVALUE function

7)DATE()

This function calculates a date specified by year, month, day and displays it in the cell's formatting. The default format of a cell containing the DATE function is the date format, but you

can format the cells with any other number format.

Syntax : **=DATE(Year; Month; Day)**

Example : =DATE(2017,06,15) , it display as – 15/06/17 as shown in Figure 2.21

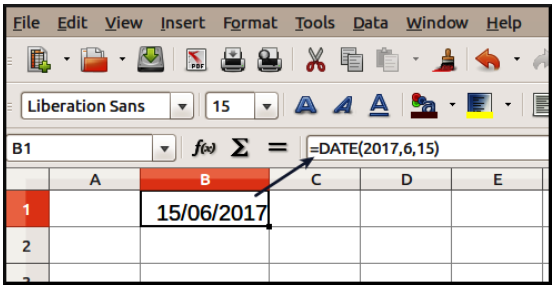


Figure 2.21 Example of DATE function

Let us consider an example , Mr.Rahul intends to apply for plus one admission this year. But the upper age limit is 22 as on 1st June 2017. Determine whether he is eligible to apply or not, based on the cut off age. His date of birth is 05/04/1994.

In this case , by using DATEVALUE() function calculate the serial number of 'date of birth' and 'as on date' by entering these formula

- **=DATEVALUE("05/04/1994")** in cell B1 and
- **=DATEVALUE("01/06/2017")** in cell B2.

The difference of two dates can be find out through the formula = **B2-B1** in cell B3. To find out the age in years enter the formula **=ROUND(B3/365.25,0)** in cell B4 . To determine his eligibility for plus one admission by using the formula **=IF(B4<=22," Eligible for admission","Not eligible for admission")** in B5. See Figure 2.22

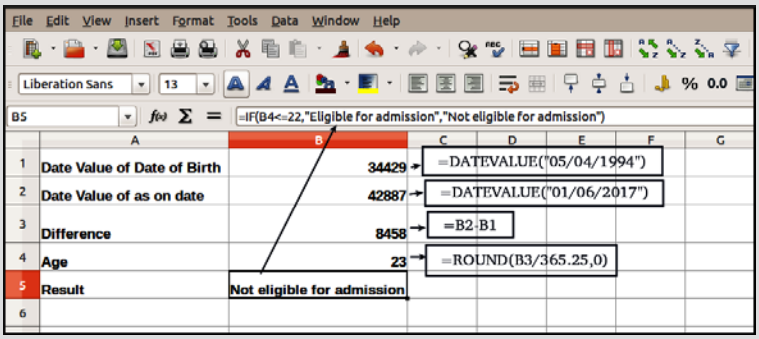


Figure 2.22 Example of DATEVALUE function

Check yourself

1. What is function?
2. List the Category of Function.
3. List out 3 Logical operators used in logical functions
4. **which** function returns the system date and time

5. Practice with IF Function: calculate sales commission @ 10% when sales reaches 30,000 or more

6. Let us practice with Naming range : Enter 5 PEN in A2:A6 Employee Name in B2:B6 and Basic Pay in C2:C6 Define Range for PEN, Employee Name and Basic Pay and sum Basic Pay

7. Which function returns the current system date.

8. Let us do Calculate your age

9. Use of **DATEVALUE** Function

10. **=DATEVALUE("01/07/2017")** – Returns ——————————-

11. Let us do Practice with Year function =YEAR() and feel the difference while entering year with double quotes and without quotes.

MATHEMATICAL FUNCTIONS

This category contains the **Mathematical** functions for Calc. To open the **Function Wizard**, choose Insert → Function.

The following mathematical functions are mainly used in business applications.

1. SUM()

Adds all the numbers in a range of cells.

This function is used to get the sum of the given numbers, cells or range of cells.

Syntax: = **SUM (number1,number2,.....)**

=SUM (cell1, cell2,.....)

=SUM(Range1,Range2,.....)

Examples: Refer figure 2.23

=SUM(10,15,20,30)

=SUM(A2,C2,D2) returns 60 in E3

=SUM(A2:D2) returns 75 in E2

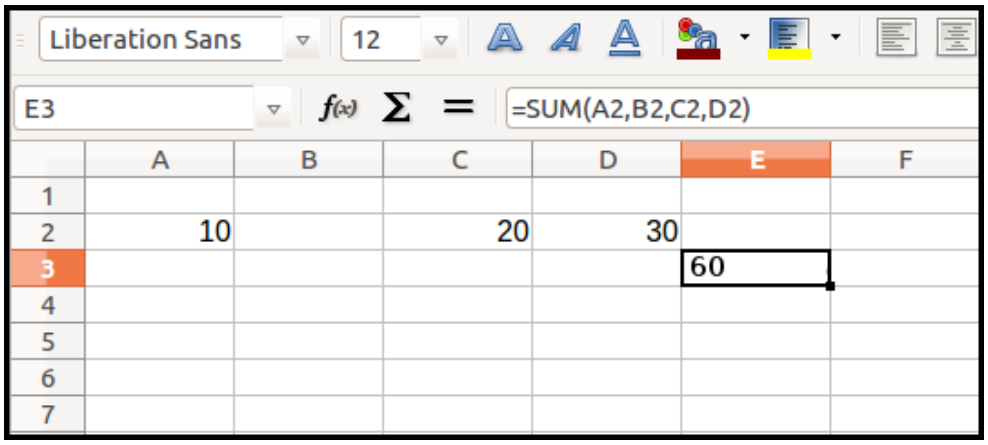


Figure 2.23 SUM function

2. SUMIF()

Adds the cells specified by a given criteria. This function is used to browse a range when you search for a certain value. It returns the sum of the cells as per a given criteria

Syntax : =SUMIF (range, "criteria", sum_range)

Where;

Range is the range to which the criteria are to be applied.

Criteria is the cell in which the search criterion is shown. If the criteria is written into the formula, it has to be surrounded by double quotes.

sum_range (Optional) is the range from which values are summed. If this parameter has not been indicated, the values found in the Range are summed.

Example: The following table (Figure 2.24) shows the values of Assets and its depreciation.

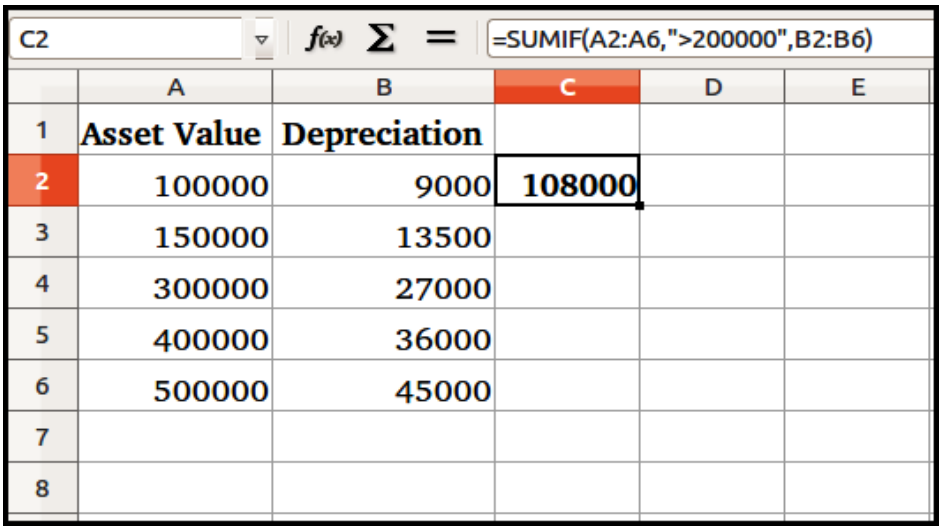


Figure 2.24 Value of assets and depreciation

No	Cases	Example	Result
	Total depreciation of those assets which is more than Rs 2,00,000	=SUMIF (A2:A6,“>200000”,B2:B6)	108000
	Total depreciation of those assets which is less than Rs 350000	=SUMIF (A2:A6,“<350000”,B2:B6)	49500
	Total depreciation of those assets which is exactly Rs 300000	=SUMIF (A2:A6,“300000”,B2:B6)	27000
	Total value of Assets which is less than Rs 180000	=SUMIF (A2:A6,“<180000”)	250000
	Total value of Assets which is more than Rs 350000	=SUMIF(A2:A6,”>350000")	900000

Table 2.3 Examples of SUMIF function

3.ROUND()

Rounds a number to a certain number of decimal places.

***Syntax* : =ROUND(Number,Count)**

Where;

Number : It is the number to round to specified number of digits

Count : It specifies the number of digits to round the number

If the number 128.758 is to be rounded to 2 decimal places, the formula to be given is

=ROUND(128.758,2) and the result will be 128.76

FORMULA	DESCRIPTION	RESULT
=ROUND(25431.2572,2)	Rounds 25431.2572 to 2 decimal place	25431.26
=ROUND(25431.2572,1)	Rounds 25431.2572 to 1 decimal place	25431.3
=ROUND(25431.2572,0)	Rounds 25431.2572 to 0 decimal place to nearest integer	25432
=ROUND(25431.2572,-1)	Rounds 25431.2572 to nearest 10 (left of the decimal)	25430
=ROUND(25431.2572,-2)	Rounds 25431.2572 to nearest 100 (left of the decimal)	25400

Table 2.4 Examples of ROUND function

If count is greater than zero, then number is rounded to the specified number of decimal places

If count is zero, then number is rounded to the nearest integer

If count is less than zero, then number is rounded to the left of the decimal point

4. ROUNDUP()

Rounds a number up, away from zero, to a certain precision. This function rounds a number up away from zero, without considering the value next to the rounding digit.

***Syntax* : =ROUNDUP(Number,Count)**

=ROUNDUP(45321.7354,2)	Rounds 45321.7354 UP to 2 decimal place	45321.74
=ROUNDUP(45321.7354,1)	Rounds 45321.7354 UP to 1 decimal place	45321.8
=ROUNDUP(45321.7354,0)	Rounds 45321.7354 UP to 0 decimal place	45322
=ROUNDUP(45321.7354,-1)	Rounds 45321.7354 UP to 1 decimal place to the left of the decimal (Rounds UP to the next 10)	45330
=ROUNDUP(45321.7354,-2)	Rounds 45321.7354 UP to 2 decimal place to the left of the decimal (Rounds UP to the next 100)	45400

Table 2.5 Examples of ROUNDUP function

5.ROUNDDOWN()

Rounds a number down, toward zero, to a certain precision.

Syntax **: =ROUNDDOWN(Number,Count)**

FORMULA	DESCRIPTION	RESULT
=ROUNDDOWN(45321.7354,2)	Rounds 45321.7354 down to 2 decimal place	45321.73
=ROUNDDOWN(45321.7354,1)	Rounds 45321.7354 down to 1 decimal place	45321.7
=ROUNDDOWN(45321.7354,0)	Rounds 45321.7354 down to 0 decimal place	45321
=ROUNDDOWN(45321.7354,-1)	Rounds 45321.7354 down to 1 decimal placeto the left of the decimal (Rounds down to the next 10)	45320
=ROUNDDOWN(45321.7354,-2)	Rounds 45321.7354 down to 2 decimal placeto the left of the decimal (Rounds down to the next 100)	45300

Table 2.6 Examples of ROUNDDOWN function

STATISTICAL FUNCTIONS

Statistical function operates on a set of data and give summarised results. LibreOffice Calc provides a number of statistical functions. A few of them are discussed in this section

1) COUNT()

This function used to count the number of cell in a range that contain numbers and counts the numbers within the list of arguments separated by commas. The numbers also include date and time . Empty cells, logical values , text and error codes ignored. Only numbers are counted here.

Syntax : **=COUNT(Value1; Value2; ...)**

= COUNT (Cell1,Cell2,Cell3,.....)

= COUNT (Range)

where,

Value1; Value2, ... are values or ranges representing the values to be counted.

Example : =COUNT(45,24,Ab,55,72,Ab,47) returns 5

=COUNT(A1:D8)

=COUNT(A1,B1,C1,D1,E1) return 3 as shown in Figure 2.25

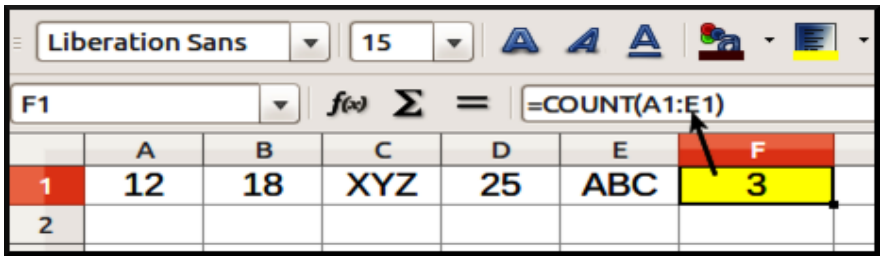


Figure 2.25 Examples of COUNT function

2) COUNTA()

The COUNTA() function counts the number of cells which contain any value. It will count numbers, text, date, time, logical values, error codes in the specified range or list of arguments. In other words, this function counts the number of cells that are not empty in a range.

Syntax **:- =COUNTA(Value1; Value2; .....)**

=COUNTA(Cell1,Cell2,Cell3,.....)

=COUNTA(Range)

where,

Value1; Value2, ... are arguments representing the values to be counted.

Example : =COUNTA(56,88,45, Ab,45,Ab,47) return 7

=COUNTA(A1,A2,A3,A4)

=COUNTA(A1:F1) return 6 as shown in Figure 2.26

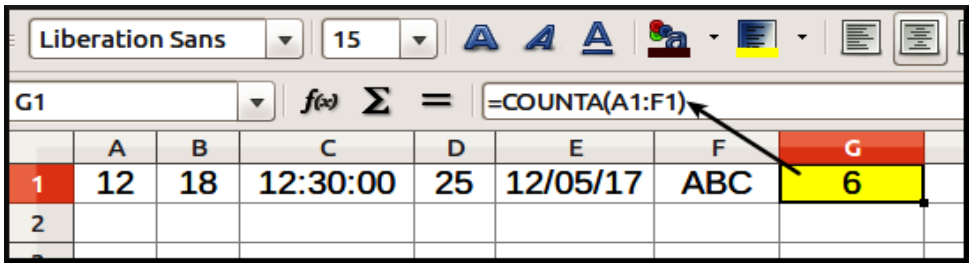


Figure 2.26 Examples of COUNTA function

3) COUNTBLANK()

COUNTBLANK function count the number of empty cell in the given range. It is the opposite function of COUNTA.

***Syntax* : =COUNTBLANK(Range)**

Example : =COUNTBLANK(A1:E1) return 1 as shown in Figure 2.27

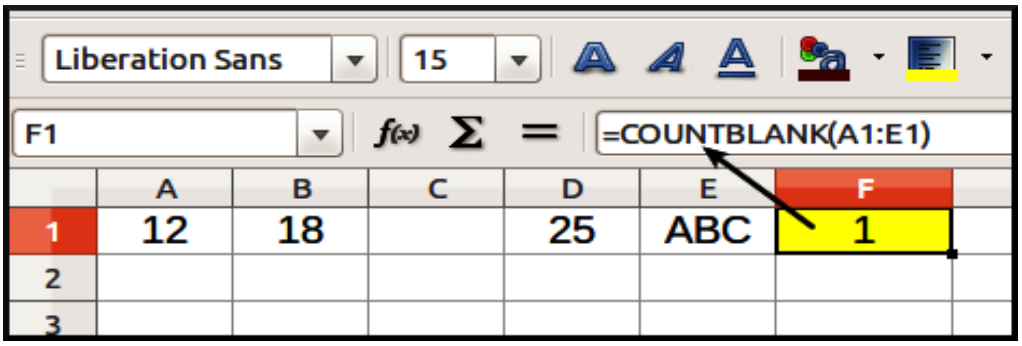


Figure 2.27 Examples of COUNTBLANK function

4) COUNTIF()

This function counts the number of cells within a given range that meet the user defined criteria or condition. Here the blank cells and text values are ignored.

Syntax : **=COUNTIF(Range; Criteria)**

Range : range to which the criteria are to be applied.

Criteria – it can be a number, an expression, cell reference or a character string. For example, criteria can be expressed as 5000, ">32", "Commerce", or C4. If you search for literal text enclose the text in **double quotes.**

Example : A1:A10 is a cell range containing the numbers 2000 to 2009. Cell B1 contains the number 2006. In cell B2,C2 and D2 enter these formula:

=COUNTIF(A1:A10;2006) - returns 1

=COUNTIF(A1:A10;B1) - returns 1

=COUNTIF(A1:A10;">=2006") - returns 4

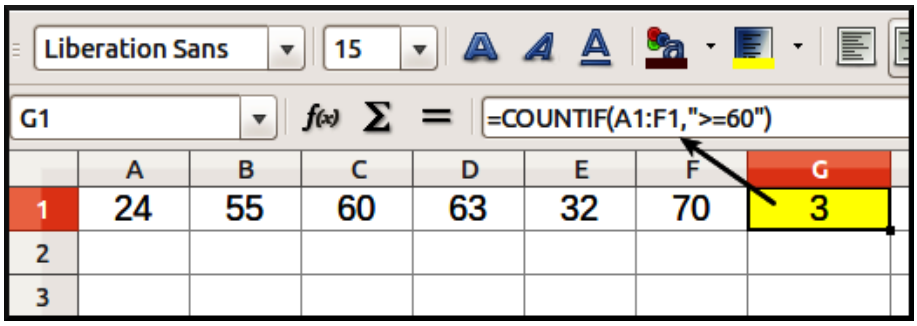


Figure 2.28 Examples of COUNTIF function

5) ROWS()

The function returns the number of rows in a reference or array

***Syntax* : =ROWS(Array)**

'Array' - is the reference or named area whose total number of rows is to be determined (See figure 2.29)

Example : =ROWS(A1:H7)

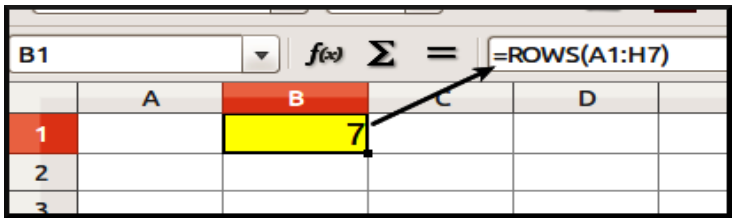


Figure 2.29 Examples of ROWS function

6) COLUMNS()

This function returns the number of columns in an array or reference

Syntax : **=COLUMNS(Array)**

'Array' - is the reference to a cell range whose total number of columns is to be found. (See figure 2.30)

Example : =COLUMNS(B1:J2)

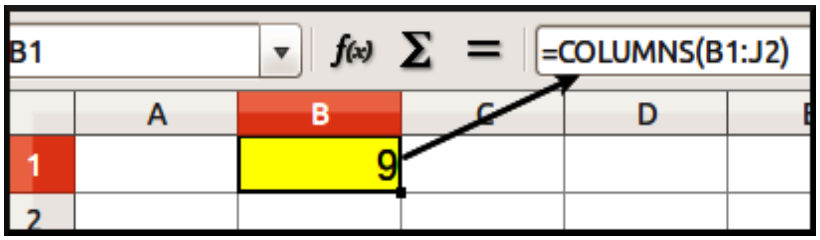


Figure 2.30 Examples of COLUMNS function

Function	Syntax	Description
COUNT	=COUNT(Value1;Value2;.....)	Count the number of cell in a range that contain numbers
COUNTA	COUNTA(Value;Value2;.....)	Counts the number of cells which contain any value
COUNTBLANK	=COUNTBLANK(Range)	Count the number of empty cell in the given range
COUNTIF	=COUNTIF(Range;Criteria)	Count the number of cells within a given range that meet the given criteria
ROWS	=ROWS(Array)	Returns the number of rows in a reference or array
COLUMNS	=COLUMNS(array)	Returns the number of columns in an array or reference

Table 2.7 Statistical Functions

TEXT MANIPULATION FUNCTION

There are two types of Text Manipulation Functions

- **TEXT**
- **CONCATENATE**

1. TEXT()

This function converts a number or numerical value into text according to a user defined format.

***Syntax* : TEXT(Number,Format)**

where,

Number is the numerical value to be converted.

Format is the text, which defines the format. Use decimal and thousands separators 0 and # according to the language set in the cell format. Format text is a numeric format as a text string enclosed in quotation marks.

This function is useful in situations to display numbers in a more readable format, or to combine numbers with text or symbols. For example, suppose cell A1 contains the number 123.45. Suppose we want to format this number by adding currency symbol " ₹ " and convert into amount using this function returns ₹123.45

To Set numeric format go to:

1. Format → Cells

2.Click on Number Tab

3.Select Numeric formats from category list (refer figure 2.31)

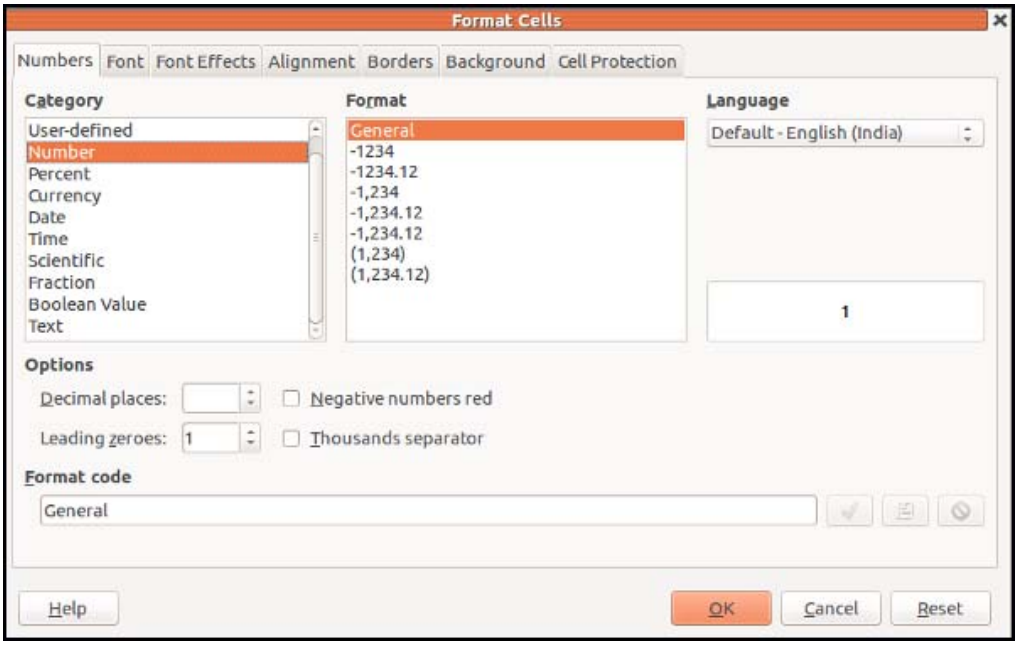


Figure 2.31 Format Cells

=TEXT(A1,"₹ 000.00") Returns the text ₹123.45

=TEXT(12.34567;"###.##") returns the text 12.35

=TEXT(12.34567;"000.00") returns the text 012.35

=TEXT(01062017,"00-00-0000") Returns the text 01-06-2017

=TEXT(TODAY(),"DD/MM/YYYY") Today's date in DD/MM/YYYY

Returns 5/06/2017

=TEXT(TODAY(),"DDDD") Today's day of the week, Returns Monday

=TEXT(NOW(),"H:MM AM/PM") Current time, Returns 1:29 PM

=TEXT(0.285,"0.0%") Percentage, Returns 28.5%

2. CONCATENATE()

This function used to Combines several text strings in different cells into one string.

***Syntax* : CONCATENATE("Text 1", ...,"Text n")**

Text 1, Text 2,... Text n represent up to n text passages which are to be combined into one string. The text items can be text strings, numbers, or single-cell references.

Example 1 :

=CONCATENATE("Good ";"Morning ";"Mrs. ";"Joel")

Returns: GoodMorningMrs.Joel.

=CONCATENATE(A1,B1,C1,D1)

Returns: Good Morning Mrs. Joel.

Example 2

If the data given in cell A1= 15/06/2017 B1=Thursday C1=7:41P.M.

=CONCATENATE(A1,B1,C1,)

Returns 15/06/2017Thursdaay7:41P.M.

Example 3.

Alpha numeric value listed in Columns with Space

A1=KL A2=11D (provide space) A3=12 A4=34

=CONCATENATE(A1,A2,A3,A4)

Returns KL11D 1234 (see figure 2.32)

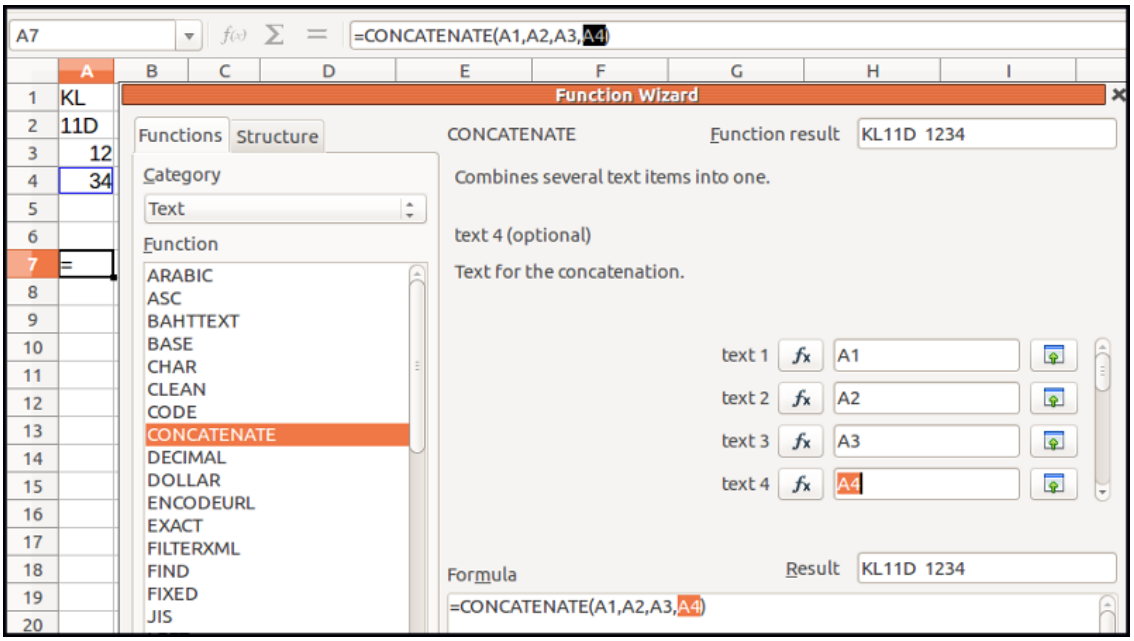


Figure 2.32 CONCATENATE function wizard

Check Yourself

1. The Syntax of SUMIF( ) is _____

2. _____ function is rounds a number to a certain number of decimal places.

3. Examples of ROUND( ) is _____

4. _____ function is used to count the number of cell in a range

5. =COUNT(45,24,Ab,55,72,Ab,47) returns _____

6. COUNTBLANK function count the number of _____ cell in the given range.

7. _____ function returns the number of rows in a reference or array

8. CONCATENATE function used to _____ several text strings in different cells into one string.

Multiple choice questions

9. Assume cell A1 = 10000, A2 =15000, A3 = 12000, A4 = 14000 What is the result of SUMIF statement: =SUMIF (A1:A4, "<14000")

   (15000, 24000, 27000, 22000)

10. What is the result of the following

    =ROUNDUP(45321.7354,0)

    (45322, 45321, 45321.7)

11. The function which rounds a number up, away from zero, to a certain precision

    [ROUND( ), ROUNDUP( ), ROUNDDOWN( )]

12. The function counts the number of cells which contain any value.

    (COUNT, COUNTA, COUNTBLANK)

Short Answer Type

13. Explain the purpose of COUNTIF Function.

14. What is the Purpose of ROUNDDOWN( )

15. Explain the use of TEXT( )

16. Write two text manipulation functions in Libreoffice Calc.

LOGICAL FUNCTIONS

Logical functions are used for comparison and checking a test condition. The major logical functions are IF, AND and OR.

IF()

IF function is used to test a condition. When the condition is TRUE , then first action is taken. When it is FALSE, then the second action is taken.

Syntax : **=IF(Test, ThenValue, OtherwiseValue)**

Where,

Test -is any value or expression that can be TRUE or FALSE.

ThenValue - is the value that is returned if the logical test is TRUE.

OtherwiseValue- is the value that is returned if the logical test is FALSE.

Example : =IF(A2>=24,"PASSED","FAILED")

If the value in cell A2 is greater than or equal to 24, the function returns "PASSED", Otherwise "FAILED".

In LibreOffice Calc functions, arguments or parameters marked as "optional" can be left out only when no parameter follows. For example, in a function with three parameters, where the last two parameters are marked as "optional", you can leave out parameter 3 or parameters 2 and 3, but you cannot leave out parameter 2 alone.

NESTED IF

Calc allows to include one function inside another function. It is called nesting of functions. The IF function can be nested, when you have multiple conditions to meet.

Syntax : **=IF(Test_1, ThenValue_1,IF(Test_2, ThenValue_2,IF(.....)))**

Example : The following are the scores obtained by students in an examination

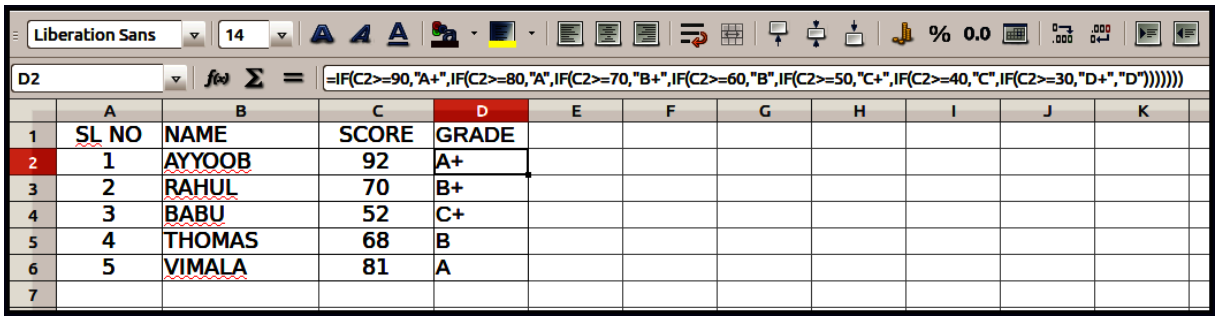
	A	B	C	D
1	**SL NO**	**NAME**	**SCORE**	**GRADE**
2	1	AYYOOB	92	
3	2	RAHUL	70	
4	3	BABU	52	
5	4	THOMAS	68	
6	5	VIMALA	81	

Ascertain the grade obtained by students based on the following criteria.

Score	Grade	Score	Grade
90-100	A+	50-59	C+
80-89	A	40-49	C
70-79	B+	30-39	D+
60-69	B	Less than 30	D

In this case the following formula may be given in D2. In this case, If the first condition tested is FALSE, it tests the next and so on. Every time the FALSE value is being replaced by another IF function to make a further test. (see figure 2.33)

=IF(C2>=90,"A+",IF(C2>=80,"A",IF(C2>=70,"B+",IF(C2>=60,"B",IF(C2>=50,"C,
IF(C2>=40,"C",IF(C2>=30,"D+","D")))))))



	A	B	C	D	E	F	G	H	I	J	K
1	SL NO	NAME	SCORE	GRADE							
2	1	AYYOOB	92	A+							
3	2	RAHUL	70	B+							
4	3	BABU	52	C+							
5	4	THOMAS	68	B							
6	5	VIMALA	81	A							
7											

Figure 2.33 Nested IF

AND()

Returns TRUE if all arguments are TRUE. If any element is FALSE, this function returns the FALSE value.

***Syntax* : AND(LogicalValue1, LogicalValue2, ...)**

Logical value 1,Logical value 2, ..., are conditions to be checked. All conditions can be either TRUE or FALSE.

Example :

=AND(10<13,14>12,7>6) returns TRUE

=AND(12<13,14>12,7<6) returns FALSE.

=AND (FALSE,TRUE) returns FALSE.

AND function can be used along with IF function. (see figure 2.34)

EXAMPLE :

FORMULA	DESCRIPTION	RESULT
=IF(AND(A1>1,A1<100),A1, "The value is out of range")	Displays the number in cell B1, if it is between 1 and 100. Otherwise , it displays the message " The value is out of range	60

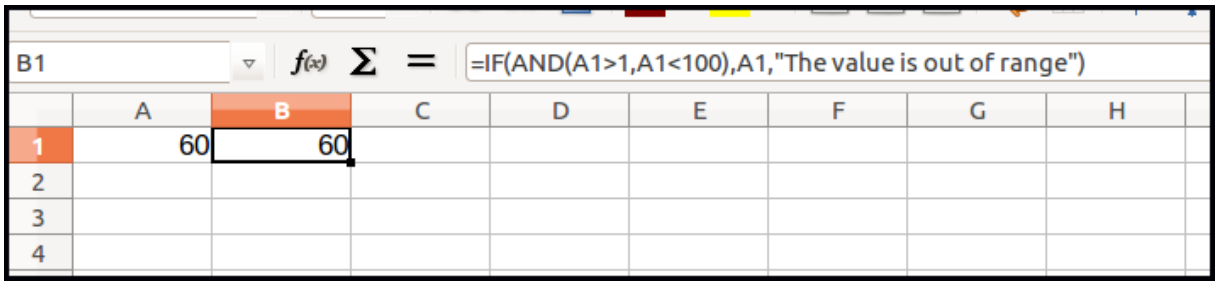


Figure 2.34 Example using AND with IF

OR()

Returns TRUE if at least one argument is TRUE. This function returns the value FALSE, if all the arguments have the logical value FALSE.

***Syntax* : OR(LogicalValue1,LogicalValue2, ...)**

LogicalValue1, LogicalValue2 ..... are conditions to be checked. All conditions can be either TRUE or FALSE.

Example

=OR(12<11,13>22,45=45) returns TRUE.

=OR(FALSE,TRUE) returns TRUE.

=OR(12<11,13>22) returns FALSE

OR function can be used along with IF function. (see figure 2.35)

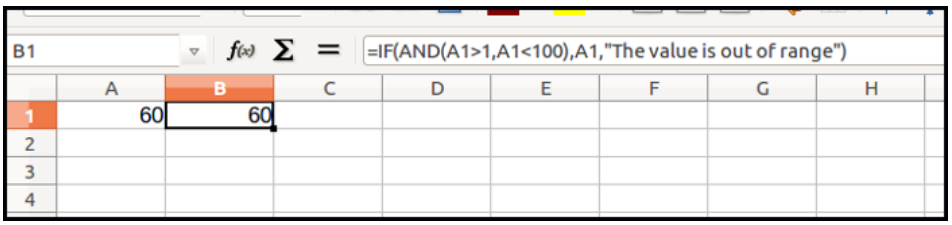


Figure 2.35 Example using OR with IF

Example :

FORMULA	DESCRIPTION	RESULT
=IF(OR(A1=24,A1>24), "Passed", "Failed")	Displays the message "Passed" in cell B1, if it is greater than or equals to 24 . Otherwise , it displays the message " Failed".	Passed

SPREAD SHEET (REFERENCE) FUNCTIONS

LOOKUP Functions

Returns the contents of a cell either from a one-row or one-column range or from an Array. Additionally, the search vector for the LOOKUP must be sorted ascending, otherwise the search will not return any usable results. If LOOKUP cannot find the search criterion, it matches the largest value in the search vector that is less than or equal to the search criterion.

Lookup function has two syntax forms

i) Vector form and ii) array form.

LOOKUP (Vector form)

***Syntax* : LOOKUP(SearchCriterion,SearchVector,ResultVector)**

where,

SearchCriterion is the value to be searched for, entered either directly or as a reference.

SearchVector is the single-row or single-column area to be searched.

ResultVector is another single-row or single-column range from which the result of the function is taken. The result is the cell of the result vector with the same index as the instance found in the search vector.

Example: We can Find out the Basic Pay of an Employee with PEN "12347" from the following table (figure 2.36)

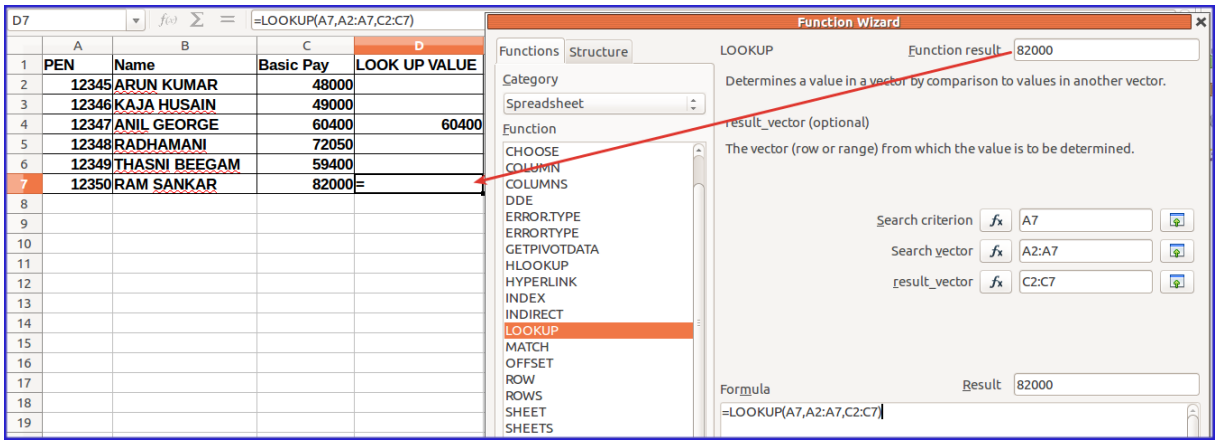


Figure 2.36 LOOKUP function wizard (Vector Form)

=LOOKUP("12347",A2:A7,C2:C7) Returns 60400

Formula	Description	Result
=LOOKUP("12345",A2:A7,C2:C7)	Looks up 12345 in column A and returns the value from column C in the same row.	48000
=LOOKUP("12351",A2:A7,C2:C7)	Looks up 12351 in column A matches the next smallest value and returns the value from column C	82000

Table 2.8 Examples of LOOKUP function (Vector Form)

LOOKUP (Array form)

An array is a collection of values in rows and columns (like a table) that you want to search.

Syntax : **LOOKUP(lookup_value, array)**

Example: We can Find out the Students Grade for different subjects based on their score (see figure 2.37)

=LOOKUP(78,B9:C15) **Returns B+**

Here Array B9:C15 has 2 columns and 7 rows So LOOKUP searches in the first column B9 and returns the Grade from Column C

Formula	Description	Result
=LOOKUP(78,B9:C15)	LOOKUP 78 In the Column C and Returns the Value from Column D	B+
=LOOKUP(93,B9:C15)	LOOKUP 93 In the Column C and Returns the Value from Column D	A+
=LOOKUP(43.B9:C15)	LOOKUP 43 In the Column C and Returns the Value from Column D	C

Table 2.9 Examples of LOOKUP function (Array Form)

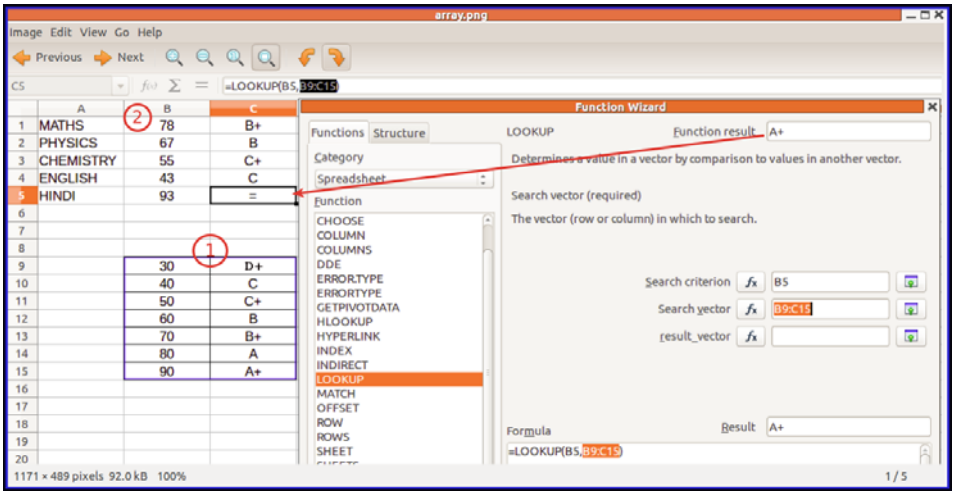


Figure 2.37 LOOKUP function wizard (Array Form)

*The array form of **LOOKUP** is very similar to the **HLOOKUP** and **VLOOKUP** functions. The difference is that **HLOOKUP** searches for the value of **lookup_value** in the first row, **VLOOKUP** searches in the first column, and **LOOKUP** searches according to the Dimensions of array.*

VLOOKUP

VLOOKUP is the Vertical LOOKUP function. Use VLOOKUP to search the first column (columns are vertical) of a block of data and return the value from another column in the same row.

***Syntax* : =VLOOKUP(SearchCriterion;Array;Index;SortOrder)**

SearchCriterion is the value searched for in the first column of the array.

Array is the reference, which is to comprise at least two columns.

Index is the number of the column in the array that contains the value to be returned. The first column has the number 1.

SortOrder is an optional parameter that indicates whether the first column in the array is sorted in ascending order. even if the search value was not matched exactly, if it is between the lowest and highest value of the sorted list.In unsorted lists, the search value must be matched exactly. Otherwise the function will return error message:

if the exact SearchCriterion is not found, the last value that is smaller than the criterion will be returned.

If the SortOrder parameter is omitted or set to TRUE or one, it is assumed that the data is sorted in ascending order.

*If SortOrder is set to FALSE or zero, an exact match must be found, otherwise the **Error:** will be the result.*

Example :

We can Find out the Basic Pay of an Employee with PEN “555” from the following table

Lookup the Basic Pay for PEN “555” in the first column (A9) and returns the matching value in the same row 4th the Column

Example : **=VLOOKUP(555,A4:D11,4,0) Returns 87000** *(Figure 2.38)*

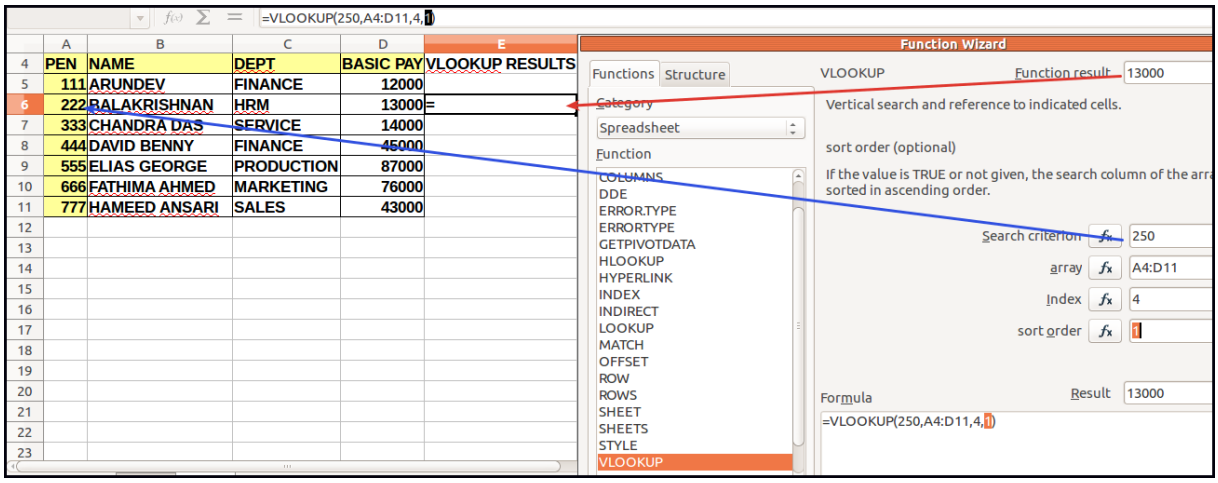


Figure 2.38 Example of VLOOKUP

FORMULA	DESCRIPTION	RESULT
=VLOOKUP(250,A4:D11,4,1)	Looks up 250 in first column (A6) matches the next smallest value i.e.(222) and returns the matching in the same row of the 4th column.	13000
=VLOOKUP(250,A4:D11,4,0)	Looks up 250 in first column (A6) *exact match not found (Sorting order 0)* and returns *the error* **Error: ##NA**	#NA
=VLOOKUP(777,A6:D13,4,0)	Looks up 777 in first column (A6) *exact match not found (Sorting order 0)* and returns the matching value in the 4th column.	43000

Table 2.10 VLOOKUP

Use VLOOKUP when:

- The data is arranged in rows and you want to return data from the same row.
- Searching the first column of a range of data.

HLOOKUP()

Searches for a value in the first row of an array and returns the corresponding value in a row of the array, named in the Index, in the same column. HLOOKUP supports the same form and arguments as VLOOKUP:

Syntax : **HLOOKUP(SearchCriteria; Array; Index; Sorted)**

SearchCriterion is the value searched for in the first row of the array.

Array is the reference, which is to comprise at least two columns.(table)

Index is the number of the row in the array that contains the value to be returned. The first column has the number 1.

SortOrder is an optional parameter

Example :

We can Find out the Basic pay in the 6th row from the following table

=HLOOKUP(D5,A5:D11,6,0) Returns 76000

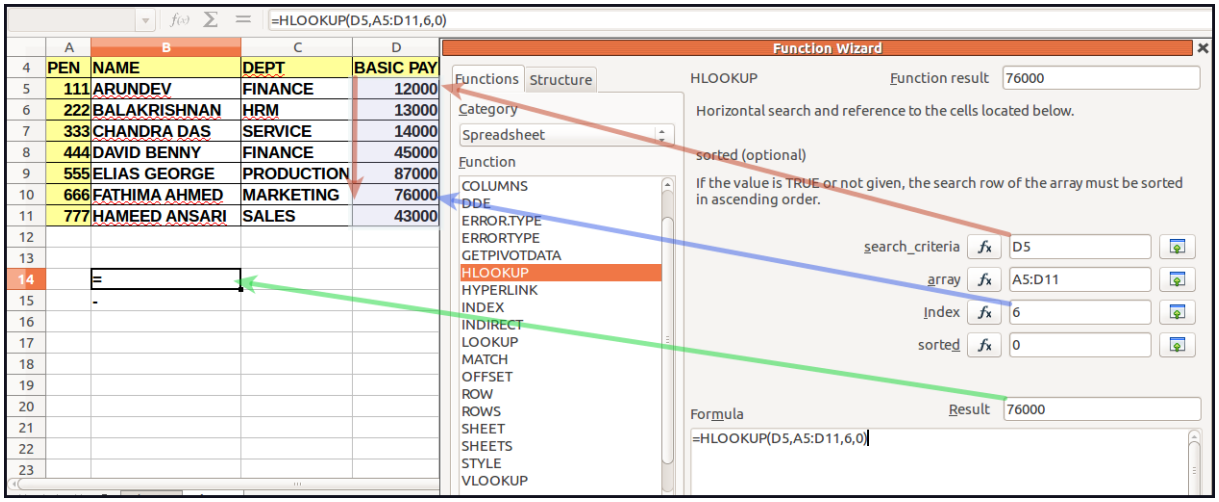


Figure 2.39

Description	Result
=HLOOKUP(B5,A5:D11,6,0)	*Looks up B5 and returns the indexed value from 6th row* FATHIMA AHMED
=HLOOKUP(C5,A4:D11,6,0)	*Looks up C5 and returns the indexed value from 6th row* MARKETING

Table 2.11

Check Yourself

1. IF function comes under .....(Logical, Statistical,Text Manipulation, Spreadsheet)
2. AND function return True if .....(all / any) arguments are true
3. The function used to test a condition is .....
4. A function inside another function is called _____

   a)Nested Function b)Round function c)Sum function d) Text function

5. Which function is used while combining more than one test condition and the result will be true if any of the condition is True

6. .....returns True if any of the condition is True;returns False if all the conditions are False

7. Find out the Odd out – (1)AND (2) OR (3) IF (4) SUM.

8. Write an examples for logical functions with examples.

9. Write down the syntax of the following function -

   a. VLOOKUP b. HLOOKUP

10. ..... is a collection of values in rows and columns (like a table) that you want to search in LOOKUP function

11. .....function in LibreOffice Calc is used to search one column of data and find data in the corresponding row.

12. .....function searches for a value in the first column and display the data from the same row.

13. .....function searches for a value in the first row and retrieves the data from the same column.

14. VLOOKUP function used to ?

(a)Find related records (b) Lookup text that contain "V" (c) Check if two cells are related.

FINANCIAL FUNCTIONS

1. ACCRINT()

Calculates the accrued interest of a security in the case of periodic payments.

Syntax : **ACCRINT**(issue, first_interest, settlement, rate, par, frequency, basis, calc_method)

Issue (required) is the issue date of the security.

FirstInterest (required) is the first interest date of the security.

Settlement (required) is the date at which the interest accrued up until then is to be calculated.

Rate (required) is the annual nominal rate of interest (coupon interest rate)

Par (required) is the par value of the security.

Frequency (required) is the number of interest payments per year (1, 2 or 4). Yearly, Half-yearly or Quarterly

Basis (optional) is chosen from a list of options and indicates how the year is tobe calculated.

Basis	Calculation
0 or missing	US method (NASD), 12 months of 30 days each
1	Exact number of days in months, exact number of days in year
2	Exact number of days in month, year has 360 days
3	Exact number of days in month, year has 365 days
4	European method, 12 months of 30 days each

In the LibreOffice Calc functions, parameters marked as "optional" can be left out only when no parameter follows.

For example, in a function with four parameters, where the last two parameters are marked as "optional",you can leave out parameter 4 or parameters 3 and 4, but you cannot leave out parameter 3 alone.

A security is issued on 01/04/2016. First interest is set for 30/09/2016. The settlement date is 01/07/2016. The Rate is 0.1 or 10% and Par is Rs.100,000 currency units. Interest is paid half-yearly (frequency is 2). The basis is the US method (0). How much interest has accrued?

=ACCRINT(B1,B2,B3,B4,B5,B6,B7) Returns ₹2500. See Figure 2.40

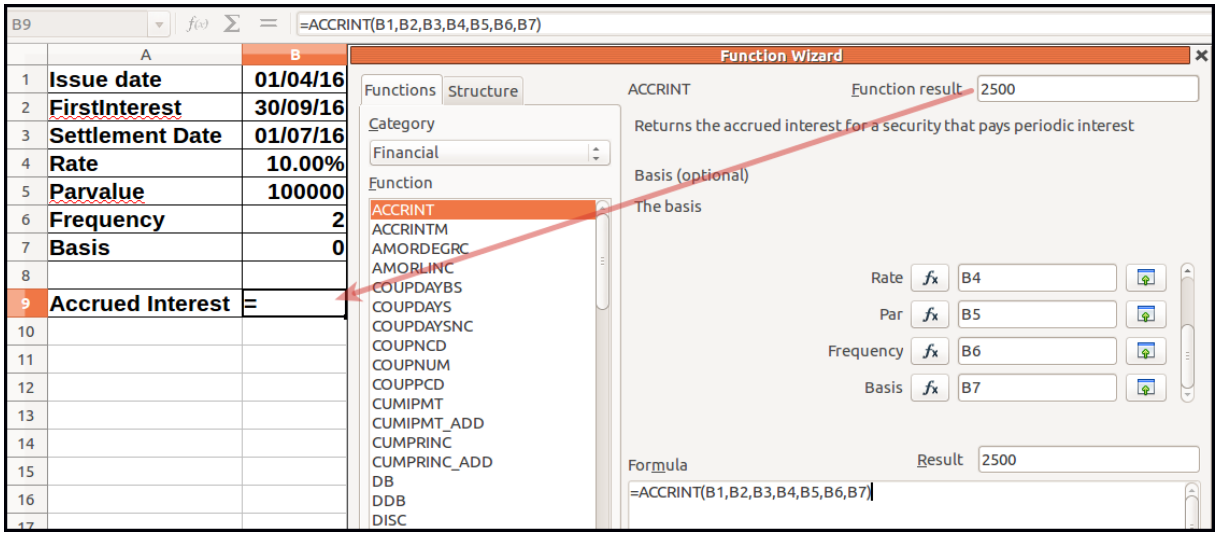


Figure 2.40 Example of ACCRINT

2. CUMIPMT()

Calculates the cumulative interest payments, that is, the total interest, for an investment based on a constant interest rate. This function returns the cumulative interest paid between two periods.

Syntax : **=CUMIPMT(Rate,NPER,PV,S,E,Type)**

Rate is the periodic interest rate. (convert the annual interest rate into monthly rate)

NPER is the payment period with the total number of periods. NPer can also be a non-integer value. (Period in months)

PV is the current value in the sequence of payments.

S is the first period. Payment periods are numbered beginning with 1.

E is the last period.

Type is the due date of the payment. (which may be either 0 or 1)

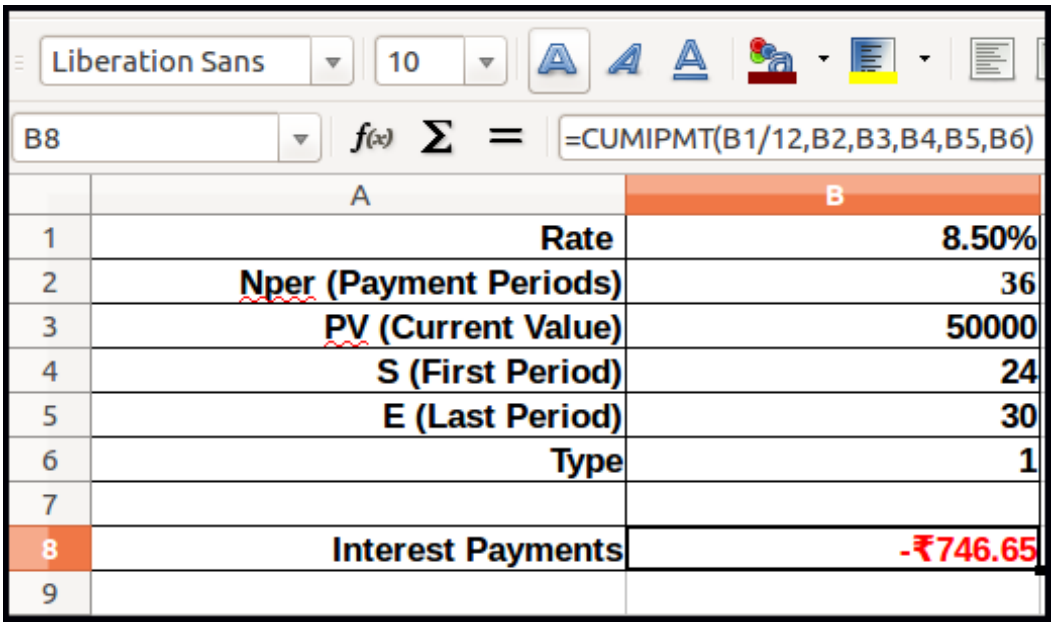
0 (zero) means Payment at the end of the period

1 means Payment at the beginning of the period.

Example :

What are the interest payments at a yearly interest rate of 8.5%, monthly payments for 3 years and a current cash value of ₹50,000. The start period is the 24th and the end period is the 30th period. The payment is due at the beginning of each period. See Figure 2.41

=CUMIPMT(B1/12,B2,B3,B4,B5,B6) Returns -₹746.65



	A	B
1	Rate	8.50%
2	Nper (Payment Periods)	36
3	PV (Current Value)	50000
4	S (First Period)	24
5	E (Last Period)	30
6	Type	1
7		
8	Interest Payments	-₹746.65
9		

Figure 2.41 Example of CUMIPMT

3. PV()

Returns the present value of an investment resulting from a series of regular payments. This function is used to calculate the amount of money needed to be invested at a fixed rate today, to receive a specific amount, over a specified number of periods.

Syntax : **=PV(Rate,NPER,Pmt,FV,Type)**

Rate defines the interest rate per period.

NPER is the total number of periods (payment period).

Pmt is the regular payment made per period.

FV (optional) defines the future value remaining after the final instalment has been made.

Type (optional) denotes due date for payments.

Example :

Raju opened a Recurring Deposit Scheme paying Rs.2500 per month for a period of 4 years with an interest rate of 8% per annum, and the payments are made at the beginning of the month.

Rate is the interest rate per period.

In the given example interest rate is 8% per annum and instalments are made the monthly payments, then the interest rate per month is 8%/12.

NPER is the total number of payment periods in an annuity.

In the given example, the period is 4 years that makes monthly payments, then period is 4*12 (or 48 months) .

Pmt is the periodical payment for an annuity.

We have to enter -2500 into the function as the pmt. If pmt is omitted, then fv must be included in the argument.

If PV is positive, PMT must be negative. Conversely, if you want PMT to be positive, simply make PV negative.

Fv is the future value, or a cash balance to attain after the last payment is made.

If Fv is omitted, it is assumed to be 0

Type Type = 1 means due at the beginning of a month and Type = 0

(default) means due at the end of the month.

=PV(8%/12,48,-2500,0,1) Returns ₹103087.48 (See Figure 2.42)

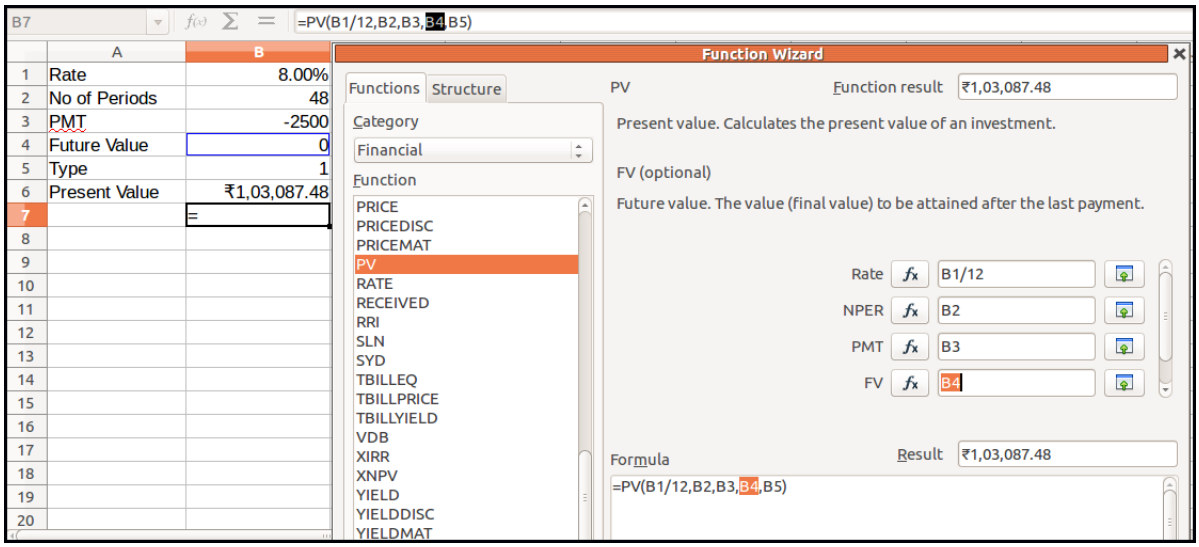


Figure 2.42 Examples of PV Function

4. PMT()

PMT function calculate the equal periodic payment for an annuity with constant interest rates.

Syntax : **=PMT(Rate,NPER,PV,FV,Type)**

- **Rate** is the periodic interest rate.
- **NPER** is the number of periods in which annuity is paid.
- **PV** is the present value (cash value) in a sequence of payments.
- **FV** (optional) is the desired value (future value) to be reached at the end of the periodic payments.
- **Type** (optional) is the due date for the periodic payments. Type=1 is payment at the beginning and Type=0 is payment at the end of each period.

Example :

Calculate the Monthly payment for a Loan of ₹25,000 availed by Raju Aravind from SBI with an interest rate @ 8% for a period of 3 years.

=PMT(8%/12,36,25000,0,0)

The periodic monthly payment is therefore ₹783.41 (See Figure 2.43)

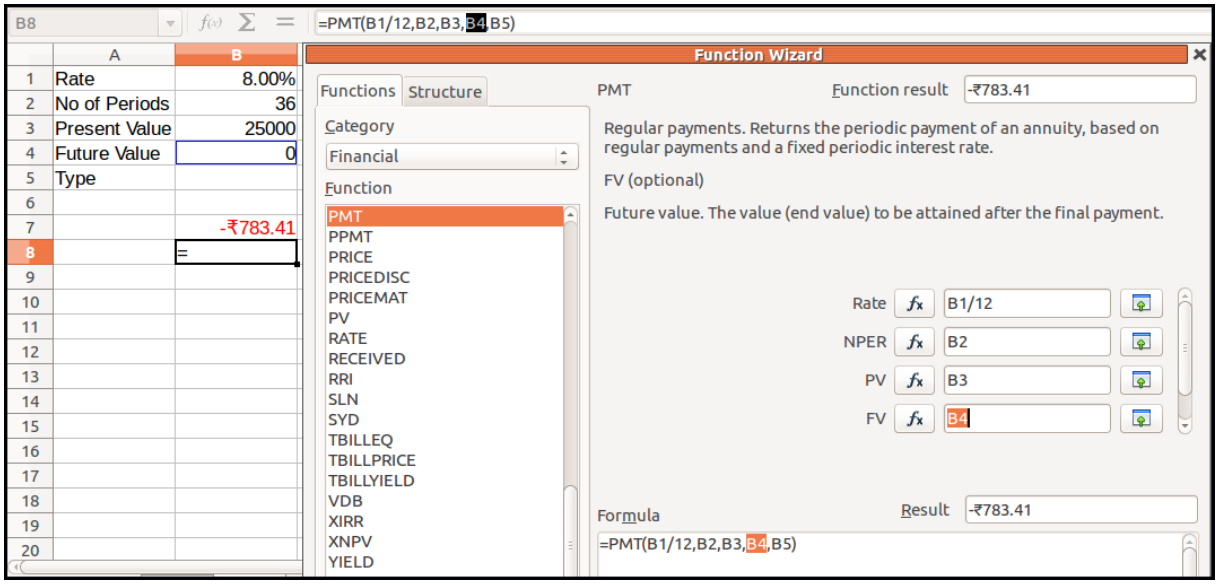


Figure 2.43 Examples of PMT Function

5. FV()

This function calculates the future value of an investment based on periodic, constant payment and a constant interest rate

Syntax : **=FV(RATE,NPER,PMT,PV,Type)**

Rate is the periodic interest rate.

NPer is the total number of periods (payment period).

Pmt is the annuity paid regularly per period.

PV (optional) is the (present) cash value of an investment.

Type (optional) defines whether the payment is due at the beginning or the end of a period.

Example : Anil Ramachandran has made a periodical payment of ₹750 with an interest rate at 4% for a period of 2 years. What is the Value at the end (Future Value). Under the following cases

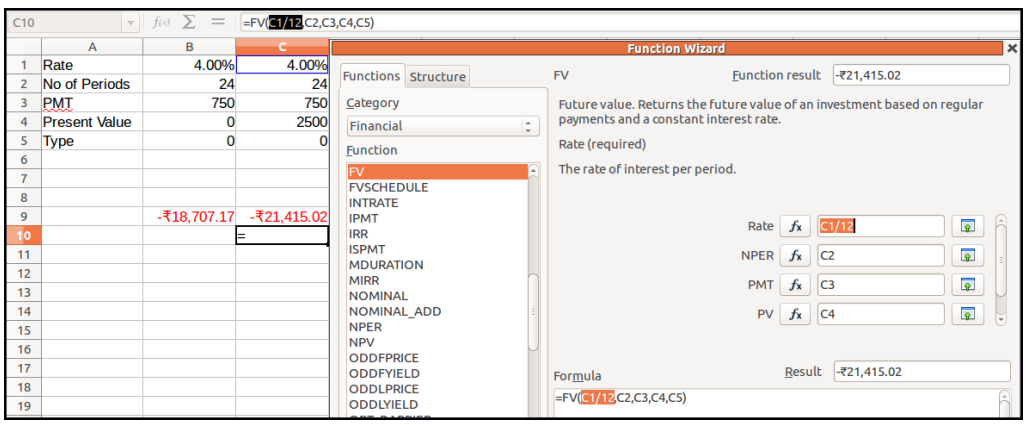


Figure 2.44 Examples of FV function

a) When the PV is assumed as Zero

=FV(4%/12,24,750,0,0) Returns Future value is ₹18707.17

b) When the PV is assumed as ₹2500

=FV(4%/12,24,750,2500,0) Returns Future value is ₹21415.02 (See Figure 2.44)

6.Rate()

Calculate the constant interest rate per period of an annuity.

Syntax **:** =RATE(NPER,PMT,PV,FV,Type,Guess)

NPER is the total number of periods, during which payments are made (payment period).

Pmt is the constant payment (annuity) paid during each period.

PV is the cash value in the sequence of payments.

FV (optional) is the future value, which is reached at the end of the periodic payments.

Type (optional) is the due date of the periodic payment, either at the beginning or at the end of a period.

Guess (optional) determines the estimated value of the interest with iterative calculation.

Example :

Asha Menon taken a loan of ₹500,000 from CAN Bank, agreed to pay ₹11,500 per month over a period of 5 years, and the monthly payments are to be made at the end of each month. Compute the rate of interest using Rate Function.

=RATE(60,-11500,500000,0,0)

Returns ₹1.12 (multiply by 12 to convert into Annual rate) (See Figure 2.45)

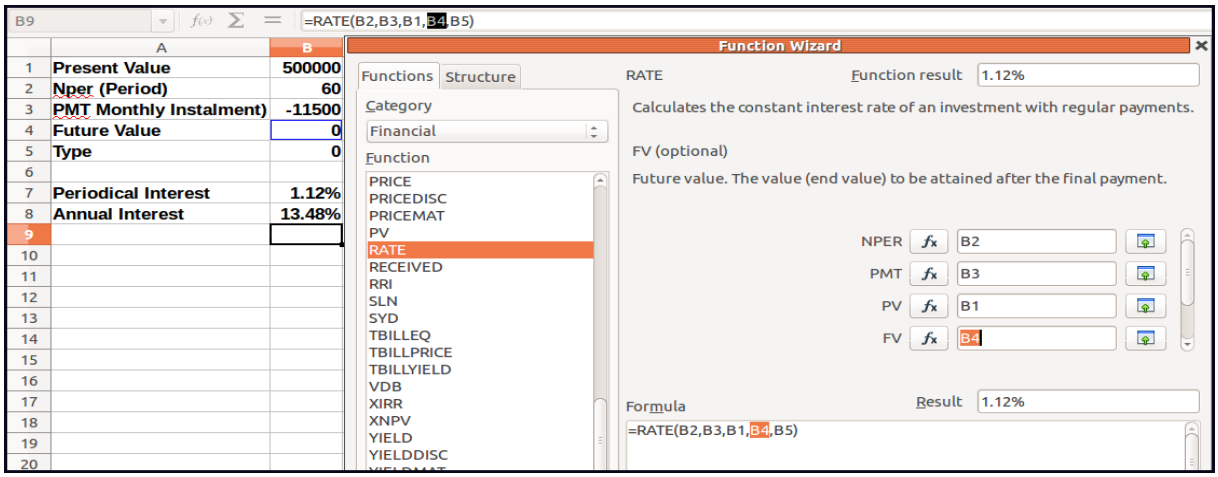


Figure 2.45 Examples of RATE functions

7.NPV()

Returns the present value of an investment based on a series of periodic cash flows and a

discount rate. To get the net present value, subtract the cost of the project (the initial cash flow at time zero) from the returned value.

***Syntax* : NPV(Rate,Value1,Value2,...)**

Rate is the discount rate for a period.

Value1;... are up to 30 values, which represent deposits or withdrawals.

Example:

Muhammed invested ₹5000 in a power project. Cash inflows from the project will be 800,950,1080,1220 and 1500 over the next 5years. Projects cost of capital is 2%

=NPV(2%,-5000,800,950,1080,1220,1500) Returns ₹196.88 (See figure 2.46)

Note: Outflow of cash treated as negative so all returrns are positive.

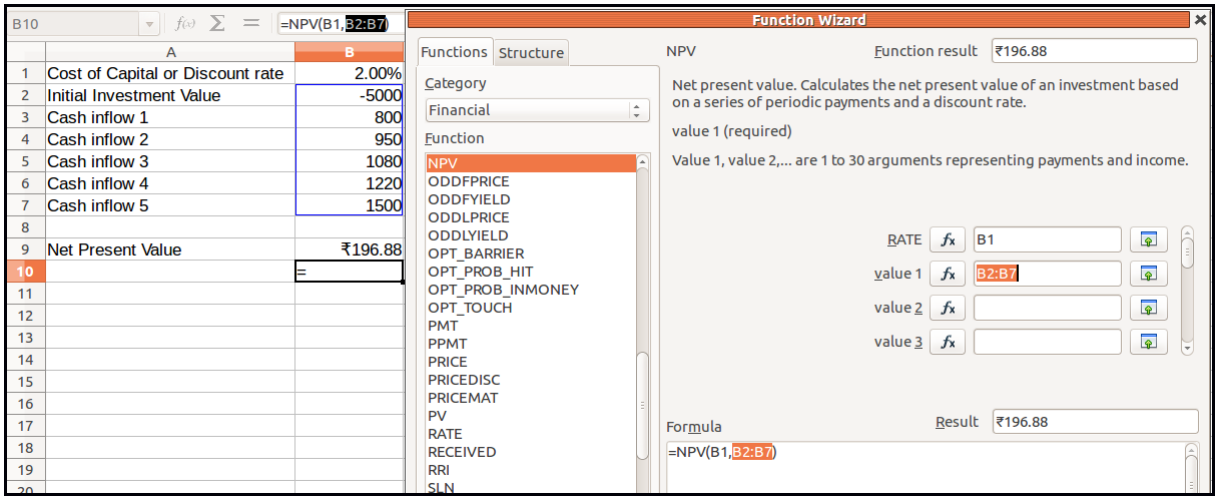


Figure 2.46 Examples of NPV functions

Check Yourself

Fill in the blanks:

1. _____ financial function is calculates the accrued interest of a security in the case of periodic payments.

2. Correct the following Syntax of ACCRINT

   =ACCRINT(par,first_interest, settlement, issue, frequency, rate,basis, calc_method)

3. The function which calculates the future value of an investment based on periodic, constant payment and a constant interest rate is called _____

4. _____ function calculates the constant interest rate per period of an annuity.

5. NPV Returns the _____ of an investment based on a series of periodic cash flows and a discount rate

TRUE/FALSE. If false correct them

1. ACCRINT function returns the cumulative interest paid between two periods.
2. The syntax of financial function PMT

   =PMT(Rate,PMT,NPER,PV,FV,Type)
3. PV Returns the present value of an investment resulting from a series of regular payments.
4. The syntax of financial function CUMIPMT

   =CUMIPMT(Rate,NPER,PV,S,E,Type)
5. FV function calculate the equal periodic payment for an annuity with constant interest rates.

Short Answer Type

1. Write down the syntax of any two financial functions
2. Briefly explain about the PV function and FV function
3. Describe about PMT function

DATA ENTRY, TEXT MANAGEMENT AND CELL FORMATTING

In any computerised business application, the basic requirement is to input data; which may be either for processing parameters or to update various data elements. In both the cases data should be correct, accurate and should be in proper format. This means that data should be validated, corrected and can be display in proper format.

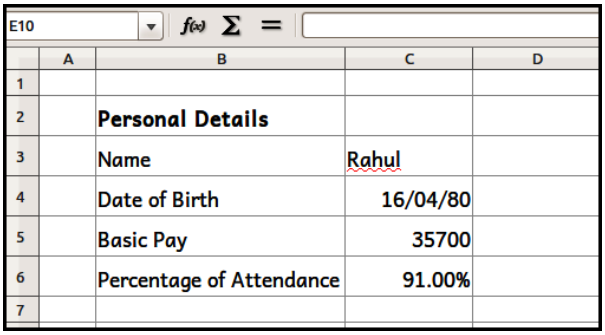


Figure 2.47 Default Alignment in Cell

By default in spreadsheet the numbers are right aligned and texts are left aligned. The spreadsheet can distinguish different types of numbers; recognise a date, a currency, or a percentage value or text etc. For example, if we type 16/04/1980 in a cell, spreadsheet will recognise it as a date and act accordingly. The software processes the data and generates the output; which should be in specific format. (see figure 2.47)

Data Entry

LibreOffice Calc facilitates fast data entry; and automatically repeats data or can fill data in different cells (column wise or row wise.) For example, if we repeatedly type the days of the week in different cells instead of that we could use the built-in data fill options to fill the different cells with the days automatically. Some of the methods for data entry are mentioned below:

The data fill options

The 'Fill' command can be used to fill data into worksheet cells. The Calc provides for entering data automatically to continue a series of numbers, number and text combinations, dates, or time periods, based on a pattern that we require. However, to fill quickly in several types of data series, we select cells and drag the fill handle (A Fill handle is the small black square in the lower-right corner of the selection. When we point to the fill handle, the pointer changes to + symbol.

For example, we want to enter data in A1:A10 starting value from 10 and in step of 10 we will get 10, 20, ... 100 by using drag option as shown in the Figure 2.48 and Figure 2.49 respectively.

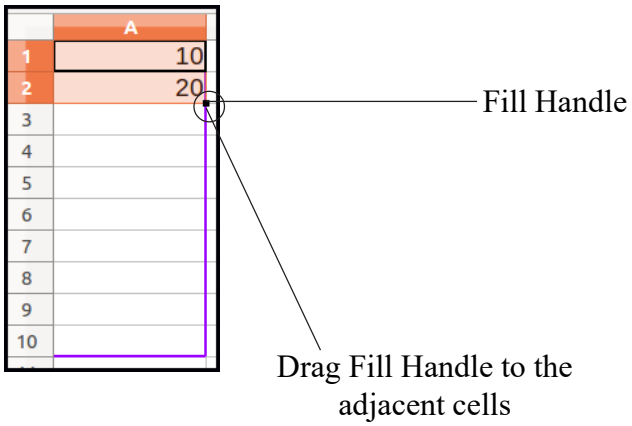


Figure 2.48 Fill Options

Option 1 Drag the fill handle to fill data into adjacent cells

(refer Figure 2.48)

- Select the cells that contain the data that we want to fill (A1:A2) into adjacent cells (A3:A10).
- Drag the fill handle across the cells that we want to fill.
- The data automatically fill to the dragged cells. (See Fig 2.48)

Option 2 fill the active cell with the contents of an adjacent cell

- Select an empty cell (A1) enter the value 10. Then select the range to fill (Select A1 to A10 for example.)
- From the Edit menu, click Fill, and then click on Series option (Fig. 2.49).

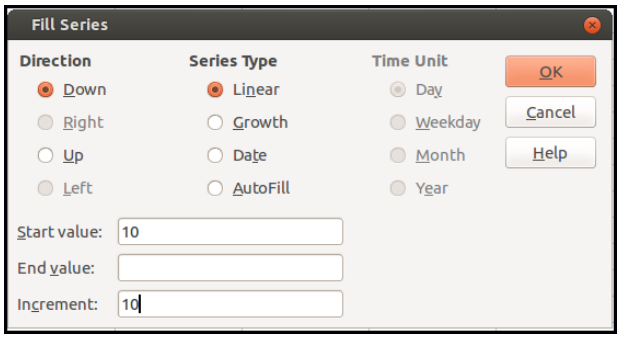


Figure 2.49 Fill Series wizard

- The option window provides direction 'Down', if the selected range in the sheet is downward. The ***Series Type*** will be 'Linear' by default. Give Start value and increment in the boxes given. In the given figure (Fig. 2.48) both entries are 10.

- Once we enter the all the options and click OK, we get data filled in the cells A2 to A10 as shown in the figure (Fig. 2.50).

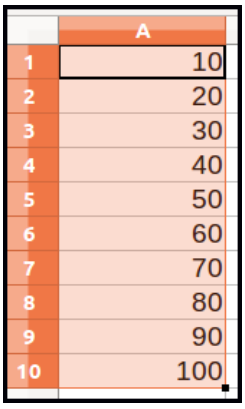


Figure 2.50
Values filled in an increment of 10

Observe another example for date data we can use fill handle. It is important to note that all the cells of the column or rows should be defined in (required) date data format using Format cells. In this example, we will enter date 24-11-2004 (or 24 Nov 2004) in cell A1 and 24-12-2004 (or 24 Dec 2004) in cell A2 (See Fig 2.51) and then drag fill handle from A3 to A10 after selecting the cells A1 and A2. Observe the result in the figure 2.52.

Import/Copy Data from other sources

We can use the following easiest method also for data entry. This method will transfer data into required cells by copying or importing and external file to Calc sheet. These data files may be either in text files or non-text files format. Text files can be directly read using a text editor. These files often have extension .txt but can have other extensions (such as .csv known as Comma Separated Values text file), easily read into Calc.

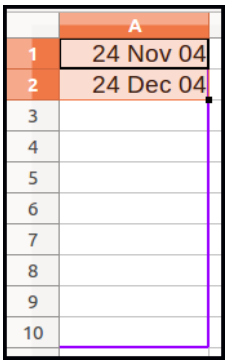


Figure 2.51

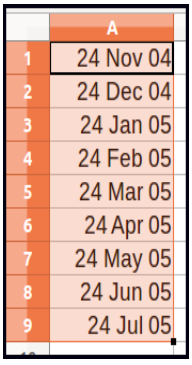


Figure 2.52

To import the data from a text file, following steps are important.

- Create data file using Text Editor (to get Text Editor; click on Applications → Accessories → Text Editor).

- A comma-separated data values in one line of this text file is a row in a spreadsheet and each entry, separated by a comma, is a column entry for that row.

- In the first line provides names for the columns of the spreadsheet.

- In the next line onward start entering the data separate by comma as per the names given in first line.

- It may possible that every data may not be of similar length but each data (even a blank data) should be separated by comma as per the names of the column.

- Open a new LibreOffice Calc.

- Click on 'Sheet from File' option from Insert menu. Select the text file or csv file with the help of dialogue box appeared and press OK button. 'Text Import' window appears (Figure 2.53).

- Ensure whether the Separator Option, here it is coma, ticked. A preview of the sheet can be seen in the bottom of 'Text Import' window. Then, press 'OK' button. 'Insert Sheet' window appears. Press 'OK' button from the appeared window and see the imported data in the sheet (Figure 2.54).

If you Saved the file in csv format, no need to import the file in to calc. Just open the file. It will open with the help of calc in sheet fom. Try it.

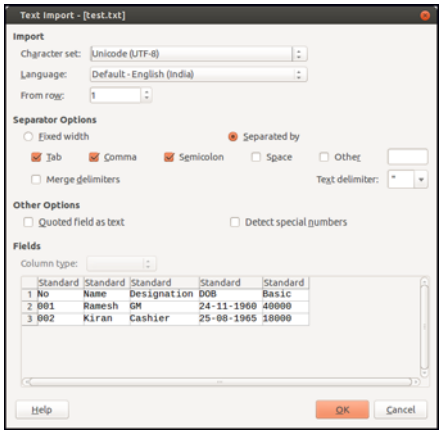


Figure 2.53 Text Import window

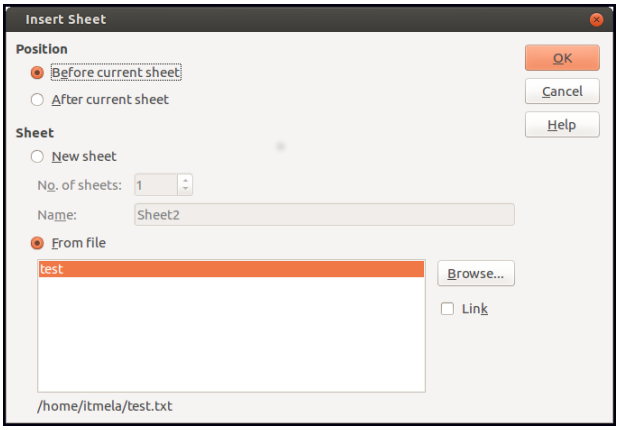


Figure 2.54 Insert Sheet window

- Each and every data from Text Editor file can be saved as an Calc data file. Calc worksheet consists of four types of data in cells: labels, values, date and formulas and data validation.
  - Labels (text) are descriptive data such as names, months and usually include alphabetic characters. Calc aligns text to the left side of the cell.
  - Values (numbers) are generally raw numbers or dates.

Whole value: If the data is a whole value, such as 34 or 5763, Calc aligns the data to the right side of the cell.

Value with a decimal: If the data is a decimal value, Calc aligns the data to the right side of the cell, including the decimal point, with the exception of a trailing 0. For example, if we enter 246.75, then 246.75 displays; such as 246.70, will display as 246.7. We can change the display appearance, column width, and alignment of data.

- Formulas are instructions for Calc to perform calculations.
- Date: If we enter a date, such 16/12, Dec 16, or 16 Dec, Calc automatically returns 16- Dec in the cell, but the Formula bar displays 16/12/2008. (The Date format is dependent to Country Specific Format selection).

Data Validation

Data validation is a feature to define restrictions on type of data entered into a cell. We can configure data validation rules for cells data that will not allow users to enter invalid data, There

may be warning messages when users tries to type wrong data in the cell. The messages also guide users to what input is expected for the cell, and instructions to correct any errors.

Data validation is invaluable because it is necessary that data must be accurate and consistent.

Validating

Set, edit validation criteria for data entry in a cell range like: whole number, decimal, list, date, time.

Validation criteria: we use validation criteria when we want to enable to involve a certain type of data in a particular cell or cells (whole number, date, decimal number, etc.). It is also possible to limit the input to the values that are in specified interval, or equal to a certain value, lesser/greater than certain value etc.

- Select the cells in which you want to change certain criteria

- Data → Validity

- When the dialog box opens, enter the desired settings

- Enter the input message and the error warning

To enter an input message, follow the steps as in the previous instructions, and then click on tab Input Help, enter the message in accordance with the criteria of evaluation. This message will appear when you highlight a cell for which the criteria is defined.

To enter an error warning, follow the steps as in the previous instructions, and then click on Error Alert and enter the requested data. This message will be displayed when you enter a value that does not meet the criteria.

Example :

Enter studetns detailes in a table (column heading as NAME OF STUDENTS, CLASS, SEX) using data validity

Solution:

Step 1: Enter column headings in A1:C1

Step 2: Enter the Name of students in A2 : A10

Step 3: Create a list of Class by using Data Validity

Step 4: Select the ranges from B2 : B10

Step 5: Select Validity from Data Menu

Step 6: Allow cell ranges from validity window and input Source the ranges from D2 : D4 (Commerce, Humanities,Science)

Step7: Select the ranges from C2 : C10

Step 8: Select Validity from Data Menu and Allow select List and enter Male, Female, Transgender

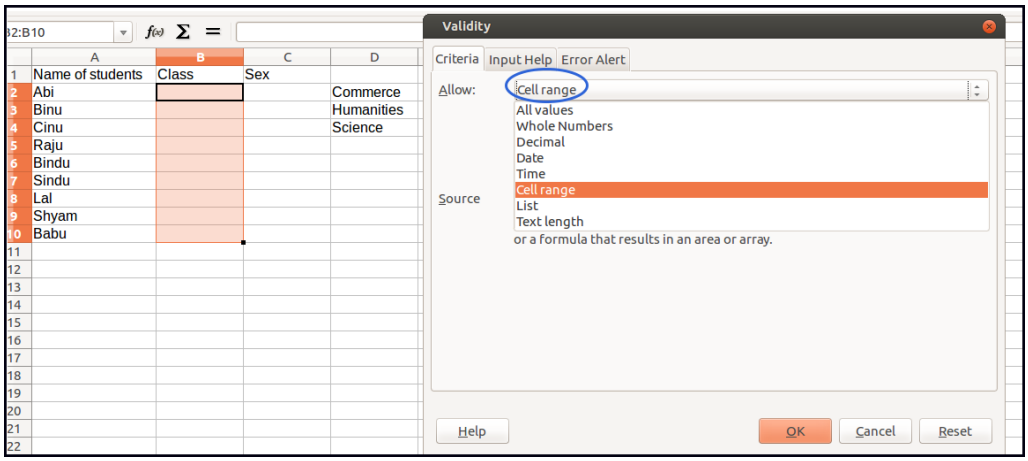


Figure 2.56

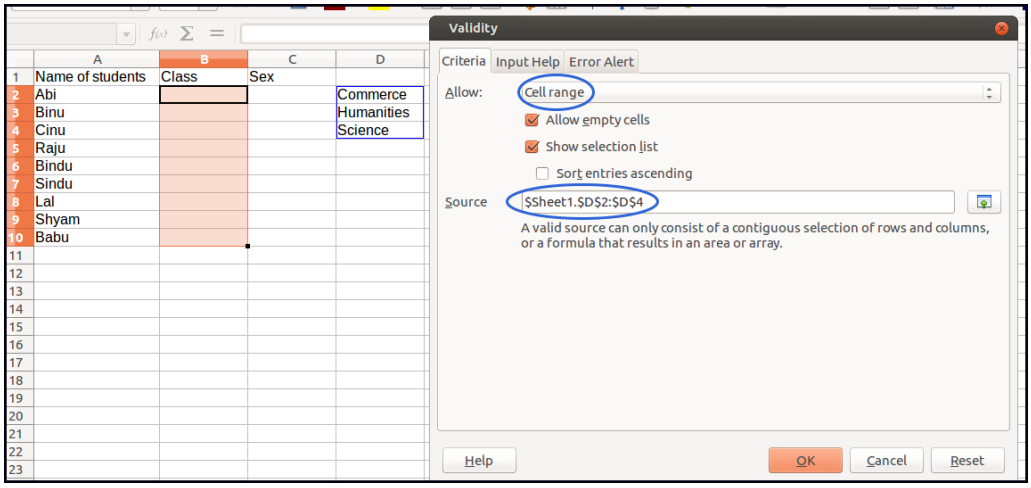


Figure 2.57

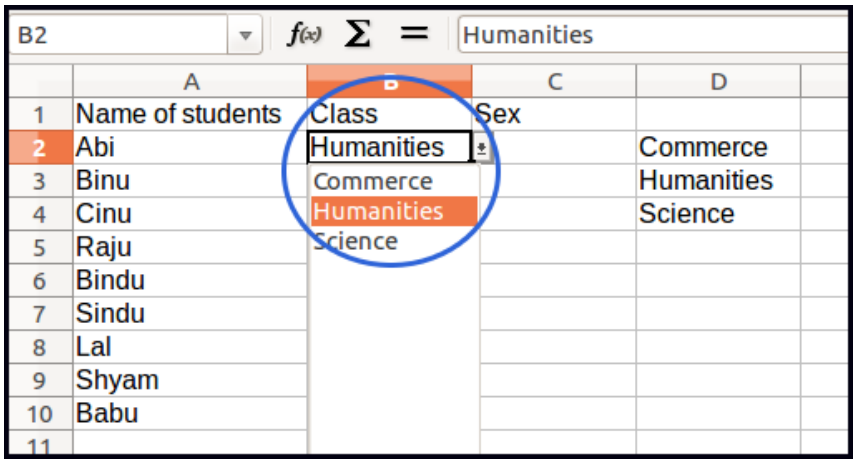


Figure 2.58

DATA FORMATTING

Formatting of spreadsheets makes easier to read and understand the important information (e.g. conditional formatting, number formatting, text and general spreadsheet formatting etc.). On the Ribbon there are several tools and shortcuts to format spreadsheets effectively. Some of them are described as follows:

Formatting Tools

The formatting tools for:

Number formatting – Number formatting includes adding per cent symbols (%), commas (,), decimal places, and currency signs (₹, Rs. etc), date, time, scientific values and as well as some special formats to a spreadsheet. The type of number format can be determined once the data is entered into the cell.

Here are some examples:

B4:E7 | f(x) Σ = | 40

	A	B	C	D	E	F	G	H
1	Sales for Home Appliences in Year 2008-2009							
2	(Figures in 000's)							
3		Q1	Q2	Q3	Q4	Actual p	Target	In %
4	ALOK	40	57	52	65	214	250	0.856
5	PRANAV	50	69	69	90	278	220	1.263636364
6	PRABODH	53	75	71	82	281	250	1.124
7	RAJENDRA	44	59	38	70	211	200	1.055
8	TOTAL					984	920	
9	Reported on							
10	16/04/09							

Figure 2.59

Currency : If we enter a financial value complete with the Rupee/currency sign and two decimal places, Calc assigns a Currency format to the cell along with the entry.

Percentages : If we enter a value representing a percentage as a whole number followed by the percent sign without any decimal places, Calc assigns to the cell the percentage format that follows this pattern along with the entry.

Dates : If we enter a date (dates are values, too) that follows one of the built-in Calc number formats, such as 16/04/2009 or 16 Apr 2009 the program assigns a Date format that follows the pattern of the date.

Example : Explaining below the steps to format range B4:E7 as currency with two decimal places using negative preset format. The sample unformatted worksheet is shown in the Figure 2.58

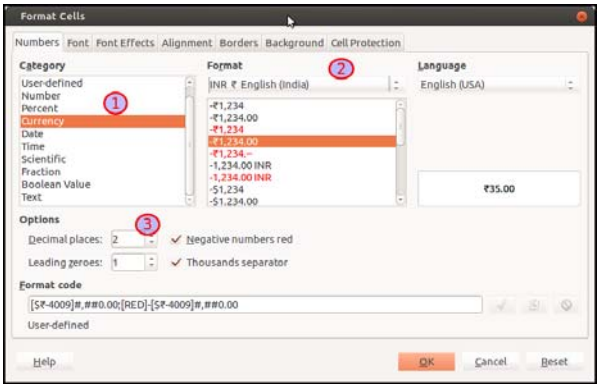


Figure 2.60

1. Select the range to format, i.e. B4:E7

2. From **Format** Menu bar, select Cell

3. Select Number tab in the '**Format Cell**' Dialog box **(Figure 2.59)**

4. Select Category as Currency. The right hand side of the box shows the different symbols of currencies.It also shows the preset for negative values either minus sign or red coloured number.

5. Select currency format as India- English for Indian Rupees with Rupee sign

6. Change the number of decimal places to two

7. Click **OK** to accept the formatting.

8. The output of these steps is shown in the Figure 2.60

9. Similarly if we select the category for the range H4:H7 as percentage with Decimal Places to 2 digits. The Figure 2.61 will show the effect of this change.

B4:E7 | f(x) Σ = | 40

	A	B	C	D	E	F	G	H
1	Sales for Home Appliences in Year 2008-2009							
2	(Figures in 000's)							
3		Q1	Q2	Q3	Q4	Actual p	Target	In %
4	ALOK	₹40.00	₹57.00	₹52.00	₹65.00	214	250	0.856
5	PRANAV	₹50.00	₹69.00	₹69.00	₹90.00	278	220	1.263636364
6	PRABODH	₹53.00	₹75.00	₹71.00	₹82.00	281	250	1.124
7	RAJENDRA	₹44.00	₹59.00	₹38.00	₹70.00	211	200	1.055
8	TOTAL					984	920	
9	Reported on							
10	16/04/09							

Figure 2.61

H4:H8 | f(x) Σ = | =F4/G4

	A	B	C	D	E	F	G	H
1	Sales for Home Appliences in Year 2008-2009							
2	(Figures in 000's)							
3		Q1	Q2	Q3	Q4	Actual p	Target	In %
4	ALOK	₹40.00	₹57.00	₹52.00	₹65.00	214	250	85.60%
5	PRANAV	₹50.00	₹69.00	₹69.00	₹90.00	278	220	126.36%
6	PRABODH	₹53.00	₹75.00	₹71.00	₹82.00	281	250	112.40%
7	RAJENDRA	₹44.00	₹59.00	₹38.00	₹70.00	211	200	105.50%
8	TOTAL					984	920	429.86%
9	Reported on							
10	16/04/09							

Figure 2.62

Changing cell colours

Select the range to format (e.g. B4:E7 – Figure 2.60)

From the Toolbar, select Format and click on cells to display Format Cells dialog box and

choose Background tab. Background tab provide Background Colour for the cell (or cell range to fill). Select the desired colour from the Colour pallets. Click **OK** and we see the result as given in Figure 2.63 and 2.64.

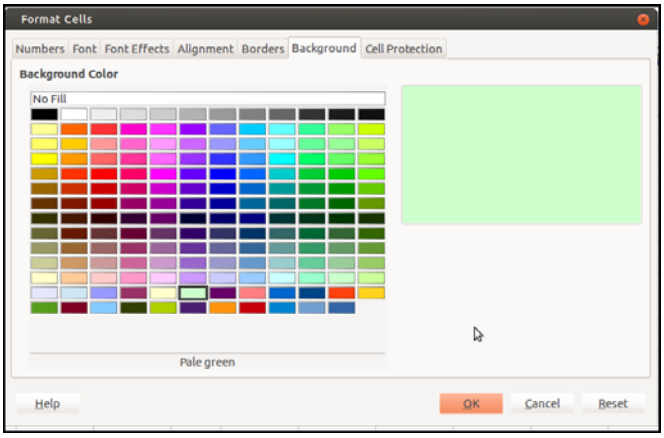


Figure 2.63

	A	B	C	D	E	F	G	H
1	Sales for Home Appliences in Year 2008-2009							
2	(Figures in 000's)							
3		Q1	Q2	Q3	Q4	Actual p	Target	In %
4	ALOK	₹40.00	₹57.00	₹52.00	₹65.00	214	250	85.60%
5	PRANAV	₹50.00	₹69.00	₹69.00	₹90.00	278	220	126.36%
6	PRABODH	₹53.00	₹75.00	₹71.00	₹82.00	281	250	112.40%
7	RAJENDRA	₹44.00	₹59.00	₹38.00	₹70.00	211	200	105.50%
8	TOTAL					984	920	429.86%
9	Reported on							
10	16/04/09							
11								

Figure 2.64

Adding text formatting

The text formatting is mostly required for presentation of final output and aesthetically reading of the information. There may be display of the text or numbers in different Fonts; Aligning in Cells,Changing Colour of Fonts, Merging of cell ranges to accommodate data etc. (Figure 2.64 &2.65) When we insert a Text , it looks quite plain. We can format text to make them stand out on the worksheet; using the options from the Tool bar; on Insert Tab with the Text dialog box. We can use these drawing tools in dressing them up with special formatting commands (Figure 2.65). These tools allow us to add decorative and explanatory material to worksheet and chart.

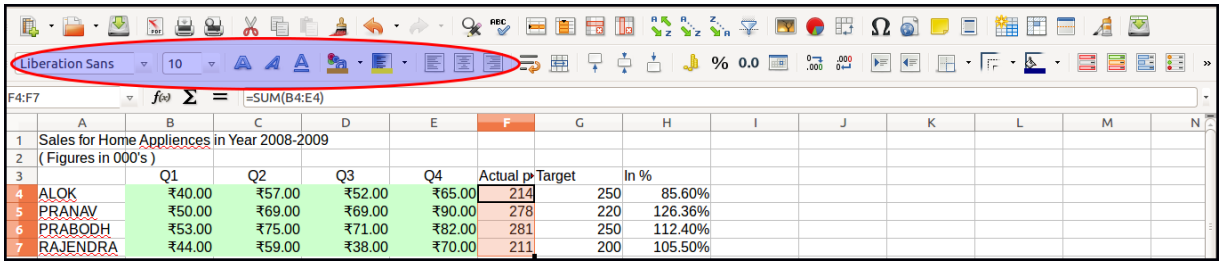


Figure 2.65

Create a custom cell border (Figure 2.64 Figure 2.65)

We can create a cell style that includes a custom border, and can apply to that cell style when we want to display the custom border around selected cells.

1. Select the properties option on the Side bar (right).

2. Click Cell Appearence.

3. We can modify cell border properties.

Conditional Formatting

Using the menu command **Format → Conditional formatting**, the dialogue allows you to define conditions per cell, which must be met in order for the selected cells to have a particular format. To apply conditional formatting, Auto Calculate must be enabled. Choose **Tools → Cell Contents → AutoCalculate** (you see a check mark next to the command when AutoCalculate is enabled). With conditional formatting, for example, highlight the totals that exceed the average value of all totals. If the totals change, the formatting changes correspondingly, without having to apply other styles manually.

To Define the Conditions

1. Select the cells to which you want to apply a conditional style.

2. Choose **Format → Conditional Formatting**.

3. Enter the condition(s) into the dialog box.

To understand the Conditional Formatting consider the following Example (Figure 2.66).

	A	B	C	D	E	F
1						
2	Sales for Home Appliances Year 2015-16					
3		(Figures in 000's)				
4						
5		Q1	Q2	Q3	Q4	
6	ALOK	40	57	52	65	
7	PRANAV	50	69	69	90	
8	PRABOSH	53	75	71	82	
9	RAJENDRA	44	59	38	70	
10						
11						

Figure 2.66 Data for conditional formatting

If we want to find the persons who cover the target, ie. which is more than 60000, conditional formatting helps us.

Follow the steps

1. Select the range or cells (B6:E9).
2. Select Format (in the menu bar) .....Conditional formatting ..... Condition.
3. In the conditional formatting window give condition ***cell value is greater than 60*** (we can select the **greater than** from the drop down menu and type **60** in the next cell).
4. In apply style select **new style** from the drop down menu and and give the font style or back ground color to differentiate the cell which have vale more than **60.**

Output is follows

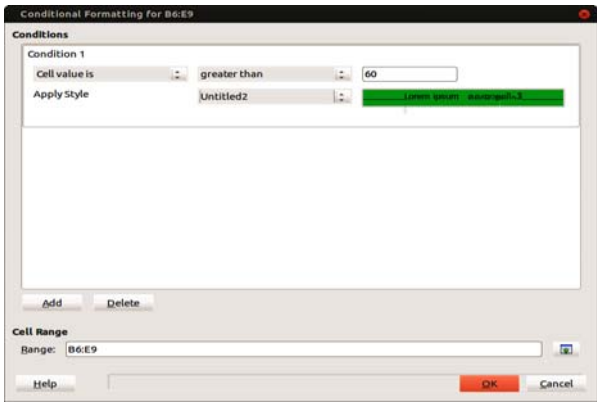


Figure 2.67 Conditional Formatting

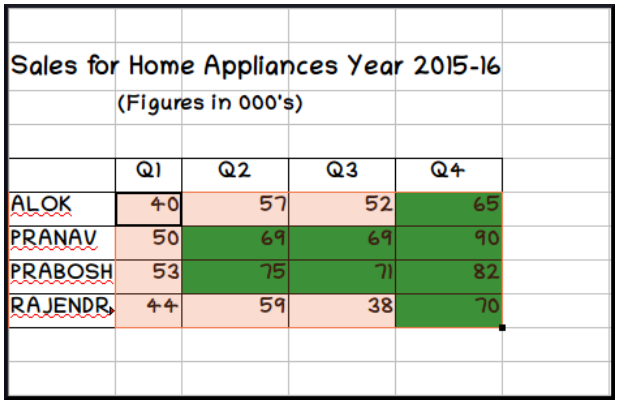


Figure 2.68 Conditional Formatting

Changing the alignment of data in cells

There are several options to change the alignment of text (data) in cells.

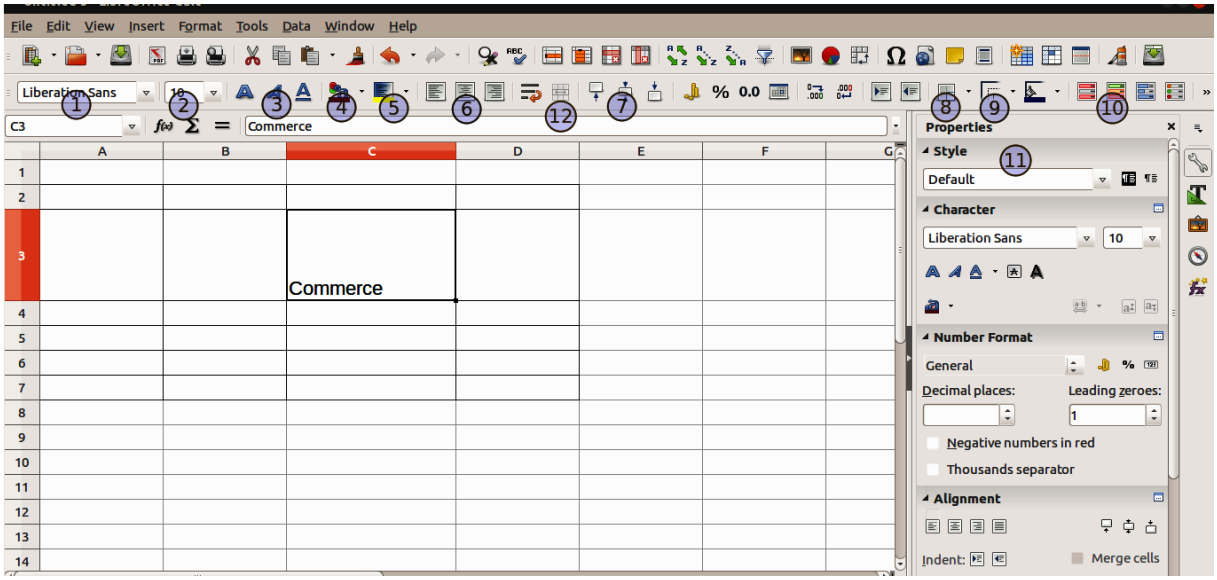


Figure 2.69 Alignment of text

Text formatting

Refer the above figure 2.69

We can format the text in the cell. Select the text in the cell by clicking and dragging it to the other end.

The font style can be changed by selecting appropriate font name (viz. Arial, Chilanka, Liberation Sans, ...) from the field indicated by **(1)**.

we can change the size of the font (10,11,12,.....32, 36, ...) by selecting the field **(2)**. If we need bold letters, Italics or underlined choose from the field **(3)**. The color of the letter can be changed using the field **(4).** The background color of the cell can be changed selecting suitable color from the field under the field **(5)**. The Text alignment in the cell either it is left, right, center or justify choosing the appropriate form the field under **(6).** The position in the cell ie. Top, center or bottom of the cell (vertical alignment can be selected under the field **(7).** If want to give boarders to the cell we can choose the field **(8)** and different types of broader is given in field **(9)**.

{These actions can also be given in Property window appears on the right of the worksheet. Another way is to select the Format option in the menu bar and then select the **cell** in the dropdown menu}

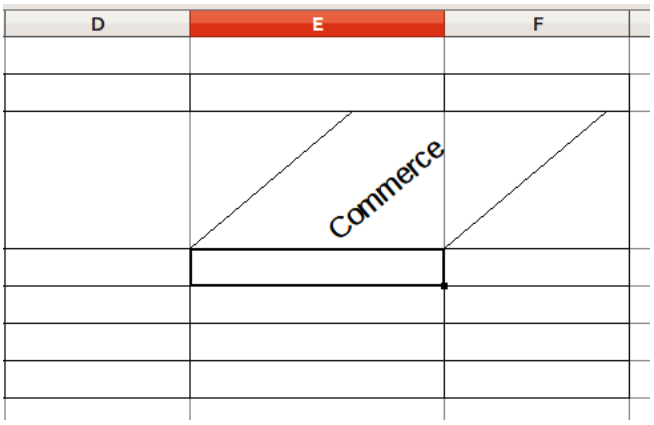


Figure 2.70 Text Formatting

The direction of the text or the orientation of the text in the cell (as shown in the figure) can be changed by the choosing Format.....Cell..... in the menu bar, will open the *format cells window* then select the alignment tab. Then give the angle in degrees (viz 20 or 25..as required).

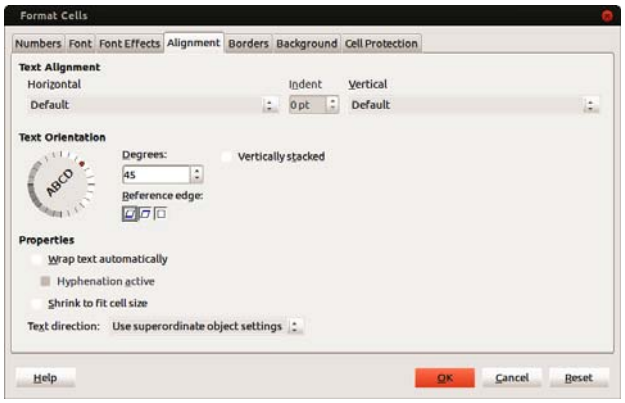


Figure 2.71 Formatting

Merging a range of Cells

Merged cells are a single cell that is created by combining two or more selected cells. The cell reference for a merged cell is the upper-left cell in the original selected range. When two or more adjacent horizontal or vertical cells are merged, the cells become one large cell and displayed across multiple columns or rows. The contents of one appear in the centre of the merged cell.

1. Select two or more adjacent cells that we want to merge.
2. Click Merge and Centre. (The filed under (12) in figure 2.72*)*. It will activated only when two or cells are selected.
3. The cells will be merged in a row or column, and the cell contents will be centered in the merged cell. To merge cells without centering, click the arrow next to Merge and Centre, and then click Merge Across or Merge Cells. The cell address of merge cells will be the address of lower active cell.
4. To change the text alignment in the merged cell, select the cell; click any of the alignment buttons in the Alignment group.

Split a merged cell

1. Select the merged cell.
2. When we select a merged cell, the Merge and Centre button also appears selected in the field under (12) in figure 2.72.
3. To split the merged cell, click Merge and Centre. The contents of the merged cell will appear in the upper-left cell of the range of split cells.

Formatting a Table

There are predefined table styles (or quick styles) that we can use to quickly format a table.

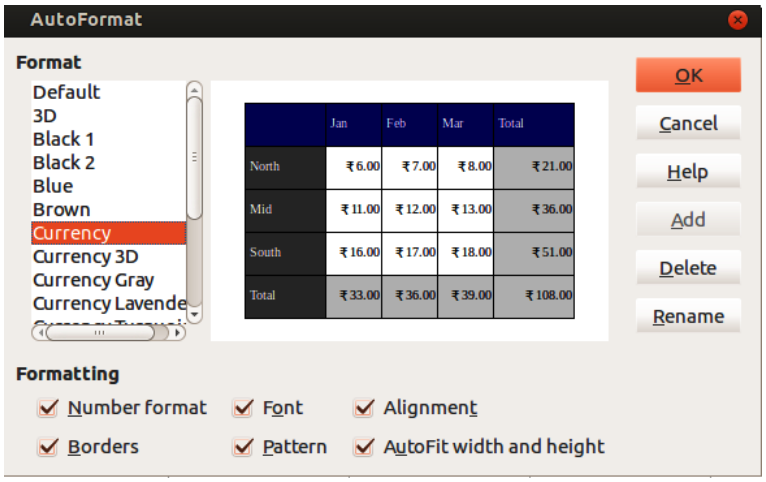


Figure 2.72 Formatting Table

Select cells where the Table has to inserted. Then select the Fomat option in the menu bar, in the dropdown menu select the auto format. We can select the predefined table format.

To choose a table style to create a table

1. On the worksheet, select a range of cells that we want to quickly format as a table.

2. On the menu bar Select Format and then select Auto format in the dropdown menu.

3. Select the predefined Table format

Adding Headers and Footers to Calc Spreadsheet

In spreadsheet, headers and footers are lines of text that are printed at the top (header) and bottom (footer) of each page the menu bar. They contain descriptive text such as titles, dates, and/or page numbers and are used to add information to a printed spreadsheet.

The headers and footers options are available under the **Insert** in the menu bar.

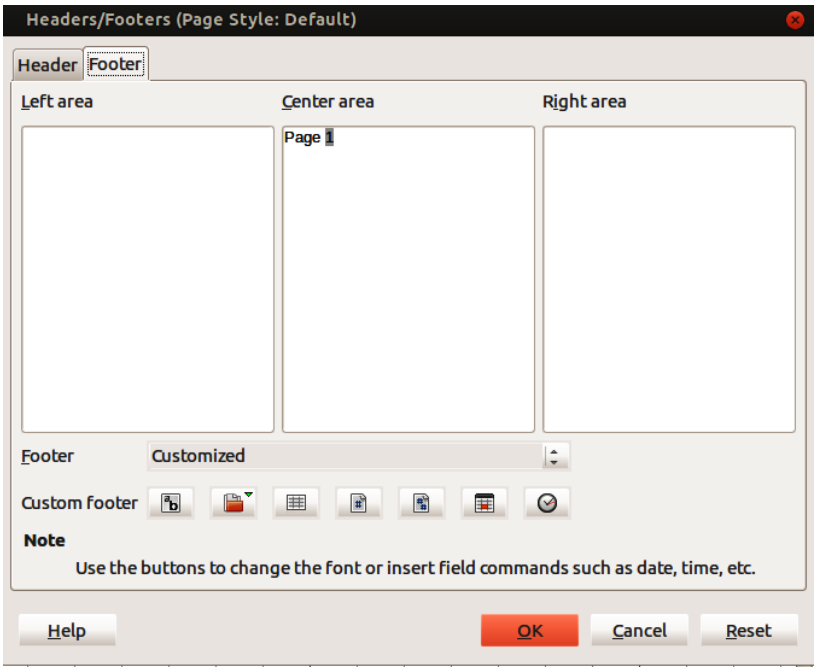


Figure 2.73 Header and Footer

If the footer is needed in the center select the center area and the Type the word to display in the footer. If page number is to be displayed type Page 1.

OUTPUT REPORTS

Page Setup

We can customise our output Report by editing the page setup option.

It can be accessed by

Format (in the file menu) → **Page** opens the *Page Style* window in the page tab

1. Paper format

The configuration of page or the type of the paper using for the output report can be selected from the Format option (A4, A3, B3...). If we need custom paper size select the width and height of the paper which we want to use. For example if we want a paper of size width 22 cm and height 25 cm, give that details in the appropriate cells.

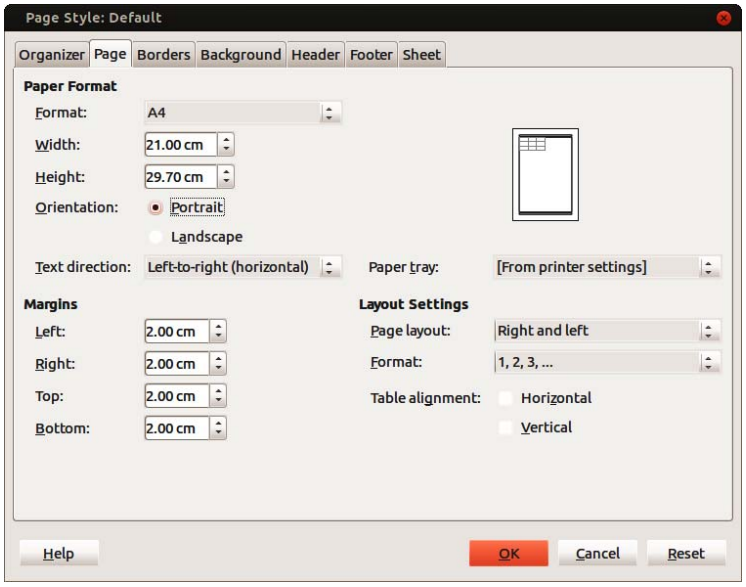


Figure 2.74 Page Style

2. Orientation is another area to consider.

We can select the orientation as Portrait or Landscape as we needed.

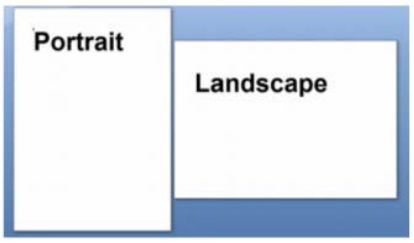


Figure 2.75 Page Orientation

We can change the Margins, that is defining the area to print, by changing the Left, Right, Top and bottom margins.

Print Out

We can print entire or partial worksheets and workbooks, one at a time, or several at once.

(a) Print a partial or entire worksheet or workbook.

(b) Print several worksheets at once.

(c) Print several workbooks at once.

(d) Print a LibreOffice Calc table.

(e) Print a workbook to a file.

(f) Print a graphic Charts and Pivot Tables.

File → **Print** or just clicking the print icon given.

Before printing we have to verify print preview which gives an idea about how the printout may come. The modification can be done by this verification.

Defining the Print Area

By default, LibreOffice Calc prints all data on the current worksheet but for specific and formatted print we have to define print area from the print Lay out from file menu.

Define The Print Area using Dialog Box Option

1. Select the Print Layout command from File menu .
2. In the print preview set the margins if needed
3. click on the print icon which opens the Print window
4. Select the printer which is to be used for printing
5. Choose from the Ranges and Copies, according to our need Select All Sheets or Selected Sheets or selected cells and Number of copies we needed.
6. We can select the pages to print from the option from which print.

Then press **OK**

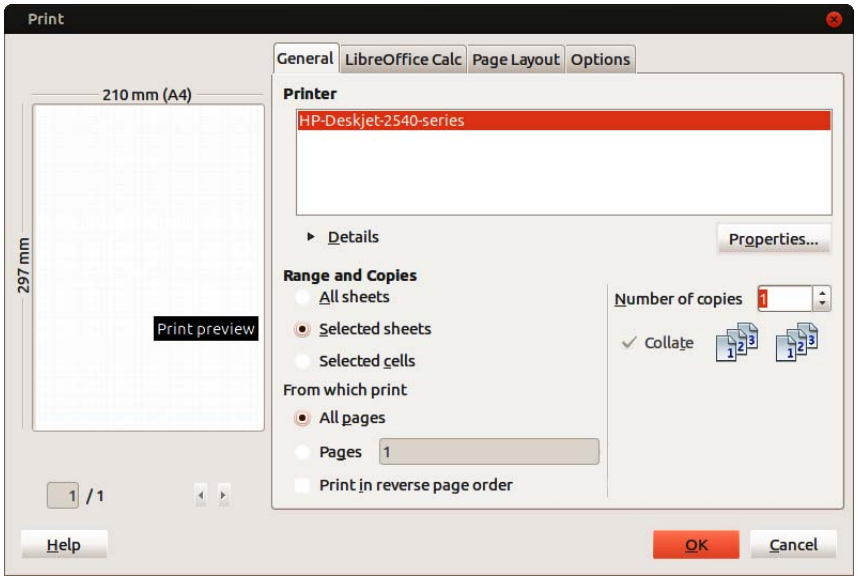


Figure 2.76 Defining Print Area

Check Yourself

1. By default in LibreOffice Calc ,the alignment of numbers are.....
2. By default in LibreOffice calc ,the alignment of texts are.....
3. Describe the different methods for data entry is possible in LibreOffice Calc.
4. The Fill option is available in .....menu in LibreOffice Calc.
5. The extension of Comma Separated Values text file is.....
6. Restriction of data entered in to a cell through .....
7. Write the procedure for importing data from external sources
8. How to set print area in LibreOffice Calc ?
9. Conditional formatting is available in .....menu in LibreOffice Calc

ONE-VARIABLE DATA TABLE

The One-Variable Data Table allows you to identify a single decision variable in your model and see how changing the values for that variable affect the values calculated by one or more formulas in your model.

Example : Blue Star Detergent Company have five Sales men, their sales in March 2017 are given below :

Name of Sales men	Amount of Sales
SANDEEP	250000
YASIR	300000
SUBAIR	240000
ALEX	340000
SHIBU	180000

Table 2.11

They got 10% commission in addition to their salary. Find out sales men's commission by using ONE VARIABLE TABLE

Solution:

Step 1 : Enter Input data (Rate of commission in B3 and sales amount in B4)

Step 2 : Set the formula in B5 to ascertain commission on sales basis

(multiply sales with Rate of commission)

=B3* B4 in B5

Step 3 : Enter the Salesmen name from D3 to D7 and Sales amount in E3 to E7

Step 4 : Select the range from E3 to F7

Step 5 : From **Data** menu select **Multiple Operations** and enter the cell address of formula ($B$5) and enter column input cell ($B$4).

Name of Sales men	Amount of Sales
SANDEEP	250000
YASIR	300000
SUBAIR	240000
ALEX	340000
SHIBU	180000

Table 2.12

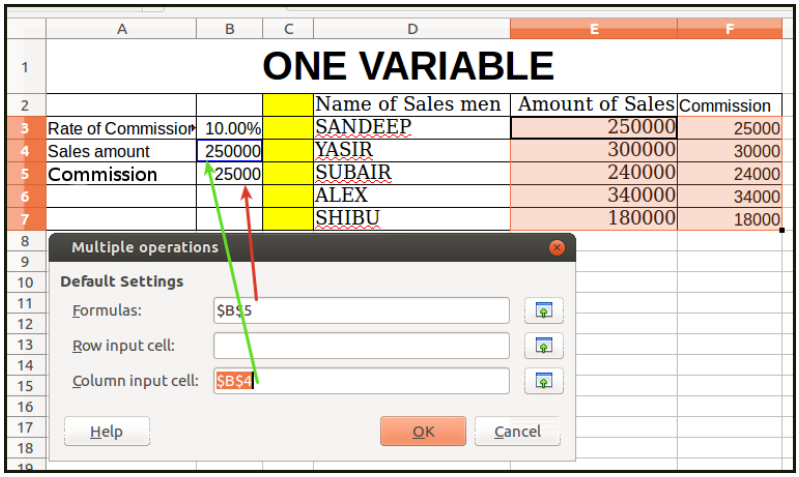


Figure 2.77 One variable

TWO VARIABLE DATA TABLE

Two-Variable Data Table works similar to the One-Variable Data Table. However, which specify two decision variables and a variety of inputs and only a single formula.

Example: Blue Star Detergent Company have five Sales men, their sales in March 2017 are given below:

Name of Sales men	Amount of Sales
SANDEEP	250000
YASIR	300000
SUBAIR	240000
ALEX	340000
SHIBU	180000

Table 2.12

Calculate the sales men's Commission in different rates (10%,11%,12%,13%,14%)

***Solution*:**

Step 1 : Enter Input data (Rate of commission in B4 and sales amount in B5)

Step 2 : Set the formula in B6 to ascertain commission on sales basis

(multiply sales with Rate of commission)

=B4* B5 in B6

Step 3 : Enter the Salesmen name from D4 to D8 ,Sales amount in E4 to E8 and Rate of Commission in F3 to J3

Step 4 : Select the range from E3 to J8

Step 5 : From **Data** menu select **Multiple Operations** and enter the cell address of formula ($B$6), enter row input cell ($B$4) and enter column input cell ($B$5).

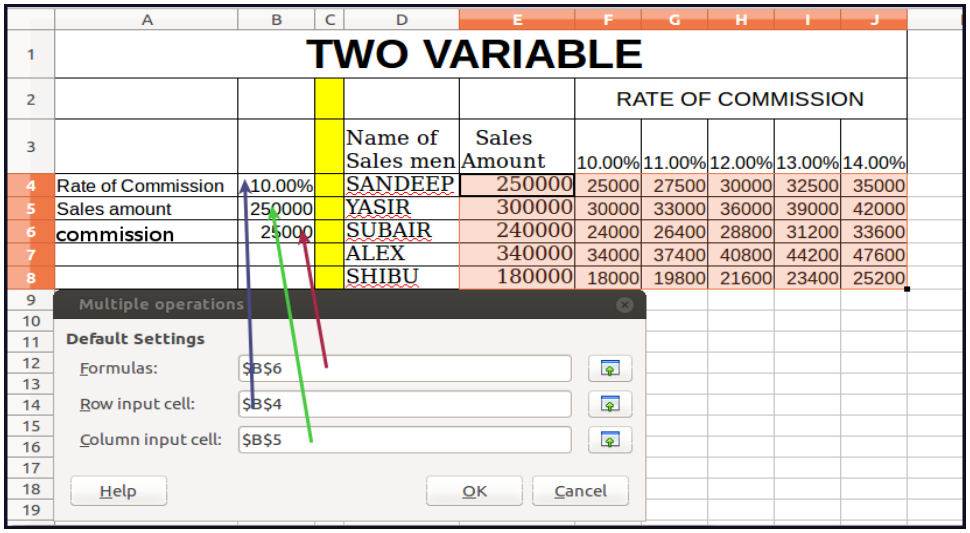


Figure 2.78 Variable

Pivot Table

Pivot table is a special type of table that allows easy compression, analysing and presenting data. It allow us to create a cross tabulation summary.

Example:

Raju sold Home Appliance in a week from his shop, on the basis of this transaction prepare a pivot table

ITEMS	BRAND	SALES PRICE
COLOUR TV	SAMSUNG	25000
WASHING MACHINE	LG	22000
REFREGATER	WHIRLPOOL	24000
WASHING MACHINE	LG	20000
COLOUR TV	WHIRLPOOL	32000
REFREGATER	SAMSUNG	26000
WASHING MACHINE	SAMSUNG	21000
COLOUR TV	LG	19000

Table 2.14

Solution:

Step 1: Input the entire data into cells

Step 2 : Select the entire data to create the pivot table

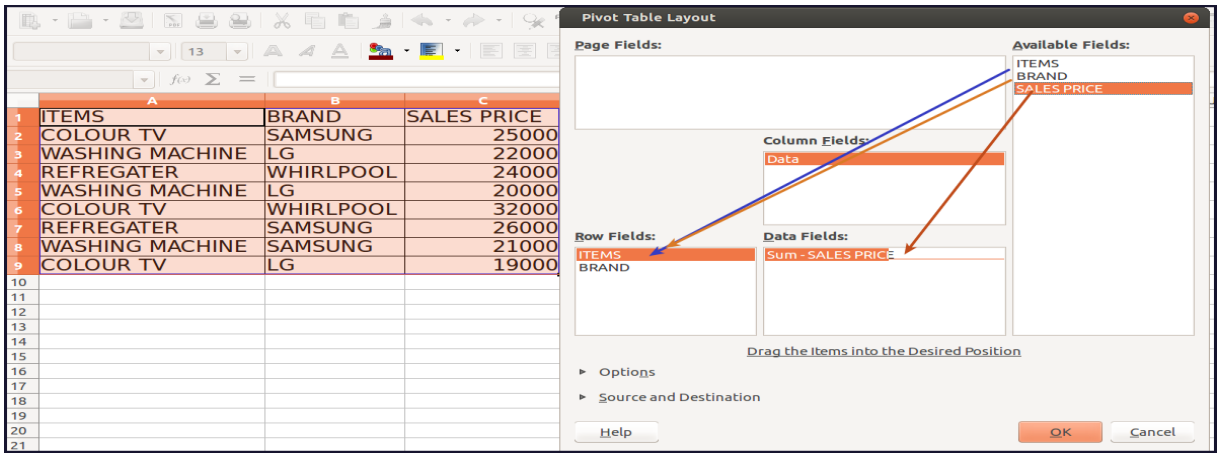
Step 3 : Select **Pivot Table** from **Data** menu and then click on Create

Step 4 : In the Pivot Table Layout, drag the available fields to the row field and data field (only numerical values) in the desired position.

Step 5 : By clicking on Source and Destination tab, we can select the Destination of the pivot table, either in a new sheet or in a desired cell (Default destination is new sheet)

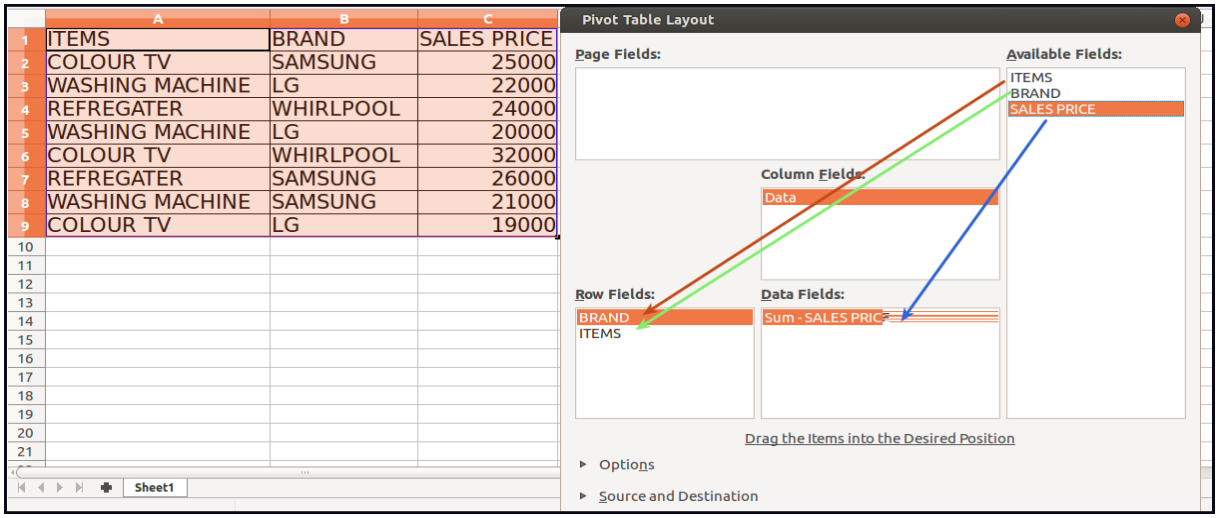
	A	B	C	D	E	F	G
1	ITEMS	BRAND	SALES PRICE		ITEMS	BRAND	Sum - SALES PRICE
2	COLOUR TV	SAMSUNG	25000		COLOUR TV	LG	19000
3	WASHING MACHINE	LG	22000			SAMSUNG	25000
4	REFREGATER	WHIRLPOOL	24000			WHIRLPOOL	32000
5	WASHING MACHINE	LG	20000		REFREGATER	SAMSUNG	26000
6	COLOUR TV	WHIRLPOOL	32000			WHIRLPOOL	24000
7	REFREGATER	SAMSUNG	26000		WASHING MACHIN	LG	42000
8	WASHING MACHINE	SAMSUNG	21000			SAMSUNG	21000
9	COLOUR TV	LG	19000		Total Result		189000
10							

Figure 2.79 Pivot Table



	A	B	C
1	ITEMS	BRAND	SALES PRICE
2	COLOUR TV	SAMSUNG	25000
3	WASHING MACHINE	LG	22000
4	REFREGATER	WHIRLPOOL	24000
5	WASHING MACHINE	LG	20000
6	COLOUR TV	WHIRLPOOL	32000
7	REFREGATER	SAMSUNG	26000
8	WASHING MACHINE	SAMSUNG	21000
9	COLOUR TV	LG	19000

Figure 2.80 Pivot Table



	A	B	C
1	ITEMS	BRAND	SALES PRICE
2	COLOUR TV	SAMSUNG	25000
3	WASHING MACHINE	LG	22000
4	REFREGATER	WHIRLPOOL	24000
5	WASHING MACHINE	LG	20000
6	COLOUR TV	WHIRLPOOL	32000
7	REFREGATER	SAMSUNG	26000
8	WASHING MACHINE	SAMSUNG	21000
9	COLOUR TV	LG	19000

Figure 2.81 Itemwise Report

	A	B	C	D	E	F	G
1	ITEMS	BRAND	SALES PRICE		BRAND	ITEMS	Sum - SALES PRICE
2	COLOUR TV	SAMSUNG	25000		LG	COLOUR TV	19000
3	WASHING MACHINE	LG	22000			WASHING MACHINE	42000
4	REFREGATER	WHIRLPOOL	24000		SAMSUNG	COLOUR TV	25000
5	WASHING MACHINE	LG	20000			REFREGATER	26000
6	COLOUR TV	WHIRLPOOL	32000			WASHING MACHINE	21000
7	REFREGATER	SAMSUNG	26000		WHIRLPOOL	COLOUR TV	32000
8	WASHING MACHINE	SAMSUNG	21000			REFREGATER	24000
9	COLOUR TV	LG	19000		Total Result		189000

Figure 2.82 Brand wise Report

COMMON ERROR CODES(MESSAGES) IN LIBREOFFICE CALC

LibreOffice Calc provides some message for errors of - miscalculation, incorrect use of functions, invalid cell references and values, and other user initiated mistakes. The error messages may be displayed within the cell that contains the error on the status bar (figure 2.84) or in both, depending on the type of error. Error messages display in a formula's cell or in the Function Wizard instead of the result.

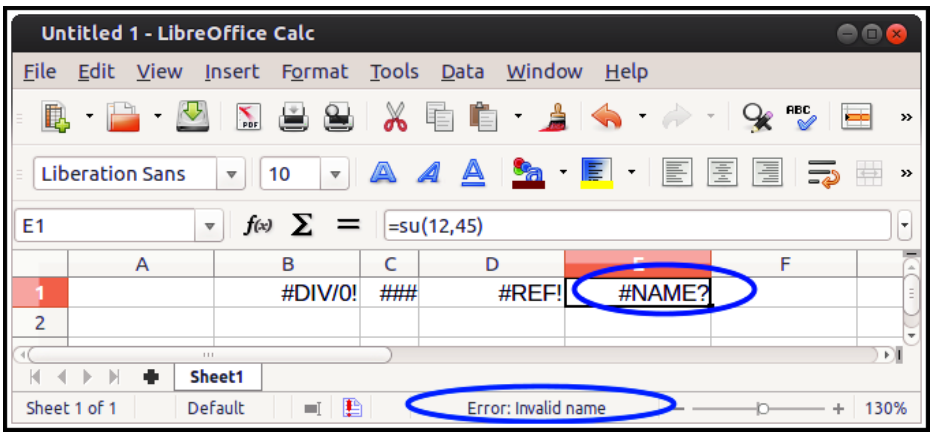


Figure 2.83 Error codes displayed within cells and error message displayed in the status bar

An error message for a formula is usually a three-digit number from 501 to 527, or sometimes an unhelpful piece of text such as #NAME?, #REF, or #VALUE. The error message /code appears in the cell and a brief explanation of the error on the right side of the status bar. Some of the error messages/codes are discussed below -

1. ### Error

This error occur when a numerical value entered in a column is not enough to display the contents. This is not really an error value, so there is no corresponding numerical error code. See Figure 2.84

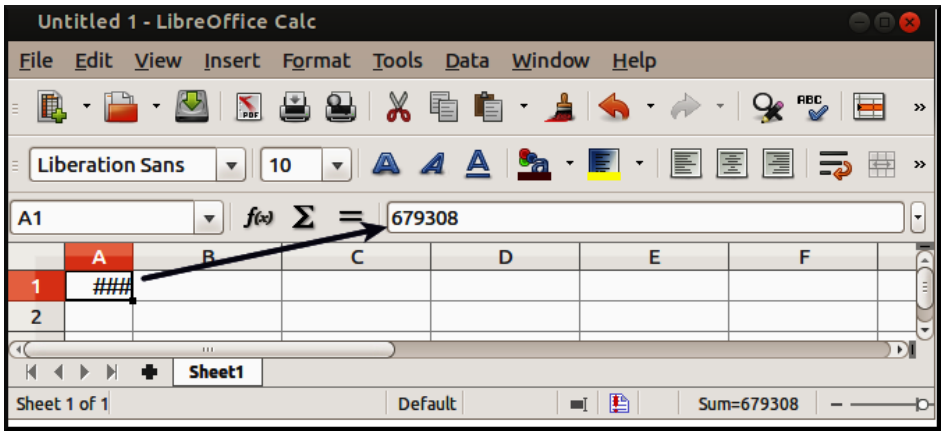


Figure 2.84Examples of ### Numerical Value entered in a cell is too large

Solution to this problem are

1) Increase the width of the column.

2) Select the column , the goto **Format '! Cells '! Alignment** and click *Shrink to fit cell size*

3) Select the Column , the goto Format '! Column '! Optimal width and Click Ok button

2.#DIV/0! Error (Error code – 532)

LibreOffice Calc shows the #DIV/0! error when a number is divided by zero (0). It happens when you enter a simple formula like **=5/0**, or when a formula refers to a cell that has 0 or is blank, as shown in this Figure 2.86.

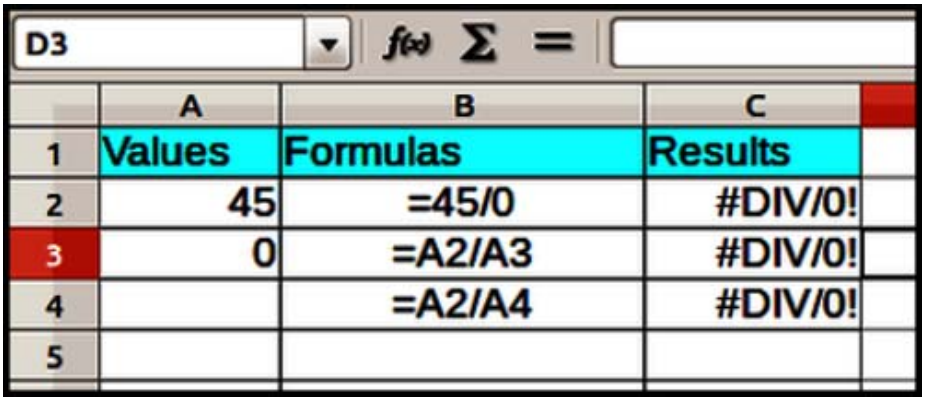


Figure 2.85 Examples of #DIV/0!, Division by zero error

Solution to this problem are

a) Change the cell reference to another cell.

b) Enter a value other than zero in the cell used as a divisor

c) Prevent the error message by using the logical function IF..

For Example =IF(B1=0," ",A1/B1) . Here cell B1 equals 0, an empty string

(" ") is displayed. If not, the result of the formula A1/B1 is displayed as shown in Figure 2.86.

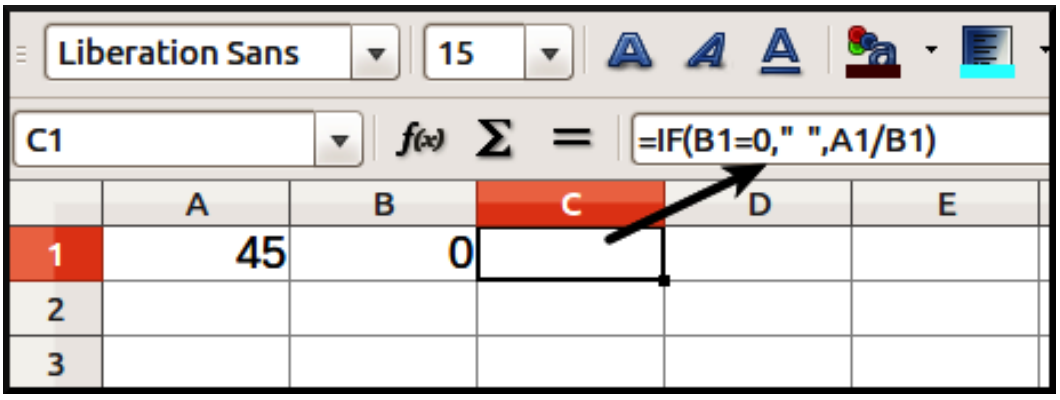


Figure 2.86 Prevent #DIV! Error using the logical function IF

3.#NAME Error (Error code – 525)

This error occurs when LibreOffice Calc does not recognise the Text in formula. This may occurs with misspelled formula or named range. The error also occurs if we forget to close a text in double quotes or omit the range operator in formula as shown in Figure 2.87. So enter proper formula name, named range,range operator to avoid these types of errors.

	A	B	C	D
1		Formula	Results	Reason for Error
2	66	=SU(A2:A3)	#NAME?	Text in formula is incorrect 'SU'
3	20	=SUM(score)	#NAME?	Using name of range not created
4	86	=SUM(A2A3)	#NAME?	Misisng (:) range seperator in formula
5	A	=IF(A5=A,A2,A3)	#NAME?	Entering text in a formula with out double quotes. Eg. =IF(A5="A",A2,A3)
6				

Figure 2.87 Examples of #NAME! error, No valid reference exists for the argument

4. #REF! Error (Error code – 524)

The **#REF!** error shows when a formula refers to a cell that is not valid . This happens most often when cells that were referenced by formulas get deleted. For example, enter the formula =A1+B1+C1 in Cell D1, then delete the Column C1, immediately the formula return **#REF!** Error as shown in Figure 2.88 and Figure 2.89. To fix the errors ,you can either delete **+#REF!** In the formula or you can undo your action by pressing Ctrl+Z.

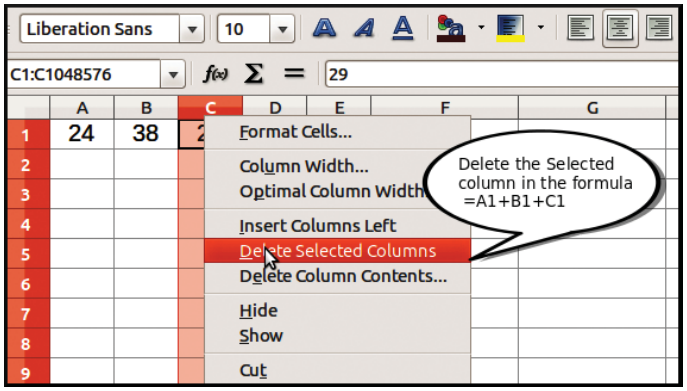


Figure 2.88 Delete the selected Column referred in formula

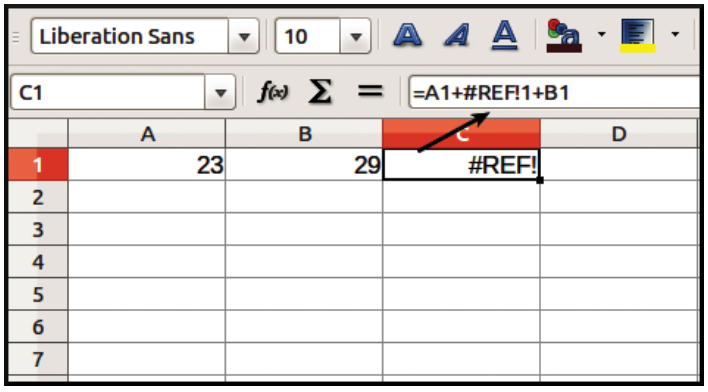


Figure 2.89 Example of #REF! Error

5. #VALUE! Error

This error occurs when a wrong argument is given in a formula. For example , if cell A1 contains a number 25 and cell A2 contain the text 'commerce'. The formula =A1+A2 return an error message #VALUE! As shown in Figure 2. 90. This type of error is generated when one of the variable in the formula is of the wrong type(e.g.:- cell referenced by the formula contains text instead of a number).

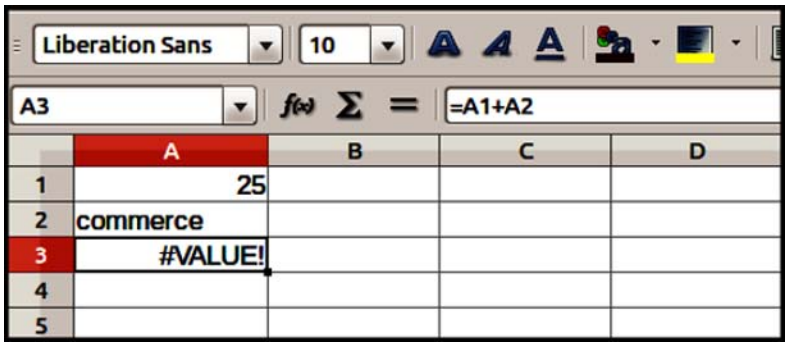


Figure 2.90 Examples of #VALUE! Error, cell referenced by the formula contains text

6. #NUM! Error (Error code – 503)

The #NUM! Error occurs when a calculation resulted in an overflow of the defined value range.

Summary of Common errors in LibreOffice Calc

Error Message	Error Code	Explanation of the error
###	N/A	When a numerical value entered in a column is not enough to display the contents.
#DIV/0!	532	Whe a number is divided by zero (0).
#NAME	525	When a Calc does not recognise the Text in formula.
#REF!	524	When a formula referes to a cell that is not valid.
#VALUE!	519	When a wrong argument is given in a formula
#NUM!	503	When a calculation resulted in an overflow of the defined value range

Chapter 4

GRAPHS AND CHARTS FOR BUSINESS DATA

INTRODUCTION TO CHARTS AND GRAPHS

One of the very useful features of spreadsheet is the capability to create charts and graphs of the data. Charts (also known as graph) are visual representations of numerical data, which has at least two dimensional relationship. A graph has at least two axes – X and Y. X axis usually horizontal while Y axis is vertical.

A picture is worth a thousand words, and a well-done graph can communicate a lot of information in a very concise form. Charts can usually be read more quickly than the raw data. Charts are often used to ease understanding of large quantities of data and the relationships between parts of the data. Charts often makes it easier to understand the data in a worksheet because users can easily pick out patterns and trends illustrated in the chart that are otherwise difficult to see. In general, a chart is a graphical representation of data. Charts allow users to see what the results of data to better understand and predict current and future data.

TYPES OF GRAPH AND CHART

A chart is a powerful tool that allows you to visually display data in a variety of different chart formats such as Bar, Column, Pie, Line, Area, Donut, Scatter, or Radar charts, etc. **LibreOffice Calc** provides various types of charts to help you to display data in different ways as per the need of the viewers. You can create a new chart or can change the existing chart, from the wide range of chart subtypes. Major chart types available are given below

Column Chart

Column chart shows a bar chart or bar graph with vertical bars. The height of each bar is proportional to its value. The X - axis shows categories and the Y - axis shows the value for each category,i.e, in column chart, categories are displayed horizontally and values are vertically. It is the default and most commonly used chart type. Column chart are used to compare values across categories. See Figure 4.1

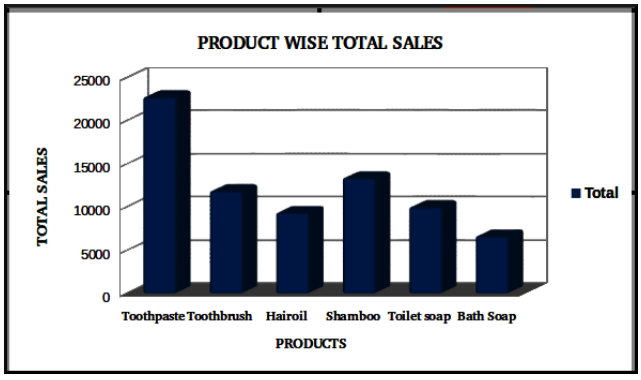


Figure 4.1 Example of Column chart

Bar chart

This type shows a bar chart or bar graph with horizontal bars. The length of each bar is proportional to its value. The Y - axis shows categories. The X - axis shows the value for each category. The bar chart is similar to the column chart, with the difference being that the data series are displayed horizontally and not vertically as shown in Figure 4.2. It is suitable for comparing multiple values.

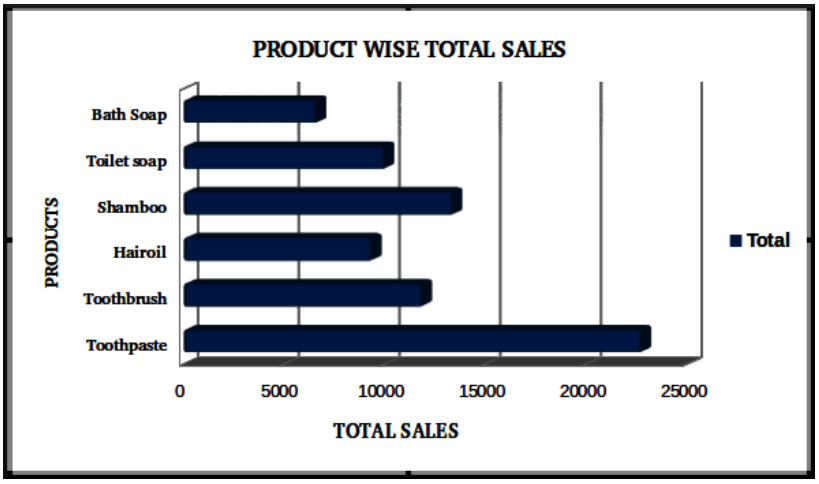


Figure 4.2 Example of Bar chart

Line chart

A line chart shows values as points on the Y- axis. The X - axis shows categories. The ***y*** values of each data series can be connected by a line as shown in Figure 4.3. The line chart show data changes for a certain period of time. It is used to display trends over time.

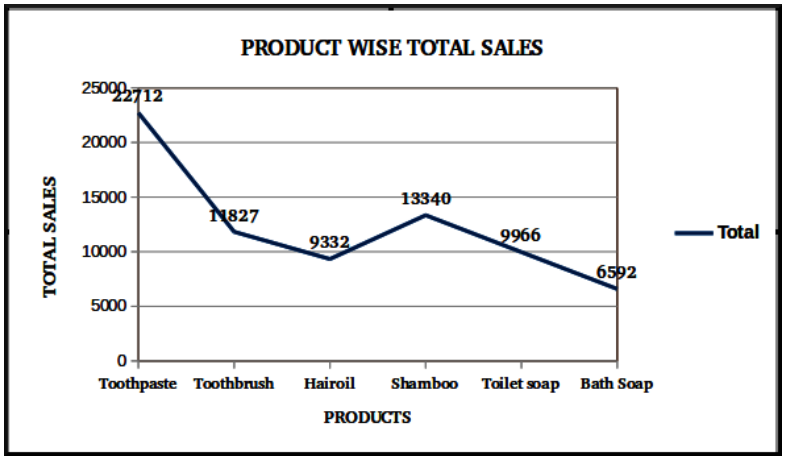


Figure 4.3 Example of Line chart

Pie chart

A pie chart displays the contribution of each value to a total. It represents multiple sub-groups of a single variable. It contains only one data series. A pie chart shows values as circular sectors of the total circle. The length of the arc, or the area of each sector, is proportional to its value. The pie chart have the following chart subtypes

a) Normal Pie Chart- This subtype shows sectors as coloured areas of the total pie, for one data column only (Figure 4.4).

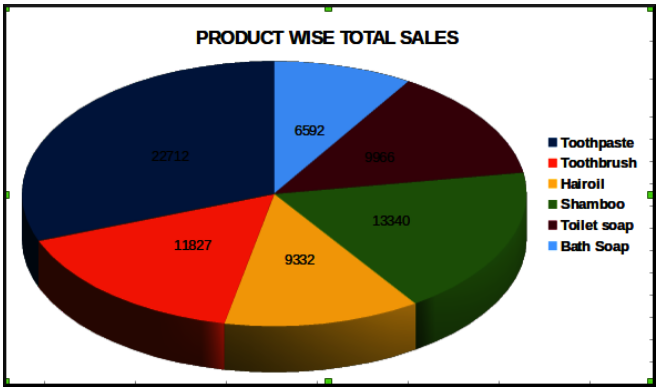


Figure 4.4 Example of Pie chart

b) Exploded pie Chart - Exploded pie chart is the kind of pie chart in which one or several slices are separated from the other. It is useful because it makes the highlighted portion more visible (Figure 4.5).

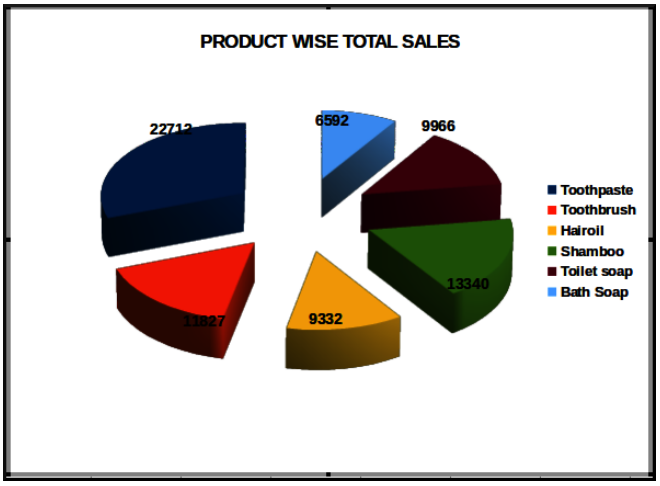


Figure 4.5 Example of Exploded Pie chart

c) Donut Chart - A Donut or Doughnut chart is a pie chart, with two exceptions: It has a hole in the middle and it can display more than one series of data. Doughnut charts display data in rings, where each ring represents a data series. The first data series is displayed in the centre of the chart (Figure 4.6).

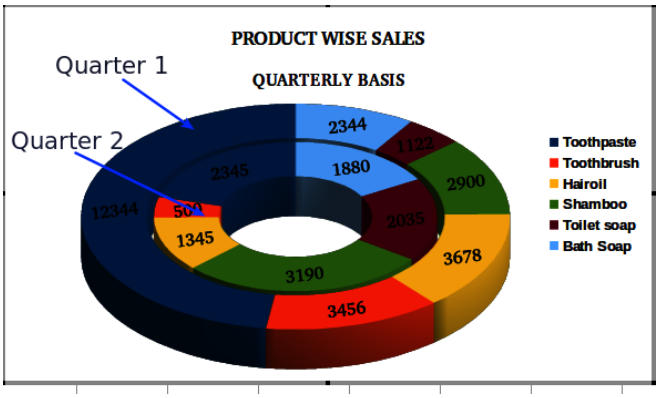


Figure 4.6 Example of Donut chart

d) Exploded Donut Chart – It is a Donut chart with all slices exploded. It shows the outer sectors already separated from the remaining Donut (Figure 4.7).

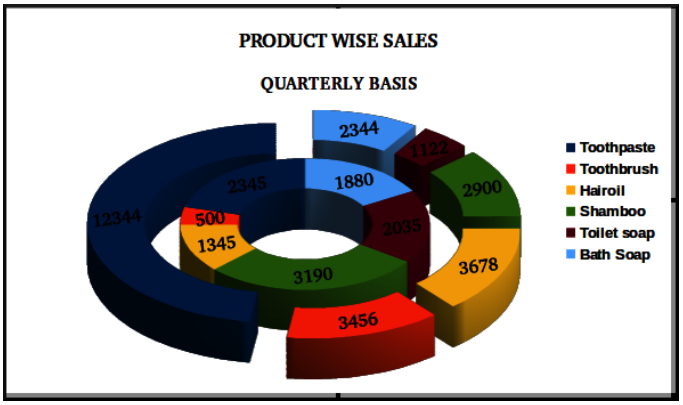


Figure 4.7 Example of Exploded Donut Chart

Area Chart

Area chart shows values as points on the Y axis. The X axis shows categories. The ***y*** values of each data series are connected by a line. The area between each two lines is filled with a colour. The area chart's focus is to emphasize the changes from one category to the next. Area Charts are like Line Charts except that the area below the plot line is solid (See Figure 4.8).

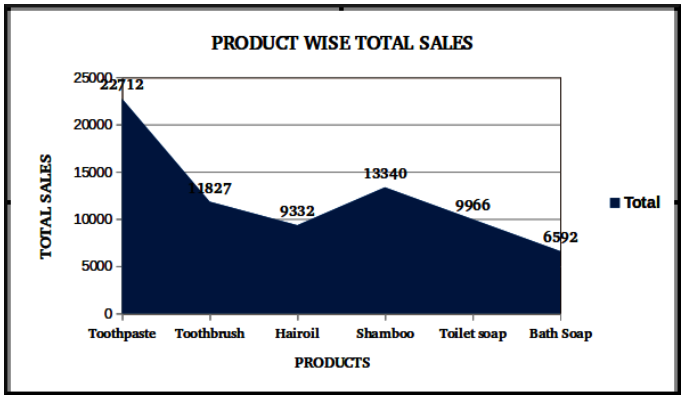


Figure 4.8 Example of Exploded Donut chart

XY (Scatter) chart

XY charts are also known as Scatter charts. The point of difference between XY charts and other types of charts is that in XY charts both axes display values. Such type of charts is generally used to show the relationship among two variables. It is commonly used for scientific, statistical, and engineering data. Figure 4.9 show a XY chart based on the data – how many hours children of different ages slept in a day.

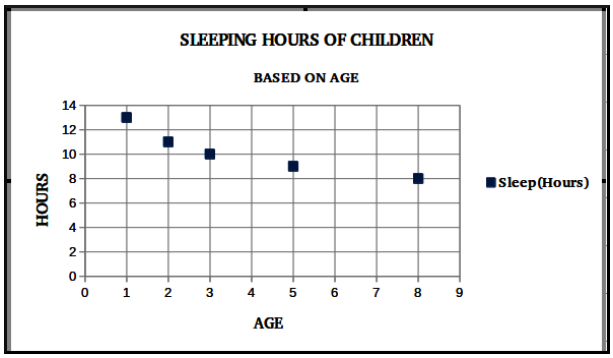


Figure 4.9 Example of XY chart

Radar Chart

It also known as Net Chart or Star Chart . A Net Chart has a separate axis for each category, and the axes extend outward from the center of the chart. The value of each data point is plotted on the corresponding axis. It displays values relative to the centre point as shown in Figure 4.10.

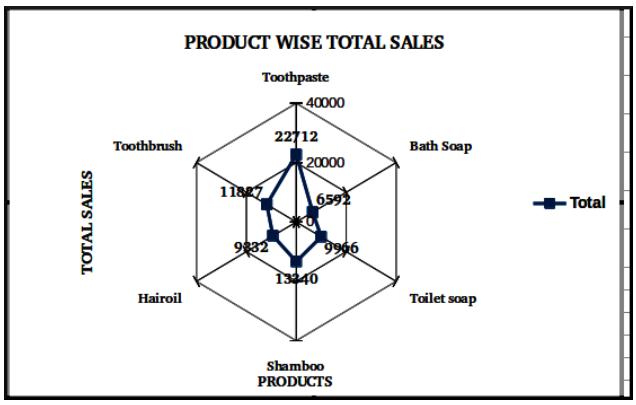


Figure 4.10 Example of Radar chart

BASICS STEPS FOR GRAPHS/CHARTS/DIAGRAMS USING CALC

The steps involved in the chart preparation is given below:

Step 1 - Data entry : In this stage, the data needed to create a chart must be entered to the Calc sheet. In the Figure 4.11, the details of six different products manufactured by ROYAL Manufacturing Company is given.

	A	B	C	D	E	F
1	**Product**	**Qtr1**	**Qtr2**	**Qtr3**	**Qtr4**	**Total**
2	Toothpaste	6042	2345	3456	4567	**16410**
3	Toothbrush	3456	500	2456	5415	**11827**
4	Hairoil	3678	1345	988	3321	**9332**
5	Shamboo	2900	3190	3480	3770	**13340**
6	Toilet soap	1122	2035	2948	3861	**9966**
7	Bath Soap	2344	1880	1416	952	**6592**
8	**Total**	**19542**	**11295**	**14744**	**21886**	**67467**

Figure 4.11 The sales details of six different products

Step 2 - Data selection: All the data entered to the sheet is not needed for plotting chart. It depends upon the need of the firm. In the given example, ROYAL Manufacturing Company needs to plot the graph of **PRODUCT-WISE TOTAL SALE** only. Then, the selection of two columns (Product and Total) needed to plot the graph. It can be done in the following way (Figure 4.12):

- Select the first column (*Product*).
- Press ***Ctrl*** key and select the next column needed (*here it is Total*)

	A	B	C	D	E	F
1	Product	Qtr1	Qtr2	Qtr3	Qtr4	Total
2	Toothpaste	6042	2345	3456	4567	16410
3	Toothbrush	3456	500	2456	5415	11827
4	Hairoil	3678	1345	988	3321	9332
5	Shamboo	2900	3190	3480	3770	13340
6	Toilet soap	1122	2035	2948	3861	9966
7	Bath Soap	2344	1880	1416	952	6592
8	Total	19542	11295	14744	21886	67467

Figure 4.12 The column Product and Total after selection

***Step 3 - Plotting the chart*:** After the selection of required columns, click on **Insert** → **Chart**. (It also can be done by clicking the chart icon from **Standard** toolbar. **Chart Wizard** window appears (Figure 4.13).

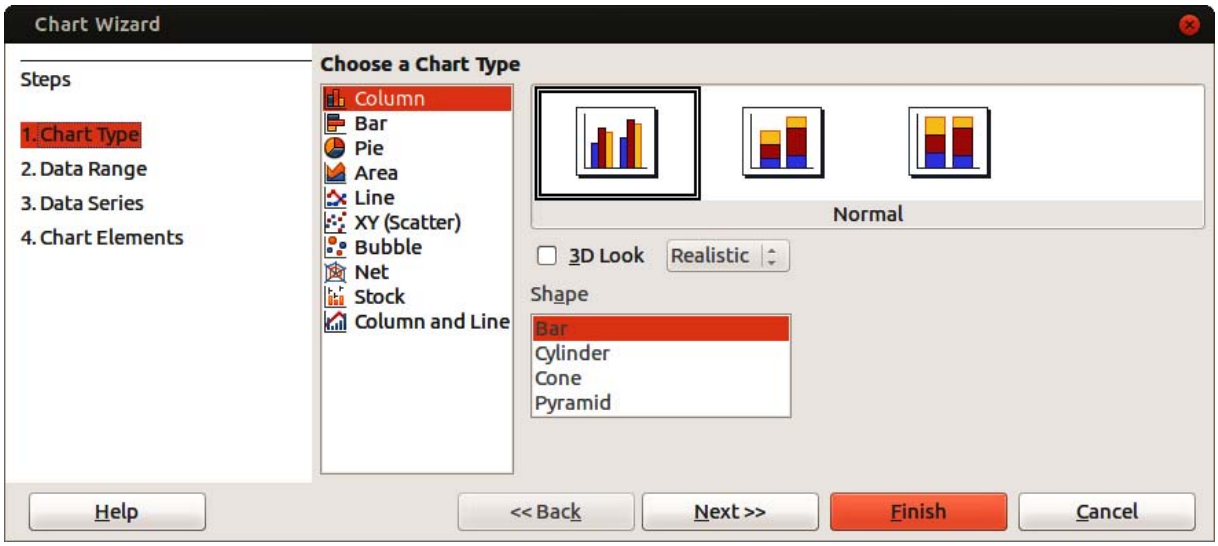


Figure 4.13 Chart wizard window

Plotting of chart is a four step procedure. These steps are displayed in the left part of the **Chart Wizard** window. The need of these steps are given below:

***a) Chart Types* :** In this stage the user can select the desired chart type from **Choose a Chart Type** list(Figure 4.13)

b) Data Range: In this stage, we can change the data ranges and axes labels(Figure 4.14)

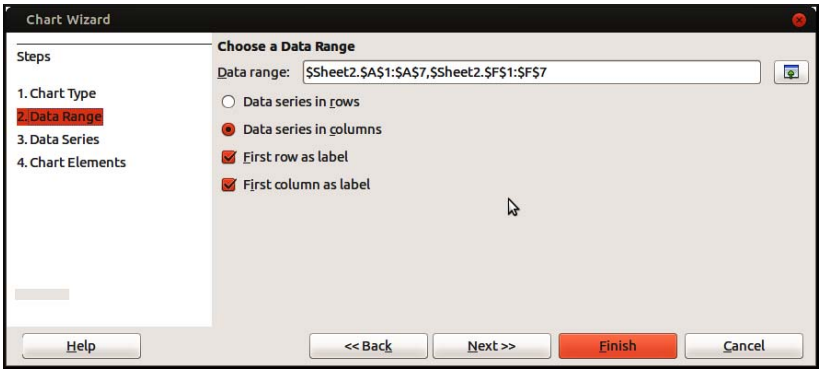


Figure 4.14 Chart Wizard with Data Range

c)Data Series: Click **Next** button after making necessary changes(Figure 4.15).

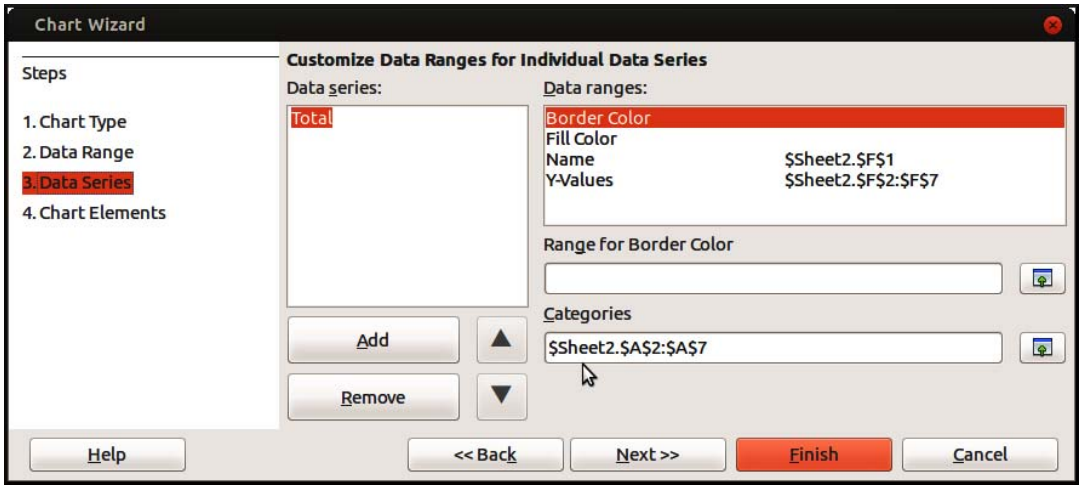


Figure 4.15 Chart Wizard with Data Series

***d) Chart Elements*:** The user can choose the chart elements like titles, legend and grid settings from this chart wizard page (Figure 4.16). After this process, press **Finish** button , we will get the chart as shown in Figure 4.17.

Figure 4.16 Chart Wizard with chart elements

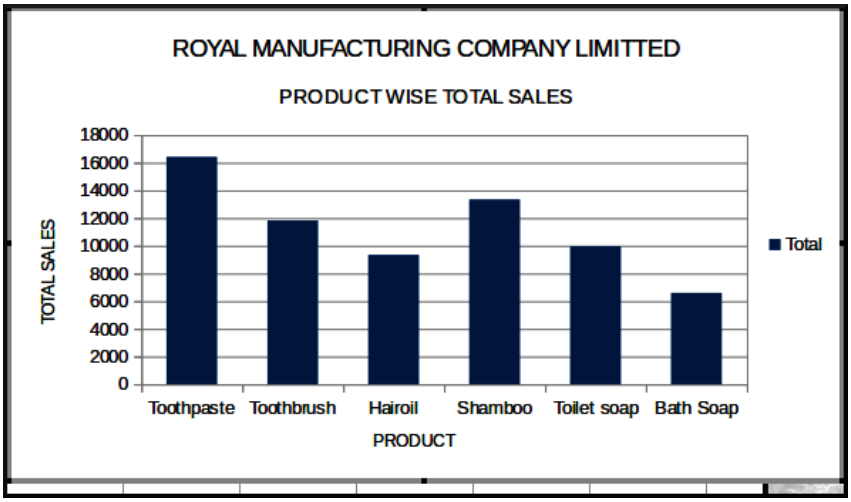


Figure 4.17 Column chart - Product wise Total Sales

ELEMENTS OF A CHART/GRAPH

The different parts that make up a chart are referred to as chart elements (Figure 4.18). The following are the important elements (Table 4.1) -

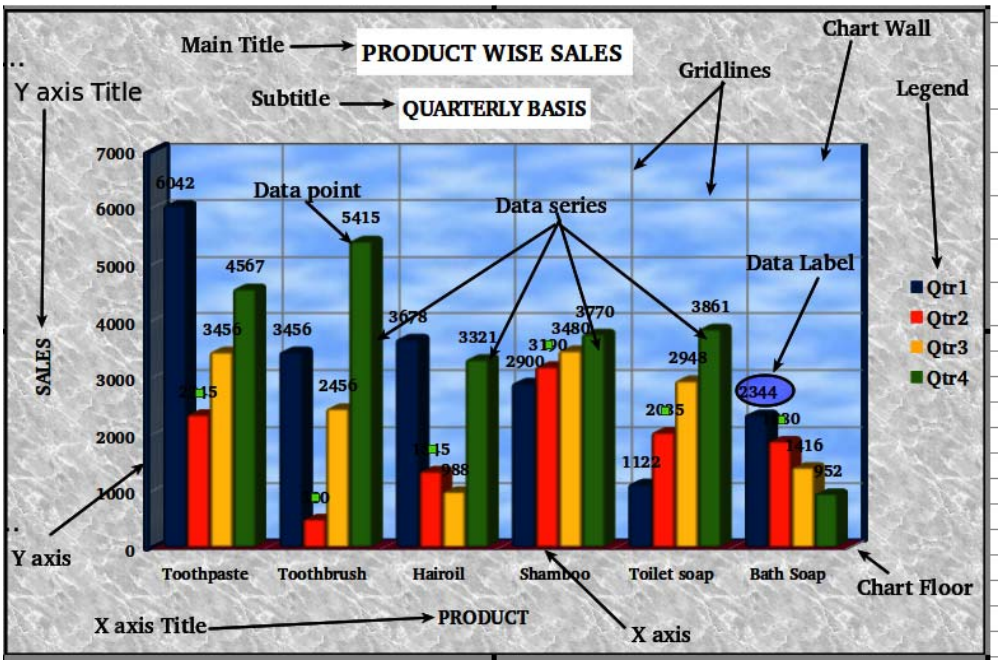


Figure 4.18 Elements of Chart

Chart Elements	Description
Chart Area	This is the total space that is enclosed by a chart. The entire chart including all elements is termed as Chart area. It is the background of the chart.
Chart Wall	It is the area in which chart is plotted. In a 2D chart, the area bounded by X and Y axis. In a 3D chart, the wall is bounded by three axes(X,Y & Z).
Chart Floor	The chart floor is the lower area in 3D charts. It is only available for 3D charts.
Chart Main Title/ Subtitle	This is the explanatory heading at the top of the chart. The title identifies the purpose of a chart.
X , Y & Z axes	In 2D chart , there are two axes namely , horizontal (X) axis which contains categories (independent values or categories) and vertical (Y) axis containing data (dependent Values). But in the case of 3D chart Z axis will also be there, which represent the depth.
Axes (X,Y and Z) Titles	Mention the name or title for X,Y and Z axes.

Data point	Individual values plotted in a chart and represented by bar, columns, line, pie or various other shapes are called data point or Data marker. It is a symbol on the chart that represent one value of data series.
Data Series	Data markers of the same colour or pattern is called data series. The data series are related data points that are plotted in the chart/ graph.
Legend	It is an identifier of a piece of information shown in the chart/ graph. Generally, these legends are attached to a symbol or colour or pattern that is associated with data series of the chart. It is used to distinguish one data series from another.
Data Label	The value of the data series plotted in a chart is known as data label. This provides additional information about a data marker to identify the details of data point in a data series – either values as number or as percentage.
Grid lines	These are the vertical and horizontal lines that appear in a chart. They are displayed in the chart wall. It increases the readability of a chart.

Table 4.1 Elements of Chart/Graph

FORMATTING OF CHART

The Format menu has many options for formatting and fine-tuning the appearance of your charts. Double-click the chart so that it is enclosed by a gray border indicating edit mode; then, select the chart element that you want to format. Choose Format from the menu bar, or right-click to display a context menu relevant to the selected element (Figure 4.19).

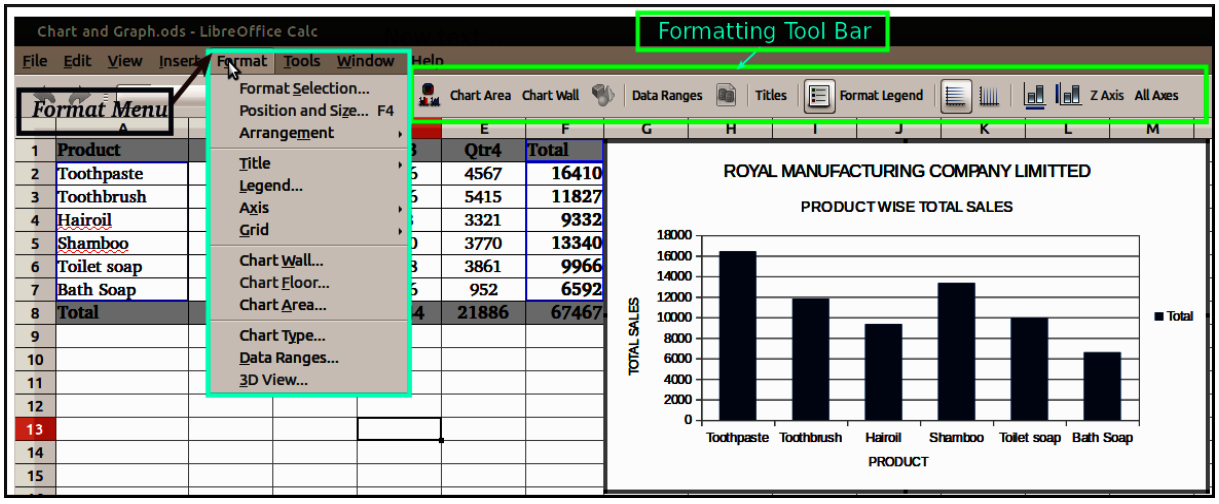


Figure 4.19 Format menu and Tool bar

The formatting choices are as follows :

Format Selection : Opens a dialog in which you can specify the area fill, borders, transparency, characters, font effects, and other attributes of the selected element of the chart.

Position and Size: Opens a dialogue (See figure 4.22). From the window, the user can rearrange the position and size of the graph.

Title : Formats the titles of the chart and its axes.

Legend : Formats the location, borders, background, and type of the legend.

Axis : Formats the lines that create the chart as well as the font of the text that appears on both the X and Y axes.

Grid : Formats the lines that create a grid for the chart.

Chart Type : Changes what kind of chart is displayed and whether it is two or three dimensional.

Data Ranges: To format data range

3D View : Formats 3D charts.

Moving Chart Elements

You may wish to move or resize individual elements of a chart, independent of other chart elements.

1) Double-click the chart so that it is enclosed by a gray border.

2) Double-click any of the elements - the title, the legend, or the chart graphic. Click and drag to move the element. If the element is already selected, then move the pointer over the element to get the move icon (small hand), then click, drag and move the element.

3) Release the mouse button when the element is in the desired position.

Changing Chart area background

The chart area is the area surrounding the chart graphic, including the main title, subtitle and legend.

1) Double-click the chart so that it is enclosed by a gray border.

2) Choose **Format → Chart Area.**

3) On the Chart Area dialogue, choose the desired format settings (Figure 4.20).

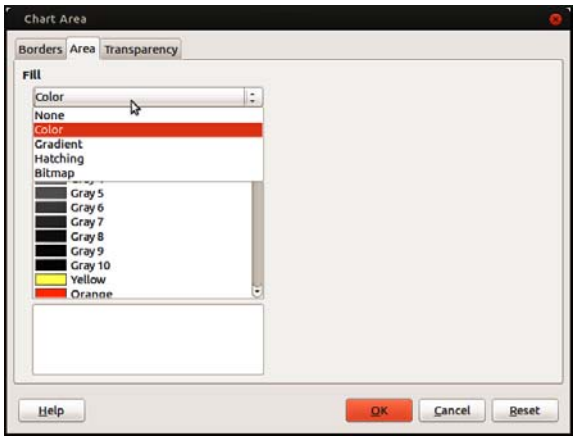


Figure 4.20 Chart Area dialogue

Changing Chart Graphics Background

The Chart Wall is the area that contains the chart graphic.

1) Double-click the chart so that it is enclosed by a gray border.

2) Choose **Format → Chart Wall.**

3) Choose your settings and click **OK** (Figure 4.21).

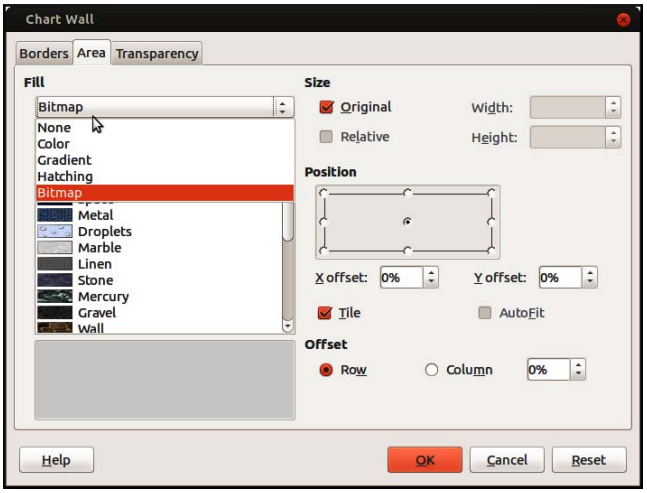


Figure 4.21 Chart Wall dialogue

CHANGING THE CHART TYPE

You can change the chart type at any time.

1) First select the chart by double-clicking on it. The chart should now be surrounded by a gray border.

2) Then do one of the following:

- Choose **Format → Chart Type** from the menu bar.
- Click the **Chart Type** icon on the Formatting toolbar.
- Right-click on the chart and choose **Chart Type**.

3) Select the replacement chart type

4) Click **OK**.

STEPS FOR CREATING A PIE CHART

Following are the important steps for creating a Pie chart

Step 1 : Open LibreOffice Calc and enter your data in two column

Step 2 : Select the cell range that contains the source data whose chart we want to make.

Step 3 : Select **Insert → Chart** from the Menu bar or Click on Chart icon

Step 4 : In the **Chart Wizard** dialog box, select Pie in the **Choose a chart type** list box and click the Next button.

Step 5 : Choose the **Data Range**.

Step 6 : Click on the **Next** button.

Step 7 : Type the title of the chart in the Title text box and click on **Finish** Button.

RESIZING AND MOVING CHART/GRAPH

We can resize or move all elements of a chart at the same time, in two ways - Interactively, or by using the Position and Size dialog.

A) Interactively

Steps to resize a chart interactively:

1) Click once on the chart to select it and square selection handles appear around the border of the chart.

2) To increase or decrease the size of the chart, click and drag one of the markers. The cursor indicates the direction in which you can increase or decrease the chart size.

Steps to move a chart interactively:

1) Click once on the chart to select and drag it to its new location.

2) Release the mouse button when the chart is in the required position.

B) Using the Position and Size Dialog

Step to resize or move a chart using the Position and Size dialog:

1) Click once on the chart to select it.

2) Right-click and select **Position and Size** from the context menu.

3) Make your choices on this dialogue box and click on **OK** to save your changes. (Figure 4.22).

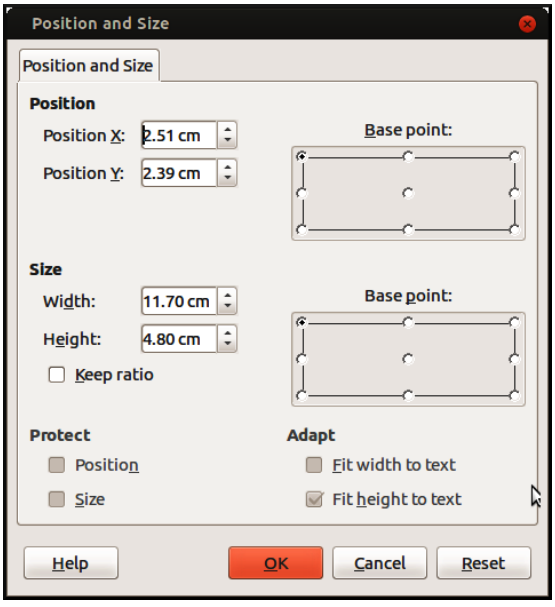


Figure 4.22 Position and Size dialogue box

2D - 3D CHARTS/GRAPHS

To create graphs we use data which are plotted in two dimensional (2D) format (X- axis and Y-axis), where as:

- Horizontal dimension is X-axis (contains categories)
- Vertical dimension is Y-axis (contains data)

When we plot the data on 2-D type of graph; the known value goes on the X-axis and derived value on Y-axis. For example monthly demand of products (in Rs.); then on X-axis we will put Month and on Y-axis we will put the data values for demand.

In the graph sometimes we have to present negative values also, which can be put on the opposite side of the axes from origin (Figure 4.23). The intersection of both the axes (X-axis and Y-axis) is called the origin (O) of the graph. We can put on the right side of the origin positive values and on left side of the origin negative values of data on X-axis. Similarly upper ward side of origin shows positive values and down ward side of the origin shows negative values of data Y-axis.

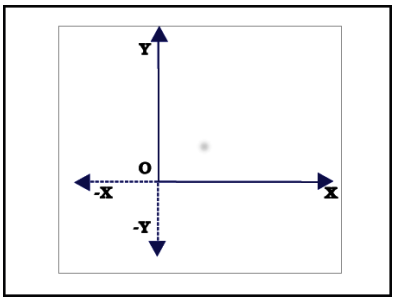


Figure 4.23 Two Dimensional (X axis and Y axis)

For example, if we have data about monthly profit and loss to be plotted on graph; here profit will be positive data values and loss will have negative data values.

Different types of 2D graphs/charts are Column chart, Line chart, Bar chart, Area chart, Radar Chart, XY (scatter) chart or bubble chart.

Sometimes graphs/charts can be prepared with three dimensional (3-D) effects. 3-D charts have a third axis, the depth axis (also known as series axis or Z axis), so that data can be plotted along the depth of a chart. In this type the third dimension is represented by Z-axis (Figure 4.24).

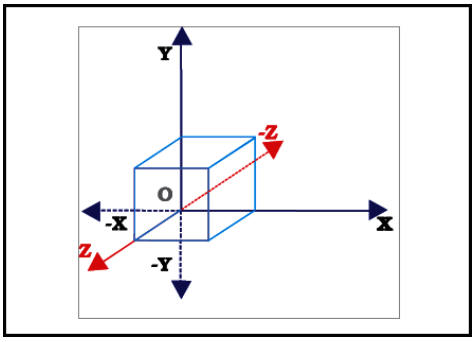


Figure 4.24 Three Dimensional (X axis, Y axis and Z axis)

For example, to represent the volume we require three parameters length (X-axis), height (Y-axis) and breadth (Z-axis).

1. Horizontal (category) axis –(X-axis).

2. Vertical (derived value) axis – (Y-axis).

3. Depth (series) axis – (Z-axis).

Different types of 3D graphs/charts are Column chart, Line chart, Bar chart, Area chart, Pie chart and XY (scatter) chart.

Radar charts do not have horizontal (category) axes, and similarly pie and doughnut charts do not have any axes. Like a pie chart, a doughnut chart shows the relationship of parts to a whole, but it can contain more than one.

ADVANTAGES IN USING GRAPH/CHART

***Help to Explore*:** Many times we would like to see if there is a relationship between variables. Suppose, that we wanted to determine if there is a relationship between: a country's GNP and the infant mortality rate, between age and genders. It may be quicker and easier to create a chart immediately to see the possible relationship of variables to one another, rather than paging through raw data.

***Help to Present*:** We want to provide information in as little time as possible. Graphing plays a key role. It seems that there is no longer any time to sit and read a newspaper in order to find out what is going on. However, newspapers, such as The Economics Times and India Today magazines (which were early users of charting techniques), seem to understand this phenomena and provide graphs to convey and sum up ideas that they are making in their articles.

***Help to Convince*:** The same way that a graph can be used to present and explore different characteristics of data, it can also be used to convince. Graphs have the ability to take large amounts of information and make them into exhibitions that are easily used to persuade.

Chapter.5

ACCOUTING SOFTWARE PACKAGE - GNUKhata

INTRODUCTION

As we know software is needed to run computers. The software may be proprietary or free and open. Users have to pay for proprietary software and cannot make any change in the same. On the other hand the Free and Open Source Software(FOSS) can be legally downloaded and copied without having to pay anything to anyone. Not only this, you can make changes in FOSS. Accounting software is also available in FOSS. GNUKhata is one of such accounting software

GNUKhata

GNUKhata, pronounced as ***jee_new_khata***, developed by Digital Freedom Foundation, a public charitable trust, in association with International Centre for Free and Open Source Software (ICFOSS). GNUKhata is an accounting software based on Double Entry Book Keeping System.

GNUKhata is a free and flexible software for accounting and inventory management. It supports a wide range of applications in every field of economic activity. While it serves traditional accounting requirements, its special feature is that it can support emerging sectors of the economy. GNUKhata is highly customizable. It can be easily transformed into Indian languages.

Features of GNUKhata

- Free and open source accounting software
- Based on double entry book keeping
- Comprehensive financial reports - Ledgers, Trial balance, Profit and loss A/c , Balance sheet etc...
- Display of dual ledger facility
- Source document attachment facility in vouchers
- Linking of sales and purchase transactions to invoices
- Export and import from Spreadsheet
- Password security and data audit facility

Installing GNUKhata

It is very simple to install GNUKhata in Linux. The installer file can be downloaded from the website www.g*nukhata.in*. Extract the downloaded file and refer 'README' file for installation steps.

Starting GNUKhata

When we open GNUKhata for the first time we will see the Welcome Screen as shown in Figure 5.1

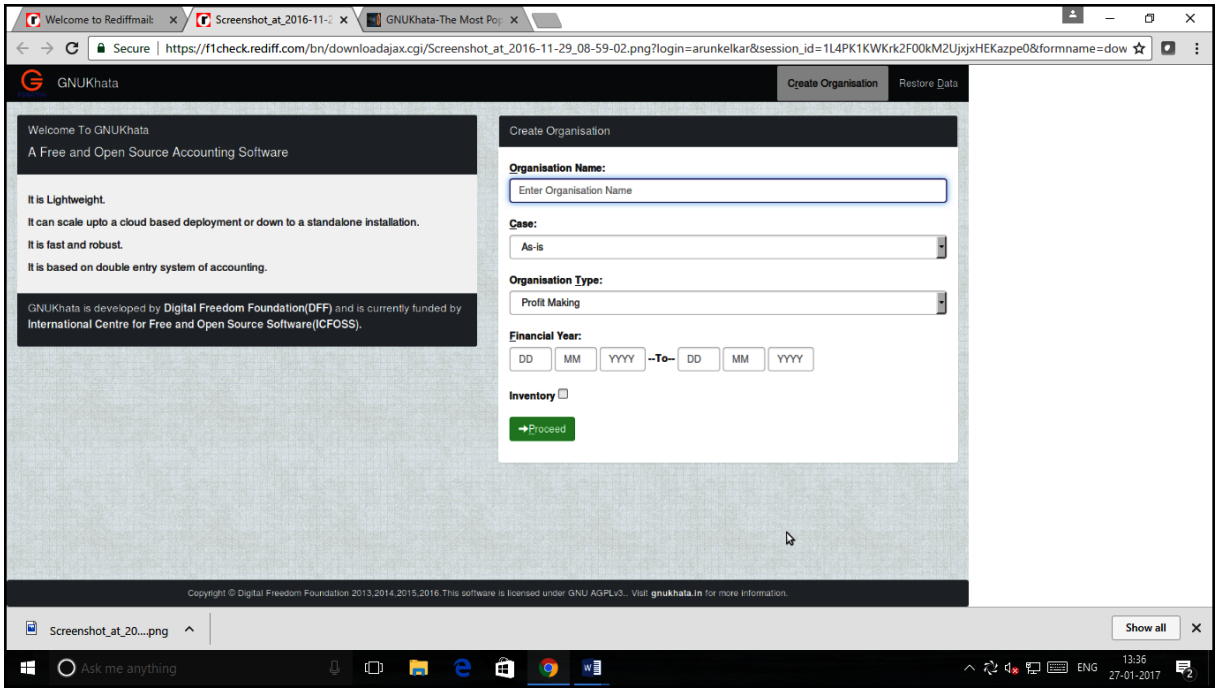


Figure 5.1 Welcome screen

CREATE ORGANISATION

The first step in GNUKhata is to create an organisation. To Create a new Organisation, click on 'Create Organisation'(or press Shift + Control + R). See Figure. 5.2

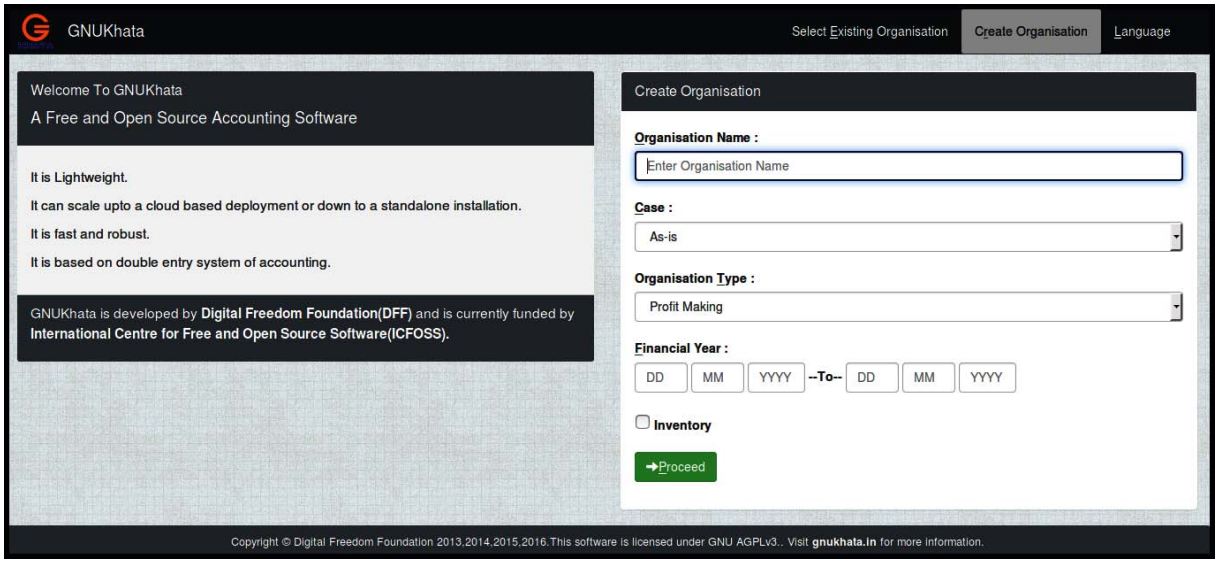


Figure 5.2 Create Organisation

While creating an organisation the following details are to be given:

Organisation Name : Enter the name of the organization and press Enter key.

Case : Choose the appearance of the organisation name. The options are As-is, Upper Case, Lower Case or Title Case. Supposing the name you entered is "bharat traders", now you can change this to BHARAT TRADERS or Bharat Traders.

Organisation Type: Select the organisation type either Profit Making or Not For Profit.

Financial Year : Enter the opening date of financial year or commencement date of the organisation then press enter key, closing date will show automatically which can be edited. Once organisation is created, the financial year can not be changed.

Inventory : Tick the box of Inventory for maintaining inventory accounts.

CREATE ADMIN AND LOG IN

The next step is the 'Create Admin' which is mandatory. Fill all the fields and Enter/click on 'Create & Login' as shown inFigure 5.3

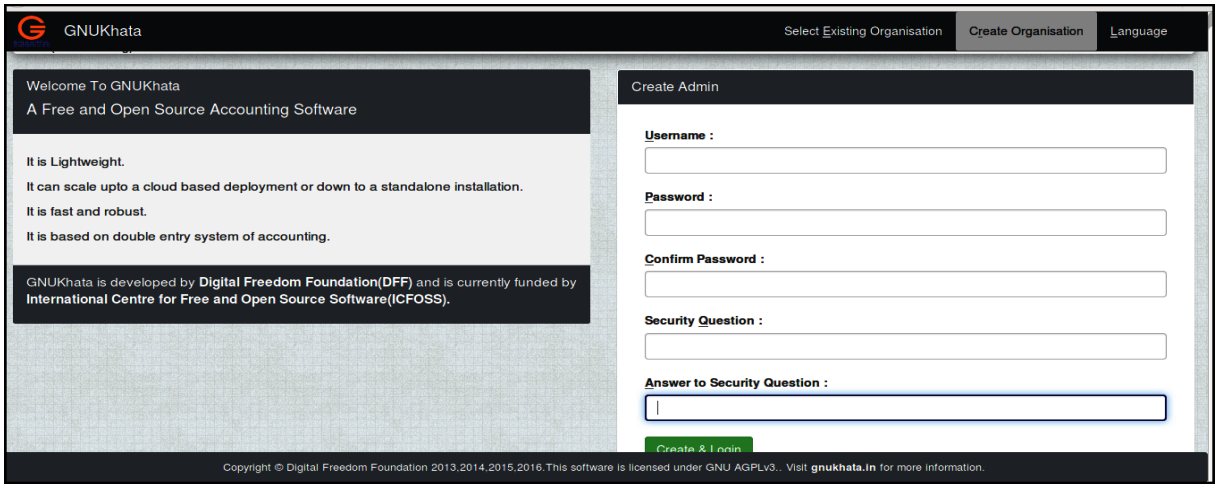


Figure 5.3 Create Admin

Now we can see a Menu Bar at the top. Click Menu item to activate as shown in Figure 5.4

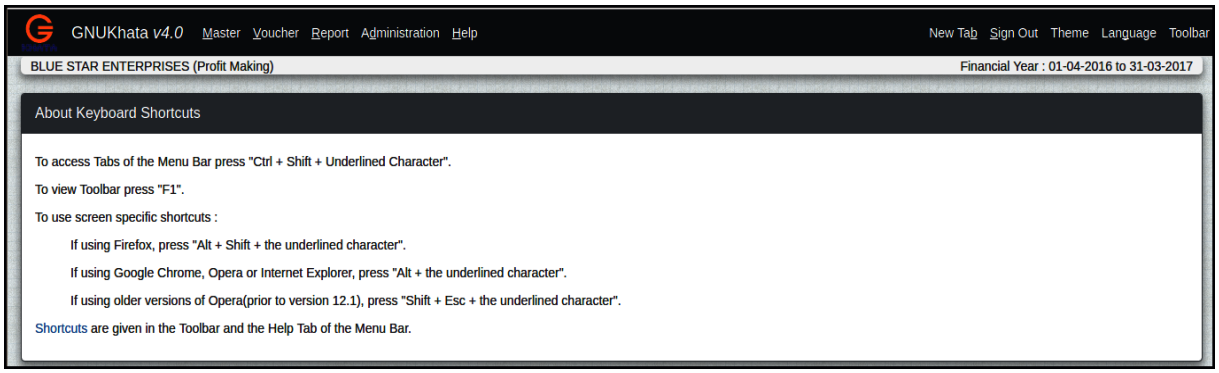


Figure 5.4 Menubar

There are four levels of users in GNUKhata 4.0, these are Admin, Manager, Operator,Internal Auditor. Each has different authorities. Only one user can log in as "admin", but there may be any number of users in the role of Manager, Operator.

ORGANISATION PARTICULARS

We can enter organisation details (like Address, Country, state, City,etc...) through **Edit Organisation particulars** in **Master** Menu. Enter or click on **Save** to save the details and click on **Reset** to clear the fields, if necessary.see Figure 5.5

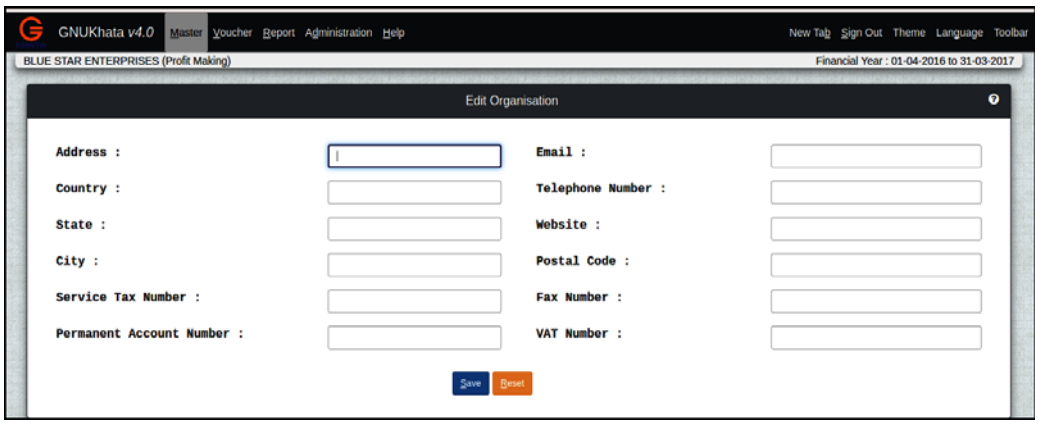


Figure 5.5 Organisation Particulars

CHANGE ORGANISATION

Select **Change Organisation**from **Sign Out** menu to exit the active organisation. To change the user, select **Logout** from **Sign Out** menu See Figure 5.6

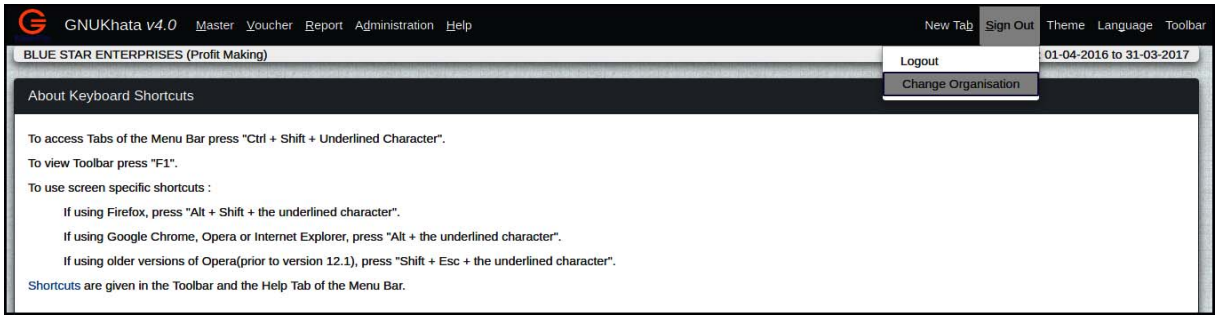


Figure 5.6 Change Organisation

SELECTING ORGANISATION

To select the existing organisation, click on **Select Existing Organisation** tab and select organisation name from the drop down menu as shown Figure 5.7

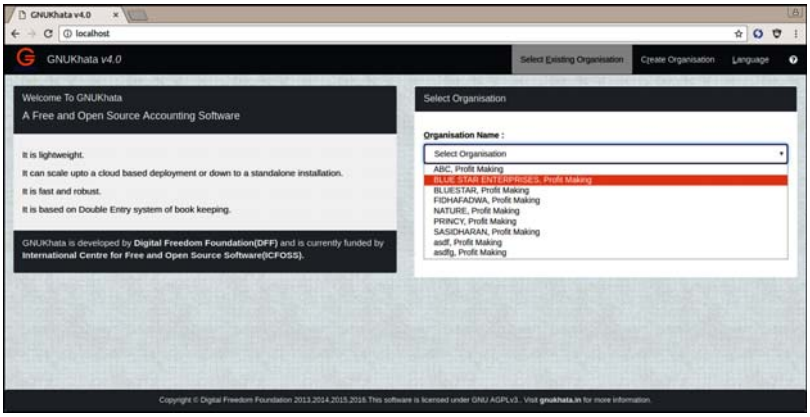


Figure 5.7 Select Organisation

DELETING ORGANISATION

After Login as 'Admin' user, select **Delete Organisation**

from **Administration** menu. You will be asked to confirm this decision and when you confirm the decision, the organization will be deleted. Remember once an organization is deleted it can never be recovered, unless the data is stored somewhere. See Figure 5.8 and Figure 5.9.

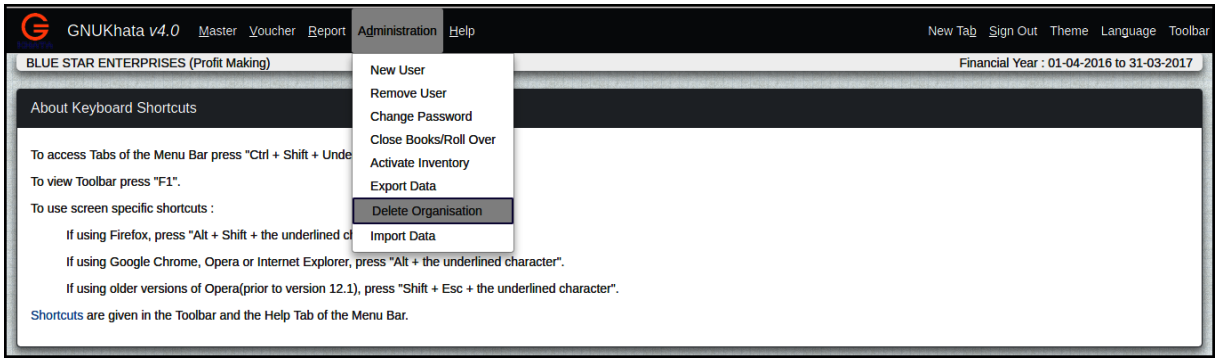


Figure 5.8 Delete Organisation

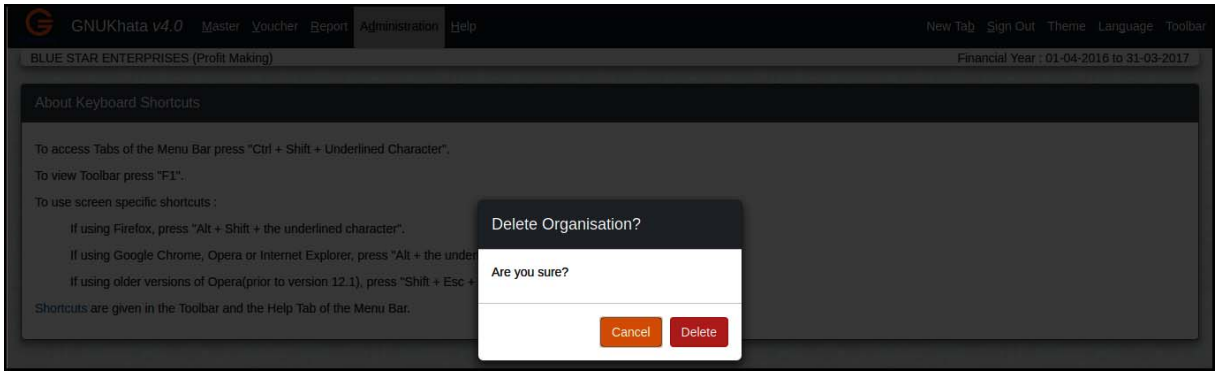


Figure 5.9 Confirm Delete Organisation

Example:1

Create an organization with the following particulars:

Name	:	BLUE STAR ENTERPRISES
Accounting Year	:	01-04-2016 to 31-03-2017
Type of Organization	:	Pofit Making
Maintain Inventory Records	:	No
Admin Name and Password	:	Rajan, Banglore
Security Question and answer	:	Best friend's name? Prabha.
Address	:	565, Link way, Mahatma Gandhi Road, Banglore

Solution:

Step 1 Start GNUKhata. (Applications → Office → GNUKhata)

Step 2 Click on 'Create Organisation'

Step 3 Enter Organization Name (Blue star enterprises), Press Enter/Tab.

Step 4 Select the Case. Use Down Arrow key to select any one of the case (As- is, Upper Case, Lower Case or Title Case) and press **Enter**.

As-is	:	Blue star enterprises
Upper case	:	BLUE STAR ENTERPRISES
Lower case	:	blue star enterprises
Title case	:	Blue Star Enterprises

Step 5 Select Organization Type - 'Profit making'and press Enter/Tab.

Step 6 Enter 'From Date' - 01.04.2016, press Enter.

Step 7 Skip the box of 'Inventory'.See Figure 5.10

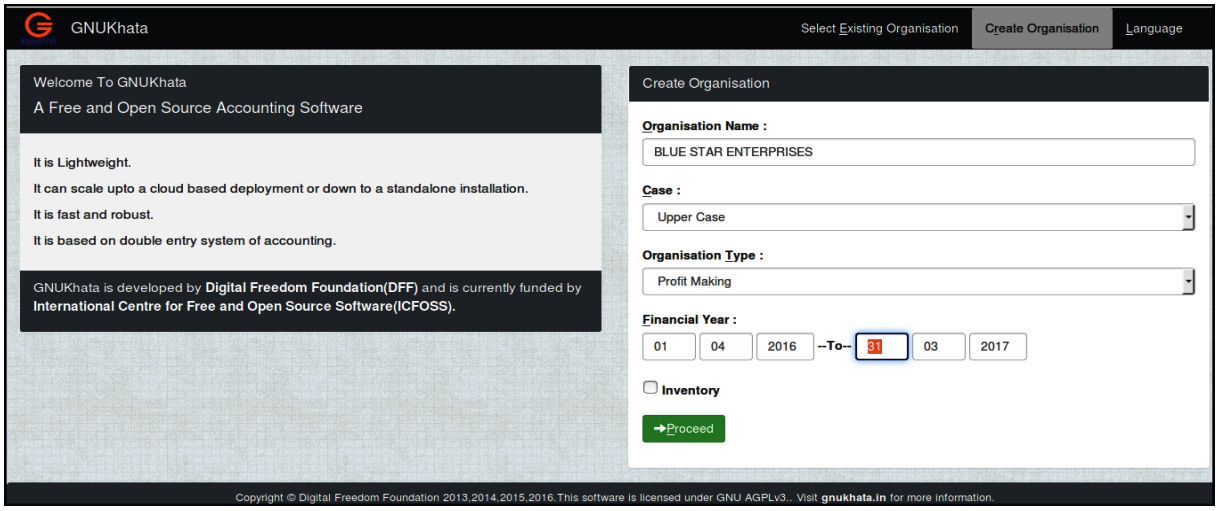


Figure 5.10 Create Organisation

Step 8 Click on **Proceed** or Press Enter to open **Create Admin** window.

Step 9 Enter User Name - 'Rajan', Password - 'Banglore', Confirm Password - 'Banglore' and Press Enter/Tab.

Step 10 Enter Security Question 'Best friend's name'. Press Enter/Tab.

Step 11 Enter 'Answer to Security Question' Prabha.

Press Enter to Login as Admin .(Figure 5.11)

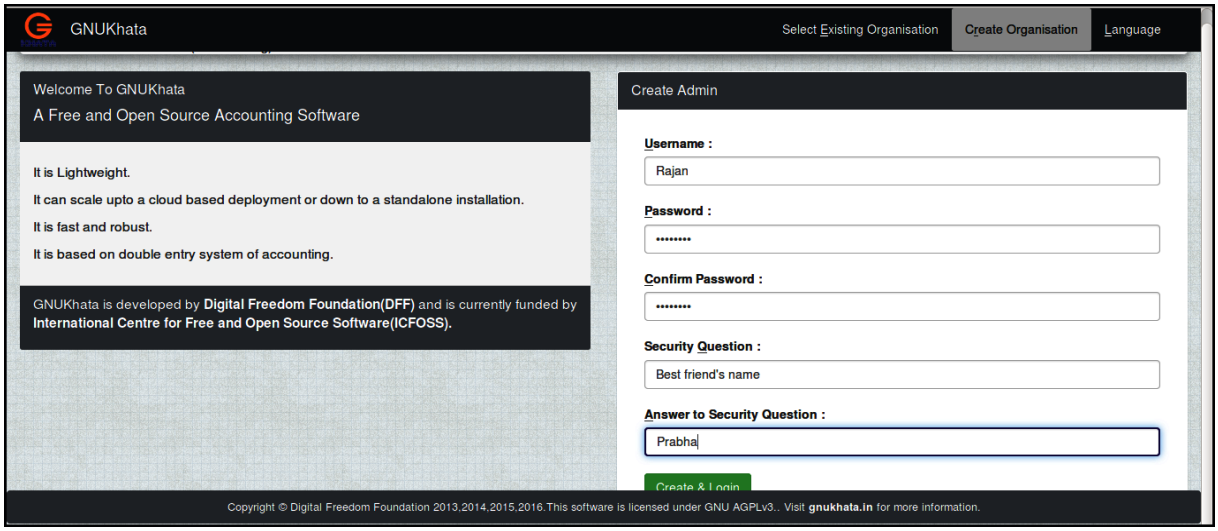


Figure 5.11 Create Admin

Step 12 Select **Edit organisation Particulars** from **Master** menu

Step 13 Enter the Address and other details of Organisation and click on **Save** as shown in Figure 5.12

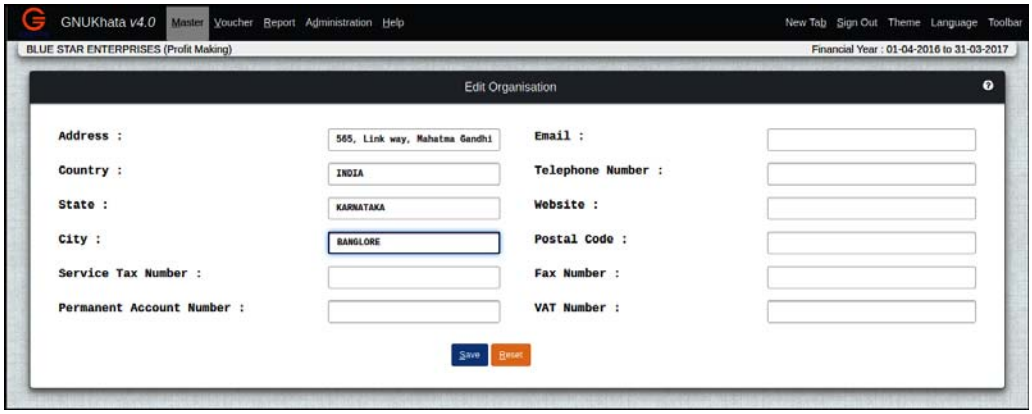


Figure 5.12 Edit Organisation Particulars

GROUPS AND SUB-GROUPS

Group of account is a method of organising the large number of ledger accounts into sequential arrangement. GNUKhata has predetermined Groups and Sub-Groups. These are as given below:

Balance Sheet Groups

The list of Balance sheet groups and sub groups are given in Table 5.1

Sl. No.	Group Name	Sub-Group Name
1.	Capital/Corpus	None
2.	Current Asset	Bank, Cash, Loans & Advance, Sundry Debtors
3.	Current Liability	Provisions, Sundry Creditors for Expense, Sundry Creditors for Purchase
4.	Fixed Assets	Building, Furniture, Land, Plant & Machinary
5.	Investments	Investment in Bank Deposits, Investment in Shares & Debentures
6.	Loans (Asset)	None
7.	Loans (Liability)	Secure, Unsecured
8.	Miscellaneous Expenses (Asset)	None
9.	Reserves	None

Table 5.1 Balance sheet group

Profit & Loss (or Income & Expenditure) Account Groups

The list of Profit & Loss (or Income & Expenditure) Groups and Sub Groups are given in Table 5.2

Sl.No.	Group Name	Sub-Group Name
1	Direct Income	None
2	Indirect Income	None
3	Direct Expense	None
4	Indirect Expense	None

Table 5.2 Profit and Loss Account Groups

Description of the Groups and Sub-Groups

1. **Capital** : Account(s) of Proprietor or Partners Capital Account(s) or Share Capital account are opened under this group.

   Corpus : Capital fund (in case of Not for Profit organizations).

2. **Current Assets** : Accounts of assets generated in the course of doing business are opened under this group. The sub groups of Current Assets are :

   - Bank: Bank accounts are opened under this sub group.
   - Cash: All cash accounts such as Cash in Hand, Petty Cash, Cash at Warehouse, Cash at Factory .
   - Inventory: Closing Stock and Stock at the Beginning are opened under this sub group.
   - Loans and Advance : Accounts of temporary advances and loans given to staff members, deposits kept for shops, offices, electricity, telephone services etc. are opened under this sub group.
   - Sundry Debtors: Accounts of all customers to whom goods are sold on credit.

3. **Current Liability** :

   The sub groups of Current Liabilities are :

   - Provisions: Accounts of PF, ESI and TDS dues etc.
   - Sundry Creditors for Expense: Accounts of outstanding expenses, such as Salary Payable, Audit Fee Payable etc. are opened under this sub group.
   - Sundry Creditors for Purchase : Under this sub group Accounts of all suppliers from whom goods are purchased on credit.

4. **Fixed Assets**: Accounts of all fixed assets are opened under this group. Thenames of Sub-Groups are Building, Furniture, Land and Plant & Machinery.

5. **Investments**: This contains accounts of investment made by the organization. The two Sub-Groups are Investment in Bank Deposits, and Investments in Shares & Debentures.

6. **Loans (Asset)**: This includes accounts of all long term loans given.

7. **Loans (Liability)**: This contains accounts of loans taken by the organization from banks and other financial institutions. The Sub Groups of loans (Liability) are -

   - Secured (Loan) : contains accounts of loans taken against the security of some property.
   - Unsecured (Loan) : contains accounts of loans taken from partners, Directors, Office Bearers or others, for which usually no Security is given.

8. **Miscellaneous Expenses (Asset)**: This includes Preliminary and pre-formation Expenses to the extent these are not written off.

9 **Direct Income**: Income from sale of goods is included in this group. In case of a service organization income from Fees from Clients or Professional Fees will come under this group. GNUKhata opens Profit & Loss/Income & Expenditure Account under this. This affects Gross Profit.

10 **Indirect Income** : Items included in this group are rent received, interest received, discount received, commission received, dividend received etc. This affects Net Profit.

11 **Direct Expense** : Expenses related with purchase or manufacturing of goods such as purchases, wages, factory lighting, carriage inwards, consumables, etc. are included in this group. GNUKhata opens Opening Stock Account under this group. These expenses affect Gross Profit.

12 **Indirect Expense** : All office, administration, selling and distribution expenses such as office staff salary, office telephone, electricity expenses, maintenance of vehicles, interest on loan, rent, depreciation etc. are included in this group. Such expenses do not affect gross profit but affect net profit.

13 **Reserves** : This contains retained earnings, reserves and surplus.

SYSTEM GENERATED LEDGER ACCOUNTS

In GNUKhata there are four system generated ledger accounts. We can neither change the name nor delete these accounts. Do not create accounts with the similar names. These are: (see Table 5.3)

SL. No.	Account Name	Group Name	Sub-Group Name
1	Closing Stock	Current Assets	Inventory
2	Opening Stock	Direct Expenses	----
3	Stock at the Beginning	Current Assets	Inventory
4	Profit & Loss Account (For Profit Making Organizations:) *Or* Income & Expenditure Account (For Not for Profit Organizations)	Direct Income	-----

Table 5.3 System generated Ledger Accounts

GNUKhata has 29 predetermined Groups and Sub-Groups. Out of these 13 are Groups and 16 are Sub Groups.

25 predefined Groups and Sub-Groups are related with Balance Sheet. Out of these 9 are Groups and 16 are Sub Groups.

4 Groups are related with Profit and Loss Account/ Income and Expenditure Account

BALANCE SHEET GROUPS, SUB-GROUPS AND LEDGER ACCOUNTS

The summary of balance sheet groups, sub-groups and ledger accounts are given in Table 5.4.

Group Name	Sub-Group Name	Ledger Account
(1) Capital / Corpus	None	➢Capital Account ➢Partner's Capital Account ➢Share Capital Account ➢Capital Fund
(2) Current Assets	(1) Bank (2) Cash (3) Inventory	➢Bank Account ➢Cash in Hand ➢Cash Account ➢Petty Cash Account ➢Closing Stock (*System Generated Ledger Account*)

	(4) Loans and Advance	➢Stock at the Beginning (*System Generated Ledger Account*) ➢Short Term Loans and Advances given to Employees ➢Prepaid Expenses
	(5) Sundry Debtors	➢All Debtors/Customers Account
(3) Current Liability	(6) Provisions	➢Account of PF,ESI, TDS dues, etc. ➢Provision for Bad debts ➢Provision for Income Tax
	(7) Sundry Creditors for Expenses	➢Outstanding Expenses
	(8) Sundry Creditors for Purchases	➢All Suppliers / Creditors Account
(4) Fixed Assets	(9) Building	➢Building Account ➢Office Building Account ➢Factory Building Account
	(10) Furniture	➢Furniture Account ➢Shop Furniture Account
	(11) Land	➢Land Account
	(12) Plant & Machinery	➢Machinery Account ➢Plant Account ➢Plant & Machinery Account
(5) Investments	(13) Investment in Bank Deposits	➢Bank Fixed Deposit
	(14) Investment in Shares & Debentures	➢Investment in Shares ➢Investment in Debentures
(6) Loans (Asset)	None	➢Accounts of all Long Term Loans given by the organisation
(7) Loans (Liability)	(15) Secured	➢Bank Loan ➢Other secured loans

	(16) Unsecured	➢Loan from Partners ➢Loan from Manager(s), etc.
(8)Miscellaneous Expenses (Assets)	None	➢Preliminary Expenses ➢Pre-operation Expenses, etc..
(9) Reserves	None	➢Retained Earnings ➢General Reserves ➢Reserve and Surplus

Table 5.4 Summary of Balance Sheet groups

PROFIT & LOSS OR INCOME & EXPENDITURE ACCOUNT GROUPS, SUB- ROUPS AND LEDGER ACCOUNTS

The summary of Profit and Loss account groups and sub groups are given in Table 5.5.

Group Name	Sub-group Name	Ledger Account
(1) Direct Income	None	♦ Sales ♦ Professional Fees ♦ Purchase Return ♦ Profit and Loss Account or Income & Expenditure Account (*System Generated Ledger Account*)
(2) Indirect Income	None	♦ Bad debt received ♦ Commission Received ♦ Discount Received ♦ Income from Investment ♦ Rent Received ♦ Interest Received, etc..
(3) Direct Expense	None	♦ Wages ♦ Carriage Inward ♦ Coal,Gas & Water of Factory ♦ Factory Expenses(Lighting,Power,etc) ♦ Freight ♦ Import Duty ♦ Octroi ♦ Factory Expenses ♦ Opening Stock Account (*System Generated Ledger Account*) ♦ Purchases ♦ Sales Return, etc..

(4) Indirect Expense	None	♦ Office Expenses ♦ Salary ♦ Rent ♦ Insurance ♦ Audit Fee ♦ Electricity ♦ Depreciation ♦ Bad debt ♦ Telephone Charge ♦ Commission Allowed ♦ Discount Allowed ♦ Export Duty ♦ Interest on Loan ♦ Legal Expenses ♦ Postage and Telegram ♦ Printing and stationery, etc..

Table 5.5 Summart of Profit and Loss A/c groups and sub groups

CREATING LEDGER ACCOUNTS

A Ledger account contains a record of all transactions relating to an asset, liability, capital, an item of expenditure or revenue. It has to be created under any of these group. Depending upon the group under which a Ledger account is created, the balance of the account will appear either in Balance sheet or in Profit & Loss Account.

To create a Ledger account, select **Create Account** option from **Master** menu**.** (SeeFigure 5.13)

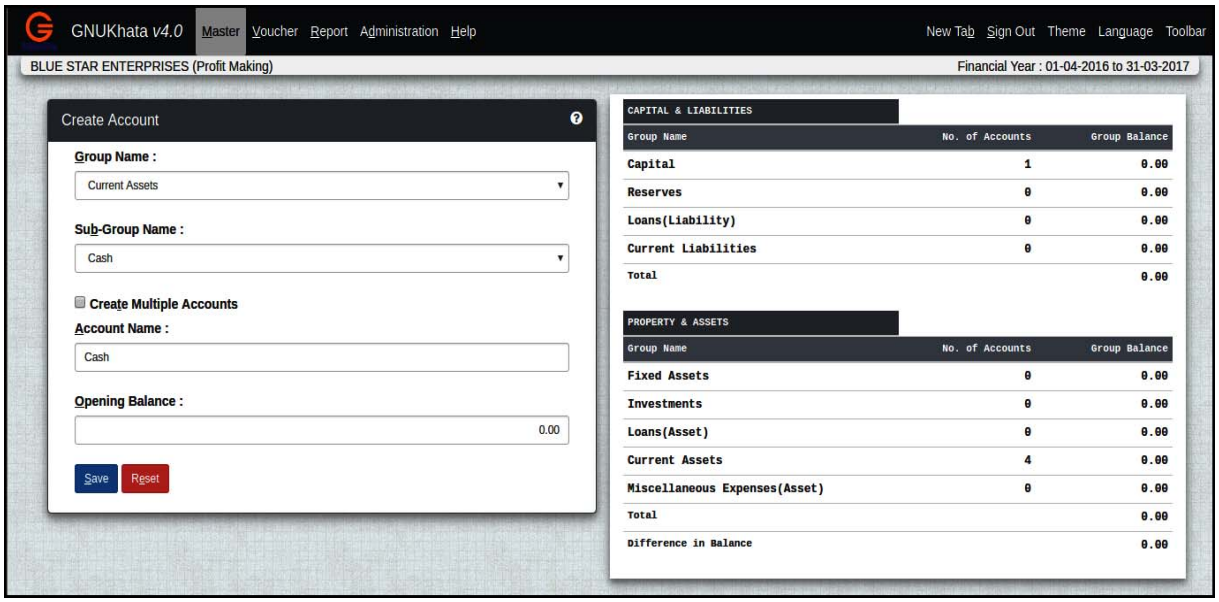


Figure 5.13 Creation of Ledger Account

We can also create multiple accounts under the same Group and Sub-group. To do this click the check box **Create Multiple Accounts**. When we do this a new dialogue box appears. (Figure 5.14) and enter the details of multiple acccounts.

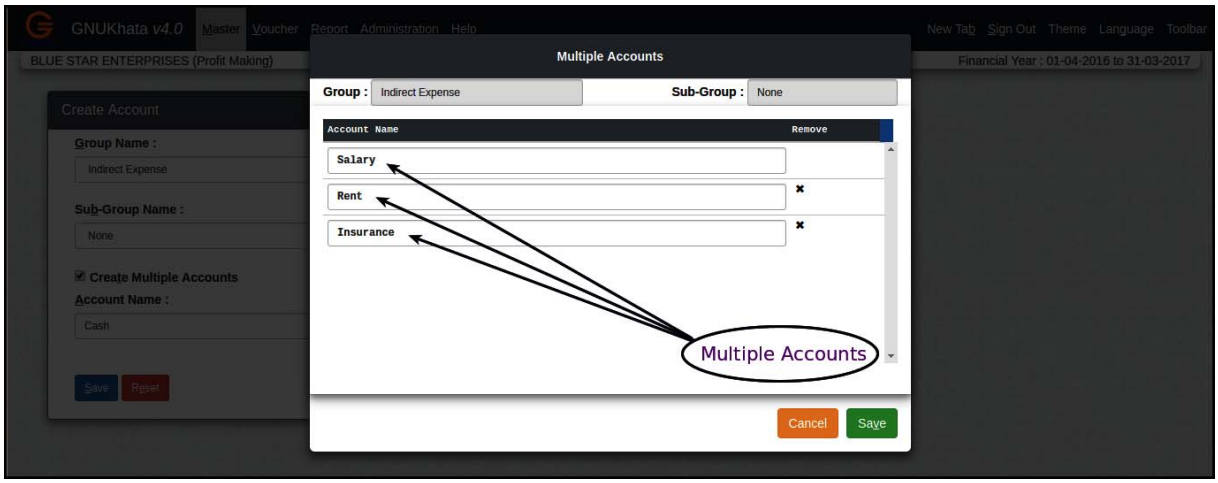


Figure 5.14 Multiple ledger creation

In GNUKhata while creating a Ledger account we have to enter opening balance only if the account belongs to a Balance Sheet Group. This means if an account belongs to Direct and Indirect Income or Expenditure Group, GNUKhata will not ask to enter opening balance because accounts from these groups can never have closing and opening balance.

When creating a Ledger account a new panel with a table is seen on the right side of the screen. In this table Groups are divided into two parts, Capital & Liabilities and Property & Assets. The number of accounts opened under each group and the total of opening balances of accounts are displayed. The difference in Balance, if any, is shown at the bottom of the panel. (See Figure 5.15).

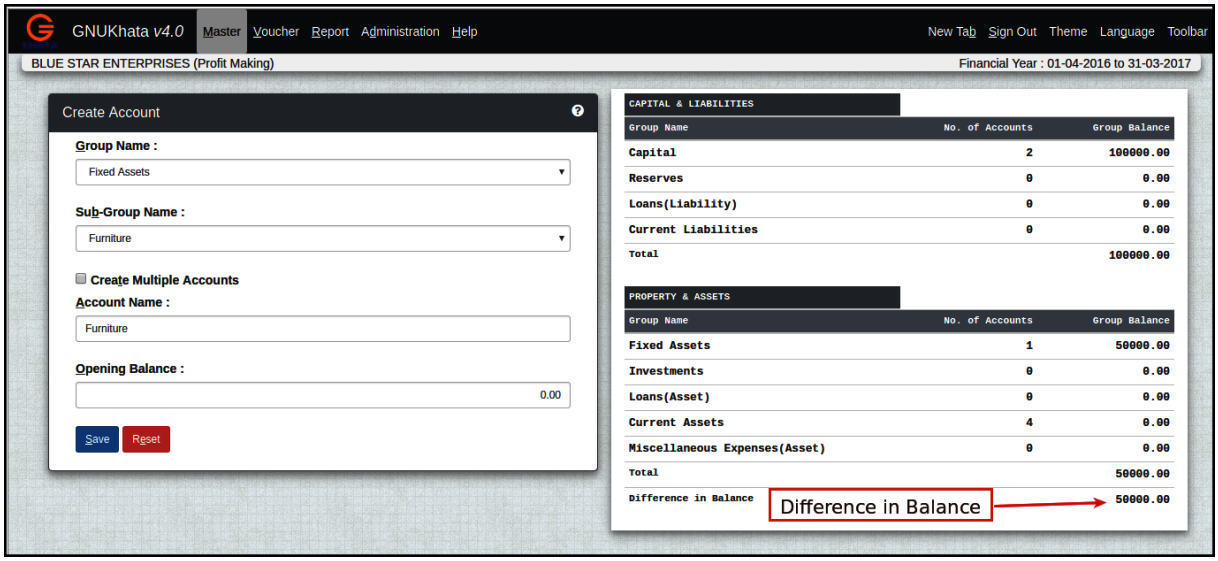


Figure 5.15 Ledger Creation with Balance

CREATION OF NEW SUB-GROUP

Select **Create Account** option from **Master** menu. Then select Group from **Group Name** and **New sub-group** from **Sub-group Name.** Enter the name of sub-group in **New Sub-Group Name,** name of account in **Account Name** and opening balance, if any, then Click on **Save**.

DISPLAY LEDGER ACCOUNTS

To display ledger accounts, select **List of Accounts** from **Report** menu. Now we can see a Table containing a list of accounts along with Group Name and Sub-Group.

EDITING A LEDGER ACCOUNT

To edit a Ledger account, select **Edit Account** from **Master** menu. Here we can change Account Name and Opening Balance, if any, but cannot change the name of Group and Sub-Group.

DELETING A LEDGER ACCOUNT

To delete a Ledger account, select **Edit Account** from **Master**menu. Select the ledger Account we want to delete, click on **Delete** Button and confirm the deletion. System generated account and the ledger account already used in voucher cannot be deleted.

TYPES OF VOUCHERS

GNUKhata has the following pre-defined voucher types as shown in Table 5.6. We cannot create a new Voucher Type.

Voucher Type	Shortcut Key(s)	Used for Recording
Receipt Payments	F4	Amounts received by way of cash or cheque.
Payment	F5	Payments made by cash or through bank.
Sales	F6	Cash and credit sale of goods.
Purchase	F7	Cash and credit purchase of raw materials, semi-finished goods and finished goods.
Contra	F8	Deposits or withdrawals of cash from bank or transfer of funds from one bank to another, transfer of cash to Pretty Cash (contra entry).
Journal	F9	Rectifying entries or transfer or adjustment entries, purchase of fixed assets on credit.
Sales Return	Ctrl+1	Return of goods by a customer
Purchase Return	Ctrl+2	Return of goods to a supplier
Credit Note	Ctrl+3	Reduction in the price charged to a customer.
Debit Note	Ctrl+4	Reduction in the price given by a supplier.

Table 5.6 Pre-defined Voucher Types

VOUCHER ENTRY

Recording a transaction through voucher is called voucher entry. While recording a voucher the debit part of the transaction recorded first and there after credit part, then we can add any number of debits and credits.

Procedure for voucher entry :

Step 1 : Select the appropriate voucher from **Voucher** menu
Step 2 : Enter voucher number and date
Step 3 : Select the **Debit** account name and enter the amount, then press **Enter**
Step 4 : Select the **Credit** account name and enter the amount
Step 5 : Enter narration and click on **Save** (see Figure 5.16)

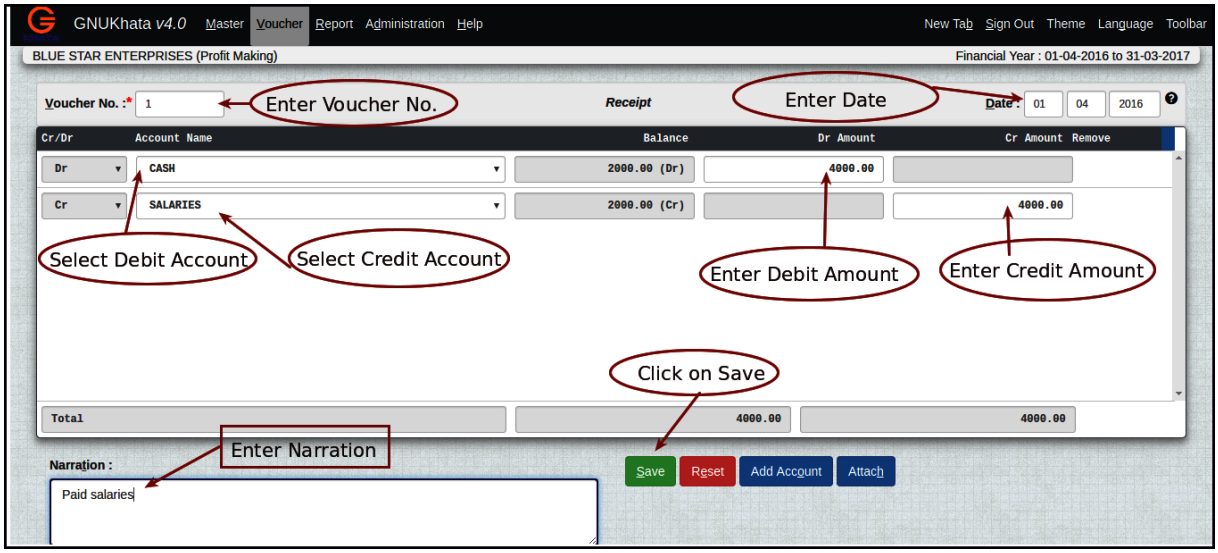


Figure 5.16 Voucher Entry

ENTERING OPENING STOCK

Step 1 : Edit the *Stock at the Beginning Account*
Master→Edit Account → Select *Stock at the Beginning Account* → Click on **Edit** →Enter Opening Balnace
Step 2 : **Voucher** → Select **Journal** voucher → Enter Voucher No & Opening date → Debit *Opening Stock* and credit *Stock at the Beginning*

ENTERING CLOSING STOCK

Step 1 : Open **Journal** voucher → Enter Voucher No: and Closing date → Debit *Closing Stock* and credit *Profit & Loss*

FINDING AND EDITING A VOUCHER

Select **Find/Edit Voucher** from **Voucher** menu. Select **Criteria** to search the required voucher. All transactions fulfilling that criteria will be displayed. Select the transaction that you want to edit and press **Enter** Key. The voucher will appear and click on Edit button. See Figure. 5.17 Make changes in the record and save.

CLONING A VOUCHER

To clone a voucher means to copy a voucher. Where a number of similar entries are to be made Cloning module is used. For this find the record you want to clone and click on Clone Button to open Clone Module. Make changes as you desire and save. The new record will be added to the data.

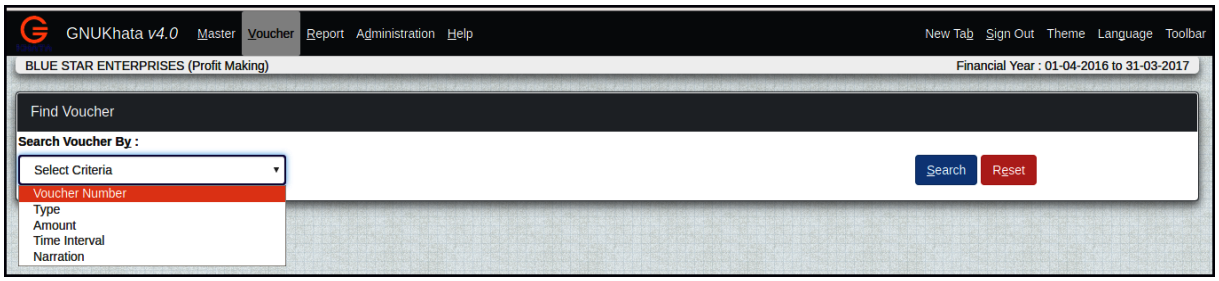


Figure 5.17 Find Voucher

DELETING A VOUCHER

To delete a voucher first find it and click on **Delete,** after confirmation, the record will be deleted. Deleted vouchers cannot be restored, we can view a **List of Deleted Vouchers** through **Report** menu.(See Figure 5.18 and 5.19)

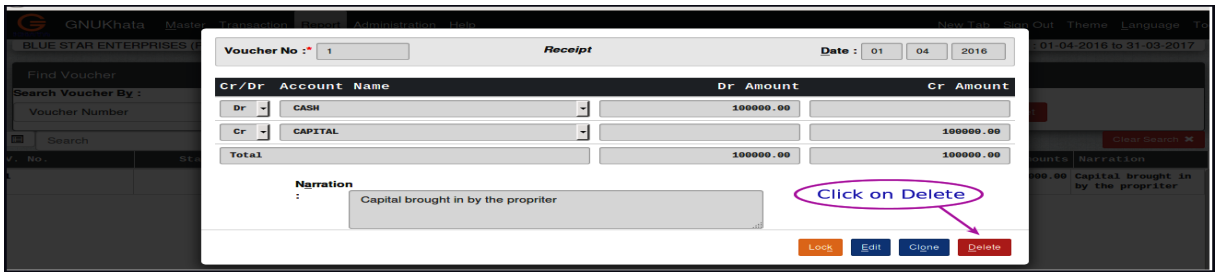


Figure 5.18 Delete Voucher

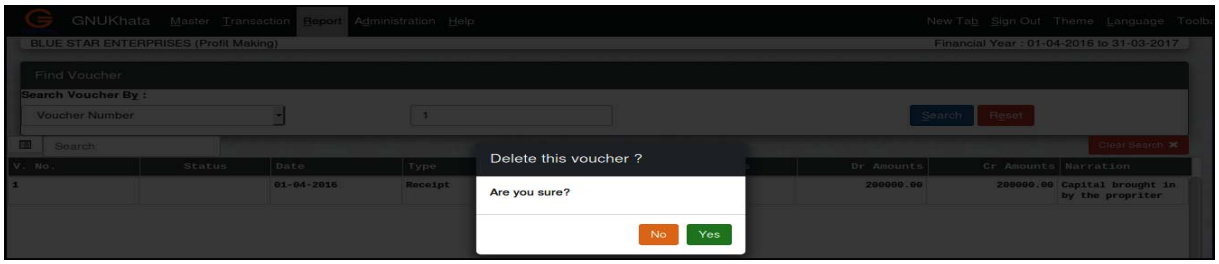


Figure 5.19 Delete Voucher Confirmation

ADD ACCOUNT WHILE IN VOUCHER ENTRY

While recording transaction in voucher entry mode, we can add a Ledger Account by clicking on **Add Account,** Select the Group, Sub-Group, enter account name and Opening Balance, if any. Click on **Save** Button and return to voucher entry mode. See Figure 5.20.

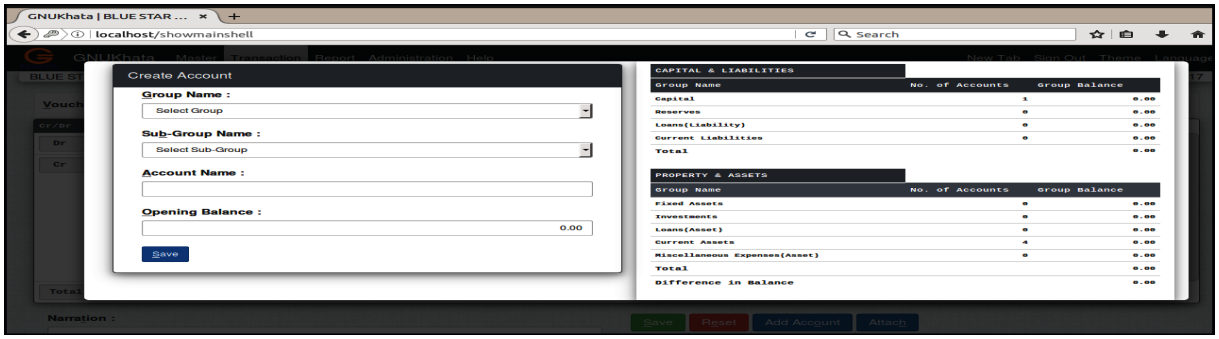


Figure 5.20 Add coount while in voucher entry

ATTACH DOCUMENT

This unique feature enables the user to attach scanned copies of the supporting documents to the record of a transaction. The attachments can be viewed, deleted and replaced. This unique feature is most useful while auditing, which enables a scanned copy of a supporting document to be viewed instantly.

REPORTS

From the **Report** menu, we will enable to view reports such as Ledger, Trial Balance, Balance Sheet, Profit & Loss Account, List of Accounts and List of Deleted Vouchers.

For all these reports you have to specify the periods.

Example:

Enter the following transactions of Royal Enterprises and Display trial Balance , Profit and Loss account and Balance sheet

Date	*Transaction*	*Amount*
1-1-2016	Commenced business with cash	100,000
2-1-2016	Opened Bank account	50,000
3-1-2016	Purchased Furniture	20,000
4-1-2016	Purchased goods from Anil Traders	10,000
5-1-2016	Wages Paid	1,000
6-1-2016	Rent paid	2,000
7-1-2016	Sold goods to Raju	5,000
8-1-2016	Received Commission	2,000
9-1-2016	Returned goods to Anil	900
10-1-2016	Goods return by Raju	700

Adjustments :

1. Closing stock valued at Rs.6000 as on 31-12-2016
2. Depreciation furniture at 10%
3. Wages outstanding Rs.500

Solution:

Step 1: Open GNUKhata

Applications → Office → GNUKhata

Step 2: Create Organisation

Click on 'Create Organisation' → Enter details

Step 3: Create Admin

Enter the details

Step 4: Create appropriate ledger account

Master → create account → Select Group → Sub group → Type Account name → Save

Date	Account Name	Group	Sub Group	Voucher	Key	Dr/ Cr	Amount
1-1-16	Cash Capital	Current Asset Capital	Cash None	Reciept	F4	Dr Cr	100000 100000
2-1-16	Bank Cash	Current Asset Current Asset	Bank Cash	Contra	F8	Dr Cr	50000 50000
3-1-16	Furniture Cash	Fixed Asset Current Asset	Furniture Cash	Payment	F5	Dr Cr	20000 20000
4-1-16	Purchase Anil Traders	Direct Expenses Current Liabilities	None Sundry Creditions for purchase	Purchase	F7	Dr Cr	10000 10000
5-1-16	Wages Cash	Direct Expenses Current Asset	None Cash	Payment	F5	Dr Cr	1000 1000
6-1-16	Rent Cash	Indirect Expenses Current Asset	None Cash	Payment	F5	Dr Cr	2000 2000
7-1-16	Raju Sales	Current Asset Direct Income	Sundry Debtors None	Sales	F6	Dr Cr	5000 5000

8-1-16	Cash Commision Received	Current Asset Indirect Income	Cash None	Reciept	F4	Dr Cr	2000 2000
9-1-16	Anil Traders Purchase Return	Current Liabilities Direct Income	Sundry creditors for purchase None	Purchase Returns	Ctrl +2	Dr Cr	900 900
10-1-16	Sales return Raju	Direct Expense Current Asset	None Sundrey Debtors	Sales Return	Ctrl +1	Dr Cr	700 700
31-12-16	Closing stock Profit & Loss A/c.	System generated account		Journal	F9	Dr Cr	6000 6000
31-12-16	Depreciation Furniture	Indirect Expenses Fixed asset	None Furniture	Journal	F9	Dr Cr	2000 2000
31-12-16	Wages Outstanding Wages	Direct Expenses Current Liabilities	None Sundry Creditors for expenses	Journal	F9	Dr Cr	500 500

Step 5 : Enter Vouchers (based on the above table)
Transaction / Voucher → Select appropriate voucher → Enter details

Step 6 : Display Trial Balance
Report → Trial Balance (See Figure 5.21)

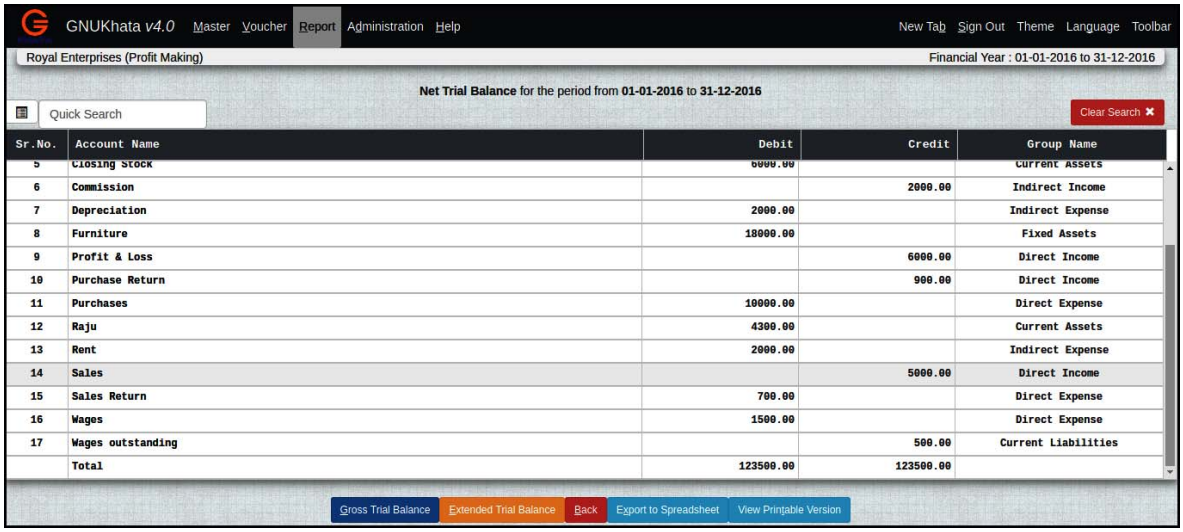


Figure 5.21 Trial Balance

Step 7 : Display Profit and Loss account

Report → profit and Loss account (See Figure 5.22)

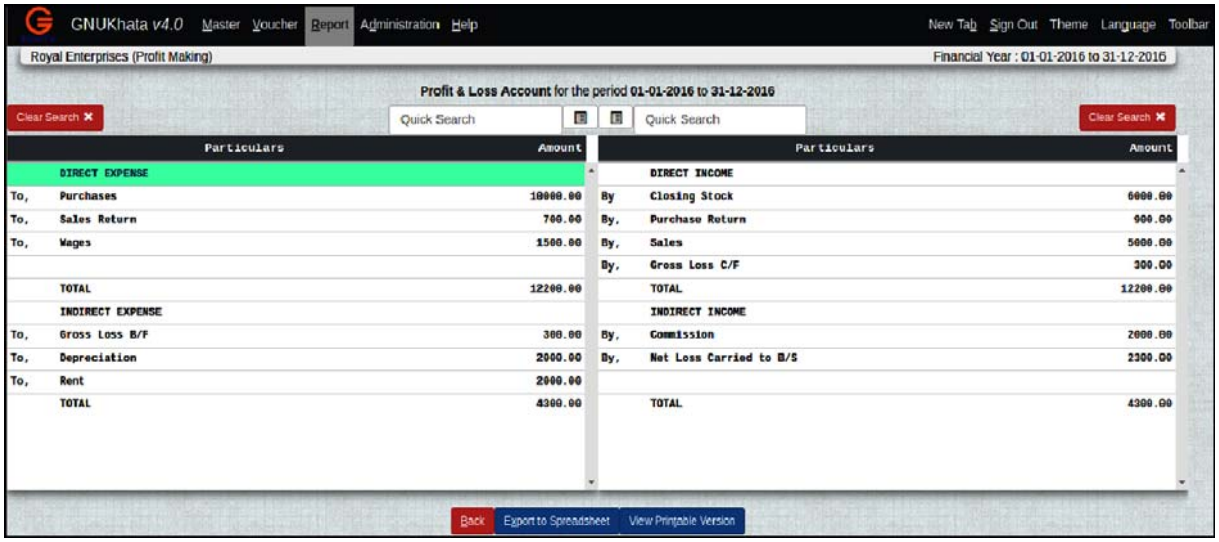


Figure 5.22 Profit and Loss account

Step 8 : Display Balance Sheet

Report → Balance Sheet

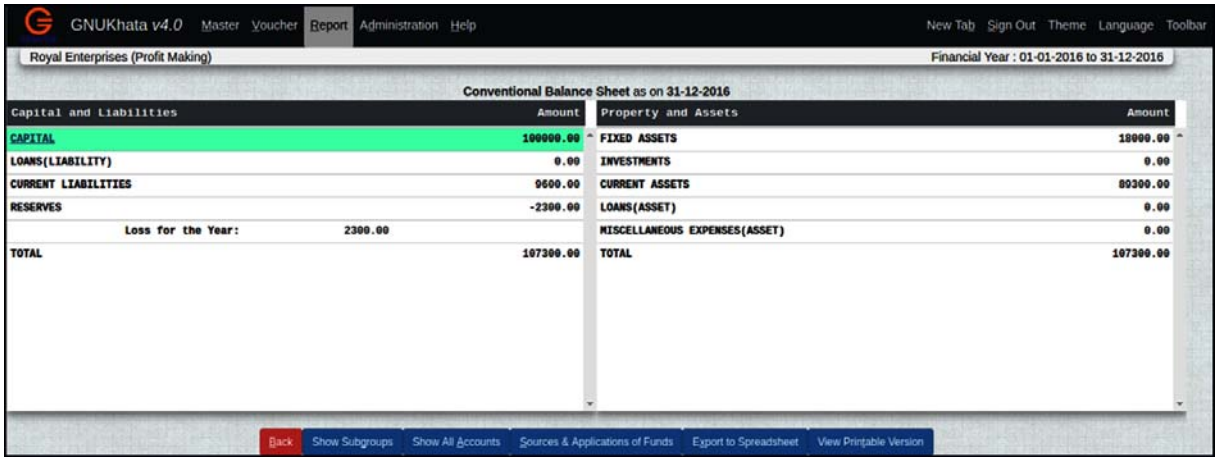


Figure 5.23 Balance sheet

BANK RECONCILIATION STATEMENT

Bank Reconciliation Statement (BRS) is prepared by an account holder to reconcile cash book balance and pass book balance on a specific date. It enables whether all cheques deposited by him are cleared and all cheques issued by him are presented or not. Bank Reconciliation Statement is prepared for bank accounts opened under the Sub-Group Bank of the Group Current Assets.

Bank Overdraft can be opened under the Group Current Asseets, Sub-Group Bank. It can also be opened under the Group Loan (Liabilities), Sub-Group Secured or Unsecured.

If we open it under the Sub-Group Bank, then we can pass contra entries for all such transactions. If we open this account under the Group Loan (Liabilities), we cannot pass Contra entries in this.

The Bank Reconciliation Statement is done by comparing two dates of a transaction. The first is the date of transaction and the second is the date on which the same transaction appears in the bank passbook or bank statement, which date is called the Clearance Date. Enter Clearance Date in that column for reconciliation.

To open Bank Reconciliation Statement, select **Bank Reconciliation Statement** from the **Master** menu

Example :

Enter the following transactions in appropriate vouchers of Blue Star Organisation and Prepare Bank Reconciliation Statement.

Date	***Transaction***	***Amount***
1-4-2013	Cash received from Rajeev, a customer	50,000
10-4-2013	Deposited into Federal Bank	25,000
10-4-2013	Issued cheque (No.1118) to Mohammed, a creditor	12,000
12-4-2013	Sent for collection cheque	13,000
12-4-2013	Paid Electricity bill by cheque (No.1119)	800
20-4-2013	Issued cheque to Santhosh, a creditor	1,000
29-4-2013	Deposited into Federal Bank	5,000

Verification of the Cash Book with Bank Pass Book revealed that the Cheque Nos.1118 and 1119 were cashed only on 1st and 2nd May respectively and the cheque sent for collection was collected only on 1st May 2013.

Solution:

Step 1 : Open GNUKhata - **Applications → Office → GNUKhata**

Step 2 : Create Organisation - Click on '**Create Organisation' → Enter details**

Step 3 : Create Admin - Enter the details

Step 4 : Create appropriate ledger account - **Master → create account → Select Group → Sub group → Type Account name → Save**

Date	Account name	Group	Sub group	Voucher	Key	Dr/Cr	Amount
1-4-2013	Cash Rajeev	Current Assets Current Assets	Cash Sundry Debtors	Receipt	F4	Dr Cr	50000 50000
10-4-2013	Federal Bank Cash	Current Assets Current Assets	Bank Cash	Contra	F8	Dr Cr	25000 25000
10-4-2013	Mohammed Federal Bank	Current Liabilities Current Asset	Sundry Creditorsfor purchases Bank	Payment	F5	Dr Cr	12000 12000
12-4-2013	Federal Bank Cash	Current Assets Current Assets	Bank Cash	Contra	F8	Dr Cr	13000 13000
12-4-2013	Electricity Federal Bank	Indirect Expenses Current Assets	None Bank	Payment	F5	Dr Cr	800 800

20-4-2013	Santhosh Federal Bank	Current Liabilities Current Asset	Sundry Creditors for purchases Bank	Payment	F5	Dr Cr	1000 1000
29-4-2013	Federal Bank Cash	Current Assets Current Assets	Bank Cash	Contra	F8	Dr Cr	5000 5000

Step 5: Enter Vouchers (based on the above table) - Transaction / Voucher → Select appropriate voucher → Enter details

Step 6: Enter Clearance Date - Master-> Bank Reconciliation Statement → Select Account Name (Federal Bank) → Set the Reconciliation Period(From Date and To date) → Click on view → Enter Clearance Date (See Figure 5.24)

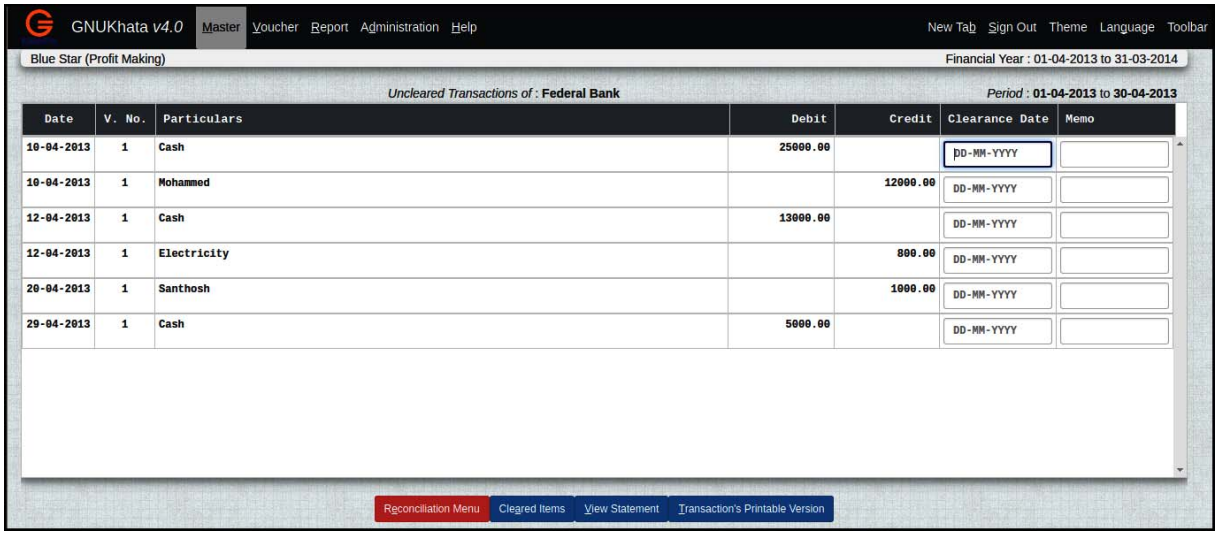


Figure 5.24 Clearence Date

Step 7: Show the Bank Reconciliation Statement - Click on **View Statement.** (See Figure 5.25)

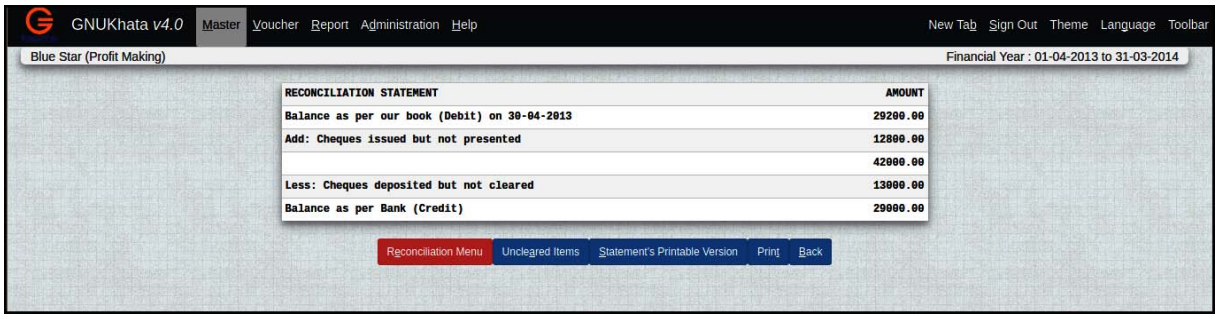


Figure 5.25 Bank Reconciliation Statement

Step 8: To show the Cleared Transactions, Click on Cleared Items (See Figure 5.26)

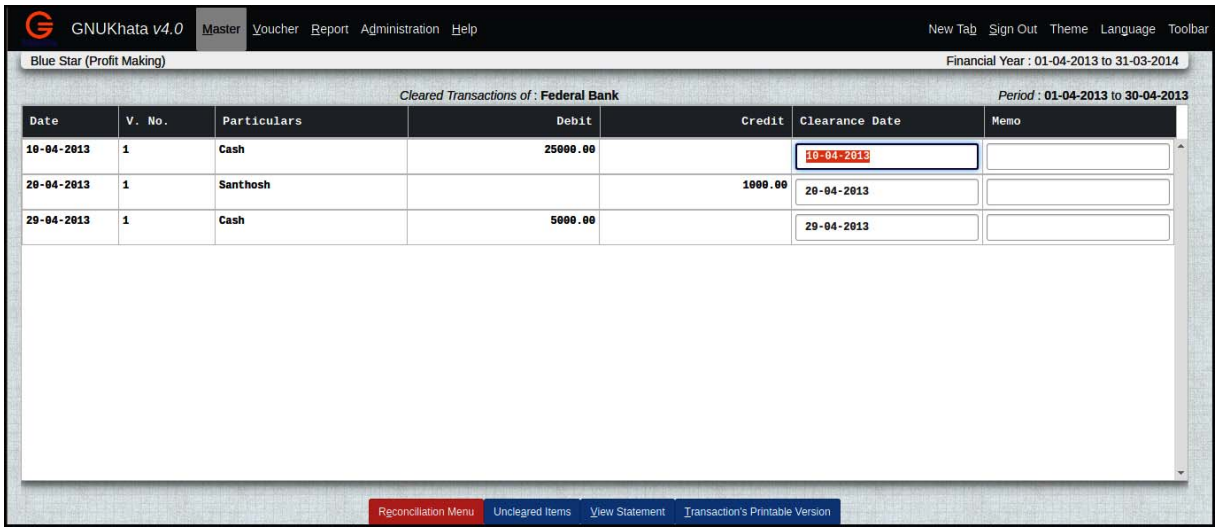


Figure 5.26 Cleared Transactions

Step 9 : To show the Uncleared Transactions, Click on Uncleared Items (See Figure 5.27)

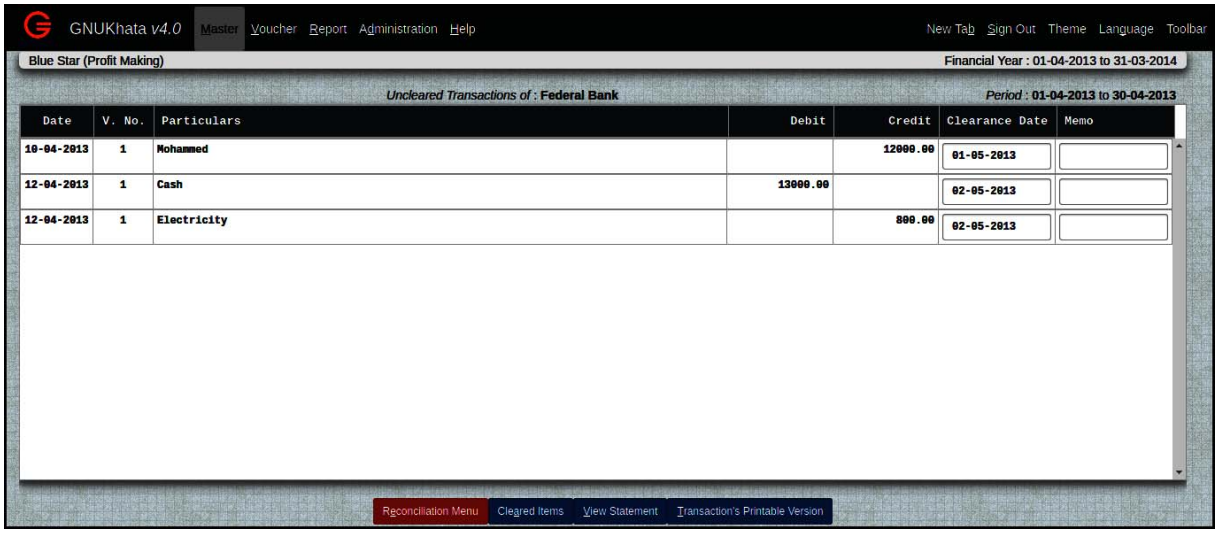


Figure 5.27 Uncleared Transactions

Model Questions:

Fill the table with suitable terms:

Account name	Group / Under	Voucher	Dr/Cr
Cash Capital	---------- Capital	Receipt	Dr Cr
Bank ----------	Current asset Current asset	Contra	Dr Cr
Furniture Cash	Fixed asset Current asset	----------	Dr Cr
Wages Cash	---------- Current Asset	Payment	Dr Cr
Depreciation Furniture	Indirect Expenses Fixed asset	----------	Dr Cr

2.

Account Name	Group / Under	Voucher	Dr/Cr
Cash Rajeevi	Current Assets Current Assets	.....	Dr Cr
Federal Bank Cash	..... Current Assets	Contra	Dr Cr
Federal Bank Commission	Current Assets	Reciepts	Dr Cr
Electricity Federal Bank	Indirect Expenses Current Assets	.....	Dr Cr
..... Cash	Current Assets Current Assets	Contra	Dr Cr

3. Anoop created 6 Ledger accounts in Royal Traders. But, while displaying the list of ledger account, he found the name of ten 10 ledger account in the displayed list.

   a) Can you explain the reason ?

   b) Name the additional ledger account displayed in the list.

   c) Give the procedure for displaying ledger accounts.

4. Identify the Group and Sub-group under which the following ledger accounts are to be created.

   a) Depreciation

   b) Carriage Inward

   c) Bills Receivable

   d) Drawing

5. Explaining the procedure and steps involved in the following under GNUKhata

   a) Creating an organisation

   b) Selecting an Organisation

   c) Deleting an Organisation

6. Write down the procedure for the creation and deletion of Ledger account in GNUKhata

7. Write a notes on :

   a) Contra Voucher

b) Debit Note Voucher

c) Credit Note Voucher

8. Explain the procedure for voucher entry in GNUKhata.

9. Give the Voucher type and the key to be selected to the following transactions.

i) Opened Bank account with Rs.5000.

ii) Salary paid Rs.2000.

iii) Depreciation Charged on Furniture Rs.1200.

iv) Returned goods to Bineesh , a supplier Rs. 1400.

10. Explain the procedure for Bank Reconciliation Statement in GNUKhata ?

APPENDIX

Shortcut Keys used in GNUKhata

Index Page

Select Existing Organisation - CTRL + SHIFT + E

Create Organisation - CTRL + SHIFT + R

Menu

Master - CTRL + SHIFT + M

Voucher - CTRL + SHIFT + V

Report - CTRL + SHIFT + R

Administration - CTRL + SHIFT + D

Help - CTRL + SHIFT + H

New Tab

Sign Out -CTRL + SHIFT + S

Toolbar - F1

Sub-Menu

Logout - CTRL - L

Change Organisation - CTRL + O

Bank Reconciliation Statement - ALT + R

Create Account - F2

Find & Edit Account - F3

Receipt - F4

Payment - F5

Sale - F6

Purchase - F7

Contra - F8

Journal - F9

Sale Return - CTRL + 1

Purchase Return - CTRL + 2

Credit Note - CTRL + 3

Debit Note - CTRL + 4

Find & Edit Voucher - F10

Ledger - F11

Trial Balance - F12

List of Accounts - CTRL+7

Balance Sheet - CTRL+8

Profit & Loss - CTRL+9

New User - ALT + N

Remove User - ALT + U

Delete Organisation - ALT + I

Chapter 6

DATABASE MANAGEMENT SYSTEM FOR ACCOUNTING

INTRODUCTION

One of the major factors for realising the need of Computerised Accounting System is the overwhelming quantity of data in our organisations. The conventionally used paper filing system, text documents, and even spread-sheets may not suffice for the growing needs of tracking this voluminous and critical information. A simple solution to this situation is available in the form of a Database Management System (DBMS) (e.g. 'Base', 'Access', 'Oracle', 'SQL Server', etc.) that provides a variety of software tools for organising, processing and querying data in a flexible manner. As we now proceed to look into practical applications of computers in generating, storing, processing, and retrieving of accounting information, we will make an assumption that you are fairly conversant with the accounting framework and operating procedure – i.e. you have the required domain knowledge. We also assume that you have adequate exposure to handling of computers and the concepts of database. However, we do not wish to make knowledge of the database management a pre-requisite for understanding of this chapter, and hence, we will restrict ourselves to the simpler and easy to comprehend 'LibreOffice Base' program for developing some practical accounting applications. In doing so, we will focus on the three major components of Base, namely, 'tables', 'queries' and 'forms'. Having done that, we will examine the methods of generating reports. This basically builds on the concepts of DBMS already learnt by you in class XI.

6.1 UNDERSTANDING AND DEFINING THE DATABASE REQUIREMENT

With the continuous improvements in computer's processing speed, storage capacity, networking techniques, operating systems, etc, the capabilities of computer applications have also gone up many folds. Various computer applications that are commercially available today not only provide fairly comprehensive tools for all conceivable needs but also have become extensively user friendly. So, when we look forward to putting into use database applications such as 'Base', we really do not need much of programming skills. Nevertheless, any programming knowledge may improve our efficiency and effectiveness in handling such applications. On the other hand, before we develop any database application, we ought to have a complete understanding of our requirements expected from the application. This is one area where application itself may not extend much help. Further, the correct understanding of our requirement also has a bearing on the choice of Database Management Systems (DBMS) - i.e. whether to go for a 'desktop database' or to choose the 'server database'.

In most of the cases, database is not directly accessible to users. Any addition, modification or retrieval of information from database is done by the user-friendly programs. Database is thus rightly referred to as 'back-end' while the interactive program is termed as 'front-end' of a database application.

While the desktop databases – residing on standard personal computers – are oriented toward single-user applications, 'server databases' are geared toward multi-user applications. Understandably the 'server database', containing additional provisions for ensuring reliability and consistency of data in a multi-user environment, are substantially costlier than the 'desktop database'. Hence, it is imperative to do a careful analysis of our requirements before investing in a database solution. Some of the questions that need to be answered in this regard are:

- What all data is required to be stored in the database?
- Who will capture or modify the data, and how frequently the data will be modified?
- Who all will be using the database, and what all tasks will they perform?
- Will the database (backend) be used by any other front-end application?
- Will access to database be given over LAN (Local Area Network)/ Internet, and for what purposes?
- What level of hardware and operating system is available?

If our requirement entails approach to database by many users, or if it involves simultaneous updating of data by different users, then perhaps server databases such as 'MySQL', 'Oracle' or 'IBM DB2' will provide desired solutions. A server -based database can give you several advantages in terms of flexibility for the choice of front-end applications, powerful performance independent of platform, and scalability to handle rapidly expanding number of users as well as amount of data.

Quite often, same database is required to be used by different applications. For example, an organisation may have automated attendance recording system which will record the arrival and departure time of employees. A variety of such systems (e.g. Card- swipe, Biometric, etc.) are available in the market, and all of them come with their own (back-end) database and requisite (front-end) programs. This attendance information may be required for generating salary bills of the employees. The payroll application of the organisation, with its own (back-end) database and (front-end) program, will also access the database of attendance recording application.

Comparatively, the desktop databases offer an inexpensive and simple solution to many of our business data storage and processing requirements. OpenOffice.org offers a very powerful database system with Base. Base is a powerful and flexible application, you need to be able to make some informed decisions while working with it and this, in turn, requires some preparation. Even on same version of the 'Base', marginal differences may appear in the illustrated diagrams and the views on your computer and this may happen on account of variations in the customised settings of the application. However, these variations are not expected to hamper your understanding of the contents of this chapter.

IDENTIFICATION OF DATA TO BE STORED IN TABLES

By now we understand that a database is simply an organised collection of data with 'tables' as its fundamental building blocks. Tables allow us to create the framework for storing information in the database. Each column (also called 'field') of the table corresponds to a specific characteristic (or 'attribute' in database terms) of the stored information. Each row (also called 'record') corresponds to a particular instance of the information.

In LibreOffice Base, the entire database is encompassed in a file (with extension .odb) which can be stored in your hard-drive or CD. There can be multiple tables (each storing a specific set of data) with in this file. Then there are multiple fields in each table according to different categories (types) of data within the table. And at every instance (occurrence) of a collection of data covering all fields, a row/record gets created.

A set of tables often with well established relationships between them constitutes the database covering total spectrum of stored information. The term 'database design' can be used to describe the structure of different parts of the overall database.

Identification of various attributes of a database is generally considered as part of requirement analysis. Such a process may entail database designer to elicit needed information from those with the domain knowledge.

We may get a better understanding of database design by referring to practical example of an accounting problem. We may take up the case of payroll accounting, concepts of which have already been examined while studying spreadsheet applications in the earlier chapter. You have already seen how different employees draw different amounts of pay based on the nature and levels of their employment, even while subjected to a definite pattern of 'pay rules' prevalent in the organisation. Since pay rules vary widely from organisations to organisations, the database design for them will also differ considerably. For our illustration in this chapter, we will consider the simplified pay pattern for an assumed organisation XYZ Pvt. Ltd. Let us begin by identifying various attributes of information that are required to be stored in our Payroll database. We may have a set of attributes pertaining to employee's personal details such as: 'Employee ID', 'Name', 'Designation', and 'Location'. On the other hand we may also deal with such attributes pertaining to employee's pay as: 'Basic Pay', 'Dearness Allowance (DA)', 'House Rent Allowance (HRA)', 'Transport Allowance (TA)', 'Provident Fund (PF) Deduction', etc. We may also want to know the attributes of 'Gross Salary' and 'Net Salary' which is obtained by subtracting 'PF Deductions' from 'Gross Salary'. However, 'Gross Salary' and 'Net Salary' attributes may not require being stored in the database as they are merely computational outcomes from other attributes. We may also require some attributes concerning pay formulations such as '% Rate of DA' which may fluctuate month to month, '% Rate of HRA' varying with the location of employee, and 'TA Slabs' varying with the designation of employee.

At this stage you may like to know the limitations of 'Calc' in payroll accounting. If you look back at payroll example in Calc spreadsheet, you will notice that you have created pay information of a set of employees for a particular month. For generating information concerning

next month, you will have to change employee specific data such as 'Actual Basic Pay', 'days of attendance', etc in respective cells. However, the moment you do this, the information of previous month will be lost. Understandably, 'Calc' is merely providing a template for certain calculations concerning salaries, and storing single (monthly) instance of the derived information. Calc eliminates redundant data and hence multiple instances of information cannot be stored in a single Speadsheet. You may avoid this situation by effecting changes in a different copy of the sheet. In this manner you are creating different spread sheet for different months. This is a cumbersome task, and furthermore the information in such a case will not be residing in a single table, or a set of related tables, from where it may be retrieved for generation of 'annual income' or 'income tax' information.

Now the main question is how to store these attributes in our database tables. Shall we make all of above attributes as part of one table, or shall we opt for multiple tables. The 'Calc' as such will not put any constraint on the number of tables we opt for, or the type of data we chose to put in any table. It will have to be entirely our decision, based on logical structuring of data that we seek to apply. You may also apreciate the fact that the purpose of a database is not so much for the storage of information, as for its quick retrieval. Hence, you ought to structure your database in such a manner that it can be queried quickly and efficiently.

LOGICAL STRUCTURING OF DATA IN TABLES

Month & Year	Emp Name	Desig.	Locat.	%DA Rate	%HRA Rate	Basic	DA	HRA	TA	Gross Salary	PF Ded.	Net Salary
Nov. 2007	Ram Kishore	Chief Manager	Delhi	26	30	25000	6500	7500	7000	46000	5000	41000
Nov. 2007	Kishan Sharma	Manager	Faridabad	26	20	22000	5720	4400	5000	37120	3000	34120
Nov. 2007	Rupali Varma	Senior Engineer	Meerut	26	15	20000	5200	3000	3500	31700	2000	29700
Nov. 2007	Surjeet Singh	Engineer	Meerut	26	15	16000	4160	2400	3500	26060	2000	24060
Dec. 2007	Ram Kishore	Chief Manager	Delhi	26	30	25000	6500	7500	7000	46000	7000	39000
Dec. 2007	Kishan Sharma	Manager	Faridabad	26	20	22000	5720	4400	5000	37120	3000	34120
Dec. 2007	Rupali Varma	Senior Engineer	Meerut	26	15	20000	5200	3000	3500	31700	3000	28700
Dec. 2007	Surjeet Singh	Engineer	Meerut	26	15	16000	4160	2400	3500	26060	3000	23060
Jan. 2008	Ram Kishore	Chief Manager	Delhi	30	30	25000	7500	7500	7000	47000	6000	41000
Jan. 2008	Kishan Sharma	Manager	Faridabad	30	20	22000	6600	4400	5000	38000	2000	36000

Repeating personal data — Pay formulation parameters — Pay data

Figure 6.1 Single table for all attributes

Suppose, we seek the simplest structure of a single table containing all attributes listed above. The rows of this table (see Figure-6.1) will contain both employee's personal details as

well as employee's pay details for every instance of the salary drawn by him or her. Since, the information will be generated on monthly basis, multiple records will be created for each employee in the table. Understandably for any particular employee, we may expect variations in the pay attributes recorded under different rows - i.e. different pay components and the resulting gross/net salary may vary from month to month. However, we do not expect monthly variations in the personal attributes of the same person under different rows. Evidently thus, the structure of single table may not be an efficient one as it will entail unwarranted recording of the same personal data multiple times.

With the understanding of the limitations of a single table, let us now examine a better approach involving creation of multiple tables. For example, we may have separate tables for personal attributes (i.e. name, designation, location etc.), pay formulation parameters (i.e. % DA rate, % HRA rate, etc.) and the pay attributes (i.e. basic pay, DA,Database Management System for Accounting HRA, TA, etc.). Personal details table may be used for recording personal attributes of every employee, consuming only one row of data (i.e. record) for each employee. The pay details table may contain records of pay attributes for all employees, with a new row of data created for an employee for every instance of the monthly salary drawn by him or her. Other tables may also be created for recording different parameters used in pay formulations of employees. Thus, we may have separate tables for storing the '% DA rate' for different months, '% HRA rate' for different locations, and the fixed quantum of 'Transport Allowances' for different designations of employees.

In order to establish relationship between any two tables, we will have to insert columns with matching values in the two related tables. Thus, we may have 'employee name' appearing in both 'personal table' as well as in 'pay table'. After establishing a relationship between these two tables, rows with a common value in 'employee name' field of both the tables get related. The two common fields used in a relationship between tables are called the key fields. In our example, any 'employee name' in the personal table appears only once (i.e. unique row); and hence this field of personal table is a primary key of the relationship. On the other hand, an 'employee name' in the pay table appears in multiple rows; and hence this field of pay table is a foreign key of the relationship. The Figure-6.2 below illustrates the relationship of some such tables, wherein the lines connecting rows of two tables represent relationships.

Emp Name	Desig.	Locat.
Ram Kishore	Chief Manager	Delhi
Kishan Sharma	Manager	Faridabad
Rupali Varma	Senior Engineer	Meerut
Surjeet Singh	Engineer	Meerut

Locat.	% HRA Rate
Delhi	30
Faridabad	20
Meerut	15

Month and Year	% DA Rate
Nov. 2007	26
Dec. 2007	26
Jan. 2008	30

Month & Year	Emp Name	Basic	DA	HRA	TA	Gross Salary	PF Ded.	Net Salary
Nov. 2007	Ram Kishore	25000	6500	7500	7000	46000	5000	41000
Nov. 2007	Kishan Sharma	22000	5720	4400	5000	37120	3000	34120
Nov. 2007	Rupali Varma	20000	5200	3000	3500	31700	2000	29700
Nov. 2007	Surjeet Singh	16000	4160	2400	3500	26060	2000	24060
Dec. 2007	Ram Kishore	25000	6500	7500	7000	46000	7000	39000

Figure 6.2 Multiple tables for different attributes

Avoiding duplication of information is key criteria of database design, which is achieved by breaking up of information into separate but related tables; and this process is called ***normalisation****. We will also have to establish links between different tables so as to reconstruct the original information; and these links in database terms are called* ***relationships****. The database created on the basis of such relationships between different data tables is called* ***relational database****.*

The process of matching rows in two tables based on their primary and foreign keys is called a Join. Joins along with Structured Query Language (SQL) serve as valuable tools for manipulating tables. But, these topics are beyond the scope of this book, and hence not elaborated further.

For our practical example, let us take the multiple-tables route. We will have altogether five tables. Our first table, with the name 'TabEmpDetails', will store employee's personal attributes in the fields of 'EmpName', 'Designation ID' and 'Location ID'. In addition we will also have a field of 'EmpID' for storing unique values of employee number. So this field will be identified as Primary Key for the 'TabEmpDetails' table; and thereby database will not permit storing of two similar values in this field. The 'DesignationID' field will merely store the ID of employee's Designation, further details of which will be stored in a separate table. Understandably, this field will have multiple repetitions of some of the values. This field will form one of the foreign keys of 'TabEmpDetails' table with the establishment of its relationship with the table containing details of Designations. The 'LocationID' field will get similar treatment with another table containing details of Locations.

You may notice that we have used combinations of shortened words for identification of table name and the names of its different fields. This is done so as not to make headings of the table unnecessarily long, at the same time leaving sufficient scope for easy identification of intended attributes. You may also mark that no blank (gap) is left in between a name. We have also used both upper and lower cases for improving our readability of these names, though LibreOffice Base program as such is case insensitive.

Our second table will be named 'TabDesignations' containing fields of 'DesgID', 'Designation', and 'TA'. Evidently, the field of 'DesgID' will form the Primary Key for this table. The third table with the name of 'TabLocations' and containing fields of 'LocationID', 'Location' and 'RateOfHRA' will have 'LocationID' as its Primary Key. Our fourth table will be used for storing the percentage rate of DA for different months of salary and it will be named as 'TabDARates'.

For writing a field name you can use up to 64 characters including letters, numbers and blank spaces. However it is advisable to avoid blanks as it entails irritating and unnecessary use of additional identifiers while using names in SQL and programming components (e.g. "Emp ID" will be written as [Emp ID] or Emp_ID, whereas it could have been easily written as EmpID in both database as well as programming). A name cannot start with a blank. You can include punctuation characters except a period (.), an exclamation mark (!), or brackets ([]).

You cannot use a field name twice in the same table. It is also advisagle to avoid use of standard words that define Base functions and properties. Same rules apply for naming objects like Tables, Queries and Forms; however, no two objects of the same type can have same name.

It will have fields of 'MonthID', 'SalMonth' and 'RateOfDA' with the field of 'MonthID' forming the Primary Key. Now our last table will be used for storing the salary information that will be generated for different employees in different months. This table, named as 'TabMonthlySalary' will contain fields of 'SalaryID', 'MonthID', 'EmpID', 'Basic', and 'DedForPF' (Deduction for Provident Fund). The field of 'SalaryID' will be used for making each salary record as unique data and hence it will form the Primary Key of the table. The 'MonthID' (forming relationship with 'TabDARates') and 'EmpID' (forming relationship with 'TabEmpDetails') will be the two foreign keys of this table. In this table, the field of 'Basic' will be entered by the Pay-clerk for every instance of salary generation for an employee, depending upon the level of employee and his or her attendance record for the month. The Pay-clerk will also record the amount of PF deduction based on the choice given by individual employee, subject to prevalent rules. You may notice that out five tables, only the 'TabMonthlySal' table will grow every month with the addition of as many records as the number of employees drawing salary for the given month.

By now, you may have noticed that we have not given any fields in any of the table for storing attributes of 'DA', 'HRA', 'TA', 'GrossSalary' and 'NetSalary'. These fields are computed fields; and as such the Pay-clerk is not required to enter any data for these fields. Then why store them unnecessarily. Rather we will generate these fields in a Query, which is another Calc object designed to extract data form one or more tables. Queries also allow computational facilities and present the outcome in a tabulated manner, as we shall see in a short while.

LibreOffice Base

LibreOffice Base is a database management system allows the creation and management of databases, that provide informations to the end users. It can be used to create small embedded databases that are stored with the document files and for more demanding tasks. It can also be used as a front-end tool. LibreOffice Base database file contains queries, reports, and forms for the database as well as a link to the database where the records are stored. Data is stored in tables.

CREATING DATABASE

Step 1: To start the LibreOffice Base, click on the Application Menu and point to Office; then click LibreOffice Base. (Application → Office → LibreOffice Base). We will be taken to the 'Database Wizard' Screen as shown in Figure-6.3. This Database Wizard helps you to create a database file or open an existing database file.

Step 2: **Select Database -** Select **Create a new database** option and click on **Finish** button. Then we get **Save** dialogue box

Step 3: Type appropriate file name (eg.“*PayRoll Application*”). The default extension of LibreOffice Base “*.odb*” will be automatically added by the application. select location and click on **Save.**

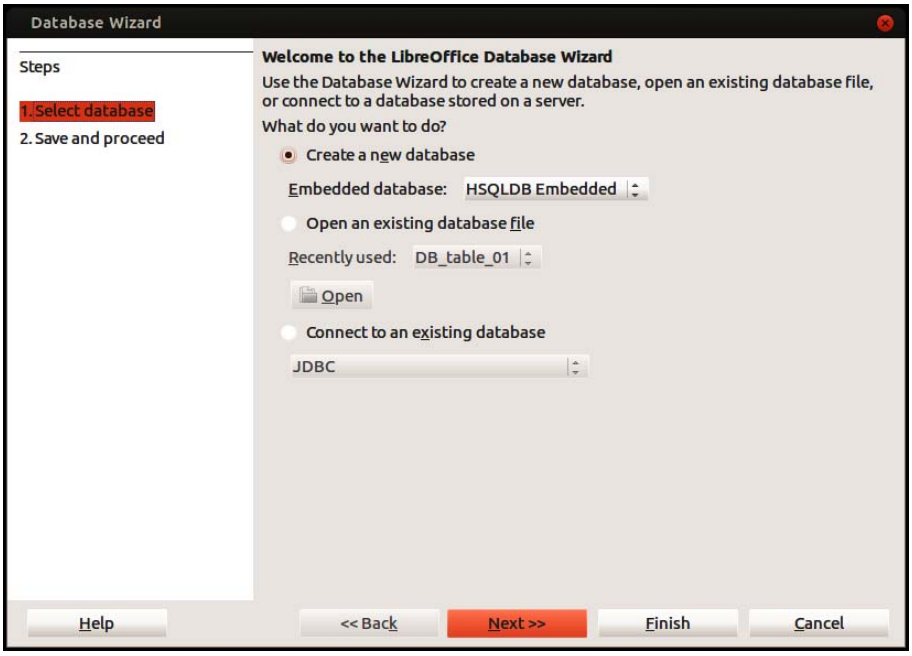


Figure 6.3 Database wizard window

Opening an Existing Database

Step 1 : Open LibreOffice Base, from the Database wizard select **Open an existing database file** option and click on **open** button.

Step 2 : Choose the file from the destination and click **Open.**

Creating Database Table in LibreOffice Base

To create a table, select **Tables** from the **Database** pane. (Figure 6.4) Which shows the options to create the table such as **Create table in Design View..., Use Wizard to Create Table...** or **Create View...**

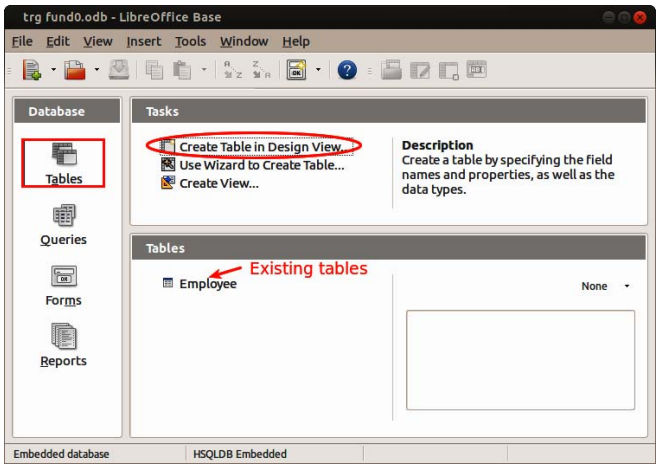


Figure 6.4 Table Creation Options

Creating a New Table With the Design View

1. Click **Tables** from Database pane.

2. Click **Create Table in Design View** in Tasks area (figure 6.4)

The Design View of the new table will appear in the working area of the window (figure 6.5). In Design View, we can see three columns viz, Field Name, Field Type and Description.

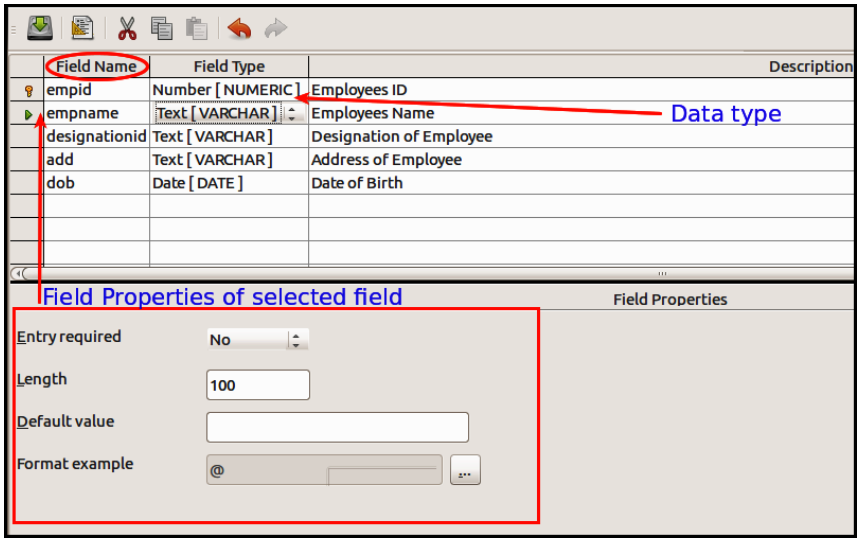


Figure 6.5 Table Creation in Design View

In the Design view, we can now create required fields for our table.

- Click the **Field Name** cell and enter a field name for each data field. *Eg. EmpId.* Enter new fields from top to bottom
- Right click on the **Field Name** required to set as unique identifier for the table, select **Primary Key** option from the pop-up menu.
- In the **Field Type,** we can select a appropriate field type from the combo box for the each fields.
- In **Description** field, we can enter a brief description for each attributes. It is optional.
- Besides the Field type, there are several other properties that can be set independently for each field. Appearing as 'Field Properties' at bottom part of the design view, these parameters of a field will vary with the choice of data type.
- **Save** the table by providing table name and again **Save** the Database by clicking Menu or Tool

Relationships

Relationship are used for connecting tables in database to get the advantage of data redundancy. Having completed the designs of all data tables, we will now move on to the task of establishing relationships between different tables. Click on the ***Tools*** Menu and then ***Relationships...*** menu as shown in Figure - 6.6. In the working area, a ***Add Tables*** dialogue box will appear. In the **Add Tables** dialogue box, select a table and click **Add** button to add it in the relationship window as shown in Figure - 6.7. Add two tables in this manner. After that Click the **Close** button of the **Show Table** dialogue box.

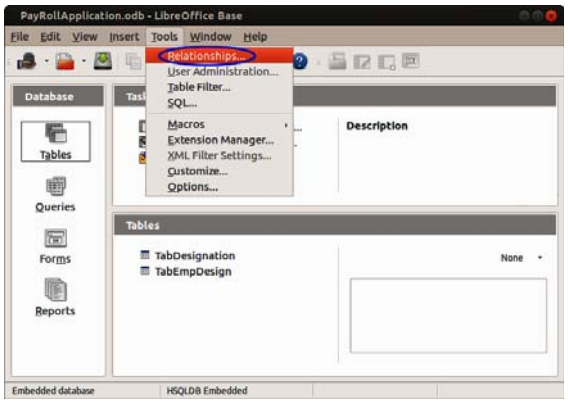


Figure 6.6 Relationship

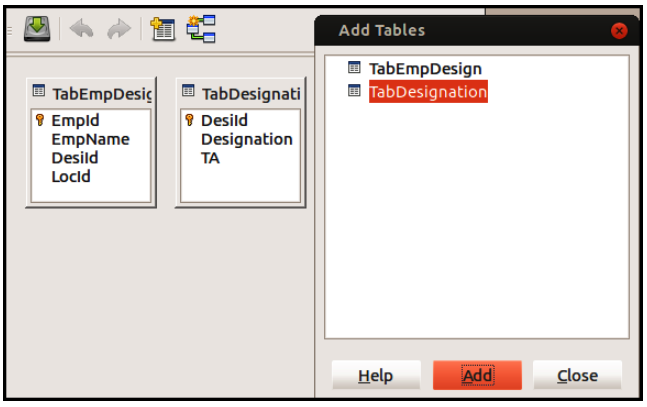


Figure 6.7 Adding Tables to set relation

In the working area we will see two table objects, each detailing the fields within them. We can reposition these table objects any where within the relationship window. Now to create a relationship between two tables, position the mouse pointer over desired field in table object, hold down the left mouse button, drag the pointer right to targeted field of the next table object and then release the mouse button. This can also be done **Insert → New Relation...** menu. A Relationships dialogue box will appear here you should specify the tables and data fields to be related as in Figure 6 .8. We will notice that the dialogue box shows the relationship as in Figure 6.9.

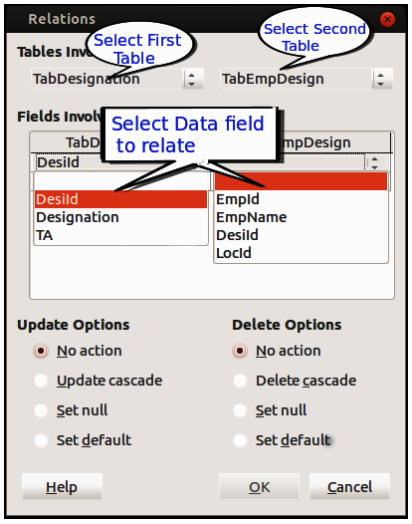


Figure 6.8 Settin Relationship

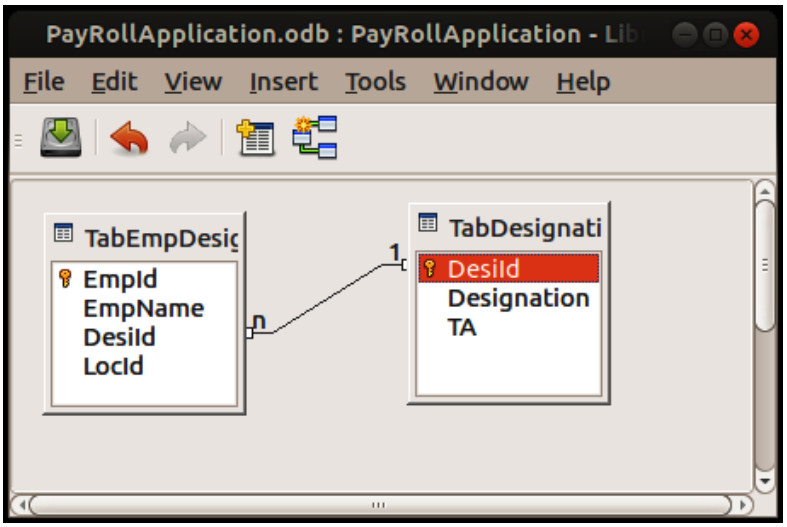


Figure 6.9 GUI for Setting Relationship

CREATION OF FORMS IN LIBREOFFICE BASE

In order to use databases, the user must be provided with an interface that allows to input, view and edit data based on certain criteria. There is a tool for this i.e, **Forms**.

Forms in Data Base Management System

In a database context, a form is a window or screen that contains numerous fields, or spaces to enter data. Each field holds a field label so that any user who views the form gets an idea of its contents. A form is more user friendly than generating queries to create tables and insert data into fields.

Prepare the form in LibreOffice Base

The following are the methods to create forms in LibreOffice Base.

♦ Create Form in Design View

♦ Use Wizard to create form

Use Wizard to Create Form.... is the easy way to create forms

Consider Database table *EmpDetails* which has the fields *EmpID*, *EmpName*, *EmpDesgn* and *EmpOffice* (Figure:6.10)

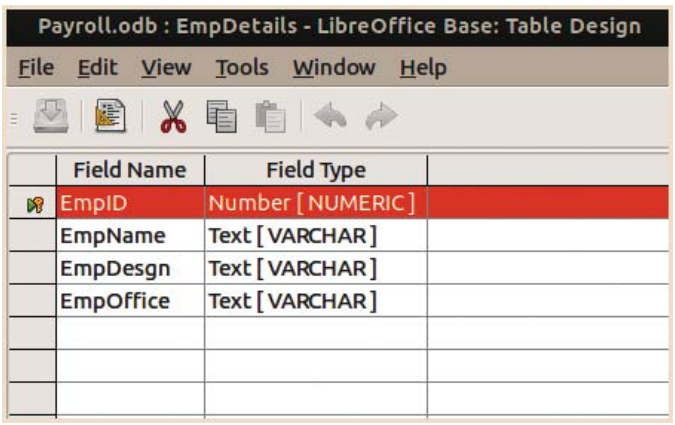


Figure 6.10 Tables structure of EmpDetails

Step 1 : Select **Forms** Option from **Database** Pane

Step 2 : Click on **Use Wizard to create Form...** in the Task Area. **T**he following Form Wizard window appears (Figure :6.11)

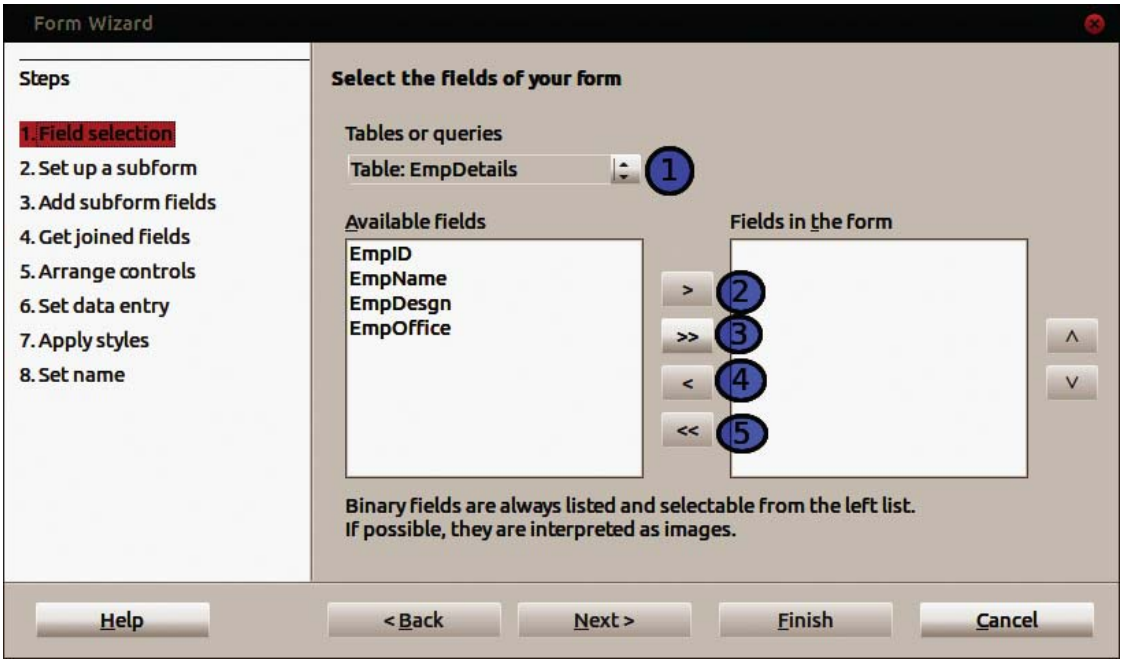


Figure 6.11 Form Wizard

Step 3 : Select the Table or Query (Figure 6.11)

Table of queries, from which we can select which table should be selected to prepare the form. Here we select the table *EmpDetails.*

The fields included in the selected table (here it is *EmpDetails*) are listed in **Available fields** list. Select the required field one by one by select (EmpID, EmpName,..) on it, and click on > to add to the form(refer 2 in the above figure). All fields can be added choosing >> (refer 3). If one added field to be removed by selecting the added field and clicking on <(refer 4). All added fields can be removed using << (refer 5).

After selecting the required field proceed by clicking **Next**.

The steps '**Add subform fields**' and '**Get joined fields**' are connected with Sub form, which will be explained later.

Step 4 : **Arrange controls**: The arrangement of Form field and appearance can be selected.

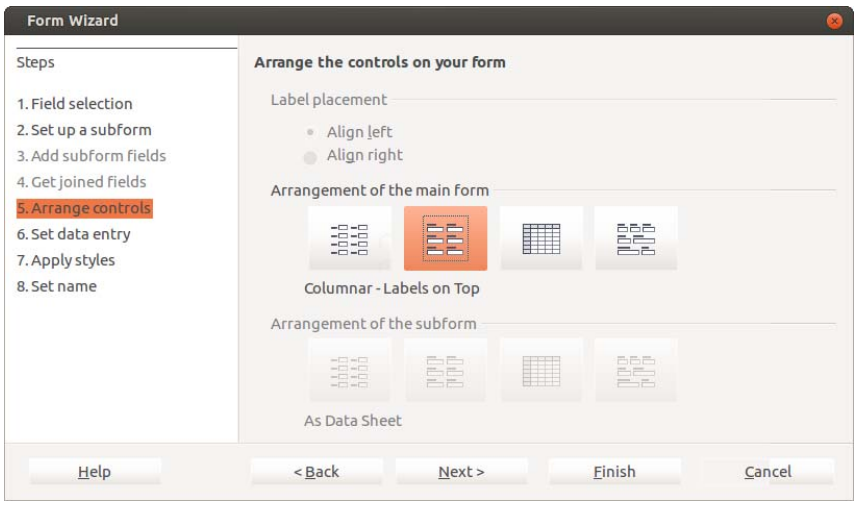


Figure 6.12

Step 5 : **Select the data entry mode:** Here determines the data entry characteristics. If the Form is used for entering new data only and the entered data not displayed, the first option may be selected.

Step 6 : **Applying style of your form:** The background color of our form can be selected from this option. Field boarder can also be selected.

Step 7 : **Set the name of the Form:** We can give the name of the form we are creating. The name must be unique and must have a relation with the data to be stored. Here we give **EmployeeDetails** as name.

There is also an option to continue to Work with form or to Edit the form by selecting the option **Modify the form**.

Then click **Finish** button.

The form will be created as shown in figure 6.13.

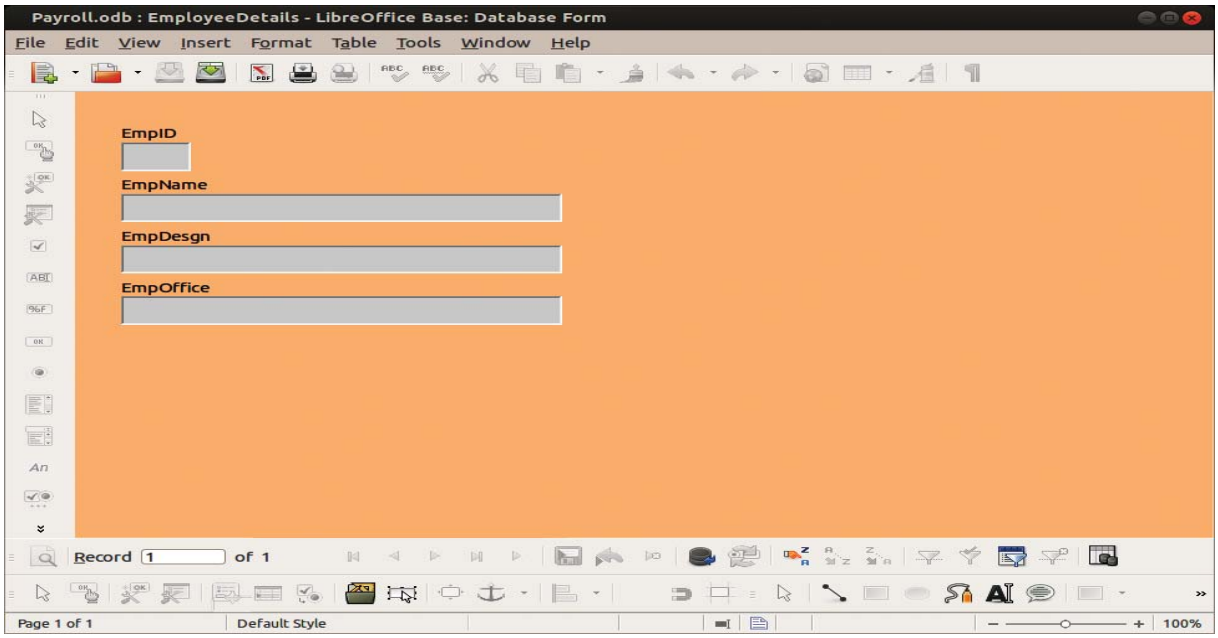


Figure 6.13

In this form we can enter the details. For example enter EmpID as 101, press Tab Key (or Enter Key) in the keyboard to move to the next field EmpName. Enter EmpName: Avinash.S, press tab, Enter EmpDegn: Higher Secondary School Teacher press Tab key to enter EmpOffice: GHSS Ranni, again press tab to save the data and get a fresh record. Like wise we can enter the entire data. Moving between the Data, we can set the record number in the search window given at left bottom corner, or using the navigation keys given at the bottom.

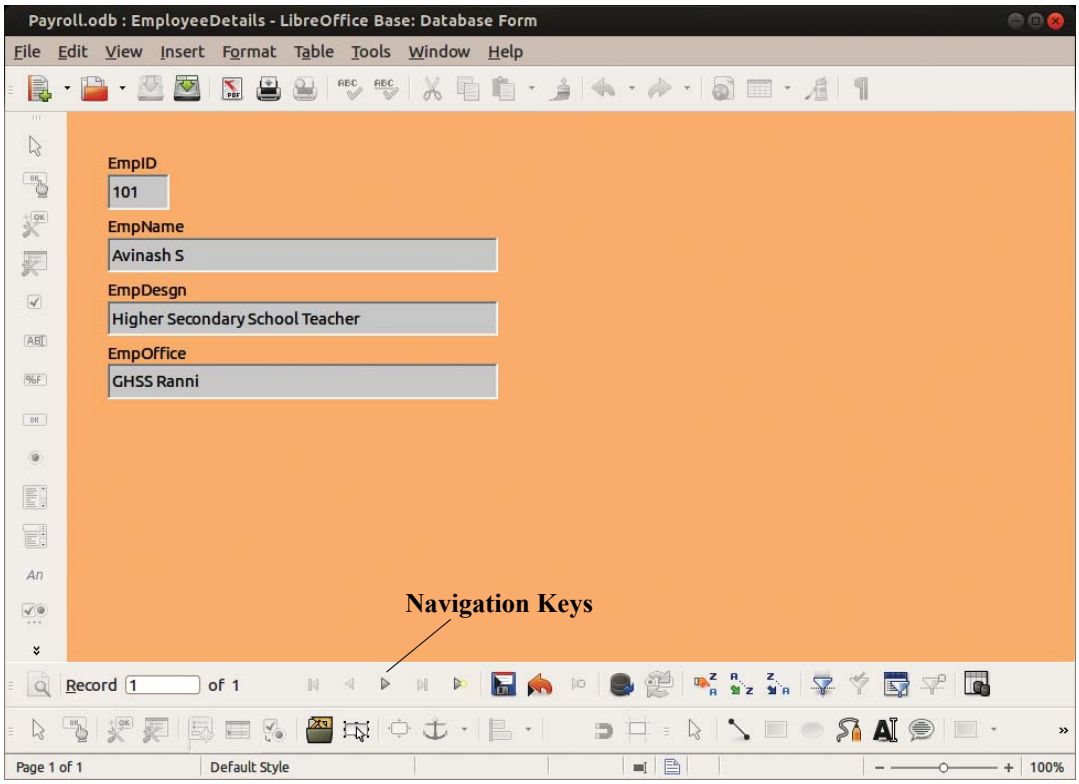


Figure 6.14 Form for Data Entry

Editing the data already entered in the Form

We can edit the data already entered in the record.

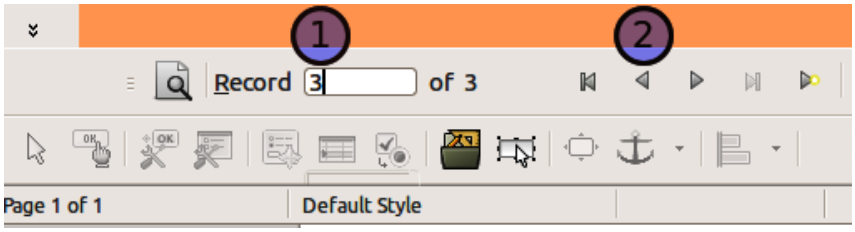


Figure 6.15

The current record number is seen in (1)in figure 6.15. We can go to the record by entering the record number and **Enter** key press. If the data of the particular field is to be changed, select it, edit it and save by pressing **Tab** key or **Enter**

Working with subforms

Subforms are used when two or more tables to be combined to form a Single LibreOffice Base Form.

For this we have to create at least two tables. Considering the above example, create one more table, ***SalaryDetails***, which has the fields, *EmpID, BasicPay* and *HRA.*

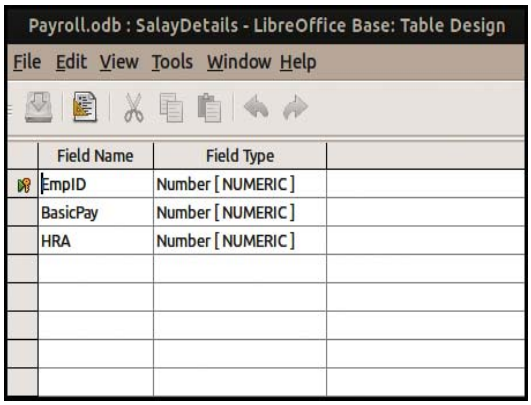


Figure 6.16

Make a relationship between the two tables, *EmpDetails* and *Salary,* using the option **Tools → Relationship**

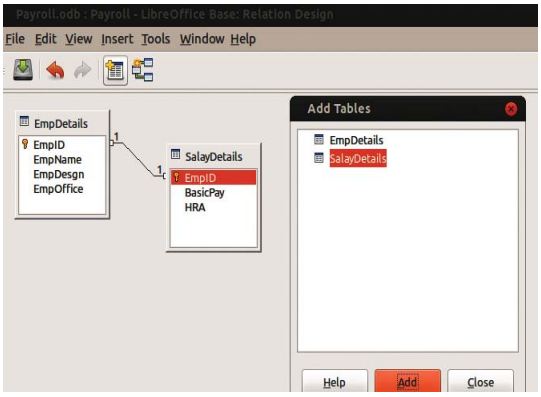


Figure 6.17 Table Relationship

Form creation with Subform

To create the form, take the option Use Wizard to create Form. The steps are given below.

Step 1: In the **Select the fields of your form** select the Table *EmpDetails* from the Tables or queries. Select required fields, as detailed above, in Form Creation.

Step 2 **: Set up a subform- Decide if you want to set up a subform.** Select the option **Add Subform** (click on the check box given - figure 6.18). Select the **Subform based on existing relation,** (click on the radio button), and the table *SalaryDetails* listed in the box(click on the *SalaryDetails*).

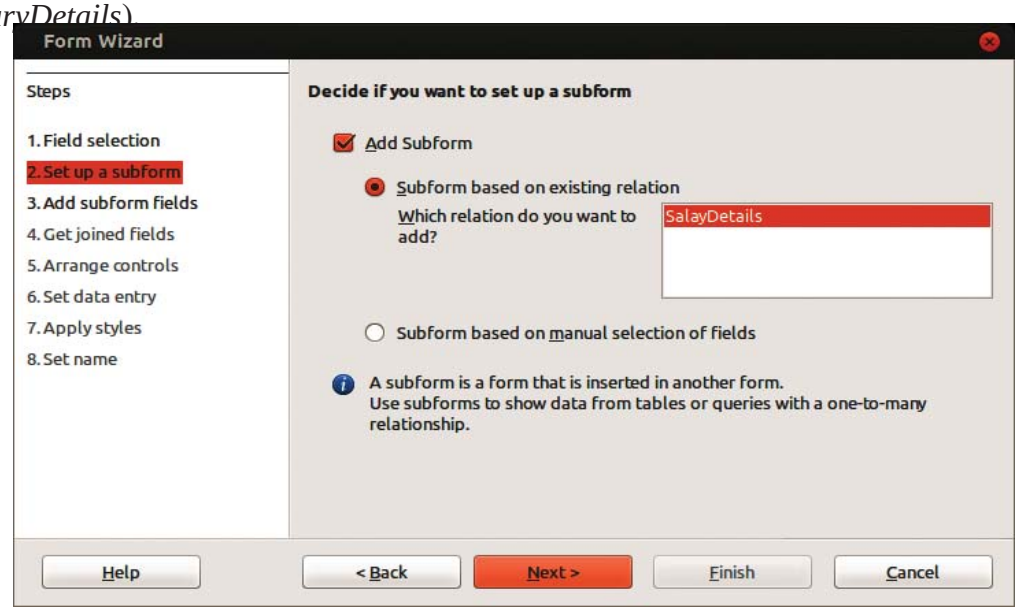


Figure 6.18 Form wizard - Subform creation

Step 3 **: Add Subform Fields** (select the fields of your subform-figure 6.19). Select the fields *BasicPay* and *HRA* to add(since the field *EmpID* has already added in the main form, it must not be added here).

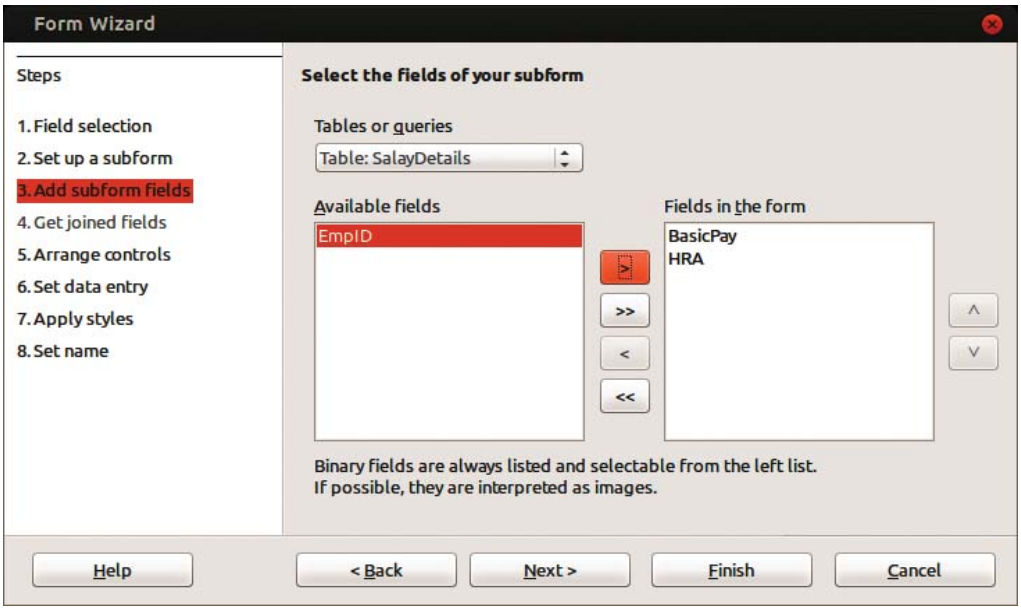


Figure 6.19 Adding Subform fields

Step 4 **: Arrange controls:** Here we have to select the Arrangement (how the fields to be arranged) of the main form as well as that of subform.

Step 6 **: Select the data entry mode** The form is to display all the data, may be selected as default.

Step 7 : **Apply style of your form:** The background design may be selected here.

Step 8 : **Set the name of the form:** Enter the name of the form which is decided to given. Since it is working with subform, the second option, **Modify the form**, have to selected. Then click on **Finish.**

It opens the form edit window (figure 6.20).

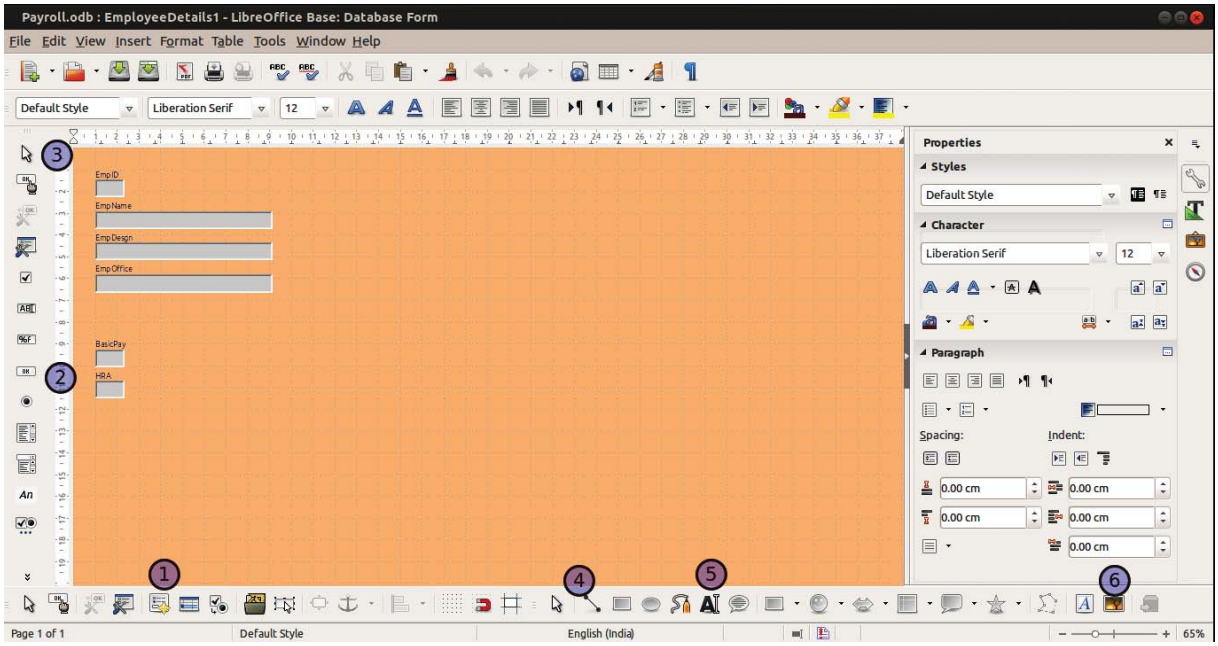


Figure 6.20 Form Edit window

In this window, the Form Navigation (ref (1) in Figure 6.20), option is available, clicking on that, Form Navigator pop-up window opens (figure 6.21).

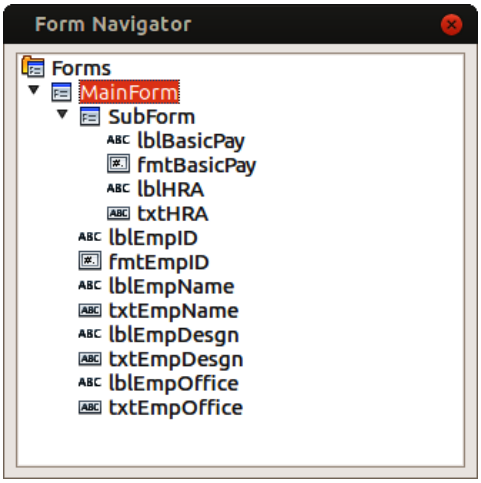


Figure 6.21 Form Navigator

Here the fields of **MainForm** and **SubForm** is listed. The item started with ***lbl*** indicates Label, and which started with **txt** indicates the field to Enter text, which start with **fmt** indicates the field to enter numeric values. Right click on the *fmtEmpID* (which is the part of **MainForm**) and select the **Properties** in the pop-up menu (figure 6.22).

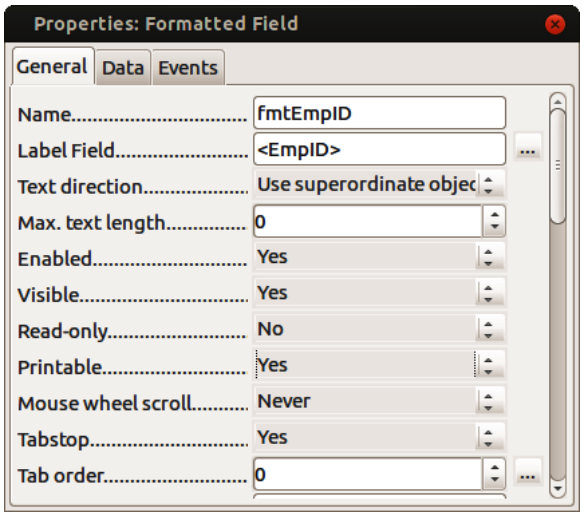


Figure 6.22 Properties window

The Tab order (in numbers) may be set according the preference. Tab order determines which field to be selected firstly, secondly, etc.,. while entering data in to the form.

In the given example, set Tab order 1 for *EmpID*, TabOrder 2 for *EmpName*, 3 for *EmpDesgn*, 4 for *EmpOffice*, 5 for *BasicPay* and 6 for *HRA*. Thus at first enter *EmpID*, then press **Tab** on the keyboard, it moves to the field *EmpName*, the data can be entered in order. After entering the field *EmpOffice*, it moves to the sub form field *BasicPay*. Finishing the data entry in to the last field it saves and moves to the next record.

Some of the Form Editing options

After creating a form, we can edit it. For this right click on the form name listed in the LibreOffice Base main window. It gives the option to edit, and opens the window which enable the editing options. The Arrow help to select a field or object.

Push Button *(refer 2 in the figure 6.20)*

This is an important feature of LibreOffice Base, which may be set, and assign an action, viz moving to the next record, moving to the first record, save record etc. Select the **Push Botton** tool, Click and drag in the form, where it has to be placed.. Double click on that, opens the **Properties:Push Button** window. In the General Tab, type the Name to be displayed in the **Push Button** to be entered in the Field Label. Select the **Action,** which opens a pop-up window. Select the **Next Record** option (see figure 6.23) .

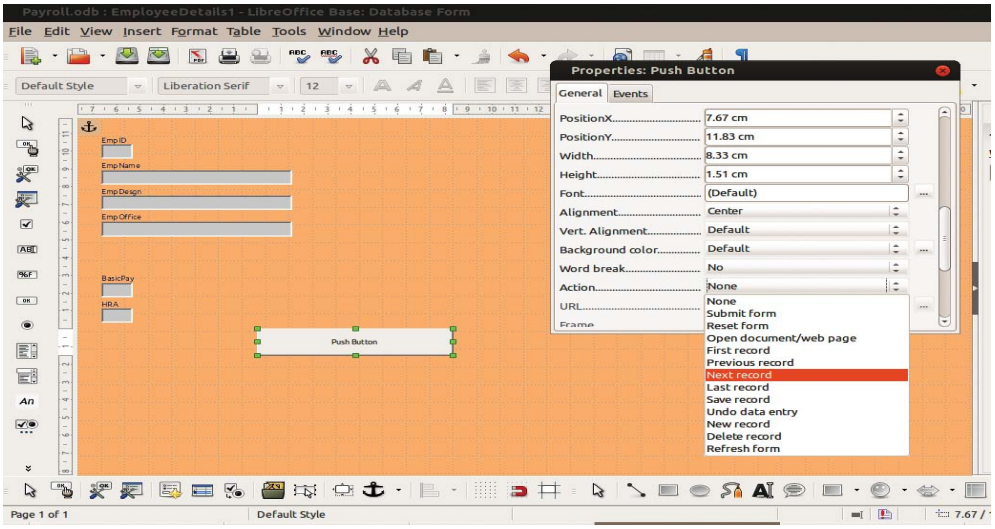


Figure 6.23 Properties - Push Button

Add Text in the Form

The Text Box can be used to add Heading or any details as text in the Form. Click on the Text Box Icon (figure 6.24) and click on the working area. The cursor will be positioned on the Top left of the working area. Then the matter which is to be included. The text entered can be formatted (Font Name, Font Size , Font Color etc. Can be changed).

Inserting Image to the form

*Image can be inserted in the form selecting **Image** insert icon (figure6.24), which opens the window to select the picture from the memory location, where the picture is kept. The size and position can be changed in the form.*

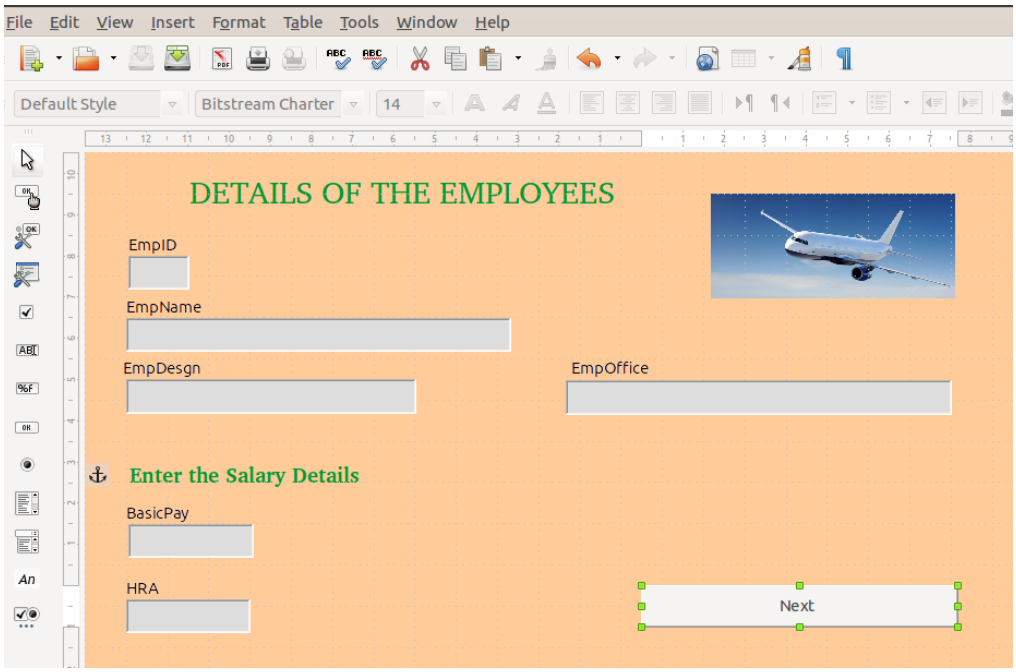


Figure 6.24 Form Edit Final

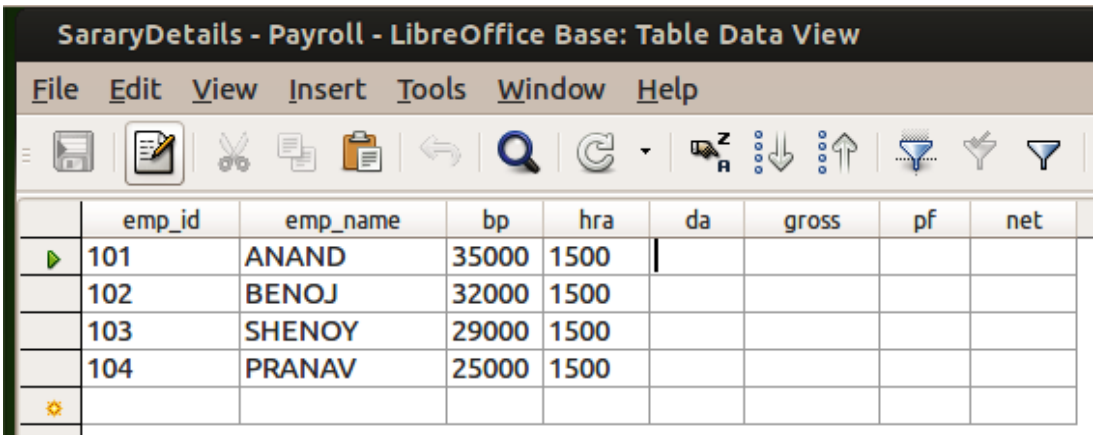
CREATION OF QUERY IN LibreOffice Base

A **query** is a request for data or information from a database and query languages are used to make queries in database. Queries are also used to manipulate the database content. Structured Query Language(SQL) is the most well-known and widely-used query language. LibreOffice Base also uses SQL commands for querying its database. Query operations can be done in different ways using LibreOffice Base.

To Execute Query statement directly

In LibreOffice Base, this method is used to display, update and delete database table content. Consider the following steps.

- Open database file using LibreOffice Base. Database table *SalaryDetails* of database *Payroll* is displayed in the given figure 6.25.



emp_id	emp_name	bp	hra	da	gross	pf	net
101	ANAND	35000	1500				
102	BENOJ	32000	1500				
103	SHENOY	29000	1500				
104	PRANAV	25000	1500				

Figure 6.25 The partially filled database table *SalaryDetails.*

- Choose **SQL** option from **Tools** Menu. A new window appears (Figure 6.26).

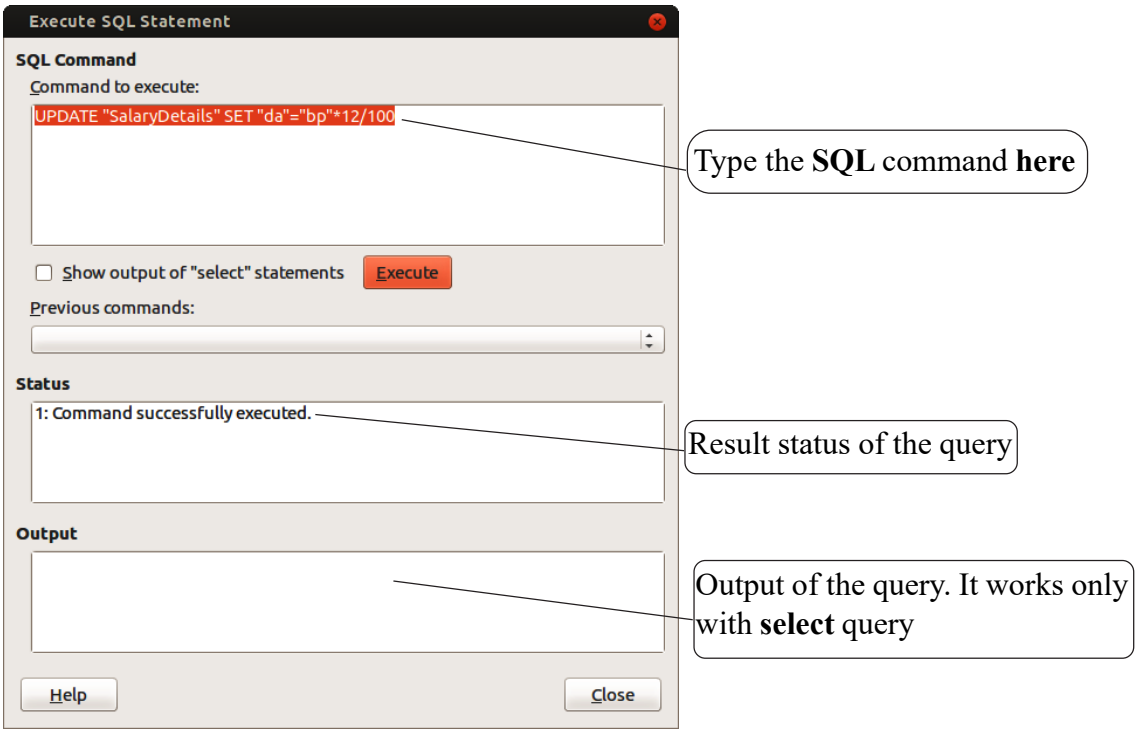
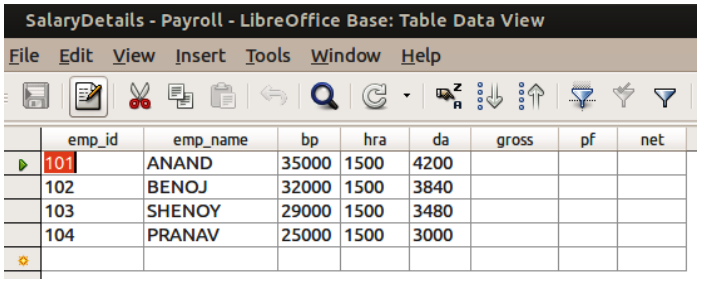


Figure 6.26 Windows used to execute Query directly

- Type the Query in the area **Command to execute.** Press **Execute** button to run the Query. If the query is correct the statement ***command successfully executed*** will be displayed in the **Status** box. If the given query is a **select** query, the result also displayed in **Output** box. The content of the resultant table also changed according to the command given. See the change in table given below (Figure 6.27). Here 'da' is automatically generated with the help of SQL command.



emp_id	emp_name	bp	hra	da	gross	pf	net
101	ANAND	35000	1500	4200			
102	BENOJ	32000	1500	3840			
103	SHENOY	29000	1500	3480			
104	PRANAV	25000	1500	3000			

Figure 6.27 Table *SalaryDetails* after the execution of SQL command

- Similarly, all the columns in database table can be filled by executing SQL commands directly.

To Create query using wizard

Query wizard is also used to create and store queries. It helps to create query without knowing the exact field names. The following steps can be used to create query with the help of wizard.

- Open database file using LibreOffice Base. Select the option **Query** from left panel. The following options (Tasks) will be displayed in the right side.

  1. Create Query in Design View
  2. Use Wizard to Create Query
  3. Create Query in SQL view

- Select the option **Use Wizard to Create Query**. Query Wizard window appears (Figure 6.28).

Figure 6.28 Query wizard window

Ø ***Field Selection :*** The first step in Query Wizard window is **Field selection**. All the tables included in the database can be seen in the **Table** list. In the figure (Figure 6.28) the table '*SalaryDetails*' is selected. All the fields of the selected table can be seen in the **Available fields** list. The user can select the field(s) needed from the list using the tools arranged (Like >, >>, < and <<) right to **Available fields** list. The selected fields can be seen in **Fields in the Query** list. The order of the selected fields can change using the tools (∧ and ∨) right to **Fields in the Query** list. Then, press **Next** button or **Finish** button.

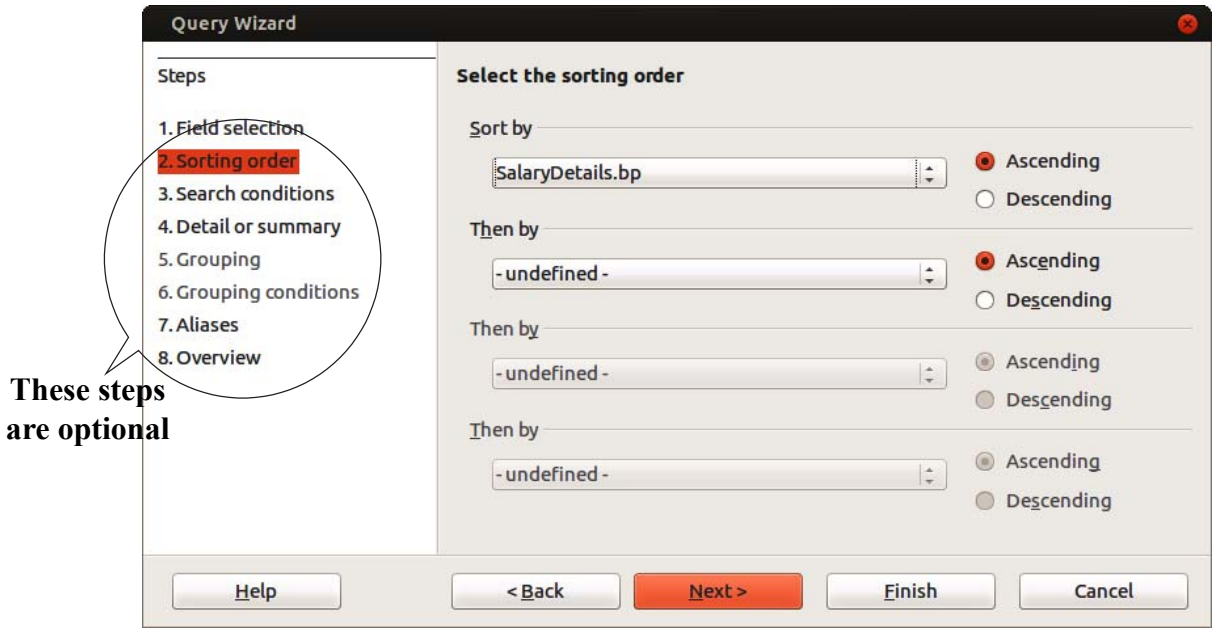


Figure 6.29 Query Wizard with Sorting order

- ***Sorting order***: In this step, the field name to sort the query result can be selected. In the given example (Figure 6.29), the basic pay (*bp*) is selected as **sorted by** field. After selecting the order (Ascending / Descending), press **Next** button. [Skip this step, if no sorting is needed]

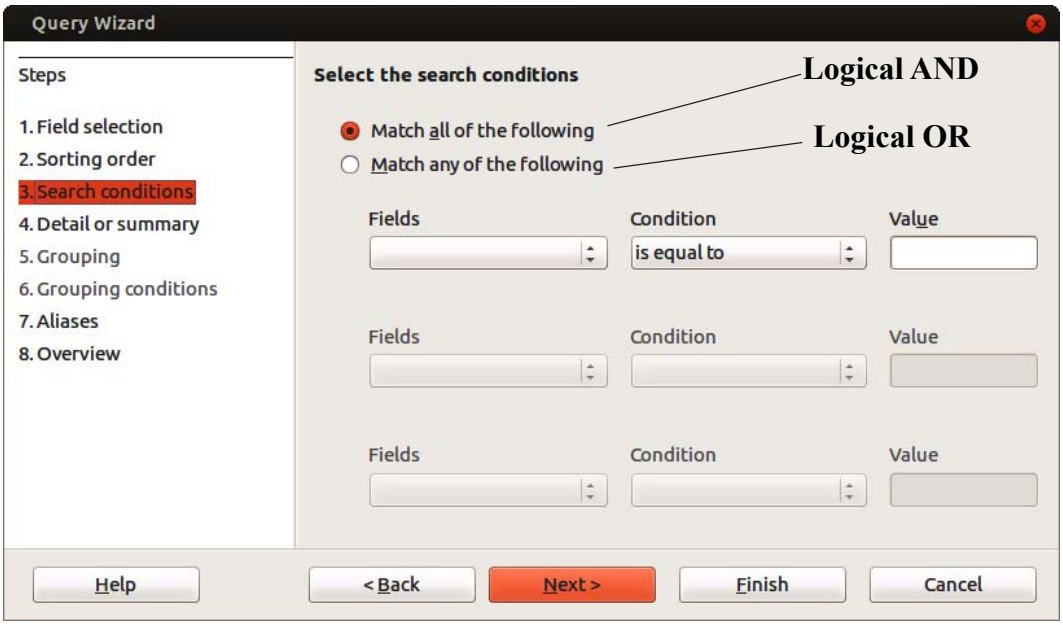


Figure 6.30 Query Wizard Window to state conditions to filter the data records

- ***Search Conditions:*** This step specifies the search conditions to filter the query. The conditions can be stated using logical AND or logical OR (Figure 6.30). After stating conditions, press **Next** button. [This can skip, if no filtering is needed]

- ***Details of Summary:*** This page specifies whether to display all records of the query, or only the results of aggregate functions. This page is only displayed when there are numerical fields in the query that allow the use of aggregate functions.

- ***Grouping***: Specifies whether to group the query. The data source must support the SQL statement 'Order by clauses' to enable this page of the Wizard.

- ***Grouping Conditions***: Specifies the conditions to group the query. The data source must support the SQL statement 'Order by clauses' to enable this page of the Wizard.

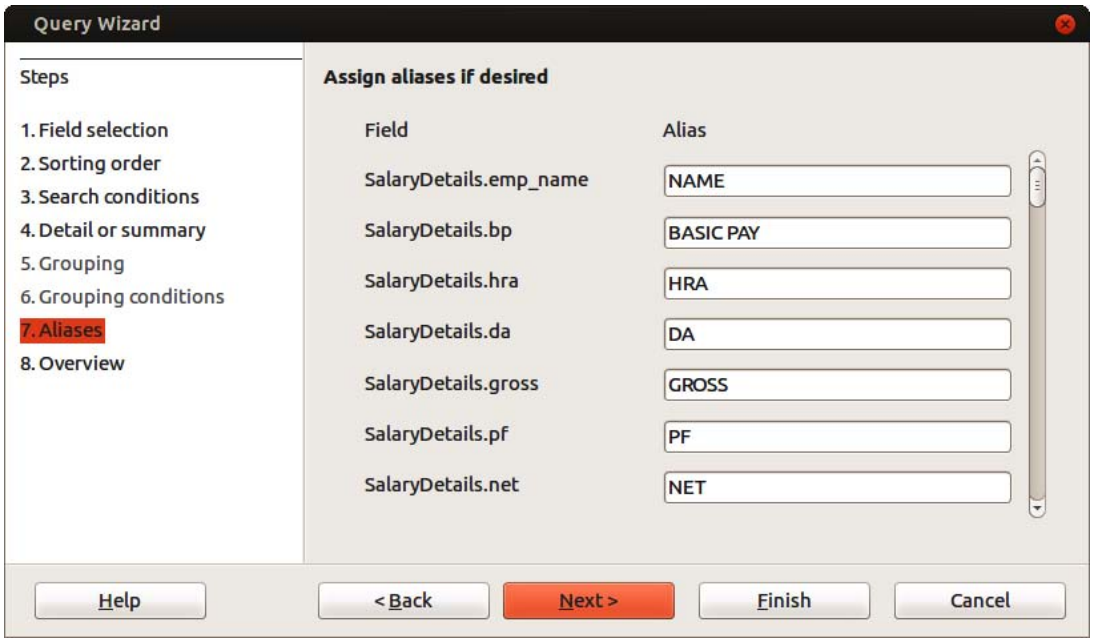
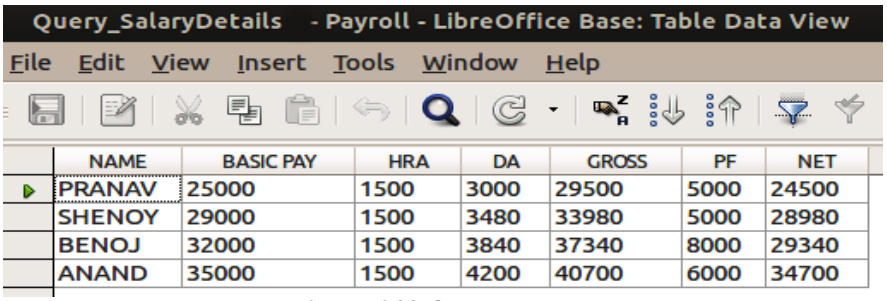


Figure 6.31 Query Wizard - After assigning aliases to field names

- ***Aliases*** : This page helps to assign aliases to field names (Figure 6.31). Aliases are optional, and can provide more user-friendly names, which are displayed in place of field names. For example, an alias can be used when fields from different tables have the same name.

- ***Overview***: This wizard page gives an overview of the query made. It helps to enter a name of the query, and specify whether to display or to modify the query after the Wizard is finished.



NAME	BASIC PAY	HRA	DA	GROSS	PF	NET
PRANAV	25000	1500	3000	29500	5000	24500
SHENOY	29000	1500	3480	33980	5000	28980
BENOJ	32000	1500	3840	37340	8000	29340
ANAND	35000	1500	4200	40700	6000	34700

Figure 6.32 Query Result

- Press **Finish** button after the completion of Query wizard entry. The query will be saved. The user can run this query at any time. See the query result in Figure 6.32. Note the replacement of field names with aliases.

How to run Saved Query?

Select **Queries** option from the left panel of the LibreOffice Base window (See figure 6.33). The saved query can be seen in the right side. Double click on the query name to run the query.

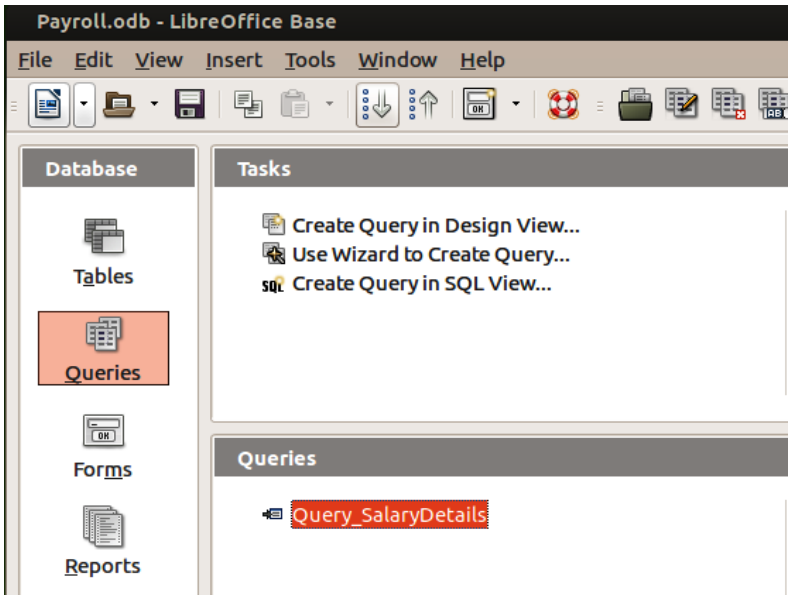


Figure 6.33 Base window after selecting *Queries* option

To create query in design view

Use the option **Create Query in Design View** from Base window (See Figure 6.33) to create query in design view. **Query Design** window appears (Figure 6.34).

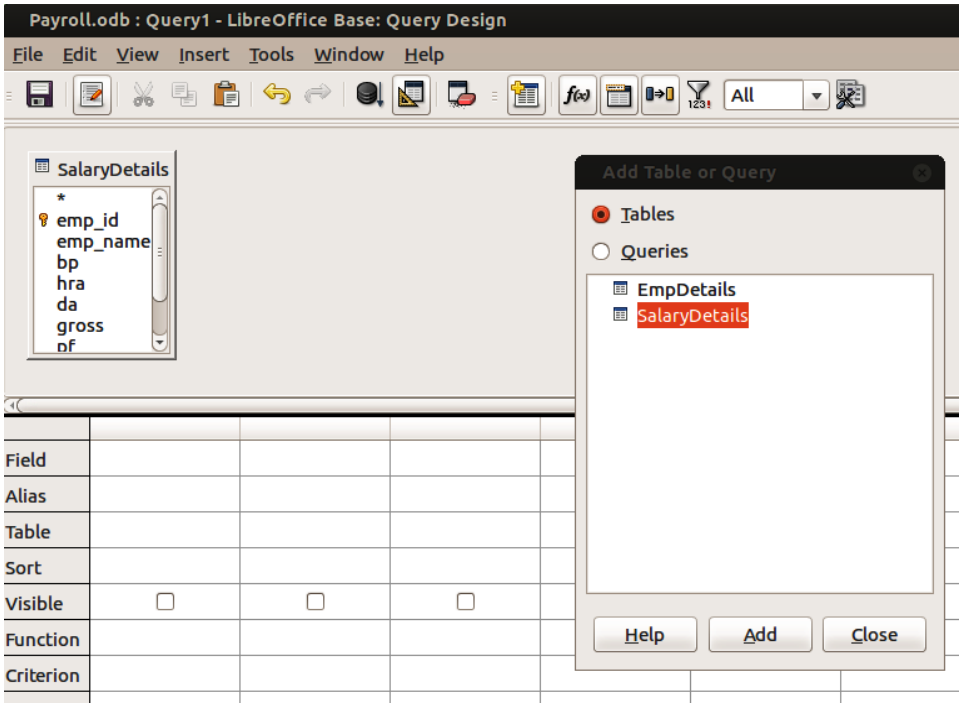
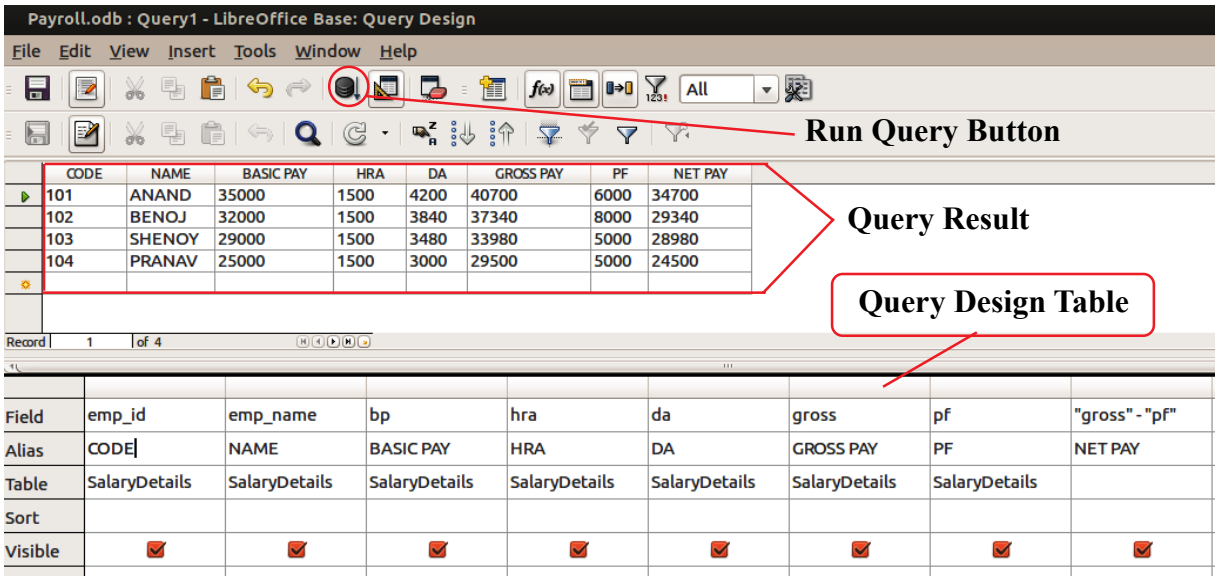


Figure 6.34 Query Design window

Use **Add Table or Query** dialogue to include table(s) to query design. See added table *SalaryDetails* in Figure 6.34. Then, include the fields and formula in the top row as shown in Figure 6.35. Give aliases in the second row, if needed. Press, **Run Query** button to execute the query. The result will be displayed on the top of query design table (Figure 6.35).



CODE	NAME	BASIC PAY	HRA	DA	GROSS PAY	PF	NET PAY
101	ANAND	35000	1500	4200	40700	6000	34700
102	BENOJ	32000	1500	3840	37340	8000	29340
103	SHENOY	29000	1500	3480	33980	5000	28980
104	PRANAV	25000	1500	3000	29500	5000	24500

Field	emp_id	emp_name	bp	hra	da	gross	pf	"gross" - "pf"
Alias	CODE	NAME	BASIC PAY	HRA	DA	GROSS PAY	PF	NET PAY
Table	SalaryDetails	SalaryDetails	SalaryDetails	SalaryDetails	SalaryDetails	SalaryDetails	SalaryDetails	
Sort								
Visible	☑	☑	☑	☑	☑	☑	☑	☑

Figure 6.35 Query Design window after the execution of Query

CREATION OF REPORTS IN LIBREOFFICE BASE

Information from a database can be generated through the Reports option in LibreOffice Base. The reports can be printed, it allows to design the layouts and formats as per the requirements of user. The reports generated in LibreOffice Base can be edited, printed and exported to PDF format.

Step 1 : Click the icon **Reports** in Database pane.

You may find the following options

- Create Report in Design View...
- Use Wizard to Create Report...

Use wizard option guides the user through a series of dialogue boxes to create report in suitable format. (Figure 6.36)

Step 2 : Click on **Use Wizard to Create Report...** option in Task Area (Figure 6.36)

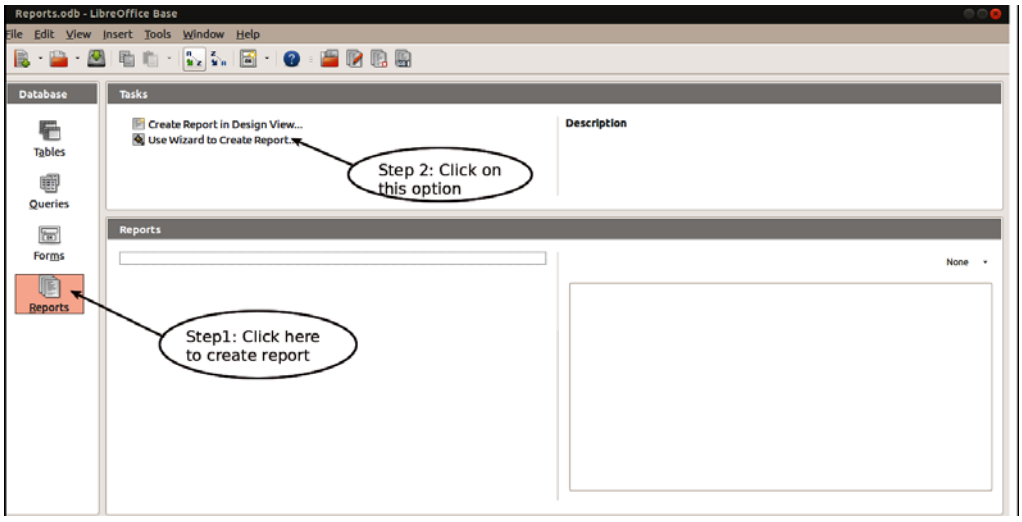


Figure 6.36 Base window after selecting *Reports* option

Step 3 : Select table or query from the drop down option for which reports need to be created (Figure 6.37)

Step 4 : Select the required fields (Figure 6.37)

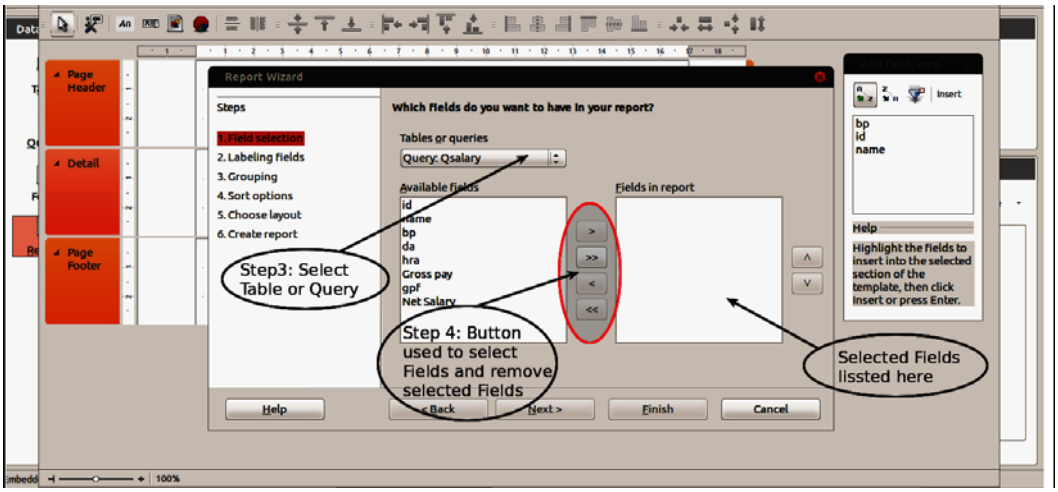


Figure 6.37 Field selection in Report Wizard

- ***Field Selection :*** All the tables and queries included in the database can be seen in the **Table or Query** list. In the figure, the query *QSalary* is selected. All the fields of the selected table can be seen in the **Available fields** list. The user can select the field(s) needed from the list using the tools arranged (Like >, >>, < and <<) right to **Available fields** list. The selected fields can be seen in **Fields in the Report** list. The order of the selected fields can change using the tools (∧ and ∨) right to **Fields in the Report** list. Then, press **Next** button or **Finish** button.

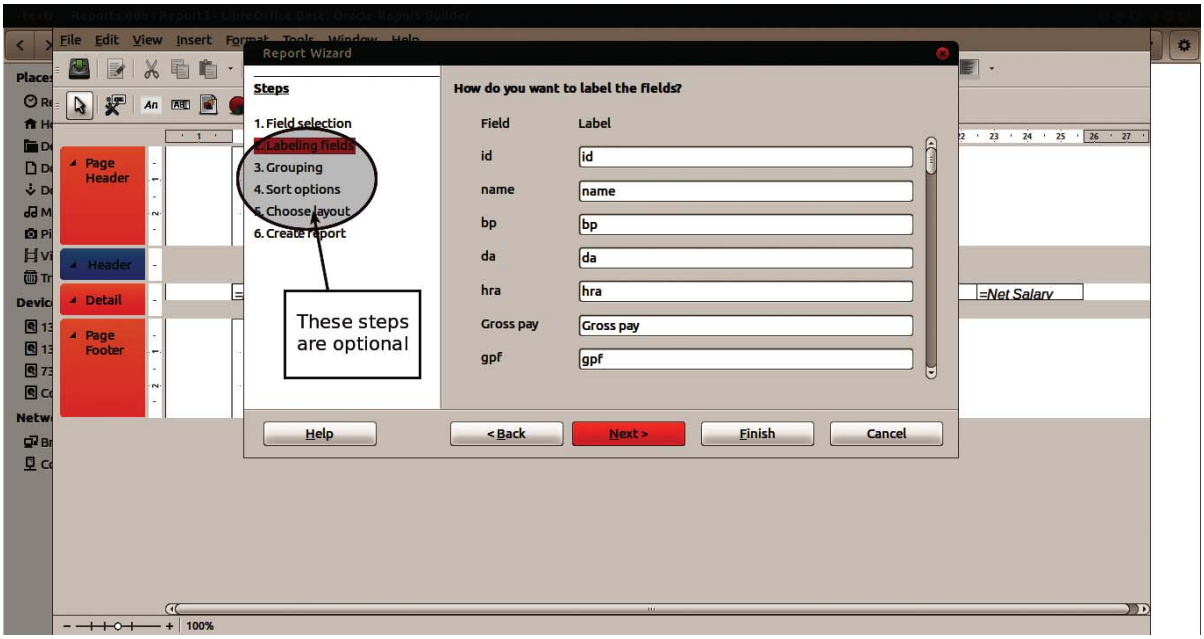


Figure 6.38 Optional steps in Report Wizard

Labeling fields, Grouping, Sort options and Choose layout are optional steps (Figure 6.38).

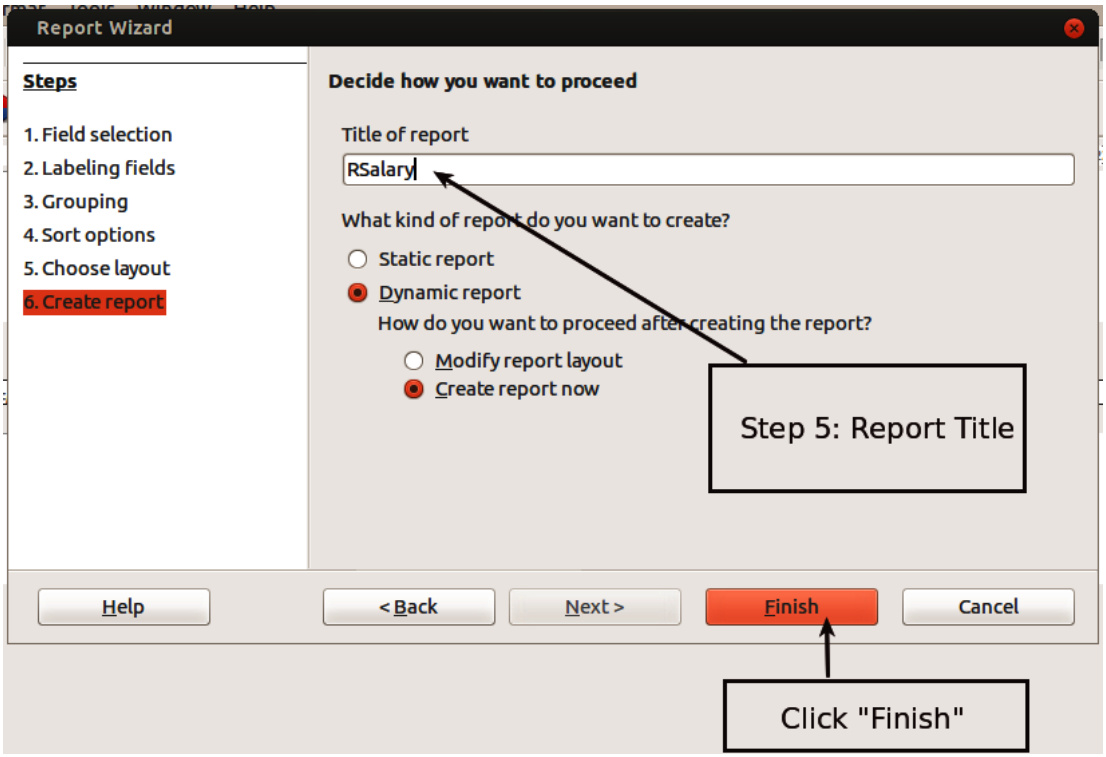


Figure 6.39 Report Wizard (Add Title)

Step 5 : Enter Title for the report in the field **Title of the report** and Click on **Finish** button as shown in Figure 6.39.

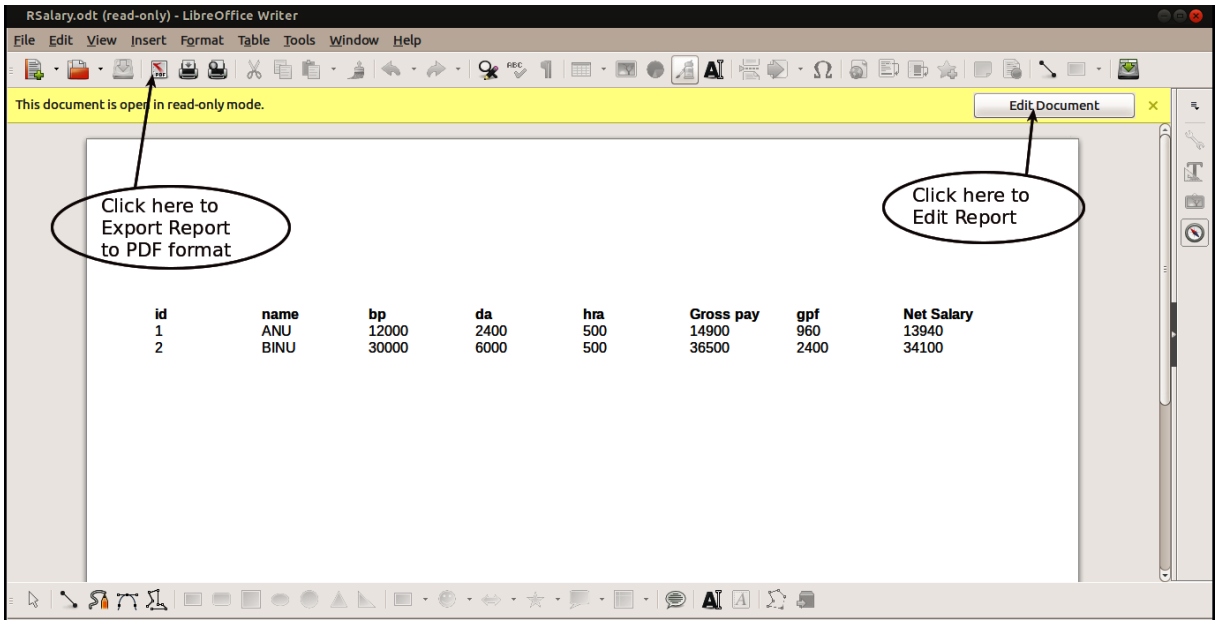


Figure 6.40 Generated Report

Step 6 : The Report generated in the read-only mode as shown in the figure 6.40. The report can be edited as per the user's requirements by clicking on **Edit Document** Option.

EXCERCISE

1. DBMS stands for :

a. Drawing Board Management Software

b. Dividend Based Marking System

c. Data Base Management System

d. Data Base Marketing Software.

2. LibreOffice Base is a :

a. Word processing Software

b. Presentation Software

c. Spread sheet Software

d. Data Base Management Software.

3. The term ‘field’ as applied to database table means :

a. Vertical column of the table

b. Size of the table

c. Horizontal row of the table

d. Name of the table.

4. The term ‘record’ as applied to a database table mean :

a. Vertical column of the table

b. Size of the table

c. Horizontal row of the table

d. Name of the table

5. The common fields used in a relationship between tables are called :

a. Joint fields

b. Main fields

c. Key fields

d. Table fields.

6. The existence of data in a Primary key fields is :

a. Not necessarily required

b. Required but need not be unique

c. Required and must be unique

d. All of above.

7. The existence of data in a secondary key fields is :

a. Not necessarily required

b. Required but need not be unique

c. Required and must be unique

d. All of the above.

8. SQL stands for :

a. Simple Questions Language

b. Simple Que line up

c. Singular Quantity Loading

d. Structured Query Language.

9. The default extension of LibreOffice Base file is :

a. .odb

b. .exl

c. .doc

d. .exe

10. Wizards in LibreOffice Base means :

a. Person who developing programme

b. Tools for simplifying the programme usage

c. Relating between tables

d. Reporting generated by programme.

11. 'Join line' in the context of Base Tables means :

a. Graphical representation of relationship between tables

b. Lines bonding the data within table

c. Line connecting two fields of a table

d. Line connecting two record of a table.

12. In order to retrieve select data meeting a specified criteria from two different tables of LibreOffice Base database, we may make use of

a. Table

b. Query

c. Form

d. Report.

13. To expect a well formatted printable data form Base database, we may use :

a. Table

b. Query

c. Form

d. Report.

Q2. Answer the following Questions

1. What do you understand by DBMS. Give names of two commarly available DBMS softwares?

2. Differentiate between 'Desktop database' and 'Server database'. List the criteria that may help you in selecting appropriate database?

3. With suitable example illustrate the meaning of 'attributes' as applied to database?

4. Why do we seek to split up in formation into different tables rather than confine it to a single table?

5. What do you understand by terms 'key field', 'primary key' and 'secondary key' in a database?

6. List the conventions that you will follow, while naming different fields of table?

7. What are the uses of 'query' object in LibreOffice Base program?

8. What do you understand by 'Form' object in LibreOffice Base and how are they useful?

9. What is the purpose of 'report' object in LibreOffice base program?

10. What do you understand by database? What are the ways in which data is stored and queried in a Base database?

11. What are the advantages of LibreOffice Base over Calc?

12. Describe in brief the function of 'Table', 'Query', 'Form' and 'Report' object of Base program?

Q3. Skill Review

1. You are starting to plan ahead for you job search. You decide to maintain a database of company information in LibreOffice Base.

2. Search for at least eight companies. You wish to apply for include company name, addresss, telephone and fax numbers. If possible, also include the name of contact person in their human resource department.

3. Create job search company safer, accounting database.

4. Open the company information table.

5. Enter at least records for the companies you researched.

6. Adjust column widths, as necessary.

7. Sort the records in ascending order of Name of Company.

8. Preview Table.

9. Format all record to a smaller font size.

10. Change the page layout so that entire table fit on the page.

11. Close company information table.

12. Close Job Search Company, Base database.

Answer

Q.1.

1. c **2.** d **3.** a **4.** c **5.** c **6.** c **7.** a **8.** d **9.** a **10.** b **11.** a

12. b **13.** d