

**Let us learn**

- DBMS concepts.
- Various table operations.
- Query and form creation.
- Report generation.
- Introduction to relational data model

Base is an open source database management system software, and designed to allow users to easily create, access, modify and view database.

**5.1 DBMS concepts**

Data means all kinds of facts, figures and details related to people, places, things or events. Data may be in any form. It may be written, oral, computerised or non-computerised. Data in its original form may not be of much use. Data must be processed in a proper way to generate the useful and meaning information.

Information is the required result obtained from processing of the data. Information is the output generated through processing of raw data. Information is important because it forms the foundation for decision making.

Database is a collection of related data items stored in an organised manner. A Database consist of different objects like table, query, form and report. **Table** is a collection of related data. **Query** is

used to retrieve information from database. **Form** is used to collect the information from user. **Report** is used to represent the data in printed form.

**Database Management System Software (DBMS) :**

A database management system is a software designed to define, manipulate, retrieve and manage data in a database. It provides various functions that allow entry, storage and retrieval of large quantities of information and provide ways to manage that information. It also defines rules to validate and manipulate the data.

**5.2 Introduction to Base**

Base is a Relational Database Management Software(RDBMS). A DBMS that is based on relational data model is called as RDBMS. A data model is the internal structure of database which describes way of storing and retrieving of data. Relational data model is one of the most popular data model because it is very simple to understand and to manipulate. In RDBMS data is stored in the most simple and versatile structure i.e table. Base is collection of related data objects known as Tables, Forms, Queries and Reports.

To start base, Install Libreoffice from its official website and click on start →

Start → All programs → Libreoffice → Libreoffice Base. A database wizard will be started. To create new database user have to select 'Create a new Database and click on 'Next' button. In 'Save and Proceed' step click on 'Finish'. A 'Save

As' window will appear on the screen, select location to save the database → Type name for database in 'File Name' box then click on 'Save' button. By default extension of Base file is '.odb'.

### LibreOffice Base Screen :

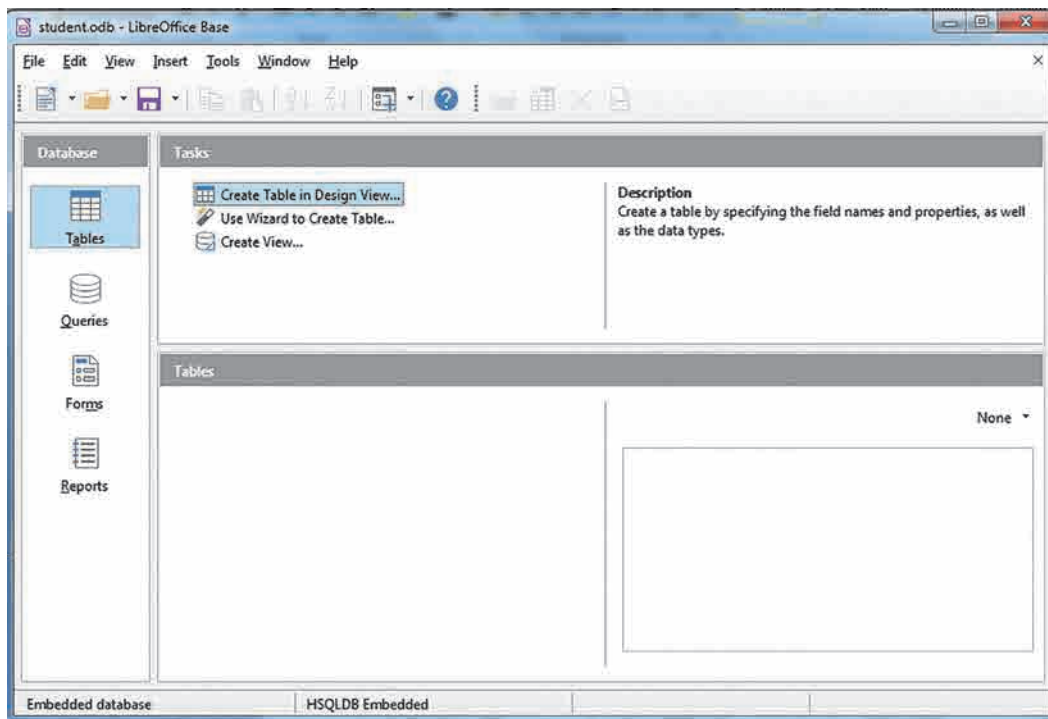


Fig. 5.1 : Database

### Screen of Base consist of following parts :

- A) **Title Bar :** It is the topmost bar present on the screen of Base. It displays icon of the application, name of the file and name of the application. By default it displays 'Filename.odb – Libre Office Base'. It consist of three buttons on right corner as minimize, maximize/ restore and close.
- B) **Menu Bar :** It is present below Title bar. It displays names of different menus as File, Edit, View, Insert,

Tools, Windows, Help etc. Each menu consist of a drop-down list (Pop-up) of various options related to that particular menu.

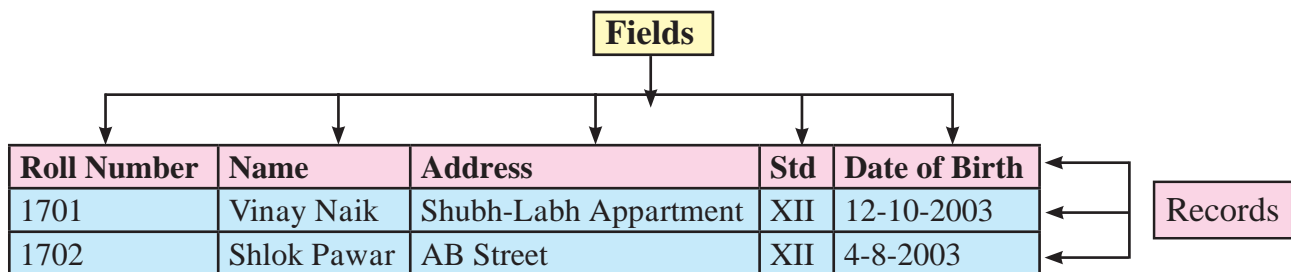
- C) **Standard Tool Bar :** Standard tool bar consist of different icons which are used for standard operations (regularly repeating operations) like opening a new document, saving a document, printing a document, cut, copy, paste, undo, redo and many more.
- D) **Working Area :** Rest of the part

below standard tool bar is called as working area. It is divided into two panes- Left pane and right pane. Left pane displays name of database objects like tables, queries, forms and reports. Right pane displays activities related to that particular object.

### 5.3 Table

Table is a basic unit for storing data in database. Tables are organised in the

form of columns and rows. Before creating a table user should first decide the entity. Entity is any real world object about which data is to be stored. Each entity has collection of attributes associated with it. For example student is an entity which has attributes like roll number, name, address, standard, date of birth etc. Attributes of an entity are stored in the form of columns. The information stored under each column forms a row which is called as record/tuple.



### Data types in Base :

Data types available in Base can be divided into three categories as alphanumeric, numeric, calender (date and time) and binary type.

Data Type Name	Description
Tiny Integer [TINYINT]	Stores small Integer.
BigInt[BIGINT]	Stores Big Integer(hopping whole number), rarely used
Image[LONGVARBINARY]	Stores Image.
Binary[VARBINARY]	Stores binary information of variable length.
Binary(fix)[BINARY]	Stores binary data of fixed length.
Memo[LONGVARCHAR]	Stores descriptive type of information i.e large block of text like article text.
Text(fix)[CHAR]	Stores fix sized text.

Data Type Name	Description
Number[NUMERIC]	Stores a number with or without decimal point. Base will display datatype's original length and allow to set decimal places after decimal point in design view of table. it is used when accuracy is required.
Decimal[DECIMAL]	
Integer[INTEGER]	Stores Integer most commonly used data type.
Small Integer[SMALLINT]	Stores integer
Float[FLOAT]	Stores a number with decimal point.
Real[REAL]	Stores a number with or without decimal point. It is used when approximate result required
Double[DOUBLE]	
Text[VARCHAR]	Stores variable sized text.
Text[VARCHAR_IGNORECASE]	A case insensitive version.
Yes/No [BOOLEAN]	Stores boolean type of data.
Date[DATE]	Stores date in mm/dd/yy format.
Time[TIME]	Stores time in hh:mm:ss format
Date/Time [TIMESTAMP]	Stores date as well as time.
Other[OTHER]	Stores any other object.

**Note :** When it is confirmed that number will be accepted as a input for a field and not used in any kind of calculations, then it is better to have that field's data type as text. For example pincode, mobile number are numeric but not used in any calculation so it can have text data type.



### Do it Yourself

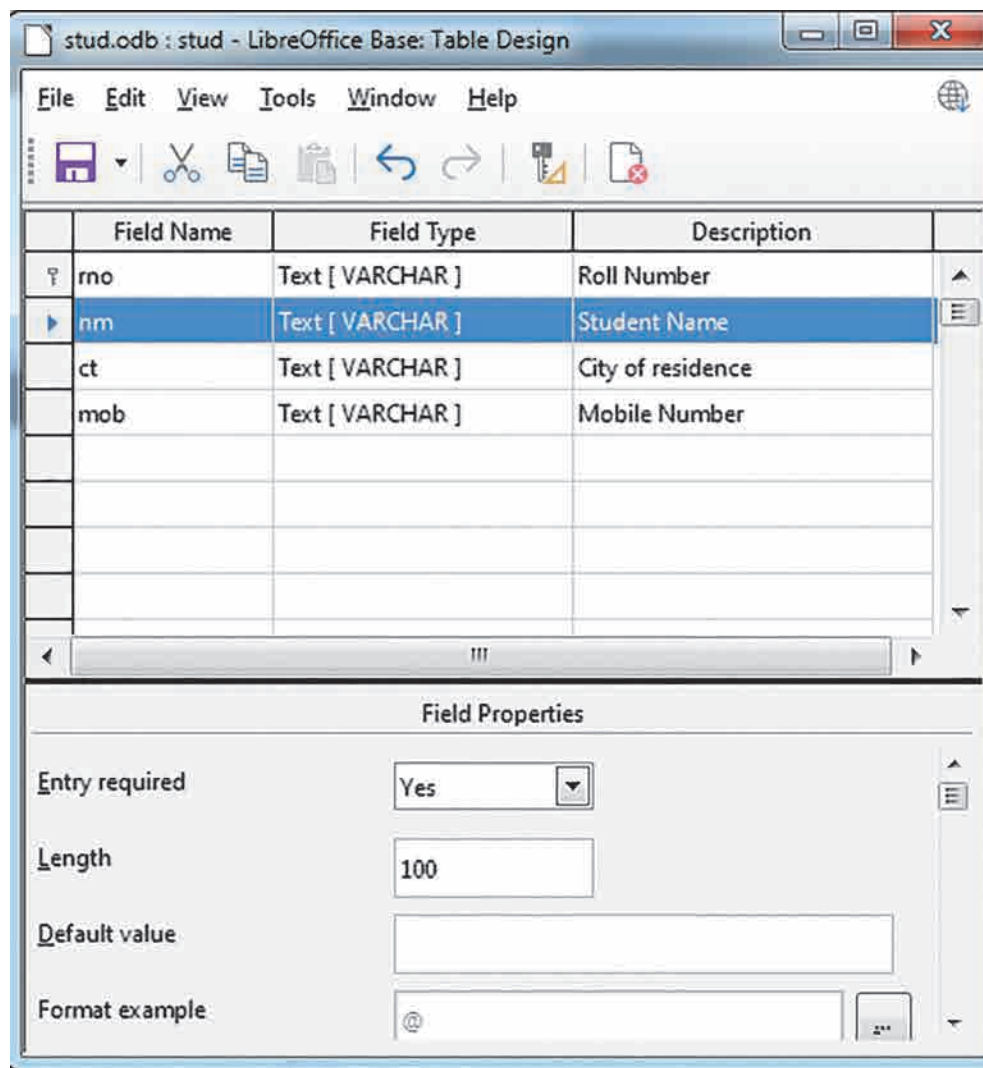
- Find the rules for naming field names in Libreoffice Base.

### 5.3.1 Creating a table :

Steps are as follows :

1. Open a database and from left pane click on 'Tables' icon
2. From right pane click on 'Create table in Design View'
3. A window will appear on the screen, type 'Field Name'

4. Select 'Field type' ( field type means type of data to be stored in that field e.g. 'Name' Field should have 'Text' as a Field Type)
  5. In next column type the description about the field.
  6. In the bottom section in 'Field Properties' select 'Yes' for 'Entry required' property.
  7. You can set default value and for numeric field types you can set decimal places also. In this way create all fields.
  8. Right click on the small button present on the left side of the field name which is decided as primary key and select 'primary key' option.
  9. Click on 'Save' icon to save the table, type name for the table, Clic kon 'ok'.
- Following fig.5.2 displays creation of a table with it's required fields and datatypes.




**Fig. 5.2 : Table Creation**

## B) Inserting records in the table :

Steps are as follows :

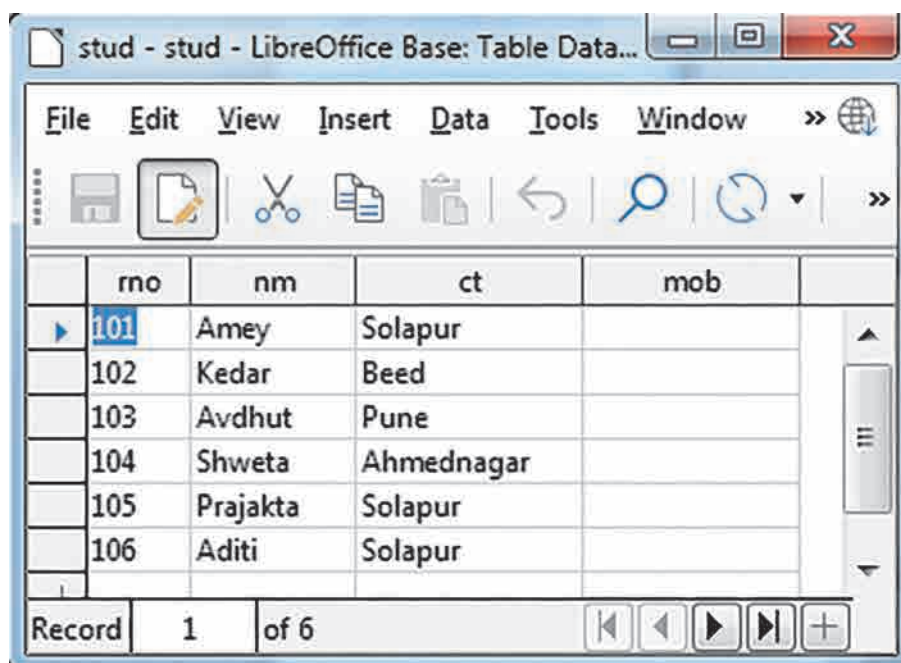
1. Open a database , from left pane click on 'Tables' icon.
2. In the right pane from bottom part double click on the name of Table in which data is to be inserted.
3. A window will appear on the screen, type data under each field and

complete one record, in this way add some records.

4. Click on 'Edit Data'  icon to make edit mode off, a window will appear.
5. Click on 'Yes' icon and click on 'close' button.

For inserting a new record 'Edit Data' must be on.

Following figure displays records are inserted into the table.



	rno	nm	ct	mob
▶	101	Amey	Solapur	
	102	Kedar	Beed	
	103	Avdhut	Pune	
	104	Shweta	Ahmednagar	
	105	Prajakta	Solapur	
	106	Aditi	Solapur	

Record 1 of 6

**Fig. 5.3 : Record insertion**

## Editing a record in the table :

Steps are as follows :

1. Double click on the required table.
2. Click on the required record for editing.
3. Do necessary changes.
4. Click on 'Edit Data' icon to make edit mode off, Click on 'Yes' and click on close button.

## Deleting a record from the table :

Steps are as follows :

1. Double click on the table , click on the record which is to be deleted
2. Click on 'Edit' menu, select 'Delete Record' option
3. Click on 'Yes' button to delete it permanently.



### Do it Yourself

- Create a table - Product with fields- pid, pnm, pdesc, pmonth, pwt, prate etc.
- Create a table- Event with appropriate fields.

## 5.4 Query Creation

A Query is a question asked within the database environment. For example how many students are in XII standard. Query displays subset of data contained in various tables of database Query is used to retrieve records from the table.

### Steps to create query by using wizard :

1. Open a database , from left pane click on 'Queries' objects.
2. From right pane click on 'Use Wizard to Create Query' (Wizard means step by step instructions provided by the computer to complete a task. )
3. Select name of 'Table' and shift fields from 'Available fields' list to 'Fields in

the query' list by clicking on arrow button and click on 'Next' button.

4. Select a field for sorting the records, select the order ascending/ descending then click on 'Next' button.
5. In next step the user has to select search condition (more than one search conditions can also be selected) select the field, select the condition, type value for condition then click on 'Next' button.
6. If you want to replace original field names with some new field names in the output of the query, then type aliases for each field and click on 'Next' button
7. Type name for the query and Click on 'Finish' button.

Following fig. displays output by executing a Query [a query is written for displaying list of records where City='Pune']\

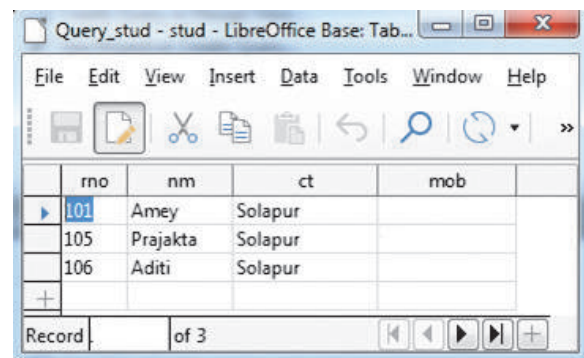
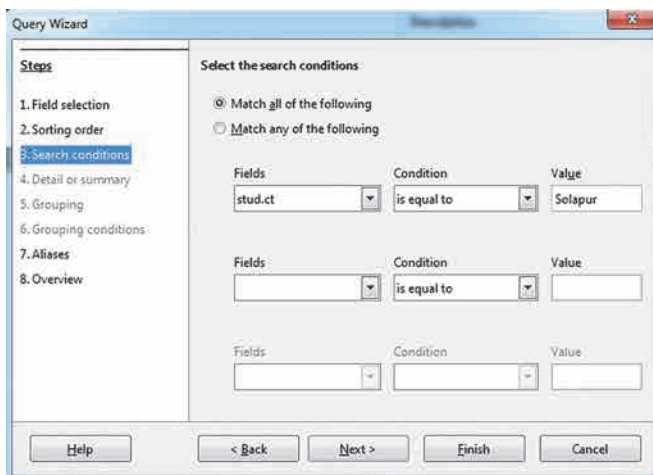


Fig. 5.4 : Output of a query

### Steps to create query in design view :

1. Open any saved database, click on 'Queries' object and in right pane click on 'Create Query in Design View' option.
2. From Add Table window select table and click on add button. Take two tables which have at least one common field. Drag the common field from first table to second table. A line will be displayed as a link between these two tables as they have one common field for joining. Close the add table window by clicking on close button.
3. In bottom section click inside first

column of 'Field' row, select one by one field to be displayed in query output, set each field's Visible property 'On'. We can type alias for each field.

4. To calculate total of marks, in field name column, type formula for addition of subjects, for example phy+chem+math. Then in next column, to calculate Percentage type formula as (phy+chem+math)/3.
5. To execute query click on 'Run Query' icon or press F5 function key. The output will be displayed in same window in above portion. This output can be saved for further use.

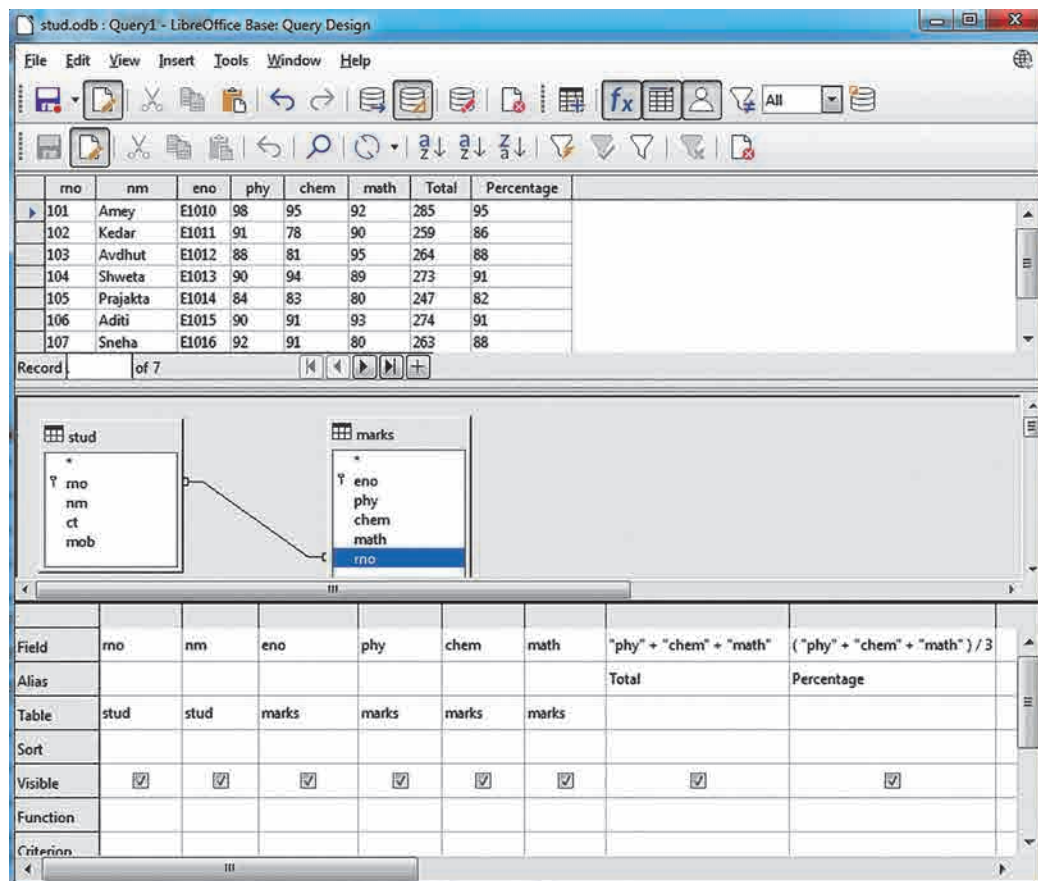


Fig. 5.5 : Query creation in Design View



### Do it Yourself

Create queries for following on product table.

1. To display product list which are manufactured in the month of December.
2. To display product list whose rate is greater than 100.

## 5.5 Form Creation

Form is an object which allows entering the data and editing or deleting existing data in the table. It consists of format, style and widgets like radio button, list boxes that provides easy and systematic way to insert records in the table Form is used to collect the data from the user. Steps to create a form are as follows :

1. Open a database, from left pane click on 'Forms' objects.
2. From right pane click on 'Use Wizard to Create Form'.
3. Select name of 'Table' and shift fields from 'Available fields' list to 'Fields in the form' list by clicking on arrow button then click on 'Next' button
4. Click on 'Next' button.
5. Select any one arrangement for main form for placing the controls (Ex-columnar labels left) and click on 'Next' button.
6. Click on 'Next' button.
7. Select style for the form , click on 'Next' button

8. Type Name for the form , click on 'Finish' button
9. To add new record click on 'New Record' icon present on form navigation tool bar(present at bottom side), fill up the record and click on 'Save record' icon. Click on 'Close' button to close the form window  
Following figure displays a form :

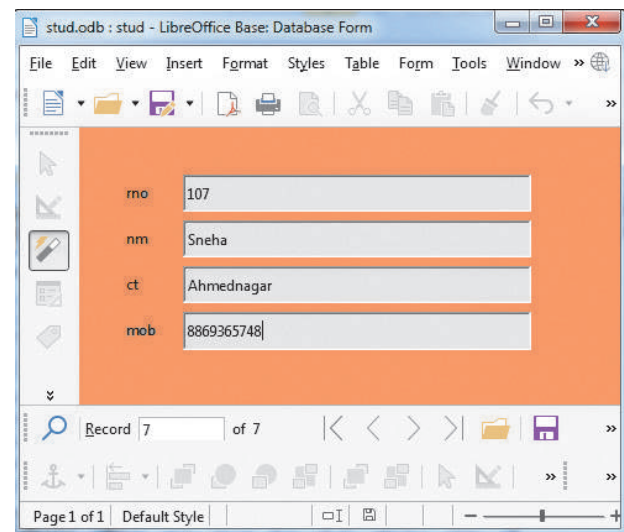


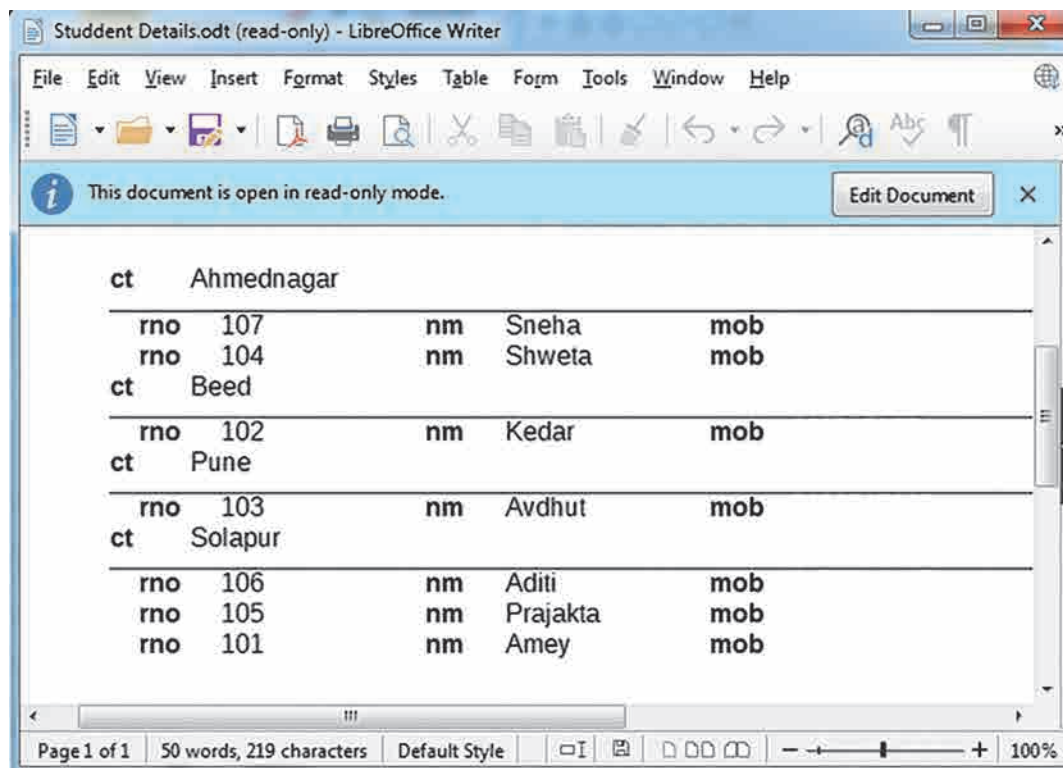
Fig. 5.6 : Form

## 5.6 Report Generation

The presentation of information in an organised and readable format as per the user's requirement is known as report. Various complex reports can be generated that can help in taking decisions by the management. Report is the representation of data in printed form. Steps to create a report are as follows :

1. Open a database, from left pane click on 'Reports' objects
2. From right pane click on 'Use Wizard to Create Report'
3. Select name of 'Table' and shift fields

- from 'Available fields' list to 'Fields in Report' list by clicking on arrow button, click on 'Next' button
4. Labels for the fields can be changed with new labels to display in report and click on 'Next' button
  5. If you want to see the records group-wise, (Ex-citywise) add a grouping level click on a field, click on arrow button and click on 'Next' button
  6. Select a field to sort the data, click on 'Next' button
  7. Choose Layout and Orientation, click on 'Next' button
  8. Type 'Title for the Report', click on 'Finish' button
  9. Report will be displayed in read-only mode in the form of 'LibreOffice Writer' file. Click on 'Close' button to close the report window. After Report generation, screen will be displayed as follows :

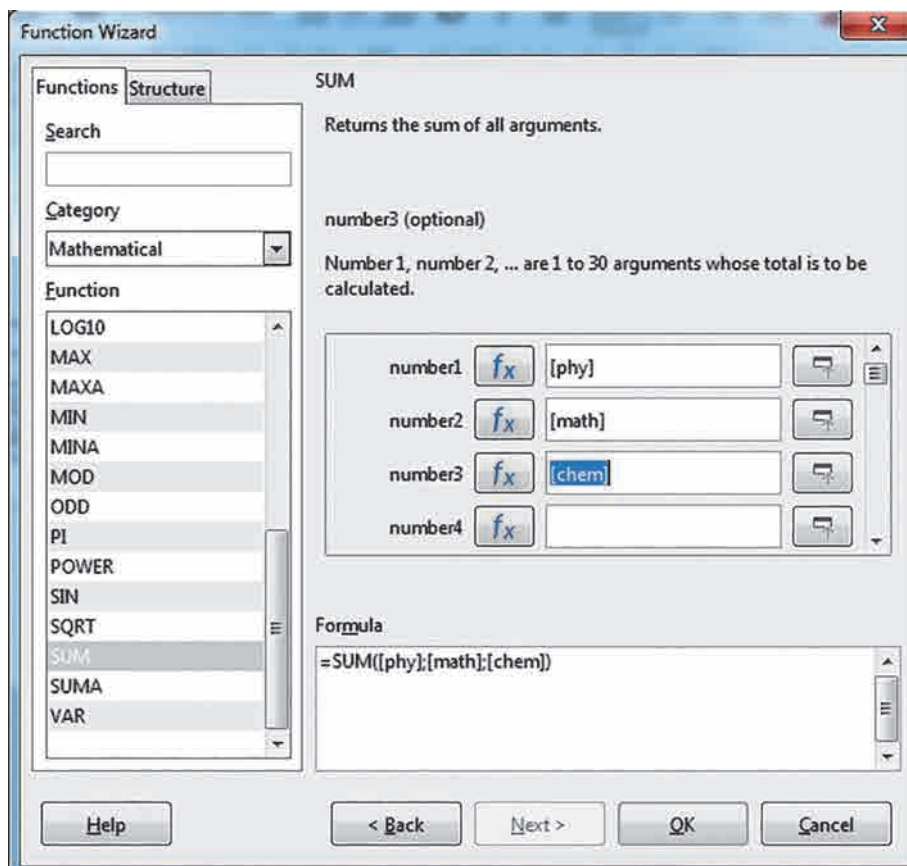


**Fig. 5.7 : Report**

**Steps to create report in design view :**

1. Open any saved database, click on 'Reports' object and in right pane click on 'Create report in design View' option.
2. From right side, setting window(can be made on/off by clicking setting button from tool bar) of the window, click on 'Data' tab and for 'Content' select table name or any saved query name.
3. A small window will appear, click on field name and click on 'Insert' button, close that window. Here we have

- selected 'Marks' table and rno, phy, math, chem fields from the same table.
4. Now to calculate total of marks draw a table and a textbox below all fields. Change table property of table as 'Total'. Click on textbox and from setting window click on 'Data' tab. Set 'Data Field Type' value as 'Field or Formula'. For 'Data Field' click on small button present beside of that field, a function wizard will start, select 'Sum' function, click on 'Next' button. Select field name to be used for addition by clicking on icon present on right side of the textbox (one field in each textbox to be selected) and click on 'Ok' button.
  5. We can give some title for the report in 'Page Header' section by using 'Label' control and changing it's label property to required Title.
  6. To see output, click on 'Execute Report' button on standard tool bar.
  7. Output will be displayed in 'Libre Office Writer' window in 'Read-Only Mode' but if we want to edit it then we can click on 'Edit Document' button, that will appear in the blue ribbon at the top of the report and save it. It will be saved as a Writer file. Report can be saved for further use or can be printed to make hard copy.



**Fig. 5.8 : Function Wizard**

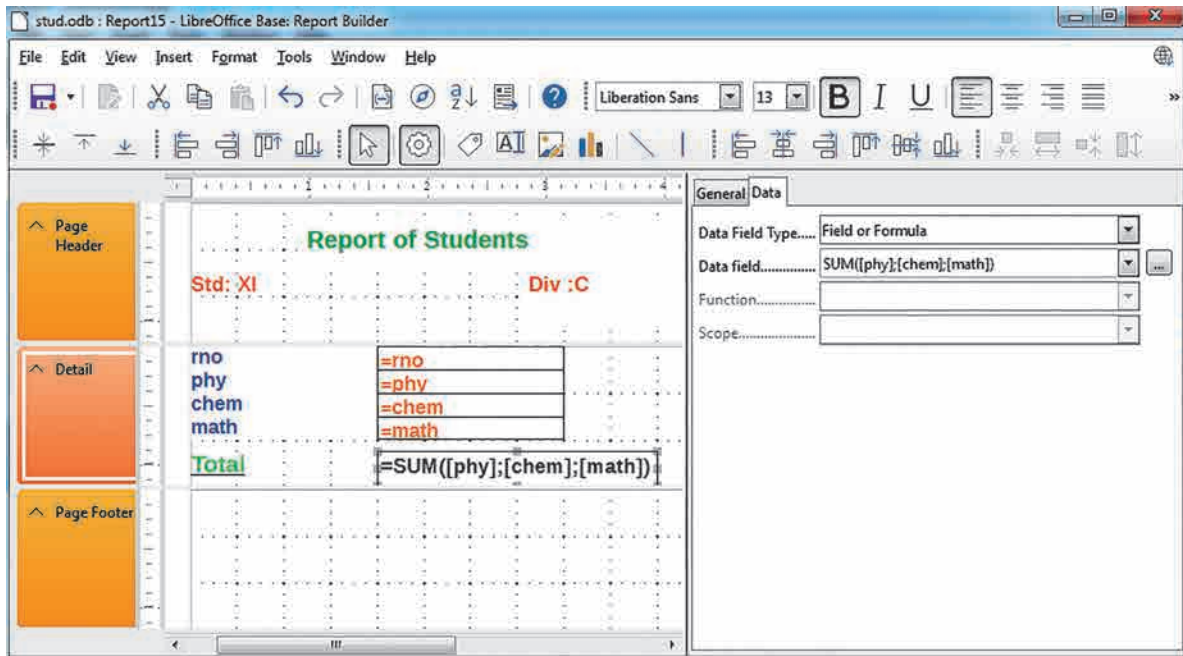


Fig. 5.9 : Design View of a Report

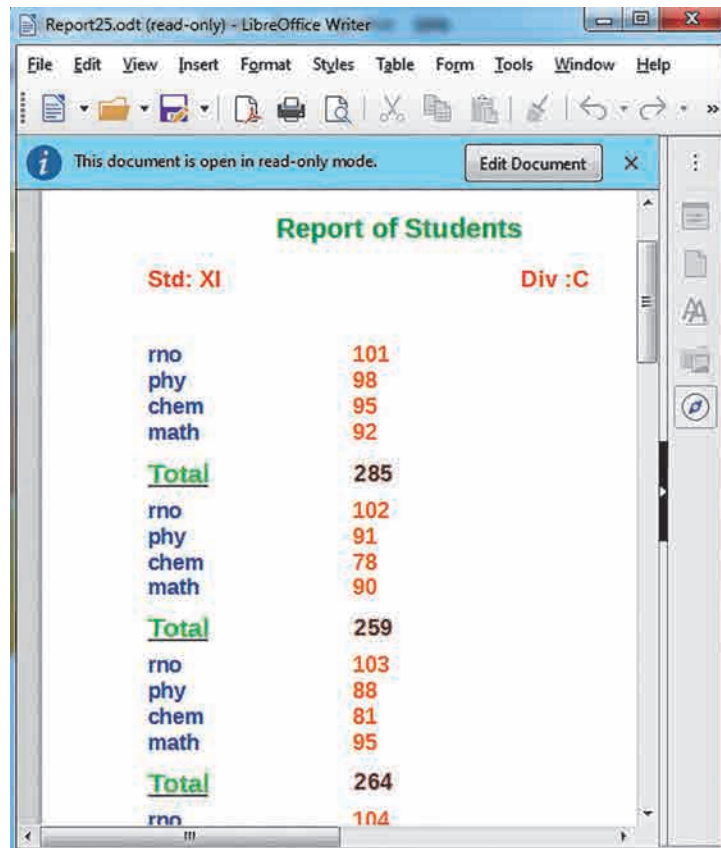


Fig. 5.10 : Print View of the Report



### Do it Yourself

Generate monthly manufactured product's report.

## 5.7 Introduction to Data Model

Data model defines how the logical structure of a database is modeled. Data model defines how data is connected to each other and how they are processed and stored inside the system. Different types of DBMS are available and their classification is done based on the underlying data model. There are many types of data models such as relational data model, network data model, hierarchical data model, object-oriented data model, entity-relationship data model etc. In this book we are introducing Relational Data Model.

- **Relational Data Model** : The most commonly used data model is Relational Data Model. The DBMS following relational data model is called as relational database management system(RDBMS). It is specifically designed for relational databases. A relational database refers to a database that stores data in a structured format, using rows and columns. This makes it easy to locate and access specific values within the database. It is "relational" because the values within each table are related to each other. Tables may also be related to other tables. In relational model, tables are called relations that store data for different columns. Each

table can have multiple columns where each column name should be unique. Each row in the table represents a related set of values. Let us now understand the commonly used terminologies in relational data model:

- Attribute** : Characteristic or parameters for which data are to be stored in a relation. The columns of a relation are the attributes which are also referred as fields.
- Tuple** : Each row of data in a relation (table) is called a tuple. In a table with n columns, a tuple is a relationship between the n related values.
- Domain** : It is a set of values from which an attribute can take a value in each row. Usually, a data type is used to specify domain for an attribute. Every attribute has some pre-defined value scope, known as attribute domain For example, in Student relation, the attribute Roll\_Number takes integer values and hence its domain is a set of integer values.
- Degree** : The number of attributes in a relation is called the Degree of the relation.
- Cardinality** : The number of tuples in a relation is called the Cardinality of the relation.

### • Relation : Student

**Table is called as Relation**

RollNumber	Name	City	BranchCode	MoNumber
101	Amey	Solapur	CS	
102	Shweta	Ahmednagar	ELE	
103	Avdhut	Pune	MECH	

Primary Key (points to RollNumber)

Domain=number(expected value) (points to MoNumber)

Row or Tuple Cardinality=3 (Total # of rows)

Column or Attribute Degree=5 (Total # of Columns)

Foreign Key(references (Branch relation)) (points to BranchCode)

- **Relation : Branch**

Primary Key



BranchCode	BranchName
IT	Information Technology
ELE	Electronics
MECH	Mechanical

- **Keys in a Relational Database :** The tuples within a relation must be unique. It means no two tuples in a relation should have same value for all attributes. That is, there should be at least one attribute in which data is unique and not NULL. So, we can uniquely identify each tuple of a relation. So, relational data model imposes some restrictions (constraints) on the values of the attributes and how the contents of one relation be referred from another relation. These restrictions are specified at the time of defining the database through different types of keys as given below :

**1. Candidate Key :** A relation can have one or more attributes that takes unique values. Any of these attributes can be used to uniquely identify the tuples in the relation. Such attributes are called candidate keys as each of them are candidates for the primary key.

In above example the relation student has five attributes out of which Roll\_Number and Mo\_Number always take unique values. No two students will have same roll number or same mobile

number. Hence, these two attributes are the candidate keys as they both are candidates for primary key.

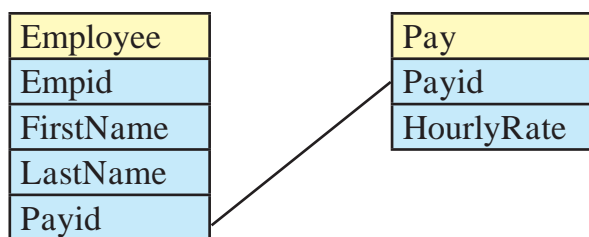
- 2. Primary Key :** Out of one or more candidate keys, the attribute used to uniquely identify the tuples in a relation is called the primary key of that relation.
- 3. Composite Primary Key :** If no single attribute in a relation is able to uniquely distinguish the tuples, then more than one attribute are taken together as primary key. Such primary key consisting of more than one attribute is called Composite Primary key.
- 4. Foreign Key :** A foreign key is used to represent the relationship between two relations. A foreign key is an attribute whose value is derived from the primary key of another relation. This means that any attribute of a relation (referencing), which is used to refer contents from another (referenced) relation, becomes foreign key if it refers to the primary key of referenced relation. The referencing relation is called Foreign Relation.

In above example Branch\_Code is foreign key in Student relation whereas it act as a primary key in Branch relation. Student relation is called as referenced relation and Branch relation is called as foreign relation.

There are 3 types of relationships in relational database design. They are as follows:

### 1. One-to-One (1:1) :

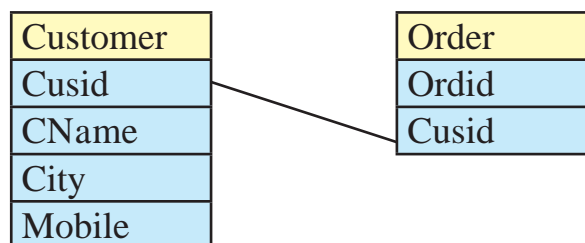
A row in table A can have only one matching row in table B, and vice versa. This is not a common relationship type, as the data stored in table B could just have easily been stored in table A. However, there are some valid reasons for using this relationship type. A one-to-one relationship can be used for security purposes, to divide a large table, and various other specific purposes.



In the above example, we could put an HourlyRate field straight into the Employee table and not bothered with the Pay table. However, hourly rate could be sensitive data that only certain database users should see. So, by putting the hourly rate into a separate table, we can provide extra security around the Pay table so that only certain users can access the data in that table.

### 2. One-to-Many (or Many-to-One) (1:M) :

This is the most common relationship type. In this type of relationship, a row in table can have one or many matching rows in table B, but a row in table B can have only one matching row in table A. One-to-Many relationships can also be viewed as Many-to-One relationships, depending on which way we look at it.

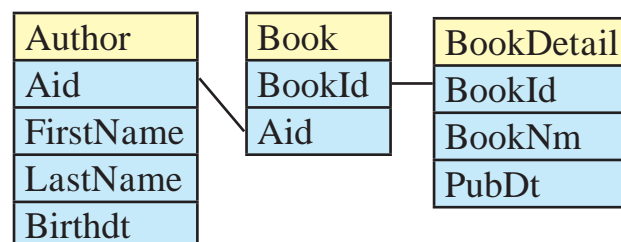


In the above example, the Order relation is the "many" and the Customer relation is the "one". Each Order can only be assigned one customer where as one customer can be assigned to many orders.

### 3. Many-to-Many (M:M) :

In a many-to-many relationship, a row in table A can have many matching rows in table B, and vice versa. A many-to-many relationship could be thought of as two one-to-many relationships, linked by an intermediary table. The intermediary table is typically referred to as a "junction table" (also as a "cross-reference table"). This table is used to link the other two tables together. It does this by having two fields that reference the primary key of each of the other two tables.

For example list of books, and a list of authors. Each book may have one or more authors, and each author may have written multiple books. In this case, you have many books related to many authors.



## Steps to develop relationship between relations in LibreOffice Base :

1. Create relations (tables) with one field common which must be a primary key of first table and the same key is referenced in another relation and called as foreign key in that table.
2. Click on 'Tools' menu and select 'Relationships' option.
3. A small window will appear, select table name and click on 'Add' button.

Place both table on the screen, now drag the common field from first table to second table. A line connecting two tables will be displayed this is called as relationship. To delete relationship just right click on line and select 'Delete'. To set some more settings about relation, right click on line and select 'Edit' option. A window will appear where we can set various update and delete options for the relationship.

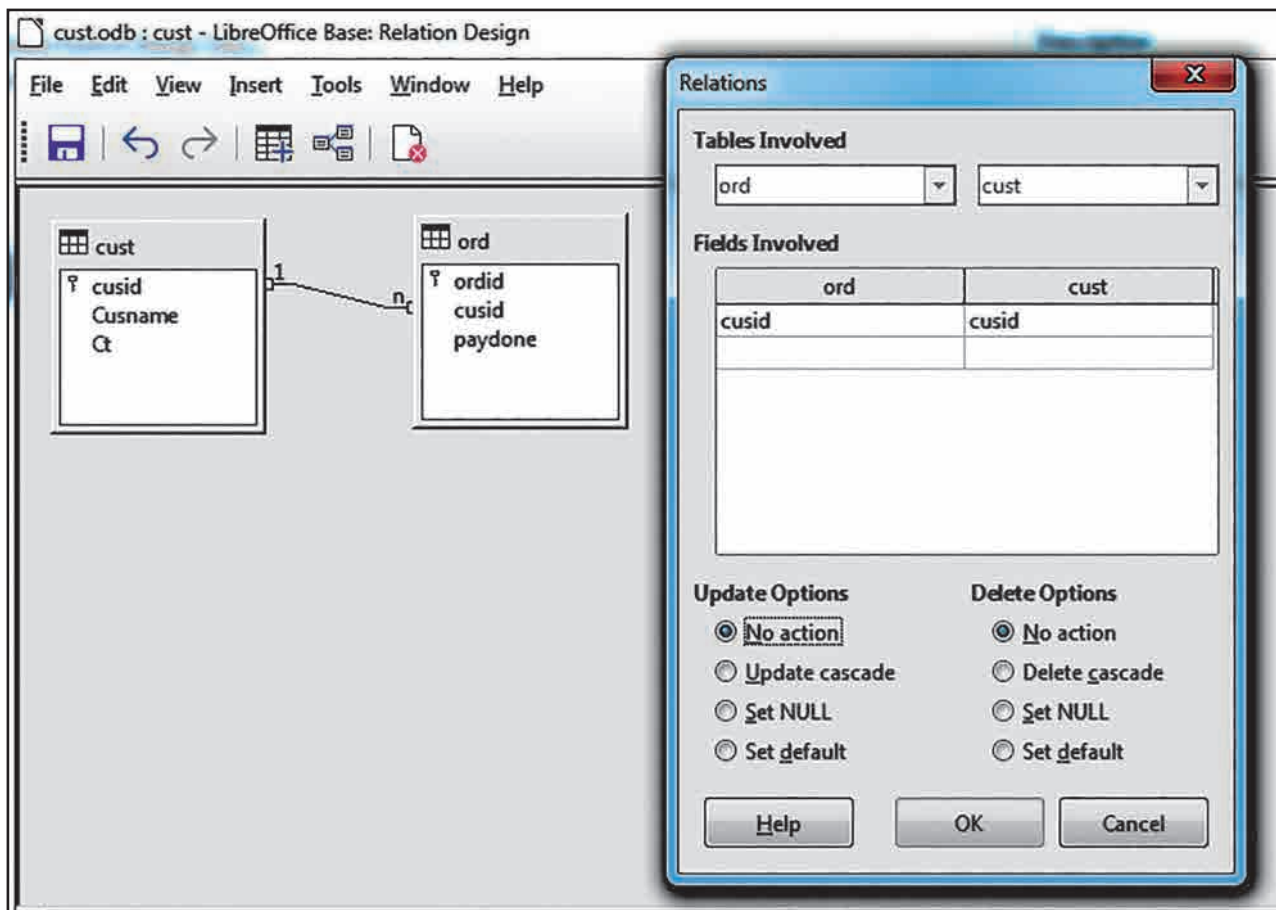


Fig. 5.11 : Relationship



### Do it Yourself

Find out databases with different relations and relationships.

## Summary

- Data means all kinds of facts, figures & details related to people, places, things or events.
- Information is the result obtained from processing of data.
- Database is a collection of related data items stored in an organised manner.
- Table is a collection of rows and columns. User can insert, edit and delete records from table.
- User can create a query to retrieve/display records from table.
- Report is a printed form of data.
- Data model defines logical structure of a database.
- Types of relationships are 1:1, 1:M, M:M etc.

## Exercise

### Q.1 Fill in the blanks.

1. .... is a collection of related data.
2. Queries are used to ..... information from database.
3. The representation of data in printed form is called as .....

### Q.2 State True/False.

1. Form is used to collect the data from the user.
2. Menu bar is present below Title bar.
3. Columns are called as records.

### Q.3 Multiple Choice Question. (1 Correct Answer)

1. Rows in Base are called as .....  
a) records                      b) fields  
c) table                          d) database

2. File extension of Base is .....

- |         |         |
|---------|---------|
| a) .odt | b) .ods |
| c) .odb | d) .odp |

### Q.4 Answer in brief.

1. Define database.
2. What is a query?
3. Define report.
4. Explain working area of Base.

### Q.5 Match the following.

- | A         | B                                 |
|-----------|-----------------------------------|
| 1. Query  | a) Collect information from user. |
| 2. Report | b) Collection of related data.    |
| 3. Form   | c) Retrieve data from database.   |
| 4. Table  | d) printed form of data.          |

