2 Introduction to LibreOffice Base

Introduction

LibreOffice Base is a free and open source desktop database tool. It comes as a part of LibreOffice Suite. It supports major database engines like MySQL, MS Access and PostgreSQL. Base includes the HSQL relational database engine as default. Users can create tables, queries, forms and reports with the help of wizards and pre-defined templates. Additionally, Base integrates well with other LibreOffice applications, providing data for mail merges in Writer and creating linked data in Calc for analysis. In this chapter, we will learn how to use LibreOffice Base and create a database and tables.

Opening Base

LibreOffice Base is not installed by default in Ubuntu Linux due to its dependencies on Java. So before we want to use it we need to install it. One of the simplest ways to install it on Ubuntu Linux is to run the command "*sudo apt install libreoffice-base*" in the terminal window. Alternatively you can download it from the official LibreOffice website. Once installed we can start using it.

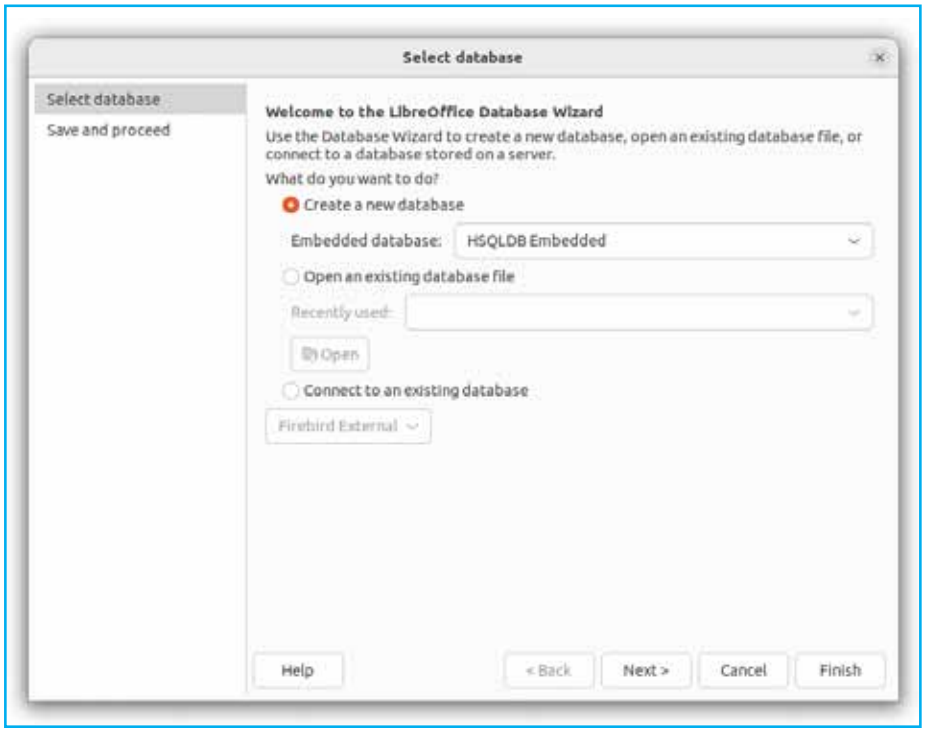

Figure 2.1 : Initial Screen of Database Wizard

You can open LibreOffice Base in multiple ways. The simplest is to pin the LibreOffice Base application to the Launcher and click on it. A *Database Wizard* dialog box as shown in figure 2.1 will pop in front of you.

A wizard is a guided, step-by-step graphical interface tool that simplifies complex tasks by prompting users for information and automates the process of performing the task.

Observe that the screen in figure 2.1 provides us with three options that we can choose from; *Create a new database*, *Open an existing database file* and *Connect to an existing database*. We will need to choose one of these options. As can be easily understood, we can create a new database using option one, while other options allow us to work with a database that has been previously created and saved in our computer.

Note: The contents visible on the screen in figure 2.1 may differ based on the type of installation.

Let us try and create a database for the tables that we had prepared in chapter 1 for the school management system. To create this new database select the *Create a new database* option and click on the *Next* button, a screen shown in figure 2.2 will appear.

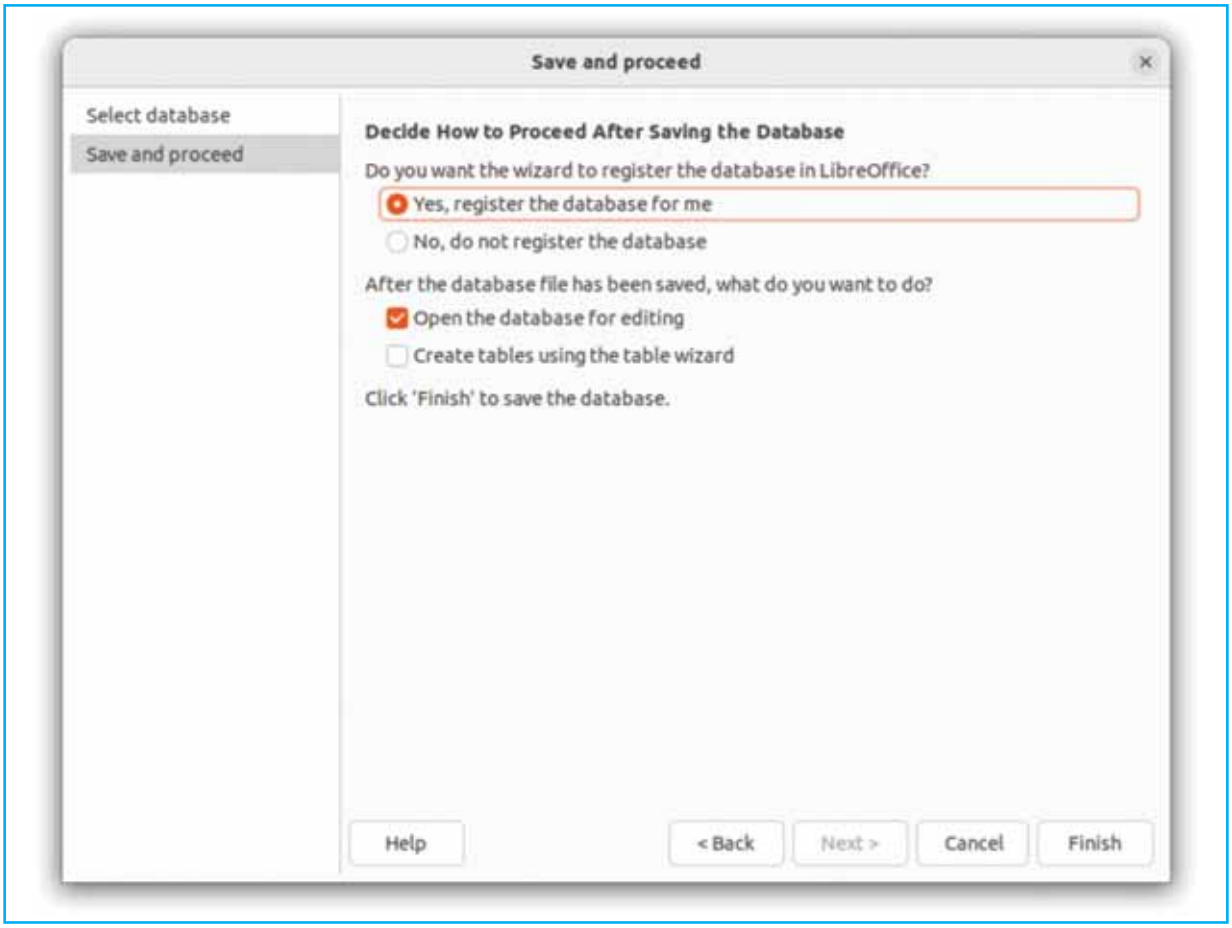

Figure 2.2 : Screen 2 of the Database Wizard

Observe that we need to perform two activities here. First, decide whether we want to register our database in LibreOffice or not? By default, the option *Yes, register the database for me* is selected. If registered the databases will be accessible from other LibreOffice suite applications like Calc and Writer. Second we need to

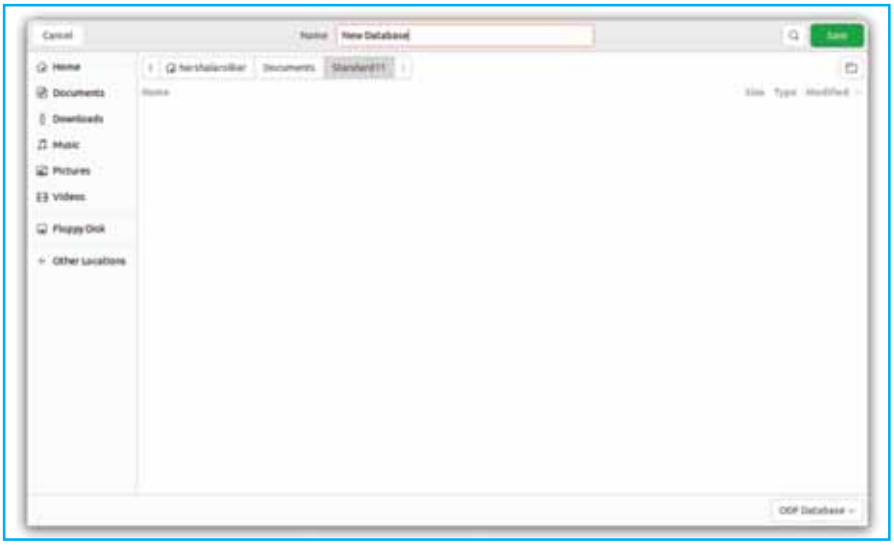

Figure 2.3 : Screen to Save the Database

decide how we want to start working with our database once it is saved. As can be seen, by default the *Open the database for editing* option is checked. We can also select the option *Create tables using the table wizard* if we want to perform both these actions. For now we will first only create the database, click on the *Finish* button. This will open a screen as shown in figure 2.3.

First select an appropriate folder to store the database file. Observe that we are trying to save our database in a folder named Standard11 within the Documents folder of the machine. In the textbox with label *Name* type ScMS instead of *New Database*. Note that by default the *ODF* (Open Document Format) *Database* is selected. Thus LibreOffice Base automatically assigns *.odb* extension to the database file that we want to create. Lastly click on the *Save* button. The database will now be saved and ready for further use and you will be presented with the screen as shown in figure 2.4.

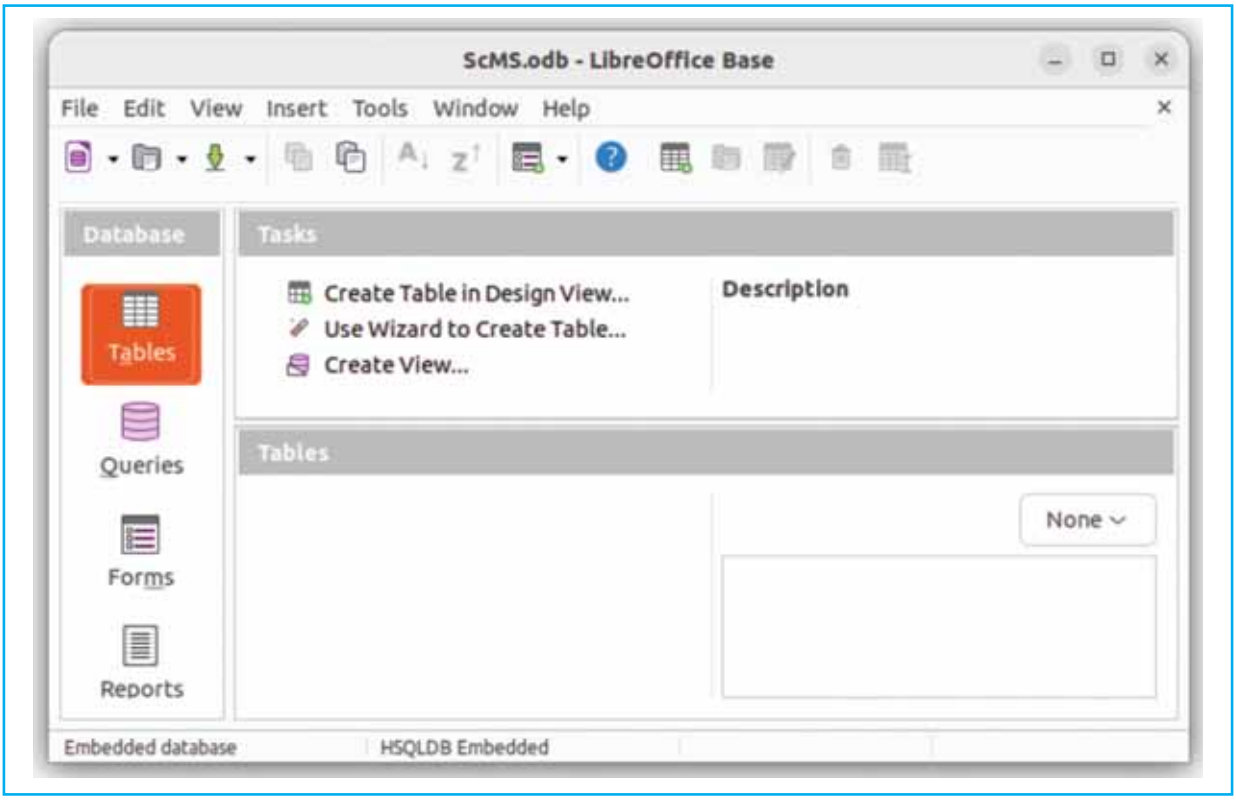

Figure 2.4 : ScMS Database Window

In the left pane, we can see different database objects like *Tables*, *Queries*, *Forms* and *Reports*. By Default, the object Tables would be visible as selected. We are now ready to work with different objects of the ScMS database.

Creating Tables using Wizard

A database without any tables does not make sense, so the next step is to create tables. We will use wizards to create tables. Click on the second option *Use Wizard to Create Table...* under the Tasks pane shown in figure 2.4. A Table Wizard dialog box as shown in figure 2.5 will appear.

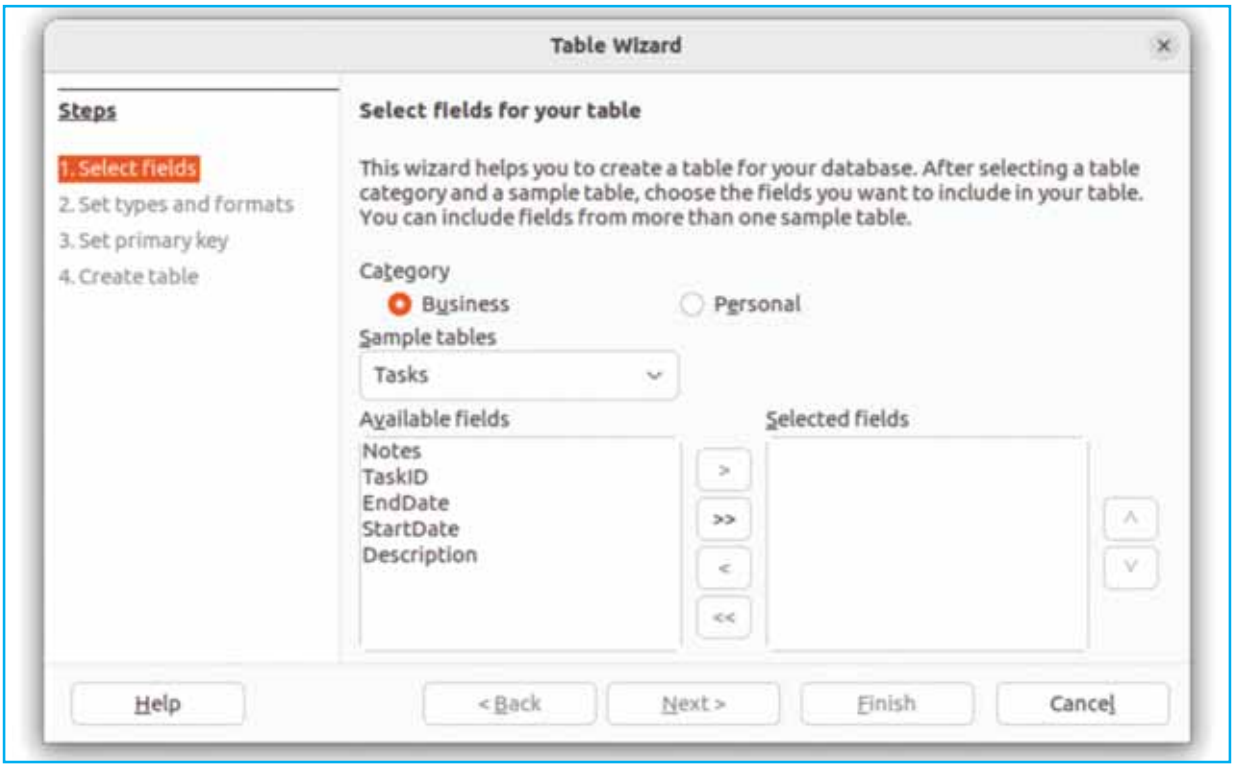

Figure 2.5 : Table Wizard Dialog Box

The *Table Wizard* provides templates of two categories of tables namely, Business and Personal. The business category tables have templates for entities like Tasks, Assets, Events, Orders and others while the personal category tables have templates for entities like Plants, Authors, Libraries, Recipes and others.

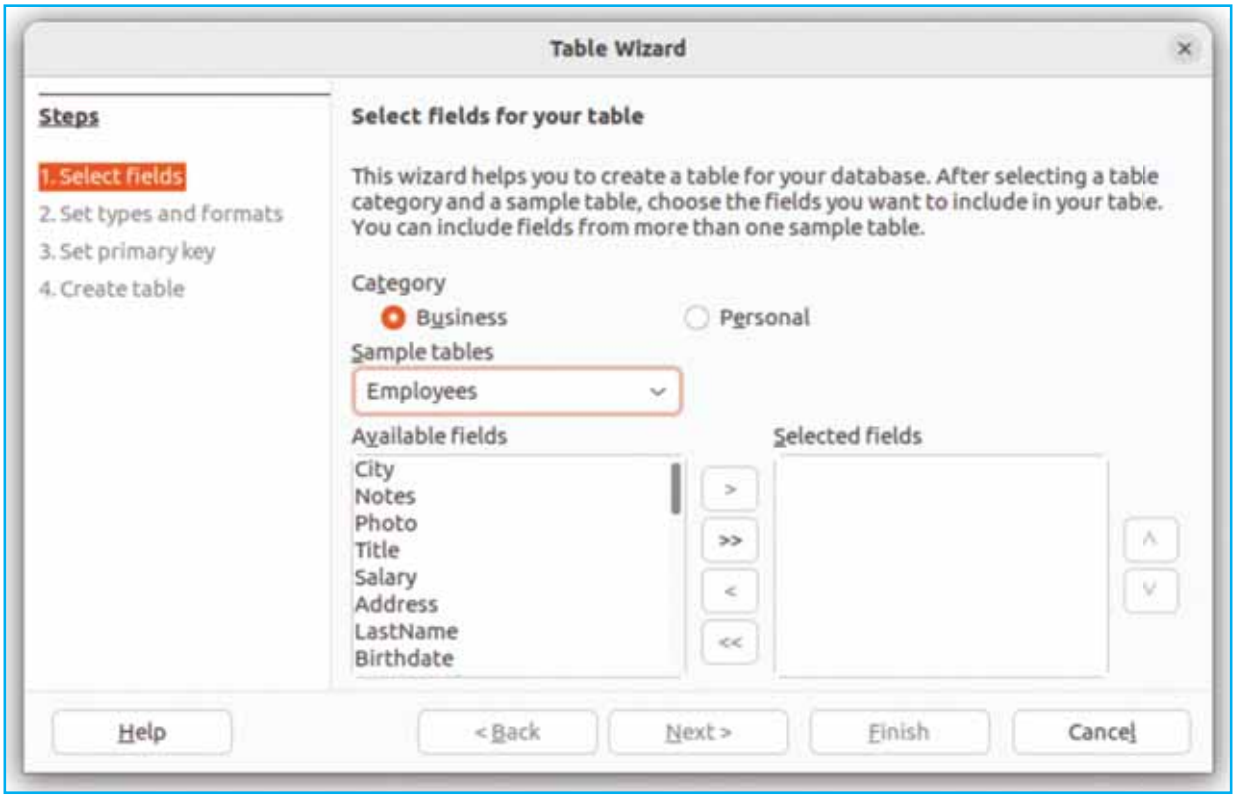

Figure 2.6 : Table Wizard Dialog Box for Creating Employees Table

We will select the Business option under Category label of figure 2.5. Click on the drop down list visible under the label *Sample tables*. For understanding purposes let us create the Employee table. Observe that the drop down list contains a table named *Employees*, select it and we will be presented a list of fields as shown in figure 2.6.

To make all the fields part of our table click on the >> button icon. Alternatively we can choose a single field or set of fields by holding the Ctrl key and selecting desired fields one by one. Let us select fields EmployeeID, FirstName, MiddleName, LastName, Birthdate, Address and DepartmentID. Once we have selected the desired field click on the > button icon to make it part of our table. All the selected fields will now be visible under the *Selected fields* list box as shown in figure 2.7.

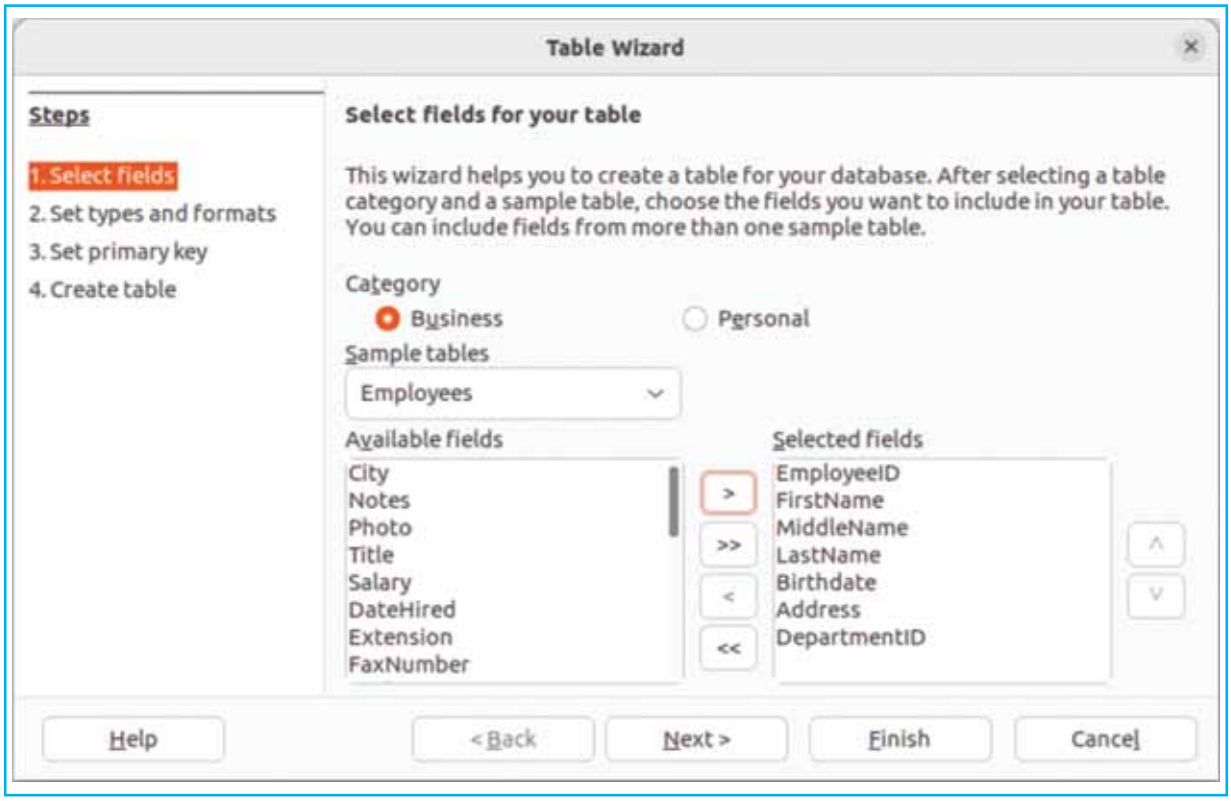

Figure 2.7 : Table Wizard Dialog Box for Selecting Fields of the Employees Table

We can rearrange the order of the fields as per our requirement using the ⋀ and ⋁ button icons. Once all fields have been arranged as per our needs click on the *Finish* button. The Employees table will pop up in a new window known as *Table Data View* as shown in figure 2.8. The *Table Data View* allows users to enter records within the table.

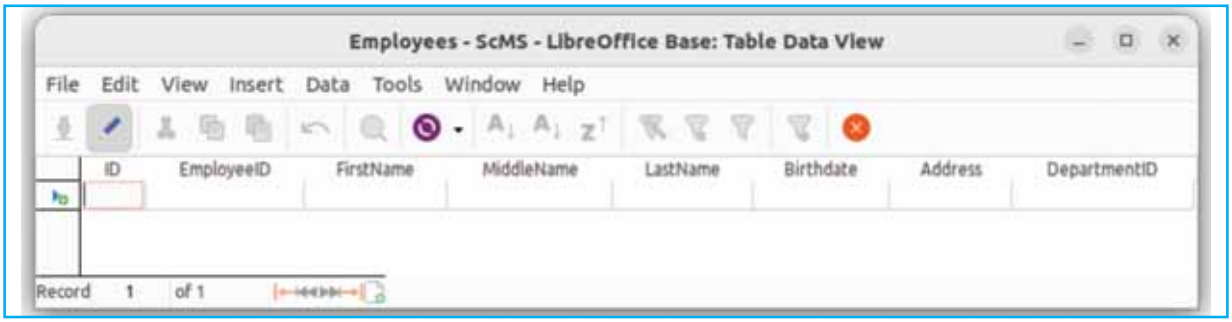

Figure 2.8 : Table Data View of Employees Table

We can start entering data in the table and once data entry is complete we can close the data view. We will be led back to the ScMS database window when we close it.

Creating Tables using Design View

The tables created using wizards many times need to be modified as per the needs of the user. To create a customized table, LibreOffice Base provides an option of Design View. When a table is created or opened in a Design View a user can create, edit, update or delete the fields of the table as per their needs. The table design view allows us to create a customized field name, select the type of data that can be stored in the field, if needed add a description of the field to assist users and also allows us to control and validate the data that can be stored in the field.

Let us now create the Student table having attributes StudentID, FirstName, MiddleName, LastName, BirthDate, Gender, Address and ClassID that we had designed in chapter 1. To create this table click on the *Create Table in Design View*... option of figure 2.4. This will open a blank table design view as shown in figure 2.9.

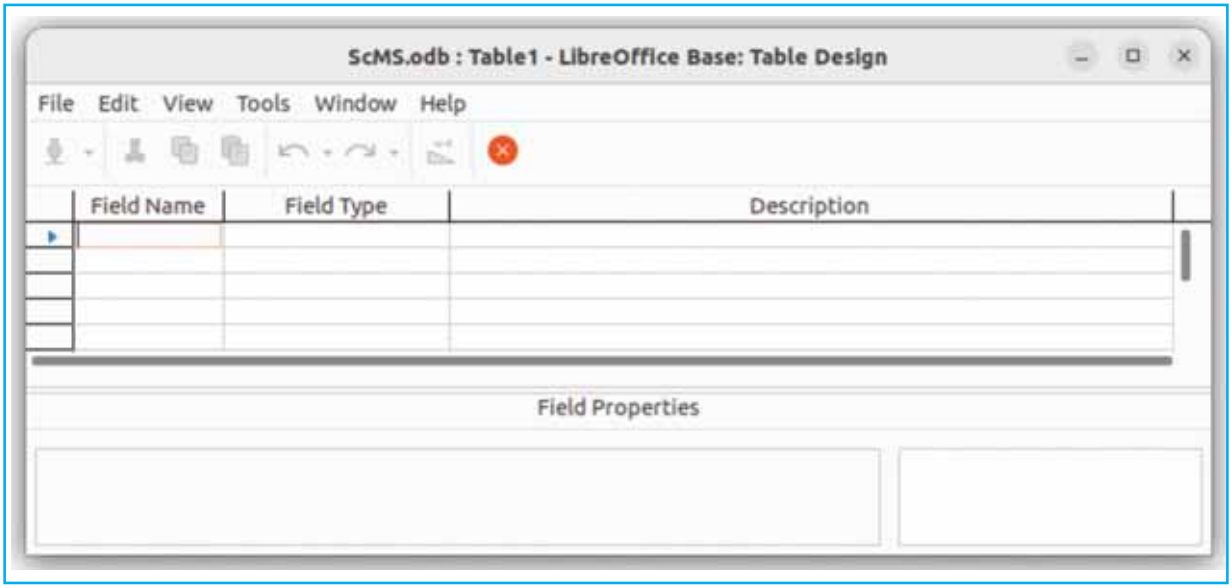

Figure 2.9 : Blank Table Design View

As can be seen, we are provided with a data grid with three columns, *Field Name*, *Field Type* and *Description*. We also have a *Field Properties* pane visible below the table grid. We can now start creating the table structure by typing field names under the *Field Name* column and selecting the required data type under the *Field Type* column. At this juncture, we will not add a description of the fields. Enter all the field names for the Student table as mentioned earlier and select their respective data types. After completion of the process the screen will look something similar to the one shown in figure 2.10.

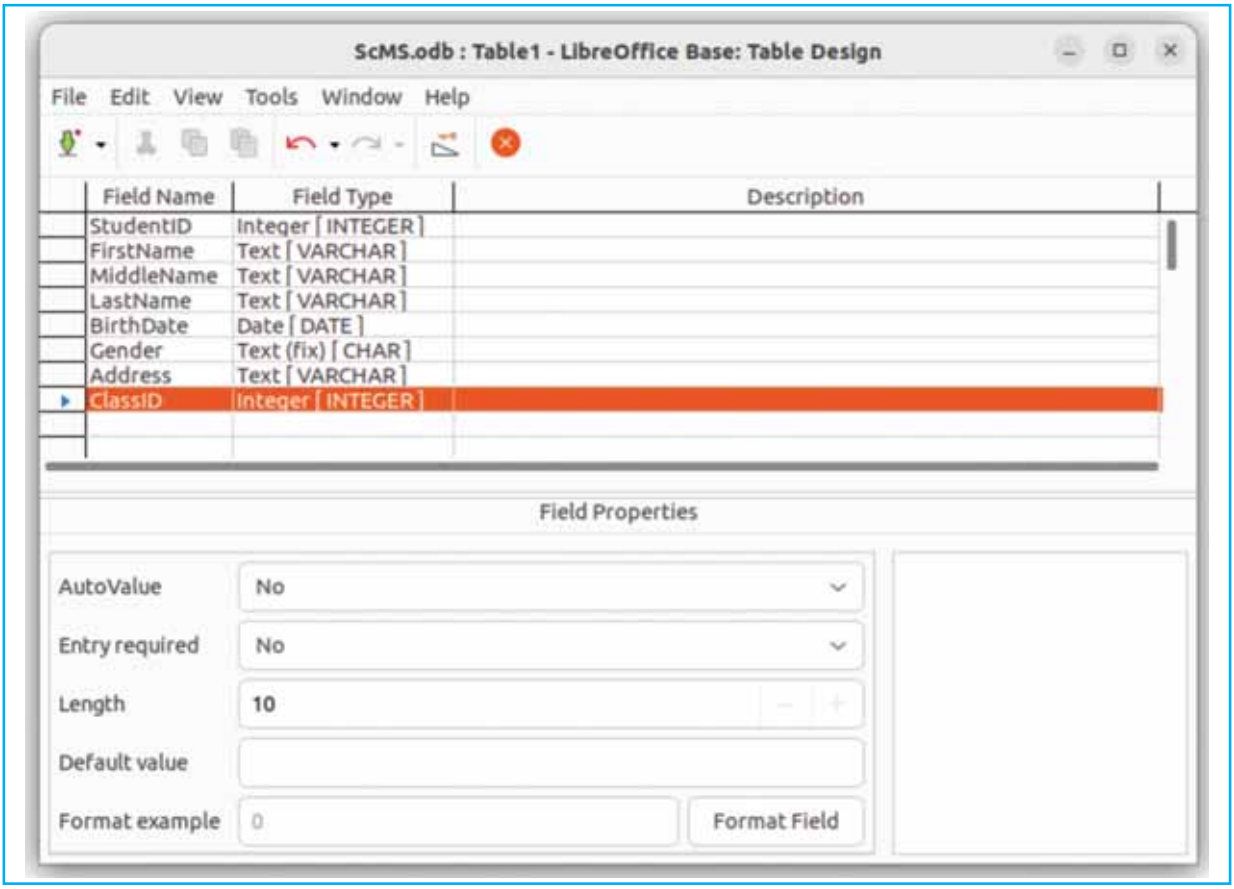

Figure 2.10 : Table Design View with Field Names

Save the table by clicking on the green down arrow with red dot (*Save* button), this will open a *Save as* dialog box as shown in figure 2.11.

Type the name of the table (Student) and click on the *OK* button, a pop up as shown in figure 2.12 will appear.

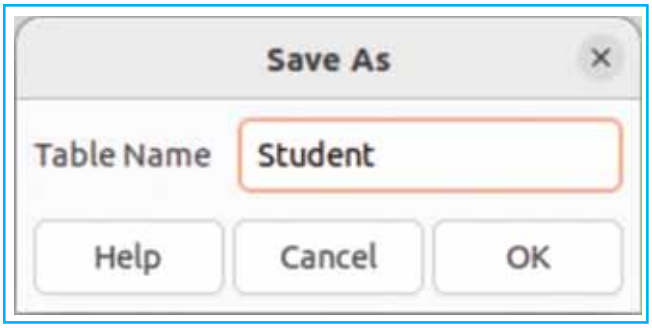

Figure 2.11 : Saving the table

Figure 2.12 : Alert for Primary Key

The screen shows an alert message indicating that the user needs to select a unique primary key. As we learned in Chapter 1, a primary key is a field that uniquely identifies a row in a table. For now click on *No* button, the table is now saved and we can perform required operations on it. Close the table, observe that the ScMS database window will now list two tables Employees and Student.

Modifying the Structure of Table

Till now we have learnt two ways to create tables in LibreOffice Base. At times we need to modify the structure of the tables that we have created. We may need to add a new field, update a name of an existing field, change a data type or may want to add or remove field properties. It is recommended that all the said operations should be performed before entering data into the table. Let us try to modify the Employees table such that it can be used as the Teacher table that we had designed in Chapter 1.

Figure 2.13 : Rename to Dialog Box

The Teacher table has attributes TeacherID, FirstName, MiddleName, LastName, BirthDate, Gender, Address and SubjectID. The Employees table also has a somewhat similar structure. To change the structure of the Employees table, go to the ScMS window, select the Employees table and right click on it. From the drop down menu select the *Rename*... option, we will get a *Rename to* dialog box as shown in figure 2.13.

Change the name from Employees to Teacher and click on the *OK* button. The table will now be renamed and we will be back in the ScMS window.

Once again select the Teacher table and right click on it, from the drop down menu select the *Edit*... option, this will open the table in the design view. Observe that we are able to see a field named ID with a blue triangle icon and a key preceding it. The blue triangle icon is known as the current row pointer that indicates which row is the current row. Select the current row by left clicking on the key. The row will now be highlighted. The ID field was not part of the table design that we had created, so to delete this field right click on the current row icon and a screen similar to the one shown in figure 2.14 will appear.

The menu shown in figure 2.14 can be used to perform different operations. To remove the field, we can use the *Cut* or *Delete* option of the menu. A new row can be inserted using the *Insert Rows* option. We can copy the field name and its data type using the *Copy* option. Lastly we can make the selected field a primary key by selecting the *Primary Key* option. Observe that the Primary Key option in the screen is preceded by a ✓ sign. This indicates that the field we are trying to work on is at present the primary key of the table.

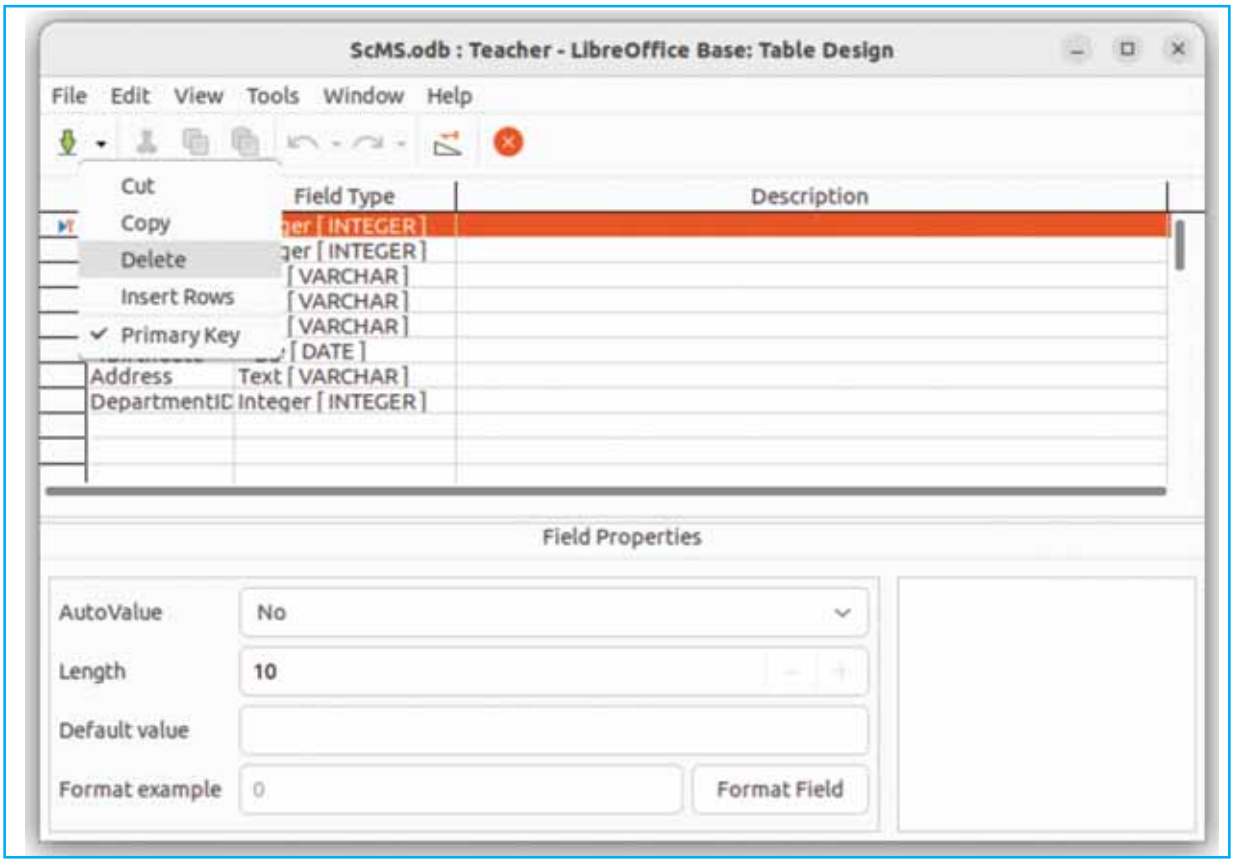

Figure 2.14 : Editing the Table Structure

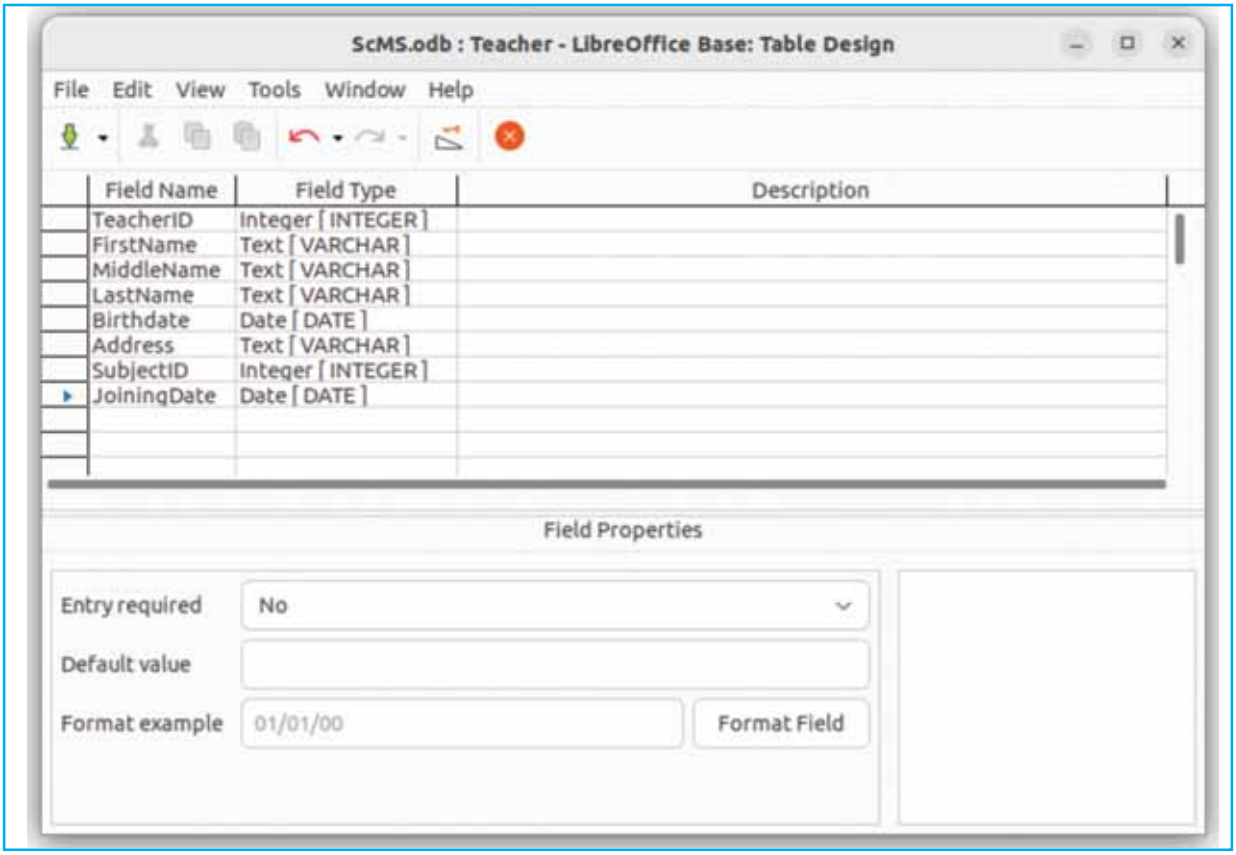

Figure 2.15 : Update Table Structure

First select the *Delete* option from the drop down menu and the field ID will be deleted and the EmployeeID field will be shown as the current record. To update the field name simply double click on the name and type the new name. Thus rename the field EmployeeID to TeacherID and DepartmentID to SubjectID. We will also add one new field JoiningDate here to show the change in the table structure. Save the changes, if we get a warning for deleting the primary key, we will ignore it for now. The screen will now look like the one shown in figure 2.15.

Create the remaining tables that we had designed in Chapter 1 using the design view so that we now have five tables in the database. We are now ready to enter data in the tables that we have created. But before we do that, let us look in the Description and the Field Properties given in the table design view.

Setting Description and Field Properties

Entering a description for each field is not mandatory but it is always a good practice to add a description as it assists in documentation purposes. It helps the user or anyone who is trying to use the database to understand what is the purpose of each field.

The *Field Properties* option when used allows us to control and validate the data that is to be entered. It also identifies how the values in the field are stored and displayed. For example, we may decide in what formats like DD-MM-YY or MM-DD-YY the user will enter the data in the BirthDate field, how it should be displayed while printing or viewing the data and what message is to be given to the user in case of invalid date entry.

A set of different field properties is displayed related to the datatype that the user selects. It is possible to change all the field properties as per our requirement. Some of the common field properties are discussed in this section:

Default value: We can specify a value that will be stored by default in a field. Once we set this property for any field, the specified default value will automatically be displayed when we add a new record in our table. A user may change the value if required at the time of data entry.

AutoValue: This property is used with numeric fields, usually the one that is assigned as the primary key. For example, the Teacher table has TeacherID set as primary key and its datatype is integer. We expect values like 1,2,3… and so on to be stored in the TeacherID field. Thus we can enable the AutoValue property for this field. Let us set the AutoValue property for the TeacherID field in the Teacher table. Open the Teacher table in the design view, select the TeacherID field, we will be able to see the *Field Properties* as shown in figure 2.16. Select "Yes" from the drop down menu visible succeeding the *AutoValue* label.

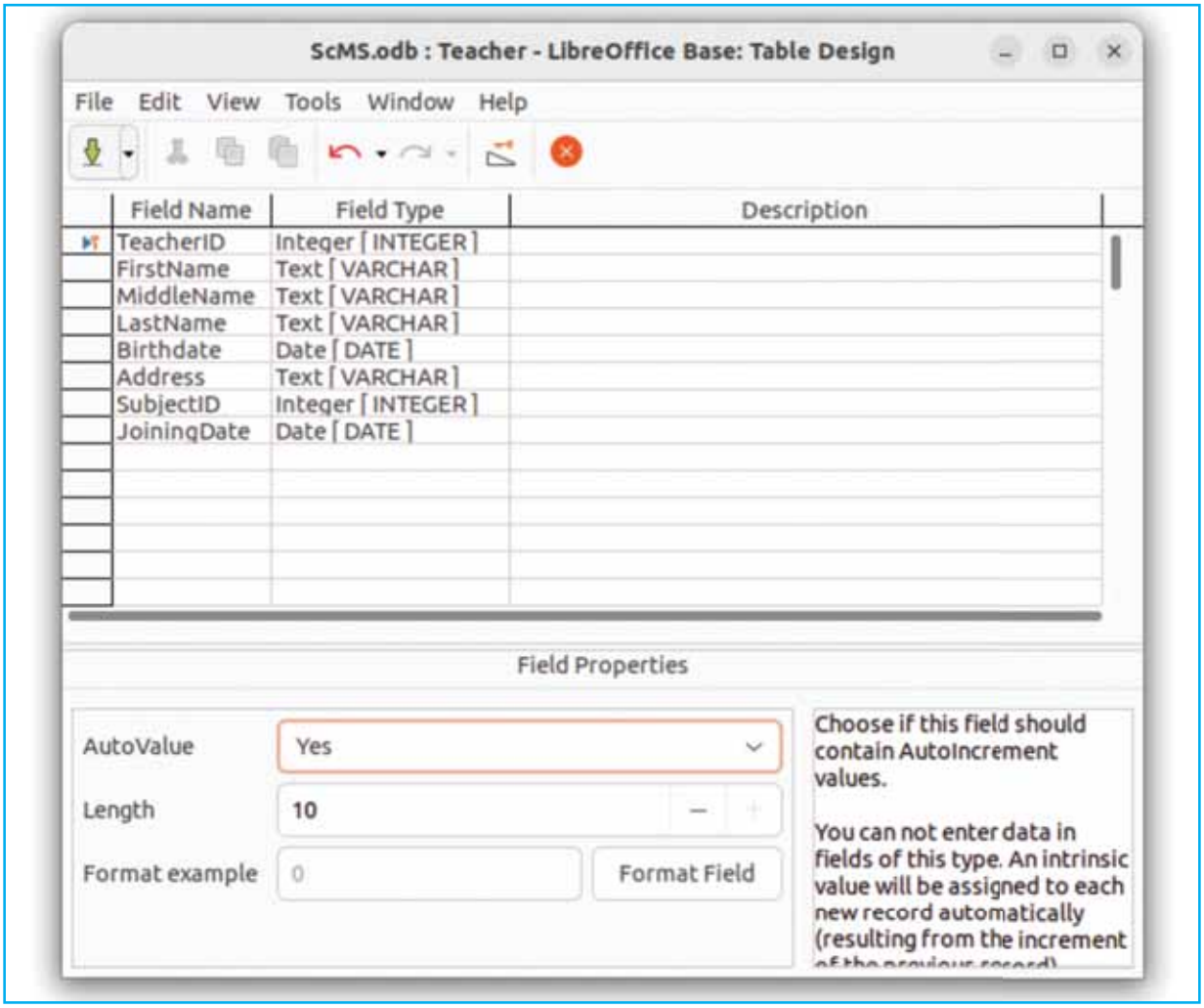

Figure 2.16 : Setting Field Properties

Length: The length property is associated with the text data type. It is automatically assigned a value for example 50 in case of Text [VARCHAR] datatype. We can assign the size of length property based on our needs to store different types of data. LibreOffice Base automatically assigns the predefined length size to different datatypes like Numeric, Date/Time, Yes/No and Memo, in such cases this property will be disabled on the screen. Observe that the value 10 is specified in figure 2.16 for length. As the field in discussion is numeric we will not be able to update it.

Entry Required: At times the data requirement in some of the fields that we use in the table is optional or mandatory. The entry required property allows us to manage such fields. To ensure that the data should not be left blank the field property needs to be set to "Yes". The default selection is "No". In the case of the Teacher table, FirstName, MiddleName and LastName fields may require this property to be set to "Yes".

Format: This property specifies a format of text, numbers or date that is used at the time of entry, display and print. The format property uses different settings for different data types. LibreOffice Base provides some predefined formats for Number, Date/Time, and Yes/No data types. Let us set the format for the joining date field of the teacher table. Open the table in the design view, select the JoiningDate field, click on the *Format Field* button similar to the one visible in figure 2.16. This will open a *Field Format* dialog box as shown in figure 2.17.

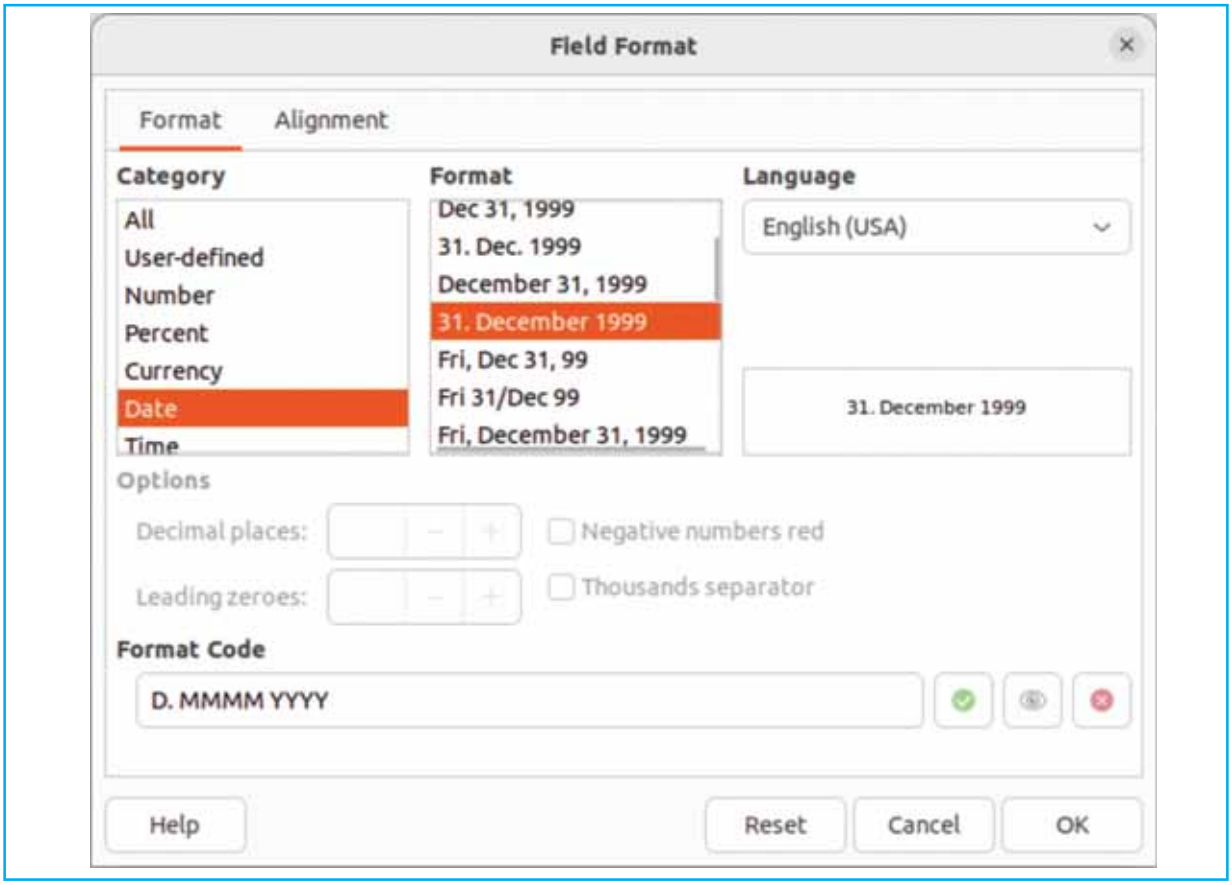

Figure 2.17 : Field Format Dialog Box

Observe that Date is selected under the *Category* label, select the option *31. December 1999* under the *Format* label. Click on the *OK* button to apply this format, LibreOffice Base will now automatically convert any date entered into D.MMMM YYYY format.

Operations on Table Data

Once the table structure is created and finalized, the next step is to insert the required data into the tables. Let us look at the different operations that can be performed on tables.

Insert Data

To insert records into the table, first we need to open the required table. To open the table go to the Tables Pane in ScMS database window and double click on the name of the table or right click on the desired table and click the *Open...* option from the menu. This will open the selected table in the Datasheet view as shown in figure 2.18. Note that if the primary key is not set we might not be able to enter the data.

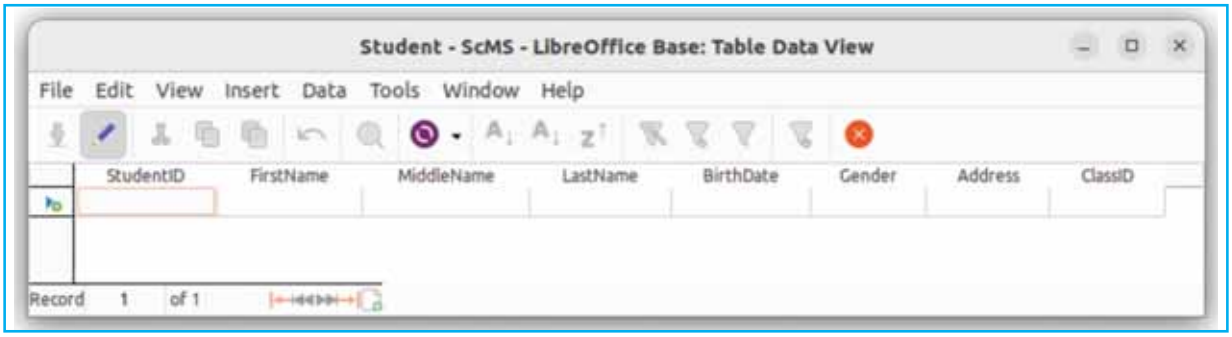

Figure 2.18 : Table Datasheet View

Observe that in figure 2.18, the field names are shown in a horizontal line known as Title line. Below the Title Line, there is a row consisting of empty boxes. The text box under the StudentID field is selected and has a blue arrow with a green + sign preceding it. The blue arrow is referred to as a *Record Selector* icon. It shows the current record that we are working on. Enter the data as shown in figure 2.19 so that we have five records in the Student table as of now.

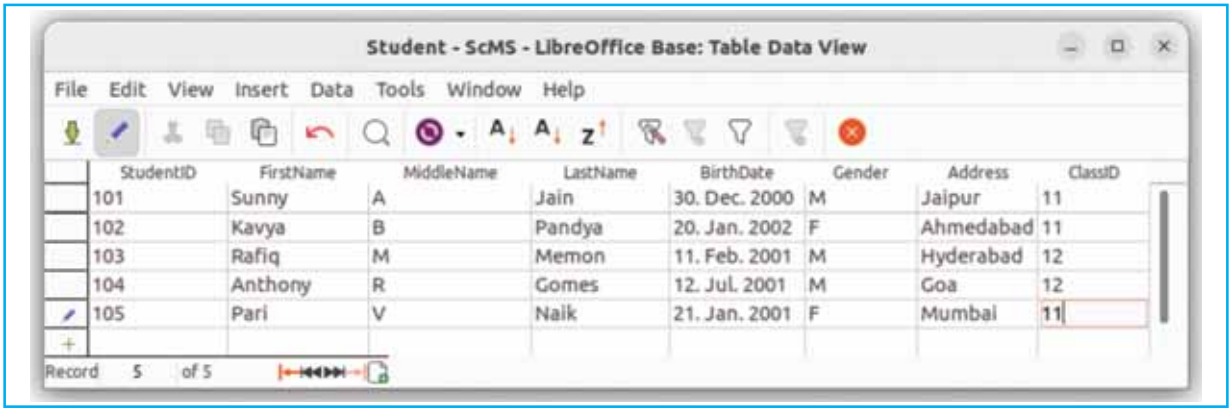

Figure 2.19 : Data of Student Table

While entering the data observe that the blue triangle changes to the pencil icon when we are entering data into a row. Also the last empty row has a green + sign visible at the beginning. To add a new record in the table we need to scroll to the last row and then click in any of its fields. Our cursor will now be positioned in the selected field and the icon will be changed to the blue arrow and green plus sign.

The bottom most part of the figure 2.18 is known as the Navigation Bar. It shows the number of records in the table as well as is capable of showing the position of any selected record. It also contains navigation buttons that allow us to scroll the records vertically. Observe that in figure 2.19 at the left bottom of the screen we are able to see 'Record 5 of 5' indicating that there are a total 5 rows in the table and currently we are at row number 5.

Edit Data

Updation or editing of data is a continuous process. The data might be updated for several reasons like wrong data entry or change in data due to genuine reasons. For example, what if Sunny shifts to Ahmedabad from Jaipur. In this case though the address of Sunny was correctly entered, we still need to edit it.

The process of correcting the data already entered in the table is usually known as Editing. To edit any data we need to open the table in the datasheet view, go to the desired field and place the cursor at the field value that we want to edit. We can thus make any changes that are needed.

Delete Data

To keep our database clean any unnecessary data or records in the table needs to be removed.

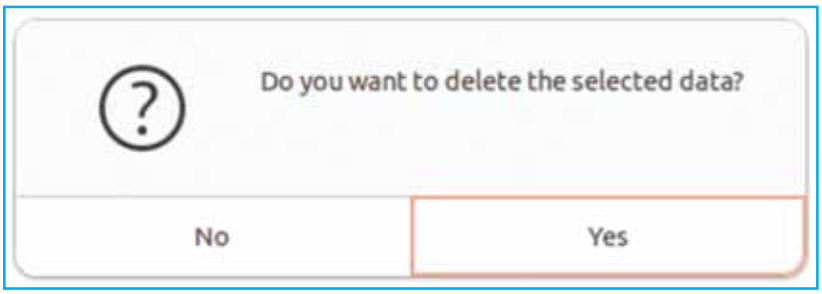

Figure 2.20 : Delete Alert Box

To delete a record from the table, open the desired table in the datasheet view, select the record or records (multiple records can be selected by selecting starting record, pressing a Shift key and selecting the last record).

The simplest way to delete the records is to press the Delete key on the keyboard. Alternatively we can select the *Delete Record* option from the *Edit* menu or right click the selected records and select the *Delete Rows* option from the sub menu. In all cases, we will be presented with a alert dialog box as shown in figure 2.20.

Click on the *Yes* button and the record will be deleted from the table. If the user selects *No* button then no action will be taken on the records.

Sort Data

The objective of storing data in a table is to be able to access it as and when needed. The data within a table should be organized in a specific way such that it is easy to obtain the desired information. As the number of records in the table grows, finding things can become a bit difficult. Sorting the data on a specific field is one way for arranging data properly. Assume that we wanted to know the students who reside in Ahmedabad by looking at the data in the Student table. We would be able to get this answer easily if the data was sorted on the Address field. Let us sort the data of the Student table.

We can sort the table in multiple ways, the simplest is to use the sort icons  shown in the datasheet view toolbar. This sort icon is similar to the *Sort...*, *Sort Ascending* or *Sort Descending* options of the *Data* menu.

The *Sort Ascending* or *Sort Descending* option is used for immediate sorting from the point of cursor. For example, if you are on the fourth record of a field, say MiddleName and we use this option, then the entire data of the table will be sorted ascendingly or descendingly based on the MiddleName field.

The *Sort...* option is an advanced sort option, it allows us to select multiple fields that we would like to sort the data on. When we use this option a Sort Order dialog box as shown in figure 2.21 is shown to the user.

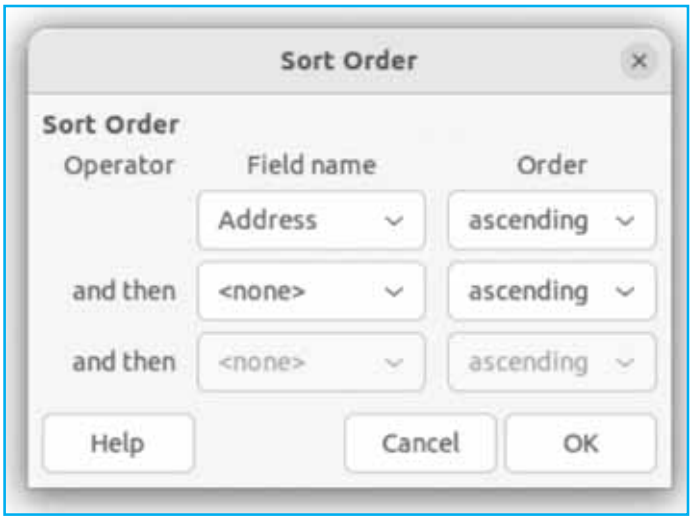

Figure 2.21 : Sort Order Dialog Box

The user now needs to choose appropriate field names from the dropdown menu under the *Field name* label. Observe in this case, we have chosen the field name Address. We can sort on a combination of a maximum of three fields. The sorted table is as shown in figure 2.22.

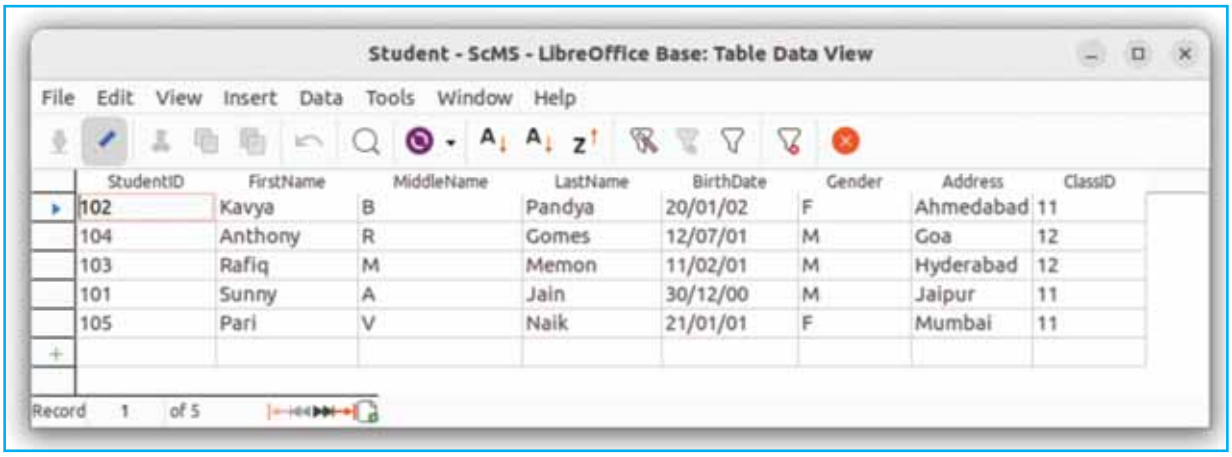

Figure 2.22 : Table Data Sorted on Address Field

Redundancy and Keys

Many times we may observe that the same piece of data is stored in more than one place, this is known as redundancy. It leads to inefficiency, inconsistency, and increased storage costs of a database. For example, the Grade table that we have created needs information about the student and subject both. If we had stored the name of the student as well as subject in it then we may have repeated the student and subject name multiple times leading to redundancy.

Redundancy is undesirable and is usually removed through a concept called database normalization. Here we split the data into related tables so each piece of information is stored only once. The primary key and foreign key discussed in Chapter 1 plays a very significant role in the normalization process, as it allows decomposition of tables into multiple tables and relating them.

We may apply other methods like using master data or automated data validation to control redundancy. In the case of the ScMS database the tables Student, Teacher, Subject and Class can be considered as master data, and are used as a single source of truth for common information about these entities. The Grade table here is a transaction table as it contains data that might be considered as temporary.

Creating Relationship

The tables that we have created so far in ScMS database are not related. We have simply used certain fields that are common. For example, the Student table has a field ClassID that is also part of the Class table, similarly the Teacher table has a field SubjectID that is also part of the Subject table. Having a common field name does not guarantee that the data entered in them will always be correct. What if a user enters a ClassID 21 in the Student table but it is not there in the Class table, in such a case the database would be termed as inconsistent. Establishing appropriate relationships among the different tables ensures that such inconsistencies do not happen.

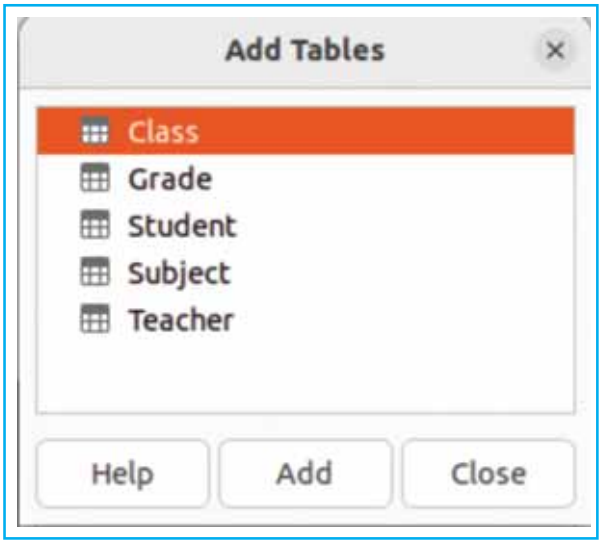

Figure 2.23 : Add Table Dialog Box

Let us now learn to establish relationships amongst different tables that we have created. Open the ScMS window, go to the *Tools* menu, select the *Relationships*... option from it. This will open a Add Tables dialog box with a list of tables created as shown in figure 2.23.

Select the tables one by one by double clicking on them or select one table at a time and click on *Add* button, we will see that the selected table along with its fields is displayed in a *Relation Design* window in the background. Once all tables are selected close the *Add Tables* dialog box. The *Relation Design* window will now look somewhat similar to figure 2.24.

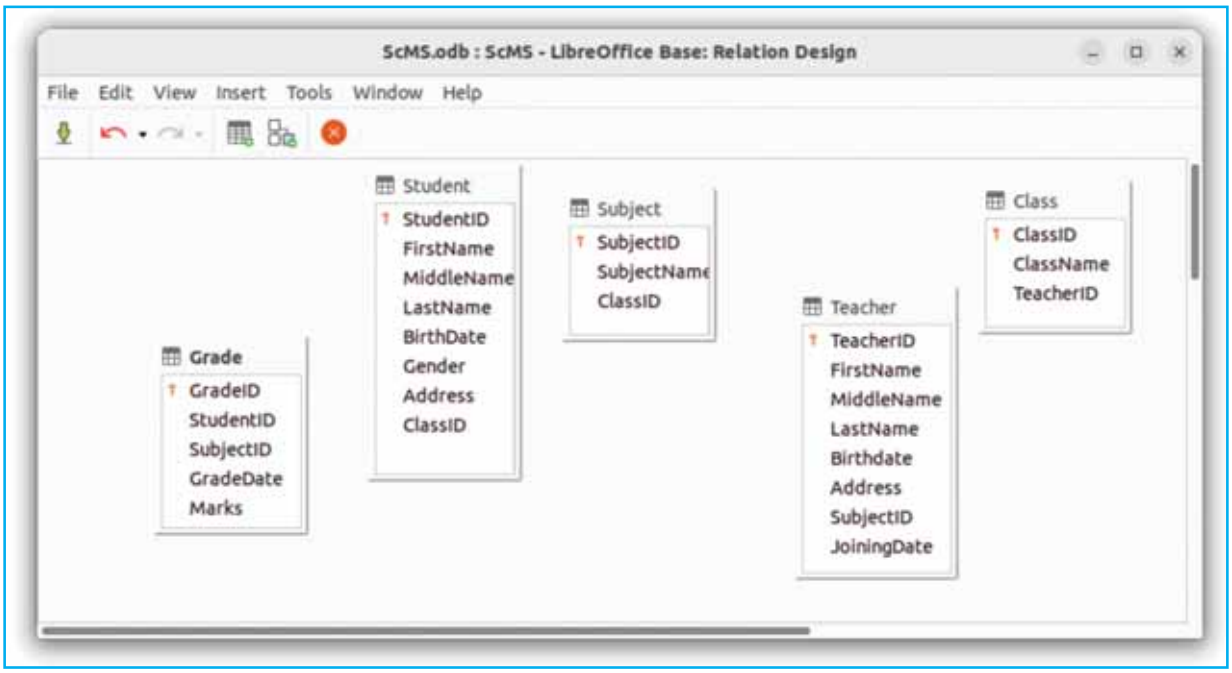

Figure 2.24 : Relation Design Window

It is possible to rearrange the positions of the table visible in figure 2.24 by using drag and drop operation. Also ensure that the primary key is set in all the tables otherwise we may get errors while creating relationships.

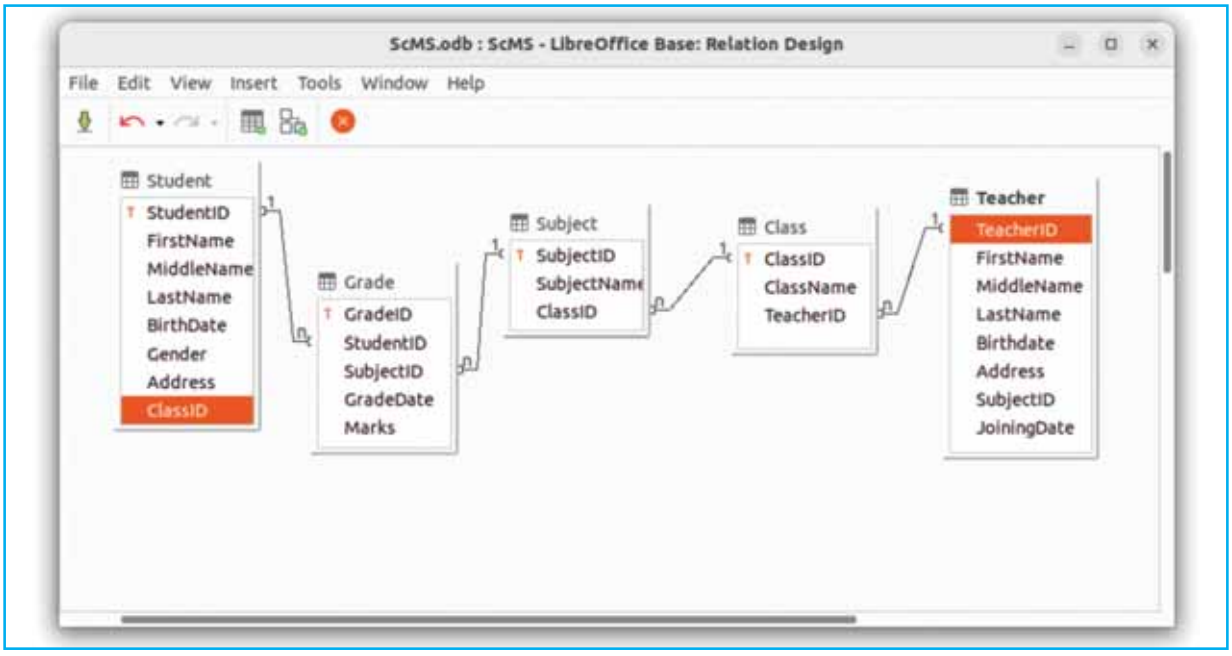

Figure 2.25 : Relation Design Window after Sample Relationship Setup

Finally to create a relationship, we will use the drag and drop operation. First click on the StudentID field of the Student table, keep the left mouse key pressed and drag the mouse on the StudentID field of the Grade table and release the left mouse key. We will see a connected line created between the two fields with labels 1 and *n*. Observe that label 1 is displayed on the StudentID (primary key) of the Student table and label *n* is displayed on StudentID (foreign key) of the Grade table. It indicates that the StudentID field in the Student table will hold unique values while the StudentID field in the Grade table may have repeated (*n*) values .

Note that when we are creating a relationship the data types of both the fields being related must be the same. Create all the relations possible and save them, the *Relation Design* window will now look somewhat similar to figure 2.25.

Once the relationship between tables is created, we can work on the data integrity constraints of the database. We are aware that the Student table and Grade table in ScMS database are related to each other. This relation states that the StudentID entered in the Grade table should be first available in the Student table. Records pertaining to students in the Student table are considered as master or parent records, while records pertaining to students in the Grade table are considered as transaction or child records.

Referential Integrity: Assume a scenario when a user deletes a parent record from the Student table. What should happen to its related child record in transaction tables? For example, if there is a record that has StudentID as 101 in the Student table and corresponding multiple records of this StudentID in the Grade table. What will happen to the records of StudentID with value 101 in the Grade table if the user deletes or updates StudentID with value 101 from the master table?

The referential integrity rule says that there must be no entry of data in the transaction table without related data in the master table. Thus there should be no unmatched foreign key values in the database.

Referential integrity needs to be enforced in a database while performing update or delete operations on records. Let us now try and enforce the referential integrity rules on the Student and Grade tables. Double click on the relationship line seen between Student and Grade table in figure 2.25. This will open a Relations dialog box as shown in figure 2.26.

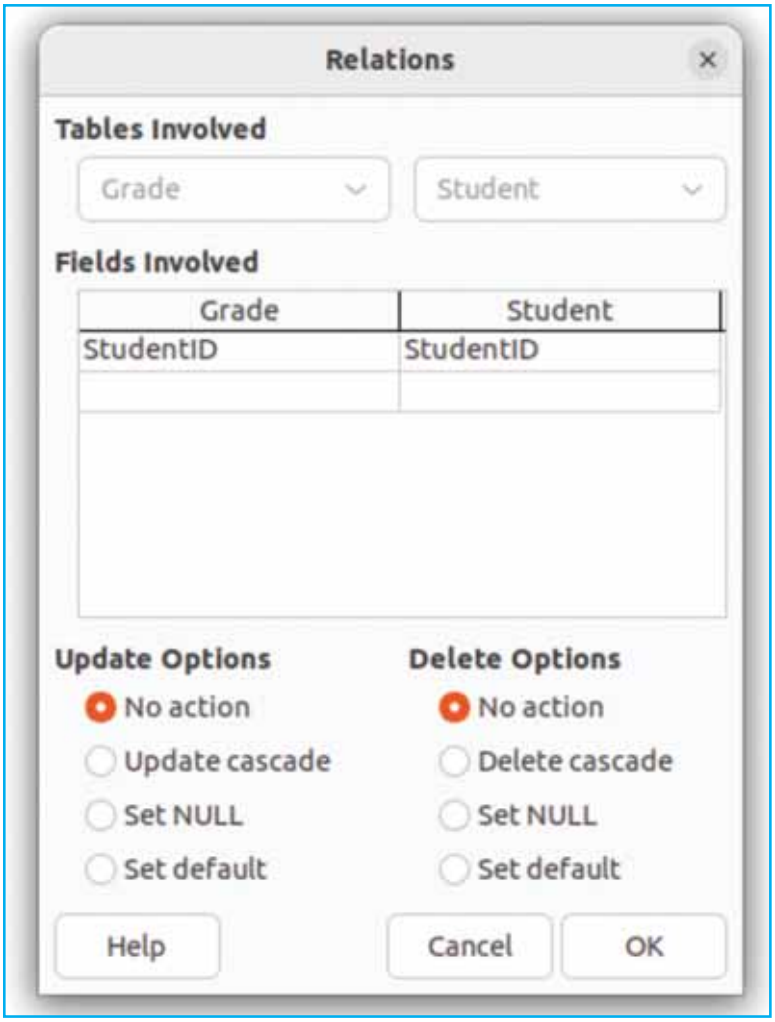

Figure 2.26 : Relations Dialog Box to set Referential Integrity

A database designer can now choose and select one of the four options seen in the figure 2.26 based on the transaction needs of the organization. We can select different option values for update and delete operations. For example, we may select *No action* option under the *Update Options* label and *Delete cascade* option under the *Delete Options* label. Let us see what each of these options are meant for.

No action: If this option is selected, the user will not be allowed to either delete or update the record if its related record exists in some other table. For example, if we try to delete a record with StudentID having value 101 in the Student table we will not be allowed to do so as corresponding records exist in the Grade table.

Update cascade: This option assigns an updated value to all the related fields if the master record is updated. For example, if we update a record with StudentID having value 101 to 125 in the Student table, then all the corresponding StudentID records in the Grade table will be assigned value 125.

Delete cascade: This option deletes all the related fields if the master record is deleted. For example, if we delete a record with StudentID having value 101 from the Student table, then all the corresponding StudentID records in the Grade table will also be deleted.

Set NULL: This option assigns a NULL value to all the related fields if the master record is deleted or updated. For example, if we delete a record with StudentID having value 101 in the Student table, then all the corresponding StudentID records in the Grade table will be assigned a NULL value.

Set default: This option assigns any fixed default value to all the related fields if the master record is deleted or updated. For example, if we delete a record with StudentID having value 101 in the Student table, then all the corresponding StudentID records in the Grade table will be assigned a default value.

We can delete or edit the relationship that has been established between the tables if required. To do so, open the *Relationship Design* window, select the desired relationship line visible between the tables, right click on it, a popup menu with *Delete* and *Edit...* option will appear. Perform the operation required and save the relationship again.

We are now ready with a database that can be used to store and analyze data. Enter proper records in all the tables designed so that we can generate information out of it using queries.

Summary

In this chapter, we learned how to use LibreOffice Base, a tool for creating a database. It provides four objects namely Tables, Queries, Forms and Reports. We saw how to create tables in two ways; using a Wizard and custom design. A wizard is a guided, step-by-step graphical interface tool that simplifies complex tasks by prompting users for information and automates the process of performing the task. We also learned about the Field Properties option, which can be used to control and validate the data that is to be entered in the records of the table. Finally we learned how to set relationships amongst tables that allows us to control the data redundancy and also establishes referential integrity. We are now ready to create tables and perform different operations on them. The next chapter will teach us how to get the required information from the tables using queries.

EXERCISE

1. Explain the term Database. What is the use of a Database?
2. What is a table? Explain the different ways in which a table can be created.
3. What is the use of Wizard?
4. Explain the significance of relations between tables.
5. Explain the significance of the labels 1 and *n* in table relationships.
6. Explain the concept of data redundancy giving an appropriate example.
7. What is the use of Normalization?
8. State why Referential Integrity is required in the database.
9. What does a field property signify?
10. Explain the importance of Format field property.

11. State whether true or false?

(1) LibreOffice Base allows us to create text documents.

(2) The Authors table is provided under the business category tables of LibreOffice Base.

(3) Customized tables are created using Table Wizard.

(4) The Field Properties option allows us to control and validate the data that is to be entered.

(5) The Record Selector icon shows the current record of the table we are editing.

12. Fill-in the blanks.

(1) LibreOffice Base assigns ..... extension by default to the database created in it.

(2) The Table Wizard provides templates of two categories of tables, ... and Personal.

(3) In the table design view we can see field name, type, ... and properties.

(4) The AutoValue property is used with ..... fields.

(5) Redundancy is removed through a concept called .....

13. Multi-choice questions. Choose the most correct answer.

(1) LibreOffice Base allows us to create databases in how many of the following ways?

(a) 1 (b) 2 (c) 3 (d) 4

(2) Which of the following objects is not visible when we open the LibreOffice Base database window?

(a) Table (b) Queries (c) Forms (d) Views

(3) Which of the following operations can be performed on the records in the database to arrange them in the proper way?

(a) Insert (b) Delete (c) Update (d) Sort

(4) Which of the following best describes data redundancy in a database?

(a) Decomposition of tables (b) Relating tables

(c) Repetition of data (d) Deletion of data

(5) Which of the following statements is not correct?

(a) Names of Primary key and Foreign key should be the same.

(b) Every table must have a primary key.

(c) A primary key can consist of multiple fields.

(d) The data type of primary key and foreign key should be the same.

(6) Which of the following labels would be visible on the relationship line if we relate the StudentID field of Student and Grade table?

(a) n…n (b) n…1 (c) 1…n (d) No labels

(7) While setting referential integrity between tables, which of the following will not allow the user to either delete or update the record if its related record exists in some other table?

(a) Set default (b) Set NULL (c) No action (d) Delete cascade

(8) Which of the following field properties can be considered as equivalent to Not NULL?

(a) Default (b) Format (c) Length (d) Required

(9) In which of the following, the data that is used as a single source of truth for common information about entities is stored?

(a) Transaction Table (b) Master Table

(c) Big Table (d) Accounts Table

(10) Which of the following is the benefit of using foreign keys?

(a) Data consistency (b) Faster data retrieval

(c) Improved data redundancy (d) Reduced storage space

Laboratory Exercise

1. For the databases listed in following table, perform the following operations:

(a) Create tables.

(b) Identify appropriate primary key and foreign key for each table.

(c) Decide what data type will be suitable for each attribute in a table.

(d) Establish relationships amongst the tables.

(e) Insert at least ten records in each table.

DB 1	Student (StudentId, StudentName, Address, City, BirthDate, ContactNo, Email) Book (BookId, BookTitle, Description, BookAuthor, Status, PublishYear) Book_Issue (BookIssueId, BookId, StudentId, IssueDate, ReturnDate, FineAmount)
DB 2	Product (ProductId, ProductName, Quantity, ProductPrice, ManufactureYear) Salesman (SalesmanId, SalesmanName, Address, BirthDate, ContactNo, Gender) SalesOrder (SalesId, SalesmanId, ProductId, QtySold)

DB 3	Customer (CustomerId, CustomerName, Gender, Address, City, Area, Email, ContactNo) Magazine (MagazineId, MagazineName, UnitRate, PublisherName, PublishedMonth) Subscription (CustomerId, MagazineId, StartDate, EndDate)
DB 4	Employee (EmployeeId, EmployeeName, Address, City, Salary, DesignationId) Designation (DesignationId, DesignationName, BasicSalary) Project (ProjectId, ProjectName, StartDate, ProjectPrice) ProjectWorkHrs(PId, ProjectId, EmployeeId, HoursWorked)
DB 5	Vehicle (VehicleId, VehicleType, Price, Description, ManufactureDate) Customer (CustomerId, CustomerName, Address, BirthDate, ContactNo) VehicleOwner (VehicleId, CustomerId, PurchaseDate, DeliveryDate)