11 Retrieving Data Using Queries

In the previous chapter, we have mentioned that database is a huge collection of data. Now, think of a situation where we want to retrieve information from a database. Now suppose we want to send information about Independence Day offers to our customers, we would need list of our customers along with their addresses. Assuming that this information is stored in a table named Customer, the details can be obtained by opening this table. Query is the feature of Base, which can be used to retrieve specific set of information from database. The user can retrieve data according to his choice, criteria and format irrespective of how it is stored in the database. We can ask questions like "Give me details of customer who has purchased products worth more than one lakh rupees in the current year" or "Give me details of products that have not been sold even once". One of the key reasons to use Database Management System is its ability to design, save and use the queries as and when required.

Defining Query

Query basically means asking question, doing inquiry or performing analysis. In Base, to query is to ask a question about the information in the database. Through a query one can tell Base to display exactly which fields and records a person would like to view from the database. It is a set of rules for fetching information from a table, or from several tables at once. The result of a query is itself in the form of a table. It consists of a set of records, organized in rows (one per record) and columns (one per field).

To create a query, open a database and click on the icon labeled as Queries in the left hand pane. The query window is organized like the other windows in Base. Different ways to create a query will be shown in the Tasks pane displayed at the top. If you have already created some queries then it will be listed under the pane titled Queries. Figure 11.1 shows the window when Queries icon is selected.

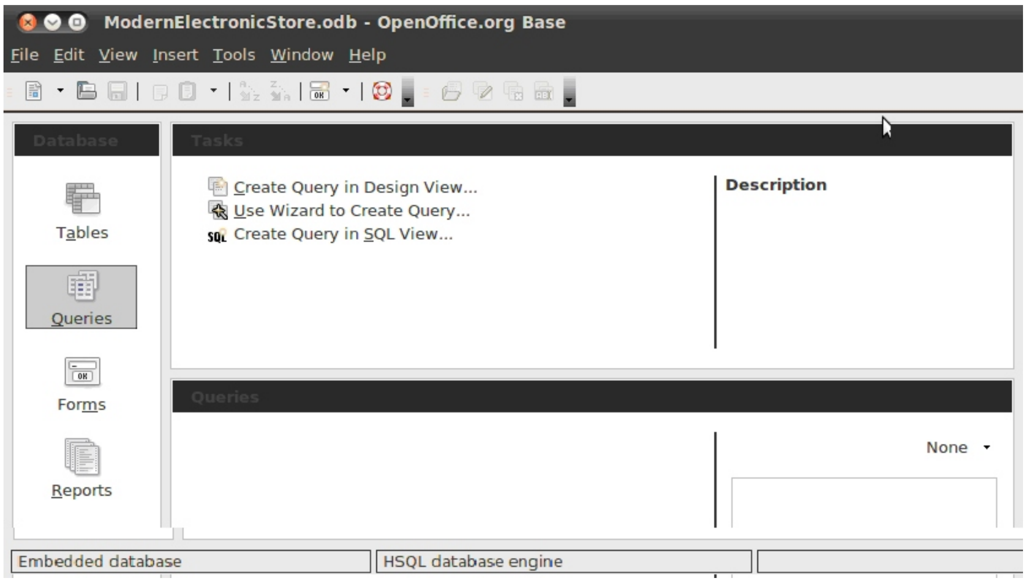

ModernElectronicStore.odb - OpenOffice.org Base
File Edit View Insert Tools Window Help
Database
Tables
Queries
Forms
Reports
Tasks
Create Query in Design View...
Use Wizard to Create Query...
Create Query in SQL View...
Description
Queries
None
Embedded database
HSQL database engine

Figure 11.1 : Queries Window

As you can see in figure 11.1 under the Tasks pane, Base offers three different ways to create a query.

- Create Query in Design View...: This view provides a minimal amount of guidance.
- Use Wizard to Create Query... : Wizard guides us through the process of query creation.
- Create Query in SQL View... : It provides no guidance at all. Users need to have knowledge about fourth generation computer language called Structured Query Language.

If you are database expert, you might prefer Design or SQL View. However, Base's Query Wizard has the advantage of helping us organize our thoughts while requiring no previous knowledge.

Let us discuss each of these options one by one. We will create a query that gives us list of customer names and addresses.

Query Creation Using Wizard

- Double click the option *Use Wizard to Create Query....* A Query Wizard dialog box as shown in figure 11.2 will be displayed. Notice the left side of dialog box. Eight steps to create a query have been given. Only first step, Field selection is compulsory. It helps us in identifying the fields that are to be displayed in our output. Other steps allow us to format the output and can be skipped if not required.

- The first step in creating the query is to select the table and the set of fields in that table from which information is to be retrieved. We may select all or some of the fields visible in the *Available fields* list box. You can use the left and right arrow buttons to move the fields from *Available fields* list box to *Field in the Query:* list box. The fields that you finally select to be part of your query will be listed under *Field in the Query:* list box. These fields can then be arranged in the order required using the up and down buttons. Once the fields are finalized click on Next button. Observe that in figure 11.2 we have selected table Customer and are able to see all the fields related to it under *Available fields* list box.

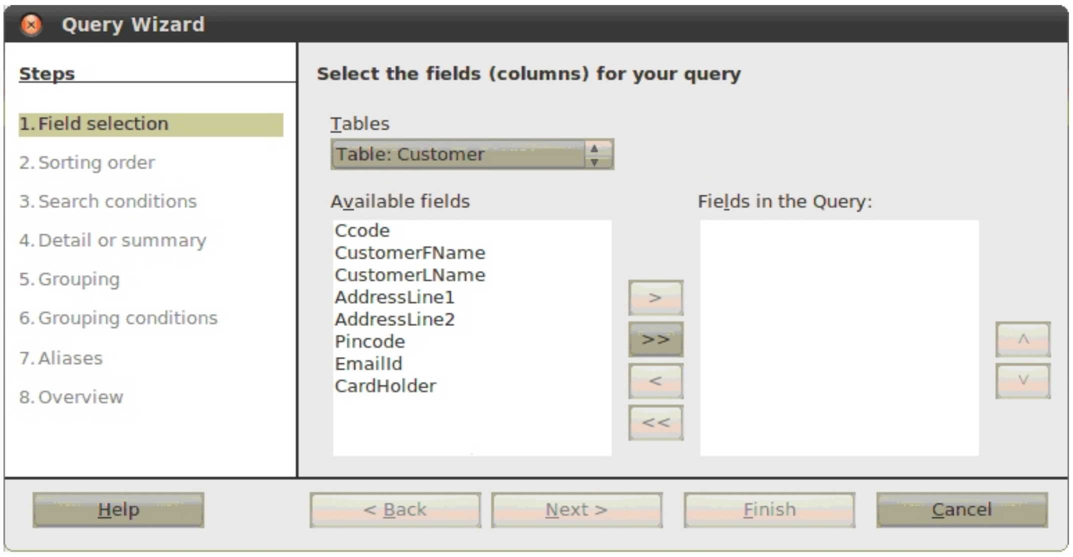
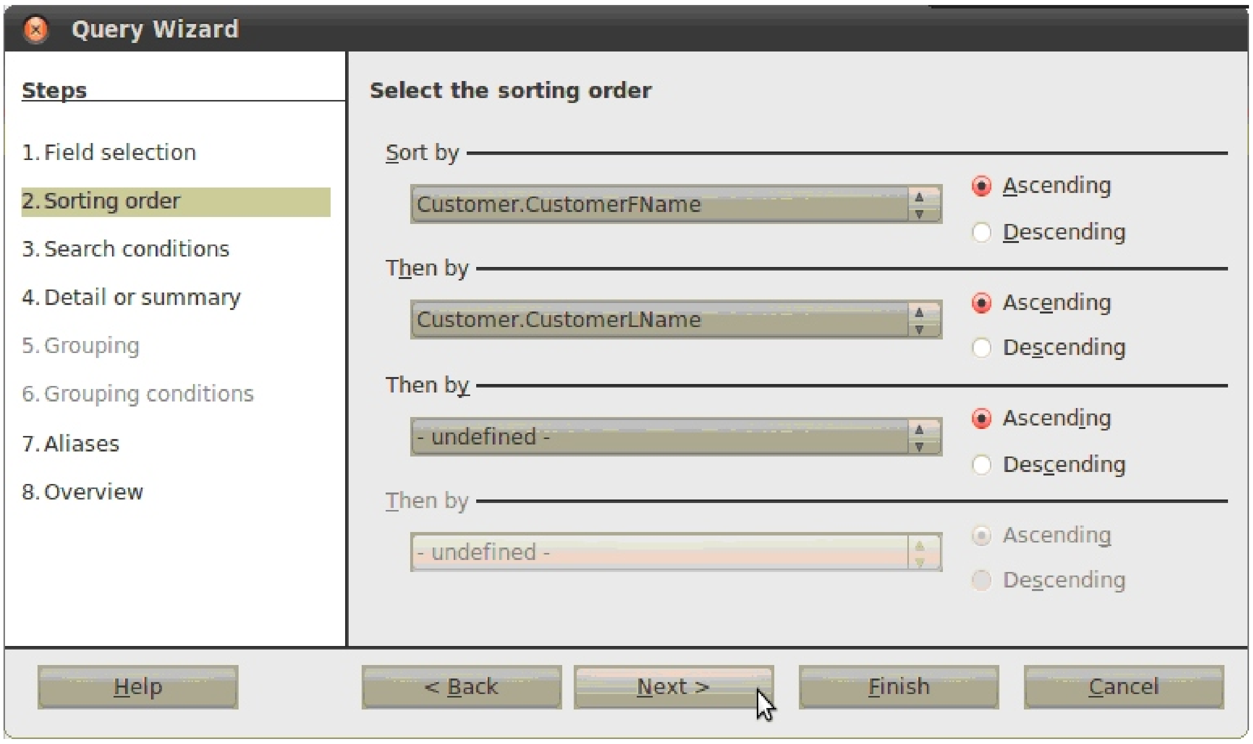


Figure 11.2 : Selection of table and fields for query

- The second step is to mention the sort order in which output of the query will be displayed. It allows you to select up to four fields for deciding the sorting order of the output. For example, we might want to display the query result sorted in order of first name initially, and then by last name of the customer. Once the sorting order is decided click on Next button. Figure 11.3 shows how to select the sorting order of fields.



Query Wizard

Select the sorting order

Sort by: Customer.CustomerFName [✓ Ascending] Descending

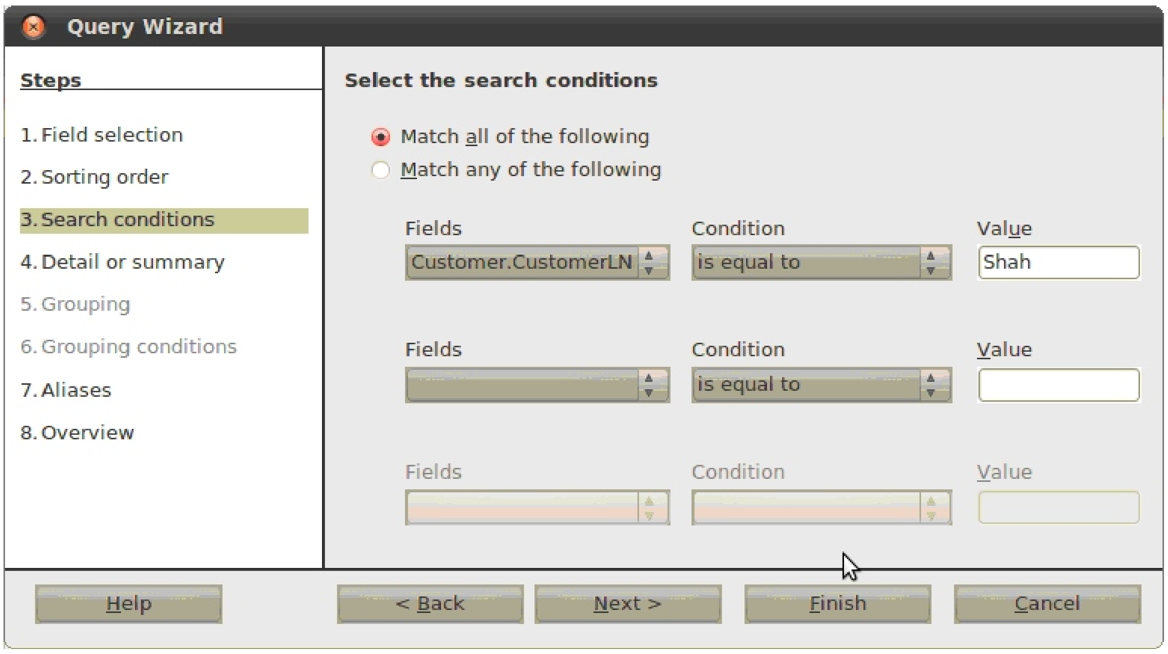
Then by: Customer.CustomerLName [✓ Ascending] Descending

Then by: - undefined - [✓ Ascending] Descending

Then by: - undefined - Ascending Descending

Figure 11.3 : Applying sorting on field

- In the third step of the wizard, we actually set up the query. Here we have to select appropriate values for the *Field*, the *Condition*, and the *Value* parameters. We can define maximum three search conditions in one query. In the case of the address query, if you want list of customers with "Shah" as their last name, the criteria would be simple: You would select the CustomerLName field from the drop down under the label *Fields*, then select *is equal to* from the drop down under the *Condition* label and finally in the text box under the label *Value*, type *Shah*. Observe that we have two options in this step. *Match all of the following* and *Match any of the following*. Since we have only one condition, we wouldn't need to change the default setting. If multiple search conditions are to be set, like if we are looking for customers with last name as Shah or Patel then, we need to choose *Match any of the following*. Once the search conditions are finalized click on Next button. Figure 11.4 shows the search condition settings for listing of all customers with last name as Shah.


Query Wizard

Steps

1. Field selection
2. Sorting order
3. Search conditions
4. Detail or summary
5. Grouping
6. Grouping conditions
7. Aliases
8. Overview

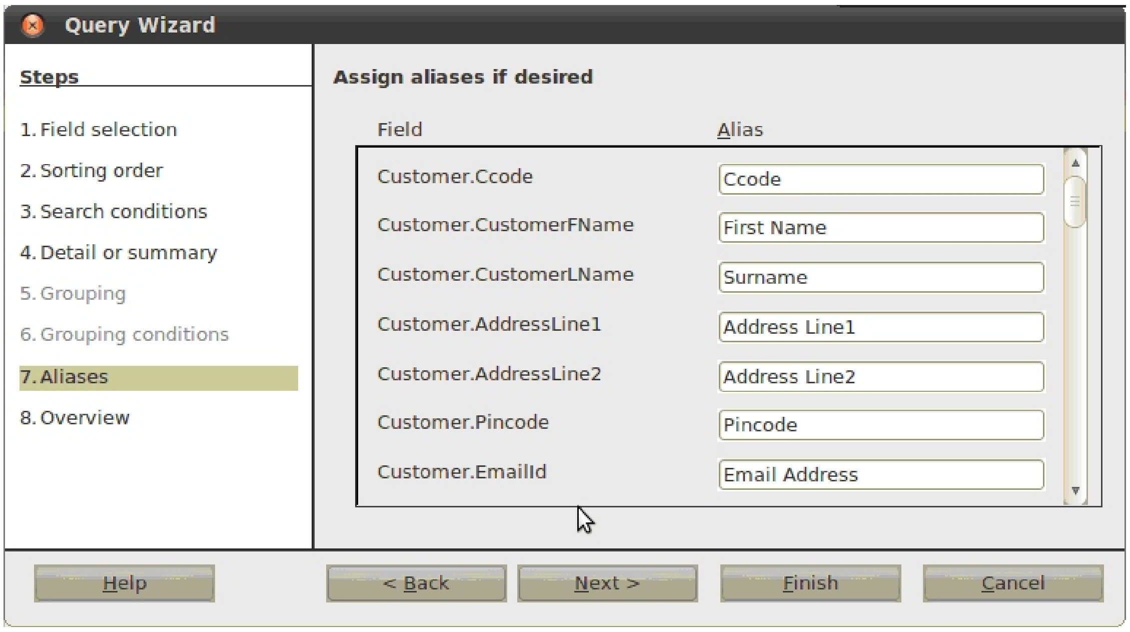
Select the search conditions

[✓ Match all of the following]
Match any of the following

Fields	Condition	Value
Customer.CustomerLN	is equal to	Shah
	is equal to	

Figure 11.4 : Applying search condition on field

- At this point Base skips three steps in its standard wizard and jumps to the seventh step. In our case, the selected Customer table does not contain any numeric field and so steps including options to summarize or perform numerical calculations are skipped.
- In the seventh step, Base expects aliases for selected field names. The purpose of this step is to make the query wizard display the field names in human readable form. It allows us to add small touches to the field name such as spaces between words and writing full forms of short field names like First Name for CustomerFName. This step is also optional, we may add aliases if required only. If we do not add aliases then field names of a table will be displayed as it is in a query. Once all aliases are decided click on Next button. Figure 11.5 shows how to add aliases.


Query Wizard

Steps

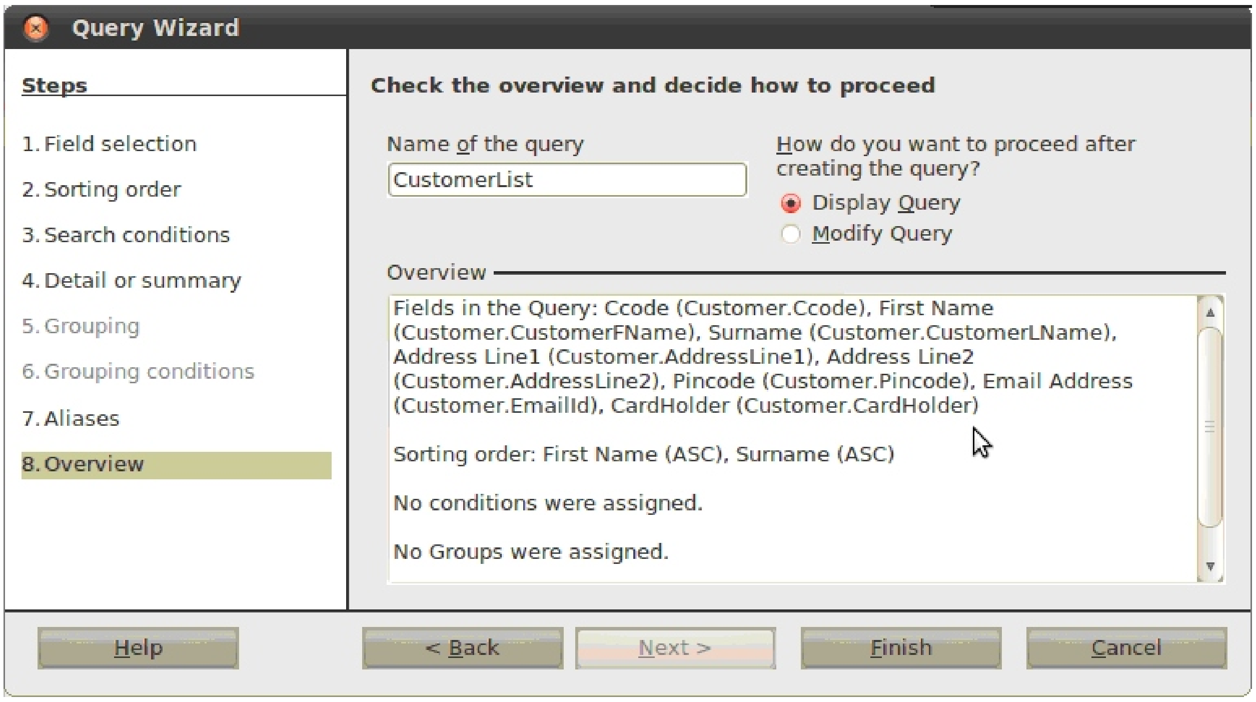
1. Field selection
2. Sorting order
3. Search conditions
4. Detail or summary
5. Grouping
6. Grouping conditions
7. Aliases
8. Overview

Assign aliases if desired

Field	Alias
Customer.Ccode	Ccode
Customer.CustomerFName	First Name
Customer.CustomerLName	Surname
Customer.AddressLine1	Address Line1
Customer.AddressLine2	Address Line2
Customer.Pincode	Pincode
Customer.EmailId	Email Address

Figure 11.5 : Adding Aliases

- Finally, in the eighth step, we are given an overview of all the steps performed till now. Figure 11.6 gives us the overview of query recently created. Assign desired name to the query by typing it in text box labeled *Name of the query,* in our case we have named it as CustomerList. Take a moment to look over what we have done. In case changes are to be made, use the Back button to make the changes desired. Once the query is created, the only way to make changes is through the Design View. Two options under the question *How do you want to proceed after creating a query?* can be noticed. We can either view the query result immediately by choosing *Display Query* option, or else open it in Design View using *Modify Query* option. Design view can be used to insert additional features in query that may be beyond the capacity of the wizard.


Query Wizard

Steps

1. Field selection
2. Sorting order
3. Search conditions
4. Detail or summary
5. Grouping
6. Grouping conditions
7. Aliases
8. Overview

Check the overview and decide how to proceed

Name of the query: CustomerList

How do you want to proceed after creating the query?
[✓ Display Query]
Modify Query

Overview

Fields in the Query: Ccode (Customer.Ccode), First Name (Customer.CustomerFName), Surname (Customer.CustomerLName), Address Line1 (Customer.AddressLine1), Address Line2 (Customer.AddressLine2), Pincode (Customer.Pincode), Email Address (Customer.EmailId), CardHolder (Customer.CardHolder)

Sorting order: First Name (ASC), Surname (ASC)

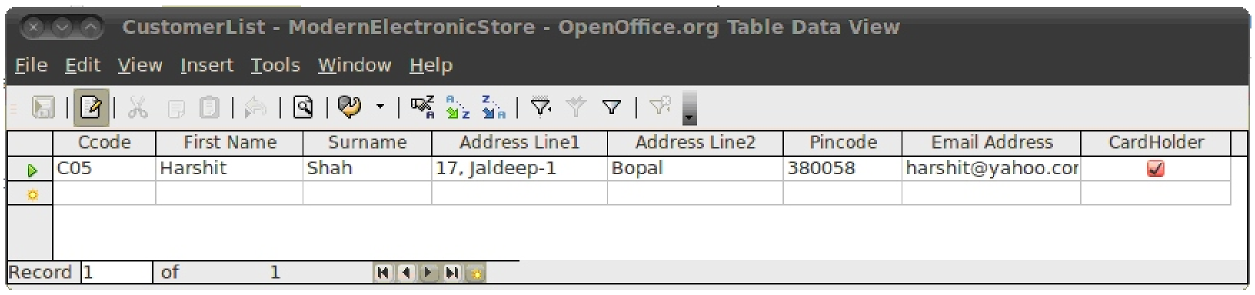
No conditions were assigned.

No Groups were assigned.

Figure 11.6 : Overview of created query

- For example, in the discussed example, where we are expecting list of customer names and addresses, we expect city name, state name and country name as well. The customer table we selected consists of only Pincode field. Details related to the Pincode field are available in other tables namely City, State and Country. However Wizard used to create a query provided us option of selecting a single table. Thus Design Views are used to create complicated queries.

- Click on Finish button and you will find that query result of the query is displayed in Data Sheet view as shown in figure 11.7.


CustomerList - ModernElectronicStore - OpenOffice.org Table Data View
File Edit View Insert Tools Window Help

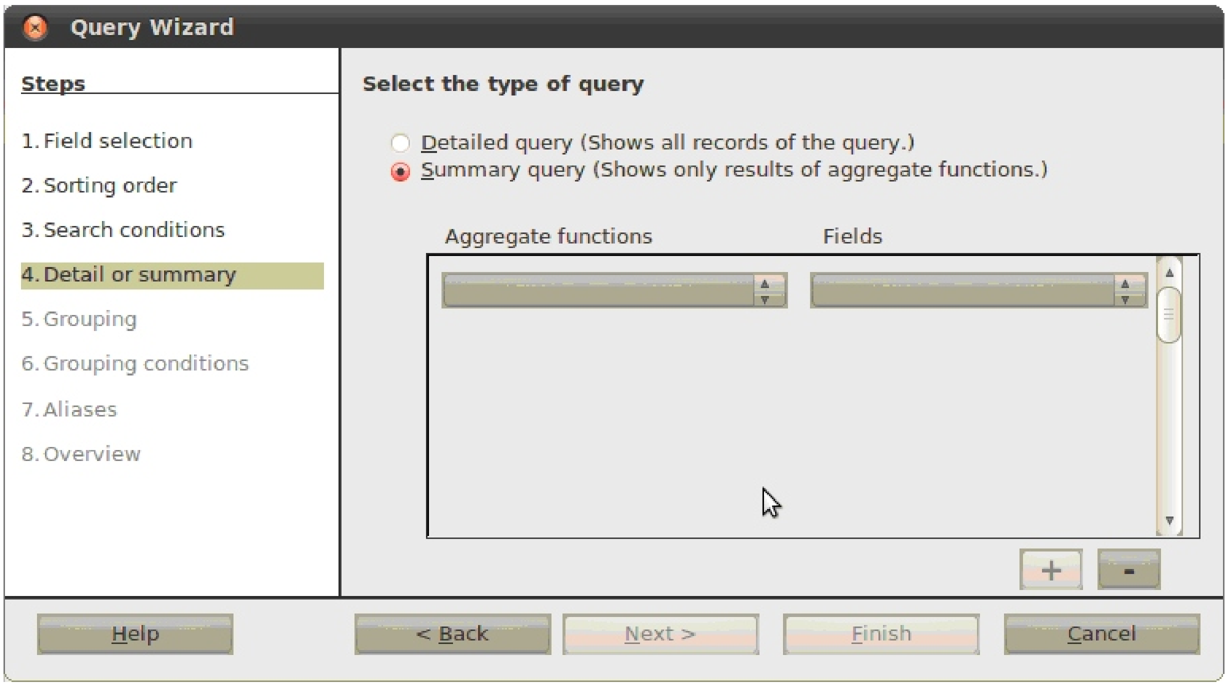
Ccode	First Name	Surname	Address Line1	Address Line2	Pincode	Email Address	CardHolder
C05	Harshit	Shah	17, Jaldeep-1	Bopal	380058	harshit@yahoo.cor	✓

Record 1 of 1

Figure 11.7 : Result of Query

Let us now calculate the amount received so far for each order. We will use query wizard once again to explore working with numeric fields. Make sure that the Queries icon is selected.

- Double click on *Use Wizard to Create Query...* option
- In step 1 of query wizard select OrderPayment table. You might have noticed that in the drop down list, the CustomerList query created recently is also included along with all the other tables. Note that Base allows us to make use of a query already created for creating another query.
- Select the OrderID and PaymentAmount fields and click on the button with greater than symbol (>) to move these fields to *Fields in the Query:* list box.
- Click on Next button.
- In step 2 select the OrderID field from the drop down box labeled as *Sort by.*
- Click on the Next button.
- As we do not want to filter records, no search criteria are to be mentioned. Hence simply skip the step 3 by clicking on the Next button. We will now see a dialog box of step 4 as shown in figure 11.8.


Query Wizard

Steps
1. Field selection
2. Sorting order
3. Search conditions
4. Detail or summary
5. Grouping
6. Grouping conditions
7. Aliases
8. Overview

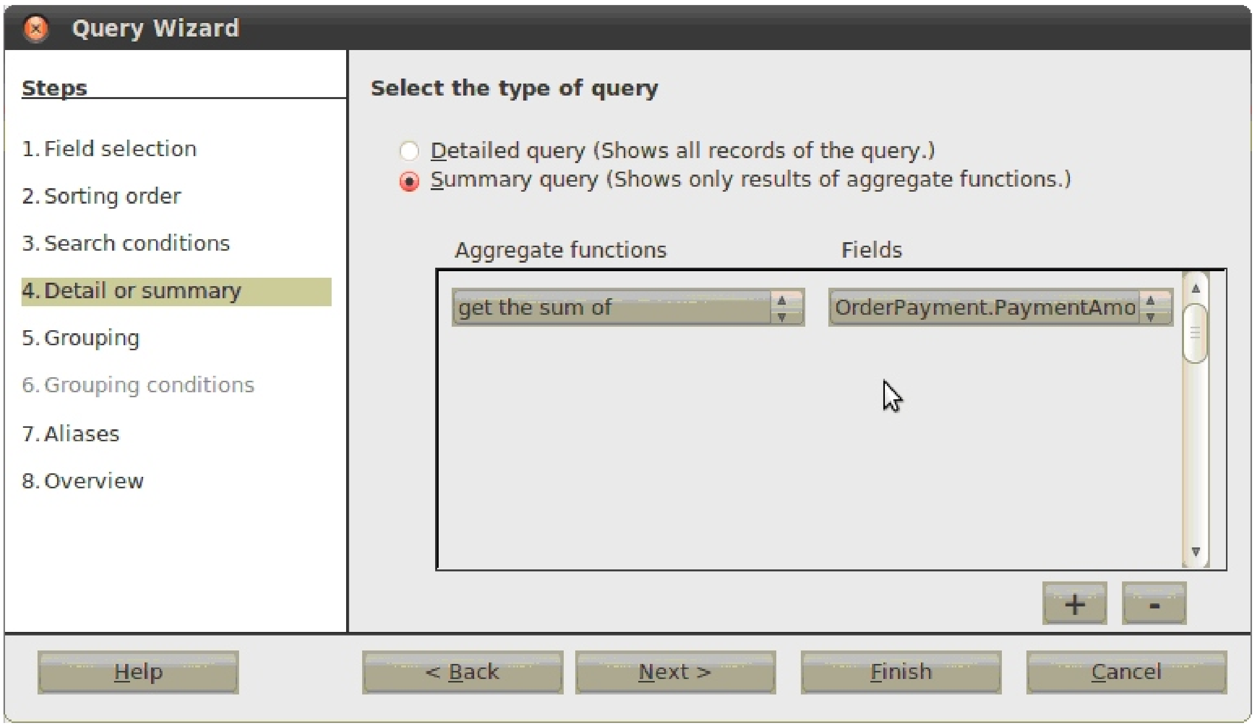
Select the type of query

Detailed query (Shows all records of the query.) [✓ Summary query (Shows only results of aggregate functions.)]

Aggregate functions   Fields

Figure 11.8 : Summarizing the result

- Select the *Summary query (Show only results of aggregate functions)* options as shown in figure 11.8.
- Click on the drop down box shown under *Aggregate functions* label. Select *get the sum of* option; similarly select the OrderPayment.PaymentAmount field from the *Fields* drop down box as we want to perform sum on the PaymentAmount field. This operation is shown in figure 11.9.


Query Wizard

Steps
1. Field selection
2. Sorting order
3. Search conditions
4. Detail or summary
5. Grouping
6. Grouping conditions
7. Aliases
8. Overview

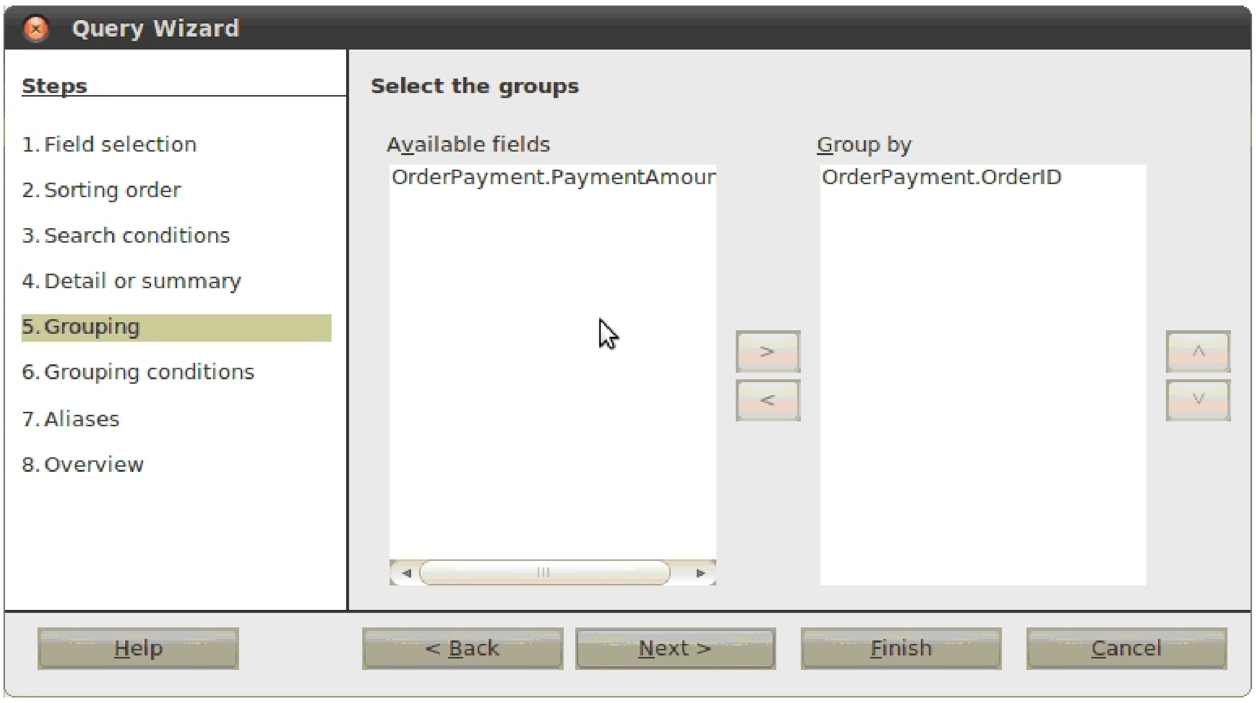
Select the type of query

Detailed query (Shows all records of the query.) [✓ Summary query (Shows only results of aggregate functions.)]

Aggregate functions	Fields
get the sum of	OrderPayment.PaymentAmo

Figure 11.9 : Usage of Aggregate function

- Click on the Next button. This will lead you to step 5. Here we have to specify the fields for which groups are to be created. We may choose to perform the sum of payment amounts of all records. But it will give us total amount received from all the customers so far by Modern Electronic Store. What we want is the details of payment received for each order. Thus if manually this operation is to be performed, records are to be grouped as per the OrderID field first. This means that we need to arrange the records having OrderID value as '1' in one group, all records having OrderID value as '2' in another group and so on. Then for each of this group, the payment amount is to be added. As a result, one record pertaining to each OrderID will appear in the query result. Thus in this step as seen in figure 11.10 the OrderID field is to be mentioned in *Group by* list box.


Query Wizard

Steps

1. Field selection
2. Sorting order
3. Search conditions
4. Detail or summary
5. Grouping
6. Grouping conditions
7. Aliases
8. Overview

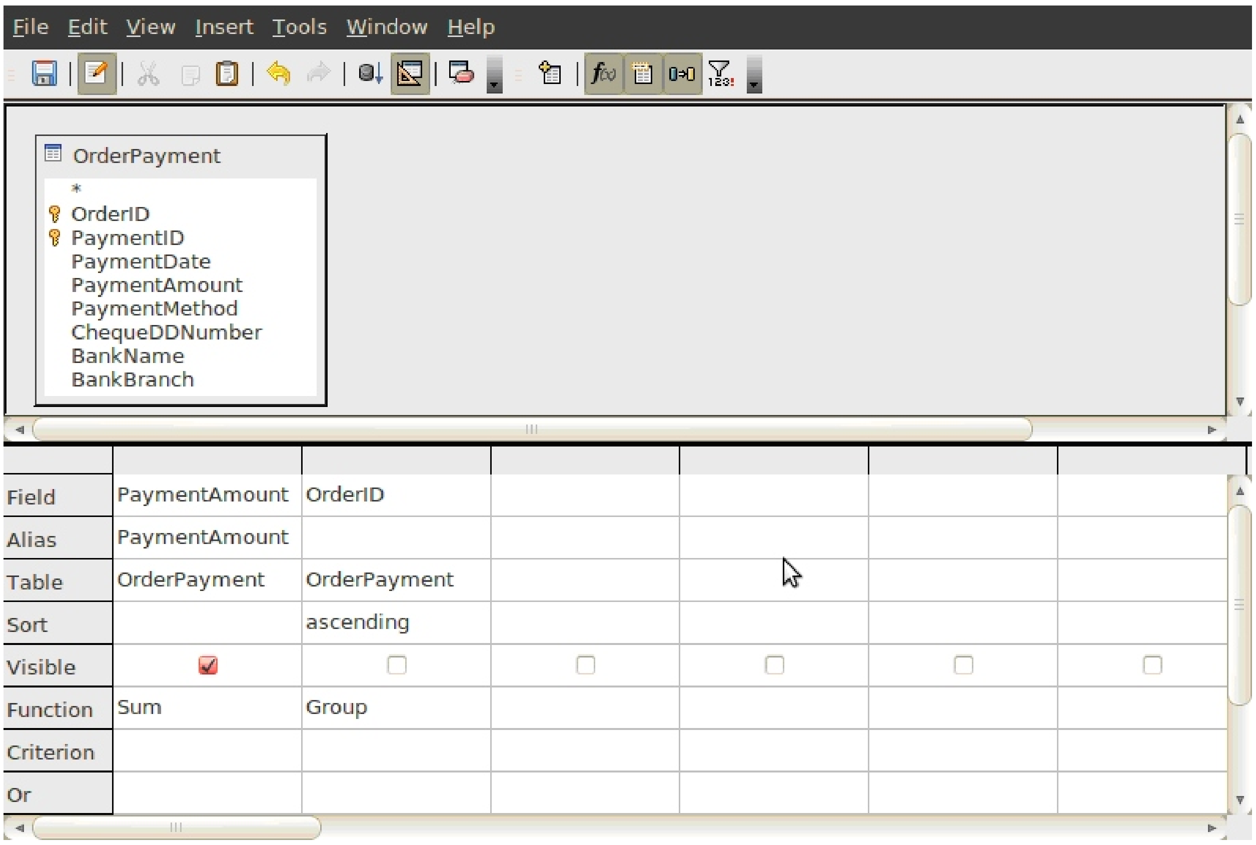
Select the groups

Available fields: OrderPayment.PaymentAmour

Group by: OrderPayment.OrderID

Figure 11.10 : Applying Group by on field(s)

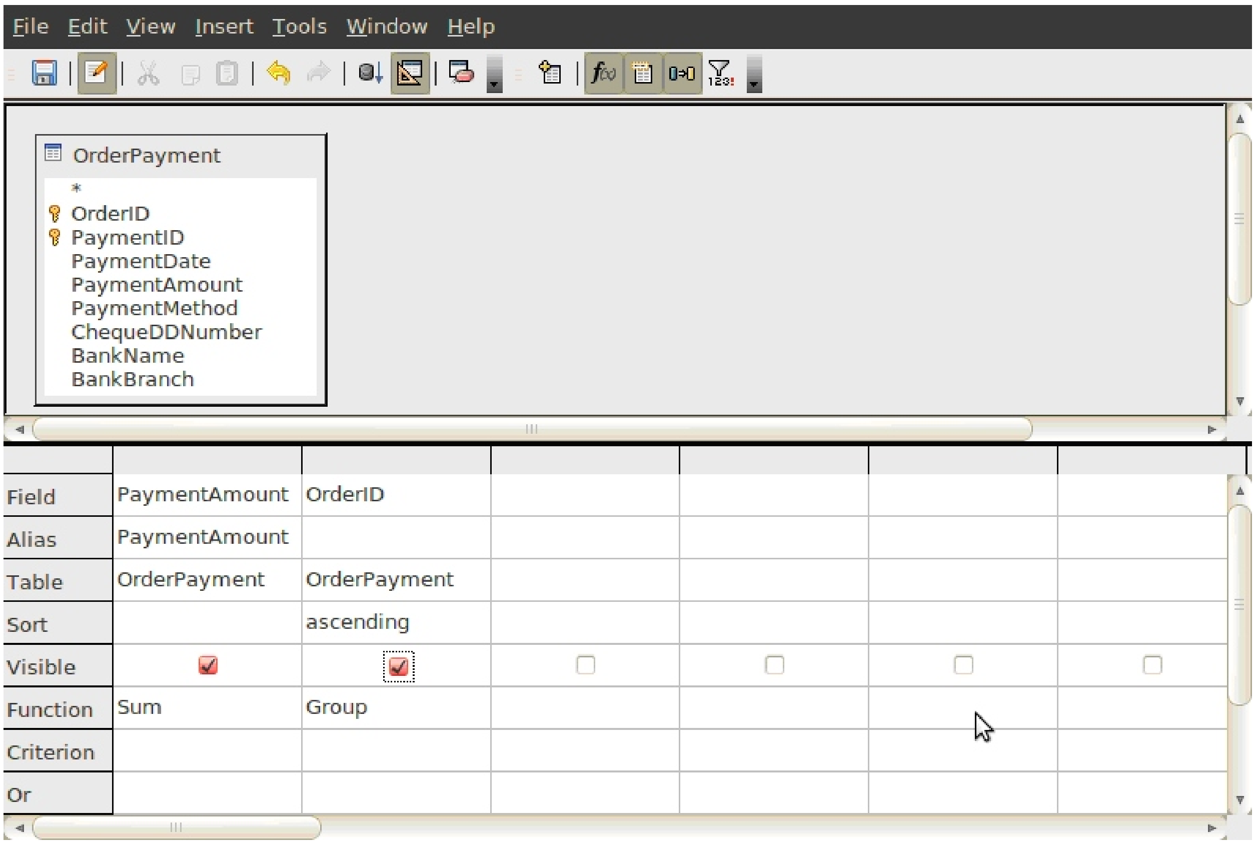
- Click on the Next button.
- In step 6, we can specify some grouping conditions to further filter the output if required. Assume that we want to retrieve only those records where payment amount is greater than Rs. 10000. We are not applying any additional filters here, so click on Next button.
- In step 7 mention Aliases if needed and click on Next button.
- In step 8 assign it a name Query_OrderPayment and select the *Modify Query* option. This option is recommended here because, if directly the query is executed, Base will show only one field, PaymentAmount corresponding to each order. However, we expect to see the OrderID along with each PaymentAmount field.
- Click on the Finish button. You will find that the query opens in a Design View as shown in figure 11.11.



Field	PaymentAmount	OrderID				
Alias	PaymentAmount					
Table	OrderPayment	OrderPayment				
Sort		ascending				
Visible	✓	☐	☐	☐	☐	☐
Function	Sum	Group				
Criterion						
Or						

Figure 11.11 : Design View of Query

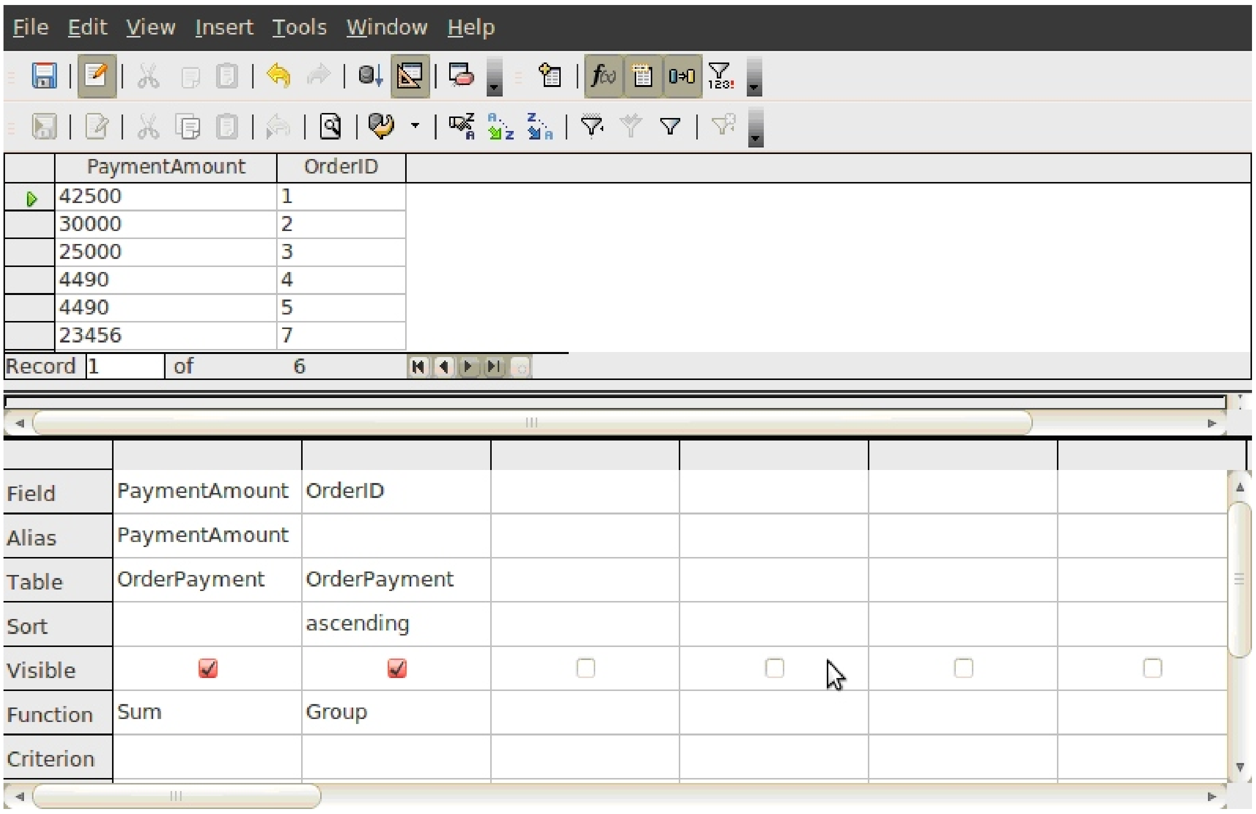
- Select the check box available under the OrderID field as shown in figure 11.12.



Field	PaymentAmount	OrderID				
Alias	PaymentAmount					
Table	OrderPayment	OrderPayment				
Sort		ascending				
Visible	✓	✓	☐	☐	☐	☐
Function	Sum	Group				
Criterion						
Or						

Figure 11.12 : Selecting field(s) for display

Click on the *Run Query* button ( ) to execute the query. You will find that the result similar to the one shown in figure 11.13 is displayed.



	PaymentAmount	OrderID
▷	42500	1
	30000	2
	25000	3
	4490	4
	4490	5
	23456	7

Record 1 of 6

Field	PaymentAmount	OrderID				
Alias	PaymentAmount					
Table	OrderPayment	OrderPayment				
Sort		ascending				
Visible	✓	✓	☐	☐	☐	☐
Function	Sum	Group				
Criterion						

Figure 11.13 : Result of aggregation

Creation of Query Using Design View

Let us now discuss usage of Design View to create a query. While creating a query to list the customer addresses, we mentioned that it is not possible to get the details from the City, State and Country table. This is due to the limitation of the wizard as it allows us to query on a single table only. So, we will now rewrite the same query using Design View.

- Click on the *Queries* icon in the Database Window.
- Double click on the *Create Query in Design View...* option in Tasks pane.
- We will see *Add Table or Query* dialog box as shown in figure 11.14.

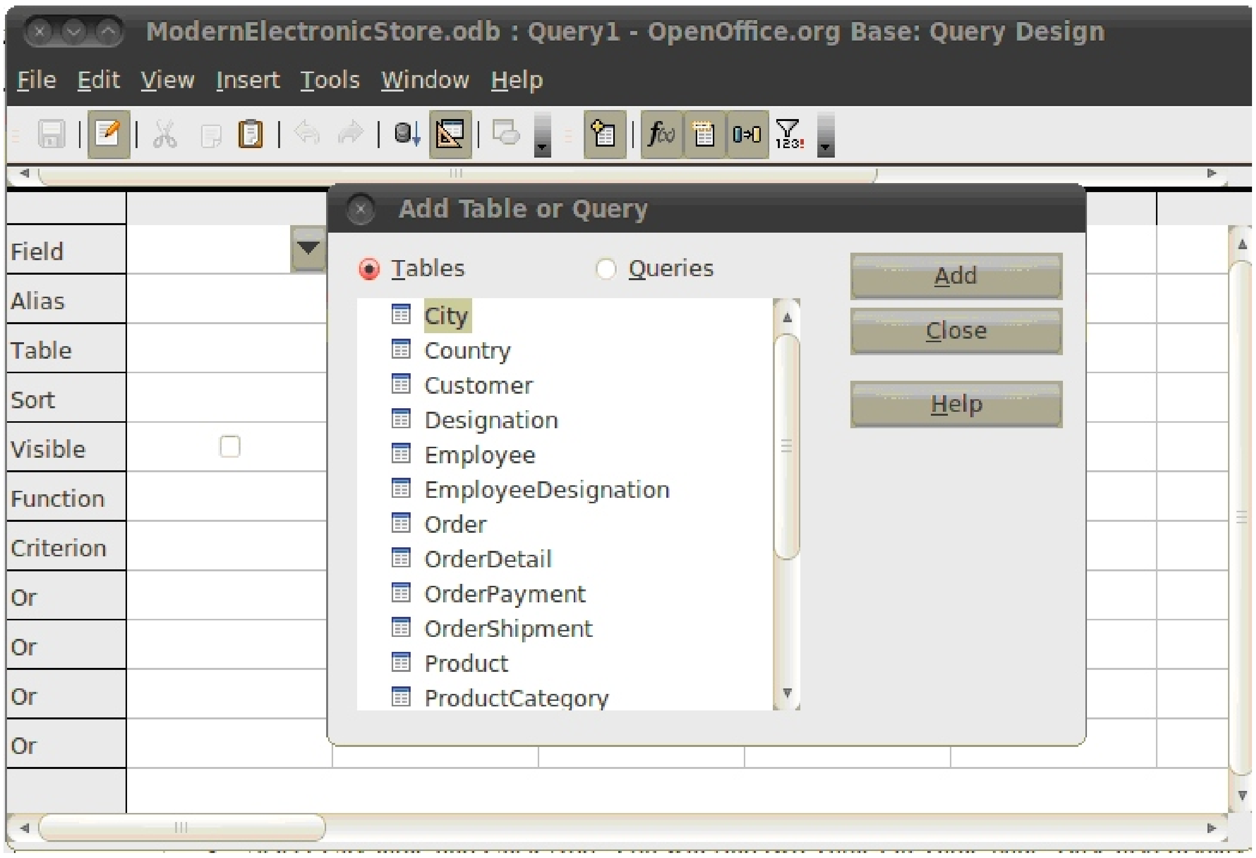
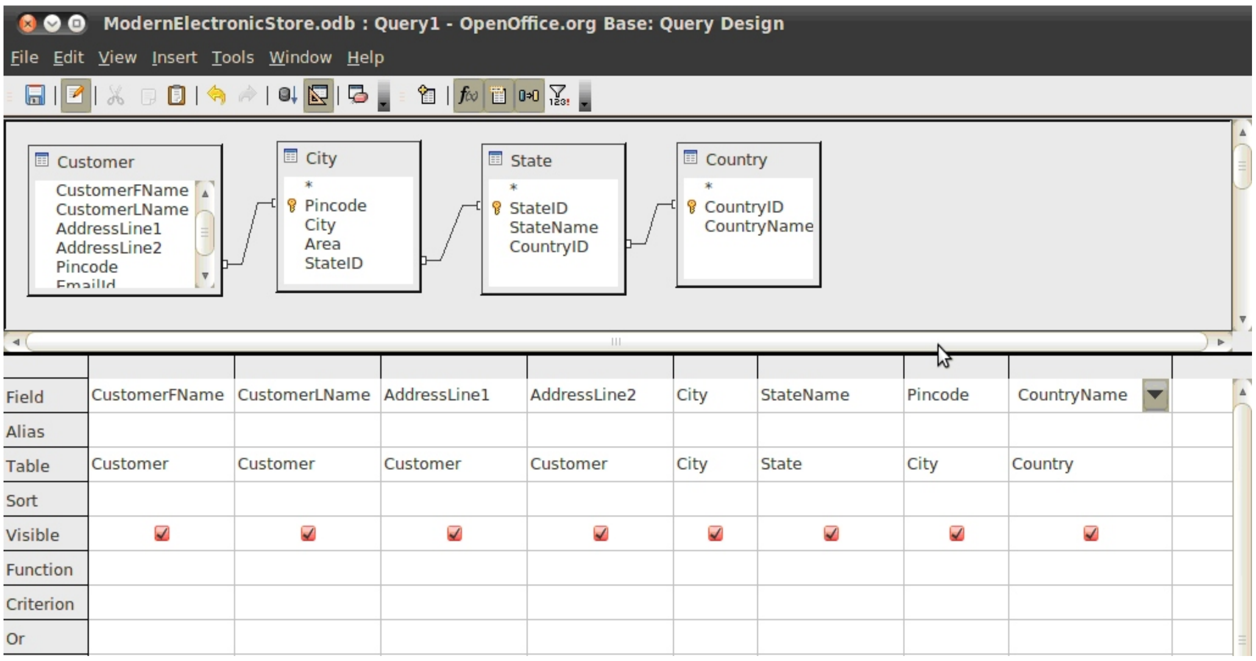


Figure 11.14 : Add Query or Table dialog box

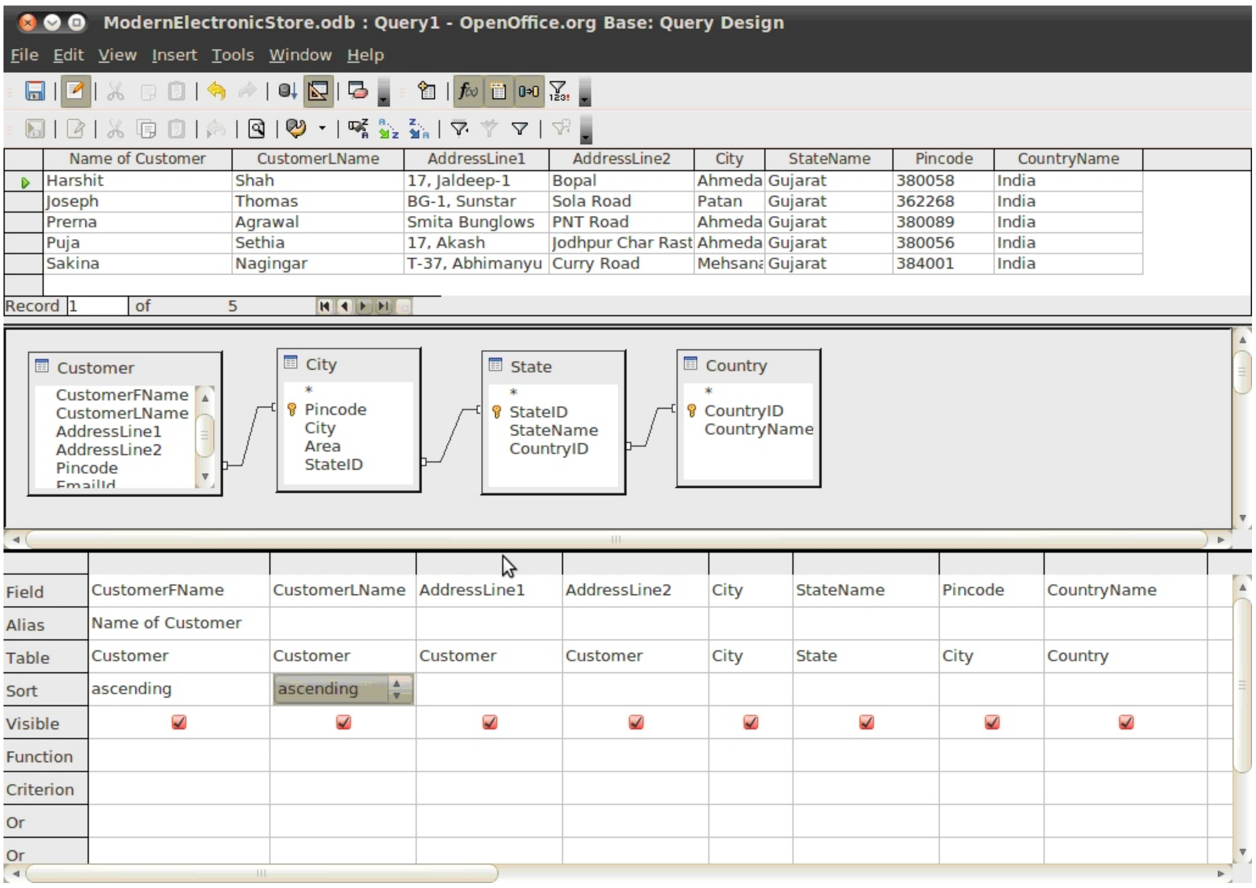
- Select the Customer table and click on Add button.
- Similarly select City, State and Country table. You will now find four tables in Table pane as shown in figure 11.15. Base also displays the relationship, which we have earlier defined between the tables.
- Click on the Close button. If you want to add some more tables, the you can open the *Add Table or Query* dialog box again by clicking on Add Table or Query button ( ) on Query design toolbar.
- Double click on CustomerFName, CustomerLName, AddressLine1, AddressLine2 fields from the Customer table. Similarly select City field from City table, StateName field from State table, Pincode field from City table and CountryName field from Country table. The field names along with their respective table names will be displayed in grid as shown in figure 11.15.



Field	CustomerFName	CustomerLName	AddressLine1	AddressLine2	City	StateName	Pincode	CountryName
Alias								
Table	Customer	Customer	Customer	Customer	City	State	City	Country
Sort								
Visible	✓	✓	✓	✓	✓	✓	✓	✓
Function								
Criterion								
Or								

Figure 11.15 : Selection of fields

- Observe that we are also able to see some record (row) headings like Alias, Sort, Visible, Function, Criterion and Or. You might have noticed that by default visible option for each field is set to true. It indicates that all selected fields will be displayed in the output.
- Alias can be used for displaying meaningful names for the fields. For example, in place of CustomerFname, we would prefer to use *Name of Customer* as column title in the query result. Type Name of Customer in the text box visible after the row heading *Alias* under the CustomerFName column.
- To display customer records in alphabetical order of his/her names, select a*scending* from drop down box visible after the row heading *Sort* under the CustomerFName column. Similarly, select a*scending* in the *Sort* option under the CustomerLName column.
- Click on the Run Query button ( ) on the Query Design toolbar. Query result similar to the one shown in figure 11.16 would be displayed.
- To save a query for later use, select the Save option from the File menu. Alternatively, click on the Close button and Base will display a Save dialog box.
- Type desired name, for example *CustomerAddresses* and click on OK button.



ModernElectronicStore.odb : Query1 - OpenOffice.org Base: Query Design

File Edit View Insert Tools Window Help

Name of Customer	CustomerLName	AddressLine1	AddressLine2	City	StateName	Pincode	CountryName
Harshit	Shah	17, Jaldeep-1	Bopal	Ahmeda	Gujarat	380058	India
Joseph	Thomas	BG-1, Sunstar	Sola Road	Patan	Gujarat	362268	India
Prerna	Agrawal	Smita Bunglows	PNT Road	Ahmeda	Gujarat	380089	India
Puja	Sethia	17, Akash	Jodhpur Char Rast	Ahmeda	Gujarat	380056	India
Sakina	Nagingar	T-37, Abhimanyu	Curry Road	Mehsan	Gujarat	384001	India

Record 1 of 5

Field	CustomerFName	CustomerLName	AddressLine1	AddressLine2	City	StateName	Pincode	CountryName
Alias	Name of Customer							
Table	Customer	Customer	Customer	Customer	City	State	City	Country
Sort	ascending	ascending						
Visible	✓	✓	✓	✓	✓	✓	✓	✓
Function								
Criterion								
Or								
Or								

Figure 11.16 : Query output

Close the query window once you have observed the output.

Editing a Query

After creating a query, one may like to change a query. For example, in the query created above, we would like to add *Surname* as an alias in CustomerLName column.

To make this change perform the following steps:

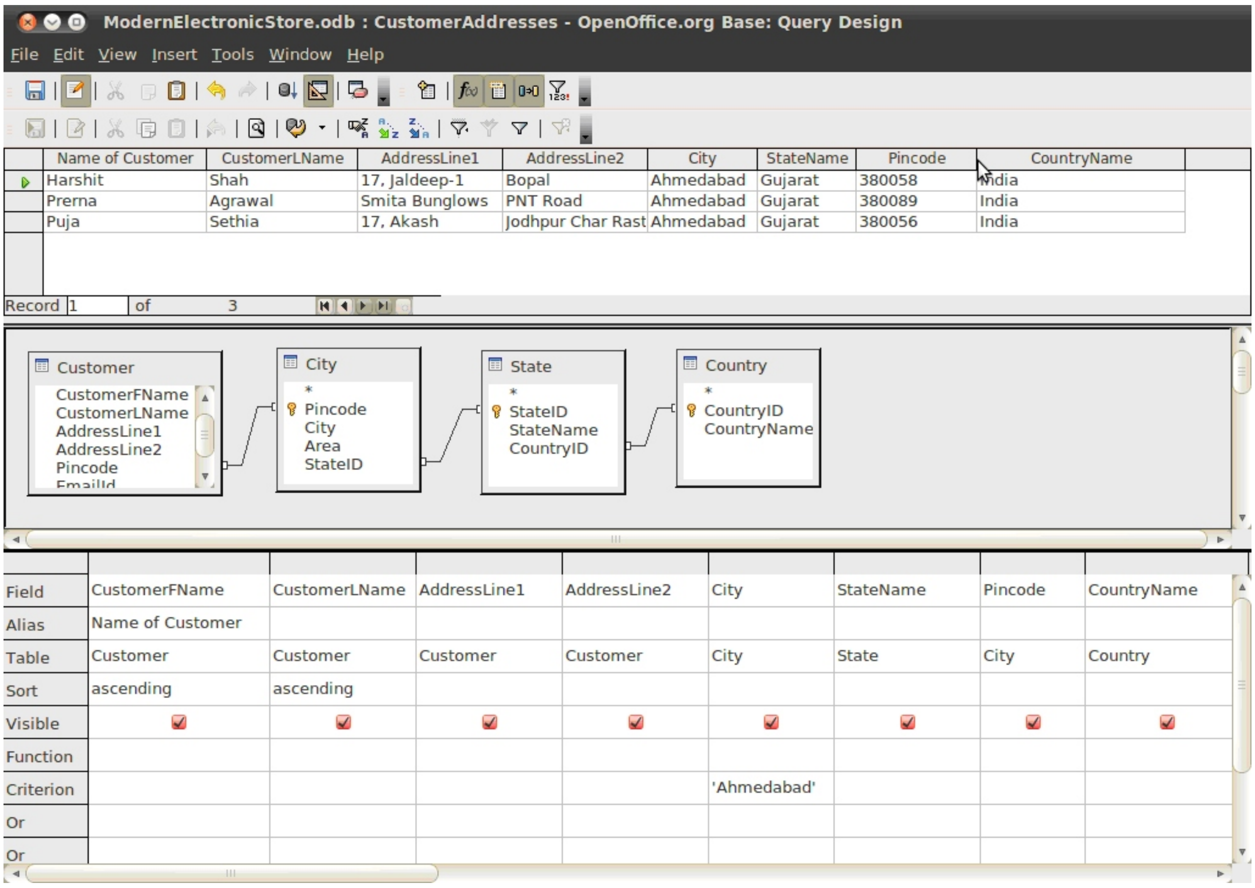
- Click on *Queries* icon. Right click on the query CustomerAddresses, from the popup menu choose *Edit* option. This will display the query in Design View.
- Type *Surname* in the text box visible after row heading *Alias* under the CustomerLName column.
- Run Query.

Applying Criteria

We have seen that we can write a query, which displays selected fields of a table. Now suppose, instead of viewing all records we wish to view the details of customers residing in the city of Ahmedabad. This means we want Base to display a subset of selected records. To do this, we can specify a criterion that limits the records to only those where the City field contains "Ahmedabad" as a value.

Using Single Field

- Right click on the CustomerAddresses Query.
- Click Edit option to open the query in Design View.
- In the *Criterion* cell of the City field type “Ahmedabad” as shown in figure 11.17. Note that text must be enclosed within a quotation (? ’) delimiter; date must be enclosed in the hash (#) delimiter while the number literals do not need any delimiters. If we miss to put delimiters, Base will not report any error; instead will apply delimiters on its own.
- Save and Run the query and you will find the desired result.



Name of Customer	CustomerLName	AddressLine1	AddressLine2	City	StateName	Pincode	CountryName
Harshit	Shah	17, Jaldeep-1	Bopal	Ahmedabad	Gujarat	380058	India
Prerna	Agrawal	Smita Bunglows	PNT Road	Ahmedabad	Gujarat	380089	India
Puja	Sethia	17, Akash	Jodhpur Char Rast	Ahmedabad	Gujarat	380056	India

Record 1 of 3

Field	CustomerFName	CustomerLName	AddressLine1	AddressLine2	City	StateName	Pincode	CountryName
Alias	Name of Customer							
Table	Customer	Customer	Customer	Customer	City	State	City	Country
Sort	ascending	ascending						
Visible	✓	✓	✓	✓	✓	✓	✓	✓
Function								
Criterion					'Ahmedabad'			
Or								
Or								

Figure 11.17 : Setting Criteria

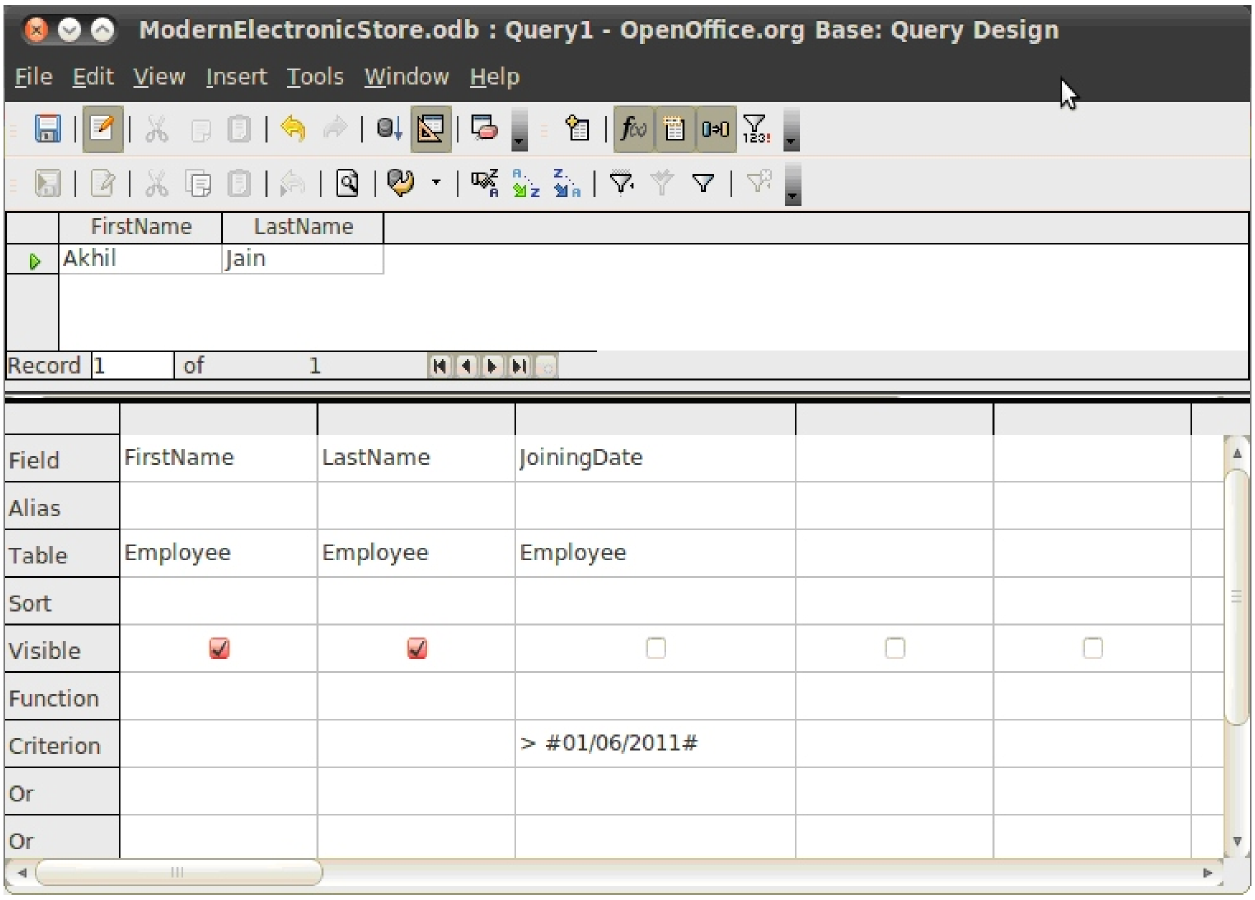
Apart from the constant values used as shown in figure 11.17, Base also allows us to design expression for defining criteria using different types of operators. The operator list is as shown in table 11.1.

Operator	Symbols
Comparison	=, >, <, >=, <=, <>
Logical	And, Or, Not
Special	Like, Is, Between, In

Table 11.1 : Operators used in Base

Suppose we want to display the list of employees who joined after 1st June 2011. Then create a new query in Design View. Add the Employee table. Then select fields FirstName, LastName and JoiningDate. Type "**> #01/06/2011#**" in the JoiningDate fields *Criterion* cell. De-select the check box displayed in the *Visible* cell under the JoiningDate column. Now Save and Run the query. The output will be similar to the one shown in figure 11.18. Observe that the JoiningDate field is not displayed.

Note : When you do not want to display any field used in the query in the output, clear the check mark shown in the Visible property of that field.


ModernElectronicStore.odb : Query1 - OpenOffice.org Base: Query Design
File
Edit
View
Insert
Tools
Window
Help

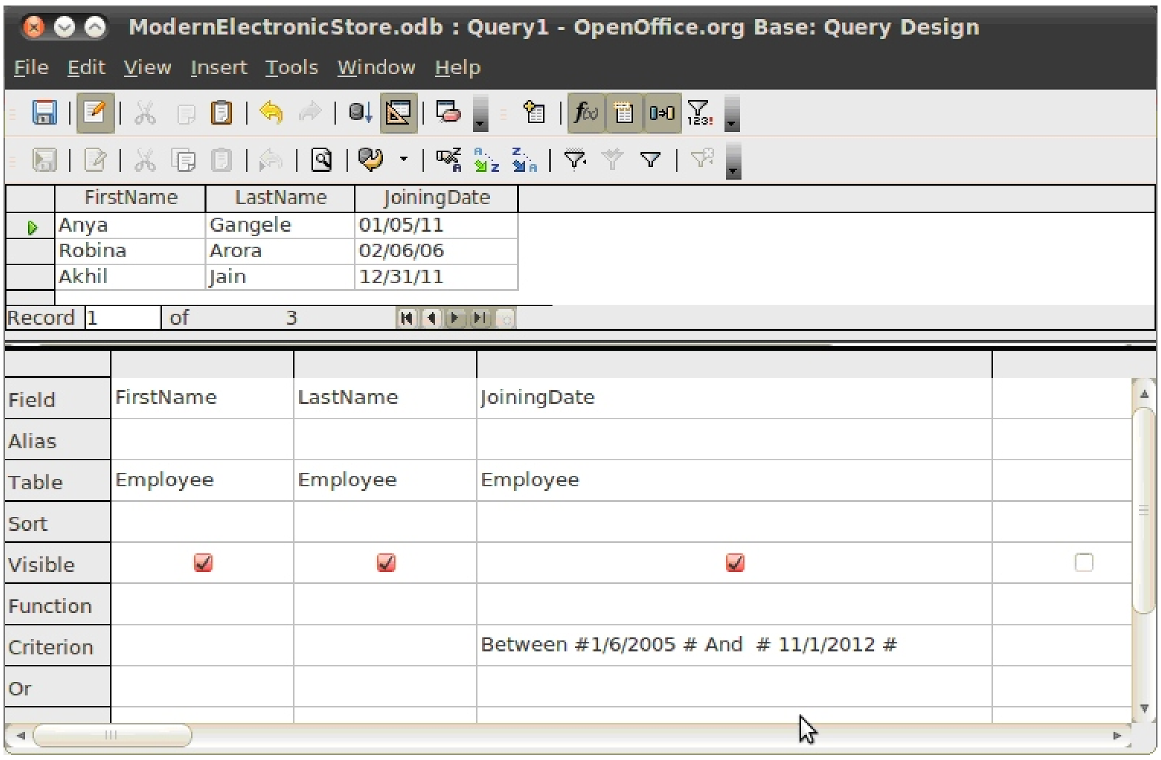
FirstName	LastName
Akhil	Jain

Record 1 of 1

Field	FirstName	LastName	JoiningDate		
Alias					
Table	Employee	Employee	Employee		
Sort					
Visible	✓	✓	☐	☐	☐
Function					
Criterion			> #01/06/2011#		
Or					
Or					

Figure 11.18 : Applying Criteria in Date Field

Similarly to display employees who joined between 1st June 2005 and 1st Nov 2012, The *Criterion* in the JoiningDate field can be set as " **>= # 1/6/2005 # And <= # 11/1/2012 #**". Base also offers *Between* operator to specify the same criteria as shown in figure 11.19.


ModernElectronicStore.odb : Query1 - OpenOffice.org Base: Query Design
File Edit View Insert Tools Window Help

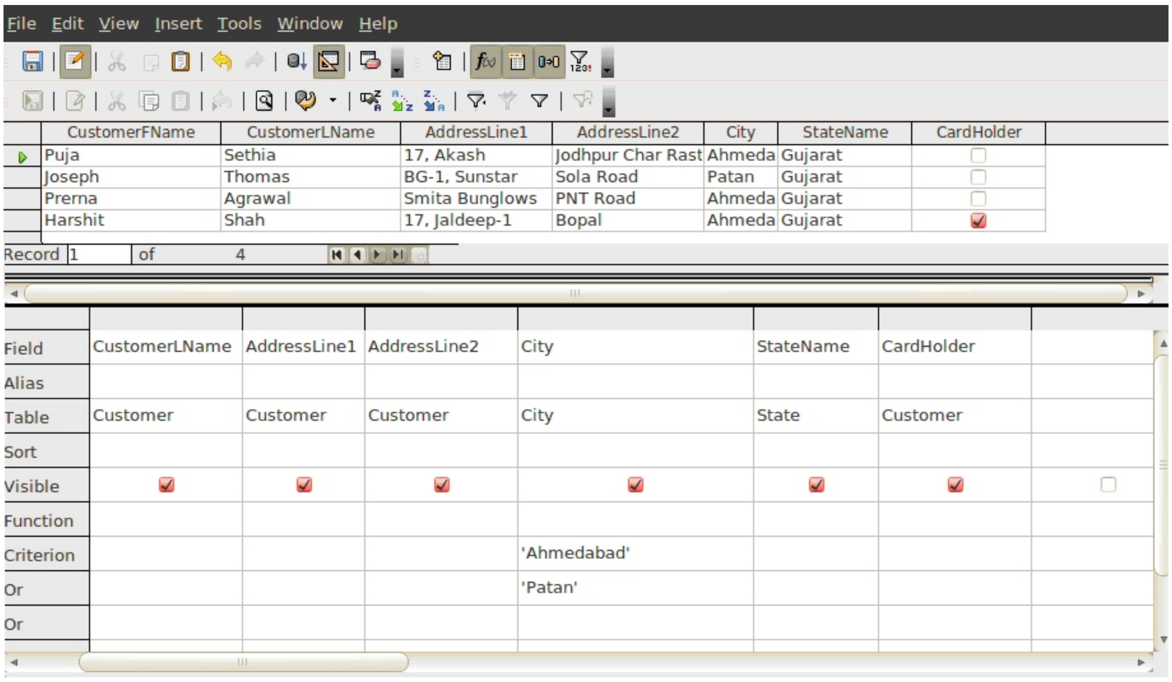
FirstName	LastName	JoiningDate
Anya	Gangele	01/05/11
Robina	Arora	02/06/06
Akhil	Jain	12/31/11

Record 1 of 3

Field	FirstName	LastName	JoiningDate	
Alias				
Table	Employee	Employee	Employee	
Sort				
Visible	✓	✓	✓	☐
Function				
Criterion			Between #1/6/2005 # And # 11/1/2012 #	
Or				

Figure 11.19 : Using the Between Operator

Now, suppose that we want to send discount coupons to customers who live in the city of Ahmedabad and Patan. We need a list of customers residing either in the city of Ahmedabad or Patan. Then create a new query in Design View. Add the Customer, City and State tables. Then select CustomerFName, CustomerLName, AddressLine1, AddressLine2, City, StateName and CardHolder fields from the tables as shown in figure 11.20. Type the criterion as can be seen in the City field. Now Save and Run the query. The output will be similar to the one shown in figure 11.20.



CustomerFName	CustomerLName	AddressLine1	AddressLine2	City	StateName	CardHolder
Puja	Sethia	17, Akash	Jodhpur Char Rast	Ahmeda	Gujarat	☐
Joseph	Thomas	BG-1, Sunstar	Sola Road	Patan	Gujarat	☐
Prerna	Agrawal	Smita Bunglows	PNT Road	Ahmeda	Gujarat	☐
Harshit	Shah	17, Jaldeep-1	Bopal	Ahmeda	Gujarat	✓

Record 1 of 4

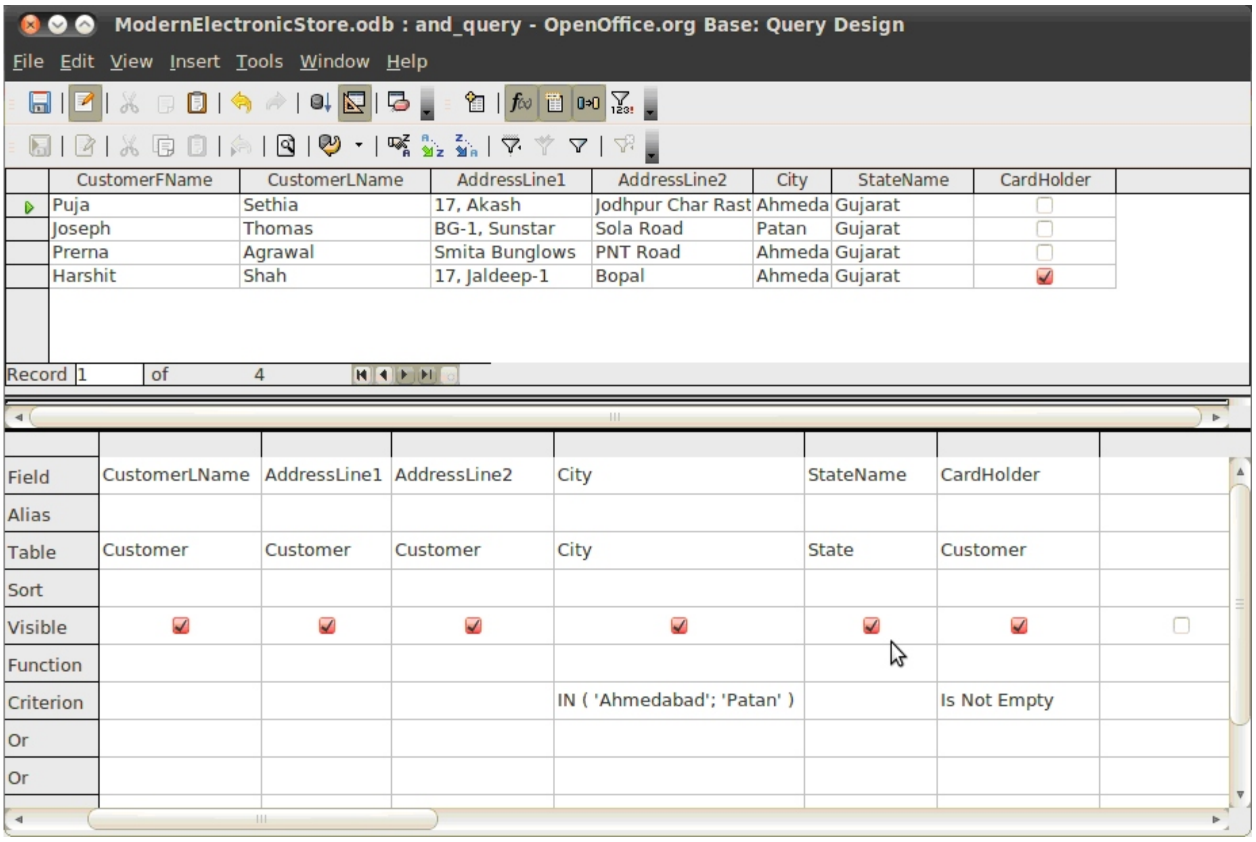
Field	CustomerLName	AddressLine1	AddressLine2	City	StateName	CardHolder	
Alias							
Table	Customer	Customer	Customer	City	State	Customer	
Sort							
Visible	✓	✓	✓	✓	✓	✓	☐
Function							
Criterion				'Ahmedabad'			
Or				'Patan'			
Or							

Figure 11.20 : Using Multiple Criteria on single field

Criteria for the above query can also be specified using the *IN* operator. Type *IN ('Ahmedabad'; 'Patan')* in the *Criterion* row of the City field and you will get the same result. You can use *NOT IN ('Ahmedabad'; 'Patan')* in *Criterion* row of the City field to retrieve records of customers from the all other places except Ahmedabad or Patan.

Using Multiple Fields

Now suppose we want to send discount coupons only to card holders of Modern Electronic Store residing in Ahmedabad or Patan. For solving this problem, we need to use the *AND* operator which specifies that City should be (either Ahmedabad or Patan) and (CardHolder field should not be empty). Here, we need to apply *OR* criteria within field and *AND* criteria between fields. Type the criteria as shown in the figure 11.21. Here the value NULL (No Check mark in CardHolder field) signifies that field is empty.


ModernElectronicStore.odb : and_query - OpenOffice.org Base: Query Design
File Edit View Insert Tools Window Help

CustomerFName	CustomerLName	AddressLine1	AddressLine2	City	StateName	CardHolder
Puja	Sethia	17, Akash	Jodhpur Char Rast	Ahmeda	Gujarat	☐
Joseph	Thomas	BG-1, Sunstar	Sola Road	Patan	Gujarat	☐
Prerna	Agrawal	Smita Bunglows	PNT Road	Ahmeda	Gujarat	☐
Harshit	Shah	17, Jaldeep-1	Bopal	Ahmeda	Gujarat	✓

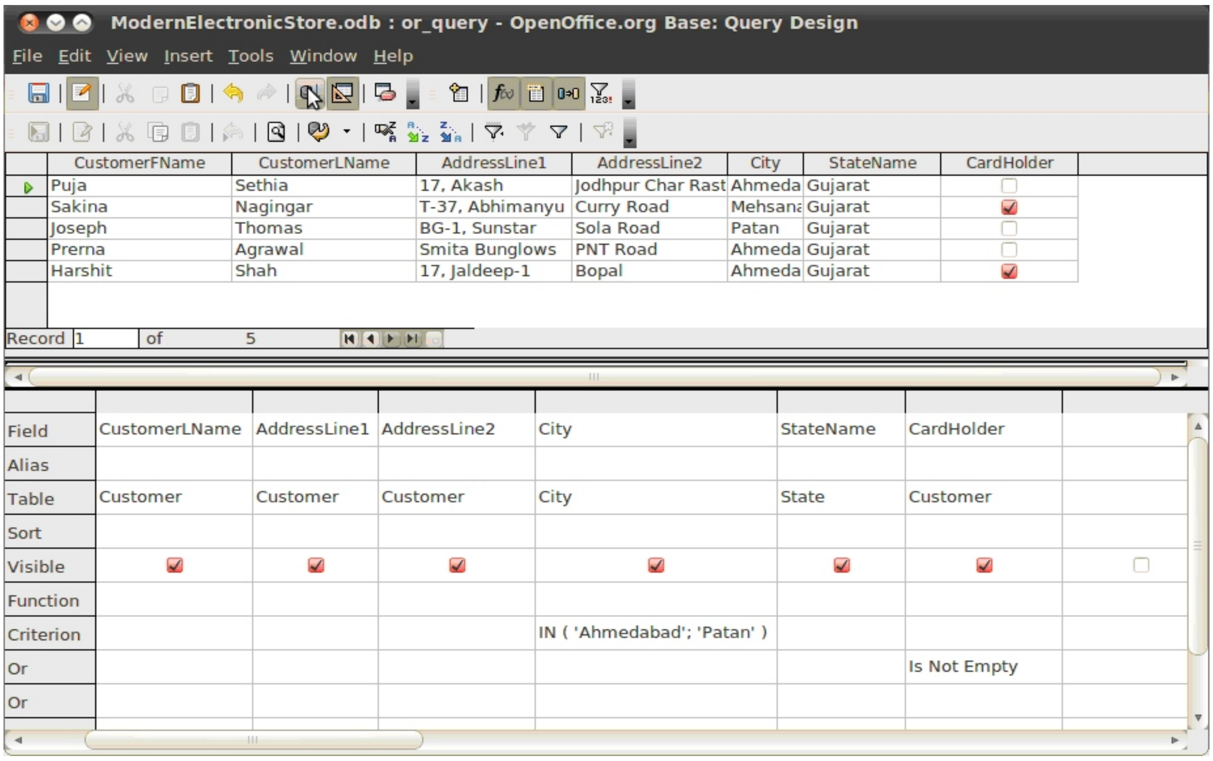
Record 1 of 4

Field	CustomerLName	AddressLine1	AddressLine2	City	StateName	CardHolder	
Alias							
Table	Customer	Customer	Customer	City	State	Customer	
Sort							
Visible	✓	✓	✓	✓	✓	✓	☐
Function							
Criterion				IN ('Ahmedabad'; 'Patan')		Is Not Empty	
Or							
Or							

Figure 11.21 : Using Criteria on multiple fields

Now, if you want to view the list of customers who can belong to either Ahmedabad or Patan or has Membership Card. Then type *IN ('Ahmedabad', 'Patan')* in the *Criterion* row of the City column. Also type *Is Not Empty* in the *Or* row of the CardHolder column (See figure 11.22).

Note the difference between previous query and this query. In the first case we wrote both the expressions in same row while applying the AND condition between two fields. While in this query, the OR condition between two fields is written in separate rows. Observe that in the result set as shown in figure 11.22, a new record of a customer from Mehsana is also listed as the customer is also a membership card holder.


ModernElectronicStore.odb : or_query - OpenOffice.org Base: Query Design

CustomerFName	CustomerLName	AddressLine1	AddressLine2	City	StateName	CardHolder
Puja	Sethia	17, Akash	Jodhpur Char Rast	Ahmeda	Gujarat	☐
Sakina	Nagingar	T-37, Abhimanyu	Curry Road	Mehsana	Gujarat	✓
Joseph	Thomas	BG-1, Sunstar	Sola Road	Patan	Gujarat	☐
Prerna	Agrawal	Smita Bunglows	PNT Road	Ahmeda	Gujarat	☐
Harshit	Shah	17, Jaldeep-1	Bopal	Ahmeda	Gujarat	✓

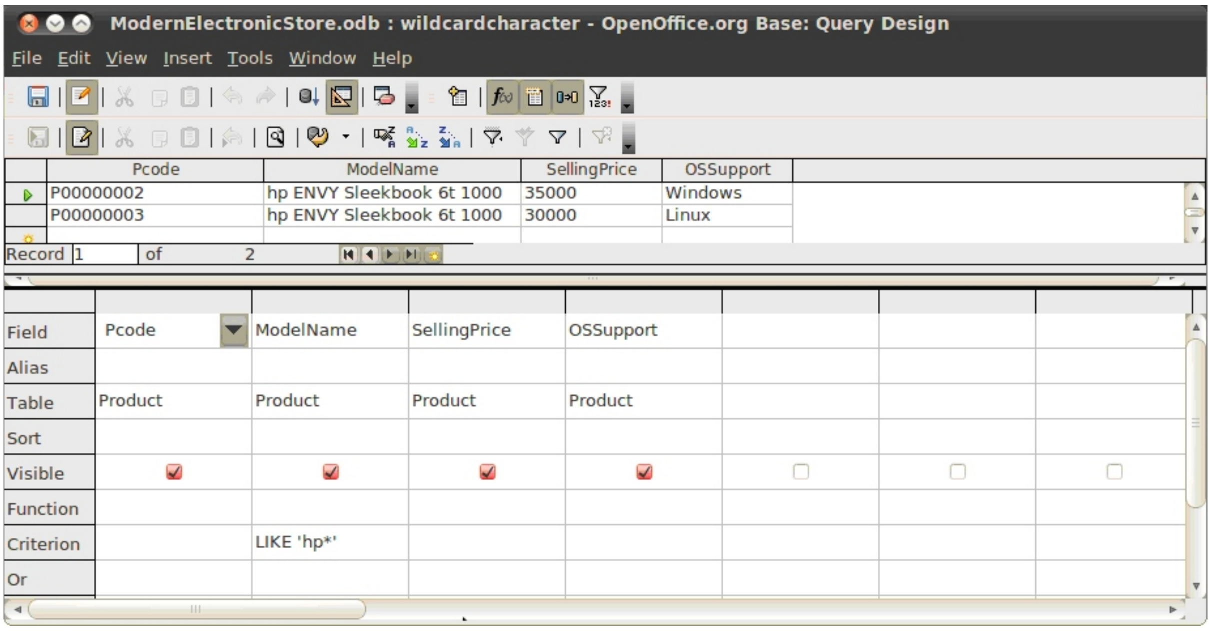
Record 1 of 5

Field	CustomerLName	AddressLine1	AddressLine2	City	StateName	CardHolder	
Alias							
Table	Customer	Customer	Customer	City	State	Customer	
Sort							
Visible	✓	✓	✓	✓	✓	✓	☐
Function							
Criterion				IN ('Ahmedabad'; 'Patan')			
Or						Is Not Empty	
Or							

Figure 11.22 : Applying OR Criteria in multiple fields

Using Wild Cards

Suppose we want to see the list of products with their model names starting with character set *hp*. Then create a new query using table Product. Select fields Pcode, ModelName, SellingPrice and OSSupport. Set the criterion as shown in figure 11.23.


ModernElectronicStore.odb : wildcardcharacter - OpenOffice.org Base: Query Design

Pcode	ModelName	SellingPrice	OSSupport
P00000002	hp ENVY Sleekbook 6t 1000	35000	Windows
P00000003	hp ENVY Sleekbook 6t 1000	30000	Linux

Record 1 of 2

Field	Pcode	ModelName	SellingPrice	OSSupport			
Alias							
Table	Product	Product	Product	Product			
Sort							
Visible	✓	✓	✓	✓	☐	☐	☐
Function							
Criterion		LIKE 'hp*'					
Or							

Figure 11.23 : Wild Cards

The asterisk symbol (*) used in expression in *Criterion* cell of ModelName field in figure 11.23 is known as wild card character. A wild card is a symbol that represents any character or combination of characters. Thus 'hp*' represents a word whose first two alphabets are hp, followed by any group of characters. Similarly, the criterion 'Like *hp' will display products with names ending with alphabets 'hp' and 'Like hp*hp' will display products with names starting and ending with alphabets 'hp'. Note that you must include the Like operator with the wild card characters.

Performing Calculations

Calculations within a query in Base can be performed using one of the following ways:

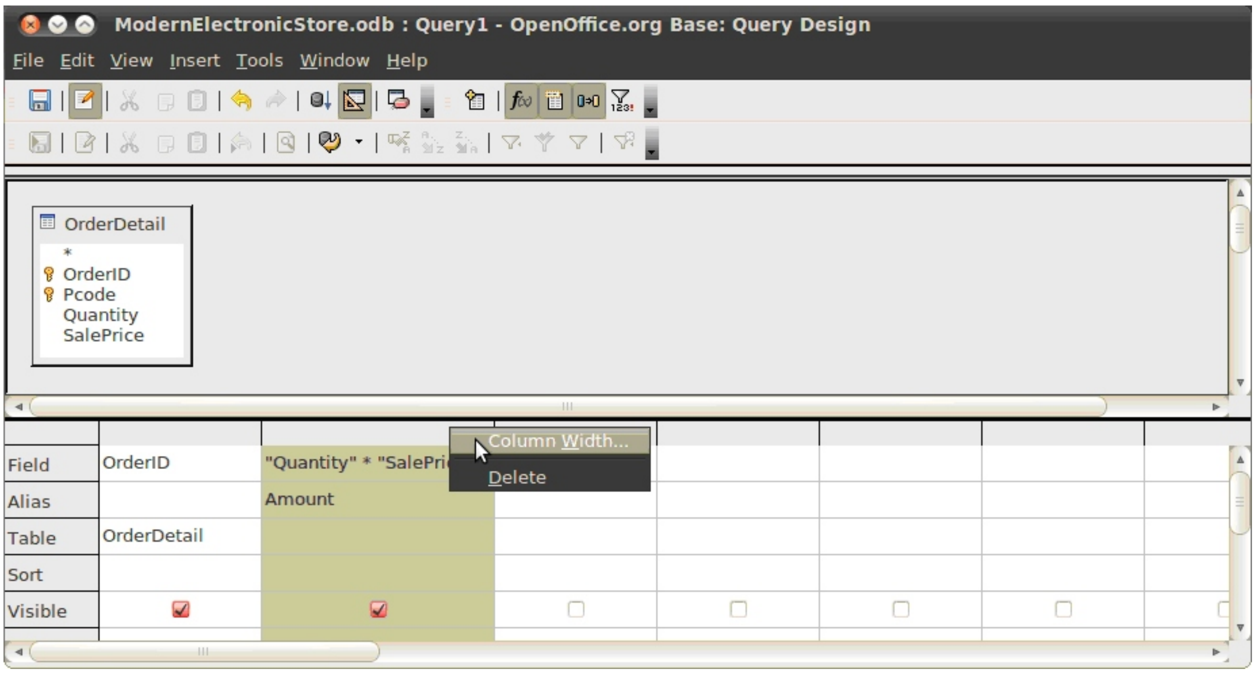
- Custom calculation
- Predefined calculation

Custom calculation: It includes performing numeric, date, and text calculations on each record using data from one or more fields. Calculations include operations like add, multiply, subtract or divide the values in two different fields. To perform custom calculation we need to add an extra field known as calculated field.

Let us discuss it with an example. Look at the OrderDetail table. If we want to know total amount pertaining to each product purchased by a customer, we can calculate it. The value of Amount is equal to the Quantity multiplied by SalePrice. The new field that stores information about Amount is known as calculated field.

Perform the following steps to find out total amount paid by each customer in each month.

- Create a new query using Design View.
- Select the OrderDetail table from *Add Table or Query* dialog box.
- Double click on the OrderID field to include it in design grid.
- Type Quantity * SalePrice in the *Field* row of the second column in the query design grid. You may not be able to see the entire entry because the *Field* row is not large enough.
- Right click the *Field* row in the second column in the design grid and then select *Column Width* from the popup menu displayed as shown in figure 11.24 would be displayed. Specify 4.20 cm as width to make text in the *Field* row visible.
- In the *Alias* row under the second column, type *Amount*. (See figure 11.24)


ModernElectronicStore.odb : Query1 - OpenOffice.org Base: Query Design
File Edit View Insert Tools Window Help
OrderDetail
*
OrderID
Pcode
Quantity
SalePrice
Column Width...
Delete

Field	OrderID	"Quantity" * "SalePri
Alias		Amount
Table	OrderDetail	
Sort		
Visible	✓	✓

Figure 11.24 : Using Calculated Fields

Predefined calculations (Summarizing the data): We have seen how to perform calculation on fields. Many times we need to perform calculations on group of records rather than on fields. For example, finding the total number of products, or computing average amount spend by each customer, we need to perform summary calculation. Base provides some predefined calculations to compute sum, average, count, minimum, maximum, standard deviation, or variance on group of records. These calculations differ from calculated fields as they are applied on multiple records within a table resulting in a single value. Let us design a query for finding total number of customers of Modern Electronic Store by performing the following steps:

- Click on *Queries* icon and select *Create Query in Design View...*
- Select the Customer table from *Add Table or Query* dialog box.
- Double click on the *Ccode* field.
- Type "Total Customers" in *Alias* row.
- In a row with a label *Function* in query design grid, open the drop down menu.
- Select the *Count* function from the list of aggregate functions as shown in figure 11.25.
- Run the query and you will get the total number of customers of Modern Electronic Store.

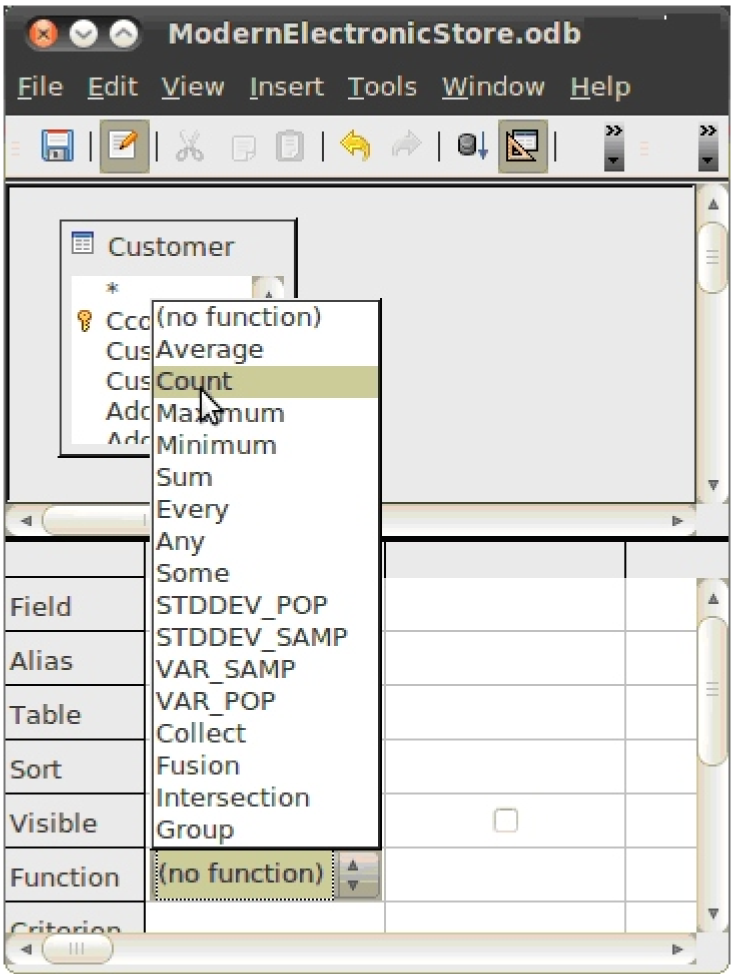


Figure 11.25 : Using Aggregate Function

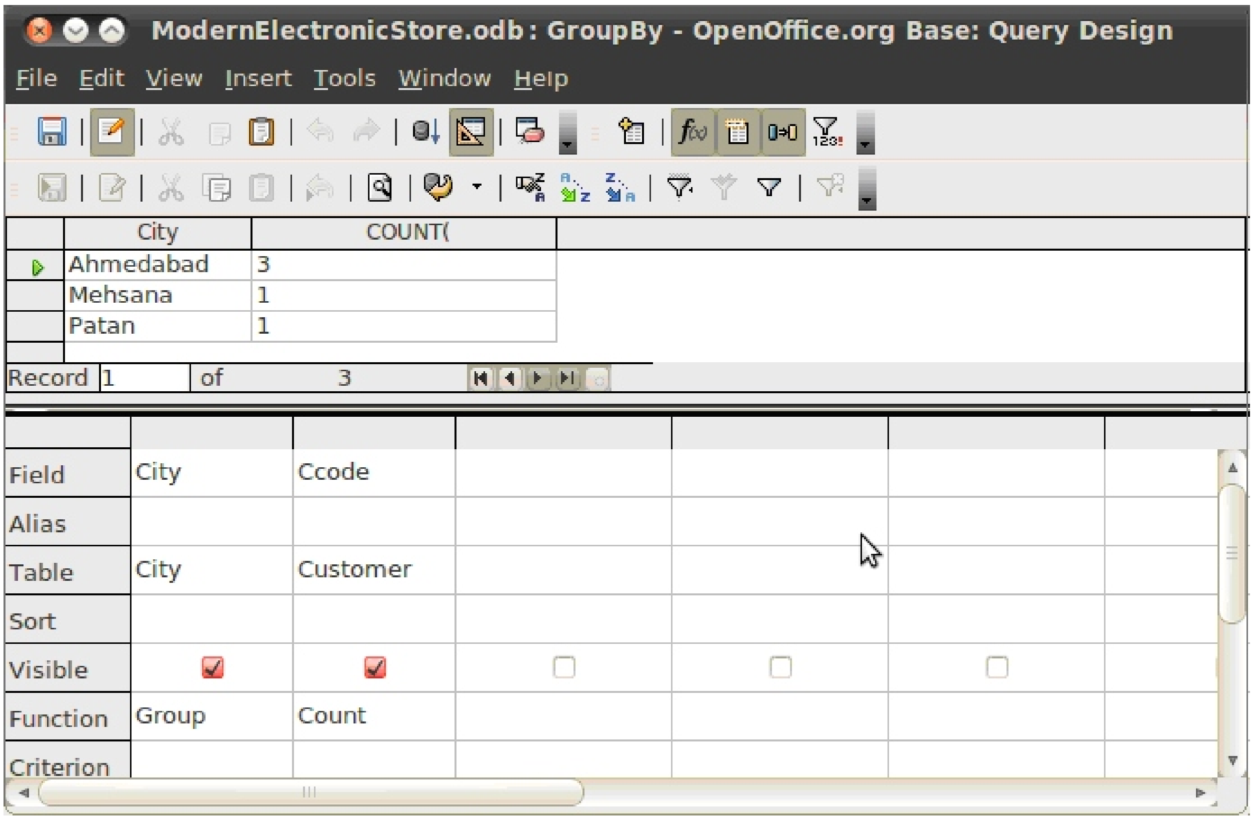
Remember that the results of a calculation in a field are only for display purpose. These results aren't actually stored in the underlying table. Instead, Base performs the calculation each time you run the query. Thus a result generated from such queries is always based on the most current data available in the database.

Grouping the Data

Earlier, we have seen how to retrieve the list of customers residing in the city of Ahmedabad or Patan. Here we provided customer details, which proved useful in sending discount coupons. Now the manager has another query. "Can I have total number of customers residing in each of these cities". Total should be corresponding to each city mentioned in the Customer table. You might think that yes, we can apply total on the CustomerFname field which will give you count. Try it and view the result. You will find that you are wrong!!!. The count you compute as a result is total number of customers residing in Ahmedabad or Patan. Fine, you can say that I will apply the count function on the City field. Try it and again you will end up with incorrect result.

Take pen and paper. Now start thinking how it is possible. Identify the cities of customers. It may be Ahmedabad, Mehsana or Patan etc. Make a group of customers in Ahmedabad, customers in Patan, and similarly for all cities. Now apply total on each group and you will get the result, which your manager requires. You might think that it's really complex. Instead it is very easy when you do it with Base. Perform the following steps:

- Open a new query in Design View.
- Add the Customer table and the City table from *Add Table or Query* dialog box.
- Double click on the City field from the City table. Similarly select the Ccode field from the Customer table.
- In the *Function* row of the City field, select *Group By* as seen in figure 11.26.
- In the *Function row* of Ccode field, select *Count*.
- Run the query and desired result will be displayed.


ModernElectronicStore.odb : GroupBy - OpenOffice.org Base: Query Design
File Edit View Insert Tools Window Help

City	COUNT(
Ahmedabad	3
Mehsana	1
Patan	1

Record 1 of 3

Field	City	Ccode
Alias		
Table	City	Customer
Sort		
Visible	✓	✓
Function	Group	Count
Criterion		

Figure 11.26 : Grouping the Records to summarize result

Parameter Query: Parameter queries are designed to accept values from the user at run time. Till now the queries that we created used fixed criterion. The criterion once defined will not be changed for every execution of the query. The output of the query may though vary depending on the current data in the table. Generally when we run a parameter query it will display a dialog box asking the use to enter the values of the parameter. These values are then assigned as criterion values for retrieving the data.

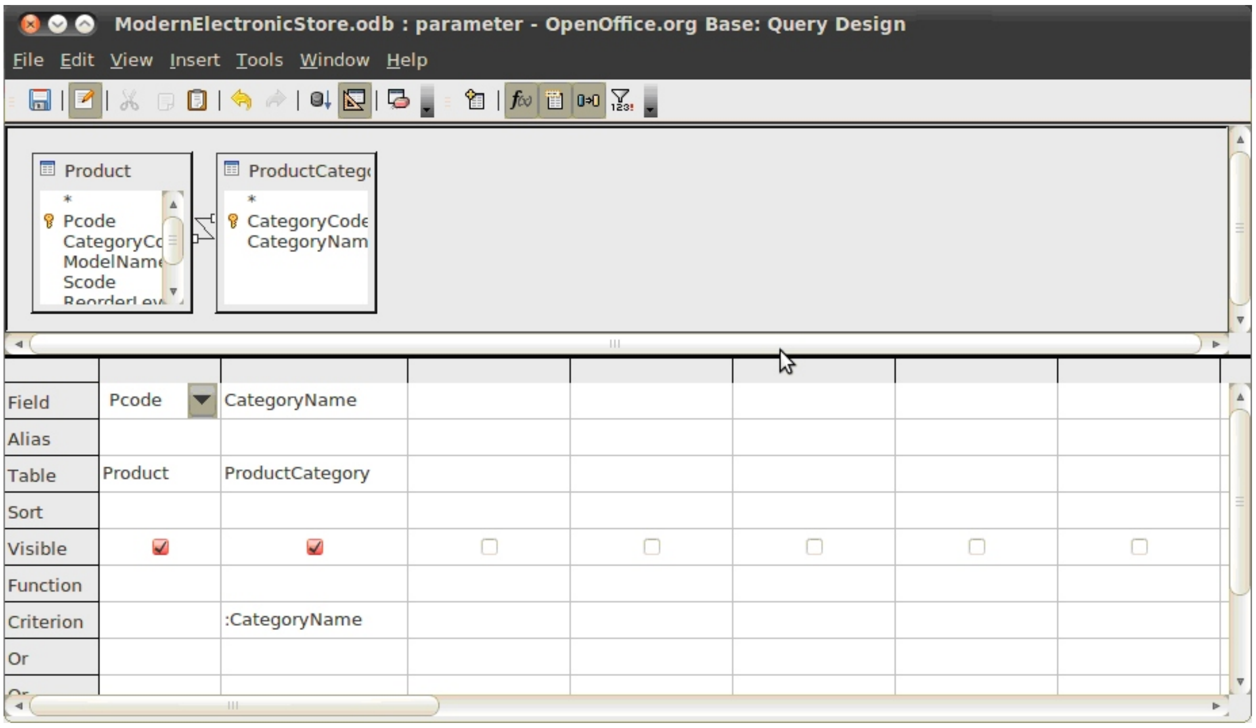
Let us design a query to display the detail of laptops available at Modern Electronic Store. Following steps when used will give us the desired result.

- Open new query in Design View.
- Add the Product and ProductCatgory table.
- Double click on * visible in the Product table. It will add all the fields of the Product table in the query.
- Double click on the CategoryName field from the Category table.
- Type *Laptop* in *Criteria* cell of the CategoryName field.
- Save the query with name *DetailsOfLaptops*.

Now, suppose you are asked to retrieve the details of smart phones. You need to design a new query with *Smart Phone* as criteria in the CategoryName field and save it as DetailsofSmartphones. What if you are asked the same question for all the product categories? You need multiple queries!!!.

Base has one more interesting and very useful feature to help us in this type of situations. We can create a parameter query. To create a Parameter Query, the design of the query will remain same, but we need to enter the parameter, rather than specifying the actual value in the *Criterion* cell. When we run the query, Base will display a dialog box that will prompt us to enter the value of the parameter specified. Perform the following steps to create a parameter query for listing out the different product categories.

- Open a new query in Design View.
- Add the Product and ProductCategory tables.
- Double click on Pcode from the Product table.
- Double click on the CategoryName field from the Category table.
- Type *:CategoryName* in the *Criterion* cell of the CategoryName. The query will look as shown in figure 11.27. Note that the criterion parameter must be preceded by a colon symbol (:).
- Click on the Run button to view query results. Base will display dialog box as shown in figure 11.28.
- Type *Laptop* in the text box under label *Value* and click on OK button. You will get the list of laptops.


ModernElectronicStore.odb : parameter - OpenOffice.org Base: Query Design

Field	Pcode	CategoryName					
Alias							
Table	Product	ProductCategory					
Sort							
Visible	✓	✓	☐	☐	☐	☐	☐
Function							
Criterion		:CategoryName					
Or							

Figure 11.27 : Parameterized Query

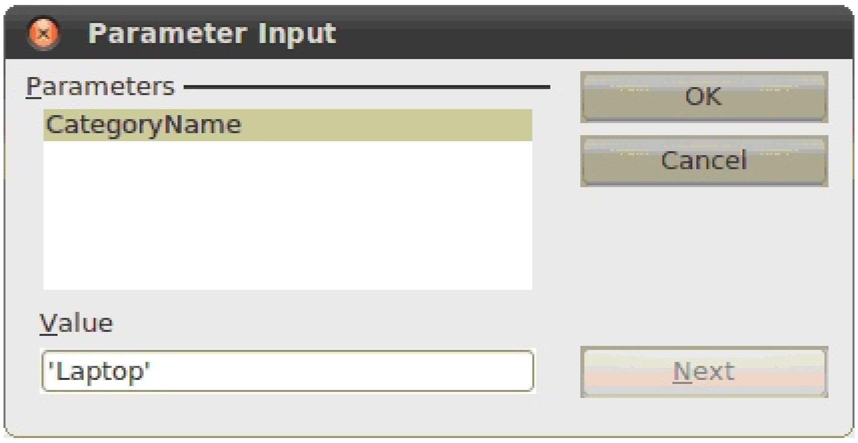


Figure 11.28 : Parameter value

Try to execute this query again with different values of the product categories and observe the output.

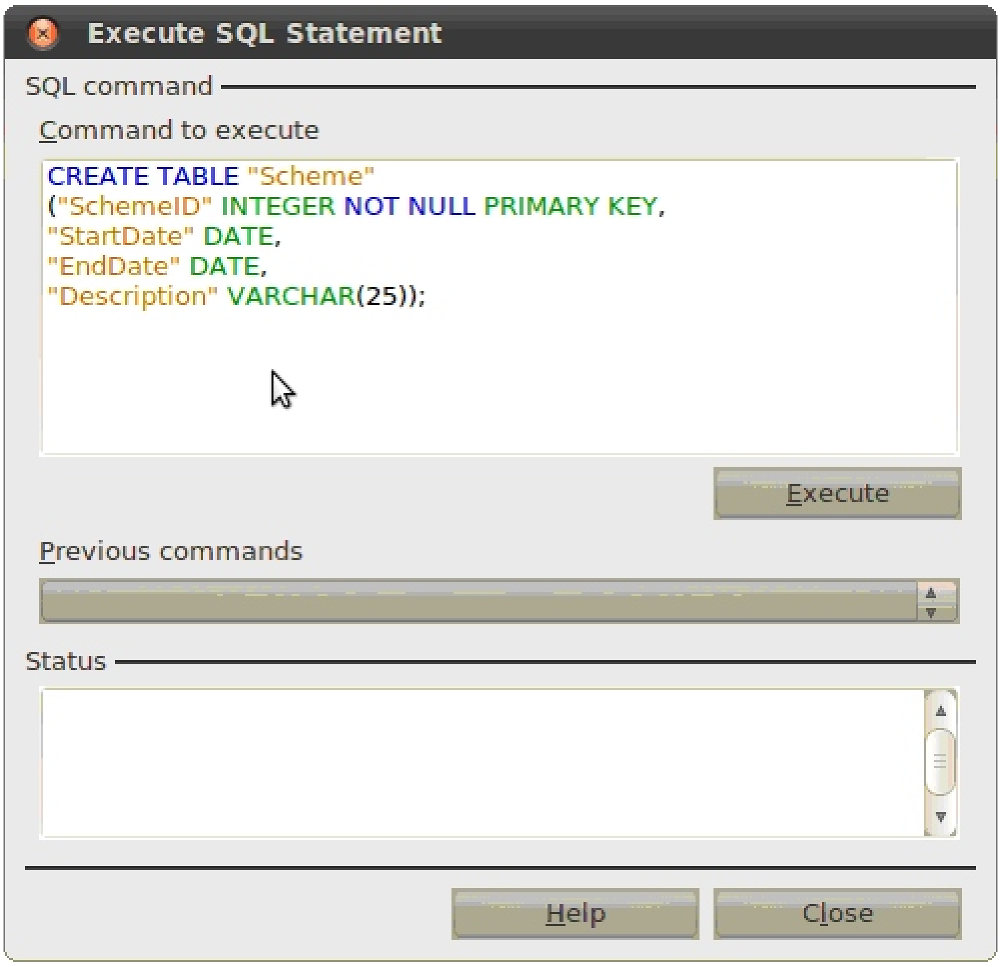
Structured Query Language

By now we have seen that there is always more than one way to do the same task in Base. For example, *Create Table in Design View...* and *Use Wizard to Create Table...* both options allow us to create table of a database. The difference is that the wizard makes the task easier while Design View gives more flexibility. We still have a third option for creating table; the SQL commands, which gives us the most flexibility and control.

SQL stands for Structured Query Language. It is a standard language used to query a relational database. The SQL queries are in the form of statements. In earlier chapters we have seen how to create a table, insert a data into it, edit and delete the data in the tables. All these operations can also be done using SQL statements. Let us create a new table using the SQL statement.

Click on the Tools options in the menu bar and then select the *SQL...* option. The *Execute SQL Statement* dialog box will get opened with a cursor blinking in the text box under the label *Command to Execute*. We can type the instructions to perform different operations related to table or a query here. Let us try to create a table named Scheme that has four fields SchemeID, StartDate, EndDate and Description. Pay attention to quote signs, capitalization and syntax to avoid errors in execution of queries. Type the statement shown below in the text box under the label *Command to Execute*. (See figure 11.29)

```
CREATE TABLE “Scheme”
(“SchemeID” INTEGER NOT NULL PRIMARY KEY,
“StartDate” DATE,
“EndDate” DATE,
“Description” VARCHAR(25));
```



```
CREATE TABLE "Scheme"
("SchemeID" INTEGER NOT NULL PRIMARY KEY,
"StartDate" DATE,
"EndDate" DATE,
"Description" VARCHAR(25));
```

Figure 11.29 : SQL Command Window

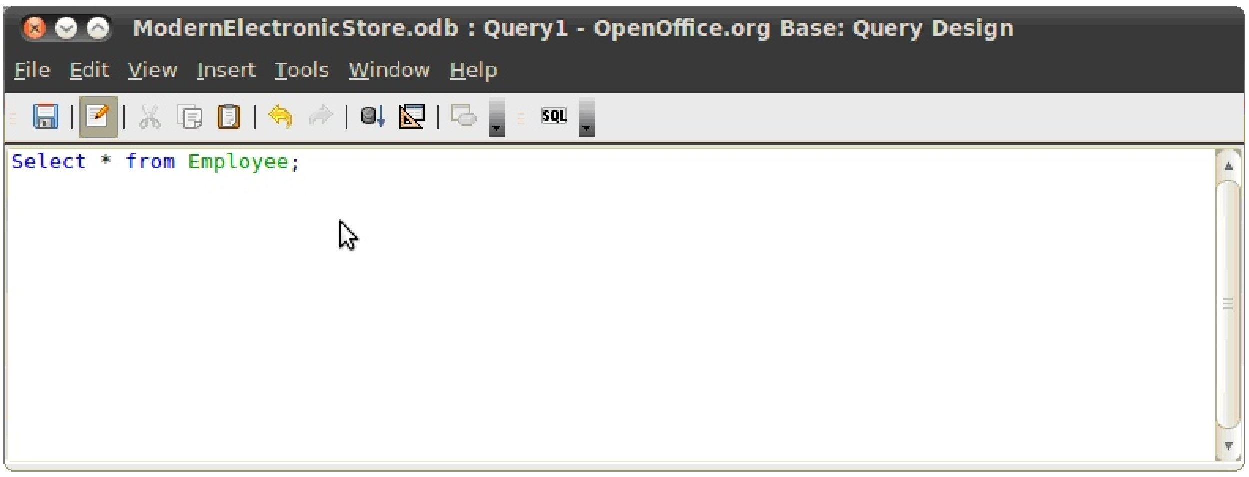
When you enter the instructions, the *Execute* button gets highlighted. When you are finished with writing the query, click on it. This will run the written SQL command and the table would be created. After a few seconds, the window will inform that the instructions have been executed. Other than that there will be no visible output on your screen. If you now go to the *View* menu and click on *Refresh Tables*, a complete list of the tables we have created will appear in the lower section of your screen including the new table recently created.

At times we may need to remove the table from a database. The DROP TABLE statement is used to remove a table. For example, if we want to drop the Scheme table created recently, then in the text box under the label *Command to Execute* write the following statement:

DROP TABLE Scheme IF EXISTS;

Now click on the Execute button and you will see that the Scheme table is removed from the database.

We can use SQL statement to retrieve information from the table. But we need to use different Window for this purpose. To open this Window click Queries icon. Three options will appear in the top panel under the label *Tasks.* Select the *Create Query in SQL view...* option. This will open a dialog box with a blinking cursor as shown in figure 11.30. We can now type the required SQL statement here to retrieve the information. The SQL statements to retrieve information start with SELECT keyword and are also known as SQL Queries. Figure 11.30 shows an example of SQL query.


ModernElectronicStore.odb : Query1 - OpenOffice.org Base: Query Design
File Edit View Insert Tools Window Help
SQL

```
Select * from Employee;
```

Figure 11.30 : SQL Select Query Window

Let us try to get the output of the query **SELECT * from Employee** visible in figure 11.30.

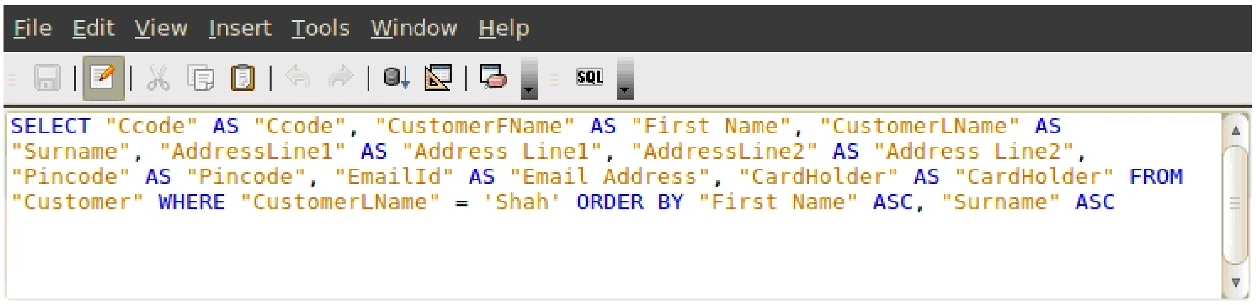
Click on Run Query button on the function bar at the top to view the result. You will find that complete detail of all the employees stored in the Employee table is displayed. To save the query, click on the Save button. Name the query as *EmployeeDetails* and close the Window.

Let us try to create some more SQL queries. Open new query in Design View. Select the *Create Query in SQL view...* option. Type the following statement:

SELECT FirstName, LastName from Employee;

Click on the Run Query button on the function bar at the top to view the result. Observe that the result now displays data for only two fields namely FirstName and LastName for all the employees.

Notice that the query created using SQL statement is also available under the Queries tab similar to queries created using Design View. Right click on any query and you will find an option *Edit in SQL View.* Click it to edit the query using the SQL statement. In fact an SQL statement for each query created as part of this chapter either, using wizard or design view can be viewed. For example, right click on the query CustomerList select *Edit in SQL View* and you will find query statement as seen in figure 11.31.



```
SELECT "Ccode" AS "Ccode", "CustomerFName" AS "First Name", "CustomerLName" AS
"Surname", "AddressLine1" AS "Address Line1", "AddressLine2" AS "Address Line2",
"Pincode" AS "Pincode", "EmailId" AS "Email Address", "CardHolder" AS "CardHolder" FROM
"Customer" WHERE "CustomerLName" = 'Shah' ORDER BY "First Name" ASC, "Surname" ASC
```

Figure 11.31 : SQL statement for Query designed using Design view

The words Ccode, First Name, Surname, Address Line1, Address Line2, Pincode, Email Address and CardHolder visible after the keyword AS are aliases. While the word Customer after keyword FROM is a table name. SQL statement in figure 11.31 also includes keyword WHERE that is used to retrieve the records based on criteria. The keyword ORDER BY is used to indicate that output needs to be sorted on field CustomerFName (represented by an alias First Name). The keyword ASC further mentions that data should be sorted in ascending order of first names.

Summary

Storing data of business is one aspect of DBMS. The important aspect is accurate retrieval of the same as and when needed. In this chapter we have discussed about how to retrieve information using queries. We learned how to create a query using a wizard, a query design view and SQL commands. We created simple queries as well as parameter query. A simple query does not ask for any input at run time, while a parameter query does ask for input at run time. Also the result of the parameter query can differ at every execution as the parameters entered might be different. In the next chapter we will use the queries that we have created in this chapter to create reports.

EXERCISE

1. What are Queries in Base? Why do we design Queries?
2. Explain the use of Criterion in a query.

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

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

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

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

7. Explain grouping giving a suitable example.

8. What are parameterized query?

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

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

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

(a) Table (b) Query

(c) Constraint (d) Relation

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

(a) Table (b) Query

(c) Form (d) Report

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

(a) Square root (b) Sum

(c) Min (d) Max

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

(a) Order by (b) Group by

(c) Arrange by (d) Set of

(5) Which of the following statement is false?

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

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

(c) Query cannot be created using wizard.

(d) Query cannot be written on a query.

(6) What is an alias?

(a) Alias is creation of duplicate query.

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

(c) Alias is used to print query.

(d) Alias is nowhere related to query.

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

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

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

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

(a) Like	(b) is
(c) equal	(d) =

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

(a) &	(b) +
(c) -	(d) *

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

(a) Select Query	(b) Insert Query
(c) Parameter Query	(d) Update query

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

(a) Comma	(b) Colon
(c) Question mark	(d) Explanation mark

(12) SQL stands for?

(a) Simple Query Language

(b) Structured Query Language

(c) Simple Question for large databases

(d) Structured queries for large databases

LABORATORY EXERCISES

1. Solve the following queries using the Student database created by us as part of exercise of chapter 9 and Chapter 10.

(a) List details of all the students studying in tenth standard.

(b) List names and address of students who have left the school.

(c) List teachers belonging to the city of Ahmedabad or Surat.

(d) List total number of subjects taught in the school.

(e) List total presence of student with Grno 10 in January 2012.

(f) How many subjects does Mr. Akhil Mehta teach in the school? Display it along with the standard that he teaches in.

(g) Calculate Percentage of Student with Grno 1 in October, 2011 in first term.

(h) Display the result of first term test conducted during October 2011 in the subject of Maths for each student.

(i) Create a parameterized query to accept city as a parameter and display students belonging to that city.

2. Design the following tables. Insert ten appropriate records in each table.

Student(StudentId , Name , Branch , Institute)

Exam (CourseNo , CourseName , DateofExam)

Appeared(StudentId , CourseNo)

Solve the following queries :

(a) List the details of exam conducted for course number 8 or 12.

(b) List the student id, his name and the course in which he appeared for exam.

(c) List the name of all the students who study in "Satyam" institute.

(d) Find total number of student registered in the course number 4.

(e) List the course name, date on which exam was conducted and names of the all the students who appeared in that exam.

(f) List the course number and name of the course whose exam is to be held on 12/2/2012.

(g) List the details of the exam whose course number is 8 or 10 and date of exam is 2/2/2012.

(h) List the branch of student whose name starts with alphabet A.

(i) Delete all the records of ABC Institute.

3. Design the following tables. Insert ten appropriate records in each table.

EMPLOYEE (EmpId , EmpName, Salary, Gender , Department, JoiningDate)

Solve the following Queries :

(a) List the details of the employee whose name starts with alphabet D.

(b) List the details of the employee whose salary is between Rs. 1000 and Rs. 3000.

(c) List the details of all the male employees.

(d) List the details of the employee who are in marketing department.

(e) Find Average salary distributed in the company.

(f) List the details of the employee whose salary is greater than Rs. 5000.

(g) List the details of the employee whose joining date is before 01/01/2012.

(h) List the details of the employee who belongs to either marketing or finance department.

(i) List the details of the employee who are not working in the purchase department.

(j) List the details of the employee who have joined after 10/09/2011 and working in finance department.