CHAPTER 1

ADVANCED SQL QUERIES

LEARNING OBJECTIVES

- ■ Writing Advanced Queries
    - ❑ Creating Sub-queries
    - ❑ Creating Unmatched and Duplicate Queries
    - ❑ Grouping and Summarizing Records using Criteria
    - ❑ Summarizing Data using Crosstab Query
    - ❑ Creating a Pivot Table and a Pivot Chart
- ■ Joining Tables in Queries
- ■ Calculated Fields

1.1 Introduction

Queries are an essential part of database. They are used to extract required data from one or more tables and present the result in a datasheet or on a Form or Report. They can be considered as questions asked to a table in a database. Access 2010 provides a graphical tool known as Query Designer to create queries. Queries are not only used to retrieve data from tables, but can also be used to insert, update or append data in tables, to filter data, to perform calculations with data, to summarize data and to automate data management tasks.

This chapter focuses on creating different types of queries and their different usage. The first part of the chapter talks about Sub-queries, in which the criterion for a query is query itself, i.e. writing a query within query. The next part of the chapter talks about working with *Query Wizard* and learning how to find unmatched data between two tables and to find duplicate rows within a table. We will also learn to create a summarised result out of queries using the crosstab queries. Crosstab queries can be taken as a tabular format of Pivot tables. This chapter also introduces the multiple views of the query and how to get query results in the form of Pivot Tables or Pivot Charts.

In addition to the above queries, we will also learn to create a query on multiple tables, specify different join types to retrieve the desired result. We can create joins between the tables using relationship window provided by Access 2010, or we can create joins at the query design window. Access 2010 also has many functions and operators which can be used to create expressions. This chapter discusses how to use these functions to create a new column from an existing column. The chapter also highlights some of the advanced queries available in Access 2010 and illustrates their significance using case studies.

1.2 Writing Advanced Queries

Access 2010 has the capability to handle far advanced queries than the simple criteria based queries. The queries in Access 2010 not only allow users to change the existing data, but also to generate summary reports. This section discusses few advanced queries handled by Access 2010, such as *Sub-queries, Crosstab Queries, Creating Unmatched* and *Duplicate Queries*, and also creating *PivotTable* and *PivotChart* through queries.

1.2.1 Creating Sub-queries

A *sub-query* is a query nested inside another query. We can use Sub-queries within **Select** query, **Action** query or within other Sub-queries. Sub-queries in **Select** statement can be the part of **Where** clause or **having** clause in **Group By** query. For *Action* queries, we can use Sub-queries to change the records which match some values in other tables.

Fig. 1.2.1 shows an example of a sub-query in a **Select** statement.

```
SELECT * FROM Products
WHERE ProductID IN
(SELECT ProductID FROM OrderDetails WHERE Discount >= .25);
```

Fig. 1.2.1. Subquery Example

This query retrieves all the columns from table *Products* based on the criteria that the discount on the products in the *OrderDetails* table must be greater than or equal to 25%. Notice that to present the criteria of discount for Products, we have used a sub-query in the WHERE clause.

Points to be noted while writing a sub-query:

- The sub-query must always be written as an SQL statement.
- The sub-query is always written in brackets ( ).
- If we are using the same table for the main query and sub-query, we need to provide aliases (alternate names) to the tables.
- If a sub-query is returning more than one row, the IN, ANY, ALL or EXISTS clause should be used in the WHERE statement.

1.2.1.1 Problem Scenario

Rohit is an Accountant in Apex Ltd. At the closing of the quarter, he has to check the Invoices for which the payments are made in Quarter-I of year 2008.

Solution

For the purpose of solving the above requirement, a query is created to retrieve all records from *Invoices* table for which the *Payment Date* in *Payments* table lies in Quarter-I i.e. between 1/1/2008 and 4/30/2008. Since we do not require any details from the *Payments* table, we use the table in a sub-query.

Steps for retrieving records from table INVOICES

1. Open the **Show Table** dialog box, by clicking on **Create** -> **Queries** -> **Query Design**.
2. Select the *Invoice* table from **Show Table** window and click **Add**, as shown in Fig. 1.2.2.

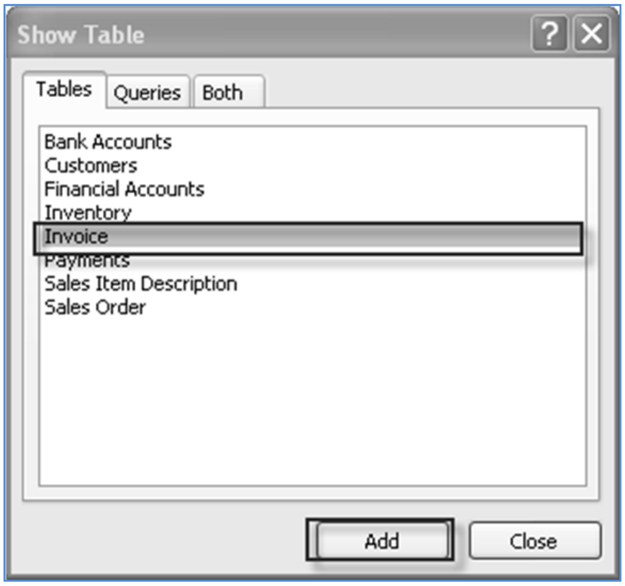


Fig. 1.2.2: Show Table window

3. Click **Close**.
4. The *Query Designer Window* appears, as shown in Fig. 1.2.3.

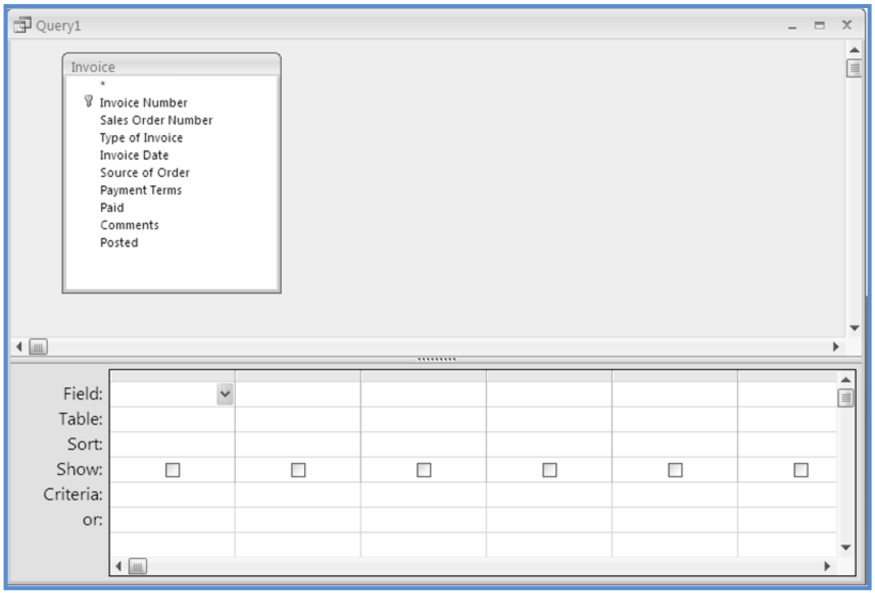
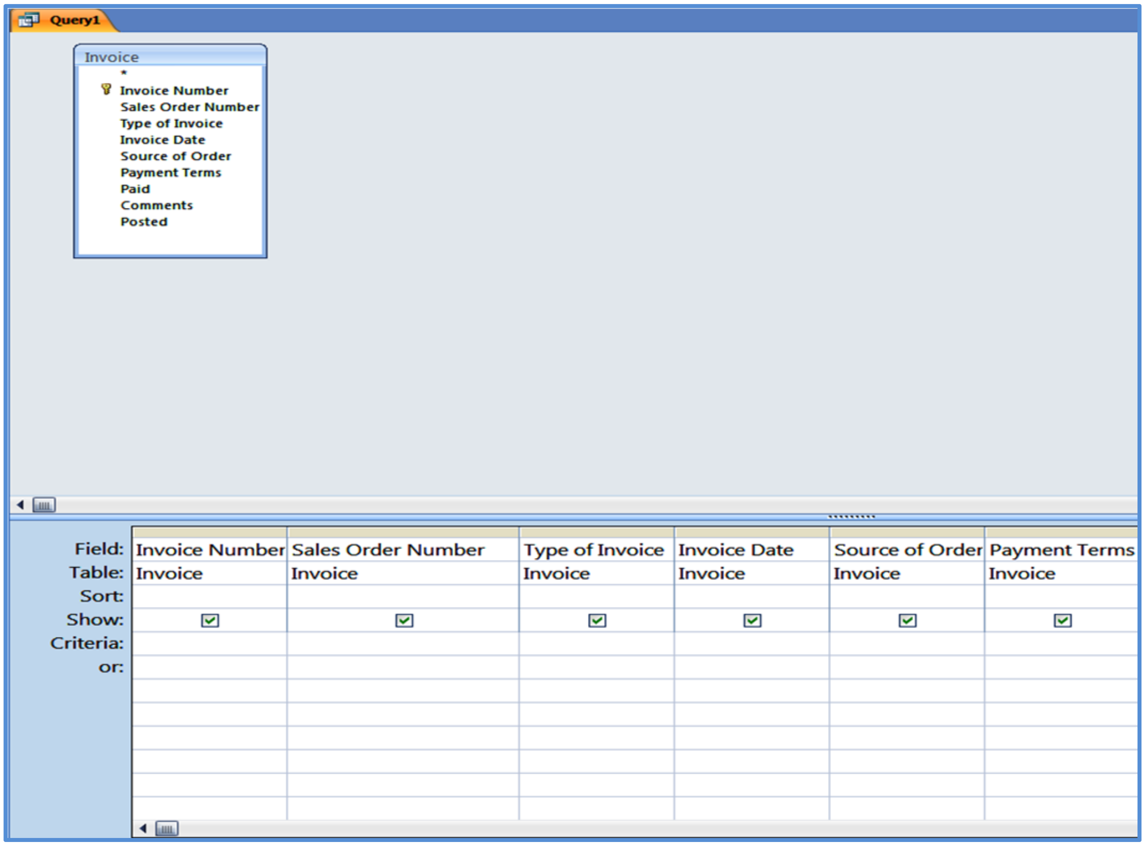


Fig. 1.2.3: Query Designer Window

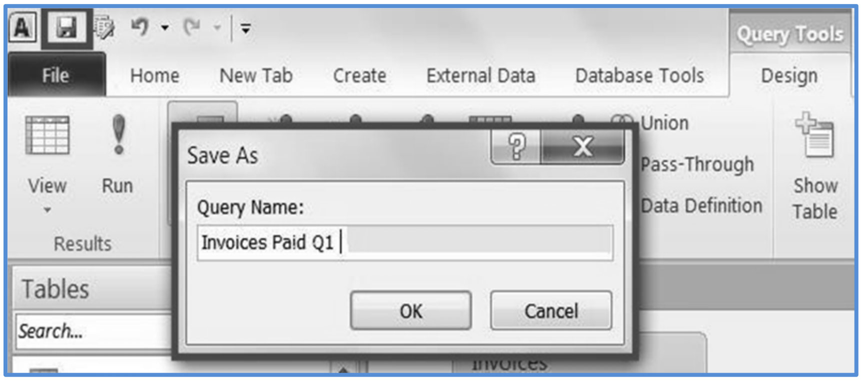
5. Select the fields *Invoice Number*, *Sales Order Number*, *Type of Invoice*, *Invoice Date*, *Source of Order* and *Payment Terms* from table *Invoice* and drag them to the Columns tab as shown in Fig. 1.2.4.



Field:	Invoice Number	Sales Order Number	Type of Invoice	Invoice Date	Source of Order	Payment Terms
Table:	Invoice	Invoice	Invoice	Invoice	Invoice	Invoice
Sort:						
Show:	☑	☑	☑	☑	☑	☑
Criteria:						
or:						

Fig. 1.2.4: Drag the required columns

6. Click the **Save** button at *Quick Access Toolbar* to save the query. Type the name of query as *Invoices Paid Q1* as shown in Fig. 1.2.5.



Save As

Query Name:

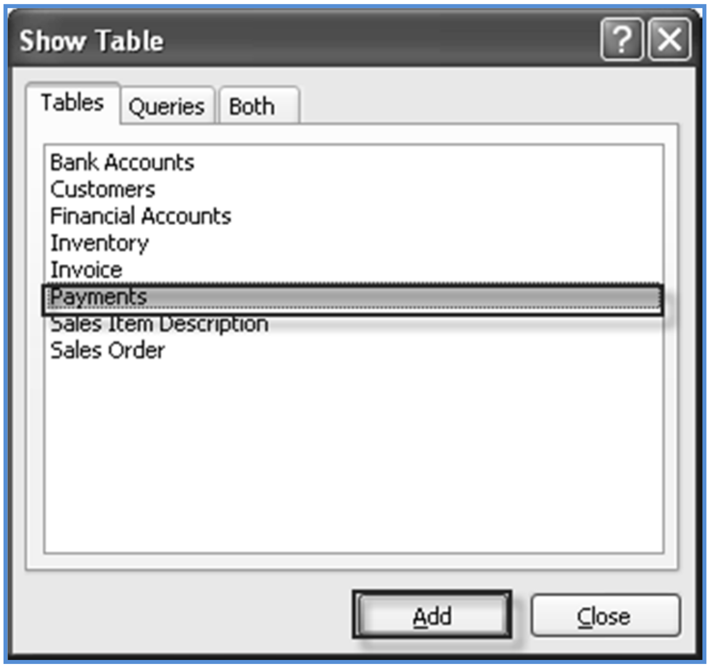
Invoices Paid Q1

Fig. 1.2.5: Save the Query

Now we create a SQL Statement for *sub-query*. The SQL Statement for the *sub-query* can be written directly in the criteria clause with the *Select* syntax or we may create another query using *Query Designer* and then copy the generated SQL. For the current scenario, we use the latter method.

Steps for writing a query to retrieve Invoice Number from Payments where Paid Date is in Quarter-I of year 2008

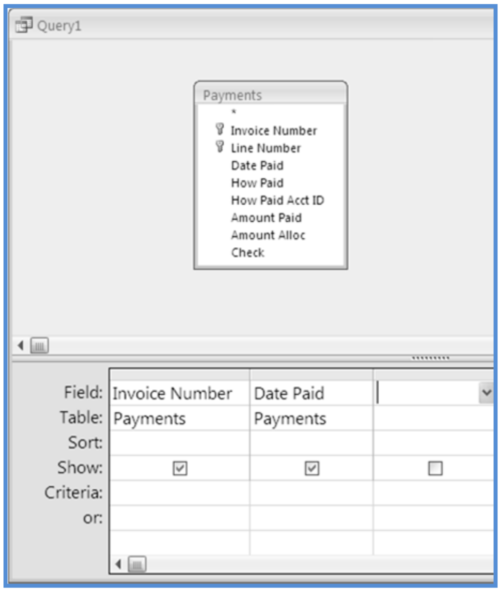
1. Open the **Show Table** dialog box, by clicking on **Create** -> **Queries** -> **Query Design**.
2. Select the *Payments* table from **Show Table** window and click **Add**, as shown in Fig. 1.2.6.



Bank Accounts
Customers
Financial Accounts
Inventory
Invoice
Payments
Sales Item Description
Sales Order

Fig. 1.2.6: Show Table window

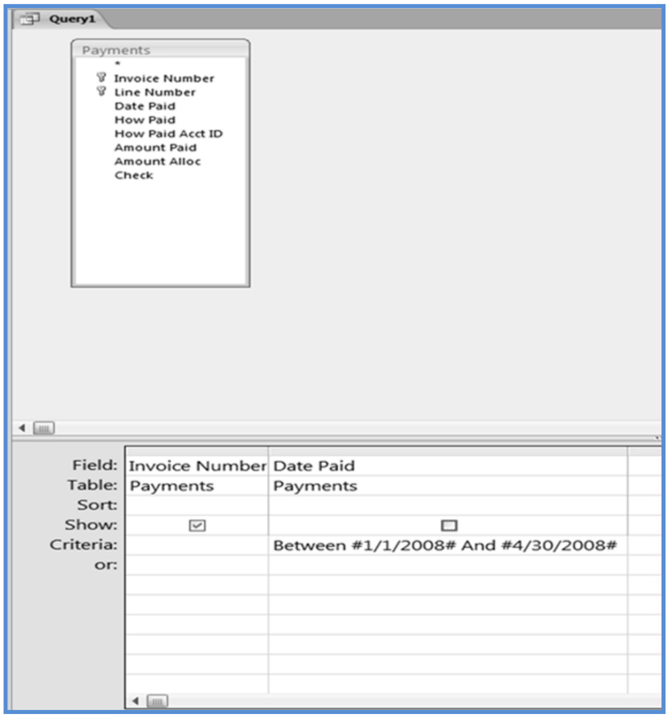
3. Click **Close**.
4. Double-click the columns *Invoice Number* and *Date Paid* so that they appear in the *Columns* tab in *Query Design* as shown in Fig. 1.2.7.



Field:	Invoice Number	Date Paid	
Table:	Payments	Payments	
Sort:			
Show:	☑	☑	□
Criteria:			
or:			

Fig. 1.2.7: Select Required columns

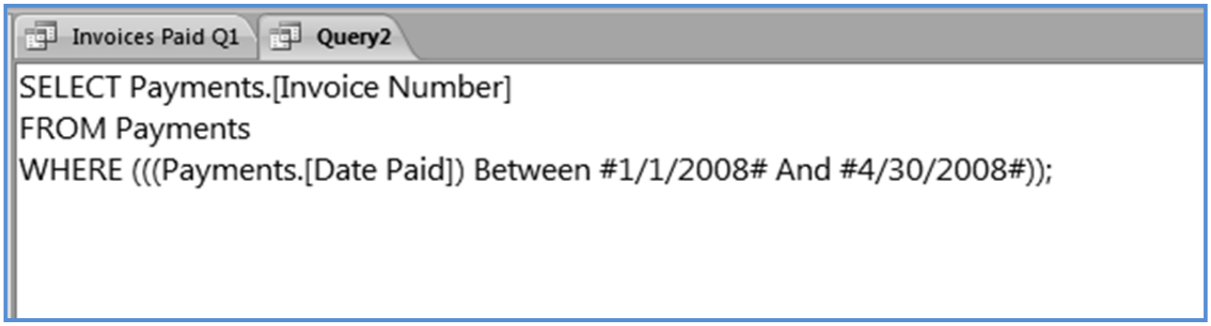
5. In the **Criteria** section of the *Date Paid* Column, write the criteria *BETWEEN #1/1/2008# AND #4/30/2008#* and clear the **Show** checkbox. The *Query Design* window should appear, as shown in Fig. 1.2.8.



Field:	Invoice Number	Date Paid
Table:	Payments	Payments
Sort:		
Show:	☑	□
Criteria:		Between #1/1/2008# And #4/30/2008#
or:		

Fig. 1.2.8: Specify the criteria

6. Click the **Run** icon Run in **Design** -> **Results** to view that the proper result is coming.
7. Click on **SQL View** from **Design** -> **Results** -> **View** drop-down to view the SQL statement of the query. The SQL statement appears for the query, as shown in Fig. 1.2.9.



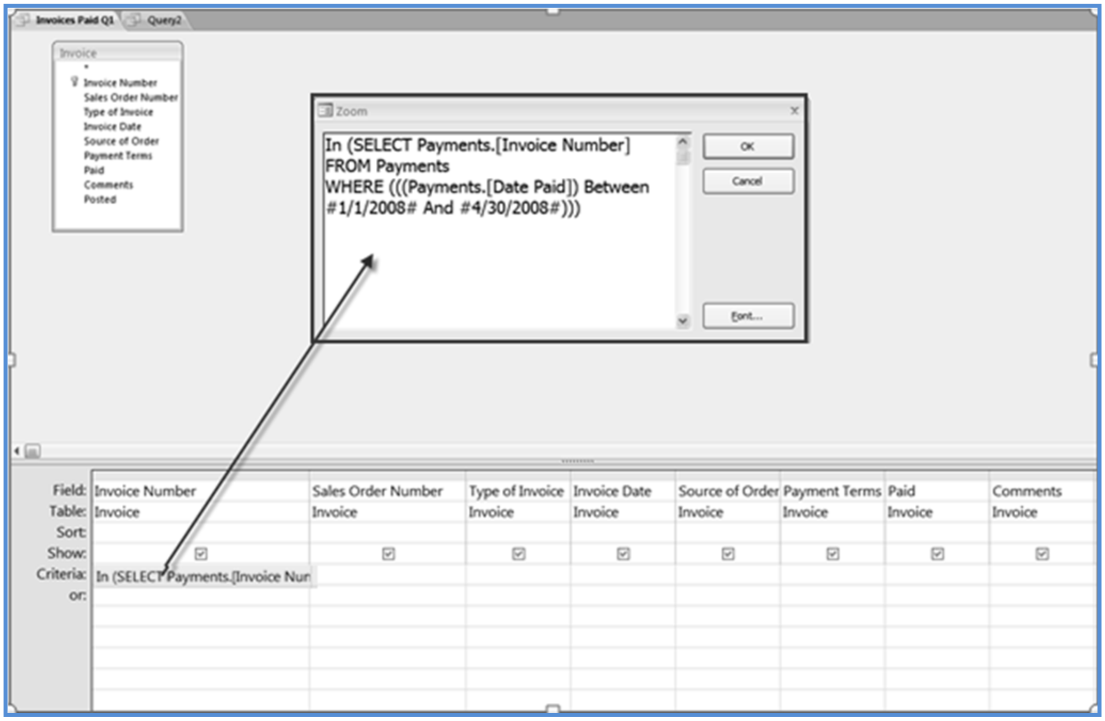
```
SELECT Payments.[Invoice Number]
FROM Payments
WHERE (((Payments.[Date Paid]) Between #1/1/2008# And #4/30/2008#));
```

Fig. 1.2.9: Query SQL View

8. Copy the SQL statement and switch to the *Query* window of Invoices Paid Q1.

Steps to add a sub-query to Invoices Paid Q1 query

9. In the **Criteria** tab of *Invoice Number* column, type IN() and paste the copied query within the braces. The Query window appears, as shown in Fig. 1.2.10.



```
In (SELECT Payments.[Invoice Number]
FROM Payments
WHERE (((Payments.[Date Paid]) Between
#1/1/2008# And #4/30/2008#)))
```

Field:	Invoice Number	Sales Order Number	Type of Invoice	Invoice Date	Source of Order	Payment Terms	Paid	Comments
Table:	Invoice	Invoice	Invoice	Invoice	Invoice	Invoice	Invoice	Invoice
Sort:								
Show:	☑	☑	☑	☑	☑	☑	☑	☑
Criteria:	In (SELECT Payments.[Invoice Nur							
or:								

Fig. 1.2.10: Write subquery in Criteria

10. Click the **Run** icon Run in **Design** -> **Results** to view that the proper result is coming.

In a similar manner as demonstrated by an example above, Sub-queries can be used as a SQL statement for criteria of *Action* queries, *Group By* queries etc.

NOTE: We can also use Sub-queries as an expression to create a new column as a query result.

1.2.2 Creating Unmatched and Duplicate Queries

Query Wizard available in Access 2010 can be used to create different types of queries. Along with several different queries, it also provides queries to find duplicate records in a table on the basis of one or more fields in a table. In a similar manner, we can create *Unmatched Query* to compare two tables and find the records that do not have matching values in given columns.

1.2.2.1 Unmatched Queries

Find Unmatched Records Query will examine the data found in two different tables/queries and compare the records based on a common field. It will return the records from the first table which do not have matching values in second table.

The easiest way to create *Unmatched Query* is by using the *Find Unmatched Query Wizard*. After the wizard builds query, we can modify the query's design to add or remove fields, or to modify joins as required.

Unmatched Query Wizard in Access 2010 asks for the names of the two tables to compare, and common field name between the tables. The wizard then confirms the fields that we wish to retrieve from the first table as a query result. Finally, the wizard prompts to enter the query name and creates the query.

This type of query can help find records that have no corresponding records in other tables. For example, we may be looking for products that have not been sold in any order or may be for customers who have not placed any orders.

1.2.2.2 Problem Scenario

Ankur Mathur, Sales Head of Apex Ltd. wishes to reduce the production of few products. To analyse which products should not be produced further, he needs to find out the products which have not been ordered so far.

Solution

For the purpose of finding the unmatched products in *Sales Item Description*, we use *Unmatched Query Wizard* and find out products from *Inventory* table which do not have a matching *Item Number* in *Sales Item Description* table.

Steps for creating Unmatched Query using *Query Wizard*:

1. Open the *Query Wizard* by selecting **Query Wizard** from **Create** -> **Queries**, as indicated in Fig. 1.2.11.

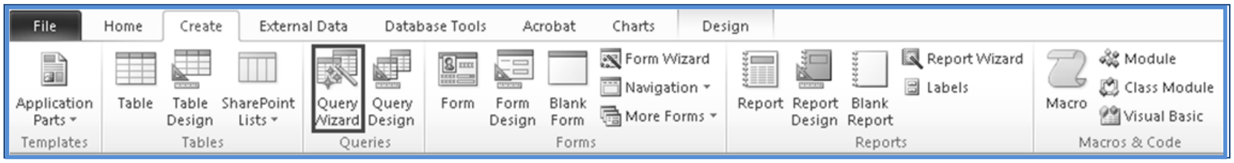
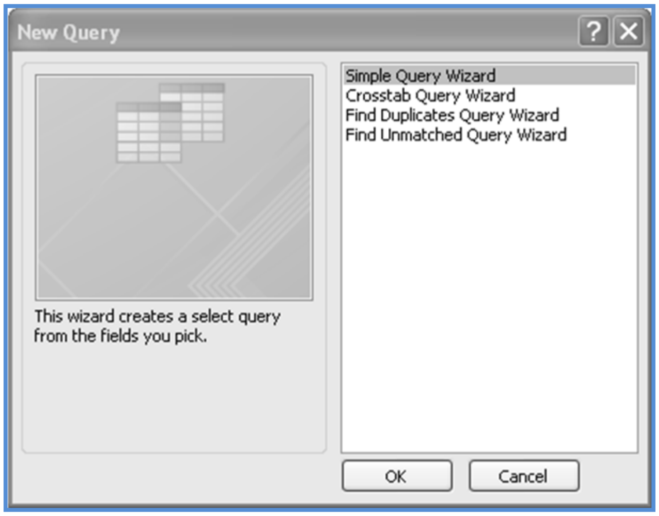


Fig. 1.2.11: Open Query Wizard

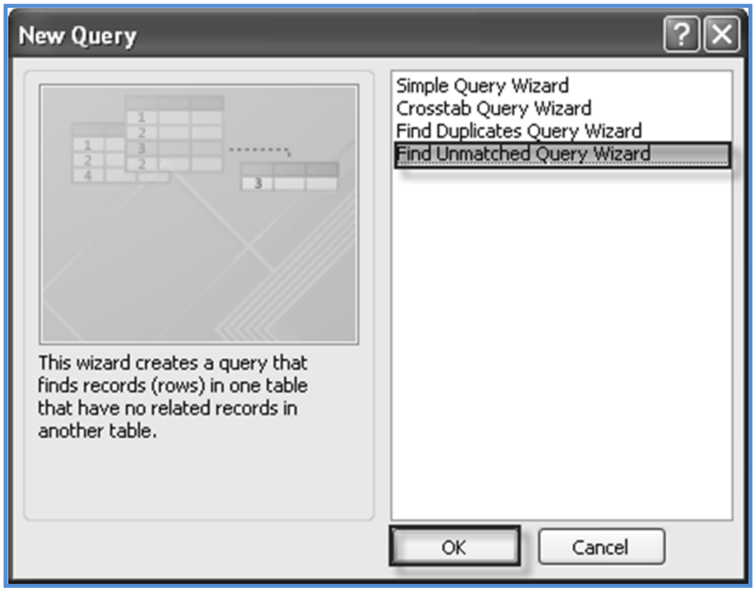
2. The New **Query** window appears, as shown in Fig. 1.2.12.



This wizard creates a select query from the fields you pick.

Fig. 1.2.12: New Query Dialog Box

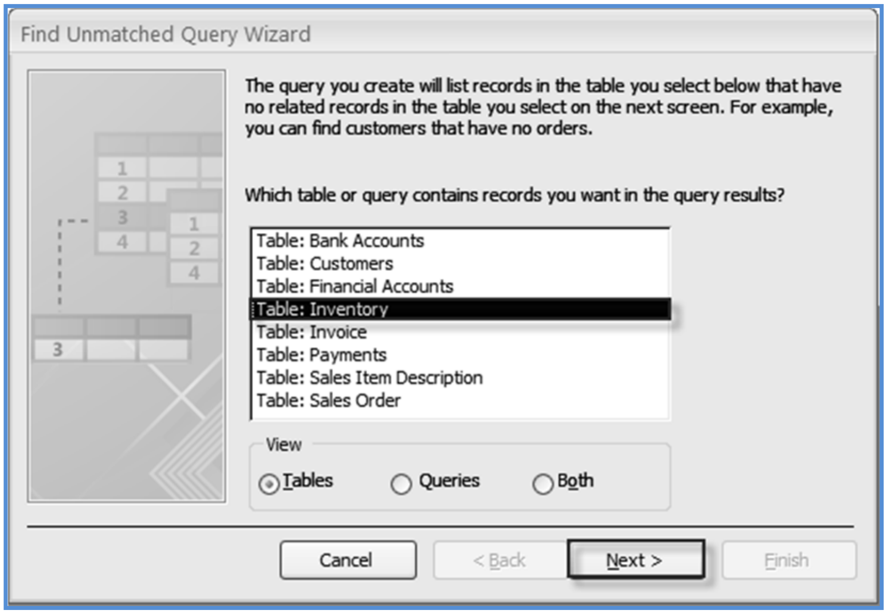
3. Select **Find Unmatched Query Wizard** from the **New Query** window and click **OK**, as shown in Fig. 1.2.13.



This wizard creates a query that finds records (rows) in one table that have no related records in another table.

Fig. 1.2.13: Select Find Unmatched Query Wizard

4. The **Find Unmatched Query Wizard** appears. Select the table *Inventory* and click **Next**, as shown in Fig. 1.2.14.



Find Unmatched Query Wizard

The query you create will list records in the table you select below that have no related records in the table you select on the next screen. For example, you can find customers that have no orders.

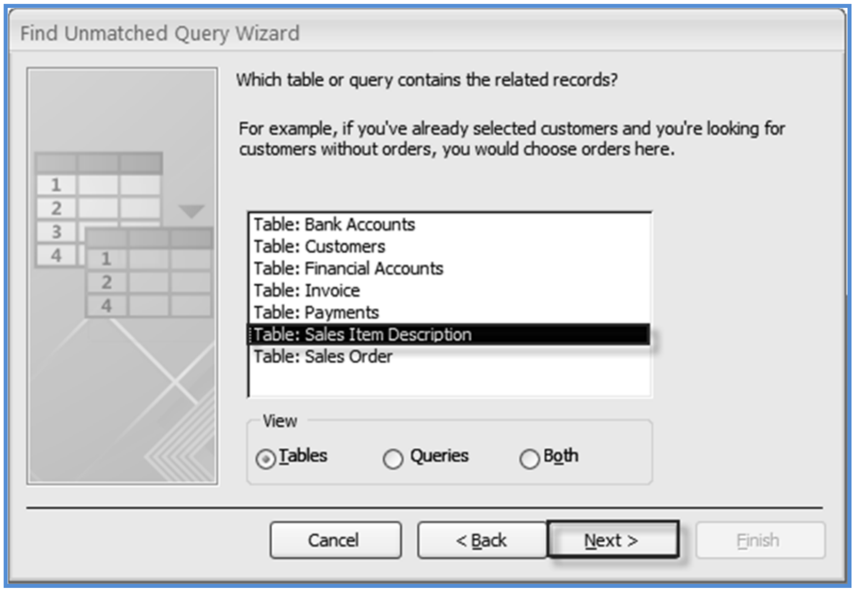
Which table or query contains records you want in the query results?

- Table: Bank Accounts
- Table: Customers
- Table: Financial Accounts
- **Table: Inventory**
- Table: Invoice
- Table: Payments
- Table: Sales Item Description
- Table: Sales Order

View: ● Tables ○ Queries ○ Both

Fig. 1.2.14: Select Table Inventory

5. Select the table *Sales Item Description* from second page of the wizard as we need to compare *Inventory* table to *Sales Item Description* table, as shown in Fig. 1.2.15. Click **Next**.



Find Unmatched Query Wizard

Which table or query contains the related records?

For example, if you've already selected customers and you're looking for customers without orders, you would choose orders here.

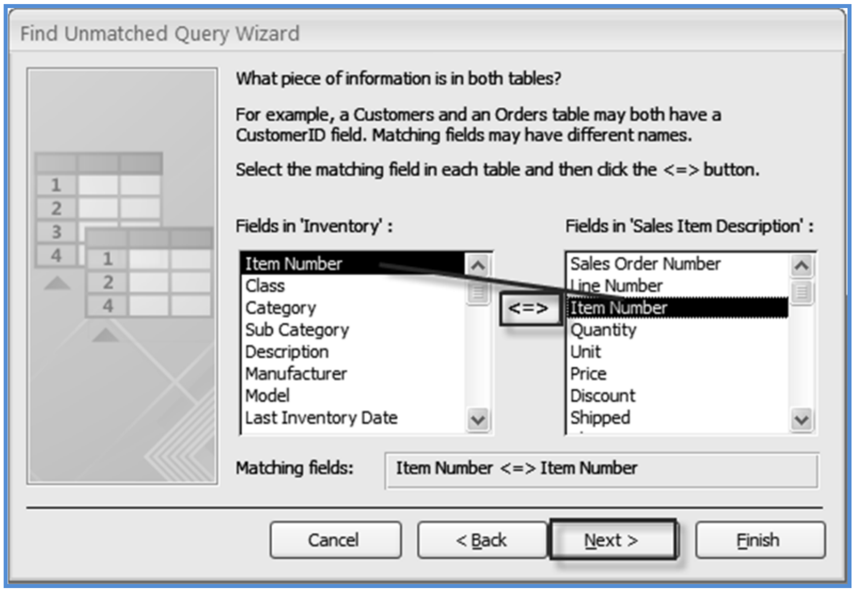
- Table: Bank Accounts
- Table: Customers
- Table: Financial Accounts
- Table: Invoice
- Table: Payments
- **Table: Sales Item Description**
- Table: Sales Order

View: ● Tables ○ Queries ○ Both

Fig. 1.2.15: Select Table Sales Item Description

6. Now, we need to mark the common field in both the tables to be compared. For this example, select the *Item Number* in both *Inventory* and *Sales Item Description* table and click <=> button, as

shown in Fig. 1.2.16. Click **Next** to move to next page.



What piece of information is in both tables?

For example, a Customers and an Orders table may both have a CustomerID field. Matching fields may have different names.

Select the matching field in each table and then click the <=> button.

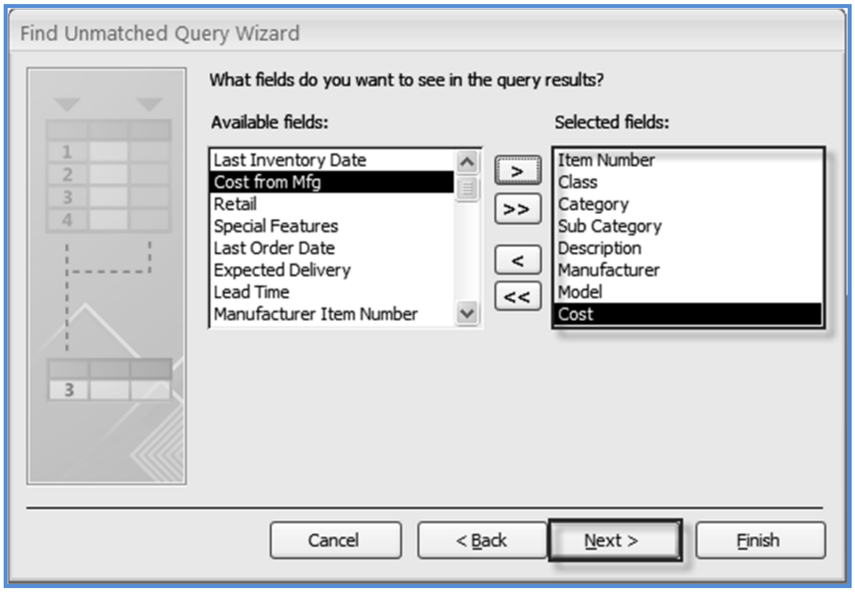
Fields in 'Inventory' : Item Number, Class, Category, Sub Category, Description, Manufacturer, Model, Last Inventory Date

Fields in 'Sales Item Description' : Sales Order Number, Line Number, Item Number, Quantity, Unit, Price, Discount, Shipped

Matching fields: Item Number <=> Item Number

Fig. 1.2.16: Match the common fields between two tables

7. On the next page in wizard, select the columns that should be the part of query result and click **Next**. In this page, select *Item Number, Class, Category, Sub Category, Description, Manufacturer, Model* and *Cost* fields as shown in Fig. 1.2.17. Click **Next** to move to next page.



What fields do you want to see in the query results?

Available fields: Last Inventory Date, Cost from Mfg, Retail, Special Features, Last Order Date, Expected Delivery, Lead Time, Manufacturer Item Number

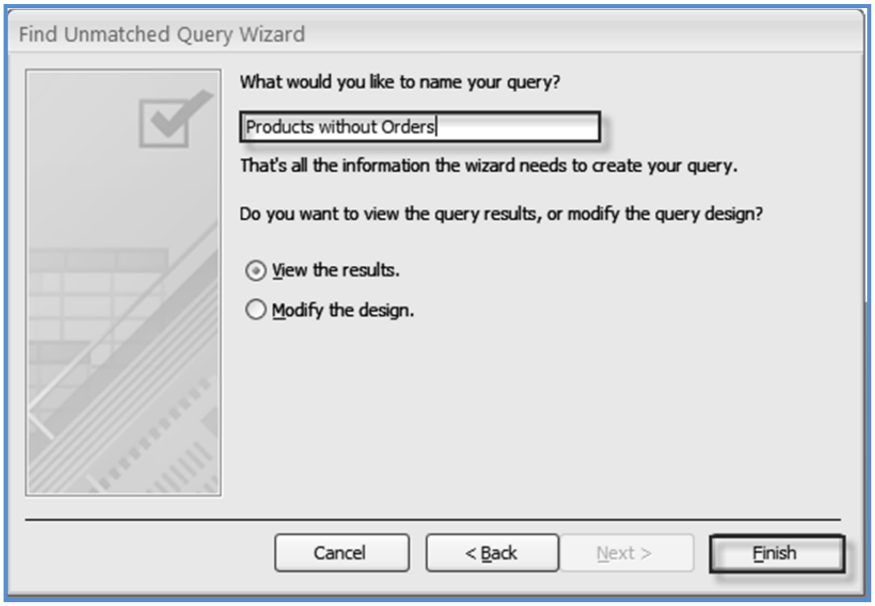
Selected fields: Item Number, Class, Category, Sub Category, Description, Manufacturer, Model, Cost

Fig. 1.2.17: Select the Fields

NOTE: Use > button to move a selected field from **Available fields** to **Selected fields**, >>

button to move all fields from **Available fields** to **Selected fields**, < button to move selected field from **Selected fields** to **Available fields**, << button to move all fields from **Selected fields** to **Available fields**.

8. Name the Query as "*Products without Orders*" in the final window that appears and click **Finish** as shown in Fig. 1.2.18.



What would you like to name your query?

Products without Orders

That's all the information the wizard needs to create your query.

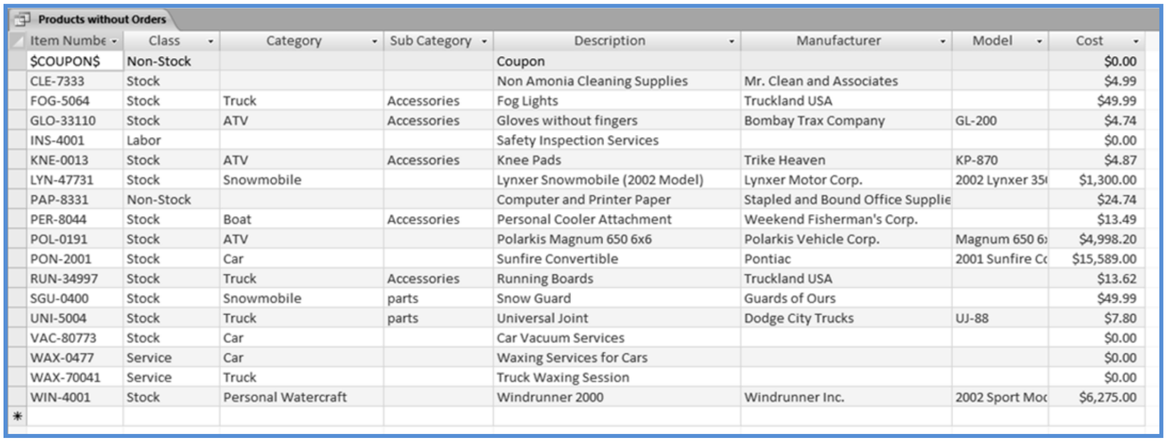
Do you want to view the query results, or modify the query design?

● View the results.

○ Modify the design.

Fig. 1.2.18: Name the query

9. The result of the query appears as in datasheet form, indicating the products that do not have an order associated with them, as shown in Fig. 1.2.19.



Item Numbe	Class	Category	Sub Category	Description	Manufacturer	Model	Cost
$COUPON$	Non-Stock			Coupon			$0.00
CLE-7333	Stock			Non Amonia Cleaning Supplies	Mr. Clean and Associates		$4.99
FOG-5064	Stock	Truck	Accessories	Fog Lights	Truckland USA		$49.99
GLO-33110	Stock	ATV	Accessories	Gloves without fingers	Bombay Trax Company	GL-200	$4.74
INS-4001	Labor			Safety Inspection Services			$0.00
KNE-0013	Stock	ATV	Accessories	Knee Pads	Trike Heaven	KP-870	$4.87
LYN-47731	Stock	Snowmobile		Lynxer Snowmobile (2002 Model)	Lynxer Motor Corp.	2002 Lynxer 35	$1,300.00
PAP-8331	Non-Stock			Computer and Printer Paper	Stapled and Bound Office Supplie		$24.74
PER-8044	Stock	Boat	Accessories	Personal Cooler Attachment	Weekend Fisherman's Corp.		$13.49
POL-0191	Stock	ATV		Polarkis Magnum 650 6x6	Polarkis Vehicle Corp.	Magnum 650 6	$4,998.20
PON-2001	Stock	Car		Sunfire Convertible	Pontiac	2001 Sunfire C	$15,589.00
RUN-34997	Stock	Truck	Accessories	Running Boards	Truckland USA		$13.62
SGU-0400	Stock	Snowmobile	parts	Snow Guard	Guards of Ours		$49.99
UNI-5004	Stock	Truck	parts	Universal Joint	Dodge City Trucks	UJ-88	$7.80
VAC-80773	Stock	Car		Car Vacuum Services			$0.00
WAX-0477	Service	Car		Waxing Services for Cars			$0.00
WAX-70041	Service	Truck		Truck Waxing Session			$0.00
WIN-4001	Stock	Personal Watercraft		Windrunner 2000	Windrunner Inc.	2002 Sport Mo	$6,275.00
*							

Fig. 1.2.19: Query Result

1.2.2.3 Duplicate Queries

The *Duplicate Queries* option creates a query that reports which records in a table are duplicated by matching one or more fields in the table. The *Query Wizard* first confirms which fields have to be used to check for duplication and then prompts to enter some other fields that may be a part of query result. Finally, Access 2010 accepts a name for the query and displays the results as a datasheet.

This type of query is useful when we have no unique indexes or primary key in the table, or the data for the table is imported from a source where we do not have mechanism to check duplicate values.

1.2.2.4 Problem Scenario

Varun Gupta, a Chartered Accountant in Apex Ltd., is required to audit the invoices and the payments. While tracking the payments he noticed that multiple invoices have been created for a single sales order. To sort out the things, he wishes to check all the sales order having duplicate invoices.

Solution

Create a *Find Duplicate Query* for table *Invoices* based on the field *Sales Order Number* so that it displays all the duplicate invoices created for a single sales order.

Steps for Finding Duplicate records in the table Invoices

1. Open the *Query Wizard* by selecting **Query Wizard** from **Create** -> **Queries**.
2. Select **Find Duplicates Query Wizard** from the **New Query** dialog box and click **OK***,* as shown in Fig. 1.2.20.

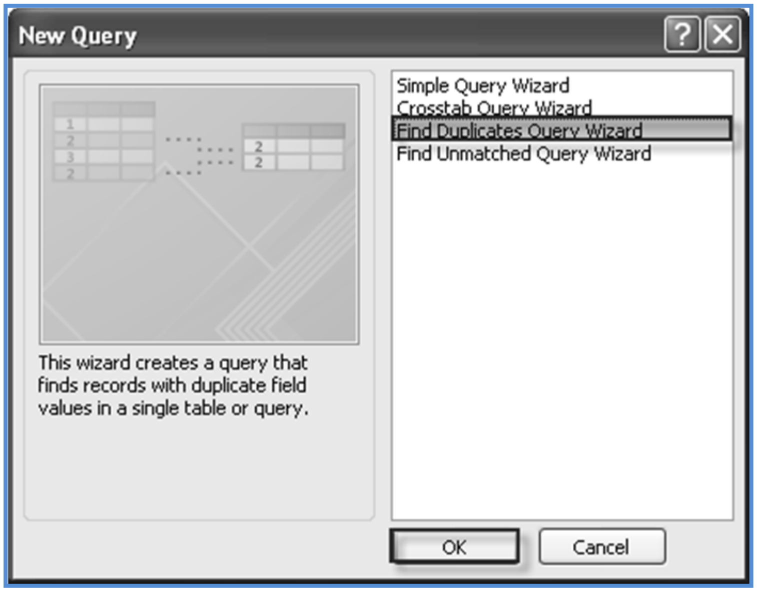
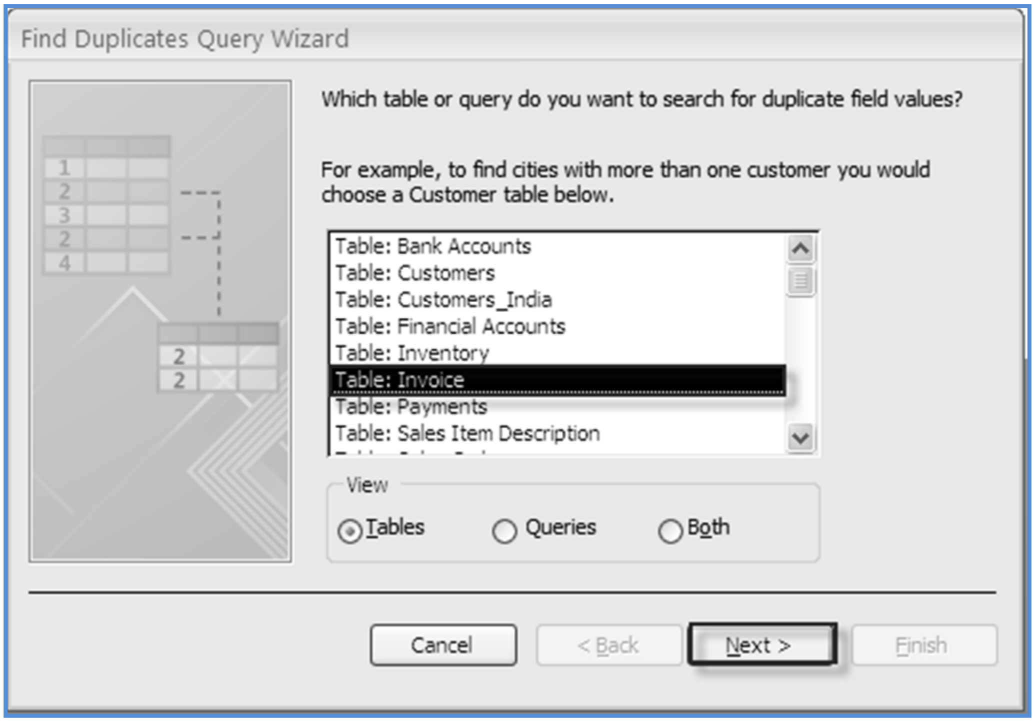

New Query
Simple Query Wizard
Crosstab Query Wizard
Find Duplicates Query Wizard
Find Unmatched Query Wizard
This wizard creates a query that finds records with duplicate field values in a single table or query.
OK
Cancel

Fig. 1.2.20: Select Find Duplicates Query Wizard

3. The **Find Duplicates Query Wizard** appears. Select the table *Invoice* and click **Next** as shown in Fig. 1.2.21.



Find Duplicates Query Wizard

Which table or query do you want to search for duplicate field values?

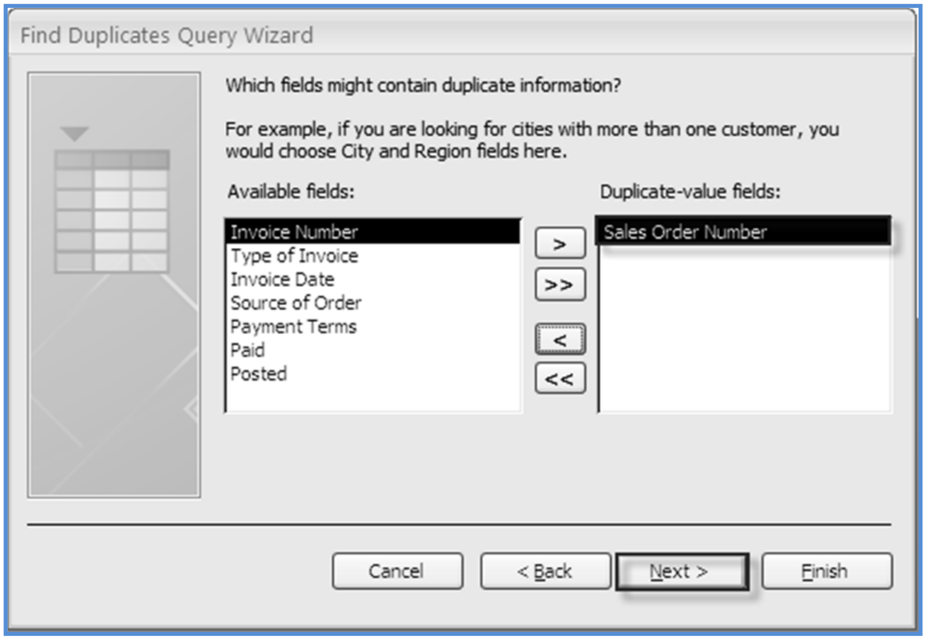
For example, to find cities with more than one customer you would choose a Customer table below.

- Table: Bank Accounts
- Table: Customers
- Table: Customers_India
- Table: Financial Accounts
- Table: Inventory
- Table: Invoice
- Table: Payments
- Table: Sales Item Description

View: ● Tables ○ Queries ○ Both

Fig. 1.2.21: Select Table Invoice

4. Select the column on the basis of which the duplicate records need to be matched and click **Next**. In this scenario, select *Sales Order Number* as shown in Fig. 1.2.22. Click **Next**.



Find Duplicates Query Wizard

Which fields might contain duplicate information?

For example, if you are looking for cities with more than one customer, you would choose City and Region fields here.

Available fields:

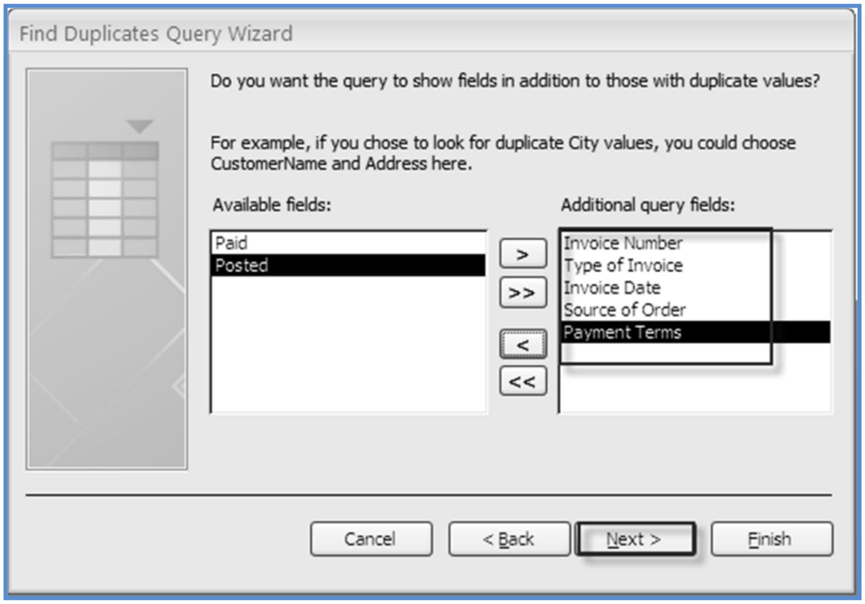
- Invoice Number
- Type of Invoice
- Invoice Date
- Source of Order
- Payment Terms
- Paid
- Posted

Duplicate-value fields:

- Sales Order Number

Fig. 1.2.22: Select the column for Duplicate values

5. Select the columns that should be the part of the query result and click **Next**. In this dialog box, select *Invoice Number*, *Type of Invoice*, *Invoice Date, Source of Order* and *Payments Terms*, as shown in Fig. 1.2.23. Click **Next**.



Find Duplicates Query Wizard

Do you want the query to show fields in addition to those with duplicate values?

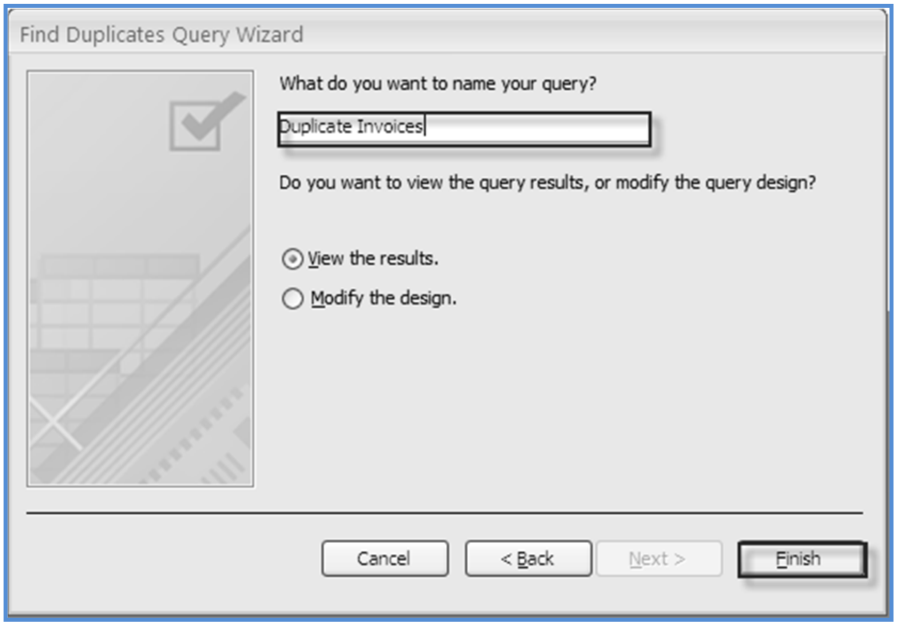
For example, if you chose to look for duplicate City values, you could choose CustomerName and Address here.

Available fields:
- Paid
- Posted

Additional query fields:
- Invoice Number
- Type of Invoice
- Invoice Date
- Source of Order
- Payment Terms

Fig. 1.2.23: Select the columns for Query Result

6. Name the Query as "*Duplicate Invoices*" in the final window that appears and click **Finish**, as shown in Fig. 1.2.24.



Find Duplicates Query Wizard

What do you want to name your query?

Duplicate Invoices

Do you want to view the query results, or modify the query design?

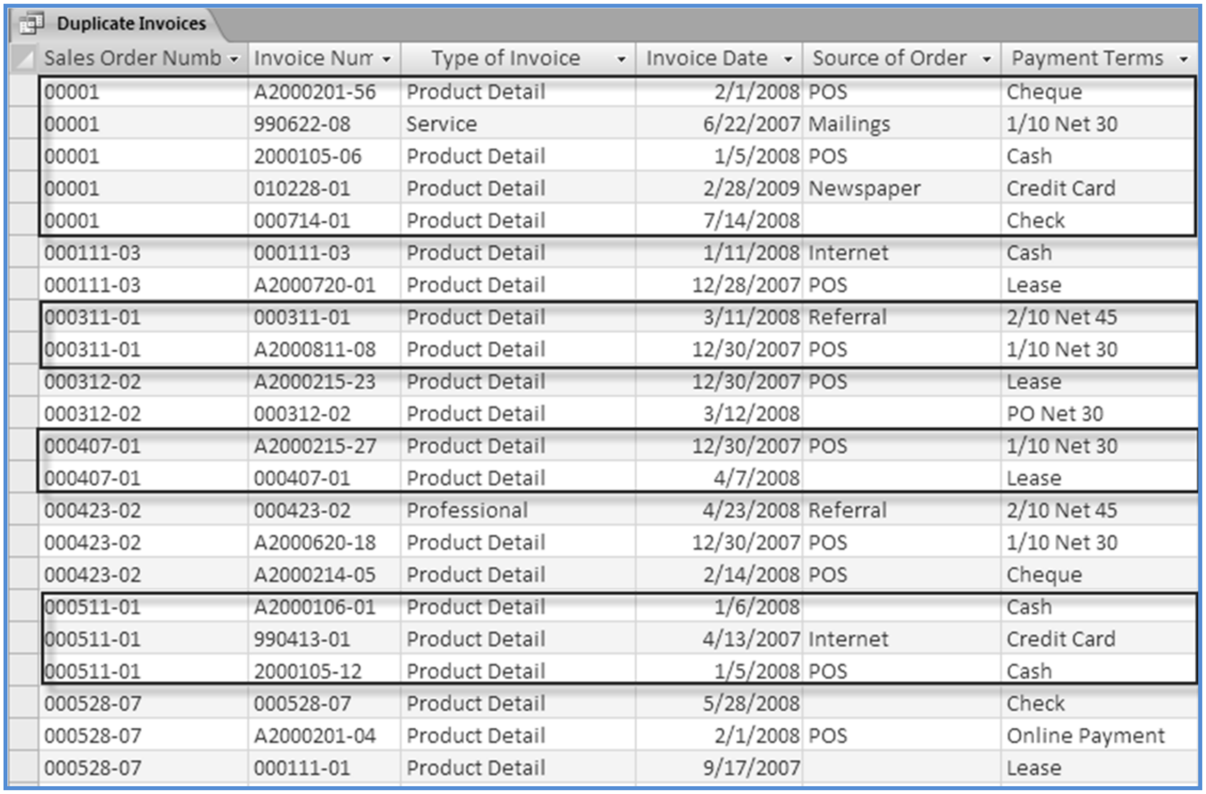
● View the results.

○ Modify the design.

Fig. 1.2.24: Name the Query

7. The result of the query appears in Datasheet form with all the Sales Order having multiple invoices, as shown in Fig. 1.2.25.

NOTE: We can create *Find Duplicate Query* by matching records on multiple fields also. This query can also be used to find duplicate records in a table by matching records on all the fields (considering that we can match only 10 fields at a time).



Duplicate Invoices

Sales Order Numb	Invoice Num	Type of Invoice	Invoice Date	Source of Order	Payment Terms
00001	A2000201-56	Product Detail	2/1/2008	POS	Cheque
00001	990622-08	Service	6/22/2007	Mailings	1/10 Net 30
00001	2000105-06	Product Detail	1/5/2008	POS	Cash
00001	010228-01	Product Detail	2/28/2009	Newspaper	Credit Card
00001	000714-01	Product Detail	7/14/2008		Check
000111-03	000111-03	Product Detail	1/11/2008	Internet	Cash
000111-03	A2000720-01	Product Detail	12/28/2007	POS	Lease
000311-01	000311-01	Product Detail	3/11/2008	Referral	2/10 Net 45
000311-01	A2000811-08	Product Detail	12/30/2007	POS	1/10 Net 30
000312-02	A2000215-23	Product Detail	12/30/2007	POS	Lease
000312-02	000312-02	Product Detail	3/12/2008		PO Net 30
000407-01	A2000215-27	Product Detail	12/30/2007	POS	1/10 Net 30
000407-01	000407-01	Product Detail	4/7/2008		Lease
000423-02	000423-02	Professional	4/23/2008	Referral	2/10 Net 45
000423-02	A2000620-18	Product Detail	12/30/2007	POS	1/10 Net 30
000423-02	A2000214-05	Product Detail	2/14/2008	POS	Cheque
000511-01	A2000106-01	Product Detail	1/6/2008		Cash
000511-01	990413-01	Product Detail	4/13/2007	Internet	Credit Card
000511-01	2000105-12	Product Detail	1/5/2008	POS	Cash
000528-07	000528-07	Product Detail	5/28/2008		Check
000528-07	A2000201-04	Product Detail	2/1/2008	POS	Online Payment
000528-07	000111-01	Product Detail	9/17/2007		Lease

Fig. 1.2.25: Query Result

1.2.3 Grouping and summarising Records using Criteria

Access 2010 provides queries that can be used for obtaining the aggregated results instead of individual records. These queries may be helpful in retrieving count of records, sum, average, and maximum or minimum of the values in a column. These queries are known as *Group By* queries or *Totals Query*. Such queries can return the aggregated results from the entire table or the records of the table filtered by a certain criteria.

1.2.3.1 Grouping and summarising Records

At times, it is required to retrieve information in tables based on a group of one or more fields. For example, total number of contacts from a particular area or a sum of all the payments received in a month. Access 2010 provides a method to obtain the desired result using queries without the need of any complex programming. It calculates the totals using several aggregate functions.

Access 2010 performs grouping of the records by using **Totals** option available in *the* **Query Tools** tab, as shown in Fig. 1.2.26.

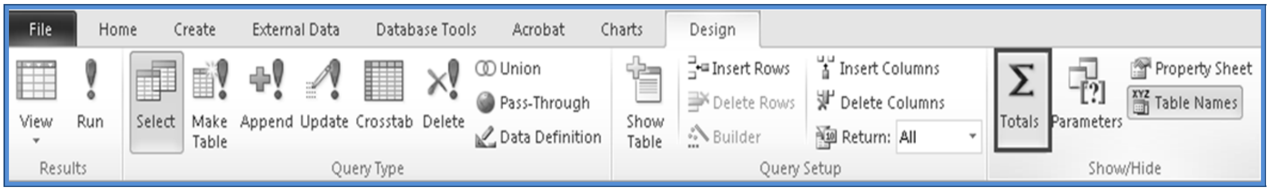


Fig. 1.2.26: Totals option in Design tab

The **Totals** option adds a new row in the *Query Designer* window which enables the application of summary functions to columns as indicated in Fig. 1.2.27.

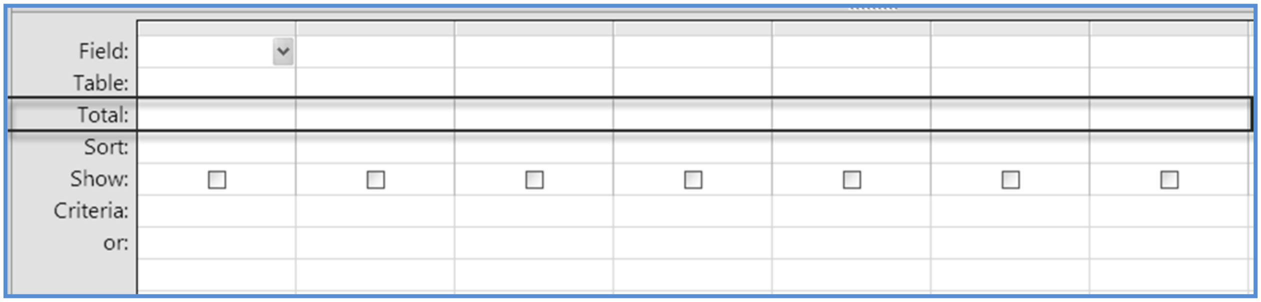


Fig. 1.2.27: Total Row added

A list of *Aggregate Functions* that can be used with *Totals* Query is given in Table 1.2.1.

FUNCTION NAME	EXPLANATION
Sum()	Returns the sum of numeric data for a column or set of values in a column
Count()	Counts the set of values that satisfy the given criteria
Avg()	Returns average of numeric data for a column or set of values in a column
Max()	Returns a maximum value from a set of values
Min()	Returns a minimum value from a set of values
Var()	Returns the variance of values in a column or set of values
Stdev()	Returns a standard deviation for a set of data values
First()	Returns the first value from a list of text values
Last()	Returns the last value from a list of text values

Table 1.2.1: Aggregate Functions

1.2.3.2 Problem Scenario

The Country Head in Apex Ltd. wishes to compute the sales volume of its products. He also would like to see the number of orders placed for each product, so that they can focus on products with greater sales volume.

Solution:

Create a Total query that calculates the count of orders placed for each product and arrange them in descending order of the counts so as to find the products which are sold more.

Steps for creating a total query

1. Open the *Query Design* window by selecting **Query Design** from **Create** -> **Queries**, as indicated in Fig. 1.2.28.

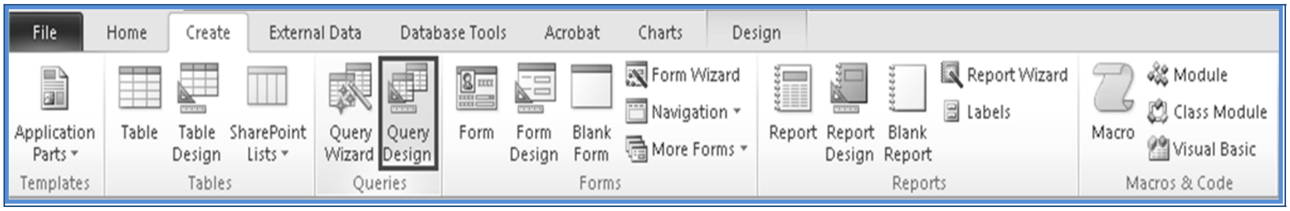
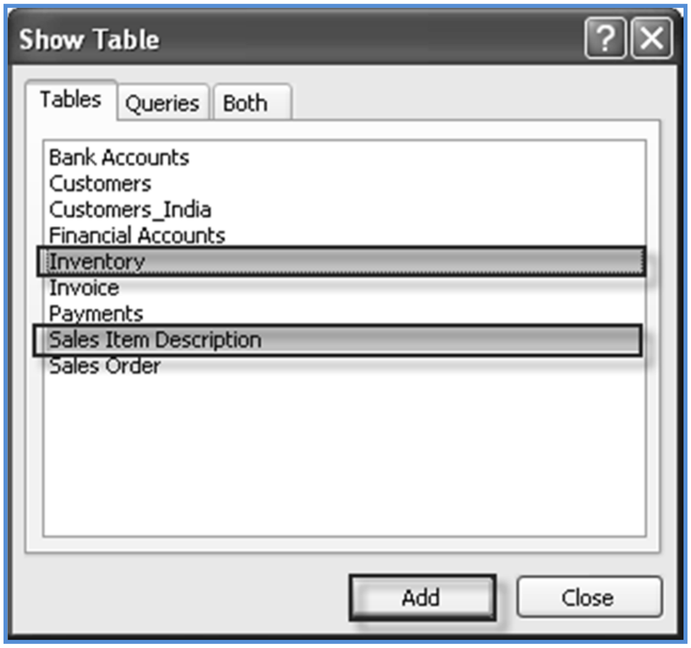


Fig. 1.2.28: Opening the Query Design window

2. Select the *Inventory* and *Sales Item Description* tables from **Show Table** dialog box and click **Add**, as shown in Fig. 1.2.29.

NOTE: Multiple tables can be selected in the **Show Table** window by pressing *CTRL* key and then selecting the tables.


Show Table

Tables | Queries | Both

Bank Accounts
Customers
Customers_India
Financial Accounts
Inventory
Invoice
Payments
Sales Item Description
Sales Order

Add | Close

Fig. 1.2.29: Show Table window

3. Click **Close**. We can see the two tables with a 1:∞ relationship. This relationship has been created while creating the database.

4. Click the **Totals** button in the **Show/Hide** group on the **Design** tab, as shown as Fig. 1.2.30.

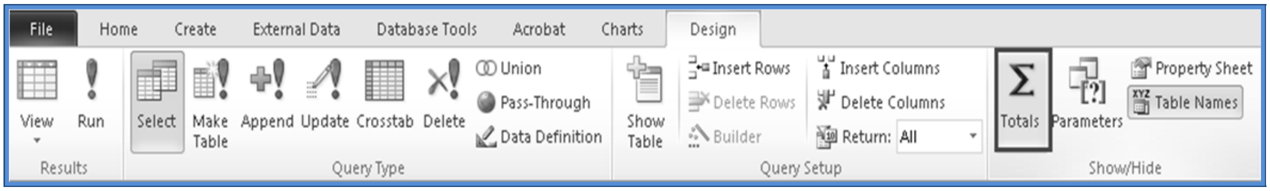
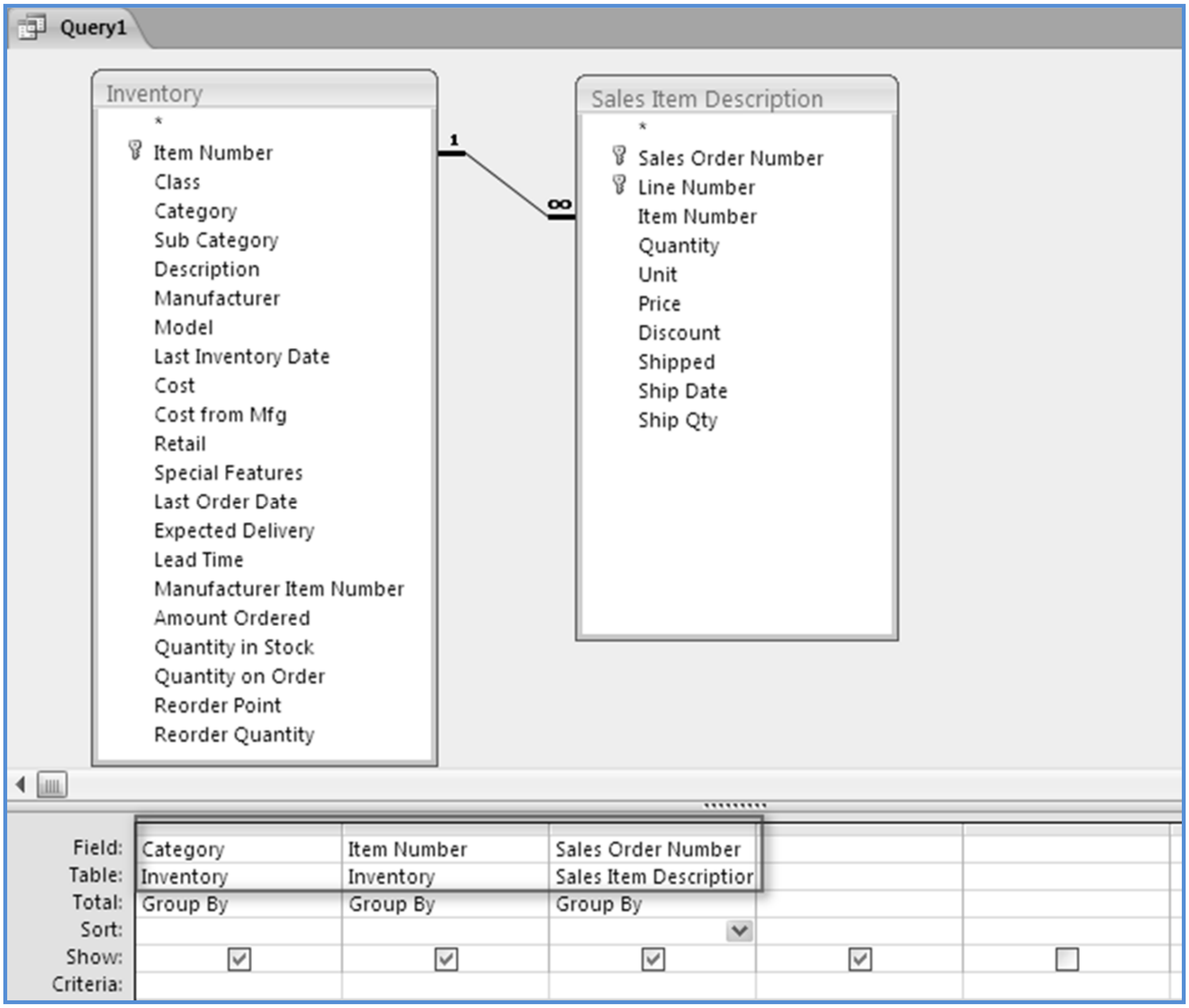


Fig. 1.2.30: Click on Totals button

5. Select the fields *Category* and *Item Number* from *Inventory* table and drag them to the Columns tab. Similarly, select fields *Sales Order Number* from *Sales Item Description* table and drag it to Columns tab. The *Query Window* appears as shown in Fig. 1.2.31.

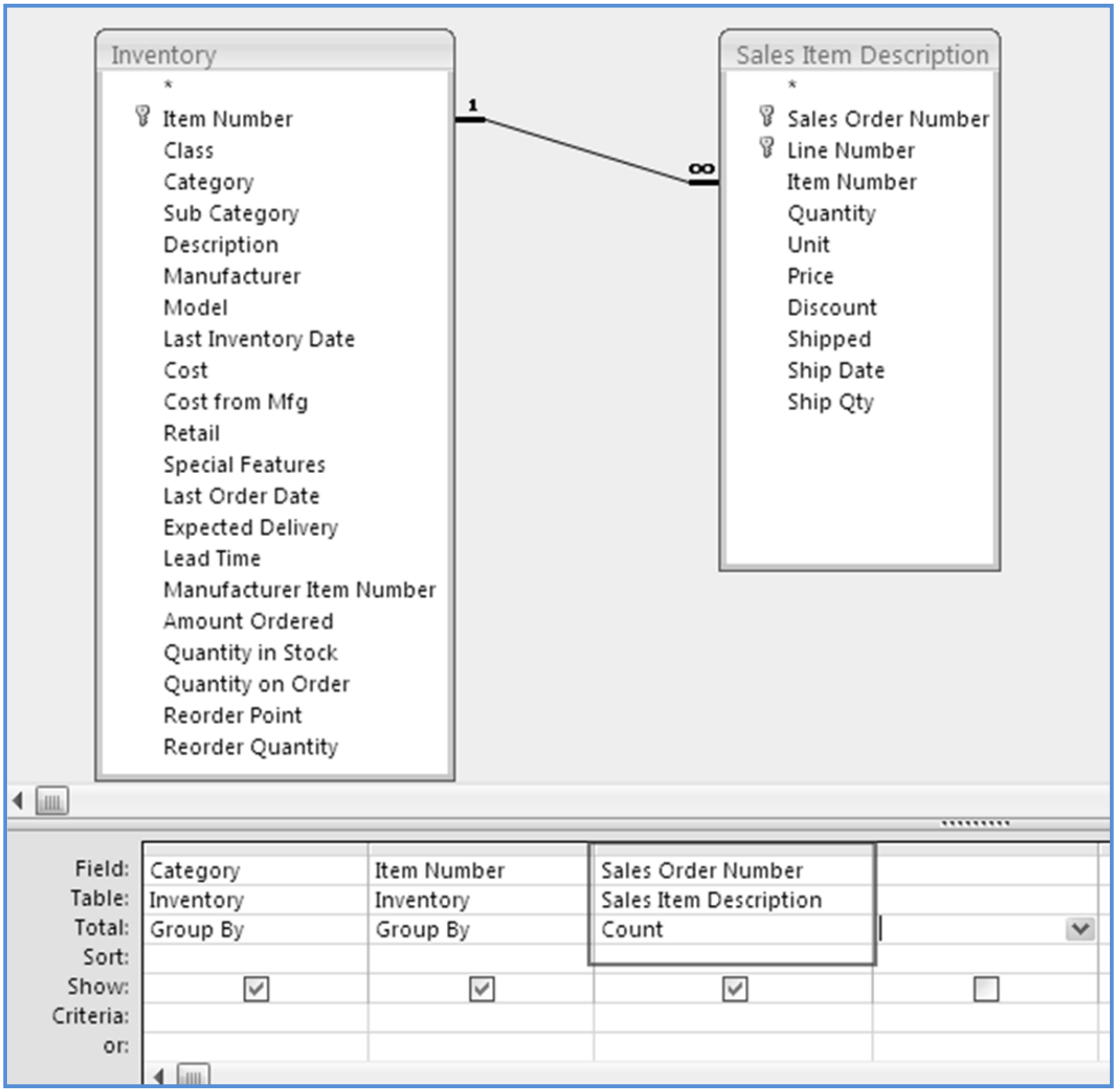


Field:	Category	Item Number	Sales Order Number		
Table:	Inventory	Inventory	Sales Item Descriptior		
Total:	Group By	Group By	Group By		
Sort:					
Show:	☑	☑	☑	☑	□
Criteria:					

Fig. 1.2.31: Select the required columns

6. In the column *Sales Order Number*, change the **Group By** function to **Count**, as shown in Fig. 1.2.32.

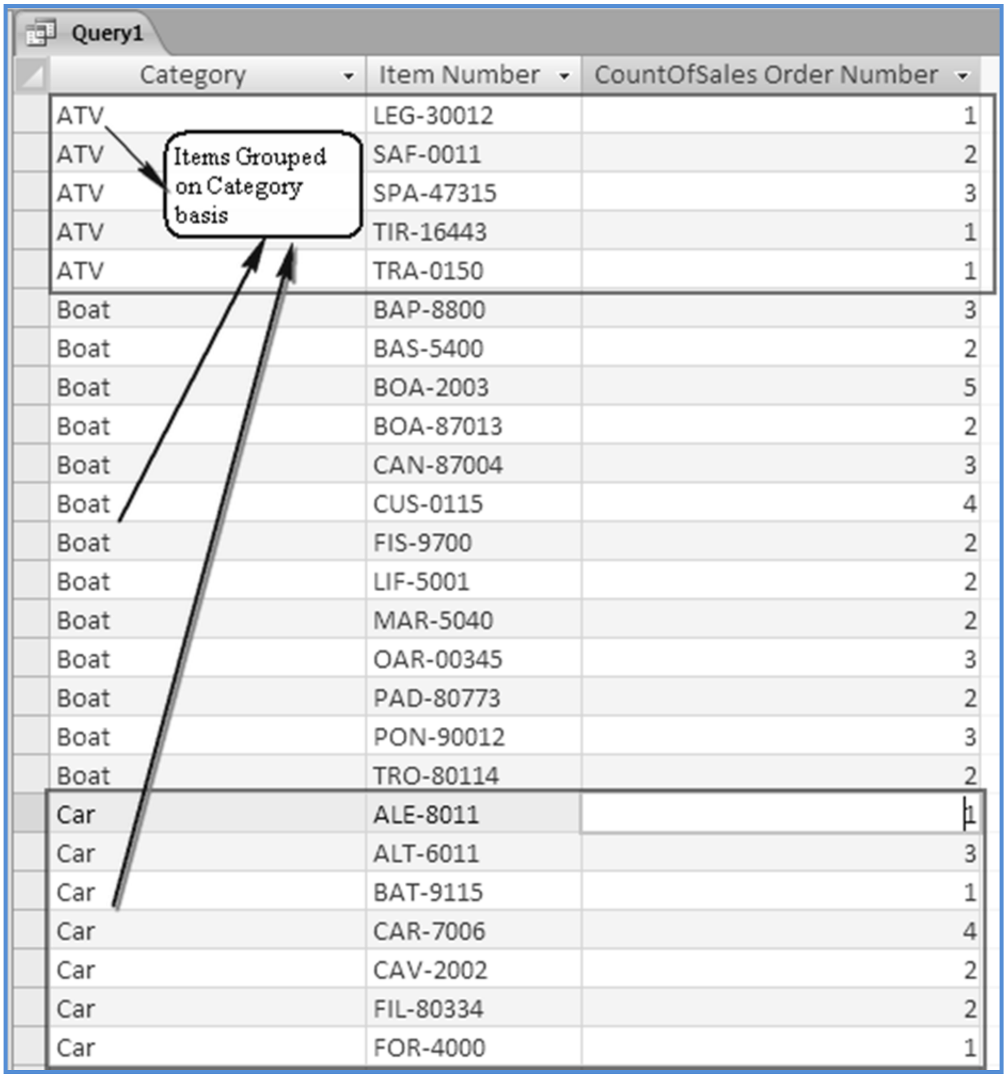
Notice that we have set **Group By** on columns *Category* and *Item Number*, so that the query first groups all the items according to their category and then all the items in same category on the basis of *Item Number*. The **Count** function with *Sales Order Number* indicates that we wish to compute the count of total Sales Orders for a group created.



Field:	Category	Item Number	Sales Order Number	
Table:	Inventory	Inventory	Sales Item Description	
Total:	Group By	Group By	Count	
Sort:				
Show:	☑	☑	☑	□
Criteria:				
or:				

Fig. 1.2.32: Set the Totals Function

7. Click **Run** in **Design** -> **Results** to view the results. The result of the query appears, as shown in Fig. 1.2.33.



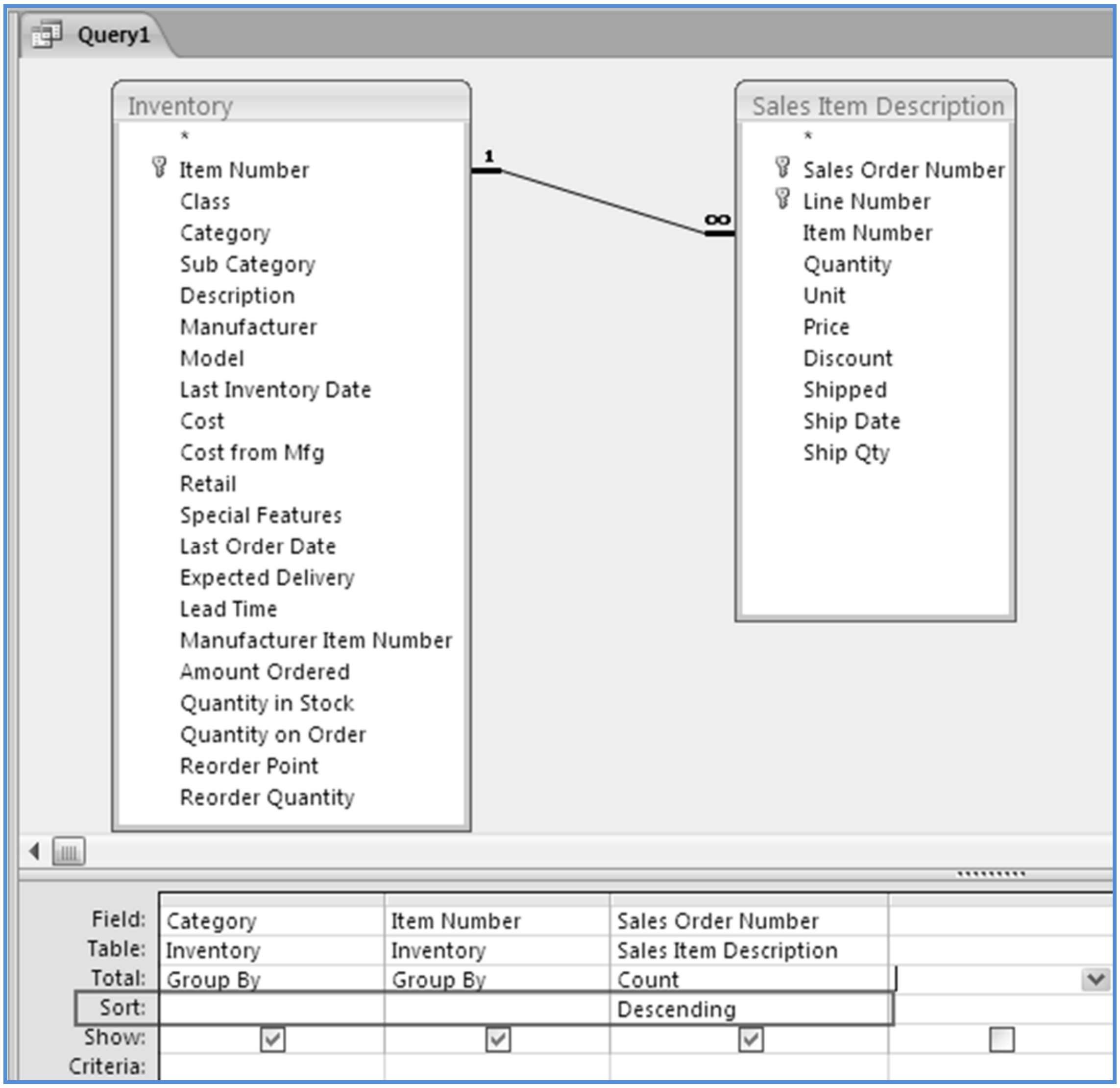
Category	Item Number	CountOfSales Order Number
ATV	LEG-30012	1
ATV	SAF-0011	2
ATV	SPA-47315	3
ATV	TIR-16443	1
ATV	TRA-0150	1
Boat	BAP-8800	3
Boat	BAS-5400	2
Boat	BOA-2003	5
Boat	BOA-87013	2
Boat	CAN-87004	3
Boat	CUS-0115	4
Boat	FIS-9700	2
Boat	LIF-5001	2
Boat	MAR-5040	2
Boat	OAR-00345	3
Boat	PAD-80773	2
Boat	PON-90012	3
Boat	TRO-80114	2
Car	ALE-8011	1
Car	ALT-6011	3
Car	BAT-9115	1
Car	CAR-7006	4
Car	CAV-2002	2
Car	FIL-80334	2
Car	FOR-4000	1

Fig. 1.2.33: Query Result

NOTE: The Query result just shows all the products arranged in group of Category and Item Number and their volume sold. To get the products which are sold more, the column *CountOfSales Order Number* has to be arranged in order that it appears from most sold to least sold.

8. Reopen the query in *Design View* by clicking **Design View** *from* **Home** -> **Views**.

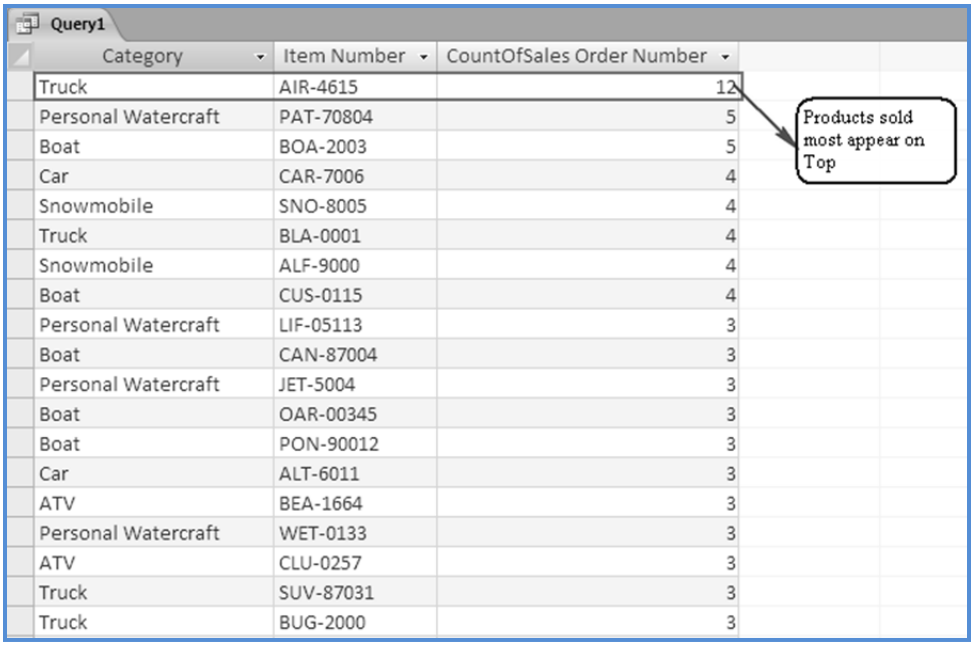
9. In the column *Sales Order Number*, set the sort order as descending, as shown in Fig. 1.2.34.



Field:	Category	Item Number	Sales Order Number	
Table:	Inventory	Inventory	Sales Item Description	
Total:	Group By	Group By	Count	
Sort:			Descending	
Show:	☑	☑	☑	□
Criteria:				

Fig. 1.2.34: Set the sort order

10. Click **Run** in **Design -> Results** to view the result. The result appears as shown in Fig. 1.2.35.



Category	Item Number	CountOfSales Order Number
Truck	AIR-4615	12
Personal Watercraft	PAT-70804	5
Boat	BOA-2003	5
Car	CAR-7006	4
Snowmobile	SNO-8005	4
Truck	BLA-0001	4
Snowmobile	ALF-9000	4
Boat	CUS-0115	4
Personal Watercraft	LIF-05113	3
Boat	CAN-87004	3
Personal Watercraft	JET-5004	3
Boat	OAR-00345	3
Boat	PON-90012	3
Car	ALT-6011	3
ATV	BEA-1664	3
Personal Watercraft	WET-0133	3
ATV	CLU-0257	3
Truck	SUV-87031	3
Truck	BUG-2000	3

Products sold most appear on Top

Fig. 1.2.35: Query Results sorted according to sales of Product

11. Click the **Save** button on *Quick Access Toolbar* to save the query. Type the name of query as *Product Sales Volume.*

NOTE: Queries are required to be saved for future use if the same result is desired again; it saves the effort and time of recreating the query.

1.2.3.3 Applying Criteria to Records

A criterion limits the records that are displayed as result, on the basis of values in a field. A criterion in Total queries can be created against *Group By*, *Aggregate Total*, *Non-Aggregate Total* fields. For *Group By* and *Aggregate Total,* criteria can be mentioned using **Criteria** tab of Query Design window. For the *Non-Aggregate Total* field, the criteria can only be specified using the *Where* clause instead of *Group By* or *Aggregate* function in *Total* tab of Query Design window.

Using any one, any two, or all three of these criteria, the scope of Total query can be limited to finite criteria.

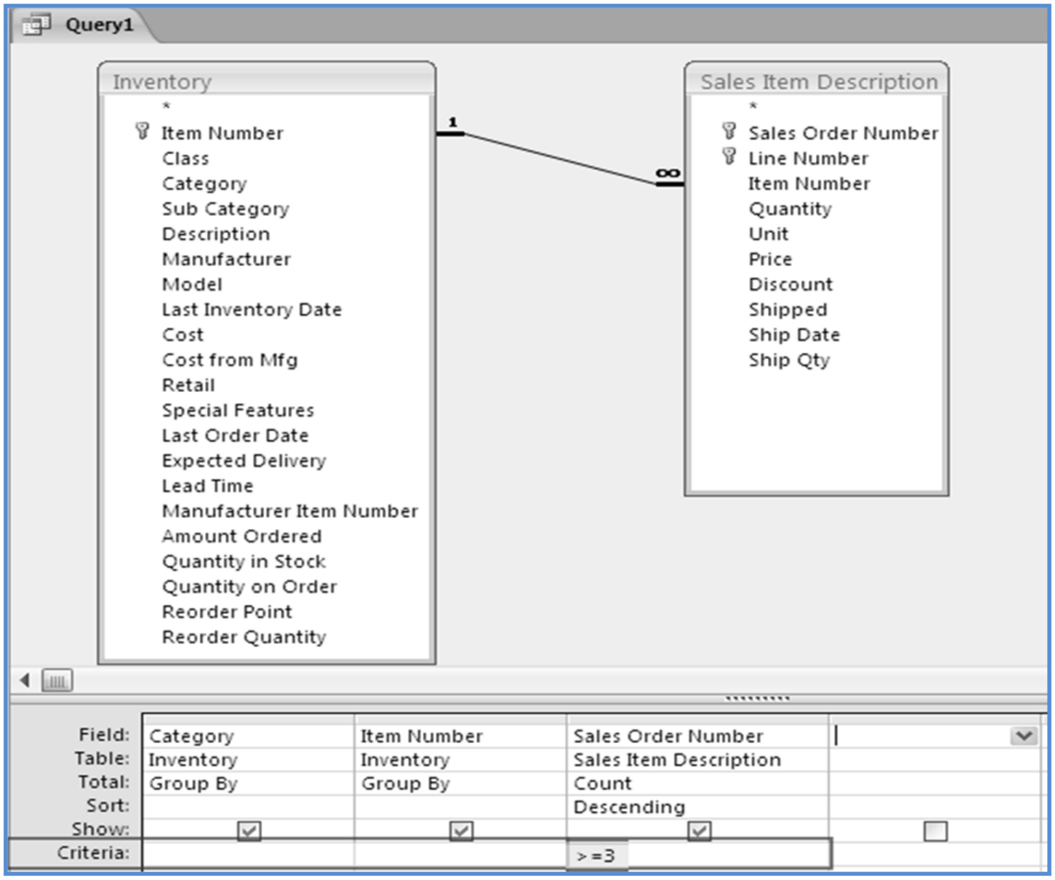
1.2.3.4 Problem Scenario

Considering the problem of retrieving Sales volume of Products as discussed in Section 2.2.3.1, Country Head wants to have a look at only the products with at least an order count of three. The management authority also thought of maintaining the products which are cheaper and have more sales volume.

Solution

Update the Total query *Product Sales Volume* created above and apply criteria “>=3” on the *Sales Order Number* count to ensure that only the products with at least an order count of three are retrieved as a query result. Also, add criteria “<50” to the *Cost* field of the products so that management can have a view on the cheaper products.

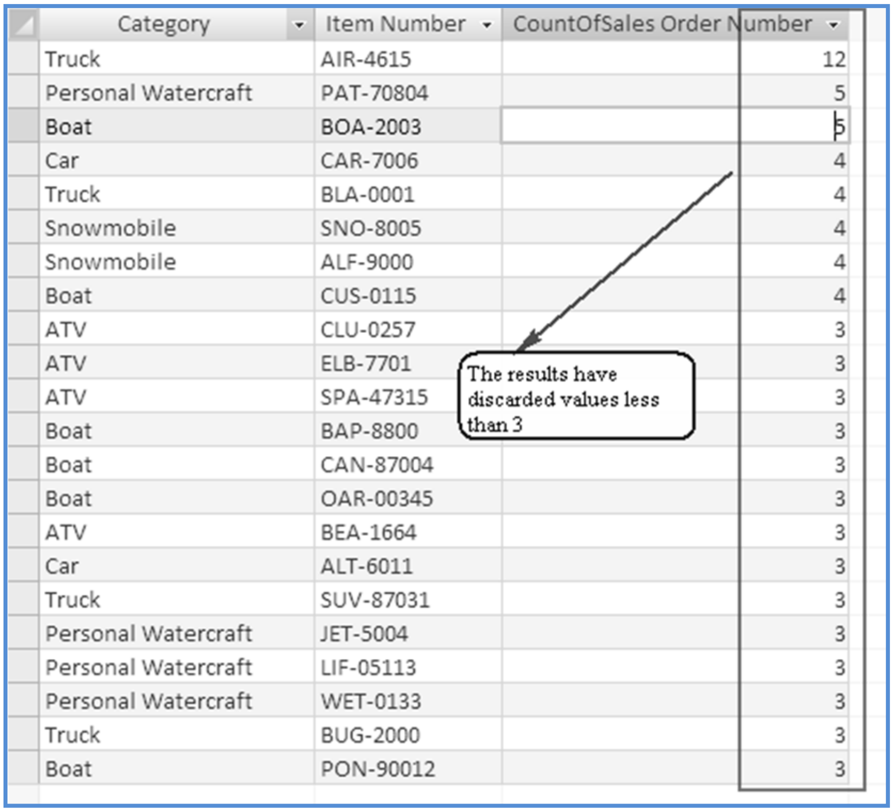
1. Open the query *Product Sales Volume* in Design view. To do this, right-click the query and select *Design View*.
2. In the column *Sales Order Number*, type “>=3” in the Criteria tab as shown in Fig. 1.2.36.



Field:	Category	Item Number	Sales Order Number	
Table:	Inventory	Inventory	Sales Item Description	
Total:	Group By	Group By	Count	
Sort:			Descending	
Show:	☑	☑	☑	□
Criteria:			>=3	

Fig. 1.2.36: Insert Criteria for Sales Order Number

3. Click **Run** in **Design** -> **Results** to view the result. The result of the query appears as shown in Fig. 1.2.37.



Category	Item Number	CountOfSales Order Number
Truck	AIR-4615	12
Personal Watercraft	PAT-70804	5
Boat	BOA-2003	5
Car	CAR-7006	4
Truck	BLA-0001	4
Snowmobile	SNO-8005	4
Snowmobile	ALF-9000	4
Boat	CUS-0115	4
ATV	CLU-0257	3
ATV	ELB-7701	3
ATV	SPA-47315	3
Boat	BAP-8800	3
Boat	CAN-87004	3
Boat	OAR-00345	3
ATV	BEA-1664	3
Car	ALT-6011	3
Truck	SUV-87031	3
Personal Watercraft	JET-5004	3
Personal Watercraft	LIF-05113	3
Personal Watercraft	WET-0133	3
Truck	BUG-2000	3
Boat	PON-90012	3

The results have discarded values less than 3

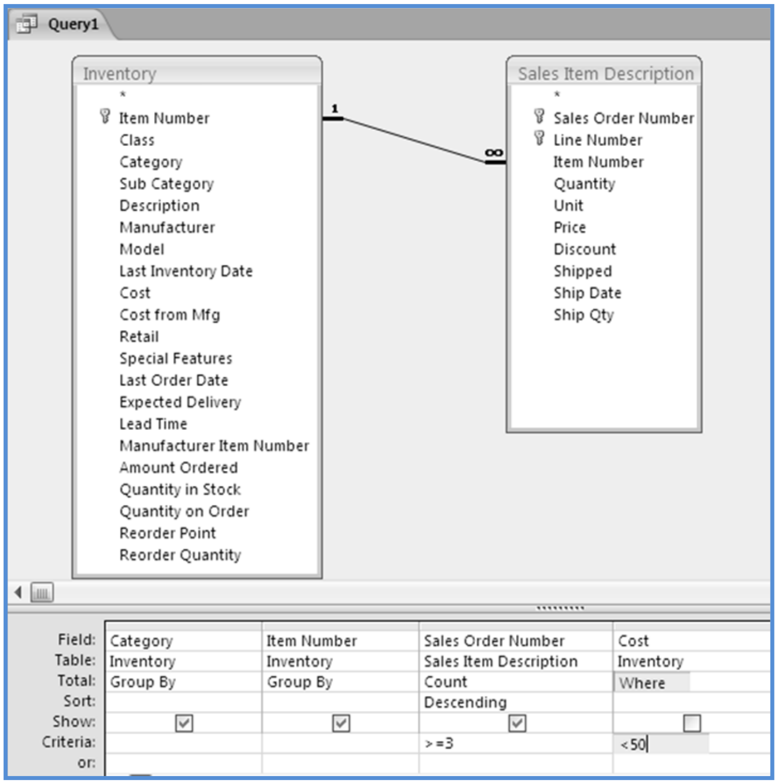
Fig. 1.2.37: Query Result for Products having at least 3 Sales Orders

Now, include only the products which cost less than 50.00 in the query result. Since *Cost* is a non-aggregate field, the criteria for the *Cost* is included in the *Where* clause of **Total** tab.

4. Double-click the field *Cost* in the *Inventory* table so as to include it into the **Field** tab of *Query Design* window.

5. Change the *Group By* function of column *Cost* to *Where* clause in **Total** tab and write "*<50"* in the **Criteria** tab, as shown in Fig. 1.2.38.

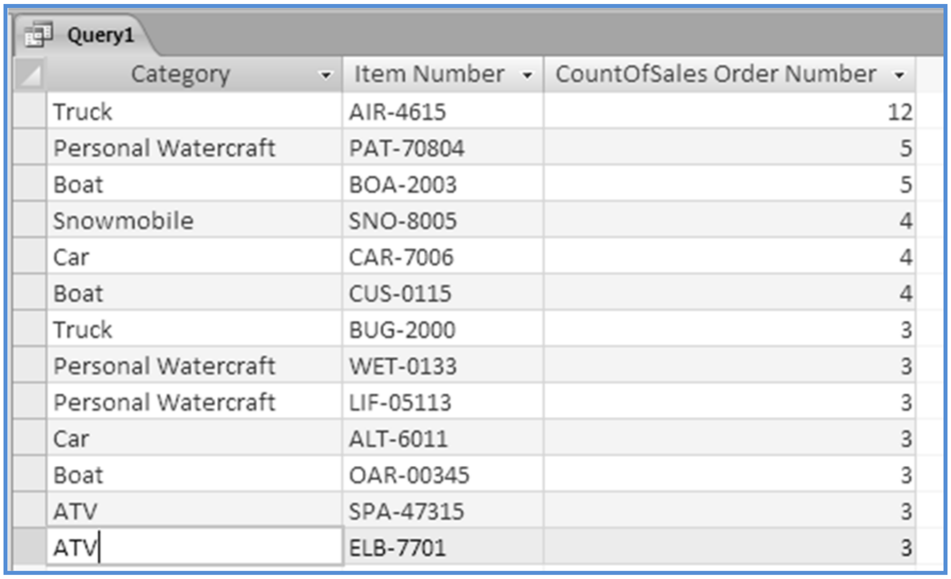
   Note that the **Show** checkbox is cleared as *Cost* is a non-aggregate field and not a part of the query result.



Field:	Category	Item Number	Sales Order Number	Cost
Table:	Inventory	Inventory	Sales Item Description	Inventory
Total:	Group By	Group By	Count	Where
Sort:			Descending	
Show:	☑	☑	☑	□
Criteria:			>=3	<50
or:				

Fig. 1.2.38: Products with cost <50

6. Click **Run** in **Design** -> **Results** to view the result of the query, as shown in Fig. 1.2.39.



Category	Item Number	CountOfSales Order Number
Truck	AIR-4615	12
Personal Watercraft	PAT-70804	5
Boat	BOA-2003	5
Snowmobile	SNO-8005	4
Car	CAR-7006	4
Boat	CUS-0115	4
Truck	BUG-2000	3
Personal Watercraft	WET-0133	3
Personal Watercraft	LIF-05113	3
Car	ALT-6011	3
Boat	OAR-00345	3
ATV	SPA-47315	3
ATV	ELB-7701	3

Fig. 1.2.39: Query Result

7. Click **Save** at *Quick Access Toolbar* to save the query.

1.2.4 Grouping and summarizing Records using a Crosstab Query

Crosstab query is an excellent analytical tool. It is a special type of query that can be created to describe one numerical quantity in terms of two other fields. Crosstab queries are useful for summarizing information, and are somewhat similar to pivot tables in MS-Excel.

For example, we might want to have a table that contains the sales figures of entire inventory for the whole year, and the amount of money that is made per product during each month of the year. In this case, a crosstab query would be the right way to display the information.

Access 2010 provides an option to create *Crosstab query* through *Query Wizard*. Each crosstab query will include one or more Row Heading, a single Column Heading and a Value to be displayed at the intersection of row and column.

1.2.4.1 Problem Scenario

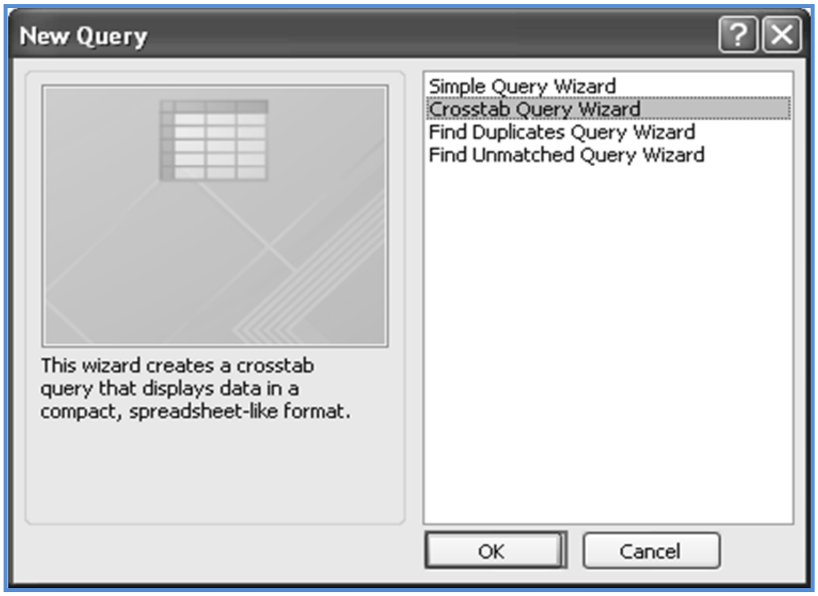
The company Apex Ltd. wishes to revise the credit limit of its customers. For this purpose, Ashish, the PRO needs a summary report that indicates the count of orders placed by each customer in every month of last financial year i.e. 2011-2012.

Solution

To obtain the desired summary report, create a *Crosstab query* on table *Sales Order* with month of *Sales Date* as row heading, *Customer Number* as column heading and count of *Sales Order Number* as values. We create this query using the *Query Wizard*.

Steps for creating Crosstab queries

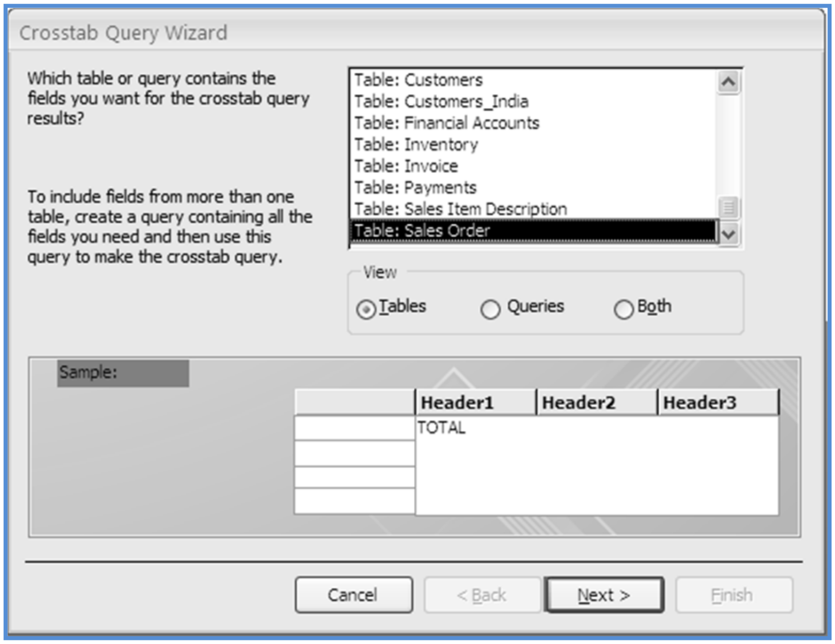
1. Open the *Query Wizard* by selecting **Query Wizard** from **Create** -> **Queries**
2. Select **Crosstab Query Wizard** from the **New Query** dialog box and click **OK** as in Fig. 1.2.40.



This wizard creates a crosstab query that displays data in a compact, spreadsheet-like format.

Fig. 1.2.40: Select Crosstab Query Wizard

3. The **Crosstab Query Wizard** appears. Select the table *Sales Order* and click **Next** as shown in Fig. 1.2.41.



Which table or query contains the fields you want for the crosstab query results?

To include fields from more than one table, create a query containing all the fields you need and then use this query to make the crosstab query.

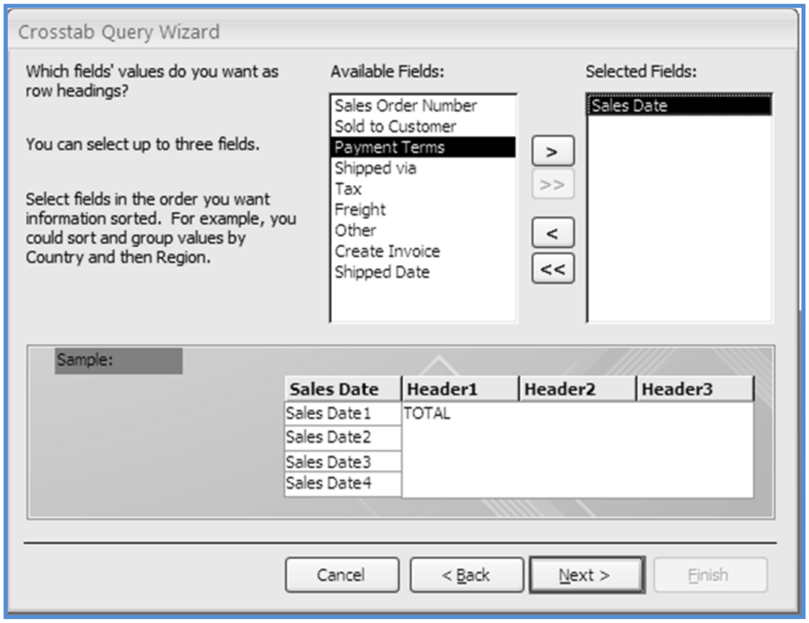
Table: Customers
Table: Customers_India
Table: Financial Accounts
Table: Inventory
Table: Invoice
Table: Payments
Table: Sales Item Description
Table: Sales Order

● Tables ○ Queries ○ Both

	Header1	Header2	Header3
	TOTAL		

Fig. 1.2.41: Select Table Sales Order

4. The next page in *Query Wizard* confirms the column to be taken as Row Heading. In this window, select the column *Sales Date* as shown in Fig. 1.2.42 and click **Next**.



Which fields' values do you want as row headings?

You can select up to three fields.

Select fields in the order you want information sorted. For example, you could sort and group values by Country and then Region.

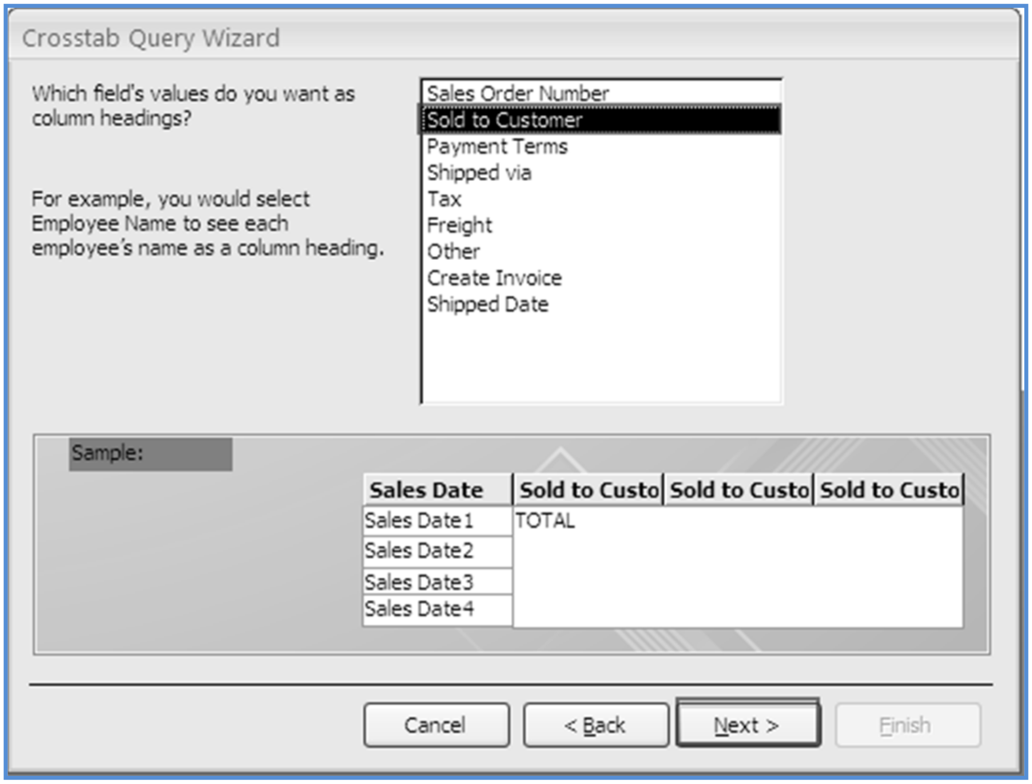
Available Fields:
Sales Order Number
Sold to Customer
Payment Terms
Shipped via
Tax
Freight
Other
Create Invoice
Shipped Date

Selected Fields:
Sales Date

Sales Date	Header1	Header2	Header3
Sales Date1	TOTAL		
Sales Date2			
Sales Date3			
Sales Date4			

Fig. 1.2.42: Select column Sales Date for Row Heading

5. Select the column to be taken as Column Heading in the next window of the *Query Wizard* that appears. In this window, select the column *Sold to Customer* as shown in Fig. 1.2.43 and click **Next**.



Which field's values do you want as column headings?

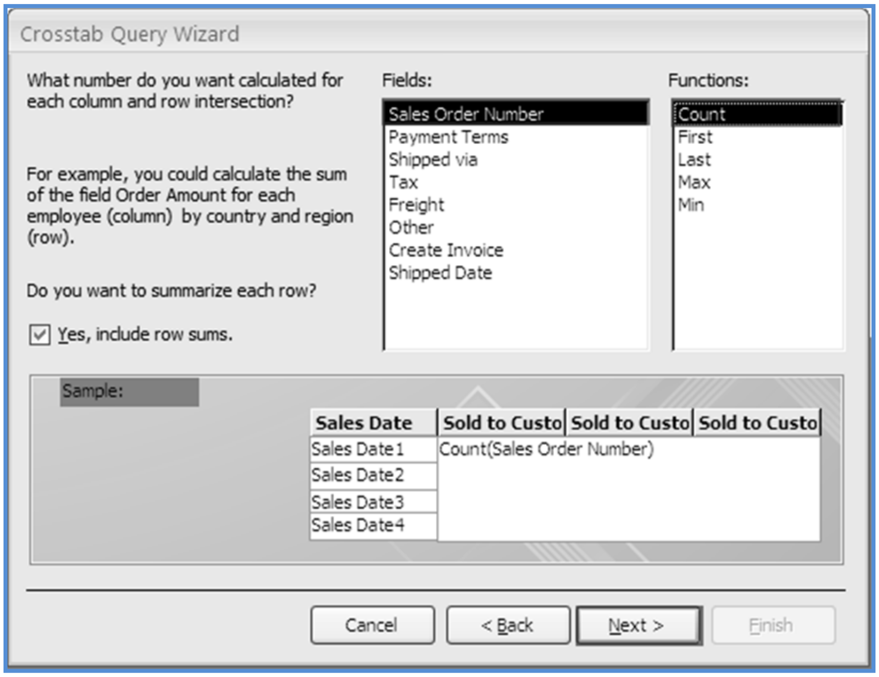
For example, you would select Employee Name to see each employee's name as a column heading.

Sales Date	Sold to Custo	Sold to Custo	Sold to Custo
Sales Date1	TOTAL		
Sales Date2			
Sales Date3			
Sales Date4			

Fig. 1.2.43: Select column Sold to Customer for Column Heading

6. Next page in the wizard prompts to specify the values that should appear on the intersection of rows and columns. Select the column *Sales Order Number* from the **Fields** tab and **Count** from the **Functions** tab in this page, as shown in Fig. 1.2.44. Click **Next**.

NOTE: This page has a checkbox **Yes, include row sums;** which if checked, allows the inclusion of the grand total of values in the row as a column in the query result.



Crosstab Query Wizard

What number do you want calculated for each column and row intersection?

For example, you could calculate the sum of the field Order Amount for each employee (column) by country and region (row).

Do you want to summarize each row?

☑ Yes, include row sums.

Fields:

- Sales Order Number
- Payment Terms
- Shipped via
- Tax
- Freight
- Other
- Create Invoice
- Shipped Date

Functions:

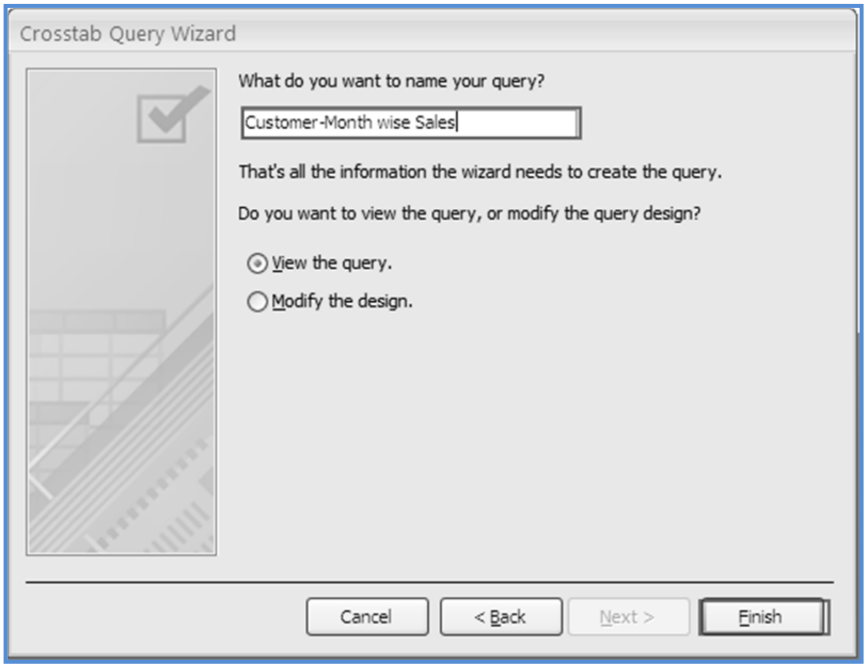
- Count
- First
- Last
- Max
- Min

Sample:

Sales Date	Sold to Custo	Sold to Custo	Sold to Custo
Sales Date1	Count(Sales Order Number)		
Sales Date2			
Sales Date3			
Sales Date4			

Fig. 1.2.44: Select Sales Order Number Count as values

7. Write the name for the query as “Customer-Month wise Sales” in the final window that appears and click **Finish**, as shown in Fig. 1.2.45. We can also choose the default query name that the wizard displays.



Crosstab Query Wizard

What do you want to name your query?

Customer-Month wise Sales

That's all the information the wizard needs to create the query.

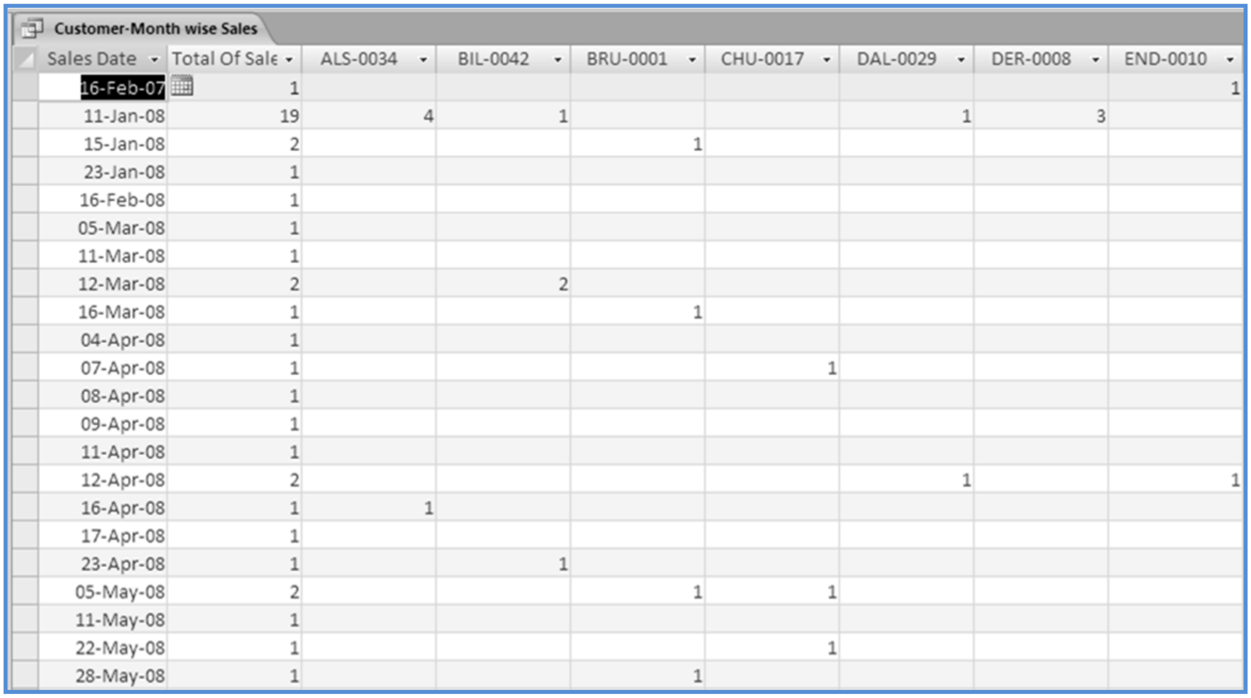
Do you want to view the query, or modify the query design?

● View the query.

○ Modify the design.

Fig. 1.2.45: Name the query

8. The result of the query appears in Datasheet form, as shown in Fig. 1.2.46.



Sales Date	Total Of Sale	ALS-0034	BIL-0042	BRU-0001	CHU-0017	DAL-0029	DER-0008	END-0010
16-Feb-07	1							1
11-Jan-08	19	4	1			1	3	
15-Jan-08	2			1				
23-Jan-08	1							
16-Feb-08	1							
05-Mar-08	1							
11-Mar-08	1							
12-Mar-08	2		2					
16-Mar-08	1			1				
04-Apr-08	1							
07-Apr-08	1				1			
08-Apr-08	1							
09-Apr-08	1							
11-Apr-08	1							
12-Apr-08	2					1		1
16-Apr-08	1	1						
17-Apr-08	1							
23-Apr-08	1		1					
05-May-08	2			1	1			
11-May-08	1							
22-May-08	1				1			
28-May-08	1			1				

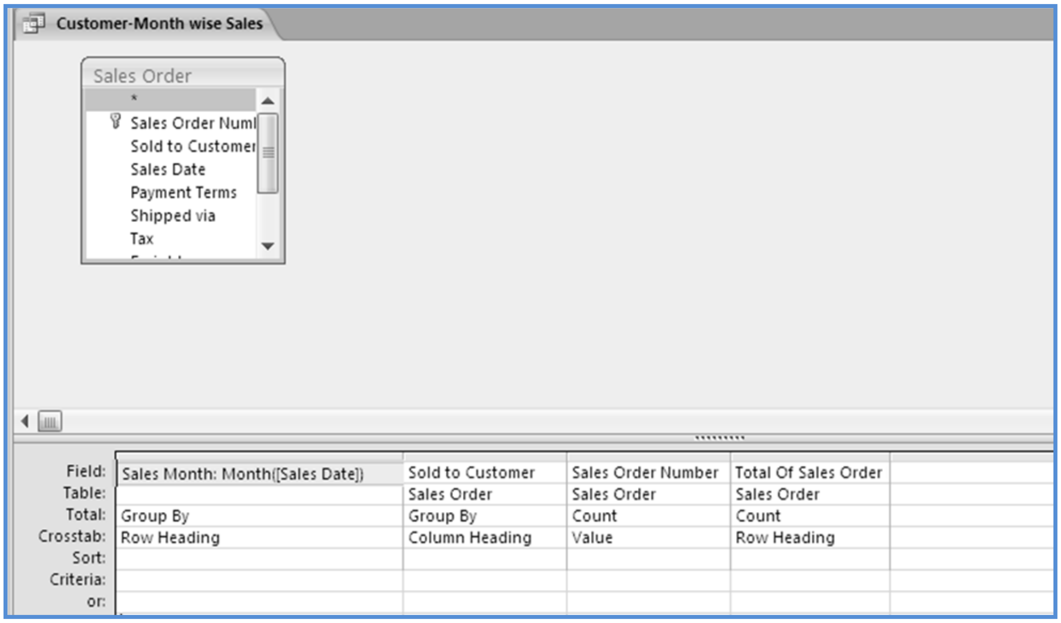
Fig. 1.2.46: Query Result

NOTE: The query result is not as desired as it displays the values on the basis of the Sales Date and not on Sales Month. To view these results on the basis of Sales Month, we are required to make changes in the query design and change the column *Sales Date* to the *Month (Sales Date)* and also provide a criterion that the months must be in last financial year.

Steps to group results on Sales Order Month

1. Reopen the query in design view by clicking the **Design View** from **Home** -> **Views**.
2. Click on field *Sales Date* and write the expression *Sales Month: Month ([Sales Date])* in **Field** tab as shown in Fig. 1.2.47.

NOTE: We can use the function **Month Name()** to get names of month instead of numbers in query results.

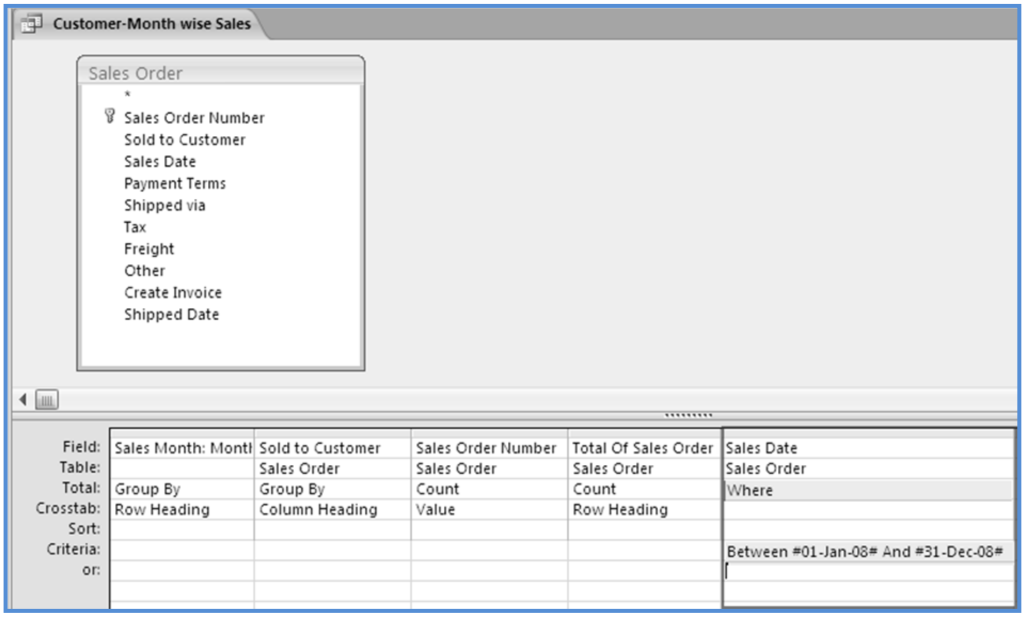


Field:	Sales Month: Month([Sales Date])	Sold to Customer	Sales Order Number	Total Of Sales Order
Table:		Sales Order	Sales Order	Sales Order
Total:	Group By	Group By	Count	Count
Crosstab:	Row Heading	Column Heading	Value	Row Heading
Sort:				
Criteria:				
or:				

Fig. 1.2.47: Change the expression of field Sales Date

Steps to provide the criteria for last financial year

1. Double-click the column *Sales Date* from table *Sales Order* such that it appears in **Field** tab. Change the value of **Total** tab to *Where* and add the expression **Between 4/1/2011 AND 3/31/2012** to the **Criteria** tab as shown in Fig. 1.2.48.

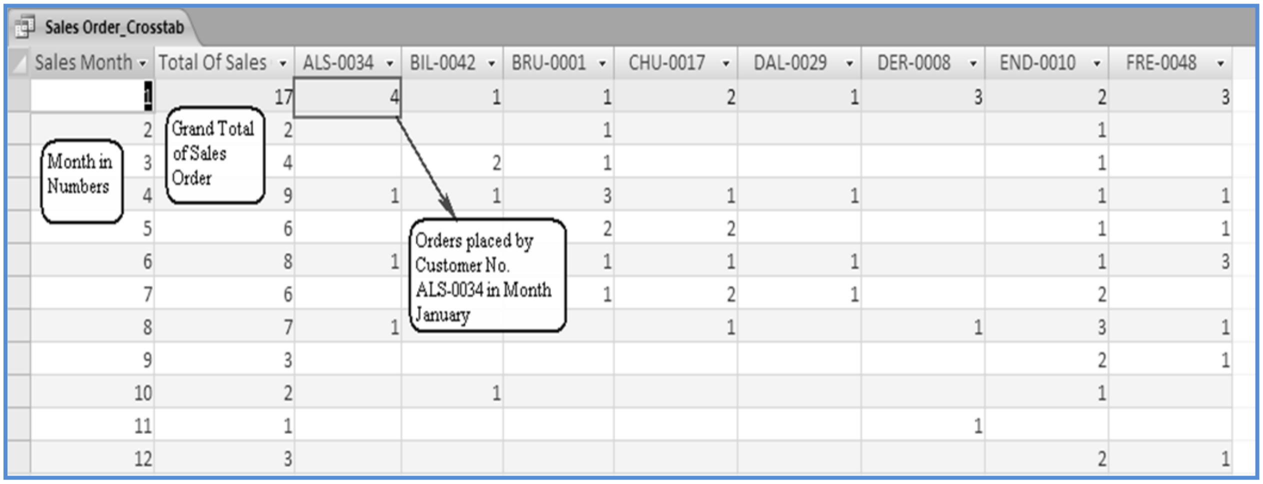


Field:	Sales Month: Montl	Sold to Customer	Sales Order Number	Total Of Sales Order	Sales Date
Table:		Sales Order	Sales Order	Sales Order	Sales Order
Total:	Group By	Group By	Count	Count	Where
Crosstab:	Row Heading	Column Heading	Value	Row Heading	
Sort:					
Criteria:					Between #01-Jan-08# And #31-Dec-08#
or:					

Fig. 1.2.48: Provide a criteria to Sales Date column

NOTE: The Access 2010 query window has changed the criteria expression suitable to match column values.

2. Click **Run** in **Design** -> **Results** to view the modified result as shown in Fig.1.2.49.



Sales Order_Crosstab

Sales Month	Total Of Sales	ALS-0034	BIL-0042	BRU-0001	CHU-0017	DAL-0029	DER-0008	END-0010	FRE-0048
1	17	4	1	1	2	1	3	2	3
2	2			1				1	
3	4		2	1				1	
4	9	1	1	3	1	1		1	1
5	6			2	2			1	1
6	8	1		1	1	1		1	3
7	6			1	2	1		2	
8	7	1			1		1	3	1
9	3							2	1
10	2		1					1	
11	1						1		
12	3							2	1

Month in Numbers
Grand Total of Sales Order
Orders placed by Customer No. ALS-0034 in Month January

Fig. 1.2.49: Query Result

NOTE: The result of the query appears as a 3-dimensional table with *Months* on rows, *Customer No.* on columns and the *Count of orders* as values.

Crosstab Queries are capable of handling much more complex calculations. These queries can be based on another query using multiple tables or may use different level of grouping by having more than one row heading. Also, Crosstab queries can be created by Query Design by changing the query type to **Crosstab**.

1.2.5 Creating a PivotTable and PivotChart

The powerful tool of MS-Excel *PivotTable* and *PivotChart* is also available in Access 2010 to summarize data. Interactive *Pivot Tables* and *Pivot Charts* enable the manipulation of summary data, and therefore can save ample time to create multiple queries and reports to achieve the same results.

1.2.5.1 PivotTable

A *PivotTable* is a view in Access Queries that allows summarising and examining data in a datasheet form. It is used to group values as rows and columns with a calculated value at the intersection of each row and column. A *PivotTable* can be considered as a modified form of Crosstab queries discussed in the above section. A *PivotTable* is created by dragging fields to the appropriate area on the design screen. Data can also be broken down to different levels of detail, such as showing earnings by year, quarter, or month.

The *PivotTable* view in a query can be obtained by selecting the **PivotTable View** from the **Views** drop-down in **Home** tab as indicated in Fig. 1.2.50.

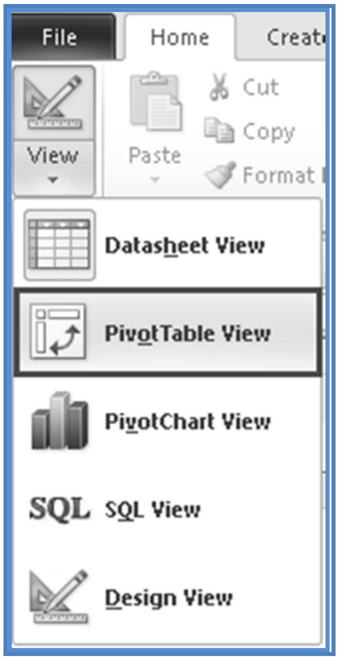


Fig. 1.2.50: PivotTable View

The *PivotTable* is shown in Fig. 1.2.51.

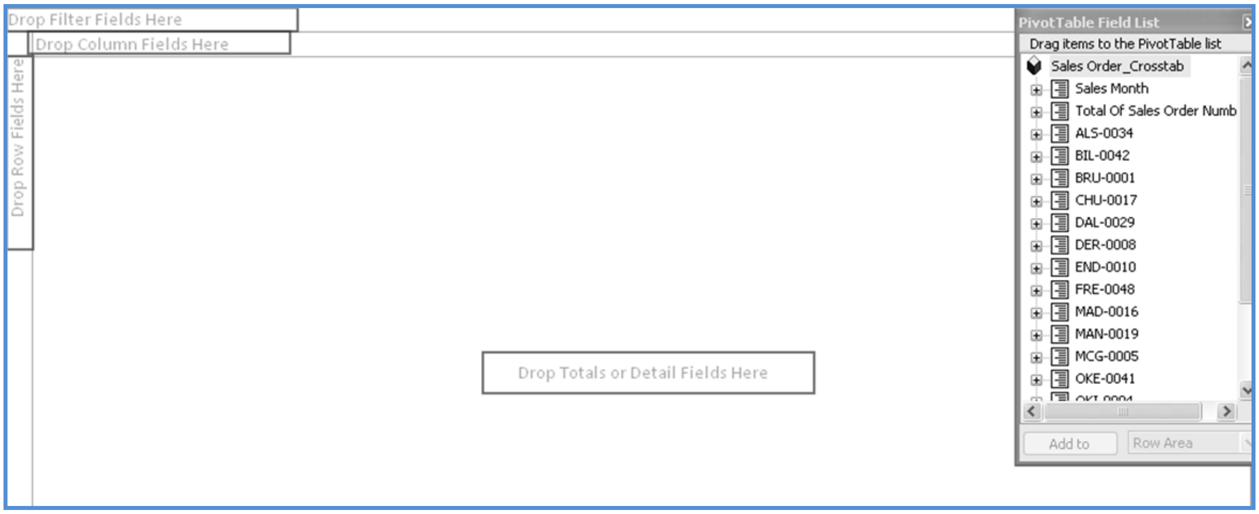


Fig. 1.2.51: PivotTable

A *PivotTable* has four areas for dropping fields whose values are summarized. The description of these areas is shown in Table 2.2.2.

Drop Area	Description
Filter Area	Is used for the fields that are used as a filter for PivotTable
Column Area	The fields selected to be displayed as column headings are included in this area
Row Area	The fields selected to be displayed as row headings are dropped in this area
Totals Or Detail Area	The field values to be used for calculations or summarization are dragged and dropped in this area, and the value is displayed at the intersection of a row and column

Table 1.2.2: Drop Area in Pivot Table

1.2.5.2 Problem Scenario

The Head of Sales Department in Apex Ltd. has demanded a Sales Summary report for review. The following are the desired requirements: the sales of the products can be viewed in terms of Year, Quarters and Months; the report should enable the user to view the products filtered by Category.

Solution

To create this Summary report, first a query is created that displays the Products and their Category, Sales amount of each product, and also the date on which they were sold. Then to create summary sheet, we create a *PivotTable* view of the query having:

- Category field as a filter
- Product field on row
- Year, Quarter and Month on columns
- Total Sales (which is computed as Quantity sold into Product cost) displayed as values

Steps for creating the query to obtain Sales Data

1. Open the *Query Design* window, by clicking **Create** -> **Queries** -> **Query Design**.
2. Select the *Inventory*, *Sales Item Description* and *Sales Order* tables from **Show Table** window and click **Add***,* as shown in Fig. 1.2.52.

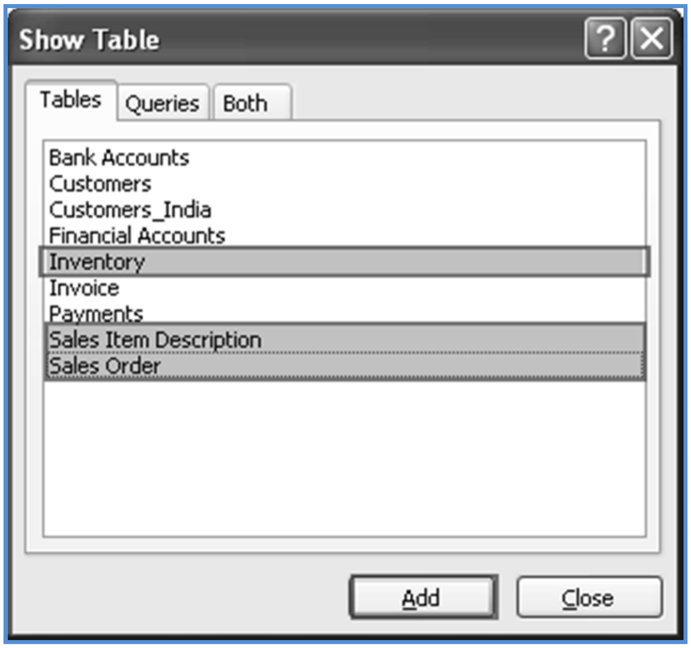
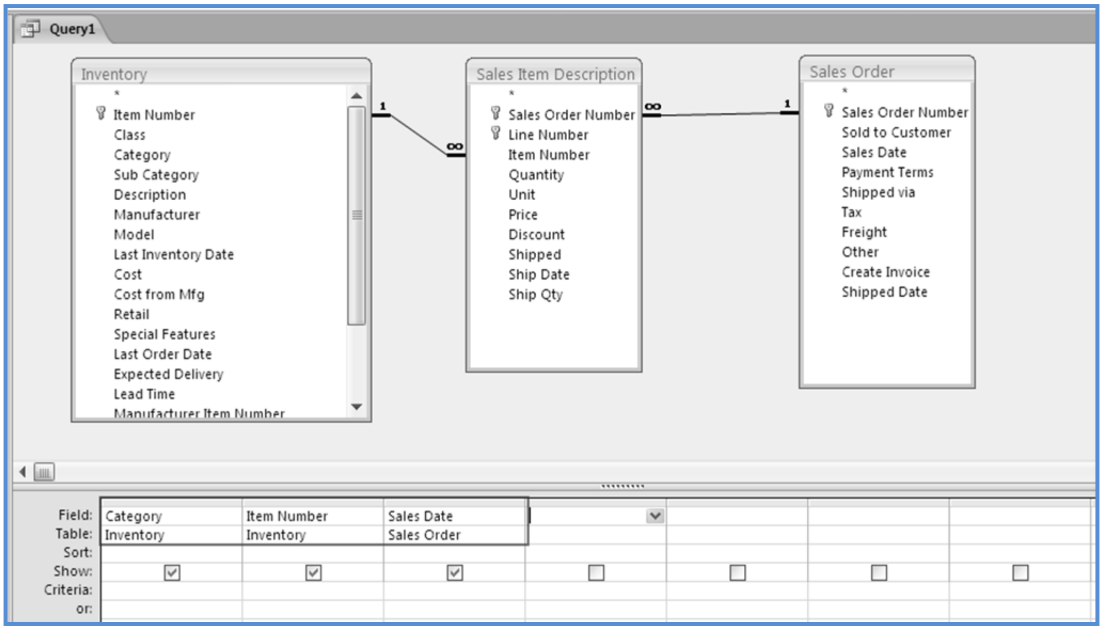


Fig. 1.2.52: Show Table window

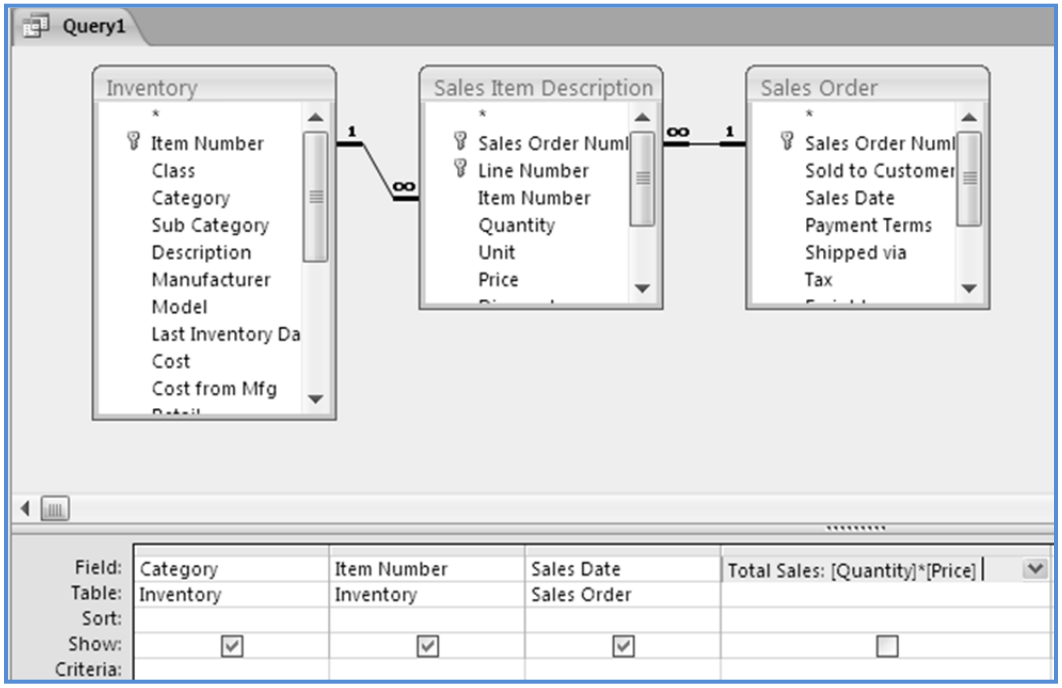
3. Select the columns *Category* and *Item Number* from *Inventory* Table and drag them to the columns tab. Similarly, select column *Sales Date* from table *Sales Order* and drag it to the columns tab. The query window appears as in Fig. 1.2.53.



Field:	Category	Item Number	Sales Date				
Table:	Inventory	Inventory	Sales Order				
Sort:							
Show:	☑	☑	☑	☐	☐	☐	☐
Criteria:							
or:							

Fig. 1.2.53: Select the required columns

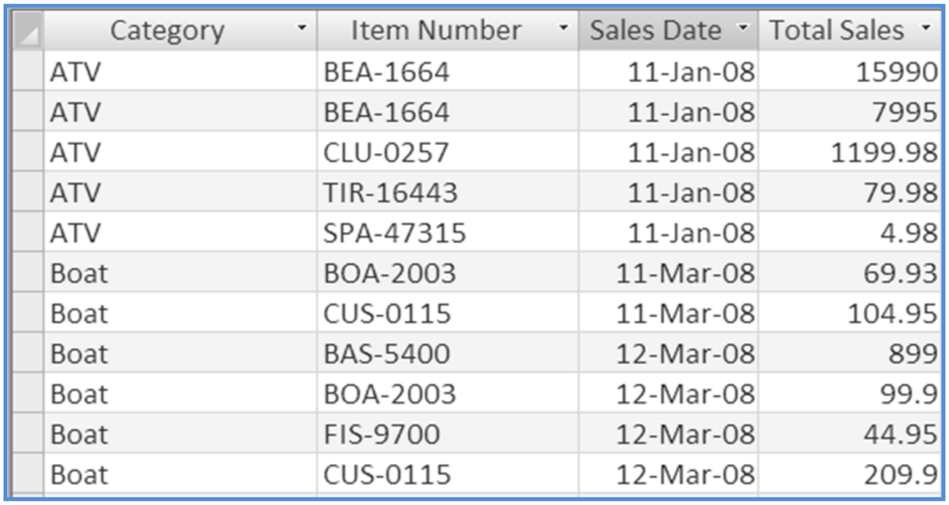
4. Now to compute the Total Sales of the Product, multiply *Quantity* * *Price* and create a new column. Write the expression *Total Sales: [Quantity]*[Price]* in the **Field** tab of the next column as displayed in Fig. 1.2.54.



Field:	Category	Item Number	Sales Date	Total Sales: [Quantity]*[Price]
Table:	Inventory	Inventory	Sales Order	
Sort:				
Show:	☑	☑	☑	☐
Criteria:				

Fig. 1.2.54: Compute Total Sales

5. Click **Run** in **Design** –> **Results** to view the query result. The result appears as shown in Fig. 1.2.55.



Category	Item Number	Sales Date	Total Sales
ATV	BEA-1664	11-Jan-08	15990
ATV	BEA-1664	11-Jan-08	7995
ATV	CLU-0257	11-Jan-08	1199.98
ATV	TIR-16443	11-Jan-08	79.98
ATV	SPA-47315	11-Jan-08	4.98
Boat	BOA-2003	11-Mar-08	69.93
Boat	CUS-0115	11-Mar-08	104.95
Boat	BAS-5400	12-Mar-08	899
Boat	BOA-2003	12-Mar-08	99.9
Boat	FIS-9700	12-Mar-08	44.95
Boat	CUS-0115	12-Mar-08	209.9

Fig. 1.2.55: Query Result

Now, we have achieved the desired result from the query. However, the Summary Report through *PivotTable* is required to be created.

Steps for Creating PivotTable from the query

1. Open the query in **PivotTable View** by clicking **PivotTable View** from **Home** -> **Views**, as displayed in Fig. 1.2.56.

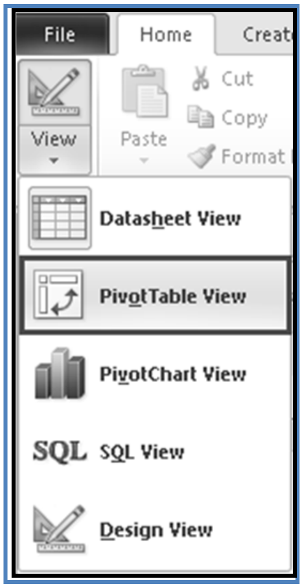


Fig. 1.2.56: Select PivotTable View

2. The PivotTable view appears with the field list on right side as displayed in Fig. 1.2.57.

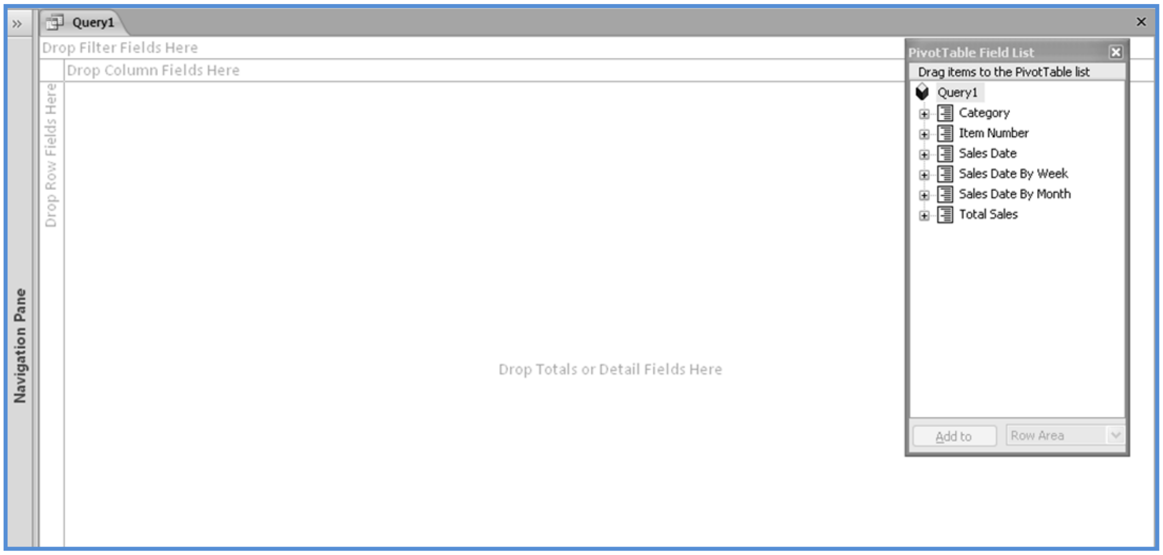


Fig. 1.2.57: PivotTable View

NOTE: If the **PivotTable Field List** does not appear, obtain the list by clicking **Field List** from **Show/Hide** group on the **Design** tab as indicated in Fig. 1.2.58.

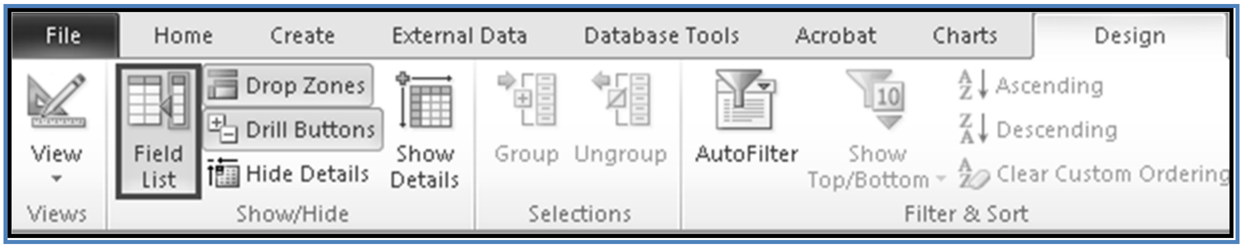
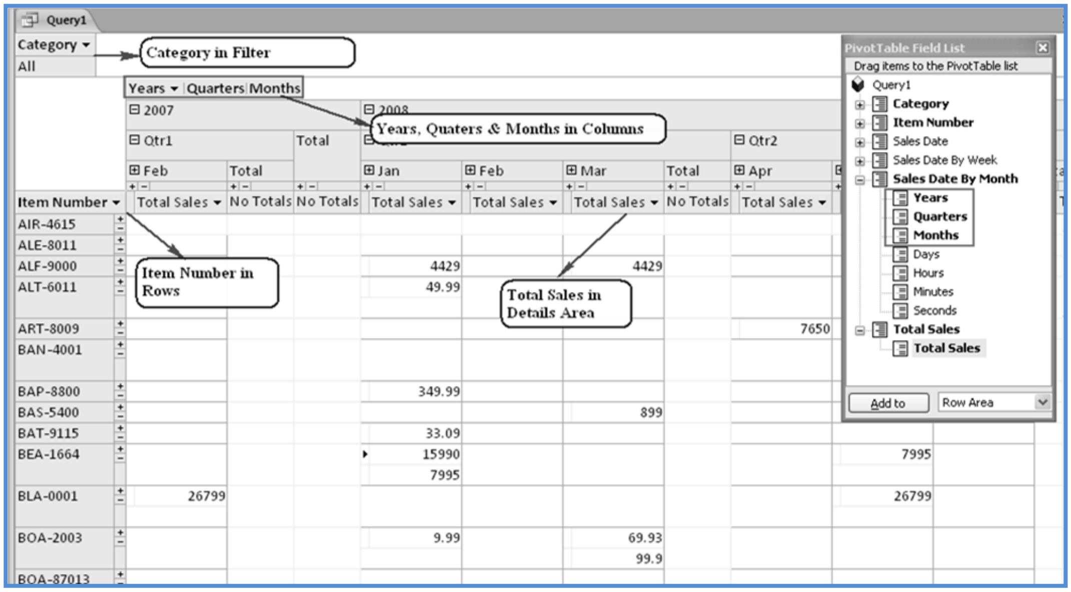


Fig. 1.2.58: Field List Button

3. Select the *Category* field from **PivotTable Field List** and drag it to the **Drop Filter Fields Here** area.
4. Select the *Item Number* field from **PivotTable Field List** and drag it to the **Drop Row Fields Here** area.
5. Select the Year by expanding the *Sales Date By Month* field and drag it to the **Drop Column Fields Here** area. Similarly, select and drag Quarter and Month to the **Drop Column Fields Here** area.
6. Select and drag the *Total Sales* field to the **Drop Totals or Detail Fields Here** area. The *PivotTable window* should appear as in Fig. 1.2.59.



	2007			2008					
	Qtr1		Total					Qtr2	
	Feb	Total		Jan	Feb	Mar	Total	Apr	
Item Number	Total Sales	No Totals	No Totals	Total Sales	Total Sales	Total Sales	No Totals	Total Sales	
AIR-4615									
ALE-8011									
ALF-9000				4429		4429			
ALT-6011				49.99					
ART-8009								7650	
BAN-4001									
BAP-8800				349.99					
BAS-5400						899			
BAT-9115				33.09					
BEA-1664				15990 7995					7995
BLA-0001	26799								26799
BOA-2003				9.99		69.93 99.9			
BOA-87013									

Fig. 1.2.59: Drag Fields in PivotTable Area

7. Click **Save** at *Quick Access Toolbar* to save the query. Type the name of the query as “Sales Summary Report”.

If we wish to see the Quarterly sales of all the Products from Personal Watercraft category, the *PivotTable* query is obtained as follows:

8. Open the drop-down in *Category* Filter Area, clear all checkboxes and check the *Personal Watercraft* and click **OK** as in Fig. 1.2.60.

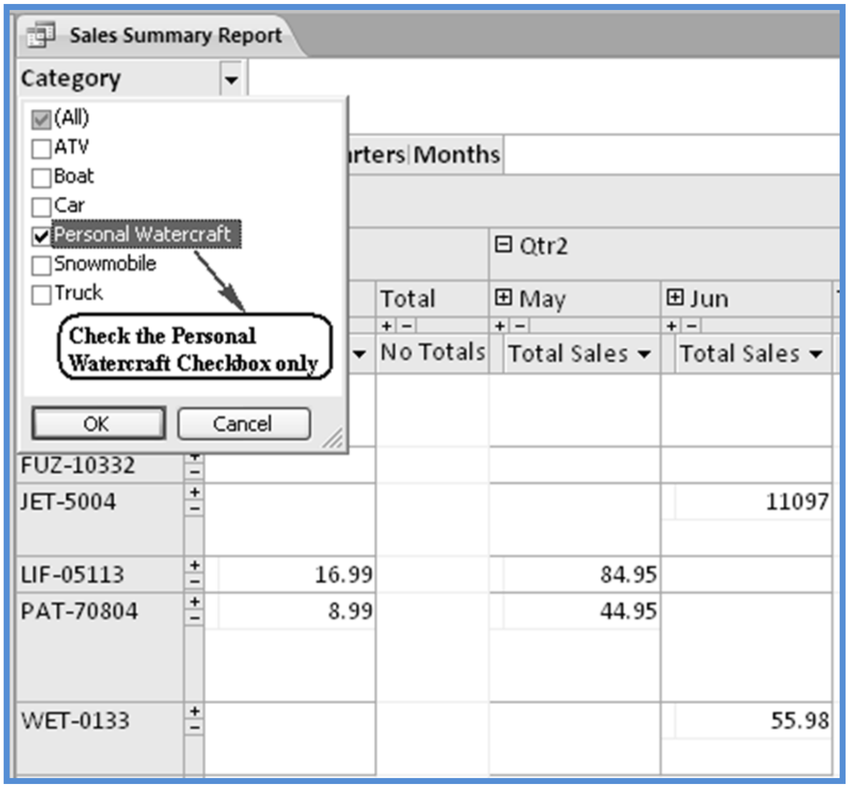
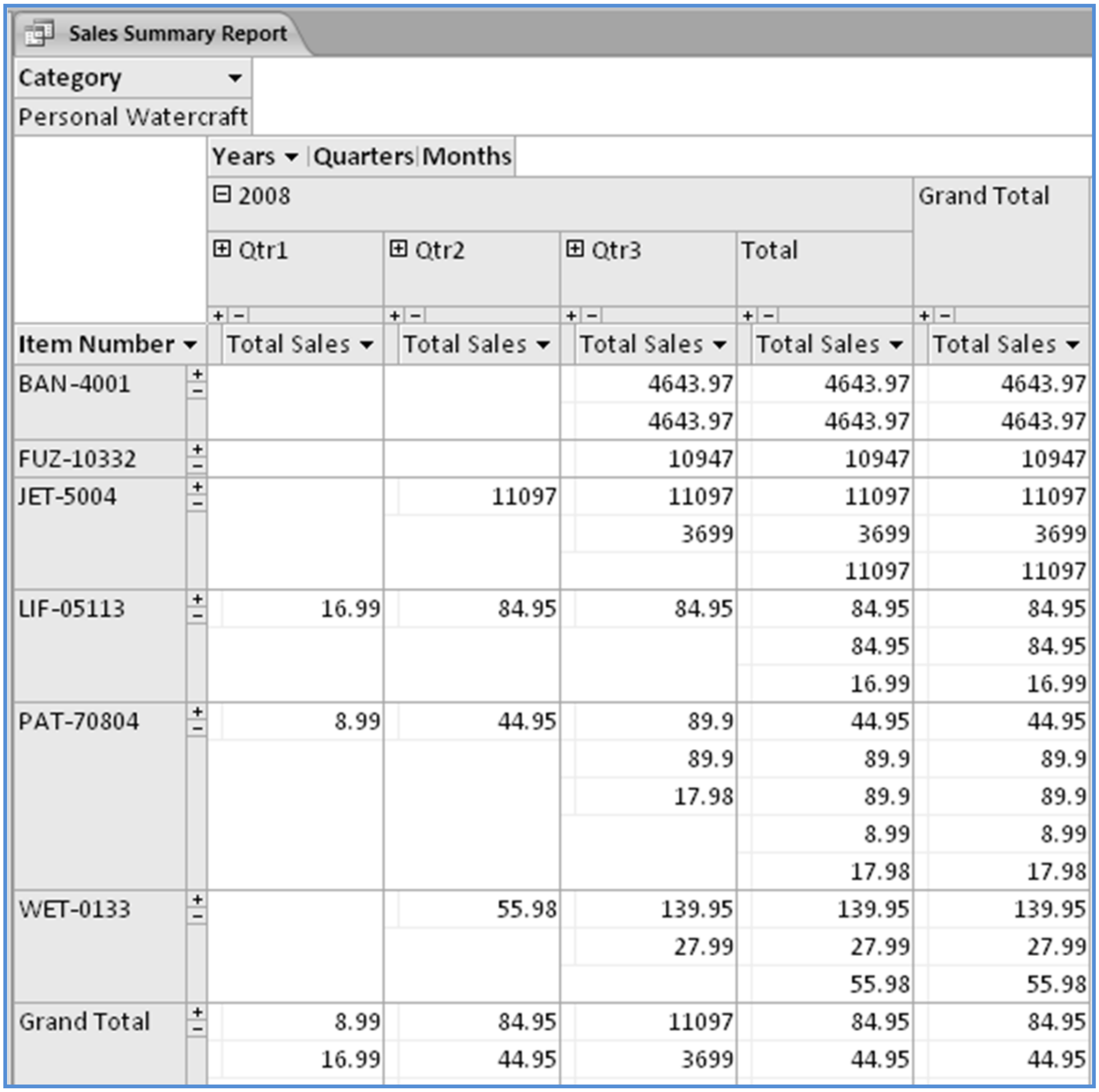


Fig. 1.2.60: Select Personal Watercraft from Category Filter

9. Click on the negative sign on *Quarter* tab so that the months disappear. Also, click on the plus sign with *Totals* to make the Grand Total appear. Make sure that all the plus signs in *Item Number* Row are clicked so as to make the Total Sales value appear. The final query should appear as shown in Fig. 1.2.61.



Sales Summary Report

Category ▾
Personal Watercraft

Years ▾ | Quarters | Months

	⊟ 2008				Grand Total
	⊞ Qtr1	⊞ Qtr2	⊞ Qtr3	Total	
Item Number ▾	Total Sales ▾	Total Sales ▾	Total Sales ▾	Total Sales ▾	Total Sales ▾
BAN-4001			4643.97	4643.97	4643.97
			4643.97	4643.97	4643.97
FUZ-10332			10947	10947	10947
JET-5004		11097	11097	11097	11097
			3699	3699	3699
				11097	11097
LIF-05113	16.99	84.95	84.95	84.95	84.95
				84.95	84.95
				16.99	16.99
PAT-70804	8.99	44.95	89.9	44.95	44.95
			89.9	89.9	89.9
			17.98	89.9	89.9
				8.99	8.99
				17.98	17.98
WET-0133		55.98	139.95	139.95	139.95
			27.99	27.99	27.99
				55.98	55.98
Grand Total	8.99	84.95	11097	84.95	84.95
	16.99	44.95	3699	44.95	44.95

Fig. 1.2.61: Quarterly sales of all the Products from Personal Watercraft category

1.2.5.3 PivotChart

A PivotChart is a tool used for graphical analysis of data. In simple terms, PivotChart helps visualize a PivotTable, Query or a Form. It can display summarized data in different chart formats and enables data analysis. Data can be presented by using different chart formats as required, and unwanted items can be hidden from being viewed.

The *PivotChart* view in a query can be obtained by selecting the **PivotChart View** from **Home** -> **Views**, as indicated in Fig. 1.2.62.

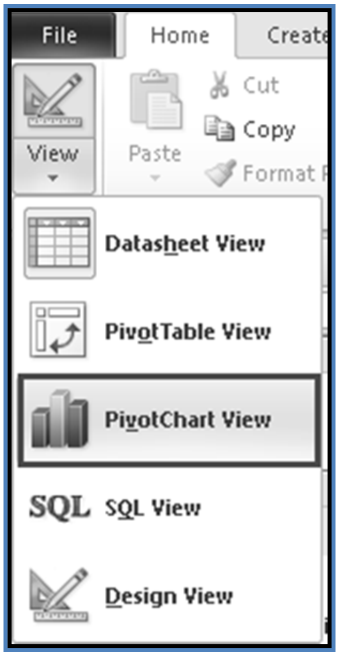


Fig. 1.2.62: PivotChart View

A sample PivotChart indicating sales of a Category of Products in each Quarter in a year is shown in Fig. 1.2.63.

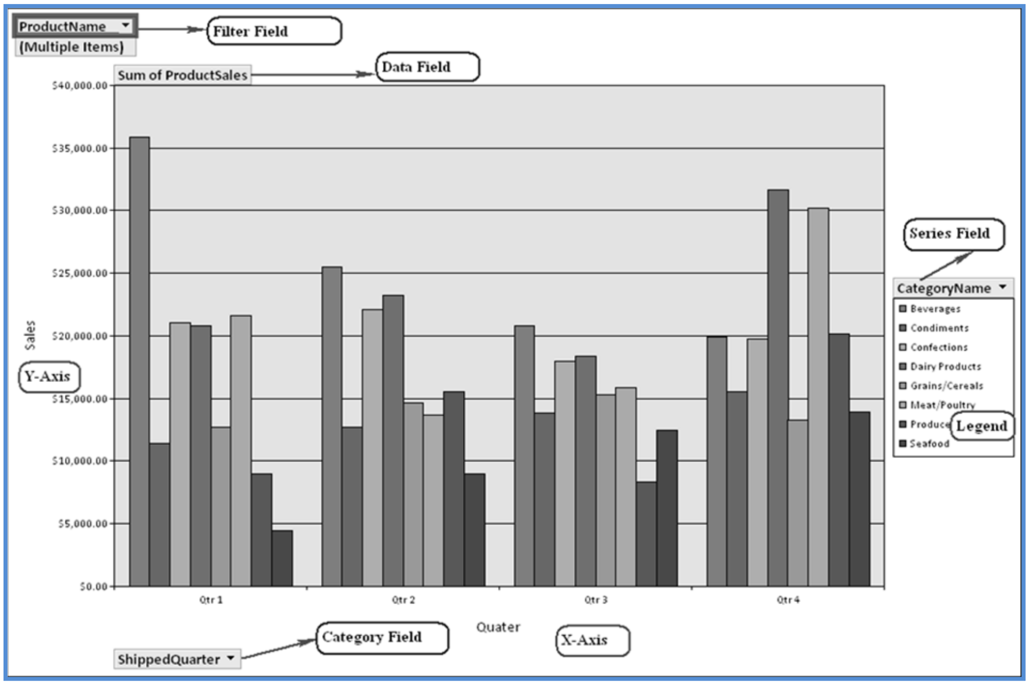


Fig. 1.2.63: PivotChart View

Description of the various headings marked in PivotChart above is given in Table 2.2.3.

Headings	Description
X-Axis	The horizontal axis in PivotChart
Y-Axis	The vertical axis in PivotChart
Legend	A table displaying the color code used for each data series in PivotChart
Filter Field	A field on basis of which filter can be applied on PivotChart.
Data Field	Field values to be shown along the Y-axis.
Category Field	Field values to be shown along the X-axis.
Series Field	Field values that will form the legend of the graph.

Table 1.2.3: Various fields in PivotChart

1.2.5.4 Problem Scenario

Considering the Summary Sales Report discussed in Section 1.2.5.1, a graphical representation of the same data is required.

Solution

To represent the data graphically, we need to create a *PivotChart*. For this purpose, we first create a query to display Summary Report and then create a *PivotChart* based on that query. In the Pivot Chart, the fields will be placed as:

- *Category* as Filter Field
- *Years* & *Quarters* field on X-Axis
- *Total Sales* on Y-Axis
- *Item Number* as Series Field

Steps to create a query of Summary Report

Follow the steps discussed in Section 1.2.5.1 to create a query *Sales Summary Report* and create another query *Graph of Sales Summary Report* with columns as *Category, Item Number, Sales Date* and *Total Sales (Quantity * Price)*.

Steps to create a copy of Summary Report

1. Double-click the *Graph of Sales Summary Report* query to open it in the Datasheet view.
2. Open the PivotChart view of the query by selecting **PivotChart View** from **Home** -> **Views** drop-down list. The PivotChart view appears as in Fig. 1.2.64.

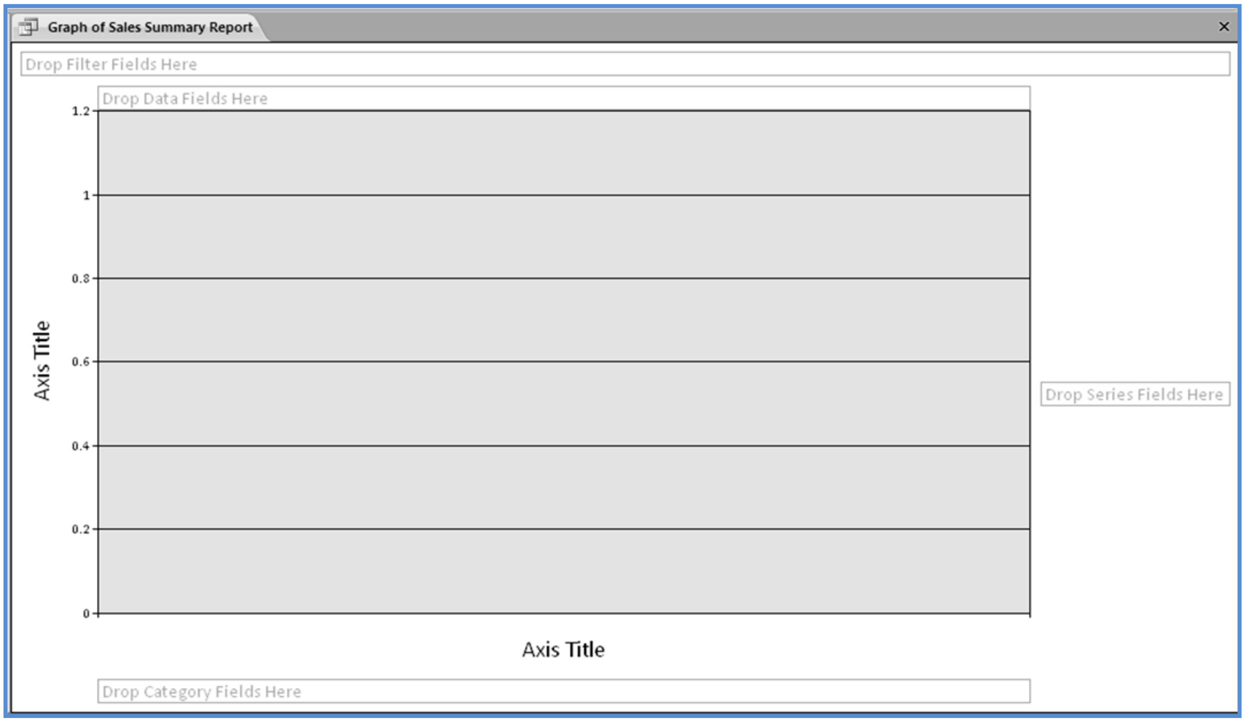


Fig. 1.2.64: PivotChart View

3. If the **Chart Field List** is not there, select the **Field List** option from the **Show/Hide** group on the **Design** tab.

4. Select the *Category* field from **Chart Field List** and drag it to the **Drop Filter Fields Here** area.

5. Select *Years* by expanding the *Sales Date By Month* field and drag it to the **Drop Category Fields Here** area. Similarly, select and drag *Quarters* to the **Drop Category Fields Here** area.

6. Select the *Item Number* field and drag it to the **Drop Series Fields Here** area.

7. Select and drag the *Total Sales* field to the **Drop Data Fields Here** area.

8. To display the legend, select **Legend** from the **Show/Hide** group on the **Design** tab, as indicated in Fig. 1.2.65.

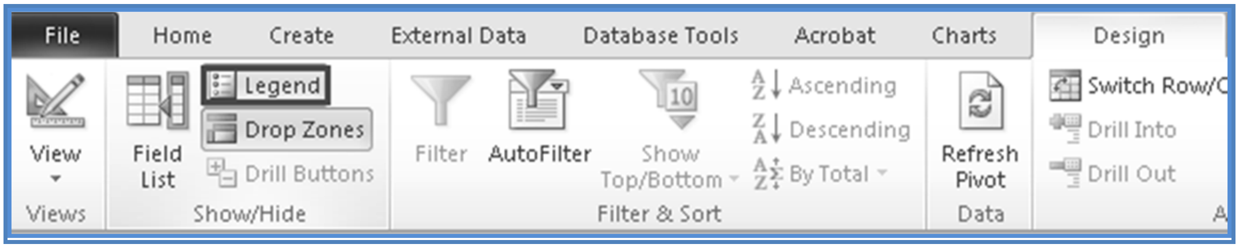


Fig. 1.2.65: Select Legend option

9. The *PivotChart* window should appear, as shown in Fig. 1.2.66.

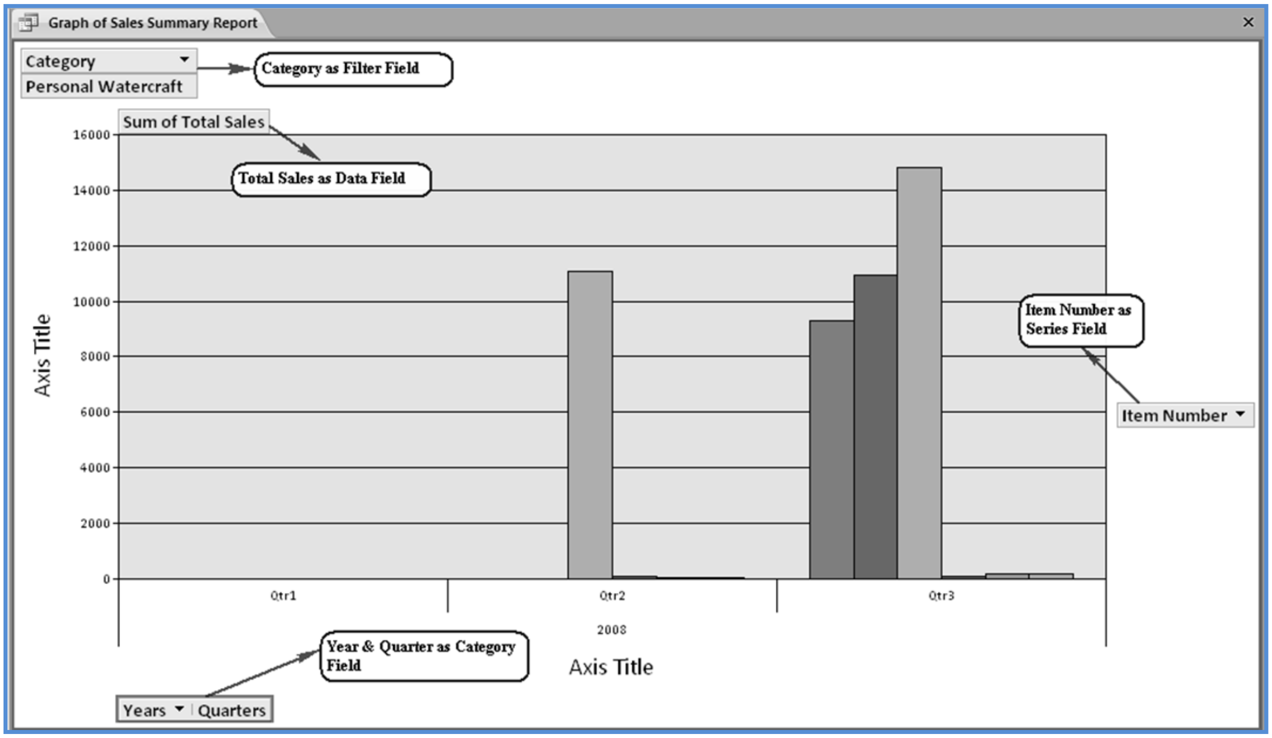


Fig. 1.2.66: Drag and Drop Fields in PivotChart Area

NOTE: The Total Sales of few products is much lesser as compared to other products that they are hardly visible on data bar. To solve this problem, the axis has to be changed to Logarithmic axis so as to make data bars more visible.

Steps for changing the axis to logarithmic axis

1. Right-click on any value in Y-axis so the entire scale is selected and select **Properties** from the menu, as shown in Fig. 1.2.67.

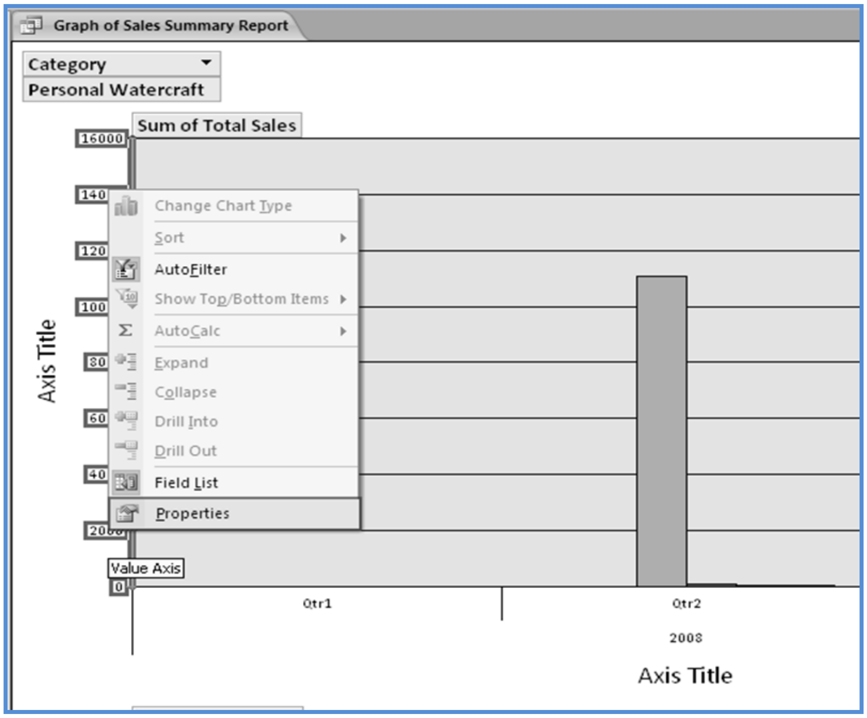
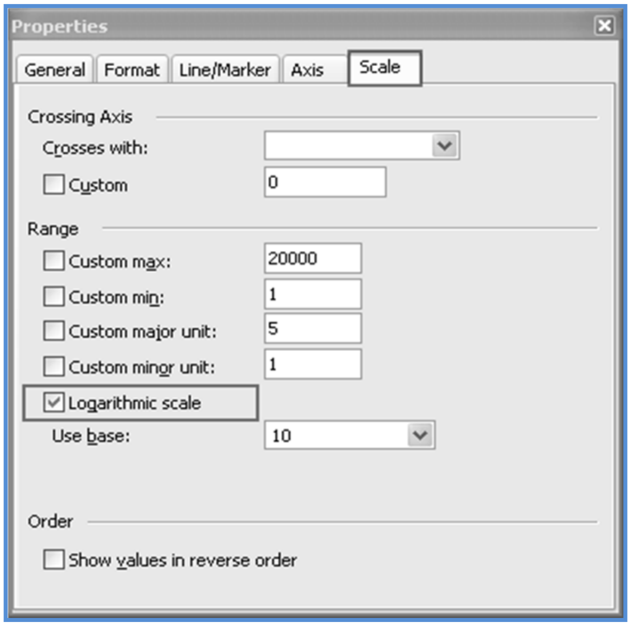


Fig. 1.2.67: Select properties for Y-Axis

2. In the **Properties** window that appears, go to the **Scale** tab and check **Logarithmic Scale** checkbox as displayed in Fig. 1.2.68.


Properties

General | Format | Line/Marker | Axis | Scale

Crossing Axis

Crosses with:

□ Custom 0

Range

□ Custom max: 20000

□ Custom min: 1

□ Custom major unit: 5

□ Custom minor unit: 1

☑ Logarithmic scale

Use base: 10

Order

□ Show values in reverse order

Fig. 1.2.68: Scale Properties

3. Close the **Properties** window.

In order to make graph more explanatory, we will provide a name to X-Axis and Y-Axis of the graph.

Steps for naming Axis in PivotChart

1. Right-click the **Axis Title** on X-Axis and select **Properties** from the menu, as shown in Fig. 1.2.69.

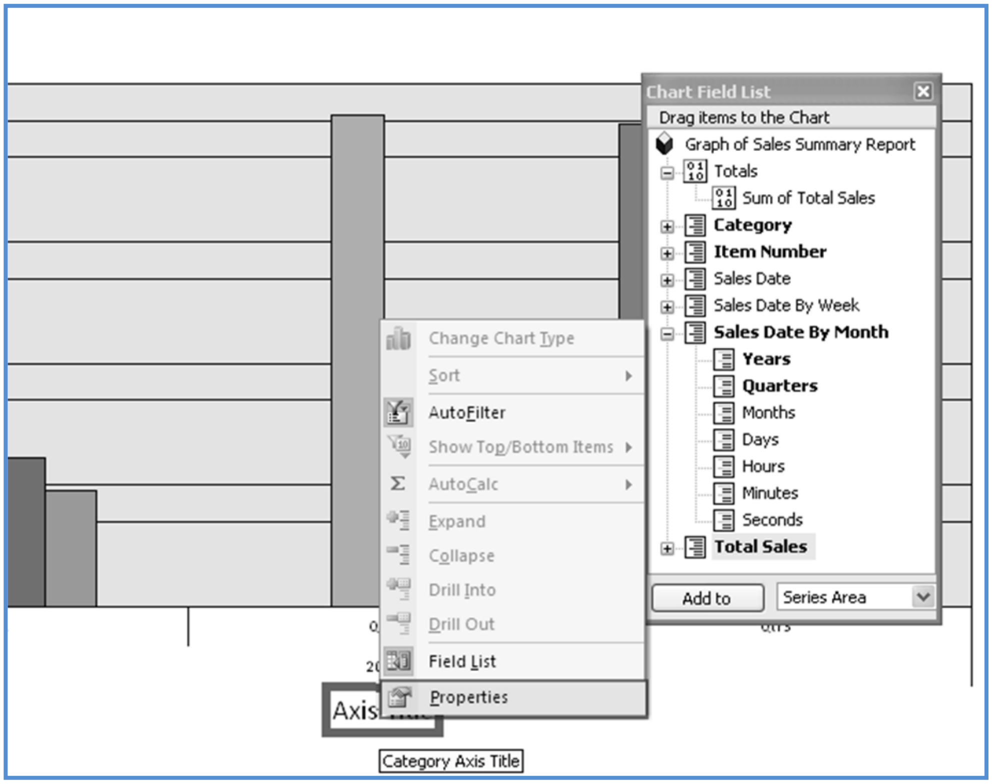
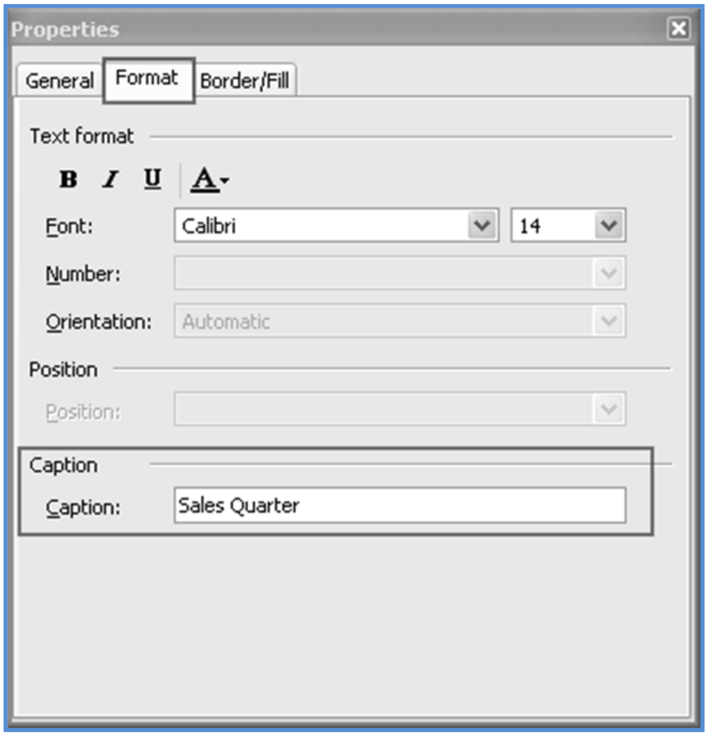


Fig. 1.2.69: Open X-Axis Properties

2. In the **Properties** window, go to the **Format** tab and change the **Caption** property to *Sales Quarter* as in Fig. 1.2.70.



Properties

General | Format | Border/Fill

Text format

Font: Calibri 14

Number:

Orientation: Automatic

Position

Position:

Caption

Caption: Sales Quarter

Fig. 1.2.70: Change caption of X-Axis

3. In a similar manner, change the caption of Y-Axis to *Total Sales*. The *PivotChart* appears as in Fig. 1.2.71.

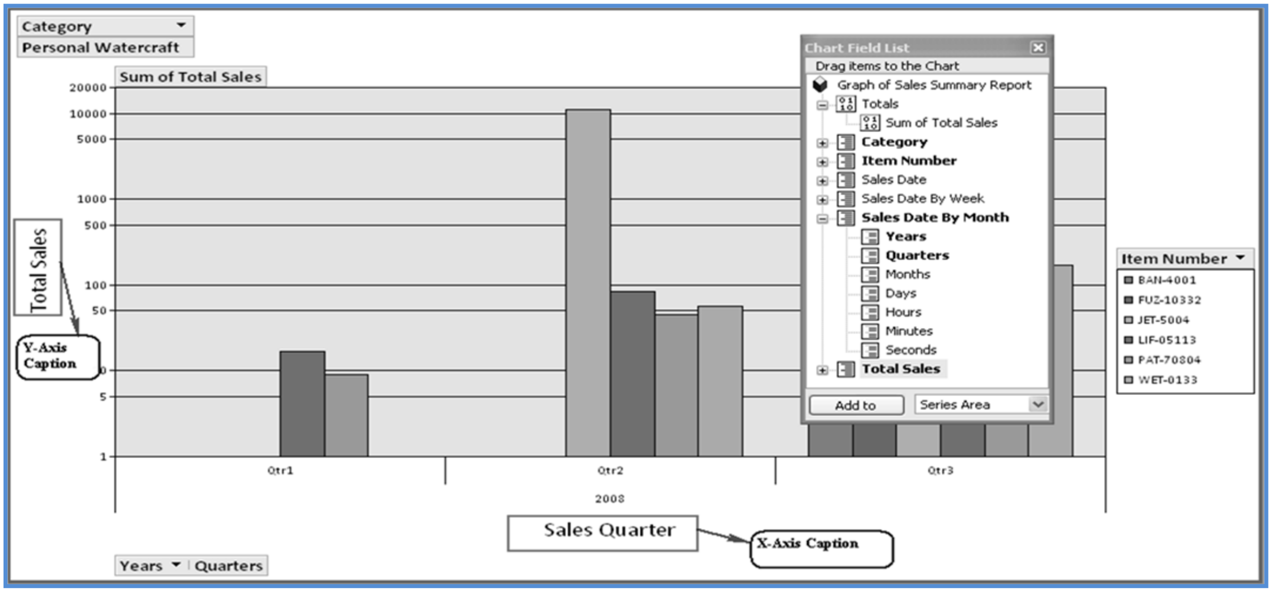


Fig. 1.2.71: Pivot Chart with Axis Captions

Now if we wish to see the chart of quarterly sales of all the products from Truck category for the year 2011, the *PivotChart* will be obtained as follows:

4. Open the drop-down in the *Category* Filter area, clear all checkboxes check the *Truck* checkbox and click **OK** as in Fig. 1.2.72.

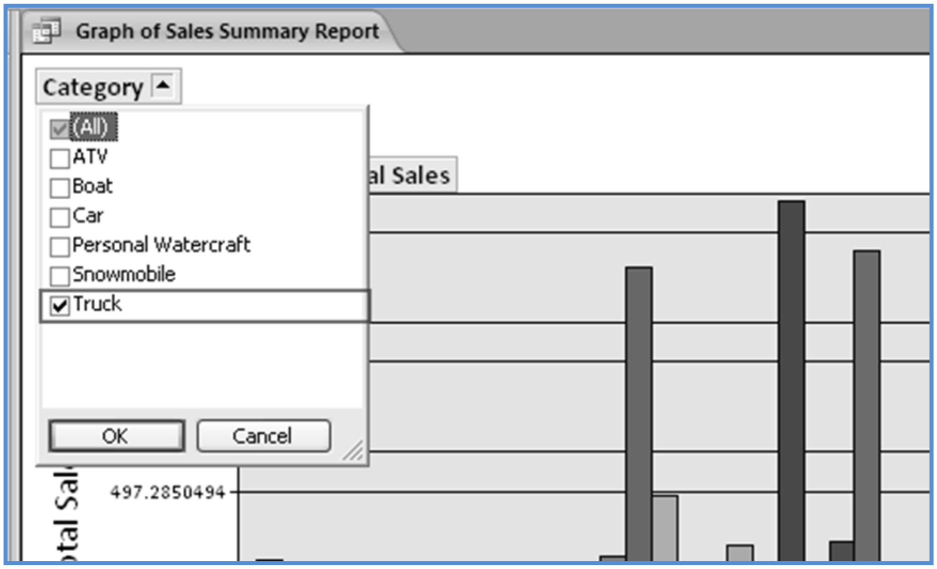


Fig. 1.2.72: Select Truck from Category

5. In a similar manner, select *2011* from *Years* drop-down in horizontal axis. The resultant chart appears as in Fig. 1.2.73 indicating the sale of different items of *Truck* category, which is maximum in Quarter2 and minimum in Quarter4.

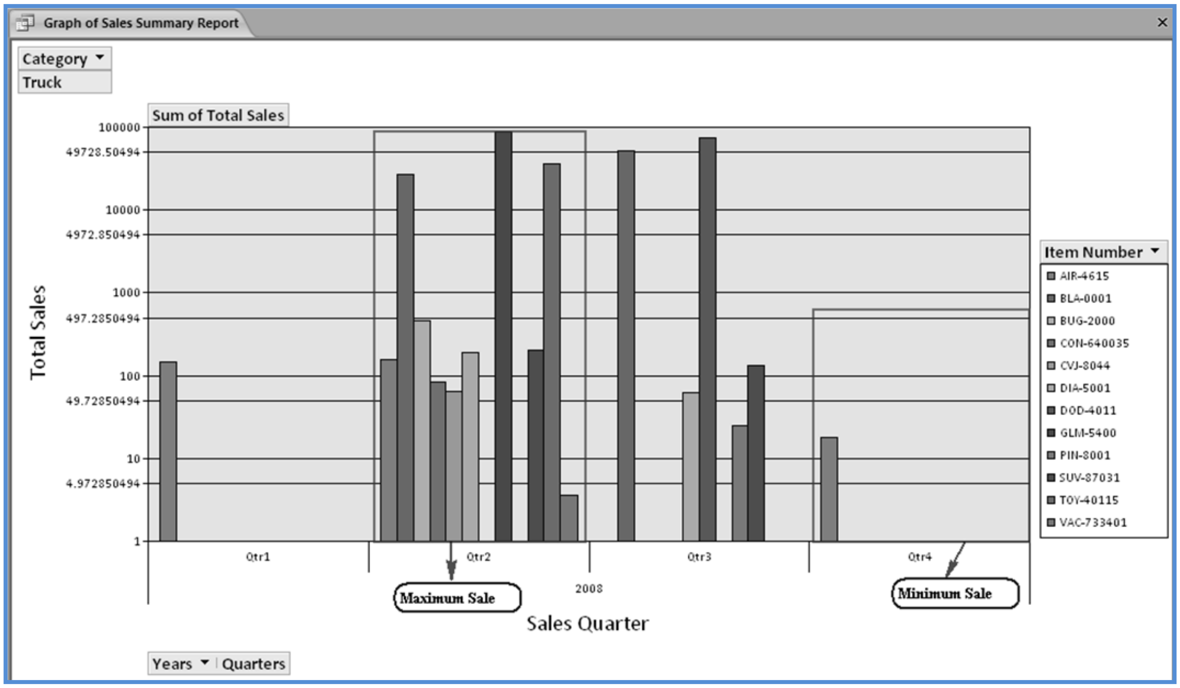


Fig. 1.2.73: Quarter-wise Truck Sales

NOTE: Different PivotCharts can be made to represent different kinds of data. Both PivotChart and PivotTable can also be made directly on tables rather than making a query first.

1.3 Joining Tables in Queries

A Join is a temporary relationship that is created between two tables in a query or the relationship window using a common field in both the tables having same data type and usually the same name. Joins created in a query are temporary and are meant for the current query only. Joining tables in a query is required to view data from two or more tables. For example, to retrieve products ordered by each customer, tables *Customers*, Sales *Order* and *Inventory* are required to be joined to get the desired output.

When two or more tables are added to the *Query Design* window, Access 2010 creates Joins between them based on the relationships that have been defined in the relationship window. Joins establish the criteria that the data must match to be included in the query operations. If the tables are not joined, the query result will match each record of one table to every record in another table, resulting in spurious rows. Different types of joins are available to get a different set of records as query result.

There are three types of joins available in Access 2010: inner join, left outer join, and right outer join.

Inner Join: Returns only those rows from both tables that match on the joining field.

Left Outer Join: The query returns all of the rows from left table, and also those rows from the right table that share a common value on both sides of the join. Since some of the rows in left table of a left outer join will not have corresponding rows in right table, some of the fields returned as a query result will be empty when the rows do not correspond.

Right Outer Join: It is just the opposite of left outer join. The query using right outer join returns all the rows from right table, and also those rows from the left table that share a common value on both sides of the join.

1.3.1 Problem Scenario

The Operations Regional Head of Apex Ltd. is required to produce a report giving the details of the payments received.

For this purpose, two reports are required to be prepared:

- First report indicating the Invoices for which payments have been made, including the invoice and payments detail.
- Second report displaying a list of all the sales order, their invoices and details of payments, including those invoices for which payments have not been received.

Solution

To get the desired result, we create two queries:

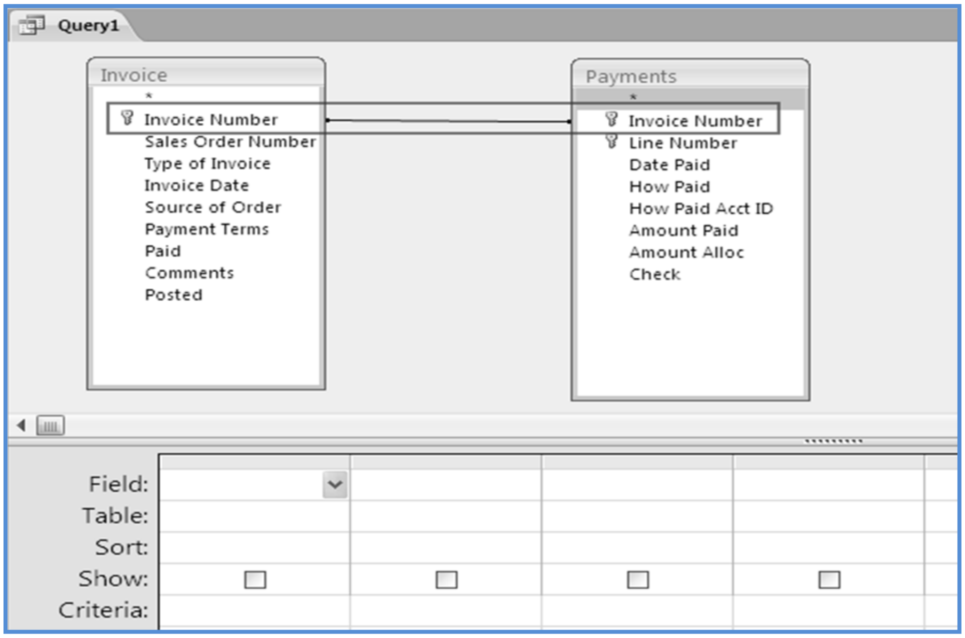
First query to fulfill the requirement using inner join between tables *Invoice* and *Payments.*

Second query to achieve the second requirement which includes *Sales Order*, *Invoice* and *Payments* tables with an left outer join between *Invoice* and *Payments* tables.

Steps to create the first query - to obtain invoices for which payments have been made

1. Open the *Query Design* window, by clicking **Create** -> **Queries** -> **Query Design**.
2. Select the *Invoice* and *Payments* tables from **Show Table** window and click **Add**. Click **Close**.
3. To join the two tables, click the *Invoice Number* from *Invoice* table and drag it to *Invoice Number* in *Payments* table. The query window appears, as shown in Fig. 1.3.1.

   NOTE: If the tables are already related through relationship window, the joining line automatically appears between the tables.


Query1

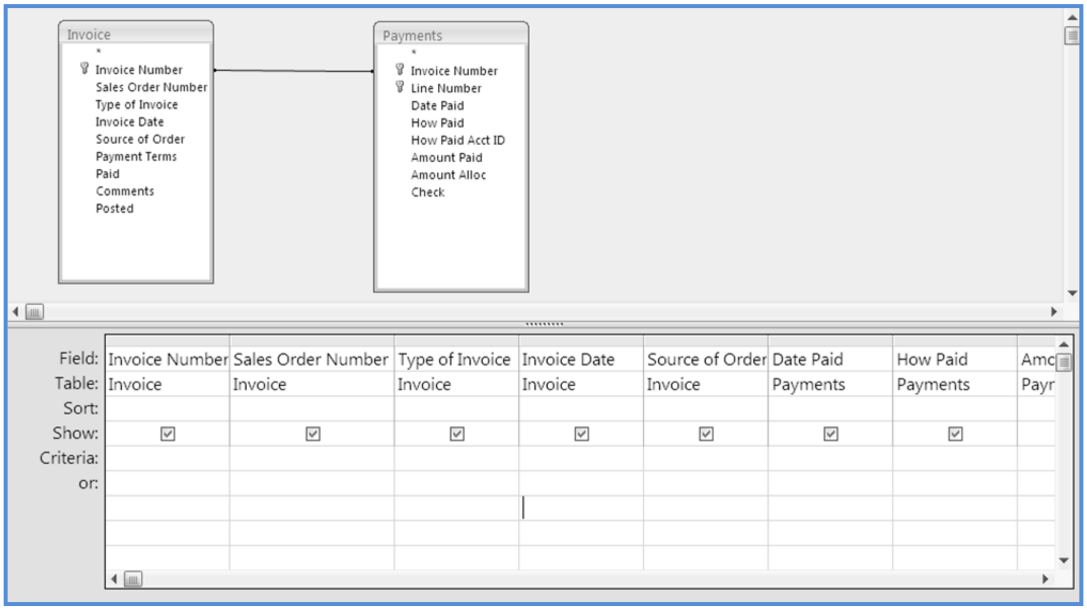
Invoice
*
Invoice Number
Sales Order Number
Type of Invoice
Invoice Date
Source of Order
Payment Terms
Paid
Comments
Posted

Payments
*
Invoice Number
Line Number
Date Paid
How Paid
How Paid Acct ID
Amount Paid
Amount Alloc
Check

Field:				
Table:				
Sort:				
Show:	□	□	□	□
Criteria:				

Fig. 1.3.1: Create Relationship between tables

4. Select the columns *Invoice Number*, *Sales Order Number*, *Type of Invoice*, *Invoice Date* and *Source of Order* from the table *Invoice* and drag them to the columns tab. Similarly, select columns *Date Paid*, *How Paid*, *Amount Paid* and *Amount Alloc* from *Payments* table and drag them to the columns tab. The query window appears as shown in Fig. 1.3.2.


Invoice
*
Invoice Number
Sales Order Number
Type of Invoice
Invoice Date
Source of Order
Payment Terms
Paid
Comments
Posted

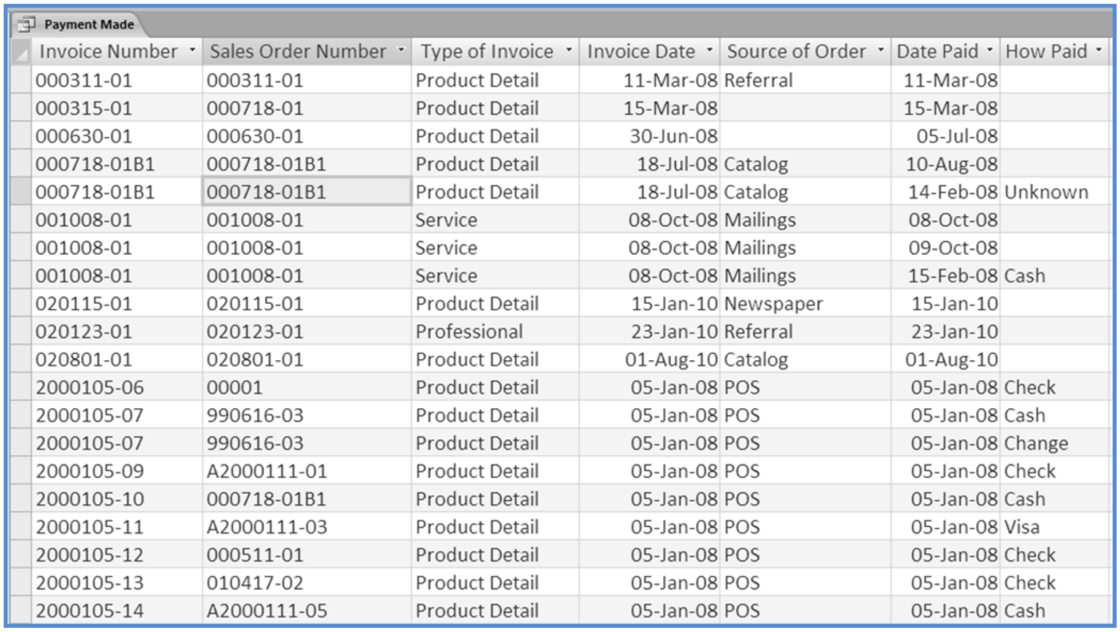
Payments
*
Invoice Number
Line Number
Date Paid
How Paid
How Paid Acct ID
Amount Paid
Amount Alloc
Check

Field:	Invoice Number	Sales Order Number	Type of Invoice	Invoice Date	Source of Order	Date Paid	How Paid	Amo
Table:	Invoice	Invoice	Invoice	Invoice	Invoice	Payments	Payments	Payr
Sort:								
Show:	☑	☑	☑	☑	☑	☑	☑	
Criteria:								
or:								

Fig. 1.3.2: Add required columns

5. Click the **Save** button on *Quick Access Toolbar* to save the query. Type the name of the query as *Payment Made*.

6. Click the **Run** icon in **Design** -> **Results** to view the query result displaying invoices with their payment details as shown in Fig. 1.3.3.



Invoice Number	Sales Order Number	Type of Invoice	Invoice Date	Source of Order	Date Paid	How Paid
000311-01	000311-01	Product Detail	11-Mar-08	Referral	11-Mar-08	
000315-01	000718-01	Product Detail	15-Mar-08		15-Mar-08	
000630-01	000630-01	Product Detail	30-Jun-08		05-Jul-08	
000718-01B1	000718-01B1	Product Detail	18-Jul-08	Catalog	10-Aug-08	
000718-01B1	000718-01B1	Product Detail	18-Jul-08	Catalog	14-Feb-08	Unknown
001008-01	001008-01	Service	08-Oct-08	Mailings	08-Oct-08	
001008-01	001008-01	Service	08-Oct-08	Mailings	09-Oct-08	
001008-01	001008-01	Service	08-Oct-08	Mailings	15-Feb-08	Cash
020115-01	020115-01	Product Detail	15-Jan-10	Newspaper	15-Jan-10	
020123-01	020123-01	Professional	23-Jan-10	Referral	23-Jan-10	
020801-01	020801-01	Product Detail	01-Aug-10	Catalog	01-Aug-10	
2000105-06	00001	Product Detail	05-Jan-08	POS	05-Jan-08	Check
2000105-07	990616-03	Product Detail	05-Jan-08	POS	05-Jan-08	Cash
2000105-07	990616-03	Product Detail	05-Jan-08	POS	05-Jan-08	Change
2000105-09	A2000111-01	Product Detail	05-Jan-08	POS	05-Jan-08	Check
2000105-10	000718-01B1	Product Detail	05-Jan-08	POS	05-Jan-08	Cash
2000105-11	A2000111-03	Product Detail	05-Jan-08	POS	05-Jan-08	Visa
2000105-12	000511-01	Product Detail	05-Jan-08	POS	05-Jan-08	Check
2000105-13	010417-02	Product Detail	05-Jan-08	POS	05-Jan-08	Check
2000105-14	A2000111-05	Product Detail	05-Jan-08	POS	05-Jan-08	Cash

Fig. 1.3.3: Query Result

Steps to create the second query - to display list of all the sales order, their invoices and payments details, including those invoices for which payments have not been received

1. Open the *Query Design* window, by clicking **Create** -> **Queries** -> **Query Design**.

2. Select the *Sales Order*, *Invoice* and *Payments* table from **Show Table** window and click **Add**. Click **Close**.

3. To join the tables, click the *Sales Order Number* from *Sales Order* table and drag it to the *Sales Order Number* in *Invoice* table. Similarly, join *Invoice* and *Payments* table on *Invoice Number* field. The query window appears as shown in Fig. 1.3.4.

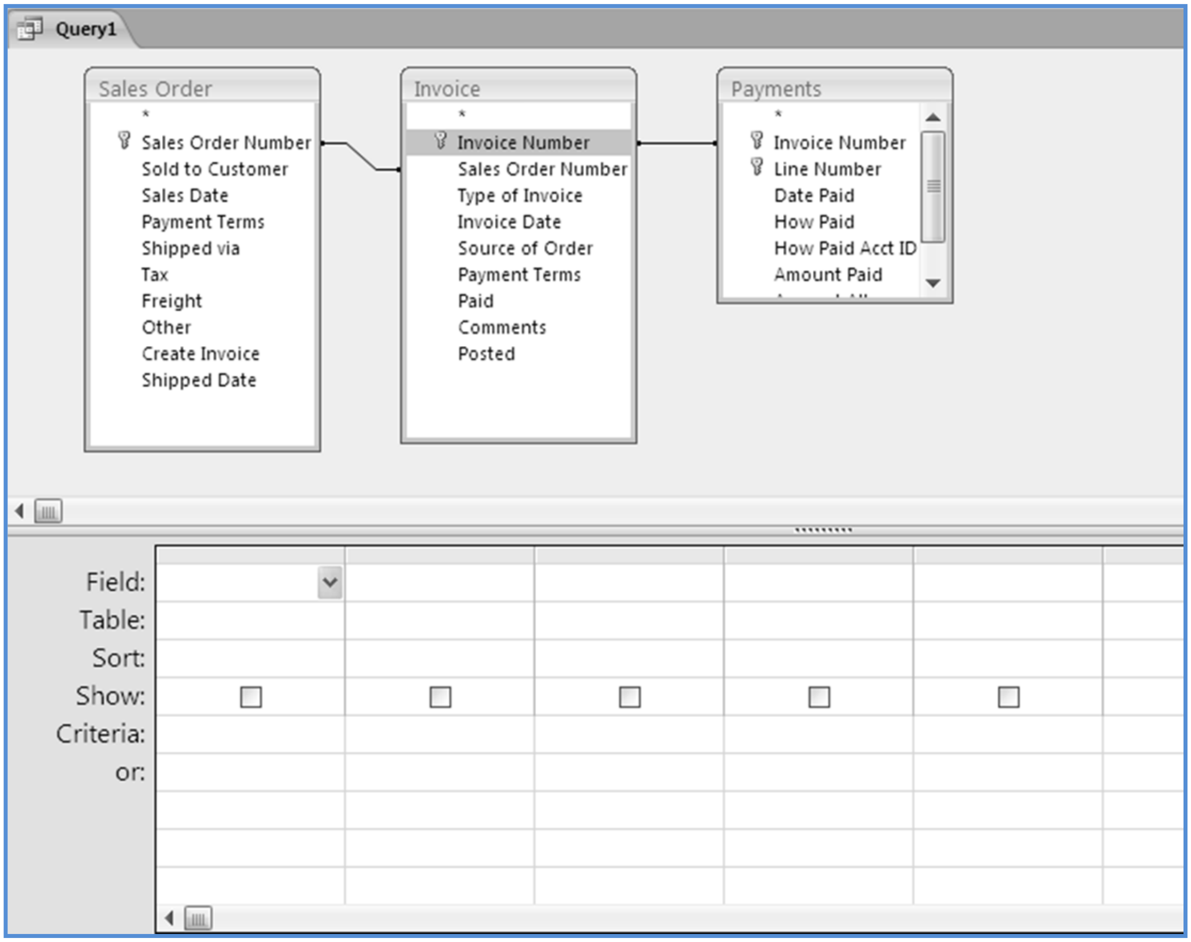
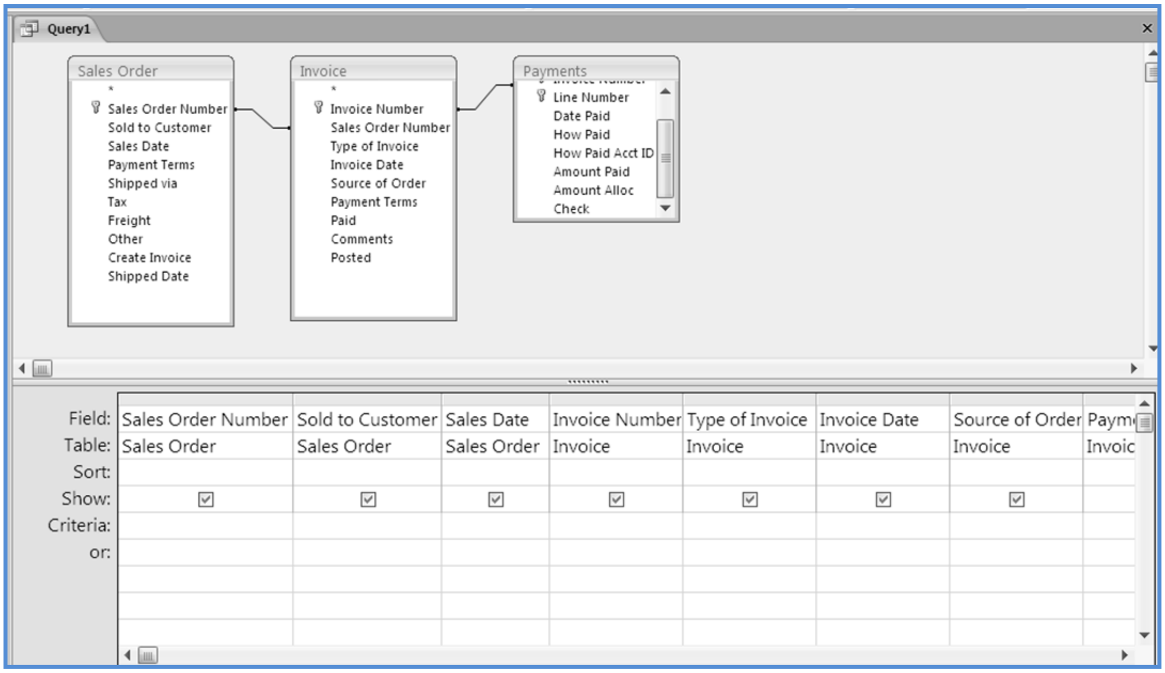


Fig. 1.3.4: Join Tables

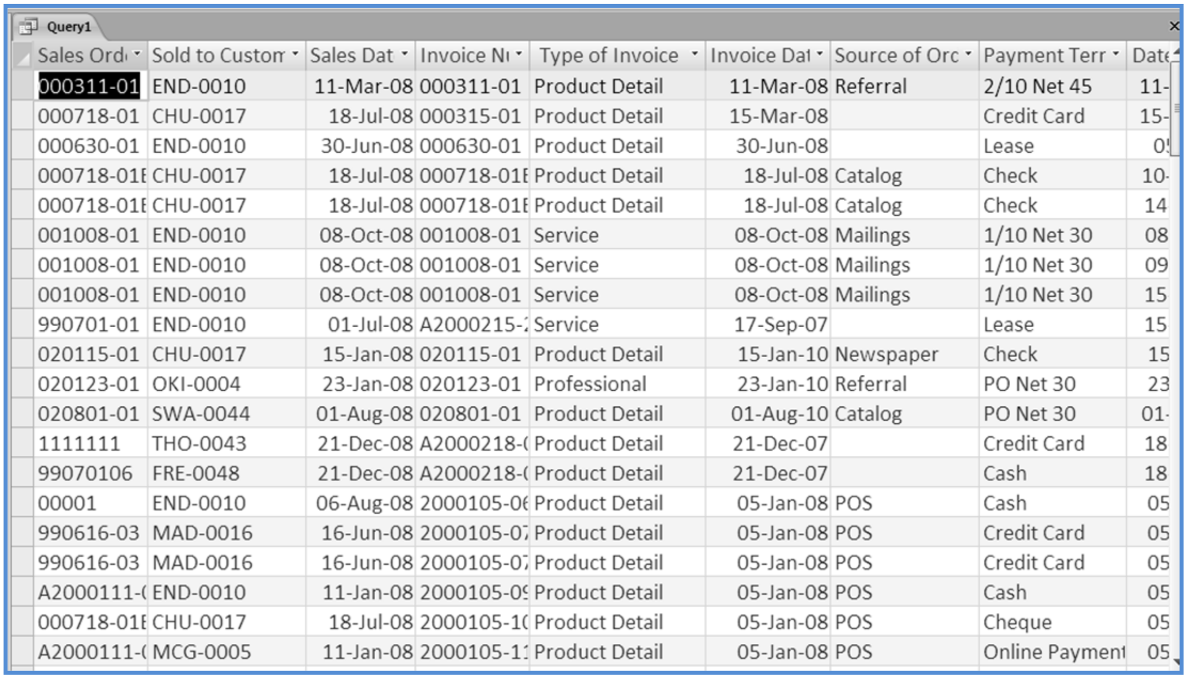
4. Select the fields *Sales Order Number*, *Sold to Customer* and *Sales Date* from *Sales Order* table and drag them to the Columns tab. Similarly, select fields *Invoice Number*, *Type of Invoice*, *Invoice Date*, *Source of Order* and *Payment Terms* from *Invoice* table and columns *Date Paid*, *How Paid*, *Amount Paid* and *Amount Alloc* from *Payments* table. The query window appears as in Fig. 1.3.5.



Field:	Sales Order Number	Sold to Customer	Sales Date	Invoice Number	Type of Invoice	Invoice Date	Source of Order	Paym
Table:	Sales Order	Sales Order	Sales Order	Invoice	Invoice	Invoice	Invoice	Invoic
Sort:								
Show:	☑	☑	☑	☑	☑	☑	☑	
Criteria:								
or:								

Fig. 1.3.5: Select required columns

5. Click **Run** Run in **Design -> Results** to view the query result as shown in Fig. 1.3.6.



Sales Ord	Sold to Custom	Sales Dat	Invoice Nu	Type of Invoice	Invoice Dat	Source of Orc	Payment Terr	Date
000311-01	END-0010	11-Mar-08	000311-01	Product Detail	11-Mar-08	Referral	2/10 Net 45	11-
000718-01	CHU-0017	18-Jul-08	000315-01	Product Detail	15-Mar-08		Credit Card	15-
000630-01	END-0010	30-Jun-08	000630-01	Product Detail	30-Jun-08		Lease	05
000718-01E	CHU-0017	18-Jul-08	000718-01E	Product Detail	18-Jul-08	Catalog	Check	10-
000718-01E	CHU-0017	18-Jul-08	000718-01E	Product Detail	18-Jul-08	Catalog	Check	14
001008-01	END-0010	08-Oct-08	001008-01	Service	08-Oct-08	Mailings	1/10 Net 30	08
001008-01	END-0010	08-Oct-08	001008-01	Service	08-Oct-08	Mailings	1/10 Net 30	09
001008-01	END-0010	08-Oct-08	001008-01	Service	08-Oct-08	Mailings	1/10 Net 30	15
990701-01	END-0010	01-Jul-08	A2000215-2	Service	17-Sep-07		Lease	15
020115-01	CHU-0017	15-Jan-08	020115-01	Product Detail	15-Jan-10	Newspaper	Check	15
020123-01	OKI-0004	23-Jan-08	020123-01	Professional	23-Jan-10	Referral	PO Net 30	23
020801-01	SWA-0044	01-Aug-08	020801-01	Product Detail	01-Aug-10	Catalog	PO Net 30	01-
1111111	THO-0043	21-Dec-08	A2000218-(	Product Detail	21-Dec-07		Credit Card	18
99070106	FRE-0048	21-Dec-08	A2000218-(	Product Detail	21-Dec-07		Cash	18
00001	END-0010	06-Aug-08	2000105-06	Product Detail	05-Jan-08	POS	Cash	05
990616-03	MAD-0016	16-Jun-08	2000105-07	Product Detail	05-Jan-08	POS	Credit Card	05
990616-03	MAD-0016	16-Jun-08	2000105-07	Product Detail	05-Jan-08	POS	Credit Card	05
A2000111-(	END-0010	11-Jan-08	2000105-09	Product Detail	05-Jan-08	POS	Cash	05
000718-01E	CHU-0017	18-Jul-08	2000105-1(	Product Detail	05-Jan-08	POS	Cheque	05
A2000111-(	MCG-0005	11-Jan-08	2000105-11	Product Detail	05-Jan-08	POS	Online Payment	05

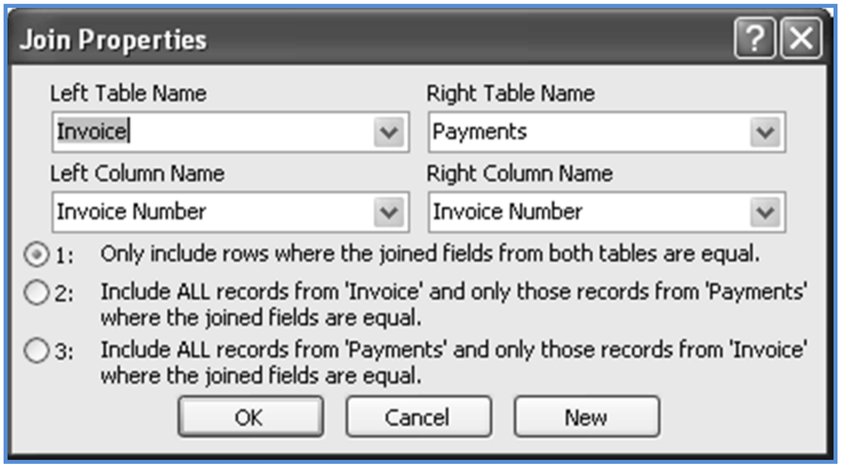
Fig. 1.3.6: Sales Order with Payments

Notice that these query results are showing only the records for which payments have been received. To get the records for which the payment are not yet received, the join between *Invoice* and *Payments* table has to be converted to left outer join.

Steps to create left outer join between *Invoice* and *Payments* tables

6. Switch to the query design view by clicking **Design View** in **Home** -> **Views**.

7. Right-click the joining line between *Invoice* and *Payments* tables and select **Join Properties** from the menu, or double-click the joining line. The **Join Properties** window appears, as shown in Fig. 1.3.7.



Left Table Name: Invoice

Right Table Name: Payments

Left Column Name: Invoice Number

Right Column Name: Invoice Number

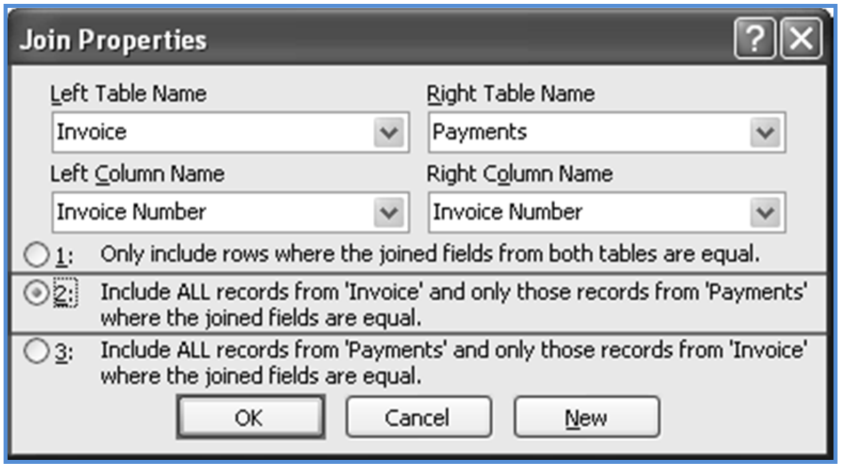
● 1: Only include rows where the joined fields from both tables are equal.

○ 2: Include ALL records from 'Invoice' and only those records from 'Payments' where the joined fields are equal.

○ 3: Include ALL records from 'Payments' and only those records from 'Invoice' where the joined fields are equal.

Fig. 1.3.7: Join Properties dialog box

8. Select the option **2: Include ALL records from 'Invoice' and only those records from 'Payments' where the joined fields are equal.**, and click **OK**. The **Join Properties** window appears as displayed in Fig. 1.3.8.



Left Table Name: Invoice

Right Table Name: Payments

Left Column Name: Invoice Number

Right Column Name: Invoice Number

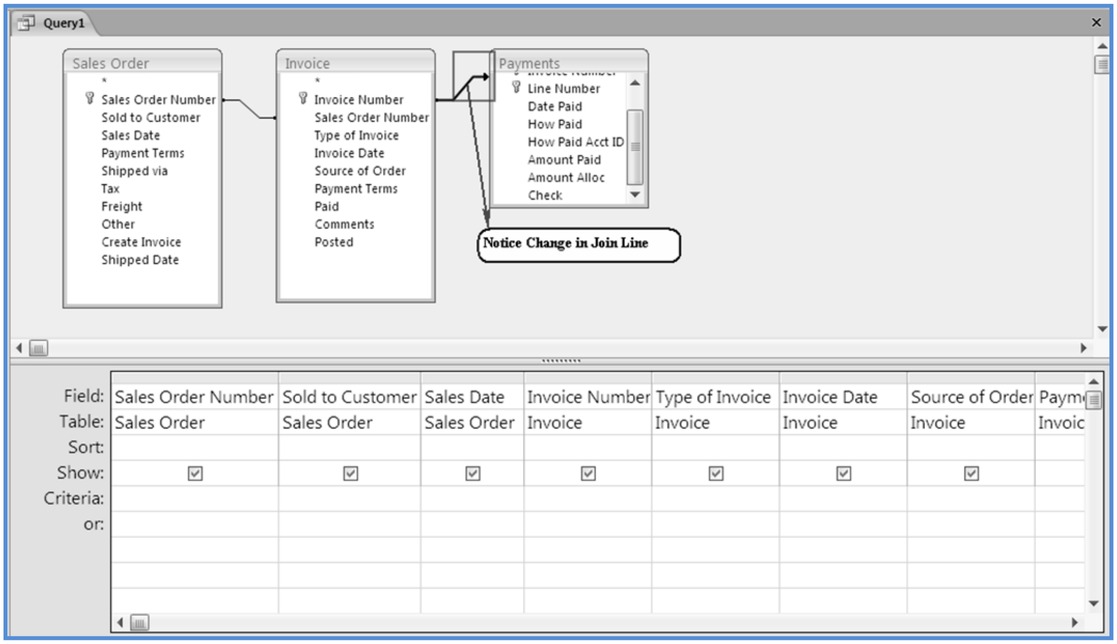
○ 1: Only include rows where the joined fields from both tables are equal.

● 2: Include ALL records from 'Invoice' and only those records from 'Payments' where the joined fields are equal.

○ 3: Include ALL records from 'Payments' and only those records from 'Invoice' where the joined fields are equal.

Fig. 1.3.8: Set the Join Properties to Left Outer Join

9. The Join line changes to indicate that it is a left outer join. Notice an arrow pointing towards the *Payments* table as indicated in Fig. 1.3.9.



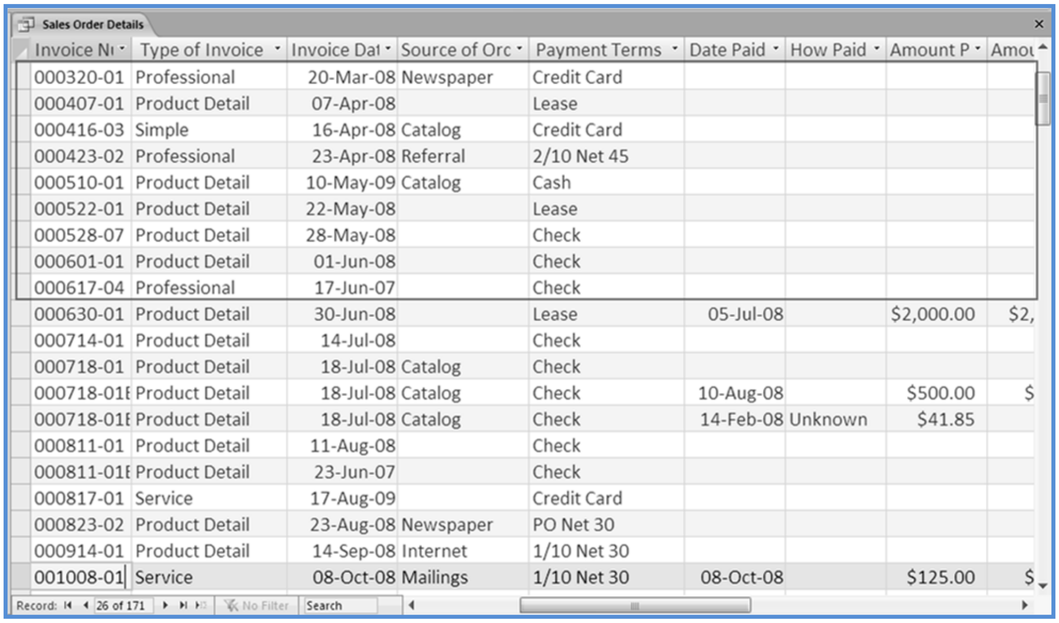
Notice Change in Join Line

Field:	Sales Order Number	Sold to Customer	Sales Date	Invoice Number	Type of Invoice	Invoice Date	Source of Order	Paym
Table:	Sales Order	Sales Order	Sales Order	Invoice	Invoice	Invoice	Invoice	Invoic
Sort:								
Show:	☑	☑	☑	☑	☑	☑	☑	
Criteria:								
or:								

Fig. 1.3.9: Left Outer Join between Invoice and Payments tables

10. Click **Save** on *Quick Access Toolbar* to save the query. Type the name of the query as *Sales Order Details*.

11. Click **Run** in **Design** -> **Results** to view the query result shown in Fig. 1.3.10. Notice that the *Date Paid*, *How Paid*, *Amount Paid* and *Amount Alloc* fields from table *Payments* are blank for few records, indicating the invoices for which payments have not been received.



Sales Order Details

Invoice Nı	Type of Invoice	Invoice Dat	Source of Orc	Payment Terms	Date Paid	How Paid	Amount P	Amoı
000320-01	Professional	20-Mar-08	Newspaper	Credit Card				
000407-01	Product Detail	07-Apr-08		Lease				
000416-03	Simple	16-Apr-08	Catalog	Credit Card				
000423-02	Professional	23-Apr-08	Referral	2/10 Net 45				
000510-01	Product Detail	10-May-09	Catalog	Cash				
000522-01	Product Detail	22-May-08		Lease				
000528-07	Product Detail	28-May-08		Check				
000601-01	Product Detail	01-Jun-08		Check				
000617-04	Professional	17-Jun-07		Check				
000630-01	Product Detail	30-Jun-08		Lease	05-Jul-08		$2,000.00	$2,
000714-01	Product Detail	14-Jul-08		Check				
000718-01	Product Detail	18-Jul-08	Catalog	Check				
000718-01E	Product Detail	18-Jul-08	Catalog	Check	10-Aug-08		$500.00	$
000718-01E	Product Detail	18-Jul-08	Catalog	Check	14-Feb-08	Unknown	$41.85	
000811-01	Product Detail	11-Aug-08		Check				
000811-01E	Product Detail	23-Jun-07		Check				
000817-01	Service	17-Aug-09		Credit Card				
000823-02	Product Detail	23-Aug-08	Newspaper	PO Net 30				
000914-01	Product Detail	14-Sep-08	Internet	1/10 Net 30				
001008-01	Service	08-Oct-08	Mailings	1/10 Net 30	08-Oct-08		$125.00	$

Fig. 1.3.10: Query Result

A more complex join can be created among any number of tables and modified accordingly to get the desired results.

1.4 Calculated Fields

The Access 2010 query's result is not restricted to the fields in tables only, but can display many other computed columns known as **Calculated Fields**. The Calculated fields can be compared to cells containing functions or formulas in Excel Worksheet. In Access, these cells can be considered as the columns of the table. Normalization forbids tables to have columns whose values can be computed using the existing fields. Calculations in a query are recomputed each time the query is run. As such, data is always current. The results of the calculations are not stored in a table. A calculated field performs some type of arithmetic calculations on one or more fields in a table to come up with a completely new field. For example, if a table has an *Order Total* field and a *Tax Rate* field, Access 2010 can calculate these two fields to find out the *Sales Tax* for each order as *[Order Total] * [Tax Rate]*.

The calculated fields create new fields in a record by combining the values of other fields in the record. Calculated fields can store numeric, date, or text fields for each record using expressions and functions.

1.4.1 Problem Scenario

Ramit, an executive in Apex Ltd. is required to produce a report displaying all the sales orders which contains the field *Total Amount Paid* as a sum of Tax, Freight and Other Charges.

Solution

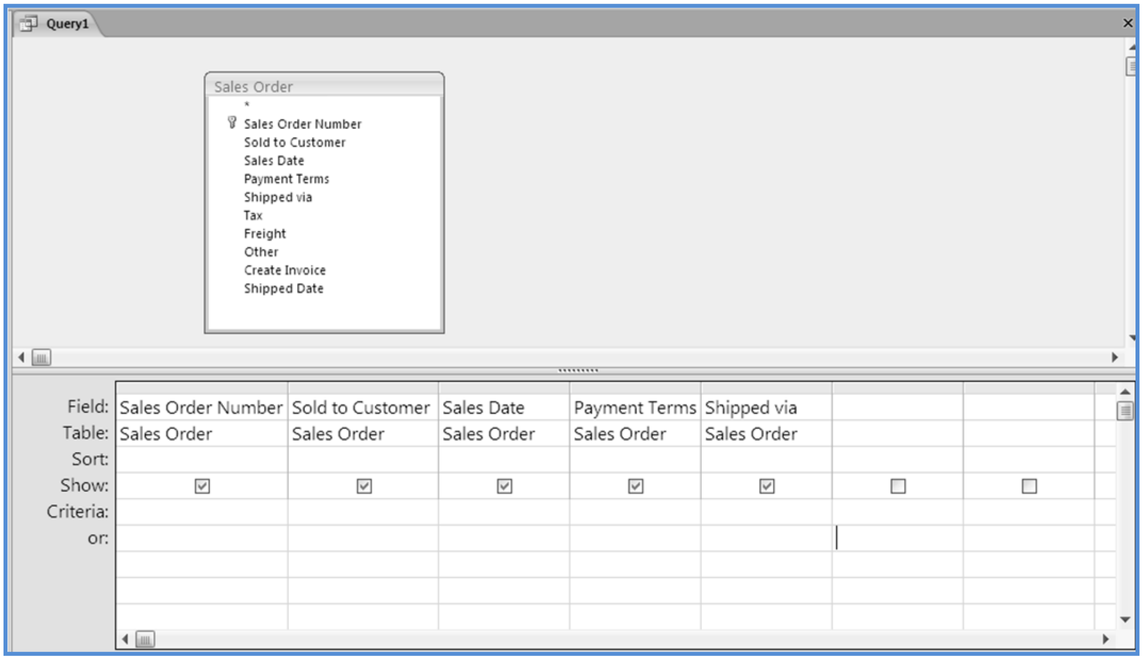
As a solution to the above problem, a query has to be created on table *Sales Order* with a calculated column *Total Amount Paid* computed as a sum of Tax, Freight and Other Charges.

Steps for creating query with Calculated Columns

1. Open the *Query Design* window, by clicking **Create** -> **Queries** -> **Query Design**.

2. Select the *Sales Order* table from **Show Table** window and click **Add**. Click **Close**.

3. Select the columns *Sales Order Number*, *Sold to Customer*, Sales Date, *Payment Terms* and *Shipped via* from *Sales Order* table and drag them to the columns tab as displayed in Fig. 1.4.1.

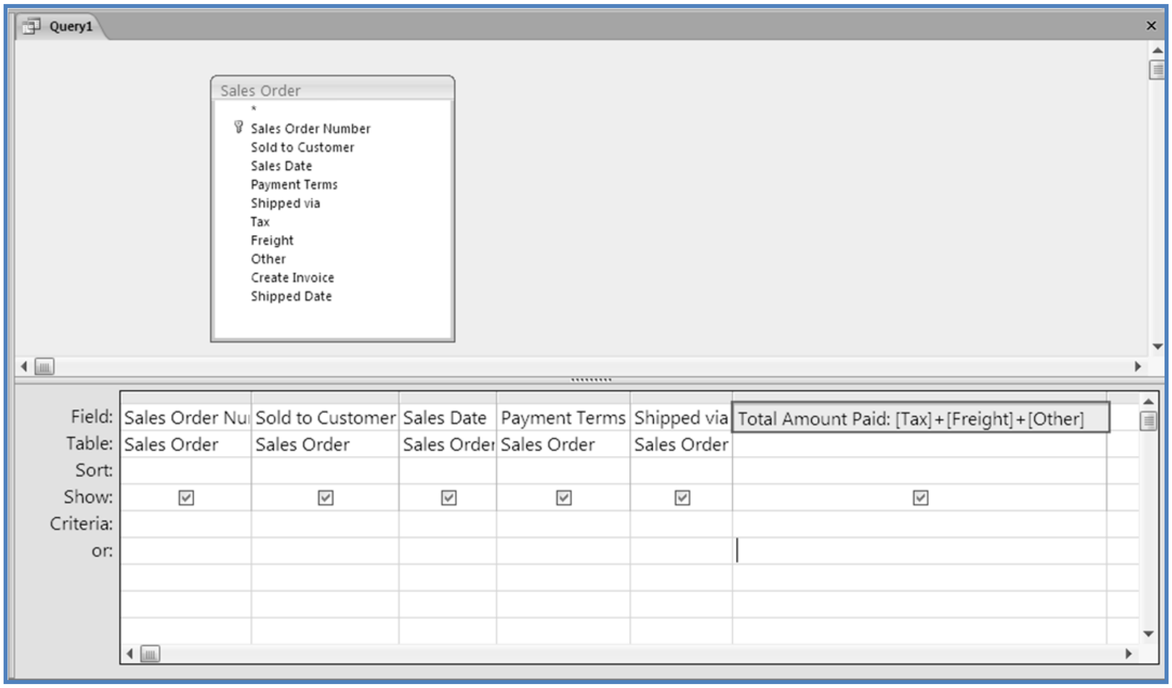


Sales Order
- *
- Sales Order Number
- Sold to Customer
- Sales Date
- Payment Terms
- Shipped via
- Tax
- Freight
- Other
- Create Invoice
- Shipped Date

Field:	Sales Order Number	Sold to Customer	Sales Date	Payment Terms	Shipped via		
Table:	Sales Order	Sales Order	Sales Order	Sales Order	Sales Order		
Sort:							
Show:	☑	☑	☑	☑	☑	□	□
Criteria:							
or:							

Fig. 1.4.1: Select Required Columns

4. To create the calculated column, write the expression *Total Amount Paid: [Tax]* + *[Freight]* + *[Other]* in the **Field** tab of the next column as displayed in Fig. 1.4.2.



Query1

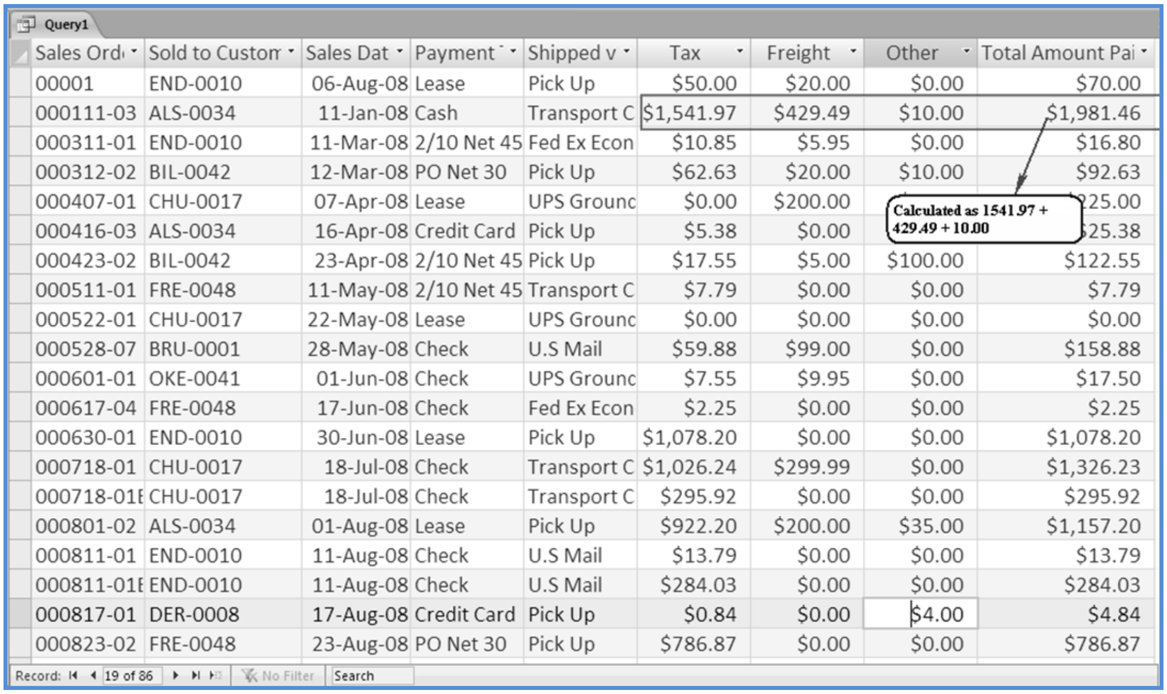
Sales Order

*
Sales Order Number
Sold to Customer
Sales Date
Payment Terms
Shipped via
Tax
Freight
Other
Create Invoice
Shipped Date

Field:	Sales Order Nu	Sold to Customer	Sales Date	Payment Terms	Shipped via	Total Amount Paid: [Tax]+[Freight]+[Other]
Table:	Sales Order	Sales Order	Sales Order	Sales Order	Sales Order	
Sort:						
Show:	☑	☑	☑	☑	☑	☑
Criteria:						
or:						

Fig. 1.4.2: Calculated Column - Total Amount Paid

5. Click **Run** ! Run in **Design** -> **Results** to view the query result. The result is displayed in Fig. 1.4.3. Note the calculated column *Total Amount Paid* in query result.



Sales Ord	Sold to Custom	Sales Dat	Payment	Shipped v	Tax	Freight	Other	Total Amount Pai
00001	END-0010	06-Aug-08	Lease	Pick Up	$50.00	$20.00	$0.00	$70.00
000111-03	ALS-0034	11-Jan-08	Cash	Transport C	$1,541.97	$429.49	$10.00	$1,981.46
000311-01	END-0010	11-Mar-08	2/10 Net 45	Fed Ex Econ	$10.85	$5.95	$0.00	$16.80
000312-02	BIL-0042	12-Mar-08	PO Net 30	Pick Up	$62.63	$20.00	$10.00	$92.63
000407-01	CHU-0017	07-Apr-08	Lease	UPS Grounc	$0.00	$200.00	[illegible]	225.00
000416-03	ALS-0034	16-Apr-08	Credit Card	Pick Up	$5.38	$0.00	[illegible]	$25.38
000423-02	BIL-0042	23-Apr-08	2/10 Net 45	Pick Up	$17.55	$5.00	$100.00	$122.55
000511-01	FRE-0048	11-May-08	2/10 Net 45	Transport C	$7.79	$0.00	$0.00	$7.79
000522-01	CHU-0017	22-May-08	Lease	UPS Grounc	$0.00	$0.00	$0.00	$0.00
000528-07	BRU-0001	28-May-08	Check	U.S Mail	$59.88	$99.00	$0.00	$158.88
000601-01	OKE-0041	01-Jun-08	Check	UPS Grounc	$7.55	$9.95	$0.00	$17.50
000617-04	FRE-0048	17-Jun-08	Check	Fed Ex Econ	$2.25	$0.00	$0.00	$2.25
000630-01	END-0010	30-Jun-08	Lease	Pick Up	$1,078.20	$0.00	$0.00	$1,078.20
000718-01	CHU-0017	18-Jul-08	Check	Transport C	$1,026.24	$299.99	$0.00	$1,326.23
000718-01E	CHU-0017	18-Jul-08	Check	Transport C	$295.92	$0.00	$0.00	$295.92
000801-02	ALS-0034	01-Aug-08	Lease	Pick Up	$922.20	$200.00	$35.00	$1,157.20
000811-01	END-0010	11-Aug-08	Check	U.S Mail	$13.79	$0.00	$0.00	$13.79
000811-01E	END-0010	11-Aug-08	Check	U.S Mail	$284.03	$0.00	$0.00	$284.03
000817-01	DER-0008	17-Aug-08	Credit Card	Pick Up	$0.84	$0.00	$4.00	$4.84
000823-02	FRE-0048	23-Aug-08	PO Net 30	Pick Up	$786.87	$0.00	$0.00	$786.87

Calculated as 1541.97 + 429.49 + 10.00

Fig. 1.4.3: Query Result

6. Save the query as *Sales Order Report* and close the query window.

1.5 Summary

Queries are the heart of every database application. Queries are responsible for converting diffuse data contained in tables into information that users can actually use. Without queries, we would have to write a complex code for every data extraction and transformation.

This chapter talks about more complex part of queries. *Sub-queries,* as the name indicates is a query within query and helps us to retrieve data from multiple tables, and can also be used to replace Joins. The *Query Wizard* can be used to create queries like *Find Unmatched* and *Find Duplicate*. Apart from this, queries can generate summary reports using *Group By* and *Crosstab Queries*. Queries can be very interactive in terms of *PivotTables* and *PivotCharts* which can be obtained by just changing the query view.

Queries can be based on multiple tables using different types of joins. Joins can be Inner Join or Outer Joins. Queries provide much more liberty, such as adding newly calculated columns to query result or restrict number of rows returned by specifying the criteria. Calculated columns enable us to implement normalization by omitting unnecessary columns in the table and introducing them as calculated columns later.

1.6 Lab Exercises

1.2.6 Case Study

For the database Apex Inventory Shipment of Apex Ltd., consider the scenario and provide their solutions.

1. The company is launching a new sales campaign for its existing customers. For this, the marketing department needs the list of customers with the highest credit limit. Create a query using sub-query to retrieve this list.
2. A sales person made a mistake and skipped entering the item description for an order made by the customer. Using query, retrieve the sales order information for which the item description has not been entered.
3. For the above exercise, retrieve the information about the customer who has placed the order so that the items can be reconfirmed.
4. The company launched its new office in India and asked the sales executive to interact with various people and enter the details of possible customers in a table named *Customers_India*. The table has the following structure:

Field Name	Data type
Customer Number	Text
Salutation	Text
First Name	Text
Last Name	Text
Company	Text
Phone	Text
Email	Text
Street	Text
City	Text
State	Text
ZIP Code	Text

   In the absence of any constraints on the table, the sales team ended up inserting duplicate records of a single customer. Write a query to find these duplicate records. (Consider the *Customer Number* as unique for each customer.)
5. The Finance Head who keeps track of their inventory shipped, requires the report displaying the quantity of items shipped every month of year 2011. Create a Totals query displaying the required data.
6. The company is issuing a discount policy to its customers. For this purpose, the Marketing Regional Manager needs a detailed report of the customers. The report should contain Customers Names on rows and Month Name as columns, and count of orders placed by each customer on the intersection cells of rows and columns.

7. Create a user interactive report using PivotTable to display payments made by customers. The user should be able to filter the customers according to their states and should be able to drill the payments in terms of years, quarters, months and days.

8. Considering the scenario given in the above exercise, create a graphical view using PivotChart indicating the total payments received quarter-wise from all the customers from a particular city. User should be able to filter the city on the basis of their country.

9. Create a query containing mailing address list for all the customers with the following field list:

1	Salutation FirstName Last Name
2	Company
3	Street, City
4	State
5	Country
6	Postal Code

10. Display the list of all customers from New York (State Code – *NY*), with details of items they have ordered and Total Amount to be paid by them.

1.2.7 Multiple Choice Questions

1. For the database Apex Inventory Shipment of the Apex Ltd., we wish to retrieve records for customers who have not placed any orders yet. What type of query can be used?

   (a) Duplicate Query

   (b) Crosstab Query

   (c) Find Unmatched Query

   (d) Group By query

2. Considering the database Apex Inventory Shipment of the Apex Ltd., how can we display sales grouped by country, state, and customer, all at the same time?

   (a) Use the Sort Descending command

   (b) Use the PivotTable View command

   (c) Use the Find command on specified groups

   (d) All of the above

3. For the database Apex Inventory Shipment, the number of items for each category is required to be computed, which query type is required to be used?

   (a) Select Query

   (b) Group By query

   (c) Crosstab Query

   (d) Duplicate Query

4. Considering the scenario in question above, which function should be used with *Item Number* in query?

   (a) Count

   (b) Compute

   (c) Sum

   (d) Calculate

5. In the database Apex Inventory Shipment, to see total amount received from *Payments* table, it should be dragged into which area of the PivotTable?

   (a) Drop Column Fields Here

   (b) Drop Row Fields Here

   (c) Drop Totals or Detail Fields Here

   (d) None of the above

6. Considering the database Apex Inventory Shipment, we wish to get all the sales orders with cash payments. What needs to be done to achieve this?

   (a) Create a sub-query with *Sales Order* as Payment type

   (b) Create a select query on *Sales Order* table and write *Cash* as criteria for *Payment Terms*

   (c) Create a Group-By query on *Payment Terms*

   (d) None of the above

7. From the database Apex Inventory Shipment, we wish to retrieve a report displaying details of all the invoices including the payments details (if already made) of the invoices. What type of Join should be used in Query window to achieve the desired result?

   (a) Left outer join

   (b) Right outer join

   (c) Default join

   (d) No joins will be used

8. How can we add a table to the *Query Design* window?

   (a) Select **Create** -> **Add Table**

   (b) Select **Database Tools** -> **Add Table**

   (c) Select **Design** -> **Show Table**

   (d) Select the table from the **Navigation Pane**

9. If we are creating a Crosstab query, the table we are querying must contain what?

   (a) Lots of confusing information

   (b) More than 100 records

(c) At least one field

(d) At least three fields

10. ________ type of query summarises information in a grid, organized by regions and months.

(a) An update query

(b) A parameter query

(c) An action query

(d) A Crosstab query

11. When we double click a query object, we open

(a) The object in design view

(b) The object in print preview

(c) The result of the query

(d) The underlying table on which the query is based

12. What is the primary difference between a PivotTable report and a Crosstab query?

(a) A PivotTable report can contain sums, counts, and averages, while a Crosstab query cannot

(b) We cannot create a PivotTable from a Crosstab query

(c) A Crosstab query lets us group similar items, while a PivotTable query does not

(d) None of the above

13. In Access 2010, the best types of queries to use for data analysis are:

(a) Select queries

(b) Parameter queries

(c) Action queries

(d) All of the above

14. Which view allows adding tables to the query?

(a) Datasheet view

(b) PivotTable view

(c) PivotChart view

(d) Design view

15. Which type of join in multi-table query permits to view all the records from one table and matching from another?

(a) Inner Join

(b) Outer Join

(c) Equi Join

(d) Non-Equi Join

16. For the database Apex Inventory Shipment, we need to present a graphical view of the sales volume of products. User should have the liberty to view all the products, or products from a particular category. For this purpose, a PivotChart is created. Which field should be placed in area **Drop Series Fields Here**?

    (a) Category

    (b) Total Sales

    (c) Month

    (d) Item Number

17. For the above question, the area **Drop Filter Fields Here** should contain _________ field.

    (a) Category

    (b) Item Number

    (c) Years

    (d) Month

18. To view the results in Datasheet view of the query created, do the following _____________.

    (a) Press F5 key

    (b) Click **Run** on **Design** tab

    (c) Press CTRL + R

    (d) All of the above

19. The Σ Totals option in **Design** ribbon permits us to create what type of queries?

    (a) Crosstab Queries

    (b) Action Queries

    (c) Parameter Queries

    (d) Group By Queries

20. For the database Apex Inventory Shipment, if we wish to delete all the invoices for which the payments were made in last quarter of year 2011, what should be done?

    (a) Create a select query and delete records manually

    (b) Create a sub-query with action query

    (c) Create simple action query

    (d) Cannot be done using queries

CHAPTER 2

DESIGNING FORMS AND REPORTS

LEARNING OBJECTIVES

- ■ Advanced Form Design
  - ❑ Adding Unbound Controls
  - ❑ Adding Graphics to Form
  - ❑ Adding Calculated Values
  - ❑ Adding Combo Boxes
- ■ Make effective use of forms
  - ❑ Displaying a calendar control on a form
  - ❑ Organising information with tab pages
  - ❑ Displaying a summary of data in a form
- ■ Advanced Reports
  - ❑ Creating customised headers and footers
  - ❑ Adding calculated values
  - ❑ Sub-reports
- ■ Make Reports more effective
  - ❑ Including a chart in a report
  - ❑ Printing data in columns
  - ❑ Cancelling the printing of a blank report

2.1 Introduction

Forms help to display, add, modify, and delete data. Different features available in Access 2010 enable to create forms such that it becomes easier for the users to handle data. Access 2010 provides various methods to make forms handier and simpler. Different controls can be added to a form to enhance their working. Access also provides ActiveX controls which are Microsoft control meant for different purposes. A form can be divided into pages to increase the readability, or can include a summary section to display grouped data.

Reports can be considered as the static version of forms. Reports are the best way to present data to higher authority and communicate the information to the people. They can be customized using header and footers, calculated values, and sub-reports to represent linked data. Access 2010 also provides the facility to create charts in reports which represents data diagrammatically. Since reports are used for business communication, it must be available for everyone.

2.2 Advanced Form Design

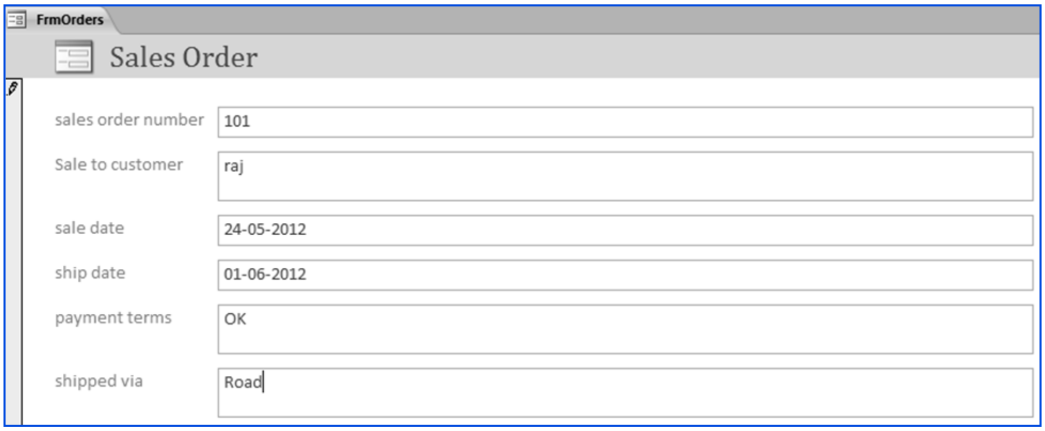
Forms are an important medium of representing data in Access 2010. Forms are not only used for entering of data, but also to view the data in a user friendly manner. Access 2010 provides various controls and utility to make the forms more presentable. For example, forms can contain unbounded controls, graphics, calculated values, and combo boxes.

2.2.1 Adding Unbound Controls

An Access 2010 Form can contain multiple controls, each having its own significance. These controls can be selected from *Create* tab in *Forms* ribbon. In general, all the controls are bounded to one field of the table or query the form is based on. However, there are few controls which retain the entered value, but are not linked with any table fields. These controls can be used for text label display, for controls such as lines and rectangles, or for holding unbound OLE objects (such as bitmap pictures or logo) that are not stored in a table, but in the form itself. Unbound controls are also known as variables or memory variables.

2.2.2 Problem Scenario

The database Apex Inventory Shipment has a form *frmOrders* as shown in Fig. 2.2.1, based on table *Sales Order* which keeps track of all the orders placed. The Sales Manager wishes to add a current date on the form, so that it becomes easy for the user to keep track of dates.



Sales Order

sales order number	101
Sale to customer	raj
sale date	24-05-2012
ship date	01-06-2012
payment terms	OK
shipped via	Road

Fig 2.2.1. frmOrders form

Solution

The Database Developer adds an unbounded control, a label to the form and sets its property to store the current date.

Steps for adding an unbounded control

1. Right-click the form *frmOrders* under *All Access Objects* -> *Forms* tab. Select *Design View* from the dropdown to open the table in *Design View,* as shown in Fig. 2.2.2.

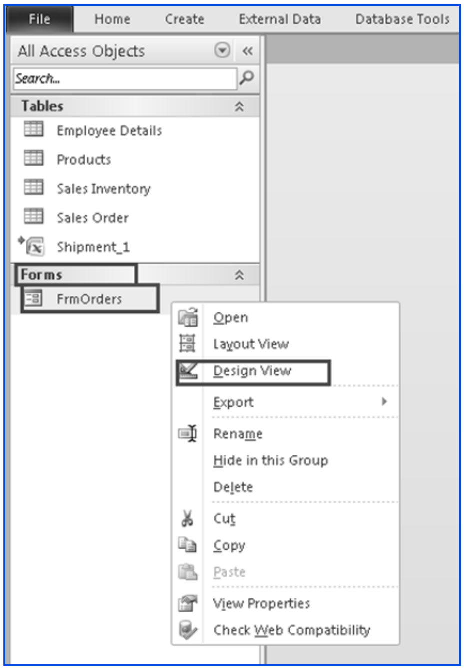
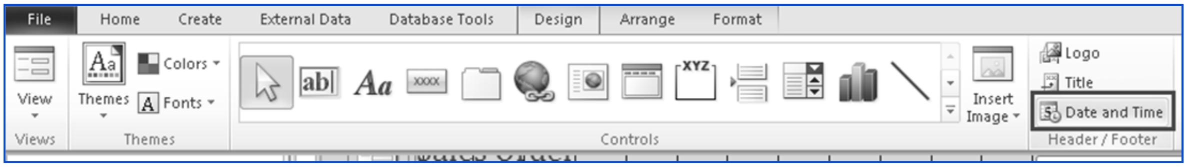
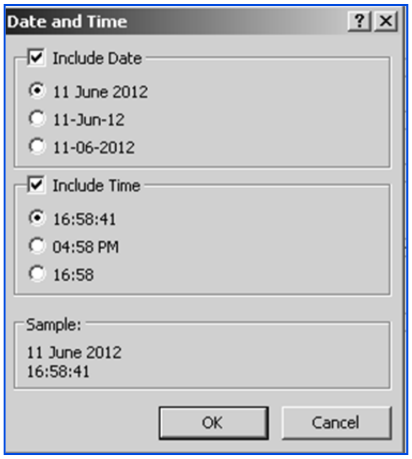


Fig 2.2.2. Open the form frmOrders in Design view

2. Select the *Date & Time* control from *Design ribbon ->Header/Footer* tab. The *Date and Time* window appears, as shown in Fig. 2.2.3 select the date & time format from this window.





Date and Time

☑ Include Date

● 11 June 2012
○ 11-Jun-12
○ 11-06-2012

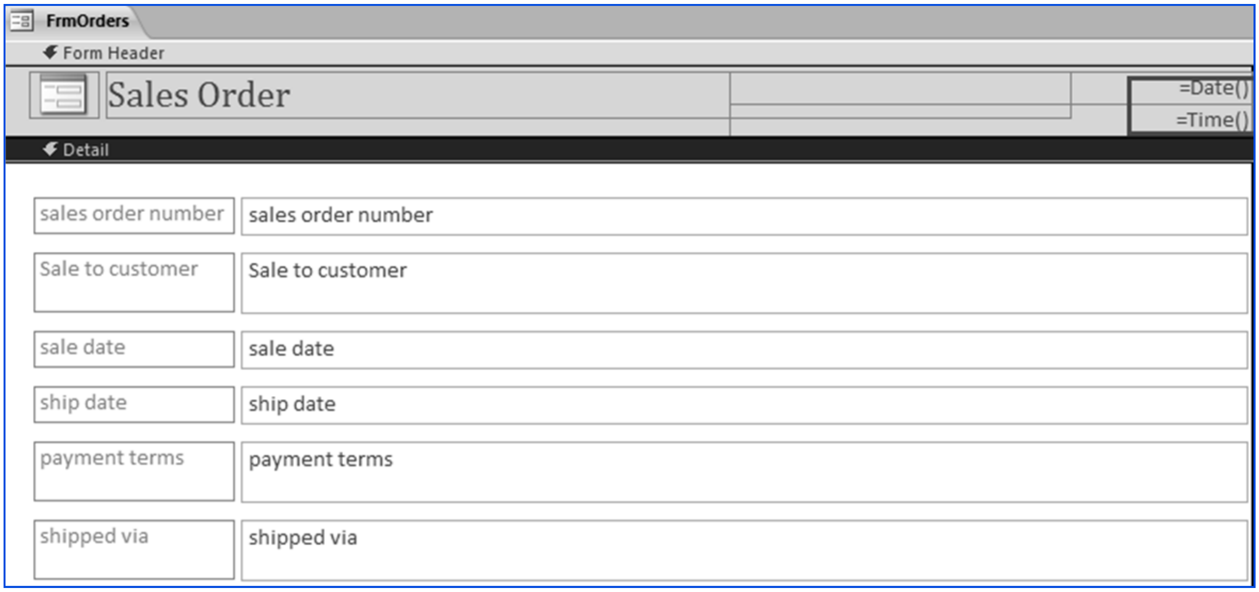
☑ Include Time

● 16:58:41
○ 04:58 PM
○ 16:58

Sample:
11 June 2012
16:58:41

Fig. 2.2.3: Date and Time window

3. Click *OK* to close the window and draw the control on the *Form Header* portion on the form. The form should appear, as shown in Fig. 2.2.4.



FrmOrders

Form Header

Sales Order

=Date()
=Time()

Detail

sales order number	sales order number
Sale to customer	Sale to customer
sale date	sale date
ship date	ship date
payment terms	payment terms
shipped via	shipped via

Fig. 2.2.4: Date and Time unbounded control added to a form

NOTE: In a same way, other controls like shapes, logo, etc. can be added to form. The properties of these controls can be set to meet user requirements.

2.2.3 Adding Graphics to Form

Attractive forms are always a valuable addition. Access 2010 makes it easy to add a graphic to the background of a form, such as a “watermark” which appear on expensive bond paper. The picture can contain a company logo, text, or any other graphic element. The picture is specified by the form’s picture property and can be embedded in the form or linked to an external file. If the picture is linked, the graphic displayed on the form changes any time the external file is edited.

The graphic can be inserted into the form in the following ways:

1. Import a graphic file (clip art, gif, jpeg, bmp, etc.) directly into the form.
2. Attach a link to the graphic that opens a website or other database object.
3. Associate the graphic with an existing macro in the database.

2.2.4 Problem Scenario

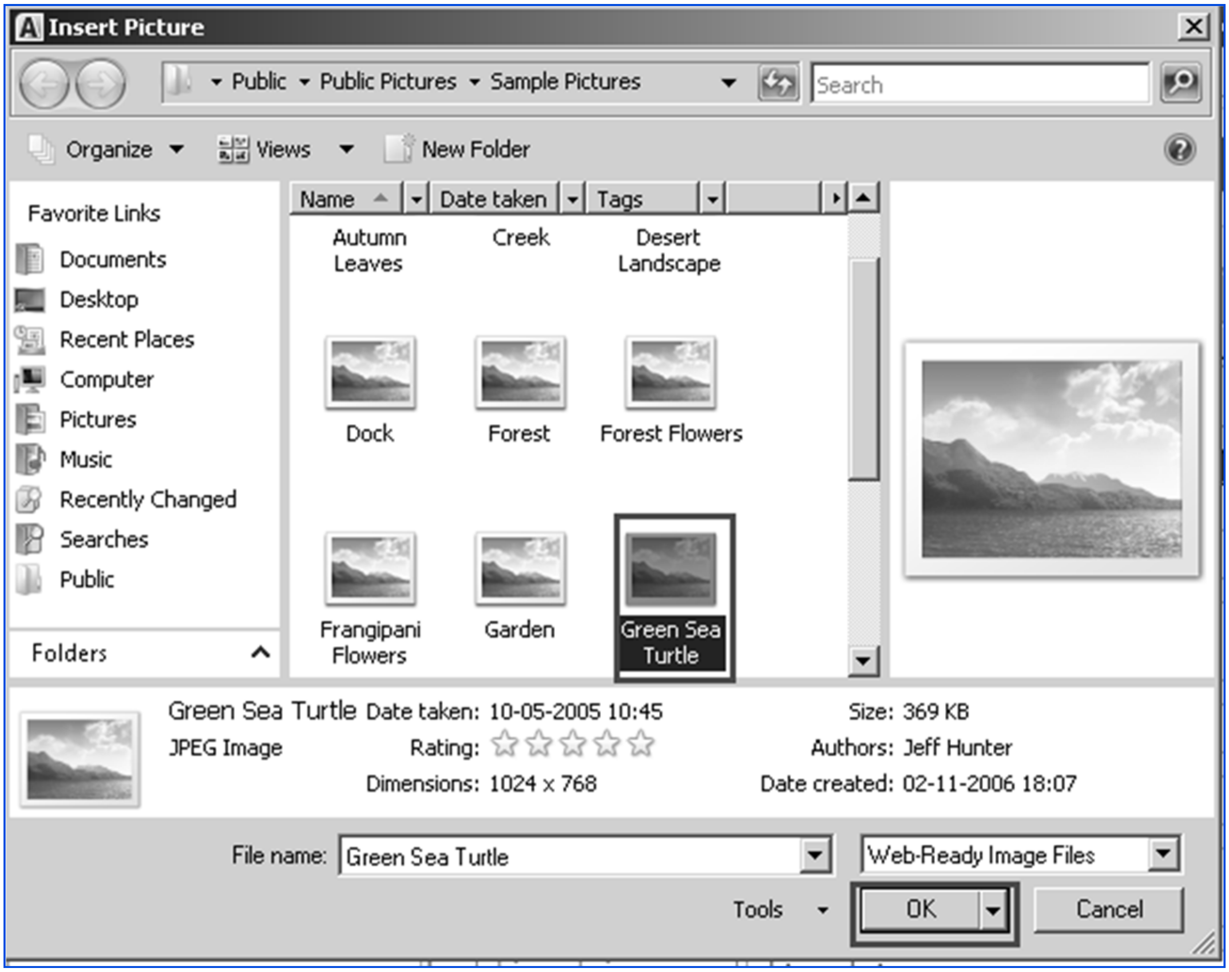
Consider the *frmOrders* discussed in section 2.2.1. The Database Developer wants to make it more presentable and thought of adding a picture to the background of the form.

Solution

The picture can be set as form background using the image control. Select the picture to be added, draw the image control and adjust its properties to make image as a background.

Steps for adding graphics

1. Right-click the form *frmOrders* under *All Access Objects* -> *Forms* tab. Select *Design View* from the dropdown to open the table in *Design View*.
2. Select the insert image button from *Design ribbon* -> *Controls* tab and drag it to cover the *Details* section of the form.
3. The *Insert Picture* window appears. Browse for the picture to be inserted and click *OK* to close the window. The *Insert Picture* window is displayed in Fig. 2.2.5.



Green Sea Turtle Date taken: 10-05-2005 10:45 Size: 369 KB

JPEG Image Rating: ☆☆☆☆☆ Authors: Jeff Hunter

Dimensions: 1024 x 768 Date created: 02-11-2006 18:07

File name: Green Sea Turtle

Web-Ready Image Files

Fig. 2.2.5. Insert Picture window

4. The form after the picture is inserted appears, as shown in Fig. 2.2.6.

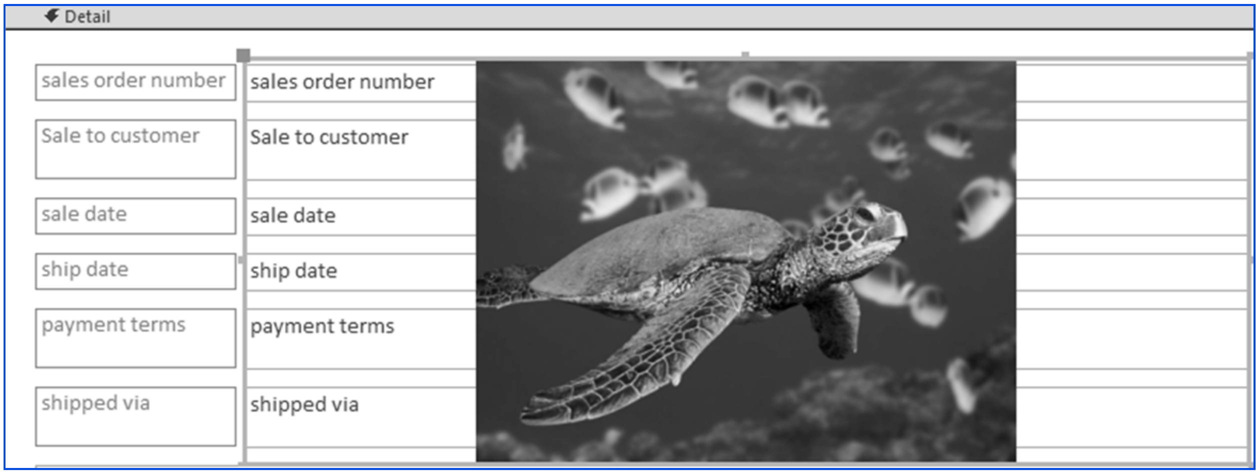


Fig. 2.2.6: Image inserted in form frmOrders

5. The picture is required to move to the back of controls. Right-click the picture and select *Position -> Send to Back* from the dropdown, as indicated in Fig. 2.2.7.

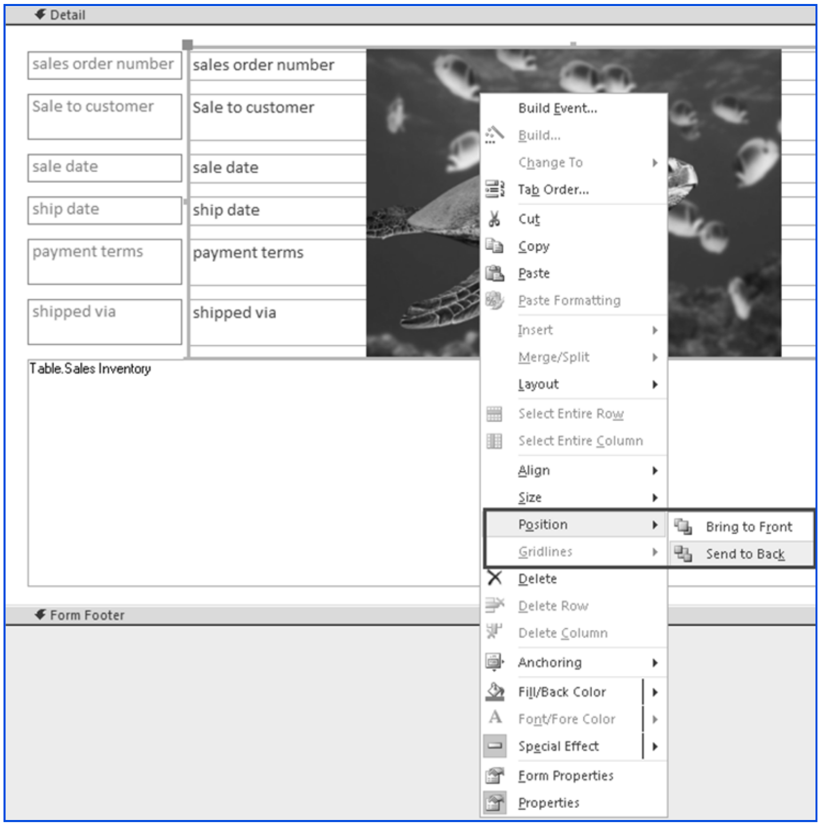


Fig. 2.2.7. Send the picture to back of controls

6. The form appears, as shown in Fig. 2.2.8.

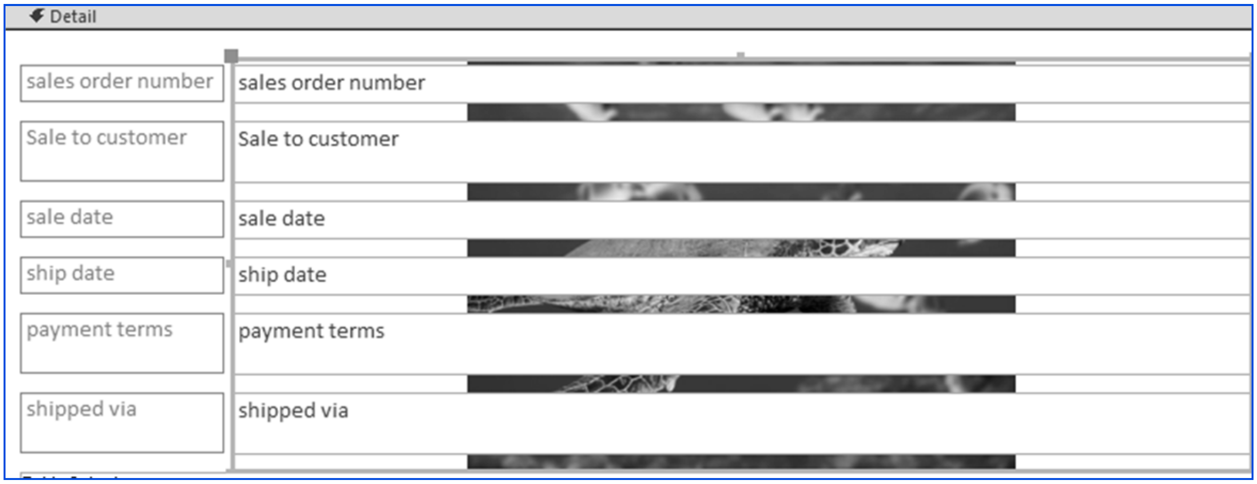
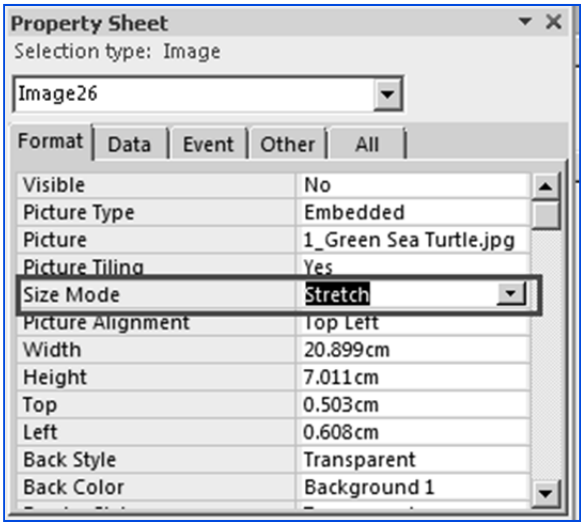


Fig. 2.2.8. Form with Graphics

7. Select the picture and press *F4* to view its properties. In the property sheet, move to format and set the *Size Mode* property of picture to stretch, as shown in Fig 3.2.9.



Selection type: Image

Image26

Visible	No
Picture Type	Embedded
Picture	1_Green Sea Turtle.jpg
Picture Tiling	Yes
Size Mode	Stretch
Picture Alignment	Top Left
Width	20.899cm
Height	7.011cm
Top	0.503cm
Left	0.608cm
Back Style	Transparent
Back Color	Background 1

Fig. 2.2.9. Set the image property

8. The final form appears, as in Fig. 2.2.10.

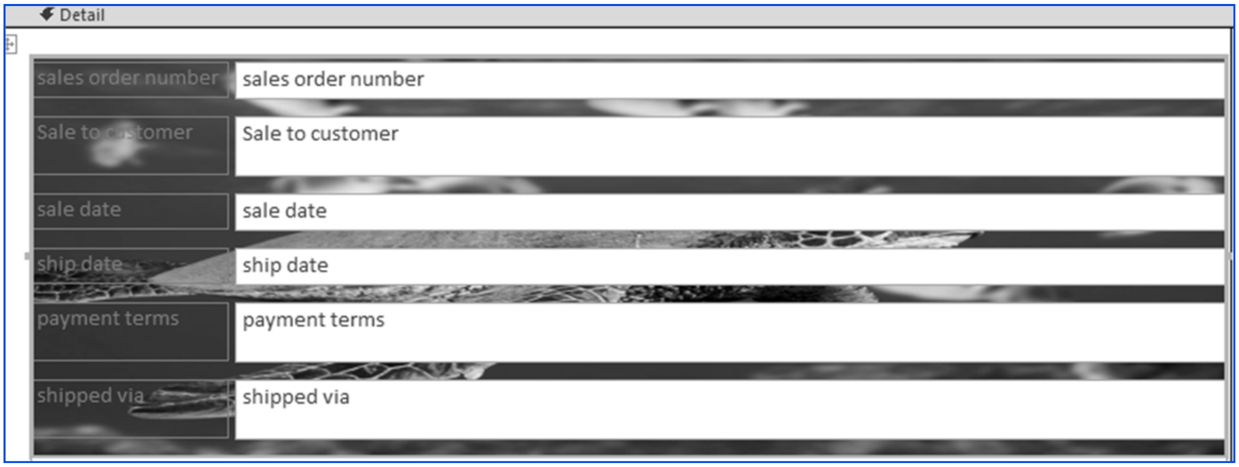


Fig. 2.2.10. frmOrders with Graphics inserted

NOTE: While adding a graphic to the form, a link to that graphic can also be created. The link ensures that the changes made to original file are also reflected in form.

2.2.5 Adding Calculated Values

Access 2010 provides many features to give forms enhanced behavior and a modern look. One of the major requirements in Access 2010form that it can be display computed results. These results may be bounded to some field in the table or may be unbounded. For example, we may need to display the complete name of the customer instead of displaying the first name and last name separately.

Calculated controls can use any of the existing function or user build function available in Access 2010. Calculated values can also be expressions computed with a combination of other fields and operators.

2.2.6 Problem Scenario

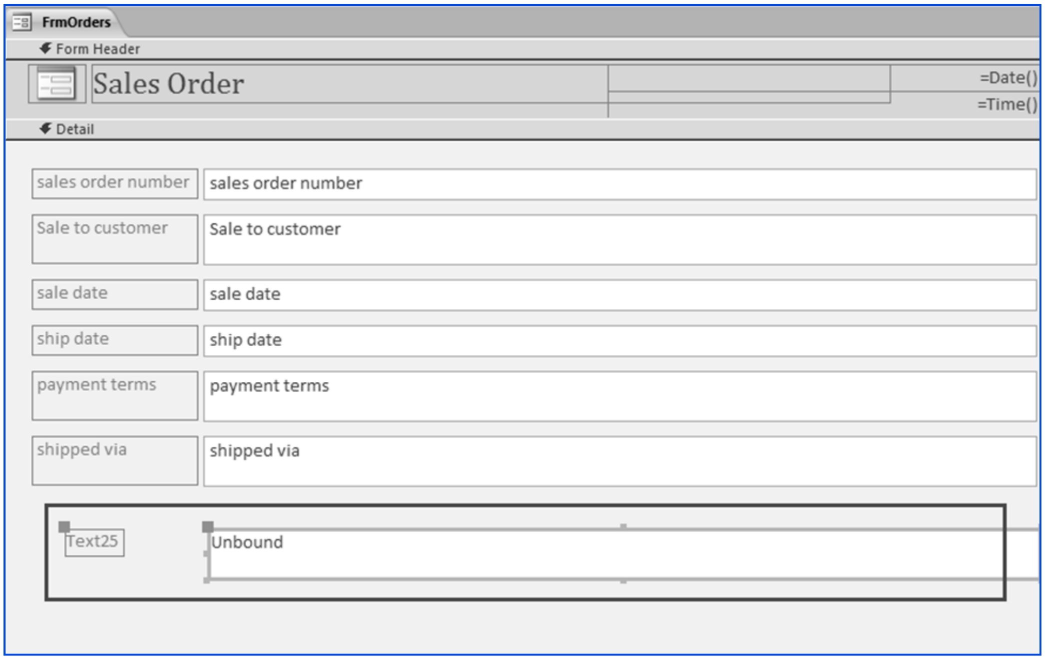
Consider the form *frmOrders* discussed in section 3.2.1. The Manager wishes to add another date in form, which may indicate when the order will reach customers. This date can be computed as 15 days after the date of shipment.

Solution

A textbox is added to the form.This textbox contains calculated value, which is computed as 15 days + date of shipment.

Steps to add calculated values to form

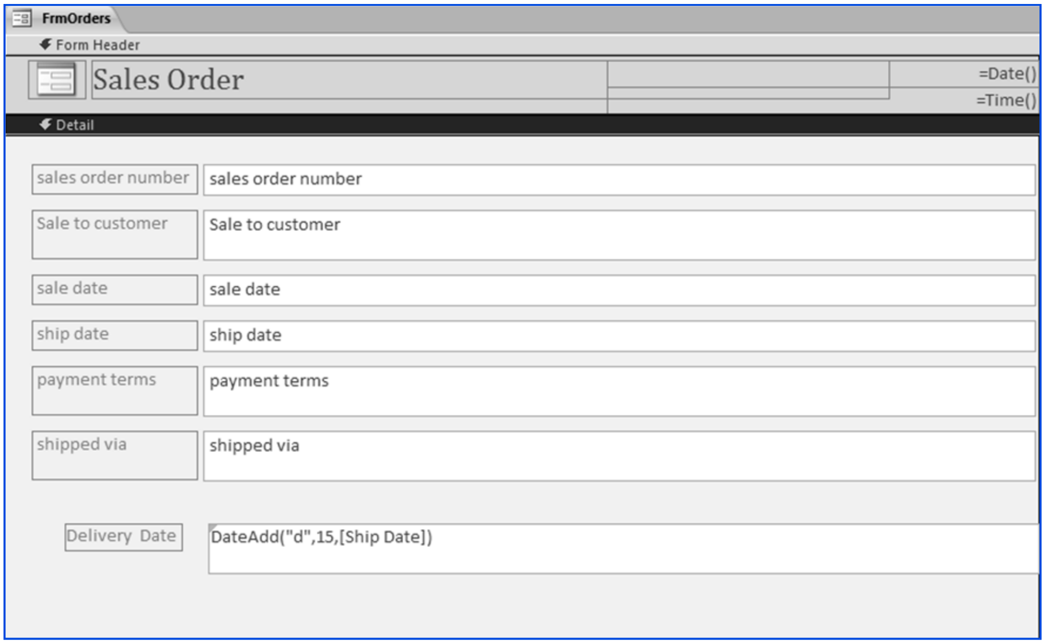
1. Right-click the form *frmOrders* under *All Access Objects -> Forms* tab. Select *Design View* from the dropdown to open the table in *Design View*.
2. Select the textbox control ab| from *Design ribbon* -> *Controls* tab and draw it on the details section of the form. The form should appear, as in Fig. 2.2.11.



sales order number	sales order number
Sale to customer	Sale to customer
sale date	sale date
ship date	ship date
payment terms	payment terms
shipped via	shipped via
Text25	Unbound

Fig. 2.2.11. Insert Textbox in Form

3. Double-click the label and type *Delivery Date* instead of *Text25*. Similarly, double-click the textbox and type *=DateAdd("d",15,[Ship Date])*. The form should appear, as shown in Fig. 2.2.12.



sales order number	sales order number
Sale to customer	Sale to customer
sale date	sale date
ship date	ship date
payment terms	payment terms
shipped via	shipped via
Delivery Date	DateAdd("d",15,[Ship Date])

Fig. 2.2.12. Specify values for Textbox control

NOTE: The formatting of the new added textbox can be copied from above cells.

4. Select *Form View* from *Home ribbon -> Views* tab to open the form in *Form View*. Confirm the textbox displays the calculated value.

2.2.7 Adding Combo Boxes

Combo boxes in Access 2010 provide a way of selecting a value from a list. This is quicker than remembering which value to type and ensures that the entered value is valid. A combo box is a compact method of presenting a list of choices and allows user to enter a value that is not in the list.

The values in a combo box are displayed by clicking the arrow at the end. In combo box values can be selected by clicking it or by typing the first few characters of the value into the text box area of the combo box. If the *Auto Expand* property is set to *Yes*, the default setting, Access 2010automatically fills in the rest of the value. Combo box consists of rows of data with one or more columns, which can appear with or without headings. One of the columns contains the values to be stored in the field (bound control) or use for other purposes (unbound control); the other columns contain explanatory information.

2.2.8 Problem Scenario

Consider the form *frmOrders* created in section 3.2.1.Theexecutives are making many mistakes while typing the reference of the customer to which orders are sold. The Sales Manager asked the Database Developer to find the solution to this problem.

Solution

The Database Developer thought that it will be a good way to add a combo box for the customers. Users will have an option to choose the value from the list or type a new value if required, thereby decreasing the typing mistakes.

Steps to add a combo box in Form

1. Right-click the form and select *Design View* from the dropdown to open the form in *Design View*. The form appears, as shown in Fig. 2.2.13.

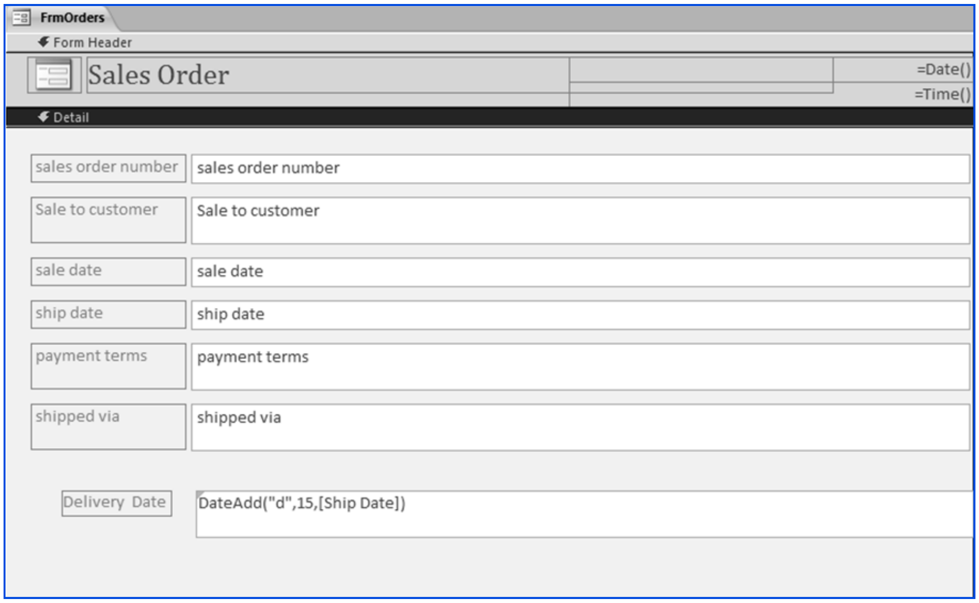
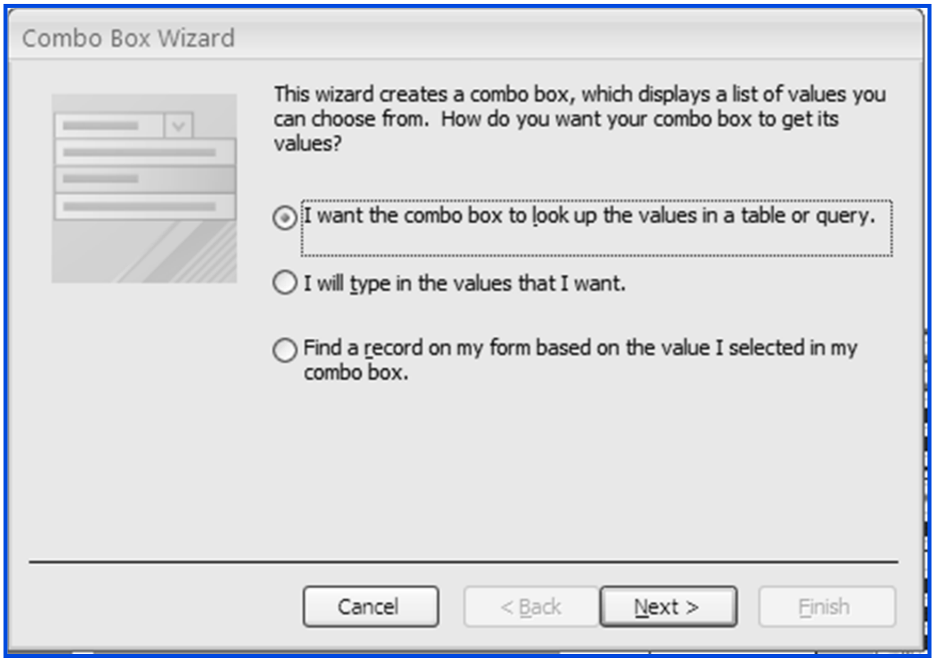


Fig. 2.2.13: frmOrders in Design View

2. Delete the *Sale to Customer* textbox.

3. Select the Combo Box control from *Design ribbon* -> *Controls* tab and draw it in the place of *Sale to Customer* textbox. The *Combo Box wizard* appears, as shown in Fig. 2.2.14.


Combo Box Wizard

This wizard creates a combo box, which displays a list of values you can choose from. How do you want your combo box to get its values?

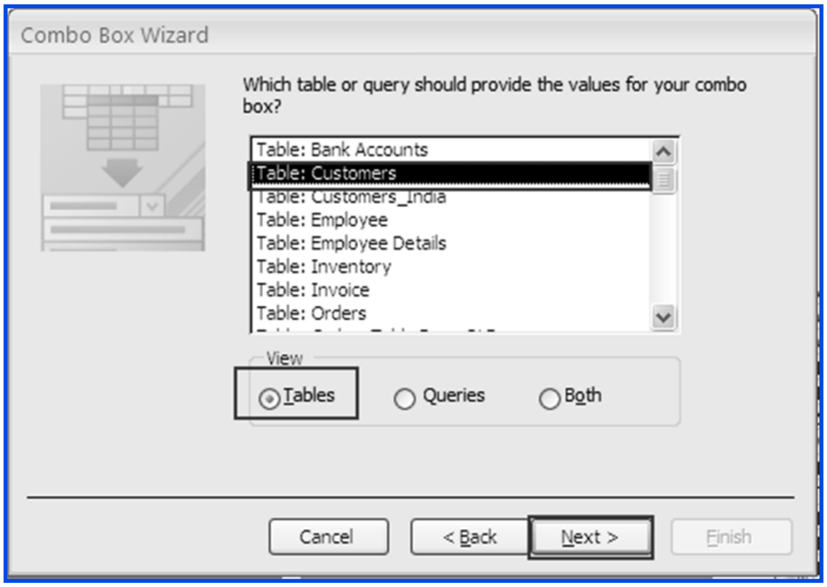
● I want the combo box to look up the values in a table or query.

○ I will type in the values that I want.

○ Find a record on my form based on the value I selected in my combo box.

Fig. 2.2.14. Combo Box wizard

4. Select the first option i.e. *I want the combo box to look up the values in a table or query* and click *Next*.
5. In the *Select Table or Query* window, select the table option from *View tab* and choose table *Customer* from the list, as indicated in Fig. 2.2.15. Click *Next* to continue.



Combo Box Wizard

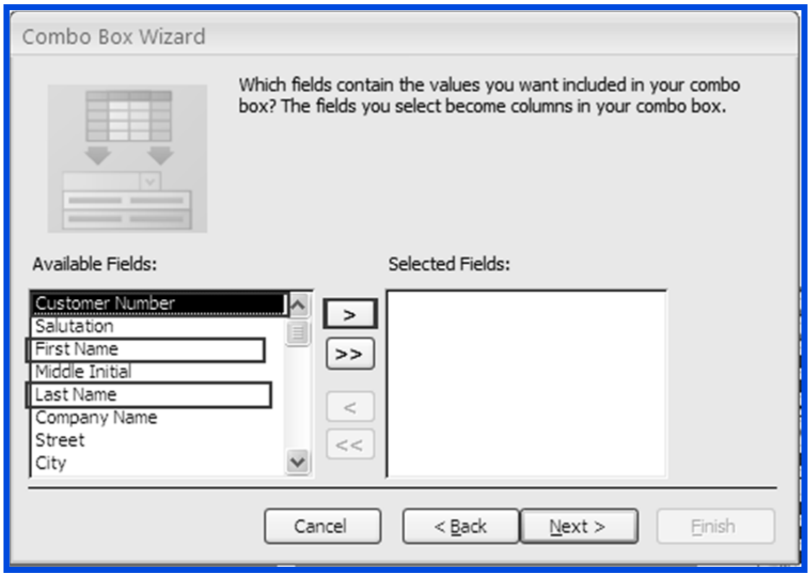
Which table or query should provide the values for your combo box?

- Table: Bank Accounts
- Table: Customers
- Table: Customers_India
- Table: Employee
- Table: Employee Details
- Table: Inventory
- Table: Invoice
- Table: Orders

View: ● Tables ○ Queries ○ Both

Fig. 2.2.15. Select the required table

6. In the *Select Field* window, select field *Customer Number* from *Available* field and click on > sign to put it into selected fields. Similarly, add *First Name* and *Last Name* to selected field. The window appears, as shown in Fig. 2.2.16. Alternatively, the user can double-click the fields to add them to selected field.



Combo Box Wizard

Which fields contain the values you want included in your combo box? The fields you select become columns in your combo box.

Available Fields:

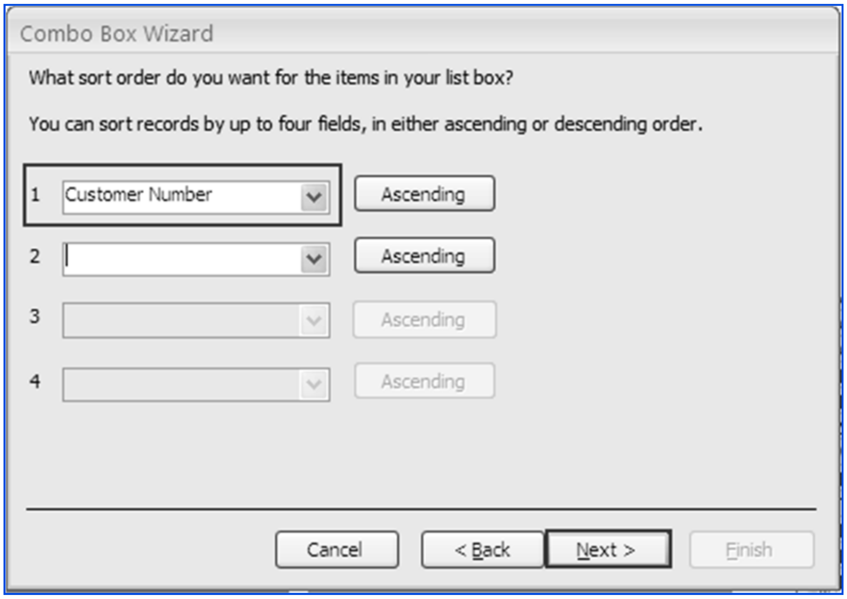
- Customer Number
- Salutation
- First Name
- Middle Initial
- Last Name
- Company Name
- Street
- City

Selected Fields:

Fig. 2.2.16. Add field to Combo Box

NOTE: Even if the multiple fields are selected to be displayed in Combo Box, the Primary Key will be stored in table or form.

7. Click *Next* to continue. In the *Sort Order* window select *Customer Number* as *1*. This is indicated in Fig. 2.2.17.



Combo Box Wizard

What sort order do you want for the items in your list box?

You can sort records by up to four fields, in either ascending or descending order.

1 Customer Number — Ascending

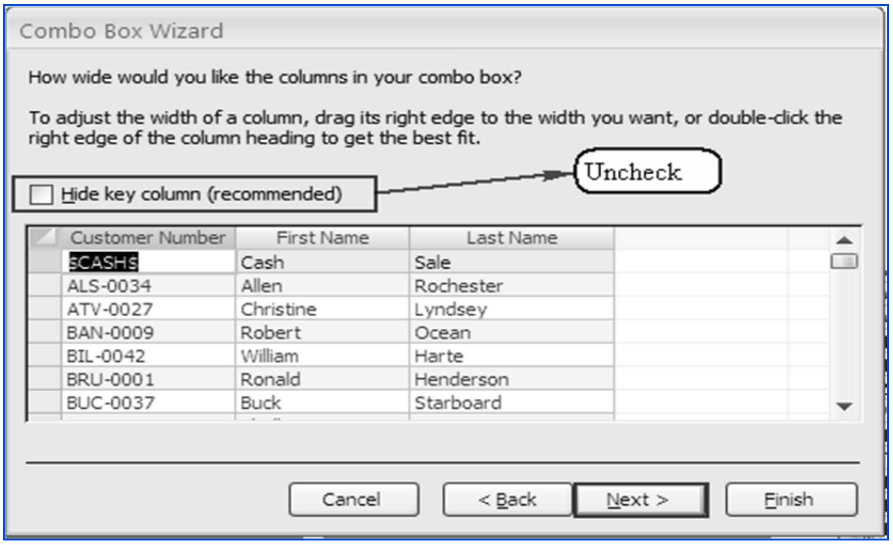
2 — Ascending

3 — Ascending

4 — Ascending

Fig. 2.2.17. Select Customer Number for Sort Order

8. In the *Hide Key Column* window, uncheck the *Hide Key Column* checkbox, as shown in Fig. 2.2.18. Click *Next* to proceed.



Combo Box Wizard

How wide would you like the columns in your combo box?

To adjust the width of a column, drag its right edge to the width you want, or double-click the right edge of the column heading to get the best fit.

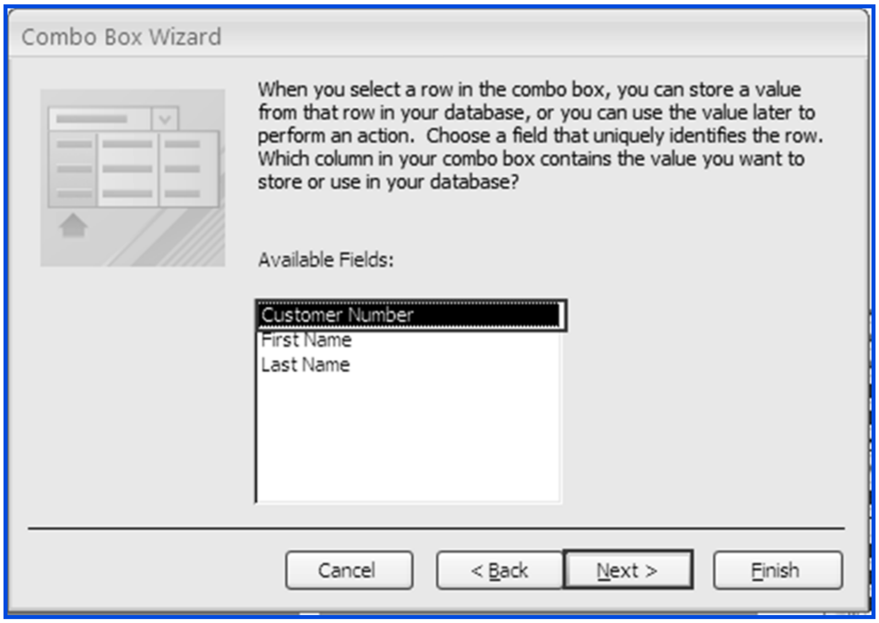
□ Hide key column (recommended)

Customer Number	First Name	Last Name
$CASH$	Cash	Sale
ALS-0034	Allen	Rochester
ATV-0027	Christine	Lyndsey
BAN-0009	Robert	Ocean
BIL-0042	William	Harte
BRU-0001	Ronald	Henderson
BUC-0037	Buck	Starboard

Fig. 2.2.18. Uncheck Hide Key Column Option

NOTE: *Hide Key Column* hides the Primary Key to be displayed as a value in combo box, though internally only Primary key is stored.

9. The next window confirms a unique value to be stored in table. Select *Customer Number* and click *Next*, as indicated in Fig. 2.2.19.



Combo Box Wizard

When you select a row in the combo box, you can store a value from that row in your database, or you can use the value later to perform an action. Choose a field that uniquely identifies the row. Which column in your combo box contains the value you want to store or use in your database?

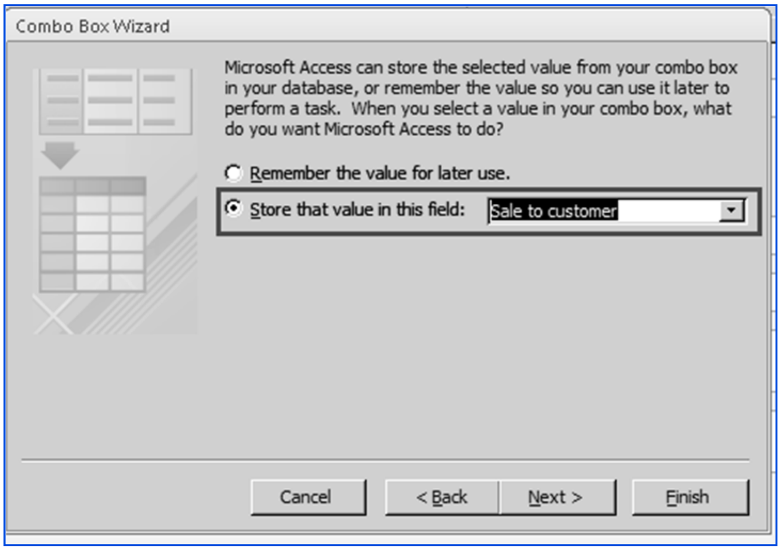
Available Fields:

Customer Number
First Name
Last Name

Fig. 2.2.19. Select Field to be stored in Table

10. In the next window that appears, select the option *Store the value in this field* and select the value *Sale to* Customer from the dropdown, as indicated in Fig. 2.2.20.

NOTE: The first option *Remember the value for later use* is used in case of unbound controls.



Combo Box Wizard

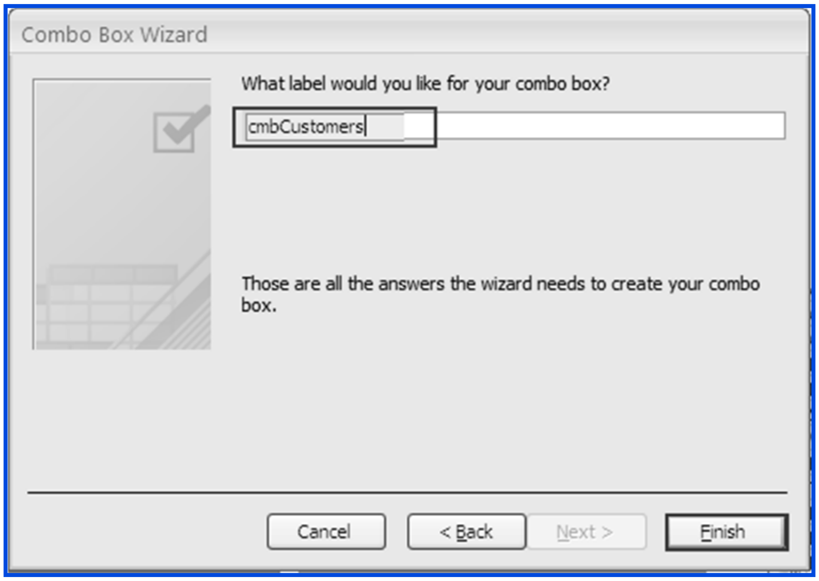
Microsoft Access can store the selected value from your combo box in your database, or remember the value so you can use it later to perform a task. When you select a value in your combo box, what do you want Microsoft Access to do?

○ Remember the value for later use.

● Store that value in this field: Sale to customer

Fig. 2.2.20. Select field to store the value

11. Click *Next* to make the *Name* window appear. Type the name *cmbCustomers* for the Combo Box and click *Finish* to close the wizard, as indicated in Fig. 2.2.21.



Combo Box Wizard

What label would you like for your combo box?

cmbCustomers

Those are all the answers the wizard needs to create your combo box.

Fig. 2.2.21: Type Name for Combo Box

12. The *Sale to* Customer combo box appears on the form, as displayed in Fig. 2.2.22.

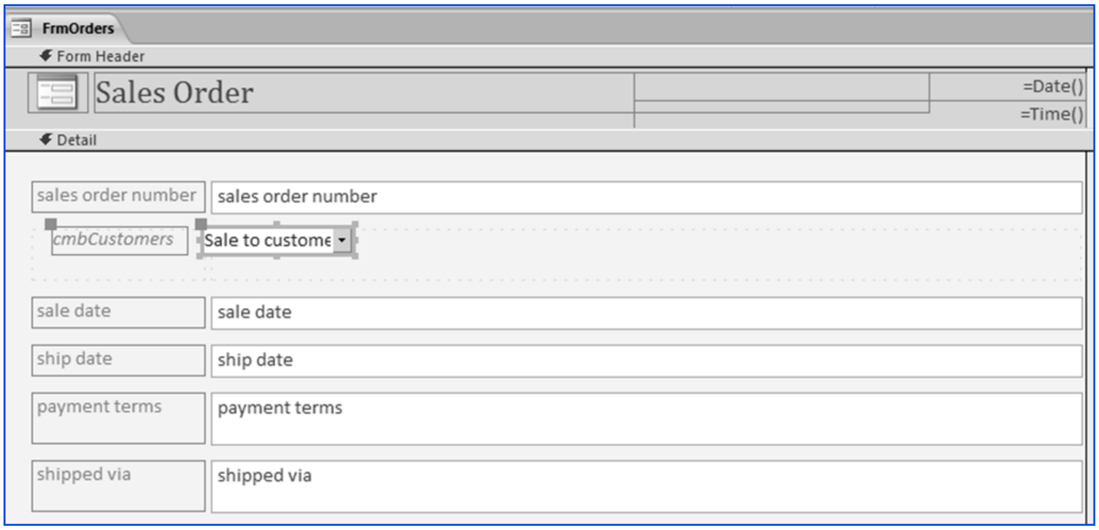
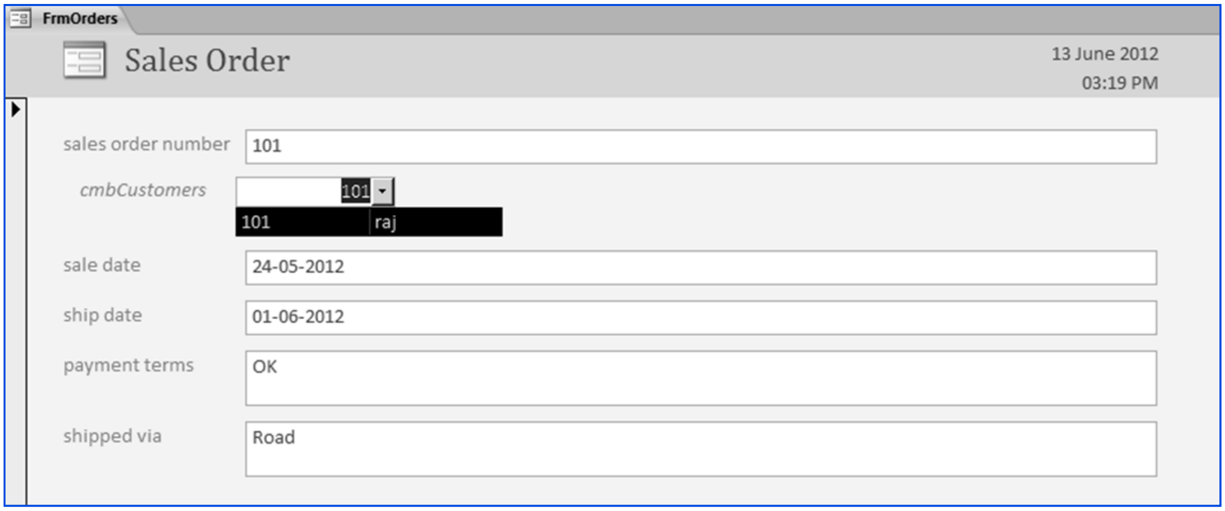


Fig. 2.2.22. Combo Box added in form

13. Set the formatting of the label and combo box to make it more presentable. Open the form in *Form View* to verify the added combo box. The form should appear, as shown in Fig. 2.2.23.



Sales Order

13 June 2012
03:19 PM

sales order number	101
cmbCustomers	101 101 raj
sale date	24-05-2012
ship date	01-06-2012
payment terms	OK
shipped via	Road

Fig. 2.2.23. The Combo Box List appears

2.3 Make effective use of Forms

Most databases provide forms for data entry and for viewing data. Access 2010 provides few techniques that help enhancing the usability of forms, and in turn, the productivity of the users of database. Forms are used by most of the users on a regular basis, and hence it is very crucial to make the effective use of forms. Adding more capabilities to forms helps to save user time and work, and also present data in a more organised way. Some of the extended functionality of the form can be:

- Adding calendar control on a form
- Organising information with tab pages
- Displaying a summary with tab pages

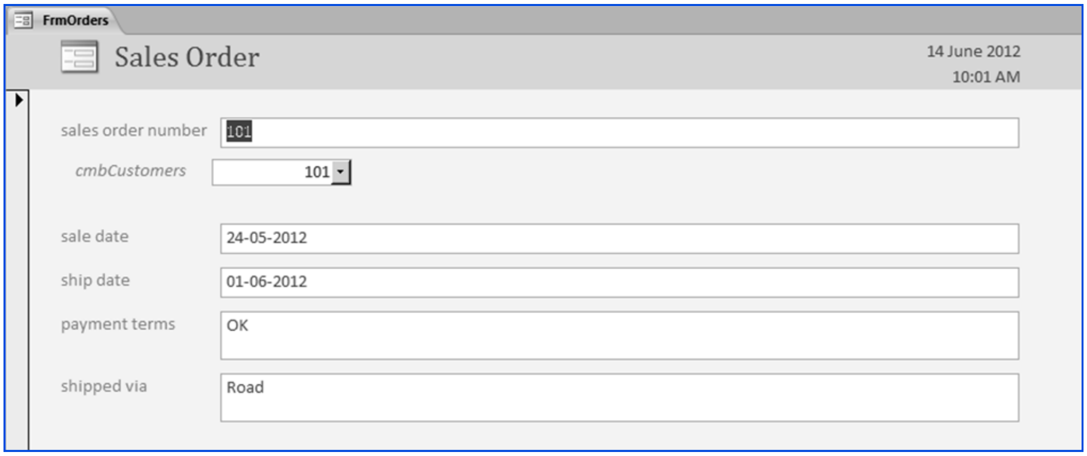
2.3.1 Displaying a calendar control on a Form

To make forms more presentable and user friendly, different types of controls can be added to forms. These controls can be as simple as controls that are available in the *design ribbon -> controls* tab, or can be a third party ActiveX control. ActiveX controls are usually graphical objects that do not operate as standalone solutions, and they run only in the Windows environment.

Calendar control is one of the most popular ActiveX controls. While working with dates, it is always helpful to have a calendar nearby. If a form contains a date field, it is always good to add a calendar control which displays dates graphically and make the form more user-friendly. The calendar control provides properties that enable to set and retrieve dates in/from a table.

2.3.2 Problem Scenario

Database Designer of Apex Ltd. has designed the *frmOrders* form to be used for *Sales Orders* table, as shown in Fig. 2.3.1 The Supervisor of the Customer Service Group informs the designer that the people taking orders often need to refer to a calendar to answer customer questions, such as when they will receive a shipment. A calendar is necessary so that the customer service employees can take weekends and holidays into account when they make an estimate as to when orders will be shipped.



Sales Order

14 June 2012
10:01 AM

sales order number	101
cmbCustomers	101
sale date	24-05-2012
ship date	01-06-2012
payment terms	OK
shipped via	Road

Fig. 2.3.1. frmOrders Form

Solution

A Calendar control is added to all the date fields to make the form convenient for the users. The Database Developer adds a calendar control for Sales Date and Ship Date for users to pick up date graphically.

Steps to add a Calendar Control

1. Right-click the *frmOrders* and select *Design View* from the dropdown to open the form in *Design View*, as shown in Fig. 2.3.2.

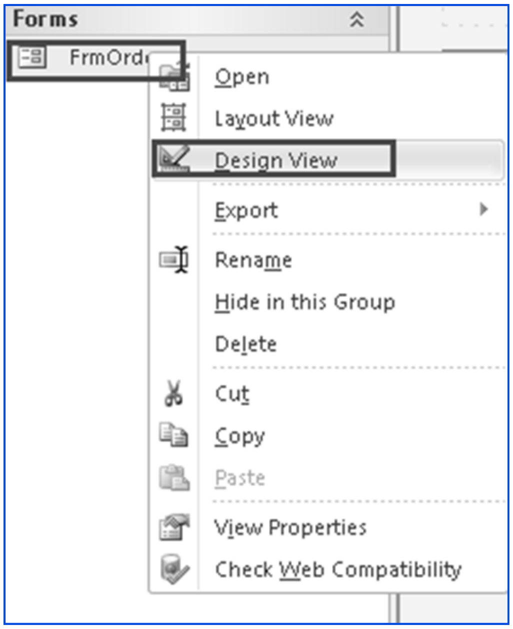
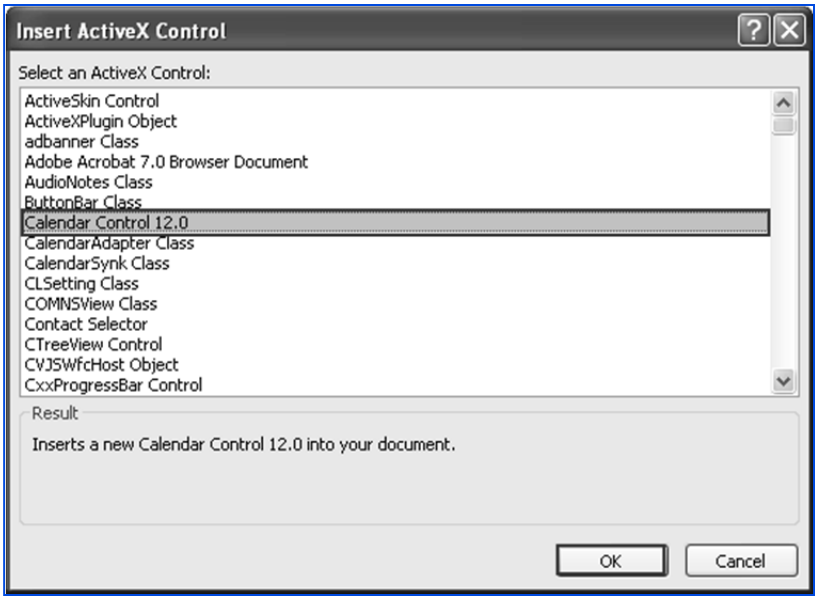


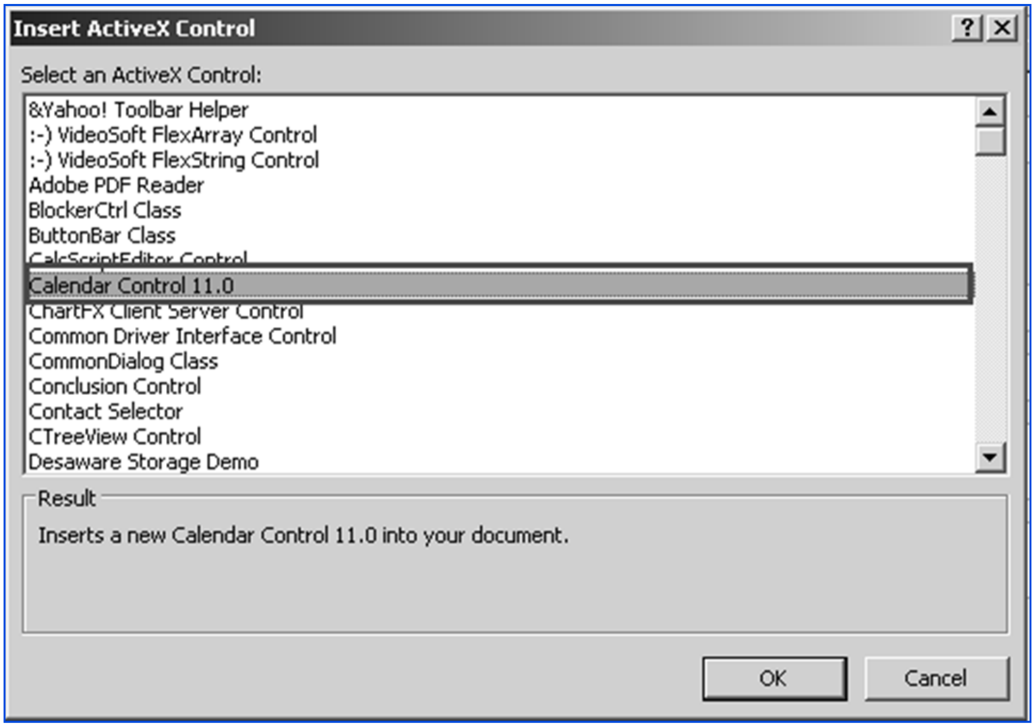
Fig. 2.3.2. Open form frmOrders in Design View

2. From *Design ribbon* -> *Controls* tab, click the *Insert ActiveX Control Command* .

3. The *Insert ActiveX* control window appears. Select*Calendar Control 11.0* from the list and click *OK*, as shown in Fig. 2.3.3.



Inserts a new Calendar Control 12.0 into your document.



Insert ActiveX Control

Select an ActiveX Control:

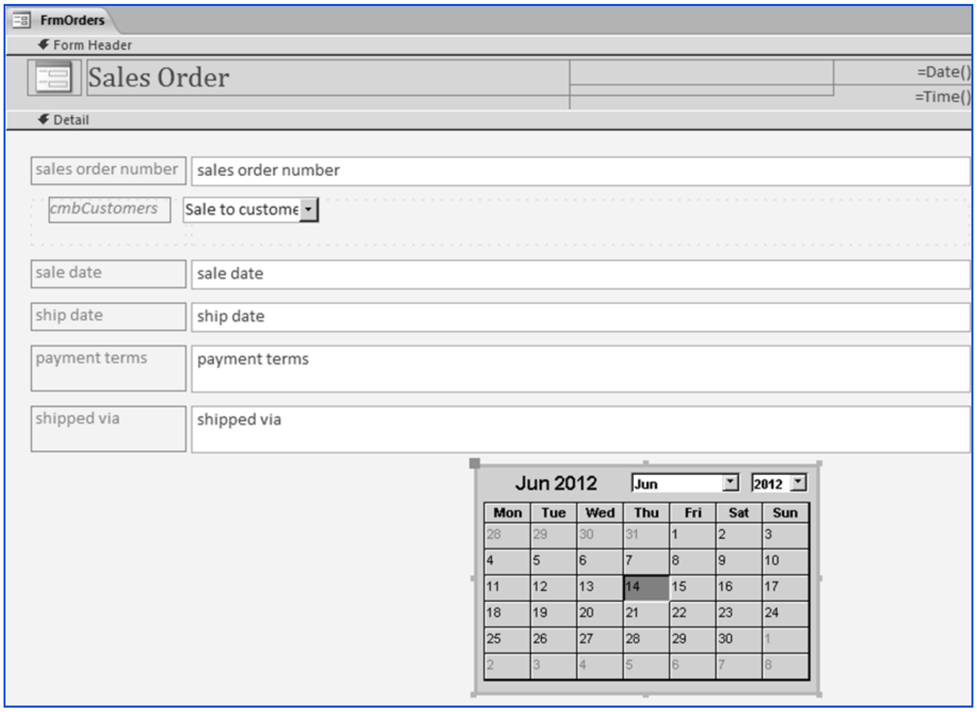
&Yahoo! Toolbar Helper
:-) VideoSoft FlexArray Control
:-) VideoSoft FlexString Control
Adobe PDF Reader
BlockerCtrl Class
ButtonBar Class
CalcScriptEditor Control
Calendar Control 11.0
ChartFX Client Server Control
Common Driver Interface Control
CommonDialog Class
Conclusion Control
Contact Selector
CTreeView Control
Desaware Storage Demo

Result

Inserts a new Calendar Control 11.0 into your document.

Fig. 2.3.3. Select Calendar control

4. The Calendar control is placed in the upper-left corner of the form. Drag it to the required position. The *frmOrder*s after inserting the Calendar control appears, as displayed in Fig. 2.3.4.

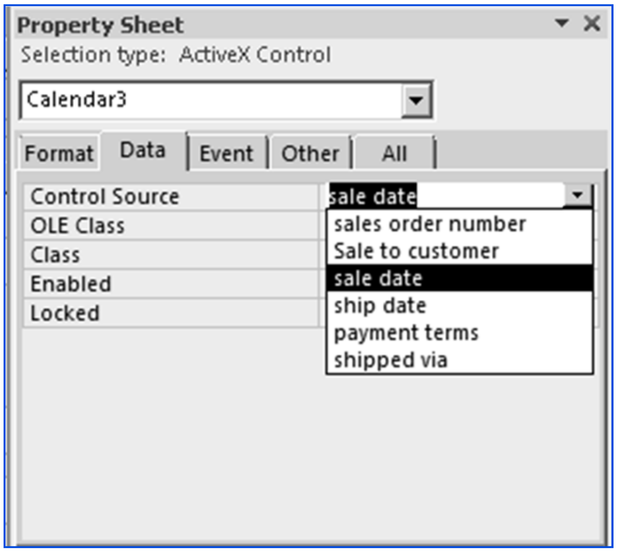


Mon	Tue	Wed	Thu	Fri	Sat	Sun
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

Fig. 2.3.4: Calendar control inserted on the form

5. Select the Calendar control and press *F4* to open the *Property Sheet* for the Calendar control

6. Move to the data tab in property sheet. Click the arrow next to the *Control Source* property, and choose *Sales Date* from the list, as shown in Fig. 2.3.5.



Selection type: ActiveX Control

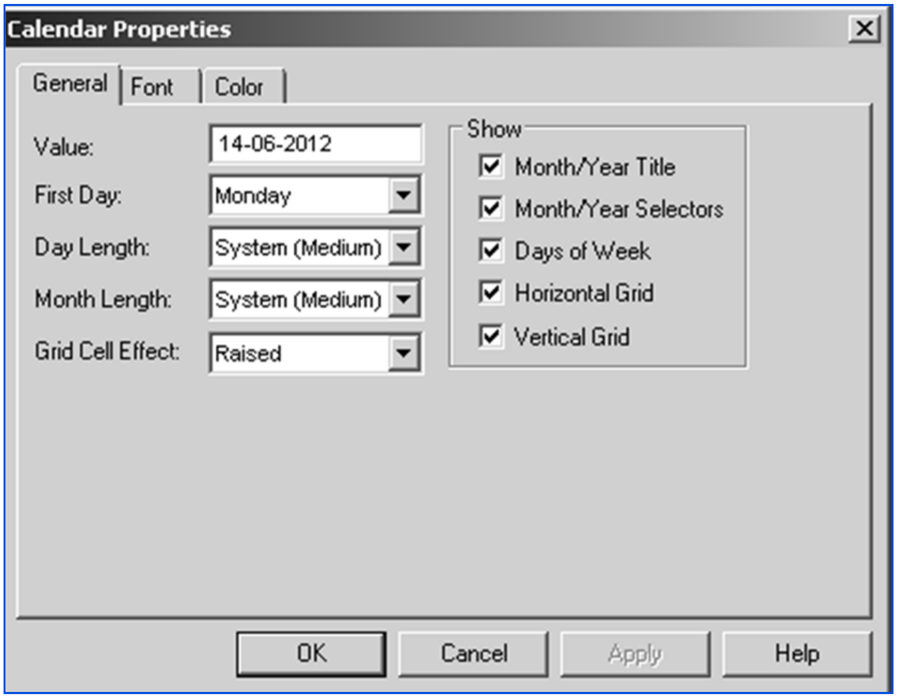
Calendar3

Control Source: sale date

Fig. 2.3.5. Set the Control Source property

NOTE: The Calendar control has many properties that you can set to create a custom appearance.

7. To set other properties of Calendar control, click on ... button in the *Custom property in Other* tab. The Calendar properties appear, as shown in Fig. 2.3.6.



Calendar Properties

Value: 14-06-2012

First Day: Monday

Day Length: System (Medium)

Month Length: System (Medium)

Grid Cell Effect: Raised

Show

☑ Month/Year Title

☑ Month/Year Selectors

☑ Days of Week

☑ Horizontal Grid

☑ Vertical Grid

Fig. 2.3.6. Calendar Properties

2.3.3 Organising information with Tab Pages

A Tab control is an Access 2010control that allows user to create multiple pages in one form. Each page is separated by its own tab and becomes active when the user selects a tab. Tab controls are useful for presenting grouped information that can be assembled by category. A tab control has pages, each with a tab of its own. Each tab page can contain all types of controls, such as text boxes, combo boxes, images, and even command buttons.

2.3.4 Problem Scenario

The Sales Manager asks the Information Analyst to store the information of the customer which could be displayed in a friendly manner, such that the customer's personal details and company details can be viewed separately.

Solution

The Information Analyst advises the developer to divide the information into separate tab in a form. One tab in the form should contain customer contact detail, while the other can contain customer's company details, and third tab can contain customer's terms with the company.

Steps to add tab in a form

1. On the *Create ribbon* -> *Forms* tab, click *Blank Form* to create a new form, as shown in Fig. 2.3.7.

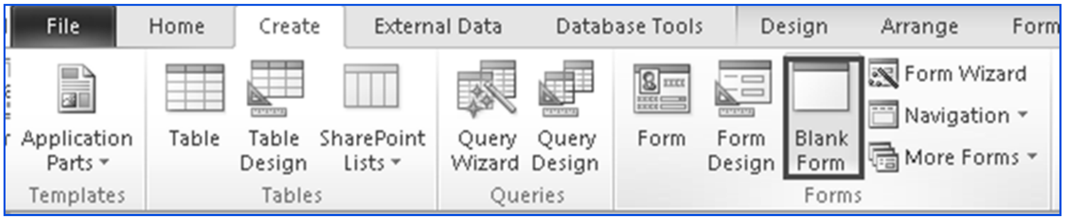
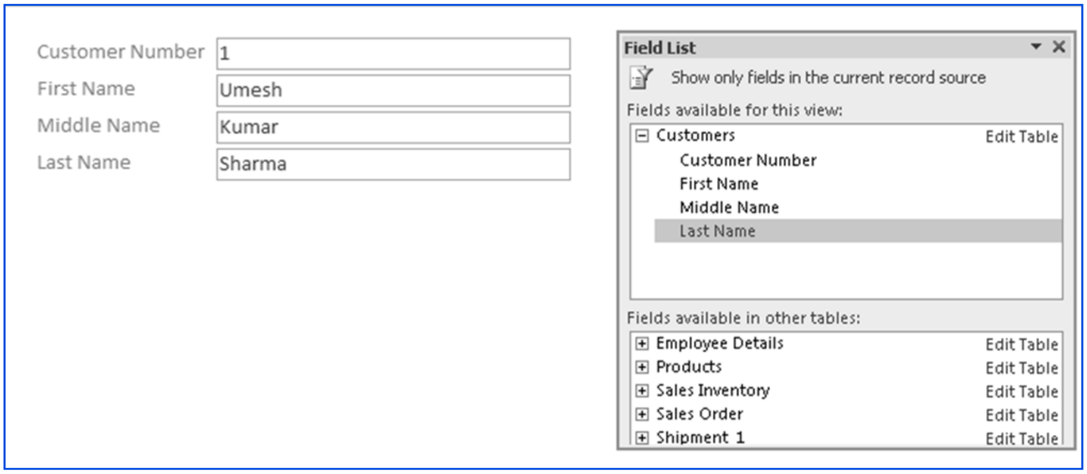


Fig. 2.3.7: Open a blank Form

2. From the *Field List* window, open the plus sign with *Customers* table and drag the fields *Customer Number*, *First Name*, *Middle Name* and *Last Name* to the form. The form appears, as shown in Fig. 2.3.8.



Customer Number	1
First Name	Umesh
Middle Name	Kumar
Last Name	Sharma

Fig. 2.3.8. Add Fields to a blank form

NOTE: If the field list window does not appear, click on *Add Existing Field* from *Design ribbon* -> *Tools* tab.

3. Click *Home ribbon* -> *Views tab* -> *View* and select *Design View* from the dropdown to open the form in *Design View*. The form appears in *Design View*, as shown in Fig. 2.3.9.

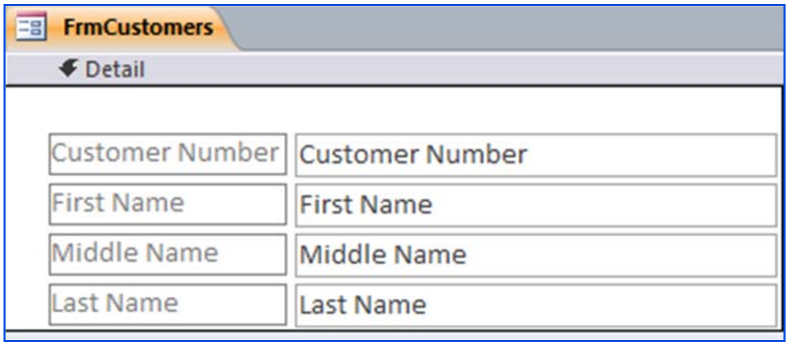
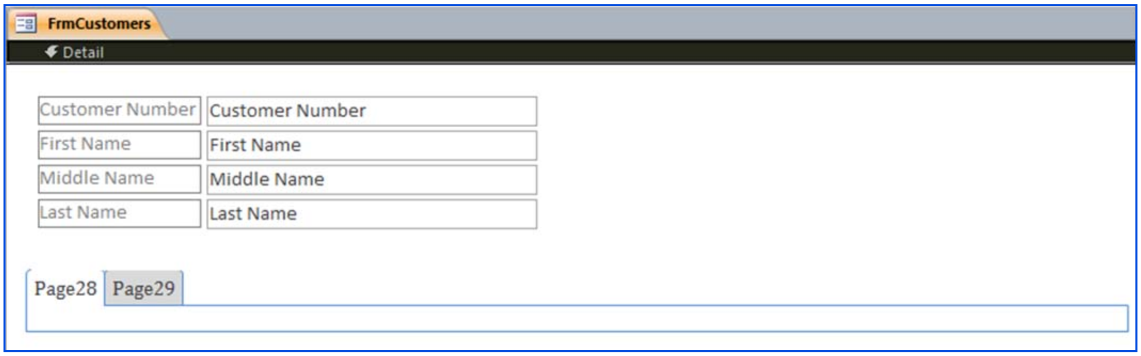


Fig. 2.3.9. Form in Design View

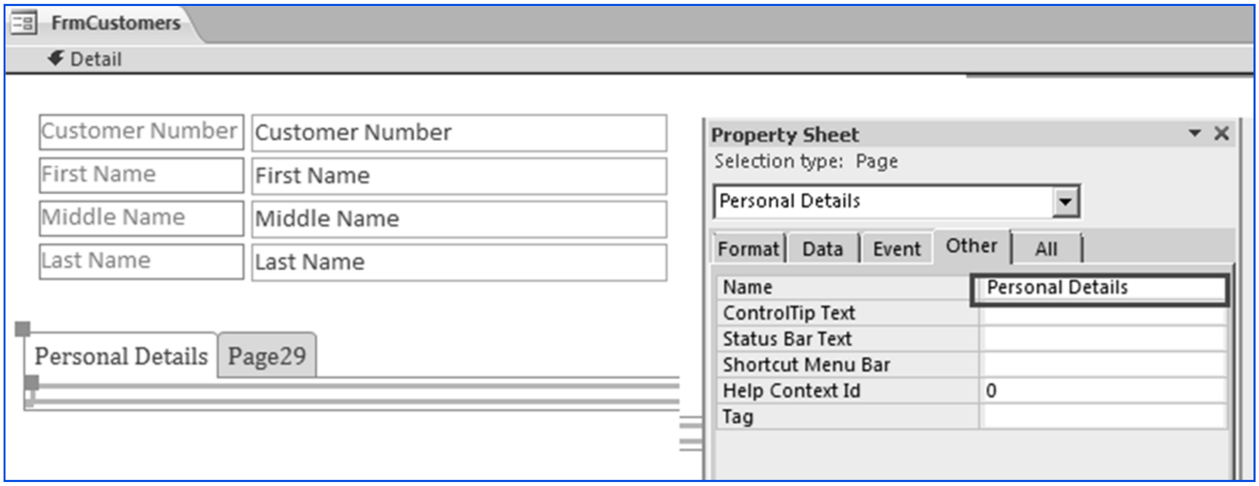
4. Extend the size of the form as required. Click on tab control in *Design ribbon -> Control*s tab and draw it into the form. The form appears, as shown in Fig. 2.3.10.



Customer Number	Customer Number
First Name	First Name
Middle Name	Middle Name
Last Name	Last Name

Fig. 2.3.10. Add tab control to a form

5. Double-click the tab *Page28*. Property sheet appears. Type the value *Personal Details* in *Name* property, as shown in Fig. 2.3.11.



Customer Number	Customer Number
First Name	First Name
Middle Name	Middle Name
Last Name	Last Name

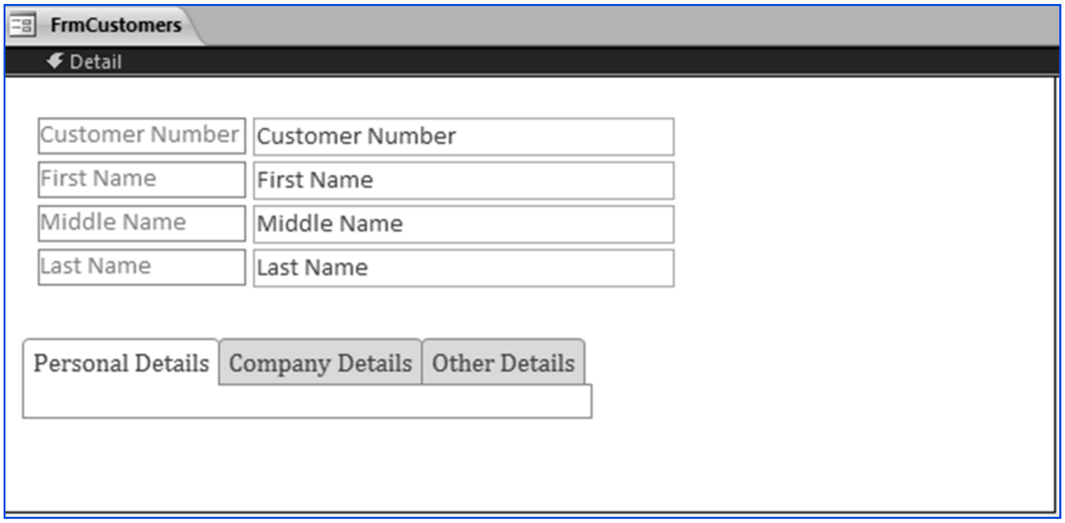
Selection type: Page

Personal Details

Name	Personal Details
ControlTip Text	
Status Bar Text	
Shortcut Menu Bar	
Help Context Id	0
Tag	

Fig. 2.3.11. Change the name of first tab

6. In a similar manner, change the name of the second tab to *Company Details*. Right-click the Tab control and select *Insert Page* from the dropdown to add a new tab and change its name to *Other Details*. The form should appear, as in Fig. 2.3.12.



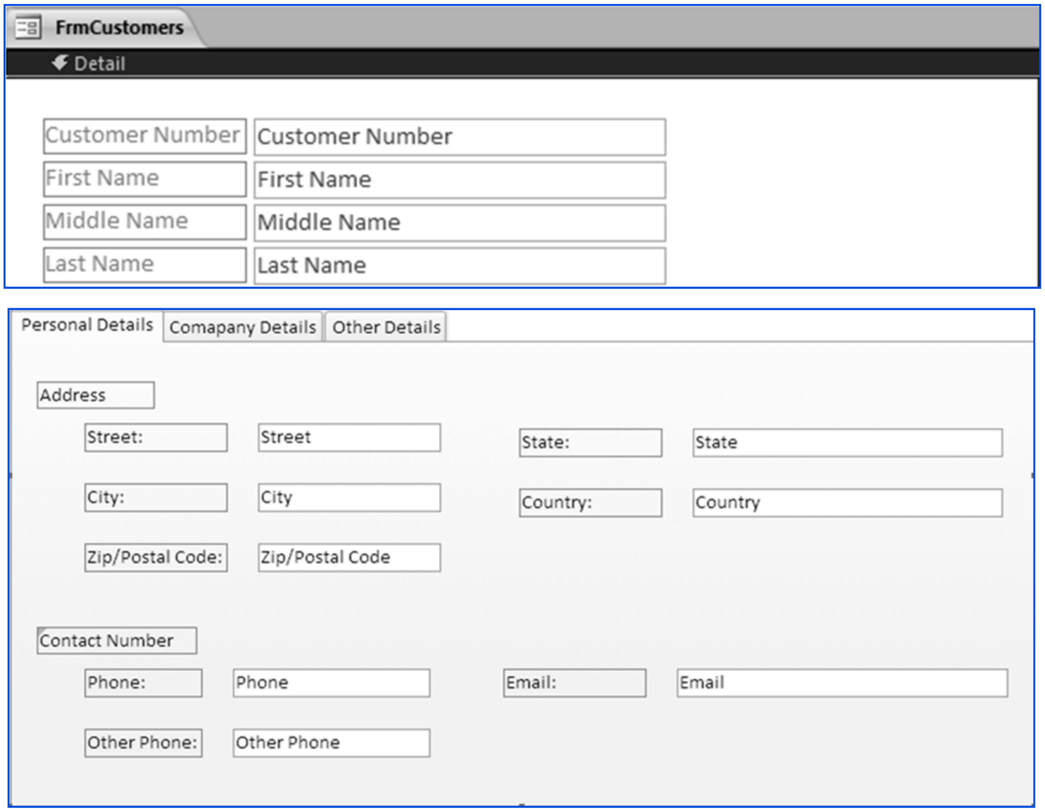
FrmCustomers

Detail

Customer Number	Customer Number
First Name	First Name
Middle Name	Middle Name
Last Name	Last Name

Fig. 2.3.12. Add three tabs to Form

7. Move to the *Personal Details* tab, and select the field *Street, City, Country, Zip/Postal Code, Phone, Other Phone*, and *Email* from the *Field List* window. The resultant form is displayed in Fig. 2.3.13.



FrmCustomers

Detail

Customer Number	Customer Number
First Name	First Name
Middle Name	Middle Name
Last Name	Last Name

Address

Street:	Street	State:	State
City:	City	Country:	Country
Zip/Postal Code:	Zip/Postal Code		

Contact Number

Phone:	Phone	Email:	Email
Other Phone:	Other Phone		

Fig. 2.3.13: Add required field to personal Details tab

8. In a similar manner, add the fields *CompanyName, Company Details* and *Job Title* to the *Company Details* tab and also the fields *Credit Limit, Payment Terms, Comments,* and *Document* submitted to the *Other Details* tab.

9. A tabbed form is prepared. Now the user can navigate between different tabs to view the information required.

10. Save the form as *frmCutomers*.

2.3.5 Displaying a Summary of Data in a Form

Presenting a summary of data can be very useful to users who access database. PivotCharts and PivotTables are created in forms to display a summary of data. Pivot Table is used to summarize and analyzes data in a form. The idea of Pivot Table is to let users slice and dice the data in any way required at a given moment in time. Pivot Table represents the data in spreadsheet form, while Pivot Chart represents the same data in a graphical form. Both Pivot Chart and Pivot Table are different views of a form.

Pivot Table represents the data in tabular form, in which one or more vales are represented in rows, another value in columns, and a summarized value at the intersection of row and column. Forms that lend themselves to be displayed in PivotTable or PivotChart view provide many ways for users to manipulate data. An example of such a form is one that contains information about country, city, salesperson, sales, and date of sale. Such form can be used to determine sales by city and salesperson for each month, or sales in each country for each salesperson during the year.

2.3.6 Problem Scenario

The Sales Manager wishes to see the summarized data of sales. He requests the Database Developer to create a form in such a manner, that it should display the summarized data in terms of cost of Item Sold per year, per country. For example, he needs a report displaying summary of the items sold country wise and state wise in year 2008 Quarter-2.

Solution

The Database Developer first creates a query based on table *Sales Order*, *Customers*, *Sales Item Description*, and *Inventory* and fetches the required data from it. Once query is created, then a form using pivot table option is created over the query.

Steps for creating a query

1. Open the *Query Design* window by clicking *Create ribbon ->Queries tab -> Query Design*.

2. Select the tables *Customers*, *Sales Order*, *Sales Item Description* and *Inventory* table from the *Show Table* window. Click *OK* to close the window. The *Query Design* window appears, as shown in Fig. 2.3.14.

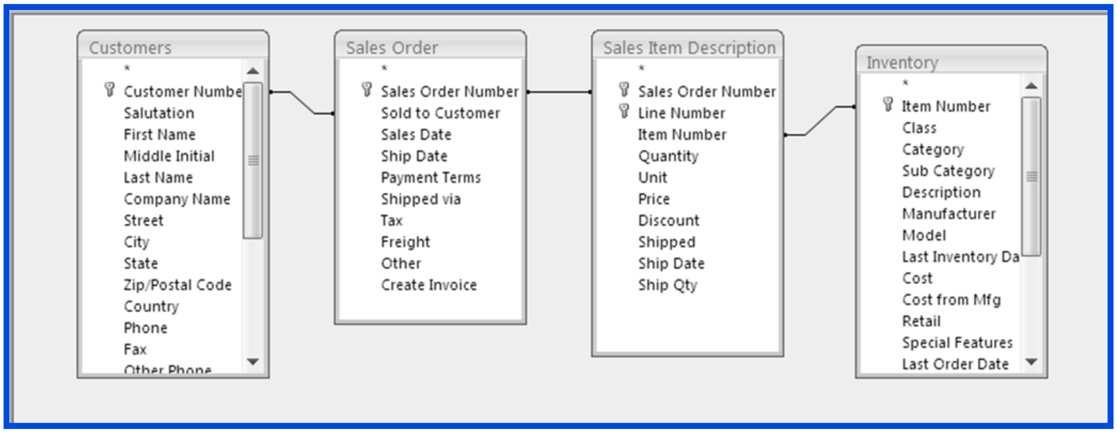
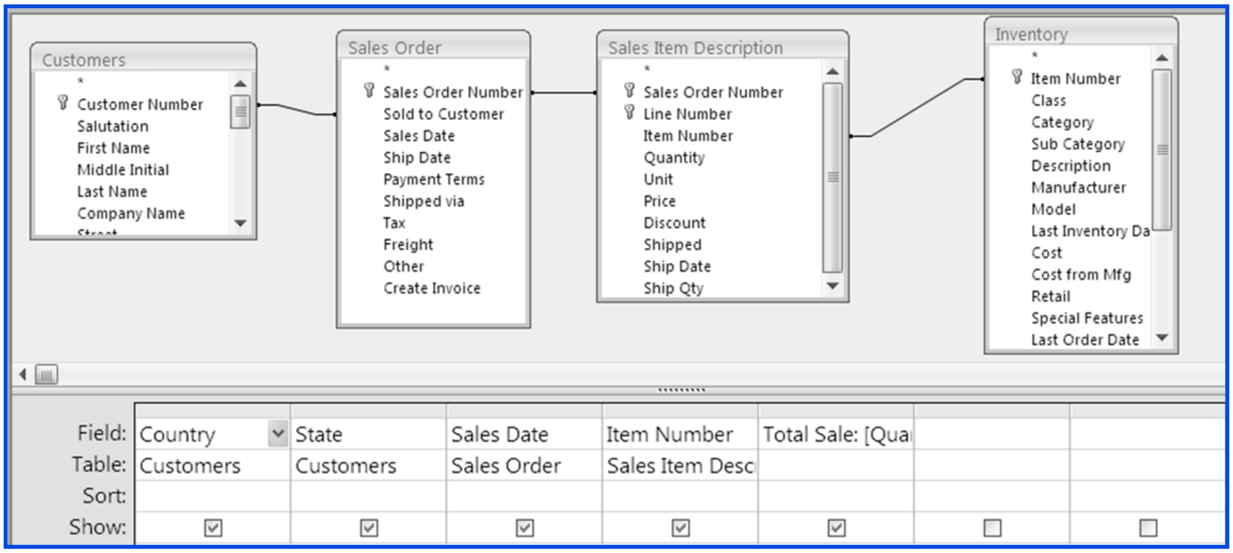


Fig. 2.3.14. Query Design window

3. Select *Country* and *State* from the *Customers* table and drag them to the *Add Columns* tab. Similarly add the column *Sales Date* from *Sales Order* table and *Item Number* from *Inventory* table. Also add a calculated column as *Total Sale: [Quantity] * [Price]*. The resultant query window is displayed in Fig. 2.3.15.



Field:	Country	State	Sales Date	Item Number	Total Sale: [Qua		
Table:	Customers	Customers	Sales Order	Sales Item Desc			
Sort:							
Show:	☑	☑	☑	☑	☑	☐	☐

Fig. 2.3.15. The query Design window

4. Save the query as *SummarizedData* and run to confirm the results.

Steps for creating a Pivot Table Form

1. Select the query *SummarizedData* under *All Access Objects -> Queries* tab. From the *Create ribbon -> Forms tab ->*, click the dropdown arrow near *More Forms* and select *Pivot Table* from the list, as indicated in Fig. 2.3.16.

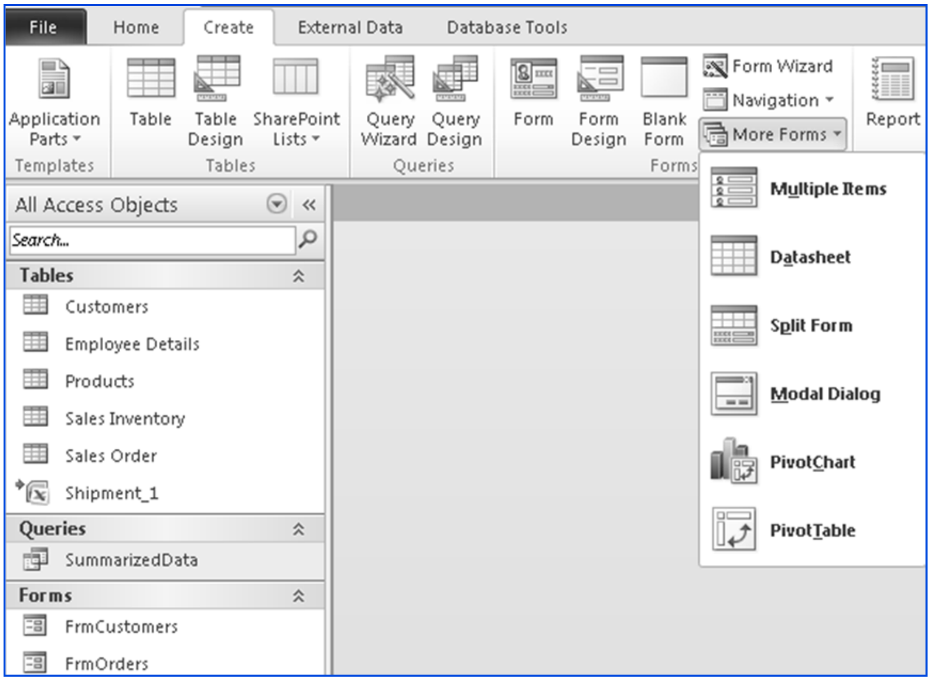


Fig. 2.3.16. Select the Pivot Table option

2. The *Pivot Table* window appears along with the field list, as shown in Fig. 2.3.17.

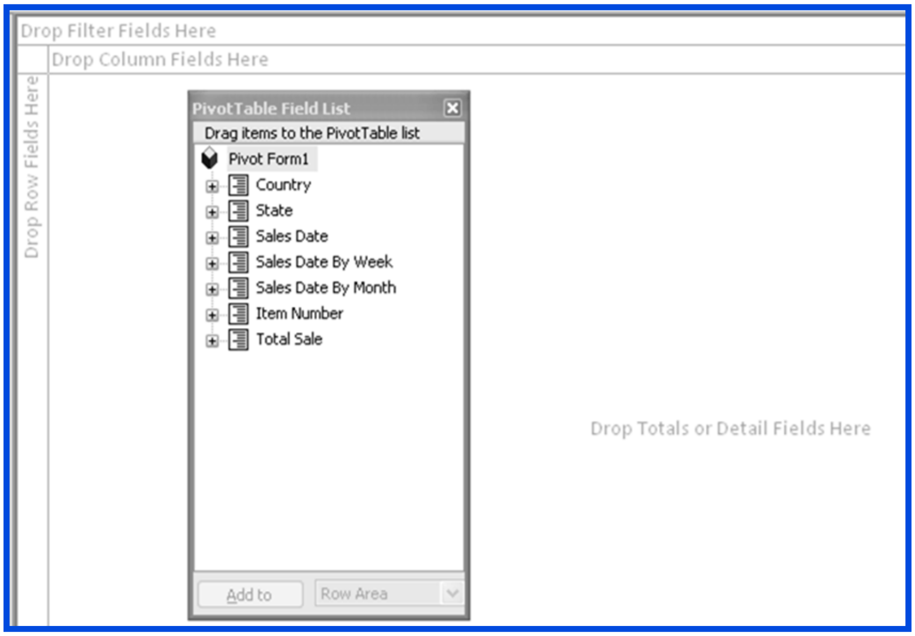


Fig. 2.3.17. The Pivot Table window

NOTE: If the *Field List* window does not appear, click *Design ribbon* -> *Show/Hide* tab and *Field List*.

3. Open the plus sign of the field *Item Number* and drag the field to *Drop Row Fields Here*. Open the plus sign of *Country* and *State* drag *Country* and then *State* to *Drop Column Fields Here*. Similarly, open the plus sign for *Total Sale* and drag the column *Total Sale* to *Drop Totals or Detail Fields Here*, and open the plus sign of *Sales Date By Month* and drag the fields *Years* and *Quarters* to *Drop Filter Field Here*. The resultant *Pivot Table* window is shown in Fig. 2.3.18.

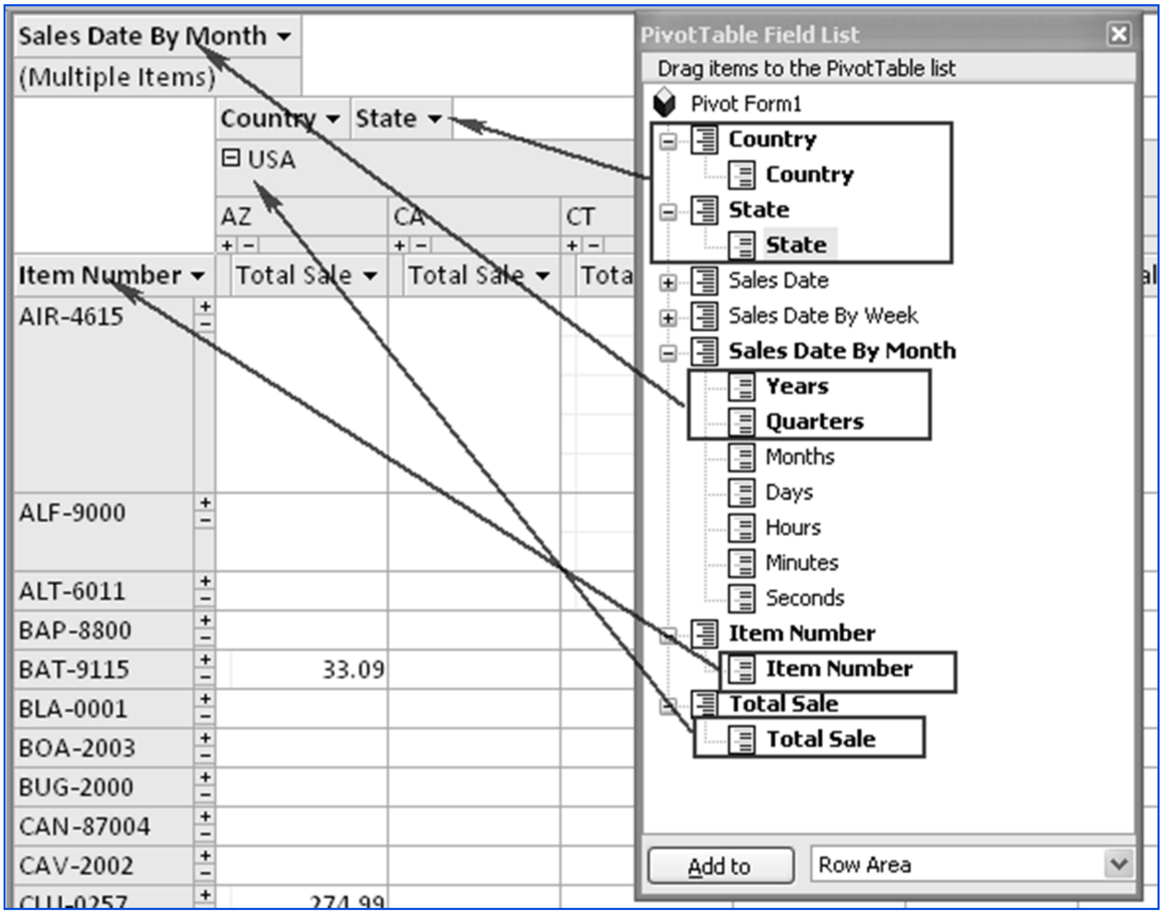


Fig. 2.3.18. Pivot Table window

4. Click the dropdown arrow of *Sales Date by Month* field. Deselect the *Select All* option and open the plus sign of year 2008, and select Qtr2 from the year 2008, as shown in Figure 2.3.19.

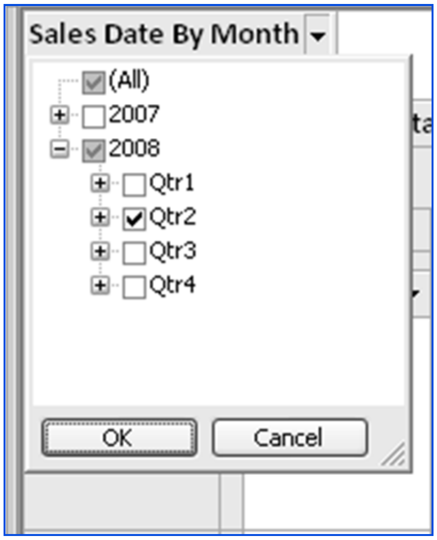


Fig. 2.3.19. Select the required year

5. The *Pivot Table* window displays the sales made for each item country and state wise in Quarter2 of year 2008.

   NOTE: Pivot Table gives you a flexibility of selecting the desired data and view the data according to your requirements.

2.4 Advanced Reports

Reports are one of the best ways to represent data. Reports can be made more advanced and more user friendly to represent the data in a more organized form. The Reports can include customized Headers and Footers such as to display the company Logo or the department. Also the calculated controls can be added to report to display computed values.

2.4.1 Creating Customised Headers and Footers

Reports can include pairs of header and footer sections:

- Report Header and Footer for printing information at the beginning and end of the report.
- Page Header and Footer for printing information at the top and bottom of each page.
- Group Header and Footer for printing information when the group starts or the group ends, if groups exist in the report.

To add a header/footer pair, right-click in the report design and choose *Page Header/Footer* or *Report Header/ Footer* from the shortcut menu. Page and report headers and footers are added as pairs, while In Group only Headers can be added.

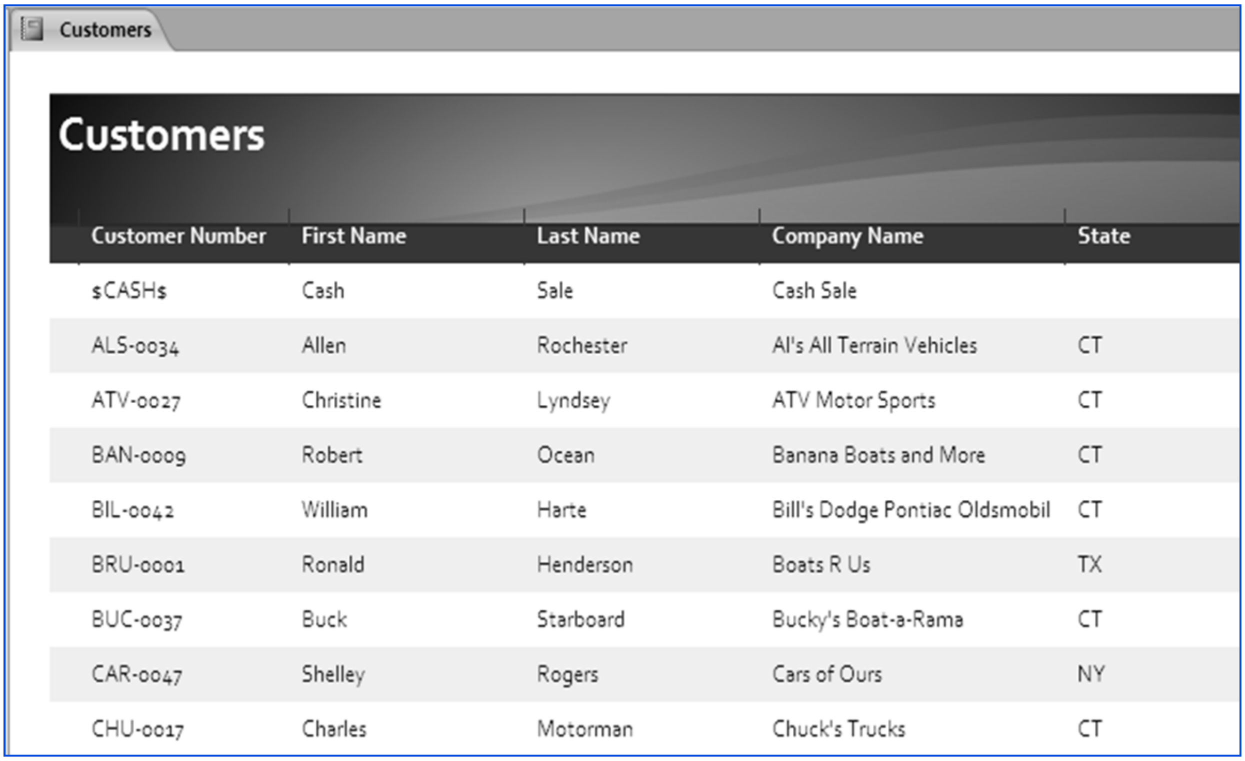
Controls in the *Report Header & Footer* section are printed only once at the beginning and the end of the report. A common use of a Report Header section is as a cover page or a cover letter, or for presenting information that needs to be communicated only once to the user of the report. The Report Footer section can be used to display the summarized data of whole report, like author of report, date & time, etc.

Controls in the *Page Header & Footer* section are normally printed at the top and bottom of every page. Typically, Page Headers serve as column headers in group/total reports; they can also contain a title for the report. A Group Header section normally displays the name of the group. Group Headers immediately precede *Detail* sections. A Group Header is added to the report if any grouping is done in the report.

Each *Header& Footer* section in reports can be customized from their traditional look to meet user's requirement. Different controls can be added to header and footer. The textboxes or labels can be used to display a user-defined message. Various functions and expression can be used to display summary results and computed data.

2.4.2 Problem Scenario

The Database Developer has prepared a report *rptCustomers* to display all the details of the customers, as shown in Fig. 2.4.1. The Zonal Sales Head requires the report to be customized; each page of the report must contain the *Report Title* at the top and date & time at bottom.



Customers

Customer Number	First Name	Last Name	Company Name	State
$CASH$	Cash	Sale	Cash Sale	
ALS-0034	Allen	Rochester	Al's All Terrain Vehicles	CT
ATV-0027	Christine	Lyndsey	ATV Motor Sports	CT
BAN-0009	Robert	Ocean	Banana Boats and More	CT
BIL-0042	William	Harte	Bill's Dodge Pontiac Oldsmobil	CT
BRU-0001	Ronald	Henderson	Boats R Us	TX
BUC-0037	Buck	Starboard	Bucky's Boat-a-Rama	CT
CAR-0047	Shelley	Rogers	Cars of Ours	NY
CHU-0017	Charles	Motorman	Chuck's Trucks	CT

Fig. 2.4.1: Report rptCustomers

Solution

The Database Developer adds a Page Header & Footer to the report and customizes the *Page Header* to hold a label with *Report Title*, and *Page Footer* to contain the date & Time using *Date Time* control.

Steps to customize Page Header & Footer of the Report

1. Right-click the report *rptCustomers* under *All Access Object* -> *Reports* tab and select *Design View* from the dropdown to open the report in *Design View*. The *Report Design View* should appear, as in Fig. 2.4.2.

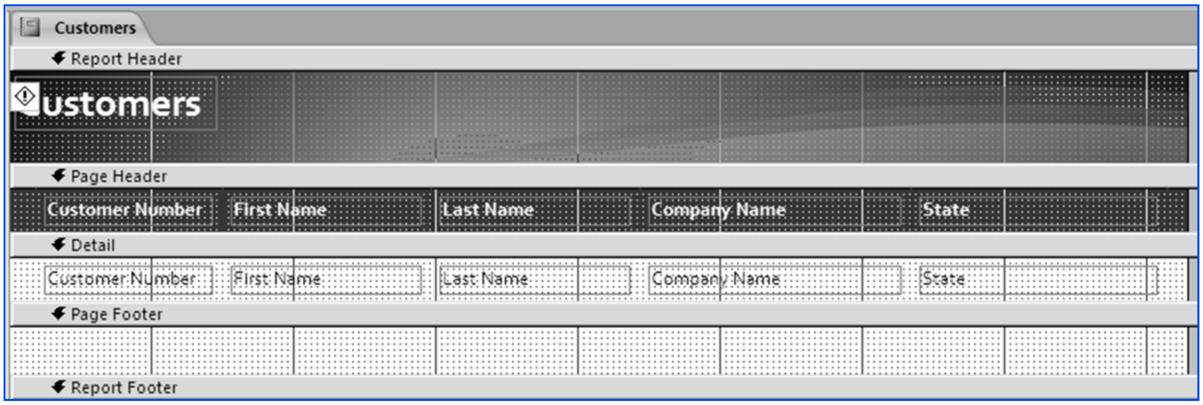


Fig. 2.4.2: Report rptCustomers Design view

2. Expand the *Page Header* section. Select a label control Aa from *Design ribbon* -> *Controls* tab and draw it on the *Page Header* section, as indicated in Fig. 2.4.3.

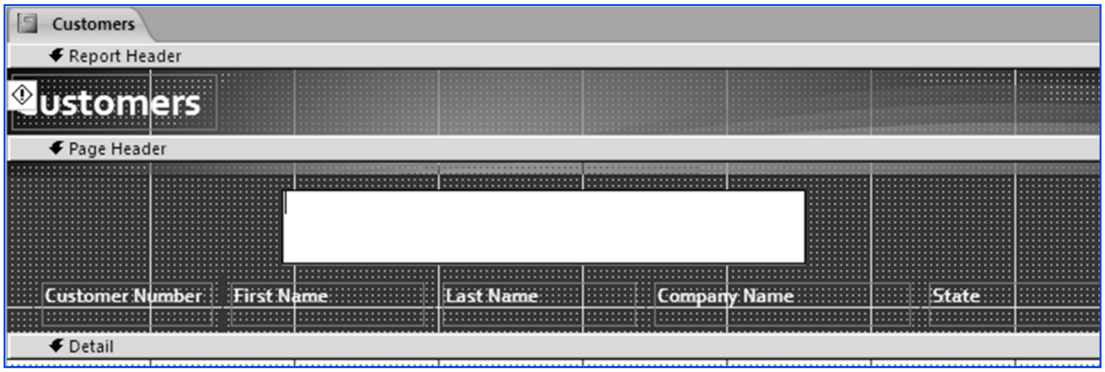


Fig. 2.4.3. Add Label to Page Header

NOTE: If *Report Header* does not appear right-click the report and select *Page Header /Footer* to view them.

3. Type the text *Customers* into the label and format it according to the requirement, as shown in Fig. 2.4.4.

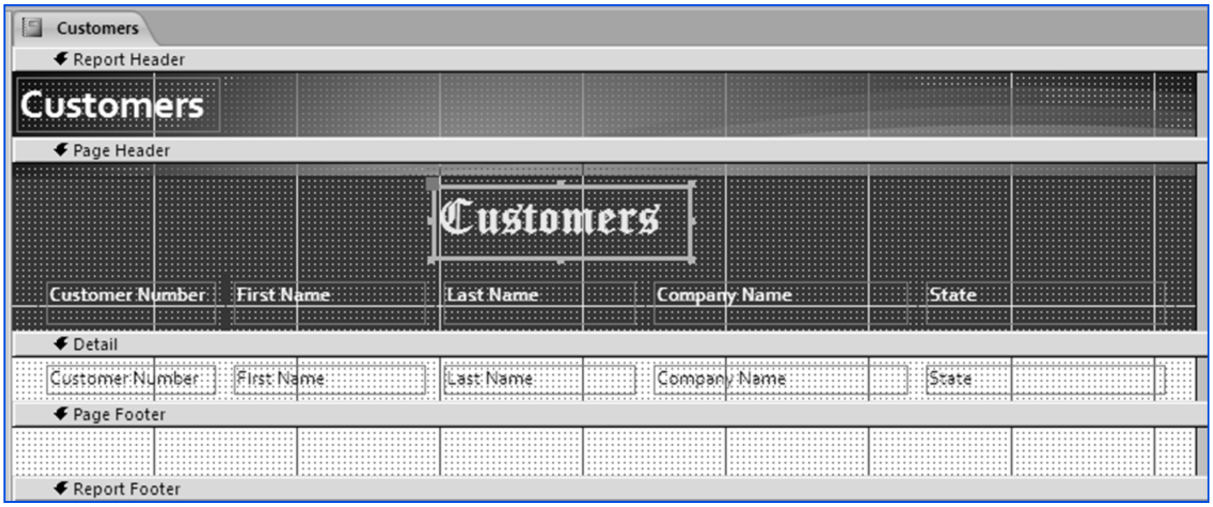


Fig. 2.4.4. Add Title to Label

4. In a similar manner, add a Textbox control ab| to the *Page Footer* of the report from *Design ribbon -> Controls* tab. The resultant report appears, as shown in Fig. 2.4.5.

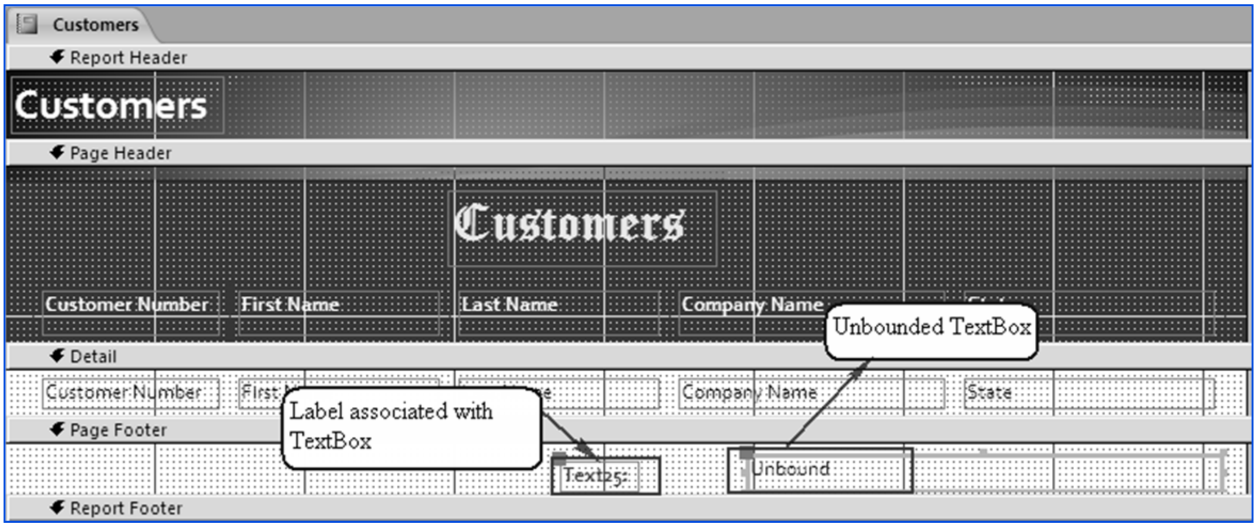


Fig. 2.4.5: Add Textbox to the Report

5. Delete the label of the Textbox and type =*Now()* in the textbox. The report appears, as shown in Fig. 2.4.6.

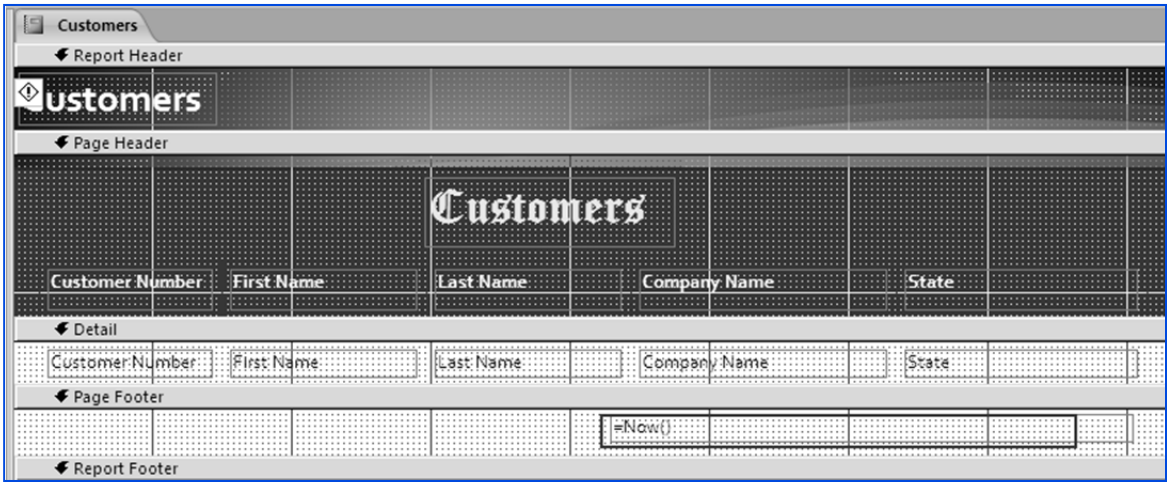
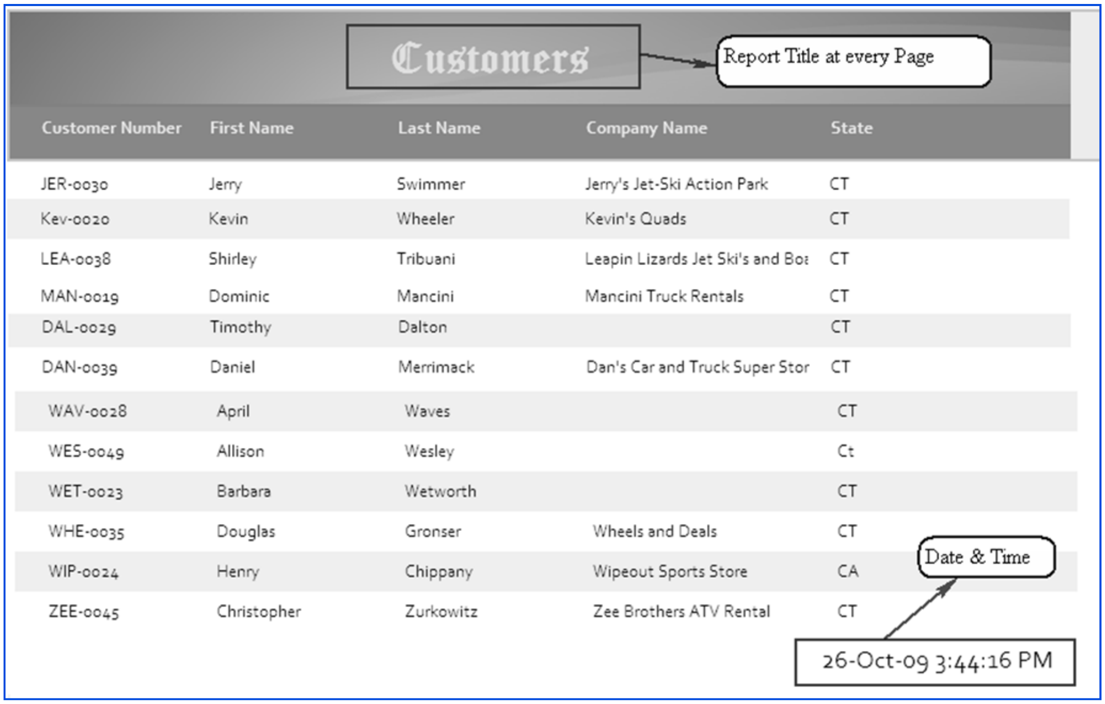


Fig. 2.4.6. Add Textbox in Page Footer

6. Right-click the report and deselect *Report Header/ Footer* to remove the report header and footer. This is optional. The report header can contain some different text and can be displayed.

7. Open the report in *Report View* format by selecting *Report View* from *Home ribbon* -> *Views* tab to verify the data. The report should appear as in Fig. 2.4.7. Note that each page of the report contains the Report Title and Date/Time.



Customer Number	First Name	Last Name	Company Name	State
JER-0030	Jerry	Swimmer	Jerry's Jet-Ski Action Park	CT
Kev-0020	Kevin	Wheeler	Kevin's Quads	CT
LEA-0038	Shirley	Tribuani	Leapin Lizards Jet Ski's and Boa	CT
MAN-0019	Dominic	Mancini	Mancini Truck Rentals	CT
DAL-0029	Timothy	Dalton		CT
DAN-0039	Daniel	Merrimack	Dan's Car and Truck Super Stor	CT
WAV-0028	April	Waves		CT
WES-0049	Allison	Wesley		Ct
WET-0023	Barbara	Wetworth		CT
WHE-0035	Douglas	Gronser	Wheels and Deals	CT
WIP-0024	Henry	Chippany	Wipeout Sports Store	CA
ZEE-0045	Christopher	Zurkowitz	Zee Brothers ATV Rental	CT

26-Oct-09 3:44:16 PM

Fig. 2.4.7. Report View

2.4.3 Adding Calculated Values

Reports in Access 2010 can be modified to display the data as per user requirements. In general, reports contain fields from the table, but many times it becomes necessary to add some computed values in reports. These computed values may count, sum, or calculate an average of the numeric values in a group created in the report. Also, these computed values can be calculated using one or more fields in a table, for example, to display the total cost by multiplying quantity and price.

The calculated values in reports are added using a text box and specifying an expression. The liberty of displaying computed values helps to display the reports in a user-friendly format. The calculated values in the report can be displayed in two ways: using queries with calculated columns and creating reports based on them, or by computing values in the report itself using *Expression* and *Functions*. The calculated values in queries are displayed as fields in reports which use them.

2.4.4 Problem scenario

Consider the report *rptCustomers* discussed in section 3.4.1.The Sales Head wishes the customer name should be displayed as a complete name, not as first name and last name. He requests the Database Developer to implement the change.

Solution

The Database Developer adds a calculated value that concatenates first name and last name in the report *rptCustomers* and replaces the *First Name* and *Last Name* textboxes.

Steps to add Calculated Values

1. Right-click the report *rptCustomers* under *All Access Objects -> Reports* tab and select *Design View* from the dropdown. The report appears in *Design View*, as shown in Fig. 2.4.8.

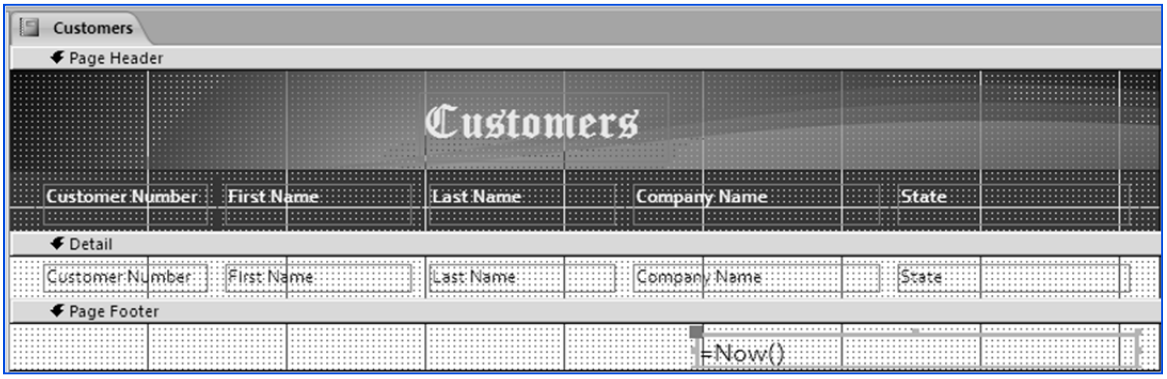


Fig.2.4.8: report in Design View

2. Select the *First Name* and *Last Name* textboxes and delete them. Add a new Textbox control abl from *Design ribbon -> Controls* tab.

3. Drag the *Textbox* in place of *First Name* and *Last Name* textboxes. The report should appear, as shown in Fig. 2.4.9.

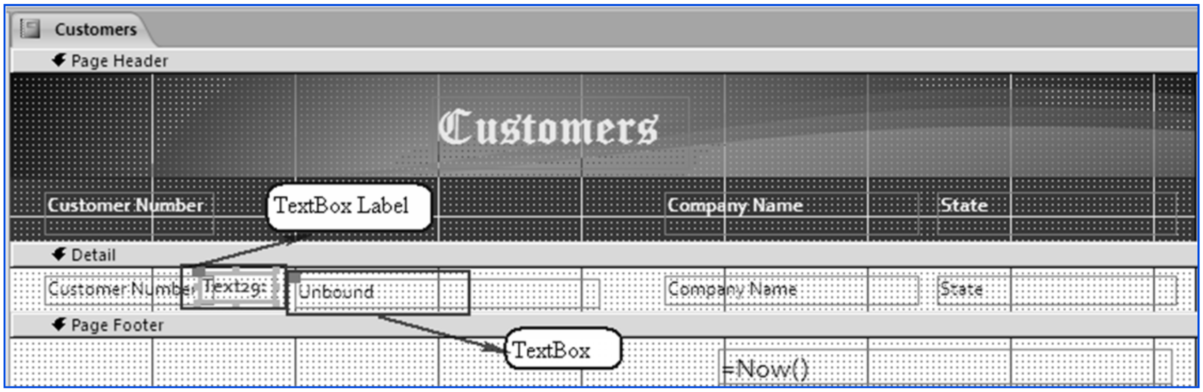


Fig. 2.4.9. Add Textbox control to Report

4. Delete the *Textbox Label* and type the text = *[First Name]* + *" "* + *[Last Name]* in the textbox. The report window appears, as shown in Fig. 2.4.10.

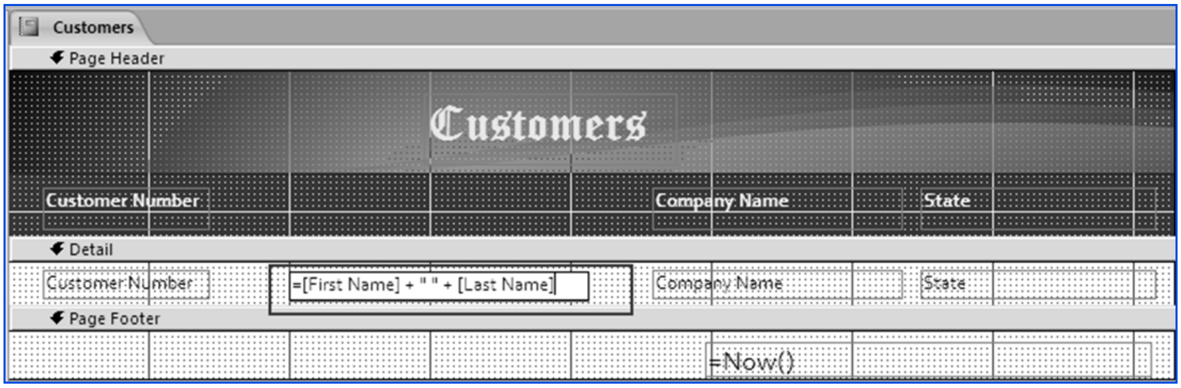


Fig. 2.4.10. Write the expression in calculated control

5. Add a label control Aa from *Design ribbon* -> *Controls* tab in the Page Header tab along with other headings. The Report should appear, as in Fig. 2.4.11.

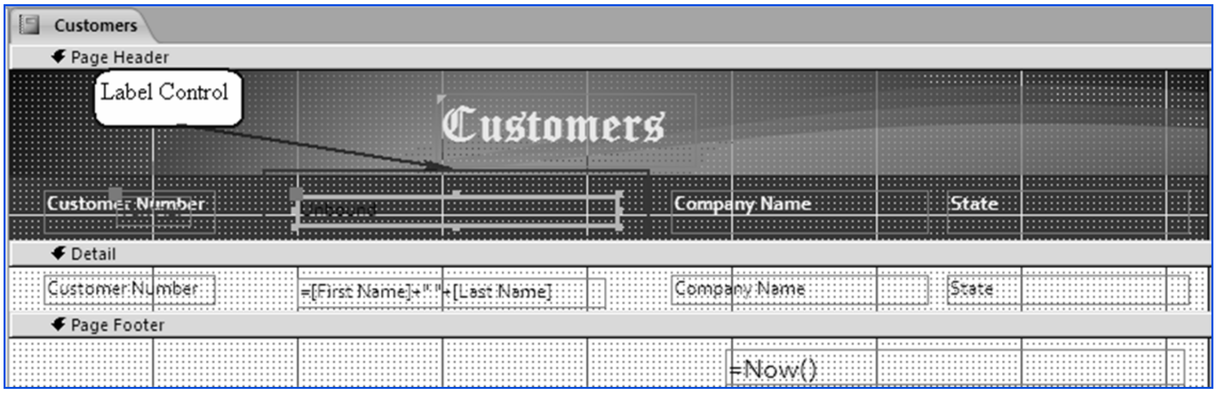


Fig. 2.4.11. Add Label Control to Page Header

6. Type text *Name* in label and format the label accordingly. The Report should appear, as shown in Fig. 2.4.12.

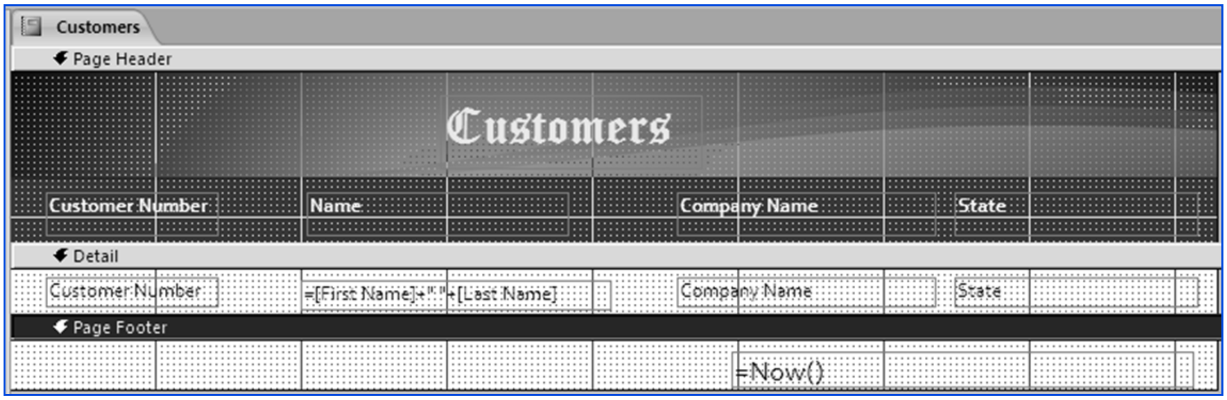
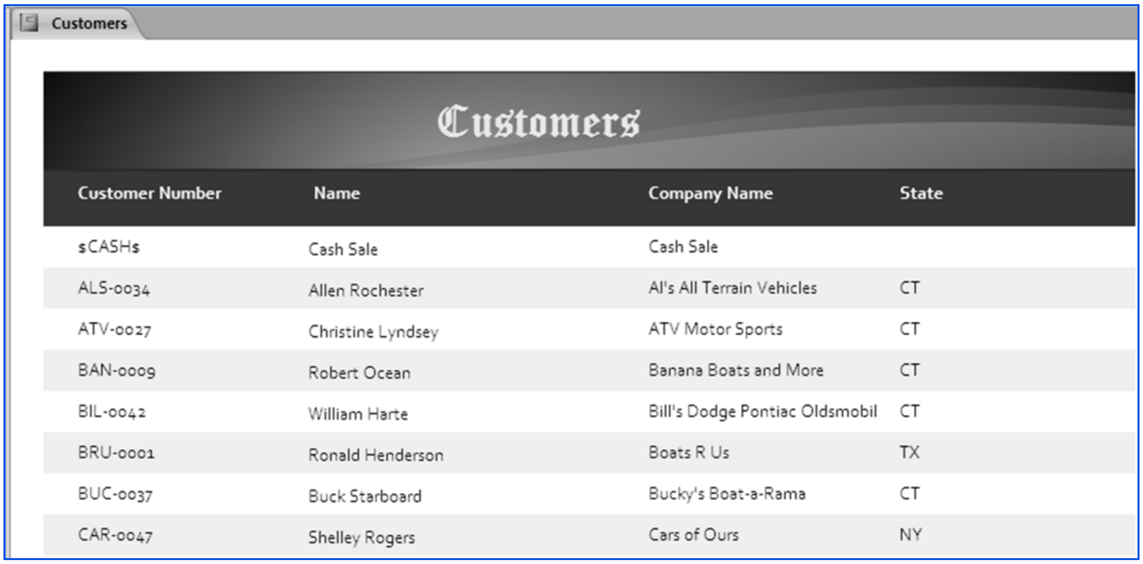


Fig. 2.4.12: Format the label

7. Select *Report View* from *Home ribbon* -> *Views* tab to open the report in *Report View*. The resultant report should appear, as in Fig. 2.4.13.



Customer Number	Name	Company Name	State
\$CASH\$	Cash Sale	Cash Sale	
ALS-0034	Allen Rochester	Al's All Terrain Vehicles	CT
ATV-0027	Christine Lyndsey	ATV Motor Sports	CT
BAN-0009	Robert Ocean	Banana Boats and More	CT
BIL-0042	William Harte	Bill's Dodge Pontiac Oldsmobil	CT
BRU-0001	Ronald Henderson	Boats R Us	TX
BUC-0037	Buck Starboard	Bucky's Boat-a-Rama	CT
CAR-0047	Shelley Rogers	Cars of Ours	NY

Fig. 2.4.13. Report view

2.4.5 Sub-Reports

Sub-Report is a report that is inserted in another report. A sub-report, a complete report in its own right, is inserted into another report, called the *Main Report*. Main Report can be either bound or unbound. A bound main report is based on a table or query and its sub-reports contain related information. An unbound main report is not based on a table or query, but can serve as a container for one or more sub-reports. A main report can include as many sub-reports as necessary. The sub-reports can be added to two hierarchy levels.

Sub-Reports are usually an extension of data in main report. For example, the main report can contain details about the sales in a year, while the sub-report can show data for sale of each item or charts and graphs summarising and illustrating the numbers in the main report. If a sub report is inserted in a bounded main report, it should contain some field to link to main report.

2.4.6 Problem Scenario

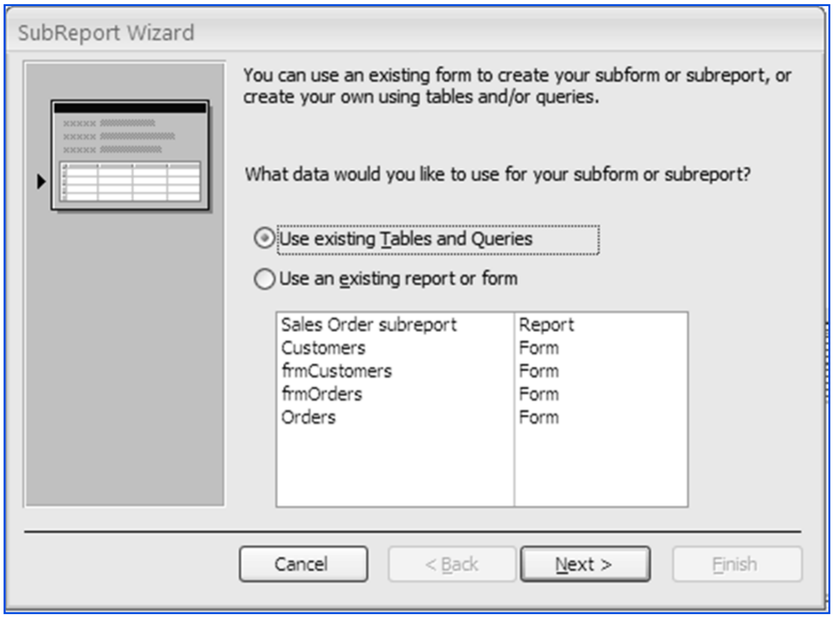
Consider the Report *rptCustomers*discussed in section 3.4.1. The Regional Head notices that the database users face a big problem while tracing the orders placed by each customer. He asks the Database Developer to create a user-friendly object which can display both the customer's details and orders placed by the customer.

Solution

The Database Developer decides to add a sub report to the report *rptCustomers* which include details of the orders placed by the customers. For this purpose, the table Sales Order is used and the fields *Sale to Customer from Sales Order* and *Customer Number from Customers* are mapped.

Steps to add a Sub Report to Main Report

1. Right-click the report *rptCustomers* and select *Design View* from the dropdown to open the report in *Design View*.
2. In the *Design View*, expand the *Details* section. Select the sub-report control from *Design ribbon -> Controls* tab and draw the control in *Details* section.
3. The *SubReport Wizard* appears, as shown in Fig. 2.4.14.



SubReport Wizard

You can use an existing form to create your subform or subreport, or create your own using tables and/or queries.

What data would you like to use for your subform or subreport?

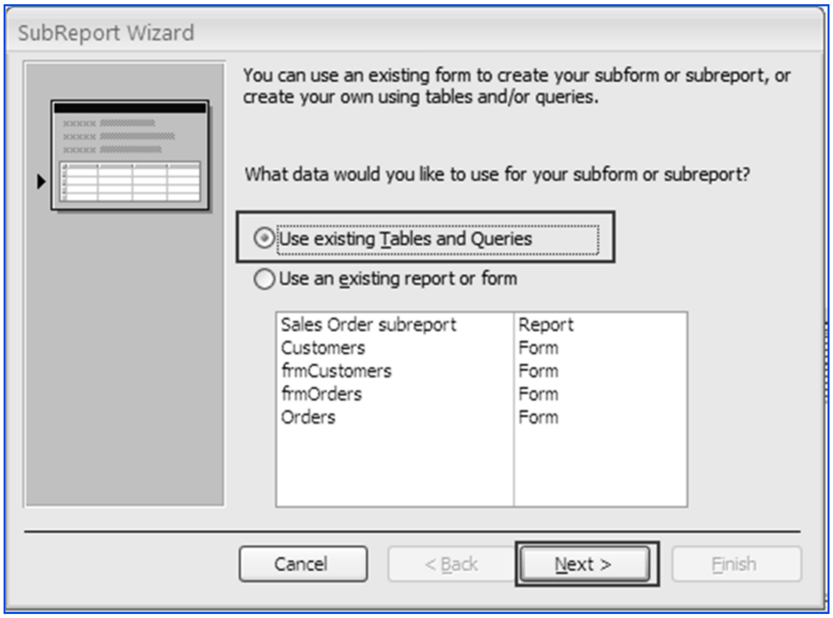
● Use existing Tables and Queries

○ Use an existing report or form

Sales Order subreport	Report
Customers	Form
frmCustomers	Form
frmOrders	Form
Orders	Form

Fig. 2.4.14. SubReport Wizard

4. Select the option *Use Existing Tables and Queries* and click *Next* to proceed further, as shown in Fig. 2.4.15.



SubReport Wizard

You can use an existing form to create your subform or subreport, or create your own using tables and/or queries.

What data would you like to use for your subform or subreport?

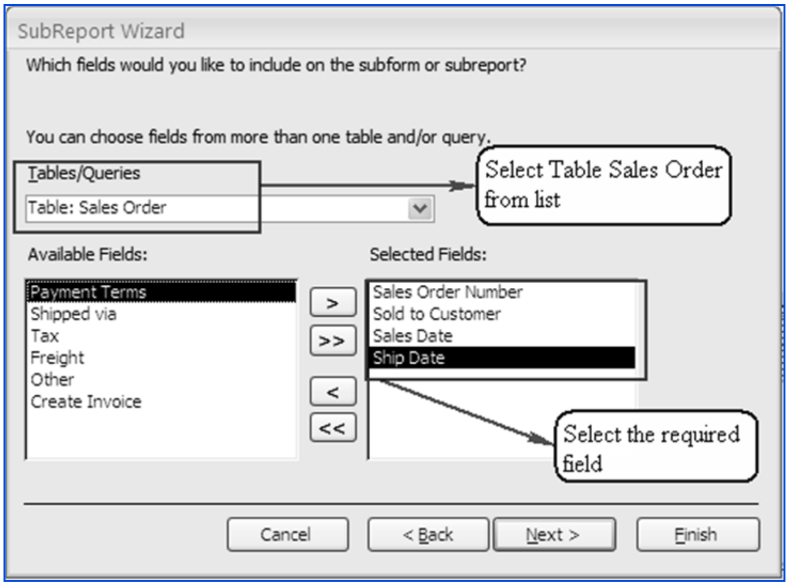
● Use existing Tables and Queries

○ Use an existing report or form

Sales Order subreport	Report
Customers	Form
frmCustomers	Form
frmOrders	Form
Orders	Form

Fig. 2.4.15. Select option to use existing table

5. In the *Select Table or Query* window, select the table *Sales Orders* from the list. Select the fields *Sales Order Number*, *Sale to Customer, Sales Date*, *Ship Date* from selected field to available field, as shown in Fig. 2.4.16.



SubReport Wizard

Which fields would you like to include on the subform or subreport?

You can choose fields from more than one table and/or query.

Tables/Queries: Table: Sales Order

Select Table Sales Order from list

Available Fields:

- Payment Terms
- Shipped via
- Tax
- Freight
- Other
- Create Invoice

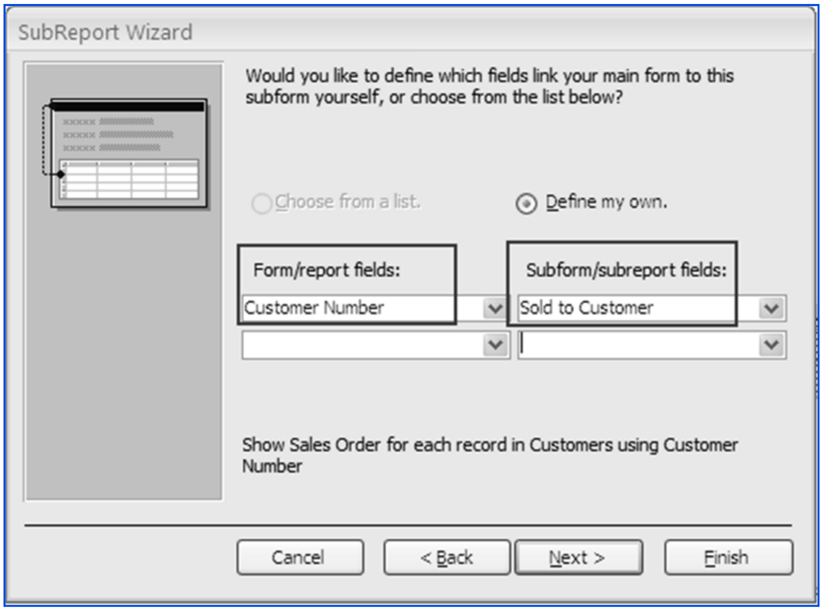
Selected Fields:

- Sales Order Number
- Sold to Customer
- Sales Date
- Ship Date

Select the required field

Fig. 2.4.16. Select required Table

6. Click *Next* to advance. The *Link Field* window appears. Select *Customer Number* from *Forms/Reports* field and *Sale to Customer from Subforms /Sub reports* field. The *Sub Report Wizard* appears, as shown in Fig. 2.4.17.



SubReport Wizard

Would you like to define which fields link your main form to this subform yourself, or choose from the list below?

○ Choose from a list. ● Define my own.

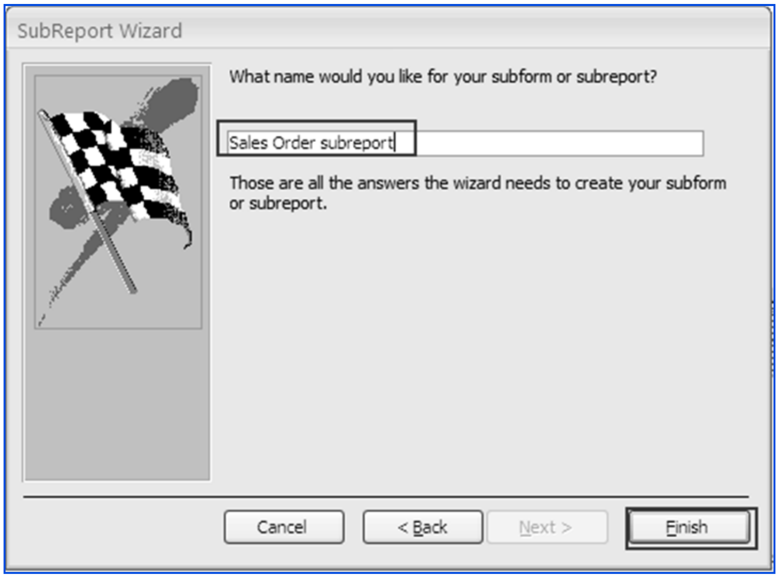
Form/report fields: Customer Number

Subform/subreport fields: Sold to Customer

Show Sales Order for each record in Customers using Customer Number

Fig. 2.4.17. Link fields of main report and sub report

7. Click *Next* to proceed to *Name of Sub report* window. Provide a suitable name to your sub report, and click *Finish* to close the window as indicated in Fig. 2.4.18.



SubReport Wizard

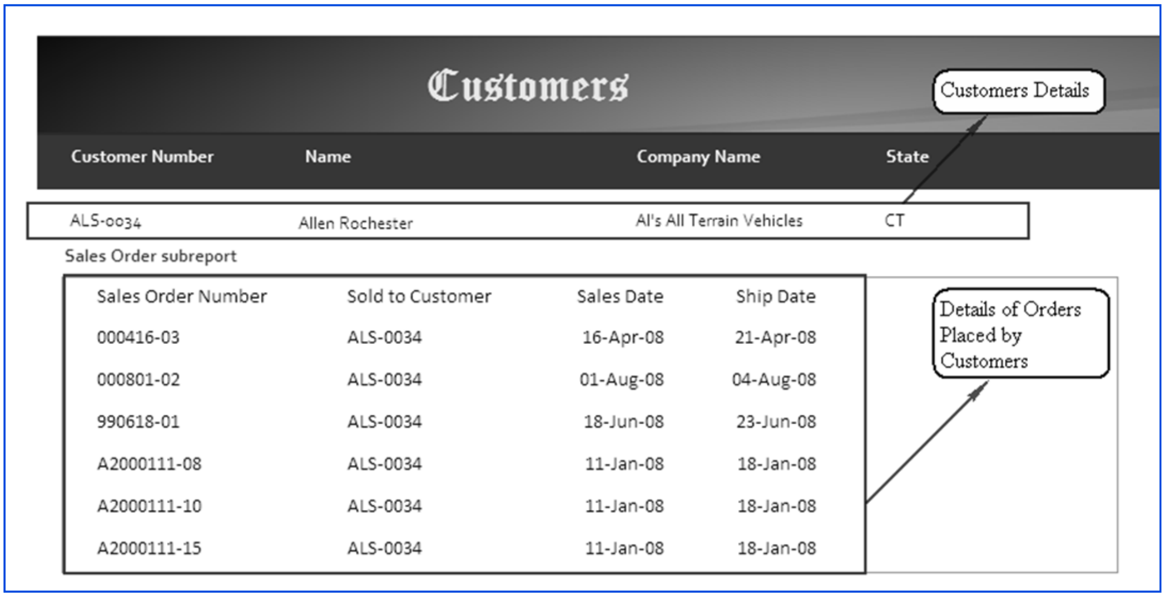
What name would you like for your subform or subreport?

Sales Order subreport

Those are all the answers the wizard needs to create your subform or subreport.

Fig. 2.4.18: Provide a name to the sub report.

8. Open the resultant report in *Report View* by selecting *Report View* from *Home ribbon* -> *Views* tab. Final report appears, as shown in Figure 2.4.19.



Customers

Customer Number	Name	Company Name	State
ALS-0034	Allen Rochester	Al's All Terrain Vehicles	CT

Sales Order subreport

Sales Order Number	Sold to Customer	Sales Date	Ship Date
000416-03	ALS-0034	16-Apr-08	21-Apr-08
000801-02	ALS-0034	01-Aug-08	04-Aug-08
990618-01	ALS-0034	18-Jun-08	23-Jun-08
A2000111-08	ALS-0034	11-Jan-08	18-Jan-08
A2000111-10	ALS-0034	11-Jan-08	18-Jan-08
A2000111-15	ALS-0034	11-Jan-08	18-Jan-08

Fig. 2.4.19. Customers and Orders placed by them

2.5 Make Reports more effective

Reports are a way of communicating database information. By customizing reports, the information can be presented in the most effective format. A customized report developed by using Access 2010 tools can reach a wider audience and enable more users to handle them. Different methods to make reports effective can be:

- including charts in a report
- printing data in columns
- canceling the printing of a blank report
- creating report snapshot

2.5.1 Including a Chart in a Report

Chart is a graphical representation of information used to illustrate quantitative relationships. It is a diagram that depicts a relationship, often functional, between two sets of numbers or between a set of numbers and a set of categories.

Microsoft Graph is used to chart data from any of the database tables or data stored within other applications. It creates graphs in a wide variety of styles, such as bar graphs, pie charts, line charts, and others. Because Microsoft Graph is an embedded OLE application, it does not work by itself. As such, it has to be run from within Access 2010. In other words, it is dependent on Access 2010.

Different Chart Types

Chart Type	Purpose
Column Chart	Used to compare multiple values of categories or differences over a period of time. The horizontal axis depicts categories and the vertical axis depicts values.
Bar Chart	Used for the same purposes as a Column Chart. However, the horizontal axis of a Bar Chart shows values and the vertical axis shows categories or periods of time.
Area Chart	Used to emphasize differences in individual values to the total, over a period of time.
Line Chart	Used to compare trends over a period of time.
Pie Chart	Used to show the relationship of a part to the whole. It is suitable for depicting one data series or data at a point in time.

Charts can be included in reports to illustrate the information more clearly. Charts enhance the data presented in reports by summarising the information and illustrating it in easily understandable ways. The reader can analyse trends and make comparisons using charting tools. The chart can be linked to a field in the underlying table or query.

2.5.2 Problem Scenario

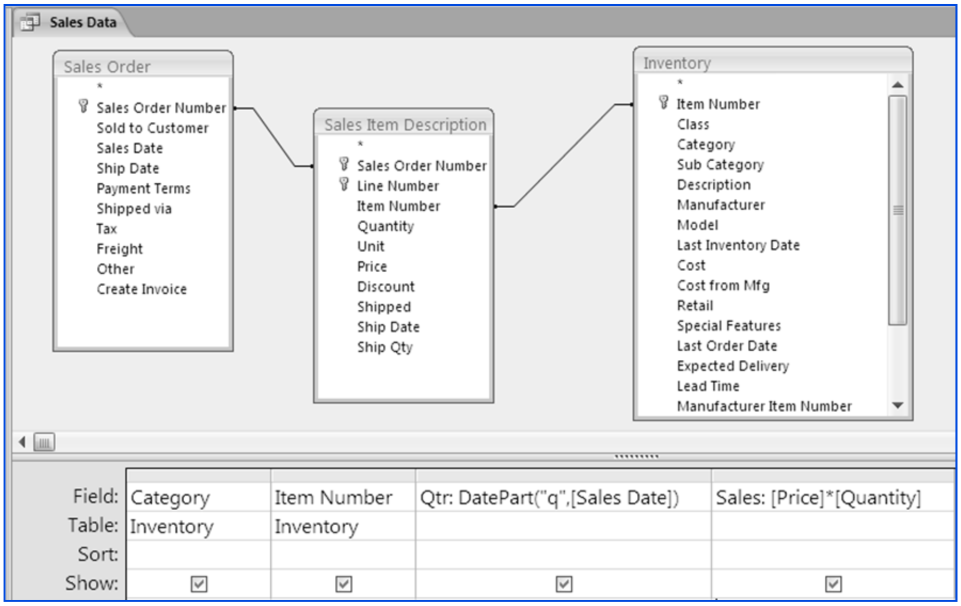
The Sales Head wishes to create a summary report. The report should show the quarterly total sales for each category. The Sales Head requests the Information Analyst to show a diagrammatic representation of the report.

Solution

The best way to display the summary report is to create charts to represent the data. The charts can be included in a report, and based on a query to show its diagrammatic representation.

Steps to Include Chart in Report

1. The Query Sales data represents the summarized data for per quarter sale. The design of the query is displayed in Fig. 2.5.1.



Sales Data

Sales Order
*
Sales Order Number
Sold to Customer
Sales Date
Ship Date
Payment Terms
Shipped via
Tax
Freight
Other
Create Invoice

Sales Item Description
*
Sales Order Number
Line Number
Item Number
Quantity
Unit
Price
Discount
Shipped
Ship Date
Ship Qty

Inventory
*
Item Number
Class
Category
Sub Category
Description
Manufacturer
Model
Last Inventory Date
Cost
Cost from Mfg
Retail
Special Features
Last Order Date
Expected Delivery
Lead Time
Manufacturer Item Number

Field:	Category	Item Number	Qtr: DatePart("q",[Sales Date])	Sales: [Price]*[Quantity]
Table:	Inventory	Inventory		
Sort:				
Show:	☑	☑	☑	☑

Fig. 2.5.1. Sales Data Query

2. Select *Report Design* from *Create ribbon* -> *Reports* tab. A blank report opens.

3. Select *Chart* control from *Design ribbon* -> *Controls* tab and draw it on the report. The *Chart Wizard* appears, as shown in Fig. 2.5.2.

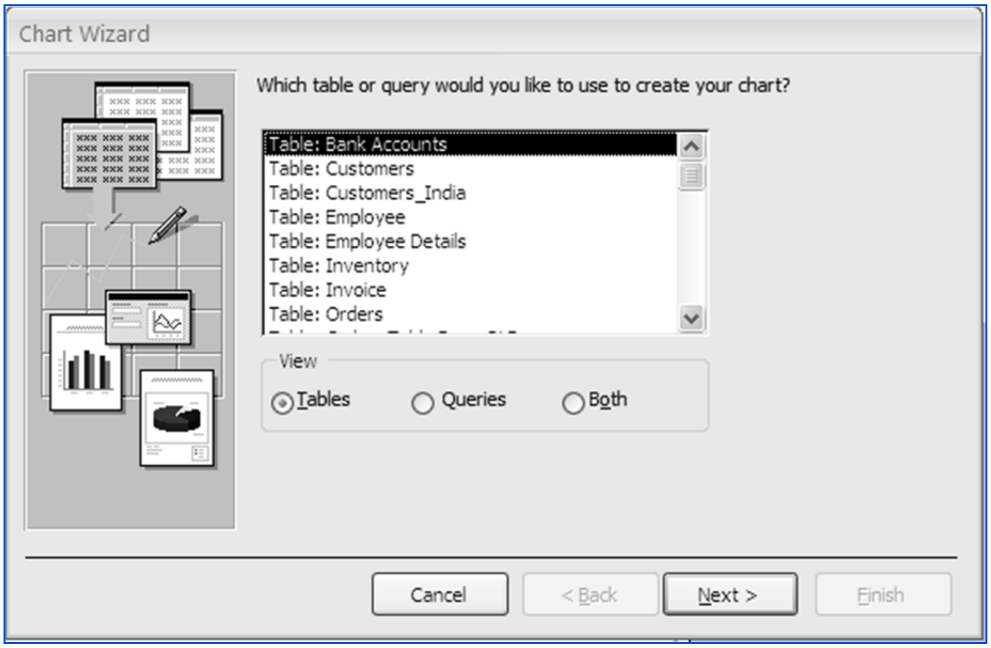


Chart Wizard

Which table or query would you like to use to create your chart?

Table: Bank Accounts
Table: Customers
Table: Customers_India
Table: Employee
Table: Employee Details
Table: Inventory
Table: Invoice
Table: Orders

View
● Tables ○ Queries ○ Both

Fig. 2.5.2. Chat Wizard

4. Select the *Queries* option, and select *Query: Sales Data* from the list, as indicated in Fig. 2.5.3.

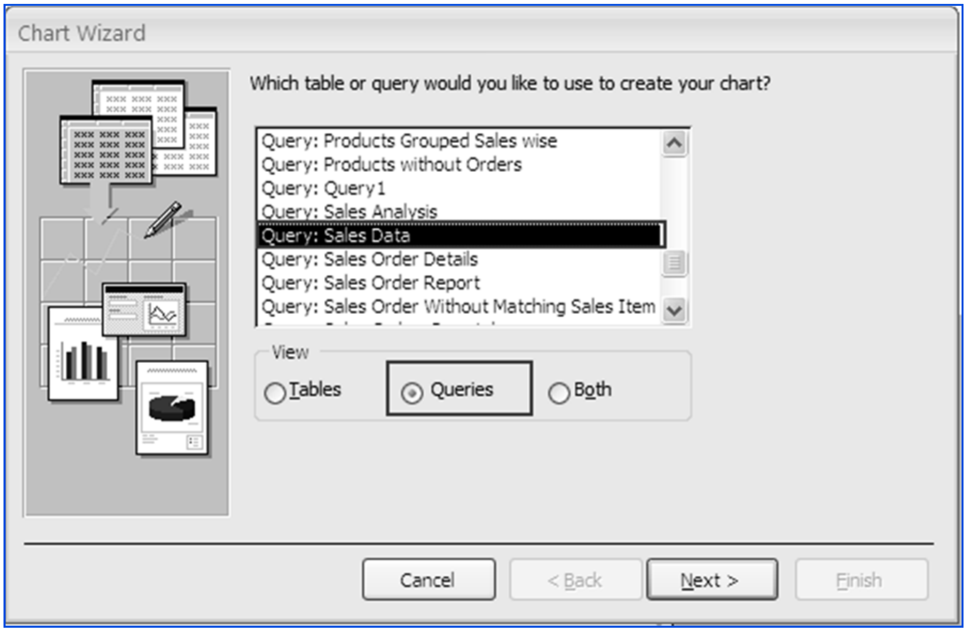


Chart Wizard

Which table or query would you like to use to create your chart?

- Query: Products Grouped Sales wise
- Query: Products without Orders
- Query: Query1
- Query: Sales Analysis
- Query: Sales Data
- Query: Sales Order Details
- Query: Sales Order Report
- Query: Sales Order Without Matching Sales Item

View: ○ Tables [✓ ● Queries] ○ Both

Fig. 2.5.3. Select Queries Sales Data

5. Click *Next* to advance. The *Select Field* window appears. Select all the fields: *Category*, *Qtr*, and *Sales*, as shown in Fig. 2.5.4.

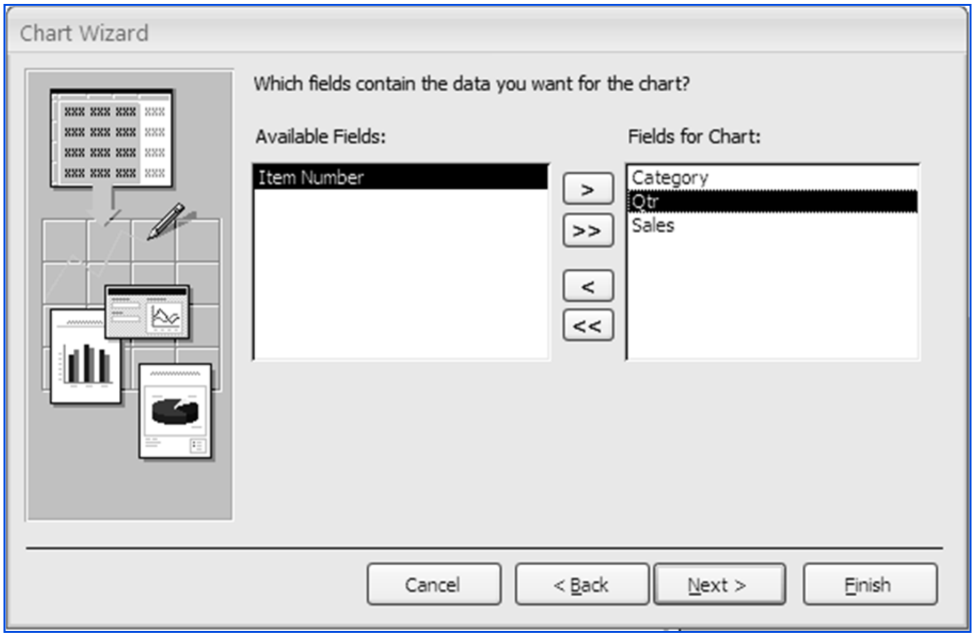


Chart Wizard

Which fields contain the data you want for the chart?

Available Fields:

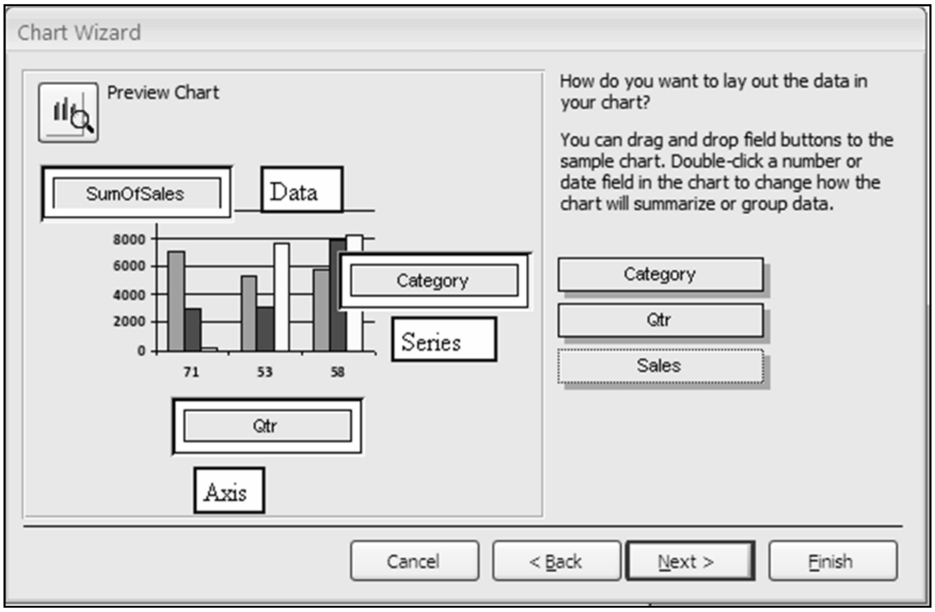
- Item Number

Fields for Chart:

- Category
- Qtr
- Sales

Fig. 2.5.4. Select required fields

6. Click *Next* to proceed. In the *Choose Chart Type,* select the *Column Chart* and click *Next*.
7. In *Preview Chart* window drag field *Qtr* to *Axis*, *Category* to *Series* and *Sales* to *Data* as shown in Fig. 2.5.5. Click *Next* to advance.



How do you want to lay out the data in your chart?

You can drag and drop field buttons to the sample chart. Double-click a number or date field in the chart to change how the chart will summarize or group data.

Fig. 2.5.5. Drag the fields to chart

8. Specify the title for the chart and select the *Display Legend* option. Click *Finish* to close the window.
9. The resultant chart appears, as shown in Fig. 2.5.6.

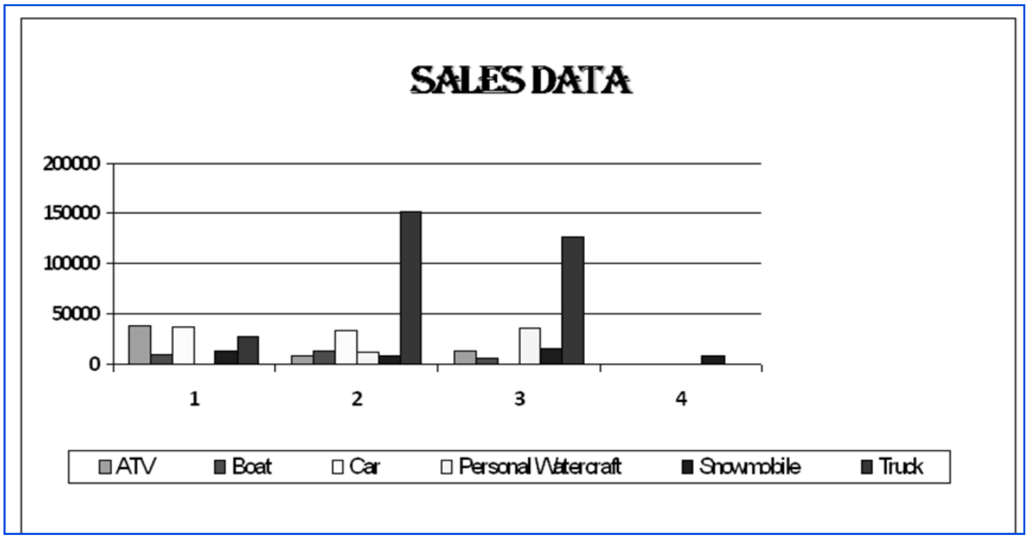


Fig. 2.5.6. The Sales Data Report

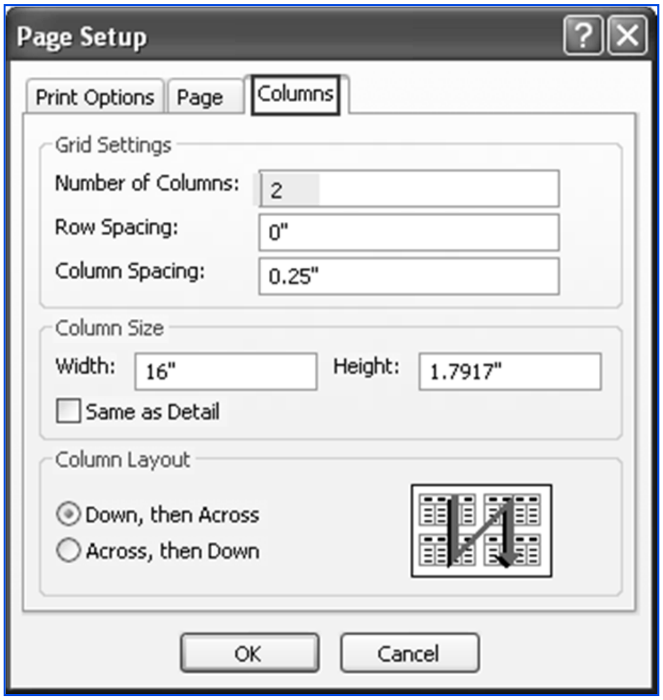
NOTE: The chart object can be modified by right-clicking and selecting *Chart Object -> Edit* from the dropdown. Also, the chart object can be embedded with existing data in the report so that the chart changes with each record display.

2.5.3 Printing Data in Columns

Reports are a handy tool to represent data. The visibility of the reports makes an impact on how data can be viewed. Some of the reports may involve long lists of just a few fields of data—such as a phone or product list. These may be best arranged in multiple columns for better readability.

While working with many databases, it is natural to come across some that consists of long lists of information. As a result, printing such data can mean printing a single column on many pages.

The multiple columns for the report can be set using the *Report Page Setup* property and setting the number of columns to 2 or more, as shown in Fig. 2.5.7.



Number of Columns: 2
Row Spacing: 0"
Column Spacing: 0.25"
Width: 16"
Height: 1.7917"
□ Same as Detail
● Down, then Across
○ Across, then Down

Fig. 2.5.7. Select multiple columns from Page Setup

2.5.4 Cancelling the Printing of a Blank Report

If a report contains no records, the detail area of the report will be blank. While printing reports, it is not be advisable to print blank report. Macros can be used to cancel printing of a blank report and thus save time and effort. Depending on the availability of data, certain records may be absent in a report and printing a blank report would be meaningless.

To cancel previewing or printing of a blank report when the underlying query has been run with no records returned, the *On No Data* event of the report is used.

2.5.5 Problem scenario

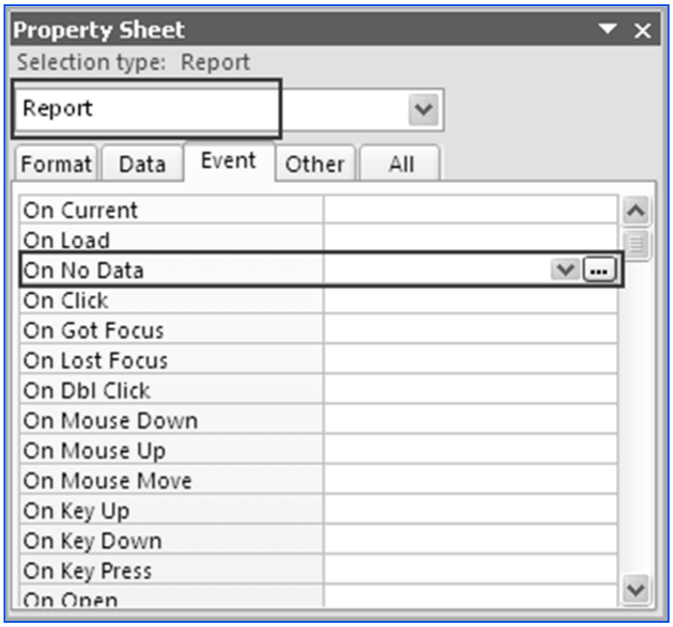
The Database Developer has prepared a report CustOrders, which accepts the customer'sfirst name and displays all orders placed by the customer in the current quarter. However, most users while printing the report found that the even when the query returns no results, the report is printed. They requested the Database Developer to resolve the problem.

Solution

The report event *On No Data* can be used to cancel the printing of report when no data is returned.

Steps for Canceling the Print

1. Open the *CustOrders Report* in *Design View*.
2. Press *F4* to open the property sheet of report. Make sure that the report is selected in the *Selection Type* textbox. On the *Event* tab, select the property *On No Data*, as indicated in Fig. 2.5.8.



Property Sheet

Selection type: Report

Report

Format | Data | Event | Other | All

On Current
On Load
On No Data
On Click
On Got Focus
On Lost Focus
On Dbl Click
On Mouse Down
On Mouse Up
On Mouse Move
On Key Up
On Key Down
On Key Press
On Open

Fig. 2.5.8. Property Sheet of Report

3. Select ... button of *On No Data* property.*Choose Builder* window appears. Select *Macro Builder* and click *OK* as shown in Fig. 2.5.9.

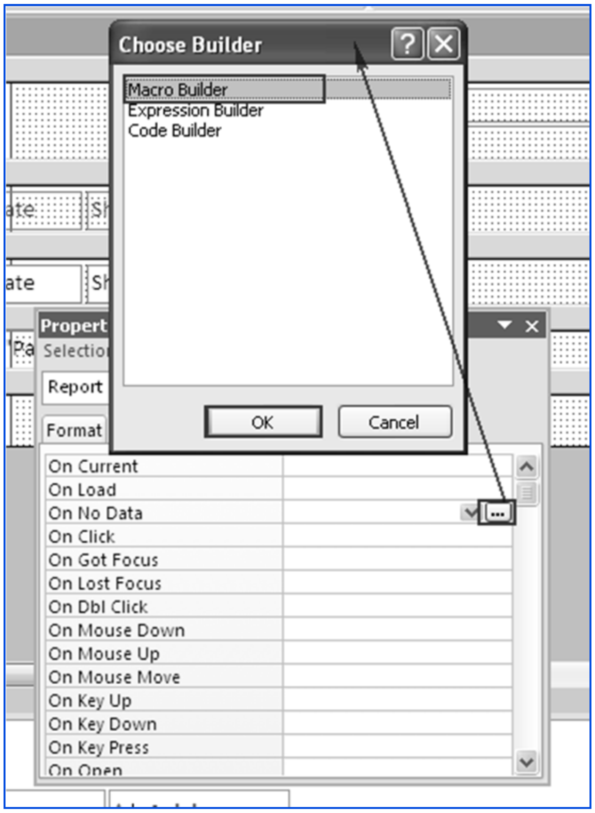
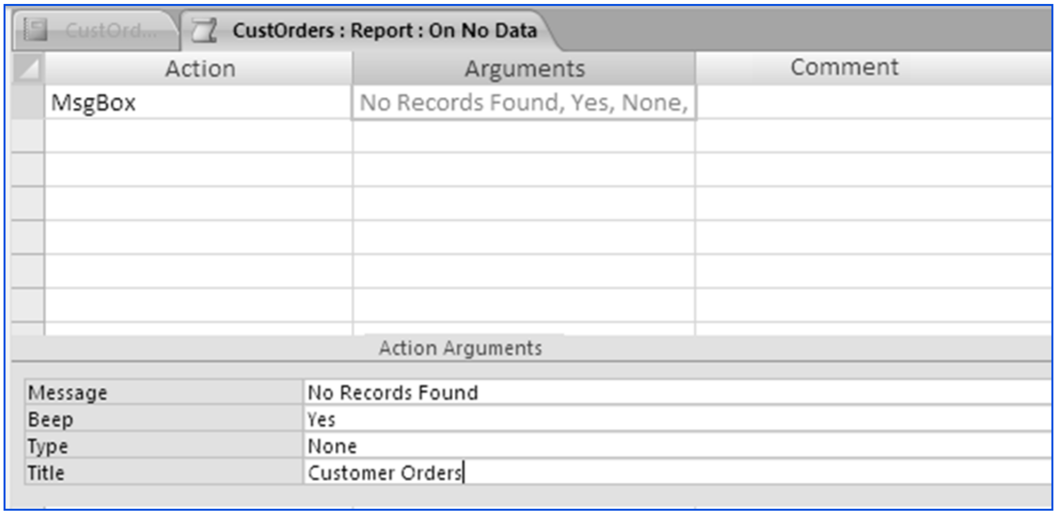


Fig. 2.5.9. Open Macro Builder

4. In the *Macro Builder* window, select *Action MsgBox* and specify the value of message *Arguments* as *No Records Found*, *Title* as *Customers Orders*, as indicated in Fig. 2.5.10.



CustOrders : Report : On No Data

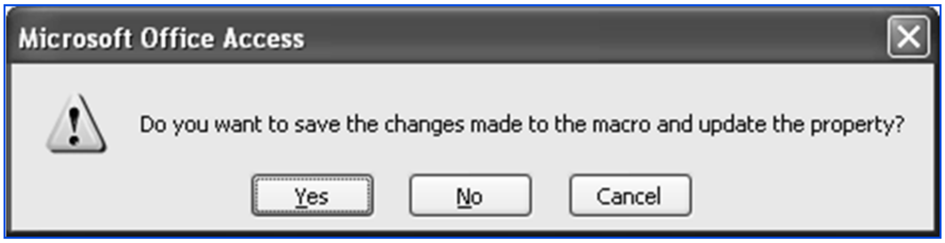
Action	Arguments	Comment
MsgBox	No Records Found, Yes, None,	

Action Arguments

Message	No Records Found
Beep	Yes
Type	None
Title	Customer Orders

Fig. 2.5.10. Specify MsgBox Action

4. Specify the second action as *CancelEvent* and click the *Close* button to close the *Macro* window. The confirmation message appears, as indicated in Fig. 2.5.11.


Microsoft Office Access

Do you want to save the changes made to the macro and update the property?

Fig. 2.5.11. Confirm to close the macro

5. Click *Yes* to save the macro and close the message window. The macro now appears on *No Data* event.
6. Open the report and verify that the macro is working.

2.6 Summary

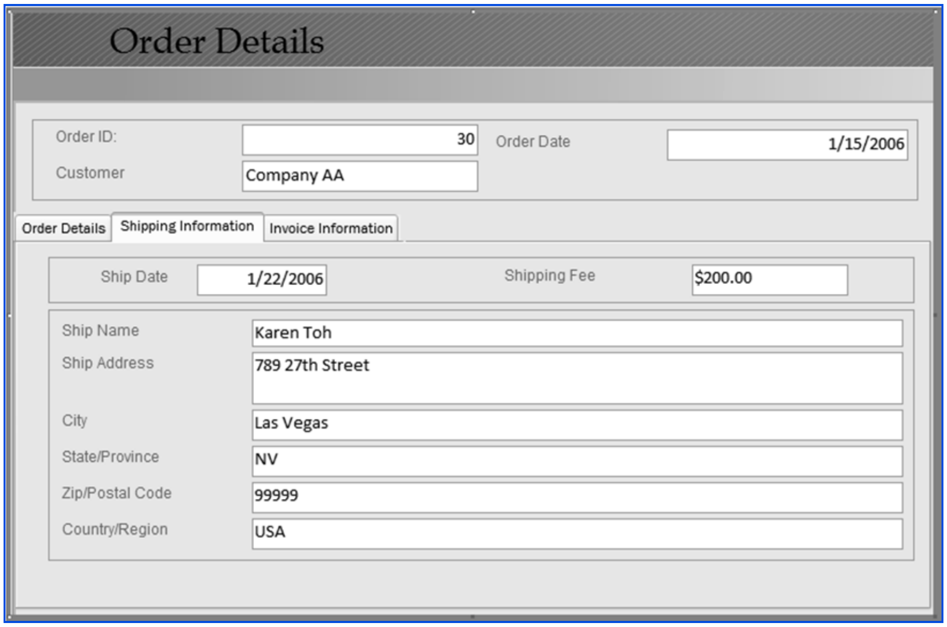
Forms and Reports are a crucial part of data representation in Access 2010. Access 2010 provides various utilities to make Forms and Reports more user-friendly and presentable. Different controls can be added to forms to display the logo of the company, date and time, and pictures. Forms also permit the inclusion calculated values and combo boxes which make it much easier to handle the controls and present data to user. This control helps make data handling easier for the user. Access 2010 provides various ActiveX like calendar control to make forms more interactive. Sometimes, it is required to display the information on a form in groups. The tab control available with Access 2010 can be used for this purpose. The popular utility for summarizing data of Pivot Table to summarize data is also available in Access 2010.

Reports represent static data, but are a useful way of communicating. Reports in Access 2010 can be customized to user's requirements including its header and footers. Another crucial feature available with reports is sub-reports, which enable data linking in several tables. Apart from this, we can add charts and calculated controls to reports to make them more visible. The properties of reports can be used to avoid printing of blank reports.

2.7 Lab Exercises

Considering the Apex Inventory Shipment database of Apex Ltd. Provide a solution to the following problem scenarios:

1. The Sales Manager of the company has demanded a form displaying the details of the orders, containing with Items ordered, Shipping Details and Invoice Information. Design the form Order Details as displayed in Figure below. Create tabs to represent the data.



Order Details

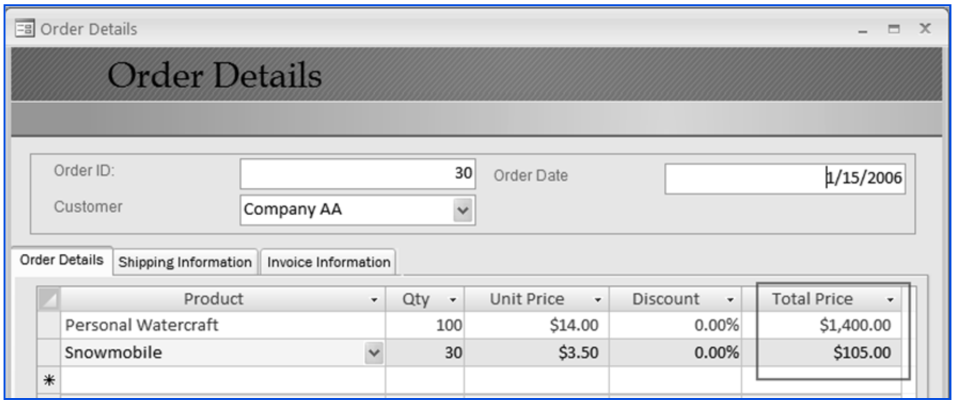
Order ID:	30	Order Date	1/15/2006
Customer	Company AA		

Ship Date	1/22/2006	Shipping Fee	$200.00

Ship Name	Karen Toh
Ship Address	789 27th Street
City	Las Vegas
State/Province	NV
Zip/Postal Code	99999
Country/Region	USA

On the Order Details Form created in Question1 implement the Question 2 to 7:

2. In the Order Details form, convert the Customer Textbox to Combo Box which contains Customer Number and Company Name.
3. To make the Order Details form more user-friendly, add a calendar control to the form to select the Order Date.
4. The Sales Manager asked the developer to add a logo of the company the Order Details form as the form will be circulated among all divisions.
5. The database developer needs to make the form more presentable and has to add a picture on the background of each tab. Implement the needful.
6. In the Order Details tab of the form, add a calculated value that calculates the Total Price as Qty * Unit Price – Discount as shown in Figure below:



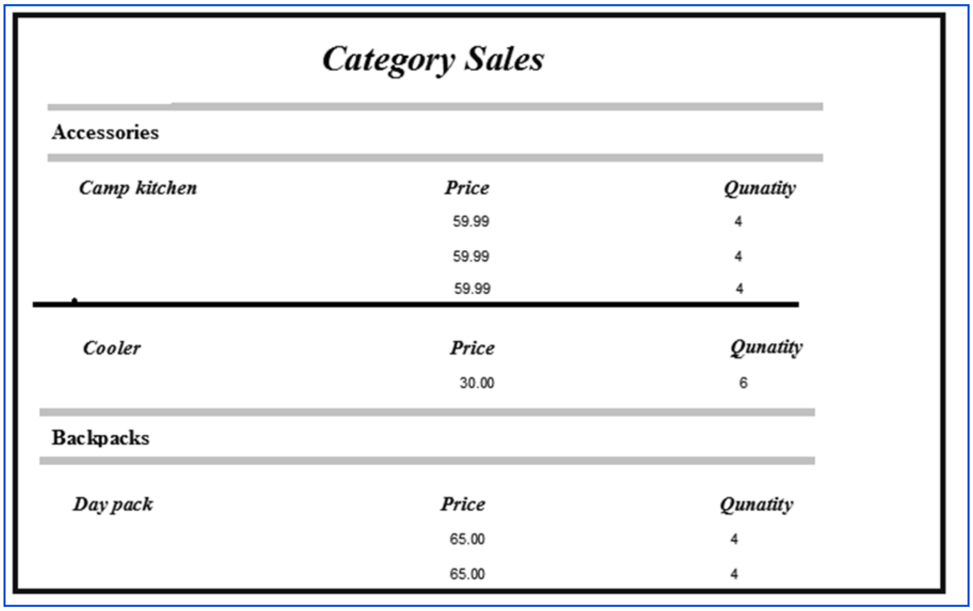
Order Details

Order ID:	30	Order Date	1/15/2006
Customer	Company AA		

Product	Qty	Unit Price	Discount	Total Price
Personal Watercraft	100	$14.00	0.00%	$1,400.00
Snowmobile	30	$3.50	0.00%	$105.00
*				

7. The Sales Manager of the Company wishes to launch some sales promotional offers. He requested the database developer to create a form which should display the summarized count and cost of Orders placed by each customer on Quarter and month basis. He should also be able to view the same results for a particular category. Create a form containing Summary of Data.

   The developer of Apex Inventory Shipment database needs to create Item Sales report. The report should be grouped by Category and then Item and should display quantity and price of Item sold. The report is displayed in Figure below:



Category Sales

Accessories

Camp kitchen	*Price*	*Qunatity*
	59.99	4
	59.99	4
	59.99	4

Cooler	*Price*	*Qunatity*
	30.00	6

Backpacks

Day pack	*Price*	*Qunatity*
	65.00	4
	65.00	4

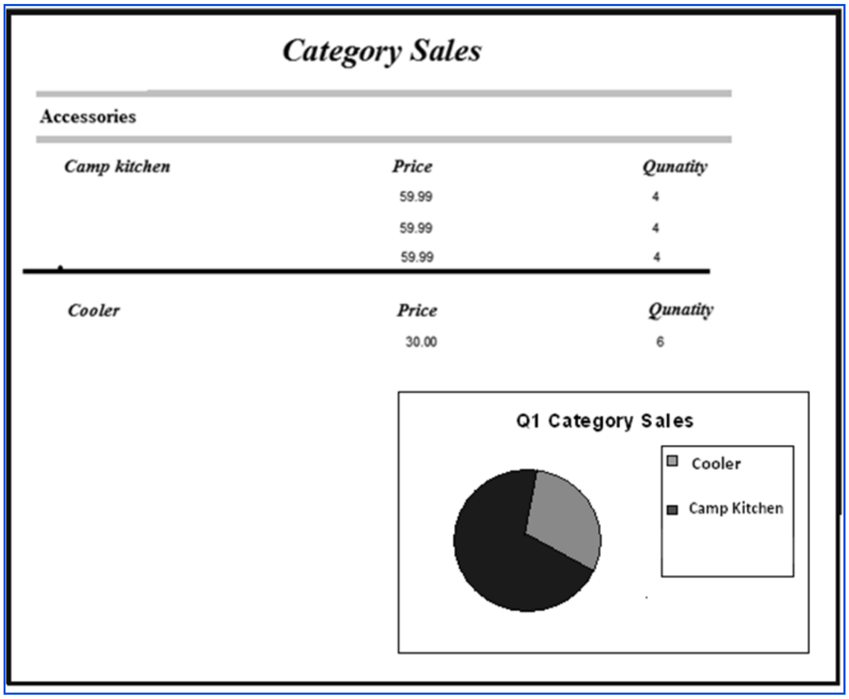
Implement the Questions 8 – 14 based on the report Item Sales.

8. Add a calculated control *Total Sales* on the report that should display the Total Sales as Price * Quantity.

9. Add a Page Header and Footer on the Report. The header must contain the logo of the company and Footer must contain the current Quarter and Month.

10. Compute the Total Sales for each product in group footer.

11. Include a pie chart in the Category group footer that should display the ratio of sales of each product in that category. The report should look like as in Figure below:



Category Sales

Accessories

Camp kitchen	*Price*	*Qunatity*
	59.99	4
	59.99	4
	59.99	4

Cooler	*Price*	*Qunatity*
	30.00	6

12. The details of the Sales of Item were to be presented in the Monthly meeting as a hardcopy. So the Manager asked the executive to get the report into columnar format so that is more readable. Remove the Page Headers and the chart, and print the report in two columns.

13. For the Annual review of the sales, the Vice President demanded a report displaying the product sales by month. The reports should display the products from a particular category, which is given by the user. The report should represent the data diagrammatically using charts. Create the required Report.

14. The Sales Manager found that there are few categories which are not produced any more by the company but are not discarded in the report. As a result if such category is entered, a blank report gets printed. As a report administrator, cancel the printing of the blank report.

Multiple Choice Questions

1. The controls that are not linked to any field of the table or query on the form are known as ________.

   (a) ActiveX control

   (b) Unbounded Controls

   (c) Graphics Control

   (d) Bound Controls

2. The Vice President of the Company wishes to add the image of the company vision statement as a background of all the forms. Which control can be used for the purpose?

   (a) Calendar Control

(b) ActiveX control

(c) Image Control

(d) Graphics Control

3. Which of the Header in reports can be viewed separately from the Footer?

(a) Page Header

(b) Group Header

(c) Report Header

(d) All of the above

4. The Manager requested the developer to create a Inventory form in a manner such that the Product's description is stored in one group, the cost and supplier information another group. Which is the best control to display the required information?

(a) Use the Tab Control

(b) Use SubForms

(c) Add ActiveX Control

(d) None of the above

5. Which property of the form can be used to insert a background image?

(a) Caption

(b) Background

(c) Record Source

(d) Picture

6. The Sales Executive while filling the details of the orders placed find it very difficult to type the name of each product every time it is ordered. They demanded that the form should provide them a drop down to select the product to be ordered. How can we implement the required?

(a) Using the query in the form

(b) Using the Combo Box control

(c) Using a SubForm

(d) Cannot be done

7. The Calendar control can be selected from _______________

(a) Microsoft Office-> Access Options

(b) Design -> Controls

(c) Design -> Controls -> ActiveX controls

(d) Create -> Forms

8. The Regional Sales Head demanded a summary report indicating the monthly sales done by each employee in each zone. Which kind of form is best to display the required data?

   (a) Use tabbed browsing

   (b) Insert an Image in form

   (c) Insert SubForm

   (d) Create Pivot Table

9. To print the multi-columnar report, the number of columns can be set through _________ property.

   (a) Report property sheet

   (b) Page Setup

   (c) Report wizard

   (d) Grouping

10. The persons from the delivery team found it very annoying that even when the reports contained no data, they are printed and they have to search for such reports among all the printed data. They requested the developer to find the solution to this problem. Which property of report can be used to implement the requirement?

    (a) CancelPrint

    (b) CancelEvent

    (c) Create a macro called On No Data

    (d) Create a macro called On Print

11. To display the data of the Products and the orders placed for each product in the current month. The following feature available in Reports can be used.

    (a) Report Wizard

    (b) Nested Reports

    (c) Grouping

    (d) SubReports

12. The Regional Head wishes to view the diagrammatic representation of data indicating the sales made by each zone in his region. Which feature can help to implement the requirement?

    (a) Graphics

    (b) ActiveX

    (c) Charts

    (d) Pivot Table

13. To display the list of employees grouped according to first letter of their name, which type of controls can be used in reports?

(a) Use Calculated values in group

(b) Add grouping control

(c) Add Function control

(d) Add ActiveX control

14. Which property of the control is used to bind it to a field of a table or a query?

(a) Data

(b) Record Source

(c) Field

(d) None of the above

15. Each tab in a tab control is known as _________.

(a) Page

(b) Data Tab

(c) Control Page

(d) Control

16. The Sales Manager requested a summary form which should enable him to choose the category and display the monthly sale of each product in the category. In the created Pivot Table Form, The category field should be placed in which area?

(a) Drop Row Fields Here

(b) Drop Column Fields Here

(c) Drop Totals or Detail Fields Here

(d) Drop Filter Field Here

17. A developer created a report displaying the information of customer grouped according to country and state. To add a count of customer in each state the count textbox should be placed in which section of the report?

(a) Page Footer

(b) Report Footer

(c) State Group Footer

(d) Country Group Footer

18. The SubReport in the main report can be inserted to _______________ hierarchy level?

(a) 7

(b) 3

(c) 2

(d) 4

CHAPTER

3

BUILDING CRITERIA EXPRESSIONS

LEARNING OBJECTIVES

- Using operands in Criteria Expressions
- Using built-in functions
- Working with Expression Builder

3.1 Introduction

Expressions in Microsoft Access 2010 can be considered similar to formulae in Microsoft Excel. Expressions are a combination of operands, operators, functions, and values that are evaluated according to their order of precedence. Expressions can be used with tables, queries, forms, reports, and macros. In Access, expressions are used to obtain calculated values, provide criteria, and query or supply constraint to table columns. Access also provides a powerful user interactive graphical tool to create expressions known as *Expression Builder*.

In this chapter, we will discuss how to build criteria expressions in Access. We will also look at using various components of an expression for building query criteria. This chapter will also identify various operators available in Access 2010. Next, we will discuss the available built-in functions in Access. We will discuss the different type of functions and their utilisation. Further, this chapter will cover how to use *Expression Builder* to create expressions using Objects, Functions, Operators, and Identifiers.

3.2 Using Operands in Criteria Expressions

Query criteria are the most important part of any query as they permit users to select only the desired records from an existing table. An operand is a value on which a calculation is performed. In other words, an operand is a data value that gets manipulated in the query expression. Operands can be *literals*, *identifiers*, or *functions*.

3.2.1 Literals

A literal is value that is not addressed by any name. It can be typed directly into the criteria expression. In Access, literal can be of type *number, text, date,* or *logical* value (i.e. True or False). Literals are also referred as constants as their values remain static throughout the evaluation of expression.

Examples of literal:

"Hello" + " " + "Everyone" , Here Hello and Everyone are Text literals

[Date] > #1/1/2011#, Here 1/1/2011 (1-Jan-2011) is a Date literal

3.2.2 Identifiers

Identifiers are variables. In Access, identifiers represent field name, table name, or control name. Identifiers are a crucial part of expression building as they specify the column to which an expression represents. While creating an expression in Access, identifiers are always represented in square brackets [ ].

Examples of identifiers used in an expression:

[Basic Salary] + [Tax]: Where Basic Salary and Tax are identifiers

3.2.3 Functions

Functions provide specialised operations to enhance the working of Access. Functions are built-in expressions that take an input, perform necessary calculations on it, and return the output. The input accepted by the function is called arguments; a function may have one or more number of arguments. Access provides us different functions to work with different type of data, such as Text functions, Date and Time functions, Numeric functions, and Mathematical functions.

For example, a text function Length takes an input string as an argument and returns the length of the string in number as:

Length (“MS Access”) will return 9.

3.2.3.1 Problem Scenario

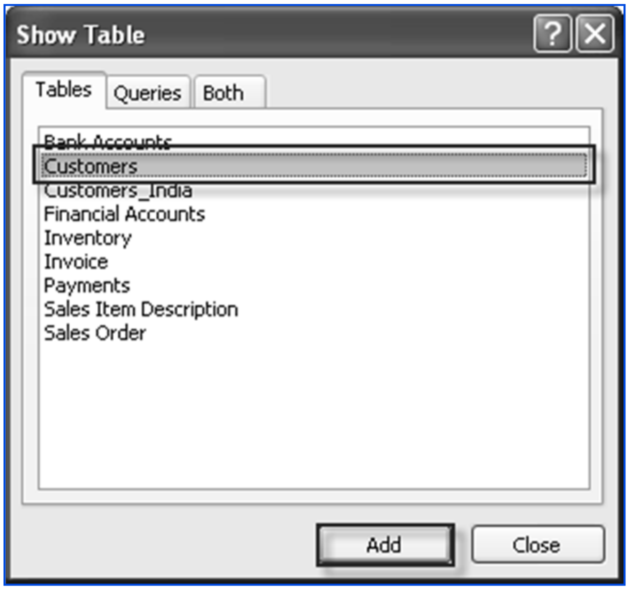
Apex Ltd. is launching a new production unit in “California”, which will also focus on some new products. To promote these products a detailed list of all existing customers from “California” (state code CA) is required.

Solution

To achieve this, a query displaying *Name*, *Contact Info*, and *Address* of the customers is required. To get only the customers from “California”, the value CA in the criteria for the state field needs to be specified. Here, CA is a literal operand and is typed direct in query criteria.

Steps for creating required query

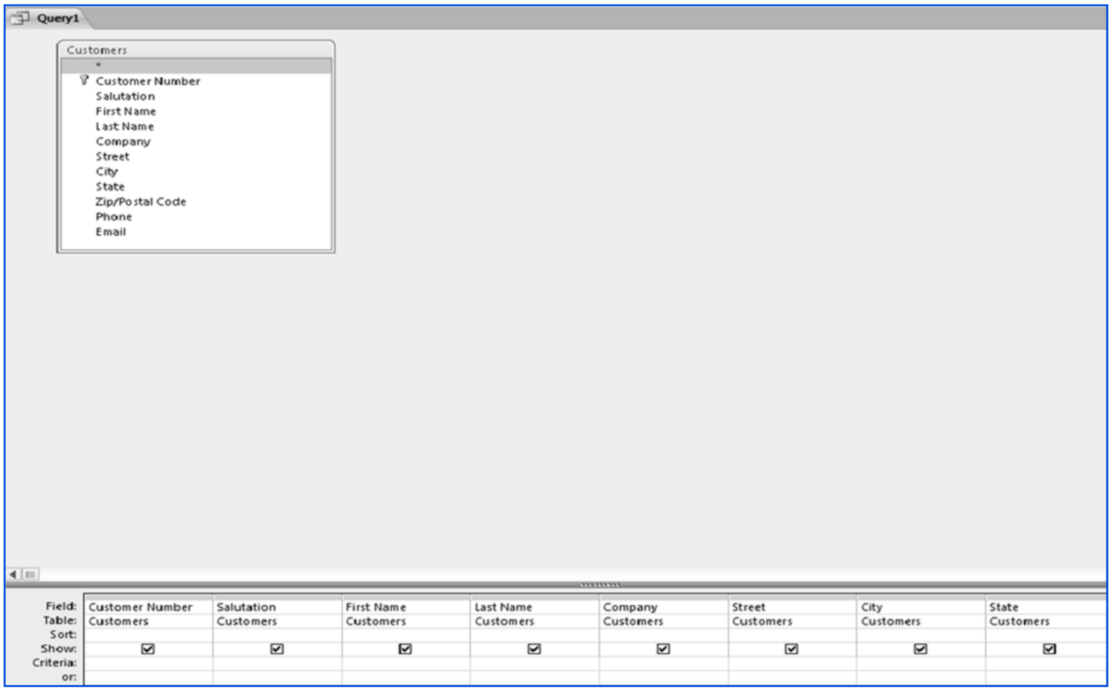
1. Click *Create ribbon* -> *Other* -> *Query Design* to open the *Query Design* window.
2. Select *Customers* from *Show Table* and click *Add*, as displayed in Fig. 3.2.1.



Bank Accounts
Customers
Customers_India
Financial Accounts
Inventory
Invoice
Payments
Sales Item Description
Sales Order

Fig. 3.2.1. Show Table window

3. Select the columns *Salutation*, *First Name, Middle Name, Last Name, Company, Street City*, *State*, *Zip/Postal Code Phone*, and *Email* from table *Customers* and drag them to the columns tab. The *Query* window appears, as shown in Fig. 3.2.2.



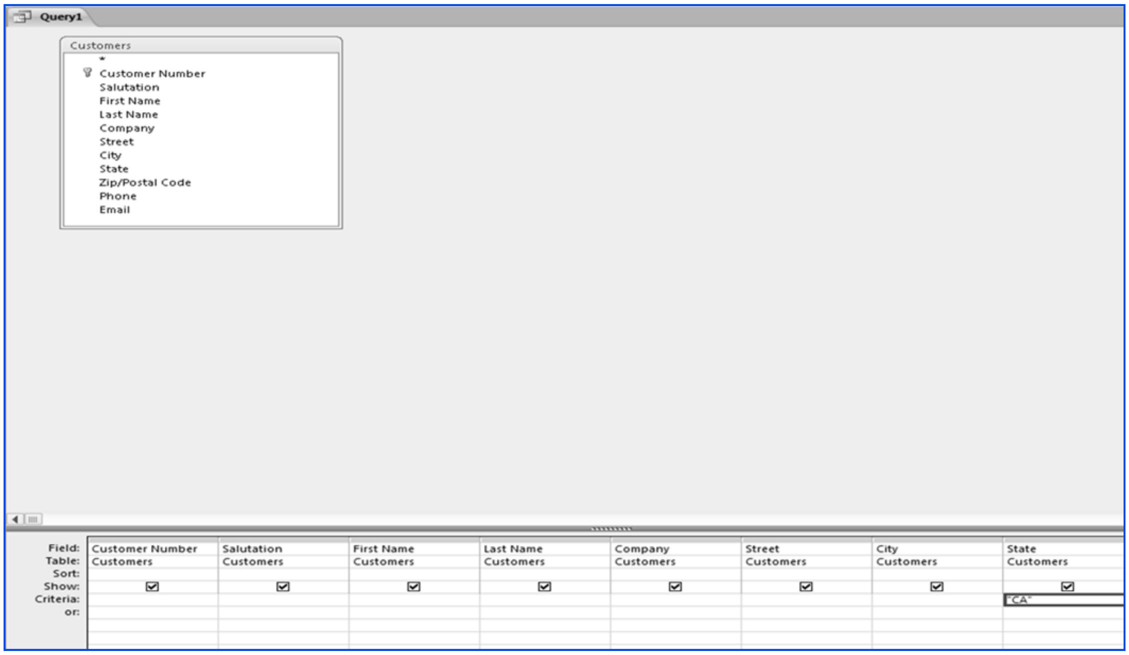
*
Customer Number
Salutation
First Name
Last Name
Company
Street
City
State
Zip/Postal Code
Phone
Email

Field:	Customer Number	Salutation	First Name	Last Name	Company	Street	City	State
Table:	Customers	Customers	Customers	Customers	Customers	Customers	Customers	Customers
Sort:								
Show:	☑	☑	☑	☑	☑	☑	☑	☑
Criteria:								
or:								

Fig. 3.2.2. Select Required Columns

Now, we will add criteria to get records from “California”.

4. In the *Criteria* tab of field *State*, write the literal “CA”, as indicated in Fig. 3.2.3.



Query1

Customers
*
Customer Number
Salutation
First Name
Last Name
Company
Street
City
State
Zip/Postal Code
Phone
Email

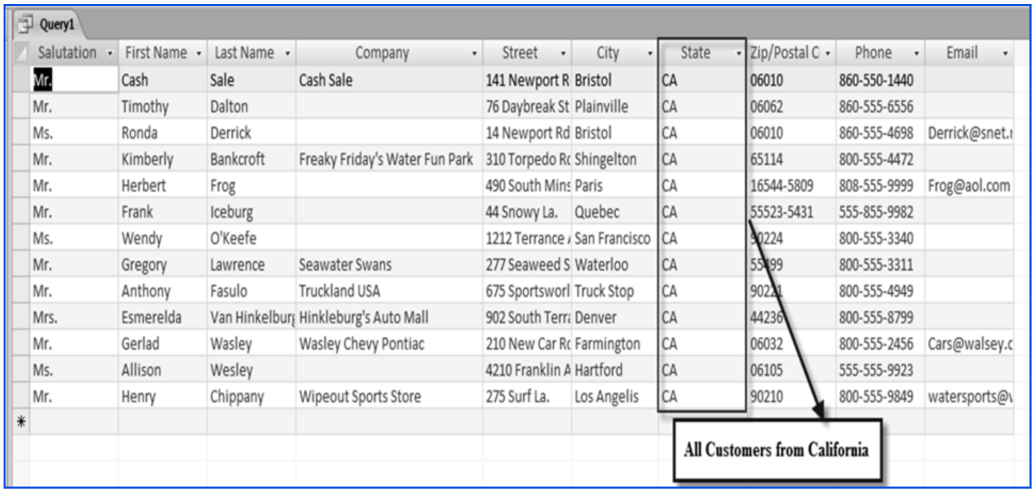
Field:	Customer Number	Salutation	First Name	Last Name	Company	Street	City	State
Table:	Customers	Customers	Customers	Customers	Customers	Customers	Customers	Customers
Sort:								
Show:	☑	☑	☑	☑	☑	☑	☑	☑
Criteria:								"CA"
or:								

Fig. 3.2.3. Specify criteria for State

5. Click *Save* in the toolbar to save the query. Type the name of query as “Customers from California”.

6. Click the *Run* sign in *Design ribbon* -> *Results* to view the results of the query, as shown in *Fig. 3.2.4.*

NOTE: Literals are usually combined with operators to form complex expressions for query criteria.



Query1

Salutation	First Name	Last Name	Company	Street	City	State	Zip/Postal C	Phone	Email
Mr.	Cash	Sale	Cash Sale	141 Newport R	Bristol	CA	06010	860-550-1440	
Mr.	Timothy	Dalton		76 Daybreak St	Plainville	CA	06062	860-555-6556	
Ms.	Ronda	Derrick		14 Newport Rd	Bristol	CA	06010	860-555-4698	Derrick@snet.r
Mr.	Kimberly	Bankcroft	Freaky Friday's Water Fun Park	310 Torpedo Rc	Shingelton	CA	65114	800-555-4472	
Mr.	Herbert	Frog		490 South Mins	Paris	CA	16544-5809	808-555-9999	Frog@aol.com
Mr.	Frank	Iceburg		44 Snowy La.	Quebec	CA	55523-5431	555-855-9982	
Ms.	Wendy	O'Keefe		1212 Terrance /	San Francisco	CA	90224	800-555-3340	
Mr.	Gregory	Lawrence	Seawater Swans	277 Seaweed S	Waterloo	CA	55[illegible]99	800-555-3311	
Mr.	Anthony	Fasulo	Truckland USA	675 Sportsworl	Truck Stop	CA	90221	800-555-4949	
Mrs.	Esmerelda	Van Hinkelburg	Hinkleburg's Auto Mall	902 South Terra	Denver	CA	44236	800-555-8799	
Mr.	Gerlad	Wasley	Wasley Chevy Pontiac	210 New Car Rc	Farmington	CA	06032	800-555-2456	Cars@walsey.c
Ms.	Allison	Wesley		4210 Franklin A	Hartford	CA	06105	555-555-9923	
Mr.	Henry	Chippany	Wipeout Sports Store	275 Surf La.	Los Angelis	CA	90210	800-555-9849	watersports@v
*									

All Customers from California

Fig. 3.2.4. Query Results

3.3 Using Operators in Criteria Expressions

Operators make the expression complete. They are special symbols, such as +, -, used with operands to perform calculations. Every operator has a specific meaning and a symbol. Operators help create expressions with the combination of identifiers and values. Every operator is executed according to its priority.

For example: In [BasicSalary] + [Tax], "+" is an operator.

Different types of operators are:

- Comparison operators
- Arithmetic operators
- Miscellaneous operators
- Compound criteria and logical operators

3.3.1 Comparison Operators

Comparison operators, also known as relational operators, define relation between two identifiers or two values by comparing them. These operators can be used with *Numeric* or *Date* data type.

Comparison Operators are listed in Table 3.1

OPERATOR	NAME	EXPLAINATION
>	Greater Than	Num1 > Num2 returns true if Num1 is greater than Num2
<	Less Than	Num1 < Num2 returns true if Num1 is less than Num2
>=	Greater Than Equal to	Num1 >= Num2 returns true if Num1 is greater than or Equal to Num2
<=	Less Than Equal To	Num1 <= Num2 returns true if Num1 is less than or equal to Num2
<>	Not Equal To	Num1 <> Num2 returns true if Num1 is not equal to Num2
=	Equal To	Num1 = Num2 returns true if Num1 is equal to Num2

Table 3.1: Comparison Operators

3.3.2 Arithmetic Operators

Arithmetic operators, commonly known as mathematical operators, are used with numeric data to perform calculations.

Arithmetic Operators are displayed in Table 3.2.

OPERATOR	NAME	EXPLAINATION
+	Addition	
-	Subtraction	
*	Multiplication	
/	Divide	Returns integer as a result of division of integer numbers and decimal as a result of division of decimal numbers, that is, 5\2 will return 2.5 and 5\2.5 will return 2.
\	Integer Divide	Returns integer as a result of division, that is, 5\2 will return 2 and 5\2.5 will return 2.
^	Exponentiation	Computes power, that is, the result of 5^3 is 125.
Mod	Modulo	Returns the remainder of the division of two integers, that is, 5/2 will return 1.

Table 3.2: Arithmetic Operator

3.3.3 Miscellaneous operators

In Access, a special set of operators is used with multiple data types. These operators provide an additional functionality to create expressions. Some of the miscellaneous operators are – LIKE, Between, IN, Is Null etc.

3.3.3.1 The LIKE Operator

The *LIKE* operator works with text or date data type. *LIKE* is used to match text patterns in the query criteria. This operator uses various wildcards to form different patterns. The various wildcards that can be used with LIKE are mentioned in Table 3.3.

Wildcard	Explanation	Example
*	Denotes any number of characters (0 or more)	LIKE ‘A*’ will match all the characters starting from A For example, Accounts, Audit
?	Denotes a single character	LIKE ‘B??K’ will match all the text with B as first letter, k as last letter and 2 letters in between. For example- Book, Back
#	Denotes a single digit	LIKE ‘#ABC’ will match text which starts from a digit followed by ABC.
[xyz]	Denotes a set of characters	LIKE [ABC]* will match all text starting from either A, B, or C.

Table 3.3: Wildcards with LIKE operators

3.3.3.2 The Between… And Operator

The *Between* operator is used with *Numeric* and *Date* data type to obtain a set of values within a specified range of values.

For Example:

Between 10 *and* 20 will give all the values within the range of 10 and 20 including 10 and 20.

Between #1/1/2011# *And* #8/1/2011# will return all the dates *between* 1-Jan-2011 *And* 1-Aug-2011.

(Note that dates are included between # and are written in “mm/dd/yyyy” format).

3.3.3.3 The IN operator

The *IN* operator is used to match a value to a set of values given. This operator can be used with *Numeric*, *Text*, or *Date* data types.

For Example:

[*Month*] IN (‘Jan’, ‘Feb’, ‘Apr’, ‘May’) will match all the month values which are either from the specified values.

3.3.3.4 The IS NULL operator

The *IS NULL* operator is used to find the null records in table. We use *IS* with *NULL* to indicate all the record which are null in the table. Note: Null is not “0” or “blank”.

For Example:

[*Discount*] *IS NULL* will return all the records with [*Discount*] value as NULL.

Similar to *IS NULL*, *IS NOT NULL* searches for non-null values.

For Example:

[*Advance Amount*] *IS NOT NULL* will return all records where [*Advance Amount*] is not null.

3.3.4 The Logical Operators

A logical operator results in expression that returns True or False. These operators are used to combine multiple expressions. They are also known as *Boolean operators*.

The logical operators are listed in Table 3.4.

OPERATOR	NAME	EXPLAINATION
And	Logical And	Returns True if both the expressions compared are True
Or	Logical Or	Returns True if either of the expressions compared is True
Eqv	Logical Exclusive Nor	Return True if either both the expressions are True or both the expressions are False
Xor	Logical Exclusive Or	Return True if either of the expressions is True
Not	Logical Not	Works with a single expression and returns True if the expression is False

Table 3.4: List of Logical Operators

3.3.5 Understanding the Operator Precedence

Access permits to create complex expressions containing multiple operators. To evaluate these expressions, Access determines which operator to evaluated first, and then which is next, and so forth according to a predetermined order. This order is known as *operator precedence*. Every operator is assigned a precedence order and is calculated in the same order.

The only exception to this rule is parenthesis (). Parentheses are used to group expressions and override the default order of precedence. Operations within parentheses are performed before any operations outside them. Within the parenthesis all operators are computed on basis of their precedence.

Operator precedence is similar to *BODMAS* order that is followed in algebra. Parenthesis or brackets over here perform the same function to change the priority order of operators.

Note: BODMAS =

B	Brackets first
O	Orders (ie Powers and Square Roots, etc.)
DM	Division and Multiplication (left-to-right)
AS	Addition and Subtraction (left-to-right)

Operators are first preceded in the order of their category and then within each category each operator has its own precedence order. *Operators precedence* according to their category is displayed in Table 3.5, from the highest to the lowest.

Operator Category	Precedence Order
Arithmetic	I
Comparison	II
Boolean	III

Table 3.5: Operator Precedence according to Category

Table 3.6 displays precedence of each operator within different categories from the highest to the lowest.

Category	Operator	Symbol	Precedence Order
Arithmetic			
	Exponentiation	^	I
	Multiplication and/or division (left to right)	* , /	II
	Integer division	\	III
	Modulo	Mod	IV
	Addition and/or subtraction (left to right)	+, -	V

Comparison			
	Equal	=	I
	Not equal	<>	II
	Less than	<	III
	Greater than	>	IV
	Less than or equal to	<=	V
	Greater than or equal to	>=	VI
Logical			
	Not		I
	And		II
	Or		III
	Xor		IV
	Eqv		V
	Imp		VI

Table 3.6: Operator Precedence within Category

3.3.6 Using Compound Criteria

Queries can help retrieve data in any form. Queries are usually created on multiple criteria, known as *compound criteria*.

There are two types of compound criteria - "AND" and "OR."

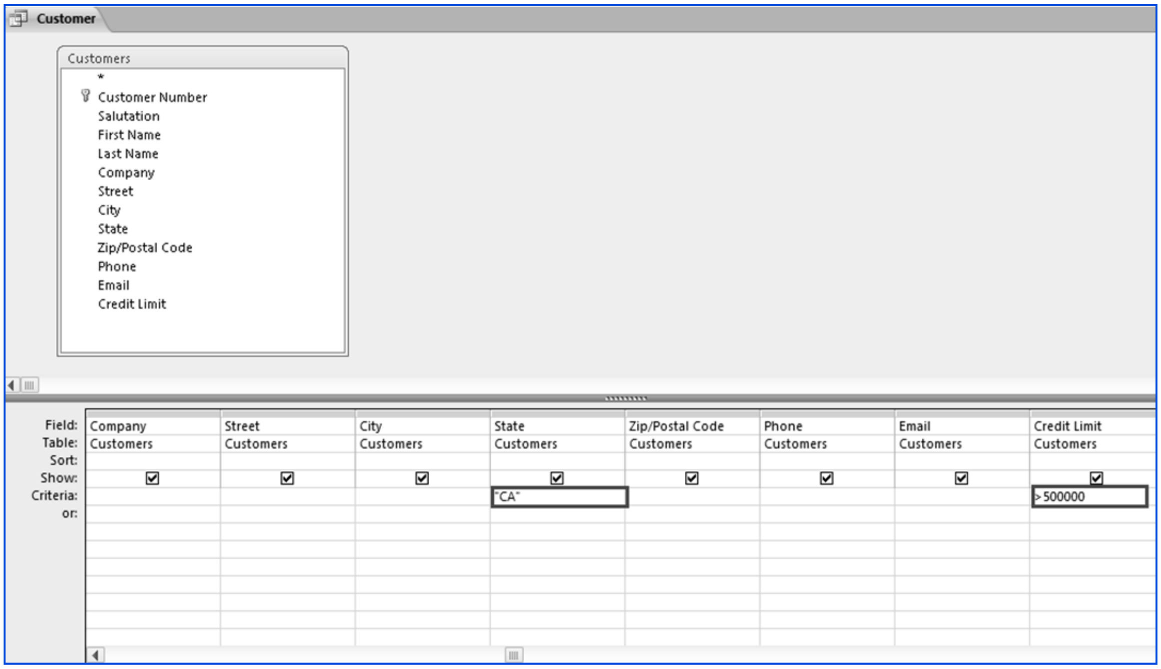
3.3.6.1 AND Criterion

In *AND*, compound criteria will return results only if each individual criterion is true.

For example, if we add criteria to the query from the *Customer* table as:

[State] = "CA" AND [Credit Limit] > 500000, it will return customers from California who have credit limit greater than 500000.

The *AND* criteria in a query designer is specified by writing all the criteria in some row of the *Criteria* tab, as shown in Fig.3.3.1.



Field:	Company	Street	City	State	Zip/Postal Code	Phone	Email	Credit Limit
Table:	Customers	Customers	Customers	Customers	Customers	Customers	Customers	Customers
Sort:								
Show:	☑	☑	☑	☑	☑	☑	☑	☑
Criteria:				"CA"				>500000
or:								

Fig. 3.3.1. Specifying AND compound criteria

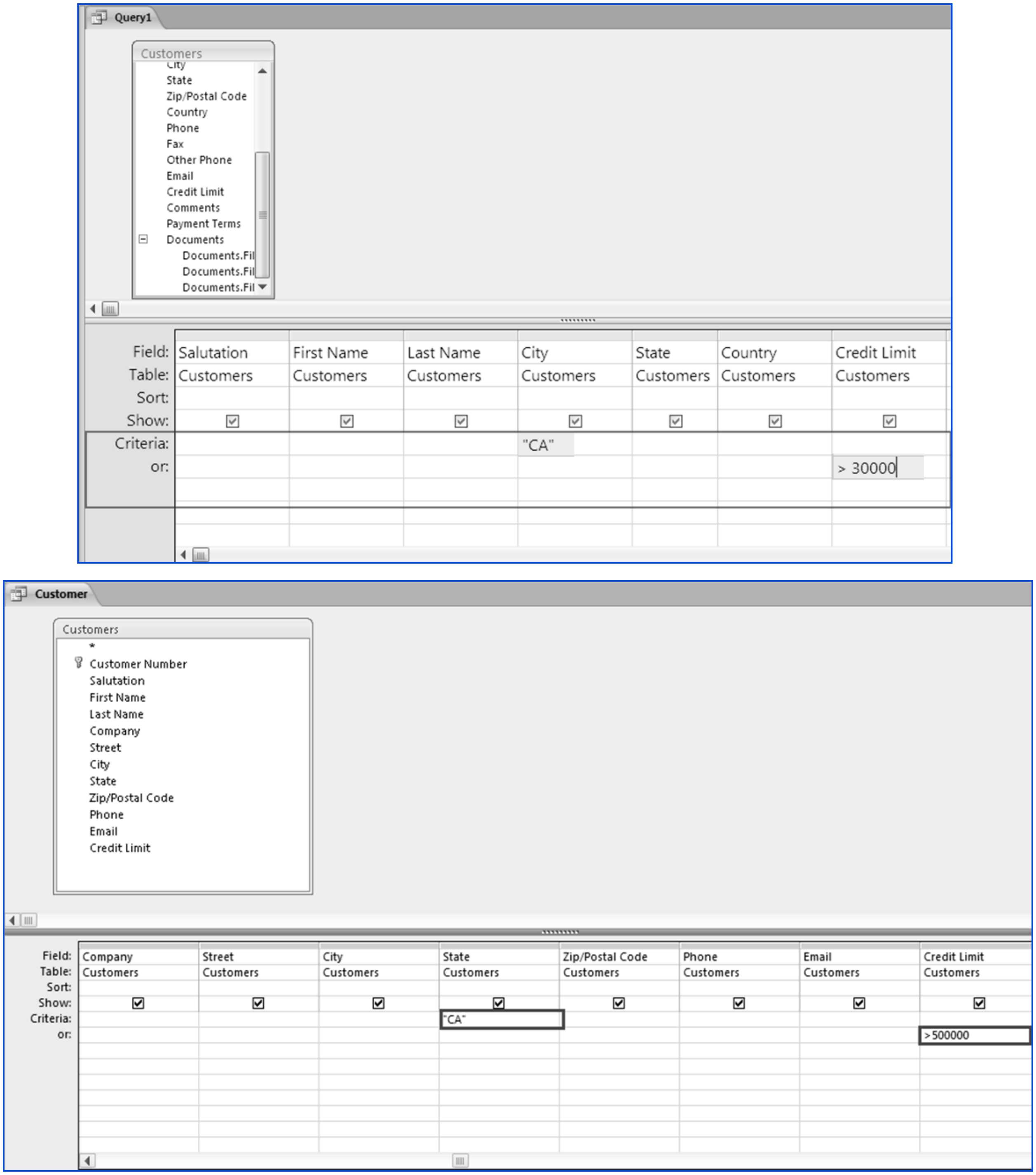
3.3.6.2 OR Criteria

The *OR* compound criteria are used in queries where we need to match either of the criterion specified in query criteria. *OR* returns result even if any criterion is true.

For example, if we add criteria to the query from the Customer table as:

[State] = "CA" *OR* [Credit Limit] > 500000, it will return all customers who are either from California or who have credit limit greater than 500000.

The *OR* criteria in a query designer is specified by writing all the criteria in different rows of the *Criteria* tab, as shown in Fig. 3.3.2.



Field:	Salutation	First Name	Last Name	City	State	Country	Credit Limit
Table:	Customers	Customers	Customers	Customers	Customers	Customers	Customers
Sort:							
Show:	☑	☑	☑	☑	☑	☑	☑
Criteria:				"CA"			
or:							> 30000

Field:	Company	Street	City	State	Zip/Postal Code	Phone	Email	Credit Limit
Table:	Customers	Customers	Customers	Customers	Customers	Customers	Customers	Customers
Sort:								
Show:	☑	☑	☑	☑	☑	☑	☑	☑
Criteria:				"CA"				
or:								>500000

Fig. 3.3.2: Specifying OR Compound Criteria

3.3.6.3 Problem Scenario

This case study focuses on the operators discussed above and how to use them as a Criteria Expression.

At the end of a quarter, Country Head of Apex Ltd. wishes to review the following data:

- A report displaying all sales orders placed in from January to March 2008.
- All inventory details from the “Car”, “Snowmobile”, and “Boat” categories along with the details of the order placed for them.
- A list of all the products that were sold with a quantity greater than 5, or the price greater than 1000.
- A contact list for all Customers whose first name starts with A, B, or C.

Solution

For creating the query containing the report of Sales Order, the query will be created on the Sales Order table, using “*Between” and ”And”* operator in query criteria.

The *inventory details* query will contain the *Inventory* table and *Sales Item Description*. The query criteria will be based on the *IN* operator.

For obtaining the list of products, query needs to be based on *Inventory* and *Sales Item Description* containing compound criteria.

To retrieve the contact list of Customers, query will be based on the *Customers* table and the criteria for the first name will contain the *LIKE* operator.

Steps for creating query containing the report of Sales Order

1. Click *Create ribbon* -> *Queries* -> *Query Design* to open the *Query Design* window.
2. Select the *Sales Order* table from the *Show Table* window and click *Add*, as shown in Fig. 3.3.3.

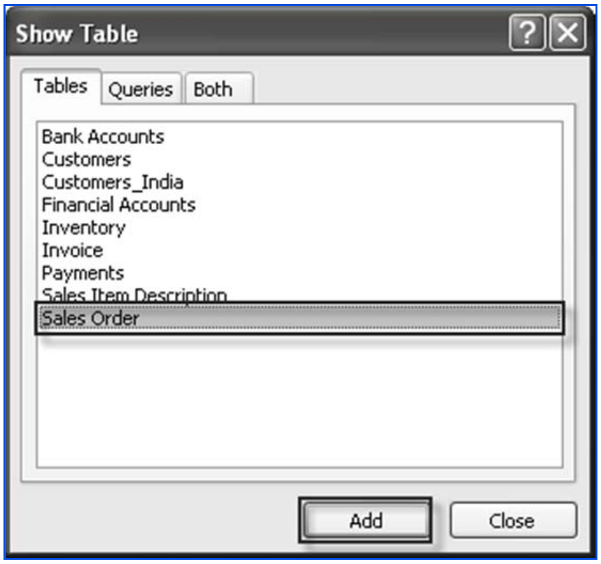
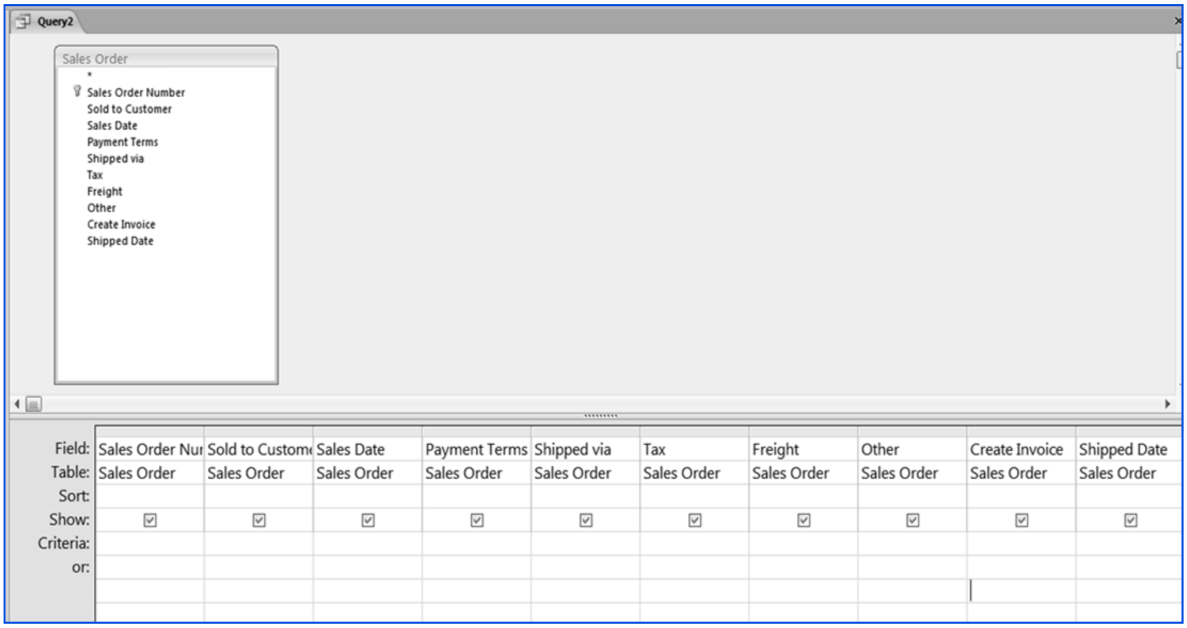


Fig. 3.3.3. Show Table window

Select all the columns from the *Sales Order* table and drag them to the *Columns* tab. The *Query* window appears, as shown in Fig. 3.3.4.

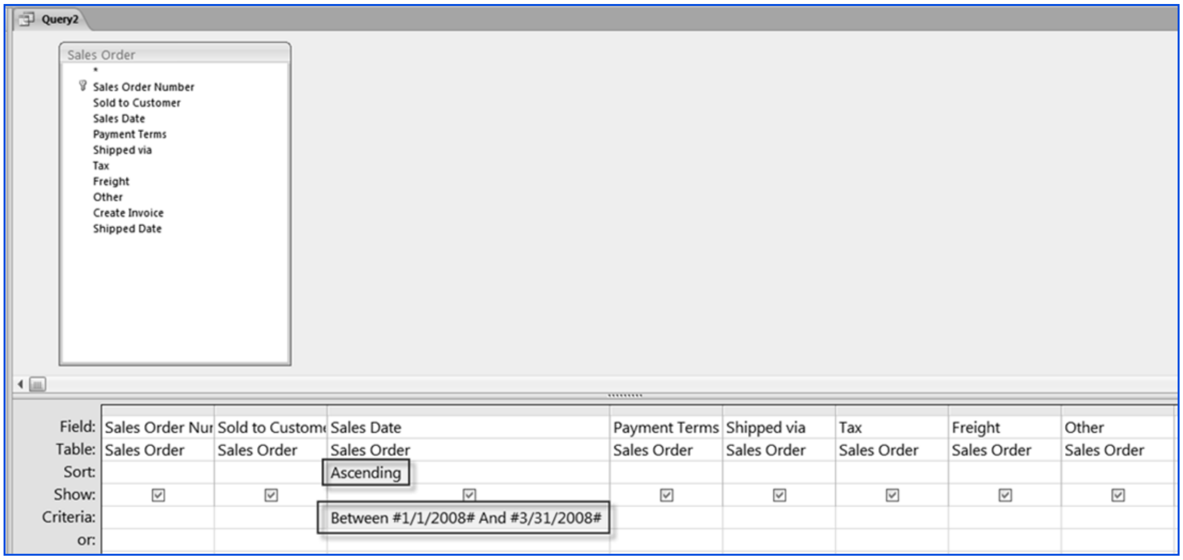


Field:	Sales Order Nur	Sold to Custom	Sales Date	Payment Terms	Shipped via	Tax	Freight	Other	Create Invoice	Shipped Date
Table:	Sales Order	Sales Order	Sales Order	Sales Order	Sales Order	Sales Order	Sales Order	Sales Order	Sales Order	Sales Order
Sort:										
Show:	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Criteria:										
or:										

Fig. 3.3.4. Select Required Columns

3. In the *Criteria* section of the *Sales Date* column, write the criteria BETWEEN 1/1/2008 AND 3/31/2008. Set the *Sort* order of column as Ascending, as shown in Fig. 3.3.5.

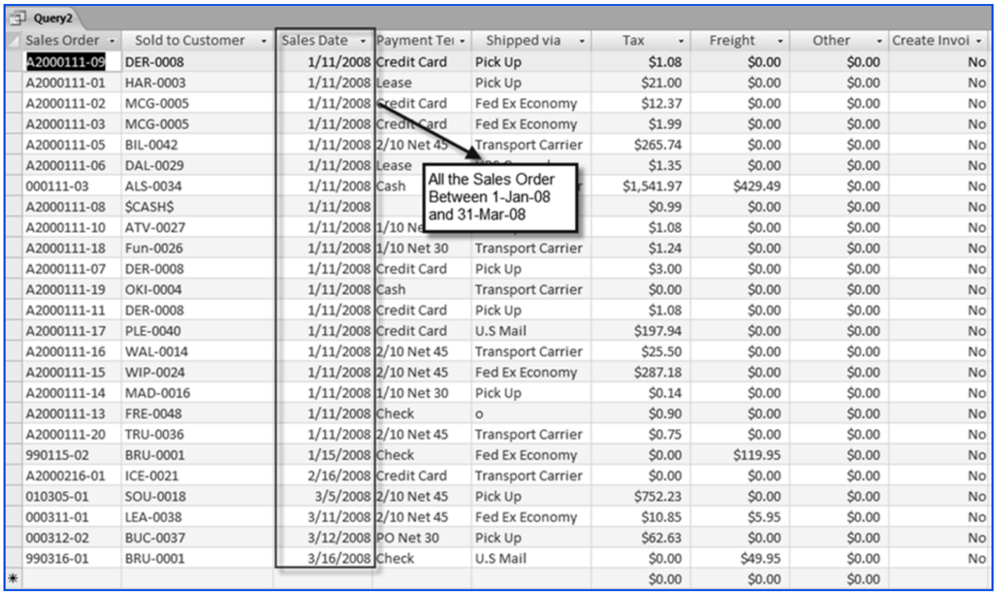
   NOTE: The *Query Builder* window formats the date with # sign.



Field:	Sales Order Nur	Sold to Custome	Sales Date	Payment Terms	Shipped via	Tax	Freight	Other
Table:	Sales Order	Sales Order	Sales Order	Sales Order	Sales Order	Sales Order	Sales Order	Sales Order
Sort:			Ascending					
Show:	☑	☑	☑	☑	☑	☑	☑	☑
Criteria:			Between #1/1/2008# And #3/31/2008#					
or:								

Fig. 3.3.5. Specify Query Criteria

4. Click the *Run* sign ❗ in *Design ribbon -> Results* to view results, as shown in Fig. 3.3.6.



Query2

Sales Order	Sold to Customer	Sales Date	Payment Te	Shipped via	Tax	Freight	Other	Create Invoi
A2000111-09	DER-0008	1/11/2008	Credit Card	Pick Up	$1.08	$0.00	$0.00	No
A2000111-01	HAR-0003	1/11/2008	Lease	Pick Up	$21.00	$0.00	$0.00	No
A2000111-02	MCG-0005	1/11/2008	Credit Card	Fed Ex Economy	$12.37	$0.00	$0.00	No
A2000111-03	MCG-0005	1/11/2008	Credit Card	Fed Ex Economy	$1.99	$0.00	$0.00	No
A2000111-05	BIL-0042	1/11/2008	2/10 Net 45	Transport Carrier	$265.74	$0.00	$0.00	No
A2000111-06	DAL-0029	1/11/2008	Lease	[illegible]	$1.35	$0.00	$0.00	No
000111-03	ALS-0034	1/11/2008	Cash	[illegible]	$1,541.97	$429.49	$0.00	No
A2000111-08	$CASH$	1/11/2008		[illegible]	$0.99	$0.00	$0.00	No
A2000111-10	ATV-0027	1/11/2008	1/10 Ne	[illegible]	$1.08	$0.00	$0.00	No
A2000111-18	Fun-0026	1/11/2008	1/10 Net 30	Transport Carrier	$1.24	$0.00	$0.00	No
A2000111-07	DER-0008	1/11/2008	Credit Card	Pick Up	$3.00	$0.00	$0.00	No
A2000111-19	OKI-0004	1/11/2008	Cash	Transport Carrier	$0.00	$0.00	$0.00	No
A2000111-11	DER-0008	1/11/2008	Credit Card	Pick Up	$1.08	$0.00	$0.00	No
A2000111-17	PLE-0040	1/11/2008	Credit Card	U.S Mail	$197.94	$0.00	$0.00	No
A2000111-16	WAL-0014	1/11/2008	2/10 Net 45	Transport Carrier	$25.50	$0.00	$0.00	No
A2000111-15	WIP-0024	1/11/2008	2/10 Net 45	Fed Ex Economy	$287.18	$0.00	$0.00	No
A2000111-14	MAD-0016	1/11/2008	1/10 Net 30	Pick Up	$0.14	$0.00	$0.00	No
A2000111-13	FRE-0048	1/11/2008	Check	o	$0.90	$0.00	$0.00	No
A2000111-20	TRU-0036	1/11/2008	2/10 Net 45	Transport Carrier	$0.75	$0.00	$0.00	No
990115-02	BRU-0001	1/15/2008	Check	Fed Ex Economy	$0.00	$119.95	$0.00	No
A2000216-01	ICE-0021	2/16/2008	Credit Card	Transport Carrier	$0.00	$0.00	$0.00	No
010305-01	SOU-0018	3/5/2008	2/10 Net 45	Pick Up	$752.23	$0.00	$0.00	No
000311-01	LEA-0038	3/11/2008	2/10 Net 45	Fed Ex Economy	$10.85	$5.95	$0.00	No
000312-02	BUC-0037	3/12/2008	PO Net 30	Pick Up	$62.63	$0.00	$0.00	No
990316-01	BRU-0001	3/16/2008	Check	U.S Mail	$0.00	$49.95	$0.00	No
*					$0.00	$0.00	$0.00	

All the Sales Order Between 1-Jan-08 and 31-Mar-08

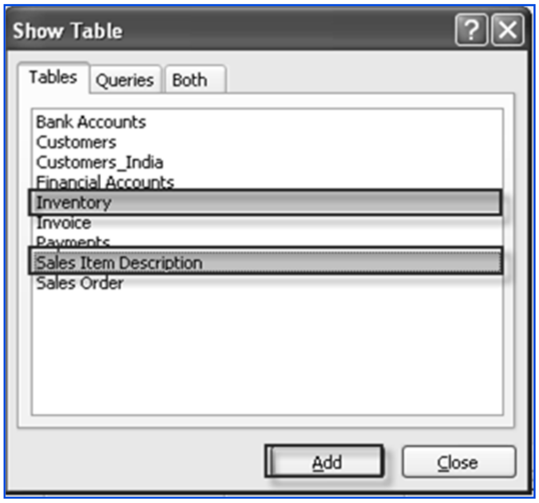
Fig. 3.3.6. Query Result

5. Click 💾 on the toolbar to save the query. Type the name of query as "Sales Order QTR1".

Steps for creating inventory details query

1. Click *Create ribbon* -> *Queries* -> *Query Design* to open the *Query Design* window.

2. Select the *Inventory and Sales Item Description* table from the *Show Table* window and click *Add*, as shown in Fig. 3.3.7.

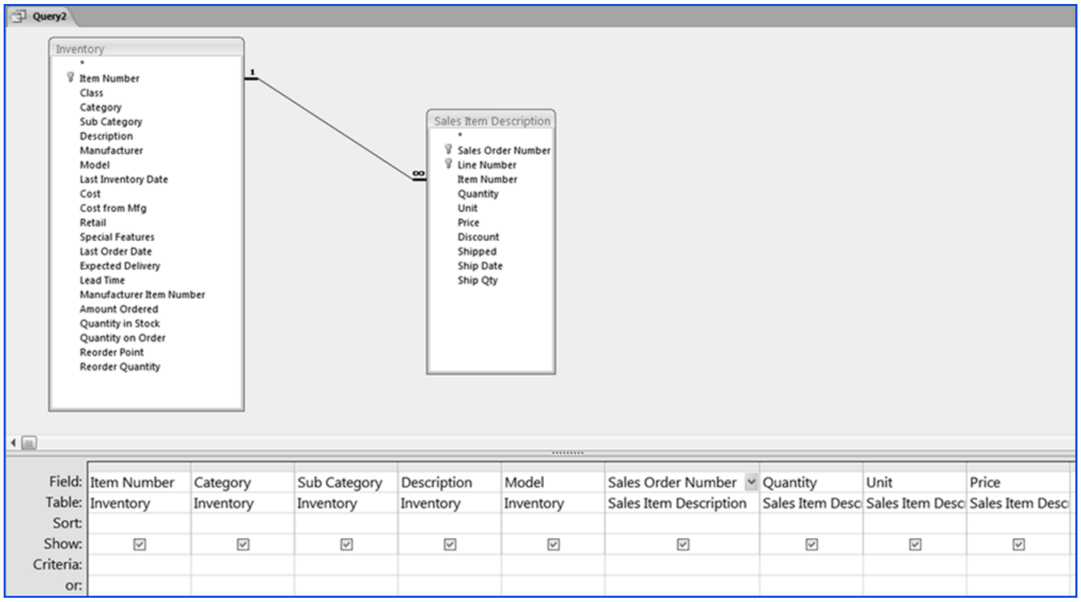


Show Table

Bank Accounts
Customers
Customers_India
Financial Accounts
Inventory
Invoice
Payments
Sales Item Description
Sales Order

Fig. 3.3.7: Show Table Window

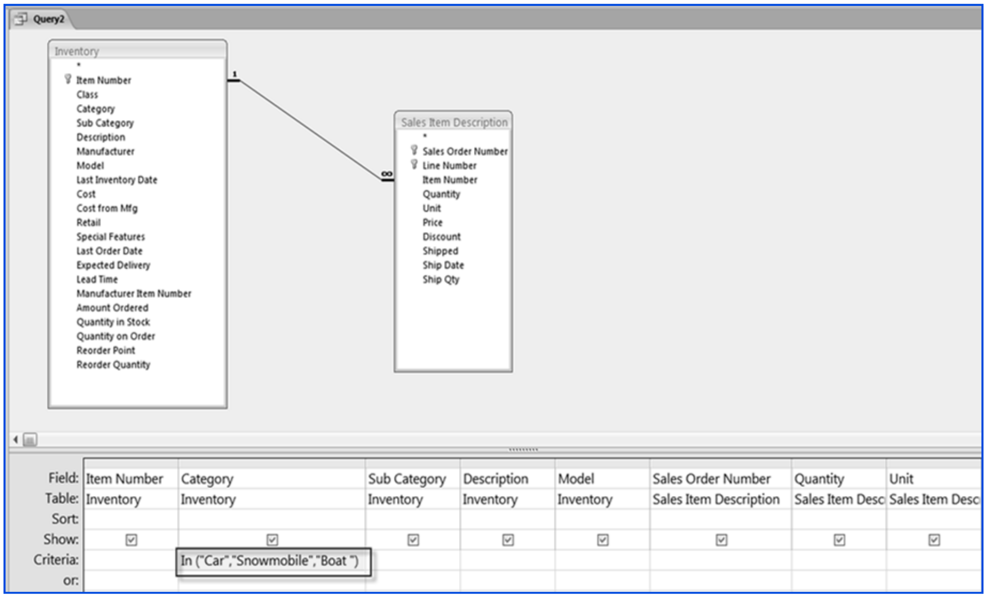
3. Select *Item Number, Category, Sub Category, Description,* and *Model* from the *Inventory* table and drag them to the *Columns* tab. Similarly drag *Sales Order Number, Quantity, Unit,* and *Price*. The *Query* window appears, as shown in Fig. 3.3.8.



Field:	Item Number	Category	Sub Category	Description	Model	Sales Order Number	Quantity	Unit	Price
Table:	Inventory	Inventory	Inventory	Inventory	Inventory	Sales Item Description	Sales Item Desc	Sales Item Desc	Sales Item Desc
Sort:									
Show:	☑	☑	☑	☑	☑	☑	☑	☑	☑
Criteria:									
or:									

Fig. 3.3.8: Specify Required Fields

4. In the Criteria section of the *Category* column write the criteria *IN* (Car, Snowmobile, Boat), as shown in Fig. 3.3.9.



Field:	Item Number	Category	Sub Category	Description	Model	Sales Order Number	Quantity	Unit
Table:	Inventory	Inventory	Inventory	Inventory	Inventory	Sales Item Description	Sales Item Desc	Sales Item Desc
Sort:								
Show:	☑	☑	☑	☑	☑	☑	☑	☑
Criteria:		In ("Car","Snowmobile","Boat ")						
or:								

Fig. 3.3.9: Specify Criteria for Column

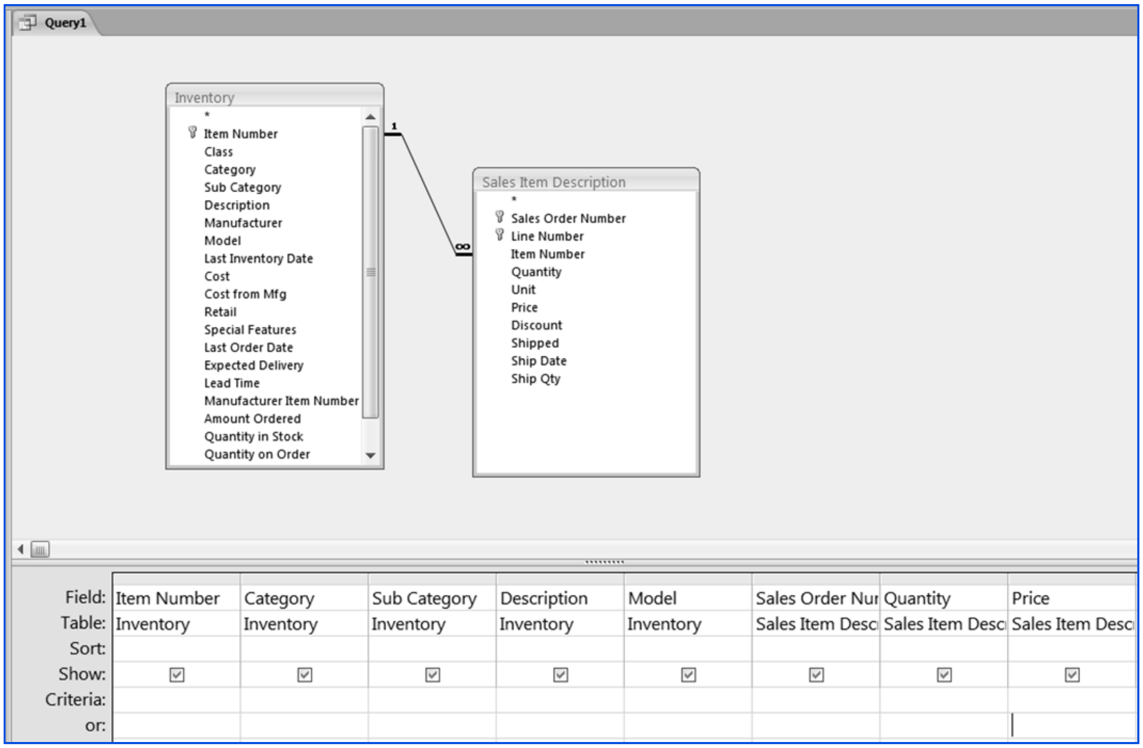
5. Click 💾 on the toolbar to save the query. Type the name of query as “Inventory Sales”.

Steps for creating the query to obtain the list of products

1. Click *Create ribbon* -> *Queries* -> *Query Design* to open the *Query Design* window.

2. Select the *Inventory and Sales Item Description* table from the *Show* Table window and click *Add*.

3. Select *Item Number, Category, Sub Category, Description,* and *Model* from the *Inventory* table and drag them to the *Columns* tab. Similarly, drag *Sales Order Number, Quantity,* and *Price*. The *Query* window appears, as shown in Fig. 3.3.10.

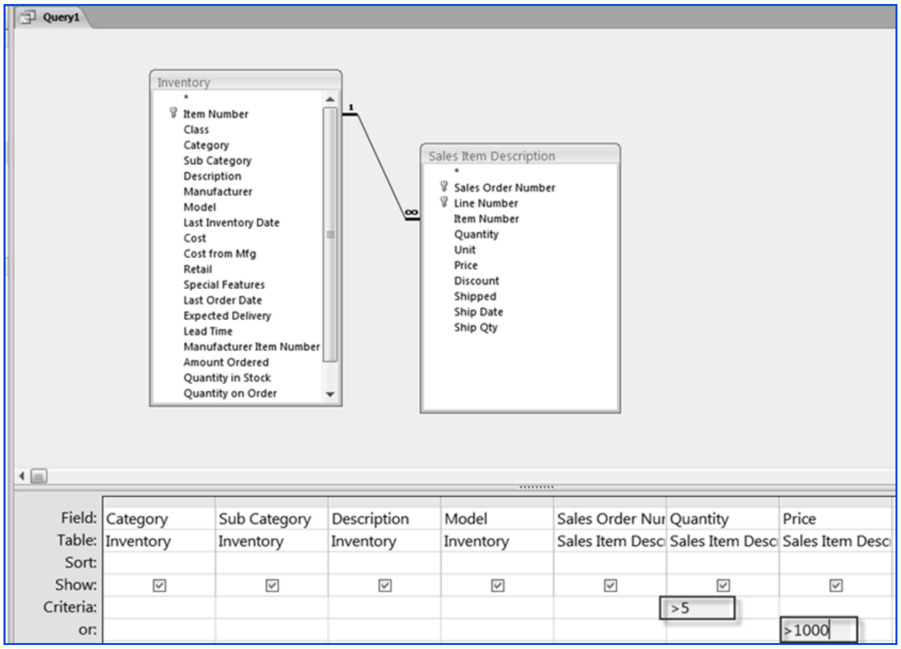


Field:	Item Number	Category	Sub Category	Description	Model	Sales Order Nur	Quantity	Price
Table:	Inventory	Inventory	Inventory	Inventory	Inventory	Sales Item Desc	Sales Item Desc	Sales Item Desc
Sort:								
Show:	☑	☑	☑	☑	☑	☑	☑	☑
Criteria:								
or:								

Fig. 3.3.10. Specify Required Columns

4. In the *Criteria* section of the *Quantity* column write the criteria > 5. In the next row, write the criteria for *Price* as > 1000. The Query design window should appear as shown in Fig. 3.3.11.

   NOTE: Since this is an *OR* compound criteria, both the criteria are written in different rows.

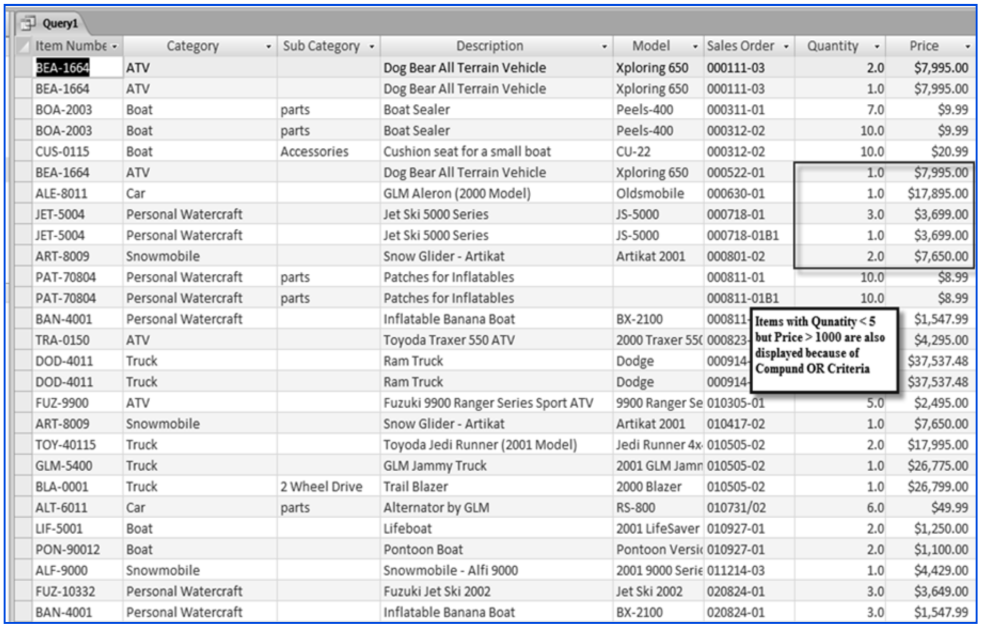


Field:	Category	Sub Category	Description	Model	Sales Order Nur	Quantity	Price
Table:	Inventory	Inventory	Inventory	Inventory	Sales Item Desc	Sales Item Desc	Sales Item Desc
Sort:							
Show:	☑	☑	☑	☑	☑	☑	☑
Criteria:						>5	
or:							>1000

Fig. 3.3.11. Specify Compound Criteria

5. Click on the toolbar to save the query. Type the name of query as “Inventory Sales – Qty Price”.

6. Click in *Design ribbon* -> *Results* to view the results, as shown in Fig. 3.3.12.



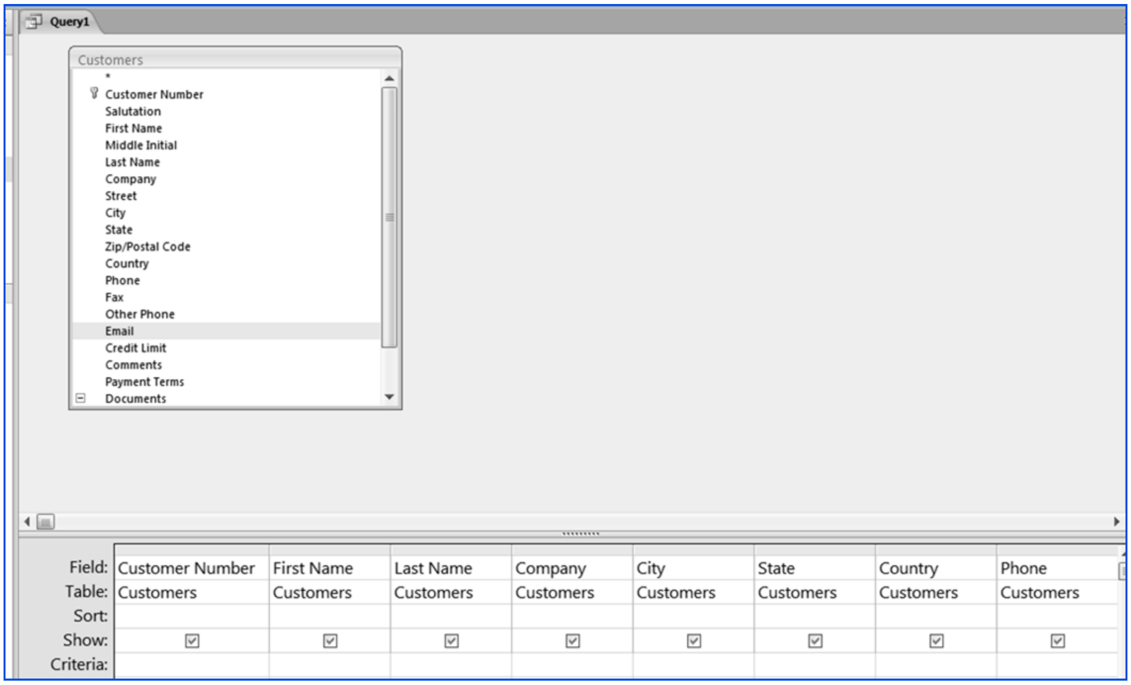
Item Numbe	Category	Sub Category	Description	Model	Sales Order	Quantity	Price
BEA-1664	ATV		Dog Bear All Terrain Vehicle	Xploring 650	000111-03	2.0	$7,995.00
BEA-1664	ATV		Dog Bear All Terrain Vehicle	Xploring 650	000111-03	1.0	$7,995.00
BOA-2003	Boat	parts	Boat Sealer	Peels-400	000311-01	7.0	$9.99
BOA-2003	Boat	parts	Boat Sealer	Peels-400	000312-02	10.0	$9.99
CUS-0115	Boat	Accessories	Cushion seat for a small boat	CU-22	000312-02	10.0	$20.99
BEA-1664	ATV		Dog Bear All Terrain Vehicle	Xploring 650	000522-01	1.0	$7,995.00
ALE-8011	Car		GLM Aleron (2000 Model)	Oldsmobile	000630-01	1.0	$17,895.00
JET-5004	Personal Watercraft		Jet Ski 5000 Series	JS-5000	000718-01	3.0	$3,699.00
JET-5004	Personal Watercraft		Jet Ski 5000 Series	JS-5000	000718-01B1	1.0	$3,699.00
ART-8009	Snowmobile		Snow Glider - Artikat	Artikat 2001	000801-02	2.0	$7,650.00
PAT-70804	Personal Watercraft	parts	Patches for Inflatables		000811-01	10.0	$8.99
PAT-70804	Personal Watercraft	parts	Patches for Inflatables		000811-01B1	10.0	$8.99
BAN-4001	Personal Watercraft		Inflatable Banana Boat	BX-2100	000811		$1,547.99
TRA-0150	ATV		Toyoda Traxer 550 ATV	2000 Traxer 55(	000823		$4,295.00
DOD-4011	Truck		Ram Truck	Dodge	000914		$37,537.48
DOD-4011	Truck		Ram Truck	Dodge	000914		$37,537.48
FUZ-9900	ATV		Fuzuki 9900 Ranger Series Sport ATV	9900 Ranger Se	010305-01	5.0	$2,495.00
ART-8009	Snowmobile		Snow Glider - Artikat	Artikat 2001	010417-02	1.0	$7,650.00
TOY-40115	Truck		Toyoda Jedi Runner (2001 Model)	Jedi Runner 4x	010505-02	2.0	$17,995.00
GLM-5400	Truck		GLM Jammy Truck	2001 GLM Jamn	010505-02	1.0	$26,775.00
BLA-0001	Truck	2 Wheel Drive	Trail Blazer	2000 Blazer	010505-02	1.0	$26,799.00
ALT-6011	Car	parts	Alternator by GLM	RS-800	010731/02	6.0	$49.99
LIF-5001	Boat		Lifeboat	2001 LifeSaver	010927-01	2.0	$1,250.00
PON-90012	Boat		Pontoon Boat	Pontoon Versi(	010927-01	2.0	$1,100.00
ALF-9000	Snowmobile		Snowmobile - Alfi 9000	2001 9000 Serie	011214-03	1.0	$4,429.00
FUZ-10332	Personal Watercraft		Fuzuki Jet Ski 2002	Jet Ski 2002	020824-01	3.0	$3,649.00
BAN-4001	Personal Watercraft		Inflatable Banana Boat	BX-2100	020824-01	3.0	$1,547.99

Items with Qunatity < 5 but Price > 1000 are also displayed because of Compund OR Criteria

Fig. 3.3.12. Query Results

Steps for creating query to retrieve the contact list of Customers

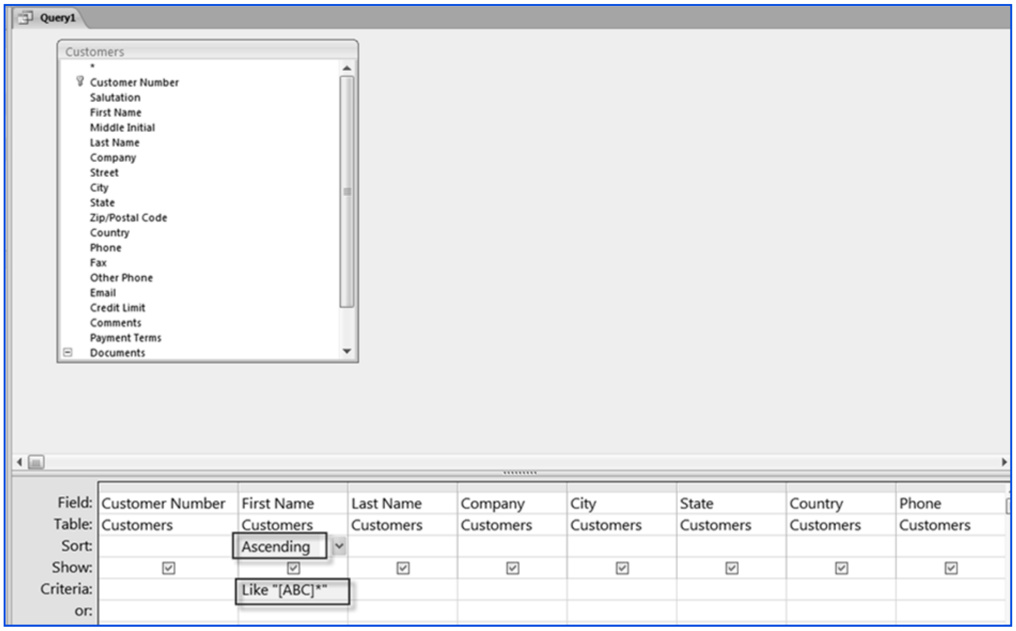
1. Click *Create ribbon* -> *Queries* -> *Query Design* to open the *Query Design* window.
2. Select the *Customers* table from the *Show Table* window and click *Add*.
3. Select *Customer Number, First Name, Last Name, Company, City, State, Country,* and *Phone* from *Customers* and drag them to the *Columns* tab. The *Query* window appears, as shown in Fig. 3.3.13.



Field:	Customer Number	First Name	Last Name	Company	City	State	Country	Phone
Table:	Customers	Customers	Customers	Customers	Customers	Customers	Customers	Customers
Sort:								
Show:	☑	☑	☑	☑	☑	☑	☑	☑
Criteria:								

Fig. 3.3.13. Specify Require Fields

4. In the *Criteria* section of the *First Name* column write the criteria LIKE “[ABC]*”. The wildcard *[ABC]** with operator *LIKE* specifies that it should match any string starting from A, B, or C. Also set the sort order of the *First Name* field to *Ascending*. The Query Design window should appear as shown in Fig. 3.3.14.

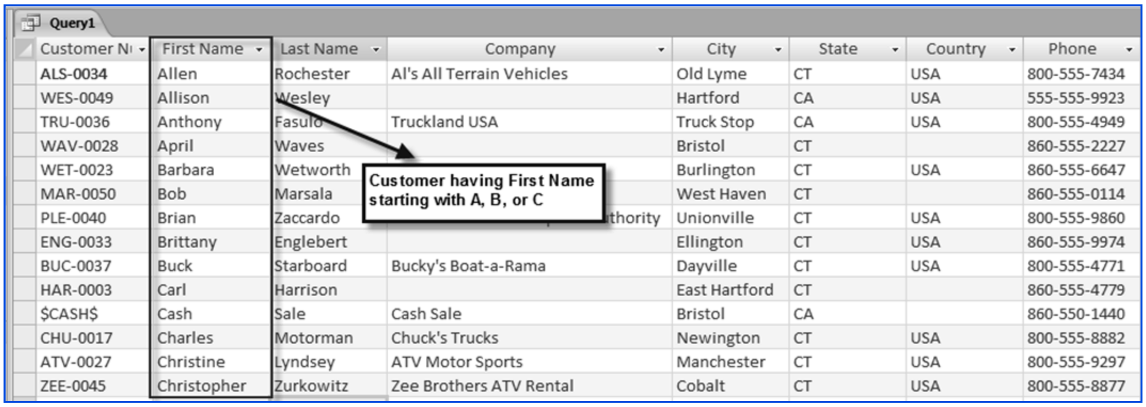


Field:	Customer Number	First Name	Last Name	Company	City	State	Country	Phone
Table:	Customers	Customers	Customers	Customers	Customers	Customers	Customers	Customers
Sort:		Ascending						
Show:	☑	☑	☑	☑	☑	☑	☑	☑
Criteria:		Like "[ABC]*"						
or:								

Fig. 3.3.14. Specify Query Criteria

5. Click on the toolbar to save the query. Type the name of query as "Customers Contact List – ABC".

6. Click Run in *Design ribbon* -> *Results* to view the results, as shown in Fig. 3.3.15.



Customer Nı	First Name	Last Name	Company	City	State	Country	Phone
ALS-0034	Allen	Rochester	Al's All Terrain Vehicles	Old Lyme	CT	USA	800-555-7434
WES-0049	Allison	Wesley		Hartford	CA	USA	555-555-9923
TRU-0036	Anthony	Fasulo	Truckland USA	Truck Stop	CA	USA	800-555-4949
WAV-0028	April	Waves		Bristol	CT		860-555-2227
WET-0023	Barbara	Wetworth		Burlington	CT	USA	860-555-6647
MAR-0050	Bob	Marsala		West Haven	CT		860-555-0114
PLE-0040	Brian	Zaccardo	thority	Unionville	CT	USA	800-555-9860
ENG-0033	Brittany	Englebert		Ellington	CT	USA	860-555-9974
BUC-0037	Buck	Starboard	Bucky's Boat-a-Rama	Dayville	CT	USA	800-555-4771
HAR-0003	Carl	Harrison		East Hartford	CT		860-555-4779
$CASH$	Cash	Sale	Cash Sale	Bristol	CA		860-550-1440
CHU-0017	Charles	Motorman	Chuck's Trucks	Newington	CT	USA	800-555-8882
ATV-0027	Christine	Lyndsey	ATV Motor Sports	Manchester	CT	USA	800-555-9297
ZEE-0045	Christopher	Zurkowitz	Zee Brothers ATV Rental	Cobalt	CT	USA	800-555-8877

Fig. 3.3.15. Query Results

Many complex *Criteria Expressions* for a query can be created using the combination of operators and operands. Combining multiple criteria and different operators can help get the required data.

3.4 Using the Built-In Functions

Built-In Functions provide specialized operations to enhance the working of Access. We can perform mathematical, financial, comparative, and other operations using functions.

Some useful types of functions available in Access are:

- Mathematical
- Date/Time
- Financial
- SQL Aggregate
- Text

3.4.1 Using Text Functions

Text functions are used to perform various operations on strings, such as manipulating strings, concatenate the string, extracting a portion of string.

Some useful string functions are listed in Table 3.7.

FUNCTION NAME	EXPLAINATION	EXAMPLE
Left()	Returns specified number of characters from left of the string	Left(“Access”,3) will return Acc
Right()	Returns specified number of characters from right of the string	Right(“Access”,3) will return ess
Mid()	Returns specified number of characters from the given position in string	Mid(“Access”,2,3) will return 3 characters from 2 positions, that is, cce
Len()	Returns the length of the given string	Len(“Access”) will return 6
Lcase()	Converts the text to lower case	Lcase(“ACCESS”) will return access
Ucase()	Converts the text to capital case	Ucase(“access”) will return ACCESS
Instr()	Returns the position of first occurrence of a string in another string	Instr(“Operations”,”ra”) will return 4
Trim()	Removes leading or trailing spaces in a text	Trim(“ Acc ess “) will return “Acc ess”
Replace()	Converts a substring from the given string into specified string	Replace(“Account Transactions”,”Account”,”Daily”) returns Daily Transactions
Strcomp()	Compares two strings Returns 0 if strings are same. 1 if first string is greater the second. -1 If first string is less than the second.	Strcomp(“Access”,”Access”) returns 0 Strcomp(“Access”,”Training”) returns -1 as “Training” is greater than “Access”
StrReverse()	Returns the string in reverse order	StrReverse(“access”) returns “ssecca”

Table 3.7: Text Functions

3.4.2 Using Date and Time Functions

These functions are used to handle Date and Time data. This group contains various functions such as extracting a part of date or adding two dates.

Few of the important Date/Time functions, are shown in Table 3.8.

FUNCTION NAME	EXPLANATION	EXAMPLE
Now()	Returns current date and time	Now() Returns 07/18/2012 12:20:55
Date()	Returns current date	Date()_ Returns 07/18/2012
Time()	Returns current time	Time()_ Returns 12:30:15
DateDiff()	Returns difference two dates. The interval for difference can be in terms of Days (“d”), months(“m”), Quarter (“q”), years(“yyyy”), weeks (“ww”)	DateDiff (“m”,#7/18/2011#,#12/3/2011#) Returns 5 as the difference between two dates in terms of months is 5.
DateAdd()	Adds a specified interval to the given Date. Intervals in this can be used as same in DateDiff	DateAdd(“q”,1,#1/11/2012#) will return 4/11/2012 As a Quarter added to January returns April.
DatePart()	Extracts a portion of a date from the given date	DatePart(“ww”,#2/3/2012#) returns 6
Month()	Returns month in integer from the given date	Month(#12/1/2011#) returns 12
MonthName()	Returns name of the month, that is, given as an integer	MonthName(12) returns December
Year()	Returns year from a given date	Year(#2/2/2012#) returns 2012

Table 3.8: Date and Time Functions

3.4.3 Using Math Functions

Math functions are used for performing calculations on Numeric data. These functions provide us the property of performing various mathematical operations.

Some important mathematical functions are listed in Table 3.9.

FUNCTION NAME	**EXPLANATION**	**EXAMPLE**
Abs()	Returns the absolute value of a number	Abs (14) returns 14 Abs(-14) returns 14
Fix()	Returns the nearest integer for a negative number	Fix(-125.64) returns -125
Int()	Returns an integer for a specific value	Int(23.64) returns 23
Round()	Returns a number rounded to specified number of digits	Round(18.234,2) returns 18.23 Round(18.246) return 18.25
Rnd()	Returns any generated random number	Rnd() returns any Random no.
Sgn()	Returns an integer representing sign of a number -1 for –ve number 1 for +ve number 0 for Zero	Sgn(-14) returns -1
Sqr()	Returns square root of a number	Sqr(16) returns 4
Log()	Returns logarithm of a number	

Table 3.9: Math Function

3.4.4 Using Financial Functions

Financial functions are used to perform many standard financial calculations, such as interest rates, annuity or loan payments, and depreciation.

Some extensively used financial functions are listed in Table 3.10.

FUNCTION NAME	**EXPLAINATION**	**EXAMPLE**

DDB()	Returns the double-declining balance method of depreciation return based on the formula:
	Depreciation / period = ((cost salvage) * factor) / life
		DDB(*cost*, *salvage*, *life*, *period*[, *factor*])
		If we calculate depreciation for Rs.5, 000 computer with a Rs.200 salvage value and an estimated useful life of three years for the first year.
		DDB(5000,200,3,1)

FV()	Returns the future value of an annuity based on periodic, fixed payment and fixed interest rate	FV(*Rate*, *Payment Periods*, *Payment* [, *Present Value*] [, *Type*])
		If we calculate Future value for a rate of 8% for 10 installments depositing an amount of Rs. 200 monthly. FV will be calculated as:
		FV(0.08/12,10,-200)

PV()	Returns the present value of an annuity	PV (*Rate*, *Payment Periods*, *Payment* [,

	based on periodic, fixed payments to be paid in future and fixed interest rate	*Future Value*] [, *Type*]) To calculate Present value of an annuity that will provide 5,000 a year for the next 20 years at a rate of .0825. We will use: PV(.08,5000,20)

SYD()	Returns the sum-of-years depreciation of an asset for a specific period	SYD(*Cost Of Asset*, *Salvage Value*, *Length Of Useful Life*, *Period*)
		To calculate the depreciation charges of a building that cost 365820 to build has a salvage value of 5390, and an estimated useful life of 30 years. We will use function SYD as:
		SYD(365820,5390,15,1)

PMT()	Returns the payment for an annuity based on periodic, fixed payment and fixed interest rate	To calculate a payment amount for a 6 percent loan of 360 months for 110000. The formula will be: PMT(.005, 360, -110000)
RATE()	Returns the interest rate per period	Rate (*nper, pmt, pv* [*, fv*] [*, type*]) To calculate interest rate on a Rs. 5,000 loan where monthly payments of Rs.250 are made for 2 years Rate(2*12,250,5000)

Table 3.10: Financial Functions

3.4.5 Problem Scenario

The Operations Manager of Apex Ltd. received many complaints from customers about the orders not reaching them on time. He wants a report to be submitted for all the orders which were delayed (under normal cycle orders must be shipped within 7 days).

Solution

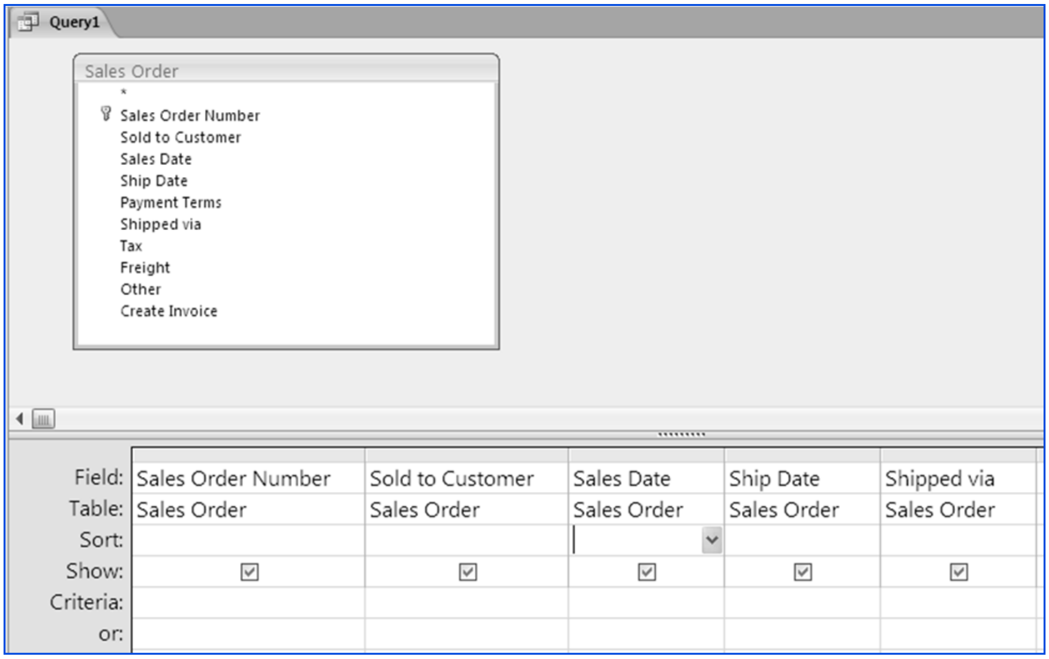
The required query will use the tables *Sales Order*, the criteria will be placed on *shipped date* using the *DateDiff* () function.

Steps for creating a query

1. Click *Create ribbon* -> *Queries* -> *Query Design* to open the *Query Design* window.

   Select the *Sales Order* table from the *Show Table* window and click *Add*.

2. Select the columns *Sales Order Number, Sold to Customer, Sales Date, Ship Date, Shipped Via*. The *Query* window appears, as shown in Fig. 3.4.1.

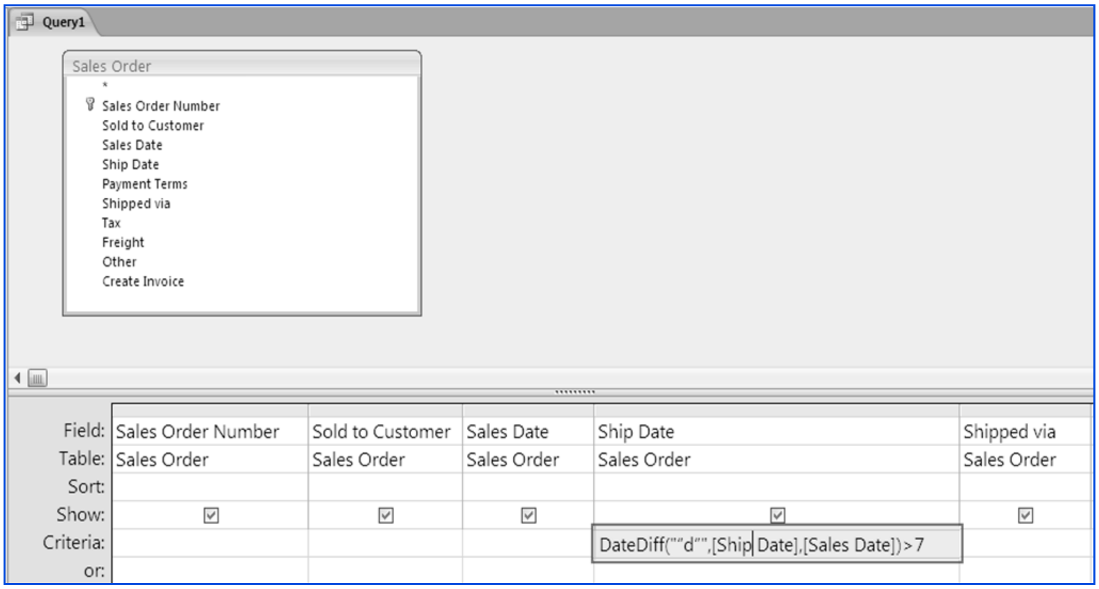


Sales Order
*
Sales Order Number
Sold to Customer
Sales Date
Ship Date
Payment Terms
Shipped via
Tax
Freight
Other
Create Invoice

Field:	Sales Order Number	Sold to Customer	Sales Date	Ship Date	Shipped via
Table:	Sales Order	Sales Order	Sales Order	Sales Order	Sales Order
Sort:					
Show:	☑	☑	☑	☑	☑
Criteria:					
or:					

Fig. 3.4.1. Select required column

Specify the criteria for *Shipped Date* as *DateDiff*(**"d",[Sales Date],[Ship Date])>7**. The *DateDiff*() function returns difference between two dates, “d” forces it to return it in terms of days. The query looks like, as shown in Fig. 3.4.2.



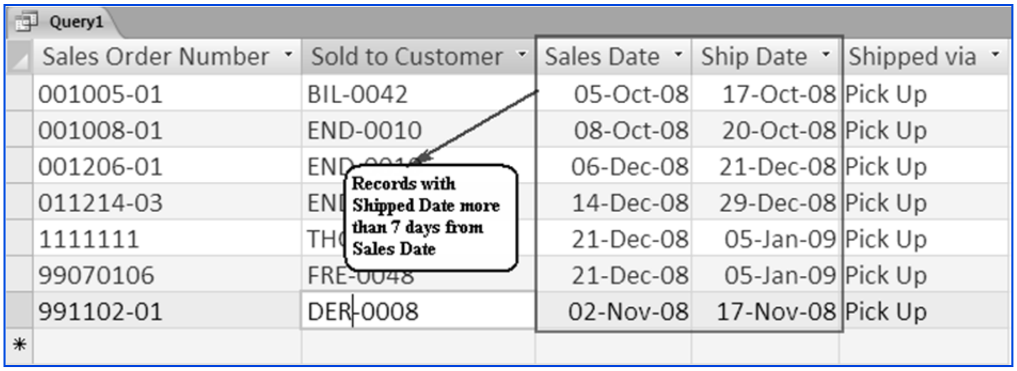
Sales Order
*
Sales Order Number
Sold to Customer
Sales Date
Ship Date
Payment Terms
Shipped via
Tax
Freight
Other
Create Invoice

Field:	Sales Order Number	Sold to Customer	Sales Date	Ship Date	Shipped via
Table:	Sales Order	Sales Order	Sales Order	Sales Order	Sales Order
Sort:					
Show:	☑	☑	☑	☑	☑
Criteria:				DateDiff("“d”",[Ship Date],[Sales Date])>7	
or:					

Fig. 3.4.2. Specify Query criteria using Functions

3. Click on the toolbar to save the query. Type the name of query as *Orders Delayed*.

4. Click in *Design ribbon* -> *Results* to view the results, as shown in Fig. 3.4.3.



Query1

Sales Order Number	Sold to Customer	Sales Date	Ship Date	Shipped via
001005-01	BIL-0042	05-Oct-08	17-Oct-08	Pick Up
001008-01	END-0010	08-Oct-08	20-Oct-08	Pick Up
001206-01	END-0010	06-Dec-08	21-Dec-08	Pick Up
011214-03	END	14-Dec-08	29-Dec-08	Pick Up
1111111	THO	21-Dec-08	05-Jan-09	Pick Up
99070106	FRE-0048	21-Dec-08	05-Jan-09	Pick Up
991102-01	DER-0008	02-Nov-08	17-Nov-08	Pick Up
*				

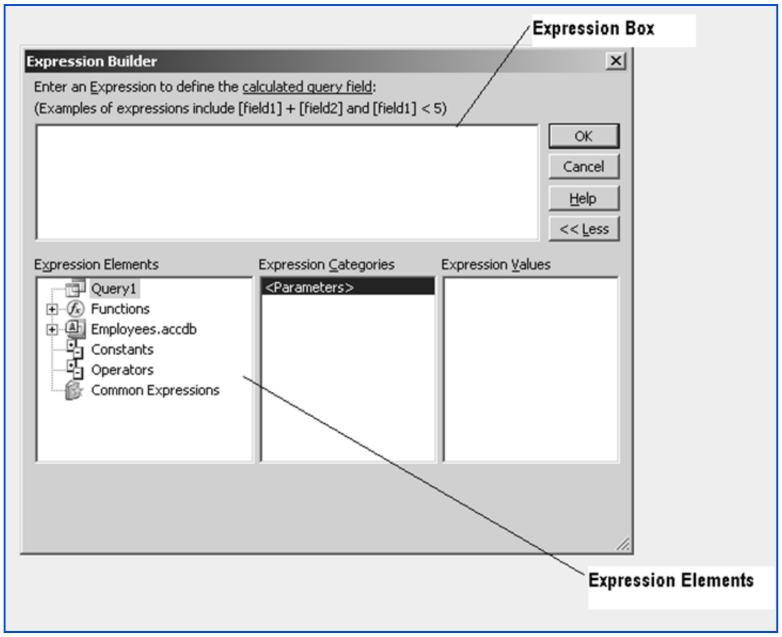
Records with Shipped Date more than 7 days from Sales Date

Fig. 3.4.3. Query Results

3.5 Working with Expression Builder

The *Expression Builder* tool in Access helps build complex expressions. It contains easy access to access names and properties of columns of the tables. *Expression Builder* also contains a set of predefined functions in Access and also some prebuilt expression as to display page numbers.

The *Expression Builder* window is shown in Fig. 3.5.1.



Expression Builder

Enter an Expression to define the calculated query field:
(Examples of expressions include [field1] + [field2] and [field1] < 5)

Fig. 3.5.1: Expression Builder Window

3.5.1 View of Expression Builder

Expression box - A text box in which expressions are written. Any value from the *Operators* button or *Expression Elements* can be pasted into *Expression Box*.

Operator buttons - Various operators are available in the *Operators Buttons* tab. We can just click any operator to get it in the *Expression Box* text box.

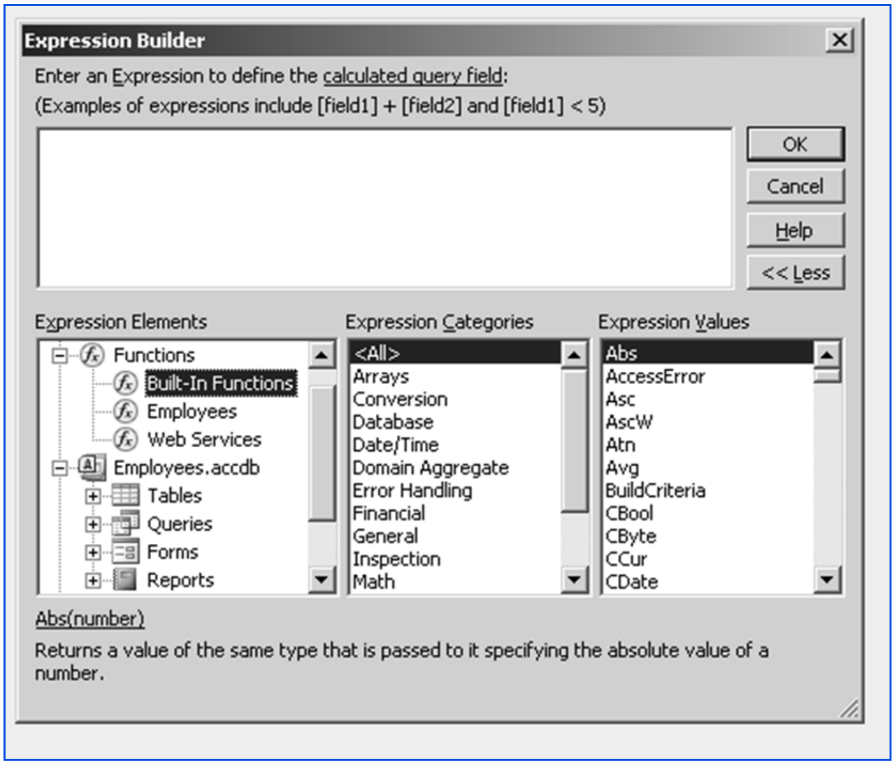
Expression elements – Expression element contains three tabs

- First one is for the objects in the database like *Tables*, *Queries*, *Forms*, or *Reports*.
- Second is for the sub elements of the objects selected in the *First* tab like fields of the table, containers in *Reports* and *Forms*.
- Third is for the properties of the element selected in second tab.

3.5.2 Complete view of all the three tabs in Expression Elements

The first tab contains functions, the second contains different types of functions, and the third contains all the functions in a particular type.

Click the *Paste* button to get any function in *Expression Box,* as shown in Fig. 3.5.2.



Expression Builder

Enter an Expression to define the calculated query field:
(Examples of expressions include [field1] + [field2] and [field1] < 5)

Abs(number)

Returns a value of the same type that is passed to it specifying the absolute value of a number.

Fig. 3.5.2. Three tabs in Expression Elements

3.5.3 Problem Scenario

Consider the case scenario discussed in section 3.3.4 database Apex Inventory Shipment. The Marketing Manager of Apex Ltd., found that there were few orders which were prepared on time, but could not be sent because of customer details are not available. By mistake, a sales executive while typing has put a wrong customer number. *Customer Number* general format is XXX-0000, where X represents character and 0 represents any digit. The executive has inserted five digits instead of four. List the names of all such customers.

Solution

For this query, again the *Sales Order* table needs to be used. The criteria would be formed using the text function *Len (),* length of *Customer Number* is normally 8 but would be 9 in this case. *Expression Builder* can be used to specify the criteria.

Steps for second creating query

1. Click *Create ribbon* -> *Queries* -> *Query Design* to open the *Query Design* window.
2. Select the *Sales Order* table from the *Show Table* window and click *Add*.
3. Select the *Sales Order Number Sold to Customer, Sales Date, Ship Date, Shipped Via* columns.
4. Right-click on the *Criteria* tab of the *Sold to Customer* column and select *Build* from the drop-down, as shown in Fig. 3.5.3. The build will open the *Expression Builder* window.

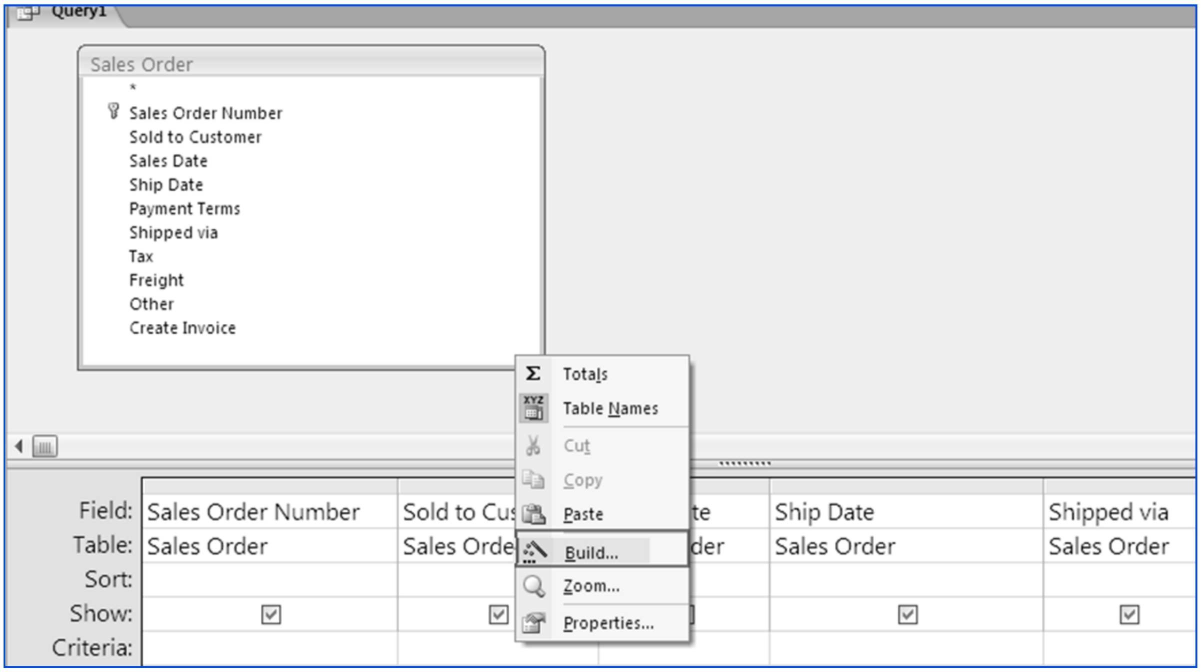
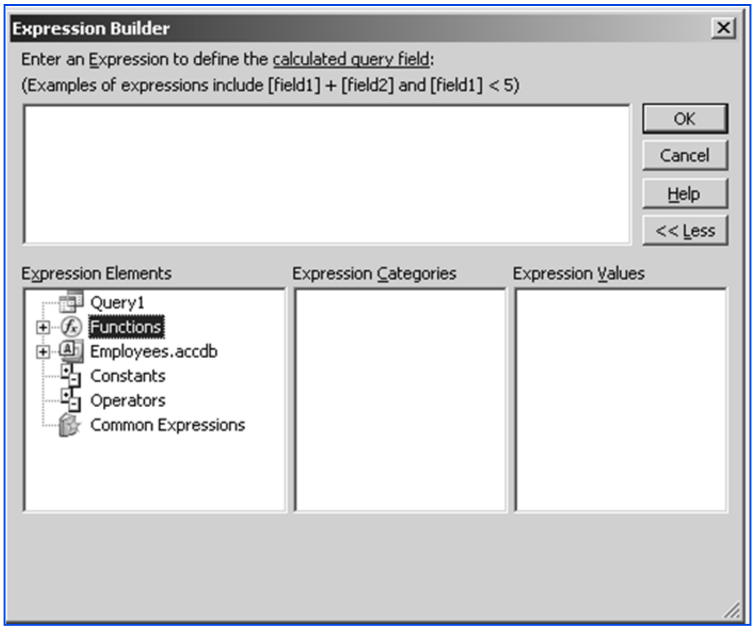


Fig. 3.5.3. Select build from drop down

5. The *Expression Builder* window appears, as shown in Fig. 3.5.4.

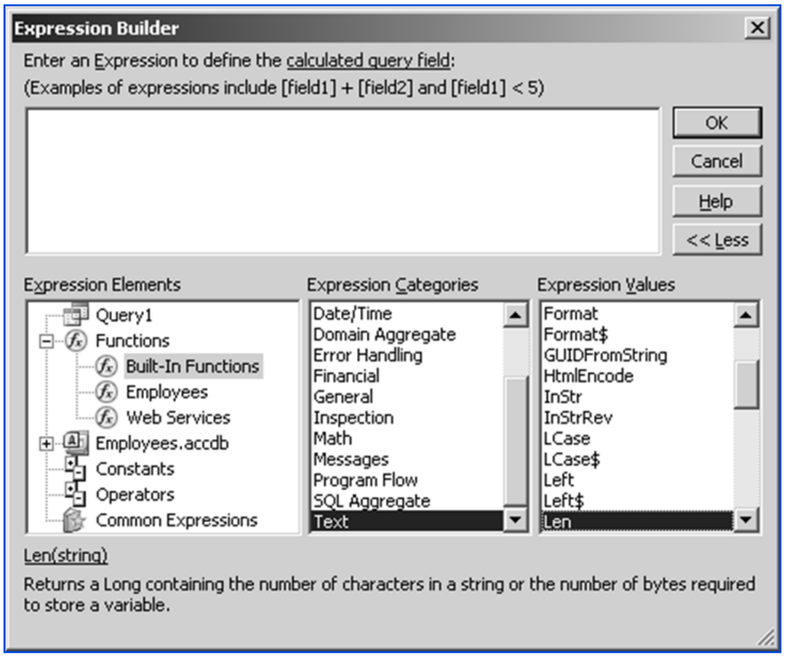


Expression Builder

Enter an Expression to define the calculated query field:
(Examples of expressions include [field1] + [field2] and [field1] < 5)

Fig. 3.5.4. Expression Builder window

6. The expression in the criteria should be *Len* ([Sold to Customer]) > 8**.** To include the length () function, select *Functions -> Built-In* functions from first tab, *Text* on the second tab, and *Len* on the third tab from *Expression Elements* in *Expression Builder,* as displayed in Fig. 3.5.5.



Expression Builder

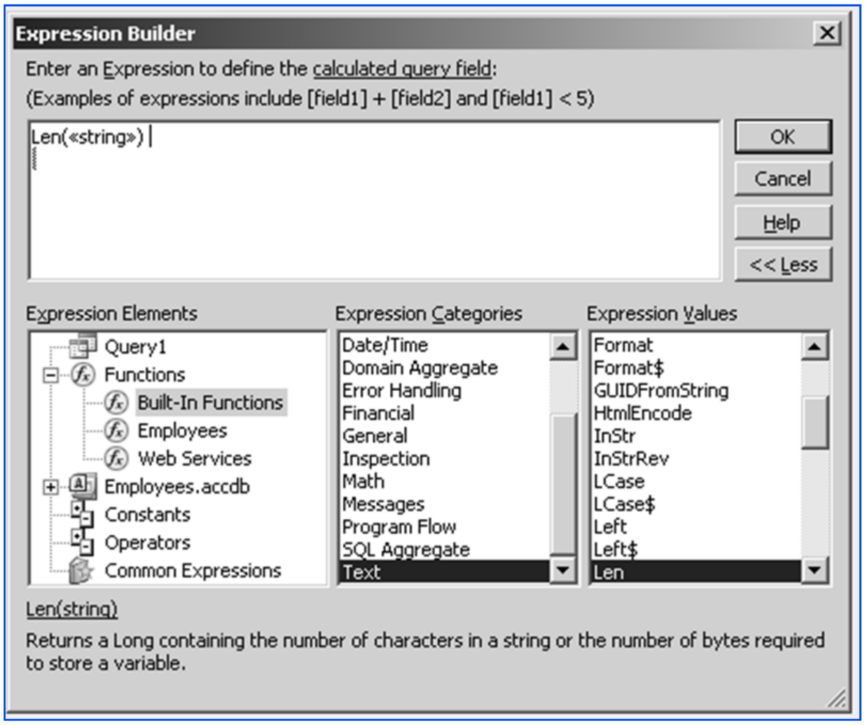
Enter an Expression to define the calculated query field:
(Examples of expressions include [field1] + [field2] and [field1] < 5)

Len(string)

Returns a Long containing the number of characters in a string or the number of bytes required to store a variable.

Fig. 3.5.5. Select Len function

7. Double Click the Len function to include expression in *Expression Box*, as shown as in Fig. 3.5.6.



Enter an Expression to define the calculated query field:
(Examples of expressions include [field1] + [field2] and [field1] < 5)

```
Len(«string»)
```

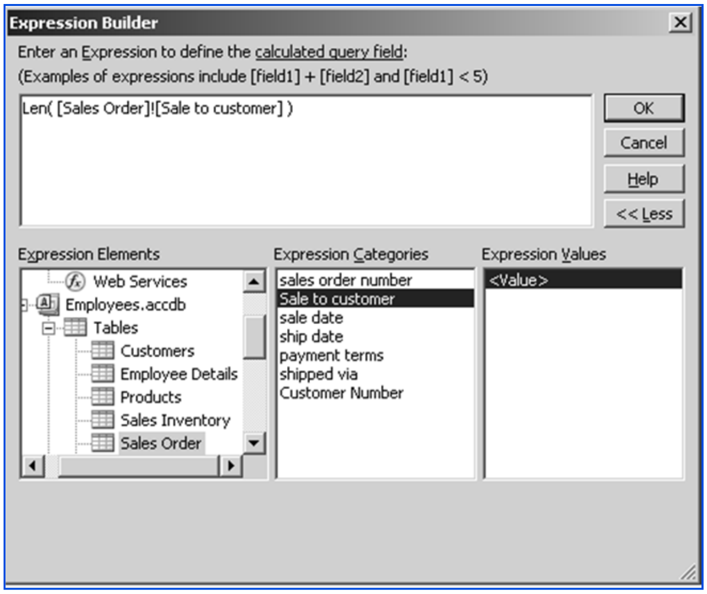
Len(string)

Returns a Long containing the number of characters in a string or the number of bytes required to store a variable.

Fig. 3.5.6. Paste the expression

8. Now the string which is a part of the *Len* () function must be replaced by field name. To do this select *Table -> Sales Order* from the first tab, *Sale to Customer* from the second tab in *Expression Elements* part of *Expression Builder*.

9. Select the *string* argument of the *Len ()* function and Double click to *Paste the argument in Len() function*. The column Name (Sale to Customer) will be pasted in place of *string*, the *Expression Builder* looks like as in Fig. 3.5.7.

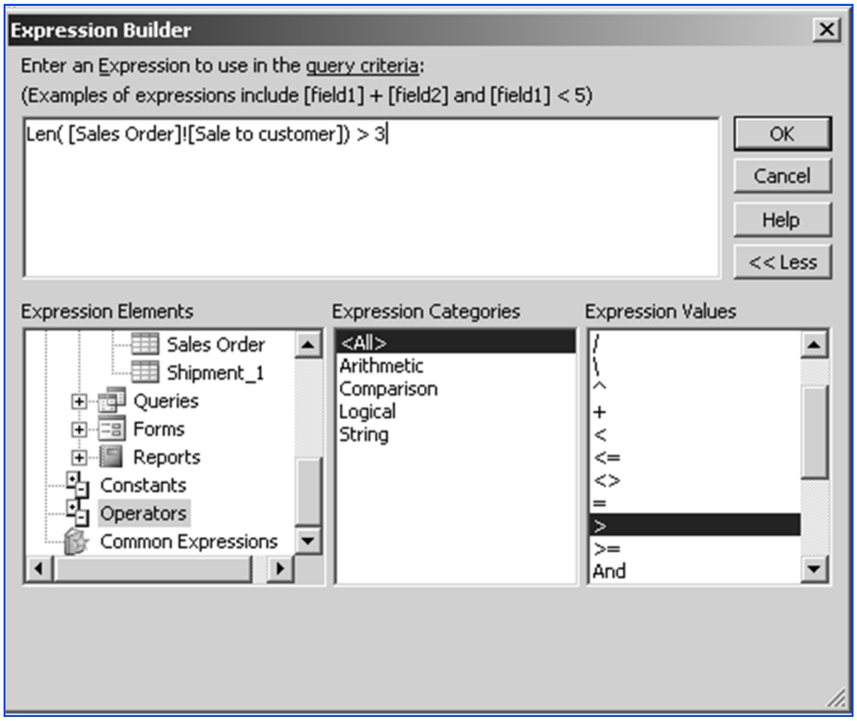


Enter an Expression to define the calculated query field:
(Examples of expressions include [field1] + [field2] and [field1] < 5)

Len([Sales Order]![Sale to customer])

Fig. 3.5.7. Place the required column as function argument

10. Now click on > (Greater Than sign) from the *Operators* tab so that it gets pasted on *Expression Box* and write 3 after that so the *Expression Builder* window looks like as in Fig. 3.5.8.

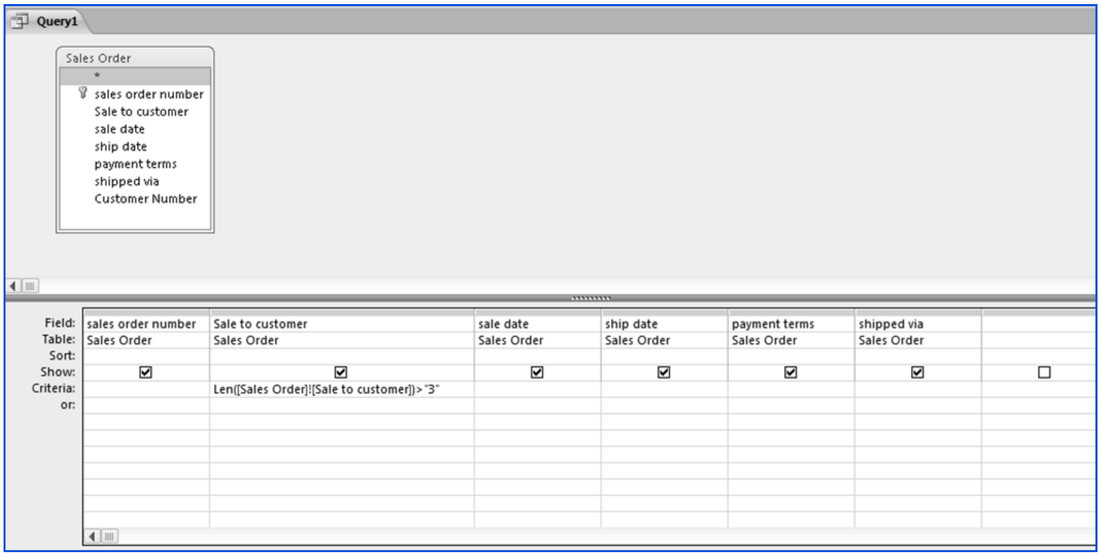


Enter an Expression to use in the query criteria:
(Examples of expressions include [field1] + [field2] and [field1] < 5)

Len([Sales Order]![Sale to customer]) > 3

Fig. 3.5.8. The Complete expression

11. Click *OK* to save the expression. Note the written expression appears in criteria of the *Sale to Customer* field, as shown in Fig. 3.5.9.

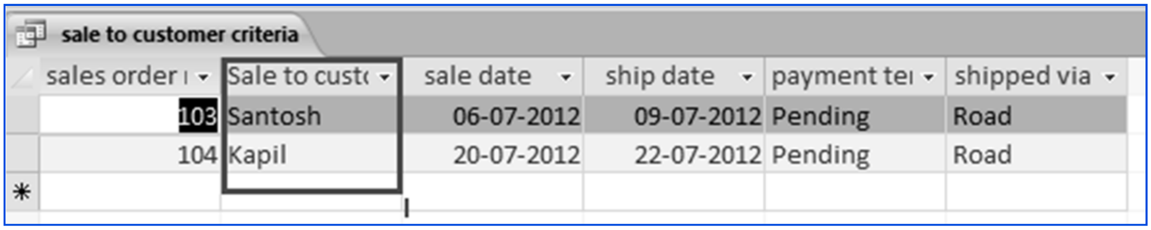


Field:	sales order number	Sale to customer	sale date	ship date	payment terms	shipped via	
Table:	Sales Order	Sales Order	Sales Order	Sales Order	Sales Order	Sales Order	
Sort:							
Show:	☑	☑	☑	☑	☑	☑	☐
Criteria:		Len([Sales Order]![Sale to customer])>"3"					
or:							

Fig. 3.5.9. Expression appears in Criteria

Click 💾 on the toolbar to save the query. Type the name of query as "Sale to customer Criteria".

12. Click ! in *Design ribbon* -> *Results* to view the results, as shown in Fig. 3.5.10.



sales order ı	Sale to custı	sale date	ship date	payment teı	shipped via
103	Santosh	06-07-2012	09-07-2012	Pending	Road
104	Kapil	20-07-2012	22-07-2012	Pending	Road
✱					

Fig. 3.5.10. Query Results

The *Expression Builder* makes it easier to remember function names and also avoids typing error in the name of the columns. The operators' option can be used to place all the available operators in criteria expression.

3.6 Summary

This chapter focused on creation of criteria expressions in Access 2010. The different components of the expression are – operators, operands, and functions. Operands can be considered as values used for performing operations. Operands can be further distinguished as literals, identifiers, and functions. Further, this chapter introduced different types of operators as – comparison, arithmetic, logical, and miscellaneous.

Functions are built in code to help us work better with expressions. There are different types of functions available in Access to work with different data types. Various functions available are - Text Function, Date and

Time Functions, Math Functions, and Financial Functions.

Expressions can be written directly or by using an interactive tool *Expression Builder* to create expression. *Expression Builder* has various tabs as *Expression Box*, *Operator Buttons*, and *Expression Elements*.

3.7 Lab Exercises

For *Apex Inventory Shipment* database of Apex Ltd. Provide a solution to the following problem scenarios:

1. A sales executive while inserting records in the Sales Order table by mistake inserted the Sales Date with a previous year. For instance, instead of 1/2/2012, he typed 1/2/2011. Write an Update query to add 1 year to all the dates which were written in previous year.
2. The Finance manager came to know that there had been a problem in receiving payments that were paid through Cheque or Electronic Card. Create a record set displaying all the invoices that were paid through Cheque, Master Card, or Visa Card.
3. For year-end review, the Vice President of the Company needs a report displaying a list of all the customers, and value and count of the orders placed by them.

   (To solve the above query create a Group By query on Customer, Sales Order, and Sales Item Description. Calculate the value of orders as Sum of Quantity and Price).
4. Considering the query in question 3, a customer named James having Customer Number as END-0010 wishes to pay his entire amount in monthly installments. Company charges an interest rate of 5% for the part payments. Calculate the amount that James has to pay as a monthly installment. (Use the PMT function).
5. Display a list of all the orders which have not been shipped. (Check for orders with NULL Shipped Date).
6. A sales person was trying to fetch all the customers from India from city – New Delhi, but he found that the city has been typed in many different ways. For instance New Delhi was also written as Delhi, N Delhi, or Old Delhi. Find all the records from table Customers which have country as India and contain ‘Delhi’ in city name.
7. The Marketing Manager defined a standard format of creating Customer Code as XXX-0000 where XXX are first three letters of Customer’s Company and 0000 is a four digit numeric code for customer. This format was not followed by some executives. Find all the customers for whom first three letters of the Customer Number does not match first three letters of their Company Name.
8. It was noticed that in the table Sales Item Description some values for Ship Qty are inserted as negative. Write a query to retrieve data having all the values of Ship Qty converted to positive and also the Price removing the decimal values. (Use math functions for the purpose).
9. Display a list of all the inventory items which are to be reordered. (Check all the Inventory which have Reorder Point greater than Inventory in Stock + Inventory on Order).
10. Considering the above query, retrieve a list of all the inventories which have not been ordered from past one month. (Make use of DateDiff and Date function)

Multiple Choice Questions

1. For the Apex Inventory Shipment database, a list of orders placed ten months ago or more from the Sales Order table needs to be created. What would be appropriate query criteria for the *Sales Date* field?

   (a) DateValue(DateAdd(“yyyy”,-3,[Sales Date]) > Today()

   (b) >= 3 Months

   (c) Between 3 And 5

   (d) DateAdd("m",3,[Sales Date])<Date()

2. Which record will be retrieved if the query criteria is < #1/1/11#?

   (a) All values less or more than 1,195

   (b) Value less than 95 characters

   (c) Records with date before 2011

   (d) All of above

3. The query criteria on *Shipped Date* for deleting all the records from the *Sales Order* table which were shipped before April 2011 or were never shipped would be ____________?

   (a) ">=#4/1/2011# OR IS NULL

   (b) Month() = April

   (c) IS NULL

   (d) None of the above

4. For the criteria BETWEEN 1/1/2011 and 12/31/2011, which rows will be displayed as result?

   (a) Display records between the dates 1/2/2011 and 1/1/2012

   (b) Display records between the dates 1/1/2011 and 12/31/2011

   (c) Display records whose dates equaled 1/1/2011 or 12/31/2011

   (d) All of the above

5. In the criteria expression Total Amount Paid: [Tax] + [Freight] + [Other], which value is an *Identifier*.

   (a) Total Amount Paid

   (b) [Tax]

   (c) +

   (d) All of the above

6. A report is needed to be prepared, checking all the orders that are pending to be shipped within one week. The criteria expression in *Shipped Date* can be?

(a) > Today()

(b) < DateAdd(“d”,7,”[Shipped Date])

(c) > Date() + 7

(d) = Now()

7. The tab in *Expression Builder* in which we write expressions is known as _____________.

(a) Expression Elements

(b) Operator Button

(c) Expression Box

(d) Expression Builder window

8. The criterion for the Category column in the *Inventory* table is IN (“Truck”, “Boat”, “Car”). This is equivalent to:

(a) [Category] Like “Truck”,”Boat”,”Car”

(b) [Category] = “Truck”,”Boat”,”Car”

(c) [Category] = “Truck” AND [Category] = ”Boat” AND [Category] = ”Car”

(d) [Category] = “Truck” OR [Category] = ”Boat” OR [Category] = ”Car”

9. Among +, <>, MOD, AND, <= operator which operator has the highest priority.

(a) +

(b) MOD

(c) AND

(d) <>

10. The criteria expression LIKE?a*.

(a) Will match all text starting from A

(b) Will match all text ending from A

(c) Will match all text having second character as A

(d) Will match all text starting with a digit

11. Which criteria would be used to find the records where the mode of payment is not known from Payments table?

(a) [How Paid] IS NULL

(b) [How Paid] = “”

(c) NOT IN [How Paid]

(d) None of the above

12. The criteria [Sales Date] > #1/1/2011# XOR [Payment Terms] = “Cheque” will return records only when …

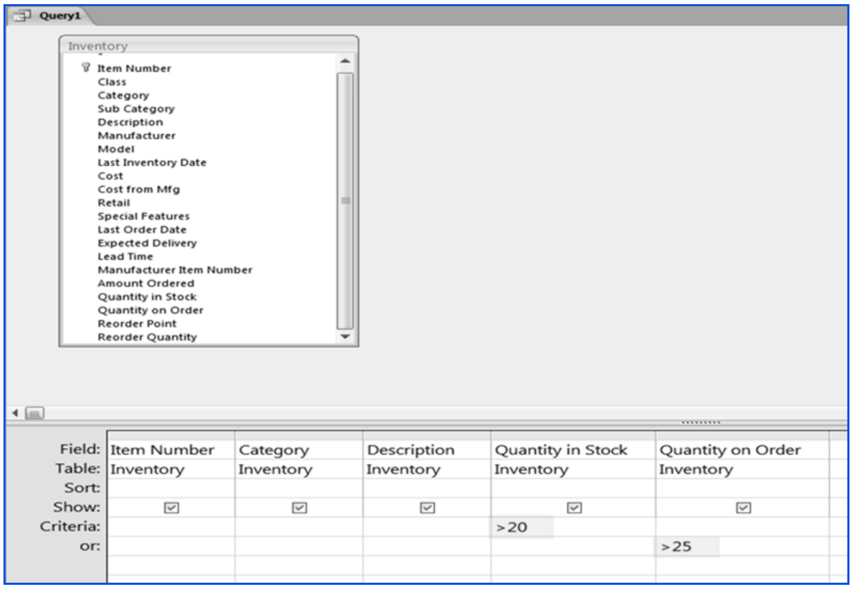
    (a) [Sales Date] is greater than 1-Jan-2011 And [Payment Terms] is “Cheque”

    (b) Either [Sales Date] is greater than 1-Jan-2011 Or [Payment Terms] is “Cheque”, but not both

    (c) [Sales Date] is greater than 1-Jan-2011 Or [Payment Terms] is “Cheque”

    (d) All of the above

13. Consider the query on the Inventory table:



Item Number
Class
Category
Sub Category
Description
Manufacturer
Model
Last Inventory Date
Cost
Cost from Mfg
Retail
Special Features
Last Order Date
Expected Delivery
Lead Time
Manufacturer Item Number
Amount Ordered
Quantity in Stock
Quantity on Order
Reorder Point
Reorder Quantity

Field:	Item Number	Category	Description	Quantity in Stock	Quantity on Order
Table:	Inventory	Inventory	Inventory	Inventory	Inventory
Sort:					
Show:	☑	☑	☑	☑	☑
Criteria:				>20	
or:					>25

This query will return:

(a) All records from Inventory table

(b) All records from Inventory table having Quantity in Stock greater than 20 AND Quantity on Order > 25.

(c) All records from Inventory table except having Quantity in Stock less than 20 and Quantity on Order > 25.

(d) All records from Inventory table having Quantity in Stock greater than 20 OR Quantity on Order > 25.

14. The Mid(“Apex Limited”,5,4) function will return

    (a) Apex

    (b) Ted

    (c) Limi

    (d) Apex Limited

15. The SGN() function is a ____________ type function:

(a) Math

(b) Financial

(c) Text

(d) Date & Time

16. The expression to combine first three characters of Customer First Name and last four characters of Sales Order Number placed by customer will be:

(a) [First Name] + [Sales Order Number]

(b) [First Name] & [Sales Order Number]

(c) Left([First Name],3) + Right([Sales Order Number],4)

(d) Left([First Name]) + Right([Sales Order Number])

17. The expression 3\4 will return

(a) 0.75

(b) ¾

(c) 0

(d) None of the above

18. The Instr(4, “XXpXXpXXPXXP" , “P”) function will return

(a) 3

(b) 6

(c) 9

(d) None

19. Which of the following is not a Date Time function?

(a) Today()

(b) Date()

(c) Now()

(d) MonthName()

20. A query with compound criteria, where both the criteria are written on the same row of two different fields. This will be creating______________ compound criteria?

(a) OR compound criteria

(b) AND compound criteria

(c) XOR compound criteria

(d) EQV compound criteria

CHAPTER 4 MACROS AND SWITCHBOARDS

LEARNING OBJECTIVES

- ■ Simplifying Task with Macros
  - ❑ Creating a Macro
  - ❑ Attaching a Macro
  - ❑ Restricting Records Using a Condition
  - ❑ Validating Data Using a Macro
  - ❑ Automating Data Entry Using a Macro
- ■ Managing Switchboards
  - ❑ Creating a Database Switchboard
  - ❑ Modifying a Database Switchboard
  - ❑ Setting the Startup Options

4.1 Introduction

A macro allows you to automate tasks and add functionality to your forms, reports, and controls. Access macros let you perform defined actions and add functionality to your forms and reports. Macros in Access can be thought of a graphical and a simpler way to do programming. Every macro has a list of actions and arguments defined for each action. Macros can be used independently or attached to a form, report, or control events. Arguments provided in a macro can restrict, validate, or automate data entry. Microsoft Access 2010 has added new features to macros to eliminate the need to writing VBA code.

Switchboards are forms available in Access 2010 to present data in the form so that users can focus on using the database as intended. A switchboard form presents the user with a limited number of choices for working with the application and makes the application easier to use. For example, a switchboard may give choice to open the tables, forms, and open or print reports.

4.2 Simplifying Task with Macros

In Access, *macros* can be considered as a simplified version of VBA programming. *Macro*s are used to execute any task that can be initiated with the keyboard or the mouse. *Macro*s are written by specifying a list of to perform and providing arguments for these actions. Access 2010 provides enables *macro*s so that they can automate responses to many types of events (events can be change in the data, the opening or closing of a form or a report, or even a change of focus from one control to another) without forcing actually using a programming language.

4.2.1 Creating a Macro

In Access, *macro*s can be created using the *Macro Design* window. This window can be opened by clicking *Create* → *Other* → *Macro,* as shown in Fig. 4.2.1

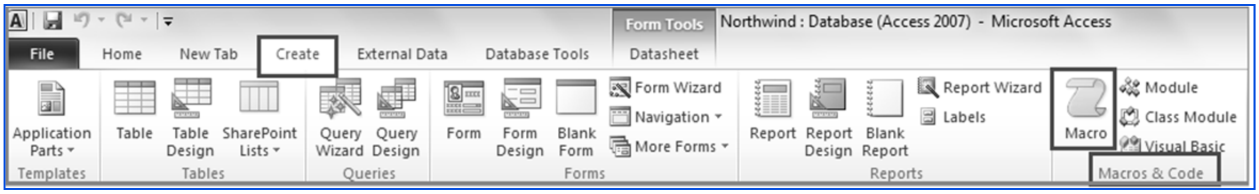


Fig. 4.2.1 Create New Macro

The *Macro Design* window is displayed in Fig. 4.2.2.

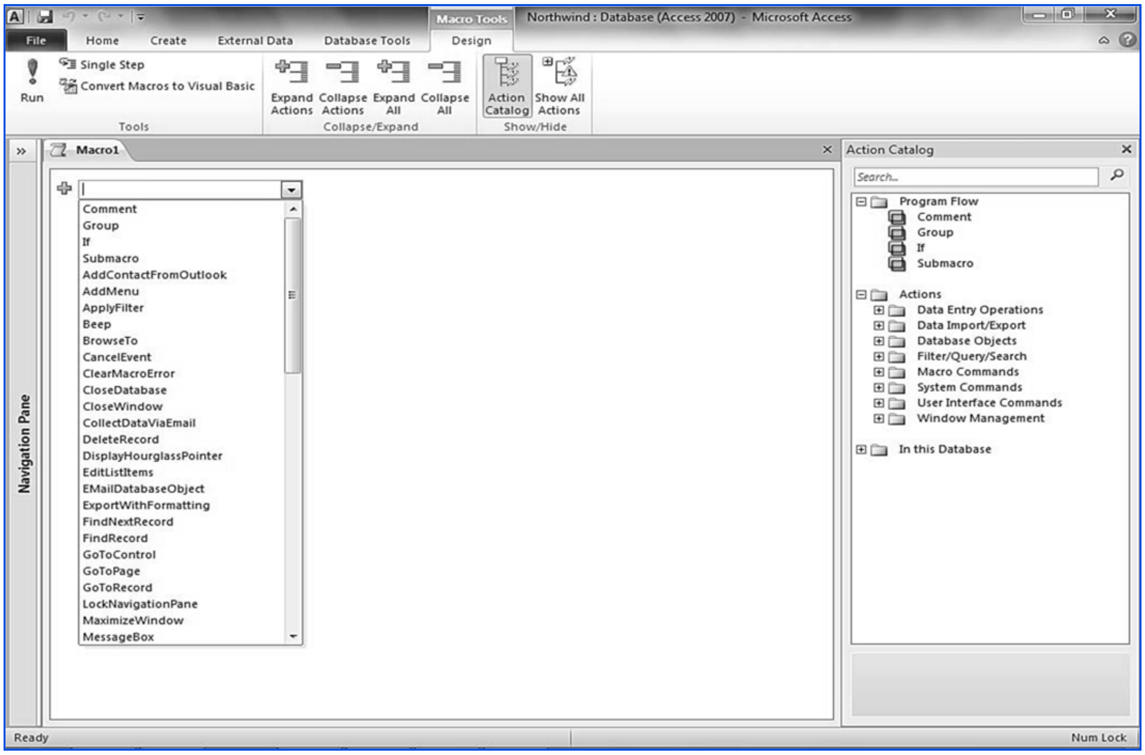
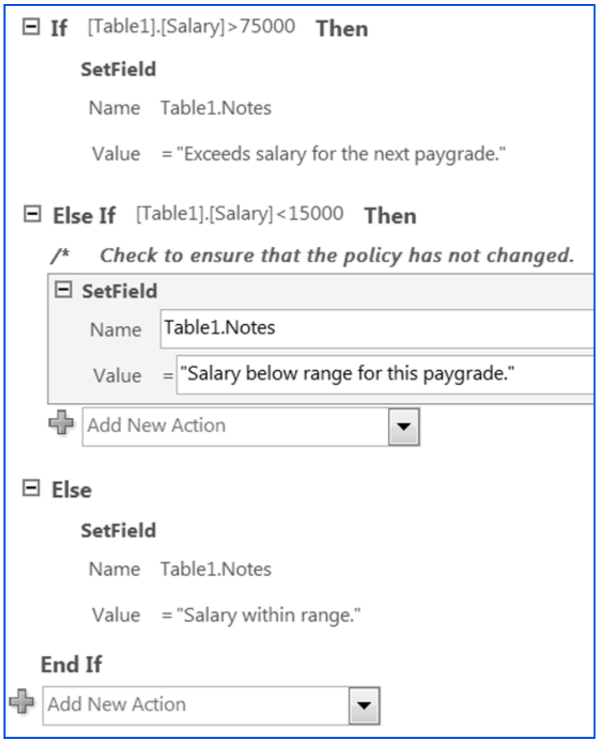


Fig. 4.2.2. Macro Designer Window

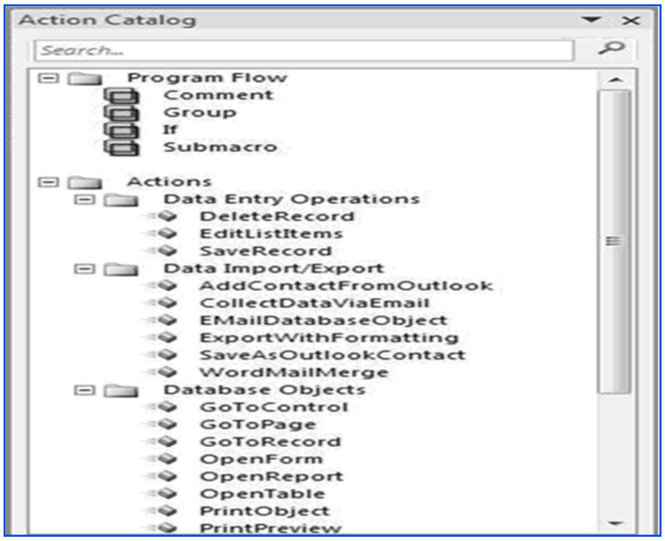
In the new Macro Designer for Access 2010, the layout more closely resembles a text editor. Actions and conditional statements displayed in a familiar top-down format that is used by programmers. Arguments are displayed inline in a dialog box as shown in Figure.4.2.3



```
⊟ If   [Table1].[Salary]>75000   Then
        SetField
          Name    Table1.Notes
          Value   = "Exceeds salary for the next paygrade."

⊟ Else If   [Table1].[Salary]<15000   Then
    /*    Check to ensure that the policy has not changed.
    ⊟ SetField
          Name    Table1.Notes
          Value = "Salary below range for this paygrade."
    Add New Action

⊟ Else
        SetField
          Name    Table1.Notes
          Value   = "Salary within range."

  End If
Add New Action
```

Fig. 4.2.3: Macro Designer Window

Adding a new action or conditional statement is simple. Either select it from the actions drop-down list, from a right-click menu, or select it from the **Action Catalog** pane as shown in Fig. 4.2.4 to the right side of the Macro Designer.



```
Action Catalog
Search...
⊟ Program Flow
      Comment
      Group
      If
      Submacro
⊟ Actions
   ⊟ Data Entry Operations
         DeleteRecord
         EditListItems
         SaveRecord
   ⊟ Data Import/Export
         AddContactFromOutlook
         CollectDataViaEmail
         EMailDatabaseObject
         ExportWithFormatting
         SaveAsOutlookContact
         WordMailMerge
   ⊟ Database Objects
         GoToControl
         GoToPage
         GoToRecord
         OpenForm
         OpenReport
         OpenTable
         PrintObject
         PrintPreview
```

Fig. 4.2.4: Action Catalog

Actions Catalog: Actions are the basic building blocks of *macro*s. MS Access 2010 provides a Action Catalog pane that contained large list of actions to be chosen, enabling a wide range of commands that can be performed. Some of the commonly used actions are: open a report, find a record, display a message box, or apply a filter to a form or report. A list of some commonly used actions in *macro* is displayed in Table 4.2.

Action	Description
ApplyFilter	Applies a filter or query to a table, form, or report.
CancelEvent	Cancels the event that caused the macro to run.
Close	Closes the specified window or the active window, if none is specified.
CopyObject	Copies the specified database object to a different Microsoft Access database or to the same database with a new name.
DeleteObject	Deletes the specified object or the object selected in the *Database* window, if no object is specified.
Echo	Hides or shows the results of a macro while it runs.
FindNext	Finds the next record that meets the criteria specified with the most recent *FindRecord* action or the *Find* dialog box. Use to move successively through records that meet the same criteria.
FindRecord	Finds the first or next record that meets the specified criteria. Records can be found in the active form or datasheet.
GoToControl	Selects the specified field on the active datasheet or form.
GoToPage	Selects the first control on the specified page of the active form.
GoToRecord	Makes the specified record the current record in a table, form, or query. Use to move to the first, last, next, or previous record.
Maximize	Maximizes the active window.
Minimize	Minimizes the active window.
MoveSize	Moves and/or changes the size of the active window.
MsgBox	Displays a message box containing a warning or informational message.
OpenForm	Opens a form in the *Form* view, *Design* view, *Print Preview*, or *Datasheet* view.
OpenModule	Opens the specified *Visual Basic* module in the *Design* view.
OpenQuery	Opens a query in the *Datasheet* view, *Design* view, or *Print Preview*.
OpenReport	Opens a report in the *Design* view or *Print Preview* or prints the report immediately.
OpenTable	Opens a table in the *Datasheet* view, *Design* view, or *Print Preview*.
OutputTo	Exports the specified database object to a Microsoft Excel file (.xls), rich-text file (.rtf), text file (.txt), or HTML file (.htm).
PrintOut	Prints the active database object. You can print datasheets, reports, forms, and modules.
Quit	Quits Microsoft Access.

Rename	Renames the specified object.
Requery	Forces a re-query of a specific control on the active database object.
Restore	Restores a maximized or minimized window to its previous size.
RunApp	Starts another program, such as Microsoft Excel or Word.
RunCode	Runs a Visual Basic Function procedure.
RunCommand	Runs a command from Microsoft Access's menus. For example, *File* ⟶ *Save*.
RunMacro	Runs a macro.
RunSQL	Runs the specified SQL statement for an action query.
Save	Saves the specified object or the active object, if none is specified.
SelectObject	Selects a specified database object. You can then run an action that applies to that object.
SendObject	Sends the specified database objects as an attachment in an e-mail.
SetValue	Sets the value for a control, field, or property on a form or report.
SetWarnings	Turns all system messages on or off. This has the same effect as clicking *OK* or *Yes* in each message box.
StopAllMacros	Stops all currently running macros.
StopMacro	Stops the currently running macro. Use to stop a macro when a certain condition is met.
TransferDatabase	Imports or exports data to or from the current database from or to another database.
TransferSpreadsheet	Imports data from a spreadsheet file into the current database or exports data from the current database into a spreadsheet file.
TransferText	Imports data from a text file into the current database or exports data from the current database into a text file.

Table 4.1: Macro Actions

4.2.1.1 Problem Scenario

Kanika Mathur, a sales executive in Apex Ltd. is required to send a detailed report of all the orders that have been placed today to the Sales Head every evening in a form of an Excel sheet, also the same sheet also has to be uploaded in a shared folder for delivery to check. She thought to automate the process of transferring records so as to save the efforts required.

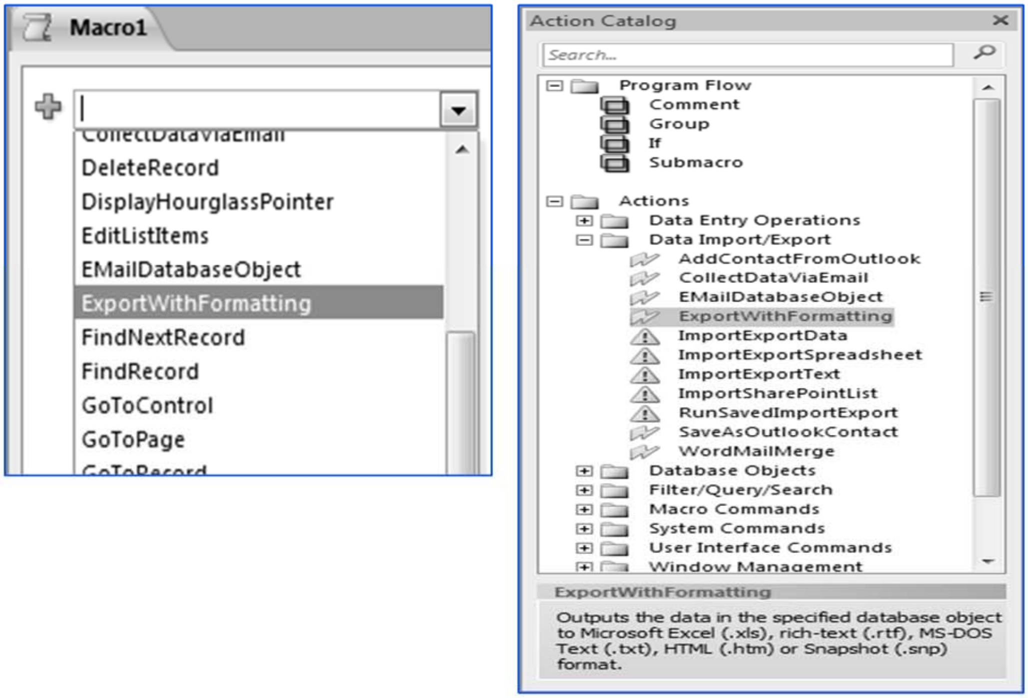
Solution

As a solution to the above problem, create a *macro* that transfers the table Orders into Excel and stores it in a shared folder using the *TransferSpreadSheet* Action. Then, use *Send Object Action* to mail it to the Sales Head.

Steps for creating the macro

1. Open *Macro Designer*. Click *Create* → *Macro & Code* → *Macro*.

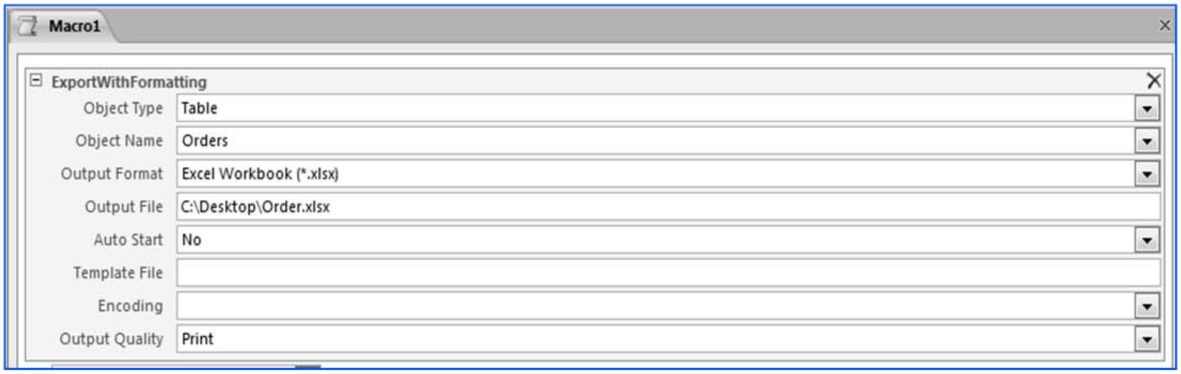
2. Click **AddNew Action** drop-down box → Select ***ExportWithFormatting***.Note that the **ExportWithFormatting** option will also be available in Action Catalog. The *Macro Designer* window appears, as shown in Fig. 4.2.5.



Outputs the data in the specified database object to Microsoft Excel (.xls), rich-text (.rtf), MS-DOS Text (.txt), HTML (.htm) or Snapshot (.snp) format.

Fig. 4.2.5 : Select Export with Formatting Action

3. When we select an action that needs more than one argument, it would appear, followed by a box for each argument as shown in Fig. 4.2.6.



ExportWithFormatting

Object Type	Table
Object Name	Orders
Output Format	Excel Workbook (*.xlsx)
Output File	C:\Desktop\Order.xlsx
Auto Start	No
Template File	
Encoding	
Output Quality	Print

Fig. 4.2.6 : Box for each argument and type necessary values

Object Type	:	Table
Object Name	:	Orders
Output Format	:	Excel Workbook (*.xlsx)

Output File	:	C:\Desktop\Order.xlsx

NOTE: Output File Name should include the complete absolute path of the shared folder. *Macro Designer* appears, as shown in Fig. 4.2.6.

4. Select the next action in the *Query Designer* window as *EmailDatabaseObject* and specify *Action Arguments*, as shown below in Fig. 4.2.5:

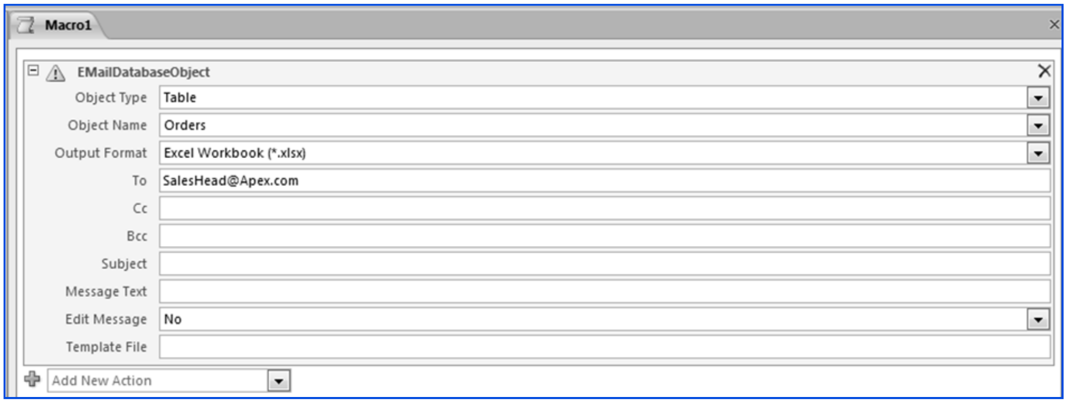
5. *Fig. 4.4: Action SendObject*

Object Type	:	Table
Object Name	:	Orders
Output Format	:	Excel Workbook (*.xlsx)
To	:	SalesHead@Apex.com <email of Sales Head>
Cc	:	
Bcc	:	
Subject	:	<Subject line for the mail>
Message Text	:	<Message to be sent along with attachment>
Edit Message	:	No (Select Yes to edit message before sending>
Template File	:	<template file for output to be generated>

NOTE: The *EmailDatabaseObject* action can only be used if the Outlook is configured and is open. The To, *Cc*, *Bcc*, *Subject*, *Message Text* options can be set according to the requirement.

6. Click *Quick Access* → 💾. The window prompts for the name of the *macro*, write the name of *macro*s as MailCurrentOrders and click *OK*. Close the *Macro Design* window.

7. Double-click the *Macro* Name under the *Macro* option in the *All Access Objects* tab to execute it.



EMailDatabaseObject

Object Type	Table
Object Name	Orders
Output Format	Excel Workbook (*.xlsx)
To	SalesHead@Apex.com
Cc	
Bcc	
Subject	
Message Text	
Edit Message	No
Template File	

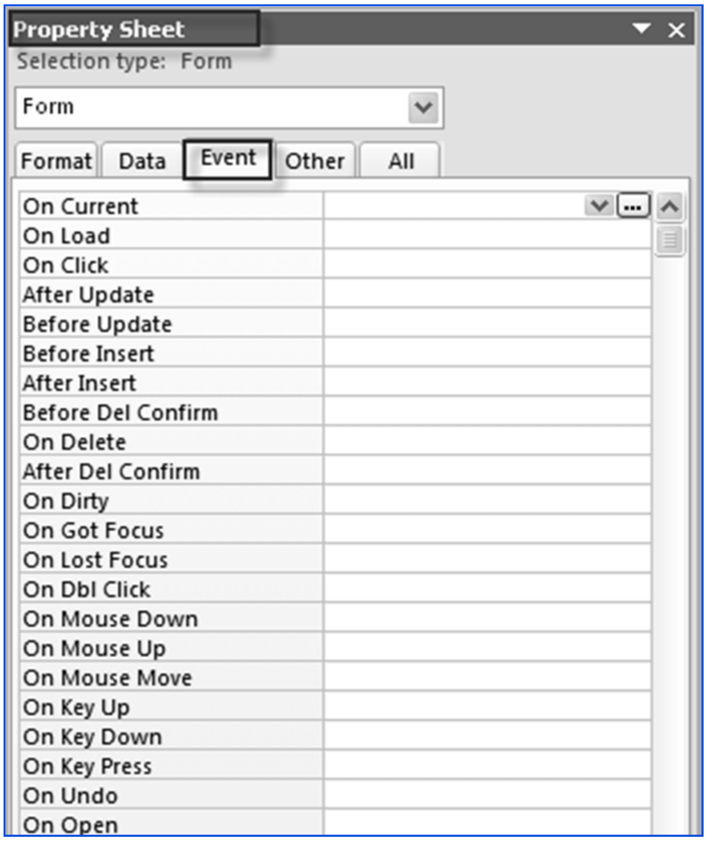
Fig. 4.2.7: EmailDatabaseObject Macro

All Macros are saved automatically in default name.

Macro Builder can use many actions in a single *macro* or create multiple *macro*s in one *macro designer* by using *macro names*. *Macro* can also be assigned shortcut keys using the *AutoKeys* macro.

4.2.2 Attaching a Macro

A *macro* can be attached to an event of a control, a form, or a report. *Macro*s are used to provide an added functionality to Access Objects. *Macro*s can automate responses to many types of *events* without using a programming language. *Events* are the property of a form or a control. *Macro*s are always attached to one event or another to perform some action. To view the events for a control, open its *Property* window and move to the *Events* tab as shown in Fig. 4.2.8. For example, if a button on a form is required to print a report, a *macro* which opens a report in print preview from can be attached to click event of the button. Access permits to create *macro*s embedded to the control or attach an existing *macro* to events.



Selection type: Form

Form

On Current
On Load
On Click
After Update
Before Update
Before Insert
After Insert
Before Del Confirm
On Delete
After Del Confirm
On Dirty
On Got Focus
On Lost Focus
On Dbl Click
On Mouse Down
On Mouse Up
On Mouse Move
On Key Up
On Key Down
On Key Press
On Undo
On Open

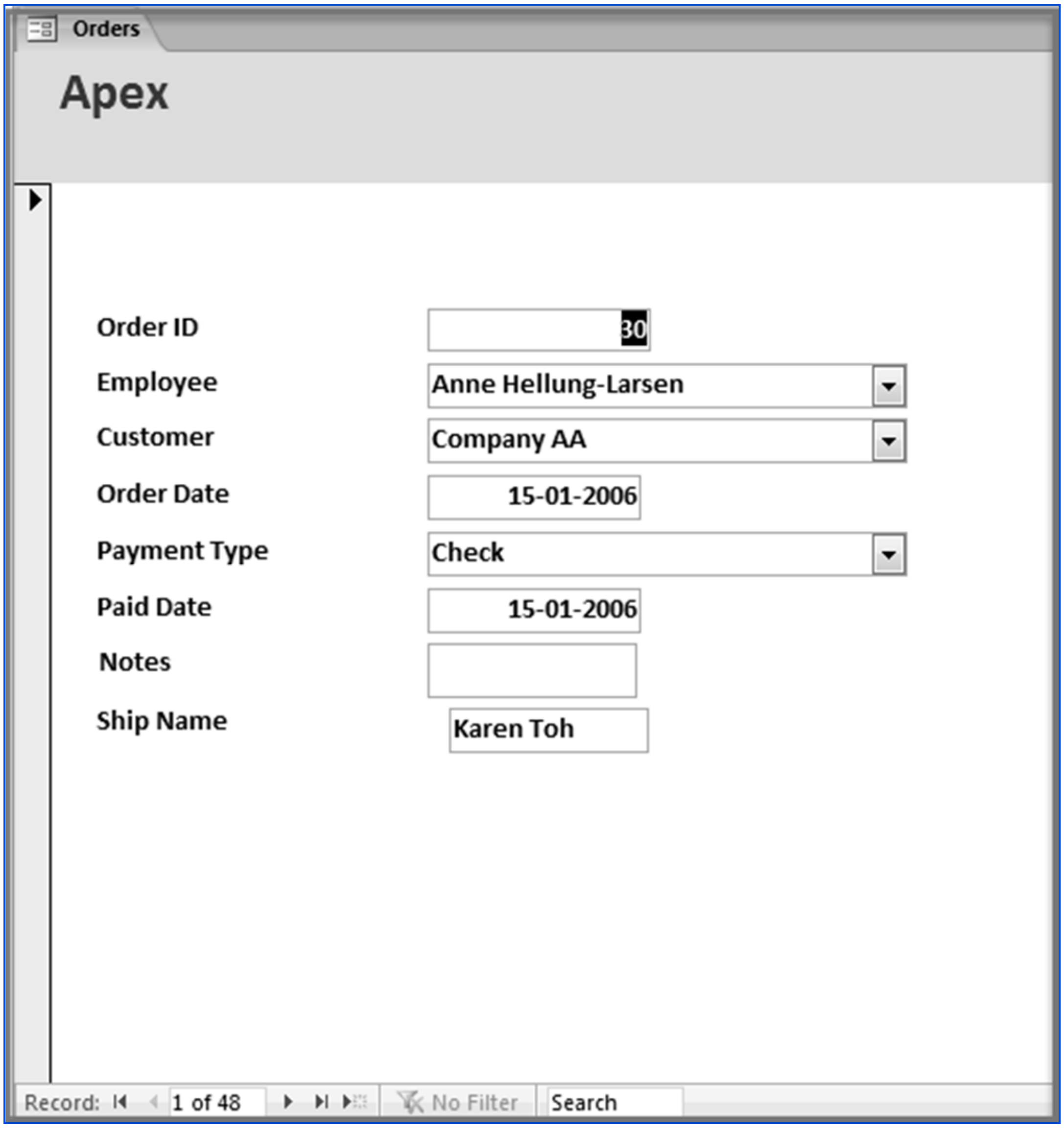
Fig. 4.2.8: Events in Form Properties

4.2.2.1 Problem Scenario

Consider the problem scenario discussed in section 4.2.1. Now the same functionality is required to be added to the *Orders* form. This means that the *Orders* form should have a button that enables the user to export and mail the table.

Solution

Consider the form *frmOrders* which is based on *Orders* table, as shown in Fig. 4.2.9.



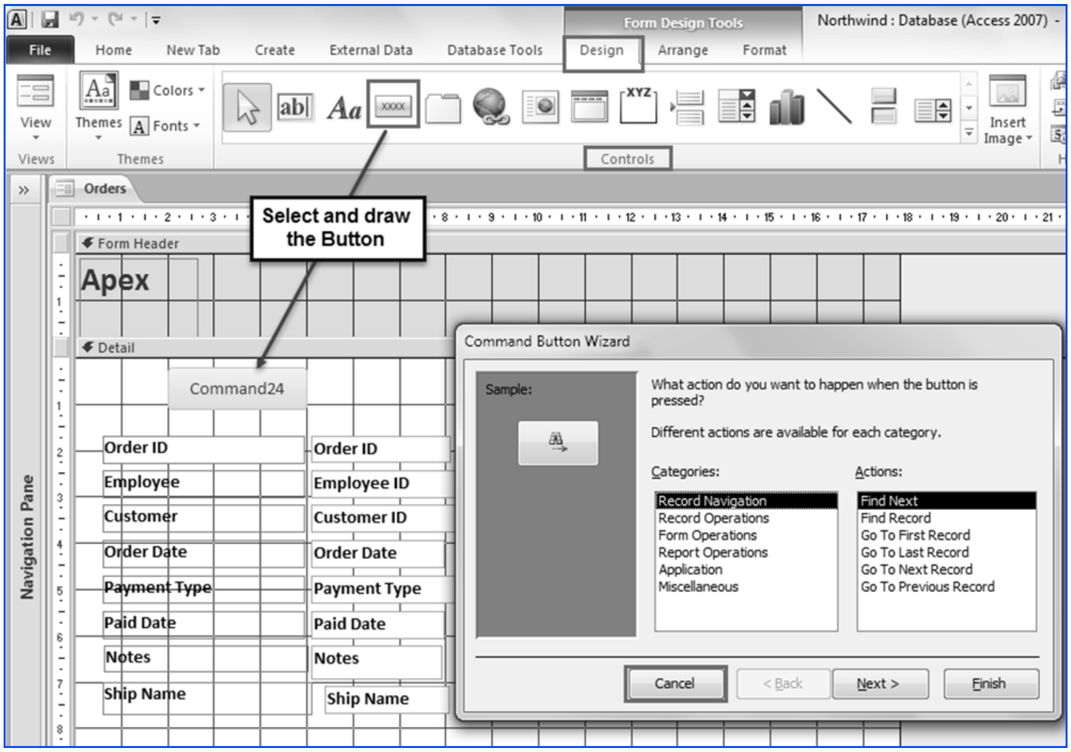
Order ID	30
Employee	Anne Hellung-Larsen
Customer	Company AA
Order Date	15-01-2006
Payment Type	Check
Paid Date	15-01-2006
Notes	
Ship Name	Karen Toh

Fig. 4.2.9: Orders Form

Now, add a *button* on the form and attach a *macro* to the click event of the *button*. *Macro* is the same as created in section 4.2.9 and exports and mails the list.

1. Open the *Orders* form in the *Design* view. To do this, right-click the form and select *Design View* from the list.

2. To add a *button* to the form, select *Design* -> *Controls* → *button* to be added and draw it on the form, as shown in Fig. 4.2.10. Cancel the *Command Button* wizard that appears.



Command Button Wizard

Sample:

What action do you want to happen when the button is pressed?

Different actions are available for each category.

Categories:
Record Navigation
Record Operations
Form Operations
Report Operations
Application
Miscellaneous

Actions:
Find Next
Find Record
Go To First Record
Go To Last Record
Go To Next Record
Go To Previous Record

Fig. 4.2.10: Draw the button on the form

NOTE: The *Command Button* wizard may not appear if *Use Control* -> *Controls* in *Design* is not selected.

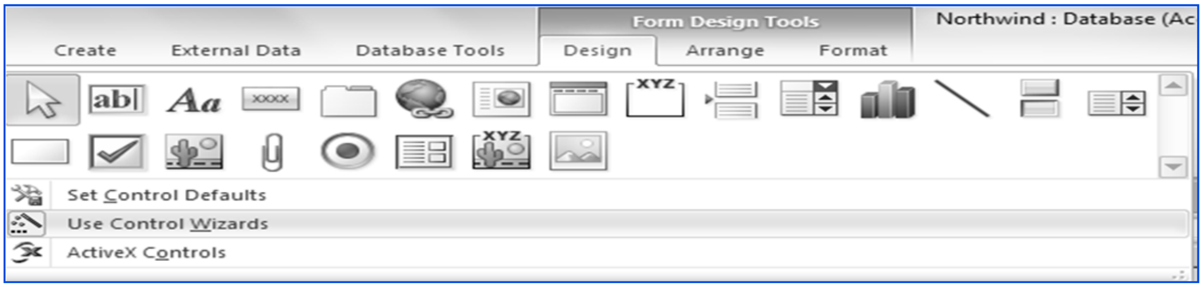


Fig. 4.2.11: Use Control Wizards option

3. Open the control properties by selecting the *Command* button and press *F4*. Alternatively, select *Property Sheet* from *Design -> Tools,* as shown in Fig. 4.2.12.

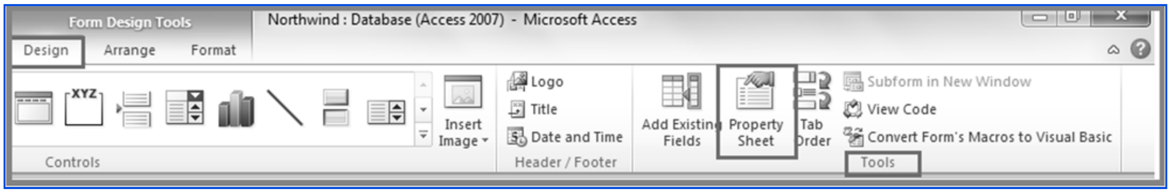
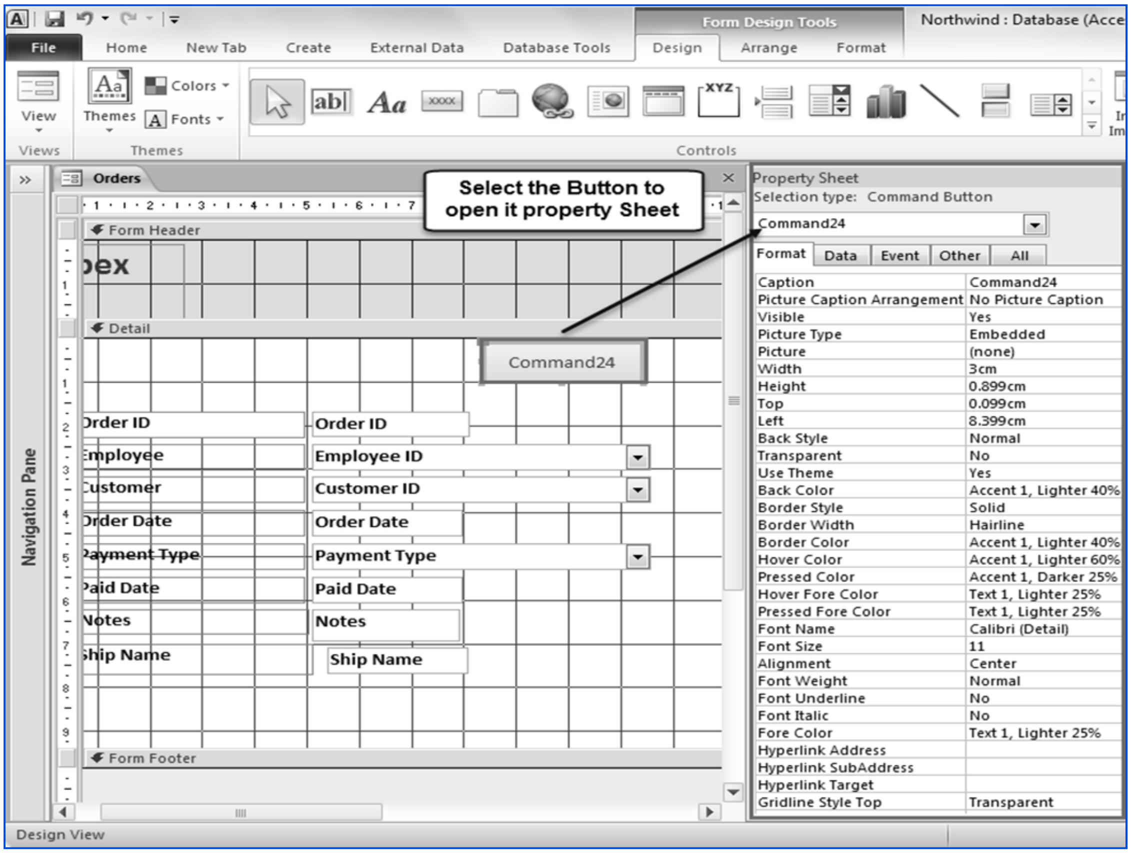


Fig. 4.2.12: Select Property Sheet

4. The *Property window* appears as shown in Fig. 4.2.13.



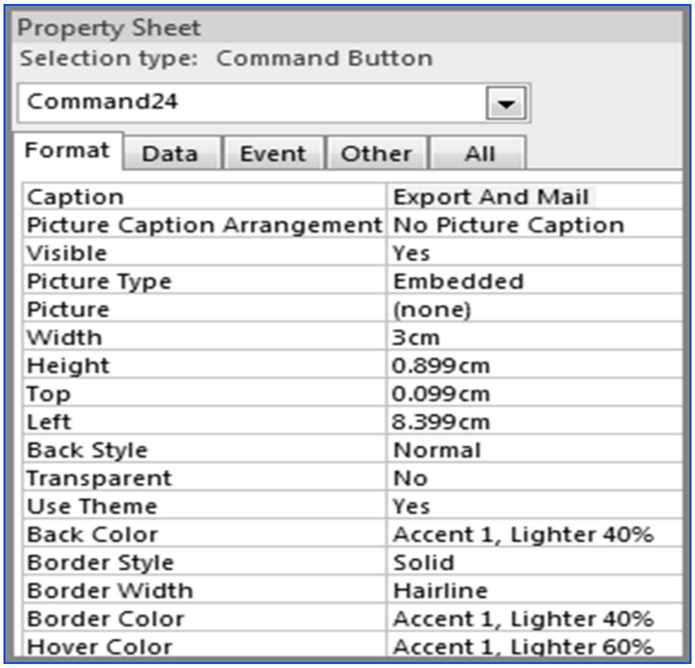
Select the Button to open it property Sheet

Property Sheet
Selection type: Command Button
Command24
Format | Data | Event | Other | All

Caption	Command24
Picture Caption Arrangement	No Picture Caption
Visible	Yes
Picture Type	Embedded
Picture	(none)
Width	3cm
Height	0.899cm
Top	0.099cm
Left	8.399cm
Back Style	Normal
Transparent	No
Use Theme	Yes
Back Color	Accent 1, Lighter 40%
Border Style	Solid
Border Width	Hairline
Border Color	Accent 1, Lighter 40%
Hover Color	Accent 1, Lighter 60%
Pressed Color	Accent 1, Darker 25%
Hover Fore Color	Text 1, Lighter 25%
Pressed Fore Color	Text 1, Lighter 25%
Font Name	Calibri (Detail)
Font Size	11
Alignment	Center
Font Weight	Normal
Font Underline	No
Font Italic	No
Fore Color	Text 1, Lighter 25%
Hyperlink Address	
Hyperlink SubAddress	
Hyperlink Target	
Gridline Style Top	Transparent

Fig. 4.2.13: Property Sheet for Button

5. In the *Format* tab of *Property Sheet*, set the value of the *Caption* property to *Export And Mail* as shown in Fig. 4.2.14.



Property Sheet

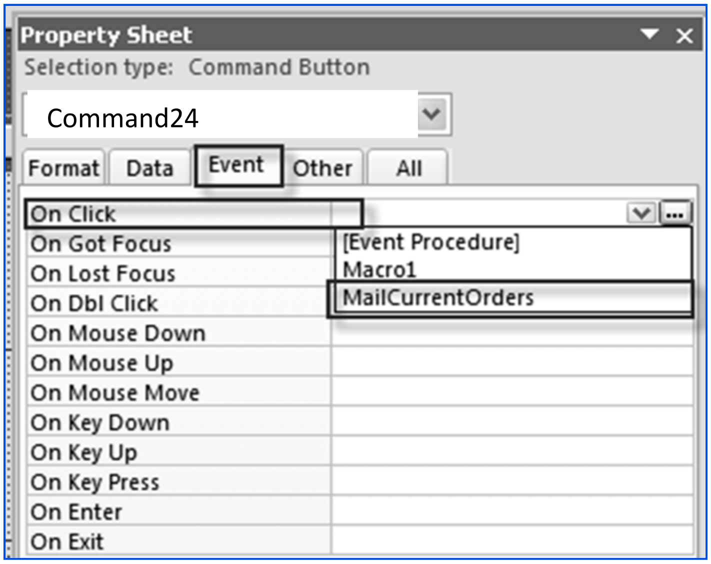
Selection type: Command Button

Command24

Caption	Export And Mail
Picture Caption Arrangement	No Picture Caption
Visible	Yes
Picture Type	Embedded
Picture	(none)
Width	3cm
Height	0.899cm
Top	0.099cm
Left	8.399cm
Back Style	Normal
Transparent	No
Use Theme	Yes
Back Color	Accent 1, Lighter 40%
Border Style	Solid
Border Width	Hairline
Border Color	Accent 1, Lighter 40%
Hover Color	Accent 1, Lighter 60%

Fig. 4.2.14: Set Caption Property

6. In the *Event* tab of the *Property* Sheet, set the value of the *On Click* event. To do this, click ▼ to open the list and select *MailCurrentOrders*, as shown in Fig. 4.2.15.



Property Sheet

Selection type: Command Button

Command24

On Click	[Event Procedure] Macro1 MailCurrentOrders
On Got Focus	
On Lost Focus	
On Dbl Click	
On Mouse Down	
On Mouse Up	
On Mouse Move	
On Key Down	
On Key Up	
On Key Press	
On Enter	
On Exit	

Fig. 4.2.15: Set the On Click event Property

NOTE: A *macro* embedded to the control can also be created, by clicking on ... button and selecting *Macro Builder* from the *Choose Builder* window that appears, as shown in Fig. 4.2.16.

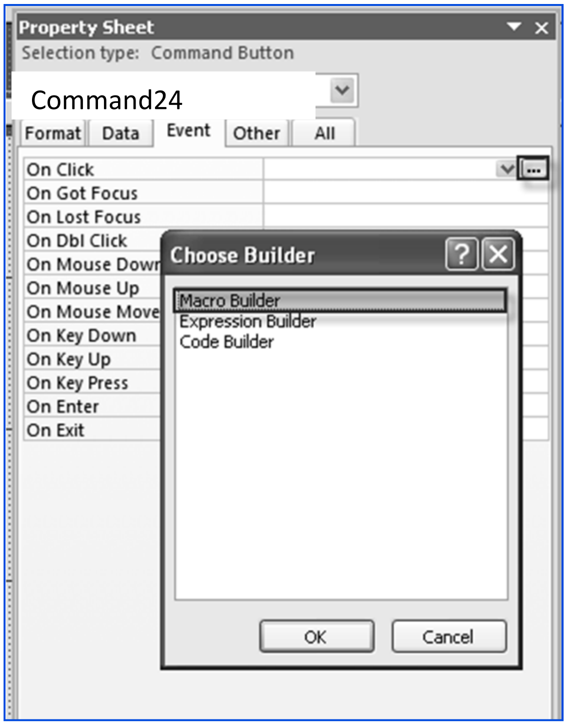
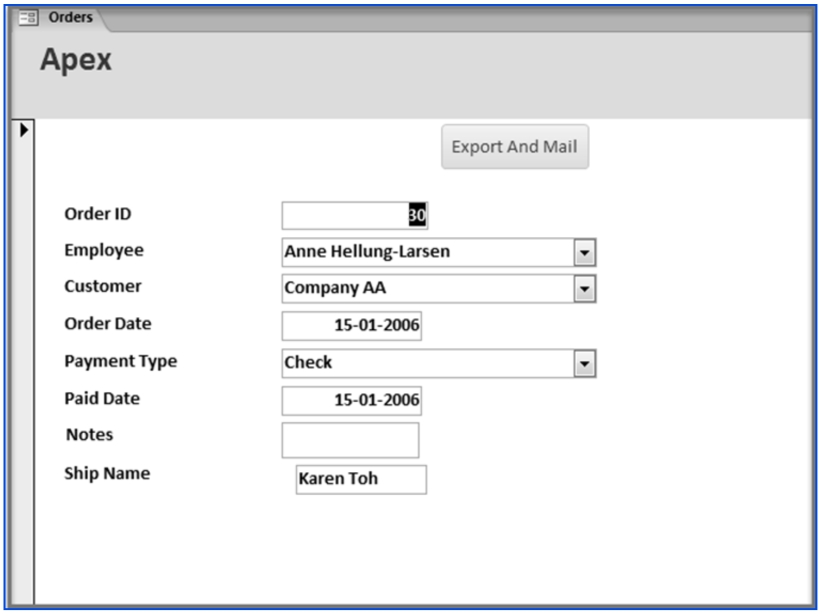


Fig. 4.2.16: Create an embedded Macro

7. The form appears as shown in Fig. 4.2.17; the *macro* will be executed on clicking the *Export And Mail* button.



Order ID	30
Employee	Anne Hellung-Larsen
Customer	Company AA
Order Date	15-01-2006
Payment Type	Check
Paid Date	15-01-2006
Notes	
Ship Name	Karen Toh

Fig. 4.2.17: Click Button to execute the Macro

Multiple *macros* can be created in a single *Macro Designer* window by giving each *macro* a different name and can be attached to an event of form or control by specifying their name.

4.2.3 Restricting Records using a Condition

A *macro condition* is an expression that enables a *macro* to perform certain tasks only if a specific situation exists. When a *condition* is used in a *macro*, the *macro* performs a defined set of tasks depending on whether the expression returns the True or False value. When the expression returns True, all the actions are performed. When the expression returns False, none of the actions are performed. Conditions can be entered in the *Conditions* column of the *Macro Builder* window. A single condition can control more than one action.

4.2.3.1 The Where Condition

The *Where* condition filters and selects records in reports or forms and their underlying tables or queries. This condition is applicable as an argument for the *macro* actions *OpenForm* and *OpenReport*. The *Where* condition is an *Action Argument* for *macro actions* such as *OpenForm* or OpenReport. For example, the *Where* condition specified for an *OpenForm* action can be used to compare and display matching records from two related forms.

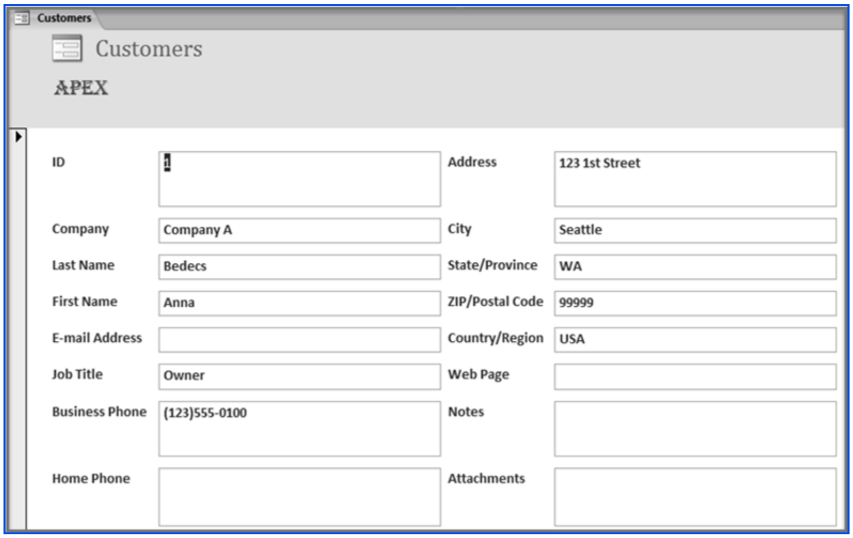
4.2.3.2 Problem Scenario

In the *Database Apex Inventory Shipment*, as an enhancement to the *Orders* form created in Section 4.2.2, the Manager wishes to see the details of the Customer who has placed the order.

Solution

As a solution to the above requirement, a command button will be added to the form. On the click event of the command, a *macro* will be created that displays the *Customers* form restricted to the *Customer Number* for the order.

For the purpose of displaying the customer information the *frmCustomers* form is created based on the *Customers* table, as shown in Fig. 4.2.18.



ID	1	Address	123 1st Street
Company	Company A	City	Seattle
Last Name	Bedecs	State/Province	WA
First Name	Anna	ZIP/Postal Code	99999
E-mail Address		Country/Region	USA
Job Title	Owner	Web Page	
Business Phone	(123)555-0100	Notes	
Home Phone		Attachments	

Fig. 4.2.18: Form frmCustomers

Steps for adding a button to frmOrders

1. Open the *frmOrders* form in *Design* view. To do this, right-click the *frmOrders* form under the *All Access Objects* tab and select *Design View* from the list. The form opens in *Design View,* as shown in Fig. 4.2.19.

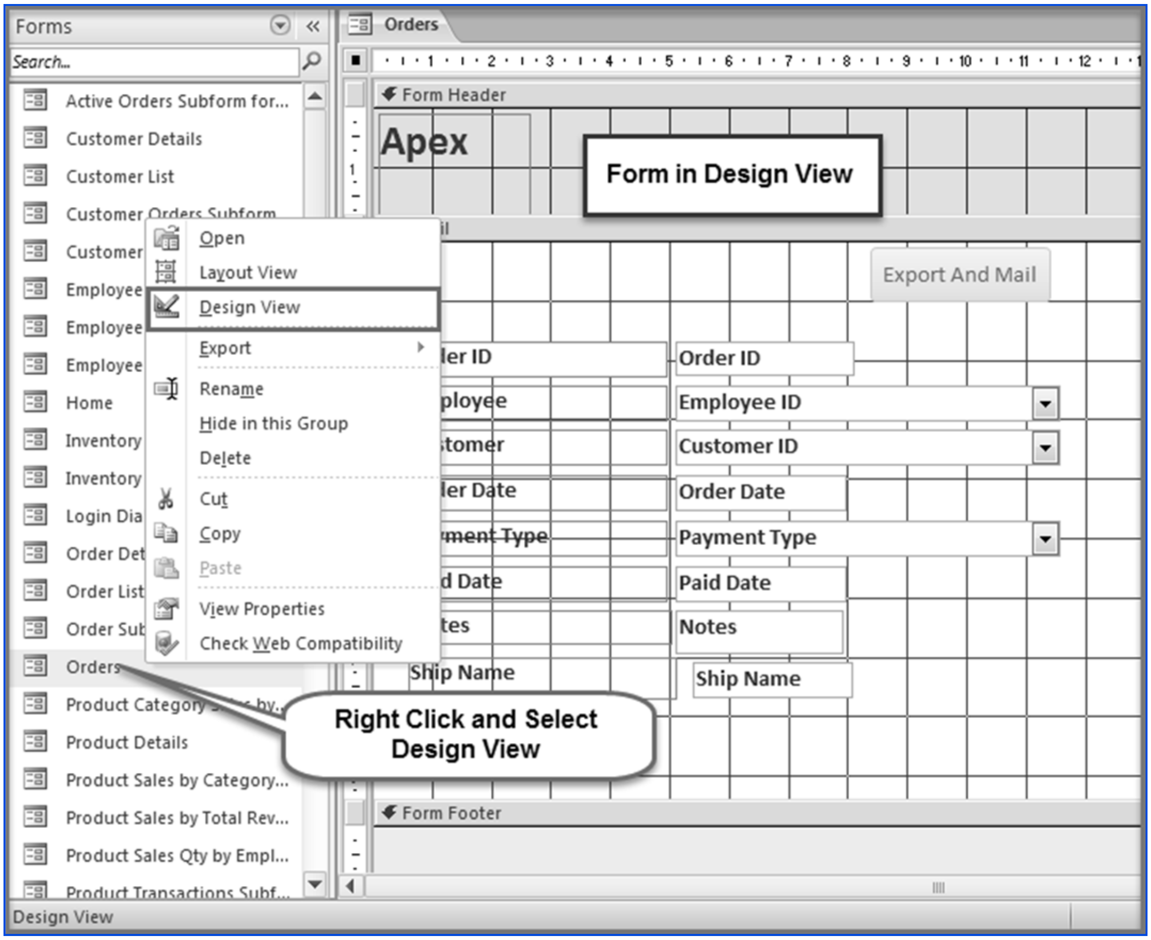
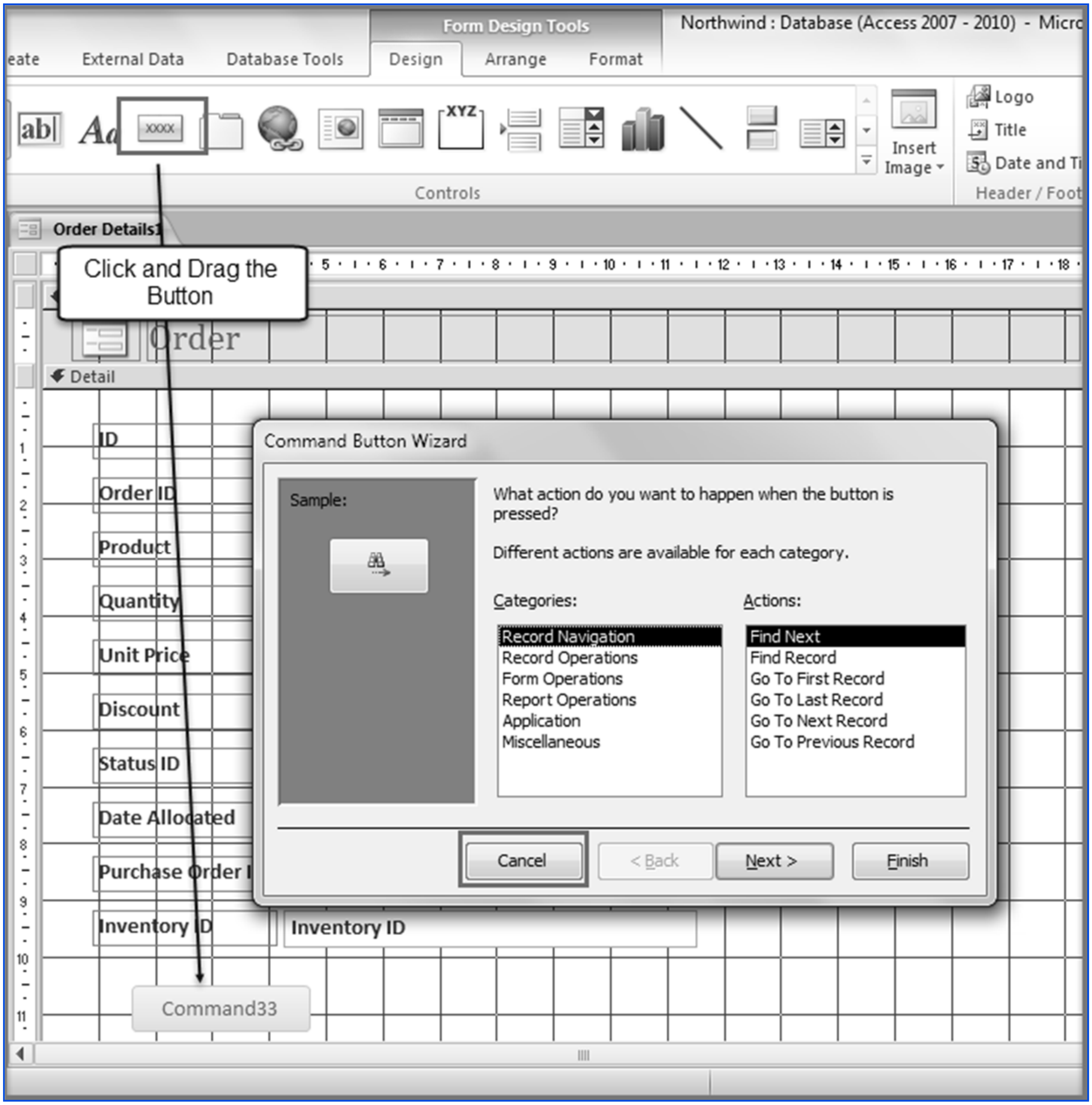


Fig. 4.2.19: Open form frmOrders in Design View

2. To add a button to the form, select the button from *Design* -> *Controls* and draw it on the form, as shown in Fig. 4.2.20. Cancel the *Command Button* wizard that appears.



Command Button Wizard

Sample:

What action do you want to happen when the button is pressed?

Different actions are available for each category.

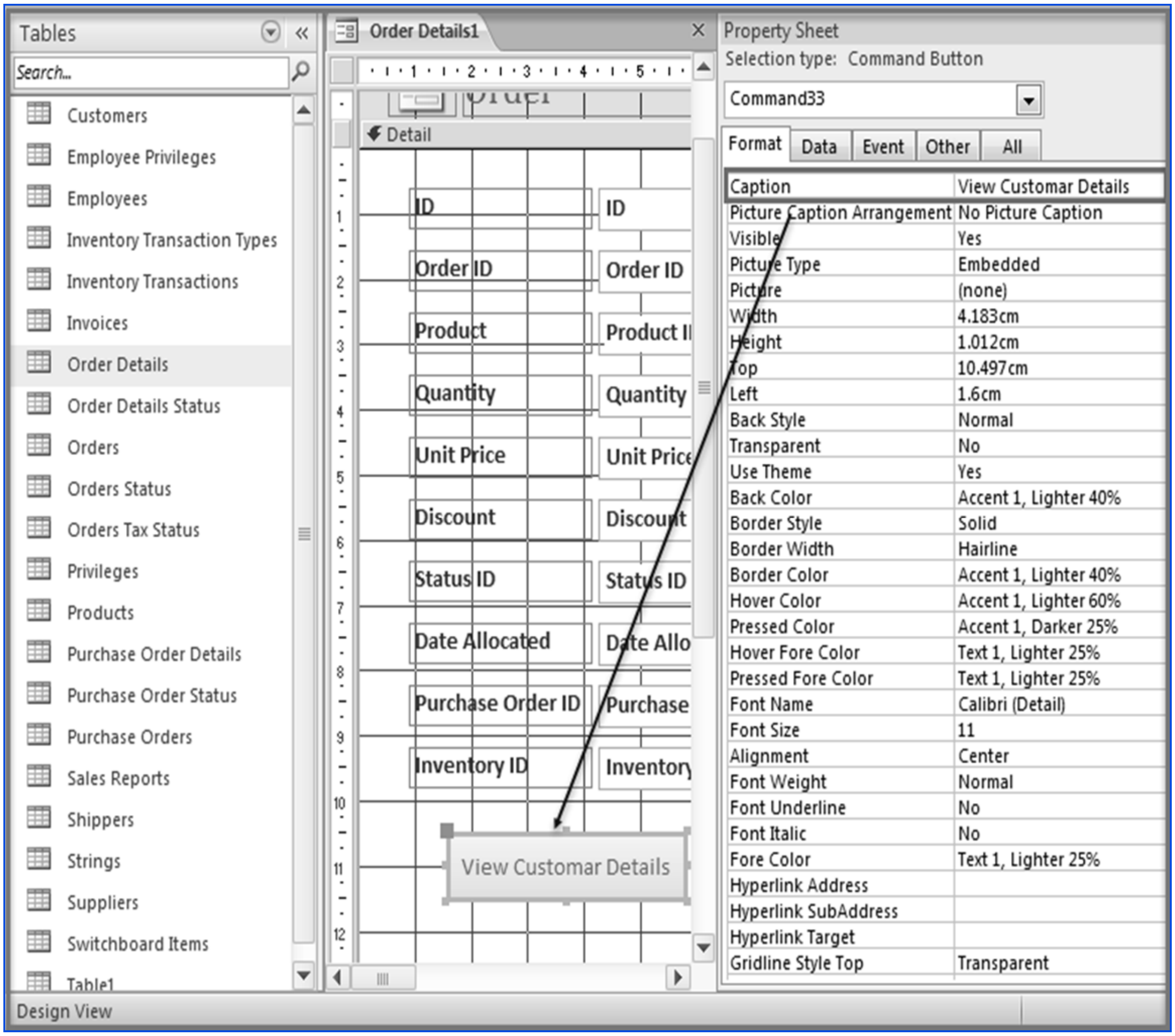
Categories:
Record Navigation
Record Operations
Form Operations
Report Operations
Application
Miscellaneous

Actions:
Find Next
Find Record
Go To First Record
Go To Last Record
Go To Next Record
Go To Previous Record

Cancel | < Back | Next > | Finish

Fig. 4.2.20: Draw Command Button on Form

3. Open the control properties by selecting the *Command* button and pressing *F4*. Set the value of *Caption* property as *View Customer Details,* as shown in Fig. 4.2.21.



Property Sheet

Selection type: Command Button

Command33

Format | Data | Event | Other | All

Caption	View Customar Details
Picture Caption Arrangement	No Picture Caption
Visible	Yes
Picture Type	Embedded
Picture	(none)
Width	4.183cm
Height	1.012cm
Top	10.497cm
Left	1.6cm
Back Style	Normal
Transparent	No
Use Theme	Yes
Back Color	Accent 1, Lighter 40%
Border Style	Solid
Border Width	Hairline
Border Color	Accent 1, Lighter 40%
Hover Color	Accent 1, Lighter 60%
Pressed Color	Accent 1, Darker 25%
Hover Fore Color	Text 1, Lighter 25%
Pressed Fore Color	Text 1, Lighter 25%
Font Name	Calibri (Detail)
Font Size	11
Alignment	Center
Font Weight	Normal
Font Underline	No
Font Italic	No
Fore Color	Text 1, Lighter 25%
Hyperlink Address	
Hyperlink SubAddress	
Hyperlink Target	
Gridline Style Top	Transparent

Fig. 4.2.21: Set Caption Property

Steps for embedding a macro to On Click event of the command button

4. In the *Event* tab of *Property Sheet*, set the value of the *On Click* event. To do this, click ... next to the *On Click* property. The *Choose Builder* window appears, as shown in Fig. 4.2.22.

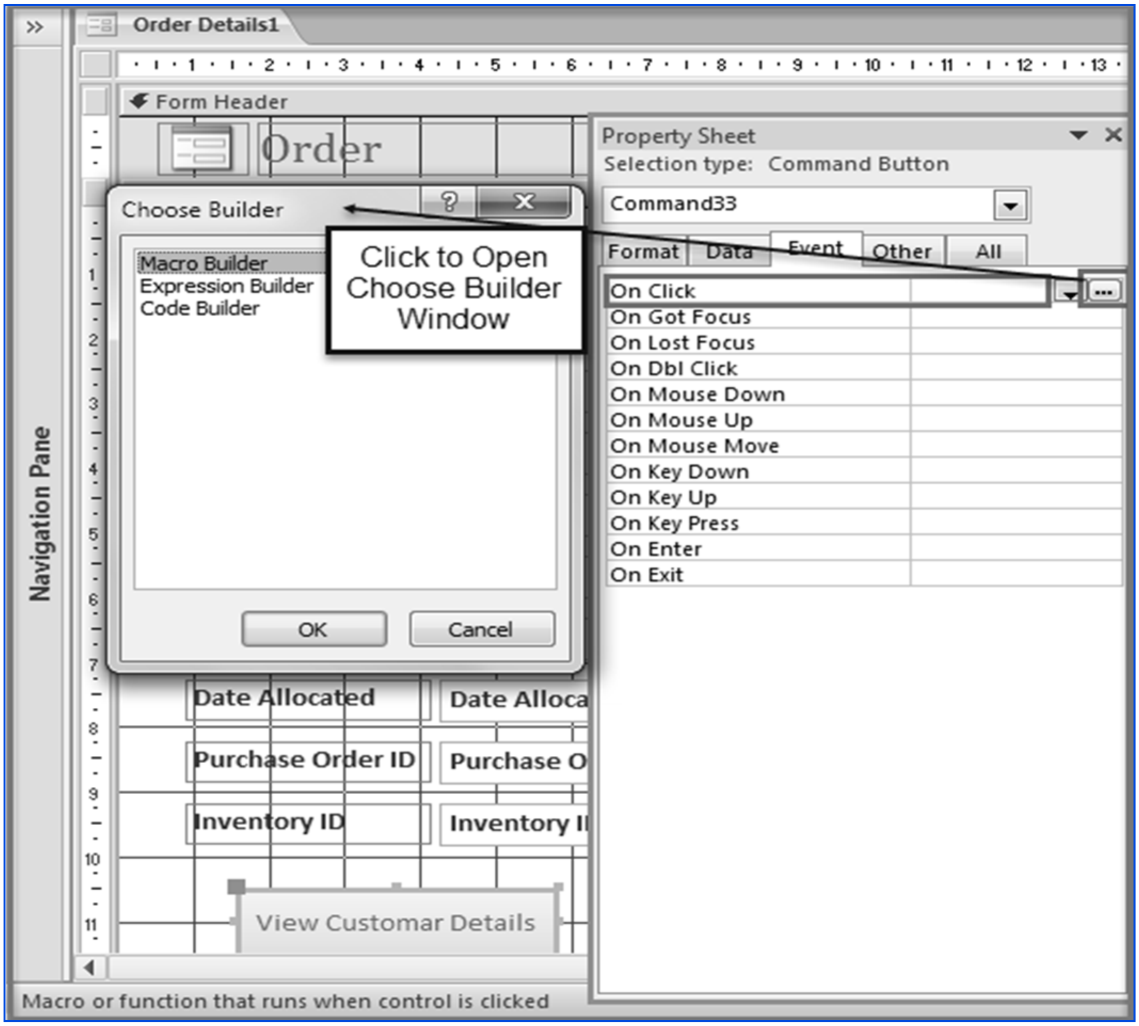


Fig. 4.2.22: Open the Choose Builder window

5. In the *Choose Builder* window, select *Macro Builder* from the list and click *OK*. The *Macro Designer* window appears, as shown in Fig. 4.2.23(A).

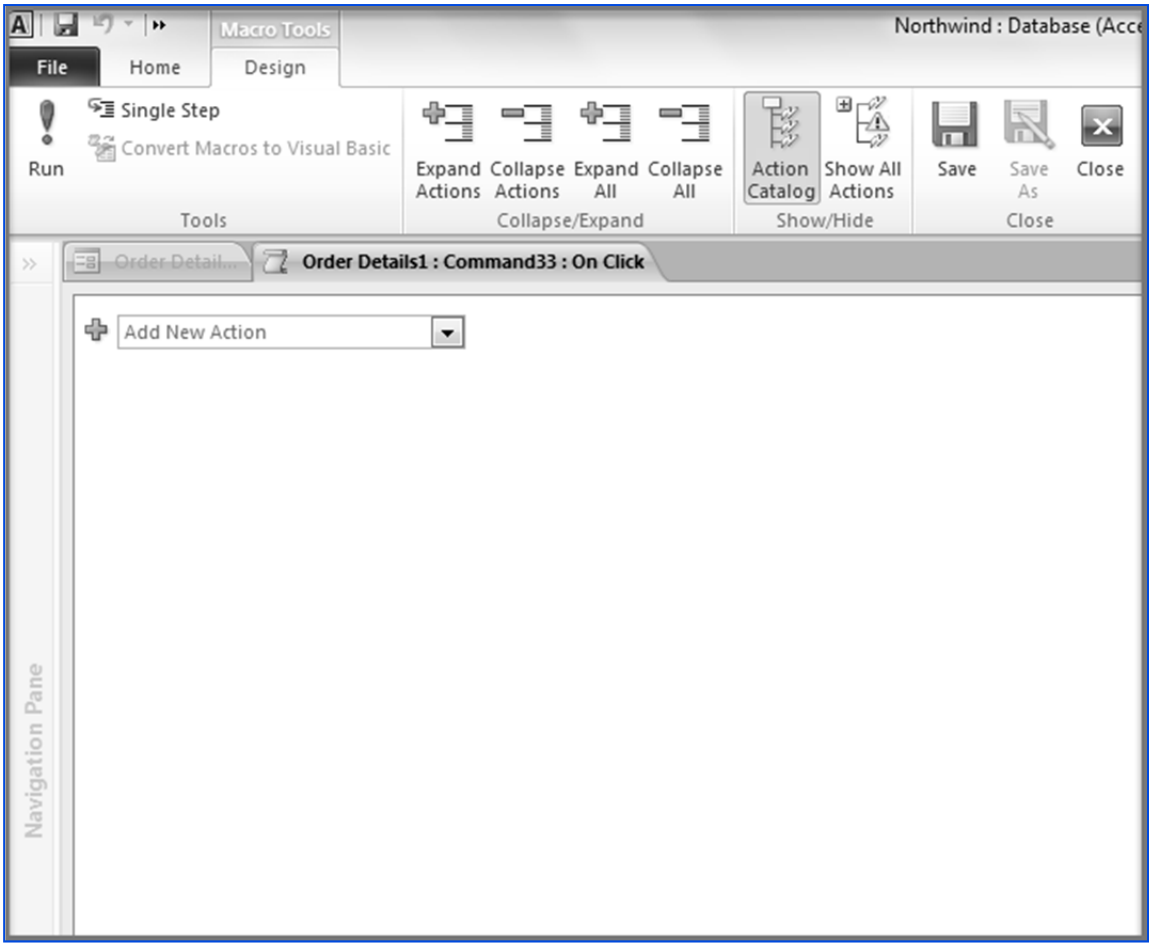
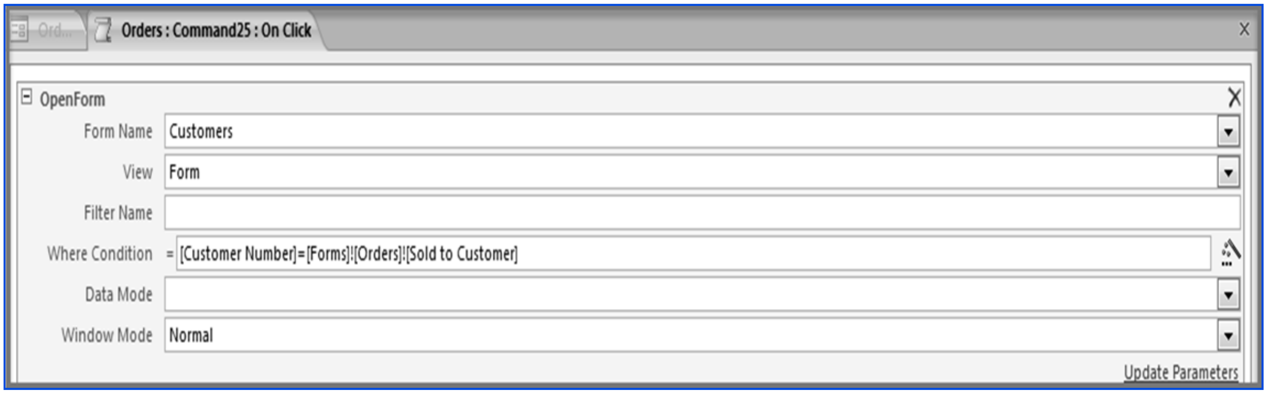


Fig. 4.2.23(A): Macro Designer window

6. In the Add New *Actions* Dropdown list, select the *OpenForm* Action from the list. Set the value of *Action arguments* as shown in Fig 4.2.23(B):



OpenForm

Form Name	Customers
View	Form
Filter Name	
Where Condition	= [Customer Number]=[Forms]![Orders]![Sold to Customer]
Data Mode	
Window Mode	Normal

Fig 4.2.23(B): Fill the arguments of OpenForm Action

Form Name	:	*Customers*
View	:	*Form*
Filter Name	:	
Where Condition	:	[Customer Number]=[Forms]![Orders]![Sold to Customer]
Data Mode	:	*Yes*
Window Mode	:	

7. Click *Design* -> *Close* -> *Close*, as shown in Fig. 4.2.24(A).

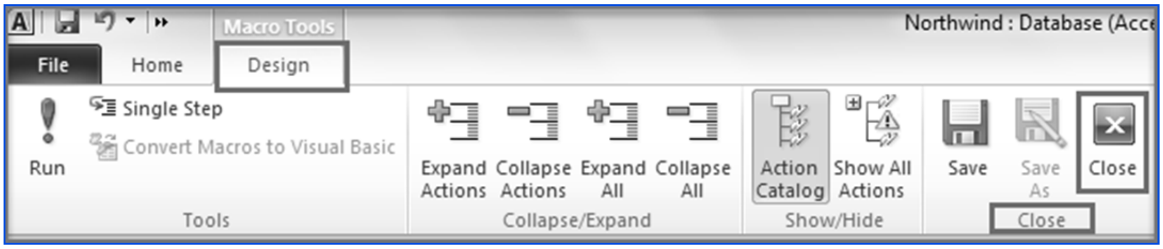
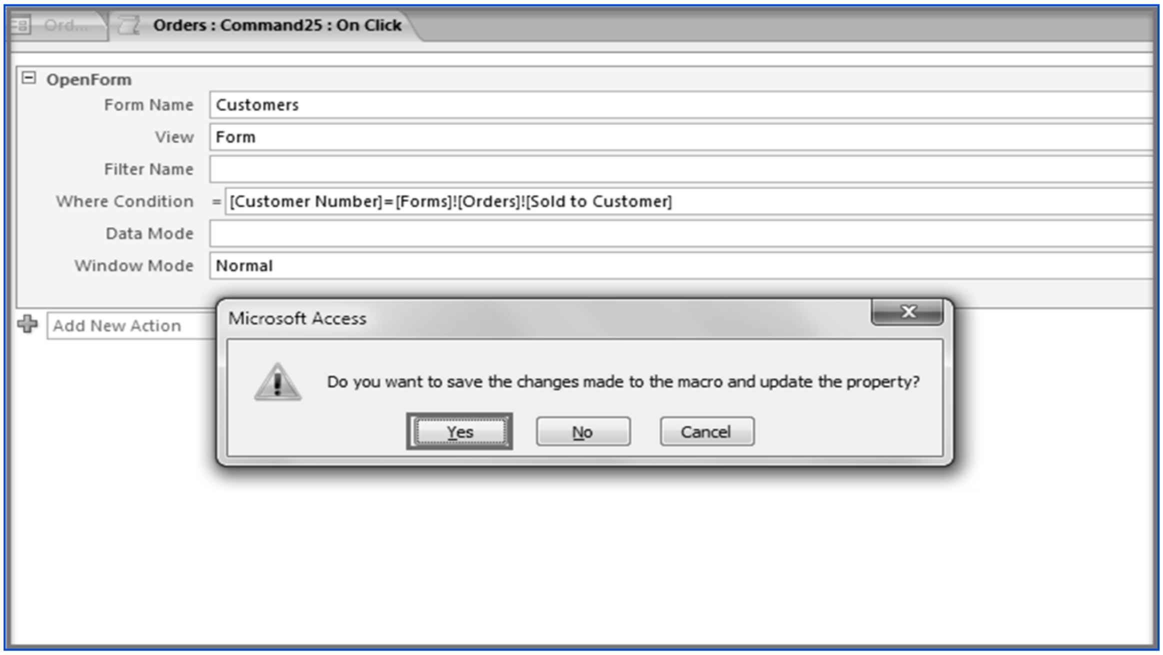


Fig. 4.2.24(A): Close the macro

8. A *message box* appears prompts to save the *macro* created, as shown in Fig. 4.2.25(B). Click *Yes* to save the *macro*.


Orders : Command25 : On Click

OpenForm

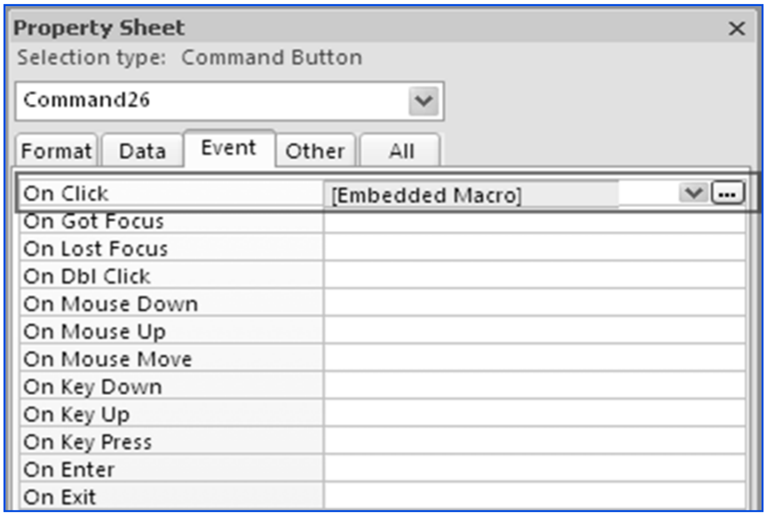
Form Name	Customers
View	Form
Filter Name	
Where Condition	= [Customer Number]=[Forms]![Orders]![Sold to Customer]
Data Mode	
Window Mode	Normal

Microsoft Access

Do you want to save the changes made to the macro and update the property?

Fig. 4.2.25(B): Confirm to save the macro

9. The created *macro* gets embedded in the *Property* window and is shown in the *On Click* property. The *Property Sheet* appears, as shown in Fig. 4.2.26.



Property Sheet

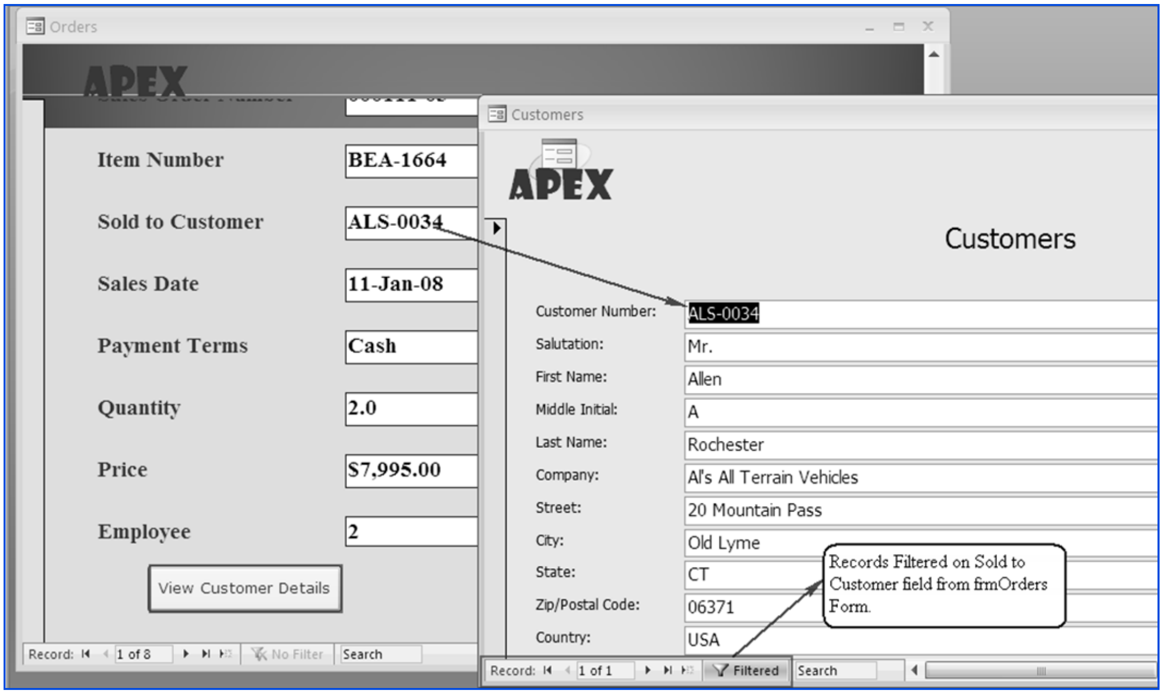
Selection type: Command Button

Command26

On Click	[Embedded Macro]
On Got Focus	
On Lost Focus	
On Dbl Click	
On Mouse Down	
On Mouse Up	
On Mouse Move	
On Key Down	
On Key Up	
On Key Press	
On Enter	
On Exit	

Fig. 4.2.26: Property Sheet

10. To check the *macro*, open the *frmOrders* form in *Form View* and click the *View Customer Details* button. This should display the *frmCustomers* form only with the record of current customer, as indicated in Fig. 4.2.27.



Item Number	BEA-1664
Sold to Customer	ALS-0034
Sales Date	11-Jan-08
Payment Terms	Cash
Quantity	2.0
Price	$7,995.00
Employee	2

Customer Number:	ALS-0034
Salutation:	Mr.
First Name:	Allen
Middle Initial:	A
Last Name:	Rochester
Company:	Al's All Terrain Vehicles
Street:	20 Mountain Pass
City:	Old Lyme
State:	CT
Zip/Postal Code:	06371
Country:	USA

Records Filtered on Sold to Customer field from frmOrders Form.

Fig. 4.2.27: Customer Records restricted by Orders form.

4.2.4. Validating Data using a Macro

Microsoft Access provides a variety of ways to control how data is entered in a database. For example, the data that a user can enter into a field can be limited to a range of values or to the format in which it must be entered. The validations can be defined at the table level by specifying table properties or the validation can be done at the form level. To validate data entry in a form, the properties of the controls may be used or alternatively all these validations can be defined using *macro*s. *Macro*s provide a good flexibility to place complex validation.

*Macro*s are useful for validating records when:

- *Validation rule* is to be applied on multiple fields in the form or to same field in different forms.
- To display customized error messages for different types of errors in the field.
- Instead of *Validation Rule*, it must be just a warning message.
- The validation involves references to controls on other forms or contains a function.

Table 4.2 lists some control events on which macro can be attached for Data Validation.

Event Property	When the Macro Will Execute
Before Update	Before the entered data is updated.
After Update	After the entered data is updated.
Before Insert	After you type in a new record.
On Delete	In response to a deletion request, but before the record is deleted.

Table 4.2: Event Property

Table 4.3 lists some macro actions that can be used as a response to validation.

Action	Use Action To
Cancel Event	Prevents a user from posting a new record unless certain conditions are met.
GoToControl	Specifies where on the form the insertion point is to be placed.
MsgBox	Displays a custom message box.

Table 4.3: Macro Actions for Data Validation

4.2.4.1 Problem Scenario

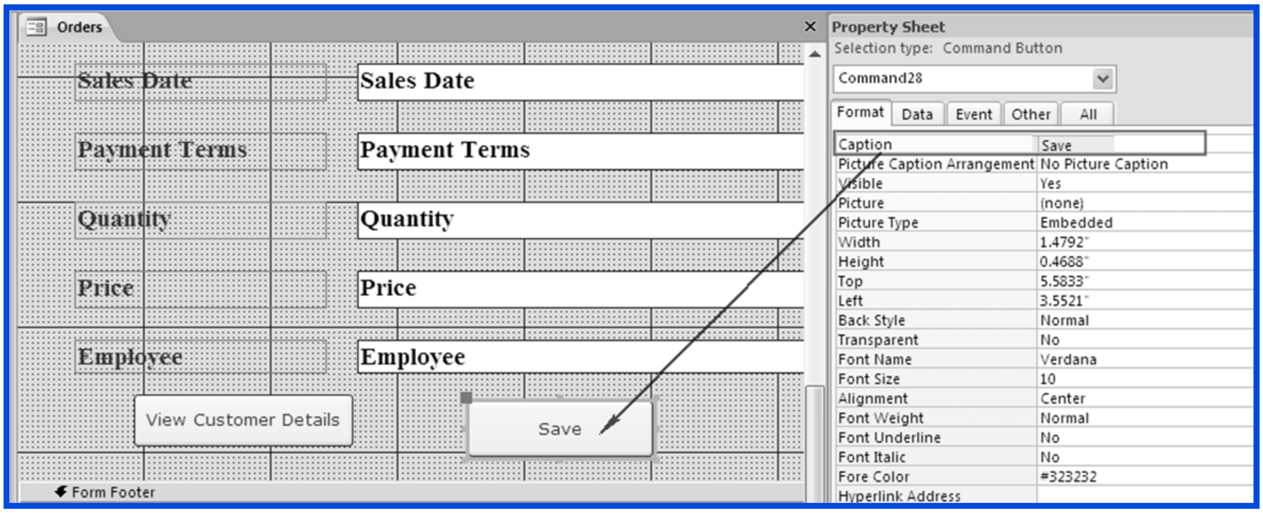
Consider the *frmOrders* form created in Section 4.2.2. Rahul Sharma, the database developer in Apex Ltd. is asked to place a check on the *Item Number* field. It is to be ensured that *Item Number* should not be left blank.

Solution

To solve the above problem, a *macro* will be embedded on the Click event property of the *Save Command* Button. This *macro* will check if the *Item Number* is blank and will display a message accordingly.

Steps for Validating Item Number

1. Open the *frmOrders* form in *Design View*. To do this right-click the frmOrders form under the *All Access Objects* tab and select *Design View* from the list.
2. Add a button to the form by selecting the button from *Design -> Controls* and draw it on the form. Cancel the *Command Button* wizard that appears.
3. Open the control properties by selecting the *Command* button and pressing *F4*. Set the value of the *Caption* property as *Save*, as shown in Fig. 4.2.28.



Property Sheet
Selection type: Command Button
Command28

Caption	Save
Picture Caption Arrangement	No Picture Caption
Visible	Yes
Picture	(none)
Picture Type	Embedded
Width	1.4792"
Height	0.4688"
Top	5.5833"
Left	3.5521"
Back Style	Normal
Transparent	No
Font Name	Verdana
Font Size	10
Alignment	Center
Font Weight	Normal
Font Underline	No
Font Italic	No
Fore Color	#323232
Hyperlink Address	

Fig. 4.2.28: Set the caption property as Save.

4. In the *Event* tab of *Property Sheet*, set the value of the *On Click* event. To do this, click ... to the right side of the *On Click* property and open the *Choose Builder* window.
5. Select *Macro Builder* from the list in the *Choose Builder* window, as shown in Fig. 4.2.29. Click the *OK* button.

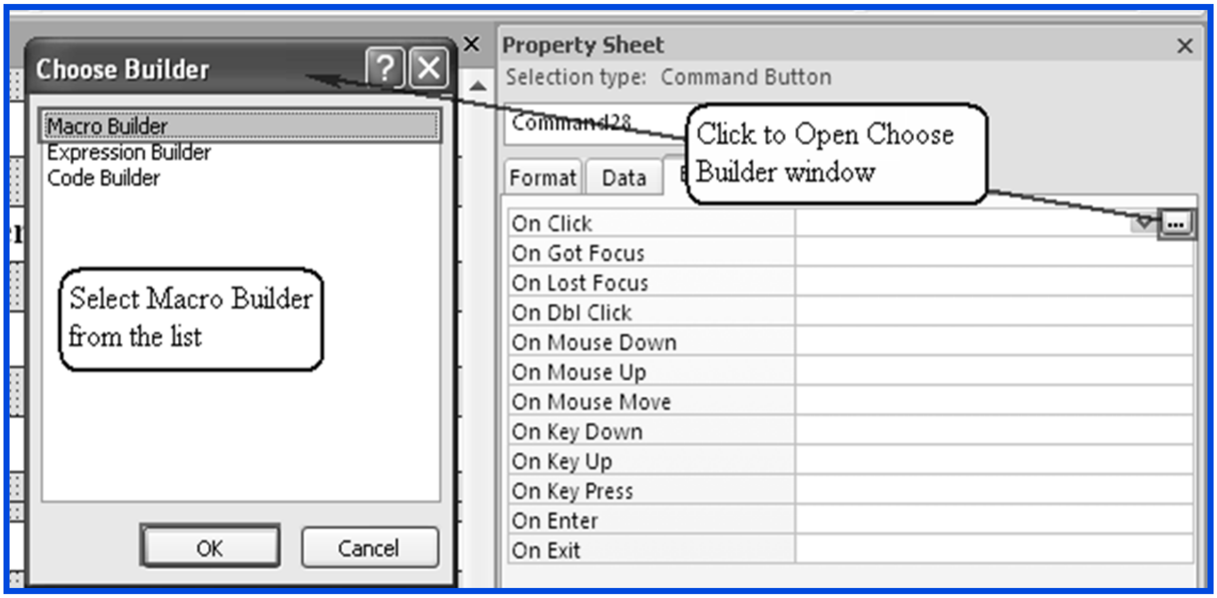


Fig. 4.2.29: Open Macro Builder

6. In the *Macro Designer* window that appears, add the conditions from selecting *If from Add New Action Menu* as displayed in Fig. 4.2.30.

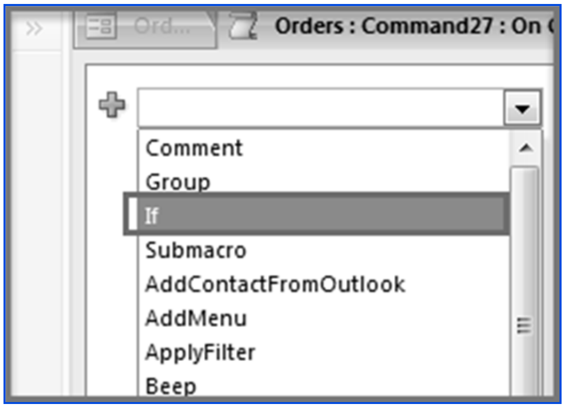
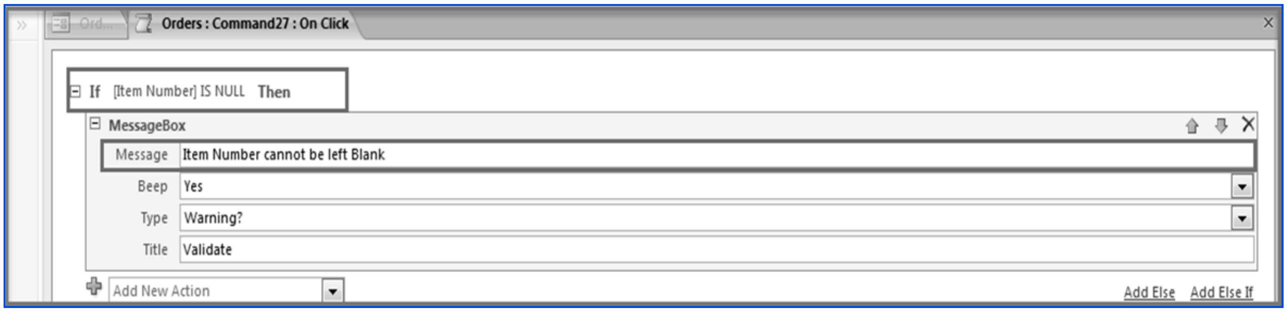


Fig. 4.2.30: Add Condition

7. In the *Condition* Argument textbox of the *Macro Designer* type the text *[Item Number] IS NULL*, this condition checks whether the *Item Number* value is blank.

8. In the Add New *Action* Dropdown, select the *CancelEvent* action from the list. The *Query Designer window* appears, as shown in Fig. 4.2.31.



Orders : Command27 : On Click

If [Item Number] IS NULL Then

MessageBox

Message Item Number cannot be left Blank

Beep Yes

Type Warning?

Title Validate

Fig. 4.2.32: Specify Condition & Action

9. In the next row, type … in the *Condition tab*. In the *Action* tab, select *MsgBox* from the list and specify *Action Arguments* as:

   Message : Item Number cannot be left Blank

   Beep : Yes

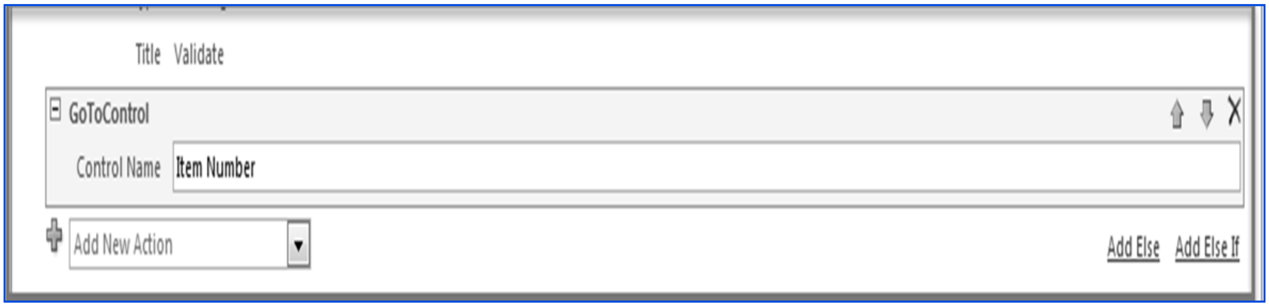
   Type : Warning?

   Title : Validate

10. In the next row, type … in the *Condition* tab. In the *Action* tab, select *GoToControl* from the list and specify *Action Arguments*:

    Control Name : Item Number

    The *Macro Designer* window appears, as shown in Fig. 4.2.33.



Title Validate

GoToControl

Control Name Item Number

Fig. 4.2.33: Specify GoToControl Action

11. Click *Design* -> *Close* -> *Close*, as shown in Fig. 4.2.34.

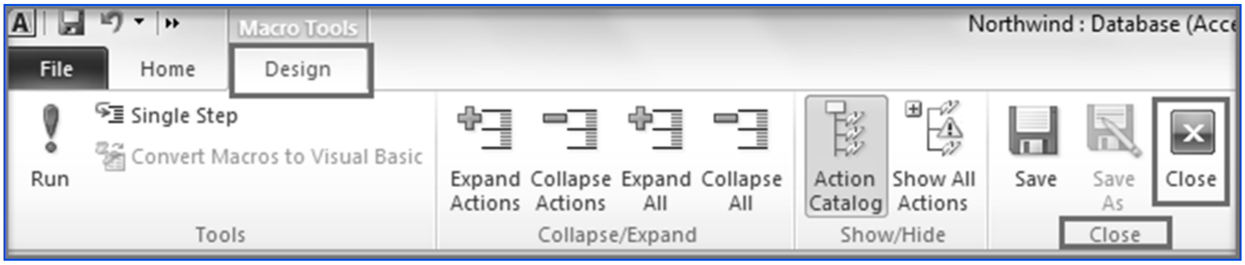
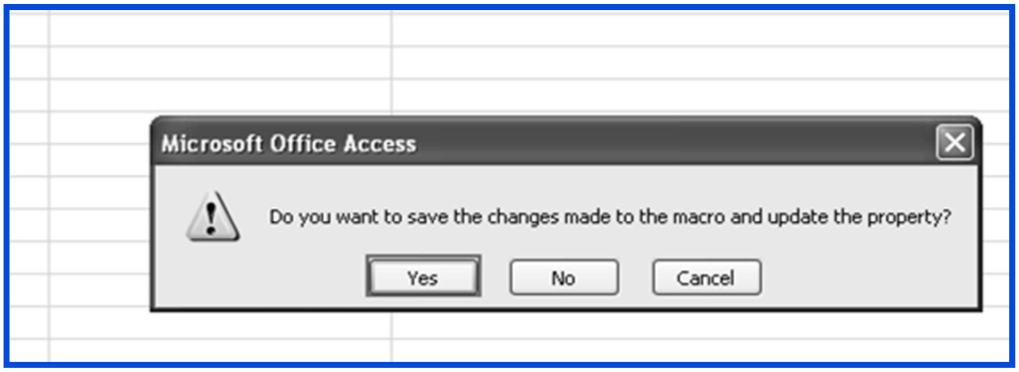


Fig. 4.2.35: Click Close to save and close the macro

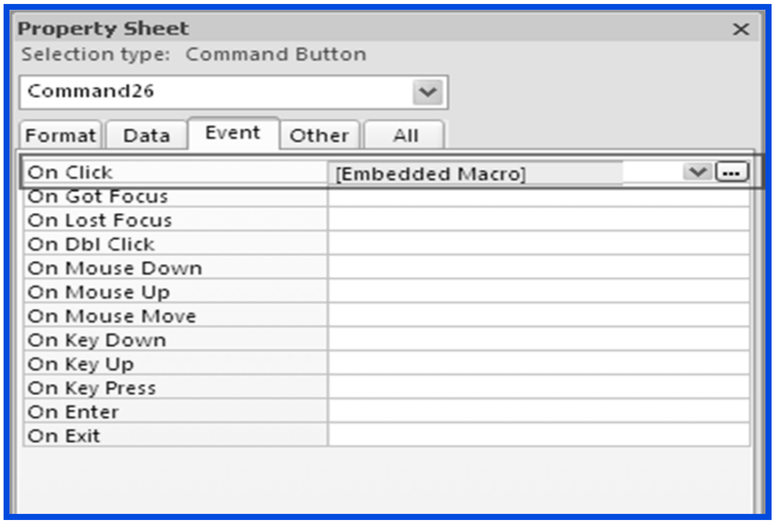
12. A message box appears prompting to save the *macro* created, as displayed in Fig. 4.2.36. Click *Yes* to save the *macro*.



Do you want to save the changes made to the macro and update the property?

Fig. 4.2.36: Confirm to Save the macro

Note that the created *macro* gets embedded in the *Property* window and is shown in the *On Click* property. *Property Sheet* appears, as shown in Fig. 4.2.37.



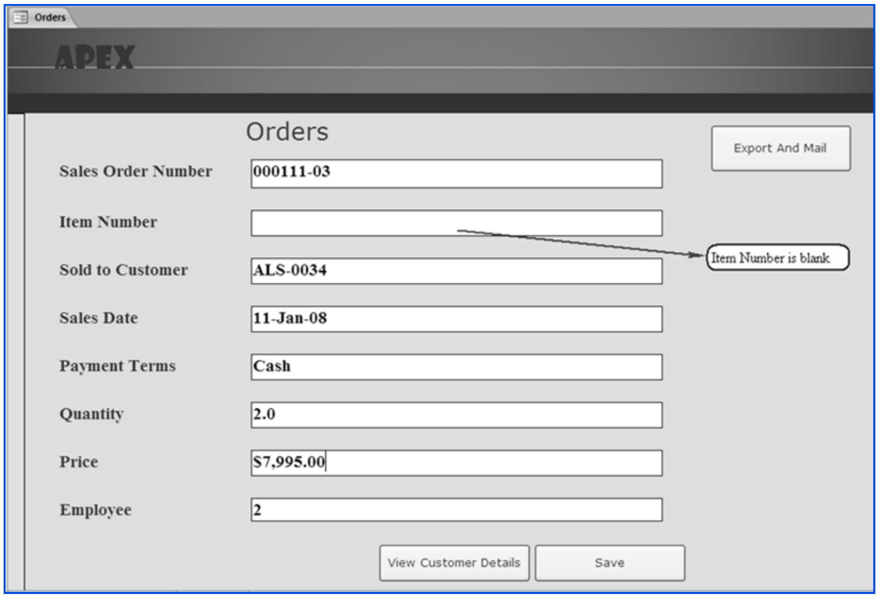
Selection type: Command Button

Command26

On Click	[Embedded Macro]
On Got Focus	
On Lost Focus	
On Dbl Click	
On Mouse Down	
On Mouse Up	
On Mouse Move	
On Key Down	
On Key Up	
On Key Press	
On Enter	
On Exit	

Fig. 4.2.37: Property Sheet

13. To check the *macro*, open the *frmOrders* form in *Form View* and enter a new record with the values, as shown in Fig. 4.2.38.



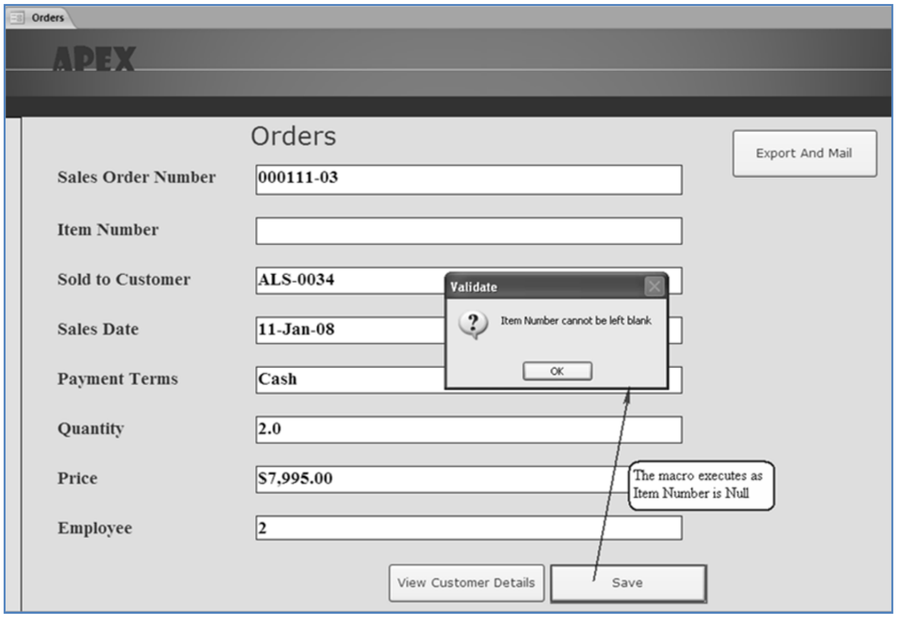
Orders

Sales Order Number	000111-03
Item Number	
Sold to Customer	ALS-0034
Sales Date	11-Jan-08
Payment Terms	Cash
Quantity	2.0
Price	$7,995.00
Employee	2

Item Number is blank

Fig. 4.2.38: Enter a New Record

14. Click the *Save* button to save the record. Since the *Item Number* field is left blank the *macro* gets executed and the message is displayed, as shown in Fig. 4.2.39.



Orders

Sales Order Number	000111-03
Item Number	
Sold to Customer	ALS-0034
Sales Date	11-Jan-08
Payment Terms	Cash
Quantity	2.0
Price	$7,995.00
Employee	2

Validate

Item Number cannot be left blank

The macro executes as Item Number is Null

Fig. 4.2.39: Macro Executes

Note that the multiple conditions and their post actions can be specified in the same *macro*; specifying the validation rule for other fields also.

4.2.5. Automating Data Entry Using a Macro

*Macro*s can be used to avoid errors and save time during data entry by automating the data entry process. Instead of having users type in the same data over and over for each record with the possibility of invalid data being entered, a *macro* can automate data entry. Microsoft Access Macros can be used to speed up the process of data entry. While automating the entry of data, it can be ensured that the possibility of errors in database data is minimized, which can have an impact on the accuracy of the data available for reports and queries.

*Macro*s can also be used for data entry where the value to be entered is dependent on value of other fields. For Example, a *macro* can be created to enter the total cost of an order if the unit price and quantity to be ordered of an item is provided.

Table 4.4 lists the common events that can be used for Automating Data Entry.

Event Property	When the Macro Will Be Executed
On Enter	Upon arriving on a control, but before the control has focus.
Before Update	Before the control data is updated.
After Update	After the changed control data is updated.
On Exit	Upon leaving a control, but before the focus is removed.

Table 4.4: Event for Automating Data Entry

Table 4.5 specifies macro action used for Automating Data Entry.

Action	Description
SetLocalVar	Enters a value automatically in a field. The field name and the value that needs to be entered in the field are mentioned as arguments. You need to enter the arguments in the Action Arguments pane for this action.
GoToControl	Specifies the field where the insertion point needs to be moved after a value.

Table 4.5: Macro Actions for Automating Data Entry

4.2.5.1 Problem Scenario

In the *frmOrders* form discussed in section 4.2.2, the Operational Manager wishes to see the total cost of the order, which would be computed as *Quantity ordered * price* of an Item.

Solution

A text box as *Total Order Cost* is added to the form, and a *macro* is created on the *Exit* event of price to compute the total cost of an order. This *macro* first checks the value of *Quantity* is not null and then computes Total Cost.

Steps for creating a macro to automate data entry

1. Open the *frmOrders* form in the *Design* view. Add a TextBox to a form, select the *TextBox* control from *Design* -> *Controls* and draw it on the form, as shown in Fig. 4.2.40.

   Note that the TextBox also creates a label on the form so as the label to the *Text* value can be specified.

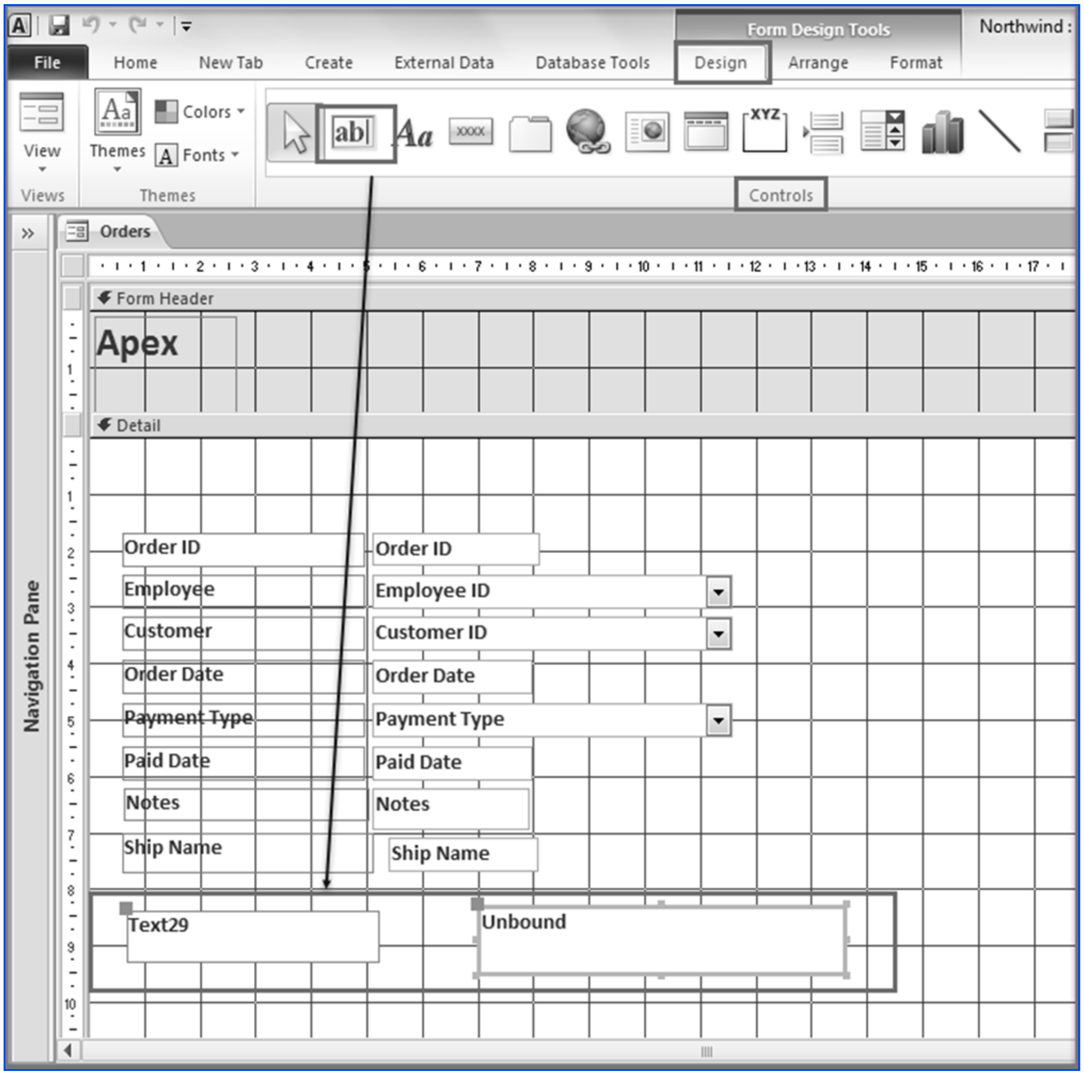


Fig. 4.2.40: Draw the TextBox on form

2. Double-click the label and write the text *Total Cost*. Change the format of the label to match it with other controls. In this case, set the format as "Times New Roman, size 12, Bold". The *Form Design* window appears, as shown in Fig. 4.2.41.

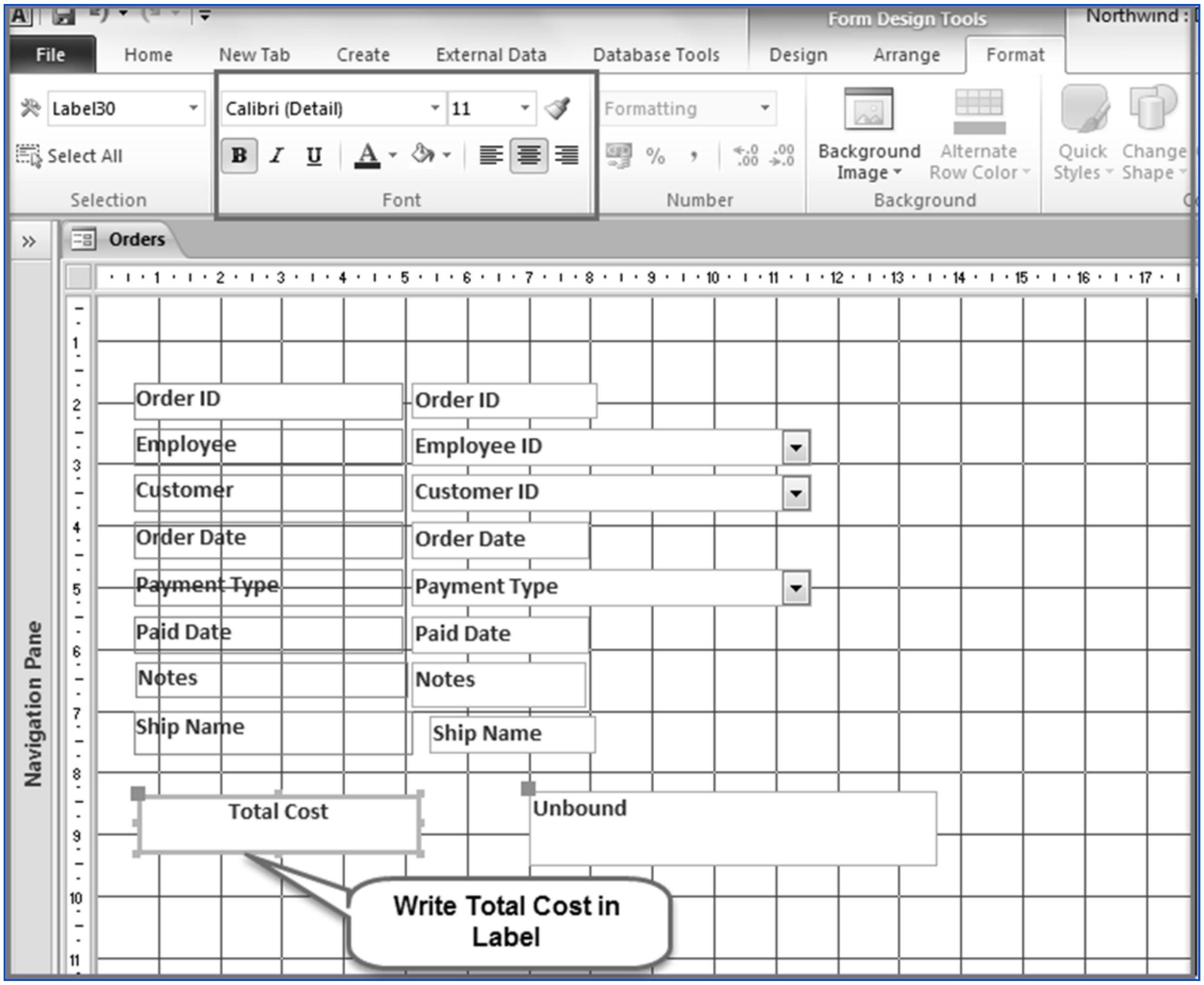
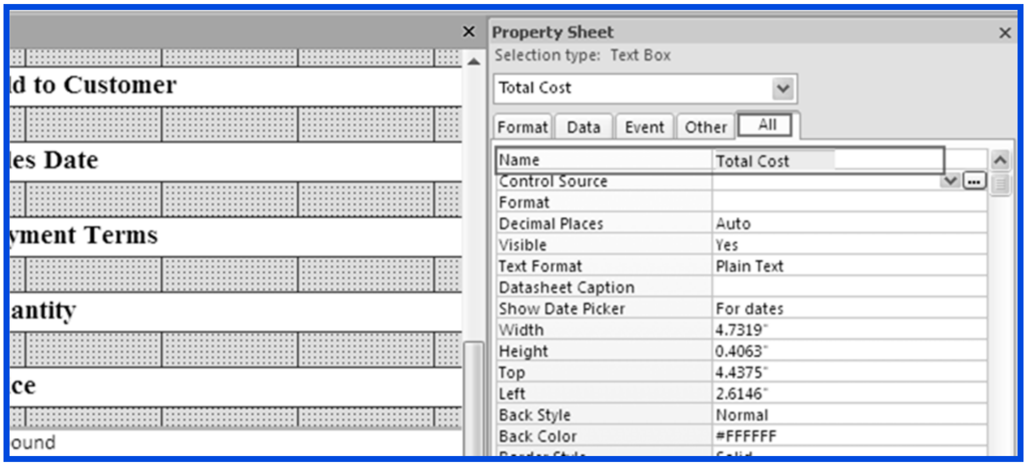


Fig. 4.2.41: Specify the text as Total Cost

3. Open the control properties by selecting *TextBox* and pressing *F4*. In the *All* tab of *Property Sheet*, set the value of the *Name* property as "Total Cost", as shown in Fig. 4.2.42.



Selection type: Text Box

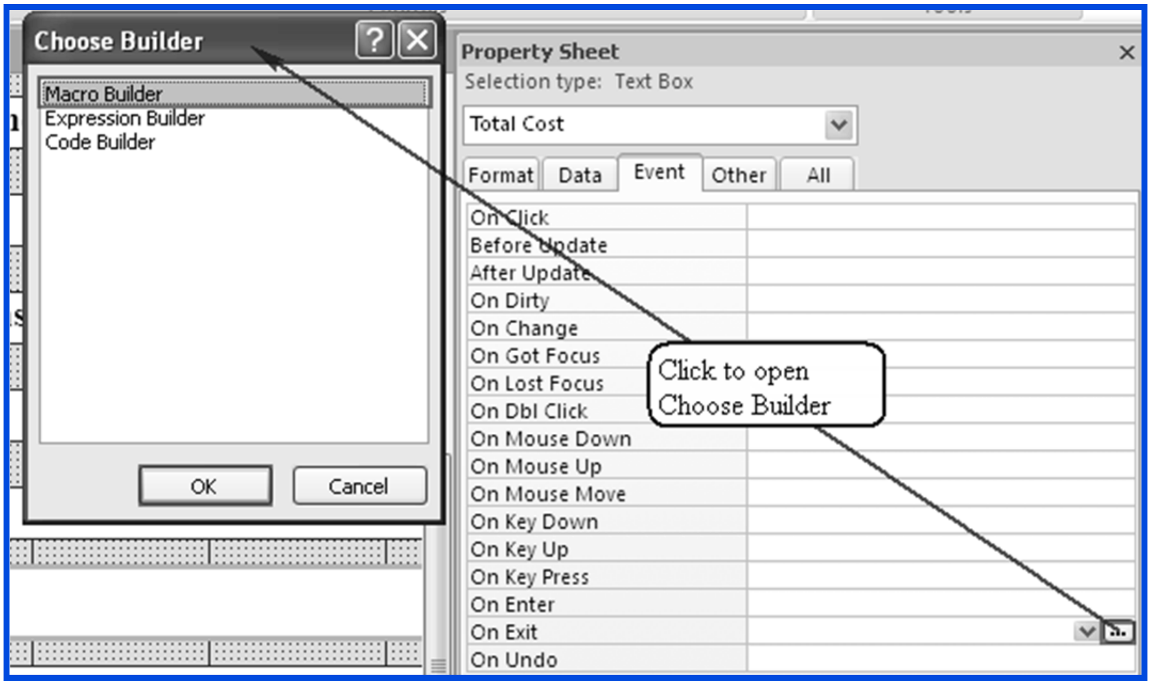
Total Cost

Name	Total Cost
Control Source	
Format	
Decimal Places	Auto
Visible	Yes
Text Format	Plain Text
Datasheet Caption	
Show Date Picker	For dates
Width	4.7319"
Height	0.4063"
Top	4.4375"
Left	2.6146"
Back Style	Normal
Back Color	#FFFFFF

Fig. 4.2.42: Set the Name property

4. In the *Event* tab of *Property Sheet*, set the value of the *On Click* event. To do this, click … to the right side of the *On Click* property and open the *Choose Builder* window.

5. Select *Macro Builder* from the list in the *Choose Builder* window, as shown in Fig. 4.2.43. Click the *OK* button.



Selection type: Text Box

Total Cost

Fig. 4.2.43: Open Macro Builder

6. In the *Macro Designer* window, add the Conditions tab by selecting *Condition* from *Design* -> *Show/Hide*.

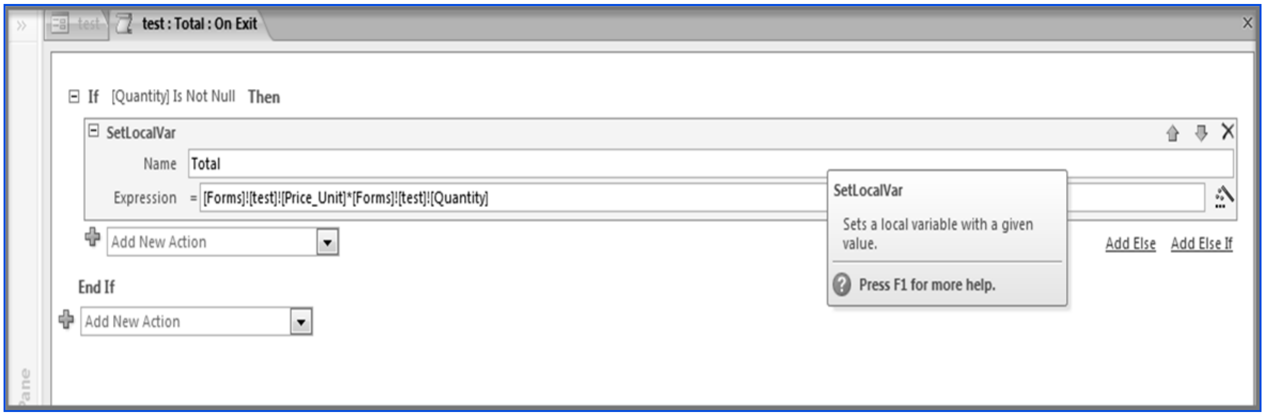
7. In the *Condition* tab of the *Macro Designer* type the text *[Quantity] Is Not Null*, this condition checks if the *Quantity* value is blank.

8. In the *Action drop down*, select the *SetLocalVar* action from the list. Set the value of *Action arguments* as:

   Name : Name of the local Vatriable

   Expression : [Quantity] * [Price]

   The Query Designer window appears, as shown in Fig. 4.2.44.



test : Total : On Exit

If [Quantity] Is Not Null Then

SetLocalVar

Name Total

Expression = [Forms]![test]![Price_Unit]*[Forms]![test]![Quantity]

End If

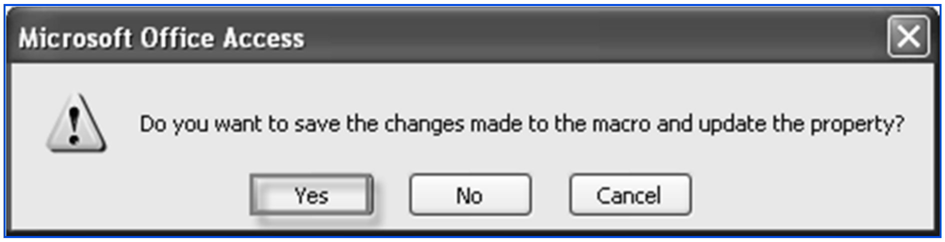
SetLocalVar

Sets a local variable with a given value.

Press F1 for more help.

Fig. 4.2.44: The Macro Designer window

9. Click *Design* -> *Close* -> *Close* to save and close the *macro*. A *message box* appears prompting to save the *macro* created, as displayed in Fig. 4.2.36. Click *Yes* to save the *macro*.

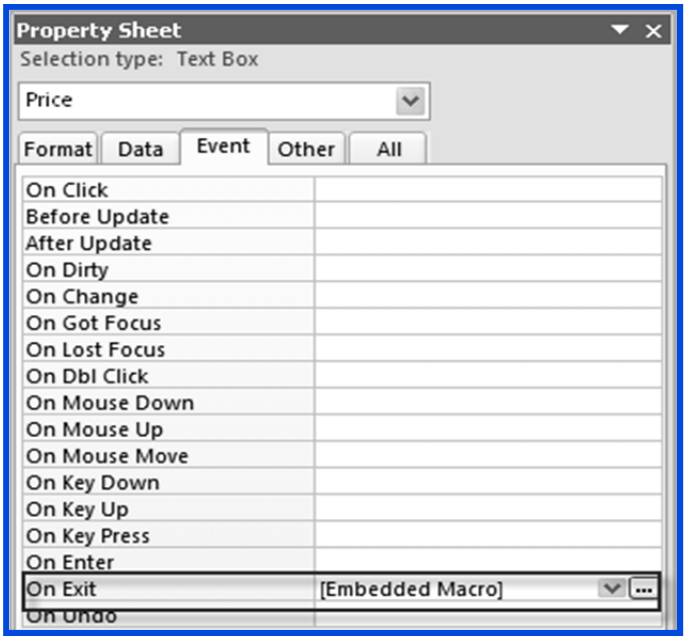


Microsoft Office Access

Do you want to save the changes made to the macro and update the property?

Fig. 4.2.45: Click Yes to save the macro

10. The created *macro* gets embedded in the *On Exit* event property of the *Price TextBox* control. *Property Sheet* appears, as shown in Fig. 4.2.46.



Property Sheet

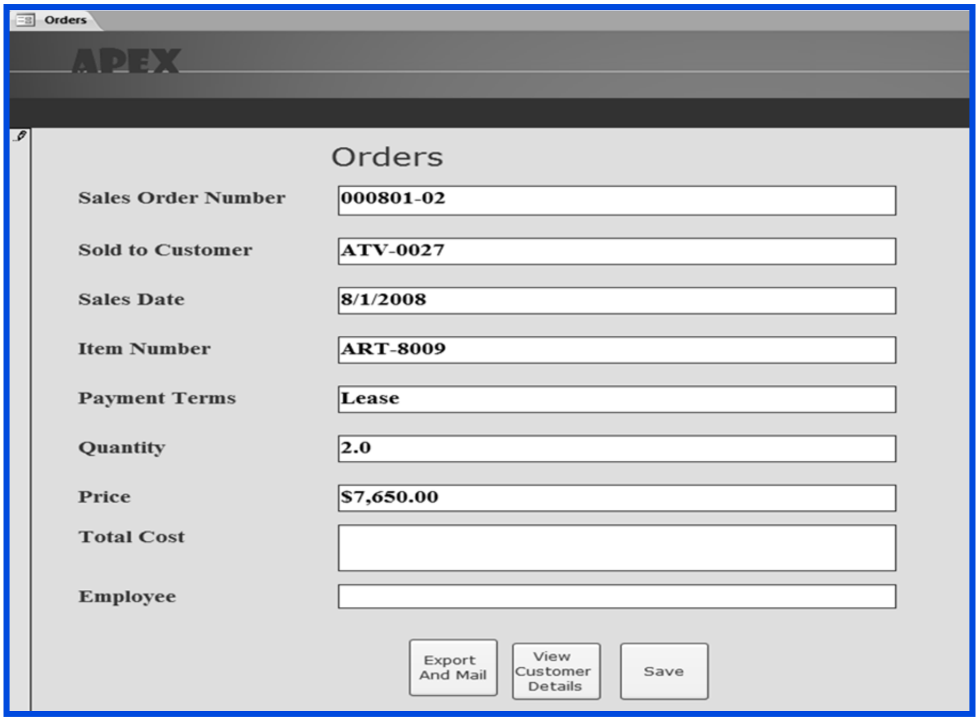
Selection type: Text Box

Price

On Click	
Before Update	
After Update	
On Dirty	
On Change	
On Got Focus	
On Lost Focus	
On Dbl Click	
On Mouse Down	
On Mouse Up	
On Mouse Move	
On Key Down	
On Key Up	
On Key Press	
On Enter	
On Exit	[Embedded Macro]
On Undo	

Fig. 4.2.46: Property Sheet

11. Open the *frmOrders* form in *Form View* and enter a new record with the values, as shown in Fig. 4.2.47.

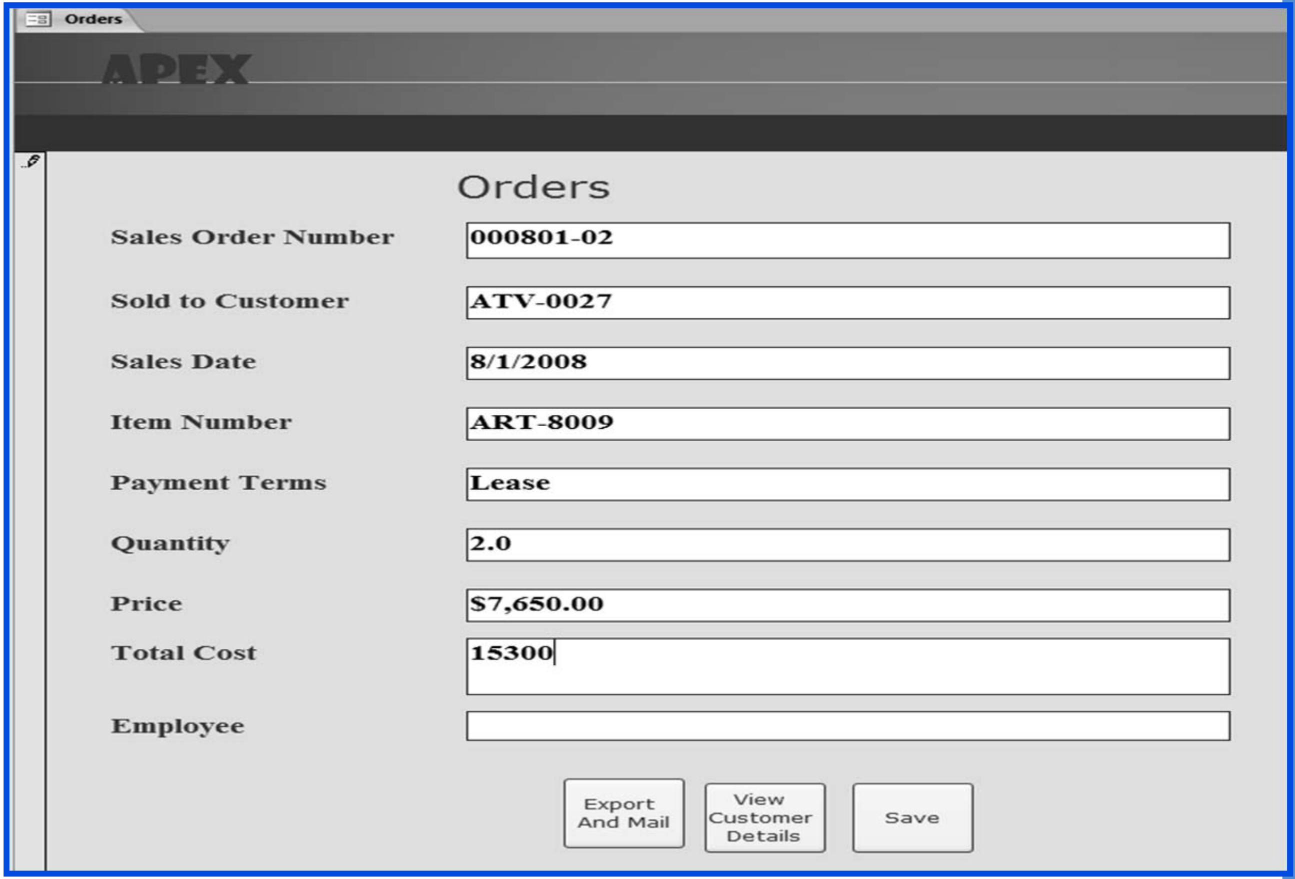


Orders

Sales Order Number	000801-02
Sold to Customer	ATV-0027
Sales Date	8/1/2008
Item Number	ART-8009
Payment Terms	Lease
Quantity	2.0
Price	$7,650.00
Total Cost	
Employee	

Fig. 4.2.47 : Insert a new record in frmOrders

12. As you enter the value for price and move to the next textbox, the *On Exit* event of the *Price* control gets executed and the total cost is calculated, as shown as Fig. 4.2.48.



Orders

Sales Order Number	000801-02
Sold to Customer	ATV-0027
Sales Date	8/1/2008
Item Number	ART-8009
Payment Terms	Lease
Quantity	2.0
Price	$7,650.00
Total Cost	15300
Employee	

Fig. 4.2.48: Automated Data Entry for Total Cost

4.3 Managing Switchboards

A *switchboard* is a Microsoft Office Access 2010 form that facilitates navigation in Access and access to different parts of an application. It functions as an interface between the user and the application. A *switchboard* is similar to the *Ribbon* of the Access application. It provides users with direct access to the specific functions of the application and acts as an interface between the user and the application.

Switchboard contains command buttons that execute specified actions. These buttons can be programmed to open *forms*, *reports*, *queries*. Each button on the *switchboard* triggers some action within the database or leads to another *switchboard* form. *Switchboard* forms are an invaluable way to keep users focused on using the database as intended.

A *switchboard* form presents the user with a limited number of choices for working with the application and makes the application easier and user specific to use. The user's login information can determine which of a number of switchboard forms to use. For instance, a manager with a higher level of privileges may be given a form with more options than a clerical worker would be given.

4.3.1. Creating a Database Switchboard

The *Switchboard Manager* is a dialog box that allows creating a *switchboard* for an Access database. It lists the *switchboards* currently available in a database and provides an option to create new ones. In each *switchboard*, command buttons can be added, deleted, or edited. The command buttons can be configured for accessing *forms*, *reports*, *macro*s, or *functions* in the database. *Switchboard Manager* only allows a maximum of eight command buttons on a *switchboard*. The operations that the command button on *switchboard* can perform are listed in Table 4.6.

Command	Action performed
Go to Switchboard	Opens a secondary switchboard.
Open Form in Add Mode	Opens a form in a mode that only allows new records to be added.
Open form in Edit Mode	Opens a form in a mode that allows any record to be added or edited.
Open Report	Opens a report in Print Preview.
Design Application	Opens the Switchboard Manager.
Exit Application	Closes the current database.
Run Macro	Runs a macro.
Run Code	Runs a Visual Basic function.

Table 4.6: Different Commands that Switchboard can perform

When a *switchboard* is created with *Switchboard Manager*, Access creates the *Switchboard Items* table that describes what the display text and action performed by the buttons on the *Switchboard* form.

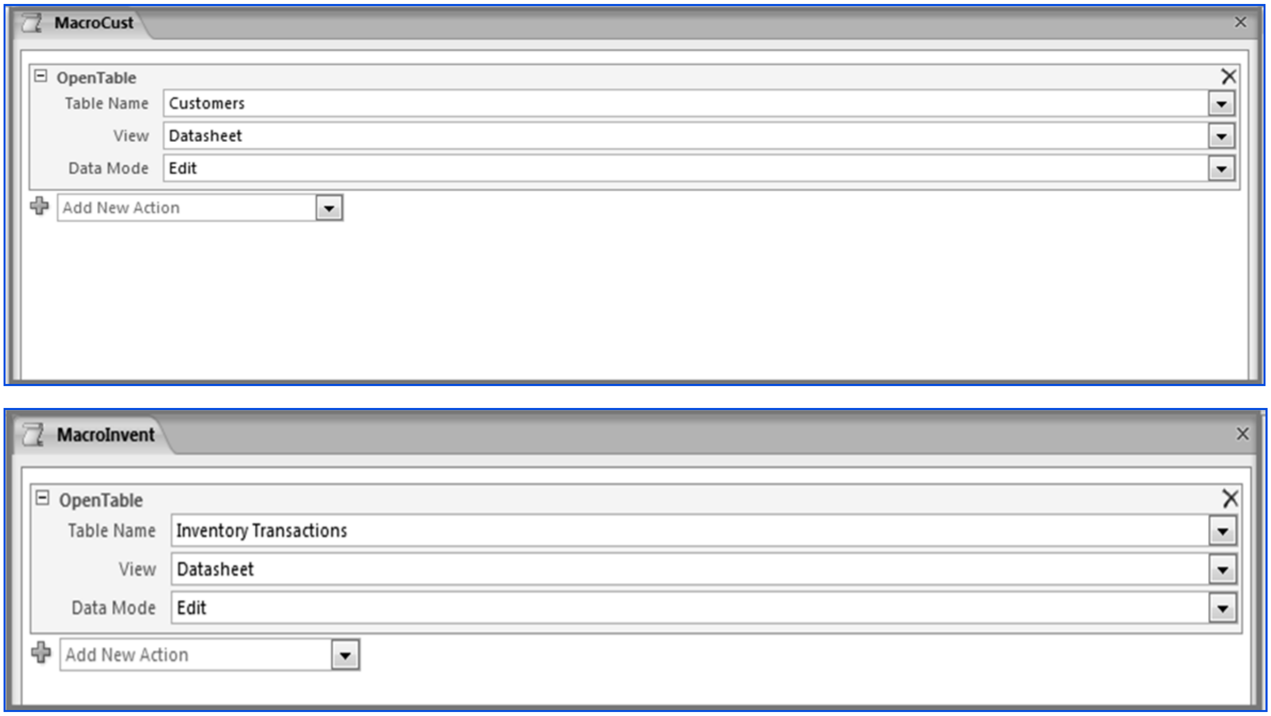
4.3.1.1 Problem Scenario

Employees of Apex Ltd. need to keep updating the company database frequently for various reasons. However, employees are confused over the interface that appears when the application is started. The application developer wishes to resolve this confusion by providing a clear and concise environment in which users can reduce the amount of time spent figuring out how to obtain the information they are looking for.

Solution

To resolve the problem of navigation in the database, a *Switchboard* form that contains buttons to open the *Customers* and *Inventory* tables and the *frmOrders* form needs to be created.

Since *switchboard* is not capable of opening the tables directly through the *Command* button, so we have created *macro*s to open the *Customers* and *Inventory* tables named as *MacroCust* and *MacroInvent* respectively. The structure of *macro*s is displayed in Fig. 4.3.1.



MacroCust

OpenTable

Table Name: Customers

View: Datasheet

Data Mode: Edit

MacroInvent

OpenTable

Table Name: Inventory Transactions

View: Datasheet

Data Mode: Edit

Fig. 4.3.1: Macro to open Customers and Inventory table

NOTE: Notice that both the *MacroCust* and *MacroInvent macro*s are created in the same *Macro Designer* window using the *Macro Name* property.

Steps for Adding Switchboard in Access 2010

The Switchboard Manager is still available in Access 2010, but it's not included in the Database Utilities Ribbon as in Access 2007 and 2003. You have to launch it as doing the following steps as shown in Fig.4.3.2:

Step 1: First click the **File** tab and **Options** button, then we will get into Access Options window;

Step 2: Click the **Quick Access Toolbar** at left bar;

Step 3: In the Choose commands from drop down box, select the **Commands Not in the Ribbon** item;

Step 4: Select the **Switchboard Manager** item in the command list box;

Step 4: Click the **Add** button;

Step 6: At last click the **Ok** button at the bottom.

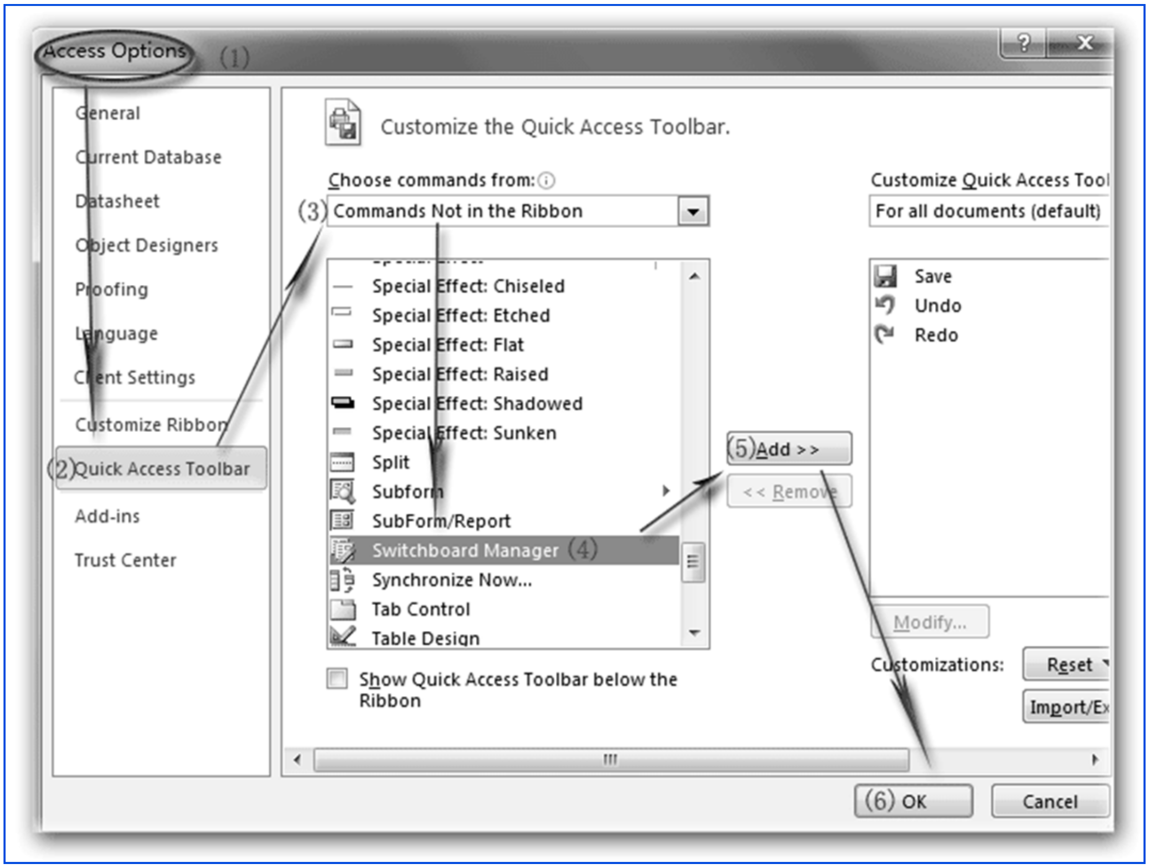


Fig. 4.3.2: Open Switchboard Manager in MS Access 2010

Steps for creating Switchboard in Access 2010:

On the *New* tab, in the Ribbon, click *Switchboard Manager,* as shown in Fig. 4.3.3.

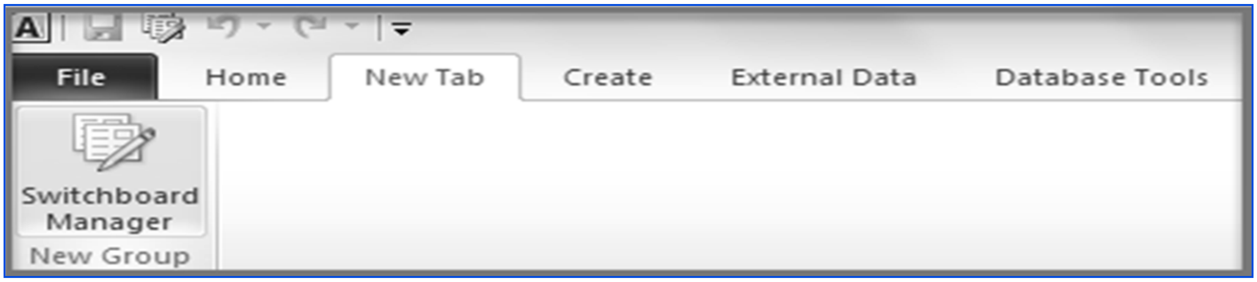
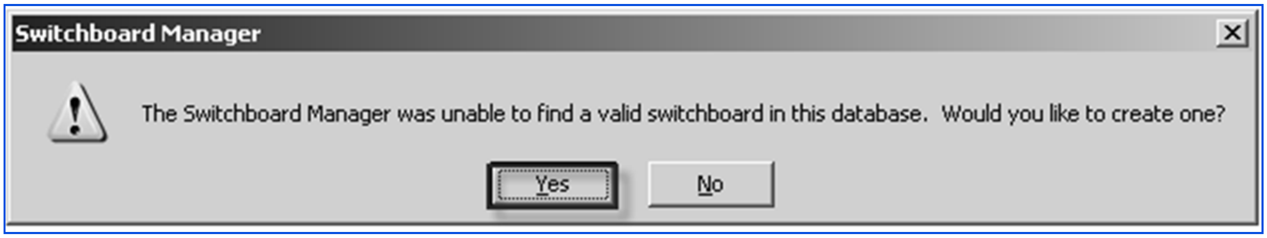


Fig. 4.3.3: Switchboard Manager

NOTE: If the database does not contain any *switchboard*, a message box as indicated in Fig. 4.3.4 appears confirming to create a new *Switchboard*.

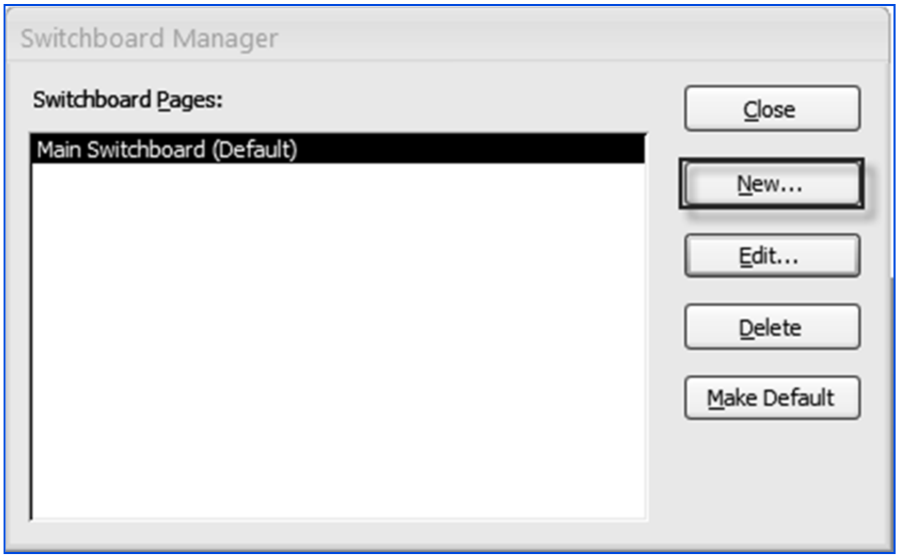


Switchboard Manager

The Switchboard Manager was unable to find a valid switchboard in this database. Would you like to create one?

Fig. 4.3.4: Confirming New Switchboard

1. The *Switchboard Manager* Dialog box appears. Click *New* to create a *Switchboard*. *Switchboard Manager* is displayed as in Fig. 4.3.5.



Switchboard Manager

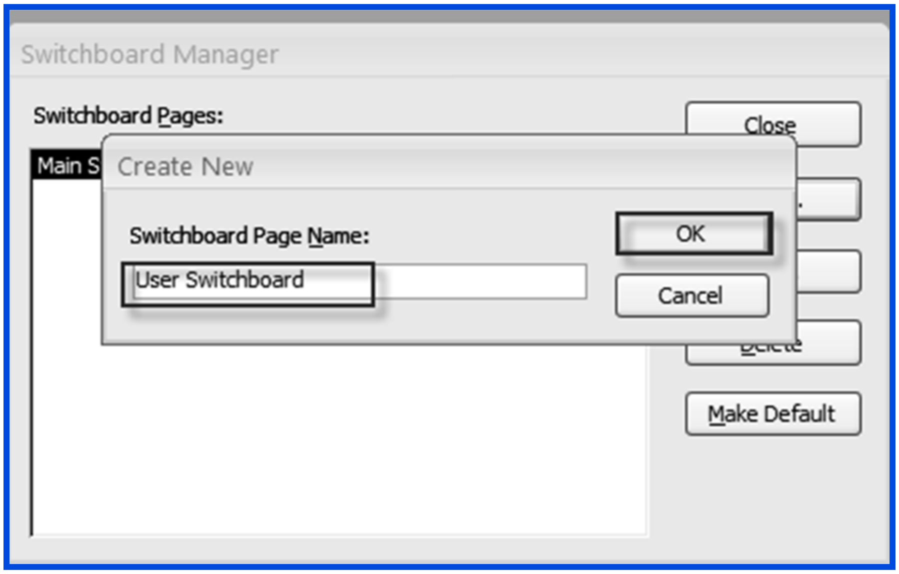
Switchboard Pages:

Main Switchboard (Default)

Fig. 4.3.5: Switchboard Manager Dialog box

NOTE: Instead *of* creating a new *Switchboard*, the default created *switchboard* can also be used.

2. In the *Create New* dialog box, in the *Switchboard Page Name* text box, enter *User Switchboard* and then click *OK* to create a *sub-switchboard* with that name, as shown in Fig. 4.3.6.



Switchboard Page Name: User Switchboard

Fig. 4.3.6: Create New Dialog Box

3. In the *Switchboard Manager* dialog box, in the *Switchboard Pages* section, verify that *User Switchboard* is selected and click *Edit,* as displayed in Fig. 4.3.7.

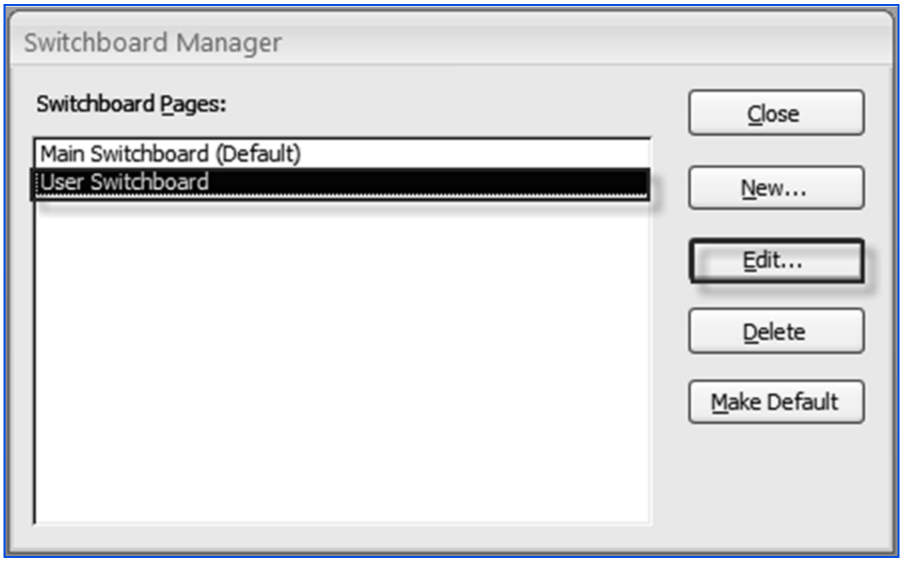
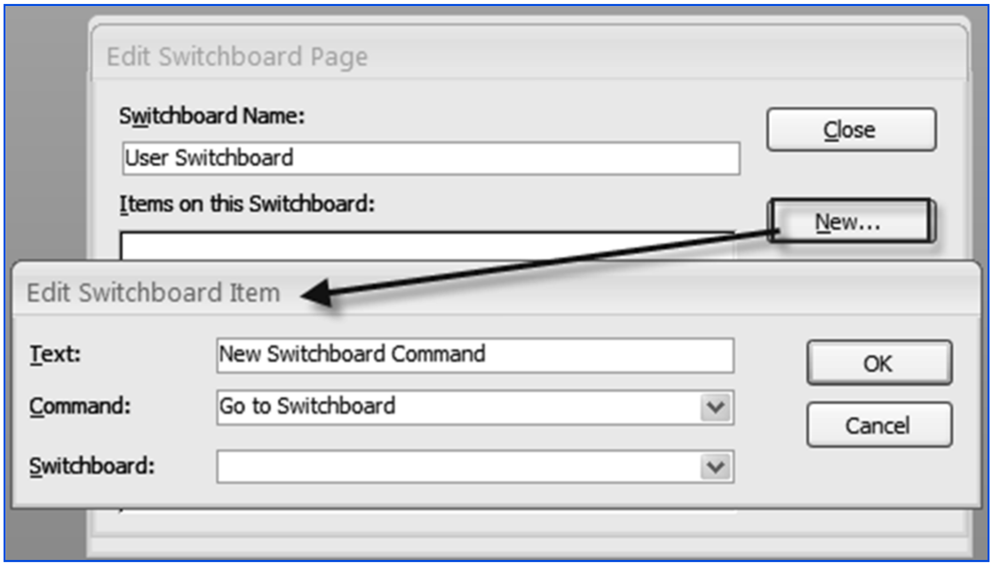


Fig. 4.3.7: Edit the switchboard to add buttons

4. In the *Edit Switchboard Page* dialog box, click *New* to add buttons to *Switchboard*. The *Edit Switchboard Item* dialog box appears, as shown in Fig. 4.3.8.



Switchboard Name: User Switchboard

Items on this Switchboard:

Text: New Switchboard Command

Command: Go to Switchboard

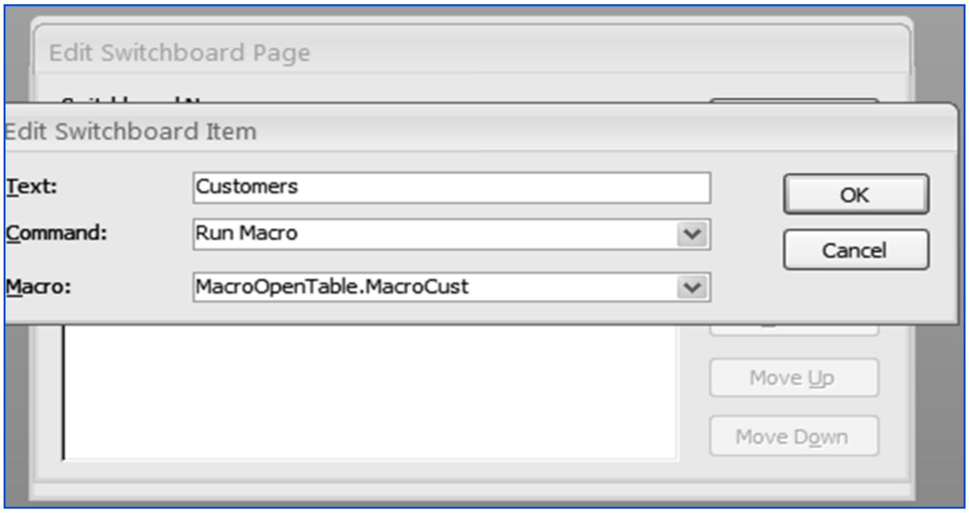
Switchboard:

Fig. 4.3.8: Create a New Button

5. In the *Edit Switchboard Item* dialog box, in the *Text* field, type *Customers* as name of the button.

6. From the *Command* list, select *Run Macro*.

7. From the Form list, select *MacroOpenTable.MacroCust* and click *OK*. The *Edit Switchboard* window appears, as shown in Fig. 4.3.9.



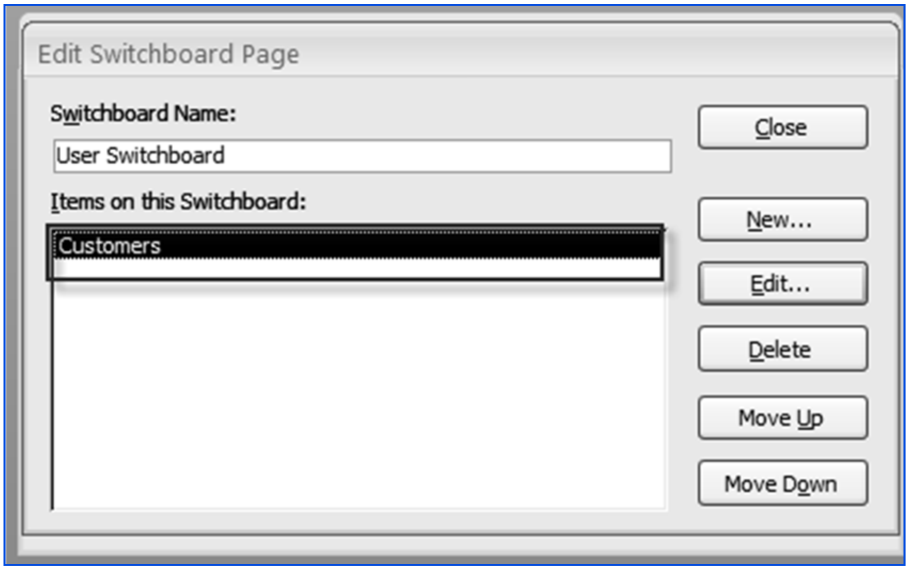
Text: Customers

Command: Run Macro

Macro: MacroOpenTable.MacroCust

Fig. 4.3.9: Provide details of the button

8. Click *OK* to close the *Edit Switchboard* window, the created button appears in *Items* on the *Switchboard* tab, as shown in Fig. 4.3.10.



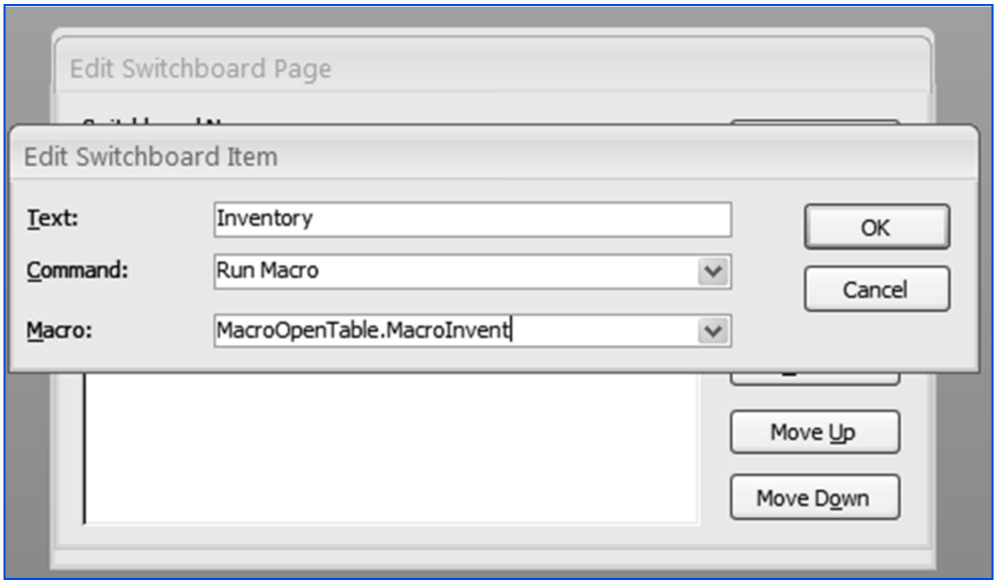
Edit Switchboard Page

Switchboard Name: User Switchboard

Items on this Switchboard: Customers

Fig. 4.3.10: Customers Button

9. Repeat steps 4 - 8 to add another button to the form which opens the *Inventory* table. The final *Edit Switchboard* window appears, as shown in Fig. 4.3.11.



Edit Switchboard Page

Edit Switchboard Item

Text: Inventory

Command: Run Macro

Macro: MacroOpenTable.MacroInvent

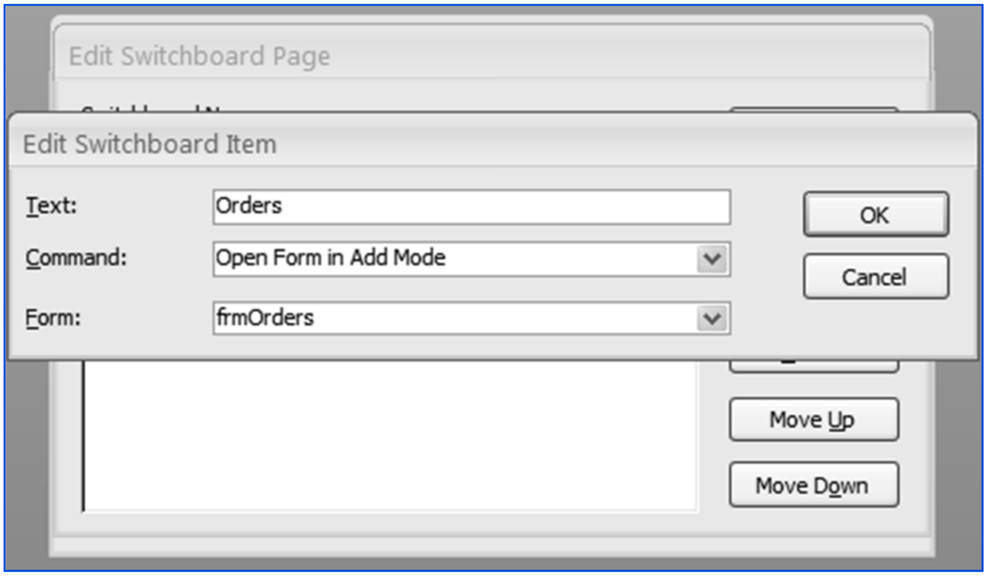
Fig. 4.3.11: Create Button Inventory

Steps to add frmOrders form to the Switchboard

10. In the *Edit Switchboard Page* dialog box, click *New* to add buttons to *Switchboard*. The *Edit Switchboard Item* dialog box appears.

11. In the *Edit Switchboard Item* dialog box, in the *Text* field, type *Orders* as name of the button. From the *Command* list, select *Open Form in Add Mode*.

12. From the *Form* list, select *frmOrders* and click *OK*. The *Edit Switchboard* window appears, as shown in Fig. 4.3.12.



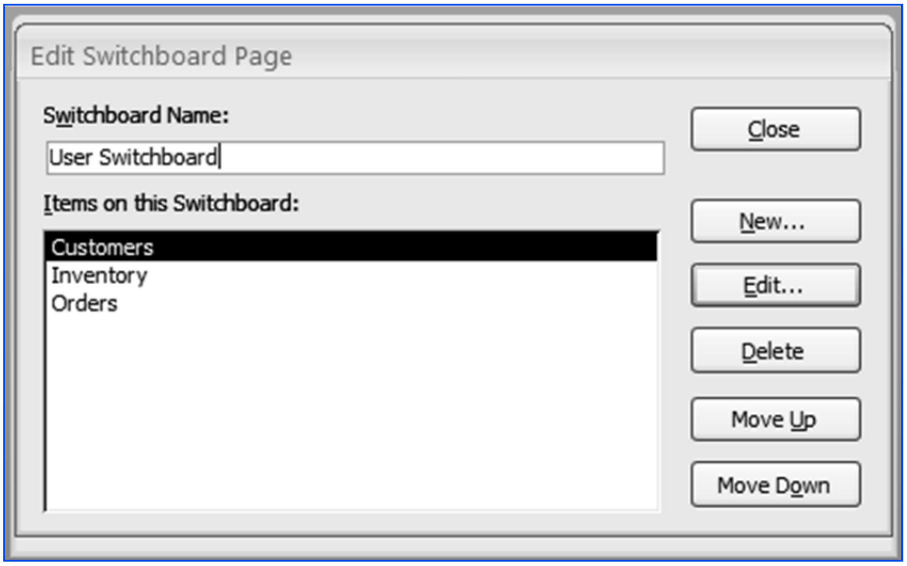
Text: Orders

Command: Open Form in Add Mode

Form: frmOrders

Fig. 4.3.12: Create Orders Button

13. The final *Edit Switchboard* window appears, as shown in Fig. 4.3.13. Click *Close* to close the *Edit Switchboard* window.



Switchboard Name: User Switchboard

Items on this Switchboard:

Customers
Inventory
Orders

Fig. 4.3.13: Final Edit Switchboard window

14. The *Switchboard Manager* window appears. Select *User Switchboard* and click the *Make Default* button. This option will set the *User Switchboard* as default switchboard. The *Switchboard Manager* window appears, as shown in Fig. 4.3.14.

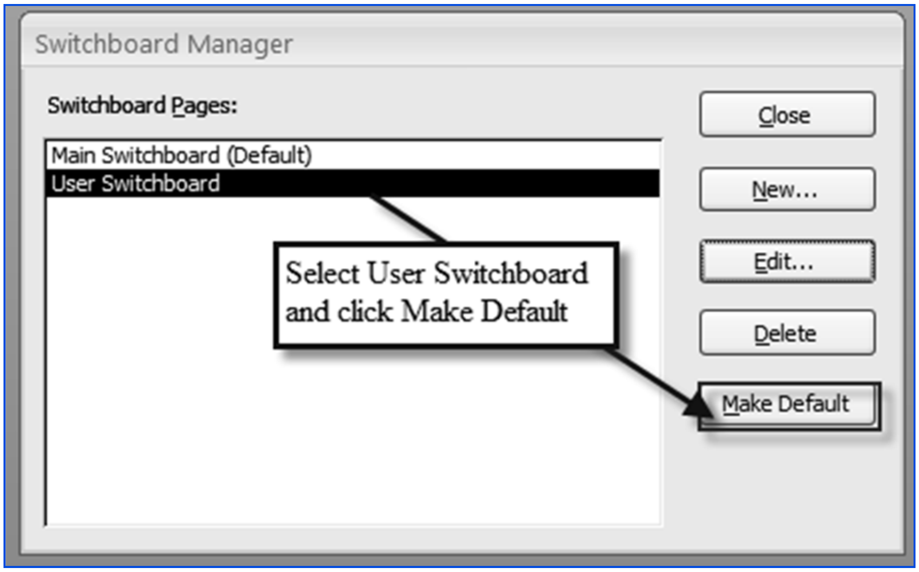


Fig. 4.3.14: Make User Switchboard as Default Switchboard

NOTE: Notice that *Switchboard Manager* has created a new table *Switchboard Items* and a new form *Switchboard*.

15. To test the switchboard created, double-click *Switchboard* under *All Access Objects* -> *Forms* to launch the *User Switchboard* window. The *switchboard form* appears, as shown in Fig. 4.3.15

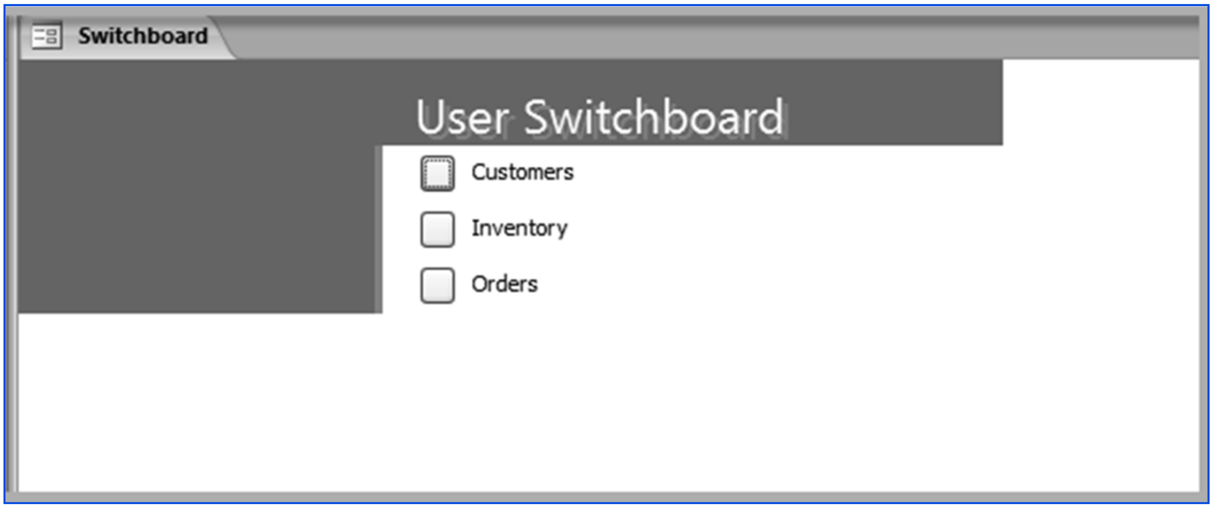


Fig. 4.3.15: User Switchboard

4.3.2 Modify a Database Switchboard

A switchboard created using *Switchboard Manager* can display only the default settings for the elements on the switchboard. *Switchboards* can be made more handy and effective by modifying the properties of a *switchboard*. Access provides different features to customize a *switchboard*.

A *Database Switchboard* can be modified in the *Design* view using the *Property Sheet* pane. The VB Editor can also be used for making switchboard modifications. *Switchboard* can be modified by changing its design, moving controls, adding text, modifying text properties, adding graphics, and changing the control properties. To add or delete buttons to *Switchboard*, the *Edit* option in *Switchboard Manager* is used.

4.3.2.1 Problem Scenario

The developer noticed that the *switchboard* created in section 4.3.15 is not giving a professional look. He wishes to add the *company logo* to the *switchboard*. In addition, he wants to add a new *button* to the *switchboard* which will help the user to create an excel copy of data and mail it to the manager.

Solution

To add a company logo to the *switchboard*, modify the *switchboard* in the *Design view*. To add a new button to the *switchboard*, use the *Edit* option in *Switchboard Manager.* Create a *command button* to execute the *macro MailCurrentOrders* created in section 4.2.1.

Steps for adding a company logo to Switchboard

1. Right-click the *switchboard* under *All Access Objects* -> *Forms* and select the *Design* view from the list to open the switchboard in the *Design* view. The *switchboard* appears, as shown in Fig. 4.3.16.

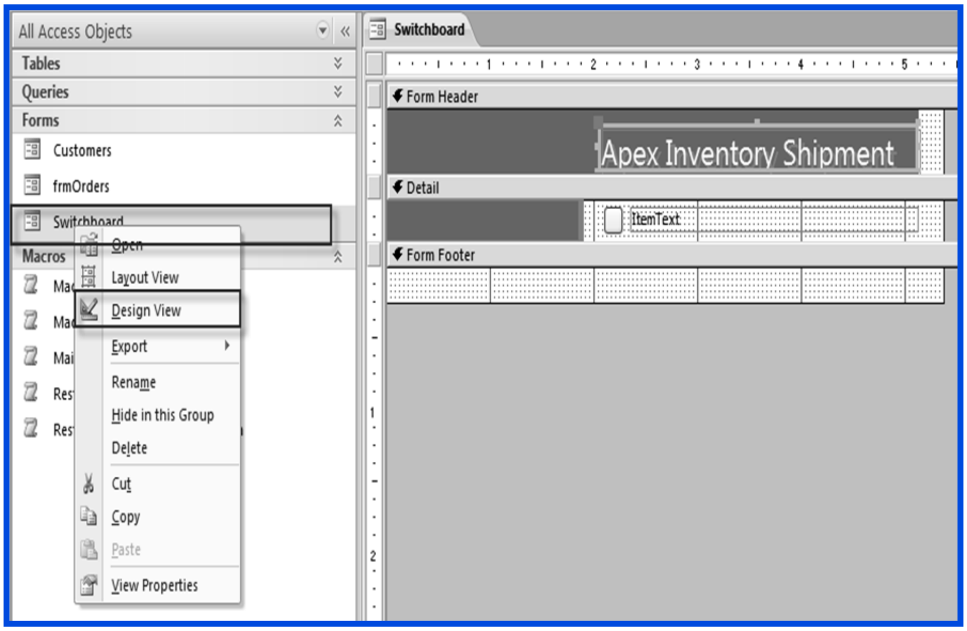


Fig. 4.3.16: Open Switchboard in Design View

2. Click *Design* -> *Controls* -> *logo* button and browse to the *logo image*, as shown in Fig. 4.3.17.

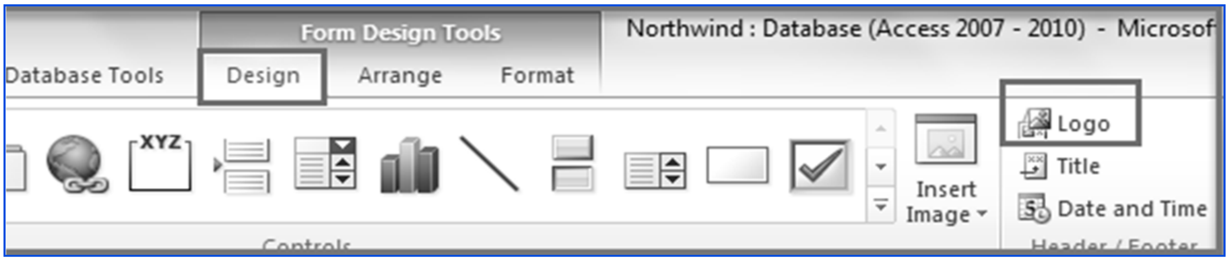


Fig. 4.3.17: Form Design Tools View

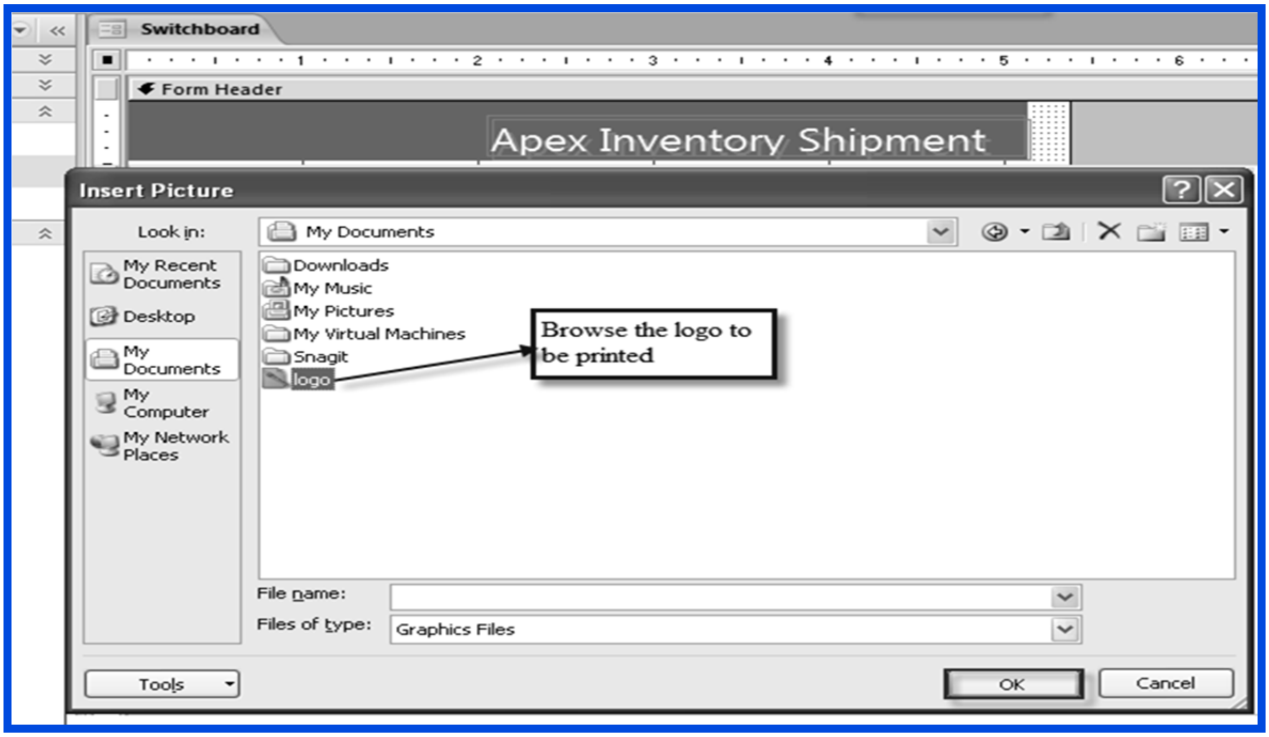


Fig. 4.3.18: Print the desired logo

3. The *logo* appears in the *Switchboard* form, as displayed in Fig. 4.3.19.

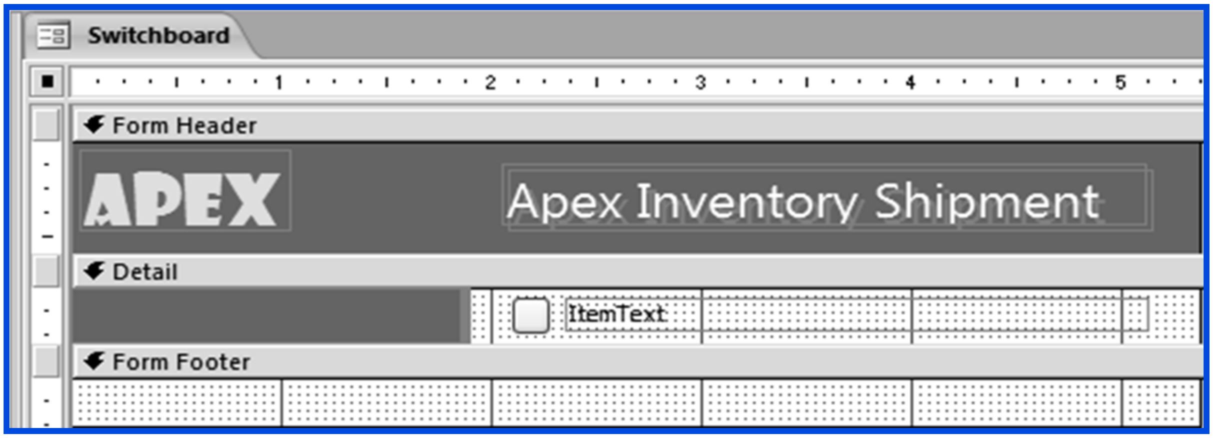
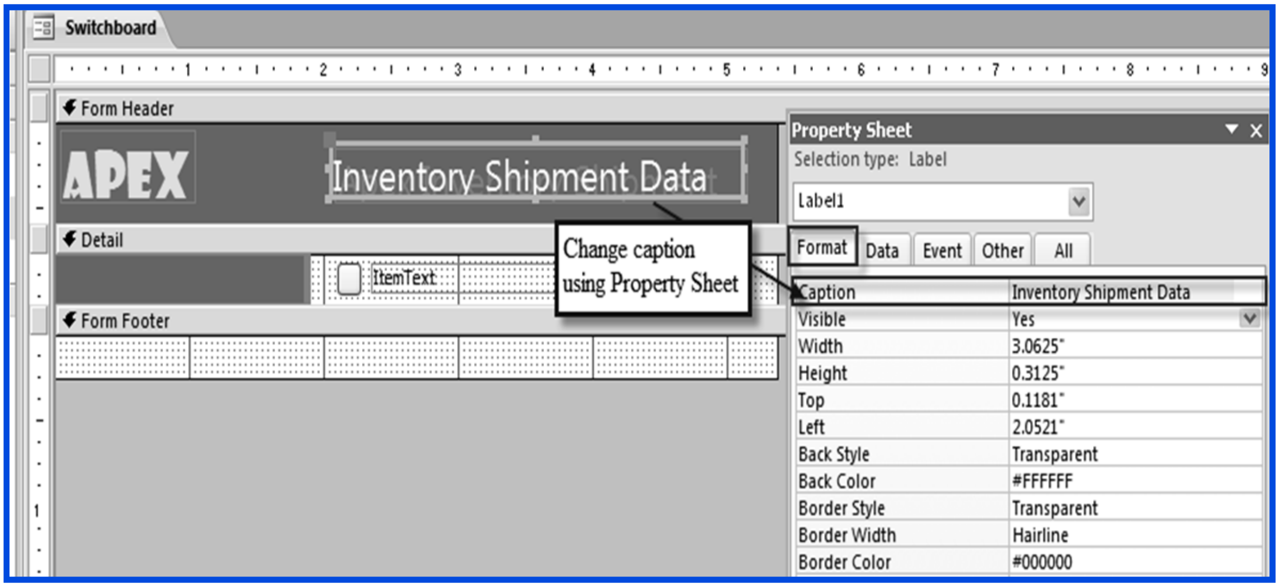


Fig. 4.3.19: The Switchboard form with logo.

Steps to customize the text in Label

1. Select the *Apex Inventory Shipment* label and press *F4* to open the *Property* window.
2. Set the value of caption property as *Inventory Shipment Data*. The *Caption* property is visible under the *Format* tab. The *Property* window appears, as shown in Fig. 4.3.20.



Change caption using Property Sheet

Property Sheet

Selection type: Label

Label1

Caption	Inventory Shipment Data
Visible	Yes
Width	3.0625"
Height	0.3125"
Top	0.1181"
Left	2.0521"
Back Style	Transparent
Back Color	#FFFFFF
Border Style	Transparent
Border Width	Hairline
Border Color	#000000

Fig. 4.3.21: Change Caption of the label

3. Save the *Switchboard* form and open it in the *Form* view to view the changes made. The *Switchboard* form appears, as shown in Fig. 4.3.22.

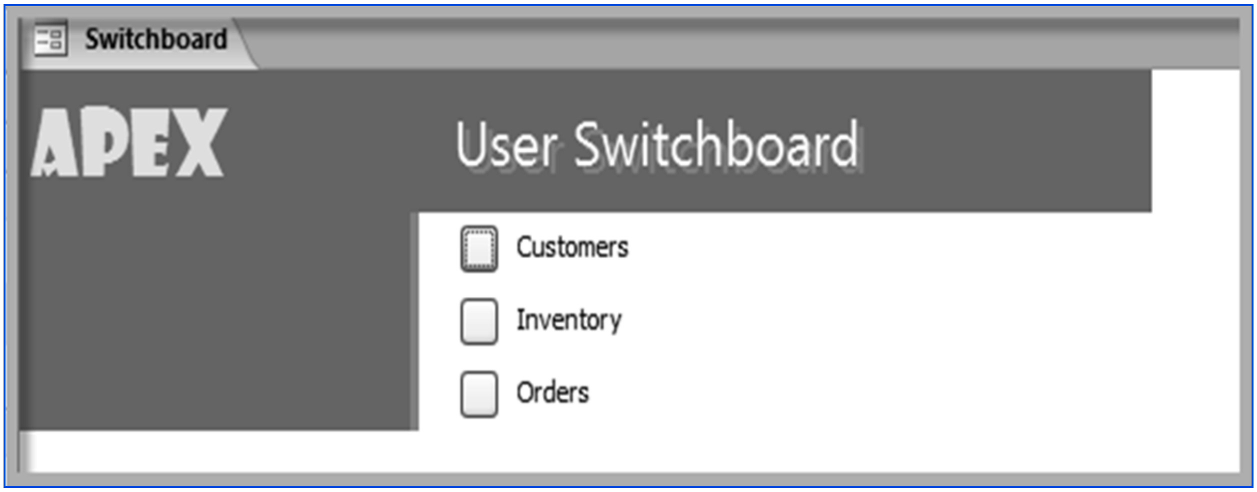


Fig. 4.3.22: Switchboard Form

Steps for adding a new button to the Switchboard Form

4. On the *New Tab or where the Switchboard Manager in your* tab, New Tab, click *Switchboard Manager*.

5. The *Switchboard Manager* Dialog box appears. Select *User Switchboard* and click *Edit* to edit the *switchboard*, as shown in Fig. 4.3.23.

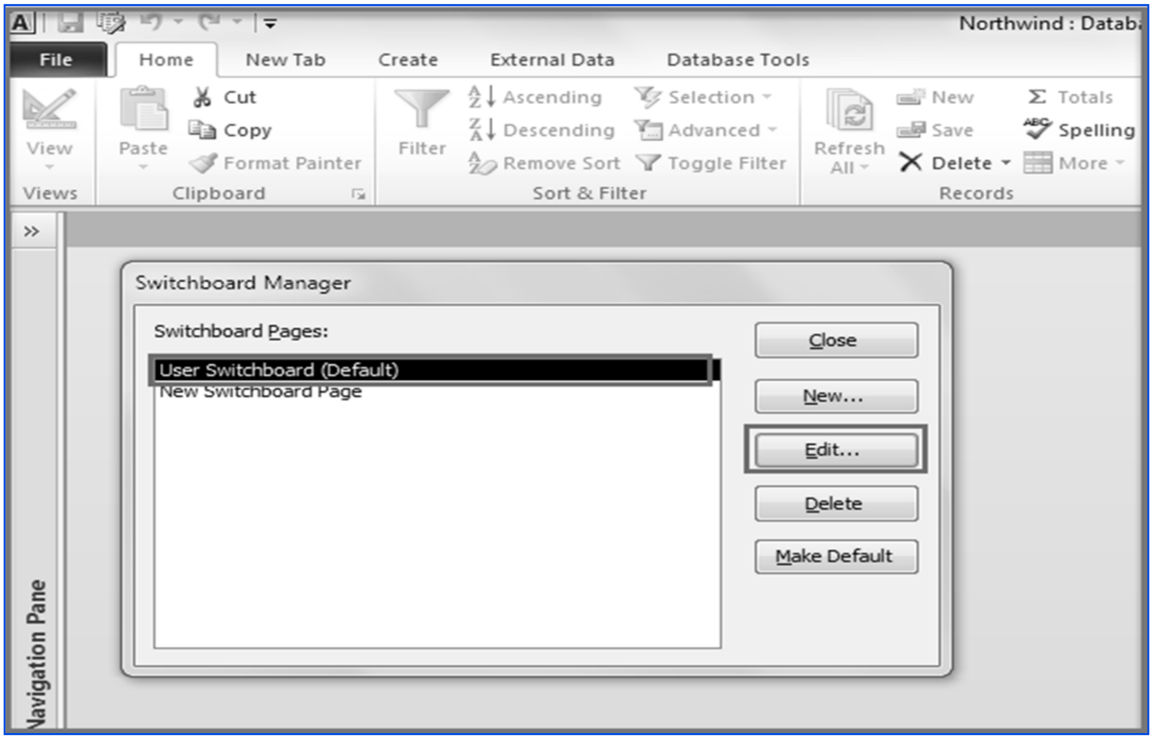
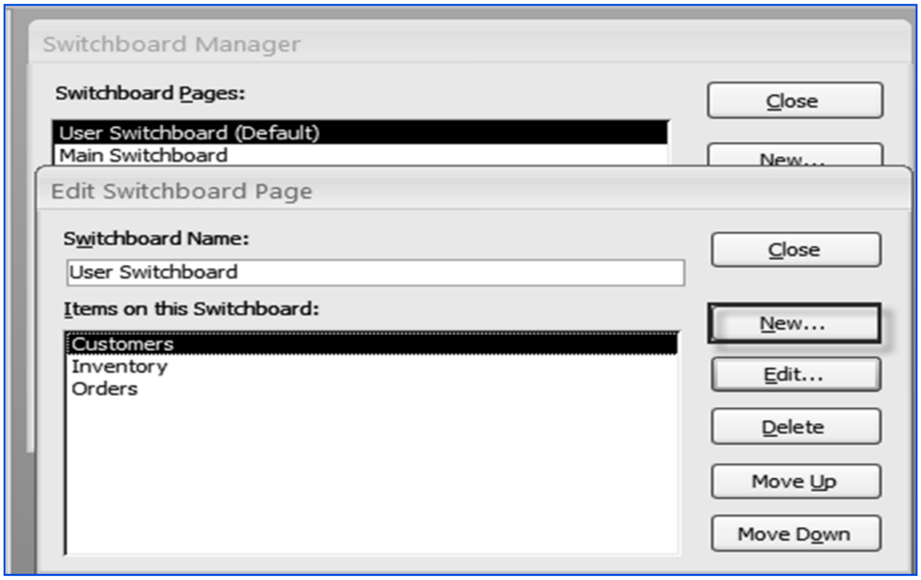


Fig. 4.3.23: Edit the Switchboard User Switchboard

6. *Edit Switchboard Page* appears. Click *New* to create a *button*, as shown in Fig. 4.3.24.



Switchboard Pages:
User Switchboard (Default)
Main Switchboard

Switchboard Name: User Switchboard

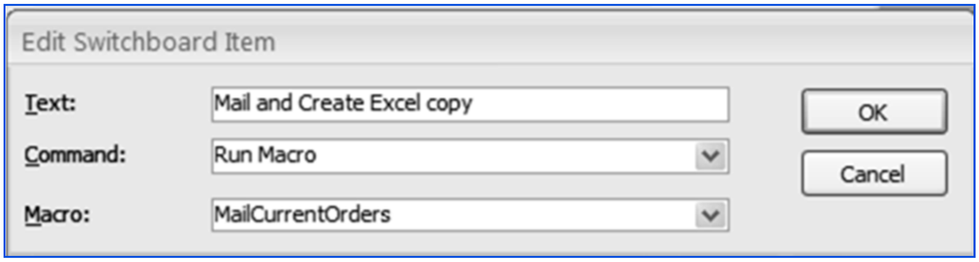
Items on this Switchboard:
Customers
Inventory
Orders

Fig. 4.3.24: Create a new button

7. In the *Edit Switchboard Item* dialog box that appears as a result, in the *Text* field, enter *Mail and Create Excel* copy.

8. From the *Command* list, select *Run Macro*.

9. From the *Form* list, select *MailCurrentOrders* and click *OK*. The *Edit Switchboard* window appears, as shown in Fig. 4.3.25.



Text: Mail and Create Excel copy
Command: Run Macro
Macro: MailCurrentOrders

Fig. 4.3.25: Add a button to run macro MailCurrentOrders

10. Click *OK* to close the *Edit Switchboard* window, the button appears in *Items* on the *Switchboard* tab, as shown in Fig. 4.3.26.

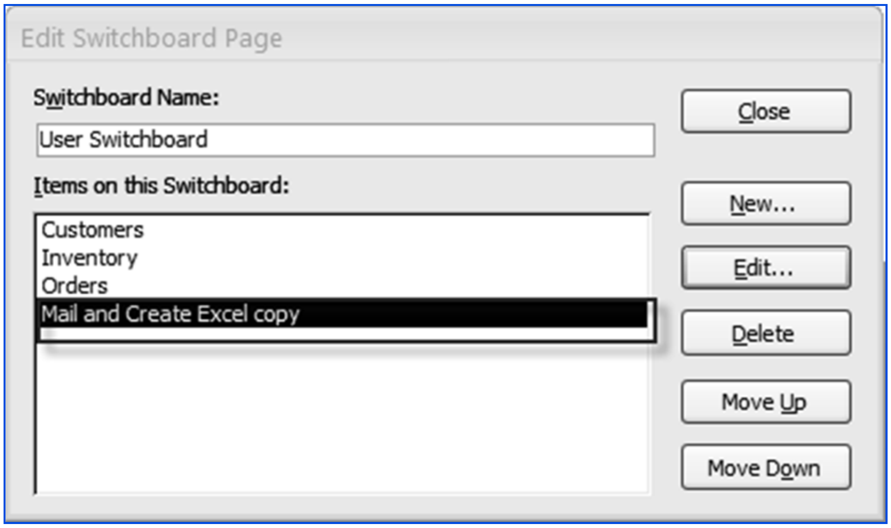


Fig. 4.3.26: Button appears in Edit Switchboard window

11. Close the *Edit Switchboard* window and the *Switchboard Manager* window.

12. To test the *switchboard* created, double-click *Switchboard* under *All Access Objects* -> *Forms* to launch the *User Switchboard* window. The *switchboard* form appears, as shown in Fig. 4.3.27.

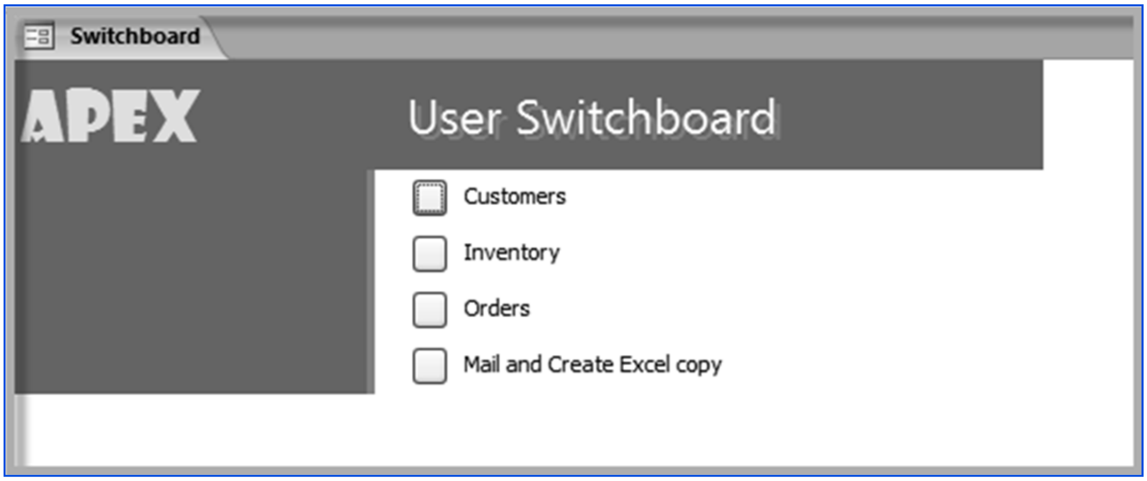


Fig. 4.3.27: Switchboard Form

4.3.3 Setting the Startup Options

Since most database users do not require direct access to the entire application, the *switchboard* can be used as a means to direct the user only to the objects specific to their job role. To make a *switchboard* appear while opening the database so as to make the application more convenient set the startup option of the database. Displaying the *switchboard* at *startup* helps to implement a level of security by hiding the key elements of the interface from the user. This allows users to access the database objects relevant to their tasks.

4.3.3.1 Problem Scenario

The Country Head of Apex Ltd., desired that the database *Apex Inventory Shipment* should be more users friendly. This means that users should not waste time navigating through various objects. The application should guide the user provide only the required functionality.

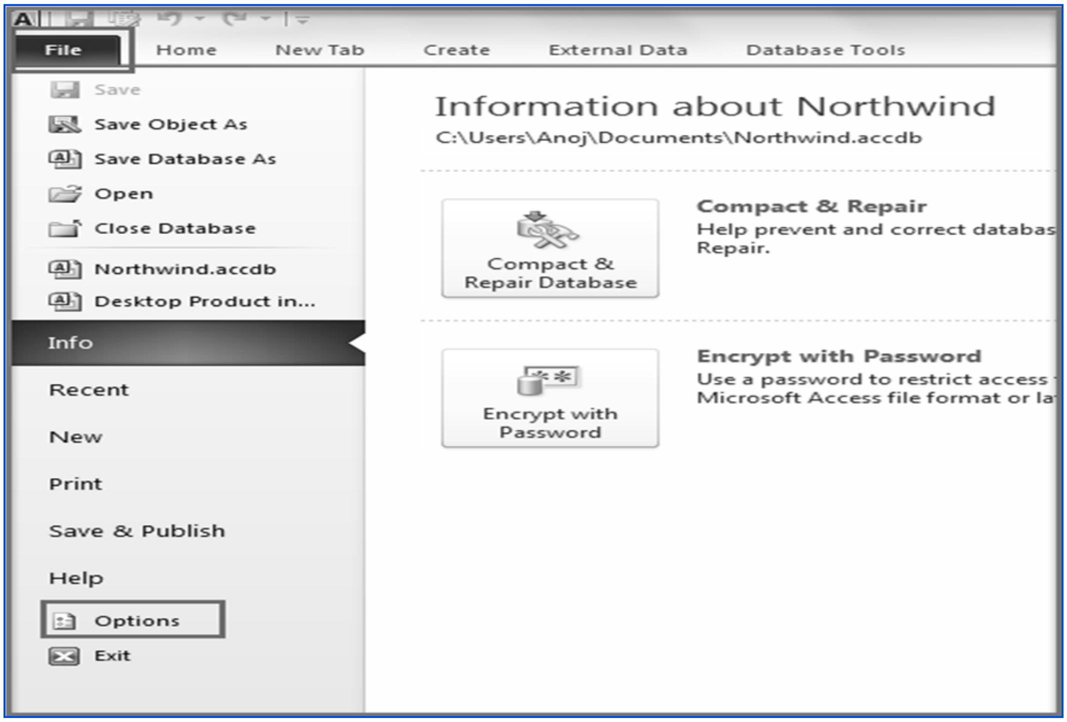
Solution

The *Switchboard* form has already made the application user friendly, with a difference that user needs to open the application search switchboard among the forms and then execute it. This is time-consuming a long process and user may be confused with various options and forms available. To avoid this, set the *switchboard form* at the *Startup option* so that as and when the user opens the application the *Switchboard* form should appear and user may select the required task from it.

Steps to set the switchboard form at startup

1. Open the *Apex Inventory Shipment* database and click File.

2. Select *Access Options* from the list, as shown in Fig. 4.3.28.



Information about Northwind

C:\Users\Anoj\Documents\Northwind.accdb

Compact & Repair

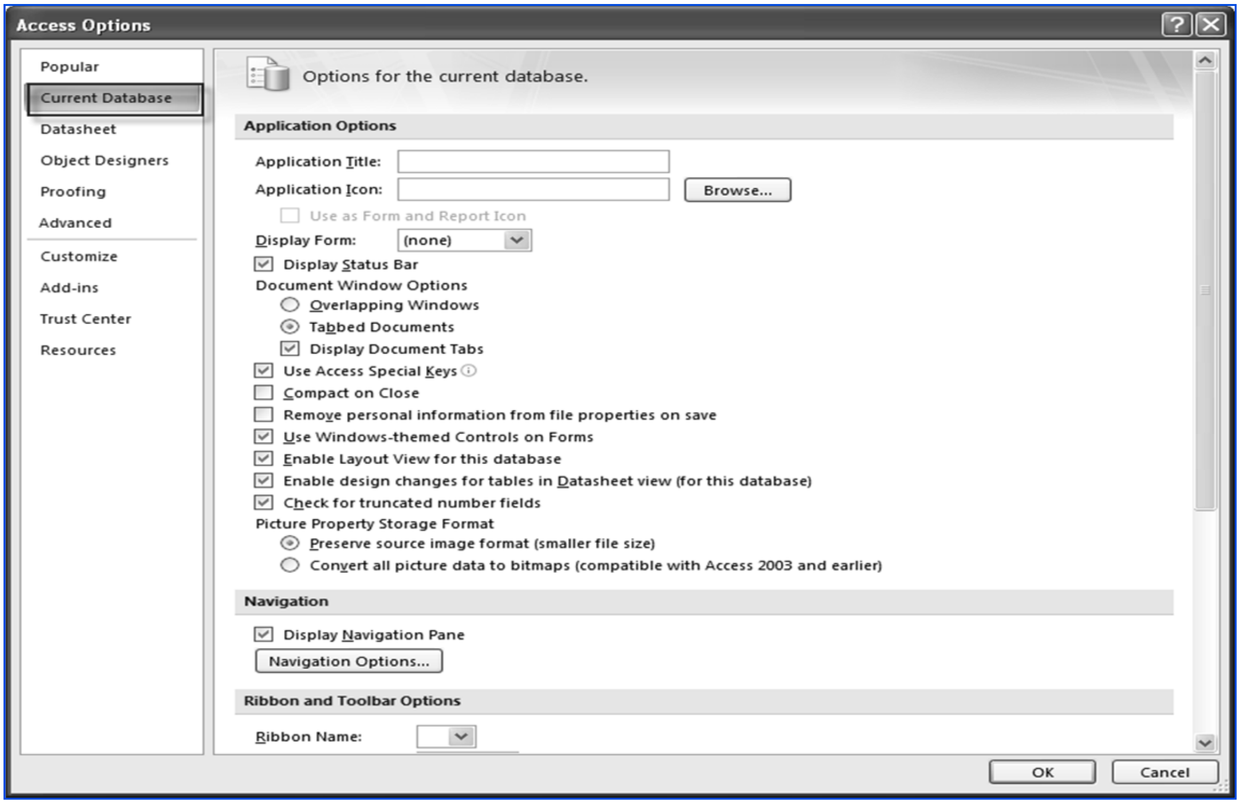
Help prevent and correct databas Repair.

Encrypt with Password

Use a password to restrict access Microsoft Access file format or la

Fig. 4.3.28: Open Access Options

3. The *Access Options* window appears select *Current Database* from the left tab, as shown in Fig. 4.3.29.



Access Options

Options for the current database.

Application Options

Application Title:

Application Icon: Browse...

□ Use as Form and Report Icon

Display Form: (none)

☑ Display Status Bar

Document Window Options

○ Overlapping Windows

● Tabbed Documents

☑ Display Document Tabs

☑ Use Access Special Keys

□ Compact on Close

□ Remove personal information from file properties on save

☑ Use Windows-themed Controls on Forms

☑ Enable Layout View for this database

☑ Enable design changes for tables in Datasheet view (for this database)

☑ Check for truncated number fields

Picture Property Storage Format

● Preserve source image format (smaller file size)

○ Convert all picture data to bitmaps (compatible with Access 2003 and earlier)

Navigation

☑ Display Navigation Pane

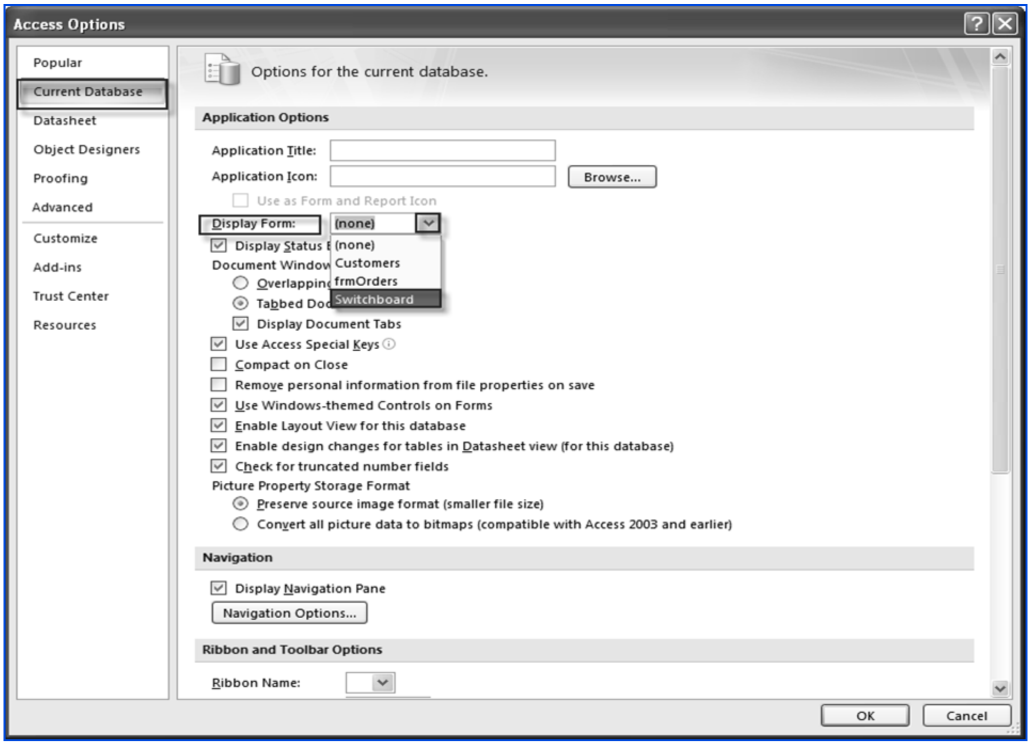
Navigation Options...

Ribbon and Toolbar Options

Ribbon Name:

Fig. 4.3.29: Select Current Database

4. Go to *Application Options*, click ▾ in the *Display* form and select *Switchboard* from the list that appears, as shown in Fig. 4.3.30.



Access Options

Popular
Current Database
Datasheet
Object Designers
Proofing
Advanced
Customize
Add-ins
Trust Center
Resources

Options for the current database.

Application Options

Application Title:
Application Icon: Browse...
☐ Use as Form and Report Icon
Display Form: (none)
(none)
Customers
frmOrders
Switchboard
☑ Display Status
Document Window
○ Overlappin
● Tabbed Doc
☑ Display Document Tabs
☑ Use Access Special Keys
☐ Compact on Close
☐ Remove personal information from file properties on save
☑ Use Windows-themed Controls on Forms
☑ Enable Layout View for this database
☑ Enable design changes for tables in Datasheet view (for this database)
☑ Check for truncated number fields
Picture Property Storage Format
● Preserve source image format (smaller file size)
○ Convert all picture data to bitmaps (compatible with Access 2003 and earlier)

Navigation

☑ Display Navigation Pane
Navigation Options...

Ribbon and Toolbar Options

Ribbon Name:

OK Cancel

Fig. 4.3.30. Select Switchboard for Display Form option

5. Click *OK* to close the window.

6. To test the *startup option*, close and reopen the database. The *Switchboard* form should display automatically.

NOTE: Other than *switchboard*, any other form can also be set as a *database startup option*. So it helps to create a *login form* or a *switchboard form* and launch it at the startup of database.

4.4 Summary

Macros can be created from simple to complex with one or multiple actions. Macro can be executed as a separate object or can be attached to an event property of form, controls, or reports. The attached macros are executed when the event occurs. Macro can also be useful in restricting the number of records to retrieve using the *Where* clause available with certain actions. Using macro names, can help one macro object hold many macros. Conditions can also be specified with macros to validate the data entry. Macros can also be embedded with the control so that it is stored in the control and moves around with the control.

Switchboards can help to navigate between the various database objects easily and manage data more effectively. The switchboard is essentially a steering wheel for users to find their way through the functions and forms that are available in the application. The switchboard is used as a navigation form, using buttons to display other forms. Switchboards are created using *Switchboard Manager*. A *Switchboard Manager* creates a switchboard form and a switchboard items table automatically. Switchboards can be set as database startup options so they should be displayed as the database is opened and can guide the user through proper options.

4.5 Lab Exercises

Consider the *Apex Inventory Shipment* database of Apex Ltd. Provide a solution to below problems.

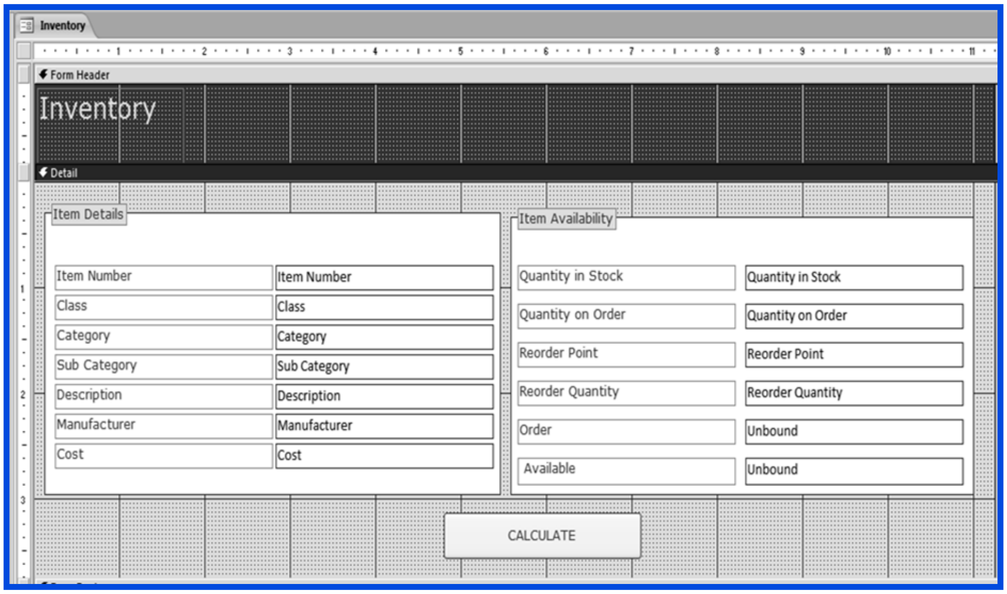
1. The executive from California has sent the details of all the Orders received today as an Excel 2007 (.xlsx) file named Orders_CA. Create a macro to convert the file to Access table and append the records to the Orders table.

   (NOTE: To append the records, create a query and execute it through macros.)

2. The executive in the Operations department are using the frmOrders form to view the orders placed. They want a functionality that the orders should be displayed only for the given period and the date should be accepted each time when they open the form. Create a macro to fulfill the requirement. (Use the Where clause)

3. The operations manager is facing a problem in checking whether the Items ordered are available or not. He wishes to have a button on frmOrders which should display the details of the Items requested in Order. Create a button on frmOrders and attach a macro to get desired functionality.

4. A form was created to track Inventory, as shown in the Fig. below. Attach a macro to the *Calculate* button that should make an auto entry to the field Order and Available. The order field should be set to *Yes* if sum of Quantity in Stock and Quantity on Order is less than Reorder Point. The Available Point should be set to *Yes* if the sum of Quantity in Stock and Quantity on Order is greater than Quantity ordered in Orders table.



Inventory Form

5. Consider the Inventory form created in previous example, create a macro to ensure that Category and Sub Category of an Item should not be left blank and Quantity on Order should not be less than Reorder Quantity. Display customized message if the validation is not followed.

6. The operations manager wishes to have the application customized for the users of Operations department. The operations manager asked the developer to create a switchboard form that should contain the link to open Inventory table, also it should display the Sales Orders along with their Invoices. Create a switchboard for the purpose.

7. Modify the above created switchboard so as when users selects inventory option another switchboard must be opened. This switchboard gives user option to display the entire Inventory, Inventory from a particular category and Items for which Quantity in Stock is more than 100.

8. Set the Switchboard created in Question No. 6 to the startup option of database so that it opens automatically as user opens the database.

9. Add a button to the *Inventory* form created in Question 4. The button should display all the orders from Sales Order Description table that were placed for the particular item. The item must be the item that is displayed currently in the form. (Attach a macro to the button for the purpose).

10. Edit the switchboard in Question No. 7. Add a button to the switchboard that should open the Inventory form.