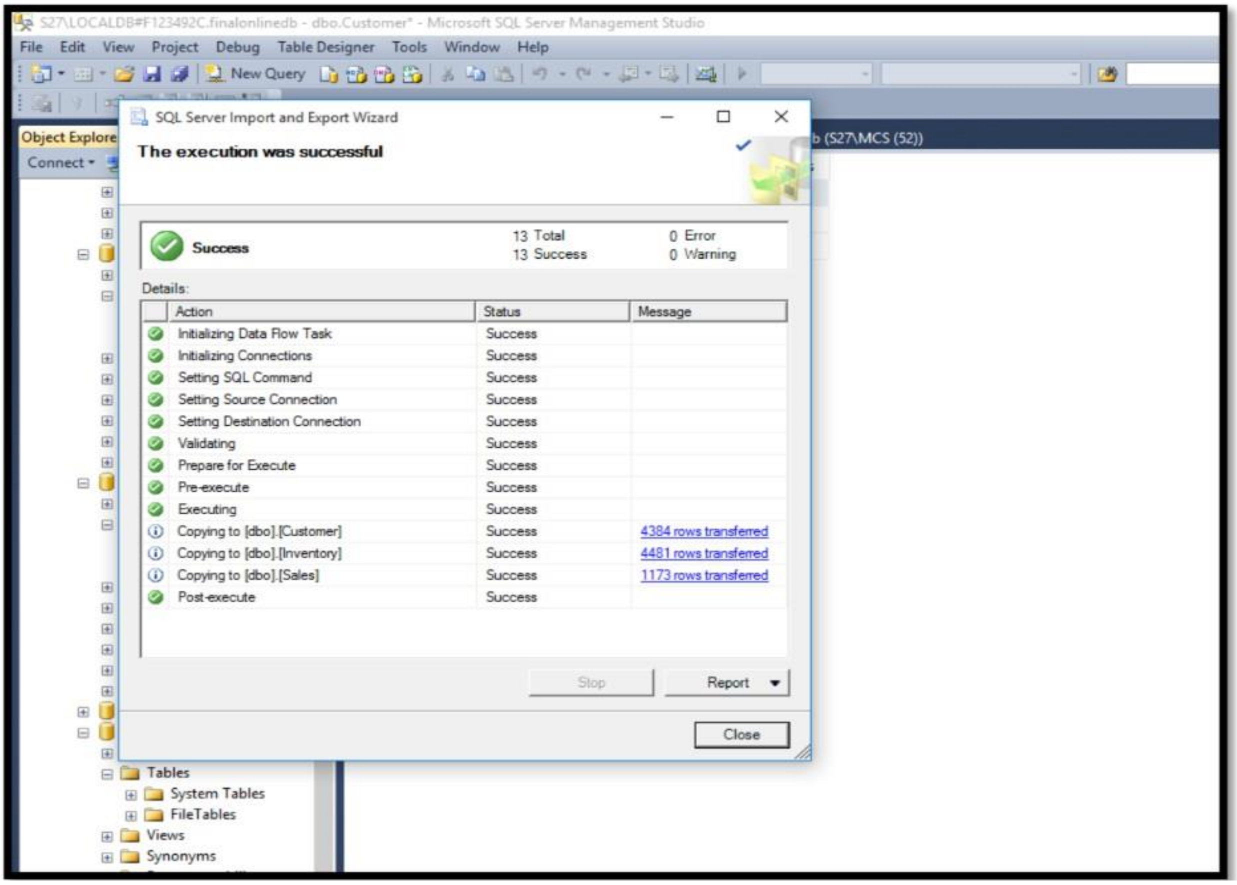
“ONLINE RETAIL II”

“IMPORTING DATASET”



S27\LOCALDB#F123492C.finalonlinedb - dbo.Customer* - Microsoft SQL Server Management Studio
File Edit View Project Debug Table Designer Tools Window Help
New Query
Object Explorer
Connect
Tables
System Tables
FileTables
Views
Synonyms
b (S27\MCS (52))

SQL Server Import and Export Wizard

The execution was successful

Success

13 Total
13 Success

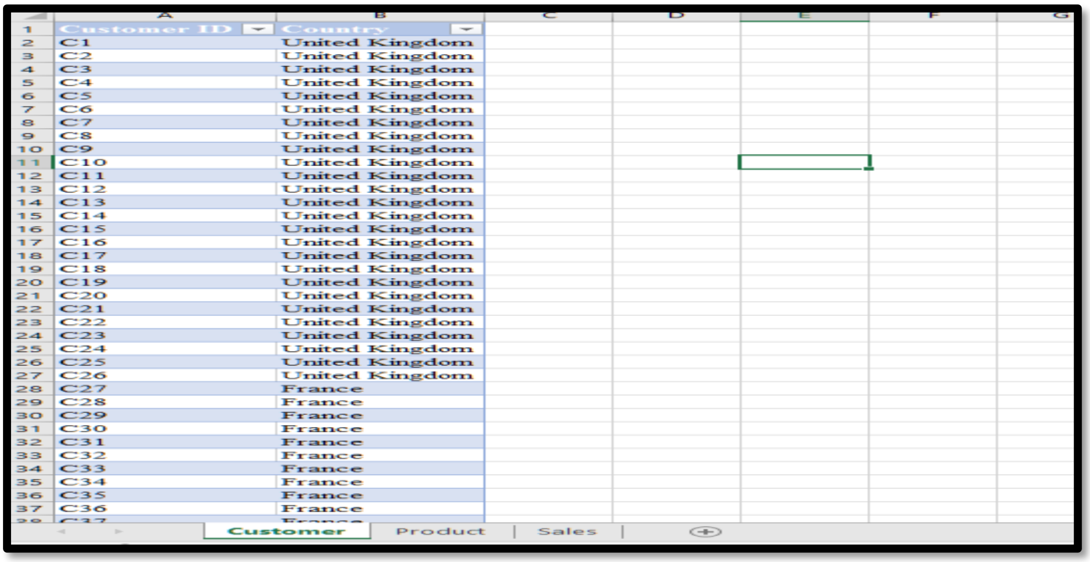
0 Error
0 Warning

Details:

Action	Status	Message
Initializing Data Flow Task	Success	
Initializing Connections	Success	
Setting SQL Command	Success	
Setting Source Connection	Success	
Setting Destination Connection	Success	
Validating	Success	
Prepare for Execute	Success	
Pre-execute	Success	
Executing	Success	
Copying to [dbo].[Customer]	Success	4384 rows transferred
Copying to [dbo].[Inventory]	Success	4481 rows transferred
Copying to [dbo].[Sales]	Success	1173 rows transferred
Post-execute	Success	

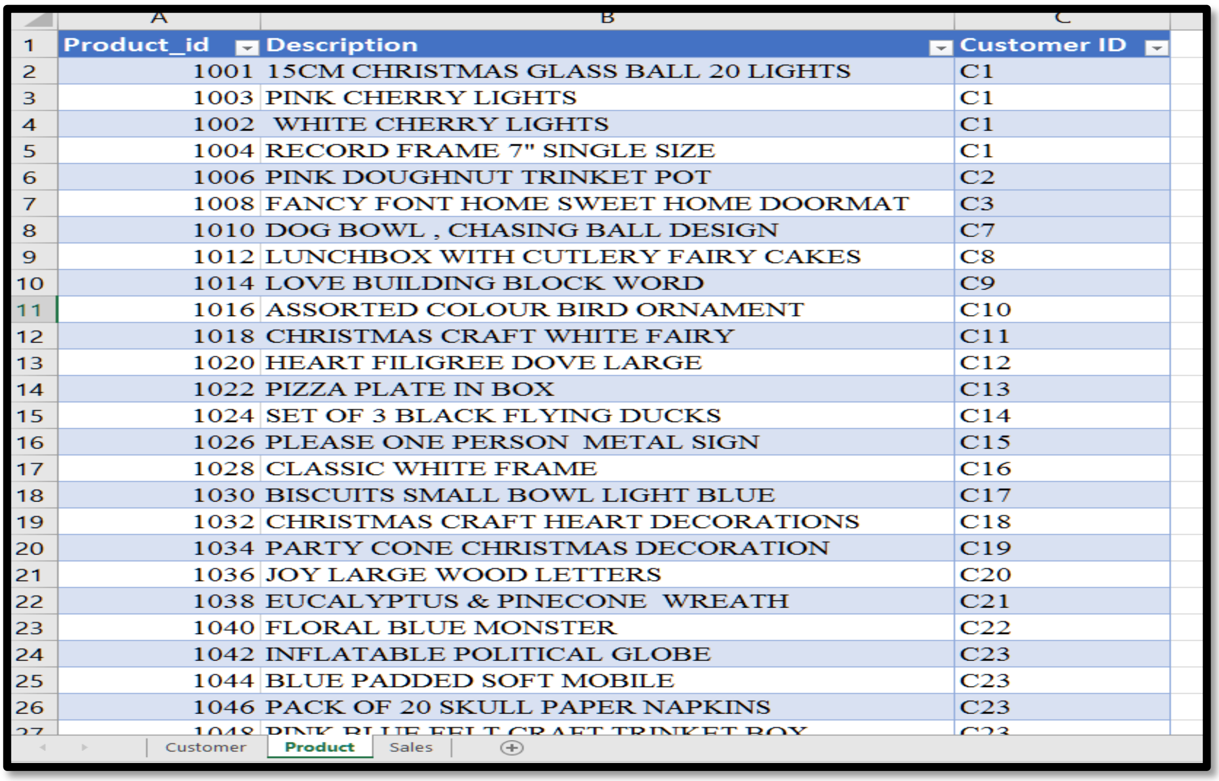
“DATASET TABLES”

CUSTOMER TABLE:



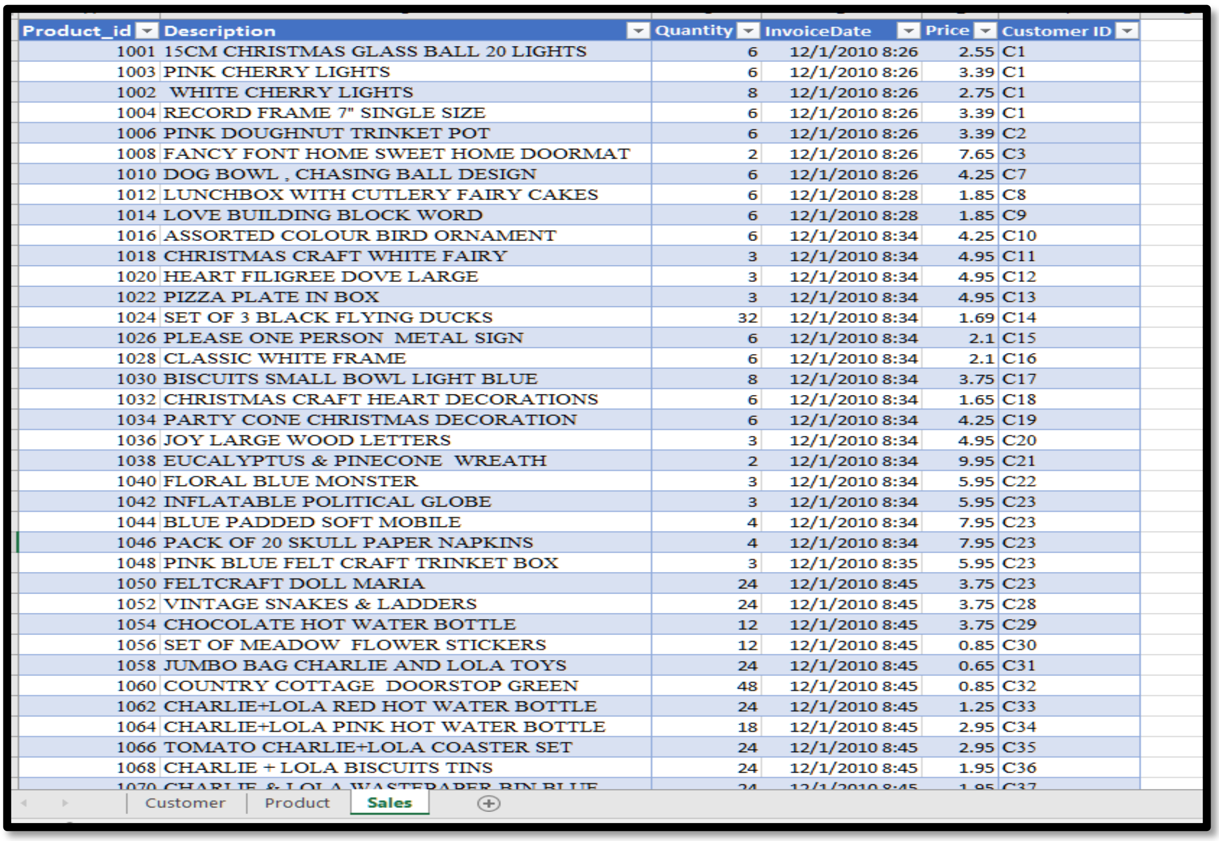
Customer ID	Country
C1	United Kingdom
C2	United Kingdom
C3	United Kingdom
C4	United Kingdom
C5	United Kingdom
C6	United Kingdom
C7	United Kingdom
C8	United Kingdom
C9	United Kingdom
C10	United Kingdom
C11	United Kingdom
C12	United Kingdom
C13	United Kingdom
C14	United Kingdom
C15	United Kingdom
C16	United Kingdom
C17	United Kingdom
C18	United Kingdom
C19	United Kingdom
C20	United Kingdom
C21	United Kingdom
C22	United Kingdom
C23	United Kingdom
C24	United Kingdom
C25	United Kingdom
C26	United Kingdom
C27	France
C28	France
C29	France
C30	France
C31	France
C32	France
C33	France
C34	France
C35	France
C36	France
C37	France

PRODUCT TABLE:



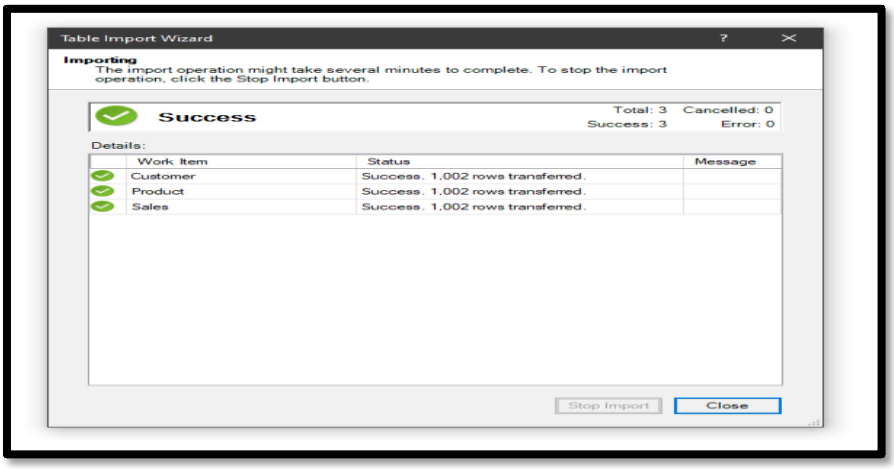
	A	B	C
1	Product_id	Description	Customer ID
2	1001	15CM CHRISTMAS GLASS BALL 20 LIGHTS	C1
3	1003	PINK CHERRY LIGHTS	C1
4	1002	WHITE CHERRY LIGHTS	C1
5	1004	RECORD FRAME 7" SINGLE SIZE	C1
6	1006	PINK DOUGHNUT TRINKET POT	C2
7	1008	FANCY FONT HOME SWEET HOME DOORMAT	C3
8	1010	DOG BOWL , CHASING BALL DESIGN	C7
9	1012	LUNCHBOX WITH CUTLERY FAIRY CAKES	C8
10	1014	LOVE BUILDING BLOCK WORD	C9
11	1016	ASSORTED COLOUR BIRD ORNAMENT	C10
12	1018	CHRISTMAS CRAFT WHITE FAIRY	C11
13	1020	HEART FILIGREE DOVE LARGE	C12
14	1022	PIZZA PLATE IN BOX	C13
15	1024	SET OF 3 BLACK FLYING DUCKS	C14
16	1026	PLEASE ONE PERSON METAL SIGN	C15
17	1028	CLASSIC WHITE FRAME	C16
18	1030	BISCUITS SMALL BOWL LIGHT BLUE	C17
19	1032	CHRISTMAS CRAFT HEART DECORATIONS	C18
20	1034	PARTY CONE CHRISTMAS DECORATION	C19
21	1036	JOY LARGE WOOD LETTERS	C20
22	1038	EUCALYPTUS & PINECONE WREATH	C21
23	1040	FLORAL BLUE MONSTER	C22
24	1042	INFLATABLE POLITICAL GLOBE	C23
25	1044	BLUE PADDED SOFT MOBILE	C23
26	1046	PACK OF 20 SKULL PAPER NAPKINS	C23
27	1048	PINK BLUE FELT CRAFT TRINKET BOX	C23

SALES TABLE:



Product_id	Description	Quantity	InvoiceDate	Price	Customer ID
1001	15CM CHRISTMAS GLASS BALL 20 LIGHTS	6	12/1/2010 8:26	2.55	C1
1003	PINK CHERRY LIGHTS	6	12/1/2010 8:26	3.39	C1
1002	WHITE CHERRY LIGHTS	8	12/1/2010 8:26	2.75	C1
1004	RECORD FRAME 7" SINGLE SIZE	6	12/1/2010 8:26	3.39	C1
1006	PINK DOUGHNUT TRINKET POT	6	12/1/2010 8:26	3.39	C2
1008	FANCY FONT HOME SWEET HOME DOORMAT	2	12/1/2010 8:26	7.65	C3
1010	DOG BOWL , CHASING BALL DESIGN	6	12/1/2010 8:26	4.25	C7
1012	LUNCHBOX WITH CUTLERY FAIRY CAKES	6	12/1/2010 8:28	1.85	C8
1014	LOVE BUILDING BLOCK WORD	6	12/1/2010 8:28	1.85	C9
1016	ASSORTED COLOUR BIRD ORNAMENT	6	12/1/2010 8:34	4.25	C10
1018	CHRISTMAS CRAFT WHITE FAIRY	3	12/1/2010 8:34	4.95	C11
1020	HEART FILIGREE DOVE LARGE	3	12/1/2010 8:34	4.95	C12
1022	PIZZA PLATE IN BOX	3	12/1/2010 8:34	4.95	C13
1024	SET OF 3 BLACK FLYING DUCKS	32	12/1/2010 8:34	1.69	C14
1026	PLEASE ONE PERSON METAL SIGN	6	12/1/2010 8:34	2.1	C15
1028	CLASSIC WHITE FRAME	6	12/1/2010 8:34	2.1	C16
1030	BISCUITS SMALL BOWL LIGHT BLUE	8	12/1/2010 8:34	3.75	C17
1032	CHRISTMAS CRAFT HEART DECORATIONS	6	12/1/2010 8:34	1.65	C18
1034	PARTY CONE CHRISTMAS DECORATION	6	12/1/2010 8:34	4.25	C19
1036	JOY LARGE WOOD LETTERS	3	12/1/2010 8:34	4.95	C20
1038	EUCALYPTUS & PINECONE WREATH	2	12/1/2010 8:34	9.95	C21
1040	FLORAL BLUE MONSTER	3	12/1/2010 8:34	5.95	C22
1042	INFLATABLE POLITICAL GLOBE	3	12/1/2010 8:34	5.95	C23
1044	BLUE PADDED SOFT MOBILE	4	12/1/2010 8:34	7.95	C23
1046	PACK OF 20 SKULL PAPER NAPKINS	4	12/1/2010 8:34	7.95	C23
1048	PINK BLUE FELT CRAFT TRINKET BOX	3	12/1/2010 8:35	5.95	C23
1050	FELTCRAFT DOLL MARIA	24	12/1/2010 8:45	3.75	C23
1052	VINTAGE SNAKES & LADDERS	24	12/1/2010 8:45	3.75	C28
1054	CHOCOLATE HOT WATER BOTTLE	12	12/1/2010 8:45	3.75	C29
1056	SET OF MEADOW FLOWER STICKERS	12	12/1/2010 8:45	0.85	C30
1058	JUMBO BAG CHARLIE AND LOLA TOYS	24	12/1/2010 8:45	0.65	C31
1060	COUNTRY COTTAGE DOORSTOP GREEN	48	12/1/2010 8:45	0.85	C32
1062	CHARLIE+LOLA RED HOT WATER BOTTLE	24	12/1/2010 8:45	1.25	C33
1064	CHARLIE+LOLA PINK HOT WATER BOTTLE	18	12/1/2010 8:45	2.95	C34
1066	TOMATO CHARLIE+LOLA COASTER SET	24	12/1/2010 8:45	2.95	C35
1068	CHARLIE + LOLA BISCUITS TINS	24	12/1/2010 8:45	1.95	C36
1070	CHARLIE & LOLA WASTEPAPER BIN BLUE	24	12/1/2010 8:45	1.95	C37

“RELATIONSHIPS”

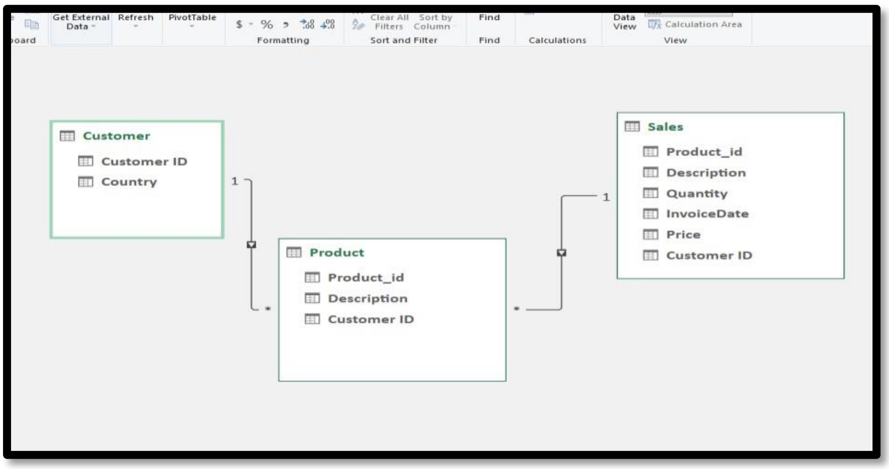


Importing
The import operation might take several minutes to complete. To stop the import operation, click the Stop Import button.

Success — Total: 3 Cancelled: 0 Success: 3 Error: 0

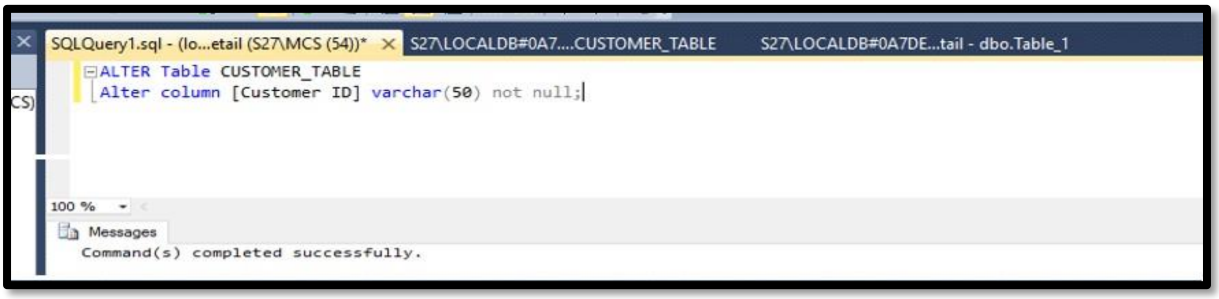
Details:

Work Item	Status	Message
Customer	Success. 1,002 rows transferred.	
Product	Success. 1,002 rows transferred.	
Sales	Success. 1,002 rows transferred.	



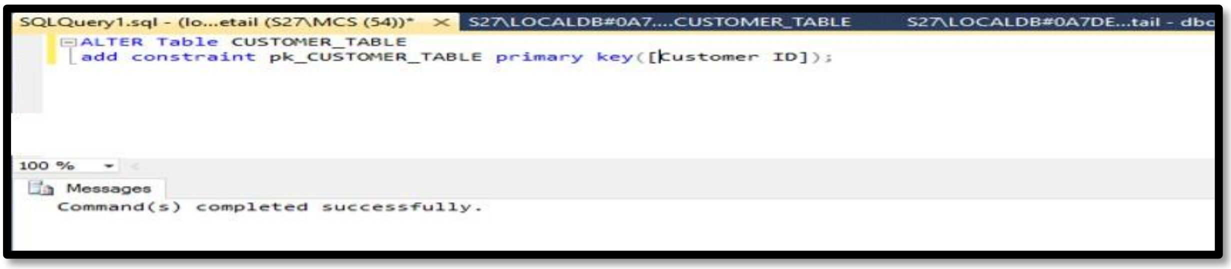
“SETTING PRIMARY AND FOREIGN KEYS”

CUSTOMER TABLE:


SQLQuery1.sql - (lo...etail (S27\MCS (54))*
S27\LOCALDB#0A7....CUSTOMER_TABLE
S27\LOCALDB#0A7DE...tail - dbo.Table_1
100 %
Messages

```
ALTER Table CUSTOMER_TABLE
Alter column [Customer ID] varchar(50) not null;
```

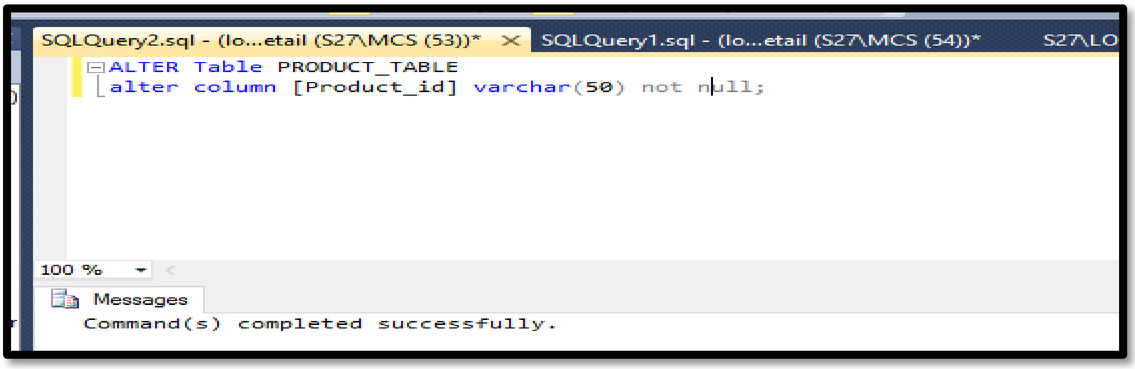
```
Command(s) completed successfully.
```


SQLQuery1.sql - (lo...etail (S27\MCS (54))*
S27\LOCALDB#0A7....CUSTOMER_TABLE
S27\LOCALDB#0A7DE...tail - dbo
100 %
Messages

```
ALTER Table CUSTOMER_TABLE
add constraint pk_CUSTOMER_TABLE primary key([Customer ID]);
```

```
Command(s) completed successfully.
```

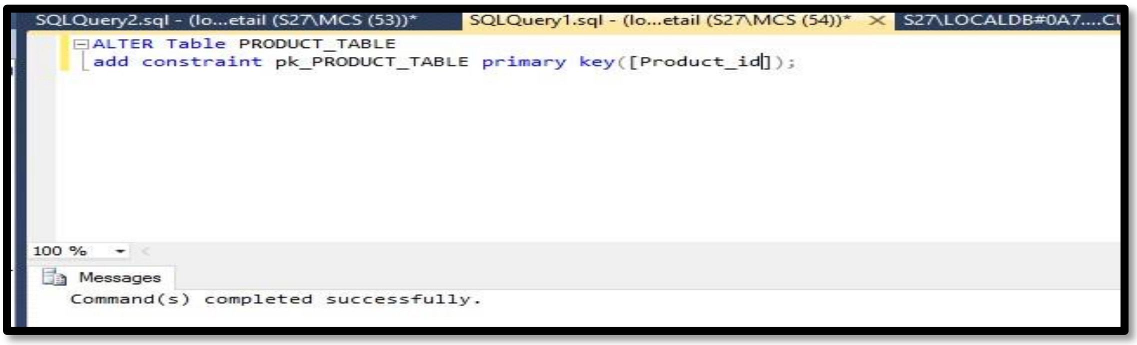
PRODUCT TABLE:



SQLQuery2.sql - (lo...etail (S27\MCS (53))*
SQLQuery1.sql - (lo...etail (S27\MCS (54))*
S27\LO
100 %
Messages

```sql
ALTER Table PRODUCT_TABLE
alter column [Product_id] varchar(50) not null;
```

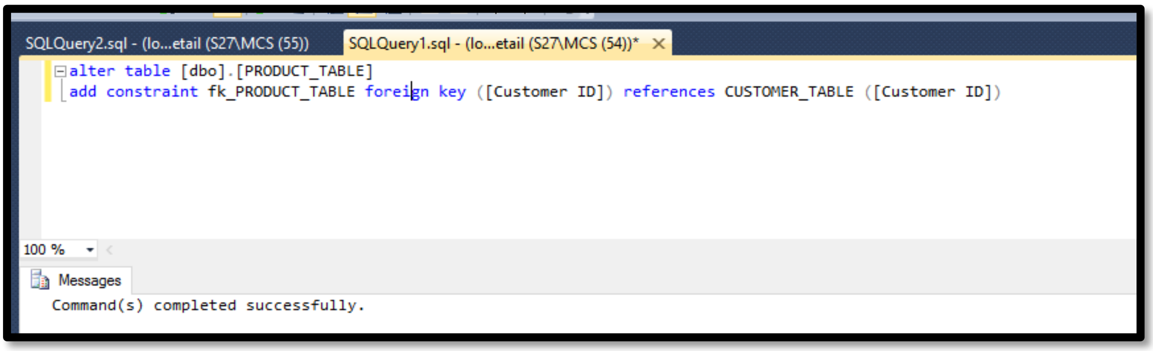
```
Command(s) completed successfully.
```



SQLQuery2.sql - (lo...etail (S27\MCS (53))*
SQLQuery1.sql - (lo...etail (S27\MCS (54))*
S27\LOCALDB#0A7....C
100 %
Messages

```sql
ALTER Table PRODUCT_TABLE
add constraint pk_PRODUCT_TABLE primary key([Product_id]);
```

```
Command(s) completed successfully.
```

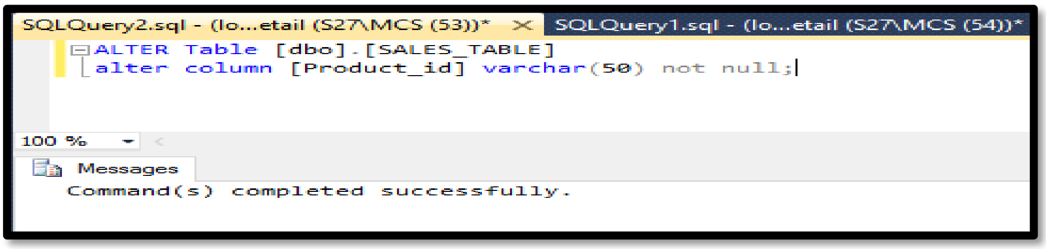


SQLQuery2.sql - (lo...etail (S27\MCS (55))
SQLQuery1.sql - (lo...etail (S27\MCS (54))*
100 %
Messages

```sql
alter table [dbo].[PRODUCT_TABLE]
add constraint fk_PRODUCT_TABLE foreign key ([Customer ID]) references CUSTOMER_TABLE ([Customer ID])
```

```
Command(s) completed successfully.
```

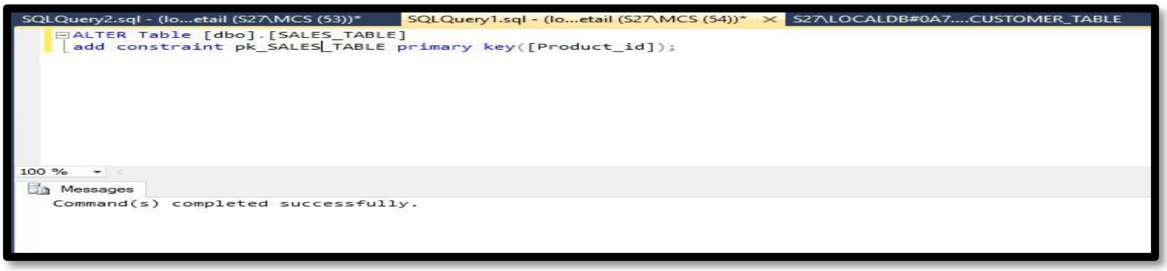
SALES TABLE:



SQLQuery2.sql - (lo...etail (S27\MCS (53))*
SQLQuery1.sql - (lo...etail (S27\MCS (54))*
100 %
Messages

```sql
ALTER Table [dbo].[SALES_TABLE]
alter column [Product_id] varchar(50) not null;
```

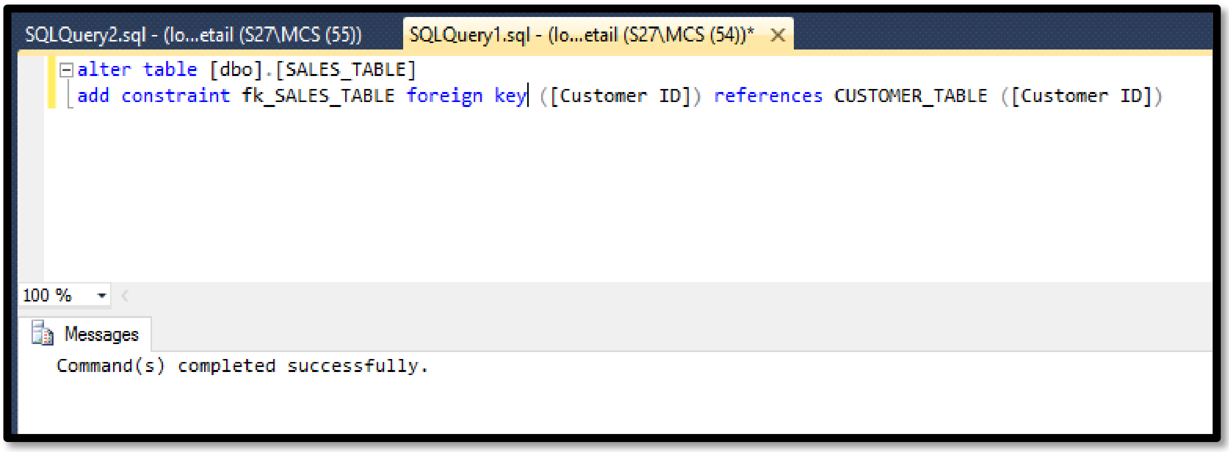
```
Command(s) completed successfully.
```



SQLQuery2.sql - (lo...etail (S27\MCS (53))*
SQLQuery1.sql - (lo...etail (S27\MCS (54))*
S27\LOCALDB#0A7....CUSTOMER_TABLE
100 %
Messages

```sql
ALTER Table [dbo].[SALES_TABLE]
add constraint pk_SALES_TABLE primary key([Product_id]);
```

```
Command(s) completed successfully.
```

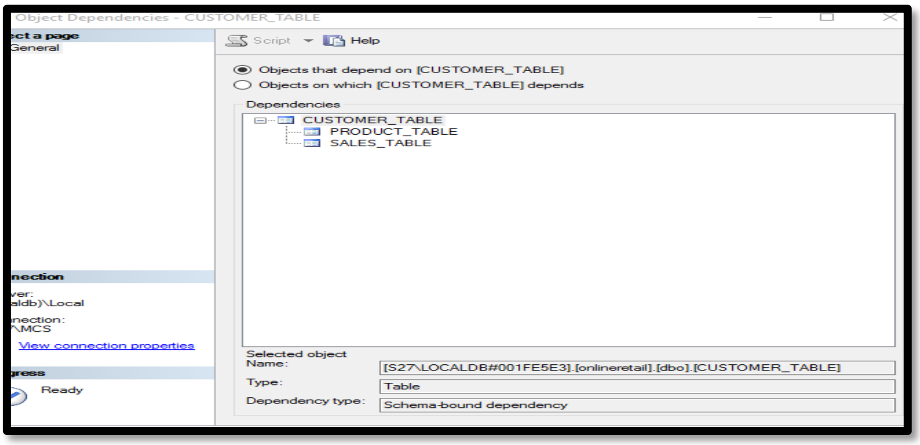


SQLQuery2.sql - (lo...etail (S27\MCS (55))
SQLQuery1.sql - (lo...etail (S27\MCS (54))*
100 %
Messages

```sql
alter table [dbo].[SALES_TABLE]
add constraint fk_SALES_TABLE foreign key ([Customer ID]) references CUSTOMER_TABLE ([Customer ID])
```

```
Command(s) completed successfully.
```

“FUNCTIONAL DEPENDENCIES AND SUPER, CANDIDATE KEYS”



● Objects that depend on [CUSTOMER_TABLE]
○ Objects on which [CUSTOMER_TABLE] depends

Dependencies

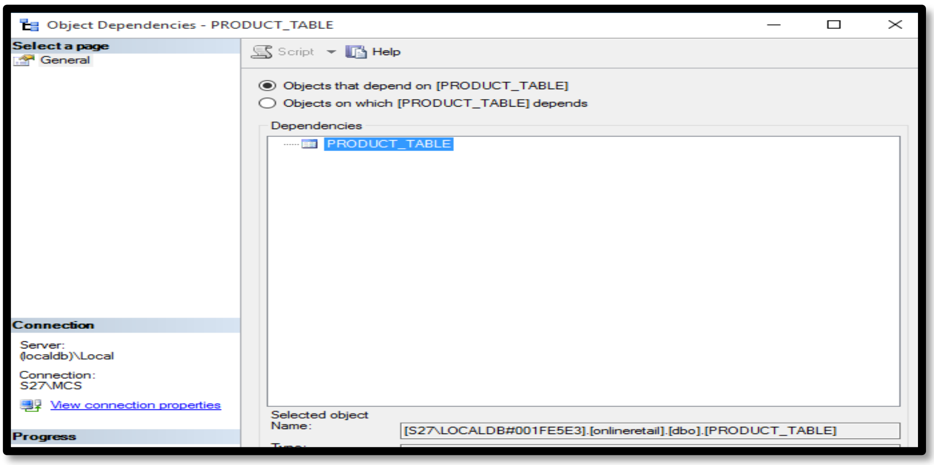
```
CUSTOMER_TABLE
    PRODUCT_TABLE
    SALES_TABLE
```

Selected object

Name: [S27\LOCALDB#001FE5E3].[onlineretail].[dbo].[CUSTOMER_TABLE]

Type: Table

Dependency type: Schema-bound dependency



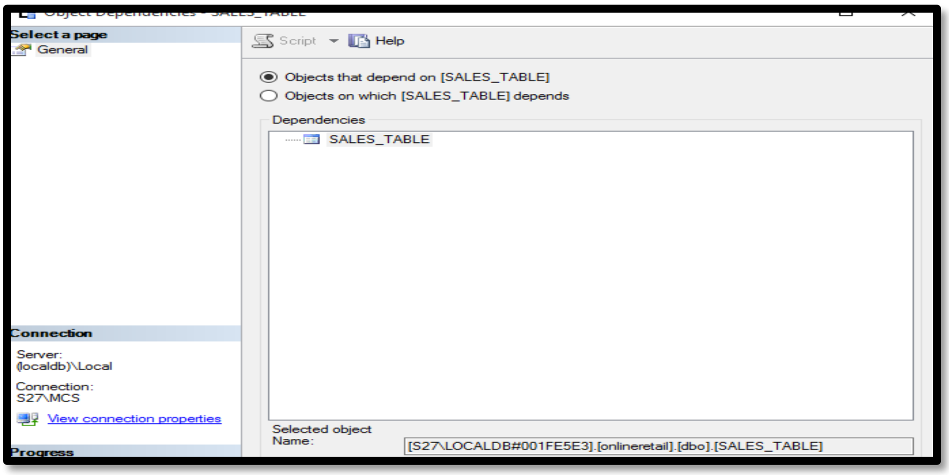
● Objects that depend on [PRODUCT_TABLE]
○ Objects on which [PRODUCT_TABLE] depends

Dependencies

```
PRODUCT_TABLE
```

Selected object

Name: [S27\LOCALDB#001FE5E3].[onlineretail].[dbo].[PRODUCT_TABLE]



○ Objects that depend on [SALES_TABLE]
○ Objects on which [SALES_TABLE] depends

Dependencies
SALES_TABLE

Server:
(localdb)\Local

Connection:
S27\MCS

View connection properties

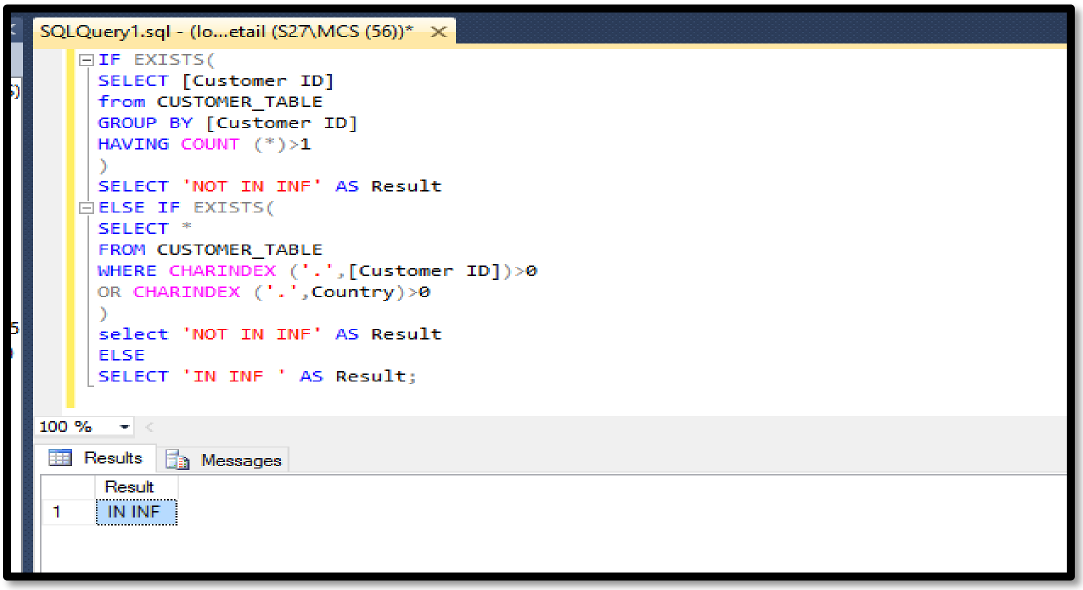
Selected object
Name: [S27\LOCALDB#001FE5E3].[onlineretail].[dbo].[SALES_TABLE]

“NORMALIZATION”

FIRST NORMAL FORM:

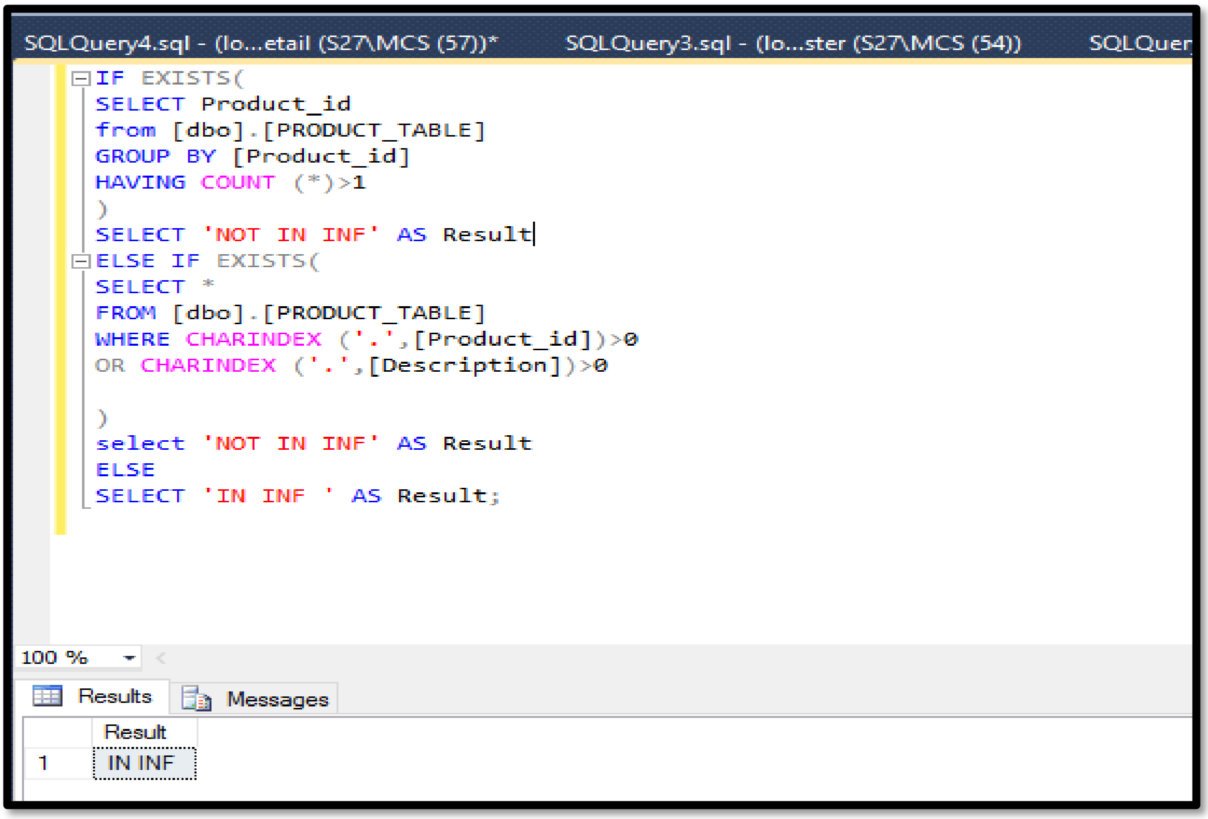
Rules for First normal form:

- Each table should have a primary key: A primary key is a unique identifier for each record. No two records should have the same primary key value.
- Data should be organized into tables: Divide your data into logical units and store related information in separate tables. Each table should represent a distinct entity or concept.
- Columns should contain atomic values: Each column should store only a single value. Avoid storing multiple values or repeating groups within a single attribute. If necessary, split them into separate columns.
- Each column should have a unique name: Ensure that each column within a table has a unique and meaningful name.



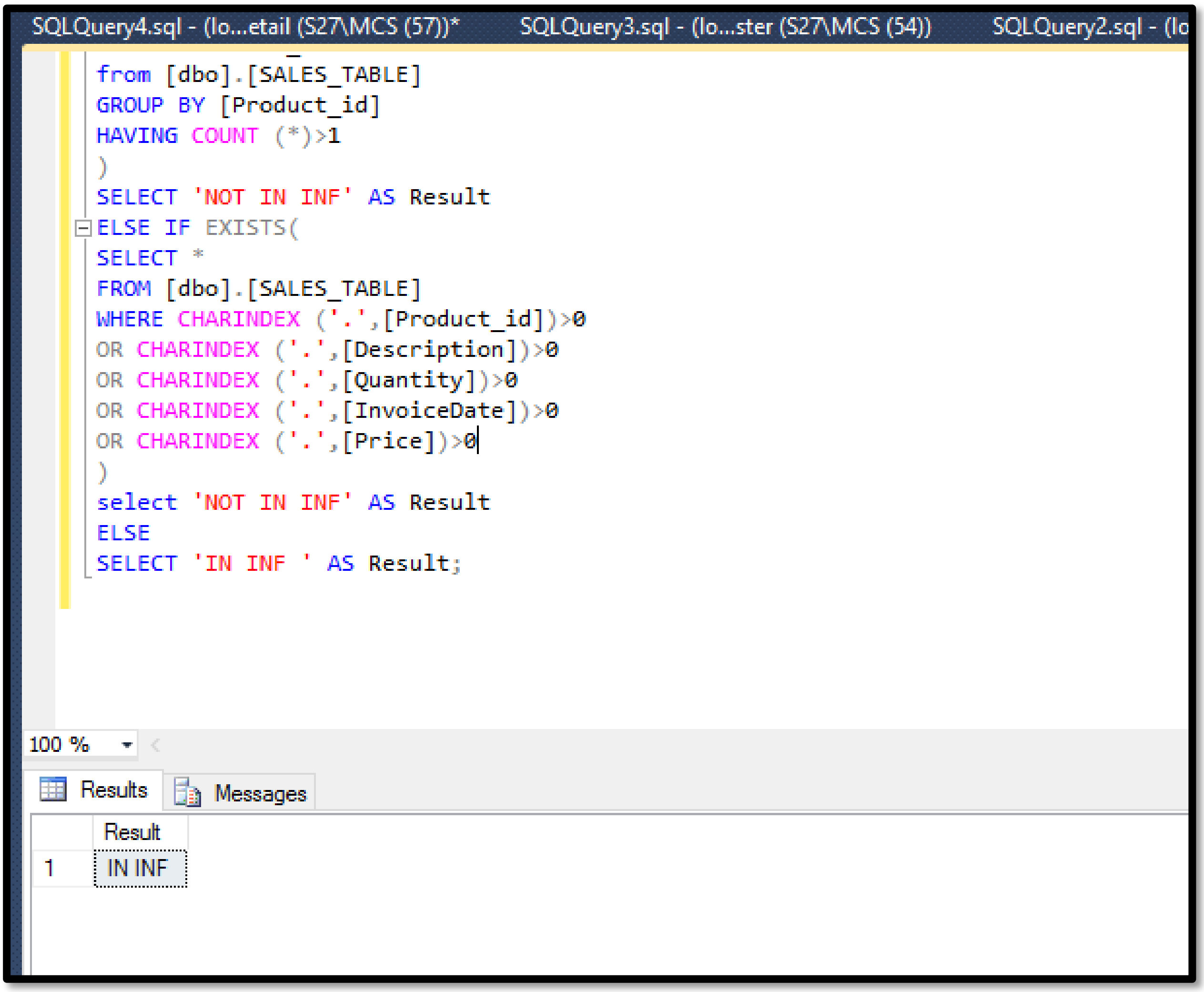
```
IF EXISTS(
SELECT [Customer ID]
from CUSTOMER_TABLE
GROUP BY [Customer ID]
HAVING COUNT (*)>1
)
SELECT 'NOT IN INF' AS Result
ELSE IF EXISTS(
SELECT *
FROM CUSTOMER_TABLE
WHERE CHARINDEX ('.',[Customer ID])>0
OR CHARINDEX ('.',Country)>0
)
select 'NOT IN INF' AS Result
ELSE
SELECT 'IN INF ' AS Result;
```

```
  Result
1 IN INF
```


SQLQuery4.sql - (lo...etail (S27\MCS (57))*
SQLQuery3.sql - (lo...ster (S27\MCS (54))
SQLQuer
100 %
Results
Messages

```
IF EXISTS(
SELECT Product_id
from [dbo].[PRODUCT_TABLE]
GROUP BY [Product_id]
HAVING COUNT (*)>1
)
SELECT 'NOT IN INF' AS Result
ELSE IF EXISTS(
SELECT *
FROM [dbo].[PRODUCT_TABLE]
WHERE CHARINDEX ('.',[Product_id])>0
OR CHARINDEX ('.',[Description])>0

)
select 'NOT IN INF' AS Result
ELSE
SELECT 'IN INF ' AS Result;
```

```
   Result
1  IN INF
```


SQLQuery4.sql - (lo...etail (S27\MCS (57))*
SQLQuery3.sql - (lo...ster (S27\MCS (54))
SQLQuery2.sql - (lo
100 %
Results
Messages

```
from [dbo].[SALES_TABLE]
GROUP BY [Product_id]
HAVING COUNT (*)>1
)
SELECT 'NOT IN INF' AS Result
ELSE IF EXISTS(
SELECT *
FROM [dbo].[SALES_TABLE]
WHERE CHARINDEX ('.',[Product_id])>0
OR CHARINDEX ('.',[Description])>0
OR CHARINDEX ('.',[Quantity])>0
OR CHARINDEX ('.',[InvoiceDate])>0
OR CHARINDEX ('.',[Price])>0
)
select 'NOT IN INF' AS Result
ELSE
SELECT 'IN INF ' AS Result;
```

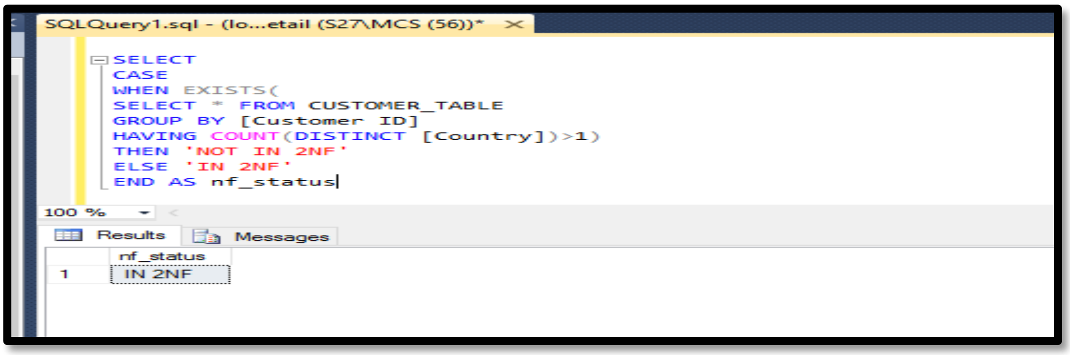
```
   Result
1  IN INF
```

…

SECOND NORMAL FORM:

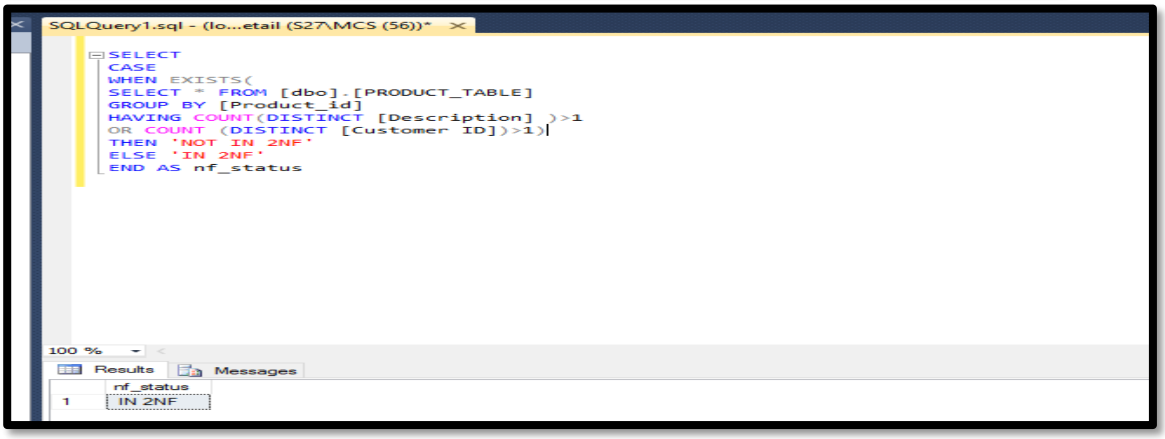
To satisfy the second normal form, a relation must meet the following criteria:

- It must already be in first normal form (1NF).
- All non-key attributes must be functionally dependent on the entire primary key, meaning there should be no partial dependencies



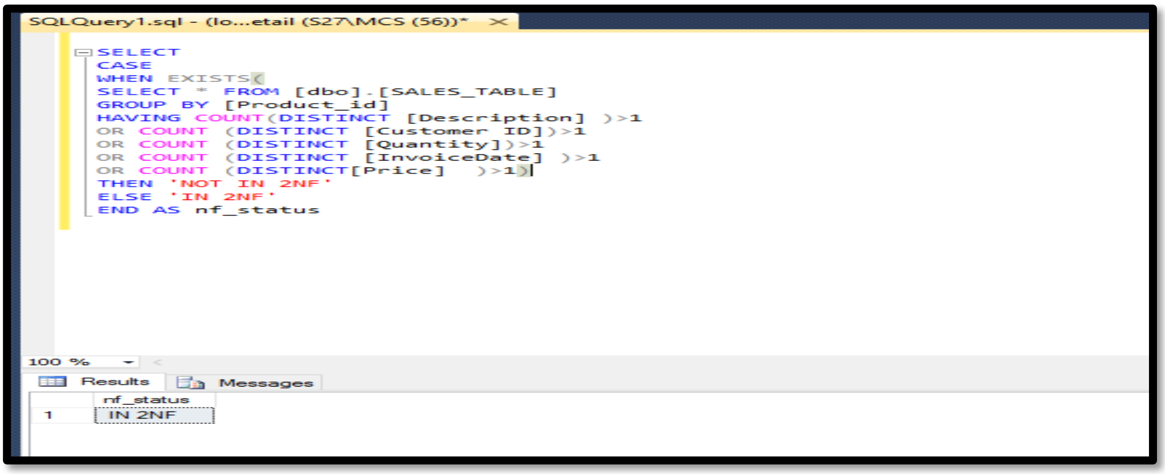
```
SELECT
CASE
WHEN EXISTS(
SELECT * FROM CUSTOMER_TABLE
GROUP BY [Customer ID]
HAVING COUNT(DISTINCT [Country])>1)
THEN 'NOT IN 2NF'
ELSE 'IN 2NF'
END AS nf_status
```

```
   nf_status
1  IN 2NF
```



```
SELECT
CASE
WHEN EXISTS(
SELECT * FROM [dbo].[PRODUCT_TABLE]
GROUP BY [Product_id]
HAVING COUNT(DISTINCT [Description] )>1
OR COUNT (DISTINCT [Customer ID])>1)
THEN 'NOT IN 2NF'
ELSE 'IN 2NF'
END AS nf_status
```

```
   nf_status
1  IN 2NF
```



```
SELECT
CASE
WHEN EXISTS(
SELECT * FROM [dbo].[SALES_TABLE]
GROUP BY [Product_id]
HAVING COUNT(DISTINCT [Description] )>1
OR COUNT (DISTINCT [Customer ID])>1
OR COUNT (DISTINCT [Quantity])>1
OR COUNT (DISTINCT [InvoiceDate] )>1
OR COUNT (DISTINCT[Price]   )>1)
THEN 'NOT IN 2NF'
ELSE 'IN 2NF'
END AS nf_status
```

```
   nf_status
1  IN 2NF
```

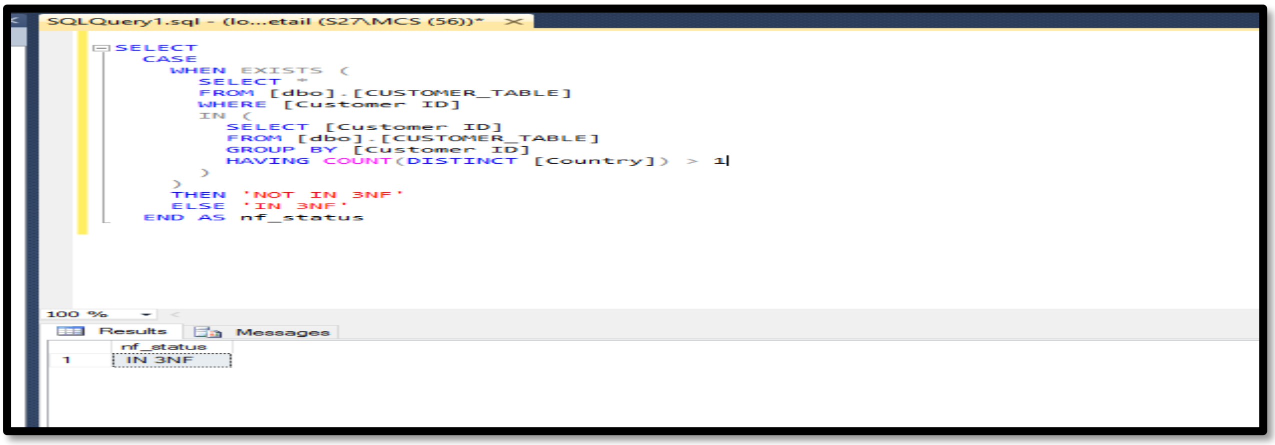
THIRD NORMAL FORM:

Rules for First normal form:

- Meet the requirements of 2NF: The table must already satisfy the rules of the Second Normal

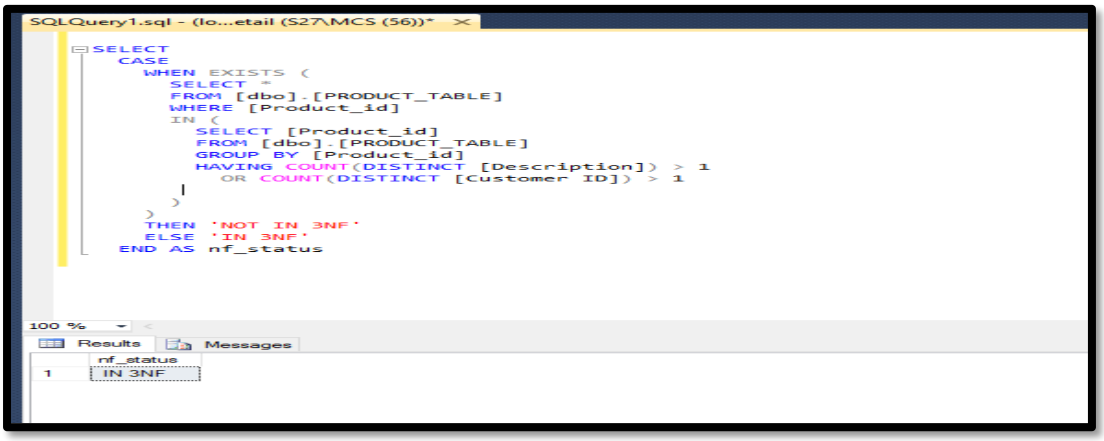
Form (2NF). This means that it should be in 2NF, and all non-key attributes should be fully functionally dependent on the primary key.

- Eliminate transitive dependencies: A transitive dependency occurs when a non-key attribute depends on another non-key attribute. To adhere to 3NF, you must remove any such dependencies by decomposing the table into multiple tables.



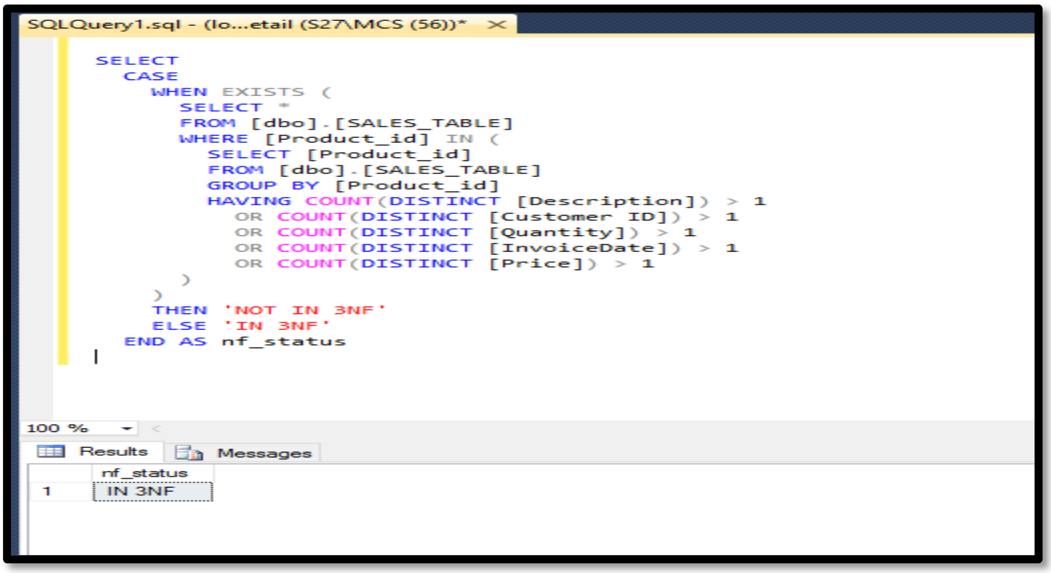
```
SELECT
    CASE
        WHEN EXISTS (
            SELECT *
            FROM [dbo].[CUSTOMER_TABLE]
            WHERE [Customer ID]
            IN (
                SELECT [Customer ID]
                FROM [dbo].[CUSTOMER_TABLE]
                GROUP BY [Customer ID]
                HAVING COUNT(DISTINCT [Country]) > 1
            )
        )
        THEN 'NOT IN 3NF'
        ELSE 'IN 3NF'
    END AS nf_status
```

```
   nf_status
1  IN 3NF
```



```
SELECT
    CASE
        WHEN EXISTS (
            SELECT *
            FROM [dbo].[PRODUCT_TABLE]
            WHERE [Product_id]
            IN (
                SELECT [Product_id]
                FROM [dbo].[PRODUCT_TABLE]
                GROUP BY [Product_id]
                HAVING COUNT(DISTINCT [Description]) > 1
                    OR COUNT(DISTINCT [Customer ID]) > 1

            )
        )
        THEN 'NOT IN 3NF'
        ELSE 'IN 3NF'
    END AS nf_status
```

```
   nf_status
1  IN 3NF
```

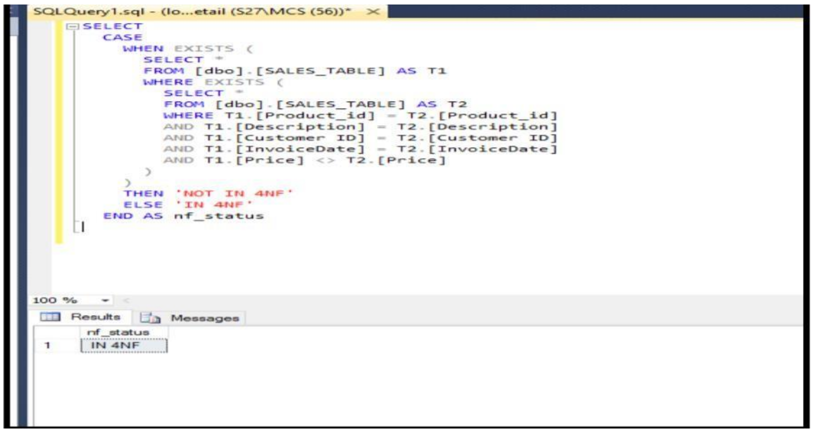


```
SELECT
    CASE
        WHEN EXISTS (
            SELECT *
            FROM [dbo].[SALES_TABLE]
            WHERE [Product_id] IN (
                SELECT [Product_id]
                FROM [dbo].[SALES_TABLE]
                GROUP BY [Product_id]
                HAVING COUNT(DISTINCT [Description]) > 1
                    OR COUNT(DISTINCT [Customer ID]) > 1
                    OR COUNT(DISTINCT [Quantity]) > 1
                    OR COUNT(DISTINCT [InvoiceDate]) > 1
                    OR COUNT(DISTINCT [Price]) > 1
            )
        )
        THEN 'NOT IN 3NF'
        ELSE 'IN 3NF'
    END AS nf_status
```

```
   nf_status
1  IN 3NF
```

FOURTH NORMAL FORM:

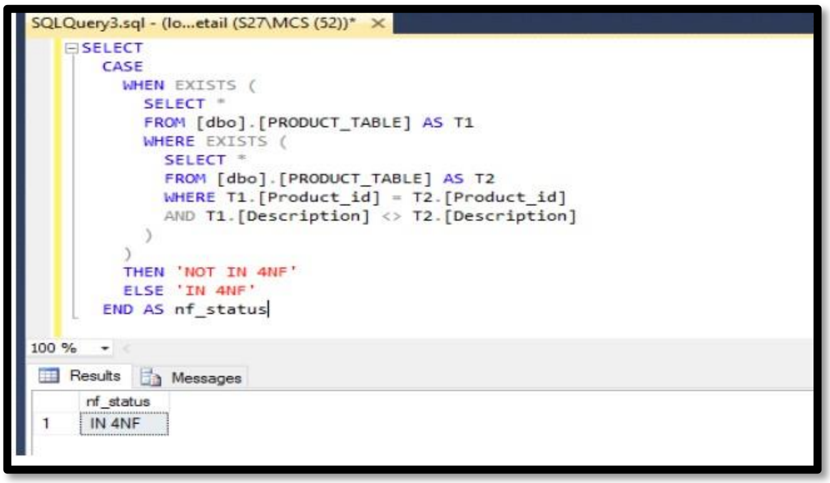
Fourth Normal Form (4NF) is an advanced level of database normalization that addresses certain types of multivalued dependencies. It builds upon the concepts of the first, second, and third normal forms (1NF, 2NF, and 3NF) and is designed to further eliminate data redundancy and anomalies in a relational database.



SQLQuery1.sql - (lo...etail (S27\MCS (56))*

```
SELECT
  CASE
    WHEN EXISTS (
      SELECT *
      FROM [dbo].[SALES_TABLE] AS T1
      WHERE EXISTS (
        SELECT *
        FROM [dbo].[SALES_TABLE] AS T2
        WHERE T1.[Product_id] = T2.[Product_id]
        AND T1.[Description] = T2.[Description]
        AND T1.[Customer ID] = T2.[Customer ID]
        AND T1.[InvoiceDate] = T2.[InvoiceDate]
        AND T1.[Price] <> T2.[Price]
      )
    )
    THEN 'NOT IN 4NF'
    ELSE 'IN 4NF'
  END AS nf_status
```

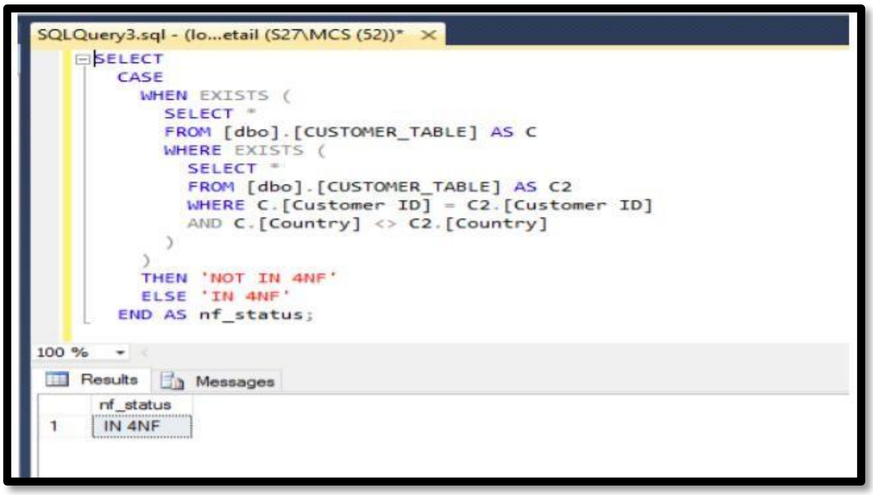
	nf_status
1	IN 4NF



SQLQuery3.sql - (lo...etail (S27\MCS (52))*

```
SELECT
  CASE
    WHEN EXISTS (
      SELECT *
      FROM [dbo].[PRODUCT_TABLE] AS T1
      WHERE EXISTS (
        SELECT *
        FROM [dbo].[PRODUCT_TABLE] AS T2
        WHERE T1.[Product_id] = T2.[Product_id]
        AND T1.[Description] <> T2.[Description]
      )
    )
    THEN 'NOT IN 4NF'
    ELSE 'IN 4NF'
  END AS nf_status
```

	nf_status
1	IN 4NF



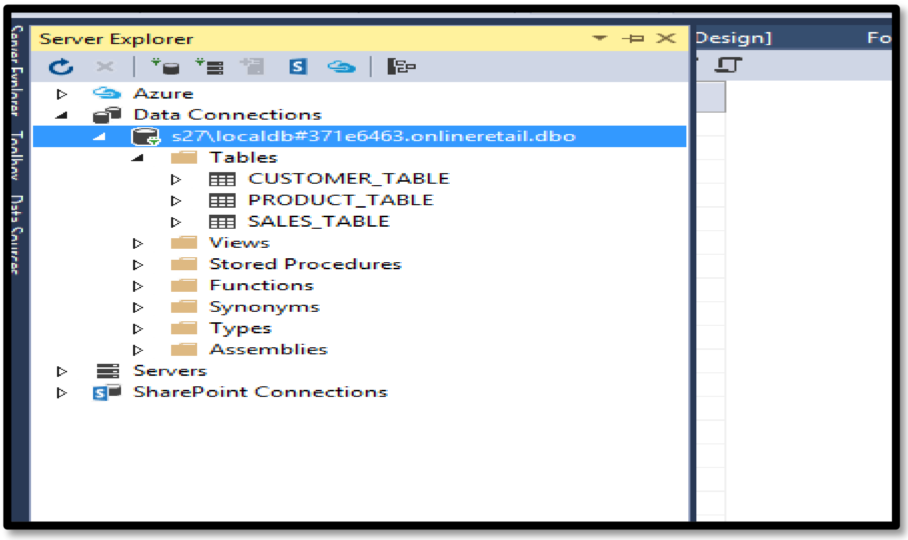
SQLQuery3.sql - (lo...etail (S27\MCS (52))*

```
SELECT
  CASE
    WHEN EXISTS (
      SELECT *
      FROM [dbo].[CUSTOMER_TABLE] AS C
      WHERE EXISTS (
        SELECT *
        FROM [dbo].[CUSTOMER_TABLE] AS C2
        WHERE C.[Customer ID] = C2.[Customer ID]
        AND C.[Country] <> C2.[Country]
      )
    )
    THEN 'NOT IN 4NF'
    ELSE 'IN 4NF'
  END AS nf_status;
```

	nf_status
1	IN 4NF

“FRONT END IN VISUAL STUDIO”

CONNECTIVITY WITH DATABASE:

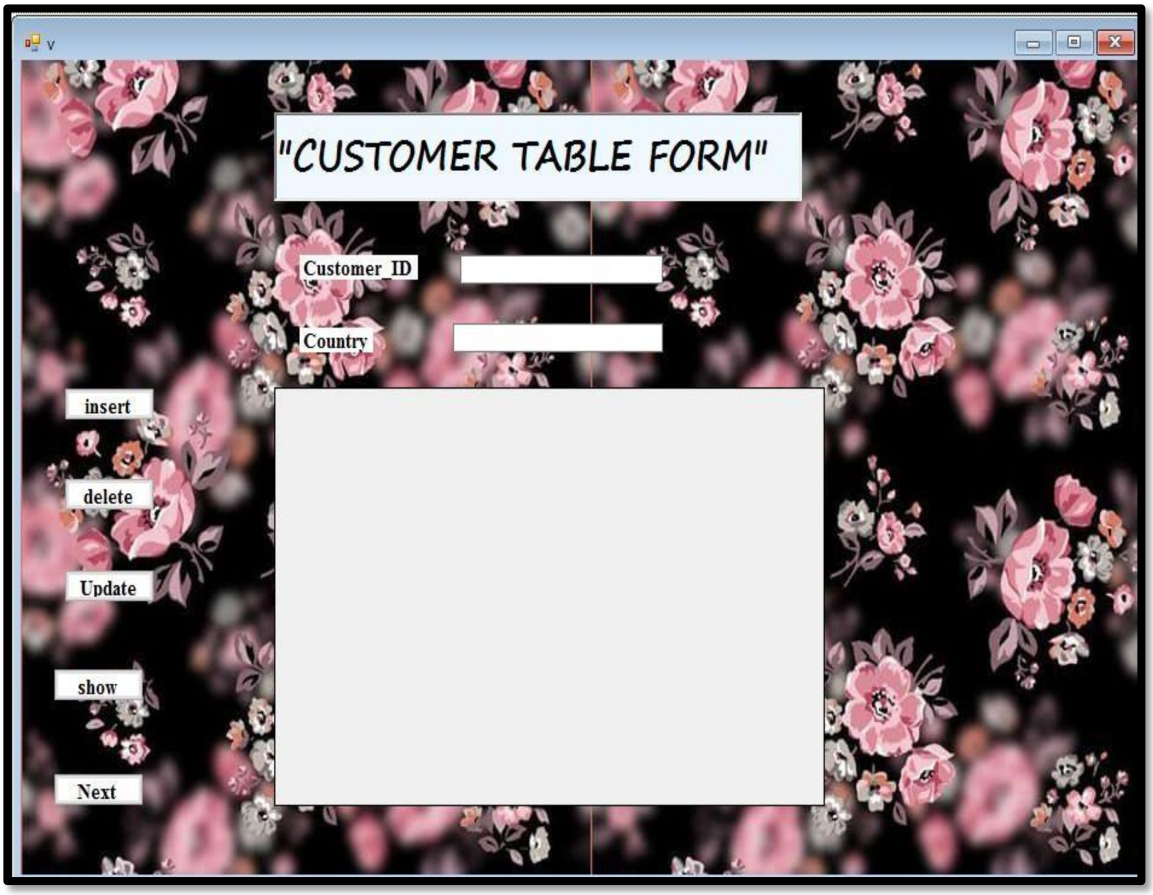


Server Explorer

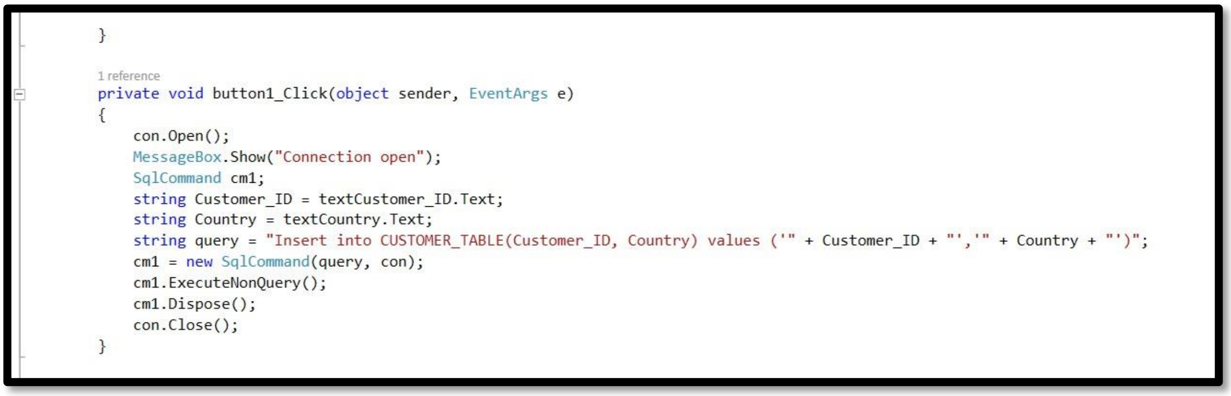
- Azure
- Data Connections
  - s27\localdb#371e6463.onlineretail.dbo
    - Tables
      - CUSTOMER_TABLE
      - PRODUCT_TABLE
      - SALES_TABLE
    - Views
    - Stored Procedures
    - Functions
    - Synonyms
    - Types
    - Assemblies
- Servers
- SharePoint Connections

“CUSTOMER TABLE FORM”

FORM:

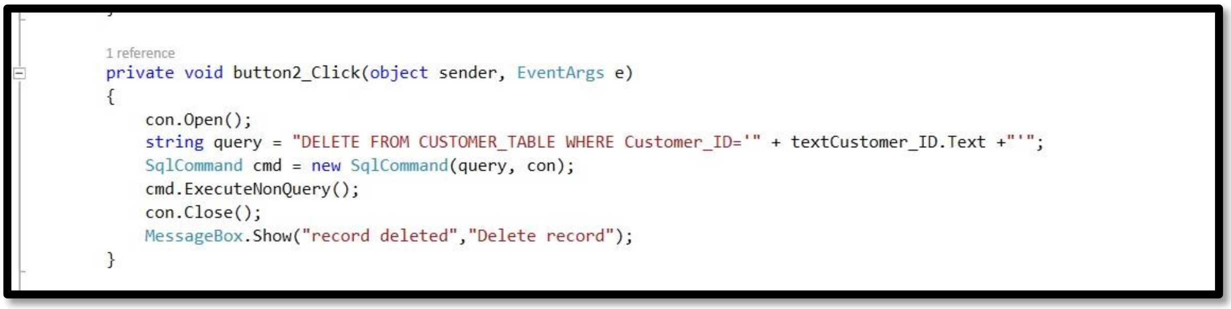


INSERT:



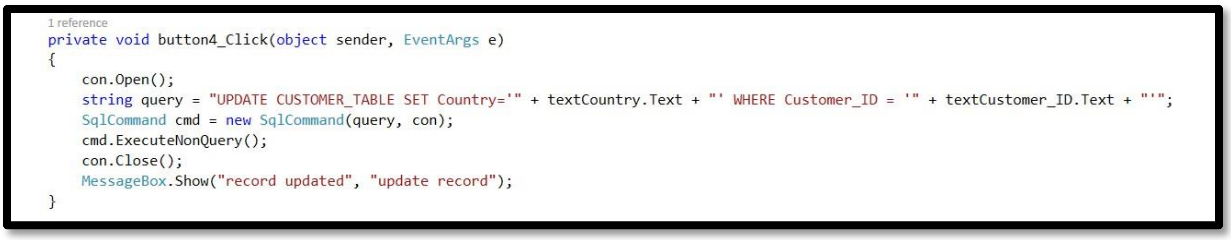
```
        }

        1 reference
        private void button1_Click(object sender, EventArgs e)
        {
            con.Open();
            MessageBox.Show("Connection open");
            SqlCommand cm1;
            string Customer_ID = textCustomer_ID.Text;
            string Country = textCountry.Text;
            string query = "Insert into CUSTOMER_TABLE(Customer_ID, Country) values ('" + Customer_ID + "','" + Country + "')";
            cm1 = new SqlCommand(query, con);
            cm1.ExecuteNonQuery();
            cm1.Dispose();
            con.Close();
        }
```

DELETE:



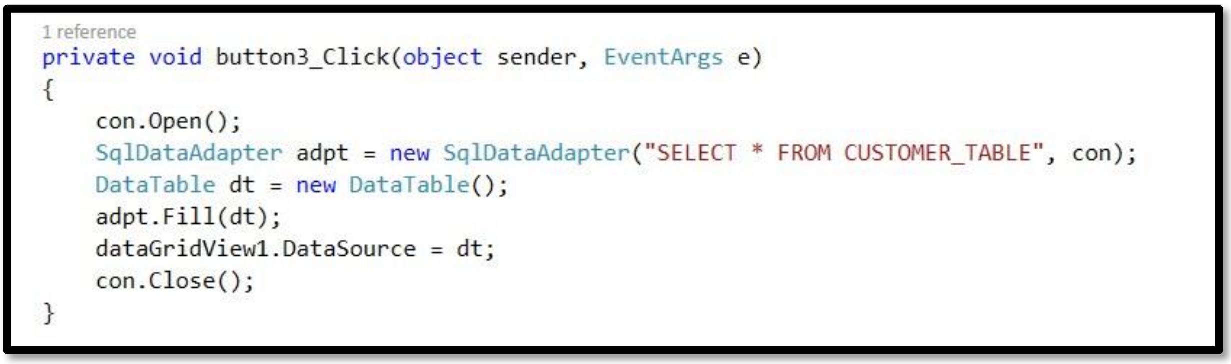
```
        1 reference
        private void button2_Click(object sender, EventArgs e)
        {
            con.Open();
            string query = "DELETE FROM CUSTOMER_TABLE WHERE Customer_ID='" + textCustomer_ID.Text +"'";
            SqlCommand cmd = new SqlCommand(query, con);
            cmd.ExecuteNonQuery();
            con.Close();
            MessageBox.Show("record deleted","Delete record");
        }
```

UPDATE:



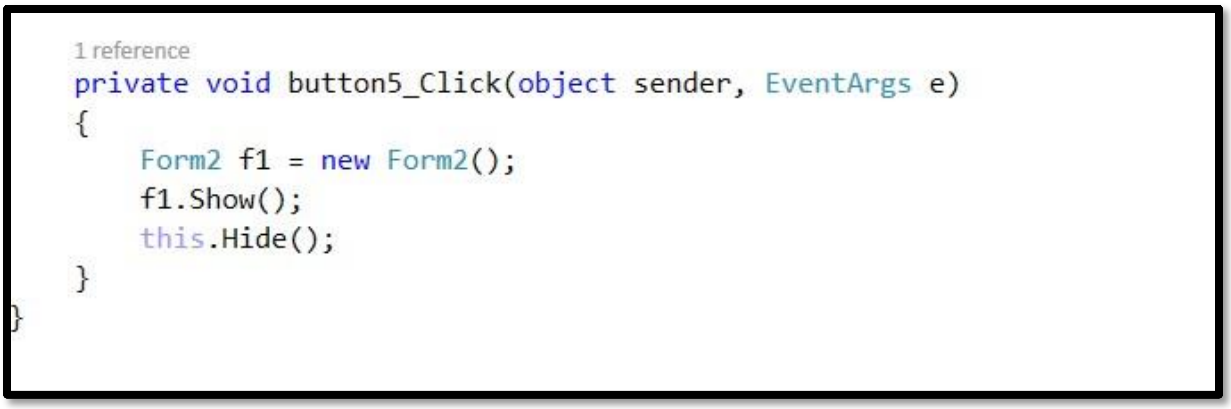
```
1 reference
private void button4_Click(object sender, EventArgs e)
{
    con.Open();
    string query = "UPDATE CUSTOMER_TABLE SET Country='" + textCountry.Text + "' WHERE Customer_ID = '" + textCustomer_ID.Text + "'";
    SqlCommand cmd = new SqlCommand(query, con);
    cmd.ExecuteNonQuery();
    con.Close();
    MessageBox.Show("record updated", "update record");
}
```

SHOW:



```
1 reference
private void button3_Click(object sender, EventArgs e)
{
    con.Open();
    SqlDataAdapter adpt = new SqlDataAdapter("SELECT * FROM CUSTOMER_TABLE", con);
    DataTable dt = new DataTable();
    adpt.Fill(dt);
    dataGridView1.DataSource = dt;
    con.Close();
}
```

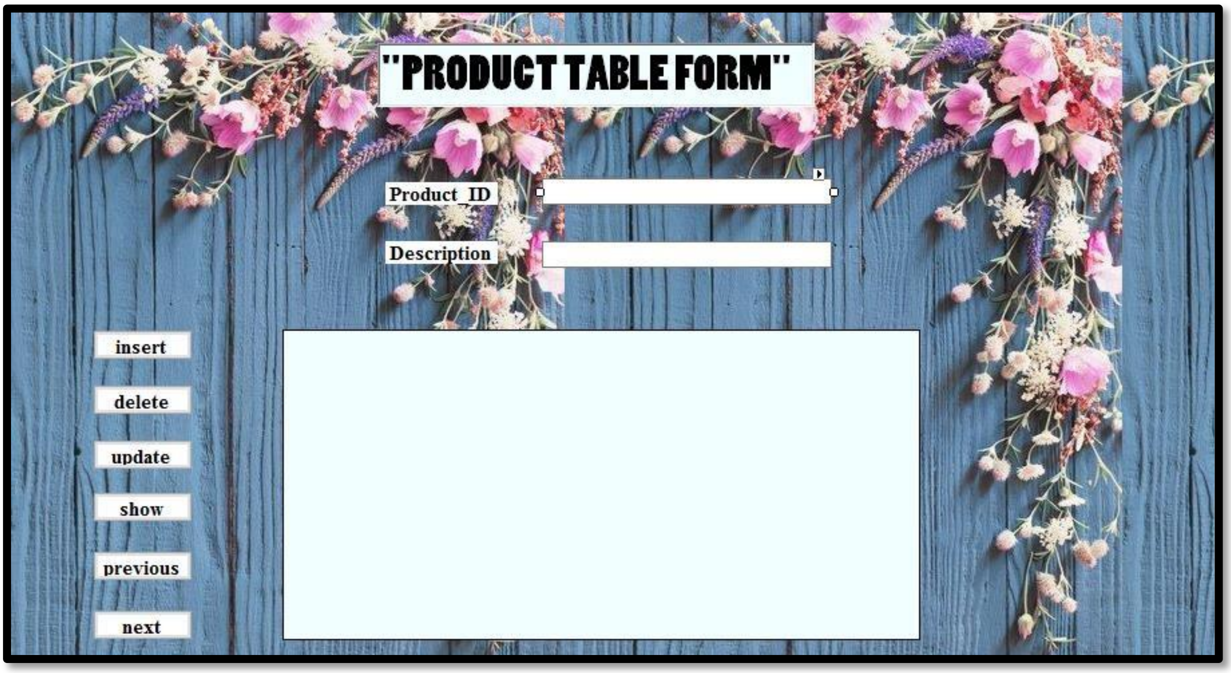
NEXT:



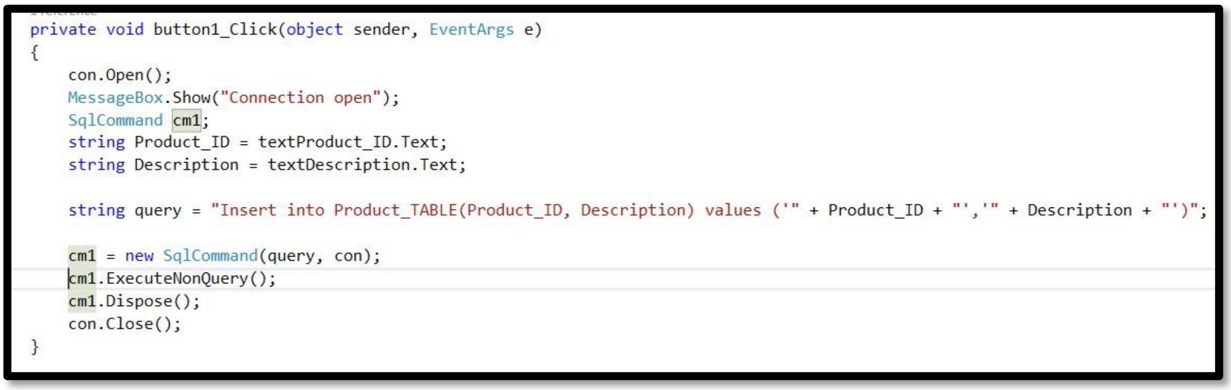
```
1 reference
private void button5_Click(object sender, EventArgs e)
{
    Form2 f1 = new Form2();
    f1.Show();
    this.Hide();
}
}
```

“PRODUCT TABLE FORM”

FORM:

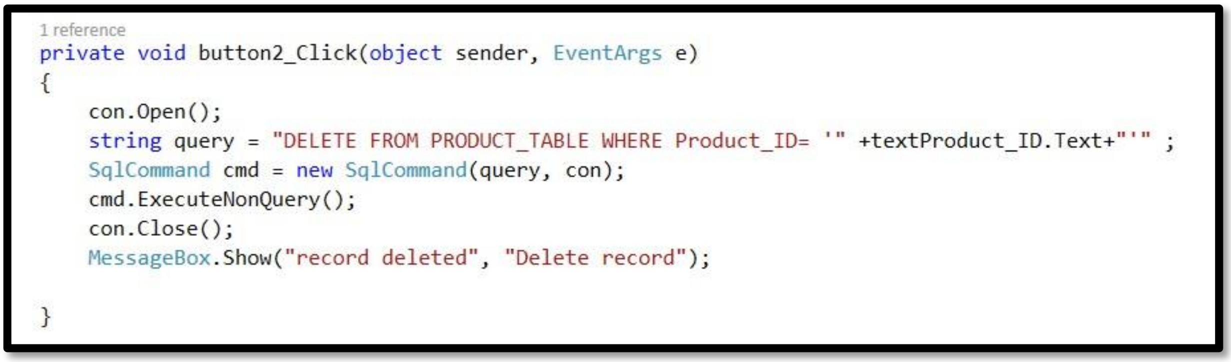


INSERT:



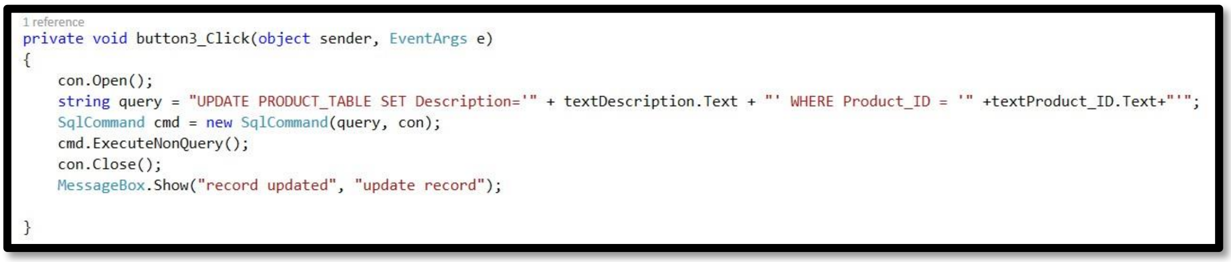
```
private void button1_Click(object sender, EventArgs e)
{
    con.Open();
    MessageBox.Show("Connection open");
    SqlCommand cm1;
    string Product_ID = textProduct_ID.Text;
    string Description = textDescription.Text;

    string query = "Insert into Product_TABLE(Product_ID, Description) values ('" + Product_ID + "','" + Description + "')";

    cm1 = new SqlCommand(query, con);
    cm1.ExecuteNonQuery();
    cm1.Dispose();
    con.Close();
}
```

DELETE:



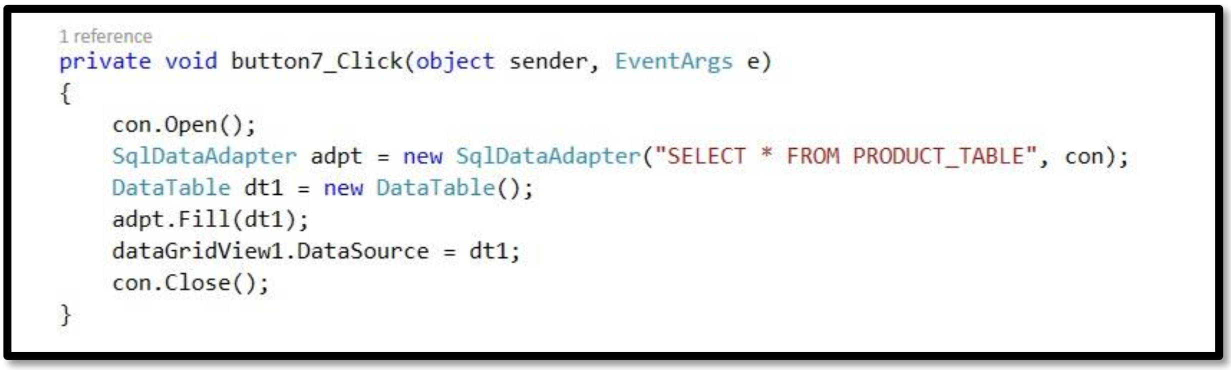
```
1 reference
private void button2_Click(object sender, EventArgs e)
{
    con.Open();
    string query = "DELETE FROM PRODUCT_TABLE WHERE Product_ID= '" +textProduct_ID.Text+"'" ;
    SqlCommand cmd = new SqlCommand(query, con);
    cmd.ExecuteNonQuery();
    con.Close();
    MessageBox.Show("record deleted", "Delete record");

}
```

UPDATE:



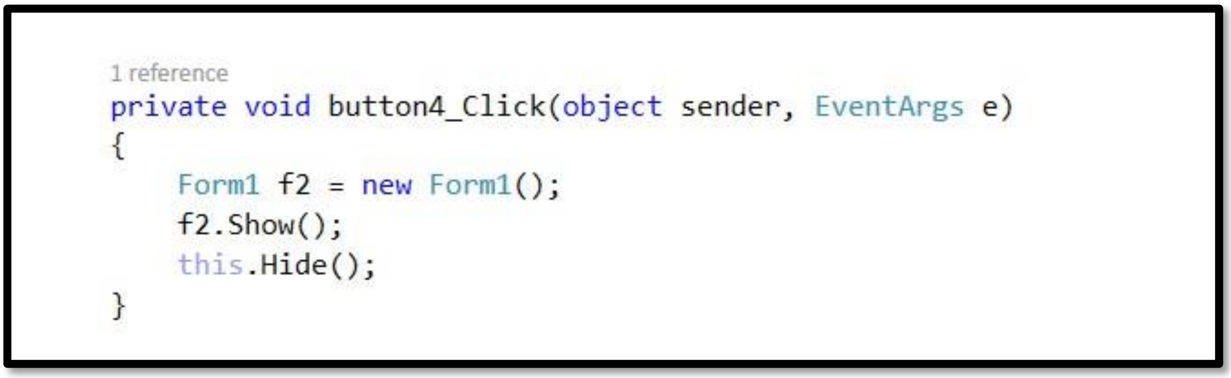
```
1 reference
private void button3_Click(object sender, EventArgs e)
{
    con.Open();
    string query = "UPDATE PRODUCT_TABLE SET Description='" + textDescription.Text + "' WHERE Product_ID = '" +textProduct_ID.Text+"'";
    SqlCommand cmd = new SqlCommand(query, con);
    cmd.ExecuteNonQuery();
    con.Close();
    MessageBox.Show("record updated", "update record");

}
```

SHOW:



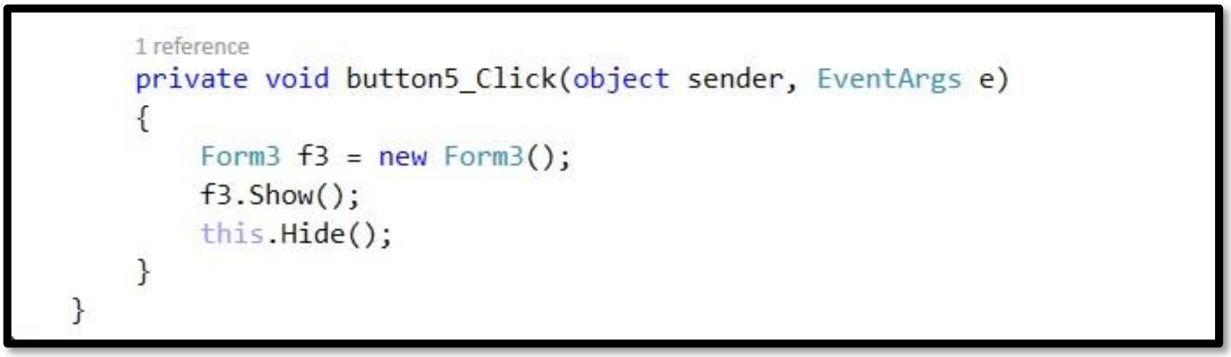
```
1 reference
private void button7_Click(object sender, EventArgs e)
{
    con.Open();
    SqlDataAdapter adpt = new SqlDataAdapter("SELECT * FROM PRODUCT_TABLE", con);
    DataTable dt1 = new DataTable();
    adpt.Fill(dt1);
    dataGridView1.DataSource = dt1;
    con.Close();
}
```

PREVIOUS:



```
1 reference
private void button4_Click(object sender, EventArgs e)
{
    Form1 f2 = new Form1();
    f2.Show();
    this.Hide();
}
```

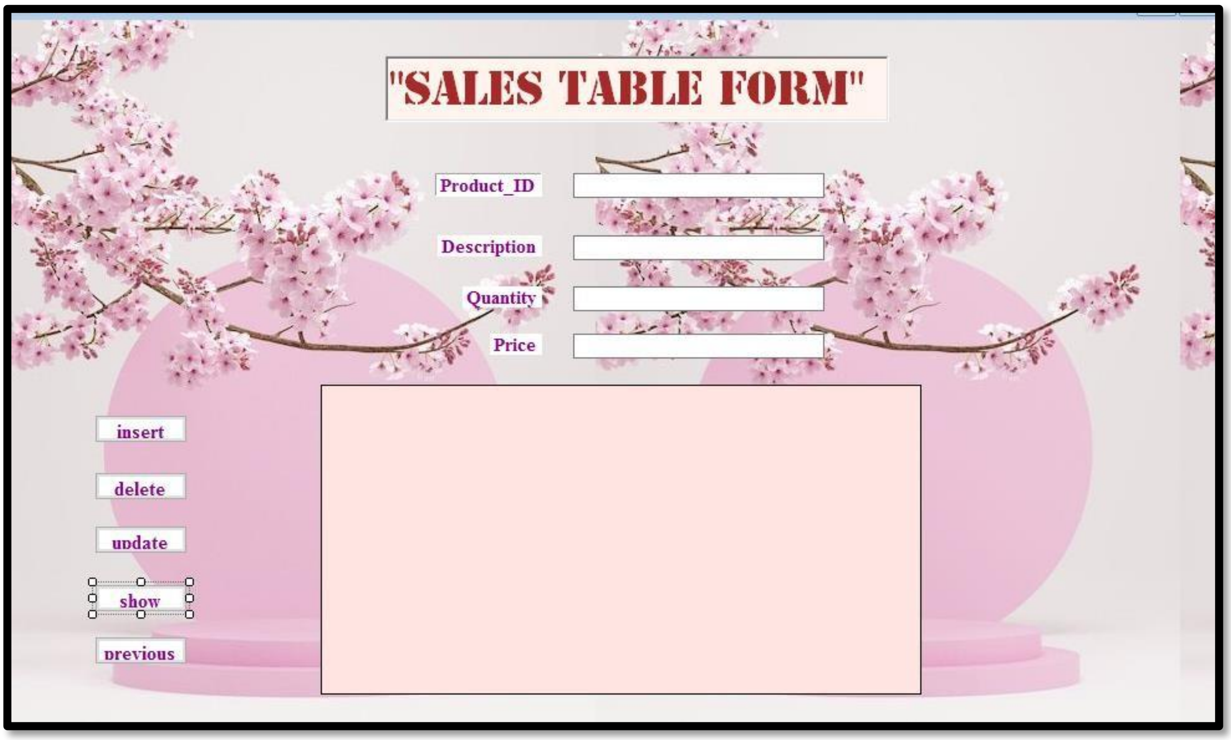
NEXT:



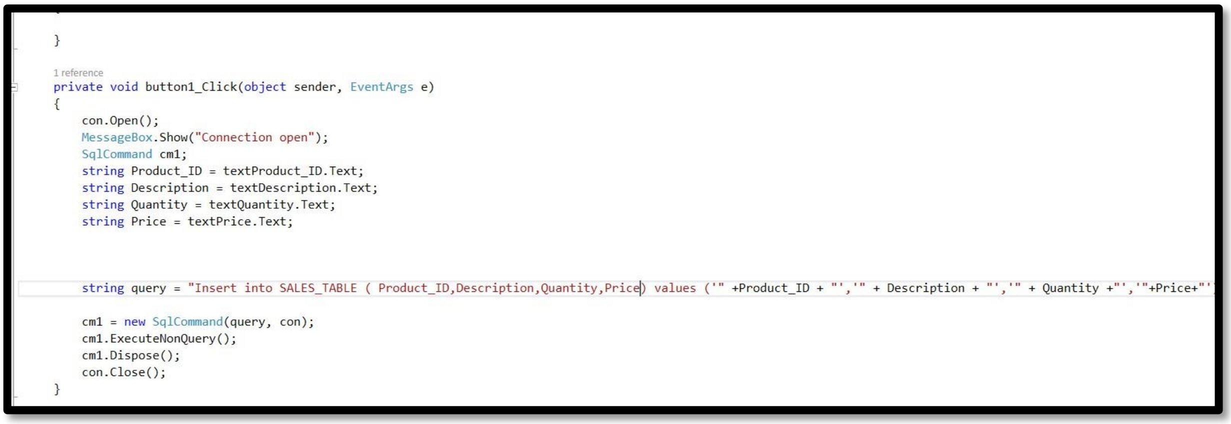
```
        1 reference
        private void button5_Click(object sender, EventArgs e)
        {
            Form3 f3 = new Form3();
            f3.Show();
            this.Hide();
        }
    }
```

“SALES TABLE FORM”

FORM:

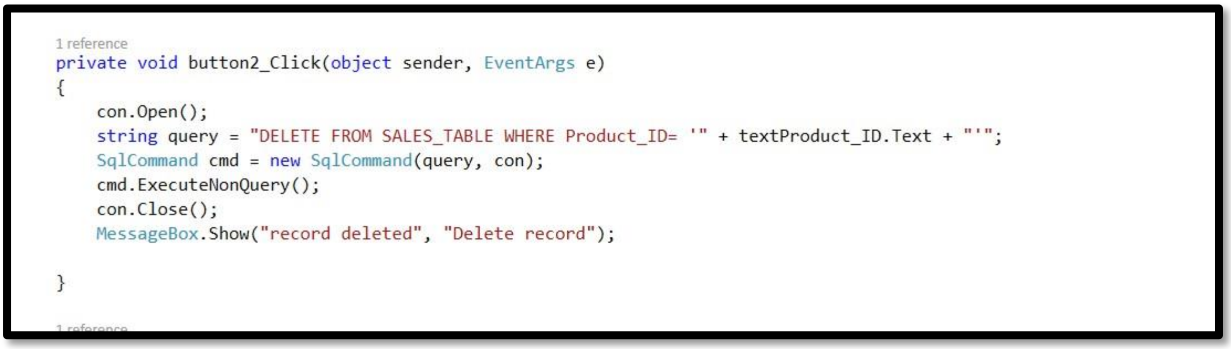


INSERT:



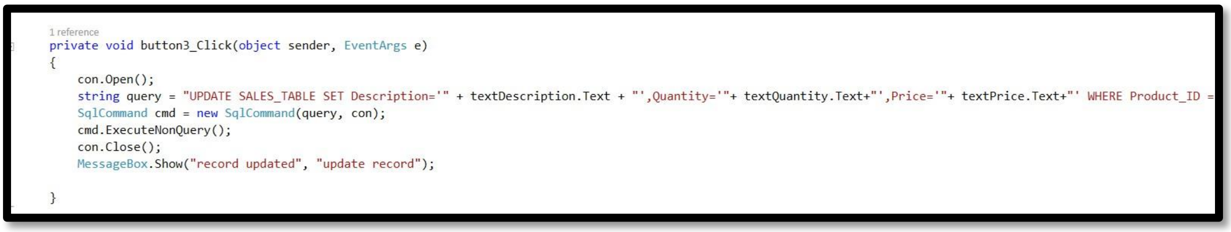
```
        }

        1 reference
        private void button1_Click(object sender, EventArgs e)
        {
            con.Open();
            MessageBox.Show("Connection open");
            SqlCommand cm1;
            string Product_ID = textProduct_ID.Text;
            string Description = textDescription.Text;
            string Quantity = textQuantity.Text;
            string Price = textPrice.Text;



            string query = "Insert into SALES_TABLE ( Product_ID,Description,Quantity,Price) values ('" +Product_ID + "','" + Description + "','" + Quantity +"','"+Price+"'

            cm1 = new SqlCommand(query, con);
            cm1.ExecuteNonQuery();
            cm1.Dispose();
            con.Close();
        }
```

DELETE:



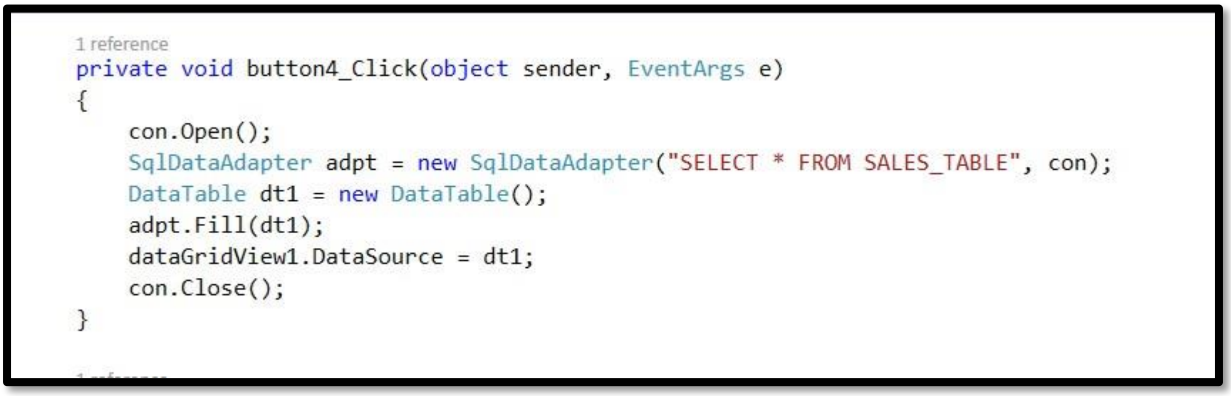
```
1 reference
private void button2_Click(object sender, EventArgs e)
{
    con.Open();
    string query = "DELETE FROM SALES_TABLE WHERE Product_ID= '" + textProduct_ID.Text + "'";
    SqlCommand cmd = new SqlCommand(query, con);
    cmd.ExecuteNonQuery();
    con.Close();
    MessageBox.Show("record deleted", "Delete record");

}

1 reference
```

UPDATE:



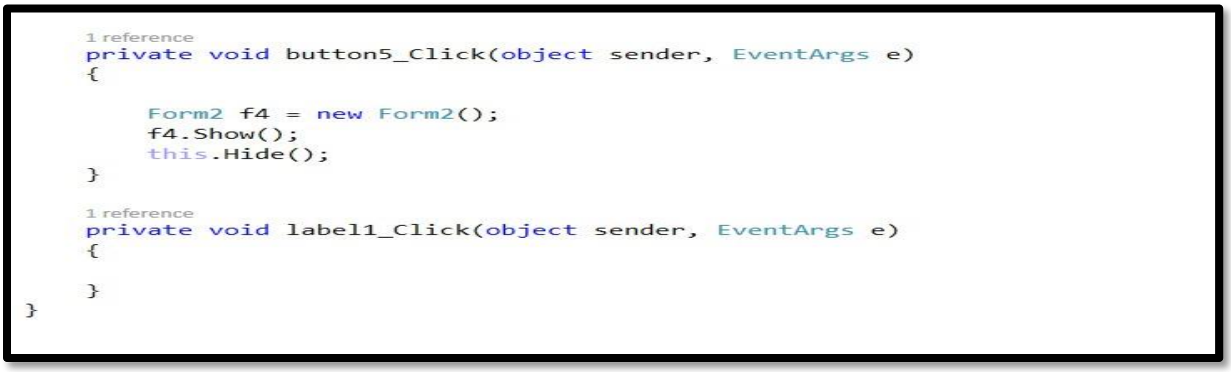
```
1 reference
private void button3_Click(object sender, EventArgs e)
{
    con.Open();
    string query = "UPDATE SALES_TABLE SET Description='" + textDescription.Text + "',Quantity='"+ textQuantity.Text+"',Price='"+ textPrice.Text+"' WHERE Product_ID =
    SqlCommand cmd = new SqlCommand(query, con);
    cmd.ExecuteNonQuery();
    con.Close();
    MessageBox.Show("record updated", "update record");

}
```

SHOW:



```
1 reference
private void button4_Click(object sender, EventArgs e)
{
    con.Open();
    SqlDataAdapter adpt = new SqlDataAdapter("SELECT * FROM SALES_TABLE", con);
    DataTable dt1 = new DataTable();
    adpt.Fill(dt1);
    dataGridView1.DataSource = dt1;
    con.Close();
}

1 reference
```

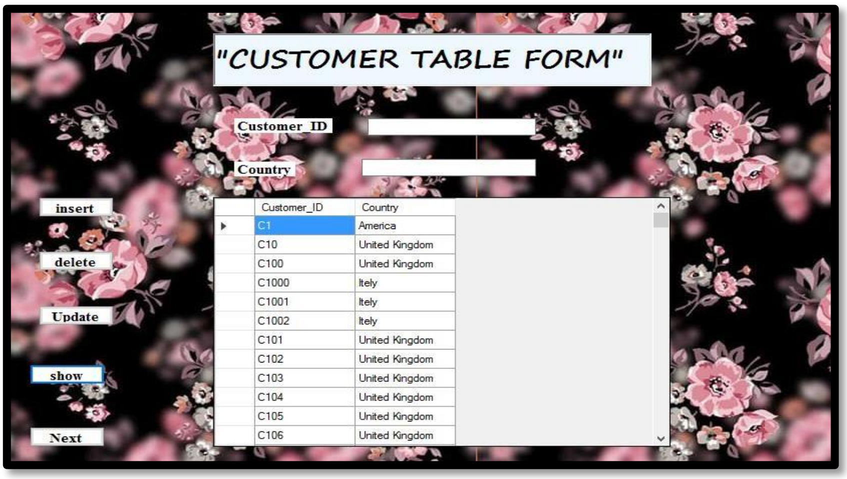
PREVIOUS:



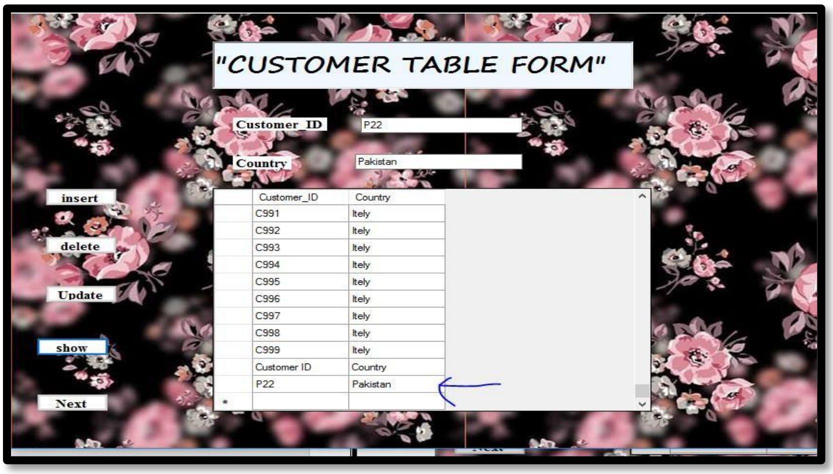
```
        1 reference
        private void button5_Click(object sender, EventArgs e)
        {

            Form2 f4 = new Form2();
            f4.Show();
            this.Hide();
        }

        1 reference
        private void label1_Click(object sender, EventArgs e)
        {

        }
    }
```

“WORKING”

INSERT:



Customer_ID	Country
C1	America
C10	United Kingdom
C100	United Kingdom
C1000	Itely
C1001	Itely
C1002	Itely
C101	United Kingdom
C102	United Kingdom
C103	United Kingdom
C104	United Kingdom
C105	United Kingdom
C106	United Kingdom

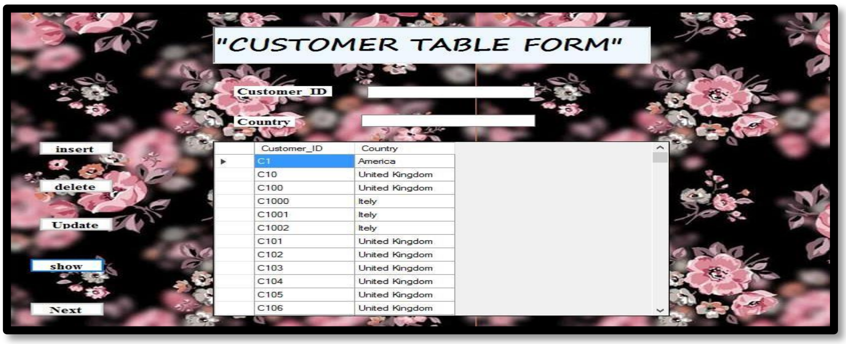


Customer_ID: P22

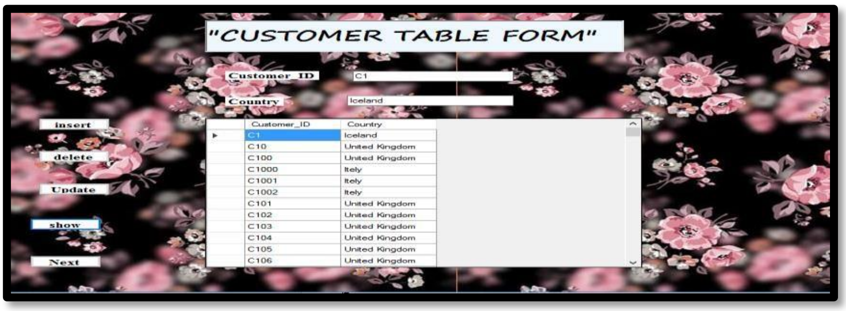
Country: Pakistan

Customer_ID	Country
C991	Itely
C992	Itely
C993	Itely
C994	Itely
C995	Itely
C996	Itely
C997	Itely
C998	Itely
C999	Itely
Customer ID	Country
P22	Pakistan

UPDATE:



Customer_ID	Country
C1	America
C10	United Kingdom
C100	United Kingdom
C1000	Itely
C1001	Itely
C1002	Itely
C101	United Kingdom
C102	United Kingdom
C103	United Kingdom
C104	United Kingdom
C105	United Kingdom
C106	United Kingdom

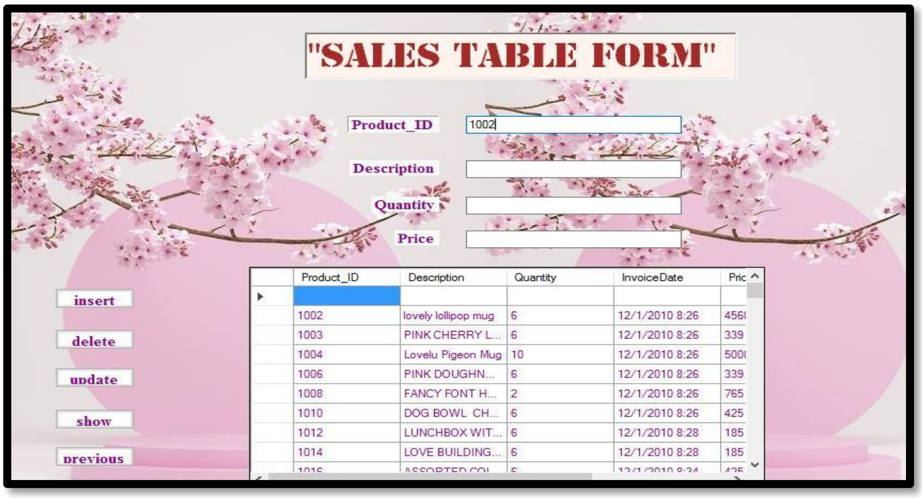


Customer_ID: C1

Country: Iceland

Customer_ID	Country
C1	Iceland
C10	United Kingdom
C100	United Kingdom
C1000	Itely
C1001	Itely
C1002	Itely
C101	United Kingdom
C102	United Kingdom
C103	United Kingdom
C104	United Kingdom
C105	United Kingdom
C106	United Kingdom

DELETE:



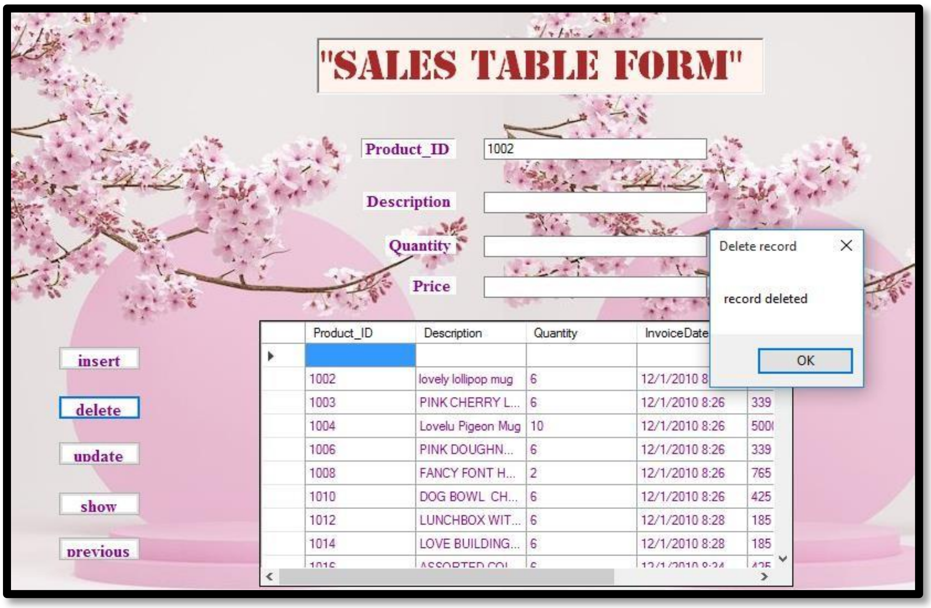
Product_ID 1002

Description

Quantity

Price

	Product_ID	Description	Quantity	InvoiceDate	Pric
▶					
	1002	lovely lollipop mug	6	12/1/2010 8:26	456
	1003	PINK CHERRY L...	6	12/1/2010 8:26	339
	1004	Lovelu Pigeon Mug	10	12/1/2010 8:26	500
	1006	PINK DOUGHN...	6	12/1/2010 8:26	339
	1008	FANCY FONT H...	2	12/1/2010 8:26	765
	1010	DOG BOWL CH...	6	12/1/2010 8:26	425
	1012	LUNCHBOX WIT...	6	12/1/2010 8:28	185
	1014	LOVE BUILDING...	6	12/1/2010 8:28	185
	1016	ASSORTED COL	6	12/1/2010 8:34	425



Product_ID 1002

Description

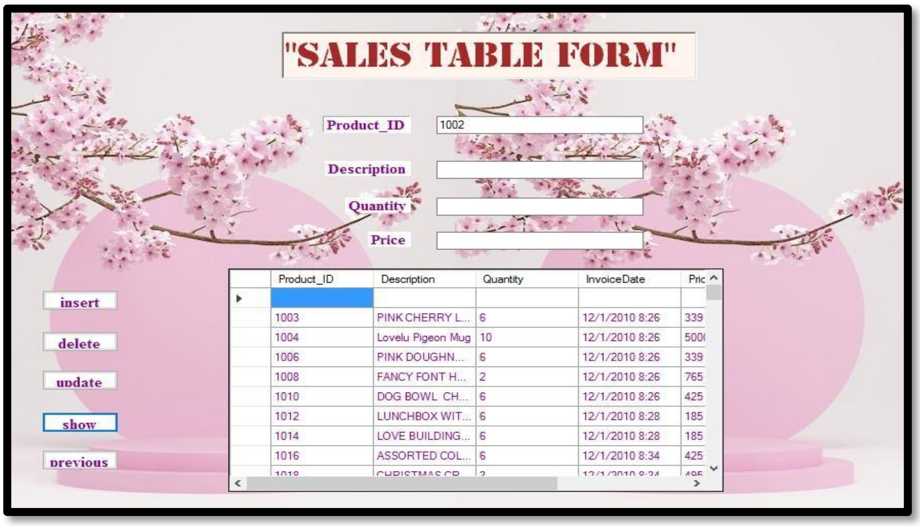
Quantity

Price

Delete record

record deleted

	Product_ID	Description	Quantity	InvoiceDate	
▶					
	1002	lovely lollipop mug	6	12/1/2010 8	
	1003	PINK CHERRY L...	6	12/1/2010 8:26	339
	1004	Lovelu Pigeon Mug	10	12/1/2010 8:26	500
	1006	PINK DOUGHN...	6	12/1/2010 8:26	339
	1008	FANCY FONT H...	2	12/1/2010 8:26	765
	1010	DOG BOWL CH...	6	12/1/2010 8:26	425
	1012	LUNCHBOX WIT...	6	12/1/2010 8:28	185
	1014	LOVE BUILDING...	6	12/1/2010 8:28	185
	1016	ASSORTED COL	6	12/1/2010 8:34	425



Product_ID 1002

Description

Quantity

Price

	Product_ID	Description	Quantity	InvoiceDate	Pric
▶					
	1003	PINK CHERRY L...	6	12/1/2010 8:26	339
	1004	Lovelu Pigeon Mug	10	12/1/2010 8:26	500
	1006	PINK DOUGHN...	6	12/1/2010 8:26	339
	1008	FANCY FONT H...	2	12/1/2010 8:26	765
	1010	DOG BOWL CH...	6	12/1/2010 8:26	425
	1012	LUNCHBOX WIT...	6	12/1/2010 8:28	185
	1014	LOVE BUILDING...	6	12/1/2010 8:28	185
	1016	ASSORTED COL...	6	12/1/2010 8:34	425
	1018	CHRISTMAS CR	3	12/1/2010 8:34	495