

### Here are more information :

```
SQL.ExecQuery("INSERT INTO
Sales.OrderLines(OrderNo,ProductID,ProductName,Quantity,UnitPrice,ExtendedPrice,DiscountRate,DiscountAmount,ValidFrom,ValidTo,InvoicedQuantity,BackOrderYesNo, " &
```

```
"BackOrderQuantity,LineTotal,DeliveryDate,TaxRate1,TaxRate2,TaxRate3,TaxAmount1,TaxAmount2,TaxAmount3,OrderPaid,CommissionPaid,ManageInventory) " &
"VALUES
(@OrderNo,@ProductID,@ProductName,@Quantity,@UnitPrice,@ExtendedPrice,@DiscountRate,@DiscountAmount,@ValidFrom,@ValidTo,@InvoicedQuantity,
@BackOrderYesNo, " &
```

```
"@BackorderQuantity,@LineTotal,@DeliveryDate,@TaxRate1,@TaxRate2,@TaxRate3,@TaxAmount1,@TaxAmount2,@TaxAmount3,@OrderPaid,@CommissionPaid,@ManageInventory) ", True)
```

### Fields touch by the insert query

OrderNo	nvarchar(50)
ProductID	int
ProductName	nvarchar(60)
Quantity	int
UnitPrice	decimal(19, 4)
ExtendedPrice	decimal(19, 2)
DiscountRate	decimal(10, 5)
DiscountAmount	decimal(19, 2)
ValidFrom	datetime
ValidTo	datetime
InvoicedQuantity	int
BackOrderYesNo	bit
BackOrderQuantity	int
LineTotal	decimal(19, 2)
DeliveryDate	date
TaxRate1	decimal(10, 5)
TaxRate2	decimal(10, 5)
TaxRate3	decimal(10, 5)
TaxAmount1	decimal(19, 2)
TaxAmount2	decimal(19, 2)
TaxAmount3	decimal(19, 2)
OrderPaid	bit

CommissionPaid        bit  
ManagelInventory     bit

**And this is on this code that the error occurs:**

```
Private Sub UpdateOrderlinesDGVWithProduct()  
    Try  
        'CLEAR EXISTING RECORD  
        SQL.DBDS?.Clear()  
        If Me.ProductCBX.Visible = True Then  
            SQL.RunQuery("SELECT * FROM Warehouse.ProductsActivated Where ProductName = '" & ProductCBX.Text & "' ")  
            If SQL.DBDS.Tables(0).Rows(0).Item(0) >= 1 Then  
                Me.InvoicedQuantityTB.Text = 0  
                Me.BackOrderYesNoCB.Checked = False  
                Me.BackorderQuantityTB.Text = 0  
                Me.ExtendedPriceTB.Text = QtyTB.Text * UnitPriceTB.Text  
                Me.SelectedProductID.Text = SQL.DBDS.Tables(0).Rows(0).Item("ProductID")  
                Me.ValidFromTB.Text = SQL.DBDS.Tables(0).Rows(0).Item("ValidFrom")  
                Me.ValidToTB.Text = SQL.DBDS.Tables(0).Rows(0).Item("ValidTo")  
                Me.ProductNameTB.Text = SQL.DBDS.Tables(0).Rows(0).Item("ProductNameOnly")  
                Me.DiscountRateTB.Text = SQL.DBDS.Tables(0).Rows(0).Item("DiscountRate")  
                Me.DeliveryDateExpectedTB.Text = Now()  
                Me.SellingPrice1TB.Text = SQL.DBDS.Tables(0).Rows(0).Item("SellingPrice1")  
                Me.SellingPrice2TB.Text = SQL.DBDS.Tables(0).Rows(0).Item("SellingPrice2")  
                Me.SellingPrice3TB.Text = SQL.DBDS.Tables(0).Rows(0).Item("SellingPrice3")  
                Me.SellingPrice4TB.Text = SQL.DBDS.Tables(0).Rows(0).Item("SellingPrice4")  
                Me.SellingPrice5TB.Text = SQL.DBDS.Tables(0).Rows(0).Item("SellingPrice5")  
                Me.TaxRate1TB.Text = SQL.DBDS.Tables(0).Rows(0).Item("TaxRate1")  
                Me.TaxRate2TB.Text = SQL.DBDS.Tables(0).Rows(0).Item("TaxRate2")  
                Me.TaxRate3TB.Text = SQL.DBDS.Tables(0).Rows(0).Item("TaxRate3")  
                Me.ManageInventoryCB.Checked = SQL.DBDS.Tables(0).Rows(0).Item("ManageInventory")  
                SQL.Params.Clear()  
                SQL.AddParam("@OrderNo", Me.SelectedOrderTB.Text)  
                SQL.AddParam("@ProductID", SQL.DBDS.Tables(0).Rows(0).Item("ProductID"))  
                SQL.AddParam("@ProductName", SQL.DBDS.Tables(0).Rows(0).Item("ProductNameOnly"))  
                SQL.AddParam("@Quantity", QtyTB.Text)  
                If Me.SellingPrice1CB.Checked = True Then  
                    SQL.AddParam("@UnitPrice", SQL.DBDS.Tables(0).Rows(0).Item("SellingPrice1"))  
                    Me.UnitPriceTB.Text = SQL.DBDS.Tables(0).Rows(0).Item("SellingPrice1")  
                    Me.DiscountAmountTB.Text = Math.Round(QtyTB.Text * UnitPriceTB.Text) * DiscountRateTB.Text / 100  
                    Me.LineTotalTB.Text = QtyTB.Text * UnitPriceTB.Text - DiscountAmountTB.Text  
                ElseIf SellingPrice2CB.Checked = True Then  
                    SQL.AddParam("@UnitPrice", SQL.DBDS.Tables(0).Rows(0).Item("SellingPrice2"))  
                    Me.UnitPriceTB.Text = SQL.DBDS.Tables(0).Rows(0).Item("SellingPrice2")  
                    Me.DiscountAmountTB.Text = Math.Round(QtyTB.Text * UnitPriceTB.Text) * DiscountRateTB.Text / 100  
                    Me.LineTotalTB.Text = QtyTB.Text * UnitPriceTB.Text - DiscountAmountTB.Text  
                ElseIf SellingPrice3CB.Checked = True Then
```

```

SQL.AddParam("@UnitPrice", SQL.DBDS.Tables(0).Rows(0).Item("SellingPrice3"))
Me.UnitPriceTB.Text = SQL.DBDS.Tables(0).Rows(0).Item("SellingPrice3")
Me.DiscountAmountTB.Text = Math.Round(QtyTB.Text * UnitPriceTB.Text) * DiscountRateTB.Text / 100
Me.LineTotalTB.Text = QtyTB.Text * UnitPriceTB.Text - DiscountAmountTB.Text
ElseIf SellingPrice4CB.Checked = True Then
SQL.AddParam("@UnitPrice", SQL.DBDS.Tables(0).Rows(0).Item("SellingPrice4"))
Me.UnitPriceTB.Text = SQL.DBDS.Tables(0).Rows(0).Item("SellingPrice4")
Me.DiscountAmountTB.Text = Math.Round(QtyTB.Text * UnitPriceTB.Text) * DiscountRateTB.Text / 100
Me.LineTotalTB.Text = QtyTB.Text * UnitPriceTB.Text - DiscountAmountTB.Text
ElseIf SellingPrice5CB.Checked = True Then
SQL.AddParam("@UnitPrice", SQL.DBDS.Tables(0).Rows(0).Item("SellingPrice5"))
Me.UnitPriceTB.Text = SQL.DBDS.Tables(0).Rows(0).Item("SellingPrice5")
Me.DiscountAmountTB.Text = Math.Round(QtyTB.Text * UnitPriceTB.Text) * DiscountRateTB.Text / 100
Me.LineTotalTB.Text = QtyTB.Text * UnitPriceTB.Text - DiscountAmountTB.Text
End If
SQL.AddParam("@TaxRate1", Me.TaxRate1TB.Text)
SQL.AddParam("@TaxRate2", Me.TaxRate2TB.Text)
SQL.AddParam("@TaxRate3", Me.TaxRate3TB.Text)
SQL.AddParam("@ExtendedPrice", Me.QtyTB.Text * Me.UnitPriceTB.Text)
SQL.AddParam("@DiscountRate", SQL.DBDS.Tables(0).Rows(0).Item("DiscountRate"))
SQL.AddParam("@DiscountAmount", Me.DiscountAmountTB.Text)
SQL.AddParam("@ValidFrom", Me.ValidFromTB.Text)
SQL.AddParam("@ValidTo", Me.ValidToTB.Text)
SQL.AddParam("@InvoicedQuantity", Me.InvoicedQuantityTB.Text)
SQL.AddParam("@BackOrderYesNo", Me.BackOrderYesNoCB.Checked)
SQL.AddParam("@BackorderQuantity", Me.BackorderQuantityTB.Text)
SQL.AddParam("@TaxAmount1", (Me.QtyTB.Text * Me.UnitPriceTB.Text - DiscountAmountTB.Text) * TaxRate1TB.Text)
SQL.AddParam("@TaxAmount2", (Me.QtyTB.Text * Me.UnitPriceTB.Text - DiscountAmountTB.Text) * TaxRate2TB.Text)
SQL.AddParam("@TaxAmount3", (Me.QtyTB.Text * Me.UnitPriceTB.Text - DiscountAmountTB.Text) * TaxRate3TB.Text)
SQL.AddParam("@LineTotal", Me.QtyTB.Text * Me.UnitPriceTB.Text - DiscountAmountTB.Text)
SQL.AddParam("@DeliveryDate", Me.DeliveryDateExpectedTB.Text)
SQL.AddParam("@OrderPaid", 0)
SQL.AddParam("@CommissionPaid", 0)
SQL.AddParam("@ManageInventory", Me.ManageInventoryCB.Checked)
SQL.ExecQuery("INSERT INTO
Sales.OrderLines(OrderNo,ProductID,ProductName,Quantity,UnitPrice,ExtendedPrice,DiscountRate,DiscountAmount,ValidFrom,ValidTo,InvoicedQuantity,BackOrderYesNo, " &

"BackOrderQuantity,LineTotal,DeliveryDate,TaxRate1,TaxRate2,TaxRate3,TaxAmount1,TaxAmount2,TaxAmount3,OrderPaid,CommissionPaid,ManageInventory) " &

"VALUES
(@OrderNo,@ProductID,@ProductName,@Quantity,@UnitPrice,@ExtendedPrice,@DiscountRate,@DiscountAmount,@ValidFrom,@ValidTo,@InvoicedQuantity,@BackOrderYesNo, " &

"@BackorderQuantity,@LineTotal,@DeliveryDate,@TaxRate1,@TaxRate2,@TaxRate3,@TaxAmount1,@TaxAmount2,@TaxAmount3,@OrderPaid,@CommissionPaid,@ManageInventory) ", True)
RecordsCounted()
Me.QtyTB.Text = 1

```

```
        If Me.RecordsCountTB.Text >= 1 Then
            Dim i As Integer
            i = OrderLinesDGV.CurrentRow.Index
            Me.OrderLineIDTB.Text = OrderLinesDGV.Item(2, i).Value
            Me.LastProductIDTB.Text = OrderLinesDGV.Item(3, i).Value
        End If
    End If
End If
If SQL.HasException(True) Then Exit Sub
Catch ex As Exception
    MsgBox(ex.Message)
    System.IO.File.AppendAllText("C:\AutoCashRegister\Documents\log.txt", ex.ToString & vbNewLine & vbNewLine)
End Try
End Sub
```