

Excel VBA Tables Cheat Sheet

Creating a Table:

```
Sub CreateTable()  
    Dim ws As Worksheet  
    Set ws = ThisWorkbook.Worksheets("Sheet1") ' Change the sheet name as  
needed  
    Dim tbl As ListObject  
    Set tbl = ws.ListObjects.Add(xlSrcRange, ws.Range("A1:D10"), , xlYes)  
    tbl.Name = "MyTable"  
End Sub
```

Referencing a Table:

```
Sub ReferenceTable()  
    Dim ws As Worksheet  
    Set ws = ThisWorkbook.Worksheets("Sheet1")  
    Dim tbl As ListObject  
    Set tbl = ws.ListObjects("MyTable")  
End Sub
```

Adding Data to a Table:

```
Sub AddDataToTable()  
    Dim ws As Worksheet  
    Set ws = ThisWorkbook.Worksheets("Sheet1")  
    Dim tbl As ListObject  
    Set tbl = ws.ListObjects("MyTable")  
  
    tbl.ListRows.Add  
    tbl.ListRows(tbl.ListRows.Count).Range(1, 1).Value = "New Value"  
    tbl.ListRows(tbl.ListRows.Count).Range(1, 2).Value = 123  
    ' Repeat for other columns  
End Sub
```

Working with Table Columns:

```
Sub WorkWithColumns()  
    Dim ws As Worksheet  
    Set ws = ThisWorkbook.Worksheets("Sheet1")  
    Dim tbl As ListObject  
    Set tbl = ws.ListObjects("MyTable")  
  
    Dim col As ListColumn  
    Set col = tbl.ListColumns("Column1")  
  
    ' Do something with the column, like renaming it  
    col.Name = "NewName"  
End Sub
```

Filtering Data in a Table:

```
Sub FilterTable()  
    Dim ws As Worksheet  
    Set ws = ThisWorkbook.Worksheets("Sheet1")  
    Dim tbl As ListObject  
    Set tbl = ws.ListObjects("MyTable")  
  
    tbl.Range.AutoFilter Field:=2, Criteria1:=">=100", Operator:=xlAnd  
End Sub
```

Removing a Table:

```
Sub RemoveTable()  
    Dim ws As Worksheet  
    Set ws = ThisWorkbook.Worksheets("Sheet1")  
    Dim tbl As ListObject  
    Set tbl = ws.ListObjects("MyTable")  
  
    tbl.Delete
```

ProjectCub!cle

End Sub

Here's a practical example of working with Excel tables using VBA. Let's create a simple program that automates the process of adding sales data to a table and applying a filter to view specific information.

```
Sub ManageSalesTable()  
    Dim ws As Worksheet  
    Set ws = ThisWorkbook.Worksheets("Sheet1") ' Change the sheet name as  
needed  
  
    ' Reference the existing table or create a new one  
    Dim tbl As ListObject  
    On Error Resume Next  
    Set tbl = ws.ListObjects("SalesTable")  
    On Error GoTo 0  
  
    If tbl Is Nothing Then  
        ' Create a new table if it doesn't exist  
        Set tbl = ws.ListObjects.Add(xlSrcRange, ws.Range("A1:C1"), , xlYes)  
        tbl.Name = "SalesTable"  
        ' Add headers to the table  
        tbl.HeaderRowRange.Value = Array("Date", "Product", "Amount")  
    End If  
  
    ' Add new sales data to the table  
    Dim newRow As ListRow  
    Set newRow = tbl.ListRows.Add  
    newRow.Range(1, 1).Value = Date  
    newRow.Range(1, 2).Value = "Product XYZ"  
    newRow.Range(1, 3).Value = 500  
  
    ' Apply a filter to view sales of a specific product  
    tbl.Range.AutoFilter Field:=2, Criteria1:="Product XYZ"  
End Sub
```

In this example, we first check if a table named "SalesTable" exists on the specified worksheet. If it doesn't, we create a new table with headers for "Date," "Product," and

ProjectCub!cle

"Amount." Then, we add a new row of sales data to the table. Finally, we apply a filter to display only the rows where the "Product" column contains "Product XYZ."

Remember to adapt the code to match your specific worksheet names, column headers, and data. Additionally, you can expand this example by adding more features like updating existing data, calculating totals, or exporting reports based on the table.