|  |  |
| --- | --- |
| https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcR3fSse5oYIfg5ywbMoW58qUszfPdpBtApuvSjTw75OI-vLxs2Uuw | Featherman’s PIVOT() Query Assignment© |

This assignment affords you the opportunity to gain much needed experience creating PIVOT () queries. While PIVOT() queries can cross-tabulate data quickly for later reporting, you can perhaps perform this operation more quickly in Excel or even SSRS using a matrix report. Rather than think of PIVOT() queries as a precursor to reporting, think of PIVOT() queries as a powerful data transformation technique. Some common usages are as a data consolidation technique during extract, transform and load (ETL) operations, or a data transformation step that is just one step in a larger series of ETL steps (like a step in solving a Rubik’s cube).

When analysts perform data analysis in an exploratory mode to better understand a problem, certainly Excel’s pivot tables are phenomenal and allow you to analyze and flip data at the speed of your intellect, rather than at the speed of your ability to remember a query statement. The challenge however is reproducing the insights from week to week as the data changes. DO you the analyst really want to re-perform all the steps necessary to create a series of reports or a dashboard? Often the analyst will call in the IT experts to productionize the pivot report so that next week the process is a one-click experience – this is where the PIVOT() query comes in (also in the various ETL processes noted above).

The overall process of using Excel may be tedious; connect to external data source, download and import data into Excel, clean and filter the data, move columns around, combine columns, make calculated fields (try to limit errors!), etc. This tedious, repetitive work is often given to junior analysts. This pain is only due to the lack of understanding T-SQL and its power to automate processes. While some analysts stay in Excel and use macros or VBA to try to free themselves from repetitive work, you can also automate the data transformation process by packaging up queries into stored procedures and calling them in a series of tasks (perhaps using SSIS, or a series of scheduled stored procedures rolled into transactions – such as on Sunday night. Transformed data can be stored into new staging or destination database tables or series of tables, or the transformed data can directly refresh the datasource that pre-made reports and dashboards draw from.

For this and other reasons, be sure to master the use of PIVOT() queries to complement your usage of GROUP BY() queries. Here are the assignment steps:

1. Use SQL Server Management Studio to generate a list of product categories and sub-categories. Save this query, and the results (copy the column of needed categories and sub-categories into notepad) as you will need these for step two.

2. Write two SSMS PIVOT() or CASE() queries. You can choose to analyze the Internet sales channel or the reseller channel. Be sure to review the PIVOT() notes that were provided to you before beginning this operation. Here is a refresher on the syntax.

 SELECT \* FROM

(SELECT [datacolumn1] that will be displayed in row down the page]

, [datacolumn2 that will make up the columns going across the page]

, [datacolumn3 that will be aggregated]

FROM tables
WHERE conditions to filter the data
 ) AS BaseTable

PIVOT

(SUM([datacolumn3] )

FOR [datacolumn2]

IN ([new column 1], [new column 2], [new column 3], [new column 4]) AS PivotTable

a) First report- analyze the units sold of all the product sub-categories (the rows down the page) for one of the product categories (the WHERE statement). Crosstab the subcategory sales data by countries (which will go across the columns). The countries are Canada, France, Australia, Germany, [United Kingdom], [United States]) the columns.

b) Second report - Crosstab the subcategory sales data (which again will go down the rows) for each month (which go across the columns) of 2007 (the WHERE statement).Use DateName() to pull out the name of the months or Datepart to pull out the month numbers. Again use units sold as the aggregated value.

Copy results into Excel and make one or two charts of the interesting results (one column at a time or one row at a time). Copy the results and charts into a word processing document and add your analysis and recommendations.

3. In this part of your report, please answer the questions that Cristiana the sales manager needs for the new marketing campaign and forecasting. Cristiana wants to institutionalize the Excel-based research you recently performed. She mentioned that she wants to see the analysis on her ipad when she is visiting clients. Her ipad does not have Excel or powerpivot, so we will need to turn the analysis into a web-hosted webpage or report. To start this process, please now make the three queries. You will also need to copy the results into Excel and make some charts (if there is enough data, sometimes the data has nulls in the columns)

a) perform a color analysis (the columns across the page), for each bicycle sub-categories (the rows down the page), examining the unit sales. The color field is in the dimproducts table. Also add a where clause to filter the products table by bikes product category.

b) perform a second color analysis (the columns across the page) for the entire bicycle category, for each of the regions (the rows down the page), examining the unit sales.

c) Perform a follow-on analysis for the different regions (the columns across the page, showing the over 50 products in the different bicycle sub-categories (the rows down the page).

Management now understands that some of the colors are not selling well for the sub-categories, and they want to track this analysis using a report that can be passed to the regional sales managers. They do not want to pass-around spreadsheets.

The final turn-in will have two major sections, each with two reports, written analysis and related charts. No need to copy in the SQL code.