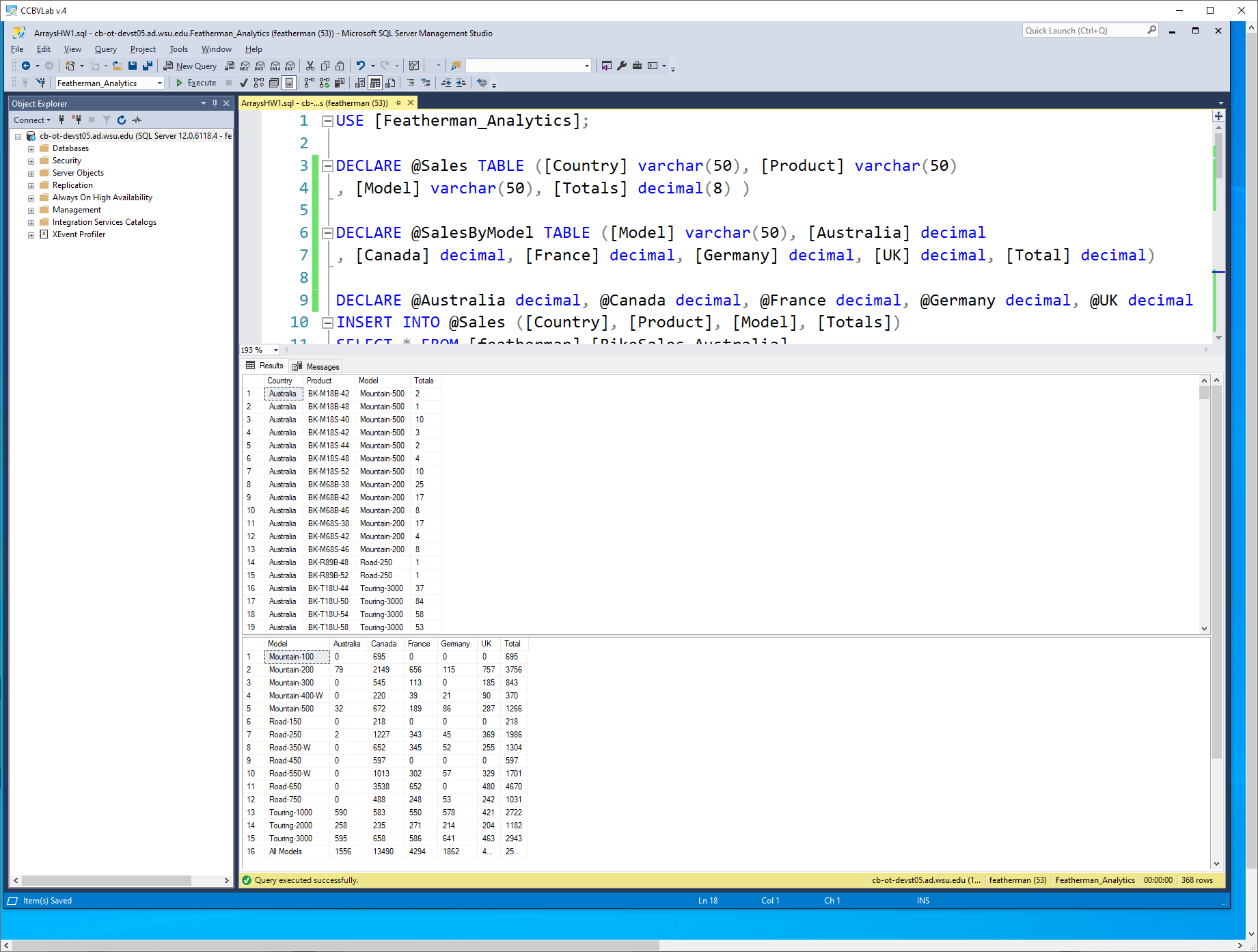
**Arrays Assignment #1**

The recommendations of your recent reports were very insightful and helpful to AdventureWorks management. As such you are building a track record as a DBA/Analyst who can solve data management problems and automate reporting. Bravo!

Today Tim from another division asked if you could help him with a reporting problem. A report stopped working three months prior, when AdventureWorks switched from SAP to Workday software. Previously the report was generated automatically by SAP, but now SAP has been decommissioned, and the report is gone.

Tim sent a screenshot of the tables to show that he has access to the database and tables. Each country manager uploads their sales sheet in a separate procedure. That procedure is still working and loads data into five tables shown in the first image on the right.  
  
The query in the appendix produces a dataset with almost 400 rows of data are presented. You should be able to run the query in the appendix and produce the dataset that matches the second image on the right. If you do not have the tables in your database, then copy them from Featherman\_Analytics into your database (use a SELECT \* INTO FROM query as demonstrated).

To copy the five tables, first close SSMS as necessary then log in using your user ID and password that was issued to you. Then use a query as shown. This copies the [BikeSales\_Australia] table into your database here giving it the table name [Australia]. One table of data has been copied. You need more queries like this to copy more table. Be sure to change the table names!

USE [MF11####]

SELECT \*

INTO [dbo].[Australia]

FROM [Featherman\_Analytics].[featherman].[BikeSales\_Australia]

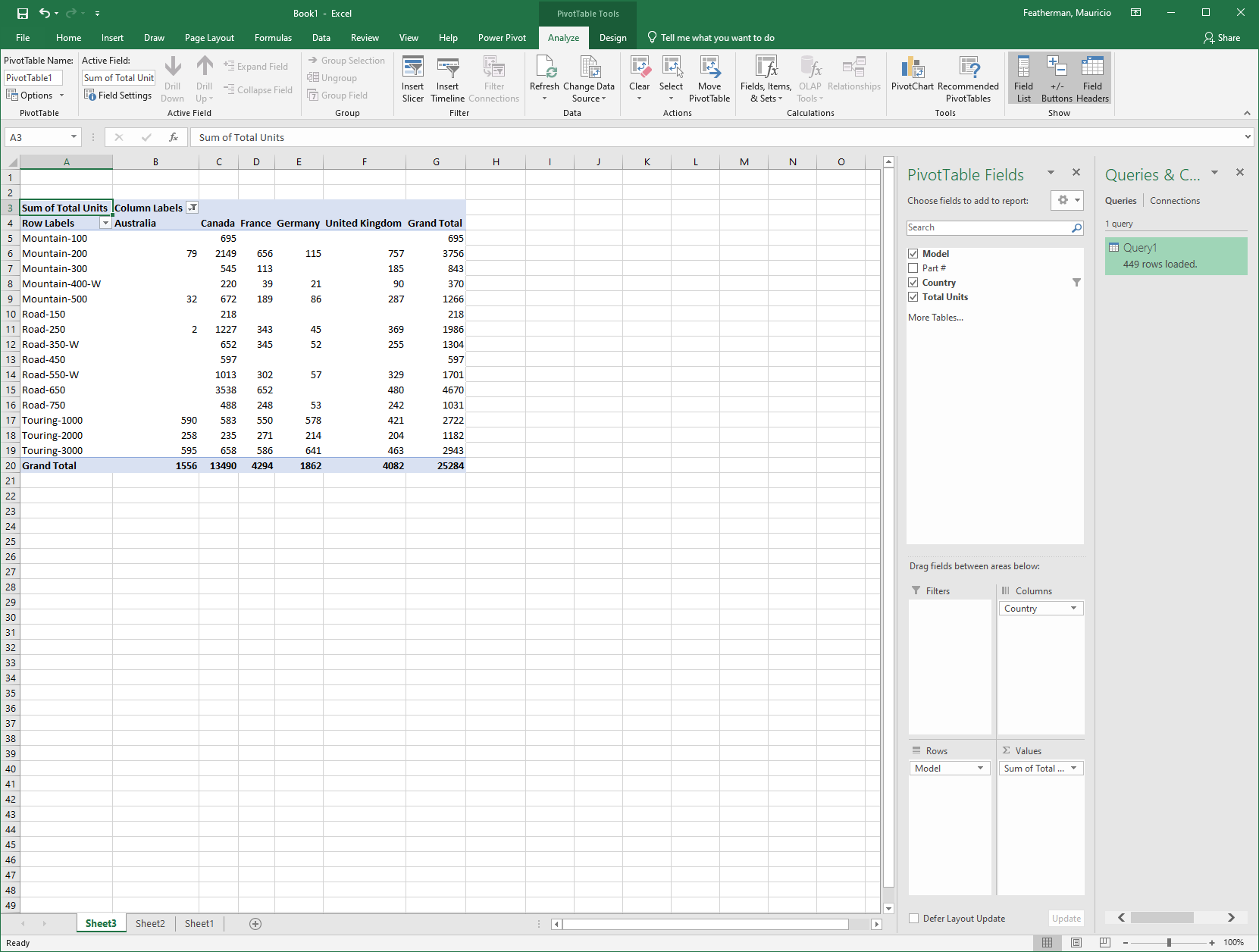
So currently Tim has access to the raw transaction data, and he has the start of a query to pull the non-USA data together (code is in the appendix). Tim does not have any idea how to continue the query to automatically compile the data in the format needed. So that is his request and your next task. You are asked by your boss to work with Tim, to understand the problem, then design and implement the solutions.

Tim tells you that they have been running the query (provided in the appendix) in Excel and producing the report, shown on the next page, using an Excel pivot table. For the past three reporting cycles, the report in the image below was manually produced. The process is not hard but also not automated. Quynh the new CFO has made it a priority that *all repetitive reports are to be automated and verified*. While at WSU, Quynh took analytics class and learned that report automation is possible, but just takes SQL skill and perseverance.

While at lunch, Tim explains the report, the data management requirements, and the urgency of process automation. Tim discussed some of the requirements as follows:

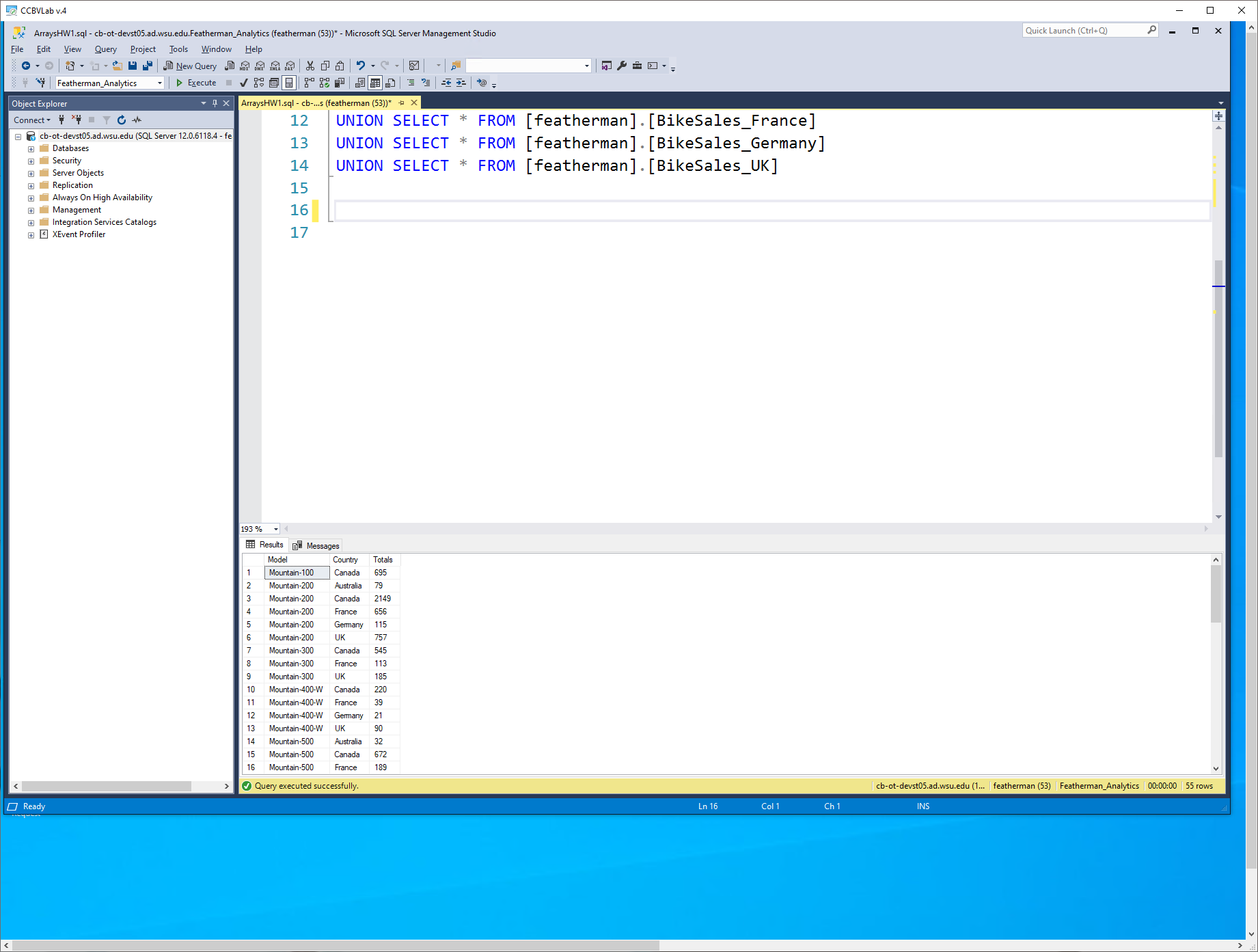
1. The data will continue to be saved into the five country tables (the red arrows above), so we need to use those in the future.
2. The data needs to be combined and saved into a table ON YOUR DATABASE called BikeSales\_All (blue arrows above). This table should show only the current data, so when data is loaded into this table, all previous data should be deleted first.
3. The data needs to be saved into another database table in the format show in the pivot table below (so it can be viewed anytime, using different software). So the data has to be pivoted and saved in this new shape.
4. Tim then discusses that the pivoted data should be saved and displayed in two formats (meaning saved into two different tables) a) numeric counts, and b) percentages (more on this in part 2 below).
5. The request is to create a new system to perform this automated data management and reporting. Tim hopes for a one-click solution.

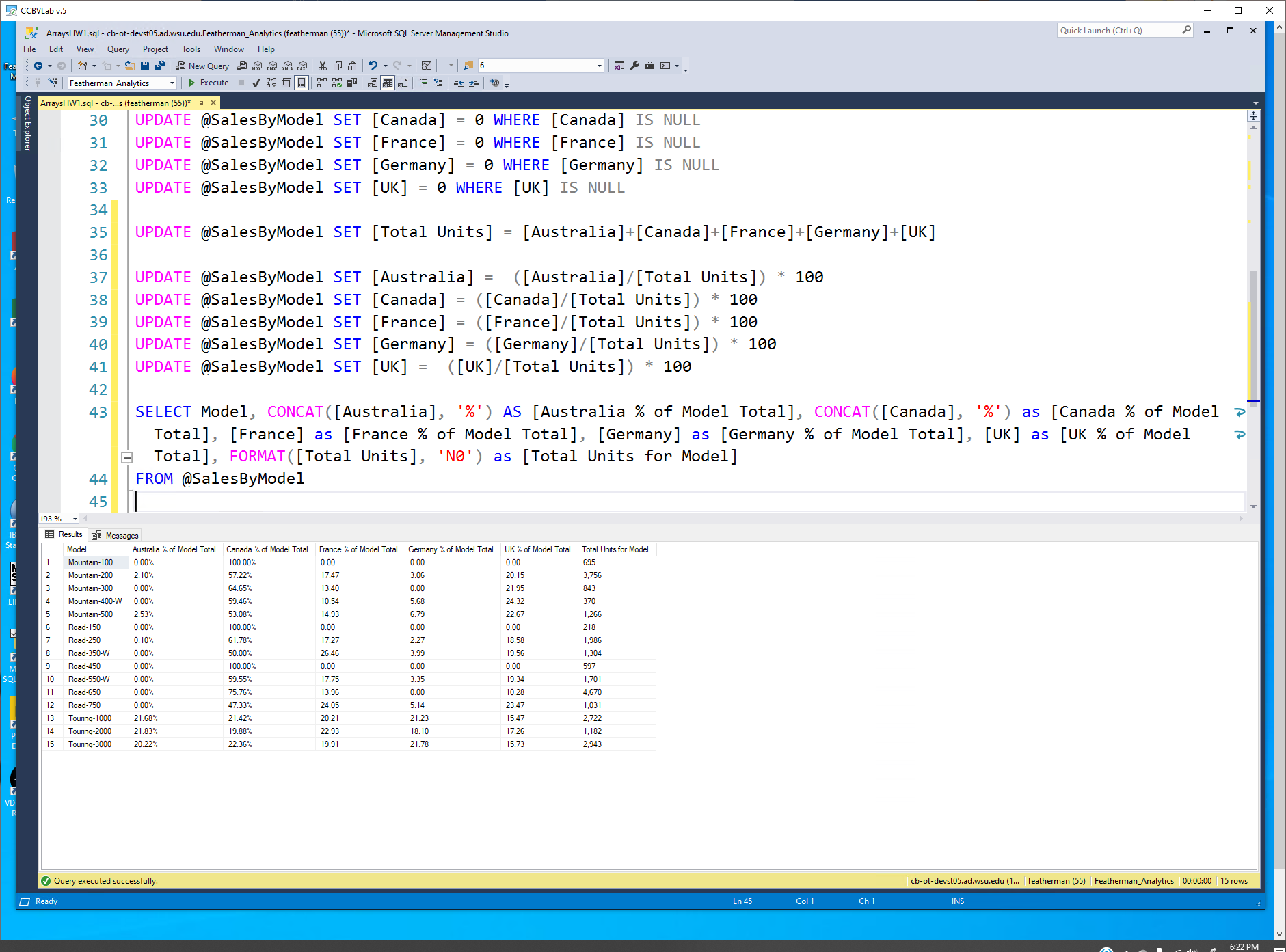
Tim does not have the option to continue manually building the report in Excel (due to CFO Quynh) and mentions that they have further plans to build related datasets. Tim asks if you can produce the same two reports. Tim never took any MIS classes when he attended UW, so you are asked to rectify this problem.

After you copy the data into your database the data is shaped like the image below on the next page, which is a GROUP BY query, by Model and Country for the Bikes Category.

Your intuition tells you that you could probably use a SQL PIVOT formula. You remember your WSU professor mentioned that a PIVOT() query has two parts, the first creates the base data, the PIVOT() part of the query produces the cross-tabulation. Well you reckon’ that you already have have the query that can produce the base data for TIM. Now to specify a) the field for the rows, b) the field for the columns, and c) the number to be counted.

Great! A good start and you are half-way done! The following describes the process and provides more project requirements. The next steps you are planning are:

1. you need to   
   a) copy the 5 tables of data in YOUR database (data for Australia, Canada, France, Germany, UK)  
   b) merge ALL the tables data into one table (show the number of records in the bottom right hand corner of your cropped screenshot of your table data. These prove you moved data into your database.  
   Show the code that performs these data management operations in the appendix of your report (ie that copies and merges all the country data into your BikeSales\_All table (which has the same schema) in your SQL Server database. You will have to create this table. *Add a screenshot of your database that shows you have created this table in your database.*
2. Now you will need to turn this data into a pivot report with the format shown in the image above. You can pull the data from the merged BikeSales\_All database table directly into your PIVOT query.  
   Turn that base data into a pivoted table with the same formatting as shown in the Excel pivot table image above. (Model, the 5 countries, and a grad total column)
3. Just because you can pivot data does not mean it is stored in an array for further processing, so create an array with the same column schema as in the image above. You want to display the models in the rows down the page, and the totals by country going across the page. When you load the array with data you can start by using the following code as shown in the sample.  
     
   INSERT INTO @ArrayforPivot(column names the pivot query will create).
4. Remove the NULLs from the columns in your array using a command such as  
   UPDATE @SalesByModel SET [Australia] = 0 WHERE [Australia] IS NULL
5. You want to add the row of totals at the bottom, and remember that a DBA mentioned that you can use an INSERT INTO statement to add a row of column totals at the bottom of the array. So total each column of the array saving the column totals for each country into local variables.  
     
   You need to create the local variables (one for the total of each country) and load them with column totals, because the INSERT INTO statement (used to add the summary row) cannot have calculations in it. Add the row of summary values, using the example provided to the class (currently in the second .docx of the arrays module).
6. You also make a note to total each row of the array across the page (totals for each model). You surmise that you will use an UPDATE SET command to total the columns for each row. Place a cropped screenshot of your array with the column and row totals into your report. You are creating both the two reports and documentation that outlines each step of the process for Tim and Quynh the Coug CFO.
7. Finally create the code that copies the newly condensed (pivoted) array data into a new table in your SQL Server database. Call the new table PivotedSales. You will have to create this table in your database using code shown in the module. Submit a cropped screenshot of the table in your database, showing the data. Save the query and include it in your submission.

**Part Two**  
  
Tim mentioned that his department also need a second report that uses the format to the right. *The percentages need to total across each row, and down each column.* For the row you can just insert 1 in each column (formatted will be 100%)  
  
The Mountain-200 units for example was sold 2% in Australia, 57% in Canada, 17.5% in France, etc. Management has come to appreciate the row of data however to see an overall metric, for example that Australia accounts for only 6% of total sales, etc. Include your SQL in the appendix.

* Hint: To covert the numbers to percentages by dividing each column by the Total Units – here is an example: UPDATE @SalesByModel SET [Australia] = ([Australia]/[Total Units]) \* 100  
    
  the () \* 100 is used to improve formatting and is optional
* Next display your output with appropriate column headings. You can use the formatting shown for Australia and Canada with the % (this was added using CONCAT() ) or use the formatting as shown for France (no concatenated %), or specify your own formatting.

**Turn-in**

Provide a word docx document that has evidence that the above numbered requirements were completed. You are playing the role of DBA/Analyst. You have been given requirements and you need to produce a professional documentation that outlines the steps you took to produce the requested functionality.

Build a .docx file of your cropped screen shots that has two sections that show the two reports requested. In each section provide documentation of the steps you took. This documentation of the steps taken and the screenshots, and your code, will be saved into a library. The report explains how the data was manipulated to produce the outcome, write the report so that other DBA’s can easily understand your data management processes.

Provide your SQL code blocks in the appendix. Add commenting in each section of each query to explain to Tim what each section is doing.

Note: Be sure to review the Arrays module documentation to practice then use the code examples during the formation of your query plan.