**Popular Analytics Paradigms – *Featherman©***

There are many different types of data that can be analyzed -- sensor data, point of sale data, genomic data, ERP system transaction data, CRM supply chain and customer engagement data, social media text comments, stock market data, etc. Most business classes start with the analysis of structured data that resides in tables with rows and columns (the data often comes from excel sheets, or database tables). Please consider self-study in unstructured data (textual) such as marketing sentiment analysis after this class ends. Consider this your first class in analytics, rather than your last.

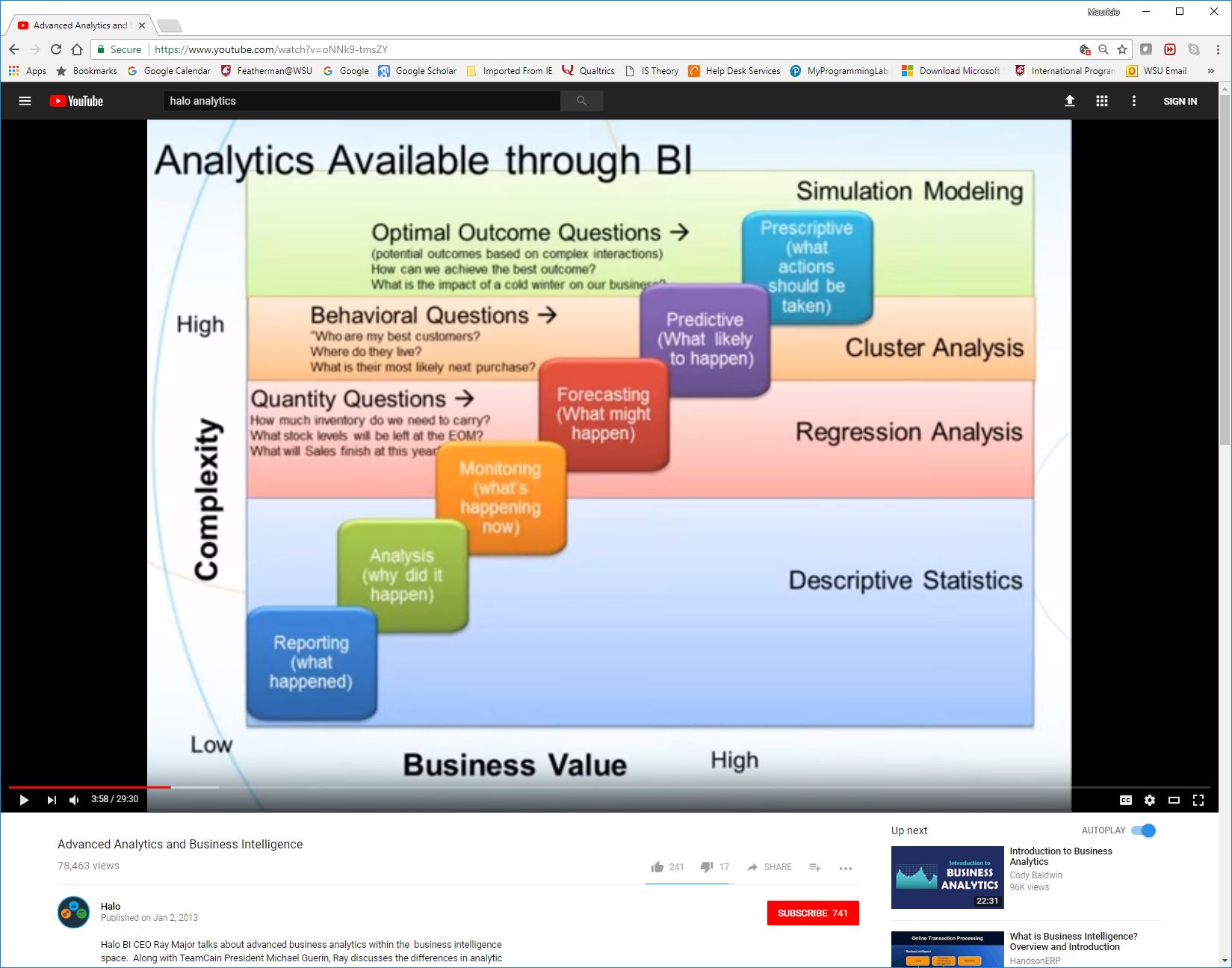
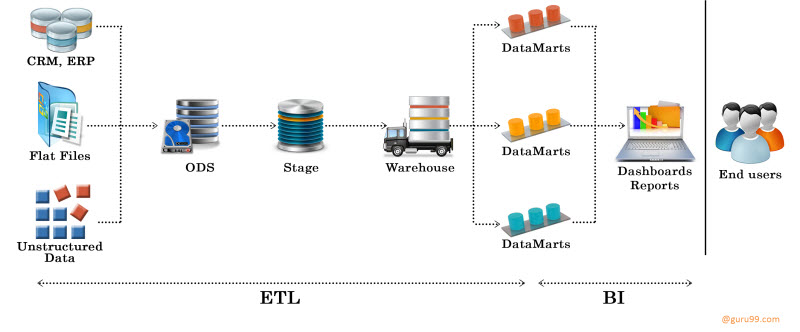
Business analytics begins with a focus on analyzing what happened in recent business processes and performance, comparing it to pre-established goals and prior time periods such as month over month (MOM) and year over year (YOY). The data can come from different systems and hopefully from centralized databases. As you will learn there are different types of analysis to answer the questions; “what happened, why, what might happen, what should we do, and what patterns and trends are evolving” ? Please take a close read of the image on the left.

Figure From Halo Consulting

When the data is structured (i.e., in rows and columns akin to an excel file) and related to business processes and performance, the compilation, analysis, and reporting of that data (and more importantly the recommendations and managerial corrections that are made) often falls into one or more of several common data processing patterns. These patterns focus on the section of the image below titled Business Intelligence” There are radically different ways to get and present reports, some are old, tried and true methods, others are new and deceptively simple to use. You will see these patterns and technologies in the internship and jobs you take.

The purpose of this document is to explain several of the common data management, analysis & presentation work scenarios that exist in corporate work, so that you fit in quickly and be productive on the first day in your new analytics position. You may in fact learn enough in this class to build new systems to semi-automate reporting. You too can be an analytics hero!

Please always remember the goal is to automate as much of the work as possible so you do not have to work 60 hours weeks suffering with hours of copy and paste, and Excel tweaks. Employees that “do everything” in Excel often work too many hours as they did not learn jack about data management and programming. This overreliance on Excel describes 90% of recent business university graduates. Let’s fix that so you can still have a fun social life as you transition to corporate careers.



**Paradigm 1   
Run a saved *SQL Query* against a database to extract a *curated, compiled* dataset then analyze it**

SQL is the lingua franca of databases and data management. SQL has been relied upon for > 50 years to pull data from database(s), both for subsequent compiling, calculations, business processing and to serve as the basis for both status and exception reporting. Different ways to leverage SQL have evolved. Also there are different analytics scenarios. SQL can be used to pull the same set of columns of data needed for reports, over and over on a schedule. This use of SQL in a productionized reporting scenario runs corporate America, and does allow a division of labor between IT workers and business analysts or managers. Analysts can also use SQL in an exploratory analysis scenario, but this requires skills. This first paradigm focuses on regular, scheduled production reports, or refreshed dashboards (running the same sql query will pull the fresh, new transaction data into the report or dashboard for the managers to monitor business operations. Managers typically rely on a small set of reports (and meetings) in this business performance monitoring. The same SQL statement can also be used to alert a manager when a metric goes out of tolerance, such as the number of defects from a manufacturing machine going over a threshold, or the wait time in a phone center queue growing over a few minutes.

The oldest methodology is to pull both dimension (relatively unchanging master) data and transaction (fact and measures) data together into one table. Different ways to pull data into one table for later reporting include pulling the results into the memory of the reporting software, generating a flat file with many columns, or a database table with an explosion of columns, or accessing a view in Excel – all discussed below).

While database professionals stress over separating dimension and fact data into different tables, when it comes time to build reports, one of the oldest solutions is to build and then save the query that merges all the data needed for the report into one results table (called a view of the data). This methodology can get out of hand though when you combine more than one dimension (such as customers) with transactional data (such as sales). Once you bring in the product data, each row can have >50 columns. This dataset generated by the saved query called a view then can be read by Excel, Crystal Reports and all the newer reporting tools. Views have been used for at least 40 years as the way to get big data inside Excel, to the point that Excel crashed from a memory overload. The view consists of a saved SQL SELECT query which is used to compile, merge, and prepare the data for reports and charts. Typically many reports can be drawn from one dataset. A weakness of views is that they pull all the data from the database into the reporting software rather than letting you filter the data based on a criteria such as state or city. An IT professional called a database administrator (DBA) has typically been called upon to create views.

The paradigm of a reporting software ‘calling a named SQL query such as a view that a DBA made’ is common because once a dataset and set of reports is created, then they can be scheduled to run on time intervals, such as weekly on Sunday night. The result is a set of reports that is refreshed with new data weekly and the reports typically keep working with no IT support. The same query runs over and over, perhaps week after week. The report formats and shapes the data into maps and charts. The only aspect that changes is that the data changes from week to week, in that some weeks the report or dashboard will highlight problems that management will stress over; other weeks the report highlights successes.   
  
Called SQL queries allows the DBA to write a massive SQL query just once, have the results verified by accountants to introduce quality control, then the query can run for years. This ‘write once use often’ paradigm allows for just one person to create a multitude of reports and dashboards. Called SQL queries are used to create productionized, repeatable, reliable reporting environments where a series of known reports are refreshed with new data and served to the report users (i.e., brand or plant managers) on a set schedule, such as Monday mornings or end of month. The reports are known as production reports as they represent a scheduled, repeating process where saved SQL SELECT queries are called from a data visualization software and used to fetch pre-compiled or partially compiled datasets. The reporting software recognizes the refreshed data and does its job to generate the charts, maps, etc. The focus is on monitoring well-known business processes rather than interactive, exploratory research sessions. This scenario is so common that since 2010 SQL Server database software has shipped with a built-in report writing software called SSRS SQL Server Reporting Services. The reporting software and the SQL database engine work together very, very well and many firms use this approach since SSRS is essentially free if you own the database software, and 80% of recent corporate database installations have been SQL server.

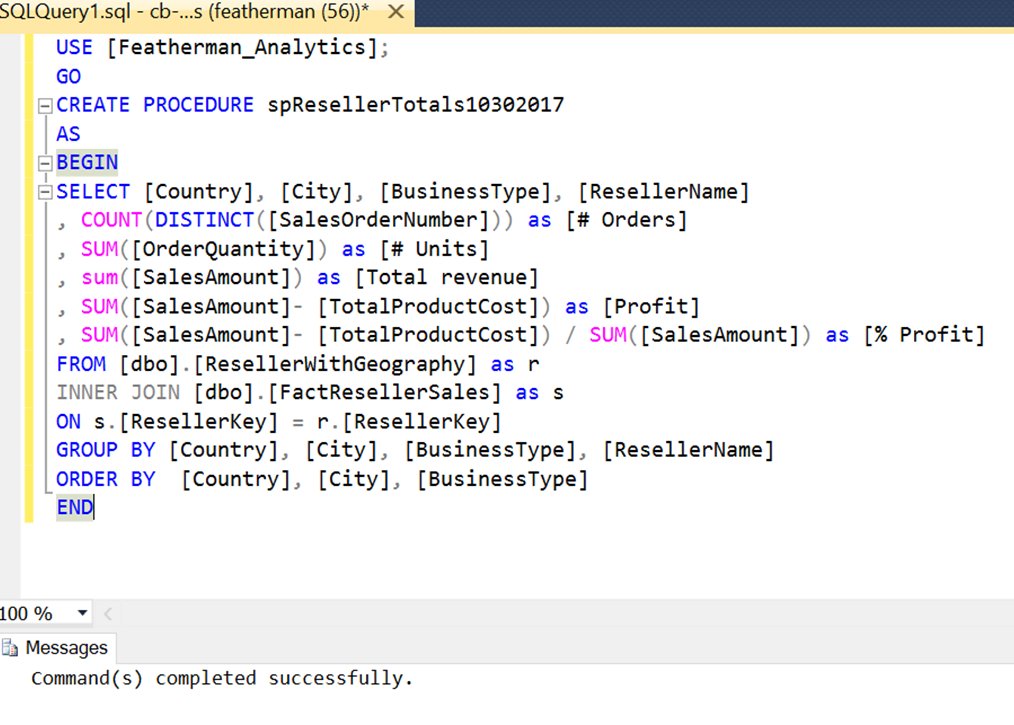
The data that is retrieved using SQL is typically curated. This means the necessary rows and columns of data are retrieved, not ALL the rows and columns. Also the SQL query typically creates new columns of calculated measures, such as ratios, or places a term in a new column after analyzing a value in a row such as $, $$, $$$, $$$$ to signify if the transaction was small ($) or huge ($$$$). The main difference between a SQL trained analyst (or DBA) and the typical office worker is that the latter will attempt to do this data curation in their reporting software, or by good old mindless copy and paste. Did you 10% of all the cells in a spreadsheet are typically erroneous? Now you know one way that business can make the wrong decision…bad data! So please use SQL and have your results checked by an accountant.  
  
Analysts in this scenario use Excel, SSRS or perhaps a professional report writing software that has tons of features. The reporting tool is used as a template to display the data in tables, charts and maps. The analyst has to do their work in the reporting software for example adding columns and metrics (this is not optimal however due to lack of quality control). Little to no data manipulation or data modeling is possible (inability to clean data – collapsing columns or merging columns for example). The concept of connecting to several tables of data (both dimensions and fact tables) is accomplished by calling and running two SQL queries and return two datasets. One dataset perhaps to fill a dropdownlist of filtering options such as selecting multiple stores or states, the other dataset would be filled with data from fact tables. The focus again is on the reporting tool running some magical SQL query to fetch the needed data. In this scenario the analysts can be relatively low wage employees as they are just making simple reports.

The main feature is the heavy use of SQL SELECT queries to pull the data that is the datasource for the report. Because the report is not changing in format from week to week, the query can be created and approved, then live for many years. The DBA manages a suite of saved queries that can each serve as the data source for a series of reports. The DBA must keep track of all the reports and queries. When column names change in databases this can break reports which depend on the named column. Many firms use a series of saved queries, and this methodology can be very efficient.

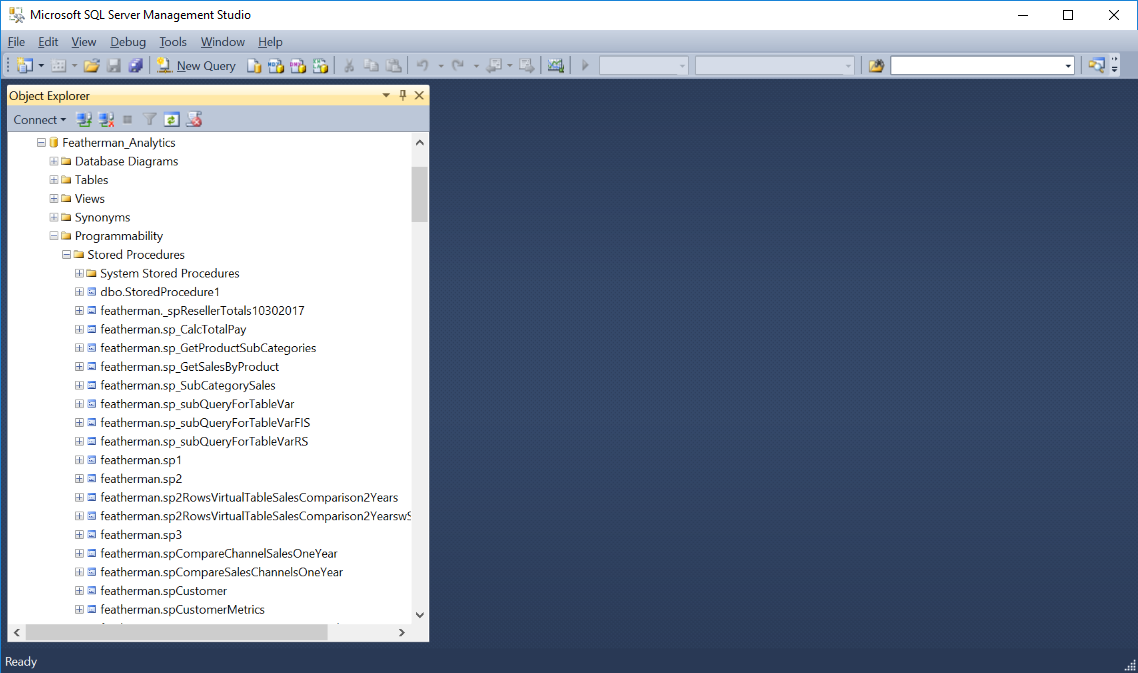
Here is a recap of different ways for the data visualization reports to pull the needed data using SQL.

1. **Executing SQL views** - One of the oldest approaches is for the analyst or DBA to create *views* that are custom-made, saved queries. Views are saved inside the database account. When the reporting software runs the view then the SQL is run against the database and a dataset of nicely formatted rows and columns is returned. Excel has had external data connections for many years that have been able to run views to retrieve a dataset. Use of views paired with pivot charts and pivot table analysis in Excel has been popular for decades.

The benefit here is that inside the query the calculated columns are created, the columns are selected and renamed, and the data is filtered down reducing the number of rows and columns. It is much safer to have the DBA create calculated columns in their query (the view or stored procedure) due to their professional training and focus on quality control, rather than to allow report writers create anything other than a simple calculated field. It is a mistake for the calculated columns to be created with formulas in the report, since the next new report that is needed, will often require similar formulas to be created. It’s better to make the majority of the calculations in the SQL query (however requires a DBA) that can be used over and over, than to bury complex calculations in each individual report.  
  
A weakness is that the view must pull all the data rows that could be needed. The view cannot be parameterized as part of running the report, so any filter the human wants to use must filter the entire dataset that is downloaded, rather than filtering the data that is pulled from the database. A result is that the view must pull all the rows of data that could be needed. If you are accessing millions of rows of data you will be waiting.  
  
In Excel if you run the view, you will retrieve the data into the spreadsheet. In SSRS or Tableau the data is retrieved into the memory of the analyst or manager’s computer.

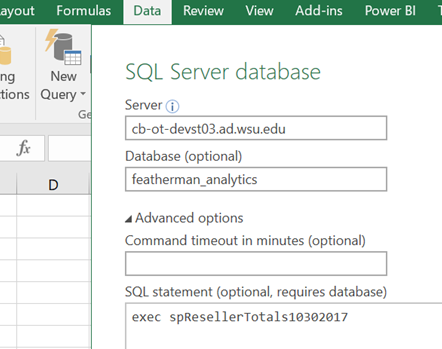
1. **Executing SQL stored procedures** is another way to pull a compiled dataset into a reporting tool such as Excel, Power BI, or Tableau. Stored procedures are SQL queries that are saved in the database, to protect them. Stored procedures are more useful in that they can be parameterized and accept filter values from the report user (i.e., the states selected in a list). A parameterized stored procedure uses that filter when running the SQL statement and retrieve only the data needed.

Views are older, stored procedures newer and thus more advanced. Stored procedures are used to pull processed data into arrays, used to perform ETL, used for data management (for example compiling data at a different level of granularity and saving the new dataset into a new database table (for further data processing, and report generation).   
Stored procedures enable control over big data. Report writing tools (Excel) do not manage RAM memory very well. So, while the reporting tool can connect to big data they get overwhelmed easily. Processing of big data needs to occur on the database level, not reporting level). To perform and automate the reporting process use stored procedures. Shown above is an example of how to create a stored procedure in SQL Server Management Services.  
  
SQL Views and stored procedures provide the dataset, security and information de-cluttering. The report user can see only the columns that they are allowed by the DBA to see (which is a huge security benefit), rather than a lot of other random columns of data.

So views and stored procedures reduce information overload. Stored procedures are parameterizable (meaning they can use filters to reduce the number of rows retrieved and pushed over the internet). Since the heavy computational work is done by the DBA, a firm could easily hire and train an employee to use the reporting tool, which seems simple to use.  


The figure on the left demonstrates where is SQL Server the query is saved.

The limitation of SQL views and stored procedures is that ALL the columns are shoved together and stored in one denormalized table (meaning there are repeating values). While it is true repeating values wastes storage space, newer technologies such as column store indexes reduce this problem.   
  
A main problem of having all the columns in one table is that the report writing software is harder to use, as the dimension and measures data is comingled.

  
The figure on the left demonstrates how Excel can execute the stored procedure to bring the compiled data into Excel for further analysis and reporting. This approach creates a clear job category distinction between DBA’s (high paid) and report writers (modest wages). The report writing tools make the process of report writing pretty simple, the true complexity of the data extraction process is shielded from the report writer.  
  
Many firms use stored procedures to automate data management and report writing, or at least the hard task of retrieving the correct data, cleaning the data, and adding columns of analytics.  
  
In Excel if you run the stored, you will retrieve the data into the spreadsheet. In SSRS or Tableau the data is retrieved into the memory of the analyst or manager’s computer.

1. **Run the SQL query in the database copy the results** - While it is common for a DBA to create a stored procedure and save it inside their SQL Server or ORACLE database for the analyst to run from inside their reporting software (as shown above) there is an older approach that is still common when the data requirement may be one time. The DBA can email the analyst a query (it’s just some text).   
     
   The analyst or report writer has a running version of SQL Server Management Studio – SSMS - (but typically has little idea how to use it) and can start a new query and paste in the emailed query. SSMS returns the resultset (a table of data) and the report writer can copy (with headers) this data and paste it into Excel. Newer versions also allow the results to be added to a data model which is discussed in the next paradigm. This scenario the data is not auto-refreshed as a view or stored procedure can perform, but at least the data is manageable and is more likely to be curated and accurate.
2. A related technology paradigm is Microsoft’s SQL Server Analysis Services (SSAS) is both used to **create data cubes, and to query them**. Data cubes can store big data with no upper limit. SSAS has a programming language called MDX which can query large datasets stored in data cubes. You can connect SSRS or Excel to a data cube (hosted in a data warehouse) and the SSRS can write the MDX code. With advances in data compression and the tabular data modeling approach, data cubes are relevant for only the largest datasets, when the analyst or manager needs super-fast query response times. With the advent of self-service BI, the use of centralized data cubes are less common, however most of WSU’s reporting is based on data cubes.

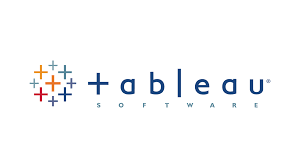
**Paradigm 2 – No SQL. Pull Excel data files together. Download massive amounts of *raw transaction data* into Excel, PowerPivot, or Power BI**

1. **Excel** **Databases (cringe!)** – Historically < 15% of all university business graduates matriculate able to design, deploy and use a database... any database. Typically, employees either use Excel as their database of business activity and use copy/paste and more advanced tricks to pull together different Excel files. Pivot tables and pivot charts have been the preferred reporting tool. The analyst relies on a system of Excel files that do not have much data quality control. The reports are inside Excel also. Many companies run exclusively using Excel databases (run while you can!), the problem is that this paradigm is a huge time waster and should have died in the 1990’s.
2. **Import a copy of raw data into Excel** – The problem here is that once you import a copy of data exported from some system, the data is outdated as new transactions are performed.  
   Many analysts find .csv file extracts of data from different systems and open the data into Excel. You have to be careful how Excel reads columns and datatypes. Currently Excel, PowerBI and Tableau (etc.) can connect to many different data sources. Prior to this advancement, many employees resorted to pulling a data extract from the enterprise system. Most systems current and back to the 1980’s could export a .csv or .txt file. The vendors of hospital systems, CRM systems, ERP or SCM systems, financial systems, etc., often will have consulting divisions whom are happy to charge $400/hr. to create reports for you, or sell their overpriced reporting module and training. As a result in the past they often skimped on the data export capabilities of their software, hence the analyst’s 40 year reliance on .csv files. The benefit is that this type of ETL can work.

The idea here is that rather than buy the reporting module of an ERP system (or similar), the analyst exports raw data and Excel is used to open and format that data. The analyst receives a data dump into some excel or csv file, then spend hours or days carving up and formatting the data per requirements. The problem is that the data is not curated (thinned out). There can be 100’s of columns of data, and it’s hard to access the correct data, and perform analytics. Many, many jobs are this sort of Excel data manipulation and reporting nightmare.  
  
An advantage here is how easy it is to copy data, however Excel has not had great memory management to store big data. Excel 2007 and earlier would crash at 64k rows of data, Excel 2010 at 128k data rows, Excel 2013 would crash at 256k rows, and currently Excel 2016 can store only 1 million rows of data, but may crash anytime. The need to store lots of data in excel (often the norm in the sciences) prompted Microsoft to introduce PowerPivot in 2010 and bake PowerPivot into Excel 2013 and newer versions (discussed next), but to this day many Excel users do not know it exists.  
  
Business consultants have typically made huge revenue moving companies to database oriented systems (Salesforce CRM, SCM, SAP ERP, etc.). With the advent of inexpensive cloud based solutions, one of the hottest careers currently is helping companies move data and processes from local Excel files (or worse) to cloud-based systems. You would be wise to learn about cloud based technologies and practice building analytical systems. Our class together will get you started.  
  
Disadvantages - Unfortunately lots can go wrong when taking complex data from enterprise systems, converting data into a .csv text file, and then converting it again to the reporting software or Excel. Data conversions are needed and this isn’t exactly fun work (which is why DBA’s automate the work with SQL, but we are in paradigm 2 though – the Excel users world). The worst problem is that after doing all the data conversion and clean-up work, the analyst or manager can naturally want to keep and share the Excel file, and not look forward to doing all the work to recreate the file more than once per week (or bi-weekly). The result is that business are often run using ‘shadow Excel files’ (which suffer from quality control). Don’t be surprised to hear that managers alter data to hide bad news.  
  
Another problem caused is that firms often use outdated data when the Excel processes are tricky, time-consuming and painful. Incidentally SQL Server Integration Services (SSIS) was created to solve this ETL problem but only serious IT professionals use this software (however SSIS does not require SQL training). PowerQuery inside Excel today, or PowerBI discussed below attempts to solve this problem of the painful data import.

1. **Import a copy of transaction and master data into a PowerPivot data** **model** – (which is inside Excel) or inside PowerBI data model (the query editor inside Power BI is the same PowerQuery technology that lives inside Excel 2013 and newer if you have the right version). The ability to upload massive amounts of different datatypes and data streams into PowerPivot (PowerBI) is a huge leap forward in technology and usefulness. The Microsoft advertising suggests that you can upload up to 2GB of data into the .xlsx or .pbix data file. The column store compression can remove duplicates in the data columns and compress the size of the files.  
     
   Since most self-service analytics and report scenarios will use < 2 GB of data this modeling is very useful. The use of data modeling and the tabular approach to connecting tables is brilliant and relatively easy to learn. You will learn this approach in our class together.

**Paradigm 3 –** **NO SQL. VIZQL.** **Direct connect the reporting tool to the database tables**. When the column chart is built, behind the curtain, the reporting software generates some type of SQL code to retrieve, process and calculate the data.

1. [Tableau direct connects to data sources](https://www.tableau.com/products/technology)  with its VIZQL technology that was originated at Stanford University. The data does not get imported into the reporting software (nor does it have to be pre-compiled), rather the reporting software must be connected to the cloud database server and mapping servers (i.e., Google or Bing Maps) and the transaction and master data is read. When the chart, table or map is created the data is retrieved, compiled and aggregated. This scenario of course requires fast processing and data transmission and can have problems scaling up to big data. This approach is best used for exploratory analysis. If big data processing is needed, or the same reports are produced on a regular schedule then let the database do the pre-processing (which databases are *very* fast at). So, with direct query be prepared to wait, if the data is huge. Now you know why Tableau is a huge leap forward.  
     
   PowerBI can also function in this same manner.
2. **Direct Connect Excel** on top of a database, data warehouse, or data cube and connect to several datatables and perform pivot table/chart analysis. The pivot table/chart writes data extraction code and compiles that data behind the scenes for you. This methodology of retrieving only the data requested by the pivot query (rather than load all the data into a data model first) is very powerful and many finance professionals use this approach.

*This document is original conceptualizing by Dr. Featherman and represents his views and understandings. Hope it makes sense and sparks you to think about the data management portion of analytics, and also help you decide where you prefer to position yourself regarding the usage of the discussed technologies.  
  
Mauricio Featherman, Ph.D.*