# Bloomberg Spreadsheet Analysis - Module 1

James Evans

05/27/2025

## Introduction to BQL

### How You Will Learn

Hi and welcome to your journey with Bloomberg Query Language, also known as BQL. I'm Elizabeth and I'm thrilled to guide you through this hands-on, practical course that will transform the way you approach investment research. In this course, you'll immerse yourself in real-world scenarios following Summit Capital, a buy-side firm, and Edgewood Securities, a sell-side brokerage. You'll use BQL to filter, analyze, and retrieve key financial data—just like many of our clients do. This isn't just about theory, it's about action. You'll roll up your sleeves and write BQL queries, step by step, test them in real time, and receive instant feedback to sharpen your skills. To make the most of this training, I highly recommend having access to the Bloomberg Terminal and Excel as you go. Practicing BQL in real time is truly the best way to learn and solidify your understanding. Don't forget to take notes along the way. Jotting down key tips, examples, and insights will make it easier to apply what you've learned later. By the end of this course, you'll know how to use BQL to analyze macroeconomic trends, equity, and fixed income data, empowering you to make informed, data-driven decisions. These are the same skills that professionals use to uncover opportunities and solve real investment challenges. So, are you ready? Let's dive in and take some notes and start building your BQL expertise together.

### Introduction to Bloomberg Query Language (BQL)

Welcome to module one, Introduction to Bloomberg Query Language or BQL. In this module, you'll step into the world of investment research, where real-world challenges require efficient tools and data-driven decision-making. You'll join Summit Capital, a buy-side investment firm tasked with managing a university endowment as they face the complex challenge of identifying high-potential investments. Through their journey, you'll discover how BQL empowers analysts and portfolio managers to filter, analyze, and extract actionable insights from vast data sets. By the end of this module, you will understand how Bloomberg Query Language (BQL) is applied in real-world investment scenarios. Learn the foundational syntax of BQL to write queries for filtering and retrieving data. Apply BQL to address a practical investment challenge. Let's dive in and see how BQL becomes an essential tool in solving this realistic investment scenario.

### Summit Capital's Challenge

With their roles defined and the stakes clear, Summit Capital's team faces a pivotal challenge. Delivering strong, sustainable returns for a $500 million university endowment, the team must navigate global markets that present a mix of opportunities and risks—from high growth regions to inflationary pressures—to make sound investment decisions. This is where BQL provides value, offering a tool that empowers the team to uncover insights efficiently and align with the endowment's ambitious goals. After meeting with the University Endowment Board, the team reconvenes to develop a strategy. Their objective is to determine which countries have the strongest macroeconomic outlook, allowing them to prioritize opportunities with robust growth potential by evaluating key indicators such as GDP growth, inflation, and unemployment. Mia, the team's Analyst, recognizes that this complex task can be streamlined using BQL, which she identifies as a tool that can efficiently filter and analyze necessary data, enabling precise, informed decisions. Raj, the Trader, emphasizes targeting regions with stable macroeconomic conditions such as low inflation and unemployment, along with favorable policy rates to meet the endowment's goals. Ritika, the Portfolio Manager, oversees the discussion, ensuring the team's strategy aligns with the client's sustainability priorities while focusing on achieving strong returns. With BQL at the center of their plan, the team prepares to tackle this multi-layered challenge.

## BQL Syntax Basics and How It Works

In the previous video, we saw how the team at Summit Capital is leveraging BQL to tackle their challenges and uncover valuable insights. Now it's your turn to explore how BQL works and why it's such a powerful tool for financial analysis. In this section, we'll start by breaking down the key components of BQL. This includes defining a target universe—essentially identifying what data you're focusing on, like a specific stock or index—and specifying the details you want to retrieve, such as prices or ratios. You'll also want to see how calculations can be applied to refine your data, and how the results are formatted in Excel for easy analysis. By the end of this section, you'll have the skills to construct your own queries and apply them to real-world financial challenges, just as the team at Summit Capital does. Let's get started with the fundamentals of BQL.

The first basic query in BQL starts with the Excel formula `=BQL`, which tells Excel that you're entering a Bloomberg Query Language command. Drag `=BQL` into cell A1. While you can start your query in any cell, for this practice exercise, we will use A1. The target universe tells BQL exactly which security you want to analyze—in this case, Microsoft's stock—so you'll use the ticker `"MSFT US Equity"`. The quotation marks indicate to Excel that you're specifying a particular security. Next, add the field `PX_LAST`, which tells BQL that you want to retrieve the last traded price. In the next section, you'll learn how to look up additional fields to retrieve different kinds of data using BQL. Drag `"MSFT US Equity"` and `PX_LAST` into the query to complete it. Now that the query is complete, let's test your knowledge: What is the target universe? Correct—the target universe is Microsoft, represented by the ticker `MSFT US Equity`. Which part of the syntax indicates the data item? Correct—the field `PX_LAST` specifies the data item to retrieve. Notice that the completed query includes parentheses and commas: the parentheses mark the beginning and end of the query, while the commas separate elements within it. Once we run the query, what output should you expect? Correct—this query returns the last traded price for Microsoft's stock. You've successfully completed this exercise. Click "Try Again" to repeat the practice, or click "Continue" to move on in the course.

So far in this section, we've covered how to define your target universe, which is the specific data you want to analyze, and how to identify the target data, which is the specific information you want to retrieve, like the last traded price or a stock's dividend yield. To include these data points in your BQL queries, you'll need to know the correct mnemonic, such as `px_last` for the last traded price. The Bloomberg Terminal is a wealth of financial information, and `FLDS<GO>`, the Data Fields Finder, is your key to unlocking it. FLDS is a powerful tool that helps you locate mnemonics for any data field, acting like a reference library to ensure your queries are precise and compatible with BQL. It allows you to identify specific data and extract that data into Excel to create customized reports. In this section, you'll take a deeper dive into how FLDS works and engage in a hands-on activity to practice navigating FLDS, finding mnemonics, and using them effectively in your queries. This guided simulation will give you the tools you need to confidently use FLDS as part of your BQL workflow. Let's get started.

First, let's start by looking up a company on the terminal. Imagine you want to gather data on Apple Incorporated. To do this, type `Apple` into the command line and select the ticker for Apple from the list. When a security is loaded for the first time, you'll see a menu with functions specifically for analyzing that security—these are called security-specific functions. For example, with Apple stock loaded, you can run the security description function by clicking `DES` to open the description page for Apple. On Apple's security description page, you'll immediately notice a wealth of information: prices, market cap, dividend yield, and more. While you can explore all of this on the terminal, to pull specific data points into Excel, you'll need to use BQL. To do that, you must find the correct mnemonics in `FLDS`. To open FLDS, click on the command line, type `FLDS`, and select it from the results. The FLDS page opens—think of FLDS as a reference library for all the data fields in the Bloomberg database. At the top, you'll see that FLDS is already focused on

Apple because we loaded it earlier. Let's practice searching for data mnemonics to retrieve data using BQL. First, set the source at the top of the page to BQL by clicking the source dropdown menu and selecting `BQL`. Next, we'll search for a data field mnemonic for use in a BQL query. Let's say you want to find the last traded price for Apple. In the FLDS search bar, type `price` and press Enter. Look at the results—what do you notice? Correct: every field relates to price. `PX_LAST` is the mnemonic for pulling a security's latest price into Excel. On the right, you'll see a description of what this mnemonic means—in this case, "last price." What is the mnemonic for retrieving the highest price? Correct: `PX_HIGH` will pull the high price for a security. You've successfully completed this exercise. Click "Try Again" to repeat the practice, or click "Continue" to move on in the course.

Now that you understand the fundamentals of `=BQL` syntax, let's explore how the `=BQL.Query` formula expands on these principles to enable more advanced and flexible data retrieval. This formula introduces structured clauses, offering even greater control and precision in your queries.

## Strings & Cell References

Now that you've learned the fundamentals of BQL syntax, let's take a closer look at an important concept. Both `=BQL` and `=BQL.Query` can use either cell references or string-based inputs. This flexibility allows you to structure queries in a way that best fits your workflow. Whether you want to reference specific cells for dynamic updates or write a complete query as a string, both methods work seamlessly in Excel. Take a look at the breakdown in the examples below to see how each method works in practice.

## Functions in BQL

Let's talk about functions in BQL and how they help us manipulate data more efficiently. Functions are one of the most powerful features in BQL because they allow us to transform, filter, and analyze data directly within a query. Whether you're calculating an average, filtering a dataset, or handling missing values, functions make it easier to extract the insights you need. For example, if you want to retrieve data for an entire group of securities, you can use a universe function like `members()` to pull all the names in an index. But that's just one type of function. BQL offers myriad functions, including those for cleaning data, such as `dropNA()`, and for calculating trends, like percent change. Throughout this section, we'll explore how functions work in BQL, breaking down their syntax and demonstrating how they can be applied to manipulate financial data and refine your analysis. Let's get started.

The `members()` function expands an index into its individual components, giving you direct access to every security within it. In this example, using `members("SPX Index")` tells Bloomberg that you want to analyze all 500 stocks that make up the S&P 500 Index. Drag `members("SPX Index")` next to `=BQL.Query`. To get the last traded prices of these stocks, use the `get()` function. Drag `get("PX_LAST")` to complete the query. Now answer the question below to check your understanding of what this query will return. What data will this query return? Correct—the mnemonic `PX_LAST` specifies the data item. Notice the completed query includes parentheses and commas: the parentheses indicate the beginning and end of the query, while the commas separate the elements inside. Once we run the query, what output should you expect? Correct—this query returns the last traded prices for all the stocks in the S&P 500 Index. You have successfully completed this exercise. Click "Try Again" to repeat the practice, or click "Continue" to move on in the course.

## Troubleshooting BQL Syntax Errors

Even experienced users run into syntax errors when writing BQL queries. These errors are usually caused by small mistakes—like missing quotation marks, misplaced parentheses, or incorrect mnemonics. But the good news is, most errors are easy to spot and fix once you know what to look for. In this section, we'll go over some of the most common BQL syntax mistakes and show you how to troubleshoot them effectively. Let's dive in.

## Practice Exercises: Syntax Basics

In this section, you'll practice writing basic BQL syntax to apply what you've learned in this module. Before you begin, ensure you're signed into the Bloomberg Terminal and have Excel open. While accessing the terminal is optional, I strongly recommend it for hands-on practice, as it helps reinforce your understanding of BQL. In the following exercises, you'll work with BQL formulas and Excel to build familiarity with the syntax. As you complete each exercise, keep in mind that the data you see may differ from the expected output because Bloomberg reflects real-time market data, which constantly changes. However, as long as the format and structure of your output match the example, your query has been written correctly. Now, log into the Bloomberg Terminal if you haven't already, open Excel, and follow the steps provided below and let's get started.