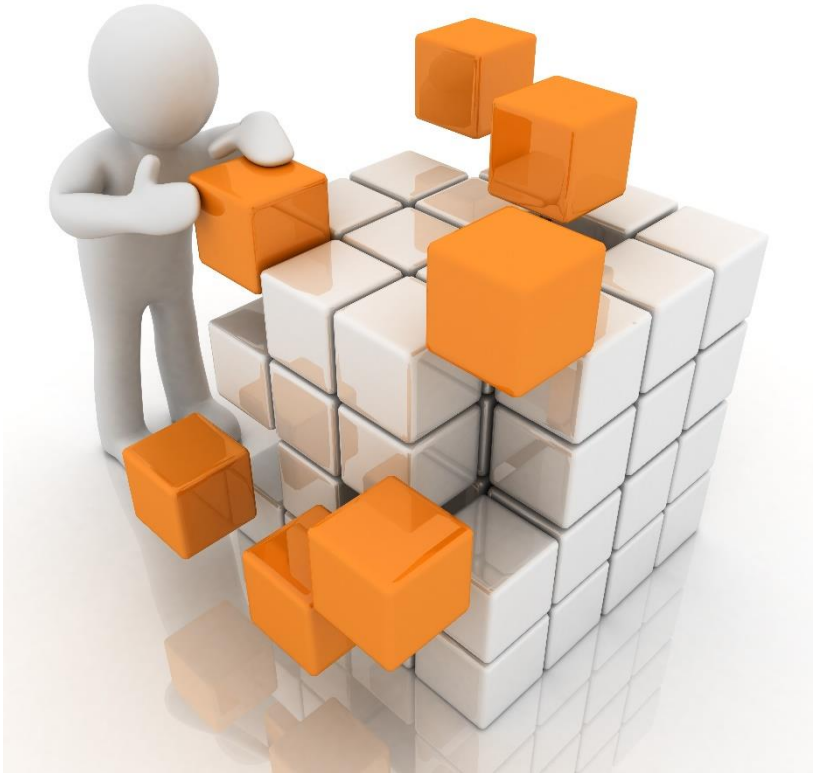
The OLAP Cube Users Answers Book

Microsoft BI SSAS Tabular and Multidimensional OLAP Cube Edition

Why and How You Need Tabular Cubes!

Edition Version 3.4



By

Cary Grant Anderson

Data Scientist

Microsoft Certified Solutions Expert -

Business Intelligence/Data Management and Analytics

Grant@CGrantAnderson.com

Dedication

To Aaron

Contents

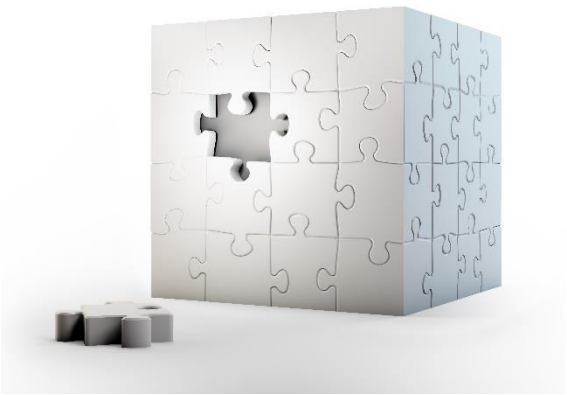
Chapter 1 – Introduction and Fore Words

This book grew out of a need that I found wherever and whenever I went to build OLAP cubes – Both Tabular and Multidimensional. Everywhere I went people, especially the report users who were to use the cubes, wanted to know what OLAP cubes were and how to use them. And how they would make their work lives better using cubes. Down through the years I would explain and teach people about them. I would even develop documentation including Quick Start and User Guides for the cubes I built. I actually started on this book quite a few years ago. Quite a few adventures later you now hold it in your hand (figuratively speaking if you're reading the ebook) and can finally read about what cubes are, both tabular and multidimensional, and how to use them effectively and happily in reporting.

1.1 Book Organization and Audience

This book is organized into a series of chapters that each ask an important question for our target audience, people who are going to use OLAP cubes, primarily Tabular Cubes, for their data and reporting needs. The focus is primarily on Tabular Cubes (which we'll capitalize) but also often applies to Multidimensional Cubes. Both are OLAP cubes and appear as virtually identical to most users in most situations. The differences between them are important and will be pointed out as we go along. Each chapter covers a particular question area and is designed to be rather standalone so that you can just read the individual chapters as you need to rather than having to read the entire book cover to cover. So, there will be some duplication of content from chapter to chapter. That's on purpose so as to prevent and eliminate you from having to hunt around the book for connecting ideas and information.

The Audience for this book are people who want and need to know what a Tabular Cube is, what it does, and how to use it in a business or non-profit organization to significantly augment their reporting systems and reporting abilities. It is specifically focused on report users who need to have a general and basic understanding of what Tabular Cubes are, how they improve reporting and reports, and how to get started using them for reporting.



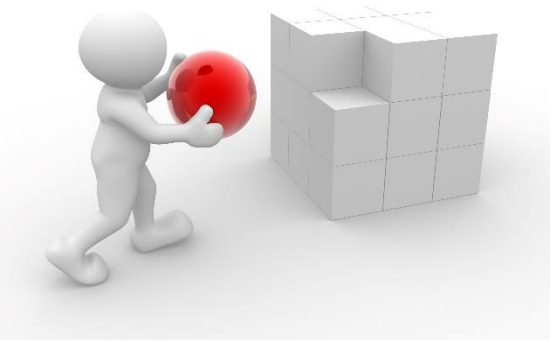
You do NOT have to be a technical person to read, use, and get benefits out of this book.

In fact, this book is oriented to non-IT, non-developers, and non-technical everyday people who only have a general idea about how to use computers, common software, and what reports in a business context and environment are. That's so we can reach the largest and widest audience of people who will potentially use cubes and benefit from them. We give you direct and to the point "Short Answers" up front. And

then "Long Answers" that you can read for more understanding and technical explanations and depth...Or just skip and perhaps refer to later. You can read this book from cover to cover...Or just a question-based chapter or two...Or use this book as a reference book. And there's a useful cube mini-dictionary/glossary for Chapter 12. And Chapter 11 gives you a nice overview of the Microsoft BI stack if you're so inclined.

1.2 What this Book is Not...And Is

This is not an over simplistic "peachy keen" too high-level book that talks around the subject without ever getting actually INTO the realistic aspects of the subject. You see this a lot with authors who have "white board" experience but no real practical experience or understanding of the technical depths and realistic aspects of subject. We don't want to be like that in this book. We try to explain things fairly simply...And then give you some additional deeper information to round out and deepen your understanding. You can just read what you need to or like to.



This is also not a jump right into the step by step use of the product/tool book. That's useful for someone learning to use a tool. But what most of these book miss is the context and the contextual, real world information that you really need but only comes from a lot of real world experience. Nor is this jump right into the technical "deep dive" book.

This book is designed to be:

1. Readable by non-technical people and everyday folks, and
2. Answer the most common and important questions and topics about Tabular cubes from a report and data users,
3. Be full of real world experience, insights, and recommendations that provide exceptionally useful practical information, knowledge, and guidance from years of experience.

This book is not a technical or developer's guide to OLAP cubes. Nor is it targeted to any other audience except report and cube users and analysts. We have different versions of this book specifically targeted to other people's perceptual and work needs.

1.3 The 7 Perspectives

In developing this book, I was faced the prospect of writing a single book to cover all the different types, questions, and situations of cube users, managers, executives, developers, etc. There are a lot of people

typically involved in a cube development project and each one has their own unique information needs and work and goal perspectives.

I call these the 7 Perspectives – The individual perspectives that people in specific job contexts and roles have for those people associated with Tabular OLAP cubes in a business, non-profit, or government organizations.

The 7 Perspectives are:

1. Users – Report and Cubes
2. Executive & Managers
3. Project Managers
4. Developers
5. Testers
6. DevOps
7. Technical Recruiters

Each of these perspectives have their own specific information needs regarding OLAP cubes. While there is a nucleus of core information about cubes that is needed for everyone to know...And this book has that core nucleus...There is a lot of experiential and real-world information that is important and specific to each of these 7 Perspectives. This particular book is focused on the first and most important Perspective – Report and cube users. The people who will actually be using the cubes. And if your people don't use the cubes you will not be able to reap the benefits of cubes and you've just wasted your cube development project.

The next book in this series will be for Executives and Manager and will address information specific to their needs such as cube ROI, project success factors, things to avoid, and other information useful to implementing, developing, and using cubes from that Perspective.

I may continue and write a dedicated book for each of the five remaining perspectives. One book for each Perspective, written and targeted specifically just for that Perspective. A lot of the core nucleus material is common and would thus be the same. But the answers are targeted to each unique Perspective. And there would be significant additional information, answers to specific questions found in each Perspective, that make each book very useful for the target audience of each book. Whether there is sufficient interest in and demand for each will determine whether I write the remaining books in the series. If you are interested in one of the specific unwritten Perspectives books, then please email me at Grant@CaryGrantAnderson.com and let me know which one(s) and what you'd like to see specifically in that book. Of course, I could combine all of the Perspectives in a single book but that would make it a huge book with a too diffuse audience and few if any people would read it.

However, if you'd like to understand the other Perspectives you can read one or more of the other book versions as they become available. This is useful to gain an understanding of the perspectives, drivers, and what's considered valuable and important for each of these areas and roles. I believe that it is important to understand how other people function and perceive things in other roles and responsibilities. What drives them, what they hold important, and how they look at things can make it

much easier to understand them, their roles, actions, and their perspectives. By understanding the diversity and differences in job roles we can all much better work together as a more integrated and understanding team. Corporate American is highly rigid and stratified nowadays and this approach is one that I believe is sorely needed. Just knowing what the other person holds valuable and perceives is a great help with communication and working together with others...True Teamwork. It is something that I think very successful companies include intrinsically in their work environments.

1.4 Author Background and Experience

I have many years of experience with both Tabular and Multidimensional OLAP Cubes. I've seen and developed/built a wide of cubes including:

1. Small cube projects,
2. Medium cube projects, and
3. Huge national mega corporation cube projects.
4. Typical reporting type cubes.
5. Experimental highly complex data analysis cubes (I've pioneered these).

In a great diversity of large and small companies and projects including:

1. AT&T Mobility
2. Cricket Wireless
3. Norfolk Southern Railroad
4. Colorado Housing and Finance Authority
5. Health Grades
6. US Government Contracts including Obamacare Reporting
7. And a variety of large and small startup and other companies for other than cube projects.

And I have worked with many different types of data, data sets, and business environments including:

1. Payroll and Expenses Data.
2. Mortgage and Housing Market Data.
3. Debit and Prepaid Card Data.
4. Financial and Accounting Data.
5. Process Workflow Data.
6. Healthcare and Health Insurance Data.
7. Telecommunications Data.
8. Hotel and Hospitality.
9. Manufacturing Data.
10. Advertising Data.

Cubes are very universal in virtually all data environments and organizations. And they are required more and more by the growing data needs of reporting. And, no, Data Science and Big Data are something rather different and do not directly replace OLAP cubes. Nor do in memory non-OLAP tools except for very small and limited data quantities.

I been doing cubes for many years, most recently Tabular Cubes. And I am a Microsoft Certified Solutions Expert (MSCE) in Business Intelligence, the previous certification, and the current certification MSCE in Data Management and Analytics. And a MSCA in SQL Server 2012/2014. Additionally, I have a wide variety of experience in other areas including:

1. Database Design, Development, and DBA work primarily in SQL Server but also Oracle and NoSQL databases including Microsoft's Cloud Cosmos DB and MongoDB.
2. Windows software development in C# .NET.
3. ASP.NET web development in C#.
4. Server backend development in C#.
5. ETL development in SSIS and also custom C# ETL.
6. Cloud based architecture and management of primarily Amazon AWS but also Microsoft Azure.
7. Project Management, planning, technical management and team leading, architecture, documentation, teaching, training, and mentoring.

I'm also a Data Scientist holding a certificate from John Hopkins University.

Through it all I've done wonders or tried to do wonders if permitted to by management. I've worked for good managers and a few very bad managers and companies that were undergoing some, ah, challenges. All this extensive and diverse experience makes for some very, very interesting stories, perspectives, situations, and real-world experiences.

I've gone into failed projects, and when given latitude to correct the problems, fix them.

So...I'm not your typical rather limited cube developer or architect. I also tell it like it is. I like Reality!

And what's in this book is not your typical cube develop or architect stuff. It's direct, to the point, and based upon real world experience. In this small book I endeavor to give you all of information and insight that I can. I'm not looking for work for or consult with your company...I'm looking to share my expertise and knowledge so that you can better use OLAP cubes and do better reporting.

1.5 How To Best Use this Book

To best use this book, read it all thoroughly if you can...Then think about how all this applies to your organization and to your specific situation. It should all be rather straightforward as to what you need to do...And to avoid. I try to defuse common myths and confusions about cubes. I've encountered so many in my career that I actually started keeping a list some years ago!

You can also use this book as simple reference. Just dip into the chapter questions when you have those questions and find out the answer. The two bonus chapters at the end of the main material will explain the Microsoft BI product terminology and its place in a reporting system as well as common OLAP cube terms, both Tabular and Multidimensional, that you may hear and would like to know what they mean.

I'm very busy creating additional books and courses to help transfer my additional knowledge into these vehicles so as to help others develop outstanding cubes and use them effectively in their organizations. I can't answer individual questions in extensive detail, but you can send me your questions and I can see about posting brief answers/info on the website and updating this book for the next edition.

1.6 Additional Resources on Our Website

We have additional Tabular Cube information and links on our book website. And some free material as well!

Please visit for additional information!

www.CGrantAnderson.com/Cubes

Chapter 2. What's a Tabular Cube?

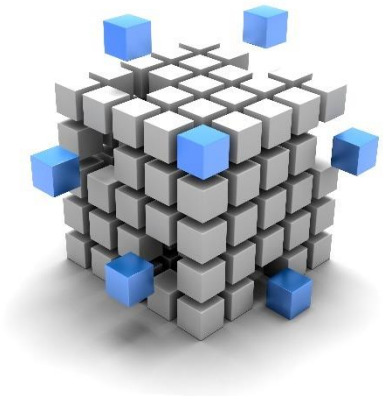
Everyone knows what software is...At least the user interface. Databases tend to be a murkier subject but at least people generally know that data is somehow kept in the database and also kept organized. Cubes are a different subject. People don't really understand cubes. This is not surprising because cubes are a very different thing than regular SQL databases. And like with software and databases people only see a user interface, a GUI or UX, of some sort. The rest of the cube (or software or database) is somewhere else, hidden and unknowable. Nowadays, people just content themselves in knowing that their data is "in the Cloud"... Somewhere... Somehow.

In this section we're going to throw some light on this murky subject. We're going to run down a series of questions and give you the short and direct answers to them. This will let you understand the two main types of cubes (Tabular and Multidimensional) and what OLAP is and a bit about how it works.

All the answers here will be "short" answers – Direct and high level. Without going into a lot of technical detail. A great place to get started!

2.1 What's an OLAP Cube?

An OLAP Cube is a special type of reporting database. OLAP stands for ***O***n***L***ine ***A***nalytical ***P***rocessing. A "cube" in simple terminology. A regular SQL type database is known as an OLTP database which stands for ***O***n***L***ine ***T***ransaction ***P***rocessing. The difference between the two is that the OLTP SQL database holds all your business transactions (records) while the OLAP cube database holds specially built cubes full of some of that OLTP reporting type data but in a special cube type format. This cube data format is used to report on the high-level aggregations (summations and totals) of the transactions.



So, for example, if your company produces computers or other products or services, the SQL relational OLTP transactional databases in your company records the manufacturing, warehousing, sales, and shipping of those products to distributors and customers. This works the same for services but without the physical aspects of manufacturing, warehousing, and shipping of course. When it's time for your sales analysts and managers and executives to know how many products or service instances that you sold in summation (aggregation) then you can just run a bunch of reports on your SQL database. But it's better and faster and more accurate in practice to develop and use OLAP cubes, especially Tabular Cubes, for this reporting. Tabular cubes take a bit of development work but pay off in big benefits and dividends in reporting. Most especially with self-serve ad hoc reporting where you can use an Excel pivot table or other reporting tool and generate any and all reports from a Tabular Cube without having to wait weeks or months for the IT department to design, develop, and deploy static reports for you. Or come up with a bunch of complex SQL queries. Which is a never ending, never fulfilled situation.

Tabular OLAP cubes make developing reports fast and simple and do it yourself. And they are fairly easy and fairly fast to build (especially compared to Multidimensional OLAP cubes which were the previous generation of OLAP cubes).

2.2 What's OLAP Mean?

OLAP stands for ***O***n***L***ine ***A***nalytical ***P***rocessing. It means a database specifically constructed for analysis and reporting. An OLAP database holds information in a Multidimensional form in a relational database in its most common type, called MOLAP. MOLAP stands for ***M***ultidimensional ***OLAP*** and is the type that Microsoft Tabular Cubes and Multidimensional Cubes use. Tabular cubes are actually MOLAP type cubes and thus technically multidimensional. But Microsoft needed to call their new generation of MOLAP cubes something other than "Multidimensional" and so chose "Tabular". Both are OLAP cubes of the MOLAP type. Both are very similar in some ways. And significantly different in other ways. Both products use what's called a Star Schema. Multidimensional Cubes can also use a more complex database table arrangement called a Snowflake Schema. From these, both types of cubes are built. To you who are using the cubes there's not really any difference except the Multidimensional cubes have more features and can be used to build bigger cubes and also analysis type cubes.

Note that no one ever really says, "MOLAP". Everyone just says "OLAP" because MOLAP is the standard for OLAP and it's much easier to

say. There are other forms of OLAP such as ROLAP (Relational OLAP) and HOLAP (Hybrid OLAP) but with the Microsoft cubes it's just OLAP. And everyone almost always just says "cubes" because it's easier and really says the same thing.

2.3 What's OLTP Mean?

OLTP stands for ***O***n***L***ine ***T***ransactional ***P***rocessing. It means a regular type of database that stores transactional type records in a relational form and also uses SQL commands for data operations and reporting. Transactional records are informational records about transactions, such as the purchases and sales at stores. Although data can take many forms, most tables in commercial databases can be considered to be filled with transactional record-oriented data. This works very well for handling this type of data since relational databases were invented and designed for this purpose.

But when there's reporting to do with large amounts of data, relational (OLTP) database will encounter performance limitations. That's where OLAP cubes come in.

2.4 How is OLTP and OLAP Different?

Your common SQL relational database is OLTP in type and is designed for transactional record storage and access...Not for reporting. OLAP is designed specifically for reporting. All this has to do with how the data is organized in and handled internally by the database. The normal OLTP format is what's called a "normalized form" or just "normalized". Essentially, it's like a bunch of spreadsheet data in the main tables and then a bunch of tables that define the acceptable values of the data in the main table. Like cities and states. The idea is to keep from putting in invalid data values, like Australia for the US state. This keeps the data clean. You can image the problems "dirty data" could cause in a database. You get your airline boarding pass and it says that your taking flight number "South America" which is boarding at get "13493.98". It is very, very important to keep the data clean in databases and there is something called "relational integrity" that is used to enforce this cleanliness, especially in reference tables values. These reference value tables are often called "lookup tables" and they play a very important part of keeping the data organized and "clean" in the database and enforcing relational integrity and clean data. OLTP database are also organized for best performance of transaction record workloads.

The OLAP data format is a de-normalization of the data in the main OLTP databases. This simply means that the data is expanded out with redundant values so that it's much easier and faster to report on. And this also prepares it for cubing. Star and Snowflake database schema models are used instead and in place of normal transactional relational models.

Generally, you're not going to worry about or be concerned about all this but it's useful to have a basic understanding when you hear IT people mention these terms.

The bottom line is that to develop cubes a cube developer must create a special database that contains the cube data in an OLAP form. From there it is ingested into the cube and made available in a cube form.

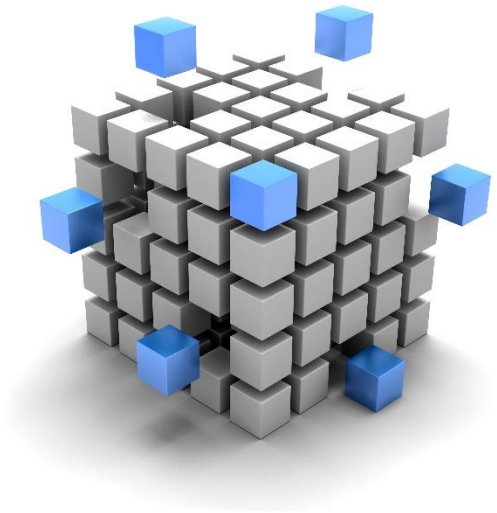
You can of course do reporting with SQL queries in an OLTP relational database but when you find that your SQL reporting queries are taking longer and longer to run and eventually can't run and complete overnight for morning reports then you really know that you need an OLAP cube or two.

OLAP and OLAP cubes were developed primarily because of this problem. They have a number of other major advantages as well as disadvantages which will be discussed later.

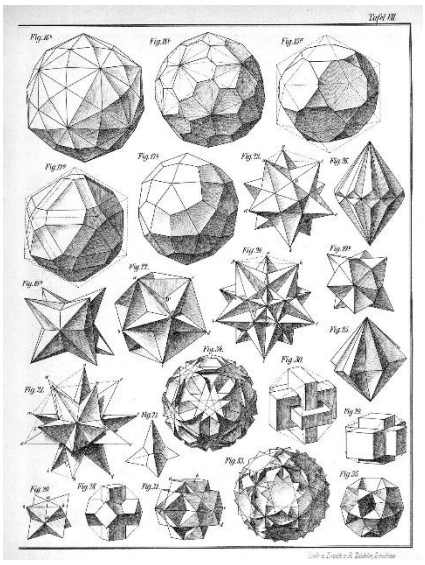
2.5 Why are They Called "Cubes"?

The reason that they are called "cubes" is because they are "multidimensional" and the cube is the simplest true multidimensional geometric form being 3D. Cubes can actually have dozens or more of dimensions, and thus truly multidimensional far beyond just 3D. But the term "cube" is a nicely simple and fast and easy term for them.

Easier



Harder



Max Bruckner, 1906

More Fun!

Grant Cubed!

Note that no one really describes a cube as "32 dimensional" when it has 32 dimensions. Technically, it could be done but in practice everyone just says, "cube". Also note that Tabular Cubes are a second

generation, special type of the first generation Multidimensional Cubes. Microsoft calls them "Tabular" to differentiate them from the first generation Multidimensional Cubes. Nowadays, people will often just call them "cubes" and only specify "tabular cubes" when they need to specifically differentiate them from the multidimensional versions. Most companies will just use Tabular Cubes and not Multidimensional Cubes so the term "cube" now infers Tabular Cubes most of the time.

2.6 What's a Tabular Cube? – The Short Answer

There's several different types of OLAP cubes. The two types supported by Microsoft are:

1. Tabular
2. Multidimensional.

While there are different vendors that sell software and servers to create and host OLAP cubes and the various variations of OLAP cube, the name "Tabular Cube" refers specifically to a special class of OLAP cubes by Microsoft. Both are part of Microsoft's Business Intelligence tool set and are included in the Business Intelligence and Enterprise versions of its flagship database product SQL Server.

The most important things to know about Tabular Cubes are:

1. They are the second generation of OLAP cubes introduced in 2012 and supported by Microsoft's OLAP cube server called SQL Server Analysis Services or SSAS for short.
2. They are much more popular than the first-generation cubes called Multidimensional Cubes by Microsoft.
3. They are a subset of Multidimensional Cubes in terms of functionality. They are greatly simplified, and a lot of features have been dropped compared to the first generation Multidimensional Cubes.
4. With simplicity comes an easier to understand and use developer experience.
5. Tabular cubes are faster to learn to develop for and also generally faster to develop.
6. From a user perspective they look the same (identical) and have the same basic features.
7. Most users would not be able to tell the two cubes types apart.

And from a more technical perspective:

1. From an internal perspective, Tabular Cubes use a column oriented internal structure that is always keep in memory (i.e. not written or paged to the disk) instead of the more common database row structure and orientation (that is paged to disk). This can greatly speed up computations and data retrieval. Although it does limit cube size to the server's available memory. And this is why it's called "Tabular" for the data is internally organized and queried by columns instead of rows. Another name for this type of database is "Columnar Database". It used to be called the "VertiPaq Engine" in the early days.

2. There has been a large number of claims made about Tabular (Columnar) databases and data organization being much faster than the older row-based method and it is true...Mostly. For

certain types of queries Tabular can be something on the order of 20 times faster than row based. However, not all queries can use just the columns so in terms of overall processing speeds, depending upon the data and such, it may not be that much faster. Also, the entire overhead of the Tabular database or cube server must be taken into account, as well as network delays (latency), so the practical real-world performance may not really be all that much better. It depends primarily on the exact type of data and the exact type of queries that are benchmarked. I've seen comparisons of both Tabular and Multidimension that do not show much difference at all or even Multidimensional faster...Please see our website for specifics.

3. However, the conventional wisdom is that Tabular Cubes are faster, and in most situations, they should be. And it wouldn't necessarily matter if they were the same speed as Multidimensional Cubes – Tabular Cubes are faster and easier to build so people will always use them wherever possible over their Multidimensional cousin.

4. The BIG and most important advantage of Tabular over Multi-dimension is that Tabular cube develop is MUCH easier to understand and to develop for most people. Tabular development uses the same database tables (for the most part as do Multidimensional) and outputs the same type of OLAP cubes (for the most part) but it's much easier for most people to learn Tabular Cube development and to implement it with an Excel like interface that uses DAX. Which is quite a bit like Excel macros. And faster and easier to learn and use than the challenging MDX of Multidimensional Cubes.

5. So...You're almost always going to be building and using Tabular cubes from now on unless you have some very complex analytics type cube needs that Tabular just can't handle.

2.7 What's a Multidimensional Cube? - The Short Answer

A Multidimensional Cube is Microsoft's first generation OLAP cubes supported by their SSAS (SQL Server Analysis Server) cube hosting server product. Multidimensional Cubes can handle huge amounts of data, huge numbers of users, and you can build extremely complex analytical type cubes with them. I've built and worked with a large number of Multidimensional Cubes and learned how to quickly and easily build them, how best to construct and use them, what to avoid and not to do, and best practices. I can build them very quickly and easily, much faster than anyone else I've worked with or know. But most people find them rather intimidating and hard to build. Especially with MDX. Multidimensional Cubes had a fairly brief surge of interest in the 2005 – 2010 time period then interest waned in them quite a bit.

The reason for this was twofold:

1. The learning curve was high, and
2. There wasn't a lot of guidance on how to make cubes work well, how best to develop and use them.

So, there were a lot of cube projects that had problems, sputtered out, or died altogether. I believe that a lot of the small to medium sized companies that tried them dropped them for these primary reasons.

And that the big companies stuck with them (or tried other vendors or solutions) due to the size of their data, their data needs, and their data problems. You can't really do night time batch reporting when a complete SQL report query takes 16 hours to run. So, when a cube can do that in 45 minutes or less, well, you have to do cubes even though they're complex for most people and companies. Sometimes you just gotta have cubes to get your reporting done!

So as interest waned for Multidimensional Cubes. And I decided not to spend time publishing books or courses on them. However, the announcement in 2012 of Tabular Cubes and their adoption over the next several years rekindled interest in OLAP cubes and has breathed a lot of new interest in using them. And rightfully so! While I can develop Multidimensional Cubes just about as fast as Tabular Cubes, most people just can't. In all my positions, I've always been able to build my cubes in a fraction of the time of other people. That's because I understand cubes and cubing and I'm very good at it. If you approach cubes from the correct development perspectives, they are very easy to build. But that's a topic for another book.

Multidimensional Cubes failed at widespread adoption due to the high learning curve not only for all the various knowledge and skill needs of cubing but also because of one big factor from my perspective: MDX. MDX stands for ***M***ulti***D***imensional e***X***pressions. It is the language of programming Multidimensional OLAP Cubes. And while it appears to be much like SQL it is actually quite different. Think about the analogy of using a power boat on a lake versus using a large sailboat in the open ocean. A very, very different scenario and situation! MDX is obtuse. Its syntax is non-intuitive. It's complex and at many times very confusing. That's where Mosha came in. Mosha Pasumansky was a Microsoft employee that blogged quite a bit and quite regularly about cubes and SSAS and MDX. Mosha was responsible for answering a huge amount of questions about MDX and teaching a lot of people about it and all its peculiarities and quirks and power too. Everyone who did OLAP cubes with SSAS back in the "first generation" days knew of Mosha and his wonderful blog. Then one day in 2009...It all stopped. No more blog entries or articles by Mosha. His final blog entry on December 29, 2009 revealed that he had moved over to the Bing search engine group about a year previously. That explained the spottiness of his OLAP blog entries. And that was a very bad day for Multidimensional Cubes.

I saw interest wane greatly after that. Multidimensional cubes are great technology and when used properly and well are great additions to any company's BI and reporting systems. However, the complexities involved with them limited their use and adoption unfortunately. And new technologies became popular and Microsoft did not promote them. Instead the siren song of brand new technologies beckoned to one and all. I continued to do cubes for a while then got interested in Data Science, more software development, and other things. I finally got back to cubes with Tabular Cubes once the interest picked up in the marketplace.

While I can tell you a huge number of things about Multidimensional Cubes, and that a very large amount of these things does apply to Tabular Cubes, the main things that you need to know about Multidimensional Cubes are:

1. They are Microsoft's first generation OLAP cubes.
2. They are still supported but you cannot run Multidimensional Cubes and Tabular Cubes in SSAS on the same server.

   a. You can, of course, install two instances of SSAS on a server using different ports, one for Tabular and one for Multidimensional but most SSAS installations nowadays are just Tabular on the default instance.
3. Multidimensional Cubes have more features and functionality than Tabular Cubes and can handle very large cubes easily.
4. You can also build complex Analysis Cubes as I call them. No one that I can find has published anything on them or how to build them but the needs and most uses for OLAP cubes are for fairly simple Reporting Cubes.
   a. Interestingly enough, along with OLAP cubes, Microsoft announced years ago a Data Mining product which was ignored by the marketplace. Now, years later, Data Science and Machine Intelligence is all the rage. But Microsoft had a Machine Intelligence product, Data Mining, many years before this. Microsoft was well ahead of the times here. It's just that people didn't really understand exactly what it was and how to use it at the time.
5. Tabular Cubes, the second generation, while not having all the features and power of Multidimensional Cubes is perfectly adequate for most cube situations and scenarios.
6. So...Tabular Cubes are and will continue to be much more popular than Multidimensional Cubes.
7. Multidimensional Cubes will still be used for very large and complex cubes.
8. If you're just getting started with cube than Tabular cubes are the way to go.

2.8 How are Tabular and Multidimensional Cubes Different?

Here's a handy high-level summary of how Tabular and Multidimensional Cubes are different:

OLAP Cube Differences – The Top 12			
#	**Aspect**	**Multidimensional**	**Tabular**
1	Complexity	High	Moderate
2	Handles Complex Data Situations	Yes	No
3	Uses Star Database Schema	Yes	Yes
4	Uses Snowflake Database Schema	Yes	No
5	Handles Large Data	Yes	No
6	Handles Huge Data	Yes	No
7	Ease and Speed to Develop	Slower	Faster
8	Challenge of Learning and Using	High	Moderate
9	Challenge of Cube Programming Language	High	Moderate
10	Uses SSAS Server	Yes	Yes (See Note #1)

OLAP Cube Differences – The Top 12			
#	Aspect	Multidimensional	Tabular
11	Users Can't Generally Tell the Difference	Yes	Yes
12	In Memory Only	No	Yes (See Note #2)

Notes:

1. You cannot run SSAS to support both Multidimensional and Tabular at the same time on the same SSAS server. You can on separate instances of SSAS on the same server however.
2. Tabular uses in memory only, i.e. it cannot and will not page to disk (ala virtual memory). When your server runs out of RAM...Your Tabular cube stops.

There's other differences as well in the features and functionality between the two.

Here's a quick list of the feature and functionality differences:

OLAP Cube Differences – Features and Functions			
#	Aspect	Multidimensional	Tabular
1	Actions	Yes	No
2	Aggregations	Yes	No
3	Calculated Column	No	Yes
4	Calculated Measures	Yes	Yes
5	Calculated Tables	No	Yes1
6	Custom Assemblies	Yes	No
7	Custom Rollups	Yes	No
8	Default Member	Yes	No
9	Display folders	Yes	Yes1
10	Distinct Count	Yes	Yes (via DAX)
11	Drillthrough	Yes	Yes (depends on client application)
12	Hierarchies	Yes	Yes
13	KPIs	Yes	Yes
14	Linked objects	Yes	Yes (linked tables)
15	M expressions	No	Yes1

OLAP Cube Differences – Features and Functions			
#	**Aspect**	**Multidimensional**	**Tabular**
16	Many-to-many relationships	Yes	No (but there is bi-directional cross filters at 1200 and higher compatibility levels)
17	Named sets	Yes	No
18	Ragged Hierarchies	Yes	Yes1
19	Parent-child Hierarchies	Yes	Yes (via DAX)
20	Partitions	Yes	Yes
21	Perspectives	Yes	Yes
22	Row-level Security	Yes	Yes
23	Object-level Security	Yes	Yes1
24	Semi-additive Measures	Yes	Yes
25	Translations	Yes	Yes
26	User-defined Hierarchies	Yes	Yes
27	Writeback	Yes	No
28	Aggregations	Yes	No

Source: https://docs.microsoft.com/en-us/sql/analysis-services/comparing-tabular-and-multidimensional-solutions-ssas

Chapter 3. Why Do We Want To Use Tabular Cubes?

That's a good question! The good answer is that we are using Tabular Cubes because of all the benefits they bring us. As well as avoiding a couple of significant problems when and where they are not used. The use of Tabular Cubes shows that your organization is proactive and is serious about using technology to make work more efficient and more accurate.

Note that this chapter question pre-supposes that your organization is already using cubes and is perhaps giving you this book as part of an educational initiative to educate its people about cubes. If not, then this chapter will inform you as to why your organization should be using them.

Tabular Cubes are important for many reasons.

3.1 Why are Tabular Cubes Important?

Why are OLAP cubes important? There's a number of reasons. We're going to go through the reasons, but before that I'm going to sketch out the overview, and then we're going to talk about each reason individually.

#1 – Cubes Do Things That You Cannot Effectively Do With SQL

Cubes are important because, number one, they let you do things that you can't do without cubes. That sounds pretty simple, but what actually do I mean by that? Well, you can look at data in different ways, in new ways, that would require a custom report. You can do this with what's called an OLAP cube browser or a reporting tool that has functionality like a cube browser. With an OLAP cube browser, you can actually let the user create their own custom reports. That sounds great! Why haven't we been doing this? Well, there's a number of reasons, but if you have Excel, you have a pivot table as well as a pivot chart and you are probably already doing something very similar to this. We'll talk more about this when we talk about user interfaces.

#2 – Cubes Can Browse The Data

The second things that OLAP cubes let you do is to let you browse the data. You can actually go in and look around and do what's called data exploration. A very popular data science term nowadays. But you

don't need data science tools to do it – Just a cube. Browsing the data is significantly more useful, and easier, than trying to explore the data through a bunch of static and dynamic reports.

So, what's the difference with this? How can I describe the difference between data browsing and exploration with cubes and paging through hundreds of pages of old fashioned reports?

The difference is really simple. Imagine yourself out exploring…Taking yourself to a local shopping mall. We've all been to a local shopping mall. Now imagine all the lights are out. There's no light anymore, it's really dark. So, you hold out your hand, you get a flashlight, you turn the flashlight on, and you start looking around the mall. You've got two minutes. I want you to take the flashlight, which isn't a really high-powered searchlight, and I want you to look all over the mall and tell me what you can find out about the mall. In just two minutes. Tell me every store that's there. Tell me all the sporting goods stores, tell me all of the clothing stores, then break them down by female to male, or both, and then tell me all the restaurants, and then tell me if there's a bookstore or two in there. Etc, etc, etc. You've got two minutes, you'll be running all around, and you won't be able to do it. Now...Bring all the lights up, look around, look wow, a store directory. So, now you have a store directory, you look at that, and you have everything you need in less than two minutes. Wow!

That's how OLAP cubes works. Cubes let you explore your data from a high level down to the smallest detail. In any combination you want. Exactly as you want it. It's the difference between hunting in the dark with a flashlight...And having all the lights on!

#3 – Cubes Can Provide Data Insights

The third thing that cubes let you do is it lets you look at the data, not only in new ways, but cubes also lets you look at it in a very concise and comprehensive way. There's a very important principle in business intelligence called one version of the truth, and one version of the truth. This department does a report from here, this other department does this report from here, and then when they compare the results they're different. Oh oh! Why are they different? Well, it's because they're using different databases and there's some variations between them. Oh, but, isn't data just data? How could it be different? And the answer is, depending on the formatting, the storage, the time period, a bunch of other considerations, the data can be quite different. If an organization is calculating data, which every organization does that, you may have some of that data in Excel spreadsheets. So, the database may not be current, and may not even reflect that data. You may have to go to a spreadsheet to really find out what's going on. And let's face it, a lot of spreadsheets are used in organizations that, if it was put into a database or also maybe into an OLAP cube, you'd have that information automatically instantaneously, you wouldn't have to work on it.

I just had lunch with a manager at a major US corporation, and he was telling me that they have already put together some OLAP cubes, and these cubes are saving one person in particular, one department, four hours a week. Okay, that's not bad. He had an estimate that they're saving eight hours of that person's time. This can continue and continue, and what about the other people that ask to see the data? Well, you're a company, and the internal aspects of companies really work on data nowadays. So that gets into the fourth reason, that's accessibility to data. When you have accessibility to data, you can

actually know what you really need to know. And that is not only priceless but a necessary requirement for getting one's work done and being successful in this day and age.

#4 – Cubes Provide Complete Data Accessibility

Number four is the accessibility of the data. The OLAP cubes really facilitate this. What happens with OLAP cubes is that you can look at data you could not look at previously, or was extremely difficult to do so, and you had to download it into Excel and spend a lot of time approving repots. You can also discover the data and you can also automate the data. And this also gives us the ability to automate reporting. And automated reporting is really what it's all about. You can actually create reports automatically and/or on demand, so people don't have to get the data, download it, do a bunch of things in Excel, and then get the data and then repeat that once a month, once a quarter, once a month, once a week, once a day. So, it saves a lot of time. When you add this up across the organization, you get into the question of how much time are we really saving? It's usually very hard to quantize. That manager was basically telling me that they were adding some cubes and adding reports and getting to the point where half of this person's work week was saved from doing reports. They just look at the cube, do a cube report, and it's done. They don't have to download the data to Excel, and sit there crunching it, or import it into some of the other reporting tool or product.

I've seen some products that were intended for one thing being used actually to create data summaries and not efficiently at all. At one company that was not using cubes at all, they actually had expensive database developers generating reports. They had one, then two, then three, then four, and they kept adding to the staff. And I asked, "Why don't we do a cube on that?" And their response was, "No, no, no, we don't want to do a cube on that. Too expensive!" And having five plus expensive database developers constantly churning out reports is not overly expensive? There can also be some not technical but business reasons and also political reasons not to cube data and get to one version of the truth. Sometimes something hides in the data, and some manages don't want that data to see the light of day.

Some managers actually want to keep the data partitioned, siloed, and protected so they can "protect" or even hide data that they don't want to share with others. This is one of the downsides when you get into cubes, is the cubes can actually show you problems in your organization and actually might bring to light some information that some managers don't want to have brought to light for political reasons or even for reasons of poor performance or data quality.

#5 – One Version of the Truth

Cube provide "one version of the truth". This means that cubes can provide a central data source for all reports that provide validated and valid numbers instead of many different SQL reports providing many different numbers. Sometimes slightly off and sometimes wildly off!

Multiple versions of the truth, i.e. different numbers for the same data and data calculations routinely happen with SQL reports due to a number of factors:

1. Different people write SQL report queries differently with slightly different results.
2. Multiple SQL report queries often implement business rules in different ways producing different results.
3. And multiple SQL report queries often implement quite different business rules entirely.
4. Slight differences in SQL report queries can produce small and almost undetectable differences in results which often get overlooked and yet magnified over time.

Cubes eliminate these problems and provide one version of the truth for data reporting.

#6 – Faster Than SQL for Aggregations

OLAP cubes, both Tabular and Multidimensional, vastly outperform SQL queries for aggregation reporting. They do this because cubes were specifically designed and developed for solving the problem of SQL queries taking too long to do aggregations (summations) in business organizations.

You can't get faster than OLAP cubes.

#7 – Faster Report Development

Report development is also much faster with cubes compared to hand crafting individual and problem prone SQL queries for each report.

In one recent instance, a report developer reported that he was able to develop reports in one tenth the time (10 times faster) than when doing the same reports with SQL queries. What would previously take him 5 days or more changed to being done in just ½ a day with using a cube as a data source. Four hours instead of forty hours! Quite an improvement!

#8 – Do It Yourself (DIY) Self-Serve Reporting

Cubes combined with Excel pivot tables and other reporting tools provide reporting that can be done by report users instead of requiring users to wait for report developers to develop reports for them. Instead, report users can quickly and easily do their own reports without need dedicated report developers. This not only frees up a lot of previously dedicated and expensive manpower but also gives a wonderful and highly productive ability to report users – Do it yourself self-serve on demand user empowered reporting.

3.2 What are the Advantages of Tabular Cubes?

Tabular Cubes have a number of advantages over regular SQL reporting:

1. Cubes are Faster:
    a. Cubes have faster performance over equivalent SQL queries.
    b. Cubes can do aggregations much faster and more accurately than aggregations in SQL reporting.
    c. With cubes one can develop reports faster when compared to hand coding individual SQL queries.

2. Cubes Gets You Unstuck:
    a. When your SQL queries are taking hours to run cubes can do them better and faster, often in minutes.
    b. When daily and nightly reports take all day to run...You need cubes.

3. Cubes Give You One Version of the Absolute Truth:
    a. SQL reports can have subtle errors in joins and aggregations. Cubes don't.
    b. Cubes provide a central single data source for reports instead of multiple custom-built SQL queries for each report which are subject to subtle and difficult to detect errors.

4. Cubes Provide Better and Superior Reporting:
    a. Cube users can browse data instead of having to try to assemble an understandable snapshot by using SQL reports.
    b. Cube users can make their own custom reports whenever they want.
        i. This is known as Self-Serve BI and Reporting.
    c. No more waiting for SQL reports to be developed (weeks and months).

3.3 What are the Disadvantages of Tabular Cubes?

Tabular Cubes have a number of disadvantages over regular SQL reporting:

1. Cubes Take Longer to Initially Build:
    a. Cubes take a bit of time and skill and effort to build.
    b. But the total time spent building a cube is quite a bit less than the total time spent over time building out SQL reporting queries.

2. Cubes are More Complex than SQL.
    a. Cubes are more than just SQL.
    b. And developers have to learn some new skills in order to build them.

3. Cubes are a Bit More Expensive.

a. Cube require the Developer, Business Intelligence, or Enterprise edition of SQL Server.
b. This costs more money than the Standard edition which is used for most SQL reports.
c. Development costs can also be a bit more than for SQL based reporting.

4. Some Training is Required to Develop and then to Use Cubes.
    a. Your developers will need cube training.
        i. It is important to get real world cube training instead of book training.
        ii. You can't replace real world experience with how to's from books or the Internet.
    b. Report developers will need only a little training if they are already using Excel pivot tables or another cube compatible tool such as Tableau.
    c. Reporting users won't need much training unless they have to be trained in Excel pivot tables or a new reporting tool for the cubes.

3.4 Why do We Need Tabular Cubes?

This is a little redundant comparatively because we've gone over this previously, but we'll briefly recap it here because I wanted to cover specific question variations in a "just look up the question format" rather than "read through the entire book to find your answer format".

Companies need and use Tabular Cubes to support their data reporting needs. But the big reasons why they need Tabular Cubes is because SQL reporting is slow and can take hours. And sometimes all night for batch reporting. Tabular Cubes can do aggregations of data much, much faster than SQL. What typically happens is that batch SQL reports take a longer and longer time to run. Eventually they take longer than the entire night, way past the point where people need them which is typically first thing in the morning. Throwing more server hardware at this is a stop gap measure and sooner or later you're going to run out of server upgrades or money for it. This is where Tabular Cubes come in. Cubes were first developed years ago due to this very reason.

Cubes can do SQL aggregations faster and better. And when I talk about faster I mean they can typically and comparatively do all of their aggregations in minutes where SQL takes hours.

The second reason is why companies need cubes is because they want to give more power to their users and to report developers. With cubes you can actually do slice and dice type ad hoc reports, also known as Self-Serve reporting, where the cubes support all sorts of variations as well as individual custom reports. It would take hundreds or perhaps a couple thousand of reports and report variations to be able to produce all the reports that one cube and a cube browser or Excel pivot table can instantly create.

Cubes basically blow away SQL reporting and make reporting much easier, faster, better, and more powerful than any SQL approach. These are the two biggest reasons why companies need OLAP cubes.

If you want faster reporting and better reporting cubes are your answer.

3.5 Why are We Using Tabular Cubes? - Conclusion

Why are we using cubes?

For all the advantages and reasons stated above. That's why!

Cubes are just better than plain old SQL reporting:

1. Cubes gets Your Reporting Unstuck
    - 1.1 Provide One Version of the Truth
    - 1.2 Provide One Central Reporting Data Source
    - 1.3 Eliminates Nightly SQL Batch Reports
    - 1.4 Let You Do Things You Cannot Effectively Do With SQL.

2. Cubes are Faster than plain old SQL.
    - 2.1 Faster than regular aggregation SQL to run.
    - 2.2 Faster to develop new reports.
    - 2.3 Faster for users to get/create new reports.
    - 2.4 Faster and more efficient reporting organization.

3. Cubes Provide New Dimensions to Reporting.
    - 3.1 Complete Data Accessibility
    - 3.2 Full Data Browsing and Exploration
    - 3.3 New Data Insights
    - 3.4 DIY Self-Serve Reporting

The Bottom Line is Simply:

> Cubes are Much Better at Aggregation Type Reports than SQL.

So, if you use totals in your business or organization, and I'm sure the you do, you are going to want to use cubes.

Chapter 4. Why Do I Want and Need to Use Tabular Cubes?

If you do reporting as part of your job responsibilities, then you definitely need to look at developing and using cubes. Even if you just use (consume) reports and don't develop them cube reporting will be very superior to SQL reporting any time aggregation type reports are needed. And in virtually all business organizations they very needed and extensively used.

For example, let's say the you are doing a regional sales report. You need to see how all the store locations in your region are doing. So, what do you do? Do you run a SQL report and wait for it to finish? Then run a couple of other reports so you can see the top performers and the poor performers? Then you want to run a report that summarizes the sales for the top 20 performers across the state and city averages...Oops! You can't run that report because you've been waiting for IT to develop it. It's on their list but there's a large backlog and you've already waited over 6 months for that one report! Looks like you're going to have one of your people divert from productive work to cut and paste the information from the individual reports into an Excel spreadsheet and crunch the numbers and generate a result table from that...

No, you don't! Not if you have an OLAP cube! Just take all that effort that's being put into SQL report development and spend it instead in developing a cube! You'll be very glad you did!

Fast forward to now having a cube...

Hmmm...You want to compare the top and bottom performers...You just open up an Excel Pivot Table or Tableau or another reporting tool and connect to your shiny brand new cube and with a few drags and drops you've sliced and diced your way to the answer.

Hmmm...Now what if I want to see average sales for the last 7 days...Then the last 30 days...Then YTD (Year To Date)....

Simple just drag and drop the appropriate measures and dimensions in your reporting tool of choice and you will quickly see that answers!

Now, reflect on the "bad old reporting days"...How long would it have taken individual SQL reports to be developed to do all this? And how must time would have been wasted?

When you want and need answers OLAP cubes can give you instant answers! And you can slice and dice to your own customized self-serve reports as much as you want. And you can browse data easily the same way and explore and discover new insights about the data. SQL reporting can't do that!

Your job will be faster and easier richer and better with it and that's the bottom line.

4.1 How and When Do I Know that I need Tabular Cubes?

This is a very easy question!

If you develop or use reports and your organization rely on reports that aggregate (sum) numbers, then most likely you need Tabular cubes!

While they take a while to build, cubes will provide you and your company with many benefits.

And, if your SQL reports are using aggregations (summing up numbers and such) are taking a long, long time to run and are overloading your server and you can't get the reports done on time...

That's when you really know that you need Tabular Cubes!

The Cube Test!

Let's do...The Cube Test!

Answer the following questions yes or no:

1. Do your report users continuously request more and new reports?
2. Is there a delay between wanting a report and getting one?
3. Is this delay significant?
4. And does this delay negatively impact productivity?
5. Does it take a long time for your report developers to create new reports?
6. Would your report users like to be able to instantly create new reports by themselves...But cannot yet do this DIY or self-serve reports?
7. Do your nightly batch SQL reports take a long time, hours, to run?
8. Do they all get done by morning when the reports are due and needed?
9. Do you have any problems with the accuracy of the numbers in your reports?
10. Have you ever encountered and/or suffered from inaccurate reports caused by SQL problems?

Scoring:

Give yourself 10 points for every "Yes" answer and 0 points for every "No" answer.

0 – 20 Points:	Looks like you're doing quite good with reporting! Already have cubes, eh?
30 – 50 Points:	You're getting into the situation where you're going need cubes.
60 – 80 Points:	You're definitely into the situation where you need cubes.
90 – 100 Points:	Wow! You really need cubes! Before you need to start looking for a new job!

4.2 When Do I Don't Need Tabular Cubes?

There's probably only a few situations where you don't need to use Tabular cubes:

1. You're not really doing any reporting and
2. You're not doing data analysis

Then you don't really need Tabular Cubes because you're not using any reporting at all.

If you have a smaller number of reports and they're wholly adequate and running fine, then that's the other situation again where you probably don't probably need Tabular Cubes.

These are really the only two times that I can really think of when you really don't want or need tablet cubes.

I will also point out that there are times when you don't think that you need Tabular Cubes...But you really do.

The Jira Cube Case Study

A few years ago, I was using a problem tracking system called that Jira for a cube project with a large company. Jira is still around, and it is still popular today. The problem I had at the time was that reporting on it was just inadequate. From my perspective I wanted and needed to see what my reporting and cube and ETL development team was doing. How many hours they were taking on things they were working on. And what were they working on. And what subsets and areas and tasks of the projects? How often were bugs being fixed versus new development? And so instead of having to go through this report and that report and try to look at another report and try to piece it all together I did something very simple. One evening I exported a snapshot of the data from Jira and in about an hour I had myself my own Jira Cube. I then wrote some SQL to automatically do the export directly from the Jira database tables. I then set up a SQL Server Agent scheduled job to do that every day for a couple of times during the business day and then automatically process the Jira Cube.

So, throughout the day I had an updated cube that I could go in and quickly and easily see who was doing what where with what are and what type of work and the priority of the work and the hours of being charged and where the time was being spent. In other words, I could instantly slice and dice the data and find out essentially everything about that data and my team's work. For me this was very simple to set up with Multidimensional Cube and a simple OLAP cube browser. It was very simple for me to do because I had the skills as I've been building cubes for years. And they are now very easy and quick for me to build. Your typical Jira user might say that Jira has reports...But when I would show him/her exactly what I could do with my little Jira Cube and how fast and easy it was to get all the answers then they would change their minds. With that little cube (small in size but BIG in POWER!) I was able to answer all of

management's many questions instantly! I didn't need any time to run reports and look through them and match up and corelate the information in either my mind or an Excel spreadsheet.

"Here's the Answer right here..." is a much better answer than, "Uh, let me run a few reports and do some Excel work and I'll have that answer for you in a couple of hours or tomorrow..."

Cubes will surprise you! Very pleasantly!

4.3 Do Tabular Cubes Replace SQL Reports?

While Tabular Cubes will replace of a lot of SQL reports and individual SQL report queries, they do not fully replace them. Why? Because cubes are meant to report on numbers, specifically the aggregation of numbers. They are not meant for reports that are primarily text based or reports that do not specifically involve numbers.

For example, let's say that you need to look at city, state, regional, national, and international sales. Want you want to see are the high level aggregated (summed) sales numbers for the year and the last quarters at each of those levels. This is an ideal use for cubes. And one where they easily outperform and out distance using plain old SQL for this.

As another example, consider now that using your Tabular Cube that you've determined which city location has the highest sales per capita for their metropolitan statistical area. Let's say that it's Madison, Wisconsin for this year. And you want to send an email and physical letter to all the team at the Madison store thanking them for making it to number one in sales per capita and giving them a gift to show your and the company's appreciation. To do that, you need a list of all the team members of the Madison, Wisconsin store, their email addresses, and their mailing addresses. While you will find their names and some other details in your Tabular Cube you probably will not find their email or mailing addresses in the cube. Why? Simply because this information doesn't really belong in the cube. It's not a number and doesn't directly connect with the numbers in the cube, like city and state and store number and such do. You don't really filter, sort, group, or slice and dice on unique email and mailing addresses. States and zip codes, yes. Typically, you would not be interested in mailing address information in a cube so this type of information, being textual, would not be put into a cube. It would only increase the size of the cube and reduce its query performance so why have it in the cube?

So, to get this information you would run a SQL type report query, a detail report, to find the details of the mailing addresses for the winning team members. This gives you a good example where cubes can be used and where SQL will still need to be used.

Cubes don't really replace all SQL reports. Just the hard ones that do high level aggregations with numbers. Details for those numbers and what they connect with on a highly detailed level are still best done with conventional SQL reports. So, there will still be a need for SQL reports in the enterprise's reporting and Business Intelligence systems and databases.

Cube complement and extend and replace a lot of SQL reports. But not all of them.

4.4 How Are Tabular Cubes Better than Regular SQL Reports?

Tabular cubes are better than regular SQL reports in these very important ways:

1. A Central Data Source for Reporting
2. One Version of the Truth
3. Faster Than SQL.
4. Better than SQL.
5. Able to do Slice and Dice Ad Hoc Self-Serve Reporting.

Let's take a quick look at each one.

A Central Data Source for Reporting

A common problem with SQL reports is that report developers hand code SQL queries for each report. This not only takes time but also is a vast duplication of effort. And requires maintenance effort as well. Cubes can act as a central master data source for a vast majority of reporting query needs. And especially the really hard reporting query needs that involve aggregations and multi-table joins. Using cubes as the primary report data source will save a lot of time for report developers.

It will also allow "One Version of the Truth".

One Version of the Truth

The phrase "One Version of the Truth" means that all the reports will all have the same numbers. And your reports will not report different, sometimes very significantly different, numbers. Imagine being in an important meeting where you are presenting some results when another department or division head interrupts you to say that his/her numbers are significantly different. And another person pipes up that his/her numbers are different from both of your numbers! Not a good situation! This has happened a lot in many companies. It stems from the use of different SQL queries being used for the same reports or near same reports. People will always write SQL in a multitude of different ways and this can result is different results. Often times SQL JOINS are done slightly differently will have rather different results and it is not always obvious to the report developers or managers or executives. But it is always very embarrassing and problematical when such a thing surfaces. And either the SQL query was wrong, different from what it should have been, or even that the underlaying business rules were wrong or interpreted wrongly.

The solution is called "One Version of the Truth". You get that from a formal data warehouse traditionally. But you really get it from using cubes and using cubes as your report data sources wherever possible.

Faster Than SQL

Real world experiences show that report developers can generate new reports about ten times (10x) faster using a cube as a data source compared to hand coding new SQL queries. See the ROI section on the Executives and Managers version of this book for more information on this.

Cube queries are also always faster than SQL aggregation queries. And prevent and eliminate long running SQL queries from interfering with report generation and use and loading servers drastically as well.

Cubes are faster than SQL for aggregations and complex report calculations.

Better than SQL

Cubes are also better than SQL for complex reports that use aggregation calculations such as counts, minimums, maximums, averages, and other complex calculations. The calculations are faster, more reliable, centralized, can be implemented faster and easier compared with SQL.

Cubes are betting then SQL for aggregations and complex report calculations.

Able to do Slice and Dice Ad Hoc Self-Serve Reporting

Cubes allow for users and report developers to run their own ad hoc self-serve reports via cube browser tools and Microsoft Excel pivot tables. Users can do “slice and dice” reporting by using dimensions and filters to see customized cross sections and intersections of data. No more long waits for SQL reports to be developed. No more having to wait for variations and new reports to be developed. No more having to use a multitude of SQL reports to try and build up an understanding of the data. Instead, cubes allow users and report developers to browse and explore the data seeing instant results on a vast multitude of custom reporting combinations. The difference is like finally turning the lights on in a very dark big room compared to using a dim penlight to try to understand what’s in that room. A very big difference!

4.5 What Type of Situations are Tabular Cubes Best Used In?

We've discussed this in other sections, but we'll summarize again here to cover this question.

Tabular Cubes are best used in situations where:

1. SQL report queries are taking too long or not completing on time,
2. Your users want/need to use cubes for slice and dice ad hoc self-serve reporting.

And

3. Your data that you want to cube is primarily numeric.
4. Your data that you want to cube is not huge.

Then you're in a situation where Tabular Cubes can help your significantly.

Additional factors to take into consideration are:

1. Do you have a developer(s) that can build Tabular Cubes?
2. Or can be trained to do so?
3. Is there time and budget to build the cubes?
4. And deploy them and give the users and report writers a bit of training to get them started?

If the answers here are "Yes!" then you are all set! Get started building and using Tabular Cubes!

4.6 Can We Cube Any Data?

Technically yes, but practically no. Cubes are meant for numeric data. And numeric data that is aggregated, i.e. summed up from a low level to a high level. That's why they were developed and that's what they excel at. As we said in section 4.3, "Do Tabular Cubes Replace SQL Reports?", cubes replace SQL reports that do aggregations but not SQL reports that do detail type reporting. SQL will always be used for that. And for reports that deal with a lot of text and text based information. Cubes are for numbers.

Numberless Count Cubes

Now there is somewhat of an exception for this – The Factless Fact Table and Cube. A "Factless Fact Table" is a data table that has no real numbers in it. And it produces a cube that has no "facts", i.e. no numbers. But it still can be very useful as a cube. I call such a cube a "Numberless Cube" or a "Counts Cube" because while there are no real numbers in the cube the cube can still be useful just by doing counts at different levels.

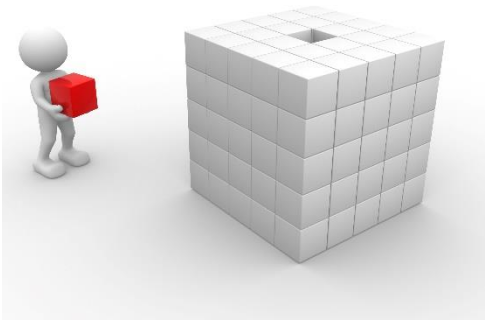
For example, image a cube that's going to count mail address corrections. The cube has mailing addresses for all store employees and customers as well. There's no numbers in the cube just a flag column that indicates that the address has been verified and if needed corrected. So, we have two non-numeric facts in the cube – Address Verified and Address Corrected. In this cube we simply want to do a count of both types. Our cube will then be able to show us the total number of address (i.e. records present), and the total numbers of Addresses Verified and Addresses Corrected for both employees and customers. Thus, we have a total of the following count type measures in our Addresses Cube:

1. Count of all records.
2. Count of all records - For employees.
3. Count of all records - For customers.
4. Count of all Address Verified records.
5. Count of all Address Verified records – For employees.
6. Count of all Address Verified records – For customers.
7. Count of all Address Corrected records.
8. Count of all Address Corrected records – For employees.
9. Count of all Address Corrected records – For customers.

We have nine measures in the cube in spite of having no real numbers in the cube. No sales numbers, number of orders, etc. Typically, this doesn't make a really good individual cube. Counts like this are most often found in another cube with more data and lots of numbers. These counts on addresses would be not a primary purpose of the cube or primary measures but more typically a secondary feature and secondary measures. But there is still usefulness in cubing this data. This type of cube or cubed information can be easily used to track status and results for programs such as our national address correction example. Let's say that this company just started using an address verification and correction service such as Smarty Streets or Melissa Data or another USPS (United States Post Office) approved address correction service. And this company has a large backlog of address verifications and corrections to do. By including this information into an existing cube and updating it as blocks of addresses are verified and corrected, so many each night for example, then it would be very, very easy to see with a cube exactly where via city, state, region, and nationally verifications and corrections were made and are pending. A useful application of cubes to where numbers are only counts!

So, cubes can do a lot of things very usefully in a lot of ways. They can even be used to count things at many levels!

4.7 Can We Cube All the Data in our Organization into Cubes?



Technically yes, but practically no. Cubes are meant for numeric data. And a lot of data in an organization is not numeric and thus generally does not belong in a cube except for counts.

Of that numeric data in an organization, there are typically a great deal if not a majority of the data being numeric/numbers. So, yes, all that data could potentially benefit from being put into a cube. The decision to which numeric data to cube is often made by priority and the importance of one grouping of data over another. Some data just isn't used that often or is as important, so it may not be worth the effort and expense of being cubed.

Data also has to be organized into groups by type and uses and then put into individual cubes. This can take a bit of time. It's important also to realize that multiple cubes will most often be needed due to the different types of data. And the type of use for the particular cube.

There are different types of cubes that can be built:

1. Reporting Cubes
2. Counting Cubes
3. Dashboard Cubes
4. Data Analysis Cubes
5. The Super Cube

Reporting Cubes

Reporting cubes are used for general reporting purposes. They are most common. And Tabular Cubes do them quite well.

Counting Cubes

Counting Cubes, also called Factless Cubes, simply count the number of records by level and a few other dimensions. They are very simple cubes and Tabular Cubes work every well for them.

Dashboard Cubes

Dashboard Cubes are cubes that are specifically built to power a dashboard. The reason to do this is performance. Dashboards need extremely quick query responses. They also have multiple tables and charts on a single screen or web page which means multiple queries. This multiplies the need for very fast queries and while a multipurpose cube can provide this problem will often occur when sharing the cube between dashboard users and general users. Every once in a while, a general user will submit an ad hoc cube query that uses a lot of resources and takes some to run. This may not seem long to that user, say 15 to 20 seconds, but when you have a group of dashboard users, typically managers and executives, who see their dashboard update hang for 20 seconds or more, the dashboards users will start to complain about the poor response time. Dashboard users expect very fast response times and using a general purpose and user cube will often cause problems with that. So, I recommend only using a dedicated dashboard cube with dashboards for maximum performance and dashboard user happiness.

Tabular cubes are appropriate for dashboards of simple to moderate complexity. Complex analysis type dashboards should use Multidimensional cubes for their additional features that are useful to data analysis.

Data Analysis Cubes

Multidimensional cubes can also be used for data analysis cubes. I haven't seen these cubes discussed or mentioned in books or online articles, but I've developed and built several of them. They are primarily used by and meant for use by data analysts for complex analysis of data instead of simple data reporting. You will want to use Multidimensional cubes for this instead of Tabular cubes. They are a great way to do detailed and comparison data analysis. The data analysts that I've build them for have always been very happy and pleased with them. And this was years before Data Science became a big thing.

Essentially, a data analysis cube is a complex cube that uses and blends multiple fact tables together often with external data such as geographic or demographic statistical data to provide the ability to generate complex analytic reports and comparisons based upon multiple sources of data. They are much more complex than reporting cubes and are specifically built and used for data analysis rather than reporting.

While typical Data Science approaches can be used, an analysis cube provides a programming free, drag and drop analysis solution. They take a bit longer to build than simple reporting cubes, but they become a very welcomed and well used tool for data analysts. All without having to develop custom software in the R or Python languages. This is a very significant application of cubes which conventional data science tools cannot match. (I should write a book about this...)

The Super Cubes

Every once in a while, I come across someone voicing the idea for a Super Cube. Or telling the story of when their company tried to build one. And failed. The basic idea of a Super Cube may sound good but it's not a good idea from a performance and size standpoint. Super Cubes essentially require Multidimensional Cubes because Tabular cubes don't have the features and capacity to handle Super Cubes. What exactly is a Super Cube? A Super Cube is the idea of taking ALL or MOST of the data in a company or organization's databases and putting it ALL into ONE BIG CUBE! While it may seem like a good idea at some time it is not a practical or a good idea. This is because the Super Cube will be so big and have so many users and so much data that it's performance will be horrible! Think about a report query or a drag and drop operation in a pivot table or cube browser that takes 45 MINUTES to complete! This is the actual amount of time for one Super Cube project that I heard about. No, it wasn't my cube. I was interviewing a self-proclaimed cube expert who tried to impress me with how big and wonderful his Super Cube was. After repeated prodding by me, he finally admitted to an average 45 minute response time. Some "expert"! Obviously, I did not hire him.

So, think of many purpose built and dedicated cubes fulfilling your corporate needs. And going Multidimensional if you need eventually to connect them all together and also need analysis type cubes. Else just use Tabular cubes. And always smile knowingly when someone brings up the idea of a Super Cube.

4.8 Can I Use My Favorite Report Tool with Tabular Cubes?

Yes! Most if not all reporting tools support both Tabular and Multidimensional OLAP cubes.

Here's a list of the Microsoft BI tools that support both:

1. Excel pivot tables and charts.
2. Power BI and Power BI Embedded.
3. SSRS – SQL Server Reporting Services:
    a. All report types
    b. Report Builder,
    c. Report Designer.
4. Power View.

Additionally, virtually all front end reporting, Dashboard, and visualization product support Microsoft Tabular and Multidimensional OLAP cubes including:

1. Tableau
2. Dundas BI
3. Domo

4. Diver | BI Software
5. Databox
6. icCube
7. SAP Business Objects
8. Qlik Sense
9. IBM Cognos
10. Panorama Necto
11. Phocas
12. SAS
13. Tibco
14. Microstrategy
15. KYUBIT
16. inetSoft
17. Yellowfin Software
18. Pentaho
19. BIRT
20. JasperSoft
21. ClicData

Note that some of these tools may have some limitations when used with Microsoft OLAP cubes. Please see their product pages and technical support for specific information. And note that this not an exhaustive list, just an illustrative list. And that some of these tools are also open source. Lots of reporting tools support OLAP cubes, both Multidimensional and Tabular.

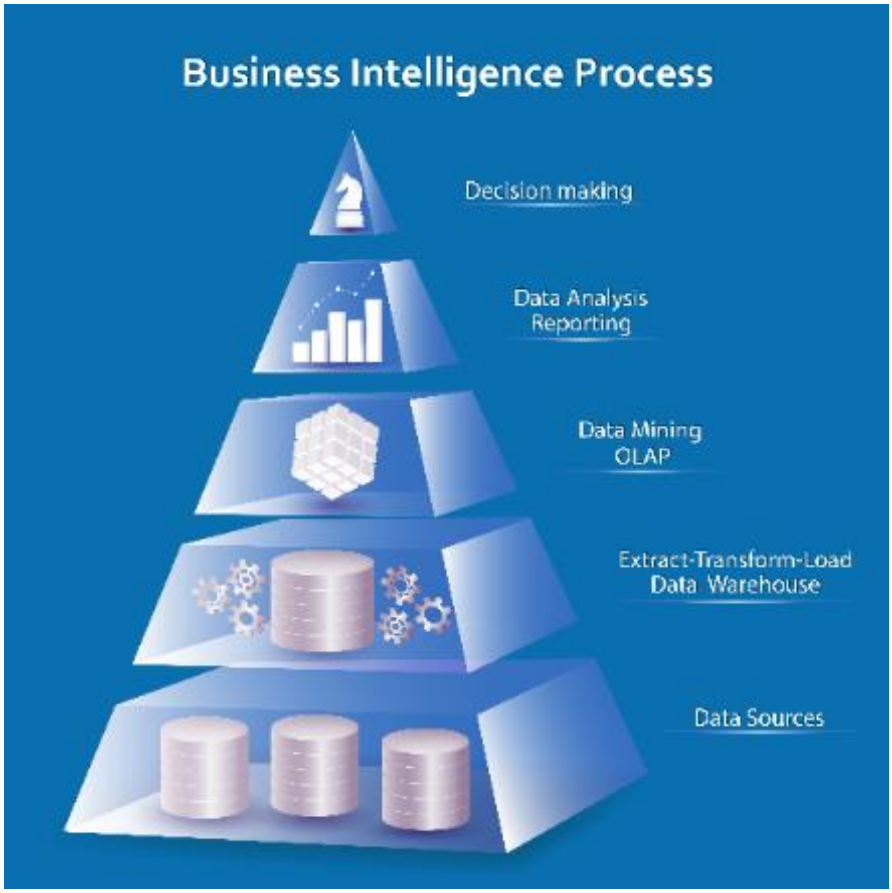
Chapter 5. How Do Tabular Cubes Work? At a High Level?

Cubes may seem mysterious at first. But once you have a basic understanding about how they work they are much easier to understand. And that which you can understand...You can use!

5.1 The Cube BI Big Picture

Cubes are part of what's called Business Intelligence or BI. BI is essentially a reporting system that functions to provide the business people with the data and data analysis and insights condensed into a report form. Cubes greatly improve this by providing greater and easier reporting to be done on data that has multiple levels and needs to be aggregated for reporting purposes.

The Cube Business Intelligence (BI) Big Picture looks like this:



The Data Source Level

In this diagram we see at the bottom, at the base a number of data sources. These are the company organizational SQL relational databases that provide the source data for the cube.

The Data Warehouse ETL and Reporting Databases Level

In the next level we see that Data Warehouse that contains the ETL level of the reporting system. A set of processes known as ETL for Extract-Transform-and Load gets the data from the data sources and puts into all into a form suitable for cubing. The source data is transformed into both a dimensional cube schema form as well as being transformed at the database column level as needed and necessary. The ETL then loads the data after transformation into the dimensional cube schema.

The Data Warehouse Cube Level

The next level of the Data Warehouse is that of the OLAP cube. An OLAP server, this case Microsoft Analysis Services, also known as SSAS, builds a cube from the dimensional cube schema and the data loaded into it that can be accessed by cube browsers, reports, Excel pivot tables and other front end cube capable reporting tools.

The User Reporting and Data Analysis Level

The next level is the actual data analysis and reporting. It is in this level that the front end reporting tools live and report users use them to access and analyze and understand the cube data.

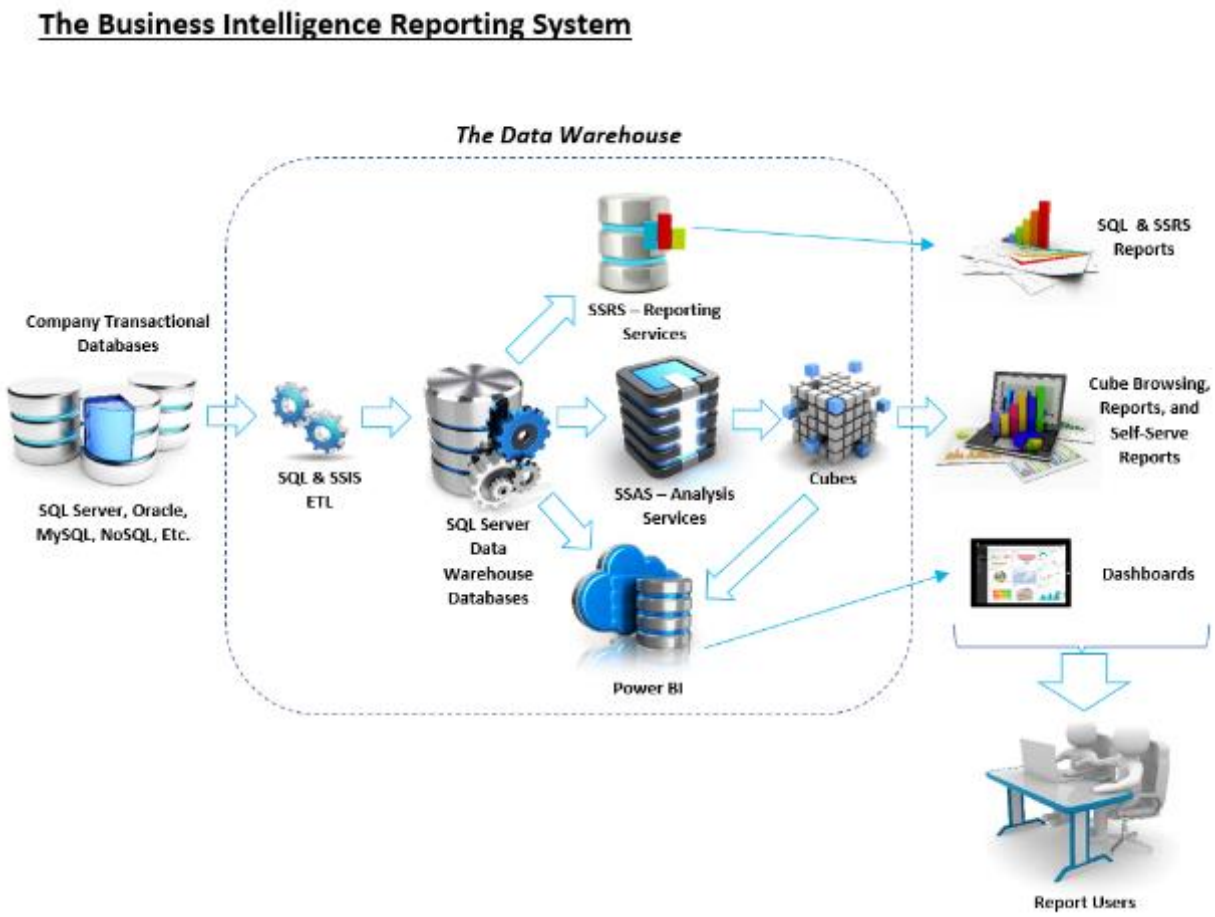
The Final Level – Decision Making

The final level is that of decision making. In this level, people act on information and data in the reports from the previous level. This is the topmost and the most important level. All of the levels underneath this level are there to support this topmost level. All of the data and operations in the reporting system is for supporting the topmost level of decision making based upon

All these levels compose the Business Intelligence Reporting System. We describe and talk about the specific data warehouse levels in more depth in the following section.

5.2 The BI Reporting System Big Picture Diagram

Exactly how a BI Reporting System works can be seen in this diagram:



A BI Reporting System consists of many components:

1. The Data Sources
2. The SQL and SSIS ETL Components
3. The SQL Server Data Warehouse Databases Components
4. The SSRS and SQL Reports Components
5. The SSRS and SQL Reports
6. The SSAS Analysis Services Cube Server Component
7. The Nightly Data Flow
8. The Cubes Component
9. The Cube Browsing and Reports
10. The Power BI Component
11. The Dashboards
12. The Report Users

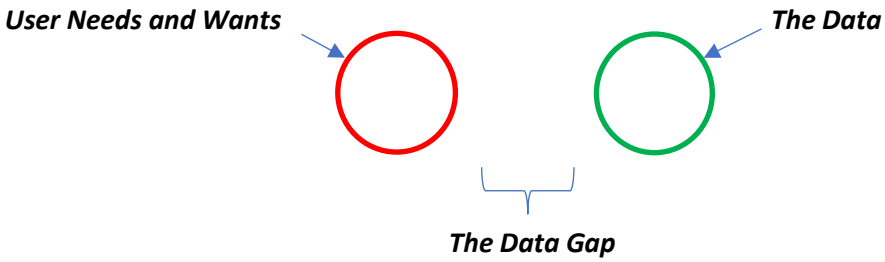
Each of these components work together to form a complete reporting system. Many people don't think of this as a system but rather as just a bunch of reports because that's what they see of it. That is the visible tip of the iceberg. As with an iceberg, 80% of it is "underwater" and not readily visible. This "invisible infrastructure" is critical to supporting the cubes and all of reporting. It needs to be done designed and implemented properly so as to function efficiently, reliability, and with high performance. To get the right data to the users in a form that they can easily and readily use.

The Successful Cube – Grant's Theory of Cube Data Unity

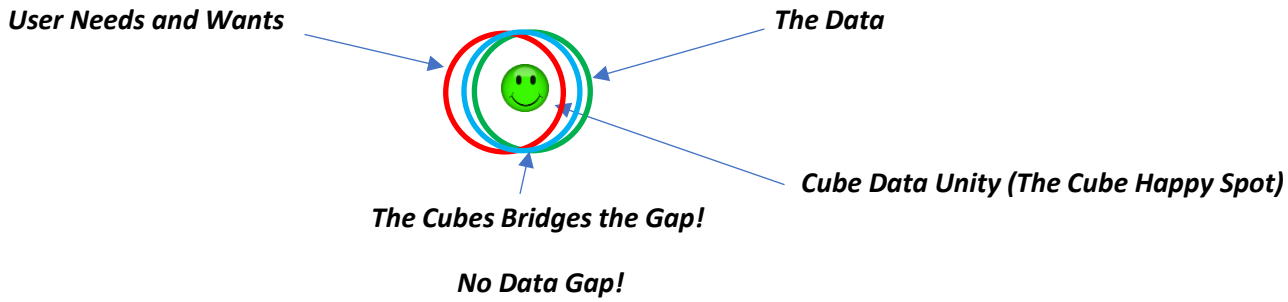
The successful cube is a combination of:

1. User Needs and Wants
2. The Data
3. And the Proper Application of Cube Technology to bring them both together

The situation before the cube looks like this:



A cube bridges that data gap:



A properly and well-built cube will bridge the gap between your data needs and the data available. The cube bridges the Data Gap. A good cube will have all of the data that you need and thus has what's called a Cube Data Unity Factor of 100%. Most cubes need a bit of refinement after their first production release to get there. You as a cube user will probably want and need some specialized calculations added as new cube measures which become apparent as you use the cube and get more familiar with it. This is a natural and normal process. A cube should have 90% percent of everything a user needs at its first production release. Additional refinement of the cube should fairly quickly bring that to almost 100%. As cube use and reporting increases in volume and with new additional people, some additional requirements will be created, and the cube can then be refined to meet these requirements. This happens naturally over time and use.

A high cube Unity Factor and fast responding cube refinement makes for a very successful cube and very happy cube users!

The Data Sources – Company Transactional Databases

The data sources for a reporting system and a cube are the company databases that hold records of the transactions and data for the company or organization. These databases can hold quite a variety of data – Customer orders, product and services data, financial data such as payroll and expense payments, testing results, machine generated information such as cellular system performance and status, and so forth. Any data that is in an organization and is useful for reporting can be imported into a reporting system or cube. Most of this data comes from traditional and convention SQL relational database systems such as Microsoft's SQL Server, Oracle, MySQL, and others. This data may also come from Cloud

based services and data sources such as Microsoft Azure, Amazon AWS, Salesforce, SAP, Workday, ADP, and others. This data may also come from flat files and spreadsheet files. Also, from databases which are new and different such as NoSQL databases including Microsoft Cosmos DB, Mongo DB, MarkLogic, and others. Cubes can use data from virtually any source.

Any data source that has structured data that needs to be reported on can be imported into a Data Warehouse. And if the data is primarily or significantly numeric, or has a count component to it, can be put into OLAP cubes either Tabular or Multidimensional.

The SQL and SSIS ETL Components

Getting the data from the source databases and files typically involves the use of SQL query and insert statements when dealing with relational database sources (most common), a specialized ETL product such as Microsoft's SSIS (SQL Server Integration Services), or customized ETL programs or software.

ETL stands for "Extract, Transform, and Load". And that says exactly what it does:

1. Extract the data.
2. Transform the data as necessary.
3. Load the data into a star or snowflake schema and then into a cube.

ETL is a generic term for whatever database components and/or software programs perform this action. The data is first extracted from the source data sources. It is often transformed in one of two ways or both:

1. The data type or format is converted to another more conducive to reporting, and/or
2. The form of the data is then converted into either a Star Schema or a Snowflake Schema for reporting and for cubing.

Both the star and snowflake schemas are using for both data warehouse reporting as well as cubes. Tabular cubes use just star schemas while Multidimensional cubes can use both star and snowflake schemas. The transformed data is then loaded into one or the other of these schemas which makes the data ready for cubing.

Most typically, the data sources are conventional relational databases systems. SQL is the most common database language is typically used for ETL processes. A Microsoft ETL tool, called SQL Server Integration Services or SSIS, comes with all versions (editions) of SQL Server is often used for ETL. It can simply

encapsulate the SQL commands that compose the ETL processes or it can also be used like a data programming language to visually compose data transformation and loading processes.

Other types of ETL tools and programs can also be used. I've built my own custom ETL process programs in Microsoft C# .NET for extracting data from Microsoft's Cosmos DB, a NoSQL database partially derived from Mongo DB. Wherever the data resides, it can be accessed and ETL'ed and used in cubes.

The SQL Server Data Warehouse Databases Components

With the Microsoft BI stack components and tools everything revolves around SQL Server the flagship database server product from Microsoft. All the main BI and cube tools, SSAS, SSIS, SSRS, and SSDT, are components of the SQL Server database product. They are included with SQL Server and some of them have reduced or full feature sets depending upon the specific edition of SQL Server. Most data warehouses, which hold reporting databases include cube schema databases, use the Enterprise version of SQL Server which contains all of all the available components and features.

Most often, one or more databases in a SQL Server installation will be used as a data warehouse for cubes and will often contain multiple star or snowflake schemas for multiple cubes.

The SSRS and SQL Reports Components

Reporting in a Microsoft BI reporting system is often of two types:

1. SQL Reports and data sources, and
2. SSRS.

SSRS is SQL Server Reporting Services and it include a complete report development and deployment system including a web based report portal and scheduling system.

SQL reports are done through various front end tools which use SQL report queries as data sources.

Additionally, some form of SQL batch reporting is common where preset SQL report queries are run nightly to generate reports that are save in report tables in SQL Server. The advantage of these reports is that they can be queried simply and quickly by front end reporting tools because they already hold the report data in the final form needed. So, they are like pre-run and pre-calculated reporting queries that both simplify and speed up reporting. Batch SQL reporting is often

done using stored procedures which are stored SQL scripts in SQL Server. Additionally, the SQL Agent service of SQL Server is often used to schedule SQL batch reporting jobs. It is also used by SSRS for scheduled reports and also for scheduling SSIS jobs including ETL and Cube processing.

The SSRS and SQL Reports

The SSRS reports and SQL reports are used in many ways for reporting, but both rely on SQL server reporting queries that act as data sources. Other front end reporting tools use SQL for report queries.

Note that SSRS can use MDX and DAX queries to query both Multidimensional and Tabular OLAP cubes.

The SSAS Analysis Services Cube Server Component

SSAS, SQL Server Analysis Services, is the server component that hosts and contains OLAP cubes, both Tabular and Multidimensional. Note that a single SSAS server instance can only host one of either type, Tabular or Multidimensional, at one time. Not both at the same time. Multiple instances of SSAS can be run the same server or server instance by using what's called a "named instance" and a different non-default port for cube connections. SQL Server is required for the SSAS service to run and operate properly.

In operation, the SSAS service hosts the cube definitions in a series of cube databases, one for each cube. A very large number of cubes can be hosted, and performance is limited to the power and size of the server or server instance (primarily RAM and number of CPU cores).

Cubes are developed on development machines that typically have a developer edition of SQL Server and thus SSAS installed on them. They are then deployed to the cube server. To load data into a cube an operation called "processing" is required. There are various types or classes of processing but the most common is called "Full Processing" and everyone will just use the default term "processing the cube" when they talk about this. Once the cube is loaded with data and processed (essentially one and the same operation) the cube is then online and available for users to connect to and use. Note that when a cube is processed the ETL for the cube is also run as it's the ETL process that pulls the updated data from the data sources in the cube schema and database. From there the cube processing will automatically read (query) the cube tables and load data from them. Most cubes are processed once a day at night. But cubes can be processed (refreshed with data) multiple times per day.

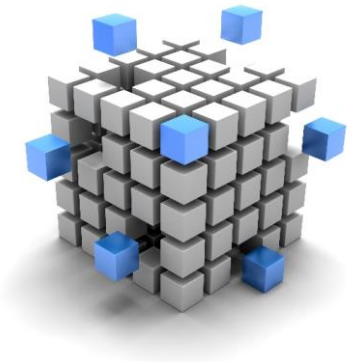
The Nightly Data Flow

Not separately shown in this diagram is a very important part of this process – When the data flows through the system. Generally, most cubes are processed only one a day, usually at night so the data is fresh in the cube for use first thing in the morning. Processing loads the data into the cube and then does the internal calculations of the cube. This can take a few minutes for very small cubes to several hours for large cubes. Speed of processing depends upon the amount of data and the power of the server.

Cubes can be processed multiple times per day or even every hour or more often. Typically, these are small to medium sized cubes that can be processed in a few minutes and contain data that is very time sensitive. For these type of cubes, often a pre-arranged schedule is arranged with the users so that they know what times during the weekdays that the cube will be refreshed with data. One such schedule could be nightly at 4:00 AM so the updated cube data is available at 6:00 AM for workers. And then the cube is processed again at Noon for late morning/early afternoon updates. When cubes are processed once an hour they are often processed at the top of the hour and the users then know that by a certain time afterwards, say 15 minutes, that the cube has fresh data in it.

Note that a cube web portal can be used to display for the users the cube status including last processed time. This is highly useful and automates cube support as well.

The Cubes Component



Cubes are hosted with SSAS (SQL Server Analysis Services) and once processed will hold the latest data. Many cubes can be hosted and can of course be big or small or somewhere in between.

The Cube Browsing and Reports

The front end cube browsing, and reporting tools are an integral part of the reporting system. They connect directly with the SSAS server and pass cube queries to it using a protocol called XML/A which means XML for Analysis Services. Cube queries use either of the two cube calculation programming languages: DAX and also MDX for Tabular cubes and MDX for Multidimensional cubes.

The Power BI Component

Power BI is primarily a cloud based Microsoft dashboard and reporting service for both SQL and cube reporting and dashboards. It can also be installed on premises/on site but most of the time it will be hosted in the cloud and use localized data connectors to pass data up to it. Power BI is very powerful and popular front end reporting component and it is free for individual users at its lowest pricing tier.

The Dashboards

Dashboards typically use SQL and sometimes cube report queries with Power BI or some other dashboard tool or product. Note that since dashboards typically have multiple queries per page, one for each of the grid, table, chart, or map display panels, they need very fast query processing times for adequate performance for most users. From a cube perspective, I recommend a dedicated dashboard cube for this, for maximum performance and to eliminate random latency delays caused by varying user loads on regular reporting cubes.

The Report Users

The report users are the topmost part of a reporting system and the most important. The purpose of the reporting system is to meet and fulfill the data needs of report users. This not only includes access to the data but also fast performance and ease of use. Cubes should be built with best practice naming standards and conventions for maximum cube user intuitive understanding and also to avoid ambiguity.

5.3 How Cubes Work – The Spreadsheet Analogy

To simply understand OLAP cubes let's first think about a spreadsheet. Most people have used a spreadsheet, such as Excel, and some people use them regularly in their work. Spreadsheets first commonly appeared in the workplace in the late 1970's and early 1980's with software products VisiCalc and Lotus 1-2-3. The idea of a spreadsheet is simple: It's just a set of rows and columns forming a grid where you place numbers and the name of things that the numbers connect to.

A Simple Spreadsheet:

Columns - Years →

Rows - Countries ↓

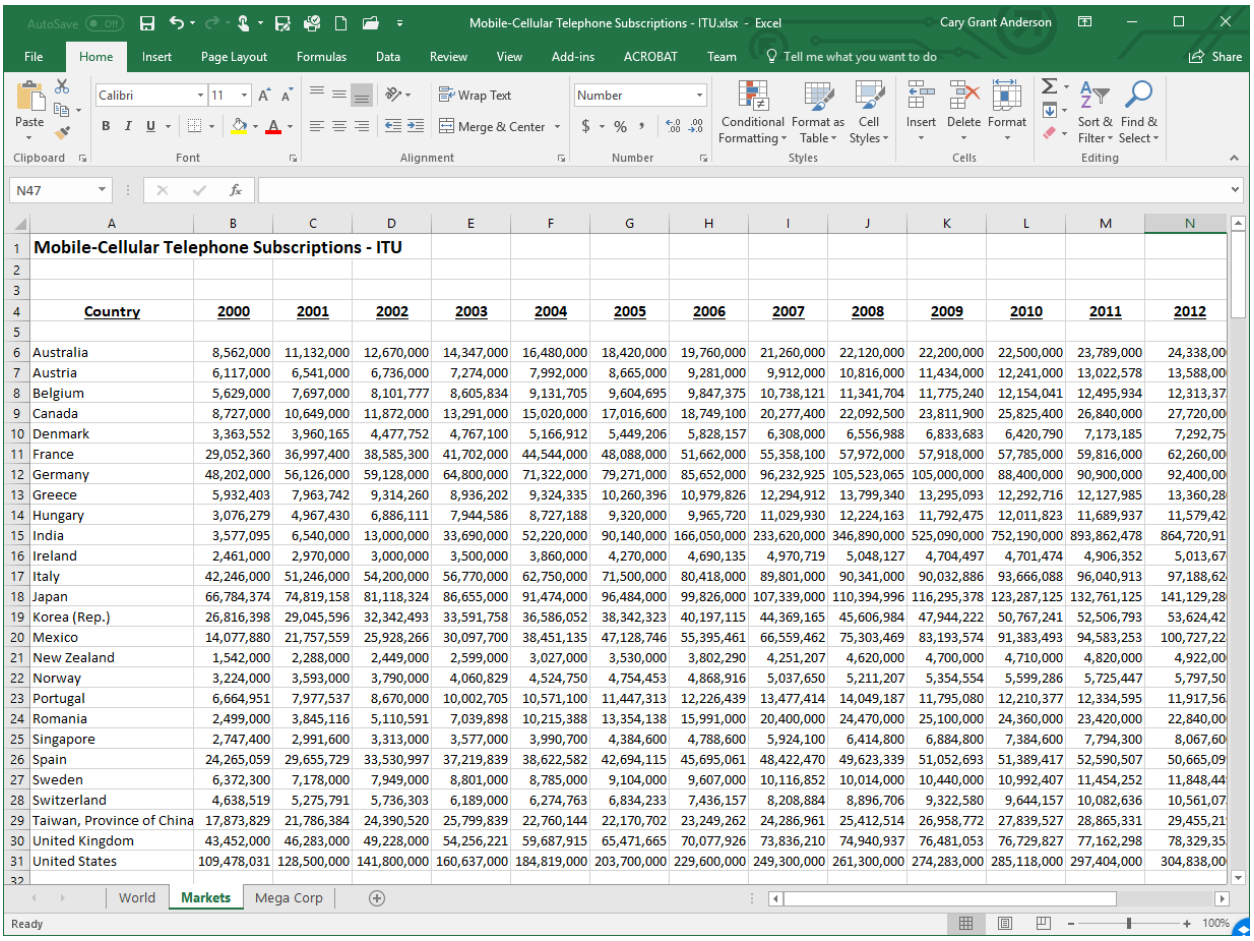
County	# of Phones	# of Cell Phones	% Cell Phones
US	418,681,311	255,395,600	61.0%
England	11,736,417	7,992,500	68.1%
Canada	3,550,190	20,277,400	52.6%

← Header Columns

← Value Cells

From a simple perspective, cubes can be thought of as a large stack of spreadsheet like grids.

Let's take a look at a simple spreadsheet that shows up world wide mobile cellular phone subscriptions by years and by country:



Mobile-Cellular Telephone Subscriptions - ITU.xlsx - Excel

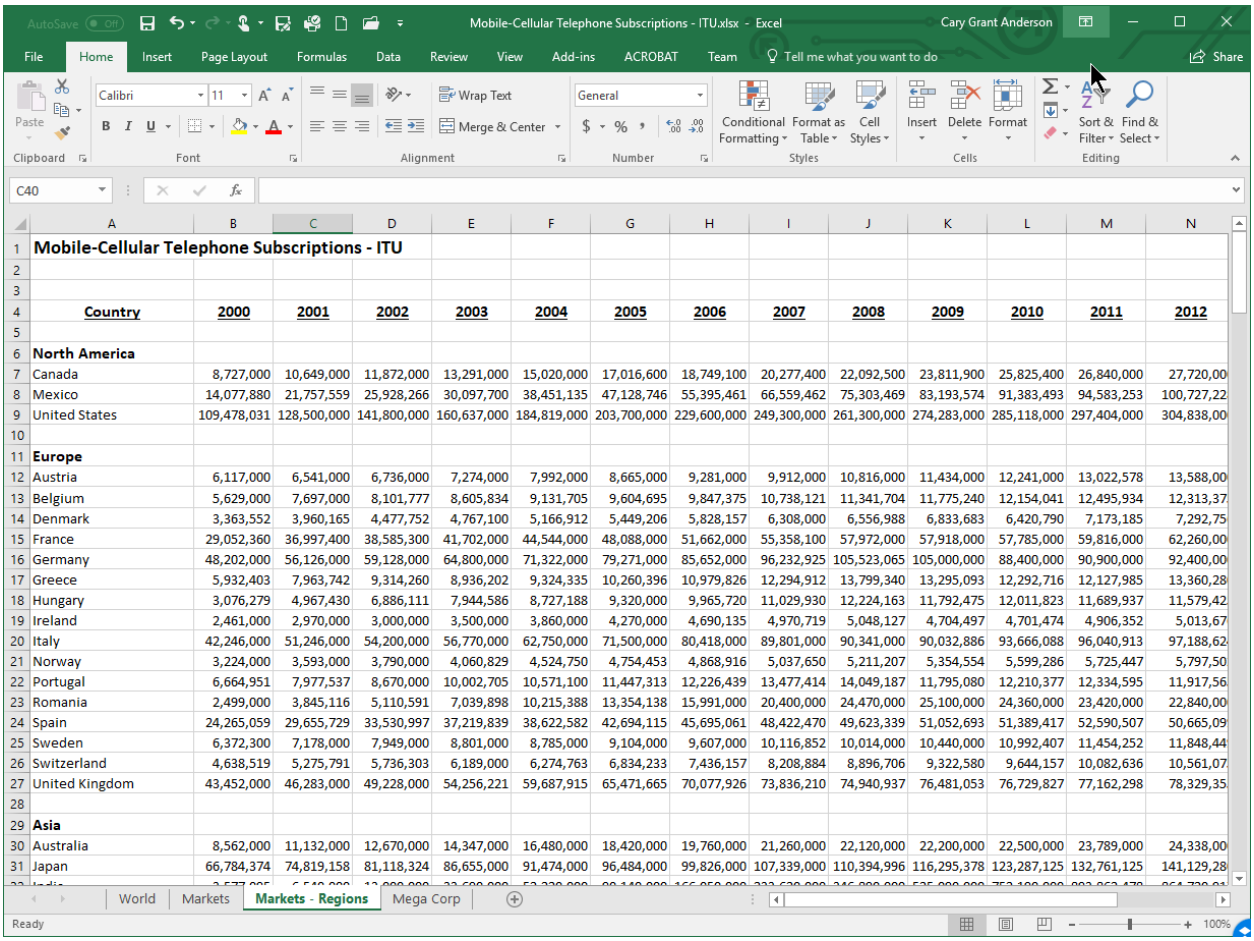
Mobile-Cellular Telephone Subscriptions - ITU

Country	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012
Australia	8,562,000	11,132,000	12,670,000	14,347,000	16,480,000	18,420,000	19,760,000	21,260,000	22,120,000	22,200,000	22,500,000	23,789,000	24,338,00
Austria	6,117,000	6,541,000	6,736,000	7,274,000	7,992,000	8,665,000	9,281,000	9,912,000	10,816,000	11,434,000	12,241,000	13,022,578	13,588,00
Belgium	5,629,000	7,697,000	8,101,777	8,605,834	9,131,705	9,604,695	9,847,375	10,738,121	11,341,704	11,775,240	12,154,041	12,495,934	12,313,37
Canada	8,727,000	10,649,000	11,872,000	13,291,000	15,020,000	17,016,600	18,749,100	20,277,400	22,092,500	23,811,900	25,825,400	26,840,000	27,720,00
Denmark	3,363,552	3,960,165	4,477,752	4,767,100	5,166,912	5,449,206	5,828,157	6,308,000	6,556,988	6,833,683	6,420,790	7,173,185	7,292,75
France	29,052,360	36,997,400	38,585,300	41,702,000	44,544,000	48,088,000	51,662,000	55,358,100	57,972,000	57,918,000	57,785,000	59,816,000	62,260,00
Germany	48,202,000	56,126,000	59,128,000	64,800,000	71,322,000	79,271,000	85,652,000	96,232,925	105,523,065	105,000,000	88,400,000	90,900,000	92,400,00
Greece	5,932,403	7,963,742	9,314,260	8,936,202	9,324,335	10,260,396	10,979,826	12,294,912	13,799,340	13,295,093	12,292,716	12,127,985	13,360,28
Hungary	3,076,279	4,967,430	6,886,111	7,944,586	8,727,188	9,320,000	9,965,720	11,029,930	12,224,163	11,792,475	12,011,823	11,689,937	11,579,42
India	3,577,095	6,540,000	13,000,000	33,690,000	52,220,000	90,140,000	166,050,000	233,620,000	346,890,000	525,090,000	752,190,000	893,862,478	864,720,91
Ireland	2,461,000	2,970,000	3,000,000	3,500,000	3,860,000	4,270,000	4,690,135	4,970,719	5,048,127	4,704,497	4,701,474	4,906,352	5,013,67
Italy	42,246,000	51,246,000	54,200,000	56,770,000	62,750,000	71,500,000	80,418,000	89,801,000	90,341,000	90,032,886	93,666,088	96,040,913	97,188,62
Japan	66,784,374	74,819,158	81,118,324	86,655,000	91,474,000	96,484,000	99,826,000	107,339,000	110,394,996	116,295,378	123,287,125	132,761,125	141,129,28
Korea (Rep.)	26,816,398	29,045,596	32,342,493	33,591,758	36,586,052	38,342,323	40,197,115	44,369,165	45,606,984	47,944,222	50,767,241	52,506,793	53,624,42
Mexico	14,077,880	21,757,559	25,928,266	30,097,700	38,451,135	47,128,746	55,395,461	66,559,462	75,303,469	83,193,574	91,383,493	94,583,253	100,727,22
New Zealand	1,542,000	2,288,000	2,449,000	2,599,000	3,027,000	3,530,000	3,802,290	4,251,207	4,620,000	4,700,000	4,710,000	4,820,000	4,922,00
Norway	3,224,000	3,593,000	3,790,000	4,060,829	4,524,750	4,754,453	4,868,916	5,037,650	5,211,207	5,354,554	5,599,286	5,725,447	5,797,50
Portugal	6,664,951	7,977,537	8,670,000	10,002,705	10,571,100	11,447,313	12,226,439	13,477,414	14,049,187	11,795,080	12,210,377	12,334,595	11,917,56
Romania	2,499,000	3,845,116	5,110,591	7,039,898	10,215,388	13,354,138	15,991,000	20,400,000	24,470,000	25,100,000	24,360,000	23,420,000	22,840,00
Singapore	2,747,400	2,991,600	3,313,000	3,577,000	3,990,700	4,384,600	4,788,600	5,924,100	6,414,800	6,884,800	7,384,600	7,794,300	8,067,60
Spain	24,265,059	29,655,729	33,530,997	37,219,839	38,622,582	42,694,115	45,695,061	48,422,470	49,623,339	51,052,693	51,389,417	52,590,507	50,665,09
Sweden	6,372,300	7,178,000	7,949,000	8,801,000	8,785,000	9,104,000	9,607,000	10,116,852	10,014,000	10,440,000	10,992,407	11,454,252	11,848,44
Switzerland	4,638,519	5,275,791	5,736,303	6,189,000	6,274,763	6,834,233	7,436,157	8,208,884	8,896,706	9,322,580	9,644,157	10,082,636	10,561,07
Taiwan, Province of China	17,873,829	21,786,384	24,390,520	25,799,839	22,760,144	22,170,702	23,249,262	24,286,961	25,412,514	26,958,772	27,839,527	28,865,331	29,455,21
United Kingdom	43,452,000	46,283,000	49,228,000	54,256,221	59,687,915	65,471,665	70,077,926	73,836,210	74,940,937	76,481,053	76,729,827	77,162,298	78,329,35
United States	109,478,031	128,500,000	141,800,000	160,637,000	184,819,000	203,700,000	229,600,000	249,300,000	261,300,000	274,283,000	285,118,000	297,404,000	304,838,00

(Note: Data from ITU 2007 statistics)

This nicely displays the data in a 2 dimensional form.

But what if we want to see the data grouped by region instead?



Mobile-Cellular Telephone Subscriptions - ITU.xlsx - Excel

Mobile-Cellular Telephone Subscriptions - ITU

Country	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012
North America													
Canada	8,727,000	10,649,000	11,872,000	13,291,000	15,020,000	17,016,600	18,749,100	20,277,400	22,092,500	23,811,900	25,825,400	26,840,000	27,720,00
Mexico	14,077,880	21,757,559	25,928,266	30,097,700	38,451,135	47,128,746	55,395,461	66,559,462	75,303,469	83,193,574	91,383,493	94,583,253	100,727,22
United States	109,478,031	128,500,000	141,800,000	160,637,000	184,819,000	203,700,000	229,600,000	249,300,000	261,300,000	274,283,000	285,118,000	297,404,000	304,838,00
Europe													
Austria	6,117,000	6,541,000	6,736,000	7,274,000	7,992,000	8,665,000	9,281,000	9,912,000	10,816,000	11,434,000	12,241,000	13,022,578	13,588,00
Belgium	5,629,000	7,697,000	8,101,777	8,605,834	9,131,705	9,604,695	9,847,375	10,738,121	11,341,704	11,775,240	12,154,041	12,495,934	12,313,37
Denmark	3,363,552	3,960,165	4,477,752	4,767,100	5,166,912	5,449,206	5,828,157	6,308,000	6,556,988	6,833,683	6,420,790	7,173,185	7,292,75
France	29,052,360	36,997,400	38,585,300	41,702,000	44,544,000	48,088,000	51,662,000	55,358,100	57,972,000	57,918,000	57,785,000	59,816,000	62,260,00
Germany	48,202,000	56,126,000	59,128,000	64,800,000	71,322,000	79,271,000	85,652,000	96,232,925	105,523,065	105,000,000	88,400,000	90,900,000	92,400,00
Greece	5,932,403	7,963,742	9,314,260	8,936,202	9,324,335	10,260,396	10,979,826	12,294,912	13,799,340	13,295,093	12,292,716	12,127,985	13,360,28
Hungary	3,076,279	4,967,430	6,886,111	7,944,586	8,727,188	9,320,000	9,965,720	11,029,930	12,224,163	11,792,475	12,011,823	11,689,937	11,579,42
Ireland	2,461,000	2,970,000	3,000,000	3,500,000	3,860,000	4,270,000	4,690,135	4,970,719	5,048,127	4,704,497	4,701,474	4,906,352	5,013,67
Italy	42,246,000	51,246,000	54,200,000	56,770,000	62,750,000	71,500,000	80,418,000	89,801,000	90,341,000	90,032,886	93,666,088	96,040,913	97,188,62
Norway	3,224,000	3,593,000	3,790,000	4,060,829	4,524,750	4,754,453	4,868,916	5,037,650	5,211,207	5,354,554	5,599,286	5,725,447	5,797,50
Portugal	6,664,951	7,977,537	8,670,000	10,002,705	10,571,100	11,447,313	12,226,439	13,477,414	14,049,187	11,795,080	12,210,377	12,334,595	11,917,56
Romania	2,499,000	3,845,116	5,110,591	7,039,898	10,215,388	13,354,138	15,991,000	20,400,000	24,470,000	25,100,000	24,360,000	23,420,000	22,840,00
Spain	24,265,059	29,655,729	33,530,997	37,219,839	38,622,582	42,694,115	45,695,061	48,422,470	49,623,339	51,052,693	51,389,417	52,590,507	50,665,09
Sweden	6,372,300	7,178,000	7,949,000	8,801,000	8,785,000	9,104,000	9,607,000	10,116,852	10,014,000	10,440,000	10,992,407	11,454,252	11,848,44
Switzerland	4,638,519	5,275,791	5,736,303	6,189,000	6,274,763	6,834,233	7,436,157	8,208,884	8,896,706	9,322,580	9,644,157	10,082,636	10,561,07
United Kingdom	43,452,000	46,283,000	49,228,000	54,256,221	59,687,915	65,471,665	70,077,926	73,836,210	74,940,937	76,481,053	76,729,827	77,162,298	78,329,35
Asia													
Australia	8,562,000	11,132,000	12,670,000	14,347,000	16,480,000	18,420,000	19,760,000	21,260,000	22,120,000	22,200,000	22,500,000	23,789,000	24,338,00
Japan	66,784,374	74,819,158	81,118,324	86,655,000	91,474,000	96,484,000	99,826,000	107,339,000	110,394,996	116,295,378	123,287,125	132,761,125	141,129,28

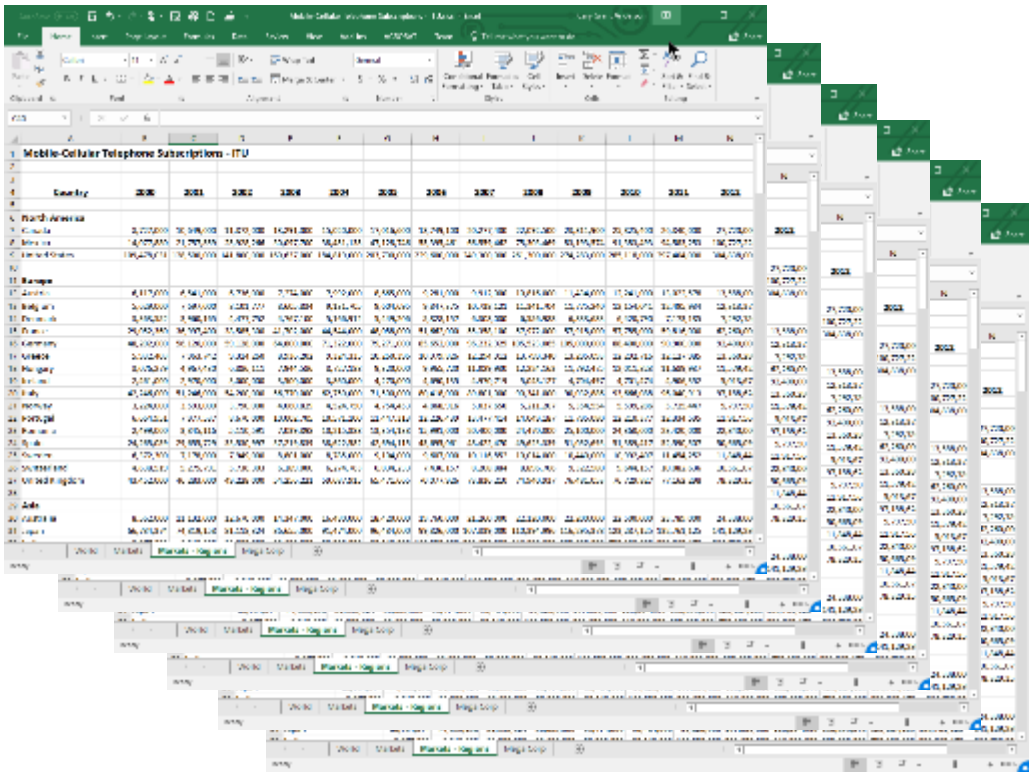
So here we've edited the spreadsheet to group the countries by region. We either have to edit the existing spreadsheet or make a copy and then modify it to be a completely new one if we want to keep the old one.

Now, let's say that we want to show the populations in each country and region. We would have to do this by year of course. And make a completely new spreadsheet for this. Then let's say that we also want to add the number of stores that we have in each country. And then we want to see the totals by quarter. And then also by months. And then by weeks. And then...

I think that you can see where this is going. Report users need more and more data presented as useful and understandable information in various forms and variations. This is really a never ending need as there is always more and more data and more and more ways to view it. And each new view often produces a need and a request for another view or two or three. Being a report developer, you'd be creating more and more new views and the requests from report users would probably never end.

However, by putting that data into a cube you can greatly simplify all of this. The numbers become measures and the countries and stores, and other things become the dimensions. You then build one cube and can then slice and dice and filter and arrange it to show your data in any way that you need it. Much faster! Much more flexibly!

If you imagine a big stack of spreadsheets:



You'll be going from a single 2 dimensional spreadsheet to a stack of 2D spreadsheets that make up a 3D spreadsheet...In effect a 3D cube. Now image that each spreadsheet has the values for one particular dimension – Such as World – Region – Country. It can even go very deep into State/Province – City – Zip/Postal Code. In each stack of spreadsheets, you'll have a World Spreadsheet, a Region Spreadsheet, a Country Spreadsheet and so one. With a Geography Dimension you'd be able to pick the specific level of the Geography, such as Region and Country and so forth. That would get for you and display that level of the spreadsheets stacked in our 3D spreadsheet cube. So, for one dimension, and a specific level within that dimension, you would see just those values and now other.

This is exactly what a cube does for a single dimension. And for each dimension.

So, a cube, being OLAP and thus multidimensional, simply displays the data for the corresponding level within a dimension. A cube automatically pre-calculates the data values so that data is available essentially instantly and does not have to be calculated each and every time someone wants to see that particular level or set of data.

And, being multidimensional, a cube can have not only one dimension but dozens, hundreds, or even thousands of dimensions. Each one behaves just like a stack of spreadsheets...Except that instead of just one stack there's essentially one for each and dimension. Thus, if your cube has 7 dimensions then you can imagine a stack of 7 spreadsheet "cubes", one for each dimension. That gets you that specific data for that dimension and that level of the stack.

And when you add more than one dimension to your pivot table then the cube automatically displays the intersection of the two dimensions with the measure or measures that you've selected. And you can select many measures and have many dimensions. This is where the spreadsheet analogy gets a bit hard to follow. Instead of a nice single 3D cube of spreadsheets, internally the cube has in its internal memory the intersections of all the dimensions that you request along with all of the measures. So that equates to a huge number of spreadsheets and levels and intersections and value cells. This is all quite complex being multidimensional but the important thing to understand is that the cube does this all for you automatically and then automagically displays the proper results to you as a 2D grid or spreadsheet type of display. You don't have worry about all of that complexity because the cube handles it.

So, if you have a cube with a dozen (10) measures (number values) and 10 dimensions (Date, Geography, Product, Store, etc.) you would have something like this:

10 Measures X 10 Dimensions = 100 Report Display Combinations

So, our 10 measures and 10 dimensions cube would easily do 100 report combinations...In a simplistic sense. When we add additional attribute items to each dimension then we also multiply the number of these possible display combinations. If we have an average of 10 attributes for each dimension, then we'd have:

10 Measures X 10 Dimensions x 10 Attributes = 1,000 Basic Report Display Combinations

That would be equal to 1,000 individual unique reports!

Internally, the cube is actually doing many more calculations than just this number. Cube calculations can produce hundreds of thousands, millions, tens of millions, and even hundreds of millions of individual cells calculations depending upon the complexity of the report query that the cube has to calculate. This can consumer gigabytes (GB) of RAM and can also take some time to run even on powerful servers. Typical report queries, however, use a lot less calculations than this and thus RAM and server resources. So, they come back quickly and easily. This is just to give you some idea of the complexities that OLAP cubes can handle.

Internally cubes in their simplest forms are much like a big stack or cube of spreadsheets and this is a convenient way to look at them. Cubes replace having to hand code individual reports with individual SQL queries and thus save a huge amount of development time compared to it you'd develop an equal number of reports or even a subset with conventional SQL reports.

Chapter 6. How do Cubes Work? A Bit More of Some of the Technical Details.

In this section we're going to briefly cover some of the more technical aspects of cubes. We're not going to do a long technical deep dive explanation. Rather, we are going to just cover some of the topics that you may encounter when talking with technical people about cubes. So that you have a good idea of what they mean and are talking about. Without all the deep technical details.

6.1 Data Sources...And Data Cleanliness

One of the most important things to know and understand about cubes and cube development is the cleanliness of data, commonly known as "Data Cleanliness". Clean data is essential for a properly operating cube. It's very important for the data from the cube sources to be in a nice, organized manner and also to be clean. Clean data means that you have some relational integrity of the data. You also have data that doesn't consist of garbage.

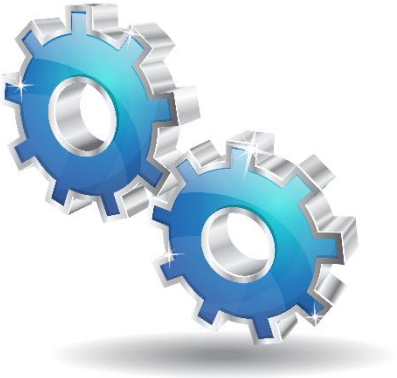
For example, one place I was at developing cubes had a physician's table, and I went in and looked around and the physician's name column had data entries like "123 Main Street". Not a valid physician's name. There were also zip codes and state names in the column. The other columns were a mess too. What happened is that the DBAs (Database Administrators) and developers over the years had tried loading data into it, and they'd made mistakes. Fairly bad and elementary mistakes, and they'd put the wrong data in the wrong columns, and there was a bunch of garbage data in that data source table that didn't make any sense at all.

I had to go and clean the table, and the table had one point two million rows in it and unfortunately, ninety plus percent of that data was garbage. By the time I cleaned it, which took a pretty serious effort, I had about ten percent of the original data. What this means, and why this is important, vitally important, is that you can't always count on your data sources being clean. In other words, having data without having accurate data without problems, without garbage. If you have garbage in your data sources, it's

going to make everything involved in cube development worse – Longer development times, cube data quality problems in the final cube(s), and problems, problems. And cube development projects are often delayed and extended due to data quality problems. At times, too often, 90% of cube development is just clean up the data.

But if you have clean databases, and I have worked with some squeaky clean databases, then cube development is straightforward. And fastest. Cube data quality will also be good to excellent as well. Most of the "squeaky clean" data sources that I've dealt with were machine created. A major telecommunications company had excellent data, because all the data came from their communications network was generated by the equipment, the machines, and the network. So, you didn't have cases where people put the wrong thing in the wrong field. This made cube development, and ETL development, and cube testing straightforward. And the cube data quality was excellent.

6.2 ETL – Extract, Transform, and Load the Data



If you have clean data, it really, really helps, because what's going to happen is you're going to do a process, and this process is called ETL. It stands for ***E***xtract, ***T***ransform, and ***L***oad. The idea is that you extract the data from your source databases, you transform the data by changing it, and then you load it into database tables for the cube.

Extracting the data from one or multiple data sources can be either simple and straightforward...Or a bit complex and involved. Note that one can extract data from multiple data sources including database such as SQL Server, Oracle, MySQL, Teradata, and others as well as files and unconventional data sources. This can include non-relational databases such as NoSQL databases including MongoDB and Microsoft's Azure Cosmos DB (formerly DocumentDB) and others. Note that extracting data from NoSQL databases can be quiet a chore as the data is not stored in a relational database form but rather a document centric JSON (JavaScript Object Notation) form. While there are some ODBC drivers that will interface SQL Server to NoSQL databases, I've had to develop my own ETL programs in Microsoft's .NET C# programming language to pull (extract) the data from the NoSQL databases and then transform it into a relational form for cubing. Why did I have to do this? Well, there are no OLAP cube products directly supporting NoSQL databases, the JSON format is quite different from relational SQL format, and no ODBC drivers or ETL programs were available that could do the job. So, I wrote my own.

Transforming the data is important so as to get it in a form (OLTP to OLAP) that is ready to be processed in the cube. Cleaning the data is also involved with transforming it. There are generally two ways that ETL is developed. The first and most common way is via SQL scripts, specifically stored procedures. These are blocks of SQL code that is stored in the database and are called by a scheduling program named SQL Server Agent. They then run and process the data into a form that can be ingested into the cube. Often these "procs" as developers call them will contain quite a bit of programming code to do the data cleaning

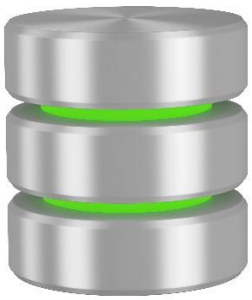
and transformation. And a large part of cube development projects can also be solely in ETL development before one starts the actual cube development. Remember, it's all a system. Not just a cube.

Another popular method for developing ETL is using Microsoft's SSIS, SQL Server Integration Services. Yes, not a very descriptive or good name. SSIS provides a graphical development environment and is rather complex. It can also be difficult to use and deploy. While it is popular with some developers, most ETL can be easily done just with SQL stored procedures so I recommend just using SQL unless there are specific reasons, such as address validation and specialty data cleaning, to use SSIS. You can of course write your own ETL programs in C#. I've done this, and one colleague has worked in a company where they replaced SSIS with their own ETL programs and it worked out very well for them. The big thing with this is that your ETL developers need to know how to code C# and few actually do. This limits this approach but, really, I think that every developer of any type in a Microsoft shop should be able to do basic C# programming.

Often, the ETL puts the data into a staging database or staging database tables. This acts as an intermediate transformational stage. Another set of ETL procs or SSIS packages then do the final data transformation into the cube database tables where the cube will ingest the data as part of cube processing.

6.3 The Cube Staging Database

A staging database is a relational SQL database that acts as an intermediate step or stage in transforming data from the SQL relational model to a dimensional model that the cube can process. What's a dimensional model? Your normal SQL databases uses a SQL OLTP relational model. What this means is data is essentially put arranged for transactional performance efficiency...Or at least that's the goal. A lot of databases are not nearly as well built as they should be, but the idea here is they're optimized for transaction processing (OLTP) and not cube processing (OLAP).



Whether a staging database is used or not is up to your cube architect and developers. Generally, if the data transformations are complex then often a staging database will be used. A staging database can be a separate database or can be contained within an existing database or data warehouse and just be a set of tables used specifically for staging. Often times these tables are named with the words "stage" or "staging".

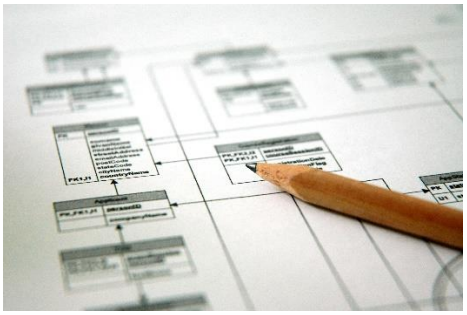
A staging database does not directly affect you as a cube user, but you may hear this term used and thus it is good to know what it actually means. A staging database is an intermediate step on transforming your OLTP data to OLAP form that can be used in the cube(s). This means transforming the data from a relational OLTP form to an OLAP form that uses a dimensional model.

Okay, what's a dimensional model?

6.4 The Dimensional Model

What's a dimensional model? Well, we're going to go over it very fast, and very simply. A dimensional model consists of one or more main tables called a fact table, and then off of it are kind of look up tables called dimensional tables. Now, why are they called dimensional tables? Because the way this is built, and this is built in a different structure than our transaction processing databases, it's called OLAP. Now what's OLAP again? OLAP means online analytical processing. Basically, it means cubes.

Let's stop for a moment here and when you hear people talk about OLAP, OLAP is kind of a generic term.



It will mean not only the dimensional model, it can mean the entire process. It can also mean OLAP cubes. People don't often call it OLAP cubes, they'll just say cubes, but when they talk about OLAP, they usually kind of mean the entire OLAP thing. When they talk about the cubes themselves, most often they are specifically meaning the cubes themselves cubes. It's pretty straight forward but you have to know the context.

A dimensional model is actually simple in many ways, complex and kind of intricate in other ways, but it's completely different from the conventional OLTP relational model. OLAP is completely different from the way OLTP databases are organized and structured and they break most of the rules of OLTP relational database design. So, it takes some time and training for regular database people to understand and become competent with designing and implementing OLAP dimensional models for OLAP cubes.

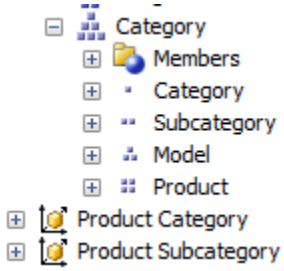
A dimensional OLAP model uses the same database objects as does an OLTP relational model – Tables, columns, data types, etc. – But it does so in a completely different way and form. In a nutshell, all the "fact" data, the core data with numbers, is put into a central table called a fact table. The data categories and groupings are organized into what's called dimension tables. Together, all these tables form the dimensional model that matches what's needed by the cube. The data is ETL'ed into these tables and from here can be ingested into the cube during cube processing.

6.5 Dimensional Hierarchies

A dimensional hierarchy is a key concept of OLAP cubes. Very simply, it's just a tree structured organization of categories or groupings.

A simple one is:

1. Product Category
2. Product Subcategory
3. Product Model
4. Product



As an example:

1. Product Category = Sports Equipment
2. Product Subcategory = Shoes & Boots
3. Product Model = Running Shoes
4. Product = Flying Feet Running Shoes

In hierarchal form:

1. Product Category = Sports Equipment
    a. Product Subcategory = Shoes & Boots
        i. Product Model = Running Shoes
            1. Product = Flying Feet Running Shoes

In practical use, your cube will have a few to a lot of dimensions. Carrying on our example here, if your cube is a sales cube, you will be able to use the above dimensional hierarchy to see all the sales data for the above specific hierarchy...And/or any other combination of product category and subcategory and model and product that you want.

And there will be additional dimensional hierarchies in your cube, such as sales date, location/store, sales person, and many others. By slicing and dicing your data by these dimensions and dimensional hierarchies you can easily and quickly see the exact data that you want and need. This is known as DIY (Do It Yourself) reporting and Self-Serve reporting and BI (Business Intelligence).

6.6 MOLAP – Multidimensional OLAP

What we use in the Microsoft OLAP cubes is known as MOLAP, which stands for ***M***ultidimensional ***OLAP***. This is the most common form of OLAP.

There are other forms of OLAP than just MOLAP. Two of the more common ones which you might encounter are:

- ROLAP – Relational OLAP
- HOLAP – Hybrid OLAP

ROLAP essentially is generating a cube directly from a relational OLTP model without using an OLAP type dimensional model. Pentaho is a reporting product that uses this approach. What's the advantage of this approach? You don't have to create a dimensional schema. So, it can be faster to implement but it is not without its disadvantages. ROLAP is much slower than MOLAP and doesn't scale well. It's not really a good approach in my view, sort of a half baked solution. The reason MOLAP is used is for both its speed and performance as well as its scalability. Why do half measures?

HOLAP is as it is named for, a hybrid approach trying to combine ROLAP with MOLAP. There's no real agreement amongst vendors as to exactly what this means or how it is implemented. It's not very popular and again, it's really an attempt to speed cube development at the expense of performance and scalability.

6.8 Cube Data Considerations

SSAS, SQL Server Analysis Services, is the server that hosts cubes. SSAS hosts cubes with data that has been loaded into your dimensional model, usually at night, but often could be several times during the day. It could be loaded at six AM, at noon, and once again during the night, or it could be loaded every hour. It all depends on how long it takes to process, which ranges in time. For example, if you have an hourly process that takes five minutes to process, then your user just has to know that, six minutes after the hour, for each and every hour of the business day, their cube is updated, and they can then get the data.

The OLAP engine uses the cube definition to load the cube, so it creates the cube in what's known as a cube database. Now, you almost never hear someone say cube database, except maybe a technical person, so you don't really need to know much about it, except that the cube database can hold multiple cubes, but everybody just uses the standard term "cube". Everyone will talk about cubes, and if they're in the same database, it's kind of a technical detail, that doesn't really matter to most people.

These cubes can range from hundreds of thousands of rows, or even less in the cases of really small cubes, up to millions to billions of rows. They can be gigabytes in size to terabytes in size. Obviously the bigger the cube, the more processing time it takes and the more processing horsepower in terms of big servers that you need. The bigger the cube the longer it can take to process it. Query times also increase for bigger cubes.

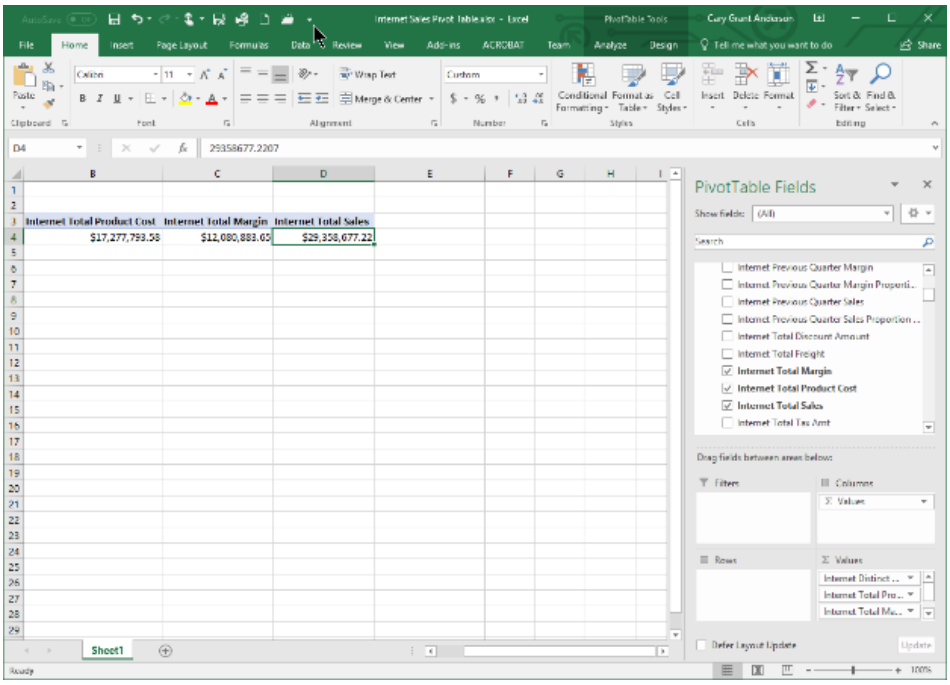
The best practice is to organize data into useful groups and have a number of reasonably sized cubes rather than one big "Super Cube".

6.10 OLAP Cube Browsers

An OLAP cube browser is a special web or desktop program that directly connects to OLAP cubes, and allows browsing and reporting with OLAP cubes, both Tabular and Multidimensional.

Excel Pivot Tables

The most popular OLAP cube browser is the Excel pivot table.



The Excel pivot table allows everyone with Microsoft Office to access cubes, browse the cube data, and create and save (and share) cube reports. Excel is very powerful in many ways, but it can be limited for complex report generation. And also, the user interface is sometimes not as efficient as it could be.

Specialty Cube Browsers

Microsoft does not have a dedicated cube browser besides Excel pivot tables.

There are a number of specialty cube browsers on the market, but none of them have a great deal of popularity.

I'm not going to detail these here because they are not often seen or used as virtually everyone uses Excel.

Custom Cube Browsers

There are a lot of good reporting tools, such as SSRS and PowerBI and Tableau, that support cubes that these are not really cube browsing tools. Instead they are report development tools. The difference here is that some specific report development knowledge and skill is needed to use a report development tool as a pseudo cube browser while a true cube browser is an end user tool that any person can use without having to know the ins and outs of cube report development.

Because of this lack, some companies have developed their own custom cube browsers and reporting systems.

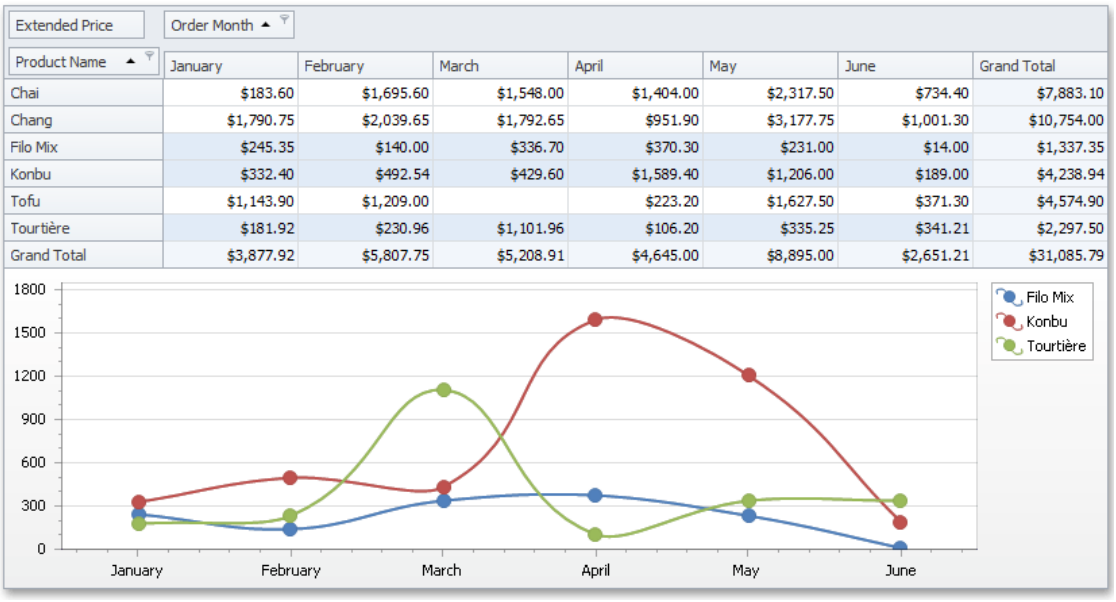
In the major transportation company that I mentioned previously, we developed a web-based front end to an OLAP cube browser, and the users were able to do point and click reports. This was a modified/extended cube browser. We defined the reports in a database table, and then the report user just had to click on the reports. So, this was not a conventional cube browser, but actually cube report tool. The advantage of this approach is its simplicity and the ease with which report users can point and click their way quickly to the reports. They don't have to email Excel reporting spreadsheets back and forth, then open them and navigate to the specific tab, set the filters manually, etc.

I've built a number of both desktop and web based custom cube browsers and reporting tools. These look at bit like the following screen shots:

City

Quantity | Discount | Country

Category Name / Products	Argentina		Austria		Belgium		Brazil		Cana
	Quantity	Discount	Quantity	Discount	Quantity	Discount	Quantity	Discount	Quantity
Beverages	82	0.00%	982	1.55%	272	0.40%	968	2.85%	303
Condiments	45	0.00%	720	1.05%	147	0.45%	568	1.95%	256
Confections	57	0.00%	575	0.95%	270	0.45%	722	1.80%	418
Dairy Products	54	0.00%	1027	2.10%	295	0.25%	683	1.75%	381
Grains/Cereals	20	0.00%	580	0.50%	145	0.00%	315	1.15%	207
Meat/Poultry			362	0.90%	89	0.20%	223	1.00%	141
Produce	33	0.00%	388	0.60%	98	0.00%	182	0.85%	74
Seafood	48	0.00%	533	0.95%	76	0.40%	635	2.15%	204
Grand Total	339	0.00%	5167	8.60%	1392	2.15%	4296	13.50%	1984



Extended Price	Order Month ▲						
Product Name ▲	January	February	March	April	May	June	Grand Total
Chai	$183.60	$1,695.60	$1,548.00	$1,404.00	$2,317.50	$734.40	$7,883.10
Chang	$1,790.75	$2,039.65	$1,792.65	$951.90	$3,177.75	$1,001.30	$10,754.00
Filo Mix	$245.35	$140.00	$336.70	$370.30	$231.00	$14.00	$1,337.35
Konbu	$332.40	$492.54	$429.60	$1,589.40	$1,206.00	$189.00	$4,238.94
Tofu	$1,143.90	$1,209.00		$223.20	$1,627.50	$371.30	$4,574.90
Tourtière	$181.92	$230.96	$1,101.96	$106.20	$335.25	$341.21	$2,297.50
Grand Total	$3,877.92	$5,807.75	$5,208.91	$4,645.00	$8,895.00	$2,651.21	$31,085.79

At present, there are no cube browser or reporting tools that I've found on the market that are as easy to use and as powerful and customizable as building a customized cube browser. I wish that Microsoft would develop one, but they are focused solely on Excel for this role for now.

6.11 Cube Web Portals

One of the most useful things for a company using cubes is to have a cube web portal. A cube web portal is a web portal site that provides a one stop location for everything cubes for the organization. I've built these for several companies and highly recommend them. There's not a commercial product that I would recommend right now. Although I am considering releasing my own open source cube web portal to help spread the word and their development and adoption.

While I don't yet have a good screen shot of a cube web portal (a screen shot being worth a thousand words), let me list some of the things that a cube web portal provides:

1. A one stop location for all cubes and all things cube.
2. An online web cube browser.
3. One click cube reports.
4. Cube and ETL status.
5. News display on cube data freshness, uptime status, maintenance plans, etc.
6. Help and support information for the cubes and reports.

A cube web portal has a lot of user usefulness and power. Once bookmarked, users know exactly where to go and can see what the status of the cubes (and ETL) are. This is very important so that they can see both that the cubes are online and also when they were last loaded with data (processed). Any delays or interruptions in the "data freshness" of a cube needs to be communicated to the users as quickly as possible. In situations where I didn't have a cube web portal, the users would often detect that something was wrong and then call or email me asking about the problem. When a cube depends upon data sources that often have data that arrives late (not a cube problem but a data source problem) I would often have 162 users contacting me wondering what the situation was. When I implemented a cube web browser that display cube status and messages from cube support (me) on the main page, cube users would instantly know what the cube and late arriving data status was before they even accessed the cubes. This was much more efficient and productive than operating in the dark.

And...I also implemented a cube monitoring system that feed real time cube and ETL and data status into the portal, so everyone had complete and up to date information on the cube system at all times.

A cube web portal is a very important thing to have in my opinion.

6.12 Summing It Up

So, this is basically it...Some of the technical components and topics associated with cube that are good to know. Without going into needless technical depth. Just what you need to know.

Chapter 7. Are There Any Good Cube Success Stories?

Yes, there are many good cube success stories. Let me tell you a bit about the first OLAP cube that I worked on. It was for a transportation company in Atlanta in the Operations Research department. I had to pass a very difficult test on graduate level computer science questions as part of the process. Given a red-black tree as follows... It was a test that only a recent computer science graduate could pass because in the real world we really do not use such theoretical things preferring the much more practical things. As I couldn't remember what I had studied about red-black trees, my only choices were guessing...Or recreating the science behind red-black trees and then using that to answer the questions. So that's what I did in something like 45 minutes and passed the to the amazement of the contracting company that was trying to get me in there. "We've put through 25 people and you're the only one that passed the test! How'd you do it?" Sometimes in life we're faced with interesting challenges. One of those challenges is how best to do things. If you're given two dozen potential ways to do something...What is the Best Way? The objective of Operations Research I found out is just that. Find the best route, the optimal balance, the most cost effective and profitable combination. Later in life having worked with Operations Research and some very, very good people at that company I started calling my version of this concept The Optimal Path.

7.1 Cubes for OR Data Analysis...Before Data Science Became Popular

Operations Research is an advanced discipline what seeks to mathematically model and find optimal solutions for such things as transportation scheduling, network optimization, routing, and shipping. Essentially, any process or situation can be mathematically modeled with advanced statistical analysis methods can be considered Operations Research. OR arose during World War II and persisted as a small yet vital field used by a few companies to make decisions using statistical processes instead of semi-educated guesswork. As such, in many ways, it foreshadowed what we now call Data Science. Statistics has been used by smart companies for many years before Data Science made it popular. While Data Science seeks to derive useful insights from its statistical analysis of data Operations Research does much the same but focuses on finding optimal solutions and arrangements of actions within data choices.

So, this department worked hard and smart to find more optimal solutions and paths for saving the company money. And it worked quite well. And one such path used OLAP Cubes. The scheduling of workers and transportation and the handling of work rules for those workers presented not only a large challenge for optimization but also for saving money. This particular cube was built with payroll and expense information and I worked with a really sharp senior consultant who introduced me to OLAP cubes and an equally sharp MBA who introduced me to the business aspects of OR and OLAP. The system consisted of SQL ETL that imported payroll and expenses data into a SQL Server database at night and then a SSAS Analysis Services process that processed that data into the T&E (Time and Expense) Cube. This cube had a custom built web based front end and we also built a basic web portal for it. We had

manager users around the country that would login to the portal to reach the custom web OLAP cube report application. Which was quite advanced for its time. Just click on a report and type in or select the parameters and your cube report was near instantly displayed. I expanded and maintained this system. The senior consultant had built it himself. He was one of the most impressive technical people that I've had the privilege of working with. I added to the cube and extended the portal so that system news messages would display on it to tell the users about the status of the cube and the data in it and also about maintenance plans and periods. This was very useful and successful! And when there was a glitch I didn't have 126 people calling me up and asking what was wrong! (Interesting side note: Years later, I don't see this idea or cube web portals in use in many/most companies! And in some place where I've suggested and demonstrated the idea...It gets rejected and not implemented! So, if this sounds like something that would be useful to you as a report user please suggest it and request it at your company.)

7.2 The Problem...And the Cube Solution

The real beauty of this cube was the money it was saving the company. Their situation was really simple. They had employees all around East of the Mississippi, and these employees were managers, that had to manage the transportation system. And they had to manage the employees, and they needed reports to do that. They had this huge Teradata data warehouse down in Atlanta. They would order these reports and it would take them months to get the reports, especially if it had anything special to do with it. Once they got the report, it was so long ago that if they had to take corrective action, talk to an employee to correct some inaction there…or to change a schedule or something, they couldn't do it because it's three, four, five, six months in the past. So how can you manage your company when your data that you right now to manage properly is now six months in the past? The answer is you can't. Tt just doesn't get done.

The other thing is the employees would have to…the managers would have to get this huge stack of paper report printouts, and they'd have to page through it, and they'd have to figure out, and they'd have to add it up by hand. Great. Does it help them do their job? No. And I'm sure they were already overworked, and they're not going to sit there and work on Sunday morning, on their own time, and then try to do this with a calculator or manually type these things into an Excel worksheet.

The transportation company realized that they had a problem. And they also had this really cool vision to do something about it. I was able to be a part of that vision. What we did was, we built a cube, to look at and display what employees were charging in terms of expenses. We finished the cube and a web front end for it and deployed it. And then the managers could get their data, essentially instantly. Sometimes it took ninety seconds for the report to run, but they could get it in ninety seconds, when they couldn't get it in six months in some instances.

The advantage there was that they had instant access to data, plus they could look at, not only a bunch of paper reports, but they could also make their own reports if they wanted to. This was great, and it didn't take a lot of time or person power. We made it very simple and easy to use. We also had a VP who was just terrific, and we went on a multi-state road show to introduce and train the managers in the new system. The VP told everyone out in the field, you're going to use this, and it's going to save the company millions of dollars. And it was easy enough that they could. Point and click using the power of the cube and see instantly who was charging what expenses and where these expenses could be avoided and thus greatly reduced.

7.3 Million Dollar Savings

The MBA was acting as project manager and he was projected that we would save, conservatively, two and a half million dollars a year. He also said, off the record, that it was probably going to add up to a lot more than that. Four to five million, maybe even as much as nine million. And so, the investment was basically eleven months of my time, to go in and get this put together and put it in play.

Now this was a big company, and they had a lot of rules and constraints and stuff, and sometimes I had to wait six weeks for a database to log into, that I could do myself in thirty seconds. So, some things were slow, but I was able to take the cube, do a little enhancement to it, do some major work on the front end, do the documentation, do the deployment, and do the other stuff. We then deployed it to three hundred and fifty people, and the overhead support wasn't very much that I could handle both support and further development. In other words, I was a one-man band on this. The senior consultant was off working on a very complex and cool project to project the need for capital expenditures and optimize them.

I had to leave the project after 11 months due to my daughter being hurt in a car accident in a distant city. I moved there to be close to her. The project, years later, was still working, it was still going great, and they don't know exactly how much money they saved with it. They never really sat down and quantified it, but they know it was a good thing and was well worth it. If they would have sat down and quantified it, which would mean taking the time to dig it all up, and they'd have to do a lot of investigation and calculations. They were just very happy that it was saving them several million dollars per year and it became a mainstay tool for the mangers.

So, the idea is OLAP cubes used properly, implemented properly, with the proper application in terms of vision, is a great and wonderful thing that can also save significant money. So, this is a good story on the successful use of OLAP cubes.

7.4 Other Successful Cube Projects

I've had a lot of other very successful cube projects. For a non-profit corporation I went in to a failed cube project and restarted and redid it. In six months I not only delivered six cubes but had also cubed all of the data in the organization and trained my replacement. For a very large communications company I developed a terabyte size data warehouse and tracking cubes, a custom front end cube reporting system, and also several communication system status and traffic tracking cubes. All were done properly and quickly and were very successful. I could write in more depth on each one and what and how I made them successful...But that's going to be another book. Most companies don't publicly discuss their internal operations and thus most cube success stories are not made public. There are, however, many companies and organizations with successful cube projects and are today using cubes as an integral and important part of their reporting system operations. The bottom line is, if you know what you're doing, then it's very easy to make cubes and also to make them successful for both users and the organization.

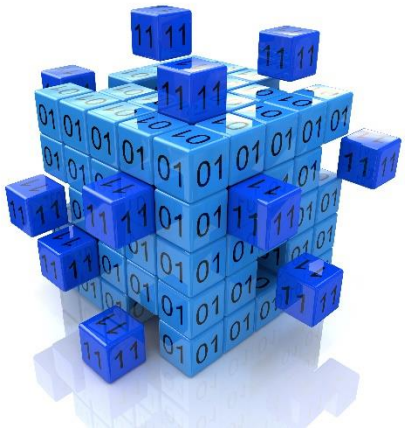
And once you start using cube for reporting you're going to wonder how you ever got along without them! We're going to talk about this in more depth in the next chapter.

Chapter 8. How Do Tabular Cubes Make My Life Easier?

This is an important chapter because there's a lot of things that you can do with OLAP cubes and they will make your work life a lot easier. But first, let's talk about some of the cool things you can do and some of the cool things that I have done with OLAP cubes as examples. And then talk about some more cube applications and uses for the future.

8.1 Cube are for Numbers

The first thing about OLAP cubes is that OLAP cubes are meant for numeric data, numbers. These numbers can be integers, like one, two, three, up to a billion or more. They can be dollars, one, two, three, negative dollars, hopefully not, but positive dollars up to a billion, trillion. They can also be numbers that represent something else. So, in other words, if you have numbers, and you want to do reporting on them, specifically, you want to be able to aggregate them. You want to be able to say, well I want to see all the numbers for all of the United States, and I want to break it down by region, I want to drill down, I want to break it down by state, city, and zip, and then I want to throw more dimensions at it. I want to look at salespeople, who's doing what where, and I want to look at product line. How does the product line contribute to our bottom line and our sales in terms of a numeric amount or dollar values…you can do all of that. That's what OLAP cubes are for.



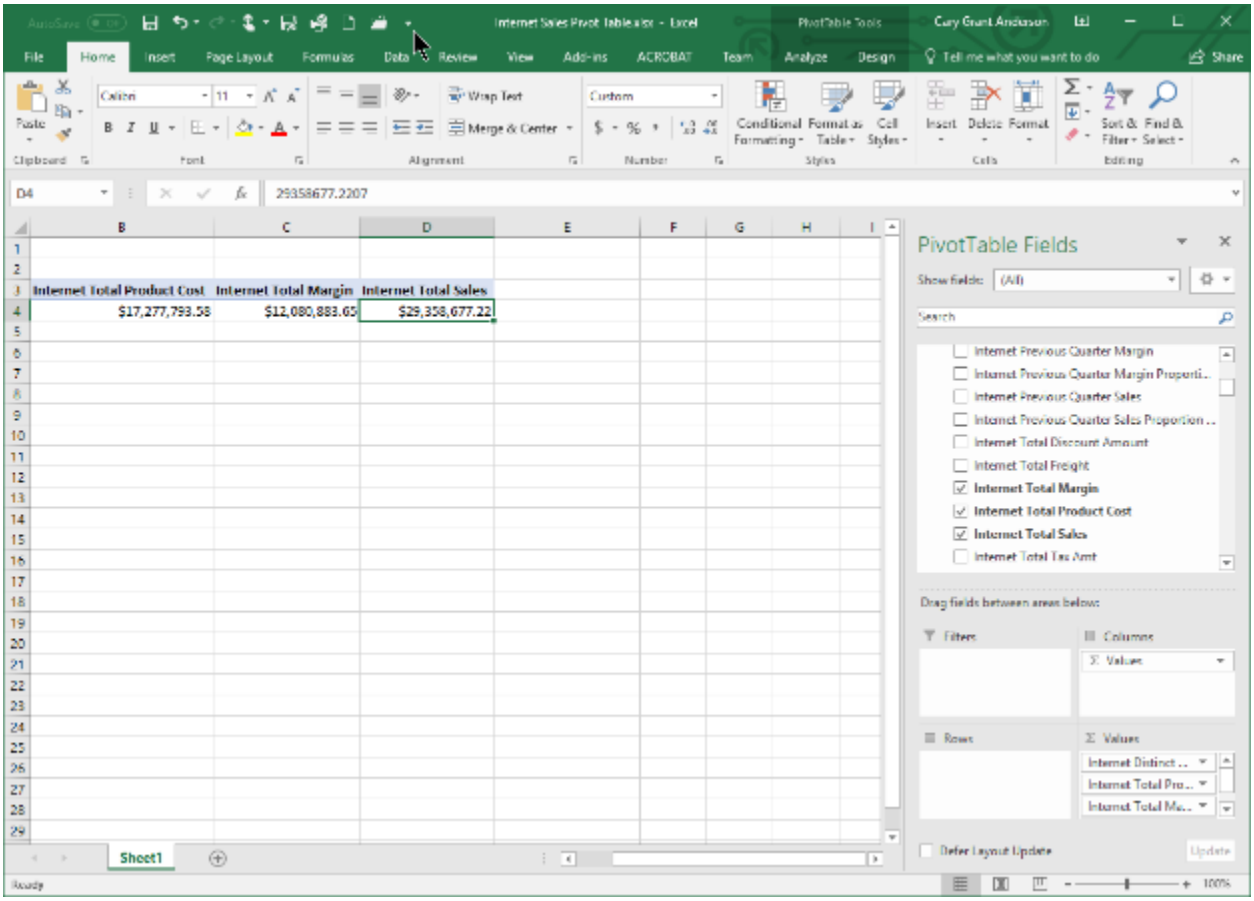
That being said, OLAP cubes are only for numeric data. You wouldn't, for example, do an OLAP cube on addresses. AS it just doesn't make sense. The addresses are not numeric data. Although there is a numeric part in addresses, it's really character data or string data. It's alphabetical data, it's not numeric data, so you couldn't do OLAP cubing on addresses, and yes ,somebody asked me on that. But, the important thing, is you can use addresses or parts of addresses in OLAP cubes. These are what would be called the dimensions, not the measures. The measures are the numbers. Most type of things, we have to use special terms, technical jargon, for specific things. Measures are the core numbers in OLAP cubes. If you think measures numbers, numbers are measures, you're there. Everything else addresses, city, state, department, product types, stuff like that, are essentially descriptors of the numbers that are connected in some way, and those can be dimensions.

8.2 Internet Sales Example

Let's take a look a real world Tabular cube using Excel pivot tables. Here we have created a new Excel spreadsheet, connected to our Internet Sales cube, and selected three measures to look at first. A measure is a calculated aggregate total, such as a sum, a maximum, a minimum, and average, or another calculation, of all the values of that particular type. In this case we are looking at:

1. Internet Distinct Count Sales Order
2. Internet Total Sales
3. Internet Total Product Cost
4. Internet Total Margin

For all Internet Sales data that is in the cube.

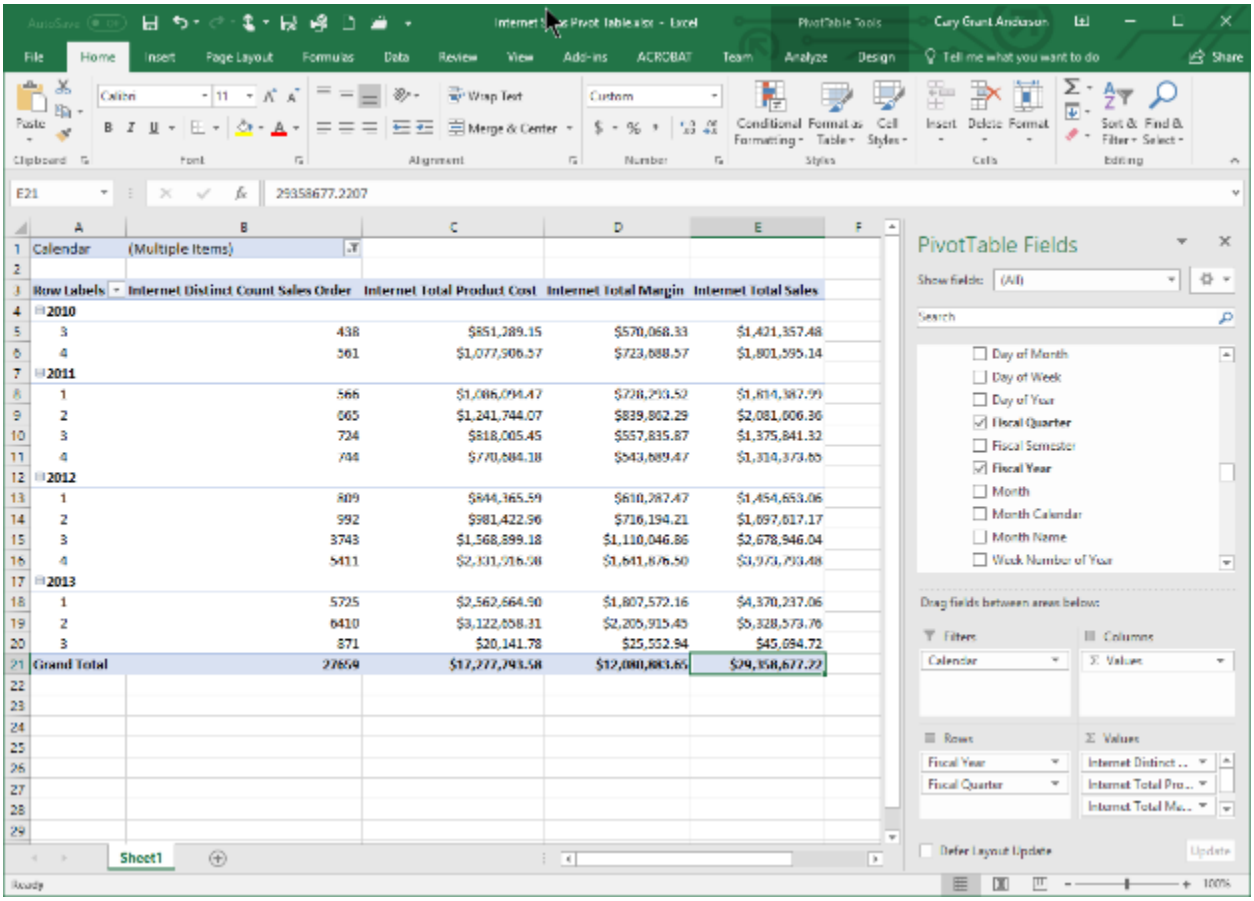


This is because we haven't specified a date filter yet.

Note that we are not in this chapter going to fully explain or train you how to use Excel pivot tables. We are just going to a fairly simple walk through of an example cube using Excel so as to show you what we can easily do with a cube in a real life reporting and data analysis situation.

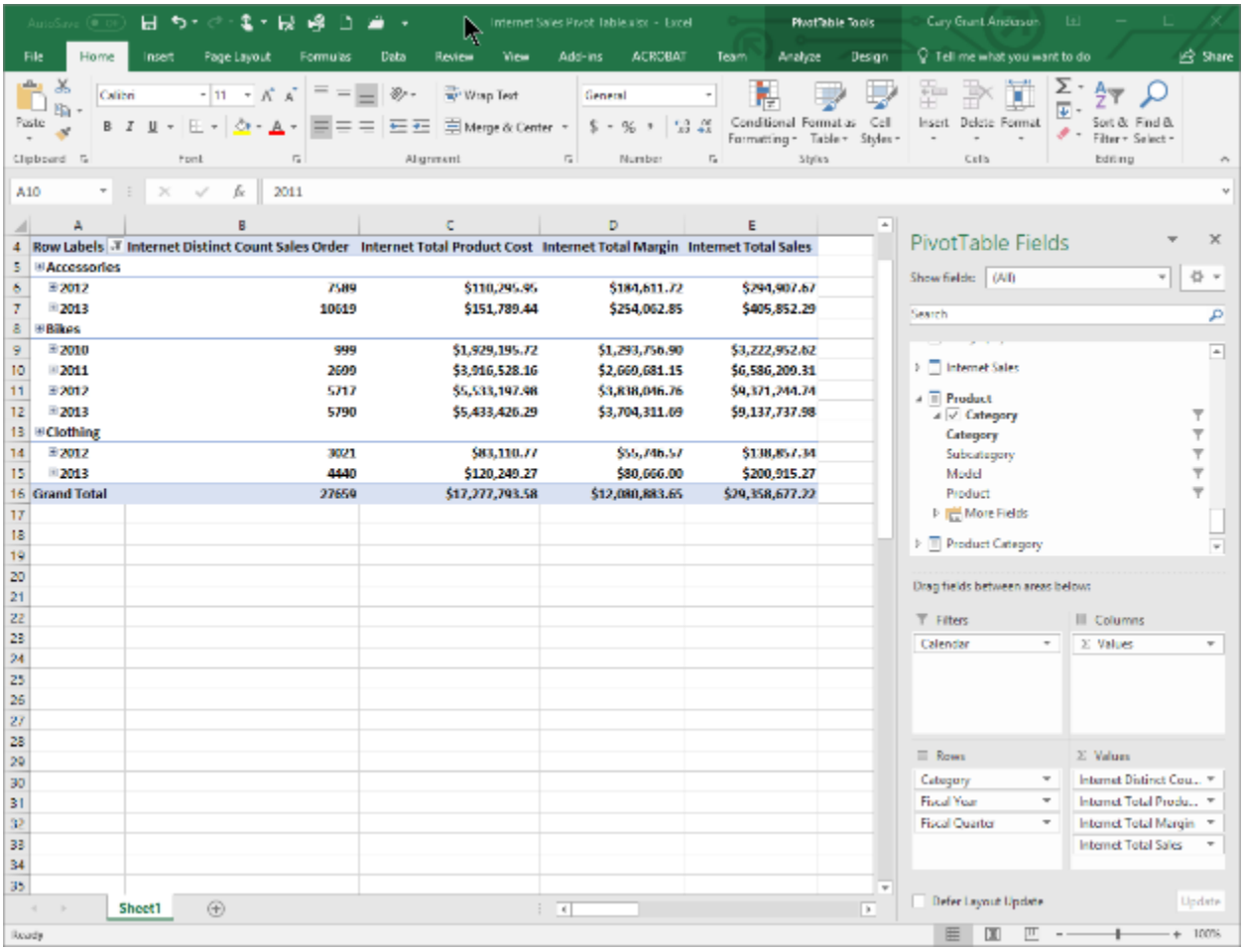
8.3 The Power of Cubes

The Power of Cubes is that you can start at a high level and then work your way down to details in any of a vast and powerful number of ways. Let's start with looking at the sales for the last five years.

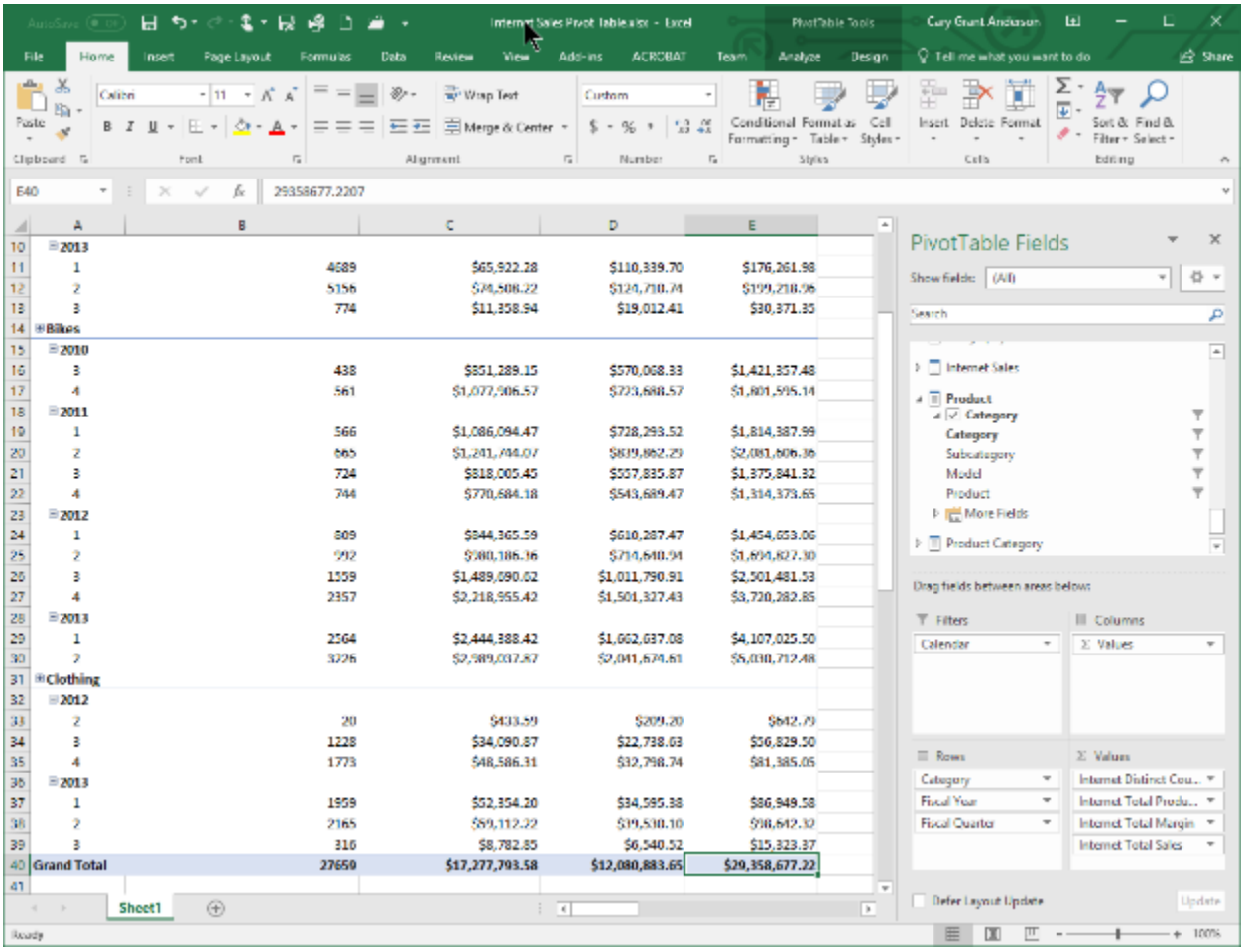


Here we've simply added the Calendar Year as a filter and selected the data for the last 5 years. Then we've added the Fiscal Year and Fiscal Quarter selections of the Date Dimension to the rows. This allows us to "slice and dice" the data by years and quarters. Now we can easily see the number of orders and the costs, margins, and sales for each year and each quarter of those years. All this took only a few drags and drops and a few clicks to get this.

Now, let's get a bit more specific. What types of products do these orders and sales represent?



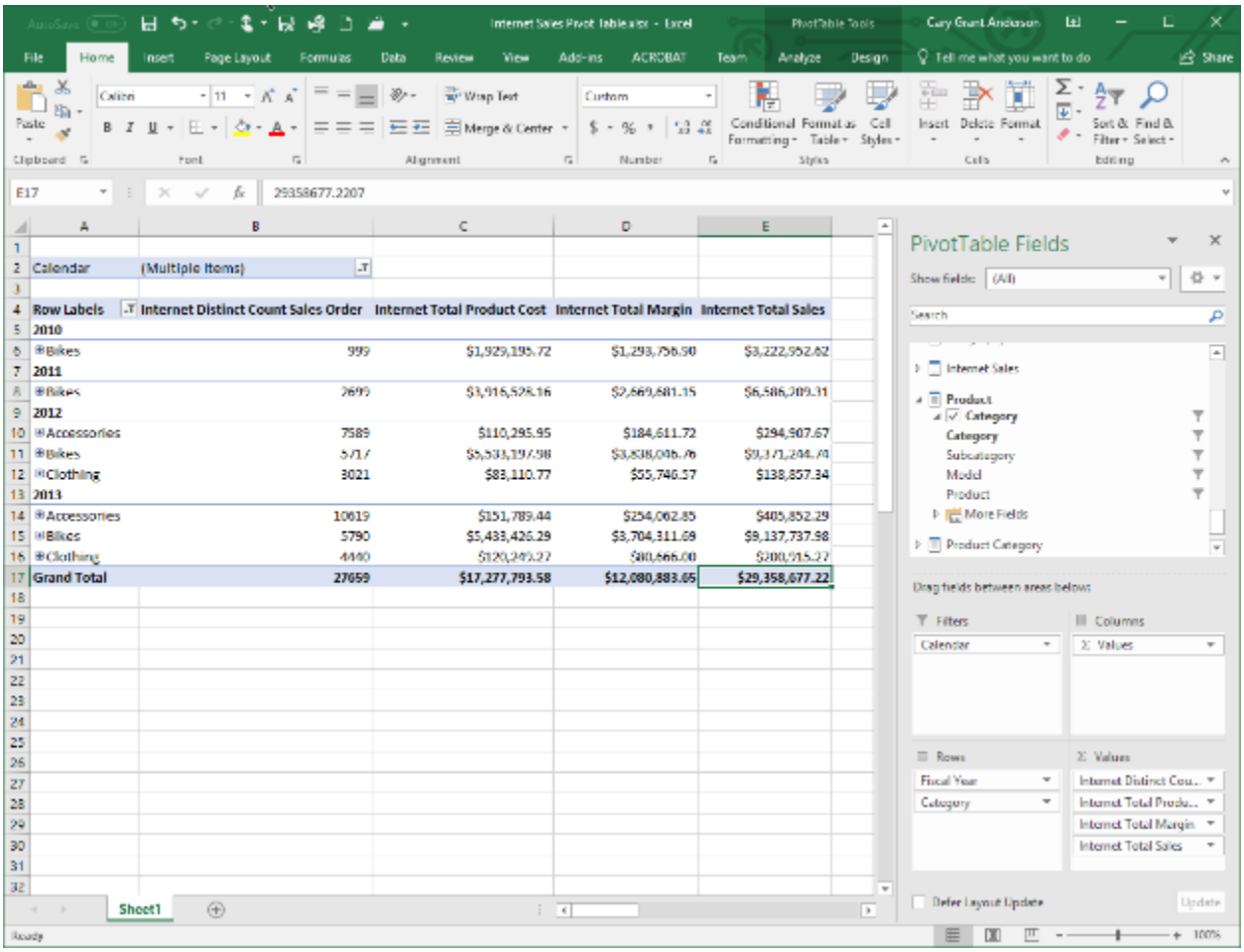
Here we've add the Product Category dimension to the pivot table and hidden the Fiscal Quarters so that we can see Products by Year. We simply need to find the Product Dimension and the Category attribute selection and drag it down to the top of the Rows area. We can now drill down and see the various numbers for the Quarters for each Year by clicking on the "+" (plus) signs.



	A	B	C	D	E
10	2013				
11	1	4689	$65,922.28	$110,339.70	$176,261.98
12	2	5156	$74,508.22	$124,710.74	$199,218.96
13	3	774	$11,358.94	$19,012.41	$30,371.35
14	Bikes				
15	2010				
16	3	438	$851,289.15	$570,068.33	$1,421,357.48
17	4	561	$1,077,906.57	$723,688.57	$1,801,595.14
18	2011				
19	1	566	$1,086,094.47	$728,293.52	$1,814,387.99
20	2	665	$1,241,744.07	$839,862.29	$2,081,606.36
21	3	724	$818,005.45	$557,835.87	$1,375,841.32
22	4	744	$770,684.18	$543,689.47	$1,314,373.65
23	2012				
24	1	809	$844,365.59	$610,287.47	$1,454,653.06
25	2	992	$980,186.16	$714,640.94	$1,694,827.10
26	3	1559	$1,489,090.62	$1,011,790.91	$2,501,481.53
27	4	2357	$2,218,955.42	$1,501,327.43	$3,720,282.85
28	2013				
29	1	2564	$2,444,388.42	$1,662,637.08	$4,107,025.50
30	2	3226	$2,989,037.87	$2,041,674.61	$5,030,712.48
31	Clothing				
32	2012				
33	2	20	$433.59	$209.20	$642.79
34	3	1228	$34,090.87	$22,738.63	$56,829.50
35	4	1773	$48,586.31	$32,798.74	$81,385.05
36	2013				
37	1	1959	$52,354.20	$34,595.38	$86,949.58
38	2	2165	$59,112.22	$39,530.10	$98,642.32
39	3	316	$8,782.85	$6,540.52	$15,323.37
40	Grand Total	27650	$17,277,793.58	$12,080,883.65	$29,358,677.22
41					

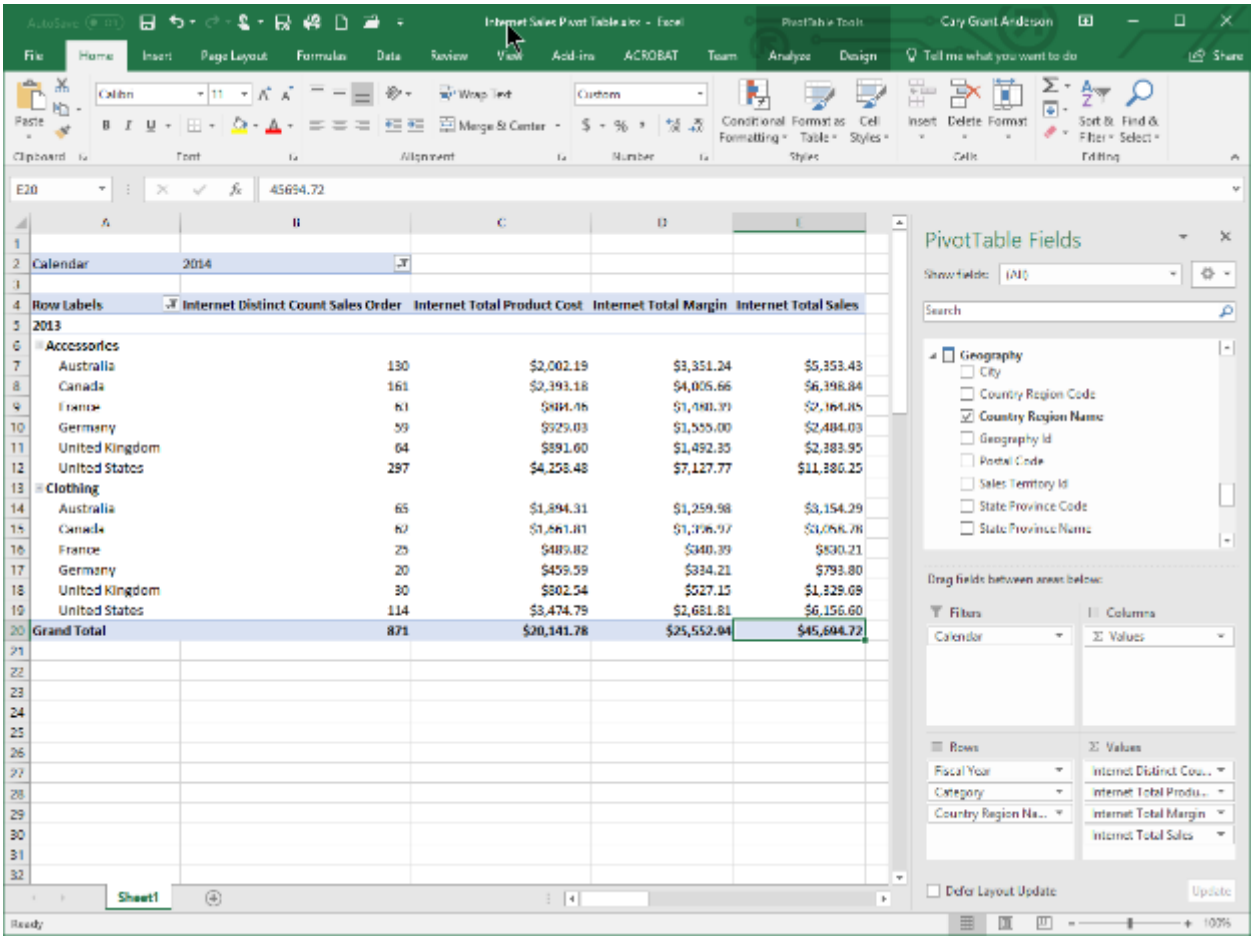
So, we can see how the sales have changed over the years and quarters by the product category. We can, however, do a lot more.

Let's change things around a bit. Let's look at the Product Categories within the Years instead. We simply drag the Product Category in the Row Area to under the Fiscal Year entry and remove the Fiscal Quarter entry. And thus, we then see this:



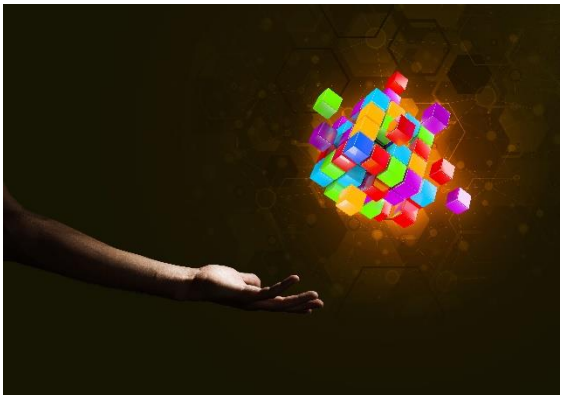
Calendar	(Multiple Items)			
Row Labels	Internet Distinct Count Sales Order	Internet Total Product Cost	Internet Total Margin	Internet Total Sales
2010				
Bikes	999	$1,929,195.72	$1,293,756.90	$3,222,952.62
2011				
Bikes	2699	$3,916,528.16	$2,669,681.15	$6,586,209.31
2012				
Accessories	7589	$110,295.95	$184,611.72	$294,907.67
Bikes	5717	$5,533,197.98	$3,838,006.76	$9,371,204.74
Clothing	3021	$83,110.77	$55,746.57	$138,857.34
2013				
Accessories	10619	$151,789.44	$254,062.85	$405,852.29
Bikes	5790	$5,433,426.29	$3,704,311.69	$9,137,737.98
Clothing	4440	$120,249.27	$80,666.00	$200,915.27
Grand Total	27659	$17,277,793.58	$12,080,883.65	$29,358,677.22

But where, in what countries, where these orders and sales done? We simply drag the Country Region Name item of the Geography Dimension into the Rows Area:



Calendar	2014			
Row Labels	Internet Distinct Count Sales Order	Internet Total Product Cost	Internet Total Margin	Internet Total Sales
2013				
Accessories				
Australia	130	$2,002.19	$3,351.24	$5,353.43
Canada	161	$2,393.18	$4,005.66	$6,398.84
France	61	$884.16	$1,480.39	$2,364.55
Germany	59	$929.03	$1,555.00	$2,484.03
United Kingdom	64	$891.60	$1,492.35	$2,383.95
United States	297	$4,258.48	$7,127.77	$11,386.25
Clothing				
Australia	65	$1,894.31	$1,259.98	$3,154.29
Canada	60	$1,661.81	$1,106.97	$2,768.78
France	25	$489.82	$340.39	$830.21
Germany	20	$459.59	$334.21	$793.80
United Kingdom	30	$802.54	$527.15	$1,329.69
United States	114	$3,474.79	$2,681.81	$6,156.60
Grand Total	871	$20,141.78	$25,552.94	$45,694.72

And we also remove four of the previous years from the Calendar Dimension Filter, so we are just seeing a single year.

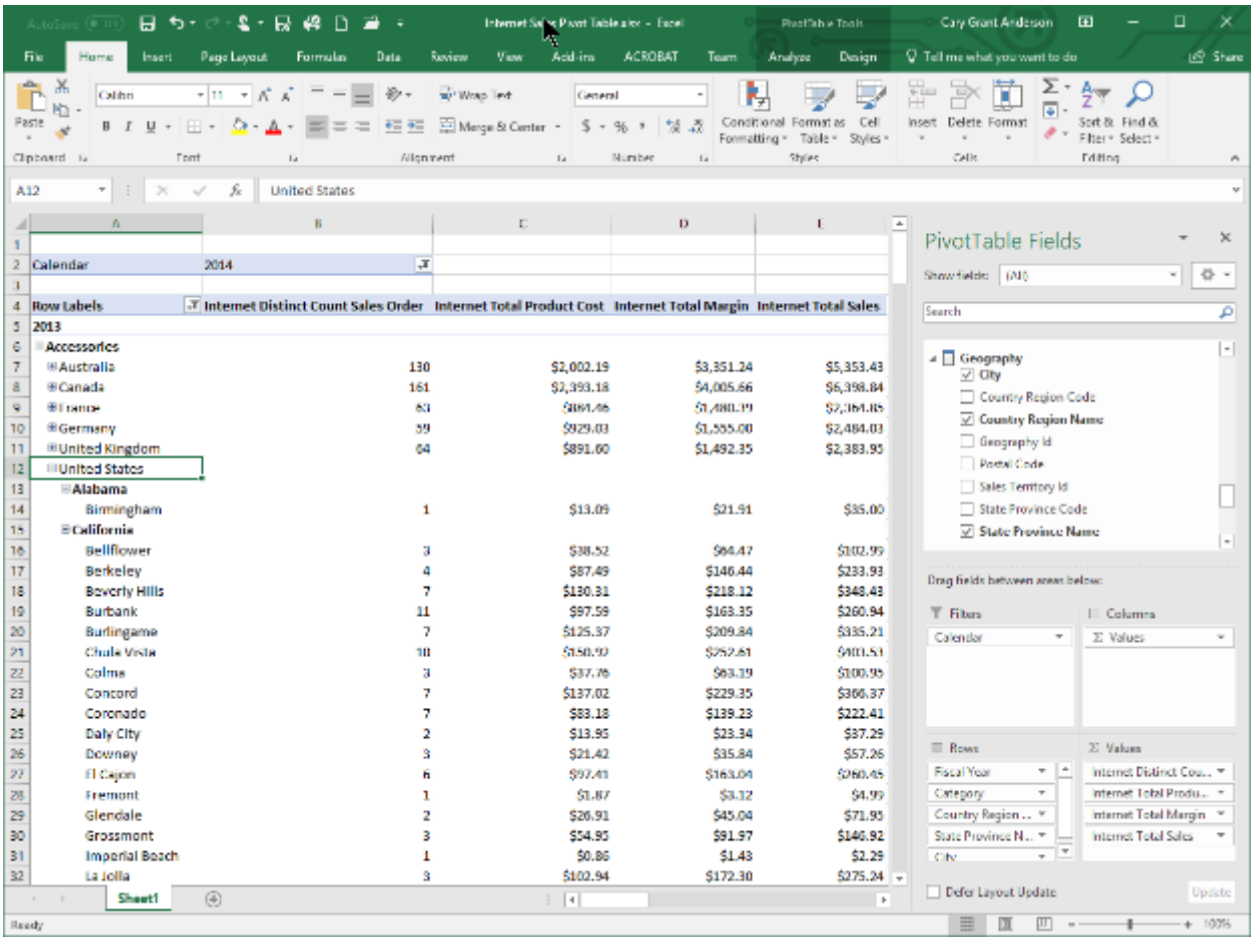


In this display we are using just four measures and three dimensions. This particular cube has 19 measures, 7 dimensions with 112 attribute items, and 2 KPIs. With a cube and a pivot table we can use any combination and variation of these measures, dimensions, attributes, and KPIs in any combination or fashion that we want...Instantly! We don't have to put in a request for a specific report and wait weeks or months for them...We can easily do it ourselves and get exactly what we need when we need it! This is the Power of Cubes! Self-serve reporting. Instant and near infinite ad hoc reports. No waiting. And one of the most useful aspects of cubes and pivot tables (and cube browsers and tools) is the ability to do on demand dynamic data exploration! You can "play around" with the data and poke around the data exploring it from top to bottom. You can discover and see data items, data relationships, and data information that you didn't know, or suspect existed! With formal reports, you can only see what the report is structured to show. Your explorations are very limited. And you cannot browse the data looking for new and useful insights. With cubes you can!

This is the Power of Cubes!

With just a few drags and drops and clicks you can browse and explore the data and create your own unique and useful cube reports and data analysis instantly!

Here we've added the Country Region Name and the State Province Name to our cube pivot table display:



Calendar	2014			
Row Labels	Internet Distinct Count Sales Order	Internet Total Product Cost	Internet Total Margin	Internet Total Sales
2013				
Accessories				
Australia	130	$2,002.19	$3,351.24	$5,353.43
Canada	161	$2,393.18	$4,005.66	$6,398.84
France	83	$891.66	$1,492.19	$2,383.85
Germany	59	$929.03	$1,555.00	$2,484.03
United Kingdom	64	$891.60	$1,492.35	$2,383.95
United States				
Alabama				
Birmingham	1	$13.09	$21.91	$35.00
California				
Bellflower	3	$38.52	$64.47	$102.99
Berkeley	4	$87.49	$146.44	$233.93
Beverly Hills	7	$130.31	$218.12	$348.43
Burbank	11	$97.59	$163.35	$260.94
Burlingame	7	$125.37	$209.84	$335.21
Chula Vista	10	$150.92	$252.61	$403.53
Colma	3	$37.76	$63.19	$100.95
Concord	7	$137.02	$229.35	$366.37
Coronado	7	$83.18	$139.23	$222.41
Daly City	2	$13.95	$23.34	$37.29
Downey	3	$21.42	$35.84	$57.26
El Cajon	6	$97.41	$163.04	$260.45
Fremont	1	$1.87	$3.12	$4.99
Glendale	2	$26.91	$45.04	$71.95
Grossmont	3	$54.95	$91.97	$146.92
Imperial Beach	1	$0.86	$1.43	$2.29
La Jolla	3	$102.94	$172.30	$275.24

And we can "slice and dice" and drill down and drill up throughout the data to find and see whichever and whatever data we want!

With 19 measures and 7 dimensions and 112 attributes we can quickly and easily assemble 14,896 basic report type displays:

19 Measures X 7 Dimensions x 112 Attributes = 14,896 Basic Report Display Combinations

And then display them in an even larger number of custom forms and variations of this measure or dimension first or second or last and so forth!

Image the amount of time that it would take you to custom code and generate SQL report queries for even a fraction of these report combinations! Years!

The Power of Cubes allows you to do this instantly...Whenever you want or need to!

This is sure to make your work life easier and much more productive!

8.4 Some Cube Limitations and Myths

There's some other limitations that are very important to OLAP cubes, and there are some myths to OLAP cubes. One of the things I've encountered is the statement that OLAP cubes can replace SQL databases, and the answer is no they can't. They will replace SQL databases in some ways, and in some situations, but they are not meant to be a complete and utter replacement for SQL databases. So, when you do OLAP cubes, you're going to need your SQL database or databases. You're also going to use your SQL databases as an underlying dimensional model, which is doing the SQL database in a certain way to get it ready for cubing. So, you're going to need SQL databases for that.

So that being said, with OLAP cubes, you have to have numbers. They don't replace SQL databases, they actually compliment them, and then you have to have kind of a clear idea of what you want to do with them. OLAP cubes are really designed for aggregating data, and that's where they came from. They were originally developed for data reports that were aggregated; in other words you have summation. For example, if you have all of the transaction data for all of the cell phones in the United States, but you said, you know, help me cultivate it. Well, you can do a SQL query in a database or you could, perish the thought, download it to a spreadsheet and try to do it there. I do not recommend it, to say the least, because even though Excel has a very large number of rows that it can handle it is not a good use of your time when you can near instantly do the same thing with a cube.

8.5 A Cube Example

If you wanted to get all of the cell data, and you put this in a database, and you wanted to query it and say okay, how many calls were made last Tuesday? In the United States? Well, you'd have to write a SQL query, and you'd have to be very good at SQL to do that, and you'd have to run a query to get a data set, and then report on that or use that, or you'd have to run it as part of a report. So, you say, well, okay, how many cell phone calls were made in Atlanta last Tuesday? Okay, well, it's similar to that query that we just did, but we have to do another one.

This is where OLAP cubes have their strength, because OLAP cubes inherent support aggregations. What you do is you make an OLAP cube of all the call data, and then you just use the browser, and you drag and drop items, and you say okay, my location, my city is Atlanta, show me everything, and then this is my

date, and you do some filter settings and you get exactly what you want. Instantly. You can see exactly what you want. So that's the power of OLAP cubes.

Getting back to the question of what you can do with them is that you can do aggregations and custom reporting on numeric data. I said before that the data can be essentially any numbers. Let me give you some ideas of some of the cubes that I've done. I started out with cubes doing a payroll cube that showed expense payments to employees of a major transportation company. This was interesting because it was a union shop, and management was interested in preventing excessive claims that employees could make if there was no oversight on the expense claims. Without adequate reporting there could be no adequate oversight. The idea here was they wanted to see many different things such as location, employee, reason codes, hours and such as to the claims so they could be reported on and analyzed properly.

OLAP cubes have traditionally been widely used in the financial industry because people there work extensively with numbers. Anything financial-wise, numbers, can be done in OLAP cubes. Additionally, marketing and sales data is often put historically into OLAP cubes. For example, there's a wine manufacturer in California that I talked with a representative of their firm…Not only are they into wine quite a bit, but they're into OLAP cubes. Why are they into OLAP cubes? Well, because they have quite a bit of a variety of wine products, and they know that each wine product addresses primarily a different market niche, so by analyzing the sales data and mixing it with the marketing data, they can determine who is buying what, or who is not buying what. So, with doing that, they can target their marketing, and they can also target new products into these areas.

What does this mean? This means they're proactive, they're very efficient, and they make a lot of money, and they really, really enjoy their wine when they're making a lot of money. I have developed a workflow cube which was a time to process flow cube. It showed the steps that a mortgage application went through at a non-profit company that processed mortgages for low income people. It showed the steps, and the time that each step took, and then the amount of time at each process step. And the total amount of time for all the different applications. The elapsed times were the measures, and the dimensional information was who was processing what, when, how long it took, what was the current status? What's hung up here? Where is the process is okay, and too slow, and who is doing what and not doing what, and what's their total throughput?

The idea with this was to feed a dashboard, so you could log in and instantly see on your own customized dashboard how you're doing, or how the people under you are doing. Pretty cool thing, huh? Let's look at some of the other cubes that I've done.

I've done health care statistical analysis cubes, and one of these got very, very complex. The idea is that you have incidents of certain conditions in the population, and you have some other numeric metrics, numbers, associated with the health care statistics, and by putting them in a cube, you can actually compare and see what's going on. You can look at the aggregates and drill down. It was pretty cool, very complex.

8.6 Case Studies: Multiple Types of Cubes Grant Has Built

At this point in time, I should point out something. Essentially, there's multiple types of OLAP cubes you can build. The most common one is a reporting cube, which is designed for reporting on specific things. Another type is an analysis cube, and with the health care statistics it was an analysis cube mostly. It was also used for reporting. The difference is, analysis is used basically for researchers, often market researchers, or in this case, there are different types of researchers, that needed access to the data to look deeply into the data and make cross-correlations between different groups…different groups…dimensional groups, and sometimes different groups of numbers.

You can do this with OLAP cubes, but they get complex. The most complex OLAP cube I've done had….it had seven fact tables, and the fact tables, you can think of them as the main tables consisting of all the records that have the numeric data in them, and then they had seventy some-odd dimensions, which are the different ways you can slice and dice the data.

Oh, introduced a new term: Slice and dice. In a cube browser, when you drag and drop different dimensions in different arrangements, to get the numbers you want to see, that's called slicing and dicing. It's a cool industry term, and people use it all the time because you can do it. Now you never talk about slicing and dicing a SQL database because it doesn't apply. You don't have a SQL database browser equivalent to an OLAP cube browser, unfortunately. It would be great if someone invented one.

Now, I also did a performance cube, which looked at performance metrics from a network of a national network of cell phone equipment. Now we use our cell phones, we make our calls, and we think, oh this is really cool, what I'm holding in my hand, but what we don't really have an appreciation for is the vast infrastructure, the vast technological infrastructure that goes to support those calls, and there's a whole army of people behind that technological infrastructure that make sure everything works properly so your calls get completed and you have a good, pleasant calling experience. There's several national companies that are doing this, and they have a lot of equipment, and they need to monitor this equipment to see what's performing well, and what's not.

For example, dropped calls. All of us have probably been in a situation where, for whatever reason, maybe in a car moving, the signal reception is not so good, we'll answer a call and the call will just stop. The equipment in the cell phone network monitors this. So, the people in the cell phone companies need to look at this to see what's going on. So, they have actually a huge amount of performance information that needs to be brought together, metric supplied, which are calculations like the calculations on an Excel spreadsheet or a calculated column in a report, and then this can be put into a cube, so you can slice and dice the cube and get your information. This is another one of the cubes I've build.

I've built a small cube just recently that tracked developed performance from a developer task tracking system, and the reason was I needed to look at my developers, what did they have for this sprint, and that sprint, and the next sprint, and how to group these, and how to make sense of this data, and I didn't have access to a relational database, so I actually went to a web interface, downloaded it into an Excel spreadsheet, spent a little time playing with it in Excel, and finally said this is ridiculous. I put it into a SQL

database, turned it into a dimensional model, made a cube, viola, an hour and forty five minutes later, I have a cube. I'm looking at it, I get all the answers I need, case closed. So, I don't have to sit there and spend even more time on a spreadsheet.

You can make all sorts of cubes, big and small, basic and complex, and use them in all sorts of situations for all sorts of purposes. Cubes are very versatile and highly useful!

Chapter 9. What Do I Need to Learn to Use Tabular Cubes?

To use Tabular cubes, or any OLAP cube, you need:

1. A general understanding of cubes, and
2. How to connect to the cubes as a data source for the cube reporting tool(s) you are using, and
3. And also of course how to use the cube reporting tools and/or cube browser.

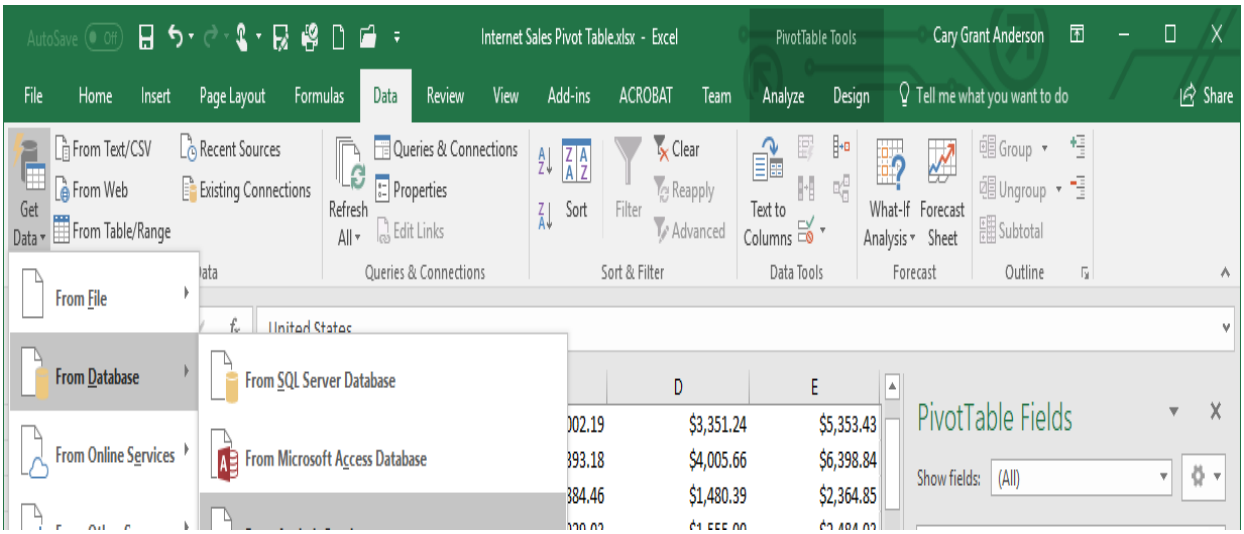
This book should provide you with more than enough information for a good general understanding of cubes.

9.1 Connecting to a Cube via Excel

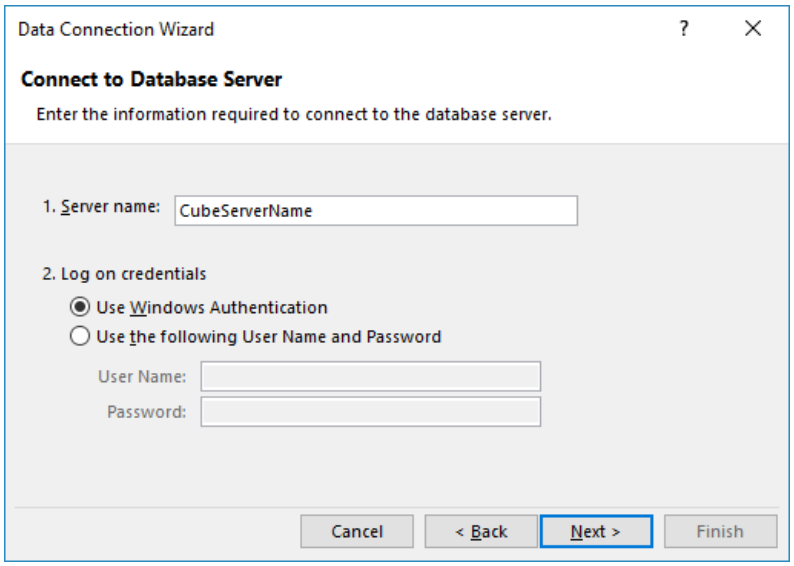
Connecting to cube data sources is simple and typically involves just a few steps. Connecting Excel to cubes is easy and if you know pivot tables then you're all set! If you don't know Excel pivot tables reading a couple of articles on them or a short book or course will get you started.

For Microsoft Excel these steps are:

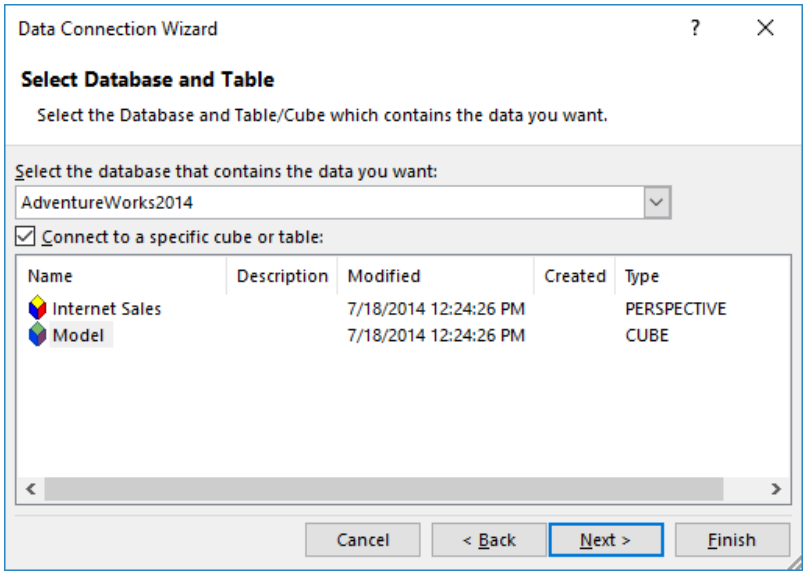
1. Select the Data tab.
2. Select the Get Data drop down on the far left of the menu strip.
3. Select From Database.
4. Then select From Analysis Services.



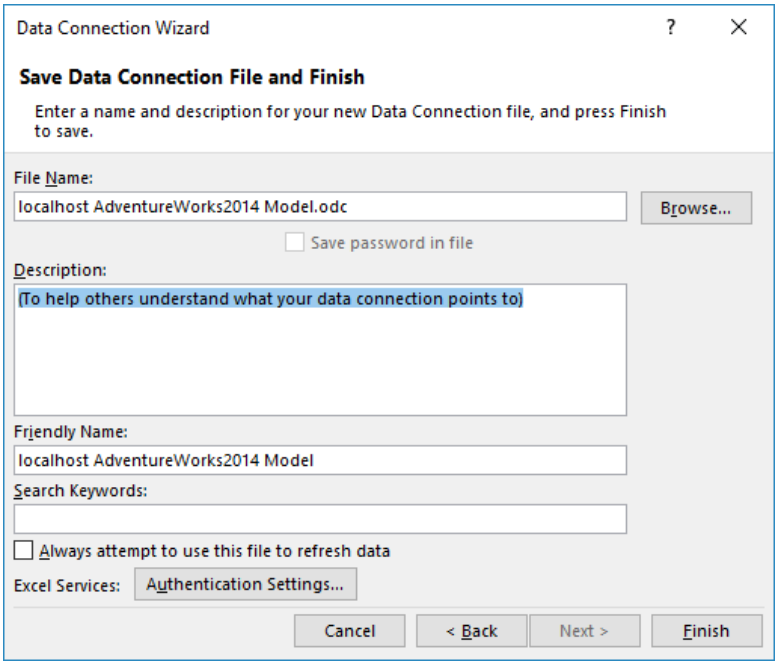
5. The Data Connection dialog will then appear.
6. Type in the server name in the “1. Server Name:” field and click “Next”.
    a. Note that SSAS servers only support Windows Authentication so just leave that option selected.



7. The Select Database and Table dialog then displays.



8. The “Select the database...” drop down combobox contains a list of the cube databases on the SSAS server.
    a. Select the one you want.
9. The list box area will then display the cubes and the perspectives in that cube database.
    a. Note that “Model” is the default name for cubes and should be changed in cube development to a more descriptive name.
    b. Full cubes will always have a type of “CUBE”.
    c. There may be one or more types of “PERSPECTIVE” listed. A Perspective is a simplified subset of the cube measures and dimensions that help make it easier for users to use a complex cube. And also, they help target and focus a subset of the complete more complex cube to a specific type or group of users (a cube can have more than one type of users).
10. Select the Name of the cube or perspective that you want.
    a. It should then highlight.
11. Click the Next button.
12. The Save Data Connection File and Finish dialog will then come up.



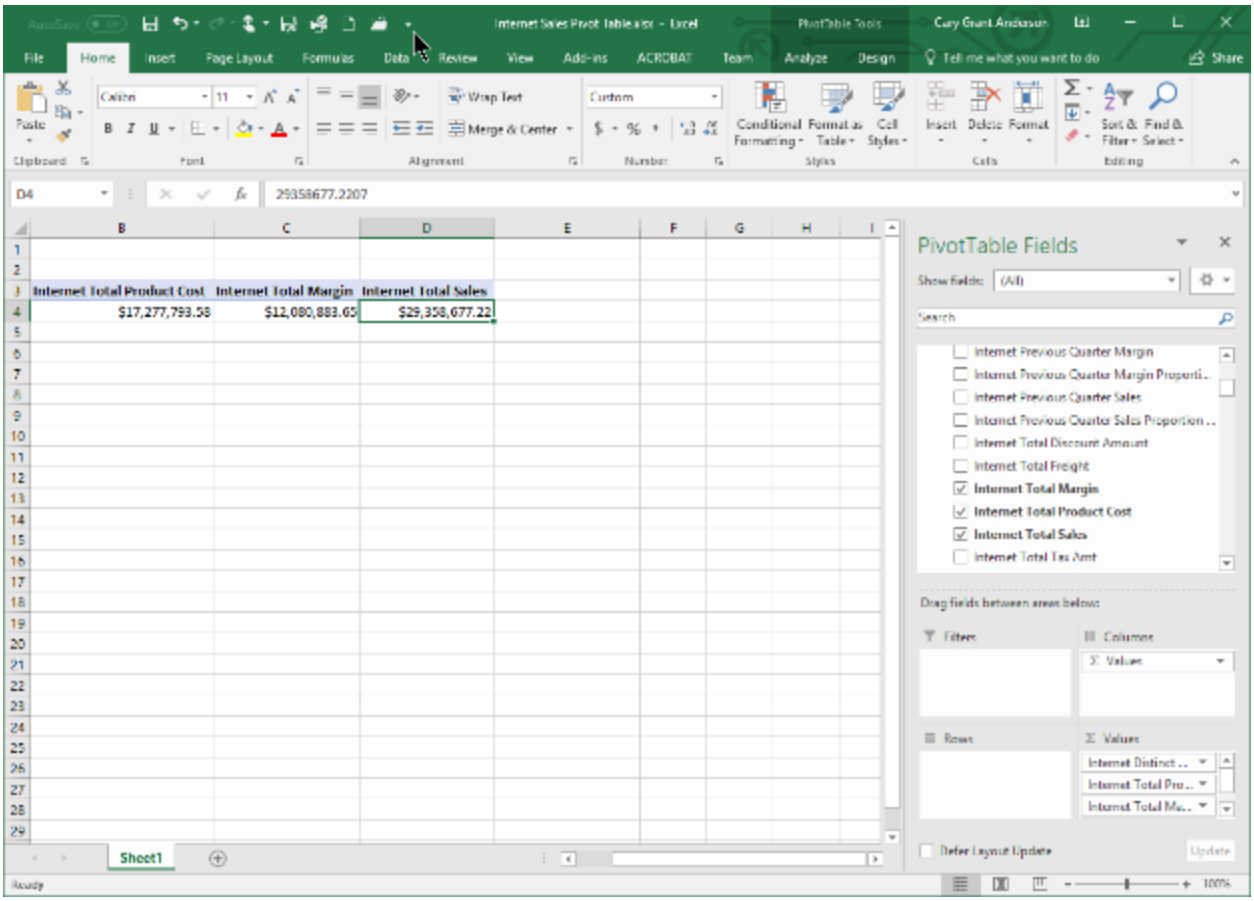
13. You can just click “Finish” here or you can also specifically name the Excel data connection file and location as well as type in a description and a Friendly Name which is a more user friend name for this data connection. Generally, it’s going to be the same as cube name. Note that the server name is by default displayed in the beginning of the Friendly Name. Here it says “localhost” which is the default name of the local development machine. This will say the server name that you put into the first dialog box.
14. Click Finish and you have just completed setting up your cube data connection for Excel!

It's that simple! Once you've created one data connection to a cube you can then reuse it within Excel. No need to create multiple data connections. Just use the "Existing Connections" button instead of the "Get Data" button for this.

Other reporting tools will have similar steps to connect to cube data sources. Please see the documentation for those reporting tools for the specific steps. And also, for specific instructions on using your reporting tool or cube browser with Tabular and/or Multidimensional cubes.

9.2 A Very Quick Start on How to Use Excel Pivot Tables with OLAP Cubes

Using Excel pivot table with OLAP cubes is very straightforward. Just add a pivot table to your spreadsheet with a data connection to your cube. You can then simply drag and drop PivotTable Fields items to the bottom four field areas on the bottom right of the spreadsheet.



While this is simple to do there's a lot more options, features, functions, techniques, and a few gotchas with using Excel pivot tables including with both Multidimensional and Tabular Cubes. This is a subject for a separate book and course and there is not sufficient space in this already large book to cover this. Like any reporting tool, Excel needs a little bit of learning to use it efficient. And when you do, you'll be creating reports with cubes fast and easily. Creating Excel pivot table reports with cube is about 10x...Yes, that's TEN TIMES FASTER...Than with having to hand craft SQL queries for each and every report data source.

And it's really easy to create a starter Excel report spreadsheet for a new cube. Just connect to the cube as detailed in the previous section, plop a pivot table down on the spreadsheet, drag and drop a couple of measures and dimensions to the quad display fields on the bottom right and save it! You've just made a starter cube Excel report spreadsheet!

Chapter 10. After Words....And Bonus Chapters!

This started out as a small and short book and expanded during writing into a much larger book. The reason for this is that there is so much information to transfer, mostly real-world experiential, and also a great deal of new concepts to explain. As I wrote this book I tried to include the information that you really need...And you really need a fair amount to understand OLAP cubes. I've seen a lot of people have all sorts of problems with cube projects that they really didn't need to encounter. As a cube user, get your cube developer(s) to give you and the other cube users:

1. Documentation on the cube.
    a. I recommend two specific documents:
    b. Cube Quick Start Guide – Basic information on the cube, what it contains, how to connect to is, and other useful getting started information.
    c. Cube Reference Guide – A listing of the measures, KPIs, and dimensions in the cube along with descriptions and definitions of the cube calculations and business rules for the measures and also the KPIs. At some point in time you'll need this information and having it handy is, well, quite handy!
    d. I also recommend that both cube documents are put online in both HTML and PDF formats, preferably right in the cube portal so users always know exactly where these documents are.

2. A brief orientation to the cube.
    a. Have them walk you through the cube to get you familiar with it.
    b. Just a basic informal walk through.
    c. They should also be able give this to every cube user as well as...

3. Training on the Cube.
    a. Developers should have been working with you closely or someone in your reporting group or department while the cube was being developed.
    b. So, they should be able to give you a nice walk through type training on the cube.
    c. And be able to answer all your questions about it
    d. Formal training is always nice but often there is not time or skill set to develop it.
    e. So informal training using the above two Guide documents and a detailed walkthrough are usually sufficient.
    f. If done remotely via a screen sharing tool such as Skype record it and post it online and on the cube portal as well for occasional reference and for new cube users.

Just building a cube is not enough. Cube users must have the information and the training that they need to properly understand and use the cube.

I hope that this book has helped demystify cubes and has given you a good understanding of cubes, what they are, how to use them, and some of the real world details and experiences with using them.

Bonus Chapters

Re-read sections of this book when necessary as a on demand reference guide. The two bonus chapters following this chapter are quite useful for this:

- Chapter 12 – Introduction to the Microsoft BI (Business Intelligence) Stack
- Chapter 13 – The Tabular OLAP Cube Terms and Concepts Mini Dictionary.

Our Book Website

Both these chapters and more are on our website. Visit and register and get email notices of book updates, freebies, special offers, and new books and courses. We'll also have an online cube where you can explore and play around with an online cube browser and reports.

Please visit for additional information and to register for future announcements and releases!

www.CGrantAnderson.com/Tabular

Our Free Posters!

Our book website has a number of free Tabular Cube posters that you can download in PDF format and print and use freely in your cube, department, or company.

They help get the message of what Tabular Cubes can do for your department, project, and company.

See our website at:

www.CGrantAnderson.com/Tabular/FreePosters

Stay tuned and sign up for release announcements on the website.

Additional Tabular Cube Books

This book is just one book in a series of books (and courses) on Tabular Cubes.

Look for The Tabular OLAP Cube Best Practices Source Book in mid-2018!

Please see our website for book and course availability and release information.

Get Your Own Branded Version of This Book!

Please note that this book can be custom branded with your own company and department name!

It's a great way to kick off an internal cube development and deployment project!

As well as letting your customers and clients know that you're really serious about Tabular Cubes!

See our website and/or contact us for more information.

Share This Book!

Also, show people this book also when necessary. Lend them a copy. Have your organization buy some of these books and distribute them to some or all of the people using cubes. I am writing additional variations of this book for managers, executives, and cube developers to specifically address their perspectives and information needs so as to help optimize the development, the use, and the adoption of the cubes throughout companies and organizations.

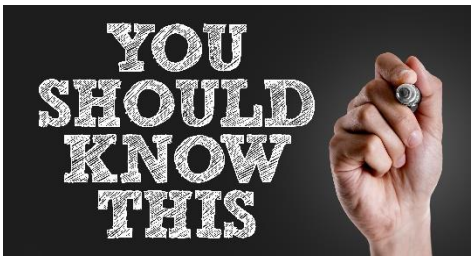
And so...It's up to you now to use your cubes to make your work life easier.

Best Wishes and Happy Cubing!

- Cary Grant Anderson

Chapter 11. Introduction to the Microsoft BI (Business Intelligence) Stack

You're going to hear some new names, terms, and acronyms with Tabular OLAP Cubes. This section will introduce you the Microsoft BI product line, commonly called the Microsoft BI "Stack", as in a stack of tools, technology, and products. Familiarizing yourself with these terms will help you understand when you talk to cube developers and other IT folks that through words like "SSAS", "SSIS", and such around with practiced ease. This chapter will help you make sense of all these terms and where and how the products fit together.



11.1 The Important Things to Know

The important things to know about the Microsoft BI (Business Intelligence) Stack are:

1. It does a lot! Probably virtually everything you need for your Business Intelligence needs.
    a. Except...Perhaps a front end reporting tool like Tableau.
2. It has everything that you need to do both Tabular and Multidimensional OLAP Cubes.
3. It's all built around SQL Server an outstanding relational database with a large following worldwide.
4. You don't need to use all of the various components and features of this BI stack.
    a. You can mix and match and only use what you need.
    b. The other components and features are there when you need to scale.
5. You can also do SQL Server and cubes in the Cloud, particularly Microsoft's Azure cloud.
6. While there are competitors providing other types and brands of OLAP and ROLAP cubes, SQL Server is in mine and other's opinion the best OLAP cube solution from features and functionality as well as performance and price considers.
    a. It is also able to quickly build cubes in the hands of an experienced developer.
    b. I've looked around from time to time and have evaluated some of the competitions claims...And none of them persuade me to leave the Microsoft BI stack.
    c. It's that good!

Also, many third party vendors support SQL Server and OLAP cubes, both Tabular and Multidimensional. These include front end reporting tools and dashboards such as the very popular product Tableau.

All in all, I'm very impressed with the Microsoft BI Stack and don't see any other product or product set that I would like to use instead.

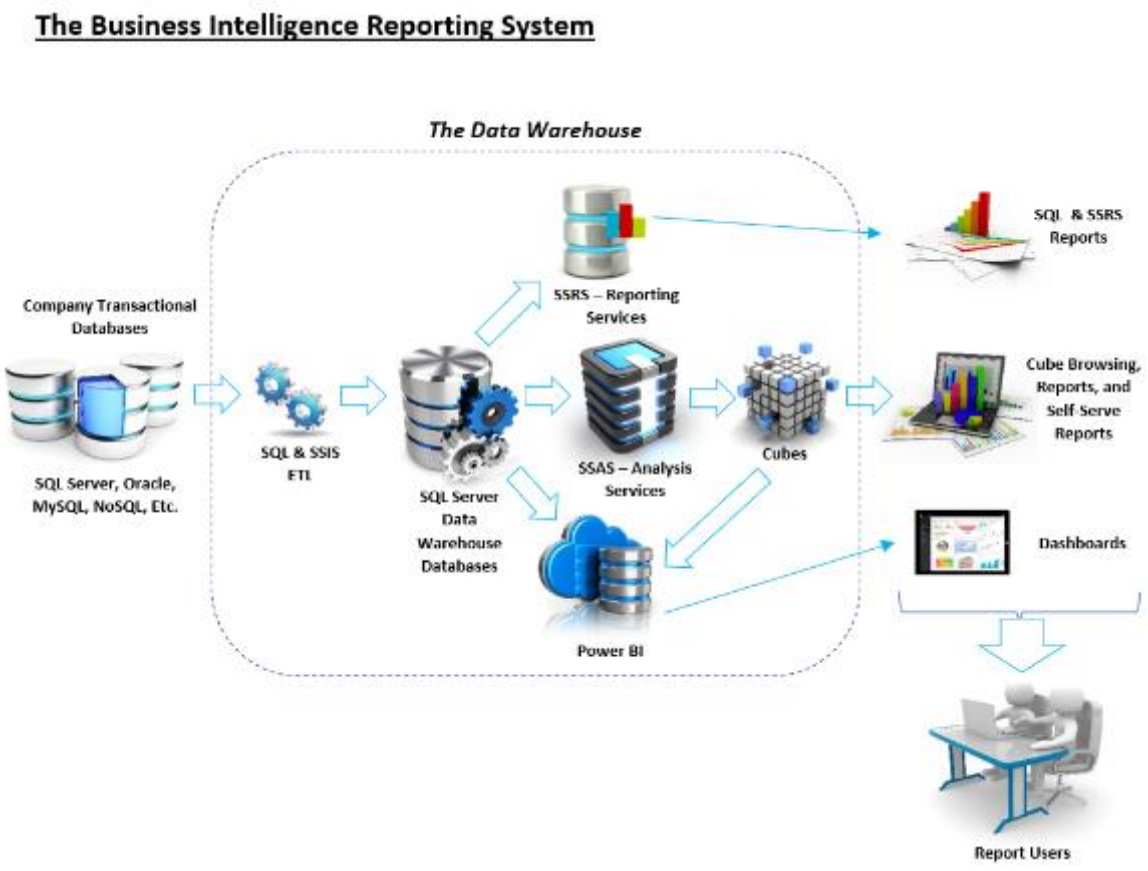
11.2 The Microsoft BI Stack – The Big Picture

The Microsoft BI (Business Intelligence) Stack has the following major components:

1. The Microsoft BI Stack – Big Picture Diagram
2. SQL Server Database
3. SSMS – SQL Server Management Studio
4. SSAS – SQL Server Analysis Services
5. Visual Studio SSDT – SQL Server Data Tools
6. SSRS – SQL Server Reporting Services
7. SSIS – SQL Server Integration Services
8. Excel Pivot Tables
9. Power BI
10. Tools From Other Vendors

We'll explain each of these in summary in the following sections. And also show you where these components and tools fit in the Microsoft BI Stack – The Big Picture Diagram:

The Big Picture Diagram – Microsoft BI Stack



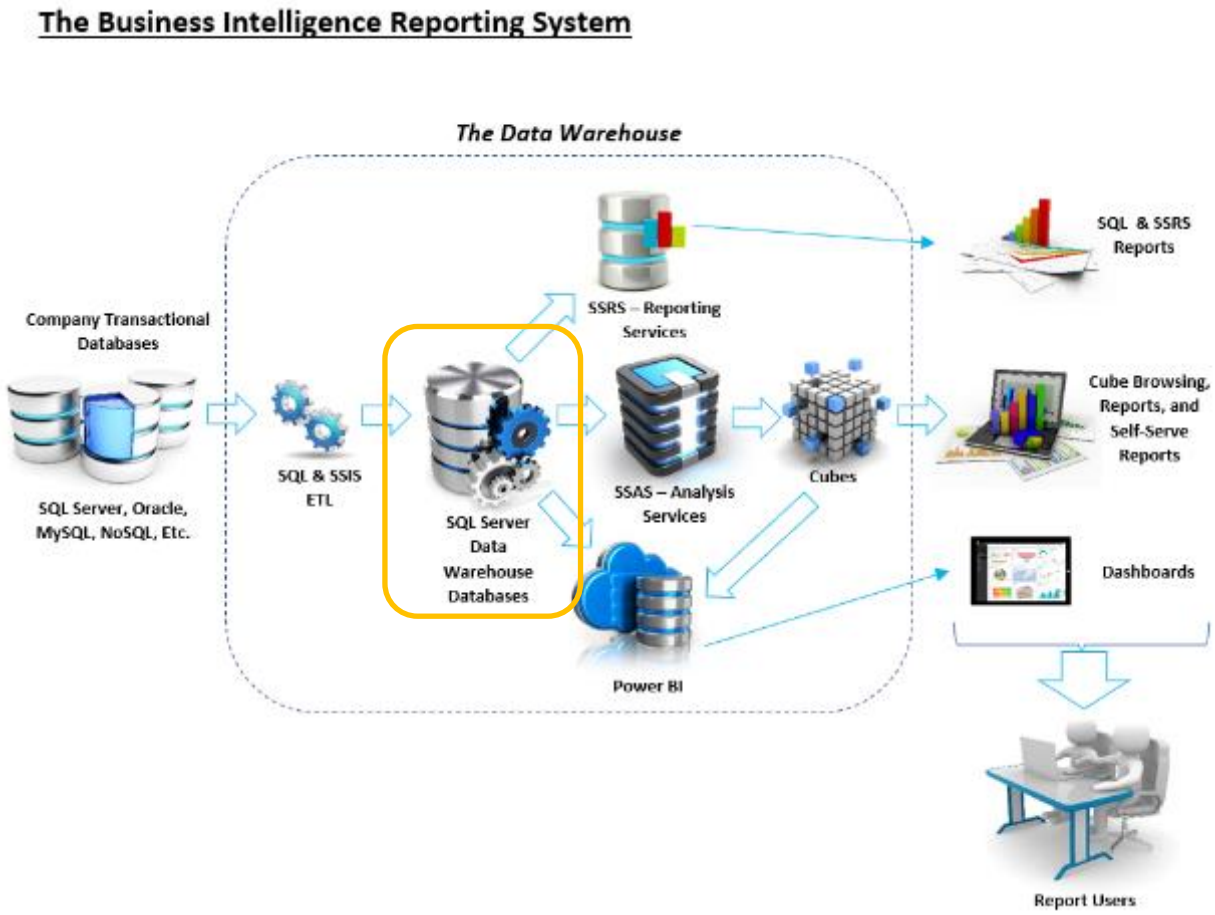
11.3 SQL Server Database

SQL server database is Microsoft's flagship SQL relational database product.

It is an outstanding and easy to use SQL relational database.

It has a large number of tools that connect to it including many front end reporting tools.

SQL Server databases are used all over the world in a variety of database applications. And it's very easy to build Tabular Cubes as well as Multidimensional cubes with it.



SQL Server databases can be found in very small companies and organizations as well as very large and also huge organizations. It scales very well and has a number of versions called editions from a free

community edition to a large enterprise edition. It has a vast number of features yet is fairly easy to manage, especially compared to some other popular databases.

SQL Server 2017 and beyond comes in five editions:

1. Enterprise Edition.
2. Standard Edition.
3. Web Edition.
4. Developer Edition.
5. Express Community Edition.

Enterprise Edition

The Enterprise Edition has all of the many components and features of SQL Server and is the version that is used by most companies for Business Intelligence (BI) reporting systems. There is full support of SSAS and both Tabular and Multidimensional cubes. And full support for massive scaling capabilities and replication and lots of other enterprise level features.

Standard Edition

The Standard Edition is a slightly scaled down version that is less expensive than the Enterprise Edition. Companies use this edition where they do not need all the many features of the Enterprise Edition. The Standard Edition supports SSAS and both Tabular and Multidimensional cubes but there a couple of cube features that it does not support or fully support:

1. Perspectives
2. DirectQuery Storage Mode
3. Semi-Additive Measures
4. Write Back Dimensions
5. Linked Measures and Dimensions
6. Multiple Partitions
7. Proactive Caching
8. Push-Mode Processing
9. Direct Writeback
10. Measure Expressions

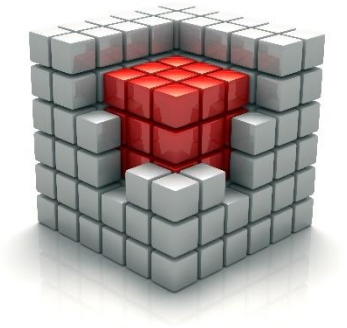
The only ones that one would really or often potentially use of this list with Tabular cubes is perspectives. Perspectives are a way to create different views of a cube which have a subset of the cube objects, measures and dimensions and KPIs, which thus provide a simplified or focused view of a cube. The cube objects that are not present in a particular perspective are essentially hidden and thus not visible. This is highly useful for:

1. Simplifying a complex cube, and

2. Targeting/Focusing a set of cube features and functionality to one or more specific user groups.

For example, let's say that your Tabular cube is a bit big and complex. Sixty two measures and over two dozen dimensions and a handful of KPIs. To make it easier for your users and also to match the cube to three different user groups – Basic Users, XYZ Product Users, and Power Users – You create two perspectives on the cube. One for Basic Users which just has the most important 12 measures and 7 dimensions that they need to use and one for the XYZ Product Users which want to use the Basic measures and dimensions and also the XYZ product type measures and dimensions.

1. Basic Perspective – 12 measure and 7 dimensions.
2. XYZ Product Perspective – 24 measures and 14 dimensions.
3. Full Cube – 62 measures and 27 dimensions.
    a. The full cube is essentially the default perspective.
    b. See Chapter 10 for some information on how Excel users see perspectives when they connect to a Tabular cube.



These two perspectives, combined with the full cube, provide three different views of the cube that nicely match what your three user groups want to see and use.

Using perspectives, you can simplify complex cubes for specific user groups and also for learning the cube as well as providing a targeted focus on specific cube aspects for specific users. Perspectives are very easy to create and use. If you need or want perspectives, however, you're going to need the Enterprise Edition. IF you don't need or plan to use perspectives, or any of the nine other features that are not supported or fully supported in the Standard Edition, then you'll want to use the Standard Edition to save money. Both Tabular and Multidimensional cubes work great on either edition and the only time you'll know that you're not running on Enterprise Edition is when a feature is not available or when you have scale significantly. Elsewise, Standard Edition will work fine for cubes.

Web Edition

The Web Edition is a scaled down version of SQL Server that you can use for web databases and application systems and products on the web. The licensing for this edition is very difficult to understand and I had to do a bunch of research to figure it out. Essentially, it is a much lower cost version of SQL Server that one can use on the web but only for services that are sold to external customers. But not your own internal users. So, if you are offering a web service of some type that uses SQL Server and you sell this service to external customers then you can use the Web Edition. But not if it's just an internal service or application that you provide to users within your organization or company. This edition is most often provided by web hosting companies and they know the license requirements, so they can explain this further if necessary.

Note that the Web Edition is a very scaled down and simplified version of SQL Server and it does not support SSAS or Tabular or Multidimensional cubes. Also, while it does supports some SQL Server components such as SSIS, these components do not have 100% of the features of SQL Server Standard or

Enterprise Editions and thus may be missing some critical components or features that you need or current use.

Developer Edition

Developer edition is essentially a full featured version of SQL Server Enterprise, but it has some very significant limits imposed on it so that you cannot use it as a replacement for SQL Server Enterprise Edition. With the Developer Edition you can do anything that you can do with the Enterprise edition, but this edition can only have 2 active connections at one time. Additionally, you're limited to the number of CPU cores it supports and thus internal performance and capacity. It's meant for learning and development purposes. Every SQL Server and cube develop has a locally installed copy of this edition on their local development PC or laptop.

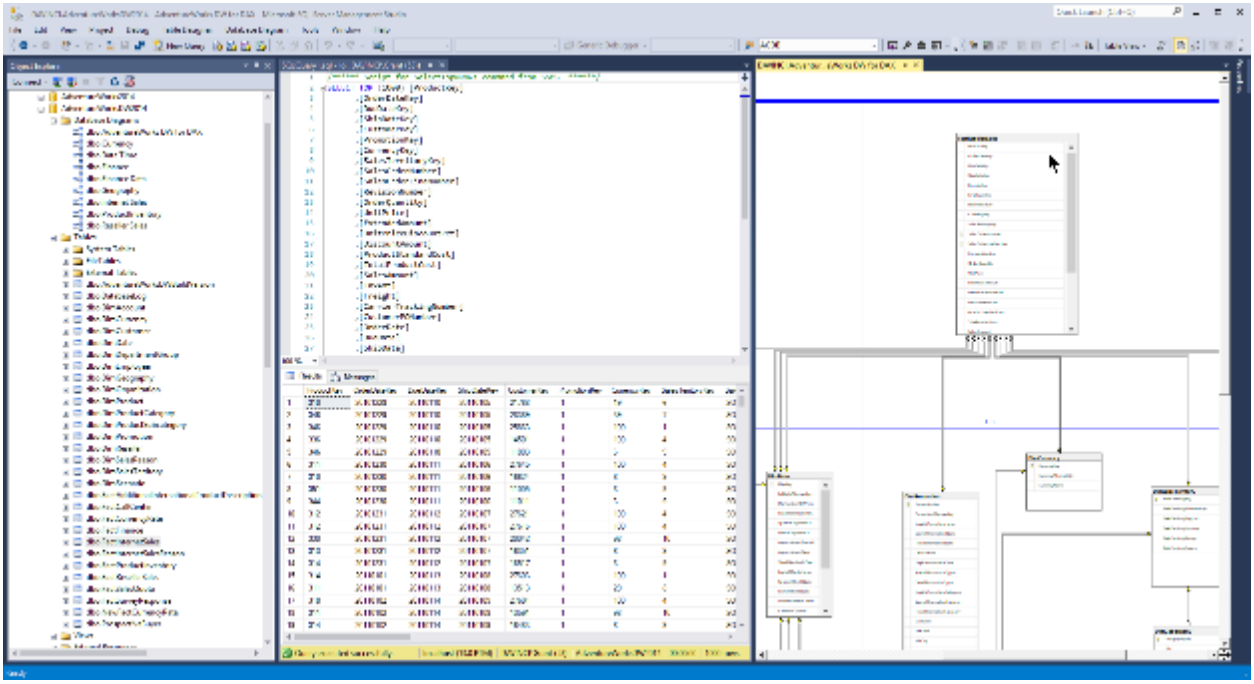
The purpose of the Developers Edition is strictly for development although it can also be used for learning. It's free and can be downloaded from the Microsoft site or purchased for about $60.00 USD for a copy on DVD media. Being meant solely for development purposes, that's why it's limited and it basically offers everything in the Enterprise Edition so that they don't have to offer multiple versions of the Developers Edition. This works really well in practice but there's one important caveat to know when doing cubes development with the Developer Edition – Cube perspectives are supported in Developer and Enterprise Edition but not Standard Edition. So, you have to be careful that if you use cube perspectives (and also a few more less often used features) that when you deploy to your test and production server instances that they are of the Enterprise Edition else your perspectives that you have developed will not be visible or usable.

Express Edition

Express is designed for very small uses it is not meant to scale or handle high loads. It also lacks a lot of features, but it does give you very nice small scale SQL server databases that you can use for both learning purposes and small scale applications and websites. Express edition is free to download and use. You're probably not going to use SQL Server express in a corporate environment. Express Edition does not support either Tabular or Multidimensional cubes.

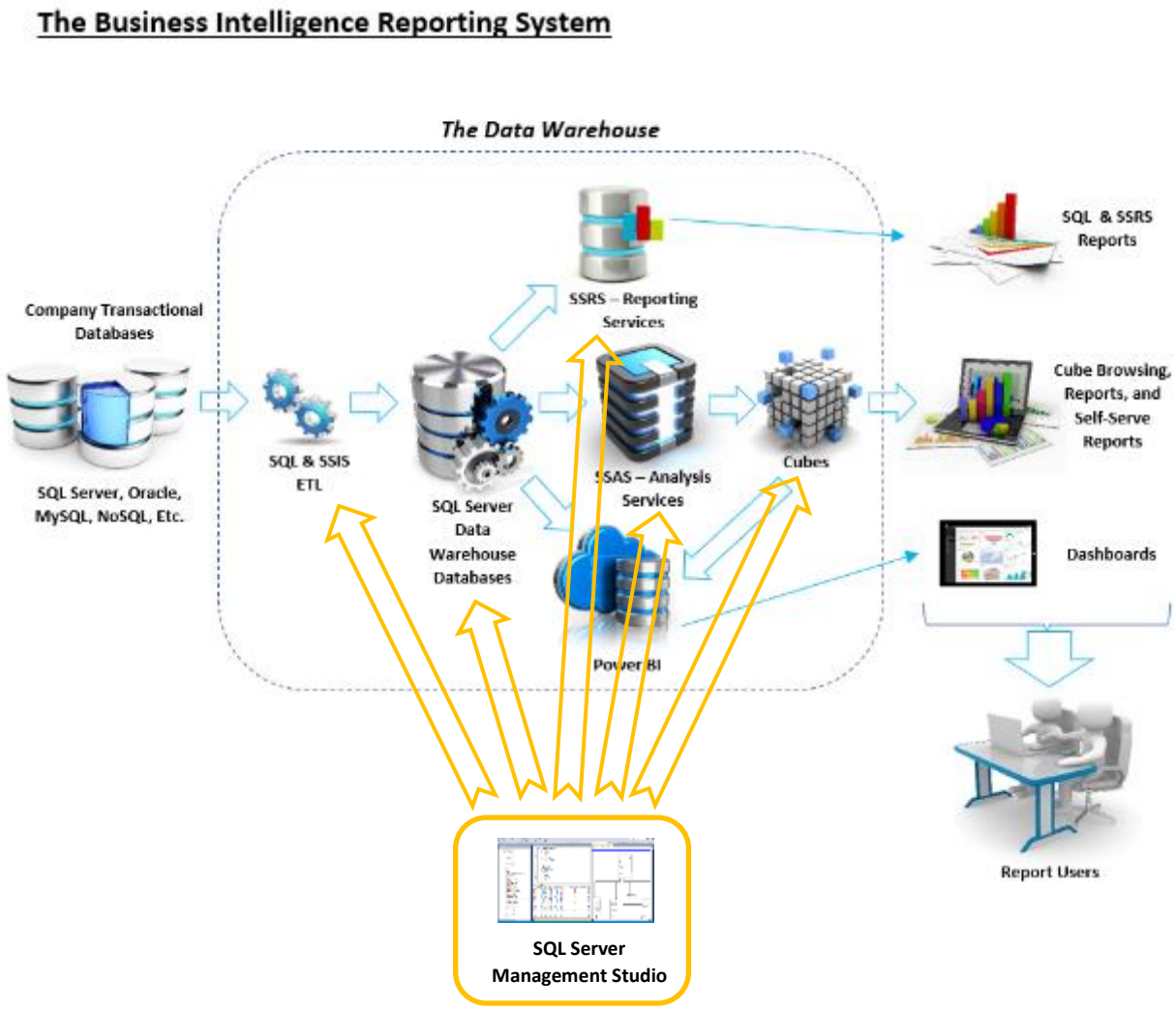
11.4 SSMS – SQL Server Management Studio

SQL Server Management Studio, known as SSMS and just "Management Studio", is the free SQL Server management, query, and development tool for SQL Server. It is free and very feature some and powerful and can be downloaded free from the Microsoft website. While it allows one to develop SQL Server database objects like tables, indexes, stored procedures, views, and so forth, it does not support the building of cubes, either Tabular or Multidimensional. This is because cube development is rather complex and specialized is done using another tool, known as SSDT – SQL Server Data Tools for Visual Studio.



Management Studio does allow you to view and browse cubes, both Tabular and Multidimensional, as well as manage SSAS Analysis Services servers. While there are commercial tools that do the same things as Management Studio, these third party tools are not used all that much in the industry since Management Studio lets you as a developer do whatever you need to do adequately enough.

Generally, report users will not use Management Studio although report developers will. Since Management Studio is free, report power users that know SQL could use it for SQL reporting queries and also to browse cubes. The included cube browser does have some advantages over Excel pivot tables in terms of how they are displayed but lacks all the features and power of Excel pivot tables.



SQL Server Management Studio (SSMS) connects to all areas and server components of SQL Server and lets one manage them all as well as develop database objects. Note, however, that while you can see, browse, manage, and process cubes with SSMS, you cannot actually build cubes with it. Another more specialized tool called SSDT – SQL Server Data Tools for Visual Studio is needed for that.

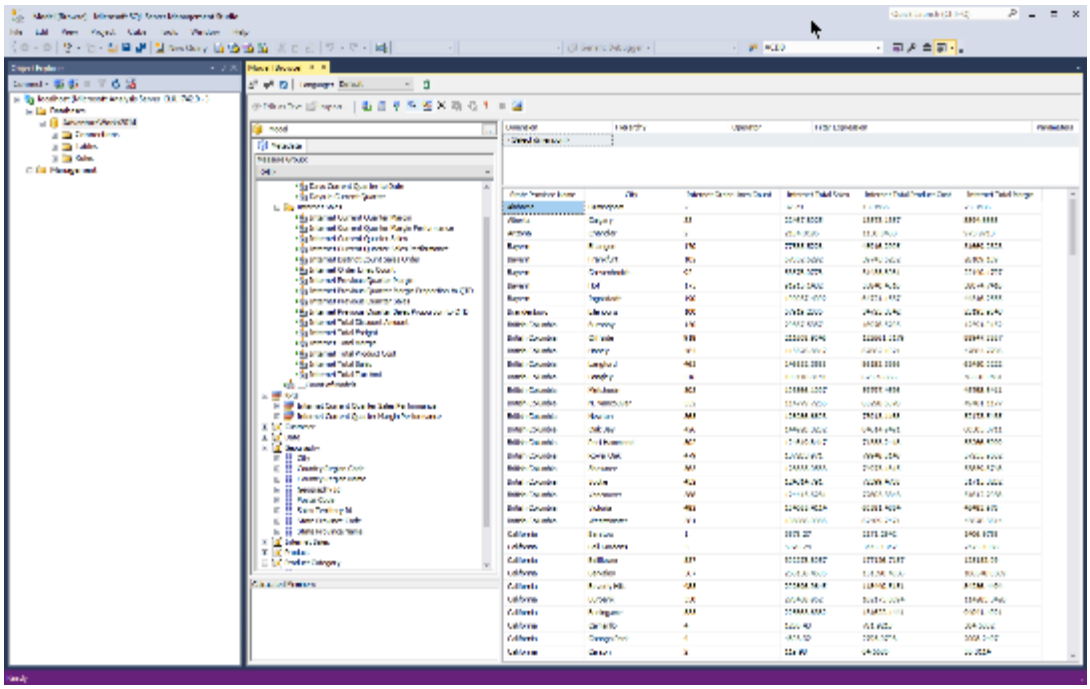
11.5 SSAS - SQL Server Analysis Services – OLAP Cube Server

Now let's talk about SSAS a bit. SSAS stands for SQL Server Analysis Services. People often call it “Analysis Services” or “SSAS” or alternately both in the same conversation. Whichever term it is called it really is a “cube server”. So please just think of it as the Cube Server.

SSAS is a server service, much like SQL Server itself, that runs on a server or cloud server instance and hosts OLAP cubes including Tabular and Multidimensional cubes. It cannot host both at the same time on the same SSAS instance. But a single server or server instance can host both by having two installations of SSAS, a default instance, and a “named instance” that uses a different TCP/IP port than the default one.

SSAS holds the cubes in cube databases which contain the cube and any perspectives for the cubes. No in practice ever really says “cube database” unless they have to. They just say “cube”. This simplifies things but can sometimes confuse people as the “cube database” can refer to either the cube database on the SSAS server or the cube schema database on SQL Server that the cube within the SSAS pulls data from when it is processed. In most cases, people mean the cube schema database.

One of the interesting things to know about SSAs is that if you install it on a developer's laptop or PC, that laptop or PC must have local administrator permissions for that developer user or else the developer is not going to be able to actually access it. Many users in corporate environments do not have local administrator permissions on their laptops or PCs but cube developers require it.



SSMS Showing a Tabular Cube in its Cube Browser on SSAS

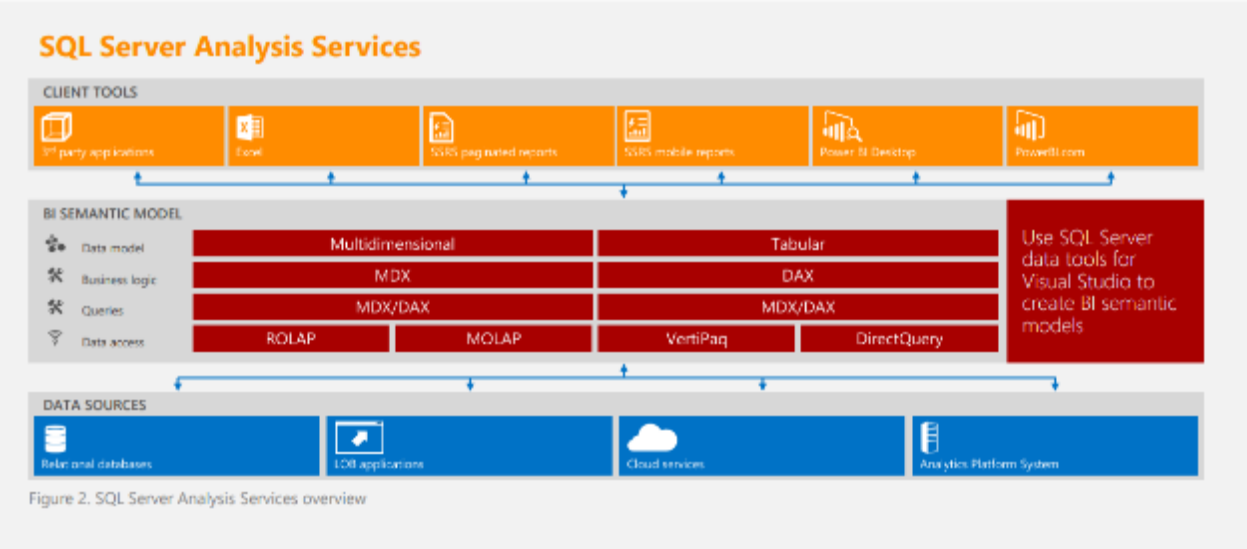
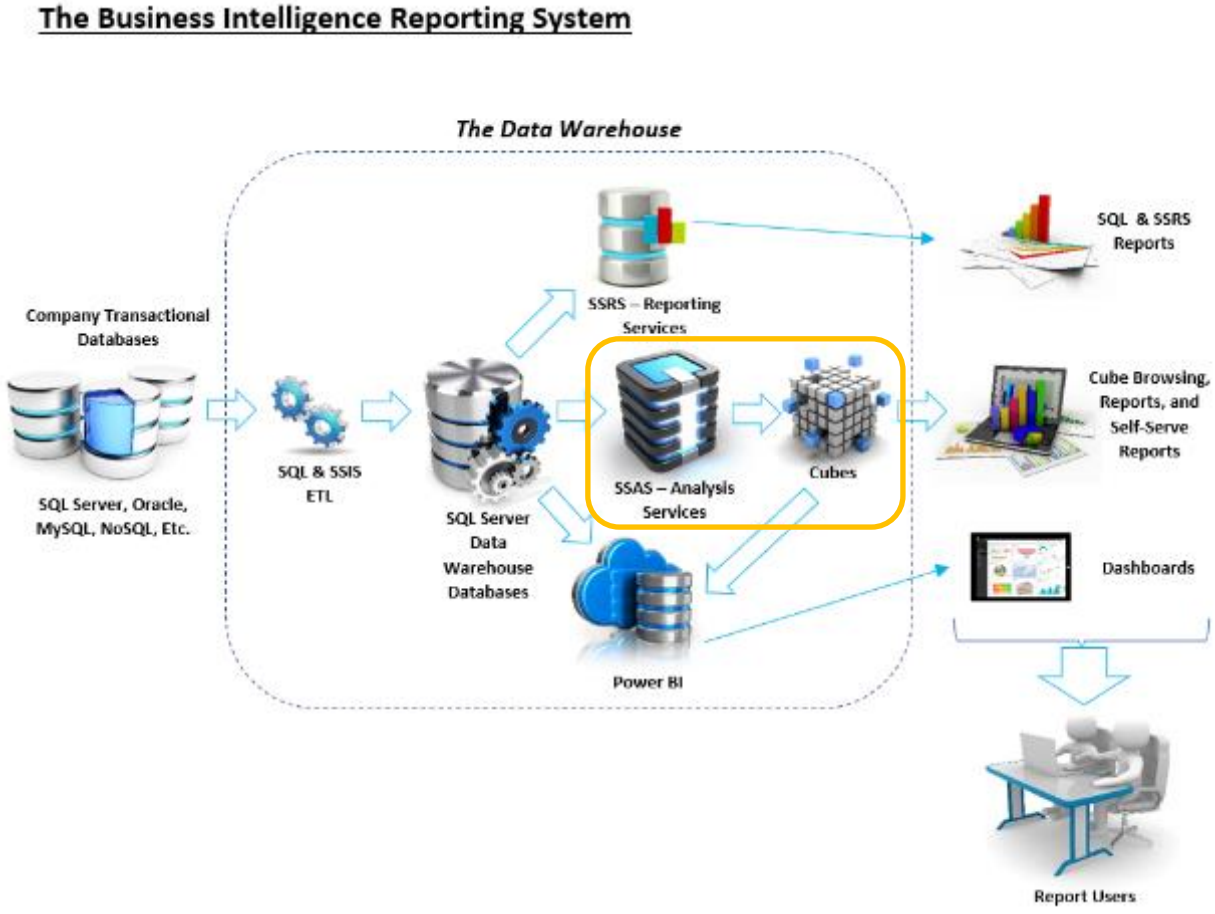


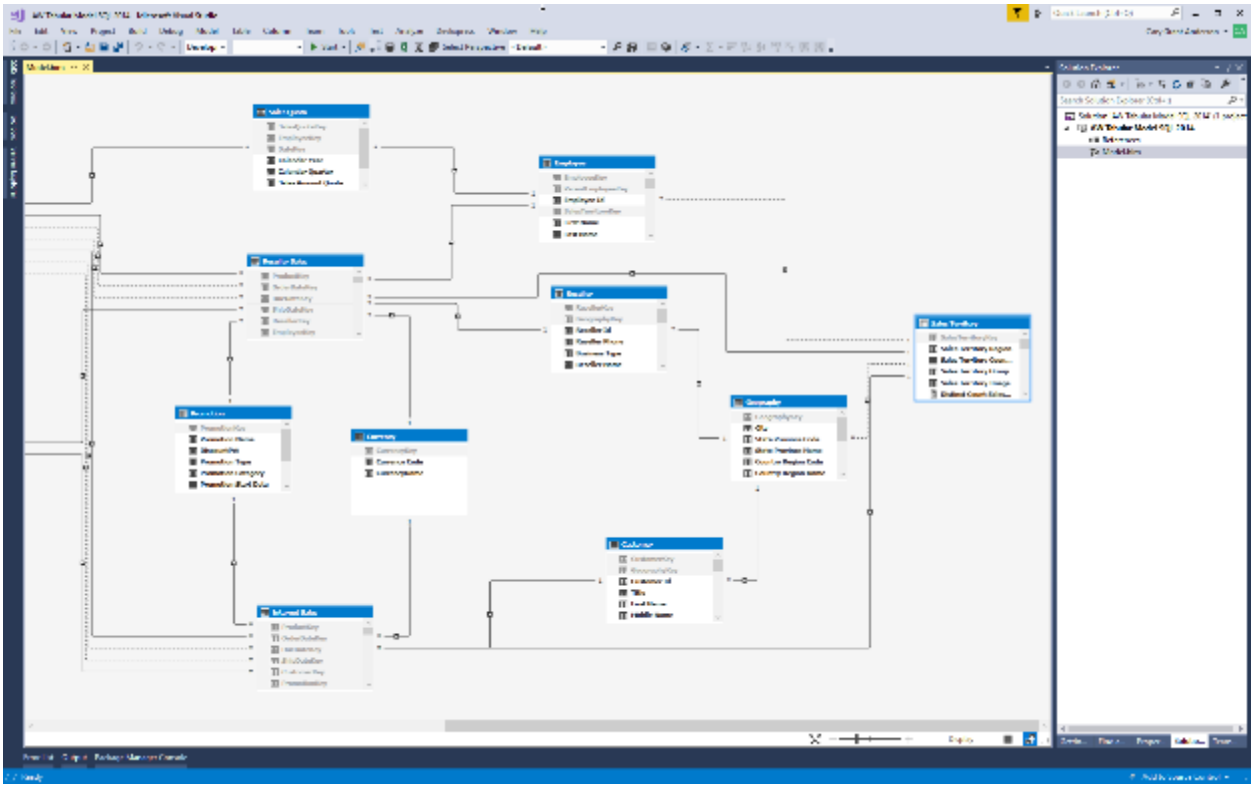
Figure 2. SQL Server Analysis Services overview



Microsoft SSAS Analysis Services is the Tabular and Multidimensional cube hosting server.

11.6 Visual Studio SSDT – Cube Development Environment

Cubes, both Tabular and Multidimensional, are developed using Visual Studio with a plugin extension called SSDT – SQL Server Data Tools. This replaces the previous special version Visual Studio called Business Intelligence Developers Studio also known as BIDS. BIDS was used for many years and was finally wrapped into Visual Studio after SQL Server 2008 R2 was replaced by SQL Server 2012. Visual Studio is Microsoft's flagship GUI IDE (Integrated Development Environment). It is used to build C# programs, web applications, SSAS cubes, SSRS reports, SSIS ETL packages, and just about every Microsoft product.



SSDT is a separate optional download from the Microsoft website. It is free but Visual Studio is not, except for the free Visual Studio Community Edition which is mostly used for learning purposes.

Visual Studio comes in three editions:

1. Professional Edition
2. Enterprise Edition
3. Community Edition

Note that there are also Visual Studio Edition versions for the Mac.

Professional Edition

The Professional Edition is used by most cube, SSIS, software, and other developers. It provides a complete development environment and most features that most developers will need or want.

SSDT can be loaded into Professional Edition and used to develop both Tabular and Multidimensional cubes. This edition is what most corporate and professional developers use.

Enterprise Edition

The Enterprise Edition has some high end features and tools bundled with it including test tools, Xamarin (for mobile development), cross platform tools, and other tools. Enterprise Edition is significantly more expensive than the Professional Edition and few companies use it finding the Standard Edition more economical and adequately feature some.

SSDT can be loaded into Professional Edition and used to develop both Tabular and Multidimensional cubes. This edition is what most corporate and professional developers use.

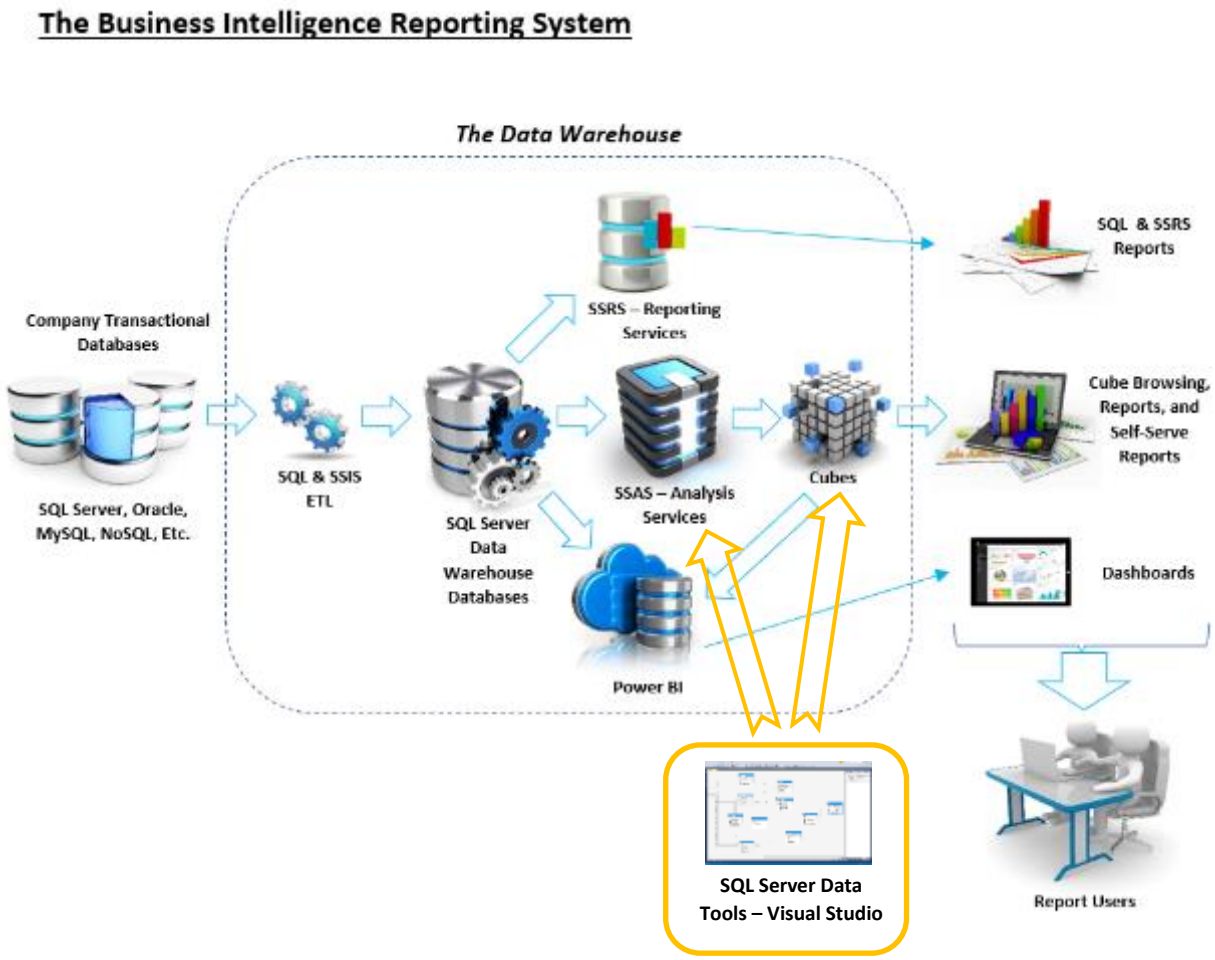
Community Edition

The Community Edition is free and can be downloaded from the Microsoft website. It is primarily targeted for learning and open source type projects. Corporations generally do not use this edition. Community Edition is essentially a fully featured version of Visual Studio Professional with the following basic limitations/restrictions:

1. No TFS (Team Foundation Server) support.
2. No testing functionality support.
3. No Virtual Environments (cloud virtual servers) support.

Many developers either do not use these features or get around them by using GIT and GitHub instead of TFS, an outside test program, and other tools for cloud support.

SSDT can be loaded into it and it can be used to develop both Tabular and Multidimensional cubes. This edition is a good way to start learning cubes.



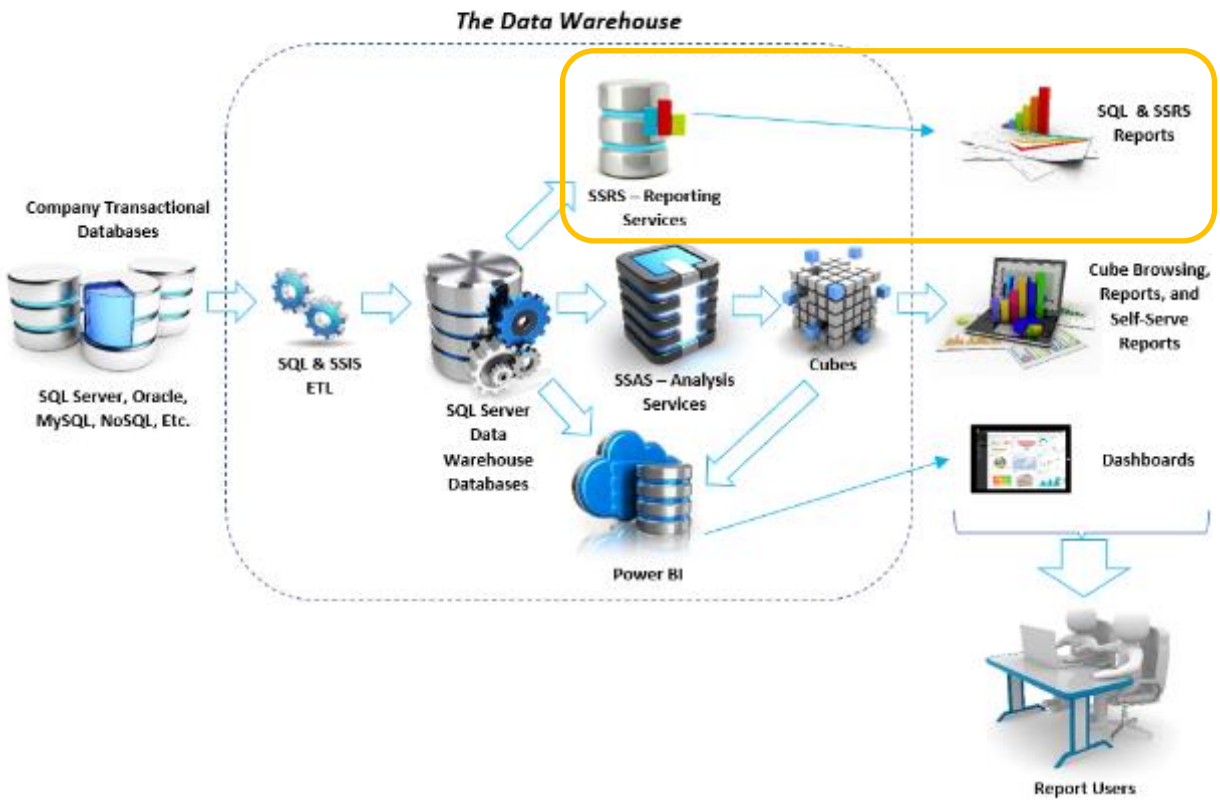
Visual Studio with SQL Server Data Tools (SSDT) is used to develop both Tabular and Multidimensional OLAP cubes.

11.7 SSRS – SQL Server Reporting Services – For Reports

SQL Server also includes a convention report server named SSRS for SQL Server Reporting Services. Reporting Services can be used to develop conventional SQL database reports as well as cube reports because it can access data from both Tabular and Multidimensional OLAP cubes.

SSRS is included with SQL Server and it is in wide spread use. Its reports can run on Windows PCs and laptops, tablets, mobile devices, as well as the web of course.

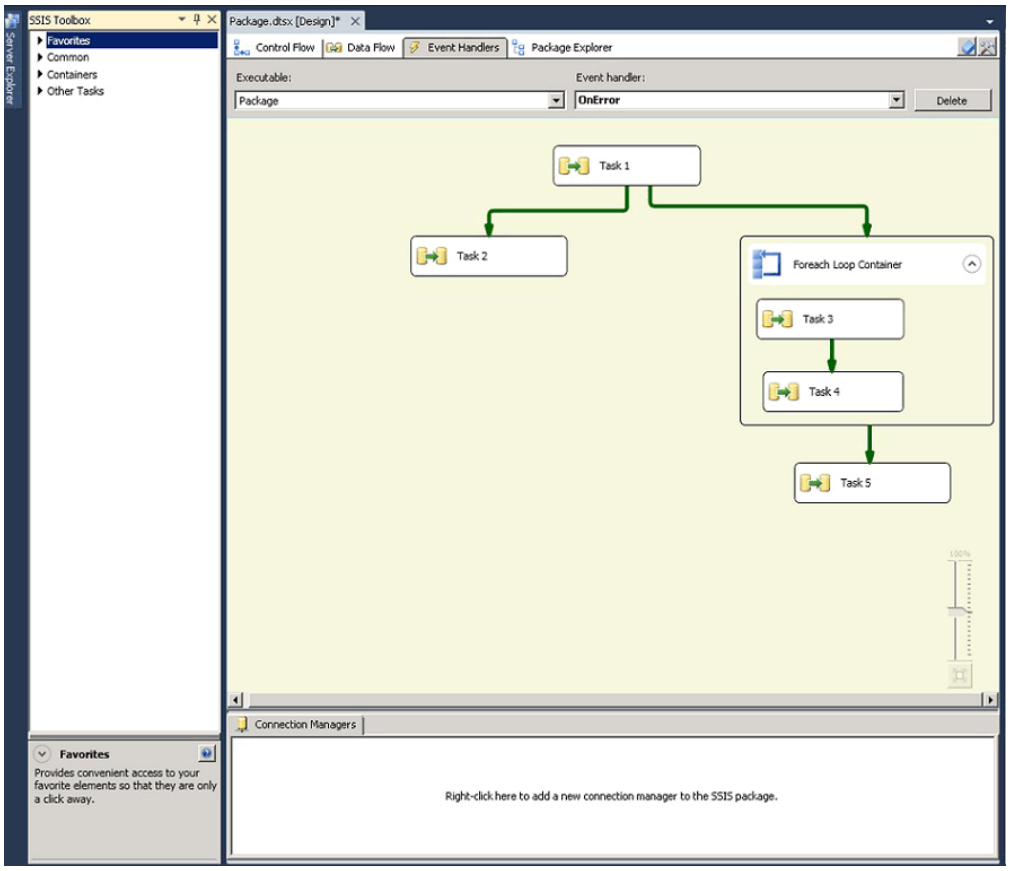
It does not include a cube browser, but it does include a user based report builder.



SSRS Reporting Services provides a ready to use report development that can support SQL as well as both Tabular and Multidimensional cubes.

11.8 SSIS – SQL Server Integration Services - For ETL

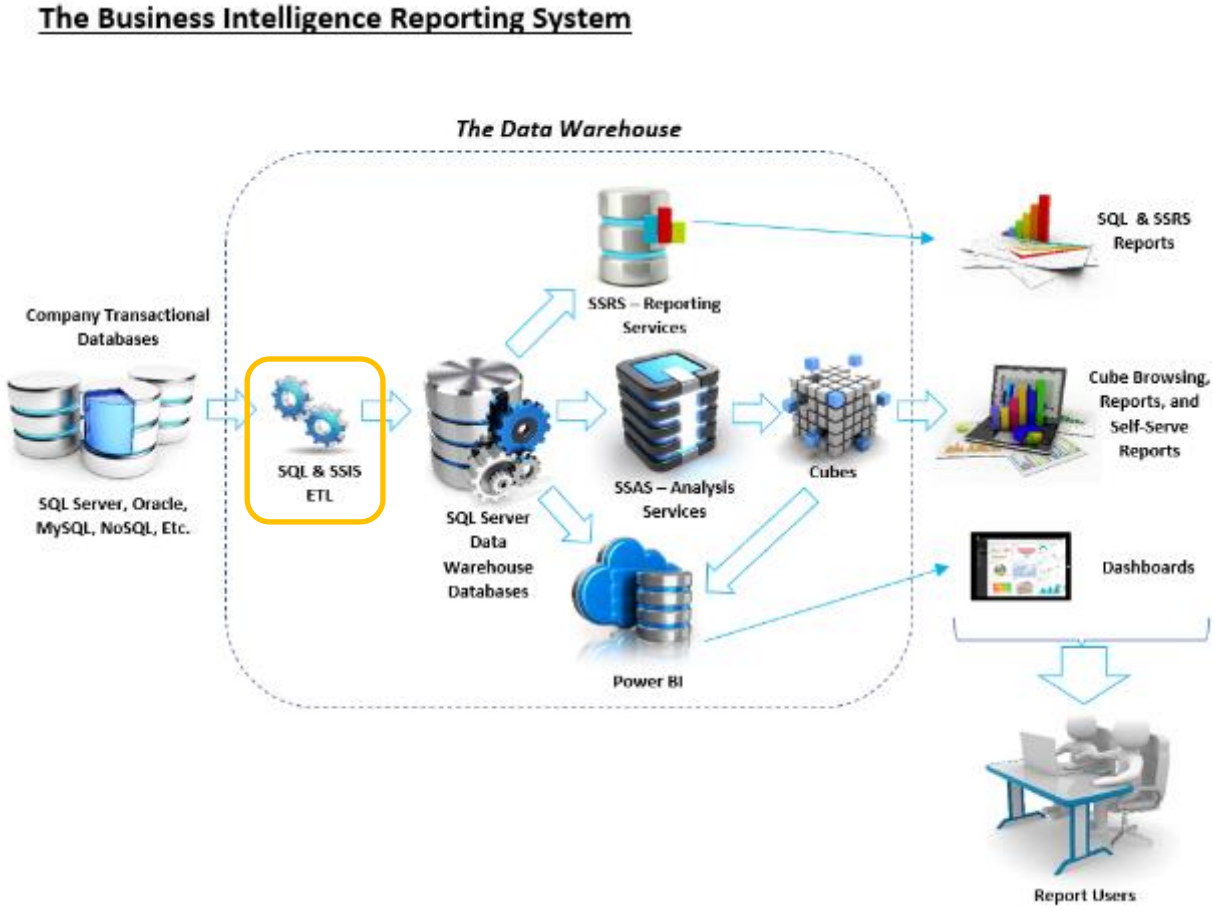
SQL Server Integration Services, also known as SSIS, is a data import/export/transformation server service that is often used for ETL with OLAP cubes. It can read from a multitude of data sources and it has a visual type of interface where you drag and drop various components and then write SQL and other scripts to customize them for specific work processes. It has some quirks and limitations but because it is essentially free with SQL Server it is highly used in the industry. When used for ETL for cubes, Visual Studio with SSDT (SQL Server Data Tools) is used to develop SSIS "packages" which encapsulate the desired data flow and transformation functionality. These SSIS packages are then deployed to a production server with the SSIS server installed and then typically run via scheduling with SQL Server Agent.



The SSIS development interface

SSIS is actually a fairly powerful data transfer and transformation platform that provides for some very useful programming level type functionality when compared to plain old SQL. Often, the programming capabilities of SSIS are not fully used and is just used as essentially an envelope for SQL ETL code. Other

times, SSIS developers will craft involved and complex ETL code in multiple packages. It's a bit of a controversary as to exactly how this is all so much better than just plain old SQL. Microsoft pushes SSIS as a replacement for SQL for ETL and lots of consultants follow this but, in most cases, if your ETL is fairly simple and straightforward there is no real need for SSIS in these scenarios. When data flows and transformations become more complex than the case is made for SSIS. Most corporate environments will use a combination of both to various degrees. A best practice case can be made for keeping things and ETL simple with SQL until the power of SSIS is needed. This helps prevent overuse of SSIS and over complicating ETL.



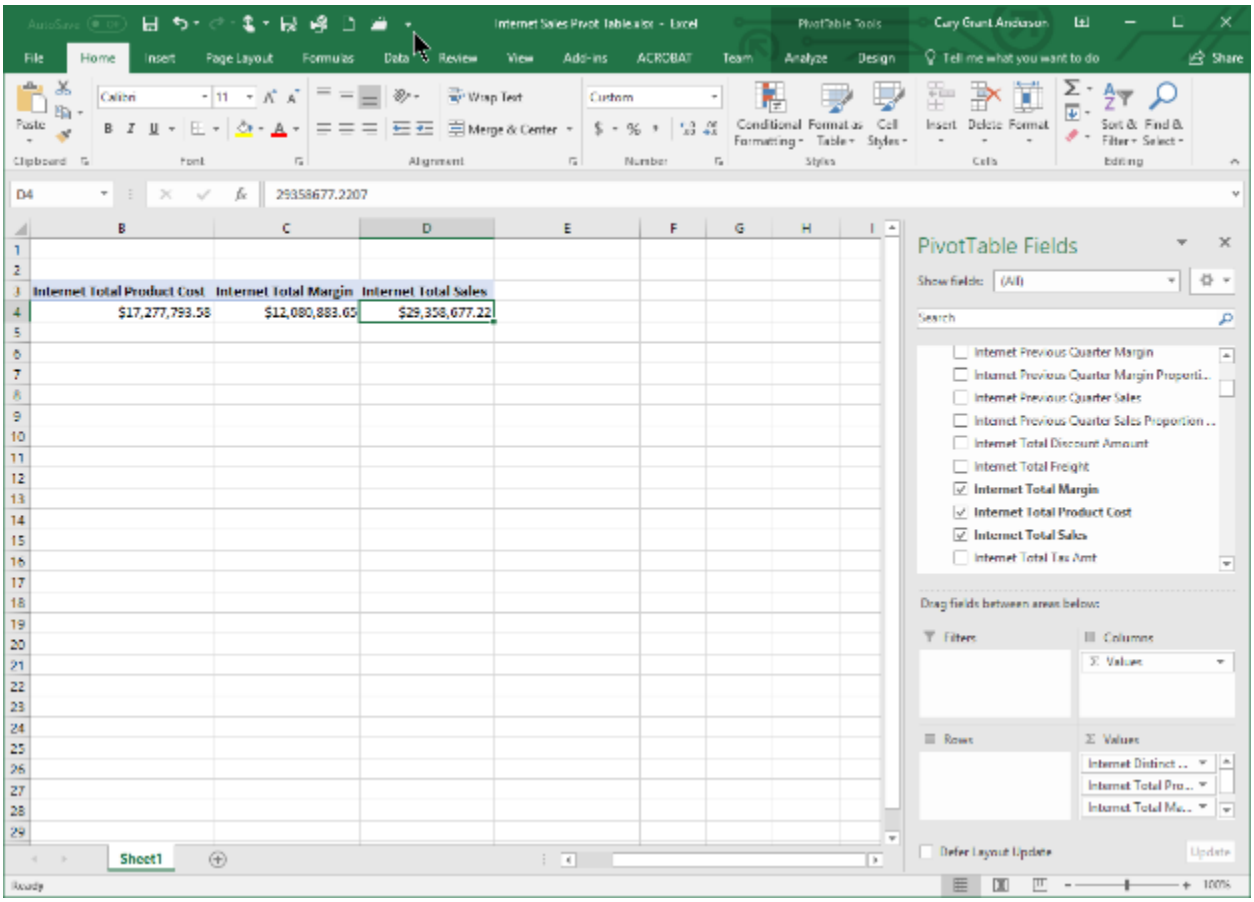
Both SQL and SSIS and a combination of SQL and SSIS are used for most common ETL needs and tasks.

Specialized ETL is sometimes needed. And some caveats exist with SSIS. One of the custom ETL components that I've developed is writing my own C# .NET ETL code to extract data from a NoSQL database, specifically Microsoft's Azure Cosmos DB database using Mongo DB drivers. This is a non-relational database that uses JSON documents instead of conventional database records. I had to set up ETL to extract data from the database and load it into a SQL Server data warehouse where I could cube it.

The Microsoft ODBC drivers did not work properly, and commercial drivers would not work with this NoSQL database as it was on the highly secured Microsoft Azure Government cloud. So, being a C# developer as well as cube developer, I developed a C# program to connect to the Cosmos DB database, find the new records each evening, then extract the data. Since the data was stored in JSON (JavaScript Object Notation) format I had to transform the data into a more relational form and then load it into a set of SQL Server staging tables. From there I used conventional SQL MERGE statements to load the data and any data updates into a SQL Server cube schema star schema and then into a Tabular reporting cube that I also built. I first tried putting the C# code into a SSIS script component that supports custom C# code but could not get it to recognize the Mongo DB driver DLLs. After several days of trying to get this to work I simply modified the code to work as a console type command line program and scheduled its nightly running via SQL Server Agent. Works perfectly! So, the moral of this story is that not everything works as intended or stated. And sometimes you need to get a little creative with your ETL. And also, that having multiple skills, such as general C# coding, is very useful.

11.9 Excel Pivot Tables – For Ad Hoc Self-Serve Reporting

Let's talk about pivot tables for a bit now. Excel pivot tables are not part of SQL Server but are a part of Microsoft Office which most corporate users have on their PCs and laptops. Excel can actually connect to Analysis Services, extract data from both Tabular and Multidimensional cubes, and be used to create Pivot Tables which are highly useful for cube reporting.



An Excel Pivot Table Connected to a Tabular Cube

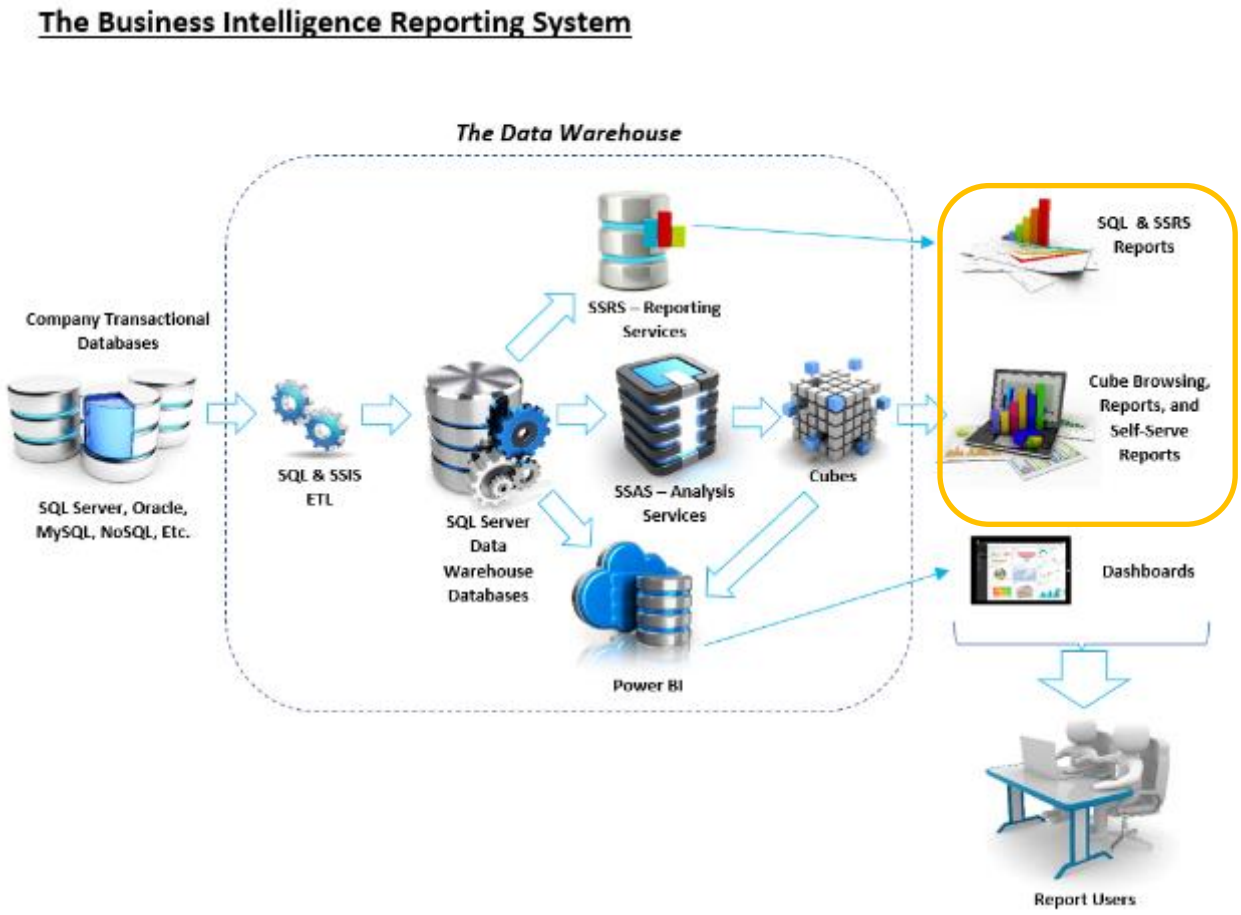
If you want to have a nice inexpensive cube browsing and reporting tool, just use Excel pivot tables. A lot of people already know Excel, but many do not know pivot tables. Pivot tables allow one to "pivot" the data in different ways. In cube terminology, this is known as "slicing and dicing" a cube. The beauty of using Excel pivot tables with cubes is that you already most likely have this front end tool already installed and available, most people know at least a little Excel, and you just have to give your users a basic introduction to both cubes and Excel pivot tables.

This is what I did when I worked at AT&T. I created a new cube and as part of the user introduction and training phase after the cube development phase, I conducted cube introduction and walkthrough training sessions. I then developed two training video screen shot based courses:

1. Introduction to OLAP Cubes
2. Using Excel Pivot Tables with OLAP Cubes

Both courses were well received by the users and were added to AT&T University's course catalog.

As a cube best practice, I recommend user introductions and walkthroughs, and recording these or creating simple video courses really help users and new user come up to speed quickly with cubes.



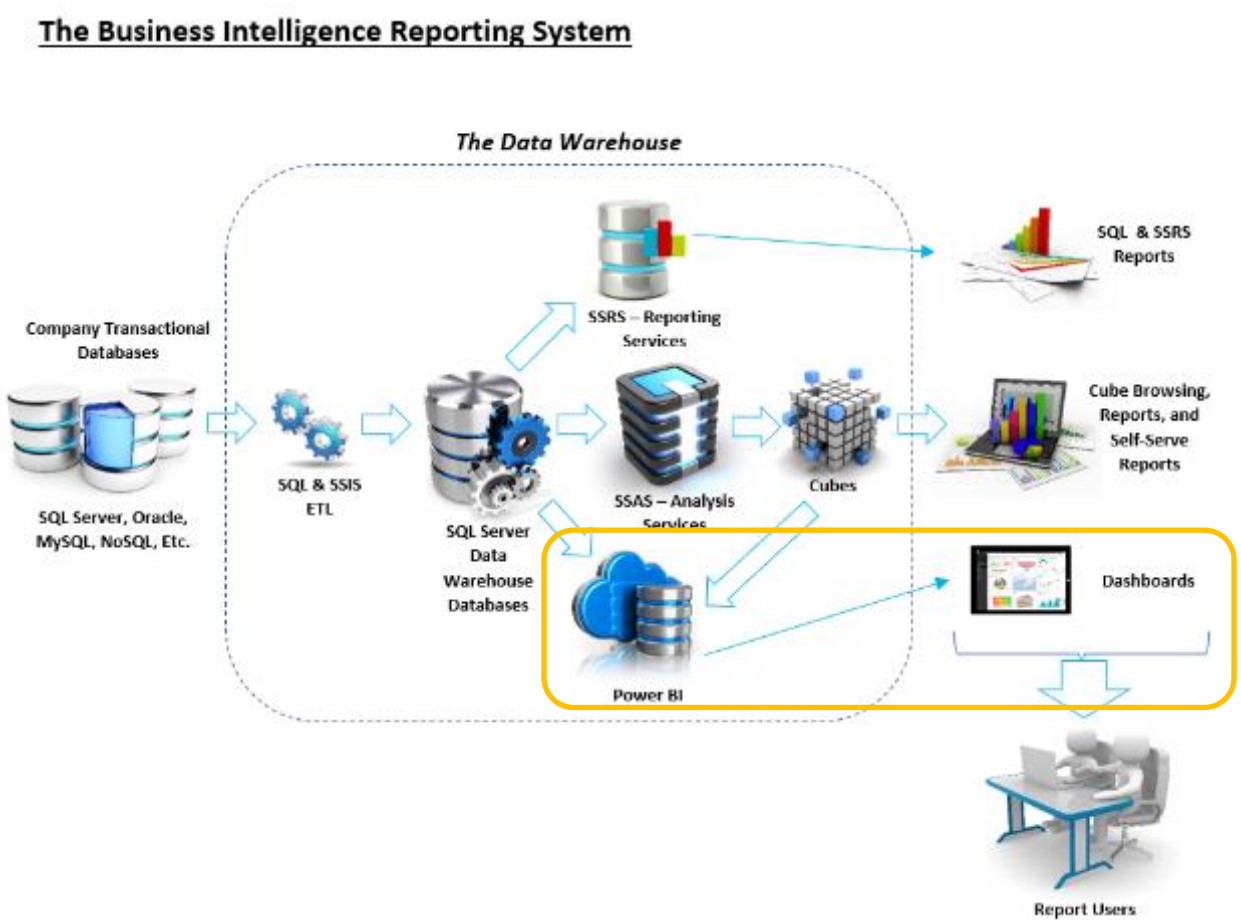
Excel makes for a readymade cube browsing and reporting front end for both Tabular and Multidimensional OLAP cubes.

11.10 Power BI

Power BI is a dashboard product from Microsoft that you can use to create colorful, powerful, and useful dashboards from both SQL databases and also Tabular and Multidimensional cubes. Power BI Desktop is free, and you can create individual dashboard with it. Upgrade to Power BI Pro and share your dashboards with others and gain other nice features such as collaboration as well.

Power BI Desktop

Most dashboard products are very expensive, and this is one way that you can quickly and easily get started with dashboards for your cubes.



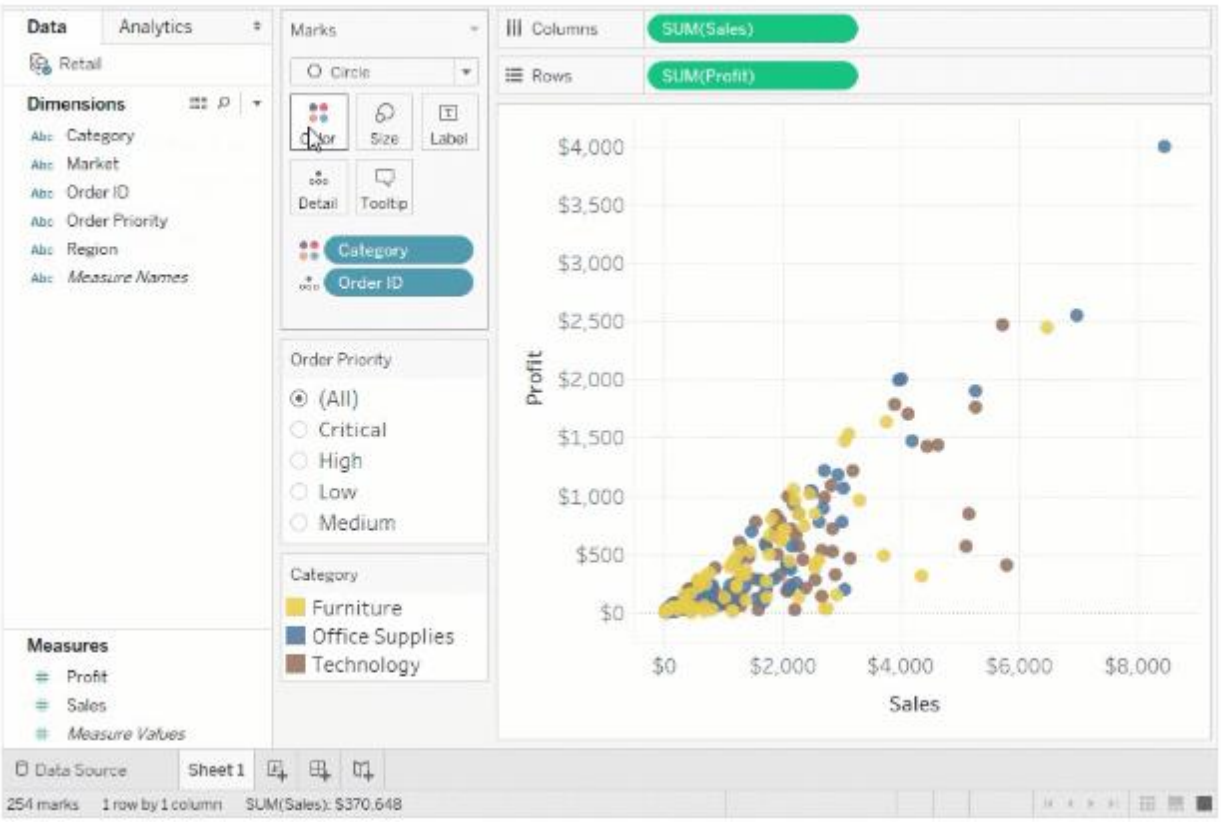
Power BI provides a powerful and low cost way to get dashboards for your cubes.

11.11 Tools from Other Vendors – Tableau, Etc.

There are a lot of other front end reporting tool and visualization vendors. Dozens if not hundreds of them in fact. Sometimes you'll want and need a bit more features and power and visualizations than what you get with the standard tools and Excel. This is where tools from other vendors come in. Many of them connect to and can be used with Microsoft's Tabular and Multidimensional cubes.

Once such very popular front end tool is Tableau.

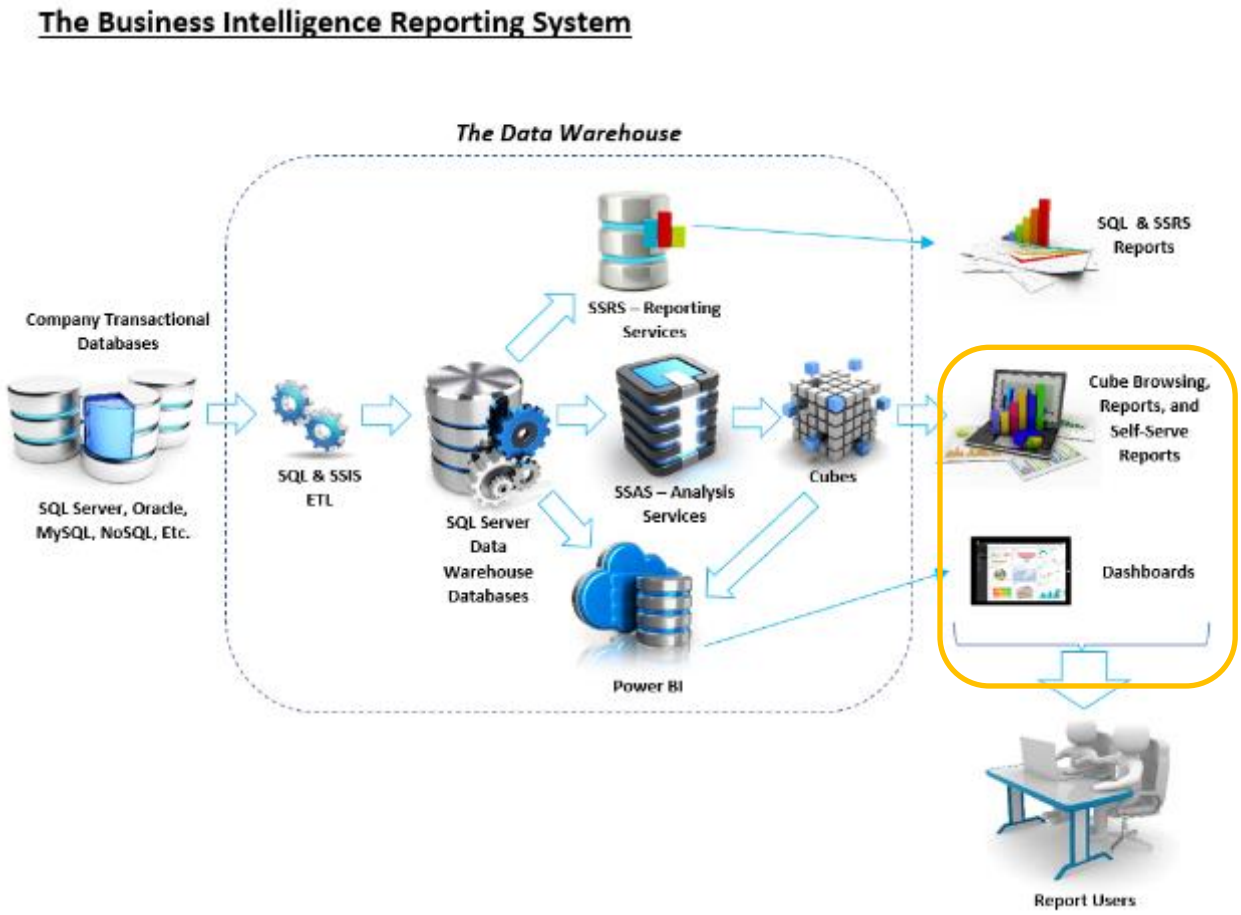
I've used it in several instances and company engagements and tableau is extremely popular and well-liked by users. It creates very appealing and powerful visualizations.



There are different versions of Tableau. There's an online version that cannot use cubes at this point in time but hopefully will in the future. There is Tableau Desktop version which is a Windows application and creates Tableau reports and dashboards. These can be upload to Tableau Server and shared with people throughout your organization.

Note that while Tableau Desktop and Server support both Tabular and Multidimensional cubes there is a learning curve associated with both and also at the time of this writing some small limitations. See the Tableau website for more specific and current information.

The bottom line is that there are plenty of front end reporting tools that can be used with cubes and enhance your reporting and dashboard capabilities and data visualizations.



Third party front end reporting tools can connect to both Tabular and Multidimensional cubes for browsing, reporting, and dashboards.

Chapter 12. The Tabular OLAP Cube Terms Mini Dictionary

Use this chapter as a reference dictionary for common Tabular and some Multidimensional OLAP cube terms. Or read it all through and you'll know all of the common cube terms that you'll hear people and cube developers speak.

Contents:

12.1 Common Cube Terms

Common Cube Terms		
#	**Term**	**Explanation**
1	Business Intelligence - BI	A term for the entire range of a variety of approaches, products, and activities to build data warehouses, reporting systems, and often cubes so as to extract as much useful data ("Intelligence") from business data for the business.
2	Cube	The Multidimensional database hosted by the Analysis Services of the Microsoft SQL Server database. This is the database that the T&E web client accesses and displays data from.
3	DAX – Data Analysis Expressions	The programming language for Microsoft Tabular Cubes. Very different from MDX.
4	Dimension	A dimension is a formal descriptor of the records in a Fact Table. A fact table having records of orders would most likely have a set of descriptive values such as product type, product name, etc. These descriptive values are dimensions of the fact table.

Common Cube Terms		
#	Term	Explanation
5	Dimensional Model	A database or cube model based upon fact tables and dimension tables. Also called a "Dimensional Schema".
6	Fact	A Fact is essentially a "fact" or a definitive numeric value about some object, some thing. Facts are found in the Fact Table. A fact in a cube is implied to be numeric.
7	KPI – Key Performance Indicator	A business metric that can be displayed in a single form with a good/acceptable level, a warning level, and a bad level. It is most often displayed in a "Traffic Light" type form where green = good, yellow = warning, and red = bad. Very useful to see status "at a glance" and often used in dashboards.
8	MDX – Multidimensional eXpression	The programming language for Microsoft Multidimensional cubes. Very different from DAX.
9	Measure	Measures are the central values of a cube that are analyzed. They are typically numeric data (numbers) that are added. Examples are sales dollar amounts, number of units sold, costs, etc. Measures can be calculated values such as an average gross amount. The T&E cube currently supports three measures: Claim gross amount, count of claims, and average gross amount (which is a calculated measure).
10	Mosha	A very popular Microsoft employee that was on the SSAS product team and blogged extensively about SSAS and Multidimensional cubes. A great learning resource while it lasted.
11	Multidimensional	Having more than one dimension. Commonly used to refer to Multidimensional cubes. Also applies of course to Tabular Cubes which are just a simplified form of Multidimensional cubes.
12	Multidimensional Cube	An OLAP cube. Specifically, here for this book, a type of OLAP cube supported by Microsoft's SSAS (SQL Server Analysis Services). One of two types of cubes it supports. The other type is the Tabular Cubes. Multidimensional Cubes are the first generation of SSAS cubes and Tabular Cubes are the second generation. While very similar in many ways there are very significant differences.
13	Multidimensional Database	A database having more than just two dimensions. A typical relational database uses two dimensional tables much like a spreadsheet. The Multidimensional database adds additional dimensions allowing data to be accessed in a variety of useful reporting combinations.

Common Cube Terms		
#	Term	Explanation
14	OLAP	On-Line Analytical Processing. OLAP is for reporting and contains aggregations (summations) of regular OLTP transactional data.
15	OLTP	On-Line Transactional Processing. OLTP is what is done in most relational databases. Transaction records are added and stored. There is no aggregating or levels of data.
16	Slice and Dice	The ability to easily do custom queries against a cube (or even a database) via a GUI that allows the users to select data values then select dimensions to get reduce the data display down to the sections of data (slices) that are useful or of interest. A slice of a cube corresponds to a flat plane or planes use to isolate data of interest.
17	Tabular / Tabular Cube	A second-generation Multidimensional cube supported by Microsoft's SSAS (SQL Server Analysis Services) OLAP Cube server. Very similar in many ways with the first-generation Multidimensional cubes but there are also very significant differences and limitations.

12.2 Grant Specific Cube Terms

Grant Specific Cube Terms		
#	Term	Explanation
1	Aggregation Cube	A cube built from high level, aggregated data as opposed to one built from low level data. See Atomic Level Cube.
2	Analysis Cube	A cube designed specifically for analysis instead of one designed specifically for reporting. An Analysis Cube typically has low level detailed data in it so that users can drill down to the deepest levels of data. Contrast this a Report Cube that is designed specifically for reporting purposes.

Grant Specific Cube Terms		
#	**Term**	**Explanation**
3	Atomic Level Cube	A cube built from low level data as opposed to one built from high level, aggregated data. Also called a Base Level Cube. See Aggregation Cube, Base Level Cube.
4	Base Level Cube	A cube built from low level data as opposed to one built from high level, aggregated data. Also called an Atomic Level Cube. See Aggregation Cube, Atomic Level Cube.
5	Comparison Cube / Validation Cube	A cube specifically built to compare two sets of data such as when reconciling financial data or workflow data. Also, useful to compare and monitor data flowing into a cube from multiple data sources so as to determine missing data. Also called a Validation Cube.
6	Dashboard Cube	A small cube designed to be fast and very responsible so as to drive dashboards with very high performance.
7	Happy Cube	An OLAP cube that greatly meets the users' data and ease of use needs. The cube is happy when the users are happy!
8	Huge Dimension	A dimension with a large number of members, typically into the thousands or tens of thousands of members. It will typically be very difficult and cumbersome for users to use as they have to scroll through thousands of entries to find the ones they want. Huge Dimensions are typically avoided or made more usable by being organized/categorized by a parent hierarchal dimension. An example of a huge dimension is one composed of all the employees of a large company (say 10,000 or more employees). The dimension will have 10,000+ members which make it hard to find any one specific employee. Categorizing the employees by the letters of the alphabet (A – Z) in a parent hierarchal dimension allows it to be easier to use in alphabetical groups.
9	Reporting Cube / Report Cube	A cube designed specifically for reporting instead of one designed specifically for analysis. A Report Cube typically has a combined or limited data set and may have pre-aggregated values (as in an Aggregation Cube). Contrast this an Analysis Cube that is designed specifically for analysis purposes.

Grant Specific Cube Terms		
#	**Term**	**Explanation**
10	User Data Technology Unity	The objective in every cube project is to bring users and data together through the appropriate and skillful use and application of technology. Image a Venn diagram of three circles: Users – Technology – Data. When all three circles mostly or fully overlap then there is near unity or unity, i.e. users can use technology to access all the data they need to fulfill their data reporting and analysis needs. This is then a successful and "Happy" cube. When there is a great non-overlapping/coverage between users, technology, and data then the cube is not successful, and the users are not happy.

12.3 Cube Data Terms

Cube Data Terms		
#	**Term**	**Explanation**
1	Action	An end-user-initiated operation upon a selected cube or portion of a cube. The operation can launch an application with the selected item as a parameter or retrieve
2	Aggregation	From the Latin term 'aggregare' meaning to collect in a flock. As a verb it refers to the process of combining two or more data items into a single item. Summing a series of numbers is a typical example of aggregating. As a noun it refers to any data item that is the result of an aggregation process, such as a total or average. A summation of some type of lower level data into a higher-level value. Aggregation types include summation, minimum, maximum, count and others. In a cube you often find a count of transactions, the sum of the numeric values, and the minimum and maximum values over a range of dates or other dimensional values (or combinations of dimensional values) in a cube.

Cube Data Terms		
#	**Term**	**Explanation**
3	Aggregation Level	The level of summation or summarization involved in data. Example: 1. Base/Atomic Level Data would be a listing of the people in a state (of the USA). 2. One Aggregation Level would be a summation (total number) of the people in the counties in the states. 3. Another higher Aggregation Level would be a summation (total number) of people in each state. The highest Aggregation Level would be the number of people in all the states living in the country.
4	All Level	(All) Level is the optional highest level of a dimension. The (All) Level contains a single member that is the summary of all members
5	All Member	The single member of (All) Level. By default, the name of the All member is All followed by a space and dimension name.
6	Ancestor	Within a hierarchy, an ancestor of a member is found along the path between the member and the root (i.e. parent, grandparent, etc.). Another less use term for this is “Antecedent”.
7	Attribute	A descriptive value of a larger object. Such as the weight of a product or the city of a state. Typically, it is not a numeric value but rather a descriptive value. Found in fact tables as well as dimension tables.
8	Axis	A set of tuples. Each tuple is a vector of members. A set of axes defines the coordinates of a multidimensional data set.
9	Calculated Measure	A measure that is not directly read from the fact table but instead is calculated from one or more fact table measures. Different from an aggregation.
10	Cell Set	The internal name for the value (cell) at the intersection of the measures and one of more dimensions and/or filters in a cube.
11	Child	In a hierarchy with distinct leaves and roots, a child of a member is any member one hierarchical unit towards the leaves from the given member.
12	Date Dimension	A dimension composed of dates. Typically found in every cube because all data has a date (time) aspect to it. This is most often a Static Dimension being loaded once for the next year or years in the future and then not updated or reloaded. It is often also a Shared or Conformed Dimension which is used by multiple fact tables across multiple cubes. See also Time Dimension and Granularity.

Cube Data Terms		
#	Term	Explanation
13	Descendent	Within a hierarchy, a given member's descendant's include any members found along the path between the member and the hierarchy's leaves.
14	Degenerate Dimension	A degenerate dimension is not really a dimension but an attribute of a fact table which does not connect to a dimension table but has some usefulness in being descriptive and generally some usefulness in being a sub-key for the data. For example, an order record could have a transaction number that isn't a dimension but is descriptive and potentially useful as a sub-key. Another form of this term is "Fact Dimension" (see).
15	Dense Cube	A dense cube is a cube with high density of data, i.e. no empty fact table columns and thus empty cells in the cube. The opposite of a Sparse Cube.
16	Dimension	A dimension is a descriptive object/thing/non-numeric value for the data in the Fact Table by which the cube can be "sliced and diced" (filtered) to see specific combinations and scenarios of data.
17	Dynamic Dimension	A dimension which is loaded with data each time the fact table is loaded. The opposite of a Static Dimension.
18	Fact	A numeric value associate with a data object of some type, most often a transaction or record. For example: The numeric weight of a product to ship or the dollar value of the product are Facts.
19	Fact Dimension	Another term for Degenerate Dimension. See Degenerate Dimension.
20	Factless Fact Table	A fact table that has no measures. It still can be used to make a cube that uses the count of records. So, a cube of transactions that have no monetary or other numeric value can be done with just counting the records. Not really a typical use of a cube but there are application scenarios where they are found and used. Generally, a cube will always have at least one fact.
21	Folder	Measures in both Tabular and Multidimensional Cubes can be organized into folders of similar measures. Very useful for organizing a lot of measures and also little used measures. Improves usability of the cube and is considered a Best Practice for cubes.
22	Junk Dimension	A term for a dimension that contains attributes that do not really belong in a fact table or dimension tables. A text column of Notes is a good example. It's not a number or a descriptor so it does not belong in either the fact table or a dimension table. It should not be in a cube.

Cube Data Terms		
#	Term	Explanation
23	Granularity	The granularity of a cube is the smallest level/value of data that the cube supports. Most often this refers to the Date Dimension of a cube and is typically a single day. A cube with a granularity of hours allows data to be reported on and analyzed at a finer/higher resolution, i.e. to the hour. It also increases the amount of by 23 times (24 hours per day instead of 1 per day, i.e. midnight). Note that cubes are designed and built with a specific granularity, most often 1 day (daily granularity). Changing the granularity of a cube is not something easily done. It will require significant refactoring (redoing) of many parts of the cube and the underlaying data model as well. Make sure you understand the desired granularity of a cube before you build it! See also Date Dimension and Time Dimension.
24	Hidden Dimension / Measure / etc.	A cube object such as a measure, a dimension, or an attribute that is not visible , i.e. hidden. This is useful to hide objects that are not yet ready for production visibility. Note that using Perspectives, cube objects can be essentially “hidden” by not including them in a particular Perspective.
25	Hierarchy	A list in which a parent child relationship exists between the list members. For example, a Date Dimension will often have several hierarchies for the various level groupings of dates such as Year-Month-Date or Year-Week-Date. All hierarchies are multi-level.
26	HOLAP	A hybrid cube approach that combines MOLAP and ROLAP.
27	Latency	The amount of time it takes for something. When applied to cubes, it most often means the amount of time that it takes for data to be received. High latency causes problems in cube with Late Arriving data for Fact Tables, Dimension Tables, or both.
28	Late Arriving	A record that arrives significantly later in time than it should and thus can cause complications with dimensions and relational integrity. For example,
29	Leaf	Any bottom-most member of a hierarchy (a member without children).
30	Level	A specific level of data within multi-level data or dimension.
31	Meta Data	Metadata is data about data. Metadata falls into several different categories. The most common categories are descriptive, usage, source, and structural.

Cube Data Terms		
#	Term	Explanation
32	Measure Group	A grouping of measures into a group or folder of related measures.
33	Member	A specific value contained within a dimension or fact table. May also refer to a member of a level within a multi-level dimension or data hierarchy.
34	MOLAP	Multidimensional OLAP. Data is stored in a Multidimensional Cube and in the file system in a proprietary cube format.
35	Parent	In a hierarchy with distinct leaves and roots, the parent of a member is the member one hierarchical unit towards the root from the member.
36	Peer	In a hierarchy, all members at the same level are considered to be peers. Note the multiple definitions for level; depending on the definition of level used, the peer group will be different.
37	Perspectives	A feature of SSAS cubes where a set of different views of a cube can be created that shows and hides different measures and dimensions. This is often used to present simplified versions of a cube or special versions of a cube to specific groups of users.
38	Regular Dimension	A normal dimension created from a dimension table.
39	Reference Dimension	A special type of dimension in a Multidimensional Cube that connects two dimensions together. Used for complex relationships between fact tables and dimensions. Not used or available in a Tabular Cube.
40	Relationship	A connection between two or more objects and/or object values. The records in a fact table having a Country column would have a relationship with the Country dimension table.
41	ROLAP	Relational OLAP. Data is stored in a normal relational SQL database and a cube is simulated in memory. Not often used with SSAS.
42	Role Playing Dimension	A single dimension that can be used multiple times in a Multidimensional Cube with different names and connections. Saves from having to duplicate dimensions. Often used with Date Dimensions. Not available in Tabular Cubes.
43	Sibling	All children of the same parent are siblings.

Cube Data Terms		
#	Term	Explanation
44	Slowly Changing Dimension (SCD)	A slowly changing dimension is a dimension that changes over time which can cause data disconnections and problems. For example, a cube has a City dimension. Cities are normally rather static but in Georgia the cities of John Creek and Milton split off from the city of Alpharetta. This affects the City dimension because your sales that were in Alpharetta are now really in three cities. So, do you just update all your old records in the fact table with the new city names (probably by zip code or address determinations)? That will overwrite all your old data which could cause problems when it comes time to calculate commissions and bonus based upon cities. If you just overwrite the data with the new values, it's called a Type 1 SCD. A Type 2 SCD creates new dimension records with the new values so that the old ones are preserved, and new ones are easily added. No data is lost. This is good! A Type 3 SCD adds new columns instead of records... It has some advantages... As do Type 4 and Type 6 and other hybrid approaches. This is an area of common debate over which to use. And it can get fairly complex.
45	Semantic Layer	An abstraction layer that allows business and other names to be used and changed without affecting the underlaying data object name or ID. For example, in a Fact Table a column is named "Date" (as in a Date Dimension). When used in a cube it is renamed "Received Date" for the date something is received in the semantic layer of the cube which is more descriptive to the users. Note that no one really ever says "semantic layer" unless they are a newbie trying to make sense of it or to impress someone or explain a technical point. Or an instructor explaining it. Everyone just says something to the extent that "this is the name in the cube that we'll use" and takes it for granted. Note that it is a bad practice to keep your database table and column names unchanged in a cube. And a best practice to alter them as necessary for better recognition and understand and also to avoid confusion with users as these are the names that they will see in the cube (and need to understand).
46	Semi-Additive Measure	A semi-additive measure has different rules of aggregation at different levels. For example, the calculation of costs or commissions at the city level may be 5% while at the state level 3% and then 2% for the region. Note that this feature is only available with Multidimensional cubes and not with tabular cubes.

Cube Data Terms		
#	Term	Explanation
47	Shared Dimension	A dimension that is shared between two or more fact tables. A Date or Time Dimension is a very common example. Also called a "Conformed Dimension".
48	Sparse Cube	A cube with many empty data values (cells). The opposite of a Dense Cube.
49	Static Dimension	A dimension that is loaded once or rarely and doesn't change. Static dimensions contain values the hardly ever change. For example, a State Dimension for the US will contain only a static, fixed listing of the 50 US states (and often territories and sometimes military base designators as well for address purposes) as the number and names of US states are currently fixed and have been for quite a few years.
50	Time Dimension	A dimension composed of dates and perhaps also hours or minutes, i.e. a Date Time Dimension. A Date Dimension, holding only dates, is often called a Time Dimension but is more properly called a Date Dimension and often has a table name of DimDate. A Time Dimension technically has a time as well as a date component. This time component can be hours. There is then an entry for each for each date in this dimension for each hour. The granularity of this is called Date Hour Granularity and other terms. A cube with a granularity of hours allows data to be reported on and analyzed at a finer/higher resolution, i.e. to the hour. It also increases the amount of by 23 times (24 hours per day instead of 1 per day, i.e. midnight). See also Date Dimension and Granularity.
51	Tuple	A tuple uniquely identifies a single cell in a Multidimensional cube, query, or definition. Most people never use this word unless they are doing MDX (see) and I gave up using it years ago because it confuses people. Just say something like "unique cell value" or simply "value" instead.
52	UDM – Unified Dimensional Model	The underlaying model architecture of Multidimensional Cubes. Very much hyped by Microsoft years ago with Multidimensional Cubes but not often mentioned or recognized now. Not used with Tabular Cubes.
53	Virtual Dimension	The ability with Multidimensional Cubes to create a dimension in the cube that does not exist as a Dimension Table. Can be used to connect multiple Fact Tables in the cube. Use by Grant for some highly complex analysis (analytical) cubes. Not supported by Tabular Cubes.

Cube Data Terms		
#	Term	Explanation
54	Virtual Cube	The ability with Multidimensional Cubes to create a cube that does not exist as a Fact Table. Can be used to connect multiple Fact Tables and also multiple separate cubes in a single cube. Use by Grant for some highly complex analysis (analytical) cubes. Not supported by Tabular Cubes.
55	Visibility	A property in the cube that allows a cube object such as a measure, a dimension, or an attribute to be visible or not (hidden).
56	XML	eXtensible Markup Language – A markup definition language similar to HTML that is used to store and transmit data.
57	XMLA	XML for Analysis command. A command language used by SSAS based on an extended XML for storing commands, command scripts, and controlling and requesting data from SSAS OLAP cubes. Supports both Multidimensional and Tabular Cubes since both are hosted in SSAS servers (though not at the same time in the same server).

12.4 Database Related Terms

Database Related Terms		
#	Term	Explanation
1	Catalog	Another name for database, specifically used in connection strings and sometimes to refer to the individual databases within a relational database or a cube database.
2	Cube Database	Microsoft Analysis Servers (SSAS) hosts cubes within cube databases. These are individual cube type databases within SSAS. Most people never use this term referring to “cubes” generically instead. However, when you browse SSAS you will see the cube listed under a “Database” heading. Cubes reside in Cube Databases which live in SSAS.
3	Cube Data Model / Schema	The underlaying database model/schema from which a cube is built.
4	Database	Generally, a relational database server that stores data in a relational form and uses SQL to access and manipulate that data. Microsoft SQL Server is a good example and a popular one.
5	Data Mart	A small Data Warehouse.
5	Data Model	The organization of database objects that implements a data bases.

Database Related Terms		
#	Term	Explanation
6	Data Schema	Another common term for Data Model. Note that in many databased the term “Schema” also refers to, and is often used somewhat interchangeably with (but is in reality something quite different), a grouping of database tables and objects. The default schema in SQL Server is “dbo” but additional schemas can be added and used such as “stage” or “staging” to denote stage tables. Not really “folders” for databases but it does provide a rudimentary type of grouping organization to tables and other database objects.
7	Data Warehouse	The primary repository of data from the company’s online databases. A data warehouse is specifically designed to be optimal for reporting instead of online transaction processing purposes. The T&E claims cube data originates from the Teradata data warehouse database.
8	De-Normalization	To take a normalized data model or schema and undo the normalization previously applied to it. Most often used to get data in a form that can be easily used in reports. The opposite of Normalization.
9	Dimensional Model	A Star or Snowflake or Cube model having a Fact Table and one or more Dimensional tables.
10	Dimension Table	A database table that defines a dimension for a Fact Table.
11	Fact Table	A reporting table holding Facts or numeric values to be reported on.
12	Foreign Key	A database relational integrity constraint that enforces that only certain values defined in another table can be entered in a column.
13	Index	A database “map” that helps speed up queries and reports by allowing the database engine to find faster paths to data rather then scanning each and every row for the desired data sequentially.
14	Inmon, Bill	A data warehouse pioneer credited for inventing the term “Data Warehouse”.
15	Join	A definition of a connection between two tables that is used in SQL queries to connect and report on data from two or more tables.
16	Kimball, Ralph	A data warehouse pioneer that influenced Microsoft’s approach to data warehouses.
17	Many to Many	A data relationship where many values in one table connect to many values in another table.
18	Many to One	A data relationship where many values in one table to connect to one and only one value in another table. Used with Dimension Tables.

Database Related Terms		
#	Term	Explanation
19	Normalization	The process of organizing, categorizing, and putting data into a relational database schema/data model/set of tables according to relational principles.
20	Null	A value in a database column row cell that indicates the lack of any specific value, i.e. a non-entry. This is essentially a “non-existent” value and is distinct from a zero or a blank value. Specially used in database queries.
21	One to Many	A data relationship where one value in one table corresponds to many values in another table.
22	One to One	A data relationship where one value in one table corresponds to one and only one value in another table.
23	Primary Key	A column or combined columns that represents a unique value and identifier for each row in a database table.
24	Relational Integrity	The aspect of a database which has Primary Keys, Foreign Keys, and other relational constraints that enforce the values and forms of data within the database. This prevents mixing of different types and forms of values. High Relational Integrity is good. Low Relational Integrity is bad and will cause all sort of data problems.
25	Relational Model	A database and database tables and objects that follows most of the rules of relational integrity and the definition of a relational database put forth by Edgar F. Codd in 1969 at the dawn of the database era.
26	Report Table	A database table that is used specifically for reporting and not transactions. Most often it is a De-Normalized Table.
27	SQL	Structured Query Language. Pronounce “SEQUEL”. The programming and query language of relational databases. Developed in the early 1970s it was first standardized in 1986 and occasionally updated through 2016. However, virtually all relational database vendors have their own versions of SQL which support most of the various SQL standards and also extend SQL in many different ways. The SQL Server version is called “Transact SQL” abbreviated “T/SQL” but everyone just says “SQL” and assumes that you know that it’s noticeability different from Oracle’s PL/SQL and others.

Database Related Terms		
#	Term	Explanation
28	Snowflake Schema	A snowflake schema is a Star Schema that has one or more additional layers/levels of dimension tables that connect to other dimension tables. For example, a City dimension could connect to a State dimension and then to a Region dimension. Note that Tabular Cubes cannot use Snowflake Schemas. They can only use Star Schemas. They can mimic with some developmental difficulty the levels and hierarchies of the Snowflake schema but only in a compressed state within a Star Schema. Multidimensional cubes can use Start and Snowflake Schemas and excel with complex Snowflake Schemas.
29	Stage / Staging Table	An intermediate table or set of tables that is used to import raw data so that it can be further processed by ETL.
30	Star Schema	A star schema has a central fact table surrounding by one to many dimension tables in one layer/level. In other words, dimensions connect only to the fact table they do not connect to other dimensions. Tabular cubes can only use Star Schemas. Multidimensional cubes can use Start and Snowflake Schemas. A star schema is an arrangement of tables in a relational database where a central “fact” table is connected to a set of “dimension” tables, one per dimension. The name “star” comes from the usual diagrammatic depiction of this schema with the fact table in the center and each dimension table shown surrounding it, like the points on a starburst.
31	Summary Table	A reporting type table that provides a readymade summary of data so that simpler SQL queries can access the pre-summarized data. This results in simple and faster performing report queries using these tables.
32	Surrogate Key	An artificial non-natural key that represents a natural key. Used for both convenience and performance reasons. For example, a date dimension may use a surrogate key of integer form (20170101) which represents the natural key of 1/1/2017. This is expected to result in higher performance in larger cubes.

12.5 Cube Processing and ETL Terms

Cube Processing and ETL Terms		
#	**Term**	**Explanation**
1	Build	With cubes, specifically refers to "building" a cube using Microsoft Visual Studio which is the tool used to develop cubes. Note that in this context "building" a cube is essentially like "compiling" a program but no one ever says "compiling a cube". This can be confusing to non-developers.
2	Cube Processing	To process a cube is to initiate the process of bringing the data into an already built and deployed cube and "processing" it which means the building in memory of the all the cube structures from the cube definition that define the cube. Cube processing is only done in SSAS. And there are several types. The most important things to know about cube processing is that for new data to be available in the cube the cube must be processed. Most cubes are processed once a night so that the data is available to users first thing in the morning. While you can process cubes many times a day it can take time to for the processing, especially with large cubes (ones with large amounts of data in them).
3	Data Cleaning / Cleansing	Data cleansing refers to the process of discovering and repairing anomalous and/or bad data within data tables.
4	Data Explosion	The situation when the amount of data in a process, cube, or reports expands greatly, often geometrically. Data Explosion is typically encountered in cubing and reporting where the availability of the new BI data access to users results in the demand for data to grow significantly, especially in situations where user access to data was previously restricted. See "Cube Explosion".
5	Data Quality	Data in a relational database is supposed to be highly organized by the type of data. And often is if good design and implementation practices are used along with Relational Integrity. When it is, the quality of the data is said to be high. When it's not then the data is said to be low or poor. This is very important from a cubing standpoint because bad/low Data Quality needs to be fixed before a cube can be properly built and used. Very often there is Data Quality issues with data to be cubed which results in extra work and delays in cube development.

Cube Processing and ETL Terms		
#	**Term**	**Explanation**
6	Data Scrubbing	Cleaning of data so that it can be put into a proper form for cubing. Also called Data Cleaning or Data Cleansing. Scrubbing, as in physically cleaning something, often means that extra effort must be expended because of Dirty Data. Note: This can be a very big job. It may take some time and be a significant part of cubing activity especially if the data is in files or a database with poor relational integrity. (Relational Integrity works to keep data clean in a database, but the database must be designed to have it. It is not an automatic part of using a database (as some people seem to think!). Examples: 1. Lack of a primary key on a database table causes duplicate data to be present that must be cleaned up before the data can be cubed. 2. Address data has a variety of non-standardized street and address variations (such as “St.”, “street”, “ST”, etc) that need to be standardized and cleaned up before it can be used in a cube dimension. Note: A Data Cleaning solution may be very intricate and require customized coding to implement. SQL Server 2005 Integration Services is often used for this and has many built-in features for data cleaning and scrubbing.
7	Data Transformation	Changing data from one type or form to another. For example, a number stored as a text string (varchar) must be changed to a proper date or datetime data type to be properly used in a cube.
8	Deployment	Publishing a cube to a SSAS server on another physical server. It can also means applying changes and new additions to the underlaying cube database tables to another server, often a production server. This can and will encounter problems if deployment (DevOps) people are not briefed or educated properly and/or if the proper procedures are not created and used. Else, properly handled, deployments can be quite smooth and trouble free.
9	Dirty Data	Data which has problems in consistency, validity, and standardization which requires Data Cleaning, Cleansing, or Scrubbing.
10	ETL	Extract, Transform, and Load. The common term of taking data from one or more data sources and putting into cube schema tables for ingestion into the cube.
11	Pre-Cube	Data that is not yet in the cube.

Cube Processing and ETL Terms		
#	Term	Explanation
12	Snowflaking	The act of creating a snowflake schema upon which to build cubes. May also mean the act of expanding an existing star schema into a snowflake schema by adding additional fact and dimension tables.
13	Staging	To prepare data in a set of tables where the data is raw and yet to be moved and transformed to a form suitable for cubing. An intermediate state of data import and transformation in the cubing and ETL process.

12.6 Microsoft Cube Products Terms

Microsoft Cube Products Terms		
#	Term	Explanation
1	BIDS	The old term for a specialized version of Visual Studio used to make cubes.
2	DTS	The old term for SSIS.
3	SSDT	SQL Server Data Tools. The Visual Studio add in necessary to develop OLAP cubes.
4	PowerBI	A dashboarding service provided by Microsoft.
5	Report Builder	A SSRS tool that allows users to build custom reports from a set of pre-developed report definitions.
6	SQL Server	Microsoft SQL Server, the advanced feature rich database produced by Microsoft. The most recent version of the product is SQL Server 2016.
7	SSAS - Analysis Services	Microsoft SQL Server Analysis Server. Hosts both Multidimensional (1st generation) Cubes and Tabular (2nd generation) Cubes (but not both at the same time).
8	SSIS	SQL Server Integration Services. A popular ETL tool from Microsoft that allows one to develop ETL programs.
9	SSMS	SQL Server Management Studio. The Microsoft tool that connects to SQL Server databases and allows one to query data, perform maintenance, and other useful database tasks.
10	SSRS	SQL Server Reporting Services. The Microsoft tool to build and host reports.

12.7 Reporting Terms

Reporting Terms		
#	**Term**	**Explanation**
1	Ad Hoc	A report that is created on demand instead of formally requested and developed through a report development process. Requires either SQL query skills or a specialized tool that support ad hoc querying.
2	Canned Report	A previously developed report that one runs to see specific data.
3	Cross Tab Report	A report that summarizes information.
4	Cube Browser	A tool that connects to an OLAP cube and allows one to browse the cube data and generate ad hoc type reports.
5	Drill Down	The ability in a report to navigate “downwards” in the data to greater detail. Example: To be able to view all divisions then “drill down” to a specific division.
6	Drill Through	The ability in a report to navigate to additional data. Example: To be able to view a division then “drill through” to specific address and location data on that division. This data would not be part of the cube but would be accessed through a specific query to another database.
7	Drill Up	The ability in a report to navigate “upwards” in the data to greater scope. Example: To be able to view all divisions after viewing a specific division.
8	Dynamic Report	A report that requires one or more parameters which customizes the report output data in specific ways. For example, given a parameter for the store and the date a report could show the sales data for that specific store for that date.
9	Field List	A listing of the business categories available for analysis in the T&E client.
10	Filter	A parameter of the database query that specifies the scope (or limiting) of the data in the report.
11	Filter Area	An area of the T&E client where data filters are displayed and can be added and removed to filter up (expand) or filter down (narrow) the data displayed in the report.
12	Parameters	Additional information needed and/or requested by for a query. For example, a report may request that the user select a specific division or cost code before submitting a query to the database.

Reporting Terms		
#	Term	Explanation
13	Pivot Table	A Microsoft Excel reporting object that allows data to be summarized and displayed. Also, can be used to connect to OLAP cubes and report on cube data.
14	Query	A request to a database for specific data.
15	Query Results	The data returned from a query to a database.
16	Report	A formatted and filtered data display or printout of data from a database or a cube.
17	Reporting System	A formal system to support and generate reports often within structured architecture. Most reporting at most companies, however, is not really a planned system just a bunch of tools and approaches that were tried and accumulated over time.
18	Roll Up	A query or report that summarizes the data, i.e. “rolls it upwards”.
19	Slice and Dice	The ability to easily do custom queries against a cube (or even a database) via a GUI that allows the users to select data values then select dimensions to get reduce the data display down to the sections of data (slices) that are useful or of interest. A slice of a cube corresponds to a flat plane or planes use to isolate data of interest.
20	Slicer	The part of the MDX statement that is used to narrow down the result set based on specific values and sets of values. Roughly corresponds to the WHERE clause in a SQL statement. Example: [insert sample MDX statement here and point out the slicer dimension]
21	Static Report	A report that has no parameters and thus doesn’t change the format or view of its data. The report can’t be changed, i.e. it’s static. The opposite of Dynamic Report.

12.8 Cube Slang

Cube Slang		
#	Term	Explanation
1	Borg	A reference to a race of half-living, half-robot de-humanized creatures in Star Trek: The Next Generation stories that fly around space in cube shaped spacecraft.

Cube Slang		
#	Term	Explanation
2	Cube Explosion	The situation when the amount of data in a cube or reports expands greatly, often geometrically. Cube Explosion is typically encountered in cubing where multiple cubes are necessary for data access, reporting, complexity, or performance reasons. Note: The use of Perspectives in AS2005 can provide simplified views of complex cubes and thus can help with Cube Explosion caused by cube complexity problems/issues. See “Data Explosion”.
3	Cube It!	An enthusiastic term for cubing data.
4	Cube Land	A generic term for the group of developers (or their physical location) that build and develop database cubes. Also may be used as a generic reference to the process of cube building as in “somewhere in Cube Land”.
5	Cubize	The process of putting data into a cube. Similar to the term “Cubing” but with a lot less use of good English.
6	Cubical	A variation of the “tubular” slang term but with a cube slant.
7	Cubing	The process of putting data into a cube.
8	Get Cubular!	An enthusiastic slang term for cubing.

12.9 Other Terms

Other Terms		
#	Term	Explanation
1	Analytics	The analysis of data often through statistics or other specialized means or tools.
2	Data Lake	A new term for the storage of a large amount of raw data.
3	Data Science	The use of statistics to analyze data often with special tools and languages such as R or Python.

Other Terms		
#	**Term**	**Explanation**
4	NoSQL	A database that is non-relational and often does not use the SQL language the relational databases do. NoSQL database are often built around JSON (JavaScript Object Notation) a very popular replacement for XML that is integrated nicely into and with JavaScript. NoSQL databases have gained a large following in developers who use then with JavaScript and seem to want to forget what they view as the complexities and limitations of conventional relational database systems. Two common and popular NoSQL databases are MongoDB and Microsoft's Cosmos DB (formerly DocumentDB).
5	Landing Page	A page that users see (land on) when they access a website. When used with cubes, a landing page and Portal (see) can provide users with automated information about the status of production cubes and data, maintenance activities, and other useful information for users.
6	Portal / Cube Portal	A portal, as in Cube Portal, is a web site used to assist users with their use of cubes. It may contain web based reporting and Cube Browsers as well as automated info about the status of production cubes and data, maintenance activities, and other useful information for users. Grant recommends a Cube Portal as a Best Practice.

Thanks!

Thanks for reading and using this book!