

Overview of the Microsoft Business Intelligence Toolset

In This Part

Chapter 1: Which Analysis and Reporting Tools Do You Need?

Chapter 2: Designing an Effective Business Intelligence Architecture

Chapter 3: Selecting the Data Architecture that Fits Your Organization

Which Analysis and Reporting Tools Do You Need?

When embarking on a business intelligence (BI) project, you should consider several things. Should a centralized data warehouse be built or can the existing operational database act as the source for business intelligence? Once that hurdle has been leaped, the next question is: Should time be spent building a semantic model (cube) or again back to the original question: Can the existing operational database act as the source for business intelligence? Finally, once you've answered those questions, you need to decide how to deliver the data to end users. In other words, which reporting tool will be used? The focus throughout this book is on selecting, designing, and delivering a business intelligence solution based on the Microsoft business intelligence tools stack.

Regardless of the approach, you must make a decision concerning which tools to use to ultimately deliver the business intelligence solution. If a data warehouse is built, which Relational Database Management System (RDBMS) will store the data? Now that you have a data warehouse, is a cube or semantic model needed? If so, which type of model should you use: Power Pivot, tabular, or multidimensional? You then need to determine if the solution offers self-service reporting and/or operational reporting capabilities.

Selecting a SQL Server Database Engine

After all the politics have been hashed out, the first step in your business intelligence solution is identifying the data sources. In most scenarios, the solution will include a plethora of data sources, ranging from flat files to relational databases. After that, you must build an Extraction, Transformation, and Loading (ETL) system, which centralizes that data into a data warehouse. The data warehouse is typically housed on an RDBMS.

Building a Data Warehouse

A valid argument could be made against building a data warehouse. However, you should consider whether you prefer to report against a centralized, single-source pristine dataset or to report against multiple, disparate questionable data sources. In other words, are reports more effective leveraging data that is definitely accurate or possibly inaccurate? Another thing to consider is the responsiveness of the business intelligence solution without centralizing the data into a single repository. Often, organizations attempt to analyze data directly against source data and quickly realize that, even though simple, this approach is not efficient nor effective. Figure 1-1 shows a sample topology of this solution.

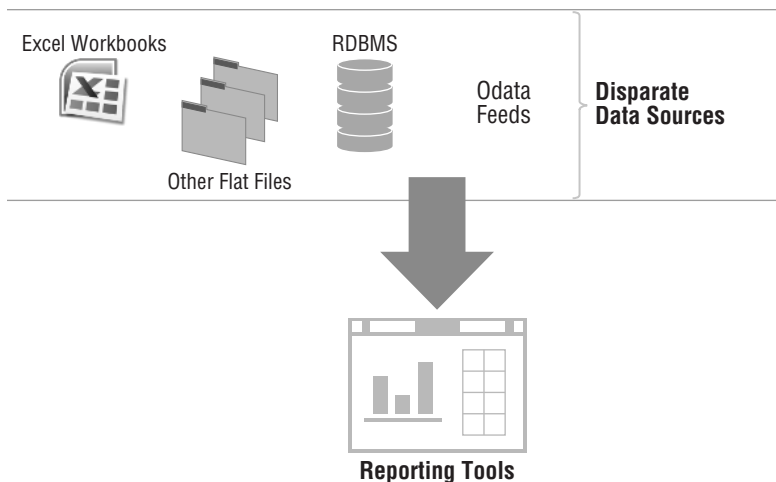


Figure 1-1: Reporting against disparate data sources

As a result, most organizations often decide to build a data warehouse. Figure 1-2 depicts a sample of a business intelligence solution that includes a data warehouse. Notice in this figure that instead of attempting to build reports against multiple data sources, a single source is used.

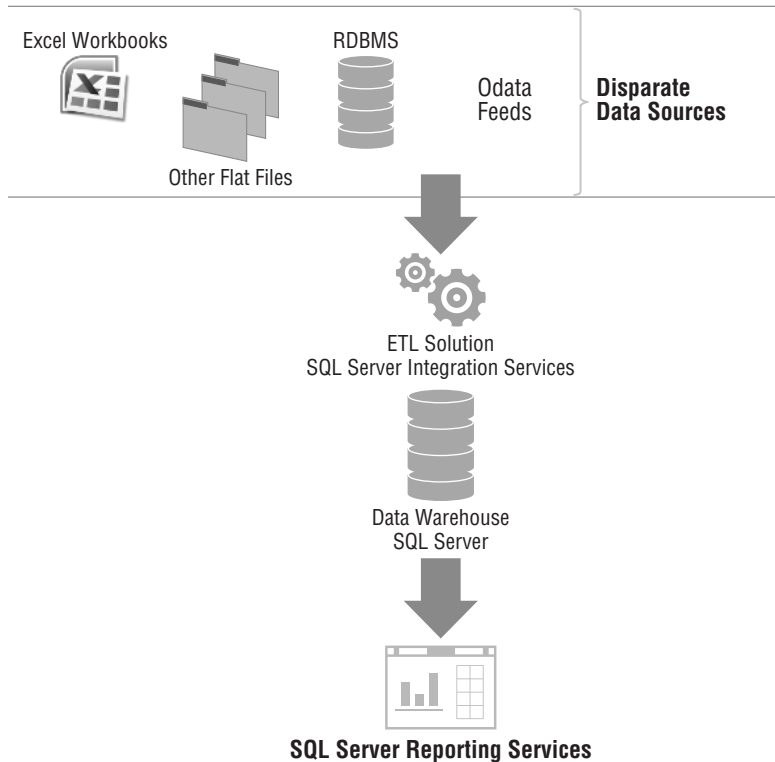


Figure 1-2: Business intelligence solution that includes ETL solution and data warehouse

Selecting an RDBMS

Once you've built a data warehouse, the next step is to select an RDBMS. The market for RDBMS systems has a wide range of choices. Selecting the correct system depends on several factors: number of users, disk space, data size, rate of growth, and frequency of data load to mention a few. Microsoft's RDBMS—SQL Server—includes several features that make it one of the more appealing systems available on the market. As of the writing of this book, SQL Server includes an in-memory Columnstore index which is designed specifically for data warehousing workloads. When included in the data warehouse design, you can achieve significant query performance and data compression. Another feature, Change Data Capture (CDC), assists in minimizing the amount of time required to load the data warehouse by providing mechanisms that detect inserts, updates, and deletes. These two features alone make SQL Server a viable Database Management System for hosting your data warehouse.

Selecting SQL Server Analysis Services

Now that a database engine is selected to host the data warehouse, the decision to build an analytical model or, in the case of a Microsoft Solution, semantic model must be made. With the latest release of SQL Server, semantic models have three choices from which you can select:

- Power Pivot
- Tabular
- Multidimensional

So not only must you decide how to build a semantic model, but also which model to use.

If the business intelligence solution requires very fast response times, ad-hoc capabilities, or predictive analytics, leveraging SQL Server Analysis Services (SSAS) is a great option. Whereas the aforementioned list is not inclusive of all factors that may drive the need for a semantic model, they definitely make a strong case in favor of it. SSAS offers a wide range of capabilities that assist in streamlining and reducing report requests, centralizing analytical formulas and key performance indicators, and—probably one of the more important robust capabilities—intuitively handling security at different levels. Figure 1-3 illustrates a business intelligence solution that includes a semantic model. Notice how the reporting tools are expanded when you compare them with Figure 1-2.

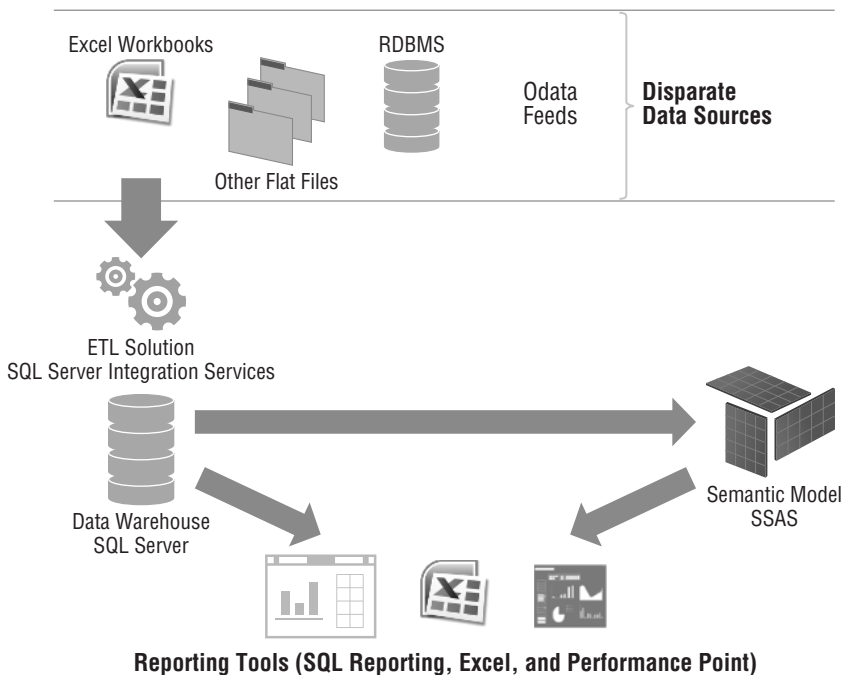


Figure 1-3: Business intelligence solution that includes SSAS semantic model

Although it is possible to report directly against a data warehouse using Excel and Performance Point (discussed later in the chapter), SSAS provides a more innate design experience with these tools. In addition, using SSAS provides end users with a larger surface of self-service capabilities that are unavailable when only a data warehouse is available. Therefore, they are excluded from Figure 1-2, but included in Figure 1-3.

For example, if you are the CEO of a company, you may require access to every aspect of data in the model. However, if you are a regional or departmental manager, you may only require access to data that is pertinent to your region or department. SSAS includes built-in capabilities that let you control access to data at the row level. In many cases, this is one of the most important and often overlooked requirements of a business intelligence solution. During most projects, you don't realize this until very late in the development process. However, when using SSAS, the implementation process is neither very difficult nor disruptive.

Working with SQL Server Reporting Services

Up to this point, all the data discussions have involved movement, transformation, and management of data. This section shifts to more data visualization and interactivity. Once the processes to implement the data warehouse and/or the semantic model are in place, your next decision is how the end users will access the data. When leveraging the Microsoft business intelligence stack, organizations have several reporting options. From an operational perspective, probably the most utilized is SQL Server Reporting Services (SSRS).

SSRS operates in two modes, which have a few slight differences, but are mostly similar in regard to features and tasks:

- **Native mode:** Access and management of reports are available via a web-based platform, also known as Report Manager.
- **SharePoint Integrated mode:** This is a site collection within SharePoint that has the same purpose as the Report Manager.

SSRS also provides two very different types of reporting experiences. Deciding which to use often poses the biggest challenge for most projects:

- **The first, Operational Reports, are typically used when delivering highly-formatted, table-based and pixel perfect reports.** They are designed to answer a specific question and are usually static in nature. In this case, you would use SSRS.

- **The second type is of a more ad-hoc nature.** End users typically access the underlying source directly, which would be a semantic model in this case, and build reports as needed. The reports are more visual containing charts, maps, gauges and scorecards. For these types of reports end-users would leverage Power View. Each one is discussed in the following sections.

In addition to developing these types of reports, SSRS provides additional capabilities that makes it a complete solution. Features include:

- Report export
- Subscription report delivery
- Data alerts (SharePoint Integrated mode only)
- Data caching
- Report printing
- Report snapshots
- Shared datasets
- Report parts
- Geospatial mapping

While this is not an exhaustive list of all the features, it should provide an overview of what is possible when developing and managing reports using SSRS.

Understanding Operational Reports

Operational reports, available since the inception of SSRS, can help you develop, deploy, and manage standard operational reports. What are operational reports? These are typically row- and column-based reports containing data that answers or meets a specific need. For example, the report shown in Figure 1-4 shows a sample Operating Summary developed using SSRS.

Contoso Education System > SQL Reports Demo > Colleges and Universities > Reports

Actions | [Refresh] | [Previous] | [1] of 1 | [Next] | [Find Next] | [100%] | [Print]

JUL-13 Operating Statement Summary

EDUBOS\PLeBlanc

**blue text denotes drill down*

	Actual
Beginning Equity	43,015,495,386.66
Less prior years capital	(890,495,450.61)
Subtotal Beginning Equity	42,124,999,936.05
Revenue	
Student tuition and fees	179,631,755.39
Tuition Discounts	(870,088.32)
Net tuition and fee revenue	178,961,669.07
Federal, state, and private grants	262,992.38
Contributions	3,242,294.60
Sales of Educational Departments	2,113,184.45
Investment Income	16,286.80
Endowment income used in operations	0.00
Other income	24,874,519.83
Total revenue	209,470,947.13
Expenditures	
Compensation:	
Salaries	11,279,098.47
Fringe benefits	4,081,301.53
Total compensation	15,360,400.00

Figure 1-4: Operating Statement Summary using SSRS

This particular report was designed for a specific audience to solve a specific problem, which in this case was a need to dynamically view operating summaries by month for those individuals in the Accounting department.

Using SSRS, developers can also build very visual reports that resemble high-level dashboards often used by executives; Figure 1-5 illustrates this.

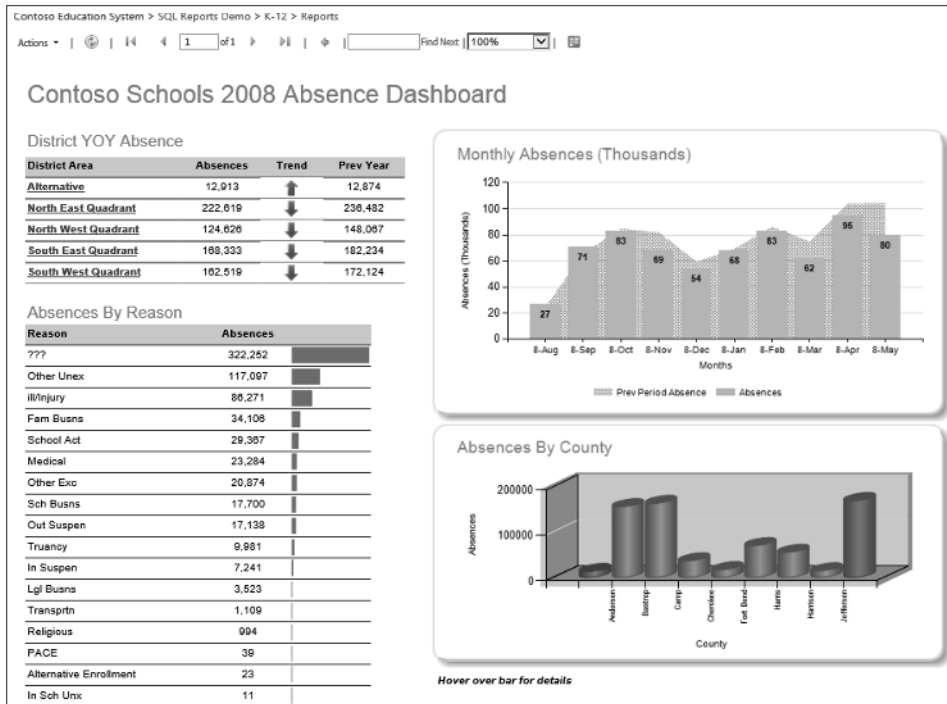


Figure 1-5: High-level dashboard created using SSRS

SSRS includes a complete toolbox of items that allow report developers to build complete reporting solutions including high-level dashboards that provide end users with drill-through ability to more detailed data.

Understanding Ad Hoc Reporting

Suppose end users want some control over the look and feel of reports. More specifically, what if they want ad-hoc access to data, which allows them to create and deploy reports as needed, instead of relying on a group of report developers creating canned reports. The latest release of SSRS integrated into SharePoint, discussed later in the book, exposes a new feature named Power View.

Power View is an ad-hoc, interactive, and presentation-ready self-service reporting tool designed specifically for end users. Instead of waiting on reports from the report development team, end users can quickly access data that is stored in either type of semantic mode and build highly visual and interactive reports. Figure 1-6 displays a sample of a Power View report.

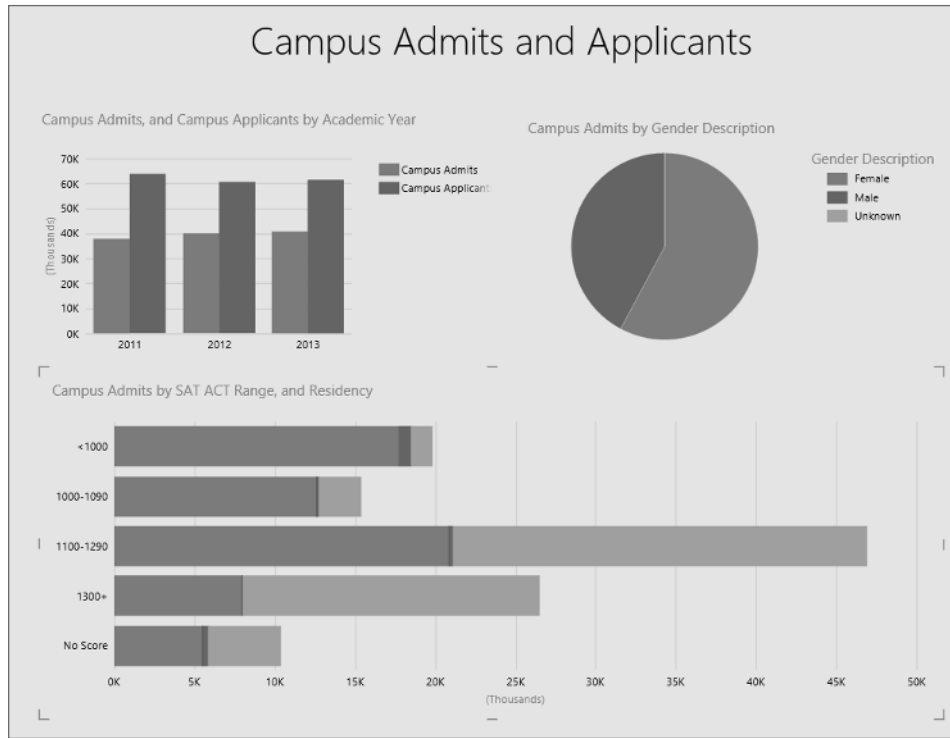


Figure 1-6: Sample Power View report

With Power View, end users become the report authors. A view of the data is made available in a field list, which provides easy access to each data element exposed via a single semantic model.

Working with SharePoint

Traditionally, when Microsoft business intelligence is included in a business intelligence project, most people automatically think of SharePoint. That is because SharePoint acts as a central portal or repository that you can easily access using any modern web browser. Instead of using multiple disparate technologies for the business intelligence solution, SharePoint can render both Microsoft and non-Microsoft reporting visualizations.

It is a common practice that individuals, teams, departments, or organizations will evaluate and select a reporting tool based on a variety of factors. These tools then tend to proliferate themselves within the department and become part of daily operations. Once the discussion of a business intelligence solution begins, it is often difficult to persuade the groups away from their established tool of choice. However, using SharePoint means the IT group responsible for the business intelligence project can couple existing report artifacts with new

technologies to produce a fully functional and comprehensive solution via a single interface. Figure 1-7 provides a pseudo view of what a SharePoint-developed solution may look like.



Figure 1-7: Pseudo SharePoint page displaying multiple technologies

In SharePoint, you can develop custom pages that consume other technologies. Figure 1-7 integrates four different technologies into a single view. The resulting web page centralizes four distinct views of data spanning features that are made available by different software vendors. In addition to customized pages, business intelligence developers can leverage Performance Point, a dashboarding feature of SharePoint. The details of Performance Point are discussed in the next section.

While SharePoint does offer this extended capability of integrating disparate technologies, the Microsoft business intelligence stack does provide a sufficient number of tools and features for deploying a holistic business intelligence solution. Therefore, if an organization is evaluating vendor solutions, leveraging SharePoint typically addresses all the business intelligence needs for a given project.

Working with Performance Point

The previous section focused on SharePoint as a whole. However, when SharePoint is deployed, you have an option to configure Performance Point services. Using Performance Point, developers can create dashboards that aggregate data from a

collection of sources such as Analysis Services, Excel Workbooks, and SharePoint lists. Similar to Power View, Performance Point gives your users a very interactive interface for analyzing data. Where Performance Point really shines over Power View is that it automatically exposes the metadata from the underlying data model as part of the end-user experience. In other words, once a dashboard deploys, end users can simply right-click a given visualization and change the look by drilling down to a different level of the data. Figure 1-8 shows a sample Performance Point dashboard.

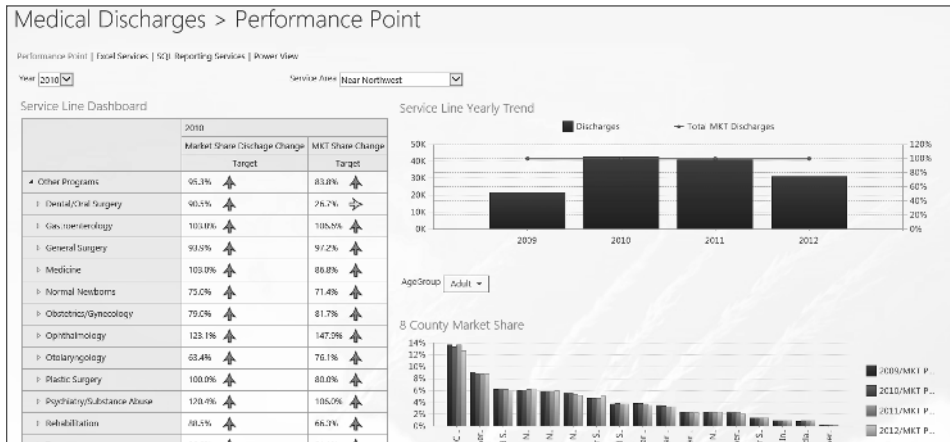


Figure 1-8: Performance Point dashboard

This particular dashboard is a high-level view of medical discharges for a given year and service area. By right-clicking a bar in the bar graph (shown in Figure 1-9) or changing a filter on the dashboard, end users can dynamically analyze the data based on the underlying data model.

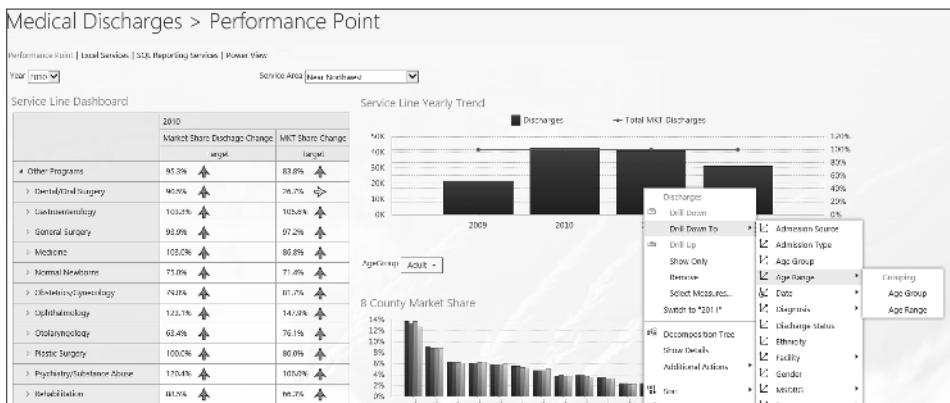


Figure 1-9: Performance Point dashboard with drill-down menu displayed

Therefore, instead of IT developing several reports of varying levels of the same data, you can create a single dashboard that provides end users with different views from one entry point.

Using Excel for Business Intelligence

When most people think of Excel, they think of spreadsheets and pivot tables. However, the latest release of Excel has definitely matured into a full-grown business intelligence authoring tool. Excel 2013 now comes with two new plug-ins in the product (Power Pivot and Power View) and has two additional plug-ins available for download (Power Query and Power Map), thus transforming Excel into a full-fledged business intelligence solution. By leveraging all four plug-ins, users of Excel can discover, model, and visualize data from a single tool. By including Excel as part or all of the business intelligence solution, you gain the primary advantage of providing a familiar tool to the entire population of end users. This results in a lower likelihood of resistance to adoption. The following sections provide a brief overview of each add-in.

What Is Power Query?

Microsoft Power Query is a self-service data discovery and data access tool. It enables end users to easily combine, transform, and share data. When Power Query is installed, not only can end users access structured data from within Excel, but they can also perform public searches. This search is similar to a Bing search; however, instead of returning a list of web page results, Power Query returns a list of datasets that match the entered query.

What Is Power Pivot?

Once all the data has been identified, end users can use Power Pivot to build in-memory data models—meaning they can perform data analysis directly inside Excel. Power Pivot has the ability to consume and process large amounts of data, beyond the normal Excel limits, while including those features of Excel familiar to most Excel users. In addition, Power Pivot introduces a new expression language, Data Analytic Expression (DAX), which provides new data analytic capabilities.

What Is Power View?

Now that you've transformed and modeled all the data, the next step is visualization. To accomplish this task, two new add-ins are now available: Power View

and Power Map. Power View, previously available only via SharePoint, has been added to Excel. The Power View experience is very similar in Excel when you compare it to the discussion in the section “Working with SQL Server Reporting Services.” The primary difference is that authoring is done directly inside of Excel rather than SharePoint. Although this may not seem like a huge difference, if you are an Excel user, it may determine whether you’ll use Power View or not. Besides that, there are a few slight variations, but nothing too significant.

Power Map

Power Map is a 3-D visualization add-in for Excel. This add-in consumes geographical and temporal data and maps it on a 3-D representation of the earth, shown in Figure 1-10.

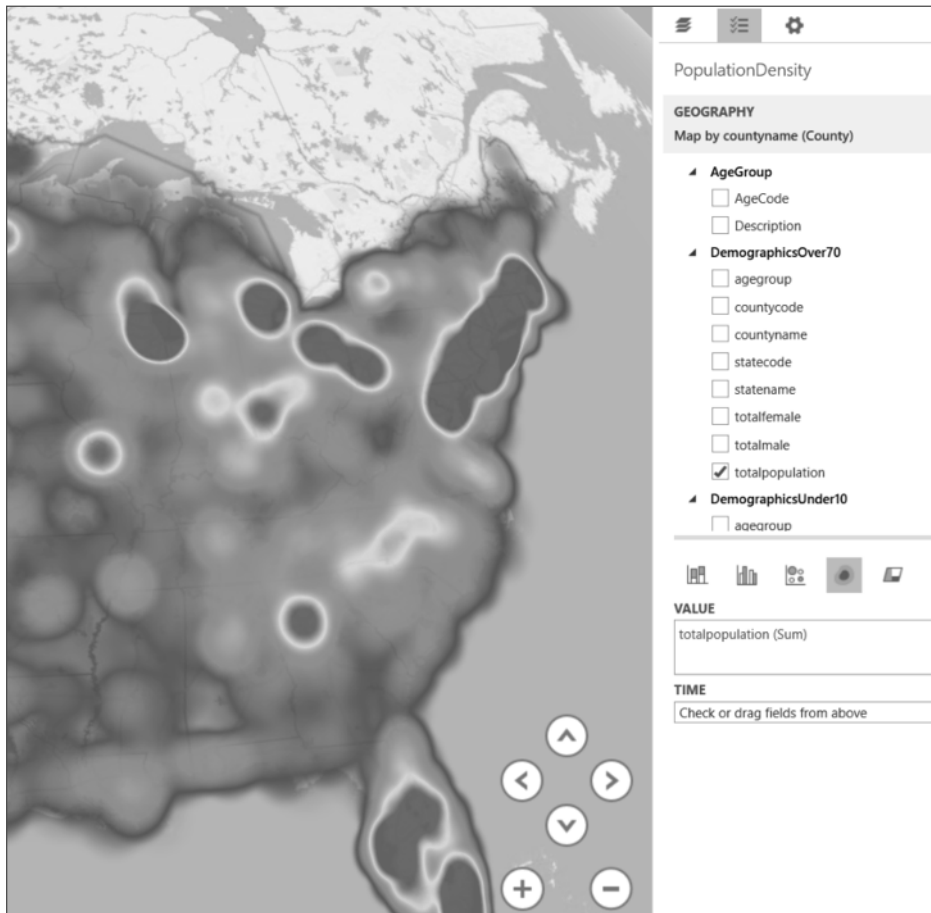


Figure 1-10: Sample Power View map with field list

This means that end users can interact with and derive new insights from their data. The end-user result can be a video that tells a complete story of the data in different scenes that represent various views.

Which Development Tools Do You Need?

So far the discussion has focused on those Microsoft tools that host data and provide end-user consumption. The focus now shifts to the tools you actually need to develop the solution. Primarily four tools are used in the development process:

- SQL Server Data Tools (SSDT)
- SQL Server Management Studio (SSMS)
- Performance Point Dashboard Designer
- Report Builder

What tools end users utilize is determined by what they're developing and who does the development. In some cases, all tools are used, and in others cases, only an abbreviated set. Some instances may require the use of tools beyond the Microsoft stack. However, for the sake of brevity and because this book is focused on the Microsoft business intelligence, we'll discuss only tools specific to Microsoft.

Using SQL Server Data Tools

SQL Server Data Tools (SSDT) is the most comprehensive set of tools in the list. SSDT offers a full-range experience from which developers can address almost every facet of a business intelligence solution from within a single environment. Figure 1-11 displays a list of the templates available to developers from within SSDT.

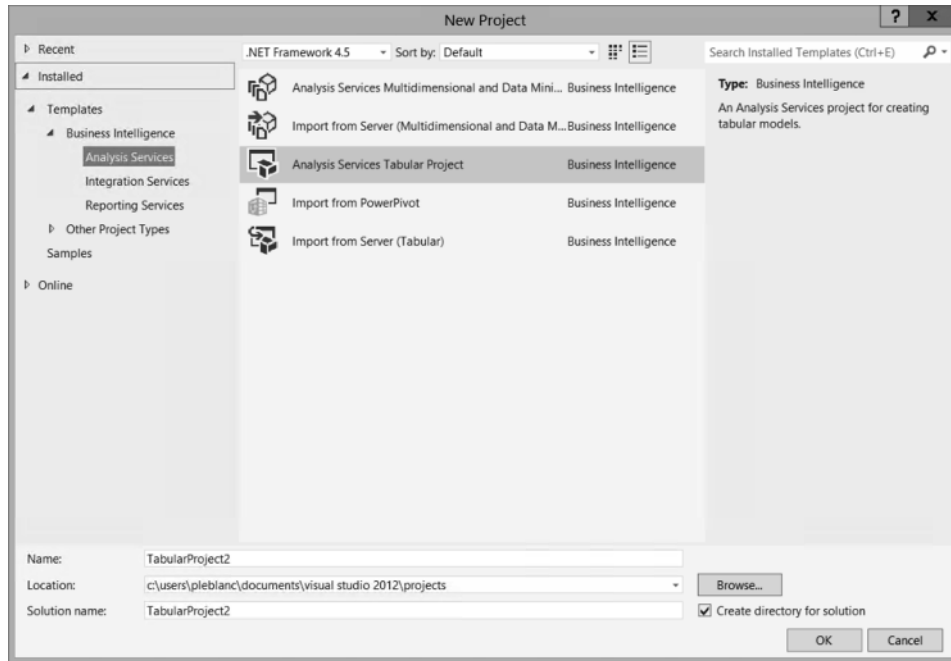


Figure 1-11: SQL Server Data Tools template list

In this tool, you can develop SQL Server Integration Services, SQL Server Reporting Services, and semantic model solutions individually or as a team. You can also develop database solutions using SSDT, which is a perfect environment for developing the data warehouse schema because SSDT provides capabilities such as refactoring code and schema compare. In addition, by leveraging SSDT as the development tool, you can version-control the entire solution using Microsoft Team Foundation Server or other third-party version-control tools like Subversion.

Using SQL Management Studio

SQL Server Management Studio (SSMS) is often considered a Database Administrators (DBA) – centric tool. However, you can also use it to develop a data warehouse schema. You could argue that it does not provide a complete development environment because it lacks certain features like version control and refactoring. Although this is true, it does provide an interface for testing and debugging Transact-SQL (TSQL) code and a diagramming feature that allows for creating and managing tables and relationships. Figure 1-12 provides a view of a typical star-schema data warehouse from the SSMS perspective.

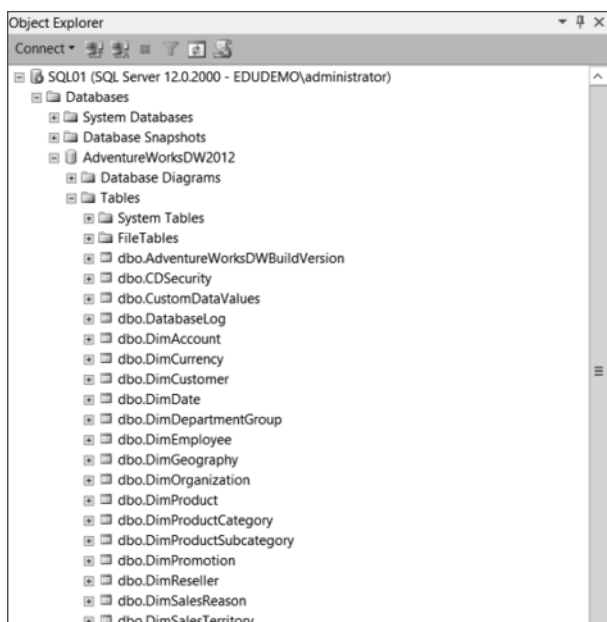


Figure 1-12: SQL Server Management Studio

Using Dashboard Designer

The starting point for authoring Performance Point dashboards, discussed earlier in the section “Working with Performance Point,” is Dashboard Designer. Dashboard Designer is a click-once application available in SharePoint that is installed on each individual machine that will author Performance Point content. It is not available by default. Once installed, you can launch Dashboard Designer to build and deploy dashboards to a SharePoint site, which is shown in Figure 1-13.

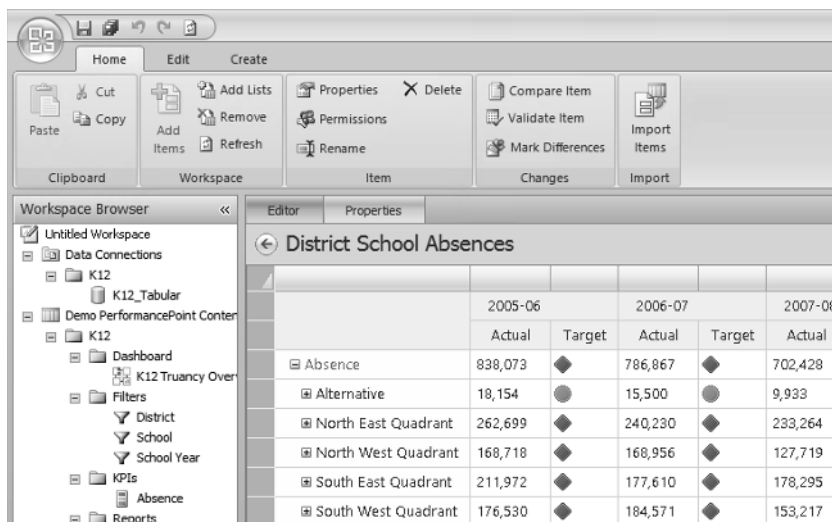


Figure 1-13: Dashboard Designer displaying scorecard

Using Dashboard Designer, you can develop each individual item that appears on the dashboard and then create a dashboard that displays the finished product. Once complete, you can then deploy the dashboard to view it.

Using Report Builder

Similar to Dashboard Designer, Report Builder is also a client tool that you must install on each individual developer's machine. You use Report Builder to develop SQL Server Reporting Services reports. When you compare it to designing reports using SSDT, the features and functionality are almost exactly the same. The difference is mainly in regard to team development and source control. A report designed in SSDT is included within a solution and a project. When designing reports with Report Builder, you can develop only a single report at a time. The concept of team collaboration is not introduced. Report Builder is a light-weight tool for people that need to quickly create and/or modify reports as needed. The application itself is installed on demand, similar to PerformancePoint.

Figure 1-14 displays a sample report being developed with Report Builder.

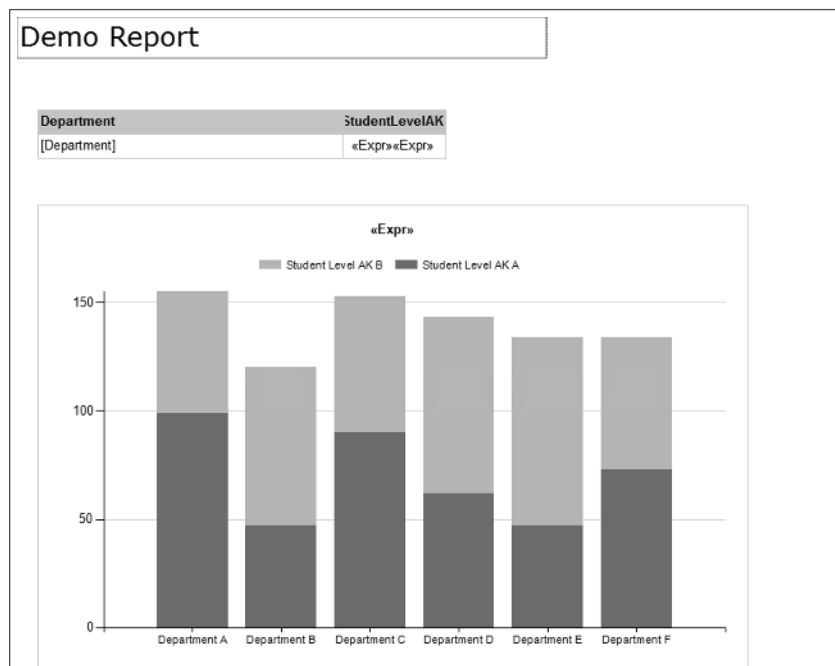


Figure 1-14: Sample Report Builder report

Summary

This chapter discussed the different Microsoft business intelligence tools that you can use to author, deploy, and maintain a business intelligence solution. In addition, overviews were provided for each tool, and, in some cases, comparisons among the different features and capabilities were provided.

The next chapter provides an overview and some discussions around designing an effective and efficient business intelligence architecture.