# SSAS Interview Questions on Measures, Actions, and Storage

##### Problem

Measures and KPIs are very important aspects of an OLAP/SSAS solution from an end user standpoint, and in general, an important aspect of any Business Intelligence application. Storage is another important aspect of SSAS from an engineering standpoint. Hence it is essential to have a fair understanding of these aspects.  Check out these SQL Server Analysis Services (SSAS) interview questions.

##### Solution

In the [first](http://www.mssqltips.com/sqlservertip/2637/ssas-interview-questions-part-i-questions-on-basic-concepts-data-sources-and-data-source-views/), [second](http://www.mssqltips.com/sqlservertip/2662/sql-server-analysis-services-interview-questions-part-ii--dimensions/), and [third](http://www.mssqltips.com/sqlservertip/2683/sql-server-analysis-services-interview-questions-on-dimensions-hierarchies-and-properties/) tips of this series, I have covered questions on basic concepts, data sources, data source views, dimensions, dimension types, and components associated with dimensions like attributes, hierarchies, and some of the dimension properties. In this fourth tip, I will be covering some of the questions on measures, KPIs, actions, partitions, storage modes, etc. within [SQL Server Analysis Services (SSAS)](http://www.mssqltips.com/sql-server-tip-category/28/analysis-services/).

### What are Measures and Measure Groups? What is the difference between them?

A Measure is any numeric quantity/value that represents a metric aligned to an organization's goals. This is the value which the business users are interested in, and are interested in viewing these values from different angles and different granularity levels. A measure is also commonly called a fact. The term "measures" and "facts" are used interchangeably.

A Measure Group is a collection/group of measures which belong to the same underlying fact table. In SSAS, typically each Measure Group is tied to each one of the underlying fact tables.

A Measure is single numeric value whereas a Measure Group is a collection of measures.

### What are the different types of Measures? Explain each one of them with an example.

Below are the most common types of measures/facts:

* **Fully Additive Facts:** These are facts which can be added across all the associated dimensions. For example, sales amount is a fact which can be summed across different dimensions like customer, geography, date, product, and so on.
* **Semi-Additive Facts:** These are facts which can be added across only few dimensions rather than all dimensions. For example, bank balance is a fact which can be summed across the customer dimension (i.e. the total balance of all the customers in a bank at the end of a particular quarter).  However, the same fact cannot be added across the date dimension (i.e. the total balance at the end of quarter 1 is $X million and $Y million at the end of quarter 2, so at the end of quarter 2, the total balance is only $Y million and not $X+$Y).
* **Non-Additive Facts:** These are facts which cannot be added across any of the dimensions in the cube. For example, profit margin is a fact which cannot be added across any of the dimensions. For example, if product P1 has a 10% profit and product P2 has a 10% profit then your net profit is still 10% and not 20%.  We cannot add profit margins across product dimensions. Similarly, if your profit margin is 10% on Day1 and 10% on Day2, then your net Profit Margin at the end of Day2 is still 10% and not 20%.
* **Derived Facts:** Derived facts are the facts which are calculated from one or more base facts, often by applying additional criteria. Often these are not stored in the cube and are calculated on the fly at the time of accessing them. For example, profit margin.
* **Factless Facts:** A factless fact table is one which only has references (Foreign Keys) to the dimensions and it does not contain any measures. These types of fact tables are often used to capture events (valid transactions without a net change in a measure value). For example, a balance enquiry at an automated teller machine (ATM). Though there is no change in the account balance, this transaction is still important for analysis purposes.
* **Textual Facts:** Textual facts refer to the textual data present in the fact table, which is not measurable (non-additive), but is important for analysis purposes. For example, codes (i.e. product codes), flags (i.e. status flag), etc.

### What is the purpose of Dimension Usage settings? Explain different types of relationships between Facts and Dimensions.

The Dimension Usage tab in the Cube Designer in [SQL Server Business Intelligence Development Studio](http://www.mssqltips.com/sqlservertutorial/204/business-intelligence-development-studio-bids/) defines the relationship between a Cube Dimension and a Measure Group (s). A Dimension which is related to one of more Measure Groups, directly/indirectly, is called as a Cube Dimension. A Cube Dimension is an instance of a database Dimension as explained in the [previous tip](http://www.mssqltips.com/sqlservertip/2683/sql-server-analysis-services-interview-questions-on-dimensions-hierarchies-and-properties/).

Following are the four different types of relationships between a Cube Dimension and a Measure Group:

* **Regular:** In a Regular relationship, primary key column of a dimension is directly connected to the fact table. This type of relationship is similar to the relationship between a dimension and a fact in a [Star Schema](http://www.mssqltips.com/sqlservertutorial/2004/creating-a-star-schema-using-a-data-source-view/), and it can be based on either the physical primary key-foreign key relationship in the underlying relational database or the logical primary key-foreign key relationship defined in the Data Source View.
* **Referenced:** In a Referenced relationship, primary key columns of a dimension is indirectly connected to the fact table through a key column in the intermediate dimension table. This type of relationship is similar to the indirect relationship between a dimension and a fact, through an intermediate dimension, in a Snowflake Schema.
* **Fact:** In a Fact relationship, the dimension table and the fact table are one and the same. Basically a [Fact Dimension or Degenerate Dimension](http://www.mssqltips.com/sqlservertip/2662/sql-server-analysis-services-interview-questions-part-ii--dimensions/)is created using one or more columns from the fact table and this degenerate dimension is used while defining/establishing the relationship in case of a fact relationship.
* **Many-to-Many:** In a Many-to-Many relationship, a dimension is indirectly connected to a Measure Group through an intermediate fact table which joins with the dimension table. It is analogous to a scenario, where one project can have multiple project managers and one project manager can manage multiple projects.

### What are Calculated Members? How do they differ from Measures?

[Calculated Members](http://www.mssqltips.com/sqlservertutorial/2013/developing-a-calculated-measure/) are members of a measure group and are defined based on a combination of one or more base measures, arithmetic/conditional operators, numeric values, and functions, etc. For example, profit is a calculated member/calculate measure, which is defined based on various base measures like selling price, cost, price, tax amount, freight amount, etc.

The value of a measure (base measure) is stored in a cube as part of the cube processing process. Whereas the value of a calculated member/measure is calculated on the fly in response to a user request and only the definition is stored in the cube.

### What are Named Sets? What are the two types of Named Sets?

A [Named Set](http://www.mssqltips.com/sqlservertutorial/2014/developing-named-sets/) is a set of dimension members (usually a subset of dimension members) and is defined using MDX (a Multidimensional Expression). Often Named Sets are defined for improved usability by the end users and client applications. Apart from that, they can also be used for various calculations at the cube level. Similar to calculated members/measures, named sets are defined using a combination of cube/dimension data, arithmetic operators, numeric values, functions, etc. Some of the examples of Named Sets are top 50 customers, top 10 products, top 5 students, etc.

Named Sets are of two types: Static Named Sets and Dynamic Named Sets.

Static Named Sets, when defined in cube, are evaluated during cube processing process. Dynamic Named Sets are evaluated each time the query is invoked by the user.

### What are KPIs? What are the different properties associated with a KPI?

KPI stands for [Key Performance Indicator](http://www.mssqltips.com/sqlservertip/2398/ssrs-2008-r2-kpis-with-bullet-graphs/). A KPI is a measure of an organization's performance in a pre-defined area of interest. KPIs are defined to align with the pre-defined organizational goals and help the business decision makers gain insights into their business performance.

Often KPIs have the following five commonly used properties:

* **Name:** Indicates the name of the Key Performance Indicator.
* **Actual/Value:** Indicates the actual value of a measure pre-defined to align with organizational goals.
* **Target/Goal:** Indicates the target value (i.e. goal) of a measure pre-defined to align with organizational goals.
* **Status:** It is a numeric value and indicates the status of the KPI like performance is better than expected, performance is as expected, performance is not as expected, performance is much lower than expected, etc.
* **Trend:** It is a numeric value and indicates the KPIs trend like performance is constant over a period of time, performance is improving over a period of time, performance is degrading over a period of time, etc.

Apart from the above listed properties, most of the times, KPIs contain the following two optional properties:

* **Status Indicator:** It is a graphical Indicator used to visually display the status of a KPI. Usually colors like red, yellow, and green are used or even other graphics like smiley or unhappy faces.
* **Trend Indicator:** It is a graphical indicator used to visually display the trend of a KPI. Usually up arrow, right arrow, and down arrow are used.

### What are Actions in SSAS? What are the different types of Actions in SQL Server Analysis Services?

Actions in SSAS allow us to extend the cube functionality and enable the users to interact with the cube. An Action in simple terms is basically an event, which can be initiated by a user/application and it can take various forms depending upon the type of Action defined.

Actions are primarily of following three types:

* **Drillthrough Actions:** A [Drillthrough Action](http://www.mssqltips.com/sqlservertip/2491/enabling-drillthrough-in-analysis-services/)retrieves the detail level information associated with the cube data based on which the Drillthrough Action is defined.
* **Reporting Actions:** A Reporting Action retrieves an SSRS report which is associated with the cube data. The command which invokes the SSRS report contains the report URL along with the report parameters.
* **Standard Actions:** A Standard Action retrieves the action element associated with the cube data. Standard actions are further categorized into 5 different subcategories and the action element varies for each of these subcategories. The following are the types of Standard Actions:
  + Dataset Action: Returns a dataset to the client application and the action content is an MDX expression.
  + Proprietary Action: Performs an operation as defined by the client application. The action content for this type of action is specific to the calling client application and the client application is responsible for interpreting the meaning of the Action.
  + Rowset Action: A Rowset Action returns a Rowset to the client application. The action content is a command to retrieve the data.
  + Statement Action: The action content for this type of Action is an OLE DB command and it returns a command string to the client application.
  + URL Action: The Action Content for this type of action is an URL and it returns a URL to the client application which can be opened usually in a web browser. This is the default action.

### What are partitions in cubes? How do they different from table partitions at a SQL Server database level?

A partition is physical storage space which contains either all or a portion of measure group data. Each measure group in SSAS has one partition by default.

A partition can be either bound to a table in the underlying relational database or a query pointing to the table(s) in the underlying database and has filters in it.

In terms of storage, cube partitions in SSAS and [table partitions](http://www.mssqltips.com/sql-server-tip-category/65/partitioning/) in a database are similar. Both these types of partitions are used to improve the performance. However, partitions in SSAS offer additional benefits including:

* Each partition can be processed separately (i.e. a measure group can be split across multiple partitions, for example one partition for each year). Only the partitions in which data has been modified can be processed thereby improving the processing time of the cube.
* Partitions provide improved manageability by allowing us to define storage mode, aggregation design, etc. at the partition level and these settings can vary between different partitions belonging to the same measure group.

### What are the different Storage Modes supported by Cube Partitions?

There are primarily two types of data in SSAS: summary and detail data. Based on the approach used to store each of these two types of data, there are three standard storage modes supported by partitions:

* **ROLAP:** ROLAP stands for Real Time Online Analytical Processing. In this storage mode, summary data is stored in the relational data warehouse and detail data is stored in the relational database. This storage mode offers low latency, but it requires large storage space as well as slower processing and query response times.
* **MOLAP:** MOLAP stands for Multidimensional Online Analytical Processing. In this storage mode, both summary and detail data is stored on the OLAP server (multidimensional storage). This storage mode offers faster query response and processing times, but offers a high latency and requires average amount of storage space. This storage mode leads to duplication of data as the detail data is present in both the relational as well as the multidimensional storage.
* **HOLAP:** HOLAP stands for Hybrid Online Analytical Processing. This storage mode is a combination of ROLAP and MOLAP storage modes. In this storage mode, summary data is stored in OLAP server (Multidimensional storage) and detail data is stored in the relational data warehouse. This storage mode offers optimal storage space, query response time, latency and fast processing times.

There are different variations of these Standard Storage Modes. Visit [this msdn article](http://msdn.microsoft.com/en-us/library/ms175646(v=sql.100).aspx) for more details.

### What is proactive caching in SQL Server Analysis Services?

Proactive caching is an advanced feature in SSAS and it enables a cube to reflect the most recent data present in the underlying database by automatically refreshing the cube based on the predefined settings. This feature allows the users to view the data in near real-time.

Proactive caching can be configured to refresh the cache (MOLAP cache) either on a pre-defined schedule or in response to an event (change in the data) from the underlying relational database. Proactive caching settings also determine whether the data is queried from the underlying relational database (ROLAP) or is read from the outdated MOLAP cache, while the MOLAP cache is rebuilt.

Proactive caching helps in minimizing latency and achieve high performance.

|  |  |
| --- | --- |
| MDX Expression Cheat Sheet | |
| **Problems** | **Calculation Expression** |
| Simple Gross Profit Calculation | [Measures].[Sales Amount] - [Measures].[Total Product Cost] |
| Sales in the USA | ([Measures].[Sales Amount], [Customer].[Country].&[United States]) |
| Year to Date Sales (Works for any level of Date Hiearchy) | Aggregate(             PeriodsToDate( [Date].[Calendar Hierarchy].[Year],             [Date].[Calendar Hierarchy].CurrentMember ),             ([Measures].[Sales])) |
| Alternate Year To Date Expression (YTD, QTD, MTD) | AGGREGATE (             YTD ([Date Order].[Calendar].CurrentMember),             Measures.[Sales Amount]) |
| Product Ranking | IIF (Product.Product.CurrentMember IS Product.Product.[All],NULL, IIF (Measures.[Sales Amount] = 0, NULL,  RANK(Product.Product.CurrentMember, ORDER (Product.Product.Members, Measures.[Sales Amount])))) |
| Sales from 365 Days Ago | (ParallelPeriod([Invoice Date].[Date Hierarchy].[Date], 365, [Invoice Date].[Date Hierarchy].CurrentMember), [Measures].[Sales Amount]) |
| Sales in the Previous Period | (Measures.[Sales Amount], [Date Order].[Calendar].CurrentMember.PrevMember) |
| Top 10 Selling Products (Named Set) | TopCount (Product.Product.Product.Members,10,Measures.[Sales Amount]) |
| Three Years Average Sales From NOW() | Avg( {ParallelPeriod( [Date].[Date].[Year], 3, StrToMember("[Date].[Date].&[" + Format(now(), "yyyyMMdd") + "]")) : StrToMember("[Date].[Date].&[" + Format(now(), "yyyyMMdd") + "]")} , [Measures].[Sales Count]) |
| Drillthrough Action Caption | 'Get Sales Details for' +[Product].[Product].CurrentMember.Member\_Caption |
| Change SSAS Calculation Text color | IIF([Measures].[Profit Percentage] < .40, 255 , 0) |
| Changing a Calculation with a SCOPE statement | SCOPE ([Measures].[Scope Profit]); THIS = ([Measures].[Sales Amount] - [Measures].[Standard Product Cost]); END SCOPE; |
| Clear Ration Value when at all level | SCOPE ([Customer].[Customer Geography].[All], Measures.RatioOverParent); THIS = NULL |
| SSAS KPI Value Expression | [Measures].[Sales Amount] \* 1.2 |
| SSAS KPI Goal Expression | Case     When IsEmpty          (ParallelPeriod            ([Date Order].[Fiscal].[Fiscal Year],              1,[Date Order].[Fiscal].CurrentMember))     Then [Measures].[Sales Amount]     Else 1.10 \*          ([Measures].[Sales Amount],           ParallelPeriod([Date Order].[Fiscal].[Fiscal Year],           1,[Date Order].[Fiscal].CurrentMember)) End |
| SSAS KPI Status Expression | Case     When KpiValue( "Sales Revenue YTD" ) / KpiGoal( "Sales RevenueYTD" ) > 1     Then 1     When KpiValue( "Sales Revenue YTD" ) / KpiGoal( "Sales Revenue YTD" ) <= 1                And                 KpiValue( "Sales Revenue YTD" ) / KpiGoal( "Sales Revenue YTD" ) >= .85     Then 0     Else -1 End |
| SSAS KPI Trend Expression | Case     When IsEmpty          (ParallelPeriod            ([Date Order].[Fiscal].[Fiscal Year],              1,[Date Order].[Fiscal].CurrentMember))     Then 0      When VBA!Abs          ((KpiValue( "Sales Revenue YTD" )- (KpiValue( "Sales Revenue YTD" ),             ParallelPeriod( [Date Order].[Fiscal].[Fiscal Year],                 1, [Date Order].[Fiscal].CurrentMember)))           /(KpiValue( "Sales Revenue YTD" ),             ParallelPeriod             ( [Date Order].[Fiscal].[Fiscal Year],               1,[Date Order].[Fiscal].CurrentMember))) <=.02     Then 0     When (KpiValue( "Sales Revenue YTD" )- (KpiValue( "Sales Revenue YTD" ),              ParallelPeriod              ( [Date Order].[Fiscal].[Fiscal Year], 1,                [Date Order].[Fiscal].CurrentMember)))          /(KpiValue( "Sales Revenue YTD" ),            ParallelPeriod( [Date Order].[Fiscal].[Fiscal Year],1,[Date Order].[Fiscal].CurrentMember)) >.02     Then 1     Else -1 End |

**MDX**

*MultiDimensional eXpressions (MDX) language* provides a specialized syntax for querying and manipulating the multidimensional data stored in [OLAP cubes](http://en.wikipedia.org/wiki/OLAP_cube).[[](http://en.wikipedia.org/wiki/MultiDimensional_eXpressions#cite_note-1)

Multi-Dimensional eXpressions (MDX) is a language used for querying and extending capabilities of Analysis Services cubes. MDX is typically used for two purposes:

1. Creating reporting queries. Such constructs contain SELECT, FROM and WHERE clauses (among other elements) and are referred to as MDX queries or MDX statements.
2. Defining cube structures such as calculated members, named sets, actions, key performance indicators and so forth. Such constructs are referred to as MDX expressions. MDX expressions do not contain SELECT, FROM or WHERE clauses; they are used to define calculations using cube dimension members and measures.

If you examine a basic MDX statement it might seem that it contains parts similar to SQL statements: the SELECT, FROM and WHERE clauses. However, that is where the similarity ends. MDX is very different from SQL because it is written for traversing dimension hierarchies and defining cube cells. MDX doesn't support variables, parameters, cursors and other common SQL structures. On the other hand, MDX is more powerful than SQL when it comes to referencing hierarchy members.

When writing MDX queries you need to make a clear distinction between the data and metadata. Cube data is what you present to your users, for example you would show the sales amount for Ford Taurus 2005 and Toyota Camry 2004. Metadata is how the data is structured in your dimensions; for example, the product dimension could have levels of brand, make, model and year. It is important to realize if you are trying to refer to data or to metadata within MDX because different functions are used to address each need. For example, if you wanted to return all brands within your product dimension you would use the "members" function, as in [product].[brand].members. On the other hand, if you wanted to see all makes within a certain brand you would use the "children" function by providing the name of the parent, as in [product].[brand].[Toyota].children.

A **member** is a value stored in a dimension. For example, [time].[January 8th, 2009] could be a member in a time dimension. Product dimension can contain members [product].[Toyota].[camry].[2003], [product].[Nissan].[maxima].[2004], [product].[Ford].[mustang].[2005], [product].[Cadillac].[Escalade].[2005] and so on. Note that each member and dimension name is enclosed in square brackets. MDX requires square brackets if a member contains a space or is a number. Members that do not contain any spaces and aren't numbers can be represented without brackets. However, for the sake of consistency you should enclose every member in square brackets. With MSAS 2005 MDX allows you to refer to a dimension member without specifying the hierarchy where the member is found. For example, [time].[December] is a valid member; however, you should include the hierarchy name to avoid confusion and potential errors. For example, you could have [fiscal] hierarchy and [calendar] hierarchy within your time dimension. Furthermore, each year included in your time dimension is likely to have the month of December. Therefore be sure to type [time].[calendar].[2005].[December] to specify exactly which member you want to reference in your queries.   
  
  
  
A **tuple** is a combination of members from one or multiple dimensions; a tuple can only contain ONE member from each dimension. A tuple that contains a member from a single dimension is synonymous to a member. For example, ([time].[January 8th, 2009]) would be a tuple even though it refers to only one dimension. A tuple that refers to multiple dimensions could be ([product].[Cadillac], [time].[January 8th, 2009]). If you attempt to reference more than one member from each dimension the tuple will become invalid; for example ([product].[Cadillac], [time].[January], [time].[February]) would be incorrect. As a rule tuples are enclosed in parenthesis; if a tuple is comprised of a single member then parenthesis aren't necessary; but, for the sake of consistency you should always use parenthesis when referring to a tuple. If you wanted to compare a cube with a spreadsheet a tuple would be equivalent of a cell in a spreadsheet file; that is why cell B30 on a spreadsheet identifies only a single cell the intersection of column B and row 30. Note however, that the intersection of column B and row 30 will occur in each sheet within your Excel file. Similarly ([product].[Cadillac], [time].[January 8th, 2009]) has intersections with other dimensions the same product could be sold in multiple stores, by different salespeople, in different cities, financed by different banks. Unless you include all cube dimensions in a tuple its definition would be incomplete. So how do we decide exactly which members to include in the tuple? If the dimension member isn't explicitly specified Analysis Services assumes the default member. Typically default member for each dimension is the [All] member, so ([product].[Cadillac], [time].[January 8th, 2009]) references sales of Cadillac on January 8th of 2009 by all salespeople in all cities regardless of the financial institution that paid for the vehicle.   
  
  
  
A **set** is a collection of zero, one or multiple tuples that have the same dimensionality. All tuples enclosed in a set must reference the same dimensions in the same order. For example ([product].[Cadillac], [time].[January]) cannot be combined with ([time].[February], [product].[Ford]) because they specify dimensions in reverse order. Nor could you combine ([time].[March], [product].[Toyota]) with ([geography].[northern USA], [product].[Audi] ) because the two reference different dimensions. However, {([product].[Cadillac], [time].[January]), ([product].[Audi], [time].[December])} is a valid set. Since a set can consist of a single tuple, and a tuple can consist of a single member the following is a perfectly valid set {([time].[January 8th, 2009])}.  
  
  
  
A set of zero tuples is an empty set. MDX requires enclosing sets in curly braces ({}). Some built-in MDX functions return sets; if this is the case you don't have to enclose the output of the function in curly braces. For the sake of consistency you should enclose all your sets in curly braces

**KPI examples**

**[**[**edit**](http://en.wikipedia.org/w/index.php?title=Performance_indicator&action=edit&section=5)**] Marketing**

Some examples are:

1. New [customers](http://en.wikipedia.org/wiki/Customer) acquired
2. Demographic analysis of individuals (potential customers) applying to become customers, and the levels of approval, rejections, and pending numbers
3. Status of existing customers
4. [Customer attrition](http://en.wikipedia.org/wiki/Customer_attrition)
5. [Turnover](http://en.wikipedia.org/wiki/Revenue) (i.e., revenue) generated by segments of the customer population
6. Outstanding balances held by segments of customers and terms of payment
7. Collection of bad debts within customer relationships
8. Profitability of customers by demographic segments and segmentation of customers by profitability

Many of these customer KPIs are developed and managed with [customer relationship management](http://en.wikipedia.org/wiki/Customer_relationship_management) software.

Faster availability of data is a competitive issue for most organizations. For example, businesses which have higher operational/credit risk (involving for example credit cards or wealth management) may want weekly or even daily availability of KPI analysis, facilitated by appropriate IT systems and tools.

**[**[**edit**](http://en.wikipedia.org/w/index.php?title=Performance_indicator&action=edit&section=6)**] Manufacturing**

[Overall equipment effectiveness](http://en.wikipedia.org/wiki/Overall_equipment_effectiveness), is a set of broadly accepted non-financial metrics which reflect manufacturing success.

* **Cycle Time** – Cycle time is the total time from the beginning to the end of your process, as defined by you and your customer. Cycle time includes process time, during which a unit is acted upon to bring it closer to an output, and delay time, during which a unit of work is spent waiting to take the next action.
* **Cycle Time Ratio** (CTR) – CTR = Standard Cycle Time / Real Cycle Time
* [Utilization](http://en.wikipedia.org/wiki/Utilization)
* [Rejection rate](http://en.wikipedia.org/wiki/Rejection_rate)

**[**[**edit**](http://en.wikipedia.org/w/index.php?title=Performance_indicator&action=edit&section=7)**] IT**

* [Availability](http://en.wikipedia.org/wiki/Availability)
* [Mean time between failure](http://en.wikipedia.org/wiki/Mean_time_between_failure)
* [Mean time to repair](http://en.wikipedia.org/wiki/Mean_time_to_repair)
* [Unplanned availability](http://en.wikipedia.org/w/index.php?title=Unplanned_availability&action=edit&redlink=1)

**[**[**edit**](http://en.wikipedia.org/w/index.php?title=Performance_indicator&action=edit&section=8)**] Supply Chain Management**

Businesses can utilize KPIs to establish and monitor progress toward a variety of goals, including lean manufacturing objectives, [minority business enterprise](http://en.wikipedia.org/wiki/Minority_business_enterprise) and diversity spending, environmental "green" initiatives, [cost avoidance](http://en.wikipedia.org/w/index.php?title=Cost_avoidance&action=edit&redlink=1) programs and [low-cost country sourcing](http://en.wikipedia.org/wiki/Low-cost_country_sourcing) targets.

Any business, regardless of size, can better manage supplier performance with the help of KPIs robust capabilities, which include:

* Automated entry and approval functions
* On-demand, real-time scorecard measures
* Rework on procured inventory.
* Single data repository to eliminate inefficiencies and maintain consistency
* Advanced workflow approval process to ensure consistent procedures
* Flexible data-input modes and real-time graphical performance displays
* Customized [cost savings documentation](http://en.wikipedia.org/w/index.php?title=Cost_savings_documentation&action=edit&redlink=1)
* Simplified setup procedures to eliminate dependence upon IT resources.

Main SCM KPIs will detail the following processes:

* Sales forecasts
* Inventory
* Procurement and suppliers
* Warehousing
* Transportation
* Reverse logistics

Suppliers can implement KPIs to gain an advantage over the competition. Suppliers have instant access to a user-friendly portal for submitting standardized cost savings templates. Suppliers and their customers exchange vital supply chain performance data while gaining visibility to the exact status of cost improvement projects and [cost savings documentation](http://en.wikipedia.org/w/index.php?title=Cost_savings_documentation&action=edit&redlink=1).

**[**[**edit**](http://en.wikipedia.org/w/index.php?title=Performance_indicator&action=edit&section=9)**] Government**

The provincial government of [Ontario, Canada](http://en.wikipedia.org/wiki/Ontario,_Canada) has been using KPIs since 1998 to measure the performance of higher education institutions in the province. All post secondary schools collect and report performance data in five areas – graduate satisfaction, student satisfaction, employer satisfaction, employment rate, and graduation rate.[[3]](http://en.wikipedia.org/wiki/Key_performance_indicator#cite_note-3)

**[**[**edit**](http://en.wikipedia.org/w/index.php?title=Performance_indicator&action=edit&section=10)**] Further performance indicators**

* Duration of a [stockout](http://en.wikipedia.org/wiki/Stockout) situation
* Customer order waiting time

**[**[**edit**](http://en.wikipedia.org/w/index.php?title=Performance_indicator&action=edit&section=11)**] Problems**

In practice, overseeing key performance indicators can prove expensive or difficult for organizations. Some indicators such as staff morale may be impossible to quantify. As such dubious KPIs can be adopted that can be used as a rough guide rather than a precise benchmark

## What languages BI uses to achieve the goal?

BI uses following languages for achieve the Goal.

MDX – Multidimensional Expressions:

This language is used for retrieving data from SSAS cubes. It looks very similar to T-SQL, but it is very different in the areas of conceptualization and implementation.

DMX – Data Mining Extensions:

This is again used for SSAS, but rather than cubes it is used for data mining structures. This language is more complicated than MDX. Microsoft has provided many wizards in its BI tools, which further reduced number of experts for learning this language, which deals with data mining structures.

XMLA – XML for Analysis:

This is mainly used for SSAS administrative tasks. It is quite commonly used in administration tasks such as backup or restore database, copy and move database, or for learning Meta data information. Again, MS BI tools provide a lot of wizards for the same.