Smarten
Augmented Analytics
Powered by ElegantJ BI

Concept Manual

Version 5.1

Document Version: 11.0
Product Version: 5.1

Document Information	
Document ID	Smarten-Concept-Manual
Document Version	Version 11.0
Product Version	Version 5.1
Date	06-December-2020
Recipient	NA
Author	EMTPL

Statement of Confidentiality, Disclaimer and Copyright

This document contains information that is proprietary and confidential to EMTPL, which shall not be disclosed, transmitted, or duplicated, used in whole or in part for any purpose other than its intended purpose. Any use or disclosure in whole or in part of this information without the express written permission of EMTPL is prohibited.

Any other company and product names mentioned are used for identification purpose only, may be trademarks of their respective owners and are duly acknowledged.

Disclaimer

This document is intended to support administrators, technology managers or developers using and implementing Smarten. The business needs of each organization will vary and this document is expected to provide guidelines and not rules for making any decisions related to Smarten. The overall performance of Smarten depends on many factors, including but not limited to hardware configuration and network throughput.

Contents

1 About this Document

This manual explains the concepts required to use the features in Smarten Augmented Analytics.

Users with no prior experience with Augmented Analytics software can refer to this guide to learn and understand the concepts of Augmented Analytics in Smarten. Users who have experience with other BI tools can refer to this guide to map the Augmented Analytics functions to the Smarten features and understand the concepts from a logical angle.

1.1 Scope and Organization of Topic Areas

Chapter 2	Introducing Smarten
Chapter 3	Designing the Data Model
Chapter 4	Analytic Functions
Chapter 5	Filters & Expressions
Chapter 6	KPI
Chapter 7	Social BI
Chapter 8	Access Rights & Security
Chapter 9	Delivery & Publishing Agent
Chapter 10	Product and Support Information

1.2 Conventions Used

This manual uses typographical conventions in the text to help you distinguish between the names of files, instructions, and other important notes that are relevant during installation. For example:

- Important notes are indicated in a different font colour as shown in the example below.

> Note:
> Apart from the data types listed above, other data types that are supported by a specific database connection driver can also be supported by Smarten cubes.

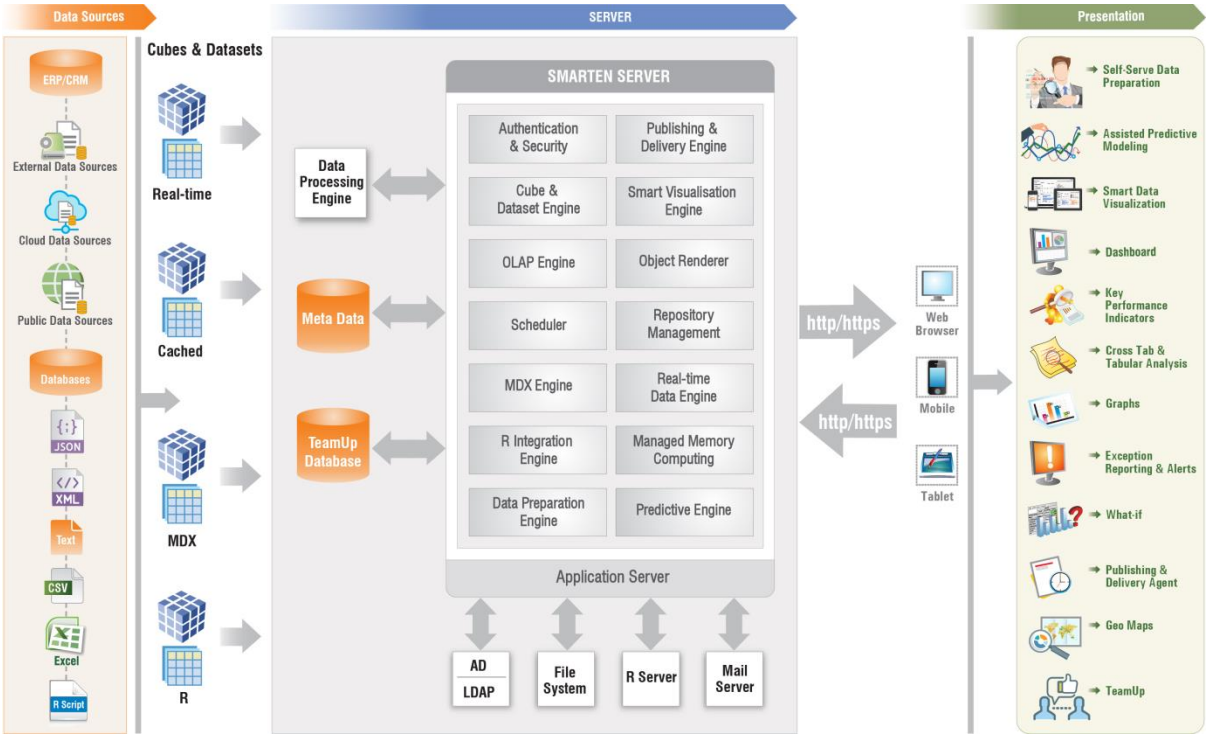
- References to documents are highlighted as below:

Reference: **Concept Manual > Designing the Data Model > Cube Generation Process-Extraction from CSV or flat files**

2 Introducing Smarten

Augmented Analytics is a set of enterprise scale applications for gathering, indexing, storing, and analyzing data from various data sources and applications. It converts data into intelligent information, leading to a smarter and agile decision-making process.

The integrated set of comprehensive features and functions in Smarten Augmented Analytics delivers actionable information to end users through dashboards, KPI, crosstab, graphs, GeoMap, and tabular.



HIGHER LEVEL ARCHITECTURE—SMARTEN

Using Smarten, users can access and analyze multidimensional data from multiple data sources such as RDBMS, text / csv files, and MDX data sources, using both real-time and cache cube architecture.

Easy-to-use tools, such as dashboards, crosstab, graphs, GeoMap, tabular, KPI, alerts, and integrated delivery & publishing agent, are built on a “zero-footprint” browser–based Smarten user interface and can be accessed through the supported browsers on desktops, laptops, tablets, and smartphones.

Smarten supports unique Managed Memory Computing that lets you choose data which will be used in-memory processing. Please refer to technical documents related to Managed Memory Computing for more details.

3 Designing the Data Model

This chapter details the basics of extracting data from different data sources, designing multidimensional objects called cubes, and preparing your data for crosstab, graphs, GeoMap, KPI, tabular, and dashboards.

Smarten supports both real-time and cache cube architecture. There is also an option for aggregation in cache cubes, and user can choose if user wants to perform aggregation for cache cubes at cube level or not.

Cache cubes will store indexed, pre-aggregated data along with metadata in the cubes. MDX and Real-time cubes will store only metadata information and will not store any data in the cubes.

Please refer to following documents for more information on different cubes types and architecture.

Reference: **Smarten-Working with Real time Cubes**

Reference: **Smarten-Working with SSAS MDX Cubes**

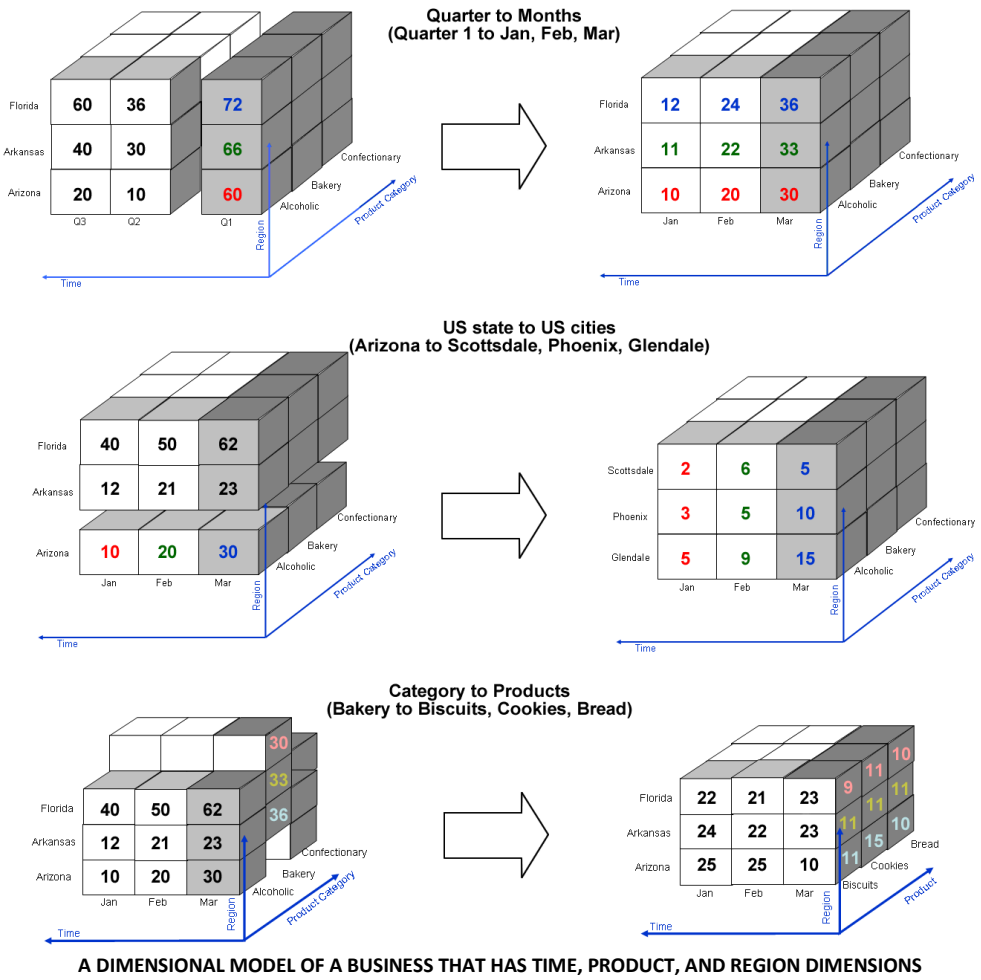
Reference: **Impact-of-Cube-Design-on-Performance > Cube type selection recommendations**

3.1 Cube Meta Data

Smarten's data extraction and cube management feature connects to the data sources to retrieve and transform the data based on logical rules and then loads that onto multidimensional cubes.

The cubes in Smarten are the main source of the data extracted from various data sources. They are indexed with multidimensional data structure and optimized for high performance, high speed, high-volume queries, and analysis needs for quick and uniform response times.

The following sections explain the underlining concepts of cube structure, such as dimensions, measures, time series, dimension hierarchy, and linked cubes.



A DIMENSIONAL MODEL OF A BUSINESS THAT HAS TIME, PRODUCT, AND REGION DIMENSIONS

3.1.1 Dimensions

Dimensions are the axes of a cube, representing x, y, and z coordinates. Aggregation of data with respect to more than one dimension is called multidimensional data.

In the above figure, time, product category, and region are three dimensions of sales.

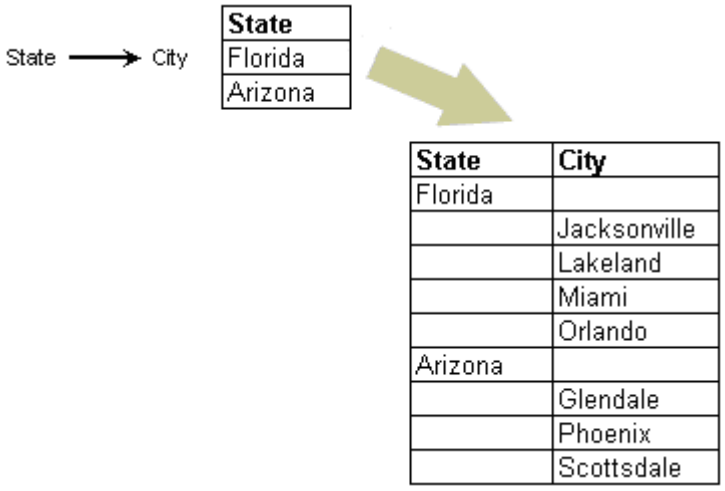
3.1.2 Measures

A measure is the scale or quantity of a dimension.
In the above figure, measure is denoted by the numeric sales quantity in different colours.

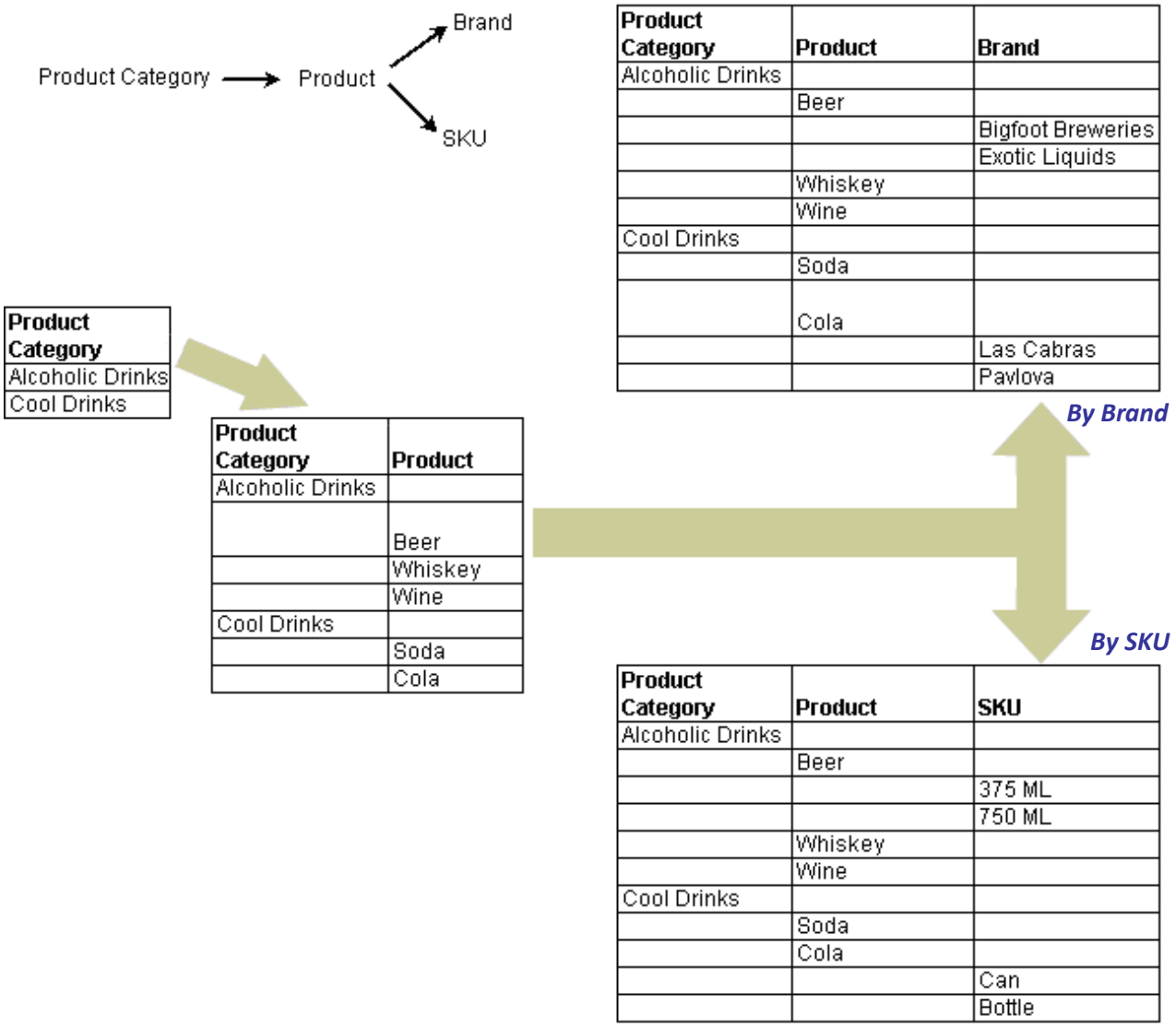
3.1.3 Dimension Hierarchy

The dimension hierarchy refers to the hierarchical levels of data within a dimension map. Dimension maps can be defined at the cube level and enable automatic drill down and drill up to users.

Dimension mapping means assigning the cube dimensions in terms of hierarchical levels. In the following illustration, the dimension city is mapped under the dimension state.



DIMENSION HIERARCHY SHOWING RELATIONSHIP BETWEEN STATE AND THEIR RESPECTIVE CITIES



Product Category
Alcoholic Drinks
Cool Drinks

Product Category	Product
Alcoholic Drinks	
	Beer
	Whiskey
	Wine
Cool Drinks	
	Soda
	Cola

Product Category	Product	Brand
Alcoholic Drinks		
	Beer	
		Bigfoot Breweries
		Exotic Liquids
	Whiskey	
	Wine	
Cool Drinks		
	Soda	
	Cola	
		Las Cabras
		Pavlova

Product Category	Product	SKU
Alcoholic Drinks		
	Beer	
		375 ML
		750 ML
	Whiskey	
	Wine	
Cool Drinks		
	Soda	
	Cola	
		Can
		Bottle

DIMENSION HIERARCHY SHOWING MULTI-CHILDREN RELATIONSHIP BETWEEN PRODUCT CATEGORY, SKU, AND BRANDS

3.2 Smarten Cache Cubes

3.2.1 Cube Generation Process

Smarten allows you to extract data from various transactional, historical, and reference data sources (e.g., CSV, a flat file, databases, and/or any other data source used for creating the cube), for example, ERP or CRM database, or monthly sales data as a flat file export from your ecommerce application and form cubes.

Cube extraction can be categorised broadly into two methods.

3.2.1.1 Extraction from Database

Typical steps for extraction from a database involves the following:

- Configuration of a database profile using JDBC/ODBC driver
- Connection with a database
- Select Aggregation or No Aggregation option
- Design SQL statement in two ways:
    - Graphical query designer
    - SQL query in editor
- Meta data definition for cube by defining and mapping data source columns and cube dimension and measure columns
- Define dimension map hierarchy

3.2.1.2 Extraction from CSV or flat files

CSV—Comma-separated values (also known as comma-separated variable) file format is a file type that stores tabular data.

Typical steps for extraction from CSV files involve the following:

- Locating CSV file and configuration of CSV data source profile
- Identify row and column separators in CSV file
- Identify field level parameters, such as data type, precision, length, scale, and format
- Meta data definition for cube by defining and mapping CSV data source columns and cube dimension and measure columns
- Define dimension map hierarchy

3.2.2 Cube Update Process

Once the cube is generated, the cube update process is used to append new data or refresh the cube with the most up-to-date data. It runs the predefined extraction query on data sources defined earlier and updates the cube according to the parameters selected.

3.2.2.1 Through Automatic Scheduler

For managing recency of the data used for multidimensional analysis, a cube should be regularly updated. The cube can be scheduled to regularly pull data from different data sources to update data at a given date/time. This is especially useful when the user knows exactly at what frequency the source data changes or when a particular type of data should be updated.

The scheduling of cube updates can be monthly, weekly, daily, hourly, or on a “as and when required” basis for some specific occurrence based on the business needs and time required to update the cube.

Examples:

Scheduler Frequency	Description
One time	Scheduler process is performed only one time on a specified date
Daily	Scheduler process is performed daily
At every “n” hours	Scheduler process is performed at every “n” number of hours. Ex: Every 2 hours
Weekly	Scheduler process is performed on a specific day of a week. Ex: Every Wednesday of the week
Monthly	Scheduler process is performed on a specific date of the month. Ex: Every 20th of the month
Yearly	Scheduler process is performed yearly on a specified date and month of the year. Ex: Every 15 June
Start time	Scheduler process is to be performed at a specific time; this can be achieved by setting the Start time in concurrence with One Time, Daily, Weekly, Monthly, or Yearly options. Ex: Schedule on 5 hours and 30 minutes daily
Term	Scheduler process is to be performed for a specific term; it is a period in which the scheduler is activated. Ex: From 1 Aug 2014 to 31 Dec 2014
Reoccurrence	Scheduler process is to be performed for some specific occurrence; it is used to end the scheduler process after some specific occurrence. Ex: End after 5 occurrences; it will end the scheduler process after 5 occurrences

Schedule On	Scheduler Frequency
One time on 1 January 2014 at 1 a.m.	One time: 1 January 2014 Start time: 1 hour 0 minute
Every night at midnight	Daily Start time: 0 hour 0 minute
Every Monday at 5 a.m.	Weekly: Monday Start time: 5 hour 0 minute

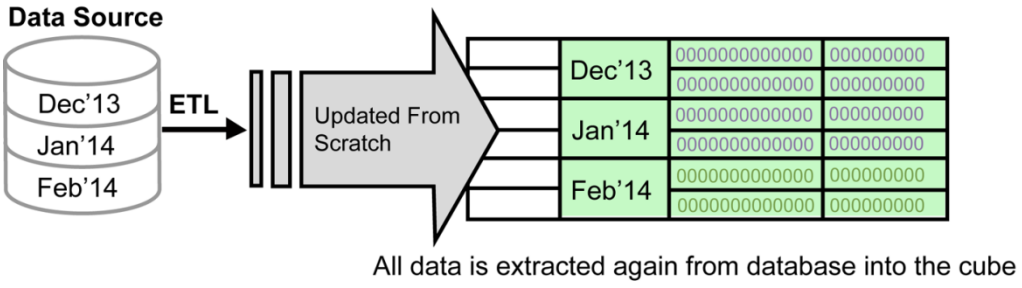
3.2.2.2 Through Manual Process

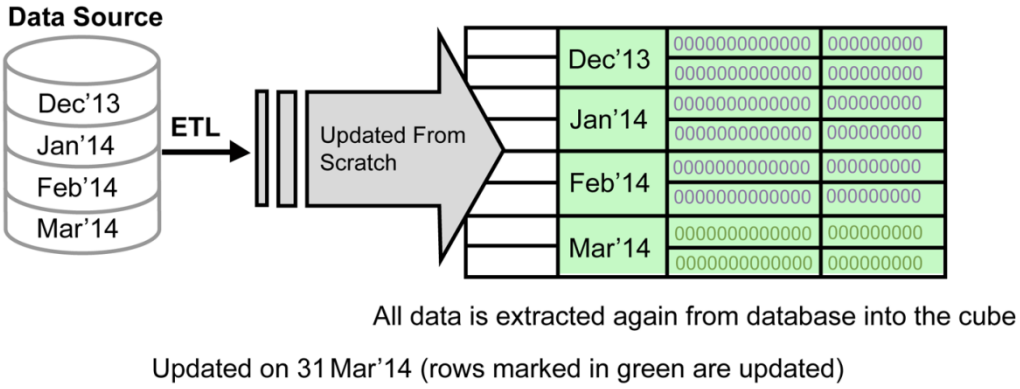
This option helps the user to manually update a cube on an “as and when required” basis.

3.2.2.3 Types of Cube Updates—From scratch or incremental

3.2.2.3.1 From scratch update

When updating a cube from scratch, the cube is rewritten with all the records from the data source.

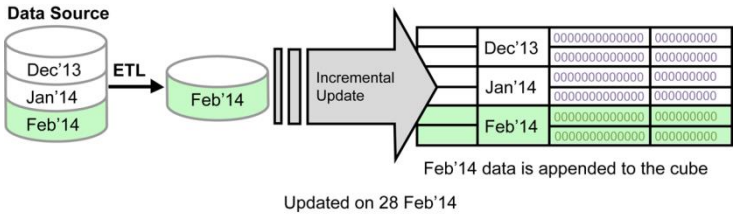


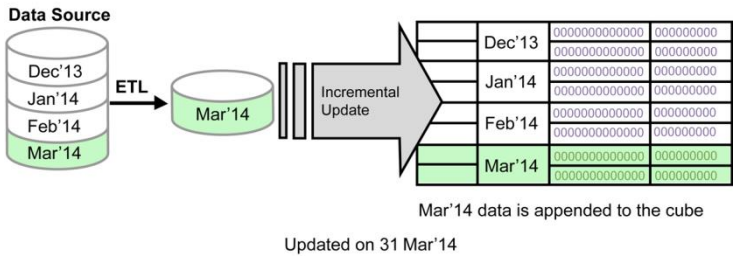


UPDATE FROM SCRATCH

3.2.2.3.2 Incremental update

With an incremental update, only the extracted data from the source is appended to the cube.





INCREMENTAL UPDATE

User can update cube with incremental option. In incremental option, system retrieves data from data source and appends only new data into the cube. Smarten supports two options for incremental update, one is, **append all rows retrieved from data source** and another is, **append new rows identified based on unique ID column**.

For example, if you have selected the 'ID' column as a unique column from a cube and the highest value in that column is '250' in the cube. When you update the cube, the system retrieves only those

records that have value greater than '250' in the 'ID' column and appends that data to the cube. Same way, if you have selected the 'Date' column as a unique column from a cube and the highest value in that column is '10-10-2020' in the cube, When you update the cube, the system retrieves only those records that have value greater than '10-10-2020' in the 'Date' column and appends that data to the cube.

3.2.3 Time Dimensions

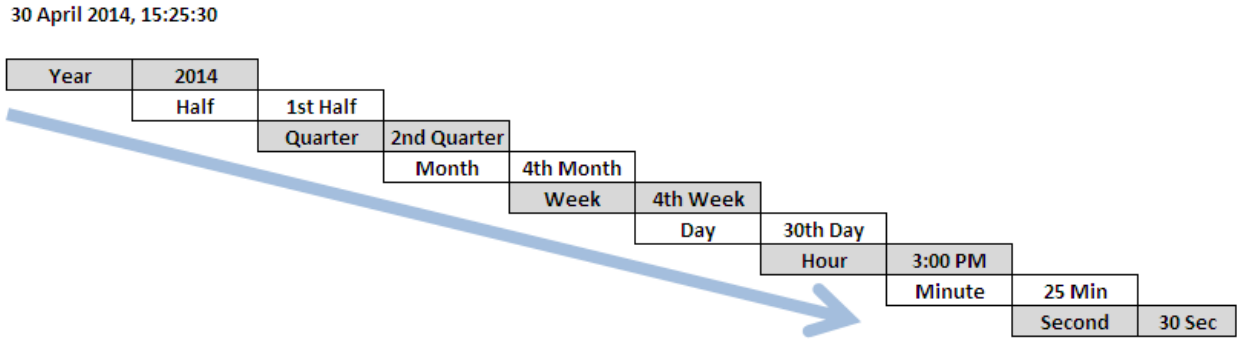
Cubes usually need a time dimension for the time period–related queries that look at the periods—weeks, months, or years. Time dimension is a descriptive attribute about the date/time stamp field in the cube, for example, day of a week, a month, a quarter, a year, etc.

Using time dimension, the hierarchical levels for the drill-down path is **Year > Half Year > Quarter > Month > Week > Day > Hour > Minute > Seconds**. The time dimension for a company can be based on either a calendar or a financial year.

3.2.3.1 Time Dimension Hierarchy

Time dimension hierarchy can be used to drill down from a summarized data of a year to a data up to a second level.

As shown in the following example, users can drill down from a year up to a second level using time dimension hierarchy. It is possible for time dimension to be based on both a calendar year and a financial year.



THE TIME DIMENSION HIERARCHY

> Note:
> The calendar year starts from 1 January

3.2.3.2 Time dimension based on a calendar year

A calendar year starts on 1 January and ends on 31 December of each year.

For any calendar year, the time dimension hierarchy would be as follows:

Year	Half 1	**Quarter 1**	January
			February
			March
		Quarter 2	April
			May
			June
	Half 2	**Quarter 3**	July
			August
			September
		Quarter 4	October
			November
			December

TIME DIMENSION HIERARCHY BASED ON A CALENDAR YEAR

3.2.3.3 Time dimension based on a financial year

This is to facilitate the users whose financial year starts on a date other than 1 January.

Example: The financial year from 1 April through 31 March.

The time dimension hierarchy would be as follows:

Financial Year	Half 1	**Quarter 1**	April
			May
			June
		Quarter 2	July
			August
			September
	Half 2	**Quarter 3**	October
			November
			December
		Quarter 4	January
			February
			March

TIME DIMENSION HIERARCHY BASED ON A FINANCIAL YEAR STARTING FROM 1 APRIL

Examples:

Date	Financial Year (Financial Year starts from 1 Apr 2014)	Calendar Year (Calendar Year starts from 1 Jan 2014)
15 July 2014	Year 2014 Quarter2 Month4 Week3	Year 2014 Quarter3 Month7 Week3
1 May 2014	Year 2014 Quarter1 Month2 Week1	Year 2014 Quarter2 Month5 Week1
10 Feb 2014	Year 2013 Quarter4 Month11 Week2	Year 2014 Quarter1 Month2 Week2
15 Dec 2014	Year 2014 Quarter3 Month9 Week3	Year 2014 Quarter4 Month12 Week3
2 Jan 2014 22:15.30	Year 2013 Quarter4 Month10 Week1 Hour22 Minute15 Second30	Year2012 Quarter1 Month1 Week1 Hour22 Minute15 Second30

3.2.4 Custom Cube Columns

3.2.4.1 Custom Cube Dimension

The custom cube dimension column is a new cube column created based on existing cube columns. The administrator can create cube columns not existing in the data source (database, CSV, or any other data source used for creating the cube).

Administrators can create new custom cube dimension by performing various string, arithmetic, date, statistics, trigonometry, or conditional functions using arithmetic operators (such as +, -, /, etc.) or comparison operators (such as =, >, < etc.) on two or more existing cube columns.

State	City	Code	Product	Qty	Mon
Arizona	Scottsdale	1001	Alcoholic	1	Jan
Arizona	Phoenix	1001	Alcoholic	4	Jan
Arizona	Glendale	1001	Alcoholic	7	Jan
Arizona	Scottsdale	1001	Alcoholic	2	Feb
Arizona	Phoenix	1001	Alcoholic	5	Feb
Arizona	Glendale	1001	Alcoholic	8	Feb
Arizona	Scottsdale	1001	Alcoholic	2	Mar
Arizona	Phoenix	1001	Alcoholic	4	Mar
Arizona	Glendale	1001	Alcoholic	1	Mar
Arizona	Scottsdale	1002	Beverages	2	Jan
Arizona	Phoenix	1002	Beverages	3	Jan
Arizona	Glendale	1002	Beverages	6	Jan
Arizona	Scottsdale	1002	Beverages	1	Feb
Arizona	Phoenix	1002	Beverages	4	Feb
Arizona	Glendale	1002	Beverages	7	Feb
Arizona	Scottsdale	1002	Beverages	3	Mar
Arizona	Phoenix	1002	Beverages	5	Mar
Arizona	Glendale	1002	Beverages	2	Mar

Product ID = Product-Code

State	City	Code	Product	ProductID	Qty	Mon
Arizona	Scottsdale	1001	Alcoholic	1001- Alcoholic	1	Jan
Arizona	Phoenix	1001	Alcoholic	1001- Alcoholic	4	Jan
Arizona	Glendale	1001	Alcoholic	1001- Alcoholic	7	Jan
Arizona	Scottsdale	1001	Alcoholic	1001- Alcoholic	2	Feb
Arizona	Phoenix	1001	Alcoholic	1001- Alcoholic	5	Feb
Arizona	Glendale	1001	Alcoholic	1001- Alcoholic	8	Feb
Arizona	Scottsdale	1001	Alcoholic	1001- Alcoholic	2	Mar
Arizona	Phoenix	1001	Alcoholic	1001- Alcoholic	4	Mar
Arizona	Glendale	1001	Alcoholic	1001- Alcoholic	1	Mar
Arizona	Scottsdale	1002	Beverages	1002- Beverages	2	Jan
Arizona	Phoenix	1002	Beverages	1002- Beverages	3	Jan
Arizona	Glendale	1002	Beverages	1002- Beverages	6	Jan
Arizona	Scottsdale	1002	Beverages	1002- Beverages	1	Feb
Arizona	Phoenix	1002	Beverages	1002- Beverages	4	Feb
Arizona	Glendale	1002	Beverages	1002- Beverages	7	Feb
Arizona	Scottsdale	1002	Beverages	1002- Beverages	3	Mar
Arizona	Phoenix	1002	Beverages	1002- Beverages	5	Mar
Arizona	Glendale	1002	Beverages	1002- Beverages	2	Mar

Custom Cube Dimension field ProductID created by expression [Concatenate (Code+'-'+Product)] has been added.

CUSTOM CUBE DIMENSION CREATION EXAMPLE

Note:
Custom cube dimensions are created by administrators on the cube data (aggregated result set of a cube).It is a one-time process after cube creation.

Once a custom cube dimension is defined, every time a cube is refreshed, this column is automatically created by the system. Users can use it like any other cube dimension in any BI objects (crosstab, graphs, GeoMap, KPI, tabular) derived from that cube.

3.2.4.2 Custom Cube Measure

Smarten's easy-to-build custom cube measures column can be created by building a numeric formula on existing cube columns. The cube columns not found in the data source (CSV, a flat file, database, or any other data source used for creating the cube) can be instantly created by the administrator.

The administrators can create custom cube measure columns from two or more existing numeric cube columns by performing various string, arithmetic, date, statistics, trigonometry, or conditional funtions using various arithmetic operators (such as +, -, /, etc.) or comparison operators (such as =, >, < etc.).

City	Product	Month	Quantity	Price
Scottsdale	AlcoholDrinks	January	12	15
Phoenix	AlcoholDrinks	January	22	18
Glendale	AlcoholDrinks	January	13	18.5
Scottsdale	Bakery	January	45	7
Phoenix	Bakery	January	55	4
Glendale	Bakery	January	43	5
Scottsdale	AlcoholDrinks	February	11	14
Phoenix	AlcoholDrinks	February	20	16
Glendale	AlcoholDrinks	February	12	17
Scottsdale	Bakery	February	41	6
Phoenix	Bakery	February	50	4
Glendale	Bakery	February	39	5
Scottsdale	AlcoholDrinks	March	13	17
Phoenix	AlcoholDrinks	March	24	20
Glendale	AlcoholDrinks	March	14	20
Scottsdale	Bakery	March	50	8
Phoenix	Bakery	March	61	4
Glendale	Bakery	March	47	6

Gross Sales = Quantity x Price

City	Product	Month	Quantity	Price	GrossSales
Scottsdale	AlcoholDrinks	January	12	15	180
Phoenix	AlcoholDrinks	January	22	18	396
Glendale	AlcoholDrinks	January	13	18.5	240.5
Scottsdale	Bakery	January	45	7	315
Phoenix	Bakery	January	55	4	220
Glendale	Bakery	January	43	5	215
Scottsdale	AlcoholDrinks	February	11	14	154
Phoenix	AlcoholDrinks	February	20	16	320
Glendale	AlcoholDrinks	February	12	17	204
Scottsdale	Bakery	February	41	6	246
Phoenix	Bakery	February	50	4	200
Glendale	Bakery	February	39	5	195
Scottsdale	AlcoholDrinks	March	13	17	221
Phoenix	AlcoholDrinks	March	24	20	480
Glendale	AlcoholDrinks	March	14	20	280
Scottsdale	Bakery	March	50	8	400
Phoenix	Bakery	March	61	4	244
Glendale	Bakery	March	47	6	282

Custom cube measure field Gross Sales is created by the expression "Quantity x Price"

CUSTOM MEASURE CREATION EXAMPLE

> Note:
> Custom cube measures are created by administrators on cube data (aggregated result set of a cube). It is a one-time process after cube creation.
>
> Once the custom cube measure is defined, every time the cube is refreshed, this column is automatically created by the system. Users can use it like any other cube measures in any BI objects (crosstab, graphs, GeoMap, KPI, tabular) derived from that cube.

3.2.5 Linked Cubes

A linked cube combines records from two or more cache cubes, resulting in a new cube. Cubes can be linked by **UNION** (union query) or **JOIN** (join query).

> Note:
> Linked cube cannot be created from Real-Time and MDX cubes.

3.2.5.1 UNION

		Alcoholic drinks		
		Jan	Feb	Mar
Arizona	Scottsdale	1	2	2
	Phoenix	4	5	4
	Glendale	7	8	1

		Alcoholic drinks		
		Jan	Feb	Mar
Florida	Orlando	3	5	5
	Miami	6	5	9
	Lakeland	9	10	11

		Alcoholic drinks		
		Jan	Feb	Mar
Washington	Vancouver	12	7	5
	Seattle	8	5	1
	Redmond	5	9	5

		Alcoholic drinks		
		Jan	Feb	Mar
Arizona	Scottsdale	1	2	2
	Phoenix	4	5	4
	Glendale	7	8	1
Florida	Orlando	3	5	5
	Miami	6	5	9
	Lakeland	9	10	11
Washington	Vancouver	12	7	5
	Seattle	8	5	1
	Redmond	5	9	5

Here three cubes containing sales quantity for different states are combined to form on single cube using UNION

CUBE RESULTING FROM UNION QUERY

3.2.5.2 JOIN

Customer

Order ID	Date	Customer	Order Total
A001	Jan, 31	Ben Hills	275
A002	Feb, 10	Tech Wells & Co.	267
A003	Feb, 25	Johnz Automotives	110

Customer.OrderID

=

Products

Order ID	Date	Product Category	Qty	Unit Price	Product Total
A001	Jan, 31	Cool Drinks	50	3	150
A002	Feb, 10	Cool Drinks	25	3	75
A001	Jan, 31	Fruit Juices	25	5	125
A003	Feb, 25	Fruit Juices	22	5	110
A002	Feb, 10	Health Drinks	32	6	192

Customer - Products

Order ID	Date	Customer	Product Category	Qty	Unit Price	Product Total	Order Total
A001	Jan, 31	Ben Hills	Cool Drinks	50	3	150	275
A001	Jan, 31	Ben Hills	Fruit Juices	25	5	125	275
A002	Feb, 10	Tech Wells & Co.	Cool Drinks	25	3	75	267
A002	Feb, 10	Tech Wells & Co.	Health Drinks	32	6	192	267
A003	Feb, 25	Johnz Automotives	Fruit Juices	22	5	110	110

Here two cubes containing customer details and product details are merged to form a single cube containing customer & sold product details by means of a common field Order ID using JOIN query

CUBE RESULTING FROM JOIN QUERY

3.2.6 Supported Data Types

A cube is formed from data sets that contain various data types.

Examples of data types and usage:

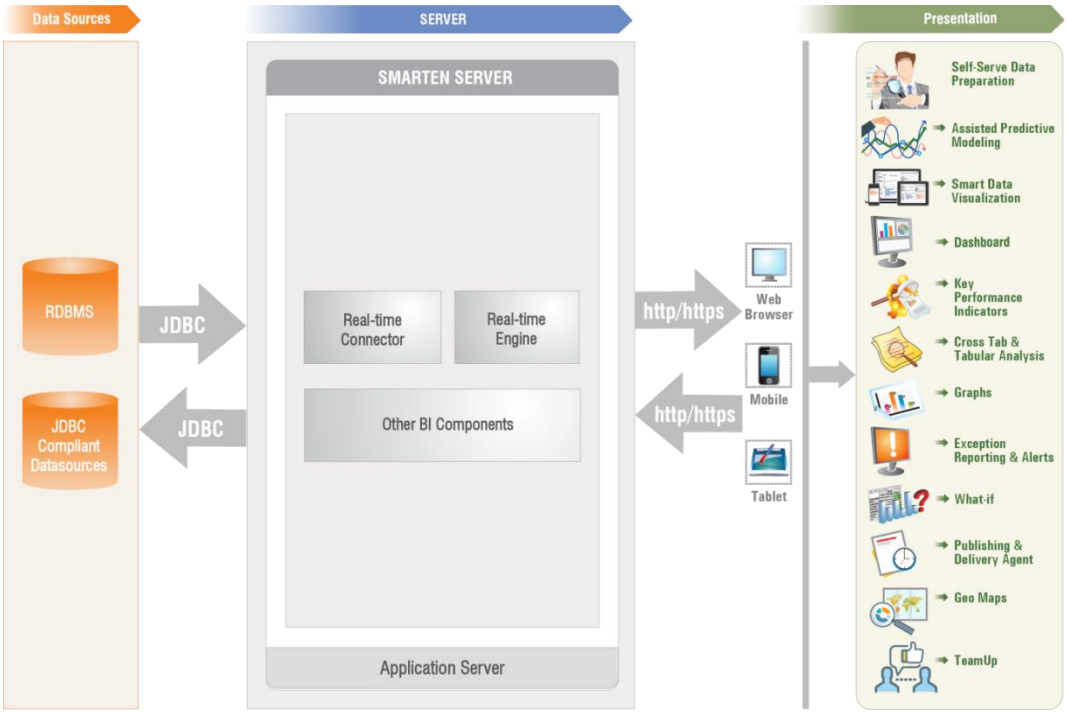
Data type	Description	Example
String	A sequence of characters, usually forming a part of text	Hello, World
Integer	A whole number that includes all negative numbers, zero, and all positive numbers	10
Double	Numbers with decimals	12.345
Date	Various date formats/expressions are possible for measuring date	15/07/2014 (dd/MM/yyyy) September 15 (MMMM dd) September 15, 2014 (MMMM dd, yyyy)
Time	Used for measuring time	07:45:40 HH:mm:ss or 07:45 HH:mm
Timestamp	Combination of date and time data types	September 15, 07:45:40 (MMMM dd, HH:mm:ss) 09-15-2014 07:45:40 (MM-dd-yyyyHH:mm:ss)
Boolean	Values with only zero and one	1 (if True) and 0 (if False)
Bit	Values generated in Bit format by any system	1 and/or 0

Note:
Some of the data types that require to store large data types, such as Blob-data type, will store cube columns with null values, i.e., these values will not be stored in cube data files.

3.3 Smarten Real-Time Cubes

Smarten offers real-time analytics through its real-time cube architecture. Real-time analytics is required in various use cases, such as the stock market, telecommunications, IT infrastructure management, and IoT, where recent data is important, and users need to access data in real time.

The Real-Time Data Connector does not store or cache any data in the cubes. It extracts the data from data sources as and when required and always retrieves the latest data from data sources. It connects to JDBC / ODBC-compliant relational databases, such as Microsoft® SQL Server, Oracle, and MySQL.



SMARTEN—REAL-TIME CUBE SYSTEM ARCHITECTURE

The Smarten Real-Time cube connector provides two ways to connect to a database:

1) through graphical UI wizard
2) through paste query option

The Smarten real-time cube connector through wizard allows the user to select databases, tables, and columns, and define relationships by a drag and drop interface. The Smarten real-time cube connector through paste query allows users to paste a generated query.

Users can access real-time cubes in Smarten by the following steps:

- Create Database Profile / Use existing database profile
- Define Real-Time Cube Profile
- Access Real-Time cubes from Smarten front-end tools

Create Data Source Profile:
Users can enter required configuration parameters, such as driver name, URL, user name, and password. It validates the connection and creates the database profile in the system.

Define Real-Time Cube:
To define the real-time cube profile, users can select the database profile created in the above step. After selecting the database profile, the system will connect to the data source and provide easy-to-use steps to define the real-time cube metadata profile within Smarten.

- Configuration of database profile using JDBC/ODBC driver
- Connection with database
- Design SQL statement in two ways:
    - Graphical Query Designer
    - SQL Query in editor
- Metadata definition for cube by designing and mapping data source columns and cube dimension and measure columns
- Define Dimension Map Hierarchy

Access Real-Time Cubes from Smarten front-end tools:
Users can access real-time cubes from front-end objects, such as dashboards, crosstab, tabular, graphs, GeoMap and KPI. SQL queries are formed dynamically based on user actions from BI front-end tools, e.g., outliner settings or add column from analysis. Dynamically generated SQL queries are sent for execution to SQL executor, and database engine returns query results, which are then processed and displayed to the user.

Please note here that cube and analytic functions available to users depend on the type of cube used in a particular front-end object. Refer to the user manual for a list of functions available while using real-time cubes.

3.4 Cube & Object Management

3.4.1 Matching Cube Criteria

Users, especially technical users, may need to associate objects (e.g., crosstab or tabular or a graph) created from one cube to another cube. It is possible to associate an object with another cube if the columns of original cube from which the object was created is matching the columns of new cube.

Criteria for identical match for columns (dimensions, measures, custom cube dimensions, custom cube measures, and dimension hierarchies) are described below.

- The Datatype of Dimensions in both cubes must be same.
  For example, if anlaysis1 is using dimension1 from Cube1, and dimension1 is of string data type, it can be matched with any dimension having string data type from target cube.
- Target cube should have at least one unique matching dimension for each dimension used in the object.
  For example, if graph1 is using dimension1 (date type) and dimension2 (string type) from Cube1, matching target cube must have at least one date data type dimension and one string data type dimension.
- One to one relationship between dimensions from target cube and dimensions used in objects.
  For example, if dimension1 of an object, is matched with dimensionx of target cube, dimension cannot be matched with any other dimension of the object.
- Any measure of an object will match with any measure from the target cube, as datatypes of measures are always the same.
- One measure in target cube can be associated with multiple measures from the object.
  For example, measurex of target cube can be associated with measure1 and measure2 of analysis1.

3.4.2 Assigning Objects to another matching cube

Users can assign any or all objects (crosstab, tabular, graphs, GeoMap and KPI) of one cube (including the objects with deleted cube) to matching columns of another cube.

> Note:
> You can associate object created from one type of cube with any other type of cube. For example, you can associate a crosstab created from cache cube with real-time or MDX cube.

3.4.3 Copy cube

This feature enables users to copy a cube with its metadata and tool templates.

This feature will copy a cube with its metadata and tool templates. This will improve the process of replicating cubes and the reusability for a template-driven deployment process.
For example, if you have created a sales cube and want to replicate this cube for a different zone wise groups of users, you can create zone wise copies of this sales cube (e.g., sales cube zone1, sales cube zone2, etc.) and provide access rights to these cubes to different groups of users (e.g., zone1 users, zone2 users, etc.). You do not need to go through a cube creation process for each zone.

3.4.4 Renaming the Cubes

This feature enables users to rename the cube.

Please consider a scenario. For example, the IT team is designing, developing, and testing various cubes and analysis objects on the development server. They created a cube and named it "**Sales-Development-Server-Cube**." Various crosstabs, graphs, GeoMaps, dashboards, and tabular are generated.

Once testing and verification are done, the **"Sales-Development-Server-Cube"** and the associated objects of this cube are moved or copied to the production server, and the cube is renamed "**Sales-Cube"** from "**Sales-Development-Server Cube**."

All analysis objects on the production server are now associated to "**Sales-Cube"** on the production server rather than "**Sales-Development-Server-Cube**." So renaming avoids redeveloping or redesigning any cubes or objects.

Once the cube is renamed, the modified cube name would automatically be reflected in the associated objects.

3.4.5 Renaming the Objects

This feature allows users to modify an object (crosstabs, KPI, tabular, graphs, GeoMap and dashboard) name.

You can rename the objects even if the cube associated with these objects is deleted.
Consider a scenario. The crosstab generated from the **Sales-Development-Server-Cube** is named **"Development Server-Sales Analysis"** during the development and testing phase. Once the cube and this crosstab move to the production server, the crosstab is renamed **"Sales Analysis"** without affecting its association with **Sales-Development-Server-Cube**.

3.4.6 Deleting the Cube without deleting dependent Objects

If a cube is deleted, user may or may not delete objects associated with that cube.

Users can reuse the dependent objects (without cube) by associating these objects to any other matching columns of another cube.

Smarten saves the profile of the deleted cube. Cube metadata, such as dimensions, measures and other parametric information, remains available in the system for reference.

Once a cube profile is permanently deleted, the Cube Profile and the metadata will no longer be available in the system.

3.5 Supported Features for Different Cubes

Following table specifies the feature availability for different cube types.

3.5.1 Cube Management Functions

Features		Smarten Cache Cubes	Smarten Real-Time Cubes	MDX SSAS Cubes
Profile Creation		✔	✔	✔
Cube Creation		✔	✔	✔
Storing transactional and aggregate data on Smarten		✔	✘	✘
Managed Memory Computing		✔	✘	✘
Cube Rebuild (Meta data update)		✔	✔	✘
Cube Rebuild (Data refresh)		✔	NA	NA
Linked Cube		✔	✘	✘
Dimension Map	Retrieval of dimension maps created on cube server	NA	NA	✔
	User defined dimension maps in Smarten	✔	✔	✔
Retrieval Parameters		✔	✔	✔
Use of Global variable in database query for rebuilding cubes		✔	✔	✘
Use of Predefined system level global variable '$currentuser$' in database query for cubes		✘	✔	✘
Custom Cube Dimension/Measure		✔		✘
Column Access Permission		✔	✔	✔
Data Access Permission		✔	✔	✔
Data Display Value Mapping		✔	✔	✔

3.5.2 Analytic Functions

Features				Smarten Cache Cubes	Smarten Real-Time Cubes	MDX SSAS Cubes
Slice and Dice				✔	✔	✔
Drill Down & Drill Up				✔	✔	✔
Drill Through				✔	✔	✔
Retrival Parameters				✔	✔	✔
Global Variables				✔	✔	✔
Time Series	Absolute			✔	✔	✔
	Relative	Full Period		✔	✔	✔
		Period-To-Date		✔	✔	✔
	Range			✔	✔	✔
Outliner Filter / Page Dimension Filter		On String Column	Particular Value	✔	✔	✔
			Value Starts with/Ends with/Contains/Null/Not Null	✔	✔	✘
			Value within range	✔	✔	✔
			Multiple Values	✔	✔	✔
		On Numeric Column	Particular Value	✔	✔	✔
			Value Greater than/Less than/Greater than equal to/Less than equal to/Null/Not Null	✔	✔	✔
			Value within range	✔	✔	✔
			Multiple Values	✔	✔	✔
		on Date Column	Particular Value	✔	✔	✔
			Value Before/After/Between/Not Between	✔	✔	✔
Cell Filter				✔	✔	✔
Advanced Filter				✔	✔	✔
Show / Hide				✔	✔	✔
Analysis Title				✔	✔	✔
Edit Label Text in Row, Column and Data Headers				✔	✔	✔
Supress Zeros in Row / Column				✔	✔	✔
Sort			General Sort	✔	✔	✔
			Custom Sort	✔	✔	✔
			Advanced Sort	✔	✔	✘
Rank				✔	✔	✔
Group / UnGroup				✔	✔	✔
SpotLighter				✔	✔	✔

Data Value / Display Value Mapping		✔	✔	✔
Data Operation	None	✔	✔	✔
	Sum Average Effective Average Count Effective Count Maximum Minimum First Last Distinct Count Distinct Sum Distinct Average Least Recent Values Most Recent Values	✔	✔	✘
	Row Percentage Row Group Percentage Column Percentage Column Group Percentage Total Percentage Relative Row Difference Relative Row Difference Percentage Relative Row Group Difference Relative Row Group Difference Percentage Relative Column Difference Relative Column Difference Percentage Relative Column Group Difference Relative Column Group Difference Percentage Row Cumulative Sum Column Cumulative Sum Row Group Cumulative Sum Column Group Cumulative Sum	✔	✔	✔
Summary Operation	Default	✔	✔	✔

	Sum Average Effective Average Count Effective Count Maximum Minimum First Last	✔	✔	✘
	Group Sum Group Average Group Count Group Maximum Group Minimum Row Percentage Row Group Percentage Column Percentage Column Group Percentage Total Percentage Relative Row Difference Relative Row Difference Percentage Relative Row Grop Difference Relative Row Group Difference Percentage Relative Column Difference Relative Column Difference Percentage Relative Column Group Difference Relative Column Group Difference Percentage Row Cumulative Sum Column Cumulative Sum Row Group Cumulative Sum Column Group Cumulative Sum	✔	✔	✔
Notes		✔	✔	✔
Format Component Properties		✔	✔	✔
Add / Remove Columns		✔	✔	✔
Add Custom Measure (UDDC)		✔	✔	✔

Add Custom Dimension Value (UDHC)	✔	✔	✔
What-If Analysis	✔	✔	✔
SubView	✔	✔	✔
Master-Detail view in Tabular report	✔	✔	✔
Auto Generate Graph from Analysis	✔	✔	✔
Export Analysis	✔	✔	✔
Save Analysis	✔	✔	✔
Refresh Analysis	✔	✔	✔
Delivery & Publishing Agent – [Publish Now] and [Publish Settings]	✔	✔	✔
Operations Summary	✔	✔	✔
Printing Analysis	✔	✔	✔
Page Preview	✔	✔	✔

Note:
Cube type should be selected based on the use case.

Reference: **Impact-of-Cube-Design-on-Performance > Cube type selection recommendations**

4 Analytic Functions

Various analytic functions are available to help users effectively analyse data within various Smarten modules. All functions may not be available in all modules, e.g., summary operations are not available in graphs and GeoMap.

4.1 Slice & Dice

"Slice & Dice" describes the functions at the core of OLAP analysis. The multidimensional tools allow users to view data from any angle. Through slice & dice, user can rotate the presentation between rows and columns in crosstabs. After generating a crosstab, graph, or tabular, a user swaps dimensions from row to column and column to row.

ProductCategory	2011	2012	2013	2014
	GrossSales	GrossSales	GrossSales	GrossSales
Alcoholic Drinks	14706701	17570431	19735160	19164319
Bakery	19456392	19075592	11794632	18094427
Confectionary	1140009	1284075	1751065	1608605
Cool Drinks	610659	827055	1054727	1089105
Fruit Juices	7213353	8155245	8332277	6978641
Health Drinks	3236351	4460681	4893260	5569268
Ice Cream	4117144	5936579	6187398	5851498
Snacks	1142557	1300112	1893006	1466646
Tea	1136134	979937	916533	857114

Year	Alcoholic Drinks	Bakery	Confectionary	Cool Drinks	Fruit Juices	Health Drinks	Ice Cream	Snacks	Tea
	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
2011	14706701	19456392	1140009	610659	7213353	3236351	4117144	1142557	1136134
2012	17570431	19075592	1284075	827055	8155245	4460681	5936579	1300112	979937
2013	19735160	11794632	1751065	1054727	8332277	4893260	6187398	1893006	916533
2014	19164319	18094427	1608605	1089105	6978641	5569268	5851498	1466646	857114

SLICE AND DICE PRODUCT CATEGORY AND REGIONWISE SALES

4.2 Drill down and Drill up

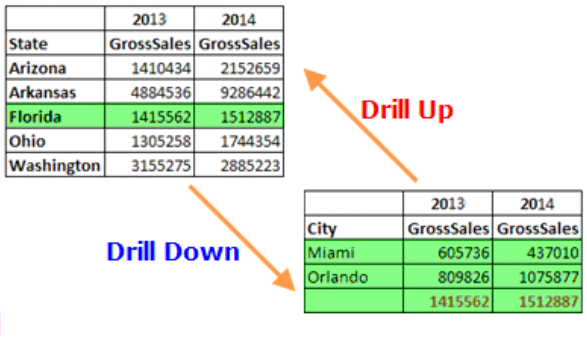
"Drill down" and "Drill up" provide interactive data analysis through predefined dimension hierarchy. In hierarchical drilling, user can interactively retrieve data at multiple levels. User can move down and up the hierarchies to see how the information at various levels is related.

4.2.1 Drill down

"Drill down" interactive data analysis allows users to navigate from less-detailed aggregated information to view more granular data. After looking at the gross sales for a state, user may wish to see the individual sales for each city of that state.

4.2.2 Drill up

It refers to the process of navigating information from the detailed (down) to the summarized (up) along a dimension hierarchy. For example, when viewing the data for the city of Miami, a drill-up operation in the location dimension would display Florida. A further drill up on Florida would display data for the USA.



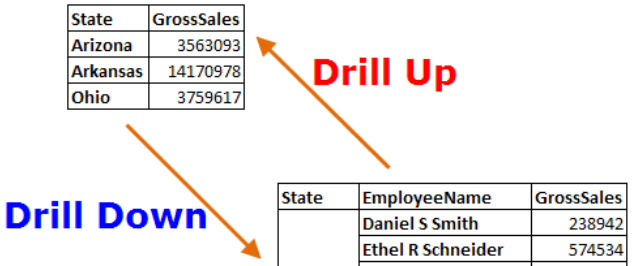
	2013	2014
State	GrossSales	GrossSales
Arizona	1410434	2152659
Arkansas	4884536	9286442
Florida	1415562	1512887
Ohio	1305258	1744354
Washington	3155275	2885223

	2013	2014
City	GrossSales	GrossSales
Miami	605736	437010
Orlando	809826	1075877
	1415562	1512887

DRILL DOWN AND DRILL UP DATA ANALYSIS

Drill down / Drill up can be based not only on predefined dimensional hierarchy, but the user can also add unrelated child levels to a parent node to see the bifurcation of the aggregated information regardless of predefined hierarchy defined at cube levels.

For example, we can see sales by employees for any given state even if state/employee hierarchy is not defined in the dimension map in the cube. Add employees to the drill-down level of states as shown in the figure below.



State	GrossSales
Arizona	3563093
Arkansas	14170978
Ohio	3759617

State	EmployeeName	GrossSales
Arizona	Daniel S Smith	238942
	Ethel R Schneider	574534
	John R Parker	1102554
	Maria L Perez	825329
	Sybil P Johnson	504247
	Vuong Paul	317487
		3563093
Arkansas	Heather Bruce	1236897
	Jenifer Jefferson	3373130
	Jennifer Paul	4136837
	Nicholls Adamson	2474969
	Paul Philip	1505416
	Vuong Smith	1443729
		14170978
Ohio	Bruce Adamson	668134
	Daichi Makiya	489929
	Dolores M Quintana	48890
	Heather A Nicholls	718197
	Jason V Mehta	570239
	Maude F Setright	501521
	Philip X Smith	518994
	Salvatore Jefferson	243713
		3759617

DRILL DOWN AND DRILL UP OF EMPLOYEE SALES BY STATE

4.3 Drill Through

Using “drill through” on analysis retrieves the detailed row or transaction level data from which the data in the cube cell was summarized. It is used to access the underlying transactional or row-level view of selected analysis columns / row or cell.

For example, user can see all transactions contributing to **GrossSales** of **1643997** for **Alcoholic Drinks** in **January**.

ProductCategory	January	February	March	April	May	June
	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Alcoholic Drinks	1643997	2004126	1315188	1236196	1380433	1381668
Bakery	1121904	1675234	1380840	1478191	942214	959798
Confectionary	111088	71674	130537	230122	198609	138397
Cool Drinks	69884	115922	101143	163810	192854	83526
Fruit Juices	373858	520300	807057	421825	698436	400619

Drill Through

Date	ProductCategory	ProductName	GrossSales
1/20/2014	Alcoholic Drinks	Wine	59917
1/15/2014	Alcoholic Drinks	Wine	44098
1/15/2014	Alcoholic Drinks	Beer	235136
1/20/2014	Alcoholic Drinks	Whisky	204865
1/20/2014	Alcoholic Drinks	Wine	40605
1/1/2014	Alcoholic Drinks	Whisky	110128
1/31/2014	Alcoholic Drinks	Wine	38978
1/31/2014	Alcoholic Drinks	Wine	6637
1/20/2014	Alcoholic Drinks	Beer	169068
1/31/2014	Alcoholic Drinks	Beer	113369
1/15/2014	Alcoholic Drinks	Whisky	191841
1/15/2014	Alcoholic Drinks	Beer	192324
1/20/2014	Alcoholic Drinks	Beer	195309
1/1/2014	Alcoholic Drinks	Whisky	20323
1/31/2014	Alcoholic Drinks	Whisky	21399

Value of Alcoholic Drinks for January → 1643997

DRILL THROUGH

If cube created with only “Store drill through data” option then drill though data retrieves from Flat cube data.

If cube created with only “Perform aggregation” option then drill through data retrieves from Aggregated data of cube.

If cube created with both “Store drill through data” and “Perform aggregation” options then based on the different scenarios drill through data retrieves from the “Flat data” or “Aggregated data” of the cube.

If an object being used for drill through is using any custom cube column (custom cube dimension or custom cube measure), then drill through data will be displayed from the aggregated data of the cube. If an object being used does not use any custom cube column, then drill through data will be displayed from flat cube data.

State	City	Product Category	Product	Quantity	Production Cost	Packing Cost
Arizona	Phoenix	Bakery	Bread	150	300	30
Arizona	Phoenix	Bakery	Bread	200	400	40

Arizona	Phoenix	Bakery	Bun	160	480	48
Arizona	Phoenix	Bakery	Bun	200	600	60
Arizona	Phoenix	Cool Drinks	Soda	200	1000	60
Arizona	Phoenix	Cool Drinks	Soda	180	720	54
Arizona	Scottsdale	Bakery	Bread	400	1200	80
Arizona	Scottsdale	Bakery	Cookies	300	900	60
Arizona	Scottsdale	Bakery	Bun	250	750	75
Arizona	Scottsdale	Bakery	Bun	200	600	60
Arizona	Scottsdale	Cool Drinks	Cola	180	900	54
Arizona	Scottsdale	Cool Drinks	Cola	190	760	57
Florida	Miami	Bakery	Bread	200	400	40
Florida	Miami	Bakery	Bread	250	500	50
Florida	Miami	Bakery	Bun	150	450	45
Florida	Miami	Bakery	Bun	200	600	60
Florida	Miami	Cool Drinks	Cola	170	850	51
Florida	Miami	Cool Drinks	Soda	150	600	45
Florida	Orlando	Bakery	Bread	270	540	54
Florida	Orlando	Bakery	Bun	180	540	54
Florida	Orlando	Cool Drinks	Cola	190	950	57
Florida	Orlando	Cool Drinks	Cola	200	1000	60
Florida	Orlando	Cool Drinks	Soda	170	680	51
Florida	Orlando	Cool Drinks	Soda	210	840	63

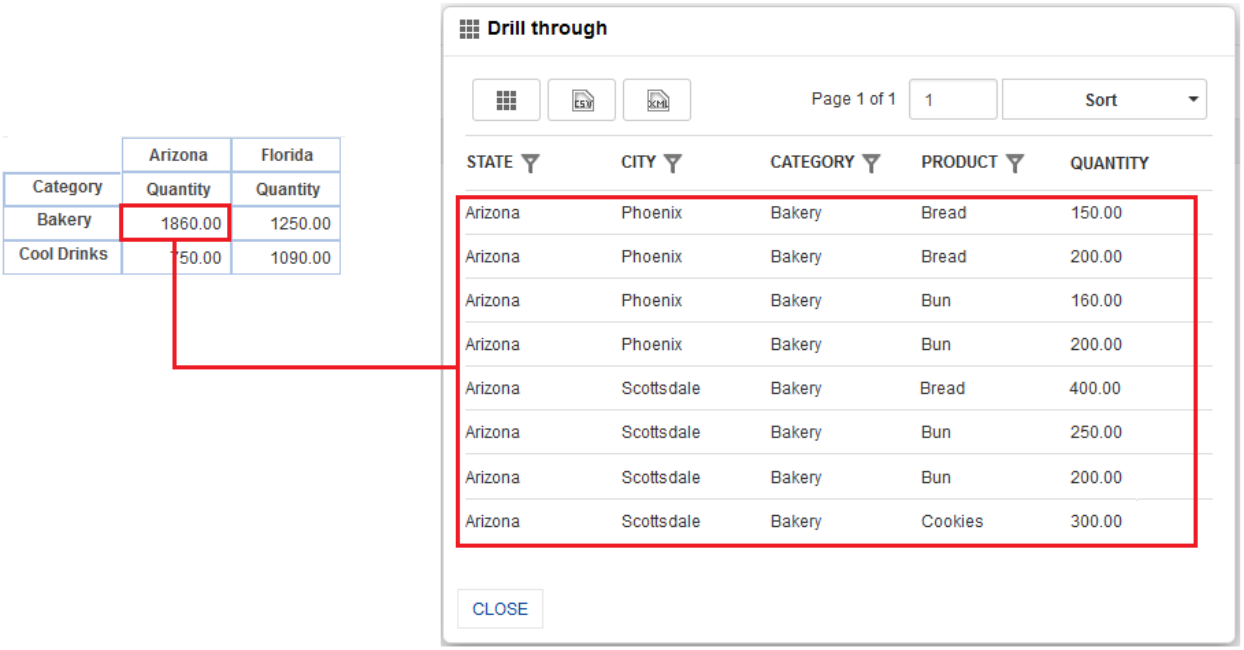
FLAT DATA SET

							Custom cube column
State	City	Product Category	Product	Qty	Production Cost	Packing Cost	Total Cost
Arizona	Phoenix	Bakery	Bread	350	700	70	770
Arizona	Phoenix	Bakery	Bun	360	1080	108	1188
Arizona	Phoenix	Cool Drinks	Soda	380	1720	114	1834
Arizona	Scottsdale	Bakery	Bread	400	1200	80	1280
Arizona	Scottsdale	Bakery	Cookies	300	900	60	960
Arizona	Scottsdale	Bakery	Bun	450	1350	135	1485
Arizona	Scottsdale	Cool Drinks	Cola	370	1660	111	1771
Florida	Miami	Bakery	Bread	450	900	90	990
Florida	Miami	Bakery	Bun	350	1050	105	1155
Florida	Miami	Cool Drinks	Cola	170	850	51	901
Florida	Miami	Cool Drinks	Soda	150	600	45	645
Florida	Orlando	Bakery	Bread	270	540	54	594
Florida	Orlando	Bakery	Bun	180	540	54	594
Florida	Orlando	Cool Drinks	Cola	390	1950	117	2067
Florida	Orlando	Cool Drinks	Soda	380	1520	114	1634

AGGREGATED DATA SET WITH CUSTOM CUBE COLUMN "TOTAL COST"

Custom cube column Total Cost = Production Cost + Packing Cost

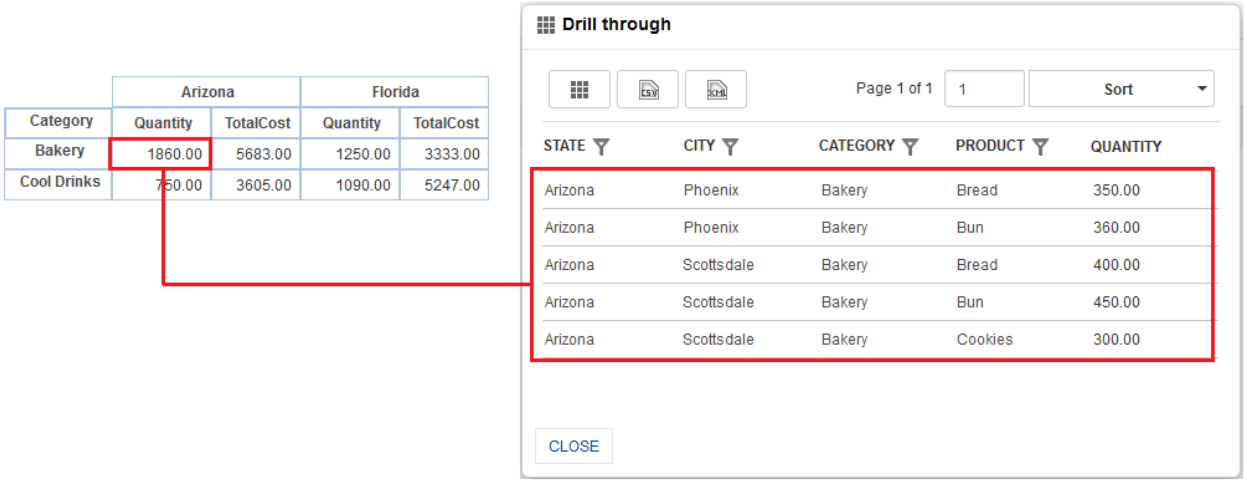
Scenario 1: Crosstab does not use any custom cube column, and no custom cube column is selected in drill through.



FLAT DATA IN DRILL THROUGH

In such a scenario, flat cube data will be displayed in drill through view.

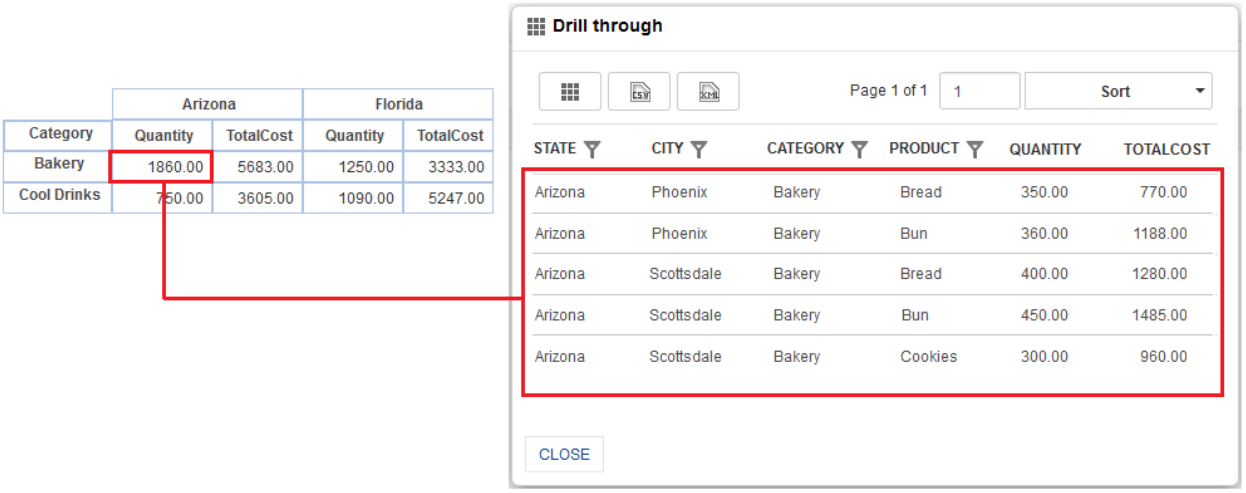
Scenario 2: Crosstab uses any custom cube column, and no custom cube column is selected in drill through.



AGGREGATED DATA IN DRILL THROUGH

In such a scenario, aggregated cube data will be displayed in drill through view.

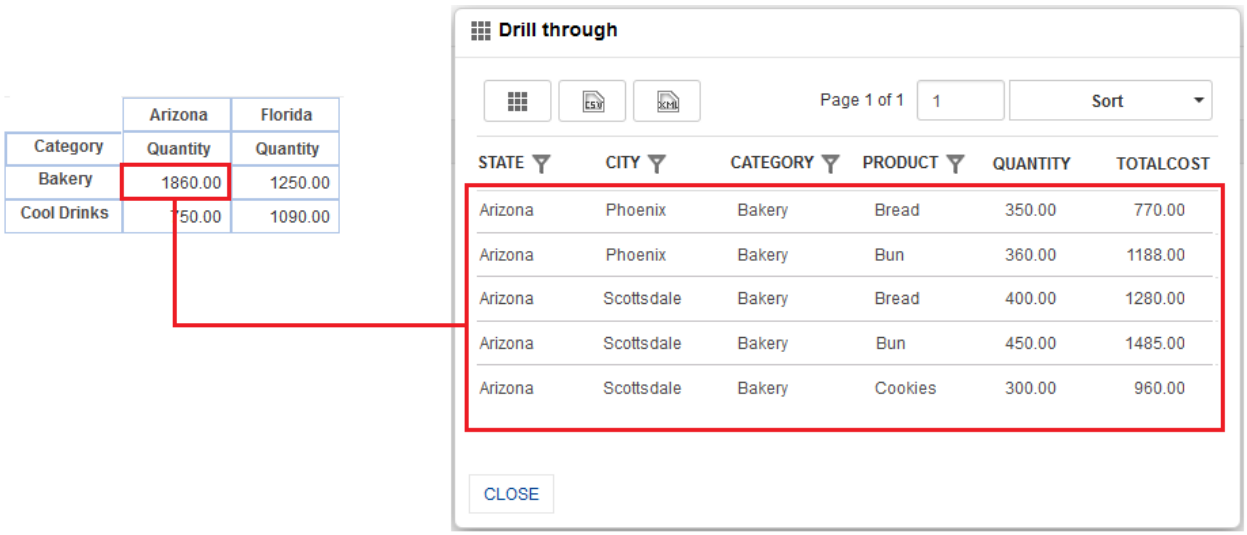
Scenario 3: Crosstab uses any custom cube column, and custom cube column is selected in drill through.



AGGREGATED DATA IN DRILL THROUGH

In such a scenario, aggregated cube data will be displayed in drill through view.

Scenario 4: Crosstab does not use any custom cube column, but custom cube column is selected in drill through.



AGGREGATED DATA IN DRILL THROUGH

In such a scenario, aggregated cube data will be displayed in drill through view.

4.4 Global Variables

The global variables are defined at the cube level. They can be accessed globally with various expressions and filters for BI objects within Smarten.

For example, users need to view the projection of growth based on variable % values of sales amount. For this, a Custom Measure Column (UDDC) **Growth** can be created that would be calculated on the basis of a variable **X** and **GrossSales**. This X can be created as a **Global Variable** and assigned different values at different times to evaluate various scenarios.

Formula for **Growth**: **GrossSales** + (**X***GrossSales**)/100

Users can change the value of X to see different projections of growth.
Any change in **X** would be reflected in all analyses where the value of **X** is used through different expressions in filters, the Custom Dimension Value **(UDHC)**, the Custom Measure Column (UDDC),

and retrieval parameters. Hence, it saves users from the tedious task of modifying various expressions and filter formula manually and provides simple "what if" analysis scenarios.

Once the global variable is defined, it would be accessible throughout the application while applying **Filters**, creating Custom Dimension Value (**UDHC)**, Custom Measure Column (**UDDC),** and **Retrieval Parameters**. Users can also use these global variables in cube query while rebuilding Cache or Real-Time cubes.

User can also use the predefined system level global variable '$currentuser$' in Real-Time cube query.

For example, user can create a real time cube using query "Select * from Sales where employeename = '$currentuser$'". In this scenario, if user1 is logged in and is using real time cube data, query expression will be: "Select * from Sales where employeename = 'User1', and if user2 is logged in and is using this real time cube data, query expression will be: "Select * from Sales where employeename = 'User2'.

> Note:
> Global variables are available within all BI objects (such as crosstab, graph, GeoMap, dashboards, and tabular) created from a cube. Global variables created for one cube cannot be accessed from within objects created from another cube.

ProductCategory	Arizona	Arkansas	Florida	Ohio	Washington
	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Alcoholic Drinks	6415757	18261094	20796408	8006101	17697251
Bakery	10197878	16804926	21361087	8731632	11325520
Confectionary	363390	1883308	1639046	647845	1250165
Cool Drinks	690706	1041085	961880	322948	564927
Fruit Juices	4249178	8438495	8224081	2722542	7045222
Health Drinks	1456274	4588049	5792470	2061809	4260958
Ice Cream	3938710	5874464	5560304	2677707	4041434
Snacks	577363	1423668	1528300	492191	1780800
Tea	1001994	797487	938838	406455	744943

Global Variable Var1 = 15

Growth = GrossSales + (GrossSales * Var1/100)

ProductCategory	Arizona		Arkansas		Florida		Ohio		Washington	
	GrossSales	Growth	GrossSales	Growth	GrossSales	Growth	GrossSales	Growth	GrossSales	Growth
Alcoholic Drinks	6415757	7378120	18261094	21000258	20796408	23915869	8006101	9207016	17697251	20351839
Bakery	10197878	11727560	16804926	19325665	21361087	24565250	8731632	10041377	11325520	13024348
Confectionary	363390	417898	1883308	2165804	1639046	1884903	647845	745021	1250165	1437690
Cool Drinks	690706	794311	1041085	1197248	961880	1106162	322948	371390	564927	649666
Fruit Juices	4249178	4886554	8438495	9704269	8224081	9457694	2722542	3130923	7045222	8102005
Health Drinks	1456274	1674715	4588049	5276256	5792470	6661341	2061809	2371080	4260958	4900101
Ice Cream	3938710	4529516	5874464	6755633	5560304	6394350	2677707	3079363	4041434	4647649
Snacks	577363	663968	1423668	1637218	1528300	1757545	492191	566020	1780800	2047921
Tea	1001994	1152294	797487	917110	938838	1079664	406455	467423	744943	856685

CUSTOM MEASURE COLUMN (GROWTH) DERIVED USING GLOBAL VARIABLE X (VALUE: 15)

ProductCategory	Arizona	Arkansas	Florida	Ohio	Washington
	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Alcoholic Drinks	6415757	18261094	20796408	8006101	17697251
Bakery	10197878	16804926	21361087	8731632	11325520
Confectionary	363390	1883308	1639046	647845	1250165
Cool Drinks	690706	1041085	961880	322948	564927
Fruit Juices	4249178	8438495	8224081	2722542	7045222
Health Drinks	1456274	4588049	5792470	2061809	4260958
Ice Cream	3938710	5874464	5560304	2677707	4041434
Snacks	577363	1423668	1528300	492191	1780800
Tea	1001994	797487	938838	406455	744943

Global Variable Var1 = 20

Growth = GrossSales + (GrossSales * Var1/100)

ProductCategory	Arizona		Arkansas		Florida		Ohio		Washington	
	GrossSales	Growth	GrossSales	Growth	GrossSales	Growth	GrossSales	Growth	GrossSales	Growth
Alcoholic Drinks	6415757	7698908	18261094	21913312	20796408	24955690	8006101	9607321	17697251	21236702
Bakery	10197878	12237454	16804926	20165911	21361087	25633304	8731632	10477958	11325520	13590624
Confectionary	363390	436068	1883308	2259970	1639046	1966855	647845	777414	1250165	1500198
Cool Drinks	690706	828847	1041085	1249302	961880	1154256	322948	387538	564927	677912
Fruit Juices	4249178	5099013	8438495	10126194	8224081	9868898	2722542	3267050	7045222	8454266
Health Drinks	1456274	1747529	4588049	5505659	5792470	6950965	2061809	2474171	4260958	5113149
Ice Cream	3938710	4726451	5874464	7049356	5560304	6672365	2677707	3213249	4041434	4849720
Snacks	577363	692836	1423668	1708401	1528300	1833960	492191	590629	1780800	2136961
Tea	1001994	1202393	797487	956984	938838	1126606	406455	487746	744943	893932

CUSTOM MEASURE COLUMN (GROWTH) DERIVED FROM MODIFIED VALUE OF GLOBAL VARIABLE X (VALUE: 20)

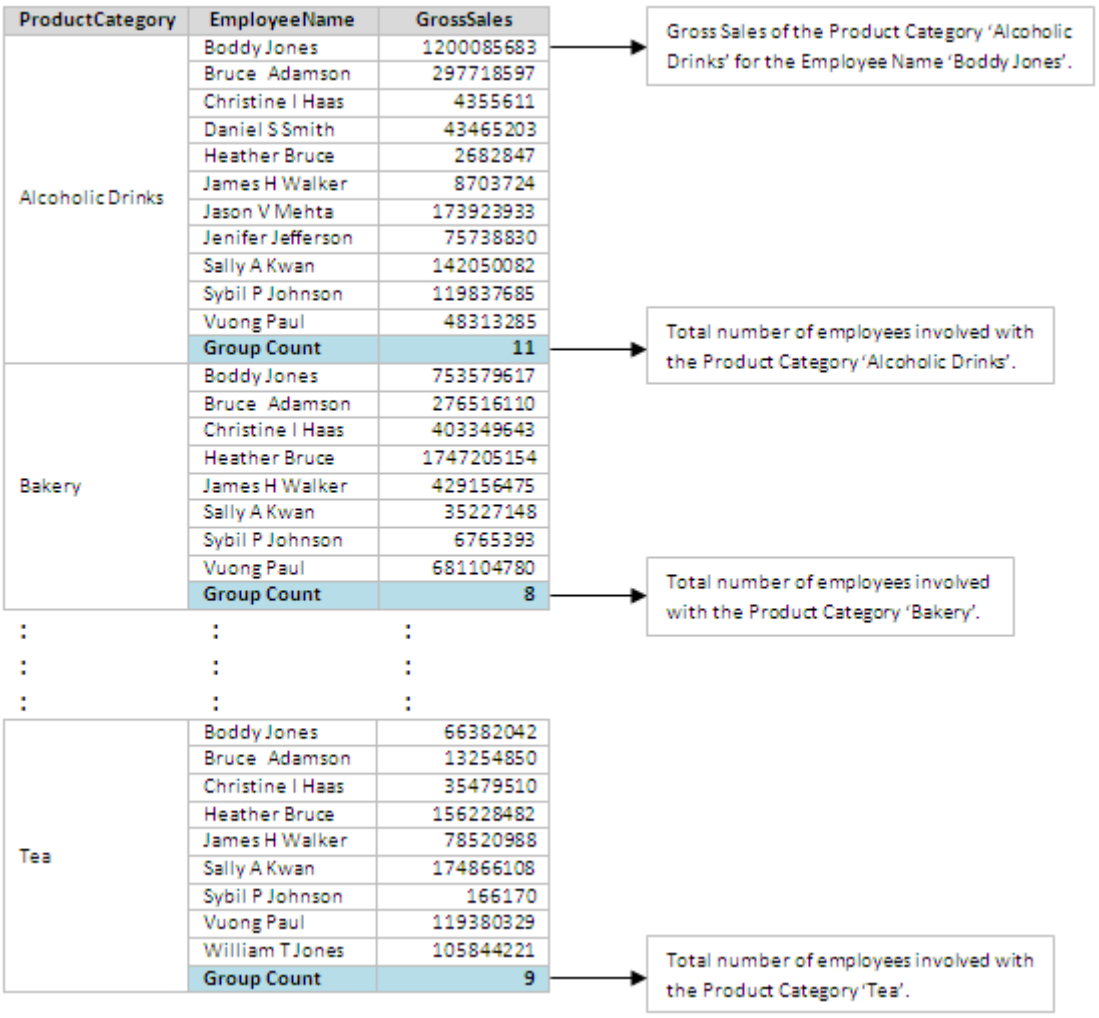
The value of global variable **X** is modified from **15** to **20**. In the column **Growth**, new value **20** will be taken into consideration, and column values will change accordingly.

> Note:
> The global variables will be available for such objects as crosstabs, graphs, GeoMap, tabular, and KPIs.

4.5 Show only Summary data

This feature is useful in scenarios when a user wants to see only summary row(s) for a group of rows / columns without displaying the actual rows / columns in crosstab or tabular. For example, if the user wants to view the total number of customers for each product category without displaying the customer details, they can use the Group Count Function to display that and select "Show Only Summary Data" option.

This feature is rather helpful to users when they need to display the group summary operations, such as group count, group average, group maximum, group minimum, etc., in this fashion.



ProductCategory	EmployeeName	GrossSales
Alcoholic Drinks	Boddy Jones	1200085683
	Bruce Adamson	297718597
	Christine I Haas	4355611
	Daniel S Smith	43465203
	Heather Bruce	2682847
	James H Walker	8703724
	Jason V Mehta	173923933
	Jenifer Jefferson	75738830
	Sally A Kwan	142050082
	Sybil P Johnson	119837685
	Vuong Paul	48313285
	Group Count	**11**
Bakery	Boddy Jones	753579617
	Bruce Adamson	276516110
	Christine I Haas	403349643
	Heather Bruce	1747205154
	James H Walker	429156475
	Sally A Kwan	35227148
	Sybil P Johnson	6765393
	Vuong Paul	681104780
	Group Count	**8**
:	:	:
:	:	:
:	:	:
Tea	Boddy Jones	66382042
	Bruce Adamson	13254850
	Christine I Haas	35479510
	Heather Bruce	156228482
	James H Walker	78520988
	Sally A Kwan	174866108
	Sybil P Johnson	166170
	Vuong Paul	119380329
	William T Jones	105844221
	Group Count	**9**

BEFORE: PRODUCT CATEGORYWISE & EMPLOYEEWISE SALES WITH PRODUCT CATEGORYWISE NUMBER OF EMPLOYEES

ProductCategory	EmployeeName	GrossSales
Alcoholic Drinks	Group Count	11
Bakery	Group Count	8
Confectionary	Group Count	9
Cool Drinks	Group Count	9
Fruit Juices	Group Count	9
Health Drinks	Group Count	8
Ice Cream	Group Count	9
Snacks	Group Count	8
Tea	Group Count	9

Total numbers of employees involved with all Product Categories.

AFTER: PRODUCT CATEGORYWISE NUMBER OF EMPLOYEES WITHOUT DISPLAYING CATEGORYWISE SALES

4.6 Sort

Data can be sorted in ascending, descending, and custom (user defined) orders, using particular Dimension or Measure fields.

4.6.1 Simple Sort

Simple sorting in ascending or descending order.

ProductCategory	Sales Quantity
Alcoholic Drinks	200
Tea	300
Ice Cream	600
Snacks	500

Data sorted in ascending order of ProductCategory

ProductCategory	Sales Quantity
Alcoholic Drinks	200
Ice Cream	600
Snacks	500
Tea	300

Data sorted in descending order of Sales Quantity

ProductCategory	Sales Quantity
Ice Cream	600
Snacks	500
Tea	300
Alcoholic Drinks	200

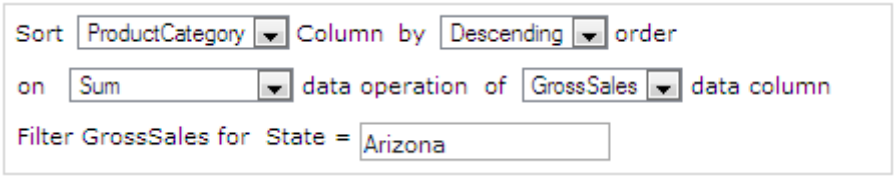
SORT BY PRODUCT CATEGORY AND SALES QUANTITY

User can also use “Advanced Sort” to sort dimension based on a data operation on a particular measure.

4.6.2 Advance Sort

Applying filter conditions for sorting of the data—Advance Sorting

User can also apply sorting of data by using various data operations on particular measure. For example, user can sort ProductCategory column in "descending" order on the Sum of GrossSales for the state of Arizona.



Advance filtering can be applied on data column using data operations, such as **Sum, Average, Effective Average, Count, Effective Count, Ineffective Count, Minimum, and Maximum**.

	Arizona	Arkansas	Florida	Ohio	Washington
ProductCategory	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Alcoholic Drinks	6415757	18261094	20796408	8006101	17697251
Bakery	10197878	16804926	21361087	8731632	11325520
Confectionary	363390	1883308	1639046	647845	1250165
Cool Drinks	690706	1041085	961880	322948	564927
Fruit Juices	4249178	8438495	8224081	2722542	7045222
Health Drinks	1456274	4588049	5792470	2061809	4260958
Ice Cream	3938710	5874464	5560304	2677707	4041434
Snacks	577363	1423668	1528300	492191	1780800
Tea	1001994	797487	938838	406455	744943

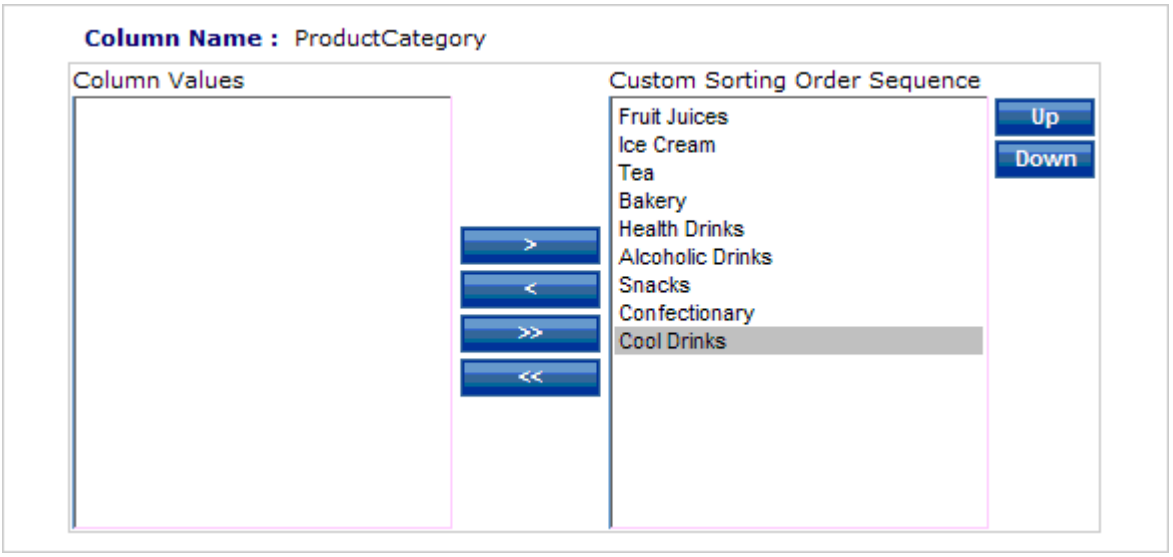
DATA SORTED ON THE PRODUCT CATEGORY DIMENSION VALUES

	Arizona	Arkansas	Florida	Ohio	Washington
ProductCategory	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	10197878	16804926	21361087	8731632	11325520
Alcoholic Drinks	6415757	18261094	20796408	8006101	17697251
Fruit Juices	4249178	8438495	8224081	2722542	7045222
Ice Cream	3938710	5874464	5560304	2677707	4041434
Health Drinks	1456274	4588049	5792470	2061809	4260958
Tea	1001994	797487	938838	406455	744943
Cool Drinks	690706	1041085	961880	322948	564927
Snacks	577363	1423668	1528300	492191	1780800
Confectionary	363390	1883308	1639046	647845	1250165

ANALYSIS AFTER APPLYING THE ADVANCE SORTING ON **THE "PRODUCTCATEGORY"** COLUMN IN **"DESCENDING"** ORDER ON **"SUM"** DATA OPERATION OF **"GROSSSALES"** DATA COLUMN FOR THE STATE **"ARIZONA."**

4.6.3 Custom Sort

Users can also sort data in custom order based on specific requirements.

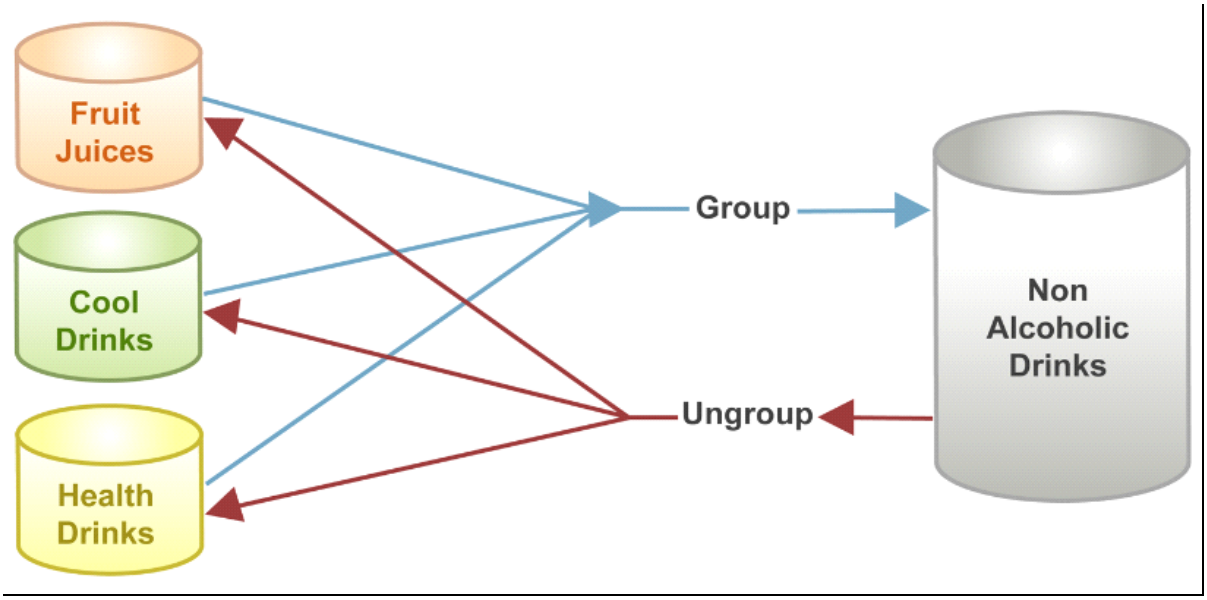


	Arizona	Arkansas	Florida	Ohio	Washington
ProductCategory	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Fruit Juices	4249178	8438495	8224081	2722542	7045222
Ice Cream	3938710	5874464	5560304	2677707	4041434
Tea	1001994	797487	938838	406455	744943
Bakery	10197878	16804926	21361087	8731632	11325520
Health Drinks	1456274	4588049	5792470	2061809	4260958
Alcoholic Drinks	6415757	18261094	20796408	8006101	17697251
Snacks	577363	1423668	1528300	492191	1780800
Confectionary	363390	1883308	1639046	647845	1250165
Cool Drinks	690706	1041085	961880	322948	564927

FILTERING RESULTS BASED ON CUSTOMIZED SORTING

4.7 Group & Ungroup

Merging and demerging the data based on logical groups is known as Packing/Unpacking. Grouping is used to merge selective cells into one cell. Ungrouping can be used to demerge the grouped data.



Ungrouped Data				
ProductCategory	GrossSales			
	2010	2011	2012	2013
Alcoholic Drinks	14706701	17570431	19735160	19164319
Bakery	19456392	19075592	11794632	18094427
Confectionary	1140009	1284075	1751065	1608605
Cool Drinks	610659	827055	1054727	1089105
Fruit Juices	7213353	8155245	8332277	6978641
Health Drinks	3236351	4460681	4893260	5569268
Ice Cream	4117144	5936579	6187398	5851498
Snacks	1142557	1300112	1893006	1466646
Tea	1136134	979937	916533	857114

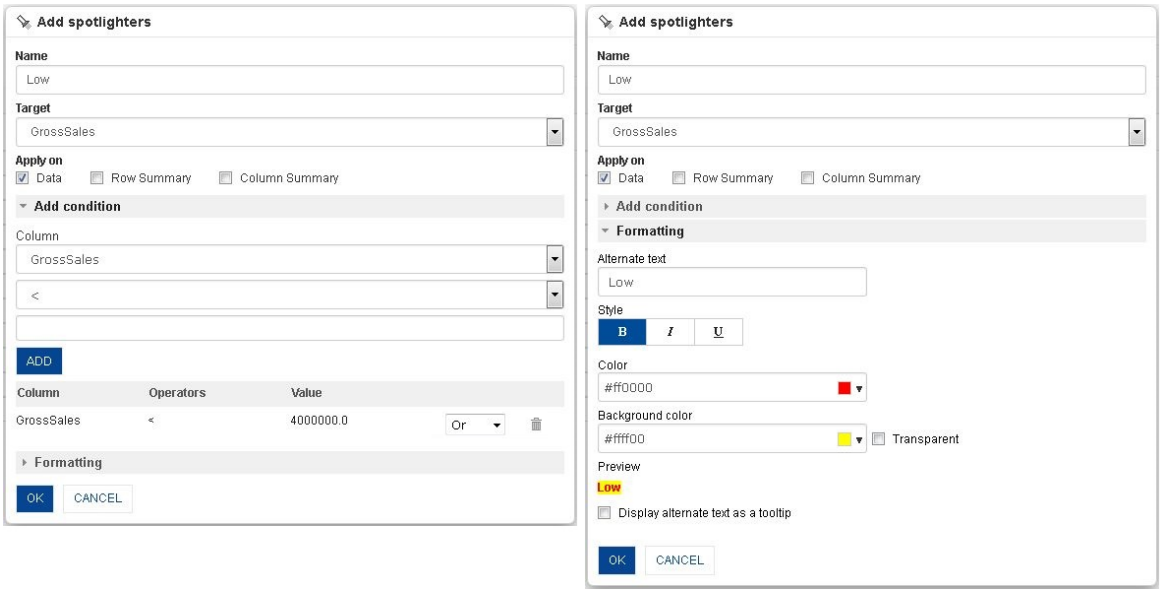
Grouped Data				
ProductCategory	GrossSales			
	2010	2011	2012	2013
Alcoholic Drinks	14706701	17570431	19735160	19164319
Non Alcoholic Drinks	12196498	14422917	15196797	14494128
Eatables	21738958	27596358	21626102	27021176

GROUPED & UNGROUPED DATA

4.8 Spotlighter

Spotlighting is used to highlight specific values based on certain conditions to identify exceptions and variations in a quick glance.

For example, to indicate the sales quantity fields with value less than 4000000 as "*low*" and also to change the field background colour, **Spotlighting** can be used.



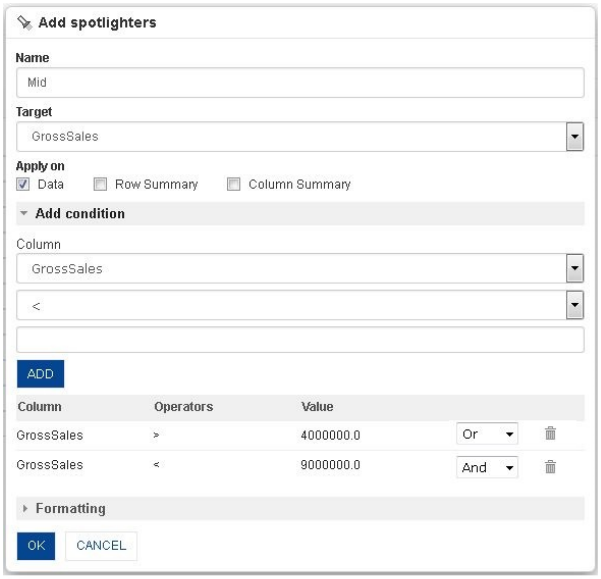
	Arizona	Arkansas	Florida	Ohio	Washington
ProductCategory	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Alcoholic Drinks	5679095.73	16949594.71	20425188.00	8062648.39	16994102.58
Bakery	8347786.60	15307382.86	21402433.27	8305388.30	11260771.01
Confectionary	328759.72	1799493.46	1626751.02	647852.72	1166554.38
Cool Drinks	581756.13	1004665.91	946936.37	320612.81	556562.63
Fruit Juices	4017164.92	7925325.90	7990909.98	2698845.48	6861975.86
Health Drinks	1288128.24	4300531.56	5684566.61	1996790.71	4133450.28
Ice Cream	3455687.72	5571239.51	5574221.29	2659830.95	4014691.94
Snacks	541650.71	1313936.19	1498123.66	489061.77	1764224.26
Tea	873712.28	741478.02	1009316.12	420579.48	738560.22

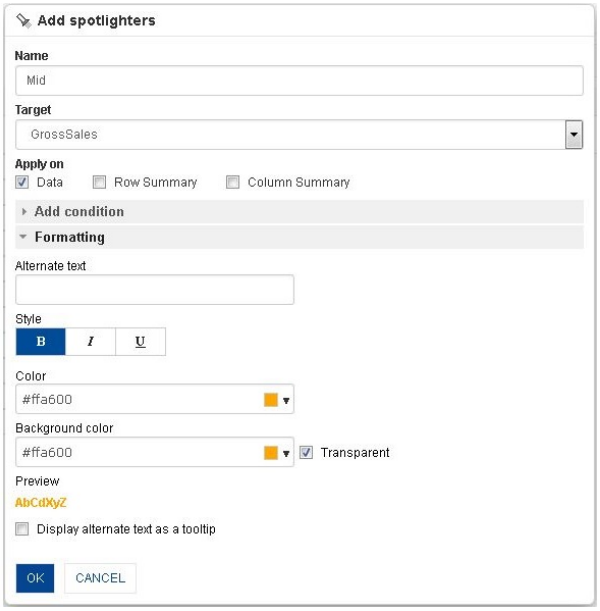
	Arizona	Arkansas	Florida	Ohio	Washington
ProductCategory	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Alcoholic Drinks	5679095.73	16949594.71	20425188.00	8062648.39	16994102.58
Bakery	8347786.60	15307382.86	21402433.27	8305388.30	11260771.01
Confectionary	Low	Low	Low	Low	Low
Cool Drinks	Low	Low	Low	Low	Low
Fruit Juices	4017164.92	7925325.90	7990909.98	Low	6861975.86
Health Drinks	Low	4300531.56	5684566.61	Low	4133450.28
Ice Cream	Low	5571239.51	5574221.29	Low	4014691.94
Snacks	Low	Low	Low	Low	Low
Tea	Low	Low	Low	Low	Low

SPOT LIGHTER SHOWING ALERTS ON LOW SALES

> Note:
> User can display static or dynamic text as alternate text for the spotlighted cells. Dynamic text will allow users to display alternate text using expressions. These expressions can be based on the columns used in the spotlighter configuration, e.g., $GrossSales – Low and $GrossSales – High in respective spotlighters will display actual Gross Sales amount, concatenate with word "Low" or "High" based on spotlighter condition, i.e., 363390 – Low, 21361087 – High.

User can apply spotlighter with multiple conditions, such as **GrossSales** greater than 40,00,000 and less than 90,00,000 as shown below.





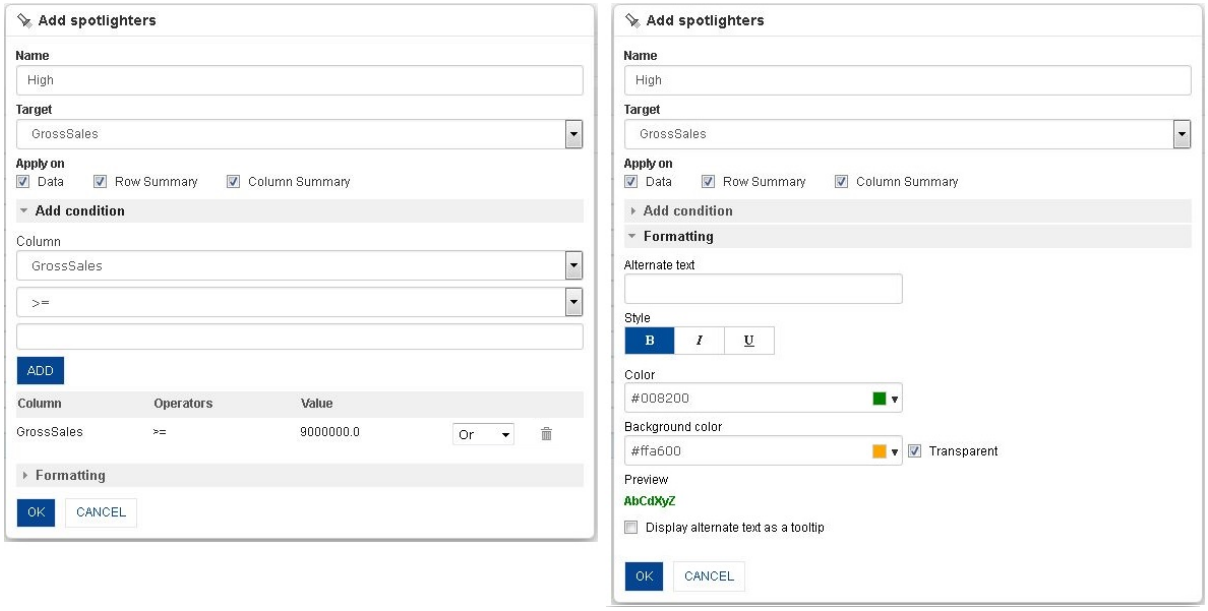
SPOTLIGHTER WITH MULTIPLE CONDITION

	Arizona	Arkansas	Florida	Ohio	Washington
ProductCategory	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Alcoholic Drinks	5679095.73	16949594.71	20425188.00	8062648.39	16994102.58
Bakery	8347786.60	15307382.86	21402433.27	8305388.30	11260771.01
Confectionary	328759.72	1799493.46	1626751.02	647852.72	1166554.38
Cool Drinks	581756.13	1004665.91	946936.37	320612.81	556562.63
Fruit Juices	4017164.92	7925325.90	7990909.98	2698845.48	6861975.86
Health Drinks	1288128.24	4300531.56	5684566.61	1996790.71	4133450.28
Ice Cream	3455687.72	5571239.51	5574221.29	2659830.95	4014691.94
Snacks	541650.71	1313936.19	1498123.66	489061.77	1764224.26
Tea	873712.28	741478.02	1009316.12	420579.48	738560.22

	Arizona	Arkansas	Florida	Ohio	Washington
ProductCategory	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Alcoholic Drinks	5679095.73	16949594.71	20425188.00	8062648.39	16994102.58
Bakery	8347786.60	15307382.86	21402433.27	8305388.30	11260771.01
Confectionary	328759.72	1799493.46	1626751.02	647852.72	1166554.38
Cool Drinks	581756.13	1004665.91	946936.37	320612.81	556562.63
Fruit Juices	4017164.92	7925325.90	7990909.98	2698845.48	6861975.86
Health Drinks	1288128.24	4300531.56	5684566.61	1996790.71	4133450.28
Ice Cream	3455687.72	5571239.51	5574221.29	2659830.95	4014691.94
Snacks	541650.71	1313936.19	1498123.66	489061.77	1764224.26
Tea	873712.28	741478.02	1009316.12	420579.48	738560.22

ANALYSIS BEFORE AND AFTER SPOTLIGHTER WITH MULTIPLE CONDITION

The spotlighter can be applied on data or on the row or column summaries as well as simultaneously on the data and the summaries.



THE OPTIONS TO APPLY THE SPOTLIGHTER ON DATA, ROW SUMMARY, COLUMN SUMMARY, OR ALL

	Arizona	Arkansas	Florida	Ohio	Washington	Summary
ProductCategory	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Alcoholic Drinks	5679095.73	16949594.71	20425188.00	8062648.39	16994102.58	68110629.41
Bakery	8347786.60	15307382.86	21402433.27	8305388.30	11260771.01	64623762.04
Confectionary	328759.72	1799493.46	1626751.02	647852.72	1166554.38	5569411.31
Cool Drinks	581756.13	1004665.91	946936.37	320612.81	556562.63	3410533.85
Fruit Juices	4017164.92	7925325.90	7990909.98	2698845.48	6861975.86	29494222.13
Health Drinks	1288128.24	4300531.56	5684566.61	1996790.71	4133450.28	17403467.39
Ice Cream	3455687.72	5571239.51	5574221.29	2659830.95	4014691.94	21275671.40
Snacks	541650.71	1313936.19	1498123.66	489061.77	1764224.26	5606996.60
Tea	873712.28	741478.02	1009316.12	420579.48	738560.22	3783646.12
Summary	25113742.04	54913648.12	66158446.32	25601610.60	47490893.15	219278340.24

	Arizona	Arkansas	Florida	Ohio	Washington	Summary
ProductCategory	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Alcoholic Drinks	5679095.73	16949594.71	20425188.00	8062648.39	16994102.58	68110629.41
Bakery	8347786.60	15307382.86	21402433.27	8305388.30	11260771.01	64623762.04
Confectionary	328759.72	1799493.46	1626751.02	647852.72	1166554.38	5569411.31
Cool Drinks	581756.13	1004665.91	946936.37	320612.81	556562.63	3410533.85
Fruit Juices	4017164.92	7925325.90	7990909.98	2698845.48	6861975.86	29494222.13
Health Drinks	1288128.24	4300531.56	5684566.61	1996790.71	4133450.28	17403467.39
Ice Cream	3455687.72	5571239.51	5574221.29	2659830.95	4014691.94	21275671.40
Snacks	541650.71	1313936.19	1498123.66	489061.77	1764224.26	5606996.60
Tea	873712.28	741478.02	1009316.12	420579.48	738560.22	3783646.12
Summary	25113742.04	54913648.12	66158446.32	25601610.60	47490893.15	219278340.24

ANALYSIS BEFORE AND AFTER SPOTLIGHT ON THE DATA, ON ROW SUMMARY, AND ON COLUMN SUMMARY

Note:
Please note that a Spot lighter created from crosstab or tabular cannot be used in GeoMap and vice versa.

4.9 Data Value / Display Value mapping

Data value / Display value mapping can display alternate text for specific field values. Displayed data (row/column) names (column headings) can be changed based on data values. For example, if quarters are available as numbers 1 to 4 (e.g., 1 for Quarter1, 2 for Quarter2), the user can specify display value for the corresponding data values from the cube. Users can view the quarter names instead of quarter numbers for a user-friendly experience.

ProductCategory	2013			
	1	2	3	4
	GrossSales	GrossSales	GrossSales	GrossSales
Alcoholic Drinks	4963311	3998298	5759433	4443277
Bakery	4177979	3380202	5565834	4970412
Confectionary	313299	567128	271379	456798
Cool Drinks	286949	440190	205681	156285
Fruit Juices	1701215	1520880	1792809	1963738
Health Drinks	879760	1750620	1390678	1548211
Ice Cream	1549218	1284510	1946652	1071116
Snacks	289046	301222	543501	332877
Tea	183340	246977	190466	236332

ProductCategory	2013			
	Q1	Q2	Q3	Q4
	GrossSales	GrossSales	GrossSales	GrossSales
Alcoholic Drinks	4963311	3998298	5759433	4443277
Bakery	4177979	3380202	5565834	4970412
Confectionary	313299	567128	271379	456798
Cool Drinks	286949	440190	205681	156285
Fruit Juices	1701215	1520880	1792809	1963738
Health Drinks	879760	1750620	1390678	1548211
Ice Cream	1549218	1284510	1946652	1071116
Snacks	289046	301222	543501	332877
Tea	183340	246977	190466	236332

ProductCategory	2013			
	Quarter1	Quarter2	Quarter3	Quarter4
	GrossSales	GrossSales	GrossSales	GrossSales
Alcoholic Drinks	4963311	3998298	5759433	4443277
Bakery	4177979	3380202	5565834	4970412
Confectionary	313299	567128	271379	456798
Cool Drinks	286949	440190	205681	156285
Fruit Juices	1701215	1520880	1792809	1963738
Health Drinks	879760	1750620	1390678	1548211
Ice Cream	1549218	1284510	1946652	1071116
Snacks	289046	301222	543501	332877
Tea	183340	246977	190466	236332

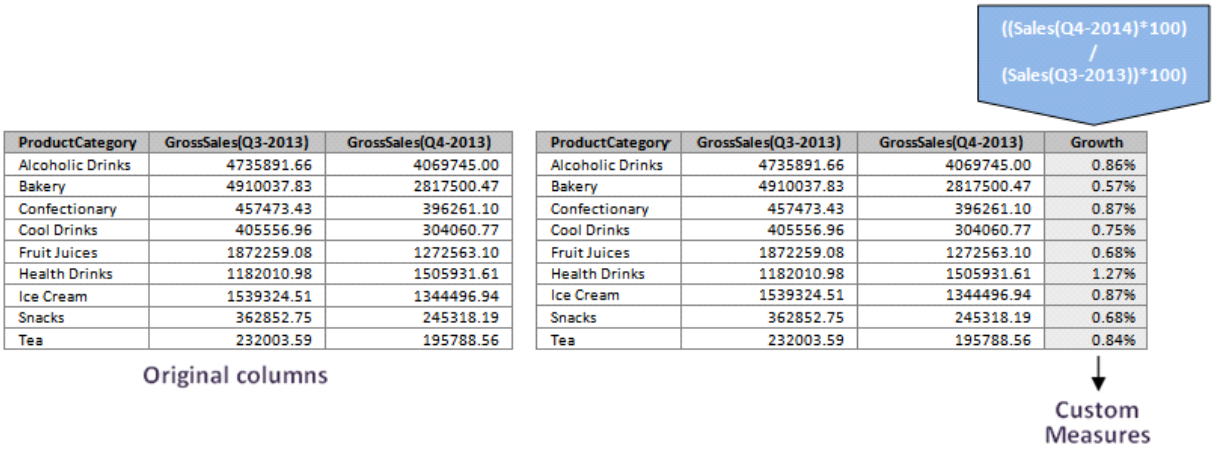
DATA VALUE/ DISPLAY VALUE MAPPING

4.10 UDDC & UDHC

4.10.1 Custom Measure (UDDC)

The custom measures in Smarten are easy to build. They can be created by building a formula on existing columns according to the crosstab or tabular requirements. The custom measures are also known as **User Defined Data Columns (UDDC)**.

Users can create custom measure columns from existing measures by performing various string, arithmetic, date, statistics, trigonometry, or conditional statements using various arithmetic operators (such as +, -, /, etc.) or comparison operators (such as =, >, < etc.).



ProductCategory	GrossSales(Q3-2013)	GrossSales(Q4-2013)
Alcoholic Drinks	4735891.66	4069745.00
Bakery	4910037.83	2817500.47
Confectionary	457473.43	396261.10
Cool Drinks	405556.96	304060.77
Fruit Juices	1872259.08	1272563.10
Health Drinks	1182010.98	1505931.61
Ice Cream	1539324.51	1344496.94
Snacks	362852.75	245318.19
Tea	232003.59	195788.56

ProductCategory	GrossSales(Q3-2013)	GrossSales(Q4-2013)	Growth
Alcoholic Drinks	4735891.66	4069745.00	0.86%
Bakery	4910037.83	2817500.47	0.57%
Confectionary	457473.43	396261.10	0.87%
Cool Drinks	405556.96	304060.77	0.75%
Fruit Juices	1872259.08	1272563.10	0.68%
Health Drinks	1182010.98	1505931.61	1.27%
Ice Cream	1539324.51	1344496.94	0.87%
Snacks	362852.75	245318.19	0.68%
Tea	232003.59	195788.56	0.84%

CUSTOM MEASURE (UDDC)

Here, **Growth** is a **Custom Measure** derived from an operation on the measures **Sales (Q4-2013)** and **Sales (Q3-2013)**. **Growth** would be available to all users as a ready-to-use measure.

Custom measures can also be created using other custom cube dimensions and measures.

For example, users can create another **Custom Measure**, **GrowthPercentage** by taking 5% of **GrossSales**. Here, the input measure is **GrossSales,** which is itself a **Custom Measure**.

Custom measures can also be created in graphs and GeoMap.

> Note:
> If UDDC is created from other columns (source columns) in the cube and the user is not granted privileges to access source columns but is granted privileges to access the resultant column, the user will be able to access the resultant column.
> For example, if a UDDC “Total_Price” is created by using the expression: Total_Price = Qty * Rate and the user is not granted access rights for Qty and / or Price column but does have rights for Total_Price, the user will be able to access the Total_Price column.
> UDDC is created on front-end data by users and not on cube data (aggregated result set of a cube). It can be used in crosstab, tabular, graphs, GeoMap and KPIs.

4.10.2 Custom Dimension Value (UDHC)

Custom dimension value columns or rows can be created by defining and applying mathematical formulae on existing row and column values as per your needs. This is also known as **User Defined Header Columns (UDHC).**

Users can create new dimension value columns by performing various conditional statements, such as string, arithmetic, date, statistics, trigonometry, or using various arithmetic operators (such as +, -, /, etc.) or comparison operators (such as =, >, < etc.) on two or more existing dimension columns or rows.

Users can also create **custom dimension values** by performing valid operations on existing dimensions.

State	Alcoholic Drinks	Cool Drinks	Fruit Juices	Health Drinks	Tea
	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Arizona	6415757	690706	4249178	1456274	1001994
Arkansas	18261094	1041085	8438495	4588049	797487
Florida	20796408	961880	8224081	5792470	938838
Ohio	8006101	322948	2722542	2061809	406455
Washington	17697251	564927	7045222	4260958	744943

Original Dimensions

Cool Drinks + Fruit Juices + Health Drinks + Tea

State	Alcoholic Drinks	Cool Drinks	Fruit Juices	Health Drinks	Tea	NonAlcoholicDrinks
	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Arizona	6415757	690706	4249178	1456274	1001994	7398152
Arkansas	18261094	1041085	8438495	4588049	797487	14865115
Florida	20796408	961880	8224081	5792470	938838	15917270
Ohio	8006101	322948	2722542	2061809	406455	5513754
Washington	17697251	564927	7045222	4260958	744943	12616049

Custom Dimension Value

CUSTOM DIMENSION VALUE (UDHC)

Product categories **Cold Drinks, Fruit Juices, Health Drinks,** and **Tea** can be grouped as "**Nonalcoholic Drinks**."

> Note:
> UDHC is created on front-end data by users and not on cube data (aggregated result set of a cube). It can be used in crosstab, tabular and graphs.
> Please note that a UDHC cannot be used in KPI and GeoMap.

Calculation Priority over Custom Measure: Users can choose the calculation priority among UDDC and UDHC while creating UDHC.

For example:
There is a Custom Measure (UDDC) column created with formula "X" (where X = rowGroupPercentage [Measure]). When users create a Custom Dimension (UHDC) with Formula "Y" (where Y = row 1 + row 2), they can have an option to prioritize the value to be displayed at the intersection cell as per formula X (based on UDDC) OR as per formula Y (based on UDHC).
In the example below for the UDHC "AA," the UDDC formula is: rowGroupPercentage (GrossSales), and the UDHC formula is: (State_Arizona + State_Arkansas).

	2014			2013		
State	CostofGoods	GrossSales	Margin %	CostofGoods	GrossSales	Margin %
Arizona	6,573,170	10,893,193	40	3,839,641	5,846,560	34
Arkansas	9,488,259	14,797,291	36	10,809,751	17,237,143	37
AA	16,061,429	25,690,484	76	14,649,392	23,083,702	71
Florida	7,297,484	12,190,108	40	10,315,676	17,608,923	41
Ohio	4,178,993	6,568,674	36	3,941,036	6,138,330	36
Washington	10,174,861	16,230,358	37	6,402,037	9,727,103	34

INTERSECTION VALUE AFTER SELECTING THE PRIORITY OVER UDDC

In this instance, 76% margin is calculated based on UDHC formula for the year 2014.

	2014			2013		
State	CostofGoods	GrossSales	Margin %	CostofGoods	GrossSales	Margin %
Arizona	6,573,170	10,893,193	40	3,839,641	5,846,560	34
Arkansas	9,488,259	14,797,291	36	10,809,751	17,237,143	37
AA	16,061,429	25,690,484	37	14,649,392	23,083,702	37
Florida	7,297,484	12,190,108	40	10,315,676	17,608,923	41
Ohio	4,178,993	6,568,674	36	3,941,036	6,138,330	36
Washington	10,174,861	16,230,358	37	6,402,037	9,727,103	34

INTERSECTION VALUE WITHOUT SELECTING THE PRIORITY OVER UDDC

In this instance, 37% margin is calculated based on UDDC formula for the year 2014.

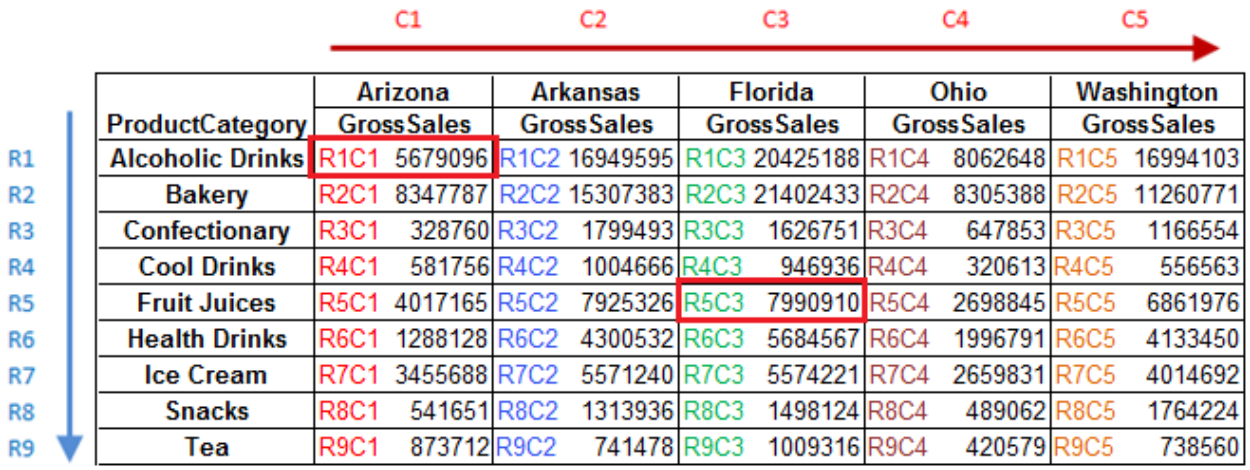
> Note:
> If UDDC or UDHC is created from other columns (source columns) in the cube and the user is not granted privileges to access source columns but is granted privileges to access the resultant column, the user will be able to access the resultant column.
>
> For example, if a UDDC "Total_Price" is created by using the expression: Total_Price = Qty * Rate and the user is not granted access rights for Qty and / or Price column but does have rights for Total_Price, the user will be able to access the Total_Price column.

4.10.3 Cell referencing in UDDC & UDHC

Cell Referencing allows users to reference a particular cell in a report and use it in user-defined data column (UDDC) and user-defined header column (UDHC) expressions.

Naming Convention of cells

A cell reference consists of Row and Column numbers that intersect at a cell's location.



ProductCategory	Arizona GrossSales	Arkansas GrossSales	Florida GrossSales	Ohio GrossSales	Washington GrossSales
Alcoholic Drinks	R1C1 5679096	R1C2 16949595	R1C3 20425188	R1C4 8062648	R1C5 16994103
Bakery	R2C1 8347787	R2C2 15307383	R2C3 21402433	R2C4 8305388	R2C5 11260771
Confectionary	R3C1 328760	R3C2 1799493	R3C3 1626751	R3C4 647853	R3C5 1166554
Cool Drinks	R4C1 581756	R4C2 1004666	R4C3 946936	R4C4 320613	R4C5 556563
Fruit Juices	R5C1 4017165	R5C2 7925326	R5C3 7990910	R5C4 2698845	R5C5 6861976
Health Drinks	R6C1 1288128	R6C2 4300532	R6C3 5684567	R6C4 1996791	R6C5 4133450
Ice Cream	R7C1 3455688	R7C2 5571240	R7C3 5574221	R7C4 2659831	R7C5 4014692
Snacks	R8C1 541651	R8C2 1313936	R8C3 1498124	R8C4 489062	R8C5 1764224
Tea	R9C1 873712	R9C2 741478	R9C3 1009316	R9C4 420579	R9C5 738560

NAMING CONVENTION OF CELLS

The **GrossSales** for **Alcoholic Drinks**, **Arizona** is referred to as **R1C1** since it is at the intersection of the first Row (R1) and the first column (C1). The **GrossSales** for Fruit Juices, Florida is referred to as **R5C3** since it is at the intersection of the fifth Row (R5) and the third column (C3).

The cell reference is not adjusted with the change of cell position based on updates in data or redesign of the report. Here, position associated with a cell is taken as a static position based reference rather than relative position based reference that keeps moving based on changed cell positions.

For example,

ProductCategory	Arizona	Arkansas	Florida	Ohio	Washington
	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Alcoholic Drinks	6415757	18261094	20796408	8006101	17697251
Bakery	10197878	16804926	21361087	8731632	11325520
Confectionary	363390	1883308	1639046	647845	1250165
Cool Drinks	690706	1041085	961880	322948	564927
Fruit Juices	4249178	8438495	8224081	2722542	7045222

To refer the **GrossSales** of **Bakery, Arizona** the position based cell reference is Second row, First column - **R2C1** with value **10197878**. Suppose, on the cube update, a row (**Aerated Drinks**) is inserted.

ProductCategory	Arizona	Arkansas	Florida	Ohio	Washington
	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Aerated Drinks	7378121	15521930	23915869	6805186	15042663
Alcoholic Drinks	6415757	18261094	20796408	8006101	17697251
Bakery	10197878	16804926	21361087	8731632	11325520
Confectionary	363390	1883308	1639046	647845	1250165
Cool Drinks	690706	1041085	961880	322948	564927
Fruit Juices	4249178	8438495	8224081	2722542	7045222

Once an additional row is inserted, **R2C1** will now refer to (second row, first column), that is the **GrossSales** of **Alcoholic Drinks, Arizona** which is **6415757.** To refer the **GrossSales** of **Bakery, Arizona** the cell reference provided has to be **R3C1**.
So, as **R2C1** is referred with static cell position based referencing, and it will assume new value of cell based on static cell position after cube updates or report redesign.

There are two types of cell references –**Absolute** and **Relative** cellreferencing.

Absolute Cell Referencing
Absolute cell referencing refers to the absolute position of a cell.

For absolute cell referencing, a cell reference should include a $ sign before the column number and / or row number. **$** indicates that cell reference is **Absolute;** it will always refer to the same position of cell (e.g., **$R2$C1** - second row, first column, in all cases).

For example,

ProductCategory	Arizona	Arkansas	Florida	Ohio	Washington
	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Alcoholic Drinks	5679096	16949595	20425188	8062648	16994103
Bakery	8347787	15307383	21402433	8305388	11260771
Confectionary	328760	1799493	1626751	647853	1166554
Cool Drinks	581756	1004666	946936	320613	556563
Fruit Juices	4017165	7925326	7990910	2698845	6861976
Health Drinks	1288128	4300532	5684567	1996791	4133450
Ice Cream	3455688	5571240	5574221	2659831	4014692
Snacks	541651	1313936	1498124	489062	1764224
Tea	873712	741478	1009316	420579	738560

To refer to the **GrossSales** of **Bakery, Arizona,** the cell reference is $**R2$C1** with value **8347787**.
Suppose, on the cube update, a row (**Aerated Drinks**) is inserted.

ProductCategory	Arizona	Arkansas	Florida	Ohio	Washington
	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Aerated Drinks	6507906	13252252	23501140	6878524	14782509
Alcoholic Drinks	5679096	16949595	20425188	8062648	16994103
Bakery	8347787	15307383	21402433	8305388	11260771
Confectionary	328760	1799493	1626751	647853	1166554
Cool Drinks	581756	1004666	946936	320613	556563
Fruit Juices	4017165	7925326	7990910	2698845	6861976
Health Drinks	1288128	4300532	5684567	1996791	4133450
Ice Cream	3455688	5571240	5574221	2659831	4014692
Snacks	541651	1313936	1498124	489062	1764224
Tea	873712	741478	1009316	420579	738560

Once an additional row is inserted, **$R2$C1** will now refer to the second row, first column of the report, that is, the **GrossSales** of **Alcoholic Drinks, Arizona,** which is **5679096.** To refer to the **GrossSales** of **Bakery, Arizona,** the cell reference provided has to be **$R3$C1**.

So, with this new data update, **$R2$C1** will return a value of **5679096 - GrossSales** of **Alcoholic Drinks, Arizona** instead of **8347787 - GrossSales** of **Bakery, Arizona**.

Relative Cell Referencing

Relative cell referencing refers to the position of a cell in relation to the current row or column being considered for the calculation.

Relative cell reference expression does not include a $ sign. For example, R2C1 refers to the value at second row and first column.
For example, there is no column dimension defined in the report. In this case, you have just one column (C1 - Gross Sales) in the report, and you are adding second column (C2) in an existing report.

ProductCategory	GrossSales
Alcoholic Drinks	R1C1 68110629.41
Bakery	R2C1 64623762.04
Confectionary	R3C1 5569411.31
Cool Drinks	R4C1 3410533.85
Fruit Juices	R5C1 29494222.13
Health Drinks	R6C1 17403467.39
Ice Cream	R7C1 21275671.40
Snacks	R8C1 5606996.60
Tea	R9C1 3783646.12

The UDDC expression to be defined for C2 will be written for new column's cell at first row (R1), that is C2R1 if you are creating second column (C2), or C3R1 if you are creating third column (C3) in the report.
For the UDDC expression, the expression is always written with reference position to the topmost left cell of the report, that is always R1C1 in any report.

	C1	C2	C3
ProductCategory	GrossSales	UDDC1 (Expression: R2C1)	UDDC2 (Expression: R2C1 - R1C1)
Alcoholic Drinks	R1C1 68110629.41	R1C2 = R2C1 64623762.04	R1C3 = (R2C1 - R1C1) -3486867.37
Bakery	R2C1 64623762.04	R2C2 = R3C1 5569411.31	R2C3 = (R3C1 - R2C1) -59054350.73
Confectionary	R3C1 5569411.31	R3C2 = R4C1 3410533.85	R3C3 = (R4C1 - R3C1) 3410533.85
Cool Drinks	R4C1 3410533.85	R4C2 = R5C1 29494222.13	R4C3 = (R5C1 - R4C1) 29494222.13
Fruit Juices	R5C1 29494222.13	R5C2 = R6C1 17403467.39	R5C3 = (R6C1 - R5C1) 17403467.39
Health Drinks	R6C1 17403467.39	R6C2 = R7C1 21275671.40	R6C3 = (R7C1 - R6C1) 21275671.40
Ice Cream	R7C1 21275671.40	R7C2 = R8C1 5606996.60	R7C3 = (R8C1 - R7C1) 5606996.60
Snacks	R8C1 5606996.60	R8C2 = R9C1 3783646.12	R8C3 = (R9C1 - R8C1) 3783646.12
Tea	R9C1 3783646.12	R9C2 = R10C1 0.00	R9C3 = (R10C1 - R9C1) 0.00

To summarise this, if you are creating second column (C2) in the report, UDDC expression will be written for expression for cell R1C2, and expression will contain reference to R1C1. If you are creating forth column (C4) in the report, UDDC expression will be written for expression for cell R1C4, and expression will contain reference to R1C1.

In the example above following expressions are used.
C2 = R2C1 means, defining value of R1C2.
C3 = R2C1 – R1C1 means, defining value of R1C3.

C2=R2C1 means that R1C2 will have value of R2C1. That means, current cell value should be fetched from cell that is one row below (from R1 to R2) and column that is one column left (from C2 to C1).

C3= R2C1 – R1C1 means that R1C3 will have value of subtraction of R1C1 from value of R2C1.
In reference to current cell R1C3, R2C1 is a cell from one row below (from R1 to R2) and column that is two columns left (from C3 to C1). And in reference to R1C3, R1C1 is a cell from same row (from R1 to R1) and column that is two columns left (from C3 to C1).

Now, if you have report with column dimension, same scenario will be replicated for all repeated column dimensions. E.g. Arizona represents first column dimension value, and Arkansas represents second column dimension value. In this case, C1 represents Sales Amount of Arizona, and C2 represents sales amount of Arkansas.

ProductCategory	Arizona	Arkansas
	GrossSales	GrossSales
Alcoholic Drinks	R1C1 5679095.73	R1C2 16949594.71
Bakery	R2C1 8347786.60	R2C2 15307382.86
Confectionary	R3C1 328759.72	R3C2 1799493.46
Cool Drinks	R4C1 581756.13	R4C2 1004665.91
Fruit Juices	R5C1 4017164.92	R5C2 7925325.90
Health Drinks	R6C1 1288128.24	R6C2 4300531.56
Ice Cream	R7C1 3455687.72	R7C2 5571239.51
Snacks	R8C1 541650.71	R8C2 1313936.20
Tea	R9C1 873712.28	R9C2 741478.02

If you add new UDDC (C2), then two UDDC columns – C2 for Arizona, and C4 for Arkansas will be created as below.

ProductCategory	Arizona		Arkansas	
	C1 GrossSales	C2 UDDC	C3 GrossSales	C4 UDDC
Alcoholic Drinks	R1C1 5679095.73	R1C2 = R1C1 5679095.73	R1C3 16949594.71	R1C4 = R1C3 16949594.71
Bakery	R2C1 8347786.60	R2C2 = R2C1 8347786.60	R2C3 15307382.86	R2C4 = R2C3 15307382.86
Confectionary	R3C1 328759.72	R3C2 = R3C1 328759.72	R3C3 1799493.46	R3C4 = R3C3 1799493.46
Cool Drinks	R4C1 581756.13	R4C2 = R4C1 581756.13	R4C3 1004665.91	R4C4 = R4C3 1004665.91
Fruit Juices	R5C1 4017164.92	R5C2 = R5C1 4017164.92	R5C3 7925325.90	R5C4 = R5C3 7925325.90
Health Drinks	R6C1 1288128.24	R6C2 = R6C1 1288128.24	R6C3 4300531.56	R6C4 = R6C3 4300531.56
Ice Cream	R7C1 3455687.72	R7C2 = R7C1 3455687.72	R7C3 5571239.51	R7C4 = R7C3 5571239.51
Snacks	R8C1 541650.71	R8C2 = R8C1 541650.71	R8C3 1313936.20	R8C4 = R8C3 1313936.20
Tea	R9C1 873712.28	R9C2 = R9C1 873712.28	R9C3 741478.02	R9C4 = R9C3 741478.02

The logic explained above for report without column dimension, will be replicated across all column dimensions in the report. So, logic explained for C2 in report without column dimension, will be replicated for C4, C6, and so on, depending on number of column dimension values.

ProductCategory	Arizona		Arkansas		Florida		Ohio		Washington	
	C1 GrossSales	C2 UDDC	C3 GrossSales	C4 UDDC	C5 GrossSales	C6 UDDC	C7 GrossSales	C8 UDDC	C9 GrossSales	C10 UDDC
Alcoholic Drinks	R1C1 5679096	R1C1 5679096	R1C3 16949595	R1C3 16949595	R1C5 20425188	R1C5 20425188	R1C7 8062648	R1C7 8062648	R1C9 16994103	R1C9 16994103
Bakery	R2C1 8347787	R2C1 8347787	R2C3 15307383	R2C3 15307383	R2C5 21402433	R2C5 21402433	R2C7 8305388	R2C7 8305388	R2C9 11260771	R2C9 11260771
Confectionary	R3C1 328760	R3C1 328760	R3C3 1799493	R3C3 1799493	R3C5 1626751	R3C5 1626751	R3C7 647853	R3C7 647853	R3C9 1166554	R3C9 1166554
Cool Drinks	R4C1 581756	R4C1 581756	R4C3 1004666	R4C3 1004666	R4C5 946936	R4C5 946936	R4C7 320613	R4C7 320613	R4C9 556563	R4C9 556563
Fruit Juices	R5C1 4017165	R5C1 4017165	R5C3 7925326	R5C3 7925326	R5C5 7990910	R5C5 7990910	R5C7 2698845	R5C7 2698845	R5C9 6861976	R5C9 6861976
Health Drinks	R6C1 1288128	R6C1 1288128	R6C3 4300532	R6C3 4300532	R6C5 5684567	R6C5 5684567	R6C7 1996791	R6C7 1996791	R6C9 4133450	R6C9 4133450
Ice Cream	R7C1 3455688	R7C1 3455688	R7C3 5571240	R7C3 5571240	R7C5 5574221	R7C5 5574221	R7C7 2659831	R7C7 2659831	R7C9 4014692	R7C9 4014692
Snacks	R8C1 541651	R8C1 541651	R8C3 1313936	R8C3 1313936	R8C5 1498124	R8C5 1498124	R8C7 489062	R8C7 489062	R8C9 1764224	R8C9 1764224
Tea	R9C1 873712	R9C1 873712	R9C3 741478	R9C3 741478	R9C5 1009316	R9C5 1009316	R9C7 420579	R9C7 420579	R9C9 738560	R9C9 738560

Building Expressions (Absolute and Relative Cell Referencing)

You can build formulas based on the absolute and relative cell referencing techniques explained above. While building expressions, all formulas are written with reference to R1C1 – first row, first column. The reference to position of row and column is based on relative value (without $ sign) and absolute (with $ sign) reference in the expression.

The table below illustrates some examples.

Cell Reference	Expression	Value
Absolute Row, Absolute Column [RxCy]	$R1$C3	Both the row and column references are absolute. Expression will always return value of the cell in first row and third column.
	$R3$C1	Both the row and column references are absolute. Expression will always return value of the cell in third row and first column.
Relative row, Absolute column [Rx$Cy]	R1$C3	Column value will remain absolute and row value will change with reference to the position of current cell. Expression will return value of the cell in third column and current row position. For example, if C2 is being created, R1C2 will represent value of R1C3, and R2C2 will represent value of R2C3 and so on. If C4 is being created, R1C4 will represent value of R1C3, and R2C4 will represent value of R2C3 and so on.
	R3$C1	Column value will remain absolute and row value will change with reference to the position of current cell. Expression will return value of the cell in first column and third row from the current row position. For example, if C2 is being created, R1C2 will represent value of R3C1, and R2C2 will represent value of R4C1 and so on. If C4 is being created, R1C4 will represent value of R3C1, and R2C4 will represent value of R4C1 and so on.
Absolute row, relative column [$RxCy]	$R1C3	Row value will remain absolute and column value will change with reference to the position of current cell. Expression will return value of the cell in third column from current column and first row position. For example, if C2 is being created, R1C2 will represent value of R1C3, and R2C2 will represent value of R1C3 and so on. If C4 is being created, R1C4 will represent value of R1C5, and R2C4 will represent

		value of R1C5 and so on.
	$R3C1	Row value will remain absolute and column value will change with reference to the position of current cell. Expression will return value of the cell in first column from current column and third row position. For example, if C2 is being created, R1C2 will represent value of R3C1, and R2C2 will represent value of R3C1 and so on. If C4 is being created, R1C4 will represent value of R3C3, and R2C4 will represent value of R3C3 and so on.
Relative Row, Relative Column [RxCy]	R1C3	Both Row value and column value will change with reference to the position of current cell. Expression will return value of the cell in third column from current column and first row from current row position. For example, if C2 is being created, R1C2 will represent value of R1C3, and R2C2 will represent value of R2C3 and so on. If C4 is being created, R1C4 will represent value of R1C5, and R2C4 will represent value of R2C5 and so on.
	R2C1	Both Row value and column value will change with reference to the position of current cell. Expression will return value of the cell in first column from current column and second row from current row position. For example, if C2 is being created, R1C2 will represent value of R2C1, and R2C2 will represent value of R3C1 and so on. If C4 is being created, R1C4 will represent value of R2C3, and R2C4 will represent value of R3C3 and so on.

Other examples:

Absolute Row, Absolute Column [$R2$C1]

ProductCategory	Arizona	Arkansas	Florida	Ohio	Washington
	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Alcoholic Drinks	R1C1 6415757	R1C2 18261094	R1C3 20796408	R1C4 8006101	R1C5 17697251
Bakery	$R2$C1 10197878	R2C2 16804926	R2C3 21361087	R2C4 8731632	R2C5 11325520
Confectionary	R3C1 363390	R3C2 1883308	R3C3 1639046	R3C4 647845	R3C5 1250165
Cool Drinks	R4C1 690706	R4C2 1041085	R4C3 961880	R4C4 322948	R4C5 564927
Fruit Juices	R5C1 4249178	R5C2 8438495	R5C3 8224081	R5C4 2722542	R5C5 7045222
Health Drinks	R6C1 1456274	R6C2 4588049	R6C3 5792470	R6C4 2061809	R6C5 4260958
Ice Cream	R7C1 3938710	R7C2 5874464	R7C3 5560304	R7C4 2677707	R7C5 4041434
Snacks	R8C1 577363	R8C2 1423668	R8C3 1528300	R8C4 492191	R8C5 1780800
Tea	R9C1 1001994	R9C2 797487	R9C3 938838	R9C4 406455	R9C5 744943

Comparing **GrossSales** of **Bakery** Category **Arizona** State with every Product Category of every State
SalesComparison = $R2$C1 - R1C1

ProductCategory	Arizona		Arkansas		Florida		Ohio		Washington	
	GrossSales	SalesComparison	GrossSales	SalesComparison	GrossSales	SalesComparison	GrossSales	SalesComparison	GrossSales	SalesComparison
Alcoholic Drinks	6415757	3782121	18261094	-8063215	20796408	-10598530	8006101	2191777	17697251	-7499373
Bakery	10197878	0	16804926	-6607047	21361087	-11163209	8731632	1466246	11325520	-1127642
Confectionary	363390	9834488	1883308	8314570	1639046	8558832	647845	9550034	1250165	8947713
Cool Drinks	690706	9507173	1041085	9156793	961880	9235998	322948	9874930	564927	9632952
Fruit Juices	4249178	5948701	8438495	1759384	8224081	1973797	2722542	7475337	7045222	3152657
Health Drinks	1456274	8741604	4588049	5609829	5792470	4405408	2061809	8136069	4260958	5936921
Ice Cream	3938710	6259169	5874464	4323415	5560304	4637574	2677707	7520171	4041434	6156445
Snacks	577363	9620515	1423668	8774210	1528300	8669579	492191	9705687	1780800	8417078
Tea	1001994	9195884	797487	9400391	938838	9259040	406455	9791423	744943	9452935

Absolute row, relative column [$R2C1]

ProductCategory	Arizona	Arkansas	Florida	Ohio	Washington
	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Alcoholic Drinks	R1C1 6415757	R1C2 18261094	R1C3 20796408	R1C4 8006101	R1C5 17697251
Bakery	$R2$C1 10197878	$R2C2 16804926	$R2C3 21361087	$R2C4 8731632	$R2C5 11325520
Confectionary	R3C1 363390	R3C2 1883308	R3C3 1639046	R3C4 647845	R3C5 1250165
Cool Drinks	R4C1 690706	R4C2 1041085	R4C3 961880	R4C4 322948	R4C5 564927
Fruit Juices	R5C1 4249178	R5C2 8438495	R5C3 8224081	R5C4 2722542	R5C5 7045222
Health Drinks	R6C1 1456274	R6C2 4588049	R6C3 5792470	R6C4 2061809	R6C5 4260958
Ice Cream	R7C1 3938710	R7C2 5874464	R7C3 5560304	R7C4 2677707	R7C5 4041434
Snacks	R8C1 577363	R8C2 1423668	R8C3 1528300	R8C4 492191	R8C5 1780800
Tea	R9C1 1001994	R9C2 797487	R9C3 938838	R9C4 406455	R9C5 744943

Comparing **GrossSales** of **Bakery** Category with **GrossSales** every Product Category Column wise
SalesComparison = $R2C1 - R1C1

ProductCategory	Arizona		Arkansas		Florida		Ohio		Washington	
	GrossSales	SalesComparison	GrossSales	SalesComparison	GrossSales	SalesComparison	GrossSales	SalesComparison	GrossSales	SalesComparison
Alcoholic Drinks	6415757	3782121	18261094	-1456168	20796408	564679	8006101	725531	17697251	-6371731
Bakery	10197878	0	16804926	0	21361087	0	8731632	0	11325520	0
Confectionary	363390	9834488	1883308	14921618	1639046	19722041	647845	8083787	1250165	10075355
Cool Drinks	690706	9507173	1041085	15763841	961880	20399207	322948	8408684	564927	10760593
Fruit Juices	4249178	5948701	8438495	8366431	8224081	13137006	2722542	6009090	7045222	4280298
Health Drinks	1456274	8741604	4588049	12216877	5792470	15568617	2061809	6669823	4260958	7064562
Ice Cream	3938710	6259169	5874464	10930462	5560304	15800783	2677707	6053925	4041434	7284086
Snacks	577363	9620515	1423668	15381258	1528300	19832787	492191	8239441	1780800	9544719
Tea	1001994	9195884	797487	16007439	938838	20422249	406455	8325177	744943	10580577

Relative Row, Absolute Column [R1$C3]

ProductCategory	Arizona	Arkansas	Florida	Ohio	Washington
	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Alcoholic Drinks	R1C1 6415757	R1C2 18261094	R1C3 20796408	R1C4 8006101	R1C5 17697251
Bakery	R2C1 10197878	R2C2 16804926	R2C3 21361087	R2C4 8731632	R2C5 11325520
Confectionary	R3C1 363390	R3C2 1883308	R3C3 1639046	R3C4 647845	R3C5 1250165
Cool Drinks	R4C1 690706	R4C2 1041085	R4C3 961880	R4C4 322948	R4C5 564927
Fruit Juices	R5C1 4249178	R5C2 8438495	R5C3 8224081	R5C4 2722542	R5C5 7045222
Health Drinks	R6C1 1456274	R6C2 4588049	R6C3 5792470	R6C4 2061809	R6C5 4260958
Ice Cream	R7C1 3938710	R7C2 5874464	R7C3 5560304	R7C4 2677707	R7C5 4041434
Snacks	R8C1 577363	R8C2 1423668	R8C3 1528300	R8C4 492191	R8C5 1780800
Tea	R9C1 1001994	R9C2 797487	R9C3 938838	R9C4 406455	R9C5 744943

Comparing **GrossSales** of **Florida** State with **GrossSales** of every ProductCategory of every State
Sales Comparison = R1$C3 - R1C1

ProductCategory	Arizona		Arkansas		Florida		Ohio		Washington	
	GrossSales	SalesComparison	GrossSales	SalesComparison	GrossSales	SalesComparison	GrossSales	SalesComparison	GrossSales	SalesComparison
Alcoholic Drinks	6415757	14380651	18261094	2535314	20796408	0	8006101	12790307	17697251	3099157
Bakery	10197878	11163209	16804926	4556161	21361087	0	8731632	12629455	11325520	10035567
Confectionary	363390	1275656	1883308	-244262	1639046	0	647845	991201	1250165	388881
Cool Drinks	690706	271174	1041085	-79205	961880	0	322948	638932	564927	396953
Fruit Juices	4249178	3974903	8438495	-214414	8224081	0	2722542	5501539	7045222	1178859
Health Drinks	1456274	4336196	4588049	1204421	5792470	0	2061809	3730661	4260958	1531512
Ice Cream	3938710	1621594	5874464	-314160	5560304	0	2677707	2882597	4041434	1518870
Snacks	577363	950937	1423668	104632	1528300	0	492191	1036109	1780800	-252500
Tea	1001994	-63156	797487	141351	938838	0	406455	532383	744943	193895

Relative Row, Relative Column [R2C1]

ProductCategory	2010	2009	2008	2007
	GrossSales	GrossSales	GrossSales	GrossSales
Alcoholic Drinks	R1C1 19164319	R1C2 19735160	R1C3 17570431	R1C4 14706701
Bakery	R2C1 18094427	R2C2 11794632	R2C3 19075592	R2C4 19456392
Confectionary	R3C1 1608605	R3C2 1751065	R3C3 1284075	R3C4 1140009
Cool Drinks	R4C1 1089105	R4C2 1054727	R4C3 827055	R4C4 610659
Fruit Juices	R5C1 6978641	R5C2 8332277	R5C3 8155245	R5C4 7213353
Health Drinks	R6C1 5569268	R6C2 4893260	R6C3 4460681	R6C4 3236351
Ice Cream	R7C1 5851498	R7C2 6187398	R7C3 5936579	R7C4 4117144
Snacks	R8C1 1466646	R8C2 1893006	R8C3 1300112	R8C4 1142557
Tea	R9C1 857114	R9C2 916533	R9C3 979937	R9C4 1136134

Comparing **GrossSales** with next category for each year
Sales Comparison = R2C1 - R1C1

ProductCategory	2010		2009		2008		2007	
	GrossSales	SalesComparison	GrossSales	SalesComparison	GrossSales	SalesComparison	GrossSale	SalesComparison
Alcoholic Drinks	19164319	-1069892	19735160	-7940528	17570431	1505161	14706701	4749691
Bakery	18094427	-16485822	11794632	-10043567	19075592	-17791517	19456392	-18316383
Confectionary	1608605	-519500	1751065	-696339	1284075	-457020	1140009	-529349
Cool Drinks	1089105	5889536	1054727	7277551	827055	7328190	610659	6602694
Fruit Juices	6978641	-1409373	8332277	-3439018	8155245	-3694564	7213353	-3977002
Health Drinks	5569268	282229	4893260	1294139	4460681	1475897	3236351	880792
Ice Cream	5851498	-4384851	6187398	-4294392	5936579	-4636466	4117144	-2974586
Snacks	1466646	-609532	1893006	-976473	1300112	-320176	1142557	-6423
Tea	857114	-857114	916533	-916533	979937	-979937	1136134	-1136134

User defined header column (UDHC) using cell reference.

Cell referencing can be applied while creating User define header column (UDHC) also.
Here the example shows state wise, city wise gross sales.

State	City	2015	2014	2013	2012
		Gross Sales	Gross Sales	Gross Sales	Gross Sales
Arizona	Phoenix	R1C1 3631634.85	R1C2 2527387.13	R1C3 3758783.23	R1C4 1755779.58
	Scottsdale	R2C1 7285432.04	R2C2 3342957.79	R2C3 1887118.78	R2C4 924648.65
	Summary	R3C1 **10917066.89**	R3C2 **5870344.92**	R3C3 **5645902.01**	R3C4 **2680428.24**
Arkansas	Conway	R4C1 8809658.88	R4C2 13268144.52	R4C3 10412512.16	R4C4 8929994.88
	Springdale	R5C1 6016127.23	R5C2 3997160.65	R5C3 2675373.16	R5C4 804676.64
	Summary	R6C1 **14825786.11**	R6C2 **17265305.16**	R6C3 **13087885.32**	R6C4 **9734671.53**
Florida	Lakeland	R7C1 6940259.28	R7C2 4756098.74	R7C3 3109996.53	R7C4 2746530.17
	Orlando	R8C1 6722827.62	R8C2 14316664.77	R8C3 13542017.53	R8C4 14024051.67
	Summary	R9C1 **13663086.91**	R9C2 **19072763.51**	R9C3 **16652014.06**	R9C4 **16770581.84**

A new row is being created to show summary that shows sum of two states (Arizona and Arkansas) minus some of one state (Florida).

In the new row (R10) being created, first cell value (Cell position - R10C1) should be represent difference between sum of Arizona (R3C1) and Arkansas (R6C1) and sum of Florida (R9C1). So, this formula has to be built based on absolute row reference and relative reference for current column for each moving column with new row R10.

For an expression (R3C1 + R6C1) – R9C1, the calculation performed is shown below.

State	City	2015	2014	2013	2012
		Gross Sales	Gross Sales	Gross Sales	Gross Sales
Arizona	Phoenix	R1C1 3631634.85	R1C2 2527387.13	R1C3 3758783.23	R1C4 1755779.58
	Scottsdale	R2C1 7285432.04	R2C2 3342957.79	R2C3 1887118.78	R2C4 924648.65
	Summary	R3C1 **10917066.89**	R3C2 **5870344.92**	R3C3 **5645902.01**	R3C4 **2680428.24**
Arkansas	Conway	R4C1 8809658.88	R4C2 13268144.52	R4C3 10412512.16	R4C4 8929994.88
	Springdale	R5C1 6016127.23	R5C2 3997160.65	R5C3 2675373.16	R5C4 804676.64
	Summary	R6C1 **14825786.11**	R6C2 **17265305.16**	R6C3 **13087885.32**	R6C4 **9734671.53**
Florida	Lakeland	R7C1 6940259.28	R7C2 4756098.74	R7C3 3109996.53	R7C4 2746530.17
	Orlando	R8C1 6722827.62	R8C2 14316664.77	R8C3 13542017.53	R8C4 14024051.67
	Summary	R9C1 **13663086.91**	R9C2 **19072763.51**	R9C3 **16652014.06**	R9C4 **16770581.84**
Sales Comparision		(R3C1+R6C1) - (R9C1) 12079766.09	(R3C2+R6C2) - (R9C2) 4062886.57	(R3C3+R6C3) - (R9C3) 2081773.26	(R3C4+R6C4) - (R9C4) -4355482.08
	Summary	**12079766.09**	**4062886.57**	**2081773.26**	**-4355482.08**

While creating values for new row R10, as column changes, Row (R3, R6 and R9) remains same in all cases whereas column (C1) relatively changed based on the current column location of the current cell.

4.10.4 Functions & Expressions

Arithmetic Functions	
Functions	Description
ABS	Returns the absolute value of a number
CEIL	Returns the smallest whole number that is greater than or equal to a specified number
EXP	Returns the exponential value of a number
FACT	Returns the factorial of a number
FLOOR	Returns the largest whole number that is smaller than or equal to a specified number
LOG	Returns the natural logarithm (base e) of a number
LOGTEN	Returns the decimal logarithm (base 10) of a number
MAX	Returns the larger of two numbers

MIN	Returns the smaller of two numbers
MOD	Returns the modulus of two numbers (the remainder after dividing the first number into the other number)
PI	Returns pi (3.14159265358979323) times a number
RANDOM	Returns a random whole number between two specified numbers
ROUND	Returns a number rounded off decimal numbers
SIGN	Returns a number (-1, 0, or 1) indicating the sign of a number
SQRT	Returns the square root of a number
TRUNCATE	Returns a number truncated to a specified number of decimal places

Date Functions	
Functions	Description
DatePart (period, source)	datePart("d",dateTime("2001-02-16 20:38:40")) Returns 16 datePart("m",dateTime("2001-02-16 20:38:40")) Returns 2 datePart("y",dateTime("2001-02-16 20:38:40")) Returns 2001 datePart("q",dateTime("2001-02-16 20:38:40")) Returns 1 datePart("h",dateTime("2001-02-16 20:38:40")) Returns 20 datePart("n",dateTime("2001-02-16 20:38:40")) Returns 38 datePart("s",dateTime("2001-02-16 20:38:40")) Returns 40 datePart("w",dateTime("2001-02-16 20:38:40")) Returns 7 Return Value: Returns an Integer value containing the specified component of a given Date value.
DateAdd (type, date, value)	dateAdd("d",10,dateTime("2001-02-16 20:38:40")) Returns 26-Feb-2001 20:38:40 dateAdd("m",2,dateTime("2001-02-16 20:38:40")) Returns 16-Apr-2001 20:38:40 dateAdd("y",2,dateTime("2001-02-16 20:38:40")) Returns 16-Feb-2003 20:38:40 dateAdd("q",2,dateTime("2001-02-16 20:38:40")) Returns 16-Aug-2001 20:38:40 dateAdd("w",2,dateTime("2001-02-16 20:38:40")) Returns 02-Mar-2001 20:38:40 dateAdd("h",2,dateTime("2001-02-16 20:38:40")) Returns 16-Feb-2001 22:38:40 dateAdd("n",2,dateTime("2001-02-16 20:38:40")) Returns 16-Feb-2001 20:40:40 dateAdd("s",2,dateTime("2001-02-16 20:38:40")) Returns 16-Feb-2001 20:38:42 Return Value: Returns a Date value containing a date and time value to which a specified time interval has been added.
DateDiff (type, date1, date2)	dateDiff("d", dateTime("2001-02-18 20:38:40"),dateTime("2001-02-16 20:38:40")) Returns 2 dateDiff("m", dateTime("2001-02-16 20:38:40"),dateTime("2001-05-16 20:38:40")) Returns -3 dateDiff("y", dateTime("2003-02-16 20:38:40"),dateTime("2001-02-16 20:38:40")) Returns 2 dateDiff("q", dateTime("2001-07-16 20:38:40"),dateTime("2001-02-16 20:38:40")) Returns 2 dateDiff("w", dateTime("2001-02-18 20:38:40"),dateTime("2001-02-06 20:38:40")) Returns 2 dateDiff("h", dateTime("2001-02-16 20:38:40"),dateTime("2001-02-16 10:38:40")) Returns 10

	dateDiff("n", dateTime("2001-02-16 20:38:40"),dateTime("2001-02-16 20:18:40")) Returns 20 dateDiff("s", dateTime("2001-02-16 20:38:40"),dateTime("2001-02-16 20:38:10")) Returns 30 Return Value: Returns a Long value specifying the number of time intervals between two Date values.
MonthName (number1, [abbreviate], [number2])	monthName(1,false, 1) Returns January monthName(1,true, 1) Returns Jan Return Value: Returns a month name representing the month for a number from 1 to 12.
WeekdayName (number1, [abbreviate], [number2])	weekdayName(2, true, 3) Returns Wed weekdayName(2, false, 3) Returns Wednesday Return Value: Returns a day name representing the day of the week for a number from 1 to 7.
FormatDate (date, “string”)	FormatDate ('2001-02-16',’yy/mm/dd’) Returns 01/02/14 formatDate(dateTime("2001-02-16 20:38:40"), "MM/dd/yyyy") Returns 02/16/2001 Return Value: Returns string of the specified format for a specified date.
date(object)	date("2001-02-16") Returns 16-Feb-2001
dateTime(object)	dateTime("2001-02-16 20:38:40") Returns 16-Feb-2001 20:38:40
day(date)	day(dateTime("2001-02-16 20:38:40")) Returns 16
dayName (date)	dayName(dateTime("2001-02-16 20:38:40")) Returns Friday
dayNumber(date)	dayNumber(dateTime("2001-02-16 20:38:40")) Returns 6
daysAfter(date , date)	daysAfter(dateTime("2001-02-16 20:38:40"),dateTime("2001-02-10 20:38:40")) Returns 6
hour(date)	hour(dateTime("2001-02-16 20:38:40")) Returns 20
minute(date)	minute(dateTime("2001-02-16 20:38:40")) Returns 38
month(date)	month(dateTime("2001-02-16 20:38:40")) Returns 2
now()	now() Returns 20:38:40 Return value : Returns current time
relativeDate(date, i)	relativeDate(dateTime("2001-02-16 20:38:40"), 5) Returns Wed Feb 21 20:38:40 IST 2001 Return value: Returns the date that occurs n days after a given date
time(object)	time("20:38:40") Returns 20:38:40
relativeTime(time , I)	relativeTime(time("20:38:40"), 5) Returns 20:38:45 Return value: Returns the time that occurs n seconds after a given time
second(time)	second(time("20:38:40")) Returns 40
today( )	today( ) Returns 16-Feb-2001 Return value: Returns the current system date
year(date)	year(dateTime("2001-02-16 20:38:40")) Returns 2001

Statistic Functions (Only applicable for Custom Measure)	
Functions	Description
AVG	Returns the average value of the underlying rows for a particular aggregated row
COUNT	Returns the count of the underlying rows for a particular aggregated row
MAXIMUM	Returns the maximum value of the underlying rows for a particular aggregated row
MINIMUM	Returns the minimum value of the underlying rows for a particular aggregated row
SUM	Returns the sum total of the underlying rows for a particular aggregated

	row
EFFECTIVE AVERAGE	Returns the effective average of the underlying rows for a particular aggregated row
EFFECTIVE COUNT	Returns the effective count of the underlying rows for a particular aggregated row
ROW PERCENTAGE	Returns the row percentage
ROW GROUP PERCENTAGE	Returns the row group percentage
COLUMN PERCENTAGE	Returns the column percentage
COLUMN GROUP PERCENTAGE	Returns the column group percentage
TOTAL PERCENTAGE	Returns the total percentage
RELATIVE ROW DIFFERENCE	Returns the relative row difference
RELATIVE ROW DIFFERENCE PERCENTAGE	Returns the relative row difference percentage
RELATIVE COLUMN DIFFERENCE	Returns the relative column difference
RELATIVE COLUMN DIFFERENCE PERCENTAGE	Returns the relative column difference percentage
ROW CUMULATIVE SUM	Returns the row cumulative sum
COLUMN CUMULATIVE SUM	Returns the column cumulative sum
FIRST	Returns the first value of the underlying rows for a particular aggregated row
LAST	Returns the last value of the underlying rows for a particular aggregated row
DISTINCT COUNT	Returns the count of unique dimension values of selected dimension from the underlying rows for a particular aggregated row
DISTINCT SUM	Returns the sum of measure for unique dimension values of selected dimension from the underlying rows for a particular aggregated row
DISTINCT AVERAGE	Returns the average of measure for unique dimension values of selected dimension from the underlying rows for a particular aggregated row
MOST RECENT	Returns the result of different aggregation methods on most recent records from the data as per the date dimensions
LEAST RECENT	Returns the result of different aggregation methods on the first records from the data as per the date dimensions

String Functions	
Functions	Description
ASC	Returns the ASCII value of a character
BOOLEANVALUE	Returns contents of a string as Boolean
BYTEVALUE	Returns contents of a string as byte
CHARVALUE	Returns contents of an integer as character
DOUBLEVALUE	Returns contents of a string as double
FILL	Returns a string of a specified length filled with occurrences of a specified string
FLOATVALUE	Returns contents of a string as float
INDEXOFCHAR	Returns the starting position of a character within a specified string
INDEXOFSTRING	Returns the starting position of a string within a specified string

INTVALUE	Returns contents of a string as integer
ISDATE	Determine if the specified string contains a valid date
ISNULL	Determine if the argument is NULL
ISNUMBER	Determine if the specified string contains a number
ISTIME	Determine if the specified string contains a valid time
LEFT	Returns a specified number of characters from a string starting with the first character
LEFTTRIM	Returns a copy of a specified string with leading blanks removed
LENGTH	Returns length of a string
LONGVALUE	Returns contents of a string as long
MATCH	Returns a determination whether a string contains a particular pattern of characters
REPLACE	Returns a copy of a specified string in which a specified number of characters starting with a specified character have been replaced with characters from another specified string
REVERSE	Reverses the order or characters in a string
RIGHT	Returns the specified number of characters from the end of a specified string
RIGHTTRIM	Returns a copy of a specified string with trailing blanks removed
SHORTVALUE	Returns contents of a string as short
SPACE	Returns a string of a specified length filled with a specified number of spaces
SUBSTRING	Returns a string containing a character copied (starting at a specified position and ending at a specified position) from a specified string
TOLOWERCASE	Returns a copy of a specified string with all uppercase letters converted to lowercase
TOSTRING	Returns a string representation of a specified object
TOUPPERCASE	Returns a copy of a specified string with all lowercase letters converted to uppercase
TRIM	Returns a string with leading and trailing blanks removed

Trigonometric Functions	
Functions	Description
COS	Cosine of number (number in radian)
SIN	Sine of number (number in radian)
TAN	Tangent of number (number in radian)
Miscellaneous Functions	
Functions	Description
IFCASE	Returns TRUE if condition is validated and returns FALSE if invalidated
WHENTHEN	Tests values of a column or expression and returns values based on the results of the test
noOfDaysByDate(StartDate, EndDate)	Returns number of days between given start and end date Example: noOfDaysByDate("2014-03-10", "2014-04-10") : returns 32
noOfDaysByFrequency(PeriodFrequency, PeriodNo,isFinancialYear)	Returns number of days in a specified frequency for a given period PeriodFrequency possible values are: “y” for yearly, “h” for half yearly, “q” for quarterly, “m” for monthly, “w” weekly, and “d” for daily PeriodNo possible values are: 0 for current period, -1 for previous period, -2 for previous to previous period, and so on isFinancialYear possible values are: true for Financial year, false for Calendar year

	Example: noOfDaysByFrequency(“y”,0,false) : returns 365 total days of current year, if current year is 2015 noOfDaysByFrequency(“m”,-1,false) : returns 30 total days of previous month, if current month is May noOfDaysByFrequency(“q”,0,true) : returns 91 total days of current quarter of current financial year, if current quarter is quarter 1 and financial year starts from April
noOfWeeksByDate(Start Date, EndDate)	Returns number of weeks between given start and end dates. Example: noOfWeeksByDate("2014-01-01", "2015-01-01") : returns 52 noOfWeeksByDate("2014-01-01", "2014-07-01") : returns 25 noOfWeeksByDate("2014-01-01", "2014-01-03") : returns 0 noOfWeeksByDate("2014-01-01", "2014-01-12") : returns 1
noOfWeeksByFrequency (PeriodFrequency, PeriodNo,isFinancialYear)	Returns number of weeks in a specified frequency for a given period PeriodFrequency possible values are: “y” for yearly, “h” for half yearly, “q” for quarterly, “m” for monthly, “w” weekly, “d” for daily PeriodNo possible values are: 0 for current period, -1 for previous period, -2 for previous to previous period, and so on isFinancialYear possible values are: true for Financial year, false for Calendar year Example: noOfWeeksByFrequency(“y”,0,false) : returns 52 total weeks of current year noOfWeeksByFrequency(“m”,-1,false) : returns 5 total weeks of previous month, if current month is May noOfWeeksByFrequency (“q”,0,true) : returns 14 total weeks of current quarter of current financial year, if current quarter is quarter 1 and financial year starts from April
noOfMonthsByDate(StartDate, EndDate)	Returns number of months between given start and end dates Example: noOfMonthsByDate("2014-01-01", "2014-12-31") : returns 12 noOfMonthsByDate("2014-01-01", "2014-07-10") : returns 6 noOfMonthsByDate("2014-01-01", "2014-05-15") : returns 4
noOfMonthsByFrequency(PeriodFrequency, PeriodNo)	Returns number of months in a specified frequency for a given period PeriodFrequency possible values are: “y” for yearly, “h” for half yearly, “q” for quarterly, “m” for monthly, “w” weekly, “d” for daily PeriodNo possible values are: 0 for current period, -1 for previous period, -2 for previous to previous period, and so on. Example: noOfMonthsByFrequency(“y”,0) : returns 12 total months of current year noOfMonthsByFrequency(“q”,-1) : returns 3 total months of previous quarter
noOfQuartersByDate(StartDate, EndDate)	Returns number of quarters between given start and end dates

	Example: noOfQuatersByDate("2014-01-01", "2014-12-31") : returns 4 noOfQuatersByDate("2014-01-01", "2014-08-15") : returns 2
noOfQuartersByFrequency(PeriodFrequency, PeriodNo)	Returns number of quarters in a specified frequency for a given period PeriodFrequency possible values are: “y” for yearly, “h” for half yearly, “q” for quarterly, “m” for monthly, “w” weekly, “d” for daily PeriodNo possible values are: 0 for current period, -1 for previous period, -2 for previous to previous period, and so on. Example: noOfQuatersByFrequency(“y”,0) : returns 4 total quarters of current year noOfQuatersByFrequency(“y”,-1) : returns 4 total quarters of previous year
noOfHalfYearsByDate(StartDate, EndDate)	Returns number of half years between a given start and end date Example: noOfHalfYearsByDate("2014-01-01", "2014-12-31") : returns 2 noOfHalfYearsByDate("2014-01-01", "2014-05-31") : returns 0 noOfHalfYearsByDate("2014-01-01", "2014-08-31") : returns 1
noOfHalfYearsByFrequency(PeriodFrequency, PeriodNo)	Returns number of half years in a specified frequency for a given period PeriodFrequency possible values are: “y” for yearly, “h” for half yearly, “q” for quarterly, “m” for monthly, “w” weekly, “d” for daily PeriodNo possible values are: 0 for current period, -1 for previous period, -2 for previous to previous period, and so on. Example: noOfHalfYearsByFrequency(“y”,0) : returns 2 total half years of current year noOfHalfYearsByFrequency(“q”,0) : returns 0 as frequency is quarterly
For KPI only	
KPIDateDimension()	Returns selected date dimension name in a KPI

	For example, if “SalesDate” date dimension is selected in KPI settings, it will return “SalesDate.” This function will be useful in functions like mostRecent and leastRecent. In these functions, date dimension is one of the parameters. Rather than specifying a static date dimension name in an expression, you can pass dynamic date dimension name through this function. Example: Static : mostRecent(GrossSales, “SalesDate”, “sum”, “”, “”) Dynamic : mostRecent(GrossSales, KPIDateDimension(), “sum”, “”, “”)
KPIIsFinancialYear()	Returns true if Financial Year option is selected in KPI; otherwise, returns false
CurrentFrequency()	Returns currently selected frequency value in a KPI. Returns “y” for yearly, “h” for half yearly, “q” for quarterly, “m” for monthly, “b” for biweekly, “w” for weekly, “d” for daily
Period()	Returns the period value in a KPI. Returns 0 for current period, -1 for previous period, -2 for previous to previous period, and so on. This function will be useful in functions like noOfDaysByFrequency, noOfWeeksByFrequency, etc. In these functions, PeriodNo is one of the parameters. Rather than specifying the static period number in an expression, you can pass the dynamic period number through this function. Example: Static : noOfDaysByFrequency(“m”, 0, false) Dynamic : noOfDaysByFrequency(“m”, Period(), false)

Summary Functions	
Function Name	Description
Sum	Displays the total/sum of all values across row or column in the analysis
Average	Average of all values at the cube record level
Effective Average	Average of all non null values at the cube record level
Group Average	Average of all values across row or column in the analysis
Count	Count of all values at the cube record level
Effective Count	Count of all non null values at the cube record level
Group Count	Count of all values at the group level
Maximum	Greatest among all the values at the cube record level
Group Maximum	Greatest among all the values across row or column in the analysis
Minimum	Lowest among all the values at the cube record level
Group Minimum	Lowest among all the values across row or column in the analysis
Row Percentage	Total horizontal percentage of that corresponding row
Row Group Percentage	Total vertical percentage of a row comprising group of rows
Total Percentage	Total percentage of a row with respect to the whole analysis
Relative Row Difference	Difference with respect to the previous row/column
Relative Row Difference Percentage	Difference with respect to the previous row/column in percentage
Column Percentage	Total vertical percentage of that corresponding column
Column Group Percentage	Total horizontal percentage of a column comprising group of columns
Relative Column Difference	Difference with respect to the previous column
Relative Column Difference Percentage	Difference with respect to the previous row/column in percentage

Row Cumulative Sum	Sum with respect to previous row

4.11 Data Operations

User can apply data operations on the fly while analysing data through crosstab, tabular, graph and GeoMap.

Example data set for all examples in this section:

Transaction id	State	City	Product Category	Product	Gross Sales
A1	Florida	Miami	Bakery	Bread	34
A2	Florida	Miami	Bakery	Bun	46
A3	Florida	Miami	Cool Drinks	Cola	17
A4	Florida	Miami	Cool Drinks	Soda	56
A5	Florida	Miami	Bakery	Bun	44
A6	Florida	Orlando	Bakery	Bread	25
A7	Florida	Orlando	Bakery	Bun	34
A8	Florida	Orlando	Cool Drinks	Cola	NULL
A9	Florida	Orlando	CoolDrinks	Soda	NULL
A10	Florida	Orlando	CoolDrinks	Cola	38
A11	Florida	Orlando	Cool Drinks	Soda	34
A12	Arizona	Phoenix	Bakery	Bread	12
A13	Arizona	Phoenix	Bakery	Bun	NULL
A14	Arizona	Phoenix	Cool Drinks	Cola	39
A15	Arizona	Phoenix	Cool Drinks	Soda	28
A16	Arizona	Phoenix	Bakery	Bread	25
A17	Arizona	Scottsdale	Bakery	Bread	17
A18	Arizona	Scottsdale	Bakery	Bun	6
A19	Arizona	Scottsdale	Cool Drinks	Cola	NULL
A20	Arizona	Scottsdale	Cool Drinks	Soda	68
A21	Arizona	Scottsdale	Bakery	Bun	34

List of all data operations:

1.	Sum	Data operations applied on cube row level data.
2.	Average	
3.	Effective Average	
4.	Count	
5.	Effective Count	
6.	Maximum	
7.	Minimum	
8.	First	
9.	Last	
10.	Distinct Count	
11.	Distinct Sum	
12.	Distinct Average	
13.	Most Recent	
14.	Least Recent	
15.	Row Percentage	Data operations applied on the front-end object view data.
16.	Row Group Percentage	
17.	Column Percentage	
18.	Column Group Percentage	
19.	Total Percentage	
20.	Relative Row Difference	

21.	Relative Row Difference Percentage	
22.	Relative Column Difference	
23.	Relative Column Difference Percentage	
24.	Row Cumulative Sum	
25.	Column Cumulative Sum	

Row / Column Data Operations

Sum—Sum of all values

	Arizona	Florida
ProductCategory	GrossSales	GrossSales
Bakery	94	183
Cool Drinks	135	145

Sum of Sales Amount for State = Arizona
Product Category = Bakery

i.e., 94 = Sum of Sales Amount

Average—Average of all values

	Arizona	Florida
ProductCategory	GrossSales	GrossSales
Bakery	15.67	36.6
Cool Drinks	33.75	24.17

Average of Sales Amount for State = Arizona
Product Category = Bakery

i.e., 15.67 = Average of Sales Amount

Effective Average—Average of all "not null" values

Note:
Effective Average implies the average of only the rows with "not null" values. Columns with value "null" are not considered in effective average calculation, but columns with value "0" are taken into consideration for effective average calculation.

		Arizona		Florida	
		Phoenix	Scottsdale	Miami	Orlando
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	34	25
	Bun	0	40	90	34
Cool Drinks	Cola	39	0	17	38
	Soda	28	68	56	34

ANALYSIS WITH THE "SUM" DATA

		Arizona		Florida	
		Phoenix	Scottsdale	Miami	Orlando
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	18.5	17	34	25
	Bun	0	20	45	34
Cool Drinks	Cola	39	0	17	38
	Soda	28	68	56	34

ANALYSIS WITH THE "EFFECTIVE AVERAGE" DATA

Effective Average of Sales Amount for
State = Arizona
City = Phoenix
Product = Bread excluding null cell
i.e., 18.50 = Effective Average of Sales Amount
(37/2), where 2 is the number of total transactions excluding null value; (A12+A16)/2

Count—Count of rows

	Arizona	Florida
ProductCategory	GrossSales	GrossSales
Bakery	6	5
Cool Drinks	4	6

Count of Sales Amount for
State = Arizona
Product Category = Bakery
i.e., 6 = Count of total row level transactions (A12, A13, A16, A17, A18, A21)

Effective Count—Count of rows with "not null" values

	Arizona	Florida
ProductCategory	GrossSales	GrossSales
Bakery	5	5
Cool Drinks	3	4

Effective Count of Sales Amount for
State = Arizona
Product Category = Bakery
i.e., 5 = Number of total transactions excluding null values (A12, A16, A17, A18, A21)

Note:
Effective Count implies the count of only the rows with "not null" values. Columns with value "null" are not considered in effective count calculation, but a column with value "0" is taken into consideration for effective count calculation.

Maximum—Highest among all the values

	Arizona	Florida
ProductCategory	GrossSales	GrossSales
Bakery	34	46
Cool Drinks	68	56

Maximum of Sales Amount for
State = Arizona
Product Category = Bakery
i.e., 34 = Maximum of Sales Amount – A21 highest among the transactions A12, A13, A16, A17, A18, A21

Maximum of Sales Amount for
State = Florida
Product Category = Cool Drinks
i.e., 56 = Maximum of Sales Amount – A4 highest among A3, A4, A8, A9, A10, A11

Minimum—Lowest among all the values

	Arizona		Florida	
	Phoenix	Scottsdale	Miami	Orlando
ProductCategory	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	12	6	34	25
Cool Drinks	28	68	17	34

Minimum of Sales Amount for
City = Phoenix
Product Category = Bakery
i.e., 12 = Minimum of Sales Amount – A12 lowest among the transactions A12, A16, A13

Minimum of Sales Amount for
City = Miami
Product Category = Bakery
i.e., 34 = Minimum of Sales Amount – A1 lowest among the transactions A1, A2, A5

First—First among all the values

	Arizona	Florida
ProductCategory	GrossSales	GrossSales
Bakery	12	34
Cool Drinks	39	17

First value of Sales Amount for
State = Arizona
Product Category = Bakery
i.e., 12 = First value of Sales Amount – A12 first among the transactions A12, A13, A16, A17, A18, A21

First value of Sales Amount for
State = Florida
Product Category = Cool Drinks
i.e., 17 = First value of Sales Amount – A3 first among A3, A4, A8, A9, A10, A11

Last—Last among all the values

	Arizona	Florida
ProductCategory	GrossSales	GrossSales
Bakery	34	34
Cool Drinks	68	34

Last value of Sales Amount for
State = Arizona
Product Category = Bakery
i.e., 34 = Last value of Sales Amount – A21 last among the transactions A12, A13, A16, A17, A18, A21

Last value of Sales Amount for
State = Florida
Product Category = Cool Drinks
i.e., 34 = Last value of Sales Amount – A11 last among A3, A4, A8, A9, A10, A11

Row Percentage—Percentage value against row level summary within the same row

		Arizona		Florida		Total
		Phoenix	Scottsdale	Miami	Orlando	GrossSales
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	
Bakery	Bread	37	17	34	25	**113**
	Bun	0	40	90	34	**164**
Cool Drinks	Cola	39	0	17	38	**94**
	Soda	28	68	56	34	**186**

ANALYSIS WITH THE "SUM" DATA

		Arizona		Florida		Total
		Phoenix	Scottsdale	Miami	Orlando	GrossSales
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	
Bakery	Bread	32.74	15.04	30.09	22.12	**100**
	Bun	0	24.39	54.88	20.73	**100**
Cool Drinks	Cola	41.49	0	18.09	40.43	**100**
	Soda	15.05	36.56	30.11	18.28	**100**

ANALYSIS WITH THE "ROW PERCENTAGE" DATA OPERATION

Row percentage of Sales Amount for State = Arizona
City = Phoenix
Product = Bread
*i.e., 32.74 = Row percentage of Sales Amount (37*100/113)*

Row percentage of Sales Amount for
State = Arizona
City = Scottsdale
Product = Soda
*i.e., 36.56 = Row percentage of Sales Amount (68*100)/186*

ROW GROUP PERCENTAGE—PERCENTAGE VALUE AGAINST THE ROW GROUP LEVEL SUMMARY (WITHIN THE SAME GROUP)

		Arizona		Florida	
		Phoenix	Scottsdale	Miami	Orlando
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	34	25
	Bun	0	40	90	34
Cool Drinks	Cola	39	0	17	38
	Soda	28	68	56	34

ANALYSIS WITH THE "SUM" DATA

		Arizona		Florida	
		Phoenix	Scottsdale	Miami	Orlando
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	100	29.82	27.42	42.37
	Bun	0	70.18	72.58	57.63
Cool Drinks	Cola	58.21	0	23.29	52.78
	Soda	41.79	100	76.71	47.22

ANALYSIS WITH THE "ROW GROUP PERCENTAGE" DATA

Row group percentage of Sales Amount for
City = Phoenix
Product Category = Cool Drinks
Product = Cola
i.e., 58.21 = Row group percentage of the sum of the group sales amount for the group Cool Drinks
*(58.21 = 39*100)/ (39+28)*

Row group percentage of Sales Amount for
City = Scottsdale
Product Category = Bakery
Product = Bun
i.e., 70.18 = Row group percentage of the sum of the group sales amount for the group Bakery

Row group percentage of Sales Amount for
City = Miami
Product Category = Cool Drinks
Product = Cola
i.e., 23.29 = Row group percentage of the sum of the group sales amount for the group Cool Drinks
*(23.29 = 17*100/ (17+56)*

TOTAL PERCENTAGE—PERCENTAGE VALUE AGAINST THE TOTAL CROSSTAB SUM

		Arizona		Florida		Total
		Phoenix	Scottsdale	Miami	Orlando	GrossSales
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	
Bakery	Bread	37	17	34	25	**113**
	Bun	0	40	90	34	**164**
Cool Drinks	Cola	39	0	17	38	**94**
	Soda	28	68	56	34	**186**
Total		**104**	**125**	**197**	**131**	**557**

ANALYSIS WITH THE "SUM" DATA

		Arizona		Florida		Total
		Phoenix	Scottsdale	Miami	Orlando	GrossSales
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	
Bakery	Bread	6.64	3.05	6.1	4.49	**20.29**
	Bun	0	7.18	16.16	6.1	**29.44**
Cool Drinks	Cola	7	0	3.05	6.82	**16.88**
	Soda	5.03	12.21	10.05	6.1	**33.39**
Total		**18.67**	**22.44**	**35.37**	**23.52**	**100**

ANALYSIS WITH THE "TOTAL PERCENTAGE" DATA OPERATION

Total percentage of Sales Amount for
State = Arizona
City = Phoenix
Product = Bread
*i.e., 6.64 = Total percentage of Sales Amount (37*100/557)*

Total percentage of Sales Amount for
State = Arizona
City = Scottsdale
Product = Soda
*i.e., 12.21 = Total percentage of Sales Amount (68*100)/557*

Total percentage of Sales Amount for
State = Florida
City = Orlando
*i.e., 6.1 = Total percentage of Sales Amount (34*100)/557*

RELATIVE ROW DIFFERENCE—DIFFERENCE WITH RESPECT TO THE PREVIOUS ROW

		Arizona		Florida	
		Phoenix	Scottsdale	Miami	Orlando
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	34	25
	Bun	0	40	90	34
Cool Drinks	Cola	39	0	17	38
	Soda	28	68	56	34

ANALYSIS WITH THE "SUM" DATA

		Arizona		Florida	
		Phoenix	Scottsdale	Miami	Orlando
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	34	25
	Bun	-37	23	56	9
Cool Drinks	Cola	39	-40	-73	4
	Soda	-11	68	39	-4

ANALYSIS WITH THE "RELATIVE ROW DIFFERENCE" DATA OPERATION

Relative Row Difference for the Sales Amount for
State = Arizona
City = Phoenix
For Product = Bread is 37{37-0}
For Product = Bun is -37{0-37}
For Product = Cola is 39{39-0}
For Product = Soda is -11{28-39}

Relative Row Difference for the Sales Amount for
State = Florida
City = Miami
For Product = Bread is 34{34-0}
For Product = Bun is 56{90-34}
For Product = Cola is -73{17-90}
For Product = Soda is 39{56-17}

RELATIVE ROW DIFFERENCE PERCENTAGE—DIFFERENCE WITH RESPECT TO THE PREVIOUS ROW VALUE IN PERCENTAGE

		Arizona		Florida	
		Phoenix	Scottsdale	Miami	Orlando
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	34	25
	Bun	0	40	90	34
Cool Drinks	Cola	39	0	17	38
	Soda	28	68	56	34

ANALYSIS WITH THE "SUM" DATA

		Arizona		Florida	
		Phoenix	Scottsdale	Miami	Orlando
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	100	100	100	100
	Bun	-100	135.29	164.71	36
Cool Drinks	Cola	100	-100	-81.11	11.76
	Soda	-28.21	100	229.41	-10.53

ANALYSIS WITH THE "RELATIVE ROW DIFFERENCE PERCENTAGE" DATA OPERATION

*Relative Row Difference Percentage of Sales Amount for State = Arizona City = Scottsdale Product = Cola is -100 {(0-40)/40*100}*

*Relative Row Difference Percentage of Sales Amount for State = Florida City = Miami Product = Bun is 164.71 {(90-34)/34*100}*

*Relative Row Difference Percentage of Sales Amount for State = Florida City = Orlando Product = Soda is -10.53 {(34-38)/38*100}*

RELATIVE ROW GROUP DIFFERENCE—DIFFERENCE WITH RESPECT TO THE PREVIOUS ROW VALUE (WITHIN SAME GROUP)

		Arizona		Florida	
		Phoenix	Scottsdale	Miami	Orlando
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	34	25
	Bun	0	40	90	34
	Summary	**37**	**57**	**124**	**59**
Cool Drinks	Cola	39	0	17	38
	Soda	28	68	56	34
	Summary	**67**	**68**	**73**	**72**

ANALYSIS WITH THE "SUM" DATA

		Arizona		Florida	
		Phoenix	Scottsdale	Miami	Orlando
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	34	25
	Bun	-37	23	56	9
	Summary	**37**	**57**	**124**	**59**
Cool Drinks	Cola	39	0	17	38
	Soda	-11	68	39	-4
	Summary	**67**	**68**	**73**	**72**

ANALYSIS WITH THE "RELATIVE ROW GROUP DIFFERENCE" DATA OPERATION FOR RESPECTIVE GROUPS

Relative Row Group Difference of Sales Amount for City = Phoenix Product Category = Bakery Product = Bun i.e., -37 = Relative row difference of Sales Amount (0-37; decrease of 37 from 37) within the group Bakery

Relative Row Group Difference of Sales Amount for City = Scottsdale Product Category = Cool Drinks Product = Soda i.e., 68 = Relative row difference of Sales Amount (68-0; increase of 68 from 0) within the group Cool Drinks

Relative Row Group Difference of Sales Amount for City = Orlando Product Category = Cool Drinks Product = Soda i.e., -4 = Relative row difference of Sales Amount (34-38; decrease of 4 from 38) within the group Cool Drinks

RELATIVE ROW GROUP DIFFERENCE PERCENTAGE—DIFFERENCE WITH RESPECT TO THE PREVIOUS ROW VALUE (WITHIN SAME GROUP) IN PERCENTAGE

		Arizona		Florida	
		Phoenix	Scottsdale	Miami	Orlando
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	34	25
	Bun	0	40	90	34
	Summary	**37**	**57**	**124**	**59**
Cool Drinks	Cola	39	0	17	38
	Soda	28	68	56	34
	Summary	**67**	**68**	**73**	**72**

ANALYSIS WITH THE "SUM" DATA

		Arizona		Florida	
		Phoenix	Scottsdale	Miami	Orlando
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	100	100	100	100
	Bun	-100	135.29	164.71	36
	Summary	**37**	**57**	**124**	**59**
Cool Drinks	Cola	100	100	100	100
	Soda	-28.21	100	229.41	-10.53
	Summary	**67**	**68**	**73**	**72**

ANALYSIS WITH THE "RELATIVE ROW GROUP DIFFERENCE PERCENTAGE" DATA OPERATION

Relative Row Group Difference percentage of Sales Amount for
City = Phoenix
Product Category = Bakery
Product = Bun
i.e., -100% = Relative row difference percentage of Sales Amount (0-37; decrease of 37 from 37).
i.e., reduction of 100% within the group Bakery.

Relative Row Group Difference percentage of Sales Amount for
City = Miami
Product Category = Bakery
Product = Bun
i.e., 164.71% = Relative row difference percentage of Sales Amount (90-34; increase of 56 from 34). i.e., increase of 64.71% within the group Cool Drinks
(100(90-34))/34*

Relative Row Group Difference percentage of Sales Amount for
City = Orlando
Product Category = Cool Drinks
Product = Soda
i.e., -10.53% = Relative row difference percentage of Sales Amounts (34-38; decrease of 4 from 38) within the group Cool Drinks
(100(34-38))/38*

COLUMN PERCENTAGE—PERCENTAGE VALUE AGAINST THE COLUMN LEVEL SUMMARY

		Arizona		Florida	
		Phoenix	Scottsdale	Miami	Orlando
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	34	25
	Bun	0	40	90	34
Cool Drinks	Cola	39	0	17	38
	Soda	28	68	56	34

ANALYSIS WITH THE "SUM" DATA

		Arizona		Florida	
		Phoenix	Scottsdale	Miami	Orlando
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	35.58	13.6	17.26	19.08
	Bun	0	32	45.69	25.95
Cool Drinks	Cola	37.5	0	8.63	29.01
	Soda	26.92	54.4	28.43	25.95

ANALYSIS WITH THE "COLUMN PERCENTAGE" DATA OPERATION

Column percentage of Sales Amount for
State = Arizona
City = Phoenix
Product Category = Bakery
Product = Bread
*i.e., 35.58 = Column percentage of Sales Amount (37*100/104)*

Column percentage of Sales Amount for
State = Florida
City = Miami
Product Category = Bakery
Product = Bun
*i.e., 45.69 = Column percentage of Sales Amount (100*90/197)*

COLUMN GROUP PERCENTAGE—PERCENTAGE VALUE AGAINST THE COLUMN GROUP LEVEL SUMMARY (WITHIN THE SAME GROUP)

		Arizona			Florida		
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	**54**	34	25	**59**
	Bun	0	40	**40**	90	34	**124**
Cool Drinks	Cola	39	0	**39**	17	38	**55**
	Soda	28	68	**96**	56	34	**90**

ANALYSIS WITH THE "SUM" DATA

		Arizona			Florida		
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	68.52	31.48	**54**	57.63	42.37	**59**
	Bun	0	100	**40**	72.58	27.42	**124**
Cool Drinks	Cola	100	0	**39**	30.91	69.09	**55**
	Soda	29.17	70.83	**96**	62.22	37.78	**90**

ANALYSIS WITH THE "COLUMN GROUP PERCENTAGE" DATA OPERATION

Column group percentage of Sales Amount for
City = Phoenix
Product = Bread
*68.52 = Column group percentage of Sales Amount for the group Arizona (37*100)/54*
i.e., 37 = 68.52% of 54

Column group percentage of Sales Amount for
City = Orlando
Product = Cola
*69.09 = Column group percentage of Sales Amount for the group Florida (38*100)/55*
i.e., 38 = 69.09% of 55

RELATIVE COLUMN DIFFERENCE—DIFFERENCE WITH RESPECT TO THE PREVIOUS COLUMN VALUE

		Arizona		Florida	
		Phoenix	Scottsdale	Miami	Orlando
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	34	25
	Bun	0	40	90	34
Cool Drinks	Cola	39	0	17	38
	Soda	28	68	56	34

ANALYSIS WITH THE "SUM" DATA

		Arizona		Florida	
		Phoenix	Scottsdale	Miami	Orlando
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	-20	17	-9
	Bun	0	40	50	-56
Cool Drinks	Cola	39	-39	17	21
	Soda	28	40	-12	-22

ANALYSIS WITH THE "RELATIVE COLUMN DIFFERENCE" DATA OPERATION

Relative Column Difference of Sales Amount for
State = Arizona and Florida
Product = Bread
City = Phoenix is 37{37-0}
City = Scottsdale is -20{17-37}
City = Miami is 17{34-17}
City = Orlando is -9{25-34}

Relative Column Difference of Sales Amount for
State = Arizona and Florida
Product = Cola
City = Phoenix is 39{39-0}
City = Scottsdale is -39{0-39}
City = Miami is 17{17-0}
City = Orlando is 21{38-17}

RELATIVE COLUMN DIFFERENCE PERCENTAGE—DIFFERENCE WITH RESPECT TO THE PREVIOUS COLUMN VALUE IN PERCENTAGE

		Arizona		Florida	
		Phoenix	Scottsdale	Miami	Orlando
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	34	25
	Bun	0	40	90	34
Cool Drinks	Cola	39	0	17	38
	Soda	28	68	56	34

ANALYSIS WITH THE "SUM" DATA

		Arizona		Florida	
		Phoenix	Scottsdale	Miami	Orlando
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	100	-54.05	100	-26.47
	Bun	100	100	125	-62.22
Cool Drinks	Cola	100	-100	100	123.53
	Soda	100	142.86	-17.65	-39.29

ANALYSIS WITH THE "RELATIVE COLUMN DIFFERENCE PERCENTAGE" DATA OPERATION

*Relative Column Difference Percentage of Sales Amount for State = Arizona City = Scottsdale Product = Bread is -54.05 {(17-37)/37*100}*

*Relative Column Difference Percentage of Sales Amount for State = Florida City = Miami Product = Bun is 125 {(90-40)/40*100}*

*Relative Column Difference Percentage of Sales Amount for State = Florida City = Orlando Product = Cola is 123.53 {(38-17)/17*100}*

RELATIVE COLUMN GROUP DIFFERENCE—DIFFERENCE WITH RESPECT TO THE PREVIOUS COLUMN VALUE (WITHIN THE SAME GROUP)

		Arizona			Florida		
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	**54**	34	25	**59**
	Bun	0	40	**40**	90	34	**124**
Cool Drinks	Cola	39	0	**39**	17	38	**55**
	Soda	28	68	**96**	56	34	**90**

ANALYSIS WITH THE "SUM" DATA

		Arizona			Florida		
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	-20	**54**	34	-9	**59**
	Bun	0	40	**40**	90	-56	**124**
Cool Drinks	Cola	39	-39	**39**	17	21	**55**
	Soda	28	40	**96**	56	-22	**90**

ANALYSIS WITH THE "RELATIVE COLUMN GROUP DIFFERENCE" DATA OPERATION

Relative Column Group Difference of Sales Amount for
State = Arizona, City = Scottsdale
Product = Bread
i.e., -20 = Relative column difference of Sales Amount (17 - 37)

Relative Column Group Difference of Sales Amount for
State = Florida, City = Orlando
Product = Bun
i.e., -56 = Relative column difference of Sales Amount (34 - 90)

Relative Column Group Difference of Sales Amount for
State = Arizona, City = Scottsdale
Product = Cola
i.e., -39 = Relative column difference of Sales Amount (0 - 39)

RELATIVE COLUMN GROUP DIFFERENCE PERCENTAGE—DIFFERENCE WITH RESPECT TO THE PREVIOUS COLUMN VALUE (WITHIN THE SAME GROUP) IN PERCENTAGE

		Arizona			Florida		
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	**54**	34	25	**59**
	Bun	0	40	**40**	90	34	**124**
Cool Drinks	Cola	39	0	**39**	17	38	**55**
	Soda	28	68	**96**	56	34	**90**

ANALYSIS WITH THE "SUM" DATA

		Arizona			Florida		
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	100	-54.05	**54**	100	-26.47	**59**
	Bun	100	100	**40**	100	-62.22	**124**
Cool Drinks	Cola	100	-100	**39**	100	123.53	**55**
	Soda	100	142.86	**96**	100	-39.29	**90**

ANALYSIS WITH THE "RELATIVE COLUMN GROUP DIFFERENCE PERCENTAGE "DATA OPERATION

Relative Column Group Difference percentage of Sales Amount for State = Arizona, Product = Bread i.e., -54.05 = Relative column difference percentage of Sales Amount for City = Scottsdale (100(17-37)/37*

Relative Column Group Difference percentage of Sales Amount for State = Florida, Product = Bun i.e., -62.22 = Relative column difference percentage of Sales Amount for City = Orlando 100(34-90)/90*

Relative Column Group Difference percentage of Sales Amount for State = Arizona, Product = Soda i.e., 142.86 = Relative column difference percentage of Sales Amount for City = Scottsdale 100(68-28)/28*

Row Cumulative Sum—Row wise cumulative sum of all previous values for every column

		Arizona		Florida	
		Phoenix	Scottsdale	Miami	Orlando
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	34	25
	Bun	0	40	90	34
Cool Drinks	Cola	39	0	17	38
	Soda	28	68	56	34

ANALYSIS WITH THE "SUM" DATA

		Arizona		Florida	
		Phoenix	Scottsdale	Miami	Orlando
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	34	25
	Bun	37	57	124	59
Cool Drinks	Cola	76	57	141	97
	Soda	104	125	197	131

ANALYSIS WITH THE "ROW CUMULATIVE SUM" DATA OPERATION

Row Cumulative Sum of Sales Amount for
State = Arizona
City = Phoenix
Product = Bread is 37{37+0}
Product = Bun is 37{0+37}
Product = Cola is 76{39+37}
Product = Soda is 104{28+76}

Row Cumulative Sum of Sales Amount for
State = Florida
City = Miami
Product = Bread is 34{34+0}
Product = Bun is 124{90+34}
Product = Cola is 141{17+124}
Product = Soda is 197{56+141}

Row Group Cumulative Sum—Row wise cumulative sum of all previous values for every column (within the same group)

		Arizona		Florida	
		Phoenix	Scottsdale	Miami	Orlando
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	34	25
	Bun	0	40	90	34
Cool Drinks	Cola	39	0	17	38
	Soda	28	68	56	34

ANALYSIS WITH THE "SUM" DATA

		Arizona		Florida	
		Phoenix	Scottsdale	Miami	Orlando
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	34	25
	Bun	37	57	124	59
Cool Drinks	Cola	39	0	17	38
	Soda	67	68	73	72

ANALYSIS WITH THE "ROW GROUP CUMULATIVE SUM" DATA

Row Group cumulative sum of Sales Amount for
City = Phoenix
Product = Bun
i.e., 37 = Row cumulative sum of Sales Amount (37+0)

Row Group cumulative sum of Sales Amount for
City = Miami
Product = Soda
i.e., 72 = Row cumulative sum of Sales Amount (38+34)

Column Cumulative Sum—Column wise cumulative sum of all previous values for every row

		Arizona		Florida	
		Phoenix	Scottsdale	Miami	Orlando
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	34	25
	Bun	0	40	90	34
Cool Drinks	Cola	39	0	17	38
	Soda	28	68	56	34

ANALYSIS WITH THE "SUM" DATA

		Arizona		Florida	
		Phoenix	Scottsdale	Miami	Orlando
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	54	88	113
	Bun	0	40	130	164
Cool Drinks	Cola	39	39	56	94
	Soda	28	96	152	186

ANALYSIS WITH THE "COLUMN CUMULATIVE SUM" DATA

Column Cumulative Sum of Sales Amount for
State = Arizona and Florida
Product = Bread
City = Phoenix is 37{37+0}
City = Scottsdale is 54{17+37}
City = Miami is 88{34+54}
City = Orlando is 113{25+88}

Column Cumulative Sum of Sales Amount for
State = Arizona and Florida
Product = Cola
City = Phoenix is 39{39+0}
City = Scottsdale is 39{0+39}
City = Miami is 56{17+39}
City = Orlando is 94{38+56}

Column Group Cumulative Sum—Column wise cumulative sum of all previous values for every row (within the same group)

		Arizona			Florida		
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	**54**	34	25	**59**
	Bun	0	40	**40**	90	34	**124**
Cool Drinks	Cola	39	0	**39**	17	38	**55**
	Soda	28	68	**96**	56	34	**90**

ANALYSIS WITH THE "SUM" DATA

		Arizona			Florida		
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	54	**54**	34	59	**59**
	Bun	0	40	**40**	90	124	**124**
Cool Drinks	Cola	39	39	**39**	17	55	**55**
	Soda	28	96	**96**	56	90	**90**

ANALYSIS WITH THE "COLUMN GROUP CUMULATIVE SUM" DATA

Column Group cumulative sum of Sales Amount for
City = Scottsdale
Product = Bread
i.e., 54 = Column cumulative sum of Sales Amount for the group Arizona (37+17)

Column Group cumulative sum of Sales Amount for
City = Orlando
Product = Bun
i.e., 124 = Column cumulative sum of Sales Amount for the group Florida (90+34)

Distinct Count – Unique (Distinct) count value of specified dimension

Transaction id	State	City	Product Category	Product	Gross Sales
A1	Florida	Miami	Bakery	Bread	34
A2	Florida	Miami	Bakery	Bun	46
A3	Florida	Miami	Cool Drinks	Cola	17
A4	Florida	Miami	Cool Drinks	Soda	56
A5	Florida	Miami	Bakery	Bun	44
A6	Florida	Orlando	Bakery	Bread	25
A7	Florida	Orlando	Bakery	Bun	34
A8	Florida	Orlando	Cool Drinks	Cola	NULL
A9	Florida	Orlando	CoolDrinks	Soda	NULL
A10	Florida	Orlando	CoolDrinks	Cola	38
A11	Florida	Orlando	Cool Drinks	Soda	34
A12	Arizona	Phoenix	Bakery	Bread	12
A13	Arizona	Phoenix	Bakery	Bun	NULL
A14	Arizona	Phoenix	Cool Drinks	Cola	39
A15	Arizona	Phoenix	Cool Drinks	Soda	28
A16	Arizona	Phoenix	Bakery	Bread	25
A17	Arizona	Scottsdale	Bakery	Bread	17
A18	Arizona	Scottsdale	Bakery	Bun	6
A19	Arizona	Scottsdale	Cool Drinks	Cola	NULL
A20	Arizona	Scottsdale	Cool Drinks	Soda	68
A21	Arizona	Scottsdale	Bakery	Bun	34
A22	Arizona	Scottsdale	Bakery	Cookies	20

	Arizona	Florida
ProductCategory	GrossSales	GrossSales
Bakery	114	183
Cool Drinks	135	145

ANALYSIS WITH THE "SUM" DATA

.

	Arizona	Florida
ProductCategory	GrossSales	GrossSales
Bakery	3	2
Cool Drinks	2	2

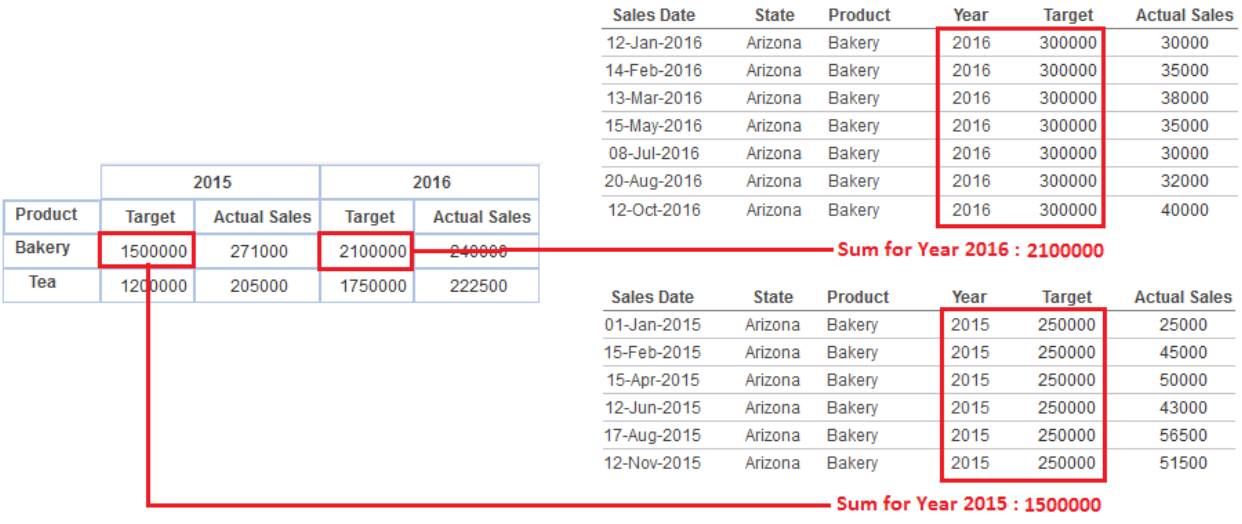
ANALYSIS WITH "DISTINCT COUNT" ON PRODUCT—PRODUCT CATEGORY-WISE – STATE-WISE

3 = Distinct Count of Product for
State = Arizona
Product Category =Bakery
(Bread, Bun, and Cookies are distinct Products)

2 = Distinct Count of Product for
State = Florida
Product Category = Cool Drinks
(Cola and Soda are two distinct Products)

Distinct Sum and Distinct Average –Sum and Average of Unique (Distinct) value of a specified dimension

Sales Date	Year	State	Product	Target (Yearly)	Actual Sales
1-Jan-2015	2015	Arizona	Bakery	250000	25000
10-Jan-2015	2015	Arizona	Tea	200000	30000
15-Feb-2015	2015	Arizona	Bakery	250000	45000
16-Feb-2015	2015	Arizona	Tea	200000	22500
15-Apr-2015	2015	Arizona	Bakery	250000	50000
16-May-2015	2015	Arizona	Tea	200000	35000
12-Jun-2015	2015	Arizona	Bakery	250000	43000
11-Jul-2015	2015	Arizona	Tea	200000	35500
17-Aug-2015	2015	Arizona	Bakery	250000	56500
9-Sep-2015	2015	Arizona	Tea	200000	42000
12-Nov-2015	2015	Arizona	Bakery	250000	51500
15-Dec-2015	2015	Arizona	Tea	200000	40000
12-Jan-2016	2016	Arizona	Bakery	300000	30000
15-Jan-2016	2016	Arizona	Tea	250000	35000
14-Feb-2016	2016	Arizona	Bakery	300000	35000
20-Feb-2016	2016	Arizona	Tea	250000	29000
13-Mar-2016	2016	Arizona	Bakery	300000	38000
5-Apr-2016	2016	Arizona	Tea	250000	29500
15-May-2016	2016	Arizona	Bakery	300000	35000
15-Jun-2016	2016	Arizona	Tea	250000	32000
8-Jul-2016	2016	Arizona	Bakery	300000	30000
16-Jul-2016	2016	Arizona	Tea	250000	30000
20-Aug-2016	2016	Arizona	Bakery	300000	32000
5-Sep-2016	2016	Arizona	Tea	250000	35000
12-Oct-2016	2016	Arizona	Bakery	300000	40000
15-Nov-2016	2016	Arizona	Tea	250000	32000



ANALYSIS WITH THE "SUM" DATA

Product	2015		2016	
	Target	Actual Sales	Target	Actual Sales
Bakery	250000	271000	300000	240000
Tea	200000	205000	250000	222500

Sales Date	State	Product	Year	Target	Actual Sales
12-Jan-2016	Arizona	Bakery	2016	300000	30000
14-Feb-2016	Arizona	Bakery	2016	300000	35000
13-Mar-2016	Arizona	Bakery	2016	300000	38000
15-May-2016	Arizona	Bakery	2016	300000	35000
08-Jul-2016	Arizona	Bakery	2016	300000	30000
20-Aug-2016	Arizona	Bakery	2016	300000	32000
12-Oct-2016	Arizona	Bakery	2016	300000	40000

Distinct Sum for Year 2016 : 300000

Sales Date	State	Product	Year	Target	Actual Sales
01-Jan-2015	Arizona	Bakery	2015	250000	25000
15-Feb-2015	Arizona	Bakery	2015	250000	45000
15-Apr-2015	Arizona	Bakery	2015	250000	50000
12-Jun-2015	Arizona	Bakery	2015	250000	43000
17-Aug-2015	Arizona	Bakery	2015	250000	56500
12-Nov-2015	Arizona	Bakery	2015	250000	51500

Distinct Sum for Year 2015 : 250000

ANALYSIS WITH "DISTINCT SUM" OF "TARGET" FOR DISTINCT COLUMN "YEAR"

Product	Target	Actual Sales
Bakery	275000	511000
Tea	225000	427500

Sales Date	State	Product	Year	Target	Actual Sales
01-Jan-2015	Arizona	Bakery	2015	250000	25000
15-Feb-2015	Arizona	Bakery	2015	250000	45000
15-Apr-2015	Arizona	Bakery	2015	250000	50000
12-Jun-2015	Arizona	Bakery	2015	250000	43000
17-Aug-2015	Arizona	Bakery	2015	250000	56500
12-Nov-2015	Arizona	Bakery	2015	250000	51500
12-Jan-2016	Arizona	Bakery	2016	300000	30000
14-Feb-2016	Arizona	Bakery	2016	300000	35000
13-Mar-2016	Arizona	Bakery	2016	300000	38000
15-May-2016	Arizona	Bakery	2016	300000	35000
08-Jul-2016	Arizona	Bakery	2016	300000	30000
20-Aug-2016	Arizona	Bakery	2016	300000	32000
12-Oct-2016	Arizona	Bakery	2016	300000	40000

Distinct Average for Year : (250000 + 300000) / 2 = 275000

ANALYSIS WITH "DISTINCT AVERAGE" OF "TARGET" FOR DISTINCT COLUMN "YEAR"

275000= Distinct Average of "Year" for

Product = Bakery

(250000 (2015) + 300000 (2016) / 2 = 275000 is Distinct Average)

Most Recent and Least Recent Functions

Applying this function returns the result of different aggregation methods on the most recent or least recent records from the data, based on the date dimension.

For analysis of data that is recorded over a period of time, aggregation-related data operations of values may not be important, but most recent and least recent values can be important.
Certain data values change frequently and recorded as closing balance or aggregated balance to date. This closing balance or aggregated values need to be analysed from time to time, as these changed values can affect the trends over a particular time span. So it becomes important to know the most recent or least recent data values to measure change in the trends.

Consider a scenario where year-to-date Gross Sales is recorded at the end of every month, which represents total sales from beginning of the year till end of that particular month.
For example, if year is beginning from January, then sales recorded in February is sales from January to February, and sales recorded in June is the sales from January to June.

State	January	February	March	April	May	June	July
	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Arizona	12500	14250	15000	16220	17150	18250	19700
Arkansas	15750	17000	18200	19100	20400	21250	22000
Florida	22000	23150	24250	25050	26000	27150	28300
Ohio	16050	17750	18400	19350	20400	21500	22620
Washington	24250	25500	26250	27000	28150	29240	31310

YTD SALES DATA—MONTHLY VIEW

On applying **Quarterly** view on this data, the YTD Most Recent **Gross Sales** for **March**, **June,** and **July** would be shown as below.

State	MostRecentValue(Quarter1)	MostRecentValue(Quarter2)	MostRecentValue(Quarter3)
	GrossSales	GrossSales	GrossSales
Arizona	15000	18250	19700
Arkansas	18200	21250	22000
Florida	24250	27150	28300
Ohio	18400	21500	22620
Washington	26250	29240	31310

MOST RECENT VALUES—QUARTERLY VIEW

Similarly, on applying **Quarterly** view on this data, the YTD Least Recent **Gross Sales** for **March**, **June,** and **July** would be shown as below.

State	LeastRecentValue(Quarter1)	LeastRecentValue(Quarter2)	LeastRecentValue(Quarter3)
	GrossSales	GrossSales	GrossSales
Arizona	12500	16220	19700
Arkansas	15750	19100	22000
Florida	22000	25050	28300
Ohio	16050	19350	22620
Washington	24250	27000	31310

LEAST RECENT VALUES—QUARTERLY VIEW

Example data set for all examples in this section:

Transaction ID	State	City	Product Category	Product	Date (MM/DD/YYYY)	Gross Sales
A1	Arizona	Phoenix	Bakery	Bread	1/2/2014	12
A2	Arizona	Phoenix	Bakery	Bread	1/23/2014	25
A3	Arizona	Phoenix	Bakery	Bun	1/4/2014	0
A4	Arizona	Phoenix	Cool Drinks	Cola	1/8/2014	39
A5	Arizona	Phoenix	Cool Drinks	Soda	1/28/2014	55
A6	Arizona	Phoenix	Cool Drinks	Soda	1/3/2014	28
A7	Arizona	Scottsdale	Bakery	Bread	1/19/2014	17
A8	Arizona	Scottsdale	Bakery	Bread	1/23/2014	20
A9	Arizona	Scottsdale	Bakery	Bun	1/12/2014	6
A10	Arizona	Scottsdale	Bakery	Bread	1/2/2014	35
A11	Arizona	Scottsdale	Bakery	Bun	1/13/2014	34
A12	Arizona	Scottsdale	Cool Drinks	Cola	1/2/2014	45
A13	Arizona	Scottsdale	Cool Drinks	Cola	1/17/2014	0
A14	Arizona	Scottsdale	Cool Drinks	Soda	1/28/2014	68
A15	Florida	Miami	Bakery	Bread	1/6/2014	34

A16	Florida	Miami	Bakery	Bread	1/2/2014	20
A17	Florida	Miami	Bakery	Bread	1/23/2014	30
A18	Florida	Miami	Bakery	Bun	1/13/2014	45
A19	Florida	Miami	Bakery	Bun	1/17/2014	44
A20	Florida	Miami	Cool Drinks	Cola	1/2/2014	40
A21	Florida	Miami	Cool Drinks	Soda	1/28/2014	50
A22	Florida	Miami	Cool Drinks	Cola	1/11/2014	17
A23	Florida	Miami	Cool Drinks	Soda	1/25/2014	56
A24	Florida	Orlando	Bakery	Bread	1/7/2014	25
A25	Florida	Orlando	Bakery	Bread	1/2/2014	15
A26	Florida	Orlando	Bakery	Bread	1/23/2014	18
A27	Florida	Orlando	Bakery	Bun	1/18/2014	34
A28	Florida	Orlando	CoolDrinks	Cola	1/2/2014	0
A29	Florida	Orlando	Cool Drinks	Cola	1/19/2014	38
A30	Florida	Orlando	Cool Drinks	Soda	1/28/2014	60
A31	Florida	Orlando	Cool Drinks	Soda	1/15/2014	34
A32	Florida	Orlando	Cool Drinks	Soda	1/21/2014	0

ORIGINAL ANALYSIS WITH THE "SUM" DATA AT THE ROW LEVEL

Most Recent values:

In the following examples, the displayed values are the Most Recent Values for the month of January.

	Arizona	Florida
ProductCategory	GrossSales	GrossSales
Bakery	20	18
CoolDrinks	68	60

ANALYSIS WITH THE "MOST RECENT" FUNCTION ON THE DATA AT THE LEVEL OF "PRODUCTCATEGORY" AND "STATE" DIMENSIONS

20 = The Most Recent value for
State = Arizona
Product Category = Bakery
Date = 23-Jan {A8}

60 = The Most Recent value for
State = Florida
Product Category = Cool Drinks
Date = 28-Jan {A30}

	Arizona		Florida	
	Phoenix	**Scottdale**	**Miami**	**Orlando**
ProductCategory	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	25	20	30	18
CoolDrinks	55	68	50	60

ANALYSIS WITH THE "MOST RECENT" FUNCTION ON THE DATA AT THE LEVEL OF "PRODUCTCATEGORY" AND "CITY" DIMENSIONS

25 = The Most Recent value for
State = Arizona
City = Phoenix
Product Category = Bakery
Date = 23-Jan {A2}

20 = The Most Recent value for
State = Arizona
City = Scottsdale
Product Category = Bakery
Date = -23-Jan {A8}

60 = The Most Recent value for State = Florida
City = Orlando
Product Category = Cool Drinks
Date = -28-Jan {A30}

		Arizona		Florida	
		Phoenix	**Scottdale**	**Miami**	**Orlando**
ProductCategory	**Product**	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	25	20	30	18
	Bun	0	34	44	34
CoolDrinks	Cola	39	0	17	38
	Soda	55	68	50	60

ANALYSIS WITH THE "MOST RECENT" FUNCTION ON THE DATA AT THE LEVEL OF "PRODUCT" AND "CITY" DIMENSIONS

Least Recent values:

In the following example, the displayed values are the **Least Recent Values** for the month of January.

	Arizona	Florida
ProductCategory	GrossSales	GrossSales
Bakery	12	20
CoolDrinks	45	40

ANALYSIS WITH THE "LEAST RECENT" FUNCTION ON THE DATA AT THE LEVEL OF "PRODUCTCATEGORY" AND "STATE" DIMENSIONS

12 = The Least Recent value (the earliest entry) for State = Arizona
Product Category = Bakery
Date = -2-Jan {A1}

45= The Least Recent value for
State = Arizona
Product Category = Cool Drinks
Date = -2-Jan {A12}

	Arizona		Florida	
	Phoenix	**Scottdale**	**Miami**	**Orlando**
ProductCategory	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	12	35	20	15
CoolDrinks	28	45	40	0

ANALYSIS WITH THE "LEAST RECENT" FUNCTION ON THE DATA AT THE LEVEL OF "PRODUCTCATEGORY" AND "CITY" DIMENSIONS

12 = The Least Recent value for
State = Arizona
City = Phoenix
Product Category = Bakery
Date = 2-Jan {A1}

35 = The Most Recent value for
State = Arizona
City = Scottsdale
Product Category = Bakery
Date = 2-Jan {A10}

40 = The Most Recent value for
State = Florida
City = Miami
Product Category = Cool Drinks
Date = 2-Jan {A20}

		Arizona		Florida	
		Phoenix	**Scottdale**	**Miami**	**Orlando**
ProductCategory	**Product**	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	12	35	20	15
	Bun	0	6	45	34
CoolDrinks	Cola	39	45	40	0
	Soda	28	68	56	34

ANALYSIS WITH THE "LEAST RECENT" FUNCTION ON THE DATA AT THE LEVEL OF "PRODUCT" AND "CITY" DIMENSIONS

12 = The Least Recent value for
State = Arizona
City = Phoenix
Product Category = Bakery Product = Bread
Date = 2-Jan {A1}

35 = The Least Recent value for
State = Arizona
City = Scottsdale
Product Category = Bakery
Product = Bread
Date = 2-Jan {A10}

6 = The Most Recent value for
State = Arizona
City = Scottsdale
Product Category = Bakery
Product = Bun
Date = 12-Jan {A9}

MOST RECENT AND LEAST RECENT WITH AGGREGATION FUNCTIONS:

If multiple records are found on the least recent or most recent date dimension, user can apply aggregation function on these records. For example, if January 28 is the most recent date available for January and there are 5 transactions available on this date, user can choose aggregation function to be applied on these records. If no aggregation function is selected, system will show the least or most recent record that is retrieved from the data.

Different Aggregated functions used on Most Recent and Least Recent Operations are as follows:

Aggregated Function	Description
None	In case of Least Recent, this shows the first transaction for the recent date. In case of Most Recent, this shows the last transaction for the recent date.
Sum	Sum of all the least or most recent date transactions
Average	Average of all the least or most recent date transactions
Minimum	Minimum value of the least or most recent date transactions
Maximum	Maximum value of the least or most recent date transactions
Effective Average	Average (excluding NULL transactions) of the least or most recent date transactions
Count	Total number of least or most recent date transactions
Effective Count	Count of Total number of least or most recent date transactions (excluding NULL transactions)

Most Recent: With Aggregation Functions

	Arizona	Florida
ProductCategory	GrossSales	GrossSales
Bakery	22.5	24
CoolDrinks	61.5	55

ANALYSIS WITH THE "MOST RECENT" FUNCTION ON THE DATA USING "AVERAGE" AGGREGATION METHOD AT THE LEVEL OF "PRODUCTCATEGORY" AND "STATE" DIMENSIONS

22.5 = Most Recent value using Average Aggregation Function for
State = Arizona;
Product Category = Bakery
Date =23-Jan {(A2+A8)/2}

55=Most Recent value using Average Aggregation Function for
State = Florida
Product Category = Cool Drinks
Date =28-Jan {(A21+A30)/2}

LEAST RECENT: WITH AGGREGATION FUNCTIONS

	Arizona	Florida
ProductCategory	GrossSales	GrossSales
Bakery	23.5	17.5
CoolDrinks	45	20

ANALYSIS WITH THE "LEAST RECENT" FUNCTION ON THE DATA USING "AVERAGE" AGGREGATION METHOD AT THE LEVEL OF "PRODUCTCATEGORY" AND "STATE" DIMENSIONS

23.5 = Least Recent value using Average Aggregation Function for
State = Arizona
Product Category = Bakery
Date = 2-Jan {(A1+A10)/2}

20 = Least Recent value using Average Aggregation Function for
State = Florida
Product Category = Cool Drinks
Date = 2-Jan {(A20+A28)/2}

Post Aggregation on Most Recent and Least Recent:
After applying Most Recent and Least Recent operation, user can also apply post aggregation data operation. Post aggregation data operation will be applied on values derived after applying most recent or least recent function at the front-end object level data.

The following post aggregation data operations can be applied:

Post Aggregation Function	Description
None	No effect
Row Percentage	Percentage value against row level summary within the same row (after applying most recent or least recent operation)
Row Group Percentage	Percentage value against the row group level summary (within the same group)(after applying most recent or least recent operation)
Column Percentage	Percentage value against column level summary within the same column (after applying most recent or least recent operation)
Column Group Percentage	Percentage value against the column group level summary (within the same group) (after applying most recent or least recent operation)
Total Percentage	Percentage value against the total crosstab sum (after applying most recent or least recent operation)
Relative Row Group Difference	Difference with respect to the previous row value (within same group) (after applying most recent or least recent operation)
Relative Row Group Difference Percentage	Difference with respect to the previous row value (within same group) in percentage (after applying most recent or least recent operation)
Relative Column Group Difference	Difference with respect to the previous column value (within the same group) (after applying most recent or least recent operation)
Relative Column Group Difference Percentage	Difference with respect to the previous column value (within the same group) in percentage (after applying most recent or least recent operation)
Row Group Cumulative Sum	Row wise cumulative sum of all previous values for every column (within the same group) (after applying most recent or least recent operation)
Column Group Cumulative Sum	Column wise cumulative sum of all previous values for every row (within the same group) (after applying most recent or least recent operation)

MOST RECENT: WITH POST AGGREGATION FUNCTION

	Arizona	Florida
ProductCategory	GrossSales	GrossSales
Bakery	22.5	24
CoolDrinks	61.5	55

ANALYSIS WITH THE "MOST RECENT" FUNCTION ON THE DATA USING "AVERAGE" AGGREGATION METHOD AT THE LEVEL OF "PRODUCTCATEGORY" AND "STATE" DIMENSIONS

	Arizona	Florida
ProductCategory	GrossSales	GrossSales
Bakery	48.39	51.61
CoolDrinks	52.79	47.21

ANALYSIS WITH THE "MOST RECENT" FUNCTION ON THE DATA USING "AVERAGE" AGGREGATION METHOD AT THE LEVEL OF "PRODUCTCATEGORY" AND "STATE" DIMENSIONS WITH "ROW PERCENTAGE" POST AGGREGATION FUNCTION

48.39 = Most Recent Function using Row Percentage Post Aggregation Function on Average Aggregation Function
For State = Arizona
Product Category = Bakery
*{(22.5/ (22.5+24))*100}*

47.21 = Most Recent Function using Row Percentage Post Aggregation Function on Average Aggregation Function
For State = Florida
Product Category = Cool Drinks
*{(55/ (61.5+55))*100}*

LEAST RECENT: WITH POST AGGREGATION FUNCTION

	Arizona	Florida
ProductCategory	GrossSales	GrossSales
Bakery	23.5	17.5
CoolDrinks	45	20

ANALYSIS WITH THE "LEAST RECENT" FUNCTION ON THE DATA USING "AVERAGE" AGGREGATION METHOD AT THE LEVEL OF "PRODUCTCATEGORY" AND "STATE" DIMENSIONS

	Arizona	Florida
ProductCategory	GrossSales	GrossSales
Bakery	57.32	42.68
CoolDrinks	69.23	30.77

ANALYSIS WITH THE "LEAST RECENT" FUNCTION ON THE DATA USING "AVERAGE" AGGREGATION METHOD AT THE LEVEL OF "PRODUCTCATEGORY" AND "STATE" DIMENSIONS WITH "ROW PERCENTAGE" POST AGGREGATION FUNCTION

57.32 = Least Recent Function using Row Percentage Post Aggregation Function on Average Aggregation Function
For State = Arizona
Product Category = Bakery
{(23.5/ (23.5+17.5))/100}

30.77 = Least Recent Function using Row Percentage Post Aggregation Function on Average Aggregation Function
For State = Florida
Product Category = Cool Drinks
{(20/ (45+20))/100}

4.12 Summary Operations

Multilevel Summary Operations

The user can define various summary operations at row and column level.

Example data set for all examples in this section:

Transaction id	State	City	Product Category	Product	Gross Sales
A1	Florida	Miami	Bakery	Bread	34
A2	Florida	Miami	Bakery	Bun	46
A3	Florida	Miami	Cool Drinks	Cola	17
A4	Florida	Miami	Cool Drinks	Soda	56
A5	Florida	Miami	Bakery	Bun	44
A6	Florida	Orlando	Bakery	Bread	25
A7	Florida	Orlando	Bakery	Bun	34
A8	Florida	Orlando	Cool Drinks	Cola	<NULL>
A9	Florida	Orlando	CoolDrinks	Soda	<NULL>
A10	Florida	Orlando	CoolDrinks	Cola	38
A11	Florida	Orlando	Cool Drinks	Soda	34
A12	Arizona	Phoenix	Bakery	Bread	12
A13	Arizona	Phoenix	Bakery	Bun	<NULL>
A14	Arizona	Phoenix	Cool Drinks	Cola	39
A15	Arizona	Phoenix	Cool Drinks	Soda	28
A16	Arizona	Phoenix	Bakery	Bread	25
A17	Arizona	Scottsdale	Bakery	Bread	17
A18	Arizona	Scottsdale	Bakery	Bun	6
A19	Arizona	Scottsdale	Cool Drinks	Cola	<NULL>
A20	Arizona	Scottsdale	Cool Drinks	Soda	68
A21	Arizona	Scottsdale	Bakery	Bun	34

Various summaries that can be added up to the nth level in the analysis.

	Arizona	Florida	Sum
	Gross Sales	Gross Sales	Gross Sales
Bakery	94	183	277
Cool Drinks	135	145	280
Sum	229	328	557

SUMMARY AT ROW AND COLUMN LEVEL

		Arizona	Florida	Sum
		Gross Sales	Gross Sales	Gross Sales
Bakery	Bread	54	59	113
	Bun	40	124	164
	Sum	94	183	277
Cool Drinks	Cola	39	55	94
	Soda	96	90	186
	Sum	135	145	280
Sum		229	328	557

MULTILEVEL ROW SUMMARIES

		Arizona			Florida			Sum
		Phoenix	Scottsdale	Sum	Miami	Orlando	Sum	
		Gross Sales	Gross Sales	Gross Sales	Gross Sales	Gross Sales	Gross Sales	Gross Sales
Bakery	Bread	37	17	54	34	25	59	113
	Bun	0	40	40	90	34	124	164
	Sum	37	57	94	124	59	183	277
Cool Drinks	Cola	39	0	39	17	38	55	94
	Soda	28	68	96	56	34	90	186
	Sum	67	68	135	73	72	145	280
Sum		104	125	229	197	131	328	557

MULTILEVEL COLUMN SUMMARIES

List of Summary Operations:

1.	Sum	
2.	Average	
3.	Effective Average	
4.	Count	
5.	Effective Count	
6.	Maximum	
7.	Minimum	
8.	First	
9.	Last	Applied on cube row level data.
10.	Default	
11.	Group Sum	
12.	Group Average	
13.	Group Count	
14.	Group Maximum	
15.	Group Minimum	
16.	Row Percentage	
17.	Row Group Percentage	
18.	Column Percentage	
19.	Column Group Percentage	
20.	Total Percentage	
21.	Relative Row Difference	
22.	Relative Row Difference Percentage	
23.	Relative Row Group Difference	
24.	Relative Row Group Difference Percentage	
25.	Relative Column Difference	
26.	Relative Column Difference Percentage	
27.	Relative Column Group Difference	
28.	Relative Column Group Difference Percentage	
29.	Row Cumulative Sum	
30.	Row Group Cumulative Sum	
31.	ColumnCumulative Sum	
32.	Column Group Cumulative Sum	Applied on the front-end object view data.

Row / Column Summary
Summary operations described below with examples.

		Arizona		Florida	
		Phoenix	Scottsdale	Miami	Orlando
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	34	25
	Bun	0	40	90	34
Cool Drinks	Cola	39	0	17	38
	Soda	28	68	56	34

ANALYSIS WITH THE "SUM" DATA OPERATION

DEFAULT—DEFAULT SUMMARY OPERATION WILL APPLY SUMMARY BASED ON DATA OPERATION APPLIED ON THE COLUMNS.

		Arizona		Florida	
		Phoenix	Scottsdale	Miami	Orlando
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	18.5	17	34	25
	Bun	0	20	45	34
Cool Drinks	Cola	39	0	17	19
	Soda	28	68	56	17

ANALYSIS WITH THE "AVERAGE" DATA OPERATION

		Arizona			Florida			Summary
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary	
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	18.5	17	**18**	34	25	**29.5**	**22.6**
	Bun	0	20	**13.33**	45	34	**41.33**	**27.33**
	Summary	**12.33**	**19**	**15.67**	**41.33**	**29.5**	**36.6**	**25.18**
Cool Drinks	Cola	39	0	**19.5**	17	19	**18.33**	**18.8**
	Soda	28	68	**48**	56	17	**30**	**37.2**
	Summary	**33.5**	**34**	**33.75**	**36.5**	**18**	**24.17**	**28**
Summary		**20.8**	**25**	**22.9**	**39.4**	**21.83**	**29.82**	**26.52**

ANALYSIS WITH THE "DEFAULT" SUMMARY OPERATION

Sum—Sum of all values

		Arizona			Florida			Summary
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary	
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	**54**	34	25	**59**	**113**
	Bun	0	40	**40**	90	34	**124**	**164**
	Summary	**37**	**57**	**94**	**124**	**59**	**183**	**277**
Cool Drinks	Cola	39	0	**39**	17	38	**55**	**94**
	Soda	28	68	**96**	56	34	**90**	**186**
	Summary	**67**	**68**	**135**	**73**	**72**	**145**	**280**
Summary		**104**	**125**	**229**	**197**	**131**	**328**	**557**

ANALYSIS WITH THE "SUM" SUMMARY OPERATION

Group Sum—Total/Sum of all values across a row or column at group level

		Arizona			Florida			Summary
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary	
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	25	17	25	34	25	34	34
	Bun	0	34	34	46	34	46	46
	Summary	25	34	34	46	34	46	46
Cool Drinks	Cola	39	0	39	17	38	38	39
	Soda	28	68	68	56	34	56	68
	Summary	39	68	68	56	38	56	68
Summary		39	68	68	56	38	56	68

ANALYSIS WITH THE "MAXIMUM" DATA OPERATION AND "DEFAULT" SUMMARY OPERATION

		Arizona			Florida			Summary
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary	
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	25	17	42	34	25	59	101
	Bun	0	34	34	46	34	80	114
	Summary	25	51	76	80	59	139	215
Cool Drinks	Cola	39	0	39	17	38	55	94
	Soda	28	68	96	56	34	90	186
	Summary	67	68	135	73	72	145	280
Summary		92	119	211	153	131	284	495

ANALYSIS WITH THE "MAXIMUM" DATA OPERATION AND "GROUP SUM" SUMMARY OPERATION

AVERAGE—AVERAGE OF ALL VALUES

		Arizona			Florida			Summary
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary	
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	**18**	34	25	**29.5**	**22.6**
	Bun	0	40	**13.33**	90	34	**41.33**	**27.33**
	Summary	**12.33**	**19**	**15.67**	**41.33**	**29.5**	**36.6**	**25.18**
Cool Drinks	Cola	39	0	**19.5**	17	38	**18.33**	**18.8**
	Soda	28	68	**48**	56	34	**30**	**37.2**
	Summary	**33.5**	**34**	**33.75**	**36.5**	**18**	**24.17**	**28**
Summary		**20.8**	**25**	**22.9**	**39.4**	**21.83**	**29.82**	**26.52**

ANALYSIS WITH THE "AVERAGE" SUMMARY OPERATION

EFFECTIVE AVERAGE—AVERAGE OF ALL "NOT NULL" VALUES

> Note:
> Effective Average implies the average of only the rows with "not null" values. Columns with value "null" are not considered in effective average calculation, but columns with value "0" are taken into consideration for effective average calculation.

		Arizona			Florida			Summary
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary	
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	**18**	34	25	**29.5**	**22.6**
	Bun	0	40	**20**	90	34	**41.33**	**32.8**
	Summary	**18.5**	**19**	**18.8**	**41.33**	**29.5**	**36.6**	**27.7**
Cool Drinks	Cola	39	0	**39**	17	38	**27.5**	**31.33**
	Soda	28	68	**48**	56	34	**45**	**46.5**
	Summary	**33.5**	**68**	**45**	**36.5**	**36**	**36.25**	**40**
Summary		**26**	**31.25**	**28.62**	**39.4**	**32.75**	**36.44**	**32.76**

ANALYSIS WITH THE "EFFECTIVE AVERAGE" SUMMARY OPERATION

Group Average—Average of all values within the same group

		Arizona			Florida			Summary
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary	
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	54	34	25	59	113
	Bun	0	40	40	90	34	124	164
	Summary	37	57	94	124	59	183	277
Cool Drinks	Cola	39	0	39	17	38	55	94
	Soda	28	68	96	56	34	90	186
	Summary	67	68	135	73	72	145	280
Summary		104	125	229	197	131	328	557

ANALYSIS WITH THE "SUM" SUMMARY OPERATION

		Arizona			Florida			Summary
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary	
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	27	34	25	29.5	56.5
	Bun	0	40	20	90	34	62	82
	Summary	18.5	28.5	23.5	62	29.5	45.75	69.25
Cool Drinks	Cola	39	0	19.5	17	38	27.5	47
	Soda	28	68	48	56	34	45	93
	Summary	33.5	34	33.75	36.5	36	36.25	70
Summary		52	62.5	57.25	98.5	65.5	82	139.25

ANALYSIS WITH THE "GROUP AVERAGE" SUMMARY OPERATION

Count—Count of all values

		Arizona			Florida			Summary
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary	
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	3	34	25	2	5
	Bun	0	40	3	90	34	3	6
	Summary	3	3	6	3	2	5	11
Cool Drinks	Cola	39	0	2	17	38	3	5
	Soda	28	68	2	56	34	3	5
	Summary	2	2	4	2	4	6	10
Summary		5	5	10	5	6	11	21

ANALYSIS WITH THE "COUNT" SUMMARY OPERATION

Effective Count—Count of all "not null" values

> Note:
> Effective Count implies the count of only the rows with "not null" values. Columns with value "null" are not considered in effective count calculation, but a column with value "0" is taken into consideration for effective count calculation.

		Arizona			Florida			Summary
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary	
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	3	34	25	2	5
	Bun	0	40	2	90	34	3	5
	Summary	2	3	5	3	2	5	10
Cool Drinks	Cola	39	0	1	17	38	2	3
	Soda	28	68	2	56	34	2	4
	Summary	2	1	3	2	2	4	7
Summary		4	4	8	5	4	9	17

ANALYSIS WITH THE "EFFECTIVE COUNT" SUMMARY OPERATION

Group Count—Count of all values within the same group

		Arizona			Florida			Summary
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary	
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	2	34	25	2	2
	Bun	0	40	2	90	34	2	2
	Summary	2	2	2	2	2	2	2
Cool Drinks	Cola	39	0	2	17	38	2	2
	Soda	28	68	2	56	34	2	2
	Summary	2	2	2	2	2	2	2
Summary		2	2	2	2	2	2	2

ANALYSIS WITH THE "GROUP COUNT" SUMMARY OPERATION

Maximum—Highest among all the values

		Arizona			Florida			Summary
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary	
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	25	34	25	34	34
	Bun	0	40	34	90	34	46	46
	Summary	25	34	34	46	34	46	46
Cool Drinks	Cola	39	0	39	17	38	38	39
	Soda	28	68	68	56	34	56	68
	Summary	39	68	68	56	38	56	68
Summary		39	68	68	56	38	56	68

ANALYSIS WITH THE "MAXIMUM" SUMMARY OPERATION

Group Maximum—Highest among all the values within the same group

		Arizona			Florida			Summary
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary	
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	**37**	34	25	**34**	**37**
	Bun	0	40	**40**	90	34	**90**	**90**
	Summary	**37**	**40**	**40**	**90**	**34**	**90**	**90**
Cool Drinks	Cola	39	0	**39**	17	38	**38**	**39**
	Soda	28	68	**68**	56	34	**56**	**68**
	Summary	**39**	**68**	**68**	**56**	**38**	**56**	**68**
Summary		**39**	**68**	**68**	**90**	**38**	**90**	**90**

ANALYSIS WITH THE "GROUP MAXIMUM" SUMMARY OPERATION

Minimum—Lowest among all the values

		Arizona			Florida			Summary
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary	
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	**12**	34	25	**25**	**12**
	Bun	0	40	**6**	90	34	**34**	**6**
	Summary	**12**	**6**	**6**	**34**	**25**	**25**	**6**
Cool Drinks	Cola	39	0	**39**	17	38	**17**	**17**
	Soda	28	68	**28**	56	34	**34**	**28**
	Summary	**28**	**68**	**28**	**17**	**34**	**17**	**17**
Summary		**12**	**6**	**6**	**17**	**25**	**17**	**6**

ANALYSIS WITH THE "MINIMUM" SUMMARY OPERATION

Group Minimum—Lowest among all the values within the same group

		Arizona			Florida			Summary
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary	
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	**17**	34	25	**25**	**17**
	Bun	0	40	**0**	90	34	**34**	**0**
	Summary	**0**	**17**	**0**	**34**	**25**	**25**	**0**
Cool Drinks	Cola	39	0	**0**	17	38	**17**	**0**
	Soda	28	68	**28**	56	34	**34**	**28**
	Summary	**28**	**0**	**0**	**17**	**34**	**17**	**0**
Summary		**0**	**0**	**0**	**17**	**25**	**17**	**0**

ANALYSIS WITH THE "GROUP MINIMUM" SUMMARY OPERATION

First—First among all the values

		Arizona			Florida			Summary
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary	
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	12	34	25	34	34
	Bun	0	40	6	90	34	46	46
	Summary	12	17	12	34	25	34	34
Cool Drinks	Cola	39	0	39	17	38	17	17
	Soda	28	68	28	56	34	56	56
	Summary	39	68	39	17	38	17	17
Summary		12	17	12	34	25	34	34

ANALYSIS WITH THE "FIRST" SUMMARY OPERATION

LAST—LAST AMONG ALL THE VALUES

		Arizona			Florida			Summary
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary	
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	17	34	25	25	17
	Bun	0	40	34	90	34	34	34
	Summary	25	34	34	44	34	34	34
Cool Drinks	Cola	39	0	0	17	38	38	0
	Soda	28	68	68	56	34	34	68
	Summary	28	68	68	56	34	34	68
Summary		25	34	34	44	34	34	34

ANALYSIS WITH THE "LAST" SUMMARY OPERATION

ROW PERCENTAGE—SUMMARY PERCENTAGE VALUE AGAINST ROW LEVEL SUMMARY

		Arizona			Florida			Summary
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary	
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	54	34	25	59	113
	Bun	0	40	40	90	34	124	164
	Summary	37	57	94	124	59	183	277
Cool Drinks	Cola	39	0	39	17	38	55	94
	Soda	28	68	96	56	34	90	186
	Summary	67	68	135	73	72	145	280
Summary		104	125	229	197	131	328	557

ANALYSIS WITH THE "SUM" DATA

		Arizona			Florida			Summary
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary	
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	47.79	34	25	52.21	100
	Bun	0	40	24.39	90	34	75.61	100
	Summary	13.36	20.58	33.94	44.77	21.3	66.06	100
Cool Drinks	Cola	39	0	41.49	17	38	58.51	100
	Soda	28	68	51.61	56	34	48.39	100
	Summary	23.93	24.29	48.21	26.07	25.71	51.79	100
Summary		18.67	22.44	41.11	35.37	23.52	58.89	100

ANALYSIS WITH THE "ROW PERCENTAGE" SUMMARY OPERATION

ROW GROUP PERCENTAGE—GROUP SUMMARY PERCENTAGE VALUE AGAINST ROW LEVEL SUMMARY (WITHIN SAME GROUP)

		Arizona			Florida			Summary
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary	
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	54	34	25	59	113
	Bun	0	40	40	90	34	124	164
	Summary	37	57	94	124	59	183	277
Cool Drinks	Cola	39	0	39	17	38	55	94
	Soda	28	68	96	56	34	90	186
	Summary	67	68	135	73	72	145	280
Summary		104	125	229	197	131	328	557

ANALYSIS WITH THE "SUM" DATA

		Arizona			Florida			Summary
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary	
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	57.45	34	25	32.24	40.79
	Bun	0	40	42.55	90	34	67.76	59.21
	Summary	35.58	45.6	41.05	62.94	45.04	55.79	49.73
Cool Drinks	Cola	39	0	28.89	17	38	37.93	33.57
	Soda	28	68	71.11	56	34	62.07	66.43
	Summary	64.42	54.4	58.95	37.06	54.96	44.21	50.27
Summary		100	100	100	100	100	100	100

ANALYSIS WITH THE "ROW GROUP PERCENTAGE" SUMMARY OPERATION

Total Percentage—Summary Percentage value against total crosstab sum

		Arizona			Florida			Summary
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary	
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	54	34	25	59	113
	Bun	0	40	40	90	34	124	164
	Summary	37	57	94	124	59	183	277
Cool Drinks	Cola	39	0	39	17	38	55	94
	Soda	28	68	96	56	34	90	186
	Summary	67	68	135	73	72	145	280
Summary		104	125	229	197	131	328	557

ANALYSIS WITH THE "SUM" DATA

		Arizona			Florida			Summary
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary	
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	9.69	34	25	10.59	20.29
	Bun	0	40	7.18	90	34	22.26	29.44
	Summary	6.64	10.23	16.88	22.26	10.59	32.85	49.73
Cool Drinks	Cola	39	0	7	17	38	9.87	16.88
	Soda	28	68	17.24	56	34	16.16	33.39
	Summary	12.03	12.21	24.24	13.11	12.93	26.03	50.27
Summary		18.67	22.44	41.11	35.37	23.52	58.89	100

ANALYSIS WITH THE "TOTAL PERCENTAGE" SUMMARY OPERATION

Relative Row Difference—Difference with respect to the previous summary row

State	ProductCategory	Product	GrossSales
Arizona	Bakery	Bread	54
		Bun	40
		Summary	94
	Cool Drinks	Cola	39
		Soda	96
		Summary	135
	Summary		229
Florida	Bakery	Bread	59
		Bun	124
		Summary	183
	Cool Drinks	Cola	55
		Soda	90
		Summary	145
	Summary		328
Summary			557

ANALYSIS WITH THE "SUM" DATA

State	ProductCategory	Product	GrossSales
Arizona	Bakery	Bread	54
		Bun	40
		Summary	**94**
	Cool Drinks	Cola	39
		Soda	96
		Summary	**41**
	Summary		**229**
Florida	Bakery	Bread	59
		Bun	124
		Summary	**48**
	Cool Drinks	Cola	55
		Soda	90
		Summary	**-38**
	Summary		**99**
Summary			**557**

ANALYSIS WITH THE "RELATIVE ROW DIFFERENCE" SUMMARY OPERATION

RELATIVE ROW DIFFERENCE PERCENTAGE—DIFFERENCE WITH RESPECT TO THE PREVIOUS SUMMARY ROW IN PERCENTAGE

State	ProductCategory	Product	GrossSales
Arizona	Bakery	Bread	54
		Bun	40
		Summary	**94**
	Cool Drinks	Cola	39
		Soda	96
		Summary	**135**
	Summary		**229**
Florida	Bakery	Bread	59
		Bun	124
		Summary	**183**
	Cool Drinks	Cola	55
		Soda	90
		Summary	**145**
	Summary		**328**
Summary			**557**

ANALYSIS WITH THE "SUM" DATA

State	ProductCategory	Product	GrossSales
Arizona	Bakery	Bread	54
		Bun	40
		Summary	**100**
	Cool Drinks	Cola	39
		Soda	96
		Summary	**43.62**
	Summary		**100**
Florida	Bakery	Bread	59
		Bun	124
		Summary	**35.56**
	Cool Drinks	Cola	55
		Soda	90
		Summary	**-20.77**
	Summary		**43.23**
Summary			**100**

ANALYSIS WITH THE "RELATIVE ROW DIFFERENCE PERCENTAGE" SUMMARY OPERATION

RELATIVE ROW GROUP DIFFERENCE—DIFFERENCE WITH RESPECT TO THE PREVIOUS SUMMARY ROW FOR RESPECTIVE GROUP

State	ProductCategory	Product	GrossSales
Arizona	Bakery	Bread	54
		Bun	40
		Summary	**94**
	Cool Drinks	Cola	39
		Soda	96
		Summary	**135**
	Summary		**229**
Florida	Bakery	Bread	59
		Bun	124
		Summary	**183**
	Cool Drinks	Cola	55
		Soda	90
		Summary	**145**
	Summary		**328**
Summary			**557**

ANALYSIS WITH THE "SUM" DATA

State	ProductCategory	Product	GrossSales
Arizona	Bakery	Bread	54
		Bun	40
		Summary	94
	Cool Drinks	Cola	39
		Soda	96
		Summary	41
	Summary		229
Florida	Bakery	Bread	59
		Bun	124
		Summary	183
	Cool Drinks	Cola	55
		Soda	90
		Summary	-38
	Summary		99
Summary			557

ANALYSIS WITH THE "RELATIVE ROW GROUP DIFFERENCE" SUMMARY OPERATION

RELATIVE ROW GROUP DIFFERENCE PERCENTAGE—DIFFERENCE WITH RESPECT TO THE PREVIOUS SUMMARY ROW IN PERCENTAGE FOR RESPECTIVE GROUP

State	ProductCategory	Product	GrossSales
Arizona	Bakery	Bread	54
		Bun	40
		Summary	94
	Cool Drinks	Cola	39
		Soda	96
		Summary	135
	Summary		229
Florida	Bakery	Bread	59
		Bun	124
		Summary	183
	Cool Drinks	Cola	55
		Soda	90
		Summary	145
	Summary		328
Summary			557

ANALYSIS WITH THE "SUM" DATA

State	ProductCategory	Product	GrossSales
Arizona	Bakery	Bread	54
		Bun	40
		Summary	100
	Cool Drinks	Cola	39
		Soda	96
		Summary	43.62
	Summary		100
Florida	Bakery	Bread	59
		Bun	124
		Summary	100
	Cool Drinks	Cola	55
		Soda	90
		Summary	-20.77
	Summary		43.23
Summary			100

ANALYSIS WITH THE "RELATIVE ROW GROUP DIFFERENCE PERCENTAGE" SUMMARY OPERATION

COLUMN PERCENTAGE—SUMMARY PERCENTAGE VALUE AGAINST COLUMN LEVEL SUMMARY WITHIN THE SAME COLUMN

		Arizona			Florida			Summary
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary	
ProductCategor	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	54	34	25	59	113
	Bun	0	40	40	90	34	124	164
	Summary	37	57	94	124	59	183	277
Cool Drinks	Cola	39	0	39	17	38	55	94
	Soda	28	68	96	56	34	90	186
	Summary	67	68	135	73	72	145	280
Summary		104	125	229	197	131	328	557

ANALYSIS WITH THE "SUM" DATA

		Arizona			Florida			Summary
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary	
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	23.58	34	25	17.99	20.29
	Bun	0	40	17.47	90	34	37.8	29.44
	Summary	35.58	45.6	41.05	62.94	45.04	55.79	49.73
Cool Drinks	Cola	39	0	17.03	17	38	16.77	16.88
	Soda	28	68	41.92	56	34	27.44	33.39
	Summary	64.42	54.4	58.95	37.06	54.96	44.21	50.27
Summary		100	100	100	100	100	100	100

ANALYSIS WITH THE "COLUMN PERCENTAGE" SUMMARY OPERATION

COLUMN GROUP PERCENTAGE—GROUP SUMMARY PERCENTAGE VALUE AGAINST COLUMN LEVEL SUMMARY (WITHIN SAME GROUP)

		Arizona			Florida			Summary
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary	
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	54	34	25	59	113
	Bun	0	40	40	90	34	124	164
	Summary	37	57	94	124	59	183	277
Cool Drinks	Cola	39	0	39	17	38	55	94
	Soda	28	68	96	56	34	90	186
	Summary	67	68	135	73	72	145	280
Summary		104	125	229	197	131	328	557

ANALYSIS WITH THE "SUM" DATA

		Arizona			Florida			Summary
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary	
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	47.79	34	25	52.21	100
	Bun	0	40	24.39	90	34	75.61	100
	Summary	39.36	60.64	33.94	67.76	32.24	66.06	100
Cool Drinks	Cola	39	0	41.49	17	38	58.51	100
	Soda	28	68	51.61	56	34	48.39	100
	Summary	49.63	50.37	48.21	50.34	49.66	51.79	100
Summary		45.41	54.59	41.11	60.06	39.94	58.89	100

ANALYSIS WITH THE "COLUMN GROUP PERCENTAGE" FUNCTION APPLIED ON SUMMARY OPERATION

Relative Column Difference—Difference with respect to the previous summary column

		Arizona			Florida			Summary
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary	
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	54	34	25	59	113
	Bun	0	40	40	90	34	124	164
	Summary	37	57	94	124	59	183	277
Cool Drinks	Cola	39	0	39	17	38	55	94
	Soda	28	68	96	56	34	90	186
	Summary	67	68	135	73	72	145	280
Summary		104	125	229	197	131	328	557

ANALYSIS WITH THE "SUM" DATA

		Arizona			Florida			Summary
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary	
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	54	34	25	5	113
	Bun	0	40	40	90	34	84	164
	Summary	37	20	94	67	-65	89	277
Cool Drinks	Cola	39	0	39	17	38	16	94
	Soda	28	68	96	56	34	-6	186
	Summary	67	1	135	5	-1	10	280
Summary		104	21	229	72	-66	99	557

ANALYSIS WITH THE "RELATIVE COLUMN DIFFERENCE" SUMMARY OPERATION

Relative Column Difference Percentage—Difference with respect to the previous summary column in percentage

		Arizona			Florida			Summary
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary	
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	54	34	25	59	113
	Bun	0	40	40	90	34	124	164
	Summary	37	57	94	124	59	183	277
Cool Drinks	Cola	39	0	39	17	38	55	94
	Soda	28	68	96	56	34	90	186
	Summary	67	68	135	73	72	145	280
Summary		104	125	229	197	131	328	557

ANALYSIS WITH THE "SUM" DATA

		Arizona			Florida			Summary
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary	
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	100	34	25	9.26	100
	Bun	0	40	100	90	34	210	100
	Summary	100	54.05	100	117.54	-52.42	94.68	100
Cool Drinks	Cola	39	0	100	17	38	41.03	100
	Soda	28	68	100	56	34	-6.25	100
	Summary	100	1.49	100	7.35	-1.37	7.41	100
Summary		100	20.19	100	57.6	-33.5	43.23	100

ANALYSIS WITH THE "RELATIVE COLUMN DIFFERENCE PERCENTAGE" SUMMARY OPERATION

RELATIVE COLUMN GROUP DIFFERENCE—DIFFERENCE WITH RESPECT TO THE PREVIOUS SUMMARY COLUMN FOR PARTICULAR GROUP

		Arizona			Florida			Summary
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary	
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	54	34	25	59	113
	Bun	0	40	40	90	34	124	164
	Summary	37	57	94	124	59	183	277
Cool Drinks	Cola	39	0	39	17	38	55	94
	Soda	28	68	96	56	34	90	186
	Summary	67	68	135	73	72	145	280
Summary		104	125	229	197	131	328	557

ANALYSIS WITH THE "SUM" DATA

		Arizona			Florida			Summary
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary	
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	54	34	25	5	113
	Bun	0	40	40	90	34	84	164
	Summary	37	20	94	124	-65	89	277
Cool Drinks	Cola	39	0	39	17	38	16	94
	Soda	28	68	96	56	34	-6	186
	Summary	67	1	135	73	-1	10	280
Summary		104	21	229	197	-66	99	557

ANALYSIS WITH THE "RELATIVE COLUMN GROUP DIFFERENCE" SUMMARY OPERATION

RELATIVE COLUMN GROUP DIFFERENCE PERCENTAGE—DIFFERENCE WITH RESPECT TO THE PREVIOUS SUMMARY COLUMN IN PERCENTAGE FOR PARTICULAR GROUP

		Arizona			Florida			Summary
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary	
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	54	34	25	59	113
	Bun	0	40	40	90	34	124	164
	Summary	37	57	94	124	59	183	277
Cool Drinks	Cola	39	0	39	17	38	55	94
	Soda	28	68	96	56	34	90	186
	Summary	67	68	135	73	72	145	280
Summary		104	125	229	197	131	328	557

ANALYSIS WITH THE "SUM" DATA

		Arizona			Florida			Summary
		Phoenix	Scottsdale	Summary	Miami	Orlando	Summary	
ProductCategory	Product	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	Bread	37	17	100	34	25	9.26	100
	Bun	0	40	100	90	34	210	100
	Summary	100	54.05	100	100	-52.42	94.68	100
Cool Drinks	Cola	39	0	100	17	38	41.03	100
	Soda	28	68	100	56	34	-6.25	100
	Summary	100	1.49	100	100	-1.37	7.41	100
Summary		100	20.19	100	100	-33.5	43.23	100

ANALYSIS WITH THE "RELATIVE COLUMN GROUP DIFFERENCE PERCENTAGE" SUMMARY OPERATION

ROW CUMULATIVE SUM—CUMULATIVE SUM OF ALL PREVIOUS ROW SUMMARIES IN THE SAME ROW

State	ProductCategory	Product	GrossSales
Arizona	Bakery	Bread	54
		Bun	40
		Summary	**94**
	Cool Drinks	Cola	39
		Soda	96
		Summary	**135**
	Summary		**229**
Florida	Bakery	Bread	59
		Bun	124
		Summary	**183**
	Cool Drinks	Cola	55
		Soda	90
		Summary	**145**
	Summary		**328**
Summary			**557**

ANALYSIS WITH THE "SUM" DATA

State	ProductCategory	Product	GrossSales
Arizona	Bakery	Bread	54
		Bun	40
		Summary	**94**
	Cool Drinks	Cola	39
		Soda	96
		Summary	**229**
	Summary		**229**
Florida	Bakery	Bread	59
		Bun	124
		Summary	**412**
	Cool Drinks	Cola	55
		Soda	90
		Summary	**557**
	Summary		**557**
Summary			**557**

ANALYSIS WITH THE "ROW CUMULATIVE SUM" SUMMARY OPERATION

ROW GROUP CUMULATIVE SUM—CUMULATIVE SUM OF ALL PREVIOUS ROW SUMMARIES IN THE SAME GROUP

State	ProductCategory	Product	GrossSales
Arizona	Bakery	Bread	54
		Bun	40
		Summary	**94**
	Cool Drinks	Cola	39
		Soda	96
		Summary	**135**
	Summary		**229**
Florida	Bakery	Bread	59
		Bun	124
		Summary	**183**
	Cool Drinks	Cola	55
		Soda	90
		Summary	**145**
	Summary		**328**
Summary			**557**

ANALYSIS WITH THE "SUM" DATA

State	ProductCategory	Product	GrossSales
Arizona	Bakery	Bread	54
		Bun	40
		Summary	**94**
	Cool Drinks	Cola	39
		Soda	96
		Summary	**229**
	Summary		**229**
Florida	Bakery	Bread	59
		Bun	124
		Summary	**183**
	Cool Drinks	Cola	55
		Soda	90
		Summary	**328**
	Summary		**557**
Summary			**557**

ANALYSIS WITH THE "ROW GROUP CUMULATIVE SUM" SUMMARY OPERATION

COLUMN CUMULATIVE SUM—CUMULATIVE SUM OF ALL PREVIOUS COLUMN SUMMARIES IN THE SAME COLUMN

	Arizona						Florida					
	Phoenix			Scottsdale			Miami			Orlando		
	Bakery	Cool Drinks	Summary	Bakery	Cool Drinks	Summary	Bakery	Cool Drinks	Summary	Bakery	Cool Drinks	Summary
Product	GrossSales	GrossSales	**GrossSales**	GrossSales	GrossSales	**GrossSales**	GrossSales	GrossSales	**GrossSales**	GrossSales	GrossSales	**GrossSales**
Bread	37		**37**	17		**17**	34		**34**	25		**25**
Bun	0		**0**	40		**40**	90		**90**	34		**34**
Cola		39	**39**		0	**0**		17	**17**		38	**38**
Soda		28	**28**		68	**68**		56	**56**		34	**34**

ANALYSIS WITH THE "SUM" DATA

	Arizona						Florida					
	Phoenix			Scottsdale			Miami			Orlando		
	Bakery	Cool Drinks	Summary	Bakery	Cool Drinks	Summary	Bakery	Cool Drinks	Summary	Bakery	Cool Drinks	Summary
Product	GrossSales	GrossSales	**GrossSales**	GrossSales	GrossSales	**GrossSales**	GrossSales	GrossSales	**GrossSales**	GrossSales	GrossSales	**GrossSales**
Bread	37		**37**	17		**54**	34		**88**	25		**113**
Bun	0		**0**	40		**40**	90		**130**	34		**164**
Cola		39	**39**		0	**39**		17	**56**		38	**94**
Soda		28	**28**		68	**96**		56	**152**		34	**186**

ANALYSIS WITH THE "COLUMN CUMULATIVE SUM" SUMMARY OPERATION

COLUMN GROUP CUMULATIVE SUM—CUMULATIVE SUM OF ALL PREVIOUS COLUMN SUMMARIES IN THE SAME COLUMN

	Arizona						Florida					
	Phoenix			Scottsdale			Miami			Orlando		
	Bakery	Cool Drinks	Summary	Bakery	Cool Drinks	Summary	Bakery	Cool Drinks	Summary	Bakery	Cool Drinks	Summary
Product	GrossSales	GrossSales	**GrossSales**	GrossSales	GrossSales	**GrossSales**	GrossSales	GrossSales	**GrossSales**	GrossSales	GrossSales	**GrossSales**
Bread	37		**37**	17		**17**	34		**34**	25		**25**
Bun	0		**0**	40		**40**	90		**90**	34		**34**
Cola		39	**39**		0	**0**		17	**17**		38	**38**
Soda		28	**28**		68	**68**		56	**56**		34	**34**

ANALYSIS WITH THE "SUM" DATA

	Arizona						Florida					
	Phoenix			Scottsdale			Miami			Orlando		
	Bakery	Cool Drinks	Summary	Bakery	Cool Drinks	Summary	Bakery	Cool Drinks	Summary	Bakery	Cool Drinks	Summary
Product	GrossSales	GrossSales	**GrossSales**	GrossSales	GrossSales	**GrossSales**	GrossSales	GrossSales	**GrossSales**	GrossSales	GrossSales	**GrossSales**
Bread	37		**37**	17		**54**	34		**34**	25		**59**
Bun	0		**0**	40		**40**	90		**90**	34		**124**
Cola		39	**39**		0	**39**		17	**17**		38	**55**
Soda		28	**28**		68	**96**		56	**56**		34	**90**

ANALYSIS WITH THE "COLUMN GROUP CUMULATIVE SUM" SUMMARY OPERATION

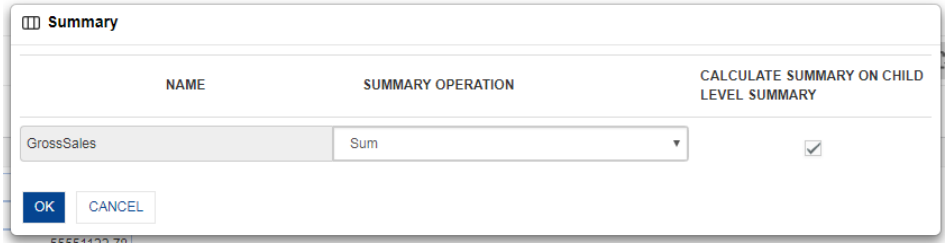
OPTION TO CALCULATE SUMMARY USING CHILD LEVEL SUMMARY DATA

In earlier versions, users were able to apply a summary based on various back-end and front-end summary operations. In this version, Smarten provides a new option to calculate a summary based on a child level summary.

For example, there is one crosstab with 2 row dimensions: ProductCategory and ProductName and one measure GrossSales with sum data operation. On the ProductName column, a summary is applied with the count summary operation. Now, if the user wants to see the sum of count values of the Productname summary in the Productcategory summary, the user needs to select “calculate summary on child level summary” check box while applying the summary on the ProductCategory column. The system will calculate the summary using the Productname summary data.

ProductCategory	ProductName	GrossSales
Bakery	Cake	9072639.26
	Cookies	55551122.78
	Count	**962**
Cool Drinks	Cola	1693762.72
	Soda	1716771.13
	Count	**497**
Sum of Count		**1459**

CALCULATE SUMMARY USING CHILD SUMMARY

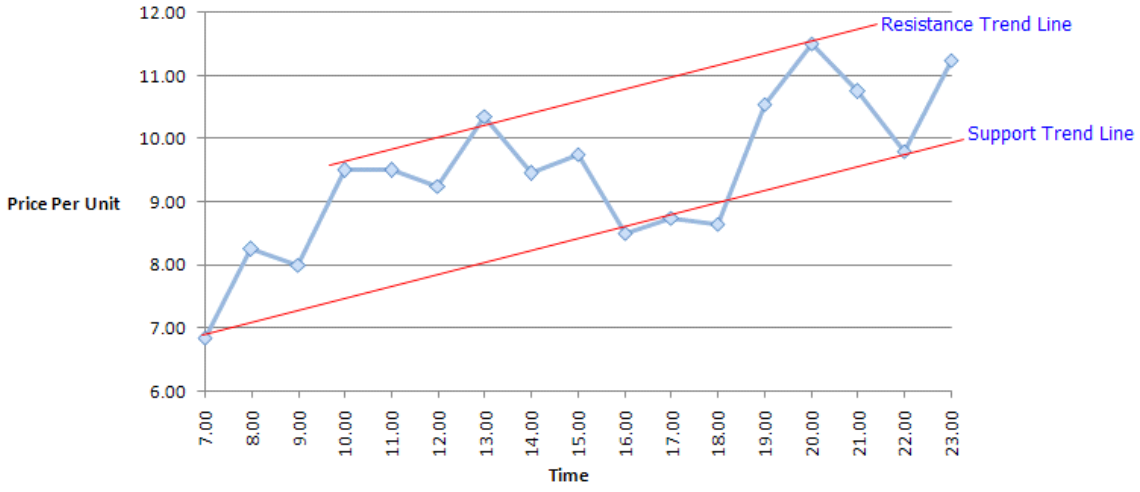


CALCULATE SUMMARY USING CHILD SUMMARY SETTING

4.13 Trend Line

Trend lines are used to determine trends and movement of trends, which is also called regression analysis.

A trend line is created when user draws a diagonal line between two or more points in a chart. When establishing trend lines, it is important to choose a chart based on time intervals. The time interval in the chart can be hourly, daily, weekly, monthly, quarterly, and yearly periods.



GRAPHIC ANALYSIS WITH TREND LINES

Types of trend lines

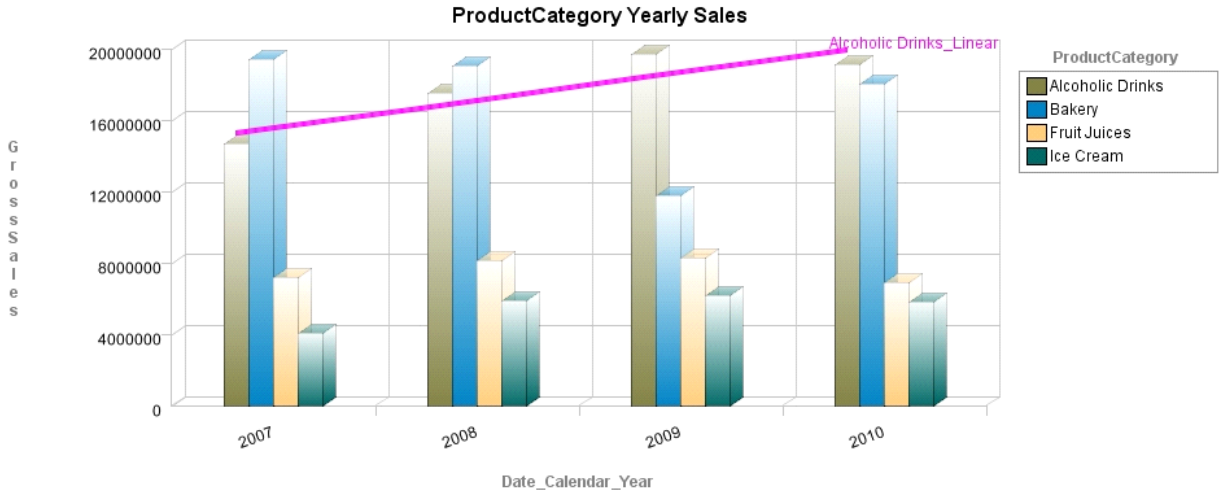
- Linear
- Logarithmic
- Polynomial
- Power
- Exponential
- Moving Average

> Note:
> Trend Lines cannot be applied to stacked and percentage graphs.

4.13.1 Linear Trend line

If data values are going up or down at a steady rate, a linear trend line is best to depict its nature. A linear trend line usually shows that something is increasing or decreasing at a steady rate.

Example: As per the increase in a state's population, consumption of products increases



EXAMPLE OF A LINEAR TREND LINE FOR ALCOHOLIC DRINKS

How does linear trend line compute?

A linear trend line is drawn by using the following equation to calculate the least squares fit for a line:

$$y = mx + b$$

where m is the slope and b is the intercept.

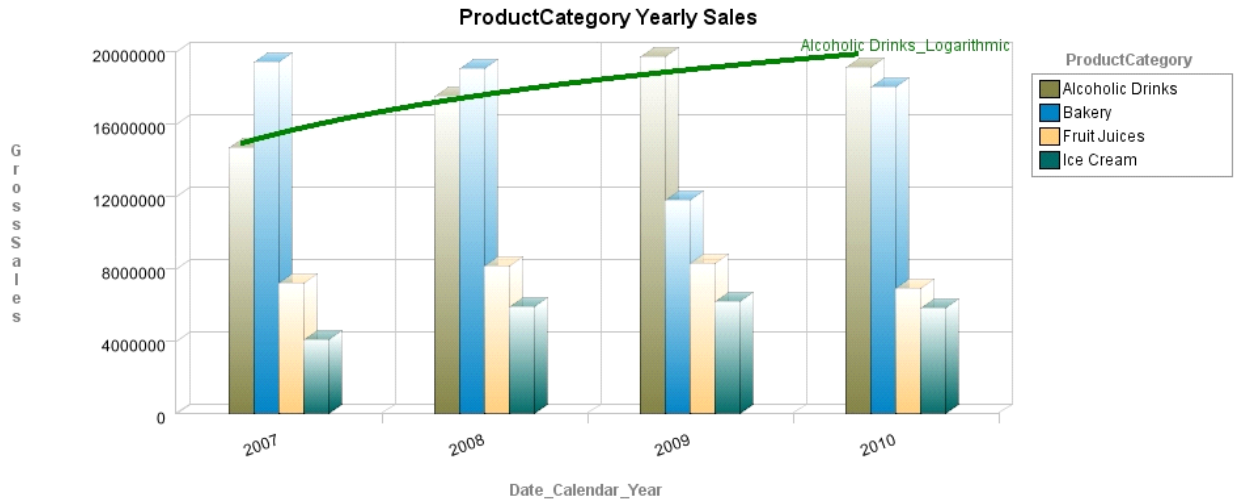
Guidelines:

- With linear data patterns, i.e., data points resembling a line either increasing or decreasing at a steady rate (simple linear data sets), a linear trend line should be applied
- A linear trend line will not adjust to any changes in the trend

4.13.2 Logarithmic Trend line

A logarithmic trend line is a best-fit curved line that is most useful when the rate of change in the data increases or decreases quickly and then levels out. A logarithmic trend line can use negative and/or positive values.

Example: Population growth in a region increases or decreases periodically



EXAMPLE OF A LOGARITHMIC TREND LINE FOR ALCOHOLIC DRINKS

How does logarithmic trend line compute?

A logarithmic trend line is drawn by using the following equation to calculate the least squares fit through points:

$$y = c\ln x + b$$

where c and b are constants, and ln is the natural logarithm function.

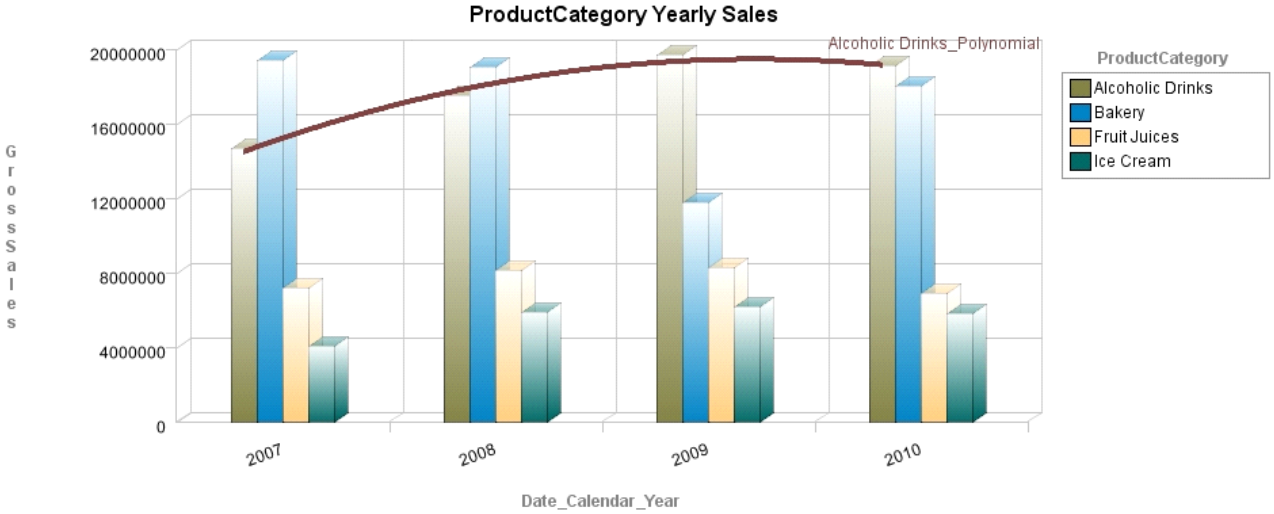
Guidelines:

- With data points having a relative sharp curve at one end and then gradually levels out, logarithmic trend line should be applied
- Logarithmic trend line tends to hide accelerating trends in the short/medium run. In accelerating trends where no steady growth rate is possible, the logarithmic trend line should not to be used

4.13.3 Polynomial Trend line

A polynomial trend line is a curved line that is used when data fluctuates in a rhythmic manner. It is useful, for example, for analyzing gains and losses over a large data set. The order of the polynomial can be determined by the number of fluctuations in the data or by how many bends (ups and downs) appear in the curve. An Order 2 polynomial trend line generally has only one up or down. Order 3 generally has one or two ups or downs, and similarly order 4 generally has up to three ups or downs and so on.

Example: Sale of seasonal fruits in the past 10 years



EXAMPLE OF A POLYNOMIAL TREND LINE FOR ALCOHOLIC DRINKS

How does polynomial trend line compute?

A polynomial or curvilinear trend line is drawn by using the following equation to calculate the least squares fit through points:

$$y = b + c_1x + c_2x^2 + c_3x^3 + \ldots + c_6x^6$$

where b and c1,c2,.....,c6 are constants.

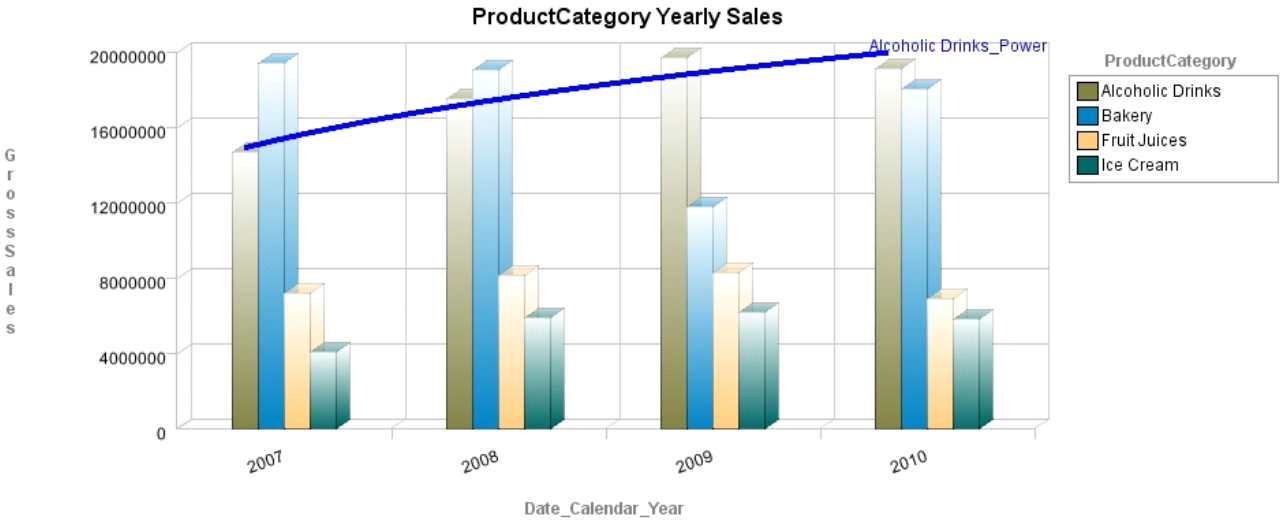
Guidelines:

- A polynomial trend line is a curved line that is used when data fluctuates. Based on the number of fluctuations, the order of a polynomial trend line can be decided
- A polynomial trend line fitted throughout a long data series will be incapable of revealing short-term market fluctuations, such as seasonal patterns

4.13.4 Power Trend line

If the graph data changes at a steadily increasing or decreasing rate as in an acceleration curve, a power trend line is a curved line that is best used. You cannot create a power trend line if your data contains zero or negative values.

Example: The acceleration of a racing car for the first 20-second interval



EXAMPLE OF A POWER TREND LINE FOR ALCOHOLIC DRINKS

How does power trend line compute?

A power trend line by using the following equation to calculate the least squares fit through points:

$$y = cx^b$$

where c and b are constants.

> Note:
> This option is not available when your data includes negative or zero values.

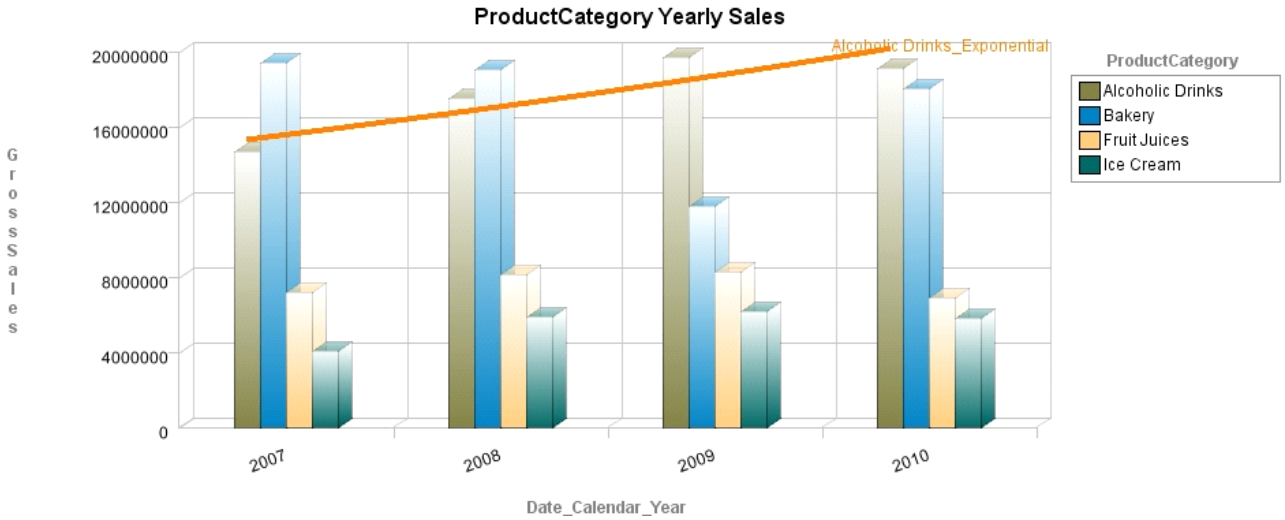
Guidelines:

- Very practical for acceleration or deceleration graph data
- Cannot be used when any of your data points are zero or negative

4.13.5 Exponential Trend line

An exponential trend line, which looks like a smoothly curving line, is most useful when graphed data values change at an ever increasing or decreasing rate. An exponential trend line cannot be created if data contains zero or negative values.

Example: Number of automobiles used in world for the last 5 years



EXAMPLE OF AN EXPONENTIAL TREND LINE FOR ALCOHOLIC DRINKS

How does exponential trend line compute?

An exponential trend line by using the following equation to calculate the least squares fit through points:

$$y = ce^{bx}$$

where c and b are constants, and e is the base of the natural logarithm.

Guidelines:

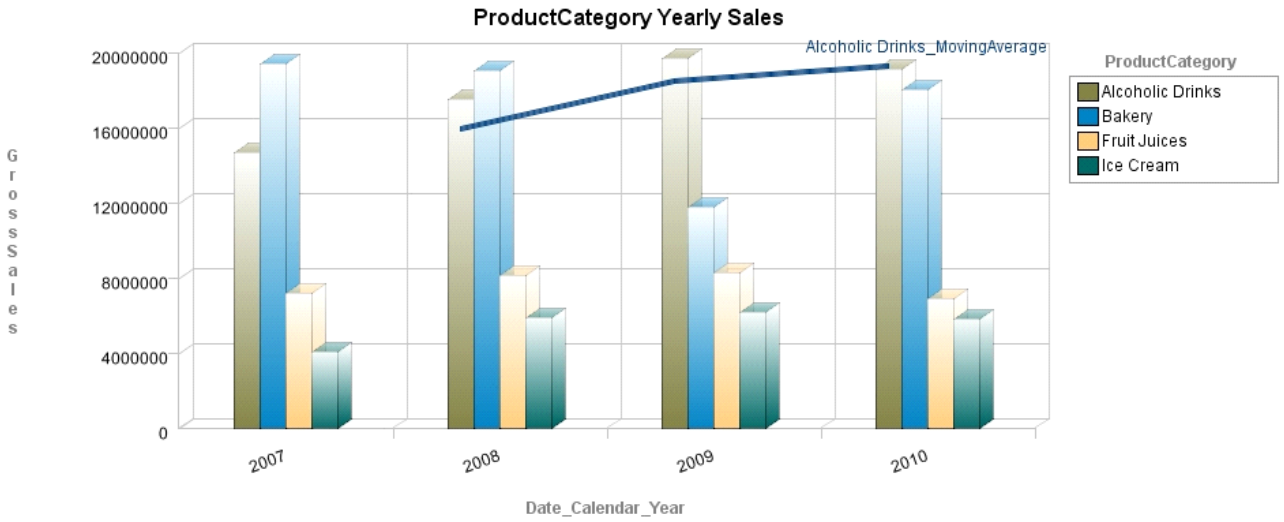
- Very practical for increase or decrease at accelerating or decelerating rates of graph data
- Cannot be used when any of your data points are zero or negative

4.13.6 Moving Average Trend line

A moving average trend line smoothes out fluctuations in the data to show a pattern or trend more clearly.

A moving average trend line uses a specific number of data points (set by the Period option), averages them, and uses the average value as a point in the trend line. If Period is set to 2, for example, then the average of the first two data points is used as the first point in the moving average trend line. The average of the second and third data points is used as the second point in the trend line and so on.

Example: Price movement of a currency over a specific time period



EXAMPLE OF A MOVING AVERAGE TREND LINE FOR ALCOHOLIC DRINKS

How does moving average trend line compute?
A moving average trend line by using the following equation:

where c and b are constants, and e is the base of the natural logarithm.

> Note:
> The number of points in a moving average (moving average: a sequence of averages computed from parts of a data series; in a chart, a moving average smooths out the fluctuations in data, thus showing the pattern or trend more clearly). The trend line equals the total number of points in the series divided by the number that you specify for the period.

Guidelines:

- It smooths out temporary high and low fluctuations in data as in trading systems.
- Area pattern can be a risk for the moving averages. The moving average values lie between high and low fluctuations.

> Note:
> User needs to specify the moving average period, which is a constant value.

4.14 Subview

Subview allows a drill-down view of the same or different set of columns from the higher level. One can view detailed information of dimensions or measures by associating them with another relevant crosstab or tabular.

Two different analyses can be related through **Subview** by providing the Join condition.

State	2013	2012	2011	Total
	GrossSales	GrossSales	GrossSales	GrossSales
Arizona	10893193	5846560	6823258	23563010
Arkansas	14797291	17237143	15216444	47250877
Florida	12190108	17608923	16450044	46249075
Ohio	6568674	6138330	9016896	21723900
Washington	16230358	9727103	12083066	38040526
Total	60679623	56558058	59589707	176827388

ANALYSIS (STATEWISE SALES)

City	2013			2012			2011			Total		
	GrossSales	Target	Variance	GrossSales	Target	Variance	GrossSales	Target	Variance	GrossSales	Target	Variance
Cleveland	3321000	3102934	107.03	2640610	2775293	95.15	4424113	5664968	78.1	10385722	11543195	89.97
Conway	8781164	6923147	126.84	13239982	15248860	86.83	12541071	18440391	68.01	34562216	40612398	85.1
Dayton	3247673	3550525	91.47	3497721	2968037	117.85	4592783	4927714	93.2	11338177	11446276	99.06
Lakeland	5660999	5008768	113.02	3484565	3132451	111.24	1898469	1962143	96.75	11044033	10103362	109.31
Orlando	6529109	4221062	154.68	14124357	15944914	88.58	14551574	18646146	78.04	35205041	38812122	90.71
Phoenix	3607761	3538099	101.97	2503602	5062397	49.45	4404635	3762620	117.06	10515997	12363116	85.06
Redmond	7090335	5835001	121.51	4470830	6558388	68.17	5191451	5286218	98.21	16752615	17679607	94.76
Scottsdale	7285432	7437475	97.96	3342958	2582690	129.44	2418623	1548373	156.2	13047013	11568538	112.78
Seattle	9140023	8039256	113.69	5256272	5084648	103.38	6891615	7551277	91.26	21287910	20675181	102.96
Springdale	6016127	4909659	122.54	3997161	9001083	44.41	2675373	4970546	53.82	12688661	18881288	67.2
Total	60679623	52565926	115.44	56558058	68358761	82.74	59589707	72760396	81.9	176827388	193685083	91.3

ANAYLSIS (CITYWISE SALES)

ProductCategory	2013			2012			2011			Total		
	GrossSales	Target	Variance	GrossSales	Target	Variance	GrossSales	Target	Variance	GrossSales	Target	Variance
Alcoholic Drinks	19164319	12816719	149.53	19735160	8236569	239.6	17570431	17213477	102.07	56469910	38266765	147.57
Bakery	18094427	8877975	203.81	11794632	32537376	36.25	19075592	30899118	61.74	48964651	72314469	67.71
Confectionary	1608605	3335102	48.23	1751065	2999486	58.38	1284075	1349638	95.14	4643745	7684226	60.43
Cool Drinks	1089105	2620667	41.56	1054727	1225074	86.09	827055	701305	117.93	2970886	4547046	65.34
Fruit Juices	6978641	5134847	135.91	8332277	4695103	177.47	8155245	7572774	107.69	23466163	17402724	134.84
Health Drinks	5569268	6668526	83.52	4893260	5764469	84.89	4460681	4588349	97.22	14923209	17021344	87.67
Ice Cream	5851498	5989675	97.69	6187398	7580701	81.62	5936579	5842490	101.61	17975475	19412866	92.6
Snacks	1466646	2213480	66.26	1893006	3151463	60.07	1300112	1326614	98	4659765	6691557	69.64
Tea	857114	4908935	17.46	916533	2168520	42.27	979937	3266631	30	2753584	10344086	26.62
Total	60679623	52565926	115.44	56558058	68358761	82.74	59589707	72760396	81.9	176827388	193685083	91.3

ANAYLSIS (PRODUCT CATEGORYWISE SALES)

EmployeeName	2013			2012			2011			Total		
	GrossSales	Target	Variance	GrossSales	Target	Variance	GrossSales	Target	Variance	GrossSales	Target	Variance
Boddy Jones	3500677	2866723	122.11	10642288	6508447	163.52	8149602	8181013	99.62	22292567	17556183	126.98
Bruce Adamson	3985668	3415998	116.68	3568365	3015256	118.34	4592783	4927714	93.2	12146816	11358968	106.94
Christine I Haas	3367338	2199374	153.1	1595471	1804893	88.4	754648	1362808	55.37	5717456	5367075	106.53
Daniel S Smith	2779519	2083913	133.38	385740	318143	121.25	900650	368440	244.45	4065909	2770496	146.76
David Brown	6833327	5217323	130.97	4069097	3409481	119.35	4496967	3661926	122.8	15399390	12288730	125.31
Heather Bruce	6241891	4448899	140.3	5666363	10852784	52.21	5893992	10245029	57.53	17802247	25546712	69.69
James H Walker	3850388	2387326	161.28	1187176	1675167	70.87	2394648	3889351	61.57	7432212	7951844	93.47
Jason V Mehta	1788071	2118506	84.4	952747	900453	105.81	2072810	2471141	83.88	4813629	5490100	87.68
Jenifer Jefferson	1162695	1479394	78.59	1629226	7212423	22.59	1093232	3034403	36.03	3885153	11726220	33.13
Jennifer K Lutz	3145588	2481108	126.78	3435000	2663391	128.97	3082174	2550555	120.84	9662762	7695054	125.57
Jennifer Paul	352544	411345	85.71	2367934	1788660	132.39	1582141	1936143	81.72	4302619	4136148	104.02
Makihiko Takako	3516098	1917731	183.35	3482069	9436467	36.9	6401972	10465133	61.17	13400140	21819331	61.41
Maria L Perez	1626144	2574690	63.16	1573291	1432392	109.84	1217858	787528	154.64	4417294	4794610	92.13
Maude F Setright	794934	1118955	71.04	1617218	1827621	88.49	2351303	3193827	73.62	4763455	6140403	77.58
Sally A Kwan	1805995	2246002	80.41	1889095	1327558	142.3	1143822	599335	190.85	4838911	4172895	115.96
Sybil P Johnson	4526426	4344543	104.19	1769667	1150298	153.84	1200765	760845	157.82	7496858	6255686	119.84
Vuong Paul	1961103	1972428	99.43	2117862	4744254	44.64	3503985	3394180	103.24	7582949	10110862	75
Vuong Smith	7040161	5493168	128.16	7573619	4396076	172.28	6647078	8195362	81.11	21260858	18084606	117.56
William T Jones	2401055	3788500	63.38	1035830	3894997	26.59	2109276	2735663	77.1	5546161	10419160	53.23
Total	60679623	52565926	115.44	56558058	68358761	82.74	59589707	72760396	81.9	176827388	193685083	91.3

ANAYLSIS (EMPLOYEEWISE SALES)

Associating analysis with State Column

State (Statewise Sales Analysis) = State (CityWise Sales Analysis)

ASSOCIATING PRODUCT CATEGORY COLUMN OF BOTH ANALYSES

Statewise Sales Analysis

State	2013	2012	2011	Total
	GrossSales	GrossSales	GrossSales	GrossSales
Arizona	10893193	5846560	6823258	23563010
Arkansas	14797291	17237143	15216444	47250877
Florida				
Ohio				
Washington				
Total				

Citywise Sales Analysis

City	2013			2012			2011			Total		
	GrossSales	Target	Variance	GrossSales	Target	Variance	GrossSales	Target	Variance	GrossSales	Target	Variance
Conway	8781164	6923147	126.84	13239982	15248860	86.83	12541071	18440391	68.01	34562216	40612398	85.1
Springda												
Total												

Productwise Sales Analysis

ProductCategory	2013			2012			2011			Total		
	GrossSales	Target	Variance	GrossSales	Target	Variance	GrossSales	Target	Variance	GrossSales	Target	Variance
Alcoholic Drinks	2469408	1447043	170.65	4496246	1516813	296.43	4247441	5263807	80.69	11213095	8227663	136.29
Bakery	3536385	1831889	193.05	2617517	7618949	34.36	3539876	7427911	47.66	9693778	16878749	57.43
Confectionary	182162	372378	48.92	609890	1167532	52.24	245116	280228	87.47	1037169	1820138	56.98
Cool Drinks	110141	301770	36.5	240961	224178	107.49	172741	127770	135.2	523842	653718	80.13
Fruit Juices	821865	613484	133.97	2293422	1023772	224.02	1632630	1795619	90.92	4747917	3432875	138.31
Health Drinks												
Ice Cream												
Snacks												
Tea												
Total												

Employeewise Sales Anal

EmployeeName	2013			2012			2011			Total		
	GrossSales	Target	Variance	GrossSales	Target	Variance	GrossSales	Target	Variance	GrossSales	Target	Variance
Heather Bruce	315310	235584	133.84	1196531	498909	239.83	848034	717266	118.23	2721661	1767754	153.96
Vuong Smith	506554	377900	134.04	1096891	524863	208.99	784595	1078353	72.76	3511603	3265256	107.54
Total	821865	613484	133.97	2293422	1023772	224.02	1632630	1795619	90.92	6233264	5033010	123.85

SUBVIEW

In the example above, **StatewiseSales** provides statewise details of **GrossSales** for various product categories. In the same analysis, if the user wants information about **GrossSales** achieved in the cities for the selected state, a new analysis **CityWiseSales**can be created.

Both **StatewiseSales** and **CityWiseSales**analysis have **State**, the common column, which is defined as **JOIN** condition between two analyses.

Some scenarios where Subview is useful:

- Analysis of total sales by customer
    - List of purchase orders for that customer
        - List of line item details within each purchase order

- Analysis of total budget by projects
    - Breakup of budgets by budget types for that project
        - Breakup of purchase orders placed against the project

> Note:
> You can associate crosstab object as subview to tabular object, and vice versa.
> You can associate crosstab or tabular created from real-time cubes with crosstab or tabular created from cache cubes, and vice versa.

4.15 What-if Analysis

What-if analysis is an important aspect of planning and managing.

The Smarten what-if analysis tool is easy to use, and users can follow simple steps to perform what-if analysis without any programming efforts. Smarten supports change in multiple input variables to analyze what-if scenarios.

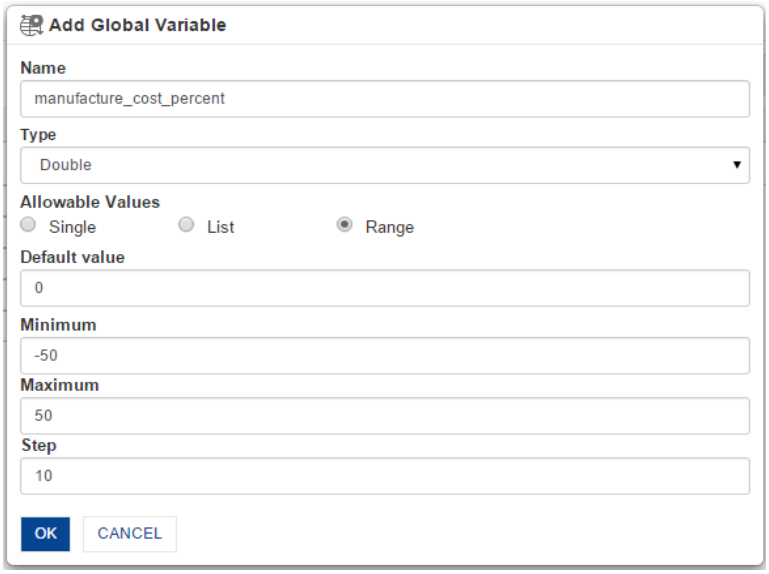
Example Scenario:
A companyis interestedin observing and understanding the impact of measures that can affect profit, which is dependent on internal factors, such as cost of the product, quantity of the product sold, and sales price.

Let’s have a look at performing What-If analysis step-by-step using Smarten.

Let us assume that the sales cube has manufacturing cost, labor cost, packaging cost, sales quantity, and sales price as a measure. Here, we have defined user-defined data columns for total cost, total sales, and profit.

Expression for TotalCost: (ManufactureCost + LabourCost + PackageCost) * qty_sales
Expression for TotalSales: qty_sales * salesprice_per_product
Expression for profit: TotalSales – Totalcost

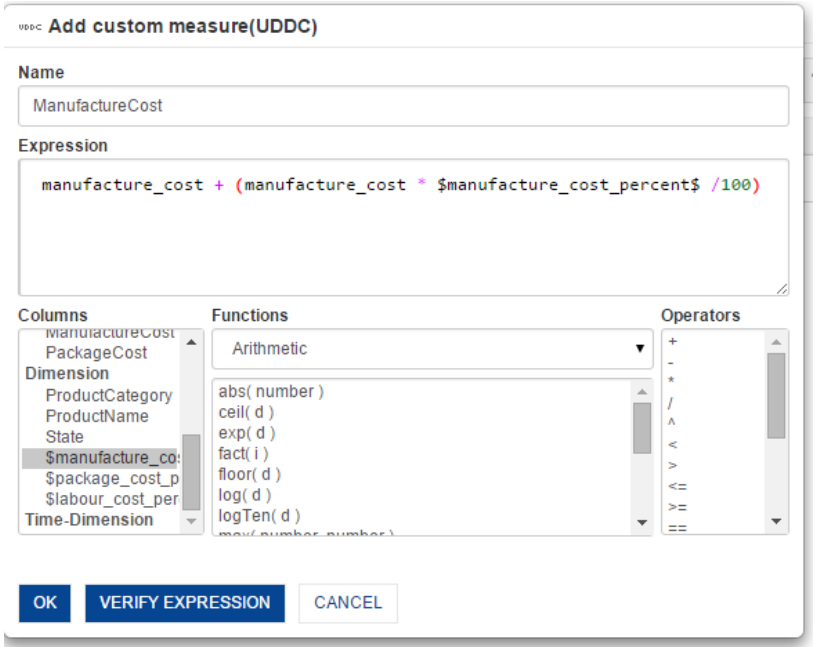
Step 1: Define global variables that can be used to tune different values of measures that impact profit.



ADD GLOBAL VARIABLE

Create global variables for manufacture_cost_percent, package_cost_percent, and labor_cost_percent.

Step 2: Define user-defined data columns to include global variable change effect in measures.

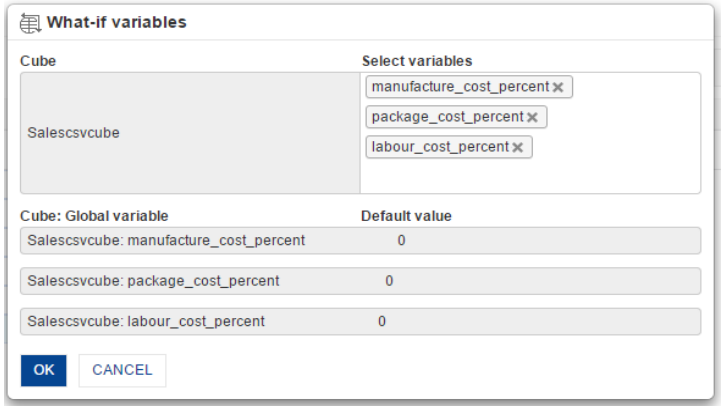


ADD CUSTOM MEASURE (USER-DEFINED DATA COLUMN)

Create user-defined columns (UDDC) for ManufactureCost, PackageCost, and LaborCost using respective global variables.

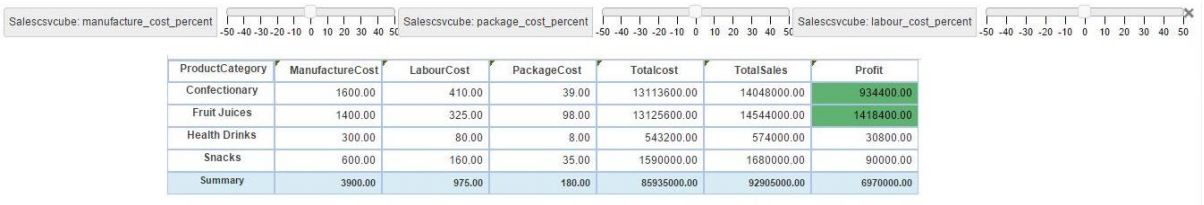
Expression for ManufactureCost: `manufacture_cost + (manufacture_cost * $manufacture_cost_percent$ / 100)`
Expression for PackageCost: package_cost + (package_cost * $package_cost_percent$ / 100)
Expression for LaborCost: Labor_cost + (labor_cost * $labor_cost_percent$ / 100)

Step 3: Select global variables to be included in what-if analysis.



SELECT WHAT-IF VARIABLES

The following screen shows crosstab with user-defined data columns and what-if variables. All global variables configured for What-If analysis are displayed with ranges defined for each variable.



ProductCategory	ManufactureCost	LabourCost	PackageCost	Totalcost	TotalSales	Profit
Confectionary	1600.00	410.00	39.00	13113600.00	14048000.00	934400.00
Fruit Juices	1400.00	325.00	98.00	13125600.00	14544000.00	1418400.00
Health Drinks	300.00	80.00	8.00	543200.00	574000.00	30800.00
Snacks	600.00	160.00	35.00	1590000.00	1680000.00	90000.00
Summary	3900.00	975.00	180.00	85935000.00	92905000.00	6970000.00

WHAT-IF ANALYSIS

User can change any variable, and that will change all the columns that are affected by that variable. For example, changing the labor_cost_percent will affect the LabourCost, TotalCost, and Profit columns.

The following screenshot shows that a 10% reduction in labor cost and a 5% reduction in manufacture cost will decrease the total cost from 8,59,35,000 to 8,04,65,250 and increase the profit from 69,70,000 to 1,24,39,750.

ProductCategory	ManufactureCost	LabourCost	PackageCost	Totalcost	TotalSales	Profit
Confectionary	1504.00	373.10	39.00	12263040.00	14048000.00	1784960.00
Fruit Juices	1316.00	295.75	98.00	12310200.00	14544000.00	2233800.00
Health Drinks	282.00	72.80	8.00	507920.00	574000.00	66080.00
Snacks	564.00	145.60	35.00	1489200.00	1680000.00	190800.00
Summary	3666.00	887.25	180.00	80465250.00	92905000.00	12439750.00

WHAT-IF ANALYSIS: AFTER APPLYING VARIABLE VALUE

Each time the user changes the input value of the what-if variable, it automatically recalculates the user-defined formula and shows the new result in all affected user-defined data columns.

Other sample scenarios for What-If analysis:

- What will be the change in the principal amount of investment if the interest rate decreases/increases by n percentage?
- How will our raw material or labor costs affect the bottom line if we make percentage changes to the salary of employees?
- What will be the change in profit if our product costs turn out to be 10% higher or lower than we have assumed?
- What will be the change in profit if we increase production by 15%?
- What will be the income tax if there is a change in the income tax slab rate by the government?
- What will be the change in EMI if the interest or loan period changes?

> Note:
> What-If analysis is available for crosstab, graph, GeoMap, KPI, tabular, and Dashboards.

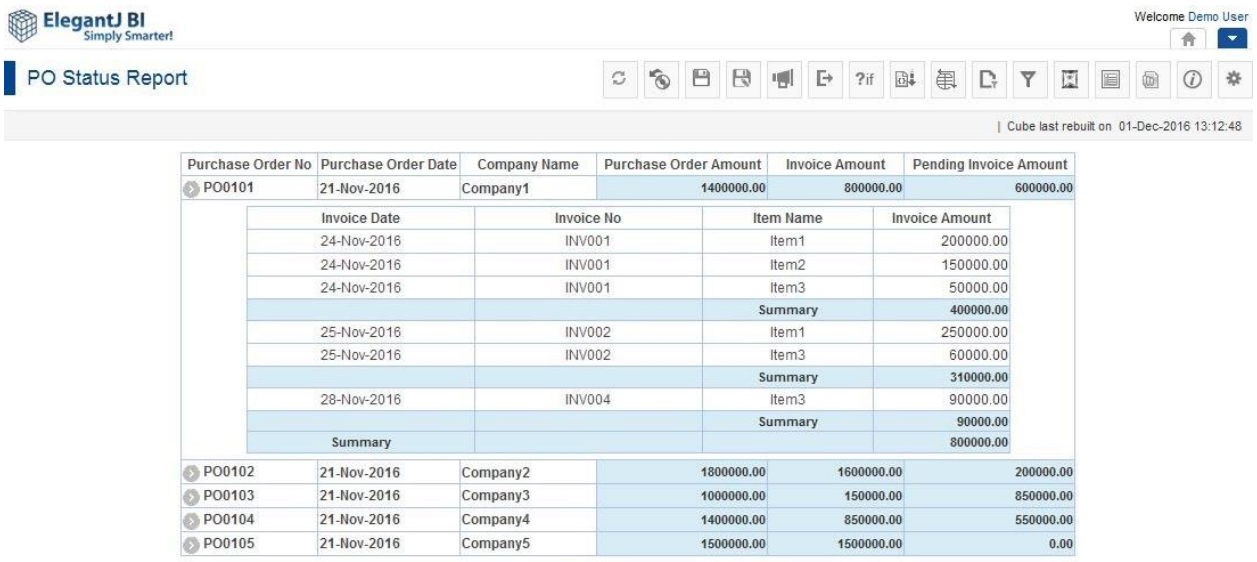
4.16 Master-Detail view in Tabular report

Master detail view in a tabular report to incorporate one to many and many to many relationships among different dimension values. For example, if data contains purchase orders and invoices and one purchase order has many invoices or vice versa, this view can be used to effectively show various display and summary options to effectively reflect this master detail relationship with a clickable view.

For example, here is the sample data with one purchase order having many invoices (one to many relationships).

Master—Purchase Order No., Purchase Order Date, Company Name, Purchase Order Amount, Total Invoice Amount, Pending Invoice Amount

Detail—Invoice Date, Invoice No., Item name, Invoice Amount



Purchase Order No	Purchase Order Date	Company Name	Purchase Order Amount	Invoice Amount	Pending Invoice Amount
PO0101	21-Nov-2016	Company1	1400000.00	800000.00	600000.00

Invoice Date	Invoice No	Item Name	Invoice Amount
24-Nov-2016	INV001	Item1	200000.00
24-Nov-2016	INV001	Item2	150000.00
24-Nov-2016	INV001	Item3	50000.00
		Summary	400000.00
25-Nov-2016	INV002	Item1	250000.00
25-Nov-2016	INV002	Item3	60000.00
		Summary	310000.00
28-Nov-2016	INV004	Item3	90000.00
		Summary	90000.00
Summary			800000.00

PO0102	21-Nov-2016	Company2	1800000.00	1600000.00	200000.00
PO0103	21-Nov-2016	Company3	1000000.00	150000.00	850000.00
PO0104	21-Nov-2016	Company4	1400000.00	850000.00	550000.00
PO0105	21-Nov-2016	Company5	1500000.00	1500000.00	0.00

TABULAR REPORT: MASTER DETAIL VIEW

Note:
Master-Detail view is available for Tabular only

5 Filters and Expressions

Various kinds of filters are available in Smarten. These filters are Outliner Filters, Retrieval Filters, Dimension Filters, Measure Filters, Advanced Filters, Time Series Filters, Show/Hide Row or Column, Data Filters (Custom Cube Dimension and Custom Cube Measure Filters), and Page Filters.

Filters are made interdependent throughout the system. If you have used two filters, setting value in one filter will filter values in the other filter.

For example, if you have used two filters (e.g., State and City), selecting “Washington” from the state filter will display only cities in “Washington” state (e.g., Redmond and Seattle) in the city filter.

Name of the Filter	Back-end	Front-end
Page Filter	✓	
Retrieval Parameters	✓	
Time Series Filters	✓	
Dashboard Section Filters (All types)	✓	
Front-end Object Filters, including advanced filters (on Dimension, Measures, UDDC)	✓	✓
	Filter is applied on cube data	Filter is applied on Front-end object data

Example of an analysis without applying any filter:

ProductCategory	ProductName	GrossSales
Alcoholic Drinks	Beer	35,924,439.37
	Whiskey	32,819,109.98
	Wine	18,610,484.55
Bakery	Cake	10,869,749.00
	Cookies	78,955,357.74
Confectionery	Mints	5,742,339.66
	Toffees	1,295,971.72
Cool Drinks	Cola	2,065,441.55
	Soda	2,187,818.82
Fruit Juices	Apple	13,710,800.92
	Mango	14,994,919.65
	Orange	9,909,306.19
Health Drinks	Chocolate	16,978,290.40
	Strawberry	4,741,740.16

CUBE DATA

ProductCategory	ProductName	GrossSales
Alcoholic Drinks	Beer	35,924,439.37
	Whiskey	32,819,109.98
	Wine	18,610,484.55
Bakery	Cake	10,869,749.00
	Cookies	78,955,357.74
Confectionery	Mints	5,742,339.66
	Toffees	1,295,971.72
Cool Drinks	Cola	2,065,441.55
	Soda	2,187,818.82
Fruit Juices	Apple	13,710,800.92
	Mango	14,994,919.65
	Orange	9,909,306.19
Health Drinks	Chocolate	16,978,290.40
	Strawberry	4,741,740.16

DATA ON THE FRONT-END WITHOUT APPLYING ANY FILTER

Example 1 of an analysis with the back-end and front-end filters applied:

ProductCategory	ProductName	GrossSales
Alcoholic Drinks	Beer	35,924,439.37
	Whiskey	32,819,109.98
	Wine	18,610,484.55
Bakery	Cake	10,869,749.00
	Cookies	78,955,357.74
Confectionery	Mints	5,742,339.66
	Toffees	1,295,971.72
Cool Drinks	Cola	2,065,441.55
	Soda	2,187,818.82
Fruit Juices	Apple	13,710,800.92
	Mango	14,994,919.65
	Orange	9,909,306.19
Health Drinks	Chocolate	16,978,290.40
	Strawberry	4,741,740.16

CUBE DATA

ProductCategory	ProductName	GrossSales
Alcoholic Drinks	Beer	35,924,439.37
	Whiskey	32,819,109.98
	Wine	18,610,484.55
Cool Drinks	Cola	2,065,441.55
	Soda	2,187,818.82
Health Drinks	Chocolate	16,978,290.40
	Strawberry	4,741,740.16

OBJECT DATA AFTER APPLYING A BACK-END FILTER ON THE COLUMN (NO FRONT-END FILTER)
PRODUCTCATEGORY = "ALCOHOLIC DRINKS," "COOL DRINKS," AND "HEALTH DRINKS"

ProductCategory	ProductName	GrossSales
Alcoholic Drinks	Beer	35,924,439.37
	Wine	18,610,484.55
Cool Drinks	Cola	2,065,441.55
Health Drinks	Chocolate	16,978,290.40

OBJECT DATA AFTER APPLYING A FRONT-END FILTER ON THE COLUMN (NO BACK-END FILTER)
PRODUCTNAME = "BEER," "WINE," "COLA," AND "CHOCOLATE"

Example 2 of an analysis with the back-end and front-end filters applied:

ProductCategory	ProductName	GrossSales
Alcoholic Drinks	Beer	35,924,439.37
	Whiskey	32,819,109.98
	Wine	18,610,484.55
Bakery	Cake	10,869,749.00
	Cookies	78,955,357.74
Confectionery	Mints	5,742,339.66
	Toffees	1,295,971.72
Cool Drinks	Cola	2,065,441.55
	Soda	2,187,818.82
Fruit Juices	Apple	13,710,800.92
	Mango	14,994,919.65
	Orange	9,909,306.19
Health Drinks	Chocolate	16,978,290.40
	Strawberry	4,741,740.16

CUBE DATA

ProductCategory	ProductName	GrossSales
Bakery	Cake	10,869,749.00
	Cookies	78,955,357.74
Confectionery	Mints	5,742,339.66
	Toffees	1,295,971.72
Fruit Juices	Apple	13,710,800.92
	Mango	14,994,919.65
	Orange	9,909,306.19

OBJECT DATA AFTER APPLYING A BACK-END FILTER ON THE COLUMN
PRODUCTCATEGORY = "BAKERY," "CONFECTIONARY," AND "FRUIT JUICES"

ProductCategory	ProductName	GrossSales
Bakery	Cookies	78,955,357.74
Confectionery	Toffees	1,295,971.72
Fruit Juices	Orange	9,909,306.19
	Mango	14,994,919.65

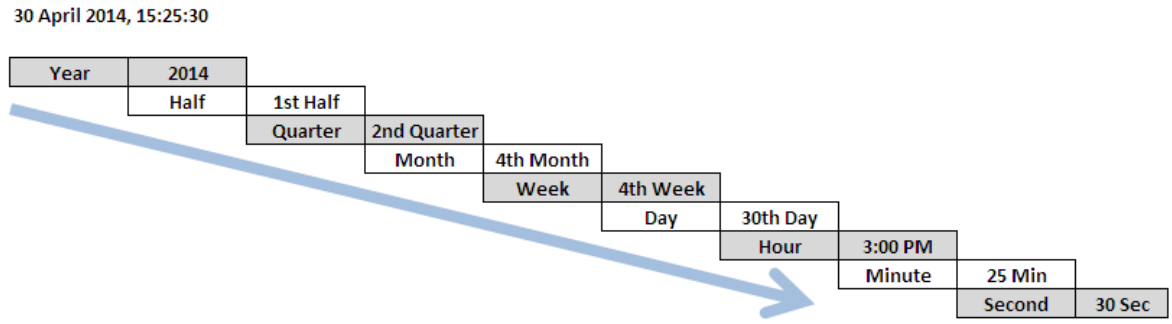
THE DISPLAY DATA AFTER APPLYING A FRONT-END FILTER ON THE COLUMN
PRODUCTNAME = "COOKIES," "TOFFEES," "SODA," "ORANGE," AND "MANGO"
(WHILE THE BACK-END FILTER IS ALREADY APPLIED)

We can see that the product name "Soda" is included in the front-end filter condition, but it is not displayed at the front end because the ProductCategory "Cool Drinks," which contains the product name "Soda," is excluded by the back-end filter, and therefore the product "Soda" is not included.

5.1 Time Series (absolute, relative, range comparisons)

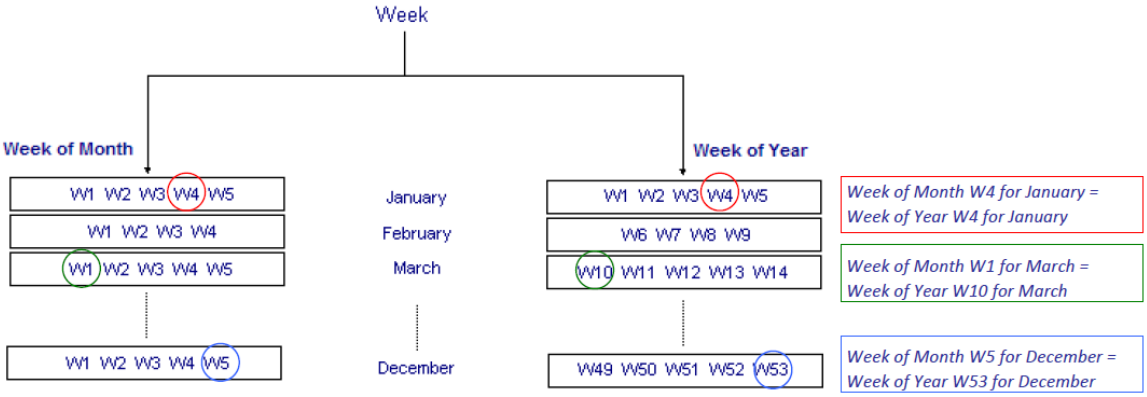
Time Series is defined as an ordered sequence of equally spaced time intervals. When monitoring business processes or tracking corporate business metrics, a need often arises for usage of time series data across financial and calendar years and then down to half years, quarters, months, weeks, days and dates, days of the year, and weeks of the year.

Smarten built-in customizable time series lets you analyze what has changed over the previous years, half years, quarters, months, weeks, days, dates, and other critical measures.



For the dimension "Week," Smarten uses two different types of representation for Week, i.e., Week of the Month and Week of the Year.

The difference between Week of the Year and Week of the Month is as follows: In Week of the Month, each starting week of a month begins as the 1st week, whereas in Week of the Year, the starting week of January is the 1st week of the year, when the year is a calendar year, starting on 1 January.



DIFFERENT REPRESENTATIONS FOR WEEK

5.1.1 Absolute Time Series

The absolute time filtering option is used to know the value of a measure at a particular year or half year or quarter or month or week or day or date. It has no dependency or relevance to the current date.

Months are displayed as M1, M2 . . . M12 for the absolute time series selection, with M1 being the first month of the year and M12 being the last month of the year. For example, if the selected date/time field has time series with start of a month as January, then M1 would be January, and M12 would be December, and if selected date/time field has time series with start month of April, then M1 would be April, and M12 would be March.

Examples:

Months	Year Starting from January	Year Starting from April
M1	January	April
M2	February	May
M3	March	June
M4	April	July
M5	May	August
M6	June	September
M7	July	October
M8	August	November
M9	September	December
M10	October	January
M11	November	February
M12	December	March

The start month for time series on any date field can be defined through the cube definition process. Here two scenarios with different starting months are shown.

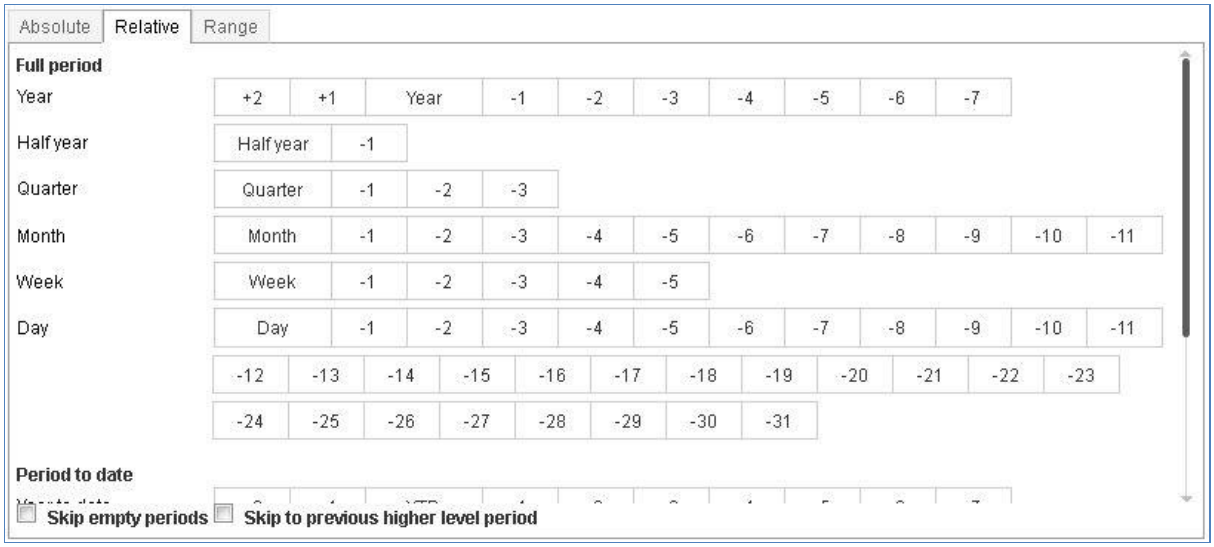
> Note:
> Time series hierarchy displayed in Time Series dialogue may vary based on the configuration of time series hierarchy settings by the administrator.

5.1.2 Relative Time Series

Relative time filtering is used to know the value of a measure at a particular period relative to the current date. Here the current date value affects the definition of time series periods.

5.1.2.1 Relative Time Series filtering using full period

Filtering from start to end date for a particular period—year, half year, quarter, month, week of month, week of year or day.



RELATIVE TIME SERIES FILTER USING FULL PERIOD

> Note:
> All examples are based on calendar year starting from 1 Jan.
> Time series hierarchy displayed in Time Series dialogue may vary based on the configuration of time series hierarchy settings by the administrator.

Examples (without skip empty periods):

Example 1: Full Data for current year and previous year

Selected Fields	Result on 31 July 2013	Result on 15 July 2012
Year: Year, Year -1	Data for 2013	Data for 2012
	Data for 2012	Data for 2011

Example 2: This week and previous week of current month for current year

Selected Fields	Result on 31 July 2013	Result on 15 July 2013
Year: Year, Month: Month Week: Week, Week -1	Data for 5th week of July 2013	Data for 3rd week of July 2013
	Data for 4th week of July 2013	Data for 2nd week of July 2013

Example 3: This week of current month of this year and previous year

Selected Fields	Result on 31 July 2013	Result on 15 July 2013
Year: Year, Year -1 Month: Month Week: Week	Data for 5th week of July 2012	Data for 3rd week of July 2012
	Data for 5th week of July 2013	Data for 3rd week of July 2013

Example 4: Today and yesterday of current month of this year

Selected Fields	Result on 31 July 2013	Result on 15 July 2013
Day: Day, Day -1	Data for 31 July 2013 Data for 30 July 2013	Data for 15 July 2013 Data for 14 July 2013

Example 5: current half year for current year

Selected Fields	Result on 31 July 2014	Result on 15 April 2014
Year: Year Half Year: Half Year	Data for July 2014	Data for Jan 2014 Data for Feb 2014 Data for Mar 2014 Data for April 2014

Example 6: current half year for current year and previous year

Selected Fields	Result on 31 July 2014	Result on 15 April 2014
Year: Year, Year -1 Half Year: Half Year	Data for July 2013 Data for July 2014	Data for Jan 2013 Data for Feb 2013 Data for Mar 2013 Data for April 2013 Data for Jan 2014 Data for Feb 2014 Data for Mar 2014 Data for April 2014

Example 7: previous half year for current year

Selected Fields	Result on 31 July 2014	Result on 15 June 2014
Year: Year Half Year: Half Year -1	Data from Jan 2014 to Jun 2014	Data from July 2013 to Dec 2013

Example 8: current month and previous month for current year

Selected Fields	Result on 31 July 2014	Result on 15 July 2013
Year: Year, Month: Month, Month -1	Data for June 2014 Data for July 2014	Data for June 2013 Data for July 2013

Example 9: current month for current year and previous year

Selected Fields	Result on 31 July 2014	Result on 15 July 2013
Year: Year, Year -1 Month: Month	Data for July 2013 Data for July 2014	Data for July 2012 Data for July 2013

Example 10: current quarter and previous quarter for current year

Selected Fields	Result on 31 July 2014	Result on 15 July 2013
Year: Year Quarter: Quarter, Quarter -1	Data for April 2014 Data for May 2014 Data for June 2014 Data for July 2014	Data for April 2013 Data for May 2013 Data for June 2013 Data for July 2013

Example 11: current quarter for current year and previous year

Selected Fields	Result on 31 July 2014	Result on 15 July 2013
Year: Year, Year -1 Month: Quarter	Data for July 2013 Data for July 2014	Data for July 2012 Data for July 2013

Example 12: 20th week for current year and previous year

Selected Fields	Result on 31 July 2014	Result on 15 July 2013
Year: Year, Year -1 Week (Year): 20	Data for 3rd Week of May 2014 Data for 3rd Week of May 2013	Data for 3rd Week of May 2012 Data for 3rd Week of May 2013

Examples (with skip empty periods):

Example 1: Full Data for current year and previous year (When data for previous year is not present)

Selected Fields	Result on 31 July 2013	Result on 15 July 2012
Year: Year, Year -1	Data for 2013	Data for 2012
	Data for 2011	Data for 2010

Example 2: This month and previous month for current year (When data for previous month is not present)

Selected Fields	Result on 31 July 2013	Result on 15 July 2013
Year: Year Month: Month, Month -1	Data for July 2013	Data for July 2013
	Data for May 2013	Data for May 2013

Example 3: This week and previous week of current month for current year (When data for previous week is not present)

Selected Fields	Result on 31 July 2013	Result on 15 July 2013
Year: Year Month: Month Week: Week, Week -1	Data for 5th week of July 2013	Data for 3rd week of July 2013
	Data for 3th week of July 2013	Data for 1nd week of July 2013

Example 4: Today and yesterday (When data for yesterday is not available)

Selected Fields	Result on 31 July 2013	Result on 15 July 2013
Day: Day, Day -1	Data for 31 July 2013 Data for 29 July 2013	Data for 15 July 2013 Data for 13 July 2013

Examples (without Skip to previous higher level period):

Example 1: Data for current Quarter and previous Quarter (when current Quarter is Quarter 1)

Selected Fields	Result on 15 Mar 2015
Year: Year Quarter: Quarter,Quarter-1	Data for Quarter 1 of 2015

Example 2: Data for current Month and previous Month (when current Month is January)

Selected Fields	Result on 31 Jan 2015
Year: Year Month: Month, Month-1	Data for Jan 2015

Example 3: Data for Week and previous Week of current Month for current Year (when Current week is 1st Week)

Selected Fields	Result on 6 Mar 2015
Year: Year Month: Month Week: Week, Week -1	Data for 1st week of Mar 2015

Examples (with Skip to previous higher level period):

Example 1: Data for current Quarter and previous Quarter (when current Quarter is Quarter 1)

Selected Fields	Result on 15 Mar 2015
Year: Year Quarter: Quarter,Quarter-1	Data for Quarter 1 of 2015 Data for Quarter 4 of 2014

Example 2: Data for current Month and previous Month (when current Month is January)

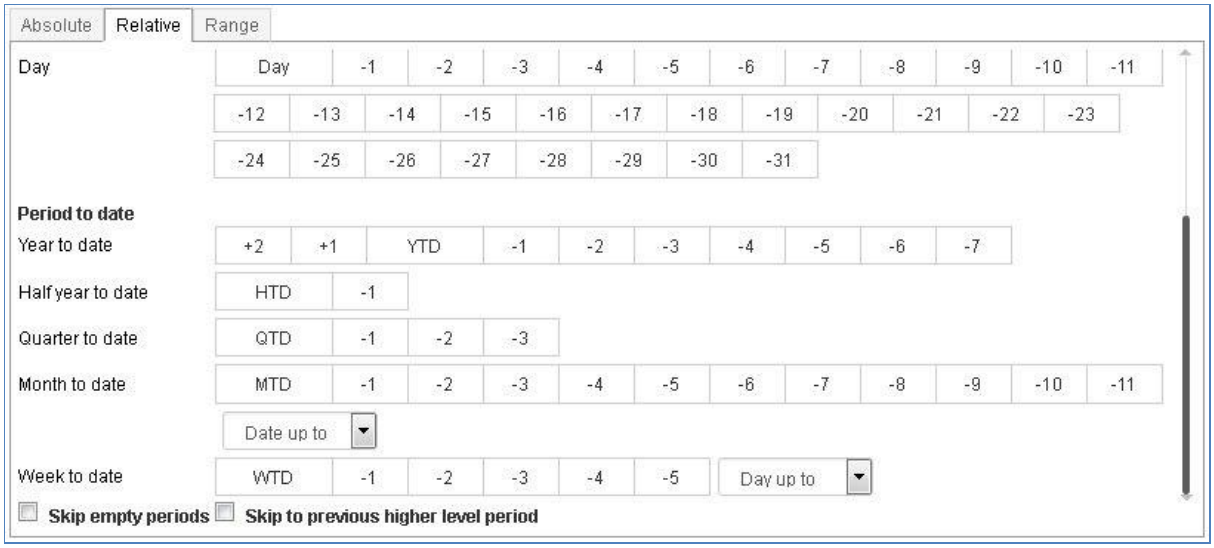
Selected Fields	Result on 31 Jan 2015
Year: Year Month: Month, Month-1	Data for Jan 2015 Data for Dec 2014

Example 3: Data for Week and previous Week of current Month for current Year (when Current week is 1st Week)

Selected Fields	Result on 6 Mar 2015
Year: Year Month: Month Week: Week, Week -1	Data for 1st week of Mar 2015 Data for 4th week of Feb 2015

5.1.2.2 Relative Time Series filtering using period to date

Filtering from the start to the current date for a particular period—for year, half year, quarter, week of month, and week of year.



RELATIVE TIME SERIES FILTER USING PERIOD TO DATE

> Note:
> All examples are based on calendar year starting from 1 Jan.
> Time series hierarchy displayed in Time Series dialogue may vary based on the configuration of time series hierarchy settings by the administrator.

Examples (without skip empty periods):

Example 1: Year to date data for current year and previous year

Selected Fields	Result on 31 July 2014	Result on 15 July 2014
YTD: YTD, YTD -1	Data for 1 Jan to 31 July 2013 Data for 1 Jan to 31 July 2014	Data for 1 Jan to 15 July 2013 Data for 1 Jan to 15 July 2014

Example 2: Half year to date data for current half year and current year

Selected Fields	Result on 31 July 2014	Result on 15 April 2014
YTD: YTD HTD: HTD	Data for 1 July 2014 to 31 July 2014	Data for 1 Jan 2014 to 15 April 2014

Example 3: Quarter to date data for previous quarter and current year

Selected Fields	Result on 31 July 2014	Result on 15 July 2014
YTD: YTD QTD: QTD -1	Data for 1 Apr 2014 to 30 Apr 2014	Data for 1 Apr 2014 to 15 Apr 2014
YTD: YTD QTD: QTD -1 MTD date up to 20th	Data for 1 Apr 2014 to 20 Apr 2014	Data for 1 Apr 2014 to 20 Apr 2014

Example 4: Month to date data for current quarter and current year

Selected Fields	Result on 31 July 2014	Result on 15 July 2014
YTD: YTD QTD: QTD Month: MTD MTD date up to Current Date	Data for 1 July to 31 July 2014	Data for 1 July to 15 July 2014
MTD date up to 20th	Data for 1 July to 20 July 2014	Data for 1 July to 20 July 2014

Example 5: Month to date data for current year and previous year

Selected Fields	Result on 31 July 2014	Result on 15 July 2014
YTD: YTD, YTD -1 QTD: QTD MTD: MTD MTD date up to Current Date	Data for 1 July to 31 July 2013 Data for 1 July to 31 July2014	Data for 1 July to 15 July 2013 Data for 1 July to 15 July 2014
MTD date up to 20 July	Data for 1 July to 20 July 2013 Data for 1 July to 20 July 2014	Data for 1 July to 20 July 2013 Data for 1 July to 15 July 2014

Example 6: Month to date data for current month and previous month for current year

Selected Fields	Result on 31 July 2014	Result on 15 July 2013
YTD: YTD MTD: MTD, MTD -1 MTD date up to Current Date	Data for 1 June to 30 June 2014 Data for 1 July to 31 July 2014	Data for 1 June to 15 June 2013 Data for 1 July to 15 July 2013
MTD date up to 20 July	Data for 1 June to 20 June 2014 Data for 1 July to 20 July 2014	Data for 1 June to 20 June 2013 Data for 1 July to 15 July 2014

Example 7: Quarter to date data for previous quarter and current quarter of current year

Selected Fields	Result on 31 March 2014	Result on 31 March 2013
YTD: YTD QTD: QTD, QTD -1	Data for 1stQtr 2014 Data for 4thQtr 2013	Data for 1stQtr 2013 Data for 4thQtr 2012

Examples (with skip empty periods):

Example 1: Year to date data for current year and previous year (When data for previous year is not present)

Selected Fields	Result on 31 July 2014	Result on 15 July 2014
YTD: YTD, YTD -1	Data for 1 Jan to 31 July 2014 Data for 1 Jan to 31July 2013	Data for 1 Jan to 15 July 2014 Data for 1 Jan to 15 July 2013

Example 2: Quarter to date data for current quarter and previous quarter for current year (When data for previous quarter is not present)

Selected Fields	Result on 31 July 2014	Result on 15 April 2014
YTD: YTD QTD: QTD, QTD -1	Data for 1 July 2014 to 31 July 2014 Data for 1 January 2014 to 31 January 2014	Data for 1 April 2014 to 15 April 2014 Data for 1 October 2013 to 15 October 2013

Example 3: Month to date data for current month and previous month of current year (When data for previous month is not present)

Selected Fields	Result on 31 July 2014	Result on 15 July 2014
YTD: YTD Month: MTD, MTD-1	Data for 1 July to 31 July 2014 Data for 1 May to 31 May 2014	Data for 1 July to 15 July 2014 Data for 1 May to 15 May 2014

Examples (without Skip to previous higher level period):

Example 1: Quarter to date Data for current Quarter and previous Quarter (when current Quarter is Quarter 1)

Selected Fields	Result on 15 Mar 2015
Year: YTD Quarter: QTD,QTD-1	Data for 1 January 2015 to 15 March 2015

Example 2: Month to date Data for current Month and previous Month (when current Month is January)

Selected Fields	Result on 15 Jan 2015
Year: YTD Month: MTD, MTD-1	Data for 1 January 2015 to 15 January 2015

Example 3: Week to date Data for Week and previous Week of current Month for current Year (when Current week is 1st Week)

Selected Fields	Result on 6 Mar 2015
Year: YTD Month: MTD Week: WTD, WTD -1	Data for 1st March to 6th March 2015

Examples (with Skip to previous higher level period):

Example 1: Quarter to date Data for current Quarter and previous Quarter (when current Quarter is Quarter 1)

Selected Fields	Result on 15 Mar 2015
Year: YTD Quarter: QTD,QTD-1	Data for 1 January 2015 to 15 March 2015 Data for 1 October 2014 to 15 December 2014

Example 2: Month to date Data for current Month and previous Month (when current Month is January)

Selected Fields	Result on 15 Jan 2015
Year: YTD Month: MTD, MTD-1	Data for 1 January 2015 to 15 January 2015 Data for 1 December 2014 to 15 December 2014

Example 3: Week to date Data for Week and previous Week of current Month for current Year (when Current week is 1st Week)

Selected Fields	Result on 6 Mar 2015
Year: YTD Month: MTD Week: WTD, WTD -1	Data for 1st March 2015 to 6th March 2015 Data for 22nd February 2015 to 27th February 2015

5.1.3 Range Time Series

This option is used to filter time based on range and custom periods. Users can apply simple time filtering based on before, after, range, and other conditions.

> Note:
> All examples are based on calendar year starting from 1 Jan.
> Time series hierarchy displayed in Time Series dialogue may vary based on the configuration of time series hierarchy settings by the administrator.

Examples:

Options	On 10 Feb 2013	On 31 July 2012	On 31 March 2011
Today	10 Feb 2013	31 July 2012	31 March 2011
This day	10th of every month	31st of every month	31st of every month
This week	2nd week of every month	5th week of every month	5th week of every month
This month	2nd month of every year	7th month of every year	3rd month of every year
This quarter	1st quarter of every year	3rd quarter of every year	1st quarter of every year
This half year	1st half year of every year	2nd half year of every year	1st half year of every year
This year	2013	2012	2011

Selected Fields	Options	Result on 31July 2014
Before	1 May 2014	Data up to 30 April 2014
After	10 Feb 2014	Data from 11 Feb 2014 to 31 July 2014
Between	15 March 2014 to 20 May 2014	Data from 15 March to 20 May 2014
Not Between	7 March 2014 to 1 July 2014	Data up to 6 March 2014 and Data from 2 July 2014 to 31 July 2014

5.2 Advanced Filter

The advanced filter is a type of filter that can be applied on the dimensions as well as measures. Users can create filters based on various string, arithmetic, date, statistics, trigonometry, or conditional statements using various arithmetic operators (such as +, -, /, etc.) or comparison operators (such as =, >, < etc.)

Arithmetic Functions	
Functions	Description
ABS	Returns absolute value of a number
CEIL	Returns the smallest whole number that is greater than or equal to a specified number
EXP	Returns exponential value of a number
FACT	Returns factorial of a number
FLOOR	Returns the largest whole number that is smaller than or equal to a specified number
LOG	Returns natural logarithm (base e) of a number
LOGTEN	Returns decimal logarithm (base 10) of a number
MAX	Returns larger of two numbers
MIN	Returns smaller of two numbers
MOD	Returns modulus of two numbers (the remainder after dividing the first number into the other number)
PI	Returns pi (3.14159265358979323) times a number
RANDOM	Returns a random whole number between two specified numbers
ROUND	Returns a number rounded off decimal numbers
SIGN	Returns a number (-1, 0, or 1) indicating the sign of a number
SQRT	Returns the square root of a number

Statistic Functions	
Functions	Description
AVG	Returns average value of the expression
COUNT	Returns number of cases in the expression
MAXIMUM	Returns maximum value of the expression
MINIMUM	Returns minimum value of the expression
SUM	Returns sum total of the expression

String Functions	
Functions	Description
ASC	Returns ASCII value of a character
BOOLEANVALUE	Returns contents of a string as Boolean
BYTEVALUE	Returns contents of a string as byte
CHARVALUE	Returns contents of an integer as character
DOUBLEVALUE	Returns contents of a string as double
FILL	Returns a string of a specified length filled with occurrences of a specified string
FLOATVALUE	Returns contents of a string as float
INDEXOFCHAR	Returns the starting position of a character within a specified string
INDEXOFSTRING	Returns the starting position of a string within a specified string
INTVALUE	Returns contents of a string as integer
ISDATE	Determine if the specified string contains a valid date
ISNULL	Determine if the argument is NULL
ISNUMBER	Determine if the specified string contains a number
ISTIME	Determine if the specified string contains a valid time
LEFT	Returns a specified number of characters from a string, starting with the first character
LEFTTRIM	Returns a copy of a specified string with leading blanks removed
LENGTH	Returns length of a string
LONGVALUE	Returns contents of a string as long
MATCH	Returns a determination whether a string contains a particular pattern of characters
REPLACE	Returns a copy of a specified string in which a specified number of characters, starting with a specified character, have been replaced with characters from another specified string
REVERSE	Reverses the order or characters in a string
RIGHT	Returns the specified number of characters from the end of a specified string
RIGHTTRIM	Returns a copy of a specified string with trailing blanks removed
SHORTVALUE	Returns contents of a string as short
SPACE	Returns a string of a specified length filled with a specified number of spaces
SUBSTRING	Returns a string containing a character copied (starting at a specified position and ending at a specified position) from a specified string
TOLOWERCASE	Returns a copy of a specified string with all uppercase letters converted to lowercase
TOSTRING	Returns a string representation of a specified object
TOUPPERCASE	Returns a copy of a specified string with all lowercase letters converted to uppercase
TRIM	Returns a string with leading and trailing blanks removed
TRUNCATE	Returns a number truncated to a specified number of decimal places

Trigonometric Functions	
Functions	Description
COS	Cosine of number (number in radian)
SIN	Sine of number (number in radian)
TAN	Tangent of number (number in radian)

Miscellaneous Functions	
Functions	Description
IFCASE	Returns TRUE if condition is validated and returns FALSE if invalidated
WHENTHEN	Tests values of a column or expression and returns values based on the results of the test

Date Functions	
Functions	Description
DatePart (period, source)	datePart("d",dateTime("2001-02-16 20:38:40")) Returns 16 datePart("m",dateTime("2001-02-16 20:38:40")) Returns 2 datePart("y",dateTime("2001-02-16 20:38:40")) Returns 2001 datePart("q",dateTime("2001-02-16 20:38:40")) Returns 1 datePart("h",dateTime("2001-02-16 20:38:40")) Returns 20 datePart("n",dateTime("2001-02-16 20:38:40")) Returns 38 datePart("s",dateTime("2001-02-16 20:38:40")) Returns 40 datePart("w",dateTime("2001-02-16 20:38:40")) Returns 7 Return Value: Returns an Integer value containing the specified component of a given Date value.
DateAdd (type, date, value)	dateAdd("d",10,dateTime("2001-02-16 20:38:40")) Returns 26-Feb-2001 20:38:40 dateAdd("m",2,dateTime("2001-02-16 20:38:40")) Returns 16-Apr-2001 20:38:40 dateAdd("y",2,dateTime("2001-02-16 20:38:40")) Returns 16-Feb-2003 20:38:40 dateAdd("q",2,dateTime("2001-02-16 20:38:40")) Returns 16-Aug-2001 20:38:40 dateAdd("w",2,dateTime("2001-02-16 20:38:40")) Returns 02-Mar-2001 20:38:40 dateAdd("h",2,dateTime("2001-02-16 20:38:40")) Returns 16-Feb-2001 22:38:40 dateAdd("n",2,dateTime("2001-02-16 20:38:40")) Returns 16-Feb-2001 20:40:40 dateAdd("s",2,dateTime("2001-02-16 20:38:40")) Returns 16-Feb-2001 20:38:42 Return Value: Returns a Date value containing a date and time value to which a specified time interval has been added.
DateDiff (type, date1, date2)	dateDiff("d", dateTime("2001-02-18 20:38:40"),dateTime("2001-02-16 20:38:40")) Returns 2 dateDiff("m", dateTime("2001-02-16 20:38:40"),dateTime("2001-05-16 20:38:40")) Returns -3 dateDiff("y", dateTime("2003-02-16 20:38:40"),dateTime("2001-02-16 20:38:40")) Returns 2 dateDiff("q", dateTime("2001-07-16 20:38:40"),dateTime("2001-02-16 20:38:40")) Returns 2 dateDiff("w", dateTime("2001-02-18 20:38:40"),dateTime("2001-02-06 20:38:40")) Returns 2 dateDiff("h", dateTime("2001-02-16 20:38:40"),dateTime("2001-02-16

	10:38:40")) Returns 10 dateDiff("n", dateTime("2001-02-16 20:38:40"),dateTime("2001-02-16 20:18:40")) Returns 20 dateDiff("s", dateTime("2001-02-16 20:38:40"),dateTime("2001-02-16 20:38:10")) Returns 30 Return Value: Returns a Long value specifying the number of time intervals between two Date values.
MonthName (number1, [abbreviate], [number2])	monthName(1,false, 1) Returns January monthName(1,true, 1) Returns Jan Return Value: Returns a month name representing the month for a number from 1 to 12.
WeekdayName (number1, [abbreviate], [number2])	weekdayName(2, true, 3) Returns Wed weekdayName(2, false, 3) Returns Wednesday Return Value: Returns a day name representing the day of the week for a number from 1 to 7.
FormatDate (date, “string”)	FormatDate ('2001-02-16',’yy/mm/dd’) Returns 01/02/14 formatDate(dateTime("2001-02-16 20:38:40"), "MM/dd/yyyy") Returns 02/16/2001 Return Value: Returns string of the specified format for a specified date.
date(object)	date("2001-02-16") Returns 16-Feb-2001
dateTime(object)	dateTime("2001-02-16 20:38:40") Returns 16-Feb-2001 20:38:40
day(date)	day(dateTime("2001-02-16 20:38:40")) Returns 16
dayName (date)	dayName(dateTime("2001-02-16 20:38:40")) Returns Friday
dayNumber(date)	dayNumber(dateTime("2001-02-16 20:38:40")) Returns 6
daysAfter(date,date)	daysAfter(dateTime("2001-02-16 20:38:40"),dateTime("2001-02-10 20:38:40")) Returns 6
hour(date)	hour(dateTime("2001-02-16 20:38:40")) Returns 20
minute(date)	minute(dateTime("2001-02-16 20:38:40")) Returns 38
month(date)	month(dateTime("2001-02-16 20:38:40")) Returns 2
now()	now() Returns 20:38:40 Return value : Returns current time
relativeDate(date, i)	relativeDate(dateTime("2001-02-16 20:38:40"), 5) Returns Wed Feb 21 20:38:40 IST 2001 Return value: Returns the date that occurs n days after a given date
time(object)	time("20:38:40") Returns 20:38:40
relativeTime(time , I)	relativeTime(time("20:38:40"), 5) Returns 20:38:45 Return value: Returns the time that occurs n seconds after a given time
second(time)	second(time("20:38:40")) Returns 40
today( )	today( ) Returns 16-Feb-2001 Return value: Returns the current system date
year(date)	year(dateTime("2001-02-16 20:38:40")) Returns 2001

Examples:

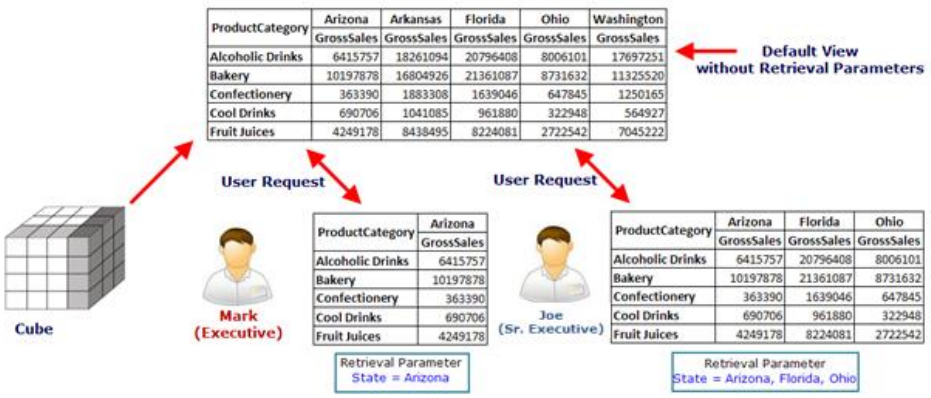
Filter Description	Expression
Filter Bakery, Cool Drinks, Health Drinks category for February, April, June of current year	(ProductCategory == "Bakery" || ProductCategory == "Cool Drinks" || ProductCategory == "Bakery") && (Month == 2 || Month == 4 || Month == 6) && (Year == 2010)
Difference of Gross Sales and Target is negative (i.e., underperforming)	(Gross Sales-Target) <= 0
Sales Quantity greater than or equal to 20,000 for the year 2010	SalesQuantity>= 20,000 && Year==2010
Filter for GrossSales greater than 10% of average GrossSales	GrossSales> (0.1 * avg(GrossSales))

5.3 Retrieval Parameters

Various analysis objects, such as graphs, GeoMap, crosstab, and tabular, KPIs are created from a cube, and by default, these objects are fully loaded with data from the cube. But in order to see filtered views, run time parameters – retrieval parameters are provided.

Users can specify the values of the desired retrieval parameters to obtain a filtered view of crosstab, graphs, GeoMap, tabular, and KPIs.

For example, the default view of Sales analysis shows the data for all **States**, **Products,** and **Employees**. But if you want to retrieve data only for certain **States**, **Products,** or **Employees**, you can select the desired values for these parameters. Hence, the Sales analysis data will be retrieved on the basis of values you selected before loading the analysis.



SETTING RETRIEVAL PARAMETER

User **Mark (Executive)** selects **Arizona** as the retrieval parameter for the Column **State**. Another User **Joe (Senior Executive)** selects **Arizona**, **Florida,** and **Ohio** as the retrieval parameters. Both users can get analysis data as per their selected retrieval parameters.

> Note:
> The retrieval parameters will be available for objects such as crosstab, graphs, GeoMap, tabular, and KPIs and will reduce the number of records on the front-end object by filtering cube data.

5.4 Global Variables

The Global variables are defined at the cube level. They can be accessed globally with various expressions and filters for BI objects within Smarten.

For example, users need to view the projection of Growth based on variable % values of Sales amount. For this, a Custom Measure Column (UDDC) **Growth** can be created that would be calculated on the basis of a variable **X** and **GrossSales**. This X can be created as a **Global Variable** and assigned different values at different times to evaluate various scenarios.

Formula for **Growth**: **GrossSales** + (**X***GrossSales**)/100.

Users can change the value of X to see different projections of Growth.

Any change in **X** would be reflected in all analyses where the value of **X** is used through different expressions in filters, Custom Dimension Value **(UDHC)**, Custom Measure Column (UDDC), and retrieval parameters. Hence it saves users from the tedious task of modifying various expressions and filter formula manually and provides simple what-if analysis scenarios.

Once the global variable is defined, it would be accessible throughout the application while applying **Filters**, creating Custom Dimension Value (**UDHC)**, Custom Measure Column (**UDDC),** and **Retrieval Parameters**.

> Note:
> Global variables are available within all BI objects (such as crosstab, graphs, GeoMap, KPIs, dashboards, and tabular) created from a cube. Global variables created for one cube cannot be accessed from within objects created from another cube.

ProductCategory	Arizona	Arkansas	Florida	Ohio	Washington
	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Alcoholic Drinks	6415757	18261094	20796408	8006101	17697251
Bakery	10197878	16804926	21361087	8731632	11325520
Confectionery	363390	1883308	1639046	647845	1250165
Cool Drinks	690706	1041085	961880	322948	564927
Fruit Juices	4249178	8438495	8224081	2722542	7045222
Health Drinks	1456274	4588049	5792470	2061809	4260958
Ice Cream	3938710	5874464	5560304	2677707	4041434
Snacks	577363	1423668	1528300	492191	1780800
Tea	1001994	797487	938838	406455	744943

Global Variable Var1 = 15

Growth = GrossSales + (GrossSales * Var1/100)

ProductCategory	Arizona		Arkansas		Florida		Ohio		Washington	
	GrossSales	Growth	GrossSales	Growth	GrossSales	Growth	GrossSales	Growth	GrossSales	Growth
Alcoholic Drinks	6415757	7378120	18261094	21000258	20796408	23915869	8006101	9207016	17697251	20351839
Bakery	10197878	11727560	16804926	19325665	21361087	24565250	8731632	10041377	11325520	13024348
Confectionery	363390	417898	1883308	2165804	1639046	1884903	647845	745021	1250165	1437690
Cool Drinks	690706	794311	1041085	1197248	961880	1106162	322948	371390	564927	649666
Fruit Juices	4249178	4886554	8438495	9704269	8224081	9457694	2722542	3130923	7045222	8102005
Health Drinks	1456274	1674715	4588049	5276256	5792470	6661341	2061809	2371080	4260958	4900101
Ice Cream	3938710	4529516	5874464	6755633	5560304	6394350	2677707	3079363	4041434	4647649
Snacks	577363	663968	1423668	1637218	1528300	1757545	492191	566020	1780800	2047921
Tea	1001994	1152294	797487	917110	938838	1079664	406455	467423	744943	856685

CUSTOM MEASURE COLUMN (GROWTH) DERIVED USING GLOBAL VARIABLE X (VALUE: 15)

ProductCategory	Arizona	Arkansas	Florida	Ohio	Washington
	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Alcoholic Drinks	6415757	18261094	20796408	8006101	17697251
Bakery	10197878	16804926	21361087	8731632	11325520
Confectionery	363390	1883308	1639046	647845	1250165
Cool Drinks	690706	1041085	961880	322948	564927
Fruit Juices	4249178	8438495	8224081	2722542	7045222
Health Drinks	1456274	4588049	5792470	2061809	4260958
Ice Cream	3938710	5874464	5560304	2677707	4041434
Snacks	577363	1423668	1528300	492191	1780800
Tea	1001994	797487	938838	406455	744943

Global Variable Var1 = 20

Growth = GrossSales + (GrossSales * Var1/100)

ProductCategory	Arizona		Arkansas		Florida		Ohio		Washington	
	GrossSales	Growth	GrossSales	Growth	GrossSales	Growth	GrossSales	Growth	GrossSales	Growth
Alcoholic Drinks	6415757	7698908	18261094	21913312	20796408	24955690	8006101	9607321	17697251	21236702
Bakery	10197878	12237454	16804926	20165911	21361087	25633304	8731632	10477958	11325520	13590624
Confectionery	363390	436068	1883308	2259970	1639046	1966855	647845	777414	1250165	1500198
Cool Drinks	690706	828847	1041085	1249302	961880	1154256	322948	387538	564927	677912
Fruit Juices	4249178	5099013	8438495	10126194	8224081	9868898	2722542	3267050	7045222	8454266
Health Drinks	1456274	1747529	4588049	5505659	5792470	6950965	2061809	2474171	4260958	5113149
Ice Cream	3938710	4726451	5874464	7049356	5560304	6672365	2677707	3213249	4041434	4849720
Snacks	577363	692836	1423668	1708401	1528300	1833960	492191	590629	1780800	2136961
Tea	1001994	1202393	797487	956984	938838	1126606	406455	487746	744943	893932

CUSTOM MEASURE COLUMN (GROWTH) DERIVED FROM MODIFIED VALUE OF GLOBAL VARIABLE X (VALUE: 20)

The value of global variable **X** is modified from **15** to **20**. In the column **Growth**, new value **20** will be taken into consideration, and column values will change accordingly.

> Note:
> The global variables will be available for objects such as crosstab, graphs, GeoMap, tabular, and KPIs.

5.5 Rank

Ranking is the positioning of one value in comparison with other values. It is used to display top *n* or bottom *n* data values. In Smarten, you can rank dimensions on row axis or column. The remaining values that are not part of the ranking are grouped as "others."

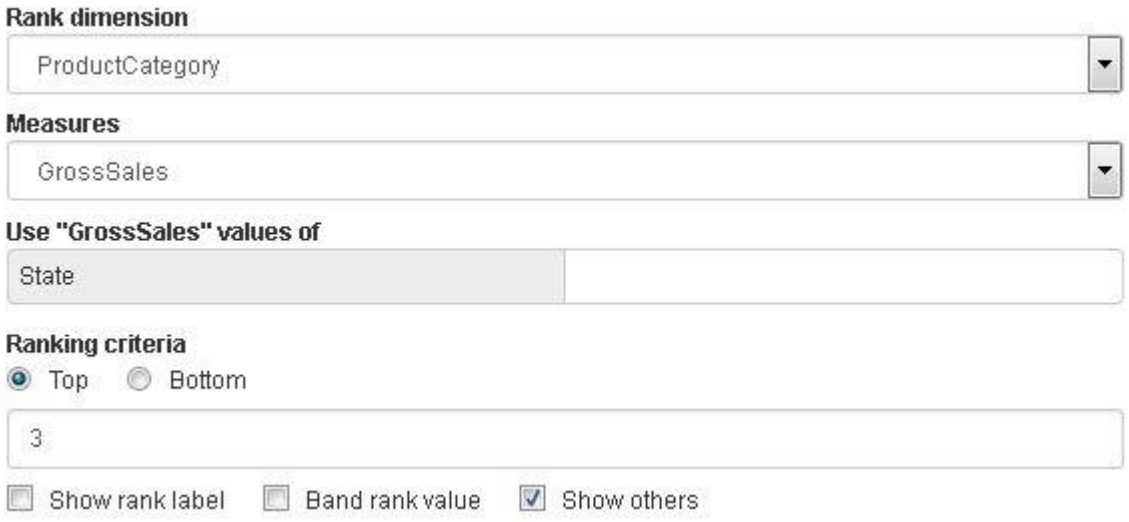
5.5.1 Simple Rank

Please refer to the scenario in the following figure.

	Arizona	Arkansas	Florida	Ohio	Washington	Total
ProductCategory	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Alcoholic Drinks	6415757	18261094	20796408	8006101	17697251	**71176611**
Bakery	10197878	16804926	21361087	8731632	11325520	**68421043**
Confectionary	363390	1883308	1639046	647845	1250165	**5783754**
Cool Drinks	690706	1041085	961880	322948	564927	**3581545**
Fruit Juices	4249178	8438495	8224081	2722542	7045222	**30679517**
Health Drinks	1456274	4588049	5792470	2061809	4260958	**18159560**
Ice Cream	3938710	5874464	5560304	2677707	4041434	**22092618**
Snacks	577363	1423668	1528300	492191	1780800	**5802323**
Tea	1001994	797487	938838	406455	744943	**3889718**
Total	**28891250**	**59112574**	**66802415**	**26069229**	**48711220**	**229586688**

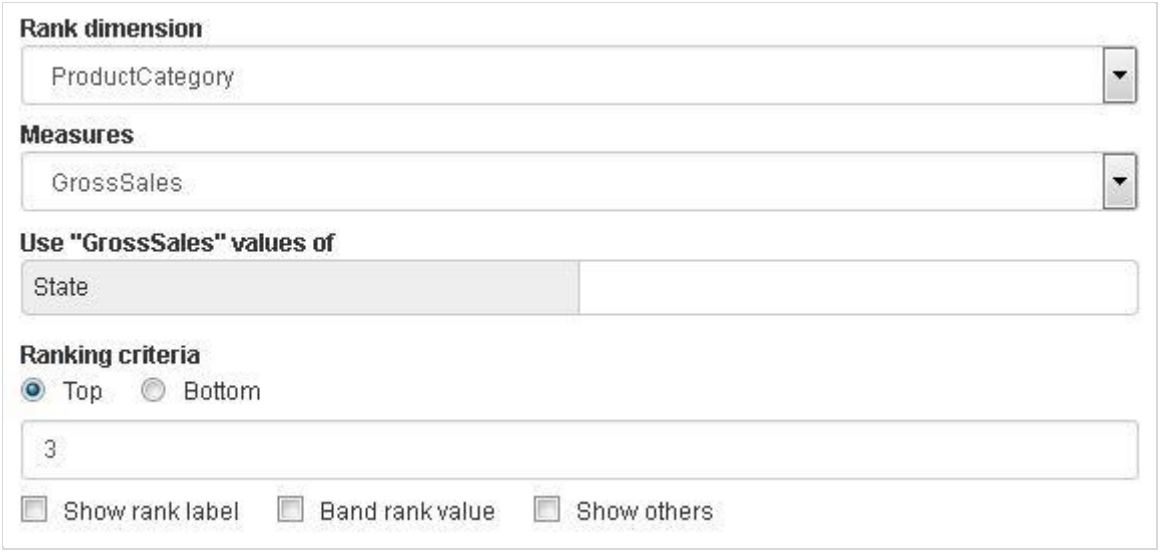
ANALYSIS WITHOUT APPLYING A RANKING

Ranking on the row summary



	Arizona	Arkansas	Florida	Ohio	Washington
ProductCategory	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Alcoholic Drinks	6415757	18261094	20796408	8006101	17697251
Bakery	10197878	16804926	21361087	8731632	11325520
Fruit Juices	4249178	8438495	8224081	2722542	7045222
Others	8028437	15608060	16420838	6608955	12643227

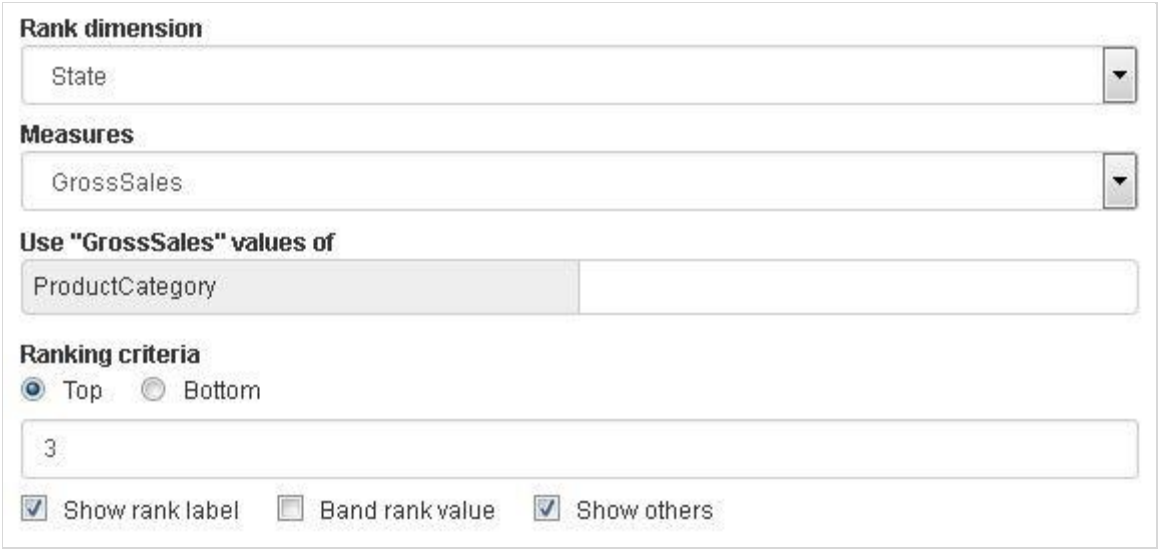
RANK APPLIED ON ROWS WITH SHOW OTHERS (PRODUCT CATEGORIES) FOR GROSSSALES FOR THE TOP 3



	Arizona	Arkansas	Florida	Ohio	Washington
ProductCategory	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Alcoholic Drinks	6415757	18261094	20796408	8006101	17697251
Bakery	10197878	16804926	21361087	8731632	11325520
Fruit Juices	4249178	8438495	8224081	2722542	7045222

RANK APPLIED ON ROWS WITHOUT SHOW OTHERS (PRODUCT CATEGORIES) FOR GROSSSALES FOR THE TOP 3

Ranking on the column summary



	Florida	Arkansas	Washington	Others
ProductCategory	GrossSales	GrossSales	GrossSales	GrossSales
Alcoholic Drinks	20796408	18261094	17697251	14421858
Bakery	21361087	16804926	11325520	18929510
Confectionary	1639046	1883308	1250165	1011234
Cool Drinks	961880	1041085	564927	1013654
Fruit Juices	8224081	8438495	7045222	6971719
Health Drinks	5792470	4588049	4260958	3518083
Ice Cream	5560304	5874464	4041434	6616417
Snacks	1528300	1423668	1780800	1069554
Tea	938838	797487	744943	1408449
Summary_Rank	1	2	3	

RANK APPLIED ON COLUMNS (STATES) FOR GROSSSALES FOR TOP 3

Applying Rank on a particular column

User can also apply ranking on data of a particular dimension. For example, user can apply ranking on GrossSales of ProductCategory for the state of Arizona.

	Arizona	Arizona_Rank	Arkansas	Florida	Ohio	Washington
ProductCategory	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	10197878	1	16804926	21361087	8731632	11325520
Alcoholic Drinks	6415757	2	18261094	20796408	8006101	17697251
Fruit Juices	4249178	3	8438495	8224081	2722542	7045222
Others	8028437		15608060	16420838	6608955	12643227

RANK APPLIED ON THE COLUMN FOR THE STATE OF ARIZONA

5.5.2 Band Rank

Users often need to apply ranking on the extracted data for the top *n* number or the bottom *n* number of the results.

Please refer to the scenario in the following figure.

	Arizona	Arkansas	Florida	Ohio	Washington
ProductCategory	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Alcoholic Drinks	9697878	1041084	961879	322948	17697251
Bakery	10197878	10804925	21361087	11325519	11325520
Confectionary	9697878	1883308	1639046	1250166	211250165
Cool Drinks	6415756	8438494	8224081	272541	32564927
Fruit Juices	1456274	4588049	5792470	2061809	127045222
Health Drinks	6415756	5874463	5560304	2677707	214260958
Ice Cream	577363	1423668	1528300	1780800	174041434
Snacks	6415756	797486	20796407	2061808	181780800
Tea	10197878	18261093	20796407	8006100	39744943

ANALYSIS WITHOUT APPLYING A RANKING

	Arizona	Arizona_Rank	Arkansas	Florida	Ohio	Washington
ProductCategory	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	10197878	1	10804925	21361087	11325519	11325520
Tea	10197878	2	18261093	20796407	8006100	39744943
Alcoholic Drinks	9697878	3	1041084	961879	322948	17697251
Others	30978787		23005471	43540610	12554833	940943506

ANALYSIS WITH CONVENTIONAL RANKING FOR THE TOP 3 PRODUCT CATEGORIES IN THE STATE OF ARIZONA FOR GROSSSALES

In the above crosstab, normal ranking is applied in the ProductCategory column based on the highest GrossSales achieved by various product categories for the state of Arizona.

Apparently, the gross sales figure 10197878 is equal among the product categories Bakery and Tea, but the categories are ranked at positions 1 and 2.

In the case when the Band Ranking is applied to the same crosstab, results would be as below.

Band Ranking:

	Arizona	Arizona_Rank	Arkansas	Florida	Ohio	Washington
ProductCategory	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales	GrossSales
Bakery	10197878	1	10804925	21361087	11325519	11325520
Tea	10197878	1	18261093	20796407	8006100	39744943
Alcoholic Drinks	9697878	2	1041084	961879	322948	17697251
Confectionary	9697878	2	1883308	1639046	1250166	211250165
Cool Drinks	6415756	3	8438494	8224081	272541	32564927
Health Drinks	6415756	3	5874463	5560304	2677707	214260958
Snacks	6415756	3	797486	20796407	2061808	181780800
Others	2033638		6011718	7320770	3842609	301086656

ANALYSIS WITH BAND RANKING FOR

THE TOP 3 PRODUCT CATEGORIES IN THE STATE OF ARIZONA FOR GROSS SALES

The Gross Sales figures for the categories Alcoholic Drinks and Bakery are the same, Confectionary and Cool Drinks have the same sales figure, and the categories Fruit Juices, Ice Cream, and Tea also have the same sales figure.

Applying Band Ranking will give the same rank to the results with the same value as shown in the illustrated example, where Alcoholic Drinks and Bakery are on the Rank 1, Confectionary and Cool Drinks are on the Rank 2, and the categories Fruit Juices, Ice Cream, and Tea are given the Rank 3.

6 KPI

Smarten's simplified KPIs allow the user to make clear, concise definition and tracking of various performance indicators. It helps an organization define performance at various levels and measure actual progress against targeted performance.

Key Performance Indicators are quantifiable measurements of performance at various levels in an organisation. Intuitive colour highlighters ensure that users can see these indicators clearly and accurately present information to the management as well as the team members. Users can further analyse performance with easy-to-use features, such as drill down, drill through, and graphical data mining.

6.1 KPI elements & conventions

Columns	Description
KPI Name	Name of KPI
Period	The Period based on which the KPI is calculated
Frequency	"How often can we measure?" The frequency ranges from Yearly to Hourly time dimensions
Polarity	If increase in value of KPI is beneficial, it is 'High' polarity KPI, whereas if increase is adverse, it is 'Low' polarity. For example, Sales is High polarity KPI, while Return Quantity is Low polarity KPI.
Previous Value	Actual KPI value for previous Period
Actual	Actual KPI value for the current Period
Target	Target KPI value as per definition
Alert	Alert threshold value as per definition. Can be manual or automatic
Warning	Warning threshold value as per definition. Can be manual or automatic
Variance	If the polarity is high, then Variance = Actual – Target If the polarity is low, then Variance = Target – Actual
Variance %	(Variance / Target)*100
Performance %	If the polarity is high, then Performance % = (Actual / Target)*100 If the polarity is low, then Performance % = (Target / Actual)*100
Performance Indicator	There are three indicators in the threshold section
	If the Actual value is less than the Alert value
	If the Actual value is less than the Warning value and greater than the Alert value
	If the Actual value is greater than the Warning value
Trend Indicator	The direction of the arrow indicates the direction of the Performance Trend (up or down compared with previous period). The colour indicates performance for the current period with respect to the target. If the target is not defined, trend indicator will be shown without any colour.
	Performance for the period is less than the target, and performance has declined compared with previous period.
	Performance for the period is less than the target, and performance has improved compared with previous period.
	Performance for the period is higher than the target, and performance has declined compared with previous period.
	Performance for the period is higher than the target, and performance has improved compared with previous period.
	Performance has declined compared with previous period

		and Target is not defined.
	△	Performance has improved compared with previous period and Target is not defined.
Trend chart	Trend chart shows the trend line for performance over a period.	

Trend indicators with example:

#	KPI Performance (November 2010)	KPI Performance (December 2010)	Polarity	Target	Performance for the period	Performance over previous period	Display
1	1900000	1700000	High	1800000	Low	Low	
2	1900000	1700000	High	1600000	High	Low	
3	1700000	1800000	High	1900000	Low	High	
4	1700000	1900000	High	1800000	High	High	
5	1900000	1700000	High	-	High	Low	
6	1700000	1900000	High	-	High	High	

6.1.1 KPI Expressions

Refer to Analytic Functions—Functions used in Custom Measures & Custom Dimension Values Formula

7 Social BI

Smarten doubles up as a social media platform for Augmented Analytics with a dedicated feature called **TeamUp**. TeamUp allows users to not only share the Smarten objects with other users but also rate them and post comments for those objects. Public or private conversations—chat threads created through TeamUp allow users to converse with each other.

The result is **TeamUp Analytics,** which enables the administrators to analyze social BI activities for important insights, such as popular data, reports or dashboards, popular conversation spots, and the overall trend and activities of users within Smarten.

The following features are provided by TeamUp:

- General and cube- and object-specific conversations
- Average rating of each object and cube
- Posting of comments, replies, and Likes for conversation threads
- Referencing other objects in conversation threads
- Marking object-specific conversations as favorites

Reference: **Working with TeamUp**

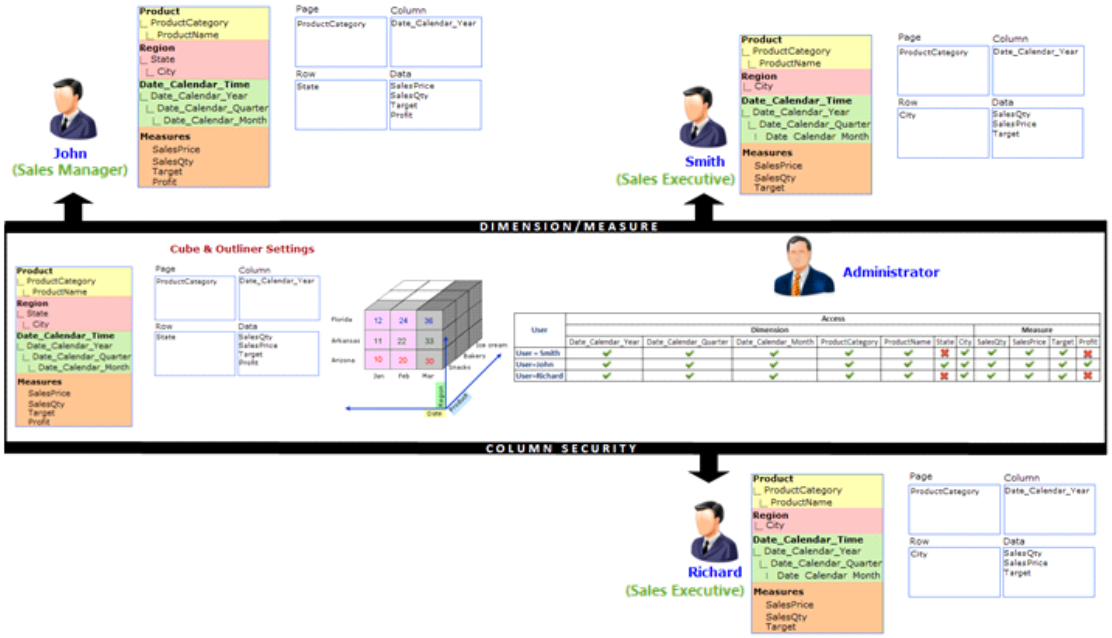
8 Access Rights & Security

8.1 Column-based Access Rights (Column Access Permission)

Dimension / Measure column-based security can be applied for all the cubes. Column Access Permission is granting or restricting access to cube dimension columns or measure columns. All the subordinate objects of a dimension, including hierarchies' levels and members, are also secured.

Column Access permissions are applied at the cube level. These are implemented by giving or limiting access to cube dimensions and measures. Example: A sales manager can view the profits made by the sales representatives. Sales representatives can only view the sales targets assigned to them.

The administrator defines this Dimension / Measure column access security. Roles (users) are then linked to these column securities.
By default, any role (user) has access to all dimension / measure columns in a crosstab/ tabular / graph object to which they have view access. In the example below, **John** is **Sales Manager** with access to Sales Price, Sales Qty, Target, and Profit measure columns. **Smith** and **Richard** are **Sales Executives** with access toSales Price, Sales Qty, and Target measure columns.

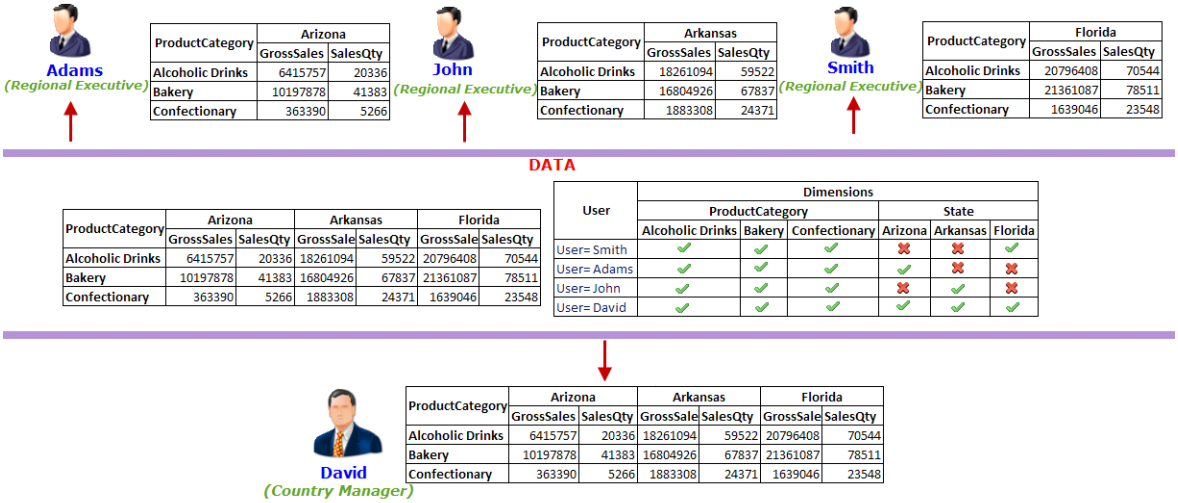


COLUMN ACCESS PERMISSION

8.2 Dimension Value–based Access Rights (Data Access Permissions)

Data value security is restricting data being retrieved by or viewed by the user.
Data value security is applied at the cube level. This is implemented using an expression defined at the cube level with User/Groups permissions. Example: Regional Executives can see only the GrossSales achieved in their assigned territories. A Country Manager can see the overall GrossSales as reported by various sales representatives.

The administrator defines the Dimension Value Security. Roles (users) are then linked to these dimension value securities.
By default, any role (user) has access to all dimension value in crosstab / tabular / graph object to which they have view access. In the scenario below, Adams, John, and Smith are Sales Representatives, whereas David is a Country Manager.



ProductCategory	Arizona	
	GrossSales	SalesQty
Alcoholic Drinks	6415757	20336
Bakery	10197878	41383
Confectionary	363390	5266

ProductCategory	Arkansas	
	GrossSales	SalesQty
Alcoholic Drinks	18261094	59522
Bakery	16804926	67837
Confectionary	1883308	24371

ProductCategory	Florida	
	GrossSales	SalesQty
Alcoholic Drinks	20796408	70544
Bakery	21361087	78511
Confectionary	1639046	23548

ProductCategory	Arizona		Arkansas		Florida	
	GrossSales	SalesQty	GrossSale	SalesQty	GrossSale	SalesQty
Alcoholic Drinks	6415757	20336	18261094	59522	20796408	70544
Bakery	10197878	41383	16804926	67837	21361087	78511
Confectionary	363390	5266	1883308	24371	1639046	23548

User	Dimensions					
	ProductCategory			State		
	Alcoholic Drinks	Bakery	Confectionary	Arizona	Arkansas	Florida
User= Smith	✓	✓	✓	✗	✗	✓
User= Adams	✓	✓	✓	✓	✗	✗
User= John	✓	✓	✓	✗	✓	✗
User= David	✓	✓	✓	✓	✓	✓

ProductCategory	Arizona		Arkansas		Florida	
	GrossSales	SalesQty	GrossSale	SalesQty	GrossSale	SalesQty
Alcoholic Drinks	6415757	20336	18261094	59522	20796408	70544
Bakery	10197878	41383	16804926	67837	21361087	78511
Confectionary	363390	5266	1883308	24371	1639046	23548

DATA ACCESS PERMISSION

9 Delivery & Publishing Agent

Smarten's smart delivery and publishing agent offers a flexible and intelligent information delivery mechanism, bringing corporate information into the hands of the users as and when needed in the format and via the channel they prefer.

DELIVERING AND SCHEDULING ANALYSIS OBJECTS TO USERS ANYTIME, ANYWHERE

Examples:

Schedule On	Scheduler Frequency
One time on 1st January, 2015 at 1 AM	One time: 1st January, 2015 Start time: 1 hour 0 minute
Every night at 12 AM	Daily Start time: 0 hour 0 minute
Every Monday morning at 5 AM	Weekly: Monday Start time: 5 hour 0 minute

10 Product and Support Information

Find more information about Smarten and its features at www.smarten.com
Support: support@smarten.com
Sales: sales@smarten.com
Feedback & Suggestions: support@smarten.com
Support & Knowledgebase Portal: support.smarten.com