LECTURE 13

OLAP CUBES AND MDX

It is necessary to use Analysis services to create OLAP cube. We use Adventure works 2017 database



Of an an an an an an Trans Fuel area

Имя	Состояние	Режим запуска	Использовать для входа	Идентифик	Тип службы
🚯 Службы SQL Server	Работает	Авто	NT Service\MsDtsServer140	22536	
🚯 SQL Full-text Filter D	Работает	Вручную	NT Service\MSSQLFDLauncher	8744	
SQL Server Launchp	Работает	Авто	NT Service\MSSQLLaunchpad	18184	
SQL Server (MSSQLS	Работает	Авто	NT Service\MSSQLSERVER	13032	SQL Server
🚯 Службы SQL Server	Работает	Авто	NT Service\MSSQLServerOLAPService	6728	Analysis Server
🐞 Обозреватель SQL	Работает	Авто	NT AUTHORITY\LOCALSERVICE	4512	
SQL Server PolyBase	Работает	Авто	NT AUTHORITY\NETWORKSERVICE	12988	SQL Server Polybase Data Movement Service
SQL Server PolyBase	Работает	Авто	NT AUTHORITY\NETWORKSERVICE	20352	SQL Server Polybase Engine
Aгент SQL Server (M	Работает	Авто	NT Service\SQLSERVERAGENT	12788	SQL Agent

To create an OLAP cube, you will need to create a Business Intelligence project in Microsoft Visual Studio.
 Install the SSDT tool and you can create this type of project in Visual Studio.

in,	Имя сервера:			
Соединение	VLAD-PC		~ Обнови	ть
	Вход на сервер			
н <i>Ю</i>	Проверка подлинности:	Проверка подлинности Windows		~
Все	Имя пользователя:			
	Пароль:			
	Подключение к базе дан	ных		
	 Выберите или введит AdventureWorks2013 	е имя базы данных: 7		~
	О Прикрепить файл ба:	зы данных:		
	Логическое имя:		Обзор	

• In the data source view, select the desired tables from which the OLAP cube will be built.

vailable objects:				Included objects:	
Vame Dim Scenario (dbo) FactAdditionalInternationalProductDescription (d FactCallCenter (dbo) FactCurrencyRate (dbo) FactFinance (dbo) FactInternetSalesReason (dbo) FactResellerSales (dbo) FactSalesQuota (dbo) FactSurveyResponse (dbo) NewFactCurrencyRate (dbo) VespectiveBuyer (dbo) vAssocSeqLineItems (dbo) vAssocSeqUineItems (dbo) vAssocSeqOrders (dbo) vDMPrep (dbo)	Type Table Table Table Table Table Table Table Table Table Table Table Table View View	~	> <	Name	Type Table Table Table
ter:		Ŧ			Add Related Tables



Dimension Wizard			_		×
Completing the Wizard Type a name for the new dimension, verify the dimension structure, and	d then click Finish t	to save the dimens	sion.	ĺ	
Name:					
DimProduct					
Preview:					
Attributes Product Key English Product Name Weight Size Color					
	< Back	Next >	Finish	Cance	1



Attributes	Hierarchies
City Customer Key English Country Region Name English Education First Name Gender Geography Key Last Name Middle Name Sales Territory Key	 Customers. Customer Key First Name Middle Name Last Name Gender English Education Geography Key City English Country Region Name Sales Territory Key Sales Territory Key English Country Region Name City City



Attributes	Hierarchies
 DimSalesTerritory Sales Territory Group Sales Territory Image Sales Territory Key Sales Territory Region 	 SalesTerritory. Sales Territory Key Sales Territory Country Sales Territory Region Sales Territory Group Sales Territory Image. <new level=""></new>

DEPLOY DIMENSIONS

Q Process Progress	_		×				
🖃 💽 Command							
😑 🕍 Processing Dimension 'DimCustomer' completed.							
Start time: 06.11.2019 4:06:21; End time: 06.11.2019 4:06:23; Duration: 0:00:01							
🕀 🔢 Processing Dimension Attribute '(All)' completed.							
🗉 📱 Processing Dimension Attribute 'City' completed. 563 rows have been read.							
🕀 🔢 Processing Dimension Attribute 'Customer Key' completed. 18485 rows have been read.							
🕢 💀 Processing Hierarchy 'Customers' completed.							
🕢 📱 Processing Dimension Attribute 'English Country Region Name' completed. 7 rows have been read.							
🕢 🔢 Processing Dimension Attribute 'English Education' completed. 6 rows have been read.							
Processing Dimension Attribute 'First Name' completed. 671 rows have been read.							
Processing Dimension Attribute 'Gender' completed. 3 rows have been read.							
🛞 🚺 Processing Dimension Attribute 'Geography Key' completed. 656 rows have been read.							
Processing Dimension Attribute 'Last Name' completed. 376 rows have been read.							
Processing Dimension Attribute 'Middle Name' completed. 45 rows have been read.							
Processing Dimension Attribute 'Sales Territory Key' completed. 12 rows have been read.							
Status:							
			_				
V Process succeeded.							
Ston Reprocess View Details		Conv					
Stop Reprocess View Details		сору					
Clo	se	Help					
			-				

CREATING THE CUBE

• After deploying dimensions, it is necessary to create the cube itself. We choose measures from one table

🚱 Cube Wizard	-		×
Select Measure Group Tables Select a data source view or diagram and then select the tables that will be used for measure groups.	r		
Data source view:			
AdventureWorksDW2017			\sim
Measure group tables:		Suggest	
□ Im Product □ Im Geography □ Im Customer □ Im FactInternetSales □ Im DimDate □ Im Sales Territory			
< Back Next > Finish >	>>	Cance	el

CREATING THE CUBE



OLAP CUBE

After the deployment of the cube, connect to Microsoft SQL Analysis Services



MDX DIMENSIONALITY

MDX QUERIES ARRANGE CUBE DIMENSIONS ON THE REPRESENTATION DIMENSIONS (HEREAFTER REFERRED AS 'AXIS' TO AVOID CONFUSION)

E.G. RETRIEVE PLANNED GSV MEASURE FOR ACCOUNTS E4098, E4398.



} ON ROWS FROM [CUBE REPORTING]

[ACCOUNT].[ACCOUNT CODE].[E4098:47] ,[ACCOUNT].[ACCOUNT CODE].[E4398:47]

} ON COLUMNS,

SELECT

[MEASURES].[PLANNED GSV]

0	Columns
1	Rows
2	Pages
3	Chapters
4	Sections

MDX provides names for each axis (till 4)

E.G. RETRIEVE PLANNED GSV MEASURE FOR ACCOUNTS E4098, E4398.

MDX QUERIES ARRANGE CUBE DIMENSIONS ON THE REPRESENTATION DIMENSIONS (HEREAFTER REFERRED AS 'AXIS' TO AVOID CONFUSION)

MDX QUERY – AXIS FRAMEWORK

```
ON AXIS(0),
{
[MEASURES].[PLANNED GSV] } ON AXIS(1)
FROM [CUBE REPORTING]
```

```
{

[ACCOUNT].[ACCOUNT CODE].[E4098:47]

,[ACCOUNT].[ACCOUNT CODE].[E4398:47]

}
```

```
SELECT
```

```
MDX QUERIES PRIMARILY DEFINE AXIS'S

SELECT

{
SOMETHING
} ON AXIS(0),
{
Planned GSV
SOMETHING ELSE
} ON AXIS(1),
FROM [CUBE NAME]
```

E4098:47

E4398:47

428007.56 49183489.29

MDX QUERY – AXIS FRAMEWORK

MDX QUERY – AXIS FRAMEWORK



ON AXIS(0),



-> MANY WAYS ARE THERE TO SPECIFY A MEMBER.

A TUPLE IS A COMBINATION OF MEMBERS FROM ONE OR MORE DIMENSIONS

-> NOT MORE THAN I MEMBER FROM A DIMENSION (SAME RULE AS CO-ORDINATE GEOMETRY)*

MDX QUERY - TUPLE AND SETS

TUPLES



Understanding tuples are key to thinking in MDX.

We will stop here till all tuple related queries are clarified.

Understanding tuple

Best analogy - coordinate geometry *

2D –space

Tuple of the form $(x_1, y_1) e.g. (3, 4)$

Tuple like (x_1, y_1, y_2) or (x_1, x_2, y_2) are invalid

Now apply same concept to *n*-dimensional cube

What are valid tuples ?

(time.year.[2011], Product.brand.[B1])

(time.year.[2011], Product.brand.[B1], time.year.[2010])

(time.year.[2011], Product.brand.[B1], Geo.[India])

(time.year.[2011], Product.brand.[B1], Product.brand.[B2])

* This analogy holds good except for hierarchies. Hierarchies in cube space can be considered as dimensions in Co-ordinate geometry

MDX QUERY – TUPLE AND SETS

SOMETHING TO REMEMBER

- I. THIS IS A SET CONTAINING A SINGLE TUPLE
- [ACCOUNT].[ACCOUNT CODE].[E4098:47]

2. THIS IS A SINGLE TUPLE

[ACCOUNT].[ACCOUNT CODE].[E4098:47]

BOTH ARE DIFFERENT THINGS CONCEPTUALLY AND PROGRAMMATICALLY (THOUGH SSAS OVERLOOKS IT).

ON I FROM [CUBE REPORTING]

CN 0, {([TIME].[2010 JAN]:[TIME].[2010 DEC]) }

{[ACCOUNT].[E4398:47]}

SELECT

([TIME].[2010 JAN]:[TIME].[2010 DEC]) - ALL MONTHS FROM JAN 2010 TO DEC 2010

	ELEMENTS SEPARATED BY ' '		E4398:47
		2010 JAN	338733
•	MULTI PART IDENTIFIER '.'	2010 FEB	601503
•	CAN BE OF THE FORM	2010 MAR	394195
		2010 APR	437042
	DIMENSION.HIERARCH I.LEVEL.MEMBER	2010 MAY	578184
	[TIME].[TIME HIERARCHY].[MONTH].&[421]	2010 JUN	715195
		2010 JUL	707481
		2010 AUG	1007624
	DIMENSION.MEMBER	2010 SEP	550235
ГТ	IMF1 [2010 JAN]	2010 OCT	743799
		2010 NOV	604130
	COLON ': ' USED TO REPRESENT A SERIES OF MEMBERS	2010 DEC	223204
		4	

A CLOSER LOOK AT TUPLE

MDX QUERY – TUPLE CONSTRUCTS

SELECT { TIME.[2010 QUARTER 1].CHILDREN }ON 0, {[ACCOUNT].[E4398:47]} ON 1 FROM [CUBE REPORTING]

		_
	E4398:47	
2010 JAN	338733	
2010 FEB	601503	
2010 MAR	394195	

-- USED TO EXPRESS THE CHILDREN OF A MEMBER

E.G. AS PER TIME HIERARCHY – MONTH IS THE CHILD OF QUARTER

.CHILDREN

MDX QUERY – THE CHILDREN FUNCTION

MDX QUERY – THE DESCENDATION FUNCTION

DESCENDANTS (*MEMBER* [, [*LEVEL*] [, *FLAG*]])

-- USED TO EXPRESS THE DESCENDANT OF A MEMBER AT A LEVEL

FLAG ALLOWS DISPLAY

[CUBE REPORTING]

	2010 JAN	338733	
SELECT {[ACCOUNT] [F4398:47]}	2010 FEB	601503	
	2010 MAR	394195	
ON 0,	2010 APR	437042	
1	2010 MAY	578184	
	2010 JUN	715195	
DESCENDANTS(TIME.[2010],MONTH,SELF)	2010 JUL	707481	
	2010 AUG	1007624	
	2010 SEP	550235	
FROM [CUBE REPORTING]	2010 OCT	743799	
	2010 NOV	604130	
	2010 DEC	223204	
SELECT {[ACCOUNT].[E4398:47]}			
ON 0.			
{			
DESCENDANTS(TIME.[2010],MONTH,SELF_AND_BEFORE)			
NON I			

E4398:47
6901325
3064852
1334431
338733
601503
394195
1730421
437042
578184
715195
3836473
2265340
707481
1007624
550235
1571133
743799
604130
223204
223204

E 4398:47

ASSIGNMENT - I

ASSUME THE FOLLOWING DIMENSIONS : TIME : YEAR <- QUARTER <- MONTH <- DAY PRODUCT : DOLLAR SALES, UNIT SALES GEO : COUNTRY <- STATE <- CITY



```
TIME.[2010 JAN] : TIME.[2010 MAR]
}
<u>ON</u> 0,
[MEASURES].[PLANNED GSV]
}ON I
FROM
```

	E4098:47	E4098:47	E 4098:47	E4400:47	E4400:47	E4400:47
	2010 JAN	2010 FEB	2010 MAR	2010 JAN	2010 FEB	2010 MAR
Planned GSV	31594.08	38569.92	38481.04	1481540	3209700	2112100

* -- CROSS JOIN

(ACCOUNT.[ACCOUNT CODE].[E4098:47]), (ACCOUNT.[ACCOUNT CODE].[E4400:47])

SELECT

PLANNED GSV FOR 2 ACCOUNTS FOR 3 MONTHS --

 $A \times B = \{ (1, X), (2, X), (3, X), (1, Y), (2, Y), (3, Y), \}$

CROSS JOIN AXB

 $\mathsf{B} = \{\mathsf{X},\mathsf{Y}\}$

A = {1,2,3}

CONCEPT :

CROSS JOINS

TWO SETS - A ,B

FILTER

CONCEPT :

[CUBE REPORTING]

FILTER (SET, EXPRESSION) .members gives all members of that level ACCOUNTS WHICH HAVE MORE THAN GSV SELECT FILTER (ACCOUNT.[ACCOUNT CODE]. EMBERS), [MEASURES].[PLANNED GSV] > 10000000 ON 0, [MEASURES].[PLANNED GSV] All DTTot3:47 E4398:47 E4400:47 E4401:47 E4414:47 E4982:47 }ON I 886166048.7521 13611790 Planned GSV 49183489.29 11623040 637964831.993 35195728.33 76271218.992 FROM

FROM

}ON I

[MEASURES].[PLANNED GSV]

		All	All	All	DTTot3:47	DTTot3:47	DTTot3:47	E 4398:47	E 4398:47	E4398:47	E4400:47	E4400:47
0,		2010 JAN	2010 FEB	2010 MAR	2010 JAN	2010 FEB	2010 MAR	2010 JAN	2010 FEB	2010 MAR	2010 JAN	2010 FEB
	Planned GSV	59230786.49708	83152066.09	70867123.5314	(null)	(null)	(null)	3593022.88	5276412.38	2256812.5	1481540	3209700

TIME.[2010 JAN] :TIME.[2010 MAR]

)* -- CROSS JOIN

([MEASURES].[PLANNED GSV]) > 10000000

(ACCOUNT.[ACCOUNT CODE].MEMBERS),

(

FILTER

ELECT

WHAT IS THE MEANING OF THIS?

FILTER

FROM

}ON I

[MEASURES].[PLANNED GSV]

ON 0,

)* -- CROSS JOIN

([MEASURES].[PLANNED GSV] ,<u>TIME.[2010 JAN]</u>) > 10000000

(ACCOUNT.[ACCOUNT CODE].MEMBERS),

FILTER

SELECT

AND THIS?

FILTER

	All	All	All	E4401:47	E4401:47	E4401:47
	2010 JAN	2010 FEB	2010 MAR	2010 JAN	2010 FEB	2010 MAR
Planned GSV	59230786.49708	83152066.09	70867123.5314	49120998.089	71687757.826	61112877.531

	All	All	All	E4401:47	E4401:47	E4401:47	Γ
	2010 JAN	2010 FEB	2010 MAR	2010 JAN	2010 FEB	2010 MAR	
lanned GSV	59230786,49708	83152066.09	70867123.5314	49120998.089	71687757.826	61112877.531	

TIME.[2010 JAN] :TIME.[2010 MAR]

Tuple reference is one of the powerful concepts in MDX

SELECT													
NON EMPTY (
ORDER													
(
(ACCOUNT.[ACCOUNT CODE].MEMBE	RS),												
([MEASURES].[PLANNED GSV])													
)													
)													
ON 0,													
ſ		All	E4616:47	E4612:47	E4598:47	E4602:47	E4547:47	E4626:47	E4546:47	E4533:47	E4627:47	E4548:47	E4619:47
1	Planned GSV	886166048.7521	13408	20880	23360	40493.3	96924	115602.5	126750	144464.7	148253.5	183610	202138
[MEASURES].[PLANNED GSV]													
}ON I													
FROM													

ORDER (SET I, EXPRESSION [, ASC | DESC | BASC | BDESC])

ACCOUNTS ORDERED BY GSV

<u>ORDER</u>

CONCEPT :

<u>ON</u> 0, All DTTot3:47 E4098:47 E4398:47 E4400:47 E4401:47 E4402:47 E4407:47 E4413:47 E4414:47 [MEASURES].[PLANNED GSV] Planned GSV 6288527.17226 886166048.7521 13611790 428007.56 49183489.29 11623040 637964831.993 1974200 8884200 35195728.33 }ON I [CUBE REPORTING]

(ACCOUNT.[ACCOUNT CODE].MEMBERS), ([MEASURES].[PLANNED GSV],TIME.[2011 JAN])

,

ORDER

NON EMPTY (

SELECT

ACCOUNTS ORDERED BY ???





(CREATE SET [CUBE REPORTING].[TEST ACCOUNTS] { [ACCOUNT].[ACCOUNT CODE].[E1373:47], [ACCOUNT].[ACCOUNT CODE].[E40301:47]

ALSO POSSIBLE TO CREATE PERSISTENT NAMED SETS

Planned GSV

31173548.79884

41130.25381

E1373:47

E40301:47

ROM [CUBE REPORTING]

} ON I

[GREAT ACCOUNTS]

NON EMPTY

<mark>DN</mark> 0,

[MEASURES].[PLANNED GSV]

SELECT

{ [ACCOUNT].[ACCOUNT CODE].[E1373:47], [ACCOUNT].[ACCOUNT CODE].[E40301:47] }

WITH SET [GREAT ACCOUNTS] AS

NAMED SETS : EASE OF REFERENCE

CALCULATED MEMBERS : THE POWER OF MDX !!!!

I. SIMPLE CALCULATED MEMBERS



Q1, 2005

Q2, 2005

Dollar Sales

Unit Sales Average Sales price

20

15

100

120

[MEASURES].[AVG SALES PRICE]

[MEASURES].[DOLLAR SALES] / [MEASURES].[UNIT SALES]

SELECT

{ [MEASURES].[DOLLAR SALES] [MEASURES].[UNIT SALES] , [MEASURES].[AVG SALES PRICE] }

ON COLUMNS,

{ [TIME].[Q1, 2005] , [TIME].[Q2, 2005]

SALES

CALCULATED MEMBERS : THE POWER OF MDX !!!!

2. CALCULATED MEMBERS OF MEDIUM COMPLEXITY

FIND THE QUARTER ON QUARTER GROWTH FOR DOLLAR SALES AND UNIT SALES FOR THE QUARTERS 2005 Q2. GROWTH IN DOLLAR SALES = 2005 Q2 DOLLAR SALES - 2005 Q1 DOLLAR SALES GROWTH IN UNIT SALES = 2005 Q2 UNIT SALES - 2005 Q1 UNIT SALES

WITH MEMBER [TIME].[QI TO Q2 GROWTH] A [TIME].[Q2, 2005] - [TIME].[Q1, 2005] SELECT { [MEASURES].[DOLLAR SALES]	\$		How does th care of both subtractions	iis take	e
, [MEASURES].[UNIT SALES] } ON COLUMNS, { [TIME].[Q1, 2005] [TIME].[Q2, 2005]		B4	▼ (°	f _x	=B3-B2
, [TIME].[Q1 TO Q2 GROWTH] }		А	В		С
ON ROWS	1		Dollar Sales	Unit	Sales
FROM SALES	2	Q1, 2005	100		ļ
WHERE ([CUSTOMER].[MA])	3	Q2, 2005	120		1
	4	Q1 to Q2 Growth	20		
	-				

[SALES]

ON ROWS

ON COLUMNS, { [TIME].[Q1, 2005], [TIME].[Q2, 2005], [TIME].[Q1 TO Q2 GROWTH] }

[MEASURES].[AVG SALES PRICE]

{ [MEASURES].[DOLLAR SALES], [MEASURES].[UNIT SALES],

[TIME].[Q2, 2005] - [TIME].[Q1, 2005]

MEMBER [TIME].[Q1 TO Q2 GROWTH] AS

[MEASURES].[DOLLAR SALES] / [MEASURES].[UNIT SALES]

MEMBER [MEASURES].[AVG SALES PRICE] AS

WITH

PRICES

PRECEDENCE RESOLUTIONS

CALCULATED MEMBERS : THE POWER OF MDX !!!!

COMBINING PREVIOUS TWO PROBLEMS, WRITE MDX TO CALCULATE QI TO Q2 GROWTH IN AVERAGE SALES

	D4		Jx	= B4/C4	
	^	D		<u> </u>	D
	A	В		C	D
1		Dollar Sales	Unit :	Sales	Average Sales price
2	Q1, 2005	100		5	20
3	Q2, 2005	120		8	15
4	Q1 to Q2 Growth	20		3	6.666666667

0

FROM [SALES]

ON COLUMNS, { [TIME].[Q1, 2005], [TIME].[Q2, 2005], [TIME].[Q1 TO Q2 GROWTH] }

[MEASURES].[AVG SALES PRICE]

{ [MEASURES].[DOLLAR SALES], [MEASURES].[UNIT SALES],

SELECT

WITH

SOLVE ORDER = I

[TIME].[Q2, 2005] - [TIME].[Q1, 2005],

[TIME].[Q1 TO Q2 GROWTH]

SOLVE_ORDER = 0

MEMBER [MEASURES].[AVG SALES PRICE] AS [MEASURES].[DOLLAR SALES] / [MEASURES].[UNIT SALES],

4	Q1 to Q2 Growth	20		3	6.666666667
-					
	G11	- (0	f_{x}		
	А	В	С		D
1		Dollar Sales	Unit Sales		Average Sales price
2	Q1, 2005	100		5	20
3	Q2, 2005	120		8	15
- U					

	А	В	С	D
1		Dollar Sales	Unit Sales	Average Sales price
2	Q1, 2005	100	5	20
3	Q2, 2005	120	8	15
4	Q1 to Q2 Growth	20	3	6.666666667
-				

▼ (@)

D4

*f*_x = B4/C4

PRECEDENCE RESOLUTIONS

COMBINING PREVIOUS TWO PROBLEMS, WRITE MDX TO CALCULATE QI TO Q2 GROWTH IN AVERAGE SALES PRICES

CALCULATED MEMBERS : THE POWER OF MDX !!!!

CALCULATED MEMBERS : THE POWER OF MDX !!!!

WRITE AN MDX TO GET THE FOLLOWING RESULT

	Sale Amount	Total Cost	Profit	Perce	ntag	ge N	/largin
Actual				//	\square		//
Planned				//	\square		//
Amount of Variance				//	7	/ /	77
Percentage of Variance	/////	/////	//////	/	\sum	$\overline{)}$	$\overline{)}$

MEASURES -> MEASURES.[SALES AMOUNT], MEASURES.[TOTAL COST]

```
DIMENSIONS -> PHASE.ACTUAL , PHASE.PLANNED
```

USE NORMAL FORMULAS FOR PROFIT, PERCENTAGE MARGIN, AMOUNT OF VARIANCE, PERCENTAGE OF VARIANCE.