Power BI – Day 6

April 2018

Insights & Data

Capgemini

Power BI Security

Capgemini

Power BI Security

Power BI Service is built on Azure

Based on 2 Clusters – Web Front End (**WFE** – for initial authentication) and the Back End

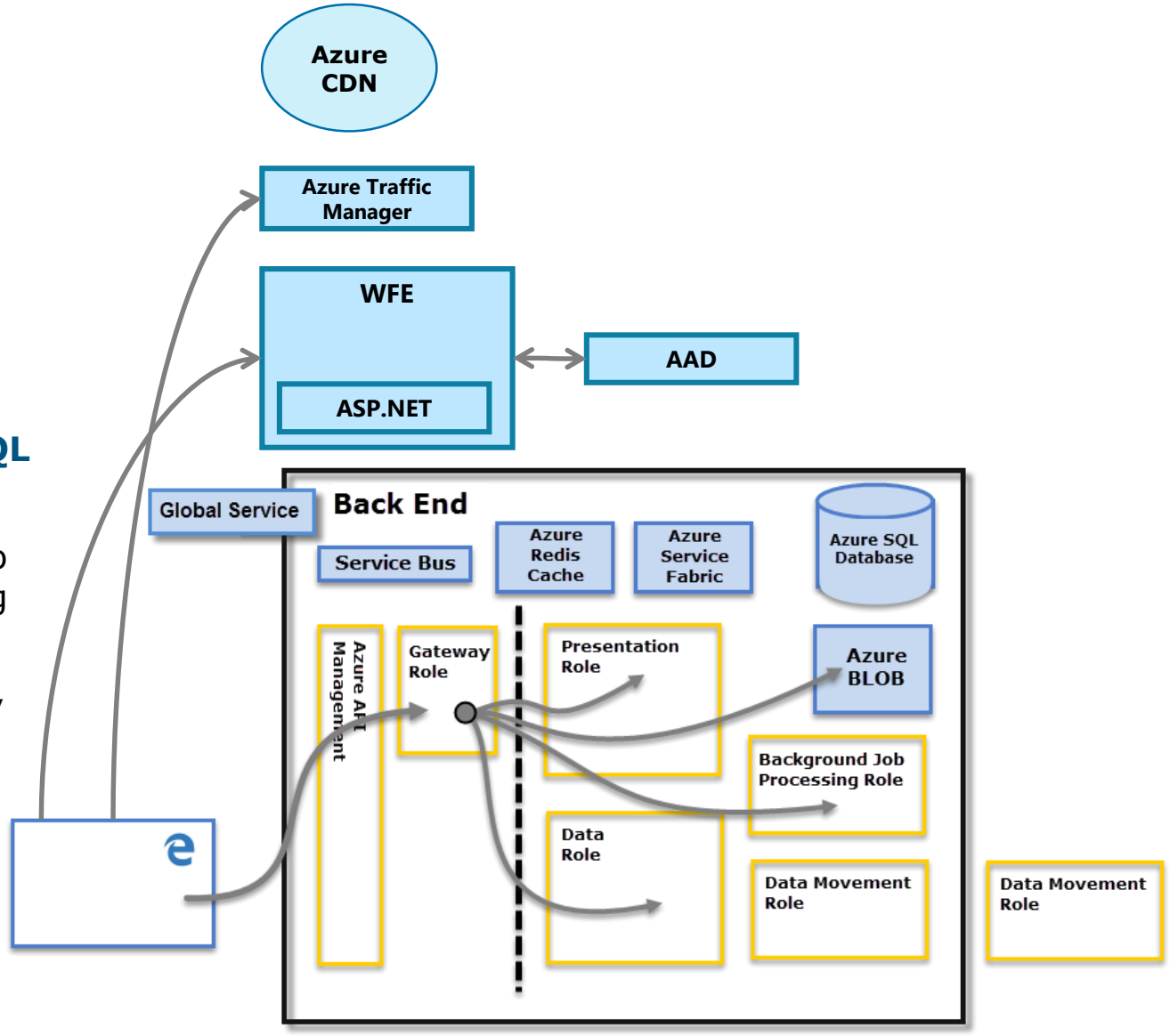
Uses Azure Active Directory (**AAD**) to store/manage user interactions.

Uses Azure Content Delivery Network (CDN) for effective delivery of static files and content to users

Data Storage Security - Uses **Azure BLOB** and **Azure SQL DB** for **data storage** and **metadata** respectively.

Uses **Azure Traffic Manager** (**ATM**) to direct user traffic to nearest datacenter using the DNS entry of the client making the connection

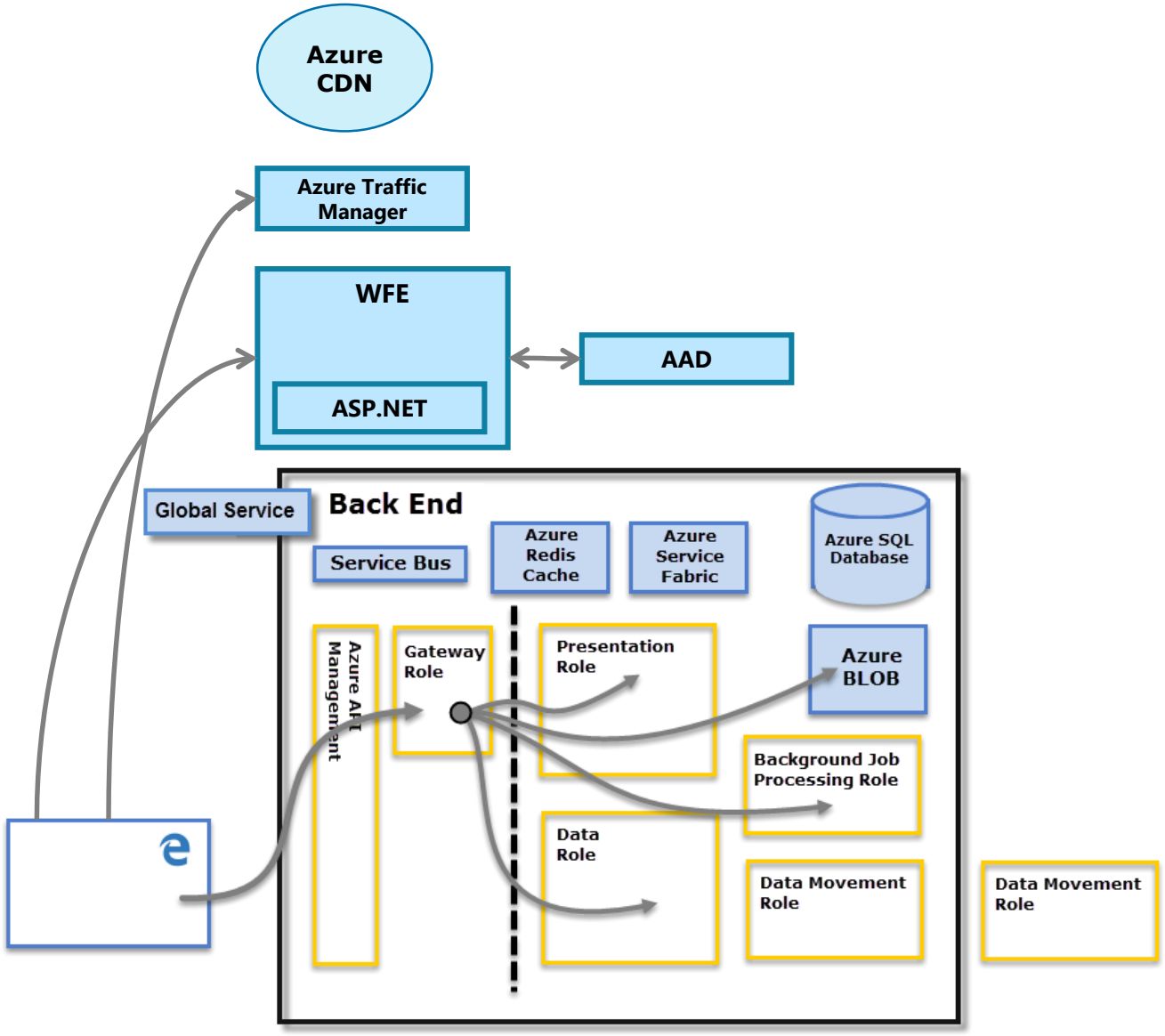
Only APIM (**Azure API Management Role**) and **Gateway Role** are accessible via public internet.



Power BI Security …. Contd..

User Authentication: Power BI uses login email id as effective username while connecting to data. The *effective name* is mapped to *User Principle Name* (UPN) and resolved to the associated Windows domain account, against which authentication is applied.

Data & Service Security: Data is encrypted at rest and in process ().



Row Level Security

Row Level Security (RLS) is used by Power BI to restrict data access for users.

Configure RLS on data models imported into Power BI or on datasets using Direct Query Mode.

How do I apply Row Level Security?

1. Define Roles and rules within Power BI Desktop
2. Validate the role in Power BI Desktop
3. Publish the report and use Power BI Service to manage security on the datasets.

DAX Introduction

Capgemini

Introduction… What is DAX?

- **D**ata **A**nalysis E**x**pression
- Language of Power BI, Power Pivot, SSAS Tabular
- Neither SQL nor MDX
- Functional language
- Simple & Straightforward yet difficult to fully use and understand

DAX Types

Numeric Types

- Integer
- Decimal (floating)
- Date (Datetime) - # of days after 30 Dec 1899
- Currency (money)
- TRUE / FALSE (Boolean)

Other Types

- String
- Binary Object

DAX Calculation Types

Calculated Columns

- Represents a single value per row
- Computed at time of creation/refresh
- Results are materialized and stored with the table
- Attached to a specific table
- Normally can only see the row they exist in
- Can be used in filters or values/results areas

Calculated Measures

- Represents a single value per data model
- Computed at run time
- Results are dynamic, based on filters

  (This is called the filter context)
- Not attached to any table
- Sees all the data at once

DAX Functions

Aggregation	Counting	Logical	Information	Text
❑ SUM	❑ COUNT	❑ AND	❑ ISBLANK	❑ CONCATENTATE
❑ AVERAGE	❑ COUNTA	❑ OR	❑ ISNUMBER	❑ REPLACE
❑ MIN	❑ COUNTBLANK	❑ NOT	❑ ISTEXT	❑ SEARCH
❑ MAX	❑ COUNTROWS	❑ IF	❑ ISNONTEXT	❑ UPPER
❑ SUMX (and other X functions)	❑ DISTINCTCOUNT	❑ IFERROR	❑ ISERROR	❑ FIXED
			❑ PATH – *manage parent child hierarchy*	

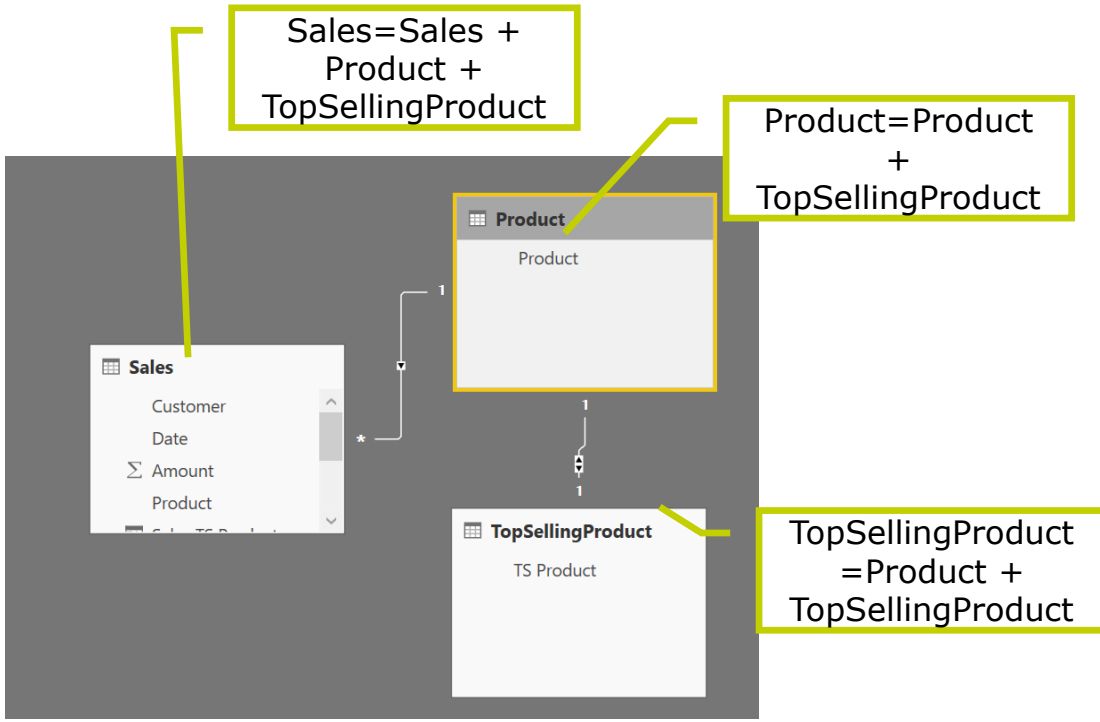
DAX Function…

Date

- DATE
- HOUR
- NOW
- EOMONTH
- WEEKDAY
- YEAR

Expanded Tables in DAX

- Expanded Tables = Base Table +Columns of tables linked to Base Table using ***:1** or **1:1** relationship
- Similar to FULL / LEFT OUTER JOIN
- Relationship Concept




```
Sales TS Product = RELATED(TopSellingProduct[TS Product])
```

Variables in DAX Expressions

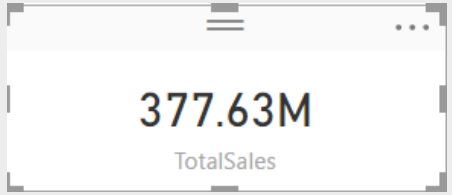
- Extremely powerful
- It can be defined anywhere in the DAX expression.
- Referring the variable again is from the cache model.

```
VariableDemoSalesMargin =

VAR TotalRevenue = SUM(bi_salesFact[Revenue])
RETURN
IF ([TotalSales] > 10000, [TotalSales]*0.2,
TotalRevenue*0.1
)
```

Evaluation Context

For which rows is the TotalSales evaluated?

```
TotalSales = SUM(bi_salesFact[Revenue])
```



Category	TotalSales
Urban	312,960,273.88
Rural	41,181,671.21
Mix	16,063,198.46
Youth	7,420,366.22
Total	**377,625,509.78**

Value computed depends on the context

Row Context & Filter Context

Row Context

RevenueMargin = SUMX(bi_salesFact, bi_salesFact[Revenue]-bi_salesFact[Revenue]*0.1)

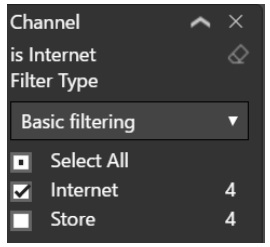
Defined by calculated column definition & various row iterator functions.

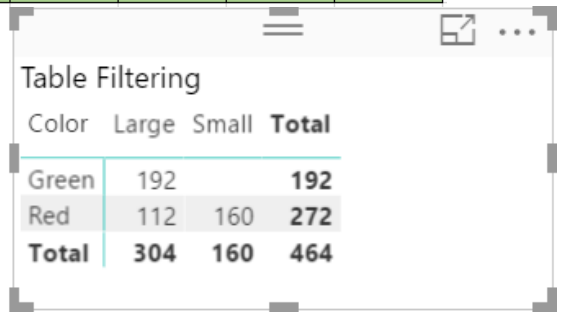
Filter Context

Set of filters applied to data model before evaluating DAX expression.

CalcAmount = SUMX(
FILTER(Orders, Orders[Price] > 1), Orders[Amount])

City	Channel	Color	Size	Quantity	Price
Paris	Store	Red	Large	1	15
Paris	Store	Red	Small	2	13
Toronto	Store	Green	Large	4	11
New York	Store	Green	Small	8	9
	Internet	Red	Large	16	7
	Internet	Red	Small	32	5
	Internet	Green	Large	64	3
	Internet	Green	Small	120	1





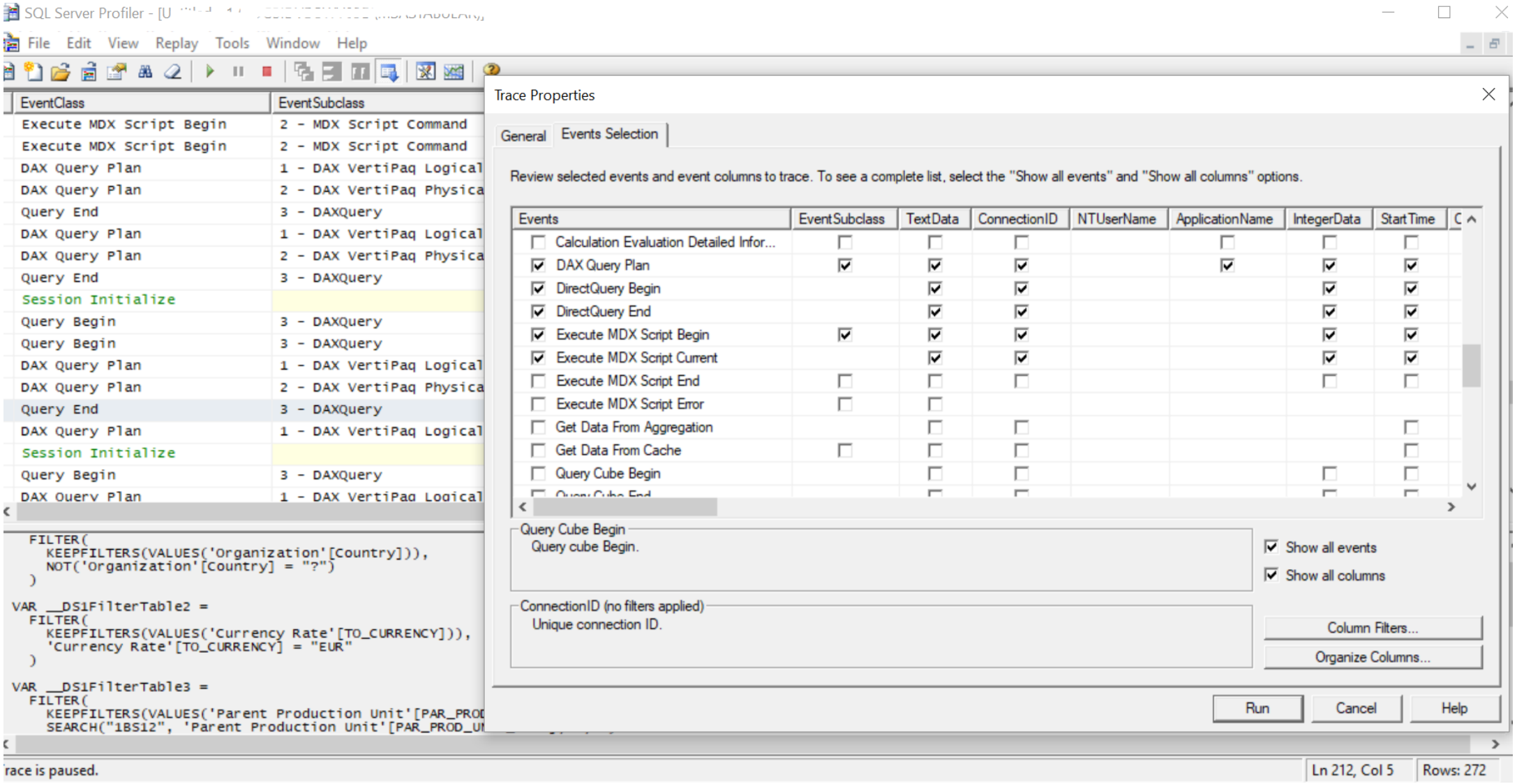
CALCULATE

- Modify filter context
- Filter and then apply Color = Green to get a table on which SUM can be applied

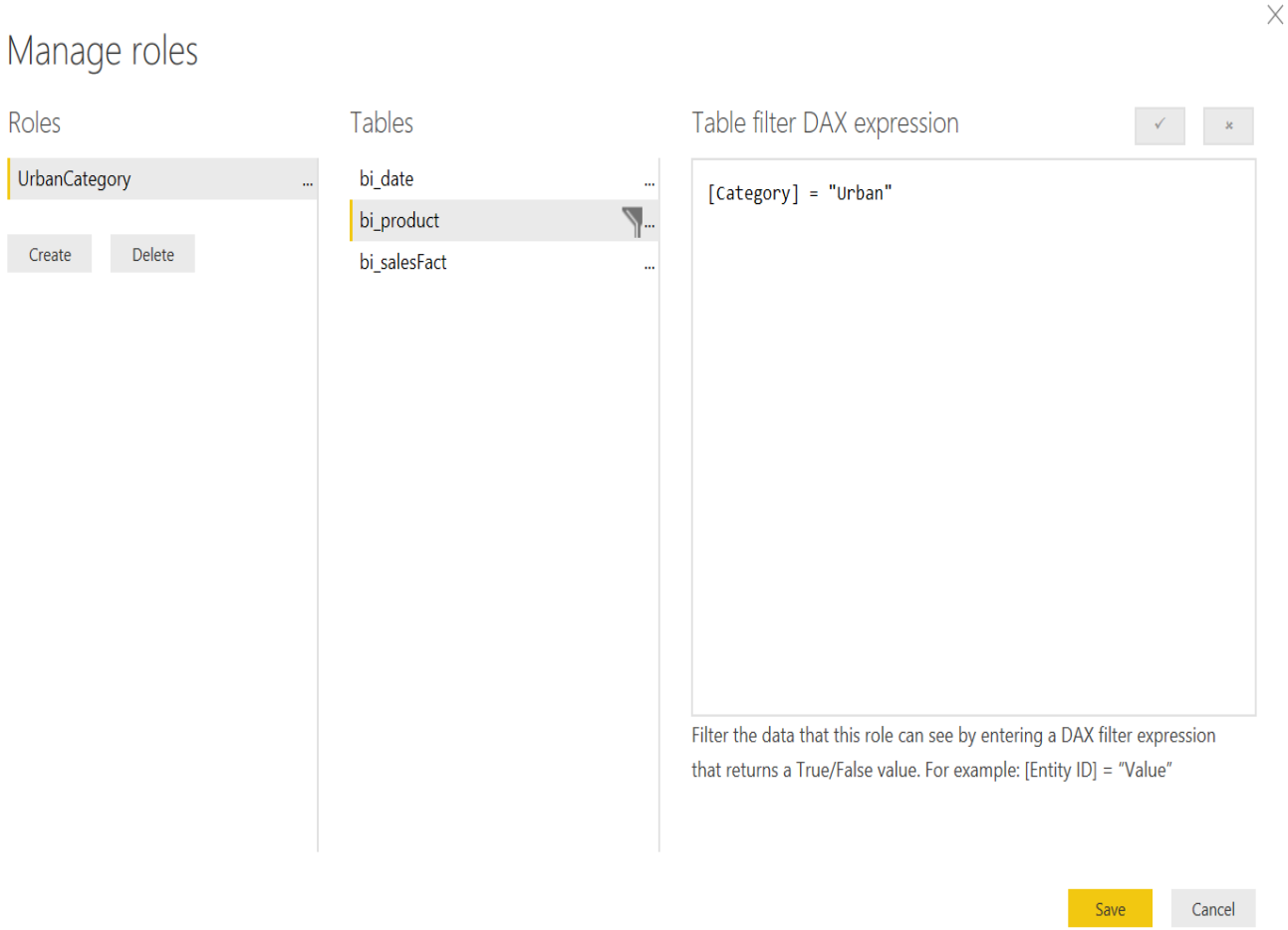
```
GreenQuantity := CALCULATE(SUM(Sales[Quantity]),

Sales[Color]="Green")
```

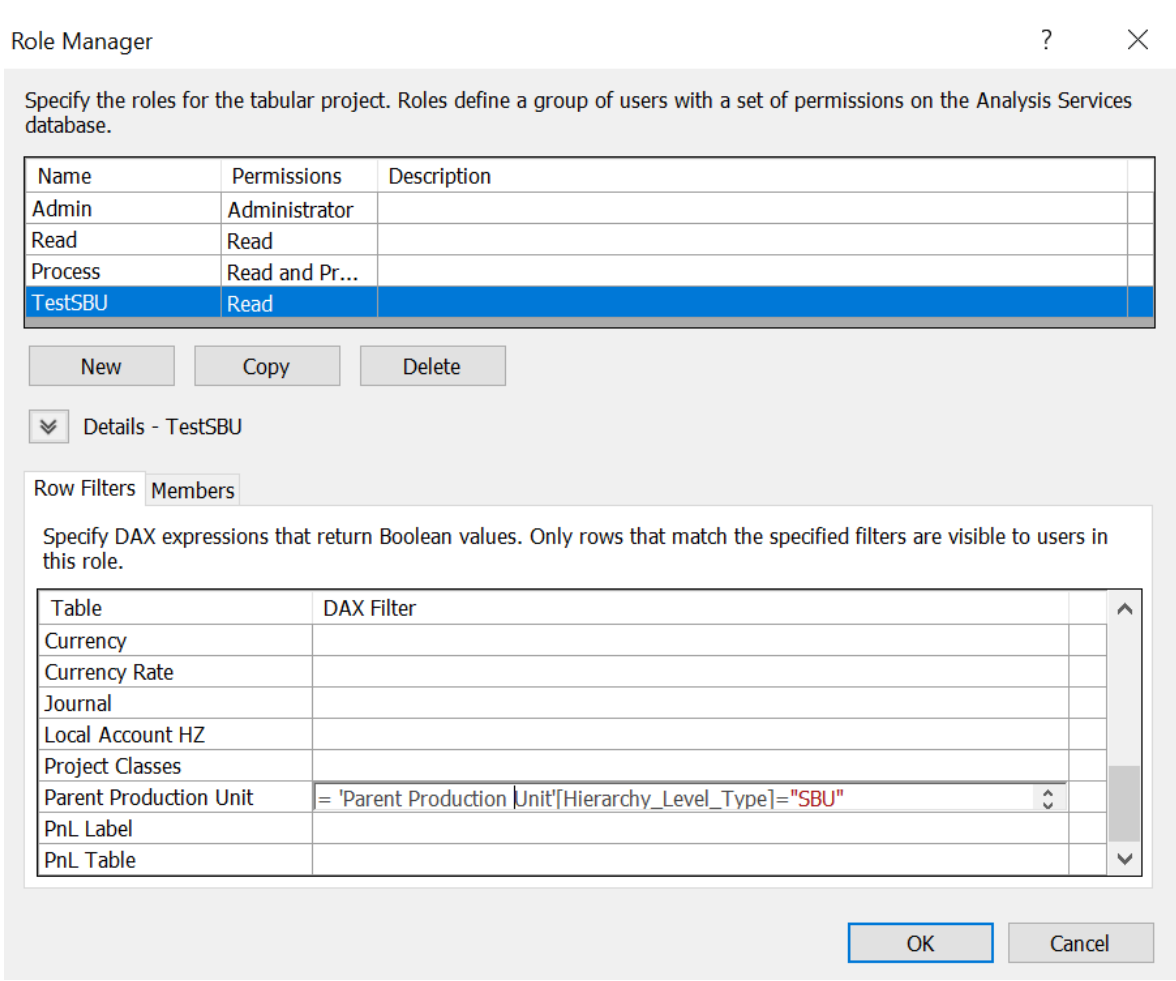
```
GreenQuantity := CALCULATE(SUM(Sales[Quantity]),
                FILTER(
                                ALL (Sales[Color]),
                                Sales[Color]="Green“
                                )
)
```

SQL Profiler



Add Security to Model using DAX

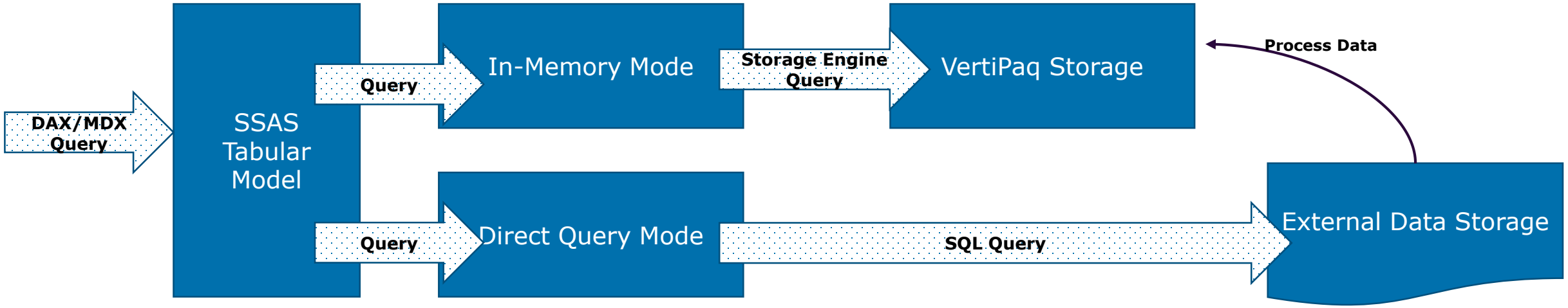




DAX Internals

Capgemini

VertiPaq Engine in DAX



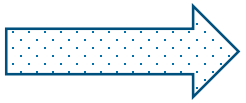
Official name of the engine on top of which DAX runs is “xVelocity in-memory Analytical Engine”

Power BI runs a special instance of AS Instance

During processing, the engine reads the data source & transforms it in the internal VertiPaq data structure.

Columnar DB

City	Channel	Color	Size	Quantity	Price
Paris	Store	Red	Large	1	15
Paris	Store	Red	Small	2	13
Toronto	Store	Green	Large	4	11
New York	Store	Green	Small	8	9
	Internet	Red	Large	16	7
	Internet	Red	Small	32	5
	Internet	Green	Large	64	3



City
Paris
Paris
Toronto
New York

Channel
Store
Store
Store
Store
Internet
Internet
Internet

Color
Red
Red
Green
Green
Red
Red
Green

Size
Large
Small
Large
Small
Large
Small
Large

Quantity
1
2
4
8
16
32
64

Price
15
13
11
9
7
5
3

Row Store – Organized in rows

Data Organized for vertical scanning to reduce read time.

Each column has its own structure

Provides quick access to single column but as calculations involves multiple columns more time is spent in calculation.

Compression

Amount
180
170
177
190
170
171
192

MAX = 192

8 bits needed

Amount(192 MAX)
12
22
15
2
22
21
0

MAX = 21

5 bits needed

Value Encoding

Color
Red
Red
Green
Green
Red
Red
Green

Color ID
0
0
1
1
0
0
1

ID	Color
0	Red
1	Green

Replaces data type with dictionary & indices.

Number of distinct values (cardinality) plays key role in column size

Dictionary Encoding

Compression

```
select QUARTER, COUNT(1) NoOfRepetation
from DIM_TIME_NEW
where "YEAR" IN (2016, 2017)
group by QUARTER
```

0 %

Results | Messages

	QUARTER	NoOfRepetation
1	Q1	181
2	Q2	182
3	Q3	184
4	Q4	184

Quarter	Start	Count
Q1	1	181
Q2	182	364
Q3	184	548
Q4	184	732

VertiPaq Store

Q.ID	Quarter
0	Q1
1	Q2
2	Q3
3	Q4

Q.ID	Count
0	181
1	182
2	184
3	184

Run Length Encoding (RLE)

A relationship is a data structure that maps **IDs in one table to row numbers in another table**

Price	
Color ID	Price
11	15
11	13
22	11
22	9
11	7
11	5
22	3
22	1

Color	
Color ID	Color
11	Red
22	Green

Row No
1
2

Price [Color ID]	Color[Row
11	1
11	1
22	2
22	2
11	1
11	1
22	2
22	2

Hierarchy & Relationship

Thank You

People matter, results count.

About Capgemini

With more than 190,000 people, Capgemini is present in over 40 countries and celebrates its 50th Anniversary year in 2017. A global leader in consulting, technology and outsourcing services, the Group reported 2016 global revenues of EUR 12.5 billion. Together with its clients, Capgemini creates and delivers business, technology and digital solutions that fit their needs, enabling them to achieve innovation and competitiveness. A deeply multicultural organization, Capgemini has developed its own way of working, the Collaborative Business Experience™, and draws on Rightshore®, its worldwide delivery model.

Learn more about us at

www.capgemini.com