Les tableaux

Objectifs

Présentation

- Comparer les tableaux et les cubes
- Explorer les tableaux
- Découvrir DAX

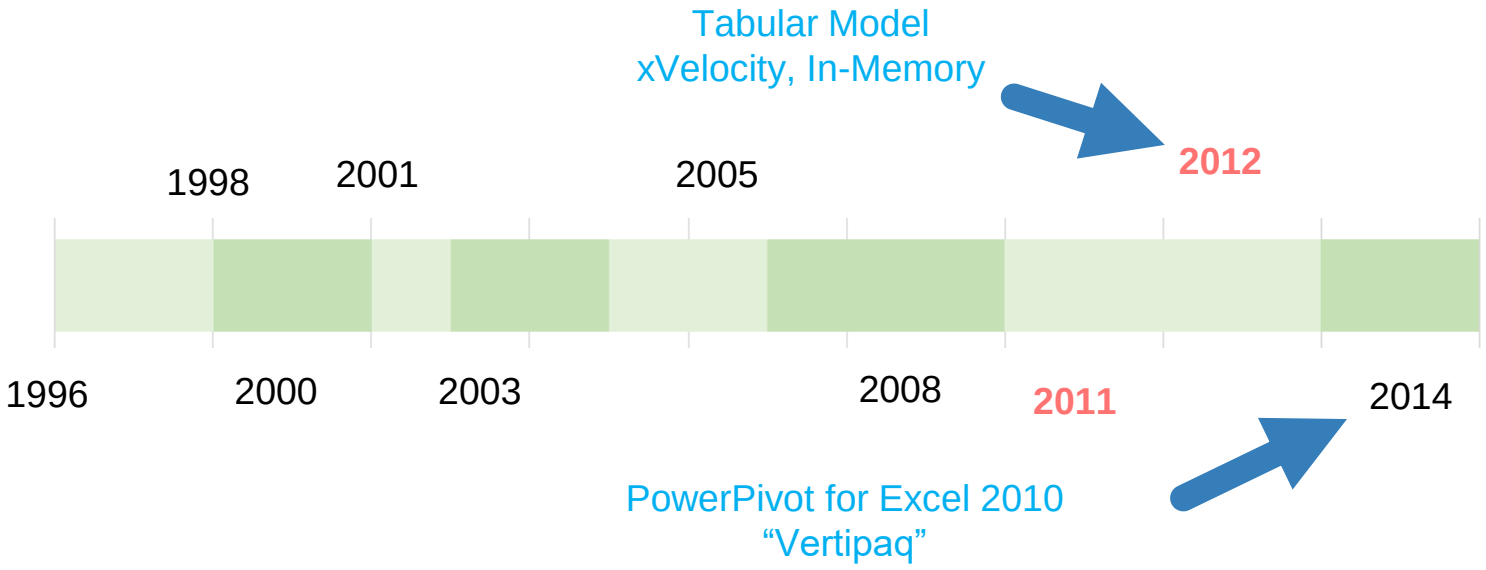
	A	B	C	D	E	F
1	Country	Salesperson	Order Date	OrderID	Units	Order Amoun
2	USA	Fuller	1/01/2011	10392	13	1,440.00
3	UK	Gloucester	2/01/2011	10397	17	716.72
4	UK	Bromley	2/01/2011	10771	18	344.00
5	USA	Finchley	3/01/2011	10393	16	2,556.95
6	USA	Finchley	3/01/2011	10394	10	442.00
7	UK	Gillingham	3/01/2011	10395	9	2,122.92
8	USA	Finchley	6/01/2011	10396	7	1,903.80
9	USA	Callahan	8/01/2011	10399	17	1,765.60
10	USA	Fuller	8/01/2011	10404	7	1,591.25
11	USA	Fuller	9/01/2011	10398	11	2,505.60
12	USA	Coghill	9/01/2011	10403	18	855.01
13	USA	Finchley	10/01/2011	10401	7	3,868.60
14	USA	Callahan	10/01/2011	10402	11	2,713.50
15	UK	Rayleigh	13/01/2011	10406	15	1,830.78
16	USA	Callahan	14/01/2011	10408	10	1,622.40
17	USA	Farnham	14/01/2011	10409	19	319.20
18	USA	Farnham	15/01/2011	10410	16	802.00

12.

Les tableaux

Découvrons ensemble les!!!!

Chronolgie


Tabular Model
xVelocity, In-Memory
2012
1998
2001
2005
1996
2000
2003
2008
2011
2014
PowerPivot for Excel 2010
“Vertipaq”

SSAS Tabulaire

Tabulaire vs Mutidimnesionnel

- Le choix n'est pas une question de "meilleur", mais de :

  - Complexité de modélisation
  - Exigences fonctionnelles (parent-enfant, many-to-many, writeback, etc.)
  - Compétences de l'équipe (MDX vs DAX)
  - Écosystème de consommation (Power BI en particulier)

SSAS Tabulaire

Architecture du modèle tabulaire

- Un modèle tabulaire repose sur :
  - Moteur de stockage en colonnes en mémoire (VertiPaq)
  - Moteur de calcul DAX
  - Modèle relationnel (tables + relations)
  - Objets de présentation (mesures, hiérarchies, perspectives)

SSAS Tabulaire

Les mesures

- formule DAX évaluée à la demande, selon le contexte (filtres, lignes, colonnes, segments).
- Usage : KPI, totaux, ratios, YoY, marges, parts de marché.
- Analogie SQL : agrégats dynamiques de type SUM, COUNT, AVG appliqués sur un jeu filtré.

Exemples:

```
Total Ventes =SUM ( Sales[Amount] )
Marge =[Total Ventes] - SUM ( Sales[Cost] )
```

SSAS Tabulaire

Les colonnes calculées

- Nouvelle colonne stockée dans le modèle, calculée lors du refresh.
- Usage : catégorisation, clés dérivées, flags, colonnes utiles pour slicers/axes, tri, relations.
- Analogie SQL : colonne calculée dans une requête ETL ou une vue, matérialisée dans le modèle.

Exemples:

```
Montant TTC = Sales[Amount] * ( 1 + Sales[VATRate] )
```

SSAS Tabulaire

Les tables calculées

- table créée via DAX, calculée lors du refresh.
- Usage : tables de pont, tables d'agrégats, snapshots, regroupements, calendriers, top N, tables de paramètres.
- Analogie SQL : création d'une table ou vue dérivée, parfois matérialisée.

Exemple 1:

```
Ventes Par Mois =
SUMMARIZECOLUMNS (
  'Date'[Year],
  'Date'[Month],
  "Total Ventes", [Total Ventes],
  "Nb Commandes", [Nb Commandes]
)
```

Exemple 2:

```
Top 10 Produits =
TOPN (
  10,
  SUMMARIZECOLUMNS ( Product[ProductName],
  "Total Ventes", [Total Ventes] ),
  [Total Ventes],
  DESC
)
```

SSAS Tabulaire

Les hiérarchies

- Structure de niveaux dans une dimension (exemple Temps : Année → Trimestre → Mois → Jour).
- Usage : exploration, drill-down, cohérence des visuels, meilleure expérience utilisateur.
- Important : la hiérarchie n'est pas un calcul ; elle organise des colonnes existantes.

Exemples:

Exemples de hiérarchies (à présenter)
Hiérarchie Temps : Année → Trimestre → Mois → Date
Hiérarchie Produit : Catégorie → Sous-catégorie → Produit
Hiérarchie Géographie : Pays → Région → Ville

SQL vs DAX

EVALUATE vs SELECT

```
SELECT *
FROM Product
```

```
EVALUATE Product
```

```
SELECT
[Product Id],
[Product Name],
 [List Price]
FROM Product
```

```
EVALUATE
ADDCOLUMNS(
    DISTINCT(
Product[Product Id] ),
    "Product Name",
CALCULATE( VALUES(
Product[Product Name] )
),
    "List Price",
CALCULATE( VALUES(
Product[List Price] ) )
)
```

```
SELECT DISTINCT
[Product Id],
[Product Name],
 [List Price]
FROM Product
```

```
EVALUATE
SUMMARIZE(
    Product,
    Product[Product Id],
    Product[Product Name],
    Product[List Price]
)
```

SQL vs DAX

La filtration

```
SELECT *
FROM DimProduct
WHERE Color = 'Red'
```

```
EVALUATE
FILTER ( Product, Product[Color] =
"Red" )
```

```
SELECT *
FROM DimProduct
WHERE Color = 'Red'
AND ListPrice > 1000
```

```
EVALUATE
FILTER (
    Product,
    AND ( Product[Color] = "Red",
Product[ListPrice] > 1000 )
)
```

```
SELECT *
FROM DimProduct
WHERE Color = 'Red'
OR Weight > 100
```

```
EVALUATE
FILTER (
    Product,
    OR ( Product[Color] = "Red",
Product[Weight] > 1000 )
)
```

SQL vs DAX

La groupement ou aggrégation

```
SELECT
    OrderDate,
    SUM(SalesAmount)
AS Sales
FROM
    FactInternetSales
GROUP BY
    OrderDate
```

```
EVALUATE
SUMMARIZE (
    'Internet Sales',
    'Internet Sales'[Order
Date],
    "Sales", SUM ( 'Internet
Sales'[Sales Amount] )
)
```

```
SELECT
    d.CalendarYear,
    SUM(s.SalesAmount) AS
Sales
FROM
    FactInternetSales s
LEFT JOIN DimDate d
    ON s.OrderDateKey =
d.DateKey
GROUP BY
    d.CalendarYear
```

```
EVALUATE
SUMMARIZE (
    'Internet Sales',
    'Date'[Calendar Year],
    "Sales", SUM ( 'Internet
Sales'[Sales Amount] )
)
ORDER BY 'Date'[Calendar
Year]
```

“

Lab:Installation et essai de DAX