

# Conception d'un cube

- Qu'est-ce qu'un cube?
- Définir
- Déployer
- Traiter
- Exploiter
- Les langages d'exploitation

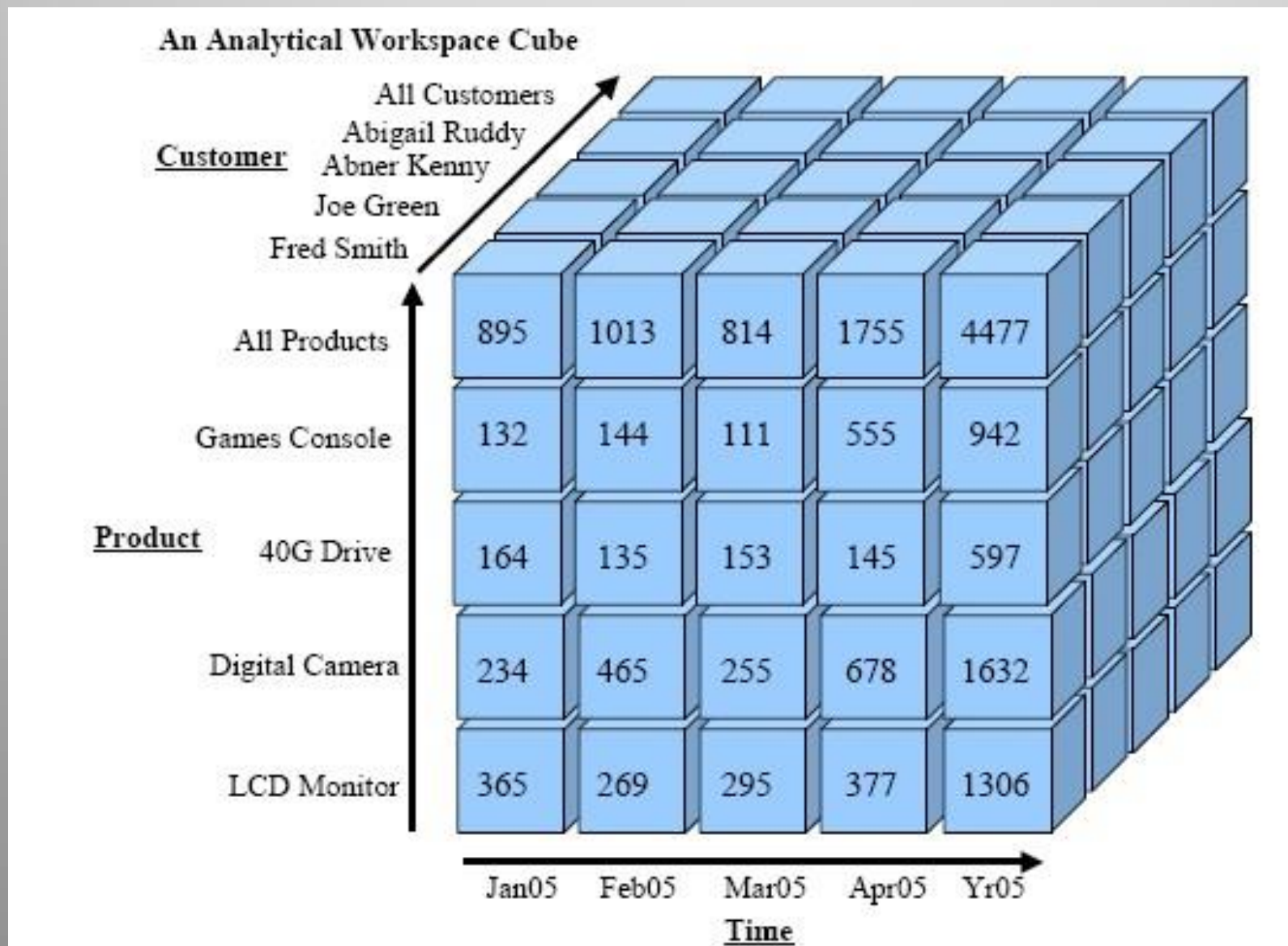
# Cube : Définition

« *Une représentation multidimensionnelle des données, ainsi que tous les totaux possibles (agrégats) précalculés* »

- Structure multidimensionnelle des données
- Tableau à N dimensions.
- Accès à une cellule par une coordonnée faite de la combinaison des valeurs des dimensions
- Contenu d'une cellule = *une mesure ou un total, en fait plusieurs mesures peuvent être stockées dans la “cellule” qui n'est pas nécessairement faite d'une seule case (comme dans excel par exemple). Pour simplifier dans les schémas, on s'en tient à la représentation d'une seule mesure pour que la représentation reste lisible. Les mesures associées à une “cellule” sont celles qui ont été spécifiées au moment de la construction du cube*

# Cube - exemple

Ici Cube à trois dimensions plus représentable



# Cube

- Le cube n'est pas une structure propre à Microsoft,, c'est un standard que nous illustrons dans l'environnement BI de Microsoft.
- Dans MS SQL BI: Le cube se définit dans SQL Server Data Tools (SSDT) (intégré à Visual Studio)
- Une fois le cube défini (construit), il faut le deployer et le traiter (process). Pourquoi?

# Définir le cube

- Définir le cube dans SSDT, c'est créer la structure du cube.

Démarche à comparer avec la définition d'un schéma relationnel pour les bases de données relationnelles.

# Déployer le cube

- Une fois la structure du cube construite, nous devons **déployer** la structure dans Sql server Analysis Services (SSAS).
- Ce processus est appelé DÉPLOIEMENT du cube. Il est comparable à la création des tables avec le langage DDL SQL.
- A ce moment le cube est construit, mais est vide

# Traiter le cube

- Une fois la structure déployée sur le serveur SSAS, nous avons besoin de charger les données dans la structure que nous avons créée.

Ce processus de chargement:

- rangement des données élémentaires à partir des tables des faits et des dimensions
- calcul et stockage des agrégations

est appelé ***traitement*** (Process) du Cube

# Traiter le cube, détails

*Autrement dit : lorsque vous traitez un cube,*

- *Les données des tables de faits et des dimensions sont chargées dans le cube*
- *les agrégations conçues pour le cube sont calculées et le cube est chargé avec les agrégations et les données calculées.*
- *Le traitement d'un cube implique donc :*
  - *La lecture des tables de dimensions pour remplir les niveaux avec les membres à partir des données réelles,*
  - *La lecture de la table des fait,*
  - *Le calcul des agrégations spécifiées,*
  - *Le stockage des données lues et des résultats des calculs dans le cube.*

*Source: Technet [https://technet.microsoft.com/en-us/library/aa216366\(v=sql.80\).aspx](https://technet.microsoft.com/en-us/library/aa216366(v=sql.80).aspx)*



# Et après, lorsque le cube est traité?

## Lorsque le cube est traité:

- Les utilisateurs peuvent l'interroger en *navigant* à l'intérieur du cube.
- Les utilisateurs peuvent l'interroger/l'exploiter suivant les droits de lecture qui ont été attribués.
- L'interrogation ou l'exploitation du cube peut se faire :
  - À l'intérieur de SSDT avec l'outil interne (navigation ou browsing)
  - Avec tout autre outil reconnaissant la structure d'une base de données OLAP (hypercube ou cube en raccourci) : SPSS, RapidMiner, SAS, Excel, PowerBI, ....

# La démarche de construction d'un cube avec SQL Server BI.

## L'environnement:

- 2 moteurs, matérialisés par des services Windows
  - SqlEngine: pour l'accès aux données du DataWareHouse/DataMart
  - Analysis Services pour le stockage du cube
- 2 outils principaux:
  - SSDT ou BIDS: pour définir, exploiter
  - SQL Server Management Studio (SSMS), pour vérification

# Le Langage

- Derrière l'outil SSDT ou BIDS dans Visual Studio Analysis Services Project, il y a un langage pour la construction et l'exploitation des données multidimensionnelles:

## Le langage MDX(**M**ultidimensional **E**xpression)

- Il existe un langage DAX mais il ne s'applique pas au modèle OLAP qui concerne ce cours. Le langage permet de travailler sur des modèles tabulaires qui gardent une structure relationnelle et non multidimensionnelle.  
Il est possible de construire des projets tabulaires dans Visual Studio Analysis Services, ils ne sont pas étudiés dans le cadre du cours.

# Le langage MDX

- [https://technet.microsoft.com/fr-fr/library/ms146020\(v=sql.120\).aspx](https://technet.microsoft.com/fr-fr/library/ms146020(v=sql.120).aspx)
  - Identificateurs
  - Types de données
  - Expressions (MDX)
  - Opérateurs
  - Fonctions
  - Commentaires
  - Mots clés réservés
  - Membres, tuples et jeux

# Exemple de requête

La requête suivante affiche des exemples de mesures calculées dont les définitions sont des expressions simples :

- WITH
- MEMBER MEASURES.CONSTANTVALUE AS 1
- MEMBER MEASURES.SCALARFUNCTION AS [Date].[Calendar Year].CURRENTMEMBER.NAME
- MEMBER MEASURES.OBJECTIDENTIFIER AS [Measures].[Internet Sales Amount]
- SELECT  
{MEASURES.CONSTANTVALUE,MEASURES.SCALARFUNCTION,MEASURES.OBJECTIDENTIFIER } ON 0,
- [Date].[Calendar Year].MEMBERS ON 1
- FROM [Adventure Works]

# Le langage Dax

- Le langage DAX (Data Analysis Expressions) est un langage de formule qui permet aux utilisateurs
  - de définir des calculs personnalisés dans les tables PowerPivot (colonnes calculées) et dans les tableaux croisés dynamiques Excel (mesures).
  - DAX inclut quelques-unes des fonctions utilisées dans les formules Excel, ainsi que des fonctions supplémentaires conçues pour l'utilisation de données relationnelles et l'agrégation dynamique.