

Corrections Exercices DAX avec AdventureWorks

Exercice colonne Calculée

1. Une colonne **Tomorrow** (table date)

```
Tomorrow = NEXTDAY(DimDate[Date])
Tomorrow 2 = DimDate[Date] + 1
```

2. Une colonne **Age** (table customer)

```
Age = DATEDIFF(DimCustomers[BirthDate], TODAY(), YEAR)
Age Yearfrac = INT(YEARFRAC(DimCustomers[BirthDate], TODAY()))
```

Cette deuxième option renvoie la fraction d'année entre les 2 dates et retourne un décimal : si la personne a 30 ans et 6 mois, elle n'a pas encore atteint sa 31e année et sera donc indiquée comme ayant INT(30,5) soit 30 ans et pas 31 : plus juste !

3. Une colonne **Weekend/weekday** (table date)

```
Weekend/Weekday = IF(DimDate[DayNumberOfWeek] in {1,7}, "Weekend", "Weekday")
```

4. Une colonne **Fiscal Quarter** (Fiscal year starting on September 1st) (table date)

```
Fiscal Quarter (StartSept) = if(DimDate[MonthNumberOfYear] in {9,10,11}, 1,
if(DimDate[MonthNumberOfYear] in {12,01,02}, 2, if(DimDate[MonthNumberOfYear] in
{3,4,5}, 3, 4)))
```

5. Une colonne **Semester** (table date)

```
Semester = IF(DimDate[MonthNumberOfYear] > 6, 2, 1)
Semester 2 = SWITCH(MONTH(DimDate[Date]), 1, "Semester1", 2, "Semester1", 3,
"Semester1", 4, "Semester1", 5, "Semester1", 6, "Semester1", "Semester2")
```

6. Une colonne **"Year-Month"** formaté: 3 juillet 2020 => 2020 - Juillet

```
Year-Month = FORMAT(DimDate[Date], "yyyy - mmmm")
```

7. Créer un **"Customer Password"** : combinaison des 2 dernières lettres du Last Name en MAJ, et 3 derniers chiffres du telephone. Ex: Jean Dubois, 888-512-452 => IS452

```
Customer Password = UPPER(RIGHT(DimCustomers[Name],2)) &
RIGHT(DimCustomers[Phone],3)
```

8. Créer un colonne **Customer mystère**: Jean Dubois => J*****s (Tips : fonction REPT)

```
Mystery Customer = LEFT(DimCustomers[Name],1) & REPT(" ",
LEN(DimCustomers[Name])-2) & RIGHT(DimCustomers[Name], 1)
```

Exercice sur Fonction d'Agrégation

1. [Total Sales Tax Paid]

Créez une nouvelle mesure pour la taxe totale

```
Total Sales Tax Paid = SUM(FactSales[TaxAmt])
```

2. [Total Margin \$]

Créez une nouvelle mesure pour la marge totale, qui correspond au total des ventes moins le coût total

```
Total Margin = SUM(FactSales[SalesAmount]) - SUM(FactSales[TotalProductCost])
```

3. [Total Margin %]

Créez une nouvelle mesure qui exprime maintenant la marge totale en pourcentage des ventes totales

```
Total Margin % = [Total Margin] / [TotalSales]
Total Margin % 2 = DIVIDE([Total Margin], [TotalSales])
```

4. [Number of Customers]

Créer une mesure renvoyant le nombre de clients dans toute la base de données

```
Number of Customers = count(DimCustomers[CustomerKey])
```

Utiliser la fonction COUNTDISTINCT ou COUNTROWS(Table) n'est pas nécessaire puisque nos valeurs sont uniques dans la Dimcustomer

5. [Number of active Customers]

Créer une mesure renvoyant le nombre de clients qui ont déjà commandé quelque chose

```
Number of Active Customers = DISTINCTCOUNT(FactSales[CustomerKey])
```

6. [% Active Customer KPI]

Créer un KPI donnant le % de client actif parmi tous les clients

```
Active Customers % = DIVIDE([Number of Active Customers], [Number of Customers])
```

7. [Address Line 2 Completeness KPI]

Certaines valeurs sont manquantes dans la colonne AddressLine 2 : Créez un KPI donnant le % de complétude de ce champ pour évaluer sa qualité (utilisez COUNTBLANK ou COUNTA)

```
Empty AddressLine 2 = COUNTBLANK(DimCustomers[AddressLine2])
AddressLine 2 Completeness % = 1 - DIVIDE([Empty AddressLine 2], [Number of Customers])
AddressLine 2 Completeness = DIVIDE(COUNTA(DimCustomers[AddressLine2]), COUNTROWS(DimCustomers))
```

8. [%Profit KPI]

Créez une mesure donnant le % de profit. Le profit est défini comme suit:
SalesAmount – Taxes – Freight – TotalProductCost

```
TotalSales = SUM(FactSales[SalesAmount])  
Profit = SUMX(FactSales, FactSales[SalesAmount] - FactSales[TaxAmt] -  
FactSales[Freight] - FactSales[TotalProductCost])  
Profit % = DIVIDE([Profit],[TotalSales])
```

9. [Avg Quantity]

Créez une mesure calculant la quantité moyenne vendue.

```
Average Quantity = AVERAGE(FactSales[OrderQuantity])
```

Fonction Calculate

1. [Total Sales of Clothing]

Utilisez la colonne Produits[Catégorie] dans votre filtre simple.

```
Total Sales of Clothing = CALCULATE([TotalSales], DimProducts[Category] =  
"Clothing")
```

2. [Total Customers Born Before 1950]

Dans ce cas, vous devez entrer la date <1er janvier 1950, dans la formule comme paramètre de filtre – et utiliser la fonction DATE() pour pouvoir faire référence à une date.

```
Total Customers Born Before 1950 =  
CALCULATE(COUNT(DimCustomers[CustomerKey]), YEAR(DimCustomers[BirthDate]) < 1950)
```

3. [Sales of Bikes to Married Men]

Vous devez utiliser plusieurs filtres sur deux tables.

```
Sales of Bikes to Married Men = CALCULATE([TotalSales], DimProducts[Category] =  
"Bikes", DimCustomers[Gender] = "M", DimCustomers[MaritalStatus] = "M")
```

4. [Total Customers Born in January]

Cette fois, vous devez utiliser la fonction MONTH() pour transformer les informations de la colonne Customers[BirthDate] en mois.

```
Total Customers Born in January = CALCULATE(DISTINCTCOUNT(DimCustomers[CustomerKey]),  
MONTH(DimCustomers[BirthDate]) = 1)
```

5. [Revenu Moyen en weekend]

Utilisez la table DimDate afin de filtrer sur le weekend.

```
Revenu Moyen en Weekend = CALCULATE(AVERAGE(FactSales[SalesAmount]),  
DimDate[Weekend/Weekday] = "Weekend")
```

6. [Cost Blue FGUK]

Calculez le coût des produits bleu pour les pays: France, Allemagne et Royaume—Uni.

```
Cost Blue FGUK = CALCULATE(SUM(FactSales[TotalProductCost]), DimProducts[Color] =  
"Blue", DimTerritory[Country] in {"France", "Germany", "United Kingdom"})
```

7. [Cost Black FGUK Date]

Calculez le coût des produits black pour les pays: France, Allemagne et Royaume—Uni où les
Uni où les commandes étaient entre commandes étaient entre 1/1/2002 et 1/5/2002.1/1/2002
et 1/5/2002.

```
Cost Black FGUK Date = CALCULATE(SUM(FactSales[TotalProductCost]),  
DimProducts[Color] = "Black", DimTerritory[Country] in {"France", "Germany",  
"United Kingdom"}, FactSales[OrderDate] > DATE(2002,1,1), FactSales[OrderDate] <  
DATE(2002,5,1))
```

```
Cost Black FGUK Datesbetween = CALCULATE(SUM(FactSales[TotalProductCost]),  
DimProducts[Color] = "Black", DimTerritory[Country] in {"France", "Germany",  
"United Kingdom"}, DATESBETWEEN(FactSales[OrderDate], DATE(2002,1,1),  
DATE(2002,5,1)))
```

Fonction ALL

1. [% of All Customer Sales]

Créez un KPI pour obtenir le ratio des ventes. Utilisez-le avec l'occupation du client dans une table.

```
Grand Total ALL Sales = CALCULATE([TotalSales], ALL(FactSales))
% ALL Sales = [TotalSales] / [Grand Total ALL Sales]
```

2. [Total Sales for All Selected Dates]

Créez un KPI pour obtenir le grand total des ventes pour une période sélectionnée. Explorez-les par trimestre dans une table

```
Total Sales ALLSELECTED Dates = CALCULATE([TotalSales], ALLSELECTED(DimDate))
```

3. [Total Sales for All Selected Products]

a/Créez un KPI pour obtenir le grand total des ventes pour une gamme de produits définie. Explorez-les par sous-catégories de produits dans une table.

```
Total Sales AllSELECTED Products =CALCULATE([TotalSales],ALLSELECTED(DimProducts))
```

b/Quel ratio représente les achats de produits bleus par les femmes? par les hommes?

```
% TotalSales AllSELECTED Products = DIVIDE([Total Sales AllSELECTED Products],[Grand Total ALL Sales])
```

4. [Total Sales All Except Category]

Créez un KPI pour obtenir le grand total des ventes par catégories. Explorez-les par sous-catégories dans une table

```
Total Sales ALLEXCEPT Category = CALCULATE(sum(FactSales[SalesAmount]), ALLEXCEPT(DimProducts,DimProducts[Category]))
```

5. [Rank Sales Country]

Essayez de créer un ranking des pays selon leur vente.

```
RANKX ALL Country by TotalSales = RANKX(ALL(DimTerritory[Country]), [TotalSales],,,Dense)
```

```
RANK ALL Country ORDER BY Sales = RANK(SKIP, ALL(DimTerritory[Country]), ORDERBY(1-[TotalSales]))
```

```
RANK ALL Country = RANK(SKIP, all(DimTerritory[Country]))
```

Utiliser la fonction RANK simple me classera les pays par ordre alphabétique.

Exercice Time Intelligence

Attention, bien relier OrderDate de la FactSale à Date de la DimDate (relation secondaire à la datekey)

1. [PreviousQuarter]

Créez une mesure utilisant les Sales du Trimestre précédent

```
PreviousQuarter = CALCULATE([TotalSales], PREVIOUSQUARTER(DimDate[Date]))
```

2. [RatioPreviousQ]

Créez le ratio montrant la progression des Sales par trimestre

```
RatioPreviousQ = ([TotalSales] - [PreviousQuarter]) / [PreviousQuarter]
```

3. [AvgRolling on 30 past days]

Créez une mesure calculant une moyenne mobile sur les 30 derniers jours

```
AvgRolling30days =  
CALCULATE([TotalSales], DATESINPERIOD(DimDate[Date], Max(DimDate[Date]), -30, day)) /  
30
```

```
Progression Previous Quarter = [TotalSales]-[PreviousQuarter] / [RatioPreviousQ]
```

```
Running Sales by Y = CALCULATE([TotalSales], DATESYTD(DimDate[Date]))
```

```
Sales One Year Before = CALCULATE([TotalSales], SAMEPERIODLASTYEAR(DimDate[Date]))
```