

## A. Création de tableaux et graphiques croisés dynamiques (TCD et GCD) : description de l'exemple

### 1. Présentation de l'exemple

Dans le cadre d'un projet informatique de la société SacEni qui commercialise des sacs, nous travaillons sur la phase de recette de l'application portail client qui permet la vente en ligne de ces sacs. L'application **EniSac\_App** se compose de trois parties : la partie Front, la partie Middle et la partie Back.



*Dans les applications informatiques, on retrouve généralement les notions de Front-End, Middleware et Back-End : la partie Front-End porte sur l'interface de l'application ; la partie Middleware permet de gérer la relation entre les commandes liées à l'interface et les données ; la partie Back-End permet le stockage sur les bases de données.*

Pendant cette phase de test, les testeurs tracent leurs actions dans des fichiers Excel. Il est également demandé par la Direction d'avoir une vue de l'avancement global et de la situation de la recette.

Un peu de vocabulaire :

- ▶ **Test d'une fonctionnalité** : une fonctionnalité est couverte par un ensemble de cas de test. Ils couvrent l'ensemble des exigences de cette fonctionnalité. La couverture de la fonctionnalité doit être exhaustive.
- ▶ **Cas de test** : le cas de test correspond à un ensemble d'étapes composant un scénario à tester. Un test est :
  - ▶ OK si l'ensemble des étapes est déroulé et que toutes les étapes sont conformes à la description ;
  - ▶ KO si l'ensemble des étapes est déroulé et qu'au moins une étape n'a pas été conforme à la description ;

- ▶ Bloqué : si à cause d'une anomalie, l'exécution n'a pas pu aller à la fin du scénario de test.
- ▶ **Anomalie** : une anomalie correspond à un cas rencontré qui n'est pas conforme à la description du cas de test.
- ▶ **Cycle de test** : un cycle de test correspond à l'exécution d'un ensemble de cas de test durant une période donnée. Il peut y avoir plusieurs cycles de test et donc plusieurs exécutions de chaque cas de test, notamment pour vérifier l'impact d'une correction éventuelle sur des tests connexes.

## 2. Présentation du fichier

Le fichier `Enoncé_4-ABC.xlsm` se décompose en cinq onglets :

- ▶ Feuille Test

La feuille **Test** récapitule l'ensemble des 60 tests de cette application avec leur état à l'instant de cet exemple.

Colonne	Libellé	Description
Colonne A	Nom	Il s'agit du nom du test, ici, les tests sont différenciés par un numéro. En théorie le nom du cas de test est suffisamment explicite pour décrire l'objet du test.
Colonne B	Priorité	La priorité est un critère déterminant pour qualifier l'importance d'un test par rapport aux autres. La priorité peut prendre les valeurs P1, P2, P3.
Colonne C	Statut du test	Il s'agit de connaître le statut du test au moment de la réalisation de l'extraction des données. Il peut être : <ul style="list-style-type: none"> <li>▶ OK</li> <li>▶ KO</li> <li>▶ Non commencé</li> <li>▶ Non livré</li> </ul>
Colonne D	Fonctionnalité	Il s'agit du sujet englobant le test.

### ► Feuille Exécution

La feuille **Exécution** contient toutes les exécutions de cas de test. Un test peut être exécuté plusieurs fois dans le cadre de différents cycles ou lorsque le cas de test est rejoué suite à une correction. Les colonnes sont les suivantes :

Colonne	Libellé	Description
Colonne A	ID Exécution	Identifiant unique de l'exécution de test. Il permet, par exemple, de faire la différence entre deux exécutions d'un même test.
Colonne B	Test	Nom du test exécuté.
Colonne C	Priorité	Priorité du test évoqué dans la colonne B (P1, P2, P3).
Colonne D	Statut de l'exécution	Situation du test au terme de son exécution. Il peut être : <ul style="list-style-type: none"> <li>► OK</li> <li>► KO</li> <li>► Bloqué</li> </ul>
Colonne E	Fonctionnalité	Il s'agit du sujet englobant le test évoqué dans la colonne B.
Colonne F	Date d'exécution	Date de l'exécution du test.
Colonne G	Testeur	Personne qui a exécuté le test : ici Cécile ou Jean.

### ► Feuille Anomalies

La feuille **Anomalies** recense les différentes anomalies détectées durant l'exécution des tests.

Colonne	Libellé	Description
Colonne A	ID	Identifiant unique de l'anomalie.
Colonne B	Libellé	Libellé de l'anomalie ici composé de l'identifiant et du test. Dans de vrais cas, l'anomalie doit avoir un nom suffisamment explicite pour décrire l'objet de l'anomalie.
Colonne C	Date d'ouverture	Date de détection et/ou création de l'anomalie dans le fichier.

Colonne	Libellé	Description
Colonne D	Statut	État d'avancement de l'anomalie. Différentes étapes sont possibles : <ul style="list-style-type: none"> <li>▶ Créée : vient d'être créée mais non qualifiée ;</li> <li>▶ En qualification : l'anomalie est avérée, mais n'a pas encore de projet attribué pour la correction ;</li> <li>▶ En correction : l'anomalie a été affectée à un projet qui se charge de la correction ;</li> <li>▶ À valider : la correction est terminée et livrée, le testeur doit repasser le cas pour terminer l'anomalie.</li> <li>▶ Terminée : l'anomalie est soit corrigée, soit abandonnée.</li> </ul>
Colonne E	Test associé	Test sur lequel l'anomalie a été détectée.
Colonne F	Date de clôture	Date à laquelle l'anomalie a été clôturée (pour les motifs vus ci-dessus).
Colonne G	Projet	Projet en charge de corriger l'anomalie détectée. Les projets de l'application sont les suivants : <ul style="list-style-type: none"> <li>▶ Front ;</li> <li>▶ Middle ;</li> <li>▶ Back ;</li> <li>▶ Non défini : si l'anomalie n'a pas encore été qualifiée.</li> </ul>
Colonne H	Testeur	Testeur qui a détecté l'anomalie.
Colonne I	Priorité	Priorité du ticket (P1, P2, P3).

▶ Feuille **Rapport**

La feuille **Rapport** est vierge initialement. Elle contiendra par la suite le rapport qui sera généré et exporté vers PowerPoint dans la deuxième partie de cet exemple.

▶ Feuille **TCD\_GCD**

La feuille **TCD\_GCD** contiendra les tableaux croisés dynamiques et les graphiques croisés dynamiques qui seront mis en place dans cet exemple.

### 3. Fonctionnalités

L'objectif de cet exemple est d'automatiser la création d'un rapport d'activité des tests sur l'application. Par conséquent, il faut réaliser différents tableaux croisés dynamiques pour récupérer l'information puis la transformer en graphiques croisés dynamiques pour la partie visuelle.

Voici les situations envisagées :

- ▶ Création d'un suivi hebdomadaire du stock de tickets ;
- ▶ Nombre d'anomalies par projet (et par priorité) ;
- ▶ Avancement des cas de test ;
- ▶ Revue des cycles de test ;
- ▶ Indicateur de situation des tests : nombre d'anomalies et pourcentage de tests OK.

## B. Création de tableaux et graphiques croisés dynamiques (TCD et GCD) : notions de cours

### 1. Créer un tableau croisé dynamique simple

Un **tableau croisé dynamique**, abrégé TCD, permet d'exposer des données en fonction d'axes définis par l'utilisateur. Les données sont agrégées selon les axes.

Un axe correspond à une définition permettant de qualifier les données. Par exemple : le sexe, le poste, le groupe... Les axes sont exprimés soit en colonne soit en ligne et leur contenu peut être filtré pour ne pas forcément afficher toutes les valeurs.

Les données sont agrégées c'est-à-dire qu'elles sont regroupées autour des valeurs de l'axe. Il est possible de les exprimer de plusieurs manières différentes : somme, moyenne, nombre (compter)... Les données sont forcément des valeurs numériques.

#### Exemple

Voici un exemple de transformation de tableau de données en TCD.

🔗 Pour réaliser cet exemple, ouvrez le fichier **ExempleCours\_Chapitre4.xlsm**, feuille TCD.

La source de données exprimée sous forme de tableau porte sur les notes d'élèves :

	A	B	C
1	Sexe	Note	Nom
2	Homme	9,5	Albert
3	Homme	12	Anthony

	A	B	C
4	Femme	15	Aurélié
5	Femme	9,5	Béatrice
6	Homme	8,5	Benoit
7	Femme	13	Bérangère

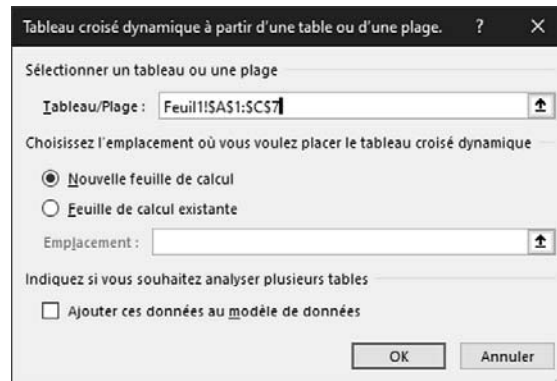
La transformation de ces données en TCD :

	Étiquettes de colonnes		
	Femme	Homme	Total général
Moyenne de Note	12,5	10	11,25

Ici l'axe correspond au sexe, et les données sont les notes dont la moyenne est réalisée en fonction de l'axe.

### Manipulation

- ✎ Pour créer un TCD, sélectionnez la plage de données du tableau (en-tête comprise) soit la plage A1:C7 puis dans l'onglet Insertion cliquez sur **Tableau croisé dynamique**.



- ✎ Choisissez si vous voulez faire apparaître le TCD dans une nouvelle feuille ou une feuille existante.
- ✎ Validez la création du TCD.

Le TCD est créé sur la page. Vous pouvez manipuler les différents champs qui se trouvent dans la partie supérieure **Champs de tableau croisé dynamique** pour les positionner en tant qu'axe et données (en bas du volet) avec un glisser-déposer. Les axes sont appelés **Étiquettes de lignes** ou **Étiquettes de colonnes**. Les données sont appelées **Valeurs**. Deux nouveaux onglets apparaissent également dans le ruban Excel : **Analyse du tableau croisé dynamique** et **Création**.

✎ Faites glisser le champ **Sexe** dans la zone **Colonnes** et le champ **Note** dans la zone **Valeurs**.

C'est le nombre de notes qui est calculé.

✎ Pour obtenir la moyenne cliquez sur **Nombre de Note** dans la zone **Valeurs**, puis sur **Paramètres des champs de valeurs** et sélectionnez **Moyenne**.

	A	B	C	D
1				
2				
3		Étiquettes de colonnes		
4		Femme	Homme	Total général
5	Moyenne de Note	14	12	13,3333333
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				

**Champs de tableau croisé dynamique**

Choisissez les champs à inclure dans le rapport :

Rechercher

Sexe  
 Note  
 Nom

Plus de tableaux...

Faites glisser les champs dans les zones voulues ci-dessous :

Filtres

Colonnes  
Sexe

Lignes

Valeurs  
Moyenne de Note



*Une donnée peut-elle être un axe ? Si vous n'avez pas encore bien compris le fonctionnement des TCD, revenez sur cette remarque plus tard, elle vous déstabilisera plus qu'elle n'apportera de nouvelles perspectives. Pour les plus aguerris aux TCD, vous pourrez constater qu'une colonne comme les notes considérées comme des données peut aussi être un axe. Et le prénom peut servir de données. En effet, si la note est considérée comme un axe, alors il est possible de compter le nombre de personnes ayant obtenu chaque note.*

## A. Objectifs du chapitre

Objet élémentaire d'Excel, les cellules seront votre première étape sur la programmation VBA des objets Excel.

Sans parcourir l'intégralité des actions possibles avec les cellules en VBA, ce chapitre vous permettra de connaître les principales propriétés et méthodes des objets élémentaires que sont les **cellules** et **plages de cellules** sous Excel.

Vous verrez ensuite des exemples de code les plus fréquents avant de terminer en validant vos nouveaux acquis au travers d'exercices.

### 1. Objet et variable Range

Lorsque vous avez créé votre première macro avec l'Enregistreur de macros dans le chapitre L'enregistreur de macros, vous avez déjà fait la connaissance de l'objet **Range**.

#### a. Objet Range

Le type de données Range est le premier que vous découvrez dans la trousse à objets de VBA Excel.

Cet objet représente aussi bien une cellule seule, une plage de cellules ou encore une série de cellules non contiguës. Lorsque vous utilisez cet objet, la syntaxe générale est la suivante :

```
Range ( Adresse )
```

*Exemple 1 : syntaxe générale de l'objet Range*

Selon vos besoins, l'adresse passée sous forme de chaîne de caractères peut représenter la ou les cellules que vous souhaitez manipuler :

```
Range ( "A1" )  
Range ( "C2:D5" )  
Range ( "A1 , B2 , G3" )  
Range ( "CelluleNommee" )
```

*Exemple 2 : possibilité de syntaxe pour l'objet Range*



## 2. Variable de type Range

Tout comme vous l'avez fait pour des chaînes de caractères, des valeurs numériques ou des booléens, il est possible de travailler avec l'objet **Range** au travers d'une variable. Ce paragraphe vous indique comment déclarer, affecter et manipuler ensuite cet objet.

### a. Déclaration

La déclaration d'une variable de type **Range** est similaire à celles vues jusqu'à présent.

```
Dim VotreCellule As Range
```

*Exemple 3 : déclaration d'une variable de type Range*

Il en est autrement pour l'affectation de valeurs.

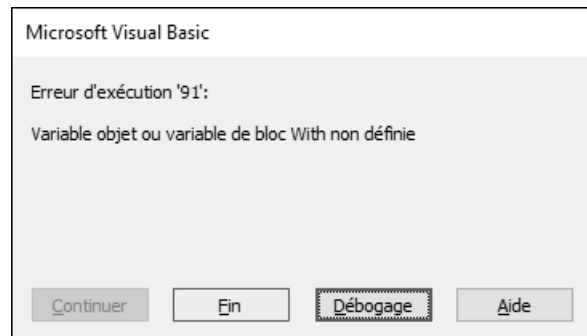
### b. Affectation

Contrairement aux types de données élémentaires comme les chaînes de caractères ou les valeurs numériques, l'affectation pour le type de donnée **Range** a pour syntaxe générale la suivante :

```
Dim rCellule As Range  
Set rCellule = Range("DateDuJour")
```

*Exemple 4 : affectation d'une variable de type Range*

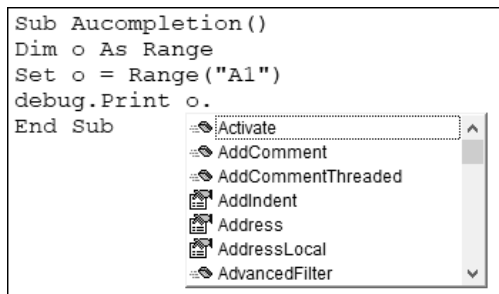
Vous pouvez constater la présence du mot-clé **Set** avant le nom de la variable **rCellule**. Si vous oubliez le mot-clé **Set**, une erreur 91 apparaîtra lors de l'exécution de la ligne de code.



L'usage du mot-clé **Set** devient nécessaire pour tous les types de données **Objet**.

### c. Utilisation

Une fois que la variable est instanciée, vous pourrez profiter de l'autocomplétion lorsque vous commencerez à vouloir accéder aux propriétés et méthodes de l'objet `Range`.



Vous pourrez saisir les premières lettres de la propriété ou de la méthode et utiliser au choix la touche `→` ou la combinaison `Ctrl Espace` pour que la ligne se renseigne automatiquement.

## B. Objet Cells

Il existe un objet plus petit encore que l'objet `Range`, qui permet lui aussi de manipuler les cellules ; il s'agit de l'objet `Cells`. Il est plus petit car là où l'objet `Range` permet de gérer une plage de cellules, l'objet `Cells` se limite à une seule cellule. Il n'est par exemple ainsi pas possible de faire référence à la plage de cellules A1:D3 avec un objet `Cells`.

Pour utiliser l'objet `Cells`, il existe deux syntaxes possibles, chacune fournissant à l'objet `Cells` une ligne et une colonne, coordonnées de la cellule.

```
'Syntaxe générale
Cells(indiceLigne, indiceColonne)
'Syntaxe avec des valeurs numériques uniquement
Debug.Print Cells(3,4) 'affiche le contenu de la cellule D3
'Syntaxe avec une valeur numérique pour la ligne et une chaîne de
caractères pour la colonne
Debug.Print Cells(5, "F") 'affiche le contenu la cellule F5
```

*Exemple 5 : différentes syntaxes avec l'objet Cells*

L'objet `Cells` est utilisable avec une variable de type `Range`.

```
Dim o As Range
Set o = Cells(1,3)
```

*Exemple 6 : utilisation d'une variable de type Range pour l'objet Cells*

Les types de données `Range` et `Cells` partageront des propriétés et méthodes communes.

## C. Quelques cellules particulières : ActiveCell, Selection et Target

Lorsque vous utilisez l'Enregistreur de macros également, certains mots-clés spécifiques peuvent apparaître pour définir une cellule ou une plage de cellules. Cette courte section a pour objectif de vous en expliquer les grandes lignes.

### 1. Cellule active : ActiveCell

Lorsque vous cliquez sur une cellule, on dit que vous l'activez. Cette **cellule active** est représentée par un objet natif en VBA : **ActiveCell**. Cet objet, de type Range, représente la cellule active dans votre classeur.

Ci-dessous un exemple de cellule active :

	A	B	C
1	Date	Chiffre d'affaires	Pr
2	2021-01-01	123000	cl

Lorsque vous sélectionnez une plage de cellules, la cellule à partir de laquelle vous effectuez votre sélection est la cellule active.

	A	B	C	D
1	Date	Chiffre d'affaires		Produits
2	2021-01-01	123000		Clavier
3	2021-02-01	148500		Souris
4	2021-03-01	112800		Écran
5	2021-04-01	135300		Casque
6				

Enfin, si vous effectuez une série de sélections de cellules non adjacentes, la dernière cellule activée est la cellule active.

	A	B	C	D	E
1	Date	Chiffre d'affaires		Produits	Chiffre d'affaires
2	2021-01-01	123000		Clavier	129900
3	2021-02-01	148500		Souris	155880
4	2021-03-01	112800		Écran	88332
5	2021-04-01	135300		Casque	145488

Vous pouvez voir le mot-clé `ActiveCell` dans l'exemple suivant :

```
Sub RemplirCellule()  
    ActiveCell.FormulaR1C1 = "Bonjour"  
End Sub
```

*Exemple 7 : apparition du mot-clé `ActiveCell` depuis l'Enregistreur de macros.*

## 2. Sélection active : Selection

Quand vous travaillez dans Excel et que vous sélectionnez une ou plusieurs cellules, VBA utilise un objet spécifique : **Selection**. Cet objet, de type `Range` comme `ActiveCell`, couvre un spectre similaire de fonctionnalités à celui-ci. Ce mot-clé pourra apparaître lorsque vous utilisez l'Enregistreur de macros.

Lorsque vous travaillez avec une seule cellule, l'objet `Selection` est identique à l'objet `ActiveCell`. Cependant, dès l'instant que vous utilisez une plage de cellules ou des cellules éparpillées, l'objet `Selection` vient trouver toute sa place, pour représenter l'ensemble des cellules sélectionnées.



*Bien que la sélection puisse représenter plusieurs cellules, vous n'aurez toujours qu'une seule cellule active.*

```
Sub UsageDeSelection()  
    Range("D3").Select  
    With Selection.Interior  
        .Pattern = xlSolid  
        .PatternColorIndex = xlAutomatic  
        .Color = 255  
        .TintAndShade = 0  
        .PatternTintAndShade = 0  
    End With  
End Sub
```

*Exemple 8 : cas d'utilisation du mot-clé `Selection` par l'Enregistreur de macros*

## 3. Cellule(s) impliquée(s) dans les événements Excel : Target

Comme vous le verrez dans le chapitre suivant Manipuler les feuilles Excel, le code VBA peut se déclencher automatiquement à partir d'événements qui ont lieu dans votre feuille ou votre classeur. Que cela soit un changement dans une feuille ou un double clic sur une cellule, ces événements utilisent le mot-clé **Target** pour représenter la référence de l'objet qui est ciblé dans la procédure événementielle. Dans ces événements qui ont trait à des cellules, le type de l'objet **Target** sera **Range**.

```
Private Sub Worksheet_BeforeDoubleClick(ByVal Target As Range, Cancel As Boolean)
    If Target.Address = "$C$3" Then
        MsgBox "Interdiction de modifier cette cellule", vbCritical + vbOKOnly
    End If
End Sub
```

*Exemple 9 : cas d'utilisation du mot clé Target dans un événement de feuille Excel*

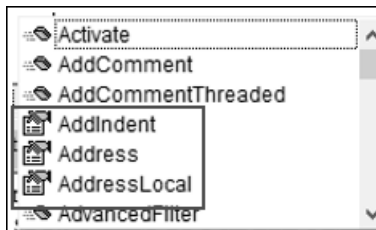
## D. Les propriétés des cellules

Dans un premier temps, nous verrons ce qu'on appelle une **propriété** puis nous verrons les principales propriétés qui peuvent servir avec les cellules.

### 1. Définition d'une propriété

En informatique, les propriétés d'un objet sont ses caractéristiques, ce qui le définit. Prenons exemple d'une voiture, vous aurez parmi ses propriétés sa marque, son modèle, sa couleur, son année de sortie. Les propriétés peuvent être de tout type, (numérique, date, chaîne de caractères). Certaines propriétés seront modifiables, d'autres seulement lisibles.

En cours de programmation, vous pouvez reconnaître une propriété à l'icône de doigt pointé :



### 2. Le contenu d'une cellule : Value, Value2

Lorsque vous commencez à programmer avec Excel, la première chose qui vous intéresse avec une cellule sera la valeur qu'elle contient. Dans ce paragraphe, vous retrouverez une partie des lignes de code que vous avez découvertes avec l'Enregistreur de macros.

La valeur contenue dans une cellule est obtenue grâce à la propriété **Value**. VBA considère d'ailleurs cette propriété comme celle par défaut.

La syntaxe générale est la suivante :

```
Debug.Print Range("A1").Value 'Affiche la valeur contenue dans la cellule A1.
```

*Exemple 10 : syntaxe générale de la propriété Value d'un objet Range*

Cette propriété peut aussi bien être lue que modifiée par votre programme.

```
Range("A1").Value = 3 'Affecte la valeur 3 dans la cellule A1
MsgBox Range("A1").Value 'Affiche la valeur de la cellule A1
```

*Exemple 11 : usage de la propriété Value*

Selon le type de valeur contenue dans la cellule, le type de donnée retourné par la propriété **Value** sera automatiquement adapté par VBA. Il conviendra donc d'utiliser les bons types de données pour vos variables, au risque d'avoir une mauvaise interprétation des valeurs par votre programme.

Dans le cas de figure où vous avez les données suivantes dans les cellules :

	A	B	C
1	123	exemple de texte	2021-08-17

Le code suivant permettra de récupérer chacune des valeurs avec le bon type de donnée :

```
Dim iNumerique As Integer, sTexte As String, dtDate As Date
iNumerique = Range("A1").Value '123
sTexte = Range("B1").Value '"exemple de texte"
dtDate = Range("C1").Value '2021-08-17
```

*Exemple 12 : récupération des valeurs au travers de la propriété Value*

De la même façon, il est possible d'écrire des valeurs dans les cellules à partir des variables :

```
Dim strTexte As String
strTexte = "Bonjour"
Range("A1").Value = strTexte
```

*Exemple 13 : affectation de valeur à une cellule par la propriété Value*

Il existe une autre propriété, **Value2**, plus rarement utilisée mais que vous pourrez rencontrer dans les macros d'autres développeurs ; elle est similaire à **Value**, à ceci près qu'elle ne retourne pas de valeur de type **Currency** ou **Date** (ce type étant remplacé par le numéro de série, par exemple 2021-08-17 qui retournera 44425).

En reprenant le cas de l'illustration précédente, la valeur récupérée sera la suivante :

```
?Range("C1").Value2
44425
```

*Exemple 14 : usage de la propriété Value2*