

Voici les 20% des idées clés qui vous permettront de comprendre 80% du TD 1 sur la découverte du tableur :

I. Fonctionnalités générales du tableur

- **Fenêtre du tableur :**
 - **Barre d'accès rapide :** Contient les commandes les plus utilisées (enregistrer, annuler, rétablir).
 - **Ruban :** Regroupe les commandes par onglets thématiques (Accueil, Insertion, Mise en page, Formules, Données, etc.). Chaque onglet contient des groupes de commandes, et chaque groupe contient des boutons de commande. Les lanceurs de boîtes de dialogue permettent d'accéder à plus d'options.
 - **Barre de formule :** Affiche l'adresse de la cellule sélectionnée et son contenu.
 - **Zone de feuille de calcul :** Zone principale de travail, composée de cellules organisées en lignes (numérotées) et colonnes (lettres).
 - **Barre d'état :** Affiche des informations sur le mode de fonctionnement et l'état du document.
- **Cellule :** Intersection d'une ligne et d'une colonne. C'est l'unité de base d'une feuille de calcul.
 - **Désigner une cellule :** Par son adresse (ex: C3) ou en la nommant.
 - **Désigner une plage de cellules :** Par l'adresse de la première et de la dernière cellule séparées par ":" (ex: B2:D5).
 - **Désigner une cellule dans une autre feuille :** NomFeuille!RéférenceCellule (ex: Feuil1!A1).

II. Les formules de calcul

- **Construction :**
 - Commencent toujours par le signe =.
 - Utilisent des opérateurs arithmétiques (+, -, *, /, ^).
 - Peuvent contenir des références à des cellules ou à des plages de cellules.
 - Peuvent contenir des fonctions prédéfinies.
- **Recopie d'une formule :**
 - Utiliser la poignée de recopie (le petit carré en bas à droite de la cellule).
 - Les références relatives (ex: A1) s'adaptent à la nouvelle position.
 - Les références absolues (ex: \$A\$1) restent fixes.
 - Les références mixtes (ex: \$A1 ou A\$1) figent soit la ligne, soit la colonne.
- **Nommer une cellule/plage :** Permet de simplifier l'écriture des formules et de rendre les références absolues.

III. Les fonctions élémentaires

- **Règles de base :**
 - Une fonction a un nom (ex: SOMME, MOYENNE).
 - Elle prend des arguments entre parenthèses, séparés par des points-virgules (ex: =SOMME(A1;B1;C1)).
 - Elle renvoie un résultat.
- **Fonctions de calculs :**
 - **SOMME :** Calcule la somme d'une plage de cellules ou d'une liste de valeurs. Syntaxe :
SOMME(plage_de_cellules) OU SOMME(cellule1;cellule2;...)
 - **MOYENNE :** Calcule la moyenne d'une plage de cellules ou d'une liste de valeurs. Syntaxe :
MOYENNE(plage_de_cellules) OU MOYENNE(cellule1;cellule2;...)
 - **MAX :** Renvoie la valeur maximale d'une plage de cellules ou d'une liste de valeurs. Syntaxe :
MAX(plage_de_cellules) OU MAX(cellule1;cellule2;...)

- o **MIN** : Renvoie la valeur minimale d'une plage de cellules ou d'une liste de valeurs. Syntaxe :
MIN(plage_de_cellules) OU MIN(cellule1;cellule2;...)
- o **NBVAL** : Compte le nombre de cellules non vides. Syntaxe : NBVAL(plage_de_cellules) ou
NBVAL(cellule1;cellule2;...)
- o **NB** : Compte le nombre de cellules contenant des nombres. Syntaxe : NB(plage_de_cellules) ou
NB(cellule1;cellule2;...)
- o **ARRONDI** : Arrondit un nombre au nombre de chiffres spécifié. Syntaxe :
ARRONDI(nombre;no_chiffres)

IV. Autres fonctions de base

- **Plancher** : Valeur maximale inférieure ou égale à une valeur donnée. Syntaxe : MAX(nombre;plancher)
- **Plafond** : Valeur minimale supérieure ou égale à une valeur donnée. Syntaxe : MIN(nombre;plafond)
- **RANG** : Détermine le rang d'une valeur dans une liste. Syntaxe : RANG(nombre;référence;ordre)

En résumé, pour le TD1, il est essentiel de comprendre les bases du tableur (cellules, plages, adressage), la construction de formules de calcul simples, la recopie de formules, le nommage de cellules et l'utilisation des fonctions élémentaires (SOMME, MOYENNE, MAX, MIN, NBVAL, NB, ARRONDI, RANG, PLANCHER, PLAFOND).

Voici les 20% des idées clés qui vous permettront de comprendre 80% du TD 2 sur les fonctions logiques :

I. La fonction SI

- **Syntaxe** : =SI(test_logique; valeur_si_vrai; valeur_si_faux)
 - o **test_logique** : Une expression qui peut être évaluée comme VRAI ou FAUX (ex: A1>10, B2="Texte", C3=D3).
 - o **valeur_si_vrai** : La valeur à renvoyer si le test logique est VRAI.
 - o **valeur_si_faux** : La valeur à renvoyer si le test logique est FAUX.
- **Utilité** : Permet d'effectuer des actions différentes en fonction d'une condition.
- **Exemple** : =SI(A1>10;"Grand";"Petit") - Si la valeur en A1 est supérieure à 10, la fonction renvoie "Grand", sinon elle renvoie "Petit".

II. Enrichissement de la fonction SI

- **SI imbriqués** : Permet de tester plusieurs conditions successives.
 - o **Syntaxe** : =SI(test1; valeur_si_vrai1; SI(test2; valeur_si_vrai2; valeur_si_faux2))
 - o **Exemple** : =SI(A1>10;"Grand";SI(A1>5;"Moyen";"Petit")) - Si A1>10, renvoie "Grand", sinon si A1>5, renvoie "Moyen", sinon renvoie "Petit".
- **Opérateurs logiques** : Permettent de combiner plusieurs conditions dans un seul test logique.
 - o **ET** : Toutes les conditions doivent être vraies. Syntaxe : ET(condition1;condition2;...)
 - o **OU** : Au moins une des conditions doit être vraie. Syntaxe : OU(condition1;condition2;...)
 - o **Exemple** : =SI(ET(A1>10;B1="Texte");"OK";"Non") - Si A1>10 ET B1="Texte", renvoie "OK", sinon renvoie "Non".
- **ESTVIDE** : Vérifie si une cellule est vide.
 - o **Syntaxe** : =ESTVIDE(cellule)

- **Exemple** : =SI(ESTVIDE(A1);"Cellule vide";"Cellule non vide") - Si A1 est vide, renvoie "Cellule vide", sinon renvoie "Cellule non vide".
- **ABS** : Renvoie la valeur absolue d'un nombre.
 - **Syntaxe** : =ABS (nombre)
 - **Exemple** : =ABS (-5) renvoie 5.

III. Points importants

- **Conditions** : Bien identifier les conditions à tester et les opérateurs de comparaison à utiliser (=, >, <, >=, <=, <>).
- **Valeurs textuelles** : Mettre les valeurs textuelles entre guillemets doubles ("").
- **Imbrication** : Le nombre de SI imbriqués est limité.
- **Alternatives** : Bien identifier les différentes actions à réaliser en fonction des résultats des tests.
- **Plages nommées** : L'utilisation de plages nommées est fortement conseillé pour simplifier la lecture et la compréhension des formules.

En résumé, pour le TD2, il est crucial de maîtriser la fonction SI, les opérateurs logiques ET/OU, la fonction ESTVIDE, la fonction ABS, et de savoir comment imbriquer plusieurs fonctions SI pour gérer des situations complexes avec plusieurs conditions.

Voici les 20% des idées clés qui vous permettront de comprendre 80% du TD 3 sur la fonction recherche et les messages d'erreur :

I. La fonction RECHERCHEV

- **Syntaxe** : =RECHERCHEV(valeur_cherchée; table_matrice; no_index_col; [valeur_proche])
 - valeur_cherchée : La valeur à rechercher dans la première colonne de la table_matrice.
 - table_matrice : La plage de cellules contenant les données. **Important : la première colonne doit contenir les valeurs de recherche.**
 - no_index_col : Le numéro de la colonne dans table_matrice qui contient la valeur à renvoyer.
 - valeur_proche :
 - FAUX ou omis : Recherche une correspondance exacte. Si aucune correspondance n'est trouvée, renvoie #N/A.
 - VRAI : Recherche la valeur la plus proche inférieure ou égale à valeur_cherchée. **Important : la première colonne de table_matrice doit être triée en ordre croissant.**
- **Utilité** : Permet de rechercher une valeur dans un tableau et de renvoyer une valeur correspondante située dans une autre colonne du même tableau.
- **Exemple** : =RECHERCHEV(A1;B1:C10;3;FAUX) recherche la valeur de A1 dans la colonne B (de B1 à B10). Si elle est trouvée, la fonction renvoie la valeur correspondante de la colonne C (3ème colonne). Si elle n'est pas trouvée, la fonction renvoie #N/A.

II. La fonction RECHERCHEH

- **Syntaxe** : =RECHERCHEH(valeur_cherchée; table_matrice; no_index_lig; [valeur_proche])
- **Utilité** : Similaire à RECHERCHEV, mais la recherche se fait horizontalement (dans la première ligne de table_matrice).

- **Exemple** : =RECHERCHEH (A1;B1:E2;2;VRAI) recherche la valeur de A1 dans la ligne B1 à E1. Si elle est trouvée, ou si une valeur proche inférieure est trouvée, la fonction renvoie la valeur de la ligne 2

III. Les fonctions de gestion des erreurs

- **ESTVIDE** : Vérifie si une cellule est vide.
 - **Syntaxe** : =ESTVIDE (cellule)
 - **Utilité** : Souvent utilisée avec la fonction SI pour éviter les erreurs dues à des cellules vides.
 - **Exemple** : =SI (ESTVIDE (A1) ; "" ; A1*2) - Si A1 est vide, renvoie une chaîne vide (""), sinon renvoie A1 multiplié par 2.
- **ESTNA** : Vérifie si une cellule contient l'erreur #N/A.
 - **Syntaxe** : =ESTNA (cellule)
 - **Utilité** : Souvent utilisée avec la fonction SI pour gérer les erreurs #N/A renvoyées par RECHERCHEV ou RECHERCHEH quand la valeur n'est pas trouvée.
 - **Exemple** : =SI (ESTNA (RECHERCHEV (A1;B1:C10;3;FAUX)) ; "Valeur non trouvée" ; RECHERCHEV (A1;B1:C10;3;FAUX))
- **SIERREUR** : Gère tous les types d'erreurs (#N/A, #VALEUR!, #REF!, #DIV/0!, #NOMBRE!, #NOM?, #NUL!).
 - **Syntaxe** : =SIERREUR (valeur; valeur_si_erreur)
 - **Utilité** : Permet de remplacer un message d'erreur par une valeur ou un message personnalisé.
 - **Exemple** : =SIERREUR (A1/B1;"Division par zéro") - Si A1/B1 génère une erreur (par exemple, si B1 est égal à 0), la fonction renvoie "Division par zéro", sinon elle renvoie le résultat de A1/B1.

IV. Points importants

- **table_matrice** :
 - Pour RECHERCHEV, la première colonne doit contenir les valeurs de recherche et doit être triée en ordre croissant si valeur_proche est VRAI.
 - Pour RECHERCHEH, la première ligne doit contenir les valeurs de recherche et doit être triée en ordre croissant si valeur_proche est VRAI.
 - Il est recommandé de nommer la table_matrice pour simplifier les formules.
- **valeur_proche** : Bien choisir entre VRAI et FAUX en fonction du type de recherche souhaité (exacte ou approximative).
- **Gestion des erreurs** : Il est important de gérer les erreurs pour éviter l'affichage de messages d'erreur dans les cellules et pour rendre les feuilles de calcul plus robustes.

En résumé, pour le TD3, il est essentiel de maîtriser les fonctions RECHERCHEV, RECHERCHEH, ESTVIDE, ESTNA et SIERREUR, ainsi que les concepts de recherche exacte et approximative, et de gestion des erreurs.

Voici les 20% des idées clés qui vous permettront de comprendre 80% du TD 4 sur les SGBDR :

I. Présentation d'un SGBDR

- **SGBDR (Système de Gestion de Bases de Données Relationnel)** : Logiciel qui permet de créer, organiser, gérer et interroger des bases de données relationnelles.
- **Base de données relationnelle** : Ensemble de données organisées en tables reliées entre elles.
- **Table** : Ensemble d'enregistrements (lignes) structurés en champs (colonnes).

- **Champ** : Attribut, propriété, caractéristique d'un élément de la table (ex: NomClient, Adresse, etc.).
- **Enregistrement (ou tuple)** : Ligne d'une table, correspondant à un élément unique (ex: un client spécifique).
- **Clé primaire** : Champ (ou combinaison de champs) qui identifie de manière unique chaque enregistrement d'une table.
- **Clé étrangère** : Champ d'une table qui fait référence à la clé primaire d'une autre table, créant ainsi une relation entre les tables.
- **Intégrité référentielle** : Ensemble de règles qui garantissent la cohérence des données dans une base de données relationnelle, notamment en vérifiant les liens entre les tables via les clés primaires et étrangères.

II. Les objets d'un SGBDR

- **Table** : Objet principal, contient les données.
- **Requête** : Objet qui permet d'interroger la base de données pour extraire, mettre à jour, ajouter ou supprimer des données.

III. Les requêtes

- **Définition** : Une requête est une demande d'extraction, de mise à jour ou de suppression de données.
- **Formulation** :
 - **Interface graphique QBE (Query By Example)** : On sélectionne les tables, les champs, on définit des critères et des tris de manière graphique.
 - **Langage SQL (Structured Query Language)** : Langage textuel standardisé pour interroger les bases de données.
- **Résultat** : Le résultat d'une requête est une table (un ensemble d'enregistrements).
- **Modes d'affichage** :
 - **Création (QBE)** : Pour construire ou modifier la requête.
 - **Feuille de données** : Pour visualiser le résultat.
 - **SQL** : Pour visualiser ou modifier la requête en langage SQL.
- **Utilisation du Query By Example (QBE)** :
 - **Sélection des tables** : Choisir la ou les tables nécessaires.
 - **Sélection des champs** : Choisir les champs à afficher.
 - **Définition des critères** : Spécifier les conditions que les enregistrements doivent respecter.
 - **Tri** : Définir l'ordre d'affichage des enregistrements.

IV. Points importants

- **Intégrité référentielle** :
 - Une clé étrangère doit obligatoirement correspondre à une clé primaire existante dans la table liée.
 - On ne peut pas supprimer un enregistrement dont la clé primaire est utilisée comme clé étrangère dans une autre table.
 - L'intégrité référentielle est essentielle pour maintenir la cohérence des données.
- **Clé primaire** :
 - Doit être unique pour chaque enregistrement.
 - Ne peut pas être vide (pas de valeur NULL).
- **Clé étrangère** :
 - Doit correspondre à une clé primaire existante dans une autre table.

- o Peut être vide (valeur NULL) si la relation n'est pas obligatoire.
- **Requêtes de mise à jour** : Permettent d'ajouter, de modifier ou de supprimer des données dans les tables.
- **Requêtes de sélection** : Permettent d'extraire des données selon des critères précis.

En résumé, pour le TD4, il est essentiel de comprendre les concepts de base d'un SGBDR (table, champ, enregistrement, clé primaire, clé étrangère, intégrité référentielle), de savoir formuler des requêtes simples en utilisant l'interface graphique QBE (sélection de tables, de champs, définition de critères et de tris), et de comprendre le principe de l'intégrité référentielle.

Concernant les syntaxes demandées :

- **RECHERCHEV** : =RECHERCHEV(valeur_cherchée; table_matrice; no_index_col; [valeur_proche])
- **RECHERCHEH** : =RECHERCHEH(valeur_cherchée; table_matrice; no_index_lig; [valeur_proche])
- **SI** : =SI(test_logique; valeur_si_vrai; valeur_si_faux)
- **ET** : =ET(condition1; condition2; ...)
- **OU** : =OU(condition1; condition2; ...)
- **ESTVIDE** : =ESTVIDE(cellule)
- **ESTNA** : =ESTNA(cellule)
- **SIERREUR** : =SIERREUR(valeur; valeur_si_erreur)
- **ABS** : =ABS(nombre)
- **SOMME** : =SOMME(plage_de_cellules) **ou** =SOMME(cellule1; cellule2; ...)
- **MOYENNE** : =MOYENNE(plage_de_cellules) **ou** =MOYENNE(cellule1; cellule2; ...)
- **MAX** : =MAX(plage_de_cellules) **ou** =MAX(cellule1; cellule2; ...)
- **MIN** : =MIN(plage_de_cellules) **ou** =MIN(cellule1; cellule2; ...)
- **NBVAL** : =NBVAL(plage_de_cellules) **ou** =NBVAL(cellule1; cellule2; ...)
- **NB** : =NB(plage_de_cellules) **ou** =NB(cellule1; cellule2; ...)
- **ARRONDI** : =ARRONDI(nombre; no_chiffres)
- **RANG** : =RANG(nombre; référence; [ordre])
- **MAX (pour plancher)** : =MAX(valeur; plancher)
- **MIN (pour plafond)** : =MIN(valeur; plafond)

Voici les 20% des idées clés qui vous permettront de comprendre 80% du TD 5 sur les fonctions conditionnelles :

I. Fonctions conditionnelles monocritères

- **SOMME.SI** : Additionne les cellules d'une plage qui répondent à un critère unique.
 - o **Syntaxe** : =SOMME.SI(plage; critère; [somme_plage])
 - o **plage** : Plage de cellules à tester.
 - o **critère** : Condition à respecter (ex: ">10", "=A1", "<>B2", "texte*", "texte", "**texte").
 - o **somme_plage (facultatif)** : Plage de cellules à additionner. Si omis, la plage est utilisée.
 - o **Exemple** : =SOMME.SI(A1:A10; ">10"; B1:B10) - Additionne les valeurs de B1:B10 si la valeur correspondante en A1:A10 est supérieure à 10.
- **MOYENNE.SI** : Calcule la moyenne des cellules d'une plage qui répondent à un critère unique.

- **Syntaxe** : =MOYENNE.SI(plage; critère; [moyenne_plage])
- **Exemple** : =MOYENNE.SI(A1:A10;">10";B1:B10) - Calcule la moyenne des valeurs de B1:B10 si la valeur correspondante en A1:A10 est supérieure à 10.
- **NB.SI** : Compte le nombre de cellules d'une plage qui répondent à un critère unique.
 - **Syntaxe** : =NB.SI(plage; critère)
 - **Exemple** : =NB.SI(A1:A10;">10") - Compte le nombre de cellules dans A1:A10 dont la valeur est supérieure à 10.

II. Fonctions conditionnelles multicritères

- **SOMME.SI.ENS** : Additionne les cellules d'une plage qui répondent à plusieurs critères.
 - **Syntaxe** : =SOMME.SI.ENS(somme_plage; plage_critère1; critère1; [plage_critère2; critère2]; ...)
 - somme_plage : Plage de cellules à additionner.
 - plage_critère1 : Plage de cellules à tester pour le premier critère.
 - critère1 : Condition à respecter pour le premier critère.
 - **Exemple** : =SOMME.SI.ENS(C1:C10;A1:A10;">10";B1:B10;"=Paris") - Additionne les valeurs de C1:C10 si la valeur correspondante en A1:A10 est supérieure à 10 ET la valeur correspondante en B1:B10 est égale à "Paris".
- **MOYENNE.SI.ENS** : Calcule la moyenne des cellules d'une plage qui répondent à plusieurs critères.
 - **Syntaxe** : =MOYENNE.SI.ENS(moyenne_plage; plage_critère1; critère1; [plage_critère2; critère2]; ...)
 - **Exemple** : =MOYENNE.SI.ENS(C1:C10;A1:A10;">10";B1:B10;"=Paris") - Calcule la moyenne des valeurs de C1:C10 si la valeur correspondante en A1:A10 est supérieure à 10 ET la valeur correspondante en B1:B10 est égale à "Paris".
- **NB.SI.ENS** : Compte le nombre de cellules respectant tous les critères
 - **Syntaxe** : =NB.SI.ENS(plage_critères1;critère1;plage_critères2;critère2;...)
 - **Exemple** : =NB.SI.ENS(B2:B8;">=5";C2:C8;"<10") compte le nombre de cellules comprises entre 5 et 10

III. Points importants

- **Critères** :
 - Les critères textuels doivent être entre guillemets doubles (ex: "=Paris").
 - Les opérateurs de comparaison doivent être entre guillemets (ex: ">10").
 - On peut utiliser des caractères génériques :
 - * remplace une chaîne de caractères quelconque (ex: "ABC*" commence par "ABC").
 - ? remplace un caractère unique (ex: "AB?" correspond à "ABC", "ABD", etc.).
 - On peut utiliser l'esperluette & pour concaténer des valeurs (ex: ">=&A1").
- **Plages nommées** : L'utilisation de plages nommées est fortement conseillée pour simplifier la lecture et la compréhension des formules.

En résumé, pour le TD5, il est essentiel de maîtriser les fonctions SOMME.SI, MOYENNE.SI, NB.SI, SOMME.SI.ENS, MOYENNE.SI.ENS et NB.SI.ENS, ainsi que la syntaxe des critères et l'utilisation des caractères génériques.