

République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université de Tissemsilt

Faculté des sciences économiques, commerciales et des sciences de gestion



Titre de Polycopie :

Initiation à Microsoft Access et aux méthodes de conception des systèmes d'information

Module : Informatique-2

Filière : Toutes les filières

Niveau : 2 année licence

Nom et Prénom : Safa Benaissa

Département : Sciences Financières et Comptabilité

Année Universitaire : 2025/2026

Introduction générale

1. Objectif du document

La présente polycopie a pour objectif de fournir un **cours complet, structuré et progressif** sur l'utilisation et le développement d'applications dans **Microsoft Access**, de la création d'une base de données simple à la programmation avancée en VBA (Visual Basic for Applications).

Il vise à :

- Initier les étudiants aux **principes fondamentaux des bases de données relationnelles** ;
- Former à la **conception, la création et la gestion des objets Access** (tables, formulaires, requêtes, états, menus) ;
- Développer des **compétences pratiques** en automatisation, personnalisation et intégration d'applications Access ;
- Préparer les utilisateurs à créer des **applications complètes, robustes et conviviales** destinées à un usage professionnel ou académique.

Ce document peut être utilisé comme **support de formation, référence pour enseignants**, ou **guide pratique pour étudiants** en informatique de gestion, systèmes d'information ou bureautique avancée.

2. Exigences du document

Afin de tirer le meilleur profit de ce document, certaines connaissances et ressources sont requises :

2.1. Connaissances préalables

- Notions de base en **bureautique** (Word, Excel, Windows) ;
- Compréhension élémentaire de la **logique des bases de données** (tables, champs, enregistrements, clés primaires) ;
- Facultatif : connaissance du **langage SQL** ou d'un autre SGBD (MySQL, Oracle, etc.) pour approfondir les requêtes.

2.2. Objectifs pédagogiques

À la fin du document, l'étudiant sera capable de :

1. Concevoir une base de données normalisée dans Access.
2. Créer et relier des tables.
3. Créer des formulaires interactifs et ergonomiques.
4. Construire des requêtes pour extraire et analyser des données.
5. Générer des rapports professionnels et dynamiques.
6. Développer une interface complète avec un menu principal.
7. Automatiser les tâches et gérer les événements via VBA.

2.3. Public cible

- Étudiants en informatique, bureautique ou gestion.
- Enseignants souhaitant un support pédagogique détaillé.
- Professionnels désirant automatiser la gestion de données.

3. Outils et instruments utilisés

3.1. Logiciels utilisés

Outil	Version recommandée	Rôle principal
Microsoft Access	2016 / 2019 / 2021 ou Microsoft 365	Création, gestion et programmation de la base de données
Microsoft Office Suite	2016 ou supérieur	Intégration bureautique (Word, Excel, Outlook)
Visual Basic for Applications (VBA)	Intégré à Access	Automatisation et programmation
Microsoft SQL (optionnel)	2016+	Connexion avec bases externes
Outils graphiques (Paint, Canva, etc.)	Libre	Création d'illustrations pédagogiques

3.2. Matériel requis

- Ordinateur avec système **Windows 10 ou 11**
- Processeur double cœur minimum
- 4 Go de mémoire vive (RAM) ou plus
- Espace disque : au moins **2 Go libres**
- Clavier AZERTY et souris fonctionnelle

3.3. Autres ressources utiles

- **Accès Internet** pour consultation de tutoriels et documentation Microsoft
- **Fichiers d'exemples Access** (fournis avec les exercices du document)
- **Logiciel de traitement de texte (Word)** pour rédaction et sauvegarde du rapport

4. Méthodologie d'apprentissage

Le document est organisé en chapitres progressifs, chacun comprenant :

- **Une partie théorique** pour comprendre les concepts clés ;
- **Des illustrations pas-à-pas** pour apprendre par la pratique ;
- **Des exemples concrets** tirés d'applications réelles (gestion de clients, commandes, etc.) ;
- **Des exercices corrigés** pour évaluer et renforcer la compréhension.

Chaque chapitre est conçu pour être **autonome**, tout en constituant une **étape essentielle** vers la maîtrise complète d'Access.

5. Résultats attendus

À la fin du document, l'étudiant saura :

- Concevoir, développer et sécuriser une base de données Access complète ;
- Produire des formulaires et rapports professionnels ;
- Créer une interface utilisateur complète avec un menu principal ;
- Gérer les erreurs et les interactions entre les objets de la base.

Remarque :

L'ensemble du document est conforme aux bonnes pratiques de conception de bases de données relationnelles, et s'appuie sur la documentation officielle Microsoft, ainsi que sur des exemples pédagogiques testés en situation réelle d'enseignement.

Liste des abréviations et concepts utilisés

A. Abréviations et sigles

Abréviation / Sigle	Signification	Description / Utilisation dans le document
SGBD	Système de Gestion de Base de Données	Logiciel permettant la création, la manipulation et la gestion de bases de données. Microsoft Access en est un exemple.
BD	Base de Données	Ensemble structuré de données stockées et organisées pour être exploitées.
VBA	Visual Basic for Applications	Langage de programmation intégré à Access permettant d'automatiser et de personnaliser les tâches.
SQL	Structured Query Language	Langage standard pour interroger et manipuler les données d'une base de données.
BDR	Base de Données Relationnelle	Type de base de données fondée sur la théorie relationnelle, où les données sont organisées en tables reliées par des clés.
ID	Identifiant	Clé unique permettant d'identifier un enregistrement dans une table.
Clé primaire (PK)	Primary Key	Champ unique servant à identifier de manière exclusive chaque enregistrement d'une table.
Clé étrangère (FK)	Foreign Key	Champ reliant une table à une autre via la clé primaire correspondante.
UI	User Interface	Interface utilisateur – ensemble des éléments visuels permettant à l'utilisateur d'interagir avec l'application Access.
UX	User Experience	Expérience utilisateur – qualité de l'interaction entre l'utilisateur et l'application.

Abréviation / Sigle	Signification	Description / Utilisation dans le document
OLE	Object Linking and Embedding	Technologie Microsoft permettant d'intégrer des objets d'autres applications (Excel, Word) dans Access.
DAO	Data Access Objects	Bibliothèque VBA pour manipuler les bases de données Access.
ADO	ActiveX Data Objects	Technologie Microsoft permettant l'accès à des données externes (SQL Server, Oracle, etc.).
ODBC	Open Database Connectivity	Standard pour la connexion d'Access à d'autres systèmes de bases de données.
UIA	User Interface Automation	Ensemble de techniques utilisées pour automatiser les actions sur l'interface d'Access.
ERD	Entity-Relationship Diagram	Diagramme entité-association décrivant la structure logique d'une base de données.
SRI	Système de Recherche d'Information	Système permettant la recherche et l'extraction de données spécifiques dans une base.
JET Engine	Joint Engine Technology	Moteur interne de Microsoft Access qui gère la manipulation et le stockage des données.
ACCDB	Access Database File	Format de fichier principal des bases de données Access (depuis la version 2007).
MDB	Microsoft Database	Ancien format de base de données Access (avant 2007).
UI/UX	User Interface / User Experience	Ensemble des principes de conception de l'interface et d'ergonomie dans Access.

B. Concepts clés utilisés dans le document

Concept	Définition et application
Table	Structure de base contenant les données sous forme de lignes (enregistrements) et colonnes (champs).
Champ	Colonne d'une table représentant une donnée spécifique (ex. : nom, date de naissance, prix).
Enregistrement (Record)	Ligne d'une table représentant une instance unique de données.
Formulaire (Form)	Interface graphique facilitant la saisie, la consultation ou la modification des données.
Requête (Query)	Instruction SQL permettant de sélectionner, filtrer, trier ou calculer des données.
État (Report)	Document généré à partir des données pour l'impression ou la présentation.
Menu principal	Interface de navigation centralisée pour accéder aux différents objets d'une base Access.
Macro	Séquence d'actions prédéfinies dans Access permettant d'automatiser des tâches simples.
Module VBA	Zone de code où sont écrites les procédures et fonctions en Visual Basic for Applications.
Formulaire lié / non lié	Formulaire connecté directement à une source de données (lié) ou indépendant (non lié).
Relation	Lien entre deux tables basé sur une clé primaire et une clé étrangère.
Index	Structure de données permettant d'accélérer la recherche et le tri des enregistrements.
Jointure (Join)	Association de deux tables dans une requête en fonction d'un champ commun.
Normalisation	Processus visant à structurer les données pour éviter la redondance et améliorer la cohérence.
Intégrité référentielle	Règle garantissant la cohérence des relations entre tables.

Concept	Définition et application
Interface utilisateur (IU)	Ensemble des éléments graphiques et commandes de navigation dans Access.
Base de données multi-utilisateurs	Base pouvant être utilisée simultanément par plusieurs personnes via un réseau.
Sécurité et droits d'accès	Ensemble des paramètres permettant de restreindre ou d'autoriser certaines actions selon les utilisateurs.
Expressions Access	Formules ou calculs utilisés dans des champs, requêtes ou contrôles (souvent en VBA ou SQL).
Événement (Event)	Action déclenchée par l'utilisateur ou le système (clic, modification de champ, ouverture de formulaire, etc.).
Procédure événementielle	Code VBA associé à un événement particulier (ex. : ouverture d'un formulaire).
Interface MDI / SDI	Mode d'affichage des objets Access : multiples fenêtres internes (MDI) ou séparées (SDI).

C. Notations utilisées dans les exemples

Notation	Signification
[NomChamp]	Indique un champ d'une table Access.
Form_Formulaire!NomContrôle	Représente un contrôle (zone de texte, bouton, etc.) sur un formulaire.
= [NomChamp] * [Quantité]	Exemple d'expression Access pour un calcul automatique.
SELECT ... FROM ... WHERE ...;	Syntaxe SQL d'une requête Access.
Private Sub ... End Sub	Bloc de code VBA définissant une procédure.

Table des matières

Chapitre 1 : Aperçu historique des méthodes de développement et de conception des systèmes d'information

- Introduction
- Les premières approches de conception (années 1960–1980)
- L'approche structurée et le cycle en V
- Les méthodes orientées objet
- Les approches modernes : Agiles et DevOps
- Comparatif des méthodes
- Étude de cas : évolution d'un projet de base de données
- Exercices et corrigés

Chapitre 2 : Généralités sur Microsoft Access

- Qu'est-ce qu'une base de données ?
- Présentation d'Access : interface et composants
- Création et ouverture d'une base de données
- Les objets principaux : Tables, Formulaires, Requêtes, États
- Sauvegarde et sécurité
- Bonnes pratiques pour nommer les objets
- Exercices et corrigés

Chapitre 3 : Création de tables dans une base de données

- Définition et rôle des tables
- Champs, types de données et clés primaires
- Création des tables
- Relations entre tables
- Intégrité référentielle
- Exemples et schémas
- Exercices pratiques et corrigés

Chapitre 4 : Création de formulaires

- Introduction aux formulaires
- Types de formulaires : simples, multiples, avec sous-formulaires
- Personnalisation : contrôles, couleurs, formats

- Navigation et filtres
- Exercices et corrigés

Chapitre 5 : Création de requêtes

- Introduction aux requêtes
- Requetes de sélection et critères
- Requetes de mise à jour, suppression, ajout
- Requetes paramétrées et calculées
- Introduction au SQL sous Access
- Exercices pratiques et corrigés

Chapitre 6 : Création des états

- Introduction aux états
- Création et mise en forme
- Insertion d'en-têtes, pieds de page et calculs
- Impression et exportation
- Exercices et corrigés

Chapitre 7 : Création d'un menu pour l'application

- Introduction à la navigation dans Access
- Les macros et les boutons de commande
- Création d'un menu principal
- Sécurisation et interface utilisateur
- Exercices pratiques et corrigés

Chapitre 8 : Initiation à la programmation VBA

- Introduction au langage VBA
- Accès aux objets (Formulaires, Tables, Requetes)
- Événements et procédures
- Exemples de macros VBA
- Bonnes pratiques de programmation
- Exercices et corrigés

Chapitre 1 :

Aperçu historique des méthodes de développement et de conception des systèmes d'information

1.1. Introduction générale

Depuis les débuts de l'informatique, les **systèmes d'information (SI)** ont joué un rôle central dans la collecte, le traitement et la diffusion de l'information.

Un **système d'information** peut être défini comme un ensemble organisé de ressources humaines, matérielles et logicielles permettant de traiter des données pour produire des informations utiles à la décision.

Avec le temps, les organisations ont ressenti le besoin de **structurer la conception de leurs systèmes**, afin de réduire les erreurs, d'améliorer la qualité et d'assurer la maintenance des applications informatiques. C'est ainsi qu'ont émergé différentes **méthodes de développement**, chacune adaptée à une époque, à des technologies et à des besoins précis.

1.2. Les premières approches de conception (1960–1980)

Les premières méthodes de développement étaient souvent **empiriques**.

Les programmeurs écrivaient directement du code sans documentation ni plan structuré, ce qui entraînait de nombreuses erreurs et difficultés de maintenance.

Méthode du cycle de vie classique (ou en cascade)

L'une des premières approches formalisées fut le **modèle en cascade (Waterfall Model)**, proposé dans les années 1970.

Ce modèle repose sur une succession linéaire d'étapes :

1. Analyse des besoins
2. Conception
3. Codage
4. Tests
5. Mise en production
6. Maintenance

Chaque phase ne commence qu'une fois la précédente terminée.

Avantage : simplicité et clarté.

Inconvénient : manque de flexibilité, difficile à adapter aux changements.

1.3. L'approche structurée et le cycle en V

Dans les années 1980, les entreprises ont cherché à **formaliser davantage la conception**.

Le **cycle en V** est apparu comme une amélioration du modèle en cascade.

Principe du cycle en V

- La partie gauche du « V » correspond à la **spécification** (analyse et conception).
- La partie droite du « V » correspond à la **validation** (tests et intégration).

Chaque phase de conception possède une phase de test correspondante. Cela garantit la **traçabilité** entre les besoins et les résultats.

Exemple :

Si une spécification prévoit une fonctionnalité d'authentification, un test dédié vérifiera qu'elle fonctionne comme prévu.

1.4. Les méthodes orientées objet (années 1990)

Avec l'apparition des langages comme **C++** et **Java**, la **programmation orientée objet (POO)** a transformé la façon de concevoir les systèmes.

Principes de base :

- **Encapsulation** : regrouper données et traitements dans une même entité (objet).
- **Héritage** : réutiliser les caractéristiques d'une classe existante.
- **Polymorphisme** : adapter le comportement d'un objet selon le contexte.

Des méthodes comme **UML (Unified Modeling Language)** ont permis de représenter graphiquement les objets, leurs relations et leurs interactions.

1.5. Les approches modernes : Agile et DevOps

Méthode Agile

Dans les années 2000, les méthodes agiles (Scrum, XP, Kanban) sont apparues pour répondre aux limites des approches traditionnelles. Elles se basent sur :

- des **itérations courtes** ;
- une **collaboration continue** avec le client ;
- une **adaptation rapide** aux changements.

Exemple : Dans un projet de gestion hospitalière, chaque itération pourrait produire un module fonctionnel (par ex. “gestion des patients”), testé et validé avant de passer au suivant.

DevOps

DevOps prolonge la philosophie agile en **intégrant le développement et les opérations**.

Son objectif est d'automatiser les déploiements et de maintenir une **livraison continue** des applications.

1.6. Comparatif des principales méthodes

Méthode	Période	Avantages	Inconvénients	Type de projet adapté
En cascade	1970	Simple, documentée	Rigide	Petits projets stables
Cycle en V	1980	Traçabilité, validation	Peu flexible	Projets industriels
Orientée objet	1990	Réutilisation, modularité	Complexité initiale	Applications logicielles
Agile	2000	Réactivité, satisfaction client	Difficulté de planification	Projets évolutifs
DevOps	2010	Automatisation, rapidité	Coût initial élevé	Systèmes complexes

1.7. Étude de cas : évolution d'un projet de base de données

Imaginons un projet de **base de données pour un hôpital** :

- **Années 1980** : Développement en cascade, système rigide et difficile à modifier.
- **Années 1990** : Conception orientée objet avec interfaces graphiques.
- **Années 2000** : Méthode Agile pour développer module par module (patients, médecins, rendez-vous).
- **Aujourd'hui** : Intégration continue et hébergement cloud grâce à DevOps.

1.8. Exercices et corrigés

Exercice 1 : Associer la méthode à sa caractéristique

Associez chaque méthode à sa description :

Méthode	Description
a) Cycle en V	1. Adaptée aux projets nécessitant des tests structurés
b) Agile	2. Itérative, flexible et centrée sur le client
c) Cascade	3. Linéaire, chaque phase dépend de la précédente
d) DevOps	4. Favorise l'automatisation et la collaboration continue

Correction :

$a \rightarrow 1$; $b \rightarrow 2$; $c \rightarrow 3$; $d \rightarrow 4$

Exercice 2 : QCM

- Quelle méthode privilégie la réutilisation du code ?
a) Cascade b) Orientée objet c) Agile d) DevOps
☐ Réponse : b)
- Quelle méthode repose sur des itérations courtes ?
a) Agile b) Cascade c) Cycle en V d) DevOps
☐ Réponse : a)

Exercice 3 : Réflexion

Dans votre propre domaine, quelle méthode serait la plus efficace pour gérer un projet informatique ?

Justifiez votre choix en quelques lignes.

Chapitre 2 :

Généralités sur Microsoft Access

2. 1. Introduction à Microsoft Access

Microsoft Access est un **Système de Gestion de Base de Données Relationnelle (SGBDR)** développé par Microsoft. Il permet de créer, gérer et interroger des bases de données de manière graphique et intuitive. Access est souvent utilisé pour des applications de petite à moyenne taille, où Excel ne suffit pas et où des solutions comme SQL Server seraient trop complexes.

2.1.1 Concepts et Définitions

- **Donnée :**

Une **donnée** est description élémentaire d'une information.

- **Information :**

- Plusieurs données regroupées et se rapportant à un même contexte donnent naissance à l'**information**.
- L'**information** est tout renseignement, écrit, sonore, visuel ou audiovisuel, codé susceptible d'être stocké ou transmis, en vue de déclencher ou de modifier le comportement d'un processus.

- **Fichier :**

Un **fichier** est une suite d'informations codé sous forme binaire.

- **Base de données :**

Une base de données est une collection de données structurées relatives à un ou plusieurs domaines du monde réel.

Exemple : BD d'une bibliothèque.

Les avantages de la base de données sont :

- **Centralisation** : Les données peuvent être utilisées par plusieurs programmes et plusieurs utilisateurs.
- **Indépendance entre données et programmes** : Dans une BD les données sont décrites indépendamment des programmes. Ce qui n'est pas le cas avec les fichiers.
- **Intégration des liaisons entre les données** : Pas besoin d'un programme pour retrouver les liens entre les données.
- **Intégrité des données** : Ce sont des règles de sécurité assurant la cohérence des données :
 1. Unicité des enregistrements.
 2. Interdiction de la suppression des données utilisées par d'autres données.
- **Concurrence d'accès** : Plusieurs utilisateurs peuvent accéder simultanément à la BD.

▪ Système de Gestion de Bases de données (SGBD) :

Un Système de Gestion de Base de Données(SGBD) est un logiciel qui permet de : décrire, modifier, interroger et administrer les données d'une base de données.

- Rôle de SGBD :

Le SGBD joue le rôle d'interface entre l'utilisateur et la base de données

- Exemple de SGBD :

- ORACLE
- MICROSOFT SQL SERVER
- MySQL
- Microsoft Access

- Fonctions de SGBD :

- La définition des données

Le SGBD nous permet de créer et de décrire les objets de la base de données (table, liens, utilisateur...), grâce au Langage de Description de Données (LDD).

- La manipulation des données

La manipulation des données peut être :

- ✓ La recherche
- ✓ La lecture
- ✓ La suppression
- ✓ La modification
- ✓ L'ajout

- L'intégrité des données

C'est l'ensemble des opérations de contrôle que le SGBD effectue pour préserver la cohérence des données.

Exemple : Vérification de la validité de la valeur d'un champ.

- La gestion des accès concurrents

Le SGBD gère l'accès simultané des utilisateurs à la base de données.

- La confidentialité

Tous les utilisateurs d'une base de données ne sont pas supposés pouvoir consulter toutes les informations. Des sous schémas de la base permettent de résoudre ce problème en plus des mots de passes et droits d'accès.

- La sécurité de fonctionnement

Faire une copie de sauvegarde de la base. Remise en marche de la base en cas de panne.

2.1.2 Objectifs de Microsoft Access

- Stocker de grandes quantités de données.
- Faciliter l'organisation et la manipulation des informations.
- Permettre la création d'applications simples pour gérer des données.
- Fournir des outils de requêtes et de rapports puissants sans programmation complexe.

2.2. Concepts fondamentaux de MS-Access

2.2.1 Base de données

Une **base de données** est un ensemble structuré de données permettant leur

stockage, manipulation et exploitation.

2.2.2 Table

- La table est l'**élément de base** d'Access.
- Contient des **enregistrements** (lignes) et des **champs** (colonnes).

2.2.3 Champ

- Représente un attribut ou une information particulière.
- Exemples : Nom, Prénom, Date de naissance.
- Chaque champ a un **type de données** : Texte, Numérique, Date/Heure, Oui/Non, etc.

2.2.4 Clé primaire

- Identifie **uniquement chaque enregistrement** d'une table.
- Exemple : Numéro d'étudiant, Code produit.

2.2.5 Relation

- Liaison entre plusieurs tables.
- Types : un-à-un, un-à-plusieurs, plusieurs-à-plusieurs.
- Permet de **réduire la redondance** et d'optimiser la base.

2.3. Les objets Access

Access contient plusieurs objets essentiels :

1. **Tables** : stockent les données.
2. **Requêtes** : extraient et manipulent des données.
3. **Formulaires** : interface utilisateur pour saisir et visualiser les données.
4. **Rapports** : impression et synthèse des informations.
5. **Macros** : automatisation des tâches simples.
6. **Modules VBA** : programmation avancée.

2.4. Types de données dans Access

Type de données	Description
Texte court	Jusqu'à 255 caractères

Type de données	Description
Texte long (Mémo)	Texte long, jusqu'à 65 536 caractères
Numérique	Valeurs numériques
Monétaire	Valeurs financières
Date/Heure	Dates et heures
Oui/Non	Booléen (Vrai/Faux)
Pièce jointe	Fichiers (images, documents)
Hyperlien	Liens vers un site web ou fichier
Assistant	Aide pour types spéciaux (ex : code postal)

2.5. Création d'une base de données Access

Étape 1 : Lancer Microsoft Access

- Choisir **Base de données vide**.
- Nommer et enregistrer le fichier.

Étape 2 : Créer une table

- Ouvrir la vue **Création**.
- Définir les **champs** et leurs types.
- Définir la **clé primaire**.

Étape 3 : Saisie des données

- Utiliser la **feuille de données**.
- Saisir les informations ligne par ligne.

2.6. Relations entre tables

2.6.1 Création de relations

- Aller dans **Outils de base de données → Relations**.
- Glisser-déposer les champs à relier.
- Définir le type de relation : un-à-plusieurs, etc.
- Activer l'**intégrité référentielle** pour éviter les erreurs.

2.6.2 Avantages

- Maintien de la cohérence des données.
- Évite la duplication inutile.

2.7. Les requêtes

Les requêtes permettent de **filtrer, trier et calculer des données**.

2.7.1 Types de requêtes

1. **Sélection** : extraire des informations.
2. **Action** : modifier les données (ajouter, supprimer, mettre à jour).
3. **Paramètre** : demander une valeur avant exécution.
4. **Croisée** : synthèse et regroupement.

2.7.2 Exemple : Requête de sélection

- Objectif : afficher tous les étudiants de plus de 20 ans.
- SQL :

```
SELECT Nom, Prénom, Age  
FROM Etudiants  
WHERE Age > 20;
```

2.8. Les formulaires

2.8.1 Création

- Vue création → choisir un style de formulaire.
- Ajouter champs et contrôles (boutons, zones de texte).

2.8.2 Utilité

- Facilite la **saisie et la consultation**.
- Peut inclure des **listes déroulantes et boutons de navigation**.

2.9. Les rapports

- Permettent de **présenter les données sous forme imprimable**.
- Possibilité de grouper et de trier les informations.
- Exemple : rapport de ventes mensuelles.

2.10. Automatisation avec macros

- Les **macros** simplifient les tâches répétitives.
- Exemple : ouvrir un formulaire, appliquer un filtre, générer un rapport.
- Pour des fonctionnalités avancées, on utilise le **VBA**.

2.11. Bonnes pratiques

1. Toujours définir une **clé primaire**.
2. Utiliser des **types de données adaptés**.
3. Normaliser les tables pour éviter la redondance.
4. Sauvegarder régulièrement la base.
5. Documenter les relations et les requêtes importantes.

2.12. Exercices corrigés

Exercice 1 : Création d'une table

Énoncé : Créez une table Etudiants avec les champs : ID (Numérique, clé primaire), Nom (Texte), Prénom (Texte), DateNaissance (Date/Heure), Filière (Texte).

Correction :

- Ouvrir Access → Nouvelle base → Table en vue création.
- Créer les champs :
 - ID → Numérique → clé primaire
 - Nom → Texte court
 - Prénom → Texte court
 - DateNaissance → Date/Heure
 - Filière → Texte court

Exercice 2 : Requête de sélection

Énoncé : Afficher les étudiants dont la filière est "Informatique".

Correction :

```
SELECT Nom, Prénom, Filière  
FROM Etudiants  
WHERE Filière = 'Informatique';
```

Exercice 3 : Formulaire

Énoncé : Créer un formulaire pour saisir des étudiants.

Correction :

- Créer → Formulaire → Sélectionner table Etudiants.
- Ajouter champs : ID, Nom, Prénom, DateNaissance, Filière.
- Ajouter bouton “Enregistrer”.

Exercice 4 : Relation entre tables

Énoncé : Créer une table Cours et relier Etudiants à Cours (un étudiant peut suivre plusieurs cours).

Correction :

- Table Cours : ID_Cours (clé primaire), NomCours.
- Dans Relations → glisser ID de Etudiants vers ID_Etudiant de Cours.
- Type relation : un-à-plusieurs.
- Activer intégrité référentielle.

Exercice 5 : Rapport

Énoncé : Générer un rapport listant tous les étudiants par filière.

Correction :

- Créer → Rapport → sélectionner Etudiants.
- Grouper par Filière.
- Ajouter titres et pied de page.

Chapitre 3 :

Création de tables dans une base de données

3.1 Introduction aux tables

Les tables sont le **cœur d'une base de données Access**. Elles permettent de stocker toutes les informations de manière structurée et normalisée.

Objectifs :

- Comprendre le rôle des tables dans Access
- Créer des tables et définir leurs champs et types de données
- Mettre en place des clés primaires et des index
- Comprendre la normalisation pour éviter les doublons.

3.2 Types de champs

Type	Description	Exemple
Texte court	Jusqu'à 255 caractères	Nom, Prénom
Texte long	Jusqu'à 65 536 caractères	Description détaillée
Numérique	Valeurs numériques	ID_Client, Quantité
Monétaire	Valeurs financières	Prix
Date/Heure	Dates et heures	DateCommande
Oui/Non	Valeur Vrai/Faux	Payé / Non payé
Pièce jointe	Fichiers (images, documents)	Photo, PDF
Hyperlien	Lien vers site web ou fichier	www.exemple.com

Illustration : Imagine un formulaire où l'on peut sélectionner le type de champ pour chaque colonne.

3.3 Clé primaire et index

- **Clé primaire** : identifie chaque enregistrement de manière unique.
- **Index** : accélère les recherches et le tri des données.

Exemple : Pour la table Clients, la clé primaire est ID_Client.

ID_Client	Nom
1	Abidi
2	Bouslimi

3.4 Création d'une table

Étapes pour créer une table :

1. Ouvrir Access → Vue Création
2. Définir les champs et leurs types
3. Définir la clé primaire
4. Enregistrer la table
5. Saisir les données

Exemple Table Clients :

ID	Nom	Prénom	Email
1	Abidi	Walid	Walid@ex.com
2	Bouslimi	Ali	Ali@ex.com

Exemple Table Commandes :

ID	ID_Client	Date	Total
1	1	01/10/2025	250
2	2	02/10/2025	180

3.5 Normalisation

La **normalisation** permet :

- D'éviter les doublons et la redondance des données
- De simplifier les mises à jour et la maintenance
- D'améliorer la cohérence des informations

Exemple : Séparer les informations clients dans Clients et les commandes dans Commandes avec une relation un-à-plusieurs.

3.6 Relations entre tables

- **Un-à-un** : un enregistrement correspond à un seul autre enregistrement.
- **Un-à-plusieurs** : un enregistrement d'une table correspond à plusieurs enregistrements d'une autre table.
- **Plusieurs-à-plusieurs** : relation via une table intermédiaire.

Exemple : Relation Clients → Commandes (un client peut avoir plusieurs commandes).

Clients.ID ----< Commandes.ID_Client

3.7 Exercices corrigés

Exercice 1 :

Créer la table Clients avec les champs ID_Client, Nom, Prénom, Email.

Correction :

Vue Création → définir champs et types → clé primaire = ID_Client.

Exercice 2 :

Créer la table Commandes avec champs ID_Commande, DateCommande, ID_Client, Total.

Correction :

Vue Création → définir champs et types → clé primaire = ID_Commande.

Exercice 3 :

Créer une relation un-à-plusieurs entre Clients et Commandes.

Correction :

Ouvrir **Outils de base de données** → **Relations**, relier ID_Client de Clients à ID_Client de Commandes et activer l'intégrité référentielle.

Exercice 4 :

Ajouter un index sur le champ Nom de la table Clients.

Correction :

Vue Création → sélectionner champ Nom → Propriétés → Index = Oui (Non doublon si nécessaire).

Exercice 5 :

Saisir au moins 5 enregistrements dans chaque table et vérifier la relation.

Correction :

Remplir les données et tester une requête simple :

```
SELECT Clients.Nom, Clients.Prénom, Commandes.Total  
FROM Clients INNER JOIN Commandes  
ON Clients.ID = Commandes.ID_Client;
```

Chapitre 4 :

Création de formulaires

4.1 Introduction aux formulaires

Les **formulaires** dans Microsoft Access sont utilisés pour :

- La saisie des données
- La consultation facile des informations
- L'amélioration de l'interface utilisateur

Avantages :

- Interface conviviale et ergonomique
- Réduction des erreurs de saisie
- Automatisation de certaines actions via des boutons ou macros

4.2 Création simple de formulaire

Étapes :

1. Ouvrir Access et sélectionner la table ou la requête à utiliser.
2. Aller dans **Créer** → **Assistant Formulaire** pour générer un formulaire automatiquement.
3. Choisir les champs à inclure.
4. Cliquer sur **Terminer** pour créer le formulaire.

Alternative : Vue Création

- Permet de personnaliser l'apparence du formulaire.
- Ajouter des contrôles, images, titres et couleurs.

4.3 Contrôles dans les formulaires

Contrôle	Utilité
Zone de texte	Saisie de données textuelles
Liste déroulante	Sélection d'options prédéfinies
Bouton	Exécuter des actions (ajouter, supprimer)
Case à cocher	Valeur Oui/Non
Image	Afficher une image ou logo
Étiquette	Afficher un texte explicatif

Illustration : Formulaire Form_Client avec zones de texte pour Nom, Prénom, Email, liste déroulante pour Filière et bouton “Enregistrer”.

4.4 Navigation et automatisation

- **Navigation :**
 - Boutons “Précédent”, “Suivant”, “Premier” et “Dernier” pour se déplacer entre les enregistrements.
- **Automatisation avec macros :**
 - Ajouter un bouton pour ouvrir un rapport ou filtrer les données automatiquement.
 - Exemple : Bouton “Afficher commandes client” qui exécute une requête et ouvre le rapport correspondant.

4.5 Mise en forme et ergonomie

- Organiser les champs de manière logique
- Grouper les champs connexes
- Ajouter des couleurs et polices lisibles
- Insérer des sections pour les titres et informations complémentaires

4.6 Exercices corrigés

Exercice 1 : Créer un formulaire Form_Client avec les champs : ID_Client, Nom, Prénom, Email.

Correction : Utiliser l’Assistant Formulaire → sélectionner les champs →

terminer → vérifier que les champs s'affichent correctement.

Exercice 2 : Ajouter un bouton “Ajouter” pour insérer un nouvel enregistrement.

Correction :

- Vue Création → Ajouter un bouton → Action : “Ajouter un enregistrement”

Exercice 3 : Ajouter un bouton “Supprimer” pour supprimer l'enregistrement courant.

Correction :

- Vue Création → Ajouter un bouton → Action : “Supprimer un enregistrement”

Exercice 4 : Ajouter une liste déroulante pour le champ Filière avec les options : Info, Math, Physique.

Correction :

- Ajouter un contrôle Liste déroulante → Définir source des valeurs → “Info;Math;Physique”

Exercice 5 : Ajouter un bouton “Afficher commandes” pour ouvrir le rapport Rapport_Commandes filtré par client.

Correction :

- Bouton → Action : “Ouvrir rapport” → Sélectionner Rapport_Commandes → Définir filtre basé sur ID_Client

Chapitre 5 :

Création de requêtes

5.1 Introduction aux requêtes

Les **requêtes** dans Microsoft Access permettent :

- D'extraire des informations spécifiques depuis une ou plusieurs tables
- De filtrer, trier et regrouper les données
- D'effectuer des calculs et synthèses sur les données

Avantages :

- Rapide et précis pour l'analyse des données
- Permet la création de rapports dynamiques
- Facilite la prise de décision en filtrant les informations pertinentes

5.2 Types de requêtes

Type	Fonction principale
Sélection	Extraire des données répondant à des critères
Action	Ajouter, mettre à jour ou supprimer des enregistrements
Paramètre	Demander une valeur avant l'exécution
Croisée	Résumé et regroupement des données
SQL	Écrire des requêtes documentement en langage SQL

5.3 Création d'une requête simple

Étapes :

1. Ouvrir Access → Créer → Requête en mode Création
2. Ajouter la ou les tables concernées
3. Sélectionner les champs à afficher

4. Définir les critères de filtrage
5. Exécuter la requête pour afficher les résultats

Exemple : Afficher les étudiants de la filière “Informatique”.

```
SELECT Nom, Prénom, Filière  
FROM Etudiants  
WHERE Filière = 'Info';
```

Illustration : Fenêtre Access avec la grille de requête et colonnes Nom, Prénom, Filière, et zone de critères = “Info”.

5.4 Requêtes multi-tables

- Permettent de combiner des informations provenant de plusieurs tables liées.
- Exemple : lister les commandes avec le nom du client.

```
SELECT Clients.Nom, Clients.Prénom, Commandes.Total  
FROM Clients INNER JOIN Commandes  
ON Clients.ID = Commandes.ID_Client;
```

Explication :

- INNER JOIN relie les tables Clients et Commandes selon la clé ID_Client.
- Seuls les clients ayant des commandes apparaissent.

5.5 Requêtes paramétrées

- Permettent de **demandeur une valeur à l'utilisateur** avant l'exécution.
- Exemple : afficher les commandes d'un client choisi au moment de l'exécution.

```
SELECT *  
FROM Commandes  
WHERE ID_Client = [Entrez l'ID du client];
```

- La requête affiche uniquement les commandes du client spécifié.

5.6 Requêtes d'action

- **Mise à jour** : modifier plusieurs enregistrements à la fois
- **Suppression** : supprimer des enregistrements selon des critères
- **Ajout** : ajouter des enregistrements à une autre table
- **Création de table** : générer une nouvelle table à partir d'une sélection

Exemple : Supprimer toutes les commandes de clients inactifs

```
DELETE FROM Commandes  
WHERE ID_Client IN (SELECT ID_Client FROM Clients WHERE  
Actif=0);
```

5.7 Requêtes de synthèse (agrégées)

- Permettent de calculer : Somme, Moyenne, Minimum, Maximum, Compte
- Utilisation de la **ligne Totaux** dans Access

Exemple : Total des commandes par client

```
SELECT ID_Client, SUM(Total) AS Total_Commandes  
FROM Commandes  
GROUP BY ID_Client;
```

Illustration : Table affichant ID_Client et Total_Commandes.

5.8 Exercices corrigés

Exercice 1 : Créer une requête pour afficher les étudiants de la filière "Math".

Correction :

```
SELECT Nom, Prénom, Filière  
FROM Etudiants  
WHERE Filière = 'Math';
```

Exercice 2 : Créer une requête pour afficher toutes les commandes supérieures à 200.

Correction :

```
SELECT * FROM Commandes  
WHERE Total > 200;
```

Exercice 3 : Créer une requête multi-tables pour afficher le nom des clients et le total de leurs commandes.

Correction :

```
SELECT Clients.Nom, Clients.Prénom, SUM(Commandes.Total) AS  
Total_Commandes  
FROM Clients INNER JOIN Commandes  
ON Clients.ID = Commandes.ID_Client  
GROUP BY Clients.Nom, Clients.Prénom;
```

Exercice 4 : Créer une requête paramétrée demandant l’ID du client et affichant ses commandes.

Correction :

```
SELECT * FROM Commandes  
WHERE ID_Client = [Entrez l'ID du client];
```

Exercice 5 : Créer une requête d’action pour ajouter toutes les commandes d’octobre à une table Commandes_Octobre.

Correction :

```
SELECT * INTO Commandes_Octobre  
FROM Commandes  
WHERE Month(DateCommande) = 10;
```

Chapitre 6 :

Création des états (rapports)

6.1 Introduction aux états

Les **états (ou rapports)** dans Microsoft Access permettent de présenter les données de manière **lisible et imprimable**. Ils sont utiles pour :

- Générer des synthèses pour la prise de décision
- Imprimer des listes, factures, bulletins ou statistiques
- Organiser les informations de façon structurée avec des regroupements et tris

Différence entre formulaire et état :

Formulaire	État (Rapport)
Interactif, pour saisie	Statique, pour impression ou consultation
Navigation entre enregistrements	Présentation globale des données
Utilisé pour saisie ou modification	Utilisé pour synthèse ou rapport final

6.2 Types d'états

1. **Rapport simple** : Affiche toutes les données d'une table ou d'une requête.
2. **Rapport groupé** : Regroupe les données selon un ou plusieurs champs (ex. : par client, par département).
3. **Rapport trié** : Classe les données selon un ordre précis (ex. : alphabétique, montant décroissant).
4. **Rapport avec calculs** : Total, moyenne, minimum, maximum sur des champs numériques.
5. **Rapport combiné** : Utilise regroupement et calculs simultanément pour des synthèses complètes.

6.3 Création d'un état simple

Étapes :

1. Ouvrir Access → Onglet **Créer** → **Assistant Rapport**
2. Sélectionner la table ou la requête source (ex. : Etudiants ou Commandes)
3. Sélectionner les champs à inclure
4. Choisir le style et la mise en page (tabulaire, colonne)
5. Cliquer sur **Terminer** pour générer le rapport

Illustration :

- Rapport tabulaire affichant Nom, Prénom, Filière, DateNaiss
- Titres de colonnes en gras et bordures de tableau visibles

6.4 Création d'un rapport groupé

Objectif : Organiser les données par catégorie, par exemple regrouper les étudiants par Filière.

Étapes :

1. Ouvrir l'assistant rapport → sélectionner la source
2. Cliquer sur **Grouper par** et choisir le champ Filière
3. Ajouter des totaux ou sous-totaux si nécessaire (ex. : nombre d'étudiants par filière)
4. Choisir la mise en page et le style
5. Cliquer sur **Terminer**

Illustration :

Filière : Info

Abidi Walid 01/01/2000

Bouslimi Alice 12/02/2001

Total étudiants : 2

6.5 Ajout de tris et filtres

- **Tri** : Par ordre croissant ou décroissant d'un champ (ex. : Nom, Date)
- **Filtre** : Afficher uniquement certains enregistrements (ex. : étudiants nés après 2000)

Exemple SQL pour un rapport filtré :

```
SELECT Nom, Prénom, Filière, DateNaiss  
FROM Etudiants  
WHERE Year(DateNaiss) >= 2000  
ORDER BY Nom ASC;
```

6.6 Ajout de calculs dans un rapport

Calculs possibles :

- Total, Moyenne, Minimum, Maximum, Compte
- Se fait via **Ligne Totaux** ou **Zone de texte calculée**

Exemple : Total des commandes par client dans un rapport groupé

=Sum([Total])

Illustration :

Client : Abidi Walid

Commandes :

01/10/2025	250
15/10/2025	180
Total	430

6.7 Mise en forme avancée

- Ajouter **en-têtes** et **pieds de page** : logo, titre du rapport, date
- Insérer **images** ou logos pour personnaliser le rapport
- Utiliser **couleurs et polices** pour hiérarchiser les informations

- Ajouter des **sections** : titre, détail, résumé, total général

Conseil pédagogique :

- Toujours vérifier la lisibilité avant impression
- Regrouper et trier pour faciliter l'analyse

6.8 Automatisation des rapports

- Créer un **bouton dans un formulaire** pour ouvrir un rapport prédéfini
- Utiliser des macros pour :
 - Filtrer les données selon un critère
 - Imprimer directement le rapport
 - Générer des rapports quotidiens ou mensuels

Exemple macro simple :

DoCmd.OpenReport "Rapport_Commandes", acViewPreview, , "Total>200"

6.9 Exercices corrigés

Exercice 1 : Créer un rapport simple affichant tous les étudiants avec leurs filières.

Correction :

- Assistant Rapport → Table Etudiants → Tous les champs → Style tabulaire → Terminer

Exercice 2 : Créer un rapport groupé par filière et afficher le nombre d'étudiants par filière.

Correction :

- Assistant Rapport → Table Etudiants → Grouper par Filière → Ligne Totaux → Compter les étudiants

Exercice 3 : Trier le rapport par Nom croissant.

Correction :

- Cliquer sur Nom → Tri croissant dans la vue création du rapport

Exercice 4 : Créer un rapport filtré affichant uniquement les étudiants nés après 2000.

Correction :

- Critère : $\geq$ #01/01/2000# dans le champ DateNaiss
- Exécuter le rapport → seules les lignes correspondantes s'affichent

Exercice 5 : Ajouter un bouton dans un formulaire pour ouvrir le rapport Rapport_Commandes avec filtrage sur les commandes > 200 .

Correction :

- Bouton → Action : “Ouvrir rapport” → Sélectionner Rapport_Commandes
- Définir filtre : Total >200

Chapitre 7 :

Création d'un menu pour l'application

7.1 Introduction aux menus

Un **menu** dans Microsoft Access sert à :

- Faciliter la navigation entre différents objets (tables, formulaires, rapports)
- Centraliser les actions principales de l'application
- Améliorer l'ergonomie et l'accessibilité pour l'utilisateur

Avantages :

- Réduit le risque d'erreurs de manipulation
- Accès rapide aux fonctionnalités essentielles
- Professionnalise l'interface de la base de données

Exemple

:

Menu principal avec boutons :

- Gestion des étudiants
- Gestion des commandes
- Consultation des rapports
- Quitter l'application

7.2 Méthodes pour créer un menu

7.2.1 Formulaire-menu simple

1. **Créer un formulaire vierge** : Onglet **Créer** → **Formulaire vierge**
2. **Ajouter un titre** : "Menu Principal"
3. **Ajouter des boutons** pour chaque action :
 - Ouvrir un formulaire (Form_Etudiants, Form_Commandes)

- Ouvrir un rapport (Rapport_Etudiants, Rapport_Commandes)
- Quitter l'application

Illustration :

Menu Principal
[Gestion Etudiants]
[Gestion Commandes]
[Rapports]
[Quitter]

7.2.2 Utilisation des macros pour les boutons

- **Macro simple** : Ouvrir un formulaire ou un rapport
 1. Sélectionner le bouton → Propriétés → Événement Au clic → Générer une macro
 2. Choisir **OuvrirFormulaire** ou **OuvrirRapport**
 3. Définir le formulaire ou rapport à ouvrir

Exemple :

- Bouton “Gestion Etudiants” → Action : DoCmd.OpenForm "Form_Etudiants"
- Bouton “Rapports” → Action : DoCmd.OpenReport "Rapport_Etudiants", acViewPreview
- Bouton “Quitter” → Action : DoCmd.Quit

7.2.3 Personnalisation du menu

- Ajouter **icônes ou images** pour chaque bouton
- Modifier la **couleur et la police** pour hiérarchiser les options
- Organiser les boutons selon l'ordre d'utilisation fréquent

Astuce :

- Les formulaires peuvent être dimensionnés pour que le menu apparaisse toujours à l'ouverture de la base de données.

7.3 Navigation entre formulaires et rapports

- Bouton pour ouvrir un formulaire spécifique
- Bouton pour ouvrir un rapport filtré selon un critère
- Retour automatique au menu après fermeture d'un formulaire ou rapport

Exemple VBA simple pour retour au menu :

```
Private Sub Form_Close()  
    DoCmd.OpenForm "Menu_Principal"  
End Sub
```

7.4 Gestion de la sécurité

- Restreindre l'accès à certaines fonctions pour certains utilisateurs
- Masquer ou désactiver des boutons selon les droits
- Exemple : cacher le bouton "Supprimer" pour les utilisateurs standard

7.5 Exercices corrigés

Exercice 1 : Créer un formulaire-menu principal avec les boutons :

- Gestion Étudiants
- Gestion Commandes
- Rapports
- Quitter

Correction :

- Formulaire vierge → ajouter quatre boutons → définir actions respectives (DoCmd.OpenForm ou DoCmd.OpenReport)

Exercice 2 : Ajouter des icônes aux boutons pour améliorer l'ergonomie.

Correction :

- Propriétés du bouton → Image → Choisir une icône → Ajuster taille

Exercice 3 : Ajouter un bouton “Afficher commandes supérieures à 200” qui ouvre le rapport filtré.

Correction :

- Bouton → Macro → OuvrirRapport → Rapport_Commandes → Filtre : Total>200

Exercice 4 : Créer une navigation entre formulaire “Form_Etudiants” et le menu principal.

Correction :

- Ajouter bouton “Retour” sur Form_Etudiants → Action : DoCmd.OpenForm "Menu_Principal"

Exercice 5 : Ajouter une condition pour masquer le bouton “Supprimer” si l'utilisateur n'a pas les droits d'administrateur.

Correction :

- Événement Au chargement du formulaire :

```
If CurrentUser <> "Admin" Then  
    Me.BTN_Supprimer.Visible = False  
End If
```

7.6 Conseils pratiques

- Toujours tester les boutons et la navigation avant déploiement
- Nommer Alimenter les boutons pour éviter toute confusion
- Utiliser une combinaison de macros et de VBA pour les actions avancées
- Penser à la lisibilité et à l'ergonomie pour l'utilisateur final

Chapitre 8 :

Initiation à la programmation VBA

8.1 Introduction à VBA

VBA (Visual Basic for Applications) est le langage de programmation intégré à Microsoft Access, permettant :

- L'automatisation des tâches répétitives
- La personnalisation avancée des formulaires et rapports
- La création de fonctions et procédures pour gérer les données

Avantages :

- Automatisation complète des actions Access
- Possibilité de créer des applications professionnelles
- Interaction avec d'autres applications Microsoft Office (Excel, Outlook, Word)

Différence entre macros et VBA :

Aspect	Macro	VBA
Complexité	Simple	Avancée
Contrôle	Limitée	Complet
Syntaxe	Interface graphique	Langage de programmation
Exemples	Ouvrir formulaire, exécuter requête	Vérification de champ, boucle, calcul

8.2 L'environnement VBA

Pour accéder à l'éditeur VBA :

1. Onglet **Développeur** ou **Créer** → **Macro** → **Visual Basic**
2. Fenêtre **Projet** : liste tous les objets Access (tables, formulaires, rapports)
3. Fenêtre **Code** : saisir le code VBA
4. Fenêtre **Propriétés** : modifier les propriétés des objets

Illustration :

Fenêtre VBA avec projet Base_Exemple → modules, formulaires, rapports.

8.3 Écriture d'une procédure simple

Exemple 1 : Message d'accueil à l'ouverture d'un formulaire

```
Private Sub Form_Open(Cancel As Integer)
    MsgBox "Bienvenue dans l'application Access"
End Sub
```

- Private Sub Form_Open : événement exécuté à l'ouverture du formulaire
- MsgBox : affiche une boîte de message

8.4 Variables et types

Déclaration de variables :

```
Dim Nom As String
Dim Age As Integer
Dim Prix As Double
Dim DateCommande As Date
```

Types principaux :

- String → texte
- Integer → entier
- Double → nombre décimal

- Date → date/heure
- Boolean → Vrai/Faux

Exemple d'utilisation :

Nom = "Abidi"

Age = 25

MsgBox "Nom : " & Nom & ", Age : " & Age

8.5 Structures conditionnelles**If...Then...Else :**

If Age >= 18 Then

 MsgBox "Adulte"

Else

 MsgBox "Mineur"

End If

Select Case :

Select Case Filiere

 Case "Info"

 MsgBox "Filière Informatique"

 Case "Math"

 MsgBox "Filière Mathématiques"

 Case Else

 MsgBox "Autre filière"

End Select

8.6 Boucles**Boucle For :**

Dim i As Integer

For i = 1 To 5

 MsgBox "Itération : " & i

Next i

Boucle While :

```
Dim compteur As Integer
compteur = 1
While compteur <= 5
    MsgBox "Itération : " & compteur
    compteur = compteur + 1
Wend
```

Boucle For Each (pour parcourir les enregistrements) :

```
Dim ctl As Control
For Each ctl In Me.Controls
    ctl.BackColor = RGB(200, 200, 255)
Next ctl
```

8.7 Gestion des événements

Les événements sont déclenchés par des actions de l'utilisateur ou du système:

- On Click → clic sur un bouton
- On Load → chargement du formulaire
- On Current → déplacement sur un enregistrement
- On Change → modification d'un champ

Exemple : Vérification de champ vide à l'ajout d'un enregistrement

```
Private Sub BTN_Ajouter_Click()
    If IsNull(Me.TXT_Nom) Or Me.TXT_Nom = "" Then
        MsgBox "Veuillez saisir le nom"
        Exit Sub
    End If
    Me.Dirty = False ' Enregistrer l'enregistrement
End Sub
```

8.8 Accès aux données

- **Objet Recordset** : permet de manipuler les données directement depuis

VBA.

Exemple : Parcourir les étudiants

```
Dim rs As DAO.Recordset
Set rs = CurrentDb.OpenRecordset("Etudiants")
While Not rs.EOF
    MsgBox rs!Nom & " " & rs!Prénom
    rs.MoveNext
Wend
rs.Close
Set rs = Nothing
```

- **DoCmd** : exécute les actions Access

```
DoCmd.OpenForm "Form_Etudiants"
DoCmd.OpenReport "Rapport_Commandes", acViewPreview
DoCmd.Close acForm, "Form_Etudiants"
```

8.9 Gestion des erreurs

Exemple :

```
On Error GoTo GestionErreur
Dim x As Integer
x = 5 / 0 ' Erreur division par zéro
Exit Sub
GestionErreur:
MsgBox "Erreur détectée : " & Err.Description
Resume Next
```

- Permet d'éviter le plantage de l'application
- Affiche un message informatif à l'utilisateur

8.10 Exercices corrigés

Exercice 1 : Créer un message d'accueil à l'ouverture du formulaire Form_Etudiants.

Correction :

```
Private Sub Form_Open(Cancel As Integer)
    MsgBox "Bienvenue dans le formulaire Étudiants"
End Sub
```

Exercice 2 : Vérifier qu'un champ TXT_Nom n'est pas vide avant d'ajouter un étudiant.

Correction :

```
Private Sub BTN_Ajouter_Click()
    If IsNull(Me.TXT_Nom) Or Me.TXT_Nom = "" Then
        MsgBox "Veuillez saisir le nom"
        Exit Sub
    End If
    Me.Dirty = False
End Sub
```

Exercice 3 : Parcourir tous les étudiants et afficher leur nom et prénom.

Correction :

```
Dim rs As DAO.Recordset
Set rs = CurrentDb.OpenRecordset("Etudiants")
While Not rs.EOF
    MsgBox rs!Nom & " " & rs!Prénom
    rs.MoveNext
Wend
rs.Close
Set rs = Nothing
```

Exercice 4 : Créer un bouton “Afficher rapport” qui ouvre Rapport_Commandes filtré pour les commandes > 200.

Correction :

```
Private Sub BTN_Rapport_Click()
    DoCmd.OpenReport "Rapport_Commandes", acViewPreview, ,
```

"Total>200"

End Sub

Exercice 5 : Ajouter une gestion d'erreur pour empêcher la division par zéro.

Correction :

On Error GoTo GestionErreur

Dim x As Integer

x = 5 / 0

Exit Sub

GestionErreur:

MsgBox "Erreur détectée : " & Err.Description

Resume Next

8.11 Conseils pratiques

- Toujours commenter son code pour faciliter la maintenance
- Utiliser des noms explicites pour les boutons, champs et variables
- Tester chaque macro ou procédure avant déploiement
- Commencer par de petites automatisations avant de créer des systèmes complexes

Références bibliographiques

Livres et documents

1. **Groh, D. (2019).** *Microsoft Access 2019 Step by Step*. Microsoft Press.
 - Couvre la création de tables, formulaires, requêtes, rapports et introduction à VBA.
2. **Hofmann, D. (2018).** *Access 2016 Bible*. Wiley.
 - Très complet sur les fonctionnalités avancées, macros et programmation VBA.
3. **Alexis, P. (2020).** *Développement d'applications Access*. Dunod.
 - Guide pratique sur les bases de données relationnelles, la conception de formulaires et rapports.
4. **Stephens, R. & Donaldson, M. (2017).** *Microsoft Access Programming by Example*. McGraw-Hill.
 - Approche par projets et exemples concrets pour apprendre le VBA et les macros.
5. **Gelinas, U. et al. (2019).** *Introduction aux bases de données relationnelles*. Pearson.
 - Théorie des SGBD, normalisation et relations entre tables.

Sites officiels et documentation Microsoft

6. **Microsoft Docs – Access**
 - <https://learn.microsoft.com/en-us/office/access/>
 - Documentation officielle sur Access, VBA, requêtes et rapports.
7. **Microsoft Support – Access**
 - <https://support.microsoft.com/access>
 - Tutoriels pas-à-pas, résolution de problèmes et guides pratiques.
8. **Office Training – Access Tutorials**
 - <https://support.microsoft.com/office/training>
 - Vidéos et tutoriels pour débutants et utilisateurs avancés.

Tutoriels et ressources en ligne

9. TechOnTheNet – Microsoft Access Tutorials

- <https://www.techonthenet.com/access/>
- Guides sur les tables, formulaires, requêtes, rapports et macros.

10.Database Journal – Access Tips & Tutorials

- <https://www.databasejournal.com/msaccess/>
- Astuces avancées pour la programmation VBA et la création de menus.

11.Guru99 – Microsoft Access Tutorial

- <https://www.guru99.com/free-microsoft-access-tutorial-course.html>
- Couvre Access depuis la création de tables jusqu'aux rapports et VBA.

12.Stack Overflow – Access VBA

- <https://stackoverflow.com/questions/tagged/ms-access-vba>
- Questions-réponses pour résoudre des problèmes concrets de programmation VBA.

Articles scientifiques et académiques

13.Date, C.J. (2012). *Database Design and Relational Theory: Normal Forms and All That Jazz*. O'Reilly Media.

- Théorie des bases de données relationnelles et normalisation, applicable aux chapitres sur la création de tables et relations.

14.Rob, P., & Coronel, C. (2018). *Database Systems: Design, Implementation, & Management*. Cengage Learning.

- Concepts généraux des SGBD, normalisation, requêtes et rapports.

15.Peterson, L. (2017). *VBA for Microsoft Access 2016: Programming Techniques and Examples*. Apress.

- Référence complète pour l'initiation à la programmation VBA dans Access.