Digitalisation de la Gestion du Centre Mindcraft

Bases de Données Relationnelles sous Microsoft Access

EL GHEMARY Farah

SIKSIK Zineb

Faculté des Sciences et Techniques
Département Mathématiques
Licence Mathématiques et Informatique Décisionnelles

Tanger, February 2026

Digitalisation de la Gestion du Centre Mindcraft

Bases de Données Relationnelles sous Microsoft Access

EL GHEMARY Farah

SIKSIK Zineb

Faculté des Sciences et Techniques
Département Mathématiques
Licence Mathématiques et Informatique Décisionnelles

Projet Fin Module

Tanger, February 2026

Contents

List of Figures iii
List of Tables 1

1 Introduction Générale **2**
1.1 Contexte du projet 2
1.2 Objectifs du projet 2
1.3 Méthodologie adoptée 3
1.4 Structure du rapport 3

2 Présentation du Cadre Professionnel **4**
2.1 Présentation générale 4
2.2 Mission et valeurs 4
2.3 Organisation interne 5
2.4 Problématique et opportunité 5

3 Analyse Fonctionnelle **6**
3.1 Périmètre fonctionnel 6
3.2 Acteurs du système 6
3.3 Cas d'utilisation 7

4 Conception et Modélisation **9**
4.1 Modèle Conceptuel de Données 9
4.2 Transformation du MCD → MLD → MPD 10
4.3 Modèle physique de données 11
4.4 Normalisation et justification 13

5 Implémentation sous Microsoft Access **14**
5.1 Création des tables et relations 14
5.2 Requêtes SQL 15
5.2.1 Exemples de requêtes importantes 16
5.3 Création des formulaires 17
5.4 Automatisation avec VBA 19
5.5 Création des rapports 21
5.5.1 Tableau de bord 21

6 Discussion et Perspectives **25**
6.1 Difficultés rencontrées 25
6.2 Points forts du système développé 26
6.3 Limites du système 26
6.4 Perspectives d'amélioration 26

6.5 Conclusion finale 27

Appendices

A Annexe A: Interfaces Access **30**
A.1 Les Tableaux Access 30
A.2 Les Formulaires Access 32

B Annexe B: Code SQL **35**
B.1 Requête de génération d'attestation de participation 35
B.2 Requête de résumé des paiements mensuels 35
B.3 Nombre d'élèves distincts par programme 36
B.4 Nombre de séances par programme 36
B.5 Synthèse complète par programme 37
B.6 Revenus totaux par programme 37
B.7 Suivi détaillé des paiements par élève 37
B.8 Emploi du temps hebdomadaire 38

C Annexe C: Code VBA **40**

List of Figures

4.1 MCD de projet 10

5.1 Relations entre les tableaux 15
5.2 Tableau Elève 15
5.3 Formulaire Élève 17
5.4 Formulaire Coach 18
5.5 Tableau Menu 18
5.6 Tableau de Board 22
5.7 Rapport des Inscriptions par programme 22
5.8 Rapport Attestation 23
5.9 Rapport paiement par élève 23
5.10 Rapport Paiement mensuels 24
5.11 Rapport Emploi du Temps 24

A.1 Tableau Coach 30
A.2 Tableau Inscription 30
A.3 Tableau Paiement 30
A.4 Tableau Programme 31
A.5 Tableau Salle 31
A.6 Tableau Séance 31
A.7 Tableau Attestation 31
A.8 Formulaire Inscription 32
A.9 Formulaire Programme 32
A.10 Formulaire Séance 33
A.11 Formulaire Inscription 33
A.12 Formulaire Paiement 34

List of Tables

4.1 Description des entités principales du système 9
4.2 Relations et cardinalités entre les entités principales 10
4.3 Description de Modèle logique des données 11
4.4 Correspondance entre types conceptuels et types physiques Access 13

5.1 Résumé des tables principales de la base de données 14

1
Introduction Générale

1.1 Contexte du projet

Dans un contexte où la digitalisation est devenue un levier essentiel d'efficacité, la gestion rigoureuse des données constitue un facteur clé pour toute organisation éducative ou professionnelle. Le centre de formation Mindcraft, spécialisé dans les domaines de la robotique, du programmation et des sciences appliquées, gère quotidiennement un grand nombre d'informations : élèves, salles, séances, programmes, paiements, formateurs et plannings.

Jusqu'à récemment, ces opérations étaient assurées à l'aide de documents Excel ou de registres manuels, ce qui entraînait des problèmes de redondance, d'erreurs de saisie, de lenteur et de manque de cohérence entre les informations. Face à ces limites, il est devenu nécessaire de mettre en place un système intégré permettant d'assurer la centralisation, la fiabilité et la traçabilité des données tout en offrant une interface simple et adaptée au personnel administratif.

1.2 Objectifs du projet

L'objectif général de ce projet est la conception et l'implémentation d'une base de données relationnelle permettant d'automatiser la gestion du centre de formation Mindcraft. Plus précisément, il s'agit de :

- Créer une base de données normalisée assurant l'intégrité et la cohérence des informations ;
- Concevoir une interface utilisateur intuitive pour la gestion des élèves, des séances et des paiements ;
- Intégrer des formulaires, requêtes et rapports dynamiques ;
- Automatiser certaines tâches (calculs, validations, impression, export PDF) à l'aide du langage VBA ;
- Mettre à disposition un outil fiable, évolutif et facilement exploitable par le personnel.

1.3 Méthodologie adoptée

Le développement de l'application a suivi une démarche structurée inspirée des étapes classiques de conception d'un système d'information :

1. **Analyse fonctionnelle :**identification des besoins du centre, des acteurs et des flux de données ;
2. **Modélisation :** élaboration du MCD puis MLD et MPD ;
3. **Implémentation :** création des tables, relations et contraintes d'intégrité dans Microsoft Access ;
4. **Développement d'interfaces :** conception de formulaires interactifs et rapports de synthèse ;
5. **Automatisation et tests :** ajout de code VBA, validation des entrées, génération de rapports et vérification du bon fonctionnement global.

Cette approche a permis d'obtenir une application complète, robuste et conforme aux principes du modèle relationnel tout en répondant aux besoins concrets du centre Mindcraft.

1.4 Structure du rapport

Le présent rapport s'articule autour de huit chapitres :

- Le chapitre 1 présente le contexte, les objectifs et la démarche suivie ;
- Le chapitre 2 introduit le cadre professionnel et l'organisation du centre ;
- Le chapitre 3 décrit l'analyse fonctionnelle et les besoins du système ;
- Le chapitre 4 détaille la modélisation conceptuelle et logique des données ;
- Le chapitre 5 explique l'implémentation sous Microsoft Access ;
- Le chapitre 6 présente les tests et les résultats obtenus ;
- Le chapitre 7 expose les difficultés rencontrées et les pistes d'amélioration ;
- Enfin, le chapitre 8 conclut le travail et évoque les perspectives futures du projet.

2 Présentation du Cadre Professionnel

2.1 Présentation générale

Mindcraft est un centre de STEM basé à Tanger, spécialisé dans l'enseignement de la robotique et du programmation pour les jeunes. Créé dans une optique d'éducation moderne et pratique, le centre propose des ateliers interactifs et des programmes pédagogiques orientés vers les STEM (Science, Technology, Engineering, Mathematics). Il s'adresse à un public varié: enfants, adolescents et étudiants, désireux de développer des compétences techniques et créatives adaptées aux besoins du monde numérique.

Au-delà de la simple transmission de connaissances, Mindcraft vise à initier les apprenants à la pensée logique, à la résolution de problèmes et à l'innovation, tout en leur offrant un environnement collaboratif et motivant. Grâce à ses partenariats avec des écoles, associations et organisations locales, le centre contribue activement à la promotion de la culture scientifique et technologique dans la région de Tanger-Tétouan-Al Hoceima.

2.2 Mission et valeurs

La mission principale de Mindcraft est de rendre les sciences et la technologie accessibles à tous, à travers une approche basée sur la pratique, l'expérimentation et la créativité. Ses valeurs fondamentales reposent sur :

- **L'apprentissage actif :** apprendre en manipulant et en expérimentant ;
- **L'innovation pédagogique :** utiliser la robotique, la programmation et la conception numérique comme outils éducatifs ;
- **L'inclusion et la curiosité :** encourager la participation de tous, quel que soit le niveau de départ ;
- **L'excellence et la rigueur :** former des jeunes capables de raisonner, d'analyser et de créer.

Ces principes guident l'ensemble des programmes et des décisions stratégiques du centre.

2.3 Organisation interne

L'organisation de Mindcraft repose sur une structure simple et fonctionnelle, favorisant la collaboration entre les différents pôles d'activité :

- **Direction du centre :** supervision générale, coordination des formations et partenariats ;
- **Pôle pédagogique :** encadrement, conception des programmes éducatifs, planification des séances ;
- **Pôle administratif :** gestion des inscriptions, suivi des paiements, communication avec les parents;

Cette répartition permet d'assurer un fonctionnement fluide et un suivi personnalisé des élèves.

2.4 Problématique et opportunité

Avant la mise en place du système Access, la gestion quotidienne du centre présentait plusieurs difficultés :

- Absence d'une base de données centralisée pour regrouper les informations des élèves, des séances et des paiements;
- Risques d'erreurs dans les montants payés ou les dates d'échéance;
- Difficulté à retrouver rapidement les informations d'un élève ou d'un programme;
- Manque de visibilité sur les paiements en attente ou les séances planifiées.

Ces constats ont mis en évidence le besoin urgent d'une solution logicielle fiable, capable d'automatiser les tâches répétitives, de structurer les données et d'offrir un accès rapide et sécurisé aux informations.

3 Analyse Fonctionnelle

L'analyse fonctionnelle a pour but d'identifier et de décrire les besoins du centre Mindcraft afin de concevoir une application cohérente, simple d'utilisation et conforme à la réalité du terrain. Elle vise à traduire les attentes des utilisateurs en fonctions concrètes que le système devra offrir, tout en respectant les contraintes techniques et organisationnelles du centre.

3.1 Périmètre fonctionnel

Le système développé couvre les besoins essentiels de gestion au sein d'un centre de formation. Il permet notamment :

- La gestion des élèves (création, mise à jour et consultation).
- La gestion des coachs et de leurs informations associées.
- La gestion des programmes pédagogiques proposés.
- La gestion des séances (planning, coach assigné, capacité...).
- La gestion des inscriptions aux programmes / séances.
- L'enregistrement et le suivi des paiements.
- La génération de rapports et d'états imprimables.
- La production d'attestations de participation.
- L'accès à un menu principal facilitant la navigation entre les différentes composantes.

3.2 Acteurs du système

Le système comporte deux acteurs principaux qui interagissent avec l'application :

Administrateur

C'est l'utilisateur principal de l'application. Il peut :

- gérer les élèves, coachs, programmes et séances ;
- inscrire des élèves dans des séances ;
- enregistrer des paiements et éditer des reçus ;
- générer des rapports variés (paiements, participations, tableau de bord) ;
- mettre à jour les données ou corriger les informations existantes.

Coach

Dans cette version, le coach dispose d'un rôle informatif. Il peut :

- consulter les séances auxquelles il est assigné ;
- vérifier la liste des élèves inscrits à ses séances ;
- consulter le programme associé à sa mission.

Chaque fonctionnalité est conçue pour répondre à un besoin réel observé dans la gestion quotidienne du centre.

- **Réduire la charge administrative** en automatisant la saisie et la mise à jour des informations ;
- **Assurer la cohérence des données** grâce à l'intégrité référentielle et à la normalisation des tables ;
- **Améliorer la visibilité** des activités du centre à travers des rapports synthétiques et précis ;
- **Simplifier l'accès à l'information** via une interface claire et une navigation intuitive.

3.3 Cas d'utilisation

Les cas d'utilisation représentent les interactions principales entre les acteurs et le système. Les fonctionnalités essentielles se déclinent comme suit :

Use Case 1: Gérer Élève

- Ajouter un nouvel élève.
- Modifier les informations d'un élève existant.
- Rechercher et consulter une fiche élève.

Use Case 2: Gérer Coach

- Ajouter un coach.
- Modifier ses données.
- Afficher les séances animées.

Use Case 3: Gérer Programme

- Définir les programmes proposés.
- Associer des séances à un programme.

Use Case 4: Planifier une Séance

- Créer une séance (date, horaire, coach, programme).
- Définir la capacité maximale.

Use Case 5: Inscrire un Élève

- Sélectionner un élève.
- Sélectionner une séance.
- Vérifier la capacité.
- Enregistrer l'inscription.

Use Case 6: Enregistrer Paiement

- Choisir un élève.
- Saisir un montant et une période.
- Générer un reçu ou un relevé.

Use Case 7: Générer Rapports

- Rapport de participation par programme.
- Attestation individuelle.
- Rapport de paiements mensuels.
- Tableau de bord synthétique.

Ces cas d'utilisation constituent le cœur fonctionnel de l'application.

4
Conception et Modélisation

À partir de l'analyse fonctionnelle, la phase de conception a pour objectif de traduire les besoins exprimés en une représentation structurée des données. Cette étape s'appuie sur la méthode de MCD, qui permet de représenter les objets du système, leurs caractéristiques et leurs relations logiques, avant de passer à l'implémentation dans Microsoft Access.

4.1 Modèle Conceptuel de Données

Le MCD représente les principales entités du système ainsi que les associations qui traduisent leurs liens fonctionnels. Dans le cadre du centre de formation Mindcraft, les entités identifiées sont les suivantes :

Table 4.1: *Description des entités principales du système*

Entité	Description	Attributs principaux
Élève	Représente chaque apprenant inscrit au centre.	ID_Élève, Nom, Prénom, DateNaissance, NomParent, EmailParent, Téléphone, École
Programme	Regroupe les informations sur les formations proposées.	ID_Programme, NomProgramme, Description, Catégorie
Séance	Décrit chaque séance programmée.	ID_Séance, JourSemaine, Heure, DateDébut, DateFin, Capacité
Coach	Représente les formateurs encadrant les séances.	ID_Coach, Nom, Prénom, Spécialité, TypeRémunération, MontantRémunération, Email, Téléphone
Paiement	Trace les transactions effectuées par les élèves.	ID_Paiement, Montant, DatePaiement, ModePaiement
Salle	Désigne les salles où se déroulent les séances.	ID_Salle, NomSalle, CapacitéSalle
Attestation	Représente les certificats délivrés aux élèves.	ID_Attestation, DateAttestation, Statut, NumeroAttestation

Associations principales:

- **Inscription** : relie un élève à une séance (1:N). Un élève peut suivre plusieurs programmes ; un programme regroupe plusieurs élèves.

- **Planification** : chaque séance appartient à un seul programme (1:N entre Programme et Séance).
- **Encadrement** : un coach anime plusieurs séances (1:N entre Coach et Séance).
- **Réservation de salle** : une salle peut accueillir plusieurs séances, mais chaque séance se déroule dans une seule salle (1:N entre Salle et Séance).
- **Règlement** : chaque paiement est lié à un seul élève (1:N entre Élève et Paiement).
- **Attestation** : un élève peut recevoir plusieurs attestations, chacune liée à un programme validé (1:N entre Élève et Attestation).

Cardinalités:

Table 4.2: *Relations et cardinalités entre les entités principales*

Relation	Cardinalité (min, max)
Élève – Programme (Inscription)	Élève (1,1) ↔ (0,N) Inscription Programme (1,1) ↔ (0,N) Inscription
Programme – Séance (Planification)	Programme (1,1) ↔ (0,N) Séance
Coach – Séance (Encadrement)	Coach (1,1) ↔ (0,N) Séance
Élève – Paiement (Règlement)	Élève (1,1) ↔ (0,N) Paiement
Salle – Séance (Occupation / Réservation)	Salle (1,1) ↔ (0,N) Séance
Élève – Attestation	Élève (1,1) ↔ (0,N) Attestation
Programme – Attestation	Programme (1,1) ↔ (0,N) Attestation

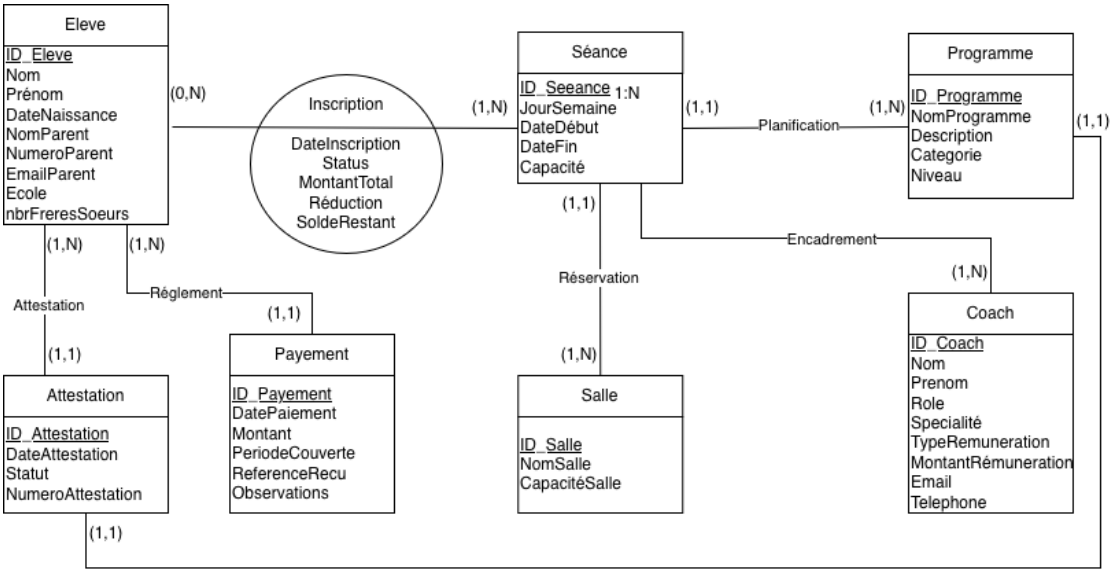

Figure 4.1: *MCD de projet*

4.2 Transformation du MCD → MLD → MPD

La transformation du MCD en modèle logique (relationnel) consiste à convertir chaque entité et chaque relation en table relationnelle tout en conservant les dépendances fonctionnelles et les contraintes d'intégrité.

- Chaque entité devient une table.

- Les identifiants deviennent les clés primaires (PK).
- Pour une relation 1:N, la clé primaire du côté « 1 » devient une clé étrangère (FK) dans la table du côté « N ».
- Pour une relation N:N, une nouvelle table associative est créée, contenant les deux clés primaires et les éventuels attributs de la relation.

Table 4.3: *Description de Modèle logique des données*

Entité	Description	Attributs principaux
Élève	Représente chaque apprenant inscrit au centre.	ID_Élève (PK), Nom, Prénom, DateNaissance, NomParent, EmailParent, Téléphone, École
Programme	Regroupe les informations sur les formations proposées.	ID_Programme (PK), NomProgramme, Description, Catégorie
Séance	Décrit chaque séance programmée.	ID_Séance (PK), JourSemaine, Heure, DateDébut, DateFin, Capacité, ID_Programme (FK), ID_Coach (FK), ID_Salle (FK)
Coach	Représente les formateurs encadrant les séances.	ID_Coach (PK), Nom, Prénom, Spécialité, TypeRémunération, MontantRémunération, Email, Téléphone
Paiement	Trace les transactions effectuées par les élèves.	ID_Paiement (PK), ID_Élève (FK), Montant, DatePaiement, ModePaiement
Salle	Désigne les salles où se déroulent les séances.	ID_Salle (PK), NomSalle, CapacitéSalle
Attestation	Représente les certificats délivrés aux élèves.	ID_Attestation (PK), ID_Élève (FK), ID_Programme (FK), DateAttestation, Statut, NuméroAttestation

4.3 Modèle physique de données

Le MPD correspond à la traduction concrète du modèle logique en tables Access, avec les types de données, les contraintes et les relations matérialisées.

```
ÉLÈVE(
    ID_Élève AUTONUM PK,
    Nom TEXT(50),
    Prénom TEXT(50),
    DateNaissance DATE,
    NomParent TEXT(50),
    EmailParent TEXT(100),
    Téléphone TEXT(20),
    École TEXT(50)
)

PROGRAMME(
    ID_Programme AUTONUM PK,
    NomProgramme TEXT(100),
    Description TEXT,
    Catégorie TEXT(50),
    Durée INTEGER
)
```

```
COACH(
    ID_Coach AUTONUM PK,
    Nom TEXT(50),
    Prénom TEXT(50),
    Spécialité TEXT(100),
    TypeRémunération TEXT(20),
    MontantRémunération CURRENCY,
    Email TEXT(100),
    Téléphone TEXT(20)
)

SALLE(
    ID_Salle AUTONUM PK,
    NomSalle TEXT(50),
    CapacitéSalle INTEGER,
    Localisation TEXT(50)
)

SÉANCE(
    ID_Séance AUTONUM PK,
    JourSemaine TEXT(20),
    Heure TEXT(10),
    DateDébut DATE,
    DateFin DATE,
    Capacité INTEGER,
    ID_Programme FK,
    ID_Coach FK,
    ID_Salle FK
)

INSCRIPTION(
    ID_Inscription AUTONUM PK,
    DateInscription DATE,
    Statut TEXT(20),
    ID_Élève FK,
    ID_Programme FK
)

PAIEMENT(
    ID_Paiement AUTONUM PK,
    Montant CURRENCY,
    DatePaiement DATE,
    ModePaiement TEXT(20),
    ID_Élève FK
)

ATTESTATION(
    ID_Attestation AUTONUM PK,
    DateAttestation DATE,
```

```
    Statut TEXT(20),
    NuméroAttestation TEXT(20),
    ID_Élève FK,
    ID_Programme FK
)
```

Types de données choisis

Table 4.4: *Correspondance entre types conceptuels et types physiques Access*

Type conceptuel	Type Access (MPD)
Identifiant unique	AUTONUM (AutoNumber)
Texte court	TEXT(n)
Texte long	LONG TEXT
Numérique entier	INTEGER / LONG
Date	DATE/TIME
Valeur monétaire	CURRENCY
Booléen / Statut	YES/NO ou TEXT(20)

Contraintes d'intégrité

- **Clés primaires (PK)** : chaque table possède une clé primaire de type AUTONUM garantissant l'unicité des enregistrements.
- **Clés étrangères (FK)** : les relations Élève–Inscription, Programme–Séance, Coach–Séance, Salle–Séance, Élève–Paiement et Élève/Programme–Attestation sont matérialisées par des clés étrangères assurant la cohérence des liens.
- **Intégrité référentielle** : activée dans toutes les relations, avec *mise à jour en cascade* pour propager les modifications d'identifiants et *interdiction de suppression* lorsqu'un enregistrement enfant existe (pour éviter les données orphelines).
- **Contraintes de domaine** : types de données adaptés (dates, montants, longueurs maximales), champs obligatoires pour les attributs critiques.

4.4 Normalisation et justification

L'ensemble des tables du MPD respecte la troisième forme normale (3FN), ce qui garantit l'absence de redondance et améliore la cohérence et la maintenabilité du système.

- **1FN Atomicité** : tous les attributs sont élémentaires (pas de champs multivalués ou composés).
- **2FN Dépendances partielles** : aucune table ne possède de clé primaire composite ; tous les attributs dépendent entièrement de la clé.
- **3FN Dépendances transitives** : aucun attribut non-clé ne dépend d'un autre attribut non-clé (par ex. un paiement ne contient pas le nom de l'élève).

Cette normalisation assure une réduction significative des anomalies de mise à jour, de suppression et d'insertion, tout en facilitant l'évolution future du système.

5 Implémentation sous Microsoft Access

Après la finalisation du modèle physique de données, l'étape suivante consiste à implémenter l'application au sein de Microsoft Access. Cette phase vise à transformer le schéma conceptuel et logique en un système complètement fonctionnel.

5.1 Création des tables et relations

Toutes les tables du MPD ont été créées en mode « Conception » dans Access.

Table 5.1: *Résumé des tables principales de la base de données*

Table	Champs essentiels	Clés
tblÉlève	ID_Élève, Nom, Prénom, DateNaissance, NomParent, EmailParent, Téléphone, École	ID_Élève (PK)
tblProgramme	ID_Programme, NomProgramme, Description, Catégorie, Durée	ID_Programme (PK)
tblCoach	ID_Coach, Nom, Prénom, Spécialité, Rémunération, Email	ID_Coach (PK)
tblSalle	ID_Salle, NomSalle, CapacitéSalle, Localisation	ID_Salle (PK)
tblSéance	ID_Séance, JourSemaine, Heure, DateDébut, DateFin, Capacité, ID_Programme, ID_Coach, ID_Salle	ID_Séance (PK), ID_Programme, ID_Coach, ID_Salle (FK)
tblInscription	ID_Inscription, DateInscription, Statut, ID_Élève, ID_Programme	ID_Inscription (PK), FKs (Élève, Programme)
tblPaiement	ID_Paiement, Montant, DatePaiement, ModePaiement, ID_Élève	ID_Paiement (PK), ID_Élève (FK)
tblAttestation	ID_Attestation, DateAttestation, Statut, ID_Élève, ID_Programme	ID_Attestation (PK), FKs (Élève, Programme)

Chaque table a reçu :

- sa clé primaire (type : Autonumération),

- les types de données corrects,
- les contraintes de validation nécessaires
- det les clés étrangères définissant les dépendances entre entités.

Schéma des relations Access

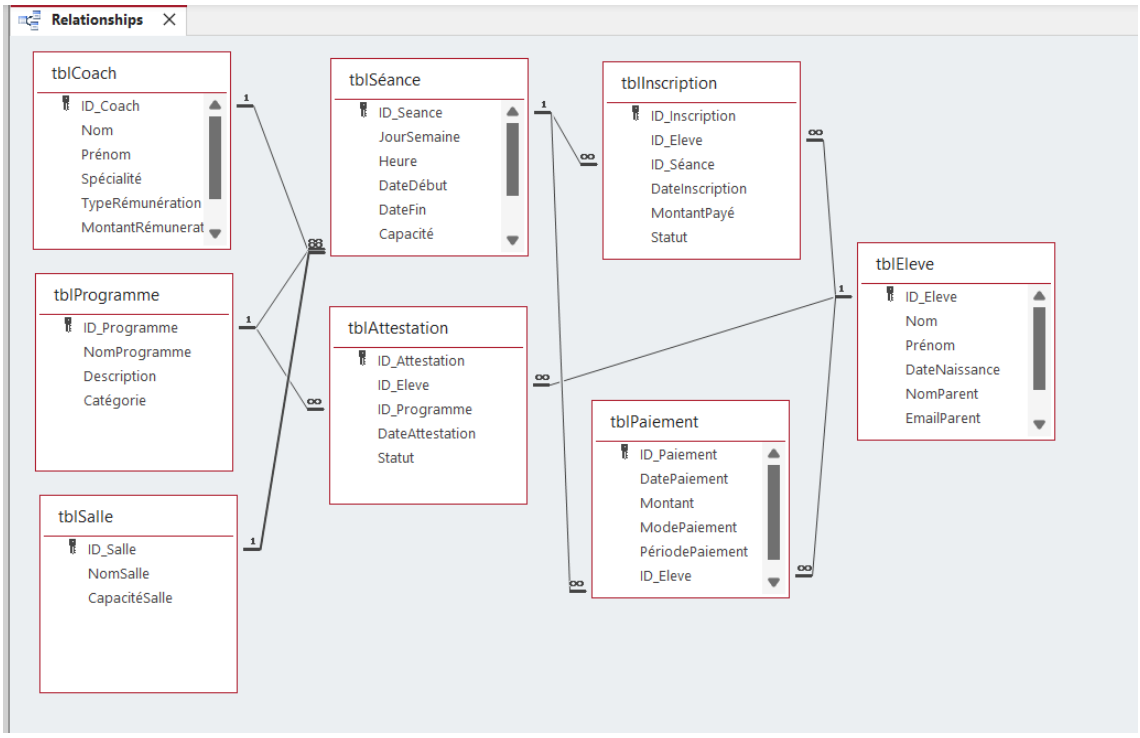

Figure 5.1: *Relations entre les tableaux*

tblEleve

ID_Eleve	Nom	Prénom	DateNaissai	NomParent	EmailParent	Telephone	Ecole	Click to Add
3	El Idrissi	Adam	4/21/2012	Salma Idrissi	salma.idrissi@	(+212) 6 12 34	Lycée Descarte	
9	Bensalem	Lina	6/10/2013	Youssef Bensa	youssef.b@exa	(+212) 6 06 67	Groupe Scolair	
10	Chraibi	Rayan	2/28/2011	Nada Chraibi	nada.chraibi@	(+212) 6 06 78	Ecole Ibn Batto	
11	Rahmani	Inès	12/3/2010	Fatima Rahma	fatima.rahman	(+212) 6 06 70	Lycée Molière	
12	Aarab	Mehdi	9/18/2012	Hicham Aarab	h.aarab@exam	(+212) 6 06 83	Groupe Scolair	
13	El Ghomari	Kenza	11/8/2014	Samira El Ghor	samira.elg@ex	(+212) 6 06 72	Groupe Scolair	
14	Amrani	Ilyas	7/25/2011	Fatiha Amrani	f.amrani@exar	(+212) 6 06 65	Lycée Regnault	
15	Oukili	Sara	3/14/2013	Hajar Oukili	hajar.oukili@e	(+212) 6 06 10	Ecole Sadiki	
16	Talbi	Yassir	9/9/2010	Amal Talbi	amal.talbi@ex	(+212) 6 06 29	Lycée Al Wahd	
17	Bennani	Hiba	10/22/2011	Rachid Bennar	r.bennani@exa	(+212) 6 06 70	Lycée Descarte	
(New)								

Figure 5.2: *Tableau Elève*

Les autres tables (*Programme*, *Coach*, *Séance*, etc.) sont présentées en annexe A.

5.2 Requêtes SQL

Les requêtes ont été construites pour :

- filtrer et formater les données affichées dans les formulaires (notamment les zones de liste déroulante),
- produire des statistiques à partir des données de paiement et de présence,

- alimenter les rapports de synthèse et d'attestation.

5.2.1 Exemples de requêtes importantes

Requêtes utilisées pour les zones de liste déroulante (ComboBox)

Afin d'afficher les *noms complets* au lieu des identifiants numériques dans les formulaires (`Élève`, `Coach`, `Programme`, `Séance`), plusieurs requêtes ont été créées. Ces requêtes, telles que `qryEleveListe`, `qryCoachListe`, `qryProgrammeListe` et `qrySeanceListe`, permettent de rendre les interfaces plus intuitives pour l'utilisateur.

Exemple : Liste des élèves (qryEleveListe)

```
SELECT
    [tblEleve].[ID_Eleve],
    [tblEleve].[Nom] & " " & [tblEleve].[Prénom] AS NomComplet,
    [tblEleve].[Ecole],
    [tblEleve].[Telephone]
FROM
    tblEleve
ORDER BY
    [tblEleve].[Nom],
    [tblEleve].[Prénom];
```

Ces requêtes de liste sont également reprises en Annexe B pour l'ensemble des tables (*Coach*, *Programme*, *Séance*).

Requête de tableau de bord (vue globale)

Cette requête récupère des statistiques globales du centre : nombre total d'élèves, de programmes, de séances, de coachs, et le revenu total cumulé.

```
SELECT
    Max(
        (
            SELECT Count([ID_Eleve]) FROM [tblEleve]
        )
    ) AS TotalÉlèves,
    Max(
        (
            SELECT Count([ID_Programme]) FROM [tblProgramme]
        )
    ) AS TotalProgrammes,
    Max(
        (
            SELECT Count([ID_Seance]) FROM [tblSéance]
        )
    ) AS TotalSéances,
    Max(
        (
            SELECT Count([ID_Coach]) FROM [tblCoach]
        )
```

```
    ) AS TotalCoachs,
    Max(
        (
            SELECT Sum([Montant]) FROM [tblPaiement]
        )
    ) AS RevenuTotal
FROM
    tblEleve;
```

Elle est utilisée pour alimenter le rapport `rpt_TableauDeBordMindcraft`.

Toutes les autres requêtes intermédiaires (de filtrage, de recherche, ou d'alimentation des formulaires) sont détaillées en Annexe B

5.3 Création des formulaires

Les formulaires constituent l'interface principale de l'utilisateur. Ils permettent la saisie, la modification et la consultation des données sans avoir à manipuler directement les tables.

Formulaire Élèves

Figure 5.3: *Formulaire Élève*

Ce formulaire permet la gestion complète des fiches des élèves. L'utilisateur peut y ajouter, modifier ou supprimer un enregistrement. Les champs principaux comprennent le nom, le prénom, la date de naissance, l'école et le contact du parent. Des boutons de navigation facilitent le passage entre les fiches, et un message d'alerte s'affiche en cas de champ obligatoire non renseigné.

Formulaire Coach

Ce formulaire regroupe toutes les informations relatives aux coachs : identité, spécialité, type de rémunération et coordonnées. Le formulaire affiche également, sous forme de tableau, la liste des séances associées à chaque coach. Il facilite la mise à jour du personnel encadrant et permet d'assigner ensuite les coachs aux séances correspondantes.

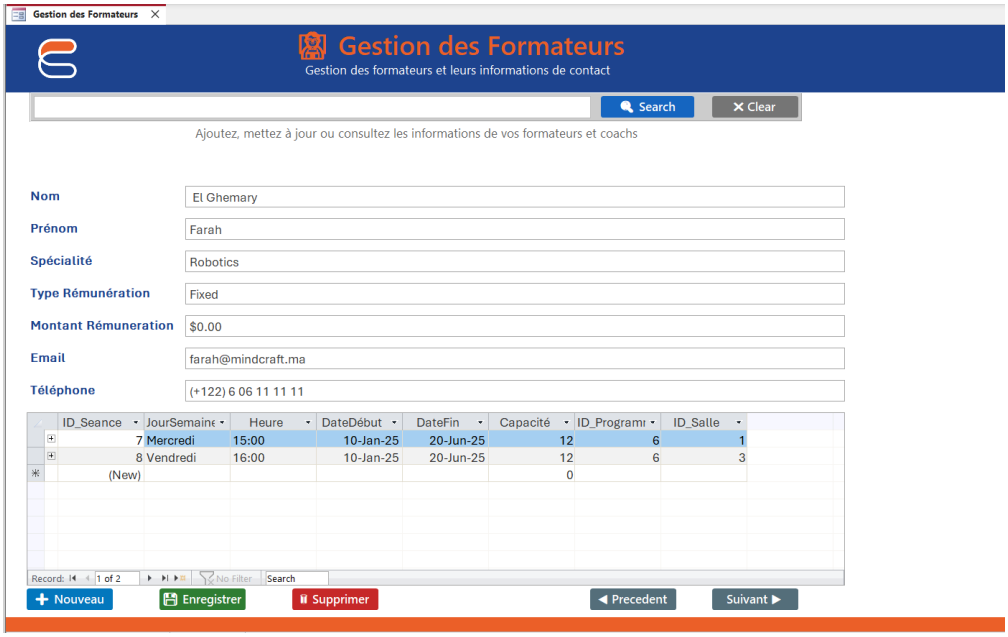

Figure 5.4: *Formulaire Coach*

Menu principal

Figure 5.5: *Tableau Menu*

Le menu principal constitue la page d'accueil de l'application Access. Il regroupe, sous forme de boutons les accès directs aux différents formulaires, requêtes et rapports. Il organise la navigation et améliore considérablement l'ergonomie de l'application.
Les autres formulaires (*Programme*, *Coach*, *Séance*, etc.) sont présentées en annexe A.

Cohérence visuelle

Tous les formulaires partagent une même charte graphique : couleurs harmonisées, polices uniformes et disposition cohérente des boutons. Cette uniformité renforce l'identité visuelle de l'application et facilite la prise en main. Chaque formulaire a été testé afin d'assurer la stabilité, la rapidité et la simplicité d'utilisation de l'ensemble du système.

5.4 Automatisation avec VBA

L'intégration du langage **VBA** (**Visual Basic for Applications**) a joué un rôle essentiel dans l'automatisation de nombreuses opérations de l'application Access et dans l'amélioration de l'expérience utilisateur. Chaque formulaire de gestion (Élève, Coach, Programme, Séance, Inscription, Paiement) inclut des procédures VBA permettant la recherche dynamique, la validation des entrées, la navigation fluide et la sécurisation des actions critiques.

Les extraits suivants illustrent les mécanismes VBA clés implémentés dans l'application.

Recherche dynamique dans les formulaires

Chaque formulaire dispose d'une zone de recherche permettant de filtrer immédiatement les enregistrements en fonction du nom ou du prénom saisi par l'utilisateur.

```
Private Sub cmdSearch_Click()
    Dim q As String
    q = Nz(Me.txtSearchStudent, "")

    If Len(q) = 0 Then
        Me.FilterOn = False
        Exit Sub
    End If

    q = Replace(q, "'", "''")
    Me.Filter = "[Nom] LIKE '*" & q & "*' OR [Prénom] LIKE '*" & q & "*'"
    Me.FilterOn = True
End Sub

Private Sub cmdClear_Click()
    Me.txtSearchStudent = Null
    Me.FilterOn = False
End Sub
```

Gestion des enregistrements (ajout, sauvegarde, suppression)

Les formulaires disposent de commandes permettant :

- d'ajouter un nouvel enregistrement,

- de sauvegarder les modifications,
- de supprimer un enregistrement après confirmation.

```
Private Sub cmdNew_Click()
    DoCmd.GoToRecord , , acNewRec
    If Not Me.NewRecord Then
        DoCmd.GoToRecord , , acNewRec
    End If
End Sub

Private Sub cmdSave_Click()
    DoCmd.RunCommand acCmdSaveRecord
End Sub

Private Sub cmdDelete_Click()
    If Me.NewRecord Then Exit Sub
    If MsgBox("Delete this student?", vbYesNo + vbQuestion, "Confirm") = vbYes Then
        DoCmd.RunCommand acCmdDeleteRecord
    End If
End Sub
```

Validation automatique avant enregistrement

Avant de sauvegarder un élève ou une modification, plusieurs vérifications sont automatiquement appliquées pour garantir la cohérence des données :

- champs obligatoires (Nom, Prénom),
- date de naissance non future,
- adresse e-mail valide.

```
Private Sub Form_BeforeUpdate(Cancel As Integer)

    If Nz(Me.Nom, "") = "" Then
        MsgBox "Please enter the student's last name.", vbExclamation
        Me.Nom.SetFocus
        Cancel = True
        Exit Sub
    End If

    If Nz(Me.Prénom, "") = "" Then
        MsgBox "Please enter the student's first name.", vbExclamation
        Me.Prénom.SetFocus
        Cancel = True
        Exit Sub
    End If

    If Not IsNull(Me.DateNaissance) Then
        If Me.DateNaissance > Date Then
            MsgBox "Birth date cannot be in the future.", vbExclamation
            Me.DateNaissance.SetFocus
            Cancel = True
```

```
            Exit Sub
        End If
    End If

    If Nz(Me.EmailParent, "") <> "" Then
        Dim e As String
        e = Me.EmailParent
        If InStr(e, "@") = 0 Or InStrRev(e, ".") < InStr(e, "@") + 2 Then
            MsgBox "Please enter a valid parent email.", vbExclamation
            Me.EmailParent.SetFocus
            Cancel = True
        End If
    End If

End Sub
```

Navigation entre formulaires

Depuis le menu principal, chaque bouton ouvre un formulaire spécifique pour faciliter la navigation.

```
Private Sub cmdEleves_Click()
    DoCmd.OpenForm "frmEleves"
End Sub

Private Sub cmdPaiements_Click()
    DoCmd.OpenForm "frmPaiements"
End Sub
```

Toutes les autres procédures VBA spécifiques aux formulaires (`frmCoach`, `frmProgramme`, `frmSéance`, `frmInscription`, `frmPaiement`) sont listées intégralement en annexe C.

5.5 Création des rapports

Les rapports ont été conçus afin de fournir à la direction du centre une vue d'ensemble claire, synthétique et professionnelle des données enregistrées dans la base. Ils permettent non seulement de visualiser les informations essentielles, mais également de les regrouper, trier et analyser selon différents critères pertinents. Chaque rapport peut être imprimé directement ou exporté automatiquement au format PDF grâce aux procédures VBA intégrées.

5.5.1 Tableau de bord

Ce rapport présente une vue globale du fonctionnement du centre : nombre total d'élèves, nombre de programmes actifs, nombre de séances planifiées, effectif des coachs et revenu total généré. Ces indicateurs sont calculés à l'aide de sous-requêtes SQL et affichés sous forme de cartes synthétiques pour une consultation rapide par la direction.

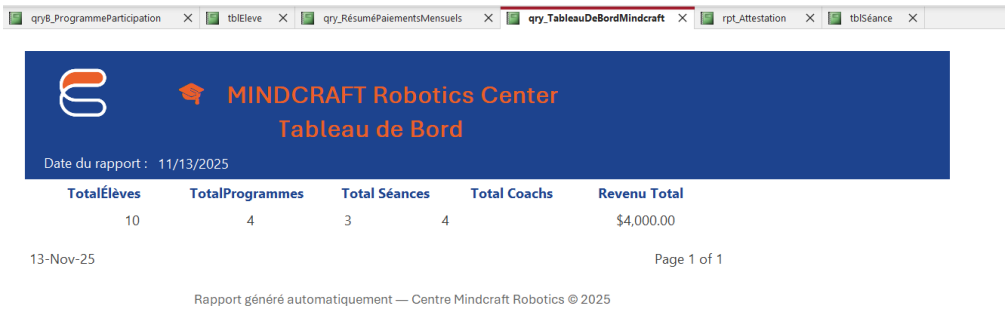

Figure 5.6: *Tableau de Board*

Participation par programme

Ce rapport regroupe, pour chaque programme, plusieurs informations essentielles : le nombre de séances planifiées, le nombre d'élèves distincts inscrits et le revenu total associé. Il permet d'évaluer la performance de chaque formation, d'optimiser la planification et d'orienter les décisions pédagogiques et financières du centre.

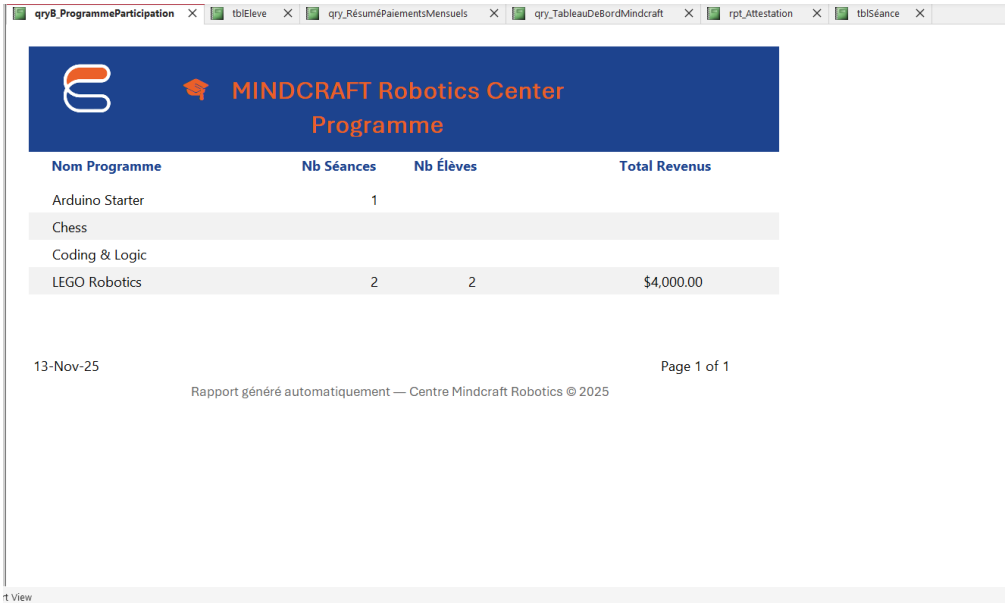

Figure 5.7: *Rapport des Inscriptions par programme*

Attestation de participation

Ce rapport génère automatiquement une attestation personnalisée pour chaque élève. Les données affichées (nom complet, programme suivi, période de participation et nom du coach) proviennent d'une requête agrégée dédiée. Le certificat peut être imprimé individuellement depuis le formulaire Élève ou en lot depuis le menu principal, et peut également être exporté au format PDF.

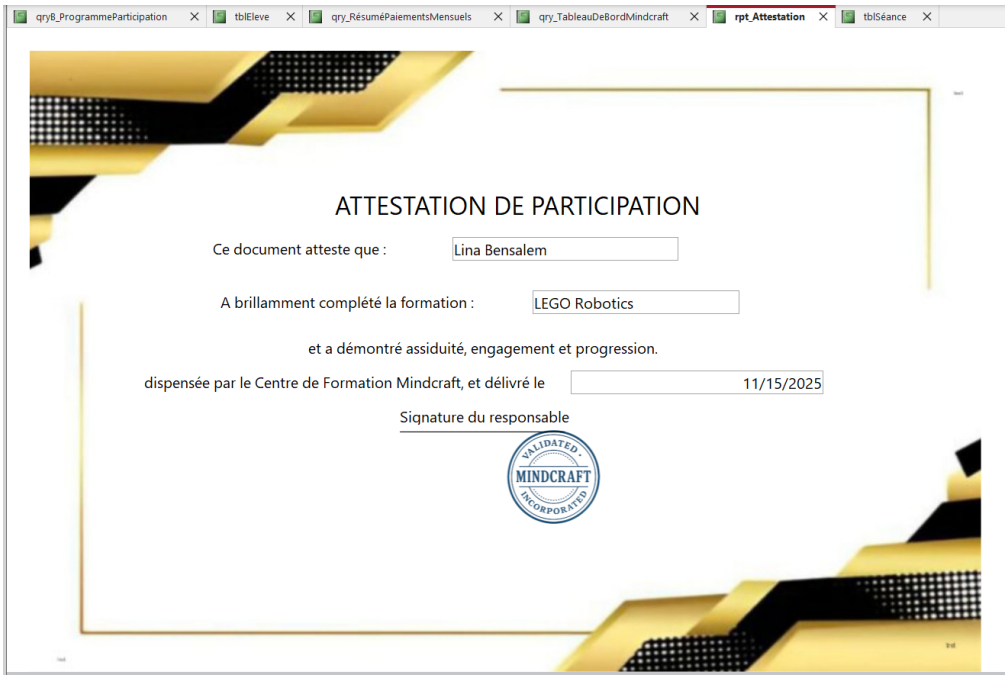

Figure 5.8: *Rapport Attestation*

Relevé des paiements par élève

Ce rapport présente l'historique détaillé des paiements effectués par chaque élève : montants versés, période payée, date de règlement, programme concerné et séance associée. Les données sont triées chronologiquement, et un total par élève est calculé automatiquement. Ce rapport constitue l'outil principal pour le suivi financier individuel et la détection des paiements manquants ou incomplets.

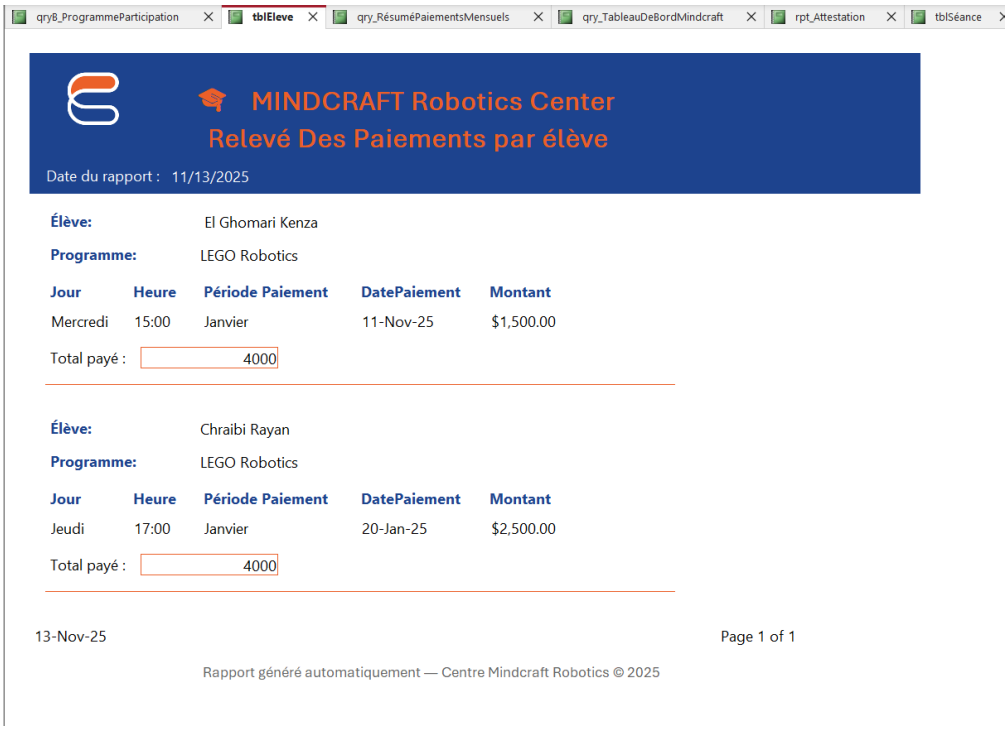

Figure 5.9: *Rapport paiement par élève*

Résumé des paiements mensuels

Ce rapport regroupe tous les paiements du centre par mois et calcule : le nombre total de paiements, le montant cumulé du mois et le montant moyen par élève. Il fournit une vue claire de l'évolution des revenus mensuels du centre et facilite le suivi comptable, la planification budgétaire et la prise de décision stratégique.

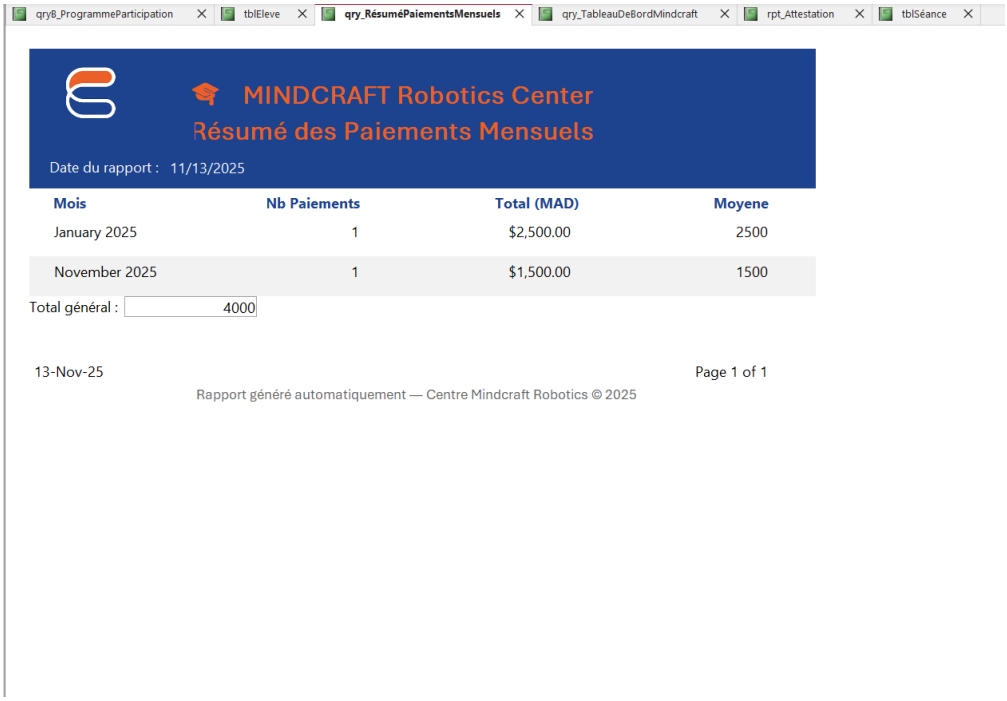

Figure 5.10: *Rapport Paiement mensuels*

Emploi du temps hebdomadaire

Ce rapport présente l'ensemble des séances planifiées au cours de la semaine. Les données affichées incluent le jour, l'heure, le programme, le coach responsable et la salle assignée lorsqu'elle est renseignée. Les séances sont triées chronologiquement (du lundi au samedi) puis par horaire, ce qui permet une lecture fluide et intuitive du planning.

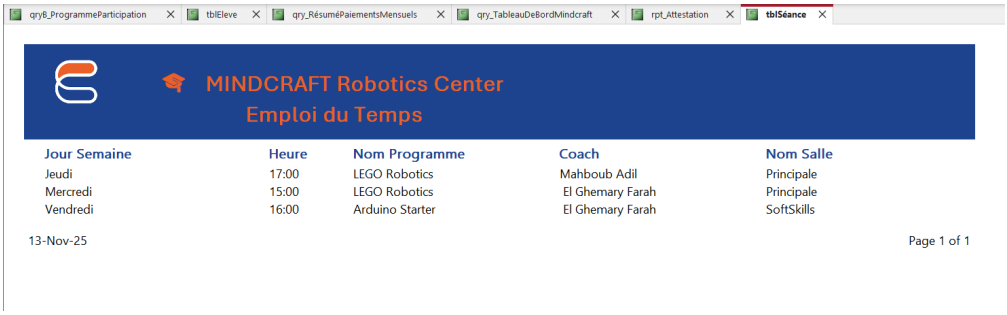

Figure 5.11: *Rapport Emploi du Temps*

6
Discussion et Perspectives

Le développement de l'application de gestion du centre de formation **Mindcraft** a permis de concrétiser les compétences acquises en bases de données relationnelles et en programmation VBA. Cette section présente une réflexion critique sur le déroulement du projet, les difficultés rencontrées, les solutions adoptées ainsi que les pistes d'amélioration envisagées pour les versions futures du système.

6.1 Difficultés rencontrées

a) Problèmes liés à l'importation et à la structuration des données. L'une des premières difficultés rencontrées concernait l'importation de données issues de fichiers Excel contenant plusieurs feuilles. Access ne permettant pas de fusionner automatiquement les données dans des tables existantes, plusieurs tentatives ont été nécessaires pour configurer correctement les options d'*Append Query*, éviter les doublons et garantir la compatibilité des types de champs. L'ajustement manuel des définitions de champs (index, clés, formats) a également été indispensable.

b) Gestion des clés étrangères et des listes déroulantes. La mise en place des relations s'est avérée plus complexe que prévu, notamment au niveau des formulaires utilisant des listes déroulantes (combo box). Des erreurs fréquentes sont apparues au début : valeurs non enregistrées, mauvais champ lié, ordres d'affichage incorrects ou encore incohérence entre l'identifiant stocké et le texte affiché. Il a fallu reconfigurer soigneusement la propriété *Control Source* et le champ *Bound Column* pour assurer le bon fonctionnement.

c) Contraintes d'intégrité référentielle et erreurs de suppression. La configuration d'intégrité référentielle a entraîné plusieurs blocages lors des suppressions ou modifications d'enregistrements. Par exemple, il était impossible de supprimer un élève déjà lié à une inscription ou un paiement, ce qui a nécessité des ajustements de logique (désactivation de certains liens, choix de cascades ou non, réorganisation des dépendances). Ces limites ont parfois rendu la phase de test plus longue qu'anticipé.

d) Problèmes de navigation et de synchronisation dans les formulaires. La synchronisation automatique entre formulaires principaux et sous-formulaires (notamment pour Inscription, Séance, Paiement) a posé des difficultés. Certains champs ne

se mettaient pas à jour sans recharger le formulaire, ce qui a nécessité l'utilisation de commandes telles que `Requery`, `Refresh` ou encore `Recalc`. Le comportement des formulaires imbriqués a demandé plusieurs ajustements.

e) Mise en place du code VBA. L'intégration du VBA, bien qu'essentielle pour l'automatisation, a représenté un défi important. Certaines erreurs comme *"Object required"*, *"You can't go to the specified record"* ou des problèmes de syntaxe ont nécessité des tests itératifs pour garantir la fiabilité du système. L'export PDF, les validations de champs et les calculs automatiques ont été particulièrement sensibles et ont exigé une attention soutenue.

f) Problèmes d'esthétique et d'ergonomie. L'harmonisation visuelle des formulaires s'est révélée plus complexe que prévu : alignement des contrôles, taille des zones de texte, couleurs, gestion des images, titres centrés, cohérence de la charte graphique... Plusieurs versions ont été nécessaires avant d'obtenir une interface stable, lisible et homogène.

g) Contraintes liées aux limites d'Access. Certaines limitations propres à Access ont également constitué des obstacles, notamment : la gestion d'un trop grand nombre de contrôles sur un formulaire, les erreurs de verrouillage de base (file locking) et les restrictions liées au multi-utilisateur. Cela a nécessité des compromis et l'adaptation de certaines fonctionnalités.

6.2 Points forts du système développé

Le système final présente plusieurs atouts notables :

- Une base de données claire, normalisée et extensible, conforme à la 3e forme normale ;
- Une interface intuitive et ergonomique, accessible aux non-spécialistes ;
- Des requêtes SQL performantes et facilement réutilisables ;
- Des automatisations VBA efficaces (calculs, vérifications, exportations) ;
- Une traçabilité complète de l'activité grâce aux rapports dynamiques.

6.3 Limites du système

Malgré sa stabilité et sa fiabilité, certaines limites persistent :

- Le système reste monoposte, ne permettant pas l'accès multi-utilisateurs
- L'interface pourrait être modernisée pour offrir une expérience plus fluide
- L'absence d'un module d'authentification limite la sécurité et la gestion des rôles.
- Les statistiques automatiques (indicateurs de performance, graphiques) ne sont pas encore intégrées.

6.4 Perspectives d'amélioration

Bien que l'application développée réponde pleinement aux besoins actuels du centre Mindcraft, plusieurs améliorations peuvent être envisagées afin d'enrichir ses fonctionnalités, d'optimiser ses performances et d'accompagner la croissance future du centre. Parmi les pistes d'évolution possibles :

- Migration vers une architecture client–serveur (MySQL, SQL Server ou PostgreSQL) pour permettre l'accès simultané par plusieurs utilisateurs sans risque de corruption des données.
- Passage à un système cloud/hybride pour permettre une sauvegarde automatique et une haute disponibilité.
- Ajout d'un module de gestion des présences (enregistrement automatique, export vers Excel, génération des absences).
- Gestion plus avancée des paiements : alertes de retard, factures automatiques, historique détaillé .
- Création d'un module d'évaluation des élèves (notes, progression, feedback).
- Génération d'un menu principal interactif, avec accès rapide aux statistiques et actions fréquentes.
- Intégration de macros ou scripts pour l'envoi automatique des rapports PDF chaque semaine ou chaque mois.
- Automatisation avancée des calculs de rémunération des coachs et du suivi comptable.
- Développement d'un tableau de bord interactif (Power BI, Tableau ou Excel) avec : taux de fréquentation, évolution des paiements, programmes les plus demandés)
- Création de KPI (indicateurs clés) pour soutenir la prise de décision.
- Développement d'une version web (PHP, Django, Laravel, Node.js) ou mobile (Flutter, React Native) permettant un accès distant sécurisé.
- Portail en ligne pour les parents : consultation des paiements, planning des séances, téléchargements des attestations.
- Mise en place d'un système d'authentification par rôles : administrateur, coach, direction.
- Protection renforcée des données sensibles (hashing, cryptage Access, mots de passe d'accès).

6.5 Conclusion finale

Ce projet de fin de module a constitué une expérience complète, alliant modélisation théorique, mise en œuvre pratique et développement d'une solution répondant à un besoin réel du centre Mindcraft.

La conception et l'implémentation de l'application ont permis d'explorer toutes les étapes du cycle de développement d'une base de données : analyse des besoins, conception du MCD/MLD/MPD, création des tables, réalisation des formulaires, automatisation via VBA et génération de rapports professionnels.

Au-delà de l'apprentissage technique, ce projet a démontré l'impact concret que peut avoir un outil numérique bien conçu sur une organisation. L'intégration de l'application au sein de Mindcraft a permis :

- une réduction significative du temps consacré à la gestion administrative ;
- une centralisation fiable des informations dans un système unique ;
- un suivi plus précis des paiements, des inscriptions et de la fréquentation ;
- une amélioration de la coordination entre les pôles administratif et pédagogique.

Grâce à cette solution, Mindcraft a franchi une étape importante dans sa transition numérique, en renforçant son efficacité opérationnelle et la qualité des services offerts aux élèves et à leurs familles.

Ainsi, la réussite de ce projet repose autant sur la qualité technique de l'application que sur la valeur ajoutée qu'elle apporte au centre. Il s'agit d'une base solide pouvant évoluer vers des fonctionnalités plus avancées, tableaux de bord interactifs, version web, automatisation étendue et ouvrant la voie à une gestion encore plus intelligente et connectée.

Appendices

A

Annexe A: Interfaces Access

A.1 Les Tableaux Access

tblEleve | tblCoach

ID_Coach	Nom	Prénom	Spécialité	TypeRémune	MontantRémuneratic	Email	Téléphone	Click to Add
8	El Ghemary	Farah	Robotics	Fixed	$0.00	farah@mindcra	(+122) 6 06 11	
9	El Amrani	Youssef	Programmatior	Fixed	$0.00	youssef@mind	(+122) 6 06 22	
10	Mahboub	Adil	3D Design & Pr	fixed	$0.00	adil@mindcraf	(+122) 6 06 33	
11	Bouziane	Oumaima	Robotique	Hourly	$250.00	oumaima.b@e	(+122) 6 06 44	
(New)					$0.00			

Figure A.1: *Tableau Coach*

tblInscription

ID_Inscriptic	ID_Eleve	ID_Séance	DateInscript	MontantPayé	Statut	Click to Add
6	10	6	15-Jan-25	$1,500.00	TRUE	
7	9	6	15-Jan-25	$2,500.00	TRUE	
8	17	6	06-Nov-25	$0.00	TRUE	
(New)				$0.00		

Figure A.2: *Tableau Inscription*

tblPaiement

ID_Paiement	DatePaieme	Montant	ModePaieme	PériodePaier	ID_Eleve	ID_Séance	Click to Add
4	20-Jan-25	$2,500.00	Cash	Janvier	10	6	
5	11-Nov-25	$1,500.00	Cash	Janvier	13	7	
(New)		$0.00					

Figure A.3: *Tableau Paiement*

tblProgramme

ID_Programı	NomPrograr	Description	Catégorie	Click to Add
5	LEGO Robotics	Build & code rc	Robotique	
6	Arduino Starteı	Electronics bas	Engineering	
7	Coding & Logic	Game design a	Programming	
8	Chess	Learn logic anc	Cognitive	
(New)				

Figure A.4: *Tableau Programme*

tblSalle

ID_Salle	NomSalle	CapacitéSal	Click to Add
1	Principale	16	
2	Science	12	
3	SoftSkills	20	
(New)		0	

Figure A.5: *Tableau Salle*

tblSéance

ID_Seance	JourSemaine	Heure	DateDébut	DateFin	Capacité	ID_Programı	ID_Coach	ID_Salle	Click to Add
6	Jeudi	17:00	10-Jan-25	20-Jun-25	12	7	10	1	
7	Mercredi	15:00	10-Jan-25	20-Jun-25	12	6	8	1	
8	Vendredi	16:00	10-Jan-25	20-Jun-25	12	6	8	3	
(New)					0				

Figure A.6: *Tableau Séance*

tblPaiement | tblAttestation

ID_Attestati	ID_Eleve	ID_Programı	DateAttesta	Statut	Click to Add
1	9	5	11/15/2025		
2	12	6	10/11/2025		
(New)					

Figure A.7: *Tableau Attestation*

A.2 Les Formulaires Access

Figure A.8: *Formulaire Inscription*

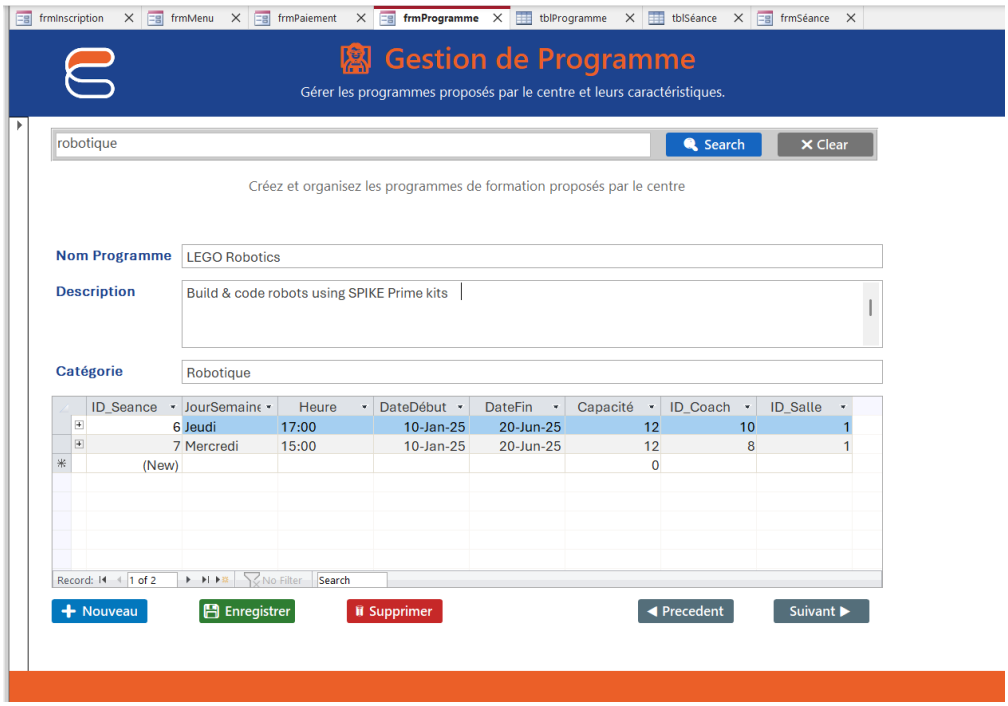

Figure A.9: *Formulaire Programme*

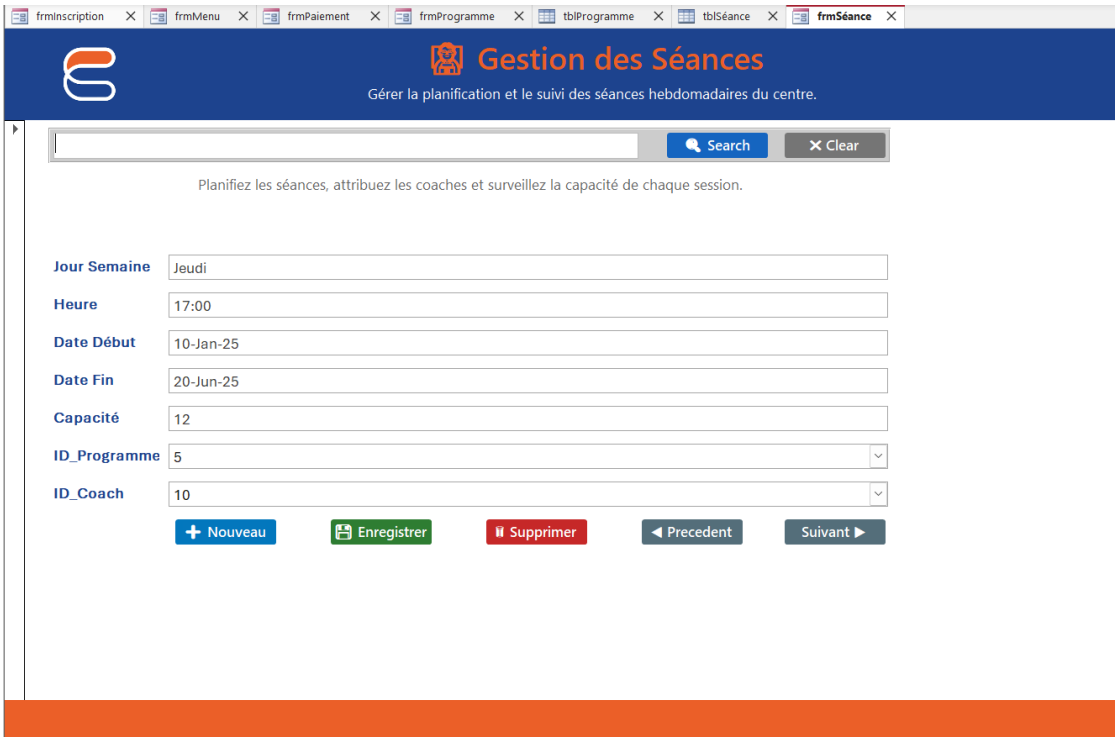

Figure A.10: *Formulaire Séance*

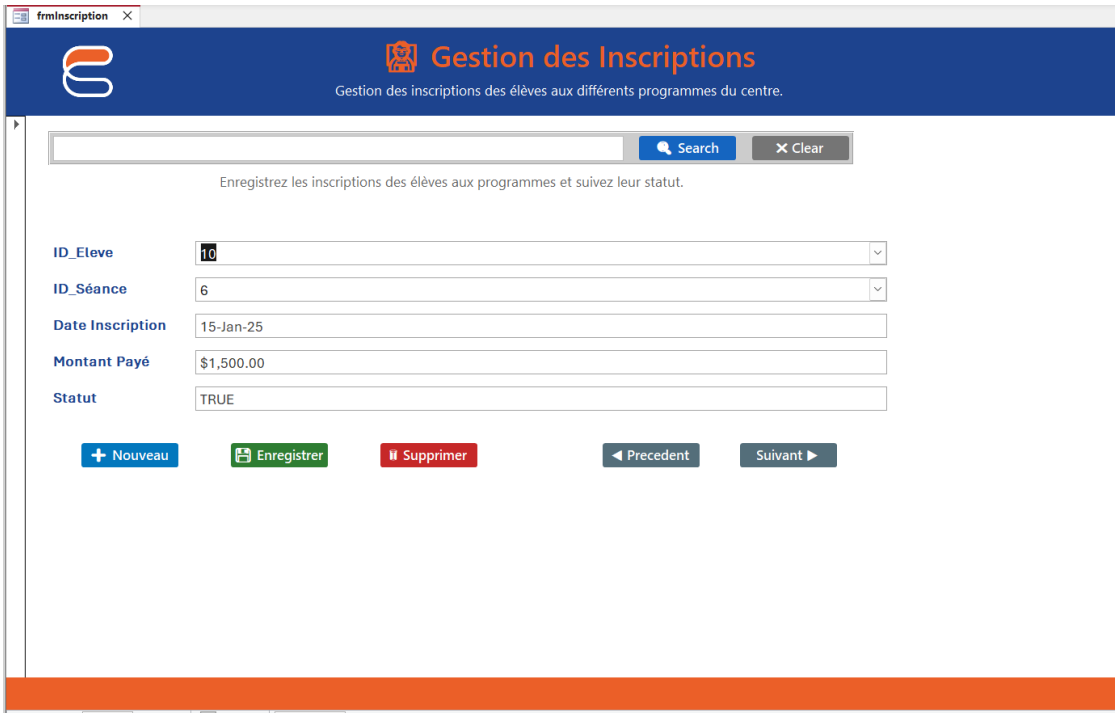

Figure A.11: *Formulaire Inscription*

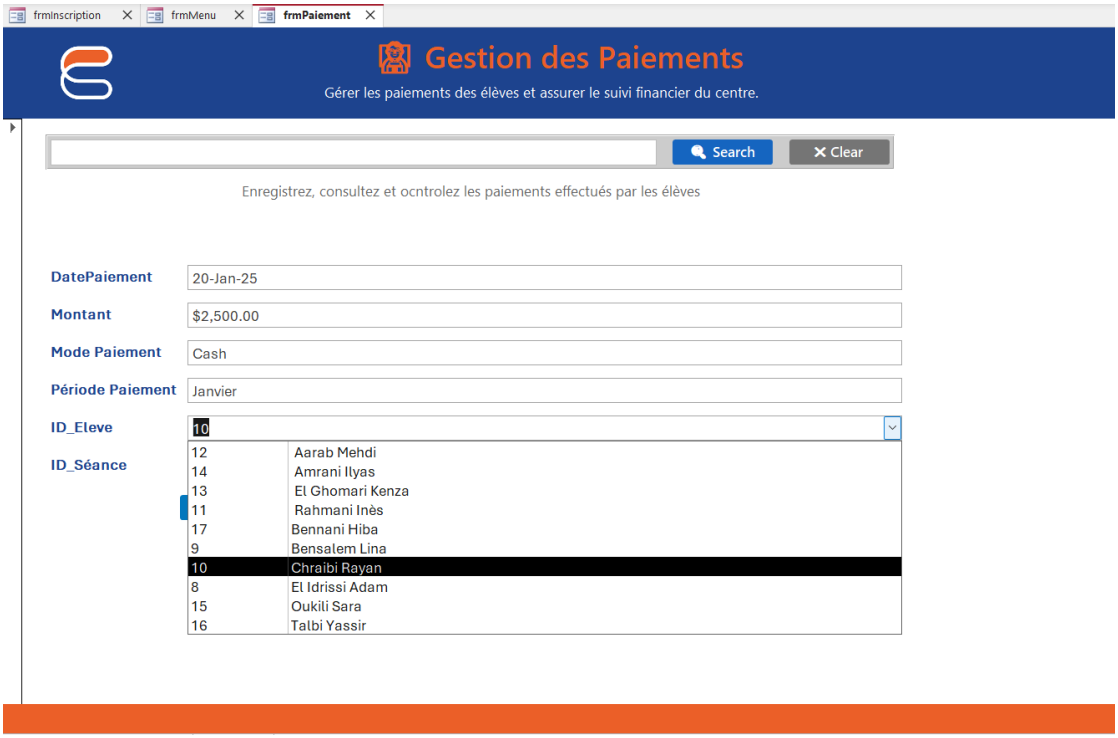

Figure A.12: *Formulaire Paiement*

B
Annexe B: Code SQL

B.1 Requête de génération d'attestation de participation

Cette requête permet d'obtenir une ligne par élève avec le nom du programme, le coach, et la période de formation. Elle est utilisée comme source de données du rapport `rpt_AttestationParticipation`.

```
SELECT
    [tblEleve].[ID_Eleve],
    [tblEleve].[Nom] & " " & [tblEleve].[Prénom] AS NomComplet,
    First ([tblProgramme].[NomProgramme]) AS NomProgramme,
    Min([tblSéance].[DateDébut]) AS DateDébut,
    Max([tblSéance].[DateFin]) AS DateFin,
    First ([tblCoach].[Nom] & " " & [tblCoach].[Prénom]) AS NomCoach
FROM
    (
        (
            (
                tblEleve
                INNER JOIN tblPaiement ON [tblEleve].[ID_Eleve] = [tblPaiement].[ID_Eleve
            )
            INNER JOIN tblSéance ON [tblPaiement].[ID_Séance] = [tblSéance].[ID_Seance]
        )
        INNER JOIN tblProgramme ON [tblSéance].[ID_Programme] = [tblProgramme].[ID_Progra
    )
    INNER JOIN tblCoach ON [tblSéance].[ID_Coach] = [tblCoach].[ID_Coach]
GROUP BY
    [tblEleve].[ID_Eleve],
    [tblEleve].[Nom],
    [tblEleve].[Prénom];
```

B.2 Requête de résumé des paiements mensuels

Cette requête regroupe les paiements par mois et calcule les totaux et moyennes. Elle alimente le rapport `rpt_RésuméPaiementsMensuels`.

```
SELECT
    Format([DatePaiement], "mmmm yyyy") AS Mois,
    Count([ID_Paiement]) AS NbPaiements,
    Sum([Montant]) AS TotalMensuel,
    Round(Avg([Montant]), 2) AS MontantMoyen
FROM
    tblPaiement
GROUP BY
    Format([DatePaiement], "mmmm yyyy")
ORDER BY
    Min([DatePaiement]);
```

B.3 Nombre d'élèves distincts par programme

Cette requête permet de compter le nombre d'élèves inscrits dans chaque programme, en tenant compte des participations distinctes. Elle est utilisée dans le rapport `rpt_ParticipationParProgramme`.

```
SELECT
    q.ID_Programme,
    COUNT(*) AS Nb_Élèves
FROM
    (
        SELECT DISTINCT
            pa.ID_Eleve,
            pr.ID_Programme
        FROM
            (
                tblProgramme AS pr
                INNER JOIN tblSéance AS se ON pr.ID_Programme = se.ID_Programme
            )
            INNER JOIN tblPaiement AS pa ON se.ID_Seance = pa.ID_Séance
    ) AS q
GROUP BY
    q.ID_Programme;
```

B.4 Nombre de séances par programme

Cette requête calcule le nombre total de séances planifiées pour chaque programme.

```
SELECT
    ID_Programme,
    COUNT(ID_Seance) AS NbSéances
FROM
    tblSéance
GROUP BY
    ID_Programme;
```

B.5 Synthèse complète par programme

Cette requête combine plusieurs sous-requêtes (`NbSéances`, `NbÉlèves`, `RevenusParProgramme`) pour produire un tableau complet : nombre de séances, nombre d'élèves distincts, et revenus totaux par programme. Elle constitue la base du rapport `rpt_ParticipationParProgramme`.

```
SELECT
    p.NomProgramme,
    s.NbSéances,
    e.Nb_Élèves,
    r.TotalRevenus
FROM
    (
        (
            tblProgramme AS p
            LEFT JOIN qryB_NbSeances_ParProgramme AS s
                ON p.ID_Programme = s.ID_Programme
        )
        LEFT JOIN qryB_NbElevesDistinct_ParProgramme AS e
            ON p.ID_Programme = e.ID_Programme
    )
    LEFT JOIN qryB_Revenus_ParProgramme AS r
        ON p.ID_Programme = r.ID_Programme
ORDER BY
    p.NomProgramme;
```

B.6 Revenus totaux par programme

Cette requête calcule le revenu total généré par chaque programme. Elle est utilisée dans la synthèse `rpt_ParticipationParProgramme`.

```
SELECT
    pr.ID_Programme,
    Sum(pa.Montant) AS TotalRevenus
FROM
    (
        tblProgramme AS pr
        INNER JOIN tblSéance AS se ON pr.ID_Programme = se.ID_Programme
    )
    INNER JOIN tblPaiement AS pa ON se.ID_Seance = pa.ID_Séance
GROUP BY
    pr.ID_Programme;
```

B.7 Suivi détaillé des paiements par élève

Cette requête fournit un historique complet des paiements par élève, avec le programme, la séance, le montant et la période couverte. Elle est utilisée dans le rapport `rpt_RelevéPaiementsÉlève`.

```
SELECT
    tblEleve.Nom & " " & tblEleve.Prénom AS NomÉlève,
```

```
    tblProgramme.NomProgramme,
    tblSéance.JourSemaine,
    tblSéance.Heure,
    tblPaiement.Montant,
    tblPaiement.PériodePaiement,
    tblPaiement.DatePaiement
FROM
    (
        (
            (
                tblEleve
                INNER JOIN tblPaiement
                    ON tblEleve.ID_Eleve = tblPaiement.ID_Eleve
            )
            INNER JOIN tblSéance
                ON tblPaiement.ID_Séance = tblSéance.ID_Seance
        )
        INNER JOIN tblProgramme
            ON tblSéance.ID_Programme = tblProgramme.ID_Programme
    )
    LEFT JOIN tblCoach
        ON tblSéance.ID_Coach = tblCoach.ID_Coach
ORDER BY
    tblEleve.Nom,
    tblProgramme.NomProgramme,
    tblPaiement.DatePaiement;
```

B.8 Emploi du temps hebdomadaire

Cette requête agrège les séances, programmes, coachs et salles pour générer l'emploi du temps imprimable du centre. Elle est utilisée par le rapport `rpt_EmploiDuTemps`.

```
SELECT
    S.JourSemaine,
    S.Heure,
    P.NomProgramme,
    C.Nom & " " & C.Prénom AS Coach,
    Sa.NomSalle
FROM
    (
        (
            tblSéance AS S
            INNER JOIN tblProgramme AS P
                ON S.ID_Programme = P.ID_Programme
        )
        INNER JOIN tblCoach AS C
            ON S.ID_Coach = C.ID_Coach
    )
    LEFT JOIN tblSalle AS Sa
        ON S.ID_Salle = Sa.ID_Salle
```

```
ORDER BY
    S.JourSemaine,
    S.Heure;
```

C

Annexe C: Code VBA

Formulaire Eleve

```
Private Sub cmdSearch_Click()
    Dim q As String
    q = Nz(Me.txtSearchStudent, "")

    If Len(q) = 0 Then
        ' No text ? remove filter
        Me.FilterOn = False
        Exit Sub
    End If

    ' Escape single quotes in the search text
    q = Replace(q, "'", "''")

    ' Filter on Nom or Prenom contains the typed text (case-insensitive)
    Me.Filter = "[Nom] LIKE '*" & q & "*' OR [Prénom] LIKE '*" & q & "*'"
    Me.FilterOn = True
End Sub


Private Sub cmdClear_Click()
    Me.txtSearchStudent = Null
    Me.FilterOn = False
End Sub


Private Sub cmdNew_Click()
    DoCmd.GoToRecord , , acNewRec
    If Not Me.NewRecord Then
        ' Safety: ensure we're really on a new record
        DoCmd.GoToRecord , , acNewRec
    End If
End Sub
```

```
Private Sub cmdSave_Click()
    ' Triggers BeforeUpdate; then saves if validation passes
    DoCmd.RunCommand acCmdSaveRecord
End Sub


Private Sub cmdDelete_Click()
    If Me.NewRecord Then Exit Sub
    If MsgBox("Delete this student?", vbYesNo + vbQuestion, "Confirm") = vbYes Then
        DoCmd.RunCommand acCmdDeleteRecord
    End If
End Sub


Private Sub Form_BeforeUpdate(Cancel As Integer)
    ' Nom and Prenom must not be empty
    If Nz(Me.Nom, "") = "" Then
        MsgBox "Please enter the student's last name.", vbExclamation, "Missing last name
        Me.Nom.SetFocus
        Cancel = True
        Exit Sub
    End If

    If Nz(Me.Prénom, "") = "" Then
        MsgBox "Please enter the student's first
        name.", vbExclamation, "Missing first name"
        Me.Prénom.SetFocus
        Cancel = True
        Exit Sub
    End If

    ' DateNaissance: if filled, must be <= Today
    If Not IsNull(Me.DateNaissance) Then
        If Me.DateNaissance > Date Then
            MsgBox "Birth date cannot be in the future.", vbExclamation, "Invalid date"
            Me.DateNaissance.SetFocus
            Cancel = True
            Exit Sub
        End If
    End If

    ' EmailParent: simple check (contains @ and . after)
    If Nz(Me.EmailParent, "") <> "" Then
        Dim e As String
        e = Me.EmailParent
        If InStr(e, "@") = 0 Or InStrRev(e, ".") < InStr(e, "@") + 2 Then
            MsgBox "Please enter a valid parent email.", vbExclamation, "Invalid email"
            Me.EmailParent.SetFocus
```

```
            Cancel = True
            Exit Sub
        End If
    End If
End Sub
```

Formulaire Coach

```
' ==============================
' ======== SEARCH / CLEAR =====
' ==============================
Private Sub cmdSearch_Click()
    Dim q As String
    q = Nz(Me.txtSearchCoach, "")

    If Len(q) = 0 Then
        Me.FilterOn = False
        Exit Sub
    End If

    q = Replace(q, "'", "''")
    ' Filtrer par nom, prénom ou spécialité
    Me.Filter = "([Nom] LIKE '*" & q & "*') OR ([Prénom] LIKE '*" & q & "*') OR
    ([Spécialité] LIKE '*" & q & "*')"
    Me.FilterOn = True
End Sub

Private Sub cmdClear_Click()
    Me.txtSearchCoach = Null
    Me.FilterOn = False
End Sub


' ==============================
' ======== NEW / SAVE / DELETE =
' ==============================
Private Sub cmdNew_Click()
    DoCmd.GoToRecord , , acNewRec
End Sub

Private Sub cmdSave_Click()
    On Error GoTo Err_Save
    If Me.Dirty Then
        DoCmd.RunCommand acCmdSaveRecord
        MsgBox "Coach enregistré avec succès.", vbInformation, "Sauvegarde"
    End If
    Exit Sub
```

```
Err_Save:
    MsgBox "Erreur lors de l'enregistrement : " & Err.Description, vbCritical, "Erreur"
End Sub

Private Sub cmdDelete_Click()
    If Me.NewRecord Then Exit Sub
    If MsgBox("Voulez-vous vraiment supprimer ce coach ?", vbYesNo + vbQuestion,
    "Confirmation") = vbYes Then
        DoCmd.RunCommand acCmdDeleteRecord
        MsgBox "Coach supprimé.", vbInformation
    End If
End Sub


' ==============================
' ======== NAVIGATION ==========
' ==============================
Private Sub cmdPrev_Click()
    On Error Resume Next
    DoCmd.GoToRecord , , acPrevious
End Sub

Private Sub cmdNext_Click()
    On Error Resume Next
    DoCmd.GoToRecord , , acNext
End Sub


' ==============================
' ======== VALIDATION ==========
' ==============================
Private Sub Form_BeforeUpdate(Cancel As Integer)
    ' Vérifier le nom
    If Nz(Me.Nom, "") = "" Then
        MsgBox "Veuillez saisir le nom du coach.", vbExclamation, "Nom requis"
        Me.Nom.SetFocus
        Cancel = True
        Exit Sub
    End If

    ' Vérifier le prénom
    If Nz(Me.Prénom, "") = "" Then
        MsgBox "Veuillez saisir le prénom du coach.", vbExclamation, "Prénom requis"
        Me.Prénom.SetFocus
        Cancel = True
        Exit Sub
    End If

    ' Vérifier l'email
    If Nz(Me.Email, "") <> "" Then
```

```
        Dim e As String
        e = Me.Email
        If InStr(e, "@") = 0 Or InStrRev(e, ".") < InStr(e, "@") + 2 Then
            MsgBox "Veuillez saisir une adresse email valide.", vbExclamation,
            "Email invalide"
            Me.Email.SetFocus
            Cancel = True
            Exit Sub
        End If
    End If

    ' Vérifier le type de rémunération
    If Nz(Me.TypeRémunération, "") = "" Then
        MsgBox "Veuillez indiquer le type de rémunération.", vbExclamation,
        "Champ requis"
        Me.TypeRémunération.SetFocus
        Cancel = True
        Exit Sub
    End If

    ' Vérifier le montant de rémunération
    If Nz(Me.MontantRémuneration, 0) < 0 Then
        MsgBox "Le montant de rémunération ne peut pas être négatif.", vbExclamation,
        "Valeur invalide"
        Me.MontantRémuneration.SetFocus
        Cancel = True
        Exit Sub
    End If
End Sub
```

Formulaire Inscription

```
' ==============================
' ======== SEARCH / CLEAR =====
' ==============================
Private Sub cmdSearch_Click()
    Dim q As String
    q = Nz(Me.txtSearchInscription, "")

    If Len(q) = 0 Then
        Me.FilterOn = False
        Exit Sub
    End If

    q = Replace(q, "'", "''")

    ' Filtrer par nom/prénom de l'élève ou par séance (jour ou heure)
```

```
    Me.Filter = "([ID_Eleve] IN (SELECT ID_Eleve FROM tblEleve WHERE Nom LIKE '*" & q & '
    & "*')) OR " & _
                "([ID_Séance] IN (SELECT ID_Séance FROM tblSéance WHERE JourSemaine
                LIKE '*" & q & "*' OR Heure LIKE '*" & q & "*'))"
    Me.FilterOn = True
End Sub

Private Sub cmdClear_Click()
    Me.txtSearchInscription = Null
    Me.FilterOn = False
End Sub


' ==============================
' ======== NEW / SAVE / DELETE =
' ==============================
Private Sub cmdNew_Click()
    DoCmd.GoToRecord , , acNewRec
End Sub

Private Sub cmdSave_Click()
    On Error GoTo Err_Save
    If Me.Dirty Then
        DoCmd.RunCommand acCmdSaveRecord
        MsgBox "Inscription enregistrée avec succès.", vbInformation, "Sauvegarde"
    End If
    Exit Sub
Err_Save:
    MsgBox "Erreur lors de l'enregistrement : " & Err.Description, vbCritical, "Erreur"
End Sub

Private Sub cmdDelete_Click()
    If Me.NewRecord Then Exit Sub
    If MsgBox("Voulez-vous vraiment supprimer cette inscription ?", vbYesNo + vbQuestion
    "Confirmation") = vbYes Then
        DoCmd.RunCommand acCmdDeleteRecord
        MsgBox "Inscription supprimée.", vbInformation
    End If
End Sub


' ==============================
' ======== NAVIGATION ==========
' ==============================
Private Sub cmdPrev_Click()
    On Error Resume Next
    DoCmd.GoToRecord , , acPrevious
End Sub

Private Sub cmdNext_Click()
```

```
    On Error Resume Next
    DoCmd.GoToRecord , , acNext
End Sub


' ==============================
' ======== VALIDATION ==========
' ==============================
Private Sub Form_BeforeUpdate(Cancel As Integer)
    ' Élève requis
    If IsNull(Me.ID_Eleve) Or Me.ID_Eleve = 0 Then
        MsgBox "Veuillez choisir un élève.", vbExclamation, "Élève requis"
        If ControlExists("cboEleve") Then Me.cboEleve.SetFocus
        Cancel = True
        Exit Sub
    End If

    ' Séance requise
    If IsNull(Me.ID_Séance) Or Me.ID_Séance = 0 Then
        MsgBox "Veuillez choisir une séance.", vbExclamation, "Séance requise"
        If ControlExists("cboSeance") Then Me.cboSeance.SetFocus
        Cancel = True
        Exit Sub
    End If

    ' Montant payé
    If Nz(Me.MontantPayé, 0) < 0 Then
        MsgBox "Le montant payé ne peut pas être négatif.", vbExclamation, "Montant
        invalide"
        Me.MontantPayé.SetFocus
        Cancel = True
        Exit Sub
    End If
End Sub


' ==============================
' ======== COMBO CASCADE =======
' ==============================
Private Sub cboEleve_AfterUpdate()
    On Error Resume Next
    If ControlExists("cboSeance") Then
        Me.cboSeance.Requery
        Me.cboSeance = Null
    End If
End Sub


' ==============================
' ======== UTILITY ============
```

```
' ==============================
' Vérifie si un contrôle existe avant de l'utiliser
Private Function ControlExists(ctrlName As String) As Boolean
    On Error Resume Next
    ControlExists = Not (Me.Controls(ctrlName) Is Nothing)
End Function
```

Formulaire Menu

```
Private Sub cmdTableauDeBord_Click()
    DoCmd.OpenReport "rpt_TableauDeBordMindcraft", acViewPreview
End Sub

Private Sub cmdProgrammeParticipation_Click()
    DoCmd.OpenReport "qryB_ProgrammeParticipation", acViewPreview
End Sub

Private Sub cmdAttestation_Click()
    DoCmd.OpenReport "rpt_AttestationParticipation", acViewPreview
End Sub

Private Sub cmdRelevePaiements_Click()
    DoCmd.OpenReport "rpt_RelevePaiementParEleve", acViewPreview
End Sub

Private Sub cmdResumePaiementsMensuels_Click()
    DoCmd.OpenReport "rpt_RésuméPaiementsMensuels", acViewPreview
End Sub
```

Formulaire Paiement

```
' ==============================
' ======== SEARCH / CLEAR =====
' ==============================
Private Sub cmdSearch_Click()
    Dim q As String
    q = Nz(Me.txtSearchPaiement, "")

    If Len(q) = 0 Then
        Me.FilterOn = False
        Exit Sub
    End If

    q = Replace(q, "'", "''")
```

```
    ' Filtrer par nom/prénom de l'élève ou par séance (jour/heure)
    Me.Filter = "([ID_Eleve] IN (SELECT ID_Eleve FROM tblEleve WHERE Nom LIKE '*" & q
    & "*' OR Prénom LIKE '*" & q & "*')) OR " & _
                "([ID_Séance] IN (SELECT ID_Séance FROM tblSéance WHERE JourSemaine
                LIKE '*" & q & "*' OR Heure LIKE '*" & q & "*'))"
    Me.FilterOn = True
End Sub

Private Sub cmdClear_Click()
    Me.txtSearchPaiement = Null
    Me.FilterOn = False
End Sub


' ==============================
' ======== NEW / SAVE / DELETE =
' ==============================
Private Sub cmdNew_Click()
    DoCmd.GoToRecord , , acNewRec
End Sub

Private Sub cmdSave_Click()
    On Error GoTo Err_Save
    If Me.Dirty Then
        DoCmd.RunCommand acCmdSaveRecord
        MsgBox "Paiement enregistré avec succès.", vbInformation, "Sauvegarde"
    End If
    Exit Sub
Err_Save:
    MsgBox "Erreur lors de l'enregistrement : " & Err.Description, vbCritical, "Erreur"
End Sub

Private Sub cmdDelete_Click()
    If Me.NewRecord Then Exit Sub
    If MsgBox("Voulez-vous vraiment supprimer ce paiement ?", vbYesNo + vbQuestion,
    "Confirmation") = vbYes Then
        DoCmd.RunCommand acCmdDeleteRecord
        MsgBox "Paiement supprimé.", vbInformation
    End If
End Sub


' ==============================
' ======== NAVIGATION ==========
' ==============================
Private Sub cmdPrev_Click()
    On Error Resume Next
    DoCmd.GoToRecord , , acPrevious
End Sub
```

```
Private Sub cmdNext_Click()
    On Error Resume Next
    DoCmd.GoToRecord , , acNext
End Sub


' ==============================
' ======== VALIDATION ==========
' ==============================
Private Sub Form_BeforeUpdate(Cancel As Integer)
    ' Élève requis
    If IsNull(Me.ID_Eleve) Or Me.ID_Eleve = 0 Then
        MsgBox "Veuillez choisir un élève.", vbExclamation, "Élève requis"
        If ControlExists("cboEleve") Then Me.cboEleve.SetFocus
        Cancel = True
        Exit Sub
    End If

    ' Séance requise
    If IsNull(Me.ID_Séance) Or Me.ID_Séance = 0 Then
        MsgBox "Veuillez choisir une séance.", vbExclamation, "Séance requise"
        If ControlExists("cboSeance") Then Me.cboSeance.SetFocus
        Cancel = True
        Exit Sub
    End If

    ' Montant requis
    If Nz(Me.Montant, 0) <= 0 Then
        MsgBox "Veuillez saisir un montant valide.", vbExclamation, "Montant invalide"
        Me.Montant.SetFocus
        Cancel = True
        Exit Sub
    End If
End Sub


' ==============================
' ======== COMBO CASCADE =======
' ==============================
Private Sub cboEleve_AfterUpdate()
    On Error Resume Next
    If ControlExists("cboSeance") Then
        Me.cboSeance.Requery
        Me.cboSeance = Null
    End If
End Sub


' ==============================
' ======== UTILITY ============
```

```
' ==============================
' Vérifie si un contrôle existe avant de l'utiliser
Private Function ControlExists(ctrlName As String) As Boolean
    On Error Resume Next
    ControlExists = Not (Me.Controls(ctrlName) Is Nothing)
End Function
```

Formulaire Programme

```
' ==============================
' ======== SEARCH / CLEAR =====
' ==============================
Private Sub cmdSearch_Click()
    Dim q As String
    q = Nz(Me.txtSearchProgramme, "")

    If Len(q) = 0 Then
        Me.FilterOn = False
        Exit Sub
    End If

    q = Replace(q, "'", "''")
    ' Filtrer par nom ou catégorie
    Me.Filter = "([NomProgramme] LIKE '*" & q & "*') OR ([Catégorie] LIKE '*" & q & "*')"
    Me.FilterOn = True
End Sub

Private Sub cmdClear_Click()
    Me.txtSearchProgramme = Null
    Me.FilterOn = False
End Sub


' ==============================
' ======== NEW / SAVE / DELETE =
' ==============================
Private Sub cmdNew_Click()
    DoCmd.GoToRecord , , acNewRec
End Sub

Private Sub cmdSave_Click()
    On Error GoTo Err_Save
    If Me.Dirty Then
        DoCmd.RunCommand acCmdSaveRecord
        MsgBox "Programme enregistré avec succès.", vbInformation, "Sauvegarde"
    End If
    Exit Sub
Err_Save:
```

```
    MsgBox "Erreur lors de l'enregistrement : " & Err.Description, vbCritical, "Erreur"
End Sub

Private Sub cmdDelete_Click()
    If Me.NewRecord Then Exit Sub
    If MsgBox("Voulez-vous vraiment supprimer ce programme ?", vbYesNo + vbQuestion,
    "Confirmation") = vbYes Then
        DoCmd.RunCommand acCmdDeleteRecord
        MsgBox "Programme supprimé.", vbInformation
    End If
End Sub


' ==============================
' ======== NAVIGATION ==========
' ==============================
Private Sub cmdPrev_Click()
    On Error Resume Next
    DoCmd.GoToRecord , , acPrevious
End Sub

Private Sub cmdNext_Click()
    On Error Resume Next
    DoCmd.GoToRecord , , acNext
End Sub


' ==============================
' ======== VALIDATION ==========
' ==============================
Private Sub Form_BeforeUpdate(Cancel As Integer)
    ' Vérifier le nom du programme
    If Nz(Me.NomProgramme, "") = "" Then
        MsgBox "Veuillez saisir le nom du programme.", vbExclamation, "Nom requis"
        Me.NomProgramme.SetFocus
        Cancel = True
        Exit Sub
    End If

    ' Vérifier la catégorie
    ' ?? Access ne gère pas les accents dans les noms de champs VBA ? utiliser le vrai
    nom de champ
    ' Si ton champ s'appelle [Catégorie] avec accent, renomme-le sans accent dans
    la table (ex: Categorie)
    If Nz(Me.Catégorie, "") = "" Then
        MsgBox "Veuillez saisir la catégorie du programme.", vbExclamation, "Catégorie
        requise"
        Me.Catégorie.SetFocus
        Cancel = True
        Exit Sub
```

```
    End If
End Sub
```

Formulaire Séance

```
' ==============================
' ======== SEARCH / CLEAR =====
' ==============================
Private Sub cmdSearch_Click()
    Dim q As String
    q = Nz(Me.txtSearchSeance, "")

    If Len(q) = 0 Then
        Me.FilterOn = False
        Exit Sub
    End If

    q = Replace(q, "'", "''")

    ' Filter by Jour, Heure, Programme, or Coach
    Me.Filter = "([JourSemaine] LIKE '*" & q & "*') OR " & _
                "([Heure] LIKE '*" & q & "*') OR " & _
                "([ID_Programme] IN (SELECT ID_Programme FROM tblProgramme WHERE
                NomProgramme LIKE '*" & q & "*')) OR " & _
                "([ID_Coach] IN (SELECT ID_Coach FROM tblCoach WHERE Nom LIKE
                '*" & q & "*' OR Prénom LIKE '*" & q & "*'))"
    Me.FilterOn = True
End Sub

Private Sub cmdClear_Click()
    Me.txtSearchSeance = Null
    Me.FilterOn = False
End Sub


' ==============================
' ======== NEW / SAVE / DELETE =
' ==============================
Private Sub cmdNew_Click()
    DoCmd.GoToRecord , , acNewRec
End Sub

Private Sub cmdSave_Click()
    If Me.Dirty Then
        DoCmd.RunCommand acCmdSaveRecord
        MsgBox "Séance enregistrée avec succès.", vbInformation, "Sauvegarde"
    End If
```

```
End Sub

Private Sub cmdDelete_Click()
    If Me.NewRecord Then Exit Sub
    If MsgBox("Voulez-vous vraiment supprimer cette séance ?", vbYesNo +
    vbQuestion, "Confirmation") = vbYes Then
        DoCmd.RunCommand acCmdDeleteRecord
        MsgBox "Séance supprimée.", vbInformation
    End If
End Sub


' ==============================
' ======== NAVIGATION ==========
' ==============================
Private Sub cmdPrev_Click()
    On Error Resume Next
    DoCmd.GoToRecord , , acPrevious
End Sub

Private Sub cmdNext_Click()
    On Error Resume Next
    DoCmd.GoToRecord , , acNext
End Sub


' ==============================
' ======== VALIDATION ==========
' ==============================
Private Sub Form_BeforeUpdate(Cancel As Integer)
    ' Vérification du programme
    If IsNull(Me.ID_Programme) Or Me.ID_Programme = 0 Then
        MsgBox "Veuillez choisir un programme.", vbExclamation, "Programme requis"
        Me.cboProgramme.SetFocus
        Cancel = True
        Exit Sub
    End If

    ' Vérification du coach
    If IsNull(Me.ID_Coach) Or Me.ID_Coach = 0 Then
        MsgBox "Veuillez choisir un coach.", vbExclamation, "Coach requis"
        Me.cboCoach.SetFocus
        Cancel = True
        Exit Sub
    End If

    ' Vérification de la cohérence des dates
    If Not IsNull(Me.DateDébut) And Not IsNull(Me.DateFin) Then
        If Me.DateFin < Me.DateDébut Then
            MsgBox "La date de fin doit être postérieure à la date de début.",
```

```
            vbExclamation, "Dates invalides"
            Me.DateFin.SetFocus
            Cancel = True
            Exit Sub
        End If
    End If
End Sub
```