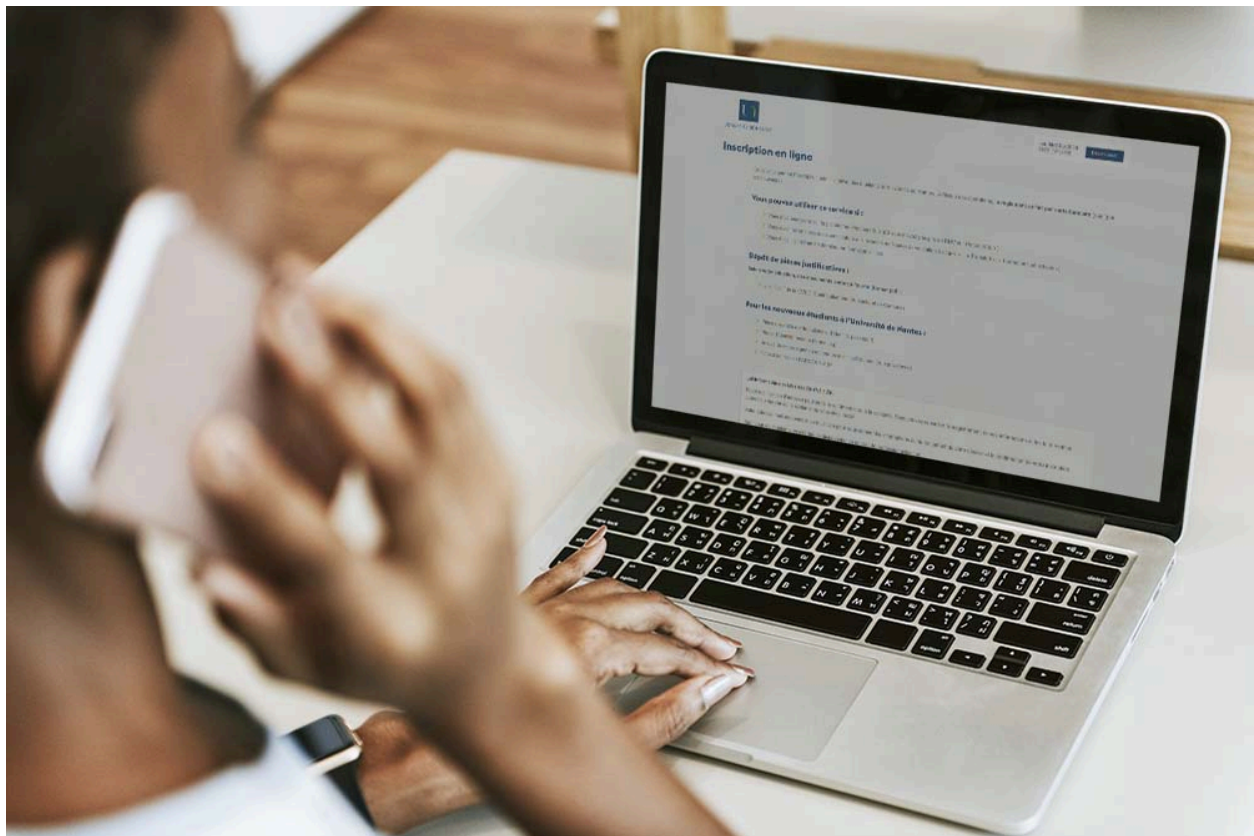


## *Section 2*

# RAPPORT DU PROJET II :

## Gestion des inscriptions en ligne pour la formation KFOKAM 48



---

# INTRODUCTION

## a. Contexte

L'école de formation KFOKAM48 accueille chaque année des centaines d'étudiants. L'inscription de ces étudiants ainsi que leur répartition dans les différentes formations n'est pas une chose aisée à gérer pour le personnel de la scolarité. Le service de scolarité doit collecter une multitude de fiches remplies par les étudiants et les classer manuellement. Il a également la lourde tâche d'attribuer des numéros de matricules à tous les étudiants inscrits. L'absence d'une application informatique permettant la gestion en ligne entraîne un travail pénible pour les agents du service de la scolarité qui se voient submergés et une longue file d'attente pour les étudiants.

## b. Objectifs

Notre projet vise à mettre sur pied une base de données qui va faciliter l'enrôlement de nouveaux étudiants à travers un système intelligent de « **Gestion des Inscriptions en Ligne** ». Le système doit permettre d'enregistrer et de suivre les étudiants, les formations, les inscriptions, et les paiements.

---

# MÉTHODOLOGIE

Pour parvenir à un système qui répond pleinement aux exigences susmentionnées, nous avons choisi de procéder méthodiquement.

## 1. Choix des entités et définition des associations

Suivant les exigences de notre système qui sont entre autres :

a. ***Pour les étudiants :***

- Création du compte étudiant ;
- **Consultation des offres de formation ;**
- **Inscription en ligne ;**
- **Suivi de l'inscription ;**
- **Gestion du compte ;**

b. ***Pour l'administration de l'école :***

- ***La gestion des inscriptions***
- Enregistrement des inscriptions ;
- Dénombrement des inscrits ;
- Recherche de inscrits par filière;

c. ***La gestion des formations :***

- Création, modification et suppression des offres de formation ;
- Affichage des formations offertes ainsi que le nombre de place disponible ;
- Appréciation des formations les plus sollicitées ;
- Définition les prérequis et les capacités d'accueil pour chaque formation ;
- Fixation des dates d'ouverture et de clôture des inscriptions.

---

d. **la gestion des paiements :**

- Effectuer le paiement en ligne
- lister les paiements effectués par les étudiants
- Émettre des factures et des reçus.

Nous avons répertorié 4 entités et 3 associations pour remplir ces exigences. Il s'agit de :

**Entités**

- **Formations** : id\_formation, nom\_formation, description, date\_debut, date\_fin, tarif
- **Etudiants** : id\_etudiant, nom, prenom, date\_naissance, email, telephone, date\_inscription
- **Inscriptions** : id\_inscription, id\_etudiant, id\_formation, date\_inscription, statut\_paiement
- **Paiements** : id\_paiement, id\_inscription, montant, date\_paiement, methode\_paiement

**Associations**

Association	Cardinalité	Explication
Prendre en compte	(1.1 : 1.N)	1 inscription prend en compte une et une seule formation ; et une formation est prise en compte par une ou plusieurs informations.
Faire	(1.1 : 1.N)	1 inscription peut être faite par 1 et 1 seul étudiant tandis qu'un étudiant peut faire une ou plusieurs inscriptions.
Demande	(1:N : 1.1)	1 inscription peut se faire en un ou plusieurs paiements et un paiement correspond à une et une seule inscription.

---

## 2. Création de la Base de données et des tables

Suivant le rendu de notre MCD présenté dans la documentation technique, nous avons créé une base de données “**ecole**” ; des tables à base du langage SQL et nous leur avons attribué les caractéristiques suivantes.

### a. Table Formations

Caractéristiques	Nom du champ	Type	NULL?
Identifiant de la formation	id_formation	AUTO_INCREMENT	Non
Nom de la formation	nom_formation	VARCHAR (50)	Non
Description de la formation	description	TEXT	Oui
Date de début de la formation	date_debut	DATE	Non
Date de fin de la formation	date_fin	DATE	Non
Coût de la formation	tarif	Double unsigned	Non
nombre de places disponibles	nombre_places	INT(50)	Non

---

**b. Table Etudiants**

Caractéristiques	Nom du champ	Type	NULL?
Identifiant de l'étudiant	id_etudiant	AUTO_INCREMENT	Non
Nom de l'étudiant	nom	VARCHAR (50)	Non
Prénom de l'étudiant	prénom	VARCHAR (50)	non
Date de naissance	Date_naissance	DATE	non
Adresse email de l'étudiant	email	VARCHAR (50)	Non
Numéro de téléphone de l'étudiant	telephone	VARCHAR (50)	Non
Date d'ouverture de compte	date_inscription	DATE	Non

**c. Table Inscriptions**

Caractéristiques	Nom du champ	Type	NULL?
Identifiant de l'inscription	id_inscription	AUTO_INCREMENT	Non
Date de l'inscription à une formation	date_inscription	DATE	Non
Etat de solvabilité d'un étudiant	Statut paiement	tinyint(1)	Non
Clé étrangère de la table etudiant	id_etudiant_Etudiants	int	Non
Clé étrangère de la table formation	id_formation_formations	int	Non

#### d. Table Paiements

Caractéristiques	Nom du champ	Type	NULL?
L'identifiant du paiement	id_paiement	AUTO_INCREMENT	Non
Montant versé par l'étudiant	montant	double	Non
Date de paiement	date_paiement	date	Non
Méthode de paiement	methode_paiement	VARCHAR (50)	Non
Clé étrangère de la table inscription	id_inscription_Inscriptions	int	Non

### 3. Ajout des données

La création de nos différentes tables étant faite, nous avons pour une opérationnalisation de notre base de données inséré une série de données pour tester son bon fonctionnement.

Chacune de ces données remplit pleinement les caractéristiques que nous avons définies dans les tables ci-dessus. Nous avons respectivement dans notre bibliothèque à l'heure actuelle :

- 06 Formations ;
- 09 potentiels étudiants ;
- 09 inscrits ;
- 17 paiements dont 7 par carte bancaire, 6 par chèque et 4 par orange money.

---

## 4. Création des requêtes

Notre système en plus de gérer les inscriptions aide le service de scolarité à émettre des rapports et à faire un bilan de gestion ceci à travers des requêtes qu'il envoie au système.

Nous avons prévu des requêtes pour :

- lister tous les étudiants inscrits à une certaine formation ;
- trouver les paiements effectués par un certain étudiant ;
- lister les inscriptions en cours d'un étudiant ;
- compter le nombre total d'inscriptions par formation.

## 5. Définition des fonctions et procédures de notre système

Une fonction est un code qui effectue une série d'instructions bien précises, et renvoie le résultat de ces instructions.

En matière de fonctions et de procédures, notre système prend en compte :

- Une procédure d'ajout d'une inscription
- Une fonction de vérification de la disponibilité d'une formation (nombre de places restantes).

## 6. Mise en place des vues et triggers

Les **vues** sont des objets de la base de données, constitués d'un nom, et d'une requête de sélection. Une fois qu'une vue est définie, on peut l'utiliser comme on le ferait avec une table ; table qui serait constituée des données sélectionnées par la requête définissant la vue. Pour notre projet nous avons défini une vue **pour afficher les détails des inscriptions** (nom de la formation, nom de l'étudiant, date d'inscription, statut de paiement).

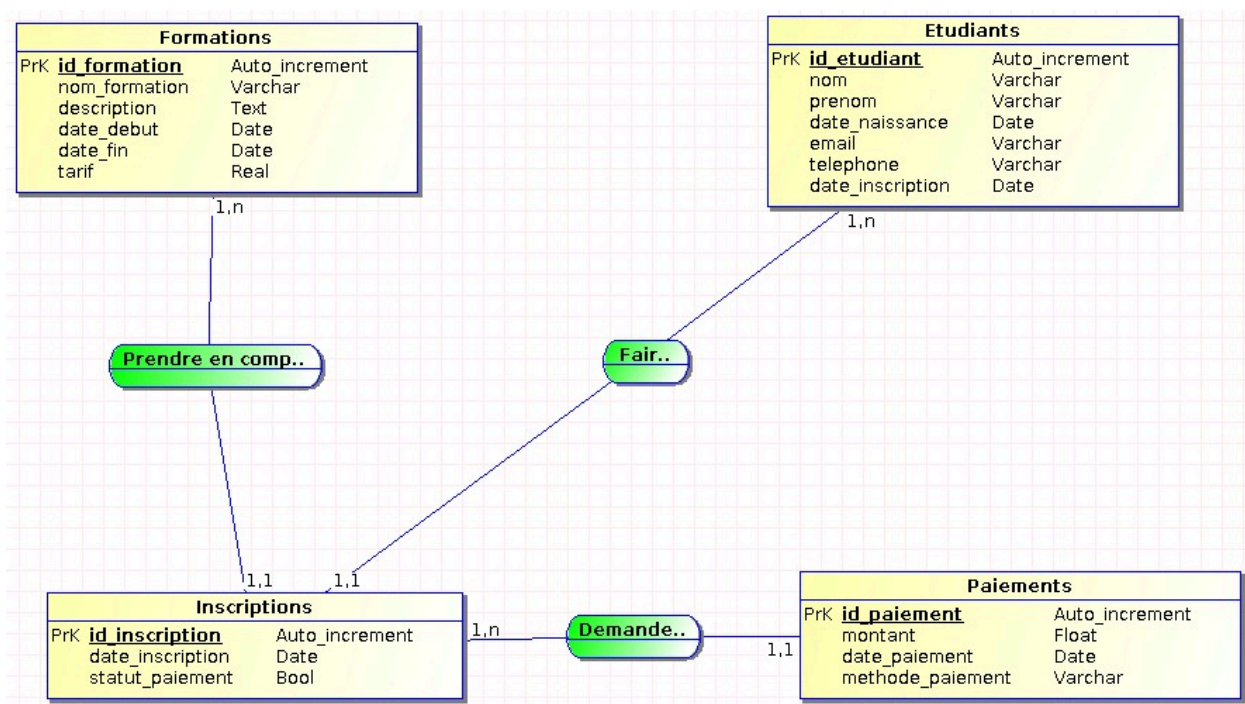
Les triggers (ou déclencheurs) sont des objets de la base de données. Attachés à une table, ils vont déclencher l'exécution d'une instruction, ou d'un bloc d'instructions, lorsqu'une, ou plusieurs lignes sont insérées, supprimées ou modifiées dans la table à laquelle ils sont attachés. Pour notre projet nous avons créé un trigger pour **mettre à jour le statut de paiement** lors de l'ajout d'un paiement



# Résultats

Dans cette section vous trouverez pour chaque étape de notre projet des captures d'écran montrant l'ensemble de nos actions sous forme de scripts SQL et le rendu lors de l'implémentation dans MySQL

## 1. Choix des entités et définition des associations



## 2. Création de la base de données et des tables

```
drop database if exists ecole;
create database ecole CHARACTER SET 'utf8';

use ecole;

DROP TABLE IF EXISTS Formations;
CREATE TABLE Formations(
    id_formation      int (25) Auto_increment not null,
    nom_formation     Varchar (50) not null,
    description        Text (2000),
    date_debut         Date not null,
    date_fin           Date not null,
    tarif              Real unsigned not null,
    nombre_places      int (50),
    PRIMARY KEY (id_formation)
)ENGINE=InnoDB;

DROP TABLE IF EXISTS Etudiants;
CREATE TABLE Etudiants(
    id_etudiant       int (25) Auto_increment not null,
    nom                Varchar (50) not null,
    prenom             Varchar (50) not null,
    date_naissance     Date not null,
    email              Varchar (50) not null,
    telephone          Varchar (50) not null,
    date_inscription   Date not null,
    PRIMARY KEY (id_etudiant)
)ENGINE=InnoDB;

DROP TABLE IF EXISTS Paiements;
CREATE TABLE Paiements(
    id_paiement        int (50) Auto_increment not null,
    montant            Float (50) not null,
    date_paiement       Date not null,
    methode_paiement    Varchar (50) not null,
    id_inscription      int not null,
    PRIMARY KEY (id_paiement)
)ENGINE=InnoDB;

DROP TABLE IF EXISTS Inscriptions;
CREATE TABLE Inscriptions(
    id_inscription     int (25) auto_increment not null,
    date_inscription    Date not null,
    statut_paiement     Bool default false not null,
    id_etudiant_Etudiants int not null,
    id_formation_Formations int not null,
    PRIMARY KEY (id_inscription)
)ENGINE=InnoDB;

ALTER TABLE Paiements ADD CONSTRAINT FK_Paiements_id_inscription_Inscriptions FOREIGN KEY
(id_inscription_Inscriptions) REFERENCES Inscriptions(id_inscription);
ALTER TABLE Inscriptions ADD CONSTRAINT FK_Inscriptions_id_etudiant_Etudiants FOREIGN KEY
(id_etudiant_Etudiants) REFERENCES Etudiants(id_etudiant);
ALTER TABLE Inscriptions ADD CONSTRAINT FK_Inscriptions_id_formation_Formations FOREIGN
KEY (id_formation_Formations) REFERENCES Formations(id_formation);
```

## → Affichage des tables dans MySQL

Tables_in_ecole
Etudiants
Formations
Inscriptions
Paielements

Field	Type	Null	Key	Default	Extra
id_formation	int	NO	PRI	NULL	auto_increment
nom_formation	varchar(50)	NO		NULL	
description	text	YES		NULL	
date_debut	date	NO		NULL	
date_fin	date	NO		NULL	
tarif	double unsigned	NO		NULL	
nombre_places	int	YES		NULL	

Field	Type	Null	Key	Default	Extra
id_etudiant	int	NO	PRI	NULL	auto_increment
nom	varchar(50)	NO		NULL	
prenom	varchar(50)	NO		NULL	
date_naissance	date	NO		NULL	
email	varchar(50)	NO		NULL	
telephone	varchar(50)	NO		NULL	
date_inscription	date	NO		NULL	

Field	Type	Null	Key	Default	Extra
id_inscription	int	NO	PRI	NULL	auto_increment
date_inscription	date	NO		NULL	
statut_paiement	tinyint(1)	NO		0	
id_etudiant_Etudiants	int	NO	MUL	NULL	
id_formation_Formations	int	NO	MUL	NULL	

Field	Type	Null	Key	Default	Extra
id_paiement	int	NO	PRI	NULL	auto_increment
montant	double	NO		NULL	
date_paiement	date	NO		NULL	
methode_paiement	varchar(50)	NO		NULL	
id_inscription_Inscriptions	int	NO	MUL	NULL	

### 3. Ajout des données

```
INSERT INTO Formations (nom_formation, description, date_debut, date_fin, tarif,
nombre_places)
VALUES
('Developpement web', 'Apprenez à créer des sites web dynamiques avec HTML CSS et
JavaScript.', '2024-09-01', '2024-12-15', 250.000, 10),
('Infographie', 'Maîtrisez les logiciels Adobe Photoshop et Illustrator pour créer des
visuels professionnels.', '2024-10-05', '2024-12-22', 250.000, 15),
('Réseaux informatiques', 'Configurez et sécurisez des réseaux informatiques pour les
entreprises.', '2024-11-10', '2025-01-26', 250.000, 25),
('Comptabilité', 'Apprenez les bases de la comptabilité et de la gestion financière.',
'2024-12-01', '2025-02-23', 250.000, 30),
('Anglais des affaires', 'Perfectionnez votre anglais pour des communications
professionnelles efficaces.', '2025-01-12', '2025-03-30', 250.000, 25),
('Marketing digital', 'Maîtrisez les techniques du marketing digital pour promouvoir
votre entreprise en ligne.', '2025-02-02', '2025-04-27', 250.000, 20);
```

```
INSERT INTO Etudiants (nom, prenom, telephone, email, date_inscription, date_naissance)
VALUES
('Foucher', 'Gabriel', '+33 3 86 40 24 65', 'zriou@example.com', '2023-01-22',
'1997-06-11'),
('Ferrand', 'Olivier', '+33 (0)3 90 32 47 67', 'vincent69@example.org', '2024-04-12',
'1997-02-17'),
('Dias', 'Michèle', '+33 6 22 12 84 98', 'leroypatricia@example.org', '2023-06-05',
'1994-04-18'),
('Dupuis', 'Jean', '+33 5 82 53 65 62', 'didiercaroline@example.net', '2024-02-11',
'1995-10-02'),
('Laporte', 'Jean', '+33 (0)4 42 53 40 03', 'celina68@example.net', '2024-01-10',
'1998-03-24'),
('Coste', 'Marie', '02 34 83 04 66', 'nathalie56@example.org', '2023-02-09',
'1997-05-10'),
('Lucas', 'Louis', '+33 (0)1 64 68 65 38', 'smeyer@example.com', '2023-02-14',
'1993-12-01'),
('Bonneau', 'Gilles', '+33 (0)4 90 10 86 05', 'martine33@example.org', '2023-03-14',
'1992-05-31'),
('Loiseau', 'Charles', '+33 (0)4 92 86 07 28', 'xavierpineau@example.com', '2023-01-14',
'1994-09-01');
```

```
INSERT INTO Inscriptions (date_inscription, statut_paieement, id_etudiant_Etudiants,
id_formation_Formations)
VALUES
('2023-01-27', 0, 1, 1),
('2024-05-22', 0, 2, 2),
('2023-07-06', 0, 3, 2),
('2024-03-29', 1, 4, 3),
('2024-02-17', 0, 5, 4),
('2023-03-01', 0, 6, 5),
('2023-03-07', 0, 7, 6),
('2023-04-06', 0, 8, 5),
('2023-02-01', 0, 9, 3);
```

```
INSERT INTO Paiements (montant, date_paiement, methode_paiement,
id_inscription_Inscriptions)
VALUES
(100.000, '2023-07-21', 'Carte bancaire', 1),
(90.000, '2023-06-15', 'Chèque', 2),
(120.000, '2023-12-25', 'Carte bancaire', 3),
(250.000, '2023-10-10', 'Chèque', 4),
(70.000, '2024-05-14', 'Orange-money', 5),
(50.000, '2024-08-30', 'Carte bancaire', 6),
(200.000, '2024-03-03', 'Chèque', 7),
(30.000, '2023-09-11', 'Orange-money', 8),
(20.000, '2023-11-01', 'Carte bancaire', 9),
(10.000, '2023-09-15', 'Chèque', 2),
(60.000, '2024-03-01', 'Carte bancaire', 1),
(15.000, '2024-02-26', 'Orange-money', 3),
(76.000, '2024-01-01', 'Chèque', 8),
(42.000, '2024-09-11', 'Orange-money', 6),
(20.000, '2024-04-07', 'Carte bancaire', 9),
(30.000, '2024-06-17', 'Chèque', 5),
(63.000, '2024-05-01', 'Carte bancaire', 7);
```

## → Affichage des tables remplies dans MySQL

```
mysql> SELECT * FROM Formations;
```

id_formation	nom_formation	description	date_debut	date_fin	tarif	nombre_places
1	Développement web	Apprenez à créer des sites web dynamiques avec HTML CSS et JavaScript.	2024-09-01	2024-12-15	250	10
2	Infographie	Maîtrisez les logiciels Adobe Photoshop et Illustrator pour créer des visuels professionnels.	2024-10-05	2024-12-22	250	15
3	Réseaux informatiques	Configurez et sécurisez des réseaux informatiques pour les entreprises.	2024-11-10	2025-01-26	250	25
4	Comptabilité	Apprenez les bases de la comptabilité et de la gestion financière.	2024-12-01	2025-02-23	250	30
5	Anglais des affaires	Perfectionnez votre anglais pour des communications professionnelles efficaces.	2025-01-12	2025-03-30	250	25
6	Marketing digital	Maîtrisez les techniques du marketing digital pour promouvoir votre entreprise en ligne.	2025-02-02	2025-04-27	250	20
7	Développement web	Apprenez à créer des sites web dynamiques avec HTML CSS et JavaScript.	2024-09-01	2024-12-15	250	10
8	Infographie	Maîtrisez les logiciels Adobe Photoshop et Illustrator pour créer des visuels professionnels.	2024-10-05	2024-12-22	250	15
9	Réseaux informatiques	Configurez et sécurisez des réseaux informatiques pour les entreprises.	2024-11-10	2025-01-26	250	25
10	Comptabilité	Apprenez les bases de la comptabilité et de la gestion financière.	2024-12-01	2025-02-23	250	30
11	Anglais des affaires	Perfectionnez votre anglais pour des communications professionnelles efficaces.	2025-01-12	2025-03-30	250	25
12	Marketing digital	Maîtrisez les techniques du marketing digital pour promouvoir votre entreprise en ligne.	2025-02-02	2025-04-27	250	20

12 rows in set (0,00 sec)

id_etudiant	nom	prenom	date_naissance	email	telephone	date_inscription
1	Foucher	Gabriel	1997-06-11	zriou@example.com	+33 3 86 40 24 65	2023-01-27
2	Ferrand	Olivier	1997-02-17	vincent69@example.org	+33 (0)3 90 32 47 67	2024-05-22
3	Dias	Michèle	1994-04-18	leroypatricia@example.org	+33 6 22 12 84 98	2023-07-06
4	Dupuis	Jean	1995-10-02	didiercaroline@example.net	+33 5 82 53 65 62	2024-03-29
5	Laporte	Jean	1998-03-24	celina68@example.net	+33 (0)4 42 53 40 03	2024-02-17
6	Coste	Marie	1997-05-10	nathalie56@example.org	02 34 83 04 66	2023-03-01
7	Lucas	Louis	1993-12-01	smeyer@example.com	+33 (0)1 64 68 65 38	2023-03-07
8	Bonneau	Gilles	1992-05-31	martine33@example.org	+33 (0)4 90 10 86 05	2023-04-06
9	Loiseau	Charles	1994-09-01	xavierpineau@example.com	+33 (0)4 92 86 07 28	2023-02-01
10	Foucher	Gabriel	1997-06-11	zriou@example.com	+33 3 86 40 24 65	2023-01-27
11	Ferrand	Olivier	1997-02-17	vincent69@example.org	+33 (0)3 90 32 47 67	2024-05-22
12	Dias	Michèle	1994-04-18	leroypatricia@example.org	+33 6 22 12 84 98	2023-07-06
13	Dupuis	Jean	1995-10-02	didiercaroline@example.net	+33 5 82 53 65 62	2024-03-29
14	Laporte	Jean	1998-03-24	celina68@example.net	+33 (0)4 42 53 40 03	2024-02-17
15	Coste	Marie	1997-05-10	nathalie56@example.org	02 34 83 04 66	2023-03-01
16	Lucas	Louis	1993-12-01	smeyer@example.com	+33 (0)1 64 68 65 38	2023-03-07
17	Bonneau	Gilles	1992-05-31	martine33@example.org	+33 (0)4 90 10 86 05	2023-04-06
18	Loiseau	Charles	1994-09-01	xavierpineau@example.com	+33 (0)4 92 86 07 28	2023-02-01

```
mysql> SELECT * FROM Inscriptions;
```

id_inscription	date_inscription	statut_paiement	id_etudiant_Etudiants	id_formation_Formations
1	2023-01-27	0	1	1
2	2024-05-22	0	2	2
3	2023-07-06	0	3	2
4	2024-03-29	1	4	3
5	2024-02-17	0	5	4
6	2023-03-01	0	6	5
7	2023-03-07	0	7	6
8	2023-04-06	0	8	5
9	2023-02-01	0	9	3

9 rows in set (0,00 sec)

```
mysql> SELECT * FROM Paiements;
```

id_paiement	montant	date_paiement	methode_paiement	id_inscription_Inscriptions
18	100	2023-07-21	Carte bancaire	1
19	90	2023-06-15	Chèque	2
20	120	2023-12-25	Carte bancaire	3
21	250	2023-10-10	Chèque	4
22	70	2024-05-14	Orange-money	5
23	50	2024-08-30	Carte bancaire	6
24	200	2024-03-03	Chèque	7
25	30	2023-09-11	Orange-money	8
26	20	2023-11-01	Carte bancaire	9
27	10	2023-09-15	Chèque	2
28	60	2024-03-01	Carte bancaire	1
29	15	2024-02-26	Orange-money	3
30	76	2024-01-01	Chèque	8
31	42	2024-09-11	Orange-money	6
32	20	2024-04-07	Carte bancaire	9
33	30	2024-06-17	Chèque	5
34	63	2024-05-01	Carte bancaire	7

```
17 rows in set (0,00 sec)
```

---

## 4. Création des requêtes

```
#Requete pour lister tous les étudiants inscrits à une certaine formation
select"-- Requete pour lister tous les étudiants inscrits à une certaine formation--";
select e.nom, e.prenom, e.id_etudiant
from Etudiants as e
join Inscriptions as i
  on i.id_etudiant_Etudiants = e.id_etudiant
join Formations as f
  on i.id_formation_Formations = f.id_formation
where f.nom_formation = "Infographie";
```

```
#Requête pour trouver les paiements effectués par un certain étudiant
select"-- Requete pour trouver les paiements effectués par un certain étudiant--";
select p.id_paiement, p.montant, p.methode_paiement, p.date_paiement
from Paiements as p
Join Inscriptions as i
  on p.id_inscription_Inscriptions = i.id_inscription
Join Etudiants as e
  on e.id_etudiant = i.id_etudiant_Etudiants
where e.nom = "Foucher";
```

```
#Requête pour lister les inscriptions en cours dun étudiant
select"--Requete pour lister les inscriptions en cours dun étudiant--";
select i.id_inscription, i.date_inscription, i.statut_paiement
from Inscriptions as i
Join Etudiants as e
  On i.id_etudiant_Etudiants = e.id_etudiant
where e.nom = "Foucher" and i.statut_paiement = 0 ;
```

```
#Requête pour compter le nombre total dinscriptions par formation
select"--Requete pour compter le nombre total dinscriptions par formation--";
select f.id_formation, f.nom_formation, COUNT(f.id_formation) as Total_inscriptions
from Formations as f
Join Inscriptions as i
  On i.id_formation_Formations = f.id_formation
Where f.nom_formation = "Infographie"
GROUP BY f.id_formation;
```



## → Affichage des requêtes dans MySql

```
+-----+
| -- Requete pour lister tous les étudiants inscrits à une certaine formation-- |
+-----+
| -- Requete pour lister tous les étudiants inscrits à une certaine formation-- |
+-----+
1 row in set (0,00 sec)
```

```
+-----+-----+-----+-----+
| nom      | prenom    | id_etudiant | nom_formation |
+-----+-----+-----+-----+
| Ferrand  | Olivier   | 2           | Infographie    |
| Dias     | Michèle   | 3           | Infographie    |
+-----+-----+-----+-----+
2 rows in set (0,00 sec)
```

```
+-----+
| -- Requete pour trouver les paiements effectués par un certain étudiant-- |
+-----+
| -- Requete pour trouver les paiements effectués par un certain étudiant-- |
+-----+
1 row in set (0,00 sec)
```

```
+-----+-----+-----+-----+-----+-----+
| id_paiement | montant | methode_paiement | date_paiement | nom      | prenom |
+-----+-----+-----+-----+-----+-----+
| 1           | 100     | Carte bancaire   | 2023-07-21    | Foucher | Gabriel |
| 11          | 60      | Carte bancaire   | 2024-03-01    | Foucher | Gabriel |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0,01 sec)
```

```
+-----+
| --Requete pour lister les inscriptions en cours dun étudiant-- |
+-----+
| --Requete pour lister les inscriptions en cours dun étudiant-- |
+-----+
1 row in set (0,00 sec)
```

```
+-----+-----+-----+-----+-----+-----+
| id_inscription | date_inscription | statut_paiement | nom_formation | nom      | prenom |
+-----+-----+-----+-----+-----+-----+
| 1              | 2023-01-27       | 0               | Développement web | Foucher | Gabriel |
+-----+-----+-----+-----+-----+-----+
1 row in set (0,00 sec)
```



```
+-----+
| --Requete pour compter le nombre total dinscriptions par formation-- |
+-----+
| --Requete pour compter le nombre total dinscriptions par formation-- |
+-----+
1 row in set (0,00 sec)

+-----+-----+-----+
| id_formation | nom_formation | Total_inscriptions |
+-----+-----+-----+
|          2 | Infographie   |          2 |
+-----+-----+-----+
1 row in set (0,00 sec)
```

---

## 5. Définition des fonctions et procédures de notre système

```
drop procedure if exists newinscription;
DELIMITER //

CREATE PROCEDURE newinscription (
    IN etudiant_id INT,
    IN formation_id INT
)
BEGIN
    INSERT INTO Inscriptions ( id_etudiant_Etudiants, id_formation_Formations,
    date_inscription, statut_paie ment)
    VALUES ( etudiant_id, formation_id, NOW(), FALSE);
END //

DELIMITER ;

drop function if exists formation_available;

DELIMITER //

CREATE FUNCTION formation_available (id_formation_Formation INT)
RETURNS bool
DETERMINISTIC
BEGIN
    DECLARE nombre_total INT;
    DECLARE nombre_inscriptions INT;
    DECLARE places_restantes bool;

    SELECT nombre_places INTO nombre_total
    FROM Formations
    WHERE id_formation = id_formation_Formation;

    SELECT COUNT(*) INTO nombre_inscriptions
    FROM Inscriptions
    WHERE id_formation_Formations = id_formation_Formation;

    SET places_restantes = nombre_total - nombre_inscriptions;

    if (places_restantes <= 0) then
        set places_restantes = false;
    else
        set places_restantes = true;
    end if ;
    RETURN places_restantes;

END //

DELIMITER ;
```

---

## → Affichage de la fonction

```
mysql> SELECT formation_available(3);
+-----+
| formation_available(3) |
+-----+
|                      1 |
+-----+
1 row in set (0,00 sec)
```

## 6. Mise en place des vues et triggers

```
#-----
# 5. view and trigger to update payment status
#-----

# 1. view to display registration details
#-----
#SELECT *****VIEW AND TRIGGER*****\n";

#SELECT *****View to display registration details*****\n";

drop view if exists inscriptionDetails;
create view inscriptionDetails as
select nom, nom_formation, Inscriptions.date_inscription from Inscriptions inner join
Etudiants on id_etudiant = id_etudiant Etudiants inner join
Formations on id_formation = id_formation Formations;

# calling the view
select * from inscriptionDetails order by date_inscription;

#-----
# 2. trigger update paymentStatus when a payment is done
#-----
#SELECT *****Trigger to update paymentStatus*****\n";

drop procedure if exists countMoney;
delimiter //

create procedure countMoney(in id int, out total float)
begin
    declare total float;
    select SUM(montant) into total from Inscriptions inner join
    Paiements on id_inscription = id_inscription Inscriptions
    where id_inscription_Inscriptions = id;
end //
delimiter ;
# select countMoney(1);

drop trigger if exists paymentStatus;
delimiter //
create trigger paymentStatus
after insert on Paiements
for each row
begin
    declare total float;
    declare price real;
    call countMoney(new.id_inscription_Inscriptions, total);

    select tarif into price from Formations inner join
    Inscriptions on id_formation = id_formation Formations
    where id_inscription = new.id_inscription_Inscriptions;

    if total > price then
        update Inscriptions set statut_paiement = false
        where id_inscription = new.id_inscription_Inscriptions;
    else
        update Inscriptions set statut_paiement = true
        where id_inscription = new.id_inscription_Inscriptions;
    end if;
end //
delimiter ;
```

## → Affichage des vues et triggers dans MySql

```
+-----+
| *****VIEW AND TRIGGER*****
|
+-----+
| *****VIEW AND TRIGGER*****
|
+-----+
1 row in set (0,00 sec)

Query OK, 0 rows affected, 1 warning (0,03 sec)

Query OK, 0 rows affected (0,11 sec)

+-----+-----+-----+
| nom      | nom_formation      | date_inscription |
+-----+-----+-----+
| Foucher  | Développement web  | 2023-01-27       |
| Loiseau  | Réseaux informatiques | 2023-02-01       |
| Coste    | Anglais des affaires | 2023-03-01       |
| Lucas    | Marketing digital   | 2023-03-07       |
| Bonneau  | Anglais des affaires | 2023-04-06       |
| Dias     | Infographie         | 2023-07-06       |
| Laporte  | Comptabilité        | 2024-02-17       |
| Dupuis   | Réseaux informatiques | 2024-03-29       |
| Ferrand  | Infographie         | 2024-05-22       |
+-----+-----+-----+
9 rows in set (0,00 sec)

Query OK, 0 rows affected, 1 warning (0,02 sec)

Query OK, 0 rows affected (0,08 sec)

Query OK, 0 rows affected, 1 warning (0,05 sec)

Query OK, 0 rows affected (0,18 sec)
```

---

# CONCLUSION

À travers notre base de données des inscriptions en ligne que nous avons créé, nous avons simplifié la collecte de données et amélioré l'efficacité de la gestion des inscriptions. Il n'est plus nécessaire de distribuer des formulaires papier à compléter et retourner.

Côté étudiants, tout est plus simple car ils peuvent s'inscrire quand et d'où ils veulent, depuis n'importe quel ordinateur ou téléphone connecté à Internet. Les informations fournies par les étudiants sont immédiatement stockées dans notre base de données.

Ce projet allège le travail des agents de la scolarité et augmente la visibilité de l'école.