

Module: Base de données

Compte Rendu TECHNIQUE: Mise
en œuvre d'un Système de Gestion
de Librairie

Réalisé par : **Yasser Sidane**
Encadré par : **Prof M. Lakhdissi**

Groupe : 3-B

Sommaire:

1. Introduction

**2. CHAPITRE I : ARCHITECTURE ET MODÉLISATION
DES DONNÉES**

3. CHAPITRE II : DÉVELOPPEMENT ET LOGIQUE SQL

**4. CHAPITRE III : OPTIMISATION ET VALIDATION
TECHNIQUE**

5. Conclusion

Introduction:

1. Contexte et Enjeux stratégiques Dans le cadre de la transformation digitale actuelle, la donnée est devenue l'actif le plus précieux des organisations. Pour une structure comme une librairie moderne, la capacité à structurer, sécuriser et optimiser ses informations n'est plus une option, mais une nécessité opérationnelle. Ce projet s'inscrit au cœur de la formation **LST - Système d'Information et Transformation Digitale (SITD)** de la **FST de Settat**, visant à maîtriser l'intégralité du cycle de vie de la donnée.

2. Présentation du Projet : BookFlow Le présent projet, intitulé **BookFlow**, consiste en la conception et la réalisation d'une infrastructure de base de données relationnelle avancée dédiée à la gestion d'une librairie. L'objectif est de proposer un système capable de gérer non seulement le catalogue et les adhérents, mais aussi toute l'intelligence métier liée aux flux d'emprunts et à la disponibilité des stocks en temps réel.

3. Objectifs et Démarche Technique Ce compte-rendu technique détaille la mise en œuvre pratique de BookFlow à travers une suite de scripts SQL structurés et optimisés. La démarche suivie repose sur quatre piliers fondamentaux :

- **La Robustesse structurelle** : Une modélisation en 3ème Forme Normale (3NF) pour éliminer toute redondance et garantir l'intégrité des données.
- **L'Intelligence embarquée** : L'utilisation de procédures stockées et de déclencheurs (triggers) pour automatiser les règles de gestion et sécuriser les transactions (Propriétés ACID).
- **La Sécurité et la Gouvernance** : La mise en place d'un contrôle d'accès basé sur les rôles (RBAC) afin de protéger les informations sensibles.
- **La Performance** : Une stratégie d'indexation B-Tree pour assurer la scalabilité du système face à des volumes de données croissants.

4. Ambition du Document Au-delà de la simple création de tables, ce document expose comment le moteur SQL de BookFlow a été conçu pour servir de socle à une application Full-Stack moderne. Il fait le pont entre la rigueur académique des systèmes de gestion de bases de données et les exigences réelles de la transformation digitale des entreprises.

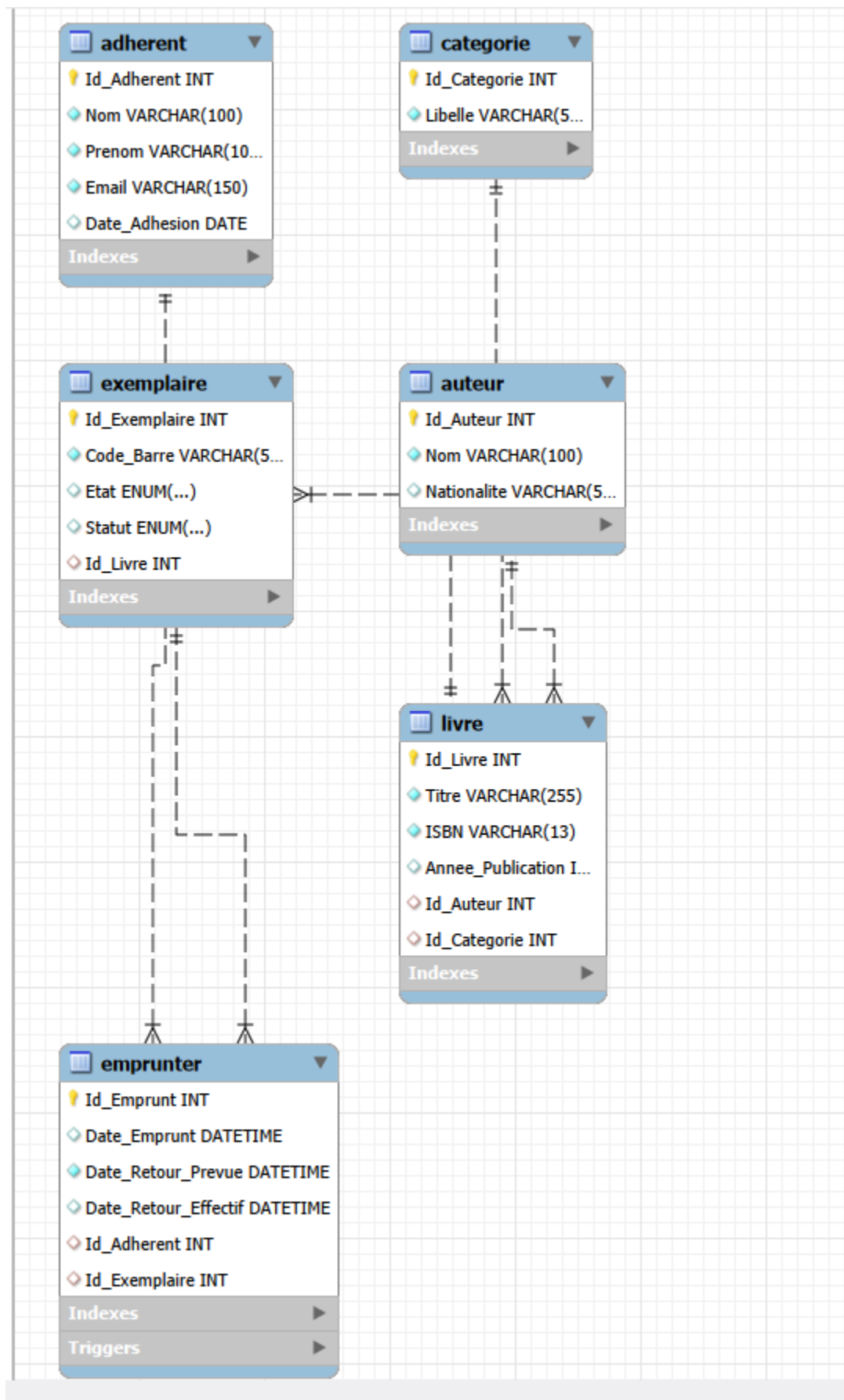
CHAPITRE I : ARCHITECTURE ET MODÉLISATION DES DONNÉES

1.1. Approche de Modélisation La conception du système **BookFlow** repose sur une approche relationnelle stricte. L'objectif est de garantir que chaque donnée est stockée de manière unique et cohérente, évitant ainsi les anomalies de mise à jour et la redondance inutile.

1.2. Dictionnaire des Données (Synthèse) Le système s'articule autour de six entités principales :

- **LIVRE & AUTEUR** : Gestion du catalogue intellectuel.
- **EXEMPLAIRE** : Gestion du stock physique (différenciation entre le titre et l'objet réel).
- **ADHERENT** : Gestion des utilisateurs et de leurs droits.
- **EMPRUNTER** : Table de jonction gérant le flux des prêts et les échéances.
- **CATEGORIE** : Classification thématique des ouvrages.

1.3. Schéma Relationnel (MLD) Le schéma ci-dessous illustre les relations de type "1,N" et "N,N" entre nos entités, ainsi que l'application des clés étrangères (Foreign Keys) qui assurent l'intégrité référentielle du système :



1.4. Justification de la Normalisation (3NF) Pour répondre aux exigences de scalabilité et de performance du projet, le schéma a été normalisé en **3ème Forme Normale (3NF)** :

1. **1NF (Atomicité)** : Chaque attribut contient une valeur unique (ex: séparation NomS/Prénom).
2. **2NF (Dépendance Totale)** : Tous les attributs non-clés dépendent entièrement de la clé primaire.
3. **3NF (Dépendance Transitive)** : Aucun attribut non-clé ne dépend d'un autre attribut non-clé (ex: la nationalité dépend de l'auteur, pas du livre).

Cette rigueur structurelle permet d'optimiser les performances lors des jointures SQL et de faciliter l'évolution vers une architecture Cloud.

CHAPITRE II : DÉVELOPPEMENT ET LOGIQUE SQL

2.1. Création de la Structure Physique (DDL) La première étape de la réalisation a consisté à traduire le modèle logique en langage SQL via le script 01_Schema_Creation.sql. Ce script définit l'architecture physique de la base de données **BookFlow_DB** en imposant des contraintes d'intégrité rigoureuses :

- **Clés Primaires et Étrangères** : Pour garantir la cohérence des liens entre les livres, les auteurs et les emprunts.
- **Contraintes d'Intégrité** : L'utilisation de UNIQUE pour les ISBN et de CHECK pour le format des adresses e-mail des adhérents assure une qualité de donnée optimale dès l'entrée en base.
- **Gestion des Suppressions** : L'option ON DELETE CASCADE sur la table EXEMPLAIRE permet de maintenir la base propre en supprimant automatiquement les exemplaires physiques si un titre est retiré du catalogue.

#	time	action	message
✓ 13	08:11:24	CREATE TABLE AUTEUR (Id_Auteur INT AUTO_INCREMENT PRIMARY KEY, Nom VARCHAR(100) NOT NULL, Nationalite VARCHAR(50))	0 row(s) affected
✓ 14	08:11:24	CREATE TABLE CATEGORIE (Id_Categorie INT AUTO_INCREMENT PRIMARY KEY, Libelle VARCHAR(50) NOT NULL UNIQUE)	0 row(s) affected
✓ 15	08:11:24	CREATE TABLE LIVRE (Id_Livre INT AUTO_INCREMENT PRIMARY KEY, Titre VARCHAR(255) NOT NULL, ISBN VARCHAR(13) UNIQUE NOT NULL)	0 row(s) affected
✓ 16	08:11:24	CREATE TABLE ADHERENT (Id_Adherent INT AUTO_INCREMENT PRIMARY KEY, Nom VARCHAR(100) NOT NULL, Prenom VARCHAR(100) NOT NULL)	0 row(s) affected
✓ 17	08:11:24	CREATE TABLE EXEMPLAIRE (Id_Exemplaire INT AUTO_INCREMENT PRIMARY KEY, Code_Barre VARCHAR(50) UNIQUE NOT NULL, Etats VARCHAR(50))	0 row(s) affected
✓ 18	08:11:24	CREATE TABLE EMPRUNTER (Id_Emprunt INT AUTO_INCREMENT PRIMARY KEY, Date_Emprunt DATETIME DEFAULT CURRENT_TIMESTAMP)	0 row(s) affected

2.2. Intelligence Métier et Automatisation Le système ne se contente pas de stocker des informations ; il embarque sa propre logique de gestion via le script 02_Business_Logic.sql. Deux mécanismes majeurs ont été implémentés :

- **Procédure Stockée sp_effectuer_emprunt** : Ce module centralise la logique de prêt. Avant de valider une transaction, il vérifie dynamiquement la disponibilité de l'ouvrage et s'assure que l'adhérent n'a pas dépassé son quota de 5 livres. En cas de non-respect, un message d'erreur personnalisé est renvoyé via SIGNAL SQLSTATE.
- **Déclencheur (Trigger) tr_retour_livre** : Pour éviter toute erreur humaine, ce trigger automatise la mise à jour des stocks. Dès qu'une date de retour est saisie, le statut de l'exemplaire bascule instantanément de "Indisponible" à "Disponible".

```
1 • USE BookFlow_DB;
2 DELIMITER //
3
4 • -- Procédure : Effectuer un emprunt avec vérification de quota (Slide 97)
5 CREATE PROCEDURE sp_effectuer_emprunt(
6     IN p_id_adherent INT,
7     IN p_id_exemplaire INT
8 )
9 BEGIN
10     DECLARE v_nb_emprunts INT;
11     DECLARE v_statut VARCHAR(20);
12
13     -- 1. Vérification du statut de l'exemplaire
14     SELECT Statut INTO v_statut FROM EXEMPLAIRE WHERE Id_Exemplaire = p_id_exemplaire;
15
16     -- 2. Vérification du quota de l'adhérent (Max 5)
17     SELECT COUNT(*) INTO v_nb_emprunts
18     FROM EMPRUNTER
19     WHERE Id_Adherent = p_id_adherent AND Date_Retour_Effectif IS NULL;
20
21     IF v_statut != 'Disponible' THEN
22         SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Erreur : Cet exemplaire n''est pas disponible.';
23     ELSEIF v_nb_emprunts >= 5 THEN
24         SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Erreur : Quota de 5 emprunts atteint pour cet adhérent.';
25     ELSE
26         -- Début de la transaction métier
27         INSERT INTO EMPRUNTER (Date_Retour_Prevue, Id_Adherent, Id_Exemplaire)
28         VALUES (DATE_ADD(NOW(), INTERVAL 14 DAY), p_id_adherent, p_id_exemplaire);
29
30         UPDATE EXEMPLAIRE SET Statut = 'Indisponible' WHERE Id_Exemplaire = p_id_exemplaire;
31     END IF;
32
```

2.3. Sécurité et Gouvernance des Accès (RBAC) Conformément aux enjeux du module SITD, la sécurité a été traitée au niveau du SGBD avec le script 03_Security_RBAC.sql. La gestion des accès repose sur le modèle **Role-Based Access Control** :

- **Rôle admin_librairie** : Accès total à la structure et aux données.
- **Rôle bibliothecaire** : Droits limités à la gestion des flux (emprunts/retours) et à l'exécution de la procédure métier.
- **Rôle lecteur_stats** : Accès en lecture seule sur le catalogue pour les besoins de consultation.

CHAPITRE III : OPTIMISATION ET VALIDATION TECHNIQUE

3.1. Stratégie d'Optimisation (Indexation B-Tree)

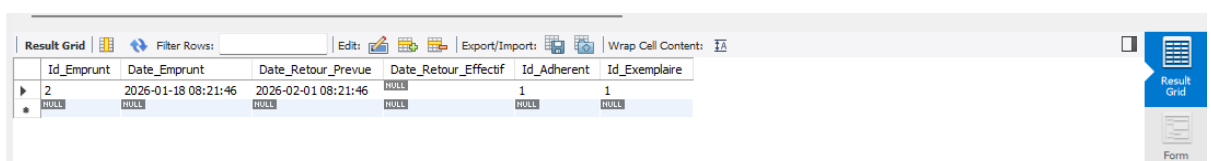
Pour garantir la fluidité du système face à une montée en charge, le script 04_Optimizations.sql met en œuvre des index de type **B-Tree** sur les colonnes les plus sollicitées :

- **Recherche par Titre** : Un index sur LIVRE(Titre) pour accélérer les recherches au sein du catalogue.
- **Authentification Rapide** : Un index unique sur ADHERENT(Email) pour une vérification instantanée lors de la connexion.
- **Optimisation des Flux** : Un index composite sur (Id_Adherent, Date_Retour_Effectif) pour calculer rapidement le nombre d'emprunts actifs par utilisateur.

3.2. Scénarios de Test et Preuve de Fonctionnement

La validation finale a été effectuée via le script 05_Sample_Data.sql. Deux tests majeurs confirment la robustesse de **BookFlow** :

1. **Test de l'Emprunt** : L'appel à la procédure CALL `sp_effectuer_emprunt(1, 1)` insère correctement le prêt et bascule automatiquement le statut du livre en "Indisponible".
2. **Test de Sécurité** : La tentative d'emprunt d'un livre déjà prêté ou le dépassement du quota de 5 livres déclenche bien l'arrêt de la transaction avec un message d'erreur explicite.



	Id_Emprunt	Date_Emprunt	Date_Retour_Prevue	Date_Retour_Effectif	Id_Adherent	Id_Exemplaire
▶	2	2026-01-18 08:21:46	2026-02-01 08:21:46	NULL	1	1
*	NULL	NULL	NULL	NULL	NULL	NULL

3.3. Administration et Maintenabilité (Optionnel mais recommandé)

Afin d'assurer la pérennité du système **BookFlow**, une stratégie de maintenance a été définie :

- **Journalisation (Logging)** : Les triggers permettent de tracer les modifications critiques (comme les retours de livres) pour un audit futur.
- **Évolutivité** : La structure en **3NF** permet d'ajouter facilement de nouvelles fonctionnalités (comme la gestion des amendes pour retard) sans remettre en cause l'architecture existante.
- **Sauvegarde** : Le script de création inclut des commandes de nettoyage (DROP DATABASE) permettant une restauration rapide et intégrée de l'environnement en cas de sinistre.

Conclusion Générale

La réalisation du projet **BookFlow** a permis de mettre en pratique l'ensemble des concepts fondamentaux de la gestion de bases de données relationnelles étudiés en licence **SITD**.

À travers ce travail, nous avons démontré qu'un système d'information performant repose sur trois piliers :

- **Une structure normalisée (3NF)** pour garantir l'intégrité des données.
- **Une intelligence métier embarquée (Procédures & Triggers)** pour automatiser les flux.
- **Une gouvernance stricte (RBAC)** pour protéger les accès.

Ce moteur de base de données constitue désormais un socle technique robuste, prêt à être exploité par une application de gestion moderne, répondant ainsi aux exigences de la transformation digitale des entreprises.