



## **Rapport d'analyse et de validation du projet de gestion de bibliothèque**

### **Réalisé par:**

- ❖ Sabrina Lamghari
- ❖ Cyril Rouillon
- ❖ Hamid Amediaz



Le déroulement du projet de base de données pour la bibliothèque a suivi une organisation structurée en plusieurs phases : conception, implémentation, validation et documentation.

La répartition des tâches s'est effectuée comme suit:

- **Sabrina:** La gestion des emprunts, le suivi des prolongations et l'analyse des comportements des abonnés et de la popularité des ouvrages.
- **Hamid:** L'optimisation des emprunts, la gestion des réservations et l'intégration des nouvelles bibliothèques au réseau.
- **Cyril:** La gestion des rôles et des privilèges d'accès, l'optimisation des transferts d'ouvrages pour minimiser les coûts logistiques.

Pour regarder ce que chaque membre du groupe a fait, veuillez le trouver sur GitHub dans des fichiers séparés.

Malgré cette planification, nous avons rencontré quelques difficultés, notamment en matière de communication et d'organisation, en raison des disponibilités variables des membres de l'équipe. Pour pallier cela, nous avons adopté des outils collaboratifs comme Github pour le suivi des tâches et Discord pour les discussions asynchrones. La complexité de certaines requêtes SQL a également posé des défis, nécessitant une décomposition en sous-étapes et une consultation approfondie de la documentation.

De plus, la gestion des contraintes d'intégrité et l'optimisation des performances ont exigé plusieurs ajustements, que nous avons résolus en procédant à des tests progressifs sur des jeux de données simulées

Malgré ces obstacles, plusieurs éléments ont contribué à la réussite du projet. La structuration initiale du travail a facilité la progression et l'intégration des différentes parties du projet. L'utilisation d'outils collaboratifs a permis une gestion plus fluide des tâches et une communication efficace. La reformulation de la globalité du sujet a également joué un rôle clé en facilitant les ajustements et la compréhension des différentes étapes. De plus, une validation rigoureuse et des tests continus ont permis d'identifier rapidement les erreurs et d'optimiser les performances et la structure de la base de données.

Si nous devions recommencer ce projet, nous apporterions certaines améliorations, notamment une meilleure estimation du temps nécessaire pour chaque tâche, en accordant une attention particulière aux phases de test et d'optimisation des performances. Il serait également pertinent de définir des points de contrôle intermédiaires plus précis afin d'éviter l'accumulation de tâches en fin de projet. L'intégration d'analyses de performances et d'indexation dès les premières étapes permettrait également d'améliorer l'efficacité globale du système. Enfin, un suivi plus régulier, avec des réunions hebdomadaires, comme cela a été fait, aiderait à mieux répartir la charge de travail et à anticiper les difficultés. Dans l'ensemble, ce projet a été une expérience enrichissante qui a permis d'acquérir des compétences en gestion de projet et en

optimisation de bases de données, tout en mettant en lumière les axes d'amélioration pour une organisation plus efficace à l'avenir.

## Jeu d'instruction:

- Trouver les ouvrages disponibles dans des bibliothèques autres que celle où l'abonné est inscrit et qui n'ont jamais été empruntés par cet abonné.

Query Query History

```
1 SELECT o.id AS ouvrage_id, o.titre, e.id AS exemplaire_id, b.nom AS bibliotheque_actuelle
2 FROM Exemplaires e
3 JOIN Ouvrages o ON e.ouvrage_id = o.id
4 JOIN Bibliothèques b ON e.bibliotheque_id = b.id
5 WHERE e.disponible = TRUE
6 AND e.bibliotheque_id != (
7     SELECT bibliotheque_id
8     FROM S_abonne
9     WHERE abonne_id = 2
10 )
11 AND e.ouvrage_id NOT IN (
12     SELECT exemplaire_id
13     FROM Loue
14     WHERE abonne_id = 2
15 );
```

Data Output Messages Notifications

SQL

Showing 1

	ouvrage_id integer	titre character varying (255)	exemplaire_id integer	bibliotheque_actuelle character varying (255)
1	1	Encyclopédie Universelle	1	Bibliothèque Centrale
2	2	Le Petit Prince	2	Bibliothèque Centrale
3	5	Astérix et Obélix	5	Bibliothèque Universitaire
4	6	Tintin au Tibet	6	Bibliothèque Régionale
5	7	L'Étranger	7	Médiathèque de Lille

- Empêcher qu'un livre emprunté soit prolongé plus de deux fois.

```
--empêcher qu'un livre emprunté soit prolongé plus de deux fois.
CREATE OR REPLACE FUNCTION limiter_prolongations()
RETURNS TRIGGER AS $$

BEGIN
    IF NEW.prolongations >= 2 THEN
        RAISE EXCEPTION 'Le livre ne peut être prolongé plus de 2 fois.';
    END IF;
    RETURN NEW;
END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER avant_prolongation
BEFORE UPDATE ON loue
FOR EACH ROW
EXECUTE FUNCTION limiter_prolongations();
```

## Query Query History

```
1  UPDATE loue
2  SET prolongations = 1
3  WHERE abonne_id = 3 AND exemplaire_id = 4;
4
5  Select * from loue;
```

## Data Output Messages Notifications

	abonne_id [PK] integer	exemplaire_id [PK] integer	date_emprunt date	date_retour date	prolongations integer
1	5	7	2025-02-06	2025-02-20	0
2	3	4	2025-02-05	2025-02-28	1

- Récupérer la liste des ouvrages les plus empruntés (empruntés plus de 5 fois):

Query Query History

```

1  CREATE VIEW ouvrages_populaires AS
2  SELECT Ouvrages.titre, COUNT(Loue.exemplaire_id) AS nombre_emprunts
3  FROM Loue
4  JOIN Ouvrages ON Loue.exemplaire_id = Ouvrages.id
5  GROUP BY Ouvrages.id
6  HAVING COUNT(Loue.exemplaire_id) > 5;
7
8  SELECT * FROM ouvrages_populaires;
9

```

Data Output Messages Notifications

Showing rows:

	titre character varying (255)	nombre_emprunts bigint
1	Les Misérables	6

- Afficher la liste des ouvrages ayant été transférés au moins 5 fois entre différentes bibliothèques:

Query Query History

```

1  SELECT Ouvrages.titre, COUNT(transferts.id) AS nombre_transferts
2  FROM transferts
3  JOIN Ouvrages ON transferts.exemplaire_id = Ouvrages.id
4  GROUP BY Ouvrages.id
5  HAVING COUNT(transferts.id) >= 5;

```

Data Output Messages Notifications

Showing rows:

	titre character varying (255)	nombre_transferts bigint
1	1984	7

- Afficher la liste des livres actuellement empruntés mais pas encore retournés, à condition que l'abonné ait moins de 2 prolongations:

Query		Query History			
1	▼	SELECT Ouvrages.titre, Abonnes.nom, Loue.date_emprunt, Loue.date_retour, Loue.prolongations			
2		FROM Loue			
3		JOIN Ouvrages ON Loue.exemplaire_id = Ouvrages.id			
4		JOIN Abonnes ON Loue.abonne_id = Abonnes.id			
5		WHERE Loue.date_retour IS NULL AND Loue.prolongations < 2;			

  

Data Output		Messages				Notifications			
≡+	📄	▼	📋	▼	🗑️	🗄️	⬇️	📶	SQL
Showing rows: 1 to 2									
Page No: 1									
	titre	nom	date_emprunt	date_retour	prolongations				
	character varying (255)	character varying (255)	date	date	integer				
1	Le Petit Prince	Dulon	2025-02-05	[null]	0				
2	Le Petit Prince	Lemoine	2025-02-06	[null]	0				

- Afficher tous les livres empruntés par un abonné spécifique:

Query		Query History			
1	▼	SELECT Ouvrages.titre, Bibliothèques.nom AS bibliotheque, Loue.date_emprunt, Loue.date_retour			
2		FROM Loue			
3		JOIN Ouvrages ON Loue.exemplaire_id = Ouvrages.id			
4		JOIN Exemplaires ON Loue.exemplaire_id = Exemplaires.id			
5		JOIN Bibliothèques ON Exemplaires.bibliotheque_id = Bibliothèques.id			
6		WHERE Loue.abonne_id = 1;			
7					

  

Data Output		Messages				Notifications			
≡+	📄	▼	📋	▼	🗑️	🗄️	⬇️	📶	SQL
Showing rows: 1 to 1									
Page No: 1									
	titre	bibliotheque	date_emprunt	date_retour					
	character varying (255)	character varying (255)	date	date					
1	Les Misérables	Bibliothèque Universitaire	2025-08-15	2025-09-01					

- Fonction qui calcule le nombre de jours restants avant le retour prévu d'un exemplaire emprunté:

Query Query History

```

1  CREATE OR REPLACE FUNCTION jours_restants(exemplaire_id_param INT)
2  RETURNS INT AS $$
3  DECLARE
4      date_retour DATE;
5  BEGIN
6      SELECT Loue.date_retour INTO date_retour
7      FROM Loue
8      WHERE Loue.exemplaire_id = exemplaire_id_param AND Loue.date_retour IS NOT NULL
9      ORDER BY Loue.date_retour DESC
10     LIMIT 1;
11
12     IF date_retour IS NOT NULL THEN
13         RETURN GREATEST((date_retour - CURRENT_DATE), 0); -- Retourne 0 si date de retour dépassée
14     ELSE
15         RETURN NULL;
16     END IF;
17 END;
18 $$ LANGUAGE plpgsql;
19 SELECT jours_restants(4);

```

Data Output Messages Notifications

Showing rows: 1 to 1 Page No: 1

jours_restants integer	
1	209

- Quels abonnés ne respectent pas les délais de prêt et quelle est leur fréquence d'infraction:

Query Query History

```

1  SELECT a.id AS abonne_id, a.nom, a.prenom, COUNT(*) AS infractions
2  FROM Loue l
3  JOIN Abonnes a ON l.abonne_id = a.id
4  WHERE l.date_retour IS NOT NULL
5  AND l.date_retour > l.date_emprunt + INTERVAL '30 days'
6  GROUP BY a.id, a.nom, a.prenom
7  ORDER BY infractions DESC;
8

```

Data Output Messages Notifications

	abonne_id integer	nom character varying (255)	prenom character varying (255)	infractions bigint
1	7	Farat	Paul	1

- Création de la vue : emprunts par bibliothèque

Cela retournera une liste des bibliothèques avec le nombre d'emprunts pour chaque bibliothèque.

Requête

Historique

1

2

3

4

5

6

7

8

9

CREATE VIEW emprunts\_par\_bibliotheque AS

SELECT Bibliothèques.nom AS bibliotheque, COUNT(Loue.exemplaire\_id) AS nombre\_emprunts

FROM Loue

JOIN Exemplaires ON Loue.exemplaire\_id = Exemplaires.id

JOIN Bibliothèques ON Exemplaires.bibliotheque\_id = Bibliothèques.id

GROUP BY Bibliothèques.id;

SELECT \* FROM emprunts\_par\_bibliotheque;

Données

Messages

Notifications

≡+

📄

▼

📋

▼

🗑️

📦

⬇️

📈

SQL

Affichage des lignes : 1 à 5

Page No: 1

	bibliotheque character varying (255)	nombre_emprunts bigint
1	Bibliothèque Régionale	1
2	Médiathèque Lyon	1
3	Bibliothèque Universitaire	6
4	Bibliothèque Centrale	3
5	Médiathèque de Lille	1

- Création d'une fonction et d'un déclencheur (trigger) pour mettre à jour le nombre d'emprunts des abonnés

Table Abonnes :





```

1  CREATE OR REPLACE FUNCTION mettre_a_jour_nombre_emprunts()
2  RETURNS TRIGGER AS $$
3  BEGIN
4      IF NEW.date_retour IS NULL THEN
5          UPDATE Abonnes
6          SET nombre_emprunts = nombre_emprunts + 1
7          WHERE id = NEW.abonne_id;
8      END IF;
9
10     IF OLD.date_retour IS NULL AND NEW.date_retour IS NOT NULL THEN
11         UPDATE Abonnes
12         SET nombre_emprunts = nombre_emprunts - 1
13         WHERE id = NEW.abonne_id;
14     END IF;
15
16     RETURN NEW;
17 END;
18 $$ LANGUAGE plpgsql;
19
20 CREATE TRIGGER mise_a_jour_emprunts
21 AFTER INSERT OR UPDATE ON Loue
22 FOR EACH ROW
23 EXECUTE FUNCTION mettre_a_jour_nombre_emprunts();

```

Requête Historique

```
1 SELECT * FROM Abonnes;
```

Données Messages Notifications

	prenom character varying (255)	email character varying (255)	telephone character varying (20)	adresse text	bibliotheque_id integer	nombre_emprunts integer
1	Sophie	sophie.bernard@mail.com	0634567890	7 boulevard Saint-Michel, Toulouse	3	0
2	Jean	jean.lemoine@mail.com	0645678901	10 place du Marché, Marseille	4	0
3	Emma	emma.garcia@mail.com	0656789012	5 rue Victor Hugo, Lille	5	0
4	Alex	alex.dulon@mail.com	0602345678	15 rue des Dolines, Paris	1	0
5	Paul	paul.farat@mail.com	0652771278	15 rue des Palais, Paris	1	0
6	Alice	alice.durand@mail.com	0612345678	15 rue des Fleurs, Paris	1	1
7	Paul	paul.martin@mail.com	0623456789	22 avenue des Champs, Lyon	2	0

Requête Historique

```

1  UPDATE Loue
2  SET date_retour = '2025-02-15'
3  WHERE abonne_id = 1 AND exemplaire_id = 1;
4
5  SELECT * FROM Abonnes;

```

Données Messages Notifications

Affichage des lignes : 1 à 7 Page No: 1 de 1

	prenom character varying (255)	email character varying (255)	telephone character varying (20)	adresse text	bibliotheque_id integer	nombre_emprunts integer
1	Sophie	sophie.bernard@mail.com	0634567890	7 boulevard Saint-Michel, Toulouse	3	0
2	Jean	jean.lemoine@mail.com	0645678901	10 place du Marché, Marseille	4	0
3	Emma	emma.garcia@mail.com	0656789012	5 rue Victor Hugo, Lille	5	0
4	Alex	alex.dulon@mail.com	0602345678	15 rue des Dolines, Paris	1	0
5	Paul	paul.farat@mail.com	0652771278	15 rue des Palais, Paris	1	0
6	Paul	paul.martin@mail.com	0623456789	22 avenue des Champs, Lyon	2	0
7	Alice	alice.durand@mail.com	0612345678	15 rue des Fleurs, Paris	1	0

- Création de la table Archive Loue et suppression d'anciens emprunts

Requête Historique

```

1  CREATE TABLE Archive_Loue AS
2  SELECT * FROM Loue
3  WHERE date_emprunt < CURRENT_DATE - INTERVAL '1 month';
4
5  DELETE FROM Loue
6  WHERE date_emprunt < CURRENT_DATE - INTERVAL '1 month';
7
8  SELECT * FROM Archive_Loue;
9

```

Données Messages Notifications

Affichage des ligne

	abonne_id integer	exemplaire_id integer	date_emprunt date	date_retour date
1	1	1	2024-01-01	2024-01-10

Créer une table Archive\_Loue contenant tous les emprunts plus anciens qu'un mois. Ensuite, les emprunts plus anciens qu'un mois sont supprimés de la table Loue.

Afficher les emprunts archivés et supprimés (table Archive\_Loue et suppression d'emprunts) : Après avoir créé la table Archive\_Loue et supprimé les anciens emprunts dans la table Loue.

- Sélection des ouvrages empruntés plus de 2 fois

Requête

Historique

1

2

3

4

5

6

▼

```
SELECT Ouvrages.titre, COUNT(Loue.exemplaire_id) AS nombre_emprunts
FROM Loue
JOIN Ouvrages ON Loue.exemplaire_id = Ouvrages.id
GROUP BY Ouvrages.id
HAVING COUNT(Loue.exemplaire_id) >= 2;
```

Données

Messages

Notifications

≡

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

Affichage des lignes : 1 à 2

✎

	titre character varying (255) 🔒	nombre_emprunts bigint 🔒
1	Les Misérables	6
2	Le Petit Prince	3

- Sélection des ouvrages utilisés uniquement en interne

Requête

Historique

1 **SELECT** titre, auteur, edition  
2 **FROM** Ouvrages  
3 **WHERE** usage\_interne = TRUE;

Données

Messages

Notifications

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

Affichage des lignes

	titre character varying (255) 🔒	auteur character varying (255) 🔒	edition character varying (255) 🔒
1	Encyclopédie Universelle	Collectif	Larousse
2	Le Petit Prince	Antoine de Saint-Exupéry	Gallimard

sélectionne les titres, auteurs et éditions des ouvrages dont l'attribut usage\_interne est défini à TRUE (utilisation interne uniquement).

Affiche les ouvrages réservés à l'usage interne.

- Vérification de la disponibilité d'un livre avant un emprunt:

```
Requête Historique
1 CREATE OR REPLACE FUNCTION check_book_availability()
2 RETURNS TRIGGER AS $$
3 BEGIN
4     IF (SELECT COUNT(*) FROM Reserve WHERE exemplaire_id = NEW.exemplaire_id) > 0 THEN
5         RAISE EXCEPTION 'Le livre est déjà réservé.';
6     END IF;
7     RETURN NEW;
8 END;
9 $$ LANGUAGE plpgsql;
10
11 CREATE TRIGGER check_availability
12 BEFORE INSERT ON Loue
13 FOR EACH ROW
14 EXECUTE FUNCTION check_book_availability();
15

Données Messages Notifications
CREATE TRIGGER

Requête exécutée avec succès en 97 msec.
```

:

Cette fonction et le déclencheur empêchent l'emprunt d'un livre déjà réservé. Avant d'ajouter un emprunt à la table Loue, le déclencheur vérifie si le livre est déjà réservé en utilisant la table Reserve.

test :

```

1  INSERT INTO Loue (abonne_id, exemplaire_id, date_emprunt, date_retour)
2  VALUES (2, 1, '2025-02-02', '2025-02-10');
3

```

Données Messages Notifications

ERREUR: Le livre est déjà réservé.  
 CONTEXT: fonction PL/pgSQL check\_book\_availability(), ligne 4 à RAISE

ERREUR: Le livre est déjà réservé.  
 État SQL : P0001

```

1  SELECT * FROM Loue;

```

Données Messages Notifications

	abonne_id [PK] integer	exemplaire_id [PK] integer	date_emprunt date	date_retour date
1	5	7	2025-02-06	2025-02-20
2	3	4	2025-02-05	2025-02-28
3	4	4	2025-03-05	2025-03-28
4	6	4	2025-04-05	2025-03-29
5	7	4	2025-06-09	2025-07-28
6	2	4	2025-08-05	2025-08-13
7	1	4	2025-08-15	2025-09-01
8	6	6	2025-02-05	2025-02-28
9	2	3	2025-02-05	2025-02-28
10	7	2	2025-02-05	2025-02-28
11	6	2	2025-02-05	[null]
12	4	2	2025-02-06	[null]

- Insertion d'une nouvelle bibliothèque et de ses exemplaires

Cette requête insère une nouvelle bibliothèque dans la base de données et copie tous les ouvrages existants dans la nouvelle bibliothèque en définissant un emplacement par défaut et en les marquant comme disponibles.

- Sélection des événements programmés dans une bibliothèque

Requête

Historique

1

2

3

4

SELECT e.nom, e.date, e.theme, e.duree

FROM Evenements e

WHERE e.bibliotheque\_id = 1;

Données

Messages

Notifications

≡

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

Affichage des lignes

	nom character varying (255) 🔒	date date 🔒	theme character varying (255) 🔒	duree integer 🔒
1	Conférence sur la littérature	2025-03-15	Littérature	120

sélectionne les événements programmés dans une bibliothèque spécifique, en filtrant les événements par bibliotheque\_id.

- Création des rôle et privilège

```
CREATE ROLE bibliothecaire_user LOGIN PASSWORD 'motdepasse';
GRANT SELECT, INSERT, UPDATE, DELETE ON Ouvrages TO bibliothecaire_user;
GRANT SELECT, INSERT, UPDATE ON Loue TO bibliothecaire_user;
```

```
CREATE ROLE gestionnaire_user LOGIN PASSWORD 'motdepasse';
GRANT ALL PRIVILEGES ON DATABASE bibliotheque TO gestionnaire_user;
```

```
CREATE ROLE admin_user LOGIN PASSWORD 'admin123';
GRANT ALL PRIVILEGES ON DATABASE bibliotheque TO admin_user;
```

Data Output Messages Notifications

GRANT

Query returned successfully in 53 msec.

Cette requête permet de créer 3 rôles représentant un abonné, un gestionnaire de bibliothèque et l'admin de la base de données.

- Abonnés ayant participé à des événements similaires

Requête

Historique

1

2

3

4

5

6

7

8

```
SELECT a.id AS abonne_id, a.nom, a.prenom AS evenement_similaire
FROM Participe p
JOIN Participants part ON p.participant_id = part.id
JOIN Evenements e1 ON p.evenement_id = e1.id
JOIN Evenements e2 ON e2.theme = e1.theme AND e2.id != e1.id
JOIN Abonnes a ON part.email = a.email
WHERE e1.bibliotheque_id = 6;
```

Données

Messages

Notifications

SQL

Affichage des lignes : 1 à 2

	abonne_id integer	nom character varying (255)	evenement_similaire character varying (255)
1	4	Lemoine	Jean
2	5	Garcia	Emma

sélectionne les abonnés ayant participé à des événements ayant un thème similaire dans la même bibliothèque.



- Fréquence de transfert entre les bibliothèques

```

1 SELECT b1.nom AS bibliotheque_source,
2        b2.nom AS bibliotheque_destination,
3        COUNT(t.id) AS nombre_transferts
4 FROM Transferts t
5 JOIN Bibliothèques b1 ON t.bibliotheque_source_id = b1.id
6 JOIN Bibliothèques b2 ON t.bibliotheque_destination_id = b2.id
7 GROUP BY b1.nom, b2.nom
8 ORDER BY nombre_transferts DESC;

```

Data Output Messages Notifications

	bibliotheque_source character varying (255)	bibliotheque_destination character varying (255)	nombre_transferts bigint
1	Bibliothèque Centrale	Médiathèque Lyon	6
2	Médiathèque Lyon	Médiathèque Lyon	3
3	Bibliothèque Universitaire	Médiathèque Lyon	1
4	Médiathèque Lyon	Bibliothèque Universitaire	1
5	Bibliothèque Régionale	Médiathèque Lyon	1

Cette requête nous montre la fréquence d'une bibliothèque source vers une bibliothèque destination, par exemple on voit que la bibliothèque centrale envoie très fréquemment des livres vers la médiathèque de Lyon

- Transfert 30 dernier jours

```

1 SELECT t.bibliotheque_source_id,
2        t.bibliotheque_destination_id,
3        STRING_AGG(t.exemplaire_id::TEXT, ', ') AS ouvrages_transferes,
4        COUNT(t.id) AS total_transferts
5 FROM Transferts t
6 WHERE t.date_transfert BETWEEN CURRENT_DATE - INTERVAL '30 days' AND CURRENT_DATE
7 GROUP BY t.bibliotheque_source_id, t.bibliotheque_destination_id;

```

Data Output Messages Notifications

	bibliotheque_source_id integer	bibliotheque_destination_id integer	ouvrages_transferes text	total_transferts integer
1	1	2	2, 4	2

Cette requête permet de récupérer les transferts sur les 30 derniers jours,

- Vues des livres actuellement empruntés

```
1
2 ✓ CREATE VIEW livres_empruntes AS
3 SELECT Ouvrages.titre, Abonnes.nom, Abonnes.prenom, Loue.date_emprunt, Loue.date_retour
4 FROM Loue
5 JOIN Ouvrages ON Loue.exemplaire_id = Ouvrages.id
6 JOIN Abonnes ON Loue.abonne_id = Abonnes.id
7 WHERE Loue.date_retour IS NULL;
8
9 SELECT * FROM livres_empruntes;
```

Data Output Messages Notifications

	titre character varying (255) 🔒	nom character varying (255) 🔒	prenom character varying (255) 🔒	date_emprunt date 🔒	date_retour date 🔒
1	Le Petit Prince	Dulon	Alex	2025-02-05	[null]
2	Le Petit Prince	Lemoine	Jean	2025-02-06	[null]

Ici 2 livres sont empruntés actuellement, attention ceci est normal un ouvrage peut avoir plusieurs exemplaire dans la bibliothèque.