

# Projet PL/pgSQL – Gestion d’une Bibliothèque

*Version 0.9 – Bêta*

ICOM – Université Lumière Lyon 2  
Master M1 Informatique (Alternance)

Année universitaire 2025–2026

## 1 Objectif du projet

L’objectif de ce projet est d’évaluer la maîtrise des étudiants sur les notions de programmation en **PL/pgSQL** à travers la conception et la manipulation d’une base de données relationnelle. Les points suivants seront particulièrement évalués :

- création et manipulation de fonctions en PL/pgSQL ;
- emploi de boucles et de conditions ;
- gestion des erreurs et utilisation d’exceptions ;
- emploi de curseurs (implicites et explicites) ;
- conception et utilisation de triggers ;
- utilisation de SQL dynamique et de requêtes paramétrées.

Le projet s’effectuera **en binôme**, sur la base **Bibliothèque+** dont le schéma est fourni.

## 2 Description de la base de données

La base **Bibliothèque+** comporte 6 tables principales :

**auteur** Informations sur les auteurs : identifiant, nom et pays d’origine.

**categorie** Catégories de livres (roman, science, essai, etc.).

**livre** Livres disponibles dans la bibliothèque. Chaque livre est associé à un auteur et une catégorie, et comporte un nombre d’exemplaires disponibles.

**emprunteur** Lecteurs inscrits, avec leur identifiant, nom et date d’inscription.

**emprunt** Historique des emprunts : quel livre, quel emprunteur, dates d’emprunt et de retour.

**reservation** Réservations effectuées par les lecteurs, avec un statut (*en attente, validée, annulée*).

## 3 Travail à réaliser

Le projet comporte deux volets : une **partie guidée** et une **partie ouverte**. Le but est de progresser du simple au complexe, tout en mobilisant progressivement les différentes fonctionnalités du langage PL/pgSQL.

### 3.1 Partie 1 : Fonctions et triggers guidés

Cette partie vise à assurer une compréhension commune des bases du PL/pgSQL. Les étudiants devront réaliser les fonctions suivantes :

#### 1. Fonctions de base

- Une fonction `ajouter_auteur(nom, pays)` qui insère un nouvel auteur dans la table `auteur`.
- Une fonction `ajouter_livre(titre, id_auteur, id_categorie, annee, nb_exemplaires)` qui ajoute un livre et vérifie que le nombre d’exemplaires est positif.
- Une fonction `nb_livres_categorie(id_categorie)` qui retourne le nombre total de livres appartenant à une catégorie donnée.

## 2. Fonction avec boucle

- Écrire une fonction `maj_annee_livres()` qui parcourt tous les livres publiés avant l'an 2000 et affiche un message indiquant leur titre et leur année (utiliser une boucle `FOR` ou `WHILE`).

## 3. Fonction avec curseur

- Écrire une fonction `liste_livres_auteur(nom_auteur)` qui parcourt, à l'aide d'un curseur explicite, les livres écrits par un auteur donné et affiche les titres.

## 4. Fonction avec SQL dynamique

- Écrire une fonction `compter_elements(table_name TEXT)` utilisant du SQL dynamique pour retourner le nombre d'enregistrements d'une table passée en paramètre.

## 5. Trigger guidé

- Créer un trigger `verif_disponibilite` sur la table `emprunt` empêchant l'insertion d'un emprunt si le livre n'a plus d'exemplaires disponibles.

### 3.2 Partie 2 : Fonctions et triggers ouverts

Les étudiants doivent concevoir de manière autonome plusieurs fonctions supplémentaires, en respectant les contraintes suivantes :

- au moins 2 fonctions « classiques » (requêtes, insertions ou statistiques) ;
- 1 fonction utilisant un **curseur implicite** ;
- 1 fonction utilisant un **curseur explicite** ;
- 1 fonction utilisant du **SQL paramétré** ;
- 1 fonction utilisant du **SQL dynamique** justifié. Il est **très important de penser à une requête qui ne pourra fonctionner correctement qu'à l'aide d'un SQL dynamique** ;
- au moins un **trigger** cohérent. Ce trigger doit être **cohérent et pertinent** par rapport à cette base de données. Justifiez et expliquez son objectif ainsi que son fonctionnement dans le rapport.

### 3.3 Rapport synthétique (5 pages maximum)

Le rapport devra contenir :

- Pour la partie **guidée** :
  - une brève description de chaque fonction et du trigger ;
  - les principaux problèmes rencontrés et les solutions proposées.
- Pour la partie **ouverte** :
  - le rôle et l'intérêt de chaque fonction ou trigger ;
  - les choix techniques effectués ;
  - les difficultés rencontrées et les pistes d'amélioration.

## Consignes de rendu

- Le travail est à réaliser en **binôme**.
- Le dépôt se fera sur **Moodle** avant le **vendredi 19 décembre 2025 à 23h59**.
- Les fichiers doivent être regroupés dans une **archive ZIP unique**, nommée selon le format suivant : `nom_premier_etudiant_nom_deuxieme_etudiant.zip`
- Cette archive doit contenir les fichiers suivants :
  - `guided_projet.sql` : code complet et commenté des requêtes guidées ;
  - `open_projet.sql` : code complet et commenté des requêtes ouvertes ;
  - `rapport.pdf` : rapport synthétique (5 pages maximum).

## Remarques

Des jeux de données différents (y compris des cas « piégés ») seront utilisés pour tester la robustesse de vos fonctions et triggers. Pensez à :

- gérer les cas inattendus (valeurs nulles, contraintes, erreurs SQL) ;
- commenter clairement votre code ;
- tester chaque fonction avant rendu.