INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

# RAPPORT


**RÉALISÉ PAR :**

**DRIDI Mohamed-Amine**
**LABASSA-DESTOUESSE Nathan**
**IVAKHNIUK Nazarii**
**SABORIT-LOPEZ Iker**


**3 MIC GROUPE E**
**12 JANVIER 2026**

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

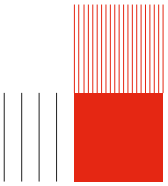
INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

# RAPPORT

## RÉALISÉ PAR :

**DRIDI Mohamed-Amine**
**LABASSA-DESTOUESSE Nathan**
**IVAKHNIUK Nazarii**
**SABORIT-LOPEZ Iker**

**3 MIC GROUPE E**
**12 JANVIER 2026**



INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

# Table des matières

# INTRODUCTION

Ce travail se place dans la continuité du projet BDD1, dans lequel nous avions réalisé le modèle conceptuel puis le modèle relationnel d'une base de données dédiée à la gestion des projets de recherche ainsi qu'au personnel du laboratoire LAAS.

Le projet BDD2 avait pour objectif de mettre en œuvre concrètement ce modèle relationnel dans PostgreSQL. Il s'agissait ensuite de tester la base à l'aide de données cohérentes, puis de valider son bon fonctionnement en répondant à un ensemble de 21 requêtes SQL imposées.
Ce projet nous a permis de travailler sur une base de données proche d'un cas réel, d'automatiser sa création et son alimentation, et de renforcer notre compréhension et notre pratique du langage SQL dans un cadre concret.

# I. PRISE EN MAIN DE LA BASE DE DONNÉES

Nous disposions au départ d'une base de données **totalement vide**, attribuée spécifiquement à notre groupe. L'ensemble des manipulations a été réalisé **exclusivement via le terminal**, conformément aux consignes du projet.

## Connexion à la base

La première étape a consisté à se familiariser avec l'outil **psql**, l'interpréteur SQL de PostgreSQL.
L'accès à la base se fait via la commande suivante, adaptée aux identifiants fournis à notre groupe :

```
psql -h srv-bdens -d bd3a_ng_14_base -U bd3a_ng_14_log -W
```

Cette étape a été réalisée individuellement par chaque membre du groupe afin que tous soient autonomes sur l'utilisation du terminal.

## Manipulations SQL de base

Avant toute implémentation définitive, nous avons effectué plusieurs tests directement dans le terminal afin de maîtriser les commandes SQL essentielles :

- création et suppression de tables (`CREATE TABLE`, `DROP TABLE`),
- insertion et suppression de données (`INSERT`, `DELETE`),
- interrogation des données (`SELECT ... FROM`).

Nous avons également utilisé les commandes spécifiques à psql :

- `\d` pour afficher la liste des tables,
- `\d nom_table` pour consulter la structure détaillée d'une table.

Une fois ces manipulations comprises par l'ensemble du groupe, les tables de test ont été supprimées et nous nous sommes déconnectés proprement de la base à l'aide de la commande `\q`.

# II. IMPLÉMENTATION DU MODÈLE RELATIONNEL

L'implémentation définitive de la base repose sur **deux scripts SQL distincts** :

1. un script de création des tables,
2. un script d'insertion des données.

Conformément aux consignes, tous les noms de tables et d'attributs ont été écrits **en minuscules, sans accents**.

## Création des tables

Avant de créer les tables, nous avons systématiquement supprimé toute table existante portant le même nom, à l'aide de la commande :

```
DROP TABLE IF EXISTS nom_table;
```

Cette suppression est effectuée en respectant **l'ordre inverse des dépendances**, afin d'éviter les erreurs liées aux clés étrangères (suppression des tables filles avant les tables mères).

Les tables ont ensuite été créées à partir du schéma relationnel validé lors du projet BDD1, en définissant :

- les clés primaires,
- les clés étrangères,
- les contraintes d'intégrité (NOT NULL, références).

Cette partie a été principalement réalisée par **Nazarii** et **Iker**, avec une attention particulière portée à la cohérence globale du schéma.

Dans certains cas, nous avons fait le choix d'ajouter des **index** sur des attributs fréquemment utilisés dans les jointures ou les filtres (`WHERE`). Ce choix vise à améliorer les performances des requêtes, notamment en limitant les parcours complets de tables lorsque cela est pertinent.

## Jeu de données

Le jeu de données n'étant pas fourni, il a été **entièrement conçu par le groupe**.
Nous avons veillé à créer des données :

- réalistes,
- suffisantes pour tester tous les cas demandés par les requêtes,

- cohérentes avec les contraintes du modèle (doctorants avec plusieurs encadrants, publications avec auteurs externes, projets partagés, etc.).

L'insertion des données a été réalisée par **Amine** et **Nathan**, à l'aide de commandes `INSERT INTO`, regroupées dans un script dédié afin de faciliter les tests et la réinitialisation de la base.

# III. ÉCRITURE ET VALIDATION DES REQUÊTES SQL

## Organisation du travail

La rédaction des requêtes constitue le cœur du projet. Les **21 requêtes SQL** demandées, ainsi que les tâches complémentaires, ont été réparties équitablement entre les membres du groupe :

- **Nazarii** : implémentation d'une partie du schéma relationnel, rédaction des requêtes 1 à 5, relecture et validation globale
- **Iker** : implémentation des tables principales et de leurs dépendances, rédaction des requêtes 6 à 10
- **Amine** : conception du jeu de données et rédaction des requêtes 11 à 16
- **Nathan** : insertion des données, rédaction des requêtes 17 à 21 et tests finaux

Chaque membre a travaillé de manière autonome sur ses requêtes, puis les requêtes ont été relues collectivement afin d'en vérifier la correction syntaxique, la conformité avec l'énoncé et la cohérence des résultats obtenus.

Toutes les requêtes sont regroupées dans un **troisième script SQL**, distinct de ceux de création et d'insertion.

## Tests et exécution

Pour tester l'ensemble du projet, nous nous reconnectons à la base de données et exécutons successivement les trois scripts SQL, dans l'ordre suivant :

1. script de création des tables,
2. script d'insertion des données,
3. script des requêtes.

L'exécution se fait via la commande :

```
\i nom_du_script.sql
```

Cette méthode permet de **reconstruire intégralement la base** en une seule fois et de vérifier rapidement la validité de l'ensemble du travail.

# IV. UTILISATION D'UNE INTERFACE GRAPHIQUE

Après avoir maîtrisé l'ensemble des manipulations via le terminal, comme exigé dans l'énoncé du projet, nous avons également pris connaissance de l'existence d'interfaces graphiques permettant d'interagir avec une base de données PostgreSQL.

Ces outils permettent notamment de visualiser plus facilement les tables, de parcourir les données et d'exécuter les requêtes SQL de manière plus intuitive. Néanmoins, en accord avec les consignes pédagogiques, nous avons fait le choix de réaliser l'ensemble du projet directement depuis le terminal, sans utiliser d'interface graphique, que ce soit pour la création des tables, l'insertion des données ou l'exécution des requêtes.

Cette démarche nous a permis d'approfondir notre compréhension du fonctionnement des scripts SQL, ainsi que d'identifier plus précisément les erreurs possibles liées aux contraintes ou aux dépendances entre les tables. Dans le cadre de ce projet, l'utilisation du terminal nous a paru plus formatrice, l'interface graphique pouvant être envisagée par la suite comme un outil de confort, mais ne devant pas se substituer à une réelle maîtrise du langage SQL.

# CONCLUSION

Ce projet de base de données nous a permis de mettre en application l'ensemble des notions étudiées en cours et en travaux dirigés de BDD2. À partir du modèle relationnel élaboré lors du projet BDD1, nous avons réalisé l'implémentation complète d'une base de données sous PostgreSQL, en automatisant à la fois sa création, son alimentation et son exploitation au moyen de scripts SQL.

Ce travail nous a appris à manipuler une base de données concrète, à gérer les dépendances entre les différentes tables, à construire un jeu de données cohérent et à rédiger des requêtes SQL plus avancées, faisant intervenir des jointures, des sous-requêtes, des agrégations ainsi que des conditions complexes. Le découpage du travail en scripts distincts nous a également permis de structurer notre démarche et de faciliter les phases de test.

Le projet a été mené de manière collaborative, avec une répartition claire des tâches au sein du groupe. Cette organisation nous a permis de travailler efficacement tout en garantissant que chaque membre du groupe soit impliqué dans des aspects techniques variés du projet.

Dans l'ensemble, ce projet nous a permis de consolider nos compétences en SQL et en conception de bases de données relationnelles, tout en nous confrontant à une méthodologie proche de celle utilisée dans un contexte professionnel.

# ANNEXES

```
bd3a_ng_14_base=> \i laas_db_schema.sql
DROP TABLE
DROP TABLE
DROP TABLE
psql:laas_db_schema.sql:10: NOTICE:  drop cascades to 3 other objects
DÉTAIL : drop cascades to view collaborateurs_internes_s001
drop cascades to view collaborateurs_externes_s001
drop cascades to view publication_pas_que_a
DROP TABLE
psql:laas_db_schema.sql:11: NOTICE:  drop cascades to view publication_par_pays
DROP TABLE
DROP TABLE
DROP TABLE
DROP TABLE
DROP TABLE
DROP TABLE
DROP TABLE
DROP TABLE
DROP TABLE
DROP TABLE
DROP TABLE
DROP TABLE
DROP TABLE
DROP TABLE
DROP TABLE
DROP TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
```

**Figure 1 :** Création des tables de la base de données

```
bd3a_ng_14_base=> \i laas_db_data.sql
psql:laas_db_data.sql:7: WARNING:  SET CONSTRAINTS can only be used in transaction blocks
SET CONSTRAINTS
BEGIN
INSERT 0 15
INSERT 0 4
INSERT 0 5
INSERT 0 3
INSERT 0 2
INSERT 0 10
INSERT 0 5
INSERT 0 6
INSERT 0 4
INSERT 0 7
INSERT 0 7
INSERT 0 6
INSERT 0 12
INSERT 0 10
INSERT 0 12
INSERT 0 4
INSERT 0 16
INSERT 0 19
INSERT 0 10
INSERT 0 18
INSERT 0 14
COMMIT
```

**Figure 2 :** Insertion des données dans la base de données

```
bd3a_ng_14_base=> \i requetes.sql
psql:requetes.sql:1: NOTICE:  view "publication_pas_que_a" does not exist, skipping
DROP VIEW
psql:requetes.sql:2: NOTICE:  view "publication_par_pays" does not exist, skipping
DROP VIEW
psql:requetes.sql:3: NOTICE:  view "collaborateurs_internes_s001" does not exist, skipping
DROP VIEW
psql:requetes.sql:4: NOTICE:  view "collaborateurs_externes_s001" does not exist, skipping
DROP VIEW
--Q1--
 nom | grade
-----+-------
(0 ligne)


--Q2--
 nom |   pays
-----+-----------
 Tan | Singapour
 Lee | Singapour
(2 lignes)


--Q3--
CREATE VIEW
CREATE VIEW
 nb_collaborateurs
-------------------
                 6
(1 ligne)


--Q4--
 count
-------
     2
(1 ligne)


--Q5--
 id_personnel | count
--------------+-------
           11 |     1
           15 |     1
            3 |     2
           13 |     2
            8 |     1
(5 lignes)
```

**Figure 3 :** Exécution des requêtes SQL demandées

```
PS C:\Users\user\Documents\INSA\3MIC IR\bdd2\Projet_BDD> psql -h srv-bdens -d bd3a_ng_14_base -U bd3a_ng_14_log -W
Mot de passe :
psql (18.1, serveur 10.23 (Ubuntu 10.23-0ubuntu0.18.04.2))
Attention : l'encodage console (850) diffère de l'encodage Windows (1252).
            Les caractères 8 bits peuvent ne pas fonctionner correctement.
            Voir la section « Notes aux utilisateurs de Windows » de la page
            référence de psql pour les détails.
Connexion SSL (protocole : TLSv1.3, chiffrement : TLS_AES_256_GCM_SHA384, compression : désactivé, ALPN : aucun)
Saisissez « help » pour l'aide.
```

**Figure 4 :** Connexion aux serveurs de bases de données de l'INSA

```
PS C:\Users\user\Documents\INSA\3MIC IR\bdd2\Projet_BDD> git add .
PS C:\Users\user\Documents\INSA\3MIC IR\bdd2\Projet_BDD> git commit -m "version 0.5"
[main 5dd99db] version 0.5
 1 file changed, 24 insertions(+), 13 deletions(-)
PS C:\Users\user\Documents\INSA\3MIC IR\bdd2\Projet_BDD> git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 524 bytes | 524.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/PeRiOuS-AD/Projet_BDD/
   5597717..5dd99db  main -> main
```

**Figure 5 :** Utilisation de Git pour le travail collaboratif

```
bd3a_ng_14_base=> \q
```

**Figure 6 :** Déconnexion de la base de données