

2025

Projet GLPI TAD

Ryad ZIOUCHE Enzo EDMOND

ING 2 GIA

# Introduction

Dans le cadre du cours de Bases de Données Avancées, ce projet vise à améliorer la gestion du système GLPI, utilisé pour l’administration du parc informatique de CY Tech. L’un des principaux enjeux réside dans la prise en compte de la structure multi-sites de l’établissement, qui dispose d’infrastructures informatiques réparties entre les campus de Cergy et de Pau. Cette spécificité implique une organisation rigoureuse des équipements et des interventions, nécessitant une base de données optimisée et centralisée.

L’objectif est de concevoir un système performant qui assure une gestion fluide et efficace des ressources sur les deux sites, tout en garantissant la cohérence et la synchronisation des données. La refonte de la base de données doit permettre d’améliorer la rapidité d’exécution des requêtes, de simplifier l’administration des équipements et des utilisateurs, et d’assurer une répartition équilibrée des ressources informatiques entre Cergy et Pau.

Ce projet repose sur une analyse approfondie du fonctionnement actuel du système et sur l’implémentation de solutions avancées telles que les triggers, les procédures stockées et les mécanismes de bases de données distribuées. L’intégration de ces technologies vise à garantir une meilleure accessibilité aux données, une gestion plus fluide des incidents et une coordination efficace des équipes techniques intervenant sur les différents sites. L’objectif final est de proposer une architecture optimisée qui réponde aux exigences spécifiques des deux campus et qui facilite l’administration du parc informatique tout en améliorant l’expérience des utilisateurs.

# Sommaire

[Introduction 1](#_Toc194608988)

[Sommaire 2](#_Toc194608989)

[I. Organisation de la gestion multisite 3](#_Toc194608990)

[1. Présentation du Système et Gestion des Données 3](#_Toc194608991)

[2. Modélisation Conceptuelle et Logique des Données 5](#_Toc194608992)

[3. Fragmentation des Données et Synchronisation avec Pau 6](#_Toc194608993)

[II. Création des rôles 8](#_Toc194608994)

[III. Index et Tablespaces 8](#_Toc194608995)

[1. Tablespaces 8](#_Toc194608996)

[2. Index 10](#_Toc194608997)

[IV. PL SQL 10](#_Toc194608998)

[1. Triggers 10](#_Toc194608999)

[2. Curseurs 12](#_Toc194609000)

[3. Procédures 13](#_Toc194609001)

[4. Fonctions 14](#_Toc194609002)

[V. Tests 15](#_Toc194609003)

[Préparation du Jeu de Données 15](#_Toc194609004)

[Exécution des Tests 15](#_Toc194609005)

[Résultats des Tests 16](#_Toc194609006)

[Conclusion 17](#_Toc194609007)

# Organisation de la gestion multisite

# Présentation du Système et Gestion des Données

Dans le cadre de l’optimisation du système GLPI pour la gestion du parc informatique de CY Tech, notre projet vise à structurer et centraliser les différentes informations liées aux utilisateurs, aux matériels et aux incidents, tout en tenant compte de la répartition des infrastructures sur les deux sites de l’établissement : Cergy et Pau. Cette gestion centralisée permet de suivre l’ensemble des équipements, d’assurer un support efficace et d’optimiser la gestion des ressources matérielles et humaines.

**Gestion des utilisateurs**

Les utilisateurs du système sont les différents membres de l’établissement, qu’ils soient administrateurs, techniciens ou simples utilisateurs. Chaque utilisateur est identifié par un identifiant unique et possède des informations essentielles telles que son nom, son prénom et son adresse email. Ces informations sont stockées dans une table dédiée, facilitant la gestion des attributions de matériel et le suivi des demandes d’intervention.

**Gestion des matériels**

Le parc informatique est constitué d’un ensemble d’équipements variés, allant des ordinateurs aux imprimantes, en passant par les serveurs et les équipements réseau. Chaque matériel est identifié de manière unique et classé selon son type. Un matériel est également associé à un site spécifique, indiquant s’il est localisé à Cergy ou à Pau. Cette organisation permet un suivi précis des ressources et une meilleure gestion du stock, notamment en ce qui concerne la maintenance et le renouvellement des équipements.

**Gestion des tickets**

Le système de gestion des tickets permet d’assurer un suivi efficace des incidents et des demandes de support technique. Lorsqu’un utilisateur rencontre un problème avec un matériel, il peut soumettre un ticket décrivant l’incident. Chaque ticket est associé à un matériel précis, un utilisateur et un site. Son statut évolue au cours de son traitement, permettant aux équipes techniques d’assurer un suivi rigoureux jusqu’à la résolution du problème.

**Gestion des Connexions Réseau**

La gestion des équipements informatiques implique également la prise en compte des infrastructures réseau. Chaque matériel connecté dispose d’une adresse IP et peut être rattaché à un réseau spécifique. L’identification des équipements au sein du réseau est essentielle pour assurer leur suivi et faciliter le diagnostic en cas de problème.

**Gestion des Événements Matériels**

Le suivi des matériels ne se limite pas uniquement à leur attribution et à leur localisation. Il est également essentiel d’enregistrer les événements majeurs qui leur sont liés. Ces événements peuvent inclure des pannes, des mises à jour, des changements de configuration ou encore des interventions de maintenance.

**Gestion des Notifications**

Afin d’améliorer la communication entre les utilisateurs et le service technique, un système de notifications est mis en place. Chaque action importante effectuée dans le système peut générer une notification envoyée à l’utilisateur concerné.

**Gestion des Stocks**

La gestion des stocks est un élément clé du suivi des matériels. Elle permet de connaître à tout moment la disponibilité des équipements et d’anticiper les besoins en approvisionnement. Chaque matériel stocké est répertorié avec une quantité et un seuil d’alerte permettant de signaler lorsque le stock devient insuffisant. Les stocks sont gérés par site, ce qui permet d’optimiser la répartition des ressources entre Cergy et Pau. Grâce à cette organisation, les responsables peuvent rapidement identifier les matériels disponibles et organiser leur transfert si nécessaire.

## Modélisation Conceptuelle et Logique des Données

Le système repose sur une base de données relationnelle centralisée sur le site de Cergy, avec certaines données fragmentées et synchronisées sur le site secondaire de Pau via des vues matérialisées. La base de données est conçue pour structurer les informations essentielles, telles que les utilisateurs, les matériels, les tickets de support, les événements et les stocks.

Voici la représentation des tables et de leurs relations sous forme de Modèle Logique des Données :

**Utilisateurs (id\_utilisateur, nom, prenom, email)**

Soumettent des tickets et se voient attribuer des matériels.

**Sites(id\_site, nom\_site)**

Représentent les deux localisations principales du système (Cergy et Pau).

**Types\_de\_Materiels(id\_type\_materiel, nom\_type\_materiel)**

Catégorisent les matériels en fonction de leur usage.

**Matériels(id\_materiel, nom\_materiel, #id\_type\_materiel, #id\_site)**

Classés par type et affectés à un site.

**Réseaux(id\_reseau, adresse\_ip, sous\_reseau, #id\_materiel)**

Identifient les équipements connectés et leur adresse IP.

**Attributions(id\_attribution, #id\_utilisateur, #id\_materiel)**

Relient les utilisateurs aux matériels qu’ils utilisent.

**Tickets(ticket\_id, titre, description, statut, date\_creation, #id\_utilisateur, #id\_materiel, #id\_site)**

Assurent le suivi des incidents et des demandes de support.

**Event\_Logs(log\_id, type, description, date\_event, #id\_materiel)**

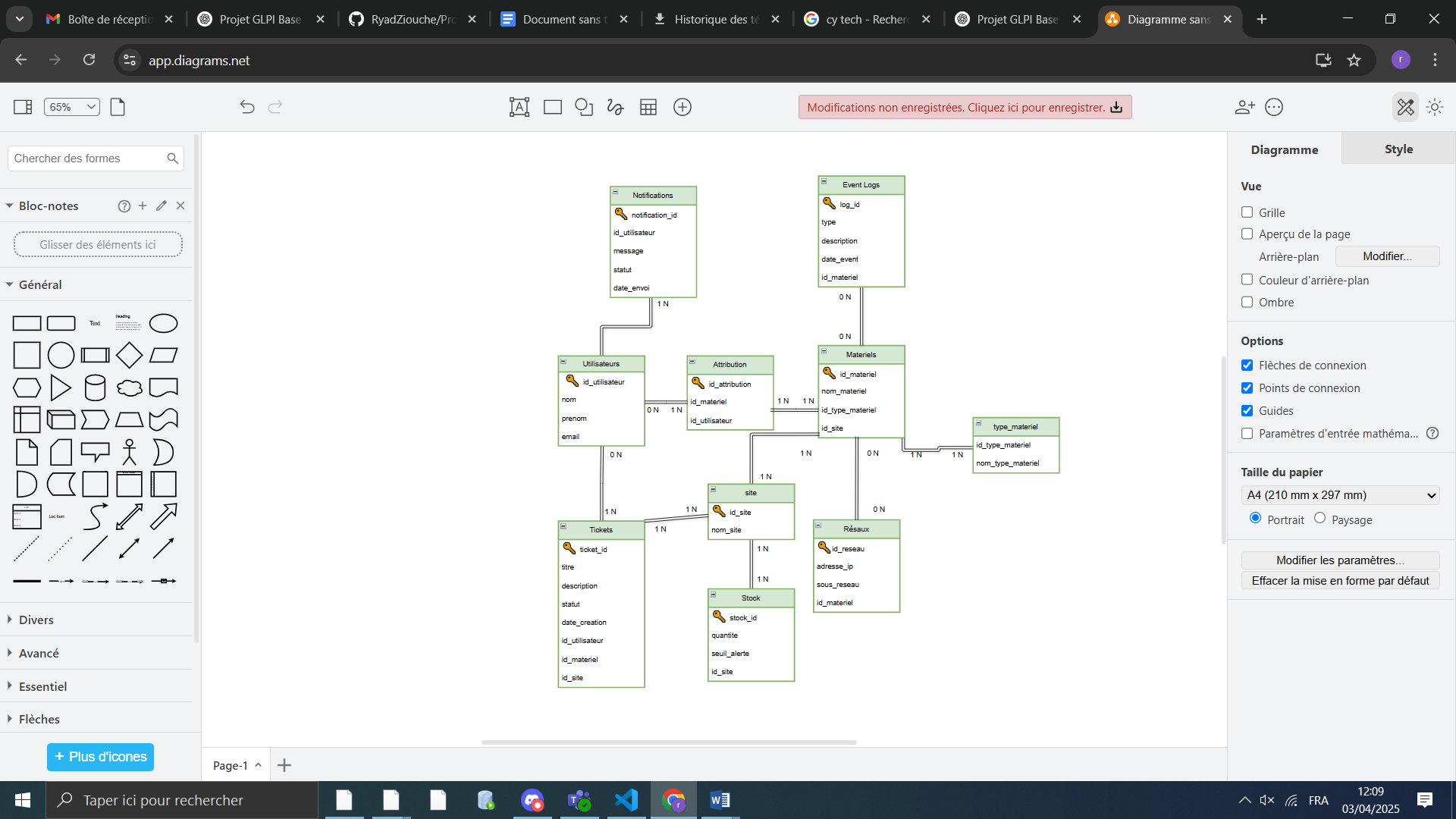
Enregistrent les événements liés aux matériels.

**Notifications(notification\_id, #id\_utilisateur, message, statut, date\_envoi)**

Informent les utilisateurs des mises à jour concernant leurs tickets.

**Stock(stock\_id, #id\_materiel, quantite, seuil\_alerte, #id\_site)**

Permet de gérer la disponibilité des équipements par site.



Modèle Conceptuel de Données

## Fragmentation des Données et Synchronisation avec Pau

Le site Pau ne stocke que les informations spécifiques à ses propres matériels et tickets. La mise à jour se fait à travers des vues matérialisées depuis la base centrale de Cergy.

Les vues matérialisées stockées sur Pau sont les suivantes :

**Materiels\_Pau(id\_materiel, nom\_materiel, #id\_type\_materiel, #id\_site)**

→ Matériels spécifiques au site Pau.

**Tickets\_Pau(ticket\_id, titre, description, statut, date\_creation, #id\_utilisateur, #id\_materiel, #id\_site)**

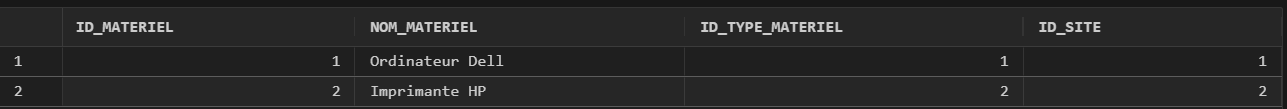
→ Tickets gérés localement à Pau.

**Stock\_Pau(stock\_id, #id\_materiel, quantite, seuil\_alerte, #id\_site)**

→ Gestion des stocks sur le site de Pau.

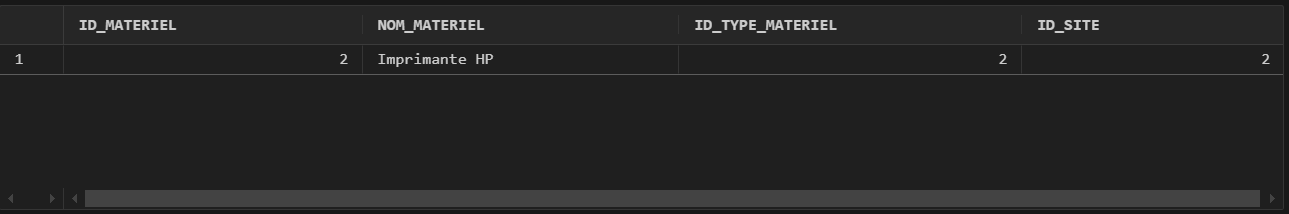
Cette fragmentation garantit que chaque site dispose uniquement des informations nécessaires à son fonctionnement tout en restant synchronisé avec la base centrale.

Illustration de la fragmentation horizontale avec la table Matériels :

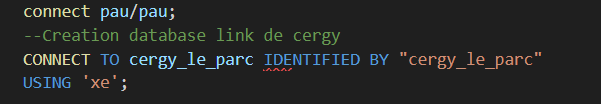
**Table Matériels centralisé sur Cergy :  
**

Fragmentation à partir de l’id site

**Table Materiels\_pau :**



Pour permettre au site Pau d’accéder aux données de Cergy, un Database Link a été créé. Un DB Link est une connexion permettant à une base de données Oracle d’accéder à des tables situées sur une autre base Oracle distante.

Dans ce projet, nous avons défini un DB Link "db\_cergy", qui permet au site Pau d’interroger la base centrale de Cergy comme s’il s’agissait de ses propres tables locales.  


Concernant le rafraîchissement des vue nous avons opté un REFRESH ON DEMAND car il permet d’éviter une surcharge inutile sur la base centrale en limitant les mises à jour aux moments nécessaires et d’optimiser la gestion du réseau, car chaque synchronisation implique un transfert de données entre Cergy et Pau.

Cette approche garantit un accès rapide aux données locales tout en assurant leur synchronisation avec la base de données principale.

# Création des rôles

Dans ce projet, nous avons défini plusieurs rôles spécifiques à différents utilisateurs de l'application GLPI, en fonction de leurs responsabilités sur les différents sites (Cergy et Pau). Pour chaque rôle, une vue combinée a été créée, regroupant plusieurs tables pertinentes. Ces vues permettent aux utilisateurs d'accéder uniquement aux informations nécessaires à leur fonction, tout en garantissant une gestion sécurisée des accès.

Les vues ont été choisies car elles permettent de simplifier l'accès aux données, tout en masquant la complexité de la structure sous-jacente des tables. Elles offrent une abstraction qui facilite l'interrogation des données sans exposer directement les tables, ce qui renforce la sécurité du système. De plus, une vue permet de combiner plusieurs tables dans une seule requête, ce qui est particulièrement utile pour les rôles qui nécessitent une vue d'ensemble sur plusieurs entités.

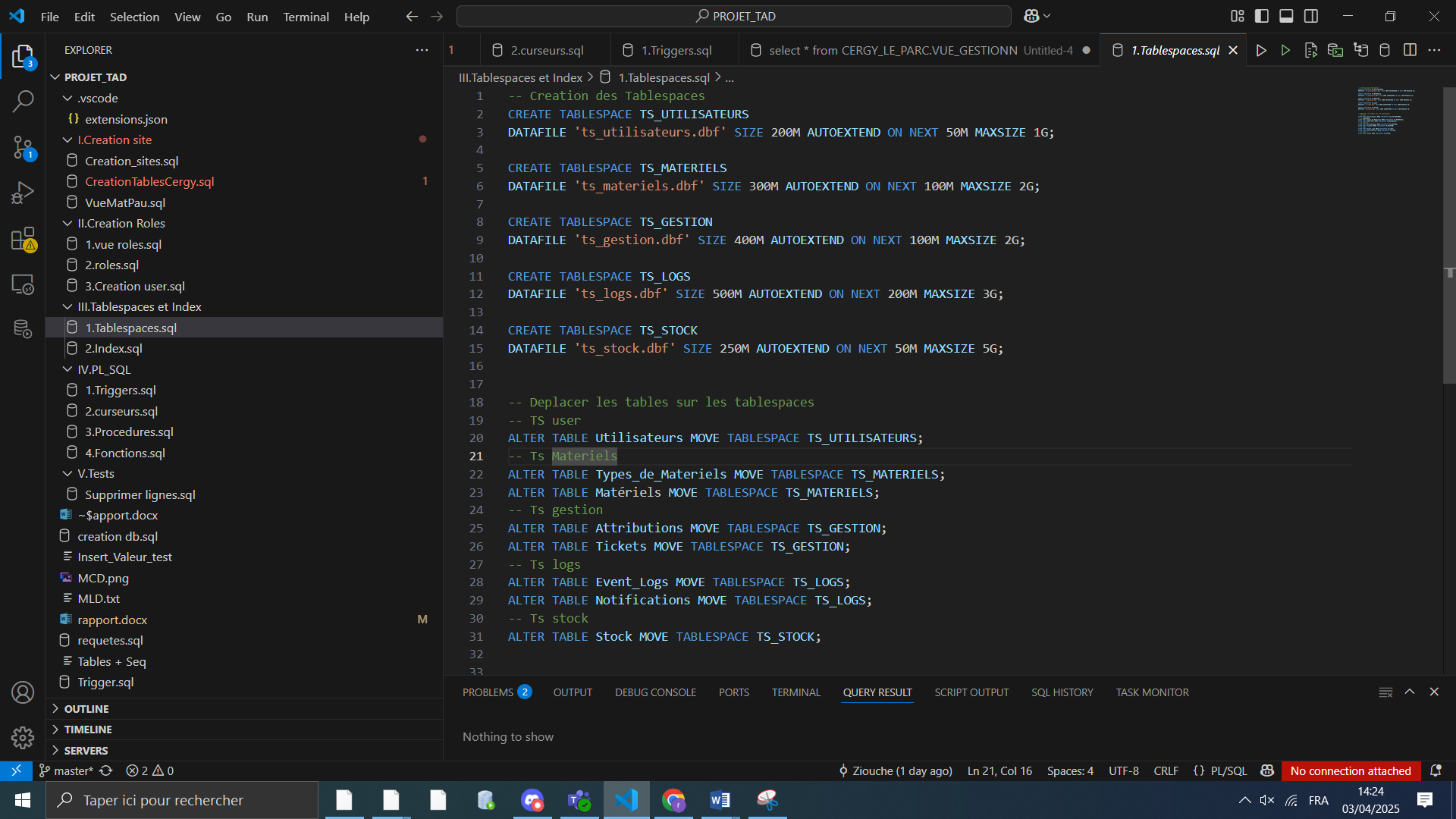
Définition des Rôles :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rôle | Vues | Priviléges | Table associé | Site |
| Administrateur | vue\_technicien | Tous | Tous | Pau et Cergy |
| Technicien Cergy/Pau | vue\_technicien | SELECT | Tickets, Matériels | Rôle par Site |
| Gestionnaire Stock Cergy/Pau | vue\_gestionnaire\_stock | SELECT | Stock, Matériels | Rôle par Site |
| Responsable Cergy/Pau | vue\_responsable | SELECT | Utilisateurs, Matériels,  Attribution,  Stock | Rôle par Site |
| Administrateur Réseau | vue\_admin\_reseau | SELECT | Réseau,  Events Logs | Cergy et Pau |
| Gestionnaire Affectation Cergy/Pau | vue\_gestionnaire\_affectation | SELECT | Attributions, Utilisateurs, Matériels, Sites | Rôle par Site |

# Index et Tablespaces

## Tablespaces

Les tablespaces sont utilisés pour organiser le stockage des données dans la base de données Oracle. Ils permettent de gérer efficacement l'allocation de l'espace disque et de garantir une répartition optimisée des données en fonction de leur usage. Dans le cadre de ce projet, plusieurs tablespaces ont été créés pour chaque type de données (utilisateurs, matériels, gestion, logs, et stock).



TS\_UTILISATEURS : Ce tablespace est dédié à la table Utilisateurs, qui contient les informations essentielles sur les utilisateurs du système. Un espace de 200 Mo a été alloué, avec une capacité d'auto-extension pour s'adapter à l'évolution du nombre d'utilisateurs.

TS\_MATERIELS : Ce tablespace regroupe les tables relatives aux matériels Matériels et Types\_de\_Materiels. En raison de la taille potentiellement importante de la table des matériels, un espace de 300 Mo a été alloué, avec une auto-extension afin de pouvoir évoluer à mesure que le parc informatique grandit.

TS\_GESTION : Ce tablespace contient des tables liées à la gestion des équipements et des tickets de support, telles que Attributions et Tickets. La taille de ce tablespace a été fixée à 400 Mo pour accommoder les données liées à la gestion des incidents et à l'affectation des matériels.

TS\_LOGS : Les Event\_Logs et Notifications, qui enregistrent les événements système et les alertes, sont placés dans ce tablespace. Un espace de 500 Mo a été attribué, avec un mécanisme d'auto-extension pour suivre l'augmentation des logs générés par l'activité du système.

TS\_STOCK : Ce tablespace contient la table Stock, qui gère les informations relatives à la disponibilité des équipements. Un espace de 250 Mo a été alloué, ce qui permet de suivre l'inventaire tout en laissant une capacité d'extension pour gérer l'évolution des stocks.

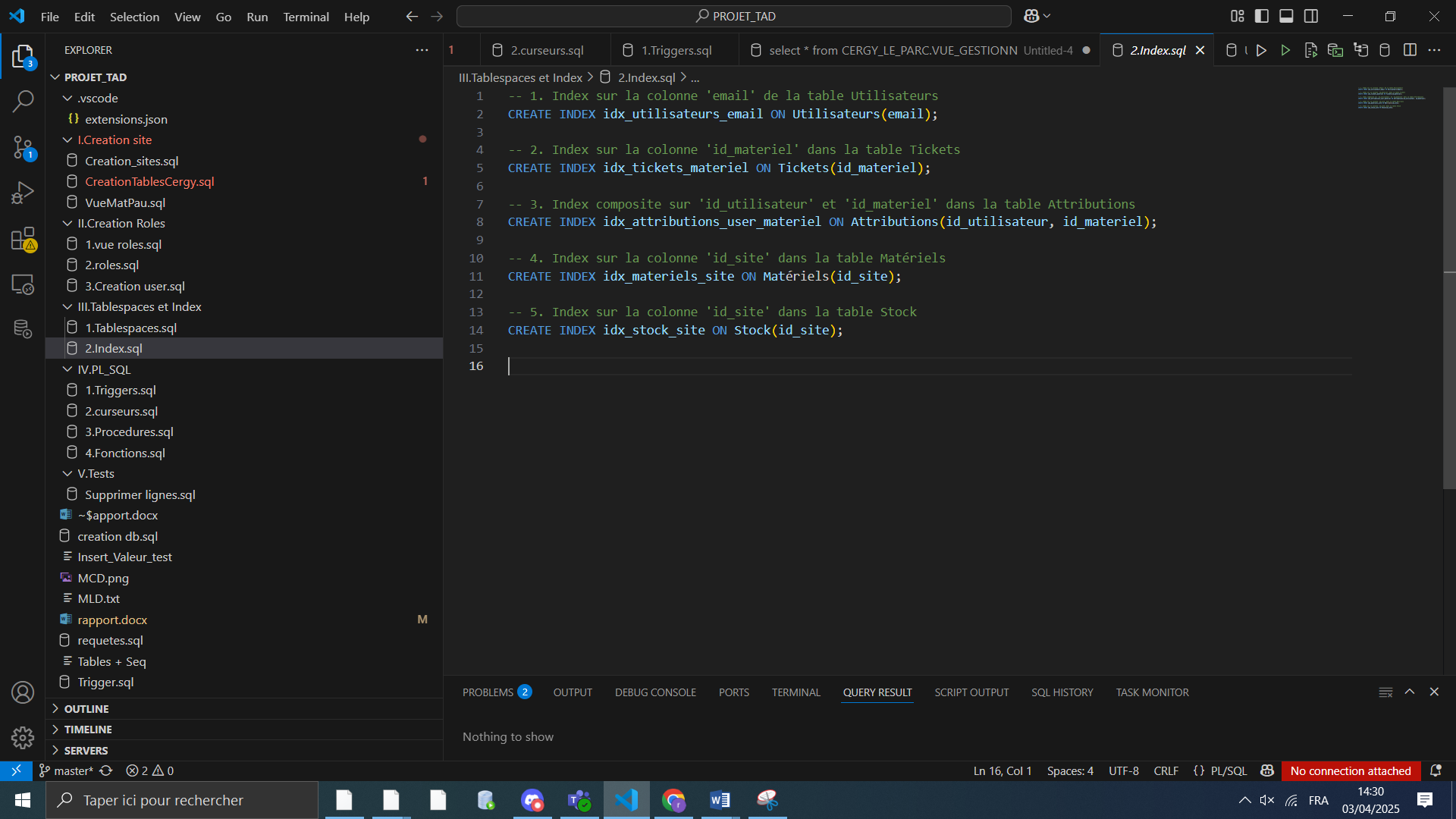
L'utilisation de ces tablespaces permet de séparer les données par catégorie fonctionnelle, optimisant ainsi la gestion des ressources et permettant un meilleur contrôle sur la performance de la base de données.

## Index

Les index améliorent les performances des requêtes en permettant un accès plus rapide aux données. Voici les index créés :

* idx\_utilisateurs\_email : Sur la colonne email de la table Utilisateurs pour accélérer les recherches par email.
* idx\_tickets\_materiel : Sur la colonne id\_materiel de la table Tickets pour optimiser les recherches par matériel.
* idx\_attributions\_user\_materiel : Index composite sur id\_utilisateur et id\_materiel dans la table Attributions pour les requêtes fréquentes.
* idx\_materiels\_site : Sur la colonne id\_site dans la table Matériels pour filtrer rapidement par site.
* idx\_stock\_site : Sur la colonne id\_site dans la table Stock pour faciliter les recherches de stocks par site.

Ces index permettent d'améliorer les performances des requêtes courantes, en particulier celles qui filtrent ou trient sur ces colonnes.

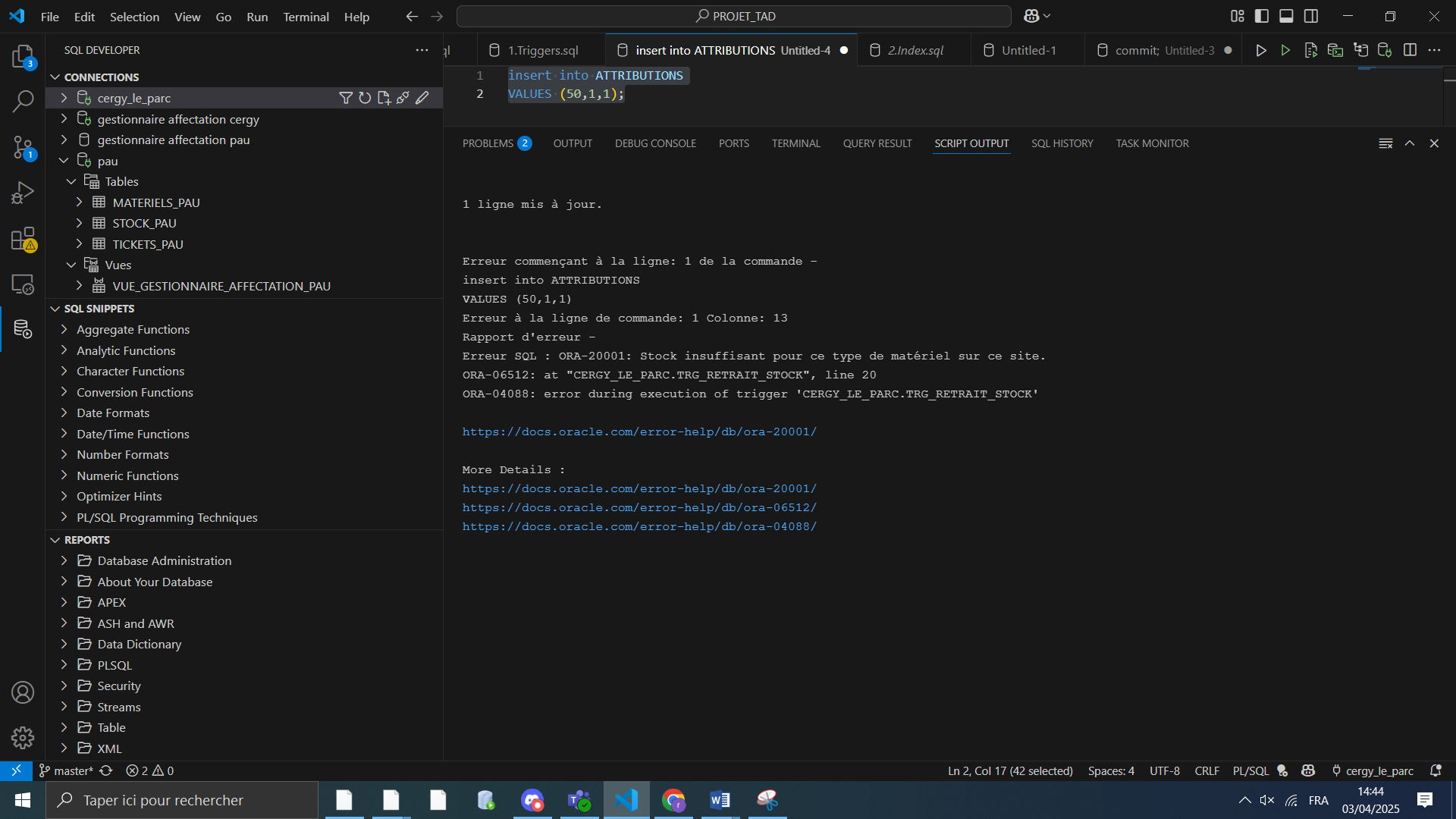
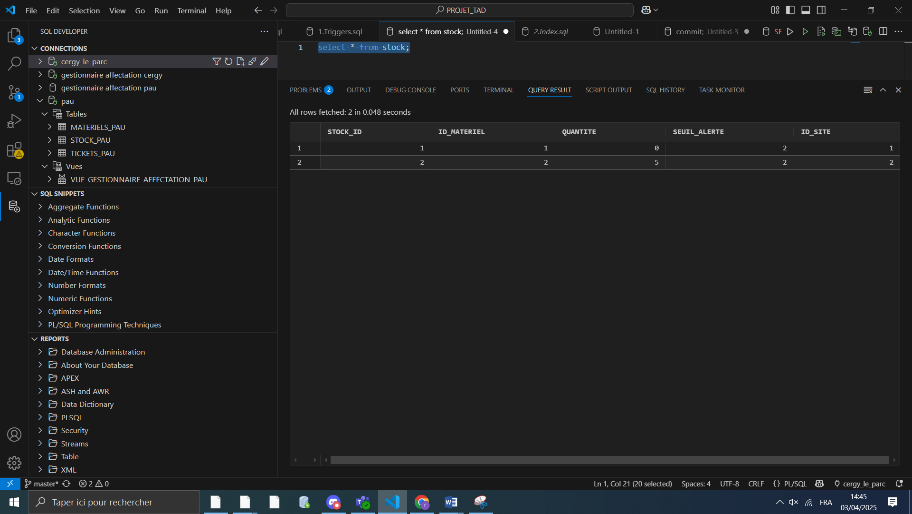


# PL SQL

## Triggers

Les triggers sont des objets dans la base de données qui exécutent automatiquement des actions en réponse à des événements spécifiques. Voici une explication détaillée des triggers créés dans ce projet :

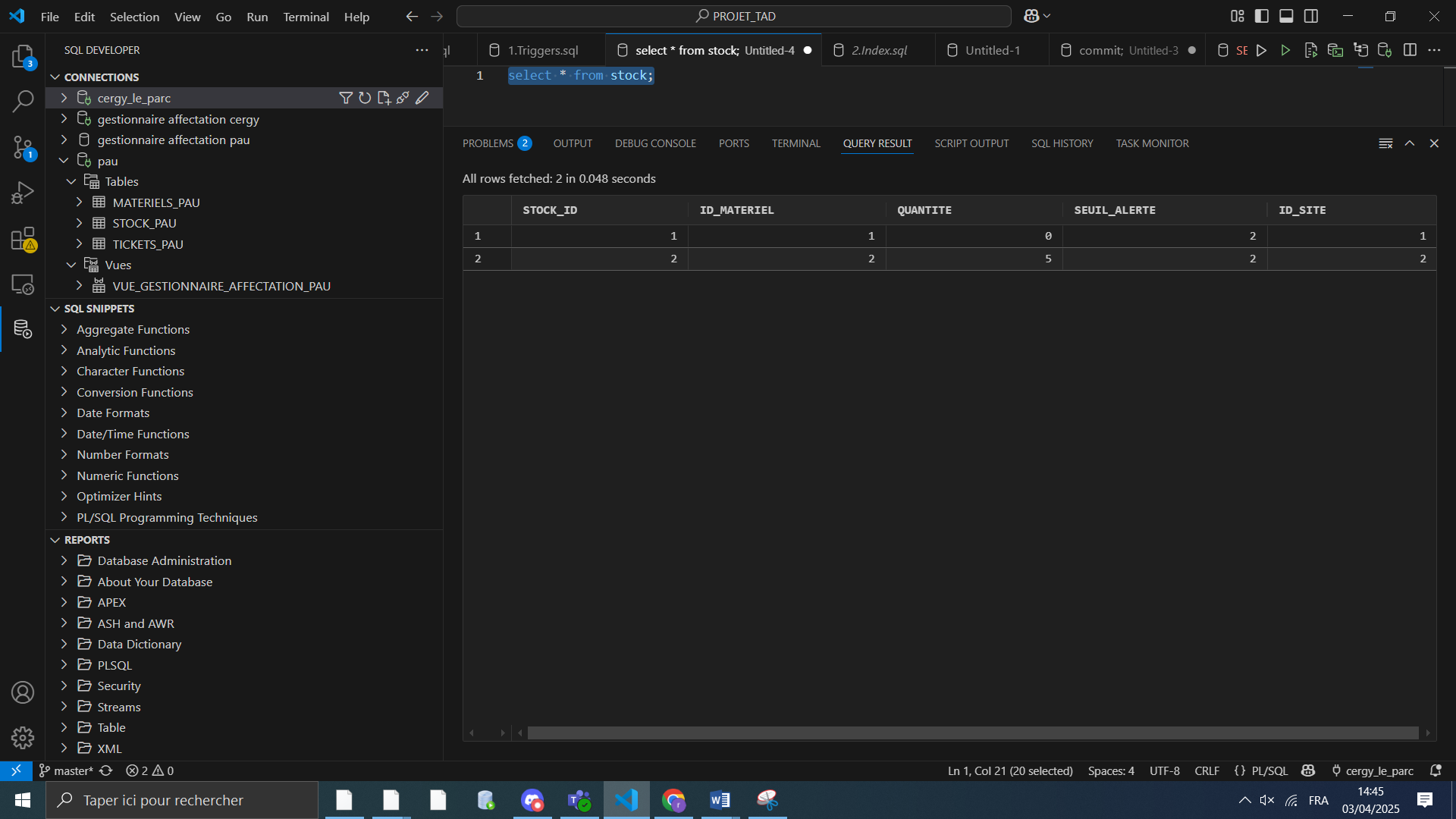
**Trigger - Retrait de stock après attribution de matériel :**



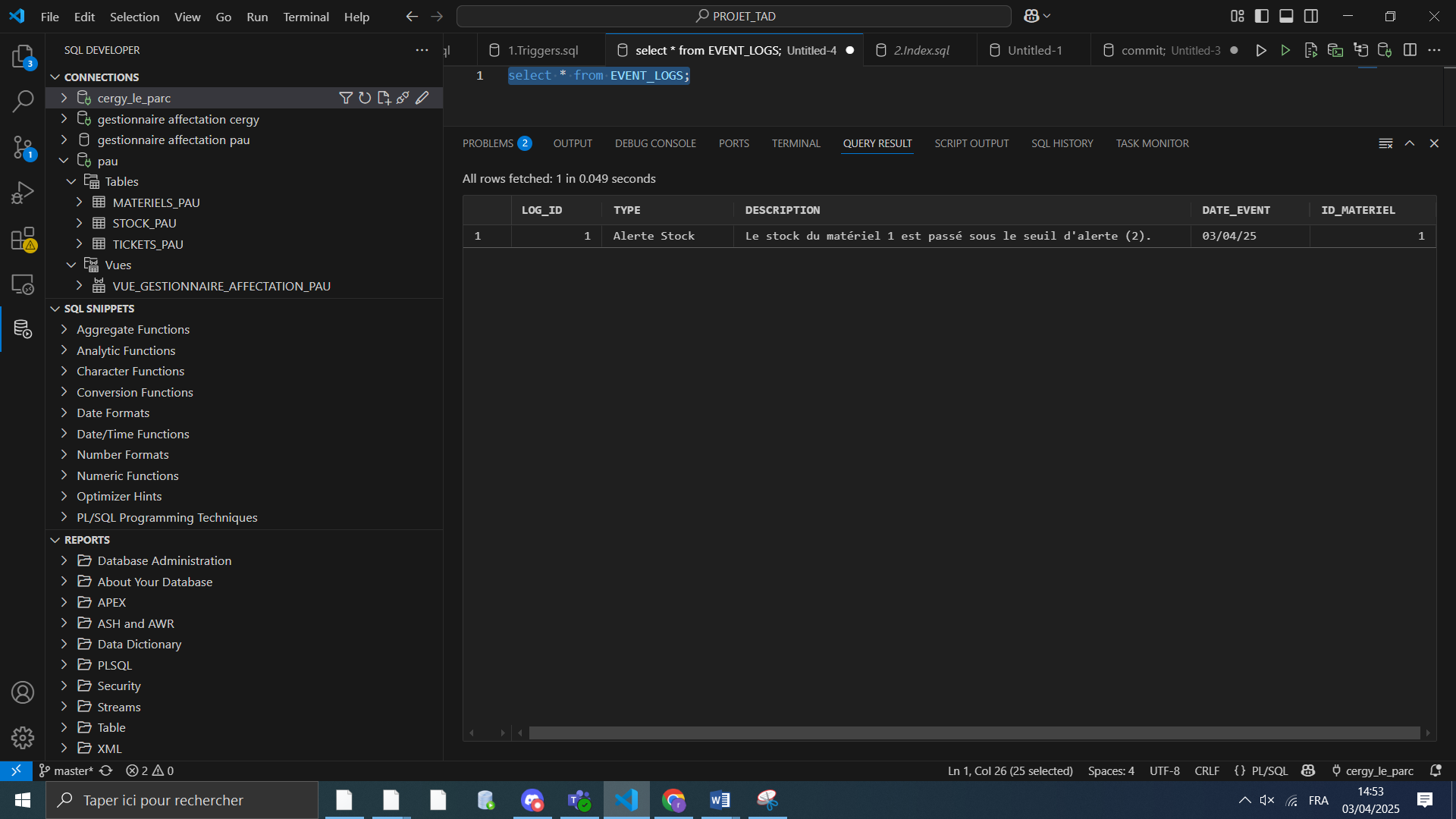
Impossible d’insertion d’une attribution car stock = 0

Ce trigger se déclenche après l'insertion d'une nouvelle attribution de matériel. Il vérifie si la quantité du matériel est suffisante et met à jour la table Stock en conséquence. Si le stock est insuffisant, il génère une erreur.

**Trigger - Seuil d'alerte de stock**



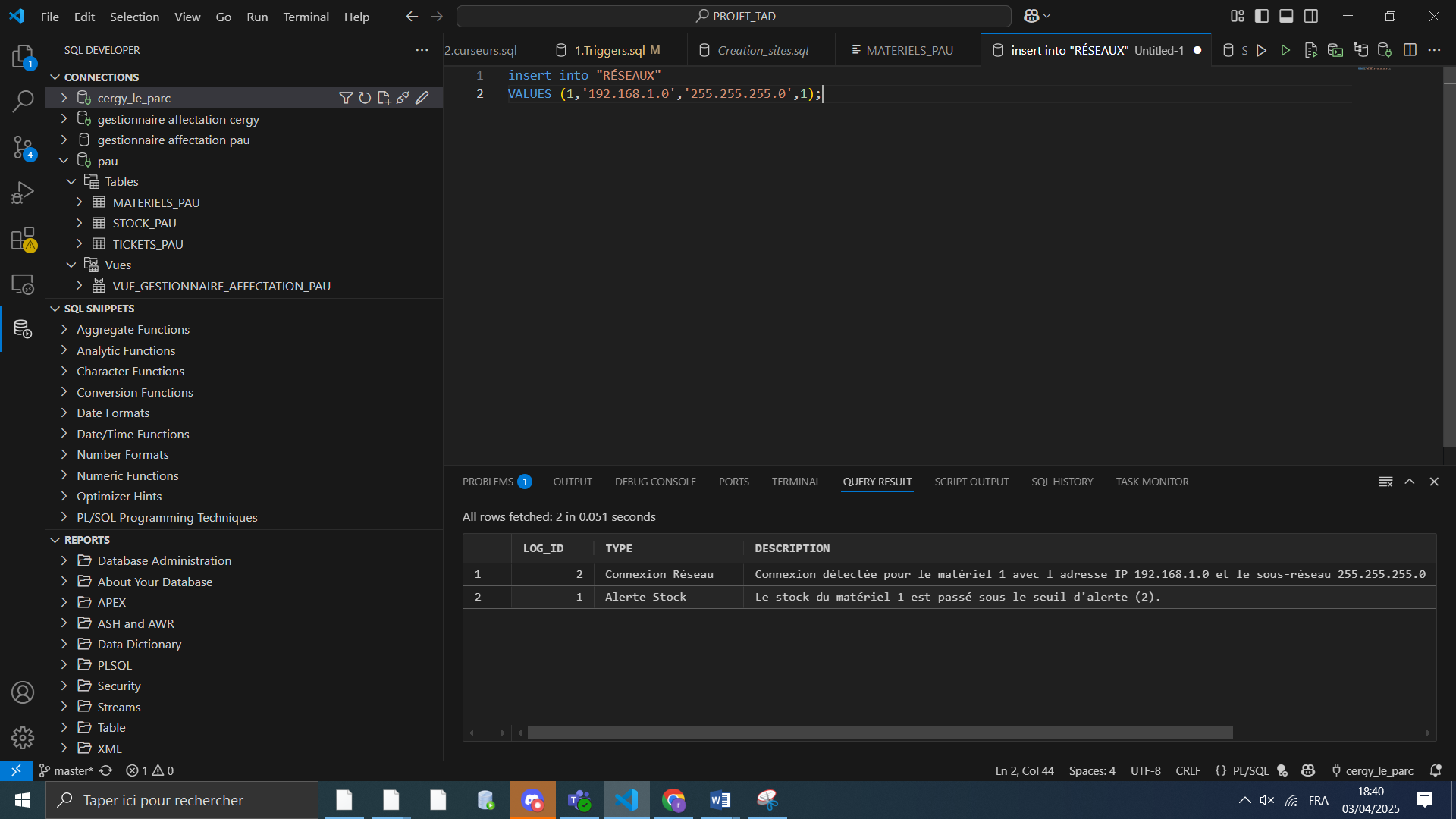
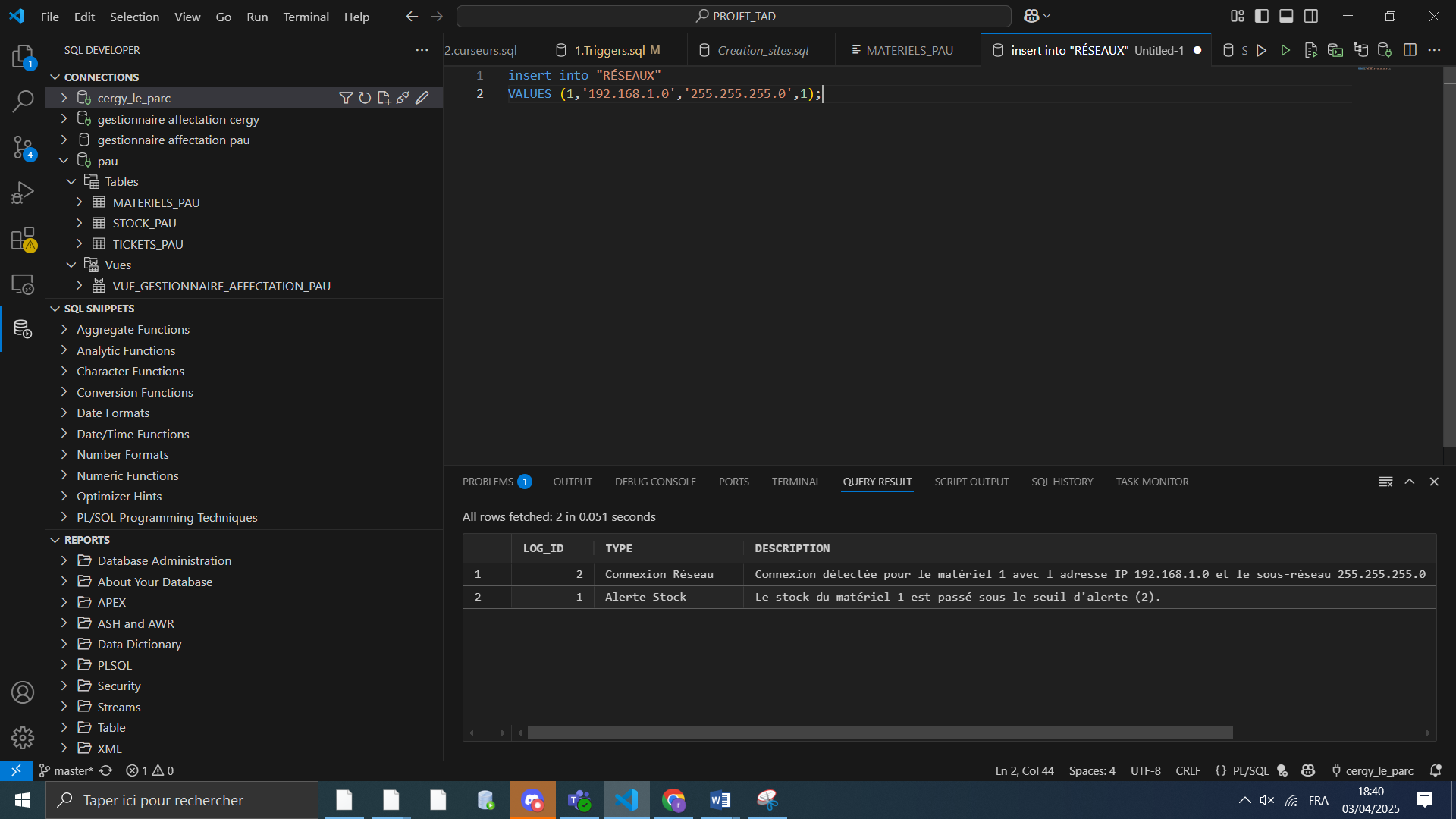
Quantité < Seuil Alerte



Ce trigger s'active lorsqu'un stock est mis à jour et que la quantité devient inférieure au seuil d'alerte. Il insère un événement dans la table **Event\_Logs** pour notifier cette situation.

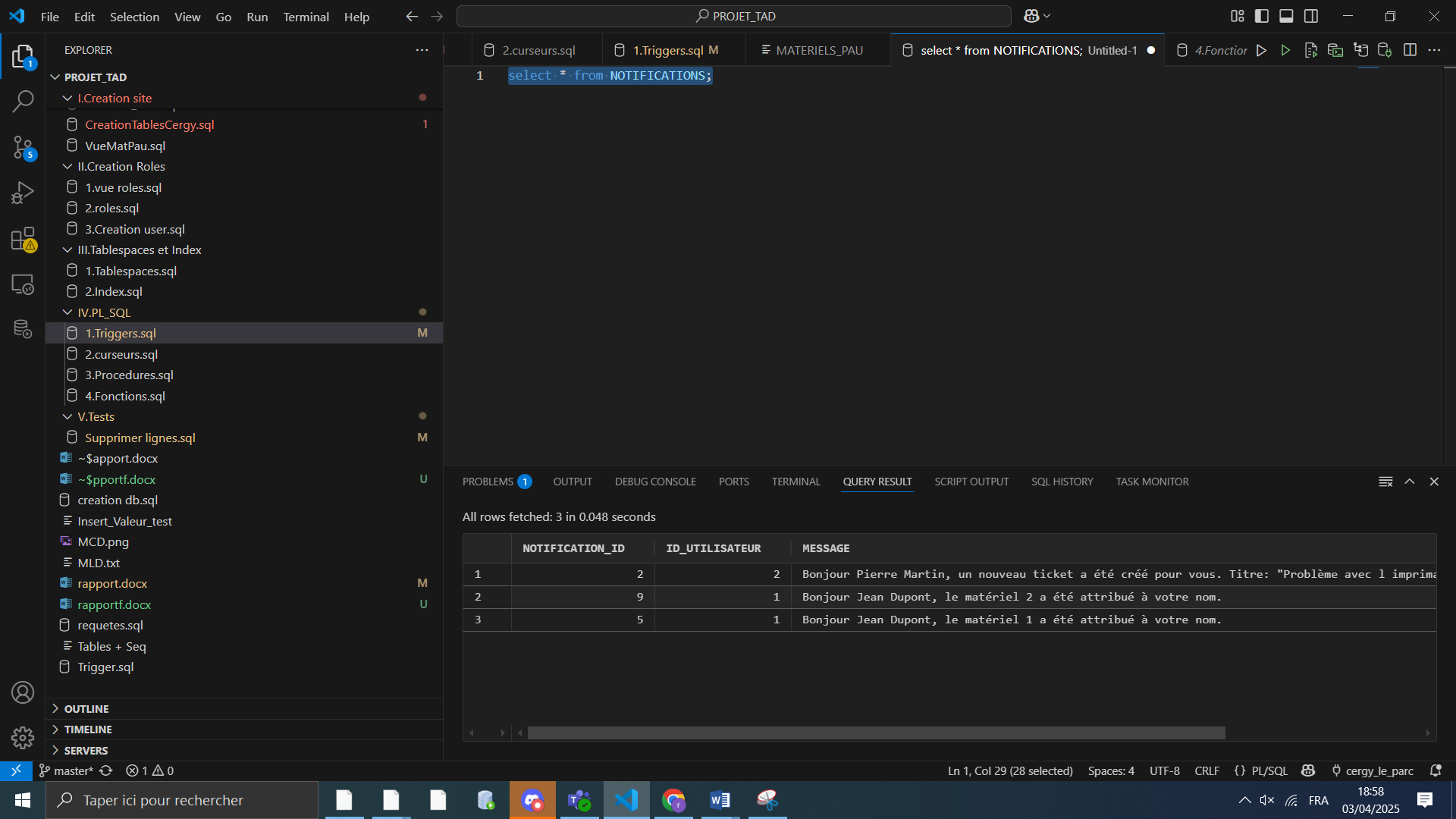
**Trigger - Connexion réseau**

Insertion connexion résaux dans table Réseau => Enregistrement dans Event logs

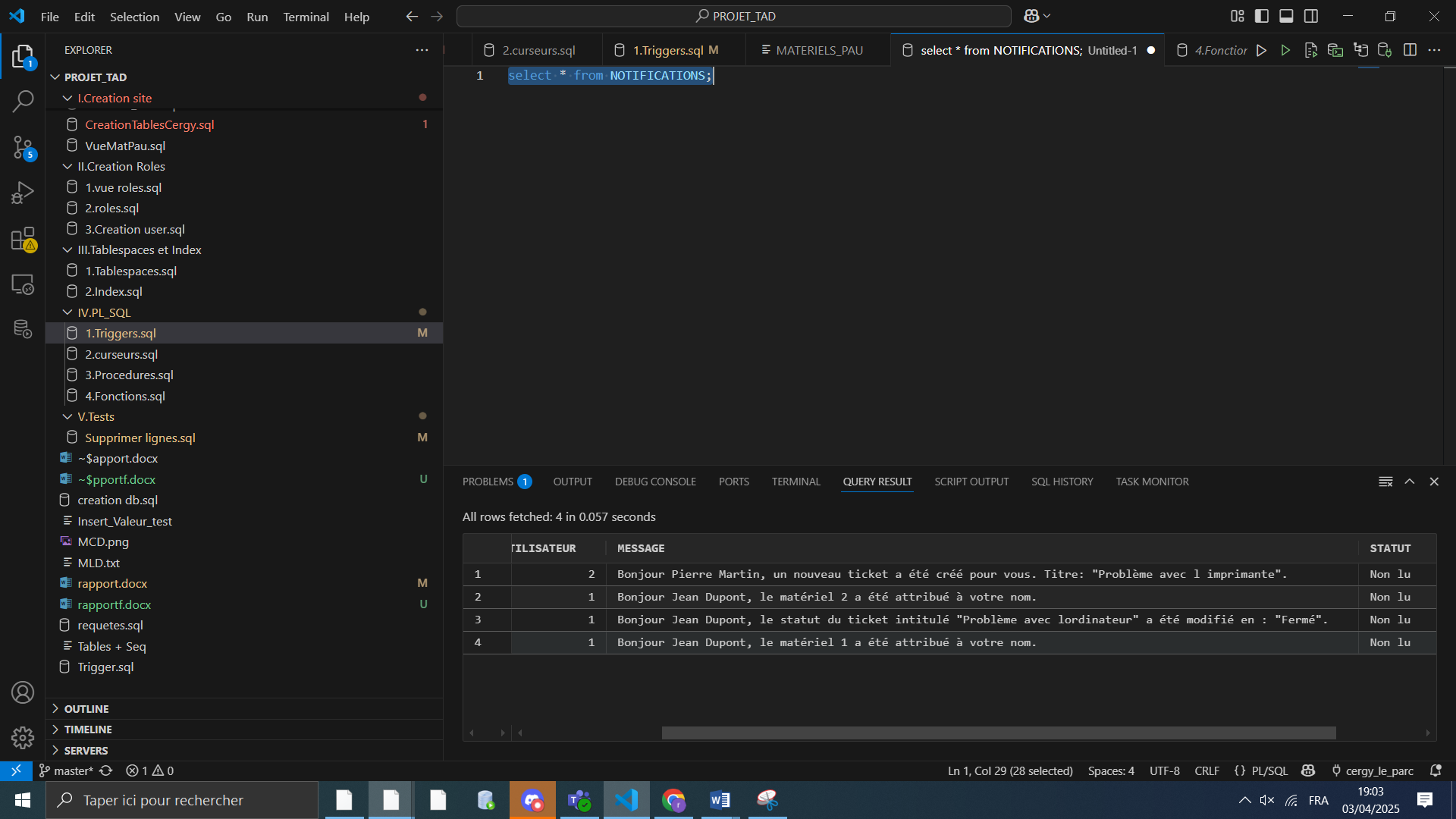


Ce trigger enregistre une connexion réseau dans les logs chaque fois qu'une nouvelle connexion est insérée dans la table Réseaux.

**Trigger - Notification lors d'une attribution de matériel**



Ce trigger envoie une notification à un utilisateur lorsqu'un matériel lui est attribué ou retiré (ajout/suppression d'une ligne dans la table **Attributions**).

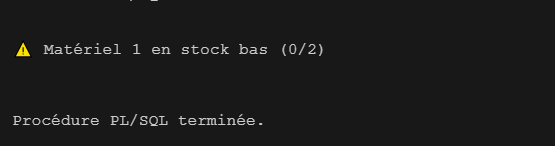
**Trigger - Notification lors d'un changement de ticket**

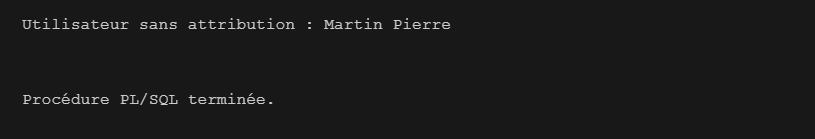
Ce trigger notifie les utilisateurs des changements effectués sur leurs tickets, que ce soit lors de la création, suppression ou mise à jour du ticket.

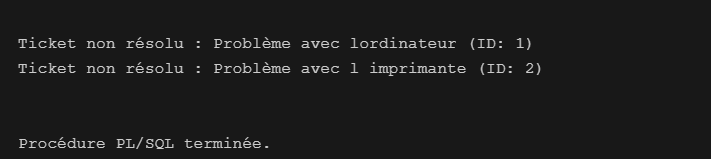
## Curseurs

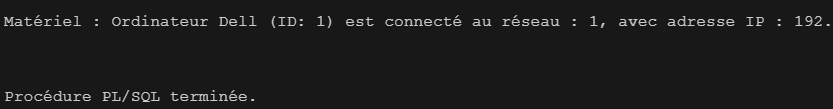
Les curseurs permettent de parcourir les données ligne par ligne dans une requête SQL. Ils sont utiles lorsqu'on doit traiter plusieurs enregistrements un par un. Voici quelques exemples de curseurs utilisés dans notre projet :

**detecter\_stock\_bas** sélectionne tous les matériels dont la quantité en stock est inférieure au seuil d'alerte.



**utilisateurs\_sans\_attribution** permet de récupérer tous les utilisateurs ayant aucun un matériel attribué   
**TICKETS\_NON\_RESOLUS** permet de récupérer les tickets qui ont le statut non résolu



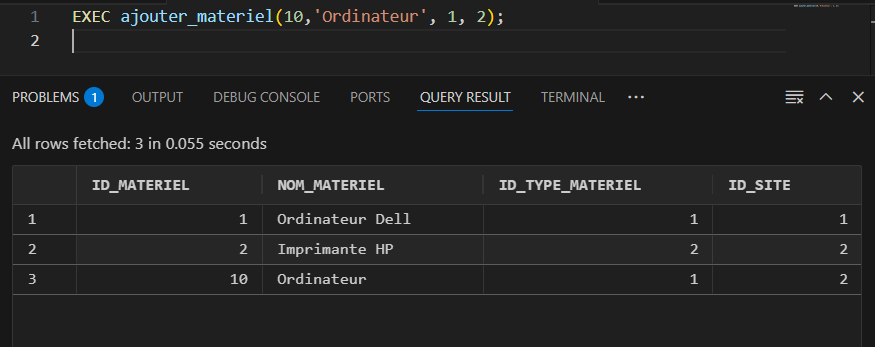
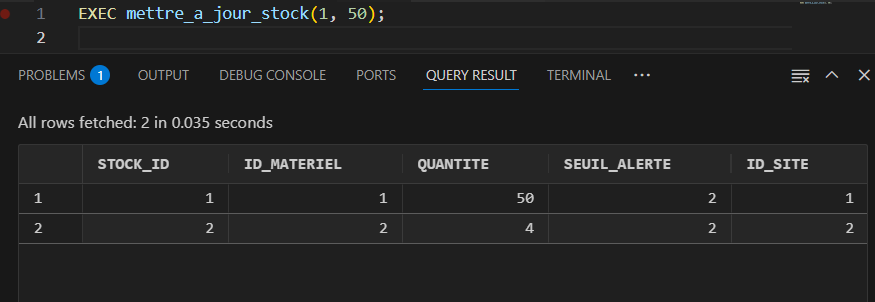
**verifier\_connexions\_reseau** permet de vérifier les connexions réseau d’un matériel.  


## Procédures

Les procédures stockées permettent d'automatiser les opérations répétitives dans la base de données et d'améliorer l'efficacité des transactions.

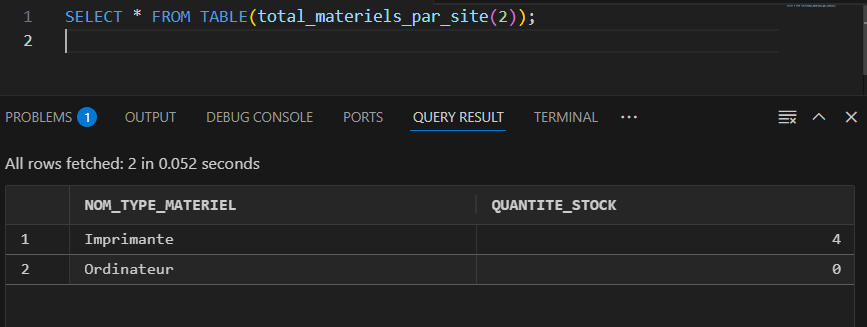
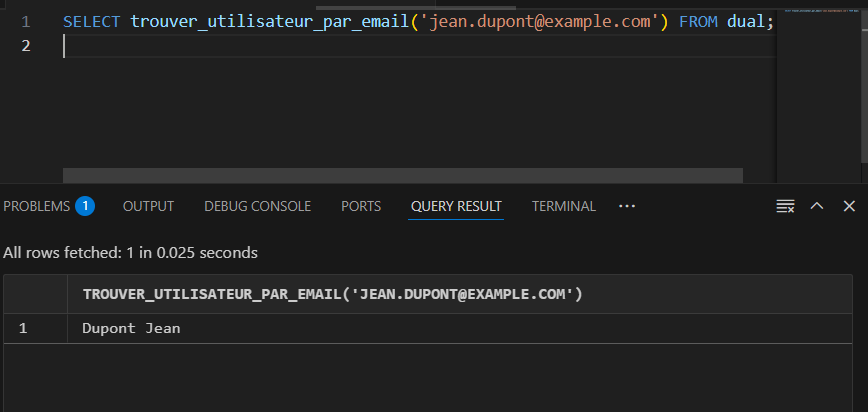
Nous avons mis en place plusieurs procédures essentielles :

**ajouter\_materiel** : Insère un nouveau matériel dans la base, facilitant la gestion des équipements.

**mettre\_a\_jour\_stock** : Met à jour la quantité de matériel disponible, garantissant un suivi précis du stock.  
  


## Fonctions

Les fonctions stockées permettent d'exécuter des calculs et de retourner une valeur unique. Elles sont particulièrement utiles lorsqu’on a besoin d’obtenir un résultat directement utilisable dans une requête SQL.

Dans notre implémentation, nous avons défini les fonctions suivantes :  
**Fonction d’affichage des types de matériels et de leur stock par site : total\_materiels\_par\_site(id\_site)**Cette fonction permet de récupérer, pour un site donné, **la liste des types de matériels disponibles ainsi que leur quantité en stock.  
  
Fonction de recherche d’un utilisateur par email : trouver\_utilisateur\_par\_mail**

Cette fonction retourne le nom et prénom d’un utilisateur à partir de son adresse e-mail.

# Tests

Dans le cadre de l’optimisation et de la validation de la structure de notre base de données, nous avons effectué des tests de performance en simulant un jeu de données conséquent représentant des utilisateurs, des matériels, des tickets, des réseaux, etc. Ces tests ont permis de mesurer la capacité de la base de données à insérer un volume important de données, à exécuter des requêtes complexes et à observer la performance générale de l'infrastructure.

## Préparation du Jeu de Données

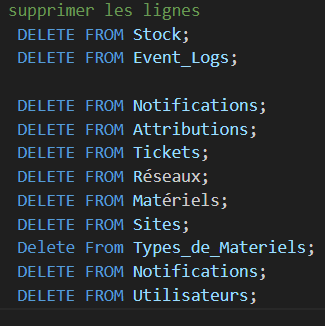
Le jeu de test a été généré en insérant un grand nombre de lignes dans les différentes tables de la base de données. Voici un résumé des données insérées :

* **Utilisateurs** : 400 utilisateurs simulés, générés par un script PL/SQL avec des noms et emails dynamiques.
* **Types de matériels** : 40 types de matériels différents.
* **Sites** : 2 sites (Cergy et Pau), représentatifs des différents lieux d'utilisation des matériels.
* **Matériels** : 100 matériels générés, répartis aléatoirement entre les types et les sites.
* **Stocks** : 100 entrées de stocks liées à ces matériels.
* **Tickets** : 600 tickets générés pour simuler des problèmes matériels et d’utilisation.
* **Connexions Réseau** : 200 connexions réseau liées à différents matériels.

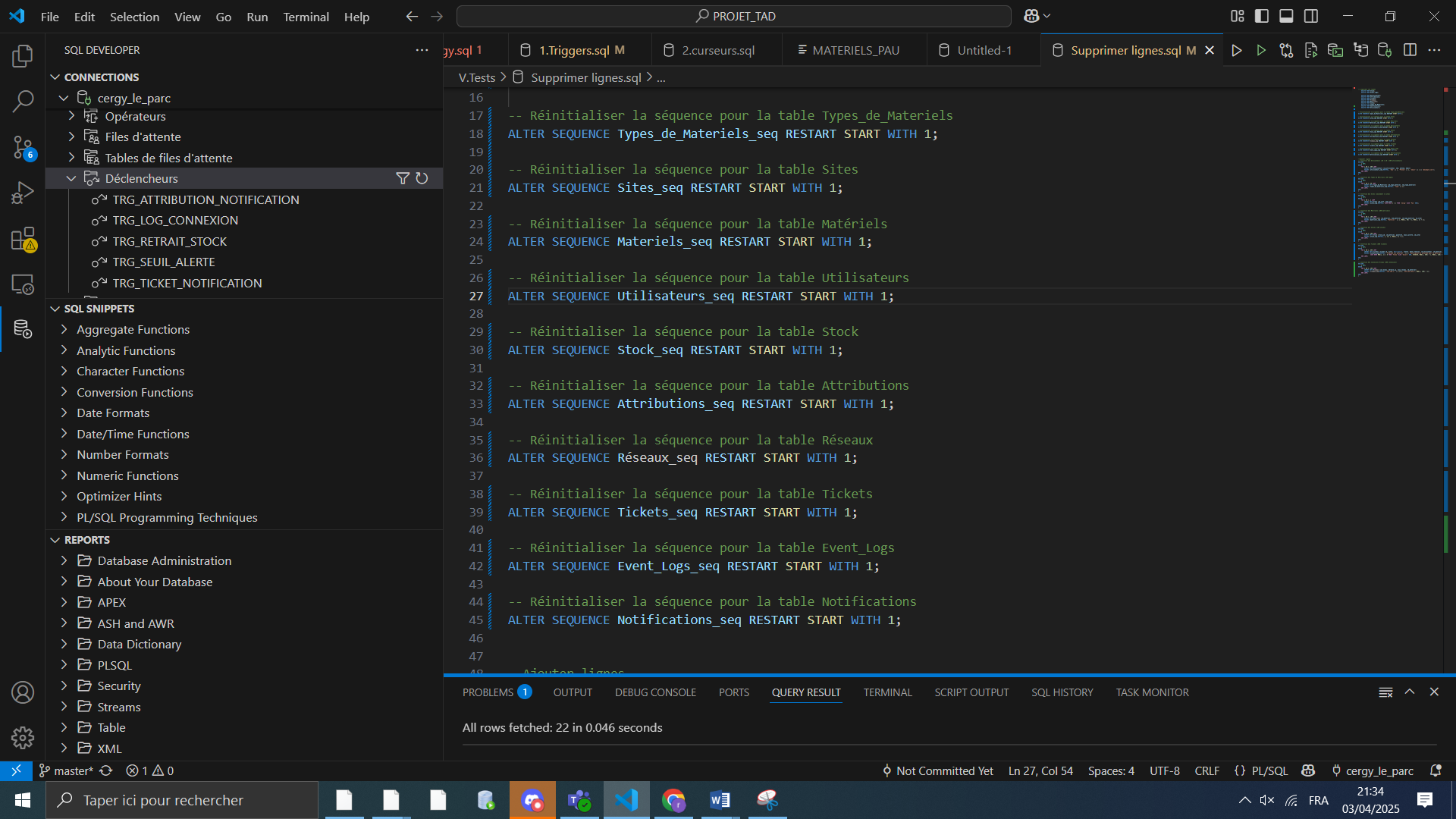
## Exécution des Tests

Les tests ont été réalisés dans un environnement de développement afin d'observer la capacité de la base de données à insérer les lignes en masse. Voici le processus suivi :

**Suppression des données existantes** : Avant de commencer les tests, nous avons supprimé toutes les lignes existantes dans les tables pour garantir que l’insertion soit effectuée sur une base vide. Voici la commande exécutée pour supprimer les anciennes lignes :



**Réinitialisation des séquences :** Ensuite, nous avons réinitialisé les séquences associées à chaque table pour commencer l'insertion à partir de 1. La commande utilisée pour réinitialiser les séquences est la suivante :

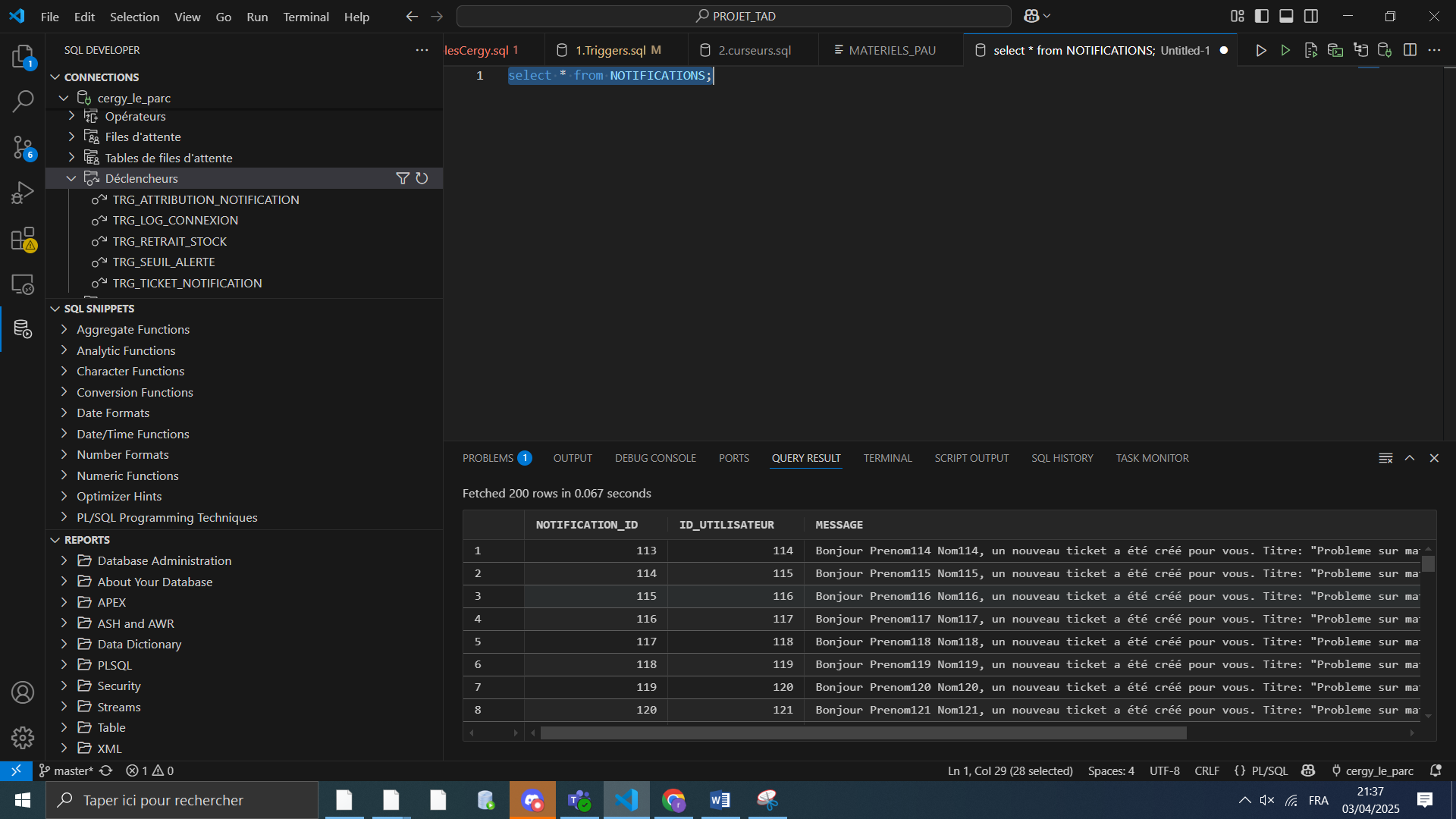


**Insertion des données :** À l’aide de scripts PL/SQL, nous avons inséré les données de manière itérative. Pour chaque entité, les boucles PL/SQL ont été utilisées pour insérer un nombre considérable de lignes (par exemple, 400 utilisateurs, 100 matériels, 600 tickets, etc.).

## Résultats des Tests

**Vitesse d’insertion :** Les tests d’insertion ont montré que le système est capable d’insérer rapidement les données dans la base, même avec un nombre élevé de lignes. Les requêtes d’insertion ont été exécutées sans ralentissement significatif et les données ont été insérées en un temps relativement court. Ces performances indiquent une gestion optimale des ressources de la base de données pour des opérations massives.

**Exécution des Triggers :** Les triggers définis dans la base ont parfaitement fonctionné. Par exemple, les triggers de mise à jour du stock, d’attribution des matériels, et de gestion des tickets ont été exécutés correctement, sans aucune erreur ou incohérence. Les tests ont permis de s'assurer que ces triggers étaient bien intégrés dans le flux de travail des opérations de la base de données et qu'ils ont contribué à maintenir l’intégrité des données.



Les tests de performance ont permis de valider la capacité de la base de données à gérer un volume conséquent de données tout en maintenant de bonnes performances d'insertion et d'exécution de requêtes. Les triggers ont fonctionné comme prévu et ont assuré une intégrité des données sans ralentir les opérations.

Les résultats montrent que l’infrastructure est bien dimensionnée pour gérer des jeux de données importants, et qu’elle est prête à être utilisée dans un environnement de production avec une charge élevée.

# Conclusion

Ce projet de refonte du système de gestion du parc informatique de CY Tech, basé sur la plateforme GLPI, a permis de mettre en place une solution robuste et optimisée, adaptée aux spécificités des deux sites de l’établissement, à savoir Cergy et Pau. L’organisation de la gestion multisite a été réussie, grâce à une centralisation efficace des données et à une fragmentation judicieuse, garantissant ainsi une synchronisation parfaite entre les sites. La modélisation conceptuelle et logique des données a permis de structurer les informations relatives aux utilisateurs, matériels, tickets, et stocks de manière cohérente, tout en assurant une gestion fluide des ressources et une traçabilité optimale.

La mise en place de mécanismes avancés, tels que les triggers, les procédures stockées, et les vues matérialisées, a contribué à l’automatisation des processus et à l’amélioration des performances du système. La création des rôles et des vues associées a également renforcé la sécurité et facilité l’accès aux données selon les besoins des utilisateurs, tout en garantissant une gestion fine des droits d’accès.

Les index et tablespaces ont été soigneusement choisis pour optimiser les performances et la gestion de l’espace de stockage, permettant ainsi une exécution rapide des requêtes et une gestion optimale de la base de données.

Les tests de performance réalisés ont montré que la base de données est capable de gérer efficacement des volumes importants de données et de maintenir une bonne réactivité même avec des charges élevées. L'exécution des triggers et des procédures a été confirmée sans erreurs, assurant ainsi l'intégrité des données tout au long du processus.

En conclusion, la solution mise en place répond aux besoins de gestion et de performance de l’infrastructure informatique de CY Tech. Elle est prête à évoluer pour supporter une augmentation future du volume de données et d’utilisateurs. Ce projet a non seulement amélioré l’architecture de la base de données, mais a également posé les bases d’une gestion optimisée et sécurisée du parc informatique sur les deux sites, Cergy et Pau.