2023-
2024

Projet INFO0505

COMPTE-RENDU DE PROJET

ANTOINE THEOLOGIEN, ROMAIN DUPONT

Sommaire

Introduction

Nous sommes chargés de gérer différents laboratoires de recherche. Afin d'informatiser cette gestion et de gérer les différents flux d'informations et de données transitant dans le système, il faut mettre en place une base de données.

Ce rapport détaillera les différents moyens mis en places pour respecter les contraintes émises au sein du cahier des charges, afin d'avoir une base de données utilisable, permettant de récupérer différentes informations, comme la liste des expériences, ou encore la liste des commandes pour chaque produit.

Des fichiers png sont disponibles dans l'archive afin de faciliter la lecture de ceux-ci.

Enfin, les scripts de création, d'insertion et de suppression de données, ainsi que les requêtes SQL et les programmes PL/SQL sont disponibles dans le dossier « fichiers_bdd ».

Cahier des charges

Nous avons donc pu établir un cahier des charges afin de mettre en place au mieux cette base de données. Un laboratoire doit pouvoir engager différents chercheurs, qui peuvent en revanche avoir d'autres activités en dehors de leur activité de chercheur.

Au sein du laboratoire, les chercheurs sont divisés en plusieurs équipes, qui mènent alors des projets et ont recours à différentes expériences afin d'obtenir différents résultats.

Ces expériences nécessitent alors différents produits, pouvant être des consommables, des enzyme, du matériel de biologie moléculaire, des produits chimiques, ou encore des cultures cellulaires. Chaque type de produit possède ses propres caractéristique, nécessitant de créer une entité par catégorie de produit.

Pour gérer le stock de ces produits, il faut qu'un laboratoire puisse commander ceux-ci, auprès d'un fournisseur. Le fournisseur peut être mis en relation avec plusieurs laboratoires, sans souci de concurrence. Pour gérer les quantités d'un produit disponible, nous avons pris l'initiative de faire figurer celle-ci dans l'entité produit, celui-ci possédant une clé étrangère vers un sous-ensemble.

De plus, afin de faciliter l'organisation au sein d'un laboratoire, un service de réservation doit être mis en place, pour gérer les produits principalement. Ainsi, un chercheur fait une réservation d'une salle et des produits qu'il compte utiliser.

A cela s'ajoute un système de réunion, permettant à des binômes de chercheurs de montrer l'avancement de leurs recherches.

Enfin, comme précisé dans l'énoncé, chaque laboratoire possède une animalerie contenant des souris, qui servent à mettre en place un PEA. Ces éléments doivent donc également apparaitre dans notre système de base de données.

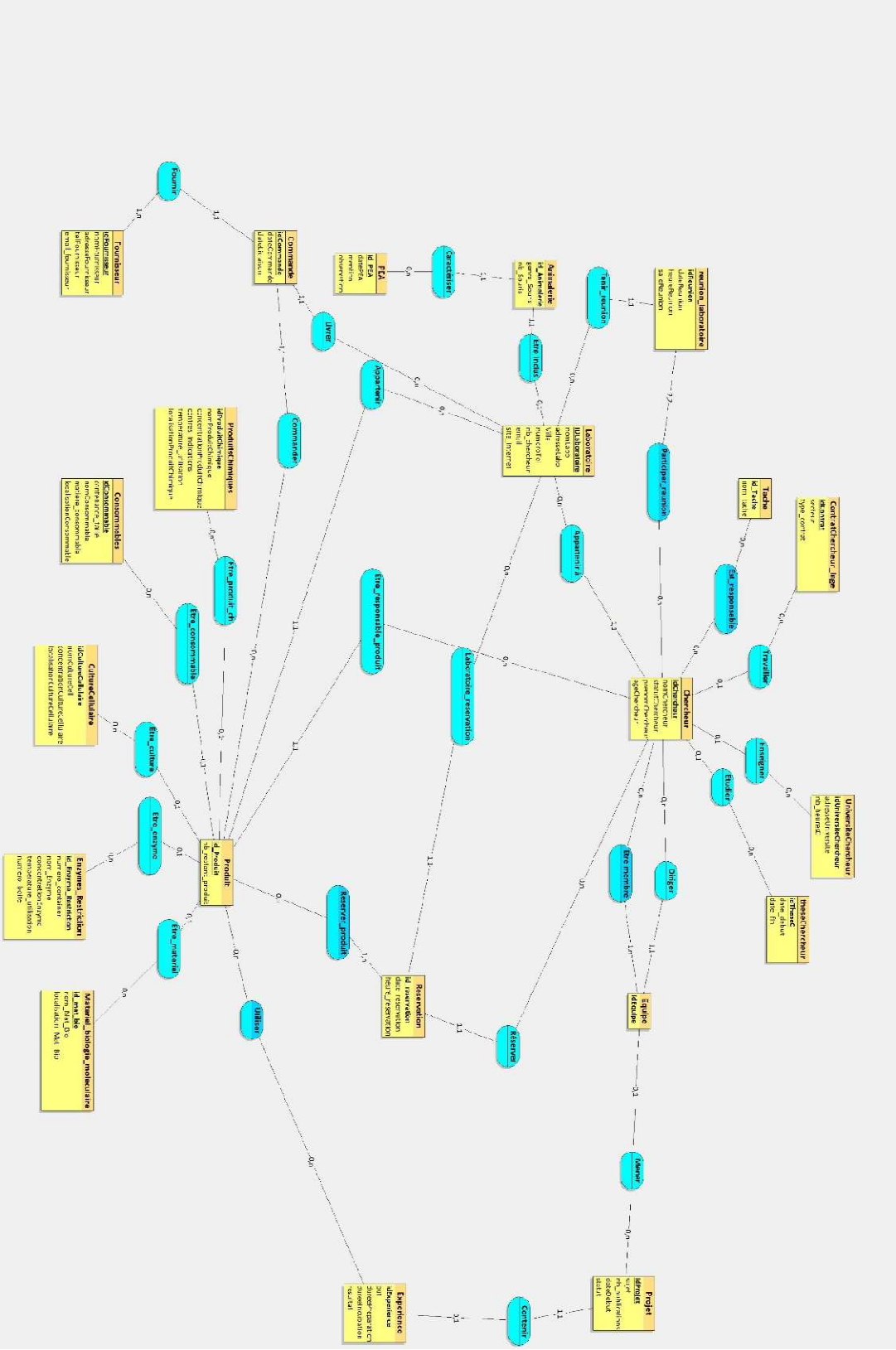
Afin de construire cette base de données, nous suivrons donc la méthode **MERISE**.

Matrice de Flux

Pour la matrice de flux, nous avons pu identifier 4 acteurs : chercheur, fournisseur, responsable équipe, et responsable produit.

->	Chercheur	Fournisseur	Responsable Equipe	Responsable Produit
Chercheur	----	---	-Demander de rejoindre l'équipe	-Réserver un produit
Fournisseur	--	---		-Livrer des produits
Responsable Equipe	Accepter/refuser demande de rejoindre l'équipe	---	----	-
Responsable Produit	Valider réservation	Réceptionner la commande	-	----

Modèle Conceptuel de données



Modèle relationnel de données

Laboratoire = (IDLaboratoire *INT*, nomLabo *VARCHAR(20)*, adresseLabo *VARCHAR(50)*, Ville *VARCHAR(20)*, numeroTel *VARCHAR(10)*, nb_chercheur *INT*, email *VARCHAR(50)*, site_internet *VARCHAR(50)*);

UniversiteChercheur = (idUniversiteChercheur *INT*, adresseUniversite *VARCHAR(50)*, nb_heuresC *INT*);

theseChercheur = (idTheseC *INT*, date_debut *DATE*, date_fin *DATE*);

ContratChercheur_Inge = (idContrat *INT*, secteur *VARCHAR(20)*, type_contrat *VARCHAR(20)*);

Experience = (idExperience *INT*, but *VARCHAR(50)*, dureePreparation *VARCHAR(50)*, dureeIncubation *VARCHAR(50)*, resultat *VARCHAR(50)*);

Projet = (idProjet *INT*, sujet *VARCHAR(50)*, nb_publications *INT*, dateDebut *DATE*, statut *VARCHAR(20)*, ***#idExperience***);

CultureCellulaire = (idCultureCellulaire *INT*, nomCultureCell *VARCHAR(50)*, concentrationCultureCellulaire *VARCHAR(50)*, localisationCultureCellulaire *VARCHAR(50)*);

ProduitsChimiques = (idProduitChimique *INT*, nomProduitChimique *VARCHAR(50)*, concentrationProduitChimique *VARCHAR(50)*, contres_indications *VARCHAR(50)*, temperature_utilisation *VARCHAR(50)*, localisationProduitChimique *VARCHAR(50)*);

Consommables = (idConsommable *INT*, contenance_taille *VARCHAR(50)*, nomConsommable *VARCHAR(50)*, matière_consommable *VARCHAR(50)*, localisationConsommable *VARCHAR(50)*);

Materiel_biologie_moleculaire = (id_mat_bio *INT*, nom_Mat_Bio *VARCHAR(50)*, localisation_Mat_Bio *VARCHAR(50)*);

Fournisseur = (idFournisseur *INT*, nomFournisseur *VARCHAR(50)*, adresseFournisseur *VARCHAR(50)*, telFournisseur *VARCHAR(50)*, email_fournisseur *VARCHAR(50)*);

PEA = (id_PEA *INT*, datePEA *DATE*, mention *VARCHAR(50)*, observation *VARCHAR(50)*);

reunion_laboratoire = (idReunion *INT*, dateReunion *DATE*, heureReunion *DATE*, salleReunion *VARCHAR(50)*, *#IDLaboratoire*);

Enzymes_Restriction = (id_Enzyme_Restriction *INT*, numero_container *INT*, nom_Enzyme *VARCHAR(50)*, concentrationEnzyme *VARCHAR(50)*, temperature_utilisation *VARCHAR(20)*, numero_boite *INT*);

Tache = (id_Tache *INT*, nom_tache *VARCHAR(50)*);

Chercheur = (idChercheur *INT*, nomChercheur *VARCHAR(20)*, statutChercheur *VARCHAR(20)*, prenomChercheur *VARCHAR(20)*, ageChercheur *INT, #IDLaboratoire, #idContrat*, #idTheseC*, #idUniversiteChercheur**);

Equipe = (idEquipe *INT, #idProjet*, #idChercheur*);

Animalerie = (id_Animalerie *INT*, genre_Souris *VARCHAR(10)*, nb_Souris *INT, #id_PEA, **#IDLaboratoire***);

Produit = (id_Produit *INT*, nb_restant_produit *INT, #idChercheur, #IDLaboratoire, #idProduitChimique*, #id_Enzyme_Restriction*, #id_mat_bio*, #idCultureCellulaire*, #idConsommable**);

Reservation = (id_reservation *INT*, date_reservation *DATE*, heure_reservation *DATE, #IDLaboratoire, #idChercheur*);

Commande = (idCommande *INT*, dateCommande *DATE*, dateLivraison *DATE, #IDLaboratoire, #idFournisseur, #id_Produit*);

Reserver_produit = (*#id_Produit, #id_reservation*);

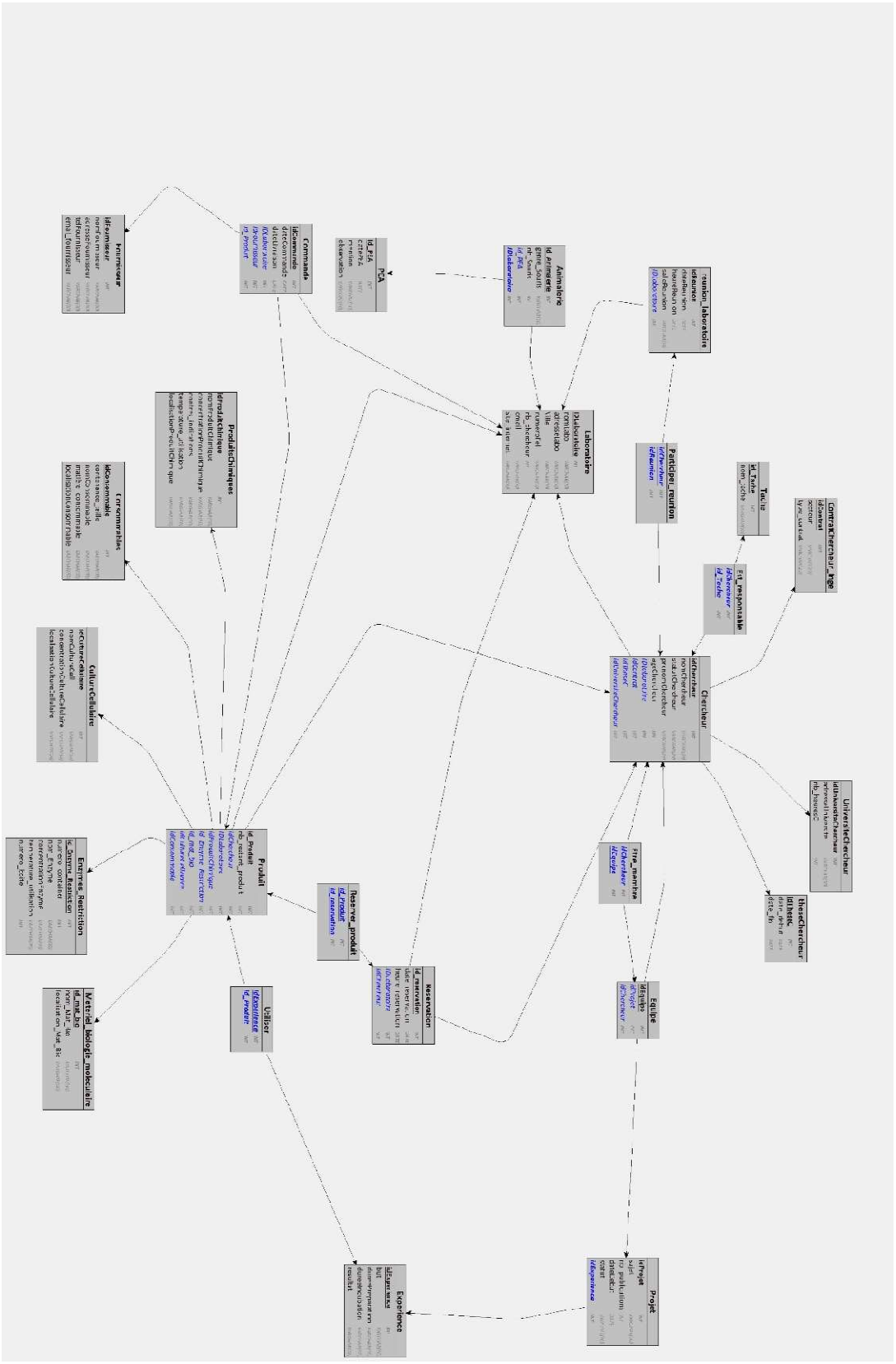
Participer_reunion = (*#idChercheur, #idReunion*);

Utiliser = (*#idExperience, #id_Produit*);

Etre_membre = (*#idChercheur, #idEquipe*);

Est_responsable = (*#idChercheur, #id_Tache*);

Modèle Logique et Physique de données



Normalisation

La normalisation consiste à convertir les différentes tables d'une base de données relationnelle afin qu'elles respectent les différents degrés de normalisation. Ils sont aux nombres de 3, et possèdent chacun des critères différents. Cela permet d'éviter les redondances au sein de la base. En ayant suivi la méthode de conception **MERISE**, notre base de données est normalement bien en 3NF, le plus haut degré de normalisation.

On peut donc vérifier que notre base est bel et bien en 3NF :

1NF : Tous les attributs de la base de données sont atomiques, nous sommes donc à minima en 1NF.

2NF : Dans toutes nos relations, aucun des attributs n'appartient à une partie de la clé, nous sommes donc à minima en 2NF.

3NF : Enfin, dans toutes nos relations, aucun des attributs ne dépend d'un autre attribut non clé, nous sommes donc bien en 3NF.

On peut prendre comme exemple la table Chercheur :

On a : Chercheur(idChercheur,nomChercheur,prenomChercheur,statutChercheur,ageChercheur) ;

On a comme dépendances fonctionnelles :

F={ idChercheur->nomChercheur
idChercheur->prenomChercheur
idChercheur->statutChercheur
idChercheur->ageChercheur
nomChercheur, prenomChercheur, statutChercheur, ageChercheur ->idP}

En effet, idChercheur s'obtient grâce à tous les autres attributs de l'entité chercheur, et permet d'obtenir chacun des attributs.

Nous sommes donc bien à minima en 1NF, puisque ces attributs sont tous atomiques.

On vérifie que idChercheur est un attribut clé :

Pour cela, on calcule la fermeture transitive de idP, c'est-à-dire idP^+. On a alors :

idChercheur + {idChercheur, nomChercheur, prenomChercheur, statutChercheur, ageChercheur }

On obtient bien tous les attributs donc idChercheur est clé. De plus, aucun des attributs de professeur ne dépend d'une partie de la clé, idChercheur n' étant qu'en une seule partie. Nous sommes donc bien en 2NF.

Enfin, nous n'avons aucun attribut non clé qui ne dépend d'un autre attribut que idChercheur, c'est-à-dire un attribut non clé, par conséquent, nous sommes bien en 3NF.

SQL-PL/SQL

Cette partie contient les différents résultats des requêtes SQL et des programmes PL/SQL. Les scripts sont accessibles dans le dossier « fichier_bdd ».

Requêtes SQL :

1 : La liste des produits commandés par le labo 1 au fournisseur "Fournitout".

ID_PRODUIT
1
4
0

2 : Les dates et heures de réunion du chercheur Clément Turpin (emploi du temps)

DATEREUNION	HEUREREUNION
24-NOV-23	16:00
01-DEC-23	16:00

3 : La liste des produits utilisés dans l'expérience du projet mené par l'équipe dont la chercheur Josiane Cardin est membre

NOMCONSOMMABLE	NOM_MAT_BIO	NOMPRODUITCHIMIQUE
Gants	Table UV	Acide Chlorhydrique

4. La liste des fournisseurs ayant fourni tous les laboratoires :

NOMFOURNISSEUR
Fournitout
Fournipresquetout

5. La liste des produits fournis par un fournisseur dont le numéro de téléphone contient en 7 :

ID_PRODUIT
1
4
0

6. Les chercheurs ingénieurs faisant parti des équipes dirigés par Beatrice Hector :

NOMCHERCHEUR	PRENOMCHERCHEUR
Robinet	Aude
Beaulne	Quentin
Chuquet	Constance
Devilliers	Nina

7. Le nombre de chercheurs qui ont une réunion de programmée le 24 novembre 2023 :

COUNT(IDCHERCHEUR)
4

8. La moyenne des quantités des produits disponibles au sein du labo 2 :

AVG(NB_RESTANT_PRODUIT)
18.6

Programmes PL/SQL :

1. Afficher les prénoms et noms des chercheurs du laboratoire dont l'identifiant est 1 :

```
Statement processed.
Le chercheur Hector se prénomme : Beatrice.
Le chercheur Chuquet se prénomme : Constance.
Le chercheur Devilliers se prénomme : Nina.
Le chercheur Beaulne se prénomme : Quentin.
Le chercheur Robinet se prénomme : Aude.
Le chercheur Kalubi se prénomme : William.
Le chercheur Aubert se prénomme : Amy.
Le chercheur Chaumont se prénomme : Bernard.
Le chercheur Martin se prénomme : Alain.
Le chercheur Bier se prénomme : Paul.
```

2. Mettre à jour le nombre de chercheurs au sein des laboratoires :

```
Statement processed.
Nombre de chercheurs mis à jour : 20
```

Le nombre affiché correspond au nombre de chercheurs après la mise à jour.

3. Ajouter un chercheur sans profession au labo 1 :

```
Statement processed.
Chercheur ajouté avec succès.
Nombre de chercheurs mis à jour : 21
```

4. Trigger permettant de diminuer la quantité disponible d'un produit après que celui-ci est été réservé :

```
1 row(s) inserted.
Quantité réservée mise à jour pour le produit : 19
```

Lors de l'insertion dans la table réserver_produit, il y a une mise à jour de la quantité de produit, on affiche donc cette nouvelle quantité.

5. Afficher les détails de la profession d'un chercheur en fonction de sa profession :

```
Statement processed.
La profession du chercheur est : Professeur
Détails du contrat professeur :
ID du contrat professeur : 0
Adresse de l'université : Rue des Crayères, 51100 Reims
Nombre d'heures enseignées :  20
```

Résultat pour le chercheur dont l'identifiant est le numéro 19. Cela fonctionne également pour les ingénieurs ou les étudiants :

```
Statement processed.
La profession du chercheur est : Ingénieur
Détails du contrat :
ID du contrat : 0
Secteur :Santé
Type de contrat :CDI
```

```
Statement processed.
La profession du chercheur est : Etudiant
Détails de la thèse :
ID de la thèse : 0
Date de début de la thèse : 01-SEP-23
Date de fin estimée de la thèse : 01-SEP-26
```