BUT Science des données

Projet SAE - Base de Données

Encadré par Mme Imen Chtourou

Résumé

Le projet a été mené du 12/3/2024 au 26/3/2024

Minh
Mathis
Ali Ben Said

BUT Science des données

Table de Matières

Table de Figures

Table Assignation des taches

1. Contexte : (Collaboration)
2. Objet : (Collaboration)
3. Schéma E/A : (Collaboration)
4. La transition MLD : (Collaboration)
5. Les contraintes : (Minh)
6. La création de la BdD : (Minh)
7. Le remplissage de la BdD : (Minh)
8. Les requêtes : (Collaboration)

1. Contexte :

Gérer une grande quantité de données relatives aux clients, aux employés, aux gares, etc., devient difficile, il est donc nécessaire de créer une base de données optimisée au maximum. On participe à sa construction et à sa gestion, cette dernière est destinée à la RATP (Régie Autonome des Transport Parisiens) au milieu des années 90. Consultez-vous sur https://www.ratp.fr/

2. Objet :

Rendre la base de données le plus accessible et facile à interroger possible en optimisant le nombre de tables nécessaires ainsi que les relations.

3. Schéma E/A :

a. Schéma Initial : (Les Tables seront colorées en jaune)

Nous avons commencé par créer les 5 tables principales de ce modèle. Ce sont Matériel, Personnel, Usagers, Abonnement, Stations.

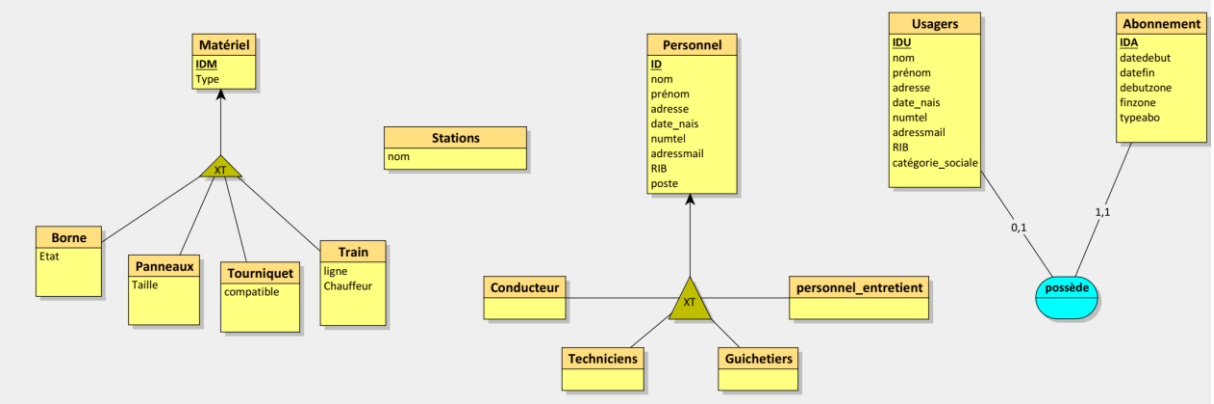

Figure 1: le schéma initial

- Les tables Personnel et Usagers contiennent des champs tels que « nom, prénom, adresse, date_naiss, numtel, adressmail, RIB » permettant de connaître l'identité des employés de l'entreprise ainsi que de ses clients en stockant leurs informations personnelles. De plus, chaque table possède sa propre clé primaire. En outre, la table Personnel possède également le champ poste pour enregistrer les postes occupés par le personnel, cela permet de savoir quel métier fait chaque employé, tandis que la table Usagers possède le champ « catégorie_sociale ».

- La table Personnel ne pouvant pas contenir toutes les informations trop nombreuses et trop diverses qui étaient demandés, nous avons donc créé des sous tables pour pouvoir respecter toutes les conditions : Techniciens, Conducteurs, Guichetiers, Controleurs (ajouté plus tard) et personnel_entretient en utilisant "héritage" comme lien car ce sont les cinq postes possibles, ces tables sont reliées à la table mère, elles récupèrent donc la clé primaire de la table mère qui agira en clé étrangères pour matérialiser le lien.

- Nous utilisons une association "Possède" pour relier Usagers à Abonnement, de cardinalité 0,1 en partant de Usagers, car un client ne peut avoir qu'un seul abonnement maximum, mais il peut également frauder et voyager sans abonnement, d'où le poste Controleurs. En revanche, dans l'autre sens, la cardinalité est de 1,1 car un Abonnement ne peut exister sans client. La table Abonnement possède des champs permettant de connaître le type de l'abonnement (ancien ou nouveau "Parigo") mais également la durée et la zone d'action de ce dernier.

- La table Matériel comme table principale, elle comprend le champ IDM comme clé principale, et le champ « type » stockera les différents types de matériel possibles.
- Pour les mêmes raisons que la table Personnel, nous avons dû créer des sous-tables de la table Matériel qui sont Borne, Panneaux, Tourniquets et Train étant les quatre types de matériels possibles dans le champ 'type' de la table Matériel, la clé primaire de Matériel agira comme une clé étrangère reliant les tables à la table mère. En outre, La table Borne contient le champ « etat » pour enregistrer l'état et ainsi savoir si elle est ancienne ou récente. La table Panneaux contient le champ « taille » pour la taille des panneaux (petit, moyen, grand). Pour la table Tourniquet, il y a une colonne « compatible » pour la nouvelle carte Parigo, et dans la table Train, il y a des variables telles que « ligne » et « chauffeur » (car certains trains ne nécessitent pas de chauffeur).

- Pour le moment, la table Stations ne possède que le champ pour renseigner son nom.

b. Schéma Final :

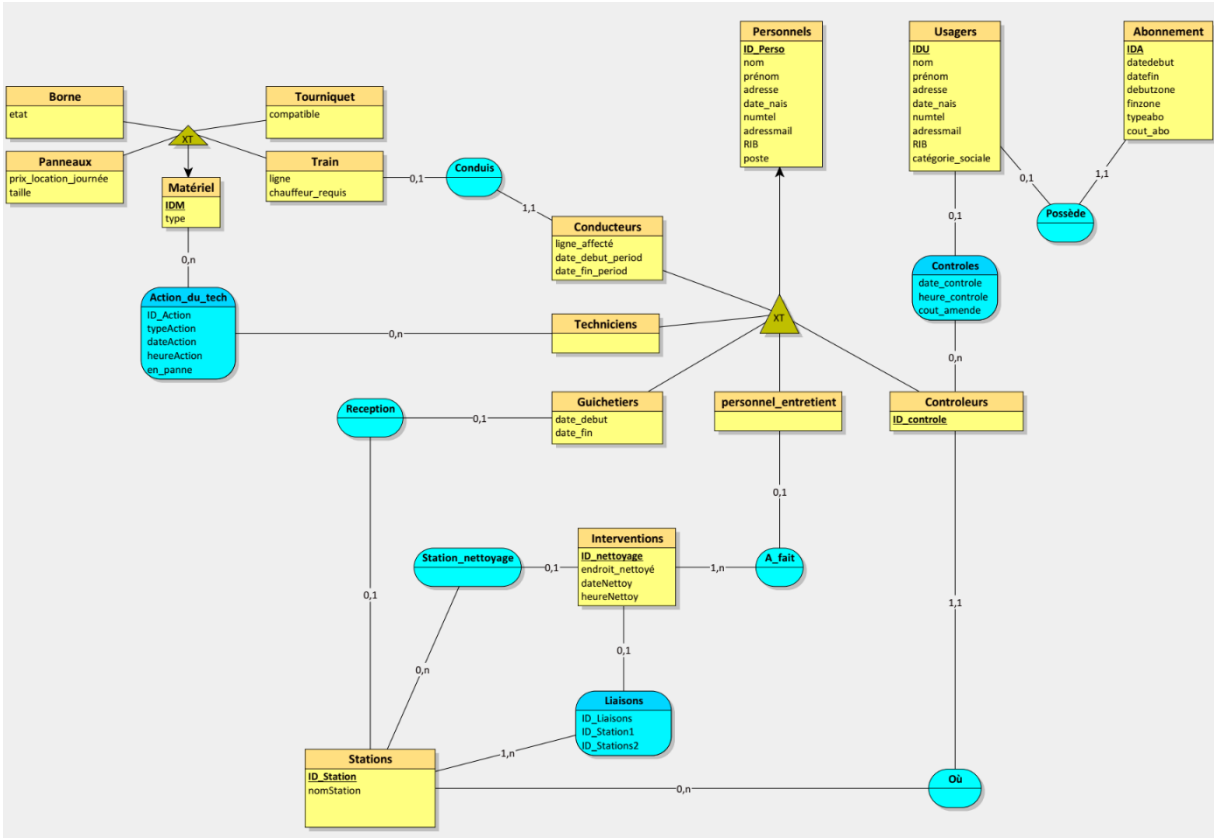

Figure 2:le schéma final

- Le champ « chauffeur » de la table Train a été optimisé en « chauffeur_requis ». Cela permet aux responsables qui ont droit d'accès à la base de données de pouvoir bien comprends cette colonne.

- Nous avons ajouté le champ « cout_abo » dans Abonnement pour renseigner le prix de l'abonnement. Ainsi que « prix_location_journée » dans panneaux pour les mêmes raisons.

❖ Problèmes restants qu'il fallait résoudre :

1. Comment lier les conducteurs aux trains dont ils sont responsables, et inversement ?

Nous associons la table Train avec la table Conducteurs à travers une association nommée « conduis » avec les cardinalités respectives de 0,1 (car un train peut avoir ou non un chauffeur, mais au maximum un seul chauffeur) et 1,1 (car il est dit que chaque chauffeur aura son train associé à un moment donné). C'est pourquoi nous ajoutons les champs « date_debut_period, date_fin_period » ainsi que « ligne_affecte » pour savoir quel train, ligne est associé au conducteur et pour quelle" période.

2. Comment savoir quel client a été contrôlé par qui, à quel moment et d'où ?

Nous avons donc ajouté la table Controleurs oublié que nous avons relié à la table Stations par la relation « où » pour savoir dans quelle station son intervention est affectée. Avec la cardinalité 1,1 car un contrôleur ne peut être affecté à plusieurs stations en même temps ou à 0 stations lors d'une seul et même intervention. En revanche, dans l'autre sens, c'est 0,N car une station peut avoir plusieurs contrôleurs affectés comme 0. Enfin, la relation « Contrôles » liant Contrôleurs et Usagers permet de savoir qui a été contrôlé, et grâce à ses champs, quand et combien a couté l'amende s'il y en a eu une, s'il n'y en a pas eu, il suffit de mettre le montant à 0. Un Usager peut ne se fait contrôler qu'une seul fois par un seul contrôleur lors d'une intervention, comme il peut ne pas se faire contrôler d'où la cardinalité 0,1, alors qu'un contrôleur peut contrôler plusieurs usagers dans la même intervention.

3. Comment savoir si deux stations sont liées entre elles ?
Il suffisait de créer une relation, table Liaisons possédant les champs « ID_Action », « ID_Station1 » et « ID_Station2 » qui sont bien évidement les deux stations reliées. Les champs « ID_Station1 et 2 » sont des clés étrangères reprenant le même ID que la clé primaire de la table Station. Liaisons permet donc aisément de savoir si deux stations sont reliés entre elles. Et si une même station est reliée à plusieurs stations, ce qui est possible d'où la cardinalité 1,n (1 car une station est obligatoirement relié à une autre au minimum, même le départ et le terminus, sinon elle ne pourrait pas recevoir de tram et ce serait incohérant), il suffit de rentrer plusieurs lignes.

4. On a donc pu grâce au point numéro 3 vérifier comment les personnels de nettoyage sont affectés aux interventions des rails entre les gares ?
En effet, en créant la table Interventions avec ses champs permettant de connaître l'endroit (station ou rail) ainsi que la date et l'heure, il était aisé de la lié à personnel d'entretien et station à l'aide de deux relations. Mais pour la relier à Liaisons, il fallait d'abord la créer, trouver comment relier deux stations entre elles.

5. Comment savoir si les techniciens ont effectué une vérification ou une réparation, et si les matériels intervenus sont toujours défectueux après l'intervention ?
Il suffisait de créer une relation, table Action_du_Tech possédant les champs « ID_Liaisons », « Type_action », « DateAction », « HeureAction » et « en_panne ». « Type_Action » permet de savoir s'il vérifie ou répare ; « en_panne » permet de savoir, si à la fin de l'action, le matériel est en panne ou non (qu'il l'était déjà avant, mais que le technicien n'est pas réussi à le réparer, ou que l'on ne savait pas et que le technicien l'a découvert en le vérifiant). Les cardinalités sont de 0, n dans les deux sens car un technicien peut vérifier plusieurs matériels lors d'une même intervention, et un matériel peut être réparer par plusieurs techniciens en même temps.

⇨ En résumé, nous avons 18 tables (en incluant les associations traitées comme des tables) et 6 associations.

4. La transition MLD :

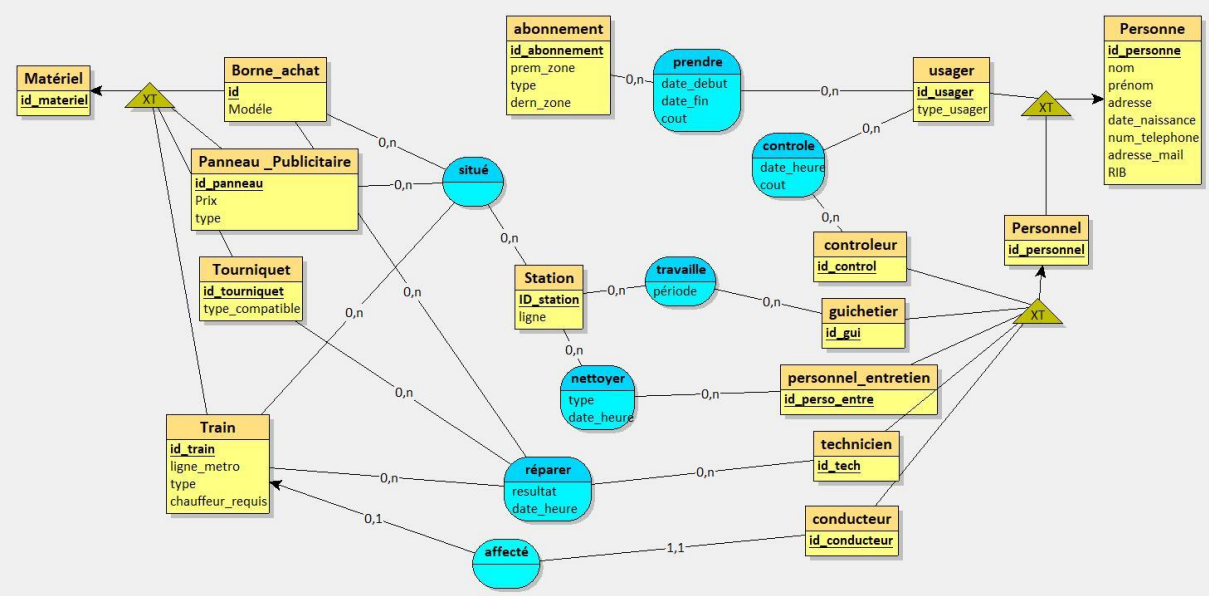

Figure 3: le schéma fourni

Personne = (id_personne, nom, prenom, adresse, date_naissance, num_telephone, adresse_mail, RIB)

usager = (id_usager, type_usager, #id_personne)

Personnel = (id_personnel, #id_personne)

controleur = (id_control, #id_personnel)

guichetier = (id_gui, #id_personnel)

personnel_entretien = (id_perso_entre, #id_personnel)

technicien = (id_tech, #id_personnel)

conducteur = (id_conducteur, #id_personnel, #id_train)

travaille = (#id_gui, #id_station , periode)

nettoyer = (#id_perso-entre, #id_station, type, date_heure)

reparer = (#id_tech, #id, #id_tourniquet ,#id_train resultat, date_heure)

abonnement = (id_abonnement, prem_zone, type, dern_zone)

prendre = (#id_abonnement, #id_usager, date_debut, date_fin,cout)

contrôle=(#id_usager, #id_control, date_heure, coute)

Borne_achat = (id, modele, #id_matériel)

Panneau_Publicitaire = (id_panneau, Prix, type, #id_matériel)

Tourniquet = (id_tourniquet, type_compatible, #id_matériel)

Train =(id_train, ligne_metro, type, chauffeur_requis, #id_matériel)

situé = (#id_train, #id_panneau, #id_station, #id)

Matériel = (id_matériel)

Station = (id_station , ligne) -> on renomme la colonne ligne Station = (id_station , nom)

5.Les contraintes :

- Toutes les clés primaires doivent être unique.
- Les champs « date_heure » sont de types DATETIME
- Le champ « Prix » de la table Panneau_Publicitaire doit être > 0
- Le champ « Cout » de la table Prendre doit être > 0
- « Modele » de la table Panneau_Publicitaire est contrainte à stocker soit 'vieux' soit 'nouveau'
- « chauffeur_requis » dans la table Train doit être entre 0,1
- Le type de données pour la colonne « Type_compatible » dans la table Tourniquet est défini comme booléen, où 1 représente Vrai et 0 représente Faux .
- « type_usager » de la table usager stocke justement « Etudiant , Chômeur , salarié , retraité »
- « resultat » de la table réparer stocke justement « en panne , succès »
- « type » de la table abonnement stocke justement « Carte Orange , Carte Parigo »

6.La création de la BdD :

```
1.  CREATE TABLE Personne(id_personne int(9) PRIMARY KEY, nom varchar(50), preno
    m varchar(50), adresse varchar(255), date_naissance date, num_telephone int(9), adr
    esse_mail varchar(255), RIB varchar(25));
2.  CREATE TABLE usager ( id_usager int(9) PRIMARY key, type_usager varchar(255), id
    _personne int(9), FOREIGN key (id_personne) REFERENCES Personne(id_personne));
    ALTER table usager add CONSTRAINT type_client check(type_usager IN("Étudiant","Ch
    ômeur","retraité","salarié"));
3.  CREATE table Personnel ( id_personnel int(9) PRIMARY KEY, id_personne int(9), FOR
    EIGN key (id_personne) REFERENCES Personne(id_personne));
4.  create table controleur (id_control int(9) PRIMARY KEY, id_personnel int(9), FOREIGN
     key (id_personnel) REFERENCES Personnel(id_personnel));
5.  create table guichetier (id_gui int(9) PRIMARY KEY, id_personnel int(9), FOREIGN key
     (id_personnel) REFERENCES Personnel(id_personnel));
6.  create table personnel_entretien (id_perso_entre int(9) PRIMARY KEY, id_personnel int
    (9), FOREIGN key (id_personnel) REFERENCES Personnel(id_personnel));
7.  create table technicien (id_tech int(9) PRIMARY KEY, id_personnel int(9), FOREIGN k
    ey (id_personnel) REFERENCES Personnel(id_personnel));
8.  create table conducteur (id_conducteur int(9) PRIMARY KEY, id_personnel int(9), FOR
    EIGN key (id_personnel) REFERENCES Personnel(id_personnel));
    ALTER TABLE conducteur ADD COLUMN id_train int(9);
9.  create table abonnement (id_abonnement int(9) PRIMARY KEY, prem_zone varchar(2
    55), type varchar(255),dern_zone varchar(255));
    ALTER TABLE abonnement ADD CONSTRAINT type_abonnement CHECK (type IN ('C
    arte Orange', 'Carte Parigo'));
10.     create table Matériel (id_matériel int(9) PRIMARY key);
11.     create table Station (id_station int(9) PRIMARY key,nom varchar(50));
```

```
12. CREATE TABLE Train ( id_train INT(9) PRIMARY KEY, ligne_metro VARCHAR(50), type VARCHAR(50) NOT NULL, chauffeur_requis VARCHAR(50), id_matériel INT(9), FOREIGN KEY (id_matériel) REFERENCES Matériel(id_matériel), CONSTRAINT requis CHECK (chauffeur_requis IN ('0', '1')) );
13. CREATE table travaille (id_gui int(9),id_station int(9),PRIMARY key(id_gui,id_station) ,periode DATE,FOREIGN key (id_gui) REFERENCES guichetier(id_gui),FOREIGN key (id_station) REFERENCES Station(id_station));
14. CREATE table nettoyer (id_perso_entre int(9),id_station int(9),PRIMARY key(id_perso_entre,id_station) ,date_heure DATETIME,FOREIGN key (id_perso_entre) REFERENCES personnel_entretien(id_perso_entre),FOREIGN key (id_station) REFERENCES Station(id_station));
15. CREATE TABLE contrôle (id_usager int(9),id_control int(9),PRIMARY key(id_usager,id_control) ,date_heure DATETIME,coute decimal(5,2),FOREIGN key (id_usager) REFERENCES usager(id_usager),FOREIGN key (id_control) REFERENCES controleur(id_control));
16. CREATE TABLE prendre (id_abonnement int(9),id_usager int(9),PRIMARY key(id_abonnement,id_usager ) ,date_debut DATE,date_fin date,coute decimal(5,2),FOREIGN key (id_abonnement) REFERENCES abonnement(id_abonnement),FOREIGN key (id_usager ) REFERENCES usager(id_usager ));
    ALTER TABLE prendre ADD CONSTRAINT check_coute_positive CHECK (coute > 0);
17. CREATE table Tourniquet( id_tourniquet int(9) PRIMARY key ,type_compatible tinyint(1),id_matériel int(9),FOREIGN KEY (id_matériel) REFERENCES Matériel(id_matériel));
18. CREATE table Panneau_Publicitaire( id_panneau int(9) PRIMARY key , type varchar(255),Prix decimal(10,2), id_matériel int(9), FOREIGN KEY (id_matériel) REFERENCES Matériel(id_matériel), CONSTRAINT type_panneau CHECK(type IN('petit','moyen','grand')));
    ALTER TABLE Panneau_Publicitaire ADD CONSTRAINT check_prix_positive CHECK (Prix > 0);
19. CREATE table Borne_achat( id int(9) PRIMARY key , modele varchar(50), id_matériel int(9), FOREIGN KEY (id_matériel) REFERENCES Matériel(id_matériel), CONSTRAINT type_born CHECK(modele IN('vieux','nouveau')));
20. CREATE table reparer (id_tech int(9),id int(9),id_tourniquet int(9),id_train int(9),PRIMARY key(id_tech,id,id_tourniquet,id_train),resultat varchar(50) ,date_heure DATEtime,FOREIGN key (id_tech) REFERENCES technicien(id_tech),FOREIGN key (id_train) REFERENCES Train(id_train),FOREIGN key (id) REFERENCES Borne_achat(id),FOREIGN key (id_tourniquet) REFERENCES Tourniquet(id_tourniquet));
    ALTER TABLE reparer ADD CONSTRAINT resultat_check CHECK (resultat IN ('succès', 'en panne'));
21. CREATE table situé (id_panneau int(9),id int(9),id_station int(9),id_train int(9),PRIMARY key(id_panneau,id,id_station,id_train),FOREIGN key (id_panneau) REFERENCES Panneau_Publicitaire(id_panneau),FOREIGN key (id_train) REFERENCES Train(id_train),FOREIGN key (id) REFERENCES Borne_achat(id),FOREIGN key (id_station) REFERENCES Station(id_station));
```

7.Le remplissage de la BdD :

```
1. INSERT INTO Personne VALUES (1, 'Dubois', 'Jean', '10 Rue de la Liberté, Paris', '1980-03-15', 123456789, 'jean.dubois@example.com', '12345678901234567890123'), (2, 'Leroy', '
```

Marie', '22 Avenue des Champs-Élysées, Lyon', '1975-07-20', 987654321, 'marie.leroy@example.com', '9876543210987654321098 7'), (3, 'Moreau', 'Pierre', '6 Rue du Commerce, Marseille', '1990-12-10', 369852147, 'pierre.moreau@example.com', '7890123456789012345678 9'), (4, 'Lefebvre', 'Sophie', '42 Boulevard Haussmann, Paris', '1982-07-12', 654321897, 'sophie.lefebvre@example.com', '4567890123456789012345 6'), (5, 'Martin', 'Julie', '14 Rue Saint-Honoré, Lyon', '1989-03-30', 753186429, 'julie.martin@example.com', '2345678901234567890123 4'), (6, 'Fournier', 'Luc', '8 Avenue des Ternes, Paris', '1970-12-08', 321458796, 'luc.fournier@example.com', '3456789012345678901234 5'), (7, 'Girard', 'Isabelle', '25 Rue de Rivoli, Marseille', '1985-08-15', 876543210, 'isabelle.girard@example.com', '[...]

2. INSERT INTO Matériel VALUES (1), (2), (3), (4), (5), (6), (7), (8), (9), (10), (11),(12),(13),(14),(15),(16),(17),(18),(19),(20);
3. INSERT INTO Personnel VALUES (1, 2), (2, 10), (3, 3), (4, 4), (5, 5), (6,9), (7, 6), (8, 7), (9, 8), (10, 1), (11, 12), (12, 11) ;
4. INSERT INTO guichetier VALUES (1, 1), (2, 5);
5. INSERT INTO personnel_entretien VALUES (1, 2);
6. INSERT INTO technicien VALUES (1, 3), (2, 4);
7. INSERT INTO Tourniquet VALUES (1, 1, 1), (2, 0, 2), (3, 1, 3);
8. INSERT INTO Panneau_Publicitaire VALUES (1, 'petit', 100.00, 4), (2, 'moyen', 150.00, 5), (3, 'grand', 200.00, 6),(4,'petit',95.00,7),(5,'moyen',120.00,8),(6,'grand',180,9);
9. INSERT INTO Borne_achat VALUES (1, 'vieux', 10), (2, 'nouveau', 11), (3, 'vieux', 12),(4, 'vieux', 13),(5, 'vieux', 14),(6, 'nouveau', 15);
10. INSERT INTO Train VALUES (1, 'Ligne 1', 'Métro', '1', 16), (2, 'Ligne 2', 'RER', '0', 17), (3, 'Ligne 2', 'Métro', '1', 18), (4, 'Ligne 4', 'Tramway', '0', 19), (5, 'Ligne 5', 'Métro', '1', 20);
11. INSERT INTO controleur VALUES (1, 6), (2, 7), (3, 8);
12. INSERT INTO usager VALUES (1, 'Étudiant', 13), (2, 'Chômeur', 14), (3, 'salarié', 15), (4, 'salarié', 16), (5, 'Étudiant', 17), (6, 'retraité', 18), (7, 'Étudiant', 19), (8, 'Étudiant', 20);
13. INSERT INTO abonnement VALUES (1, 'Zone A', 'Carte Orange' , 'Zone B'), (2, 'Carte Parigo', 'Annuel', 'Zone D'), (3, 'Zone E', 'Carte Parigo', 'Zone F'), (4, 'Zone G', 'Carte Parigo', 'Zone H'), (5, 'Zone I', 'Carte Orange', 'Zone J');
14. INSERT INTO prendre VALUES (1, 8, '2023-01-01', '2023-12-31', 25.00), (2, 2, '2023-02-01', '2023-12-31', 100.00), (3, 3, '2023-03-01', '2023-12-31', 150.00), (4, 4, '2023-04-01', '2023-12-31', 45.00), (5, 5, '2023-05-01', '2023-12-31', 360.00), (4, 6, '2023-06-01', '2023-12-31', 40.00), (3, 7, '2023-06-01', '2023-12-31', 90.00), (2, 1, '2023-06-01', '2023-12-31', 220.00);
15. INSERT INTO contrôle VALUES (3, 1, '2024-03-14 09:00:00', 50.00), (2, 2, '2024-03-14 09:30:00', 25.00), (3, 3, '2024-03-14 10:00:00', 50.00), (7, 3, '2024-03-14 10:30:00', 20.00), (2, 3, '2024-03-14 11:00:00', 25.00), (6, 1, '2024-03-14 11:30:00',25.00);
16. INSERT INTO conducteur VALUES (1, 9, 1), (2, 10, 5), (3, 11, 3), (4, 12, 3);
17. INSERT INTO reparer VALUES (1, 4, 1, 5, 'succès', '2024-03-15 08:00:00'), (2, 2, 2, 2, 'en panne', '2024-03-16 10:30:00'), (2, 3, 3, 3, 'en panne', '2024-03-17 13:45:00'), (1, 3, 3, 3, 'succès', '2024-03-18 15:20:00'), (1, 5, 1, 5, 'succès', '2024-03-19 09:10:00');

```
18. INSERT INTO Station VALUES (1, 'Châtelet - Les Halles'), (2, 'Gare de Lyon'), (3, 'Nation'), (4, 'Charles de Gaulle - Étoile'), (5, 'Gare du Nord'), (6, 'Gare Saint-Lazare');
19. INSERT INTO travaille (id_gui, id_station, periode) VALUES (1, 1, '2022-03-14'), (2, 3, '2022-03-14'), (2, 6, '2023-03-10'), (1, 2, '2023-03-10'), (2, 5, '2024-03-14');
20. INSERT INTO nettoyer (id_perso_entre, id_station, date_heure) VALUES (1, 1, '2024-03-13 08:00:00'), (1, 3, '2024-03-14 09:30:00'), (1, 5, '2024-03-15 09:30:00');
21. INSERT INTO situé (id_panneau, id, id_station, id_train) VALUES (1, 1, 3, 2), (2, 2, 5, 1), (3, 3, 2, 4), (4, 4, 6, 3), (5, 5, 1, 5);
```

8.Simulation et Test d'une requête.

1.Probleme : Le directeur souhaite gérer les incidents et les réparations de manière efficace afin d'éviter les interruptions de service prolongées.

Solution :

Mesurer l'efficacité du travail des ingénieurs en comptabilisant le nombre de fois où une réparation a été effectuée mais que le problème persiste. J'ai utilise INNER JOIN , GROUP BY,COUNT ainsi que CASE pour compter le nombre de succès ou le nombre d'échecs après la réparation dans 2 colonnes nommées « nbEnPanne » et « nbSucces ».

```
SELECT id_tech,Personne.nom,Personne.prenom,
COUNT(CASE WHEN resultat = 'en panne' THEN 1 END) AS nbEnPanne,
COUNT(CASE WHEN resultat = 'succès' THEN 1 END) AS nbSucces
FROM reparer
INNER JOIN technicien using(id_tech) INNER JOIN Personnel using (id_personnel) INNER Join Personne using(id_personne) GROUP BY id_tech;
```

id_tech	nom	prenom	nbEnPanne	nbSucces
1	Moreau	Pierre	0	3
2	Lefebvre	Sophie	2	0

Figure 4 : le résultat de la solution du problème

Résultat :

Cela indique précisément l'efficacité de chaque ingénieur dans son travail, ce qui permet de décider s'il faut recruter plus de techniciens, augmenter ou réduire les salaires, ou prendre d'autres mesures appropriées.