

Dossier

Pour répondre aux besoins de la société Travel Train, nous avons besoin d'un Schéma de base de données.

1er étape :

On vérifie que toutes les classes ont un identifiant. Ici la classe **Trajet** n'a pas d'identifiant. On va donc utiliser la règle [R-2.1](#). Cette règle permet de faire une association binaire avec 1 à plusieurs. Il faut l'utiliser entre la classe **Trajet** et **Train** ainsi que **Trajet** et **Voyage** (pour avoir l'identifiant de **Trajet** qui sera IdTrain, IdVoyage). Ceci nous a permis d'avoir un identifiant relatif.

Identifiant de **Trajet** (IdTrain, IdVoyage)

La classe associative **Réservation** n'a pas d'identifiant. Pour mettre un identifiant à une classe associative, il faut utiliser la règle [R-4.1](#). Cette règle permet de prendre les identifiants de l'association entre **Client** et **Voyage** pour les mettre en clé. Chaque attribut de la clé devient clé étrangère.

Identifiant de **Réservation** (IdPersonne, IdVoyage).

2ème étape :

Une fois qu'on a vérifié que toutes les classes ont un identifiant, on va maintenant faire la suppression de l'héritage.

-héritage "Personne" [R-5.3](#)

On garde les 3 étant donné que nous avons une totalité et qu'une personne peut être soit un client, soit un employé, soit les deux.

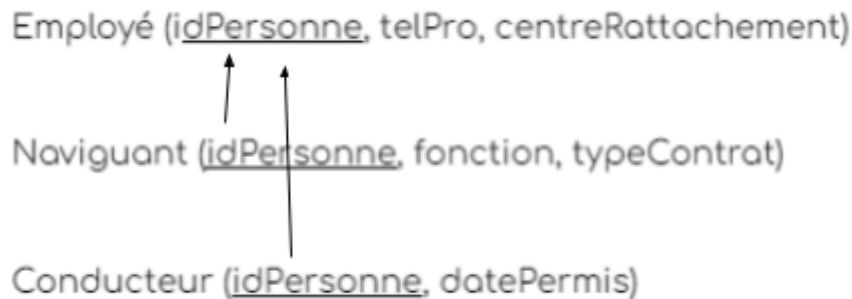
Personne (IdPersonne, nomPers, prenomPres, adressePers, telPers)

Client (IdPersonne, age, codeReduc)

Employé (IdPersonne, telPro, centreRattachement)

-héritage "Employé" R-5.3

On garde les 3 étant donné qu'il y a une partition et que les 3 classes subissent une association.



3ème étape :

On réécrit toutes les classes et attribut en relation grâce à la règle R-1.

Train (idTrain, typetrain, duplex)

Voiture (numVoiture, position, dateMiseCirculation)

Gare (idgare, ville, nomgare)

Trajet (idTrain, idVoyage, dateDepart, heureDepart, heureArrivee, tarifBase)

Voyage (idVoyage, dateCreation, codeTarif)

On supprime maintenant les associations.

Il faut maintenant ajouter toutes les clés étrangères. Pour ce faire, on utilise la règle R-2.1. On ne peut l'utiliser que quand on a une association binaire 1 à plusieurs. Ici, nous l'utilisons pour Voyage, Réservation, Voiture et Train.

Si on a une association plusieurs à plusieurs, il faut utiliser la règle R-2.2. Cette règle va créer une classe en plus contenant que les clés primaires des deux classes. Ici, les classes Train et Navigant utilisent cette règle, qui donne donc la classe Train-Navigant.

Les classes Client et Voyage utilisent aussi cette règle, ce qui donne la classe Client-Voyage.

Schéma base de donnée

Train (idTrain, typetrain, duplex, IdPersone, idGare)

Voiture (numVoiture, position, dateMiseCirculation, idTrain)

Gare (idgare, ville, nomgare)

Personne (idPersonne, nomPers, prenomPres, adressePers, telPers)

Client (IdPersonne, age, codeReduc)

Employé (idPersonne, telPro, centreRattachement)

Navigant (idPersonne, fonction, typeContrat)

Conducteur (idPersonne, datePermis,)

Trajet (idTrain, idVoyage, dateDepart, heureDepart, heureArrivee, tarifBase)

Voyage (idVoyage, dateCreation, codeTarif, idTrain)

Reservation (idPersonne, idVoyage, numPlace, dateReserv, NumVoiture)

Train_Navigant (idTrain, idPersonne)

Client-Voyage (idPersonne, idVoyage)

Création des tables en SQL

```
CREATE TABLE Train (  
    idTrain INT UNIQUE NOT NULL,  
    typeTrain VARCHAR(20),  
    duplex VARCHAR(7),  
    idPersonne INT UNIQUE NOT NULL,  
    idGare INT UNIQUE NOT NULL,  
    PRIMARY KEY (idTrain),  
    FOREIGN KEY (idPersonne) REFERENCES Conducteur(idPersonne),  
    FOREIGN KEY (idGare) REFERENCES Gare(idGare)  
);
```

```
Create Table Voiture (  
    numVoiture INT UNIQUE NOT NULL,  
    position INT NOT NULL,  
    dateMiseCirculation DATE,  
    idTrain INT UNIQUE NOT NULL,  
    PRIMARY KEY (numVoiture),  
    FOREIGN KEY (idTrain) REFERENCES Train (idTrain)  
);
```

```
Create Table Gare (  
    idGare INT UNIQUE NOT NULL,  
    ville VARCHAR(30),  
    nomgare VARCHAR(30),  
    PRIMARY KEY (idGare)  
);
```

```
Create Table Personne (  
    idPersonne INT UNIQUE NOT NULL,  
    nomPers VARCHAR(30),  
    prenomPers VARCHAR(30),  
    adressePers VARCHAR(30),  
    telPers VARCHAR(12),  
    PRIMARY KEY (idPersonne)  
);
```

```
Create Table Client (  
    idPersonne INT UNIQUE NOT NULL,  
    age INT NOT NULL,  
    codeReduc VARCHAR(20),  
    PRIMARY KEY (idPersonne),  
    FOREIGN KEY (idPersonne) REFERENCES Personne(idPersonne)  
);
```

```
Create Table Employe (  
    idPersonne INT UNIQUE NOT NULL,  
    telPro VARCHAR(12),  
    centreRattachement VARCHAR(30),  
    PRIMARY KEY (idPersonne),  
    FOREIGN KEY (idPersonne) REFERENCES Personne(idPersonne)  
);
```

```
Create Table Naviguant (  
    idPersonne INT UNIQUE NOT NULL,  
    fonction VARCHAR(30),  
    typeContrat VARCHAR(3),  
    PRIMARY KEY (idPersonne),  
    FOREIGN KEY (idPersonne) REFERENCES Employe (idPersonne)  
);
```

```
Create Table Conducteur (  
    idPersonne INT UNIQUE NOT NULL,  
    datePermis DATE,  
    PRIMARY KEY (idPersonne),  
    FOREIGN KEY (idPersonne) REFERENCES Employe (idPersonne)  
);
```

```
Create Table Trajet (  
    idTrain INT UNIQUE NOT NULL,  
    idVoyage INT UNIQUE NOT NULL,  
    dateDepart DATE,  
    heureDepart TIME,  
    heureArrivee TIME,  
    tarifBase INT,  
    PRIMARY KEY (idTrain, idVoyage),  
    FOREIGN KEY (idTrain) REFERENCES Train (idTrain),  
    FOREIGN KEY (idVoyage) REFERENCES Voyage (idVoyage)  
);
```

```
Create Table Voyage (  
    idVoyage INT UNIQUE NOT NULL,  
    dateCreation DATE,  
    codeTarif VARCHAR(30),  
    idTrain INT UNIQUE NOT NULL,  
    PRIMARY KEY (idVoyage),  
    FOREIGN KEY (idTrain) REFERENCES Trajet (idTrain)  
);
```

```
Create Table Reservation (  
    idPersonne INT UNIQUE NOT NULL,
```

```

    idVoyage INT UNIQUE NOT NULL,
    numplace INT,
    dateReserv DATE,
    numVoiture INT UNIQUE NOT NULL,
    PRIMARY KEY ( idPersonne, idVoyage),
    FOREIGN KEY ( NumVoiture) REFERENCES Voiture (numVoiture)
);

Create Table Train_Navigant (
    idTrain INT UNIQUE NOT NULL,
    idPersonne INT UNIQUE NOT NULL,
    PRIMARY KEY (idTrain, idPersonne),
    FOREIGN KEY (idTrain) REFERENCES Train(idTrain),
    FOREIGN KEY (idPersonne) REFERENCES Naviguant (idPersonne)
);

Create Table Client_Voyage (
    idPersonne INT UNIQUE NOT NULL,
    idVoyage INT UNIQUE NOT NULL,
    PRIMARY KEY (idPersonne, idVoyage),
    FOREIGN KEY (idPersonne) REFERENCES Client (idPersonne),
    FOREIGN KEY (idVoyage) REFERENCES Voyage (idVoyage)
);

```