

# R403 - Qualité et au-delà du rationnel

## TD 4

---

Floran MARTEL  
Clovis LEPORT

1.

Pour savoir si les tables qui nous sont fournies sont bien en 3FNBC, il faut qu'on applique l'algorithme de décomposition. Mais pour appliquer l'algorithme de décomposition, il faut connaître les relations, or nous n'avons que les données des tables.

Pour trouver les relations entre les colonnes, il faut supposer toutes les relations possibles et vérifier si elles sont vraies. Alors comme cette tâche est très fastidieuse et nécessite peu d'imagination nous avons décidé d'investir notre temps en développant un petit algorithme capable de trouver toutes les dépendances qui ne dépendent que d'un élément. Cet algorithme nous a donc permis de dégrossir une partie du travail.

Après avoir exécuté notre algorithme et chercher les autres dépendances parmi toutes les tables, nous avons trouvé des dépendances qui semblent ne pas avoir de sens, par exemple dans la table personne, on trouve que le nom de la ville dépend du nom de la personne, ce qui évidemment est faux. Ces dépendances illogiques sont présentes, car la quantité des données est trop faible pour explorer toutes les combinaisons possibles. Nous avons donc dû analyser les relations trouvées pour supprimer celle qui n'avait pas de sens.

Après tout ce processus, nous obtenons ces tables.

Véhicule		Clé : plaque_immatriculation	
plaque_immatriculation	-->	marque,	
plaque_immatriculation	-->	energie,	2NF
plaque_immatriculation	-->	nombre_places,	
plaque_immatriculation	-->	annee_mise_en_service	
marque,	-->	type,	

Personne		Clé : id_personne	
id_personne	-->	nom,	
id_personne	-->	prenom,	2NF
id_personne	-->	nom_ville	
id_personne	-->	adresse_rue,	
nom_ville, adresse_rue	-->	code_postal	

Conducteur		clé : numero_agrement	
(numero_agrement,	-->	date_agrement,	
(numero_agrement,	-->	certifie,	3NF
(numero_agrement,	-->	#id_personne	

Trajet		clé : id_trajet	
id_trajet	-->	heure_dep,	
id_trajet	-->	heure_des,	2NF
#code_postal_des,#code_postal_dep	-->	longueur,	
id_trajet	-->	tarif,	
id_trajet	-->	#numero_agrement,	
id_trajet	-->	#plaque_immatriculation,	
id_trajet	-->	#code_postal_dep)	
id_trajet	-->	#code_postal_des	

Voyage		Clé : #id_trajet, #id_personne	
			3NF

Avis		Clé : id_avis	
id_avis	-->	message,	
id_avis	-->	etoiles,	
id_avis	-->	#id_personne_envoie	3NF
id_avis	-->	#id_personne_recu	

Une fois, toutes les relations trouvées, nous avons pu exécuter l'algorithme de décomposition

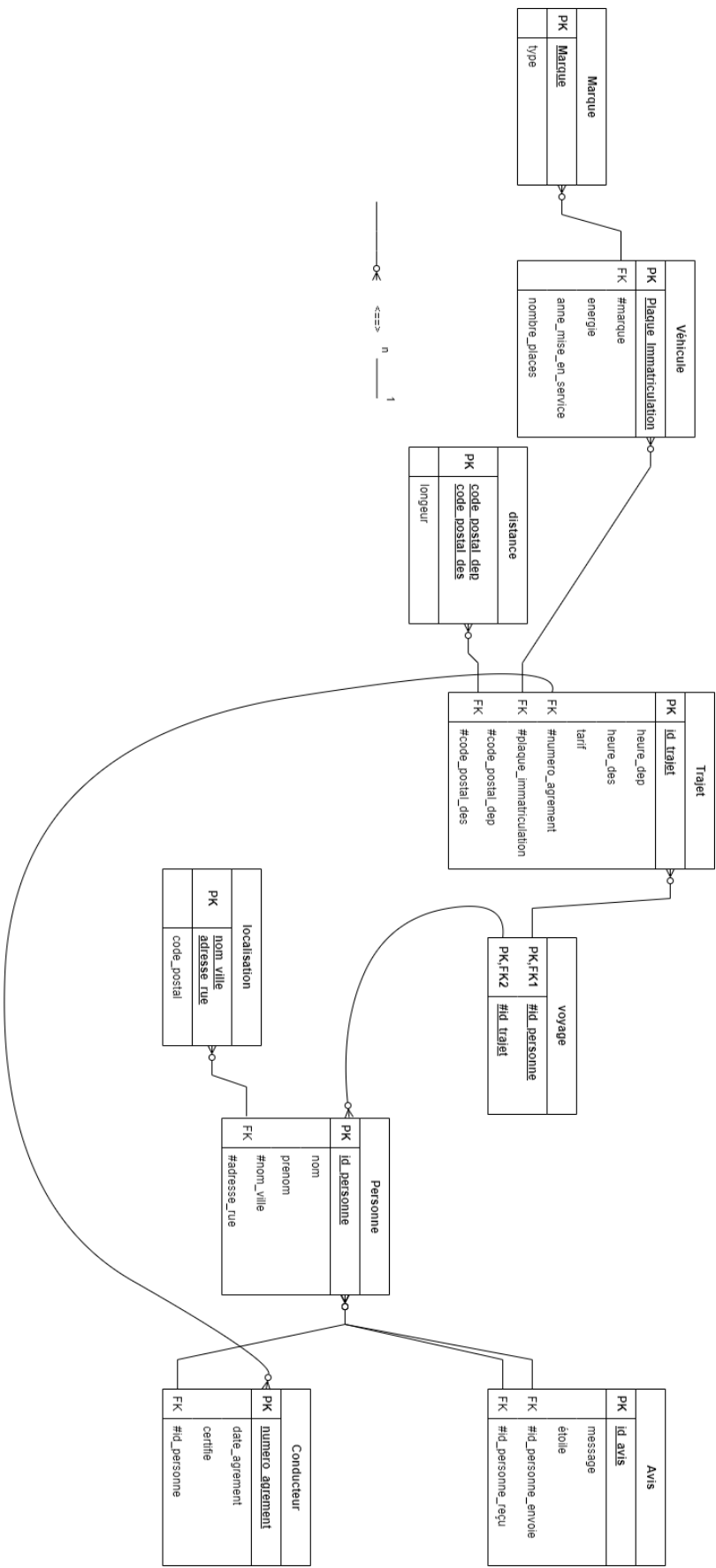
Et nous obtenons donc ces nouvelles tables en 3FNBC.

Marque (V1)	Clé : marque		Véhicule (V2)	plaque_immatriculation	
marque	--> type	3FNBC	plaque_immatriculation	--> marque,	
			plaque_immatriculation	--> energie,	3FNBC
			plaque_immatriculation	--> nombre_places,	
			plaque_immatriculation	--> annee_mise_en_service	

adresse (P1)	Clé : nom_ville, adresse_rue		Personne (P2)	id_personne	
nom_ville, adresse_rue	--> code_postal	3FNBC	id_personne	--> nom,	
			id_personne	--> prenom,	
			id_personne	--> nom_ville	3FNBC
			id_personne	--> adresse_rue,	

distance (T1)	Clé : #code_postal_des,#code_postal_dep		Trajet (T2)	id_trajet	
#code_postal_des,#code_postal_dep	--> longueur	3FNBC	id_trajet	--> heure_dep,	
			id_trajet	--> heure_des,	
			id_trajet	--> tarif,	
			id_trajet	--> #numéro_agrement,	
			id_trajet	--> #plaque_immatriculation,	
			id_trajet	--> #code_postal_dep)	3FNBC
			id_trajet	--> #code_postal_des	

2.  
Nous obtenons donc ce schéma



3.

Une fois, toute la base de données définies, nous souhaitons l'importer dans MongoDB. Pour l'importer, il faut d'abord créer un ou plusieurs fichiers .JSON qui contiendra nos données. Nous avons donc transformé notre base de données en fichier .JSON qui sera l'imbrication de certaines tables pour qu'au final nous n'ayons que 2 fichiers JSON, nommer Trajet et Personne.

fichier json Trajet

```
[
{
  "id_trajet" : 1,
  "heure_dep" : "2022-02-20 20-00",
  "heure_des" : "2022-02-20 20-52",
  "tarif" : 5,
  "conducteur" : {
    "numero_agrement" : 13978,
    "date_agrement" : "2017-01-28",
    "certife" : 1,
    "id_personne" : 5
  },

  "vehicule" : {
    "plaque_immatriculation" : "ab123xz",
    "marque" : {
      "marque" : "Skoda",
      "type" : "voiture"
    },
    "energie" : "essence",
    "anne_mise_en_service" : 1999,
    "nombre_places" : 4
  },

  "distance" : {
    "code_postal_dep" : 44000,
    "code_postal_des" : 49300,
    "longueur" : 59
  },

  "id_voyageurs" : [
    2,12,15,3
  ]
},
]
```

```
{
  "id_trajet" : 2,
  .....
}
```

fichier json Personne

```
[
  {
    "id_personne" : 0,
    "nom": "Hoareau",
    "prenom": "Jacqueline",

    "localisation": {
      "nom_ville": "Nantes",
      "adresse_rue": "9 Chem. de la Pelletterie",
      "code_postale": 44000
    },

    "voyages": {
      "id_trajet": []
    },

    "avis_reçu": []
  },
  {
    "id_personne" : 1,
    "nom": "Camus",
    "prenom": "Suzanne",

    "localisation": {
      "nom_ville": "Nantes",
      "adresse_rue": "53 Rue du Douet Garnier",
      "code_postale": 44000
    },

    "voyages": {
      "id_trajet": []
    },

    "avis_reçu": [{
      "id_avis": 6,
      "message": "A",
      "etoiles": 1,
      "id_personne_source": 13
    }],
  },
]
```

```

{
  "id_avis":7,
  "message": "B",
  "etoiles":5,
  "id_personne_source": 15
},
{
  "id_avis":8,
  "message": "A",
  "etoiles":5,
  "id_personne_source": 18
},
{
  "id_avis":9,
  "message": "C",
  "etoiles":5,
  "id_personne_source": 7
}
]
},
{
  "id_personne" : 2,
  .....
}
]

```

Les fichiers complets sont dans les fichiers Trajet.json et Personne.json

4.

Une fois nos fichiers créés, nous pouvons les utiliser dans mongodb et exécuter les requêtes demandées pour obtenir certaines informations.

```

//1
db.Trajet.find({ $and: [ { "distance.code_postal_dep": 44000}, {"distance.code_postal_des":
49300 } ] },{_id:0})

```

//2

```
db.Trajet.aggregate([

  {
    $match : {
      "vehiclue.anne_mise_en_service" : {$lt:2000}
    }
  },

  { $group:
    { _id: {
      numero_agrement : "$conducteur.numero_agrement",
      date_agrement : "$conducteur.date_agrement",
      certife : "$conducteur.certife",
      id_personne : "$conducteur.id_personne",
      vehiclue_anne_mise_en_service : "$vehiclue.anne_mise_en_service",
      vehiclue_energie : "$vehiclue.energie"
    },

    }
  },

  {
    $lookup : {
      from : "Personne",
      localField : "_id.id_personne",
      foreignField : "id_personne",
      as : "personne"
    },
  },

  {
    $unwind : '$personne'
  },

  {
    $match : {
      "personne.localisation.nom_ville" : "Nantes"
    }
  },

  {
    $addFields: { numero_agrement: "$_id.numero_agrement",
      vehiclue_anne_mise_en_service : "$_id.vehiclue_anne_mise_en_service",
      id_personne : "$_id.id_personne",
      nom_ville : "$personne.localisation.nom_ville",
      vehiclue_energie : "$_id.vehiclue_energie"}
  },

  {$project : {
    _id : 0, personne:0
  }}
])
```



```
//3
db.Trajet.aggregate([
  { $group:
    { _id: {
      numero_agrement : "$conducteur.numero_agrement",
      date_agrement : "$conducteur.date_agrement",
      certife : "$conducteur.certife",
      id_personne : "$conducteur.id_personne",
    },

    }
  },
  {
    $lookup : {
      from : "Personne",
      localField : "_id.id_personne",
      foreignField : "id_personne",
      as : "personne"
    },
  },
  {
    $unwind : '$personne'
  },
  {
    $match : {
      "personne.localisation.nom_ville" : "Nantes",
      "personne.avis_reçu.etoiles":{$not: {
        "$lt":2.5
      }
    }
  },
  {
    $addFields: { id_personne: "$personne.id_personne",
      nom : "$personne.nom",
      prenom : "$personne.prenom",
      localisation : "$personne.localisation",
      avis_reçu : "$personne.avis_reçu"
    },
    {$project : {
      _id : 0, personne:0
    }}
  }
])
```

```
//4
db.Trajet.aggregate([

  { $group:
    { _id : "$conducteur.numero_agrement",
      nbtrajet : {$sum : 1}
    }
  }
])
```

```
    }
  },
  {
    $sort : {nbtrajet : -1}
  }
])
```

```
//5

db.Trajet.aggregate([
  {
    $unwind : '$id_voyageurs'
  },
  {
    $lookup: {
      from: "Personne",
      localField: "id_voyageurs",
      foreignField: "id_personne",
      as: "passager"
    }
  },
  {
    $unwind : '$passager'
  },
  {
    $match : {
      "passager.localisation.code_postale" : "$distance.code_postal_des"
    }
  }
])
```

```
//6
db.Trajet.aggregate([

  { $addFields: { date: { $toDate: "$conducteur.date_agrement" } } },

  {
    $match : {
      "date" : {$gt: new Date("2014-01-01")}
    }
  },

  { $group:
    { _id: {
      numero_agrement : "$conducteur.numero_agrement",
      date_agrement : "$conducteur.date_agrement"
    },
    }
  },

  {
    $addFields: { numero_agrement: "$_id.numero_agrement",
      date_agrement : "$_id.date_agrement"
    },
    {$project : {
      _id : 0
    }}
  }
])
```

```
//7
db.Trajet.aggregate([

  { $group:
    { _id: {
      numero_agrement : "$conducteur.numero_agrement",
      id_personne : "$conducteur.id_personne",
    },
      nbvoyageur : {$sum : {$size : "$id_voyageurs"}}
    }
  },

  {
    $match : {
      nbvoyageur : 0
    }
  },

  {
    $addFields: { numero_agrement: "$_id.numero_agrement",
      id_personne : "$_id.id_personne",
    }
  },

  {
    $project : {
      _id : 0
    }
  }
])
```

```

    {$project : {
      _id : 0
    }}
  })

```

```

//8
db.Trajet.aggregate([

  { $addField: { nbpassagerT: { $size: "$id_voyageurs" } } },

  { $group:
    { _id: "$conducteur.numero_agrement" ,
      nbVoyageur : { $sum : "$nbpassagerT" }
    }
  },
  {
    $sort : {nbVoyageur : -1}
  },
  { $limit : 1 }
])

```

```

//9
db.Trajet.aggregate([

  {
    $match : {"conducteur.numero_agrement" : 13978}
  },

  { $group:
    { _id: {
      numero_agrement : "$conducteur.numero_agrement",
      id_personne : "$conducteur.id_personne",
    },

    }
  },
  {
    $lookup : {
      from : "Personne",
      localField : "_id.id_personne",
      foreignField : "id_personne",
      as : "personne"
    },
  },
  {
    $unwind : '$personne'
  },
  {

```

```

    $unwind : '$personne.avis_reçu'
  },
  {
    $match : {
      "personne.avis_reçu.etoiles":{
        "$lt":2.5
      }
    }
  },
  {
    $group : {
      _id : "$personne.id_personne",
      nbAvisNégatif : {$sum : 1}
    }
  }
]
])

```

```

//10
db.Trajet.aggregate([
  {
    $match : {
      "conducteur.numero_agrement" : 13978
    }
  },
  {
    $lookup : {
      from : "Personne",
      localField : "conducteur.id_personne",
      foreignField : "id_personne",
      as : "personne"
    },
  },
  {
    $unwind : '$personne'
  },
  {
    $unwind : '$personne.avis_reçu'
  },
  {
    $match : {
      "personne.avis_reçu.etoiles" : {$lte:3}
    }
  },
  { $addFields: {
    id_trajet: "$id_trajet",

```

```

        heure_dep: '$heure_dep',
        heure_des: '$heure_des',
        code_postal_dep: '$distance.code_postal_dep',
        code_postal_des: '$distance.code_postal_des'}
    },

    {$project: {
        _id:0,
        id_trajet: 1,
        heure_dep : 1,
        heure_des : 1,
        code_postal_dep : 1,
        code_postal_des : 1,
    }}
]
])

```

```

//11

db.Trajet.aggregate([

    {
        $lookup : {
            from : "Personne",
            localField : "conducteur.id_personne",
            foreignField : "id_personne",
            as : "personne"
        },
    },
    {
        $unwind : '$personne'
    },
    {
        $match : {
            "personne.prenom" : "Suzanne"
        }
    },
    {
        $project: { _id: 0}
    }
]
])

```