

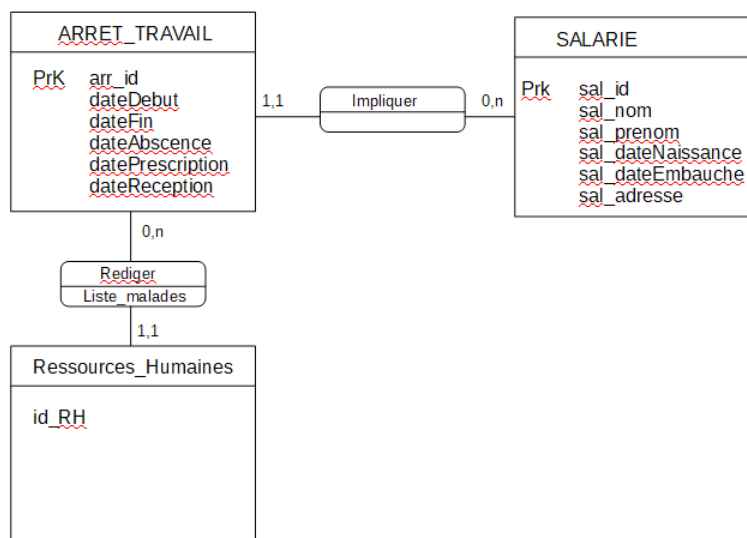
## Cas Déménagement

### Mission 1 : Améliorer la planification des déménagements

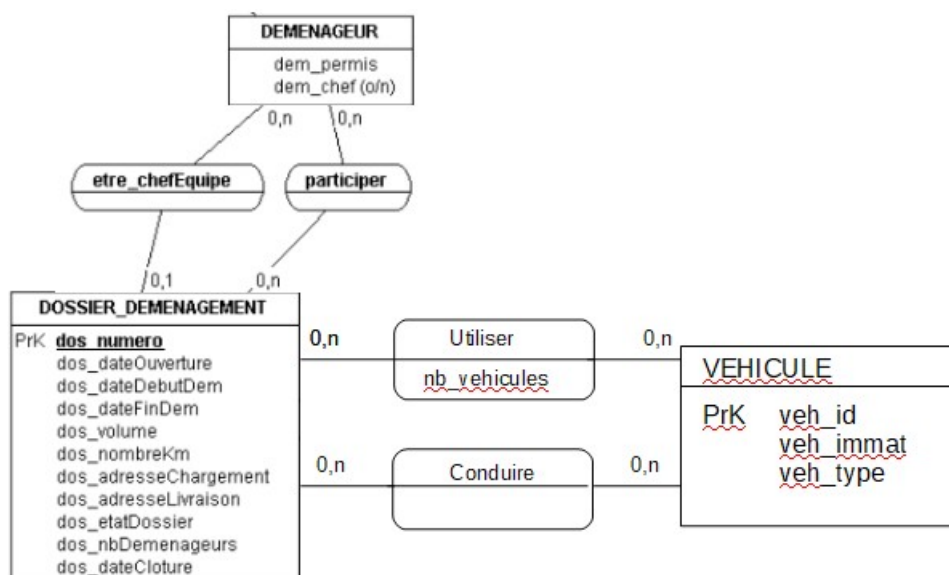
1. 

```
SELECT * FROM salarie
WHERE sal_agence = @idAgence and sal_type = 'D'
and sal_id in ( select par_dossier from participer
               where dossier_demenagement = dos_chefEquipe)
```

3.



4.



5.

The screenshot shows a software interface with three main sections: 'AFFECTATION DES PERSONNELS', 'AFFECTATION DES VEHICULES', and 'CONDUCTEURS'.  
1. 'AFFECTATION DES PERSONNELS': At the top right, 'Dossier déménagement' is set to '1254'. Below, 'Déménageurs disponibles' lists Durand Hervé, Franek Alain, Lorient Pascale, Rampon Fabrice, and Hugues Martial. 'Déménageurs participants' lists Dubois Aline and Jaraf Mohammed. Between these lists are buttons 'Participe >>' and '<< Ne participe pas'. Below this is a dropdown for 'Chef d'équipe responsable' with 'Jaraf Mohammed' selected.  
2. 'AFFECTATION DES VEHICULES': 'Véhicules disponibles' lists DT-011-TK, EZ-947-WZ, FC-132-YD, FC-095-LA, and FC-524-FJ. 'Véhicules utilisés' lists CD-158-BS. Between these lists are buttons 'Utilise >>' and '<< N'utilise pas'.  
3. 'CONDUCTEURS': A box containing the names 'Dubois Aline' and 'Jaraf Mohammed'.

6. REQ\_001    Select dos\_chefEquipe, dos\_numero  
              from dossier\_demenagement  
              where dos\_numero = 2814 ;

REQ\_002    Select count(veh\_id), count(par\_demenageur), dos\_numero  
              From vehicule, par\_demenageur, dossier\_demenageur  
              where  
              Group by dossier\_demenagement ;

REQ\_003    Select dos\_numero, dos\_DateDebutDem  
              From dossier\_demenagement  
              where par\_demenageur from participer is null ;

### Mission 2 : Gérer les absences des salariés

7. // constructeur de la classe Salarie

```
public Salarie (String matricule, string nom, string prenom)
{
    matricule = unMatricule
    nom= unNom
    prenom = unPrenom
    lesAbsences = new List <Absence> ;
}
```

8. // méthode getLesAbsences() de la classe Salarie

```
public List <Absence> LesAbsences() { return lesAbsences ; }
```

9. // méthode ajouterAbsence() de la classe Salarie

```
public void ajouterAbsence(Absence uneAbsence)
{
    LesAbsences.Add ( new Absence(uneAbsence)) ;
}
```

10. //méthode getLesAbsencesEnAttente() de la classe Salarie

```
public List<Absence> getLesAbsencesEnAttente() { return AbsencesEnAttente}
// Accepte l'absence qui débute à une DateDebut
public void accepterAbsence(DateFr uneDateDebut) {
    Absence uneAbsence = lesAbsences.Find(element => element.GetDateDebut()==
    uneDateDebut) ;
    if (uneAbsence!= null) {
        uneAbsence .accepter() ;
    }
}
```

11. // méthode getLesAbsencesAccepteesDuMois() de la classe Salarie

```
public List <Absence> getLesAbsencesAccepteesDuMois() { return AbsencesDuMois}
public void accepterAbsenceDuMois(DateFr unMois) {
    Absence uneAbsence = lesAbsences.Find(element => element.GetDate(Month)==
    unMois) ;
    if (etat = « ACC ») {
        uneAbsence.accepterAbsenceDuMois() ;
    }
}
```

12. // méthode refuserAbsence() de la classe salarié

```
// refuse l'absence qui débute à une DateDebut
public void refuserAbsence(DateFr uneDateDebut, string unMotifRefus) { return refuserAbsence}
    Absence uneAbsence = lesAbsences.Find(element =>
    element.GetDateDebut()==uneDateDebut) ;
    if (uneAbsence!= null){
        uneAbsence.refuser() ;
    }
}
```

13. // ajoute l'absence dans l'objet le salaire

```
absence.Add(leSalaire) ;
```

14. //refuser l'absence

```
public void refuser(string unMotifRefus){  
    etat = "REF" ;  
    dateReponse = new DateFr() ;  
}
```

15. // méthode getLesJoursOuvrables() dans la classe Absence

```
public int getMois(unMois=12){return mois}  
public int getAnne(uneAnnee = 2018) {return annee} ;
```

### Mission 3 : Suivre la réalisation des déménagements

20.

```
[  
  {  
    "Dossier recherché" : ""  
  },  
  {  
    "Phase" : "chargement",  
    "Temps assemblage" : "04:25 "  
  },  
  "véhicules utilisés" : [  
    {  
      "Immatriculation" : "EZ-786-WL"  
      "Heure arrivée" : "2016-11-05 08:45:00"  
      "Heure départ" : "2016-11-05 12:30:00"  
      "cubage" : "27"  
    },  
    {  
      "Immatriculation" : "ET-565-XQ"  
      "Heure arrivée" : "2016-11-05 09:00:00"  
      "Heure départ" : "2016-11-05 12:30:00"  
      "cubage" : "33"  
    },  
  ]  
]
```

21. La méthode getJSON() est utilisée pour obtenir les données JSON à l'aide d'une requête AJAX HTTP GET.

22.