

**E5SD : PRODUCTION ET  
FOURNITURE DE SERVICES**

Durée : 4 heures

Coefficient : 5

**CAS DÉMGUIV**

ÉLÉMENTS DE CORRECTION

**Compétences évaluées :**

**D2.3 Gestion des problèmes et des changements**

**D4.1 Conception et réalisation d'une solution applicative**

- Développement, utilisation ou adaptation de composants logiciels
- Conception ou adaptation d'une base de données

**1. Réaliser les modifications nécessaires à la correction de l'anomalie *Ano\_chef1*.**

Faire une nouvelle fonction pour alimenter la liste du chef d'équipe :

```
CREATE FUNCTION [dbo].[chefsPresents](@idAgence int, @debutDossier date, @finDossier date) RETURNS TABLE
AS
RETURN (SELECT * FROM salarie
        where sal_agence = @idAgence and sal_type='D'
        and sal_chef = 1
        and sal_id not in (select con_salarie from conge
                          where con_dateFin >= @debutDossier and con_dateDebut <= @finDossier )
        and sal_id not in (select par_demenageur from participer, DOSSIER_DEMENAGEMENT
                          where par_dossier = dos_numero and dos_dateFinDem >= @debutDossier
                          and dos_dateDebutDem <= @finDossier )
        )
```

OU

```
CREATE function [dbo].[chefsPresents](@idAgence int,@debutDossier date,@finDossier date) returns table
as
return (select * from demenageursPresents(@idAgence,@debutDossier,@finDossier) where sal_chef=1)
```

OU

Dans l'application cliente, indiquer comme source de la liste :  
select \* from demenageursPresents(2, '25/05/2016', '30/05/2016') where sal\_chef=1

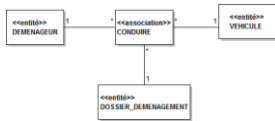
**2. Rédiger une courte note indiquant le principe d'une solution permettant de corriger l'anomalie *Ano\_chef2*.**

Ecrire un trigger pour update sur la table dossier :  
Si dos\_chefEquipe n'est pas null alors  
Vérifier qu'il est bien dans la table participer pour ce dossier

OU

Gérer la contrainte dans l'applicatif.





CONDUIRE(cnd\_demenageur, cnd\_vehicule, cnd\_dossier\_demenagement)  
 Clé primaire : cnd\_demenageur, cnd\_vehicule, cnd\_dossier\_demenagement  
 Clé étrangère : cnd\_demenageur en référence à sal\_id de SALARIE  
 Clé étrangère : cnd\_vehicule en référence à veh\_id de VEHICULE  
 Clé étrangère : cnd\_dossier\_demenagement en référence  
 à dos\_numero de DOSSIER\_DEMENAGEMENT

Mission 2 : Gérer les véhicules de déménagement

---

Compétences évaluées :

**D1.1 Analyse de la demande**  
**D2.3 Gestion des problèmes et des changements**  
**D4.1 Conception et réalisation d'une solution applicative**  
 • Conception ou adaptation d'une base de données

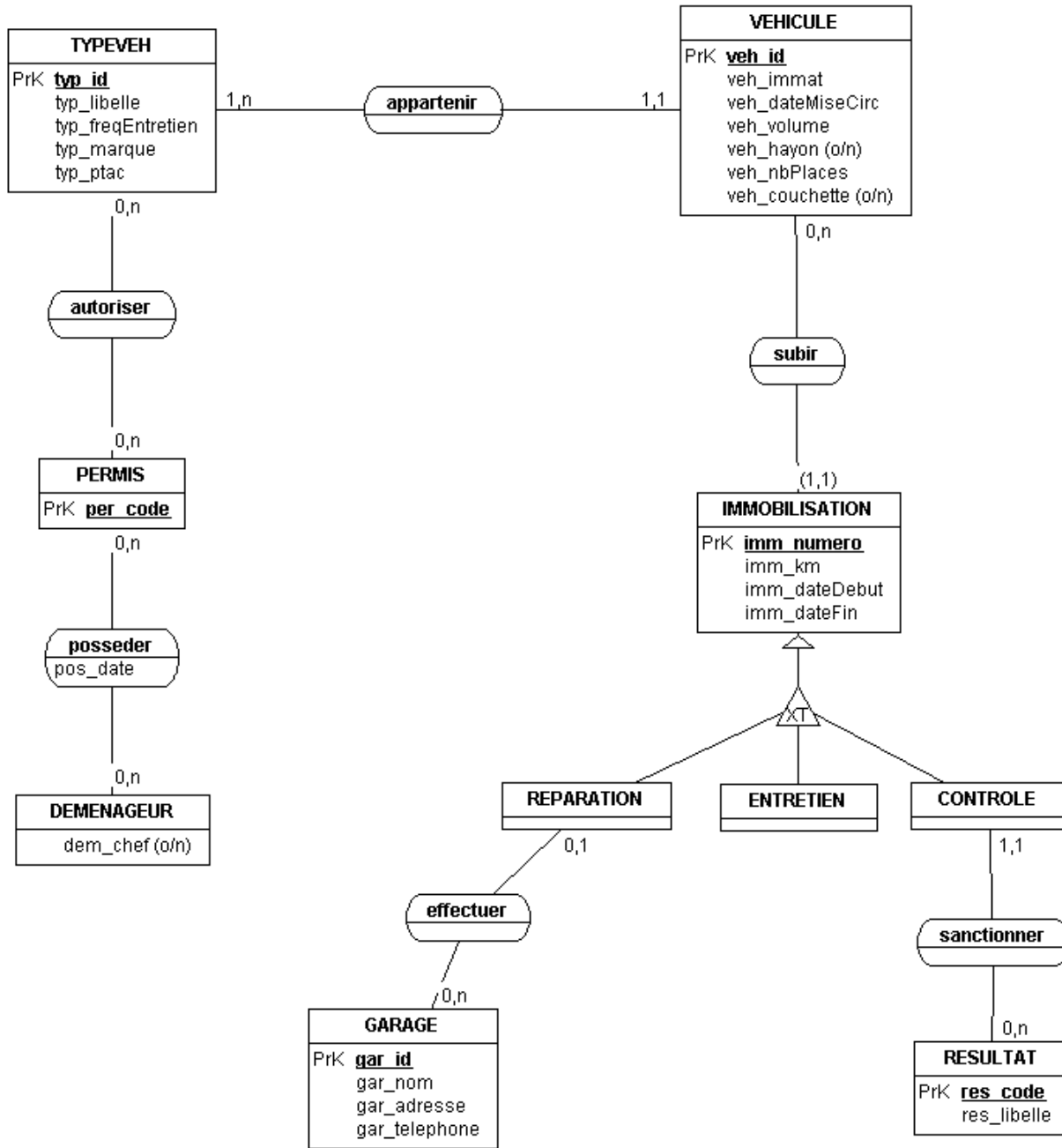
**5. En se référant à la note concernant les besoins en information et pour chacun des besoins exprimés restant à traiter (de REQ\_003 à REQ\_005), indiquer si la base de données actuelle permet de le satisfaire. Si oui, présenter la requête SQL nécessaire. Si non, indiquer brièvement la raison de cette impossibilité.**

--REQ\_003  
 select veh\_immat from occupationVehicules('12/05/2016')  
 where veh\_agence = 1

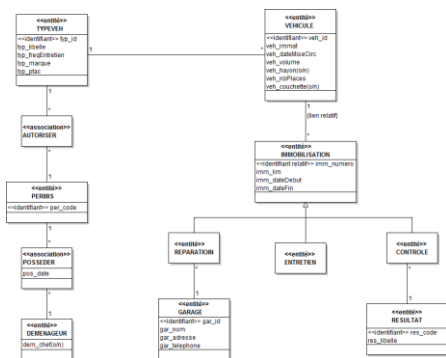
--REQ\_004 : impossible, on ne gère pas les entretiens. Il faudrait a minima garder trace dans véhicule du kilométrage au moment du dernier entretien.

--REQ\_005  
 select SUM(dos\_nombreKm)  
 from DOSSIER\_DEMENAGEMENT  
 where dos\_dateDebutDem > '12/02/2016'  
 and dos\_numero in  
 (select uti\_dossier  
 from UTILISER, vehicule  
 where uti\_vehicule=veh\_id  
 and veh\_immat=' BC-095-LA')

Proposer une modélisation de la nouvelle base de données à mettre en place en intégrant la base existante, la gestion des permis, des véhicules et la fiche de suivi. Seuls les éléments du schéma existant qui sont concernés par l'évolution seront repris dans le schéma proposé.



Remarque : il y a plusieurs solutions pour les permis. On peut citer une réflexive pour la hiérarchisation et une cardinalité 0,1 vers le permis de plus haut niveau des déménageurs.



TYPEVEH(typ\_id, typ\_libelle, typ\_freqEntretien, typ\_marque, typ\_ptac)  
Clé primaire : typ\_id

PERMIS (per\_code)  
Clé primaire : per\_code

AUTORISER(aut\_typeVeh, aut\_permis)  
Clé primaire : aut\_typeVeh, aut\_permis  
Clé étrangère : aut\_typeVeh en référence à typ\_id de TYPEVEH  
Clé étrangère : aut\_permis en référence à per\_code de PERMIS

SALARIE(sal\_id, sal\_nom, sal\_prenom, sal\_dateNaissance, sal\_dateEmbauche, sal\_adresse, sal\_agence, sal\_type, sal\_chef, sal\_experience)  
Clé primaire : sal\_id  
Clé étrangère : sal\_agence en référence à age\_id de Agence

POSSEDER(pos\_permis, pos\_salarie, pos\_date)  
Clé primaire : pos\_permis, pos\_salarie  
Clé étrangère : pos\_permis en référence à per\_code de PERMIS  
Clé étrangère : pos\_salarie en référence à sal\_id de SALARIE

VEHICULE(veh\_id, veh\_immat, veh\_dateMiseCirc, veh\_volume, veh\_hayon(o/n), veh\_nbPlaces, veh\_couchette(o/n), veh\_typeVeh)  
Clé primaire : veh\_id  
Clé étrangère : veh\_typeVeh en référence à typ\_id de TYPEVEH

GARAGE(gar\_id, gar\_nom, gar\_adresse, gar\_telephone)  
Clé primaire : gar\_id

RESULTAT(res\_code, res\_libelle)  
Clé primaire : res\_code

IMMOBILISATION(imm\_vehicule, imm\_numero, imm\_km, imm\_dateDebut, imm\_dateFin,  
imm\_type, imm\_garage, imm\_resultat)

Clé primaire : imm\_vehicule, imm\_numero

Clé étrangère : imm\_vehicule en référence à veh\_id de VEHICULE

Clé étrangère : imm\_garage en référence à gar\_id de GARAGE

Clé étrangère : imm\_resultat en référence à res\_code de RESULTAT

Remarque : imm\_type permet de distinguer les réparations, les entretiens et les contrôles

Mission 3 : Gérer les absences des salariés

---

**Compétences évaluées :**

**D2.3 Gestion des problèmes et des changements**

**D4.1 Conception et réalisation d'une solution applicative**

• Développement, utilisation ou adaptation de composants logiciels

**6. Fournir le code de la méthode *ajouterAbsence* de la classe *Salarie*.**

```
public void ajouterAbsence(Absence uneAbsence) {  
    lesAbsences.add(uneAbsence);  
}
```

**7. Fournir le code de la méthode *refuser* de la classe *DemandeAbsence*.**

```
public void refuser(String unMotifRefus) {  
    laAbsence.setLeEtat("REF");  
    leMotifRefus = unMotifRefus;  
}
```

**8. Fournir le code de la méthode *getAbsencesEnAttente* de la classe *Salarie*.**

```
public List<Absence> getAbsencesEnAttente() {  
    List<Absence> enAttente = new ArrayList<Absence>();  
    for(Absence uneAbsence : lesAbsences) {  
        if (uneAbsence.getLeEtat().equals("ATT")){  
            enAttente.add(uneAbsence);  
        }  
    }  
    return enAttente;  
}
```

Remarque : l'utilisation de « equals » n'est pas exigée.

**9. Fournir le code de la méthode *getJoursAbsence* de la classe *Absence*.**

```
public List<DateFr> getJoursAbsence(int unMois, int uneAnnee) {  
    List<DateFr> joursAbsence = new ArrayList<DateFr>();  
    DateFr dateCourante = laDateDebut;  
    while (! dateCourante.estApres(laDateFin)) {  
        if (dateCourante.estJourOuvrable())  
            && dateCourante.getMois() == unMois  
            && dateCourante.getAnnee() == uneAnnee) {  
                joursAbsence.add(dateCourante);  
            }  
        dateCourante = dateCourante.getJourSuivant();  
    }  
    return joursAbsence;  
}
```

**Compétences évaluées :**

**D1.2 Choix d'une solution**

- Élaboration et présentation d'un dossier de choix de solution technique
- Évaluation des risques liés à l'utilisation d'un service

**D4.1 Conception et réalisation d'une solution applicative**

- Développement, utilisation ou adaptation de composants logiciels

**10. Rédiger une courte note expliquant à M. Guivarch la phrase « Protéger l'information consiste à garantir sa disponibilité, son intégrité, sa confidentialité et sa traçabilité ».**

Disponibilité : données accessibles.

Intégrité : données contrôlées et valides.

Confidentialité : données protégées contre les accès non autorisés.

Traçabilité : connaissance de l'origine de l'acquisition et de la mise à jour des données.

**11. Proposer un tableau destiné à alimenter la réflexion de M. Guivarch présentant les avantages et les inconvénients de la dématérialisation des documents de déménagement.**

Avantages	Inconvénients
<ul style="list-style-type: none"><li>• Dématérialisation de la lettre de voiture légale</li><li>• Informations disponibles rapidement (presque en temps réel)</li><li>• Enregistrement instantané des informations légales possible à chaque étape pour l'acteur concerné</li><li>• Disparition des documents manuscrits peu lisibles</li><li>• Possibilité d'intégrer des informations au PGI de l'entreprise</li></ul>	<ul style="list-style-type: none"><li>• Sécurité des données : nécessite la mise en place d'une stratégie de sécurité souvent complexe</li><li>• Risque de perte des données (piratage, mauvaises manipulations, applications défectueuses ...)</li><li>• Nécessite du matériel complémentaire pour la saisie sur site (PDA, téléphone mobile GPRS, PC, etc.)</li><li>• Nécessité d'accompagner le changement</li></ul>

**12. Proposer une représentation des données échangées après le chargement (au format *Json*) entre l'application cliente et l'application serveur prenant en compte les nouvelles informations demandées par M. Guivarch.**

```
{
  "dossier" : "1917",
  "phase" : "chargement",
  "tempsEmballage" : "xx:xx"
  "vehicule" : [
    {
      "immatriculation" : "EZ-786-WL",
      "arrivee" : "2016-04-05 08:45:00",
      "depart" : "2016-04-05 12:30:00",
      "cubage" : "xx"
    },
    {
      "immatriculation" : "ET-565-XQ",
      "arrivee" : "2016-04-05 09:00:00",
      "depart" : "2016-04-05 12:30:00",
      "cubage" : "xx"
    }
  ]
}
```



**13. Coder la partie de la méthode getFicheInstructionAction() du contrôleur RealisationController (fichier RealisationController.php) correspondant à l'envoi de la réponse.**

```
if( $ficheInstruction == null ){
    $reponse->setStatusCode(404) ;
}
else {
    $reponse->headers->set('ContentType','application/json') ;
    $reponse->setContent( $ficheInstruction->toJson() ) ;
    $reponse->setStatusCode(200) ;
}
```