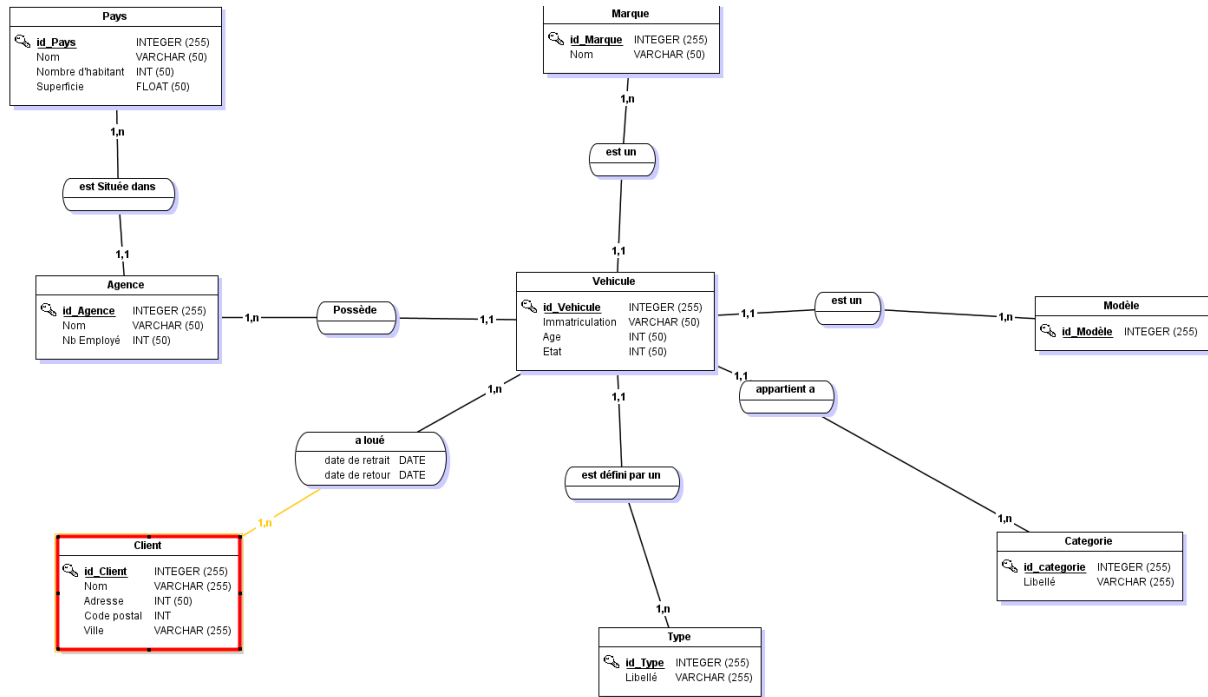
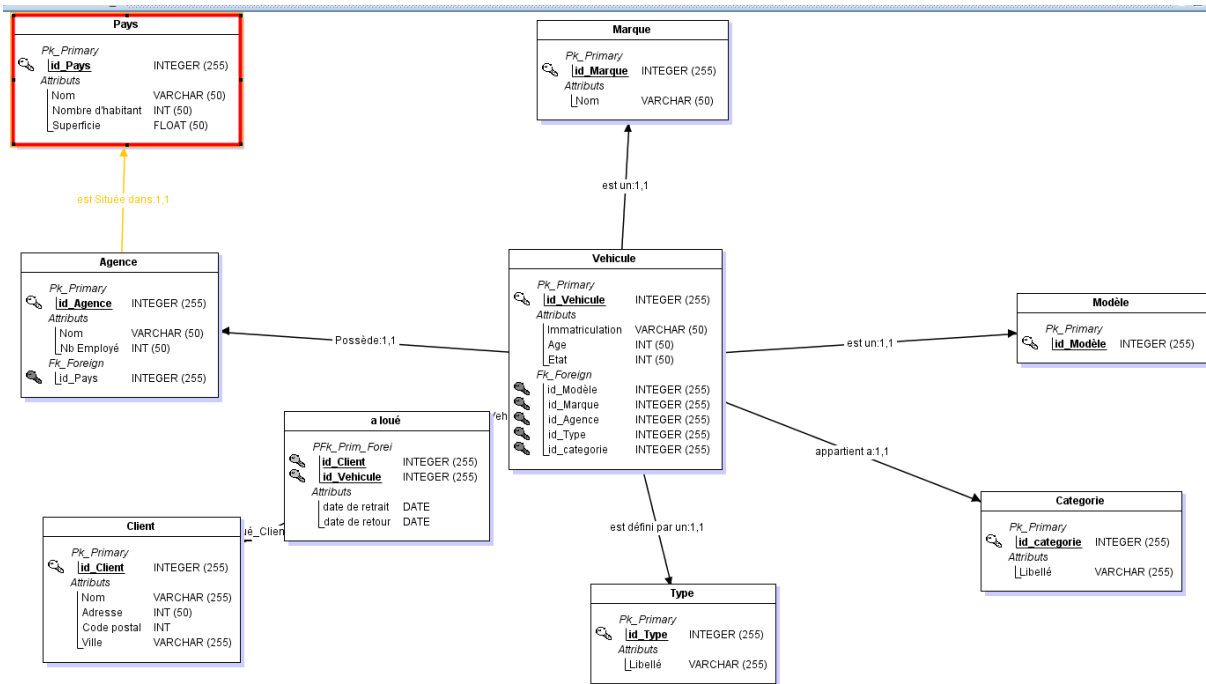


## MISSION 3

MCD :



MLD :



SQL :

```

#-----
# Table: Pays
#-----

CREATE TABLE Pays(
    id_Pays Integer NOT NULL ,
    Nom Varchar (50) NOT NULL ,
    Nombre_d_habitant Int NOT NULL ,
    Superficie Float NOT NULL
    ,CONSTRAINT Pays_PK PRIMARY KEY (id_Pays)
)ENGINE=InnoDB;

#-----
# Table: Marque
#-----

CREATE TABLE Marque(
    id_Marque Integer NOT NULL ,
    Nom Varchar (50) NOT NULL
    ,CONSTRAINT Marque_PK PRIMARY KEY (id_Marque)
)ENGINE=InnoDB;

#-----
# Table: Modèle
#-----

CREATE TABLE Modele(
    id_Modele Integer NOT NULL
    ,CONSTRAINT Modele_PK PRIMARY KEY (id_Modele)
)ENGINE=InnoDB;

#-----
# Table: Agence
#-----

CREATE TABLE Agence(
    id_Agence Integer NOT NULL ,
    Nom Varchar (50) NOT NULL ,
    Nb_Employe Int NOT NULL ,
    id_Pays Integer NOT NULL
    ,CONSTRAINT Agence_PK PRIMARY KEY (id_Agence)

    ,CONSTRAINT Agence_Pays_FK FOREIGN KEY (id_Pays) REFERENCES Pays(id_Pays)
)ENGINE=InnoDB;

#-----
# Table: Categorie
#-----

CREATE TABLE Categorie(
    id_categorie Integer NOT NULL ,
    Libelle Varchar (255) NOT NULL
    ,CONSTRAINT Categorie_PK PRIMARY KEY (id_categorie)
)ENGINE=InnoDB;

#-----
# Table: Type
#-----

CREATE TABLE Type(
    id_Type Integer NOT NULL ,
    Libelle Varchar (255) NOT NULL
    ,CONSTRAINT Type_PK PRIMARY KEY (id_Type)
)ENGINE=InnoDB;

#-----
# Table: Vehicule
#-----

CREATE TABLE Vehicule(
    id_Vehicule Integer NOT NULL ,
    Immatriculation Varchar (50) NOT NULL ,
    Age Int NOT NULL ,
    Etat Int NOT NULL ,
    id_Modele Integer NOT NULL ,
    id_Marque Integer NOT NULL ,
    id_Agence Integer NOT NULL ,
    id_Type Integer NOT NULL ,
    id_categorie Integer NOT NULL
    ,CONSTRAINT Vehicule_PK PRIMARY KEY (id_Vehicule)

    ,CONSTRAINT Vehicule_Modele_FK FOREIGN KEY (id_Modele) REFERENCES Modele(id_Modele)
    ,CONSTRAINT Vehicule_Marque0_FK FOREIGN KEY (id_Marque) REFERENCES Marque(id_Marque)
    ,CONSTRAINT Vehicule_Agence1_FK FOREIGN KEY (id_Agence) REFERENCES Agence(id_Agence)
    ,CONSTRAINT Vehicule_Type2_FK FOREIGN KEY (id_Type) REFERENCES Type(id_Type)
    ,CONSTRAINT Vehicule_Categorie3_FK FOREIGN KEY (id_categorie) REFERENCES Categorie(id_categorie)
)ENGINE=InnoDB;

```

```

#-----
# Table: Client
#-----

CREATE TABLE Client(
  id_Client      Integer NOT NULL ,
  Nom            Varchar (255) NOT NULL ,
  Adresse        Int NOT NULL ,
  Code_postal    Int NOT NULL ,
  Ville          Varchar (255) NOT NULL
  ,CONSTRAINT Client_PK PRIMARY KEY (id_Client)
)ENGINE=InnoDB;

#-----
# Table: a loué
#-----

CREATE TABLE a_loue(
  id_Client      Integer NOT NULL ,
  id_Vehicule     Integer NOT NULL ,
  date_de_retrait Date NOT NULL ,
  date_de_retour  Date NOT NULL
  ,CONSTRAINT a_loue_PK PRIMARY KEY (id_Client,id_Vehicule)
  ,CONSTRAINT a_loue_Client_FK FOREIGN KEY (id_Client) REFERENCES Client(id_Client)
  ,CONSTRAINT a_loue_Vehicule0_FK FOREIGN KEY (id_Vehicule) REFERENCES Vehicule(id_Vehicule)
)ENGINE=InnoDB;

```

#### Exercice 1 :

- a. SELECT \* FROM véhicule WHERE Client Nom=T122 ;
- b. SELECT \* FROM locations INNER JOIN Client ON Client id\_Client=location id\_client WHERE Client Nom=T122 ;
- c. SELECT Immatriculation, Age, État FROM véhicule ;
- d. SELECT nom, Adresse FROM client WHERE ville='Nice' ;
- e. SELECT nom FROM Client ORDER BY Nom;
- f. ALTER TABLE vehicule ADD kilométrage INT NOT NULL ;  
SELECT \*  
FROM véhicule  
ORDER BY Kilometrage DESC ;
- g. SELECT id\_Client, Nom, Adresse, Code\_postal, Ville FROM Client INNER JOIN Location ON WERE véhicule = 'EW 25 EW' ;
- h. SELECT \* FROM véhicule WHERE couleur 'noir' ;
- i. SELECT \* FROM véhicule WHERE kilométrage < 10 000 ;
- j. SELECT \* FROM locations WHERE YEAR (< 2018) ;
- k. SELECT AVG(kilometrage) as moyenne\_kilometrage FROM voiture ;
- l. SELECT \* FROM locations WHERE YEAR (2018) ;
- m. SELECT COUNT(\*) FROM véhicule WHERE kilometrage < 10 000 ;

#### Partie 2

- 1) SELECT id\_vehicule, agence FROM Vehicule INNER JOIN Location ON Vehicule.id\_Vehicule =Location.id\_Vehicule INNER JOIN Agence ON Vehicule.id\_Agence =Agence.id\_Agence WHERE Location.Date\_retrai=location Date\_retour;
- 2) SELECT marque count(\*) FROM véhicule GROUP BY marque ;
- 3) SELECT nom FROM Client HAVING COUNT(véhicule) > 10 ;
- 4) SELECT agent, employé FROM pays ;

#### Exercice 2 :

- 1) SELECT \* FROM ETUDIANT ORDER BY NomEt ASC ;
- 2) SELECT NomEns, GradeEns FROM ENSEIGNANT INNER JOIN MATIERE ON ENSEIGNANT.CodeMat=MATIERE.CodeMat WHERE Nom NomMat='BD' ;
- 3) SELECT DISTINCT NomMat, CoefMat FROM MATIERE INNER JOIN ENSEIGNANT ON ENSEIGNANT.CodeMat=MATIERE.CodeMat WHERE Nom NomMat='BD' ;
- 4) SELECT NomMat, CoefMat FROM MATIERE INNER JOIN MATIERE ON NOTE.CodeMat=MATIERE.CodeMat WHERE CodeEt='ET321' ;
- 5) SELECT COUNT(Enseignant CodeEns) FROM ENSEIGNANT INNER JOIN MATIERE ON ENSEIGNANT.CodeMat=MATIERE.CodeMat WHERE NomMat='Informatique' ;

Exercice 3 :

- 1) SELECT NumCoureur, NomCoureur, CodePays FROM Coureur INNER JOIN equipe on NumCoureur = CodeEquipe WHERE NomEquipe = 'Festina' ;

- 2) SELECT COUNT(km) FROM Attribuer\_Bonification ;

3)

```
SELECT SUM(NbKm) AS TotalKmHauteMontagne FROM ETAPE WHERE CodeType = ( SELECT CodeType FROM TYPE_ETAPE WHERE LibelleType = 'Haute Montagne' );
```

4)

```
SELECT DISTINCT NomCoureur FROM COUREUR WHERE NuméroCoureur NOT IN ( SELECT DISTINCT NuméroCoureur FROM ATTRIBUER_BONIFICATION );
```

5)

```
SELECT NomCoureur FROM COUREUR WHERE NOT EXISTS ( SELECT etape.NuméroEtap FROM ETAPE WHERE NOT EXISTS ( SELECT * FROM PARTICIPER WHERE PARTICIPER.NuméroEtap = ETAPE.NuméroEtap AND PARTICIPER.NuméroCoureur = COUREUR.NuméroCoureur ) );
```

6)

```
SELECT COUREUR.NomCoureur, EQUIPE.CodeEquipe, COUREUR.CodePays, SUM(TIME_TO_SEC(PARTICIPER.TempsRealisé) + ATTRIBUER_BONIFICATION.NbSecondes) AS TempsTotal FROM COUREUR JOIN PARTICIPER ON COUREUR.NuméroCoureur = PARTICIPER.NuméroCoureur JOIN ETAPE ON PARTICIPER.NuméroEtap = ETAPE.NuméroEtap JOIN EQUIPE ON COUREUR.CodeEquipe = EQUIPE.CodeEquipe LEFT JOIN ATTRIBUER_BONIFICATION ON PARTICIPER.NuméroEtap = ATTRIBUER_BONIFICATION.NuméroEtap AND PARTICIPER.NuméroCoureur = ATTRIBUER_BONIFICATION.NuméroCoureur WHERE ETAPE.NuméroEtap <= 13 GROUP BY COUREUR.NuméroCoureur ORDER BY TempsTotal;
```

7)

```
SELECT EQUIPE.NomEquipe, SUM(TIME_TO_SEC(PARTICIPER.TempsRealisé) + ATTRIBUER_BONIFICATION.NbSecondes) AS TempsTotal FROM EQUIPE JOIN COUREUR ON EQUIPE.CodeEquipe = COUREUR.CodeEquipe JOIN PARTICIPER ON COUREUR.NuméroCoureur = PARTICIPER.NuméroCoureur JOIN ETAPE ON PARTICIPER.NuméroEtap = ETAPE.NuméroEtap LEFT JOIN ATTRIBUER_BONIFICATION ON PARTICIPER.NuméroEtap = ATTRIBUER_BONIFICATION.NuméroEtap AND PARTICIPER.NuméroCoureur = ATTRIBUER_BONIFICATION.NuméroCoureur WHERE ETAPE.NuméroEtap <= 13 GROUP BY EQUIPE.CodeEquipe ORDER BY TempsTotal;
```

Exercice 4 :

- 1)SELECT \* FROM Clients WHERE 'Marrakech';
- 2) SELECT Numprod, désignation, prix FROM Produit ORDER BY prix DESC;

- 3) SELECT NomVendeur, AdresseVendeur FROM vendeur WHERE letter 'M';
- 4) SELECT \* FROM Commande INNER JOIN Vendeur ON Commande.IdVendeur = Vendeur.IdVendeur WHERE NomVendeur = 'Mohammed' AND date\_com BETWEEN '2020-01-01' AND '2020-01-30' ;
- 5) SELECT COUNT(\*) FROM CommandeWhere NumProd = 365;

Exercice 5:

- 1) SELECT \* FROM Etudiant;
- 2) SELECT nom\_matiere, coefficient FROM matiere;
- 3) SELECT Numéro\_carte\_étudiant FROM Note GROUP BY Numéro\_carte\_étudiant HAVING AVG(note\_exam) >= 7 AND AVG(note-examen) <= 12;
- 4) SELECT \* FROM Etudiant WHERE Nom LIKE 'ben';
- 5) SELECT COUNT(DISTINCT numéro\_carte\_étudiant) FROM Note WHERE code\_matière = 12518 ;
- 6) SELECT SUM(Coefficient) FROM matière ;
- 7) SELECT Nom, Prénom FROM Etudiant Where numéro\_carte\_étudiant IN ( SELECT numéro\_carte\_étudiant FROM Note WHERE note\_examen > 10 ;
- 8) SELECT Nommatiere, coefficient FROM matière INNER JOIN Note ON Matiere.code\_matière = Note.code\_matière WHERE Note.numéro\_carte\_étudiant = 01234568 ;