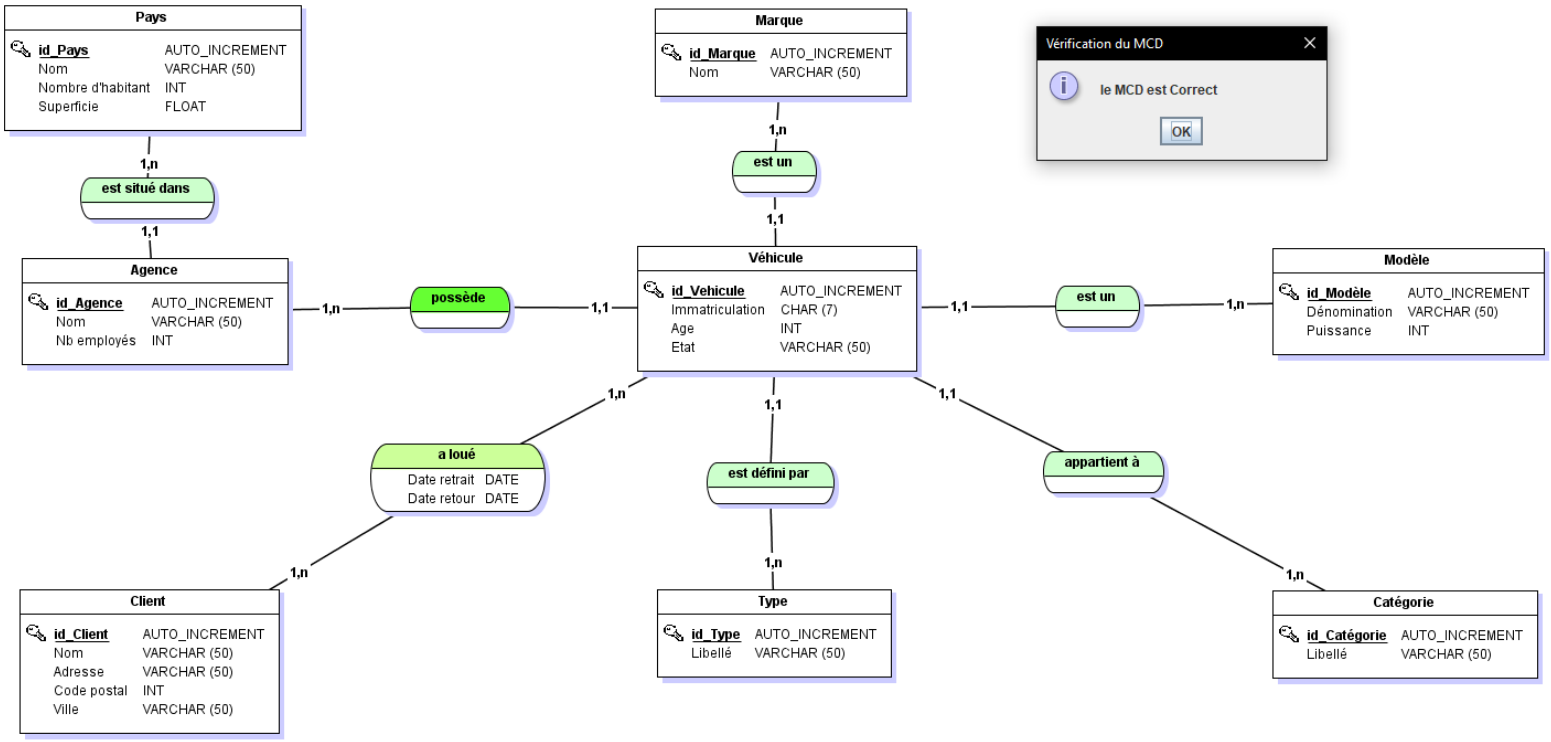
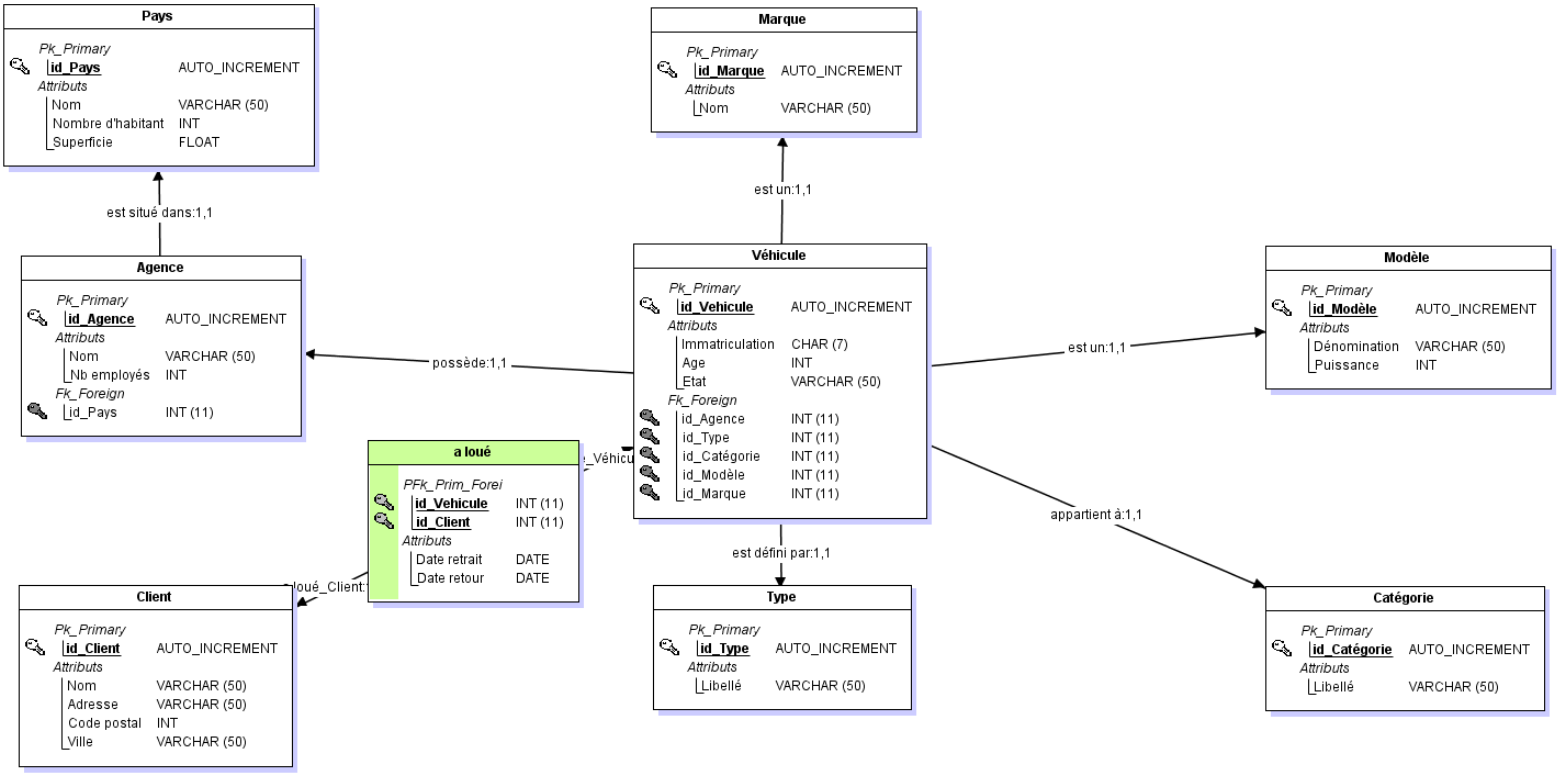
JMerise :

MCD :



MLD :



SQL :

```
#------------------------------------------------------------
#        Script MySQL.
#------------------------------------------------------------


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

CREATE TABLE Pays(
        id_Pays          Int  Auto_increment  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: Agence
#------------------------------------------------------------

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

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


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

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


#------------------------------------------------------------
# Table: Modèle
#------------------------------------------------------------

CREATE TABLE Modele(
```

```
    id_Modele    Int  Auto_increment  NOT NULL ,
    Denomination Varchar (50) NOT NULL ,
    Puissance    Int NOT NULL
      ,CONSTRAINT Modele_PK PRIMARY KEY (id_Modele)
)ENGINE=InnoDB;


#------------------------------------------------------------
# Table: Catégorie
#------------------------------------------------------------

CREATE TABLE Categorie(
    id_Categorie Int  Auto_increment  NOT NULL ,
    Libelle      Varchar (50) NOT NULL
      ,CONSTRAINT Categorie_PK PRIMARY KEY (id_Categorie)
)ENGINE=InnoDB;


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

CREATE TABLE Type(
    id_Type Int  Auto_increment  NOT NULL ,
    Libelle Varchar (50) NOT NULL
      ,CONSTRAINT Type_PK PRIMARY KEY (id_Type)
)ENGINE=InnoDB;


#------------------------------------------------------------
# Table: Véhicule
#------------------------------------------------------------

CREATE TABLE Vehicule(
    id_Vehicule     Int  Auto_increment  NOT NULL ,
    Immatriculation Char (7) NOT NULL ,
    Age             Int NOT NULL ,
    Etat            Varchar (50) NOT NULL ,
    id_Agence       Int NOT NULL ,
    id_Type         Int NOT NULL ,
    id_Categorie    Int NOT NULL ,
    id_Modele       Int NOT NULL ,
    id_Marque       Int NOT NULL
      ,CONSTRAINT Vehicule_PK PRIMARY KEY (id_Vehicule)

      ,CONSTRAINT Vehicule_Agence_FK FOREIGN KEY (id_Agence) REFERENCES
Agence(id_Agence)
      ,CONSTRAINT Vehicule_Type0_FK FOREIGN KEY (id_Type) REFERENCES
Type(id_Type)
      ,CONSTRAINT Vehicule_Categorie1_FK FOREIGN KEY (id_Categorie) REFERENCES
Categorie(id_Categorie)
      ,CONSTRAINT Vehicule_Modele2_FK FOREIGN KEY (id_Modele) REFERENCES
```

```
Modele(id_Modele)
        ,CONSTRAINT Vehicule_Marque3_FK FOREIGN KEY (id_Marque) REFERENCES
Marque(id_Marque)
)ENGINE=InnoDB;


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

CREATE TABLE Client(
      id_Client   Int  Auto_increment  NOT NULL ,
      Nom         Varchar (50) NOT NULL ,
      Adresse     Varchar (50) NOT NULL ,
      Code_postal Int NOT NULL ,
      Ville       Varchar (50) NOT NULL
        ,CONSTRAINT Client_PK PRIMARY KEY (id_Client)
)ENGINE=InnoDB;


#------------------------------------------------------------
# Table: Location
#------------------------------------------------------------

CREATE TABLE Location(
      id_Vehicule  Int NOT NULL ,
      id_Client    Int NOT NULL ,
      Date_retrait Date NOT NULL ,
      Date_retour  Date NOT NULL
        ,CONSTRAINT a_loue_PK PRIMARY KEY (id_Vehicule,id_Client)

        ,CONSTRAINT a_loue_Vehicule_FK FOREIGN KEY (id_Vehicule) REFERENCES
Vehicule(id_Vehicule)
        ,CONSTRAINT a_loue_Client0_FK FOREIGN KEY (id_Client) REFERENCES
Client(id_Client)
)ENGINE=InnoDB;
```

Représentation simplifiée :

Véhicule (id_Vehicule, Immatriculation, Age, Etat, #id_Agence, #id_Type, #id_Catégorie, #id_Modèle, #id_Marque)
Agence (id_Agence, Nom, Nb_employés, #id_Pays)
Pays (id_Pays, Nom, Nombre_habitants, Superficie)
Client (id_Client, Nom, Adresse, Code_postal, Ville)
Location (#id_Vehicule, #id_Client, Date_retrait, Date_retour)
Type (id_Type, Libellé)
Modèle (id_Modèle, Dénomination, Puissance)
Catégorie (id_Catégorie, Libellé)
Marque (id_Marque, Nom)

Requêtes :

a.
```
SELECT Vehicule.id_Vehicule, Immatriculation, Age, Etat
FROM Vehicule
INNER JOIN Location ON Vehicule.id_Vehicule = Location.id_Vehicule
INNER JOIN Client ON Client.id_Client = Location.id_Client
WHERE Client.Nom = 'T122'
AND Location.Date_retour > '03/05/2023';
```

b.
```
SELECT *
FROM Location
INNER JOIN Client ON Client.id_Client = Location.id_Client
WHERE Client.Nom = 'T122';
```

c.
```
SELECT Immatriculation, Age, Etat
FROM Vehicule;
```

d.
```
SELECT Nom, Adresse, Code_Postal, Ville
FROM Client
WHERE Ville = 'Nice';
```

e.
```
SELECT Nom
FROM Client
ORDER BY Nom;
```

f.
```
ALTER TABLE 'Vehicule' ADD 'Kilometrage' INT NOT NULL;
SELECT *
FROM Vehicule
ORDER BY Kilometrage DESC;
```

g.
```
SELECT Client.id_Client, Nom, Adresse, Code_postal, Ville
FROM Client
INNER JOIN Location ON Client.id_Client = Location.id_Client
INNER JOIN Vehicule ON Vehicule.id_Vehicule = Location.id_Vehicule
WHERE Vehicule.Immatriculation = 'EW25EW';
```

h. On ne stocke pas la couleur des véhicules dans la base de données, il faut ajouter la colonne:
```
ALTER TABLE 'Vehicule' ADD 'Couleur' VARCHAR(50) NOT NULL;
SELECT *
FROM Vehicule
WHERE Couleur = 'Noir';
```

i.
```
SELECT *
FROM Vehicule
WHERE Kilometrage < 10000;
```

j.
```
SELECT *
FROM Location
WHERE Date_retrait < '01/01/2018';
```

k.
```
SELECT AVG(Kilometrage)
FROM Vehicule;
```

l.
```
SELECT *
FROM Location
WHERE Date_retrait >= '01/01/2018'
AND Date_retrait < '01/01/2019';
```

m.
```
SELECT COUNT(*)
FROM Vehicule
WHERE Kilometrage < 10000;
```

Partie 2 :

1)
```
SELECT Vehicule.id_Vehicule, Agence.id_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_retrait = Location.Date_retour;
```

2)
```
SELECT Marque.Nom, COUNT(*)
FROM Vehicule
INNER JOIN Marque ON Marque.id_Marque = Vehicule.id_Marque
GROUP BY id_marque;
```

3)
```
SELECT Client.Nom
FROM Client
INNER JOIN Location ON Location.id_Client = Client.id_Client
INNER JOIN Vehicule ON Vehicule.id_Vehicule = Location.id_Vehicule
INNER JOIN Marque ON Marque.id_Marque = Vehicule.id_Marque
WHERE Marque.Nom = 'Renault'
AND COUNT(Location.id_Vehicule) > 10;
```

4)
```
SELECT COUNT(Agence.id_Agence), SUM(Agence.Nb_employés),      Pays.Nom
FROM Agence
INNER JOIN Pays ON Pays.id_Pays = Agence.id_Pays
GROUP BY Pays.id_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 NomMat = 'BD';
```

3.
```
SELECT DISTINCT NomMat, CoefMat
FROM MATIERE
INNER JOIN ENSEIGNANT ON ENSEIGNANT.CodeMat = MATIERE.CodeMat
WHERE GradeEns = 'Grd3';
```

4.
```
SELECT NomMat, CoefMat
FROM MATIERE
INNER JOIN NOTE 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 NuméroCoureur, NomCoureur, CodePays
FROM COUREUR
INNER JOIN EQUIPE ON COUREUR.CodeEquipe = EQUIPE.CodeEquipe
WHERE NomEquipe = 'Festina';
```

2.
```
SELECT SUM(NbKm)
FROM ETAPE;
```

3.
```
SELECT SUM(NbKm)
FROM ETAPE
INNER JOIN TYPE ON ETAPE.CodeType = TYPE.CodeType
WHERE LibelleType = 'Haute Montagne';
```

4.
```
SELECT NomCoureur
FROM COUREUR
WHERE NuméroCoureur NOT IN (
    SELECT NuméroCoureur
    FROM BONIFICATION
);
```

5.
```
SELECT NomCoureur
FROM COUREUR
INNER JOIN PARTICIPER ON COUREUR.NuméroCoureur =
    PARTICIPER.NuméroCoureur
GROUP BY NomCoureur
HAVING COUNT(NomCoureur) = (
    SELECT COUNT(NuméroEtape)
    FROM ETAPE
);
```

6.
```
SELECT NomCoureur, CodeEquipe, CodePays, SUM(NbSecondes)
FROM COUREUR
INNER JOIN BONIFICATION ON BONIFICATION.NuméroCoureur =
    COUREUR.NuméroCoureur
WHERE COUREUR.NuméroCoureur IN (
    SELECT DISTINCT NuméroCoureur
    FROM PARTICIPER
    WHERE NuméroEtape <= 13
)
ORDER BY SUM(NbSecondes);
```

7.
```
SELECT NomEquipe, SUM(NbSecondes)
INNER JOIN COUREUR ON COUREUR.CodeEquipe = EQUIPE.CodeEquipe
INNER JOIN BONIFICATION ON BONIFICATION.NuméroCoureur =
    COUREUR.NuméroCoureur
GROUP BY CodeEquipe
WHERE COUREUR.NuméroCoureur IN (
    SELECT DISTINCT NuméroCoureur
    FROM PARTICIPER
    WHERE NuméroEtape <= 13
```

```
)
ORDER BY SUM(NbSecondes);
```

Exercice 4 :

1.
```
SELECT *
FROM Client
WHERE adressecli LIKE '%Marrakech%;
```

2.
```
SELECT Numprod, désignation, prix
FROM Produit
ORDER BY prix desc;
```

3.
```
SELECT Nomvendeur, adresse_vend
FROM Vendeur
WHERE Nomvendeur LIKE 'M%';
```

4.
```
SELECT *
FROM Commande
INNER JOIN Vendeur ON Commande.Idvendeur = Vendeur.Idvendeur
WHERE Nomvendeur = 'Mohammed'
AND date_com BETWEEN '01/01/2020' AND '30/01/2020';
```

5.
```
SELECT COUNT(*)
FROM Commande
WHERE Numprod = 365;
```

Exercice 5 :

1.
```
SELECT *
FROM Etudiant;
```

2.
```
SELECT nom_matière, coefficient
FROM Matière;
```

3.
```
SELECT numéro_carte_etudiant
FROM Note
WHERE note_examen BETWEEN 7 AND 12;
```

4.
```
SELECT *
FROM Etudiant
WHERE Nom LIKE 'ben%';
```

5.
```
SELECT COUNT(*)
FROM Note
WHERE code_matière = 12518;
```

6.
```
SELECT SUM(coefficient)
FROM Matière;
```

7.
```
SELECT Nom
FROM Etudiant
INNER JOIN Note ON Etudiant.numéro_carte_etudiant = Note.numéro_carte_etudiant
WHERE note_examen > 10;
```

8.
```
SELECT nom_matière, coefficient
FROM Matière
INNER JOIN Note ON Note.code_matière = Matière.code_matière
WHERE numéro_carte_etudiant = 01234568;
```