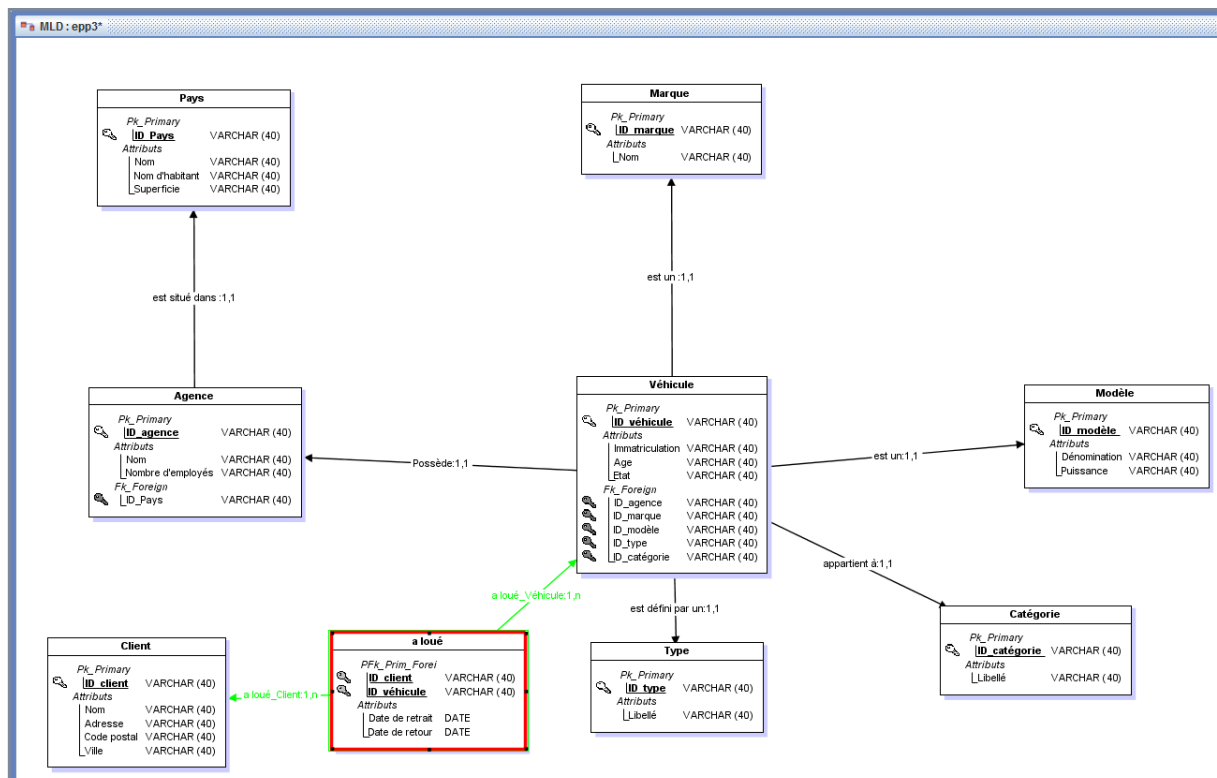
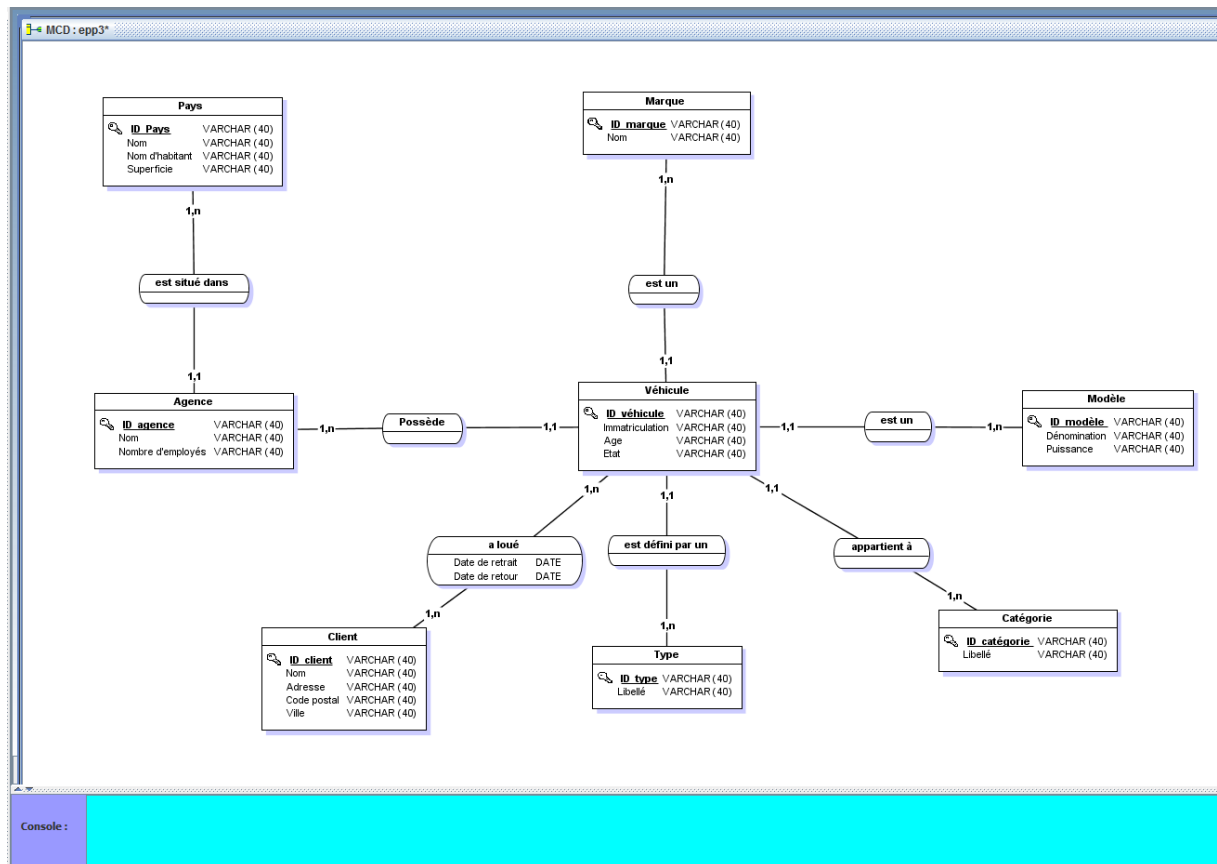


Rimbert quentin EPP

Je n'ai pas réussi à faire la case total alors j'ai mis un nombre générer aléatoire pour montrer que j'arrivais à faire le reste des calculs en JavaScript



-Pays (ID_Pays, Nom, Nombre d'habitant, Superficie)

Agence (ID_agence, Nom, Nombre d'employés, #ID_Pays)

-Véhicules (ID_véhicule, Immatriculation, Age, Etat, #ID_agence, #ID_marque, #ID_modèle, #ID_type, #ID_catégorie)

-marque (ID_marque, Nom)

-modèle (ID_modèle, Dénomination, Puissance)

-Catégorie (ID_catégorie, Libellé)

-Type (ID_type, Libellé)

-a loué (#ID_client, #ID_véhicule, Date de retrait, Date de retour)

-Client (ID_Client, Nom, Adresse, Code postal, Ville)

A. Afficher toutes les informations sur les véhicules loués par le Client n°T122

```
SELECT Véhicules *  
FROM Véhicules V  
Inner join Loue L on V.ID_Véhicule = L.ID_véhicules  
WHERE L.ID_client = 'T122'
```

B. Afficher toutes les locations réalisées par le client n° T122

```
SELECT *
```

```
FROM `a_loue`
```

```
WHERE `ID_client`= 'T122'
```

C. Afficher l'immatriculation, l'âge et l'état de tous les véhicules.

```
SELECT `Immatriculation`,`Age`,`Etat`
```

```
FROM `vehicule`
```

D. Afficher les noms des clients et les adresses, des clients qui habitent à

<< Nice >>.

```
SELECT `Nom`,`Adresse`
```

```
FROM `client`
```

```
WHERE `Ville`= 'Nice'
```

E. Affiche la liste des clients par ordre alphabétique croissant des noms

```
SELECT *  
  
FROM `client`  
  
Order by `Nom` ASC
```

F. Affiche la liste des voitures par ordre décroissant desCompteurs (kilométrage)

```
SELECT *  
  
FROM `vehicule`  
  
Order by `Kilometrage` desc
```

G. Afficher les informations sur les clients qui ont loué la voiture EW 25EW

```
SELECT DISTINCT*  
  
FROM client C  
  
Inner join Loue L on C.ID_client = L.ID_client  
  
Inner join Véhicule V on L.ID_véhicule = V.ID_véhicule
```

```
WHERE immatriculation= 'EW25EW'
```

H. Afficher toutes les voitures noires :

```
SELECT *  
  
FROM `vehicule`  
  
WHERE couleur= 'noires'
```

I. Afficher toutes les voitures ayant un kilométrage <10000 km

```
SELECT *  
  
FROM `vehicule`  
  
WHERE `Kilometrage` <10000
```

J. Afficher toutes les informations sur les locations réalisées avant 2018

```
SELECT *  
FROM loue  
  
WHERE `Date_de_retrait` < '2018-01-01'
```

K. Afficher la moyenne des kilométrages de tous les véhicules du parc.

```
SELECT AVG ('Kilometrage')
```

FROM Véhicule

L. Afficher toutes les locations réalisées en 2018

SELECT *

FROM `a_loue`

WHERE `Date_de_retrait`

BETWEEN 2018-01-01

AND 2018-12-31

M. Afficher le nombre de voitures ayant un kilométrage <10 000 kilomètres

SELECT COUNT(*)

FROM `vehicule`

WHERE `Kilometrage`<1000

Partie 2

1°

SELECT V.ID_véhicule, A.nom, L.Date_de_retrait, L.date_de_retour

FROM Véhicule V

Inner join Loue L on V.ID_véhicule = L.ID_véhicule

Inner join Agence A on A.A.ID_agence= V.ID_agence

WHERE L.date_de_retrait = L.date_de_retour

2°

SELECT M.nom, COUNT(*)

FROM Véhicule v

Inner join Marque M on V.ID_marque = M.ID_marque

GROUP BY M.nom

3°

SELECT C.nom, COUNT(*)

FROM Véhicule V

Inner join Marque M on V.ID_marque = M.ID_marque

Inner join Client C on L.ID_client = C.ID_client

WHERE M.nom = 'Renault'

GROUP BY M.nom

HAVING COUNT(*)>10

4°

SELECT P.nom; Nombre_d'employés, COUNT(A.ID_agence) AS « Nombre agence »

FROM Agence A

Inner join Pays P on A.ID_pays = P.ID_pays

GROUP BY P.nom

Ex2

ETUDIANT (CodeEt, NomEt, DatnEt)

MATIERE(CodeMat, NomMat, CoefMat)

ENSEIGNANT (CodeEns, NomEns, GradeEns, #CodeMat)

NOTE (#CodeEt, #CodeMat, note)

1°

SELECT *

FROM ETUDIANT

Order by `NomEt` ASC ;

2°

SELECT E.NomEns, E.GradeEns

FROM ENSEIGNANT E

Inner join MATIERE M on M.CodeMat = E.CodeMat

Where NomMat= 'BD'

3°

SELECT DISTINCT M.NomMat, M.CoefMat

FROM MATIERE M

Inner join ENSEIGNANT E on M.CodeMat = E.CodeMat

Where GradeEns= 'Grd3'

4°

SELECT M.NomMat, M.CoefMat

FROM MATIERE M

Inner join ETUDIANT

5°

SELECT COUNT(*)

FROM ENSEIGNANT

Where NomMat = 'Informatique'

Ex3

EQUIPE(CodeEquipe, NomEquipe, DirecteurSportif)

COUREUR(NuméroCoureur, NomCoureur, #CodeEquipe, #CodePays)

PAYS(CodePays, NomPays)

TYPE_ETAPE(CodeType, LibelleType)

ETAPE(NuméroEtap, DateEtape, VilleDép, VilleArr, NbKm, #CodeType

PARTICIPER(#NuméroCoureur, #NuméroEtape, TempsRealisé)

ATTRIBUER_BONIFICATION(#NuméroEtape, #NuméroCoureur, km, Rang,

NbSecondes)

1°

```
SELECT C.NuméroCoureur, C.NomCoureur, P.CodePays
```

```
FROM EQUIPE E
```

```
Inner join COUREUR C on E.CodeEquipe = C.CodeEquipe
```

```
Inner join PAYS P on P.CodePays = C.CodePays
```

```
WHERE E.NomEquipe = 'Festina'
```

2°

```
SELECT SUM(nbkm) as « Nombre de kilomètres total du tour de France 97
```

```
FROM ETAPE
```

3°

```
SELECT SUM(E.nbkm) as « Nombre de kilometers total des étapes de type Haute Montagne »
```

```
FROM ETAPE E
```

```
Inner join TYPE_ETAPE T on T.CodeType = E.CodeType
```

```
WHERE T.LibelleType = 'Haute Montagne'
```

4°

```
SELECT C.NomCoureur
```

```
FROM COUREUR C
```

```
WHERE C.NuméroCoureur not in (
```

```
SELECT A.NuméroCoureur
```

```
FROM ATTRIBUER_BONIFICATION AB
```

```
)
```

5°

```
SELECT C.NomCoureur
```

```
FROM Coureur C
```

```
WHERE NO
```

```

EXISTS (

SELECT

FROM Etape E

WHERE E.NuméroEtape not in (

SELECT P.NuméroEtape

FROM Participer P

WHERE P.NuméroCoureur = C.NuméroCoureur

))

```

6°

```

SELECT C.NomCoureur, C.CodeEquipe, C.CodePays ; SUM(P.Tempsrealisé + ab.NbSeconde) as Total

FROM COUREUR C

Inner join PARTICIPER P on C.NuméroCoureur = P.NuméroCoureur

Inner join ATTRIBUER_BONIFICATION AB on P.NuméroCoureur=AB.NuméroCoureur

And p.NuméroEtape = AB.NuméroEtape

Inner join ETAPE E on E.NuméroEtape = P.NuméroEtape

WHERE E.NuméroEtape <=13

GROUP BY C.NuméroCoureur

ORDER BY Total

```

7°

```

SELECT NomEquipe, SUM(Tempsrealisé + NbSeconde) as Total

FROM EQUIPE E

Inner join COUREUR C on C.CodeEquipe = E.CodeEquipe

Inner join PARTICIPER P on P.NuméroCoureur = C.NuméroCoureur

LEFT JOIN ATTRIBUER_BONIFICATION AB on AB.NuméroCoureur = C.NuméroCoureur

```


GROUPE BY NomEquipe

ORDER BY Total

Ex 4

Client (Numcli, Nomcli, Prenomcli, adressecli, mailcli)

Produit (Numprod, désignation, prix , qte_stock)

Vendeur (Idvendeur, Nomvendeur, adresse_vend)

Commande (Numcom, #Numcli, #Idvendeur, #Numprod, date_com, qte_com)

1°

SELECT *

FROM Client

WHERE adressecli= 'Marrakech'

2°

SELECT Numprod, désignation, prix

FROM Produit

Order By prix DESC

3°

SELECT Nomvendeur, adresse

FROM Vendeur

WHERE Nomvendeur LIKE 'M%'

4°

SELECT C.*

FROM Commande C

Inner join Vendeur V on C.IdVendeur= V.IdVendeur

Where NomVendeur = 'Mohammed'

And date_com between '2020-01-01' and '2020-01-30'

5°

```
SELECT COUNT(*) as « Nombre de commande contenant le produit n°365 »
```

```
FROM Commande
```

```
WHERE Numprod= 365
```

Ex 5

1°

```
SELECT *
```

```
FROM Etudiant
```

2°

```
SELECT coefficient, nom_matière
```

```
FROM Matière
```

3°

```
SELECT numero_carte_etudiant
```

```
FROM Etudiant
```

```
WHERE numero_carte_etudiant
```

```
IN (
```

```
SELECT numero_carte_etudiant
```

```
FROM Note
```

```
GROUP BY
```

```
HAVING AVG (note_examen)
```

```
BETWEEN 7
```

```
AND 12
```

```
)
```

4°

```
SELECT *
```

FROM Etudiant

WHERE nom LIKE 'ben%'

5°

SELECT COUNT(*) AS "Nombre d'étudiants qui ont comme matière 12518 »

FROM (

SELECT DISTINCT numero_carte_etudiant

FROM Note

WHERE code_matiere = '12518')

AS etudiant

6°

SELECT SUM(coefficient) as "total coefficient"

FROM Matière

7°

SELECT Nom

FROM Etudiant E

Inner join Note N on E.numéro_carte_etudiant = N. numéro_carte_etudiant

WHERE note_examen >10

8°

SELECT nom_matière, coefficient

FROM Matière M

Inner join Note N on N.code_matière=M.code_matière

Inner join Etudiant E on E.numéro_carte_etudiant = N. numéro_carte_etudiant

WHERE E.numéro_carte_etudiant = '012345678'