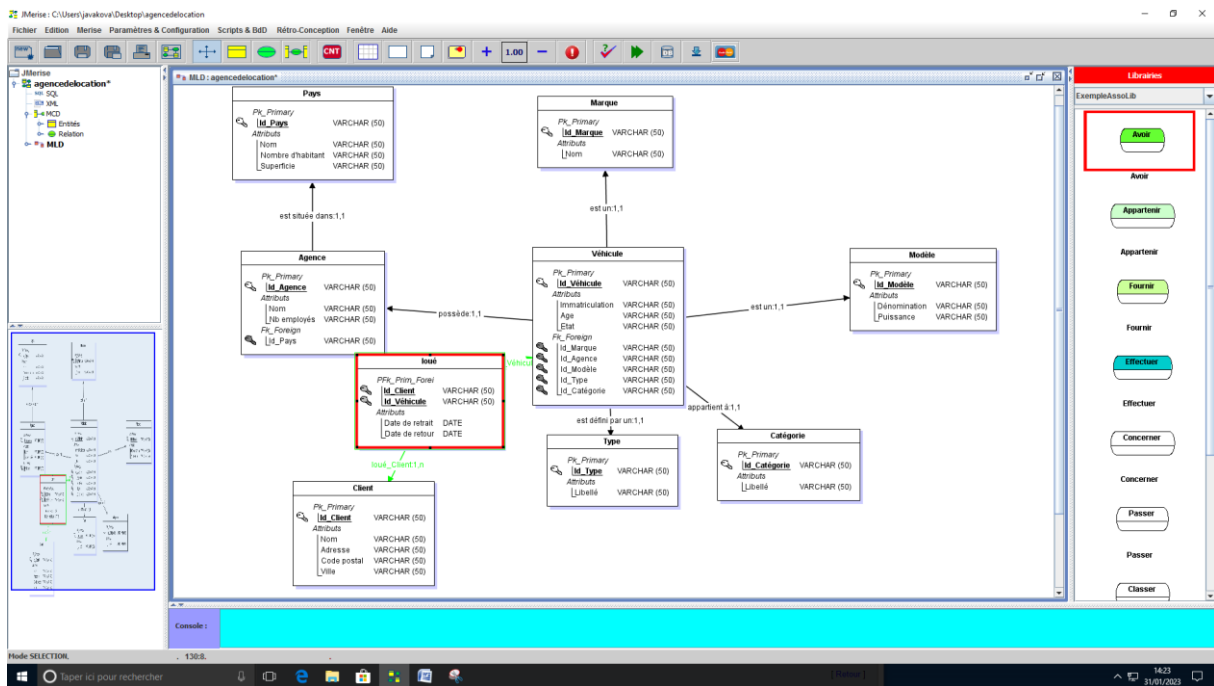
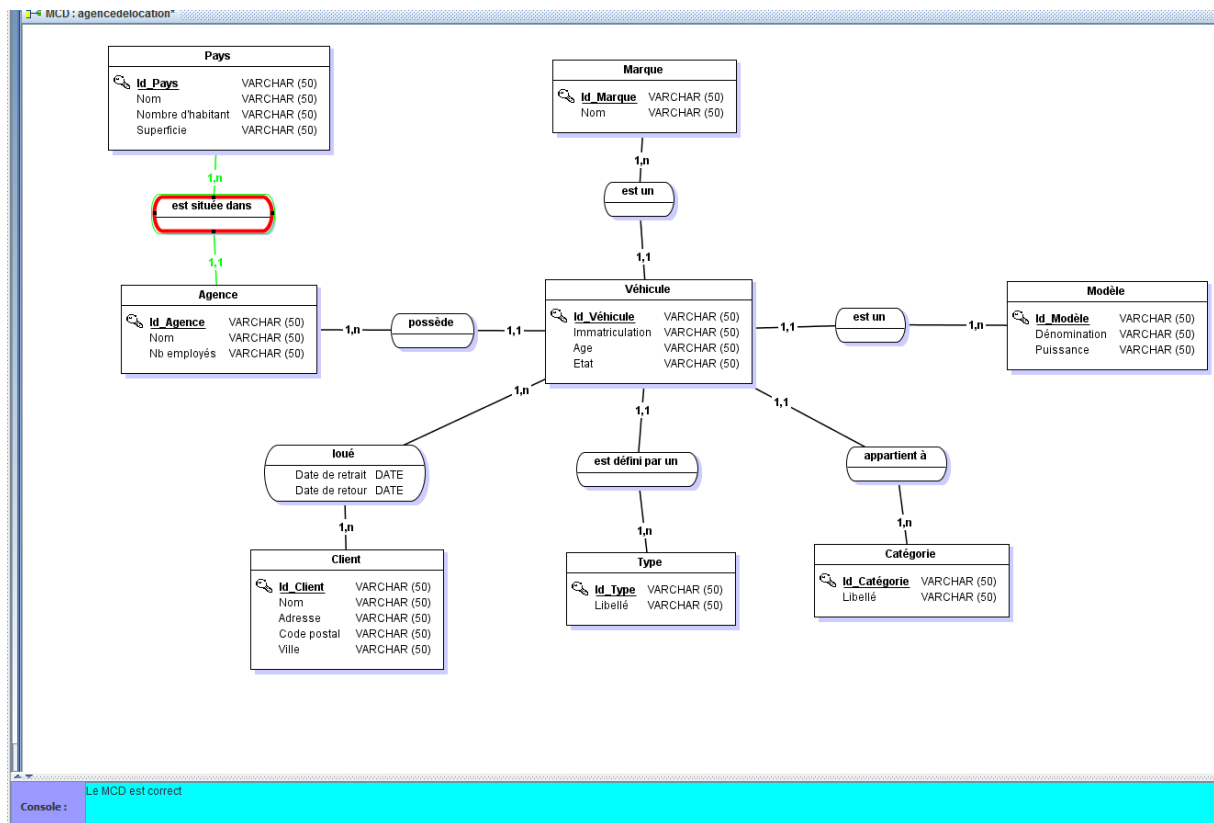
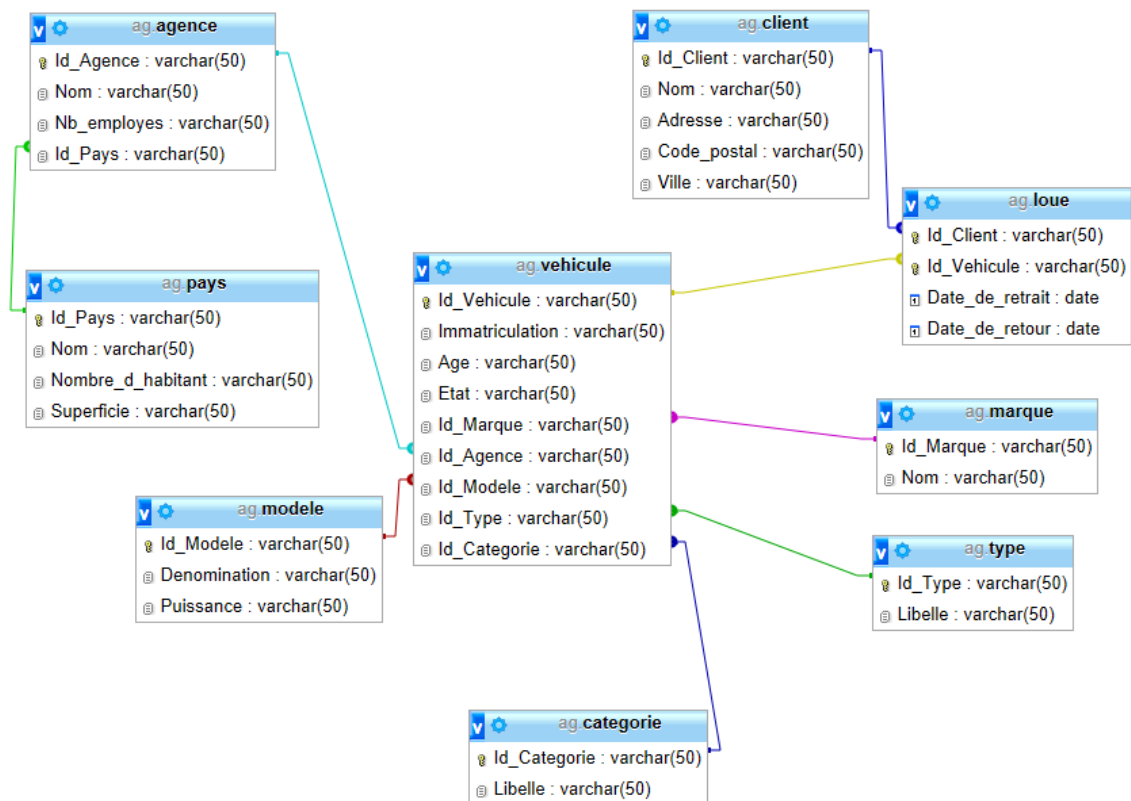


Avakova Jasmine 1 BTS SIO B





Pays (**Id Pays**, Nom, Nombre d'habitants, Superficie)

Agence (**Id Agence**, Nom, Nb employées, #Id_Pays)

Loué (#Id_Client, #Id_Véhicule, Date de retrait, Date de retour)

Clients (**Id Client**, Nom, Adresse, Code postal, Ville)

Marque (**Id Marque**, Nom)

Véhicules (**Id Véhicule**, Immatriculation, Age, Etat, #Id_marque, #Id_Agence, #Id_Modèle, #Id_Type, #Id_Catégorie)

Type (**Id Type**, Libellé)

Catégorie (**Id Catégorie**, Libellé)

Modèle (**Id modèle**, Dénomination, Puissance)

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

```

SELECT v . *
FROM `vehicule` v
INNER JOIN a_loue al ON v.id_vehicule = al.id_vehicule
WHERE al.id_client = "T122"
LIMIT 0 , 30
  
```

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

```
SELECT v.Immatriculation
FROM `vehicule` v
INNER JOIN a_loue al ON v.id_vehicule = al.id_vehicule
WHERE al.id_client = "T122"
LIMIT 0 , 30
```

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

```
SELECT Immatriculation, Age, Etat
FROM `vehicule`
LIMIT 0 , 30
```

- d. Afficher les noms des clients et les adresses, des clients qui habitent à << Nice >>.

```
SELECT `Nom` , `Adresse`
FROM `client`
WHERE `Ville` = "Nice"
LIMIT 0 , 30
```

- e. Affiche la liste des clients par ordre alphabétique croissant des noms

```
SELECT *
FROM client
ORDER BY nom ASC
LIMIT 0 , 30
```

- f. Affiche la liste des voitures par ordre alphabétique décroissant des compteurs (kilométrage)

```
SELECT *
FROM vehicule
ORDER BY `kilometrage` DESC *
LIMIT 0 , 30
```

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

```
SELECT c.*
FROM `client` c
INNER JOIN a_loue al ON c.id_Client = al.id_client
INNER JOIN vehicule v ON al.id_vehicule = v.id_vehicule
WHERE v.immatriculation = "EW25EW"
LIMIT 0 , 30
```

- h. Afficher toutes les voitures noires :)

```
SELECT *
FROM `vehicule`
WHERE `couleur` = 'noir'
LIMIT 0 , 30
```

- i. Afficher toutes les voitures ayant un kilométrage <10000

```
SELECT *
FROM `vehicule`
WHERE `kilometrage` <10000
LIMIT 0 , 30
```

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

```
SELECT *  
FROM a_loue  
WHERE `Date_de_retrait` < '2018-01-01'  
LIMIT 0 , 30
```

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

```
SELECT AVG( `kilometrage` )  
FROM vehicule
```

L. Afficher toutes les locations réalisées en 2018

```
SELECT *  
FROM a_loue  
WHERE `Date_de_retrait`  
BETWEEN '2018-01-01'  
AND '2018-12-31'  
LIMIT 0 , 30
```

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

```
SELECT COUNT( * )  
FROM `vehicule`  
WHERE `kilometrage` <10000
```

Partie 2 •

Obtenir la liste des véhicules empruntés et rendu le même jour ainsi que l'agence de rattachement

```
SELECT Agence.Nom, Immatriculation  
FROM Agence  
JOIN Véhicule  
USING ( Id_Agence )  
JOIN A_loué  
USING ( Id_Véhicule )  
WHERE Date_de_retrait = Date_de_retour  
LIMIT 0 , 30
```

Obtenir le nombre véhicules pour chaque marque

```
SELECT Marque.Nom, COUNT( * )  
FROM Véhicule  
JOIN Marque  
USING ( Id_Marque )  
GROUP BY Marque.Nom  
LIMIT 0 , 30
```

Obtenir les noms des clients qui ont loué plus de 10 véhicules de marque « Renault »

```
SELECT Client.Nom  
FROM Client  
JOIN A_loué  
USING ( Id_Client )
```

```

JOIN Véhicule
USING ( Id_Véhicule )
JOIN Marque
USING ( Id_Marque )
WHERE Marque.Nom = "Renault"
GROUP BY Client.Nom
HAVING COUNT( Id_Véhicule ) >10
LIMIT 0 , 30

```

Obtenir le nombre d'agences et d'employés par pays.

```

SELECT COUNT( Id_Agence ) , SUM( Nb_employés ) , Pays.Nom
FROM Agence
JOIN Pays
USING ( Id_Pays )
GROUP BY Pays.Nom
LIMIT 0 , 30

```

Exercice 2

ETUDIANT(CodeEt, NomEt, DatnEt)•
 MATIERE(CodeMat, NomMat, CoefMat)•
 ENSEIGNANT(CodeEns, NomEns, GradeEns, #CodeMat)•
 NOTE(#CodeEt, #CodeMat, note)•

1. Les informations relatives aux étudiants (Code, Nom et Date de naissance) selon l'ordre alphabétique croissant du nom

```

SELECT *
FROM ETUDIANT
ORDER BY NomEt ASC
LIMIT 0 , 30

```

2. Les noms et les grades des enseignants de la matière dont le nom est 'BD'.

```

SELECT E.NomEns , E.GradeEns
FROM ENSEIGNANT E
INNER JOIN MATIERE M ON M.CodeMat = E.CodeMat
WHERE M.NomMat = "BD"
LIMIT 0 , 30

```

3. La liste distincte formée des noms et les coefficients des différentes matières qui sont enseignées par des enseignants de grade 'Grd3'.

```

SELECT DISTINCT (
M.NomMat
) , M.CoefMat
FROM ENSEIGNANT E
INNER JOIN MATIERE M ON M.CodeMat = E.CodeMat
WHERE E.GradeEns = "Grd3"
LIMIT 0 , 30

```

4. La liste des matières (Nom et Coefficient) qui sont suivies par l'étudiant de code 'Et321'.

```
SELECT M.NomMat, M.CoeffMat
FROM MATIERE M
INNER JOIN NOTE N ON M.CodeMat = N.CodeMat
INNER JOIN ETUDIANT E ON E.CodeEt = N.CodeEt
WHERE E.CodeEt = "Et321"
LIMIT 0 , 30
```

5. Le nombre d'enseignants de la matière dont le nom est 'Informatique

```
SELECT COUNT( * )
FROM ENSEIGNANT E
INNER JOIN MATIERE M ON M.CodeMat = E.CodeMat
WHERE M.NomMat = "Informatique"
```

Exercice 3 :

EQUIPE(CodeEquipe, NomEquipe, DirecteurSportif) COUREUR(NuméroCoureur, NomCoureur, #CodeEquipe, #CodePays)
 PAYS(CodePays, NomPays)
 TYPE_ETAPE(CodeType, LibelleType)
 ETAPE(NuméroEtap, DateEtap, VilleDép, VilleArr, NbKm, #CodeType)
 PARTICIPER(#NuméroCoureur, #NuméroEtap, TempsRealisé)
 ATTRIBUER_BONIFICATION(#NuméroEtap, #NuméroCoureur, km, Rang, NbSecondes)

Exprimez en SQL les requêtes suivantes :

1. Quelle est la composition de l'équipe Festina (Numéro, nom et pays des coureurs) ?

```
SELECT NumeroCoureur, NomCoureur, NomPays
FROM EQUIPE A, COUREUR B, PAYS C
WHERE A.CodeEquipe = B.CodeEquipe
AND B.CodePays = C.CodePays
AND NomEquipe = "FESTINA"
LIMIT 0 , 30
```

2. Quel est le nombre de kilomètres total du Tour de France 97 ?

```
SELECT SUM( Nbkm )
FROM ETAPE
```

3. Quel est le nombre de kilomètres total des étapes de type "Haute Montagne" ?

```
SELECT SUM( Nbkm )
FROM ETAPE A, TYPE_ETAPE B
WHERE A.CodeType = B.CodeType
AND LibelleType = "HAUTE MONTAGNE"
```

4. Quels sont les noms des coureurs qui n'ont pas obtenu de bonifications ?

```
SELECT NomCoureur
FROM COUREUR
WHERE NumeroCoureur NOT
IN (
```

```
SELECT NumeroCoureur
FROM ATTRIBUER_BONIFICATION
)
LIMIT 0 , 30
```

5. Quels sont les noms des coureurs qui ont participé à toutes les étapes ?

```
SELECT NomCoureur
FROM PARTICIPER A, COUREUR B
WHERE A.NumeroCoureur = B.NumeroCoureur
GROUP BY NomCoureur
HAVING COUNT( * ) = (
SELECT COUNT( * )
FROM ETAPE )
LIMIT 0 , 30
```

6. Quel est le classement général des coureurs (nom, code équipe, code pays et temps des coureurs) à l'issue des 13 premières étapes sachant que les bonifications ont été intégrées dans les temps réalisés à chaque étape ?

```
SELECT NomCoureur, CodeEquipe, CodePays, SUM( TempsRealise ) AS Total
FROM PARTICIPER A, COUREUR B
WHERE A.NumeroCoureur = B.NumeroCoureur
AND NumeroEtape <=13
GROUP BY NomCoureur, CodeEquipe, CodePays
ORDER BY Total
LIMIT 0 , 30
```

7. Quel est le classement par équipe à l'issue des 13 premières étapes (nom et temps des équipes) ?

```
SELECT NomEquipe, SUM( TempsRealise ) AS Total
FROM PARTICIPER A, COUREUR B, EQUIPE C
WHERE A.NumeroCoureur = B.NumeroCoureur
AND B.CodeEquipe = C.CodeEquipe
AND NumeroEtape <=13
GROUP BY NomEquipe
ORDER BY Total
LIMIT 0 , 30
```

Exercice 4 : La représentation textuelle suivante est une description simplifiée d'une base de données de gestion de facturation d'une entreprise commerciale.

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)

Exprimer en SQL les requêtes suivantes : Créer les tables : Client, Produit, Vendeur et Commande.

```
CREATE TABLE Client(
    Numcli varchar(10) primary key ,
```

```

        Nomcli varchar(10) ,
        Prenomcli varchar(10) ,
        adressecli varchar(20) ,
        mailcli varchar(20) ,
    );
CREATE TABLE Produit(
    Numprod varchar primary key ,
    designation varchar(10),
    prix float,
    qte_stock int default 0
);

CREATE TABLE Vendeur(
    idvendeur varchar(10) primary key ,
    Nomvendeur varchar(10) ,
    adresse_vend varchar(10) ,
);
CREATE TABLE commande(
    Numcom varchar(10) primary key ,
    Numcli int ,
    idvendeur varchar(10) ,
    Numprod varchar(10),
    date_com date,
    qte_com varchar(10),
    FOREIGN KEY(Numcli) REFERENCES Client(Numcli),
    FOREIGN KEY(idvendeur) REFERENCES Vendeur(idvendeur),
    FOREIGN KEY(Numprod) REFERENCES Produit(Numprod)
);

```

1. la liste des clients de Marrakech.

```

SELECT *
FROM Client
WHERE adressecli LIKE "%marrakech%"
LIMIT 0 , 30

```

2. la liste des produits (Numprod, désignation, prix) classés de plus cher au moins cher.

```

SELECT Numprod, designation, prix
FROM Produit
ORDER BY prix ASC
LIMIT 0 , 30

```

3. noms et adresses des vendeurs dont le nom commence par la lettre 'M'

```

SELECT Nomvendeur, adresse_vend
FROM Vendeur
WHERE Nomvendeur LIKE "M%"
LIMIT 0 , 30

```


4. la liste des commandes effectuées par le vendeur "Mohammed" entre le 1er et 30 janvier 2020.

```
SELECT Nymcom, Numcli, Idvendeur, Numprod, date_com, qte_com
FROM Commande cmd, Vendeur vend
WHERE cmd.Idvendeur = vend.Idvendeur
AND vend.Nomvendeur = "mohammed"
AND cmd.date_com
BETWEEN "2020-01-01"
AND "2020-01-30"
LIMIT 0 , 30
```

5. le nombre des commandes contenant le produit n° 365.

```
SELECT COUNT( * )
FROM Commande
WHERE Numprod =365
LIMIT 0 , 30
```

Exercice 5 : Soit la base de données suivante :

Ecrire les commandes SQL permettant de rechercher :

1. La liste de tous les étudiants.

```
SELECT *
FROM Etudiant
LIMIT 0 , 30
```

2. Nom et coefficient des matières.

```
SELECT nom_matiere, coefficient
FROM Matiere
LIMIT 0 , 30
```

3. Les numéros des cartes d'identité des étudiants dont la moyenne entre 7 et 12.

```
SELECT numero_carte_etudiant
FROM Note, Matiere mat
WHERE Note.code_matiere = Mat.code_matiere
GROUP BY numero_carte_etudiant
HAVING (
SUM( note_examen * coefficient ) / SUM( coefficient )
)
BETWEEN 7
AND 12
LIMIT 0 , 30
```

4. La liste des étudiants dont le nom commence par 'ben'.

```
SELECT *
FROM Etudiant
```

```
WHERE Nom LIKE "Ben%"
LIMIT 0 , 30
```

5. Le nombre des étudiants qui ont comme matière '12518'.

```
SELECT *
FROM Note
WHERE code_matiere =12518
LIMIT 0 , 30
```

6. La somme des coefficients des matières.

```
SELECT SUM( coefficient )
FROM Matiere
```

7. Les noms des étudiants qui une note_examen >10.

```
SELECT DISTINCT Nom
FROM Note, Etudiant
WHERE Note.numero_carte_etudiant = Etudiant.numero_carte_etudiant
AND note_examen >10
LIMIT 0 , 30
```

8- Afficher les noms et les coefficients des matières étudiées par l'étudiant "01234568".

```
SELECT nom_matiere, coefficient
FROM Note, Matiere
WHERE Note.numero_carte_etudiant = "01234568"
LIMIT 0 , 30
```