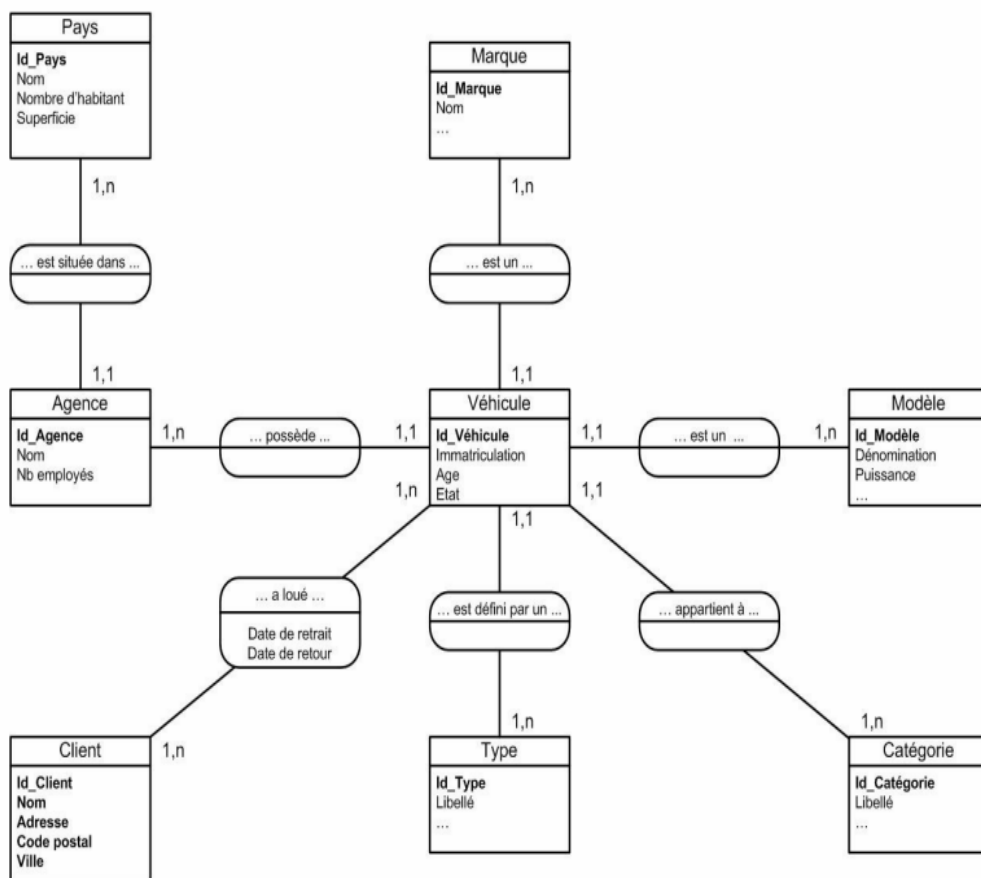


Mission 3 : Les bases de données

Tout le travail demandé doit être réalisé sur un fichier doc et déposé sur votre site Rubrique Mission sous format PDF. Travail de Verger—Doucy Brice

Dessiner ce MCD du logiciel d'une agence de location de véhicules sur un outil de conception (Jmerise/ PowerAmc...)

Voici le MCD du logiciel d'une agence de location de véhicules.

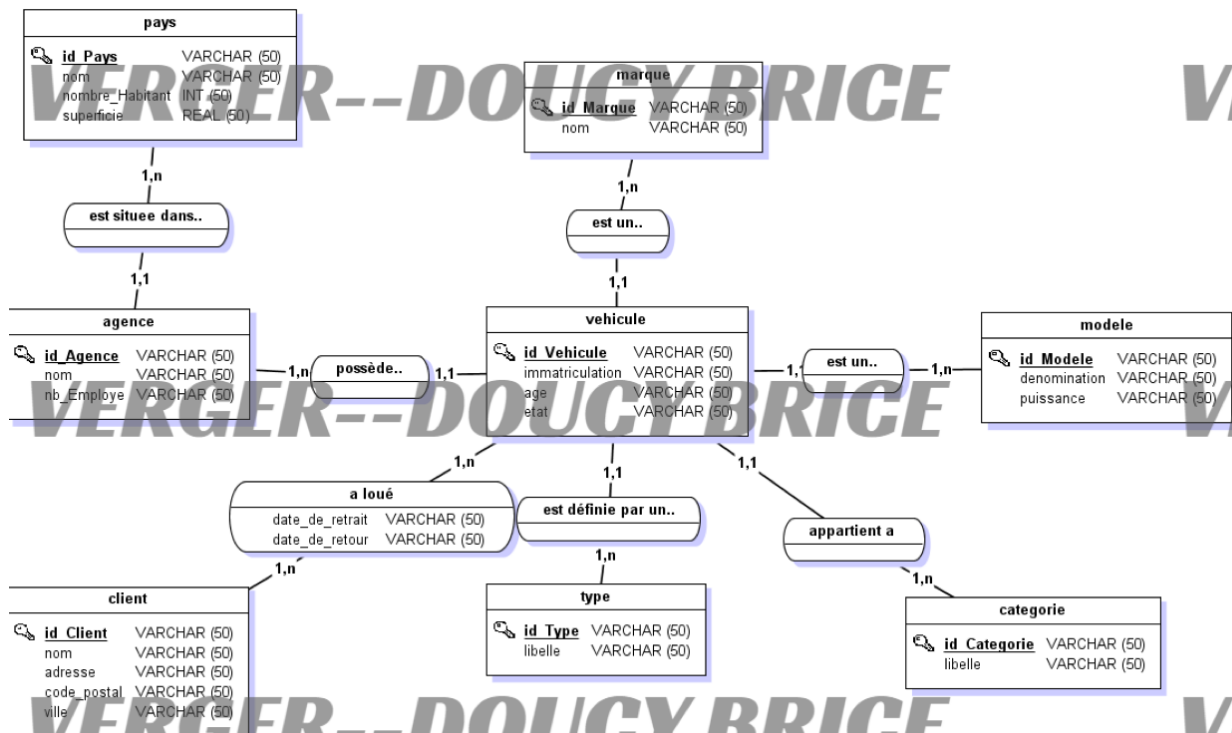


Produisez le MLD et le code sql de votre base de

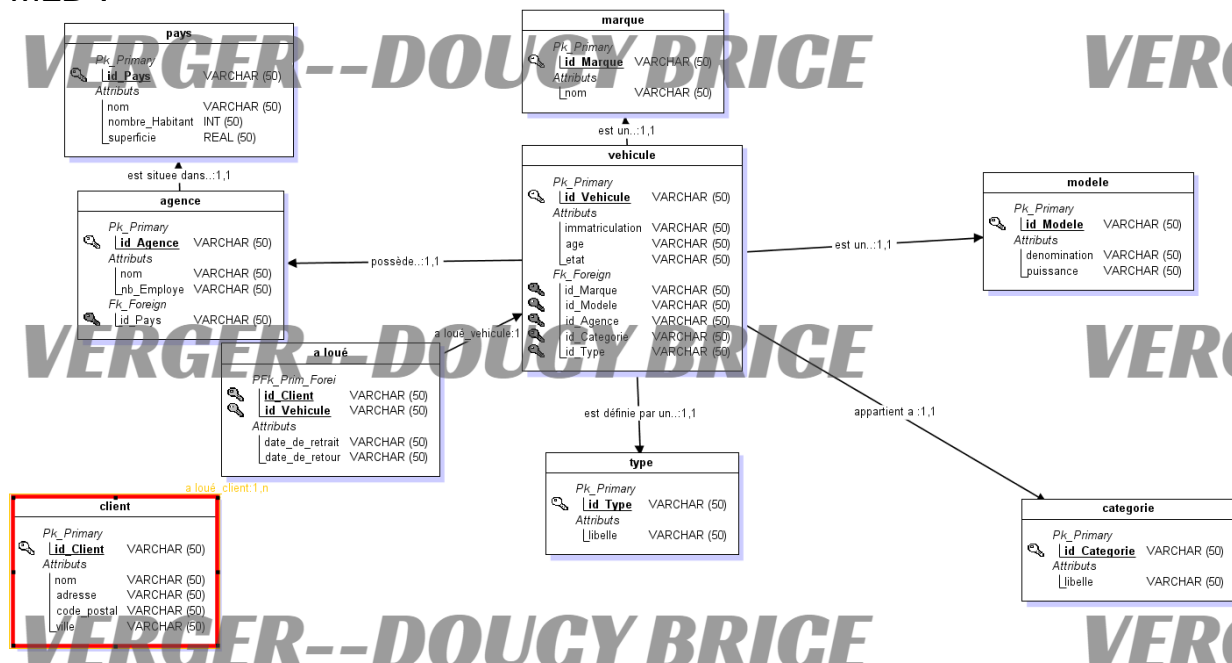
données. Importer la BD sur easyPhp

Enregistrer les captures écran du MCD et du MLD, les copier sur votre fichier doc.

MCD :



MLD :



Traduire le MLD Graphique en un représentation textuelle simplifiée d'une base de données

Voici un exemple :

- Représentation (Num_Rep, titre_Rep, lieu)
- Musicien (Num_mus, nom, #Num_Rep)
- Programmeur (Date, #Num_Rep, tarif)

::

pays(**id Pays**,nom,nb_Habitant,superficie)
marque(**id Marque**,nom)
agence(**id Agence**,nom,nb_Employe,#id_Pays)
modele(**id Modele**,denomination,puissance)
client(**id Client**,nom,adresse,code_postal,ville)
type(**id Type**,libelle)
categorie(**id Categorie**,libelle)
vehicule(**id Vehicule**,immatriculation,age,etat,**#id Marque,#id Modele,#id Agence,#id Categorie,#id Type**)
a_loue(**id Client**,nom,adresse,code_postal,ville)

Travail de Verger—Doucy Brice

Réalisez les requêtes suivantes :

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

```
SELECT V.*
FROM `vehicule` V
INNER JOIN a_loue A ON V.id_Vehicule = A.id_Vehicule
WHERE A.id_Client = "T122"
```

Travail de Verger—Doucy Brice

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

```
SELECT V.immatriculation
FROM `vehicule` V
INNER JOIN a_loue A ON V.id_Vehicule = A.id_Vehicule
WHERE A.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. Ajouter l'attribut kilométrage et Afficher la liste des voitures par ordre décroissant des compteurs (kilométrage)

```
SELECT *
FROM vehicule
ORDER BY `kilometrage` DESC*
```

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

```
SELECT C.*
FROM `client` C
INNER JOIN a_loue A ON C.id_Client = A.id_Client
```

```
INNER JOIN vehicule V ON A.id_Vehicule = V.id_Vehicule
WHERE V.immatriculation = "EW25EW"
```

h. Afficher toutes les voitures noires :)

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

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

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

Travail de Verger—Doucy Brice

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

```
SELECT *
FROM a_loue
WHERE `date_de_retrait` < '2018-01-01';
```

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';
```

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 V.id_vehicule, A.nom, AL.date_de_retrait, AL.date_de_retour
FROM vehicule V
INNER JOIN a_loue AL ON V.id_Vehicule = AL.id_Vehicule
INNER JOIN agence A ON A.id_Agence = V.id_Agence
WHERE AL.date_de_retrait = AL.date_de_retour
```

- Obtenir le nombre véhicules pour chaque marque

Travail de Verger—Doucy Brice

VERGER—DOUCY BRICE 1BTSSIO BT

```
SELECT M.nom, COUNT(*)
FROM vehicule V
INNER JOIN marque M ON V.id_Marque = M.id_Marque
GROUP BY M.nom
```

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

»

```
SELECT C.nom, COUNT(*)
FROM vehicule V
INNER JOIN marque M ON V.id_Marque = M.id_Marque
INNER JOIN a_loue AL ON V.id_Vehicule = AL.id_Vehicule
INNER JOIN client C ON AL.id_Client = C.id_Client
WHERE M.nom = "Renault"
GROUP BY M.nom
HAVING COUNT(*) >10
```

Travail de Verger—Doucy Brice

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

```
SELECT P.nom, nb_Employe, COUNT(a.id_Agence) AS "nombre
d'agence"
FROM agence A
INNER JOIN pays P ON A.id_pays = P.id_pays
GROUP BY P.nom
```

## Exercice 2

Ci-après, on donne la représentation textuelle simplifiée d'une base de données concernant un cycle de formation destiné à des étudiants. Il regroupe un ensemble de matières. On considère que chaque enseignant n'enseigne qu'une seule matière et qu'à la fin du cycle de formation, une note par matière, est attribuée à chaque étudiant. D'autre par, les étudiants peuvent ne pas suivre les mêmes matières.

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

□

Ecrire les requêtes SQL permettant d'afficher :

## Travail de Verger—Doucy Brice

1. Les informations relatives aux étudiants (Code, Nom et Date de naissance) selon l'ordre alphabétique croissant du nom  
**SELECT** CodeEt, NomEt, DatnEt  
**FROM** ETUDIANT  
**ORDER BY** NomEt **ASC**
2. Les noms et les grades des enseignants de la matière dont le nom est 'BD'.

```
SELECT NomEns, GradeEns
FROM ENSEIGNANT
WHERE CodeMat = (
SELECT CodeMat
FROM MATIERE
WHERE NomMat = 'BD')
```

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

```
SELECT DISTINCT NomMat, CoefMat
FROM MATIERE
WHERE CodeMat
IN
(SELECT CodeMat
FROM ENSEIGNANT
WHERE GradeEns = 'Grd3')
```

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

```
SELECT NomMat, CoefMat
FROM MATIERE
WHERE CodeMat
IN
```

```
(SELECT CodeMat
FROM NOTE
WHERE CodeEt = 'Et321')
```

Travail de Verger—Doucy Brice

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

```
SELECT COUNT(*)
FROM ENSEIGNANT
WHERE CodeMat = (
SELECT CodeMat
FROM MATIERE
WHERE NomMat = 'Informatique')
```



**Exercice 3 :**

Soit le modèle relationnel suivant relatif à la gestion simplifiée des étapes du Tour de France 97, dont une des étapes de type "contre la montre individuel" se déroula à Saint-Etienne :

- 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)

Exprimez en SQL les requêtes suivantes :

Travail de Verger—Doucy Brice

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

```
SELECT C.NumeroCoureur, C.NomCoureur, P.NomPays
FROM COUREUR C
INNER JOIN EQUIPE E ON C.CodeEquipe = E.CodeEquipe
INNER JOIN PAYS P ON C.CodePays = P.CodePays
WHERE E.NomEquipe = 'Festina'
```

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

```
SELECT SUM(NbKm) AS "nombre kilomètres total"
FROM etape
```

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

```
SELECT SUM(E.NbKm) AS "nombre de kilomètres total des étapes
pour le type 'Haute Montagne'"
FROM etape E
INNER JOIN type_etape T ON E.CodeType = T.CodeType
WHERE T.LibelleType = 'Haute Montagne'
```

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

```
SELECT C.NomCoureur
FROM coureur C
```

```
WHERE C.NumeroCoureur NOT
IN
```

```
(SELECT A.NumeroCoureur
FROM ATTRIBUER_BONIFICATION A)
```

Travail de Verger—Doucy Brice

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

```
SELECT C.NomCoureur
FROM coureur C
WHERE NOT
EXISTS
```

```
(SELECT E.NumeroEtap
FROM etape E
WHERE E.NumeroEtap NOT
IN
(SELECT P.NumeroEtap
FROM participer P
WHERE P.NumeroCoureur = C.NumeroCoureur))
```

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 C.NomCoureur, C.CodeEquipe, C.CodePays, SUM(P.Temps
Realise + AB.NbSecondes) AS totaltemps
FROM coureur C
INNER JOIN participer P ON C.NumeroCoureur = P.NumeroCoureur
INNER
JOIN ATTRIBUER_BONIFICATION AB ON P.NumeroCoureur = AB.Nu
meroCoureur
AND P.NumeroEtap = AB.NumeroEtap
INNER JOIN etape E ON E.NumeroEtap = P.NumeroEtap
WHERE E.NumeroEtap <=13
GROUP BY C.NumeroCoureur
ORDER BY totaltemps
```

Travail de Verger—Doucy Brice

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

Travail de Verger—Doucy Brice

```
SELECT NomEquipe, SUM(TempsRealise + NbSecondes) AS tempstotal
FROM equipe E
```

VERGER—DOUCY BRICE 1BTSSIO BT

**INNER JOIN** coureur C **ON** C.CodeEquipe = E.CodeEquipe

**INNER JOIN** participer P **ON** P.NumeroCoureur = C.NumeroCoureur

**LEFT JOIN** ATTRIBUER\_BONIFICATION AB **ON** AB.NumeroCoureur = C.  
NumeroCoureur

**GROUP BY** NomEquipe

**ORDER BY** tempstotal

**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) On suppose que Numcli, Numprod, Idvendeur et Numcom sont de type numérique. Le nom, le prénom et l'adresse des clients ainsi que les vendeurs sont des informations obligatoires, le mail peut ne pas être indiqué.

La valeur par défaut de la quantité en stock des produits (qte\_stock) est égale à 0 Exprimer en SQL les requêtes suivantes :

Créer les tables : Client, Produit, Vendeur et Commande.

```
CREATE TABLE Client (
 Numcli INT PRIMARY KEY,
 Nomcli VARCHAR(50) NOT NULL,
 Prenomcli VARCHAR(50) NOT NULL,
 adressecli VARCHAR(50) NOT NULL,
 mailcli VARCHAR(50)
```

```
); Travail de Verger—Doucy Brice
```

```
CREATE TABLE Produit (
 Numprod INT PRIMARY KEY,
 designation VARCHAR(50) NOT NULL,
 prix INT NOT NULL,
 qte_stock INT DEFAULT 0
);
```

```
CREATE TABLE Vendeur (
 Idvendeur INT PRIMARY KEY,
 Nomvendeur VARCHAR(50) NOT NULL,
 adresse_vend VARCHAR(50) NOT NULL
);
```

```
CREATE TABLE Commande (
 Numcom INT PRIMARY KEY,
 Numcli INT NOT NULL,
 Idvendeur INT NOT NULL,
 Numprod INT NOT NULL,
```

VERGER—DOUCY BRICE 1BTSSIO BT  
date\_com DATE NOT NULL,  
qte\_com INT NOT NULL,  
FOREIGN KEY (Numcli) REFERENCES Client(Numcli),  
FOREIGN KEY (Idvendeur) REFERENCES Vendeur(Idvendeur),  
FOREIGN KEY (Numprod) REFERENCES Produit(Numprod)  
); Travail de Verger—Doucy Brice

1. la liste des clients de Marrakech.

```
SELECT Nomcli, Prenomcli, adressecli
FROM Client
WHERE adressecli = 'marrakech'
```

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

```
SELECT Numprod, designation, prix
FROM Produit
ORDER BY prix DESC
```

Travail de Verger—Doucy Brice

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

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

Travail de Verger—Doucy Brice

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

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

```
SELECT COUNT(*) AS "nombre de commande conteant le produit n°365"
FROM Commande
WHERE Numprod =365
```

## Exercice 5 :

Soit la base de données suivante :

### *Etudiant*

| numéro_carte_etudiant | Nom         | Prénom | Date_naissance | Section      |
|-----------------------|-------------|--------|----------------|--------------|
| 01234567              | Ben Salah   | Ahmed  | 12/08/1988     | Informatique |
| 01234568              | Ben Mahmoud | Sami   | 02/09/1990     | Math         |
| 01234569              | Marzougui   | Rami   | 23/01/1988     | Informatique |

### *Matière*

| code_matière | nom_matière     | coefficient |
|--------------|-----------------|-------------|
| 12508        | Base de données | 1.5         |
| 12518        | Algorithme      | 3           |

### *Note*

| numéro_carte_etudiant | code_matière | note_examen |
|-----------------------|--------------|-------------|
| 01234567              | 12508        | 15.5        |
| 01234567              | 12518        | 5.5         |
| 01234568              | 12518        | 10.5        |
| 01234569              | 12518        | 8.75        |

Ecrire les commandes SQL permettant de rechercher :

1. La liste de tous les étudiants.

```
SELECT nom, prenom
FROM etudiant
Travail de Verger—Doucy Brice
```

2. Nom et coefficient des matières.

```
SELECT nom_matiere, coefficient
FROM matiere
Travail de Verger—Doucy Brice
```

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

```
SELECT numero_carte_etudiant
FROM etudiant
WHERE numero_carte_etudiant
IN
```

```
(SELECT numero_carte_etudiant
FROM note
GROUP BY numero_carte_etudiant
HAVING AVG(note_examen)
BETWEEN 7
```

AND 12)

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

```
SELECT nom, prenom, numero_carte_etudiant
FROM etudiant
WHERE nom LIKE 'ben%'
```

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

```
SELECT COUNT(*) AS "nombre des étudiants qui ont comme
matières 12518"
FROM
(SELECT DISTINCT numero_carte_etudiant
FROM NOTE
WHERE code_matiere = '12518') AS students
```

6. La somme des coefficients des matières.

```
SELECT SUM(coefficient) AS "coefficient matière total"
FROM matiere
```

Travail de Verger—Doucy Brice

7. Les noms des étudiants qui une note\_examen >10.

```
SELECT E.nom, E.prenom
FROM etudiant E
INNER
JOIN note N ON E.numero_carte_etudiant = N.numero_carte_etudiant
WHERE note_examen >10
```

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

```
SELECT M.nom_matiere, M.coefficient
FROM matiere M
INNER JOIN note N ON N.code_matiere = M.code_matiere
INNER JOIN etudiant E ON N.numero_carte_etudiant = E.numero_carte_et
udiant
WHERE E.numero_carte_etudiant = '01234568'
```