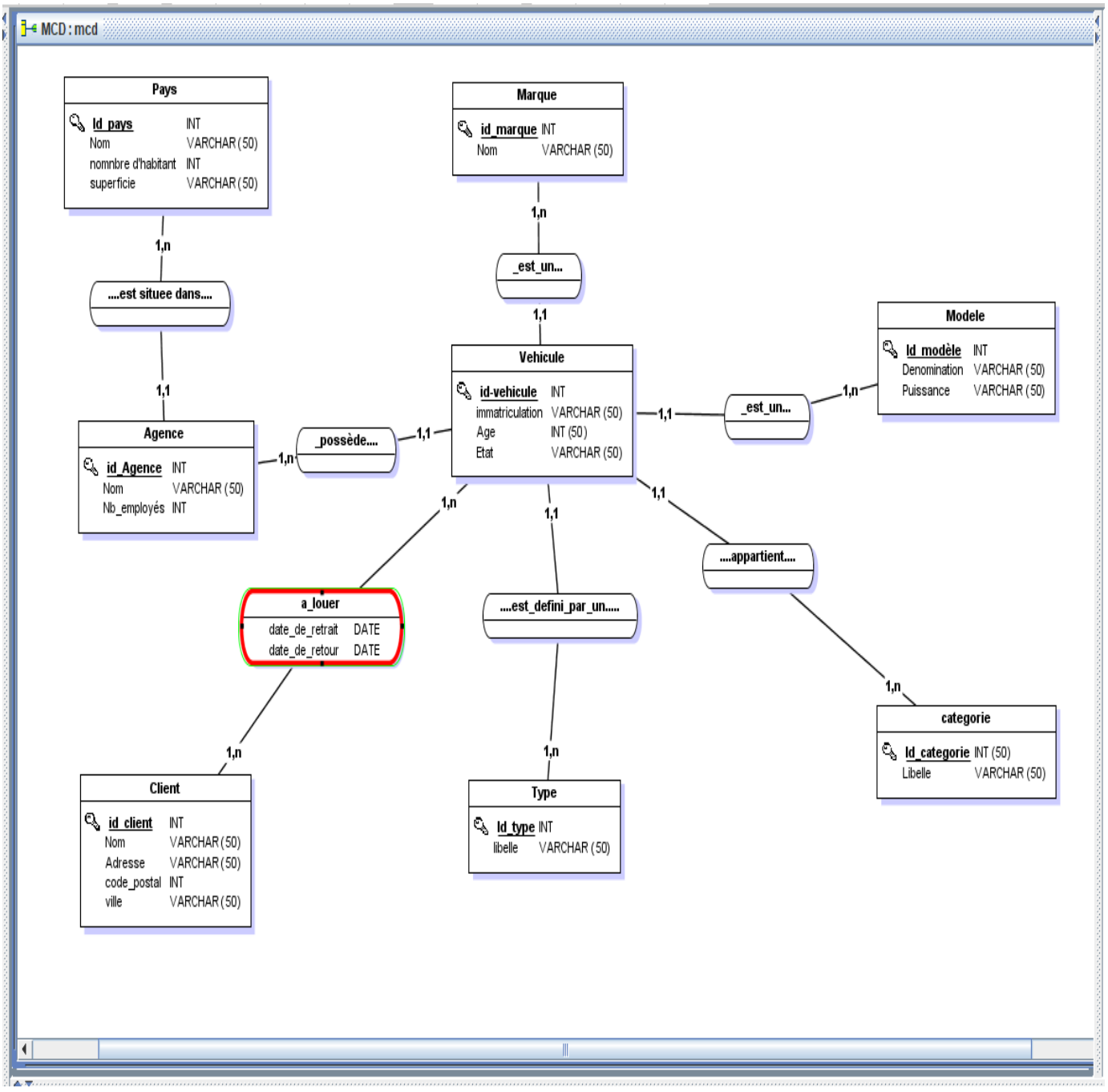
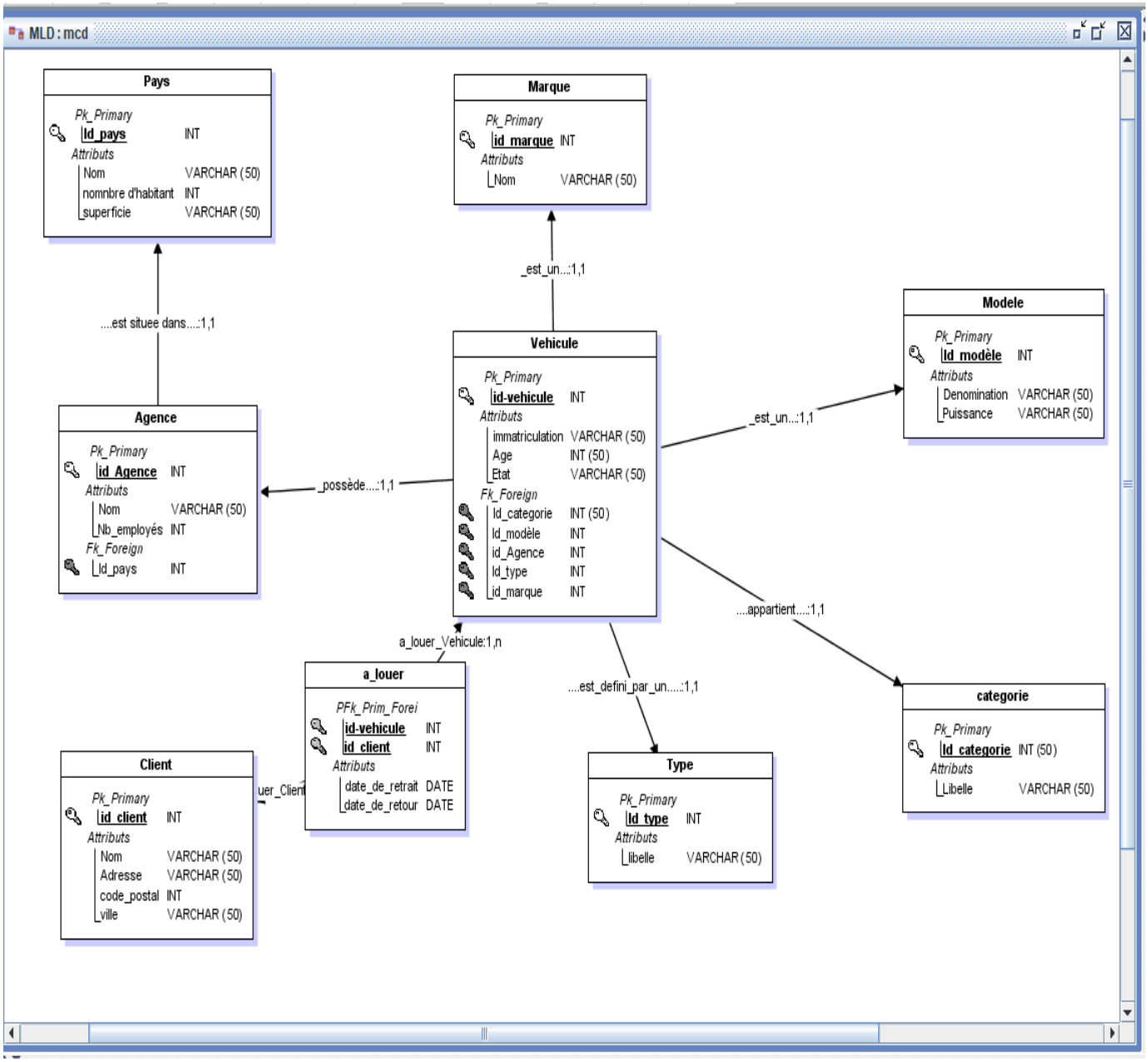
EXERCICE 1 :

MCD



MLD



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

Pays (id_pays, nom, nombre d'habitant, superficie)

Agence (id Agence, Nom, Nb_employés, #id_pays)

Client (id client, Nom, Adresse, code_postal, ville)

a_loue (id client, id-vehicule, date_de_retrait, date_de_retour)

Véhicule (id-vehicule, immatriculation, Age, Etat, #id_Agence, #id_marque, #id_type, #id_categorie, #id_modèle)

Marque (id marque, Nom)

Type (id type, libelle)

Catégorie (id categorie, libelle)

Modèle (id modele, denomination, puissance)

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_louer 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

```sql
SELECT vehicule.id_vehicule
FROM vehicule
JOIN a_louer ON vehicule.id_vehicule =
vehicule.id_vehicule
WHERE id_client = 'T122'
LIMIT 0 , 30
```

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

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

D/ Afficher les noms des clients et les adresses, des clients qui habitent à<< Nice >>.

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

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

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

F/ Ajouter l'attribut kilométrage et Afficher la liste des voitures par ordre décroissant des compteurs (kilométrage)

```
ALTER TABLE vehicule

ADD COLUMN kilometrage INTEGER;


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 *
FROM client
JOIN a_louer ON client.id_client = a_louer.id_client
JOIN vehicule ON a_louer.id_vehicule = vehicule.id_vehicule
WHERE vehicule.immatriculation = 'EW 25';

LIMIT 0 , 30
```

H/ Afficher toutes les voitures noires :)

```
ALTER TABLE vehicule
```

```
ADD COLUMN COUELEUR INTEGER;

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

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

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

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

```
SELECT *
FROM `a_louer`
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 vehicul
```

L/ Afficher toutes les locations réalisées en 2018

```
SELECT *
FROM a_louer
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 v.id_vehicule, a.nom, al.date_de_retrait,
  al.date_de_retour
  FROM vehicule v
  JOIN a_louer al ON v.id_vehicule = al.id_vehicule
  JOIN agence a ON a.id_agence = v.id_agence
  WHERE al.date_de_retrait = al.date_de_retour
  LIMIT 0 , 30
  ```
  ```

- **Obtenir le nombre véhicules pour chaque marque**

```
SELECT marque.Nom, COUNT( vehicule.id_vehicule ) AS
nb_vehicules
FROM marque
LEFT JOIN vehicule ON vehicule.id_marque =
marque.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 c.nom, COUNT( * )
FROM vehicule v
INNER JOIN marque m ON v.id_marque = m.id_marque
INNER JOIN a_louer 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
LIMIT 0 , 30
```

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

```
SELECT agence.pays, COUNT( DISTINCT agence.id_agence )
AS nb_agences, COUNT( employe.id_employe ) AS
```

```
nb_employes
FROM agence
LEFT JOIN employe ON agence.id_agence =
employe.id_agence
GROUP BY agence.pays;

LIMIT 0 , 30
```

Exercice 2:

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

```
SELECT CodeEt, NomEt, DatnEt
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 NomEns, GradeEns
FROM ENSEIGNANT
JOIN MATIERE ON ENSEIGNANT.CodeMat =
MATIERE.CodeMat
WHERE 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 NomMat, CoefMat
FROM MATIERE
JOIN ENSEIGNANT ON MATIERE.CodeMat =
ENSEIGNANT.CodeMat
WHERE GradeEns = 'Grd3'
LIMIT 0 , 30
```

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

```
SELECT NomMat, CoefMat
FROM MATIERE
JOIN NOTE ON MATIERE.CodeMat = NOTE.CodeMat
WHERE CodeEt = 'Et321'
LIMIT 0 , 30
```

5/ Le nombre d’enseignants de la matière dont le nom est ‘Informatique’

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

EXOERCIE 3:

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

```
SELECT NumeroCoureur, NomCoureur, PAYS.NomPays
FROM COUREUR
JOIN EQUIPE ON COUREUR.CodeEquipe =
EQUIPE.CodeEquipe
JOIN PAYS ON COUREUR.CodePays = PAYS.CodePays
WHERE NomEquipe = 'Festina'
LIMIT 0 , 30
```

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

```
SELECT SUM( NbKm ) AS TotalKm
FROM ETAPE
```

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

```
SELECT SUM( NbKm ) AS TotalKm
FROM ETAPE
JOIN TYPE_ETAPE ON ETAPE.CodeType =
TYPE_ETAPE.CodeType
WHERE 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

)
LIMIT 0 , 30
```

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

```
SELECT NomCoureur
FROM COUREUR
WHERE NOT
EXISTS (

SELECT NumeroEtap
FROM ETAPE
WHERE NOT
EXISTS (

SELECT *
FROM PARTICIPER
WHERE PARTICIPER.NumeroEtap = ETAPE.NumeroEtap
```

```
AND PARTICIPER.NumeroCoureur =
COUREUR.NumeroCoureur

)

)
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 COUREUR.NomCoureur, EQUIPE.CodeEquipe,
PAYS.NomPays, SUM( PARTICIPER.TempsRealise ) AS
TempsTotal
FROM COUREUR
JOIN EQUIPE ON COUREUR.CodeEquipe =
EQUIPE.CodeEquipe
JOIN PAYS ON COUREUR.CodePays = PAYS.CodePays
JOIN PARTICIPER ON COUREUR.NumeroCoureur =
PARTICIPER.NumeroCoureur
JOIN (

SELECT NumeroCoureur, MIN( NumeroEtap ) AS PremierEtap
FROM PARTICIPER
GROUP BY NumeroCoureur

) AS Premier ON COUREUR.NumeroCoureur =
Premier.NumeroCoureur
JOIN (
```

```
SELECT NumeroCoureur, MAX( NumeroEtap ) AS DernierEtap
FROM PARTICIPER
GROUP BY NumeroCoureur

) AS Dernier ON COUREUR.NumeroCoureur =
Dernier.NumeroCoureur
JOIN ETAPE ON ETAPE.NumeroEtap >= Premier.PremierEtap
AND ETAPE.NumeroEtap <= Dernier.DernierEtap
AND PARTICIPER.NumeroEtap = ETAPE.NumeroEtap
GROUP BY COUREUR.NumeroCoureur, EQUIPE.CodeEquipe,
PAYS.NomPays
ORDER BY TempsTotal
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 + NbSecondes ) AS
TempsTotal
FROM equipe e
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
LIMIT 0 , 30
```

EXERCICE 4

CREATION DES TABLES

```
CREATE TABLE Client(

Numcli NUMERIC PRIMARY KEY ,
Nomcli VARCHAR( 50 ) NOT NULL ,
Prenomcli VARCHAR( 50 ) NOT NULL ,
adressecli VARCHAR( 100 ) NOT NULL ,
mailcli VARCHAR( 100 )

);# MySQL n'a retourné aucune ligne.
CREATE TABLE Produit(

Numprod NUMERIC PRIMARY KEY ,
designation VARCHAR( 50 ) NOT NULL ,
prix NUMERIC NOT NULL ,
qte_stock NUMERIC DEFAULT 0

);# MySQL n'a retourné aucune ligne.
CREATE TABLE Vendeur(

Idvendeur NUMERIC PRIMARY KEY ,
Nomvendeur VARCHAR( 50 ) NOT NULL ,
adresse_vend VARCHAR( 100 ) NOT NULL

);# MySQL n'a retourné aucune ligne.
CREATE TABLE Commande(

Numcom NUMERIC PRIMARY KEY ,
Numcli NUMERIC REFERENCES Client( Numcli ) ,
Idvendeur NUMERIC REFERENCES Vendeur( Idvendeur ) ,
Numprod NUMERIC REFERENCES Produit( Numprod ) ,
```

```
date_com DATE,
qte_com NUMERIC NOT NULL
);
```

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 DESC ;

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 *
FROM Commande
WHERE Idvendeur = (
SELECT Idvendeur
FROM Vendeur
WHERE Nomvendeur = 'Mohammed' )
AND 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
```

EXERCIE 5:

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 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

)
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 COUNT( * ) AS "nombre des étudiants qui ont comme matière 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 )
FROM matiere
```

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

```
SELECT nom, prenom
FROM etudiant e
INNER JOIN note n ON e.numero_carte_etudiant = n.numero_carte_etudiant
WHERE note_examen >10
LIMIT 0 , 30
```

8/ Afficher les noms et les coefficients des matières étudier par l’étudiant "01234568".

```
SELECT nom_matiere, coefficient
FROM matiere m
INNER JOIN note n ON n.code_matiere = n.code_matiere
INNER JOIN etudiant e ON n.numero_carte_etudiant =
e.numero_carte_etudiant
WHERE e.numero_carte_etudiant = '01234568'
LIMIT 0 , 30
```