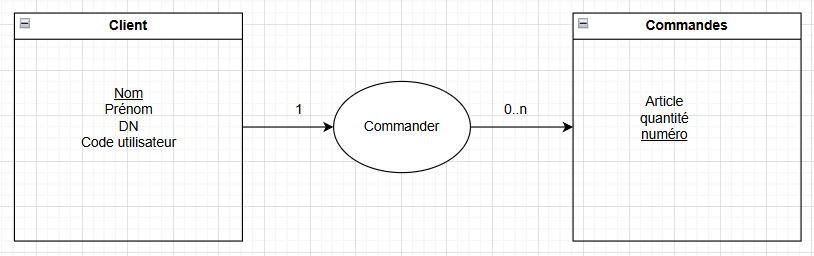
**Exercices BDD**

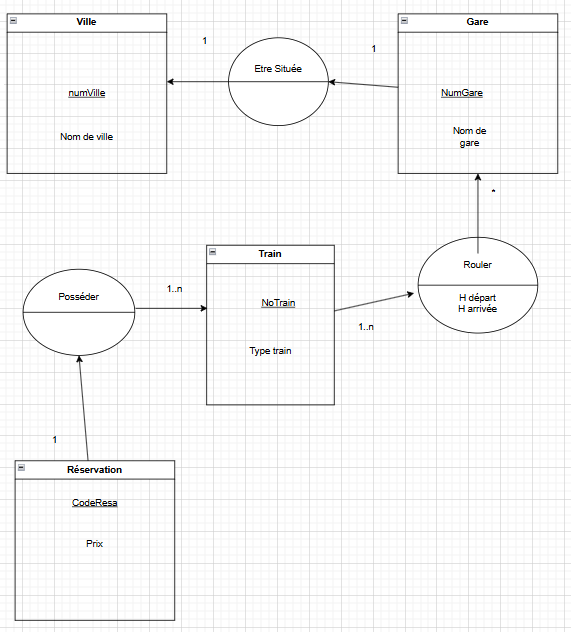
**Exercice 1**



**Exercice 2**

Une réservation possède un ou plusieurs N° de train.

Ces trains roulent vers une ou plusieurs Gares qui sont situées dans une ville.



**Exercice 3**

CLIENT (id Clt, Nom, Prénom, Date\_Naiss, Adresse)

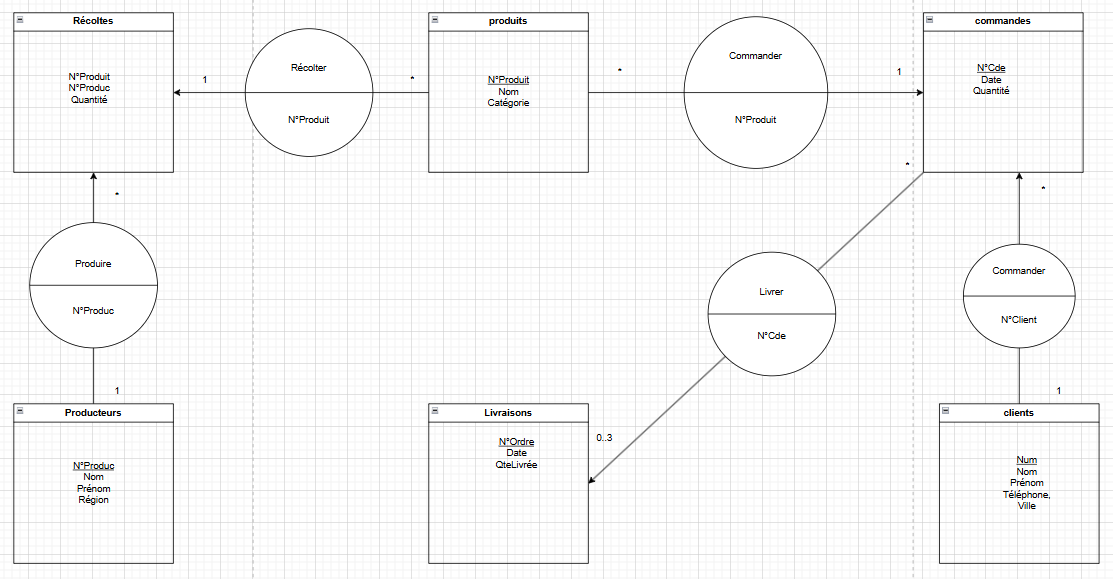
DISQUE (Num\_ref\_Dsg, #Id\_Clt, Titre, Date\_Gravure)

RANGEE(Id\_Rangee, #Id\_Disque, #Id\_Rayon)

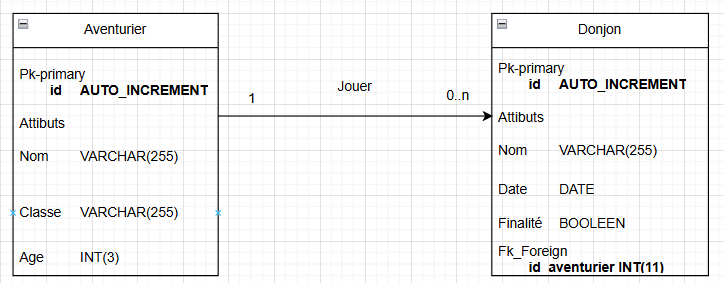
RAYON (Id\_Rayon, #Num\_ref\_Dsg, Libellé, Num\_Rayon)

ARTISTE (Id\_Artiste, #Num\_ref\_Dsg, Nom, Prénom, Pays, Domaine)

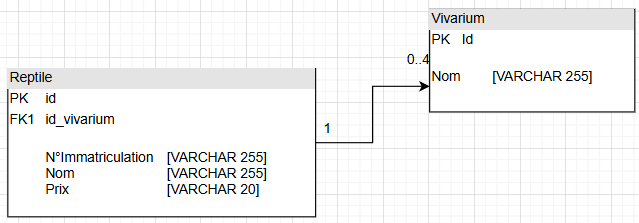
**Exercice 4**



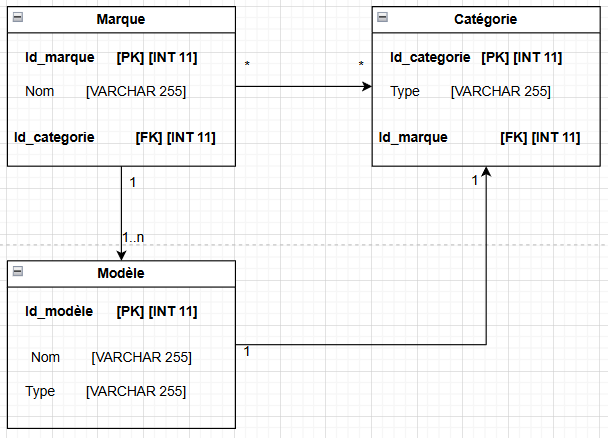
**Exercice 5**



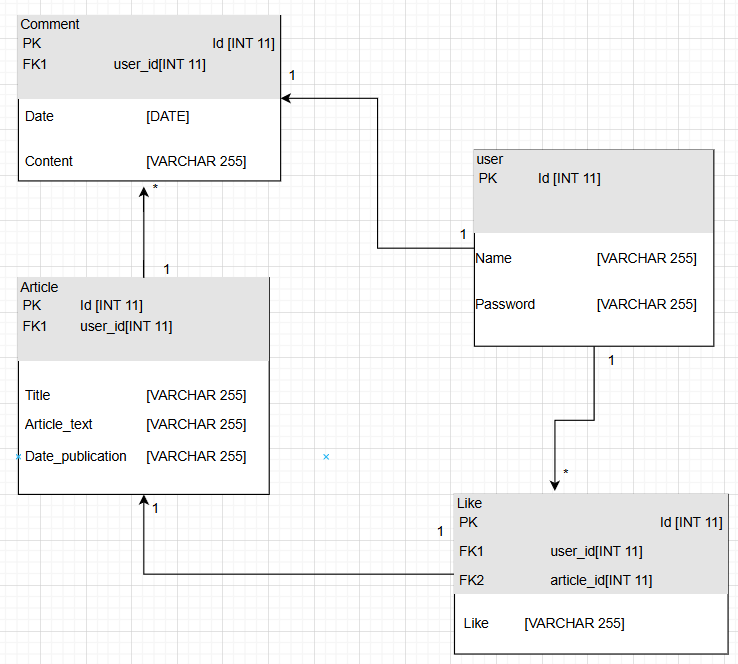
**Exercice 6**



**Exercice 7**

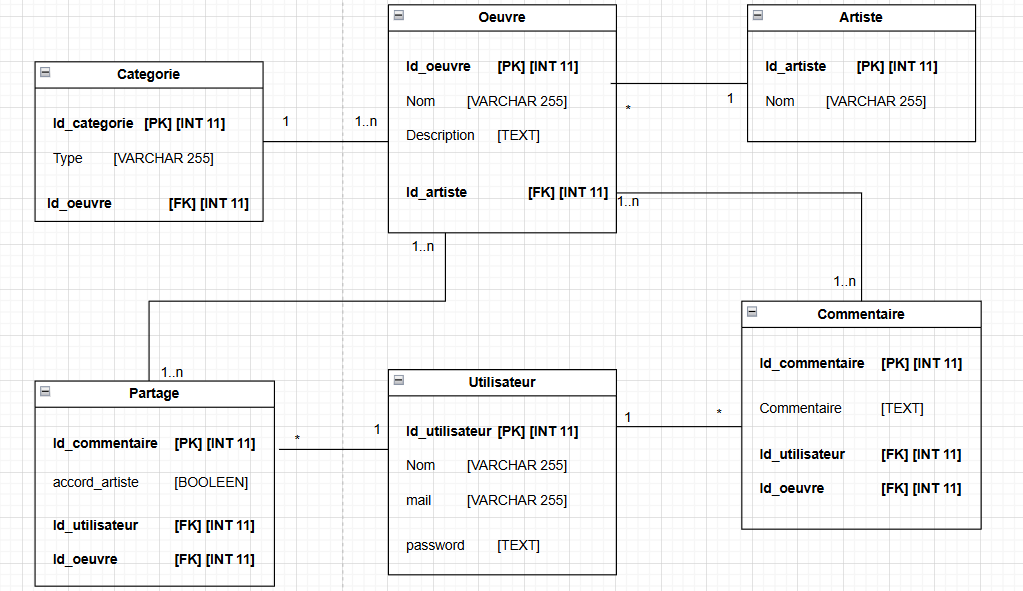


**Exercice 8**



Exercice 9

Plateforme où des artistes exposeront leurs oeuvres en ligne. Les oeuvres ont une catégorie. Les oeuvres peuvent être récupérer par les utilisateurs si l’artiste l’a permis au préalable. Les utilisateurs peuvent laisser un commentaire pour chaque oeuvre.



**Exercices SQL**

**Excercice1**

\*\*\*\*\*\*

CREATE DATABASE film

\*\*\*\*\*\*\*

CREATE TABLE fiche\_film

(

id INT PRIMARY KEY,

titre VARCHAR(255),

durée\_en\_minute INT(10),

date\_de\_sortie YEAR(4)

)

\*\*\*\*\*\*\*

INSERT INTO fiche\_film (titre, durée\_en\_minute, date\_de\_sortie)

VALUES

('The Gentlement', 113, 2020),

('Kuzco', 75,2000),

('Joker', 122, 2019),

('Sweeney todd', 116, 2007),

('Princesse mononoké', 134, 1997);

\*\*\*\*\*\*\*\*\*

DELETE FROM fiche\_film

WHERE date\_de\_sortie<2010

AND durée\_en\_minute<120;

RESULTAT 

**Exercice 2**

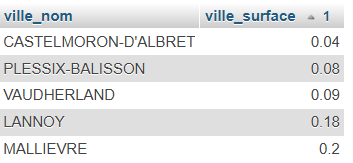
1°

SELECT ville\_nom, ville\_surface

FROM villes\_france\_free

ORDER BY ville\_surface ASC

LIMIT 5



2°

SELECT ville\_code\_postal, ville\_population\_2010

FROM villes\_france\_free

ORDER BY ville\_population\_2010 DESC

LIMIT 15

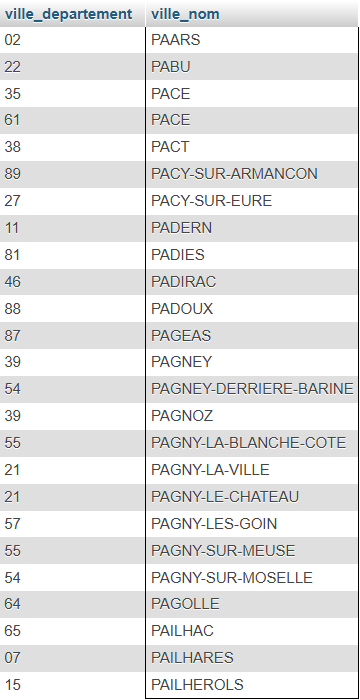


3°

SELECT ville\_departement, ville\_nom

FROM villes\_france\_free

WHERE ville\_nom LIKE 'P%'

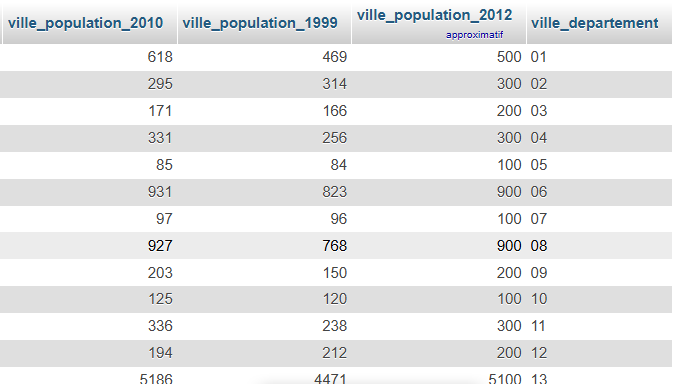


4°

SELECT ville\_population\_2010,ville\_population\_1999, ville\_population\_2012, ville\_departement

FROM villes\_france\_free

GROUP BY ville\_departement

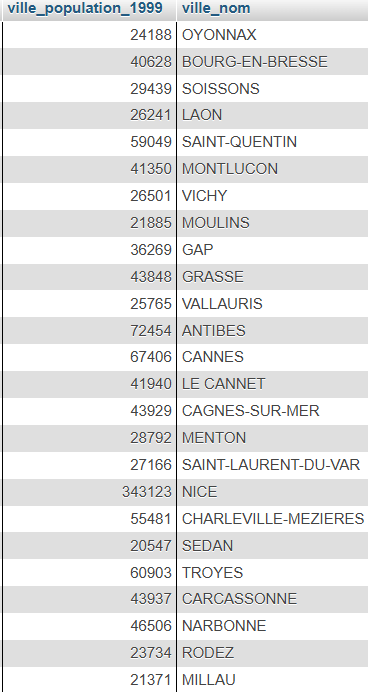


5°

SELECT ville\_population\_1999, ville\_nom

FROM villes\_france\_free

WHERE ville\_population\_1999>20000

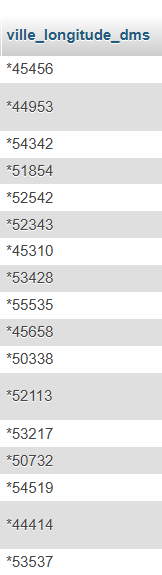


6°

UPDATE villes\_france\_free

SET ville\_longitude\_dms=REPLACE (ville\_longitude\_dms, '+','\*')

WHERE ville\_longitude\_dms LIKE '%+%’



7°

UPDATE villes\_ france\_free

SET ville\_nom = REVERSE(ville\_nom)

WHERE ville\_departement LIKE '97%'



8°

Jointure des deux tables

SELECT \*

FROM villes\_france\_free

INNER JOIN departement ON

villes\_france\_free.ville\_departement=departement.departement\_code

\*\*\*\*\*\*\*\*\*\*\*\*

SELECT ville\_nom, ville\_surface, departement.departement\_nom

FROM villes\_france\_free

INNER JOIN departement ON

villes\_france\_free.ville\_departement=departement.departement\_code

ORDER BY ville\_surface DESC

LIMIT 5

9°

SELECT departement.departement\_nom

FROM villes\_france\_free

INNER JOIN departement ON

villes\_france\_free.ville\_departement=departement.departement\_code

ORDER BY ville\_population\_2010 DESC

LIMIT 5

Exercice 3

1°

SELECT Modele, Marque, Categorie

FROM darty

2°

DELETE FROM darty

WHERE marque='indesit'

3°

UPDATE darty

SET Prix= Prix-50

WHERE Prix>600

Excercice 4

1°

SELECT \*

FROM personnage

WHERE metier != ' '

2°

SELECT \*

FROM personnage

3°

SELECT metier

FROM personnage

WHERE metier = ' '

Exercice 5

SELECT p.produit, ROUND(AVG(n.note),1)

FROM produits AS p

INNER JOIN note as n

ON p.id\_produits=n.id\_produits

GROUP BY p.id\_produits