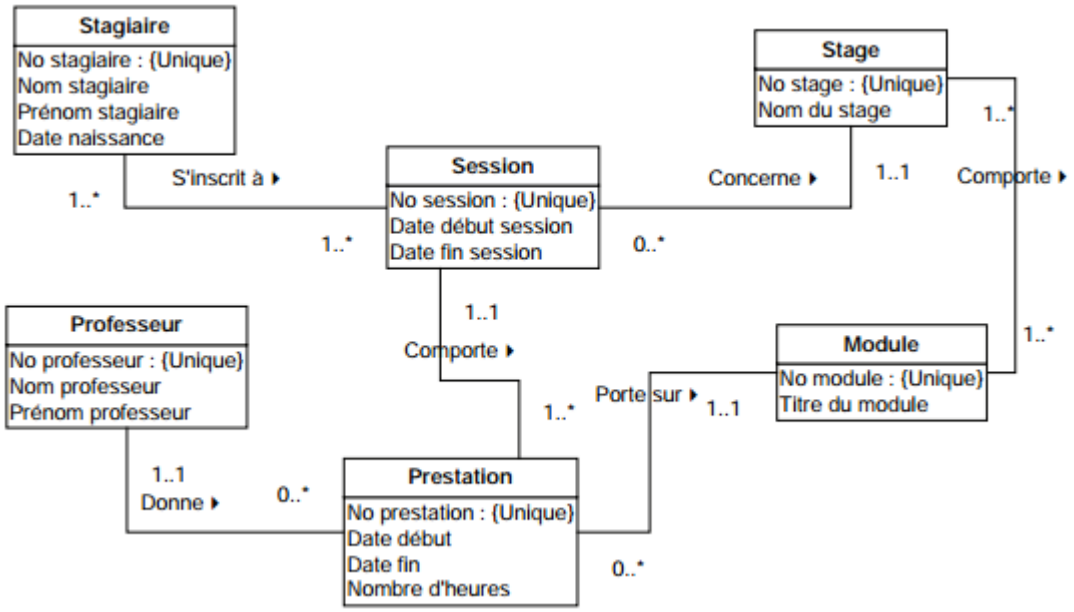
Matière : Base de données

Chargée de cours : Farah Barika Ktata **Chargée de TD :** Emna Ammar Elhadjamor	**TD1**

Exercice 1 :

Soit le modèle conceptuel des données suivant :



1. Traduire le modèle conceptuel en son équivalent relationnel en respectant les règles de dérivation.

Note : ***Module*** et ***Session*** sont des mots clés du langage SQL. Ils doivent par conséquent figurer entre crochets dans le script pour ne pas entraîner d'erreur de syntaxe.

2. Traduire le modèle relationnel obtenu en son équivalent physique en tenant compte de la contrainte suivante : l'attribut ***Nombre d'heures*** possède une valeur par défaut = 1.

Exercice 2 :

Une entreprise de location de vidéocassettes possède de nombreux magasins répartis dans tout le pays.

- Chaque magasin dispose d'une adresse composée du numéro de la rue, du nom de la rue, de la ville, du code postal et d'un numéro de téléphone.Un numéro exclusif est attribué au magasin pour l'ensemble de l'entreprise.
- Chaque magasin recrute son équipe, dont un directeur. Chaque employé possède un numéro unique son nom, son prénom, sa fonction et son salaire.
- Chaque magasin dispose d'un stock de cassettes vidéo. A chaque cassette vidéo est attribué un numéro unique puisqu'il existe plusieurs copies d'un même film, le numéro de catalogue du film, le titre du film, sa catégorie, le montant de la location quotidienne, le prix d'achat de la cassette, le statut (loué ou non), les noms des acteurs principaux et du réalisateur du film.

- Le film appartient à l'une des catégories suivantes : action, famille, drame, comédie, horreur, fantastique, science-fiction ou adulte. Le statut indique si la cassette est disponible pour la location.
- Avant qu'un client puisse louer une cassette auprès d'un magasin de location, il doit s'inscrire en tant que membre auprès d'un magasin de location. Le nom et le prénom, l'adresse et la date d'inscription du client sont enregistrés dans la base de données. Chaque membre reçoit un numéro unique.
- Les données sur chaque cassette louée sont un numéro de location unique, le nom du membre, le numéro de la cassette, le titre du film, les dates de location et de retour.

1. Etablir le modèle conceptuel des données avec le formalisme entité-association.
2. Traduire le modèle conceptuel obtenu en son équivalent relationnel en respectant les règles de dérivation appropriées.
3. Traduire le modèle relationnel obtenu en son équivalent physique en tenant compte des contraintes suivantes :

- Contraintes de domaine: l'attribut ***Salaire employé*** est strictement supérieur à 0; les valeurs acceptables pour l'attribut ***Catégorie*** sont 'Horreur', 'Tragédie', 'Comédie', 'Policier'.
- Contraintes d'intégrité: ***Date location <= Date restitution***.

Matière : Base de données

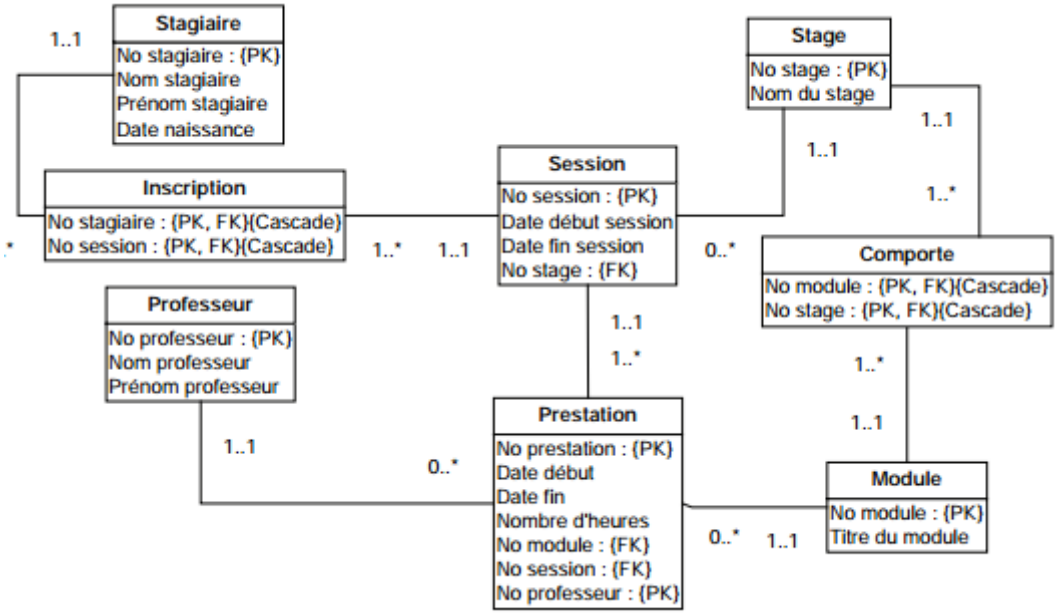
Chargée de cours : Farah Barika Ktata
Chargée de TD : Emna Ammar Elhadjamor

Correction TD1

Exercice 1 :

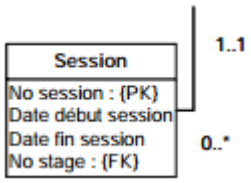
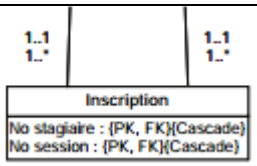
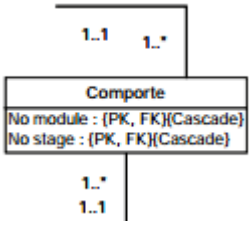
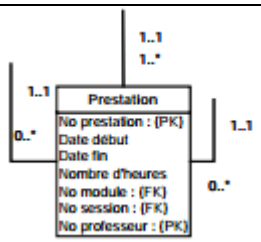
1. Traduire le modèle conceptuel en son équivalent relationnel en respectant les règles de dérivation.

Note : ***Module*** et ***Session*** sont des mots clés du langage SQL. Ils doivent par conséquent figurer entre crochets dans le script pour ne pas entraîner d'erreur de syntaxe.



2. Traduire le modèle relationnel obtenu en son équivalent physique en tenant compte de la contrainte suivante : l'attribut ***Nombre d'heures*** possède une valeur par défaut = 1.

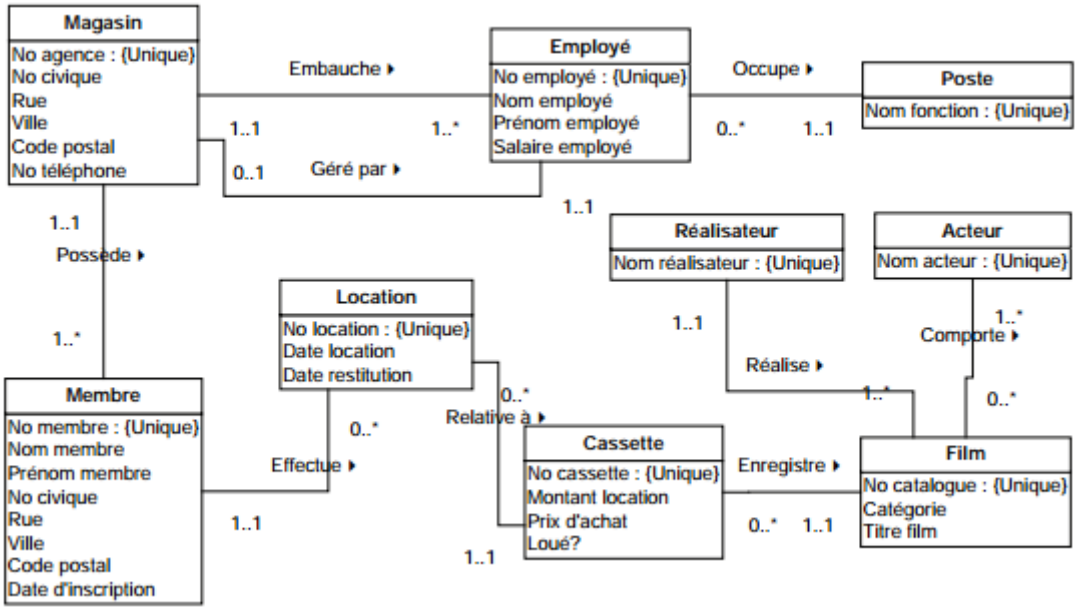
Table	Instruction SQL
Stagiaire No stagiaire : {PK} Nom stagiaire Prénom stagiaire Date naissance	`CREATE TABLE Stagiaire ([No stagiaire] SMALLINT NOT NULL PRIMARY KEY, [Nom stagiaire] VARCHAR(30), [Prénom stagiaire] VARCHAR(30), [Date naissance] DATE)`
Stage No stage : {PK} Nom du stage	`CREATE TABLE Stage ([No stage] SMALLINT NOT NULL PRIMARY KEY, [Nom du stage] VARCHAR(30))`
Professeur No professeur : {PK} Nom professeur Prénom professeur	`CREATE TABLE Professeur ([No professeur] SMALLINT NOT NULL PRIMARY KEY, [Nom professeur] VARCHAR(30), [Prénom professeur] VARCHAR(30))`
Module No module : {PK} Titre du module	`CREATE TABLE [Module] ([No module] SMALLINT NOT NULL PRIMARY KEY, [Titre du module] VARCHAR(30))`

	CREATE TABLE [Session] ([No session] SMALLINT NOT NULL PRIMARY KEY, [Date début session] DATE, [Date fin session] DATE, [No stage] SMALLINT NOT NULL REFERENCES Stage([No stage]))
	CREATE TABLE [Inscription] ([No stagiaire] SMALLINT NOT NULL REFERENCES Stagiaire([No stagiaire]) ON DELETE CASCADE ON UPDATE CASCADE, [No session] SMALLINT NOT NULL REFERENCES [Session]([No session]) ON DELETE CASCADE ON UPDATE CASCADE, PRIMARY KEY([No stagiaire], [No session]))
	CREATE TABLE [Comporte] ([No module] SMALLINT NOT NULL REFERENCES [Module]([No module]) ON DELETE CASCADE ON UPDATE CASCADE, [No stage] SMALLINT NOT NULL REFERENCES Stage([No stage]) ON DELETE CASCADE ON UPDATE CASCADE, PRIMARY KEY([No module], [No stage]))
	CREATE TABLE Prestation ([No prestation] SMALLINT NOT NULL PRIMARY KEY, [Date début] DATE, [Date fin] DATE, [Nombre d'heures] SMALLINT DEFAULT 1, [No module] SMALLINT NOT NULL REFERENCES [Module]([No module]), [No session] SMALLINT NOT NULL REFERENCES [Session]([No session]), [No professeur] SMALLINT NOT NULL REFERENCES Professeur([No professeur]),

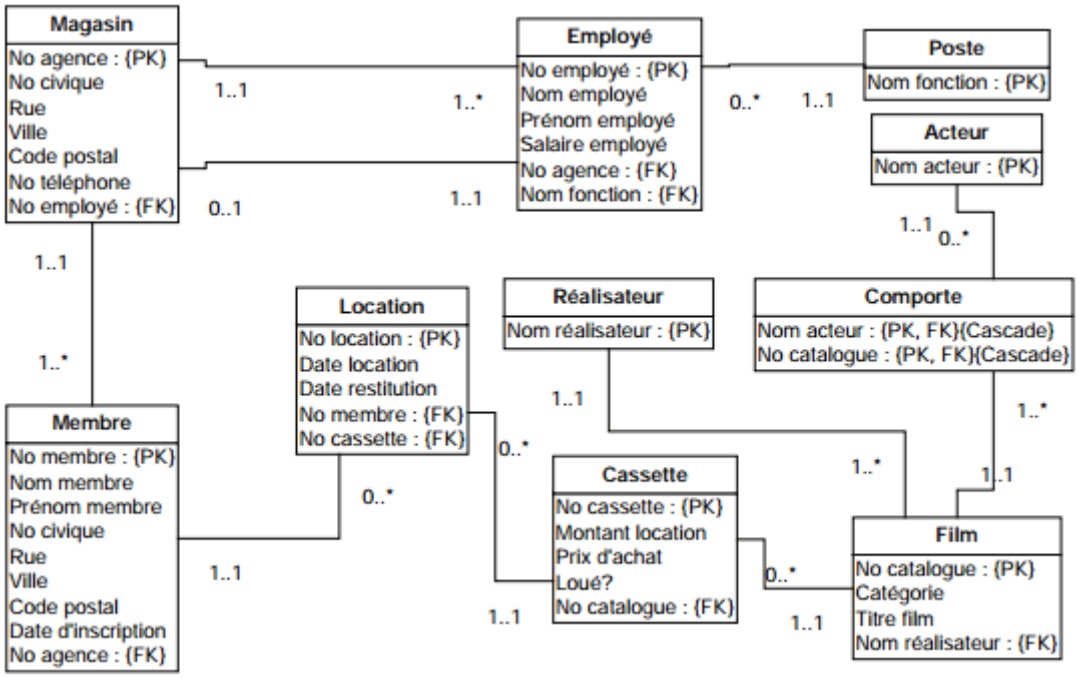
Nombre de tables: 8

Exercice 2 :

1. Etablir le modèle conceptuel des données avec le formalisme entité-association.



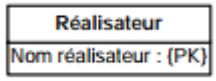
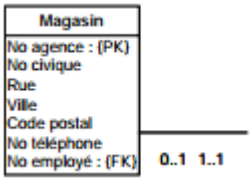
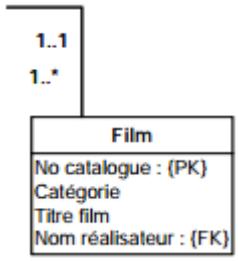
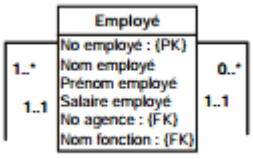
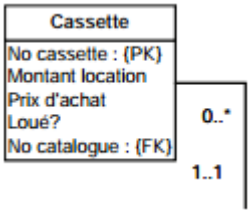
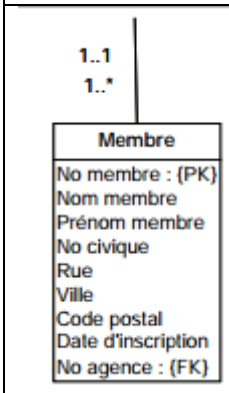
2. Traduire le modèle conceptuel obtenu en son équivalent relationnel en respectant les règles de dérivation appropriées.

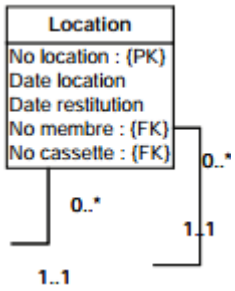
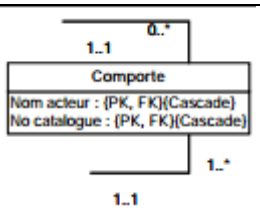


3. Traduire le modèle relationnel obtenu en son équivalent physique en tenant compte des contraintes suivantes :

- Contraintes de domaine: l'attribut ***Salaire employé*** est strictement supérieur à 0; les valeurs acceptables pour l'attribut ***Catégorie*** sont 'Horreur', 'Tragédie', 'Comédie', 'Policier'.
- Contraintes d'intégrité: ***Date location <= Date restitution***.

Table	Instruction SQL
Poste Nom fonction : {PK}	CREATE TABLE Poste ([Nom fonction] VARCHAR(30) NOT NULL PRIMARY KEY)
Acteur Nom acteur : {PK}	CREATE TABLE Acteur ([Nom acteur] VARCHAR(30)

	NOT NULL PRIMARY KEY)
	CREATE TABLE Réalisateur ([Nom réalisateur] VARCHAR(30) NOT NULL PRIMARY KEY)
	CREATE TABLE Magasin ([No agence] SMALLINT NOT NULL PRIMARY KEY, [No civique] VARCHAR(6), [Rue] VARCHAR(30), [Ville] VARCHAR(30), [Code postal] VARCHAR(6))
	CREATE TABLE Film ([No catalogue] SMALLINT NOT NULL PRIMARY KEY, [Catégorie] VARCHAR(10), [Titre film] VARCHAR(30), [Nom réalisateur] VARCHAR(30) NOT NULL REFERENCES Réalisateur([Nom réalisateur]), CONSTRAINT CatValide CHECK(Catégorie IN ('Horreur', 'Tragédie','Comédie', 'Policier')))
	CREATE TABLE Employé ([No employé] SMALLINT NOT NULL PRIMARY KEY, [Nom employé] VARCHAR(30), [Prénom employé] VARCHAR(30), [Salaire employé] CURRENCY, [No agence] SMALLINT NOT NULL REFERENCES Magasin([No agence]), [Nom fonction] VARCHAR(30) NOT NULL REFERENCES Poste([Nom fonction]), CONSTRAINT SalSupZéro CHECK([Salaire employé]>0)) ALTER TABLE Magasin ADD COLUMN [No employé] SMALLINT NOT NULL UNIQUE REFERENCES Employé([No employé])
	CREATE TABLE Cassette ([No cassette] SMALLINT NOT NULL PRIMARY KEY, [Montant location] CURRENCY, [Prix d'achat] CURRENCY, [Loué?] BIT, [No catalogue] SMALLINT NOT NULL REFERENCES Film([No catalogue]))
	CREATE TABLE Membre ([No membre] SMALLINT NOT NULL PRIMARY KEY, [Nom membre] VARCHAR(30), [Prénom membre] VARCHAR(30), [No civique] VARCHAR(6), [Rue] VARCHAR(30), [Ville] VARCHAR(30), [Code postal] VARCHAR(6), [Date d'inscription] DATE, [No agence] SMALLINT NOT NULL REFERENCES Magasin([No agence]))

	CREATE TABLE [Location] ([No location] SMALLINT NOT NULL PRIMARY KEY, [Date location] DATE, [Date restitution] DATE, [No membre] SMALLINT NOT NULL REFERENCES Membre([No membre]), [No cassette] SMALLINT NOT NULL REFERENCES Cassette([No cassette]), CONSTRAINT LocationInfRestitution CHECK([Date location]<= [Date restitution]))
	CREATE TABLE [Comporte] ([Nom acteur] VARCHAR(30) NOT NULL REFERENCES Acteur([Nom acteur]) ON DELETE CASCADE ON UPDATE CASCADE, [No catalogue] SMALLINT NOT NULL REFERENCES Film([No catalogue]) ON DELETE CASCADE ON UPDATE CASCADE, PRIMARY KEY([Nom acteur],[No catalogue]))

Matière : Base de données

Chargée de cours : Farah Barika Ktata
Chargée de TD : Emna Ammar Elhadjamor

TD2

Exercice 1 :

Soit les deux relations suivantes :

ma_cuisine(Appareil, Couleur)

Appareil	Couleur
Réfrigérateur	rouge
Robot	mauve
Cuisinière	jaune

sa_cuisine(Appareil, Couleur)

Appareil	Couleur
Réfrigérateur	mauve
Cuisinière	jaune
Hotte	bleue

Donnez les resultats des requetes suivantes :

- ma_cuisine ∩ sa_cuisine
- ma_cuisine – (ma_cuisine – sa_cuisine)

Exercice 2 :

Soit les deux relations suivantes :

- Table Personne

Numéro	Nom	Prénom	Age	Ville
5	Durand	Caroline	29	Paris
1	Germain	Stan	32	Lyon
2	Claude	Anna	32	Paris
12	Dupont	Lisa	54	Paris
3	Germain	Rose-Marie	6	Montpellier

- Table Cadeau

Age	Article	Prix
29	livre	45
6	poupée	25
32	montre	87

Donnez les résultats des requetes suivantes :

- πAge(Personne)
- πAge(σNom= ’Caroline’(PERSONNE))
- Personne ⋈ Cadeau

Exercice 3 :

Supposons le schéma relationnel suivant dans lequel les clés primaires apparaissent en gras et les clés étrangères en italique:

- EtudiantsLicence (**NumEtd**, NomEtd, PrénomEtd, DateNaissanceEtd, AdresseEtd, MoyGenEtd)
- EtudiantsMaster (**NumEtd**, NomEtd, PrénomEtd, DateNaissanceEtd, AdresseEtd, MoyGen)
- Livre(**NumLivre**,TitreLivre,*NumAuteur*,*NumEditeur*,*NumTheme*,AnneeEdition)
- Auteur(**NumAuteur**,NomAuteur,AdresseAuteur)
- Editeur(**NumEditeur**,NomEditeur,AdresseEditeur)
- Theme(**NumTheme**,IntituléTheme)
- Prêt(**NumEtd**,**NumLivre**,**DatePret**,DateRetour)

Ecrire les requetes suivantes en langage algébrique:

- La date de naissance et adresse de l'étudiante en licence Lina Hfaidh.
- Le nom, le prénom et l'adresse de l'étudiant en master avec le nom 'Salem'.
- Les numeros d'étudiants dont la moyenne generale en Licence est inferieure à 12,5.
- Le nom et l'adresse de l'auteur du livre 'Base de données'.
- Les livres qui sont publié après le 01/01/2020.
- Les livres de l'auteur "Sami" ou 'Belhadj'"publiés par l'éditeur " Springer ".
- La liste des livres qui n'ont jamais été empruntés.
- La liste des noms des étudiants en licence et des titres des livres pour tous les prets d'avant le 17/02/2019.

Exercice 3 :

Soit les quatre schémas de la base de données suivants:

Table Cours :

IdCours	Jour	Heure
Archi	Lu	9h
Algo	Ma	9h
Algo	Ve	9h
Syst	Ma	14h

Table Etudiant :

IdEtudiant	Nom	Adresse
100	Toto	Nice
200	Tata	Paris
300	Titi	Rome

Table Note_Etu :

IdCours	IdEtudiant	Note
Archi	100	A
Archi	300	A
Syst	100	B
Syst	200	A
Syst	300	B
Algo	100	C
Algo	200	A

Table Affectation_Salle :

IdCours	IdSalle
Archi	S1
Algo	S2
Syst	S1

1. Donnez les résultats des requetes suivantes :
   - R1 = π IdCours (Cours)

- R2 = π IdEtudiant (Etudiant)
- R3 = σ IdCours = ’Algo’ (Note_Etu)
- R4 = Cours ⋈ Affectation_Salle
- R5 = π IdEtudiant, IdCours (Note_Etu)
- R6 = R5 ÷ R1

2. Traduire en français les requêtes suivantes qui sont en exprimées en algèbre relationnelle et donner leurs résultats
   - R7 = R2 × R1
   - R8 = R7 – R5
   - R9 = π IdEtudiant (R5)
   - R10 = π IdEtudiant (R8)
   - R11 = R9 – R10
3. Comparez le résultat de R6 avec celui de R11, que représente-t-il ?
4. Exprimer les requêtes suivantes dans le langage algébrique
   - Listez les noms des étudiants qui participent au cours ‘Algo’
   - Préciser les notes en 'Archi' des étudiants ayant pour nom 'Titi'.
   - Indiquez les horaires (jour, heure) durant lesquels la salle 'S1' est réservée à un cours.
   - Identifiez les étudiants qui ont obtenu uniquement des notes 'A'.
   - Précisez la salle dans laquelle se trouve 'Toto' le lundi à 9h00.

Exercice 4 :

Considérons le schéma suivant :

Usine(Uid, Unom, adresse)

Pièce(Pid, pnom,marque, couleur)

Fabrique (Uid, Pid,dateF cout)

Option(oID, description, prix)

Avoir(Pid, oID)

1. Ecrivez les requêtes suivantes en algèbre relationnelle.
   - Trouver les noms des usines qui fabriquent les pièces rouges.
   - Trouver les ID des usines qui fabriquent les pièces rouge ou vertes.
   - Trouver une formulation équivalente que question (b) en utilisant l'opérateur d'union
   - Trouvez les noms des usines qui fabriquent les pièces rouge dont leurs cout est inférieur à 100dt.
   - Trouvez les noms des usines qui fabriquent les pièces avant le 1er novembre 2021.
   - Trouvez les noms des usines qui fabriquent une pièce rouge ou qui sont habités au Tunis.

- Trouvez les identifiants des usines qui fabriquent les pièces rouge et les pièces verte en utilisant l'intersection.
- Trouvez les identifiants des usines qui ne fabriquent que des pièces rouges.
- Chercher les noms des usines qui n'ont pas fournis les pièces rouges.

2. Pour chacune des requêtes d'algèbre relationnelle suivantes, dites ce qu'elles signifient.
   - πUnom(πUid,Unom(σcouleur='rouge'∧ cout<'100' ((Pièce) ⋈ Fabrique) ⋈ Fournisseur) ∩ πUid,Unom(σcouleur='green'∧ cout>'200' ((Pièce) ⋈ Fabrique) ⋈ Fournisseur))
   - πdescription ((π Pid (Pièce) - π Pid (σnom<>'P1'(Pièce)) ⋈ Avoir) ⋈ Options)
   - πmarque(Pièce ×(Avoir ÷ πoID (Options)

Matière : Base de données

Chargée de cours : Farah Barika Ktata
Chargée de TD : Emna Ammar Elhadjamor

Correction TD2

Exercice 1 :

1. Donnez les resultats des requetes suivantes :

ma_cuisine - sa_cuisine

Appareil	Couleur
Réfrigérateur	rouge
Robot	mauve

ma_cuisine – (ma_cuisine - sa_cuisine)

Appareil	Couleur
Cuisinière	jaune

Exercice 2 :

2. Donnez les résultats des requetes suivantes :

- $\pi_{Age}(PERSONNE)$

Age
29
32
32
54
6

- $\pi_{Age}(\sigma_{Nom='\text{ Caroline }'}(PERSONNE))$

Age
29

- *Personne* ⋈ *Cadeau*

Numéro	Nom	Prénom	Age	Ville	Article	Prix
5	Durand	Caroline	29	Paris	livre	45
1	Germain	Stan	32	Lyon	montre	87
2	Claude	Anna	32	Paris	montre	87
3	Germain	Rose-Marie	6	Montpellier	poupée	25

Exercice 3 :

- La date de naissance et adresse de l'étudiante en licence Lina Hfaidh
  π [DateNaissanceEtd, AdresseEtd] σ (PrénomEtd = " Lina " ∧ NomEtd = " Hfaidh " (EtudiantsLicence))
- Le nom, le prénom et l'adresse de l'étudiant en master avec le nom 'Salem'
  π NomEtd, PrenomEtd, AdresseEtd (σ NomEtd = 'Salem' (EtudiantsMaster))

- les numeros d'etudiants dont la moyenne generale en Licence est inferieure à 12,5
  πNumEtu (σMoyGen<12:5(EtudiantsLicence))
- Le nom et l'adresse de l'auteur du livre 'Base de données'
  π NomAuteur, AdresseAuteur (σ TitreLivre = 'Base de données' (Livre ⋈ Auteur))
- Les Livres qui sont publié après le 01/01/2020
  π NumLivre,TitreLivre,NumAuteur,NumEditeur,NumTheme,AnneeEdition (σ AnneeEdition>01/01/2020) (Livre)
- Les Livres de l'auteur "Sami" ou 'Belhadj'"publiés par l'éditeur " Springer ".
  π NumLivre,TitreLivre,NumAuteur,NumEditeur,NumTheme,AnneeEdition (σ NomAuteur = 'Sami' ∨ NomAuteur = 'Belhadj' ∧ NomEditeur='Springer' (Livre ⋈ Editeur) ⋈ Auteur)
- La liste des livres qui n'ont jamais été empruntés
  π NumLivre,TitreLivre,NumAuteur,NumEditeur,NumTheme,AnneeEdition ((π NumLivre (Livre) - π NumLivre (Prêt)) ⋈ Livre)
- La liste des noms des étudiants en licence et des titres des livres pour tous les prets d'avant le 17/02/2019.
  πNomEtd; TitreLivre (σ DatePret <17/02/2019(σ EtudiantsLicence .NumEtu = Prêt .NumEtu ^ Livre.NumLivre = Prêt .NumLivre (Prêt ⋈ EtudiantsLicence ⋈Livre)

Exercice 3 :

1. Donnez les résultats des requetes suivantes :

- R1 = π IdCours (Cours)
- R2 = π IdEtudiant (Etudiant)

R1

IdCours
Archi
Algo
Syst

R2

IdEtudiant
100
200
300

- R3 = σ IdCours = 'Algo' (Note_Etu)

R3

IdCours	IdEtudiant	Note
Algo	100	C
Algo	200	A

- R4 = Cours ⋈ Affectation_Salle

R4

IdCours	Jour	Heure	IdSalle
Archi	Lu	9h	S1
Algo	Ma	9h	S2
Algo	Ve	9h	S2
Syst	Ma	14h	S1

- R5 = π IdEtudiant, IdCours (Note_Etu)

R5	IdEtudiant	IdCours
	100	Archi
	300	Archi
	100	Syst
	200	Syst
	300	Syst
	100	Algo
	200	Algo

- R6 = R5 ÷ R1

R6	IdEtudiant
	100

2. Traduire en français les requêtes suivantes qui sont en exprimées en algèbre relationnelle

- R7 = R2 × R1 : ensemble de toutes les inscriptions possibles

R7	IdEtudiant	IdCours
	100	Archi
	100	Algo
	100	Syst
	200	Archi
	200	Algo
	200	Syst
	300	Archi
	300	Algo
	300	Syst

- R8 = R7 – R5 : ensemble des inscriptions manquantes

R8	IdEtudiant	IdCours
	200	Archi
	300	Algo

- R9 = π IdEtudiant (R5): liste des étudiants qui sont inscrits à certains cours

R9	IdEtudiant
	100
	200
	300

- R10 = π IdEtudiant (R8): liste des étudiants qui ne sont pas inscrits à certains cours

R10	IdEtudiant
	200
	300

- R11 = R9 – R10 : liste des étudiants qui sont inscrits à tous les cours

R11	IdEtudiant
	100

3. Comparez le résultat de R6 avec celui de R11, que représente-t-il ?

   Le résultat est le même, c'est l'ensemble des étudiants qui suivent tous les cours.
4. Exprimer les requêtes suivantes dans le langage algébrique

- Listez les noms des étudiants qui participent au cours 'Algo'
  RES = π Nom (σ IdCours = 'Algo'(Note_Etu ⋈ Etudiant))
- Préciser les notes en 'Archi' des étudiants ayant pour nom 'Titi'.
  RES = π Note (σ (Nom = 'Titi') ^ (IdCours = 'Archi') Note_Etu ⋈ Etudiant)
- Indiquez les horaires (jour, heure) durant lesquels la salle 'S1' est réservée à un cours.
  RES = π (Jour, Heure) (σ IdSalle = 'S1' (Affectation_Salle ⋈ Cours))
- Identifiez les étudiants qui ont obtenu uniquement des notes 'A'
  R1 = σ Note = 'A' (Note_Etu)
  R2 = σ Note <> 'A' (Note_Etu)
  R3 = R1 – R2
  RES = π IdEtudiant (R3)
- Précisez la salle dans laquelle se trouve 'Toto' le lundi à 9h00
  RES = π IdSalle (σHeure = '9h'^ Jour = 'Lu'^ Nom = 'Toto'(((Note_Etu ⋈ Etudiant) ⋈ Cours) ⋈ Affectation_Salle))

Exercice 4 :

1) Ecrivez les requêtes suivantes en algèbre relationnelle.

- Trouver les noms des usines qui fabriquent les pièces rouges.
  πUnom((σcouleur='rouge' (Pièce) ⋈ Fabrique) ⋈Fournisseur))
- Trouver les ID des usines qui fabriquent les pièces rouge ou vertes.
  πUid(σcouleur='rouge' ∨ couleur='green'(Pièce) ⋈ Fabrique)
- Trouver une formulation équivalente que question (b) en utilisant l'opérateur d'union
  πUid(σcouleur='rouge' (Pièce) ⋈ Fabrique) ∪ πUid(σcouleur='green' (Pièce) ⋈ Fabrique)
- Trouvez les noms des usines qui fabriquent les pièces rouge dont leurs cout est inférieur à 100dt.
  πUnom(σcouleur='rouge' ∧ cout<'100' ((Pièce) ⋈ Fabrique) ⋈ Fournisseur)
- Trouvez les noms des usines qui fabriquent les pièces avant le 1er novembre 2021.
  πUnom(σdateF<'01/11/2021' ((Pièce) ⋈ Fabrique) ⋈ Fournisseur)
- Trouvez les noms des usines qui fabriquent une pièce rouge ou qui sont habités au Tunis.
  πUnom(σcouleur='rouge' (Pièce) ⋈ Fabrique ⋈ Fournisseur ∪ σadresse='tunis' (Fournisseur)
- Trouvez les identifiants des usines qui fabriquent les pièces rouge et les pièces verte en utilisant l'intersection.
  πUid(σcouleur='rouge' (Pièce) ⋈ Fabrique) ∩ πUid(σcouleur='green' (Pièce) ⋈ Fabrique)
- Trouvez les identifiants des usines qui ne fabriquent que des pièces rouges.
  R1 = πUid(σcouleur='rouge' (Fabrique ⋈ Pièce))
  R2 = πUid(σcouleur<>'rouge' (Fabrique ⋈ Pièce))
  R3 = R1 – R2
- Chercher les noms des usines qui n'ont pas fournis les pièces rouges.

ΠUnom (Fournisseur) - ΠUnom (σ = Couleur ‘rouge’' (Fournisseur ⋈ Fabrique ⋈ Pièce))

2) Pour chacune des requêtes d'algèbre relationnelle suivantes, dites ce qu'elles signifient.

- πUnom(πUid,Unom(σcouleur=‘rouge’∧ cout<’100’ ((Pièce) ⋈ Fabrique) ⋈ Fournisseur) ∩ πUid,Unom(σcouleur=‘green’∧ cout>’200’ ((Pièce) ⋈ Fabrique) ⋈ Fournisseur))
  →Trouvez les noms des usines qui fabriquent les pièces rouge dont le cout est moins de 100dt et les pièces verte dont leurs cout est supérieur à 200 DT.
- πdescription ((π Pid (Pièce) - π Pid (σnom<>’P1’(Pièce)) ⋈ Avoir) ⋈ Options)
  → description des options qui se trouvent seulement dans les pièces dont leurs nom est « P1 »
- πmarque(Pièce ×(Avoir ÷ πoID (Options)
  → marque des pièces avec toutes les options.

Matière : Base de données

Chargée de cours : Farah Barika Ktata **Chargée de TD :** Emna Ammar Elhadjamor	**TD3**

Exercice 1 :

Quelle valeur la dernière instruction SQL de la liste ci-dessous renvoie-t-elle ?

```
1 CREATE TABLE t (
2 a integer,
3 b integer,
4 c integer);
5 INSERT INTO t VALUES( 0,0,0 );
6 INSERT INTO t VALUES( 1,0,0 );
7 SELECT sum(a) + count(b) FROM t;
```

Exercice 2 :

Soit la base de données LocationChambre permettant de représenter les données relatives à la gestion des locations des chambres dans un hôtel. Cette BD comporte les tables Chambre, TypeChambre, Client et Location.

Table TypeChambre		
Colonne	**Description**	**Contrainte**
NTC	Numéro Type de Chambre de type **SMALLINT**	*clé primaire*
Nom	Nom du type de chambre de type **VARCHAR(10)**	
FL	Frais de location par nuit pour ce type de chambre de type **DECIMAL(6,2)**	

Table Chambre		
Colonne	**Description**	**Contrainte**
NCH	Numéro de Chambre de type **SMALLINT**	clé primaire
NTC	Numéro du type de chambre associé à cette chambre de type **SMALLINT**	clé étrangère
FS	Frais Supplémentaires pour cette chambre de type **DECIMAL(5,2)**	

Table Client		
Colonne	**Description**	**Contrainte**

NC	Numéro Client de type **SMALLINT**	clé primaire
Nom	Nom du Client de type **VARCHAR(12)**	
Prenom	Prénom du Client de type **VARCHAR(12)**	
Adresse	Adresse du Client de type **VARCHAR(30)**	

Table Location		
Colonne	**Description**	**Contrainte**
NC	Numéro Client ayant fait la location de type **SMALLINT**	clé primaire, clé étrangère
NCH	Numéro de chambre loué par ce client de type **SMALLINT**	clé primaire, clé étrangère
DL	Date de location de cette chambre par ce client de type **DATE**	clé primaire
PL	Période de location (en nombre de nuits) de type **SMALLINT UNSIGNED**	

1. Créer la table TypeChambre
2. Insérer les types de chambre suivants :
   (1,'Chambre Double', 70)
   (2,'Chambre Simple', 100)
   (3,'Chambre Double avec TV', 120)
   (4,'Chambre Simple avec TV', 200)
3. Créer la table Chambre
4. Insérer pour chaque type de chambre existant deux ou trois chambres
5. Créer les deux tables 'Client' et 'Location'
6. Insérer des clients dans la table Client
7. Insérer pour chaque client des locations
8. Afficher tous les types de chambre ordonnés par ordre croissant sur les frais de location
9. Afficher les noms et les frais de locations des types de chambre
10. Afficher la valeur maximale et minimale des frais de location des types de chambre
11. Afficher les types de chambre dont les frais de location sont supérieurs à 100 DT
12. Afficher les types de chambre dont les frais de location sont compris entre 70DT et 170DT
13. Afficher le nombre de types de chambre existants
14. Ajouter 10 DT aux frais de location de tous les types de chambre
15. Afficher la somme des frais de location de tous les types de chambre
16. Afficher le ou les types de chambre ayant les plus petits frais de location
17. Modifier le numéro du type de chambre N°3, le nouveau numéro est 30
18. Ajouter 10% aux frais de location de tous les types de chambre

19. Ajouter 15% aux frais de location des types de chambre dont les frais de location sont inférieurs à 150
20. Retrancher 0.5% des frais de location des types de chambre dont le numéro est entre 2 et 4
21. Supprimer les types de chambre dont les frais de locations sont inférieurs à 100 ou supérieurs à 300
22. Afficher les numéros de chambres, les frais de location ainsi que les frais supplémentaires de chaque chambre
23. Modifier la requête précédente en affichant directement la somme des FL et FS
24. Afficher toutes les chambres du type de chambre N°1
25. Afficher toutes les chambres dont les frais de locations sont supérieurs à 280
26. Afficher toutes les chambres dont les frais de locations sont compris entre 180 et 250
27. Afficher toutes les chambres dont le numéro du type de chambre est 1, 3 ou 5
28. Afficher toutes chambres Simples (càd le nom du type de chambre contient le mot 'Simple')
29. Afficher pour chaque type de chambre le numéro du type de chambre ainsi que le nombre de chambres existantes
30. Afficher pour chaque type de chambre le numéro du type de chambre ainsi que la somme et la moyenne des frais supplémentaires
31. Afficher pour chaque type de chambre le numéro du type de chambre ainsi que le maximum et le minimum des frais supplémentaires.
32. Ajouter 20DT aux FS des chambres dont le type de chambre a des frais de location inférieurs à 120DT
33. Retrancher 15DT des FS des chambres 'Double' (càd dont le nom du type de chambre associé contient le mot 'Double')
34. Renommer la table TypeChambre, le nouveau nom est CategorieChambre
35. Ajouter une contrainte d'unicité sur les champs Nom et FL de la table TypeChambre
36. Afficher par étage le nombre de chambres
37. Afficher par étage la somme et la moyenne des frais supplémentaires de toutes les chambres
38. Afficher par étage la somme et la moyenne des frais de locations de toutes les chambres
39. Afficher le nombre de chambres pour lesquelles les frais supplémentaires sont supérieurs à 6
40. Afficher le nombre de chambres Simples pour lesquelles les frais supplémentaires sont inférieurs à 5
41. Afficher par étage le nombre de chambres mais uniquement pour les étages contenant plus de 3 chambres
42. Afficher par étage le nombre de chambres Simples mais uniquement pour les étages dont la somme des frais supplémentaires est inférieure à 18
43. Afficher par type de chambre le minimum et le maximum des frais de locations mais uniquement pour les types de chambre dont le minimum des frais de location est inférieur à 120
44. Afficher par type de chambre le nombre de chambres existant pour lesquelles la moyenne des frais de location est entre 100 et 200
45. Afficher les clients ayant loué la chambre N°2
46. Afficher les clients ayant loué une chambre dont les frais de location dépassent 250

47. Afficher pour chaque location ses frais (FL*PL)
48. Afficher pour chaque location, le nom et le prénom du client, le nom du type de la chambre louée, la période de location, les frais de location par jour (FL) ainsi que les frais globaux de cette location (FL * PL).
49. Afficher les chambres louées avant la date du '1-1-2021'
50. Afficher les chambres louées pour une période comprise entre 2 et 5 jours
51. Afficher pour chaque client le nombre de chambres louées
52. Idem mais uniquement pour des périodes de locations comprises entre 3 et 7 nuits
53. Idem mais uniquement pour les chambres dont les FL ne dépassent pas 150DT
54. Afficher pour chaque client la somme des FL*PL pour toutes les locations faites par ce client
55. Idem mais uniquement pour les clients dont le numéro est 1, 3 ou 5
56. Idem mais uniquement si le nombre de locations dépasse 4
57. Idem mais uniquement si la somme des (FL*PL) est entre 500 et 2000 DT
58. Créer une vue sur les types de chambres dont les frais de locations sont supérieurs à 220

Matière : Base de données

Chargée de cours : Farah Barika Ktata
Chargée de TD : Emna Ammar Elhadjamor

Correction TD3

Exercice 1 :

Quelle valeur la dernière instruction SQL de la liste ci-dessous renvoie-t-elle ?

- Contenus de la table :

  Après Create : Ø

  Après Insert n°1 :

0	0	0

Après Insert n°2 :

0	0	0
1	0	0

- On a sum(a) = 1 et count(b) = 2 donc La requête renvoie 3

Exercice 2 :

1. Créer la table TypeChambre

```
create table typechambre(
   NTC smallint(2) not null primary key,
   Nom varchar(10) not null,
   FL decimal(6,2) not null,
   );
```

2. Insérer les types de chambre suivants :

(1,'Chambre Double', 70)
(2,'Chambre Simple', 100)
(3,'Chambre Double avec TV', 120)
(4,'Chambre Simple avec TV', 200)

```
insert into typechambre values
   (1,'chambre double',70),
   (2,'chambre simple',100),
   (3,'double.TV',120),
   (4,'simple.TV',200);
```

3. Créer la table Chambre

```
create table chambre (
   NCH smallint(2) not null primary key,
   NTC smallint(2) not null,
   FS decimal(5,2) not null,
   constraint FK foreign key(NTC) references typechambre(NTC)
   );
```

4. Insérer pour chaque type de chambre existant deux ou trois chambres

```
insert into chambre values
   (1,1,20),
   (2,1,20),
   (3,2,40),
   (4,2,40);
```

5. Créer les deux tables 'Client' et 'Location'

```
create table Client (
NC smallint(2) not null primary key,
   Nom varchar(12) not null,
   Prenom varchar(12) not null,
   Adresse varchar(30) not null);

   create table location (
   NC smallint(2) not null,
   NCH smallint(2) not null,
   DL date,
   PL smallint unsigned,
   constraint PK1 primary key(NC,NCH,DL),
   constraint FK1 foreign key(NC) references client(NC),
   constraint FK2 foreign key(NCH) references chambre(NCH)
   );
```

6. Insérer des clients dans la table Client

```
insert into client values
   (1,'seffar','saad','av.mohamed 5 Tunis),
   (2,'zerta','kamal','av slam Ariana),
   (3,'oulidi','younes','avenue la republique Monastir');
```

7. Insérer pour chaque client des locations

```
insert into location values
(1,1,'2008-2-4',15),
   (2,2,'2008-2-6',10),
   (2,3,'2008-3-2',5);
```

8. Afficher tous les types de chambre ordonnés par ordre croissant sur les frais de location

```
select * from typechambre order by FL asc;
```

9. Afficher les noms et les frais de locations des types de chambre

```
select nom,FL from typechambre;
```

10. Afficher la valeur maximale et minimale des frais de location des types de chambre

```
select max(FL),min(FL) from typechambre;
```

11. Afficher les types de chambre dont les frais de location sont supérieurs à 100 DT

```
select NTC from typechambre where FL > 100;
```

12. Afficher les types de chambre dont les frais de location sont compris entre 70DT et 170DT

```
select NTC from typechambre where FL between 70 and 170;
```

13. Afficher le nombre de types de chambre existants

```
select count(NTC) from typechambre;
```

14. Ajouter 10 DT aux frais de location de tous les types de chambre

```
update typechambre set FL = FL + 10 ;
```

15. Afficher la somme des frais de location de tous les types de chambre

```
select sum(FL) from typechambre;
```

16. Afficher le ou les types de chambre ayant les plus petits frais de location

```
select NTC from typechambre where FL = (select min(FL) from typechambre);
```

17. Modifier le numéro du type de chambre N°3, le nouveau numéro est 30

```
update typechambre set NTC = 30 where NTC = 3;
```

18. Ajouter 10% aux frais de location de tous les types de chambre

```
update typechambre set FL = FL + 0.1 * FL;
```

19. Ajouter 15% aux frais de location des types de chambre dont les frais de location sont inférieurs à 150

```
update typechambre set FL = FL + 0.15 * FL where FL < 150;
```

20. Retrancher 0.5% des frais de location des types de chambre dont le numéro est entre 2 et 4

```
update typechambre set FL = FL - 0.005 * FL where NTC between 2 and 4;
```

21. Supprimer les types de chambre dont les frais de locations sont inférieurs à 100 ou supérieurs à 300.

```
delete from typechambre where FL < 100 OR FL > 300;
```

22. Afficher les numéros de chambres, les frais de location ainsi que les frais supplémentaires de chaque chambre

```
select NCH, FL, FS from typechambre T,chambre C where C.NTC=T.NTC;
```

23. Modifier la requête précédente en affichant directement la somme des FL et FS

```
select NCH, sum(FL+FS) from typechambre T,chambre C where C.NTC=T.NTC;
```

24. Afficher toutes les chambres du type de chambre N°1

```
select * from chambre C Where C.NTC=1;
```

25. Afficher toutes les chambres dont les frais de locations sont supérieurs à 280

```
select * from chambre C,typechambre T where C.NTC=T.NTC and FL > 280;
```

26. Afficher toutes les chambres dont les frais de locations sont compris entre 180 et 250.

```
select C.* from chambre C,typechambre T where C.NTC=T.NTC and FL between 180
and 250;
```

27. Afficher toutes les chambres dont le numéro du type de chambre est 1, 3 ou 5.

```
select * from chambre C where C.NTC in (1,3,5);
```

28. Afficher toutes chambres Simples (càd le nom du type de chambre contient le mot ‘Simple’)

```
select * from chambre C,typechambre T where C.NTC=T.NTC and Nom like
"%simple%";
```

29. Afficher pour chaque type de chambre le numéro du type de chambre ainsi que le nombre de chambres existantes

```
select T.NTC, count(NCH) from chambre1 C, typechambre T where C.NTC=T.NTC group by T.NTC;
```

30. Afficher pour chaque type de chambre le numéro du type de chambre ainsi que la somme et la moyenne des frais supplémentaires

```
select NTC, sum(FS), avg(FS) from chambre1 group by NTC;
```

31. Afficher pour chaque type de chambre le numéro du type de chambre ainsi que le maximum et le minimum des frais supplémentaires.

```
select NTC, min(FS), max(FS) from chambre1 group by NTC;
```

32. Ajouter 20DT aux FS des chambres dont le type de chambre a des frais de location inférieurs à 120DT

```
update chambre set FS = FS +20 where NTC in (select NTC from typechambre where FL < 120);
```

33. Retrancher 15DT des FS des chambres 'Double' (càd dont le nom du type de chambre associé contient le mot 'Double')

```
update chambre set FS = FS - 15 where NTC in (select NTC from typechambre where Nom like "%double%");
```

34. Renommer la table TypeChambre, le nouveau nom est CategorieChambre

```
alter table typechambre rename categoriechambre;
```

35. Ajouter une contrainte d'unicité sur les champs Nom et FL de la table TypeChambre

```
alter table categoriechambre add constraint FL_Nom unique(FL,nom);
```

36. Afficher par étage le nombre de chambres

```
select NE, count(NCH) from chambre group by NE;
```

37. Afficher par étage la somme et la moyenne des frais supplémentaires de toutes les chambres

```
select NE, sum(FS),avg(FS) from chambre group by NE;
```

38. Afficher par étage la somme et la moyenne des frais de locations de toutes les chambres

```
select NE, sum(FL),avg(FL) from chambre C, categoriechambre T where C.NTC=T.NTC group by NE;
```

39. Afficher le nombre de chambres pour lesquelles les frais supplémentaires sont supérieurs à 6

```
select count(NCH) from chambre where FS>6;
```

40. Afficher le nombre de chambres Simples pour lesquelles les frais supplémentaires sont inférieurs à 5

```
select count(NCH),FS,Nom from chambre C, categoriechambre T where C.NTC=T.NTC and Nom like "%simple%" and FS < 5;
```

41. Afficher par étage le nombre de chambres mais uniquement pour les étages contenant plus de 3 chambres

```
select NE, count(NCH) from chambre group by NE having count(NC) > 3;
```

42. Afficher par étage le nombre de chambres Simples mais uniquement pour les étages dont la somme des frais supplémentaires est inférieure à 18

```
select NE, count(NCH) from chambre C, categoriechambre T where C.NTC=T.NTC and nom like "%simple%" group by NE having sum(FS) < 18;
```

43. Afficher par type de chambre le minimum et le maximum des frais de locations mais uniquement pour les types de chambre dont le minimum des frais de location est inférieur à 120

```
select NTC,min(FL),max(FL) from categoriechambre group by NTC having min(FL) < 120 ;
```

44. Afficher par type de chambre le nombre de chambres existant pour lesquelles la moyenne des frais de location est entre 100 et 200

```
select count(NC),C.NTC from chambre C, categoriechambre T where C.NTC=T.NTC group by C.NTC having avg(FL) between 100 and 200;
```

45. Afficher les clients ayant loué la chambre N°2

```
select CL.* from client CL, location L, chambre CH where L.NC=CL.NC and L.NCH=CH.NCH and CH.NCH=2;
```

46. Afficher les clients ayant loué une chambre dont les frais de location dépassent 250

```
select CL.NC,CL.Nom,Prenom,CH.NCH,FL from client CL,location L,chambre CH,categoriechambre T
   where L.NC=CL.NC and L.NCH=CH.NCH and CH.NTC=T.NTC and FL>250;
```

47. Afficher pour chaque location ses frais (FL*PL)

```
select L.NC,L.NCH,L.DL,FL,FL*PL from location L,chambre CH,categoriechambre T
   where L.NCH=CH.NCHH and CH.NTC=T.NTC group by L.DL;
```

48. Afficher pour chaque location, le nom et le prénom du client, le nom du type de la chambre louée, la période de location, les frais de location par jour (FL) ainsi que les frais globaux de cette location (FL * PL).

```
select CL.Nom,Prenom,T.Nom,PL,FL,FL*PL from client CL,location L,chambre CH,categoriechambre T
   where L.NC=CL.NC and L.NCH=CH.NCHH and CH.NTC=T.NTC group by DL;
```

49. Afficher les chambres louées avant la date du '1-1-2021'

```
select NCH from location where DL < '2021-1-1';
```

50. Afficher les chambres louées pour une période comprise entre 2 et 5 jours

```
select NCH from location where PL between 2 and 5;
```

51. Afficher pour chaque client le nombre de chambres louées (COUNT(DISTINCT(champ))

```
select CL. NC, Nom, Prenom, count(NCH) from location L, client CL where L.NC=CL.NC group by CL.NC;
```

52. Idem mais uniquement pour des périodes de locations comprises entre 3 et 7 nuits

```
select NCH,count(NC) from location where PL between 3 and 7 group by NCH;
```

53. Idem mais uniquement pour les chambres dont les FL ne dépassent pas 150DT

```
select NCH, count(L.NC) from client CL, location L, chambre CH, categoriechambre T where L.NC=CL.NC and L.NCH=CH.NCH and CH.NTC=T.NTC and FL < 150 and PL between 3 and 7 group by NCH;
```

54. Afficher pour chaque client la somme des FL*PL pour toutes les locations faites par ce client

```
select CL. NC, CL. Nom, Prenom, sum(FL*PL) from client CL, location L, chambre CH, categoriechambre T where L.NC=CL.NC and L.NCH=CH.NCH and CH.NTC=T.NTC group by CL.NC;
```

55. Idem mais uniquement pour les clients dont le numéro est 1, 3 ou 5

```
select CL.NC,CL.Nom,Prenom,sum(FL*PL) from client CL,location L,chambre CH,categoriechambre T where L.NC=CL.NC and L.NCH=CH.NCH and CH.NTC=T.NTC group by CL.NC having CL.NC in (1,3,5);
```

56. Idem mais uniquement si le nombre de locations dépasse 4

```
select CL.NC,CL.Nom,Prenom,sum(FL*PL) from client CL,location L,chambre CH,categoriechambre T where L.NC=CL.NC and L.NCH=CH.NCH and CH.NTC=T.NTC group by CL.NC having count(DL) > 4;
```

57. Idem mais uniquement si la somme des (FL*PL) est entre 500 et 2000 DT

```
select CL. NC, CL. Nom, Prenom, sum(FL*PL) from client CL, location L, chambre CH, categoriechambre T where L.NC=CL.NC and L.NCH=CH.NCH and CH.NTC=T.NTC group by CL.NC having sum(FL*PL) between 500 and 2000;
```

58. Créer une vue sur les types de chambres dont les frais de locations sont supérieurs à 220

```
Create view V as
(Select NTC from categoriechambre where FL>220);
```

Matière : Base de données

Chargée de cours : Farah Barika Ktata
Chargée de TD : Emna Ammar Elhadjamor

TD4

Exercice 1 :

Deux tables sont utilisées :

- La table salarié (No_salarié, Nom_salarié, Grade, Adresse, Date_Embauche, Salaire, Comm,#No_service)
- La table service (No_service, Nom_service)

Donner les requêtes SQL permettant de trouver :

1. Liste des salariés dont le garde est "MANAGER" dans les services 20 et 30
2. Liste des salariés qui habitent à Tunis et qui ont été embauchés avant le 1er janvier 1995 ou après le 1er janvier 1996
3. Liste des salariés qui n'ont pas la fonction "MANAGER" et qui ont été embauchés en 90
4. Les numéros et les noms des salariés de Gabés ayant un nom qui commence par la lettre ‘A’
5. Liste des salariés ayant un nom qui qui se termine par la lettre ‘I’
6. Liste des salariés ayant un "M" et un "A" dans leur nom
7. Liste des salariés ayant deux "A" dans leur nom
8. Liste des salariés qui ont le même nom et une adresse différente
9. Liste des salariés dont le salaire est compris entre 1000 et 2000
10. Liste des salariés ayant une commission
11. Liste des salariés qui ne perçoivent pas de commission et qui ont un salaire supérieur à 2000
12. Le numéro, le nom et le montant de la commission du salarié qui a la plus faible commission non nulle
13. Liste des salariés triés par ancienneté (les derniers embauchés d'abord)
14. Le numéro et le nom du dernier salarié embauché
15. Liste des salariés qui ont été embauchés avant le salarié numéro 12
16. Liste des salariés qui ont été embauchés le même jour que le salarié numéro 20
17. Liste des salariés qui ont été embauchés avant tous les salariés du service numéro 10
18. Liste des salariés de service 'RECHERCHE' embauchés le même jour que quelqu'un du service 'VENTE'
19. Le salaire moyen en tenant compte des commissions
20. Le nombre des salariés pour chaque grade
21. Liste des salariés ayant le salaire le plus élevé dans chaque service

22. Ajouter au salaire des managers 5% du salaire du salarié numéro =10

23. Supprimer les salariés dont le salaire est inférieur au salaire moyen de leur service

24. Augmenter les salaires des managers de 10%

Exercice 2 :

Soit la base de données relationnelle suivante :

Voiture

NumVoit	Marque	Type	Type
1	Peugeot	404	Rouge
2	Citroen	SM	Noire
3	Opel	GT	Blanche
4	Peugeot	403	Blanche
5	Renault	Alpine A310	Rose
6	Renault	Floride	Bleue

Personne

NumAch	Nom	Age	Ville	Sexe
1	Nestor	96	Paris	M
2	Irma	20	Lille	F
3	Henri	45	Paris	M
4	Josette	34	Lyon	F
5	Jacques	50	Bordeaux	M

Vente

DateVente	Prix	NumVoit	NumAch
1985-12-03	10 000	1	1
1996-03-30	70 000	2	4
1998-06-14	30 000	4	1
2000-04-02	45 000	5	2

1. Écrire l'expression du produit cartésien de la table 'voiture' et de la table 'personne'. Combien de lignes et de colonnes possèdent la table « résultat »?
2. Donner les noms des personnes et la marque des voitures qu'elles ont achetées
3. Trouver les enregistrements de la table vente dont le prix est supérieur à 50 000
4. Trouver les voitures blanches ou rouges
5. Trouver les voitures de couleur « Blanche » ou de marque « Peugeot »
6. Trouver les personnes dont l'âge est compris en 40 et 60
7. Trouver les personnes n'habitant pas Paris
8. Trouver l'âge des personnes qui n'habitent pas Paris
9. Trouver l'âge moyen des personnes habitant Paris
10. Trouver le nombre de voitures par marques
11. Trouver la liste des personnes qui ont acheté une voiture, mais pas rouge

12. Afficher le chiffre d'affaires, c'est-à-dire la somme des prix de vente, toutes taxes (20 %) en considérant que la table contient les prix hors taxes. Renommez la colonne « Résultat » en « CA_TTC »

13. Calculer de la moyenne des prix de vente pour la table 'vente'

14. Calculer la moyenne des prix de vente par marques en ne considérant que les marques dont cette moyenne est supérieure à 40 000

15. Afficher les prix qui sont supérieurs à la moyenne

16. Calculer le nombre de personnes de la table 'personne'

17. Calculer la moyenne d'âge par ville à partir de la table 'personne'

18. Calculer le nombre de voitures par marque de la table 'voiture' dont le nombre est supérieur à 1

19. Calculer le nombre de voitures par marque de la table 'voiture' dont la couleur n'est pas « Rouge »

20. Trier par Marque de la table 'voiture'

21. Trier par Prix de la table 'vente' en ordre décroissant

22. Trier par Ville et par Age de la table 'personne'

23. Écrire les requêtes SQL permettant de :

a. Créer la table 'personne' avec les colonnes suivantes :
   - NumAch : type entier, clé de la relation.
   - Nom : type caractère, de taille 30, ne doit pas être vide.
   - Age : type entier, compris entre 16 et 100.
   - Ville: type caractère, de taille 40.
   - Sexe : type caractère, de taille 1, doit contenir « H » ou « F »

b. Créer la table voiture (en tenant compte que la colonne 'couleur' ne peut prendre que les valeurs: 'Rouge' 'Vert' ou 'Bleu') et la table vente (les valeurs identifiantes des personnes 'NumAch' et des voitures 'NumVoit' de la table 'vente' existent bien dans les tables de référence 'personne' et 'voiture)

c. Insérer un enregistrement dans la table 'voiture'

d. Insérer les valeurs suivantes dans la table 'personne' : NumAch : 100, Nom : Essai, Ville : Paris

e. Supprimer des voitures ayant le couleur 'rouge'

f. Modifier le prix de vente (+10 %) des voitures

g. Modifier le nom de la ville 'Paris' par 'Paris-Centre' dans la table 'personne'

h. Ajouter la colonne 'Teinte' de type 'CHAR' à la table 'voiture'

i. Recopier des données de la colonne 'Couleur' dans la colonne 'Teinte'

j. Supprimer la colonne 'Couleur' de la table 'voiture'

Exercice 3 :

Considérons les instructions CREATE TABLE suivantes.

- `CREATE TABLE Artists( name TEXT PRIMARY KEY);`
- `CREATE TABLE Songs( title TEXT PRIMARY KEY, length INT NOT NULL, artistName TEXT NOT NULL, FOREIGN KEY (artistName) REFERENCES Artists(name), CONSTRAINT SongTitleAndArtistNameIsUnique UNIQUE (title, artistName) );`
- `CREATE TABLE Playlists ( id TEXT PRIMARY KEY, name TEXT, owner TEXT);`
- `CREATE TABLE PlaylistSongs ( playlist TEXT REFERENCES Playlists(id), song TEXT, artist TEXT, position INTEGER, FOREIGN KEY (song,artist) REFERENCES Songs(title,artistName), PRIMARY KEY (playlist,position) );`

1. Créez une vue qui, pour une liste de lecture (playlist) pour l'id M123, avec la structure suivante

```
position | song | artist | length
1        | xxx | AAA | 232
2        | yyy | BBB | 187
```

Exercice 4 :

Soit la base de donnée contenant la relation parent (parent, enfant) suivante:

```
+--------+--------+
| parent | enfant |
+--------+--------+
| Ali    | Fatima |
| Ali    | Kacem  |
| Fatima | Amina  |
| Fatima | Aziz   |
| Kacem  | Aziza  |
| Aziz   | Saida  |
| Saida  | Farid  |
+--------+--------+
```

1. Définir les vues suivantes (ou la requête SQL pour créer la relation correspondante):

- `Relation Grand parent GP`
- `Relation frère`

Matière : Base de données

Chargée de cours : Farah Barika Ktata
Chargée de TD : Emna Ammar Elhadjamor

Correction TD4

Exercice 1 :

1. Liste des salariés dont le garde est "MANAGER" dans les services 20 et 30

```
Select * from salarié where Grade = 'MANAGER' and No_service in (20,30);
```

2. Liste des salariés qui habitent à Tunis et qui ont été embauchés avant le 1er janvier 1995 ou après le 1er janvier 1996

```
Select * from salarié where Adresse='Tunis' and Date_Embauche not between '01/01/1995' and '01/01/1996';
```

3. Liste des salariés qui n'ont pas la fonction "MANAGER" et qui ont été embauchés en 90

```
a : Select No_salarié, Grade, Date_Embauche from salarié where Grade != 'MANAGER' and TO_CHAR(Date_ Embauche,'YY') = '90'
B : Select No_salarié, Grade, Date_Embauche from salarié where Grade <> 'MANAGER' and Date_ Embauche between = '01/01/90' and '31/12/90'
```

4. Les numéros et les noms des salariés de Gabés ayant un nom qui commence par la lettre ‘A’

```
select No_salarié, Nom_salarié from salarié where adresse ='Gabes' AND Nom_salarié LIKE 'A%';
```

5. Liste des salariés ayant un nom qui qui se termine par la lettre ‘I’

```
select * from salarié where Nom_salarié LIKE '%I';
```

6. Liste des salariés ayant un "M" et un "A" dans leur nom

```
select No_salarié from salarié where Nom_salarié like '%M%' and Nom_salarié like '%A%'
```

7. Liste des salariés ayant deux "A" dans leur nom

```
select No_salarié from salarié where Nom_salarié like '%A%A%'
```

8. Liste des salariés qui ont le même nom et une adresse différente

```
select s1.Nom_salarié, s1.adresse from salarié s1, salarié s2 where s1.Nom_salarié=s2.Nom_salarié and s1.adresse !=s2.adresse ;
```

9. Liste des salariés dont le salaire est compris entre 1000 et 2000

```
select Nom_salarié, Salaire from salarié where Salaire between 1000 and 2000;
```

10. Liste des salariés ayant une commission

```
select * from salarié where comm IS NOT NULL ;
```

11. Liste des salariés qui ne perçoivent pas de commission et qui ont un salaire supérieur à 2000

```
select * from salarié where Salaire>2000 and comm IS NULL;
```

12. Le numéro, le nom et le montant de la commission du salarié qui a la plus faible commission non nulle.

```
select No_salarié, Nom_salarié, comm from salarié where comm = (select min (comm) from salarié);
```

13. Liste des salariés triés par ancienneté (les derniers embauchés d'abord)

```
select No_salarié, No_service, Grade, Date_Embauche from salarié order by No_service,
Grade, Date_Embauche DESC
```

14. Le numéro et le nom du dernier salarié embauché

```
select No_salarié, Nom_salarié from salarié where Date_Embauche = (select max
(Date_Embauche )from salarié);
```

15. Liste des salariés qui ont été embauchés avant le salarié numéro 12

```
a: select e1.No_salarié, e1.Date_Embauche from salarié e1, salarié bl where No_salarié =
'12'and e1.Date_Embauche<bl.Date_Embauche
b: select No_salarié, Date_Embauche from salarié where Date_Embauche < (select Date_Embauche
from salarié where No_salarié = '12')
```

16. Liste des salariés qui ont été embauchés le même jour que le salarié numéro 20

```
a : select * from salarié where No_salarié<>'20'and Date_Embauche=(select Date_Embauche from
salarié where No_salarié='20')
b : select e1.* from salarié e1, salarié e2 where e1.Date_Embauche=e2.Date_Embauche and
e2.No_salarié='20'
```

17. Liste des salariés qui ont été embauchés avant tous les salariés du service 10

```
select * from salarié where Date_Embauche < (select min(Date_Embauche) from salarié where
No_service = 10)
```

18. Liste des salariés de service 'RECHERCHE' embauchés le même jour que quelqu'un du service 'VENTE'

```
select salarié.* from salarié, service where salarié.No_service=service.No_service and
Nom_service ='RECHERCHE' and
Date_Embauche IN
(select Date_Embauche from salarié, service where salarié.No_service=service.No_service and
Nom_service ='VENTE');
```

19. Le salaire moyen en tenant compte des commission

```
select AVG(salaire+comm) as "Salaire moyen" from salarié;
```

20. Le nombre des salariés pour chaque Grade

```
select Grade, count(*) as "Nombre d'salariés" from salarié GROUP BY Grade ;
```

21. Liste des salariés dayant le salaire le plus élevé dans chaque service

```
select No_service, MAX (salaire) from salarié GROUPE BY No_service
```

22. Ajouter au salaire des managers 5% du salaire No_salarié=10

```
UPDATE salarié E SET salaire= (select E.salaire+0.05*salarié.salaire from salarié where
No_salarié='10') where Grade='MANAGER';
```

23. Supprimer les salariés dont le salaire est inférieur au salaire moyen de leur service

```
DELETE FROM salarié where salaire<(select avg(E.salaire) from salarié E where
E.No_service=salarié.No_service);
```

24. Augmenter les salaires des managers de 10%

```
UPDATE salarié SET salaire=salaire+0.1*salaire where Grade='MANAGER';
```

Exercice 2 :

1. Écrire l'expression du produit cartésien de la table 'voiture' et de la table 'personne'. Combien de lignes et de colonnes possèdent la table « résultat »?

```
SELECT *
FROM personne, voiture ;
```

2. Donner les noms des personnes et la marque des voitures qu'elles ont achetées

```
SELECT vo.Marque, vo.Couleur, ve.Prix, pe.Nom, pe.Age
FROM voiture AS vo, vente AS ve, personne AS pe
WHERE (vo.NumVoit=ve.NumVoit) AND (pe.NumAch=ve. NumAch);
```

3. Trouver les enregistrements de la table vente dont le prix est supérieur à 50 000

```
SELECT *
FROM vente
WHERE Prix > 50 000 ;
```

4. Trouver les voitures blanches ou rouges

```
SELECT *
FROM voiture
WHERE Couleur IN ("Blanc","Rouge")
```

5. Trouver les voitures de couleur « Blanche » ou de marque « Peugeot »

```
SELECT *
FROM voiture
WHERE Couleur="Blanche" OR Marque="Peugeot" ;
```

6. Trouver les personnes dont l'âge est compris en 40 et 60

```
SELECT *
FROM personne
WHERE Age BETWEEN 40 AND 60;
```

7. Trouver les personnes n'habitant pas Paris

```
SELECT *
FROM personne
WHERE NOT (Ville='Paris');
```

8. Trouver l'âge des personnes qui n'habitent pas Paris

```
SELECT Age, Ville
FROM personne
WHERE NOT (Ville='Paris') ;
```

9. Trouver l'âge moyen des personnes habitant Paris

```
SELECT AVG(Age) AS Moyenne_Paris
FROM Personne
WHERE Ville='Paris'
```

10. Trouver le nombre de voitures par marques

```
SELECT Marque, COUNT(*) AS Nombre
FROM voiture
GROUP BY Marque ;
```

11. Trouver la liste des personnes qui ont acheté une voiture, mais pas rouge

```
SELECT pe.Nom
FROM voiture AS vo JOIN vente AS ve JOIN personne AS pe ON (vo.NumVoit=ve.NumVoit)
AND (pe.NumAch=ve. NumAch)
WHERE vo.Couleur != 'Rouge';
```

12. Afficher le chiffre d'affaires, c'est-à-dire la somme des prix de vente, toutes taxes (20 %) en considérant que la table contient les prix hors taxes. Renommez la colonne « Résultat » en « CA_TTC »

```
SELECT SUM(Prix*1.2) AS CA_TTC
FROM vente ;
```

13. Calculer de la moyenne des prix de vente pour la table 'vente'

```
SELECT AVG(Prix) AS Prix_Moyen
FROM vente;
```

14. Calculer la moyenne des prix de vente par marques en ne considérant que les marques dont cette moyenne est supérieure à 40 000

```
SELECT vo.Marque, AVG(ve.Prix) AS Moyenne
FROM voiture vo, vente ve where vo.NumVoit=ve.NumVoit
GROUP BY vo.Marque
HAVING Moyenne > 40000;
```

15. Afficher les prix qui sont supérieurs à la moyenne

```
SELECT ve1.Prix
FROM vente AS ve1
WHERE ve1.Prix > (SELECT AVG(ve2.Prix) FROM vente AS ve2)
```

16. Calculer le nombre de personnes de la table 'personne'

```
SELECT COUNT(*) AS Nombre_Personne
FROM personne ;
```

17. Calculer la moyenne d'âge par ville à partir de la table 'personne'

```
SELECT Ville, AVG(Âge) AS Moyenne_Age
FROM personne
GROUP BY Ville ;
```

18. Calculer le nombre de voitures par marque de la table 'voiture' dont le nombre est supérieur à 1

```
SELECT Marque, COUNT(*) AS Compte
FROM voiture
GROUP BY Marque
HAVING Compte > 1;
```

19. Calculer le nombre de voitures par marque de la table 'voiture' dont la couleur n'est pas « Rouge »

```
SELECT Marque, COUNT(*) AS Compte
FROM voiture
WHERE NOT (Couleur='Rouge')
GROUP BY Marque;
```

20. Trier par Marque de la table 'voiture'

```
SELECT Marque, Type
FROM voiture
ORDER BY Marque ;
```

21. Trier par Prix de la table 'vente' en ordre décroissant

```
SELECT Prix, DateVente
FROM vente
ORDER BY Prix DESC ;
```

22. Trier par Ville et par Age de la table 'personne'

```
SELECT Nom, Age, Ville
FROM personne
ORDER BY Ville, Age ;
```

23. Écrire les requêtes SQL permettant de :

a. Créer la table ‘personne’:

```
CREATE TABLE personne
(NumAch INT PRIMARY KEY,
Nom CHAR(30) NOT NULL,
Age INT,
Ville CHAR(40),
Sexe CHAR(1),
CHECK (Age BETWEEN 16 AND 100),
CHECK Sexe IN (‘H’,’F’)
);
```

b. Créer la table voiture et la table vente

```
CREATE TABLE voiture(
NumVoit INT PRIMARY KEY,
Marque CHAR(30) NOT NULL,
Type CHAR(20),
Couleur CHAR(40),
CHECK Couleur in (‘Rouge’,’Vert’,’Bleu’)

CREATE TABLE vente (
DateAch DATE,
PRIX INT,
NumAch INT NOT NULL REFERENCES personne(NumAch),
NumVoit INT NOT NULL REFERENCES voiture(NumVoit),
PRIMARY KEY (NumAch, NumVoit)
) ;
```

c. Insérer un enregistrement dans la table ‘voiture’

```
INSERT INTO voiture
(NumVoit, Marque, Couleur)
VALUES (10,’Triumph’,’Bleue’) ;
```

d. Insérer les valeurs suivantes dans la table ‘personne’

```
INSERT INTO personne
(NumAch, Ville, Nom)
VALUES (100,’Paris’,’Essai’) ;
```

e. Supprimer des voitures ayant le couleur ‘rouge’

```
DELETE FROM voiture
WHERE Couleur=‘Rouge’ ;
```

f. Modifier le prix de vente (+10 %) des voitures

```
UPDATE vente
SET Prix=Prix*1.1 ;
```

g. Modifier le nom de la ville ‘Paris’ par ‘Paris-Centre’ dans la table ‘personne’

```
UPDATE personne
SET Ville='Paris-Centre'
WHERE Ville='Paris' ;
```

h. Ajouter la colonne 'Teinte' de type 'CHAR' à la table 'voiture'

```
ALTER TABLE voiture
ADD COLUMN Teinte CHAR(60);
```

i. Recopier des données de 'Couleur' dans 'Teinte'

```
UPDATE voiture
SET Teinte=Couleur ;
```

j. Supprimer la colonne 'Couleur' de la table 'voiture'

```
ALTER TABLE voiture
DROP COLUMN Couleur;
```

Exercice 3 :

```
CREATE VIEW PlaylistM123 AS
( SELECT position, song, artist, length FROM PlaylistSongs, Songs WHERE playlist =
'M123' AND PlaylistSongs.song = Songs.title AND PlaylistSongs .artist =
Songs.artistName ORDER BY position );
```

Exercice 4 :

Relation Grand parent GP

```
create view gp (gp, pf) as
select a.parent, b.enfant
from parent a, parent b
where a.enfant = b.parent;

select * from gp;

+--------+-------+
| gp     | pf    |
+--------+-------+
| Ali    | Amina |
| Ali    | Aziz  |
| Ali    | Aziza |
| Fatima | Saida |
| Aziz   | Farid |
+--------+-------+
```

Relation frère

```
create view frere (f1, f2) as
select a.enfant, b.enfant
from parent a, parent b
where a.parent = b.parent
      and a.enfant > b.enfant;

select * from frere;

+-------+--------+
| f1    | f2     |
+-------+--------+
| Kacem | Fatima |
| Aziz  | Amina  |
+-------+--------+
```