

Base de données

I. Notion de base de données :

- Une donnée : Renseignement que l'on obtient sur quelqu'un ou sur quelque chose → description élémentaire d'une information.
- Une information : Est représentée par des données, qui doivent être interprétées et corrélées pour fournir de l'information.
- Eléments Constituant une information :
 - a. Persistance = mémorisation et disponibilité
 - b. Organisation :
 - Papier
 - Fichier (file) : Les données sont stockées sous forme de fichier.
→ Chaque application de traitement de données utilise un format de fichier de données qui lui est spécifique.
- Une base de données (BD) : est un ensemble exhaustif, persistant et non redondant de données structurées concernant un ou plusieurs domaines du monde réel.

Les modèles des bases de données

- ✓ Le modèle hiérarchique
- ✓ Le modèle réseau
- ✓ Le modèle relationnel

II. Notion de Système de Gestion de Bases de données :

1. Définition d'un SGBD :

- Un système de gestion de base de données (SGBD) est un logiciel qui permet de décrire, modifier, interroger et administrer les données d'une base de données.

2. Les fonctions d'un SGBD :

- **La définition des données** : Le SGBD permet de créer et de décrire les objets de la base de données (tables, liens, utilisateurs...), grâce au Langage de Description des Données (LDD).
- **La manipulation des données** : Le SGBD permet de faire la recherche, l'ajout, la modification et la suppression de données grâce au Langage de Manipulation des Données (LMD).
- **L'intégrité des données** : assurer la cohérence des données.
- **La gestion des accès concurrents** : Le SGBD gère l'accès simultané des utilisateurs à la base de données grâce au Langage de Contrôle des Données (LCD).
- **La confidentialité** : Tous les utilisateurs d'une base de données ne sont pas supposés pouvoir consulter toutes les informations.

3. Structure d'une base de données relationnelle

- Une table est une structure regroupant un ensemble de données relatives à un même sujet.
- Une colonne (champ) représente une propriété élémentaire de l'entité décrite par cette table.
- Une ligne (enregistrement) représente une occurrence du sujet représenté par la table.
- La clé primaire : est un champ ou un ensemble de champs permettant d'identifier de manière unique chaque enregistrement de la table.
- La clé étrangère représente un champ (ou des champs) qui pointe vers la clé primaire d'une autre table.

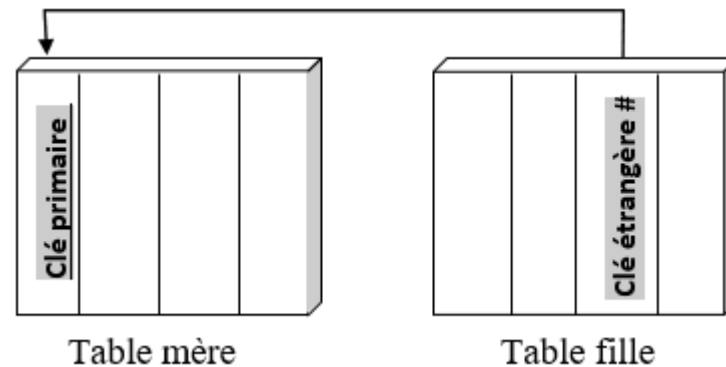
Exemple la table élève est décrit par :

COLONNES					
ELEVE	N°	Prénom	Nom	Date de Naissance	classe
	1	Maha	Ben Ahmed	12/11/1997	4 SI
	2	Ramzi	Ben Amor	17/03/1998	4 EG
	3	Dora	Mokhtar	22/06/1997	4 M

LIGNES

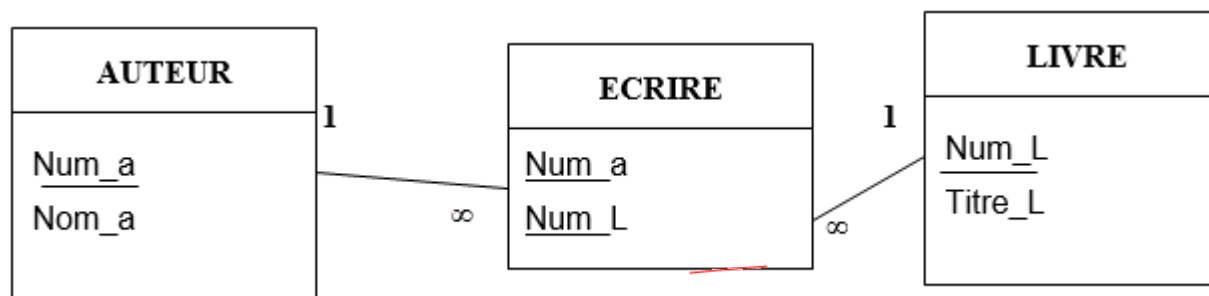
4. Liens entre les tables :

- Liens de type **1, n** : Un lien entre deux tables A et B (A est associée à une ou plusieurs occurrences de B et B est associée à une et une seule occurrence de A).
- La table cible (celle contenant la clé primaire) s'appelle table mère et la table source (celle contenant la clé étrangère) s'appelle table fille.



- Liens de type **n, n** : Ce type de lien entraîne la création d'une troisième table dite intermédiaire (Association) :
Exemple :

- ❖ Un **auteur** peut écrire un ou plusieurs livres
- ❖ Un **livre** peut être écrit par un ou plusieurs auteurs.
- Donne naissance à une association **ECRIRE**



5. Notion de contrainte d'intégrité :

Une contrainte d'intégrité est une règle appliquée à un champ ou à une table et qui doit être toujours vérifiée

- a. **Les contraintes de domaine** : ce sont des contraintes appliquées à des colonnes (valide si).

Exemple : une moyenne n'est valide que si elle est comprise entre 0 et 20.

b. **Les contraintes d'intégrité de tables** : permettent d'assurer que chaque table a une clé primaire

c. **Les contraintes d'intégrité référentielle** :

- Permettent de s'assurer que les valeurs introduites dans une colonne figurent dans une autre colonne en tant que clé primaire.
- La suppression d'un enregistrement d'une table mère A utilisé par une table B fille est interdit.

6. Représentation de la structure d'une BD

La représentation des différentes structures d'une BD est appelée schéma ou modèle.

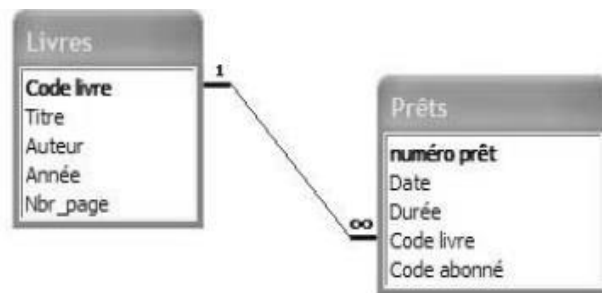
Cette représentation peut être effectuée selon deux formalismes :

1. La représentation textuelle :

Livre (Code livre, Titre, Auteur, Année, Nbre de page)

Prêts (Numéro prêt, Date, Durée, Code-abonné, Code livre #)

2. La représentation graphique :



III. Création et modification de la structure d'une base de données

Langage de définition des données (LDD) : Permet de modifier la structure de la base de données.

Création d'une table :

CREATE TABLE nom_table (définition_ colonne | définition_ contrainte, ...);

Exemple : Create table produit (code int(3) PRIMARY KEY, libelle varchar(30) not null, prix_unit decimale(8,3) check(pu>0));

Modification de la structure d'une BD :

```
ALTER TABLE nom_table  
[ADD COLUMN définition_colonne]  
[ADD CONSTRAINT définition_contrainte]  
[MODIFY définition_colonne]  
[DROP COLUMN nom_colonne]  
[DROP CONSTRAINT nom_contrainte]  
[ENABLE | DISABLE nom_contrainte] ;
```

- **ADD COLUMN** : Ajouter une nouvelle colonne

Exemple : Alter table produit ADD COLUMN (Qte_stock int(3)) ;

- **ADD CONSTRAINT**: Ajouter une nouvelle contrainte

Exemple : Alter table produit ADD CONSTRAINT check(Qte_stock>=0);

- **MODIFY COLUMN** : modifier la structure d'une colonne

Exemple : Alter table produit MODIFY COLUMN (libelle varchar(50)) ;

- **DROP COLUMN** : Supprimer une colonne

Exemple : Alter table produit DROP COLUMN (Qte_stock) ;

- **DROP CONSTRAINT**: Supprimer une contrainte

Exemple : Alter table produit DROP CONSTRAINT check(Qte_stock>=0);

- **DISABLE CONSTRAINT**: Désactiver une contrainte

Exemple : Alter table produit DISABLE CONSTRAINT PRIMARY KEY;

- **ENABLE CONSTRAINT**: Activer une contrainte

Exemple : Alter table produit ENABLE CONSTRAINT PRIMARY KEY;

Suppression d'une table :

```
DROP TABLE nom_table ;
```

Exemple : DROP table produit ;

IV. Manipulation d'une base de données

Langage de manipulation des données (LMD) : Permet de consulter / modifier le contenu de la base de données, Mise à jour des données.

- **Insertion de ligne :**

Syntaxe : insert into nom_table (colonne1, colonne2,...) values (valeur1, valeur2,...) ;

Exemple : insert into client(code,nom,adresse) values (11,"Ines", "Nabeul") ;

Ou bien, on peut écrire aussi : insert into client values (11,"Ines", "Nabeul") ;

- **La modification d'une ligne :**

Syntaxe : update nom_table set champ1 = expression1, champ2=expression2,... [where condition] ;

Exemple : update produit set prix_unit=prix_unit + 100 where code=11

- **Suppression d'une ligne :**

Syntaxe : Delete from nom_table [where condition];

Exemple : Delete from client where code_cli=11;

- **Recherche de données :**

La recherche peut concerner :

- Une projection (sans la clause where) : Afficher certaines colonnes d'une table.

Syntaxe : select colonne1, colonne2,... from nom_table ;

Exemple : Afficher les noms et les adresse de tous les clients

Select nom, adresse from client ; Ou bien : Select * from client ;

- Une sélection (avec la clause where) : Afficher certaines lignes d'une table.

Syntaxe : select colonne1, colonne2,... from nom_table where condition ;

Exemple : Afficher les noms et les adresse de tous les clients de Tunis

Select nom, adresse from client where Adresse="Tunis" ;

Ou bien afficher dans l'ordre alphabétique, toutes les informations de tous les clients de Bizerte

Select * from client where Adresse="Bizerte" order by nom;



❖ Ecriture des conditions : Opérateurs de comparaison :

● = égal	● WHERE Continent = 'Asie'
● <> différent	● WHERE NomF <> 'Sarra'
● > plus grand que	● WHERE Population > 8
● >= plus grand ou égal	● WHERE Population >= 8
● < plus petit que	● WHERE Surface < 83
● <= plus petit ou égal	● WHERE Surface <= 83

❖ Ecriture des conditions : Opérateurs logiques :

- Tous les prédicats des prédicats : AND

WHERE population<10 AND surface<500

- Un des prédicats prédicats : OR

WHERE population<10 OR surface<500

- Négation de la condition : NOT

/*Noms des pays différents ayant le même nom de capitales*/

SELECT P1.nom, P2.nom, P1.capitale FROM PAYS P1, PAYS P2

WHERE P1.capitale = P2.capitale

AND NOT P1.nom = P2.nom ;

❖ Ecriture des conditions : IN, BETWEEN, LIKE

- ✓ **IN, NOT IN :**

WHERE monnaie = 'Dinar' OR monnaie = 'Schilling' OR monnaie = 'Franc' ;

→ WHERE monnaie IN ('Dinar', 'Schilling', 'Franc') ;

- ✓ **BETWEEN, NOT BETWEEN :**

WHERE population BETWEEN 50 AND 60 ;

- ✓ **LIKE, NOT LIKE : Condition partielle**

WHERE pays LIKE '%lande' ;

(Irlande, Islande, Finlande, Hollande)



WHERE pays LIKE 'I_lande' ;
(Irlande, Islande)

Remarque :

% : 0 ou n caractères

_ : exactement 1 caractère

Notion de Jointure :

Une jointure : Recherche à partir de plusieurs tables.

Exemple : Soit la base de données BD_location suivante :

Client (code, Nom, ville)

Voiture (matricule, Marque, couleur, prix_jour)

Location (Code, matricule, date_loc, nb_jours_loc)

Q1 : Afficher la liste de clients qui ont loués des voitures après entre le 07/06/2021 et le 19/09/2021

Select C.code, C.Nom From Client C, Location L Where (C.code = L.code) and (date_loc between 07/06/2021 and 19/09/2021) ;

→ Dans la requête ci-dessus, on a une seule jointure

Q2 : Afficher les marques de voitures qui ont été loués par des clients de Tunis et de couleur rouge.

Select marque From Client C, Location L, Voiture V Where (C.code = L.code) and (V.matricule = L.matricule) and (ville="Tunis") and (couleur="rouge") ;

→ Dans la requête ci-dessus, on a deux jointures

Les fonctions agrégats

- **COUNT** Exemple : Calculer le nombre de clients total

Select COUNT(*) from client ;

- **SUM** Exemple : Afficher pour chaque client son nom et son nb_jour_loc total

Select nom, SUM(nb_jour_loc) from client C, location L Where (C.code=L.code) group by nom ;

- **MAX** Exemple : Afficher le prix_jour maximal
Select MAX (prix_jour) from voiture ;
- **MIN** Exemple : Afficher le prix_jour minimal
Select MIN (prix_jour) from voiture ;
- **AVG** Exemple : Afficher la moyenne de nb_jour_loc
Select AVG (nb_jour_loc) from location ;

V. Sécurisation d'une base de données

Langage de contrôle des données (LCD) : Permet de gérer la sécurité de la BD et la cohérence des données.

Utilisateur :

- **Nom user**
- **Mot de passe**
- **Ensemble de privilèges**

Création d'un utilisateur :

```
CREATEUSER NOM_UTILISATEUR IDENTIFIED BY MOT_DE_PASSE ;
```

Modification d'un utilisateur :

```
ALTER USER NOM_UTILISATEUR IDENTIFIED BY MOT_DE_PASSE ;
```

Suppression d'un utilisateur :

```
DROP USER NOM_UTILISATEUR ;
```

Gestion des privilèges

Il existe 2 types de privilèges :

1. **Privilège système** : droit d'exécuter une action particulière sur n'importe quel type d'objet.

Exemples : ALTER ANY (TABLE,...), CREATE ANY (TABLE,...), CREATEUSER, DROPANYTABLE...

Syntaxe : GRANT{PRIVILEGES_SYSTEME}

TO {NOMS_UTILISATEURS | PUBLIC} [WITH ADMINOPTION] ;



Exemple : GRANT CREATE USER TO SARRA WITH ADMINOPTION ;

2. Privilège objet : droit d'exécuter une action spécifique sur un objet spécifique.

Exemples : {SELECT | DELETE | INSERT| ALTER| ALL} ON NOM_TABLE ...

Syntaxe : GRANT {PRIVILEGES_OBJETS}

TO {NOMS_UTILISATEURS| PUBLIC} [WITH GRANT
OPTION];

Exemple : GRANT SELECT ON ETUDIANT TO SARRA WITH GRANT OPTION ;

Retirer les droits :

Syntaxe : Revoke droits On objet From utilisateur ;

Exemple : Revoke all on client from user10 ;



Exercice 1 :

Soit la représentation textuelle suivante :

- Formateur (mat_formateur, nom, prénom)
- Cours (code, lib_c, niv_c)
- Animer (code_c#, mat_formateur#, date_c, nb_heures)

Donner la représentation graphique de la base de données précédente.

1. Répondre par vrai ou faux en se basant sur la représentation textuelle :

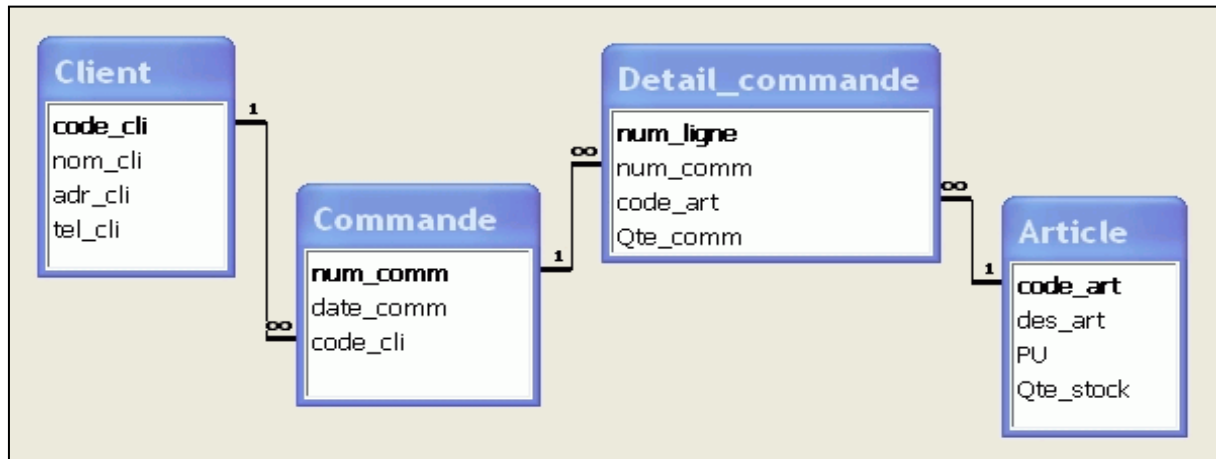
Propositions	Vrai / Faux
Deux formateurs différents peuvent avoir le même nom et le même prénom.	Vrai
Deux formateurs différents peuvent avoir le même identifiant.	Faux
Un formateur peut animer plusieurs cours le même jour.	Faux
Le même cours peut être animé par plusieurs formateurs.	Faux
Un formateur peut animer le même cours plusieurs fois le même jour.	Faux

2. Répondre aux questions suivantes :

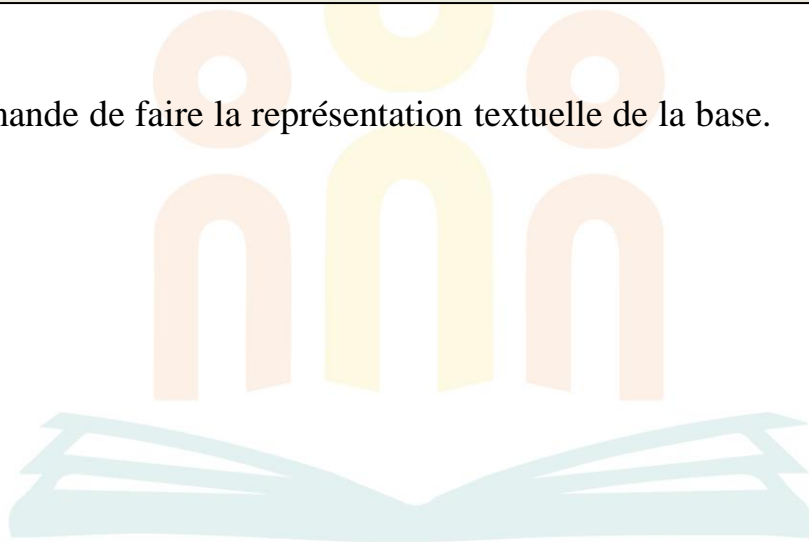
• Le nombre de clés primaires =		• Le nombre de tables mères =	
• Le nombre de clés étrangères =		• Le nombre de tables filles =	

Exercice 2 :

Soit la représentation graphique suivante d'une base de données intitulée « Gestion_Commandes » :



On vous demande de faire la représentation textuelle de la base.



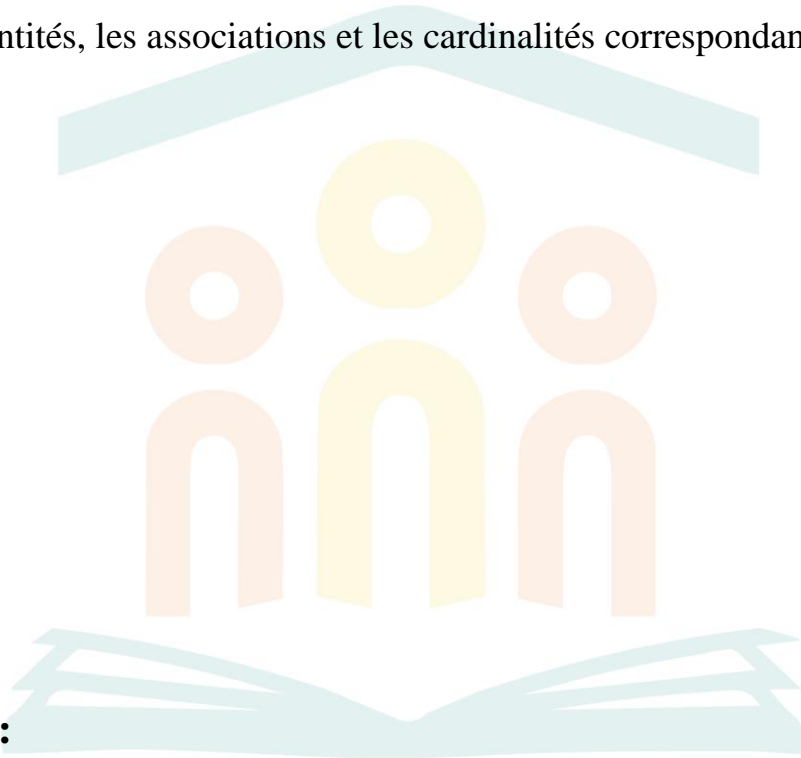
Exercice 3 :

L'agence de voyages **Tunisia Tours** organise des voyages avec des visites d'endroits touristiques.

Le schéma relationnel relatif à son système d'information est décrit par les relations suivantes :

- Client (NoClient, NomClient, Adresse)
- Voyages (NoVoyage, VilleDépart, VilleArrivée, DateDépart, DateRetour, Prix)
- Visite (NoVisite, Endroit)
- Inscription (#NoVoyage, #NoClient, DateInscription)
- Programmes (#NoVisite, #NoVoyage, DateVisite)

Déduire, à partir de ce modèle relationnel, le modèle graphique Tunisia Tours. Préciser les entités, les associations et les cardinalités correspondantes.



Exercice 4 :

On donne la représentation textuelle simplifiée d'une base de données 'Etudiants' concernant un cycle de formation destiné à des étudiants.

ETUDIANT (CodeEt, NomEt, DatnEt)

MATIERE (CodeMat, NomMat, CoefMat)

NOTE (CodeEt#, CodeEns#, Note)

ENSEIGNANT (CodeEns, NomEns, GradeEns, CodeMat#)

Intitulé	Libellé
CodeEt	Code de l'étudiant
NomEt	Nom de l'étudiant
DatnEt	Date de naissance de l'étudiant
CodeMat	Code de la matière
NomMat	Nom de la matière
CoefMat	Coefficient de la matière
Note	Note obtenue par l'étudiant dans une matière
CodeEns	Code de l'enseignant
NomEns	Nom de l'enseignant
GradeEns	Grade de l'enseignant (Grd1, Grd2, ...)

Écrire les requêtes SQL permettant de :

1. Créer la base de données « Etudiants » puis toutes les tables de la base en respectant toutes les contraintes.
2. Afficher les informations relatives aux étudiants (Code, Nom et Date de naissance) selon l'ordre alphabétique croissant du nom.
3. Afficher les noms et les grades des enseignants de la matière dont le nom est 'BD'.
4. Afficher la liste distincte formée des noms et les coefficients des différentes matières qui sont enseignées par des enseignants de grade 'Grd3'.
5. Afficher le nombre d'enseignants de la matière dont le nom est 'TIC'.