Chapitre 1 :

Rappel sur les bases de données et le langage SQL

hamdi.aloulou@isima.u-monastir.tn

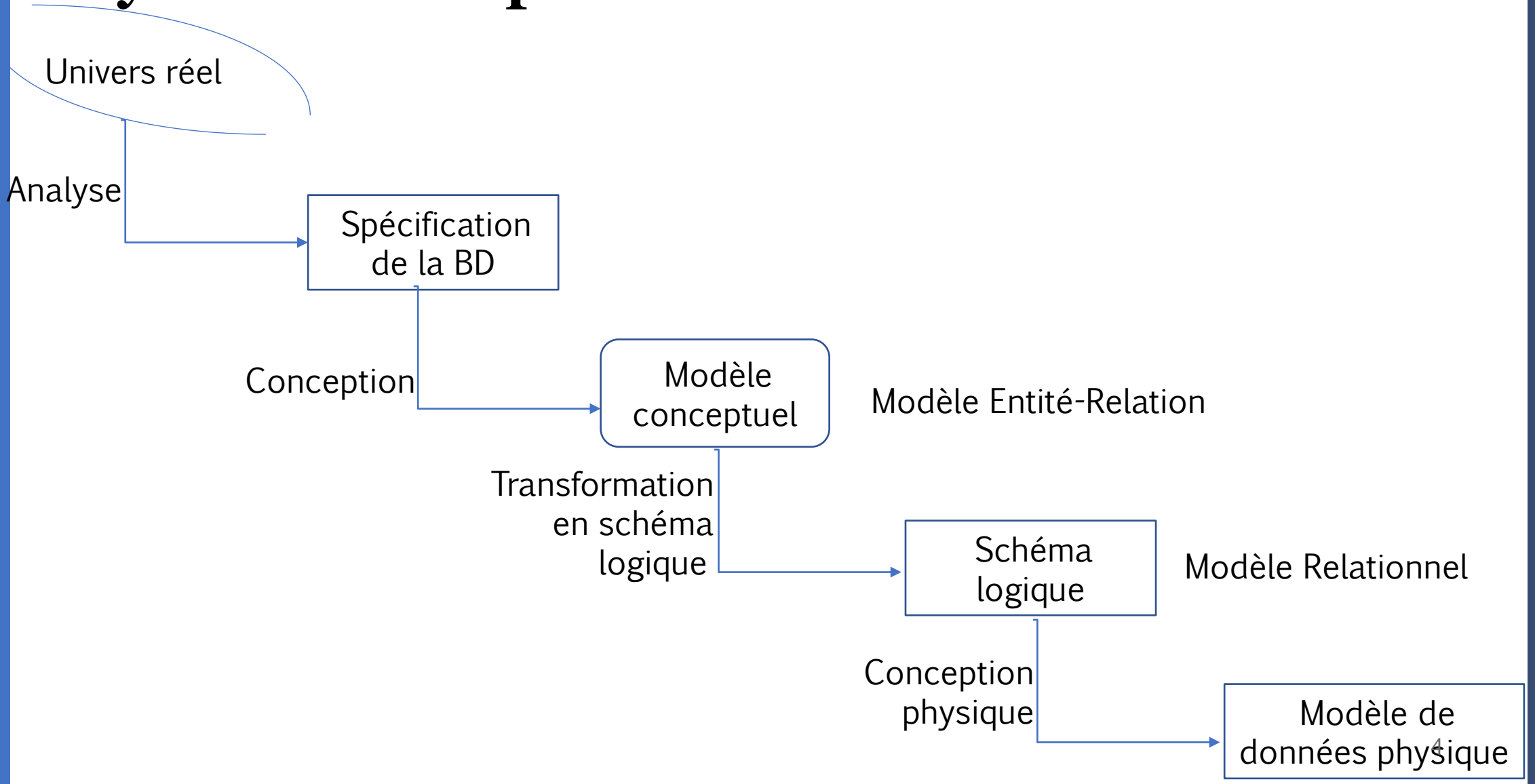
Objectifs du chapitre

- Rappel sur les bases de données et les systèmes de gestion de bases de données
- Rappel sur le langage SQL et ses sous langages

Qu’est ce qu’une base de données

- Une Base de Données (BD) est un **ensemble structuré et organisé de données**.
- Les données sont généralement organisées et structurées pour **modéliser des aspects réels**.
- C’est un « conteneur » **stockant des données de natures différentes** (chiffres, texte, mots, etc.) et plus ou moins **reliées entre elles**.
- L’objectif est de **partager les données**, **réduire la redandance, l’incoherence et la consistence** des données.

Cycle de conception et de création d'une BD



Qu'est ce qu'un Système de Gestion de Base de Données

- SGBD : un *s*ystème de *g*estion de *b*ase de *d*onnées (*D*ata *B*ase *M*anagement *S*ystem) est un logiciel qui **gère, manipule et contrôle l'accès à une BD** partagée par plusieurs utilisateurs simultanément.
- Le SGBD reçoit les **demandes de manipulation de contenu** et **d'administration**, et effectue les opérations nécessaires sur le(s) fichier(s) de la BD.
- Le SGBD **cache la complexité des opérations** et **offre une vue synthétique sur le contenu**.

Que doit permettre un SGBD ?

- **Décrire les données** : Créer et manipuler la structure des tables

  ➔ ***Langage de définition des données (LDD)***

- **Manipuler les données** : insertion, modification et suppression des données

  ➔ ***Langage de manipulation des données (LMD)***

- **Interroger les données** : Répondre à toute demande d'information portant sur les données contenues dans la base.

  ➔ ***Langage d'interrogation des données (LID)***

Que doit permettre un SGBD ?

› **Contrôler les données :** exprimer toutes les règles qui contraignent les valeurs pouvant être enregistrées de façon à éviter toute erreur qui peut être détectée.

- ✓ Intégrité : vérification de contraintes d'intégrité
- ✓ Confidentialité
- ✓ Contrôle des droits d'accès, autorisation

➔ *Langage de contrôle des données (LCD)*

Que doit permettre un SGBD ?

› **Assurer le partage et la sécurité des données** :

- ✓ Une BD est partagée entre plusieurs utilisateurs en même temps ⇒ contrôle des accès concurrents
- ✓ Les données doivent pouvoir être protégées contre les accès non autorisés.
- ✓ Associer à chaque utilisateur des droits d'accès aux données.

Le langage SQL

› SQL (Structured Query Language : langage structuré de requêtes)

› Un standard utilisé par presque tous les ***SGBDR*** modernes tel que Oracle, MySQL, PostgreSQL, SQL Server, …

› SQL est fortement basé sur l'algèbre relationnelle.

Le langage SQL

On peut diviser le langage SQL en 5 parties :

- **Définition de données (LDD)**
  - ✓ CREATE, ALTER, DROP
- **Manipulation de données (LMD)**
  - ✓ INSERT, UPDATE, DELETE
- **Interrogation des données (LID)**
  - ✓ SELECT
- **Contrôle des données (LCD)**
  - ✓ GRANT, REVOKE
- **Contrôle des transactions (LCT)**
  - ✓ COMMIT, ROLLBACK

Langage de définition de données : CREATE TABLE

- **Commande :** CREATE TABLE
- **Besoin de définir les clés primaires et étrangères et les contraintes d'intégrité :**

```
CREATE TABLE nom_table
( nom_col1 type_col1 [[CONSTRAIN nom_contrainte_col1] contrainte_col1],
  nom_col2 type_col2[[CONSTRAIN nom_contrainte_col2] contrainte_col2],
  nom_col3 type_col3 [[CONSTRAIN nom_contrainte_col3] contrainte_col3],
 […],
 [[CONSTRAINT nom_contrainte_tab1] contrainte_tab1],
 [[CONSTRAINT nom_contrainte_tab2] contrainte_tab2],
 […]
);
```

Création de tables : Les contraintes de colonnes

- Les attributs des **contraintes de colonne** peuvent être l'une des contraintes d'intégrité suivantes:
  - [NOT] NULL
  - DEFAULT
  - CHECK **(pour une condition sur une seule colonne)**
  - UNIQUE
  - PRIMARY KEY **(pour une clé primaire simple)**
  - REFERENCES nom_table [(nom-col)*] **(pour une clé étrangère simple)**
- Unique et Primary Key ne peuvent pas être utilisés pour la même colonne.

Création de tables : Les contraintes de table

- Les attributs de contrainte pour les contraintes de table
  - CHECK (conditions) * : **Conditions sur une ou plusieurs colonnes.**
  - UNIQUE (nom-col)* : **Défini une ou plusieurs colonnes comme uniques.**
  - PRIMARY KEY (nom-col)* : **Pour une clé simple ou composée.**
    - **Une clé primaire composé doit impérativement être déclaré comme contrainte de table**
  - FOREIGN KEY (nom-col1)* REFERENCES nom_table [(nom-col2)*] : **Pour une clé étrangère simple ou composée.**
    - **Une clé étrangère composée doit impérativement être déclaré comme contrainte de table**

Langage de définition de données : CREATE TABLE

› Exemple :

```
CREATE TABLE Employe (
Id              INTEGER,
Nom             VARCHAR(32),
Depart          INTEGER,
CONSTRAINT      pk_Id       PRIMARY KEY (Id),
CONSTRAINT      fk_IdDepart  FOREIGN KEY (Depart) REGERENCES Departement(Id)
ON DELETE CASCADE
);
```

Exercice

- REPRESENTATION (n_representation, titre_representation, lieu)

  MUSICIEN (nom, #n_representation)

  PROGRAMME (date, #n_representation, tarif)

- Contraintes d'intégrité :
  - Titre_presentation ne doit pas être nul.
  - Tarif doit être > 0

Exercice

› 1- Créer les tables en respectant les contraintes d'intégrité

CREATE TABLE representation (
n_representation number **PRIMARY KEY**,
titre_representation varchar(30) **NOT NULL**,
Lieu varchar(30));

CREATE TABLE musicien (
Nom varchar(20) **PRIMARY KEY**,
n_representation number **REFERENCES** representation);

Exercice

› 1- Créer les tables en respectant les contraintes d'intégrité

```
CREATE TABLE programme (
date date,
n_representation number REFERENCES representation,
tarif number CHECK(tarif > 0),
PRIMARY KEY(date, n_representation));
```

Langage de définition de données : ALTER TABLE

- **Commande :**

- ALTER TABLE <nom_de_table> **RENAME …**

  ADD …

  MODIFY …

  DROP …

  ENABLE/DISABLE …

Langage de définition de données : ALTER TABLE

- **Commande :** ALTER TABLE
- **Renommer une table :**
  - **ALTER TABLE** <nom_de_table> **RENAME TO** nouv_nom_table;
- **Renommer une colonne :**
  - **ALTER TABLE** <nom_de_table> **RENAME COLUMN** old_name **TO** new_name;
- **Renommer une contrainte**
  - **ALTER TABLE** <nom_de_table> **RENAME CONSTRAINT** <ancient_nom_contrainte> **TO <**nouveau_nom_contrainte>;
- **Ajouter une ou plusieurs colonnes [avec des contraintes de colonnes] :**
  - **ALTER TABLE** <nom_de_table> **ADD (**attribut1 type1 [const_col1], attribut2 type2 [const_col2], ...**)**;
- **Ajouter de nouvelles contraintes de table :**
  - **ALTER TABLE** <nom_de_table> **ADD (**[CONSTRAINT <nom_const_table1>] <contrainte_de_table1>, [CONSTRAINT <nom_const_table2>] <contrainte_de_table2>,..**)**

Langage de définition de données : ALTER TABLE

- **Commande :** ALTER TABLE
- **Modifier les types des colonnes/ajouter de nouvelles contraintes de colonnes:**
  - **ALTER TABLE** <nom_de_table> **MODIFY (**<attribut1 new_type1 [new_constraint1], attribut2 new_type2 [new_constraint2], ...>**)**;
- **Ajouter de nouvelles contraintes de colonnes :**
  - **ALTER TABLE** <nom_de_table> **MODIFY (**<attribut1 [CONSTRAINT <nom_const_col1>] new_constraint1, attribut2 [CONSTRAINT <nom_const_col2>] new_constraint2, ...>**)**;
- **Supprimer une ou plusieurs colonnes :**
  - **ALTER TABLE** <nom_de_table> **DROP (**attribut1, attribut2, ...>**)**;
- **Supprimer une contrainte :**
  - **ALTER TABLE** <nom_de_table> **DROP CONSTRAINT** <nom_contrainte>;
- **Activer ou désactiver une contrainte :**
  - **ALTER TABLE** <nom_de_table> **ENABLE/DISABLE CONSTRAINT** <nom_contrainte>;

Langage de définition de données : DROP TABLE

```
DROP TABLE nom_table [CASCADE CONSTRAINTS];
```

- Pour supprimer une table qui est référencée par une contrainte de clé étrangère d'une autre table, CASCADE doit être spécifié (CASCADE supprimera la contrainte de clé étrangère, pas l'autre table elle-même)
- Exemple :

```
DROP TABLE Departement CASCADE CONSTRAINTS;
```

Exercice

› REPRESENTATION (n_representation, titre_representation, lieu)

MUSICIEN (nom, #n_representation)

PROGRAMME (date, #n_representation, tarif)

› Contraintes d'intégrité :

- Titre_presentation ne doit pas être nul.
- Tarif doit être > 0

Exercice

› 2- Ajouter une colonne duree_presentation de type entier à la table representation

ALTER TABLE representation **ADD**(duree_presentation number);

› 3- Donner la commande pour supprimer la table representation

DROP TABLE representation **CASCADE CONSTRAINTS;**

Langage de manipulation de données : INSERT

```
INSERT INTO nom_table [ (liste de colonnes) ]
        {VALUES (liste de valeurs) | requête};
```

› Exemple :

```
INSERT INTO Employe (Nom, Prenom, CIN, Telephone)
VALUES ('Ben Salem', 'Ahmed', 01876543, '21123456');
```

Langage de manipulation de données : UPDATE

```
UPDATE nom_table
      SET nom_colonne1 = valeur1 [, nom_colonne2 = valeur2 ...]
      [WHERE expression];
```

› Exemple :

```
UPDATE Employe SET Ville= 'Mahdia', Pays = 'Tunisie';
```

Langage de manipulation de données : DELETE

```
DELETE FROM nom_table
      [WHERE expression];
```

› Exemple :

```
DELETE FROM Employe WHERE Ville = 'Mahdia';
```

Exercice

- REPRESENTATION (n_representation, titre_representation, lieu)

  MUSICIEN (nom, #n_representation)

  PROGRAMME (date, #n_representation, tarif)

- Contraintes d'intégrité :
  - Titre_presentation ne doit pas être nul.
  - Tarif doit être > 0

Exercice

› REPRESENTATION (n_representation, titre_representation, lieu)
MUSICIEN (nom, #n_representation)
PROGRAMME (date, #n_representation, tarif)

4- insérer les informations suivantes dans la table representaion :

n_representation	titre_representation	lieu
1	La Cenerentola	Opéra Bastille

Insert Into representation

Values (1, 'La Cenerentola', 'Opéra Bastille');

Exercice

› REPRESENTATION (n_representation, titre_representation, lieu)
MUSICIEN (nom, #n_representation)
PROGRAMME (date, #n_representation, tarif)

5- changer le lieu de tous les representation à Opéra Bastille vers Palais Garnier :

Update representation

Set lieu = 'Palais Garnier'

Where lieu = 'Opéra Bastille';

Exercice

› REPRESENTATION (n_representation, titre_representation, lieu)
MUSICIEN (nom, #n_representation)
PROGRAMME (date, #n_representation, tarif)

6- annuler toutes les représentations programmées avec une tarif inférieur à 25 euros :

Delete from programme

Where tarif < 25;