

bd-initialisation.zip sur site

Système de Gestion de Base de Données

Christophe Viroulaud

Terminale - NSI

**BDD 03**

# Système de Gestion de Base de Données

Christophe Viroulaud

Terminale - NSI

**BDD 03**

## Organisation

Un logiciel

Retour historique

Un langage

## Contraintes d'intégrité

Ouvrir un SGBD

Contrainte de domaine

Contrainte d'entité

Contrainte de référence

Compléter la base de  
données

Le modèle relationnel présenté dans le cours précédent est un modèle mathématique qu'il faut maintenant concrétiser sur machine.

Quels sont les outils permettant de construire une base de données ?

Le modèle relationnel présenté dans le cours précédent est un modèle mathématique qu'il faut maintenant concrétiser sur machine.

Quels sont les outils permettant de construire une base de données ?

## Organisation

[Un logiciel](#)[Retour historique](#)[Un langage](#)Contraintes  
d'intégrité[Ouvrir un SGBD](#)[Contrainte de domaine](#)[Contrainte d'entité](#)[Contrainte de référence](#)[Compléter la base de données](#)

## 1. Organisation

- 1.1 Un logiciel
- 1.2 Retour historique
- 1.3 Un langage

## 2. Contraintes d'intégrité

## Sommaire

## 1. Organisation

## 1.1 Un logiciel

## 1.2 Retour historique

## 1.3 Un langage

## 2. Contraintes d'intégrité

## Organisation

Un logiciel

Retour historique

Un langage

Contraintes  
d'intégrité

Ouvrir un SGBD

Contrainte de domaine

Contrainte d'entité

Contrainte de référence

Compléter la base de  
données

Un **Système de Gestion de Base de Données (SGBD)** est un logiciel permettant de manipuler les données d'une base de données.

# Organisation - un logiciel

Un **Système de Gestion de Base de Données (SGBD)** est un logiciel permettant de manipuler les données d'une base de données.

Ils sont la plupart du temps basés sur un modèle client-serveur :

- la base de données se trouve sur un serveur,
- un logiciel client va interroger le serveur et transmettre la réponse que ce-dernier lui aura donné.

**Remarque**

Un SGBD qui implémente le modèle relationnel est noté SGBDR.

Ils sont la plupart du temps basés sur un modèle client-serveur :

- la base de données se trouve sur un *serveur*,
- un *logiciel client* va interroger le serveur et transmettre la réponse que ce-dernier lui aura donné.

**Remarque**

Un SGBD qui implémente le modèle relationnel est noté SGBDR.



MySQL



Oracle



PostgreSQL



MariaDB



SQLite

FIGURE 1 – Principaux systèmes



FIGURE 2 – Modèles d'accès aux données

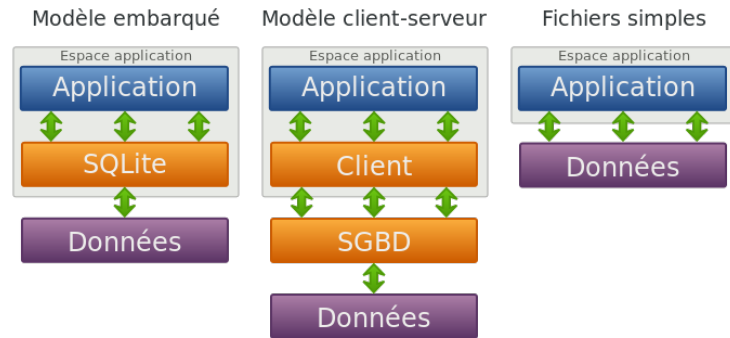


FIGURE 2 – Modèles d'accès aux données

# Sommaire

## 1. Organisation

### 1.1 Un logiciel

### 1.2 Retour historique

### 1.3 Un langage

## 2. Contraintes d'intégrité





FIGURE 3 – avant 1960 : données sur bandes magnétiques

**Remarque**

L'arrivée des stockages à accès direct change la manière

## Retour historique



FIGURE 3 – avant 1960 : données sur bandes magnétiques

**Remarque**

L'arrivée des stockages à accès direct change la manière

remplace modèle hiérarchique



FIGURE 4 – 1970 : Edgar Codd propose le modèle relationnel

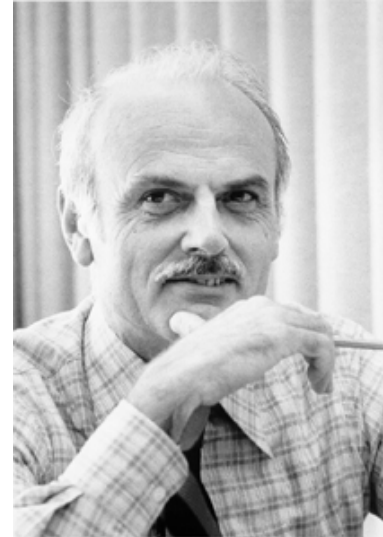


FIGURE 4 – 1970 : Edgar Codd propose le *modèle relationnel*

normalisé en 1986 par ISO (International Organization for Standardization)



FIGURE 5 – 1974 : Donald Chamberlin développe le langage SQL pour communiquer avec les bases de données



FIGURE 5 – 1974 : Donald Chamberlin développe le langage **SQL** pour communiquer avec les bases de données

au début pour missiles embarqués



FIGURE 6 – fin 70 : Oracle (système propriétaire)



FIGURE 7 – 1985 : PostgreSQL (logiciel libre fondé sur une communauté mondiale de développeurs)



FIGURE 7 – 1985 : PostgreSQL (logiciel libre fondé sur une communauté mondiale de développeurs)

le + utilisé



FIGURE 8 – 1995 : Mickael Winedius développe MySQL (Licence GNU - General Public License)



FIGURE 8 – 1995 : Mickael Winedius développe MySQL (Licence GNU - General Public License)

au début pour missiles embarqués



FIGURE 9 – 2000 : Sqlite (n'utilise pas de système client-serveur)



FIGURE 9 – 2000 : Sqlite (n'utilise pas de système client-serveur)

le + utilisé



FIGURE 10 – 2009 : Winedius développe MariaDB suite au rachat de MySQL par Sun puis Oracle



FIGURE 10 – 2009 : Winedius développe MariaDB suite au rachat de MySQL par Sun puis Oracle



# Sommaire

## 1. Organisation

### 1.1 Un logiciel

### 1.2 Retour historique

### 1.3 Un langage

## 2. Contraintes d'intégrité

Les SGBD stockent et optimisent les données de manière efficace mais très complexe. Il n'est pas possible d'y accéder directement. Il faut effectuer des **requêtes** à l'aide d'un langage adapté.

## Un langage

Les SGBD stockent et optimisent les données de manière efficace mais très complexe. Il n'est pas possible d'y accéder directement. Il faut effectuer des **requêtes** à l'aide d'un langage adapté.

**À retenir**

Le SQL (Structured Query Language) est utilisé dans une écrasante majorité des SGBDR.

concurrents : QBE (Query By Example) utilisé dans Microsoft Access (SGBDR de type "fichier")

modèle "fichier simple" : traitement de texte ; .doc, .odt = xml empaqueté

**À retenir**

Le **SQL (Structured Query Language)** est utilisé dans une écrasante majorité des SGBDR.

## Organisation

Un logiciel

Retour historique

**Un langage**Contraintes  
d'intégrité

Ouvrir un SGBD

Contrainte de domaine

Contrainte d'entité

Contrainte de référence

Compléter la base de  
données

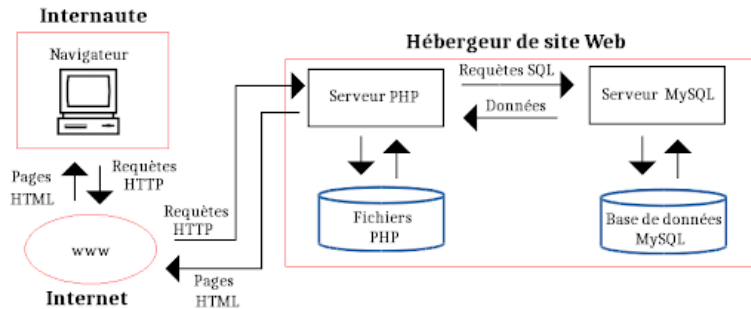


FIGURE 11 – Répartition des rôles

# Sommaire

## 1. Organisation

## 2. Contraintes d'intégrité

### 2.1 Ouvrir un SGBD

### 2.2 Contrainte de domaine

### 2.3 Contrainte d'entité

### 2.4 Contrainte de référence

### 2.5 Compléter la base de données

## Activité 1 :

1. Télécharger et extraire la version portable de *DB Browser for SQLite* depuis le site officiel <https://sqlitebrowser.org/dl/>
2. Télécharger et extraire la base *bd-initialisation.zip* depuis le site <https://cviroulaud.github.io>
3. Ouvrir la base avec le browser.
4. Se concentrer d'abord sur l'onglet *Parcourir les données* et observer les tables existantes.

## Contraintes d'intégrité - Ouvrir un SGBD

## Activité 1 :

1. Télécharger et extraire la version portable de *DB Browser for SQLite* depuis le site officiel <https://sqlitebrowser.org/dl/>
2. Télécharger et extraire la base *bd-initialisation.zip* depuis le site <https://cviroulaud.github.io>
3. Ouvrir la base avec le browser.
4. Se concentrer d'abord sur l'onglet *Parcourir les données* et observer les tables existantes.

# Sommaire

## 1. Organisation

## 2. Contraintes d'intégrité

### 2.1 Ouvrir un SGBD

### 2.2 Contrainte de domaine

### 2.3 Contrainte d'entité

### 2.4 Contrainte de référence

### 2.5 Compléter la base de données

## Système de Gestion de Base de Données

## └─ Contraintes d'intégrité

## └─ Contrainte de domaine

## └─ Contrainte de domaine

## Contrainte de domaine

Aux domaines abstraits du modèle relationnel correspondent les types de données du langage SQL.

Nom du type	Description
SMALLINT	Entier 16 bits signé
INT	Entier 32 bits signé
BIGINT	Entier 64 bits signé
REAL	Flottant 32 bits
CHAR(n)	Chaîne de n caractères exactement
VARCHAR(n)	Chaîne d'au plus n caractères
TEXT	Chaîne de taille quelconque
DATE	Date au format AAAA-MM-JJ
TIME	Heure au format hh:mm:ss
TIMESTAMP	Instant au format AAAA-MM-JJ hh:mm:ss

1. TINYINT 1 octet
2. BOOLEAN garantit que 2 valeurs ; mais pas également supporté

## Contrainte de domaine

Aux domaines abstraits du modèle relationnel correspondent les types de données du langage SQL.

Nom du type	Description
SMALLINT	Entier 16 bits signé
INT	Entier 32 bits signé
BIGINT	Entier 64 bit signé
REAL	Flottant 32 bits
CHAR(n)	Chaîne de n caractères exactement
VARCHAR(n)	Chaîne d'au plus n caractères
TEXT	Chaîne de taille quelconque
DATE	Date au format AAAA-MM-JJ
TIME	Heure au format hh:mm:ss
TIMESTAMP	Instant au format AAAA-MM-JJ hh:mm:ss

## Organisation

Un logiciel

Retour historique

Un langage

Contraintes  
d'intégrité

Ouvrir un SGBD

Contrainte de domaine

Contrainte d'entité

Contrainte de référence

Compléter la base de  
données



## Activité 2 :

1. Quelle est la valeur maximale que peut prendre un **SMALLINT** ?
2. Quelle sa taille en mémoire ?
3. Dans le browser, se rendre dans l'onglet *Structure de la base de données*.
4. Dérouler la table **Auteurs** et repérer les types de chaque attribut.

## Activité 2 :

1. Quelle est la valeur maximale que peut prendre un **SMALLINT** ?
2. Quelle sa taille en mémoire ?
3. Dans le browser, se rendre dans l'onglet *Structure de la base de données*.
4. Dérouler la table **Auteurs** et repérer les types de chaque attribut.

SMALLINT sur 2 octets donc :

- le maximum est  $2^{16} - 1$
- ou  $2^{15} - 1$  si l'entier est signé

**Remarque**

Le SGBD Sqlite simplifie les types (INTEGER, REAL, TEXT) en l'adaptant dynamiquement en fonction de la valeur stockée.

SMALLINT sur 2 octets donc :

- le maximum est  $2^{16} - 1$
- ou  $2^{15} - 1$  si l'entier est signé

**Remarque**

Le SGBD Sqlite simplifie les types (INTEGER, REAL, TEXT) en l'adaptant dynamiquement en fonction de la valeur stockée.

# Sommaire

## 1. Organisation

## 2. Contraintes d'intégrité

### 2.1 Ouvrir un SGBD

### 2.2 Contrainte de domaine

### 2.3 Contrainte d'entité

### 2.4 Contrainte de référence

### 2.5 Compléter la base de données

## Activité 3 :

1. Dans le schéma de la table **Auteurs** comment identifie-t-on la clé primaire ?
2. Quel est le rôle du mot clé **AUTOINCREMENT** ?

# Contrainte d'entité

Chaque entité est identifiée de manière unique grâce à la *clé primaire*.

## Activité 3 :

1. Dans le schéma de la table **Auteurs** comment identifie-t-on la clé primaire ?
2. Quel est le rôle du mot clé **AUTOINCREMENT** ?

- clé primaire : **PRIMARY KEY**
- **AUTOINCREMENT**: augmentation automatique de l'identifiant à la création d'une nouvelle identité

## Correction

- clé primaire : **PRIMARY KEY**
- **AUTOINCREMENT**: augmentation automatique de l'identifiant à la création d'une nouvelle identité

### Organisation

Un logiciel

Retour historique

Un langage

### Contraintes d'intégrité

Ouvrir un SGBD

Contrainte de domaine

**Contrainte d'entité**

Contrainte de référence

Compléter la base de  
données

# Sommaire

## 1. Organisation

## 2. Contraintes d'intégrité

### 2.1 Ouvrir un SGBD

### 2.2 Contrainte de domaine

### 2.3 Contrainte d'entité

### 2.4 Contrainte de référence

### 2.5 Compléter la base de données

Afin de garantir la cohérence des données lors de modifications, on utilise une *clé étrangère*. C'est une référence à une clé primaire d'une autre relation.

**Activité 4 :**

1. Dérouler la table **Bandes\_dessinees**.
2. Rappeler les attributs qui sont des clés étrangères.
3. Glisser la souris sur le schéma de cette table. Quels mots clés sont utilisés pour créer une clé étrangère ?

## Contrainte de référence

Afin de garantir la cohérence des données lors de modifications, on utilise une *clé étrangère*. C'est une référence à une clé primaire d'une autre relation.

**Activité 4 :**

1. Dérouler la table **Bandes\_dessinees**.
2. Rappeler les attributs qui sont des clés étrangères.
3. Glisser la souris sur le schéma de cette table. Quels mots clés sont utilisés pour créer une clé étrangère ?

# Correction

clé étrangère : **FOREIGN KEY(id\_genre) REFERENCES  
Genres.id**



# Sommaire

## 1. Organisation

## 2. Contraintes d'intégrité

### 2.1 Ouvrir un SGBD

### 2.2 Contrainte de domaine

### 2.3 Contrainte d'entité

### 2.4 Contrainte de référence

### 2.5 Compléter la base de données

**Activité 5** : Depuis l'onglet *Exécuter le SQL*, créer les tables **Emprunteurs** et **Emprunts**, en prenant modèle sur les schémas des relations existantes.

**Remarque**

Le langage SQL est insensible à la casse. Nous pouvons écrire indifféremment *CREATE* ou *CreaTE*. Il est d'usage d'écrire les instructions SQL en majuscules.

# Compléter la base de données

**Activité 5** : Depuis l'onglet *Exécuter le SQL*, créer les tables **Emprunteurs** et **Emprunts**, en prenant modèle sur les schémas des relations existantes.

## Remarque

Le langage SQL est insensible à la casse. Nous pouvons écrire indifféremment *CREATE* ou *CreaTE*. Il est d'usage d'écrire les instructions SQL en majuscules.

```

1 CREATE TABLE Emprunteurs( id INTEGER PRIMARY KEY
  AUTOINCREMENT,
2 prenom TEXT,
3 nom TEXT,
4 naissance TEXT);
5
6 CREATE TABLE Emprunts( isbn INTEGER PRIMARY KEY,
7 id_emprunteurs INTEGER,
8 FOREIGN KEY (isbn) REFERENCES Bandes_dessinees(isbn),
9 FOREIGN KEY (id_emprunteurs) REFERENCES Emprunteurs(
  id) );

```

## Correction

```

1 CREATE TABLE Emprunteurs( id INTEGER PRIMARY KEY
  AUTOINCREMENT,
2 prenom TEXT,
3 nom TEXT,
4 naissance TEXT);
5
6 CREATE TABLE Emprunts( isbn INTEGER PRIMARY KEY,
7 id_emprunteurs INTEGER,
8 FOREIGN KEY (isbn) REFERENCES Bandes_dessinees(isbn),
9 FOREIGN KEY (id_emprunteurs) REFERENCES Emprunteurs(
  id) );

```