

# Ingénierie des systèmes d'information

## Concevoir un système d'information

Manon Ansart

ESIREM, LEAD

2023

# La dernière fois



[wooclap.com/ISI3](https://wooclap.com/ISI3)

Propriétés souhaitées pour un système d'information de bonne qualité :

- **Cohérence** : pas des réponses contradictoires
- **Disponibilité** : vitesse de chargement, latence
- **Robustesse** : persistance des données dans le temps même face aux erreurs utilisateurs
- **Confidentialité, sécurité** : accès non autorisé impossible

# Sommaire

- 1 Concevoir sa base de données
- 2 Implémenter son SI
- 3 Optimiser l'accès aux données
- 4 Intégrer le SI dans un environnement plus large

# Plan

## 1 Concevoir sa base de données

- Choix des clés
- Normalisation

## 2 Implémenter son SI

## 3 Optimiser l'accès aux données

## 4 Intégrer le SI dans un environnement plus large

# Clés et clés étrangères

- Clé primaire : Attribut ou liste d'attributs permettant d'identifier un tuple de manière unique
- Clé étrangère :

## Activité

Objectif : assurer la cohérence des données dès la conception  
→ rendre les insertions incohérentes impossibles

Comment ?

- Eviter la redondance de l'information



### Définition : 1FN

Tout attribut a une valeur atomique

Contre-exemples :

- PATIENT(NOM, PRENOM, MEDECIN(PRENOM, NOM))
- PATIENT(NOM, PRENOM, [NOM\_MEDECIN]+)

## Définition : 2FN

Tout attribut non-clé ne dépend pas que d'une partie de la clé

Exemples :

- PATIENT(ID\_PATIENT, ID\_MEDECIN, SPECIALITÉ)
- Proposition

## Définition : 3FN

Tout attribut non-clé dépend directement de la clé

Exemples :

- RDV(ID\_PATIENT, ID\_MEDECIN, SPECIALITÉ, DATE)
- RDV(ID\_RDV, ID\_PATIENT, ID\_MEDECIN, SPECIALITÉ, DATE)
- Exemple

### Définition : 4FN

Il n'est pas possible de construire deux tables **plus petites** permettant de recréer cette table.

Exemples :

- PRESCRIPTION(DATE, TRAITEMENT, ID\_PATIENT, ID\_MEDECIN)
- RDV(ID\_RDV, ID\_PATIENT, ID\_MEDECIN, DATE) avec PRESCRIPTION(ID\_RDV, TRAITEMENT)

# Pourquoi normaliser ?

## Avantages :

- Limiter les redondances
- Assurer la cohérence
- Diminuer la volumétrie
- Limiter le nombre des mises à jour

## Inconvénients :

- Beaucoup des tables
- Temps d'accès potentiellement plus longs, nombreuses jointures

# Plan

- 1 Concevoir sa base de données
- 2 Implémenter son SI
  - Contraintes d'intégrités
  - Triggers
  - Vues
  - Autres outils
- 3 Optimiser l'accès aux données
- 4 Intégrer le SI dans un environnement plus large

# Contraintes d'intégrités

## Définition : Contrainte d'intégrité

Assertion vérifiée par les données, à tout moment

- Géré par le SGBD : variations d'un SGBD à l'autre, mais toujours le même principe
- Les assertions sont généralement vérifiées **après chaque mise à jour**
- Met en jeu une ou plusieurs tables
- SQLite : NOT NULL, UNIQUE, CHECK, FOREIGN KEY

# Exemple SQLite

PRAGMA foreign\_keys=ON

```
CREATE TABLE artist(  
  artistid    INTEGER PRIMARY KEY,  
  artistname  TEXT  
);
```

```
CREATE TABLE track(  
  trackid     INTEGER,  
  trackname   TEXT,  
  trackartist INTEGER,  
  FOREIGN KEY(trackartist) REFERENCES artist(artistid)  
);
```

<https://www.sqlite.org/foreignkeys.html>



# Exemple SQLite

```
sqlite> SELECT * FROM artist;
artistid  artistname
-----
1         Dean Martin
2         Frank Sinatra

sqlite> SELECT * FROM track;
trackid   trackname      trackartist
-----
11        That's Amore   1
12        Christmas Blues 1
13        My Way      2

sqlite> -- This fails because the value inserted into the trackartist column (3)
sqlite> -- does not correspond to row in the artist table.
sqlite> INSERT INTO track VALUES(14, 'Mr. Bojangles', 3);
SQL error: foreign key constraint failed
```

<https://www.sqlite.org/foreignkeys.html>

# Exemple SQLite

```
CREATE TABLE SALLE  
(ID_SALLE INTEGER PRIMARY KEY,  
TYPE TEXT NOT_NULL,  
CAPACITE INTEGER CHECK (CAPACITE > 0));
```

## Définition : trigger

Opération effectuée sur la base de données lorsqu'un évènement à lieu

- Définition de base en SQL, extensions (PL/SQL...)
- Variations d'une extension à l'autre, mais toujours le même principe
- Avant ou après (BEFORE/AFTER) INSERT, UPDATE ou DELETE

# Exemple SQLite

```
CREATE TRIGGER update_customer_address UPDATE OF address ON customers
BEGIN
    UPDATE orders SET address = new.address WHERE customer_name = old.name;
END;
```

With this trigger installed, executing the statement:

```
UPDATE customers SET address = '1 Main St.' WHERE name = 'Jack Jones';
```

causes the following to be automatically executed:

```
UPDATE orders SET address = '1 Main St.' WHERE customer_name = 'Jack Jones';
```

[https://www.sqlite.org/lang\\_createttrigger.html](https://www.sqlite.org/lang_createttrigger.html)

- Les triggers sont peu utilisés
  - Flemme
  - Pas facile à vérifier/tester
- Il paraît souvent plus facile d'effectuer ses opération côté application
- Mais utiliser triggers (et contraintes d'intégrités) est une meilleure pratique (plus grand contrôle, plus fiable)

## Définition : vue

Synthèse d'une requête sur la base données. Aucune table n'est créée/stockée, c'est une table virtuelle calculée dynamiquement à chaque utilisation de la vue.

```
CREATE VIEW PROF_ESIREM(ID, NOM, PRENOM, ADRESSE, FONCTION)
AS SELECT E.ID, E.NOM, E.PRENOM, E.ADRESSE, A.FONCTION
FROM EMPLOYES_UB E, AFFECTATION_EMPLOYES_UB A
WHERE E.ID = F.ID AND
      A.COMPOSANTE = 'ESIREM'
```

# Objectif des vues

- Limiter qui voit quoi (le personnel administratif de l'ESIREM n'a accès qu'aux employés de l'ESIREM)
- Plus pratique : ne pas taper les mêmes commandes plein de fois

## Attention

Les vues sont calculées dynamiquement à chaque fois, attention à la rapidité !

## Activité



Les curseurs servent à traiter une table tuple par tuple (et non toute la table d'un coup).

- Similaire à un fichier qu'on accède ligne par ligne : on passe à la suivante, on ne revient pas en arrière
- Traitement complexe ligne par ligne
- C'est une pratique assez avancée
- Nous l'utiliserons en TP, pour exécuter une requête SQL depuis Python et récupérer le résultat.

En pratique :

- on crée une variable
- on crée un curseur
- on utilise le curseur pour mettre des valeurs dans la variable (FETCH), avec une mise à jour dans une boucle
- on ferme et désalloue le curseur

En pratique :

- on crée une variable
- on crée un curseur
- on utilise le curseur pour mettre des valeurs dans la variable (FETCH), avec une mise à jour dans une boucle
- on ferme et désalloue le curseur

Activité

# Plan

- 1 Concevoir sa base de données
- 2 Implémenter son SI
- 3 Optimiser l'accès aux données
  - Index
  - Arbres
- 4 Intégrer le SI dans un environnement plus large

# Rappel : index

## Définition

Un index est un fichier structuré (table de correspondance) contenant pour chaque **clé** l'adresse de l'enregistrement correspondant

Propriétés :

- accélère la recherche (select) et donc les jointures
- insertion et modification potentiellement plus lentes
- il faut trouver un équilibre !

# Exemples d'index

## Exemple 1 : Index sur les attributs

- index des termes techniques utilisés dans un livre ou d'ingrédients pour un livre de recette
- l'attribut (mot clé, ingrédient) est clé de l'index
- **index dense** : tous les enregistrements sont présents dans l'index

# Exemples d'index

## Exemple 2 : Index sur des clés

- index des numéros étudiants, ou des mots dans le dictionnaire
- le fichier (base des étudiants, dictionnaire) est **ordonné** selon la clé
- la clé de la table (numéro étudiant, mot) est clé de l'index
- index non-dense : seulement certains enregistrements sont présents dans l'index (un par bloc), et permettent de retrouver les autres plus rapidement dans le fichier ordonné



# Types d'index

- Index dense : contient toutes les valeurs de la clé
- Index non-dense : ne contient qu'une clé par bloc

# Choix des index

- Il est possible de faire un index sur plusieurs colonnes, mais c'est au delà du cadre de ce cours
- Toujours se demander si ça vaut le coup : recherche vs insertion / modification
- Les clés des tables sont déjà indexées
- Si une clé est composée de plusieurs attributs, ils sont généralement indexés ensemble et non individuellement

Choix des colonnes :

- Where
- Jointures

