

Introduction à la conception d'une base de données

Walter RUDAMETKIN

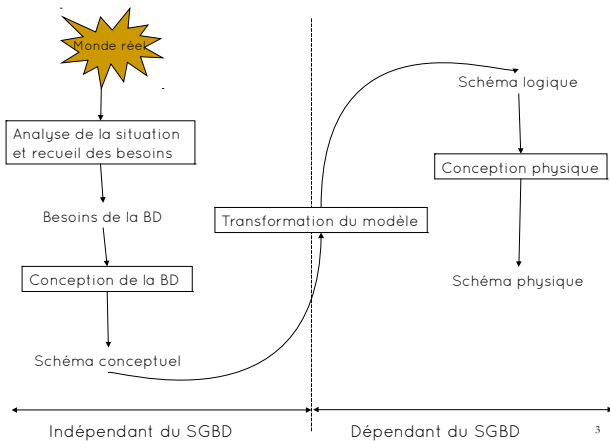
Bureau F011
Walter.Rudametkin@polytech-lille.fr
<https://rudametw.github.io/teaching>

Étapes de la conception d'une base de données

- **Analyse** de la situation existante et des besoins
- Création d'une série de **modèles conceptuels** pour représenter tous les aspects importants du problème
- **Traduction** des modèles conceptuels en modèle logique
- **Implémentation** d'une base de données dans un SGBD à partir du modèle logique

2

Conception d'une base de données



3

Étape d'analyse (1/2)

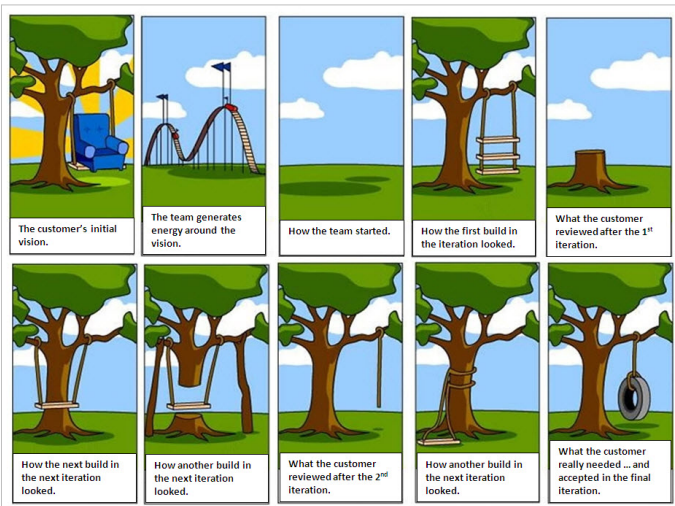
Recueil de documents

- Ex : cahier des charges, interviews, factures, listings...
- Ces documents peuvent déjà exister ou doivent être conçus lors de cette phase

De la qualité de cette étape dépend la pertinence de la base de données par rapport aux usages

- Plus d'un projet sur deux est un échec car les besoins des utilisateurs sont mal pris en compte

4



Étape d'analyse (2/2)

Extraire les informations élémentaires

Deux types d'information :

- Les données
- Les traitements

Les données manipulées sont relatives aux métiers

- Discuter avec les experts métiers permet de fixer la terminologie

La base de données concerne des utilisateurs cibles qui produiront et consommeront les données de la base

- Discuter avec les usagers permet de fixer les besoins réels

La base de données s'insère généralement parmi un ensemble d'autres logiciels informatiques

- Analyser les systèmes existants permet de préparer l'intégration de la base avec ces autres systèmes

6

Identification des informations (1/2)

Obtenue à partir des documents fournis, **une information est décrite** par :

- Un nom : (n client, n facture, nomClient, ...)
- Son **type**
 - numérique (entier, réel, date) ou alphanumérique
- Sa longueur (facultatif)
- Son origine (quel document(s) ?)
- Des observations :
 - Des **contraintes** d'intégrité :
 - exemples : salaire > smic, age > 0
 - Des **règles** de calcul :
 - exemple : nouvelle valeur du salaire > ancienne valeur

7

Identification des informations (2/2)

(suite) :

- **Sa nature** : donnée calculée (C) ou non calculée (NC)
 - ex : passer une commande :
 $\text{prix (NC)} * \text{quantité (NC)} = \text{prixCommande(C)}$
- remarque : certaines informations calculées nécessitent d'être conservées, on parle d'informations calculées et mémorisées (CM)

Vérification :

- **absence de polysémie** : deux données sont polysèmes si elles ont le même nom et expriment deux valeurs différentes.
 - exemple : date pour dateFacturation et pour dateCommande
- **absence de synonyme** : noms différents pour une même valeur.

8

Étape de conception de la BD (1/2)

Permet de traduire les besoins en un **modèle conceptuel**

- Passer du langage naturel à un **modèle formel** moins ambigu.
- Différentes méthodes existent : E-A, Merise...
- Représentation graphique simple

Abstraction des données et structuration en paquets homogènes

Quelques consignes :

- Se limiter au domaine d'étude
- Éviter au maximum les redondances
- Attention avec les attributs calculés

9

Étape de conception de la BD (2/2)

Note :

- Il n'existe **pas qu'une seule solution**
- Certains choix de conception doivent être faits, il faut en être conscients et les comprendre.
- Cette étape n'est pas systématique et dépend de l'expertise de l'ingénieur
- De cette étape dépend les suivantes
- Il existe quelques règles pour avoir une **bonne** BD (cf. normalisation)

10

Normalisation

Théorie destinée à concevoir correctement le schéma d'une base de données relationnelle

- **Sans redondance** d'information
- **Sans anomalie** en mise à jour

Repose sur la notion de dépendances fonctionnelles qui traduisent des contraintes sur les données.

11

UML : Langage de Modélisation

UML : **Unified Modeling Language** (<http://www.omg.org>)

UML est une notation graphique.

UML intègre l'objet et est donc plus adaptée aux SGBDOO

UML est adaptée à la programmation orientée-objet

Le diagramme de classe UML repose sur le modèle **entité-association**.

Quelques outils :

- Modelio, ArgoUML, Papyrus (gratuit, UML) : génération code Java
- Together (gratuit, commercial, UML) : génération code C++ et Java
- Rational Rose (commercial, UML) : tables, C++, Java, ...

12

Étapes de transformation et d'implémentation

- Existence d'algorithmes de traduction pour obtenir le modèle logique à partir du modèle conceptuel
- Existence d'instructions pour implémenter le modèle physique
- Étapes moins complexes que les précédentes car systématiques
- Certains logiciels permettent d'**automatiser** ces algorithmes.