

Parcours Data

Introduction



- "La donnée est partout"
- 99.9% des programmes sont des data pipes

Sommaire

- La donnée, c'est quoi ?
- Type de bases de données
- Identification et ownership
- Normalisation
- Déroulé de l'atelier

La donnée, c'est quoi ?



La langue

- Besoin de conserver et transmettre de l'**information**.
- Débute avec l'écriture.
- Besoin grandissant de complexité :
 - Complexification de la langue,
 - Création d'oeuvre graphique, sonores
- Introduction d'imprécision, subjectivité.

Fichier "plat"

- Fast-forward \pm 5000 ans
 - Transposition du document papier au numérique
 - Contenu non structuré
 - Pas de construction inter-documentaire



```
# Parcours Data  
## Introduction
```

```
# Data ?
```

```
## La langue
```

- * Besoin de conserver et transmettre de l'**information**.
 - * Débute avec l'écriture.
 - * Besoin grandissant de complexité :
 - * Complexification de la langue,
 - * Crédit d'oeuvre graphique, sonores
 - * Introduction d'imprécision, subjectivité.
- ![bg left:33%](intro/hieroglyphe.jpg)
-

```
## Fichier "plat"
```

- * Fast-forward ± 5000 ans
-

```
# Notions d'identité
```

```
# Ownership et contraintes
```

Fichier structuré

- Exemples: markdown, json, csv
- Introduction de structure technique
- Permet un premier niveau de traitement systématique :
 - Validation de forme
 - Transformations

Fichier structuré validant

- Exemple: XML
- Découle d'une volonté normative
- Traitement systématique avancés:
 - Validation technique et **sémantique**
 - Transformations
- Pas de construction inter-documentaire
"built-in"

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://maven.apache.org/POM/4.0.0"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://
<modelVersion>4.0.0</modelVersion>

<parent>
  <artifactId>ui</artifactId>
  <groupId>fr.epita.side</groupId>
  <version>1.0-SNAPSHOT</version>
</parent>

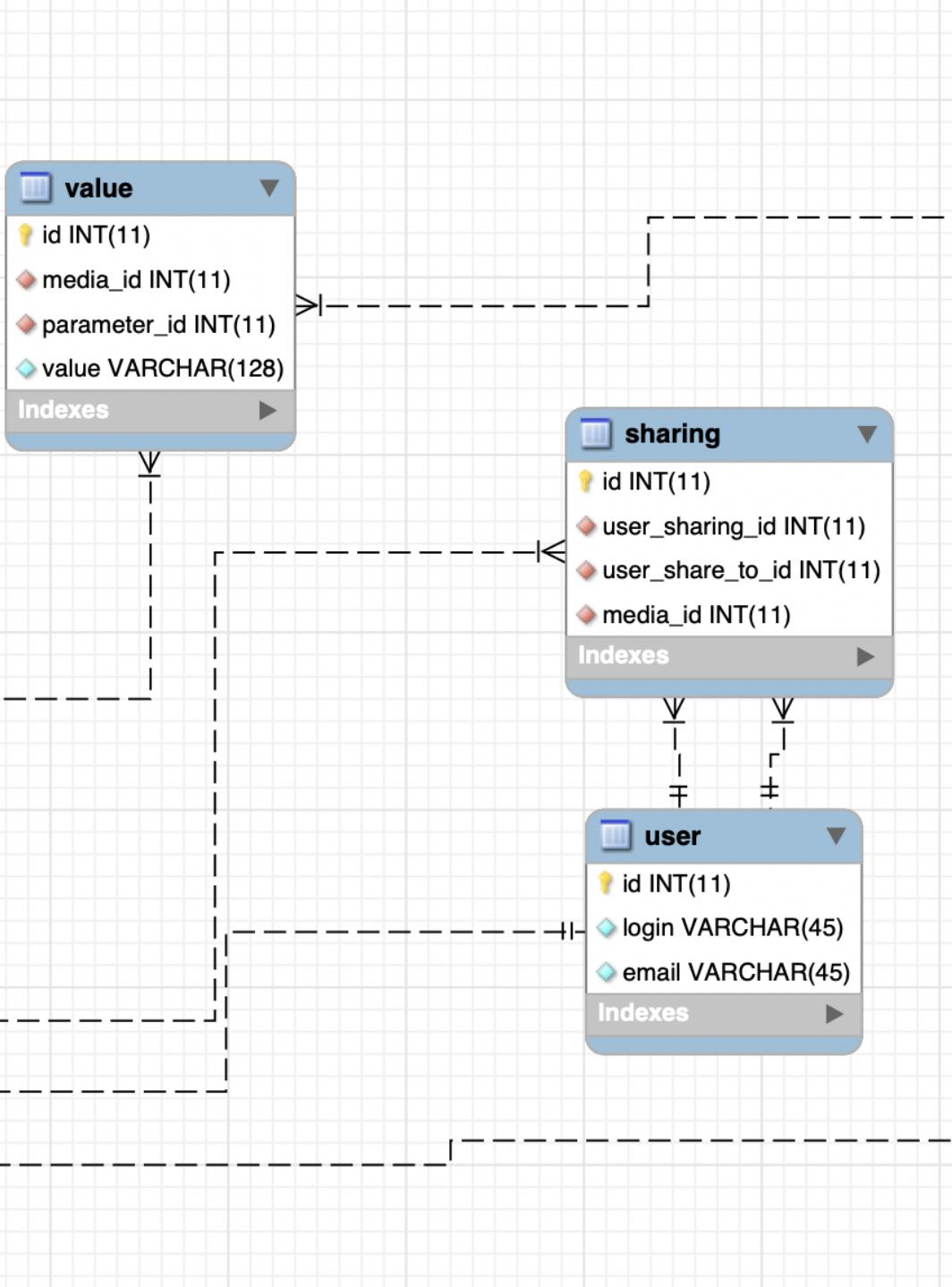
<artifactId>vaadin-runtime</artifactId>

<dependencies>
  <dependency>
    <groupId>fr.epita.side</groupId>
    <artifactId>framework-lang</artifactId>
  </dependency>

  <dependency>
    <groupId>io.quarkus</groupId>
    <artifactId>quarkus-core</artifactId>
  </dependency>

  <dependency>
    <groupId>io.quarkus</groupId>
    <artifactId>quarkus-undertow</artifactId>
  </dependency>

  <dependency>
    <groupId>com.vaadin</groupId>
    <artifactId>vaadin-core</artifactId>
  </dependency>
</dependencies>
```



Bases de données

- Application serveur dédiée au stockage et traitement des données
- Donnée: particules unitaire d'information
- Les **relations** entre les données créent **l'information**

La donnée c'est ...

- les "morceaux" d'information
- ± organisés
- ± atomiques
- interconnectés (idéalement).



10

Type de bases de données

```

FROM UserFriends AS UF
  JOIN UserGames AS UG
    ON UF.user1 = UG.[user] OR UF.u
      E user1 = 'Penny' OR user2 = 'Pe

```

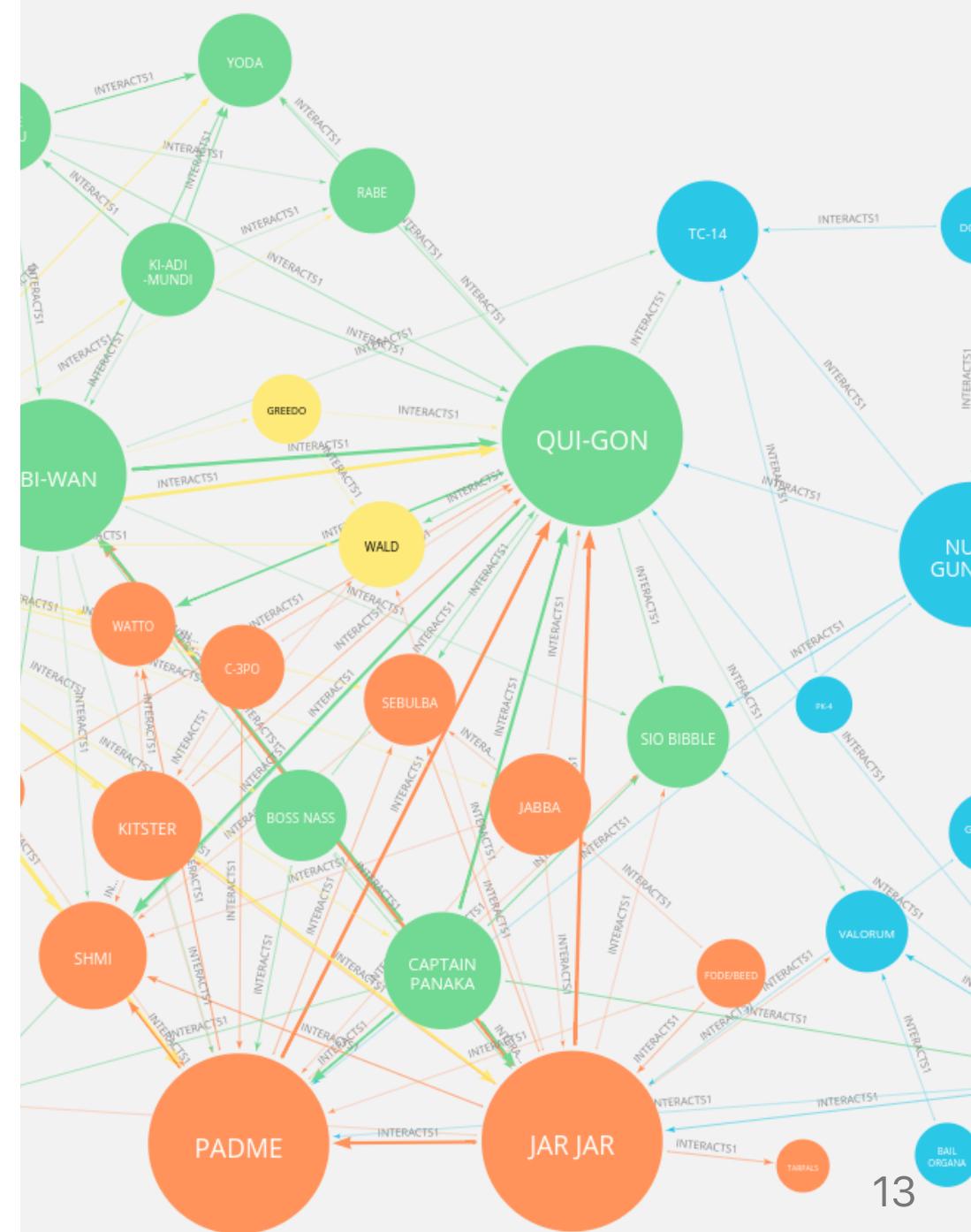
Différents types de BDD

- Relationnelle : fondée sur la théorie mathématique de l'algèbre relationnelle (Oracle ...)
- Hiérarchique : les données sont organisées en arborescence (ftp, gopher, ldap)
- Documentaire : l'unité de traitement est le document (Mongo DB)

Date	user	game	hoursPlayed	acquisitionDate
	Migue	Crashex Legends	0.00	2019-04-25 19:50:41.10
	Penny	Crashex Legends	0.00	2019-04-25 19:50:41.10
	Penny	Overwatch	0.00	2019-04-25 19:50:41.10
	Penny	Portal	0.00	2019-05-16 19:09:01.84
	Penny	Crashex Legends	0.00	2019-04-25 19:50:41.10
	Penny	Overwatch	0.00	2019-04-25 19:50:41.10
	Penny	Portal	0.00	2019-05-16 19:09:01.84
	Test	Crashex Legends	0.00	2019-04-25 19:51:36.30
	Test	Overwatch	0.00	2019-05-16 19:08:37.61
	Test	Portal 2	0.00	2019-04-25 19:50:41.10

Différents types de BDD

- Index documentaire : pour une recherche efficace (SolR, ElasticSearch)
 - Graph : s'intéresse aux liens autant qu'aux nodes eux même, le parcours est simplifié (Neo4J, Titan)
 - Et tous les autres (clé -> valeur, colonne...)
 - **Par défaut** : SGBD-R, pour Système de Gestion de Base de Données Relationnel (RDBMS).

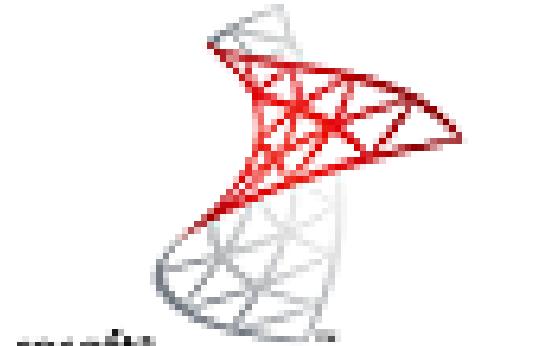


ACID: Caractéristiques d'un SGBDR valide

- A : Atomicity - une transaction est effective intégralement ou pas du tout.
- C : Consistency - le système ne passe que d'un état **valide** à un autre.
- I : Isolation - toute transaction s'exécute en **isolation** des autres.
- D : Durability - le système est résistant au pannes, globalement résilient.

Quelques SGBDR connus:

- Oracle
- SQL Server
- Postgres SQL
- MySQL / MariaDB



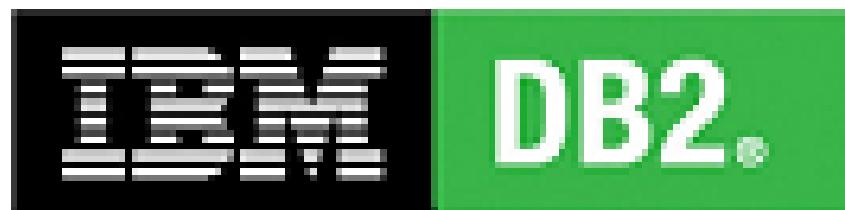
ORACLE



PostgreSQL



SYBASE



Dans un SGBDR

- les données sont organisées en tables ...
- ... composées de colonnes, fortement typées
- on peut créer des liens entre objets.

Identification et ownership

Problème d'identification de la donnée:

- Pour être accessible, une donnée doit être identifiable de manière unique,
- Comment choisir une propriété identifiant ?
 - Identifiant naturel ?
 - Identifiant synthétique ?
- Clé primaire

Problème d'identification de la donnée:

- Comment référencer une donnée externe ?
 - Clé étrangère
- Comment identifier une donnée qui exprime une **relation** entre N autres données ?
 - Clé composée

Identifiant et ownership

- Le problème des AUTO INCREMENT
- Celui qui définit la clé primaire est le propriétaire de la donnée
- La solution du UUID / GUID

Normalisation

Normalisation

- Formes normales
- Règles de conception de modèles relationnel
- Cherche à garantir la validité des données



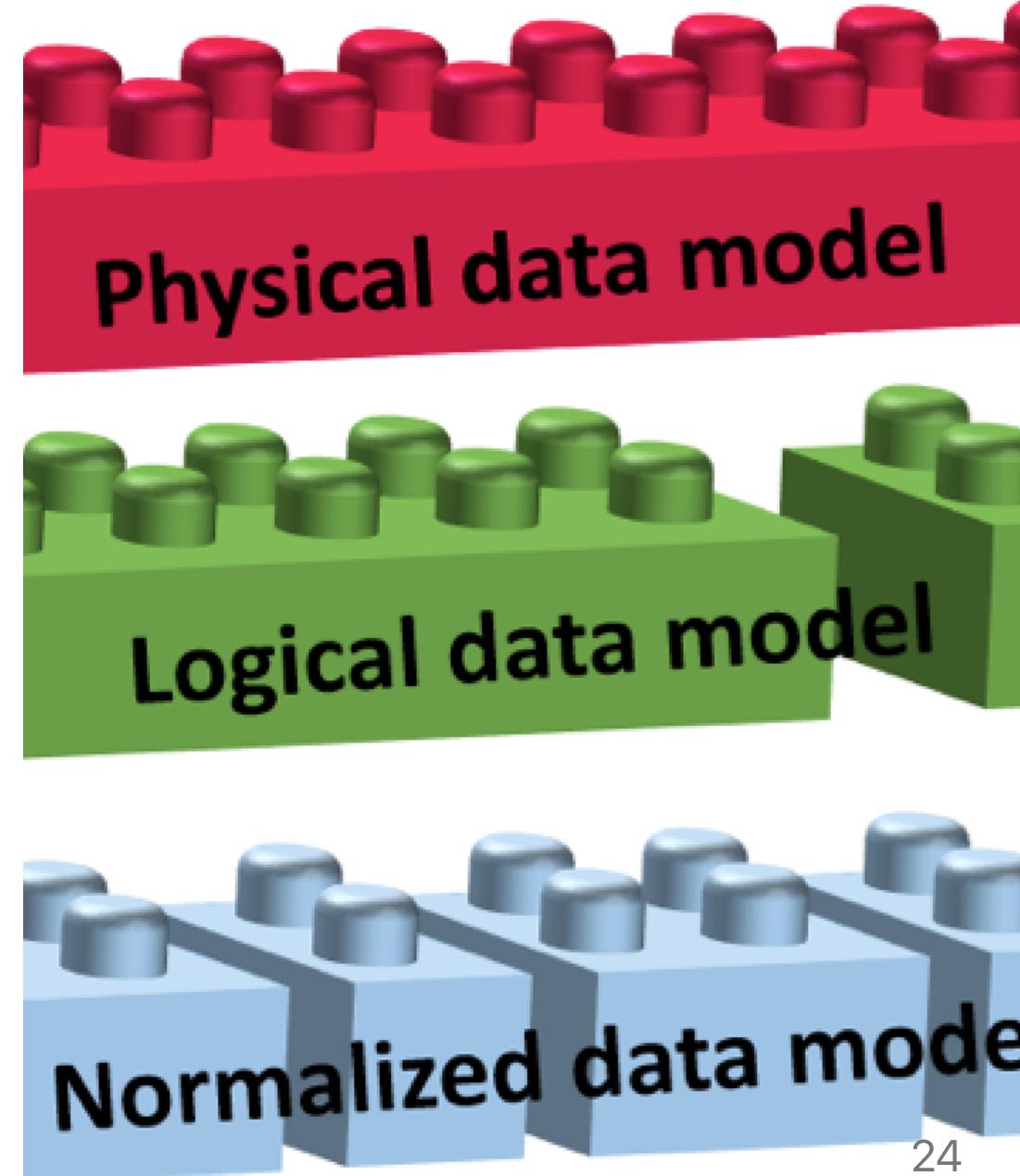
Normalisation

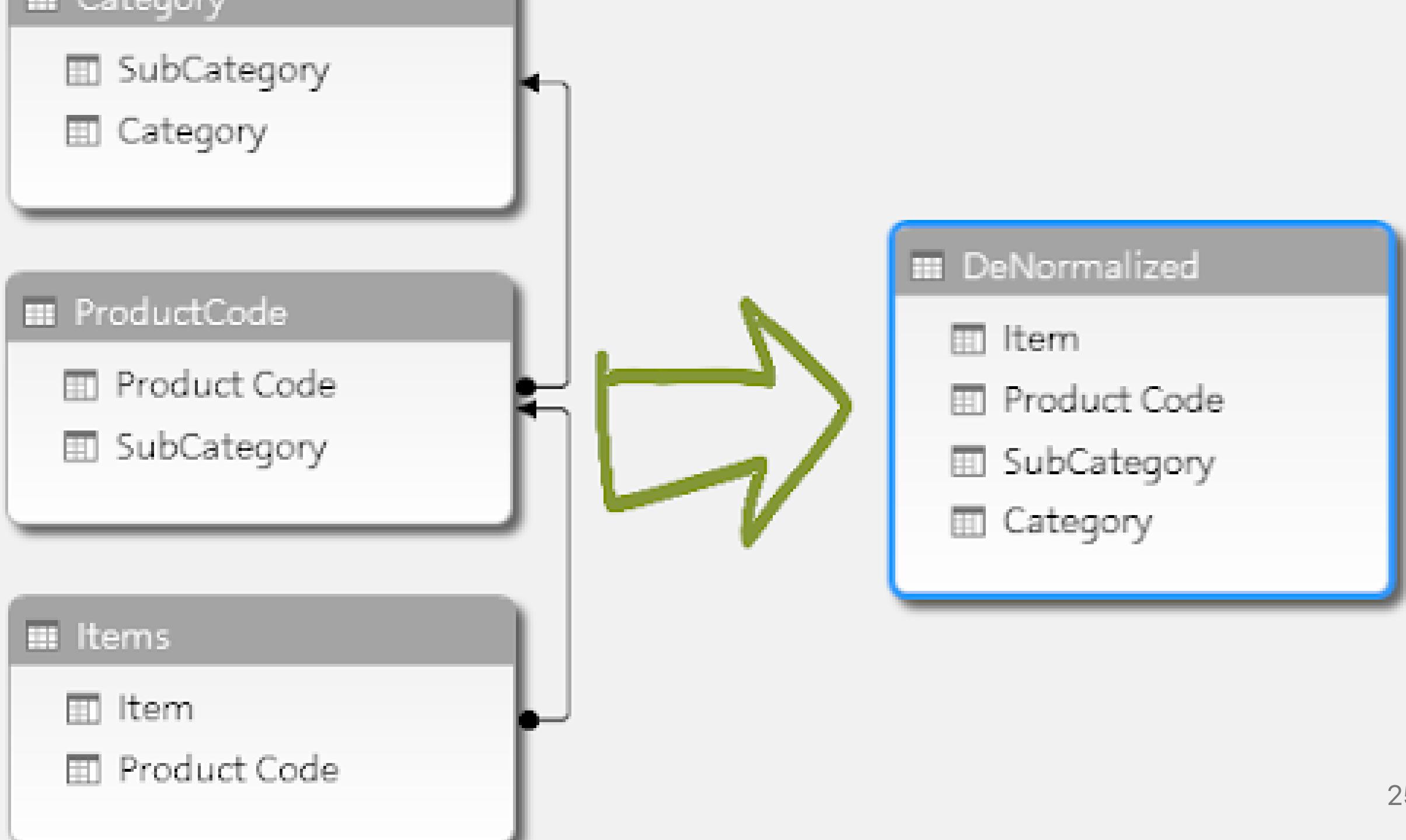
- Chaque type a son propre formalisme et son ensemble de règles et bonnes pratiques :
- Nommage, nature, type des données
- Articulation des données entre elles etc...



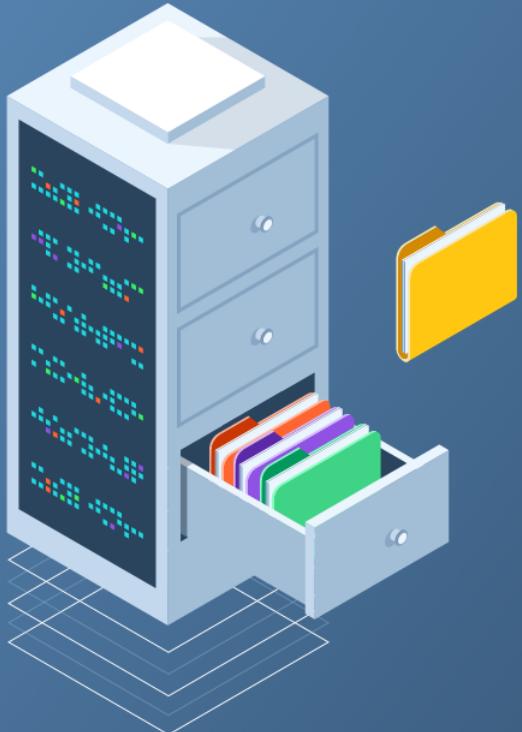
Dé-normaliser

- Parfois, l'emphase n'est pas mise (localement) sur la validité des données
- On peut alors dé-normaliser son modèle
- Trade-off coût / bénéfice





Déroulé de l'atelier



Objectifs :

- Premier contact avec les problématiques data
- Prise en main des modèles relationnels
- Pratique de SQL



D1

- 10:30 - 12:00 : Conférence d'introduction
- 14:30 - 17:30 : TD
- 17:30 - 22:00 : Permanences assistants



D2 - D4

- 10:00 - 13:00 : TD
- 14:30 - 17:30 : TD
- 17:30 - 22:00 : Permanences assistants



Validations

- Notions unitaires -> exercices en ligne
- Notions agrégées -> évaluation authentique (projet, fin S6)