

Les langages en informatique



Gestion de projets

Humanb∞ster
TRANSFORMEZ VOS CODES

Nathalia Pereira,
Emma Minary,
Joseph Le Piouffle

DWWM PE9 Villeurbanne

Sommaire

1. Les langages de programmation

- 1.1. Les paradigmes de programmation
- 1.2. Niveau de langage
- 1.3. Langages compilés et interprétés
- 1.4 **Langages impératifs**
 - 1.4.1 Orienté objet
 - 1.4.2 Procédural et structural
- 1.5. **Langages déclaratifs**
 - 1.5.1 Fonctionnel et logique

2. Les formats de données

2. 1. Qu'est ce qu'un format de données ?

- 2. 1. 1. Définition
- 2. 1. 2. L'importance des formats de données dans l'informatique

2. 2. Principaux formats de données

- 2. 2. 1. JSON (JavaScript Object Notation)
- 2. 2. 2. YAML (Ain't Markup Language)
- 2. 2. 3. XML (eXtensible Markup Language)
- 2. 2. 4. La représentation des données

3. Les langages de conception

3.1. UML définition et objectif

- 3.1.1. Types de diagrammes UML
 - 3.1.1.1. Diagrammes de classes
 - 3.1.1.2. Diagrammes de cas d'utilisation

3.2. MERISE définition et objectif

- 3.2.1. Les niveaux
 - 3.2.1. 1. MCD
 - 3.2.1. 2. MLD

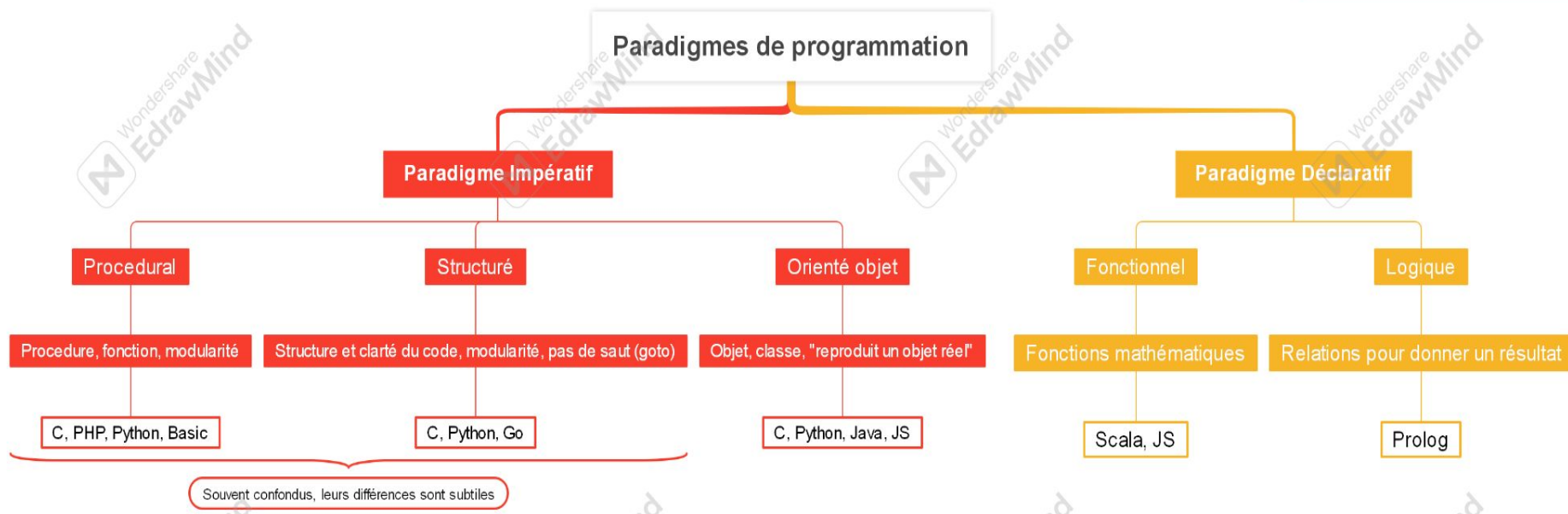
1.1 Les paradigmes de programmation

1/2

Une classification des langages de programmation	Selon leurs capacités. Ce qu'ils peuvent faire ou ne pas faire.
Dans le but de résoudre un problème. Dois-je découper en petit morceau et remonter ou faire l'inverse : Top bottom ou bottom down	Un langage peut appartenir à plusieurs paradigmes.

1.1 Les paradigmes de programmation

1/2



1.2. Niveau de langage

Low-level Languages

Machine Language

```
01010101 01001000
10001001 11100101
10001011 00010101
10110010 00000011
...
```

Assembly Language

```
pushq %rbp
movq %rsp, %rbp
movl alice(%rip), %edx
movl bob(%rip), %eax
imull %edx, %eax
movl %eax, carol(%rip)
...
```

High-level Languages

Compiled & Interpreted Languages (Python, C/C++, ..)

```
int alice = 123;
int bob = 456;
int carol;
main(void)
{
    carol = alice*bob;
}
```

Pseudocode

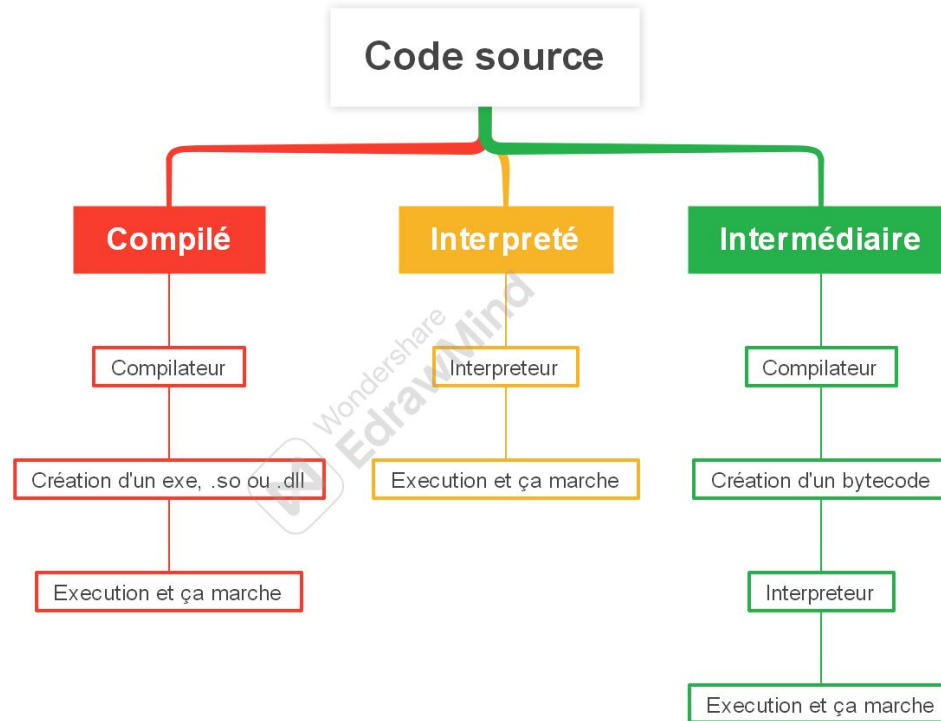
```
- Initialize alice to 123 and
  bob to 456
- Multiply alice and bob and
  store the result into carol
```

Natural Languages

English, Turkish, ...

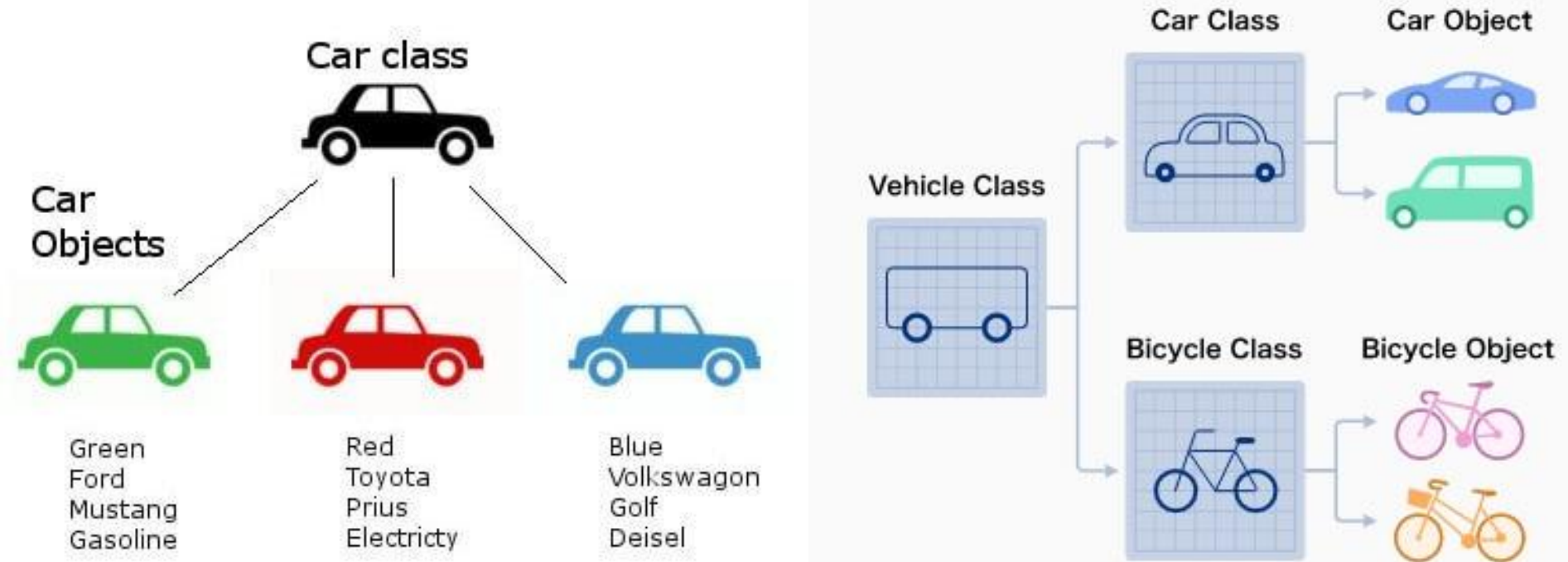
```
Given two variables called
alice and bob with initial
values 123 and 456,
respectively, multiply them
and store the result into
another variable called carol.
```

1.3 Langages compilés et interprétés



1.4 Langages impératifs

.1 Orienté objet



1.4 Langages impératifs

.2 Procédural et structurel

Procédural	Structurel
Plus facile à compiler ou interpréter Modulaire Moins gourmand en mémoire	Plus simple à concevoir et reprendre Modulaire Pas de Goto (saut d'une ligne à une autre ou d'un bloc à un autre) -> Plus fiable

1.5 Déclaratifs

.1 Fonctionnel et logique

Fonctionnel	Logique
Evite les Goto Travaille avec des fonctions en arguments et en valeur Les données traitées ne peuvent être modifiées (pas d'effet secondaire) Rapide et fiable	Fait des relations plutôt que d'exécuter Donne des résultats demandés et parfois non demandés Plus dur à apprendre

2. Les formats de données

2. 1 Qu'est ce qu'un format de données ?

2. 1. 1 Définition

Les formats de données, souvent appelés **formats de sérialisation**, définissent la manière dont **l'information est structurée et organisée**. C'est comme une règle ou un modèle qui va définir de quelle façon les données doivent être présentées, et qui va permettre à l'ordinateur de comprendre comment les lire. C'est une sorte de « mode d'emploi ».

Il existe de nombreux formats de données, chacun ayant ses caractéristiques propres pour répondre à des besoins spécifiques. Le choix dépend des exigences spécifiques des données et des systèmes avec lesquels on travaille.

2. 1 Qu'est ce qu'un format de données ?

2. 1. 2 L'importance des formats de données dans l'informatique

- Cohérence des informations car il y a une structure claire et prévisible
- Simplification car il y a une standardisation des formats des données
- Communication puisqu'il y a une compréhension commune
- Traitement efficace des informations

En résumé, une structure claire et prévisible, associée à une standardisation, simplifie la communication entre logiciels, facilite le traitement des informations de manière efficace, et contribue au bon fonctionnement des systèmes informatiques.

2. 2. Les principaux formats de données

Les plus couramment utilisés et populaires sont :

2. 2. 1. JSON (JavaScript Object Notation)

2. 2. 2. YAML (Ain't Markup Language)

2. 2. 3. XML (eXtensible Markup Language)

2. 2. Les principaux formats de données

2. 2. 1. JSON (JavaScript Object Notation)

2. 2. 1. 1. Définition

Le format JSON est un format standard utilisé pour représenter des données structurées de façon semblable aux objets Javascript.

C'est un langage qui permet de décrire des données d'une manière facile à lire et à écrire, et qui privilégie la lisibilité pour les humains.

Il utilise des paires **clé-valeur** pour représenter les données, où une **clé** est une chaîne de caractères et la **valeur** peut être un nombre, une chaîne, un booléen, un tableau, un objet ou la valeur nulle.

Sa syntaxe s'appuie sur des **accolades**.

2. 2. Les principaux formats de données

2. 2. 1. JSON (JavaScript Object Notation)

2. 2. 1. 2. Exemple

- La structure générale est définie avec des accolades {} pour l'objet principal
- Les paires **clé-valeur** sont utilisées pour représenter les différents aspects de l'entreprise, comme ici, son adresse

```
{  
  "entreprise": {  
    "nom": "Tech Solutions",  
    "secteur": "Technologie",  
    "adresse": {  
      "ville": "VilleTech",  
      "rue": "Rue de l'Innovation",  
      "code_postal": "12345"  
    },  
  },  
}
```

2. 2. Les principaux formats de données

2. 2. 2. YAML (Ain't Markup Language)

2. 2. 2. 1. Définition

Tout comme le langage Json, c'est un langage qui permet de décrire des données d'une manière facile à lire et à écrire, qui privilégie la lisibilité pour les humains. Il est simple d'utilisation.

C'est aussi un langage qui utilise des paires **clé-valeur** pour représenter les données. Il prend en charge plusieurs types de données, notamment les chaînes de caractères, les nombres, les booléens, les listes, les structures imbriquées et les valeurs nulles.

Sa syntaxe s'appuie sur l'**indentation**.

Il est souvent utilisé dans le développement logiciel, la configuration de serveurs, et d'autres domaines où la lisibilité humaine et la simplicité sont importantes.

2. 2. Les principaux formats de données

2. 2. 2. YAML (Ain't Markup Language)

2. 2. 2. 2. Exemple

- Les **clés** et les **valeurs** doivent être séparées par des deux-points (:)

```
entreprise:  
  nom: "Tech Solutions"  
  secteur: "Technologie"  
  adresse:  
    ville: "VilleTech"  
    rue: "Rue de l'Innovation"  
    code_postal: "12345"
```

2. 2. Les principaux formats de données

2. 2. 3. XML (eXtensible Markup Language)

2. 2. 3. 1. Définition

Ce langage utilise des **balises** pour délimiter et décrire les éléments d'information.

Chaque balise a un nom qui décrit ce qu'elle contient. Il y a toujours une balise ouvrante et une balise fermante.

XML est souvent utilisé pour échanger des données entre différents systèmes informatiques de manière standardisée et extensible.

2. 2. Les principaux formats de données

2. 2. 3. XML (eXtensible Markup Language)

2. 2. 3. 2. Exemple

- Les balises peuvent être imbriquées pour représenter la structure hiérarchique des données
- Les balises autorisent une structuration arborescente des données

```
<entreprise>  
  <nom>Tech Solutions</nom>  
  <secteur>Technologie</secteur>  
  <adresse>  
    <ville>VilleTech</ville>  
    <rue>Rue de l'Innovation</rue>  
    <code_postal>12345</code_postal>  
  </adresse>  
</entreprise>
```

2. 2. Les principaux formats de données

2. 2. 4. La représentation des données

	XML	JSON	YAML
	Langage de balisage	Formats de données	Formats de données
Définition des éléments	Balises	paires clé/valeur	paires clé/valeur
Styles d'indentation	Structuration arborescente des données	tabulations	trait d'union (-) suivi d'espaces

3. Les langages de conception

3. 1. UML (*Unified Modeling Language*, ou Langage de Modélisation Unifié)

Le langage UML a été pensé pour être un langage de modélisation visuelle commun, et riche sémantiquement et syntaxiquement. Il est destiné à l'architecture, la conception et la mise en œuvre de systèmes logiciels complexes par leur structure aussi bien que leur comportement.

L'UML a une relation directe avec l'analyse et la conception orientées objet.

Son objectif : créer un langage visuel commun dans le monde complexe du développement de logiciels, un langage qui serait également compris par les utilisateurs professionnels et tous ceux qui veulent comprendre un système.

3. 1.1. Types de diagrammes UML

- Diagrammes UML structurels

- **Diagramme de classes**
- Diagramme de composants
- Diagramme de structure composite
- Diagramme de déploiement
- Diagramme d'objets
- Diagramme de paquets

3. 1.1. Types de diagrammes UML

- Diagrammes UML comportementaux

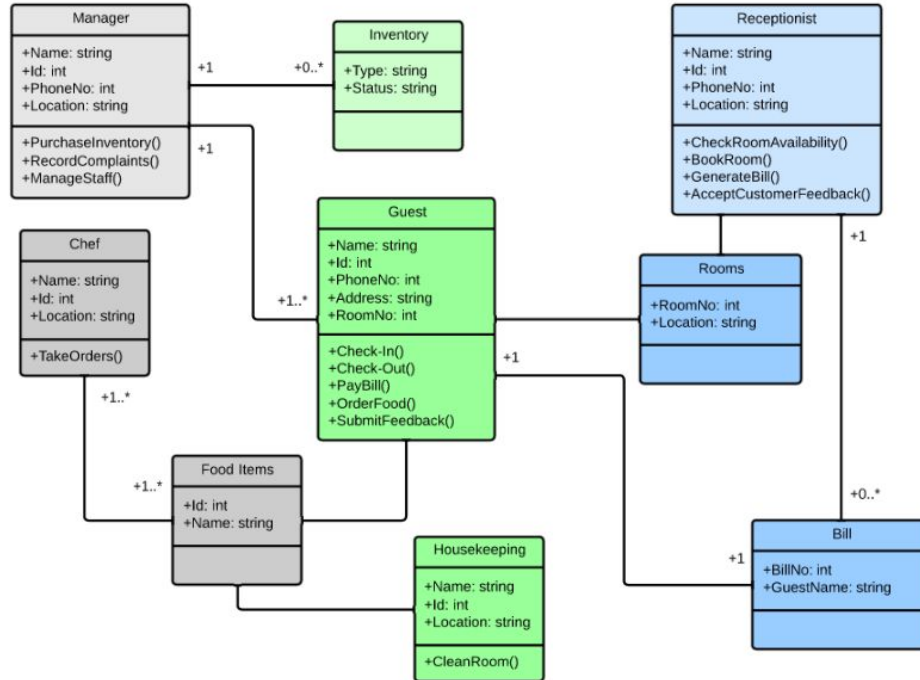
- Diagrammes d'activités
- Diagramme de communication
- Diagramme global d'interaction
- Diagramme de séquence
- Diagramme états-transitions
- Diagramme de temps
- **Diagramme de cas d'utilisation**

3. 1.1.1. Diagrammes de classes

- **Diagramme de classes :**
Les diagrammes de classes sont l'un des types de diagrammes UML les plus utiles, car ils décrivent clairement la structure d'un système particulier en modélisant ses classes, ses attributs, ses opérations et les relations entre ses objets.

3. Les langages de conception

1/5



Exemple:

Diagramme de classes d'un système de gestion hôtelière

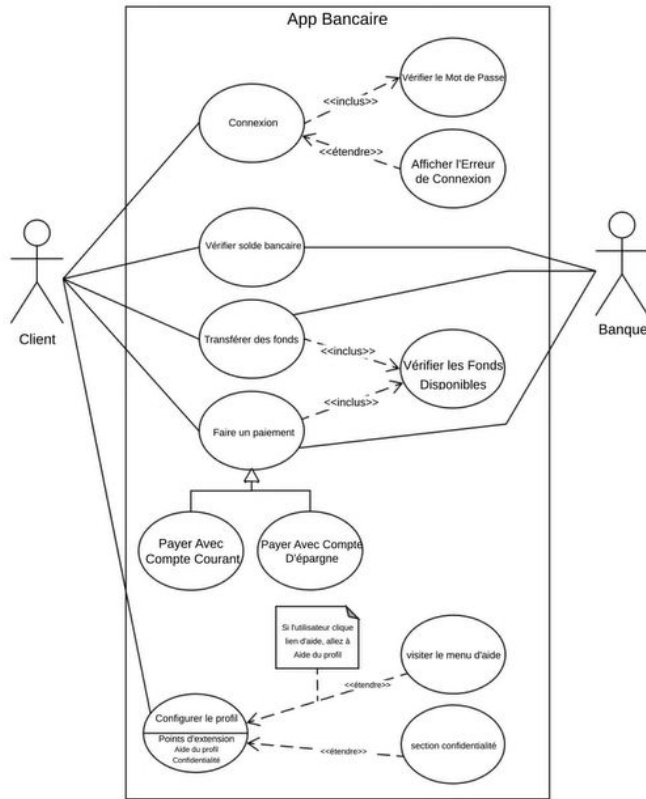
Un diagramme de classes peut montrer les relations entre chaque objet dans un système de gestion hôtelière, y compris les informations des clients, les tâches du personnel et l'occupation des chambres.

3. 1.1.2. Diagrammes de cas d'utilisation

- **Diagramme de cas d'utilisation :**
Il représente une fonctionnalité spécifique dans un système et est créé pour illustrer comment différentes fonctionnalités sont interconnectées et montrer leurs contrôleurs (ou acteurs) internes et externes.

3. Les langages de conception

1/6



Exemple:

Application bancaire simple

Interaction client et banque avec l'application bancaire sur la page de connexion.

3. 2. **MERISE** (Méthode d'Étude et de Réalisation Informatique des Systèmes d'Entreprise)

Superficiellement, s'agit d'une méthode qui définit un certain nombre de schémas et diagrammes.

La méthode Merise se caractérise par :

- une approche systémique en ayant une vue de l'entreprise en termes de systèmes
- une séparation des données (le côté statique) et des traitements (le côté dynamique)
- **une** **approche** **par** **niveaux**

3. 2.1. Les niveaux

> Niveau conceptuel

- **MCD (Modèle Conceptuel de Données)**
- MCT (Modèle Conceptuel des Traitements)

3. 2.1. Les niveaux

> Niveau logique ou d'organisation

- **MLD (Modèle Logique des Données)**
- MLT (Modèle Logique des Traitements)

3. 2.1. Les niveaux

> Niveau physique

- MPD ou MPhD (Modèle Physique des Données)
- MOT ou MOpT (Modèle Opérationnel des Traitements)

3. 2.1.1. MCD

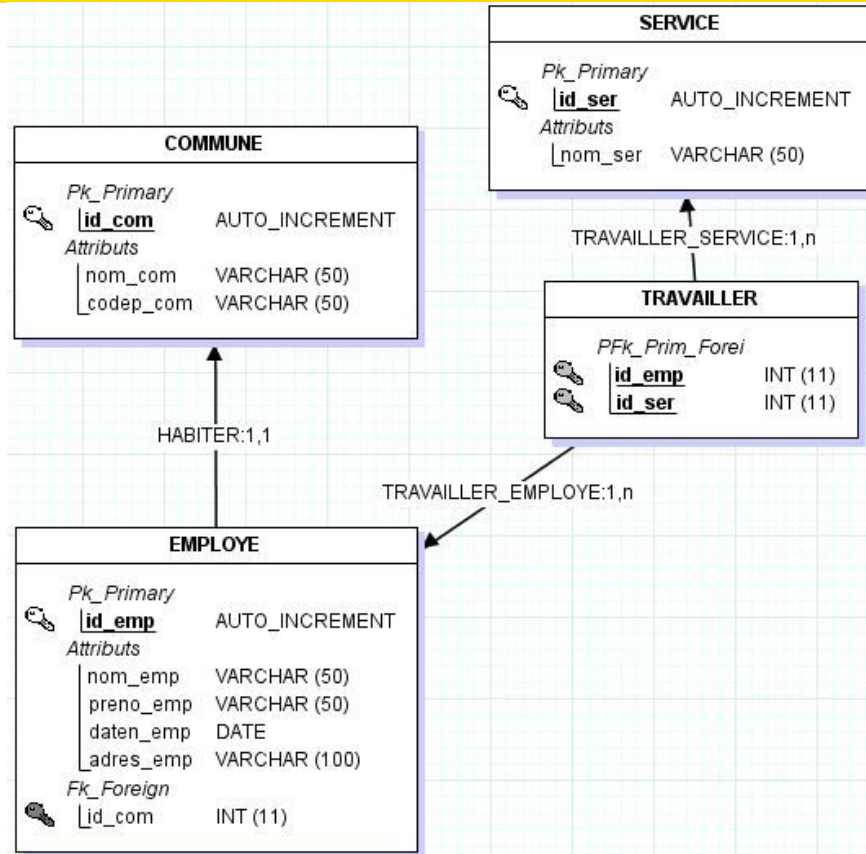
MCD (Modèle Conceptuel de Données)

Le modèle conceptuel des données (MCD) a pour but d'écrire de façon formelle les données qui seront utilisées par le système d'information. Il s'agit donc d'une représentation des données, facilement compréhensible, permettant de décrire le système d'information à l'aide d'entités.

Le MCD possède une syntaxe et un vocabulaire qui lui est propre. On parle d'entités et d'associations. C'est pourquoi le MCD est parfois appelé modèle entités-associations.

3. Les langages de conception

2/6



Représentation du schéma MCD et MPD

3. 2.1.2. MLD

MLD (Modèle Logique des Données)

Est une étape de la méthodologie Merise. C'est une des dernières étapes proposée et elle permet d'implémenter la base de données en transcrivant le MCD/MPD en instructions SQL adaptées au SGBDR prévu. Concrètement, le MLD permet de connaître le nombre de tables ainsi que leurs contraintes (liaisons entre tables) à mettre en œuvre dans une base de données relationnelle.

On représente ainsi les données issues de la modélisation Merise sous la forme suivante :

- Chaque ligne représente une table ;
- C'est toujours le nom de la table qui est écrit en premier ;
- Les champs sont listés entre parenthèses et séparés par des virgules ;
- Les clés primaires sont soulignées et placées au début de la liste des champs ;
- Les clés étrangères sont préfixées par un dièse.

3. Les langages de conception

2/8

```
chargedappui (char_id, char_email, char_prenomnom, char_mdp, char_droits, char_actif, #structureappui_stra_id)
structureappui (stra_id, stra_nom, stra_actif, #vill_id)
champperso (cham_id, cham_nom, cham_type, cham_description, #stra_id)
fiche (fich_id, fich_date, fich_courte, fich_pointdappui, fich_tempspasse, fich_comp_besoinapporte_autre, fich_comp_typereponse_autre, fich_comp_competence_autre, fich_comp_orientation_serviceinterne, fich_comp_or
contact (cont_id, cont_prenom, cont_nom, cont_tel, cont_email, cont_sexe, cont_dejasuiviinformation, cont_orientationinformation)
structure (stru_id, stru_nom, stru_tel, stru_email, stru_adresse, stru_siret, stru_nombremembres, stru_organismeemployeur, stru_nombresalaries, stru_nombreetp, stru_situationprojet, stru_situationprojet_autre, str
cadreducontact (pcad_id, pcad_intitule, pcad_actif)
besoinapporte (pbes_id, pbes_fiche_sommaire, pbes_intitule, pbes_actif, pbes_explication)
typereponse (ptyp_id, ptyp_intitule, ptyp_actif)
competenceaccompagnement (pcom_id, pcom_intitule, pcom_actif)
orientation (pori_id, pori_intitule, pori_actif)
secteuractivite (psec_id, psec_intitule, psec_actif, psec_explication)
echelledintervention (pech_id, pech_intitule, pech_actif)
```

MLD représenté de manière
pratique

Sources

Langages de programmations

beefington, u/. (n.d.). Eli5:what is the difference between "compiled vs interpreted ... - reddit. Reddit.
https://www.reddit.com/r/explainlikeimfive/comments/3tnrrr/eli5what_is_the_difference_between_compiled_vs/

pankaj, pp_. (2022, June 28). Differences between procedural and Object Oriented Programming. GeeksforGeeks.
<https://www.geeksforgeeks.org/differences-between-procedural-and-object-oriented-programming/>

coll, anon. (n.d.). 2. A broad look at programming and programming languages¶ colab open the notebook in Colab. 2. A Broad Look at Programming and Programming Languages - Programming with Python for Engineers 1.0 documentation. https://pp4e-book.github.io/chapters/ch2_programming.html

Aryan, A. (2019, September 30). Functional programming in JavaScript: Toptal®. Toptal Engineering Blog. <https://www.toptal.com/javascript/functional-programming-javascript>

Sources

Les formats de données

Format de données ; Wikipedia ; https://fr.wikipedia.org/wiki/Format_de_donn%C3%A9es

OpenAI. (2023). ChatGPT. Formats de données en informatique définition ; définition YAML ; définition JSON ; définition XML ; Différences entre YAML JSON et XML ; Quand est-ce qu'on utilise XML ; Quand est-ce qu'on utilise JSON ; Quand est-ce qu'on utilise YAML. <https://chat.openai.com/>

Manipuler des données JSON ; mdn web docs ;
<https://developer.mozilla.org/fr/docs/Learn/JavaScript/Objects/JSON>

Techno-science.net ; <https://www.techno-science.net/definition/658.html>

Romain Frutos, (8 décembre 2023), Un guide complet sur les formats de données structurées : JSON, XML, YAML et CSV ; <https://talks.freelancerepublik.com/donnes-structurees-json-xml-csv-yaml/>

CANTALOUBE J. (2018) , Eduscol ; TP1 : CSV, XML, JSON;
<https://eduscol.education.fr/sti/sites/eduscol.education.fr.sti/files/ressources/pedagogiques/9734/9734-1-les-format-s-csv-xml-json.pdf>

Sources

Les langages de conception

-UML

Outils UML : Lucidchart <https://www.lucidchart.com/>

Sources :

Lucidchart: <https://www.lucidchart.com/pages/fr/langage-uml>

Louis Vandeveld

<https://louisvandeveld.be/index.php?dos=my&fic=meris#:~:text=Le%20MCD%20permet%20de%20repr%C3%A9senter,l'information%20statique%20et%20durable.>

-Merise

Outil MCD : Draw Io: <https://www.drawio.com/>

Sources :

Wikipedia: [https://fr.wikipedia.org/wiki/Merise_\(informatique\)](https://fr.wikipedia.org/wiki/Merise_(informatique))

Editions ENI:

<https://www.editions-eni.fr/livre/merise-guide-pratique-3e-edition-modelisation-des-donnees-et-des-traitements-manipulations-avec-le-langage-sql-9782409015342/presentation-de-la-methode-merise>

Prospection Ciblee: <https://www.prospection-ciblee.com/mcd-modele-conceptuel-de-donnees-base-de-donnees/>

Base de Donnees.com: https://www.base-de-donnees.com/mld/?utm_content=cmp-true