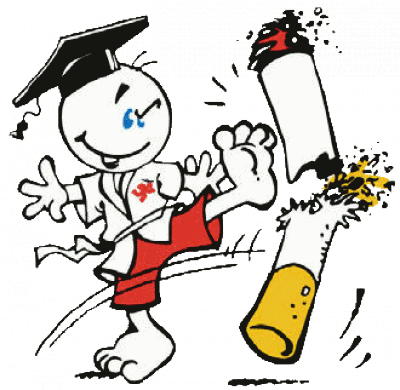
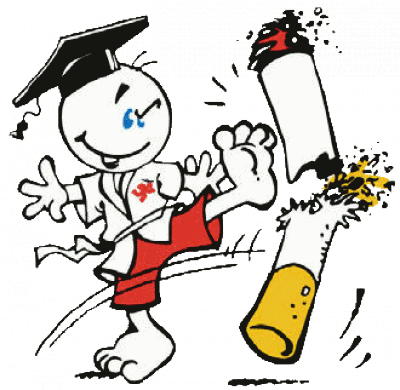
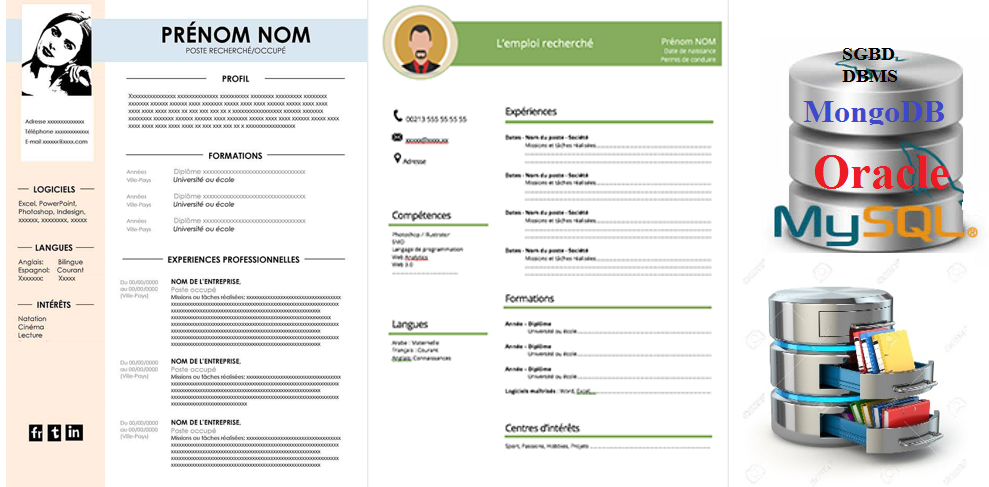
|  |  |
| --- | --- |
| **Université Sorbonne Paris Nord (USPN)** | Année Universitaire |
| Institut Galilée (**IG**) | 2020 / 2021 |
| Laboratoire d’Informatique de Paris Nord (**LIPN**) | ***La Maison des Sciences Numérique (La MSN)*** |
| ***La Data e(s)t le monde de demain*** | 🚹🚹🚹**M. F. Boufarès** |
| faouzi.boufares@sorbonne-paris-nord.fr | [boufares@lipn.univ-paris13.fr](mailto:boufares@lipn.univ-paris13.fr) |
| **La Data/La Donnée** | ***Bases de Données Avancées - Entrepôts de Données*** |





**Projet Annuel : DES BASES AUX ENTREPÔTS DE DONNÉES,**

**Données structurées ou NON structurées (Oracle, MySQL ou MongoDB ?)**

***From PDF Files to SQL or NoSQL Data Bases***

Exploration des Curriculum-Vitae (CV) des étudiant.e.s

qui candidatent pour le Maser 2 Informatique Exploration Informatique des Données et Décisionnel.

Réalisé le 17 décembre 2020 (Partie 1) par l’équipe :

|  |  |  |  |
| --- | --- | --- | --- |
| ***Groupe*** | ***Binôme*** | ***NOM*** | ***Prénom*** |
| G1 | B18 | BERHILI | Faïza |
| BOUSSAC | Emiliano |
| B19 | ALSADIK | Ammar |
| CHAMBRIER | Julian |
| ***Encadrant (Enseignant chercheur) Monsieur F.B BOUFARES*** | | | |



Table des matières

[1. Introduction 3](#_Toc58415868)

[2. Schémas de Base de Données : Structures de données & Méta-Données 4](#_Toc58415869)

[***2.1.*** ***Le schéma conceptuel de données (EER)*** 5](#_Toc58415870)

[***2.2.*** ***Le schéma conceptuel de données (UML)*** 6](#_Toc58415871)

[***2.3.*** ***Le schéma logique de données (Schéma relationnel)*** 7](#_Toc58415872)

[***2.4.*** ***Le dictionnaire de données & les métadonnées*** 8](#_Toc58415873)

[***2.5.*** ***Expression des contraintes sur les données*** 9](#_Toc58415874)

[3. Exemple de contenu de la base de données (instance de BD) 10](#_Toc58415875)

[***3.1. Exemple d’instances de la BD des CV*** 11](#_Toc58415876)

[***3.2. Création des structures de données (SQL-ORACLE)*** 12](#_Toc58415877)

[***3.2.Bis. Extraction automatique des données*** 12](#_Toc58415877)

[***3.3. Insertion des données dans une table et qualité des données (SQL/PLSQL-ORACLE)*** 13](#_Toc58415878)

[***3.4. Interrogations & Manipulations (SQL/PLSQL-ORACLE)*** 14](#_Toc58415879)

[4. Sujet du Projet Annuel 15](#_Toc58415880)

[***4.1. : Conception & Implantation de Bases de Données*** 15](#_Toc58415881)

[***4.2. : Conception & Implantation d’Entrepôts de Données*** 15](#_Toc58415882)

[5. Conclusion 16](#_Toc58415883)

# Introduction

Le système d’information de l’Université Sorbonne Paris Nord(USPN) est articulé autour de la base de données de nom BD-USPN. Elle contient les données relatives aux : (i) Ressources Humaines (Tels que les Enseignant.e.s, les Chercheur.se.s , les Ingénieur.e.s, les Administratifs) (ii) aux Etablissements (Tels que les Universités, les Ecoles, les Instituts, Hôpitaux, Entreprises, Organismes, Editeurs) partenaires, (iii) Eudiant.e.s (Tels que les anciens, les actuels, les candidats nationaux et internationaux retenus et non retenus). Une partie de cette base de données fera l’objet de notre étude.

En effet, dans le cadre de notre projet annuel intitulé «  **DES BASES AUX ENTREPÔTS DE DONNÉES, données structurées ou NON structurées (Oracle, MySQL ou MongoDB ?) ; *From PDF Files to SQL or NoSQL Data Bases*** ».

Il s’agit de la construction de données structurées à partir de données NON structurées (à partir de documents papiers ou numériques qui correspondent aux Curriculum-Vitae(CV) des étudiant.e.s qui candidatent pour le Maser 2 Informatique Exploration Informatique des Données et Décisionnel. Cette BD devra contenir les candidatures sur plusieurs années afin d’étudier le profil des candidat.e.s en entrée et évidemment ceux en sortie c’est-à-dire les diplômé.e.s.

Elle est décrite par le schéma conceptuel ainsi que le schéma relationnel (les tables) ci-après.

Ce document se présente comme suit. La premier partie traite de la construction des l'entrepôt de données, avec la conception et la création des tables nécessaire au bon fonctionnement de notre entrepôt. La seconde partie traite des insertions dans ces tables. Nous verrons que ces insertions ce font après la vérification préalable de contraintes stockées dans des méta-tables. La dernière partie correspond à la validation de notre entrepôt de données grâce à des requêtes. Ces requêtes permettent aussi d'observer les tendances, les habitudes, les profils qui se dégagent pour l'acceptation ou le refus des candidats dans le Master 2 - Exploration Informatique des Données et Décisionnel.

La chose primordiale pour notre équipe dans ce projet, était de construire un algorithme automatique avec de bon résultats, sans réelle interaction humaine afin d’extraire les informations des CV et les insérer dans des tables.

On ne voulait pas simplement faire des Insert Into et Create table. On voulait vraiment quelque chose digne d’un projet et tout ça de manière entièrement automatisé. On va donc vous présenter nos concepts.

# Schémas de Base de Données : Structures de données &Méta-Données

Le schéma conceptuel de données (SCD) ou le modèle conceptuel de données (MCD) peut être décrit selon plusieurs formalismes.

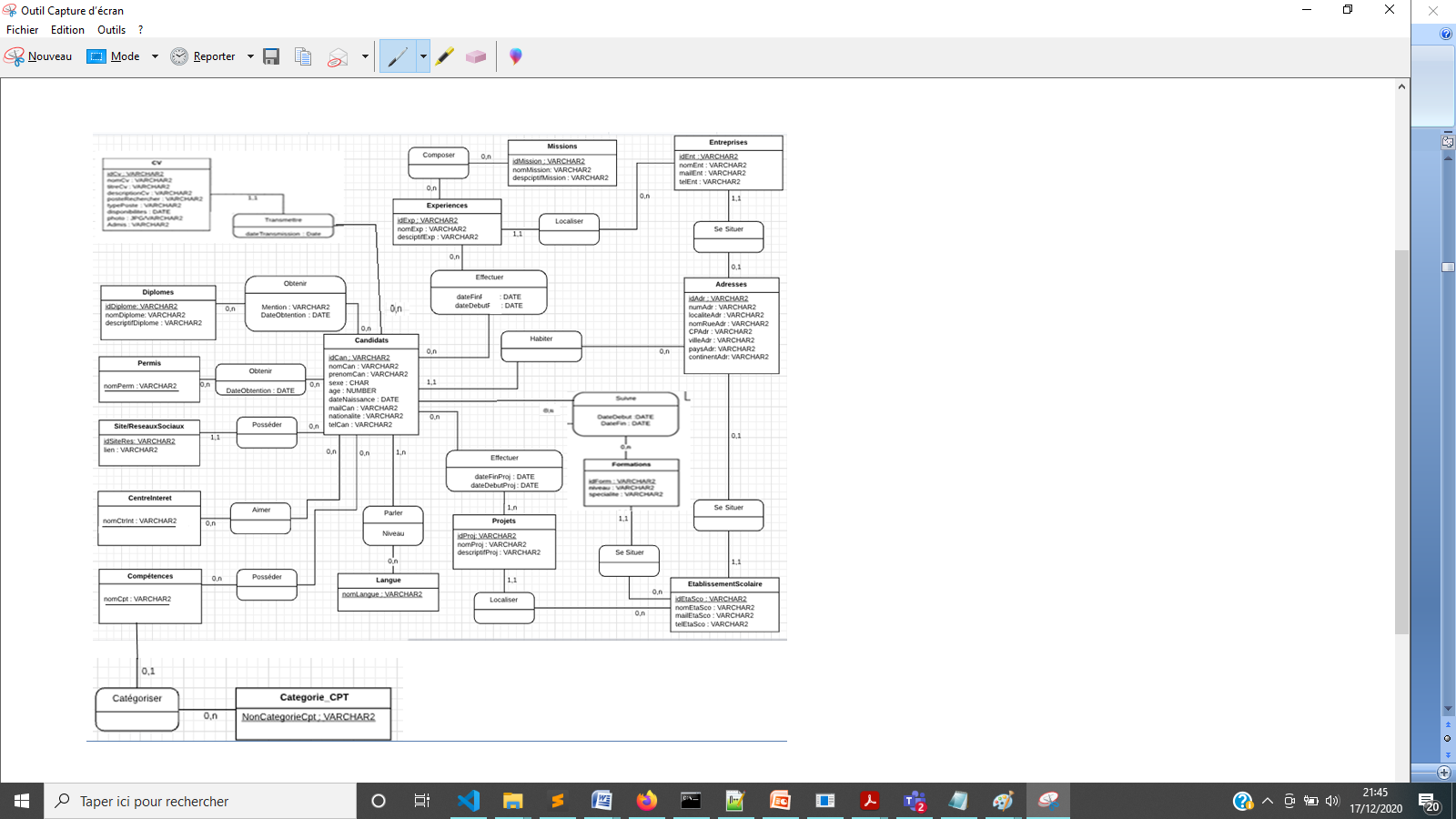
Dans le formalisme « Entity-Relationship (ER) ou Extended Entity-Relationship (EER), appelé aussi EA », la description des données comporte plusieurs entités (ou type d’entité = les rectangles) reliées entre elle par des associations (ou type d’association = les ovales), les arcs sont étiquetés par des contraintes de cardinalité (Exemple 1,1 ; 1,n et 0,n). La figure 1 représente le schéma conceptuel de la Base de Données, selon le formalisme Entité/Association (EA).

Dans le formalisme « Unified Modeling Language : UML », La description des données comporte plusieurs classes reliées entre elles par des arcs étiquetés par des contraintes de multiplicité (Exemple 1 ; 1,\* et \*) :

La structure (le schéma) de la BD est donnée ci-dessous.

## Le schéma conceptuel de données (EER)

Le **schéma conceptuel de données** (SCD) ou **le modèle conceptuel de données** (MCD) est décrit ci-dessous selon formalisme « Entity-Relationship(ER, EA) ou Extended Entity-Relationship(EER) ».



**Entités :**

Les attributs de l'entité *CANDIDATS* sont : **IdCan**, NomCan, PrenomCan, Sexe, Age, DateNaissance, MailCan, Nationalite, TelCan.

Les attributs de l'entité *CV* sont : **IdCv**, NomCv, TitreCv, DescriptionCv, PosteRechercher, TypePoste, Disponibilites, Photo, Admis.

Les attributs de l'entité *ADRESSES* sont : **IdAdr**, NumAdr, LocaliteAdr, NomRueAdr, CPAdr, VilleAdr, PaysAdr, ContinentAdr.

Les attributs de l'entité *DIPLOMES* sont : **IdDiplome**, NomDiplome, DescriptionDiplome.

Les attributs de l'entité *PERMIS* sont : **NomPerm**.

Les attributs de l'entité *SITES\_RESEAUX\_SOCIAUX* sont : **IdSiteRes**, Lien.

Les attributs de l'entité *CENTREINTERETS* sont : **NomCtrInt**

Les attributs de l'entité *COMPETENCES* sont : **NomCpt**, NomCategorie.

Les attributs de l'entité *CATEGORIE\_CPT* sont : **NomCategorie**.

Les attributs de l'entité *LANGUES* sont : **NomLangue**.

Les attributs de l'entité *PROJETS* sont : **IdProj**, NomProj, DescriptionProj.

Les attributs de l'entité *ETABLISSEMENTSCOLAIRE* sont : **IdEtaSco**, NomEtaSco, MailEtaSco,TelEtaSco.

Les attributs de l'entité *ENTREPRISES* sont : **IdEnt**, NomEnt, MailEnt,TelEnt.

Les attributs de l'entité *EXPERIENCES* sont : **IdExp**, NomExp, DescriptionExp.

Les attributs de l'entité *FORMATIONS* sont : **IdForm**, Niveau, Specialite

L'attribut de l'entité *CATEGORIE\_CPT* est : **NomCategorie**

**Associations :**

Les attributs de l’association *OBTENIR (OBTENTIONDIPLOME)* sont : DateObtention, Mention.

L'attribut de l’association *OBTENIR (OBTENTIONPERMIS)* est : DateObtention.

Les attributs de l’association *POSSEDER* sont : Aucun attribut supplémentaire !

Les attributs de l’association *AIMER (RELATION\_CENTINT\_CAN)* sont : Aucun attribut supplémentaire !

Les attributs de l’association *POSSEDER (RELATION\_COMP\_CAN)* sont : Aucun attribut supplémentaire !

L'attribut de l’association *PARLER (RELATION\_LANG\_CAN)* est : Niveau.

Les attributs de l’association *EFFECTUER (RELATION\_PROJET\_CAN)* sont : DateDebutProj, DateFinProj.

Les attributs de l’association *EFFECTUER (RELATION\_EXP\_CAN)* sont : DateDebutProj, DateFinProj.

Les attributs de l’association *SUIVRE (SUIT\_FORMATION)* sont : DateDebutForm, DateFinForm.

L'attribut de l’association *TRANSMETTRE* est : DateTransmission.

Les attributs de l’association *LOCALITE(1)* sont : Aucun attribut supplémentaire !

Les attributs de l’association *LOCALITE(2)* sont : Aucun attribut supplémentaire !

Les attributs de l’association *SE SITUER(1)* sont : Aucun attribut supplémentaire !

Les attributs de l’association *SE SITUER(2)* sont : Aucun attribut supplémentaire !

Les attributs de l’association *SE SITUER(3)* sont : Aucun attribut supplémentaire !

Les attributs de l’association *HABITER* sont : Aucun attribut supplémentaire !

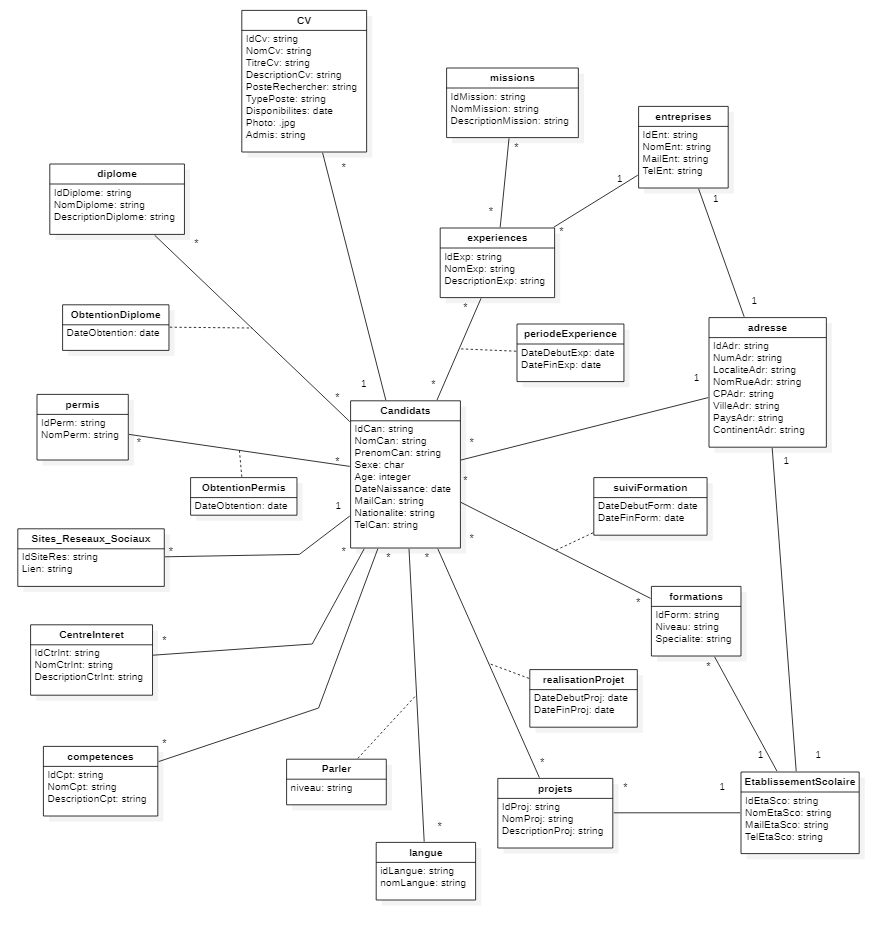
Les attributs de l’association *CATEGORISER* sont : Aucun attribut supplémentaire !

Les attributs de l’association *COMPOSER(RELATION\_EXP\_MISS*)sont : Aucun attribut supplémentaire !

Les attributs de l’association *CLASSER* sont : Aucun attribut supplémentaire !

## Le schéma conceptuel de données (UML)

Le **schéma conceptuel de données** (SCD) ou **le modèle conceptuel de données** (MCD) est décrit ci-dessous selon formalisme « Unified Modeling Language : UML ».



## Le schéma logique de données (Schéma relationnel)

**Le Schéma relationnel de la Base de Données (Modèle Logique de Données MLD) est composé de plusieurs tables.** Elles sont présentées ci-dessous :

Table des adresses :

**ADRESSES** (🗝**IdAdr**, NumAdr, LocaliteAdr, NomRueAdr, CPAdr, VilleAdr, PaysAdr, ContinentAdr)

Table des candidats :

**CANDIDATS** (🗝**IdCan**, IdAdr🡩, NomCan, PrenomCan, Sexe, Age, DateNaissance, MailCan, Nationalite, TelCan)

Table des établissements scolaires :

**ETABLISSEMENTSCOLAIRE** (🗝**IdEtaSco**, IdAdr🡩, NomEtaSco, MailEtaSco,TelEtaSco)

Table des entreprises :

**ENTREPRISES** (🗝**IdEnt**, IdAdr🡩, NomEnt, MailEnt,TelEnt)

Table des projets :

**PROJETS** (🗝**IdProj**, IdEtaSco🡩, NomProj, DescriptionProj)

Table des langues :

**LANGUES** (🗝**NomLangue**)

Table des compétences :

**COMPETENCES** (🗝**NomCpt**, NomCatégorie🡩)

Table des catégories des compétences :

**CATEGORIE\_CPT** (🗝**NomCategorie**)

Table des centres intérêts :

**CENTREINTERET** (🗝**NomCtrInt**)

Table des sites et réseaux sociaux:

**SITES\_RESEAUX** (🗝**IdSiteRest**, IdCan🡩, Lien)

Table des permis :

**PERMIS** (🗝**IdPerm**, NomPerm)

Table des diplômes :

**DIPLOMES** (🗝**IdDiplome**, NomDiplome, Mention, DescriptifDiplome)

Table des Cv:

**CV** (🗝**IdCv**, IdCan🡩, NomCv, TitreCv, DescriptionCv, PosteRechercher, TypePoste, Disponibilites, Photo, Admis, DateTransmission)

Table des expériences :

**EXPERIENCES** (🗝**IdExp**, IdEnt🡩, NomExp, DescriptifExp)

Table des missions :

**MISSION** (🗝**IdMission**, NomMission, DescriptifMission)

Table des formations :

**FORMATIONS** (🗝**IdForm**, IdEtaSco🡩, Niveau, Specialite)

Table d'obtention des diplômes :

**OBTENTIONDIPLOME** (🗝***IdDiplome*🡩, *IdCan***🡩, DateObtention)

Table d'obtention des permis :

**OBTENTIONPERMIS** (🗝***IdPermis*🡩, *IdCan***🡩, DateObtention)

Table des centres d'intérêts des candidats :

**RELATION\_CENTINT\_CAN** (🗝***IdCtrInt*🡩, *IdCan***🡩)

Table des langues parlées par les candidats :

**RELATION\_LANG\_CAN** (🗝***IdLangue*🡩, *IdCan***🡩, Niveau)

Table des projets réalisés par les candidats :

**RELATION\_PROJET\_CAN** (🗝***IdProj*🡩, *IdCan***🡩, DateDebutProj, DateFinProj)

Table des expériences des candidats :

**RELATION\_EXP\_CAN** (🗝***IdExp*🡩, *IdCan***🡩, DateDebutExp, DateFinExp)

Table des missions durant les expériences :

**RELATION\_EXP\_MISS** (🗝***IdExp*🡩, *IdMission***🡩)

Table des formations suivies par les candidats :

**SUIT\_FORMATIONS** (🗝***IdForm*🡩, *IdCan***🡩, DateDebutForm, DateFinForm)

Un **dictionnaire de données** contient les détails de toutes les descriptions des différentes colonnes de toutes les tables de la BD.

Ces descriptifs constituent une partie des **métadonnées** de la BD.

Le dictionnaire de données de la BD est détaillé ci-dessous :

## Le dictionnaire de données & les métadonnées

Le **dictionnaire de données** de la BD est le suivant (ces descriptifs constituent un sous ensemble des **métadonnées** de la BD) :

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribut**  **Colonne** | **Signification - Description** | **Domaine -Type de données**  **Contrainte Naturelles** | **Contrainte (Meta Table)**  **Expression Régulière** |
| Admis | Réponse d'acceptation ou de refus du candidat | Caractères/Texte, 150, ACCEPTE ou REFUSE | '^(ACCEPTE|REFUSE)$' |
| Age | Age du candidat | Entier, Maximum 3, >= 0 | '^([0-9]{1,3})|NULL$' |
| ContinentAdr | Adresse (Nom du continent) | Caractères/Texte, 150, en MAJUSCULE | '^[A-Z]+[(-| )?[A-Z]+]\*$' |
| CPAdr | Adresse (Code Postal) | Caractères/Texte, 150 | '^([0-9]{5})|NULL$' |
| DateDebutExp | Date de début d'expérience | Date |  |
| DateDebutForm | Date de début de formation | Date |  |
| DateDebutProj | Date de début de projet | Date |  |
| DateFinExp | Date de fin d'expérience | Date |  |
| DateFinForm | Date de fin de formation | Date |  |
| DateFinProj | Date de fin de projet | Date |  |
| DateNaissance | Date de naissance du candidat | Date |  |
| DateTransmission | Date de transmission du Cv | Date |  |
| DescriptifCpt | Descriptif de la compétence | Caractères/Texte, 50 |  |
| DescriptifCtrInt | Descriptif du centre d'intérêt | Caractères/Texte, 50 |  |
| DescriptifCv | Descriptif du cv | Caractères/Texte, 20 |  |
| DescriptifDiplome | Description du diplome | Caractères/Texte, 50 |  |
| DescriptifExp | Descriptif de l'expérience | Caractères/Texte, 50 |  |
| DescriptifMission | Descriptif de la mission | Caractères/Texte, 50 |  |
| DescriptifProj | Description du projet | Caractères/Texte, 50 |  |
| Disponibilite | Disponibilité du candidat | Date |  |
| **IdAdr** | Numéro de l'adresse (Unique par adresse) | Caractères/Texte, 150 | '^[0-9]+$' |
| **IdCan** | Numéro du candidat (Unique par candidat) | Caractères/Texte, 150 | '^[0-9]+$' |
| **IdCv** | Numéro de Cv (Unique par Cv) | Caractères/Texte, 150 | '^[0-9]+$' |
| **IdDiplome** | Numéro de diplôme (Unique par diplôme) | Caractères/Texte, 150 | '^[0-9]+$' |
| **IdEtaSco** | Numéro de l'établissement scolaire (Unique par établissement) | Caractères/Texte, 150 | '^[0-9]+$' |
| **IdEnt** | Numéro de l'entreprise (Unique par entreprise) | Caractères/Texte, 150 | '^[0-9]+$' |
| **IdExp** | Numéro de l'expérience (Unique par expérience) | Caractères/Texte, 150 | '^[0-9]+$' |
| **IdForm** | Numéro de la formation (Unique par formation) | Caractères/Texte, 150 | '^[0-9]+$' |
| **IdMission** | Numéro de la mission (Unique par mission) | Caractères/Texte, 150 | '^[0-9]+$' |
| **IdProj** | Numéro du projet (Unique par projet) | Caractères/Texte, 150 | '^[0-9]+$' |
| **IdSiteRes** | Numéro du site/réseau social (Unique par site/réseau social) | Caractères/Texte, 150 | ^[0-9]+$' |
| Lien | Lien vers le site / Réseau social | Caractères/Texte, 150 |  |
| LocaliteAdr | Adresse (Type de la voie) | Caractères/Texte, 150, en MAJUSCULE | '^(RUE|BOULEVARD|AVENUE|QUAI|IMPASSE|PONT|PLACE|SQUARE|ALLEE|ALLEES|VOIE|MONTEE|ESPLANADE|ROUTE|VOIRIE|CITE|CHEMIN|PARVIS)&$' |
| MailCan | Mail du candidat | Caractères/Texte, 150 | '^([A-Za-z]+[A-Za-z0-9--\.\_]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4})|NULL$' |
| MailEnt | Mail de l'entreprise | Caractères/Texte, 150 | '^([A-Za-z]+[A-Za-z0-9--\.\_]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4})|NULL$' |
| MailEtaSco | Mail de l'établissement Scolaire | Caractères/Texte, 150 | '^([A-Za-z]+[A-Za-z0-9--\.\_]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4})|NULL$' |
| Mention | Mention obtenue | Caractères/Texte, 150 | '^[A-Za-z]+[A-Za-z0-9.]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4}$' |
| Nationalite | Nationalité du candidat | Caractères/Texte, 150 | '^([A-Z]+)|NULL$' |
| Niveau | Le niveau de langue du candidat | Caractères/Texte, 150 | '^[A-Z][a-zç]$' |
| NomCan | Nom du candidat | Caractères/Texte, 150en MAJUSCULE | '^[A-Z]+[(-| )?[A-Z]+]\*$' |
| **NomCategorie** | Nom de la catégorie des compétences (Unique pour chaque catégorie) | Caractères/Texte, 150 | ''^([A-Z]+[(-| )?[A-Z]+]\*)|NULL$' |
| **NomCpt** | Nom de la compétence (Unique par compétence) | Caractères/Texte, 150 | '[A-Z#\.-\_/+ ]' |
| **NomCtrInt** | Nomdu centre d'intérêt (Unique par centre d'intérêt) | Caractères/Texte, 150 | '[A-Z#\.-\_/+ ]' |
| NomCv | Nom du Cv | Caractères/Texte, 150 | '^[A-Za-z0-9\_\-/]+\.pdf$' |
| NomDiplome | Nom du diplôme | Caractères/Texte, 150 |  |
| NomEnt | Nom de l'entreprise | Caractères/Texte, 150; en MAJUSCULE |  |
| NomEtaSco | Nom de l'établissement Scolaire | Caractères/Texte, 150en MAJUSCULE |  |
| NomExp | Nom de l'expérience | Caractères/Texte, 150 |  |
| **NomLangue** | Nom de la langue (Unique par langue) | Caractères/Texte, 150en MAJUCULE | '^[A-Z]+[(-| )?[A-Z]+]\*$' |
| NomMission | Nom de la mission | Caractères/Texte, 150 |  |
| **NomPerm** | Nom de permis (Unique par permis) | Caractères/Texte, 150dans la liste des permis existants | '^(AM|BSR|A|A1|A2|B|B1|B2|BE|BVA|C|C1|CE|C1E|D|D1|D2|DE|DE1)$' |
| NomProj | Nom du projet | Caractères/Texte, 150 |  |
| NomRueAdr | Adresse (Nom de la voie) | Caractères/Texte, 150 | '^[A-Za-z0-9]+((\-| )[A-Za-z0-9]+)?$' |
| NumAdr | Adresse (Numéro dans la voie) | Caractères/Texte, 150 | '^[0-9]{1,5}( BIS| TER)?$' |
| PaysAdr | Adresse (Nom du Pays) | Caractères/Texte, 150 en MAJUSCULE | '^[A-Z]+[(-| )?[A-Z]+]\*$' |
| PosteRechercher | Nom du poste recherché | Caractères/Texte, 150 |  |
| PrenomCan | Prénom du candidat | Caractères/Texte, 150, 1 ere lettre de chaque prénom en MAJUSCULE, le reste en minuscule | '^[A-Z][a-zéïàèîôû]+((\-| )[A-Z][a-zéïàèîôû]+)?$' |
| Sexe | Sexe du candidat | Caractères/Texte, 1, M (Masculin) ou F (Féminin) | '^(M|F)|NULL$' |
| TelCan | Téléphone du candidat | Caractères/Texte, 150 | '^((\+33|0)[1-9][0-9]{8})|NULL$' |
| TelEnt | Téléphone de l'entreprise | Caractères/Texte, 150 | '^((\+33|0)[1-9][0-9]{8})|NULL$' |
| TelEtaSco | Téléphone de l'établissement Scolaire | Caractères/Texte, 150 | '^((\+33|0)[1-9][0-9]{8})|NULL$' |
| TitreCv | Titre du Cv | Caractères/Texte, 150 |  |
| TypePoste | Type de poste recherché | Caractères/Texte, 150 |  |
| VilleAdr | Adresse (Nom de la ville) | Caractères/Texte, 150en MAJUSCULE | '^[A-Z]+[(-| )?[A-Z]+]\*$' |

## Expression des contraintes sur les données

L’expression des contraintes sur les données est une étape primordiale pour valider les données. La liste des contraintes à valider est donnée ci-dessous.

La majorité des contraintes sont gérées avec des expressions régulières présentes dans des méta-tables sous les noms *META\_NomTable*.

- Adresses :

Le **numéro d'identification** doit être unique pour chaque adresse. Le **numéro de la rue** doit être un entier éventuellement suivi de BIS ou TER. Le **type de la voie dans une adresse** prend une valeur dans la liste suivante : {RUE, BOULEVARD, AVENUE, QUAI, IMPASSE, PONT, PLACE, SQUARE, ALLÉE, ALLÉES, VOIE, MONTÉE, ESPLANADE, ROUTE, VOIRIE, CITÉ, CHEMIN, PARVIS…}, les valeurs sont du type alphabétique. Le **code postal** doit être un être composé de 5 chiffres. Les **villes, pays et continents** sont constitués essentiellement de majuscules avec éventuellement des espaces ou des tirets pour ceux avec des noms composés.

- Candidats :

Le **numéro d'identification** doit être unique pour chaque candidat. Le **numéro d'identification** de l'adresse doit être présent dans la table adresse. Le **nom** d'un candidat est une chaîne alphabétique entièrement en majuscule avec éventuellement des espaces ou tirets pour les noms composés. Le **prénom** d'un candidat est une chaîne alphabétique dont la première lettre de chaque prénom (pour les prénoms composés) doit être en majuscule et tout le reste des prénoms doit être en minuscule. L'**âge** d'un candidat est un entier positif avec maximum 3 chiffres. Le **sexe** d'un candidat est représenté par un caractère présent dans la liste {M, F} pour masculin et féminin respectivement. Le **numéro de téléphone** d'un candidat doit commencer par +33 ou 0 suivi de 9 chiffres. Le **mail** d'un candidat doit correspondre à l'expression régulière des mails.

- Établissements :

Le **numéro d'identification** doit être unique pour chaque établissement scolaire. Le **numéro d'identification** de l'adresse doit être présent dans la table adresse. Le **nom de l'établissement** doit être en majuscule. Le **mail** de l'établissement doit correspondre à l'expression régulière des mails. Le **numéro de téléphone** d'un établissement doit commencer par +33 ou 0 suivi de 9 chiffres.

- Entreprises :

Le **numéro d'identification** doit être unique pour chaque entreprise. Le **numéro d'identification** de l'adresse doit être présent dans la table adresse. Le **nom de l'entreprise** doit être en majuscule. Le **mail** de l'entreprise doit correspondre à l'expression régulière des mails. Le **numéro de téléphone** d'un établissement doit commencer par +33 ou 0 suivi de 9 chiffres.

- Projets :

Le **numéro d'identification** doit être unique pour chaque projet. Le **numéro d'identification** de l'établissement scolaire doit être présent dans la table etablissementscolaire. Le **nom du projet** est composé d’une suite de lettres, de tirets ou d’espaces ou de chiffres. La **description du projet** peut utiliser n’importe quel caractères ou aucun.

- Langues :

Le **nom** de la langue doit être unique pour chaque langue. Il doit être essentiellement constitué de majuscules.

- Catégorie\_Cpt :

La **nom** de la catégorie doit être unique pour chaque catégorie.

- Compétences :

Le **nom de la compétence** doit être unique pour chaque compétence et en lettres majuscules, avec acceptation de certains caractères spéciaux. La catégorie d'une compétence doit exister dans la table Categorie\_Cpt.

- Centre Intérêt:

Le **nom du centre d’intérêt** doit être unique pour chaque centre d'intérêt et peut être constitué de n’importe quel caractère.

- Sites/Réseaux sociaux :

Le **numéro d'identification** doit être unique pour chaque site/réseau social. Le **numéro d’identification** du candidat doit être présent dans la table des candidats. Le **lien** du site ou du réseau social correspond à l’expression régulière d’une URL.

- Permis :

Le **nom du permis** doit être unique pour chaque permis et doit être présent dans la liste { AM,BSR,A,A1,A2,B,B1,B2,BE,BVA,C,C1,CE,C1E,D,D1,D2,DE,DE1}.

- Diplômes :

Le **numéro d'identification** doit être unique pour chaque diplôme. Le **nom du diplôme** est une chaîne de caractères alphanumériques et sa **description** peut être faite par l’utilisation de différents caractères.

- Cv :

Le **numéro d'identification** doit être unique pour chaque Cv. Le **numéro d’identification** du candidat doit être présent dans la table des candidats. Le **nom** du Cv et la **photo** (correspond au chemin)ne contiennent pas d’espace, la **description** du CV et le **poste recherché** sont constitués d’une chaîne de caractères alphanumérique. **Description**, **type de poste** et **disponibilité** peuvent être constitués de n’importe quel caractère. Le champ **admis** peut avoir soit la valeur ‘ACCEPTE’ ou ‘REFUSE’.

- Expériences :

Le **numéro d'identification** doit être unique pour chaque expérience. Le **numéro d’identification** de l'entreprise doit être présent dans la table des entreprises. Le **nom de l’expérience** est constitué de caractères alphanumériques et sa **description** peut être constituée de n’importe quel caractère ou peut peut être vide.

- Missions :

Le **numéro d'identification** doit être unique pour chaque mission. Le **nom de la mission** est constitué de caractères alphanumériques et sa **description** peut être constituée de n’importe quel caractère ou peut peut être vide.

- Formations :

Le **numéro d'identification** doit être unique pour chaque formation. Le **numéro d’identification** de l’établissement scolaire doit être présent dans la table des établissements scolaires. Le **niveau** et la **spécialité** sont constitués de caractères alphanumériques.

Les clés étrangères correspondantes à l'association des numéros d'identification doivent être présentes dans les tables associées.

# 3. Exemple de contenu de la base de données (instance de BD)

Une instance de la base de données des CV est donnée ci-dessous. Les scripts SQL correspondants sont données dans les fichiers :

- de création des structures des tables de nom : Gx\_CreatDon\_CV.sql

- de création des données de nom : Gx\_InsertDon\_CV.sql

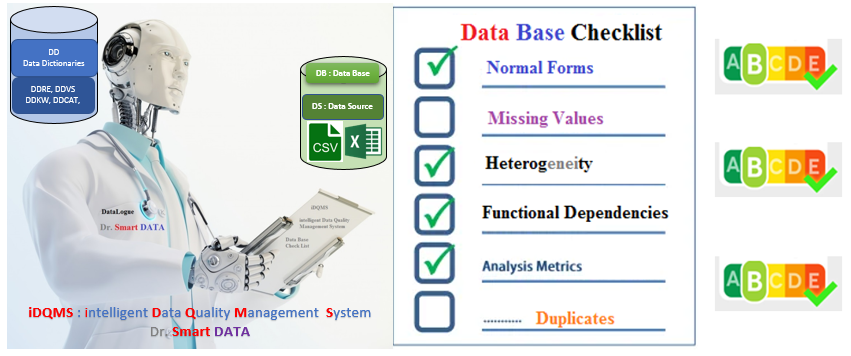
- de manipulation des données de nom : Gx\_ManipDon\_CV.sql

Un jeu de test est primordial pour valider notre travail. Nous donnons, dans ce qui suit, les détails de ce jeu de données.

Il va falloir définir très précisément des « règles » (des contraintes très précises) sur les données (Construire/Définir les métadonnées) afin d’assurer/améliorer la qualité des données

Exemple : Le nom d’un(e) client(e) doit être :

1. NON vide, alphabétique,
2. en MAJUSCULE,
3. sans espace superflu (devant et entre les mots),
4. les seuls caractères spéciaux autorisés sont – et les lettres françaises avec accent…



## 3.1. Exemple d’instances de la BD des CV

Une instance inspirée des CV de la promotion M2EID2 de l’année 2020-2021 est donnée ci-dessous.

Avec l'insertion d'un seul CV, on a l'instance suivante :

Table Adresses :

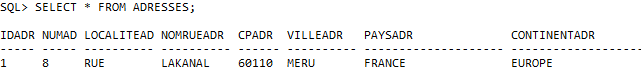


Table Candidats :



Table CentreInteret :

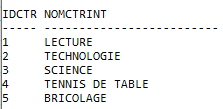


Table Relation\_CtrInt\_Can :

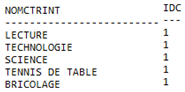


Table Langues :



Table Relation\_Lang\_Can :

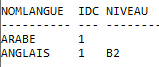
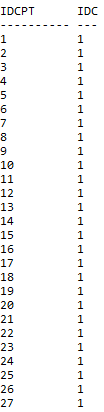


Table Competences :

A FAIRE

Table Relation\_Comp\_Can :



## 3.2. Création des structures de données (SQL-ORACLE)

La création des tables, en SQL2, est détaillée ci-dessous. Les tables sont créées dans le fichier G1\_CreatDon\_CV.sql.

Notre entrepôt de données est constitué de 25 tables et 15 méta-tables.

La création des tables se fait avec les commandes

CREATE TABLE NomTable(

Attr1 Type1,

Attr2 Type2,

…

CONSTRAINT PKNomTable PRIMARY KEY(Attr),

[CONSTRAINT FK\_NomTab1\_NomTab2\_Attr FOREIGN KEY(Attr) REFERENCES NomTable2(Attr),...]

);

De plus, la création des méta-tables qui permettent la vérification des contraintes sur les colonnes sont créées avec les commandes :

CREATE TABLE META\_NomTable

(

CONST\_Attr1 VARCHAR2(2000) DEFAULT NULL

[CONST\_Attr2 VARCHAR2(2000) DEFAULT NULL,...]

);

Les CONST\_Attr permettent de stocker les expressions régulières que doivent respecter les attributs.

Elles vont donc permettre :

- 1) On a facilement accès à la visualisation des contraintes sur les colonnes. En faisant simplement un 'SELECT \*' sur les méta-tables.

- 2) On peut facilement ajouter des contraintes sur un attribut quand bon nous semble, avec un simple INSERT INTO dans la méta-table avec l'expression régulière que l'on souhaite ajouter. Ainsi, si on souhaite faire évoluer notre table avec d'autres données, il est assez facile de faire ce procédé.

- 3) Enfin les méta-tables vont permettent des insertions plus sûres. En effet, avec une procédure on va pourvoir vérifier les contraintes lorsque l'on souhaite insérer une donnée. Toutes nos procédures d'insertions de nouvelles données vont vérifier que les contraintes des méta-tables sont vérifiées.

## 3.2.Bis. Extraction automatique des données.

L'extraction de données des CV au format PDF se fait avec le script Python : Extract\_PDF\_V2.py.

Il va procéder de différentes manières.

Prérequis : Deux dossiers comportant les CV acceptés et les CV non acceptés sous forme PDF que l'on souhaite traiter. Idéalement, il est préférable que les fichiers PDF aient le nom que vous avez demandé afin d’accélérer le traitement des informations.

La seconde étape est de lancer le Script Python extract\_PDF\_V2.py. Pendant l'exécution de ce script on peut voir l'avancement de l'extraction des données grâce à l'affichage d'une barre de progression que l'on à implémenter sur le Terminal.

1) Le script commence par ouvrir le fichier PDF avec des fonctions que l’on a implémentées

2) Puis le script va convertir le fichier PDF en texte.

3) Il va **convertir le texte en FRANÇAIS** (ex : United Kingdom -> Royaume Unis). Ainsi, nous n'avons plus de problème liés aux langues. On passe de **N'IMPORTE QUELLES LANGUES** aux français.

4) Puis il réalise la détections et l'extraction des informations contenues dans le texte

Cette étape se fait avec deux procédés différents selon les cas :

1) Soit on utilise des expressions régulières.

2) Le deuxième procédé concerne principalement les prénoms. En effet, il n'est pas facile de créer une expression régulière pour les prénoms car c'est simplement du texte et il peut facilement être confondu avec un mot quelconque.

Pour pallier le problème, nous avons implémentés deux solutions en fonction des cas rencontrés :

La première consiste à trouver les mots qui se trouvent dans l'adresse mail et/ou dans le nom du fichier. En effet,  la plupart des gens ont leur prénom dans leur adresse mail ou dans le nom du fichier qu'il transmette.

La seconde méthode est l'utilisation d'une base de données avec des milliers de prénoms et on va chercher les mots du texte du CV dont le SOUNDEX et la distance de  LEVENSHTEIN sont proches avec les prénoms de la base de données des Prénoms. Puis on prend celui qui maximise les scores.

Toutes les extractions des données textuelles se font de cette manière (soit avec expression régulières soit avec les algorithmes du SOUDEX et LEVENSHTEIN).

De ce fait, l'extraction des données textuelles permettent déjà de faire un premier tri, dans le respect des contraintes, avant même le passage par les procédures de validation.

Maintenant que nous avons extrait toutes les informations textuelles via des expressions ou des algorithmes de similarité, il reste une donnée à extraire : la photo.

Nous avons fait le choix de ne pas utiliser le dossier des photos, pour une raison simple : Quand on pense plus grand dans l'extension de notre entrepôt de données. Il est très peu fréquent qu'une personne envoie sa photo en plus de son CV.

On a donc voulu extraire la photo directement à partir du CV, si elle existe.

Pour ce faire, on va convertir le fichier PDF en Image.

A partir de cette image, nous allons appliquer un algorithme de détection de visage, puis on va sauvegarder dans un dossier Photo\_CV, la détection de cet algorithme. A la fin de l’exécution du script, on aura donc un dossier avec toutes les photos des CV qui comportant une photo.

Une fois toutes les informations du CV extraites, le script Python va générer automatiquement des requêtes, correspondant au différente insertion dans les tables et va les ajouter à la suite de notre fichier SQL d'insertion G1\_InsertDon\_CV.sql.

Là encore sans aucune intervention de notre part, tout est automatique.

Ainsi, après l'exécution de ce script, on peut exécuter les fichiers SQL, soit en faisant copier coller des commandes SQL soit en exécutant directement le script. Après l’exécution de ces requêtes, on aura les tables remplies

## 3.3. Insertion des données dans une table et qualité des données (SQL/PLSQL-ORACLE)

L’insertion des données dans une table devra se faire de manière à respecter toutes les contraintes définies précédemment.

La «**bonne et complète**» insertion des données dans une table devrait se faire avec une **procédurePL/SQL** qui permet de **contrôler** le contenu de chaque colonne et non seulement avec la commande SQL « **INSERT INTO NomTable (…) VAUES (…) ;** » :

CREATE OR REPLACE PROCEDURE **INSERT\_NomTable**(

Val1 Type1,

Val2 Type2,

…

) AUTHID CURRENT\_USER AS

CURSOR cur IS SELECT \* FROM META\_NomTable;

anomalie VARCHAR2(20) := 'FALSE';

BEGIN

FOR ligne IN cur LOOP

IF (ligne.CONST\_Val1 IS NOT NULL AND NOT REGEXP\_LIKE(Val1, ligne.CONST\_Val1)) OR

(ligne.CONST\_Val2 IS NOT NULL AND NOT REGEXP\_LIKE(Val2, ligne.CONST\_Val2)) OR

…

THEN

anomalie := 'TRUE';

END IF;

END LOOP;

IF anomalie = 'TRUE' THEN

DBMS\_OUTPUT.PUT\_LINE(' /!\ ERROR : La ligne ne respecte pas les contraintes !! /!\ ');

ELSE

INSERT INTO **NomTable** VALUES(Val1, Val2,…);

DBMS\_OUTPUT.PUT\_LINE('Ligne inseree avec succes.');

END IF;

END;

/

Les insertions dans notre entrepôt de données se font via des procédures nommées INSERT\_NomTable. Ces procédures s'appuient sur les méta-tables des tables en questions. Ainsi, à chaque insertion, en plus de vérifier les contraintes de tables, on va pouvoir pour chaque donnée vérifier une ou plusieurs expressions régulières définies dans les méta-tables.

Leur fonctionnement est simple. Pour une données (une ligne) qu'on veut insérer, on va vérifier tous les attributs indépendamment grâce aux expressions régulières qui leurs correspondent. Si, un des attributs ne respecte pas ses contraintes, la ligne ne sera pas insérée.

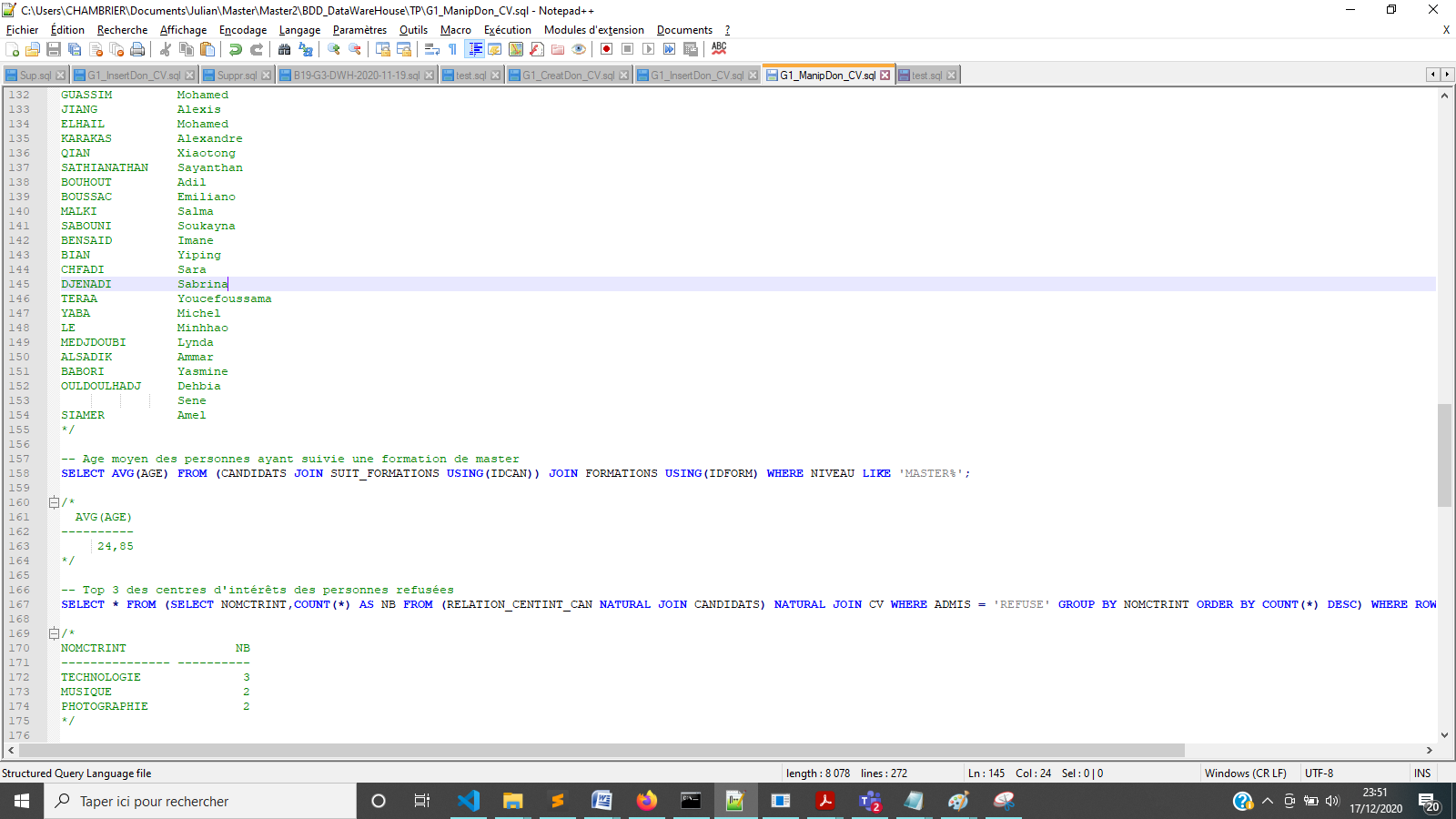
## 3.4. Interrogations & Manipulations (SQL/PLSQL-ORACLE)

Plusieurs requêtes sont disponibles dans le fichier : G1\_ManipDon\_CV.sql.

Voici quelques exemples :

-- Age moyen des personnes ayant suivie une formation de master

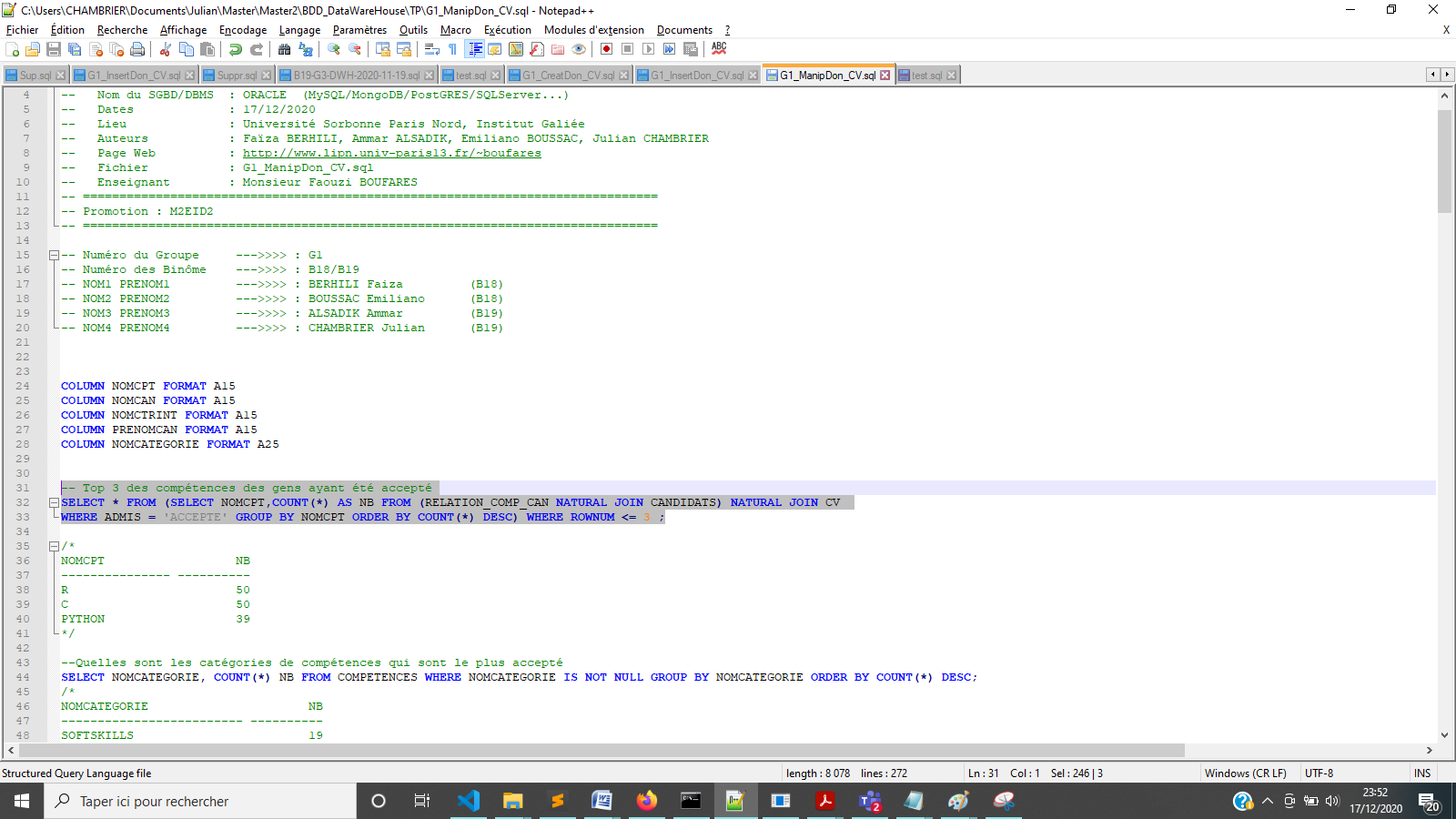
SELECT AVG(AGE) FROM (CANDIDATS JOIN SUIT\_FORMATIONS USING(IDCAN)) JOIN FORMATIONS USING(IDFORM) WHERE NIVEAU LIKE 'MASTER%';



-- Top 3 des compétences des gens ayant été accepté

SELECT \* FROM (SELECT NOMCPT,COUNT(\*) AS NB FROM (RELATION\_COMP\_CAN NATURAL JOIN CANDIDATS) NATURAL JOIN CV

WHERE ADMIS = 'ACCEPTE' GROUP BY NOMCPT ORDER BY COUNT(\*) DESC) WHERE ROWNUM <= 3 ;



# 4. Sujet du Projet Annuel

Le sujet de notre projet annuel est « DES BASES AUX ENTREPÔTS DE DONNÉES, données structurées ou NON structurées (Oracle, MySQL ou MongoDB ?) ». Il s’agit d’un travail à réaliser par groupe d’étudiant.e.s.

Le travail à faire consiste à :

- Concevoir & développer une interface…

- Présentation orale à faire la semaine qui précède l’examen du mois de mars (fin des cours).

- Rapport (Word/PDF) + Diapositives (PPT/PDF)

- Codes (Oracle, MySQL et MongoDB)

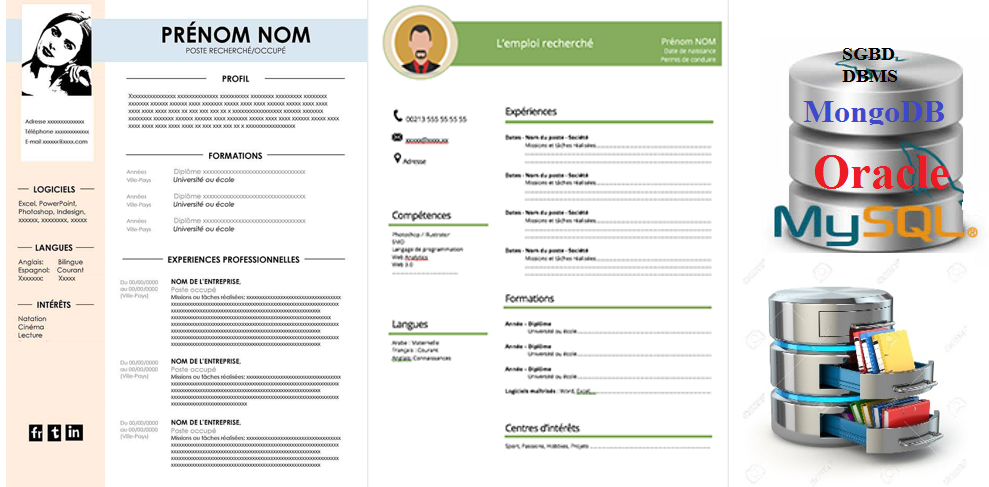
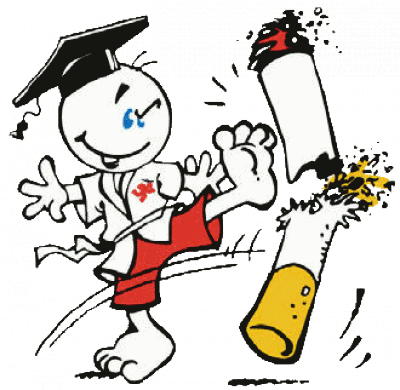
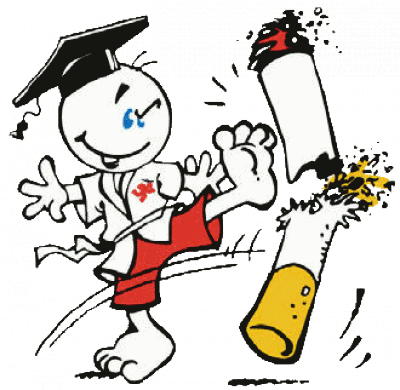
## 4.1. : Conception & Implantation de Bases de Données

Construction de **données structurées** à partir de **données NON structurées** (à partir de documents papiers ou numériques qui correspondent à des CVs ; ***From PDF Files to SQL or NoSQL Data Bases****).*

**8.1.1. 🡪 Donnez le schéma conceptuel** (en utilisant l’un des deux formalismes **UML** ou **E/A)**, le **schéma logique relationnel structuré SQL**et le **schéma logique Non-relationnel Non-structuré NoSQL** de la BD qui permet de stocker les données pertinentes que l’on peut trouver dans les CVs des personnes qui candidatent pour une inscription en Master 2 Informatique « Exploration Informatique et Décisionnel » (M2EID2).

**8.1.2. 🡪 Donnez une instance** (quelques lignes, au moins 50, ou objets ou enregistrements réalistes) de votre BD !

**8.1.3. 🡪 Donnez TROIS implémentations sur Oracle, MySQL et MongoDB**



## 4.2. : Conception & Implantation d’Entrepôts de Données

En disposant de données (stockées dans la BD des CV) sur une période de 10 années,

**4.2.1. 🡪 Concevez un entrepôt de données** qui permet d’analyser profondément le profil des candidats en M2EID2 ! (Vous êtes libres de prendre les hypothèses que vous voulez !).

**4.2.2. 🡪Donnez une instance** (quelques lignes, au moins 50, ou objets ou enregistrements réalistes) de votre ED !

**4.2.3. 🡪 Donnez TROIS implémentations sur Oracle, MySQL et MongoDB**

# Conclusion

Dans cette première partie du projet, nous avons donc développés des algorithmes qui permettent l’extraction automatique de données à partir de PDF. On a simplement à lancer un script et tout se fait automatiquement, de l’extraction des données à l’insertion dans les tables.

Cette partie d’extraction nous a demandé énormément de travail, car elle demande beaucoup de compétences techniques.

De plus, nous avons commencé à ouvrir notre projet avec, par exemple, la détection de la photo du candidat pour coller au mieux à la réalité de terrain.

Dans la seconde partie du projet, on compte améliorer cette partie en intégrant l’apprentissage supervisé pour construire nos tables. On va donc concevoir un algorithme qui va apprendre sur des CV d’apprentissage. Et quand on lui donne un CV sur lequel il n’a pas appris, il sera capable automatiquement et sans expressions régulières de détecter les différents éléments constitutifs d’un CV pour créer un entrepôt de données. De plus, on aimerait à partir d’un CV, nous dire s’il a des chances d’être accepté ou non, grâce à du Machine Learning.

Par exemple, on peut prendre une base de données de CV en texte et on peut donner la position de chaque information à l’algorithme d’apprentissage, par exemple le nom se trouve entre le 10e caractère et le 16e caractère, une expérience de situé entre le 800e caractère et le 1000e caractère, pour bien lui faire apprendre. C’est un processus long, car on veut apprendre sur une base d’apprentissage relativement élevé, mais on pense pouvoir obtenir de meilleurs résultats.

Si tout se passe bien pour un CV de test qu'il ne connaît pas, l'algorithme doit renvoyer la position de chaque information du CV dans le texte et ainsi on a plus qu'à récupérer toutes ces données et à les insérer dans les tables.

On va donc allier plusieurs matières au sein de ce projet. C'est souvent ce qu'on retrouve dans le monde professionnel. Il nous permet donc de nous immerger au plus près des attentes d'une entreprise.