YouCode / Youssoufia

Développeur Data

2023-2024

Projet : Job Analysis

# **Introduction**

En tant que développeur Data, dans le cadre de ce projet nous souhaitons réaliser une analyse approfondie du marché du travail dans ces domaines émergents afin de mieux cibler nos initiatives de recrutement, d'acquisition de talents et de développement de compétences.

Nous souhaitons également visualiser les résultats de manière claire et informative, permettant ainsi aux parties prenantes de mieux comprendre les conclusions tirées de l'analyse.

De plus, nous cherchons à mettre en place un système de stockage des données pour faciliter la gestion et l'accès aux informations collectées tout au long du projet.

# **Planification du travail**

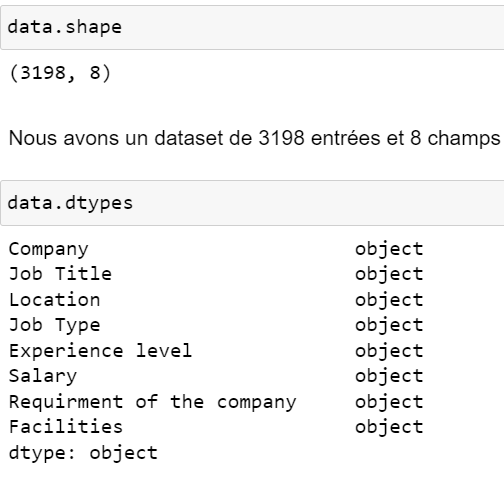
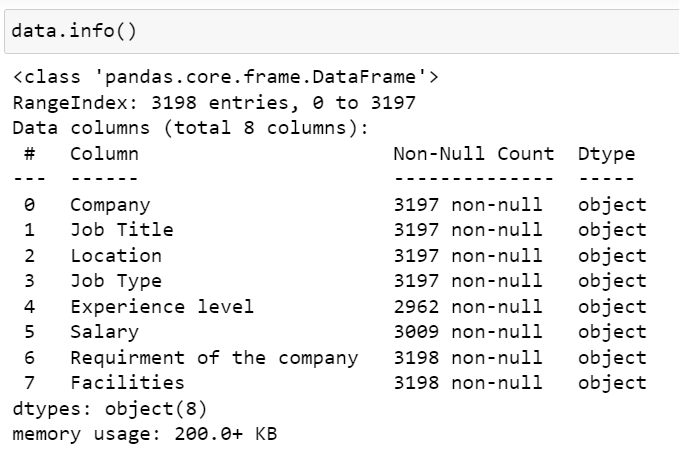
Pour une bonne gestion et du projet, nous avons divisé le travail en plusieurs taches avec l’outil Trello.

Le projet était réparti sur quatre (O4) axes comme présentés sur la figure ci-dessous :



# **Exploration & Compréhension des données**

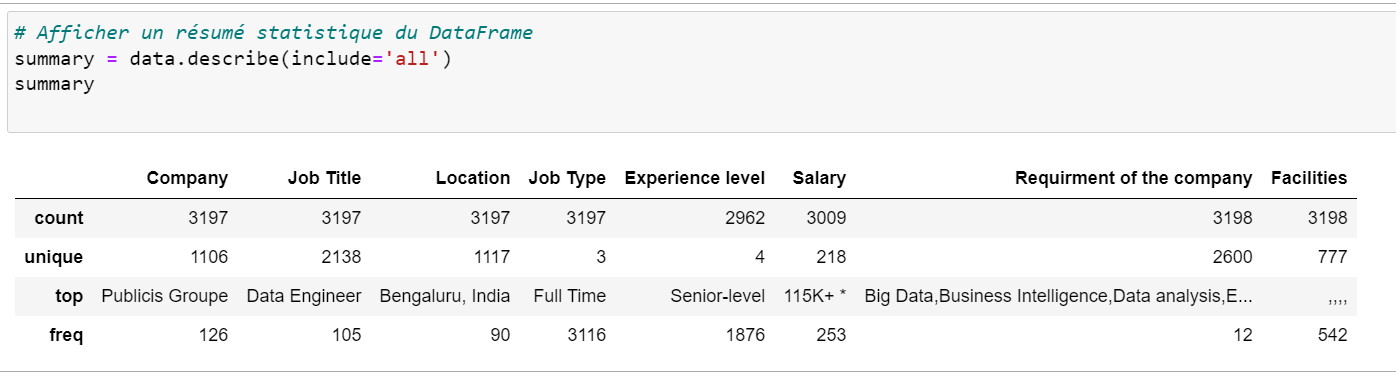
Nous allons exploiter un dataset de 3198 lignes et 8 colonnes, de types objet

Nous avons 3198 entrées.

* Seules les colonnes "**Facilities**" et "Requirment of company " semblent toutes bien remplies
* Les colonnes "**Company**", "**Job Title**", "**Location**" ont seulement une valeur manquante
* La colonne **"Experience level**" **a 2962** entrées **sur 3198**. Donc près **de 93%** de valeurs remplies et **7%** de valeurs manquantes
* La colonne "**Salary**" a **3009** entrées sur **3198**. Donc **94%** de valeurs remplies et **6%** de valeurs manquantes.

Déjà, nous nous rendons comptons que le dataset a quelques impuretés. Pour nous assurer de réaliser une analyse correcte, nous allons procéder à un nettoyage du dataset (data cleaning).



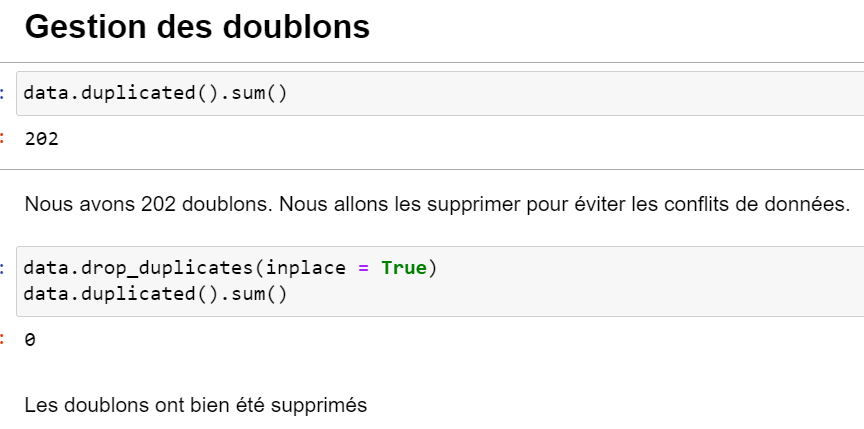
Des statistiques résumées sur les différents champs

# **Data Cleaning (Nettoyage)**

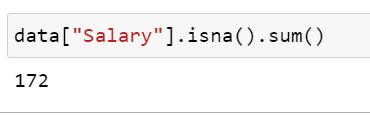
Cette étape consiste à nettoyer les données en corrigeant les erreurs, en supprimant les impuretés pour ne conserver que les valeurs propres, correctes, utiles et pertinentes. Nous serons amenés à dans ce projet à :

* Gérer les doublons
* Gérer les valeurs manquantes (Salary)
* Extraction d’information (Location =🡺Country)
* Uniformiser les unités (monnaie)

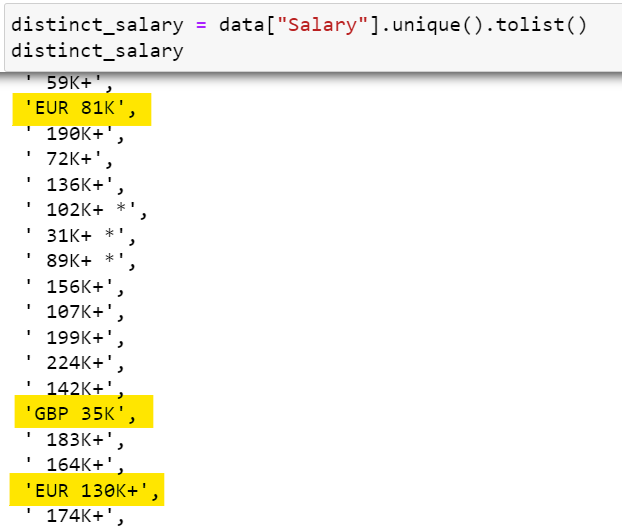
**Gestion des doublons**

****

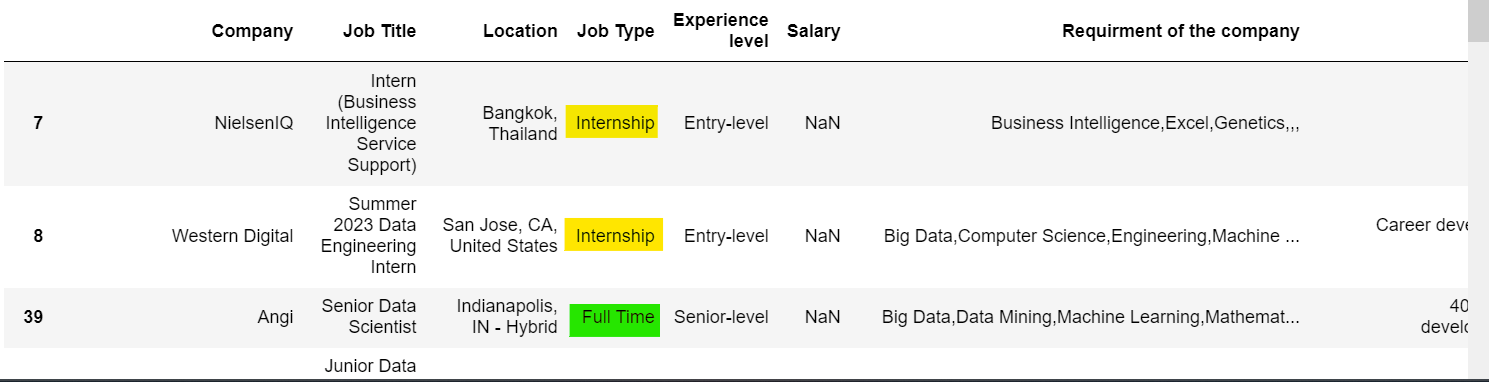
Nous allons maintenant procéder au cas par cas avec les colonnes notamment la colonne Salary et Location.



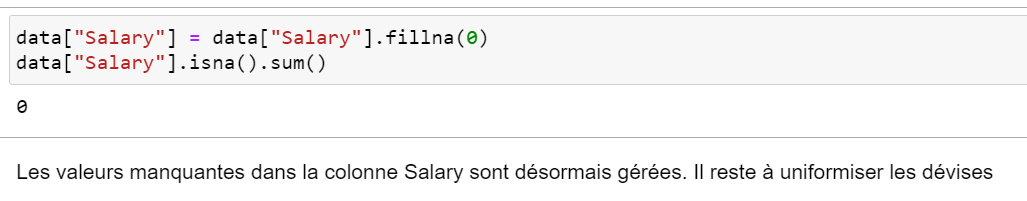
La colonne **Salary** a 172 valeurs manquantes. Après une exploration plus approfondie, nous avons trouvé que les devises ne sont pas uniformes.

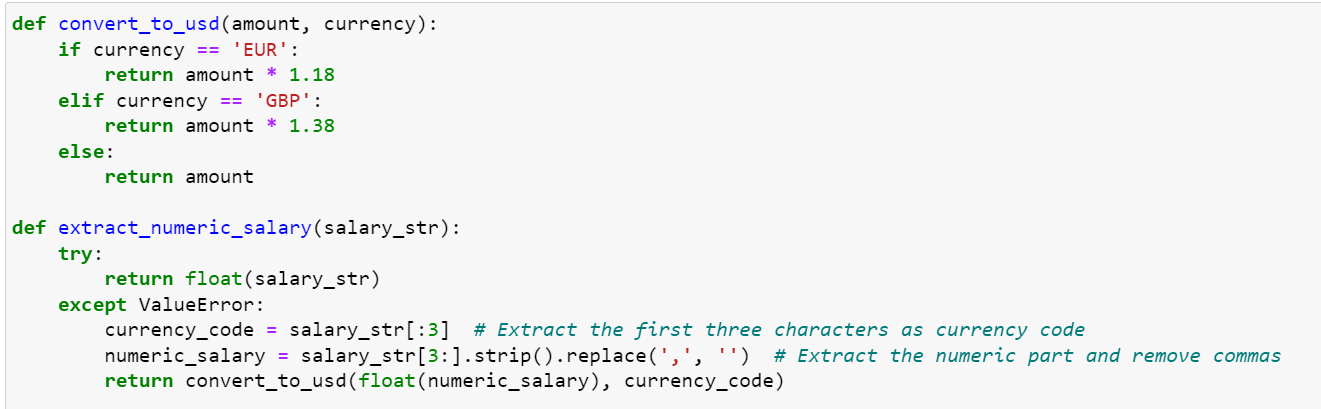


Après avoir vérifié les types de job de valeurs manquantes de la colonne Salary, on se rend compte que la plupart sont des stagiaires. Les stagiaires n’ont généralement pas de salaire. Ce qui semble logique, c’est de remplacer ces valeurs manquantes par 0.

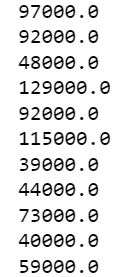


Il n’y a pas que les stagiaires, certains professionnels même sénior ont des valeurs manquantes. Une explication plausible ‘est qu’ils sont en négociation se salaire. Cependant, pour ne pas biaiser l’analyse avec des valeurs approximatives, nous avons décidé de les mettre dans la même catégorie que les stagiaires (Salaire=0).

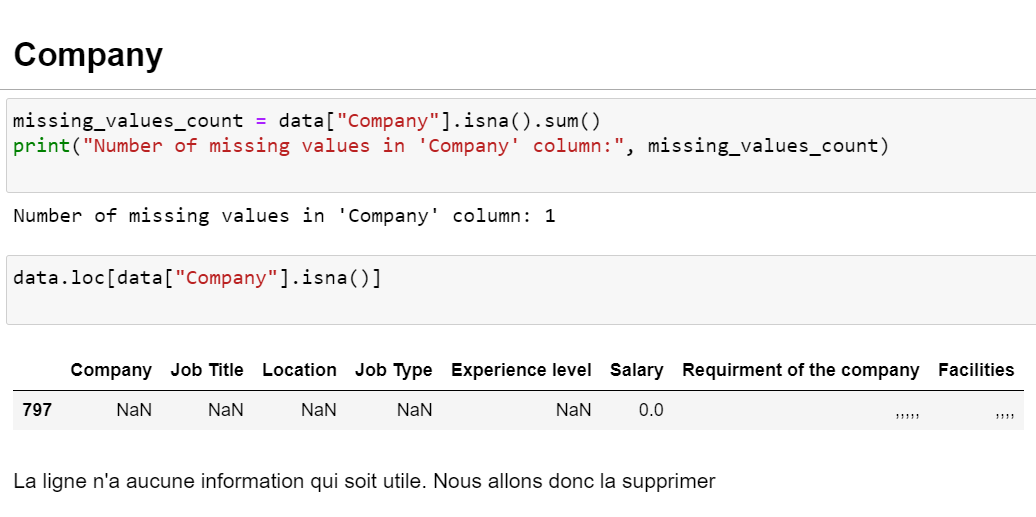




En faisant la conversion, nous avons transformer la colonne en valeur numérique décimale comme présenté dans la figure ci-après :

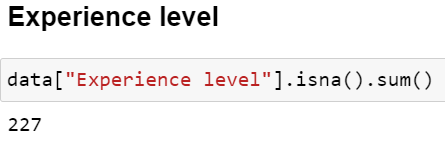


Ensuite, nous nous sommes attaqués à la colonne Company :



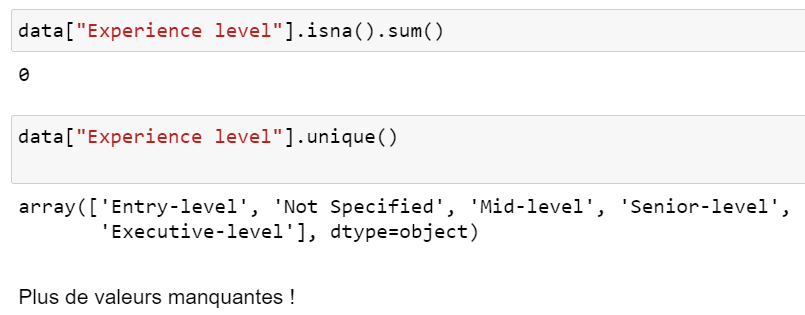
En supprimant la ligne, on se rend compte qu’elle contenait les seules valeurs manquantes des colonnes **Job Title**, **Job Type**, **Location**.

Nous nous attaquons à la colonne « **Experience** **level** » où nous avons remplacé les 227 valeurs manquantes par « **Not** **Specified ».**



Vérifions si tout s’est passé comme souhaité





Pour la colonne **Location**, plus tard dans l’étape de l’analyse, nous avons extrait les noms des pays pour les stocker dans une autre colonne **Country.**

# **Data Analysis**

Cette étape va consister à trouver des informations cachées dans le dataset. Et pour y arriver, nous essayerons de trouver réponse à des questions jugées pertinentes et intéressantes.

L’objectif est de :

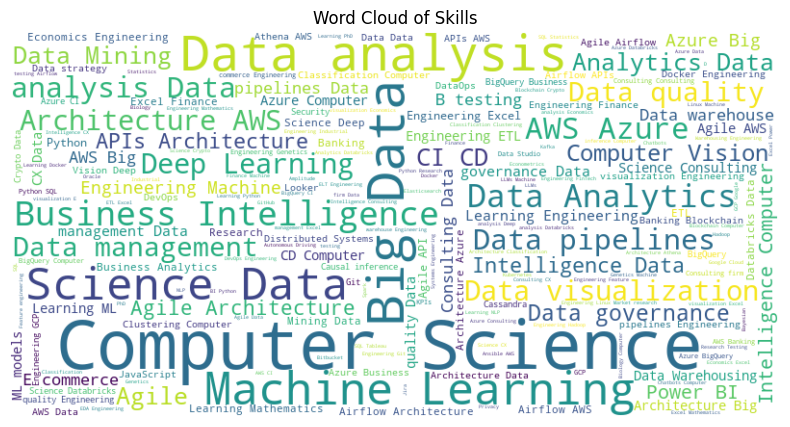
* Identifier les tendances clés du marché de l'emploi en IA, DS et Big Data.
* Mettre en évidence les compétences les plus demandées et les titres de poste les plus courants dans ces domaines.
* Analyser les différences et les similitudes entre les opportunités d'emploi en IA, DS et Big Data.

Pour cela, nous nous sommes posés un certain nombre de questions :

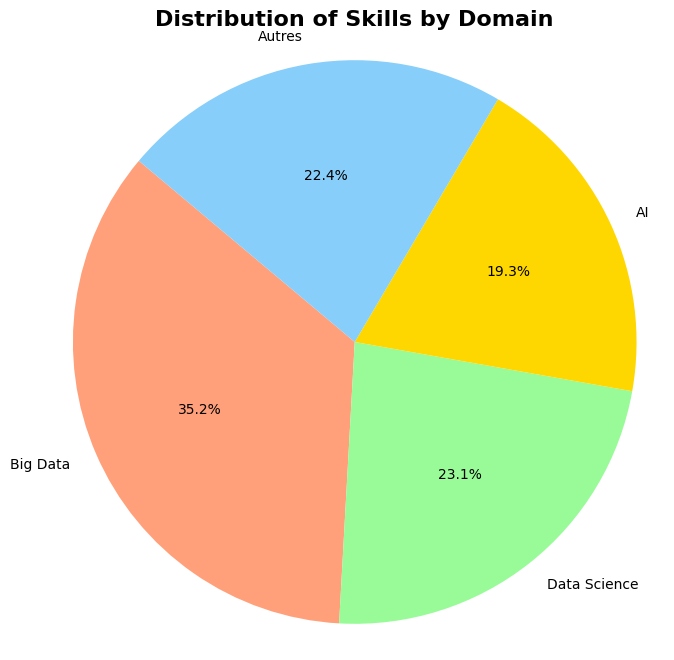
* **Quelles sont les compétences les plus demandées ?**

En affichant le word Cloud de la colonne **Requirment of the company,** on peut facilement se rendre compte que les compétences les plus demandées de façon générales sont :

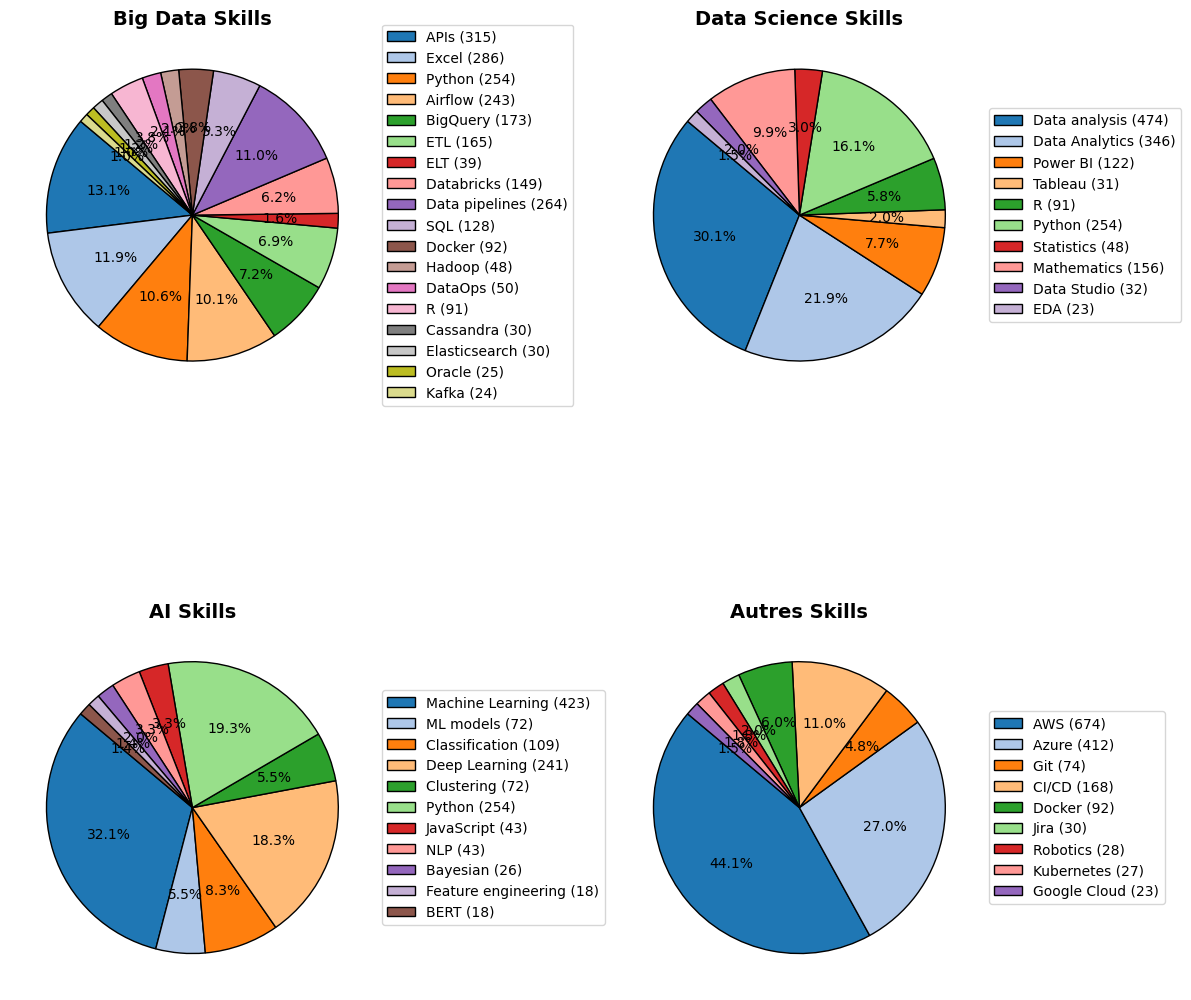
* Computer Science
* Data Analysis
* Big Data
* Machine Learning
* Business Intelligence entre autres

****

Mais lorsqu’on analyse les 100 compétences les plus demandées, on s’aperçoit du pourcentage de chaque domaine.



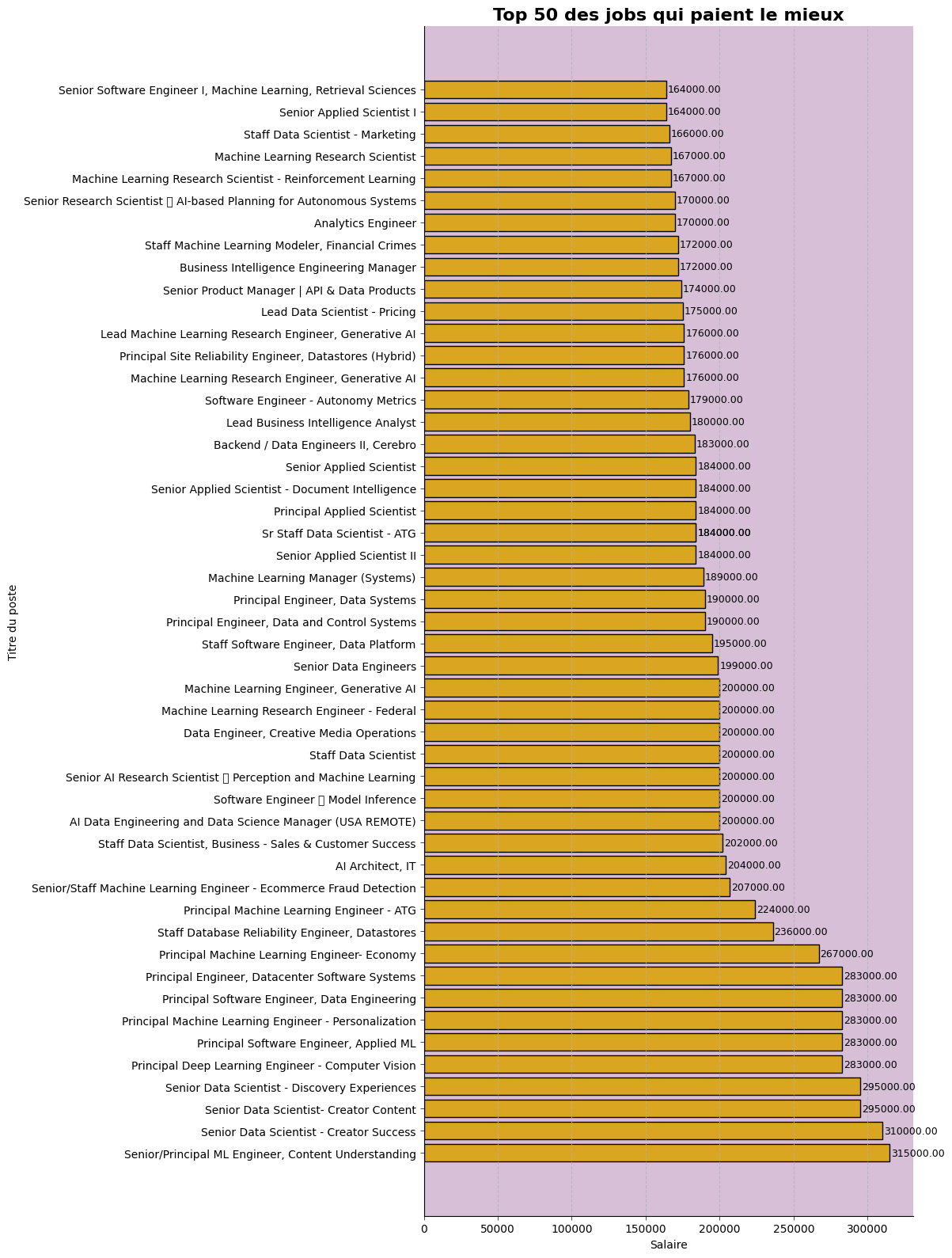
Avec un zoom sur chaque domaine, on détecte les compétences les plus demandées dans chaque domaine :



**Quels sont les jobs les mieux les payés ?**

Nous allons pour cela vérifier les jobs associés aux salaires les plus élevés

Les jobs associés au top 50 des salaires les plus élevés sont répertoriés dans ce graphique



**Dans quels pays se situent les compagnies qui paient le mieux ?**

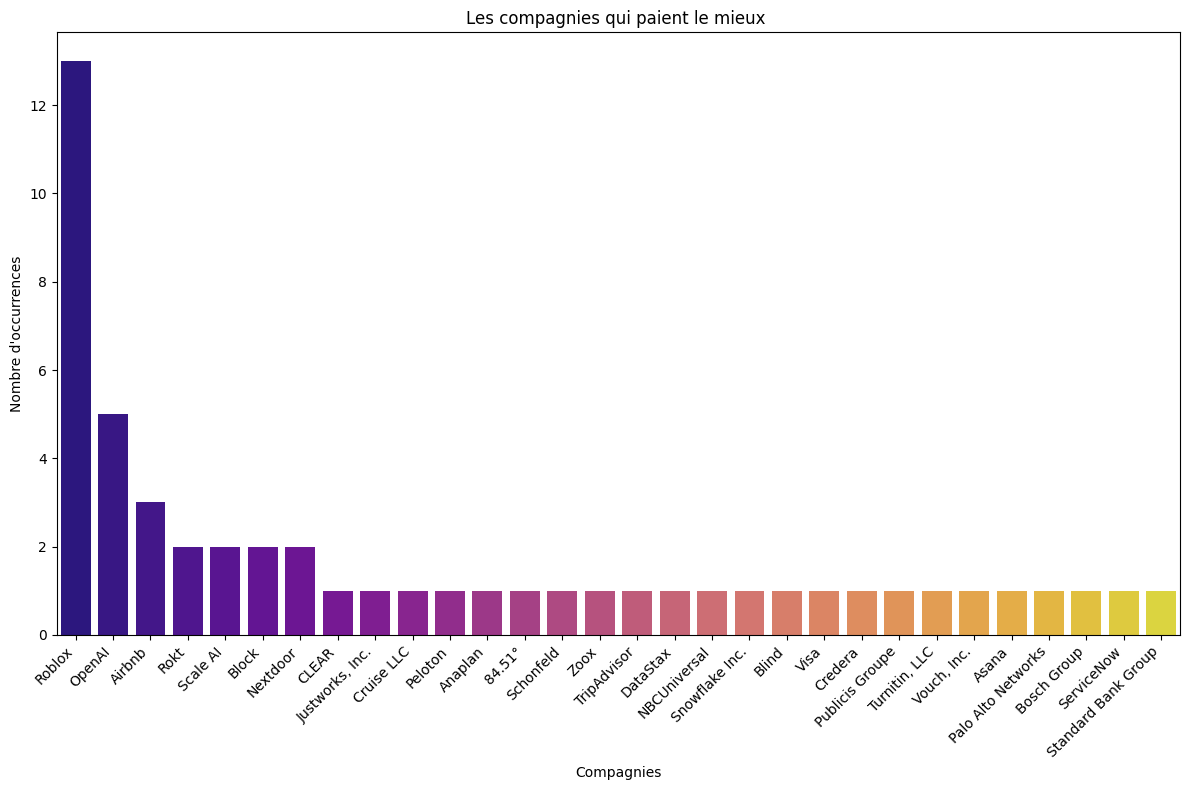
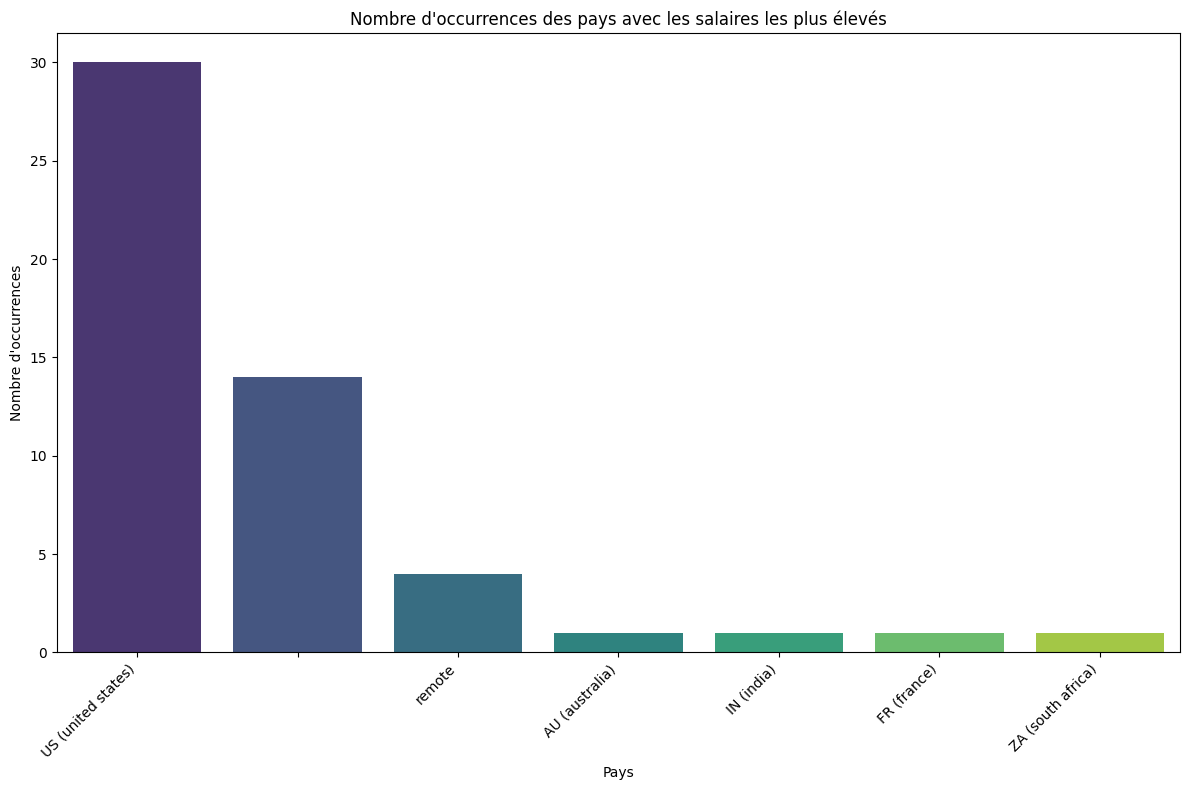
**Quelles sont les compétences recherchées par ces compagnies ?**

Ces compagnies sont entre autres

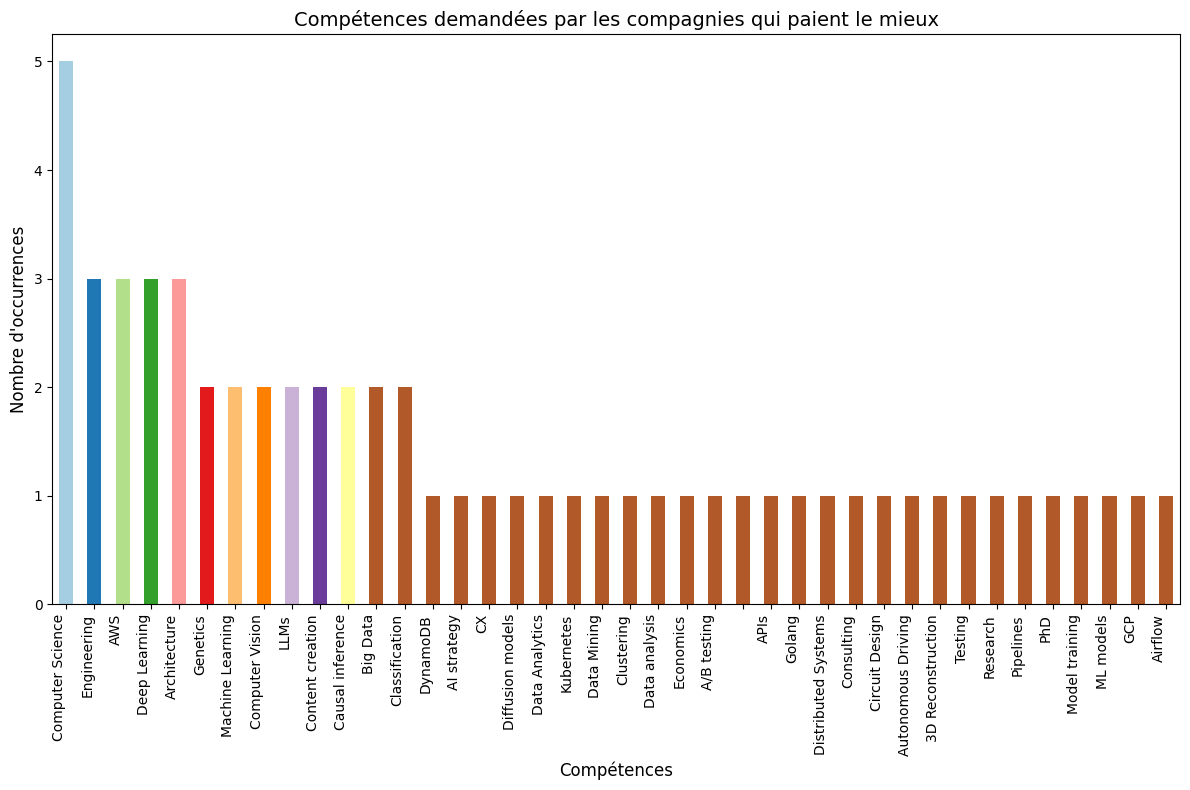
* Roblox
* OpenAI
* Airbnb

Et les pays associés sont entre autres :

* Les Etats-Unis
* L’Australie
* L’Inde
* La France

Les compétences demandées sont les suivantes :



**Quels sont les pays qui paient le mieux ?**

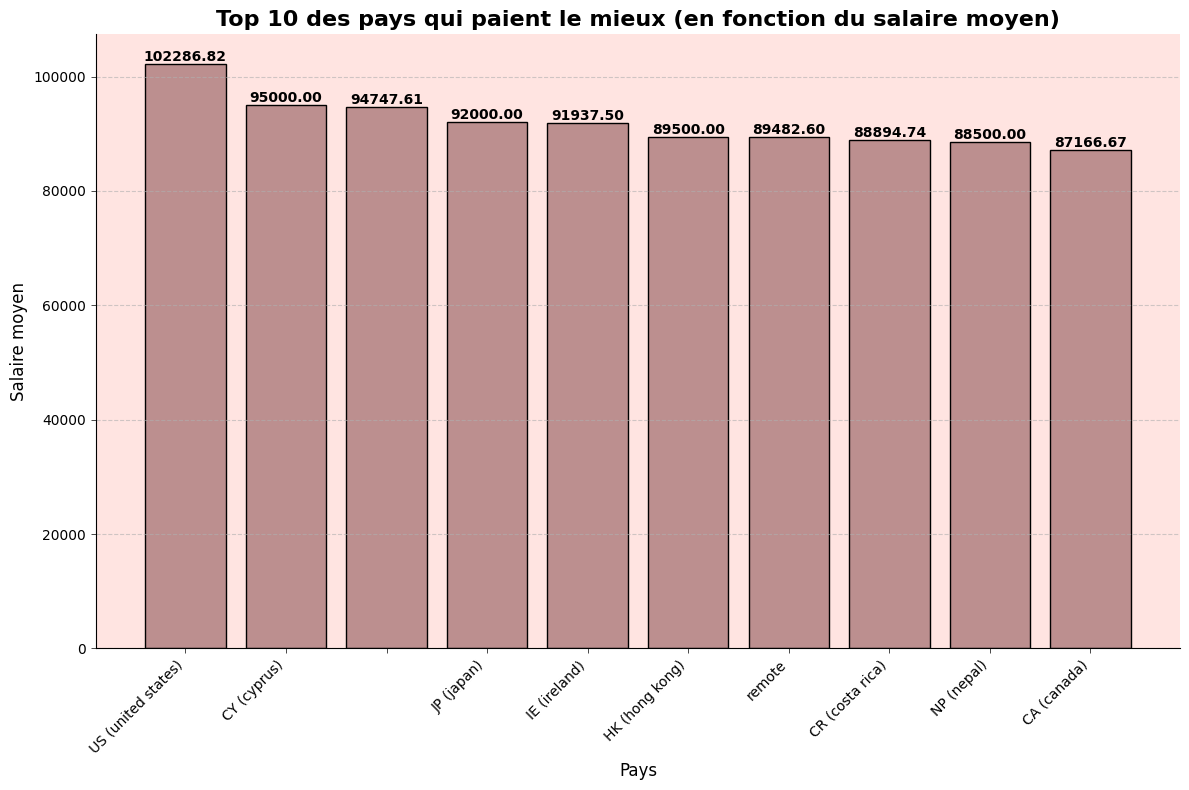
On fait une comparaison des pays selons le salaire moyen, on obtient :



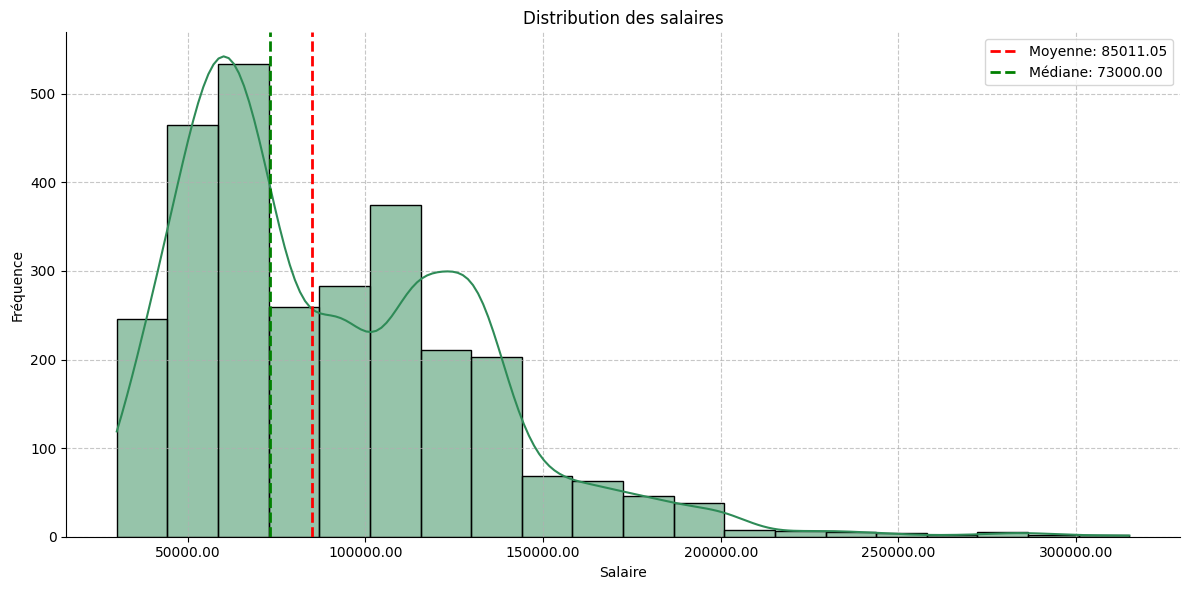
Parmi ces pays, on voit clairement :

* Les Etats-Unis
* Le Japon
* Costa Rica etc

Le top 10 de ces pays sont les suivants :



**La distribution des salaires**

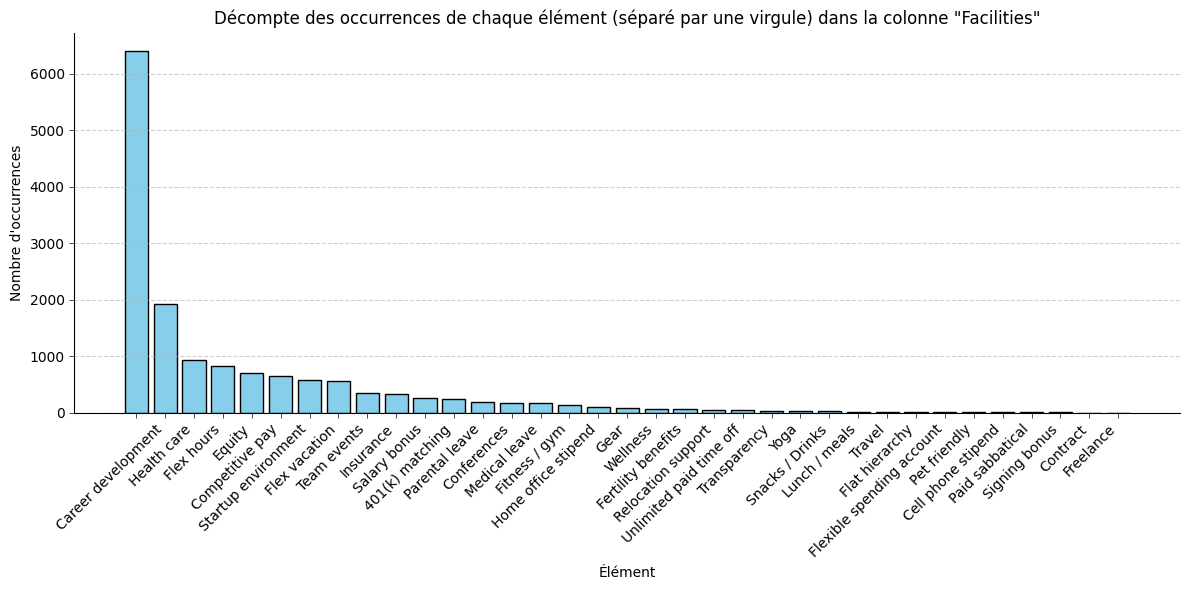


On est ici en présence d'une distribution bimodale. Elle indique que la distribution des salaires n'est pas normale. Deux pics se présentent pour révéler qu'il y a en fait deux catégories de salaires. Les très bons salaires et les salaires moyens. On remarque également que les salariés moyens et ceux qui ont les plus gros salariés sont presqu'à nombre égal.

**Quelles sont les promesses des entreprises ?**

Les promesses les plus fréquentes des entreprises sont exprimées à travers ce graphe parmi lesquelles nous avons :

* career development
* Health care
* Flex hours
* Equity
* Competitive pay
* Startup environment
* Flex vacation



Tout ceci nous permet de conclure qu'en plus du salaire, pour attirer des employés, il faut aussi proposer des possibilités de développement personnel et de carrière.

Nous allons à présent stocker nos données dans une base de données. Pour cela, nous allons procéder en trois étapes (MCD-MLD-MPD)

# **Data Modeling**

* **Modèle Conceptuel de Données**

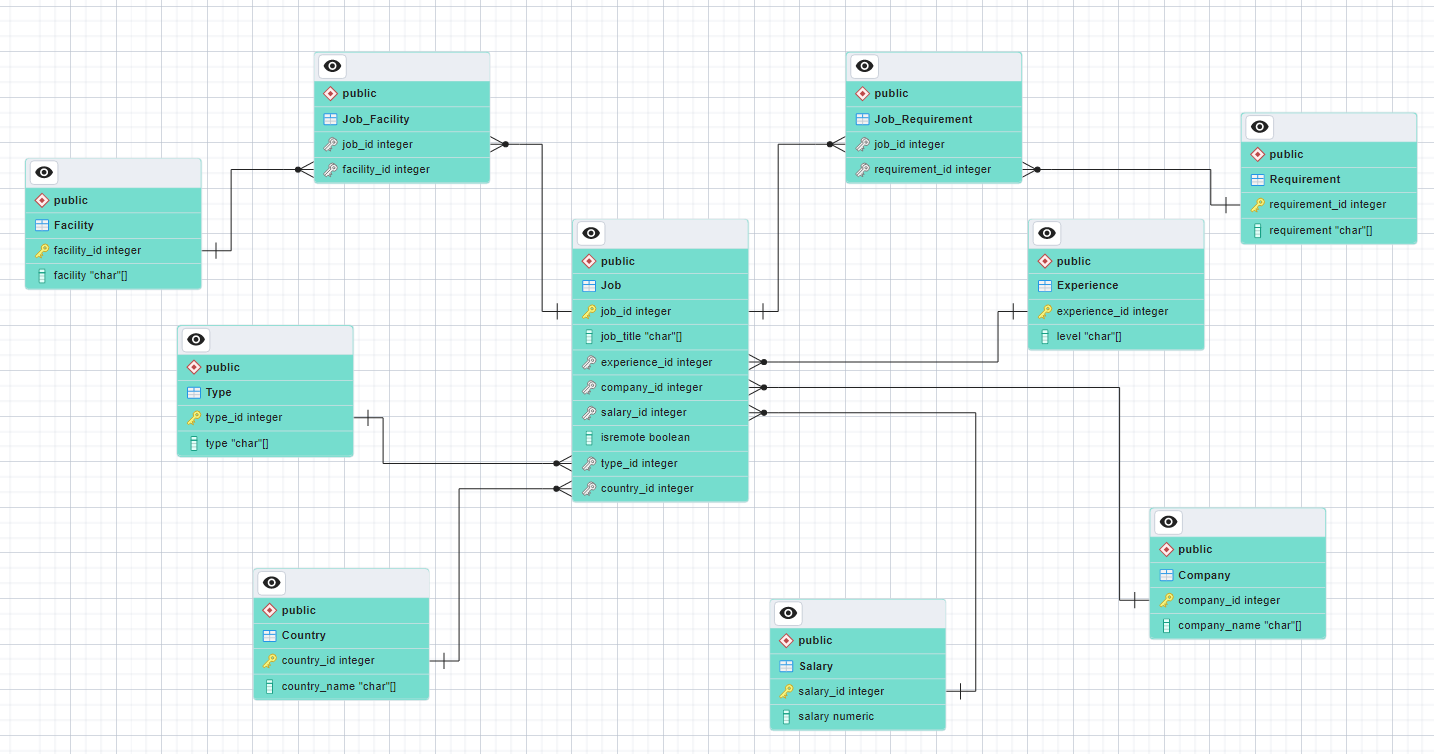
Les entités qui ont été détectées sont les suivantes :

* **Company** (company\_id, company\_name)
* **Job** (job\_id, job\_title, experience\_id, company\_id, type\_id, salary\_id, requirement\_id, isremote, facility\_id, country\_id)
* **Experienc**e (experience\_id, level)
* **Type** (type\_id, type)
* **Salary** (salary\_id, salary)
* **Country** (country\_id, country\_name)
* **Requirement** (requirement\_id, requirement)
* **Facility** (facility\_id, facility)
* **Modèle Logique des Données**
* Une compagnie peut avoir plusieurs offres jobs d'emploi **(one-to-many) (Company\_Job)**
* Un niveau d'expérience peut être associé à plusieurs **jobs (one-to-many) (Job\_Experience)**
* Un pays peut être relié à plusieurs **jobs (one-to-many) (Job\_Country)**
* Un type de job peut être associé à plusieurs jobs **(one-to-many) (Job\_Type)**
* Un salaire peut être associé à plusieurs jobs **(one-to-many) (Job\_Salary)**
* Une compétence peut être associée à plusieurs jobs et plusieurs jobs peuvent être associés à une compétence (**many-to-many) (Job\_Requirement)**
* Un avantage peut associer à plusieurs jobs et inversement **(many-to-many) (Job\_Facility)**

Pour implémenter les relations **many-to-many**, nous allons créer des tables-relations (**Job\_Requiremen**t, **Job\_Facility**)

* **Job\_Requirement** (job\_id, job\_requirement)
* **Job\_Facility** (job\_id, facility\_id)

Ce qui nous permet de réaliser ce diagramme d'entité-relation



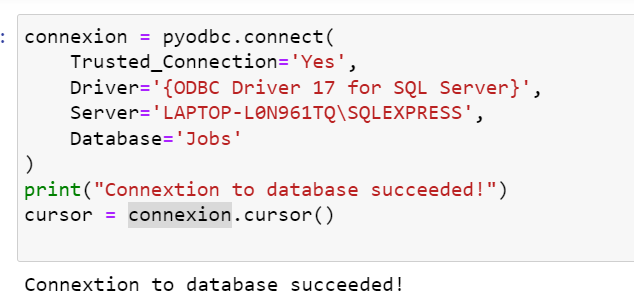
* **Modèle Physique des Données**

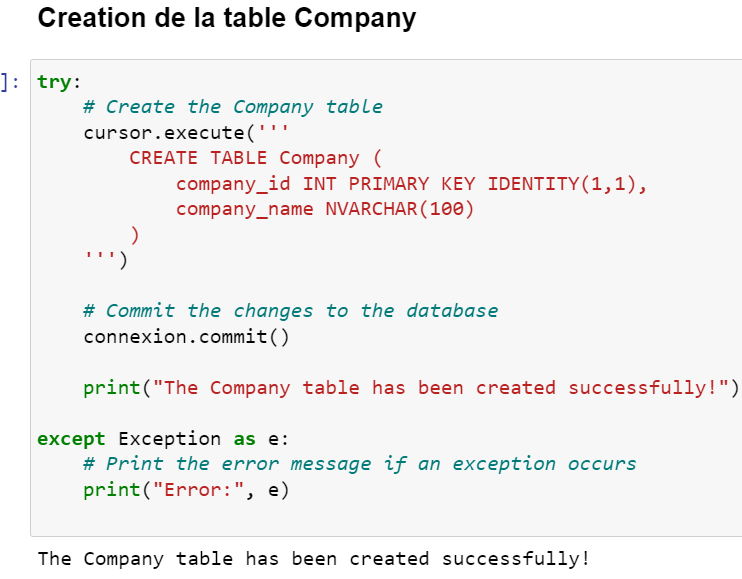
Dans cette dernière étape, nous avons créé la base de données et toute sa structure (tables) puis avons inséré les données correspondantes.

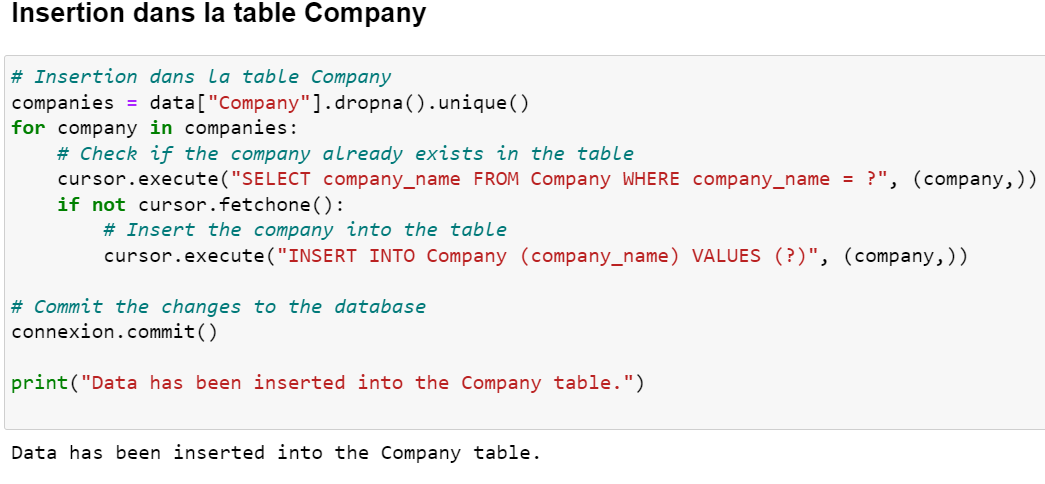
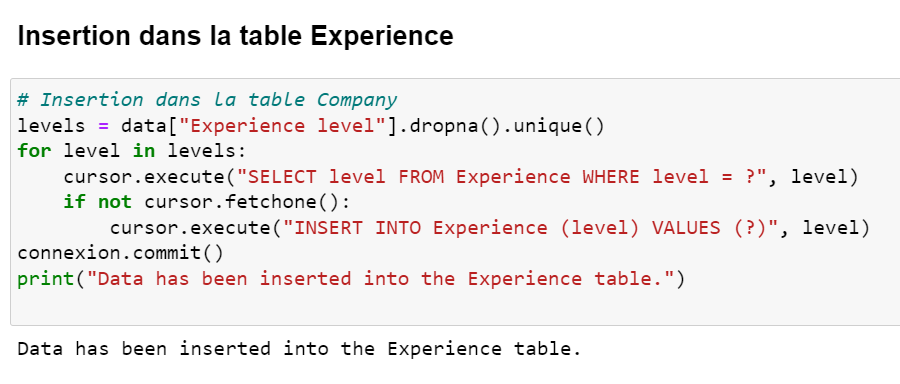
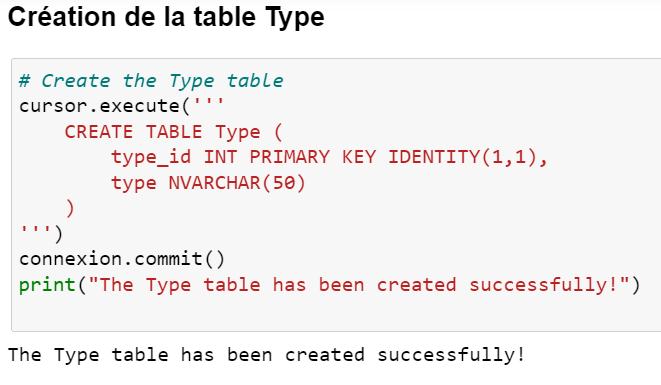
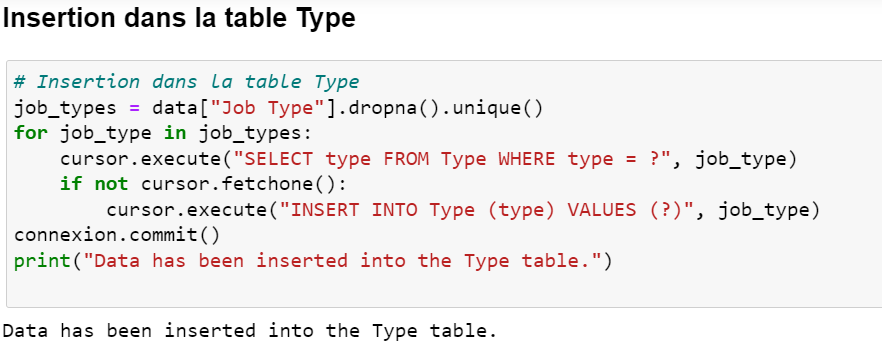
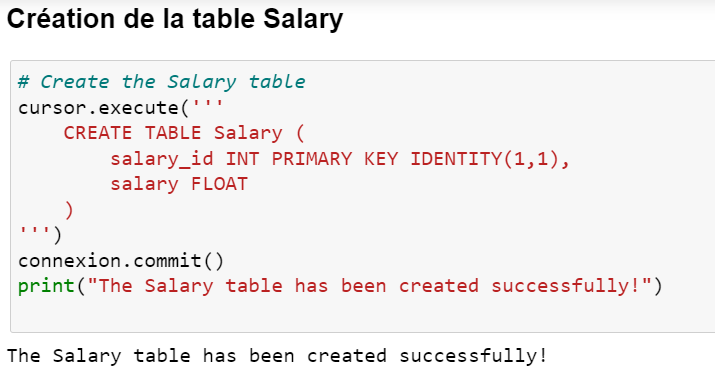
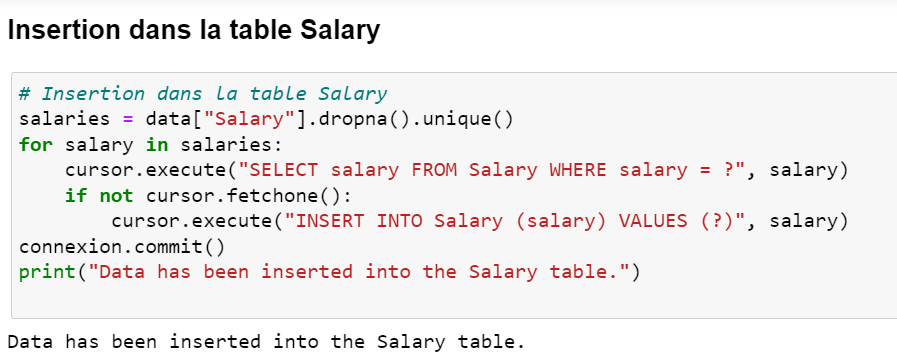
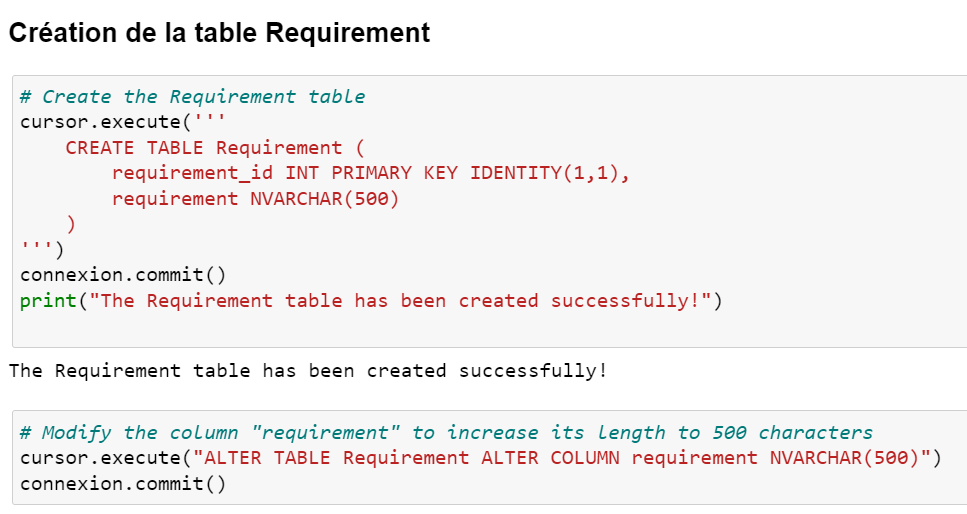
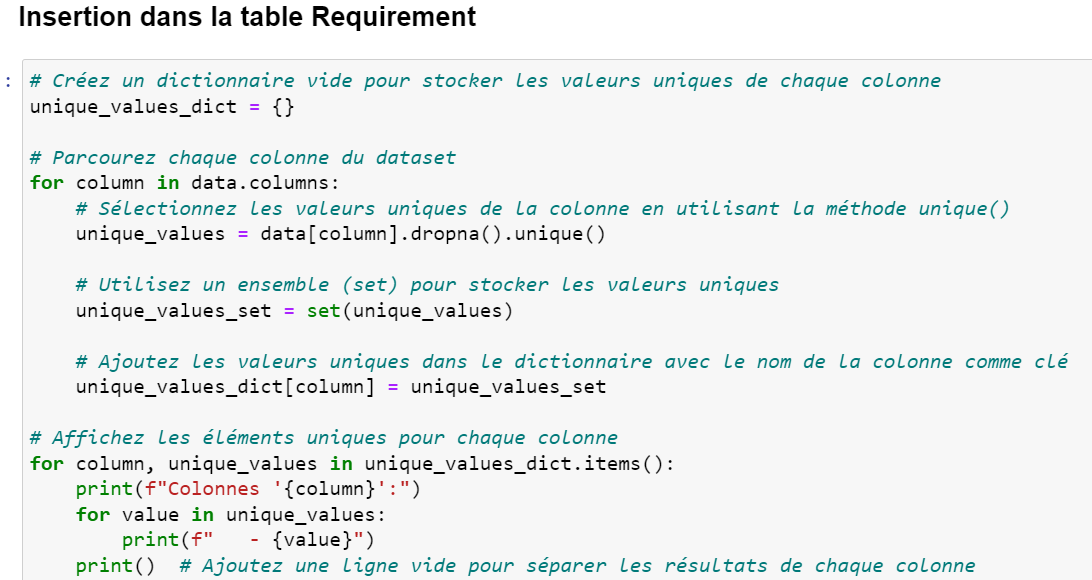
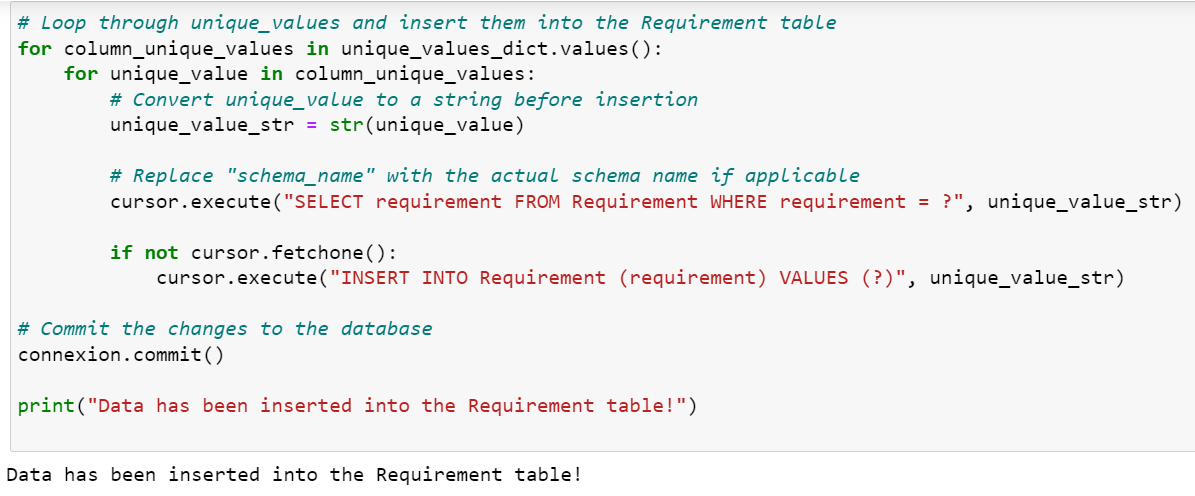
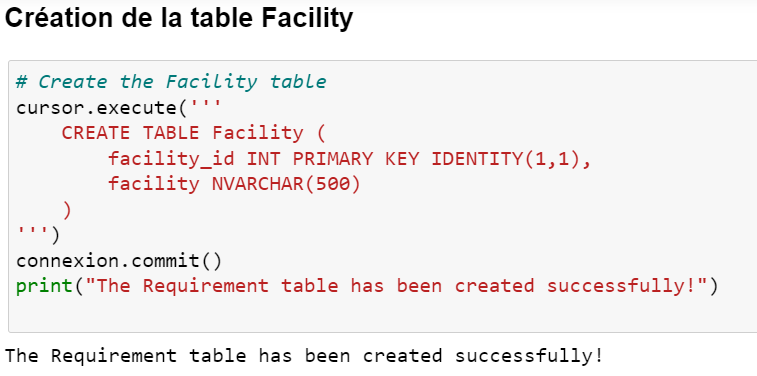
Le SGBD choisi est SQL Server. Pour effectuer la connexion avec la base de

Données, nous avons utilisé **pyodbc.**

La connexion s’est faite grâce au code suivant :





Ainsi de suite pour toutes les autres tables

