Le jeu Palworld et analyse des données

Présentation du jeu Palworld

Palworld est un jeu de survie en monde ouvert développé par Pocket Pair, sorti en janvier 2024. Ce jeu combine des éléments de survie, de crafting, de collection de créatures et de construction de base. Il se déroule dans un monde ouvert peuplé de créatures appelées "Pals" que les joueurs peuvent capturer, élever et utiliser pour diverses tâches.

Les aspects clés du jeu incluent:

Aspect
La capture et la collection de Pals (similaire à Pokémon)
L'utilisation des Pals pour combattre
L'exploitation des Pals pour des tâches comme la construction, l'agriculture, et le travail en usine
La construction et le développement d'un campement
L'exploration d'un monde ouvert avec différentes zones et biomes

Le jeu a connu un immense succès dès sa sortie en raison de son concept innovant mêlant des mécaniques de jeux populaires comme Pokémon et Minecraft, mais avec une approche plus mature et parfois controversée.

📁 Analyse des Fichiers CSV et leur Rôle dans le Jeu

Vue d'ensemble des Données Palworld

Sur la base des descriptions fournies, voici une analyse de chaque fichier de données et son importance dans le jeu :

1. Palworld_Data--Palu combat attribute table.csv

Contenu :

- Statistiques de base (santé, attaque, défense, vitesse)
- Types élémentaires (Feu, Eau, Électricité, etc.)
- Compétences de combat spécifiques

Rôle dans le jeu : Ces données déterminent l'efficacité d'un Pal au combat. Les joueurs utilisent ces informations pour construire des équipes équilibrées et stratégiques pour les combats contre d'autres Pals ou les boss.

2. Palworld_Data--Palu refresh level.csv

Contenu : Informations sur les niveaux auxquels les Pals apparaissent dans différentes zones du jeu.

Rôle dans le jeu : Cette information est cruciale pour les joueurs qui cherchent à capturer des Pals spécifiques. Elle permet de savoir où trouver certains Pals et à quel niveau ils apparaîtront, ce qui affecte leur difficulté de capture et leurs statistiques initiales.

3. Palworld_Data-Palu Job Skills Table.csv

Contenu : Compétences de travail que possèdent les différents Pals :

- Minage
- Bûcheronnage
- Agriculture
- Construction
- Cuisine
- Etc.

Rôle dans le jeu : Ces compétences déterminent quelles tâches un Pal peut effectuer dans votre campement. Un Pal avec une compétence de minage élevée sera efficace pour extraire des minerais, tandis qu'un Pal avec des compétences agricoles sera utile pour cultiver des ressources alimentaires.

4. Palworld_Data-Tower BOSS attribute comparison.csv

Contenu : Attributs des boss spéciaux qui apparaissent dans les tours du jeu.

Rôle dans le jeu : Les tours sont des défis endgame où les joueurs affrontent des boss puissants. Ces données aident à comprendre la puissance relative de ces boss et à préparer des stratégies adaptées pour les combattre.

5. Palworld_Data-comparison of ordinary BOSS attributes.csv

Contenu : Attributs des boss ordinaires que l'on trouve dans le monde ouvert.

Rôle dans le jeu : Ces boss représentent des défis importants durant l'exploration et fournissent des récompenses précieuses. Comprendre leurs attributs permet aux joueurs de se préparer adéquatement à ces affrontements.

6. Palworld_Data-hide pallu attributes.csv

Contenu : Informations sur des attributs cachés des Pals qui ne sont pas immédiatement visibles dans le jeu.

Rôle dans le jeu : Ces attributs cachés peuvent inclure des taux de croissance spécifiques, des affinités particulières, ou des capacités spéciales qui se débloquent dans certaines conditions. Ces informations sont précieuses pour les joueurs qui cherchent à optimiser leurs Pals.

Implications pour l'Analyse

Cette structure de données nous permettra d'analyser :

- **Optimisation des équipes de combat** via les attributs de combat
- **Gestion efficace des ressources** via les compétences de travail
- **Stratégies de capture** via les zones d'apparition et probabilités
- **Progression du jeu** via les niveaux et raretés des Pals

Import des bibliothèques Python de l'étude

In [34]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import mariadb
from sqlalchemy import create_engine
import plotly.express as px
import plotly.graph_objects as go
import os
import csv
from pathlib import Path
```

Chargement des données brutes

In [35]:
```
# Définition du chemin vers les fichiers fournis
data_path = "../database/"

# Vérification de l'existence du repertoire , if not utiliser le repertoire cour
if not os.path.exists(data_path):
    data_path = "./"

# Affichage des fichiers dispo pour la vérif
files = [f for f in os.listdir(data_path) if f.endswith('.csv')] # ici on liste
print("Fichiers dispo:")
for file in files:
    print(f"-{file}")

# Chargement des données brutes & gestion d'erreurs grâce à pd.read_csv en défin
try:
    combat_attribute = pd.read_csv(f'{data_path}Palworld_Data--Palu combat attri
    refresh_area = pd.read_csv(f'{data_path}Palworld_Data--Palu refresh level.cs
    ordinary_boss = pd.read_csv(f'{data_path}Palworld_Data-comparison of ordinar
    tower_boss = pd.read_csv(f'{data_path}Palworld_Data-Tower BOSS attribute com
    job_skill = pd.read_csv(f'{data_path}Palworld_Data-Palu Job Skills Table.csv
    hidden_attribute = pd.read_csv(f'{data_path}Palworld_Data-hide pallu attribu
    print("\nDonnées brutes chargées")
```

```
except FileNotFoundError as e :
  print(f"Erreur: {e}. Vérifier les noms des fichiers et le chemin.")
```

```
Fichiers dispo:
-Palworld_Data--Palu combat attribute table.csv
-Palworld_Data--Palu refresh level.csv
-Palworld_Data-comparison of ordinary BOSS attributes.csv
-Palworld_Data-hide pallu attributes.csv
-Palworld_Data-Palu Job Skills Table.csv
-Palworld_Data-Tower BOSS attribute comparison.csv

Données brutes chargées
```

Création de la fonction d'analyse exploratoires des données fournies

In [36]:
```
def explore_dataframe(df, name):
    # Cette Fonction inclus un affichage optimisé

    # Sauvegarde et réglage des options d'affichage
    pd.set_option('display.max_columns', None)
    pd.set_option('display.max_colwidth', 30)
    pd.set_option('display.width', max(1200, df.shape[1] * 20))
    print(f"\n===== Exploration de {name} =====")

    # 1. Informations de base (similaire à df.info())
    print("\n--- Informations de base ---")
    print(f"Class: {type(df)}")
    # Index
    idx = df.index
    print(f"Index: {type(idx)} | {idx}")
    # Nombre total d'entrées et de colonnes
    print(f"Entries: {len(df)} | Columns: {df.shape[1]}")

    # Détail des colonnes et non-null counts
    structure = pd.DataFrame({
        'Column': df.columns,
        'Non-Null Count': df.notnull().sum().values,
        'Dtype': df.dtypes.values
    })
    display(structure)

    # Usage mémoire
    mem_usage = df.memory_usage(deep=True).sum()
    print(f"Memory usage: {mem_usage / 1024:.2f} KB")

    # 2. Dimensions
    print(f"\nDimensions: {df.shape[0]} lignes, {df.shape[1]} colonnes")

    # Aperçu des premières lignes
    print("\nAperçu des premières lignes:")
    display(df.head(3))

    # Types de données
    print("\nTypes de données:")
    display(df.dtypes)

    # Statistiques descriptives
```

```
print("\nStatistiques descriptives:")
display(df.describe(include='all'))

# Valeurs manquantes
print("\nValeurs manquantes:")
missing = df.isnull().sum()
missing_percent = (missing / len(df)) * 100
missing_data = pd.concat([missing, missing_percent], axis=1, keys=['Total',
display(missing_data[missing_data['Total'] > 0])
if missing_data['Total'].sum() == 0:
    print("Aucune valeur manquante détectée")

# Doublons
duplicates = df.duplicated().sum()
print(f"\nNombre de lignes dupliquées: {duplicates}")

# Colonnes catégorielles
print("\nAnalyse des colonnes catégorielles:")
categorical_cols = df.select_dtypes(include=['object']).columns

if len(categorical_cols) > 0:
    for col in categorical_cols:
        unique_values = df[col].unique()
        n_unique = len(unique_values)

        print(f"\n{col}:")
        print(f"  - {n_unique} valeurs uniques")

        if n_unique <= 10:
            print(f"  - Valeurs: {unique_values}")
        elif n_unique <= 20:
            value_counts = df[col].value_counts().head(10)
            print(f"  - Top 10 des valeurs les plus fréquentes:")
            for val, count in value_counts.items():
                print(f"    • {val}: {count} occurrences")
        else:
            value_counts = df[col].value_counts()
            print(f"  - Top 5 des valeurs les plus fréquentes:")
            for val, count in value_counts.head(5).items():
                print(f"    • {val}: {count} occurrences")
            print(f"  - Valeur la moins fréquente: {value_counts.tail(1).ind
else:
    print("Aucune colonne catégorielle détectée")
# Détection des valeurs aberrantes
print("\nValeurs aberrantes (méthode IQR):")
numeric_cols = df.select_dtypes(include=[np.number]).columns


    # Détection des valeurs aberrantes
print("\nValeurs aberrantes (méthode IQR):")
numeric_cols = df.select_dtypes(include=[np.number]).columns

for col in numeric_cols:
    Q1 = df[col].quantile(0.25)
    Q3 = df[col].quantile(0.75)
    IQR = Q3 - Q1
    outliers = df[(df[col] < Q1 - 1.5*IQR) | (df[col] > Q3 + 1.5*IQR)][col]

    print(f"{col}: {len(outliers)} valeurs aberrantes ({len(outliers)/len(df
```

```
        # AFFICHAGE DES VALEURS ABERRANTES
        if len(outliers) > 0:
            if len(outliers) <= 10:
                print(f"  Valeurs: {sorted(outliers.tolist())}")
            else:
                print(f"  Premières 10: {sorted(outliers.tolist())[:10]}")
    # Rétablissement des options d'affichage
    pd.reset_option('display.max_columns')
    pd.reset_option('display.max_info_columns')
    pd.reset_option('display.max_colwidth')
    pd.reset_option('display.width')

    # Séparateur
    print("\n" + "="*70)
```

Analyse exploratoire des données fournies

In [37]:
```
explore_dataframe(job_skill, 'Job Skills')
```

```
===== Exploration de Job Skills =====

--- Informations de base ---
Class: <class 'pandas.core.frame.DataFrame'>
Index: <class 'pandas.core.indexes.range.RangeIndex'> | RangeIndex(start=0, stop=
139, step=1)
Entries: 139 | Columns: 23
```

	Column	Non-Null Count	Dtype
0	Related Links Directory Pa...	139	object
1	Unnamed: 1	139	object
2	Unnamed: 2	139	object
3	Unnamed: 3	139	object
4	Unnamed: 4	139	object
5	Unnamed: 5	26	object
6	Unnamed: 6	139	object
7	Unnamed: 7	139	object
8	Unnamed: 8	139	object
9	Unnamed: 9	139	object
10	Unnamed: 10	139	object
11	Unnamed: 11	139	object
12	Unnamed: 12	139	object
13	Unnamed: 13	139	object
14	Unnamed: 14	139	object
15	Unnamed: 15	139	object
16	Unnamed: 16	139	object
17	Unnamed: 17	139	object
18	Unnamed: 18	139	object
19	Unnamed: 19	57	object
20	Unnamed: 20	14	object
21	Unnamed: 21	14	object
22	Unnamed: 22	14	object

```
Memory usage: 170.69 KB
```

```
Dimensions: 139 lignes, 23 colonnes
```

```
Aperçu des premières lignes:
```

Related Links Directory Palu Combat Attribute Table Food BUFF Work Disease DEBUFF	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnar
0	ID	English name	Chinese name	Volume size	Food intake	night shift	Total skills	Ma
1	1	Lamball	Mian Youyou	smallest	2	NaN	3	
2	2	Cattiva	Naughty cat	smallest	2	NaN	4	

Types de données:

```
Related Links Directory Palu Combat Attribute Table Food BUFF Work Disease DEBUFF
object
Unnamed: 1
object
Unnamed: 2
object
Unnamed: 3
object
Unnamed: 4
object
Unnamed: 5
object
Unnamed: 6
object
Unnamed: 7
object
Unnamed: 8
object
Unnamed: 9
object
Unnamed: 10
object
Unnamed: 11
object
Unnamed: 12
object
Unnamed: 13
object
Unnamed: 14
object
Unnamed: 15
object
Unnamed: 16
object
Unnamed: 17
object
Unnamed: 18
object
Unnamed: 19
object
Unnamed: 20
object
Unnamed: 21
object
Unnamed: 22
object
dtype: object
Statistiques descriptives:
```

	Related Links Directory Palu Combat Attribute Table Food BUFF Work Disease DEBUFF	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6
count	139	139	139	139	139	26	139
unique	112	139	138	6	10	2	13
top	12	English name	Thunderbird	big	3	yes	3
freq	2	1	2	43	26	25	27

Valeurs manquantes:

	Total	Pourcentage
Unnamed: 5	113	81.294964
Unnamed: 19	82	58.992806
Unnamed: 20	125	89.928058
Unnamed: 21	125	89.928058
Unnamed: 22	125	89.928058

```
Nombre de lignes dupliquées: 0

Analyse des colonnes catégorielles:

Related Links Directory Palu Combat Attribute Table Food BUFF Work Disease DEBUF
F:
  - 112 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • 12: 2 occurrences
    • 13: 2 occurrences
    • 37: 2 occurrences
    • 40: 2 occurrences
    • 45: 2 occurrences
  - Valeur la moins fréquente: 111 (1 occurrence(s))

Unnamed: 1:
  - 139 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • English name: 1 occurrences
    • Lamball: 1 occurrences
    • Cattiva: 1 occurrences
    • Chikipi: 1 occurrences
    • Lifmunk: 1 occurrences
  - Valeur la moins fréquente: Jetragon (1 occurrence(s))

Unnamed: 2:
  - 138 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • Thunderbird: 2 occurrences
    • Chinese name: 1 occurrences
    • Naughty cat: 1 occurrences
    • Pipi Chicken: 1 occurrences
    • green leaf rat: 1 occurrences
  - Valeur la moins fréquente: Vortex Dragon (1 occurrence(s))

Unnamed: 3:
  - 6 valeurs uniques
  - Valeurs: ['Volume size' 'smallest' 'Small' 'big' 'medium' 'maximum']

Unnamed: 4:
  - 10 valeurs uniques
  - Valeurs: ['Food intake' '2' '1' '3' '8' '5' '4' '7' '6' '9']

Unnamed: 5:
  - 3 valeurs uniques
  - Valeurs: ['night shift' nan 'yes']

Unnamed: 6:
  - 13 valeurs uniques
  - Top 10 des valeurs les plus fréquentes:
    • 3: 27 occurrences
    • 1: 22 occurrences
    • 4: 20 occurrences
    • 2: 19 occurrences
    • 5: 14 occurrences
    • 6: 13 occurrences
    • 9: 7 occurrences
    • 8: 5 occurrences
    • 10: 5 occurrences
    • 7: 3 occurrences
```

Unnamed: 7:
- 6 valeurs uniques
- Valeurs: ['Make a fire' '0' '1' '2' '3' '4']

Unnamed: 8:
- 6 valeurs uniques
- Valeurs: ['watering' '0' '1' '2' '3' '4']

Unnamed: 9:
- 6 valeurs uniques
- Valeurs: ['planting' '0' '1' '2' '3' '4']

Unnamed: 10:
- 6 valeurs uniques
- Valeurs: ['generate electricity' '0' '1' '2' '3' '4']

Unnamed: 11:
- 6 valeurs uniques
- Valeurs: ['manual' '1' '0' '2' '3' '4']

Unnamed: 12:
- 6 valeurs uniques
- Valeurs: ['collection' '0' '1' '2' '3' '4']

Unnamed: 13:
- 5 valeurs uniques
- Valeurs: ['logging' '0' '1' '2' '3']

Unnamed: 14:
- 6 valeurs uniques
- Valeurs: ['Mining' '0' '1' '2' '3' '4']

Unnamed: 15:
- 5 valeurs uniques
- Valeurs: ['pharmaceutical' '0' '1' '2' '3']

Unnamed: 16:
- 6 valeurs uniques
- Valeurs: ['cool down' '0' '1' '2' '3' '4']

Unnamed: 17:
- 3 valeurs uniques
- Valeurs: ['pasture' '1' '0']

Unnamed: 18:
- 6 valeurs uniques
- Valeurs: ['carry' '1' '0' '2' '3' '4']

Unnamed: 19:
- 33 valeurs uniques
- Top 5 des valeurs les plus fréquentes:
  - 250: 6 occurrences
  - 270: 4 occurrences
  - 275: 4 occurrences
  - 150: 4 occurrences
  - 320: 3 occurrences
- Valeur la moins fréquente: 500 (1 occurrence(s))

Unnamed: 20:

```
  - 12 valeurs uniques
  - Top 10 des valeurs les plus fréquentes:
    • wool: 3 occurrences
    • gold: 2 occurrences
    • ranch items: 1 occurrences
    • Egg: 1 occurrences
    • Palu Ball Advanced Palu Ball Arrow Gold Coin: 1 occurrences
    • milk: 1 occurrences
    • Marshmallow: 1 occurrences
    • red wild berries: 1 occurrences
    • Honey: 1 occurrences
    • fire breathing organ: 1 occurrences

Unnamed: 21:
  - 6 valeurs uniques
  - Valeurs: ['pasture minimum output' '1' nan
 'According to partner skill level 1,1,2,1,1 0,0,0,1,1 1,1,2,3,4 10,20,30,40,50'
 '10' '2']

Unnamed: 22:
  - 6 valeurs uniques
  - Valeurs: ['The largest ranch (Rank = partner skill level)' 'Rank' nan
 'According to partner skill level 1,2,3,1,2 0,0,0,1,1 1,2,3,4,5 10,20,30,40,50'
 'Rank*10' '1+Rank']

Valeurs aberrantes (méthode IQR):

Valeurs aberrantes (méthode IQR):

======================================================================
```

In [215…

```
explore_dataframe(refresh_area, 'Refresh area')
```

```
===== Exploration de Refresh area =====

--- Informations de base ---
Class: <class 'pandas.core.frame.DataFrame'>
Index: <class 'pandas.core.indexes.range.RangeIndex'> | RangeIndex(start=0, stop=
149, step=1)
Entries: 149 | Columns: 19
```

	Column	Non-Null Count	Dtype
0	PS: This table only refres...	138	object
1	Unnamed: 1	137	object
2	Unnamed: 2	137	object
3	Unnamed: 3	137	object
4	Unnamed: 4	0	float64
5	Unnamed: 5	147	object
6	Unnamed: 6	146	object
7	Unnamed: 7	146	object
8	Unnamed: 8	146	object
9	Unnamed: 9	146	object
10	Unnamed: 10	18	object
11	Unnamed: 11	146	object
12	Unnamed: 12	0	float64
13	Unnamed: 13	141	object
14	Unnamed: 14	140	object
15	Unnamed: 15	140	object
16	Unnamed: 16	140	object
17	Unnamed: 17	7	object
18	Unnamed: 18	140	object

```
Memory usage: 146.81 KB
```

```
Dimensions: 149 lignes, 19 colonnes
```

```
Aperçu des premières lignes:
```

	PS: This table only refreshes the data and does not include all occurrences of Palu. For example, the level 10 Balrog Sheep and Mammoth fights that appear in Maple Leaf Forest are random events. pending upgrade	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Un
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
1	Pallu refresh level reference	NaN	NaN	NaN	NaN	Minimum appearance level o...	NaN	
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

Types de données:

```
PS: This table only refreshes the data and does not include all occurrences of Pa
lu. For example, the level 10 Balrog Sheep and Mammoth fights that appear in Mapl
e Leaf Forest are random events. pending upgrade     object
Unnamed: 1
object
Unnamed: 2
object
Unnamed: 3
object
Unnamed: 4
float64
Unnamed: 5
object
Unnamed: 6
object
Unnamed: 7
object
Unnamed: 8
object
Unnamed: 9
object
Unnamed: 10
object
Unnamed: 11
object
Unnamed: 12
float64
Unnamed: 13
object
Unnamed: 14
object
Unnamed: 15
object
Unnamed: 16
object
Unnamed: 17
object
Unnamed: 18
object
dtype: object
Statistiques descriptives:
```

	PS: This table only refreshes the data and does not include all occurrences of Palu. For example, the level 10 Balrog Sheep and Mammoth fights that appear in Maple Leaf Forest are random events. pending upgrade	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unname
count	138	137	137	137	0.0	147	
unique	113	136	40	31	NaN	113	
top	12	Thunderbird	2	45	NaN	32	Marshma
freq	2	2	12	31	NaN	3	
mean	NaN	NaN	NaN	NaN	NaN	NaN	N
std	NaN	NaN	NaN	NaN	NaN	NaN	N
min	NaN	NaN	NaN	NaN	NaN	NaN	N
25%	NaN	NaN	NaN	NaN	NaN	NaN	N
50%	NaN	NaN	NaN	NaN	NaN	NaN	N
75%	NaN	NaN	NaN	NaN	NaN	NaN	N
max	NaN	NaN	NaN	NaN	NaN	NaN	N

```
Valeurs manquantes:
```

	Total	Pourcentage
PS: This table only refreshes the data and does not include all occurrences of Palu. For example, the level 10 Balrog Sheep and Mammoth fights that appear in Maple Leaf Forest are random events. pending upgrade	11	7.382550
Unnamed: 1	12	8.053691
Unnamed: 2	12	8.053691
Unnamed: 3	12	8.053691
Unnamed: 4	149	100.000000
Unnamed: 5	2	1.342282
Unnamed: 6	3	2.013423
Unnamed: 7	3	2.013423
Unnamed: 8	3	2.013423
Unnamed: 9	3	2.013423
Unnamed: 10	131	87.919463
Unnamed: 11	3	2.013423
Unnamed: 12	149	100.000000
Unnamed: 13	8	5.369128
Unnamed: 14	9	6.040268
Unnamed: 15	9	6.040268
Unnamed: 16	9	6.040268
Unnamed: 17	142	95.302013
Unnamed: 18	9	6.040268

Nombre de lignes dupliquées: 1

Analyse des colonnes catégorielles:

PS: This table only refreshes the data and does not include all occurrences of Palu. For example, the level 10 Balrog Sheep and Mammoth fights that appear in Maple Leaf Forest are random events. pending upgrade:
- 114 valeurs uniques
- Top 5 des valeurs les plus fréquentes:
  - 12: 2 occurrences
  - 13: 2 occurrences
  - 32: 2 occurrences
  - 37: 2 occurrences
  - 40: 2 occurrences
- Valeur la moins fréquente: 111 (1 occurrence(s))

Unnamed: 1:
- 137 valeurs uniques
- Top 5 des valeurs les plus fréquentes:
  - Thunderbird: 2 occurrences
  - name: 1 occurrences
  - Naughty cat: 1 occurrences
  - Mian Youyou: 1 occurrences
  - green leaf rat: 1 occurrences
- Valeur la moins fréquente: Vortex Dragon (1 occurrence(s))

Unnamed: 2:
- 41 valeurs uniques
- Top 5 des valeurs les plus fréquentes:
  - 2: 12 occurrences
  - 40: 10 occurrences
  - 18: 8 occurrences
  - 3: 7 occurrences
  - 30: 7 occurrences
- Valeur la moins fréquente: 49 (1 occurrence(s))

Unnamed: 3:
- 32 valeurs uniques
- Top 5 des valeurs les plus fréquentes:
  - 45: 31 occurrences
  - 40: 15 occurrences
  - 38: 14 occurrences
  - 13: 13 occurrences
  - 29: 11 occurrences
- Valeur la moins fréquente: 48 (1 occurrence(s))

Unnamed: 5:
- 114 valeurs uniques
- Top 5 des valeurs les plus fréquentes:
  - 32: 3 occurrences
  - 88: 3 occurrences
  - 58: 3 occurrences
  - 13: 2 occurrences
  - 12: 2 occurrences
- Valeur la moins fréquente: 111 (1 occurrence(s))

Unnamed: 6:
- 137 valeurs uniques
- Top 5 des valeurs les plus fréquentes:
  - Marshmallow: 2 occurrences

```
    • Pa Pa Pa Catfish: 2 occurrences
    • fire unicorn: 2 occurrences
    • knight bee: 2 occurrences
    • Hanging spirit: 2 occurrences
  - Valeur la moins fréquente: Vortex Dragon (1 occurrence(s))

Unnamed: 7:
  - 41 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • 2: 13 occurrences
    • 40: 10 occurrences
    • 3: 9 occurrences
    • 18: 8 occurrences
    • 30: 7 occurrences
  - Valeur la moins fréquente: 49 (1 occurrence(s))

Unnamed: 8:
  - 137 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • 895: 2 occurrences
    • 1070: 2 occurrences
    • 1420: 2 occurrences
    • 1190: 2 occurrences
    • 1390: 2 occurrences
  - Valeur la moins fréquente: 90 (1 occurrence(s))

Unnamed: 9:
  - 7 valeurs uniques
  - Valeurs: [nan 'Pallu refresh type' 'Creeps' 'Secret Domain BOSS' 'event'
 'Random dungeon boss' 'Wild BOSS']

Unnamed: 10:
  - 3 valeurs uniques
  - Valeurs: [nan 'Night only' 'yes']

Unnamed: 11:
  - 12 valeurs uniques
  - Top 10 des valeurs les plus fréquentes:
    • grassland: 49 occurrences
    • dungeon: 18 occurrences
    • islands: 16 occurrences
    • game reserve: 14 occurrences
    • volcano: 13 occurrences
    • Snow: 11 occurrences
    • forest: 9 occurrences
    • desert: 8 occurrences
    • Random events: 4 occurrences
    • forest snow: 3 occurrences

Unnamed: 13:
  - 114 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • 85: 3 occurrences
    • 12: 2 occurrences
    • 13: 2 occurrences
    • 37: 2 occurrences
    • 40: 2 occurrences
  - Valeur la moins fréquente: 111 (1 occurrence(s))

Unnamed: 14:
```

```
  - 137 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • surf duck: 2 occurrences
    • Thunderbird: 2 occurrences
    • Tianyulong: 2 occurrences
    • Pecorone: 2 occurrences
    • Pipi Chicken: 1 occurrences
  - Valeur la moins fréquente: Vortex Dragon (1 occurrence(s))

Unnamed: 15:
  - 32 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • 45: 32 occurrences
    • 40: 15 occurrences
    • 13: 14 occurrences
    • 38: 14 occurrences
    • 29: 11 occurrences
  - Valeur la moins fréquente: 48 (1 occurrence(s))

Unnamed: 16:
  - 7 valeurs uniques
  - Valeurs: [nan 'Pallu refresh type' 'Creeps' 'Random dungeon boss' 'Wild BOSS'
 'event' 'Secret Domain BOSS']

Unnamed: 17:
  - 3 valeurs uniques
  - Valeurs: [nan 'Night only' 'yes']

Unnamed: 18:
  - 11 valeurs uniques
  - Top 10 des valeurs les plus fréquentes:
    • dungeon: 74 occurrences
    • grassland: 19 occurrences
    • game reserve: 18 occurrences
    • Snow: 7 occurrences
    • forest: 6 occurrences
    • Random events: 4 occurrences
    • desert: 4 occurrences
    • forest snow: 4 occurrences
    • volcano: 3 occurrences
    • refresh area: 1 occurrences

Valeurs aberrantes (méthode IQR):

Valeurs aberrantes (méthode IQR):
Unnamed: 4: 0 valeurs aberrantes (0.0%)
Unnamed: 12: 0 valeurs aberrantes (0.0%)

======================================================================
```

In [216…

```
explore_dataframe(ordinary_boss, 'Ordinary Boss')
```

```
===== Exploration de Ordinary Boss =====

--- Informations de base ---
Class: <class 'pandas.core.frame.DataFrame'>
Index: <class 'pandas.core.indexes.range.RangeIndex'> | RangeIndex(start=0, stop=
117, step=1)
Entries: 117 | Columns: 8
```

	Column	Non-Null Count	Dtype
0	The BOSS version of each P...	12	object
1	Unnamed: 1	11	object
2	Unnamed: 2	0	float64
3	Unnamed: 3	7	object
4	Unnamed: 4	7	object
5	Unnamed: 5	0	float64
6	Unnamed: 6	115	object
7	Unnamed: 7	115	object

```
Memory usage: 32.91 KB

Dimensions: 117 lignes, 8 colonnes

Aperçu des premières lignes:
```

	The BOSS version of each Palu only takes 20%~40% damage, has a capture rate of about 70%, has a riding sprint speed of 100 points, and is larger in size.	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unna
0	In addition, a small numbe...	NaN	NaN	NaN	NaN	NaN	NaN	
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2	name	HP	NaN	name	Remote attack	NaN	name	R s (BC

```
Types de données:
```

```
The BOSS version of each Palu only takes 20%~40% damage, has a capture rate of ab
out 70%, has a riding sprint speed of 100 points, and is larger in size.     obje
ct
Unnamed: 1
object
Unnamed: 2
float64
Unnamed: 3
object
Unnamed: 4
object
Unnamed: 5
float64
Unnamed: 6
object
Unnamed: 7
object
dtype: object
Statistiques descriptives:
```

	The BOSS version of each Palu only takes 20%~40% damage, has a capture rate of about 70%, has a riding sprint speed of 100 points, and is larger in size.	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6
count	12	11	0.0	7	7	0.0	115
unique	12	7	NaN	7	5	NaN	113
top	In addition, a small numbe...	260	NaN	name	90	NaN	Thunderbird
freq	1	2	NaN	1	2	NaN	2
mean	NaN	NaN	NaN	NaN	NaN	NaN	NaN
std	NaN	NaN	NaN	NaN	NaN	NaN	NaN
min	NaN	NaN	NaN	NaN	NaN	NaN	NaN
25%	NaN	NaN	NaN	NaN	NaN	NaN	NaN
50%	NaN	NaN	NaN	NaN	NaN	NaN	NaN
75%	NaN	NaN	NaN	NaN	NaN	NaN	NaN
max	NaN	NaN	NaN	NaN	NaN	NaN	NaN

```
Valeurs manquantes:
```

	Total	Pourcentage
The BOSS version of each Palu only takes 20%~40% damage, has a capture rate of about 70%, has a riding sprint speed of 100 points, and is larger in size.	105	89.743590
Unnamed: 1	106	90.598291
Unnamed: 2	117	100.000000
Unnamed: 3	110	94.017094
Unnamed: 4	110	94.017094
Unnamed: 5	117	100.000000
Unnamed: 6	2	1.709402
Unnamed: 7	2	1.709402

```
Nombre de lignes dupliquées: 0

Analyse des colonnes catégorielles:

The BOSS version of each Palu only takes 20%~40% damage, has a capture rate of ab
out 70%, has a riding sprint speed of 100 points, and is larger in size.:
  - 13 valeurs uniques
  - Top 10 des valeurs les plus fréquentes:
    • In addition, a small number of BOSS have differences in HP and attack, see
the table below. There is no difference in other attributes except those listed b
elow: 1 occurrences
    • name: 1 occurrences
    • Chaos Knight BOSS: 1 occurrences
    • Chaos Knight: 1 occurrences
    • Winter Caller Beast BOSS: 1 occurrences
    • Winter Caller: 1 occurrences
    • Night Caller BOSS: 1 occurrences
    • Night Caller: 1 occurrences
    • Vortex Dragon BOSS: 1 occurrences
    • Vortex Dragon: 1 occurrences

Unnamed: 1:
  - 8 valeurs uniques
  - Valeurs: [nan 'HP' '260' '130' '420' '140' '330' '110']

Unnamed: 3:
  - 8 valeurs uniques
  - Valeurs: [nan 'name' 'Melupa BOSS' 'Melupa' 'Volt Meow BOSS' 'Volt Meow'
 'Snow Mammoth BOSS' 'snow mammoth']

Unnamed: 4:
  - 6 valeurs uniques
  - Valeurs: [nan 'Remote attack' '90' '75' '80' '85']

Unnamed: 6:
  - 114 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • Thunderbird: 2 occurrences
    • Thunderbird BOSS: 2 occurrences
    • Zixia Deer: 1 occurrences
    • Zixialu BOSS: 1 occurrences
    • Suzaku: 1 occurrences
  - Valeur la moins fréquente: White Velvet Snow Monster (1 occurrence(s))

Unnamed: 7:
  - 21 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • 800: 21 occurrences
    • 900: 17 occurrences
    • 700: 12 occurrences
    • 1200: 7 occurrences
    • 650: 7 occurrences
  - Valeur la moins fréquente: Riding speed (BOSS is 100 higher) (1 occurrence
(s))

Valeurs aberrantes (méthode IQR):

Valeurs aberrantes (méthode IQR):
Unnamed: 2: 0 valeurs aberrantes (0.0%)
Unnamed: 5: 0 valeurs aberrantes (0.0%)
```

```
======================================================================
```

In [38]:
```
explore_dataframe(tower_boss, 'Tower Boss')
```

```
===== Exploration de Tower Boss =====

--- Informations de base ---
Class: <class 'pandas.core.frame.DataFrame'>
Index: <class 'pandas.core.indexes.range.RangeIndex'> | RangeIndex(start=0, stop=
16, step=1)
Entries: 16 | Columns: 11
```

	Column	Non-Null Count	Dtype
0	name	16	object
1	Victor & Heterogeneous Gri...	16	object
2	Heterogeneous Griffin	16	object
3	Zoe & Lightning Bear	16	object
4	Lightning Bear	16	object
5	Marcus & Horus	16	object
6	Horus	16	object
7	Lily & Lily Queen	16	object
8	lily queen	16	object
9	Axel & Pollux	16	object
10	Pollux	16	object

```
Memory usage: 10.58 KB

Dimensions: 16 lignes, 11 colonnes

Aperçu des premières lignes:
```

	name	Victor & Heterogeneous Griffin	Heterogeneous Griffin	Zoe & Lightning Bear	Lightning Bear	Marcus & Horus	Horus	Lily & Lily Queen
0	HP	8000	120	6000	105	6500	100	5500
1	melee attack	130	130	100	120	100	100	100
2	Remote attack	200	120	100	100	120	105	110

```
Types de données:
```

```
name                              object
Victor & Heterogeneous Griffin    object
Heterogeneous Griffin             object
Zoe & Lightning Bear              object
Lightning Bear                    object
Marcus & Horus                    object
Horus                             object
Lily & Lily Queen                 object
lily queen                        object
Axel & Pollux                     object
Pollux                            object
dtype: object
Statistiques descriptives:
```

	name	Victor & Heterogeneous Griffin	Heterogeneous Griffin	Zoe & Lightning Bear	Lightning Bear	Marcus & Horus	Horus	Li Qu
count	16	16	16	16	16	16	16	
unique	16	14	13	11	12	14	14	
top	HP	TRUE	120	100	100	100	100	
freq	1	2	2	5	4	2	2	

Valeurs manquantes:

	Total	Pourcentage

```
Aucune valeur manquante détectée

Nombre de lignes dupliquées: 0

Analyse des colonnes catégorielles:

name:
  - 16 valeurs uniques
  - Top 10 des valeurs les plus fréquentes:
    • HP: 1 occurrences
    • melee attack: 1 occurrences
    • Remote attack: 1 occurrences
    • defense: 1 occurrences
    • Support: 1 occurrences
    • experience ratio: 1 occurrences
    • slow walking speed: 1 occurrences
    • walking speed: 1 occurrences
    • running speed: 1 occurrences
    • riding speed: 1 occurrences

Victor & Heterogeneous Griffin:
  - 14 valeurs uniques
  - Top 10 des valeurs les plus fréquentes:
    • TRUE: 2 occurrences
    • 80: 2 occurrences
    • 200: 1 occurrences
    • 220: 1 occurrences
    • 8000: 1 occurrences
    • 130: 1 occurrences
    • 30: 1 occurrences
    • 90: 1 occurrences
    • 1100: 1 occurrences
    • 850: 1 occurrences

Heterogeneous Griffin:
  - 13 valeurs uniques
  - Top 10 des valeurs les plus fréquentes:
    • 120: 2 occurrences
    • FALSE: 2 occurrences
    • 80: 2 occurrences
    • 130: 1 occurrences
    • 140: 1 occurrences
    • 1: 1 occurrences
    • 90: 1 occurrences
    • 1200: 1 occurrences
    • 850: 1 occurrences
    • 465: 1 occurrences

Zoe & Lightning Bear:
  - 11 valeurs uniques
  - Top 10 des valeurs les plus fréquentes:
    • 100: 5 occurrences
    • TRUE: 2 occurrences
    • 6000: 1 occurrences
    • 30: 1 occurrences
    • 80: 1 occurrences
    • 470: 1 occurrences
    • 140: 1 occurrences
    • 650: 1 occurrences
    • 287: 1 occurrences
```

    - 9: 1 occurrences

Lightning Bear:
  - 12 valeurs uniques
  - Top 10 des valeurs les plus fréquentes:
    - 100: 4 occurrences
    - FALSE: 2 occurrences
    - 120: 1 occurrences
    - 105: 1 occurrences
    - 1: 1 occurrences
    - 80: 1 occurrences
    - 470: 1 occurrences
    - 140: 1 occurrences
    - 550: 1 occurrences
    - 210: 1 occurrences

Marcus & Horus:
  - 14 valeurs uniques
  - Top 10 des valeurs les plus fréquentes:
    - 100: 2 occurrences
    - TRUE: 2 occurrences
    - 120: 1 occurrences
    - 125: 1 occurrences
    - 90: 1 occurrences
    - 6500: 1 occurrences
    - 30: 1 occurrences
    - 150: 1 occurrences
    - 1000: 1 occurrences
    - 200: 1 occurrences

Horus:
  - 14 valeurs uniques
  - Top 10 des valeurs les plus fréquentes:
    - 100: 2 occurrences
    - FALSE: 2 occurrences
    - 110: 1 occurrences
    - 90: 1 occurrences
    - 1: 1 occurrences
    - 105: 1 occurrences
    - 150: 1 occurrences
    - 200: 1 occurrences
    - 1400: 1 occurrences
    - 1000: 1 occurrences

Lily & Lily Queen:
  - 12 valeurs uniques
  - Top 10 des valeurs les plus fréquentes:
    - 100: 4 occurrences
    - TRUE: 2 occurrences
    - 110: 1 occurrences
    - 5500: 1 occurrences
    - 105: 1 occurrences
    - 30: 1 occurrences
    - 450: 1 occurrences
    - 60: 1 occurrences
    - 650: 1 occurrences
    - 275: 1 occurrences

lily queen:
  - 11 valeurs uniques

```
  - Top 10 des valeurs les plus fréquentes:
    • 100: 4 occurrences
    • 110: 2 occurrences
    • FALSE: 2 occurrences
    • 105: 1 occurrences
    • 1: 1 occurrences
    • 450: 1 occurrences
    • 60: 1 occurrences
    • 550: 1 occurrences
    • 275: 1 occurrences
    • 5: 1 occurrences

Axel & Pollux:
  - 12 valeurs uniques
  - Top 10 des valeurs les plus fréquentes:
    • 100: 4 occurrences
    • TRUE: 2 occurrences
    • 140: 1 occurrences
    • 6500: 1 occurrences
    • 110: 1 occurrences
    • 30: 1 occurrences
    • 900: 1 occurrences
    • 185: 1 occurrences
    • 1300: 1 occurrences
    • 542: 1 occurrences

Pollux:
  - 10 valeurs uniques
  - Valeurs: ['100' '130' '1' '185' '900' '1200' '250' 'FALSE' '5' '140']

Valeurs aberrantes (méthode IQR):

Valeurs aberrantes (méthode IQR):

======================================================================
```

In [149…

```
explore_dataframe(combat_attribute, 'Combat Attribut')
```

```
===== Exploration de Combat Attribut =====

--- Informations de base ---
Class: <class 'pandas.core.frame.DataFrame'>
Index: <class 'pandas.core.indexes.range.RangeIndex'> | RangeIndex(start=0, stop=
139, step=1)
Entries: 139 | Columns: 50
```

	Column	Non-Null Count	Dtype
0	Table of Contents Palu Job...	139	object
1	Unnamed: 1	139	object
2	Unnamed: 2	139	object
3	Unnamed: 3	139	object
4	Unnamed: 4	2	object
5	Unnamed: 5	1	object
6	Unnamed: 6	2	object
7	Unnamed: 7	139	object
8	Unnamed: 8	139	object
9	Unnamed: 9	139	object
10	Unnamed: 10	34	object
11	Unnamed: 11	139	object
12	Unnamed: 12	139	object
13	Unnamed: 13	139	object
14	Unnamed: 14	27	object
15	Unnamed: 15	139	object
16	Unnamed: 16	139	object
17	Unnamed: 17	139	object
18	Unnamed: 18	139	object
19	Unnamed: 19	26	object
20	Unnamed: 20	139	object
21	Unnamed: 21	139	object
22	Unnamed: 22	139	object
23	Unnamed: 23	139	object
24	Unnamed: 24	139	object
25	Unnamed: 25	139	object
26	Unnamed: 26	139	object
27	Non-bonus panel attack pow...	139	object
28	Unnamed: 28	139	object
29	Unnamed: 29	139	object
30	Unnamed: 30	139	object
31	Unnamed: 31	1	object
32	Unnamed: 32	139	object

	Column	Non-Null Count	Dtype
33	Unnamed: 33	139	object
34	Unnamed: 34	139	object
35	Unnamed: 35	139	object
36	Unnamed: 36	62	object
37	Unnamed: 37	139	object
38	Unnamed: 38	139	object
39	Unnamed: 39	139	object
40	Unnamed: 40	139	object
41	Unnamed: 41	18	object
42	Unnamed: 42	5	object
43	Partner skills	139	object
44	Unnamed: 44	139	object
45	Unnamed: 45	139	object
46	Unnamed: 46	139	object
47	Unnamed: 47	139	object
48	Unnamed: 48	139	object
49	Unnamed: 49	139	object

```
Memory usage: 409.73 KB
```

```
Dimensions: 139 lignes, 50 colonnes
```

```
Aperçu des premières lignes:
```

Table of Contents Palu Job Skill Table Skill Learning Level Tower BOSS Attributes Comparison with Normal BOSS Attributes Comparison of Partner Skill Values Individual Value Calculator	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5
0	ID	Chinese name	Name	CodeName	OverrideNameTextID	NamePrefixID	Ove
1	1	Mian Youyou	Lamball	SheepBall	NaN	NaN
2	2	Naughty cat	Cattiva	PinkCat	NaN	NaN

Types de données:

```
Table of Contents Palu Job Skill Table Skill Learning Level Tower BOSS Attributes
Comparison with Normal BOSS Attributes Comparison of Partner Skill Values Individ
ual Value Calculator    object
Unnamed: 1
object
Unnamed: 2
object
Unnamed: 3
object
Unnamed: 4
object
Unnamed: 5
object
Unnamed: 6
object
Unnamed: 7
object
Unnamed: 8
object
Unnamed: 9
object
Unnamed: 10
object
Unnamed: 11
object
Unnamed: 12
object
Unnamed: 13
object
Unnamed: 14
object
Unnamed: 15
object
Unnamed: 16
object
Unnamed: 17
object
Unnamed: 18
object
Unnamed: 19
object
Unnamed: 20
object
Unnamed: 21
object
Unnamed: 22
object
Unnamed: 23
object
Unnamed: 24
object
Unnamed: 25
object
Unnamed: 26
object
Non-bonus panel attack power range reference
object
Unnamed: 28
object
Unnamed: 29
```

```
object
Unnamed: 30
object
Unnamed: 31
object
Unnamed: 32
object
Unnamed: 33
object
Unnamed: 34
object
Unnamed: 35
object
Unnamed: 36
object
Unnamed: 37
object
Unnamed: 38
object
Unnamed: 39
object
Unnamed: 40
object
Unnamed: 41
object
Unnamed: 42
object
Partner skills
object
Unnamed: 44
object
Unnamed: 45
object
Unnamed: 46
object
Unnamed: 47
object
Unnamed: 48
object
Unnamed: 49
object
dtype: object
Statistiques descriptives:
```

	Table of Contents Palu Job Skill Table Skill Learning Level Tower BOSS Attributes Comparison with Normal BOSS Attributes Comparison of Partner Skill Values Individual Value Calculator	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed:
count	139	139	139	139	2	
unique	112	138	139	139	2	
top	12	Thunderbird	Name	CodeName	OverrideNameTextID	NamePrefixI
freq	2	2	1	1	1	

Valeurs manquantes:

	Total	Pourcentage
Unnamed: 4	137	98.561151
Unnamed: 5	138	99.280576
Unnamed: 6	137	98.561151
Unnamed: 10	105	75.539568
Unnamed: 14	112	80.575540
Unnamed: 19	113	81.294964
Unnamed: 31	138	99.280576
Unnamed: 36	77	55.395683
Unnamed: 41	121	87.050360
Unnamed: 42	134	96.402878

```
Nombre de lignes dupliquées: 0

Analyse des colonnes catégorielles:

Table of Contents Palu Job Skill Table Skill Learning Level Tower BOSS Attributes
Comparison with Normal BOSS Attributes Comparison of Partner Skill Values Individ
ual Value Calculator:
  - 112 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • 12: 2 occurrences
    • 13: 2 occurrences
    • 37: 2 occurrences
    • 40: 2 occurrences
    • 45: 2 occurrences
  - Valeur la moins fréquente: 111 (1 occurrence(s))

Unnamed: 1:
  - 138 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • Thunderbird: 2 occurrences
    • Chinese name: 1 occurrences
    • Naughty cat: 1 occurrences
    • Pipi Chicken: 1 occurrences
    • green leaf rat: 1 occurrences
  - Valeur la moins fréquente: Vortex Dragon (1 occurrence(s))

Unnamed: 2:
  - 139 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • Name: 1 occurrences
    • Lamball: 1 occurrences
    • Cattiva: 1 occurrences
    • Chikipi: 1 occurrences
    • Lifmunk: 1 occurrences
  - Valeur la moins fréquente: Jetragon (1 occurrence(s))

Unnamed: 3:
  - 139 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • CodeName: 1 occurrences
    • SheepBall: 1 occurrences
    • PinkCat: 1 occurrences
    • ChickenPal: 1 occurrences
    • Carbunclo: 1 occurrences
  - Valeur la moins fréquente: JetDragon (1 occurrence(s))

Unnamed: 4:
  - 3 valeurs uniques
  - Valeurs: ['OverrideNameTextID' nan 'PAL_NAME_PlantSlime']

Unnamed: 5:
  - 2 valeurs uniques
  - Valeurs: ['NamePrefixID' nan]

Unnamed: 6:
  - 3 valeurs uniques
  - Valeurs: ['OverridePartnerSkillTextID' nan 'PARTNERSKILL_PlantSlime']

Unnamed: 7:
  - 2 valeurs uniques
```

```
  - Valeurs: ['IsPal' 'TRUE']

Unnamed: 8:
  - 138 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • EPalTribeID::PlantSlime: 2 occurrences
    • Tribe: 1 occurrences
    • EPalTribeID::PinkCat: 1 occurrences
    • EPalTribeID::ChickenPal: 1 occurrences
    • EPalTribeID::Carbunclo: 1 occurrences
  - Valeur la moins fréquente: EPalTribeID::JetDragon (1 occurrence(s))

Unnamed: 9:
  - 139 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • BPClass: 1 occurrences
    • SheepBall: 1 occurrences
    • PinkCat: 1 occurrences
    • ChickenPal: 1 occurrences
    • Carbunclo: 1 occurrences
  - Valeur la moins fréquente: JetDragon (1 occurrence(s))

Unnamed: 10:
  - 3 valeurs uniques
  - Valeurs: ['variant' nan 'yes']

Unnamed: 11:
  - 6 valeurs uniques
  - Valeurs: ['Volume size' 'XS' 'S' 'L' 'M' 'XL']

Unnamed: 12:
  - 12 valeurs uniques
  - Top 10 des valeurs les plus fréquentes:
    • 1: 28 occurrences
    • 6: 17 occurrences
    • 8: 15 occurrences
    • 7: 15 occurrences
    • 2: 12 occurrences
    • 5: 11 occurrences
    • 9: 11 occurrences
    • 3: 10 occurrences
    • 4: 10 occurrences
    • 20: 5 occurrences

Unnamed: 13:
  - 10 valeurs uniques
  - Valeurs: ['Element 1' 'generally' 'Wood' 'fire' 'water' 'electricity' 'ice'
'dark'
 'land' 'dragon']

Unnamed: 14:
  - 10 valeurs uniques
  - Valeurs: ['Element 2' nan 'ice' 'land' 'dark' 'dragon' 'water' 'electricity'
'Wood'
 'fire']

Unnamed: 15:
  - 8 valeurs uniques
  - Valeurs: ['GenusCategory' 'EPalGenusCategoryType::Humanoid'
 'EPalGenusCategoryType::Bird' 'EPalGenusCategoryType::FourLegged'
```

```
 'EPalGenusCategoryType::Other' 'EPalGenusCategoryType::Fish'
 'EPalGenusCategoryType::Dragon' 'EPalGenusCategoryType::Monster']

Unnamed: 16:
  - 2 valeurs uniques
  - Valeurs: ['Organization' 'EPalOrganizationType::None']

Unnamed: 17:
  - 2 valeurs uniques
  - Valeurs: ['weapon' 'EPalWeaponType::None']

Unnamed: 18:
  - 2 valeurs uniques
  - Valeurs: ['WeaponEquip' 'FALSE']

Unnamed: 19:
  - 3 valeurs uniques
  - Valeurs: ['nocturnal' nan 'yes']

Unnamed: 20:
  - 45 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • 395: 10 occurrences
    • 330: 8 occurrences
    • 310: 8 occurrences
    • 280: 7 occurrences
    • 375: 7 occurrences
  - Valeur la moins fréquente: 460 (1 occurrence(s))

Unnamed: 21:
  - 18 valeurs uniques
  - Top 10 des valeurs les plus fréquentes:
    • 70: 20 occurrences
    • 100: 20 occurrences
    • 90: 19 occurrences
    • 80: 14 occurrences
    • 110: 14 occurrences
    • 120: 9 occurrences
    • 75: 8 occurrences
    • 95: 7 occurrences
    • 60: 7 occurrences
    • 105: 5 occurrences

Unnamed: 22:
  - 12 valeurs uniques
  - Top 10 des valeurs les plus fréquentes:
    • 100: 68 occurrences
    • 70: 27 occurrences
    • 80: 11 occurrences
    • 110: 10 occurrences
    • 90: 8 occurrences
    • 150: 5 occurrences
    • 130: 3 occurrences
    • 120: 2 occurrences
    • 50: 2 occurrences
    • melee attack: 1 occurrences

Unnamed: 23:
  - 19 valeurs uniques
  - Top 10 des valeurs les plus fréquentes:
```

    - 70: 21 occurrences
    - 80: 16 occurrences
    - 105: 14 occurrences
    - 100: 13 occurrences
    - 95: 12 occurrences
    - 90: 11 occurrences
    - 85: 8 occurrences
    - 75: 8 occurrences
    - 110: 6 occurrences
    - 115: 5 occurrences

Unnamed: 24:
  - 19 valeurs uniques
  - Top 10 des valeurs les plus fréquentes:
    - 70: 28 occurrences
    - 80: 25 occurrences
    - 90: 21 occurrences
    - 100: 17 occurrences
    - 120: 8 occurrences
    - 110: 6 occurrences
    - 75: 6 occurrences
    - 105: 5 occurrences
    - 60: 5 occurrences
    - 95: 5 occurrences

Unnamed: 25:
  - 11 valeurs uniques
  - Top 10 des valeurs les plus fréquentes:
    - 100: 116 occurrences
    - 70: 4 occurrences
    - 120: 4 occurrences
    - 90: 3 occurrences
    - 140: 3 occurrences
    - 30: 2 occurrences
    - 110: 2 occurrences
    - 150: 2 occurrences
    - support: 1 occurrences
    - 80: 1 occurrences

Unnamed: 26:
  - 2 valeurs uniques
  - Valeurs: ['Speed of work' '100']

Non-bonus panel attack power range reference:
  - 16 valeurs uniques
  - Top 10 des valeurs les plus fréquentes:
    - 107-109: 25 occurrences
    - 105-106: 21 occurrences
    - 106-108: 19 occurrences
    - 106-107: 16 occurrences
    - 107-110: 14 occurrences
    - 105-107: 8 occurrences
    - 109-112: 7 occurrences
    - 108-110: 6 occurrences
    - 104-105: 5 occurrences
    - 108-111: 5 occurrences

Unnamed: 28:
  - 19 valeurs uniques
  - Top 10 des valeurs les plus fréquentes:

    - 205-236: 21 occurrences
    - 220-256: 16 occurrences
    - 257-304: 14 occurrences
    - 250-295: 13 occurrences
    - 242-285: 12 occurrences
    - 235-275: 11 occurrences
    - 227-265: 8 occurrences
    - 212-246: 8 occurrences
    - 265-314: 6 occurrences
    - 272-324: 5 occurrences

Unnamed: 29:
  - 19 valeurs uniques
  - Top 10 des valeurs les plus fréquentes:
    - 362-441: 21 occurrences
    - 400-490: 16 occurrences
    - 493-611: 14 occurrences
    - 475-587: 13 occurrences
    - 456-563: 12 occurrences
    - 437-538: 11 occurrences
    - 418-514: 8 occurrences
    - 381-465: 8 occurrences
    - 512-636: 6 occurrences
    - 531-660: 5 occurrences

Unnamed: 30:
  - 6 valeurs uniques
  - Valeurs: ['AIRResponse' 'Friendly' 'Escape_to_Battle' 'Escape' 'Warlike'
 'NotInterested']

Unnamed: 31:
  - 2 valeurs uniques
  - Valeurs: ['AISightResponse' nan]

Unnamed: 32:
  - 11 valeurs uniques
  - Top 10 des valeurs les plus fréquentes:
    - 100: 123 occurrences
    - 150: 3 occurrences
    - 300: 3 occurrences
    - 280: 2 occurrences
    - 130: 2 occurrences
    - endurance: 1 occurrences
    - 160: 1 occurrences
    - 170: 1 occurrences
    - 220: 1 occurrences
    - 230: 1 occurrences

Unnamed: 33:
  - 30 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    - 100: 32 occurrences
    - 50: 22 occurrences
    - 80: 14 occurrences
    - 70: 11 occurrences
    - 30: 11 occurrences
  - Valeur la moins fréquente: 600 (1 occurrence(s))

Unnamed: 34:
  - 39 valeurs uniques

- Top 5 des valeurs les plus fréquentes:
  - 100: 24 occurrences
  - 150: 12 occurrences
  - 60: 11 occurrences
  - 200: 10 occurrences
  - 120: 9 occurrences
- Valeur la moins fréquente: 800 (1 occurrence(s))

Unnamed: 35:
- 32 valeurs uniques
- Top 5 des valeurs les plus fréquentes:
  - 400: 20 occurrences
  - 300: 14 occurrences
  - 600: 14 occurrences
  - 700: 12 occurrences
  - 500: 10 occurrences
- Valeur la moins fréquente: 1700 (1 occurrence(s))

Unnamed: 36:
- 21 valeurs uniques
- Top 5 des valeurs les plus fréquentes:
  - 800: 11 occurrences
  - 700: 10 occurrences
  - 900: 6 occurrences
  - 550: 5 occurrences
  - 1100: 4 occurrences
- Valeur la moins fréquente: 3300 (1 occurrence(s))

Unnamed: 37:
- 9 valeurs uniques
- Valeurs: ['(being) damage multiplier' '200%' '100%' '130%' '120%' '180%' '17
0%'
 '160%' '110%']

Unnamed: 38:
- 7 valeurs uniques
- Valeurs: ['catch rate' '150%' '90%' '110%' '100%' '120%' '130%']

Unnamed: 39:
- 2 valeurs uniques
- Valeurs: ['Experience multiplier' '100%']

Unnamed: 40:
- 119 valeurs uniques
- Top 5 des valeurs les plus fréquentes:
  - 1000: 6 occurrences
  - 6830: 3 occurrences
  - 6720: 3 occurrences
  - 1310: 2 occurrences
  - 1020: 2 occurrences
- Valeur la moins fréquente: 8680 (1 occurrence(s))

Unnamed: 41:
- 13 valeurs uniques
- Top 10 des valeurs les plus fréquentes:
  - legend: 5 occurrences
  - hard skin: 3 occurrences
  - timid: 1 occurrences
  - Must bring entry 1: 1 occurrences
  - rough: 1 occurrences

    - Uncharacteristically: 1 occurrences
    - Emperor Yan: 1 occurrences
    - Greedy: 1 occurrences
    - Yandi: 1 occurrences
    - Neptune: 1 occurrences

Unnamed: 42:
  - 6 valeurs uniques
  - Valeurs: ['Must bring entry 2' nan 'holy day' 'Hades' 'ice King' 'Shenlong']

Partner skills:
  - 37 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    - Attack power bonus%: 25 occurrences
    - Skill power: 13 occurrences
    - Skill power multiplier: 13 occurrences
    - Moving speed%: 13 occurrences
    - Ranch skills: 10 occurrences
  - Valeur la moins fréquente: Elemental power bonus%_Ice (1 occurrence(s))

Unnamed: 44:
  - 29 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    - 0: 25 occurrences
    - 50: 22 occurrences
    - 10: 16 occurrences
    - 40: 11 occurrences
    - 1.1: 10 occurrences
  - Valeur la moins fréquente: 13 (1 occurrence(s))

Unnamed: 45:
  - 33 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    - 55: 20 occurrences
    - 11: 16 occurrences
    - 10: 12 occurrences
    - 0: 11 occurrences
    - 50: 11 occurrences
  - Valeur la moins fréquente: 14.3 (1 occurrence(s))

Unnamed: 46:
  - 35 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    - 65: 20 occurrences
    - 13: 16 occurrences
    - 12: 12 occurrences
    - 0: 11 occurrences
    - 60: 11 occurrences
  - Valeur la moins fréquente: 16.9 (1 occurrence(s))

Unnamed: 47:
  - 32 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    - 80: 21 occurrences
    - 16: 16 occurrences
    - 15: 12 occurrences
    - 0: 11 occurrences
    - 70: 11 occurrences
  - Valeur la moins fréquente: 20.9 (1 occurrence(s))

```
Unnamed: 48:
  - 30 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • 20: 28 occurrences
    • 100: 20 occurrences
    • 80: 11 occurrences
    • 0: 11 occurrences
    • 2.5: 10 occurrences
  - Valeur la moins fréquente: 26 (1 occurrence(s))

Unnamed: 49:
  - 121 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • Can be moved around on its back. Two jumps are possible during the ride.: 3
occurrences
    • If it is in the party, players will be able to deal more damage when harves
ting.: 2 occurrences
    • Can be moved around on its back. It can also continuously attack with a gre
nade launcher while riding.: 2 occurrences
    • After activation, the sixth sense can be activated to detect the location o
f nearby dungeons.: 2 occurrences
    • When fighting alongside it, players will be able to deal more damage to ene
my weak points.: 2 occurrences
  - Valeur la moins fréquente: You can ride on its back and fly in the air. It ca
n also continuously attack with its missile launcher while riding. (1 occurrence
(s))

Valeurs aberrantes (méthode IQR):

Valeurs aberrantes (méthode IQR):

========================================================================
```

In [218…

```
explore_dataframe(hidden_attribute, 'Hidden Attribute')
```

```
===== Exploration de Hidden Attribute =====

--- Informations de base ---
Class: <class 'pandas.core.frame.DataFrame'>
Index: <class 'pandas.core.indexes.range.RangeIndex'> | RangeIndex(start=0, stop=
177, step=1)
Entries: 177 | Columns: 71
```

	Column	Non-Null Count	Dtype
0	Chinese name	177	object
1	code name	177	object
2	OverrideNameTextID	160	object
3	NamePrefixID	160	object
4	OverridePartnerSkillTextID	160	object
...	...	...	...
66	pasture	177	int64
67	Passive skill 1	17	object
68	Passive skill 2	4	object
69	Passive skill 3	0	float64
70	Passive skill 4	0	float64

71 rows × 3 columns

```
Memory usage: 279.95 KB

Dimensions: 177 lignes, 71 colonnes

Aperçu des premières lignes:
```

	Chinese name	code name	OverrideNameTextID	NamePrefixID	OverrideP
0	Victor & Heterogeneous Gri...	GYM_BlackGriffon	PAL_NAME_SnowBoss	GYM_NAME_Snow	PARTNE
1	Zoe & Lightning Bear	GYM_ElecPanda	PAL_NAME_GrassBoss	GYM_NAME_Meadow	PARTN
2	Marcus & Horus	GYM_Horus	PAL_NAME_DessertBoss	GYM_NAME_Desert	P

```
Types de données:
Chinese name                  object
code name                     object
OverrideNameTextID            object
NamePrefixID                  object
OverridePartnerSkillTextID    object
                              ...
pasture                        int64
Passive skill 1               object
Passive skill 2               object
Passive skill 3              float64
Passive skill 4              float64
Length: 71, dtype: object
Statistiques descriptives:
```

	Chinese name	code name	OverrideNameTextID	NamePrefixID	OverrideP
count	177	177	160	160	
unique	143	177	159	159	
top	does not appear in the game	GYM_BlackGriffon	PAL_NAME_PlantSlime	BOSS_NAME_PlantSlime	PARTNE
freq	34	1	2	2	
mean	NaN	NaN	NaN	NaN	
std	NaN	NaN	NaN	NaN	
min	NaN	NaN	NaN	NaN	
25%	NaN	NaN	NaN	NaN	
50%	NaN	NaN	NaN	NaN	
75%	NaN	NaN	NaN	NaN	
max	NaN	NaN	NaN	NaN	

Valeurs manquantes:

	Total	Pourcentage
OverrideNameTextID	17	9.604520
NamePrefixID	17	9.604520
OverridePartnerSkillTextID	17	9.604520
ZukanIndexSuffix	177	100.000000
Element 2	146	82.485876
AISightResponse	177	100.000000
Passive skill 1	160	90.395480
Passive skill 2	173	97.740113
Passive skill 3	177	100.000000
Passive skill 4	177	100.000000

```
Nombre de lignes dupliquées: 0

Analyse des colonnes catégorielles:

Chinese name:
  - 143 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • does not appear in the game: 34 occurrences
    • Ye NiNi: 2 occurrences
    • Zoe & Lightning Bear: 1 occurrences
    • Marcus & Horus: 1 occurrences
    • Lily & Lily Queen: 1 occurrences
  - Valeur la moins fréquente: 绿苔绒怪 (1 occurrence(s))

code name:
  - 177 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • GYM_BlackGriffon: 1 occurrences
    • GYM_ElecPanda: 1 occurrences
    • GYM_Horus: 1 occurrences
    • GYM_LilyQueen: 1 occurrences
    • GYM_ThunderDragonMan: 1 occurrences
  - Valeur la moins fréquente: DarkMutant (1 occurrence(s))

OverrideNameTextID:
  - 160 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • PAL_NAME_PlantSlime: 2 occurrences
    • PAL_NAME_GrassBoss: 1 occurrences
    • PAL_NAME_SnowBoss: 1 occurrences
    • PAL_NAME_ForestBoss: 1 occurrences
    • PAL_NAME_VolcanoBoss: 1 occurrences
  - Valeur la moins fréquente: PAL_NAME_Yeti_Grass (1 occurrence(s))

NamePrefixID:
  - 160 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • BOSS_NAME_PlantSlime: 2 occurrences
    • GYM_NAME_Meadow: 1 occurrences
    • GYM_NAME_Snow: 1 occurrences
    • GYM_NAME_Forest: 1 occurrences
    • GYM_NAME_Volcano: 1 occurrences
  - Valeur la moins fréquente: BOSS_NAME_Yeti_Grass (1 occurrence(s))

OverridePartnerSkillTextID:
  - 160 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • PARTNERSKILL_PlantSlime: 2 occurrences
    • PARTNERSKILL_lecPanda: 1 occurrences
    • PARTNERSKILL_lackGriffon: 1 occurrences
    • PARTNERSKILL_ilyQueen: 1 occurrences
    • PARTNERSKILL_hunderDragonMan: 1 occurrences
  - Valeur la moins fréquente: PARTNERSKILL_Yeti_Grass (1 occurrence(s))

Tribe:
  - 154 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • EPalTribeID::BlackGriffon: 2 occurrences
    • EPalTribeID::ElecPanda: 2 occurrences
    • EPalTribeID::Horus: 2 occurrences
```

```
    • EPalTribeID::LilyQueen: 2 occurrences
    • EPalTribeID::ThunderDragonMan: 2 occurrences
  - Valeur la moins fréquente: EPalTribeID::Yeti_Grass (1 occurrence(s))

BPClass:
  - 174 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • BeardedDragon: 2 occurrences
    • GuardianDog: 2 occurrences
    • GrassDragon: 2 occurrences
    • GYM_LilyQueen: 1 occurrences
    • GYM_Horus: 1 occurrences
  - Valeur la moins fréquente: DarkMutant (1 occurrence(s))

Size:
  - 4 valeurs uniques
  - Valeurs: ['XL' 'S' 'M' 'L']

Element 1:
  - 9 valeurs uniques
  - Valeurs: ['Dark' 'Electricity' 'Fire' 'Leaf' 'Dragon' 'Normal' 'Earth' 'Ice'
 'Water']

Element 2:
  - 9 valeurs uniques
  - Valeurs: [nan 'Electricity' 'Dark' 'Dragon' 'Ice' 'Earth' 'Water' 'Leaf' 'Fir
e']

GenusCategory:
  - 7 valeurs uniques
  - Valeurs: ['EPalGenusCategoryType::FourLegged' 'EPalGenusCategoryType::Humanoi
d'
 'EPalGenusCategoryType::Bird' 'EPalGenusCategoryType::Dragon'
 'EPalGenusCategoryType::Monster' 'EPalGenusCategoryType::Fish'
 'EPalGenusCategoryType::Other']

Organization:
  - 1 valeurs uniques
  - Valeurs: ['EPalOrganizationType::None']

weapon:
  - 1 valeurs uniques
  - Valeurs: ['EPalWeaponType::None']

(being) damage multiplier:
  - 4 valeurs uniques
  - Valeurs: ['100%' '40%' '20%' '10%']

Capture probability:
  - 7 valeurs uniques
  - Valeurs: ['100%' '70%' '77%' '63%' '105%' '91%' '84%']

AIRResponse:
  - 5 valeurs uniques
  - Valeurs: ['Warlike' 'Escape_to_Battle' 'NotInterested' 'Boss' 'Escape']

BattleBGM:
  - 6 valeurs uniques
  - Valeurs: ['EPalBattleBGMType::TowerBoss' 'EPalBattleBGMType::Cute'
 'EPalBattleBGMType::FieldBoss' 'EPalBattleBGMType::Legend'
```

```
 'EPalBattleBGMType::Cool' 'EPalBattleBGMType::Strong']

Passive skill 1:
  - 12 valeurs uniques
  - Top 10 des valeurs les plus fréquentes:
    • Legend: 5 occurrences
    • Deffence_up1: 3 occurrences
    • ElementBoost_Earth_2_PAL: 1 occurrences
    • PAL_rude: 1 occurrences
    • ElementBoost_Fire_2_PAL: 1 occurrences
    • PAL_FullStomach_Up_1: 1 occurrences
    • ElementBoost_Leaf_2_PAL: 1 occurrences
    • PAL_ALLAttack_down1: 1 occurrences
    • ElementResist_Normal_1_PAL: 1 occurrences
    • ElementBoost_Thunder_2_PAL: 1 occurrences

Passive skill 2:
  - 5 valeurs uniques
  - Valeurs: [nan 'ElementBoost_Dark_2_PAL' 'ElementBoost_Ice_2_PAL'
 'ElementBoost_Dragon_2_PAL' 'ElementBoost_Normal_2_PAL']

Valeurs aberrantes (méthode IQR):

Valeurs aberrantes (méthode IQR):
Pictorial ID: 5 valeurs aberrantes (2.8%)
  Valeurs: [-2, -2, -2, -2, -2]
ZukanIndexSuffix: 0 valeurs aberrantes (0.0%)
rarity: 5 valeurs aberrantes (2.8%)
  Valeurs: [20, 20, 20, 20, 20]
HP: 12 valeurs aberrantes (6.8%)
  Premières 10: [150, 150, 260, 260, 330, 420, 420, 5500, 6000, 6500]
melee attack: 6 valeurs aberrantes (3.4%)
  Valeurs: [140, 150, 150, 150, 150, 150]
Remote attack: 2 valeurs aberrantes (1.1%)
  Valeurs: [145, 200]
defense: 4 valeurs aberrantes (2.3%)
  Valeurs: [135, 140, 145, 220]
support: 28 valeurs aberrantes (15.8%)
  Premières 10: [30, 30, 50, 70, 70, 80, 80, 80, 80, 80]
CraftSpeed: 0 valeurs aberrantes (0.0%)
Experience multiplier: 22 valeurs aberrantes (12.4%)
  Premières 10: [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
price: 0 valeurs aberrantes (0.0%)
AISightResponse: 0 valeurs aberrantes (0.0%)
slow walking speed: 1 valeurs aberrantes (0.6%)
  Valeurs: [600]
walking speed: 5 valeurs aberrantes (2.8%)
  Valeurs: [350, 350, 350, 400, 800]
running speed: 1 valeurs aberrantes (0.6%)
  Valeurs: [1700]
Riding sprint speed: 4 valeurs aberrantes (2.3%)
  Valeurs: [1600, 1600, 1600, 3300]
Handling speed: 1 valeurs aberrantes (0.6%)
  Valeurs: [1250]
MaxFullStomach: 0 valeurs aberrantes (0.0%)
FullStomachDecreaseRate: 0 valeurs aberrantes (0.0%)
FoodAmount: 0 valeurs aberrantes (0.0%)
ViewingDistance: 0 valeurs aberrantes (0.0%)
ViewingAngle: 0 valeurs aberrantes (0.0%)
HearingRate: 0 valeurs aberrantes (0.0%)
```

```
BiologicalGrade: 19 valeurs aberrantes (10.7%)
  Premières 10: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
endurance: 16 valeurs aberrantes (9.0%)
  Premières 10: [70, 130, 130, 150, 150, 150, 160, 170, 220, 230]
Male probability: 15 valeurs aberrantes (8.5%)
  Premières 10: [0, 0, 10, 10, 20, 20, 30, 30, 30, 85]
fecundity: 5 valeurs aberrantes (2.8%)
  Valeurs: [9999, 9999, 9999, 9999, 9999]
Breathing fire: 25 valeurs aberrantes (14.1%)
  Premières 10: [1, 1, 1, 1, 1, 1, 1, 2, 2, 2]
watering: 15 valeurs aberrantes (8.5%)
  Premières 10: [1, 1, 1, 1, 1, 1, 2, 2, 2, 2]
planting: 24 valeurs aberrantes (13.6%)
  Premières 10: [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
generate electricity: 13 valeurs aberrantes (7.3%)
  Premières 10: [1, 1, 1, 2, 2, 2, 2, 2, 3, 3]
manual: 10 valeurs aberrantes (5.6%)
  Valeurs: [3, 3, 3, 3, 3, 3, 3, 3, 3, 4]
collection: 5 valeurs aberrantes (2.8%)
  Valeurs: [3, 3, 3, 3, 4]
logging: 43 valeurs aberrantes (24.3%)
  Premières 10: [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
Mining: 29 valeurs aberrantes (16.4%)
  Premières 10: [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
OilExtraction (not shown in game): 9 valeurs aberrantes (5.1%)
  Valeurs: [1, 1, 1, 2, 2, 3, 3, 3, 3]
pharmaceutical: 27 valeurs aberrantes (15.3%)
  Premières 10: [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
cool down: 18 valeurs aberrantes (10.2%)
  Premières 10: [1, 1, 1, 1, 1, 1, 2, 2, 2, 2]
carry: 0 valeurs aberrantes (0.0%)
pasture: 13 valeurs aberrantes (7.3%)
  Premières 10: [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
Passive skill 3: 0 valeurs aberrantes (0.0%)
Passive skill 4: 0 valeurs aberrantes (0.0%)

======================================================================
```

Fichier hide-palu-attributs

Nous avons supprimer la colonne chinese name car nous avons vue que les noms des pals etait représenter par:

- des idéogrammes chinois
- et des valeurs "does not appear in the game

Ensuite sur ce fichier nous avons supprimmer la colonnes vide ZukanIndexSuffix

Nous avons verifier que le types des données de chaque colonnes avec des valeurs numeriques soit enregistrer avec le types nombre entiers.

Nous avons gerer les données manquantes en leur mettant des valeurs NULL.

Nous avons gerer les noms des pals en ne laissant que leur nom au lieu de par exemple Boss_Alpaca et si les colonnes corresponder exactement a une autre alors on decider de complétement suprimmer cette colonne.

Fichier comparison ordinary boss attribut

Nous avons remplacer les valeur manquantes par des Null et renommer chaque colonnes en mettant la 1ere lignes contenant les titres en en-tete

Fichier Palu combat attribute

- Nous avons suprimmer la colonnes id car quand nous allons l'inserer dans la base de données l'id sera gerer automatiquement.
- remplir les données manquantes par des valeurs null.
- la 1ere ligne étant le nom des colonnes nous avons decider de les utiliser comme entete.
- la colonnes nocturnal contenant dans certaines celulles la valeur 'yes' nous avons decider de remplir les données manquantes par des 'no'
- De changer ces no et yes par des 0 et des 1 pour avoir des données plus facilement utilisable plus tard
- separation de l'intervalle des colonnes level 1, level 20 et level 50 en deux colonnes a chaque fois c'est a dire avoir des colonnes min et max pour ces 3 colonnes

Fichier palu job skills

- Nous avons suprimmer la colonne ID vue qu'il seront gerer automatiquement lors de l'insertion dans la base de données
- Colonne night shift remplacer yes par 1 et no/valeur null par 0
- Pour les valeurs des colonnes carry, logging, etc.. nous avons decider:
  - c'etait des colonnes d'action des pals de la formater avec des 1 et 0 car soit le pal fait cette action soit il ne la fait pas
  - cette colonnes contenait des valeurs tels que 2,3,4 nous avons considerer que cela equivalait a un 1 et que le pal faisait cette action
- dans les colonnes pasture minimum output et the largest ranch:
  - enlever les phrases according to partner skill level
  - mettre des valeurs null pour les données manquantes

Fichier Palu refresh level

- suppression de la colonne id car gerer automatiquement lors de l'insertion
- colonne night only formater avec des 1 et des 0 pour dire qu'il n'apparait que pendant la nuit(1) ou que le jour(0)

Fichier Tower BOSS attribute comparison

- mettre les colonne sur le bon types de données: - colonnes avec des chiffres non décimaux nombres entier - colonnes avec des nombres décimaux --> nombre decimales -colonnes avec true/false changer en 0 et 1 et en binaire

Création de la fonction d'analyse exploratoires des données propres

In [44]:
```
# Définition du chemin vers les fichiers propres
data_path = "../data/"

# Vérification de l'existence du repertoire , if not utiliser le repertoire cour
if not os.path.exists(data_path):
    data_path = "./"

# Affichage des fichiers dispo pour la vérif
files = [f for f in os.listdir(data_path) if f.endswith('.csv')] # ici on liste
print("Fichiers dispo:")
for file in files:
    print(f"-{file}")

# Chargement des données brutes & gestion d'erreurs grâce à pd.read_csv en défin
try:
    combat_attribute = pd.read_csv(f'{data_path}Clean_Data--Palu-combat-attribut
    refresh_area = pd.read_csv(f'{data_path}Clean_Data-Palu-refresh-level.csv')
    ordinary_boss = pd.read_csv(f'{data_path}Clean_Data-comparison-of-ordinary-B
    tower_boss = pd.read_csv(f'{data_path}Clean_Data-Tower-BOSS-attribute-compar
    job_skill = pd.read_csv(f'{data_path}Clean_Data-Palu-Job-Skills.csv')
    hidden_attribute = pd.read_csv(f'{data_path}Clean_Data-hide-pallu-attributes
    print("\nDonnées brutes chargées")
except FileNotFoundError as e :
  print(f"Erreur: {e}. Vérifier les noms des fichiers et le chemin.")
```

```
Fichiers dispo:
-Clean_Data--Palu-combat-attribute.csv
-Clean_Data-comparison-of-ordinary-BOSS-attributes.csv
-Clean_Data-hide-pallu-attributes.csv
-Clean_Data-Palu-Job-Skills.csv
-Clean_Data-Palu-refresh-level.csv
-Clean_Data-Tower-BOSS-attribute-comparison.csv

Données brutes chargées
```

In [55]:
```
explore_dataframe(job_skill, 'Job Skills')
```

```
===== Exploration de Job Skills =====

--- Informations de base ---
Class: <class 'pandas.core.frame.DataFrame'>
Index: <class 'pandas.core.indexes.range.RangeIndex'> | RangeIndex(start=0, stop=
138, step=1)
Entries: 138 | Columns: 22
```

	Column	Non-Null Count	Dtype
0	English name	138	object
1	Chinese name	138	object
2	Volume size	138	object
3	Food intake	138	int64
4	night shift	138	bool
5	Total skills	138	int64
6	Make a fire	138	bool
7	watering	138	bool
8	planting	138	bool
9	generate electricity	138	bool
10	manual	138	bool
11	collection	138	bool
12	logging	138	bool
13	Mining	138	bool
14	pharmaceutical	138	bool
15	cool down	138	bool
16	pasture	138	bool
17	carry	138	bool
18	Handling speed	56	float64
19	ranch items	13	object
20	pasture minimum output	12	float64
21	The largest ranch	12	object

```
Memory usage: 42.15 KB

Dimensions: 138 lignes, 22 colonnes

Aperçu des premières lignes:
```

	English name	Chinese name	Volume size	Food intake	night shift	Total skills	Make a fire	watering	planting	generate electricity
0	Lifmunk	green leaf rat	smallest	1	False	5	False	False	True	False
1	Foxparks	tinder fox	smallest	2	False	1	True	False	False	False
2	Rooby	Fire Deer	Small	3	False	1	True	False	False	False

```
Types de données:
English name               object
Chinese name               object
Volume size                object
Food intake                 int64
night shift                  bool
Total skills                int64
Make a fire                  bool
watering                     bool
planting                     bool
generate electricity         bool
manual                       bool
collection                   bool
logging                      bool
Mining                       bool
pharmaceutical               bool
cool down                    bool
pasture                      bool
carry                        bool
Handling speed             float64
ranch items                 object
pasture minimum output     float64
The largest ranch           object
dtype: object
Statistiques descriptives:
```

	English name	Chinese name	Volume size	Food intake	night shift	Total skills	Make a fire	watering	p
count	138	138	138	138.000000	138	138.000000	138	138	
unique	138	137	5	NaN	2	NaN	2	2	
top	Lifmunk	Thunderbird	big	NaN	False	NaN	False	False	
freq	1	2	43	NaN	113	NaN	116	123	
mean	NaN	NaN	NaN	4.550725	NaN	4.188406	NaN	NaN	
std	NaN	NaN	NaN	2.263878	NaN	2.703204	NaN	NaN	
min	NaN	NaN	NaN	1.000000	NaN	1.000000	NaN	NaN	
25%	NaN	NaN	NaN	3.000000	NaN	2.000000	NaN	NaN	
50%	NaN	NaN	NaN	5.000000	NaN	4.000000	NaN	NaN	
75%	NaN	NaN	NaN	7.000000	NaN	6.000000	NaN	NaN	
max	NaN	NaN	NaN	9.000000	NaN	12.000000	NaN	NaN	

```
Valeurs manquantes:
```

	Total	Pourcentage
Handling speed	82	59.420290
ranch items	125	90.579710
pasture minimum output	126	91.304348
The largest ranch	126	91.304348

```
Nombre de lignes dupliquées: 0

Analyse des colonnes catégorielles:

English name:
  - 138 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • Lifmunk: 1 occurrences
    • Foxparks: 1 occurrences
    • Rooby: 1 occurrences
    • Jolthog Cryst: 1 occurrences
    • Gumoss: 1 occurrences
  - Valeur la moins fréquente: Orserk (1 occurrence(s))

Chinese name:
  - 137 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • Thunderbird: 2 occurrences
    • tinder fox: 1 occurrences
    • Fire Deer: 1 occurrences
    • ice agouti: 1 occurrences
    • Ye NiNi: 1 occurrences
  - Valeur la moins fréquente: Pollux (1 occurrence(s))

Volume size:
  - 5 valeurs uniques
  - Valeurs: ['smallest' 'Small' 'medium' 'big' 'maximum']

ranch items:
  - 11 valeurs uniques
  - Top 10 des valeurs les plus fréquentes:
    • wool: 3 occurrences
    • gold: 2 occurrences
    • Palu Ball Advanced Palu Ball Arrow Gold Coin: 1 occurrences
    • Egg: 1 occurrences
    • milk: 1 occurrences
    • Marshmallow: 1 occurrences
    • red wild berries: 1 occurrences
    • high quality cloth: 1 occurrences
    • Honey: 1 occurrences
    • fire breathing organ: 1 occurrences

The largest ranch:
  - 2 valeurs uniques
  - Valeurs: [nan 'Rank']

Valeurs aberrantes (méthode IQR):

Valeurs aberrantes (méthode IQR):
Food intake: 0 valeurs aberrantes (0.0%)
Total skills: 0 valeurs aberrantes (0.0%)
Handling speed: 0 valeurs aberrantes (0.0%)
pasture minimum output: 3 valeurs aberrantes (2.2%)
  Valeurs: [2.0, 10.0, 10.0]

================================================================================
```

In [50]:
```
explore_dataframe(refresh_area, 'Refresh area')
```

```
===== Exploration de Refresh area =====

--- Informations de base ---
Class: <class 'pandas.core.frame.DataFrame'>
Index: <class 'pandas.core.indexes.range.RangeIndex'> | RangeIndex(start=0, stop=
136, step=1)
Entries: 136 | Columns: 7
```

	Column	Non-Null Count	Dtype
0	name	136	object
1	minimum level	136	int64
2	maximum level	136	int64
3	fecondity	136	int64
4	pallu refresh type	136	object
5	night only	136	bool
6	refresh area	136	object

```
Memory usage: 29.85 KB

Dimensions: 136 lignes, 7 colonnes

Aperçu des premières lignes:
```

	name	minimum level	maximum level	fecondity	pallu refresh type	night only	refresh area
0	Mian Youyou	1	14	1470	Creeps	False	grassland
1	Naughty cat	1	13	1460	Creeps	False	grassland
2	Pipi Chicken	1	13	1500	Creeps	False	grassland

```
Types de données:
name                 object
minimum level         int64
maximum level         int64
fecondity             int64
pallu refresh type   object
night only             bool
refresh area         object
dtype: object
Statistiques descriptives:
```

	name	minimum level	maximum level	fecondity	pallu refresh type	night only	refresh area
count	136	136.000000	136.000000	136.000000	136	136	136
unique	135	NaN	NaN	NaN	5	2	10
top	Thunderbird	NaN	NaN	NaN	Creeps	False	grassland
freq	2	NaN	NaN	NaN	98	119	49
mean	NaN	20.823529	33.036765	865.639706	NaN	NaN	NaN
std	NaN	14.480067	12.140784	420.953005	NaN	NaN	NaN
min	NaN	1.000000	10.000000	10.000000	NaN	NaN	NaN
25%	NaN	9.000000	22.000000	487.500000	NaN	NaN	NaN
50%	NaN	18.000000	38.000000	897.500000	NaN	NaN	NaN
75%	NaN	32.250000	45.000000	1240.000000	NaN	NaN	NaN
max	NaN	50.000000	50.000000	1500.000000	NaN	NaN	NaN

```
Valeurs manquantes:
```

Total	Pourcentage

```
Aucune valeur manquante détectée

Nombre de lignes dupliquées: 0

Analyse des colonnes catégorielles:

name:
  - 135 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • Thunderbird: 2 occurrences
    • Naughty cat: 1 occurrences
    • Mian Youyou: 1 occurrences
    • green leaf rat: 1 occurrences
    • tinder fox: 1 occurrences
  - Valeur la moins fréquente: Winter Caller (1 occurrence(s))

pallu refresh type:
  - 5 valeurs uniques
  - Valeurs: ['Creeps' 'Secret Domain BOSS' 'event' 'Random dungeon boss' 'Wild B
OSS']

refresh area:
  - 10 valeurs uniques
  - Valeurs: ['grassland' 'islands' 'dungeon' 'Snow' 'forest' 'volcano' 'game res
erve'
 'desert' 'forest snow' 'Random events']

Valeurs aberrantes (méthode IQR):

Valeurs aberrantes (méthode IQR):
minimum level: 0 valeurs aberrantes (0.0%)
maximum level: 0 valeurs aberrantes (0.0%)
fecondity: 0 valeurs aberrantes (0.0%)

======================================================================
```

In [53]:
```
explore_dataframe(ordinary_boss, 'Ordinary Boss')
```

```
===== Exploration de Ordinary Boss =====

--- Informations de base ---
Class: <class 'pandas.core.frame.DataFrame'>
Index: <class 'pandas.core.indexes.range.RangeIndex'> | RangeIndex(start=0, stop=
129, step=1)
Entries: 129 | Columns: 4
```

	Column	Non-Null Count	Dtype
0	name	129	object
1	HP	26	float64
2	Remote attack	18	float64
3	Riding speed (BOSS is 100 ...	129	int64

```
Memory usage: 12.18 KB

Dimensions: 129 lignes, 4 colonnes

Aperçu des premières lignes:
```

	name	HP	Remote attack	Riding speed (BOSS is 100 higher)
0	Heterogeneous Griffin BOSS	NaN	NaN	1300
1	Heterogeneous Griffin	NaN	NaN	1200
2	Ye Panda BOSS	NaN	NaN	800

```
Types de données:
name                                 object
HP                                  float64
Remote attack                       float64
Riding speed (BOSS is 100 higher)     int64
dtype: object
Statistiques descriptives:
```

	name	HP	Remote attack	Riding speed (BOSS is 100 higher)
count	129	26.000000	18.00000	129.000000
unique	119	NaN	NaN	NaN
top	Snow Mammoth BOSS	NaN	NaN	NaN
freq	2	NaN	NaN	NaN
mean	NaN	238.076923	82.50000	922.480620
std	NaN	121.128666	6.47393	248.665284
min	NaN	110.000000	75.00000	500.000000
25%	NaN	132.500000	75.00000	750.000000
50%	NaN	200.000000	82.50000	900.000000
75%	NaN	330.000000	90.00000	1100.000000
max	NaN	420.000000	90.00000	1600.000000

```
Valeurs manquantes:
```

	Total	Pourcentage
HP	103	79.844961
Remote attack	111	86.046512

```
Nombre de lignes dupliquées: 0

Analyse des colonnes catégorielles:

name:
  - 119 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • Snow Mammoth BOSS: 2 occurrences
    • Thunderbird: 2 occurrences
    • Thunderbird BOSS: 2 occurrences
    • snow mammoth: 2 occurrences
    • Melupa BOSS: 2 occurrences
  - Valeur la moins fréquente: Cloud sea deer (1 occurrence(s))

Valeurs aberrantes (méthode IQR):

Valeurs aberrantes (méthode IQR):
HP: 0 valeurs aberrantes (0.0%)
Remote attack: 0 valeurs aberrantes (0.0%)
Riding speed (BOSS is 100 higher): 0 valeurs aberrantes (0.0%)

======================================================================
```

In [54]:
```
explore_dataframe(tower_boss, 'Tower Boss')
```

```
===== Exploration de Tower Boss =====

--- Informations de base ---
Class: <class 'pandas.core.frame.DataFrame'>
Index: <class 'pandas.core.indexes.range.RangeIndex'> | RangeIndex(start=0, stop=
10, step=1)
Entries: 10 | Columns: 17
```

	Column	Non-Null Count	Dtype
0	name	10	object
1	HP	10	int64
2	melee attack	10	int64
3	remote attack	10	int64
4	defense	10	int64
5	Support	10	int64
6	experience ratio	10	int64
7	slow walking speed	10	int64
8	walking speed	10	int64
9	running speed	10	int64
10	riding speed	10	int64
11	Handling speed	10	int64
12	ignore the bluntness	10	bool
13	ignore displacement	10	bool
14	BiologicalGrade	10	int64
15	endurance	10	int64
16	fecundity	10	int64

```
Memory usage: 1.94 KB
```

```
Dimensions: 10 lignes, 17 colonnes
```

```
Aperçu des premières lignes:
```

	name	HP	melee attack	remote attack	defense	Support	experience ratio	slow walking speed	walking speed
0	Victor & Heterogeneous Gri...	8000	130	200	220	90	30	80	80
1	Heterogeneous Griffin	120	130	120	140	90	1	80	80
2	Zoe & Lightning Bear	6000	100	100	100	100	30	80	140

```
Types de données:
```

```
name                    object
HP                       int64
melee attack             int64
remote attack            int64
defense                  int64
Support                  int64
experience ratio         int64
slow walking speed       int64
walking speed            int64
running speed            int64
riding speed             int64
Handling speed           int64
ignore the bluntness      bool
ignore displacement       bool
BiologicalGrade          int64
endurance                int64
fecundity                int64
dtype: object
Statistiques descriptives:
```

	name	HP	melee attack	remote attack	defense	Support	exp
count	10	10.000000	10.000000	10.000000	10.000000	10.000000	10.
unique	10	NaN	NaN	NaN	NaN	NaN	
top	Victor & Heterogeneous Gri...	NaN	NaN	NaN	NaN	NaN	
freq	1	NaN	NaN	NaN	NaN	NaN	
mean	NaN	3303.500000	108.000000	123.500000	121.500000	96.000000	15.
std	NaN	3426.634282	13.165612	29.818898	36.897004	5.163978	15.
min	NaN	100.000000	100.000000	100.000000	100.000000	90.000000	1.
25%	NaN	106.250000	100.000000	106.250000	101.250000	90.000000	1.
50%	NaN	2810.000000	100.000000	115.000000	107.500000	100.000000	15.
75%	NaN	6375.000000	115.000000	127.500000	121.250000	100.000000	30.
max	NaN	8000.000000	130.000000	200.000000	220.000000	100.000000	30.

Valeurs manquantes:

Total	Pourcentage

```
Aucune valeur manquante détectée

Nombre de lignes dupliquées: 0

Analyse des colonnes catégorielles:

name:
  - 10 valeurs uniques
  - Valeurs: ['Victor & Heterogeneous Griffin' 'Heterogeneous Griffin'
 'Zoe & Lightning Bear' 'Lightning Bear' 'Marcus & Horus' 'Horus'
 'Lily & Lily Queen' 'lily queen' 'Axel & Pollux' 'Pollux']

Valeurs aberrantes (méthode IQR):

Valeurs aberrantes (méthode IQR):
HP: 0 valeurs aberrantes (0.0%)
melee attack: 0 valeurs aberrantes (0.0%)
remote attack: 1 valeurs aberrantes (10.0%)
  Valeurs: [200]
defense: 1 valeurs aberrantes (10.0%)
  Valeurs: [220]
Support: 0 valeurs aberrantes (0.0%)
experience ratio: 0 valeurs aberrantes (0.0%)
slow walking speed: 2 valeurs aberrantes (20.0%)
  Valeurs: [150, 150]
walking speed: 0 valeurs aberrantes (0.0%)
running speed: 0 valeurs aberrantes (0.0%)
riding speed: 0 valeurs aberrantes (0.0%)
Handling speed: 0 valeurs aberrantes (0.0%)
BiologicalGrade: 0 valeurs aberrantes (0.0%)
endurance: 2 valeurs aberrantes (20.0%)
  Valeurs: [230, 250]
fecundity: 0 valeurs aberrantes (0.0%)

======================================================================
```

In [48]:
```
explore_dataframe(combat_attribute, 'Combat Attribut')
```

```
===== Exploration de Combat Attribut =====

--- Informations de base ---
Class: <class 'pandas.core.frame.DataFrame'>
Index: <class 'pandas.core.indexes.range.RangeIndex'> | RangeIndex(start=0, stop=
138, step=1)
Entries: 138 | Columns: 52
```

	Column	Non-Null Count	Dtype
0	Chinese name	138	object
1	Name	138	object
2	CodeName	138	object
3	OverrideNameTextID	1	object
4	NamePrefixID	0	float64
5	OverridePartnerSkillTextID	1	object
6	IsPal	138	int64
7	Tribe	138	object
8	BPClass	138	object
9	variant	138	int64
10	Volume size	138	object
11	rarity	138	int64
12	Element 1	138	object
13	Element 2	138	object
14	GenusCategory	138	object
15	Organization	0	float64
16	weapon	0	float64
17	WeaponEquip	138	int64
18	nocturnal	138	int64
19	total_4D	138	int64
20	HP	138	int64
21	melee attack	138	int64
22	Remote attack	138	int64
23	defense	138	int64
24	support	138	int64
25	Speed of work	138	int64
26	Level1Min	138	int64
27	Level1Max	138	int64
28	Level20Min	138	int64
29	Level20Max	138	int64
30	Level50Min	138	int64
31	Level50Max	138	int64
32	AIRResponse	138	object

	Column	Non-Null Count	Dtype
33	AISightResponse	138	int64
34	endurance	138	int64
35	slow walking speed	138	int64
36	walking speed	138	int64
37	running speed	138	int64
38	Riding sprint speed	138	int64
39	damage multiplier	138	object
40	catch rate	138	object
41	Experience multiplier	138	int64
42	price	138	int64
43	Must bring entry 1	138	object
44	Must bring entry 2	138	object
45	Numerical description	138	object
46	Skill description	138	object
47	lv1.1	138	float64
48	lv2.1	138	float64
49	lv3.1	138	float64
50	lv4.1	138	float64
51	lv5.1	138	float64

```
Memory usage: 198.21 KB
```

```
Dimensions: 138 lignes, 52 colonnes
```

```
Aperçu des premières lignes:
```

	Chinese name	Name	CodeName	OverrideNameTextID	NamePrefixID	OverridePartnerSkil
0	Mian Youyou	Lamball	SheepBall	NaN	NaN	
1	Pipi Chicken	Chikipi	ChickenPal	NaN	NaN	
2	green leaf rat	Lifmunk	Carbunclo	NaN	NaN	

```
Types de données:
```

```
Chinese name                  object
Name                          object
CodeName                      object
OverrideNameTextID            object
NamePrefixID                 float64
OverridePartnerSkillTextID    object
IsPal                          int64
Tribe                         object
BPClass                       object
variant                        int64
Volume size                   object
rarity                         int64
Element 1                     object
Element 2                     object
GenusCategory                 object
Organization                 float64
weapon                       float64
WeaponEquip                    int64
nocturnal                      int64
total_4D                       int64
HP                             int64
melee attack                   int64
Remote attack                  int64
defense                        int64
support                        int64
Speed of work                  int64
Level1Min                      int64
Level1Max                      int64
Level20Min                     int64
Level20Max                     int64
Level50Min                     int64
Level50Max                     int64
AIRResponse                   object
AISightResponse                int64
endurance                      int64
slow walking speed             int64
walking speed                  int64
running speed                  int64
Riding sprint speed            int64
damage multiplier             object
catch rate                    object
Experience multiplier          int64
price                          int64
Must bring entry 1            object
Must bring entry 2            object
Numerical description         object
Skill description             object
lv1.1                        float64
lv2.1                        float64
lv3.1                        float64
lv4.1                        float64
lv5.1                        float64
dtype: object
Statistiques descriptives:
```

	Chinese name	Name	CodeName	OverrideNameTextID	NamePrefixID	OverridePa
count	138	138	138	1	0.0	
unique	137	138	138	1	NaN	
top	Thunderbird	Lamball	SheepBall	PAL_NAME_PlantSlime	NaN	PARTNEF
freq	2	1	1	1	NaN	
mean	NaN	NaN	NaN	NaN	NaN	
std	NaN	NaN	NaN	NaN	NaN	
min	NaN	NaN	NaN	NaN	NaN	
25%	NaN	NaN	NaN	NaN	NaN	
50%	NaN	NaN	NaN	NaN	NaN	
75%	NaN	NaN	NaN	NaN	NaN	
max	NaN	NaN	NaN	NaN	NaN	

Valeurs manquantes:

	Total	Pourcentage
OverrideNameTextID	137	99.275362
NamePrefixID	138	100.000000
OverridePartnerSkillTextID	137	99.275362
Organization	138	100.000000
weapon	138	100.000000

```
Nombre de lignes dupliquées: 0

Analyse des colonnes catégorielles:

Chinese name:
  - 137 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • Thunderbird: 2 occurrences
    • Pipi Chicken: 1 occurrences
    • green leaf rat: 1 occurrences
    • tinder fox: 1 occurrences
    • surf duck: 1 occurrences
  - Valeur la moins fréquente: Vortex Dragon (1 occurrence(s))

Name:
  - 138 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • Lamball: 1 occurrences
    • Chikipi: 1 occurrences
    • Lifmunk: 1 occurrences
    • Foxparks: 1 occurrences
    • Fuack: 1 occurrences
  - Valeur la moins fréquente: Jetragon (1 occurrence(s))

CodeName:
  - 138 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • SheepBall: 1 occurrences
    • ChickenPal: 1 occurrences
    • Carbunclo: 1 occurrences
    • Kitsunebi: 1 occurrences
    • BluePlatypus: 1 occurrences
  - Valeur la moins fréquente: JetDragon (1 occurrence(s))

OverrideNameTextID:
  - 2 valeurs uniques
  - Valeurs: [nan 'PAL_NAME_PlantSlime']

OverridePartnerSkillTextID:
  - 2 valeurs uniques
  - Valeurs: [nan 'PARTNERSKILL_PlantSlime']

Tribe:
  - 137 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • PlantSlime: 2 occurrences
    • ChickenPal: 1 occurrences
    • Carbunclo: 1 occurrences
    • Kitsunebi: 1 occurrences
    • Blueplatypus: 1 occurrences
  - Valeur la moins fréquente: JetDragon (1 occurrence(s))

BPClass:
  - 138 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • SheepBall: 1 occurrences
    • ChickenPal: 1 occurrences
    • Carbunclo: 1 occurrences
    • Kitsunebi: 1 occurrences
    • BluePlatypus: 1 occurrences
```

```
  - Valeur la moins fréquente: JetDragon (1 occurrence(s))

Volume size:
  - 5 valeurs uniques
  - Valeurs: ['XS' 'S' 'L' 'M' 'XL']

Element 1:
  - 9 valeurs uniques
  - Valeurs: ['generally' 'Wood' 'fire' 'water' 'electricity' 'ice' 'dark' 'land'
 'dragon']

Element 2:
  - 9 valeurs uniques
  - Valeurs: ['none' 'ice' 'land' 'dark' 'Wood' 'electricity' 'dragon' 'water' 'f
ire']

GenusCategory:
  - 7 valeurs uniques
  - Valeurs: ['Humanoid' 'Bird' 'FourLegged' 'Other' 'Fish' 'Dragon' 'Monster']

AIRResponse:
  - 5 valeurs uniques
  - Valeurs: ['Friendly' 'Escape' 'Escape_to_Battle' 'Warlike' 'NotInterested']

damage multiplier:
  - 8 valeurs uniques
  - Valeurs: ['2' '1' '1,3' '1,2' '1,8' '1,7' '1,6' '1,1']

catch rate:
  - 6 valeurs uniques
  - Valeurs: ['1,5' '0,9' '1,1' '1' '1,2' '1,3']

Must bring entry 1:
  - 12 valeurs uniques
  - Top 10 des valeurs les plus fréquentes:
    • none: 121 occurrences
    • legend: 5 occurrences
    • hard skin: 3 occurrences
    • timid: 1 occurrences
    • rough: 1 occurrences
    • Uncharacteristically: 1 occurrences
    • Elf King: 1 occurrences
    • Yandi: 1 occurrences
    • Thunder Emperor: 1 occurrences
    • Greedy: 1 occurrences

Must bring entry 2:
  - 5 valeurs uniques
  - Valeurs: ['none' 'holy day' 'Hades' 'ice King' 'Shenlong']

Numerical description:
  - 36 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • Attack power bonus%: 25 occurrences
    • Skill power: 13 occurrences
    • Skill power multiplier: 13 occurrences
    • Moving speed%: 13 occurrences
    • Ranch skills: 10 occurrences
  - Valeur la moins fréquente: Elemental power bonus%_Ice (1 occurrence(s))
```

```
Skill description:
  - 120 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • Can be moved around on its back. Two jumps are possible during the ride.: 3
occurrences
    • If it is in the team, the elemental Fire Palu's attack power will be increa
sed.: 2 occurrences
    • Assign it to Palu Ranch and it will have a chance to dig Money out of the g
round.: 2 occurrences
    • If it is in the party, players will be able to deal more damage when harves
ting.: 2 occurrences
    • After activation, the sixth sense can be activated to detect the location o
f nearby dungeons.: 2 occurrences
  - Valeur la moins fréquente: You can ride on its back and fly in the air. It ca
n also continuously attack with its missile launcher while riding. (1 occurrence
(s))

Valeurs aberrantes (méthode IQR):

Valeurs aberrantes (méthode IQR):
NamePrefixID: 0 valeurs aberrantes (0.0%)
IsPal: 0 valeurs aberrantes (0.0%)
variant: 33 valeurs aberrantes (23.9%)
  Premières 10: [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
rarity: 5 valeurs aberrantes (3.6%)
  Valeurs: [20, 20, 20, 20, 20]
Organization: 0 valeurs aberrantes (0.0%)
weapon: 0 valeurs aberrantes (0.0%)
WeaponEquip: 0 valeurs aberrantes (0.0%)
nocturnal: 25 valeurs aberrantes (18.1%)
  Premières 10: [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
total_4D: 0 valeurs aberrantes (0.0%)
HP: 0 valeurs aberrantes (0.0%)
melee attack: 6 valeurs aberrantes (4.3%)
  Valeurs: [140, 150, 150, 150, 150, 150]
Remote attack: 0 valeurs aberrantes (0.0%)
defense: 0 valeurs aberrantes (0.0%)
support: 22 valeurs aberrantes (15.9%)
  Premières 10: [30, 30, 50, 70, 70, 70, 70, 80, 90, 90]
Speed of work: 0 valeurs aberrantes (0.0%)
Level1Min: 0 valeurs aberrantes (0.0%)
Level1Max: 0 valeurs aberrantes (0.0%)
Level20Min: 0 valeurs aberrantes (0.0%)
Level20Max: 0 valeurs aberrantes (0.0%)
Level50Min: 0 valeurs aberrantes (0.0%)
Level50Max: 0 valeurs aberrantes (0.0%)
AISightResponse: 0 valeurs aberrantes (0.0%)
endurance: 15 valeurs aberrantes (10.9%)
  Premières 10: [130, 130, 150, 150, 150, 160, 170, 220, 230, 250]
slow walking speed: 1 valeurs aberrantes (0.7%)
  Valeurs: [600]
walking speed: 6 valeurs aberrantes (4.3%)
  Valeurs: [280, 350, 350, 350, 400, 800]
running speed: 1 valeurs aberrantes (0.7%)
  Valeurs: [1700]
Riding sprint speed: 1 valeurs aberrantes (0.7%)
  Valeurs: [3300]
Experience multiplier: 0 valeurs aberrantes (0.0%)
price: 0 valeurs aberrantes (0.0%)
lv1.1: 20 valeurs aberrantes (14.5%)
```

```
  Premières 10: [200.0, 220.0, 220.0, 250.0, 400.0, 500.0, 500.0, 500.0, 500.0, 5
00.0]
lv2.1: 22 valeurs aberrantes (15.9%)
  Premières 10: [130.0, 130.0, 220.0, 260.0, 260.0, 285.0, 440.0, 600.0, 750.0, 7
50.0]
lv3.1: 20 valeurs aberrantes (14.5%)
  Premières 10: [260.0, 320.0, 320.0, 325.0, 520.0, 700.0, 900.0, 1100.0, 1100.0,
1100.0]
lv4.1: 20 valeurs aberrantes (14.5%)
  Premières 10: [320.0, 400.0, 400.0, 400.0, 640.0, 800.0, 1050.0, 1500.0, 1500.
0, 1500.0]
lv5.1: 20 valeurs aberrantes (14.5%)
  Premières 10: [400.0, 500.0, 500.0, 500.0, 800.0, 900.0, 1200.0, 1700.0, 2000.
0, 2000.0]

======================================================================
```

In [49]:
```
explore_dataframe(hidden_attribute, 'Hidden Attribute')
```

```
===== Exploration de Hidden Attribute =====

--- Informations de base ---
Class: <class 'pandas.core.frame.DataFrame'>
Index: <class 'pandas.core.indexes.range.RangeIndex'> | RangeIndex(start=0, stop=
144, step=1)
Entries: 144 | Columns: 67
```

	Column	Non-Null Count	Dtype
0	OverrideNameTextID	144	object
1	OverridePartnerSkillTextID	144	object
2	IsPal	144	int64
3	Tribe	144	object
4	BPClass	144	object
...	...	...	...
62	pasture	144	int64
63	Passive skill 1	17	object
64	Passive skill 2	4	object
65	Passive skill 3	0	float64
66	Passive skill 4	0	float64

67 rows × 3 columns

```
Memory usage: 176.62 KB

Dimensions: 144 lignes, 67 colonnes

Aperçu des premières lignes:
```

	OverrideNameTextID	OverridePartnerSkillTextID	IsPal	Tribe	
0	SakuraSaurus_Water	PARTNERSKILL_SakuraSaurus_...	1	SakuraSaurus_Water	BOSS_Sak
1	AmaterasuWolf	PARTNERSKILL_AmaterasuWolf	1	AmaterasuWolf	BOSS
2	RobinHood	PARTNERSKILL_RobinHood	1	RobinHood	B

```
Types de données:
OverrideNameTextID             object
OverridePartnerSkillTextID     object
IsPal                           int64
Tribe                          object
BPClass                        object
                               ...
pasture                         int64
Passive skill 1                object
Passive skill 2                object
Passive skill 3                float64
Passive skill 4                float64
Length: 67, dtype: object
Statistiques descriptives:
```

	OverrideNameTextID	OverridePartnerSkillTextID	IsPal	Tribe	
count	144	144	144.0	144	
unique	143	143	NaN	138	
top	PlantSlime	PARTNERSKILL_PlantSlime	NaN	BlackGriffon	BOSS_SakuraS
freq	2	2	NaN	2	
mean	NaN	NaN	1.0	NaN	
std	NaN	NaN	0.0	NaN	
min	NaN	NaN	1.0	NaN	
25%	NaN	NaN	1.0	NaN	
50%	NaN	NaN	1.0	NaN	
75%	NaN	NaN	1.0	NaN	
max	NaN	NaN	1.0	NaN	

```
Valeurs manquantes:
```

	Total	Pourcentage
ZukanIndexSuffix	144	100.000000
Element 2	117	81.250000
Organization	144	100.000000
weapon	144	100.000000
AISightResponse	144	100.000000
Passive skill 1	127	88.194444
Passive skill 2	140	97.222222
Passive skill 3	144	100.000000
Passive skill 4	144	100.000000

```
Nombre de lignes dupliquées: 0

Analyse des colonnes catégorielles:

OverrideNameTextID:
  - 143 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • PlantSlime: 2 occurrences
    • AmaterasuWolf: 1 occurrences
    • SakuraSaurus_Water: 1 occurrences
    • RedArmorBird: 1 occurrences
    • ColorfulBird: 1 occurrences
  - Valeur la moins fréquente: LilyQueen_Dark (1 occurrence(s))

OverridePartnerSkillTextID:
  - 143 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • PARTNERSKILL_PlantSlime: 2 occurrences
    • PARTNERSKILL_AmaterasuWolf: 1 occurrences
    • PARTNERSKILL_SakuraSaurus_Water: 1 occurrences
    • PARTNERSKILL_RedArmorBird: 1 occurrences
    • PARTNERSKILL_ColorfulBird: 1 occurrences
  - Valeur la moins fréquente: PARTNERSKILL_LilyQueen_Dark (1 occurrence(s))

Tribe:
  - 138 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • BlackGriffon: 2 occurrences
    • PlantSlime: 2 occurrences
    • Horus: 2 occurrences
    • ElecPanda: 2 occurrences
    • LilyQueen: 2 occurrences
  - Valeur la moins fréquente: LilyQueen_Dark (1 occurrence(s))

BPClass:
  - 144 valeurs uniques
  - Top 5 des valeurs les plus fréquentes:
    • BOSS_SakuraSaurus_Water: 1 occurrences
    • BOSS_AmaterasuWolf: 1 occurrences
    • BOSS_RobinHood: 1 occurrences
    • BOSS_RedArmorBird: 1 occurrences
    • BOSS_ColorfulBird: 1 occurrences
  - Valeur la moins fréquente: BOSS_LilyQueen_Dark (1 occurrence(s))

Size:
  - 4 valeurs uniques
  - Valeurs: ['XL' 'L' 'S' 'M']

Element 1:
  - 9 valeurs uniques
  - Valeurs: ['Leaf' 'Fire' 'Normal' 'Dark' 'Water' 'Earth' 'Ice' 'Electricity'
 'Dragon']

Element 2:
  - 9 valeurs uniques
  - Valeurs: ['Water' nan 'Leaf' 'Earth' 'Ice' 'Electricity' 'Dark' 'Dragon' 'Fir
e']

GenusCategory:
  - 7 valeurs uniques
```

```
  - Valeurs: ['FourLegged' 'Humanoid' 'Bird' 'Dragon' 'Other' 'Monster' 'Fish']

damageMultiplier:
  - 4 valeurs uniques
  - Valeurs: ['0,2' '0,4' '0,1' '1']

CaptureProbability:
  - 7 valeurs uniques
  - Valeurs: ['0,7' '1,05' '0,91' '0,63' '0,77' '0,84' '1']

AIRResponse:
  - 3 valeurs uniques
  - Valeurs: ['NotInterested' 'Boss' 'Warlike']

BattleBGM:
  - 3 valeurs uniques
  - Valeurs: ['FieldBoss' 'Legend' 'TowerBoss']

Passive skill 1:
  - 12 valeurs uniques
  - Top 10 des valeurs les plus fréquentes:
    • Legend: 5 occurrences
    • Deffence_up1: 3 occurrences
    • ElementBoost_Earth_2_PAL: 1 occurrences
    • PAL_FullStomach_Up_1: 1 occurrences
    • PAL_ALLAttack_down1: 1 occurrences
    • ElementBoost_Thunder_2_PAL: 1 occurrences
    • ElementBoost_Fire_2_PAL: 1 occurrences
    • ElementBoost_Aqua_2_PAL: 1 occurrences
    • PAL_rude: 1 occurrences
    • ElementResist_Normal_1_PAL: 1 occurrences

Passive skill 2:
  - 5 valeurs uniques
  - Valeurs: [nan 'ElementBoost_Normal_2_PAL' 'ElementBoost_Dragon_2_PAL'
 'ElementBoost_Ice_2_PAL' 'ElementBoost_Dark_2_PAL']

Valeurs aberrantes (méthode IQR):

Valeurs aberrantes (méthode IQR):
IsPal: 0 valeurs aberrantes (0.0%)
Pictorial ID: 5 valeurs aberrantes (3.5%)
  Valeurs: [-2, -2, -2, -2, -2]
ZukanIndexSuffix: 0 valeurs aberrantes (0.0%)
rarity: 5 valeurs aberrantes (3.5%)
  Valeurs: [20, 20, 20, 20, 20]
Organization: 0 valeurs aberrantes (0.0%)
weapon: 0 valeurs aberrantes (0.0%)
WeaponEquip: 0 valeurs aberrantes (0.0%)
HP: 10 valeurs aberrantes (6.9%)
  Valeurs: [260, 260, 330, 420, 420, 5500, 6000, 6500, 6500, 8000]
melee attack: 6 valeurs aberrantes (4.2%)
  Valeurs: [140, 150, 150, 150, 150, 150]
Remote attack: 2 valeurs aberrantes (1.4%)
  Valeurs: [145, 200]
defense: 1 valeurs aberrantes (0.7%)
  Valeurs: [220]
support: 22 valeurs aberrantes (15.3%)
  Premières 10: [30, 30, 50, 70, 70, 80, 90, 90, 90, 90]
CraftSpeed: 0 valeurs aberrantes (0.0%)
```

```
ExperienceMultiplier: 5 valeurs aberrantes (3.5%)
  Valeurs: [30, 30, 30, 30, 30]
price: 0 valeurs aberrantes (0.0%)
AISightResponse: 0 valeurs aberrantes (0.0%)
slow walking speed: 1 valeurs aberrantes (0.7%)
  Valeurs: [600]
walking speed: 6 valeurs aberrantes (4.2%)
  Valeurs: [280, 350, 350, 350, 400, 800]
running speed: 1 valeurs aberrantes (0.7%)
  Valeurs: [1700]
Riding sprint speed: 4 valeurs aberrantes (2.8%)
  Valeurs: [1600, 1600, 1600, 3300]
Handling speed: 1 valeurs aberrantes (0.7%)
  Valeurs: [1250]
IsBoss: 0 valeurs aberrantes (0.0%)
IsTowerBoss: 5 valeurs aberrantes (3.5%)
  Valeurs: [1, 1, 1, 1, 1]
IgnoreLeanBack: 5 valeurs aberrantes (3.5%)
  Valeurs: [1, 1, 1, 1, 1]
IgnoreBlowAway: 5 valeurs aberrantes (3.5%)
  Valeurs: [1, 1, 1, 1, 1]
MaxFullStomach: 0 valeurs aberrantes (0.0%)
FullStomachDecreaseRate: 0 valeurs aberrantes (0.0%)
FoodAmount: 0 valeurs aberrantes (0.0%)
ViewingDistance: 0 valeurs aberrantes (0.0%)
ViewingAngle: 0 valeurs aberrantes (0.0%)
HearingRate: 0 valeurs aberrantes (0.0%)
NooseTrap: 0 valeurs aberrantes (0.0%)
Nocturnal: 26 valeurs aberrantes (18.1%)
  Premières 10: [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
BiologicalGrade: 1 valeurs aberrantes (0.7%)
  Valeurs: [0]
Predator: 0 valeurs aberrantes (0.0%)
Edible: 0 valeurs aberrantes (0.0%)
endurance: 16 valeurs aberrantes (11.1%)
  Premières 10: [70, 130, 130, 150, 150, 150, 160, 170, 220, 230]
Male probability: 15 valeurs aberrantes (10.4%)
  Premières 10: [0, 0, 10, 10, 20, 20, 30, 30, 30, 85]
fecundity: 5 valeurs aberrantes (3.5%)
  Valeurs: [9999, 9999, 9999, 9999, 9999]
Breathing fire: 24 valeurs aberrantes (16.7%)
  Premières 10: [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
watering: 15 valeurs aberrantes (10.4%)
  Premières 10: [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
planting: 22 valeurs aberrantes (15.3%)
  Premières 10: [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
generate electricity: 13 valeurs aberrantes (9.0%)
  Premières 10: [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
manual: 0 valeurs aberrantes (0.0%)
collection: 0 valeurs aberrantes (0.0%)
logging: 0 valeurs aberrantes (0.0%)
Mining: 23 valeurs aberrantes (16.0%)
  Premières 10: [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
pharmaceutical: 17 valeurs aberrantes (11.8%)
  Premières 10: [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
cool down: 18 valeurs aberrantes (12.5%)
  Premières 10: [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
carry: 0 valeurs aberrantes (0.0%)
pasture: 13 valeurs aberrantes (9.0%)
  Premières 10: [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
```

```
Passive skill 3: 0 valeurs aberrantes (0.0%)
Passive skill 4: 0 valeurs aberrantes (0.0%)

======================================================================
```

Création de la base de données et de ses tables

Création de la base de données

In [127…

```python
try:
    # Connexion sans spécifier de base de données pour pouvoir la créer
    conn = mariadb.connect(
        user='root',
        password='cN06+#P34',
        host='localhost',
        port=3307
    )
    cursor = conn.cursor()

    # Création de la base de données
    cursor.execute("CREATE DATABASE IF NOT EXISTS palworld_database")
    print("Base de données 'palworld_database' créée ou déjà existante.")

except mariadb.Error as e:
    print(f"Erreur : {e}")
    exit()

finally:
    if conn:
        conn.close()
```

```
Base de données 'palworld_database' créée ou déjà existante.
```

Création des tables, insertion des données et validations

In [155…

```python
# Création table job_skills & Insertion des données dans les tables

# === Config base de données ===
config = {
    'user': 'root',
    'password': 'cN06+#P34',
    'host': 'localhost',
    'port': 3307,
    'database': 'palworld_database'
}

# === Connexion ===
try:
    conn = mariadb.connect(**config)
    cursor = conn.cursor()
    print("Connexion réussie à MariaDB !")
except mariadb.Error as e:
    print(f"Erreur de connexion : {e}")
    exit(1)

# === Création de la table job_skills ===
create_table_query = """
```

```
CREATE TABLE IF NOT EXISTS job_skills (
    id INT AUTO_INCREMENT PRIMARY KEY,
    `English name` VARCHAR(255),
    `Chinese name` VARCHAR(255),
    `Volume size` VARCHAR(50),
    `Food intake` INT,
    `night shift` BOOLEAN,
    `Total skills` INT,
    `Make a fire` BOOLEAN,
    `watering` BOOLEAN,
    `planting` BOOLEAN,
    `generate electricity` BOOLEAN,
    `manual` BOOLEAN,
    `collection` BOOLEAN,
    `logging` BOOLEAN,
    `Mining` BOOLEAN,
    `pharmaceutical` BOOLEAN,
    `cool down` BOOLEAN,
    `pasture` BOOLEAN,
    `carry` BOOLEAN,
    `Handling speed` BOOLEAN,
    `ranch items` VARCHAR(255),
    `pasture minimum output` VARCHAR(255),
    `The largest ranch` VARCHAR(255)
);
"""

try:
    cursor.execute(create_table_query)
    conn.commit()
    print("Table 'job_skills' prête.")
except mariadb.Error as e:
    print(f"Erreur création table : {e}")
    conn.close()
    exit(1)

#  Lecture CSV
csv_file_path = r'C:\Users\Windows\Desktop\projets\1a\MariaDB\pals-analysis\data

try:
    df = pd.read_csv(csv_file_path)
    print(f"\nLe fichier CSV contient {df.shape[0]} lignes et {df.shape[1]} colo
except FileNotFoundError:
    print(f"Fichier non trouvé : {csv_file_path}")
    conn.close()
    exit(1)

# Nombre de colonnes
cursor.execute("SELECT COUNT(*) FROM information_schema.columns WHERE table_name
column_count = cursor.fetchone()[0]
print(f"La table 'job_skills' contient maintenant {column_count} colonnes.")

#  Fonction de conversion
def convert(value, col_name):
    if value == '' or str(value).lower() == 'none':
        return None
    if col_name.lower() in ['night shift', 'make a fire', 'watering', 'planting'
        return str(value).strip().lower() == 'true'
    value = value.replace(',', '.')
    try:
```

```
        if '.' in value:
            return float(value)
        return int(value)
    except ValueError:
        return value

# Fonction de nettoyage pour les colonnes name
def clean_name(name):
    if name is None:
        return None
    # Convertion en minuscules, supprime les underscores et les espaces
    return str(name).lower().replace('_', '').replace(' ', '')

# Insertion des données
try:
    with open(csv_file_path, mode='r', encoding='utf-8') as file:
        reader = csv.DictReader(file)
        headers = reader.fieldnames

        sql_headers = [h for h in headers]  #  noms initiaux
        placeholders = ', '.join(['%s'] * len(sql_headers))
        field_names = ', '.join(f'`{col}`' for col in sql_headers)
        insert_query = f"INSERT INTO job_skills ({field_names}) VALUES ({placeho

        count = 0
        for row in reader:
            values = []
            for i, h in enumerate(headers):
                value = convert(row[h], h)
                # Nettoyer spécifiquement les colonnes "English name" et "Chines
                if h.lower() in ['english name', 'chinese name']:
                    value = clean_name(value)
                values.append(value)
            try:
                cursor.execute(insert_query, values)
                count += 1
            except mariadb.Error as e:
                print(f"Erreur insertion ligne {count + 1}: {e} — valeurs: {valu

        conn.commit()
        print(f"\n{count} lignes insérées dans la table 'job_skills'.")
except FileNotFoundError:
    print(f"Fichier non trouvé : {csv_file_path}")

# Renommage des colonnes
rename_queries = [
    "ALTER TABLE job_skills CHANGE `English name` english_name VARCHAR(255);",
    "ALTER TABLE job_skills CHANGE `Chinese name` chinese_name VARCHAR(255);",
    "ALTER TABLE job_skills CHANGE `Volume size` volume_size VARCHAR(50);",
    "ALTER TABLE job_skills CHANGE `Food intake` food_intake INT;",
    "ALTER TABLE job_skills CHANGE `night shift` night_shift BOOLEAN;",
    "ALTER TABLE job_skills CHANGE `Total skills` total_skills INT;",
    "ALTER TABLE job_skills CHANGE `Make a fire` make_a_fire BOOLEAN;",
    "ALTER TABLE job_skills CHANGE `watering` watering BOOLEAN;",
    "ALTER TABLE job_skills CHANGE `planting` planting BOOLEAN;",
    "ALTER TABLE job_skills CHANGE `generate electricity` generate_electricity B
    "ALTER TABLE job_skills CHANGE `manual` manual BOOLEAN;",
    "ALTER TABLE job_skills CHANGE `collection` collection BOOLEAN;",
    "ALTER TABLE job_skills CHANGE `logging` logging BOOLEAN;",
    "ALTER TABLE job_skills CHANGE `Mining` mining BOOLEAN;",
```

```
    "ALTER TABLE job_skills CHANGE `pharmaceutical` pharmaceutical BOOLEAN;",
    "ALTER TABLE job_skills CHANGE `cool down` cool_down BOOLEAN;",
    "ALTER TABLE job_skills CHANGE `pasture` pasture BOOLEAN;",
    "ALTER TABLE job_skills CHANGE `carry` carry BOOLEAN;",
    "ALTER TABLE job_skills CHANGE `Handling speed` handling_speed BOOLEAN;",
    "ALTER TABLE job_skills CHANGE `ranch items` ranch_items VARCHAR(255);",
    "ALTER TABLE job_skills CHANGE `pasture minimum output` pasture_minimum_outp
    "ALTER TABLE job_skills CHANGE `The largest ranch` the_largest_ranch VARCHAR
]

print("\nRenommage des colonnes...")
for query in rename_queries:
    try:
        cursor.execute(query)
    except mariadb.Error as e:
        print(f"Erreur lors du renommage : {e}")

conn.commit()
print("Renommage terminé.")

# === Fermeture ===
cursor.close()
conn.close()
print("Connexion fermée.")
```

```
Connexion réussie à MariaDB !
Table 'job_skills' prête.

Le fichier CSV contient 138 lignes et 22 colonnes.
La table 'job_skills' contient maintenant 23 colonnes.

138 lignes insérées dans la table 'job_skills'.

Renommage des colonnes...
Renommage terminé.
Connexion fermée.
```

In [129…

```
# Création de la table palu_combat_attribute et insertion des données du csv

# Configuration de la base de données
config = {
    'user': 'root',
    'password': 'cN06+#P34',
    'host': 'localhost',
    'port': 3307,
    'database': 'palworld_database'
}

#  Connexion à MariaDB
try:
    conn = mariadb.connect(**config)
    cursor = conn.cursor()
    print("Connexion réussie à MariaDB !")
except mariadb.Error as e:
    print(f"Erreur de connexion : {e}")
    exit(1)

# Création de la table
create_table_query = """
CREATE TABLE IF NOT EXISTS palu_combat_attribute (
```

```
    id INT AUTO_INCREMENT PRIMARY KEY,
    `Chinese name` VARCHAR(255),
    `Name` VARCHAR(255),
    `CodeName` VARCHAR(255),
    `OverrideNameTextID` VARCHAR(255),
    `NamePrefixID` VARCHAR(255),
    `OverridePartnerSkillTextID` VARCHAR(255),
    `IsPal` BOOLEAN,
    `Tribe` VARCHAR(255),
    `BPClass` VARCHAR(255),
    `variant` VARCHAR(255),
    `Volume size` VARCHAR(255),
    `rarity` INT,
    `Element 1` VARCHAR(255),
    `Element 2` VARCHAR(255),
    `GenusCategory` VARCHAR(255),
    `Organization` VARCHAR(255),
    `weapon` VARCHAR(255),
    `WeaponEquip` VARCHAR(255),
    `nocturnal` BOOLEAN,
    `total_4D` INT,
    `HP` INT,
    `melee attack` INT,
    `Remote attack` INT,
    `defense` INT,
    `support` INT,
    `Speed of work` INT,
    `Level1Min` INT,
    `Level1Max` INT,
    `Level20Min` INT,
    `Level20Max` INT,
    `Level50Min` INT,
    `Level50Max` INT,
    `AIRResponse` VARCHAR(255),
    `AISightResponse` INT,
    `endurance` INT,
    `slow walking speed` INT,
    `walking speed` INT,
    `running speed` INT,
    `Riding sprint speed` INT,
    `damage multiplier` FLOAT,
    `catch rate` INT,
    `Experience multiplier` FLOAT,
    `price` INT,
    `Must bring entry 1` VARCHAR(255),
    `Must bring entry 2` VARCHAR(255),
    `Numerical description` VARCHAR(255),
    `Skill description` TEXT,
    `lv1.1` INT,
    `lv2.1` INT,
    `lv3.1` INT,
    `lv4.1` INT,
    `lv5.1` INT
);
"""

try:
    cursor.execute(create_table_query)
    conn.commit()
    print("Table 'palu_combat_attribute' prête.")
```

```
except mariadb.Error as e:
    print(f"Erreur création table : {e}")
    conn.close()
    exit(1)

# Lecture CSV
csv_file_path = r'C:\Users\Windows\Desktop\projets\1a\MariaDB\pals-analysis\data

try:
    df = pd.read_csv(csv_file_path)
    print(f"\nLe fichier CSV contient {df.shape[0]} lignes et {df.shape[1]} colo
except FileNotFoundError:
    print(f"Fichier non trouvé : {csv_file_path}")
    conn.close()
    exit(1)

# Nombre de colonnes dans la table
cursor.execute("SELECT COUNT(*) FROM information_schema.columns WHERE table_name
column_count = cursor.fetchone()[0]
print(f"La table 'palu_combat_attribute' contient maintenant {column_count} colo

# Fonction pour convertir les valeurs
def convert(value, col_name):
    if value == '' or str(value).lower() == 'none':
        return None
    if col_name.lower() in ['ispal', 'nocturnal']:
        return str(value).strip().lower() == 'true'
    value = value.replace(',', '.')
    try:
        if '.' in value:
            return float(value)
        return int(value)
    except ValueError:
        return value

# Fonction de nettoyage pour les colonnes name
def clean_name(name):
    if name is None:
        return None
    # Convertir en minuscules, supprimer les underscores et les espaces
    return str(name).lower().replace('_', '').replace(' ', '')

#  Insertion des données
try:
    with open(csv_file_path, mode='r', encoding='utf-8') as file:
        reader = csv.DictReader(file)
        headers = reader.fieldnames

        sql_headers = [h for h in headers]  # on conserve les noms initiaux
        placeholders = ', '.join(['%s'] * len(sql_headers))
        field_names = ', '.join(f'`{col}`' for col in sql_headers)
        insert_query = f"INSERT INTO palu_combat_attribute ({field_names}) VALUE

        count = 0
        for row in reader:
            values = []
            for i, h in enumerate(headers):
                value = convert(row[h], h)
                # Nettoyer spécifiquement les colonnes "Chinese name", "Name" et
                if h.lower() in ['chinese name', 'name', 'codename']:
```

```
                    value = clean_name(value)
                values.append(value)
            try:
                cursor.execute(insert_query, values)
                count += 1
            except mariadb.Error as e:
                print(f"Erreur insertion ligne {count + 1}: {e} — valeurs: {valu

        conn.commit()
        print(f"\n{count} lignes insérées dans la table 'palu_combat_attribute'.
except FileNotFoundError:
    print(f"Fichier non trouvé : {csv_file_path}")

# Renommage des colonnes
rename_queries = [
    "ALTER TABLE palu_combat_attribute CHANGE `Chinese name` chinese_name VARCHA
    "ALTER TABLE palu_combat_attribute CHANGE `Name` name VARCHAR(255);",
    "ALTER TABLE palu_combat_attribute CHANGE `CodeName` code_name VARCHAR(255);
    "ALTER TABLE palu_combat_attribute CHANGE `OverrideNameTextID` override_name
    "ALTER TABLE palu_combat_attribute CHANGE `NamePrefixID` name_prefix_id VARC
    "ALTER TABLE palu_combat_attribute CHANGE `OverridePartnerSkillTextID` overr
    "ALTER TABLE palu_combat_attribute CHANGE `IsPal` is_pal BOOLEAN;",
    "ALTER TABLE palu_combat_attribute CHANGE `Tribe` tribe VARCHAR(255);",
    "ALTER TABLE palu_combat_attribute CHANGE `BPClass` bp_class VARCHAR(255);",
    "ALTER TABLE palu_combat_attribute CHANGE `Volume size` volume_size VARCHAR(
    "ALTER TABLE palu_combat_attribute CHANGE `Element 1` element_1 VARCHAR(255)
    "ALTER TABLE palu_combat_attribute CHANGE `Element 2` element_2 VARCHAR(255)
    "ALTER TABLE palu_combat_attribute CHANGE `GenusCategory` genus_category VAR
    "ALTER TABLE palu_combat_attribute CHANGE `Organization` organization VARCHA
    "ALTER TABLE palu_combat_attribute CHANGE `weapon` weapon VARCHAR(255);",
    "ALTER TABLE palu_combat_attribute CHANGE `WeaponEquip` weapon_equip VARCHAR
    "ALTER TABLE palu_combat_attribute CHANGE `nocturnal` nocturnal BOOLEAN;",
    "ALTER TABLE palu_combat_attribute CHANGE `total_4D` total_4d INT;",
    "ALTER TABLE palu_combat_attribute CHANGE `melee attack` melee_attack INT;",
    "ALTER TABLE palu_combat_attribute CHANGE `Remote attack` remote_attack INT;
    "ALTER TABLE palu_combat_attribute CHANGE `Speed of work` speed_of_work INT;
    "ALTER TABLE palu_combat_attribute CHANGE `Level1Min` level1_min INT;",
    "ALTER TABLE palu_combat_attribute CHANGE `Level1Max` level1_max INT;",
    "ALTER TABLE palu_combat_attribute CHANGE `Level20Min` level20_min INT;",
    "ALTER TABLE palu_combat_attribute CHANGE `Level20Max` level20_max INT;",
    "ALTER TABLE palu_combat_attribute CHANGE `Level50Min` level50_min INT;",
    "ALTER TABLE palu_combat_attribute CHANGE `Level50Max` level50_max INT;",
    "ALTER TABLE palu_combat_attribute CHANGE `AIRResponse` air_response VARCHAR
    "ALTER TABLE palu_combat_attribute CHANGE `AISightResponse` ai_sight_respons
    "ALTER TABLE palu_combat_attribute CHANGE `slow walking speed` slow_walking_
    "ALTER TABLE palu_combat_attribute CHANGE `walking speed` walking_speed INT;
    "ALTER TABLE palu_combat_attribute CHANGE `running speed` running_speed INT;
    "ALTER TABLE palu_combat_attribute CHANGE `Riding sprint speed` riding_sprin
    "ALTER TABLE palu_combat_attribute CHANGE `damage multiplier` damage_multipl
    "ALTER TABLE palu_combat_attribute CHANGE `catch rate` catch_rate INT;",
    "ALTER TABLE palu_combat_attribute CHANGE `Experience multiplier` experience
    "ALTER TABLE palu_combat_attribute CHANGE `Must bring entry 1` must_bring_en
    "ALTER TABLE palu_combat_attribute CHANGE `Must bring entry 2` must_bring_en
    "ALTER TABLE palu_combat_attribute CHANGE `Numerical description` numerical_
    "ALTER TABLE palu_combat_attribute CHANGE `Skill description` skill_descript
    "ALTER TABLE palu_combat_attribute CHANGE `lv1.1` lv1_1 INT;",
    "ALTER TABLE palu_combat_attribute CHANGE `lv2.1` lv2_1 INT;",
    "ALTER TABLE palu_combat_attribute CHANGE `lv3.1` lv3_1 INT;",
    "ALTER TABLE palu_combat_attribute CHANGE `lv4.1` lv4_1 INT;",
    "ALTER TABLE palu_combat_attribute CHANGE `lv5.1` lv5_1 INT;"
```

```
]

print("\nRenommage des colonnes...")
for query in rename_queries:
    try:
        cursor.execute(query)
    except mariadb.Error as e:
        print(f"Erreur lors du renommage : {e}")

conn.commit()
print("Renommage terminé.")

# Fermeture connexion
cursor.close()
conn.close()
print("Connexion fermée.")
```

```
Connexion réussie à MariaDB !
Table 'palu_combat_attribute' prête.

Le fichier CSV contient 138 lignes et 52 colonnes.
La table 'palu_combat_attribute' contient maintenant 53 colonnes.

138 lignes insérées dans la table 'palu_combat_attribute'.

Renommage des colonnes...
Renommage terminé.
Connexion fermée.
```

In [130…

```
# Création table refresh et insertion des données du csv
# Configuration de la base de données
config = {
    'user': 'root',
    'password': 'cN06+#P34',
    'host': 'localhost',
    'port': 3307,
    'database': 'palworld_database'
}
# === Connexion à MariaDB ===
try:
    conn = mariadb.connect(**config)
    cursor = conn.cursor()
    print("Connexion réussie à MariaDB !")
except mariadb.Error as e:
    print(f"Erreur de connexion : {e}")
    exit(1)
# === Création de la table avec tous les champs ===
create_table_query = """
CREATE TABLE IF NOT EXISTS palu_refresh_level (
    id INT AUTO_INCREMENT PRIMARY KEY,
    `name` VARCHAR(255),
    `minimum_level` INT,
    `maximum_level` INT,
    `fecondity` INT,
    `pallu_refresh_type` VARCHAR(255),
    `night_only` BOOLEAN,
    `refresh_area` VARCHAR(255)
);
"""
try:
```

```
    cursor.execute(create_table_query)
    conn.commit()
    print("Table 'palu_refresh_level' prête.")
except mariadb.Error as e:
    print(f"Erreur création table : {e}")
    conn.close()
    exit(1)
# === Lecture rapide CSV avec pandas pour afficher nombre lignes/colonnes ===
csv_file_path = r'C:\Users\Windows\Desktop\projets\1a\MariaDB\pals-analysis\data
try:
    df = pd.read_csv(csv_file_path)
    print(f"\nLe fichier CSV contient {df.shape[0]} lignes et {df.shape[1]} colo
except FileNotFoundError:
    print(f"Fichier non trouvé : {csv_file_path}")
    conn.close()
    exit(1)
# Fonction pour convertir les valeurs
def convert(value, field=None):
    if value == '' or str(value).lower() == 'none':
        return None
    value = str(value).replace(',', '.')  # gestion des virgules décimales
    if field == 'night_only':
        return 1 if str(value).lower() == 'true' else 0
    try:
        if '.' in value:
            return float(value)
        return int(value)
    except ValueError:
        return value  # retourner chaîne

#  Fonction de nettoyage pour la colonne name
def clean_name(name):
    if name is None:
        return None
    # Convertir en minuscules, supprimer les underscores et les espaces
    return str(name).lower().replace('_', '').replace(' ', '')

# === Lecture CSV & insertion classique avec csv.DictReader ===
try:
    with open(csv_file_path, mode='r', encoding='utf-8') as file:
        reader = csv.DictReader(file)
        headers = reader.fieldnames
        # Transformation noms colonnes pour SQL
        sql_headers = [h.strip().replace(' ', '_').replace('.', '_').replace('-'
        placeholders = ', '.join(['%s'] * len(headers))
        field_names = ', '.join(f'`{col}`' for col in sql_headers)
        insert_query = f"INSERT INTO palu_refresh_level ({field_names}) VALUES (
        count = 0
        for row in reader:
            values = []
            for i, h in enumerate(headers):
                value = convert(row[h], h.replace(' ', '_'))
                # Nettoyer spécifiquement la colonne "name"
                if h.lower() == 'name' or sql_headers[i].lower() == 'name':
                    value = clean_name(value)
                values.append(value)
            try:
                cursor.execute(insert_query, values)
                count += 1
            except mariadb.Error as e:
```

```
                print(f"Erreur insertion ligne {count + 1} : {e} — valeurs: {val
        conn.commit()
        print(f"\n{count} lignes insérées dans la table 'palu_refresh_level'.")
except FileNotFoundError:
    print(f"Fichier non trouvé : {csv_file_path}")
# === Fermeture connexion ===
cursor.close()
conn.close()
print("Connexion fermée.")
```

```
Connexion réussie à MariaDB !
Table 'palu_refresh_level' prête.

Le fichier CSV contient 136 lignes et 7 colonnes.

136 lignes insérées dans la table 'palu_refresh_level'.
Connexion fermée.
```

In [131…

```
## Création table & Insertion des données CSV dans la table boss_comparison
#  Config base de données
config = {
    'user': 'root',
    'password': 'cN06+#P34',
    'host': 'localhost',
    'port': 3307,
    'database': 'palworld_database'
}
#  Connexion
try:
    conn = mariadb.connect(**config)
    cursor = conn.cursor()
    print("Connexion réussie à MariaDB !")
except mariadb.Error as e:
    print(f"Erreur de connexion : {e}")
    exit(1)
# === Création de la table adaptée ===
create_table_query = """
CREATE TABLE IF NOT EXISTS ordinary_boss_comparison (
    id INT AUTO_INCREMENT PRIMARY KEY,
    `name` VARCHAR(255),
    `HP` INT,
    `Remote_attack` INT,
    `Riding_speed_BOSS_is_100_higher` INT
);
"""
try:
    cursor.execute(create_table_query)
    conn.commit()
    print("Table 'ordinary_boss_comparison' prête.")
except mariadb.Error as e:
    print(f"Erreur création table : {e}")
    conn.close()
    exit(1)
#  Affichage nombre colonnes et lignes
try:
    cursor.execute("SELECT COUNT(*) FROM ordinary_boss_comparison")
    count_rows = cursor.fetchone()[0]
    count_columns = len([column[0] for column in cursor.description])
    print(f"\nAvant insertion : {count_rows} lignes, {count_columns} colonnes da
except mariadb.Error as e:
```

```
    print(f"Erreur comptage lignes table : {e}")
# === Lecture rapide CSV avec pandas pour afficher nombre lignes/colonnes ===
csv_file_path = r'C:\Users\Windows\Desktop\projets\1a\MariaDB\pals-analysis\data
try:
    df = pd.read_csv(csv_file_path)
    print(f"\nLe fichier CSV contient {df.shape[0]} lignes et {df.shape[1]} colo
except FileNotFoundError:
    print(f"Fichier non trouvé : {csv_file_path}")
    conn.close()
    exit(1)
# Fonction de conversion adaptée
def convert(value):
    if value == '' or str(value).lower() == 'none':
        return None
    value = str(value).replace(',', '.')  # gestion décimales
    try:
        if '.' in value:
            return float(value)
        return int(value)
    except ValueError:
        return value  # texte

#  Fonction de nettoyage pour  name
def clean_name(name):
    if name is None:
        return None
    # Convertir en minuscules, supprimer les underscores et les espaces
    return str(name).lower().replace('_', '').replace(' ', '')

# Lecture CSV & insertion
try:
    with open(csv_file_path, mode='r', encoding='utf-8') as file:
        reader = csv.DictReader(file)
        headers = reader.fieldnames
        # Adaptation noms colonnes pour SQL
        sql_headers = [h.replace(' ', '_').replace('(', '').replace(')', '') for
        placeholders = ', '.join(['%s'] * len(headers))
        field_names = ', '.join(f'`{col}`' for col in sql_headers)
        insert_query = f"INSERT INTO ordinary_boss_comparison ({field_names}) VA
        count = 0
        for row in reader:
            values = []
            for i, h in enumerate(headers):
                value = convert(row[h])
                # Nettoyer  la colonne "name"
                if h.lower() == 'name' or sql_headers[i].lower() == 'name':
                    value = clean_name(value)
                values.append(value)
            try:
                cursor.execute(insert_query, values)
                count += 1
            except mariadb.Error as e:
                print(f"Erreur insertion ligne {count + 1}: {e} — valeurs: {valu
        conn.commit()
        print(f"\n{count} lignes insérées dans la table 'ordinary_boss_compariso
except FileNotFoundError:
    print(f"Fichier non trouvé : {csv_file_path}")
# Fermeture connexion
cursor.close()
```

```
conn.close()
print("Connexion fermée.")
```

```
Connexion réussie à MariaDB !
Table 'ordinary_boss_comparison' prête.

Avant insertion : 0 lignes, 1 colonnes dans la table 'ordinary_boss_comparison'.

Le fichier CSV contient 129 lignes et 4 colonnes.

129 lignes insérées dans la table 'ordinary_boss_comparison'.
Connexion fermée.
```

In [137…

```
# Création table tower_bosses & Insertion des données dans les tables

#  Importations nécessaires
import mariadb
import pandas as pd
import csv

# Config base de données
config = {
    'user': 'root',
    'password': 'cN06+#P34',
    'host': 'localhost',
    'port': 3307,
    'database': 'palworld_database'
}

# Connexion
try:
    conn = mariadb.connect(**config)
    cursor = conn.cursor()
    print("Connexion réussie à MariaDB !")
except mariadb.Error as e:
    print(f"Erreur de connexion : {e}")
    exit(1)

# Création de la table tower_bosses
create_table_query = """
CREATE TABLE IF NOT EXISTS tower_bosses (
    id INT AUTO_INCREMENT PRIMARY KEY,
    `name` VARCHAR(255),
    `HP` INT,
    `melee attack` INT,
    `remote attack` INT,
    `defense` INT,
    `Support` INT,
    `experience ratio` INT,
    `slow walking speed` INT,
    `walking speed` INT,
    `running speed` INT,
    `riding speed` INT,
    `Handling speed` INT,
    `ignore the bluntness` BOOLEAN,
    `ignore displacement` BOOLEAN,
    `BiologicalGrade` INT,
    `endurance` INT,
    `fecundity` INT
);
```

```
"""

try:
    cursor.execute(create_table_query)
    conn.commit()
    print("Table 'tower_bosses' prête.")
except mariadb.Error as e:
    print(f"Erreur création table : {e}")
    conn.close()
    exit(1)

#  Lecture CSV
csv_file_path = r'C:\Users\Windows\Desktop\projets\1a\MariaDB\pals-analysis\data

try:
    df = pd.read_csv(csv_file_path)
    print(f"\nLe fichier CSV contient {df.shape[0]} lignes et {df.shape[1]} colo
except FileNotFoundError:
    print(f"Fichier non trouvé : {csv_file_path}")
    conn.close()
    exit(1)

# Nombre de colonnes dans la table tower_bosses
cursor.execute("SELECT COUNT(*) FROM information_schema.columns WHERE table_name
column_count = cursor.fetchone()[0]
print(f"La table 'tower_bosses' contient maintenant {column_count} colonnes.")

#  Fonction de conversion
def convert(value, col_name):
    if value == '' or str(value).lower() == 'none':
        return None
    if col_name.lower() in ['ignore the bluntness', 'ignore displacement']:
        return str(value).strip().lower() == 'true'
    value = str(value).replace(',', '.')
    try:
        if '.' in value:
            return float(value)
        return int(value)
    except ValueError:
        return value

# Fonction de nettoyage pour name
def clean_name(name):
    if name is None:
        return None
    # Convertir en minuscules, supprimer les underscores et les espaces
    return str(name).lower().replace('_', '').replace(' ', '')

#  Insertion des données dans tower_bosses
try:
    with open(csv_file_path, mode='r', encoding='utf-8') as file:
        reader = csv.DictReader(file)
        headers = reader.fieldnames

        # Encadrement des noms de colonnes avec des backticks pour gérer les esp
        sql_headers = [f'`{h}`' for h in headers]
        placeholders = ', '.join(['%s'] * len(sql_headers))
        field_names = ', '.join(sql_headers)
        insert_query = f"INSERT INTO tower_bosses ({field_names}) VALUES ({place
```

```
        count = 0
        for row in reader:
            values = []
            for i, h in enumerate(headers):
                value = convert(row[h], h)
                # Nettoyer spécifiquement la colonne "name"
                if h.lower() == 'name':
                    value = clean_name(value)
                values.append(value)
            try:
                cursor.execute(insert_query, values)
                count += 1
            except mariadb.Error as e:
                print(f"Erreur insertion ligne {count + 1}: {e} — valeurs: {valu

        conn.commit()
        print(f"\n{count} lignes insérées dans la table 'tower_bosses'.")
except FileNotFoundError:
    print(f"Fichier non trouvé : {csv_file_path}")

# Renommage des colonnes
rename_queries = [
    "ALTER TABLE tower_bosses CHANGE `name` name VARCHAR(255);",
    "ALTER TABLE tower_bosses CHANGE `melee attack` melee_attack INT;",
    "ALTER TABLE tower_bosses CHANGE `remote attack` remote_attack INT;",
    "ALTER TABLE tower_bosses CHANGE `Support` support INT;",
    "ALTER TABLE tower_bosses CHANGE `experience ratio` experience_ratio INT;",
    "ALTER TABLE tower_bosses CHANGE `slow walking speed` slow_walking_speed INT
    "ALTER TABLE tower_bosses CHANGE `walking speed` walking_speed INT;",
    "ALTER TABLE tower_bosses CHANGE `running speed` running_speed INT;",
    "ALTER TABLE tower_bosses CHANGE `riding speed` riding_speed INT;",
    "ALTER TABLE tower_bosses CHANGE `Handling speed` handling_speed INT;",
    "ALTER TABLE tower_bosses CHANGE `ignore the bluntness` ignore_the_bluntness
    "ALTER TABLE tower_bosses CHANGE `ignore displacement` ignore_displacement B
    "ALTER TABLE tower_bosses CHANGE `BiologicalGrade` biological_grade INT;"
]

print("\nRenommage des colonnes...")
for query in rename_queries:
    try:
        cursor.execute(query)
    except mariadb.Error as e:
        print(f"Erreur lors du renommage : {e}")

conn.commit()
print("Renommage terminé.")

# === Fermeture ===
cursor.close()
conn.close()
print("Connexion fermée.")
```

```
Connexion réussie à MariaDB !
Table 'tower_bosses' prête.

Le fichier CSV contient 10 lignes et 17 colonnes.
La table 'tower_bosses' contient maintenant 18 colonnes.

10 lignes insérées dans la table 'tower_bosses'.

Renommage des colonnes...
Renommage terminé.
Connexion fermée.
```

In [132…

```python
#Création et remplissage de la table monsters
# Config base de données
config = {
    'user': 'root',
    'password': 'cN06+#P34',
    'host': 'localhost',
    'port': 3307,
    'database': 'palworld_database'
}

# Connexion
try:
    conn = mariadb.connect(**config)
    cursor = conn.cursor()
    print("Connexion réussie à MariaDB !")
except mariadb.Error as e:
    print(f"Erreur de connexion : {e}")
    exit(1)

#  Création de la table
create_table_query = """
CREATE TABLE IF NOT EXISTS monsters (
    id INT AUTO_INCREMENT PRIMARY KEY,
    `OverrideNameTextID` VARCHAR(255),
    `OverridePartnerSkillTextID` VARCHAR(255),
    `IsPal` INT,
    `Tribe` VARCHAR(255),
    `BPClass` VARCHAR(255),
    `Pictorial ID` INT,
    `ZukanIndexSuffix` FLOAT,
    `Size` VARCHAR(50),
    `rarity` INT,
    `Element 1` VARCHAR(50),
    `Element 2` VARCHAR(50),
    `GenusCategory` VARCHAR(50),
    `Organization` FLOAT,
    `weapon` FLOAT,
    `WeaponEquip` INT,
    `HP` INT,
    `melee attack` INT,
    `Remote attack` INT,
    `defense` INT,
    `support` INT,
    `CraftSpeed` INT,
    `damageMultiplier` VARCHAR(50),
    `CaptureProbability` VARCHAR(50),
    `ExperienceMultiplier` INT,
    `price` INT,
```

```
    `AIRResponse` VARCHAR(50),
    `AISightResponse` FLOAT,
    `slow walking speed` INT,
    `walking speed` INT,
    `running speed` INT,
    `Riding sprint speed` INT,
    `Handling speed` INT,
    `IsBoss` INT,
    `IsTowerBoss` INT,
    `BattleBGM` VARCHAR(255),
    `IgnoreLeanBack` INT,
    `IgnoreBlowAway` INT,
    `MaxFullStomach` INT,
    `FullStomachDecreaseRate` INT,
    `FoodAmount` INT,
    `ViewingDistance` INT,
    `ViewingAngle` INT,
    `HearingRate` INT,
    `NooseTrap` INT,
    `Nocturnal` INT,
    `BiologicalGrade` INT,
    `Predator` INT,
    `Edible` INT,
    `endurance` INT,
    `Male probability` INT,
    `fecundity` INT,
    `Breathing fire` INT,
    `watering` INT,
    `planting` INT,
    `generate electricity` INT,
    `manual` INT,
    `collection` INT,
    `logging` INT,
    `Mining` INT,
    `pharmaceutical` INT,
    `cool down` INT,
    `carry` INT,
    `pasture` INT,
    `Passive skill 1` VARCHAR(255),
    `Passive skill 2` VARCHAR(255),
    `Passive skill 3` FLOAT,
    `Passive skill 4` FLOAT
);
"""

try:
    cursor.execute(create_table_query)
    conn.commit()
    print("Table 'monsters' prête.")
except mariadb.Error as e:
    print(f"Erreur création table : {e}")
    conn.close()
    exit(1)

#  Fonction de conversion des valeurs
def convert(value):
    if value == '' or value == 'None':
        return None
    try:
        value = value.replace(',', '.')
```

```
        if '.' in value:
            return float(value)
        return int(value)
    except ValueError:
        return value

# Fonction de nettoyage pour OverrideNameTextID
def clean_override_name(name):
    if name is None:
        return None
    # Convertir en minuscules, supprimer les underscores et les espaces
    return str(name).lower().replace('_', '').replace(' ', '')

# Chemin vers le fichier CSV
csv_file_path = r'C:\Users\Windows\Desktop\projets\1a\MariaDB\pals-analysis\data

# === Lecture du CSV et insertion ===
try:
    with open(csv_file_path, mode='r', encoding='utf-8') as file:
        reader = csv.DictReader(file)
        headers = reader.fieldnames

        placeholders = ', '.join(['%s'] * len(headers))
        field_names = ', '.join(f'`{h}`' for h in headers)
        insert_query = f"INSERT INTO monsters ({field_names}) VALUES ({placehold

        count = 0
        for row in reader:
            values = []
            for h in headers:
                value = convert(row[h])
                # Nettoyer spécifiquement la colonne "OverrideNameTextID"
                if h == 'OverrideNameTextID':
                    value = clean_override_name(value)
                values.append(value)
            try:
                cursor.execute(insert_query, values)
                count += 1
            except mariadb.Error as e:
                print(f"Erreur insertion ligne {count + 1}: {e} — valeurs: {valu

        conn.commit()
        print(f"{count} lignes insérées dans la table 'monsters'.")
except FileNotFoundError:
    print(f"Fichier non trouvé : {csv_file_path}")

# Renommage des colonnes
print("Renommage des colonnes en cours...")

rename_queries = [
    "ALTER TABLE monsters CHANGE `Pictorial ID` `Pictorial_ID` INT",
    "ALTER TABLE monsters CHANGE `Element 1` `Element_1` VARCHAR(50)",
    "ALTER TABLE monsters CHANGE `Element 2` `Element_2` VARCHAR(50)",
    "ALTER TABLE monsters CHANGE `melee attack` `melee_attack` INT",
    "ALTER TABLE monsters CHANGE `Remote attack` `Remote_attack` INT",
    "ALTER TABLE monsters CHANGE `slow walking speed` `slow_walking_speed` INT",
    "ALTER TABLE monsters CHANGE `walking speed` `walking_speed` INT",
    "ALTER TABLE monsters CHANGE `running speed` `running_speed` INT",
    "ALTER TABLE monsters CHANGE `Riding sprint speed` `Riding_sprint_speed` INT
    "ALTER TABLE monsters CHANGE `Handling speed` `Handling_speed` INT",
```

```
    "ALTER TABLE monsters CHANGE `Male probability` `Male_probability` INT",
    "ALTER TABLE monsters CHANGE `Breathing fire` `Breathing_fire` INT",
    "ALTER TABLE monsters CHANGE `generate electricity` `generate_electricity` I
    "ALTER TABLE monsters CHANGE `cool down` `cool_down` INT",
    "ALTER TABLE monsters CHANGE `Passive skill 1` `Passive_skill_1` VARCHAR(255
    "ALTER TABLE monsters CHANGE `Passive skill 2` `Passive_skill_2` VARCHAR(255
    "ALTER TABLE monsters CHANGE `Passive skill 3` `Passive_skill_3` FLOAT",
    "ALTER TABLE monsters CHANGE `Passive skill 4` `Passive_skill_4` FLOAT"
]

try:
    for query in rename_queries:
        cursor.execute(query)

    conn.commit()
    print("Toutes les colonnes ont été renommées avec succès !")

except mariadb.Error as e:
    print(f"Erreur lors du renommage : {e}")

# Comptage colonnes et lignes CSV ET table
try:
    with open(csv_file_path, mode='r', encoding='utf-8') as f:
        reader = csv.reader(f)
        csv_rows = list(reader)
        csv_col_count = len(csv_rows[0]) if csv_rows else 0
        csv_row_count = len(csv_rows) - 1 if len(csv_rows) > 0 else 0

    print(f"\nCSV : {csv_row_count} lignes, {csv_col_count} colonnes.")

    cursor.execute("SHOW COLUMNS FROM monsters")
    db_columns = cursor.fetchall()
    db_col_count = len(db_columns)

    cursor.execute("SELECT COUNT(*) FROM monsters")
    db_row_count = cursor.fetchone()[0]

    print(f"Table 'monsters' : {db_row_count} lignes, {db_col_count} colonnes.")

except Exception as e:
    print(f"Erreur lors du comptage des lignes/colonnes : {e}")

# Fermeture
cursor.close()
conn.close()
print("\nConnexion fermée.")
```

```
Connexion réussie à MariaDB !
Table 'monsters' prête.
144 lignes insérées dans la table 'monsters'.
Renommage des colonnes en cours...
Toutes les colonnes ont été renommées avec succès !

CSV : 144 lignes, 67 colonnes.
Table 'monsters' : 144 lignes, 68 colonnes.

Connexion fermée.
```

Afficharges des informations des tables créées

In [159…

```
#Affichage de la structure et des données de job_skills

# Reconnexion à la base de données
try:
    conn = mariadb.connect(**config)
    cursor = conn.cursor()
    print("Connexion réussie à MariaDB !")
except mariadb.Error as e:
    print(f"Erreur de connexion : {e}")
    exit(1)

# Affichage de la structure de la table
print("\n=== Structure de la table 'job_skills' ===")
try:
    cursor.execute("DESCRIBE job_skills")
    structure = cursor.fetchall()
    for column in structure:
        print(f"Colonne: {column[0]:<25} Type: {column[1]}")
except mariadb.Error as e:
    print(f"Erreur récupération structure : {e}")

# Affichage des 3 premières lignes
print("\n=== 3 premières lignes de la table ===")
try:
    cursor.execute("SELECT * FROM job_skills LIMIT 3")
    rows = cursor.fetchall()

    # Récupération des noms de colonnes
    cursor.execute("SELECT column_name FROM information_schema.columns WHERE tab
    columns = [col[0] for col in cursor.fetchall()]

    # Affichage pandas
    import pandas as pd
    df = pd.DataFrame(rows, columns=columns)
    print(df)
except mariadb.Error as e:
    print(f"Erreur récupération données : {e}")

# Fermeture
cursor.close()
conn.close()
```

```
Connexion réussie à MariaDB !

=== Structure de la table 'job_skills' ===
Colonne: id                        Type: int(11)
Colonne: english_name              Type: varchar(255)
Colonne: chinese_name              Type: varchar(255)
Colonne: volume_size               Type: varchar(50)
Colonne: food_intake               Type: int(11)
Colonne: night_shift               Type: tinyint(1)
Colonne: total_skills              Type: int(11)
Colonne: make_a_fire               Type: tinyint(1)
Colonne: watering                  Type: tinyint(1)
Colonne: planting                  Type: tinyint(1)
Colonne: generate_electricity      Type: tinyint(1)
Colonne: manual                    Type: tinyint(1)
Colonne: collection                Type: tinyint(1)
Colonne: logging                   Type: tinyint(1)
Colonne: mining                    Type: tinyint(1)
Colonne: pharmaceutical            Type: tinyint(1)
Colonne: cool_down                 Type: tinyint(1)
Colonne: pasture                   Type: tinyint(1)
Colonne: carry                     Type: tinyint(1)
Colonne: handling_speed            Type: tinyint(1)
Colonne: ranch_items               Type: varchar(255)
Colonne: pasture_minimum_output    Type: varchar(255)
Colonne: the_largest_ranch         Type: varchar(255)

=== 3 premières lignes de la table ===
   id english_name  chinese_name volume_size  food_intake  night_shift  \
0   1      lifmunk  greenleafrat    smallest            1            0
1   2     foxparks     tinderfox    smallest            2            0
2   3        rooby      firedeer       Small            3            0

   total_skills  make_a_fire  watering  planting  ...  logging  mining  \
0             5            0         0         1  ...        1       0
1             1            1         0         0  ...        0       0
2             1            1         0         0  ...        0       0

   pharmaceutical  cool_down  pasture  carry  handling_speed  ranch_items  \
0               1          0        0      0            None         None
1               0          0        0      0            None         None
2               0          0        0      0            None         None

   pasture_minimum_output the_largest_ranch
0                    None              None
1                    None              None
2                    None              None

[3 rows x 23 columns]
```

In [162…

```python
# Affichage de la structure et des données de la table palu_combat_attribute

# Reconnexion
try:
    conn = mariadb.connect(**config)
    cursor = conn.cursor()
    print("Connexion réussie à MariaDB !")
except mariadb.Error as e:
    print(f"Erreur de connexion : {e}")
    exit(1)
```

```python
#  Affichage de la structure de la table
print("\n=== Structure de la table 'palu_combat_attribute' ===")
try:
    cursor.execute("DESCRIBE palu_combat_attribute")
    structure = cursor.fetchall()
    for column in structure:
        print(f"Colonne: {column[0]:<25} Type: {column[1]}")
except mariadb.Error as e:
    print(f"Erreur récupération structure : {e}")

#  Affichage des 3 premières lignes
print("\n=== 3 premières lignes de la table ===")
try:
    cursor.execute("SELECT * FROM palu_combat_attribute LIMIT 3")
    rows = cursor.fetchall()

    # Récupération des noms de colonnes
    cursor.execute("SELECT column_name FROM information_schema.columns WHERE tab
    columns = [col[0] for col in cursor.fetchall()]

    # Affichage avec pandas
    import pandas as pd
    df = pd.DataFrame(rows, columns=columns)
    print(df)
except mariadb.Error as e:
    print(f"Erreur récupération données : {e}")

# Fermeture de la connexion
cursor.close()
conn.close()
print("\nConnexion fermée.")
```

```
Connexion réussie à MariaDB !

=== Structure de la table 'palu_combat_attribute' ===
Colonne: id                          Type: int(11)
Colonne: chinese_name                Type: varchar(255)
Colonne: name                        Type: varchar(255)
Colonne: code_name                   Type: varchar(255)
Colonne: override_name_text_id       Type: varchar(255)
Colonne: name_prefix_id              Type: varchar(255)
Colonne: override_partner_skill_text_id Type: varchar(255)
Colonne: is_pal                      Type: tinyint(1)
Colonne: tribe                       Type: varchar(255)
Colonne: bp_class                    Type: varchar(255)
Colonne: variant                     Type: varchar(255)
Colonne: volume_size                 Type: varchar(255)
Colonne: rarity                      Type: int(11)
Colonne: element_1                   Type: varchar(255)
Colonne: element_2                   Type: varchar(255)
Colonne: genus_category              Type: varchar(255)
Colonne: organization                Type: varchar(255)
Colonne: weapon                      Type: varchar(255)
Colonne: weapon_equip                Type: varchar(255)
Colonne: nocturnal                   Type: tinyint(1)
Colonne: total_4d                    Type: int(11)
Colonne: HP                          Type: int(11)
Colonne: melee_attack                Type: int(11)
Colonne: remote_attack               Type: int(11)
Colonne: defense                     Type: int(11)
Colonne: support                     Type: int(11)
Colonne: speed_of_work               Type: int(11)
Colonne: level1_min                  Type: int(11)
Colonne: level1_max                  Type: int(11)
Colonne: level20_min                 Type: int(11)
Colonne: level20_max                 Type: int(11)
Colonne: level50_min                 Type: int(11)
Colonne: level50_max                 Type: int(11)
Colonne: air_response                Type: varchar(255)
Colonne: ai_sight_response           Type: int(11)
Colonne: endurance                   Type: int(11)
Colonne: slow_walking_speed          Type: int(11)
Colonne: walking_speed               Type: int(11)
Colonne: running_speed               Type: int(11)
Colonne: riding_sprint_speed         Type: int(11)
Colonne: damage_multiplier           Type: float
Colonne: catch_rate                  Type: int(11)
Colonne: experience_multiplier       Type: float
Colonne: price                       Type: int(11)
Colonne: must_bring_entry_1          Type: varchar(255)
Colonne: must_bring_entry_2          Type: varchar(255)
Colonne: numerical_description       Type: varchar(255)
Colonne: skill_description           Type: text
Colonne: lv1_1                       Type: int(11)
Colonne: lv2_1                       Type: int(11)
Colonne: lv3_1                       Type: int(11)
Colonne: lv4_1                       Type: int(11)
Colonne: lv5_1                       Type: int(11)

=== 3 premières lignes de la table ===
   id  chinese_name     name   code_name override_name_text_id name_prefix_id  \
0   1    mianyouyou  lamball   sheepball                  None            None
```

```
1   2   pipichicken  chikipi  chickenpal                None          None
2   3  greenleafrat  lifmunk   carbunclo                None          None

  override_partner_skill_text_id  is_pal       tribe    bp_class  ... price  \
0                           None       0   SheepBall   SheepBall  ...  1000
1                           None       0  ChickenPal  ChickenPal  ...  1000
2                           None       0   Carbunclo   Carbunclo  ...  1010

  must_bring_entry_1  must_bring_entry_2 numerical_description  \
0               None                None          Ranch skills
1               None                None          Ranch skills
2               None                None           Skill power

                                skill_description lv1_1 lv2_1 lv3_1 lv4_1  \
0  When activated. it will transform into a shiel...     0     0     0     0
1  Assign it to Palu Ranch and it will have a cha...     0     0     0     0
2  After being activated. it will sit on the play...    10    11    13    16

   lv5_1
0      0
1      0
2     20

[3 rows x 53 columns]

Connexion fermée.
```

In [153…

```python
#  Affichage de la structure et des données de palu_refresh_level

# Reconnexion à la base de données
try:
    conn = mariadb.connect(**config)
    cursor = conn.cursor()
    print("Connexion réussie à MariaDB !")
except mariadb.Error as e:
    print(f"Erreur de connexion : {e}")
    exit(1)

# Affichage de la structure de la table
print("\n=== Structure de la table 'palu_refresh_level' ===")
try:
    cursor.execute("DESCRIBE palu_refresh_level")
    structure = cursor.fetchall()
    for column in structure:
        print(f"Colonne: {column[0]:<20} Type: {column[1]}")
except mariadb.Error as e:
    print(f"Erreur récupération structure : {e}")

#  Affichage des 3 premières lignes
print("\n=== 3 premières lignes de la table ===")
try:
    cursor.execute("SELECT * FROM palu_refresh_level LIMIT 3")
    rows = cursor.fetchall()

    # Récupération des noms de colonnes
    cursor.execute("SELECT column_name FROM information_schema.columns WHERE tab
    columns = [col[0] for col in cursor.fetchall()]

    # Affichage
    df = pd.DataFrame(rows, columns=columns)
```

```
    print(df)
except mariadb.Error as e:
    print(f"Erreur récupération données : {e}")

# Affichage du nombre total d'entrées
try:
    cursor.execute("SELECT COUNT(*) FROM palu_refresh_level")
    count = cursor.fetchone()[0]
    print(f"\nLa table contient actuellement {count} entrées.")
except mariadb.Error as e:
    print(f"Erreur comptage lignes : {e}")

# Fermeture de la connexion
cursor.close()
conn.close()
print("\nConnexion fermée.")
```

```
Connexion réussie à MariaDB !

=== Structure de la table 'palu_refresh_level' ===
Colonne: id                   Type: int(11)
Colonne: name                 Type: varchar(255)
Colonne: minimum_level        Type: int(11)
Colonne: maximum_level        Type: int(11)
Colonne: fecondity            Type: int(11)
Colonne: pallu_refresh_type   Type: varchar(255)
Colonne: night_only           Type: tinyint(1)
Colonne: refresh_area         Type: varchar(255)

=== 3 premières lignes de la table ===
   id         name  minimum_level  maximum_level  fecondity  \
0   1   mianyouyou              1             14       1470
1   2   naughtycat              1             13       1460
2   3  pipichicken              1             13       1500

  pallu_refresh_type  night_only refresh_area
0             Creeps           0    grassland
1             Creeps           0    grassland
2             Creeps           0    grassland

La table contient actuellement 136 entrées.

Connexion fermée.
```

In [143…

```
# Structure de la table ordinary_boss_comparison et controle des 3 premières lig

# Configuration pour se connecter à la base de données
config = {
    'user': 'root',
    'password': 'cN06+#P34',
    'host': 'localhost',
    'port': 3307,
    'database': 'palworld_database'
}

try:
    # Connexion à la base de données
    conn = mariadb.connect(**config)
    cursor = conn.cursor()
```

```
    # Afficher la structure de la table
    cursor.execute("DESCRIBE ordinary_boss_comparison")
    structure = cursor.fetchall()
    print("Structure de la table 'ordinary_boss_comparison' :")
    for column in structure:
        print(f"Colonne: {column[0]}, Type: {column[1]}")

    # Afficher premières lignes de la table
    cursor.execute("SELECT * FROM ordinary_boss_comparison LIMIT 3")
    rows = cursor.fetchall()
    print("\nTrois premières lignes de la table :")
    for row in rows:
        print(row)

except mariadb.Error as e:
    print(f"Erreur lors de la connexion à MariaDB : {e}")

finally:
    # Fermer la connexion
    if 'conn' in locals():
        cursor.close()
        conn.close()
        print("\nConnexion à la base de données fermée.")
```

```
Structure de la table 'ordinary_boss_comparison' :
Colonne: id, Type: int(11)
Colonne: name, Type: varchar(255)
Colonne: HP, Type: int(11)
Colonne: Remote_attack, Type: int(11)
Colonne: Riding_speed_BOSS_is_100_higher, Type: int(11)

Trois premières lignes de la table :
(1, 'heterogeneousgriffinboss', None, None, 1300)
(2, 'heterogeneousgriffin', None, None, 1200)
(3, 'yepandaboss', None, None, 800)

Connexion à la base de données fermée.
```

In [142…

```
# Affichage de la structure et des données de tower_bosses

# Reconnexion à la base de données
try:
    conn = mariadb.connect(**config)
    cursor = conn.cursor()
    print("Connexion réussie à MariaDB !")
except mariadb.Error as e:
    print(f"Erreur de connexion : {e}")
    exit(1)

# Affichage de la structure de la table
print("\n=== Structure de la table 'tower_bosses' ===")
try:
    cursor.execute("DESCRIBE tower_bosses")
    structure = cursor.fetchall()
    for column in structure:
        print(f"Colonne: {column[0]:<20} Type: {column[1]}")
except mariadb.Error as e:
    print(f"Erreur récupération structure : {e}")

# Affichage des 3 premières lignes
```

```
print("\n=== 3 premières lignes de la table ===")
try:
    cursor.execute("SELECT * FROM tower_bosses LIMIT 3")
    rows = cursor.fetchall()

    # Récupération des noms de colonnes
    cursor.execute("SELECT column_name FROM information_schema.columns WHERE tab
    columns = [col[0] for col in cursor.fetchall()]

    # Affichage
    import pandas as pd
    df = pd.DataFrame(rows, columns=columns)
    print(df)
except mariadb.Error as e:
    print(f"Erreur récupération données : {e}")

# Affichage du nombre total d'entrées
try:
    cursor.execute("SELECT COUNT(*) FROM tower_bosses")
    count = cursor.fetchone()[0]
    print(f"\nLa table contient actuellement {count} entrées.")
except mariadb.Error as e:
    print(f"Erreur comptage lignes : {e}")

# Fermeture de la connexion
cursor.close()
conn.close()
print("\nConnexion fermée.")
```

```
Connexion réussie à MariaDB !

=== Structure de la table 'tower_bosses' ===
Colonne: id                   Type: int(11)
Colonne: name                 Type: varchar(255)
Colonne: HP                   Type: int(11)
Colonne: melee_attack         Type: int(11)
Colonne: remote_attack        Type: int(11)
Colonne: defense              Type: int(11)
Colonne: support              Type: int(11)
Colonne: experience_ratio     Type: int(11)
Colonne: slow_walking_speed   Type: int(11)
Colonne: walking_speed        Type: int(11)
Colonne: running_speed        Type: int(11)
Colonne: riding_speed         Type: int(11)
Colonne: handling_speed       Type: int(11)
Colonne: ignore_the_bluntness Type: tinyint(1)
Colonne: ignore_displacement  Type: tinyint(1)
Colonne: biological_grade     Type: int(11)
Colonne: endurance            Type: int(11)
Colonne: fecundity            Type: int(11)

=== 3 premières lignes de la table ===
   id                        name    HP  melee_attack  remote_attack  \
0   1  victor&heterogeneousgriffin  8000           130            200
1   2         heterogeneousgriffin   120           130            120
2   3            zoe&lightningbear  6000           100            100

   defense  support  experience_ratio  slow_walking_speed  walking_speed  \
0      220       90                30                  80             80
1      140       90                 1                  80             80
2      100      100                30                  80            140

   running_speed  riding_speed  handling_speed  ignore_the_bluntness  \
0            850          1100             465                     1
1            850          1200             465                     0
2            470           650             287                     1

   ignore_displacement  biological_grade  endurance  fecundity
0                    1                 9        100       9999
1                    0                 5        250         60
2                    1                 9        100       9999

La table contient actuellement 10 entrées.

Connexion fermée.
```

In [141…

```python
# Affichage de la structure et des données de monsters

# Reconnexion à la base de données
try:
    conn = mariadb.connect(**config)
    cursor = conn.cursor()
    print("Connexion réussie à MariaDB !")
except mariadb.Error as e:
    print(f"Erreur de connexion : {e}")
    exit(1)

#Affichage de la structure de la table
print("\n=== Structure de la table 'monsters' ===")
```

```
try:
    cursor.execute("DESCRIBE monsters")
    structure = cursor.fetchall()
    for column in structure:
        print(f"Colonne: {column[0]:<25} Type: {column[1]}")
except mariadb.Error as e:
    print(f"Erreur récupération structure : {e}")

#Affichage des 3 premières lignes
print("\n=== 3 premières lignes de la table ===")
try:
    cursor.execute("SELECT * FROM monsters LIMIT 3")
    rows = cursor.fetchall()

    cursor.execute("SELECT column_name FROM information_schema.columns WHERE tab
    columns = [col[0] for col in cursor.fetchall()]

    import pandas as pd
    df = pd.DataFrame(rows, columns=columns)
    print(df)
except mariadb.Error as e:
    print(f"Erreur récupération données : {e}")

# Affichage du nombre total d'entrées
try:
    cursor.execute("SELECT COUNT(*) FROM monsters")
    count = cursor.fetchone()[0]
    print(f"\nLa table contient actuellement {count} entrées.")
except mariadb.Error as e:
    print(f"Erreur comptage lignes : {e}")

# Fermeture de la connexion
cursor.close()
conn.close()
print("\nConnexion fermée.")
```

```
Connexion réussie à MariaDB !

=== Structure de la table 'monsters' ===
Colonne: id                          Type: int(11)
Colonne: OverrideNameTextID          Type: varchar(255)
Colonne: OverridePartnerSkillTextID Type: varchar(255)
Colonne: IsPal                       Type: int(11)
Colonne: Tribe                       Type: varchar(255)
Colonne: BPClass                     Type: varchar(255)
Colonne: Pictorial_ID                Type: int(11)
Colonne: ZukanIndexSuffix            Type: float
Colonne: Size                        Type: varchar(50)
Colonne: rarity                      Type: int(11)
Colonne: Element_1                   Type: varchar(50)
Colonne: Element_2                   Type: varchar(50)
Colonne: GenusCategory               Type: varchar(50)
Colonne: Organization                Type: float
Colonne: weapon                      Type: float
Colonne: WeaponEquip                 Type: int(11)
Colonne: HP                          Type: int(11)
Colonne: melee_attack                Type: int(11)
Colonne: Remote_attack               Type: int(11)
Colonne: defense                     Type: int(11)
Colonne: support                     Type: int(11)
Colonne: CraftSpeed                  Type: int(11)
Colonne: damageMultiplier            Type: varchar(50)
Colonne: CaptureProbability          Type: varchar(50)
Colonne: ExperienceMultiplier        Type: int(11)
Colonne: price                       Type: int(11)
Colonne: AIRResponse                 Type: varchar(50)
Colonne: AISightResponse             Type: float
Colonne: slow_walking_speed          Type: int(11)
Colonne: walking_speed               Type: int(11)
Colonne: running_speed               Type: int(11)
Colonne: Riding_sprint_speed         Type: int(11)
Colonne: Handling_speed              Type: int(11)
Colonne: IsBoss                      Type: int(11)
Colonne: IsTowerBoss                 Type: int(11)
Colonne: BattleBGM                   Type: varchar(255)
Colonne: IgnoreLeanBack              Type: int(11)
Colonne: IgnoreBlowAway              Type: int(11)
Colonne: MaxFullStomach              Type: int(11)
Colonne: FullStomachDecreaseRate     Type: int(11)
Colonne: FoodAmount                  Type: int(11)
Colonne: ViewingDistance             Type: int(11)
Colonne: ViewingAngle                Type: int(11)
Colonne: HearingRate                 Type: int(11)
Colonne: NooseTrap                   Type: int(11)
Colonne: Nocturnal                   Type: int(11)
Colonne: BiologicalGrade             Type: int(11)
Colonne: Predator                    Type: int(11)
Colonne: Edible                      Type: int(11)
Colonne: endurance                   Type: int(11)
Colonne: Male_probability            Type: int(11)
Colonne: fecundity                   Type: int(11)
Colonne: Breathing_fire              Type: int(11)
Colonne: watering                    Type: int(11)
Colonne: planting                    Type: int(11)
Colonne: generate_electricity        Type: int(11)
Colonne: manual                      Type: int(11)
```

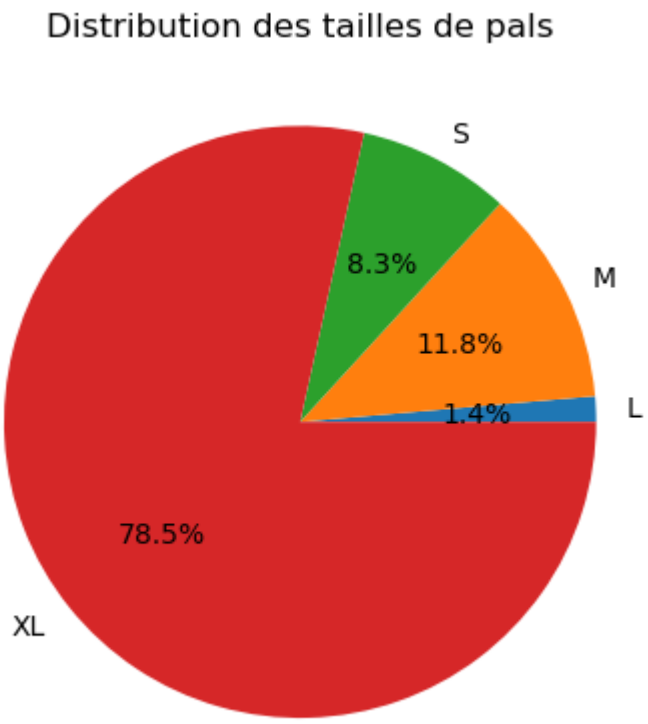
```
Colonne: collection               Type: int(11)
Colonne: logging                  Type: int(11)
Colonne: Mining                   Type: int(11)
Colonne: pharmaceutical           Type: int(11)
Colonne: cool_down                Type: int(11)
Colonne: carry                    Type: int(11)
Colonne: pasture                  Type: int(11)
Colonne: Passive_skill_1          Type: varchar(255)
Colonne: Passive_skill_2          Type: varchar(255)
Colonne: Passive_skill_3          Type: float
Colonne: Passive_skill_4          Type: float

=== 3 premières lignes de la table ===
   id OverrideNameTextID       OverridePartnerSkillTextID  IsPal  \
0   1  sakurasauruswater  PARTNERSKILL_SakuraSaurus_Water      1
1   2      amaterasuwolf       PARTNERSKILL_AmaterasuWolf      1
2   3          robinhood           PARTNERSKILL_RobinHood      1

               Tribe                  BPClass  Pictorial_ID ZukanIndexSuffix  \
0  SakuraSaurus_Water  BOSS_SakuraSaurus_Water            -1             None
1       AmaterasuWolf       BOSS_AmaterasuWolf            -1             None
2           RobinHood           BOSS_RobinHood            -1             None

  Size  rarity  ... logging Mining pharmaceutical cool_down carry  pasture  \
0   XL       8  ...       0      0              0         0     0        0
1   XL       6  ...       0      0              0         0     0        0
2   XL       0  ...       1      0              1         0     1        0

   Passive_skill_1  Passive_skill_2  Passive_skill_3  Passive_skill_4
0             None             None             None             None
1             None             None             None             None
2             None             None             None             None

[3 rows x 68 columns]

La table contient actuellement 144 entrées.

Connexion fermée.
```

Identification des leviers de performance via l'exploration des données de Palworld

In [ ]:
```
### A. Quelle est la distribution de la taille des Pals ?
```

In [173…
```
# Config base de données
config = {
    'user': 'root',
    'password': 'cN06+#P34',
    'host': 'localhost',
    'port': 3307,
    'database': 'palworld_database'
}

#  Connexion via SQLAlchemy engine
engine = create_engine(
    f"mariadb+mariadbconnector://{config['user']}:{config['password']}@{config['
)
```

```
#  Requête SQL
query = """
 select size, count(size) as total from monsters group by size;
"""
# Lecture des données via pandas
df = pd.read_sql(query, engine)

plt.pie(df["total"], labels=df["size"], autopct="%1.1f%%")

# Ajout de titre
plt.title("Distribution des tailles de pals")

# Affichage de la figure
plt.show()

engine.dispose()
```



B Distribution des Pals selon la catégorie

Objectif :

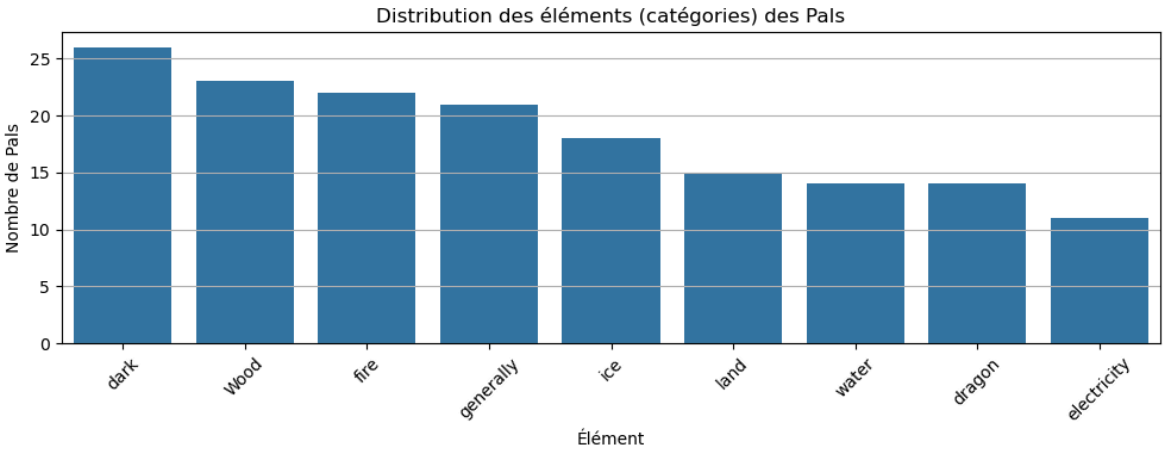
Visualiser la répartition de tous les types élémentaires utilisés par les Pals, qu'ils soient en élément principal (element_1) ou secondaire (element_2).

In [176…

```
# Configuration de la base
config = {
    'user': 'root',
    'password': 'cN06+#P34',
    'host': 'localhost',
    'port': 3307,
    'database': 'palworld_database'}

# === Connexion à la base ===
```

```
engine = create_engine(
    f"mariadb+mariadbconnector://{config['user']}:{config['password']}@{config['
)

#  Requête SQL :
query = """
SELECT element_1, element_2
FROM palu_combat_attribute
WHERE (element_1 IS NOT NULL AND element_1 != '')
   OR (element_2 IS NOT NULL AND element_2 != '');
"""

df_elements = pd.read_sql(query, engine)

# Fusion des deux colonnes pour analyser toutes les occurrences
all_elements = pd.concat([df_elements['element_1'], df_elements['element_2']])
all_elements = all_elements.dropna().str.strip()

#  Comptage des fréquences
element_counts = all_elements.value_counts()

# === Affichage texte ===
print("Répartition des éléments des Pals :")
print(element_counts)

#  Visualisation
plt.figure(figsize=(10, 4))
sns.barplot(x=element_counts.index, y=element_counts.values)
plt.title("Distribution des éléments (catégories) des Pals")
plt.xlabel("Élément")
plt.ylabel("Nombre de Pals")
plt.xticks(rotation=45)
plt.grid(axis='y')
plt.tight_layout()
plt.show()

# Déconnexion
engine.dispose()
```

```
Répartition des éléments des Pals :
dark           26
Wood           23
fire           22
generally      21
ice            18
land           15
water          14
dragon         14
electricity    11
Name: count, dtype: int64
```



analyse:

Cette visualisation met en évidence la répartition des éléments principaux (ou types) associés aux Pals du jeu. Ces éléments déterminent généralement les forces, faiblesses et affinités de chaque Pal dans les combats ou les tâches spécifiques.

Parmi les neuf éléments recensés, on observe que :

L'élément dark (obscurité) est le plus représenté, avec plus de 25 Pals. Cela peut refléter une orientation stratégique vers des capacités puissantes mais potentiellement coûteuses ou rares.

Les éléments wood (bois), fire (feu) et generally (neutre ou non spécifié) sont également bien représentés, suggérant leur polyvalence et leur utilité dans diverses situations de jeu.

Les éléments comme ice, land et water sont présents de façon modérée, avec entre 15 et 20 représentants chacun.

Enfin, les éléments dragon et electricity sont les moins fréquents, ce qui suggère une rareté potentielle de ces types ou une spécialisation plus poussée.

C Distribution des points de vie (HP) des Pals

Objectif :

Analyser comment les points de vie (HP) des Pals sont répartis :

-Moyenne, médiane, min, max

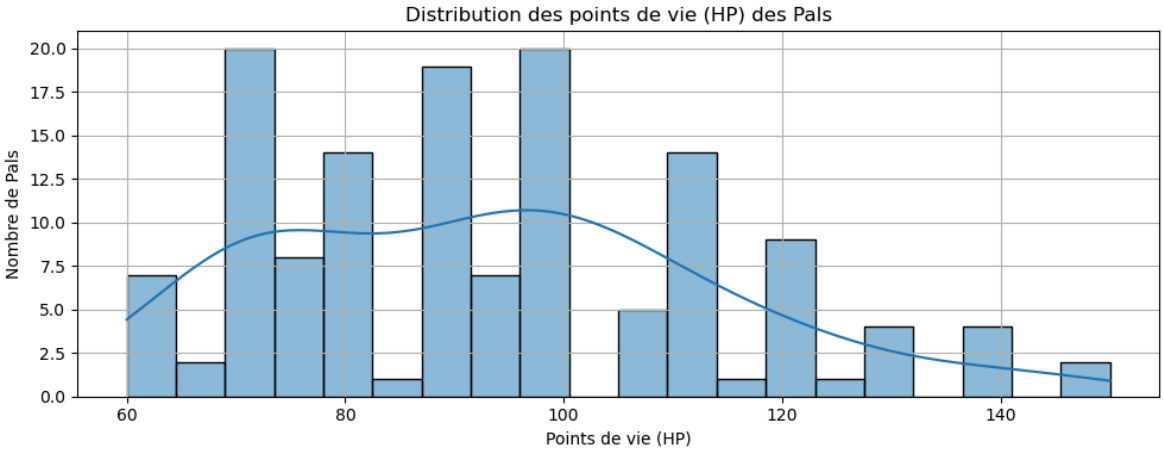
-Visualisation par histogramme

In [175…

```
# Configuration connexion
config = {
    'user': 'root',
    'password': 'cN06+#P34',
    'host': 'localhost',
    'port': 3307,
```

```
    'database': 'palworld_database'
}


# Connexion via SQLAlchemy
engine = create_engine(
    f"mariadb+mariadbconnector://{config['user']}:{config['password']}@{config['
)

# Requête SQL pour HP
query = """
SELECT name, HP
FROM palu_combat_attribute
WHERE HP IS NOT NULL;
"""

df_hp = pd.read_sql(query, engine)

# Nettoyage
df_hp['HP'] = pd.to_numeric(df_hp['HP'], errors='coerce')
df_hp.dropna(subset=['HP'], inplace=True)

# Statistiques
print("Statistiques des HP des Pals :")
print(df_hp['HP'].describe())

# Visualisation
plt.figure(figsize=(10, 4))
sns.histplot(df_hp['HP'], bins=20, kde=True)
plt.title("Distribution des points de vie (HP) des Pals")
plt.xlabel("Points de vie (HP)")
plt.ylabel("Nombre de Pals")
plt.grid(True)
plt.tight_layout()
plt.show()

# Déconnexion propre
engine.dispose()
```

```
Statistiques des HP des Pals :
count    138.000000
mean      93.442029
std       20.855785
min       60.000000
25%       75.000000
50%       90.000000
75%      108.750000
max      150.000000
Name: HP, dtype: float64
```



analyse:

L'histogramme ci-dessus montre la répartition des points de vie (HP) des Pals, un indicateur fondamental de leur capacité de survie en combat. La majorité des Pals se situent dans une plage de HP comprise entre 70 et 110, avec un pic autour de 90 HP, ce qui correspond également à la médiane de l'échantillon.

La distribution est asymétrique légèrement étendue à droite, indiquant que bien que la majorité des Pals aient une endurance modérée, il existe une minorité de Pals très robustes avec des HP supérieurs à 120.

La concentration autour de 80 à 100 HP suggère que les développeurs ont cherché à standardiser la résistance de base pour la majorité des Pals.

Les Pals dotés de plus de 120 HP peuvent être considérés comme les plus résistants dans une équipe de combat, capables d'encaisser de nombreux dégâts.

D Distribution de la rareté des Pals

Objectif :

Étudier la répartition de la rareté (rarity) des Pals dans le jeu. Cela permet d'observer :

Combien de Pals sont très rares, communs ou intermédiaires.

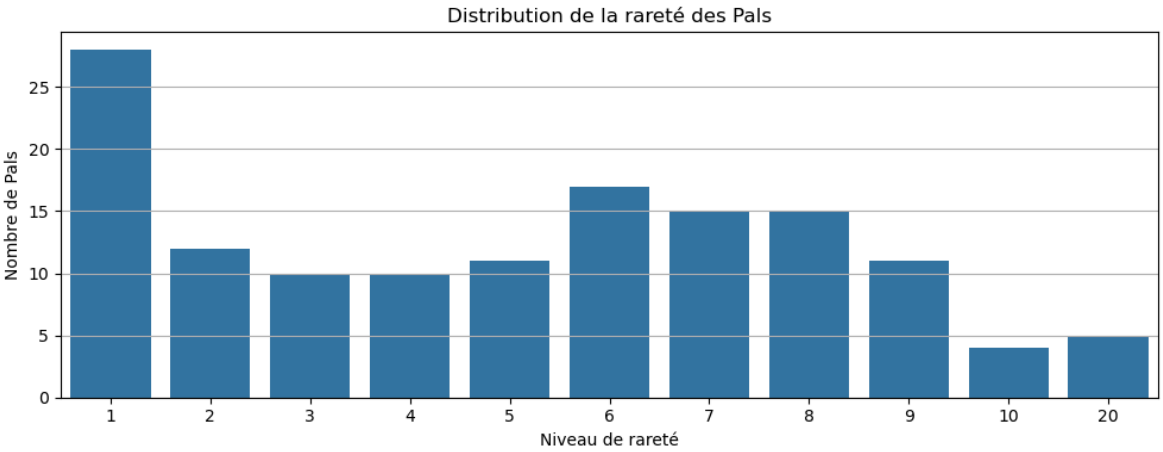
La structure de difficulté ou d'équilibre du jeu.

In [172…

```
# Configuration de la base de données
config = {
    'user': 'root',
    'password': 'cN06+#P34',
    'host': 'localhost',
    'port': 3307,
    'database': 'palworld_database'
}

#  Connexion SQLAlchemy
engine = create_engine(
    f"mariadb+mariadbconnector://{config['user']}:{config['password']}@{config['
```

```
)

# Requête SQL rareté
query = """
SELECT name, rarity
FROM palu_combat_attribute
WHERE rarity IS NOT NULL;
"""

df_rarity = pd.read_sql(query, engine)


#  Statistiques descriptives
print("Statistiques sur la rareté des Pals :")
print(df_rarity['rarity'].describe())

# === Distribution nombre de Pals /  rareté
rarity_counts = df_rarity['rarity'].value_counts().sort_index()

# Affichage
print("\nNombre de Pals par niveau de rareté :")
print(rarity_counts)

# Visualisation
plt.figure(figsize=(10, 4))
sns.barplot(x=rarity_counts.index.astype(str), y=rarity_counts.values)
plt.title("Distribution de la rareté des Pals")
plt.xlabel("Niveau de rareté")
plt.ylabel("Nombre de Pals")
plt.grid(axis='y')
plt.tight_layout()
plt.show()

# Déconnexion
engine.dispose()
```

```
Statistiques sur la rareté des Pals :
count    138.000000
mean       5.384058
std        4.006058
min        1.000000
25%        2.000000
50%        5.000000
75%        7.750000
max       20.000000
Name: rarity, dtype: float64

Nombre de Pals par niveau de rareté :
rarity
1     28
2     12
3     10
4     10
5     11
6     17
7     15
8     15
9     11
10     4
20     5
Name: count, dtype: int64
```



analyse:

Cette visualisation montre comment les Pals sont répartis selon leur niveau de rareté, un indicateur central qui reflète à la fois leur disponibilité et souvent leur puissance ou utilité stratégique dans le jeu.

Le niveau de rareté 1 est le plus fréquent, avec 28 Pals, ce qui représente la base commune du bestiaire.

La majorité des Pals ont une rareté comprise entre 3 et 9, avec une concentration autour de 6 à 8, ce qui correspond à une rareté modérée à élevée.

On note un petit groupe de Pals exceptionnels aux raretés 10 et 20, qui ne comptent que quelques représentants. Ces derniers sont probablement des Pals très puissants ou difficiles à obtenir.

In [ ]:
```
### E  Distribution de la consommation alimentaire des Pals
```

Objectif :

Analyser la répartition de la quantité de nourriture consommée par chaque Pal. Cela permet d'identifier :

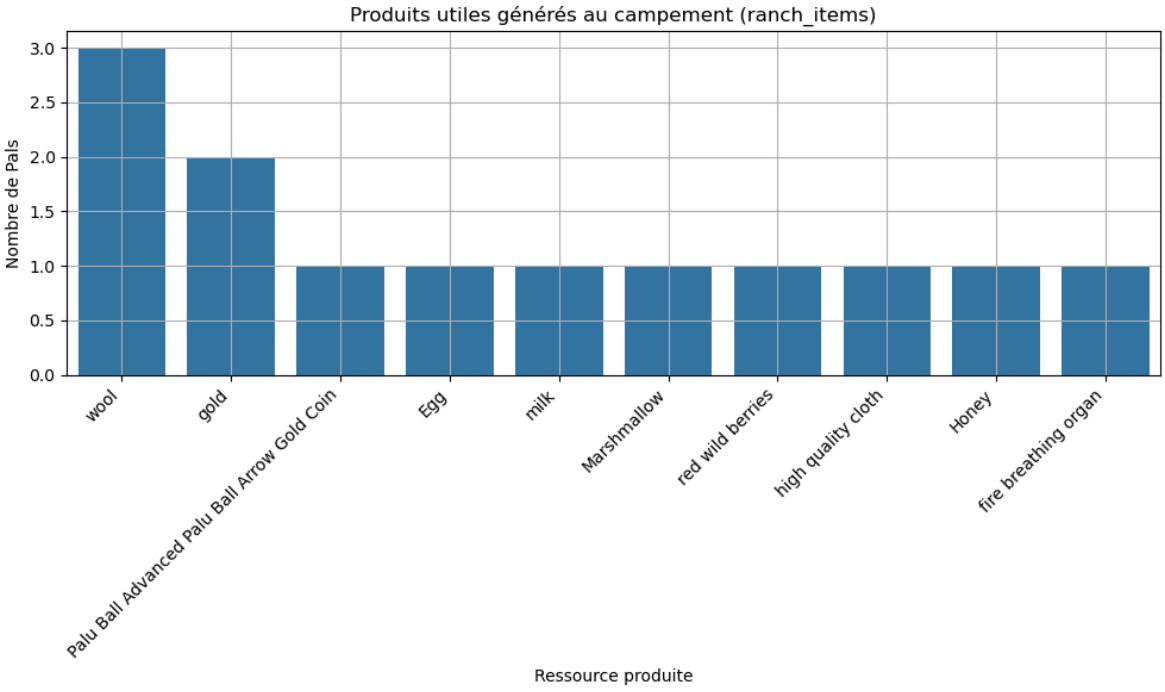
Quels Pals consomment beaucoup (ressources critiques),

Les plus économes, utiles en survie/gestion de base.

In [170…
```
# Récupération des données depuis la table job_skills
query = """
SELECT english_name, ranch_items
FROM job_skills
WHERE ranch_items IS NOT NULL AND ranch_items != '';
"""

df_ranch = pd.read_sql(query, engine)

# Nettoyage de base
df_ranch['ranch_items'] = df_ranch['ranch_items'].str.strip()

# Affichage des Pals et ressources
print("Pals qui produisent des objets utiles au ranch :")
display(df_ranch)

# Statistiques  fréquence  ressources
resource_counts = df_ranch['ranch_items'].value_counts().reset_index()
resource_counts.columns = ['Ressource produite', 'Nombre de Pals']

# Affichage
plt.figure(figsize=(10, 6))
sns.barplot(data=resource_counts, x='Ressource produite', y='Nombre de Pals')
plt.title("Produits utiles générés au campement (ranch_items)")
plt.xticks(rotation=45, ha='right')
plt.ylabel("Nombre de Pals")
plt.grid(True)
plt.tight_layout()
plt.show()
```

```
Pals qui produisent des objets utiles au ranch :
```

	english_name	ranch_items
0	chikipi	Egg
1	vixy	Palu Ball Advanced Palu Ball Arrow Gold Coin
2	cremis	wool
3	mau	gold
4	maucryst	gold
5	mozzarina	milk
6	woolipop	Marshmallow
7	caprity	red wild berries
8	melpaca	wool
9	sibelyx	high quality cloth
10	lamball	wool
11	beegarde	Honey
12	flambelle	fire breathing organ



Analyse:

Ce graphique présente les ressources que les Pals peuvent produire lorsqu'ils sont affectés à un ranch, via la variable ranch_items. Ces produits jouent un rôle crucial dans la gestion du campement, le craft, ou encore l'économie du jeu. Lecture du graphique : Le produit le plus courant est la laine (wool), générée par 3 Pals, confirmant son rôle de ressource de base.

Suivent l'or (gold), les œufs (egg) et le lait (milk), chacun produit par 1 à 2 Pals, avec une utilité probable dans les recettes, le commerce, ou la reproduction.

D’autres produits plus spécifiques apparaissent comme :

high quality cloth (tissu de qualité)

honey (miel)

fire breathing organ (organe cracheur de feu)

red wild berries ou marshmallow (nourriture/ressources avancées)

Interprétation stratégique :
Cette répartition met en lumière une diversité utile mais concentrée : la majorité des produits sont générés par un très petit nombre de Pals, ce qui renforce leur valeur stratégique.

Certains produits rares peuvent être cruciaux dans les étapes avancées du jeu (tissu de qualité, or, organes).

Les produits comme la laine ou le lait peuvent servir de commencement pour maintenir l’économie du campement.

G la distribution de la puissance de combat parmi les Pals

Objectif :

Cette analyse va permettre d'identifier les 10 Pals les plus puissants en se basant sur leurs attributs de combat.

Pour cela nous devons calculer une puissance globale pour chaque Pal basée sur :

melee_attack

remote_attack

defense

support

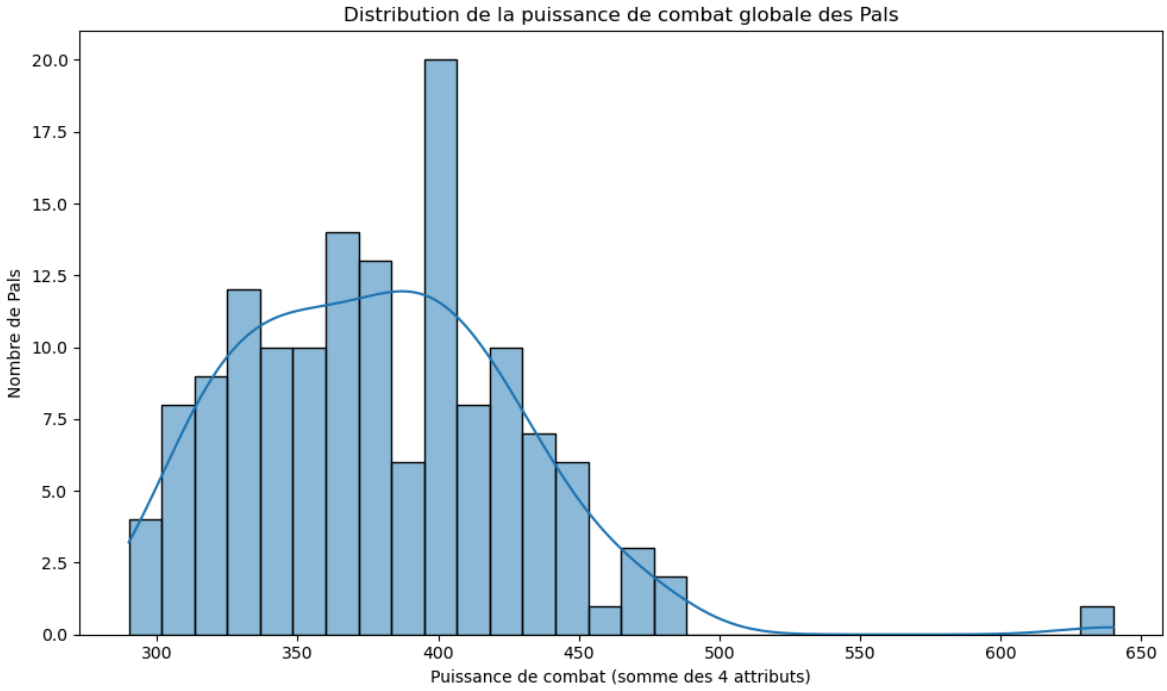
Visualiser la distribution générale de cette puissance

Extraire le Top 10 des Pals les plus puissant

In [169…

```
# 1. Requête pour  attributs de combat
query = """
SELECT OverrideNameTextID, melee_attack, remote_attack, defense, support
FROM monsters
WHERE melee_attack IS NOT NULL AND remote_attack IS NOT NULL AND defense IS NOT
"""
df_combat = pd.read_sql(query, engine)

# Calcul de la puissance globale
df_combat['combat_power'] = df_combat['melee_attack'] + df_combat['remote_attack
```

```
# Distribution visu
plt.figure(figsize=(10, 6))
sns.histplot(df_combat['combat_power'], bins=30, kde=True)
plt.title("Distribution de la puissance de combat globale des Pals")
plt.xlabel("Puissance de combat (somme des 4 attributs)")
plt.ylabel("Nombre de Pals")
plt.tight_layout()
plt.show()

# Top 10  Pals + puissants
top_10_pals = df_combat.sort_values(by='combat_power', ascending=False).head(10)
print("🔥 Top 10 des Pals les plus puissants :")
display(top_10_pals)
```



🔥 Top 10 des Pals les plus puissants :

	OverrideNameTextID	melee_attack	remote_attack	defense	support	combat_power
135	snowboss	130	200	220	90	640
45	blackgriffon	130	120	140	90	480
91	umihebifire	150	130	100	100	480
116	saintcentaur	110	120	145	100	475
126	umihebi	150	120	100	100	470
123	blackcentaur	100	145	120	100	465
115	anubis	130	130	100	100	460
30	whitetiger	140	100	110	100	450
118	jetdragon	100	140	110	100	450
129	volcanoboss	100	140	110	100	450

Analyse:

Cette analyse repose sur la somme de quatre attributs clés : melee_attack, remote_attack, defense, support pour construire un indicateur agrégé de puissance de combat globale.

La majorité des Pals affichent une puissance totale entre 300 et 450, avec une concentration autour de 400 points. Cela montre un équilibrage global autour d'un profil "standard", probablement destiné à la majorité des situations de jeu.

Cependant, la courbe présente une queue asymétrique vers la droite, traduisant l'existence de Pals exceptionnellement puissants, rares mais surpuissant.

Interprétation stratégique :

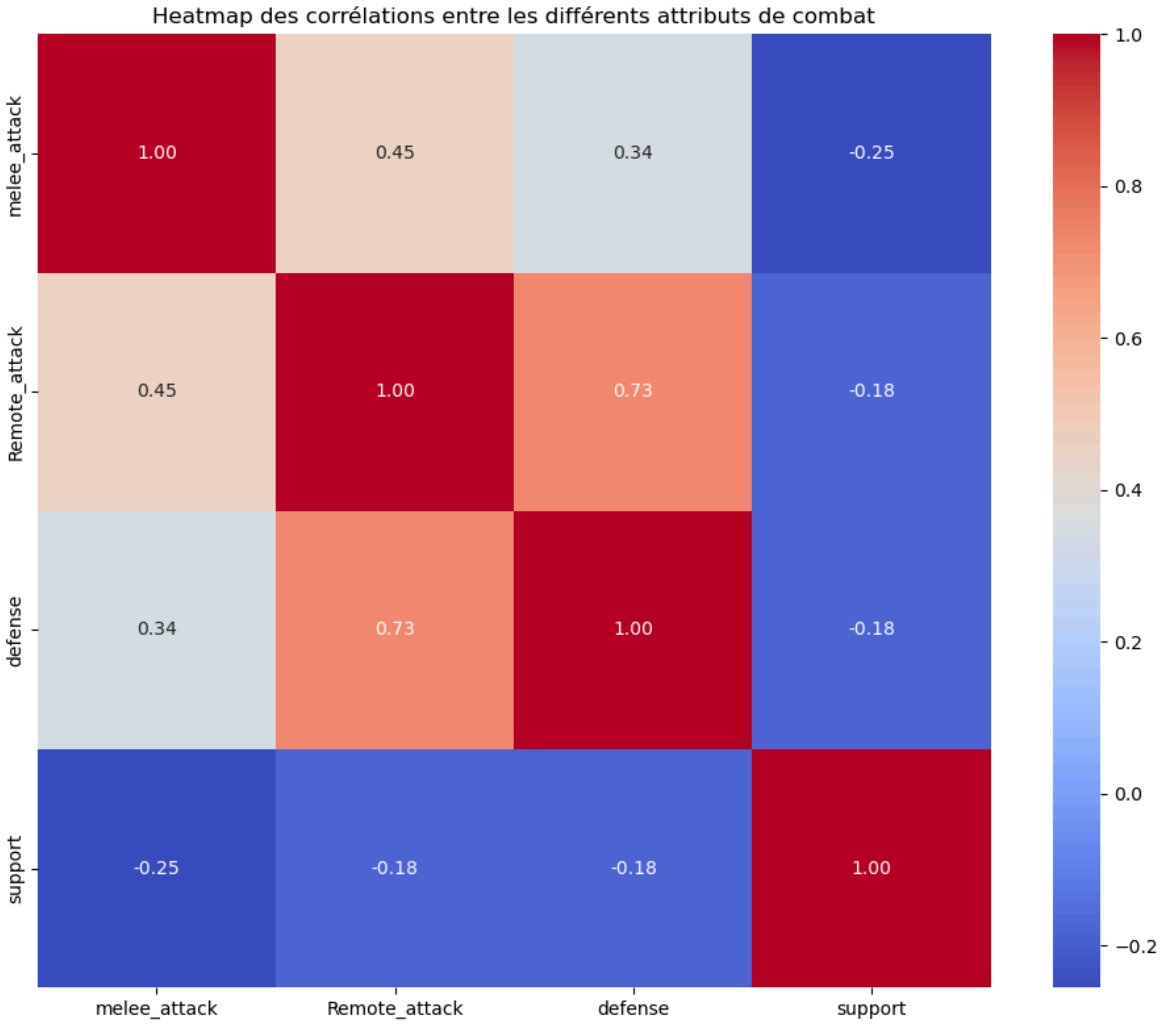
Snowboss se détache clairement avec une puissance de 640, bien au-dessus de la moyenne. Il pourrait représenter un boss ou un Pal légendaire.

La valeur stratégique de ces Pals justifie probablement une rareté plus élevée et une consommation de ressources plus importante.

Il est intéressant de noter que la puissance n'est pas uniquement corrélée à l'attaque, mais aussi à la défense et au support, ce qui encourage des compositions d'équipes variées.

In [8]:
```
### H. Quelles sont les corrélations entre les différents attributs de combat ?
```

In [9]:
```
# === Config base de données ===
config = {
    'user': 'root',
    'password': 'cN06+#P34',
    'host': 'localhost',
    'port': 3307,
    'database': 'palworld_database'
}

# === Connexion via SQLAlchemy engine ===
engine = create_engine(
    f"mariadb+mariadbconnector://{config['user']}:{config['password']}@{config['
)

# Requête SQL
 select melee_attack, Remote_attack, defense,support from monsters;
"""

# === Lecture des données via pandas ===
df_pal_comb_attr = pd.read_sql(query, engine)

correlation_combat_attribut = df_pal_comb_attr.corr(numeric_only=True)

plt.figure(figsize=(10, 8))
sns.heatmap(correlation_combat_attribut, annot=True, cmap='coolwarm', fmt=".2f",
plt.title("Heatmap des corrélations entre les différents attributs de combat")
plt.tight_layout()
plt.show()

engine.dispose()
```



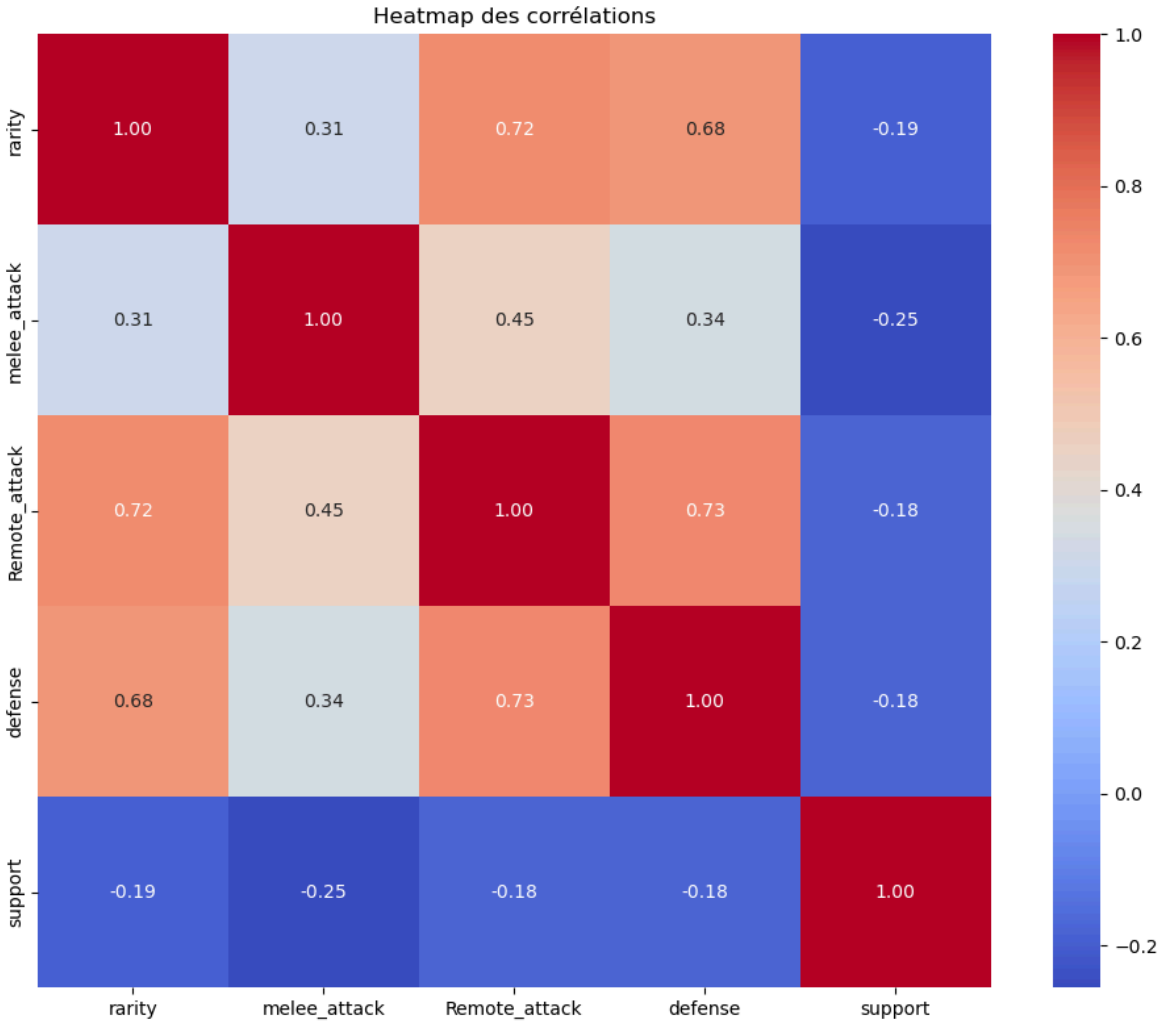
Nous pouvons voir dans ce graphique qu'il existe differentes correlations sur les attributs de combats:

Correlation modérée entre defense et melee_attack Correlation modérée entre remote_attack et melee_attack Correlation forte entre remote_attack et defense

I. Comment la rareté d'un Pal affecte-t-elle les valeurs de ses attributs de base ?

In [10]:
```
#  Config base de données
config = {
    'user': 'root',
    'password': 'cN06+#P34',
    'host': 'localhost',
    'port': 3307,
    'database': 'palworld_database'
}

#  Connexion via SQLAlchemy engine
engine = create_engine(
    f"mariadb+mariadbconnector://{config['user']}:{config['password']}@{config['
)

# Requête SQL
query = """
 select rarity, melee_attack, Remote_attack, defense,support from monsters;
```

```
"""

# Lecture des données via pandas
df_pal_comb_attr = pd.read_sql(query, engine)

correlation_combat_attribut = df_pal_comb_attr.corr(numeric_only=True)

plt.figure(figsize=(10, 8))
sns.heatmap(correlation_combat_attribut, annot=True, cmap='coolwarm', fmt=".2f",
plt.title("Heatmap des corrélations")
plt.tight_layout()
plt.show()

engine.dispose()
```



J Rareté moyenne des Pals ayant la puissance d'attaque la plus élevée

Objectif :

Identifier les Pals ayant les meilleures attaques (melee_attack, remote_attack)

Calculer la rareté moyenne parmi ces Pals

In [165…

```
# Extraire attaque et rareté
query = """
```

```
SELECT OverrideNameTextID, melee_attack, remote_attack, rarity
FROM monsters
WHERE melee_attack IS NOT NULL AND remote_attack IS NOT NULL AND rarity IS NOT N
"""
df = pd.read_sql(query, engine)

#  colonne de puissance offensive
df['attack_power'] = df['melee_attack'] + df['remote_attack']

#   top 10
top_attackers = df.sort_values(by='attack_power', ascending=False).head(10)

# Calcul de la rareté moyenne
mean_rarity = top_attackers['rarity'].mean()

# Affichage
print(" Rareté moyenne des 10 Pals avec la plus forte attaque :", round(mean_rar
display(top_attackers[['OverrideNameTextID', 'melee_attack', 'remote_attack', 'r
```

```
Rareté moyenne des 10 Pals avec la plus forte attaque : 9.2
```

	OverrideNameTextID	melee_attack	remote_attack	rarity	attack_power
135	snowboss	130	200	10	330
91	umihebifire	150	130	9	280
124	kingbahamut	150	125	9	275
126	umihebi	150	120	8	270
115	anubis	130	130	10	260
47	baphometdark	150	105	5	255
45	blackgriffon	130	120	10	250
66	baphomet	150	100	4	250
123	blackcentaur	100	145	20	245
30	whitetiger	140	100	7	240

In [ ]:

```
### Analyse:
L'analyse ci-dessus présente les 10 Pals affichant la plus forte puissance offen
Les Pals les plus puissants en attaque sont-ils aussi les plus rares ?
Résultats :
La rareté moyenne de ces 10 Pals est de 9.2, ce qui est significativement supéri

Plusieurs Pals offensifs ont une rareté de 10 ou plus : snowboss, umihebi, anubi

Quelques exceptions existent :

baphometdark (rareté 5)

baphomet (rareté 4)

Interprétation stratégique :

Globalement, on observe une corrélation claire entre puissance offensive et rare

Cependant, l'existence de Pals puissants à rareté modérée montre qu'il est possi
```

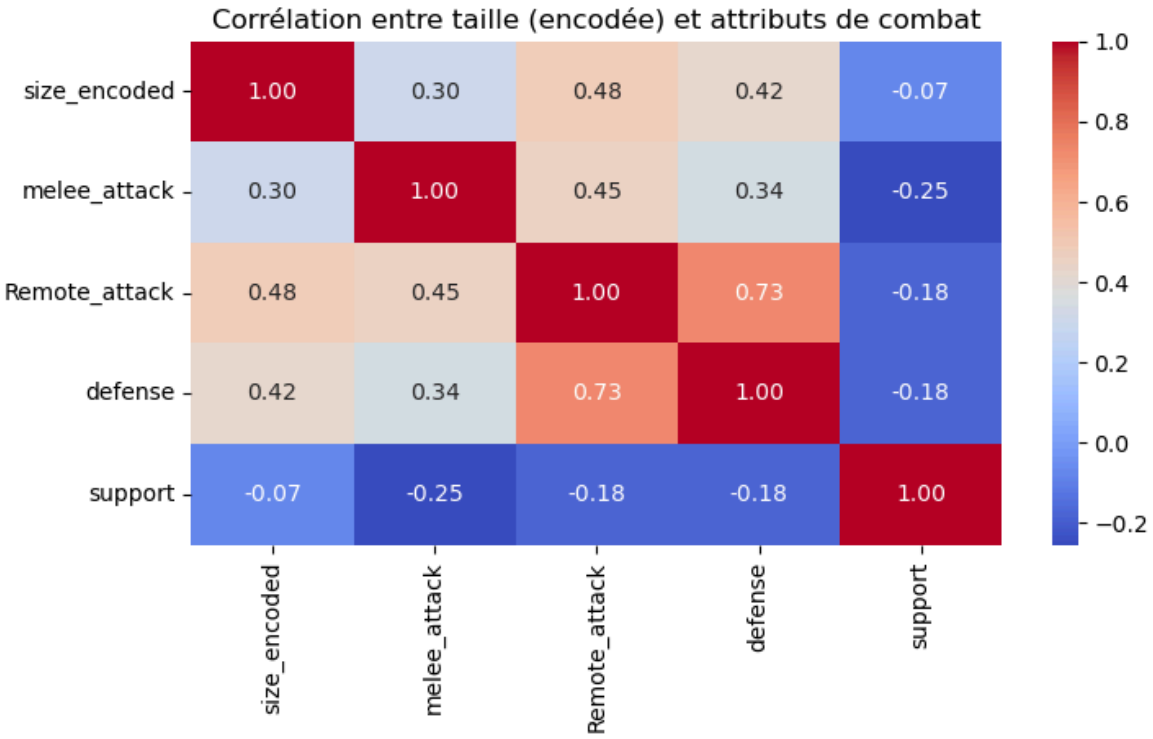
```
Ces exceptions comme baphometdark offrent un excellent rapport attaque/rareté, i
```

K. La taille des Pals affecte-t-elle leur performance au combat ?

In [164…

```
# === Config base de données ===
config = {
    'user': 'root',
    'password': 'cN06+#P34',
    'host': 'localhost',
    'port': 3307,
    'database': 'palworld_database'
}

# === Connexion à MariaDB via SQLAlchemy engine ===
engine = create_engine(
    f"mariadb+mariadbconnector://{config['user']}:{config['password']}@{config['
)

# === Requête SQL ===
query2 = """
 select size, melee_attack, Remote_attack, defense, support from monsters;
"""

# === Lecture des données via pandas ===
df_size_comb_attr = pd.read_sql(query2, engine)

size_order = {'XS': 1, 'S': 2, 'M': 3, 'L': 4, 'XL': 5}
df_size_comb_attr['size_encoded'] = df_size_comb_attr['size'].map(size_order)

correlations = df_size_comb_attr[['size_encoded', 'melee_attack', 'Remote_attack
print(correlations['size_encoded'])

plt.figure(figsize=(8, 4))
sns.heatmap(correlations, annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Corrélation entre taille (encodée) et attributs de combat")
plt.show()
```

```
size_encoded     1.000000
melee_attack     0.301194
Remote_attack    0.484397
defense          0.422419
support         -0.072035
Name: size_encoded, dtype: float64
```



Analyse:

Il existe une corrélation modérée positive entre la taille des Pals et leur capacité de défense et attaque à distance. Cela signifie que les Pals de grande taille sont en moyenne plus résistants et mieux armés à distance.

La corrélation est plus faible pour l'attaque de mêlée et quasi inexistante pour les capacités de support, ce qui indique que ces rôles ne dépendent pas particulièrement de la taille.

L'attribut support montre même une légère corrélation négative, ce qui suggère que les Pals de plus petite taille peuvent exceller dans le soutien (collecte, soins,etc).

interpretation stratégique: Oui, la taille influence la performance au combat, mais principalement pour des rôles défensifs ou de puissance à distance.

Toutefois, elle n'est pas déterminante pour les rôles de mêlée ou de support, ce qui confirme l'importance de la diversité morphologique dans la constitution d'une équipe de Pals.

L. Corrélation entre la vitesse et l'efficacité en combat.

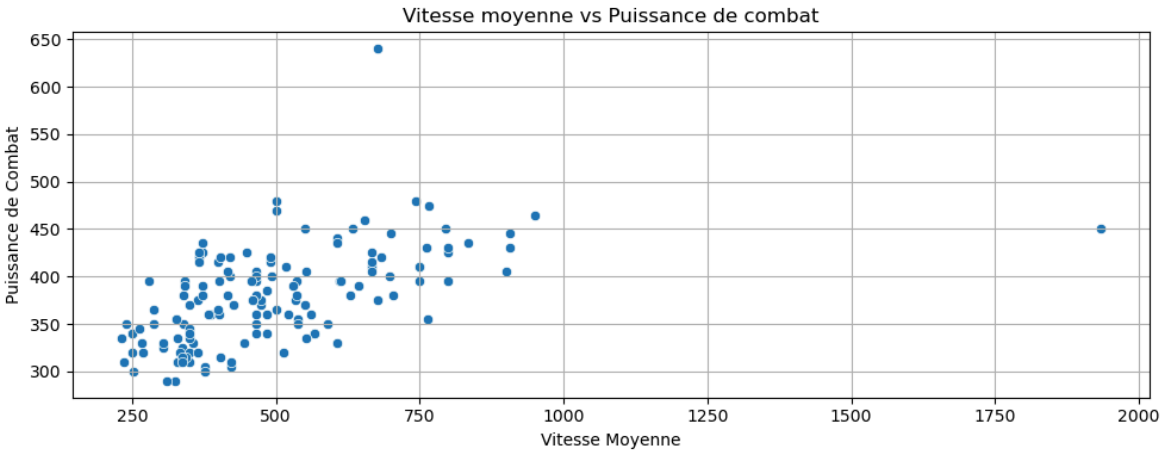
Le graphique ci-dessus visualise la relation entre la vitesse moyenne des Pals calculée à partir des vitesses de marche, course et sprint et leur puissance de combat totale, qui regroupe les attributs : melee_attack, remote_attack, defense, support.

In [163…

```
# Config base de données
config = {
    'user': 'root',
    'password': 'cN06+#P34',
```

```
    'host': 'localhost',
    'port': 3307,
    'database': 'palworld_database'
}

#  Connexion via SQLAlchemy engine ===
engine = create_engine(
    f"mariadb+mariadbconnector://{config['user']}:{config['password']}@{config['
)
query = """
SELECT OverrideNameTextID, walking_speed, running_speed, Riding_sprint_speed,
       melee_attack, remote_attack, defense, support
FROM monsters
WHERE walking_speed IS NOT NULL AND running_speed IS NOT NULL AND Riding_sprint_
  AND melee_attack IS NOT NULL AND remote_attack IS NOT NULL AND defense IS NOT
"""
df = pd.read_sql(query, engine)

# Moyenne des vitesses
df['avg_speed'] = df[['walking_speed', 'running_speed', 'Riding_sprint_speed']].

# Puissance de combat totale
df['combat_power'] = df[['melee_attack', 'remote_attack', 'defense', 'support']]

# Corrélation
correlation = df['avg_speed'].corr(df['combat_power'])
print(f" Corrélation entre vitesse moyenne et puissance de combat : {correlation

# Visualisation
plt.figure(figsize=(10, 4))
sns.scatterplot(data=df, x='avg_speed', y='combat_power')
plt.title("Vitesse moyenne vs Puissance de combat")
plt.xlabel("Vitesse Moyenne")
plt.ylabel("Puissance de Combat")
plt.grid(True)
plt.tight_layout()
plt.show()
```

```
Corrélation entre vitesse moyenne et puissance de combat : 0.54
```



Analyse:

La corrélation obtenue est de 0.54, ce qui indique une corrélation modérée à significative entre les deux variables.

On observe une tendance croissante sur le nuage de points : les Pals les plus rapides tendent à avoir une puissance de combat plus élevée.

Cependant, la dispersion reste notable, notamment chez les Pals très rapides mais au combat plus modeste.

conclusion:

Les Pals rapides sont généralement plus performants, mais la vitesse seule ne suffit pas à prédire la puissance.

La vitesse peut être vue comme un atout complémentaire, à combiner avec d'autres critères (attaque, endurance, taille, etc.) pour choisir un Pal de combat efficace.

Nous pouvons voir que la taille du pals a une correlation modérée sur les attributs de combat

M. Dreamteam en Phase 1 sur visualisation Streamlit

Objectif : Optimisation Grassland Phase 1

Cibler les Pals en Grasslands permet une progression rapide et sûre :

- Zones riches en ressources (bois, pierre, nourriture)
- Faible dangerosité en early game
- Nombreux Pals capturables avec une rareté raisonnable
- Progression naturelle par paliers de niveaux (1-3, 3-4, 4-6, 12)

Phase 1 – Équipe Ciblée (5 Profils)

1 – Garde-Combat Efficace *(Niv. 2-3)*

- **Rôle :** Défense + survie pendant la collecte
- **Impact :** Sécurisation de la progression

2 – Collecteur Polyvalent *(Niv. 3-4)*

- **Rôle :** Récolte bois + pierre pour outils / établi
- **Impact :** Lancement du crafting essentiel

3 – Capture XP Multi-Cibles *(Niv. 4-6)*

- **Rôle :** Farming XP par captures rapides
- **Impact :** Accès niveau 12 accéléré

4 – Monture Précoce *(Niv. 7-10)*

- **Rôle :** Mobilité + exploration élargie
- **Impact :** Déblocage de zones avancées

5 – Builder de Base *(Niv. 10 - 12)*

- **Rôle :** Construction rapide d'une base fonctionnelle
- **Impact :** Autonomie pour phase suivante

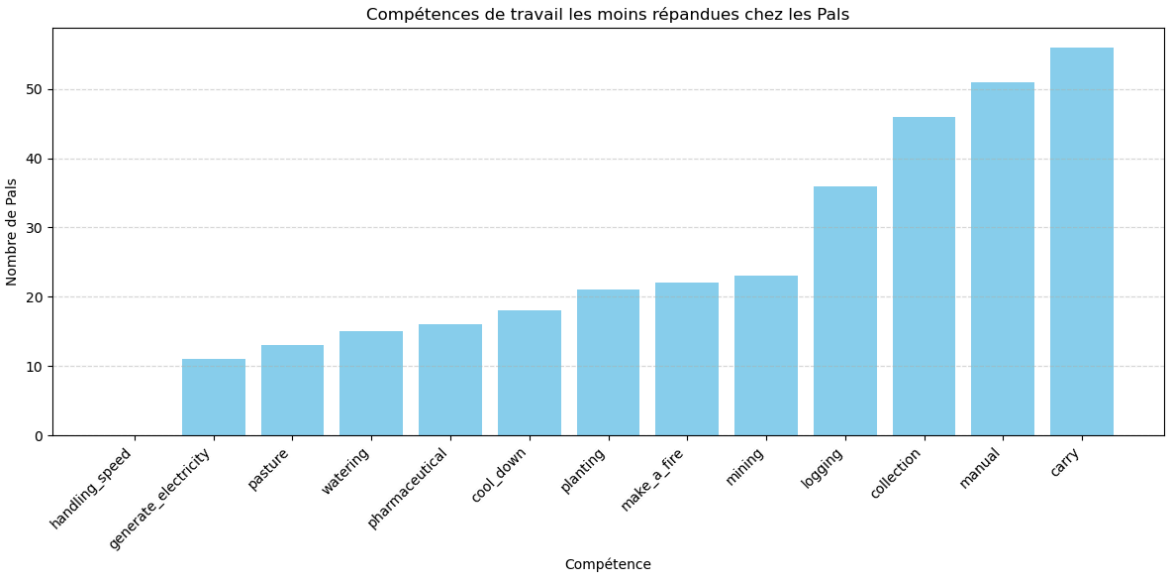
N et O. Mise en lumière des compétences de travail les plus et moins répendues

In [161…

```
#  répartition  compétences de travail

#  connexion à MariaDB
config = {
     'user': 'root',
    'password': 'cN06+#P34',
    'host': 'localhost',
    'port': 3307,
    'database': 'palworld_database'
}

# Connexion à la base de données
try:
    conn = mariadb.connect(**config)
    cursor = conn.cursor()
    print("Connexion réussie à MariaDB !")
except mariadb.Error as e:
    print(f"Erreur de connexion : {e}")
    exit(1)

# Requête SQL :
print("\n=== Compétences les moins répandues chez les Pals ===")
try:
    query = """
        SELECT 'make_a_fire' AS skill, SUM(make_a_fire) AS count FROM job_skills
        UNION ALL SELECT 'watering', SUM(watering) FROM job_skills
        UNION ALL SELECT 'planting', SUM(planting) FROM job_skills
        UNION ALL SELECT 'generate_electricity', SUM(generate_electricity) FROM
        UNION ALL SELECT 'manual', SUM(manual) FROM job_skills
        UNION ALL SELECT 'collection', SUM(collection) FROM job_skills
        UNION ALL SELECT 'logging', SUM(logging) FROM job_skills
        UNION ALL SELECT 'mining', SUM(mining) FROM job_skills
        UNION ALL SELECT 'pharmaceutical', SUM(pharmaceutical) FROM job_skills
        UNION ALL SELECT 'cool_down', SUM(cool_down) FROM job_skills
        UNION ALL SELECT 'pasture', SUM(pasture) FROM job_skills
        UNION ALL SELECT 'carry', SUM(carry) FROM job_skills
        UNION ALL SELECT 'handling_speed', SUM(handling_speed) FROM job_skills
        ORDER BY count ASC;
    """
    cursor.execute(query)
    results = cursor.fetchall()
```

```
    # DataFrame
    df = pd.DataFrame(results, columns=["Compétence", "Nombre de Pals"])

    # Affichage texte
    print(df)

    # Graphique:
    plt.figure(figsize=(12, 6))
    plt.bar(df["Compétence"], df["Nombre de Pals"], color="skyblue")
    plt.title("Compétences de travail les moins répandues chez les Pals")
    plt.xlabel("Compétence")
    plt.ylabel("Nombre de Pals")
    plt.xticks(rotation=45, ha='right')
    plt.tight_layout()
    plt.grid(axis="y", linestyle="--", alpha=0.5)
    plt.show()

except mariadb.Error as e:
    print(f"Erreur lors de la requête : {e}")

# Fermeture  connexion
cursor.close()
conn.close()
print("\nConnexion fermée.")
```

```
Connexion réussie à MariaDB !

=== Compétences les moins répandues chez les Pals ===
             Compétence Nombre de Pals
0         handling_speed              0
1   generate_electricity             11
2                pasture             13
3               watering             15
4         pharmaceutical             16
5              cool_down             18
6               planting             21
7           make_a_fire             22
8                mining             23
9               logging             36
10           collection             46
11               manual             51
12                carry             56
```



```
Connexion fermée.
```

Une analyse de la table `job_skills` a permis d'identifier les compétences les plus et les moins répandues chez les Pals.

Compétences les plus courantes :

- carry
- manual
- collection

Compétences les moins courantes :

- generate_electricity
- pasture
- watering

Ces résultats reflètent une répartition inégale des rôles dans le jeu, avec une forte concentration sur les tâches de base et une rareté des compétences spécialisées.

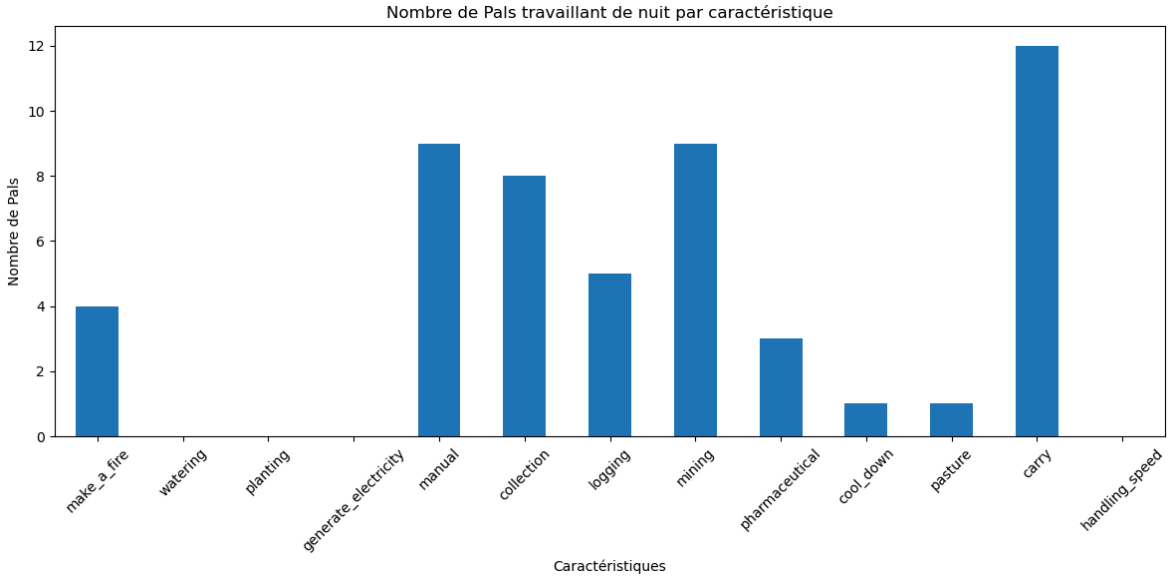
P. Nombre de Pals adaptés au travail de nuit

In [62]:
```
# Config base de données
config = {
    'user': 'root',
    'password': 'cN06+#P34',
    'host': 'localhost',
    'port': 3307,
    'database': 'palworld_database'
}

# Connexion via SQLAlchemy engine ===
engine = create_engine(
    f"mariadb+mariadbconnector://{config['user']}:{config['password']}@{config['
)

# === Requête SQL ===
query = """
SELECT COUNT(*) AS nombre_pals_nuit
FROM job_skills
WHERE night_shift = 1;
"""

# Exécution et affichage
result = pd.read_sql(query, engine)
nombre_pals = result.iloc[0]['nombre_pals_nuit']

print(f"Nombre de Pals adaptés au travail de nuit : {nombre_pals}")

#  Fermeture du engine
engine.dispose()
```

```
Nombre de Pals adaptés au travail de nuit : 25
```

Q. Caractéristiques communes des Pals qui conviennent au travail de nuit

In [86]:
```
# Configuration  base de données
config = {
    'user': 'root',
    'password': 'cN06+#P34',
    'host': 'localhost',
    'port': 3307,
    'database': 'palworld_database'
}

# Connexion via SQLAlchemy engine
engine = create_engine(
    f"mariadb+mariadbconnector://{config['user']}:{config['password']}@{config['
)

# Requête SQL  caractéristiques des Pals  travail de nuit
query = """
SELECT make_a_fire, watering, planting,
       generate_electricity, manual, collection, logging, mining, pharmaceutical
       pasture, carry, handling_speed
FROM job_skills
WHERE night_shift = 1;
"""

# Lecture des données via pandas
df_night_shift = pd.read_sql(query, engine)

# Fermeture  connexion
engine.dispose()

# Compter le nombre de Pals ayant chaque caractéristique
characteristics_count = df_night_shift.sum()

# Création du graphique en barres
plt.figure(figsize=(12, 6))
characteristics_count.plot(kind='bar')
plt.xlabel('Caractéristiques')
plt.ylabel('Nombre de Pals')
plt.title('Nombre de Pals travaillant de nuit par caractéristique')
plt.xticks(rotation=45)
plt.tight_layout()

# Affichage du graphique
plt.show()
```



Les résultats montrent que les Pals travaillant de nuit ont des compétences particulièrement élevées dans les domaines suivants :

- **Mining** : Cette caractéristique est cruciale pour l'extraction des ressources, ce qui suggère que les Pals travaillant de nuit sont efficaces pour les tâches d'extraction minière.
- **Manual** : Les compétences manuelles indiquent une capacité à effectuer des tâches physiques et de construction, ce qui est essentiel pour le développement et l'entretien des infrastructures.
- **Logging** : Cette compétence est importante pour la collecte de bois et d'autres ressources forestières, ce qui est vital pour la construction et le crafting.

Interprétation:

Les Pals travaillant de nuit semblent être optimisés pour des tâches qui nécessitent une forte capacité physique et une efficacité dans la collecte de ressources. Ces caractéristiques suggèrent que les Pals travaillant de nuit sont particulièrement utiles pour les opérations nécessitant une main-d'œuvre robuste et efficace dans des conditions de faible luminosité, ce qui peut être un avantage significatif pour les tâches nocturnes.

R. Analyse des Pals par Compétences et Rareté

In [87]:
```
# Configuration de la base de données
config = {
    'user': 'root',
    'password': 'cN06+#P34',
    'host': 'localhost',
    'port': 3307,
    'database': 'palworld_database'
}

# Connexion  via SQLAlchemy engine
engine = create_engine(
    f"mariadb+mariadbconnector://{config['user']}:{config['password']}@{config['
)
```

```
# Requête SQL Pals avec + de compétences
query = """
SELECT m.rarity
FROM monsters m
JOIN job_skills j ON m.id = j.id
ORDER BY j.total_skills DESC
LIMIT 10;
"""

# Lecture des données via pandas
df = pd.read_sql(query, engine)

# Fermeture de la connexion
engine.dispose()

# Calcul  rareté moyenne  Pals sélectionnés
rarity_mean = df['rarity'].mean()

# Affichage résultat
print(f"La rareté moyenne des Pals possédant le plus de compétences est : {rarit
```

```
La rareté moyenne des Pals possédant le plus de compétences est : 5.5
```

In [ ]:
```
## Méthodologie
- Jointure de Tables: Utilisation d'une jointure entre `monsters` et `job_skills
- Tri et Sélection: Les Pals sont triés par nombre de compétences et rareté, pui
```

S. Quels sont les Pals qui ont la vitesse de travail la plus élevée ?

In [32]:
```
# === Config base de données ===
config = {
    'user': 'root',
    'password': 'cN06+#P34',
    'host': 'localhost',
    'port': 3307,
    'database': 'palworld_database'
}

# === Connexion à MariaDB via SQLAlchemy engine ===
engine = create_engine(
    f"mariadb+mariadbconnector://{config['user']}:{config['password']}@{config['
)

# === Requête SQL ===
query = """
SELECT OverrideNameTextID, CraftSpeed
FROM monsters
WHERE CraftSpeed IS NOT NULL
ORDER BY CraftSpeed DESC
LIMIT 144;
"""

# === Lecture des données via pandas ===
df = pd.read_sql(query, engine)

# === Affichage des résultats ===
print("=== Pals les moins feignants ===")
```

```
for index, row in df.iterrows():
    print(f"{index}: {row['OverrideNameTextID']} - Vitesse de travail: {row['Cra

# === Fermeture du engine ===
engine.dispose()
```

```
=== Pals les moins feignants ===
0: SakuraSaurus_Water - Vitesse de travail: 100
1: AmaterasuWolf - Vitesse de travail: 100
2: RobinHood - Vitesse de travail: 100
3: RedArmorBird - Vitesse de travail: 100
4: ColorfulBird - Vitesse de travail: 100
5: Kelpie_Fire - Vitesse de travail: 100
6: PinkRabbit - Vitesse de travail: 100
7: SakuraSaurus - Vitesse de travail: 100
8: Alpaca - Vitesse de travail: 100
9: CatVampire - Vitesse de travail: 100
10: ChickenPal - Vitesse de travail: 100
11: SharkKid - Vitesse de travail: 100
12: Kelpie - Vitesse de travail: 100
13: Ronin - Vitesse de travail: 100
14: PlantSlime - Vitesse de travail: 100
15: PlantSlime - Vitesse de travail: 100
16: IceFox - Vitesse de travail: 100
17: CatBat - Vitesse de travail: 100
18: CatMage - Vitesse de travail: 100
19: ElecLion - Vitesse de travail: 100
20: Carbunclo - Vitesse de travail: 100
21: RobinHood_Ground - Vitesse de travail: 100
22: IceDeer - Vitesse de travail: 100
23: FengyunDeeper - Vitesse de travail: 100
24: Penguin - Vitesse de travail: 100
25: CaptainPenguin - Vitesse de travail: 100
26: ElecCat - Vitesse de travail: 100
27: DarkScorpion - Vitesse de travail: 100
28: Bastet_Ice - Vitesse de travail: 100
29: Hedgehog_Ice - Vitesse de travail: 100
30: WhiteTiger - Vitesse de travail: 100
31: WindChimes_Ice - Vitesse de travail: 100
32: BluePlatypus - Vitesse de travail: 100
33: NegativeOctopus - Vitesse de travail: 100
34: GrassPanda - Vitesse de travail: 100
35: WindChimes - Vitesse de travail: 100
36: WizardOwl - Vitesse de travail: 100
37: Bastet - Vitesse de travail: 100
38: GhostBeast - Vitesse de travail: 100
39: Ganesha - Vitesse de travail: 100
40: Eagle - Vitesse de travail: 100
41: SkyDragon - Vitesse de travail: 100
42: DreamDemon - Vitesse de travail: 100
43: VolcanicMonster_Ice - Vitesse de travail: 100
44: CuteButterfly - Vitesse de travail: 100
45: BlackGriffon - Vitesse de travail: 100
46: Monkey - Vitesse de travail: 100
47: Baphomet_Dark - Vitesse de travail: 100
48: ElecPanda - Vitesse de travail: 100
49: Werewolf - Vitesse de travail: 100
50: LizardMan - Vitesse de travail: 100
51: Suzaku - Vitesse de travail: 100
52: SheepBall - Vitesse de travail: 100
53: SweetsSheep - Vitesse de travail: 100
54: GrassMammoth - Vitesse de travail: 100
55: MopBaby - Vitesse de travail: 100
56: MopKing - Vitesse de travail: 100
57: FairyDragon_Water - Vitesse de travail: 100
58: FlowerRabbit - Vitesse de travail: 100
```

59: LazyDragon_Electric - Vitesse de travail: 100
60: Serpent_Ground - Vitesse de travail: 100
61: Suzaku_Water - Vitesse de travail: 100
62: Serpent - Vitesse de travail: 100
63: BerryGoat - Vitesse de travail: 100
64: Kitsunebi - Vitesse de travail: 100
65: FireKirin - Vitesse de travail: 100
66: Baphomet - Vitesse de travail: 100
67: FlameBuffalo - Vitesse de travail: 100
68: LizardMan_Fire - Vitesse de travail: 100
69: BirdDragon - Vitesse de travail: 100
70: FoxMage - Vitesse de travail: 100
71: VolcanicMonster - Vitesse de travail: 100
72: FlameBambi - Vitesse de travail: 100
73: Manticore - Vitesse de travail: 100
74: Manticore_Dark - Vitesse de travail: 100
75: Garm - Vitesse de travail: 100
76: CuteFox - Vitesse de travail: 100
77: WeaselDragon - Vitesse de travail: 100
78: HawkBird - Vitesse de travail: 100
79: Yeti - Vitesse de travail: 100
80: NegativeKoala - Vitesse de travail: 100
81: BlueDragon - Vitesse de travail: 100
82: Deer_Ground - Vitesse de travail: 100
83: Mutant - Vitesse de travail: 100
84: NaughtyCat - Vitesse de travail: 100
85: WoolFox - Vitesse de travail: 100
86: FairyDragon - Vitesse de travail: 100
87: Deer - Vitesse de travail: 100
88: SharkKid_Fire - Vitesse de travail: 100
89: WhiteMoth - Vitesse de travail: 100
90: Yeti_Grass - Vitesse de travail: 100
91: Umihebi_Fire - Vitesse de travail: 100
92: FlowerDoll - Vitesse de travail: 100
93: FlowerDinosaur - Vitesse de travail: 100
94: Boar - Vitesse de travail: 100
95: LittleBriarRose - Vitesse de travail: 100
96: Horus - Vitesse de travail: 100
97: VioletFairy - Vitesse de travail: 100
98: LavaGirl - Vitesse de travail: 100
99: LazyCatfish - Vitesse de travail: 100
100: GrassRabbitMan - Vitesse de travail: 100
101: ThunderBird - Vitesse de travail: 100
102: CuteMole - Vitesse de travail: 100
103: FireKirin_Dark - Vitesse de travail: 100
104: GrassMammoth_Ice - Vitesse de travail: 100
105: FlowerDinosaur_Electric - Vitesse de travail: 100
106: HadesBird - Vitesse de travail: 100
107: GrassPanda_Electric - Vitesse de travail: 100
108: Kirin - Vitesse de travail: 100
109: Hedgehog - Vitesse de travail: 100
110: BirdDragon_Ice - Vitesse de travail: 100
111: NightFox - Vitesse de travail: 100
112: ThunderDog - Vitesse de travail: 100
113: FlyingManta - Vitesse de travail: 100
114: DarkCrow - Vitesse de travail: 100
115: Anubis - Vitesse de travail: 100
116: SaintCentaur - Vitesse de travail: 100
117: IceHorse_Dark - Vitesse de travail: 100
118: JetDragon - Vitesse de travail: 100

```
119: IceHorse - Vitesse de travail: 100
120: LazyDragon - Vitesse de travail: 100
121: PInkCat - Vitesse de travail: 100
122: ThunderDragonMan - Vitesse de travail: 100
123: BlackCentaur - Vitesse de travail: 100
124: KingBahamut - Vitesse de travail: 100
125: DrillGame - Vitesse de travail: 100
126: Umihebi - Vitesse de travail: 100
127: Gorilla - Vitesse de travail: 100
128: BlackMetalDragon - Vitesse de travail: 100
129: VolcanoBoss - Vitesse de travail: 100
130: CowPal - Vitesse de travail: 100
131: KingAlpaca - Vitesse de travail: 100
132: ForestBoss - Vitesse de travail: 100
133: DessertBoss - Vitesse de travail: 100
134: RaijinDaughter - Vitesse de travail: 100
135: SnowBoss - Vitesse de travail: 100
136: GrassBoss - Vitesse de travail: 100
137: PinkLizard - Vitesse de travail: 100
138: QueenBee - Vitesse de travail: 100
139: KingAlpaca_Ice - Vitesse de travail: 100
140: LilyQueen - Vitesse de travail: 100
141: HerculesBeetle - Vitesse de travail: 100
142: SoldierBee - Vitesse de travail: 100
143: LilyQueen_Dark - Vitesse de travail: 100
```

Analyse de la Vitesse de Travail des Pals:

Tous les Pals ont la même vitesse de travail selon les données du CSV.

In [ ]:
```
### T. Stratégie optimale de capture des Pals
```

In [60]:
```
# Config base de données
config = {
    'user': 'root',
    'password': 'cN06+#P34',
    'host': 'localhost',
    'port': 3307,
    'database': 'palworld_database'
}

# Connexion via SQLAlchemy engine ===
engine = create_engine(
    f"mariadb+mariadbconnector://{config['user']}:{config['password']}@{config['
)

#  Requête SQL
query = """
SELECT OverrideNameTextID, CaptureProbability, Nocturnal FROM monsters
ORDER BY CaptureProbability DESC
LIMIT 20;
"""

# Lecture des données
df = pd.read_sql(query, engine)

#  Affichage résultats
print("=== Pals les plus faciles à capturer ===")
for index, row in df.iterrows():
```

```
    print(f"{index}: {row['OverrideNameTextID']} - Capture: {row['CaptureProbabi

# Fermeture du engine
engine.dispose()
```

```
=== Pals les plus faciles à capturer ===
0: ChickenPal - Capture: 1.05 - Non-nocturne
1: SheepBall - Capture: 1.05 - Non-nocturne
2: PInkCat - Capture: 1.05 - Non-nocturne
3: VolcanoBoss - Capture: 1 - Non-nocturne
4: ForestBoss - Capture: 1 - Non-nocturne
5: DessertBoss - Capture: 1 - Non-nocturne
6: SnowBoss - Capture: 1 - Non-nocturne
7: GrassBoss - Capture: 1 - Non-nocturne
8: PlantSlime - Capture: 0.91 - Non-nocturne
9: PlantSlime - Capture: 0.91 - Non-nocturne
10: Ganesha - Capture: 0.91 - Non-nocturne
11: CuteFox - Capture: 0.91 - Non-nocturne
12: WoolFox - Capture: 0.91 - Non-nocturne
13: Hedgehog - Capture: 0.84 - Non-nocturne
14: ElecCat - Capture: 0.77 - Non-nocturne
15: BluePlatypus - Capture: 0.77 - Non-nocturne
16: Monkey - Capture: 0.77 - Non-nocturne
17: MopBaby - Capture: 0.77 - Non-nocturne
18: Kitsunebi - Capture: 0.77 - Non-nocturne
19: FlameBambi - Capture: 0.77 - Non-nocturne
```

Connaître les Pals les plus faciles à capturer dans Palworld permet de cibler efficacement ceux-ci, optimisant ainsi les ressources, le temps et la progression dans le jeu. Cela aide à construire rapidement une équipe solide et à planifier stratégiquement les sessions de jeu. De plus, savoir s'ils sont nocturnes ou non permet d'adapter les périodes de chasse pour maximiser les chances de capture.

U. Quel Tower Boss a le score d'attributs de combat combinés le plus élevé ?

In [7]:
```
# Config base de données
config = {
    'user': 'root',
    'password': 'cN06+#P34',
    'host': 'localhost',
    'port': 3307,
    'database': 'palworld_database'
}
# Reconnexion à la base de données
try:
    conn = mariadb.connect(**config)
    cursor = conn.cursor()
    print("Connexion réussie à MariaDB !")
except mariadb.Error as e:
    print(f"Erreur de connexion : {e}")
    exit(1)

    #  Requête SQL
query = """
SELECT id, name,
  (HP + melee_attack + remote_attack + defense) / 2 AS medium
FROM
```

```
   tower_bosses
ORDER BY
   medium DESC
LIMIT 1;
"""
try:
    cursor.execute(query)
    result = cursor.fetchone()
    if result:
        print('Le gagnant est : ')
    print(f"ID : {result[0]}, Nom : {result[1]}, Moyenne de combat: {result[2]}"
    conn.commit()
except mariadb.Error as e:
    print(f" Erreur création table : {e}")
    conn.close()
    exit(1)
```

```
Connexion réussie à MariaDB !
Le gagnant est :
ID : 1, Nom : Victor & Heterogeneous Griffin, Moyenne de combat: 4275.0000
```

Sélection du Tower Boss le Plus Puissant

Choix des colonnes

Les colonnes utilisées dans la requête SQL sont les suivantes :

- `id` : identifiant unique du boss.
- `name` : nom du boss pour l'affichage.
- Une moyenne calculée à partir de `HP`, `melee_attack`, `remote_attack` et `defense`.

Ces quatre attributs sont essentiels dans un contexte de combat :

- **HP** évalue la capacité de survie.
- **melee_attack** et **remote_attack** couvrent l'ensemble des types d'agression possibles.
- **defense** mesure la résistance générale.

La combinaison de ces données permet une comparaison cohérente entre les entités, en mettant en valeur leur efficacité globale en combat.

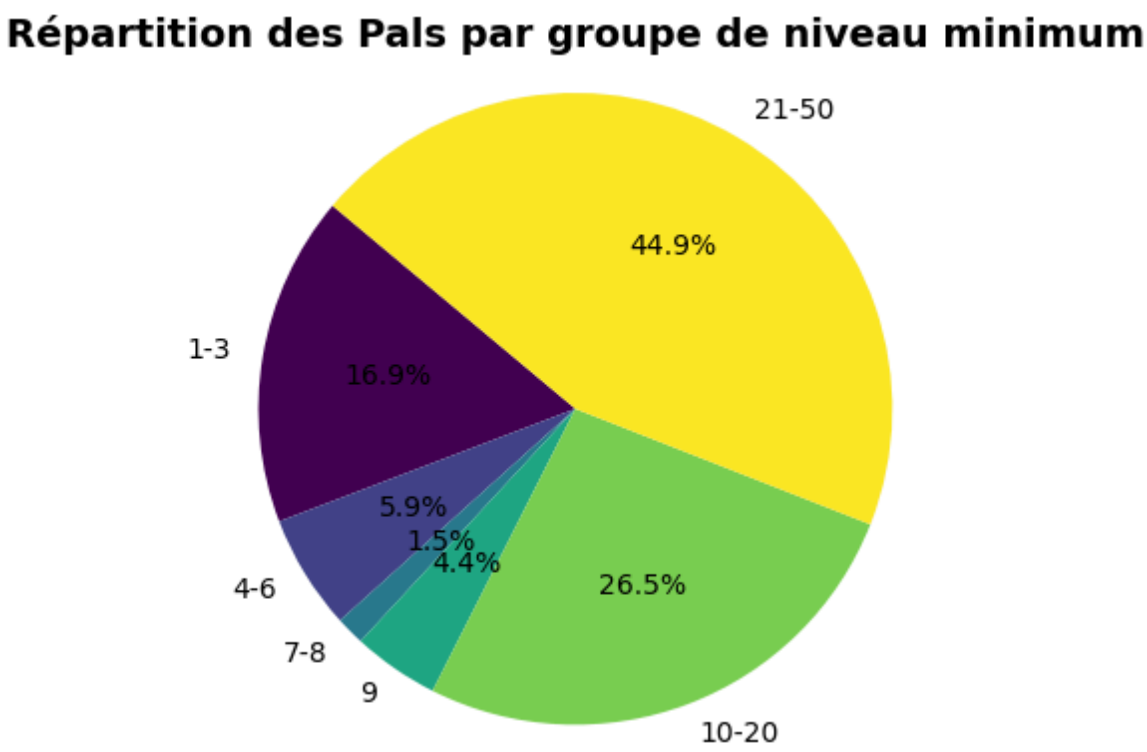
L'approche garantit une sélection objective du boss le plus complet sur le plan statistique.

V. Quelle est la répartition des niveaux d'apparition des Pals ?

In [161…

```
# Connexion via SQLAlchemy ===
config = {
    'user': 'root',
    'password': 'cN06+#P34',
    'host': 'localhost',
    'port': 3307,
    'database': 'palworld_database'
```

```
}

# Création de l'engine
engine = create_engine(
    f"mariadb+mariadbconnector://{config['user']}:{config['password']}@{config['
)

# Requête SQL
query = """
SELECT minimum_level
FROM palu_refresh_level
WHERE minimum_level IS NOT NULL;
"""

# Lecture des données
df = pd.read_sql(query, engine)

# Définition des tranches
bins = [0, 3, 6, 8, 9, 20, 50]
labels = ['1-3', '4-6', '7-8', '9', '10-20', '21-50']

# Découpage des niveaux minimum en catégories
df['level_group'] = pd.cut(df['minimum_level'], bins=bins, labels=labels, right=

# Comptage des valeurs par groupe
group_counts = df['level_group'].value_counts().sort_index()

#  Affichage  camembert
plt.figure(figsize=(4, 4))
plt.pie(
    group_counts,
    labels=group_counts.index,
    autopct='%1.1f%%',
    startangle=140,
    colors=plt.cm.viridis(np.linspace(0, 1, len(group_counts)))
)
plt.title("Répartition des Pals par groupe de niveau minimum", fontsize=14, weig
plt.axis('equal')
plt.tight_layout()
plt.show()

# Fermeture
engine.dispose()
```



Ce que notre analyse de répatition des niveaux min révèle : Notre visualisation nous permet d'identifier les proportions exactes de Pals accessibles selon notre capacité réelle de capture en début de jeu. L'insight de notre analyse : En observant la répartition dans notre graphique en camembert, nous pouvons directement voir :

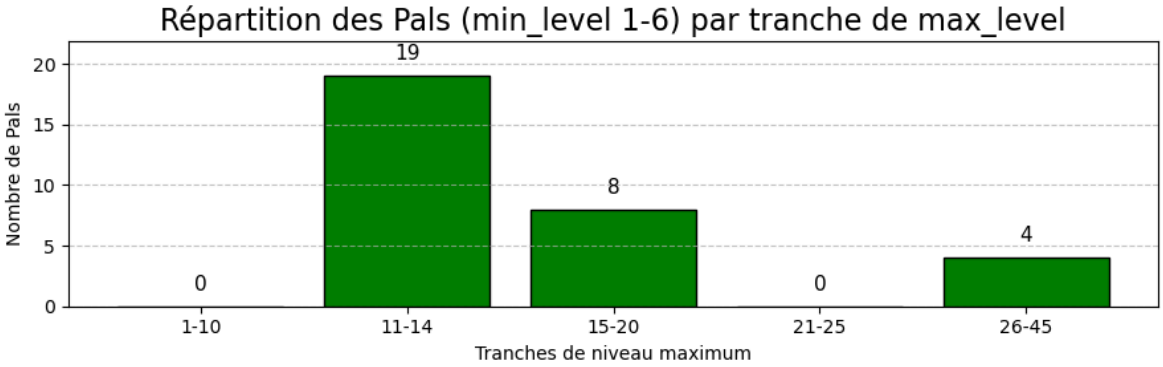
Quelle proportion de Pals est en capture quasi-garantie (niveau 1-3) Quelle proportion reste viable avec de bonnes chances (niveau 4-6) Quelle proportion demande plus d'efforts mais reste possible (niveau 7-8) Quelle proportion est pratiquement hors de portée (niveau 9+)

L'utilité stratégique : Notre code nous donne une cartographie précise de l'accessibilité du contenu Pals. Nous savons exactement quelle portion du roster nous est réellement disponible selon nos contraintes de capture en début de partie. Cette visualisation transforme les mécaniques abstraites de capture en données concrètes sur la richesse du contenu accessible.

In [162…

```
# Configuration base de données
config = {
    'user': 'root',
    'password': 'cN06+#P34',
    'host': 'localhost',
    'port': 3307,
    'database': 'palworld_database'
}

# Création du moteur SQLAlchemy
engine = create_engine(
    f"mariadb+mariadbconnector://{config['user']}:{config['password']}@{config['
)

# Requête SQL pour Pals avec minimum_level entre 1 et 3
query = """
```

```
SELECT name, minimum_level, maximum_level
FROM palu_refresh_level
WHERE minimum_level BETWEEN 1 AND 6;
AND ispal
"""

df = pd.read_sql(query, engine)

# Définition des tranches maximum_level
bins = [0, 10, 14, 20, 25, 45]
labels = ['1-10', '11-14', '15-20', '21-25', '26-45']
df['max_level_group'] = pd.cut(df['maximum_level'], bins=bins, labels=labels, ri

# Comptage du nombre de Pals / groupe
group_counts = df['max_level_group'].value_counts().sort_index()

# Visualisation en barres
plt.figure(figsize=(9,3))
bars = plt.bar(group_counts.index, group_counts.values, color='green', edgecolor

# Ajouter les valeurs au-dessus des barres
for bar in bars:
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, height + 1, f'{int(height)}', ha='

plt.title("Répartition des Pals (min_level 1-6) par tranche de max_level", fonts
plt.xlabel("Tranches de niveau maximum")
plt.ylabel("Nombre de Pals")
plt.ylim(0, max(group_counts.values)*1.15)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()

engine.dispose()
```



Stratégie de départ : sélection optimisée des Pals

Nous avons filtré les Pals avec un minimum_level entre 1 et 6 pour identifier ceux accessibles dès le début. En regroupant leur maximum_level par tranches (1–10, 10–14, 15–20, 20–25, 25–45), nous avons évalué leur potentiel d'évolution.

Les Pals de bas niveau avec un fort niveau max offrent un excellent retour sur investissement : ils restent utiles plus longtemps, optimisent la gestion des ressources et limitent les remplacements précoces.

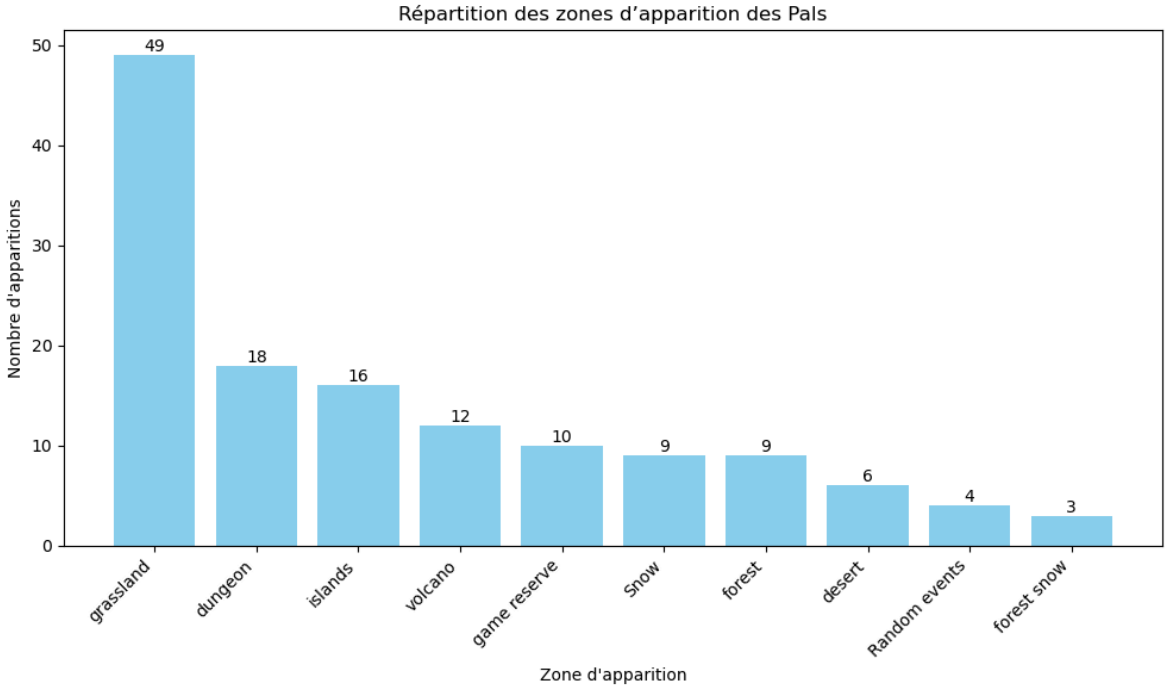
Cette approche guide un choix stratégique dès les premières heures de jeu.

W. Quelle est la répartition des zones d'apparition ?

In [170…

```
# Config base de données
config = {
    'user': 'root',
    'password': 'cN06+#P34',
    'host': 'localhost',
    'port': 3307,
    'database': 'palworld_database'
}

# Connexion via SQLAlchemy engine
engine = create_engine(
    f"mariadb+mariadbconnector://{config['user']}:{config['password']}@{config['
)

# Requête SQL
query = """
SELECT refresh_area, COUNT(*) AS total
FROM palu_refresh_level
GROUP BY refresh_area
ORDER BY total DESC;
"""

# Lecture des données
df = pd.read_sql(query, engine)

#  Affichage graphique
plt.figure(figsize=(10, 6))
bars = plt.bar(df['refresh_area'], df['total'], color='skyblue')
plt.xlabel("Zone d'apparition")
plt.ylabel("Nombre d'apparitions")
plt.title("Répartition des zones d'apparition des Pals")

# Ajout nombre sur chaque barre
for bar in bars:
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, height, f'{int(height)}', ha='cent

plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

```
# Fermeture du engine
engine.dispose()
```



Ce que notre analyse révèle : Notre visualisation identifie la densité de Pals par zone d'apparition. L'utilité stratégique : Nous savons exactement quelles zones prioriser pour maximiser nos rencontres de nouveaux Pals et quelles zones éviter car pauvres en diversité. Notre analye transforme l'exploration aléatoire en stratégie d'exploration ciblée.

In [158…

```
# Exemple de jointure par le nom
# Configuration de la connexion
config = {
     'user': 'root',
    'password': 'cN06+#P34',
    'host': 'localhost',
    'port': 3307,
    'database': 'palworld_database'
}

# Connexion  base de données
try:
    conn = mariadb.connect(**config)
    cursor = conn.cursor()
    print("Connexion réussie à MariaDB !")
except mariadb.Error as e:
    print(f"Erreur de connexion : {e}")
    exit(1)

# Requête avec tri palu_combat_attribute
print("\n=== job_skills réorganisé selon palu_combat_attribute ===")
try:
    query = """
        SELECT j.*
        FROM job_skills j
        JOIN palu_combat_attribute p
        ON j.chinese_name = p.chinese_name
        ORDER BY p.id
```

```
    """
    cursor.execute(query)
    rows = cursor.fetchall()

    # Récupération des noms de colonnes
    cursor.execute("""
        SELECT column_name
        FROM information_schema.columns
        WHERE table_name = 'job_skills'
        AND table_schema = 'palworld_database'
        ORDER BY ordinal_position
    """)
    columns = [col[0] for col in cursor.fetchall()]

    # Affichage
    df = pd.DataFrame(rows, columns=columns)
    print(df)

except mariadb.Error as e:
    print(f"Erreur lors de l'exécution de la requête : {e}")

# Fermeture
cursor.close()
conn.close()
print("\nConnexion fermée.")
```

```
Connexion réussie à MariaDB !

=== job_skills réorganisé selon palu_combat_attribute ===
     id     english_name   chinese_name volume_size  food_intake  night_shift  \
0    83          lamball     mianyouyou    smallest            2            0
1    73          chikipi    pipichicken    smallest            1            0
2     1          lifmunk   greenleafrat    smallest            1            0
3     2         foxparks      tinderfox    smallest            2            0
4    85            fuack       surfduck    smallest            2            0
..   ..              ...            ...         ...          ...          ...
135  64         paladius  knightoflight         big            9            0
136  65         necromus    chaosknight         big            9            1
137  66      frostallion   wintercaller         big            7            0
138  67  frostallionnoct    nightcaller         big            7            1
139  68         jetragon   vortexdragon     maximum            9            0

     total_skills  make_a_fire  watering  planting  ...  logging  mining  \
0               3            0         0         0  ...        0       0
1               2            0         0         0  ...        0       0
2               5            0         0         1  ...        1       0
3               1            1         0         0  ...        0       0
4               3            0         1         0  ...        0       0
..            ...          ...       ...       ...  ...      ...     ...
135             4            0         0         0  ...        1       1
136             4            0         0         0  ...        1       1
137             4            0         0         0  ...        0       0
138             4            0         0         0  ...        0       0
139             3            0         0         0  ...        0       0

     pharmaceutical  cool_down  pasture  carry  handling_speed  ranch_items  \
0                 0          0        1      1             0.0         wool
1                 0          0        1      0             NaN          Egg
2                 1          0        0      0             NaN         None
3                 0          0        0      0             NaN         None
4                 0          0        0      1             0.0         None
..              ...        ...      ...    ...             ...          ...
135               0          0        0      0             NaN         None
136               0          0        0      0             NaN         None
137               0          1        0      0             NaN         None
138               0          0        0      0             NaN         None
139               0          0        0      0             NaN         None

     pasture_minimum_output  the_largest_ranch
0                         1               Rank
1                         1               Rank
2                      None               None
3                      None               None
4                      None               None
..                      ...                ...
135                    None               None
136                    None               None
137                    None               None
138                    None               None
139                    None               None

[140 rows x 23 columns]

Connexion fermée.
```