Summary Document

```
import pandas as pd
import numpy as np
import re
```

Loading the Data

```
df = pd.read_csv(“messy_data1.csv”)
```

Display initial data overview

```
print("Initial Data Overview:")
display(df.head())
display(df.info())
```

Inspect the Data

```
print(df.head())
print(df.info())
```

Record QA Issues (Export QA Issues to Excel)

```
qa_issues = []

def log_issue(field, issue, issue_type, line_number, solution, suggestion):
    qa_issues.append({
        "Field": field,
        "QA Issue": issue,
        "Issue Type": issue_type,
        "Data Line Number": line_number,
        "Solution": solution,
        "Suggestion": suggestion
    })
```

Handle Missing

```
df.dropna(subset=["ID", "Name", "Email", "Join Date"], inplace=True)  # Drop critical missing data
df["Age"].fillna(df["Age"].median(), inplace=True)
df["Salary"].fillna(df["Salary"].mean(), inplace=True)
df["Department"].fillna("Unknown", inplace=True)
```

Remove Duplicates

```
df.drop_duplicates(subset=["ID"], keep="first", inplace=True)
```

Correct Email Formats

```
def validate_email(email):
    pattern = r'^[a-zA-Z0-9_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+$'
    return bool(re.match(pattern, str(email)))
```

```
df = df[df["Email"].apply(validate_email)]

def is_professional_email(email):
    return not any(domain in email for domain in ["gmail.com", "yahoo.com", "hotmail.com"])

df = df[df["Email"].apply(is_professional_email)]
```

Clean Name Fields

```
df["Name"] = df["Name"].str.strip().str.title()
```

Standardize Date Formats

```
df["Join Date"] = pd.to_datetime(df["Join Date"], errors='coerce')

df.dropna(subset=["Join Date"], inplace=True)  # Remove rows where date conversion failed
```

Correct Department Names

```
Department_mapping = {
    "Hr": "HR", "Human Resources": "HR",
    "Eng": "Engineering", "Engg": "Engineering", "Engineer": "Engineering",
    "Mkt": "Marketing", "Mkting": "Marketing",
    "Sal": "Sales", "Sls": "Sales",
    "Supp": "Support", "Cust Support": "Support"
}
df["Department"] = df["Department"].replace(Department_mapping)
#df["department"] = df["department"].replace(Department_mapping)
df["Department"] = df["Department"].str.replace(r'^Sal.*$', 'Sales', regex=True)
df["Department"] = df["Department"].str.replace(r'^HR.*$', 'Human Resource', regex=True)
df["Department"] = df["Department"].str.replace(r'^Eng.*$', 'Engineering', regex=True)
df["Department"] = df["Department"].str.replace(r'^Mark.*$', 'Marketing', regex=True)
df["Department"] = df["Department"].str.replace(r'^Sup.*$', 'Support', regex=True)
```

Handle Salary Noise

```
q99 = df["Salary"].quantile(0.99)
df["Salary"] = df["Salary"].clip(upper=q99)
```

Export QA Issues to Excel

```
qa_df = pd.DataFrame(qa_issues)
qa_df.to_excel("QA_Issues_Report.xlsx", index=False)
```

Save the Cleaned Data

```
df.to_csv("cleaned_data.csv", index=False)

print("\nData Cleaning Completed!")
```