Environment: Python 3.8.5 and Jupyter notebook

Libraries used:

- pandas
- re
- uuid

## Import libraries

```
# Code to import libraries as you need in this assessment, e.g.,
import pandas as pd
import re
import uuid
```

## Integration

integrate two dataset to get desired outcome and resolve any schema level and data level conflicts.

```
# Code to audit data
```

```
# Read dataset2
df2 = pd.read_csv('dataset2.csv')
```

### Preliminary data checks

```
print(df2.shape)
```

```
(334, 9)
```

```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 334 entries, 0 to 333
Data columns (total 9 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   Opening           334 non-null    object
 1   Closing           334 non-null    object
 2   Job Title         334 non-null    object
 3   Organisation      334 non-null    object
 4   Location          334 non-null    object
 5   Category          334 non-null    object
 6   Salary per month  334 non-null    int64 
 7   Fraction          3 non-null      object
 8   Contract Type     327 non-null    object
dtypes: int64(1), object(8)
memory usage: 23.6+ KB
```

```
df2.head()
```

| | Opening | Closing | Job Title | Organisation | Location | Category | Salary per month | Fraction | Contract Type |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 2013-10-06 00:00:00 | 2013-12-05 00:00:00 | Aviation loans administration | cer Financial | London | Finance and Accounting | 2800 | NaN | Contract |
| **1** | 2012-10-03 12:00:00 | 2012-11-02 12:00:00 | Payroll Analyst City upto **** , **** | LMA Recruitment Ltd | London | Finance and Accounting | 2917 | NaN | Permanent |
| **2** | 2012-01-01 00:00:00 | 2012-01-31 00:00:00 | Investment Team Assistant for leading Private ... | Austin Andrew Ltd | London | Finance and Accounting | 3750 | NaN | Permanent |
| | | | SWAPS | | | | | | |

```
df2.columns
```

```
Index(['Opening', 'Closing', 'Job Title', 'Organisation', 'Location',
       'Category', 'Salary per month', 'Fraction', 'Contract Type'],
      dtype='object')
```

```
# Checking Values for fraction
df2['Fraction'].unique()
```

```
array([nan, 'Part Time'], dtype=object)
```

```
df2[df2['Fraction'].notnull()]
```

| | Opening | Closing | Job Title | Organisation | Location | Category | Salary per month | Fraction | Contract Type |
|---|---|---|---|---|---|---|---|---|---|
| **88** | 2013-05-21 15:00:00 | 2013-06-20 15:00:00 | Investment Operations Reconciliations Administ... | James Associates Recruitment Ltd | London | Finance and Accounting | 2292 | Part Time | NaN |
| **120** | 2013-12-10 15:00:00 | 2014-02-08 15:00:00 | Compliance Supervisor | MW Recruitment Ltd | London | Finance and Accounting | 5833 | Part Time | NaN |

```
#### Lets read the dataset1_solution and compare the columns and check both dataset
df1 = pd.read_csv('dataset1_solution.csv')
```

▾ Preliminary data checks

```
print(df1.shape)
```

```
(55169, 11)
```

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 55169 entries, 0 to 55168
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Id            55169 non-null  int64
 1   Title         55169 non-null  object
 2   Location      55169 non-null  object
 3   Company       55169 non-null  object
 4   ContractType  55169 non-null  object
 5   ContractTime  55169 non-null  object
 6   Category      55169 non-null  object
 7   Salary        55169 non-null  float64
 8   OpenDate      55169 non-null  object
 9   CloseDate     55169 non-null  object
 10  Source        55169 non-null  object
dtypes: float64(1), int64(1), object(9)
memory usage: 4.6+ MB
```

```
df1.head()
```

| | Id | Title | Location | Company | ContractType | ContractTime | Category | Salary |
|---|---|---|---|---|---|---|---|---|
| **0** | 12612628 | Engineering Systems Analyst | DORKING | GREGORY MARTIN INTERNATIONAL | FULLTIME | PERMANENT | Engineering Jobs | 25000.0 |
| **1** | 12612830 | Stress Engineer Glasgow | GLASGOW | GREGORY MARTIN INTERNATIONAL | FULLTIME | PERMANENT | Engineering Jobs | 30000.0 |
| **2** | 12612844 | Modelling and simulation analyst | HAMPSHIRE | GREGORY MARTIN INTERNATIONAL | FULLTIME | PERMANENT | Engineering Jobs | 30000.0 |
| | | Engineering Systems | | GREGORY | | | | |

▾ As we can see above, The name of few, columns are different in both data set, The columns, Id and source are missing in dataset2. So to merge the two dataset we need to get a global Key, which will act as an ID. The Salary, column is only per month in dataset2.

The Fraction in dataset2 is Contract Type in dataset1, With only three non null Values.

The ContractType in dataset2, is ContractTime in dataset1.

Organisation in dataset2 is company in dataset1.

Opening and Closing in dataset2, are actually OpenDate and CloseDate in dataset1.

Job title is named as only Title in dataset1.

Lets change the column names of dataset2, to get more nifromity in column names and add the column with Annual Salary before merging.

We will also remove any special characters from dataste2 and convert everything in upper case in both data set for unifomity.

```
df2.rename(columns = {'Job Title':'Title2'}, inplace = True)
df2.rename(columns = {'Opening':'OpenDate2'}, inplace = True)
df2.rename(columns = {'Closing':'CloseDate2'}, inplace = True)
df2.rename(columns = {'Organisation':'Company2'}, inplace = True)
df2.rename(columns = {'Fraction':'ContractType2'}, inplace = True)
df2.rename(columns = {'Contract Type':'ContractTime2'}, inplace = True)
df2.rename(columns = {'Location':'Location2'}, inplace = True)
df2.rename(columns = {'Category':'Category2'}, inplace = True)
```

```
df2['Salary2'] = (df2['Salary per month']*12).astype(float)
```

```
df2.head()
```

| | OpenDate2 | CloseDate2 | Title2 | Company2 | Location2 | Category2 | Salary per month | ContractType2 | Contrac |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 2013-10-06 00:00:00 | 2013-12-05 00:00:00 | Aviation loans administration | cer Financial | London | Finance and Accounting | 2800 | NaN | ( |
| **1** | 2012-10-03 12:00:00 | 2012-11-02 12:00:00 | Payroll Analyst City upto **** , **** | LMA Recruitment Ltd | London | Finance and Accounting | 2917 | NaN | Per |
| **2** | 2012-01-01 00:00:00 | 2012-01-31 00:00:00 | Investment Team Assistant for leading Private ... | Austin Andrew Ltd | London | Finance and Accounting | 3750 | NaN | Per |

```
# We need the Column with Annual Salary in our final output
# So as we got a column already in dataset2 with Annual Salary Lets drop the Salary per month Column
df2 = df2.drop(['Salary per month'], axis = 1)
```

```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 334 entries, 0 to 333
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   OpenDate2      334 non-null    object
 1   CloseDate2     334 non-null    object
 2   Title2         334 non-null    object
 3   Company2       334 non-null    object
 4   Location2      334 non-null    object
 5   Category2      334 non-null    object
 6   ContractType2  3 non-null      object
 7   ContractTime2  327 non-null    object
 8   Salary2        334 non-null    float64
dtypes: float64(1), object(8)
memory usage: 23.6+ KB
```

## Checking and Removing Special characters

```
df2['Title2'].unique()
```

```
array(['Aviation loans administration',
       'Payroll Analyst  City upto **** , ****',
       'Investment Team Assistant for leading Private Equity firm',
       'SWAPS COLLATERAL CONTROL OFFICER', 'Loans Administration  Temp',
       'Oversight and Compliance Manager  Global Custody Operations',
       'ALM Actuary', 'SUPPORT ANALYST  Front Office',
       'Forex Sales Manager', 'Management Accountant  Commodity Trading',
       'CFD Sales Trader',
       'Compliance Monitoring Asset Management London',
       'Credit Trade Finance Manager', 'Sales Support Executive',
```

```
       'Accounts Receivable / Sales ledger Assistant',
       'Internal Resourcing Coordinator', 'Cheque Processing Clerk',
       'Cashier  Fluent Mandarin',
       'Regulatory control change project manager', 'VP, Regulatory Risk',
       'Transaction Reporting Specialist',
       'Compliance Monitoring Manager  Banking',
       'Model Validation Quantitative Analyst', 'VP Operational Risk',
       'Risk Review Compliance Officer', 'Middle Office Analyst FTC',
       'Financial Controller / Group Accountant', 'Trade Control VP',
       'Business Partner  Treasury', 'Business Analyst  Change',
       'Senior Accountant  Group Reporting',
       'Product Control Analyst  Oil', 'Portfolio / Performance Analyst',
       'Risk Controller', 'Project Manager  Private Client',
       'Ops Process Management Analyst',
       'Transactional Management Team Leader',
       'Oversight and Compliance Manager', 'Fund Accountant  London',
       'Voluntary Corporate Actions', 'Loans Documentation Analyst',
       'Regulatory Accountant',
       'DO YOU HAVE 2 YEARS CURRENT BUY SIDE OPERATIONS EXPERIENCE?',
       'Trade Compliance Analyst', 'Compliance Assistant',
       'Shares and Market Options Sales Consultant', 'Operations Analyst',
       'Fund Accountant  Hedge Fund', 'Management Support Executive',
       'RFP Writer', 'Senior Equity Settlements',
       'Assistant Accountant  Business Partnering',
       'Treasury Dealer FX Dealer', 'Billing Coordinator',
       'Settlements Adminstrator  Investment Management',
       'Middle Office Support', 'Equity Research Sales  Private Clients',
       'Banking Cashier', 'Portfolio Finance Planning Manager',
       'Senior Audit Investigations Manager',
       'Treasury Risk Controller : Middle Office Compliance : Oil Gas Co',
       'Business Change Analyst / Project Manager',
       'CREDIT ANALYST  up to ****k  London',
       'Assistant Vice President  Auditor',
       'Senior Business Analyst / Business Analyst  Claims',
       'Business Strategy Analyst', 'Insurance  Financial Accountant',
       'Committee Secretary  London  ****k',
       'Royalties Manager / FP A Manager',
       'International Communications Manager',
       'Operational Risk and Control Manager',
       'Sourcing Manager  Professional Fees',
       'Mortgage Protection AdvisorEstate Agent',
       'PMO / Senior PMO Analyst',
       'SECONDARY LOANS / SYNDICATIONS ADMIN / SETTLEMENTS',
       'Management Account',
       'Real Estate Finance  Asset Management Analyst',
       'Chartered IFA Farnham Up to **** , **** (********k OTE)',
       'Business Change Senior Project Manager  East Midlands',
       'KYC Remediation Analyst AML  JMLSG  OnBoarding'
```

```
df2['Company2'].unique()
```

```
array(['cer Financial', 'LMA Recruitment Ltd', 'Austin Andrew Ltd',
       'Brian Durham Recruitment Services Limited', 'Citifocus Limited',
       'Reed Insurance', 'Newside Consulting Limited',
       'MDM Consultants Limited', 'Eximius Operations Ltd',
       'Next Employment Limited', 'Capco', 'Venn Group Limited',
       'Prime Personnel Services Ltd', 'C.K.R. Recruitment Limited',
       'Black Swan Associates Limited', 'Oliver James Associates Limited',
       'Walker Hamill Ltd', 'Badenoch Clark Ltd',
       'Ambition Europe Limited', 'Procura Ltd', 'Harrison Holgate',
       'Insight Professional Solutions Ltd', 'Twenty Recruitment Ltd',
       'Sharon Gay Associates Limited',
       'Randstad Financial Professional Ltd',
       'Morgan McKinley Group Limited', 'Charles Levick Limited',
       'Hill Newton Recruitment Limited', 'Kennedy Pearce Consulting Ltd',
       'Maldon Partners Ltd', 'Saul Partners', 'Hays Financial Markets',
       'Dante Recruitment Ltd', 'Bruin Financial Limited',
       'Daniel Recruitment Ltd', 'Mayford James Limited',
       'The Ocean Partnership Limited', 'Austin Benn',
       'Rebecca Poulter Recruitment',
       'Argyll Scott International Limited', 'Alexander Lloyd',
       'Eaglecliff Limited', 'Livingston Edwards Ltd',
       'Incite Solutions Ltd', 'Experis Limited', 'ASK Recruitment',
       'Idex Consulting LLP', 'EDIT Professional Ltd',
       'James Associates Recruitment Ltd', 'Apache Associates Ltd',
       'Taylor James Resourcing Limited', 'Wavelength Recruitment',
       'Robert Half', 'Insight Recruitment Solutions Ltd',
       'PeopleGenius Ltd', 'Brooker Associates Limited',
       'Montpellier Resourcing Ltd', 'MW Recruitment Ltd',
       'REC Solutions Ltd', 'MCGREGOR BOYALL ASSOCIATES LIMITED',
       'Goodman Masson Recruitment Services Ltd',
       'Balanced People Limited', 'Lesley Ray',
       'Astbury Marsden Partners Ltd', 'Sublime Resourcing Limited',
       'People First Recruitment Limited', 'Clark James Ltd',
       'Concilium Finance Limited', 'Paritas Recruitment Limited',
       'High Finance Group', 'Campbell James Limited',
       'Cahill Personnel Ltd', 'Jefferson Tiley Ltd',
       'Jigsaw Recruiting Ltd', 'The Oakland Partnership Limited',
       'TalentFlow Ltd', 'Reed Finance', 'Rees Draper Wright Ltd',
       'Plumstead Bond Limited', 'Blayney Personnel Limited',
```

```
       'Momenta Group Limited', 'Newedge',
       'Colyer Dodd and Company Limited', 'Selby Jennings Ltd',
       'Matador Recruitment', 'Brightpool',
       'Monument Financial Recruitment Ltd', 'Marks Sattin  Ltd',
       'Employment Specialists Ltd', 'IPS Group Ltd', 'Hewitson Walker',
       'PBR Construction LTD', 'Generation Resourcing Limited',
       'MW Appointments Ltd', 'Carnegie Consulting Ltd',
       'JHA Recruitment Consultancy', 'Reed Purchasing',
       'City East Recruitment Limited', 'Arthur Financial Ltd',
       'Albany Beck Consulting Ltd'], dtype=object)
```

```
df2['Location2'].unique()
```

```
array(['London', 'City', 'The City', 'South East England', 'West End',
       'UK', 'West London', 'Berkshire', 'Central London', 'Croydon',
       'Farnham', 'East Midlands', 'Crawley', 'Cheltenham', 'Dorking',
       'Redhill', 'Hertfordshire', 'Birmingham', 'Hampshire', 'Fareham',
       'Chelmsford', 'South West London', 'North West England', 'Reading',
       'Bristol', 'Peterborough', 'Southampton', 'Buckinghamshire',
       'Basingstoke', 'South East London', 'West Sussex', 'Guildford',
       'North London', 'Bath'], dtype=object)
```

```
df2['Category2'].unique()
```

```
array(['Finance and Accounting', 'PR, Advertising and Marketing',
       'Information Technology'], dtype=object)
```

```
df2['Title2'].str.contains(r'([^A-Za-z]|\s+|\W+|\.|\-|\_)').any()
```

```
/Users/Akshata/opt/anaconda3/lib/python3.8/site-packages/pandas/core/strings.py:2001: UserWarning: This pattern has match groups. T
  return func(self, *args, **kwargs)
True
```

```
df2['Location2'].str.contains(r'([^A-Za-z]|\s+|\W+|\.|\-|\_)').any()
```

```
True
```

```
df2['Category2'].str.contains(r'([^A-Za-z]|\s+|\W+|\.|\-|\_)').any()
```

```
True
```

```
df2['Company2'].str.contains(r'([^A-Za-z]|\s+|\W+|\.|\-|\_)').any()
```

```
True
```

```
# Normalising Title and Removing special characters.
def removeSpecialChar(Title):

  # normalize to upper case letters
   Title= Title.upper()
   # remove all characters except alphabets
   Title = re.sub(r'([^A-Z])', ' ', Title)
   # replace multiple spaces with a single space, also trim spaces on both side
   Title = re.sub( '\s+', ' ', Title).strip()
   return Title
df2['Title2'] = df2.Title2.apply(lambda x: removeSpecialChar(x))
```

```
df2['Title2'].unique()
```

```
array(['AVIATION LOANS ADMINISTRATION', 'PAYROLL ANALYST CITY UPTO',
       'INVESTMENT TEAM ASSISTANT FOR LEADING PRIVATE EQUITY FIRM',
       'SWAPS COLLATERAL CONTROL OFFICER', 'LOANS ADMINISTRATION TEMP',
       'OVERSIGHT AND COMPLIANCE MANAGER GLOBAL CUSTODY OPERATIONS',
       'ALM ACTUARY', 'SUPPORT ANALYST FRONT OFFICE',
       'FOREX SALES MANAGER', 'MANAGEMENT ACCOUNTANT COMMODITY TRADING',
       'CFD SALES TRADER',
       'COMPLIANCE MONITORING ASSET MANAGEMENT LONDON',
       'CREDIT TRADE FINANCE MANAGER', 'SALES SUPPORT EXECUTIVE',
       'ACCOUNTS RECEIVABLE SALES LEDGER ASSISTANT',
       'INTERNAL RESOURCING COORDINATOR', 'CHEQUE PROCESSING CLERK',
       'CASHIER FLUENT MANDARIN',
       'REGULATORY CONTROL CHANGE PROJECT MANAGER', 'VP REGULATORY RISK',
       'TRANSACTION REPORTING SPECIALIST',
       'COMPLIANCE MONITORING MANAGER BANKING',
       'MODEL VALIDATION QUANTITATIVE ANALYST', 'VP OPERATIONAL RISK',
       'RISK REVIEW COMPLIANCE OFFICER', 'MIDDLE OFFICE ANALYST FTC',
       'FINANCIAL CONTROLLER GROUP ACCOUNTANT', 'TRADE CONTROL VP',
       'BUSINESS PARTNER TREASURY', 'BUSINESS ANALYST CHANGE',
       'SENIOR ACCOUNTANT GROUP REPORTING', 'PRODUCT CONTROL ANALYST OIL',
       'PORTFOLIO PERFORMANCE ANALYST', 'RISK CONTROLLER',
```

```
            'PROJECT MANAGER PRIVATE CLIENT', 'OPS PROCESS MANAGEMENT ANALYST',
            'TRANSACTIONAL MANAGEMENT TEAM LEADER',
            'OVERSIGHT AND COMPLIANCE MANAGER', 'FUND ACCOUNTANT LONDON',
            'VOLUNTARY CORPORATE ACTIONS', 'LOANS DOCUMENTATION ANALYST',
            'REGULATORY ACCOUNTANT',
            'DO YOU HAVE YEARS CURRENT BUY SIDE OPERATIONS EXPERIENCE',
            'TRADE COMPLIANCE ANALYST', 'COMPLIANCE ASSISTANT',
            'SHARES AND MARKET OPTIONS SALES CONSULTANT', 'OPERATIONS ANALYST',
            'FUND ACCOUNTANT HEDGE FUND', 'MANAGEMENT SUPPORT EXECUTIVE',
            'RFP WRITER', 'SENIOR EQUITY SETTLEMENTS',
            'ASSISTANT ACCOUNTANT BUSINESS PARTNERING',
            'TREASURY DEALER FX DEALER', 'BILLING COORDINATOR',
            'SETTLEMENTS ADMINSTRATOR INVESTMENT MANAGEMENT',
            'MIDDLE OFFICE SUPPORT', 'EQUITY RESEARCH SALES PRIVATE CLIENTS',
            'BANKING CASHIER', 'PORTFOLIO FINANCE PLANNING MANAGER',
            'SENIOR AUDIT INVESTIGATIONS MANAGER',
            'TREASURY RISK CONTROLLER MIDDLE OFFICE COMPLIANCE OIL GAS CO',
            'BUSINESS CHANGE ANALYST PROJECT MANAGER',
            'CREDIT ANALYST UP TO K LONDON',
            'ASSISTANT VICE PRESIDENT AUDITOR',
            'SENIOR BUSINESS ANALYST BUSINESS ANALYST CLAIMS',
            'BUSINESS STRATEGY ANALYST', 'INSURANCE FINANCIAL ACCOUNTANT',
            'COMMITTEE SECRETARY LONDON K', 'ROYALTIES MANAGER FP A MANAGER',
            'INTERNATIONAL COMMUNICATIONS MANAGER',
            'OPERATIONAL RISK AND CONTROL MANAGER',
            'SOURCING MANAGER PROFESSIONAL FEES',
            'MORTGAGE PROTECTION ADVISORESTATE AGENT',
            'PMO SENIOR PMO ANALYST',
            'SECONDARY LOANS SYNDICATIONS ADMIN SETTLEMENTS',
            'MANAGEMENT ACCOUNT',
            'REAL ESTATE FINANCE ASSET MANAGEMENT ANALYST',
            'CHARTERED IFA FARNHAM UP TO K OTE',
            'BUSINESS CHANGE SENIOR PROJECT MANAGER EAST MIDLANDS',
            'KYC REMEDIATION ANALYST AML JMLSG ONBOARDING',
            'INVESTMENT MANAGER', 'SENIOR HR CONSULTANT',
            'FIXED INCOME TRADE SUPPORT INVESTMENT MANAGEMENT',
            'CORPORATE RETAIL BANKING PMO PERM',
```

```
# Normalising and removing Special Characters from Category.
def removeSpecialChar(Category):

       # normalize to upper case letters
        Category = Category.upper()
         # change " & " back to " AND "
        Category = Category.replace("AND", "&")
        # remove all characters except alphabets
        Category = re.sub(r'([^A-Z | ^&] )', '', Category)
        # replace IT to InformationTechnology
        #Category = Category.replace("[^IT]", "INFORMATIONTECHNOLOGY")
        # replace multiple spaces with a single space, also trim spaces on both side
        Category = re.sub( '\s+', ' ', Category).strip()
        return Category
df2['Category2'] = df2.Category2.apply(lambda x: removeSpecialChar(x))


df2['Category2'].unique()
```

```
array(['FINANCE & ACCOUNTING', 'PRADVERTISING & MARKETING',
       'INFORMATION TECHNOLOGY'], dtype=object)
```

```
# Normalising Location
def removeSpecialChar(Location):

       # normalize to upper case letters
        Location = Location.upper()
        # remove all characters except alphabets
        Location = re.sub(r'([^A-Z])', '', Location)
        #We can see above we have TheCity and City which is same Location, So lets replace TheCity to only City.
        #remove The from Thecity
        Location = re.sub(r'(^THE)','',Location)
        # replace multiple spaces with a single space, also trim spaces on both side
        Location = re.sub( '\s+', ' ', Location).strip()
        return Location
df2['Location2'] = df2.Location2.apply(lambda x: removeSpecialChar(x))


df2['Location2'].unique()
```

```
array(['LONDON', 'CITY', 'SOUTHEASTENGLAND', 'WESTEND', 'UK',
       'WESTLONDON', 'BERKSHIRE', 'CENTRALLONDON', 'CROYDON', 'FARNHAM',
       'EASTMIDLANDS', 'CRAWLEY', 'CHELTENHAM', 'DORKING', 'REDHILL',
       'HERTFORDSHIRE', 'BIRMINGHAM', 'HAMPSHIRE', 'FAREHAM',
       'CHELMSFORD', 'SOUTHWESTLONDON', 'NORTHWESTENGLAND', 'READING',
       'BRISTOL', 'PETERBOROUGH', 'SOUTHAMPTON', 'BUCKINGHAMSHIRE',
       'BASINGSTOKE', 'SOUTHEASTLONDON', 'WESTSUSSEX', 'GUILDFORD',
       'NORTHLONDON', 'BATH'], dtype=object)
```

```
# Normalising Company
def removeSpecialChar(Organisation):

        # normalize to upper case letters
        Organisation = Organisation.upper()
        # replace LIMITED with LTD
        Organisation = Organisation.replace("LIMITED", "LTD")
        # remove all special characters except space and dot
        Organisation = re.sub(r'([^\w\s\.]|_)', '', Organisation)
        # change "AND" back to " & "
        Organisation = Organisation.replace("AND", "&")
        # replace multiple spaces and trim spaces
        Organisation = re.sub( '\s+', ' ', Organisation).strip()
        return Organisation

df2['Company2'] = df2.Company2.apply(lambda x: removeSpecialChar(x))


df2['Company2'].unique()
```

```
array(['CER FINANCIAL', 'LMA RECRUITMENT LTD', 'AUSTIN &REW LTD',
       'BRIAN DURHAM RECRUITMENT SERVICES LTD', 'CITIFOCUS LTD',
       'REED INSURANCE', 'NEWSIDE CONSULTING LTD', 'MDM CONSULTANTS LTD',
       'EXIMIUS OPERATIONS LTD', 'NEXT EMPLOYMENT LTD', 'CAPCO',
       'VENN GROUP LTD', 'PRIME PERSONNEL SERVICES LTD',
       'C.K.R. RECRUITMENT LTD', 'BLACK SWAN ASSOCIATES LTD',
       'OLIVER JAMES ASSOCIATES LTD', 'WALKER HAMILL LTD',
       'BADENOCH CLARK LTD', 'AMBITION EUROPE LTD', 'PROCURA LTD',
       'HARRISON HOLGATE', 'INSIGHT PROFESSIONAL SOLUTIONS LTD',
       'TWENTY RECRUITMENT LTD', 'SHARON GAY ASSOCIATES LTD',
       'R&STAD FINANCIAL PROFESSIONAL LTD', 'MORGAN MCKINLEY GROUP LTD',
       'CHARLES LEVICK LTD', 'HILL NEWTON RECRUITMENT LTD',
       'KENNEDY PEARCE CONSULTING LTD', 'MALDON PARTNERS LTD',
       'SAUL PARTNERS', 'HAYS FINANCIAL MARKETS', 'DANTE RECRUITMENT LTD',
       'BRUIN FINANCIAL LTD', 'DANIEL RECRUITMENT LTD',
       'MAYFORD JAMES LTD', 'THE OCEAN PARTNERSHIP LTD', 'AUSTIN BENN',
       'REBECCA POULTER RECRUITMENT', 'ARGYLL SCOTT INTERNATIONAL LTD',
       'ALEX&ER LLOYD', 'EAGLECLIFF LTD', 'LIVINGSTON EDWARDS LTD',
       'INCITE SOLUTIONS LTD', 'EXPERIS LTD', 'ASK RECRUITMENT',
       'IDEX CONSULTING LLP', 'EDIT PROFESSIONAL LTD',
       'JAMES ASSOCIATES RECRUITMENT LTD', 'APACHE ASSOCIATES LTD',
       'TAYLOR JAMES RESOURCING LTD', 'WAVELENGTH RECRUITMENT',
       'ROBERT HALF', 'INSIGHT RECRUITMENT SOLUTIONS LTD',
       'PEOPLEGENIUS LTD', 'BROOKER ASSOCIATES LTD',
       'MONTPELLIER RESOURCING LTD', 'MW RECRUITMENT LTD',
       'REC SOLUTIONS LTD', 'MCGREGOR BOYALL ASSOCIATES LTD',
       'GOODMAN MASSON RECRUITMENT SERVICES LTD', 'BALANCED PEOPLE LTD',
       'LESLEY RAY', 'ASTBURY MARSDEN PARTNERS LTD',
       'SUBLIME RESOURCING LTD', 'PEOPLE FIRST RECRUITMENT LTD',
       'CLARK JAMES LTD', 'CONCILIUM FINANCE LTD',
       'PARITAS RECRUITMENT LTD', 'HIGH FINANCE GROUP',
       'CAMPBELL JAMES LTD', 'CAHILL PERSONNEL LTD',
       'JEFFERSON TILEY LTD', 'JIGSAW RECRUITING LTD',
       'THE OAKL& PARTNERSHIP LTD', 'TALENTFLOW LTD', 'REED FINANCE',
       'REES DRAPER WRIGHT LTD', 'PLUMSTEAD BOND LTD',
       'BLAYNEY PERSONNEL LTD', 'MOMENTA GROUP LTD', 'NEWEDGE',
       'COLYER DODD & COMPANY LTD', 'SELBY JENNINGS LTD',
       'MATADOR RECRUITMENT', 'BRIGHTPOOL',
       'MONUMENT FINANCIAL RECRUITMENT LTD', 'MARKS SATTIN LTD',
       'EMPLOYMENT SPECIALISTS LTD', 'IPS GROUP LTD', 'HEWITSON WALKER',
       'PBR CONSTRUCTION LTD', 'GENERATION RESOURCING LTD',
       'MW APPOINTMENTS LTD', 'CARNEGIE CONSULTING LTD',
       'JHA RECRUITMENT CONSULTANCY', 'REED PURCHASING',
       'CITY EAST RECRUITMENT LTD', 'ARTHUR FINANCIAL LTD',
       'ALBANY BECK CONSULTING LTD'], dtype=object)
```

## Lets check dataset1 unique Values.

```
df1['Title'].unique()
```

```
array(['Engineering Systems Analyst', 'Stress Engineer Glasgow',
       'Modelling and simulation analyst', ...,
       'Curriculum Leader  Mathematics',
       'TEACHER OF BUSINESS STUDIES AND LAW',
       'Barclays Future Leaders Development Programmes '], dtype=object)
```

```
### As we can see dataset1 Title Values are not normalised
def removeSpecialChar(Title):

       # normalize to upper case letters
        Title= Title.upper()

        # remove all characters except alphabets
```

```
        Title = re.sub(r'([^A-Z])', ' ', Title)
        # replace multiple spaces with a single space, also trim spaces on both side
        Title = re.sub( '\s+', ' ', Title).strip()
        return Title
df1['Title'] = df1.Title.apply(lambda x: removeSpecialChar(x))
```

```
df1['Title'].unique()
```

```
array(['ENGINEERING SYSTEMS ANALYST', 'STRESS ENGINEER GLASGOW',
       'MODELLING AND SIMULATION ANALYST', ...,
       'SENIOR PROJECT MANAGER EVENT EXHIBITION AGENCY',
       'CURRICULUM LEADER MATHEMATICS',
       'TEACHER OF BUSINESS STUDIES AND LAW'], dtype=object)
```

```
df1['Category'].unique()
```

```
array(['Engineering Jobs', 'Accounting & Finance Jobs',
       'Healthcare & Nursing Jobs', 'Hospitality & Catering Jobs',
       'IT Jobs', 'Sales Jobs', 'Teaching Jobs',
       'PR, Advertising & Marketing Jobs'], dtype=object)
```

```
# Normalising and removing Special Characters from Category.
def removeSpecialChar(Category):

       # normalize to upper case letters
        Category = Category.upper()
         # change " & " back to " AND "
        Category = Category.replace("AND", "&")
        # remove all characters except alphabets
        Category = re.sub(r'([^A-Z | ^&] )', '', Category)
        # remove Jobs
        Category = re.sub( r'(JOBS)$', ',', Category)
        # replace multiple spaces with a single space, also trim spaces on both side and remove and commas.
        Category = re.sub( '\s+|,', ' ', Category).strip()
        return Category
df1['Category'] = df1.Category.apply(lambda x: removeSpecialChar(x))
```

```
df1['Category'].unique()
```

```
array(['ENGINEERING', 'ACCOUNTING & FINANCE', 'HEALTHCARE & NURSING',
       'HOSPITALITY & CATERING', 'IT', 'SALES', 'TEACHING',
       'PRADVERTISING & MARKETING'], dtype=object)
```

```
# Compare with df2 Category
df2['Category2'].unique()
```

```
array(['FINANCE & ACCOUNTING', 'PRADVERTISING & MARKETING',
       'INFORMATION TECHNOLOGY'], dtype=object)
```

As we can see, df2 has FINANCE & ACCOUNTING but df1 has ACCOUNTING & FINANCE, and IT in df1 as Information Technology in df2.

Lets get the uniform Values for each Category, So lets change the values in df2.

```
# replace InformationTechnology to IT and Finance & Accounting to Accounting & Finance
df2['Category2'] = df2.Category2.replace("INFORMATION TECHNOLOGY", "IT")

df2['Category2'] = df2.Category2.replace("FINANCE & ACCOUNTING", "ACCOUNTING & FINANCE")
```

```
df2['Category2'].unique()
```

```
array(['ACCOUNTING & FINANCE', 'PRADVERTISING & MARKETING', 'IT'],
      dtype=object)
```

Both the dataset donot have any common unique Id. So it is very essential to create a unique Identifier to merge the dataframe accurately. This an be done by creating, a unique ID using the different attributes and concatenating them. If we use, dates or combination of close and Open Dates, as unique ID, There are still chances of getting duplicates. So lets try to create unique ID using Category, Location and Company

```
df2['globalId'] = df2['Category2']+"_"+df2['Location2'] + "_" + df2['Company2']
```

```
# Lets check if any duplicates in df2 for global Id
df2['globalId'].duplicated().any()
```

```
True
```

If we create the global key using Category, Location and Company, we still getting the deuplicate Values, which can cause issues and inaccurate merging of dataset. Lets try add Title and Check if we can create a global key in df2 with no duplicate values.

```
# Lets add Title as this will reduce tha chance of duplicate Values in global Id.
df2['globalId'] = df2['Title2']+"_"+df2['Category2']+"_"+df2['Location2'] + "_" + df2['Company2']
```

```
# Lets check if any duplicates in df2 for global Id
df2['globalId'].duplicated().any()
```

```
False
```

```
# Creating global key in df1 as well, so we can merge two dataset on this globalId.
df1['globalId'] = df1['Title']+"_"+df1['Category']+"_"+df1['Location'] + "_" + df1['Company']
```

## The Global Key

Write down the identified global key:

**| Title | Category| Location | Company |**

As now We have got the global Id we can merge both the dataset, using outer merge. We are selecting Outer merge as we want all the data from dataset1 and dataset 2. Because, Id column is missing in dataset2, so we could have lots of useful data that is not in dataset1 but in dataset2 and viceversa.

Lets first common data present in both the dataset and check if actually globalId is mergingis correctly or not.

```
# Merge data using inner to check, how many Values are present in both dataset.
df3 = pd.merge(df1, df2, on=['globalId'], how='inner')
```

```
print(df3.shape)
```

```
(32, 21)
```

```
# Display all columns and rows, to make sure that the dataset is merging accurately and our global key is correct.
pd.set_option("display.max_rows", None, "display.max_columns", None)
df3
```

| | Id | Title | Location | Company | ContractType | ContractTime | Category |
|---|---|---|---|---|---|---|---|
| **0** | 60685252 | LENDING MANAGER SENIOR MANAGER COMMERCIAL REAL... | LONDON | HAYS FINANCIAL MARKETS | FULLTIME | PERMANENT | ACCOUNTING & FINANCE |
| **1** | 62008199 | SUPPORT ANALYST FRONT OFFICE | LONDON | NEWSIDE CONSULTING LTD | FULLTIME | PERMANENT | ACCOUNTING & FINANCE |
| **2** | 66426264 | QUANTITATIVE ANALYTICS MANAGER | LONDON | KENNEDY PEARCE CONSULTING LTD | FULLTIME | PERMANENT | ACCOUNTING & FINANCE |
| **3** | 66584157 | CASH ORIGINATOR | LONDON | GOODMAN MASSON RECRUITMENT SERVICES LTD | FULLTIME | PERMANENT | ACCOUNTING & FINANCE |
| **4** | 67763247 | BANKING CUSTOMER SERVICES PAYMENTS CLERK | CITY | LMA RECRUITMENT LTD | FULLTIME | CONTRACT | ACCOUNTING & FINANCE |
| **5** | 67946646 | MODELLING ANALYSTLLOYDS UNDERWRITING QUANTITAT... | CITY | TAYLOR JAMES RESOURCING LTD | FULLTIME | PERMANENT | ACCOUNTING & FINANCE |
| **6** | 68060099 | INTERNAL AUDIT COMMODITIES INVESTMENT BANKING ... | LONDON | MORGAN MCKINLEY GROUP LTD | FULLTIME | PERMANENT | ACCOUNTING & FINANCE |
| **7** | 68061837 | BUSINESS INTELLIGENCE DATA ANALYTIC MANAGERS | UK | OLIVER JAMES ASSOCIATES LTD | FULLTIME | PERMANENT | ACCOUNTING & FINANCE |
| **8** | 68089076 | RISK CONTROLLER | LONDON | CER FINANCIAL | FULLTIME | PERMANENT | ACCOUNTING & FINANCE |
| **9** | 68218723 | ASSISTANT ACCOUNTANT FINANCIAL SERVICES EXPERI... | LONDON | MALDON PARTNERS LTD | FULLTIME | PERMANENT | ACCOUNTING & FINANCE |
| **10** | 68361517 | CREDIT RISK REPORTING MANAGER ASSISTANT MANAGER | LONDON | CER FINANCIAL | FULLTIME | PERMANENT | ACCOUNTING & FINANCE |
| **11** | 68509785 | MOTOR CLAIMS COMPLAINTS TEAM LEADER | UK | CLARK JAMES LTD | FULLTIME | PERMANENT | ACCOUNTING & FINANCE |
| **12** | 68674061 | INFORMATION SECURITY MANAGER FINANCIAL SERVICE... | LONDON | OLIVER JAMES ASSOCIATES LTD | FULLTIME | PERMANENT | IT |
| **13** | 68693540 | SENIOR AUDIT MANAGER MARKET MODEL RISK | LONDON | PARITAS RECRUITMENT LTD | FULLTIME | PERMANENT | ACCOUNTING & FINANCE |
| **14** | 68840315 | ACCOUNTS PAYABLE COMPLIANCE ADMINISTRATOR | LONDON | PRIME PERSONNEL SERVICES LTD | FULLTIME | PERMANENT | ACCOUNTING & FINANCE |
| **15** | 69022388 | SALES CONSULTANT BIRMINGHAM UP TO BONUS | BIRMINGHAM | IDEX CONSULTING LLP | FULLTIME | PERMANENT | ACCOUNTING & FINANCE |
| **16** | 69042777 | SENIOR COMPLIANCE ASSOCIATE | LONDON | BLACK SWAN ASSOCIATES LTD | FULLTIME | PERMANENT | ACCOUNTING & FINANCE |
| **17** | 69266259 | TRADE STRUCTURED FINANCE MANAGER | LONDON | LMA RECRUITMENT LTD | FULLTIME | PERMANENT | ACCOUNTING & FINANCE |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | COMMODITY TRADER | | LTD | | | |
| **18** | 69538758 | MANDARIN SPEAKING INTERNAL AUDITOR | LONDON | PEOPLE FIRST RECRUITMENT LTD | FULLTIME | PERMANENT | ACCOUNTING & FINANCE |
| **19** | 69687363 | CREDIT ASSISTANT MANAGER PROPERTY DIVISION | LONDON | MONUMENT FINANCIAL RECRUITMENT LTD | FULLTIME | PERMANENT | ACCOUNTING & FINANCE |
| **20** | 69992853 | COMPLIANCE SUPERVISOR | LONDON | MW RECRUITMENT LTD | PARTTIME | PERMANENT | ACCOUNTING & FINANCE |
| **21** | 70099824 | SENIOR OPERATIONAL RISK ANALYST AVP | LONDON | C.K.R. RECRUITMENT LTD | FULLTIME | PERMANENT | ACCOUNTING & FINANCE |
| **22** | 70770491 | MODEL VALIDATION QUANTITATIVE ANALYST | LONDON | LMA RECRUITMENT LTD | FULLTIME | PERMANENT | ACCOUNTING & FINANCE |
| **23** | 70783605 | FIXED INCOME TRADE SUPPORT INVESTMENT MANAGEMENT | WESTEND | AUSTIN BENN | FULLTIME | PERMANENT | ACCOUNTING & FINANCE |
| **24** | 71406069 | CASHIER FLUENT MANDARIN | WESTEND | PRIME PERSONNEL SERVICES LTD | FULLTIME | CONTRACT | ACCOUNTING & FINANCE |
| **25** | 71430259 | KYC REMEDIATION | LONDON | THE OCEAN PARTNERSHIP LTD | FULLTIME | CONTRACT | ACCOUNTING & FINANCE |

We can see that our global Id looks correct as the data merge we can see excatly same rows for both dataset. Now lets do Outer merge to get our final data set.

ASSISTANT TO JIGSAW

```
# Merging data set using outer join
final_df = pd.merge(df2,df1,
               on = 'globalId', how = 'outer')
```

SENIOR DATA

data checks

...

```
print(final_df.shape)
```

```
(55471, 21)
```

```
final_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 55471 entries, 0 to 55470
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   OpenDate2      334 non-null    object 
 1   CloseDate2     334 non-null    object 
 2   Title2         334 non-null    object 
 3   Company2       334 non-null    object 
 4   Location2      334 non-null    object 
 5   Category2      334 non-null    object 
 6   ContractType2  3 non-null      object 
 7   ContractTime2  327 non-null    object 
 8   Salary2        334 non-null    float64
 9   globalId       55471 non-null  object 
 10  Id             55169 non-null  float64
 11  Title          55169 non-null  object 
 12  Location       55169 non-null  object 
 13  Company        55169 non-null  object 
 14  ContractType   55169 non-null  object 
 15  ContractTime   55169 non-null  object 
 16  Category       55169 non-null  object 
 17  Salary         55169 non-null  float64
 18  OpenDate       55169 non-null  object 
 19  CloseDate      55169 non-null  object 
 20  Source         55169 non-null  object 
```

```
dtypes: float64(3), object(18)
memory usage: 9.3+ MB
```

```
final_df.head()
```

| | OpenDate2 | CloseDate2 | Title2 | Company2 | Location2 | Category2 | ContractType2 | Cont |
|---|---|---|---|---|---|---|---|---|
| **0** | 2013-10-06 00:00:00 | 2013-12-05 00:00:00 | AVIATION LOANS ADMINISTRATION | CER FINANCIAL | LONDON | ACCOUNTING & FINANCE | NaN | |
| **1** | 2012-10-03 12:00:00 | 2012-11-02 12:00:00 | PAYROLL ANALYST CITY UPTO | LMA RECRUITMENT LTD | LONDON | ACCOUNTING & FINANCE | NaN | |
| **2** | 2012-01-01 00:00:00 | 2012-01-31 00:00:00 | INVESTMENT TEAM ASSISTANT FOR LEADING PRIVATE ... | AUSTIN &REW LTD | LONDON | ACCOUNTING & FINANCE | NaN | |
| **3** | 2012-10-14 00:00:00 | 2012-11-13 00:00:00 | SWAPS COLLATERAL CONTROL OFFICER | BRIAN DURHAM RECRUITMENT SERVICES LTD | CITY | ACCOUNTING & FINANCE | NaN | |
| **4** | 2012-11-17 12:00:00 | 2013-01-16 12:00:00 | LOANS ADMINISTRATION TEMP | CER FINANCIAL | LONDON | ACCOUNTING & FINANCE | NaN | |

As we can see above, we have lots of columns, and nan Values. Lets fill all na using Value for same column present in other dataset columns. And get one neat column showing all the Value for that column in the final data set.

```
final_df['Title'] = final_df['Title'].fillna(final_df['Title2'])
final_df['OpenDate'] = final_df['OpenDate'].fillna(final_df['OpenDate2'])
final_df['CloseDate'] = final_df['CloseDate'].fillna(final_df['CloseDate2'])
final_df['Company'] = final_df['Company'].fillna(final_df['Company2'])
final_df['Location'] = final_df['Location'].fillna(final_df['Location2'])
final_df['Category'] = final_df['Category'].fillna(final_df['Category2'])
final_df['ContractType'] = final_df['ContractType'].fillna(final_df['ContractType2'])
final_df['ContractTime'] = final_df['ContractTime'].fillna(final_df['ContractTime2'])
final_df['Salary'] = final_df['Salary'].fillna(final_df['Salary2'])
```

```
final_df.head()
```

| | OpenDate2 | CloseDate2 | Title2 | Company2 | Location2 | Category2 | ContractType2 | Cont |
|---|---|---|---|---|---|---|---|---|
| **0** | 2013-10-06 00:00:00 | 2013-12-05 00:00:00 | AVIATION LOANS ADMINISTRATION | CER FINANCIAL | LONDON | ACCOUNTING & FINANCE | NaN | |
| **1** | 2012-10-03 12:00:00 | 2012-11-02 12:00:00 | PAYROLL ANALYST CITY UPTO | LMA RECRUITMENT LTD | LONDON | ACCOUNTING & FINANCE | NaN | |
| **2** | 2012-01-01 00:00:00 | 2012-01-31 00:00:00 | INVESTMENT TEAM ASSISTANT FOR LEADING PRIVATE ... | AUSTIN &REW LTD | LONDON | ACCOUNTING & FINANCE | NaN | |
| **3** | 2012-10-14 00:00:00 | 2012-11-13 00:00:00 | SWAPS COLLATERAL CONTROL OFFICER | BRIAN DURHAM RECRUITMENT SERVICES LTD | CITY | ACCOUNTING & FINANCE | NaN | |
| **4** | 2012-11-17 12:00:00 | 2013-01-16 12:00:00 | LOANS ADMINISTRATION TEMP | CER FINANCIAL | LONDON | ACCOUNTING & FINANCE | NaN | |

```
final_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 55471 entries, 0 to 55470
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   OpenDate2      334 non-null    object
 1   CloseDate2     334 non-null    object
 2   Title2         334 non-null    object
 3   Company2       334 non-null    object
```

```
 4   Location2      334 non-null    object
 5   Category2      334 non-null    object
 6   ContractType2  3 non-null      object
 7   ContractTime2  327 non-null    object
 8   Salary2        334 non-null    float64
 9   globalId       55471 non-null  object
 10  Id             55169 non-null  float64
 11  Title          55471 non-null  object
 12  Location       55471 non-null  object
 13  Company        55471 non-null  object
 14  ContractType   55171 non-null  object
 15  ContractTime   55465 non-null  object
 16  Category       55471 non-null  object
 17  Salary         55471 non-null  float64
 18  OpenDate       55471 non-null  object
 19  CloseDate      55471 non-null  object
 20  Source         55169 non-null  object
dtypes: float64(3), object(18)
memory usage: 9.3+ MB
```

This looks much better, We can see, Source and Id have more null Values, as these Columns are missing in dataset2, Also ContractType have more null Values as compared to Other Columns present in dataset2, which is because even in dataset2 it had only 3 non null Values.

Now We can delete all unwanted Columns, and also our globalId column.

```
# drop columns
final_df = final_df.drop(["Title2", "Location2", "Company2", "ContractType2",
                  "ContractTime2", "Category2", "Salary2", "OpenDate2", "CloseDate2", "globalId"], axis=1)
```

```
print(final_df.shape)
```

```
(55471, 11)
```

```
final_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 55471 entries, 0 to 55470
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Id            55169 non-null  float64
 1   Title         55471 non-null  object
 2   Location      55471 non-null  object
 3   Company       55471 non-null  object
 4   ContractType  55171 non-null  object
 5   ContractTime  55465 non-null  object
 6   Category      55471 non-null  object
 7   Salary        55471 non-null  float64
 8   OpenDate      55471 non-null  object
 9   CloseDate     55471 non-null  object
 10  Source        55169 non-null  object
dtypes: float64(2), object(9)
memory usage: 5.1+ MB
```

```
final_df.head()
```

| | Id | Title | Location | Company | ContractType | ContractTime | Category | Salary | O |
|---|---|---|---|---|---|---|---|---|---|
| **0** | NaN | AVIATION LOANS ADMINISTRATION | LONDON | CER FINANCIAL | NaN | Contract | ACCOUNTING & FINANCE | 33600.0 | 2 |
| **1** | NaN | PAYROLL ANALYST CITY UPTO | LONDON | LMA RECRUITMENT LTD | NaN | Permanent | ACCOUNTING & FINANCE | 35004.0 | 2 |
| **2** | NaN | INVESTMENT TEAM ASSISTANT FOR LEADING PRIVATE ... | LONDON | AUSTIN &REW LTD | NaN | Permanent | ACCOUNTING & FINANCE | 45000.0 | 2 |
| | | SWAPS COLLATERAL | | BRIAN DURHAM | | | ACCOUNTING | | 2 |

```
final_df['Id'].unique()
```

```
array([      nan, 62008199., 72689786., ..., 72703193., 72705197.,
       72705203.])
```

We have Id as unique Identifier, with 8 digit number in final dataset, which we got from dataset1. But we have Values missing in final dataset for Id column as well, due to Id not present in dataset2, So lets replace all those null Values, in final_df by generating random 8 digit number.

```
# generating unique 8 didgit number and replacing all null Values with it, then converting the Column to int type.
final_df['Id'] = final_df['Id'].fillna(final_df.apply(lambda _: str(uuid.uuid4().int)[:8], axis=1)).astype(int)
```

```
# Lets check if we generated correct random Id with out any duplicates.
final_df['Id'].duplicated().sum()
```

```
0
```

```
final_df.head()
```

| | Id | Title | Location | Company | ContractType | ContractTime | Category | Salar |
|---|---|---|---|---|---|---|---|---|
| **0** | 22503175 | AVIATION LOANS ADMINISTRATION | LONDON | CER FINANCIAL | NaN | Contract | ACCOUNTING & FINANCE | 33600 |
| **1** | 32795672 | PAYROLL ANALYST CITY UPTO | LONDON | LMA RECRUITMENT LTD | NaN | Permanent | ACCOUNTING & FINANCE | 35004 |
| **2** | 21048485 | INVESTMENT TEAM ASSISTANT FOR LEADING PRIVATE ... | LONDON | AUSTIN &REW LTD | NaN | Permanent | ACCOUNTING & FINANCE | 45000 |
| | | SWAPS COLLATERAL | | BRIAN DURHAM | | | ACCOUNTING | |

In the Final format we need the date in the format as 20131006T000000 this.

So lets convert, OpenDate and Close Date in that format.

```
# We have to remove : and - from the Date, and then insert T at 8th place. remove any unwanted spaces.

def removeSpecialChar(Date):
       # remove : and -
        Date = re.sub(r'(:|-)','', Date)
        # Add T before Time
        Date = re.sub(r'(?<=^.{8})','T', Date)
        # remove spaces
        Date = re.sub( '\s+|,', '', Date).strip()
        return Date
final_df['OpenDate'] = final_df.OpenDate.apply(lambda x: removeSpecialChar(x))
```

```
# We have to remove : and - from the Date, and then insert T at 8th place. remove any unwanted spaces.

def removeSpecialChar(Date):
       # remove : and -
        Date = re.sub(r'(:|-)','', Date)
        # Add T before Time
        Date = re.sub(r'(?<=^.{8})','T', Date)
        # remove spaces
        Date = re.sub( '\s+|,', '', Date).strip()
        return Date
final_df['CloseDate'] = final_df.CloseDate.apply(lambda x: removeSpecialChar(x))
```

```
final_df.head()
```

Looks better, as we wanted output format

```
# Checking if all datatypes are as per required format.
final_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 55471 entries, 0 to 55470
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Id            55471 non-null  int64  
 1   Title         55471 non-null  object 
 2   Location      55471 non-null  object 
 3   Company       55471 non-null  object 
 4   ContractType  55171 non-null  object 
 5   ContractTime  55465 non-null  object 
 6   Category      55471 non-null  object 
 7   Salary        55471 non-null  float64
 8   OpenDate      55471 non-null  object 
 9   CloseDate     55471 non-null  object 
 10  Source        55169 non-null  object 
dtypes: float64(1), int64(1), object(9)
memory usage: 5.1+ MB
```

## Saving data

Save the integrated data

```
# code to save output data
final_df.to_csv('dataset_integrated.csv',index=False)
```

## Summary

Give a short summary and anything you would like to talk about assessment 2 part 2 here.

The global key was created using four attributes, Title, Category, Location, and Company, in both the dataset.

The data was merged using outer merge, to get all the valuable data from both dataset on created global key, as there was no any common Unique Identifier.

Then the Value of NA was replaced by other dataset Value to get one uniform Column.

8 digit, Unique Id was generated using uuid for all the missing Id data in final dataset.

The OpenDate and CloseDate was converted to the required format.

We still have missing Values in Source, ContractTime, ContractType, we can think of those to be replaced by using Classification methods, as we have more categorical data which can be better handled using classification models.

Double-click (or enter) to edit

0s completed at 22:00