

Collect and Load data into database

```
In [102]: # Collect and Load data into database

import sqlite3
import pandas as pd
import numpy as np

#Connect to the SQLite database
conn = sqlite3.connect("F:\Assignment\Jobdb.sqlite")
cur = conn.cursor()

# Load the data into pandas dataframe
job_main_df = pd.read_sql_query("SELECT * FROM job_main", conn)
responsibilities_df = pd.read_csv("C:\Users\Administrator\Downloads\responsibilities.csv")

print(job_postings_data)
print(job_main_df.head())
print(responsibilities_data)
print(responsibilities_df.head())

Job Postings Data:
   scraped_id  webid  companyid  date_scraped \
0         16      1         16.0  2022-03-29 08:59:56.687006
1         17      1         17.0  2022-03-29 08:59:56.687006
2         24      2         24.0  2022-03-29 09:00:03.610569
3         45      1         45.0  2022-03-29 08:59:56.687006
4         59      1         61.0  2022-03-29 08:59:56.687006

   job_title \
0      Digital Marketing Executive
1      Credit Control Executive / Regional
2      Credit Controller
3      Digital Marketing Accounts Executive (SEO)
4      Account Executive (Marketing agency / up to 3k)

   date_posted  career_level  year_experience_min \
0  2022-03-17 20:46:49.000000  Not Specified      NaN
1  2022-02-27 16:08:06.000000  Junior Executive      3.0
2  2022-03-04 19:45:29.000000  Not Specified      NaN
3  2022-03-18 01:00:09.000000  Junior Executive      1.0
4  2022-03-18 14:01:46.000000  Junior Executive      2.0

   year_experience_max  currency  salary_min  salary_max  remote  source \
0      NaN      SGD      NaN      NaN      NaN      None
1      NaN      SGD      NaN      NaN      NaN      None
2      NaN      MYR      NaN      NaN      NaN      None
3      NaN      SGD      2500.0      4000.0      NaN      None
4      NaN      SGD      2500.0      3000.0      NaN      None

   last_seen  date_expired  salary
0  2022-03-29 08:59:56.687006      None      None
1  2022-03-29 08:59:56.687006      None      None
2  2022-03-29 09:00:03.610569      None      None
3  2022-03-29 08:59:56.687006      None      None
4  2022-03-29 08:59:56.687006      None      None

Responsibilities Data:
Unnamed: 0      responsibility
0      0      Design and influence a PR strategy and SMART P...
1      1      Ensure consistent and relevant customer Commun...
2      2      Research, write press releases and ensure that...
3      3      Maintain relationships with influential lifest...
4      4      Manage photo shoots within the hotel for fashi...
```

Cleaning Responsibilities text

```
In [103]: # Cleaning Responsibilities text

from collections import Counter
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
import string

# Download NLTK resources
nltk.download('punkt')
nltk.download('stopwords')

# Preprocess text data
def preprocess_text(text):
    tokens = word_tokenize(text.lower())
    tokens = [token for token in tokens if token not in string.punctuation and token not in stopwords.words('english')]
    return tokens

# Concatenate responsibilities text
responsibilities_text = ' '.join(responsibilities_df['responsibility'])

responsibilities_tokens = preprocess_text(responsibilities_text)

responsibilities_freq = Counter(responsibilities_tokens)

print("Most common responsibilities:")
print(responsibilities_freq.most_common(20))

[nltk_data] Downloading package punkt to
[nltk_data] C:\Users\Administrator\AppData\Local\NltkData\
[nltk_data] Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data] C:\Users\Administrator\AppData\Local\NltkData\
[nltk_data] Package stopwords is already up-to-date!
Most common responsibilities:
[('marketing', 510), ('media', 249), ('social', 216), ('campaigns', 179), ('sales', 178), ('digital', 141), ('manage', 135), ('content', 114), ('company', 107), ('strategies', 106), ('market', 104), ('brand', 104), ('customer', 1
03), ('develop', 93), ('plan', 82), ('new', 82), ('events', 80), ('activities', 76), ('support', 76), ('team', 75)]
```

Data Cleaning And Exploration

```
In [104]: # Check the data
job_main_df

Out[104]:
```

	scraped_id	webid	companyid	date_scraped	job_title	date_posted	career_level	year_experience_min	year_experience_max	currency	salary_min	salary_max	remote	source	last_seen	date_expired	salary
0	16	1	16.0	2022-03-29 08:59:56.687006	Digital Marketing Executive	2022-03-17 20:46:49.000000	Not Specified	NaN	NaN	SGD	NaN	NaN	NaN	None	2022-03-29 08:59:56.687006	None	None
1	17	1	17.0	2022-03-29 08:59:56.687006	Credit Control Executive / Regional	2022-02-27 16:08:06.000000	Junior Executive	3.0	NaN	MYR	NaN	NaN	NaN	None	2022-03-29 08:59:56.687006	None	None
2	24	2	24.0	2022-03-29 09:00:03.610569	Credit Controller	2022-03-04 19:45:29.000000	Not Specified	NaN	NaN	MYR	NaN	NaN	NaN	None	2022-03-29 09:00:03.610569	None	None
3	45	1	47.0	2022-03-29 08:59:56.687006	Digital Marketing Accounts Executive (SEO)	2022-03-18 01:00:09.000000	Junior Executive	1.0	NaN	SGD	2500.0	4000.0	NaN	None	2022-03-29 08:59:56.687006	None	None
4	59	1	61.0	2022-03-29 08:59:56.687006	Account Executive (Marketing agency / up to 3k)	2022-03-18 14:01:46.000000	Junior Executive	2.0	NaN	SGD	2500.0	3000.0	NaN	None	2022-03-29 08:59:56.687006	None	None
...
9996	4384856	4	189270.0	2022-12-16 00:00:13.78157	Marketing Executive	2023-11-20 07:23:10.000000	None	NaN	NaN	None	NaN	NaN	NaN	NaN	2023-12-20 00:51:37.201623	None	None
99967	4385152	4	101229.0	2022-12-16 00:00:13.78157	Marketing & Sales Executive	2023-11-16 04:46:19.000000	None	NaN	NaN	IDR	400000.0	700000.0	NaN	None	2023-12-16 02:25:09.970593	None	None
99968	4385626	4	101224.0	2022-12-16 00:00:13.78157	Sales & Marketing Executive (Freight Forwarding)	2023-11-17 04:42:05.000000	None	NaN	NaN	IDR	400000.0	600000.0	NaN	None	2023-12-16 18:03:24.893906	None	None
99969	4385734	4	305149.0	2022-12-16 00:00:13.78157	Marketing Executive	2023-11-16 08:13:25.000000	None	NaN	NaN	None	NaN	NaN	NaN	NaN	2023-12-16 18:04:51.022703	None	None
99970	4386010	4	439450.0	2022-12-16 00:00:13.78157	Marketing Executive	2023-11-16 02:57:42.000000	None	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2023-12-16 18:06:20.342627	None	None

59971 rows x 17 columns

```
In [105]: # Remove Unnecessary Column
job_main_df.drop(['webid', 'companyid', 'date_scraped', 'date_posted', 'source', 'last_seen', 'date_expired', 'year_experience_max'], axis=1, inplace=True)
job_main_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 59971 entries, 0 to 59970
Data columns (total 9 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   scraped_id            59971 non-null  int64
1   job_title             59971 non-null  object
2   career_level          42495 non-null  object
3   year_experience_min    25738 non-null  float64
4   currency              58154 non-null  object
5   salary_min            36455 non-null  float64
6   salary_max            36567 non-null  float64
7   remote                3492 non-null   float64
8   salary                0 non-null      object
dtypes: float64(4), int64(1), object(4)
memory usage: 4.1+ MB
```

```
In [106]: # Create Salary Column
job_main_df['salary'] = [job_main_df['salary_min'] + job_main_df['salary_max']]
```

```
In [107]: # Change Column name
job_main_df.rename(columns = {'year_experience_min': 'year_experience'}, inplace=True)
```

```
In [108]: # Create Experience Level Segment
job_main_df['experience_level'] = pd.cut(job_main_df['year_experience'], bins=[0,3,6,10], labels=['Entry Level', 'Middle Level', 'Senior Level'])
```

```
In [109]: job_main_df['experience_level'].value_counts()
```

```
Entry Level    22583
Middle Level   2381
Senior Level    517
Name: experience_level, dtype: int64
```

```
In [110]: job_main_df
```

```
Out[110]:
```

	scrapedid	job_title	career_level	year_experience	currency	salary_min	salary_max	remote	salary	experience_level
0	16	Digital Marketing Executive	Not Specified	NaN	SGD	NaN	NaN	NaN	NaN	NaN
1	17	Credit Control Executive / Regional	Junior Executive	3.0	SGD	NaN	NaN	NaN	NaN	Entry Level
2	24	Credit Controller	Not Specified	NaN	MYR	NaN	NaN	NaN	NaN	NaN
3	45	Digital Marketing Accounts Executive (SEO)	Junior Executive	1.0	SGD	2600.0	4000.0	NaN	6600.0	Entry Level
4	59	Account Executive (Marketing agency / up to 3k)	Junior Executive	2.0	SGD	2500.0	3000.0	NaN	5500.0	Entry Level
...
9996	4384856	Marketing Executive	None	NaN	None	NaN	NaN	NaN	NaN	NaN
99967	4385152	Marketing & Sales Executive	None	NaN	IDR	400000.0	700000.0	NaN	1100000.0	NaN
99968	4385626	Sales & Marketing Executive (Freight Forwarding)	None	NaN	IDR	400000.0	600000.0	NaN	1000000.0	NaN
99969	4385734	Marketing Executive	None	NaN	None	NaN	NaN	NaN	NaN	NaN
99970	4386010	Marketing Executive	None	NaN	None	NaN	NaN	NaN	NaN	NaN

59971 rows x 10 columns

```
In [111]: # Remove Non Value in the column
# Sum non value in each column
job_main_df.isnull().sum()
```

```
Out[111]:
```

scrapedid	0
job_title	0
career_level	17476
year_experience	34235
currency	1817
salary_min	29516
salary_max	29484
remote	56479
salary	29650
experience_level	34490
dtype: int64	

```
In [112]: # Drop Non Value in Salary and Experience Level
job_main_df.dropna(subset = ['salary', 'experience_level'], inplace = True)
job_main_df.isnull().sum()
```

```
Out[112]:
```

scrapedid	0
job_title	0
career_level	125
year_experience	0
currency	0
salary_min	0
salary_max	0
remote	14018
salary	0
experience_level	0
dtype: int64	

```
In [113]: # Replace Non value into Non-specific
job_main_df['remote'].fillna("Non-Specific", inplace=True)
replace = (0: 'No', 1: 'Yes')
job_main_df['remote'] = job_main_df['remote'].replace(replace)
```

```
# Check unique value in Remote Column
job_main_df['remote'].unique()
```

```
Out[113]: array(['Non-Specific', 'No', 'Yes'], dtype=object)
```

```
In [114]: # Count the employee of Remote Work
job_main_df['remote'].value_counts()
```

```
Out[114]:
```

Non-Specific	14018
1	119
3	3
Yes	remote, dtype: int64

```
In [115]: # Replace Column Name
job_main_df.rename(columns = {'currency': 'location'}, inplace=True)
```

```
In [116]: job_main_df['location'].unique()
```

```
Out[116]: array(['SGD', 'MYR', 'IDR', 'RM'], dtype=object)
```

```
In [117]: # Replace value in the Column
job_main_df['location']=job_main_df['location'].str.replace('SGD', 'Singapore')
job_main_df['location']=job_main_df['location'].str.replace('MYR', 'Malaysia')
job_main_df['location']=job_main_df['location'].str.replace('RM', 'Malaysia')
job_main_df['location']=job_main_df['location'].str.replace('IDR', 'Indonesia')
```

```
In [118]: # Count value of location
job_main_df['location'].value_counts()
```

```
Out[118]:
```

Malaysia	6984
Singapore	6058
Indonesia	2998
Name: location, dtype: int64	

```
In [119]: job_main_df
```

```
Out[119]:
```

	scrapedid	job_title	career_level	year_experience	location	salary_min	salary_max	remote	salary	experience_level
3	45	Digital Marketing Accounts Executive (SEO)	Junior Executive	1.0	Singapore	2600.0	4000.0	Non-Specific	6600.0	Entry Level
1	59	Account Executive (Marketing agency / up to 3k)	Junior Executive	2.0	Singapore	2500.0	3000.0	Non-Specific	5500.0	Entry Level
0	206	Sales Marketing Executive	Senior Executive	3.0	Malaysia	3500.0	6000.0	Non-Specific	9500.0	Entry Level
10	219	Digital Marketing Executive	Junior Executive	2.0	Malaysia	3000.0	4000.0	Non-Specific	7500.0	Entry Level
11	221	Marketing Executive	Junior Executive	3.0	Malaysia	1500.0	2000.0	Non-Specific	3500.0	Entry Level
...
59611	4258931	Marketing Executive	None	1.0	Singapore	2500.0	5000.0	Yes	7500.0	Entry Level
59614	4259106	Sales & Marketing Executive	None	1.0	Singapore	1800.0	10000.0	No	11800.0	Entry Level
59774	4356997	Customer Relation & Marketing Executive	None	1.0	Singapore	1500.0	2500.0	No	4000.0	Entry Level
59776	4357056	Marketing & Customer Relation Executive	None	1.0	Singapore	1500.0	2500.0	No	4000.0	Entry Level
59778	4357261	Marketing Executive	None	3.0	Singapore	3000.0	3000.0	No	6000.0	Entry Level

14140 rows x 10 columns

```
In [120]: # Create Fair Salary Range Column And Calculation by Grouping Data
grouped_data = job_main_df.groupby(['job_title']).agg({'salary_min': 'mean', 'salary_max': 'mean'}).reset_index()

# Calculate fair salary range for each job title
grouped_data['fair_salary_range'] = grouped_data['salary_max'] - grouped_data['salary_min']

print(grouped_data)
```

```
Out[120]:
```

		job_title	salary_min	
0	\$4000 - \$4800	Digital Marketing Executive	4800.0	
1	\$4500 / Senior Marketing Executive / Kallang / ...	4800.0		
2	(E-commerce) Local Marketing Executive / Advert...	15800860.0		
3	(GOVT) Marketing Admin Executive - SY	3300.0		
4	(GOVT) Marketing Executive Contract Degree...	2800.0		
...
5684	市场部专员	MARKETING EXECUTIVE	2800.0	
5685	数码销售精英 (日语)	Digital Marketing Executive (Japanes...	4500.0	
5686	社交媒体营销专员	Social Media Marketing Executive	2600.0	
5687	高级数字营销专员	Senior Digital Marketing Executive	5000.0	
5688	高级数码营销专员	Senior Digital Marketing Executive	5500.0	

```
Out[120]:
```

		salary_max	fair_salary_range
0	4800.0	800.0	
1	4500.0	400.0	
2	18000000.0	3000000.0	
3	3450.0	150.0	
4	3450.0	650.0	
...
5684	3500.0	700.0	
5685	4000.0	1000.0	
5686	3500.0	900.0	
5687	6000.0	1000.0	
5688	7000.0	1500.0	

[5689 rows x 4 columns]

```
In [121]: job_main_df[['job_title']].nunique()
```

```
Out[121]: 5689
```

```
In [122]: # Merge Fair Salary Range into data column
job_main_df = pd.merge(job_main_df, grouped_data[['job_title', 'fair_salary_range']], on='job_title', how='left')
```

```
Out[122]: job_main_df
```

```
Out[122]:
```

	scrapedid	job_title	career_level	year_experience	location	salary_min	salary_max	remote	salary	experience_level	fair_salary_range
0	45	Digital Marketing Accounts Executive (SEO)	Junior Executive	1.0	Singapore	2600.0	4000.0	Non-Specific	6600.0	Entry Level	1050.000000
1	59	Account Executive (Marketing agency / up to 3k)	Junior Executive	2.0	Singapore	2500.0	3000.0	Non-Specific	5500.0	Entry Level	500.000000
2	206	Sales Marketing Executive	Senior Executive	3.0	Malaysia	3500.0	6000.0	Non-Specific	9500.0	Entry Level	317231.818182
3	219	Digital Marketing Executive	Junior Executive	2.0	Malaysia	3000.0	4000.0	Non-Specific	7500.0	Entry Level	73239.301355
4	221	Marketing Executive	Junior Executive	3.0	Malaysia	1500.0	2000.0	Non-Specific	3500.0	Entry Level	217963.830294
...
14135	4258931	Marketing Executive	None	1.0	Singapore	2500.0	5000.0	Yes	7500.0	Entry Level	217963.830294
14136	4259106	Sales & Marketing Executive	None	1.0	Singapore	1800.0	10000.0	No	11800.0	Entry Level	96244.211538
14137	4356997	Customer Relation & Marketing Executive	None	1.0	Singapore	1500.0	2500.0	No	4000.0	Entry Level	1000.000000
14138	4357056	Marketing & Customer Relation Executive	None	1.0	Singapore	1500.0	2500.0	No	4000.0	Entry Level	1000.000000
14139	4357261	Marketing Executive	None	3.0	Singapore	3000.0	3000.0	No	6000.0	Entry Level	217963.830294

14140 rows x 11 columns

```
In [124]: # Count the top 10 job in the data
job_main_df[['job_title']].value_counts().nlargest(10)
```

```
Out[124]:
```

Marketing Executive	2251
Digital Marketing Executive	1268
Sales & Marketing Executive	529
Senior Marketing Executive	329
MARKETING EXECUTIVE	245
DIGITAL MARKETING EXECUTIVE	131
Senior Digital Marketing Executive	141
SALES & MARKETING EXECUTIVE	89
MARKETING EXECUTIVE	86
IT Executive	56
Name: job_title, dtype: int64	

```
In [125]: # Replace the name of the top 10 name
job_main_df[['job_title']] = job_main_df[['job_title']].str.replace('Sales and Marketing Executive', 'Sales & Marketing Executive')
job_main_df[['job_title']] = job_main_df[['job_title']].str.replace('SALES & Marketing Executive', 'Sales & Marketing Executive')
job_main_df[['job_title']] = job_main_df[['job_title']].str.replace('DIGITAL Marketing Executive', 'Digital Marketing Executive')
job_main_df[['job_title']] = job_main_df[['job_title']].str.replace('MARKETING EXECUTIVE', 'Marketing Executive')
```

```
In [126]: # Check if the data already change
job_main_df[['job_title']].value_counts().nlargest(10)
```

```
Out[126]:
```

Marketing Executive	2496
Digital Marketing Executive	1268
Sales & Marketing Executive	729
Senior Marketing Executive	329
Senior Digital Marketing Executive	141
DIGITAL MARKETING EXECUTIVE	90
SALES & Marketing Executive	86
IT Executive	56
Social Media Marketing Executive	55
Support	51
Name: job_title, dtype: int64	

```
In [127]: # Load Data of type of work into database
job_type_df = pd.read_sql_query("SELECT * FROM job_type", conn)
print("Job type Data:")
print(job_type_df.head())
```

```
Job type Data:
   scraped_id  type
0  2656811  full-time
1   911238  full-time
2  4273934  full-time
3    6844  full-time
4  4140110  full-time
```