In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

```
C:\Users\LENOVO\anaconda3\lib\site-packages\pandas\core\computation\expressio
ns.py:21: UserWarning: Pandas requires version '2.8.4' or newer of 'numexpr'
(version '2.8.1' currently installed).
  from pandas.core.computation.check import NUMEXPR_INSTALLED
C:\Users\LENOVO\anaconda3\lib\site-packages\pandas\core\arrays\masked.py:60:
UserWarning: Pandas requires version '1.3.6' or newer of 'bottleneck' (versio
n '1.3.4' currently installed).
  from pandas.core import (
C:\Users\LENOVO\anaconda3\lib\site-packages\scipy\__init__.py:155: UserWarnin
g: A NumPy version >=1.18.5 and <1.25.0 is required for this version of SciPy
(detected version 1.26.4
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

In [2]:
```
df=pd.read_csv(r"C:\Users\LENOVO\Downloads\Task_Dataset\Task_Dataset\Employees
```

In [3]:
```
df
```

Out[3]:

	ID	Emploee Name	Education	Passport NO	Phone Number	Department	Job Status	Location	Sta Da
0	8A78C6	Aba' Shahada	Institute	N964213362	5.802648e+09	FSL	Full Time	United Arab Emirates	2 Au
1	1N28R7	librahim Alhamid	Bachelor	N386537014	5.378887e+09	FSL	Full Time	Saudi Arabia	0 Fe
2	9S94G5	librahim Alhamid	Prof	N800905161	5.658057e+09	NFI	Full Time	United Arab Emirates	1 Ju
3	9N59A9	librahim Alqatish	Doctor	N954891059	5.195859e+09	FSL	Full Time	Syria	0 Ja
4	1D69A7	librahim Almasri	Bachelor	N160988977	5.063557e+09	FSL	Full Time	United Arab Emirates	1 Ju
...	...	...	...	...	...	...	...	...	
944	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
945	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
946	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
947	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
948	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na

949 rows × 13 columns

In [4]:
```
df.isnull().sum()
```

Out[4]:
```
ID              209
Emploee Name    209
Education       209
Passport NO     209
Phone Number    209
Department      209
Job Status      209
Location        209
Start Date      209
Years           209
 Salary         209
Job Rate        209
 Permissions    209
dtype: int64
```

In [5]:
```
df.dropna()
```

Out[5]:

	ID	Emploee Name	Education	Passport NO	Phone Number	Department	Job Status	Location
0	8A78C6	Aba' Shahada	Institute	N964213362	5.802648e+09	FSL	Full Time	United Arab Emirates
1	1N28R7	Iibrahim Alhamid	Bachelor	N386537014	5.378887e+09	FSL	Full Time	Saudi Arabia
2	9S94G5	Iibrahim Alhamid	Prof	N800905161	5.658057e+09	NFI	Full Time	United Arab Emirates
3	9N59A9	Iibrahim Alqatish	Doctor	N954891059	5.195859e+09	FSL	Full Time	Syria
4	1D69A7	Iibrahim Almasri	Bachelor	N160988977	5.063557e+09	FSL	Full Time	United Arab Emirates
...	...	...	...	...	...	...	...	...
735	5F82R8	Muhamad Eurul	Academic	N631479661	5.818445e+09	Protection	Full Time	United Arab Emirates
736	8U56Z9	Muhamad Eizat Almaghribiu Almisriu	Doctor	N692504829	5.269412e+09	Education	Full Time	United Arab Emirates
737	6C14T3	Muhamad Eataya	Institute	N591334404	5.699178e+09	IT	Full Time	United Arab Emirates
738	5U84O8	Muhamad Eaqad	Academic	N252728874	5.698131e+09	Training	Contract	United Arab Emirates
739	3C97D6	Muhamad Eala' Aldiyn Qamar	Academic	N924200229	5.458637e+09	NFI	Full Time	Saudi Arabia

740 rows × 13 columns

In [6]:
```
df.drop_duplicates(inplace=True)
df
```

Out[6]:

	ID	Emploee Name	Education	Passport NO	Phone Number	Department	Job Status	Location
0	8A78C6	Aba' Shahada	Institute	N964213362	5.802648e+09	FSL	Full Time	United Arab Emirates
1	1N28R7	Iibrahim Alhamid	Bachelor	N386537014	5.378887e+09	FSL	Full Time	Saudi Arabia
2	9S94G5	Iibrahim Alhamid	Prof	N800905161	5.658057e+09	NFI	Full Time	United Arab Emirates
3	9N59A9	Iibrahim Alqatish	Doctor	N954891059	5.195859e+09	FSL	Full Time	Syria
4	1D69A7	Iibrahim Almasri	Bachelor	N160988977	5.063557e+09	FSL	Full Time	United Arab Emirates
...	...	...	...	...	...	...	...	...
736	8U56Z9	Muhamad Eizat Almaghribiu Almisriu	Doctor	N692504829	5.269412e+09	Education	Full Time	United Arab Emirates
737	6C14T3	Muhamad Eataya	Institute	N591334404	5.699178e+09	IT	Full Time	United Arab Emirates
738	5U84O8	Muhamad Eaqad	Academic	N252728874	5.698131e+09	Training	Contract	United Arab Emirates
739	3C97D6	Muhamad Eala' Aldiyn Qamar	Academic	N924200229	5.458637e+09	NFI	Full Time	Saudi Arabia
740	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

741 rows × 13 columns

In [7]:
```
df.dropna(inplace=True)
```

In [8]:
```
df
```

Out[8]:

	ID	Emploee Name	Education	Passport NO	Phone Number	Department	Job Status	Location
0	8A78C6	Aba' Shahada	Institute	N964213362	5.802648e+09	FSL	Full Time	United Arab Emirates
1	1N28R7	Iibrahim Alhamid	Bachelor	N386537014	5.378887e+09	FSL	Full Time	Saudi Arabia
2	9S94G5	Iibrahim Alhamid	Prof	N800905161	5.658057e+09	NFI	Full Time	United Arab Emirates
3	9N59A9	Iibrahim Alqatish	Doctor	N954891059	5.195859e+09	FSL	Full Time	Syria
4	1D69A7	Iibrahim Almasri	Bachelor	N160988977	5.063557e+09	FSL	Full Time	United Arab Emirates
...	...	...	...	...	...	...	...	...
735	5F82R8	Muhamad Eurul	Academic	N631479661	5.818445e+09	Protection	Full Time	United Arab Emirates
736	8U56Z9	Muhamad Eizat Almaghribiu Almisriu	Doctor	N692504829	5.269412e+09	Education	Full Time	United Arab Emirates
737	6C14T3	Muhamad Eataya	Institute	N591334404	5.699178e+09	IT	Full Time	United Arab Emirates
738	5U84O8	Muhamad Eaqad	Academic	N252728874	5.698131e+09	Training	Contract	United Arab Emirates
739	3C97D6	Muhamad Eala' Aldiyn Qamar	Academic	N924200229	5.458637e+09	NFI	Full Time	Saudi Arabia

740 rows × 13 columns

In [9]:
```
df.describe()
```

Out[9]:

	Phone Number	Job Rate	Permissions
count	7.400000e+02	740.000000	740.000000
mean	5.509942e+09	5.291892	7.435135
std	2.809182e+08	3.454900	4.125876
min	5.000713e+09	1.000000	1.000000
25%	5.280794e+09	3.000000	4.000000
50%	5.499114e+09	5.000000	7.000000
75%	5.751572e+09	8.000000	11.000000
max	5.996845e+09	13.000000	14.000000

In [56]:
```
df['Phone Number'] = df['Phone Number'].astype(str)
df['Years'] = df['Years'].astype(str)
```

In [10]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 740 entries, 0 to 739
Data columns (total 13 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   ID             740 non-null    object 
 1   Emploee Name   740 non-null    object 
 2   Education      740 non-null    object 
 3   Passport NO    740 non-null    object 
 4   Phone Number   740 non-null    float64
 5   Department     740 non-null    object 
 6   Job Status     740 non-null    object 
 7   Location       740 non-null    object 
 8   Start Date     740 non-null    object 
 9   Years          740 non-null    object 
 10   Salary        740 non-null    object 
 11  Job Rate       740 non-null    float64
 12   Permissions   740 non-null    float64
dtypes: float64(3), object(10)
memory usage: 80.9+ KB
```

In [11]:
```
df.describe(include='object')
```

Out[11]:

	ID	Emploee Name	Education	Passport NO	Department	Job Status	Location	Start Date	Years	S
count	740	740	740	740	740	740	740	740	740	
unique	740	733	5	740	21	3	4	668	10	
top	8A78C6	librahim Alhamid	Academic	N964213362	Protection	Full Time	United Arab Emirates	16-Dec-16	2	
freq	1	2	312	1	151	393	278	3	97	

In [12]:
```
df.columns
```

Out[12]:
```
Index(['ID', 'Emploee Name', 'Education', 'Passport NO', 'Phone Number',
       'Department', 'Job Status', 'Location', 'Start Date', 'Years',
       ' Salary ', 'Job Rate', ' Permissions '],
      dtype='object')
```

In [13]:
```
df = df.rename(columns={' Salary ': 'Salary', ' Permissions ': 'Permissions'})
```

In [14]:
```
df.columns
```

Out[14]:
```
Index(['ID', 'Emploee Name', 'Education', 'Passport NO', 'Phone Number',
       'Department', 'Job Status', 'Location', 'Start Date', 'Years', 'Salar
y',
       'Job Rate', 'Permissions'],
      dtype='object')
```

In [15]:
```
df['Salary'] = df['Salary'].replace({'\$': '', ',': ''}, regex=True).astype(fl

pd.DataFrame(df.groupby('Location')['Salary'].mean())
```

Out[15]:

	Salary
Location	
Egypt	1548.005405
Saudi Arabia	1530.760870
Syria	1526.545946
United Arab Emirates	1486.992806

In [57]:
```
df.describe()
```

Out[57]:

	Salary	Job Rate	Permissions
count	740.000000	740.000000	740.000000
mean	1517.575676	5.291892	7.435135
std	594.863047	3.454900	4.125876
min	650.000000	1.000000	1.000000
25%	970.750000	3.000000	4.000000
50%	1580.500000	5.000000	7.000000
75%	2074.000000	8.000000	11.000000
max	2500.000000	13.000000	14.000000

Salary Distribution: The salary range spans from 650 to 2500, with a significant variation (standard deviation of $594.86). This wide range and high variation might indicate diverse job roles or levels within the organization. Ensuring competitive and equitable salaries could be essential for employee satisfaction and retention.

Job Rate Consistency: The job rates range from 1 to 13, with a mean of 5.29. The variation suggests that there are multiple job categories or evaluation criteria. It may be beneficial to align job rates more closely with job responsibilities to ensure fairness and transparency in compensation.

Permissions Distribution: The permissions range from 1 to 14, with a mean of 7.44. The broad range indicates variability in access levels, which might be linked to different job roles or responsibilities. Reviewing and possibly standardizing permissions could enhance security and operational efficiency.

In [16]:
```
pd.DataFrame(df.groupby('Location')['Salary'].sum())
```

Out[16]:

	Salary
Location	
Egypt	286381.0
Saudi Arabia	140830.0
Syria	282411.0
United Arab Emirates	413384.0

In [17]:
```
pd.DataFrame(df.groupby('Job Status')['Salary'].mean())
```

Out[17]:

Job Status	Salary
Contract	871.488095
Full Time	2032.760814
Part Time	1100.168421

In [18]:
```
pd.DataFrame(df.groupby('Job Status')['Salary'].sum())
```

Out[18]:

Job Status	Salary
Contract	219615.0
Full Time	798875.0
Part Time	104516.0

In [19]:
```
df['Years'].value_counts()
```

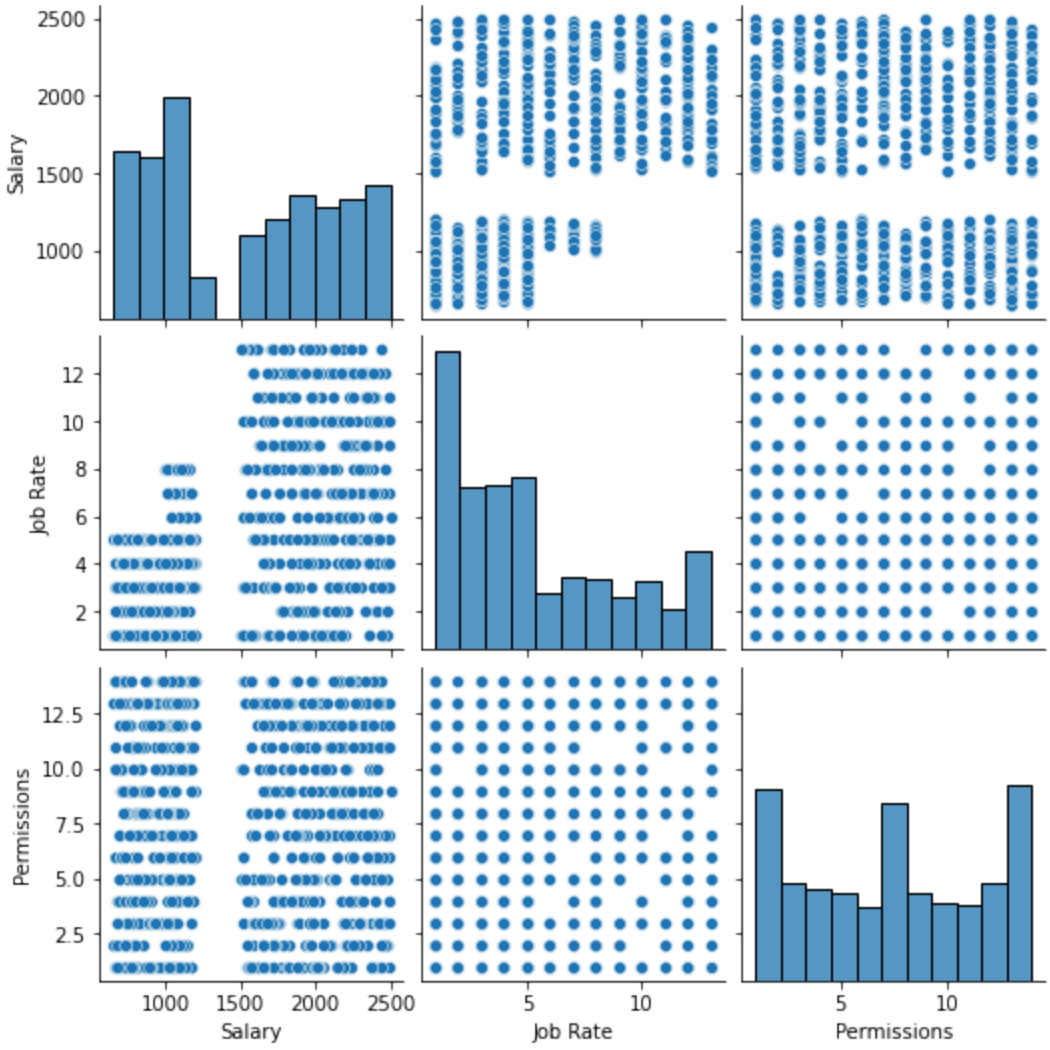
Out[19]:
```
Years
2       97
3       94
6       91
5       88
7       81
4       80
1       78
-       72
8       54
9        5
Name: count, dtype: int64
```

In [20]:
```
df['Years'] = df['Years'].str.strip().replace('-', '0')

df['Years'] = df['Years'].astype(str)

print(df['Years'].value_counts())
```

```
Years
2    97
3    94
6    91
5    88
7    81
4    80
1    78
0    72
8    54
9     5
Name: count, dtype: int64
```

In [59]:
```
sns.pairplot(df)
```

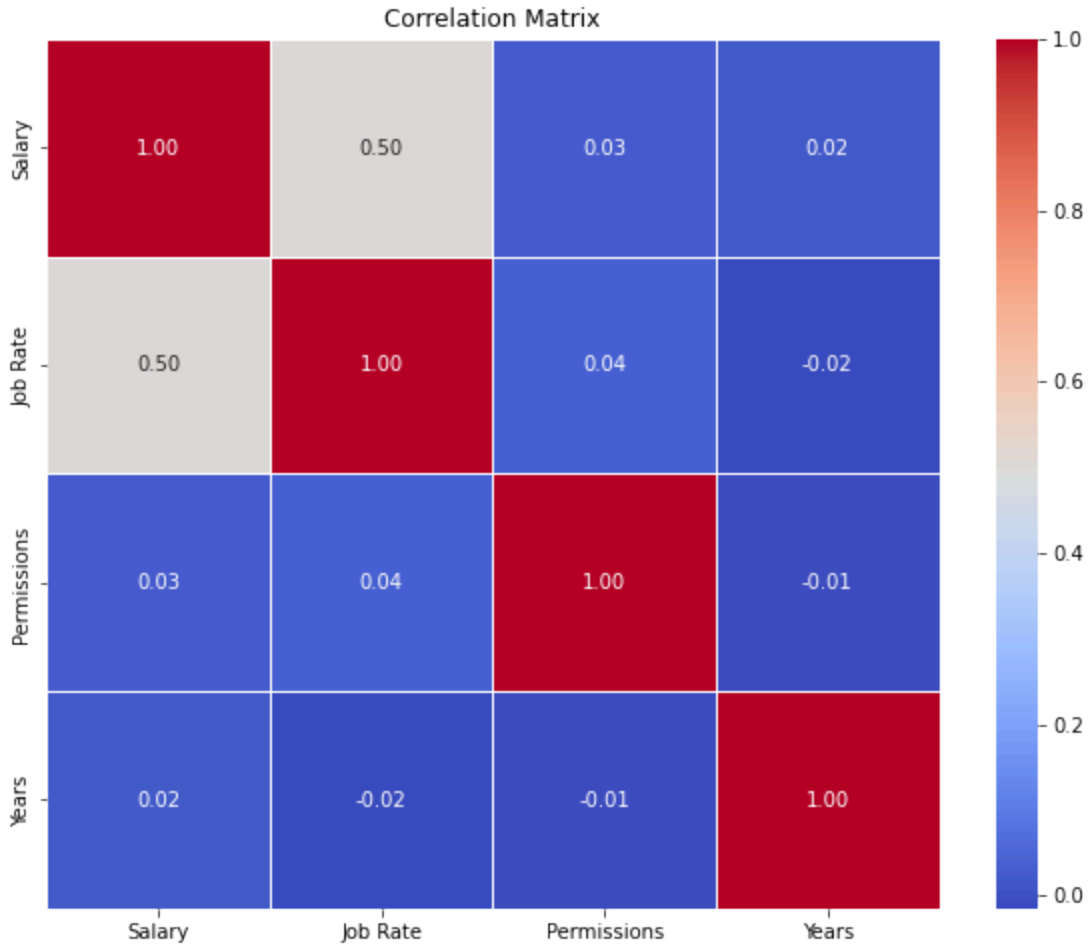
Out[59]: <seaborn.axisgrid.PairGrid at 0x20ee9d05f40>



In [85]:
```
df.info()
```

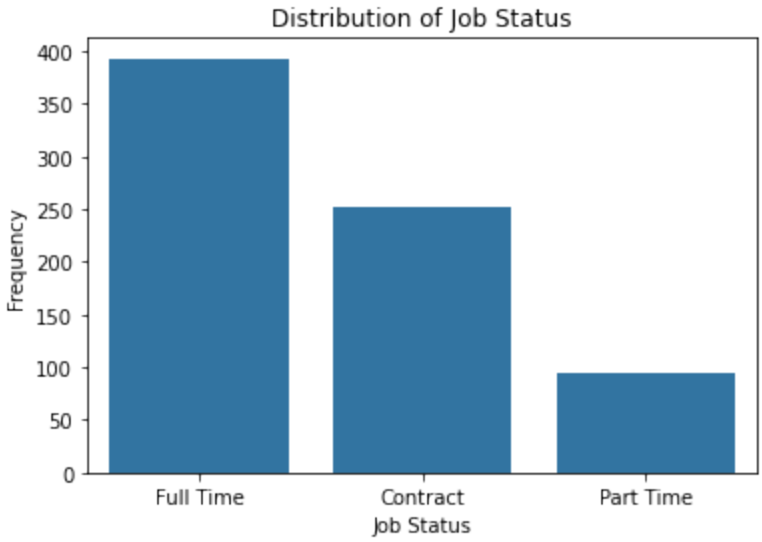
```
<class 'pandas.core.frame.DataFrame'>
Index: 740 entries, 0 to 739
Data columns (total 13 columns):
 #   Column         Non-Null Count  Dtype   
---  ------         --------------  -----   
 0   ID             740 non-null    object  
 1   Emploee Name   740 non-null    object  
 2   Education      740 non-null    object  
 3   Passport NO    740 non-null    object  
 4   Phone Number   740 non-null    object  
 5   Department     740 non-null    category
 6   Job Status     740 non-null    object  
 7   Location       740 non-null    category
 8   Start Date     740 non-null    object  
 9   Years          740 non-null    object  
 10  Salary         740 non-null    float64 
 11  Job Rate       740 non-null    float64 
 12  Permissions    740 non-null    float64 
dtypes: category(2), float64(3), object(8)
memory usage: 71.7+ KB
```

In [86]:
```
copy=df.copy()
copy['Years']=copy['Years'].astype(int)
df_subset = copy[['Salary', 'Job Rate', 'Permissions', 'Years']]
correlation_matrix = df_subset.corr()
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewi
plt.title('Correlation Matrix')
plt.show()
```



Most of these columns doesn't have high correleation between them. Just job rate and salary have a moderate correlation between them

In [22]:
```
sns.countplot(x='Job Status', data=df)

plt.xlabel('Job Status')
plt.ylabel('Frequency')
plt.title('Distribution of Job Status')

plt.show()
```

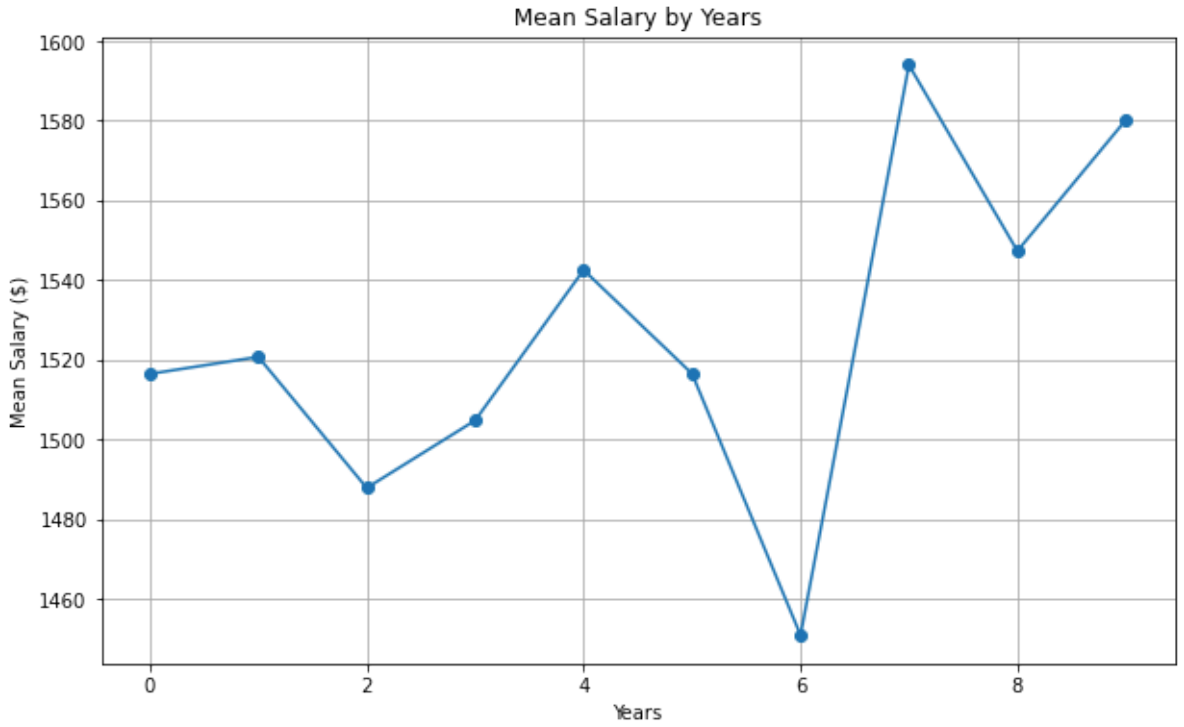


Most of workers have a "Full time" contract more than any other type

In [60]:
```
df.groupby("Years")['Salary'].mean()
```

Out[60]:
```
Years
0    1516.430556
1    1520.692308
2    1487.793814
3    1504.755319
4    1542.525000
5    1516.465909
6    1450.846154
7    1593.987654
8    1547.296296
9    1580.200000
Name: Salary, dtype: float64
```

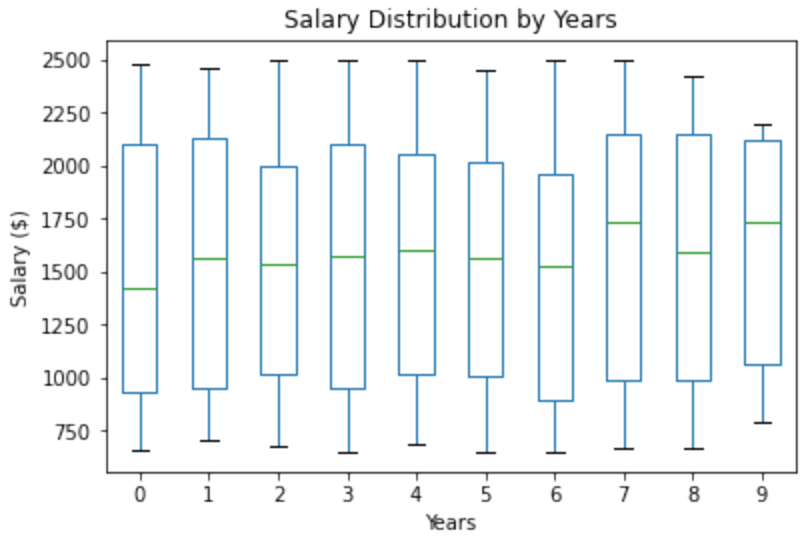
In [61]:
```
mean_salary_by_year = df.groupby("Years")['Salary'].mean()

plt.figure(figsize=(10, 6))
mean_salary_by_year.plot(kind='line', marker='o')

plt.xlabel('Years')
plt.ylabel('Mean Salary ($)')
plt.title('Mean Salary by Years')

plt.grid(True)
plt.show()
```



the data has slightly a direct relation between years and the mean of salaries except in "6-year" salaries has the lowest mean salaries

In [25]:
```
plt.figure(figsize=(12, 8))
df.boxplot(column='Salary', by='Years', grid=False)
plt.title('Salary Distribution by Years')
plt.suptitle('')
plt.xlabel('Years')
plt.ylabel('Salary ($)')
plt.show()
```

```
<Figure size 864x576 with 0 Axes>
```



The salaries are normally distributed along the years of work

In [62]:
```
year_6_rows = df[df['Years'] == '6']
pd.DataFrame(year_6_rows)
```

Out[62]:

	ID	Emploee Name	Education	Passport NO	Phone Number	Department	Job Status	Location	S
14	7C96O1	Ahmad Aldwltali	Institute	N465807878	5893682917.0	Wash	Full Time	Syria	
16	2B14Y4	Ahmad Alshamy	Academic	N451552435	5909614671.0	Protection	Contract	United Arab Emirates	
19	9G76H4	Ahmad Alghurani	Prof	N896508694	5075624441.0	FSL	Contract	Saudi Arabia	
30	1F88Q3	Ahmad Dahman	Doctor	N567879762	5574358586.0	FSL	Contract	Syria	
44	8R63D7	Ahmad Laylana	Doctor	N167783383	5868021164.0	Education	Full Time	United Arab Emirates	
...	...	...	...	...	...	...	...	...	...
717	7I35U1	Muhamad Saeadat	Prof	N533414049	5255811383.0	Finance	Contract	United Arab Emirates	
718	8P45B5	Muhamad Siedih	Academic	N863854592	5555753912.0	Logistics	Full Time	Egypt	
720	1F77M2	Muhamad Suqabani	Academic	N587263727	5411553824.0	Protection	Contract	Saudi Arabia	
726	5X28X6	Muhamad Shrbjy	Academic	N768773282	5260845960.0	M&E	Full Time	United Arab Emirates	
727	4H56S5	Muhamad Sharif Aldaghly	Academic	N762641326	5383931057.0	NFI	Contract	United Arab Emirates	

91 rows × 13 columns

In [64]:
```
year_6_rows['Job Status'].value_counts()
```

Out[64]:
```
Job Status
Full Time    46
Contract     37
Part Time     8
Name: count, dtype: int64
```

In [65]:
```
year_6_rows['Location'].value_counts()
```

Out[65]:
```
Location
United Arab Emirates    35
Egypt                   22
Syria                   21
Saudi Arabia            13
Name: count, dtype: int64
```

In [66]:
```
year_6_rows['Department'].value_counts()
```

Out[66]:
```
Department
Protection      24
Finance         12
Logistics       11
FSL             10
IT               8
Marketing        5
Wash             4
Emeergincy       3
NFI              3
Shelter          3
Training         3
Livelihoods      1
M&E              1
Researches       1
TPM              1
Education        1
Health           0
Media            0
Projects         0
HR               0
Supply chain     0
Name: count, dtype: int64
```

In [67]:
```
year_6_rows['Education'].value_counts()
```

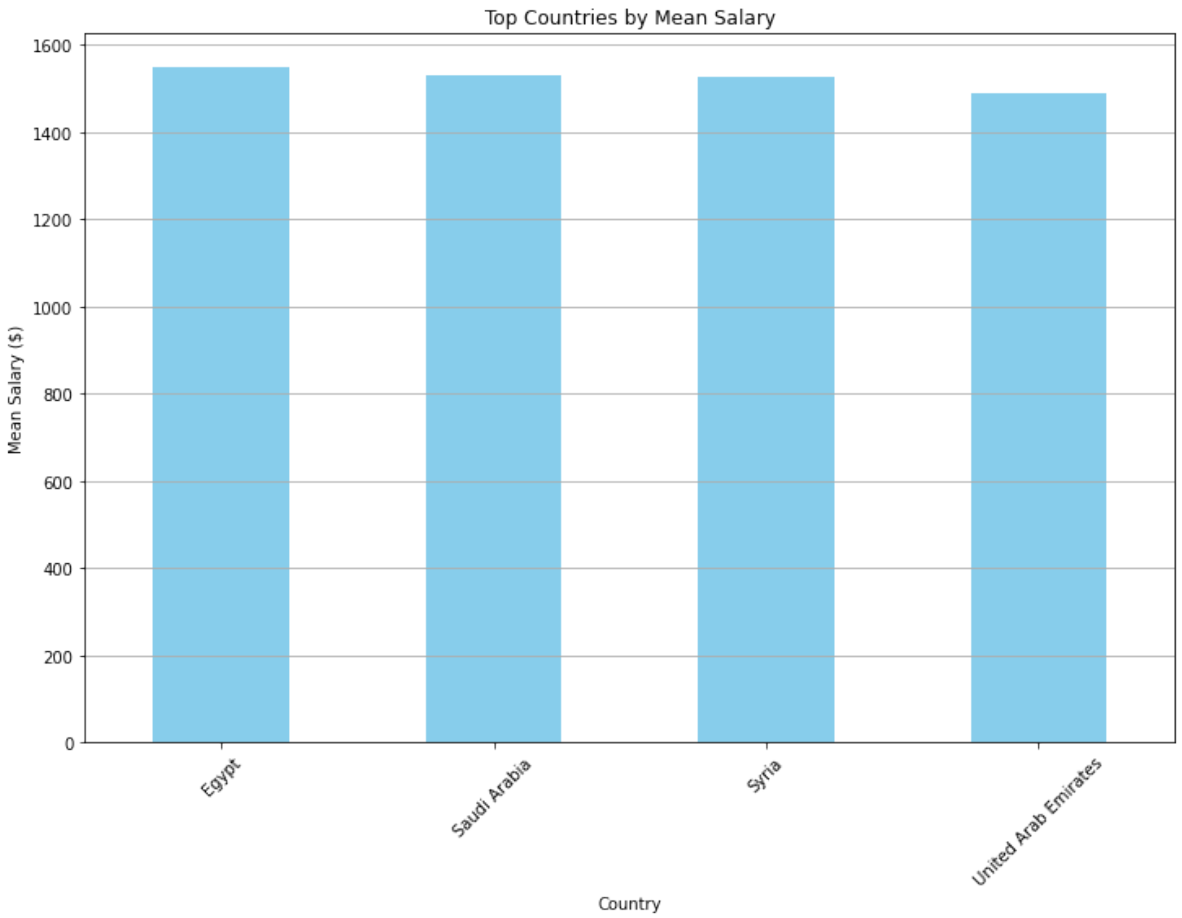
Out[67]:
```
Education
Academic     37
Institute    22
Doctor       12
Bachelor     12
Prof          8
Name: count, dtype: int64
```

The reason of 6-years have the lowest mean salaries might be for some reasons :

- Salary Plateau that Employees with 6 years of experience might be at a salary plateau, where their salary growth has slowed or stagnated compared to those with slightly fewer or more years of experience.
- Promotion Cycles that Employees with 6 years of experience might be in a transition phase, moving from mid-level to senior roles, which could affect their current salary if they have not yet been promoted.

- Job Market Trends that Market conditions during the period could have impacted salaries for employees with around 6 years of experience, possibly due to an oversupply of such professionals
- Data Anomalies or Outliers that There might be outliers or data anomalies skewing the

In [31]:
```
mean_salary_by_country = df.groupby('Location')['Salary'].mean()

sorted_mean_salary_by_country = mean_salary_by_country.sort_values(ascending=F

top_countries = sorted_mean_salary_by_country.head(10)

plt.figure(figsize=(12, 8))
top_countries.plot(kind='bar', color='skyblue')
plt.xlabel('Country')
plt.ylabel('Mean Salary ($)')
plt.title('Top Countries by Mean Salary')
plt.xticks(rotation=45)
plt.grid(axis='y')

plt.show()
```



In [68]:
```
df.groupby('Location')['Salary'].mean()
```

```
C:\Users\LENOVO\AppData\Local\Temp\ipykernel_10716\2867078878.py:1: FutureWar
ning: The default of observed=False is deprecated and will be changed to True
in a future version of pandas. Pass observed=False to retain current behavior
or observed=True to adopt the future default and silence this warning.
  df.groupby('Location')['Salary'].mean()
```

Out[68]:
```
Location
Egypt                   1548.005405
Saudi Arabia            1530.760870
Syria                   1526.545946
United Arab Emirates    1486.992806
Name: Salary, dtype: float64
```

Egypt has the highest mean salaries among these countries

In [32]:
```
mean_salary_by_department = df.groupby('Department')['Salary'].mean()

sorted_mean_salary_by_department = mean_salary_by_department.sort_values(ascen

print(sorted_mean_salary_by_department.head())
```

```
Department
Media         1796.222222
Emeergincy    1740.187500
Health        1715.571429
Education     1707.000000
Shelter       1664.764706
Name: Salary, dtype: float64
```

- Employees in Media Department has the highest salaries than any other department

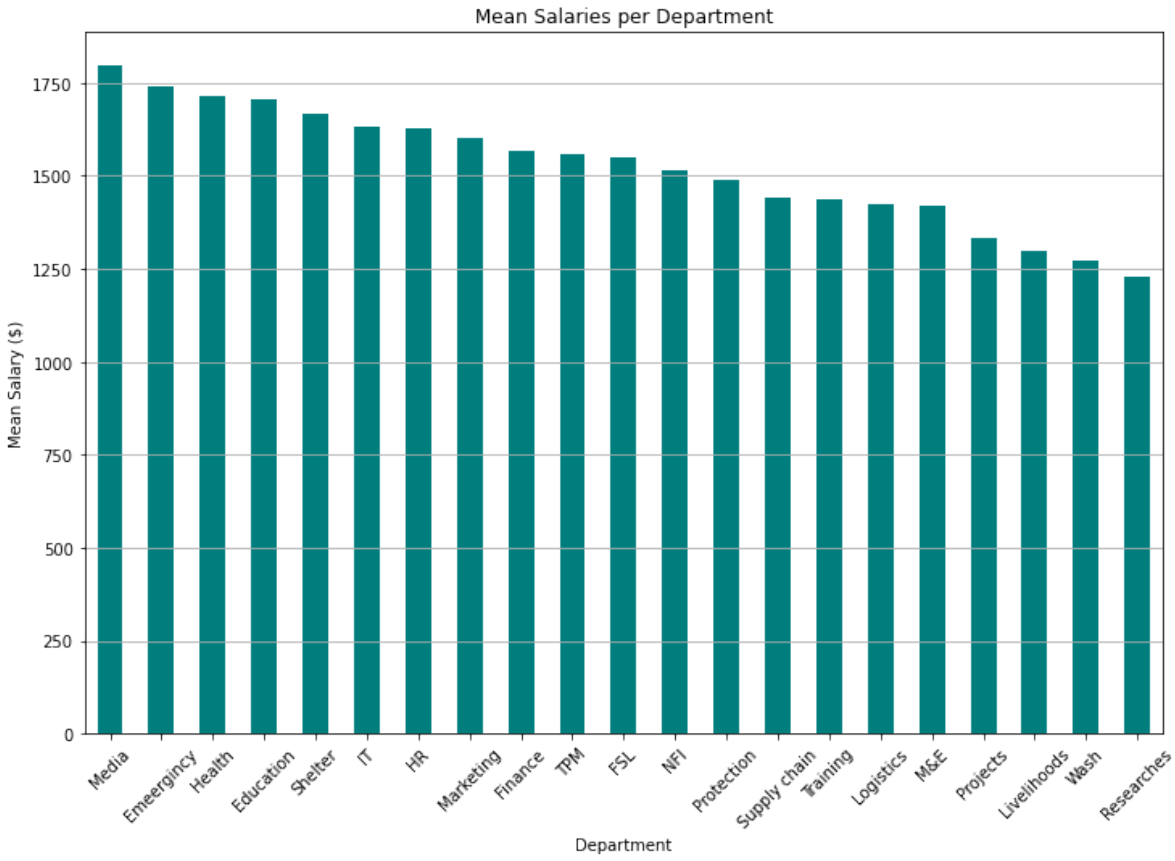
In [40]:
```
df['Location'] = df['Location'].astype('category')
df['Department'] = df['Department'].astype('category')
```

In [39]:
```
df['Department'].value_counts()
```

Out[39]:
```
Department
Protection      151
Logistics        93
Finance          88
FSL              73
Training         59
Marketing        51
IT               44
NFI              38
Wash             21
M&E              19
Shelter          17
Emeergincy       16
Education        14
Media             9
HR                9
Livelihoods       8
TPM               8
Health            7
Projects          5
Researches        5
Supply chain      5
Name: count, dtype: int64
```

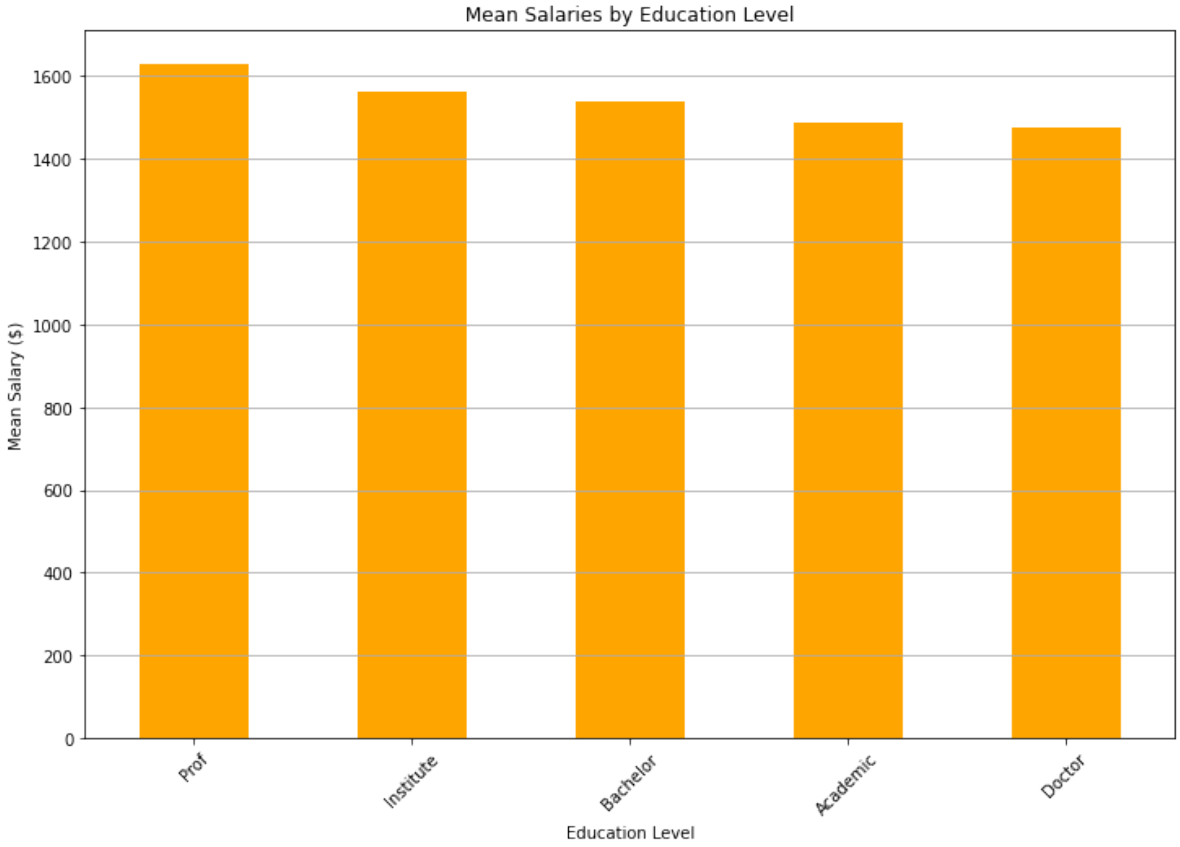
In [42]:
```
df['Salary'] = pd.to_numeric(df['Salary'], errors='coerce')

mean_salary_by_department = df.groupby('Department')['Salary'].mean().dropna()

plt.figure(figsize=(12, 8))
mean_salary_by_department.sort_values(ascending=False).plot(kind='bar', color=
plt.xlabel('Department')
plt.ylabel('Mean Salary ($)')
plt.title('Mean Salaries per Department')
plt.xticks(rotation=45)
plt.grid(axis='y')
plt.show()
```

```
C:\Users\LENOVO\AppData\Local\Temp\ipykernel_10716\3995139720.py:3: FutureWar
ning: The default of observed=False is deprecated and will be changed to True
in a future version of pandas. Pass observed=False to retain current behavior
or observed=True to adopt the future default and silence this warning.
  mean_salary_by_department = df.groupby('Department')['Salary'].mean().dropn
a()
```



- The chart show the distribution of the Salaries and shows the same results that media has the highest mean salaries

In [44]:
```
df['Salary'] = pd.to_numeric(df['Salary'], errors='coerce')

mean_salary_by_education = df.groupby('Education')['Salary'].mean().dropna()

top_education_salaries = mean_salary_by_education.sort_values(ascending=False)

plt.figure(figsize=(12, 8))
top_education_salaries.plot(kind='bar', color='orange')
plt.xlabel('Education Level')
plt.ylabel('Mean Salary ($)')
plt.title('Mean Salaries by Education Level')
plt.xticks(rotation=45)
plt.grid(axis='y')
plt.show()
```

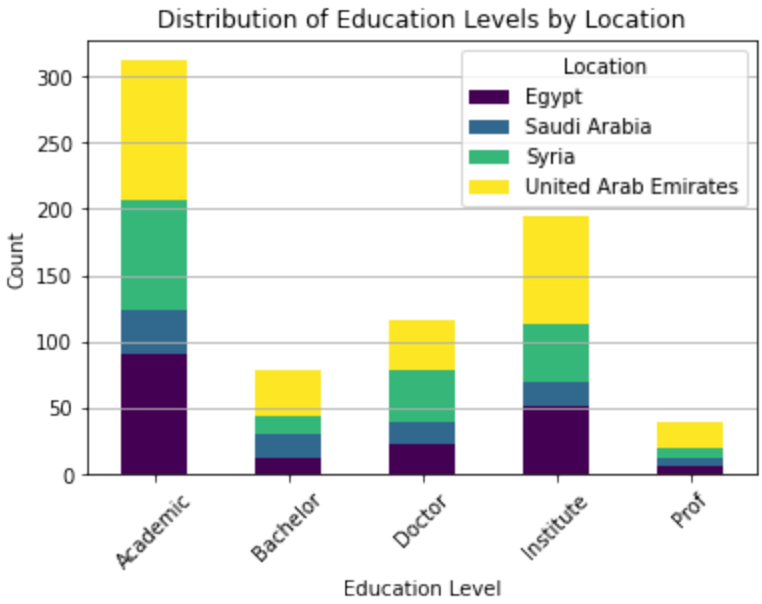


- Prof has the highest mean salaries, more than institue and bachelor. Academic octors is the lowest !
- It might due to the ignorance of governoments for the Academic Staff in universities and their salaries

In [49]:
```
education_counts = df.groupby(['Location', 'Education']).size().reset_index(na

education_pivot = education_counts.pivot(index='Education', columns='Location'
plt.figure(figsize=(14, 8))
education_pivot.plot(kind='bar', stacked=True, colormap='viridis')
plt.xlabel('Education Level')
plt.ylabel('Count')
plt.title('Distribution of Education Levels by Location')
plt.legend(title='Location')
plt.xticks(rotation=45)
plt.grid(axis='y')
plt.show()
```

```
C:\Users\LENOVO\AppData\Local\Temp\ipykernel_10716\1428949604.py:1: FutureWar
ning: The default of observed=False is deprecated and will be changed to True
in a future version of pandas. Pass observed=False to retain current behavior
or observed=True to adopt the future default and silence this warning.
  education_counts = df.groupby(['Location', 'Education']).size().reset_index
(name='Count')
```

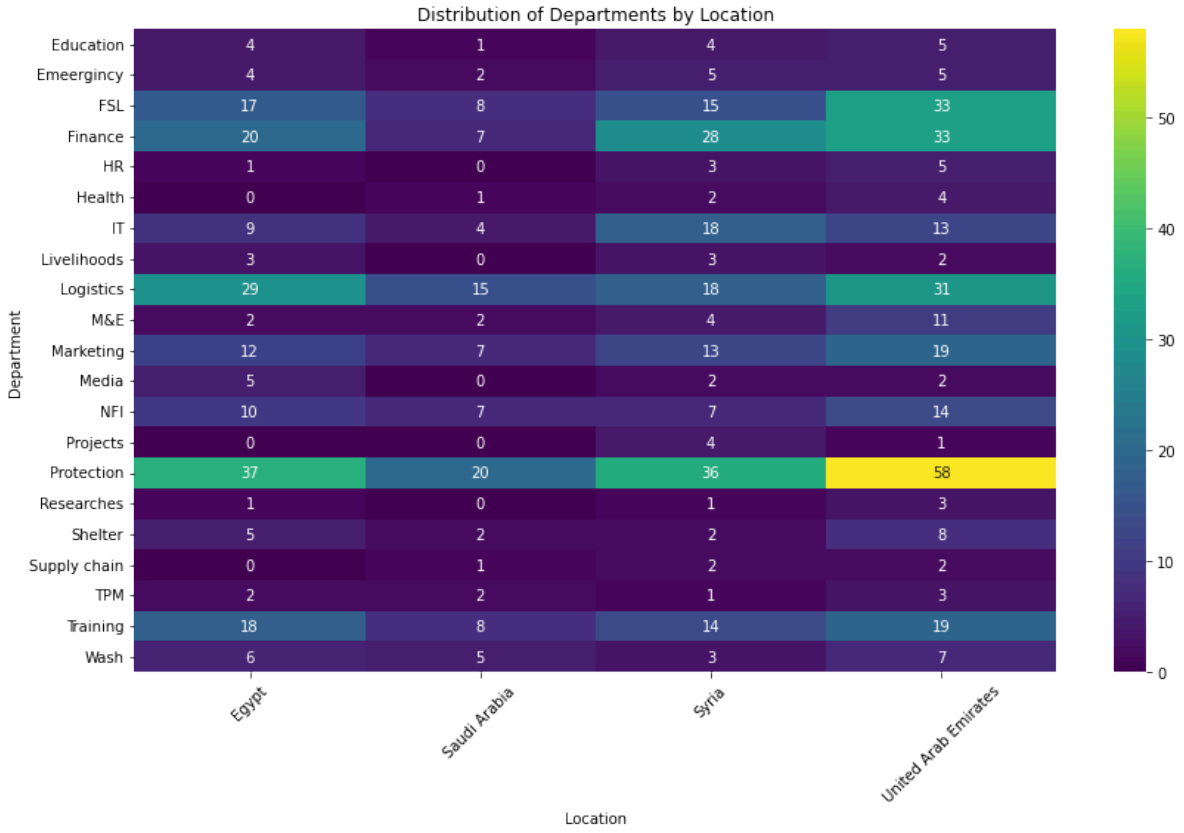
```
<Figure size 1008x576 with 0 Axes>
```



Academic are more tradition in UAE and Egypt. And doctor is shown more in Syria. Being Doctors has the lowest mean salaries. It might be due to their highly distribution in Syria while syria has one of the lowest salaries. Also for academic in spite of being more distributed in Egypt and UAE but has a big portion of them working in Syria. Also UAE has the lowest salaries already. So it affects may shows why this data tells us why Academic and Doctor has the 2 lowest salaries.

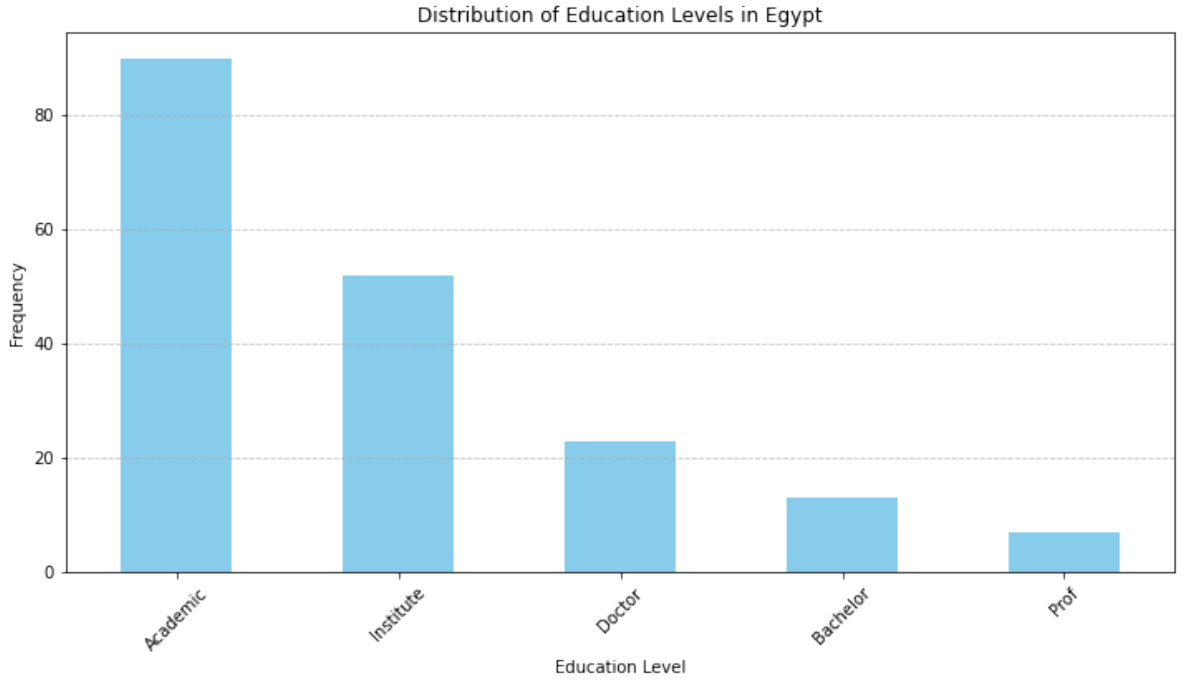
In [50]:
```
department_counts = df.groupby(['Location', 'Department']).size().reset_index(

department_pivot = department_counts.pivot(index='Department', columns='Locati

plt.figure(figsize=(14, 8))
sns.heatmap(department_pivot, annot=True, fmt='g', cmap='viridis', cbar=True)
plt.xlabel('Location')
plt.ylabel('Department')
plt.title('Distribution of Departments by Location')
plt.xticks(rotation=45)
plt.yticks(rotation=0)
plt.show()
```

```
C:\Users\LENOVO\AppData\Local\Temp\ipykernel_10716\2701398049.py:1: FutureWar
ning: The default of observed=False is deprecated and will be changed to True
in a future version of pandas. Pass observed=False to retain current behavior
or observed=True to adopt the future default and silence this warning.
  department_counts = df.groupby(['Location', 'Department']).size().reset_ind
ex(name='Count')
```



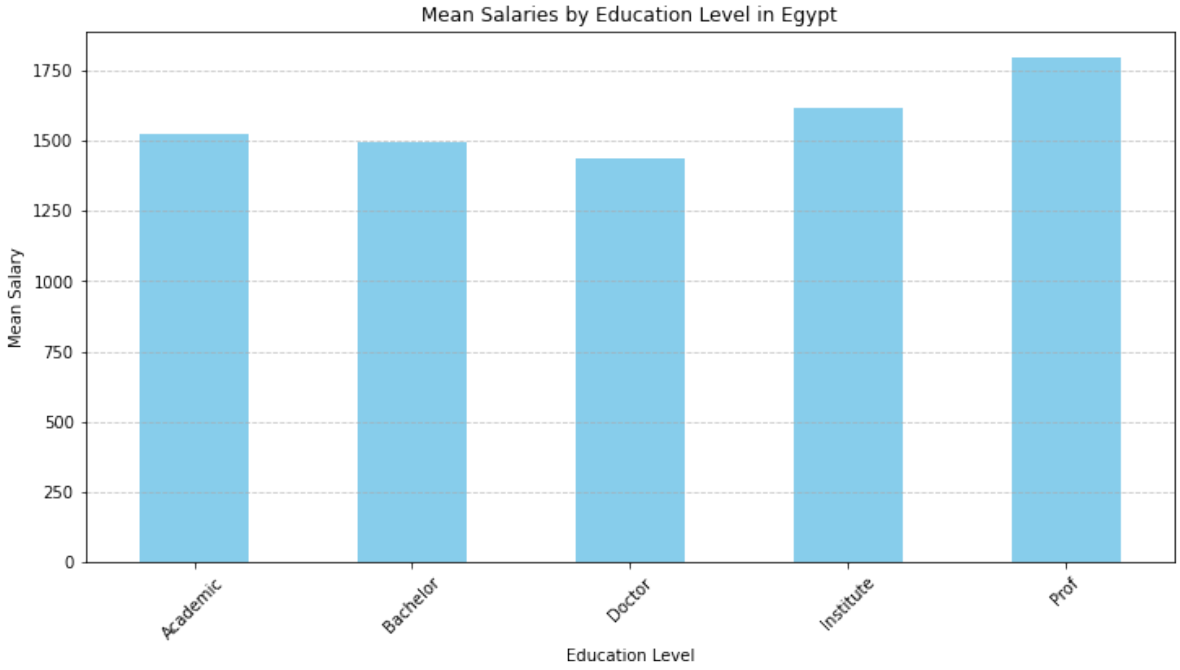
As the protection is the most shown department, It's also the most shown for each country. Specially in UAE.

In [51]:
```
egypt_df = df[df['Location'] == 'Egypt']

education_counts = egypt_df['Education'].value_counts()

plt.figure(figsize=(12, 6))
education_counts.plot(kind='bar', color='skyblue')

plt.xlabel('Education Level')
plt.ylabel('Frequency')
plt.title('Distribution of Education Levels in Egypt')
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



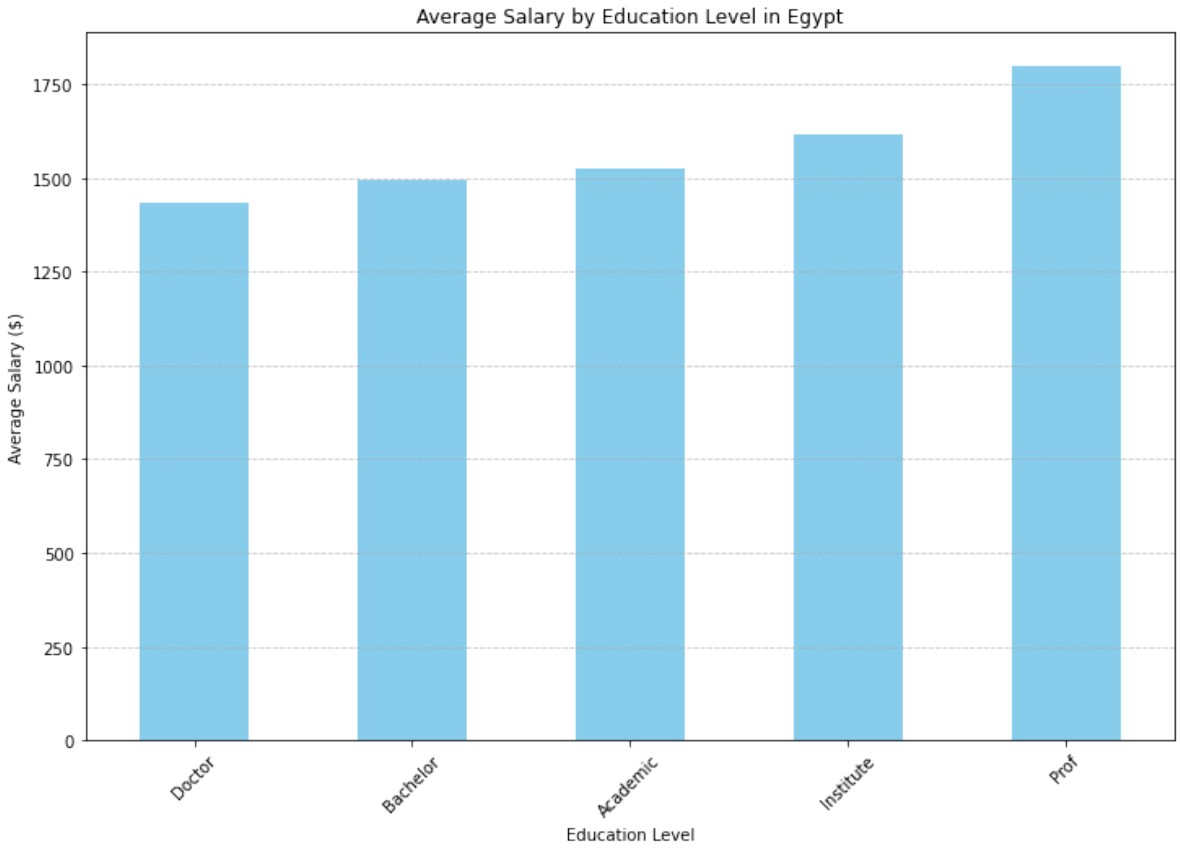
- Academic is the most widely spread education in Egypt among any other education

In [71]:
```
egypt_df = df[df['Location'] == 'Egypt']

education_counts = egypt_df.groupby('Education')['Salary'].mean()

plt.figure(figsize=(12, 6))
education_counts.plot(kind='bar', color='skyblue')

plt.xlabel('Education Level')
plt.ylabel('Mean Salary')
plt.title('Mean Salaries by Education Level in Egypt')
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



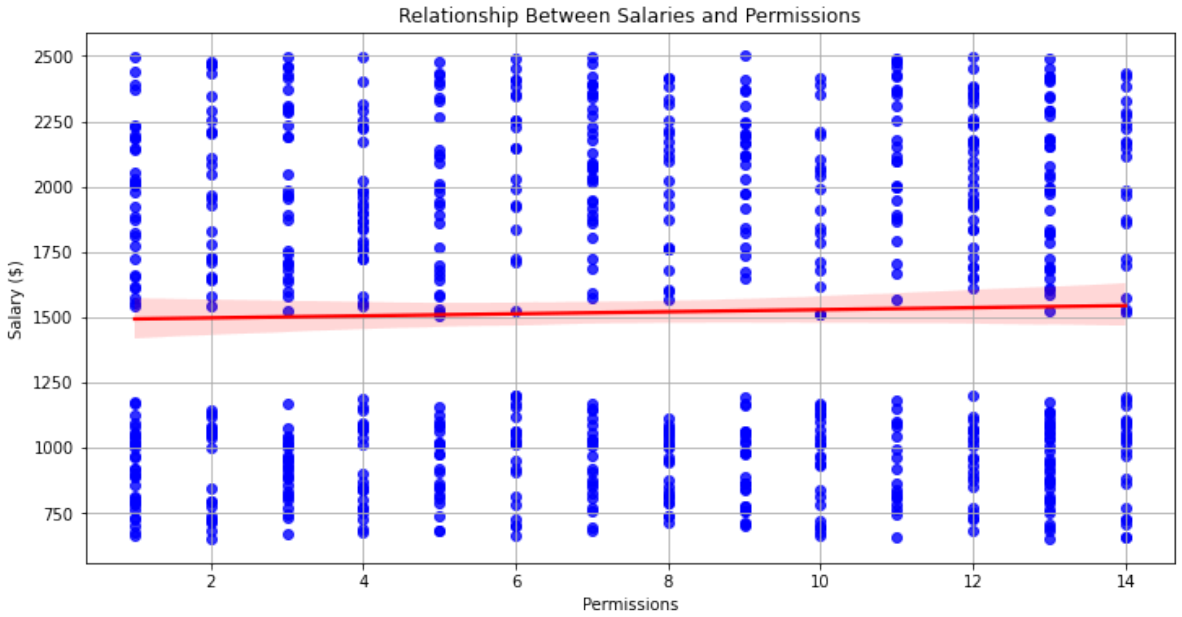
- Salaries are have no outliers and normally distributed for Academic, Institute and prof. But has a skewness in Bachelor and Doctor. That might cause the decreasing in salaries for them
- The skewness in academic salaries reveals that while there are some higher-paying academic positions, the majority of academic salaries are relatively low. This concentration of lower salaries can contribute to the overall perception of lower average salaries in the field.

In [53]:
```
avg_salary_by_education = egypt_df.groupby('Education')['Salary'].mean().sort_

plt.figure(figsize=(12, 8))
avg_salary_by_education.plot(kind='bar', color='skyblue')

plt.xlabel('Education Level')
plt.ylabel('Average Salary ($)')
plt.title('Average Salary by Education Level in Egypt')
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



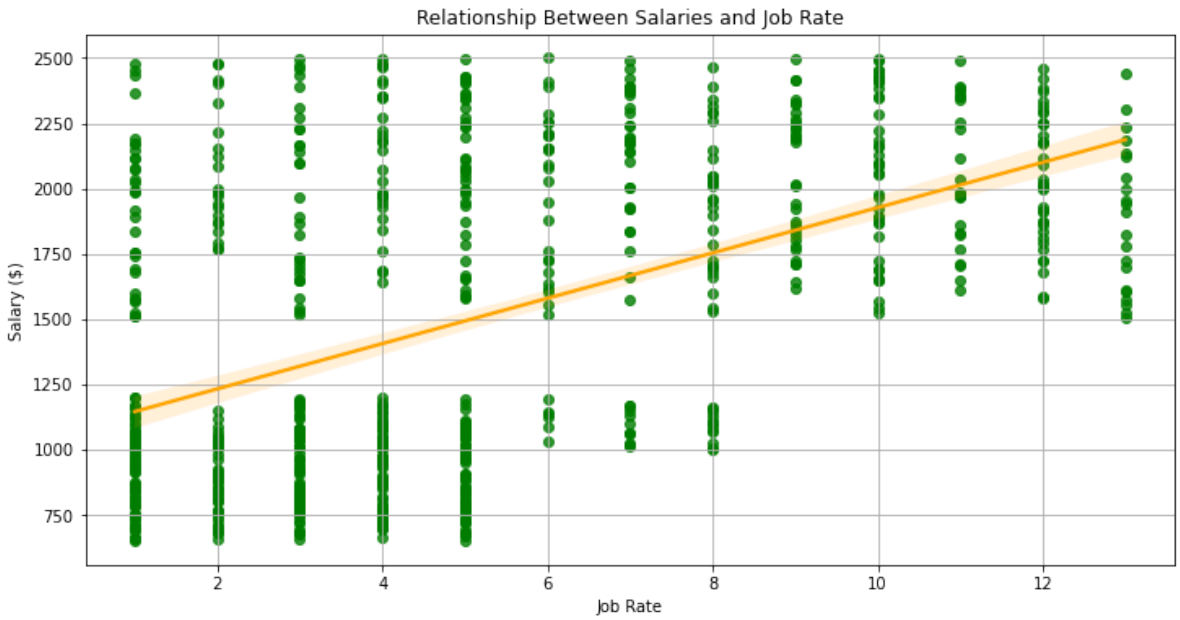
- In spite of being academic are the most shown in Egypt but it doesn't the 1st or 2nd of highest salaries. And it might also be another reason of why academic being one of the lowest salaries. Also doctors have the lowest salaries !

In [82]:
```
plt.figure(figsize=(12, 6))
sns.regplot(x='Permissions', y='Salary', data=df, scatter_kws={'color': 'blue'
plt.xlabel('Permissions')
plt.ylabel('Salary ($)')
plt.title('Relationship Between Salaries and Permissions')
plt.grid(True)
plt.show()
```



The level of permission doesn't have that big effect on the salaries. Just too small effect on salaries.

In [80]:
```
plt.figure(figsize=(12, 6))
sns.regplot(x='Job Rate', y='Salary', data=df, scatter_kws={'color': 'green'},
plt.xlabel('Job Rate')
plt.ylabel('Salary ($)')
plt.title('Relationship Between Salaries and Job Rate')
plt.grid(True)
plt.show()
```



Job rating has a direct relation with salaries. The Higher job rating, the higher salaries

Conclusion

Business Insights

- Regional Salary Discrepancies: Highest Salaries: Egypt has the highest average salaries, indicating a robust job market or higher pay scales in this region. Lowest Salaries: Syria and the UAE offer the lowest average salaries. This could be due to economic conditions, market demand, or differing cost-of-living factors.
- Educational Impact: Academics: Academic professionals, while widely spread, show a skewed salary distribution with a concentration of lower salaries. This suggests that academic roles might be undervalued in terms of compensation. Doctors: Similar to academics, doctors are widely spread but show a broader range of salaries, indicating variability in compensation based on experience, specialization, or location.
- Experience and Salary: Experience Trends: Professionals with 6 years of experience have the lowest average salaries. This anomaly might reflect transitional career phases or market adjustments affecting mid-career professionals.
- Most Frequent Departments: Distribution: Certain departments have a higher frequency across locations. Understanding the departments that are most common can help in talent acquisition and resource allocation.

Technical Recommendations

- Advanced Statistical Analysis: Multivariate Analysis: Conduct multivariate analysis to understand how multiple factors (e.g., permissions, job rates, and education) interact and impact salaries simultaneously. Techniques like Principal Component Analysis (PCA) or Factor Analysis could be useful. Outlier Detection: Implement outlier detection methods to identify unusual salary patterns and assess their impact. This can help in refining salary benchmarks and identifying anomalies.
- Predictive Analytics: Predictive Modeling: Develop predictive models using machine learning techniques to forecast future salary trends based on historical data and influencing factors. Algorithms like Linear Regression, Random Forest, or Gradient Boosting could be employed. Scenario Analysis: Create scenarios to model how changes in factors such as job rates or permissions might affect salaries. This can help in strategic planning and policy formulation. Data Quality and Integrity:

Business Recommendations

- Market Adjustments: Salary Reassessment: For regions like Syria and the UAE, consider revisiting salary structures to improve competitiveness and attract talent.
- Educational Investments: Compensation Review: Review and potentially adjust the compensation for academic roles and other widely spread positions to reflect their value and contributions.
- Experience Management: Career Development: Address the low salary issue for professionals with 6 years of experience by offering career development programs or reassessing salary structures to ensure equitable compensation progression.
- Departmental Focus: Resource Allocation: Allocate resources and tailor recruitment strategies based on the most common departments and their needs across different locations. This structured approach will help in understanding the underlying trends and making data-driven decisions to improve overall salary structures and career development strategies.

In [88]:
```
df.to_csv('new_salaries.csv')
```

In [ ]:
```

```