

ASSIGNMENT-2



5007_ANANTH S <ananth12704@gmail.com>

Hash Agile Interns/Freshers 12-Hour Assignment Task Instructions

1 message

Recruitment <recruitment@hashagile.com>
To: Recruitment <recruitment@hashagile.com>

4 November 2024 at 13:21

Dear Candidate,

Congratulations on progressing to the next stage of the interview process.

Please complete the following assignment:

• Installation Instructions

- a) Read about Elasticsearch: Getting Started with Elasticsearch.
- b) Install Elasticsearch on your local machine.
- c) Start the service in 8989 port instead of the default port
- d) Download Employee Data set from <https://www.kaggle.com/datasets/williamlucas0/employee-sample-data>

• Function Definitions

Using Any of the programming language implement below functions

- a) createCollection(p_collection_name)
- b) indexData(p_collection_name, p_exclude_column): Index the given employee data into the specified collection, excluding the column provided in p_exclude_column.
- c) searchByColumn(p_collection_name, p_column_name, p_column_value): Search within the specified collection for records where the column p_column_name matches the value p_column_value.
- d) getEmpCount(p_collection_name)
- e) delEmpById(p_collection_name, p_employee_id)
- f) getDepFacet(p_collection_name): Retrieve the count of employees grouped by department from the specified collection.

• Function Executions

• Once the functions are implemented, execute the functions in the given order with the parameters mentioned and capture separate full screenshots for each step mentioned below

- a) Var v_nameCollection = 'Hash_<Your Names>'
- b) Var v_phoneCollection = 'Hash_<Your Phone last four digits>'
- c) createCollection(v_nameCollection)
- d) createCollection(v_phoneCollection)
- e) getEmpCount(v_nameCollection)
- f) indexData(v_nameCollection, 'Department')
- g) indexData(v_phoneCollection, 'Gender')
- h) getEmpCount(v_nameCollection)
- i) delEmpById(v_nameCollection, 'E02003')
- j) getEmpCount(v_nameCollection)

Github Link:

https://github.com/ananth127/hash_agile_round_2

Installation Instructions

- a) Read about Elasticsearch: Getting Started with Elasticsearch.
- b) Install Elasticsearch on your local machine.
- c) Start the service in 8989 port instead of the default port
- d) Download Employee Data set
from <https://www.kaggle.com/datasets/williamlucas0/employee-sample-data>

Function Definitions

Using Any of the programming language implement below functions

- a) createCollection(p_collection_name)
- b) indexData(p_collection_name, p_exclude_column): Index the given employee data into the specified collection, excluding the column provided in p_exclude_column.
- c) searchByColumn(p_collection_name, p_column_name, p_column_value): Search within the specified collection for records where the column p_column_name matches the value p_column_value.
- d) getEmpCount(p_collection_name)
- e) delEmpById(p_collection_name, p_employee_id)
- f) getDepFacet(p_collection_name): Retrieve the count of employees grouped by department from the specified collection.

Function Executions

· Once the functions are implemented, execute the functions in the given order with the parameters mentioned and capture separate full screenshots for each step mentioned below

- a) Var v_nameCollection = 'Hash_<Your Name>'
- b) Var v_phoneCollection = 'Hash_<Your Phone last four digits>'
- c) createCollection(v_nameCollection)
- d) createCollection(v_phoneCollection)
- e) getEmpCount(v_nameCollection)
- f) indexData(v_nameCollection, 'Department')
- g) indexData(v_phoneCollection, 'Gender')
- h) getEmpCount(v_nameCollection)
- i) delEmpById (v_nameCollection, 'E02003')
- j) getEmpCount(v_nameCollection)
- k) searchByColumn(v_nameCollection, 'Department', 'IT')
- l) searchByColumn(v_nameCollection, 'Gender', 'Male')
- m) searchByColumn(v_phoneCollection, 'Department', 'IT')
- n) getDepFacet(v_nameCollection)
- o) getDepFacet(v_phoneCollection)

Program:

main.py

```
from elasticsearch import Elasticsearch, helpers, exceptions
import pandas as pd

es = Elasticsearch("http://localhost:8989", verify_certs=False)

try:
    if es.ping():
        print("Connected to Elasticsearch.")
    else:
        print("Failed to connect to Elasticsearch.")
except exceptions.ConnectionError:
    print("Error: Could not connect to Elasticsearch.")
    exit()

name = input("Enter your name: ")
phone = input("Enter your phone number (last four digits): ")

v_nameCollection = f'hash_{name.lower()}'
v_phoneCollection = f'hash_{phone[-4:]}'

def createCollection(p_collection_name):
    try:
        if not es.indices.exists(index=p_collection_name):
            es.indices.create(index=p_collection_name)
            print(f'Collection '{p_collection_name}' created.")
        else:
            print(f'Collection '{p_collection_name}' already exists.")
    except exceptions.ConnectionError as e:
        print(f"Connection error: {e}")
    except exceptions.TransportError as e:
        print(f"Elasticsearch error: {e}")

def indexData(p_collection_name, p_exclude_column):
    try:
        data = pd.read_csv("employee_data.csv", encoding='ISO-8859-1').drop(columns=[p_exclude_column])
        actions = [
            {
                "_index": p_collection_name,
                "_id": row["Employee ID"],
                "_source": row.to_dict()
            }
            for _, row in data.iterrows()
        ]
        helpers.bulk(es, actions)
```

```

        print(f"Data indexed into collection '{p_collection_name}' excluding '{p_exclude_column}'
column.")
    except FileNotFoundError:
        print("CSV file not found.")
    except exceptions.ConnectionError as e:
        print(f"Connection error: {e}")
    except Exception as e:
        print(f"An error occurred: {e}")

def searchByColumn(p_collection_name, p_column_name, p_column_value):
    try:
        query = {
            "query": {
                "match": {
                    p_column_name: p_column_value
                }
            }
        }
        res = es.search(index=p_collection_name, body=query)
        print(f"Search results for {p_column_name}={p_column_value} in '{p_collection_name}':")
        return res['hits']['hits']
    except exceptions.NotFoundError:
        print(f"Collection '{p_collection_name}' not found.")
    except exceptions.ConnectionError as e:
        print(f"Connection error: {e}")

def getEmpCount(p_collection_name):
    try:
        res = es.count(index=p_collection_name)
        print(f"Employee count in '{p_collection_name}': {res['count']}")
        return res['count']
    except exceptions.ConnectionError as e:
        print(f"Connection error: {e}")

def delEmpById(p_collection_name, p_employee_id):
    try:
        res = es.delete(index=p_collection_name, id=p_employee_id, ignore=[404])
        print(f"Deleted employee ID {p_employee_id} from '{p_collection_name}'.")
        return res
    except exceptions.NotFoundError:
        print(f"Employee ID {p_employee_id} not found.")
    except exceptions.ConnectionError as e:
        print(f"Connection error: {e}")

def getDepFacet(p_collection_name):
    try:
        query = {
            "size": 0,
            "aggs": {
                "departments": {
                    "terms": {

```

```

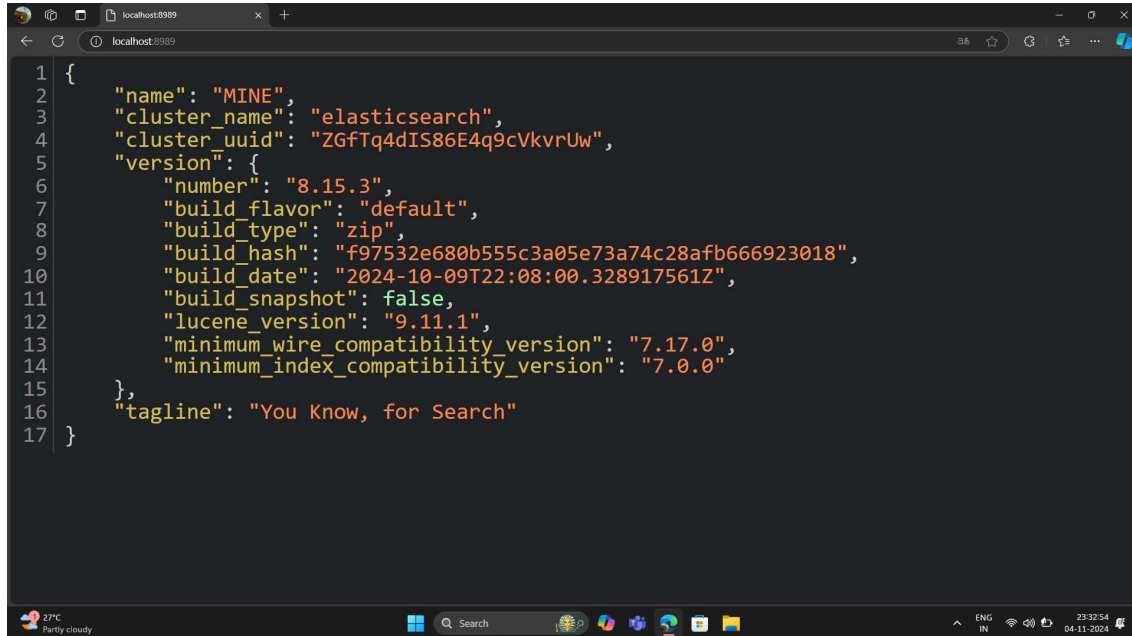
        "field": "Department.keyword"
    }
}
}
}
res = es.search(index=p_collection_name, body=query)
print("Employee count grouped by department:")
return res['aggregations']['departments']['buckets']
except exceptions.ConnectionError as e:
    print(f"Connection error: {e}")

createCollection(v_nameCollection)
createCollection(v_phoneCollection)
getEmpCount(v_nameCollection)
indexData(v_nameCollection, 'Department')
indexData(v_phoneCollection, 'Gender')
getEmpCount(v_nameCollection)
delEmpById(v_nameCollection, 'E02003')
getEmpCount(v_nameCollection)
print(searchByColumn(v_nameCollection, 'Department', 'IT'))
print(searchByColumn(v_nameCollection, 'Gender', 'Male'))
print(searchByColumn(v_phoneCollection, 'Department', 'IT'))
print(getDepFacet(v_nameCollection))
print(getDepFacet(v_phoneCollection))

```

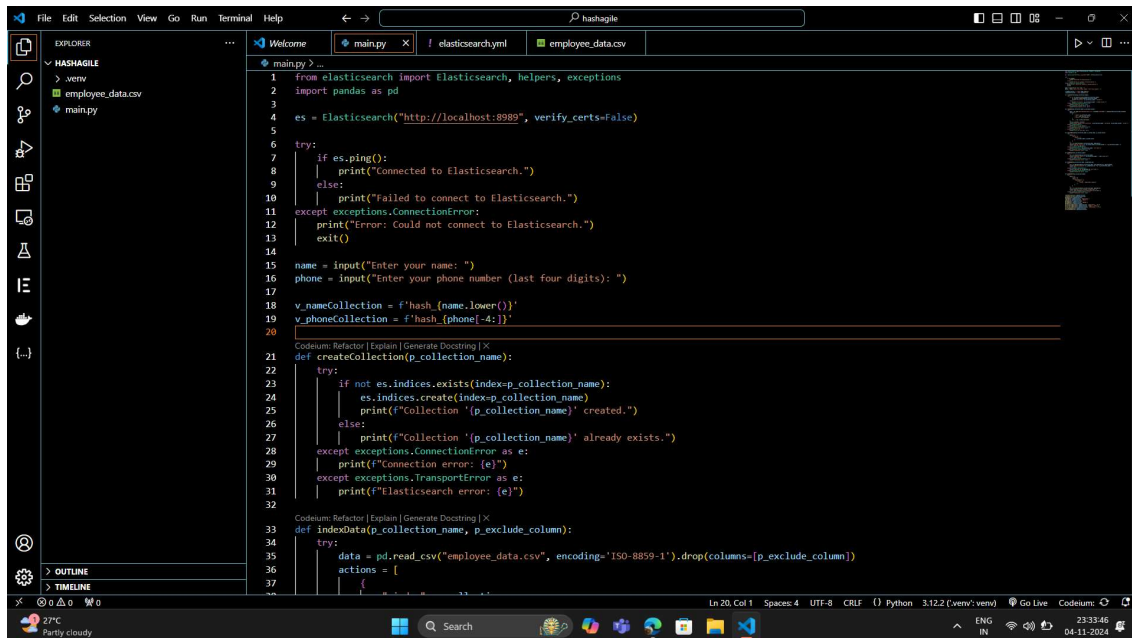
Service running on port 8989:

<http://localhost:8989>



A screenshot of a web browser window displaying a JSON response from the URL `localhost:8989`. The JSON object contains information about an Elasticsearch cluster named "elasticsearch".

```
1 {
2   "name": "MINE",
3   "cluster_name": "elasticsearch",
4   "cluster_uuid": "ZGfTq4dIS86E4q9cVkvUw",
5   "version": {
6     "number": "8.15.3",
7     "build_flavor": "default",
8     "build_type": "zip",
9     "build_hash": "f97532e680b555c3a05e73a74c28afb666923018",
10    "build_date": "2024-10-09T22:08:00.328917561Z",
11    "build_snapshot": false,
12    "lucene_version": "9.11.1",
13    "minimum_wire_compatibility_version": "7.17.0",
14    "minimum_index_compatibility_version": "7.0.0"
15  },
16   "tagline": "You Know, for Search"
17 }
```



A screenshot of a VS Code editor window showing a Python script named `main.py`. The script is designed to connect to an Elasticsearch instance at `localhost:8989`, verify the connection, and create an index if it doesn't exist. It also reads data from a CSV file and drops specific columns.

```
1 from elasticsearch import Elasticsearch, helpers, exceptions
2 import pandas as pd
3
4 es = Elasticsearch("http://localhost:8989", verify_certs=False)
5
6 try:
7     if es.ping():
8         print("Connected to Elasticsearch.")
9     else:
10        print("Failed to connect to Elasticsearch.")
11 except exceptions.ConnectionError:
12    print("Error: Could not connect to Elasticsearch.")
13    exit()
14
15 name = input("Enter your name: ")
16 phone = input("Enter your phone number (last four digits): ")
17
18 v_nameCollection = f'hash_{name.lower()}'
19 v_phoneCollection = f'hash_{phone[-4:]}'
20
21 def createCollection(p_collection_name):
22     try:
23         if not es.indices.exists(index=p_collection_name):
24             es.indices.create(index=p_collection_name)
25             print(f"Collection '{p_collection_name}' created.")
26         else:
27             print(f"Collection '{p_collection_name}' already exists.")
28     except exceptions.ConnectionError as e:
29         print(f"Connection error: {e}")
30     except exceptions.TransportError as e:
31         print(f"Elasticsearch error: {e}")
32
33 def indexData(p_collection_name, p_exclude_column):
34     try:
35         data = pd.read_csv("employee_data.csv", encoding='ISO-8859-1').drop(columns=[p_exclude_column])
36         actions = [
37             {
38                 "_source": data.to_dict(orient='records')
```

>>Enter your phone number : Hash_7127

```
File Edit Selection View Go Run Terminal Help hashaghe
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL COMMENTS PORTS
res = es.delete(index=p_collection_name, id=p_employee_id, ignore=[404])
Deleted employee ID E02803 from 'hash_ananth'.
Employee count in 'hash_ananth': 53
Search results for Department-II in 'hash_ananth':
[]
Search results for Gender=Male in 'hash_ananth':
[{'index': 'hash_ananth', 'id': 'E02807', 'score': 0.6747981, 'source': {'Employee ID': 'E02807', 'Full Name': 'Erna Wu', 'Job Title': 'Network Administrator', 'Business Unit': 'Manufacturing', 'Gender': 'Male', 'Ethnicity': 'Asian', 'Age': 62.0, 'Hire Date': '4/22/2004', 'Annual Salary': '$66,227', 'Bonus %': '0%', 'Country': 'United States', 'City': 'Phoenix', 'Exit Date': '2/14/2014'}}], {'index': 'hash_ananth', 'id': 'E02808', 'score': 0.6747981, 'source': {'Employee ID': 'E02808', 'Full Name': 'Jacob Cheng', 'Job Title': 'Director', 'Business Unit': 'Manufacturing', 'Gender': 'Male', 'Ethnicity': 'Chinese', 'Age': 51.25, 'Hire Date': '11/24/1999', 'Annual Salary': '$155,889', 'Bonus %': '1%', 'Country': 'United States', 'City': 'Columbus', 'Exit Date': '11/27/2082'}], {'index': 'hash_ananth', 'id': 'E02105', 'score': 0.6747981, 'source': {'Employee ID': 'E02105', 'Full Name': 'Ezekiel Brown', 'Job Title': 'Manager', 'Business Unit': 'Corporate', 'Gender': 'Male', 'Ethnicity': 'Caucasian', 'Age': 27.0, 'Hire Date': '3/14/2022', 'Annual Salary': '$120,315', 'Bonus %': '8%', 'Country': 'United States', 'City': 'Miami', 'Exit Date': '7/17/2022'}], {'index': 'hash_ananth', 'id': 'E02113', 'score': 0.6747981, 'source': {'Employee ID': 'E02113', 'Full Name': 'Roman Liang', 'Job Title': 'Sr. Analyst', 'Business Unit': 'Manufacturing', 'Gender': 'Male', 'Ethnicity': 'Asian', 'Age': 28.0, 'Hire Date': '5/4/2021', 'Annual Salary': '$45,206', 'Bonus %': '0%', 'Country': 'China', 'City': 'Chongqing', 'Exit Date': '7/8/2022'}], {'index': 'hash_ananth', 'id': 'E02167', 'score': 0.6747981, 'source': {'Employee ID': 'E02167', 'Full Name': 'Nolan Howard', 'Job Title': 'Quality Engineer', 'Business Unit': 'Research & Development', 'Gender': 'Male', 'Ethnicity': 'Caucasian', 'Age': 26.0, 'Hire Date': '4/18/2022', 'Annual Salary': '$52,963', 'Bonus %': '0%', 'Country': 'United States', 'City': 'Austin', 'Exit Date': '6/23/2022'}], {'index': 'hash_ananth', 'id': 'E02175', 'score': 0.6747981, 'source': {'Employee ID': 'E02175', 'Full Name': 'Christopher Robinson', 'Job Title': 'Vice President', 'Business Unit': 'Specialty Products', 'Gender': 'Male', 'Ethnicity': 'Caucasian', 'Age': 35.0, 'Hire Date': '7/11/2016', 'Annual Salary': '$254,287', 'Bonus %': '31%', 'Country': 'United States', 'City': 'Austin', 'Exit Date': '10/17/2022'}], {'index': 'hash_ananth', 'id': 'E02186', 'score': 0.6747981, 'source': {'Employee ID': 'E02186', 'Full Name': 'Logan Reyes', 'Job Title': 'Quality Engineer', 'Business Unit': 'Corporate', 'Gender': 'Male', 'Ethnicity': 'Latino', 'Age': 62.0, 'Hire Date': '4/10/1995', 'Annual Salary': '$70,251', 'Bonus %': '0%', 'Country': 'Brazil', 'City': 'Sao Paulo', 'Exit Date': '3/19/2020'}], {'index': 'hash_ananth', 'id': 'E02202', 'score': 0.6747981, 'source': {'Employee ID': 'E02202', 'Full Name': 'Charles Ramos', 'Job Title': 'Sr. Analyst', 'Business Unit': 'Corporate', 'Gender': 'Male', 'Ethnicity': 'Latino', 'Age': 57.0, 'Hire Date': '4/4/2010', 'Annual Salary': '$74,597', 'Bonus %': '0%', 'Country': 'United States', 'City': 'Austin', 'Exit Date': '1/30/2016'}], {'index': 'hash_ananth', 'id': 'E02220', 'score': 0.6747981, 'source': {'Employee ID': 'E02220', 'Full Name': 'Logan Reyes', 'Job Title': 'Quality Engineer', 'Business Unit': 'Corporate', 'Gender': 'Male', 'Ethnicity': 'Latino', 'Age': 62.0, 'Hire Date': '4/10/1995', 'Annual Salary': '$78,251', 'Bonus %': '0%', 'Country': 'Brazil', 'City': 'Sao Paulo', 'Exit Date': '3/19/2020'}], {'index': 'hash_ananth', 'id': 'E02220', 'score': 0.6747981, 'source': {'Employee ID': 'E02220', 'Full Name': 'Charles Ramos', 'Job Title': 'Sr. Analyst', 'Business Unit': 'Corporate', 'Gender': 'Male', 'Ethnicity': 'Latino', 'Age': 57.0, 'Hire Date': '4/4/2010', 'Annual Salary': '$74,597', 'Bonus %': '0%', 'Country': 'United States', 'City': 'Austin', 'Exit Date': '1/30/2016'}], {'index': 'hash_ananth', 'id': 'E02220', 'score': 0.6747981, 'source': {'Employee ID': 'E02220', 'Full Name': 'Charles Ramos', 'Job Title': 'Sr. Analyst', 'Business Unit': 'Corporate', 'Gender': 'Male', 'Ethnicity': 'Latino', 'Age': 57.0, 'Hire Date': '4/4/2010', 'Annual Salary': '$74,597', 'Bonus %': '0%', 'Country': 'United States', 'City': 'Austin', 'Exit Date': '1/30/2016'}], {'index': 'hash_ananth', 'id': 'E02220', 'score': 0.6747981, 'source': {'Employee ID': 'E02220', 'Full Name': 'Charles Ramos', 'Job Title': 'Sr. Analyst', 'Business Unit': 'Corporate', 'Gender': 'Male', 'Ethnicity': 'Latino', 'Age': 57.0, 'Hire Date': '4/4/2010', 'Annual Salary': '$74,597', 'Bonus %': '0%', 'Country': 'United States', 'City': 'Austin', 'Exit Date': '1/30/2016'}], {'index': 'hash_ananth', 'id': 'E02220', 'score': 0.6747981, 'source': {'Employee ID': 'E02220', 'Full Name': 'Charles Ramos', 'Job Title': 'Sr. Analyst', 'Business Unit': 'Corporate', 'Gender': 'Male', 'Ethnicity': 'Latino', 'Age': 57.0, 'Hire Date': '4/4/2010', 'Annual Salary': '$74,597', 'Bonus %': '0%', 'Country': 'United States', 'City': 'Austin', 'Exit Date': '1/30/2016'}], {'index': 'hash_ananth', 'id': 'E02220', 'score': 0.6747981, 'source': {'Employee ID': 'E02220', 'Full Name': 'Charles Ramos', 'Job Title': 'Sr. Analyst', 'Business Unit': 'Corporate', 'Gender': 'Male', 'Ethnicity': 'Latino', 'Age': 57.0, 'Hire Date': '4/4/2010', 'Annual Salary': '$74,597', 'Bonus %': '0%', 'Country': 'United States', 'City': 'Austin', 'Exit Date': '1/30/2016'}], {'index': 'hash_ananth', 'id': 'E02220', 'score': 0.6747981, 'source': {'Employee ID': 'E02220', 'Full Name': 'Charles Ramos', 'Job Title': 'Sr. Analyst', 'Business Unit': 'Corporate', 'Gender': 'Male', 'Ethnicity': 'Latino', 'Age': 57.0, 'Hire Date': '4/4/2010', 'Annual Salary': '$74,597', 'Bonus %': '0%', 'Country': 'United States', 'City': 'Austin', 'Exit Date': '1/30/2016'}], {'index': 'hash_ananth', 'id': 'E02220', 'score': 0.6747981, 'source': {'Employee ID': 'E02220', 'Full Name': 'Charles Ramos', 'Job Title': 'Sr. Analyst', 'Business Unit': 'Corporate', 'Gender': 'Male', 'Ethnicity': 'Latino', 'Age': 57.0, 'Hire Date': '4/4/2010', 'Annual Salary': '$74,597', 'Bonus %': '0%', 'Country': 'United States', 'City': 'Austin', 'Exit Date': '1/30/2016'}], {'index': 'hash_ananth', 'id': 'E02220', 'score': 0.6747981, 'source': {'Employee ID': 'E02220', 'Full Name': 'Charles Ramos', 'Job Title': 'Sr. Analyst', 'Business Unit': 'Corporate', 'Gender': 'Male', 'Ethnicity': 'Latino', 'Age': 57.0, 'Hire Date': '4/4/2010', 'Annual Salary': '$74,597', 'Bonus %': '0%', 'Country': 'United States', 'City': 'Austin', 'Exit Date': '1/30/2016'}], {'index': 'hash_ananth', 'id': 'E02220', 'score': 0.6747981, 'source': {'Employee ID': 'E02220', 'Full Name': 'Charles Ramos', 'Job Title': 'Sr. Analyst', 'Business Unit': 'Corporate', 'Gender': 'Male', 'Ethnicity': 'Latino', 'Age': 57.0, 'Hire Date': '4/4/2010', 'Annual Salary': '$74,597', 'Bonus %': '0%', 'Country': 'United States', 'City': 'Austin', 'Exit Date': '1/30/2016'}], {'index': 'hash_ananth', 'id': 'E02220', 'score': 0.6747981, 'source': {'Employee ID': 'E02220', 'Full Name': 'Charles Ramos', 'Job Title': 'Sr. Analyst', 'Business Unit': 'Corporate', 'Gender': 'Male', 'Ethnicity': 'Latino', 'Age': 57.0, 'Hire Date': '4/4/2010', 'Annual Salary': '$74,597', 'Bonus %': '0%', 'Country': 'United States', 'City': 'Austin', 'Exit Date': '1/30/2016'}], {'index': 'hash_ananth', 'id': 'E02220', 'score': 0.6747981, 'source': {'Employee ID': 'E02220', 'Full Name': 'Charles Ramos', 'Job Title': 'Sr. Analyst', 'Business Unit': 'Corporate', 'Gender': 'Male', 'Ethnicity': 'Latino', 'Age': 57.0, 'Hire Date': '4/4/2010', 'Annual Salary': '$74,597', 'Bonus %': '0%', 'Country': 'United States', 'City': 'Austin', 'Exit Date': '1/30/2016'}], {'index': 'hash_ananth', 'id': 'E02220', 'score': 0.6747981, 'source': {'Employee ID': 'E02220', 'Full Name': 'Charles Ramos', 'Job Title': 'Sr. Analyst', 'Business Unit': 'Corporate', 'Gender': 'Male', 'Ethnicity': 'Latino', 'Age': 57.0, 'Hire Date': '4/4/2010', 'Annual Salary': '$74,597', 'Bonus %': '0%', 'Country': 'United States', 'City': 'Austin', 'Exit Date': '1/30/2016'}], {'index': 'hash_ananth', 'id': 'E02220', 'score': 0.6747981, 'source': {'Employee ID': 'E02220', 'Full Name': 'Charles Ramos', 'Job Title': 'Sr. Analyst', 'Business Unit': 'Corporate', 'Gender': 'Male', 'Ethnicity': 'Latino', 'Age': 57.0, 'Hire Date': '4/4/2010', 'Annual Salary': '$74,597', 'Bonus %': '0%', 'Country': 'United States', 'City': 'Austin', 'Exit Date': '1/30/2016'}], {'index': 'hash_ananth', 'id': 'E02220', 'score': 0.6747981, 'source': {'Employee ID': 'E02220', 'Full Name': 'Charles Ramos', 'Job Title': 'Sr. Analyst', 'Business Unit': 'Corporate', 'Gender': 'Male', 'Ethnicity': 'Latino', 'Age': 57.0, 'Hire Date': '4/4/2010', 'Annual Salary': '$74,597', 'Bonus %': '0%', 'Country': 'United States', 'City': 'Austin', 'Exit Date': '1/30/2016'}], {'index': 'hash_ananth', 'id': 'E02220', 'score': 0.6747981, 'source': {'Employee ID': 'E02220', 'Full Name': 'Charles Ramos', 'Job Title': 'Sr. Analyst', 'Business Unit': 'Corporate', 'Gender': 'Male', 'Ethnicity': 'Latino', 'Age': 57.0, 'Hire Date': '4/4/2010', 'Annual Salary': '$74,597', 'Bonus %': '0%', 'Country': 'United States', 'City': 'Austin', 'Exit Date': '1/30/2016'}], {'index': 'hash_ananth', 'id': 'E02220', 'score': 0.6747981, 'source': {'Employee ID': 'E02220', 'Full Name': 'Charles Ramos', 'Job Title
```