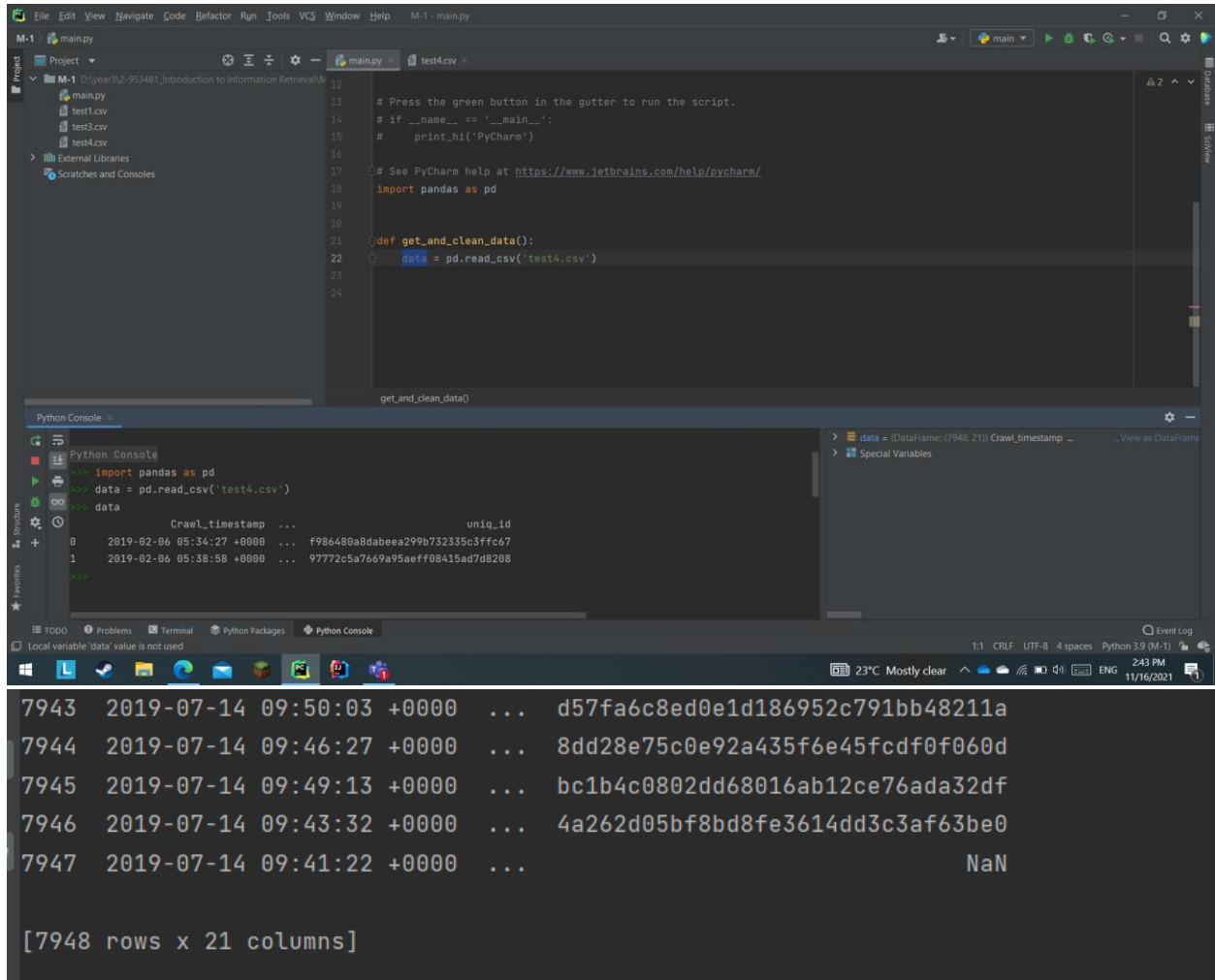


Show the evidence that you have be able to follow (codes and running results) the walkthrough up the page#22 of the slide.



The screenshot shows the PyCharm IDE interface. The main editor window displays a Python script named `main.py` with the following code:

```
12
13 # Press the green button in the gutter to run the script.
14 # if __name__ == '__main__':
15 #     print_hi('PyCharm')
16
17 # See PyCharm help at https://www.jetbrains.com/help/pycharm/
18 import pandas as pd
19
20
21 def get_and_clean_data():
22     data = pd.read_csv('test4.csv')
23
24
```

The Python Console at the bottom shows the execution of the code:

```
>>> import pandas as pd
>>> data = pd.read_csv('test4.csv')
>>> data
```

The console output displays a DataFrame with 2 rows and 4 columns: `Crawl_timestamp`, `...`, `...`, and `uniq_id`.

	Crawl_timestamp	uniq_id
0	2019-02-06 05:34:27 +0000	...	f986480a8dabaea299b732335c3ffc67	
1	2019-02-06 05:38:58 +0000	...	97772c5a7669a95aef08415ad7d8288	

The console also shows the output of the `get_and_clean_data()` function, which is a DataFrame with 7948 rows and 21 columns:

```
> data = (DataFrame: (7948, 21)) Crawl_timestamp ... ..View as DataFrame
> Special Variables
```

The bottom status bar indicates the file encoding is UTF-8, the line ending is CRLF, and the Python version is 3.9 (M-1).

```
7943 2019-07-14 09:50:03 +0000 ... d57fa6c8ed0e1d186952c791bb48211a
7944 2019-07-14 09:46:27 +0000 ... 8dd28e75c0e92a435f6e45fcd0f060d
7945 2019-07-14 09:49:13 +0000 ... bc1b4c0802dd68016ab12ce76ada32df
7946 2019-07-14 09:43:32 +0000 ... 4a262d05bf8bd8fe3614dd3c3af63be0
7947 2019-07-14 09:41:22 +0000 ... NaN

[7948 rows x 21 columns]
```

head () and markdown (): list the top 10 of this data and assign the table for them.

```
print(data.head(10).to_markdown())
```

	Crawl_timestamp	url
0	2019-02-06 05:34:27 +0000	https://www.careerbuilder.com/job/J3W7NK6NM05PGMKPC9W
1	2019-02-06 05:38:58 +0000	https://www.dice.com/jobs/detail/C%2523-Lead-Software-Developer-3coast-Middletown-NJ-07748/
2	2019-02-06 05:31:44 +0000	https://www.dice.com/jobs/detail/Senior-Software-Developer-s.com-Hoboken-NJ-07030/cxcrstff/
3	2019-02-06 05:27:30 +0000	https://www.dice.com/jobs/detail/Senior-Software-Developer-Mitchell-Martin%252C-Inc.-Hoboken-NJ-07030/cxcrstff/
4	2019-02-06 05:37:30 +0000	https://www.dice.com/jobs/detail/C%2523-Lead-Software-Developer-3coast-Philadelphia-PA-19255/
5	2019-02-06 05:47:13 +0000	https://www.dice.com/jobs/detail/Software-Developer-3-Kforce-Technology-Staffing-Mountain-V
6	2019-02-06 05:53:38 +0000	https://www.dice.com/jobs/detail/C%252B%252B-Software-Developer-SigmaWay-San-Jose-CA-95101/
7	2019-02-06 05:55:55 +0000	https://www.dice.com/jobs/detail/C%252B%252B-Software-Developer-%2526%252345-Local-W2-only-1
8	2019-02-06 05:58:22 +0000	https://www.careerbuilder.com/job/J3W4VB6RVDK8W3C5CS5
9	2019-02-06 05:45:09 +0000	https://www.careerbuilder.com/job/J3N4BD6VPK0SBGR2P0J

job_title	category	company_name
Sr. Software Developer	architecture and engineering	Aerotek
C# Lead Software Developer	nan	3coast
Senior Software Developer	nan	s.com
Senior Software Developer	nan	Mitchell Martin, Inc.
C# Lead Software Developer	nan	3coast
Software Developer 3	nan	Kforce Technology Staffing
C++ Software Developer	nan	SigmaWay
C++ Software Developer - Local W2 only	nan	Talent Space, Inc.
Software Developer - .Net	architecture and engineering	National General Insurance
.Net Software Developer	architecture and engineering	Robert Half Technology

regex= ^job: find the sequence after “job”.

	^	Begins with	^where
<pre>>>> print(data.filter(regex='^job').head(10).to_markdown())</pre>			
	job_title	job_description	
---- :----- :----- :-----			
0	Sr. Software Developer	The chosen Sr. Software Developer will be part of a larger engineering	
1	C# Lead Software Developer	Position: C# Lead Software Developer Location: Middletown, NJ C	
2	Senior Software Developer	Senior Software Developer Hoboken, NJ Starts as 9-12 month contract - p	
3	Senior Software Developer	Our client, a multinational publishing and education company, is seekin	
4	C# Lead Software Developer	Position: C# Lead Software Developer Location: Philadelphia, PA	
5	Software Developer 3	RESPONSIBILITIES: Kforce has a client seeking a Software Developer 3 in	
6	C++ Software Developer	Apply by Email/Direct Application at udit.sharma@sigmaway.org C++ Sof	
7	C++ Software Developer - Local W2 only	Talent Space Inc is looking for C++ Software Developer in Sunnyvale. Th	
8	Software Developer - .Net	Primary Purpose: Development and maintenance of software applications b	
9	.Net Software Developer	Ref ID: 02760-0010838185 Classification: Software Engineer Compensation	
<pre>>>> print(data.filter(regex='job').head(10).to_markdown())</pre>			
	job_title	job_description	
---- :----- :----- :-----			
0	Sr. Software Developer	The chosen Sr. Software Developer will be part of a larger engineering team dev	
1	C# Lead Software Developer	Position: C# Lead Software Developer Location: Middletown, NJ Compensat	
2	Senior Software Developer	Senior Software Developer Hoboken, NJ Starts as 9-12 month contract - possibili	
3	Senior Software Developer	Our client, a multinational publishing and education company, is seeking a Seni	
4	C# Lead Software Developer	Position: C# Lead Software Developer Location: Philadelphia, PA Compens	
5	Software Developer 3	RESPONSIBILITIES: Kforce has a client seeking a Software Developer 3 in Mountai	
6	C++ Software Developer	Apply by Email/Direct Application at udit.sharma@sigmaway.org C++ Software De	
7	C++ Software Developer - Local W2 only	Talent Space Inc is looking for C++ Software Developer in Sunnyvale. The hiring	
8	Software Developer - .Net	Primary Purpose: Development and maintenance of software applications built usi	
9	.Net Software Developer	Ref ID: 02760-0010838185 Classification: Software Engineer Compensation: DOE PO	

Declare new variable to assign the job_description and show the result

```
>>> description = data['job_description']
>>> description
0      The chosen Sr. Software Developer will be part...
1      Position: C# Lead Software Developer Locat...
2      Senior Software Developer Hoboken, NJ Starts a...
3      Our client, a multinational publishing and edu...
4      Position: C# Lead Software Developer Locat...
...
7943    Career Evolutions is searching for a Junior So...
7944    RESPONSIBILITIES:Kforce has a client that is s...
7945    Mid Level Software Developer, Dahlgren, Virgin...
7946    This company is looking for a full stack devel...
7947    Experienced .Net Web Developers-our client has...
Name: job_description, Length: 7948, dtype: object
```

Use lambda to clear the punctuation that mean any space, comma, dot and so on in excel file.

apply: for each entry in description

punctuation: '!"#\$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'

```
>>> import string
>>> cleaned_description = description.apply(lambda s: s.translate(str.maketrans(' ', '', string.punctuation + u'\xa0')))
>>> print(cleaned_description.head(10).to_markdown())
|   | job_description
|---|:-----
| 0 | The chosen Sr Software Developer will be part of a larger engineering team developing software for Medical Devices used with
| 1 | Position C Lead Software Developer Location Middletown NJ Compensation 90000 110000 Excellent Benefits Paid Relocation Jo
| 2 | Senior Software Developer Hoboken NJ Starts as 912 month contract possibility to extend Required Skills 5 years professiona
| 3 | Our client a multinational publishing and education company is seeking a Senior Software Developer Location Hoboken NJ Posit
| 4 | Position C Lead Software Developer Location Philadelphia PA Compensation 90000 110000 Excellent Benefits Paid Relocation
| 5 | RESPONSIBILITIES Kforce has a client seeking a Software Developer 3 in Mountain View California CA REQUIREMENTS MS in Compu
| 6 | Apply by EmailDirect Application at uditsharmasigmawayorg C Software Developer Duration 12 Months Location Sunnyvale CA To
| 7 | Talent Space Inc is looking for C Software Developer in SunnyvaleThe hiring we are working with is looking software develope
>>>
```

Delete duplicated data and reverse the word to lower case.

Old length of data

Length: 7948, dtype: object

```
>>> data = pd.read_csv('test4.csv')
>>> description = data['job_description']
>>> cleaned_description = description.apply(lambda s: s.translate(str.maketrans(' ', '', string.punctuation + u'\xa0')))
>>> cleaned_description = cleaned_description.apply(lambda s: s.lower())
>>> cleaned_description = cleaned_description.apply(lambda s: s.translate(str.maketrans(string.whitespace, ' ' * len(string.whitespace), '')))
>>> cleaned_description = cleaned_description.drop_duplicates()
>>>
```

```
>>> cleaned_description
0      the chosen sr software developer will be part ...
1      position c lead software developer location mi...
2      senior software developer hoboken nj starts as...
3      our client a multinational publishing and educ...
4      position c lead software developer location ph...
...
7943    career evolutions is searching for a junior so...
7944    responsibilitieskforce has a client that is se...
7945    mid level software developer dahlgren virginia...
7946    this company is looking for a full stack devel...
7947    experienced net web developersour client has a...
Name: job_description, Length: 6267, dtype: object
```

Split any space of each description to keep as array

```
>>> def get_and_clean_data():
>>>     data = pd.read_csv('test4.csv')
>>>     description = data['job_description']
>>>     cleaned_description = description.apply(lambda s: s.translate(str.maketrans(',', ' ', string.punctuation + u'\xa0')))
>>>     cleaned_description = cleaned_description.apply(lambda s: s.lower())
>>>     cleaned_description = cleaned_description.apply(lambda s: s.translate(str.maketrans(string.whitespace, ' ' * len(string.whitespace), '')))
>>>     cleaned_description = cleaned_description.drop_duplicates()
>>>     return cleaned_description
>>>
>>> def simple_tokenize(data):
>>>     cleaned_description = data.apply(lambda s: [x.strip() for x in s.split()])
>>>     return cleaned_description
>>>
>>> cleaned_description = simple_tokenize(cleaned_description)
>>> print(cleaned_description.head().to_markdown())
|   | job_description
|---|:-----
| 0 | ['the', 'chosen', 'sr', 'software', 'developer', 'will', 'be', 'part', 'of', 'a', 'larger', 'engineering', 'team', 'developing', 'software',
| 1 | ['position', 'c', 'lead', 'software', 'developer', 'location', 'middletown', 'nj', 'compensation', '90000', '110000', 'excellent', 'benefits
| 2 | ['senior', 'software', 'developer', 'hoboken', 'nj', 'starts', 'as', '912', 'month', 'contract', 'possibility', 'to', 'extend', 'required',
>>>
```

After split try find total of python and MySQL

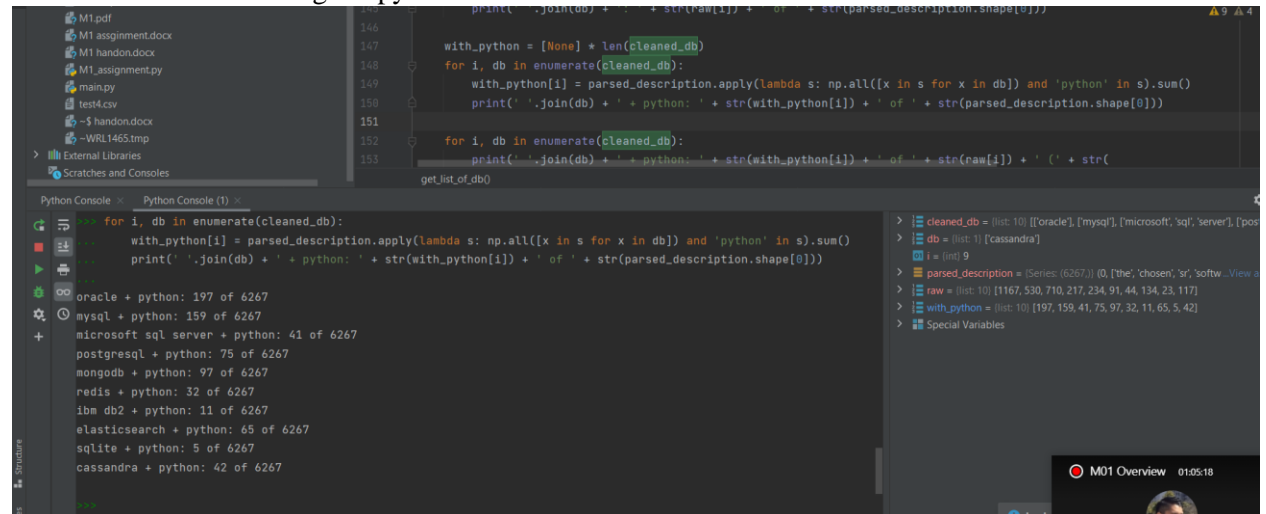
```
>>> def get_and_clean_data():
>>>     data = pd.read_csv('test4.csv')
>>>     description = data['job_description']
>>>     cleaned_description = description.apply(lambda s: s.translate(str.maketrans(',', ' ', string.punctuation + u'\xa0')))
>>>     cleaned_description = cleaned_description.apply(lambda s: s.lower())
>>>     cleaned_description = cleaned_description.apply(lambda s: s.translate(str.maketrans(string.whitespace, ' ' * len(string.whitespace), '')))
>>>     cleaned_description = cleaned_description.drop_duplicates()
>>>     return cleaned_description
>>>
>>> def simple_tokenize(data):
>>>     cleaned_description = data.apply(lambda s: [x.strip() for x in s.split()])
>>>     return cleaned_description
>>>
>>> def parse_job_description():
>>>     cleaned_description = get_and_clean_data()
>>>     cleaned_description = simple_tokenize(cleaned_description)
>>>     return cleaned_description
>>>
>>> parsed_description = parse_job_description()
>>> count_python = parsed_description.apply(lambda s: 'python' in s).sum()
>>> count_mysql = parsed_description.apply(lambda s: 'mysql' in s).sum()
>>> print('python: ' + str(count_python) + ' of ' + str(parsed_description.shape[0]))
python: 1125 of 6267
>>> print('mysql: ' + str(count_mysql) + ' of ' + str(parsed_description.shape[0]))
mysql: 530 of 6267
```

Fetch data from the website to separate and keep it as an array and stick on each their group.

```
def parse_db():
    html_doc = requests.get("https://db-engines.com/en/ranking").content
    soup = BeautifulSoup(html_doc, 'html.parser')
    db_table = soup.find("table", {"class": "dbi"})
    all_db = [' '.join(s.find('a').findAll(text=True, recursive=False)).strip() for s in db_table.findAll("th", {"class": "pad-l"})]
    all_db = list(dict.fromkeys(all_db))
    db_list = all_db[:10]
    db_list = [s.lower() for s in db_list]
    db_list = [[x.strip() for x in s.split()] for s in db_list]
    # [['oracle'], ['mysql'], ['microsoft', 'sql', 'server'], ['postgresql'], ['mongodb'], ['redis'], ['ibm', 'db2'],
    #  ['elasticsearch'], ['sqlite'], ['cassandra']]
    return db_list

db_list
[['oracle'], ['mysql'], ['microsoft', 'sql', 'server'], ['postgresql'], ['mongodb'], ['redis'], ['ibm', 'db2'], ['elasticsearch'], ['sqlite'], ['cassandra']]
```

Find the DB that use along the python.



```
print(' '.join(db) + ' ' + str(raw[i]) + ' of ' + str(parsed_description.shape[0]))

with_python = [None] * len(cleaned_db)
for i, db in enumerate(cleaned_db):
    with_python[i] = parsed_description.apply(lambda s: np.all([x in s for x in db]) and 'python' in s).sum()
    print(' '.join(db) + ' ' + python: ' + str(with_python[i]) + ' of ' + str(parsed_description.shape[0]))

for i, db in enumerate(cleaned_db):
    print(' '.join(db) + ' ' + python: ' + str(with_python[i]) + ' of ' + str(raw[i]) + ' (' + str(
get_list_of_db()
```

Python Console (1)

```
for i, db in enumerate(cleaned_db):
    with_python[i] = parsed_description.apply(lambda s: np.all([x in s for x in db]) and 'python' in s).sum()
    print(' '.join(db) + ' ' + python: ' + str(with_python[i]) + ' of ' + str(parsed_description.shape[0]))
```

oracle + python: 197 of 6267
mysql + python: 159 of 6267
microsoft sql server + python: 41 of 6267
postgresql + python: 75 of 6267
mongodb + python: 97 of 6267
redis + python: 32 of 6267
ibm db2 + python: 11 of 6267
elasticsearch + python: 65 of 6267
sqlite + python: 5 of 6267
cassandra + python: 42 of 6267

cleaned_db = (list: 10) [['oracle'], ['mysql'], ['microsoft', 'sql', 'server'], ['postgresql'], ['mongodb'], ['redis'], ['ibm', 'db2'], ['elasticsearch'], ['sqlite'], ['cassandra']]
db = (list: 1) ['cassandra']
i = (int) 9
parsed_description = (Series: (6267)) 0, [the, 'chosen', 'sr', 'softw... View a
raw = (list: 10) [1167, 530, 710, 217, 234, 91, 44, 134, 23, 117]
with_python = (list: 10) [197, 159, 41, 75, 97, 32, 11, 65, 5, 42]
Special Variables

M01 Overview 01:05:18

Compute as a percentage

```
>>> for i, db in enumerate(cleaned_db):
...     print(' '.join(db) + ' ' + python: ' + str(with_python[i]) + ' of ' + str(raw[i]) + ' (' + str(
...         np.around(with_python[i] / raw[i] * 100, 2)) + '%')
... 
```

oracle + python: 197 of 1167 (16.88%)
mysql + python: 159 of 530 (30.0%)
microsoft sql server + python: 41 of 710 (5.77%)
postgresql + python: 75 of 217 (34.56%)
mongodb + python: 97 of 234 (41.45%)
redis + python: 32 of 91 (35.16%)
ibm db2 + python: 11 of 44 (25.0%)
elasticsearch + python: 65 of 134 (48.51%)
sqlite + python: 5 of 23 (21.74%)
cassandra + python: 42 of 117 (35.9%)

Indexed matrices illustrated

```
def create_index():
    lang = [['java'], ['python'], ['c'], ['kotlin'], ['swift'], ['rust'], ['ruby'], ['scala'], ['julia'], ['lua']]
    parsed_description = parse_job_description()
    parsed_db = parse_db()
    all_terms = lang + parsed_db
    query_map = pd.DataFrame(parsed_description.apply
                              (lambda s: [1 if np.all([d in s for d in db]) else 0 for db in all_terms])
                              .values.tolist(), columns=[' '.join(d) for d in all_terms])
```

```
>>> query_map
   java  python  c  kotlin  ...  ibm db2  elasticsearch  sqlite  cassandra
0      0      0  1      0  ...      0      0      0      0
1      0      0  1      0  ...      0      0      0      0
2      0      0  0      0  ...      0      0      0      0
3      0      0  0      0  ...      0      0      0      0
4      0      0  1      0  ...      0      0      0      0
...    ... ..  ...  ...    ...    ...    ...    ...    ...
6262   0      0  1      0  ...      0      0      0      0
6263   1      0  0      0  ...      0      0      0      0
6264   1      0  1      0  ...      0      0      0      0
6265   0      0  0      0  ...      0      0      0      0
6266   0      0  0      0  ...      0      0      0      0
```

[6267 rows x 20 columns]

```
>>> print(query_map.head(20).to_markdown())
|   |  java |  python |  c |  kotlin |  swift |  rust |  ruby |  scala |  julia |  lua |  oracle |  mysql |
|---|-----|-----|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0 |    0 |    0 |  1 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |
| 1 |    0 |    0 |  1 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |
| 2 |    0 |    0 |  0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |
| 3 |    0 |    0 |  0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |
| 4 |    0 |    0 |  1 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |
| 5 |    0 |    0 |  0 |    0 |    1 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |
| 6 |    0 |    0 |  1 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |
| 7 |    0 |    0 |  1 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |
| 8 |    0 |    0 |  1 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |
| 9 |    0 |    0 |  0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |
|10 |    1 |    1 |  1 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    1 |    1 |
|11 |    1 |    0 |  0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |
|12 |    1 |    0 |  0 |    0 |    1 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |
|13 |    0 |    0 |  1 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |    0 |
|14 |    0 |    0 |  0 |    0 |    0 |    0 |    1 |    0 |    0 |    0 |    0 |    0 |
```

Query any data that invoke with java

```
query_map[query_map['java'] > 0].apply(lambda s: np.where(s == 1)[0],axis=1).apply(lambda s: list(query_map.columns[s]))
10      [java, python, c, oracle, mysql, mongodb]
11                                     [java]
12      [java, swift, redis]
16      [java, c, swift, oracle, cassandra]
19      [java, python]
...
6250      [java, python, c, ruby]
6252      [java, python, c]
6259      [java, c, ruby]
6263      [java]
6264      [java, c]
Length: 2701, dtype: object
```