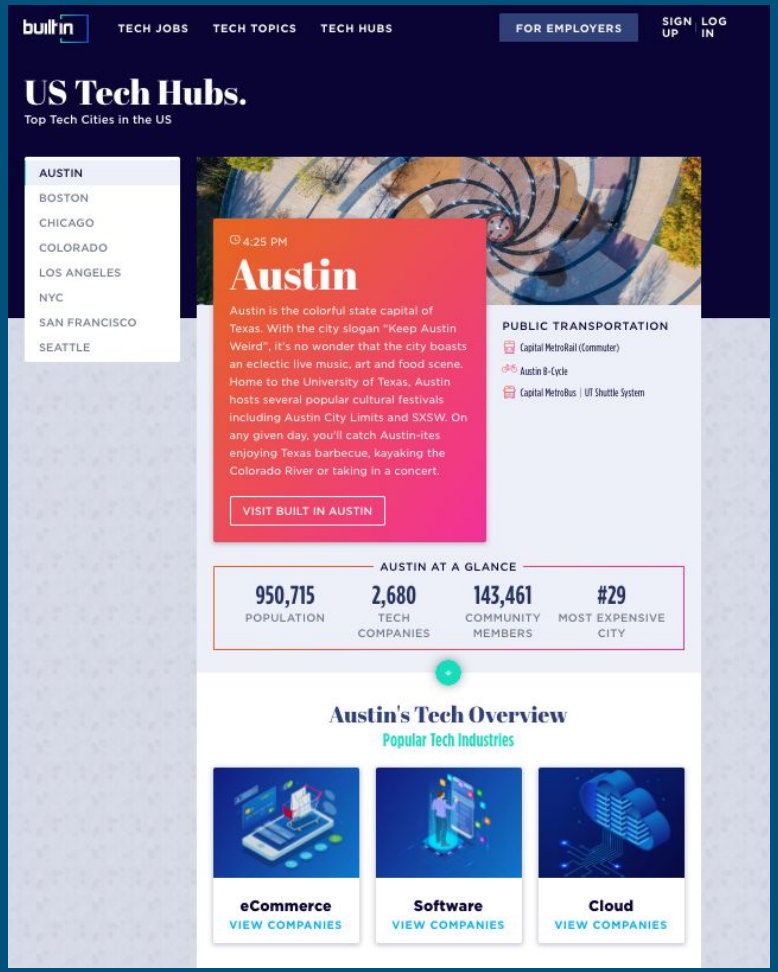
Job Hunt ETL Project

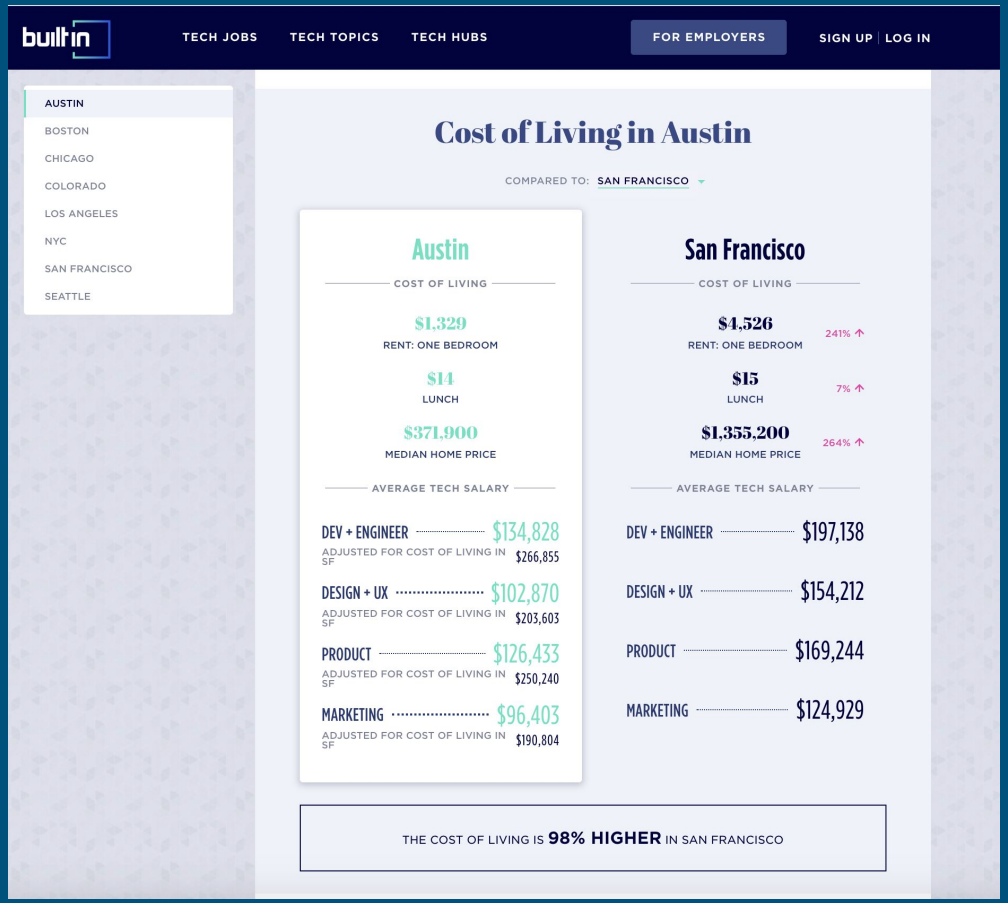
Connor Rubenstein, Liz Stevens,
Jimmy Walsh

Project Overview

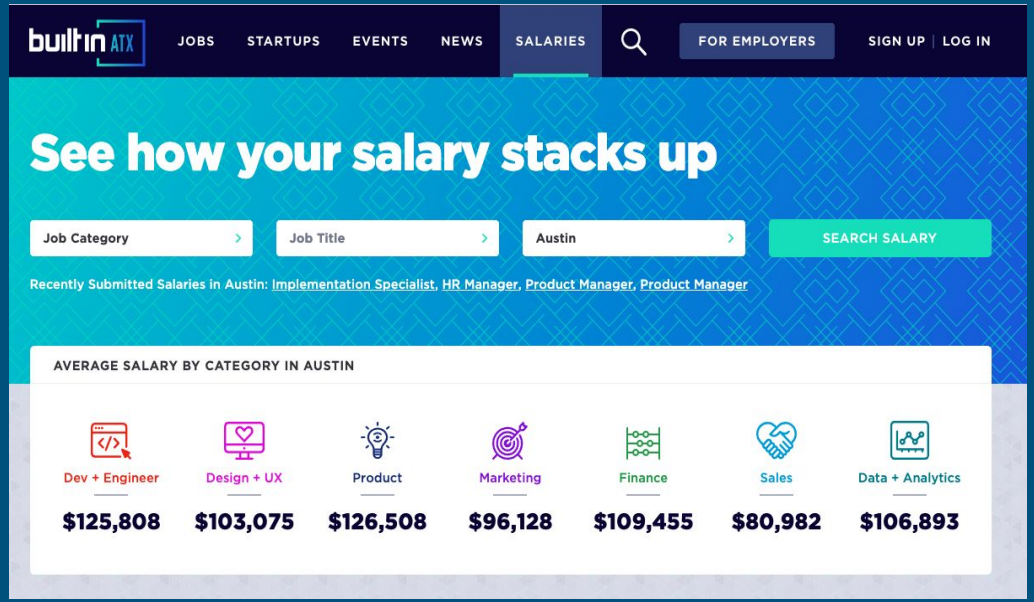
- Scrape data from builtin.com & associated TechHub Websites
    - Data: job postings, company, TechHub + location details, avg. salary, popular industries, etc.
- Design ERD + PostgreSQL schema to relate + manage data
- Clean/Transform data in Pandas; convert to csv files
- Upload data into schema; building relationships with primary + foreign key IDs

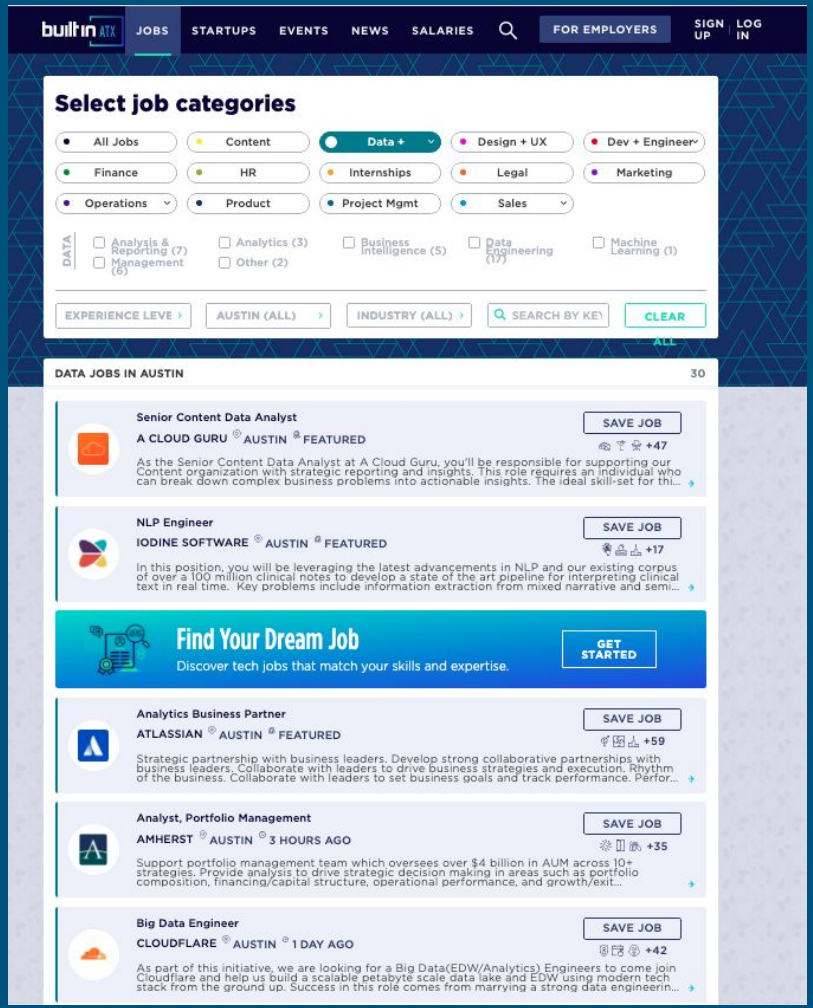
Main builtin.com TechHub page + Cost of Living per TechHub





Independent TechHub websites w/ Salary and Jobs pages
Example for Austin, TX: builtinaustin.com





Finding Data (Extract)

Web scraping loops w/ BeautifulSoup, Splinter + Selenium

In [1]:
```
from bs4 import BeautifulSoup
from splinter import Browser
import time
import requests
import os
from selenium import webdriver
import pandas as pd
```

```
#find and collect all cities listed on page
cities = browser.find_elements_by_xpath('//div[@class="block block-techhub-cost-block"]//div[@

cities2 = []

for i in range(len(cities)):
    cities2.append(cities[i].text)

del cities2[1::2]
print(cities2)
```

```
['Austin', 'Boston', 'Chicago', 'Colorado', 'Los Angeles', 'NYC', 'San Francisco', 'Seattle']
```

In [2]:
```
url = "https://builtin.com/tech-hubs"
browser = webdriver.Chrome()
browser.get(url)
time.sleep(10)
html = browser.page_source
soup = BeautifulSoup(html, 'html.parser')
print(soup.prettify())
```

```
pop_field = pd.DataFrame({"popular_fields2":popular_fields2})
pop_field2 = pop_field.popular_fields2.unique()
#len(popular_fields2)
pop_field2
```

```
array(['eCommerce', 'Software', 'Cloud', 'Edtech', 'Hardware', 'Security',
       'Fintech', 'Healthtech', 'Big Data', 'Adtech', 'Mobile',
       'Consumer Web', 'Digital Media', 'Real Estate',
       'Artificial Intelligence', 'Marketing Tech'], dtype=object)
```

Finding Data (Extract) cont.

```
#Webscrapping Average Salaries by Category in Austin

url2 = "https://builtinaustin.com/salaries"
browser = webdriver.Chrome()
browser.get(url2)
time.sleep(2)
html = browser.page_source
soup = BeautifulSoup(html, 'html.parser')
Austin_fields = soup.find_all('span', class_="field-content")
#austin_fields

field_title = []
field_price = []

for c in range(len(Austin_fields)):
    title = Austin_fields[c].find("div", class_="field-title").text
    price = Austin_fields[c].find("div", class_="field-price").text
    #print(title)
    #print(price)
    #print("----------")
    field_title.append(title)
    field_price.append(price)

Austin_df = pd.DataFrame({"Category_Name":field_title,"Avg Salary":field_price})
Austin_df
```

	Category_Name	Avg Salary
0	Dev + Engineer	$125,808
1	Design + UX	$103,075
2	Product	$126,508
3	Marketing	$96,128
4	Finance	$109,455
5	Sales	$80,982
6	Data + Analytics	$106,893

```
#click link to tech hubs selenium full xpath code
#browser.find_element_by_xpath("""/html/body/div[1]/div/header/div/div/div/div/div[1]/div[1]/div/div[
```

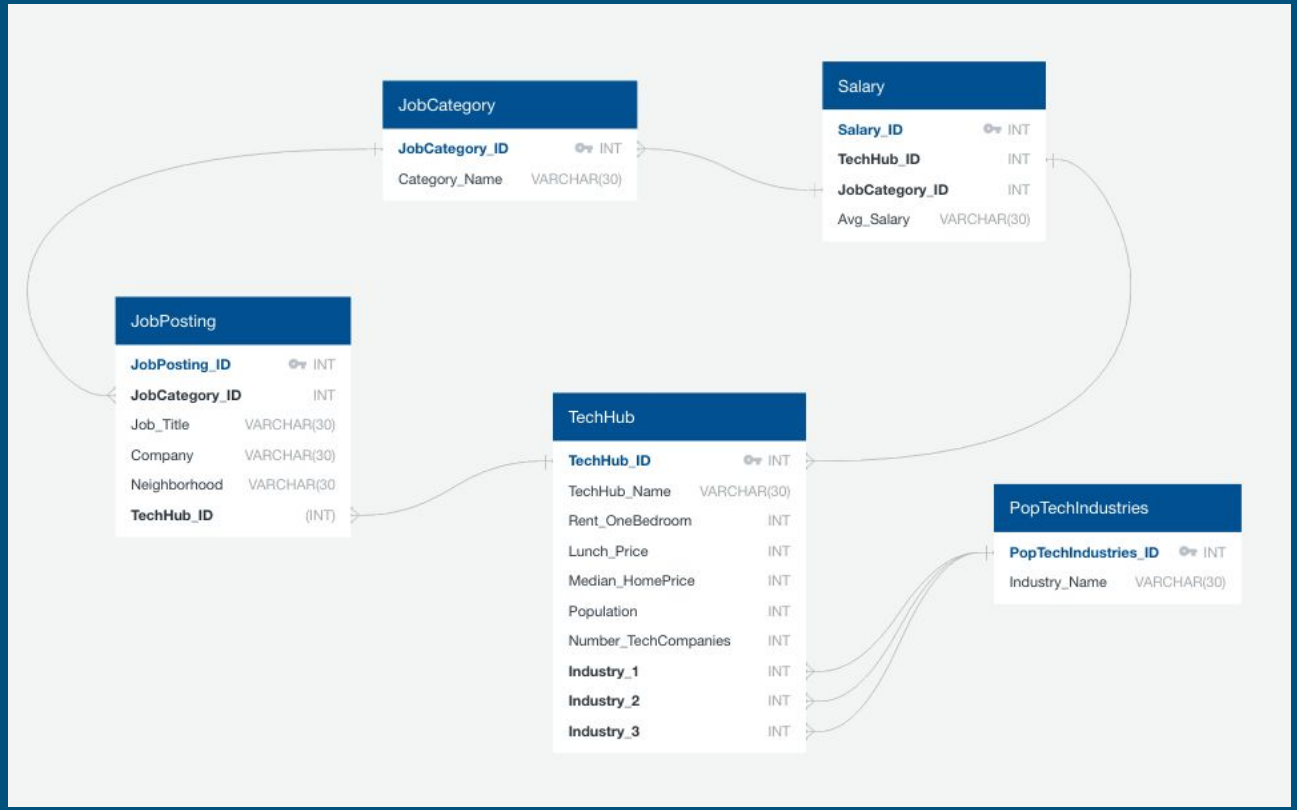
```
#Create lists to hold data to put into columns for a dataframe
Job_Title = []
Company = []
City = []
Field = []
Tech_Hub = []
#Loop of Tech Hubs, but changed to 5 because the hubs of sf and seattle don't have the same page structure.
city_loop = ["chicago","boston","austin","colorado","nyc","la"]
#List of job fields that I wish to click through as I loop through the different Tech Hubs
Fields = ["Dev + Engineer","Design + UX","Product","Marketing","Finance","Sales","Data + Analytics"]

#loop to go through the different tech hubs
for c in city_loop:
    #logic adapt urls based on the tech hub desired. logic to handle the fact that chicago was .org and the rest were .com
    if c in ["chicago"]:
        url =f"https://www.builtin{c}.org/jobs"
    else:
        url =f"https://www.builtin{c}.com/jobs"
    # browse to the tech hub url
    browser.visit(url)
    # slight delay to let the page load
    time.sleep(1)
    #loop to go through and click the different job field options
    for field in Fields:

        #How I found out sf and seattle didn't function
        #click field buttons
        browser.click_link_by_partial_text(field)
        print(field)
        #pause to let the page load
        time.sleep(2)
```

- Loop through each TechHub & job category.
- Match job category + job title for each TechHub.
- Scrape 5 jobs per category per city
- Seattle and SF needed different parameters.

```
#grab the html code of the page
html = browser.html
#read and store html as a soup object
soup = BeautifulSoup(html, 'html.parser')
#grab all subsections of "left-col" which are the containers with the job postings and store in a list
comp_jobs = soup.find_all('div', class_="left-col")
k=0
#Loop through the subsections and do so as integers so that list items can be referenced
while k < 5:
    try:
        #store title, company, and city
        title = comp_jobs[k].find('h2',class_="title").text
        company=comp_jobs[k].find('div',class_="company-title").text
        print(f"{title},{company}")
        Field.append(field)
        Job_Title.append(title)
        Company.append(company)
        Tech_Hub.append(c)
        try:
            city=comp_jobs[k].find('div',class_="job-location").text
            City.append(city)
        except:
            City.append("0")
        k+=1
    except:
        pass
time.sleep(0.5)
browser.click_link_by_partial_text(field)
time.sleep(1)
```

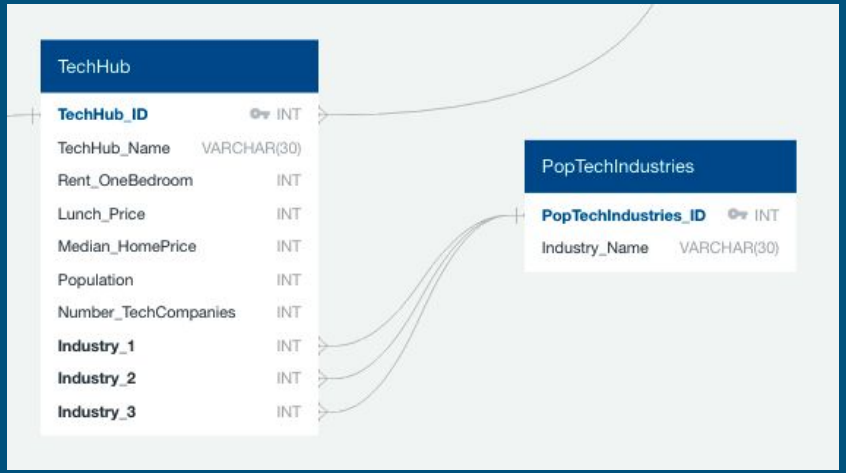
Data Cleanup (Transform)

Create ERD to relate extracted data together



Data Cleanup (Transform) cont.

create PostgreSQL schema to guide our data transformation in pandas



Dashboard | Properties | SQL | Statistics | Dependencies | Dependents | Completed_Schema.sql *

No limit

Final_ETL_Project/postgres@PostgreSQL 11

Query Editor | Query History

```
1  -- CREATE one table at a time, upload matching data (including IDs) and then proceed
2  -- to the next table
3
4  --Created, loaded
5  CREATE TABLE PopTechIndustries (
6      PopTechIndustries_ID SERIAL PRIMARY KEY NOT NULL,
7      Industry_Name VARCHAR(30) NOT NULL
8  );
9
10 --Created, loaded
11 CREATE TABLE TechHub (
12     TechHub_ID INT PRIMARY KEY NOT NULL,
13     TechHub_Name VARCHAR(30) NOT NULL,
14     Rent_OneBedroom VARCHAR(30),
15     Lunch_Price VARCHAR(30),
16     Median_HomePrice VARCHAR(30),
17     Population VARCHAR(30),
18     Number_TechCompanies VARCHAR(30),
19     Industry_1 INT,
20     Industry_2 INT,
21     Industry_3 INT,
22     FOREIGN KEY (Industry_1) REFERENCES PopTechIndustries(PopTechIndustries_ID),
23     FOREIGN KEY (Industry_2) REFERENCES PopTechIndustries(PopTechIndustries_ID),
24     FOREIGN KEY (Industry_3) REFERENCES PopTechIndustries(PopTechIndustries_ID)
25 );
26
```

Data Cleanup (Transform) cont.

Normalize data in lists and build dataframes w/ Pandas

```
#Combing all found data above and putting it into a clean DataFrame

TechHub = pd.DataFrame({"TechHub_Name":cities2,"Rent_OneBedroom":living_cost2,"Lunch
TechHub
```

	TechHub_Name	Rent_OneBedroom	Lunch_Price	Median_HomePrice	Population	Number_TechCompanies
0	Austin	$1,329	$14	$371,900	950,715	2,680
1	Boston	$2,440	$15	$592,300	698898	2335
2	Chicago	$1,504	$15	$226,500	2749360	6003
3	Colorado	$1,568	$14	$421,400	5770000	3939
4	Los Angeles	$2,293	$17	$689,700	3946000	6106
5	NYC	$2,299	$20	$671,700	8558000	5243
6	San Francisco	$4,526	$15	$1,355,200	884363	4362
7	Seattle	$1,878	$16	$714,100	755745	2155

```
Austin_df2 = pd.DataFrame({"TechHub_Name":cities2[0],
Austin_df2
```

	TechHub_Name	Category_Name	Avg Salary
0	Austin	Dev + Engineer	$125,808
1	Austin	Design + UX	$103,075
2	Austin	Product	$126,508
3	Austin	Marketing	$96,128
4	Austin	Finance	$109,455
5	Austin	Sales	$80,982
6	Austin	Data + Analytics	$106,893

```
= pd.read_csv("Tech_Hub2.csv")
2=th[["Unnamed: 0","TechHub_Name"]]
3 = th2.rename(columns = {"Unnamed: 0":"TechHub_ID"})
3
```

TechHub_ID	TechHub_Name
0	Austin
1	Boston
2	Chicago
3	Colorado
4	Los Angeles
5	NYC
6	San Francisco
7	Seattle

```
a_df = Austin_df2.append(Boston_df2)
#combo_df
b_df = a_df.append(Chicago_df2)
c_df = b_df.append(Colorado_df2)
d_df = c_df.append(la_df2)
e_df = d_df.append(nyc_df2)
f_df = e_df.append(sf_df2)
Salary_df = f_df.append(Seattle_df2)

Salary_df
```

	TechHub_Name	Category_Name	Avg Salary
0	Austin	Dev + Engineer	$125,808
1	Austin	Design + UX	$103,075
2	Austin	Product	$126,508
3	Austin	Marketing	$96,128
4	Austin	Finance	$109,455
5	Austin	Sales	$80,982
6	Austin	Data + Analytics	$106,893
0	Boston	Dev + Engineer	$127,741
1	Boston	Design + UX	$111,524
2	Boston	Product	$138,685
3	Boston	Marketing	$120,244
4	Boston	Finance	$126,181
5	Boston	Sales	$90,887
6	Boston	Data + Analytics	$106,290
0	Chicago	Dev + Engineer	$123,880

Data Cleanup (Transform) cont.

Prep all formatted data frames w/ IDs to match PostgreSQL schema

```
Categories = pd.DataFrame({"Category_Name": ["Dev
Categories.to_csv("Category.csv")
```

```
Categories["JobCategory_ID"]=[0,1,2,3,4,5,6]
```

```
Categories
```

	Category_Name	JobCategory_ID
0	Dev + Engineer	0
1	Design + UX	1
2	Product	2
3	Marketing	3
4	Finance	4
5	Sales	5
6	Data + Analytics	6

Number	1	2	3
TechHub_Name			
Austin	0.0	1.0	2.0
Boston	3.0	4.0	5.0
Chicago	6.0	7.0	8.0
Colorado	1.0	9.0	10.0
Los Angeles	11.0	12.0	0.0
NYC	13.0	6.0	12.0
San Francisco	6.0	14.0	NaN
Seattle	1.0	15.0	7.0

```
final_tech = TechHub.merge(final,  on = "TechHub_Name")
```

```
final_tech = final_tech.rename(columns = {1:"Industry_1",2:"Industr
final_tech.fillna(100, inplace = True)
```

```
final_tech["TechHub_Name"]=final_tech["TechHub_Name"].astype(str)
```

```
ws = Salary_df.merge(th3, on = "TechHub_Name")
ws2 = ws.merge(Categories, on = "Category_Name")
salaryfinal = ws2[["TechHub_ID","JobCategory_ID","Avg Salary"]]
salaryfinal.to_csv("salary.csv")
```

```
ws2
```

	TechHub_Name	Category_Name	Avg Salary	TechHub_ID	JobCategory_ID
0	Austin	Dev + Engineer	$125,808	0	0
1	Boston	Dev + Engineer	$127,741	1	0
2	Chicago	Dev + Engineer	$123,880	2	0
3	Colorado	Dev + Engineer	$114,849	3	0
4	Los Angeles	Dev + Engineer	$135,399	4	0
5	NYC	Dev + Engineer	$141,021	5	0
6	San Francisco	Dev + Engineer	$162,254	6	0
7	Seattle	Dev + Engineer	$137,105	7	0
8	Austin	Design + UX	$103,075	0	1
9	Boston	Design + UX	$111,524	1	1

Data Cleanup (Transform) cont.

Prep all formatted data frames w/ IDs to match PostgreSQL schema

```
#used lists to create data frame with appropriate column headers
Job_Results_df = pd.DataFrame({"Category_Name":Field,"Job_Title":Job_Title,"Com
Job_Results_df.replace({"chicago":"Chicago","boston":"Boston", "austin": "Austi
Job_Results_df.head(2)
```

	Category_Name	Job_Title	Company	TechHub_Name	City
0	Dev + Engineer	Senior Software Engineer, Front End	Kalderos	Chicago	Chicago
1	Dev + Engineer	Senior Site Reliability Engineer	Optiver	Chicago	Chicago

```
jc = pd.read_csv("Category.csv")
jc2 = jc.rename(columns = {"Unnamed: 0":"ID"})
jc3=jc2["Category_Name"]
jc2
```

	ID	Category_Name
0	0	Dev + Engineer
1	1	Design + UX
2	2	Product
3	3	Marketing
4	4	Finance
5	5	Sales
6	6	Data + Analytics

```
th = pd.read_csv("Tech_Hub2.csv")
th2=th[["Unnamed: 0","TechHub_Name"]]
th2=th2.rename(columns = {"Unnamed: 0":"TechHub_ID"})
th2
```

	TechHub_ID	TechHub_Name
0	0	Austin
1	1	Boston
2	2	Chicago
3	3	Colorado
4	4	Los Angeles
5	5	NYC
6	6	San Francisco
7	7	Seattle

```
JR_2 = Job_Results_df.merge(th2, on = "TechHub_Name")
JR3 = JR_2.merge(jc2, on= "Category_Name")
JR3
```

	Category_Name	Job_Title	Company	TechHub_Name	City	TechHub_ID	JobCategory_ID
0	Dev + Engineer	Senior Software Engineer, Front End	Kalderos	Chicago	Chicago	2	0
1	Dev + Engineer	Senior Site Reliability Engineer	Optiver	Chicago	Chicago	2	0

```
JobPosting = JR3[["JobCategory_ID","Job_Title","Company","City","TechHub_ID"]]
JobPosting
```

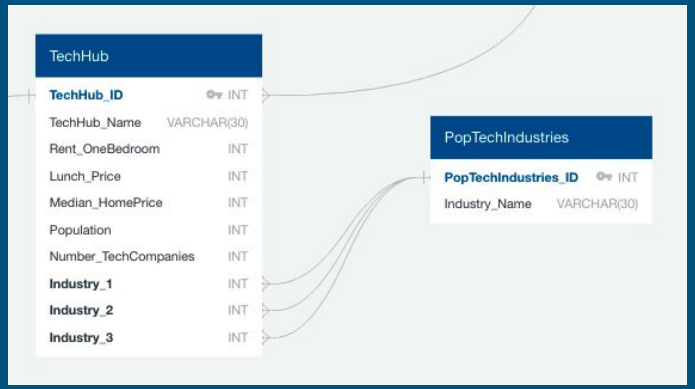
	JobCategory_ID	Job_Title	Company	City	TechHub_ID
0	0	Senior Software Engineer, Front End	Kalderos	Chicago	2
1	0	Senior Site Reliability Engineer	Optiver	Chicago	2
2	0	Cloud Engineer	Slalom	Chicago	2
3	0	Sr. Microsoft Cloud Engineer	DFIN	Chicago	2
4	0	Senior Site Reliability Engineer, Observability	Cisco Meraki	Remote	2
...	...	...	...	...	...
290	6	Senior Associate, Digital Marketing	Remitly	Seattle	7
291	6	Senior Marketing Manager	Ekata	Seattle	7
292	6	Global Communications Leader	Qualtrics	Seattle	7
293	6	Senior Digital Marketing Manager	Convoy	Seattle	7
294	6	Product Marketing Manager, Sales Enablement an...	Convoy	Seattle	7

Data Analysis (Load)

Example: TechHub Table

Data Output | Explain | Messages | Notifications

	techhub_id [PK] integer	techhub_name character varying (30)	rent_onebedroom character varying (30)	lunch_price character varying (30)	median_homeprice character varying (30)	population character varying (30)	number_techcompanies character varying (30)	industry_1 integer	industry_2 integer	industry_3 integer
1	0	Austin	$1,329	$14	$371,900	950,715	2,680	0	1	2
2	1	Boston	$2,440	$15	$592,300	698898	2335	3	4	5
3	2	Chicago	$1,504	$15	$226,500	2749360	6003	6	7	8
4	3	Colorado	$1,568	$14	$421,400	5770000	3939	1	9	10
5	4	Los Angeles	$2,293	$17	$689,700	3946000	6106			
6	5	NYC	$2,299	$20	$671,700	8558000	5243			
7	6	San Francisco	$4,526	$15	$1,355,200	884363	4362			
8	7	Seattle	$1,878	$16	$714,100	755745	2155			



Data Output | Explain | Messages | Notifications

	poptechindustries_id [PK] integer	industry_name character varying (30)
1	0	eCommerce
2	1	Software
3	2	Cloud
4	3	Edtech
5	4	Hardware
6	5	Security
7	6	Fintech
8	7	Healthtech
9	8	Big Data
10	9	Adtech
11	10	Mobile
12	11	Consumer Web
13	12	Digital Media
14	13	Real Estate
15	14	Artificial Intelligence

Data Analysis (Load) cont.

Data Output | Explain | Messages | Notifications

	jobposting_id [PK] integer	jobcategory_id integer	job_title character varying (250)	company character varying (30)	neighborhood character varying (30)	techhub_id integer
296	295	6	Sr. Business Intelligence Engi...	Alliant Credit Union	Chicago	2
297	296	6	Senior Manager, Data Enginee...	Integral Ad Science	Chicago	2
298	297	6	Senior Big Data Engineer- Dat...	Integral Ad Science	Chicago	2
299	298	6	Cell Culture Research Associa...	Tempus	Chicago	2
300	299	6	Senior Database Engineer, SQ...	DRW	Chicago	2
301	300	6	Machine Learning Engineer	Kensho Technologies	Cambridge	1
302	301	6	Senior Cloud DevOps Engineer	Agero + Swoop	Greater Boston Area	1
303	302	6	Business Intelligence Analyst, ...	Drizly	Greater Boston Area	1
304	303	6	Software Engineer in Big Data ...	Arcadia	Greater Boston Area	1
305	304	6	Data Integration Engineer	Arcadia	Greater Boston Area	1
306	305	6	Database Administrator	LeanDNA	Austin	0
307	306	6	Manager, Data Engineering	A Cloud Guru	Austin	0
308	307	6	NLP Engineer	Iodine Software	Austin	0
309	308	6	Big Data Engineer	Cloudflare	Austin	0
310	309	6	Senior Data Engineer	Bestow	Austin	0

Data Output | Explain | Messages | Notifications

	jobcategory_id [PK] integer	category_name character varying (30)
1	0	Dev + Engineer
2	1	Design + UX
3	2	Product
4	3	Marketing
5	4	Finance
6	5	Sales
7	6	Data + Analytics

Data Output | Explain | Messages | Notifications

	salary_id [PK] integer	techhub_id integer	jobcategory_id integer	avg_salary character varying (30)
42	41	1	5	$90,887
43	42	2	5	$89,642
44	43	3	5	$76,160
45	44	4	5	$95,601
46	45	5	5	$97,993
47	46	6	5	$130,187
48	47	7	5	$90,755
49	48	0	6	$106,893
50	49	1	6	$106,290
51	50	2	6	$103,817
52	51	3	6	$99,918
53	52	4	6	$112,475
54	53	5	6	$120,117
55	54	6	6	$144,804
56	55	7	6	$118,251

Job #300, Machine Learning Engineer in the Data + Analytics Category from Boston TechHub has avg_salary of $106,290! Top industries are EdTech, Hardware, and Security.

Confirming Database (testing db with joins)

```
57
58  -- Test joins to make sure the schema works with primary + foreign keys.
59
60  Select jc.category_name, jp.Job_Title, jp.Company, jp.neighborhood, th.TechHub_Name, s.avg_salary
61  FROM JobPosting jp
62  INNER JOIN TechHub th
63      on jp.techhub_id = th.techhub_id
64  INNER JOIN jobcategory jc
65      on jp.jobcategory_id = jc.jobcategory_id
66  INNER JOIN Salary s
67      on s.techhub_id = th.techhub_id
68      AND s.jobcategory_id = jc.jobcategory_id;
69
70
```

Data Output | Explain | Messages | Notifications

	category_name character varying (30)	job_title character varying (250)	company character varying (30)	neighborhood character varying (30)	techhub_name character varying (30)	avg_salary character varying (30)
1	Dev + Engineer	Software Engineering Manager	Farmer's Fridge	Chicago	Chicago	$123,880
2	Dev + Engineer	Salesforce – CPQ Solution Ar...	Slalom	Chicago	Chicago	$123,880
3	Dev + Engineer	Software Engineering Manager	OppLoans	Chicago	Chicago	$123,880
4	Dev + Engineer	Senior .Net Software Engineer...	DocuSign	Chicago	Chicago	$123,880
5	Dev + Engineer	Windows Engineer	Optiver	Chicago	Chicago	$123,880
6	Dev + Engineer	Senior Engineering Manager	CircleCI	Greater Boston Area	Boston	$127,741
7	Dev + Engineer	Software Architect	CrunchTime!	Greater Boston Area	Boston	$127,741
8	Dev + Engineer	Lead Data Scientist	Vistaprint	Waltham	Boston	$127,741
9	Dev + Engineer	DevOps Engineer	Arcadia	Greater Boston Area	Boston	$127,741
10	Dev + Engineer	Senior Software Engineer - IoT	Markforged	Watertown	Boston	$127,741
11	Dev + Engineer	Lead Salesforce Engineer	ThousandEyes	Austin	Austin	$125,808
12	Dev + Engineer	Software Test Engineer	Compeat	Austin	Austin	$125,808
13	Dev + Engineer	Principal Database Reliability ...	Cognite	Austin	Austin	$125,808

Things We Learned

- The logic we added to our tables by adding IDs in Pandas (via serial numbered columns), when uploaded to PostgreSQL through CSVs, does map onto our Primary Keys and Foreign Keys we pre-created in our PostgreSQL schema
    - We ALSO explored writing directly to PostgreSQL in python using *df.to_sql* (SQLAlchemy), but realized that the Primary Key and Foreign Key properties that we created would be overwritten when the data is inserted.

- *Time management on ourselves; what we wanted to do vs what we were able to do*
- Using Selenium (trying to learn another way to code to extract the exact value of what we were looking for)
- The HTML of a page can dynamically change with the size.

Roadblocks/Setbacks

- Webpage/window size
- Realized jobs were listed under multiple categories (categories seen as tags)
- “Remote” listed as a Neighborhood for each TechHub
- Some links rerouted us to a “.org” compared to a “.com”
- Having to reformat values we scraped from the websites to merge with our Database (e.g. convert integer values to strings)
- Tried to load data to Database via Jupyter Notebook with python before converting to csv

Additional Exploration

- Google API (geographical location to job position)
- Heat map of TechHubs based on # of posted job positions
- Compare values we found on this website versus other job search sites