CS 327E Class 9

April 8, 2019

No Quiz Today :)

Announcements

- What to expect from upcoming Milestones:

  Milestone 9: Find your secondary dataset, load into BQ and model the data with SQL transforms

  Milestone 10: Create Beam pipelines that transform the data

  Milestone 11: Create cross-dataset queries and data visualizations

  Milestone 12: Create workflow with Apache Airflow

  Milestone 13: Present and demo your project

- Review your secondary dataset today in class: http://tinyurl.com/y7d2jzjj

H1B Case Study

Questions:

- How likely are young tech companies to sponsor H1B workers?
- How does the compensation of H1B workers compare to that of domestic workers who are performing the same role and living in same region?

Datasets:

- Main Dataset: H1B applications for years 2015 - 2018 (source: US Dept of Labor)
- Secondary Dataset: Corporate registrations for various states (source: Secretary of States)
- Secondary Dataset: Occupational Employment Survey for years 2015 - 2018 (source: Bureau of Labor Statistics)

H1B Case Study

Cross-Dataset Queries:

- Join H1B's Employer table with the Secretary of State's Corporate Registry table on the employer's name and city. Get the age of the company from the incorporation date in the registry record. Group the employers into age buckets to see how many young tech companies sponsor H1B workers.

- Technical challenges:
  1) matching employers within the H1B dataset due to inconsistent spellings of the company's name
  2) matching employers across H1B and Corporate Registry datasets due to inconsistent spellings of the company's name and address.

Main Dataset

H1B_Applications_2018

Schema | Details | Preview

Field name	Type
CASE_NUMBER	STRING
CASE_STATUS	STRING
CASE_SUBMITTED	DATE
DECISION_DATE	DATE
VISA_CLASS	STRING
EMPLOYMENT_START_DATE	DATE
EMPLOYMENT_END_DATE	DATE
EMPLOYER_NAME	STRING
EMPLOYER_BUSINESS_DBA	STRING
EMPLOYER_ADDRESS	STRING
EMPLOYER_CITY	STRING
EMPLOYER_STATE	STRING
EMPLOYER_POSTAL_CODE	STRING
EMPLOYER_COUNTRY	STRING
EMPLOYER_PROVINCE	STRING
EMPLOYER_PROVINCE	STRING
EMPLOYER_PHONE	STRING
EMPLOYER_PHONE_EXT	STRING
AGENT_REPRESENTING_EMPLOYER	BOOLEAN
AGENT_ATTORNEY_NAME	STRING
AGENT_ATTORNEY_CITY	STRING
AGENT_ATTORNEY_STATE	STRING
JOB_TITLE	STRING
SOC_CODE	STRING
SOC_NAME	STRING
NAICS_CODE	STRING
TOTAL_WORKERS	INTEGER
NEW_EMPLOYMENT	INTEGER
CONTINUED_EMPLOYMENT	INTEGER
CHANGE_PREVIOUS_EMPLOYMENT	INTEGER
NEW_CONCURRENT_EMP	INTEGER
CHANGE_EMPLOYER	INTEGER
AMENDED_PETITION	INTEGER
FULL_TIME_POSITION	STRING
PREVAILING_WAGE	STRING
PW_UNIT_OF_PAY	STRING
PW_WAGE_LEVEL	STRING
PW_SOURCE	STRING
PW_SOURCE_YEAR	STRING
PW_SOURCE_OTHER	STRING
WAGE_RATE_OF_PAY_FROM	STRING
WAGE_RATE_OF_PAY_TO	FLOAT
WAGE_UNIT_OF_PAY	STRING
H1B_DEPENDENT	STRING
WILLFUL_VIOLATOR	BOOLEAN
SUPPORT_H1B	STRING
LABOR_CON_AGREE	STRING
PUBLIC_DISCLOSURE_LOCATION	BOOLEAN
WORKSITE_CITY	STRING
WORKSITE_COUNTY	STRING
WORKSITE_STATE	STRING
WORKSITE_POSTAL_CODE	STRING
ORIGINAL_CERT_DATE	STRING

Raw Table Stats

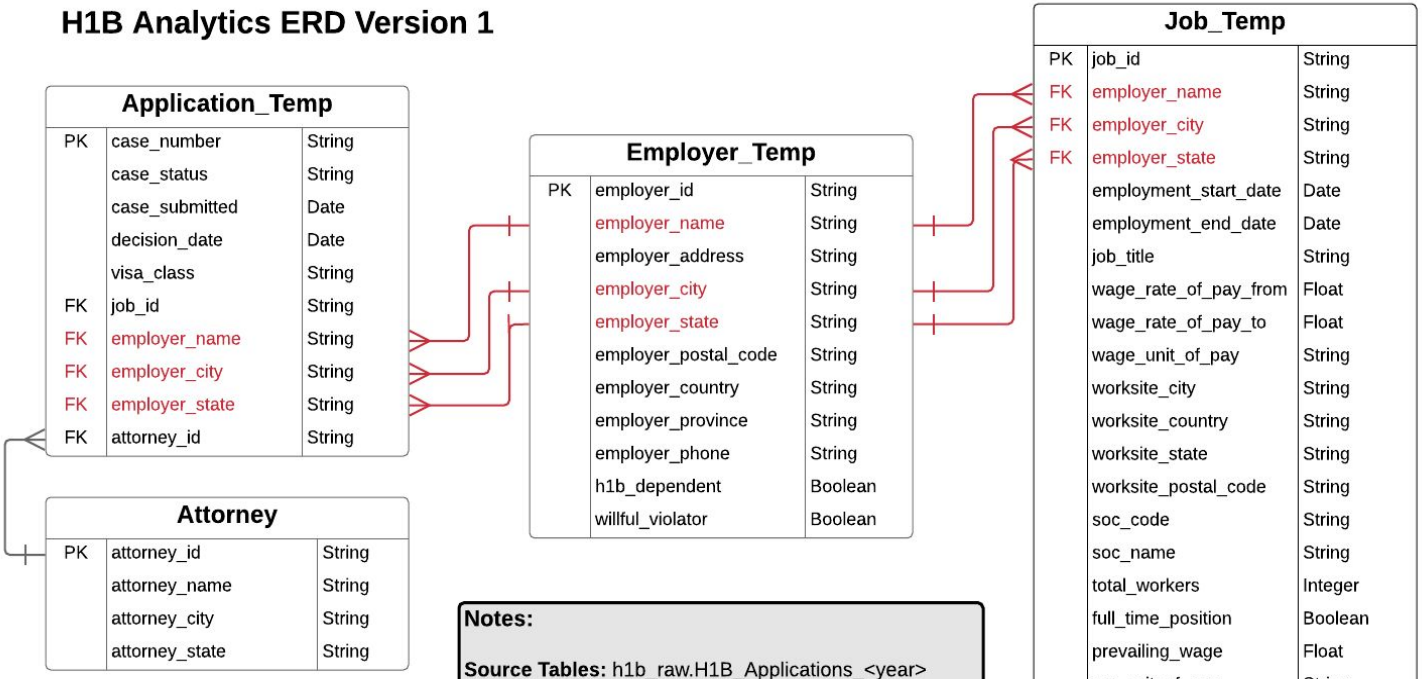
Year	Table Size	# Rows	# Columns
2015	241 MB	618,804	41
2016	233 MB	647,852	41
2017	253 MB	624,650	52
2018	283 MB	654,162	52

SQL Transforms

```
8  CREATE TABLE h1b_split.Employer_Temp AS
9  SELECT generate_uuid() as employer_id, *
10 FROM
11 (SELECT DISTINCT employer_name, employer_address, employer_city, employer_state,
12  employer_postal_code, employer_country, employer_province, CAST(employer_phone AS STRING) as employer_phone,
13  CAST(CASE WHEN h1b_dependent = 'N' THEN 'False'
14  WHEN h1b_dependent = 'Y' THEN 'True'
15  ELSE NULL END as BOOL) AS h1b_dependent,
16  willful_violator
17  FROM `cs327e-fa2018.h1b_raw.H1B_Applications_2018`
18  WHERE employer_name IS NOT NULL AND employer_name != '1' AND employer_city IS NOT NULL
19  UNION DISTINCT
20  SELECT DISTINCT employer_name, employer_address, employer_city, employer_state,
21  employer_postal_code, employer_country, employer_province, employer_phone, h1b_dependent, willful_violator
22  FROM `cs327e-fa2018.h1b_raw.H1B_Applications_2017`
23  WHERE employer_name IS NOT NULL AND employer_name != '1' AND employer_city IS NOT NULL
24  UNION DISTINCT
25  SELECT DISTINCT employer_name, employer_address, employer_city, employer_state,
26  employer_postal_code, employer_country, employer_province, employer_phone, h1b_dependent, willful_violator
27  FROM `cs327e-fa2018.h1b_raw.H1B_Applications_2016`
28  WHERE employer_name IS NOT NULL AND employer_name != '1' AND employer_city IS NOT NULL
29  UNION DISTINCT
30  SELECT DISTINCT employer_name, CONCAT(employer_address1, ' ', employer_address2) as employer_address,
31  employer_city, employer_state, employer_postal_code, employer_country, employer_province, employer_phone,
32  h1b_dependent, willful_violator
33  FROM `cs327e-fa2018.h1b_raw.H1B_Applications_2015`
34  WHERE employer_name IS NOT NULL AND employer_name != '1' AND employer_city IS NOT NULL
35 )
36 ORDER BY employer_name, employer_city;
```

Source File: https://github.com/shirleycohen/h1b_analytics/blob/master/h1b_ctas.sql

H1B Analytics ERD Version 1



Notes:

Source Tables: h1b_raw.H1B_Applications_<year> where <year> = 2015 - 2018.
Each table was loaded from a different CSV file.

Target Tables: Tables showed in this ERD were created in the h1b_split dataset. All tables created and populated from CTAS statements.

Issues with Target Tables:
- Employer_Temp contains duplicate records due to mispellings of the employer name and city.
- Job_Temp and Application_Temp are missing references to Employer table via employer_id.

Beam Pipeline: Employer Table

- Normalizes the employer name, city and state
- Removes duplicate employer records

```
147  with beam.Pipeline('DirectRunner', options=opts) as p:
148
149      query_results = p | 'Read from BigQuery' >> beam.io.Read(beam.io.BigQuerySource(query='SELECT *
150                                                                                          FROM h1b_split.Employer_Temp LIMIT 100'))
151
152      # write PCollection to log file
153      query_results | 'Write to File 1' >> WriteToText('output_query_results.txt')
154
155      # apply ParDo to the Employer records
156      tuple_pcoll = query_results | 'Transform Employer' >> beam.ParDo(TransformEmployer())
157
158      # write PCollection to log file
159      tuple_pcoll | 'Write to File 2' >> WriteToText('output_pardo_employer_tuple.txt')
160
161      deduped_pcoll = tuple_pcoll | 'Dedup Employer Records' >> beam.GroupByKey()
162
163      # write PCollection to log file
164      deduped_pcoll | 'Write to File 3' >> WriteToText('output_group_by_key.txt')
165
166      # apply second ParDo to the PCollection
167      out_pcoll = deduped_pcoll | 'Make BigQuery Records' >> beam.ParDo(MakeBigQueryRecord())
```

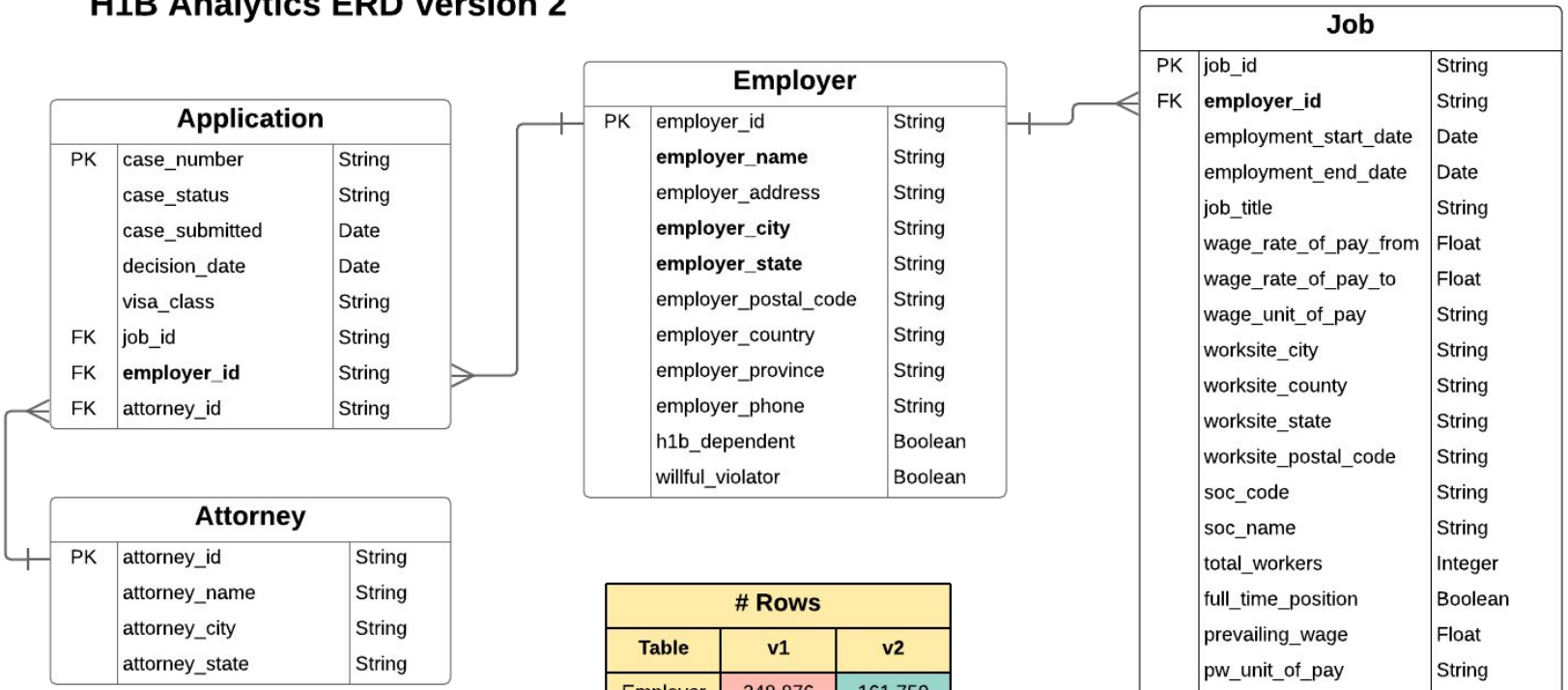
Source Files: https://github.com/shirleycohen/h1b_analytics/blob/master/transform_employer_table_single.py
https://github.com/shirleycohen/h1b_analytics/blob/master/transform_employer_table_cluster.py

Beam Pipelines: Job and Application Tables

- Read the records from the Employer and Job/Application tables in BigQuery and create a `PCollection` from each source
- Normalize the employer's name, city and state from the Job/Application `PCollection` (using `ParDo`)
- Join the Job/Application and Employer `PCollections` on employer's name and city (using `CoGroupByKey`).
- Extract the matching employer_id from the joined results and add it to the Job/Application element (using `ParDo`)
- Remove employer's name and city from the Job/Application `PCollections` (using `ParDo`)
- Write new Job/Application table to BigQuery

Source Files: https://github.com/shirleycohen/h1b_analytics/blob/master/transform_job_table_cluster.py
https://github.com/shirleycohen/h1b_analytics/blob/master/transform_application_table_cluster.py

H1B Analytics ERD Version 2



# Rows		
Table	**v1**	**v2**
Employer	348,876	161,759
Job	2,230,779	2,230,625
Application	2,633,426	2,633,156
Attorney	19,861	19,861

Secondary Dataset

Table Details: Corporate_Registrations_CA

Schema | Details | Preview

so_file_number	STRING
corporation_number	INTEGER
corporation_status	STRING
corporation_classification	STRING
corporation_name	STRING
care_of_name	STRING
mail_address_line_1	STRING
mail_address_line_2	STRING
mail_address_city	STRING
mail_address_state_or_country	STRING
mail_address_zip_code	STRING
corporation_type	STRING
incorporation_date	DATE
so_file_date	DATE
term_expiration_date	DATE
chief_executive_officer_name	STRING

chief_executive_officer_address_line_1	STRING
chief_executive_officer_address_line_2	STRING
chief_executive_officer_address_city	STRING
chief_executive_officer_address_state_or_county	STRING
chief_executive_officer_address_zip_code	STRING
agent_name	STRING
agent_address_line_1	STRING
agent_address_line_2	STRING
agent_address_city	STRING
agent_address_state_or_county	STRING
agent_address_zip_code	STRING
state_or_foreign_country	STRING
ftb_suspension_status	STRING
corporation_tax_base	STRING
transaction_julian_date	DATE
ftb_suspension_string	STRING
filler	STRING

Table Details

State	Size	Rows
AZ	225 MB	869,943
CA	1.1 GB	3,792,457
CO	38 MB	160,808
CT	192 MB	796,877
GA	418 MB	4,139,935
MA	221 MB	1,066,639
MN	1.1 GB	5,761,069
MO	652 MB	4,479,627
NC	262 MB	1,389,877
OH	497 MB	2,408,556
NY	512 MB	2,587,015
VA	111 MB	334,008
WA	205 MB	1,152,30

SQL Transforms

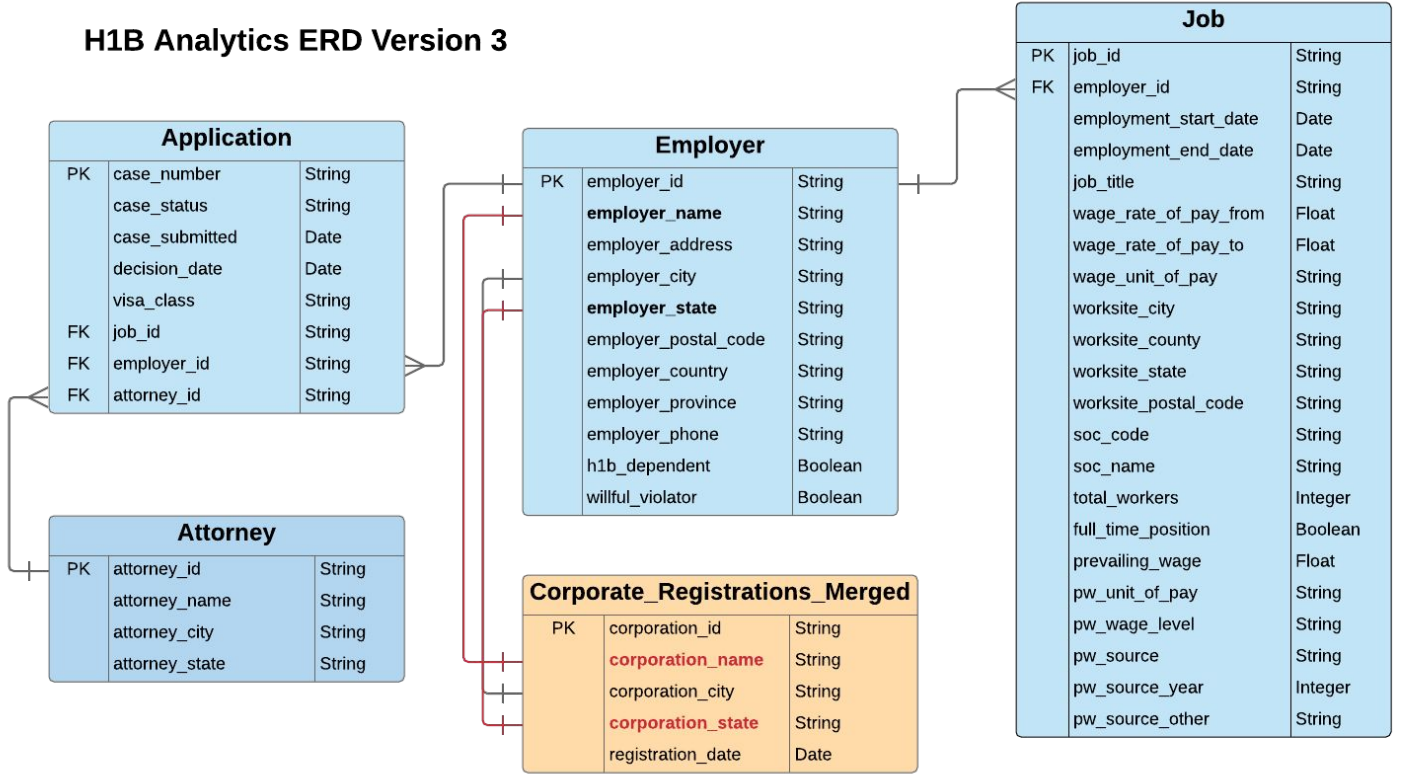
```sql
create table sec_of_state.Corporate_Registrations_Merged
(
        corporation_id STRING,
        corporation_name STRING,
        corporation_city STRING,
        corporation_state STRING,
        registration_date DATE,
        empty_date DATE
)
PARTITION BY empty_date
CLUSTER BY corporation_state;

--AZ
insert into sec_of_state.Corporate_Registrations_Merged (corporation_id, corporation_name, corporation_city,
                                                         corporation_state, registration_date)
select distinct File_Number, Corporation_Name, First_Address_City, 'AZ', Date_of_Incorporation
from sec_of_state.Corporate_Registrations_AZ
where First_Address_State = 'AZ'
order by corporation_name;

--CA
insert into sec_of_state.Corporate_Registrations_Merged (corporation_id, corporation_name, corporation_city,
                                                         corporation_state, registration_date)
select CAST(corporation_number as STRING), corporation_name, mail_address_city, 'CA', incorporation_date
from sec_of_state.Corporate_Registrations_CA
where corporation_type = 'Articles of Incorporation'
and mail_address_state_or_country = 'CA'
order by corporation_name;
```

Source File: https://github.com/shirleycohen/h1b_analytics/blob/master/corporate_registrations_ctas.sql

H1B Analytics ERD Version 3



Notes:

Source Tables:
sec_of_state.Corporate_Registrations_<state>
where <state> = AZ, CA, CO, CT, GA, MA, MN, MO, NC, NY, OH, VA, WA.
Each state table was loaded from a CSV file.
Most of the states had one file, a few had two.

Target Table:
-sec_of_state.Corporate_Registrations_Merged
-created and populated from CTAS statements.
-Table size: 390 MB with 16,379,107 rows.

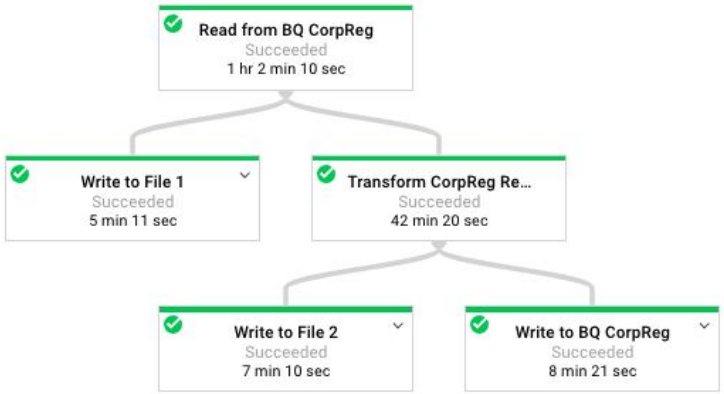
Issues with Target Table:
- punctuation marks found in corporation_name corporation_city values
- suffixes found in corporation_name values (e.g. LLC, INC, etc.)
- **only 2.4% employers matched a corporate registration record.**

Beam Pipeline: Corporate Registrations

```
89  with beam.Pipeline('DataflowRunner', options=opts) as p:
90
91      query_str = 'SELECT corporation_id, corporation_name, corporation_city, corporation_state, registration_date ' \
92                  'FROM `sec_of_state.Corporate_Registrations_Merged` WHERE corporation_name IS NOT NULL ' \
93                  'AND corporation_city IS NOT NULL'
94
95      query_results = p | 'Read Corp Reg' >> beam.io.Read(beam.io.BigQuerySource(query=query_str, use_standard_sql=True))
96
97      query_results | 'Write to File 1' >> WriteToText(DIR_PATH + 'output_query_results.txt')
98
99      clean_pcoll = query_results | 'Transform Corp Reg Record' >> beam.ParDo(TransformCorpRegRecord())
100
101     clean_pcoll | 'Write to File 2' >> WriteToText(DIR_PATH + 'output_bq_records.txt')
102
103     qualified_table_name = PROJECT_ID + ':sec_of_state.Corporate_Registrations_Cleaned'
104     table_schema = 'corporation_id:STRING,corporation_name:STRING,corporation_city:STRING,corporation_state:STRING,' \
105                    'registration_date:DATE'
106
107     clean_pcoll | 'Write Corp Reg' >> beam.io.Write(beam.io.BigQuerySink(qualified_table_name,
108                                                     schema=table_schema,
109                                                     create_disposition=beam.io.BigQueryDisposition.CREATE_NEVER,
110                                                     write_disposition=beam.io.BigQueryDisposition.WRITE_TRUNCATE))
111
```

Source File: https://github.com/shirleycohen/h1b_analytics/blob/master/transform_corpreg_table_cluster.py

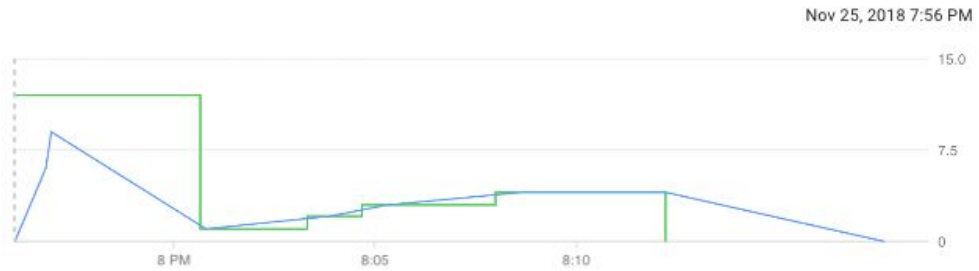
Dataflow Execution



Job summary

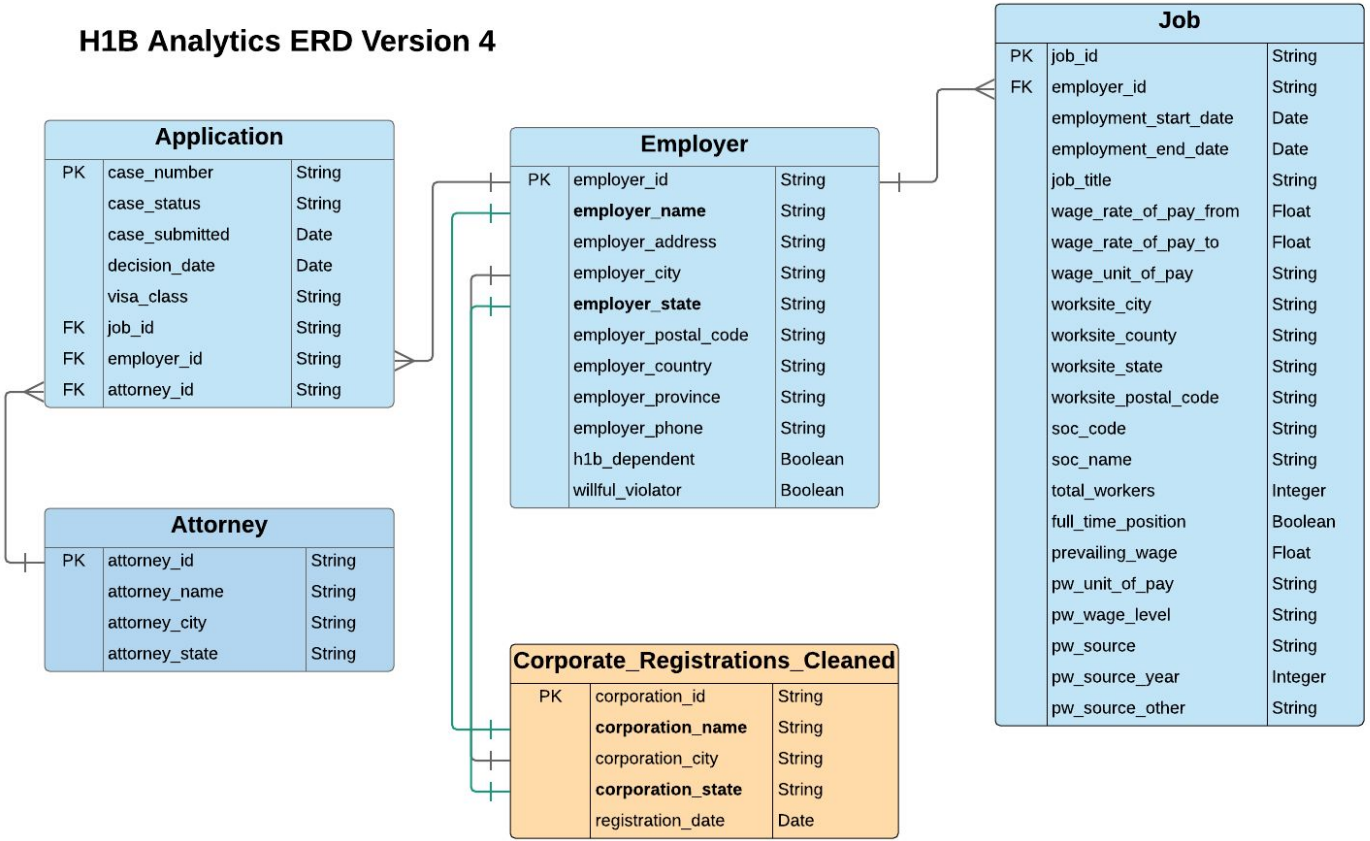
Job name	transform-corp-reg-table
Job ID	2018-11-25_17_55_55-2850952765719790096
Region	us-central1
Job status	Succeeded
SDK version	Google Cloud Dataflow SDK for Python 2.5.0
Job type	Batch
Start time	Nov 25, 2018, 7:55:57 PM
Elapsed time	16 min 20 sec

Worker history



Target workers	Timestamp	Rationale
0	Nov 25, 2018, 8:08:08 PM	Stopping worker pool.
4	Nov 25, 2018, 8:04:58 PM	Autoscaling: Raised the number of workers to 4 based on the rate of progress in the currently running step(s).
3	Nov 25, 2018, 8:02:28 PM	Autoscaling: Raised the number of workers to 3 based on the rate of progress in the currently running step(s).
2	Nov 25, 2018, 8:01:28 PM	Autoscaling: Raised the number of workers to 2 based on the rate of progress in the currently running step(s).
1	Nov 25, 2018, 7:59:28 PM	Autoscaling: Reduced the number of workers to 1 based on the rate of progress in the currently running step(s).
12	Nov 25, 2018, 7:56:02 PM	Starting a pool of 12 workers.

H1B Analytics ERD Version 4



Notes:

New Source Tables:
sec_of_state.Corporate_Registrations_Merged.

New Target Table:
-sec_of_state.Corporate_Registrations_Cleaned.
-generated from Beam pipeline.

Changes since previous version:
- removed punctuation marks and suffixes from corporation_name.
- **percentage of employers with corp registration matches increased to 40%**

# Table Rows		
	v1	**v2**
Corporate_Registrations	16,379,107	16,321,932
Employer	348,876	161,759
v_Tech_Employer_13_States	N/A	31,758

Cross-Dataset Queries

v_Tech_Employer_Age:

- Joins Employer and Corporate Registrations on name and state
- Calculates age of employer from registration_date

v_Tech_Employer_Age_Label:

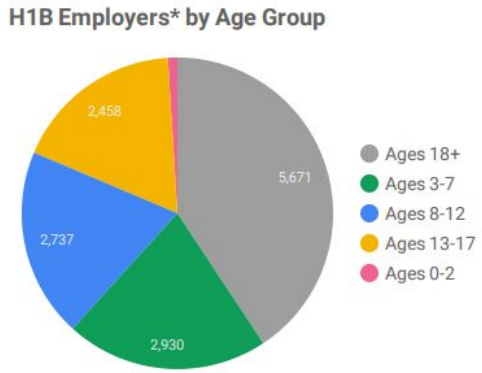
- Assigns a label to the employer based on their age range (0, 1-2, 3-12, 13-17, 18+)

v_Tech_Employer_Age_Label_report:

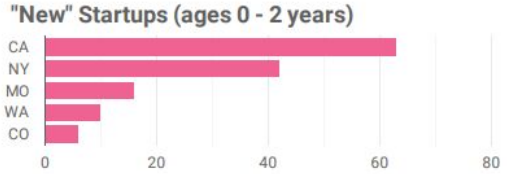
- Groups employers by age label and state combination
- Calculates employer count per group

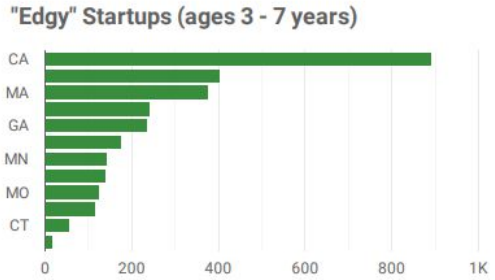
Source File: https://github.com/shirleycohen/h1b_analytics/blob/master/employer_views.sql

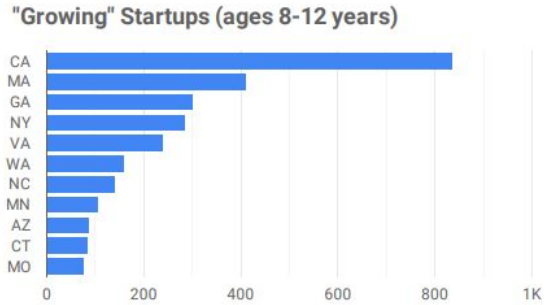
Data Studio Report

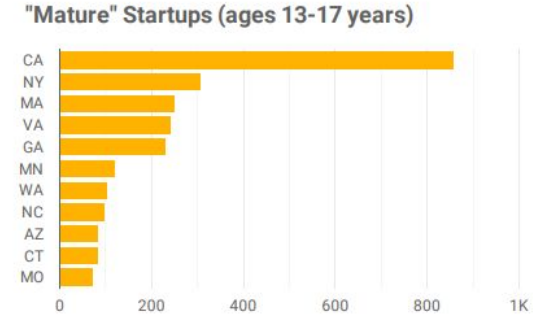


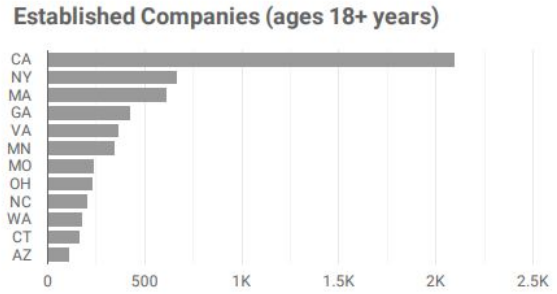
*Only includes employers who sponsor H1B workers in technical roles.











Milestone 9

http://www.cs.utexas.edu/~scohen/milestones/Milestone9.pdf