CS 327E Class 11

December 3, 2018
Announcements

- Group presentations start next Monday!
- Presentation schedule is [posted](#)
- CIS is open, take survey by next Monday. Your voice matters :)}
1) Which representation of a Sales_Order uses a denormalized structure?


B. `Sales_Order(order_id:STRING, cust_id:INTEGER, timestamp:DATETIME, location:STRING); Purchase_Item(sku:INTEGER, description:STRING, quantity:INTEGER, price:FLOAT, order_id:STRING); Customer(cust_id:INTEGER, cust_name:STRING, cust_email:STRING);`
2) What is the main advantage of *denormalized* structures?

A. They reduce query time, especially on large tables.
B. They require less storage than normalized structures.
C. They are more performant for update and delete operations.
3) What is the main disadvantage of \textit{denormalized} structures?

A. They can only store JSON data in a table.
B. They make it more challenging to maintain data integrity.
C. They can’t express $m:n$ relationships between tables.
4) Which BigQuery operator lets you access individual elements of a repeated field?

A. ARRAY
B. FLATTEN
C. UNNEST
5) What does the **view switching** technique require?

A. Masking duplicate records of a table.
B. Maintaining two copies of data during update cycles.
C. Capturing each insert, update, and delete operation in a separate history table.
Case Study: Part 3
H1B Analytics ERD Version 4

**Application**
- PK case_number: String
- case_status: String
- case_submitted: Date
- decision_date: Date
- visa_class: String
- job_id: String
- employer_id: String
- attorney_id: String

**Employer**
- PK employer_id: String
- employer_name: String
- employer_address: String
- employer_city: String
- employer_state: String
- employer_postal_code: String
- employer_country: String
- employer_province: String
- employer_phone: String
- h1b_dependent: Boolean
- willful_violator: Boolean

**Attorney**
- PK attorney_id: String
- attorney_name: String
- attorney_city: String
- attorney_state: String

**Corporate_Registrations_Cleaned**
- PK corporation_id: String
- corporation_name: String
- corporation_city: String
- corporation_state: String
- registration_date: Date

**Job**
- PK job_id: String
- employment_start_date: Date
- employment_end_date: Date
- job_title: String
- wage_rate_of_pay_from: Float
- wage_rate_of_pay_to: Float
- wage_unit_of_pay: String
- worksite_city: String
- worksite_county: String
- worksite_state: String
- worksite_postal_code: String
- soc_code: String
- soc_name: String
- total_workers: Integer
- full_time_position: Boolean
- prevailing_wage: Float
- pw_unit_of_pay: String
- pw_wage_level: String
- pw_source: String
- pw_source_year: Integer
- pw_source_other: String

**Notes:**
- New Target Table: generated from Beam pipeline.
- Changes since previous version:
  - Removed punctuation marks and suffixes from corporation_name.
  - Performed simple validation of corporation_city.
  - Cross-dataset join returns 12,856 results (instead of only 804 results).

<table>
<thead>
<tr>
<th>Number of Rows</th>
<th>v1</th>
<th>v2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corporate_Registrations</td>
<td>16,379,107</td>
<td>16,321,932</td>
</tr>
<tr>
<td>Employer</td>
<td>348,876</td>
<td>161,759</td>
</tr>
<tr>
<td>v_Tech_Employer_13_States</td>
<td>29,658</td>
<td></td>
</tr>
</tbody>
</table>
Third Dataset
### Wages Table Details:
2015: 29.2 MB size, 473,717 rows
2016: 29.9 MB size, 484,390 rows
2017: 29.9 MB size, 484,390 rows
2018: 29.9 MB size, 485,211 rows

<table>
<thead>
<tr>
<th>Row</th>
<th>Area</th>
<th>SocCode</th>
<th>GeoLvl</th>
<th>Level1</th>
<th>Level2</th>
<th>Level3</th>
<th>Level4</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>485200</td>
<td>5100003</td>
<td>27-1022</td>
<td>4</td>
<td>18.57</td>
<td>28.24</td>
<td>37.92</td>
<td>47.59</td>
<td>37.92</td>
</tr>
<tr>
<td>485201</td>
<td>5100004</td>
<td>27-1022</td>
<td>4</td>
<td>18.57</td>
<td>28.24</td>
<td>37.92</td>
<td>47.59</td>
<td>37.92</td>
</tr>
<tr>
<td>485202</td>
<td>5400001</td>
<td>27-1022</td>
<td>4</td>
<td>18.57</td>
<td>28.24</td>
<td>37.92</td>
<td>47.59</td>
<td>37.92</td>
</tr>
<tr>
<td>485203</td>
<td>5400002</td>
<td>27-1022</td>
<td>4</td>
<td>18.57</td>
<td>28.24</td>
<td>37.92</td>
<td>47.59</td>
<td>37.92</td>
</tr>
<tr>
<td>485204</td>
<td>6600001</td>
<td>27-1022</td>
<td>4</td>
<td>18.57</td>
<td>28.24</td>
<td>37.92</td>
<td>47.59</td>
<td>37.92</td>
</tr>
<tr>
<td>485205</td>
<td>73050</td>
<td>27-1022</td>
<td>4</td>
<td>18.57</td>
<td>28.24</td>
<td>37.92</td>
<td>47.59</td>
<td>37.92</td>
</tr>
<tr>
<td>485206</td>
<td>74950</td>
<td>27-1022</td>
<td>4</td>
<td>18.57</td>
<td>28.24</td>
<td>37.92</td>
<td>47.59</td>
<td>37.92</td>
</tr>
</tbody>
</table>

### Geography Table Details:
2015: 340 KB size, 4,765 rows
2016: 357 KB size, 4,991 rows
2017: 357 KB size, 4,991 rows
2018: 357 KB size, 4,991 rows

<table>
<thead>
<tr>
<th>Row</th>
<th>Area</th>
<th>AreaName</th>
<th>StateAb</th>
<th>State</th>
<th>CountyTownName</th>
</tr>
</thead>
<tbody>
<tr>
<td>4416</td>
<td>71654</td>
<td>Boston-Cambridge-Newton, MA NECTA Division</td>
<td>MA</td>
<td>MASSACHUSETTS</td>
<td>NORFOLK (STOUGHTON)</td>
</tr>
<tr>
<td>4417</td>
<td>71654</td>
<td>Boston-Cambridge-Newton, MA NECTA Division</td>
<td>MA</td>
<td>MASSACHUSETTS</td>
<td>NORFOLK (FRANKLIN)</td>
</tr>
<tr>
<td>4418</td>
<td>71654</td>
<td>Boston-Cambridge-Newton, MA NECTA Division</td>
<td>MA</td>
<td>MASSACHUSETTS</td>
<td>NORFOLK (MEDWAY)</td>
</tr>
<tr>
<td>4419</td>
<td>71654</td>
<td>Boston-Cambridge-Newton, MA NECTA Division</td>
<td>MA</td>
<td>MASSACHUSETTS</td>
<td>NORFOLK (NORWOOD)</td>
</tr>
<tr>
<td>4420</td>
<td>71654</td>
<td>Boston-Cambridge-Newton, MA NECTA Division</td>
<td>MA</td>
<td>MASSACHUSETTS</td>
<td>NORFOLK (CANTON)</td>
</tr>
<tr>
<td>4421</td>
<td>71654</td>
<td>Boston-Cambridge-Newton, MA NECTA Division</td>
<td>MA</td>
<td>MASSACHUSETTS</td>
<td>NORFOLK (DEDHAM)</td>
</tr>
<tr>
<td>4422</td>
<td>71654</td>
<td>Boston-Cambridge-Newton, MA NECTA Division</td>
<td>MA</td>
<td>MASSACHUSETTS</td>
<td>NORFOLK (DOVER)</td>
</tr>
</tbody>
</table>
SQL Transforms

- Unions the yearly staging tables for Wages and Geography (2015 - 2018)
- Excludes all unwanted fields from result tables
- Calculates annual salaries from average wages per occupation and geo area

```sql
create table bureau_labor_stats.All_Industries_Wages
(
    area INT64,
    year INT64,
    soc_code STRING,
    annual_salary FLOAT64,
    empty_date DATE
)
PARTITION BY empty_date
CLUSTER BY year;

insert into bureau_labor_stats.All_Industries_Wages (area, year, soc_code, annual_salary) select Area, 2015, SocCode,
    (case when Average < 300 then round(((Average*8)/365), 2)
    when Average > 15000 then round(Average, 2)
    else NULL end)
from bureau_labor_stats.All_Industries_Wages_2015;

insert into bureau_labor_stats.All_Industries_Wages (area, year, soc_code, annual_salary) select Area, 2016, SocCode,
    (case when Average < 300 then round(((Average*8)/365), 2)
    when Average > 15000 then round(Average, 2)
    else NULL end)
from bureau_labor_stats.All_Industries_Wages_2016;

create table bureau_labor_stats.Geography
(
    area INT64,
    year INT64,
    state STRING,
    county STRING,
    empty_date DATE
)
PARTITION BY empty_date
CLUSTER BY year;

insert into bureau_labor_stats.Geography (area, year, state, county) select Area, 2015, StateAb, CountyTownName
from bureau_labor_stats.Geography_2015;

insert into bureau_labor_stats.Geography (area, year, state, county) select Area, 2016, StateAb, CountyTownName
from bureau_labor_stats.Geography_2016;

insert into bureau_labor_stats.Geography (area, year, state, county) select Area, 2017, StateAb, CountyTownName
from bureau_labor_stats.Geography_2017;
```
Data Integrity Issues

New Query

```
select * from hlb_split.Job where worksite_county like '% COUNTY'
```

Query complete (4.6s elapsed, 563 MB processed)

<table>
<thead>
<tr>
<th>worksite_city</th>
<th>worksite_county</th>
<th>worksite_state</th>
<th>worksite_postal_code</th>
<th>soc_code</th>
<th>soc_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOUSTON</td>
<td>HARRIS COUNTY</td>
<td>TX</td>
<td>77027</td>
<td>19-4081</td>
<td>SOCIAL SCIENCE RESEARCH ASSISTANT</td>
</tr>
<tr>
<td>BROOKLYN</td>
<td>KINGS COUNTY</td>
<td>NY</td>
<td>11201</td>
<td>27-3031</td>
<td>PUBLIC RELATIONS SPECIALISTS</td>
</tr>
<tr>
<td>LAGUNA BEACH</td>
<td>ORANGE COUNTY</td>
<td>CA</td>
<td>92651</td>
<td>17-1012</td>
<td>LANDSCAPE ARCHITECTS</td>
</tr>
<tr>
<td>HUNTINGTON BEACH</td>
<td>ORANGE COUNTY</td>
<td>CA</td>
<td>92648</td>
<td>23-1012</td>
<td>JUDICIAL LAW CLERKS</td>
</tr>
<tr>
<td>SEATTLE</td>
<td>KING COUNTY</td>
<td>WA</td>
<td>98125</td>
<td>19-3092</td>
<td>GEOGRAPHERS</td>
</tr>
<tr>
<td>ONTARIO</td>
<td>SAN BERNARDINO COUNTY</td>
<td>CA</td>
<td>91761</td>
<td>13-2031</td>
<td>BUDGET ANALYSTS</td>
</tr>
<tr>
<td>MIAMI LAKES</td>
<td>DADE COUNTY</td>
<td>FL</td>
<td>33014</td>
<td>23-2099</td>
<td>LEGAL SUPPORT WORKERS, ALL OTHER</td>
</tr>
<tr>
<td>THE WOODLANDS</td>
<td>MONTGOMERY COUNTY</td>
<td>TX</td>
<td>77380</td>
<td>13-1051</td>
<td>COST ESTIMATORS</td>
</tr>
</tbody>
</table>
# Data Integrity Issues

## New Query

```sql
select * from hlb_split.Job where length(worksite_country) = 2
```

## Results

<table>
<thead>
<tr>
<th>age_unit_of_pay</th>
<th>worksite_city</th>
<th>worksite_country</th>
<th>worksite_state</th>
<th>worksite_postal_code</th>
<th>soc_code</th>
<th>soc_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>our</td>
<td>SAIPAN</td>
<td>MP</td>
<td>MP</td>
<td>96950</td>
<td>11-9041</td>
<td>ARCHITECTURAL AND ENGINEERING M.</td>
</tr>
<tr>
<td>SAR</td>
<td>NEW YORK</td>
<td>NY</td>
<td>NY</td>
<td>10016</td>
<td>41-4011</td>
<td>SALES REPRESENTATIVES, WHOLESALE</td>
</tr>
<tr>
<td>SAR</td>
<td>SAN FRANCISCO</td>
<td>CA</td>
<td>CA</td>
<td>94103</td>
<td>41-9031</td>
<td>SALES ENGINEERS</td>
</tr>
<tr>
<td>SAR</td>
<td>WORCESTER</td>
<td>MA</td>
<td>MA</td>
<td>1611</td>
<td>25-1124</td>
<td>FOREIGN LANGUAGE AND LITERATURE</td>
</tr>
<tr>
<td>north</td>
<td>WENATCHEE</td>
<td>WA</td>
<td>WA</td>
<td>98801</td>
<td>41-4012</td>
<td>SALES REPRESENTATIVES, WHOLESALE</td>
</tr>
<tr>
<td>SAR</td>
<td>HAUPPAUGE</td>
<td>NY</td>
<td>NY</td>
<td>11788</td>
<td>19-2031</td>
<td>CHEMISTS</td>
</tr>
<tr>
<td>our</td>
<td>FAIRBANKS</td>
<td>AK</td>
<td>AK</td>
<td>95712</td>
<td>27-2022</td>
<td>COACHES AND SCOUTS</td>
</tr>
<tr>
<td>SAR</td>
<td>NEW YORK</td>
<td>NY</td>
<td>NY</td>
<td>10013</td>
<td>27-2012</td>
<td>PRODUCERS AND DIRECTORS</td>
</tr>
</tbody>
</table>
with beam.Pipeline('DataflowRunner', options=opts) as p:

    job_query_str = 'SELECT *, EXTRACT(YEAR FROM employment_start_date) AS employment_start_year' \
    ' FROM `h1b_split.JobTemp` ORDER BY employer_name'

    emp_query_str = 'SELECT employer_id, employer_name, employer_city FROM ' \
    ' `h1b_split.Employer` ORDER BY employer_name'

    job_query_results = p | 'Read from BigQuery Job' >> beam.io.Read(beam.io.BigQuerySource(query=job_query_str, use_standard_sql=True))
    emp_query_results = p | 'Read from BigQuery Employer' >> beam.io.Read(beam.io.BigQuerySource(query=emp_query_str, use_standard_sql=True))

    # apply ParDo to the Job records
    job_tuple_pcoll = job_query_results | 'Transform Job Record' >> beam.ParDo(TransformJobRecord())
    emp_tuple_pcoll = emp_query_results | 'Transform Employer Record' >> beam.ParDo(TransformEmployerRecord())

    # remove suffix from worksite_county
    if worksite_county != None:
        worksite_county = worksite_county.replace(' COUNTY', '')
        job_record['worksite_county'] = worksite_county

    # county and state should not be equal to each other
    if worksite_county == worksite_state:
        worksite_county = None
        job_record.pop('worksite_county')
    if worksite_county != None and worksite_county.isdigit():
        worksite_county = None
        job_record.pop('worksite_county')
Dataflow Job

Job summary
- Job name: transform-job-table
- Job ID: 2018-11-30_12_41_02-01-1572724817649036704
- Region: us-central1
- Job status: Succeeded
- SDK version: Google Cloud Dataflow SDK for Python 2.5.0
- Job type: Batch
- Start time: Nov 30, 2018, 2:41:03 PM
- Elapsed time: 14 min 42 sec

Autoscaling
- Workers: 0
- Current state: Worker pool stopped.

See more history
## Cross-Dataset Query

### New Query

```sql
SELECT j.job_id, j.worksite_county, g.county, w.area, j.worksite_state, g.state, j.soc_code, j.soc_name, j.employment_start_year as year, j.wage_rate_of_pay_from as job_salary, w.annual_salary as national_salary
FROM hlb_split.v_Tech_Hub j JOIN bureau_labor_stats.GeoLocation g
ON (j.worksite_county = g.county AND j.employment_start_year = g.year)
JOIN bureau_labor_stats.All_Industries_Wages w
ON (g.area = w.area AND g.year = w.year)
WHERE j.soc_code = w.soc_code AND j.employment_start_year = w.year
ORDER BY j.job_id
```

### Results

<table>
<thead>
<tr>
<th>job_id</th>
<th>worksite_county</th>
<th>county</th>
<th>area</th>
<th>worksite_state</th>
<th>state</th>
<th>soc_code</th>
<th>soc_name</th>
<th>year</th>
<th>job_salary</th>
<th>national_salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000035e-b44f-4785-b204-a8d10e4e8857</td>
<td>TOMPKINS</td>
<td>TOMPKINS</td>
<td>27080</td>
<td>NY</td>
<td>NY</td>
<td>15-1131</td>
<td>COMPUTER PROGRAMMERS</td>
<td>2016</td>
<td>64000.0</td>
<td>108850.4</td>
</tr>
<tr>
<td>000013d8-82cd-4ba5-8d17-cbab48d9d5b6</td>
<td>KING</td>
<td>KING</td>
<td>42644</td>
<td>WA</td>
<td>WA</td>
<td>15-1131</td>
<td>COMPUTER SYSTEMS ANALYSTS</td>
<td>2017</td>
<td>84000.0</td>
<td>139138.0</td>
</tr>
<tr>
<td>000013d8-82cd-4ba5-8d17-cbab48d9d5b6</td>
<td>KING</td>
<td>KING</td>
<td>480001</td>
<td>WA</td>
<td>TX</td>
<td>15-1131</td>
<td>COMPUTER SYSTEMS ANALYSTS</td>
<td>2017</td>
<td>84000.0</td>
<td>95805.2</td>
</tr>
<tr>
<td>00001e51-1308-46ff-9fed-e8173fb62d54</td>
<td>MARICOPA</td>
<td>MARICOPA</td>
<td>38080</td>
<td>AZ</td>
<td>AZ</td>
<td>15-1132</td>
<td>SOFTWARE DEVELOPERS, APPLICATIONS</td>
<td>2017</td>
<td>92435.0</td>
<td>132655.6</td>
</tr>
<tr>
<td>0000287e-c470-4130-858e-d428865b70ce</td>
<td>BENTON</td>
<td>BENTON</td>
<td>22220</td>
<td>AR</td>
<td>AR</td>
<td>15-1131</td>
<td>COMPUTER PROGRAMMERS</td>
<td>2018</td>
<td>65187.0</td>
<td>92534.8</td>
</tr>
<tr>
<td>0000287e-c470-4130-858e-d428865b70ce</td>
<td>BENTON</td>
<td>BENTON</td>
<td>28200</td>
<td>IN</td>
<td>IN</td>
<td>15-1131</td>
<td>COMPUTER PROGRAMMERS</td>
<td>2018</td>
<td>65187.0</td>
<td>91658.8</td>
</tr>
<tr>
<td>0000287e-c470-4130-858e-d428865b70ce</td>
<td>BENTON</td>
<td>BENTON</td>
<td>41060</td>
<td>AR</td>
<td>MN</td>
<td>15-1131</td>
<td>COMPUTER PROGRAMMERS</td>
<td>2018</td>
<td>65187.0</td>
<td>127487.2</td>
</tr>
<tr>
<td>0000287e-c470-4130-858e-d428865b70ce</td>
<td>BENTON</td>
<td>BENTON</td>
<td>290001</td>
<td>AR</td>
<td>MO</td>
<td>15-1131</td>
<td>COMPUTER PROGRAMMERS</td>
<td>2016</td>
<td>65187.0</td>
<td>100185.2</td>
</tr>
</tbody>
</table>
H1B Analytics ERD Version 7

Application
- PK: case_number, case_status, case_submitted, decision_date, visa_class
- FK: job_id, employer_id, attorney_id

Employer
- PK: employer_id
- FK: employment_start_year, employment_end_date, job_title, wage_rate_of_pay, wage_unit_of_pay, worksite_city, worksite_county, worksite_state, worksite_postal_code

Attorney
- PK: attorney_id, attorney_name, attorney_city, attorney_state

Corporate_Registrations_Cleaned
- PK: corporation_id, corporation_name, corporation_city, corporation_state, registration_date

Job
- PK: job_id
- FK: employer_id

All_Industries_Wages
- PK: area, year
- PK: soc_code, annual_salary

Geography
- PK: area, year, county, state

Notes:
1) *year. All_Industries_Wages joins on year. Geography and Job.employment_start_year.
2) county names not unique => join Geography and Job tables on county and state.
Revised Cross-Dataset Query

```sql
SELECT j.job_id, j.worksite_county, g.county, w.area, j.worksite_state, g.state,
j.soc_code, j.soc_name, j.employment_start_year as year, j.wage_rate_of_pay_from as job_salary,
w.annual_salary as national_salary
FROM hib_split_v tech_job j JOIN bureau_labor_stats.Geography g
ON (j.worksite_county = g.county AND j.worksite_state = g.state AND j.employment_start_year = g.year)
JOIN bureau_labor_stats.All Industries Wages w
ON (g.area = w.area and g.year = w.year)
WHERE j.soc_code = w.soc_code AND j.employment_start_year = w.year
ORDER BY j.job_id
```

### Results

<table>
<thead>
<tr>
<th>row</th>
<th>job_id</th>
<th>worksite_county</th>
<th>county</th>
<th>area</th>
<th>worksite_state</th>
<th>state</th>
<th>soc_code</th>
<th>soc_name</th>
<th>job_salary</th>
<th>year</th>
<th>national_salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0000035e-b44f-47b5-b204-a8d10e48857</td>
<td>TOMPKINS</td>
<td>TOMPKINS</td>
<td>27060</td>
<td>NY</td>
<td>NY</td>
<td>15-1131</td>
<td>COMPUTER PROGRAMMERS</td>
<td>6400.00</td>
<td>2016</td>
<td>109850.4</td>
</tr>
<tr>
<td>2</td>
<td>0000138b-820d-4ba5-8d17-cabab96d85b6</td>
<td>KING</td>
<td>KING</td>
<td>42644</td>
<td>WA</td>
<td>WA</td>
<td>15-1121</td>
<td>COMPUTER SYSTEMS ANALYSTS</td>
<td>8400.00</td>
<td>2017</td>
<td>139138.0</td>
</tr>
<tr>
<td>3</td>
<td>00001e5f-130f-4a0f-8f6e-a8713b62b54</td>
<td>MARICOPA</td>
<td>MARICOPA</td>
<td>38060</td>
<td>AZ</td>
<td>AZ</td>
<td>15-1132</td>
<td>SOFTWARE DEVELOPERS, APPLICATIONS</td>
<td>92435.0</td>
<td>2017</td>
<td>132655.6</td>
</tr>
<tr>
<td>4</td>
<td>0000287a-c070-4110-858e-d42886b70ce</td>
<td>BENTON</td>
<td>BENTON</td>
<td>22220</td>
<td>AR</td>
<td>AR</td>
<td>15-1131</td>
<td>COMPUTER PROGRAMMERS</td>
<td>65187.0</td>
<td>2018</td>
<td>92934.8</td>
</tr>
<tr>
<td>5</td>
<td>000029ac-3e3a-417c-a5a9-3a281d116483</td>
<td>ERIE</td>
<td>ERIE</td>
<td>15380</td>
<td>NY</td>
<td>NY</td>
<td>11-3021</td>
<td>COMPUTER AND INFORMATION SYSTEMS MANAGERS</td>
<td>105000.00</td>
<td>2016</td>
<td>166751.2</td>
</tr>
<tr>
<td>6</td>
<td>00004036-4c15-4ebe-b570-ce7770c198ba3</td>
<td>LARIMER</td>
<td>LARIMER</td>
<td>22680</td>
<td>CO</td>
<td>CO</td>
<td>15-1132</td>
<td>SOFTWARE DEVELOPERS, APPLICATIONS</td>
<td>77500.00</td>
<td>2017</td>
<td>140650.2</td>
</tr>
<tr>
<td>7</td>
<td>000043af-3f63-4b89-a8e8-cd2440864c4</td>
<td>COOK</td>
<td>COOK</td>
<td>16974</td>
<td>IL</td>
<td>IL</td>
<td>15-1134</td>
<td>WEB DEVELOPERS</td>
<td>63419.00</td>
<td>2017</td>
<td>110697.2</td>
</tr>
<tr>
<td>8</td>
<td>00005496-a905-411a-4af1-684e6a9628f</td>
<td>SAN FRANCISCO</td>
<td>SAN FRANCISCO</td>
<td>41884</td>
<td>CA</td>
<td>CA</td>
<td>15-1121</td>
<td>COMPUTER SYSTEMS ANALYSTS</td>
<td>125000.00</td>
<td>2017</td>
<td>167403.6</td>
</tr>
</tbody>
</table>
Cross-Dataset Views

v_Tech_Job:
● Filters out all non-tech jobs from the Job table

v_Tech_Job_Salary_Comparison:
● 3-table join between Job, Geography, and Wages
● Calculates salary delta between h1b job and national average

v_Tech_Job_Salary_Delta_by_Occupation:
● Groups jobs by occupation
● Calculates salary delta as h1b job - national average

Source File: https://github.com/shirleycohen/h1b_analytics/blob/master/job_salary_views.sql
Data Studio Report

Salary delta between H1B jobs and national average

Number of H1B jobs with lower pay than national average

Number of H1B jobs with higher pay than national average

CA
salary_delta: 259,918
Data Studio Report

Pay Gaps by Occupation:

Occupations which pay H1B workers **higher** than domestic workers

<table>
<thead>
<tr>
<th>Occupation</th>
<th>h1b_job_pay</th>
<th>national_average_pay</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPUTER SCIENCE TEACHER, POSTSECONDARY</td>
<td>Blue</td>
<td>Red</td>
</tr>
<tr>
<td>COMPUTER SUPPORT SPECIALISTS</td>
<td>Blue</td>
<td>Red</td>
</tr>
<tr>
<td>COMPUTER NETWORK SUPPORT SPECIALIST</td>
<td>Blue</td>
<td>Red</td>
</tr>
<tr>
<td>COMPUTER SOFTWARE ENGINEERS/ARCHITECTS</td>
<td>Blue</td>
<td>Red</td>
</tr>
</tbody>
</table>

Occupations which pay H1B workers **lower** than domestic workers

<table>
<thead>
<tr>
<th>Occupation</th>
<th>h1b_job_pay</th>
<th>national_average_pay</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPUTER SYSTEMS ANALYSTS</td>
<td>Blue</td>
<td>Red</td>
</tr>
<tr>
<td>PROGRAMMER ANALYST</td>
<td>Blue</td>
<td>Red</td>
</tr>
<tr>
<td>COMPUTER PROGRAMMERS</td>
<td>Blue</td>
<td>Red</td>
</tr>
<tr>
<td>COMPUTER INFORMATION SYSTEMS MANAGERS</td>
<td>Blue</td>
<td>Red</td>
</tr>
</tbody>
</table>
View Switching Technique

Using v_Tech_Job_Salary_Comparison as example:

- Create “shadow” tables from “main” Job, Geography, and Wages tables
- Populate “shadow” tables with new records (e.g. 2019) and changed records for time period 2015 - 2018
- Create “shadow” view that points to “shadow” tables
- Verify correctness of “shadow” tables by querying the “shadow” view
- Promote “shadow” tables to “main” tables
- Recreate “main” view to point to new “main” tables
Milestone 9

Milestone 10