Class 5 BigQuery

Elements of Databases

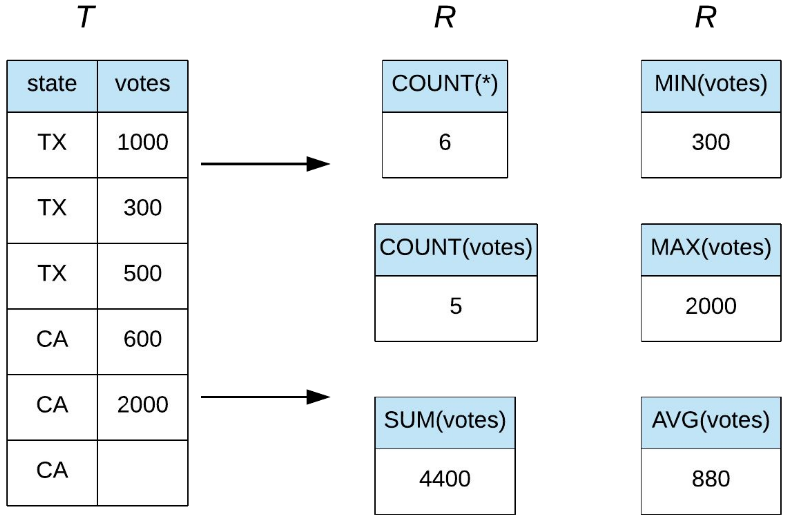
Sept 24, 2021

Global Aggregate Queries

```
SELECT <aggregate function>
      [, <aggregate function>]
FROM <single table>
[JOIN <single table>
 ON <join condition>]
[WHERE <boolean condition>]
~~ORDER BY <field(s) to sort on>~~
```

Global Aggregate Queries

```
SELECT <aggregate function>
      [, <aggregate function>]
FROM <single table>
[JOIN <single table>
 ON <join condition>]
[WHERE <boolean condition>]
~~ORDER BY <field(s) to sort on>~~
```



T

state	votes
TX	1000
TX	300
TX	500
CA	600
CA	2000
CA	

R

COUNT(*)
6

COUNT(votes)
5

SUM(votes)
4400

R

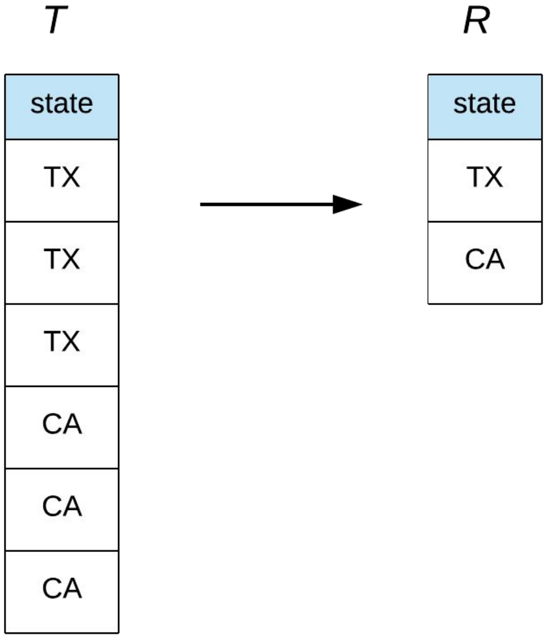
MIN(votes)
300

MAX(votes)
2000

AVG(votes)
880

Group By Queries

```
SELECT <unaggregated field(s)>
FROM <single table>
[JOIN <single table>
ON <join condition>]
[WHERE <boolean condition>]
GROUP BY <unaggregated field(s)>
```



T

state
TX
TX
TX
CA
CA
CA

R

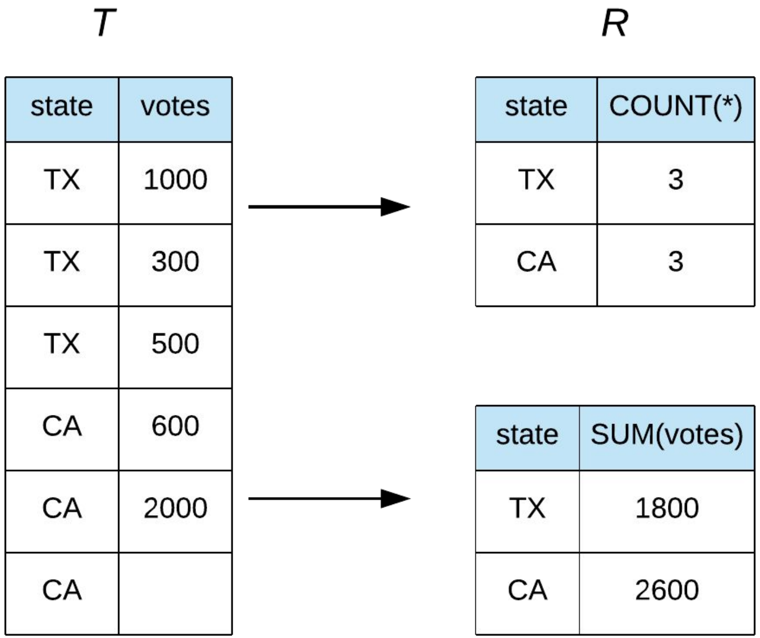
state
TX
CA

Aggregate Group By Queries

```
SELECT <unaggregated field(s)>,
       <aggregate function(s)>
FROM <single table>
[JOIN <single table>
 ON <join condition>]
[WHERE <boolean condition>]
GROUP BY <unaggregated field(s)>
[HAVING <boolean condition>]
[ORDER BY <field(s) to sort on>]
```

Aggregate Group By Queries

```
SELECT <unaggregated field(s)>,
       <aggregate function(s)>
FROM <single table>
[JOIN <single table>
 ON <join condition>]
[WHERE <boolean condition>]
GROUP BY <unaggregated field(s)>
[HAVING <boolean condition>]
[ORDER BY <field(s) to sort on>]
```



T

state	votes
TX	1000
TX	300
TX	500
CA	600
CA	2000
CA	

R

state	COUNT(*)
TX	3
CA	3

state	SUM(votes)
TX	1800
CA	2600

The semantics of COUNT()

```
SELECT COUNT(*)
FROM Employee

SELECT COUNT(department)
FROM Employee

SELECT DISTINCT department
FROM Employee

SELECT COUNT(DISTINCT department)
FROM Employee
```

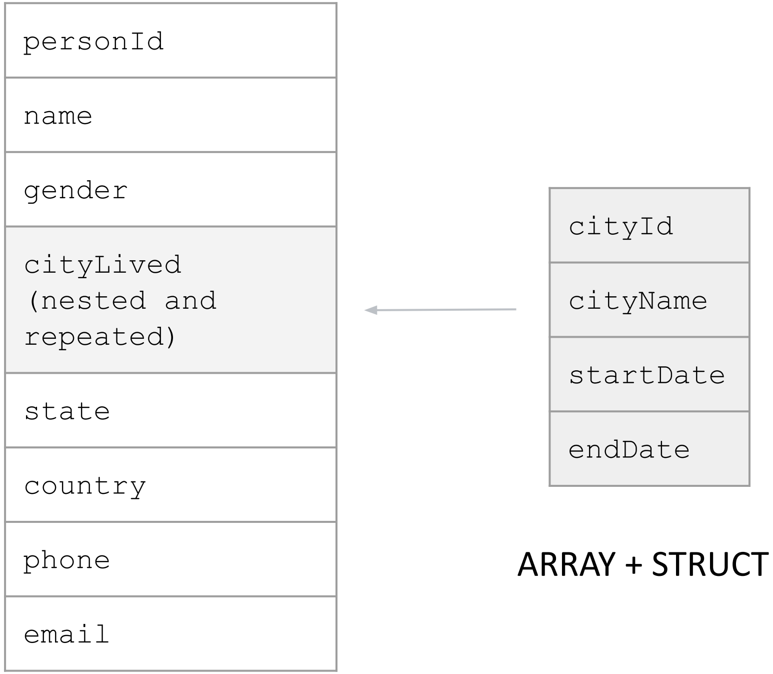
Employee

row	employee	department
1	Sunil	ENG
2	Morgan	ENG
3	Rama	Product
4	Drew	
5	Jeff	Research
6	Danielle	HR
7	Grace	ENG

BigQuery Overview

- Data warehouse / analytics database service
- Distributed database system
- Optimized for large data (petabyte-scale)
- Data model: tables with optional nesting
- Query language: standard SQL
- Data Types:
  - Primitive: BOOL, BYTES, FLOAT64, INT64, NUMERIC, STRING
  - Temporal: DATE, DATETIME, TIME, TIMESTAMP
  - Geospatial: GEOGRAPHY
  - Complex: ARRAY, STRUCT
- No provisioning, easy to use
- Not an operational database, no referential integrity

Nested Columns



personId

name

gender

cityLived (nested and repeated)

state

country

phone

email

cityId

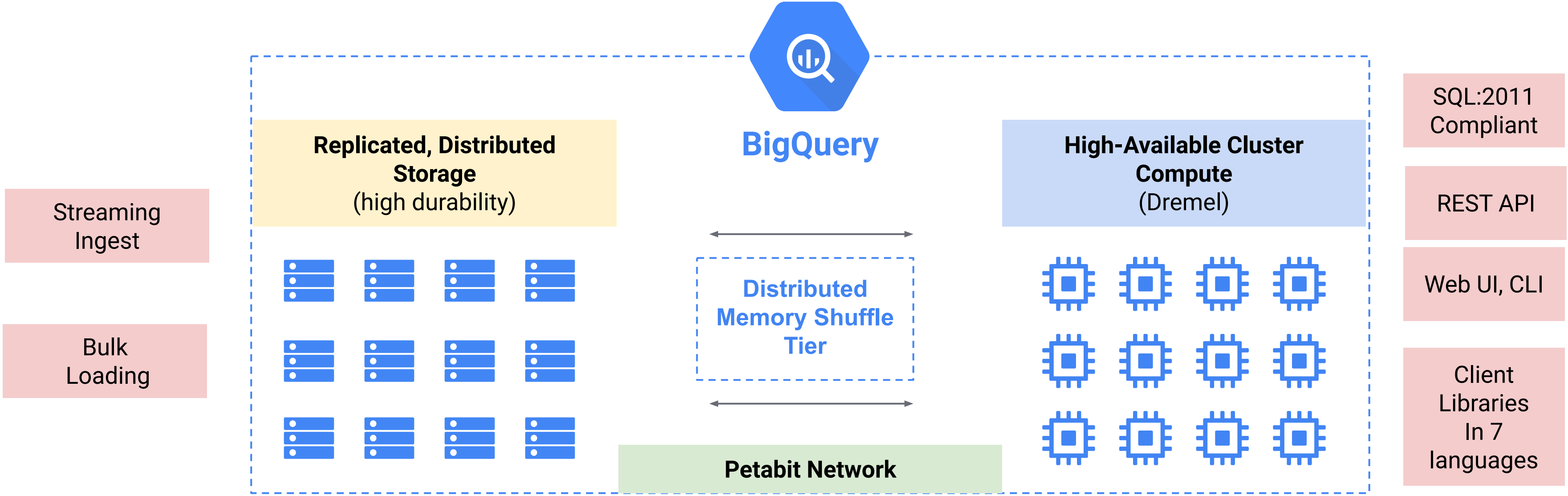
cityName

startDate

endDate

ARRAY + STRUCT type

BQ Architecture*



Streaming Ingest

Bulk Loading

Replicated, Distributed Storage
(high durability)

BigQuery

Distributed Memory Shuffle Tier

Petabit Network

High-Available Cluster Compute
(Dremel)

SQL:2011 Compliant

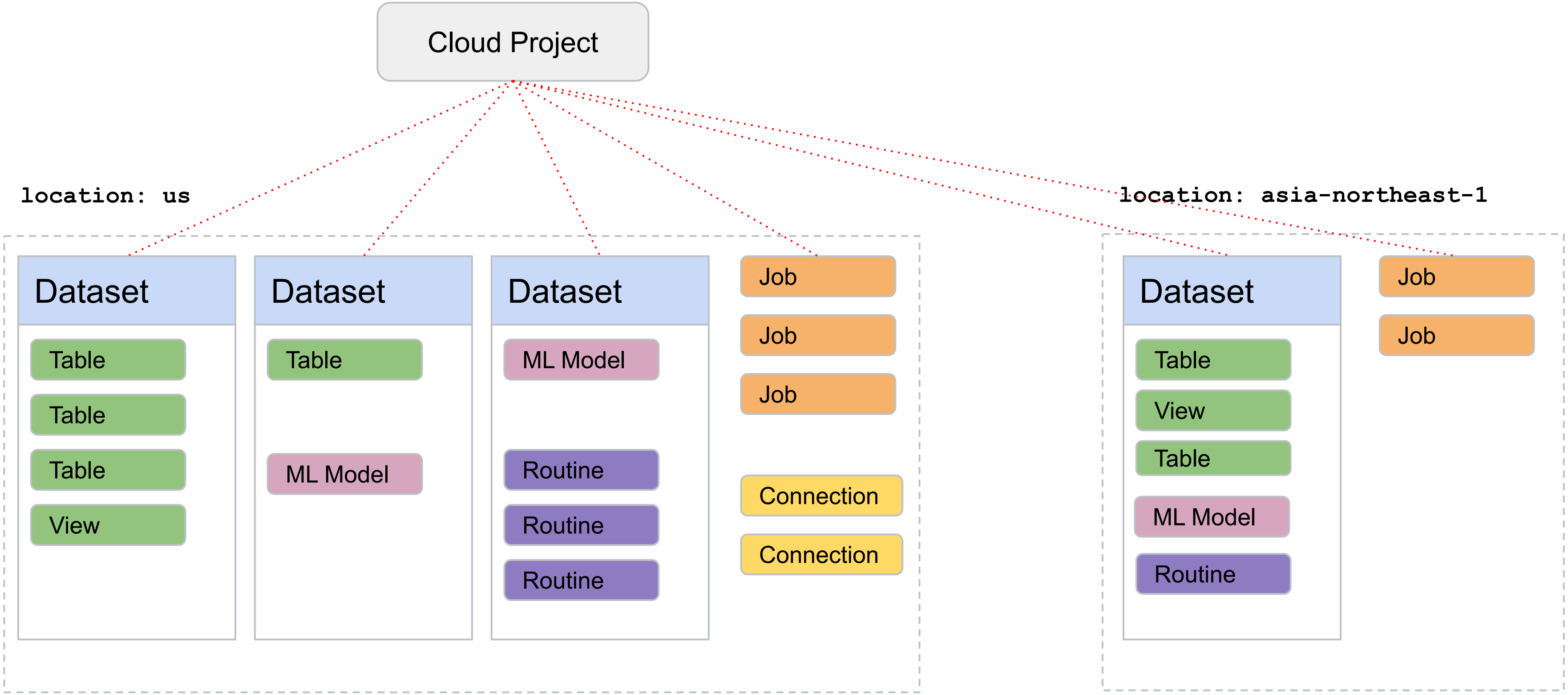
REST API

Web UI, CLI

Client Libraries In 7 languages

* Very approximate

Resource Model



BigQuery code lab

- Clone snippets repo
- Open bigquery notebook
- Create college dataset
- Populate college tables
- Explore the data
- Write aggregate queries

Practice Problems

1. For each class in the database, obtain the number of students taking the class. Return the cno of the class and its enrollment count.

2. For each class in the database which has at least two students enrolled, how many students are taking the class? Return the cno of the class and its enrollment count.

Database Schema:

Student(sid, fname, lname, dob,
status)

Class(cno, cname, credits)

Instructor(tid, fname, lname, dept)

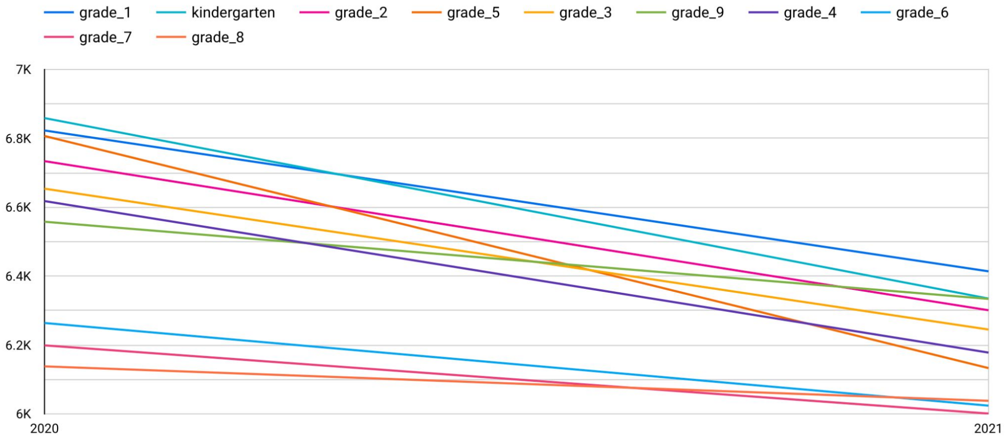
Takes(sid, cno, grade)

Teaches(tid, cno)

Project 4: school enrollments

https://www.cs.utexas.edu/~scohen/projects/Project4.pdf

- Open project4 notebook
- Create dataset
- Create and populate tables
- Explore the data
- Write sample aggregate query
- Create database view
- Create Data Studio report


grade_1
kindergarten
grade_2
grade_5
grade_3
grade_9
grade_4
grade_6
grade_7
grade_8
7K
6.8K
6.6K
6.4K
6.2K
6K