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>
```
Group By Queries

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

<table>
<thead>
<tr>
<th>state</th>
<th>votes</th>
</tr>
</thead>
<tbody>
<tr>
<td>TX</td>
<td>1000</td>
</tr>
<tr>
<td>TX</td>
<td>300</td>
</tr>
<tr>
<td>TX</td>
<td>500</td>
</tr>
<tr>
<td>CA</td>
<td>600</td>
</tr>
<tr>
<td>CA</td>
<td>2000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>state</th>
<th>COUNT(*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TX</td>
<td>3</td>
</tr>
<tr>
<td>CA</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>state</th>
<th>SUM(votes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TX</td>
<td>1800</td>
</tr>
<tr>
<td>CA</td>
<td>2600</td>
</tr>
</tbody>
</table>
The semantics of \texttt{COUNT()} \\

\texttt{SELECT COUNT(*)} \\
FROM Employee \\

\texttt{SELECT COUNT(department)} \\
FROM Employee \\

\texttt{SELECT DISTINCT department} \\
FROM Employee \\

\texttt{SELECT COUNT(DISTINCT department)} \\
FROM Employee \\

<table>
<thead>
<tr>
<th>row</th>
<th>employee</th>
<th>department</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sunil</td>
<td>ENG</td>
</tr>
<tr>
<td>2</td>
<td>Morgan</td>
<td>ENG</td>
</tr>
<tr>
<td>3</td>
<td>Rama</td>
<td>Product</td>
</tr>
<tr>
<td>4</td>
<td>Drew</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Jeff</td>
<td>Research</td>
</tr>
<tr>
<td>6</td>
<td>Danielle</td>
<td>HR</td>
</tr>
<tr>
<td>7</td>
<td>Grace</td>
<td>ENG</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>personId</th>
<th>name</th>
<th>gender</th>
<th>cityLived (nested and repeated)</th>
<th>state</th>
<th>country</th>
<th>phone</th>
<th>email</th>
</tr>
</thead>
</table>

- **ARRAY + STRUCT type**
  - cityId
  - cityName
  - startDate
  - endDate
BQ Architecture*

- Streaming Ingest
- Bulk Loading

**Petabit Network**

**Replicated, Distributed Storage** (high durability)

**BigQuery**

- **High-Available Cluster Compute** (Dremel)

  **Distributed Memory Shuffle Tier**

  **REST API**
  - Web UI, CLI

**Client Libraries**
- In 7 languages

* Very approximate
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


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