Not Only SQL

Rob Kellington

TRUST SCIENCE®

Caveat

- SQL and RDBMS Systems are Fantastic!
- Nearly **every** project I have worked on has been based on one of these databases
    - DB2
    - Oracle
    - SQL Server
    - SQLite
    - PostgreSQL

What is SQL?

- A “SQL” Database
  - Language - https://en.wikipedia.org/wiki/SQL:2016
  - Database Engine
  - Storage System
- An RDBMS combines these

Why SQL?

- SQL is a language specifically designed to deal with data
- Many product options - licences and open source
- Amazing integration with supporting tools & tech
- Resources easy to find
- Best choice for most data management solutions

Why not SQL?

- Data Variety - data types and complex structures
- Data Volume & Velocity
- Performance & Scalability
- Specific use cases
- Plus … alternatives continue to improve

Our Challenges

1. Complex API JSON Documents
2. Logs
3. Billing

Plus - we build using a serverless strategy

JSON Data

- Supporting Analytics and Machine Learning Pipeline
- Multiple documents - inputs, intermediate, results
- Relational model would require ***many*** tables
    - This is a challenge with complex data warehouses as well
- Dealing with schema changes & versioning
- Context and processing rules/flags

JSON Data

- Use AWS S3 as our canonical truth for data
- Different bucket/structure for different use cases
- Allows flexibility and operational benefits
- S3 provides versioning - no updates
- PostgreSQL for indexing each S3 file
    - uses JSON datatype in Postgresql for flexibility

JSON Data

{} JSON S3 Structure Example.json × {} JSON Schema Example.json AWS Cost & Usage.sql

Users > Rob > Desktop > {} JSON S3 Structure Example.json

```
<S3 Bucket>
    /<client>
        /a3a9e470-90c8-11e9-b5b7-f37c87eba0bd (requestId)

        /PipelineStep1
            /PipelineStep1/Source1/a3a9e470-90c8-11e9-b5b7-f37c87eba0bd.json
            /PipelineStep1/Source2/a3a9e470-90c8-11e9-b5b7-f37c87eba0bd.json
            /PipelineStep1/data/a3a9e470-90c8-11e9-b5b7-f37c87eba0bd-thisInfo.json
            /PipelineStep1/data/a3a9e470-90c8-11e9-b5b7-f37c87eba0bd-thatInfo.json
            /PipelineStep1/data/a3a9e470-90c8-11e9-b5b7-f37c87eba0bd-otherDataInfo.json
            /PipelineStep1/data/a3a9e470-90c8-11e9-b5b7-f37c87eba0bd-derivedInfo.json
            /PipelineStep1/data/a3a9e470-90c8-11e9-b5b7-f37c87eba0bd-PRIVATEInfo.json

        /PipelineStep2
            /PipelineStep2/a3a9e470-90c8-11e9-b5b7-f37c87eba0bd-centralityInfo.json

        /PipelineStep3
            /PipelineStep3/a3a9e470-90c8-11e9-b5b7-f37c87eba0bd-request.json
            /PipelineStep3/a3a9e470-90c8-11e9-b5b7-f37c87eba0bd-results.json

        /PipelineStep4
            /PipelineStep4/identity/fac5c5d219dc7112af565cddb3b273da9b757bbeb74f39c5b936091790e1
            /PipelineStep4/model/fac5c5d219dc7112af565cddb3b273da9b757bbeb74f39c5b936091790e1442
            /PipelineStep4/request/a3a9e470-90c8-11e9-b5b7-f37c87eba0bd.json
            /PipelineStep4/request/raw/a3a9e470-90c8-11e9-b5b7-f37c87eba0bd.json
            /PipelineStep4/result/a3a9e470-90c8-11e9-b5b7-f37c87eba0bd-results.json
            /PipelineStep4/result/mining-identity-tks-hash.json

        /PipelineStep5
            /PipelineStep5/cleanerRequest/a3a9e470-90c8-11e9-b5b7-f37c87eba0bd-model1a.json
            /PipelineStep5/cleanerRequest/a3a9e470-90c8-11e9-b5b7-f37c87eba0bd-model2.json
            /PipelineStep5/cleanerRequest/a3a9e470-90c8-11e9-b5b7-f37c87eba0bd-model5.json
            /PipelineStep5/cleanerRequest/a3a9e470-90c8-11e9-b5b7-f37c87eba0bd-model3.json
            /PipelineStep5/formatterRequest/a3a9e470-90c8-11e9-b5b7-f37c87eba0bd.json
            /PipelineStep5/predictorRequest/a3a9e470-90c8-11e9-b5b7-f37c87eba0bd-model1a.json
            /PipelineStep5/predictorRequest/a3a9e470-90c8-11e9-b5b7-f37c87eba0bd-model2.json
```

{} JSON S3 Structure Example.json {} JSON Schema Example.json × AWS Cost & Usage.sql

Users > Rob > Desktop > {} JSON Schema Example.json > {} dataContext > {} clientConfig > {} callbacks > abc web


```
{
    "schemaVersion": "2.0.0",
    "scoring": {
        "type": "Scoring",
        "package": "Regular",
        "useCase": "Lending-Payday",
        "jurisdiction": {
            "country": "CAN",
            "state": "ON"
        },
        "generateReport": true
    },
    "control": {
        "tags": [
            "mobile"
        ],
        "callbackHeader": "X-Api-Key: 9tXKB8aZoLiqWvdvasdfqr4gZAkMhw1JbhrvcP7f54nJ",
        "environment": "prod",
        "correlationId": 1571937202686
    },
    "source": "Portal",
    "user": "bill@client.com",
    "batchName": "single",
    "sendMobileInvite": false,
    "data": {
        "person": {
            "firstName": "mary",
            "lastName": "jones",
            "emails": [
                {
                    "primary": "some.name@hotmail.com"
                }
            ],
            "phones": [
                {
                    "home": ""
```

JSON Data

- Supporting Analytics
    - “Research Data Extractor”
        - Uses PostgreSQL for search
- Copies to new bucket individual files
- Tools can load/process as logical table
        - Python Pandas, Spark Dataframes, Presto SQL

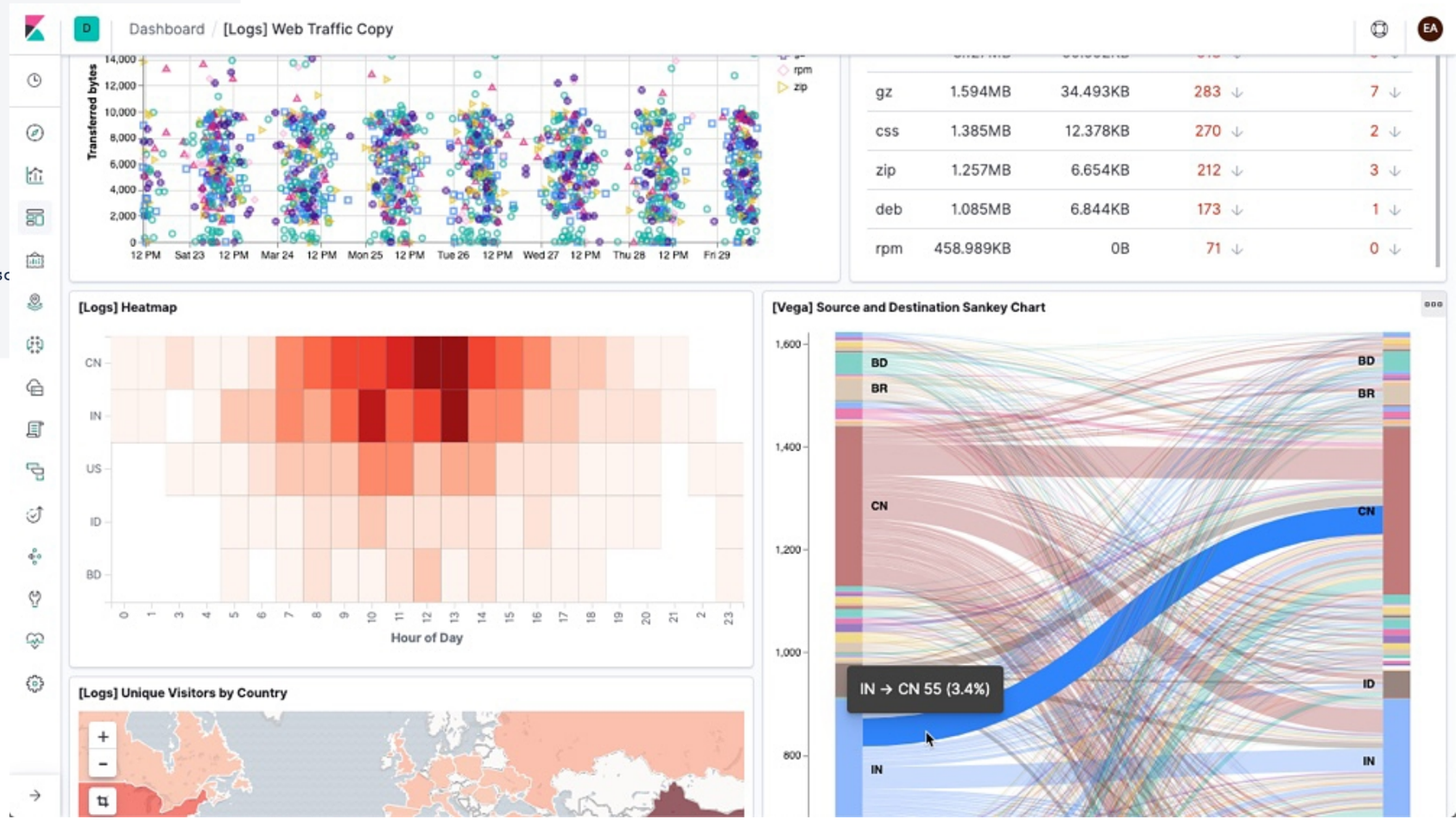
Log Data

- AWS - logs from all processes go to CloudWatch
    - Challenge to track and investigate
- Stream CloudWatch Logs Data to Elasticsearch
- NonSQL Query - but great for text search
- ElasticSearch & Kibana (variation of ELK stack)

Log Data

Fetch Items of a Given RequestId:

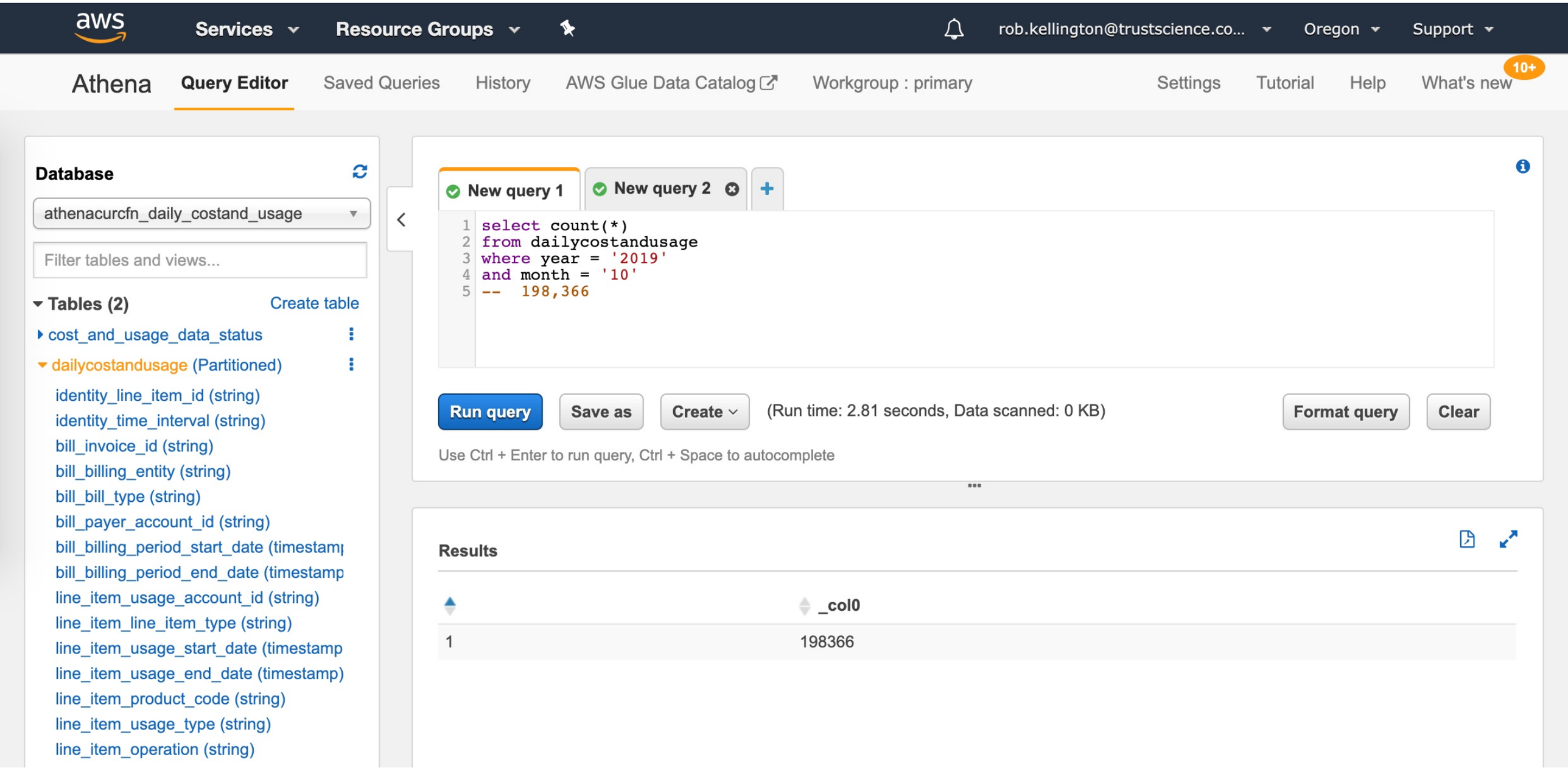
```
method: POST
endpoing: /<index>/_search

{
   "query":{
      "bool":{
         "must":{
            "bool":{
               "should":[
                  {
                     "match":{
                        "scoringRequestId":"<RequestId>"
                     }
                  }
               ]
            }
         }
      }
   },
  "sort": [
      {
          "timestamp" : {"order" : "asc"}
      } # In most cases, user prefer to read the history log in asc
  ],
}
```



Billing Data

- AWS AWS Cost & Usage Report
    - Very large files - 195 columns, 200k records per month
- Rarely accessed
    - Standard queries once a month and occasional inquiry
- Process into Parquet files
    - columnar, compressed files in S3
- Athena - managed Presto service
- Access - standard SQL

Billing Data - Athena



Other NoSQL

- Key Value
    - DynamoDB
- Document
    - Mongo
- Column
    - Google’s BigTable
- Graph
    - Neo4J
- Analytics Engines
    - Spark, Hadoop
- Blockchain
    - Hyperledger Fabric and Ethereum, Ledger Database

Summary

- SQL is great
    - But lots of alternatives exist that may work better for some of your use cases
- Cloud / Managed services make NoSQL options easier
    - S3 storage, RDS PostgreSQL, Athena, ElasticSearch/ Kibana
-