Migrating to PostgreSQL
Migrating to PostgreSQL

Migration of energy efficiency data processing from MS SQL Server to PostgreSQL.

Talk Roadmap

01 Data problem
Introduce myself, the CEDARS project, and the data requirements

02 Migration
Key motivations and roadblocks to migration, and how we are overcame them.

03 Lessons
What we’ve learned, and our next steps to complete the migration.
Migrating to PostgreSQL

We are a woman-owned small business that specializes in creating custom data-driven systems for clients in science and engineering.

We're based in Seattle, WA with team members around the globe.

We like nothing better than solving data problems for our clients by creating smart, flexible, real-time information systems that automate data processing and quality control.

**Our mission**

To define and build efficient long term solutions  
To never stop iterating and improving  
To provide superior consulting and data services  
To think deeply  
To empower our clients
Migrating to PostgreSQL
Migrating to PostgreSQL

Sound Data

The Hydrophone Data Repository (Hydra)
Funders: UW, NOAA, US-ACE, UC-Davis, Seattle City Light, and more
Team: Jennifer, Eric, Chris

Hydra has been online since 2008 and was Sound Data’s first online database project. Hydra provides data management and sharing for fisheries researchers who study the movement of aquatic animals using acoustic telemetry. Hydra has enjoyed tremendous success, growing from a grassroots regional solution to serve more than 200 researchers from across the west coast of North America.

From Hydra’s about page:

Around the Pacific Northwest, researchers from a variety of federal and local agencies, universities, and tribes in aggregate are using several hundred hydrophones to conduct research studies on movement patterns of aquatic animals. Each research program is characterized by numerous tagged animals that move and a relatively limited number of acoustic receivers that are located to address a significant question for individual programs. Importantly, these tagged animals move over larger domains than individual receiver arrays. These researchers have recognized the value of coordinating placement of hydrophones to improve their collective listening capability and ability to address emergent, larger scale management questions. Researchers needed the ability to efficiently share detections of each other’s tag codes to enable the larger research collaboration. Hydra was developed to facilitate data sharing and research coordination for these researchers. The researchers were happy with Hydra’s service so they told their colleagues who are also doing acoustic telemetry about Hydra and they joined too. Currently Hydra extends from British Columbia to Baja California.

California Energy Data and Reporting System (CEDARS)
Client: California Public Utility Commission (CPUC) Energy Division ESI
Team: Jennifer, Eric, Chris, Javier, Sky

Sound Data has handled every aspect of the CEDARS project since its inception, including developing and articulating client business processes, change management, system requirements, security requirements, platform specification, software development, testing, client management, and automation of data integration between disparate data systems. CEDARS goals are to automate manual data processing and quality control tasks that have been time consuming, expensive, and prone to error, to communicate transparent data specification and quality control rules, to be robust to change over time, and to integrate what had been disparate data sets to enable analysis and review; the CEDARS project has been highly successful, delivering multiple modules on-time despite tight deadlines, and the CPUC continues to expand the scope of CEDARS. CEDARS also accomplished the high-level goals of integrating with the CPUC Database for Energy Efficiency Resources (IDER) for purposes of claim and filing data validation, as well as integrating with the CPUC Cost Effectiveness Tool (CET) for purposes of calculating cost effectiveness of the portfolio.
Migrating to PostgreSQL

California Public Utility Commission (CPUC)

Our Mission

The California Public Utilities Commission serves the public interest by protecting consumers and ensuring the provision of safe, reliable utility service and infrastructure at just and reasonable rates, with a commitment to environmental enhancement and a healthy California economy. We regulate utility services, stimulate innovation, and promote competitive markets, where possible, in the communications, energy, transportation, and water industries.

http://www.cpuc.ca.gov/
Energy efficiency program administrators are required to submit data demonstrating their accomplishments that demonstrate they have a cost effective total portfolio.

If program administrators hit the savings goals set by the CPUC, they are awarded financial incentives.

The CPUC authorizes a $1B per year investment in energy efficiency in CA.
In the beginning, there was chaos.

Data stream flowing
In 2009, we had three years of data in hand and more data arriving 5x per year.

Come as you are
The first three years of data we held were requested as “give us your data” without specification.

Manual work
The team was manually processing and Qcing claim data. Data were split between different groups and each group developed their own manual processes.
Migrating to PostgreSQL

ACCESSIBILITY
SECURITY
CHANGE MGMT
INTEGRATION
DATA SPEC
AUTOMATION
PERFORMANCE
Migrating to PostgreSQL

DEER
PostgreSQL
Value lists

CEDARS
SQL Server → PostgreSQL
PA claim processing
Specification management
Quality control
Data access

CET
SQL Server
Calculation engine
Migrating to PostgreSQL
Migrating to PostgreSQL

How can we make energy efficiency data reporting more efficient?

**Automate**
Fully automate data processing and QC feedback. Give program administrators real-time data QC feedback.

**Integrate**
Build API integrations between CEDARS and external systems.

**Migrate**
Move data processing to an open source platform and eliminate costs while improving performance.
Migrating to PostgreSQL

Comparing RDBMS

Key factors

Database administration overhead
Development expense
Performance
Integration
Cost
Tools
What the CPUC stood to gain

Security
Move SFTP off database server

User Access
Eliminate SFTP users, add CPUC ED users, improve public user access

Automation
Data processing and cross-server automations

Transparency
Ensure consistency and transparency of data specifications

Performance
Data processing that took months is now done in minutes.

Reporting
Provide advanced on-demand data reporting across multiple data sets.
Migrating to PostgreSQL

(Statewide) Claim Dashboard

- Download PA CET data
- PA Confirmed Dashboard
  - Confirm Claim
  - PA Active Dashboard
    - SoT version control
    - Specification
      - Download SoT
      - Download value lists
- Upload Claim
  - QC
    - Event Tracking
    - Upload History
      - Upload Summary and QC feedback
      - Download QC Feedback

Download Goal Attainment Summary
Download Claim Summary
Download Program CE Summary
CET outputs
CET
Migrating to PostgreSQL

Claims value lists

Claims value lists as of 2019/03/04 (download all):

- Avoided Cost Combo
- Building HVAC
- Building Location
- Building Type
- Building Vintage
- Combustion Type
- Delivery Type
- E3 Climate Zone
- E3 Gas Saving Profile
- E3 Gas Sector
- EUL
- GSIA
- Labor Rate
- Location Cost Adjustment
- Measure Application Type
- Measure Impact Calculation Type
- Measure Impact Type
- Normal Unit
- NTG
- PA
- PrgID
- Rate Schedule for Electricity
- Rate Schedule for Gas
- Sector
- SourceDesc
- Technology Group
- Technology Type
- Use Category
- Use Subcategory
- Version

Claims source of truth files

Claims Source of Truth files as of 2018/11/07:

- claim_spec.sql: Tables and Single Field QC.
- claim_spec_PII.sql: Tables and QC for personally identifiable information.
- validation_rules.csv: Multi Field QC Rules.
- warning_rules.csv: Multi Field QC Warnings.
- Readme.txt: Syntax, high-level compliance, savings equations, cross-table validation.
- claim_metadata.csv: What the fields in claim_spec.sql represent.
- claim_good.zip: Known-good sample upload.
- Changelog.txt: What’s new in the Source of Truth files.

Claim_SourceOfTruth.zip: Download all current SoT files in one single zipfile.

You may also browse all previous versions.
Migrating to PostgreSQL

```sql
CREATE TABLE Site (
    SiteID nvarchar(255) NOT NULL PRIMARY KEY,
    SiteCity nvarchar(255) NOT NULL,
    SiteState nvarchar(255) NOT NULL,
    SiteZipCode nvarchar(255) NOT NULL,
    Residential_Flag bit DEFAULT False,
    NAICSCode nvarchar(255)
);
```

```sql
CREATE TABLE ProgramCost (
    ProgID nvarchar(255) NOT NULL PRIMARY KEY,
    CedarsValueList ProgID ForeignKey Program NotEmpty,
    PrgYear nvarchar(4) NOT NULL,
    ClaimYearQuarter CedarsYearQuarter NotEmpty,
    AdminCostsOverheadAndGA numeric NotEmpty,
    AdminCostsOther numeric NotEmpty,
    MarketingOutreach numeric NotEmpty,
    DIACTIVITY numeric NotEmpty,
    DIBase numeric NotEmpty,
    DILoan numeric NotEmpty,
    DILoanAndMaterials numeric NotEmpty,
    DIREbateAndInsurance numeric NotEmpty,
    EMV numeric NotEmpty,
    UserInputIncentive numeric NotEmpty,
    OnBillFinancing numeric NotEmpty,
    CostsRecoveredFromOtherSources numeric NotEmpty,
    PA nvarchar(255) CedarsValueList PA NotEmpty,
    primary key (ProgID, PrgYear)
);
```

```sql
CREATE TABLE dbo.CustomMeasure (
    ClaimID nvarchar(255) NOT NULL PRIMARY KEY,
    MeasCode nvarchar(255),
    MeasAppType nvarchar(255) CedarsValueList MeasAppType NotEmpty,
    MeasDescription nvarchar(255) NotEmpty,
    UseCategory nvarchar(255) CedarsValueList UseCategory NotEmpty,
    UseSubCategory nvarchar(255) CedarsValueList UseSubCategory NotEmpty,
    TechGroup nvarchar(255) CedarsValueList TechGroup NotEmpty,
    TechType nvarchar(255) CedarsValueList TechType NotEmpty,
    UnitW1stBaseline numeric NotEmpty,
    UnitW1stBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
    UnitW2ndBaseline numeric NotEmpty,
<table>
<thead>
<tr>
<th>Table</th>
<th>ProgramCost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field</td>
<td>PrgYear</td>
</tr>
<tr>
<td>Rule</td>
<td>matches_year_quarter</td>
</tr>
<tr>
<td>Value</td>
<td>ClaimYearQuarter</td>
</tr>
</tbody>
</table>
Migrating to PostgreSQL

<table>
<thead>
<tr>
<th>Table</th>
<th>Claim</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field</td>
<td>E3GasSavProfile</td>
</tr>
<tr>
<td>Rule</td>
<td>is not null</td>
</tr>
<tr>
<td>Where Connector</td>
<td>all</td>
</tr>
<tr>
<td>Where Table 1</td>
<td>coalesce (DeemedMeasure, CustomMeasure)</td>
</tr>
<tr>
<td>Where Field 1</td>
<td>UnitTherm1stBaseline</td>
</tr>
<tr>
<td>Where Rule 1</td>
<td>does not equal</td>
</tr>
<tr>
<td>Where Value 1</td>
<td>0</td>
</tr>
<tr>
<td>Where Table 2</td>
<td>Claim</td>
</tr>
<tr>
<td>Where Field 2</td>
<td>DeliveryType</td>
</tr>
<tr>
<td>Where Rule 2</td>
<td>does not equal</td>
</tr>
<tr>
<td>Where Value 2</td>
<td>'C&amp;S'</td>
</tr>
</tbody>
</table>
Migrating to PostgreSQL
Migrating to PostgreSQL

pgloader loads data into PostgreSQL and allows you to implement Continuous Migration from your current database to PostgreSQL. Read the White Paper to learn how to limit risks and control your budget, and start your PostgreSQL migration today!

Introduction

pgLoader has two modes of operation. It can either load data from files, such as CSV or Fixed-File Format; or migrate a whole database to PostgreSQL.

pgLoader supports several RDBMS solutions as a migration source, and fetches information from the catalog tables over a connection to then create an equivalent schema in PostgreSQL. This means that you can migrate to PostgreSQL in a single command-line!

Supported operations include:

- Migrate from MySQL to PostgreSQL
- Migrate from SQLite to PostgreSQL
- Migrate from MS SQL Server® to PostgreSQL

You can also migrate database files in the DBF and IXF formats, where pgLoader can inspect the target table format for you automatically in the file headers.

https://pgloader.io/
https://github.com/dimitri/pgloader

$ apt-get install pgloader
Manual processing of the data stream needs to continue while new tools are built.

Agree to build new tools to solve data latency issues and improve data quality.

The new tools take over data processing. Tasks that took months now take minutes.
CPUC is consolidating server resources into their new AWS Gov cloud. CEDARS was the first project they brought into AWS Gov. Our three CEDARS database servers cost less than our one MSSS server; all four servers are the same instance type.

We run three PostgreSQL servers on Ubuntu for less cost than our one Windows server.

CPUC is consolidating server resources into their new AWS Gov cloud. CEDARS was the first project they brought into AWS Gov. Our three CEDARS database servers cost less than our one MSSS server; all four servers are the same instance type.
Migrating to PostgreSQL

**DEER**
- Value lists

**CEDARS**
- PA claim processing
- Specification management
- Quality control
- Data access

**CET**
- Calculation engine
APIs to external systems enable real-time synchronization. Manual reconciliation was eliminated.

External integrations
APIs to external systems enable real-time synchronization. Manual reconciliation was eliminated.

CEDARS improves communication and collaboration between program administrator and CPUC staff.

Transparency
Data specifications are version controlled. Updates are made per the change management protocol.

Improved relationship
CEDARS improves communication and collaboration between program administrator and CPUC staff.

Low latency
Data processing is fully automated and no manual work is required. Processing happens in minutes.

Reduced cost
Lower server costs, elimination of manual tasks, and elimination of software costs.

Better access
Public data access becomes real-time and is expanded. CPUC staff can directly access data.

External integrations
APIs to external systems enable real-time synchronization. Manual reconciliation was eliminated.

Transparency
Data specifications are version controlled. Updates are made per the change management protocol.

Low latency
Data processing is fully automated and no manual work is required. Processing happens in minutes.

Reduced cost
Lower server costs, elimination of manual tasks, and elimination of software costs.

Better access
Public data access becomes real-time and is expanded. CPUC staff can directly access data.

External integrations
APIs to external systems enable real-time synchronization. Manual reconciliation was eliminated.
Migrating to PostgreSQL

PostgreSQL for the win

- Lighter database administration overhead
- Lighter development expense
- Better performance
- Easier integrations
- Much lower cost
- Powerful tools

And the migration to PostgreSQL was painless.