# Proposed Solution: TrackMan API to PostgreSQL ETL Pipeline

Z Skigen

October 5, 2025

## Motivation and Background

The Pomona-Pitzer baseball program currently relies on manual downloads of large TrackMan CSV files. Each file must be cleaned, merged, and analyzed separately, which makes the process slow and difficult to reproduce. Analysts often spend more time manipulating spreadsheets than actually studying player performance. TrackMan's proprietary software provides summary reports, but not flexible access to the underlying pitch-level data that analysts need to evaluate players or design training programs. This project aims to solve that by creating a robust and automated data pipeline that extracts, cleans, and stores TrackMan data in a reproducible and queryable form.

## Pipeline Overview

The TrackMan API provides access to game, pitch, and player data in JSON format. However, this raw output is extremely messy. Different API endpoints (for example, `/balls` and `/plays`) return data with overlapping but inconsistent keys. Game sessions are often fragmented by date or session ID, and missing or duplicated entries are common. Even within a single JSON

file, the nesting is irregular: certain fields such as spin rate or pitch movement may appear under multiple sub-dictionaries depending on the session type, and timestamps are not always synchronized across objects.

Because of this irregularity, an automated extraction and cleaning process is essential. The pipeline first authenticates with the TrackMan server using client credentials to obtain an access token, then retrieves all available game sessions in manageable 30-day intervals to avoid API rate limits. For each session, it queries the corresponding pitch- and play-level data, cleans field names, removes missing or null values, and standardizes timestamps into a consistent UTC format. The cleaned data is then loaded into a structured PostgreSQL database, organized by tables for games, players, pitches, and outcomes. This modular ETL (Extract, Transform, Load) design turns an unstructured, fragmented data stream into a stable dataset suitable for analysis.

# Design Rationale

PostgreSQL was chosen as the storage solution because it is open-source, scalable, and well-suited for structured, relational data. Unlike spreadsheets, a relational database enforces consistent typing, indexing, and relationships between entities such as games and players. This schema design mirrors the real-world structure of baseball: each pitch belongs to a plate appearance, each appearance to a game, and each game to a team and date. By enforcing this relational logic, the database ensures that future analyses—like computing average pitch velocities, locating trends across seasons, or generating scouting summaries—can be done with a single reproducible query instead of multiple spreadsheet merges.

# Automation

To ensure the database remains current, the ETL pipeline will run automatically on a fixed schedule using a local or cloud-based job scheduler. Each

run will authenticate, check for new sessions, download only new data, and append it to the existing database. By integrating automated error handling and logging, the system will be resilient to API downtime or partial data failures. Over time, this automation will make TrackMan data ingestion fully hands-free, so analysts and coaches can focus on higher-level models rather than file maintenance.

## Impact and Future Work

Automating this workflow will dramatically reduce the time spent on data cleaning and increase the reproducibility of team analyses. Coaches will be able to access live, organized data for scouting and player development, while analysts can focus on evaluating performance rather than troubleshooting inconsistencies. In the future, the pipeline could be extended to integrate batted-ball video clips, link to player tracking data, or scale to multiple teams in the SCIAC conference. Ultimately, the project will provide Pomona-Pitzer Baseball with a sustainable and extensible foundation for all future analytics work.