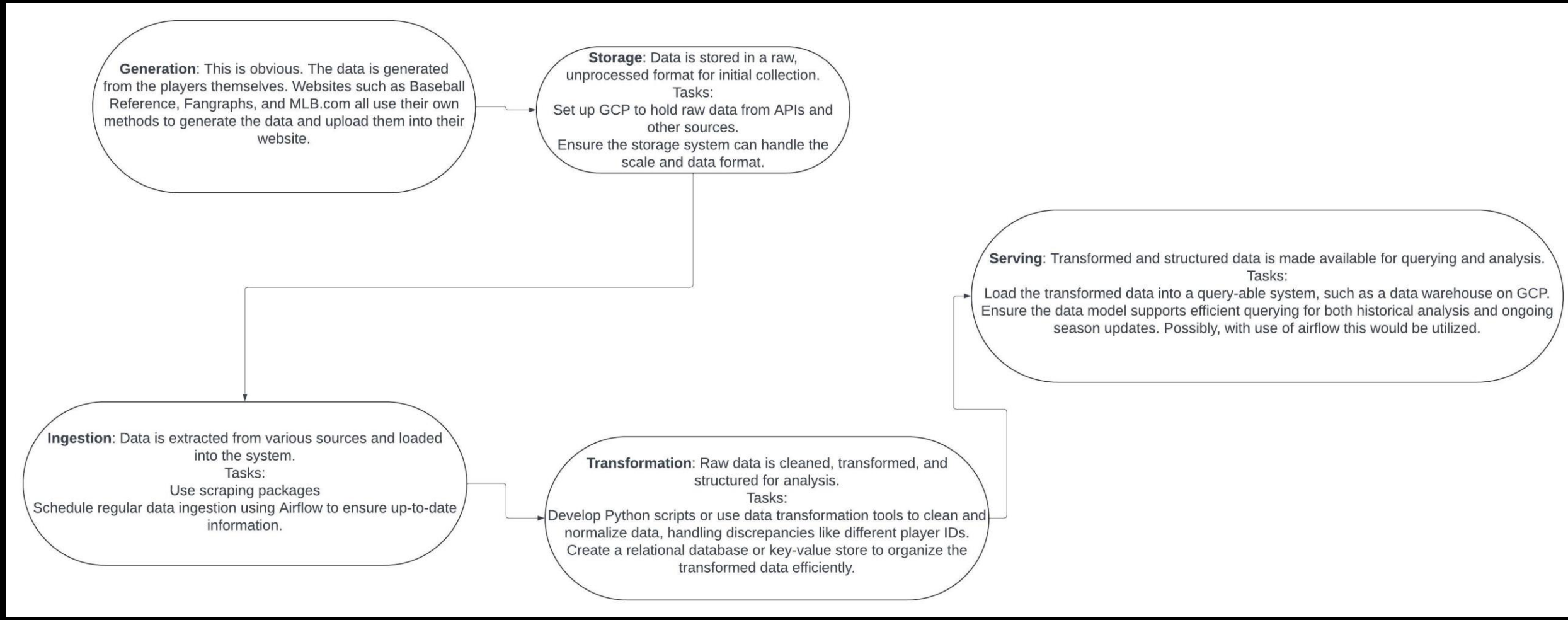
BASEBALL STATS PIPELINE PROJECT
BY SCOTT SILVERSTEIN

Why a Baseball Stats Pipeline

Because it’s the reason I am here!

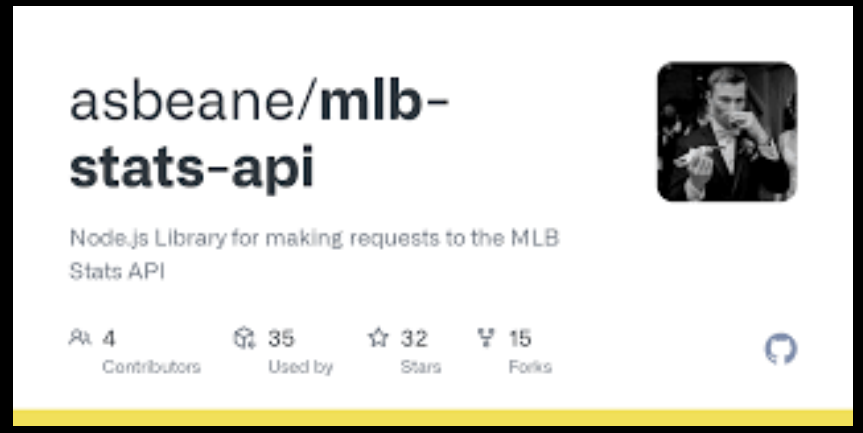
What I did?

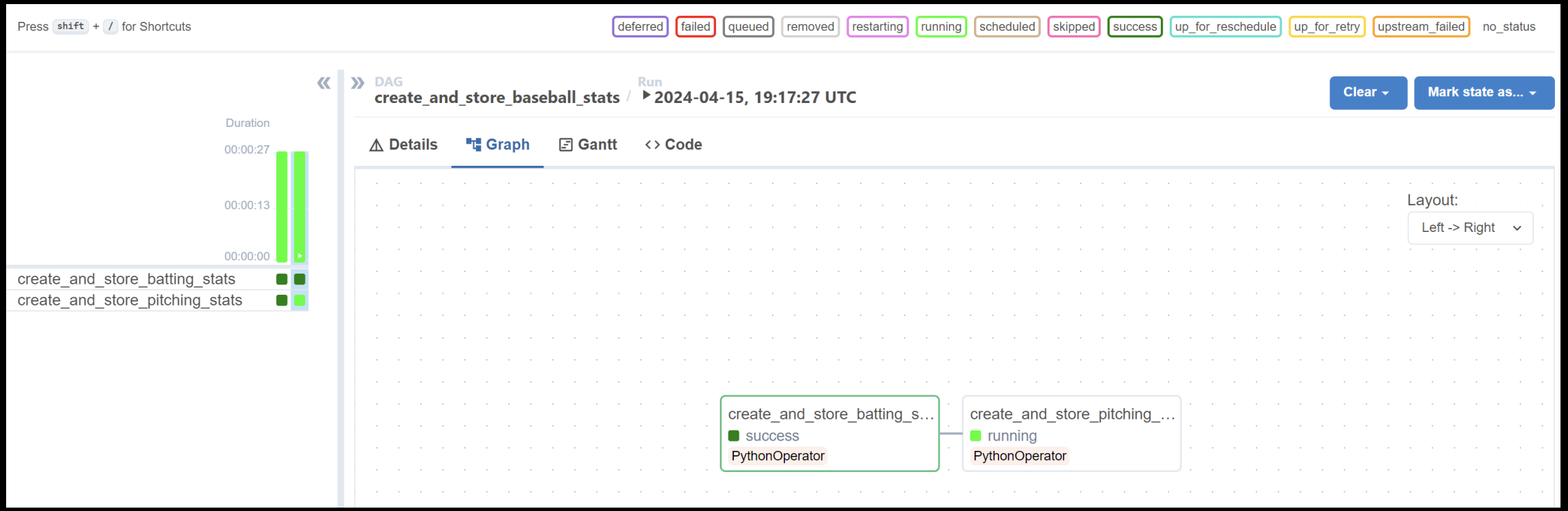

Generation: This is obvious. The data is generated from the players themselves. Websites such as Baseball Reference, Fangraphs, and MLB.com all use their own methods to generate the data and upload them into their website.
Storage: Data is stored in a raw, unprocessed format for initial collection.
Tasks:
Set up GCP to hold raw data from APIs and other sources.
Ensure the storage system can handle the scale and data format.
Serving: Transformed and structured data is made available for querying and analysis.
Tasks:
Load the transformed data into a query-able system, such as a data warehouse on GCP.
Ensure the data model supports efficient querying for both historical analysis and ongoing season updates. Possibly, with use of airflow this would be utilized.
Ingestion: Data is extracted from various sources and loaded into the system.
Tasks:
Use scraping packages
Schedule regular data ingestion using Airflow to ensure up-to-date information.
Transformation: Raw data is cleaned, transformed, and structured for analysis.
Tasks:
Develop Python scripts or use data transformation tools to clean and normalize data, handling discrepancies like different player IDs.
Create a relational database or key-value store to organize the transformed data efficiently.

Generation

Pybaseball Package

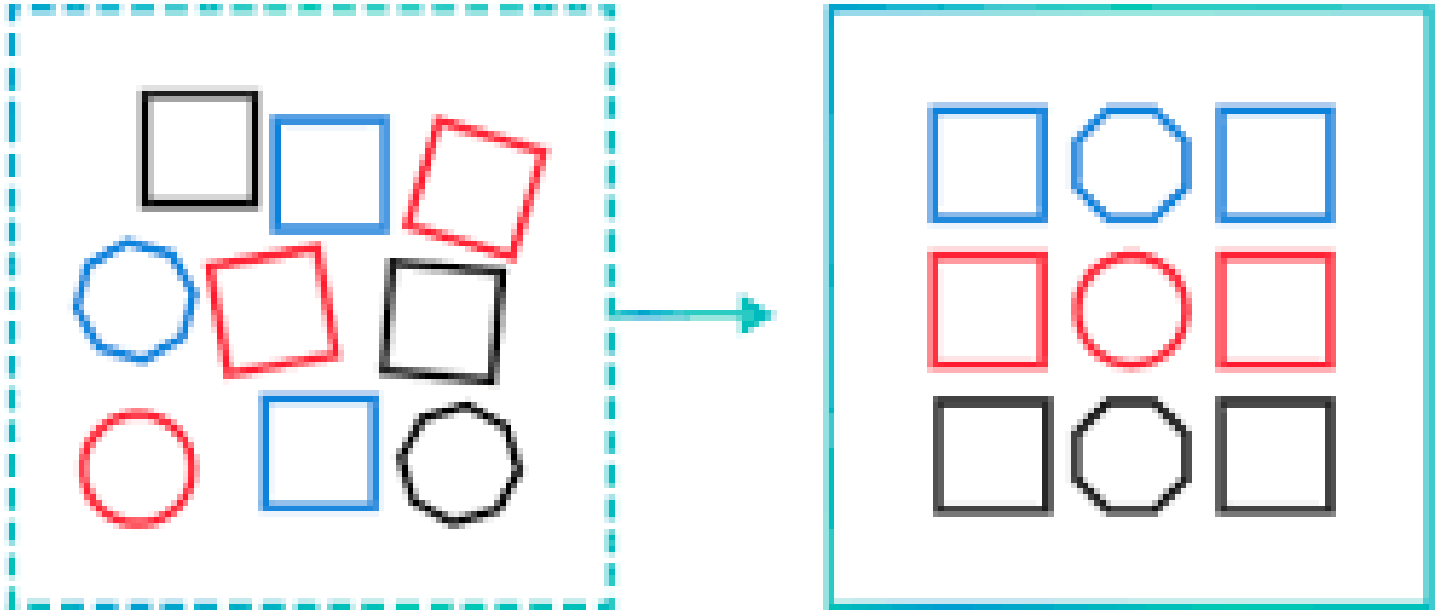
- I used Pybaseball package to scrape data
- Many options
  - Baseball_scraper
  - Mlb Stats API
- MLB API is both great and terrible





Ingestion

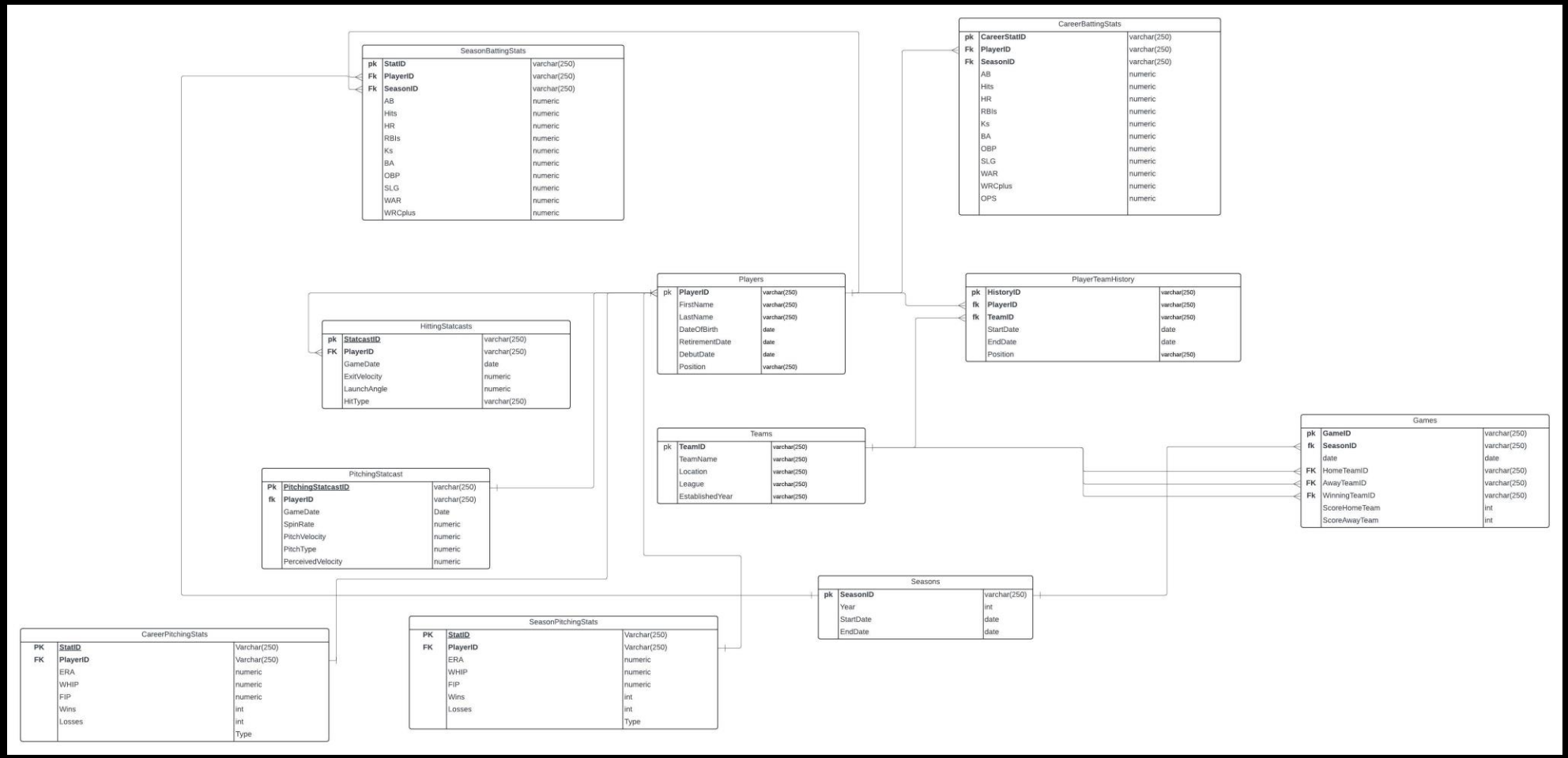
- This was done by creating tables and using functions to fetch data from pybaseball scraper
- Eventually took these functions and put them into an airflow for easier future automation



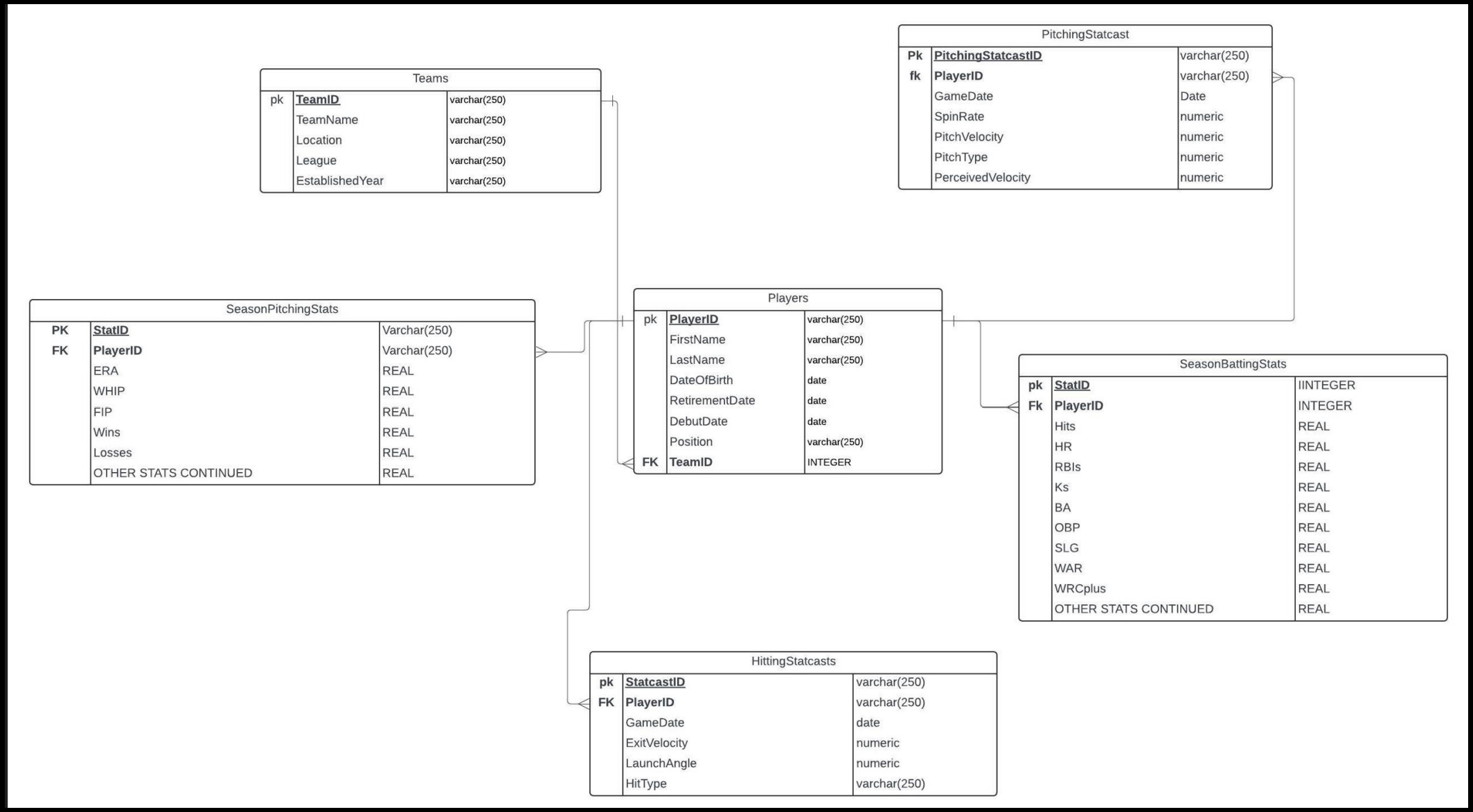
Transformation

- This was done by converting all the datasets generated in pybaseball and formatting them to fit my logical model.

```
Number of columns in SeasonBattingStats: 317
```

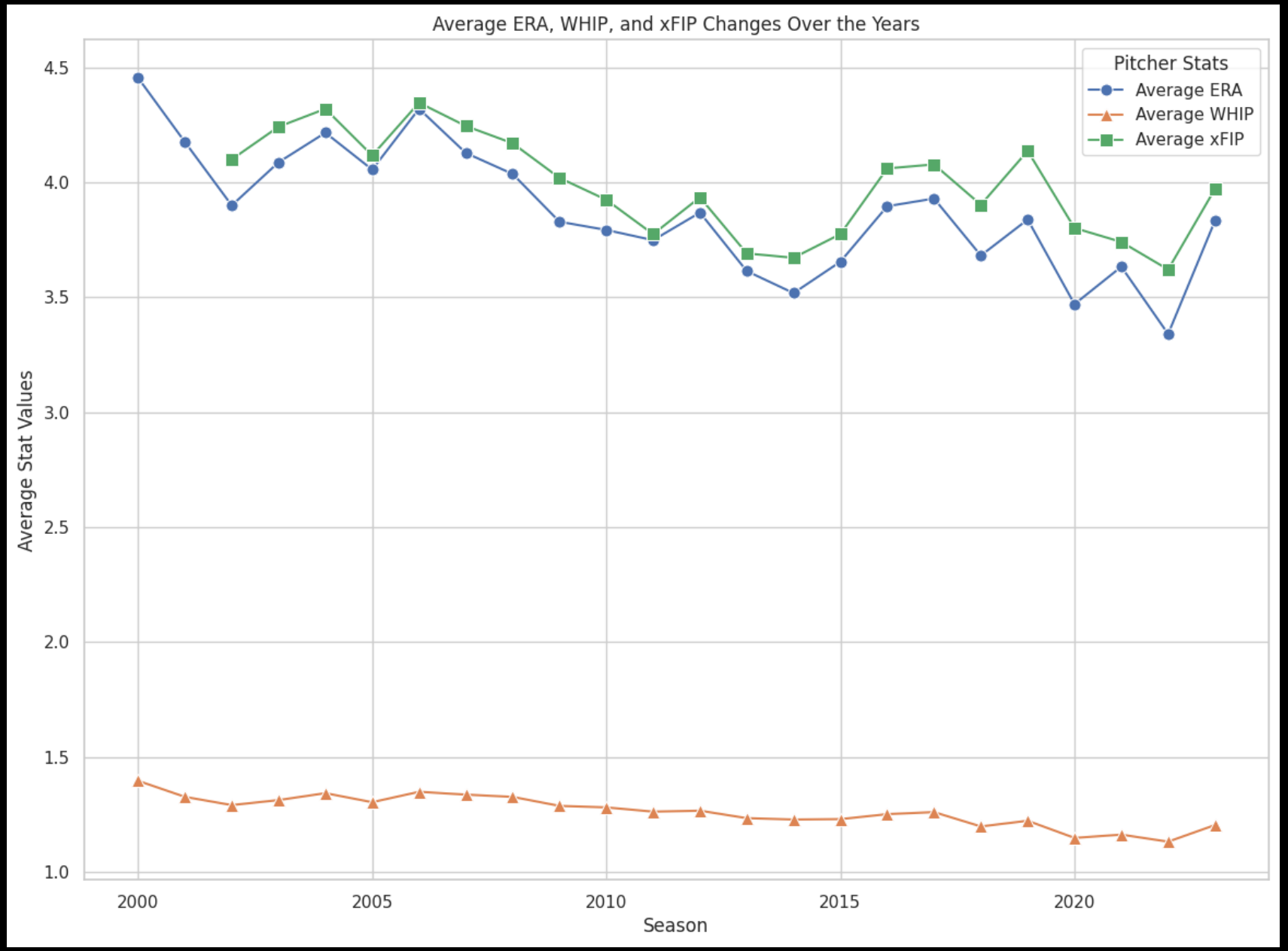

SeasonBattingStats
pk StatID varchar(250)
Fk PlayerID varchar(250)
Fk SeasonID varchar(250)
AB numeric
Hits numeric
HR numeric
RBIs numeric
Ks numeric
BA numeric
OBP numeric
SLG numeric
WAR numeric
WRCplus numeric
CareerBattingStats
pk CareerStatID varchar(250)
Fk PlayerID varchar(250)
Fk SeasonID varchar(250)
AB numeric
Hits numeric
HR numeric
RBIs numeric
Ks numeric
BA numeric
OBP numeric
SLG numeric
WAR numeric
WRCplus numeric
OPS numeric
Players
pk PlayerID varchar(250)
FirstName varchar(250)
LastName varchar(250)
DateOfBirth date
RetirementDate date
DebutDate date
Position varchar(250)
PlayerTeamHistory
pk HistoryID varchar(250)
fk PlayerID varchar(250)
fk TeamID varchar(250)
StartDate date
EndDate date
Position varchar(250)
HittingStatcasts
pk StatcastID varchar(250)
FK PlayerID varchar(250)
GameDate date
ExitVelocity numeric
LaunchAngle numeric
HitType varchar(250)
Teams
pk TeamID varchar(250)
TeamName varchar(250)
Location varchar(250)
League varchar(250)
EstablishedYear varchar(250)
Games
pk GameID varchar(250)
fk SeasonID varchar(250)
date date
FK HomeTeamID varchar(250)
FK AwayTeamID varchar(250)
Fk WinningTeamID varchar(250)
ScoreHomeTeam int
ScoreAwayTeam int
PitchingStatcast
Pk PitchingStatcastID varchar(250)
fk PlayerID varchar(250)
GameDate Date
SpinRate numeric
PitchVelocity numeric
PitchType numeric
PerceivedVelocity numeric
Seasons
pk SeasonID varchar(250)
Year int
StartDate date
EndDate date
CareerPitchingStats
PK StatID Varchar(250)
FK PlayerID Varchar(250)
ERA numeric
WHIP numeric
FIP numeric
Wins int
Losses int
Type
SeasonPitchingStats
PK StatID Varchar(250)
FK PlayerID Varchar(250)
ERA numeric
WHIP numeric
FIP numeric
Wins int
Losses int
Type

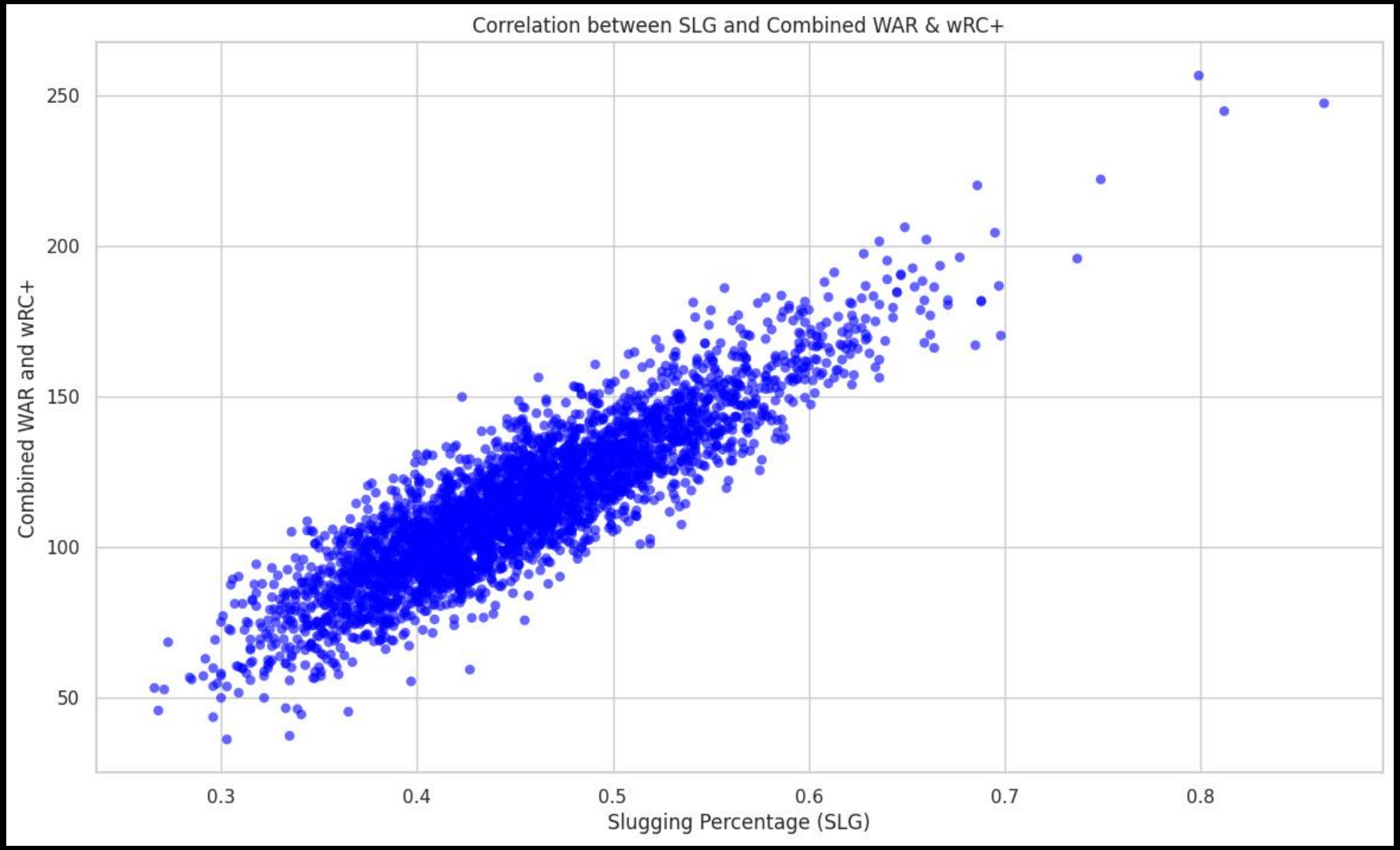
Challenges


Teams
pk TeamID varchar(250)
TeamName varchar(250)
Location varchar(250)
League varchar(250)
EstablishedYear varchar(250)
PitchingStatcast
Pk PitchingStatcastID varchar(250)
fk PlayerID varchar(250)
GameDate Date
SpinRate numeric
PitchVelocity numeric
PitchType numeric
PerceivedVelocity numeric
SeasonPitchingStats
PK StatID Varchar(250)
FK PlayerID Varchar(250)
ERA REAL
WHIP REAL
FIP REAL
Wins REAL
Losses REAL
OTHER STATS CONTINUED REAL
Players
pk PlayerID varchar(250)
FirstName varchar(250)
LastName varchar(250)
DateOfBirth date
RetirementDate date
DebutDate date
Position varchar(250)
FK TeamID INTEGER
SeasonBattingStats
pk StatID IINTEGER
Fk PlayerID INTEGER
Hits REAL
HR REAL
RBIs REAL
Ks REAL
BA REAL
OBP REAL
SLG REAL
WAR REAL
WRCplus REAL
OTHER STATS CONTINUED REAL
HittingStatcasts
pk StatcastID varchar(250)
FK PlayerID varchar(250)
GameDate date
ExitVelocity numeric
LaunchAngle numeric
HitType varchar(250)

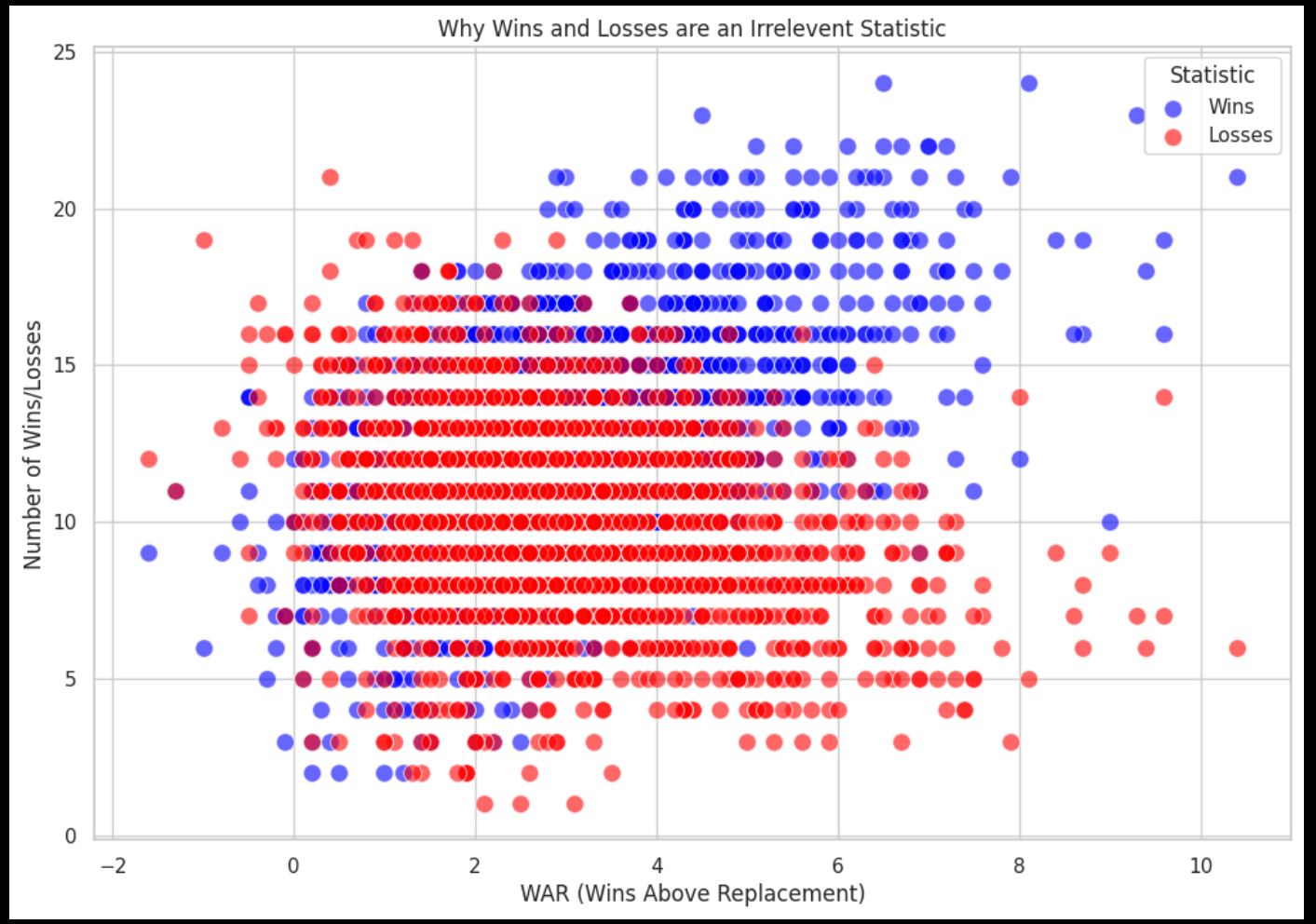
Final ERD

Serving

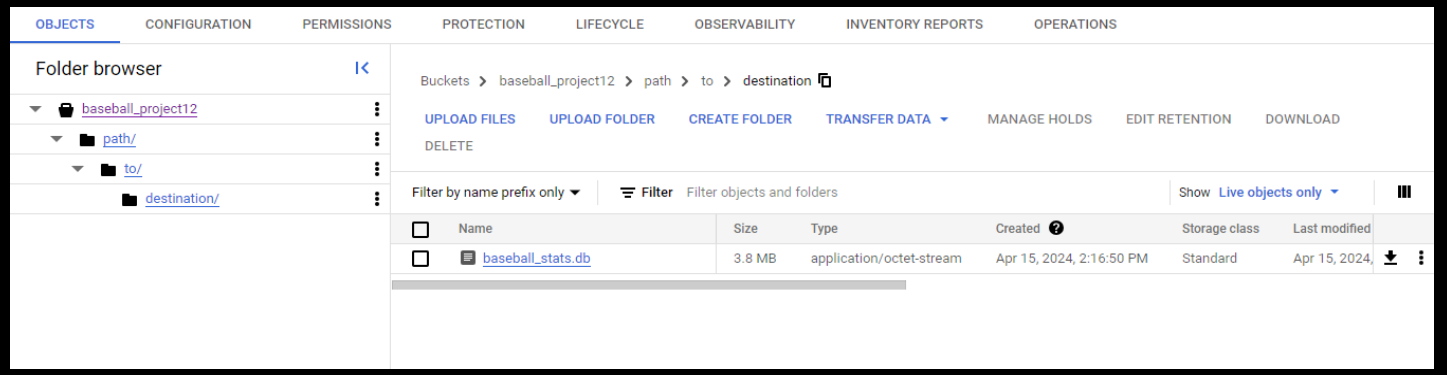

Average ERA, WHIP, and xFIP Changes Over the Years
Pitcher Stats
Average ERA
Average WHIP
Average xFIP
Average Stat Values
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
2000
2005
2010
2015
2020
Season


Correlation between SLG and Combined WAR & wRC+
Combined WAR and wRC+
250
200
150
100
50
0.3
0.4
0.5
0.6
0.7
0.8
Slugging Percentage (SLG)

	Parameter	Wins	Losses
0	Coefficient	1.1576772471390062	-0.5858824832461207
1	P-value	3.2309953917954 63e-131	2.709156638605293e-41
2	R-squared	0.2814402070241847	0.09587028552839727
3	Adj. R-squared	0.28104056309038283	0.09536743251701152



Storage


OBJECTS
CONFIGURATION
PERMISSIONS
PROTECTION
LIFECYCLE
OBSERVABILITY
INVENTORY REPORTS
OPERATIONS
Folder browser
baseball_project12
path/
to/
destination/
Buckets > baseball_project12 > path > to > destination
UPLOAD FILES
UPLOAD FOLDER
CREATE FOLDER
TRANSFER DATA
MANAGE HOLDS
EDIT RETENTION
DOWNLOAD
DELETE
Filter by name prefix only
Filter
Filter objects and folders
Show Live objects only
Name
Size
Type
Created
Storage class
Last modified
baseball_stats.db
3.8 MB
application/octet-stream
Apr 15, 2024, 2:16:50 PM
Standard
Apr 15, 2024,


Google Cloud

Future Plans

- I now have a working database and an airflow that can easily automatically update my data!
- Try to figure out issues with different ID’s for different websites
- Make a more cohesive ERD with information on player contracts, player history, etc.