SKILLSTORM

By Christopher Lee, Stanley Liu, and Javier Martinez

P2: GitHub Archive ETL Pipeline

SKILLSTORM

Agenda

01 | Separation of Work

02 | Data Overview

03 | Data Quality Standards

04 | Architecture

05 | Tools & Platforms

06 | Entity Relationship Diagrams (Silver & Gold)

07 | Bronze to Silver Pipeline

08 | Silver to Gold Pipeline

09 | Visualizations

10 | Conclusion

Separation of Work

Each layer of the pipeline was separated between the group members.
This agreement was reached upon because of the following:

- ★ Reduce stress
- ★ Encourage Discussion
- ★ Code Review
- ★ Everyone understands everything
- ★ Faster completion of each phase

Data Overview

GH Archive Data

- 73 GB Gzipped JSON files
- Data is messy

Pipeline

- Bronze to Silver
- Silver to Gold

Ready for analytics and creating visuals

Data Quality Standards

To ensure our data is of the highest quality, the following quality standards will be enforced:

- All columns containing URLs, Hrefs, .sha, numbered columns (ex. labels.0 , labels.1) will not be considered, as the information they contain provides no analytical value.
- Any ID columns must have only one value associated with that id. For example, the actor.id column can contain multiple actor.logins. Tackling this issue is a must.
- Columns with redundant data, although they may have analytical value, will be dropped. (ex. The ‘public’ column, all values in this column are true)
- Columns that have very little analytical data (mostly nulls) will be dropped. As those columns are not an accurate representation of the data.

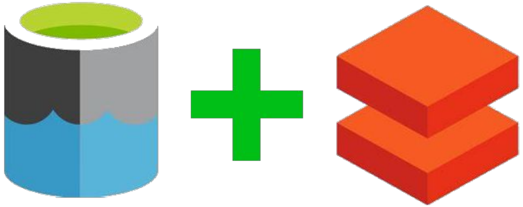
Tools & Platform

ADLS Gen2

- Cloud Storage System
- Stores all data (raw bronze, clean silver, analytics gold)

Databricks (Spark)

- Processing large volumes of data
- Ease of collaborating with others
- Ease of creating tables and visualizations



SKILLSTORM

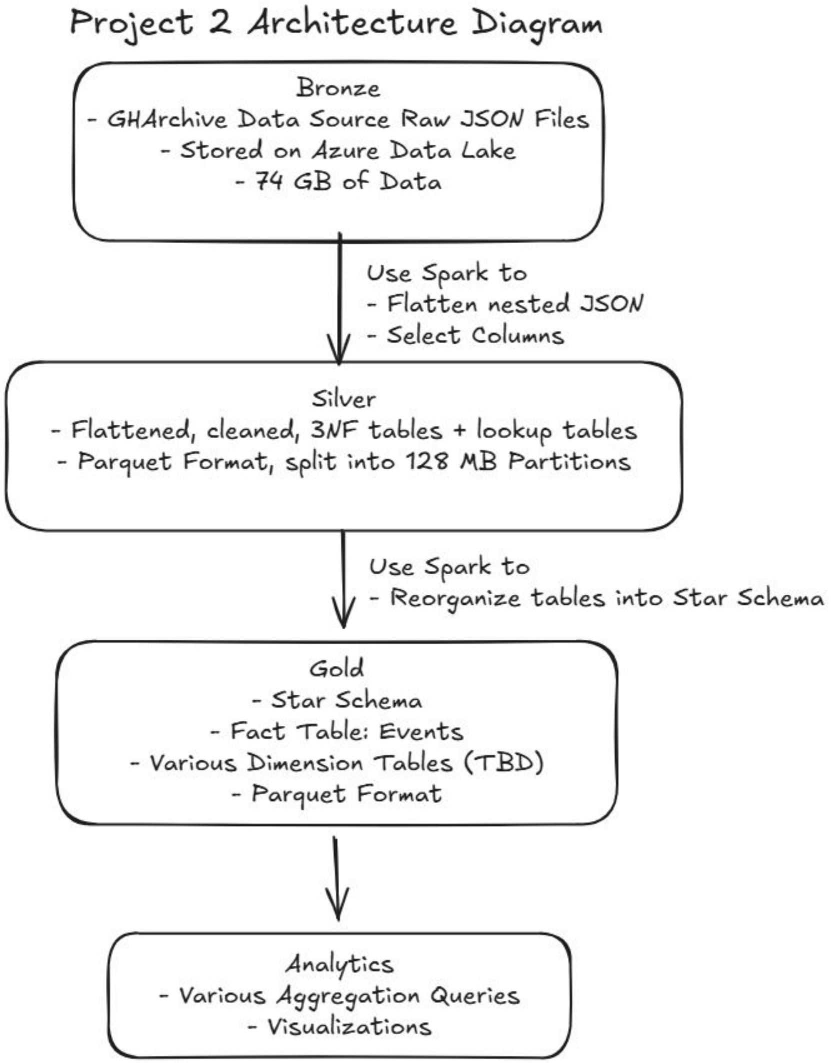
Architecture Diagram

Store the raw GH Archive Data into Bronze folder in ADLS

Run a pipeline that will clean and normalize Bronze into cleaned, Silver data.

Run a second pipeline that will reorganize our silver data into gold data, analytics ready.

Create visuals, prove data has worth.



Project 2 Architecture Diagram

Bronze
- GHArchive Data Source Raw JSON Files
- Stored on Azure Data Lake
- 74 GB of Data

Use Spark to
- Flatten nested JSON
- Select Columns

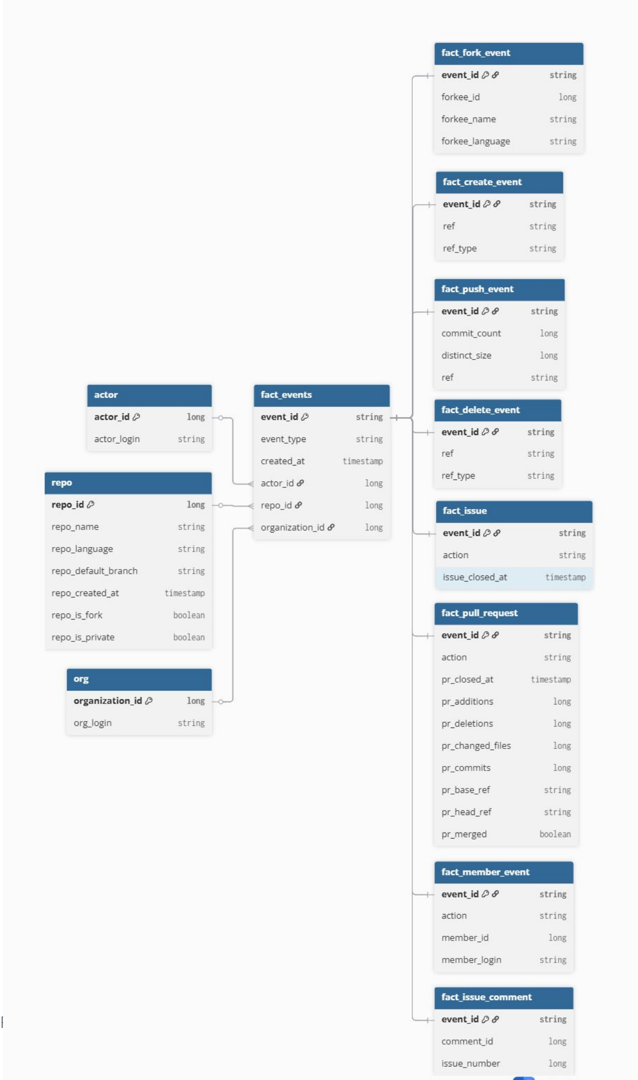
Silver
- Flattened, cleaned, 3NF tables + lookup tables
- Parquet Format, split into 128 MB Partitions

Use Spark to
- Reorganize tables into Star Schema

Gold
- Star Schema
- Fact Table: Events
- Various Dimension Tables (TBD)
- Parquet Format

Analytics
- Various Aggregation Queries
- Visualizations

Silver ERD



actor	
actor_id	long
actor_login	string

repo	
repo_id	long
repo_name	string
repo_language	string
repo_default_branch	string
repo_created_at	timestamp
repo_is_fork	boolean
repo_is_private	boolean

org	
organization_id	long
org_login	string

fact_events	
event_id	string
event_type	string
created_at	timestamp
actor_id	long
repo_id	long
organization_id	long

fact_fork_event	
event_id	string
forkee_id	long
forkee_name	string
forkee_language	string

fact_create_event	
event_id	string
ref	string
ref_type	string

fact_push_event	
event_id	string
commit_count	long
distinct_size	long
ref	string

fact_delete_event	
event_id	string
ref	string
ref_type	string

fact_issue	
event_id	string
action	string
issue_closed_at	timestamp

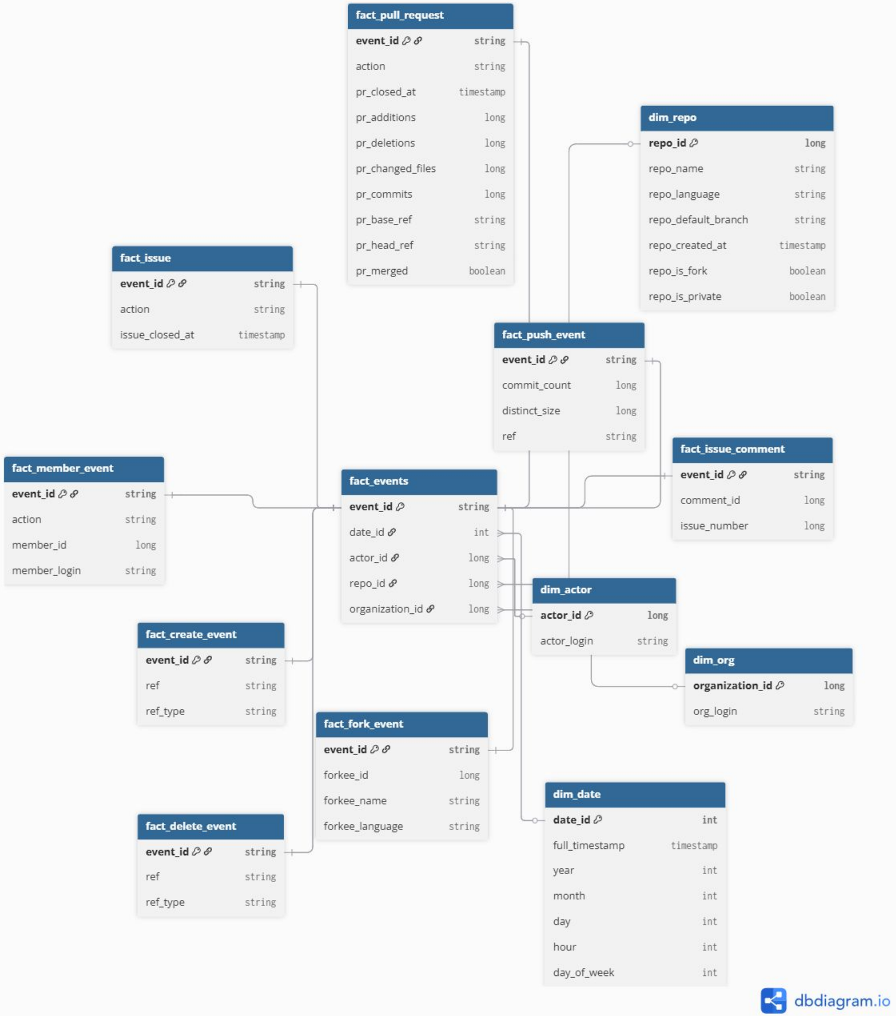
fact_pull_request	
event_id	string
action	string
pr_closed_at	timestamp
pr_additions	long
pr_deletions	long
pr_changed_files	long
pr_commits	long
pr_base_ref	string
pr_head_ref	string
pr_merged	boolean

fact_member_event	
event_id	string
action	string
member_id	long
member_login	string

fact_issue_comment	
event_id	string
comment_id	long
issue_number	long

SKILLSTORM

Gold ERD



fact_pull_request	
event_id 🔑🔗	string
action	string
pr_closed_at	timestamp
pr_additions	long
pr_deletions	long
pr_changed_files	long
pr_commits	long
pr_base_ref	string
pr_head_ref	string
pr_merged	boolean

dim_repo	
repo_id 🔑	long
repo_name	string
repo_language	string
repo_default_branch	string
repo_created_at	timestamp
repo_is_fork	boolean
repo_is_private	boolean

fact_issue	
event_id 🔑🔗	string
action	string
issue_closed_at	timestamp

fact_push_event	
event_id 🔑🔗	string
commit_count	long
distinct_size	long
ref	string

fact_issue_comment	
event_id 🔑🔗	string
comment_id	long
issue_number	long

fact_member_event	
event_id 🔑🔗	string
action	string
member_id	long
member_login	string

fact_events	
event_id 🔑	string
date_id 🔗	int
actor_id 🔗	long
repo_id 🔗	long
organization_id 🔗	long

dim_actor	
actor_id 🔑	long
actor_login	string

dim_org	
organization_id 🔑	long
org_login	string

fact_create_event	
event_id 🔑🔗	string
ref	string
ref_type	string

fact_fork_event	
event_id 🔑🔗	string
forkee_id	long
forkee_name	string
forkee_language	string

fact_delete_event	
event_id 🔑🔗	string
ref	string
ref_type	string

dim_date	
date_id 🔑	int
full_timestamp	timestamp
year	int
month	int
day	int
hour	int
day_of_week	int

Bronze to Silver Pipeline

EDA & Selecting Data

- Explore the data
- Flatten JSON into one dataframe
- Select columns with analytical value

Clean & Normalize

- Handle nulls and duplicates
- Normalize data (clean, organized, and not unnecessarily repeated)

Repartition & Write

- Estimate size of each cleaned dataframe
- Repartition so each parquet files would be optimal size
- Files stored in ADLS

Silver to Gold Pipeline

Fact Tables

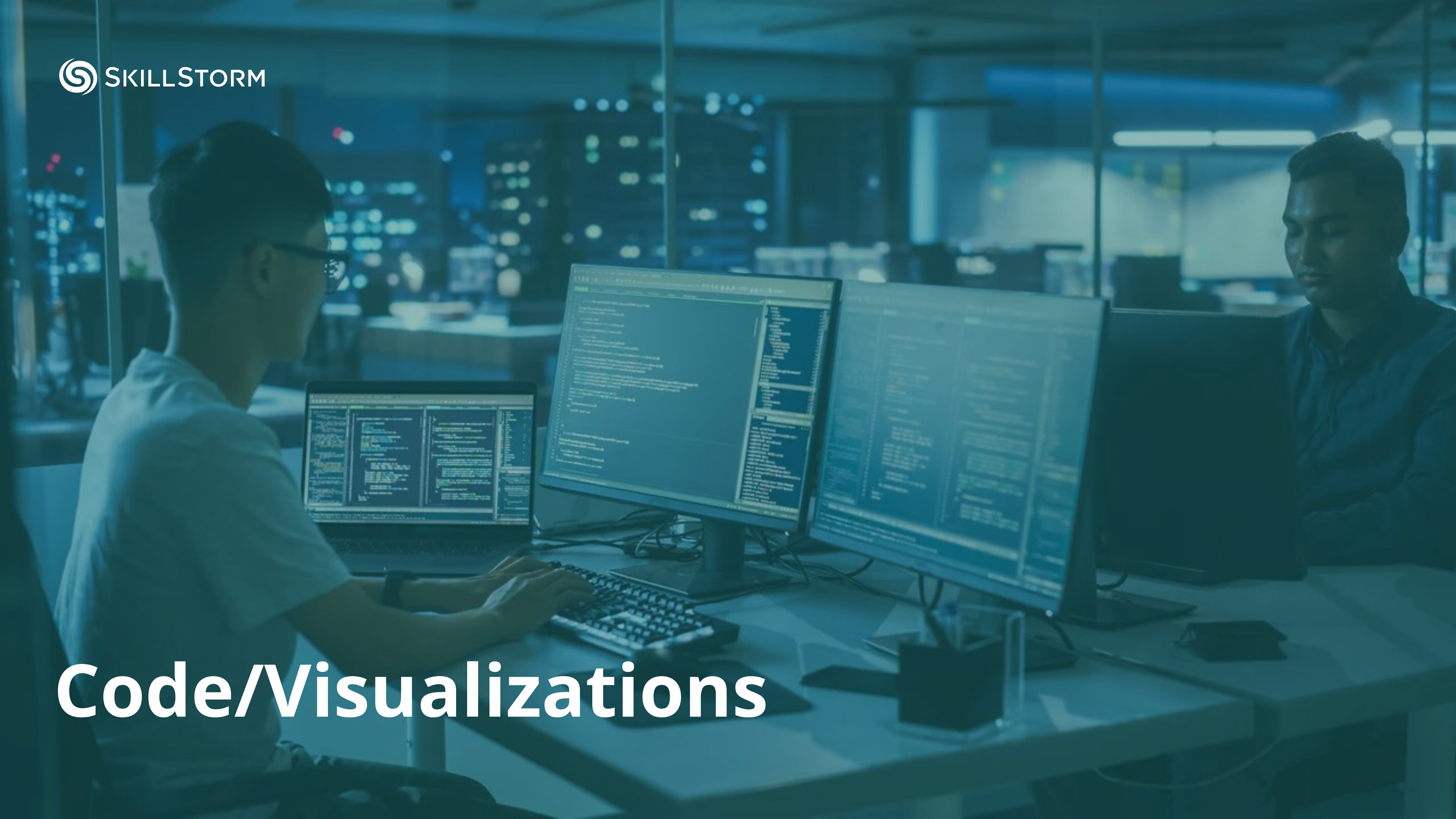
- Organize data into the main tables that track important info and measurements
- One row = One event

Dimension Tables

- Supporting tables referenced by fact tables
- Add context like names and descriptions

Aggregation

- Created from Fact Tables
- Summarize data to find patterns
- Used to create visuals and business insights



Code/Visualizations

Conclusions

Our project shows an end-to-end Databricks pipeline that:

- Took in 73 GBs of gzipped raw JSON data
- Cleaned and normalized into 6.8 GB of clean silver parquets
- Organized and aggregated into ~600 MB of gold parquets
- Easy analytics and visualizations from gold data