NYC Motor Vehicle Collision Data Warehouse

Sourcing, cleaning, and modeling NYC Open Data end to end

@Shaquille J.

The Business Problem

NYC Open Data publishes every police reported motor vehicle collision across the five boroughs. The raw feed amasses ~2.2 million rows with no analytical layer. Answering any question means scanning the raw file from scratch each time.

- Limited way to filter or aggregate by borough, street, time of day, or contributing factor
- Manual spreadsheet analysis breaks down once you're past a few hundred thousand rows
- No repeatable pipeline, so every analysis starts over instead of pulling from clean, modeled tables
- No self-service reporting layer for non-technical stakeholders

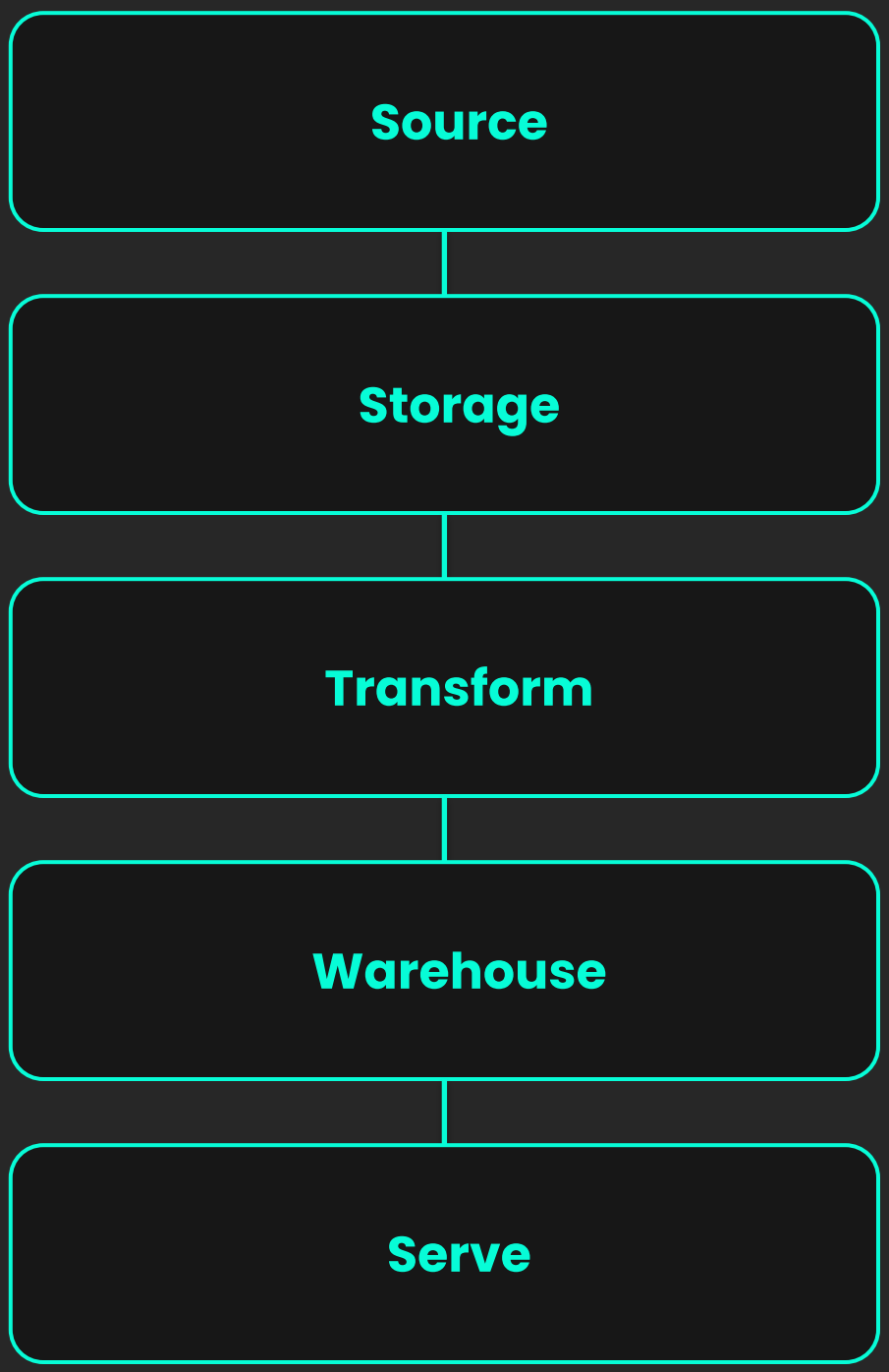
Solution: A Cloud-Native Data Pipeline

To transform this raw open data into something users could actually query, I designed and built an end-to-end ETL pipeline covering sourcing, storage, transformation, warehousing, and serving.

- **Source:** scripted pull from the NYC Open Data Web API
- **Storage:** raw files staged in a cloud storage bucket, dated on ingestion
- **Transform:** Python and Pandas clean, standardize, and restructure the data
- **Modeling:** a star schema separates measurable facts from descriptive context
- **Serve:** a BI dashboard and a CSV export API put the data in front of end users

@Shaquille J.

Pipeline Architecture



NYC Open Data Web API

Cloud Storage bucket (raw)

Python / Pandas ETL

Star schema (fact + dimensions)

BI dashboard + CSV export API

@Shaquille J.

Star Schema Design

The data is modeled around a single grain, one row per collision involving at least one injury or fatality.

- **fact_collision:** the grain: one collision event, with injury and fatality counts, coordinates, and foreign keys to every dimension

- **dim_date:** calendar attributes (year, month, day, weekday) for trend analysis

- **dim_time:** clock time broken into hour and minute for time-of-day patterns

- **dim_location:** borough and zip code for geographic analysis

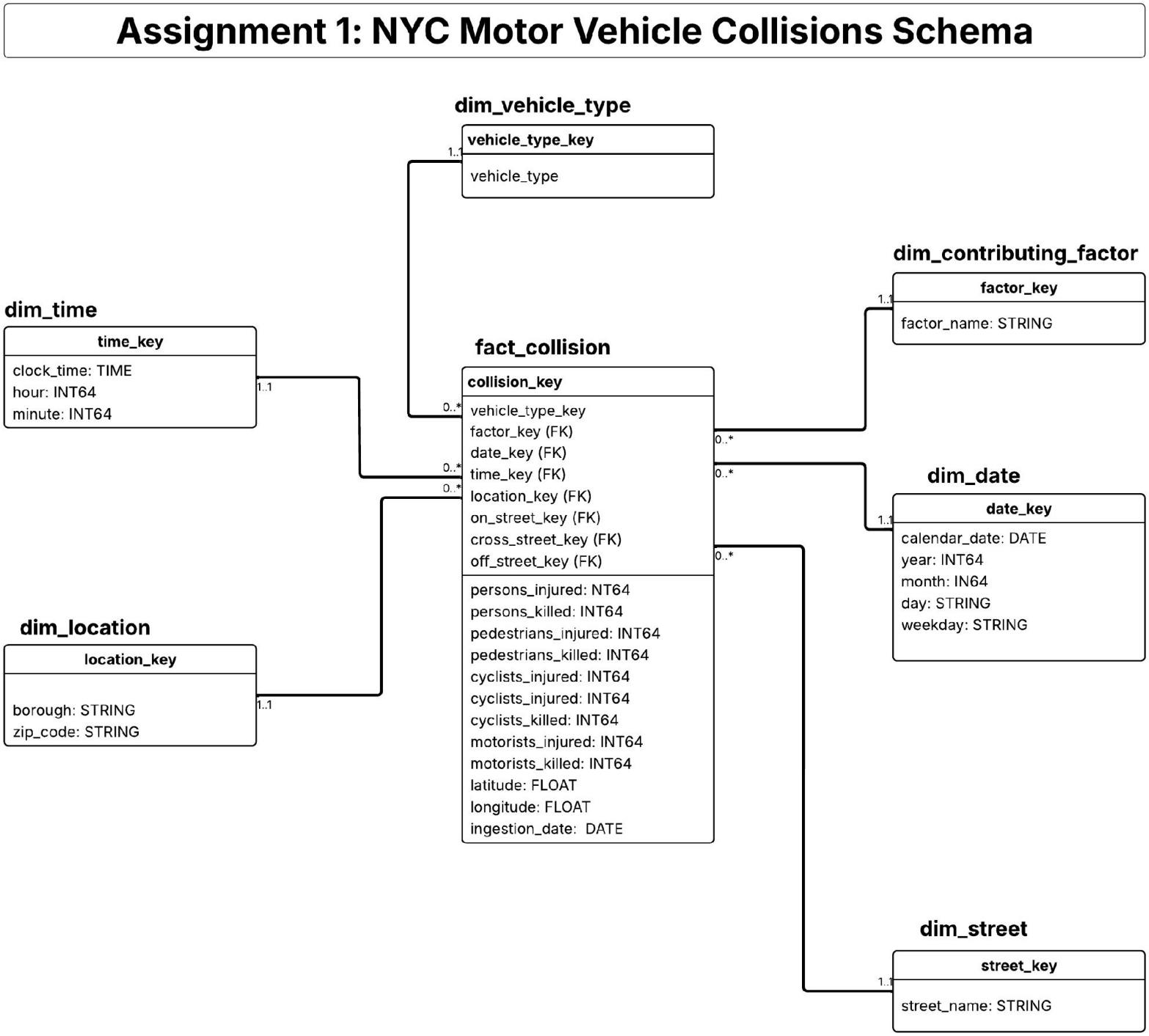
- **dim_street:** normalized on-street, cross-street, and off-street names

- **dim_vehicle_type:** standardized primary vehicle classification

- **dim_contributing_factor:** standardized primary contributing factor

@Shaquille J.

Entity Relationship Diagram



@Shaquille J.

Sample Transformation Code

```python
# parse crash datetime
dt = pd.to_datetime(df["crash_date"] + " " + df["crash_time"])

# unified date/time pieces for the star schema
df["calendar_date"] = dt.dt.date
df["year"]    = dt.dt.year
df["weekday"] = dt.dt.day_name()
df["hour"]    = dt.dt.hour
```

```python
# build dim_date and assign a surrogate key
dim_date = (
    df[["calendar_date", "year", "weekday"]]
    .drop_duplicates(subset=["calendar_date"])
    .reset_index(drop=True)
)
dim_date["date_key"] = dim_date.index + 1
```

Full ETL breakdown: *LINK*

@Shaquille J.

Business Questions & Insights

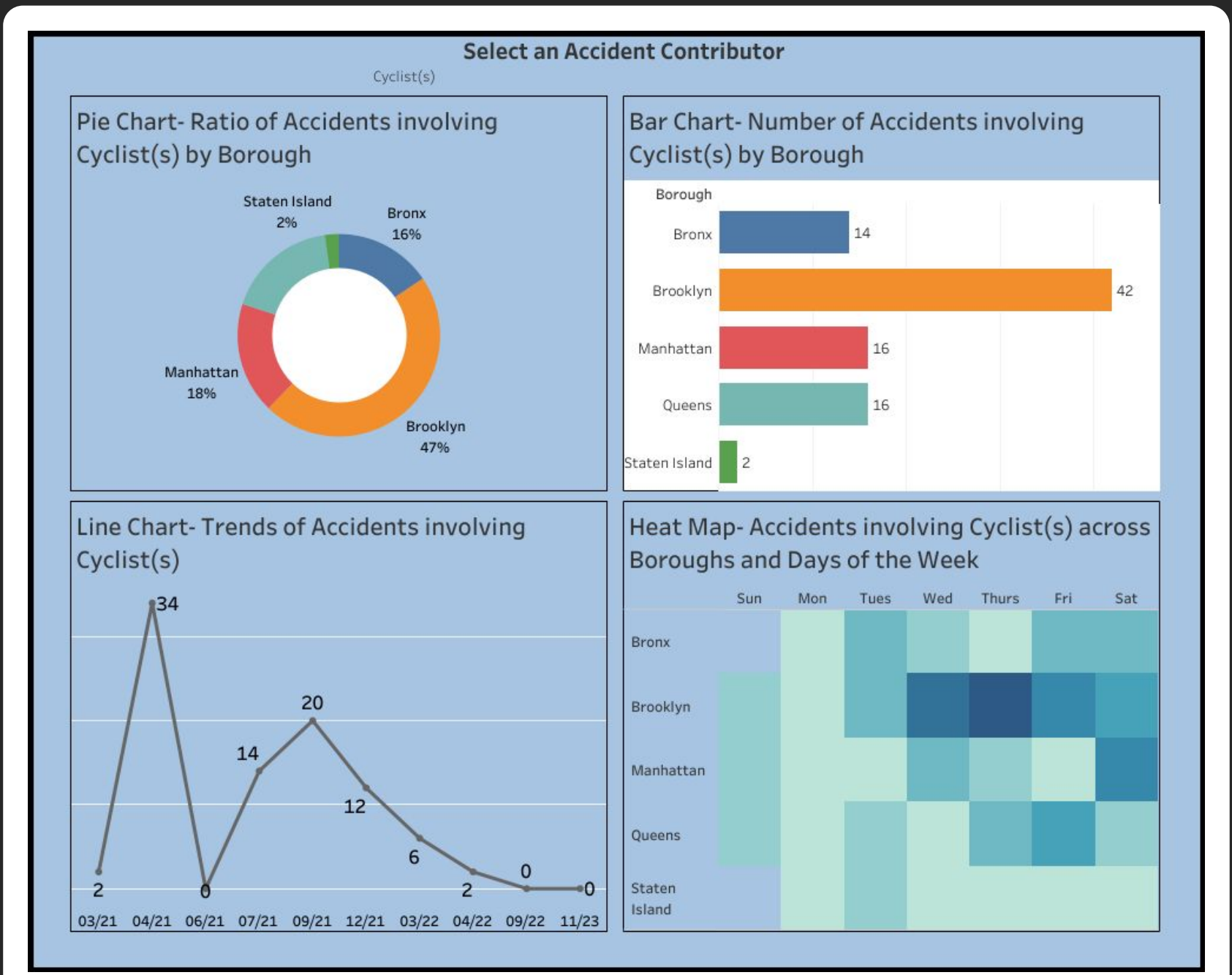
Once the star schema was in place, borough-level and time-based safety questions that used to take hours in spreadsheets started resolving in a single query.

- Which boroughs see the highest concentration of cyclist-involved collisions?
- What time of day and day of week see the most crashes?
- Which contributing factors appear most often?
- How have injury and fatality trends shifted over time?

Sample Insight: Brooklyn accounts for 47% of all cyclist-involved collisions citywide, more than 2.5x the next highest borough.

Data Serving

The star schema feeds an interactive Tableau dashboard with a borough and date filter, a pie chart, a bar chart, a trend line, and a day-of-week heat map, so any stakeholder can get answers without writing SQL.



@Shaquille J.

Analytical Impact

Faster Insight
Borough- and time-based safety questions that took hours in spreadsheets now resolve in seconds.

Scalable by Design
The warehouse handles millions of collision records without the row limits of a spreadsheet.

Self-Service Access
A CSV export API lets analysts pull governed data without warehouse access or SQL knowledge.

@Shaquille J.

Upcoming Improvements

Automated Orchestration
Move from manual runs to a scheduled pipeline that ingests new collision data automatically.

Enriched Context
Join in weather and traffic-volume data to correlate conditions with collision risk.

Predictive Modeling
Flag high-risk corridors and time windows before incidents happen.

Why a Star Schema?

A star-schema warehouse provides:

- **Query performance at scale:** aggregations run in seconds, even over millions of rows
- **Conformed dimensions:** one shared definition of date, location, and street reused across every fact
- **Decoupled layers:** raw data stays untouched while the warehouse serves analysis-ready tables
- **BI-ready modeling:** plugs directly into Tableau, Looker, or Power BI with no extra transformation

Lessons Learned

Building this pipeline end to end taught me a few things:

- **Real-world data is messy:** inconsistent casing, missing values, and duplicate zero-impact rows all had to be cleaned up before modeling could even start

- **Grain is the hardest decision:** it determines every join and every analysis you'll ever run

- **Cloud-native tooling removes overhead:** serverless storage and a managed warehouse expedited development with zero servers to provision or patch

Conclusion

I took raw NYC Open Data and turned it into a fully vetted, analysis-ready warehouse through API sourcing, cloud storage, star-schema modeling, and BI-ready serving.

01 Faster, self-service safety insights for any stakeholder

02 A reusable schema ready for new data sources

03 A foundation for predictive, proactive safety analytics

@Shaquille J.

Thank You!

@Shaquille J.