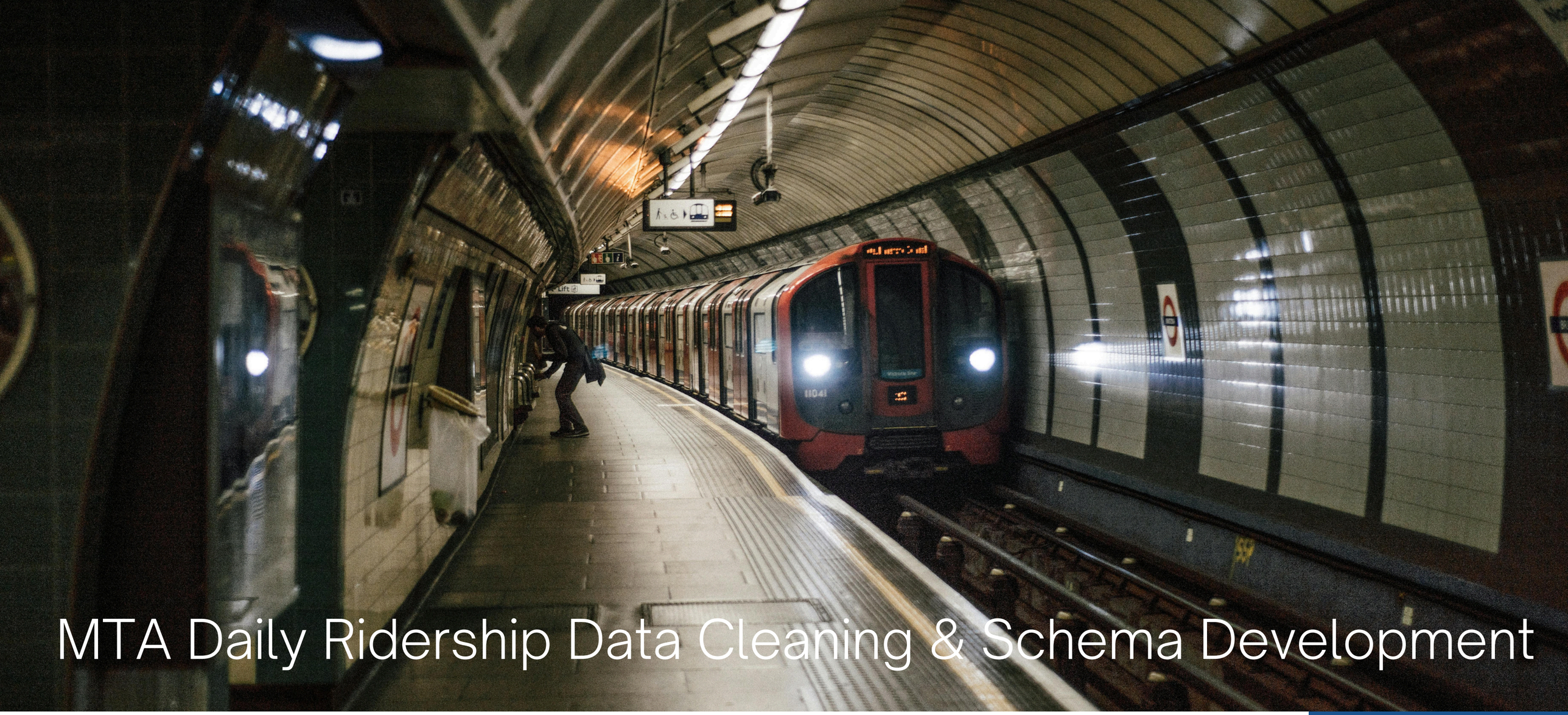


MTA Daily Ridership Data Cleaning & Schema Development

SQL & Python Implementation

O

• OBJECTIVE:

– Develop a robust process to clean and structure raw MTA daily ridership data.
– Establish a solid database schema to support further analysis.

D

• DATASET CONTEXT:

– Data covers multiple MTA agencies (NYCT, LIRR, Metro-North, Access-A-Ride, Bridges & Tunnels).
– Focus on daily ridership counts and percentage comparisons to pre-pandemic levels.

Data Challenges & Initial Issues

- Identified Problems:

– Missing values (e.g., ride_date_str, station_id, ridership counts).
– Negative or out-of-range numeric values.
– Extra spaces in text fields.
– Duplicate records that needed removal.

- Impact:

– Inconsistent raw data required a comprehensive cleaning process to ensure data integrity.

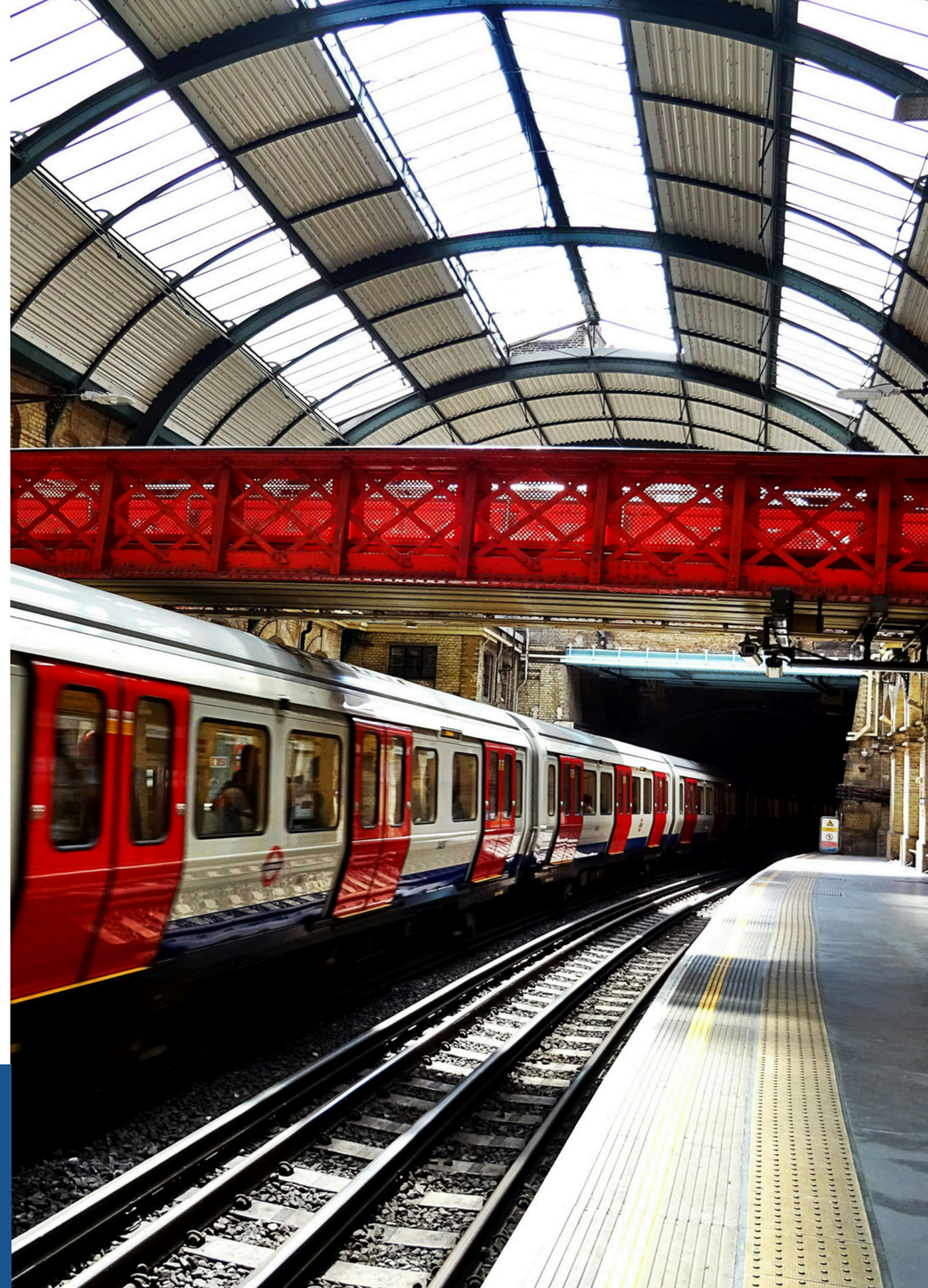
SQL Data Cleaning & Schema Creation

- SQL Cleaning Process:

– Date Conversion: Converted raw text dates to proper DATE types.
– Numeric Data:
- Replaced NULL or negative values with 0.
- Capped extreme values (e.g., ridership > 100,000; percentages > 200).
– Duplicate Removal: Employed a window function (ROW_NUMBER()) to remove duplicate entries based on date.

- Schema Development:

– Designed a relational schema to store transit modes and ridership data.
– See ERD details from the schema document.



Python Data Transformation

01

• DATA TYPE ADJUSTMENTS:

– Converted percentage columns from int64 to float64 for greater precision.

02

• HANDLING MISSING VALUES & DUPLICATES:

– Utilized pandas (df.fillna(0)) to address missing values.
– Verified and removed duplicates with df.duplicated() and df.drop_duplicates().

03

• OUTCOME:

– A cleaned and consistent DataFrame ready for further analysis.

Database Schema & ERD

- **Schema Overview:**

– Transit_Modes Table:
- **TransitModeID (Primary Key), Name, etc.**

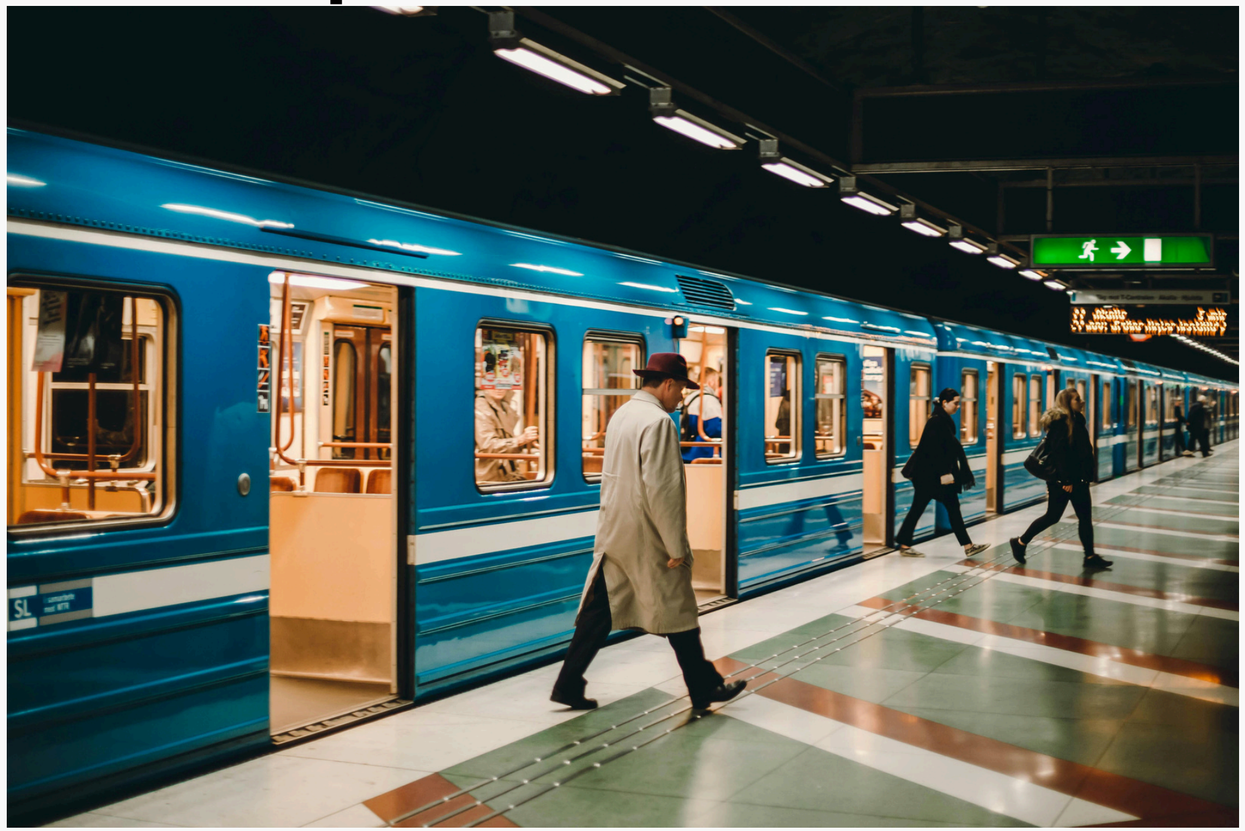
– Ridership Table:
- **Date, TransitModeID (Foreign Key), TotalValue, PercentPrePandemic, etc.**

- **ERD Details:**

– The ERD illustrates the relationship between transit modes and daily ridership data .

– This schema ensures data integrity and facilitates efficient querying.

Quality Checks & Performance Optimization



- **DATA QUALITY CHECKS (SQL):**

– Verified that no erroneous default dates (e.g., '1900-01-01') remain.
– Checked for any remaining negative or outlier values.

- **PERFORMANCE ENHANCEMENTS:**

– Created indexes on key columns (e.g., station_id, ride_date) to improve query speed.
– Ensured deduplication for optimal database performance.

Summary & Next Steps

- **NEXT STEPS:**

– Use the clean, structured dataset for further analysis.
– Consider expanding the schema or integrating additional datasets as needed.

- **ACHIEVEMENTS:**

– Developed a robust SQL-based cleaning process and a clear schema.
– Performed additional data transformations using Python to ensure consistency.