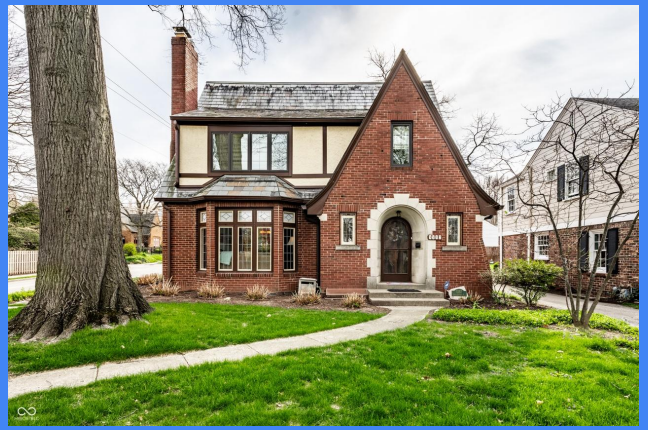
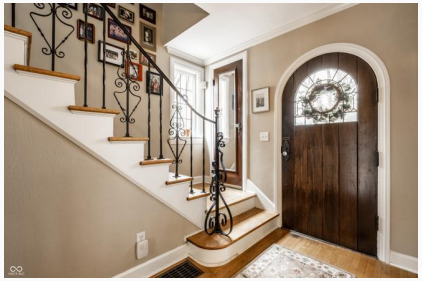
Real Estate Market Research App Demo



by: Ronald Daley

Introduction

- My Background
  - Passionate about real estate investing
  - Worked in real estate industry for 4 years
- Project Motivation
  - Develop a natural language (NL) tool that allows investors to interact with real estate-related databases to make data driven decisions when investing in a new market.
- Project Goal
  - The tool will convert questions from investors into postgreSQL queries for efficient data retrieval, analysis, and modification.



Data

- Zillow Housing Data tables
  - House value
  - For Sale Inventory
  - New Listing Inventory
  - Median List Price
- Assumptions
  - Data Type = Time series (ALL TABLES)
  - Frequency = Monthly
  - Geography = Metro level (City, State)
  - Duration = 5 years of data:
    - start = '2020-02-28'
    - end = '2025-02-28'
- Data prep and EDA in Python

Sample data: Home Value table

	RegionID	2020-02-29	2020-03-31	2020-04-30	2020-05-31	2020-06-30	2020-07-31	2020-08-31	2020-09-30	2020-10-31	...	2024-07-31	2024-08-31	2024-09-30	2024-10-31	2024-11-30	2
0	102001	4.710990e+05	4.734481e+05	4.753970e+05	4.762181e+05	4.763861e+05	4.770996e+05	4.797424e+05	4.847230e+05	4.914446e+05	...	6.930996e+05	6.941272e+05	6.957628e+05	6.978330e+05	6.998435e+05	7.0
1	394913	7.417605e+05	7.429928e+05	7.441946e+05	7.442005e+05	7.434160e+05	7.432395e+05	7.457132e+05	7.515983e+05	7.604155e+05	...	1.102751e+06	1.108185e+06	1.113485e+06	1.118237e+06	1.121632e+06	1.
2	753899	1.129119e+06	1.131086e+06	1.132641e+06	1.129972e+06	1.123791e+06	1.121600e+06	1.128527e+06	1.145479e+06	1.166966e+06	...	1.729674e+06	1.738847e+06	1.750788e+06	1.761684e+06	1.770720e+06	1.
3	394463	4.267689e+05	4.283210e+05	4.296867e+05	4.299479e+05	4.291088e+05	4.284059e+05	4.287944e+05	4.313542e+05	4.358986e+05	...	5.897441e+05	5.916298e+05	5.940620e+05	5.967046e+05	5.994185e+05	6.0
4	394514	4.487976e+05	4.509599e+05	4.520351e+05	4.519391e+05	4.514519e+05	4.516843e+05	4.536836e+05	4.577002e+05	4.631510e+05	...	7.124083e+05	7.112924e+05	7.113514e+05	7.125698e+05	7.139178e+05	7.

5 rows × 64 columns

4-08-31	2024-09-30	2024-10-31	2024-11-30	2024-12-31	2025-01-31	2025-02-28	City	State
272e+05	6.957628e+05	6.978330e+05	6.998435e+05	7.013544e+05	7.023234e+05	7.027477e+05	United States	NaN
185e+06	1.113485e+06	1.118237e+06	1.121632e+06	1.123018e+06	1.124979e+06	1.128410e+06	New York	NY
847e+06	1.750788e+06	1.761684e+06	1.770720e+06	1.777742e+06	1.778364e+06	1.775890e+06	Los Angeles	CA
298e+05	5.940620e+05	5.967046e+05	5.994185e+05	6.015685e+05	6.039141e+05	6.060862e+05	Chicago	IL
924e+05	7.113514e+05	7.125698e+05	7.139178e+05	7.141215e+05	7.132128e+05	7.112661e+05	Dallas	TX

Data Insert Process

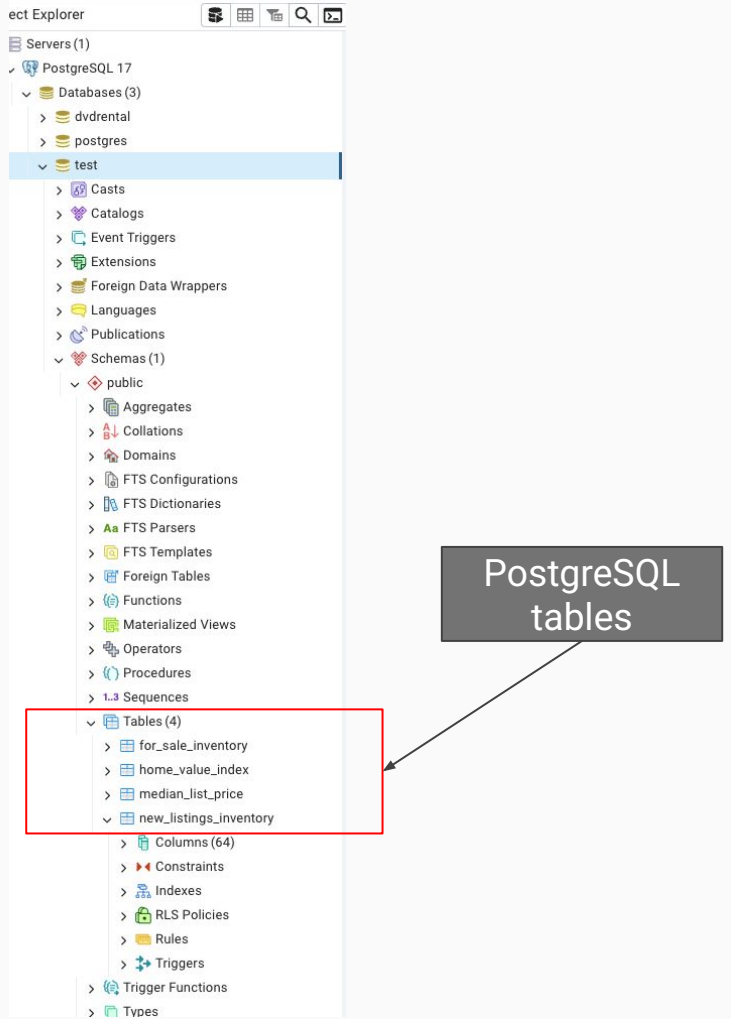
- Download data from Zillow
- Clean data and create Dataframe tables via Python
- Insert data from DataFrames into into PostGres tables via Python

df1_to_table: home value index table

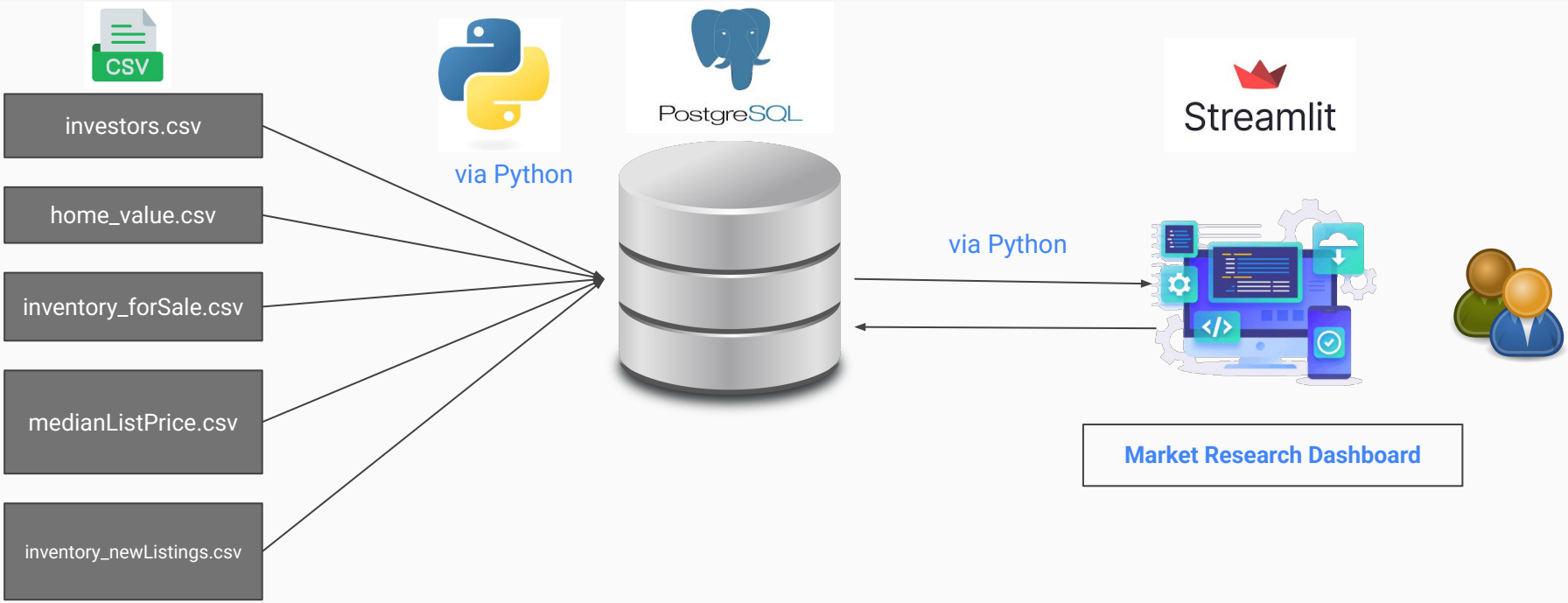
```
# map pandas dype to Postgres types

type_map = {
    "int64": "BIGINT",
    "float64": "DOUBLE PRECISION",
    "object": "TEXT",
    "datetime64[ns]": "TIMESTAMP"
}

columns = []

for col, dtype in zip(df1_to_table.columns, df1_to_table.dtypes):
    pg_type = type_map.get(str(dtype), "TEXT")
    columns.append(
        sql.SQL("{} {}").format(
            sql.Identifier(col),
            sql.SQL(pg_type)
        )
    )

create_table = sql.SQL("CREATE TABLE {} ({})").format(
    sql.Identifier("home_value_index"),
    sql.SQL(", ").join(columns)
)

cur.execute(create_table)

# Bulk insert into table

insert_stmt = sql.SQL("INSERT INTO {} ({}) VALUES %s").format(
    sql.Identifier("home_value_index"),
    sql.SQL(" ,").join(map(sql.Identifier, df1_to_table.columns))

)

# convert Nan to Null

data = [
    tuple(None if pd.isna(val) else val for val in row)
    for row in df1_to_table.itertuples(index = False, name = None)

]
```

RDBMS Database

- PostgreSQL
- PgAdmin

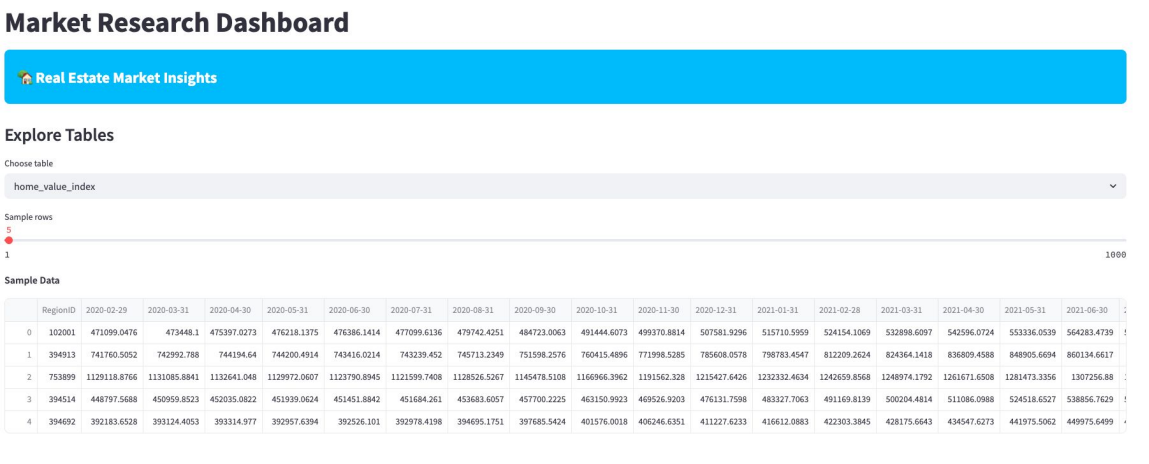


Architecture Design



Home Affordability Dashboard

- Streamlit app
- Dashboard Functionality:
  - Explore Tables
  - Pre-built Sample Queries
  - Query Execution (NL→SQL)
  - Insert, Create, Delete into database via NL



DEMO

(v2app.py)

Conclusion

- Developed a **relational database using PostgreSQL** to efficiently store and query time-series real estate market data, facilitating structured and scalable analysis.

- Created an interactive **market research dashboard that utilizes large language models** to answer questions and provide prospective investors with data-driven insights and customizable queries.

Next Steps

- Add more data variety (i.e. income, population, occupation, demographic statistics, crime rate, % of renters)
- Add more visualizations (interactive maps, charts, etc..)
- Improve Overall User Experience of dashboard
    - Migrate to Tableau

Thank you!

Appendix

Demo Notes

- Update query execution
- Include a JOIN clause sql
- IMPORTANT: Make sure ReadMe includes descriptive steps on implementation
- For the database, submit the .tar file?
- Remove API keys