Hospi Housing | A complete ETL pipeline and BI Dashboard

# Introduction

Hospi Housing is an application that connects hosts who want to offer their houses for living to students seeking accommodation in the Netherlands. The application collects various types of data about hosts, students, and their interactions, storing it in DynamoDB, which is configured in AWS. Our task was to set up an ETL pipeline to extract data from DynamoDB, transform it into the correct format, and make it available for visualization. To accomplish this, we utilized several AWS services, including S3, Glue, Athena, and QuickSight. Figure 1.1 shows the workflow of our ETL pipeline

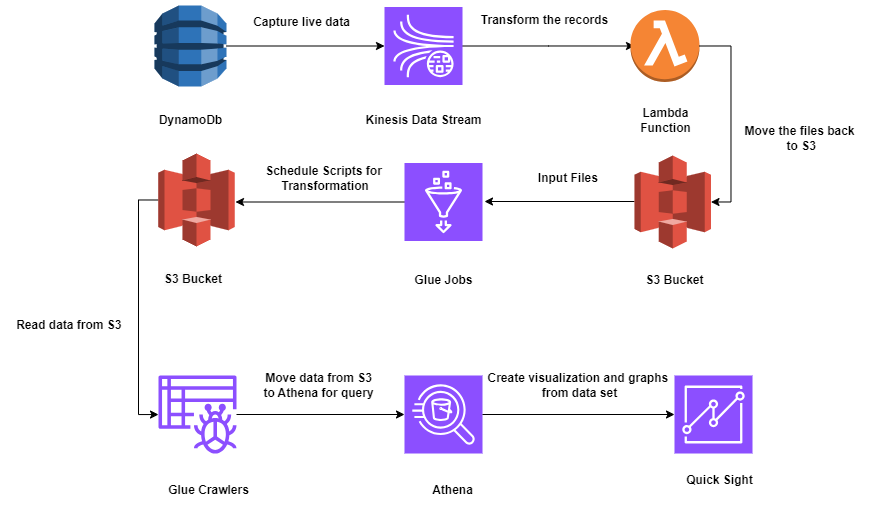


Figure 1.1 Workflow of the ETL pipeline for Hospi Housing

# Data Extraction

Figure 2.1 provides a snapshot of our data in DynamoDB, a NoSQL database that organizes data into key-value pairs with primary keys assigned to each table's respective IDs. Initially, we performed a full export of our tables, capturing all historical data across the columns. This export was saved in S3 as JSON files, as illustrated in figure 2.2.

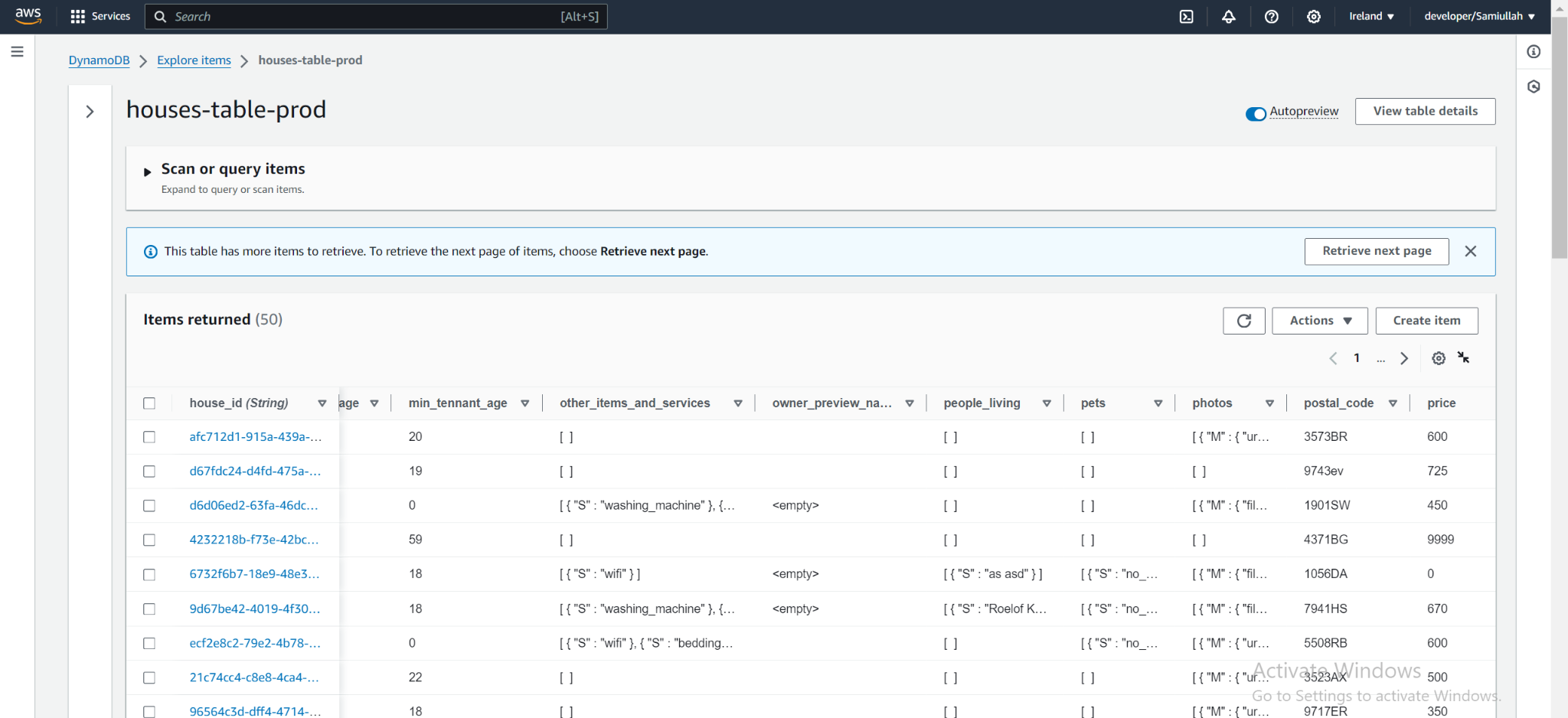


Figure 2.1 Key-value pair of data in DynamoDB



Figure 2.2 JSON file of data stored in S3

Apart from historical, Kinesis data stream was also in place along with AWS lambda function that was giving us the hourly live data into the S3 bucket in the form of parquet file as shown in figure 2.3

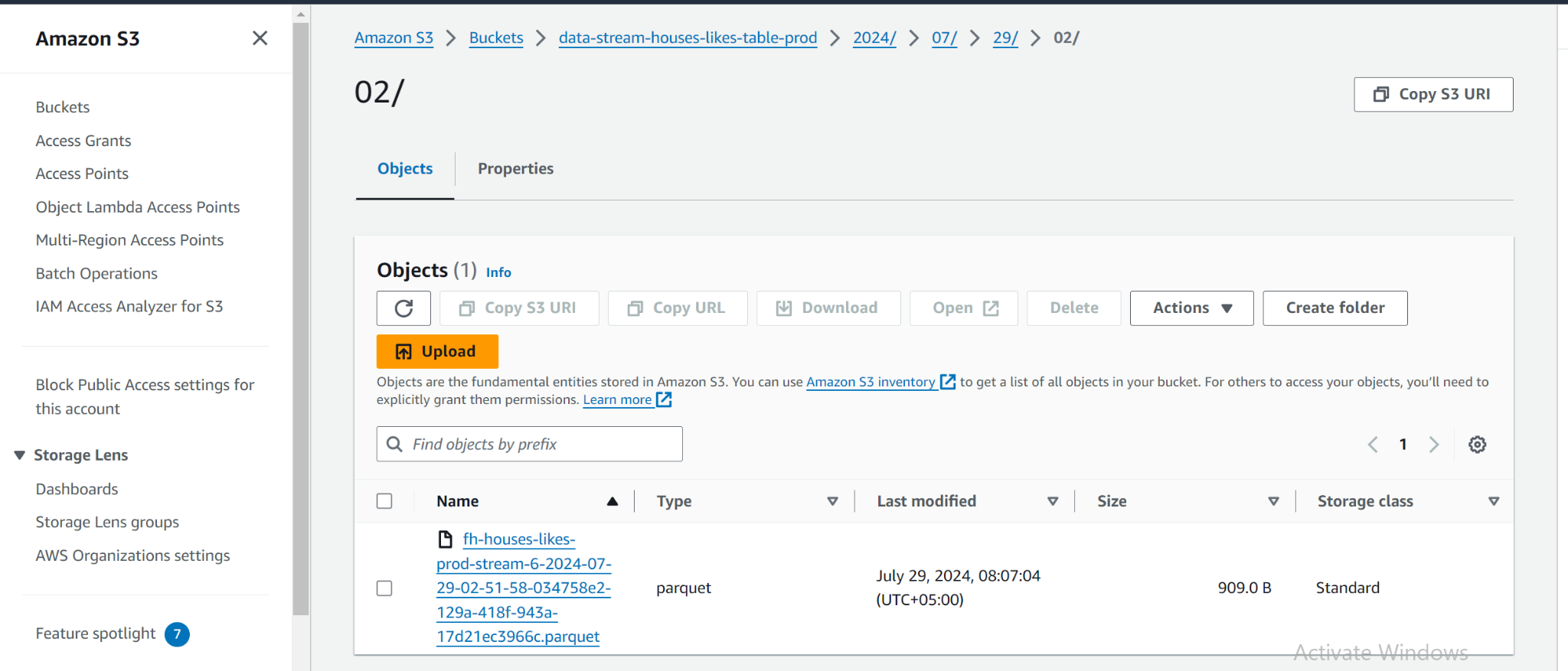


Figure 2.3 Parquet file coming from Kinesis live data stream in S3

# Data Transformation

There are two types of transformation that is undergoing in our process

1. Transformation using Python via Glue jobs:
   1. Transformed the semi-structured JSON data into a DataFrame by properly casting column names and attributes, then stored the historical data in S3 as Parquet files.
   2. A Glue job has been scheduled to concatenate live data from Kinesis with historical data, remove duplicates, and export the incremental data as a Parquet file to S3 as shown in figure 3.1. This process is dynamic, with jobs scheduled to run every 24 hours, automatically concatenating the next day's data each time the Glue job executes.

After transformation, Glue crawlers were set that created the tables in the glue catalog

which were queried in Athena using SQL

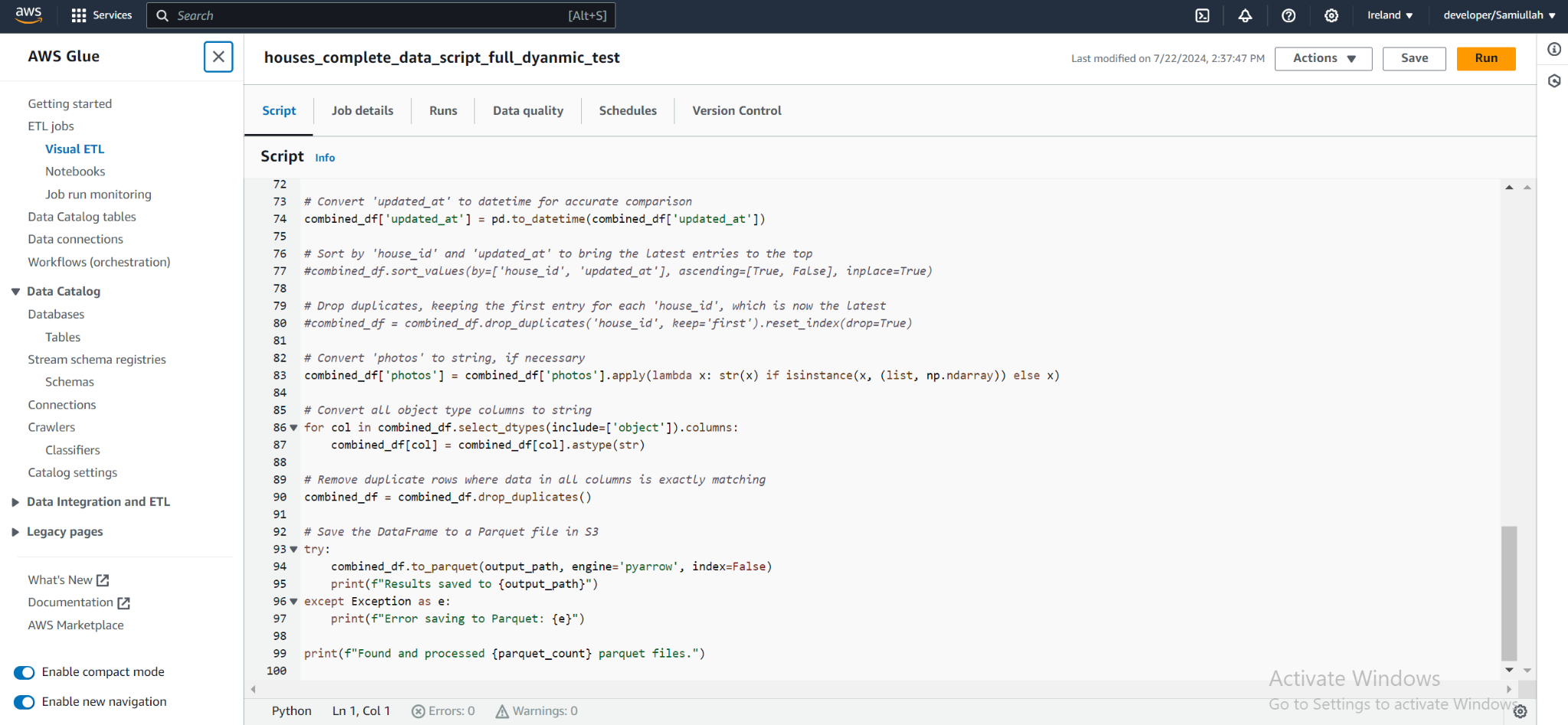


Figure 3.1 Glue job that transforms and updates the data after every 24 hours

1. Transformation using SQL queries

As our crawler ran and the data became available in the Glue catalog, we observed that further cleaning was required to achieve the desired visualization results. Consequently, we executed SQL queries and created views as part of our final transformed tables as shown in figure 3.2. This process included joining tables, flattening columns, and correcting data types according to the columns.

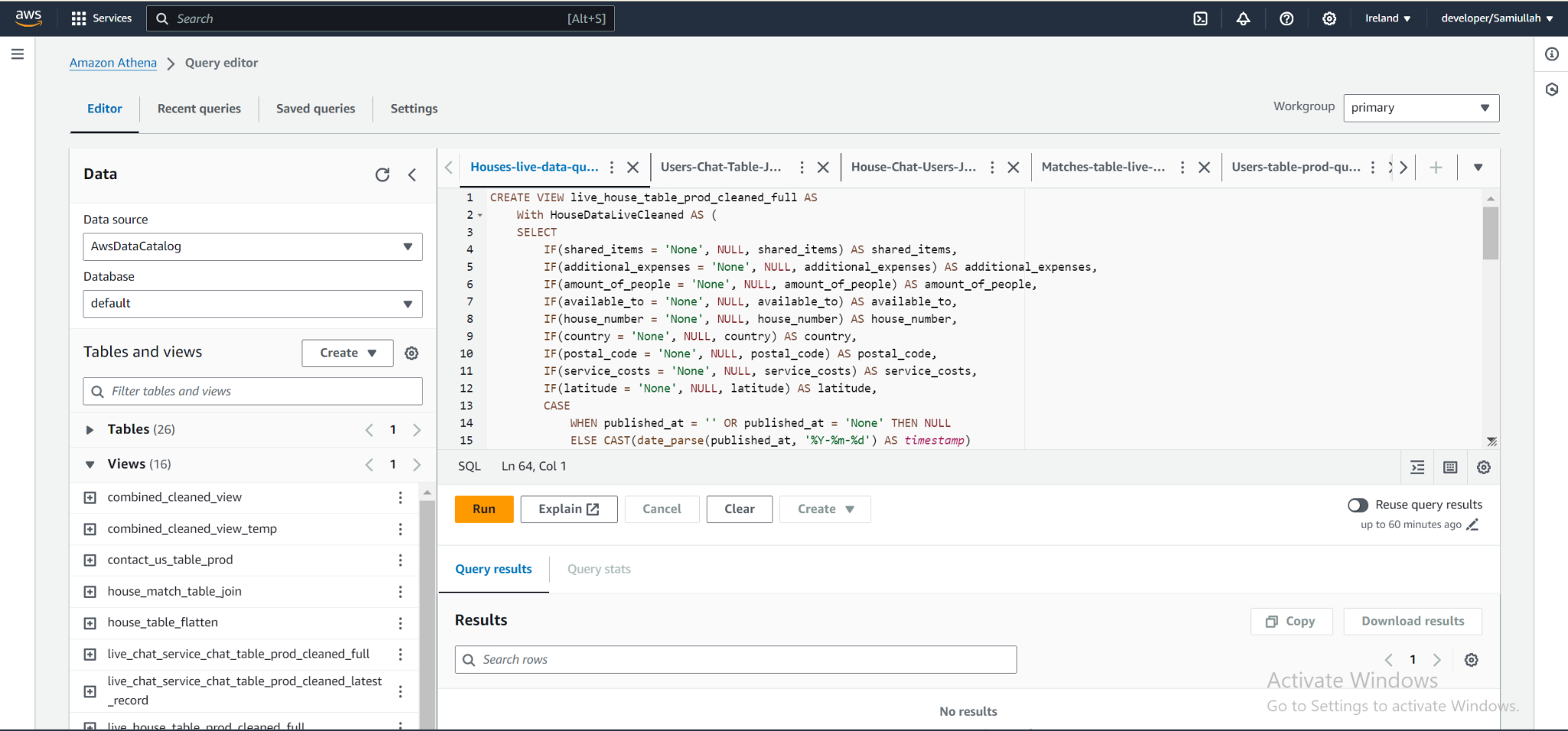


Figure 3.2 Creating Views of tables using Athena

# Data Loading & BI:

The final views created in the Glue catalog via Athena are used to establish connections with QuickSight, a visualization tool that allows us to create interactive dashboards for generating valuable insights from data. We set up Athena connections for these views, redirecting the data to our analysis sheet. This enabled us to integrate various graphs, such as time-series graphs, bar graphs, maps, percentage funnels, and line graphs, as illustrated in Figure 4.1 and 4.2

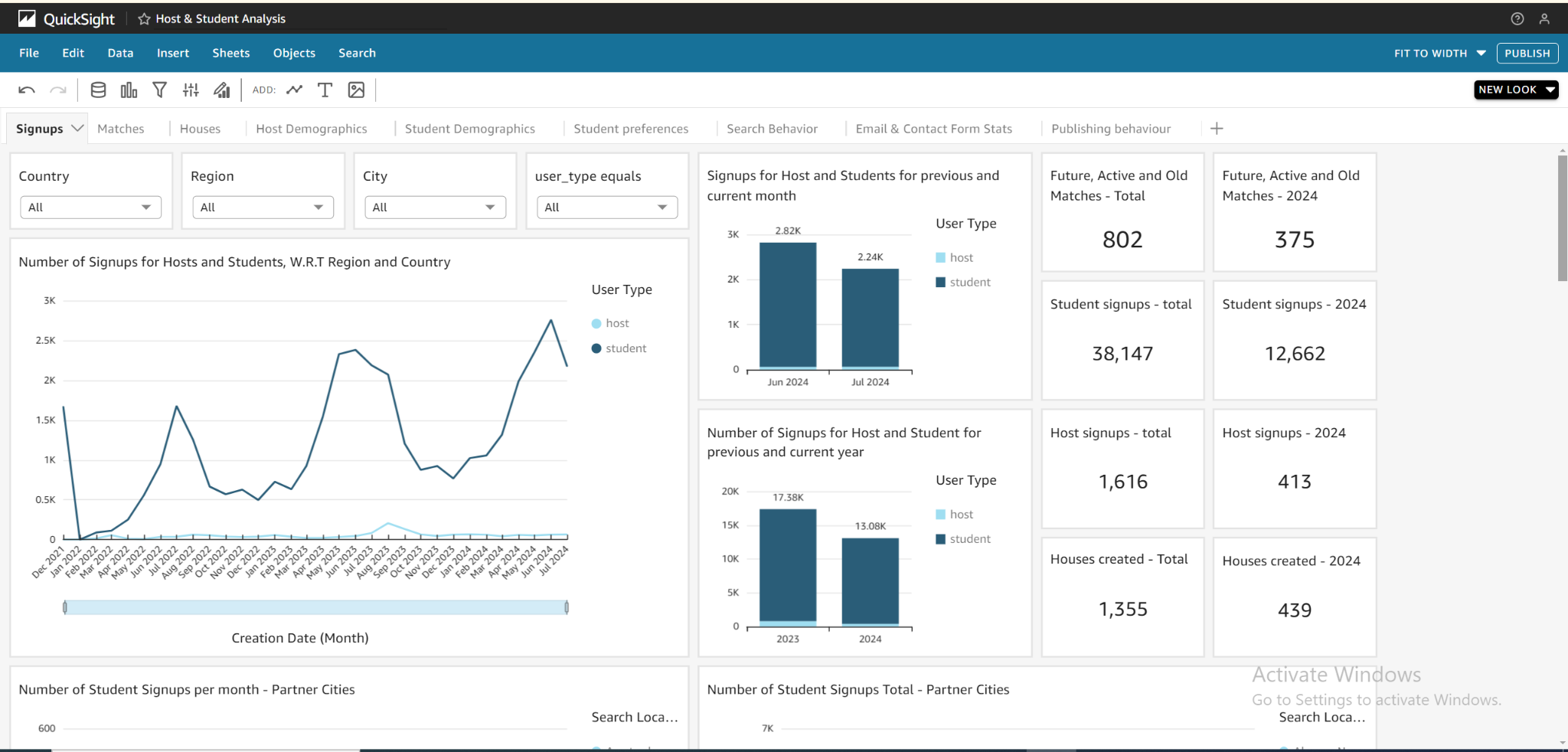


Figure 4.1 Graphical representation of User Signups in Hospi Housing

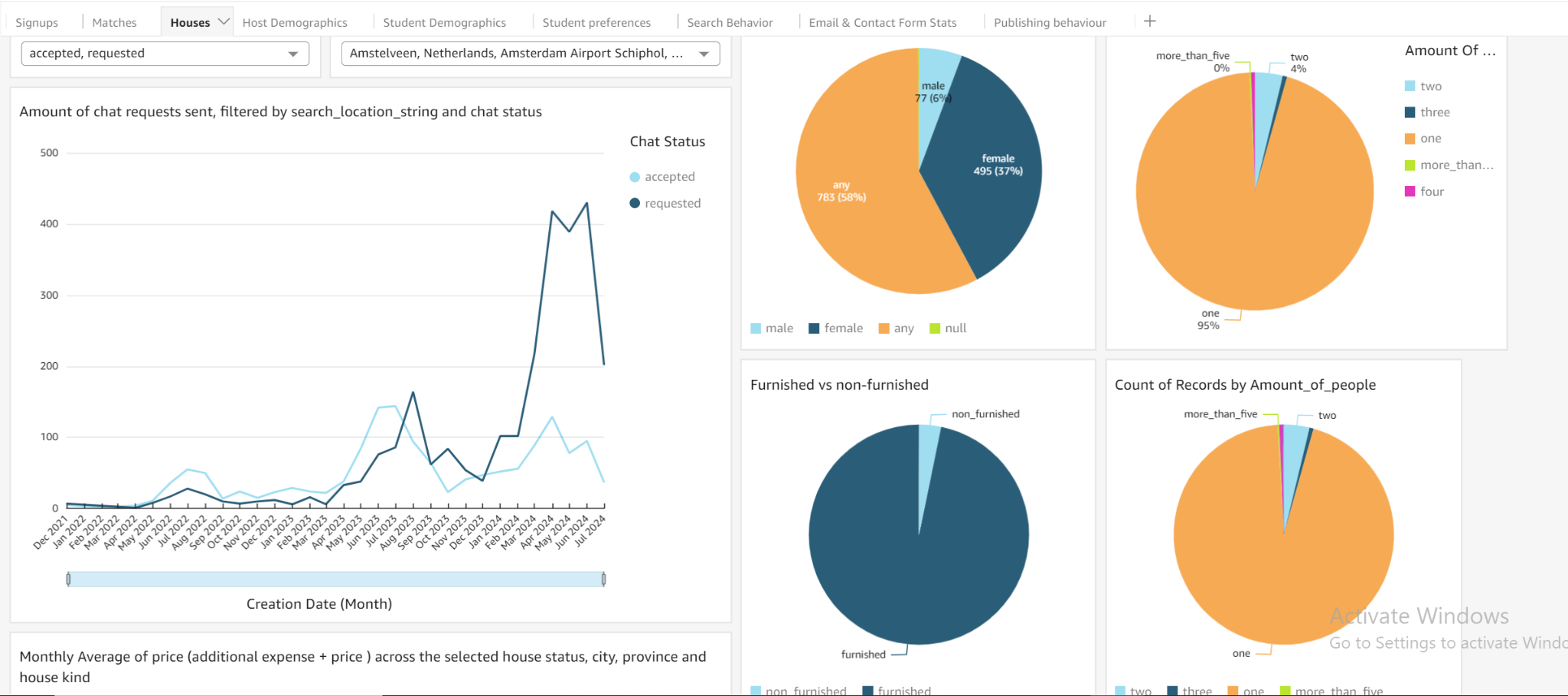


Figure 4.2 Graphical representation of Houses data in Hospi Housing