



TECHNICAL TEST DATA ANALYST - MNC MEDIA



Taqiyudin Muhammad Khalil

TABLE OF CONTENT

01 - ARCHITECTURE PLAN

This section outlines the design and structure of the system, detailing the components, data flow, and technologies involved in supporting the process.

02 - ETL/ELT PROCESS

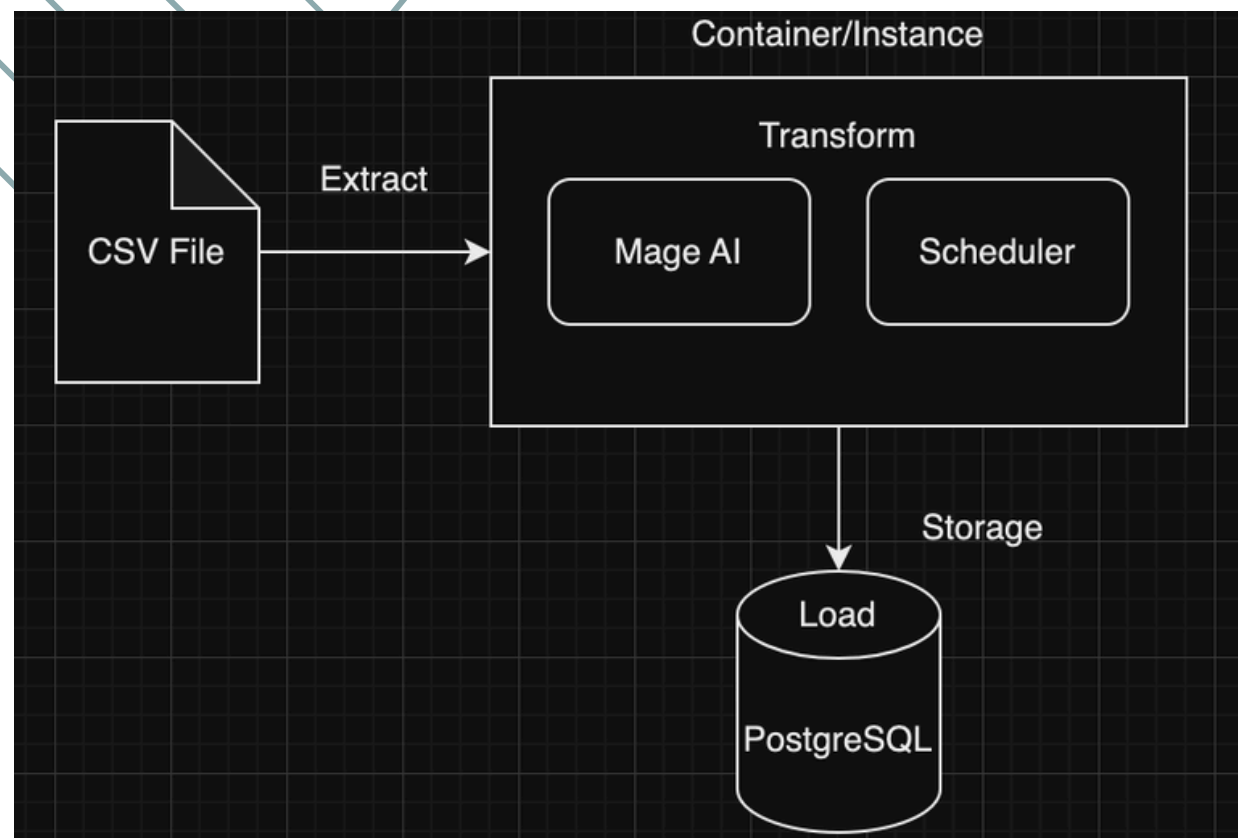
This section describes the procedures for ETL/ELT data from data sources to the target system, ensuring data quality and integration.

03 - RESULT

This section presents the outcomes and insights derived from the system's processing,



ARCHITECTURE PLAN



Components:

1. Docker-Compose: Containerized Mage AI and PostgreSQL services.
2. Mage AI: Orchestration tool to manage ETL/ELT pipelines.
3. PostgreSQL: Data storage.
4. CSV File: Input data for extraction.



ETL PROCESS EXTRACTION

Process:

- 1.Run Docker Compose
- 2.Upload the provided CSV file to the first pipeline in Mage AI.
- 3.Use Mage AI's connectors to read the dataset.
- 4.Extract data from the CSV and prepare it for transformation.

Tools Used: Mage AI, Docker container for Mage AI service.

```
PY DATA LOADER load_data_csv ← Edit parents
1 import io
2 import pandas as pd
3 import requests
4
5
6 @data_loader
7 def load_data_from_api(*args, **kwargs):
8     path = 'magic/data_loaders/dataset_user_behavior_for_test.csv'
9     return pd.read_csv(path, sep=',')
10
11
12 @test
13 def test_row_count(df, *args) → None:
14     assert len(df.index) ≥ 1000, 'The data does not have enough rows.'
15
```

ETL PROCESS

TRANSFORM

```
PY TRANSFORMER transform_data_csv ← 1 parent

33 def transform(df, *args, **kwargs):
34     # Add number of meals for each user
35     df_new_column = number_of_rows_per_key(df, 'Iduser', 'total_row')
36     df = df.join(df_new_column, on='Iduser')
37
38     # Clean column names
39     df.columns = [clean_column(col) for col in df.columns]
40     df = preprocess(df)
41     df['start_watching'] = pd.to_datetime(df['start_watching'], format='%m/%d/%Y %')
42
43     return df
44
45
46 @test
47 def test_number_of_columns(df, *args) → None:
48     assert len(df.columns) < 11, 'There needs to be at least 11 columns.'
49
1/1 tests passed.

OUTPUT 0
iduser  start_watching  device_id  province  city
```

Process:

1. Apply transformations like data cleaning, normalization, or aggregation.
2. Handle missing values, duplicate data, and ensure consistency.
3. Convert date format, standardize columns.

Tools Used: Mage AI (Python).

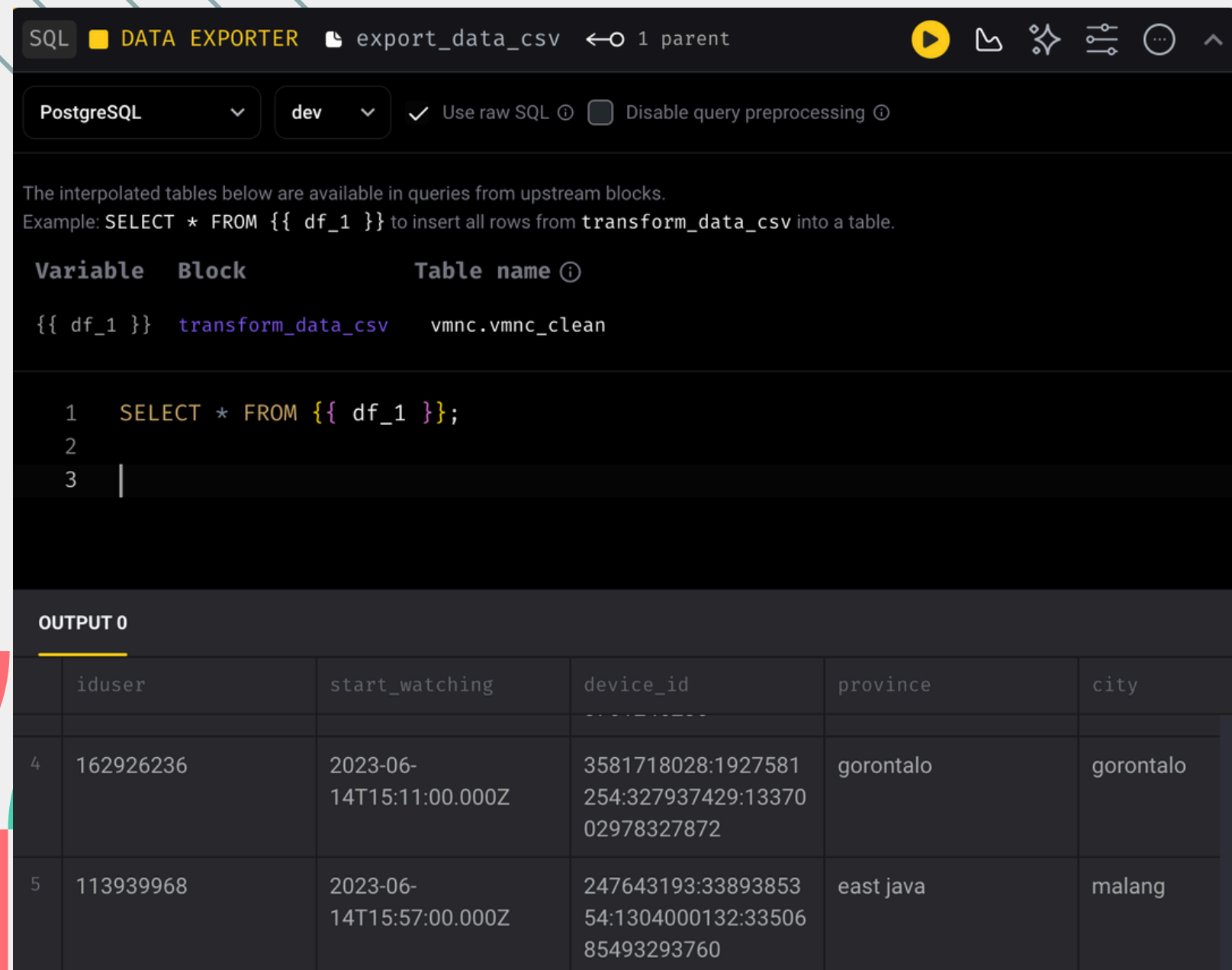
Quality Checks:

1. Missing Data: Ensure there are no missing or NULL values in critical columns.
2. Data Types: Validate that the data types are consistent (e.g., integers, dates).
3. Duplicates: Identify and remove any duplicate rows.

Validation Steps:

1. Verify data integrity before loading into PostgreSQL.
2. Validate data types and formats post-transformation.

ETL PROCESS LOAD



The screenshot shows the Mage AI interface for a PostgreSQL block named 'export_data_csv'. The query being executed is `SELECT * FROM {{ df_1 }};`. The output, labeled 'OUTPUT 0', is a table with 5 rows and 6 columns: `iduser`, `start_watching`, `device_id`, `province`, and `city`. The first row is highlighted in red.

	iduser	start_watching	device_id	province	city
4	162926236	2023-06-14T15:11:00.000Z	3581718028:1927581254:327937429:1337002978327872	gorontalo	gorontalo
5	113939968	2023-06-14T15:57:00.000Z	247643193:3389385354:1304000132:3350685493293760	east java	malang

Process:

1. Once the data is transformed, load it into PostgreSQL.
2. Use Mage AI's PostgreSQL connector to insert the cleaned data into the database.

Tools Used: PostgreSQL container (via Docker), Mage AI.

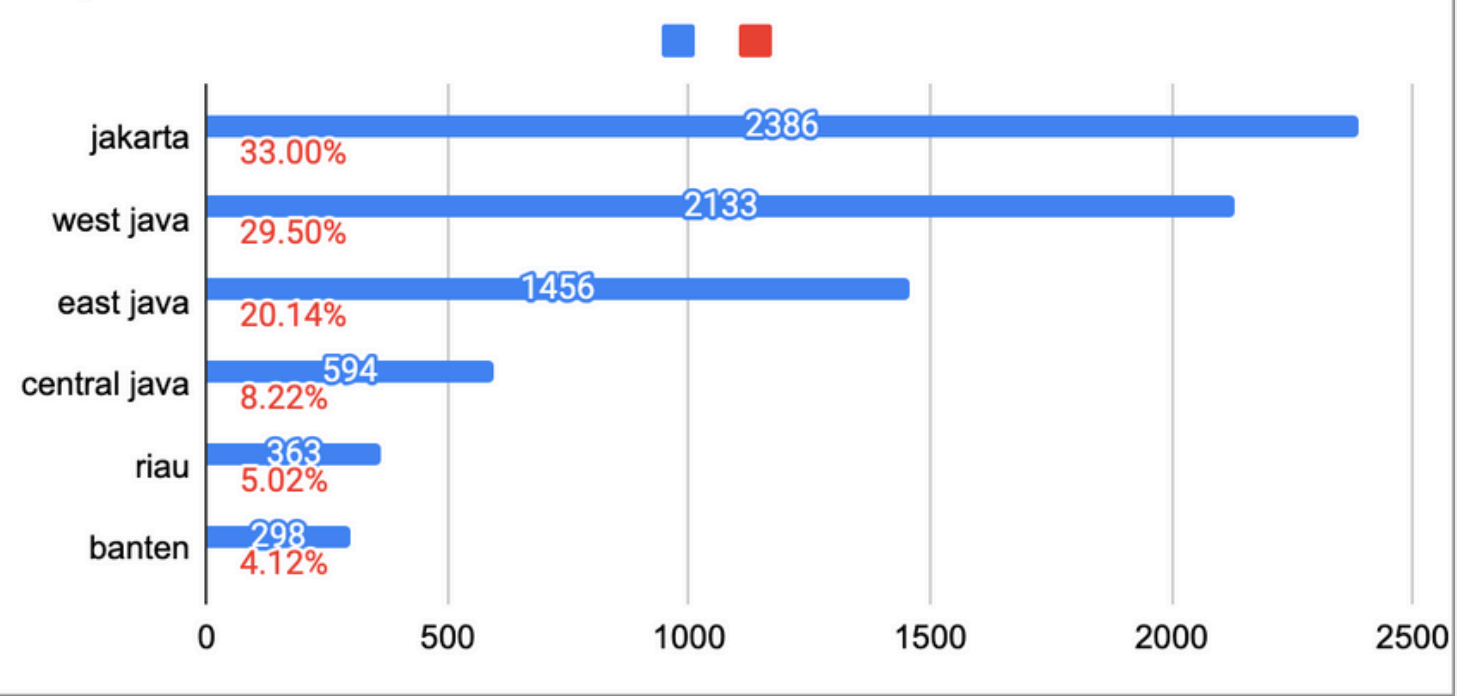
TABLE RESULT

```
SQL DATA LOADER load_data_contenttype Edit parents
PostgreSQL dev Use raw SQL Disable query preprocessing

1 DROP TABLE IF EXISTS vmnc.vmcnc_content_type;
2
3 CREATE TABLE vmnc.vmcnc_content_type (
4   province VARCHAR(255) PRIMARY KEY,
5   total_user INT
6 );
7
8
9 INSERT INTO vmnc.vmcnc_content_type
10 SELECT
11   content_type,
12   COUNT(DISTINCT iduser) AS total_user
13 FROM vmnc.vmcnc_clean
14 GROUP BY 1
15 ORDER BY 2 DESC;
```

province	total_user
jakarta	2386
west java	2133
east java	1456
central java	594
riau	363
banten	298

Top 5 Province

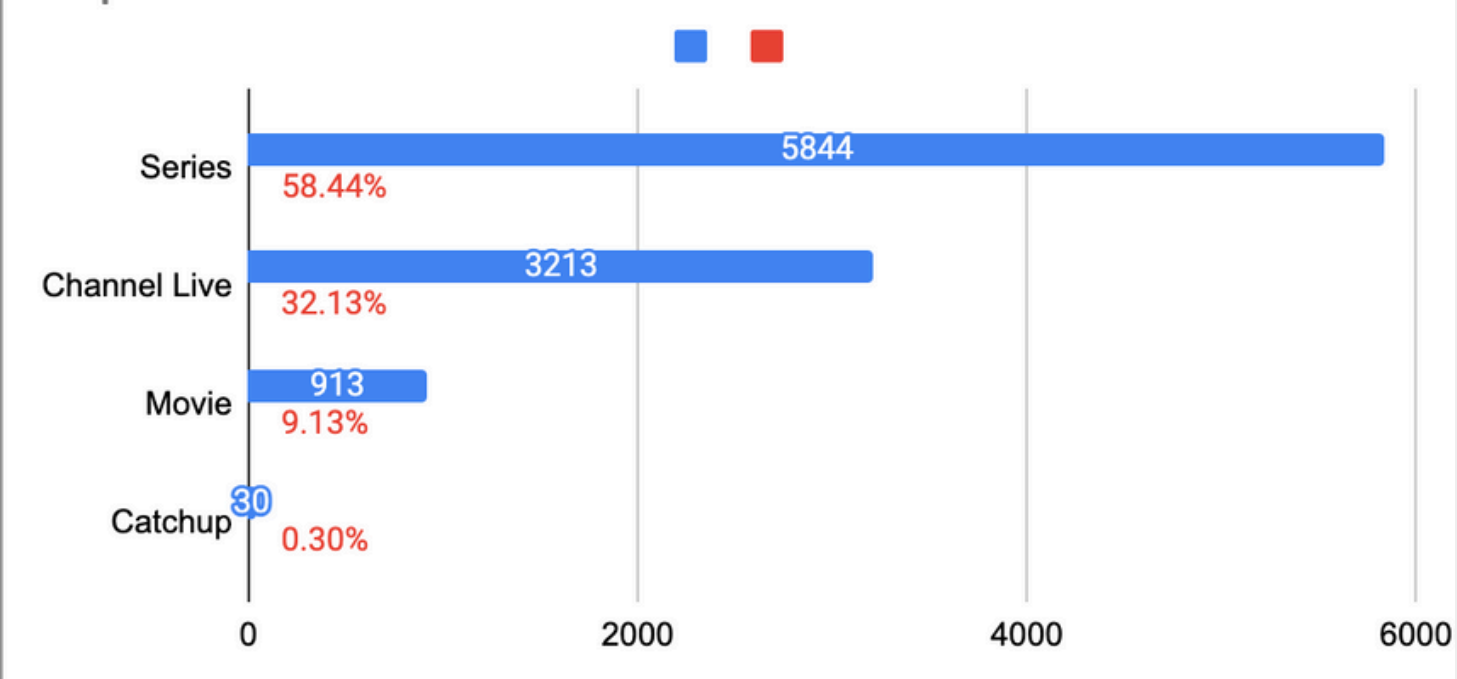


```
SQL DATA LOADER load_data_province Edit parents
PostgreSQL dev Use raw SQL Disable query preprocessing

1 DROP TABLE IF EXISTS vmnc.vmcnc_province;
2
3 CREATE TABLE vmnc.vmcnc_province (
4   province VARCHAR(255) PRIMARY KEY,
5   total_user INT
6 );
7
8
9 INSERT INTO vmnc.vmcnc_province
10 SELECT
11   province,
12   COUNT(DISTINCT iduser) AS total_user
13 FROM vmnc.vmcnc_clean
14 GROUP BY 1
15 ORDER BY 2 DESC;
```

province	total_user
jakarta	2386
west java	2133
east java	1456
central java	594
riau	363
banten	298

Top 4 Content



The top 5 provinces with the highest number of users are still dominated by Java Island, accounting for 94.98%, with Jakarta leading at 33%. Riau is the only province outside Java Island that ranks in the top 5. This presents a potential opportunity to increase users around that region.

The top content is still dominated by series, with 58.44%. Movie and catch-up content have user numbers below 10%. This needs to be evaluated, and innovations should be sought to boost users in these content categories.

The background features four decorative geometric patterns in the corners. The top-left corner has a series of parallel diagonal lines in a light blue-grey color. The top-right corner contains a cluster of overlapping semi-circles in yellow, red, teal, and dark blue. The bottom-left corner also features a cluster of overlapping semi-circles in red, teal, and dark blue. The bottom-right corner has a series of parallel diagonal lines in a light blue-grey color, mirroring the top-left pattern.

THANK YOU