Outbound Roaming Data Analysis and Processing

Neural Technologies Indonesia
Data Engineer Bootcamp

Presented by Reza Septian Kamajaya

About Company

Neural Technologies Indonesia (NTI), established in 2007, is a premier IT company specializing in telecommunications, healthcare, medical devices, and various other industries, including oil and gas. NTI offer an extensive range of services, from supplying equipment to providing comprehensive IT support across multiple sectors. Our commitment to excellence is reflected in our team of highly skilled engineers and experts, dedicated to addressing the unique challenges of each client.

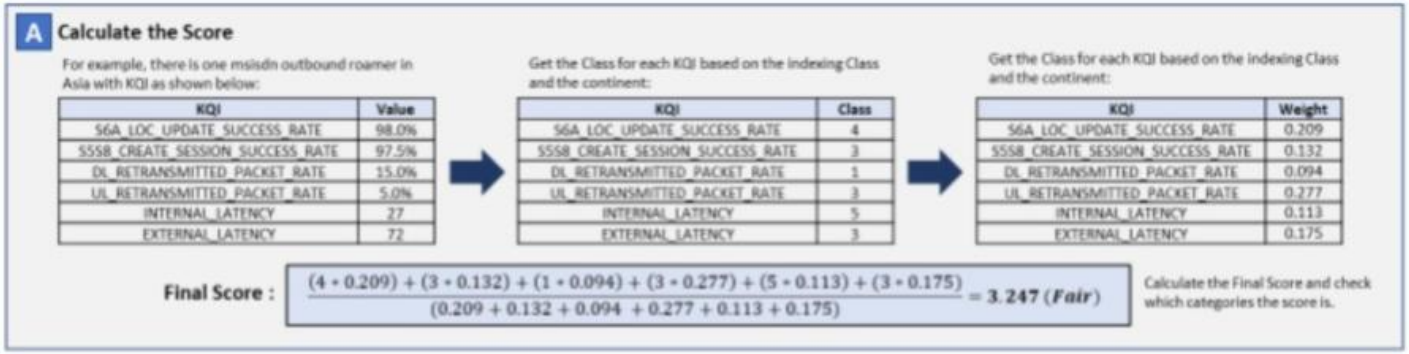
Objective

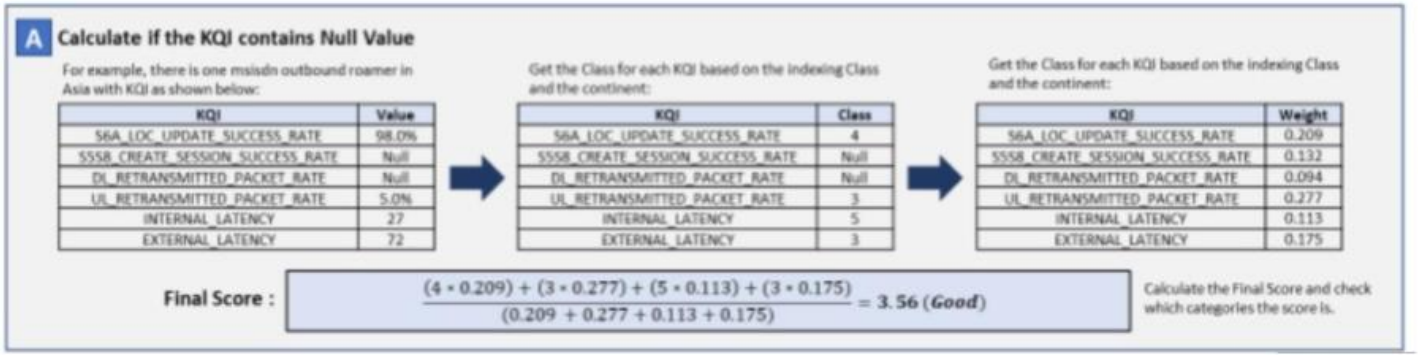
1. Analyze Raw Data
2. Insert raw data into database
3. Create Database function for :
    a) List all countries name
    b) List all operators name with optional params country name
    c) List all countries and operators, show country name, operator name, active user, traffic mb, create session success rate, loc update success rate, re-transmitted packet ul, re-transmitted packet dl, internal latency, externallatency. Country name and operator name optional (if value null then show ALL country / operator). Start Date and End Date mandatory parameter
    d) List countries with score and remark (scoring calculation in next slide). Start Date and End Date mandatory Parameter.
    e) List 5 worst operator based on score for each country. Start Date and End Date mandatory Parameter.

Scoring

Weight	Continent	KQI	1	2	3	4	5
0,209	America	S6A_LOC_UPDATE_SUCCESS_RATE	0 <= x <= 31	31 < x <= 69.14	69.14 < x <= 93.31	93.31 < x <= 99.37	99.37 < x <= 100
0,132	America	S5S8_CREATE_SESSION_SUCCESS_RATE	0 <= x <= 50	50 < x <= 66.667	66.667 < x <= 85.714	85.714 < x <= 96.97	96.97 < x <= 100
0,094	America	DL_RETRANSMITTED_PACKET_RATE	100 >= x >= 12.612	8.965 > x >= 12.612	8.965 > x >= 4.348	4.348 > x >= 2.297	2.297 > x >= 0
0,277	America	UL_RETRANSMITTED_PACKET_RATE	100 >= x >= 12.5	12.5 > x >= 7.688	7.688 > x >= 3.125	3.125 > x >= 0.784	0.784 > x >= 0
0,113	America	INTERNAL_LATENCY	∞ > x >= 769.27	769.27 > x >= 600.78	600.78 > x >= 463.5	463.5 > x >= 404	404 > x >= 0
0,175	America	EXTERNAL_LATENCY	∞ < x >= 408	408 > x >= 236.872	236.872 > x >= 97	97 > x >= 48.333	48.333 > x >= 0
0,209	Africa	S6A_LOC_UPDATE_SUCCESS_RATE	0 <= x <= 31	31 < x <= 69.14	69.14 < x <= 93.31	93.31 < x <= 99.37	99.37 < x <= 100
0,132	Africa	S5S8_CREATE_SESSION_SUCCESS_RATE	0 <= x <= 96.154	96.154 < x <= 75	75 < x <= 50	50 < x <= 20	20 < x <= 100
0,094	Africa	DL_RETRANSMITTED_PACKET_RATE	100 >= x >= 8.265	5.902 > x >= 8.265	5.902 > x >= 2.722	2.722 > x >= 1.025	1.025 > x >= 0
0,277	Africa	UL_RETRANSMITTED_PACKET_RATE	100 >= x >= 12.602	12.602 > x >= 7.15	7.15 > x >= 2.513	2.513 > x >= 0.532	0.532 > x >= 0
0,113	Africa	INTERNAL_LATENCY	∞ > x >= 847.371	847.371 > x >= 662.632	662.632 > x >= 488.667	488.667 > x >= 426.976	426.976 > x >= 0
0,175	Africa	EXTERNAL_LATENCY	∞ < x >= 386.279	386.279 > x >= 233.318	233.318 > x >= 94.278	94.278 > x >= 51	51 > x >= 0
0,209	Europe	S6A_LOC_UPDATE_SUCCESS_RATE	0 <= x <= 31	31 < x <= 69.14	69.14 < x <= 93.31	93.31 < x <= 99.37	99.37 < x <= 100
0,132	Europe	S5S8_CREATE_SESSION_SUCCESS_RATE	0 <= x <= 98.936	98.936 < x <= 97.802	97.802 < x <= 90	90 < x <= 40	40 < x <= 100
0,094	Europe	DL_RETRANSMITTED_PACKET_RATE	100 >= x >= 12.415	9.091 > x >= 12.415	9.091 > x >= 5.044	5.044 > x >= 2.823	2.823 > x >= 0
0,277	Europe	UL_RETRANSMITTED_PACKET_RATE	100 >= x >= 12.677	12.677 > x >= 7.321	7.321 > x >= 2.749	2.749 > x >= 0.632	0.632 > x >= 0
0,113	Europe	INTERNAL_LATENCY	∞ > x >= 602	602 > x >= 459	459 > x >= 355.667	355.667 > x >= 311	311 > x >= 0
0,175	Europe	EXTERNAL_LATENCY	∞ < x >= 280.5	280.5 > x >= 161.333	161.333 > x >= 62.667	62.667 > x >= 32	32 > x >= 0
0,209	Asia	S6A_LOC_UPDATE_SUCCESS_RATE	0 <= x <= 31	31 < x <= 69.14	69.14 < x <= 93.31	93.31 < x <= 99.37	99.37 < x <= 100
0,132	Asia	S5S8_CREATE_SESSION_SUCCESS_RATE	0 <= x <= 99.667	99.667 < x <= 99.308	99.308 < x <= 97.196	97.196 < x <= 60	60 < x <= 100
0,094	Asia	DL_RETRANSMITTED_PACKET_RATE	100 >= x >= 7.083	4.545 > x >= 7.083	4.545 > x >= 1.922	1.922 > x >= 0.799	0.799 > x >= 0
0,277	Asia	UL_RETRANSMITTED_PACKET_RATE	100 >= x >= 13.294	13.294 > x >= 6.928	6.928 > x >= 2.439	2.439 > x >= 0.704	0.704 > x >= 0
0,113	Asia	INTERNAL_LATENCY	∞ > x >= 414.5	414.5 > x >= 333	333 > x >= 267	267 > x >= 236.117	236.117 > x >= 0
0,175	Asia	EXTERNAL_LATENCY	∞ < x >= 327.667	327.667 > x >= 185.129	185.129 > x >= 66	66 > x >= 29	29 > x >= 0
0,209	Asean & Oceania	S6A_LOC_UPDATE_SUCCESS_RATE	0 <= x <= 31	31 < x <= 69.14	69.14 < x <= 93.31	93.31 < x <= 99.37	99.37 < x <= 100
0,132	Asean & Oceania	S5S8_CREATE_SESSION_SUCCESS_RATE	0 <= x <= 97.565	97.565 < x <= 75	75 < x <= 50	50 < x <= 20	20 < x <= 100
0,094	Asean & Oceania	DL_RETRANSMITTED_PACKET_RATE	100 >= x >= 5.882	3.763 > x >= 5.882	3.763 > x >= 1.581	1.581 > x >= 0.668	0.668 > x >= 0
0,277	Asean & Oceania	UL_RETRANSMITTED_PACKET_RATE	100 >= x >= 12.5	12.5 > x >= 6.667	6.667 > x >= 2.563	2.563 > x >= 0.841	0.841 > x >= 0
0,113	Asean & Oceania	INTERNAL_LATENCY	∞ > x >= 340.222	340.222 > x >= 249.286	249.286 > x >= 167.719	167.719 > x >= 138.4	138.4 > x >= 0
0,175	Asean & Oceania	EXTERNAL_LATENCY	∞ < x >= 213	213 > x >= 118	118 > x >= 45.667	45.667 > x >= 22.5	22.5 > x >= 0

Scoring





Remark:

5 Excellent

4 Good

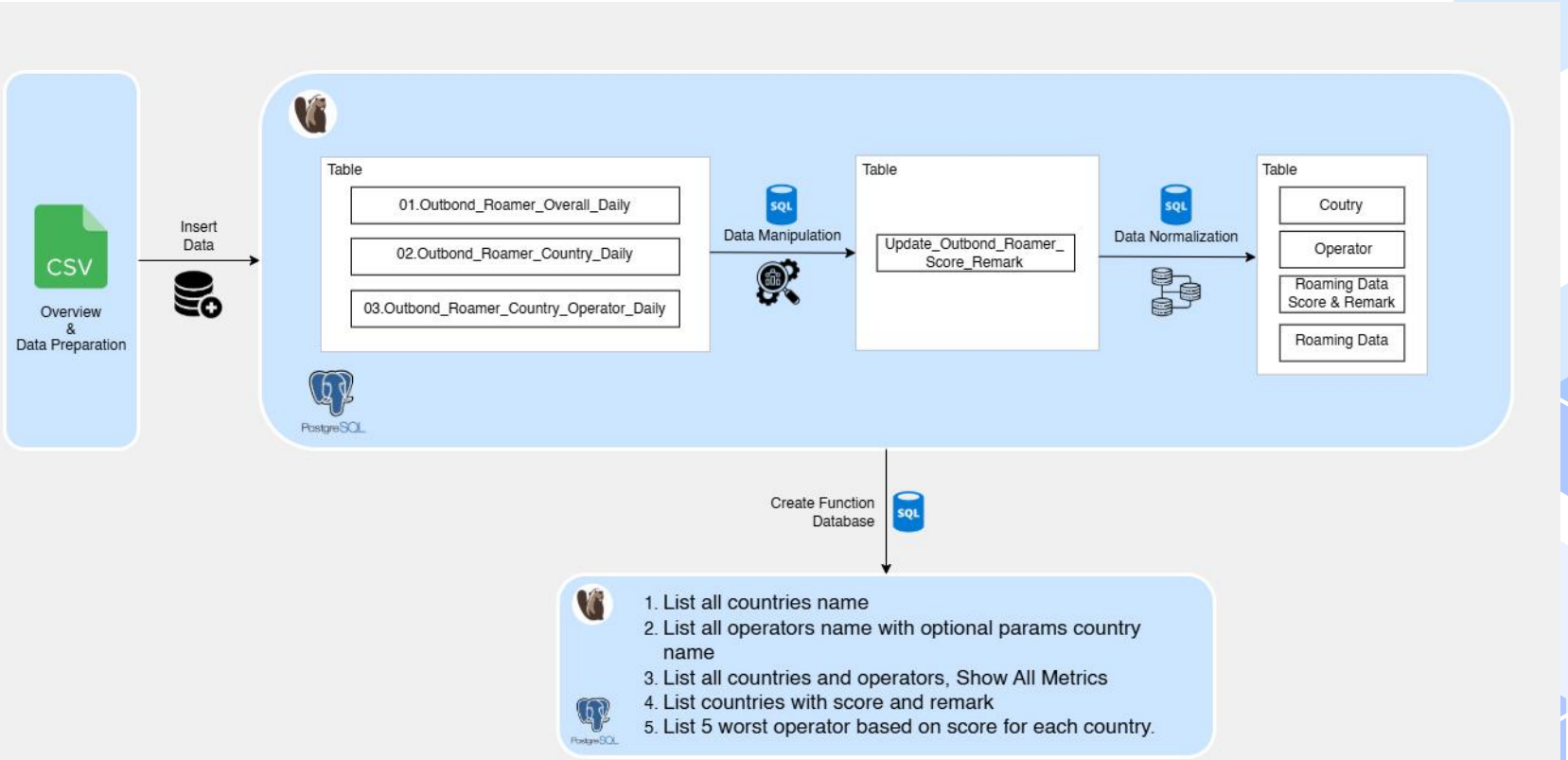
3 Fair

2 Poor

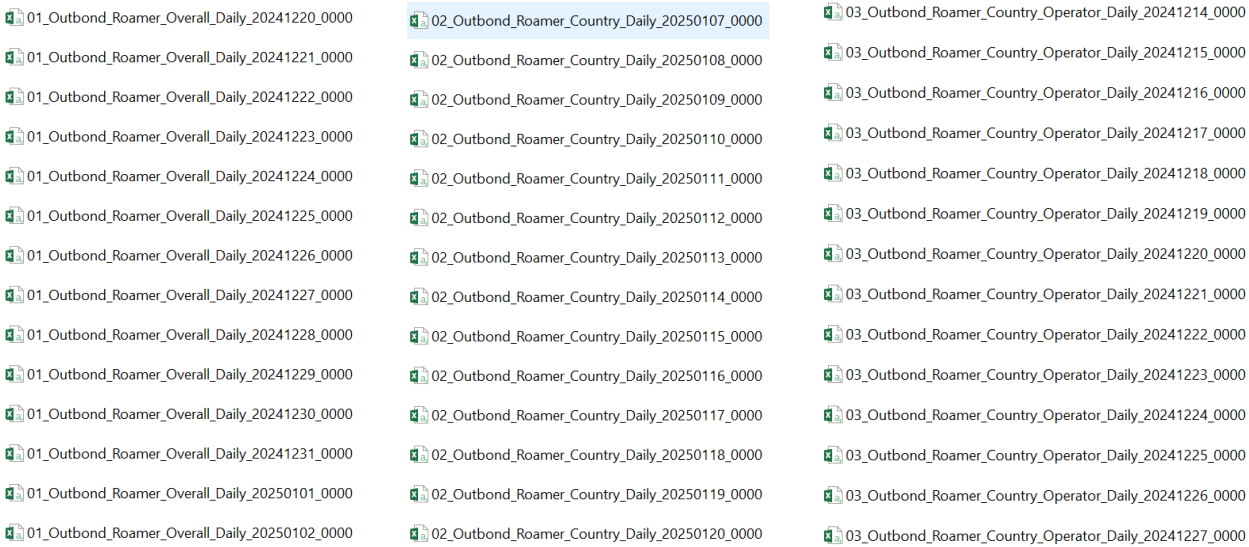
1 Bad

1. Overview & Data Preparation
2. Insert Data
3. Data Manipulation
4. Data Normalization
5. Create Function

Flow


CSV
Overview
&
Data Preparation
Insert
Data
Table
01.Outbond_Roamer_Overall_Daily
02.Outbond_Roamer_Country_Daily
03.Outbond_Roamer_Country_Operator_Daily
Data Manipulation
Table
Update_Outbond_Roamer_
Score_Remark
Data Normalization
Table
Coutry
Operator
Roaming Data
Score & Remark
Roaming Data
PostgreSQL
Create Function
Database
1. List all countries name
2. List all operators name with optional params country name
3. List all countries and operators, Show All Metrics
4. List countries with score and remark
5. List 5 worst operator based on score for each country.

Overview



Types of Daily Outbound Data Files

There are three types of daily outbound data files, categorized by the following codes:

01: Aggregated by country and operator

02: Aggregated by operator

03: Raw daily outbound data

Each file type consists of **52 files**, resulting in a **total of 156 files**.

Overview

File Header Information

Each file initially contained **10 headers**:

- datedd
- OPERATOR_NAME
- COUNTRY_NAME
- Numuser
- active_user
- Trafficmb
- Session
- duration_s
- s5s8_create_session_success_rate
- s6a_loc_update_success_rate

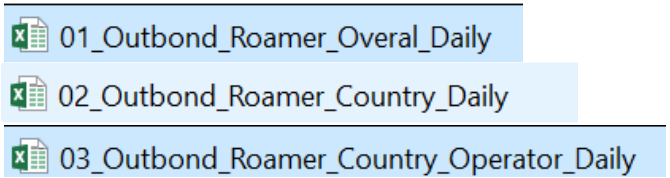
Starting from **2025-01-07**, **4 new headers** were added, bringing the total to **14 headers**:

- dl_retransmitted_packet_rate
- ul_retransmitted_packet_rate
- internal_latency
- external_latency

Data Preparation

s6a_loc_update_success_rate	dl_retransmitted_packet_rate	ul_retransmitted_packet_rate	internal_Latency	external_latency
99.91275378862888	NULL	NULL	NULL	NULL
99.74597798475868	NULL	NULL	NULL	NULL
99.6	NULL	NULL	NULL	NULL
100.0	NULL	NULL	NULL	NULL
100.0	NULL	NULL	NULL	NULL
100.0	NULL	NULL	NULL	NULL
100.0	NULL	NULL	NULL	NULL
100.0	NULL	NULL	NULL	NULL
100.0	NULL	NULL	NULL	NULL
100.0	NULL	NULL	NULL	NULL
100.0	NULL	NULL	NULL	NULL
100.0	NULL	NULL	NULL	NULL
100.0	NULL	NULL	NULL	NULL

Added **4 new columns** with **NULL values** to files that did not initially contain these columns.



Merged all files based on their respective prefixes (**01, 02, and 03**) into a single file for each category.

Insert Data

```sql
CREATE TABLE roaming_data."01_Outbond_Roamer_Overal_Daily" (
  datedd DATE
  , operator_name VARCHAR(30)
  , country_name VARCHAR(30)
  , numuser INT
  , active_user INT
  , trafficmb FLOAT
  , session BIGINT
  , duration_s FLOAT
  , s5s8_create_session_success_rate FLOAT
  , s6a_loc_update_success_rate FLOAT
  , dl_retransmitted_packet_rate FLOAT
  , ul_retransmitted_packet_rate FLOAT
  , internal_latency FLOAT
  , external_latency FLOAT
 );

CREATE TABLE roaming_data."02_Outbond_Roamer_Country_Daily" (
  datedd DATE
  , operator_name VARCHAR(30)
  , country_name VARCHAR(100)
  , numuser INT
  , active_user INT
  , trafficmb FLOAT
  , session BIGINT
  , duration_s FLOAT
  , s5s8_create_session_success_rate FLOAT
  , s6a_loc_update_success_rate FLOAT
  , dl_retransmitted_packet_rate FLOAT
  , ul_retransmitted_packet_rate FLOAT
  , internal_latency FLOAT
  , external_latency FLOAT
 );
```

```sql
CREATE TABLE roaming_data."03_Outbond_Roamer_Country_Operator_Daily" (
  datedd DATE
  , operator_name VARCHAR(100)
  , country_name VARCHAR(100)
  , numuser INT
  , active_user INT
  , trafficmb FLOAT
  , session BIGINT
  , duration_s FLOAT
  , s5s8_create_session_success_rate FLOAT
  , s6a_loc_update_success_rate FLOAT
  , dl_retransmitted_packet_rate FLOAT
  , ul_retransmitted_packet_rate FLOAT
  , internal_latency FLOAT
  , external_latency FLOAT
 );
```

Creating three tables to insert three files into the database.

Insert Data

```bash
#!/bin/bash

# Konfigurasi Database
DB_NAME="postgres"
DB_USER="postgres"
DB_HOST="localhost"
DB_PORT="5432"
SCHEMA_NAME="roaming_data"
export PGPASSWORD="1234"

# Menambahkan PostgreSQL ke PATH jika belum ada
if ! command -v psql &> /dev/null; then
    export PATH="/c/Program Files/PostgreSQL/17/bin:$PATH"
fi

# Cek apakah psql tersedia
if ! command -v psql &> /dev/null; then
    echo "❌ ERROR: psql tidak ditemukan! Pastikan PostgreSQL sudah terinstal."
    exit 1
fi

# Mapping tabel dan file CSV
declare -A TABLES
TABLES["01_Outbond_Roamer_Overal_Daily"]="01_Outbond_Roamer_Overal_Daily.csv"
TABLES["02_Outbond_Roamer_Country_Daily"]="02_Outbond_Roamer_Country_Daily.csv"
TABLES["03_Outbond_Roamer_Country_Operator_Daily"]="03_Outbond_Roamer_Country_Operator_Daily.csv"

CSV_DIR="C:\\NTI\\Final_Project"  # Gunakan path Windows

echo "🔄 Memulai proses import data..."

for TABLE_NAME in "${!TABLES[@]}"; do
    FILE_NAME="${TABLES[$TABLE_NAME]}"
    FILE_PATH="${CSV_DIR}\\${FILE_NAME}"  # Gunakan path Windows

    if [ ! -f "$(echo "$FILE_PATH" | sed 's/\\/\//g')" ]; then
        echo "❌ ERROR: File $FILE_PATH tidak ditemukan!"
        continue
    fi

    echo "🗑️ Mengosongkan tabel: $TABLE_NAME..."
    psql -h "$DB_HOST" -p "$DB_PORT" -U "$DB_USER" -d "$DB_NAME" -c "TRUNCATE TABLE \"$SCHEMA_NAME\".\"$TABLE_NAME\";"

    echo "📥 Mengimpor file $FILE_NAME ke tabel $TABLE_NAME..."
    psql -h "$DB_HOST" -p "$DB_PORT" -U "$DB_USER" -d "$DB_NAME" -c "\COPY \"$SCHEMA_NAME\".\"$TABLE_NAME\" FROM '$FILE_PATH' WITH DELIMITER '|' CSV HEADER;"

    if [ $? -eq 0 ]; then
        echo "✅ Berhasil mengimpor $FILE_NAME ke $TABLE_NAME!"
    else
        echo "❌ Gagal mengimpor $FILE_NAME ke $TABLE_NAME! Cek error_log.txt untuk detail."
    fi
done

unset PGPASSWORD
echo "🎉 Semua proses import selesai!"
```

Insert Data Process

This script automates importing **CSV files** into a **PostgreSQL database** by:

1.Configuring Database
– Sets connection details and PostgreSQL authentication.

2.Checking PostgreSQL (psql)
– Ensures psql is available and updates PATH if needed.

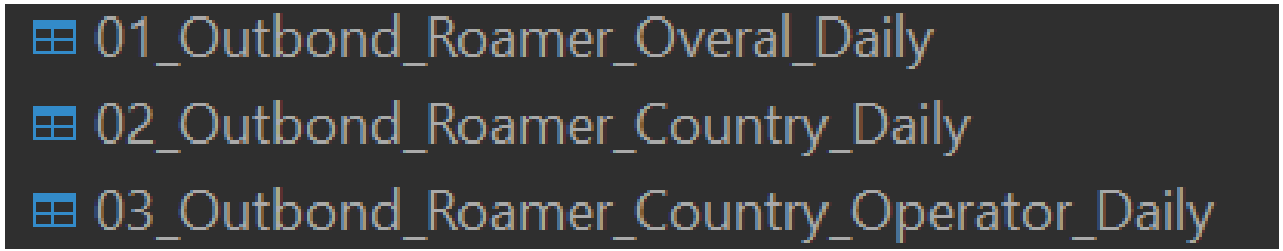
3.Mapping Files to Tables
– Defines table names and corresponding CSV files.

4.Importing Data –
- Checks if the file exists.
- Truncates the table before inserting new data.
- Uses \COPY to load CSV files into tables.
- Finalization – Unsets the password and confirms process completion.

This ensures efficient and automated data insertion with error handling.

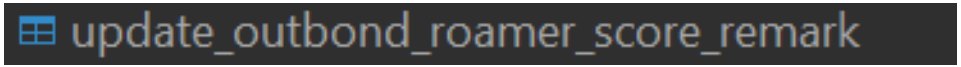
Insert Data

The data is imported into the PostgreSQL database using DBeaver as a third-party tool.
The import process generates three tables,
distinguished by codes **01, 02, & 03**, corresponding to the CSV file names:
01_Outbond_Roamer_Overal_Daily
02_Outbond_Roamer_Country_Daily
03_Outbond_Roamer_Country_Operator_Daily



Data Manipulation

```
CREATE TABLE update_outbond_roamer_score_remark AS
SELECT * FROM roaming_data."03_Outbond_Roamer_Country_Operator_Daily";
```



The new table update_outbond_roamer_score_remark
is created by selecting all data and columns from the
03_Outbond_Roamer_Country_Operator_Daily
table within the roaming_data schema. This table will be used for data manipulation,
including country_name, operator daily, and relevant metrics,
without modifying the original data in the source table.

Data Manipulation

datedd	operator_name	country_name
2024-12-13	Airtel (Kolkata)--India(404)	India (404)
2024-12-13	Airtel (Tamil Nadu)--India(404)	India (404)
2024-12-13	Airtel (Madhya Pradesh)--India(404)	India (404)
2024-12-13	Vodafone (Tamil Nadu)--India(404)	India (404)
2024-12-13	Airtel (Mumbai)--India(404)	India (404)
2024-12-13	AirTel (Chennai)--India(404)	India (404)
2024-12-13	Airtel (Kerala)--India(404)	India (404)
2024-12-13	Airtel (Himachal Pradesh)--India(404)	India (404)
2024-12-13	Airtel (Uttar Pradesh W)--India(404)	India (404)
2024-12-13	Airtel (Rajasthan),Hexacom--India(404)	India (404)

datedd	operator_name	country_name
2024-12-13	Airtel	India
2024-12-13	Airtel	India
2024-12-13	Airtel	India
2024-12-13	Airtel	India
2024-12-13	Airtel	India
2024-12-13	Airtel	India
2024-12-13	Airtel	India
2024-12-13	Airtel	India
2024-12-13	Airtel	India
2024-12-13	Airtel	India

Data Cleaning: Standardizing Country & Operator Names
Overview:
The data contained unnecessary characters in the operator_name and country_name columns, such as region details and numeric codes. To improve consistency and readability, these extra characters were removed.
Key Cleaning Steps:
✓ Removed region details and symbols from operator_name (e.g., "Airtel (Kolkata)--India(404)" → "Airtel").
✓ Removed numeric codes from country_name (e.g., "India (404)" → "India").
✓ Ensured uniform formatting for better analysis.
This cleanup process improves data accuracy and usability for reporting and insights.

Data Manipulation

operator_name	country_name	numuser	active_user	trafficmb	session	durat
Hutchison--Indonesia	Indonesia	1	1	948.1814012527	24	
Hutchison--Indonesia	Indonesia	1	1	332.7190713882	215	
Hutchison--Indonesia	Indonesia	2	1	10.9675960541	192	
Hutchison--Indonesia	Indonesia	2	2	742.3620977402	116	
Hutchison--Indonesia	Indonesia	2	2	376.907910347	300	
Hutchison--Indonesia	Indonesia	1	1	6,540.8964195251	1,460	495,674
Hutchison--Indonesia	Indonesia	1	1	43.9227514267	12	
Hutchison--Indonesia	Indonesia	2	2	56.0606527328	112	
Hutchison--Indonesia	Indonesia	2	2	1,245.9874582291	440	319,754
Hutchison--Indonesia	Indonesia	2	2	2,766.6797113419	804	
Hutchison--Indonesia	Indonesia	1	1	1,491.4274377823	687	
Hutchison--Indonesia	Indonesia	4	4	1,689.4779920578	1,139	198,074
Hutchison--Indonesia	Indonesia	3	3	889.9518203735	6,748	297,754
Indosat (IDNSL)--Indo	Indonesia	1	1	2,805.5269899368	1,365	
Hutchison--Indonesia	Indonesia	2	2	4,287.8108873367	4,665	373,606
Hutchison--Indonesia	Indonesia	2	2	228.7299690247	466	
Hutchison--Indonesia	Indonesia	1	1	322.1853485107	696	

datedd	operator_name	country_name	numuser	active_user	trafficmb	session

```
DELETE FROM roaming_data.update_outbond_roamer_score_remark
WHERE country_name = 'Indonesia';
```

Data Cleaning: Removing Indonesia Data

Rows with **country_name = Indonesia** were removed as they represent **outbound roamers**, which are irrelevant for analysis. This ensures cleaner data, improves accuracy, and prevents misleading insights.

Data Manipulation

```sql
ALTER TABLE roaming_data.update_outbond_roamer_score_remark
ADD COLUMN weight1 FLOAT,
ADD COLUMN weight2 FLOAT,
ADD COLUMN weight3 FLOAT,
ADD COLUMN weight4 FLOAT,
ADD COLUMN weight5 FLOAT,
ADD COLUMN weight6 FLOAT,
ADD COLUMN final_score FLOAT,
ADD COLUMN remark TEXT;
```

Adding & Updating Columns for Scoring
Added New Columns:

- continent (region classification)
- weight1 - weight6 (scoring factors)
- final_score (computed score)
- remark (notes/comments)

Updated Metrick Column:
Categorized into **5 score levels (1-5)**.
Applied only to records in **specific continents**.

```sql
UPDATE roaming_data.update_outbond_roamer_score_remark
SET score_s6a_loc_update_success_rate =
    CASE
        WHEN score_s6a_loc_update_success_rate BETWEEN 0 AND 31 THEN 1
        WHEN score_s6a_loc_update_success_rate > 31 AND score_s6a_loc_update_success_rate <= 69.14 THEN 2
        WHEN score_s6a_loc_update_success_rate > 69.14 AND score_s6a_loc_update_success_rate <= 93.31 THEN 3
        WHEN score_s6a_loc_update_success_rate > 93.31 and score_s6a_loc_update_success_rate <= 99.37 THEN 4
        WHEN score_s6a_loc_update_success_rate > 99.37 AND score_s6a_loc_update_success_rate <= 100 THEN 5
    END
WHERE continent IN ('America', 'Africa', 'Europe', 'Asia', 'Asean & Oceania');
```

Data Manipulation

Filling the *continent* Column The continent column is updated based on country_name, grouping countries into specific continents such as America, Africa, Europe, Asia, Asean & Oceania, or Unknown if not Llisted

```
UPDATE roaming_data.update_outbond_roamer_score_remark
SET continent =
    CASE
        WHEN country_name IN ('Argentina', 'Bolivia', 'Brazil', 'Canada', 'Chile', 'Colombia', 'Ecuador',
                              'Mexico', 'Peru', 'Puerto Rico', 'United States of America', 'Uruguay', 'Turks & Caicos',
                              'Aruba', 'Jamaica') --15
        THEN 'America'

        WHEN country_name IN ('Algeria', 'Egypt', 'Ethiopia', 'Gambia', 'Kenya', 'Mauritius', 'Morocco',
                              'Seychelles', 'Sierra Leone', 'South Africa', 'Tanzania', 'Tunisia') --12
        THEN 'Africa'

        WHEN country_name IN ('Albania', 'Austria', 'Belarus', 'Belgium', 'Bosnia and Herzegovina', 'Bulgaria',
                              'Croatia', 'Cyprus', 'Czech Republic', 'Denmark', 'Estonia', 'Finland',
                              'Germany', 'Greece', 'Hungary', 'Iceland', 'Ireland', 'Italy', 'Latvia',
                              'Liechtenstein', 'Lithuania', 'Luxembourg', 'Malta', 'Moldova', 'Netherlands',
                              'Norway', 'Poland', 'Portugal', 'Romania', 'Russia', 'Serbia', 'Slovak Republic',
                              'Slovenia', 'Spain', 'Sweden', 'Switzerland', 'Ukraine', 'United Kingdom', 'Georgia') --39
        THEN 'Europe'

        WHEN country_name IN ('Afghanistan', 'Armenia', 'Azerbaijan', 'Bahrain', 'Bangladesh', 'China',
                              'Georgia', 'Hongkong', 'India', 'Iraq', 'Israel', 'India', 'Japan', 'Jordan', 'Kazakhstan',
                              'Kuwait', 'Kyrgyz Republic', 'Lebanon', 'Macau', 'Maldives', 'Mongolia', 'Nepal',
                              'Oman', 'Pakistan', 'Qatar', 'Saudi Arabia', 'South Korea', 'Sri Lanka', 'Tajikstan',
                              'Taiwan', 'Turkey', 'United Arab Emirates', 'Uzbekistan') --33
        THEN 'Asia'

        WHEN country_name IN ('Brunei', 'Cambodia', 'East Timor', 'Laos', 'Malaysia',
                              'Philippines', 'Singapore', 'Thailand', 'Vietnam', 'Australia', 'Fiji',
                               'New Caledonia', 'New Zealand', 'Papua New Guinea') --14
        THEN 'Asean & Oceania'

        ELSE 'Unknown'
    END;
```

Filling the weight1 – weight6 Columns
Each weight1 to weight6 column is updated with predefined numerical values for further analysis.

```
UPDATE roaming_data.update_outbond_roamer_score_remark
SET weight1 = 0.209;

UPDATE roaming_data.update_outbond_roamer_score_remark
SET weight2 = 0.132;

UPDATE roaming_data.update_outbond_roamer_score_remark
SET weight3 = 0.094;

UPDATE roaming_data.update_outbond_roamer_score_remark
SET weight4 = 0.277;

UPDATE roaming_data.update_outbond_roamer_score_remark
SET weight5 = 0.113;

UPDATE roaming_data.update_outbond_roamer_score_remark
SET weight6 = 0.175;
```

Data Manipulation

Determining the Final Score

The **final_score** column is calculated using a **weighted average** formula. Each score (S1-S6) is multiplied by its corresponding weight (W1-W6), and the sum is divided by the total weight:

```
UPDATE roaming_data.update_outbond_roamer_score_remark
SET final_score =
    (
        (CASE WHEN score_s5s8_create_session_success_rate IS NOT NULL AND weight1 IS NOT NULL
              THEN score_s5s8_create_session_success_rate * weight1 ELSE 0 END) +
        (CASE WHEN score_s6a_loc_update_success_rate IS NOT NULL AND weight2 IS NOT NULL
              THEN score_s6a_loc_update_success_rate * weight2 ELSE 0 END) +
        (CASE WHEN score_dl_retransmitted_packet_rate IS NOT NULL AND weight3 IS NOT NULL
              THEN score_dl_retransmitted_packet_rate * weight3 ELSE 0 END) +
        (CASE WHEN score_ul_retransmitted_packet_rate IS NOT NULL AND weight4 IS NOT NULL
              THEN score_ul_retransmitted_packet_rate * weight4 ELSE 0 END) +
        (CASE WHEN score_internal_latency IS NOT NULL AND weight5 IS NOT NULL
              THEN score_internal_latency * weight5 ELSE 0 END) +
        (CASE WHEN score_external_latency IS NOT NULL AND weight6 IS NOT NULL
              THEN score_external_latency * weight6 ELSE 0 END)
    ) /
    NULLIF(
        (CASE WHEN weight1 IS NOT NULL AND score_s5s8_create_session_success_rate IS NOT NULL THEN weight1 ELSE 0 END) +
        (CASE WHEN weight2 IS NOT NULL AND score_s6a_loc_update_success_rate IS NOT NULL THEN weight2 ELSE 0 END) +
        (CASE WHEN weight3 IS NOT NULL AND score_dl_retransmitted_packet_rate IS NOT NULL THEN weight3 ELSE 0 END) +
        (CASE WHEN weight4 IS NOT NULL AND score_ul_retransmitted_packet_rate IS NOT NULL THEN weight4 ELSE 0 END) +
        (CASE WHEN weight5 IS NOT NULL AND score_internal_latency IS NOT NULL THEN weight5 ELSE 0 END) +
        (CASE WHEN weight6 IS NOT NULL AND score_external_latency IS NOT NULL THEN weight6 ELSE 0 END),
    0);
```

$$\text{final\_score} = \frac{(S_1 \times W_1) + (S_2 \times W_2) + \ldots + (S_6 \times W_6)}{W_1 + W_2 + \ldots + W_6}$$

- If a **score or weight is NULL**, it is treated as **0**.
- If the **total weight is 0**, the final score is set to **NULL** to avoid division errors.

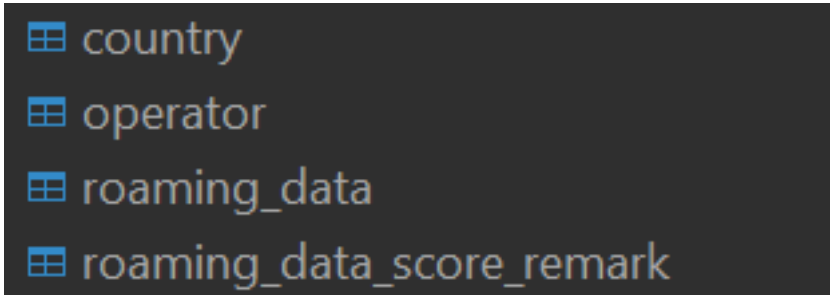
Data Manipulation

Assigning Remarks
Based on Final ScoreThe remark column is determined by categorizing final_score into predefined quality levels.
Each score range is assigned a descriptive label:
0 - 1 → **Bad**
1 - 2 → **Poor**
2 - 3 → **Fair**
3 - 4 → **Good**
4 - 5 → **Excellent**

This classification helps in interpreting roaming performance more intuitively.

```
UPDATE roaming_data.update_outbond_roamer_score_remark
SET remark =
    CASE
        WHEN final_score BETWEEN 0 AND 1 THEN 'Bad'
        WHEN final_score > 1 AND final_score <= 2 THEN 'Poor'
        WHEN final_score > 2 AND final_score <= 3 THEN 'Fair'
        WHEN final_score > 3 AND final_score <= 4 THEN 'Good'
        WHEN final_score > 4 AND final_score <= 5 THEN 'Excellent'
    end;
```

Data Normalization



After data manipulation, **normalization** is applied by splitting the data into four tables to improve efficiency and consistency:
Country – Contains information about countries and continents to avoid data duplication.
Operator – Stores a list of operators across all countries for better structuring.
Roaming Data – Holds the main data related to roaming usage, such as the number of users and data traffic.
Roaming Data Score Remark – Stores scores and additional remarks related to roaming data for further analysis.
This normalization ensures better data organization, reduces redundancy, and enhances the accuracy and efficiency of analysis.

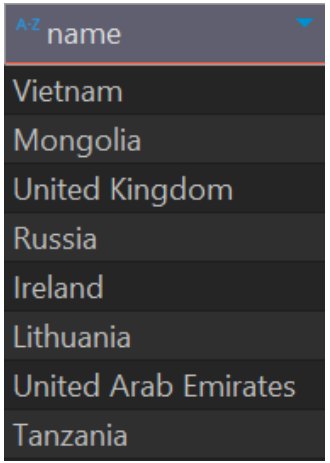
Create Function

Function – get_all_countries

Function Purpose : Retrieves a list of countries from the country table.
Can return all countries or filter by a specific country name.

```sql
---Function get all countries---
CREATE OR REPLACE FUNCTION get_all_countries(p_country_name VARCHAR(30) DEFAULT NULL)
RETURNS TABLE(name VARCHAR(30)) AS $$
BEGIN
    RETURN QUERY
    SELECT c.country_name
    FROM roaming_data.country c
    WHERE p_country_name IS NULL OR c.country_name = p_country_name;
END;
$$ LANGUAGE plpgsql;

--Memanggil function--
SELECT * FROM get_all_countries();
SELECT * FROM get_all_countries('India');
```



Create Function

Function – get_all_operators

Function Purpose : Retrieves a list of operator from the operator table.
Can return all operator or filter by a specific country name.

```sql
---Function get_all_operators---
CREATE OR REPLACE FUNCTION get_all_operators(p_country_name VARCHAR DEFAULT NULL)
RETURNS TABLE(operator_name VARCHAR, country_name VARCHAR) AS
$$
BEGIN
    RETURN QUERY
    SELECT o.operator_name, o.country_name
    FROM roaming_data.operator o
    WHERE p_country_name IS NULL OR o.country_name = p_country_name;
END;
$$ LANGUAGE plpgsql;

--Memanggil fungsi
SELECT * FROM get_all_operators();
SELECT * FROM get_all_operators('India');
```

operator_name	country_name
Ncell Spice	Nepal
MegaFon	Russia
NTT Docomo	Japan
Viettel Vietnam	Vietnam
Cable & Wireless	Seychelles
Mobitel	Georgia
HT Mobile	Bosnia and Herzegovina
Vodafone Czech Republic	Czech Republic
Digicel Fiji	Fiji
Elisa	Finland
Al-Wataniya	Kuwait
Orange	Austria

Create Function

Function – get_country_operator_metrics

This function, get_country_operator_metrics, retrieves roaming data metrics for a specified date range, country, and operator. It filters data from roaming_data based on the given parameters, returning key performance indicators such as active users, traffic, success rates, and latency.

```
---Calling the Function
--Without Country & Operator Filters (All data within a specified date range)
```

date	country_name	operator_name	act	trafficmb	s5s8_create_s	s6a_loc_up
024-12-16	Japan	KDDI	264	293,129.9570331574	[NULL]	99.9943883277
024-12-15	Portugal	Optimus	21	21,992.4435758591	79.7760887428	80.95703125
024-12-15	Gambia	Africell	0	[NULL]	0.073800738	[NULL]

```
--Filter by Country (Example: India)
SELECT * FROM get_country_operator_metrics('2024-12-13', '2025-01-30', 'India');
```

date	country_name	operator_name	act	trafficmb	s5s8_create_s	s6a_loc_up
024-12-22	India	Airtel	32	46,537.4113912582	91.7447728911	100
024-12-22	India	Airtel	251	396,995.6897754669	84.190844715	100
024-12-22	India	Airtel	1	1.0255775452	63.3333333333	100

```
--Filter by Operator (Example: Orange)
SELECT * FROM get_country_operator_metrics('2024-12-13', '2025-01-30', NULL, 'Orange');
```

country_name	operator_name	act	trafficmb	s5s8_create_s	s6a_loc_up
Liechtenstein	Orange	2	2.520015	0.0231213873	100
Austria	Orange	19	648.313538	0.4523475277	100
Switzerland	Orange	62	3,368.347307	0.1235069226	99.9759217457

Create Function

Function – get_countries_with_score

	datedd	country_name	avg_score	remark
73	2024-12-13	Spain	2.5483870968	Fair
74	2024-12-13	Russia	2.5483870968	Fair
75	2024-12-13	Tanzania	2.5483870968	Fair
76	2024-12-13	Israel	2.5483870968	Fair
77	2024-12-13	Afghanistan	2.5483870968	Fair
78	2024-12-13	Vietnam	2.5483870968	Fair
79	2024-12-13	Switzerland	2.5483870968	Fair
80	2024-12-13	Egypt	2.5483870968	Fair
81	2024-12-13	Bahrain	2.5483870968	Fair
82	2024-12-13	India	3.1612903226	Good
83	2024-12-13	Denmark	3.1612903226	Good
84	2024-12-13	Ecuador	3.1612903226	Good
85	2024-12-13	Thailand	3.1612903226	Good
86	2024-12-13	China	3.3655913978	Good
87	2024-12-13	United States of America	3.4435483871	Good
88	2024-12-13	Brazil	3.4677419355	Good
89	2024-12-13	Japan	3.7741935484	Good
90	2024-12-13	Argentina	3.7741935484	Good
91	2024-12-13	Uruguay	3.7741935484	Good
92	2024-12-13	Pakistan	3.7741935484	Good
93	2024-12-13	Peru	3.7741935484	Good
94	2024-12-13	Chile	3.7741935484	Good

The get_countries_with_score function calculates the average final_score for each country within a specified date range and categorizes it into remarks such as Bad, Poor, Fair, Good, or Excellent. The data is filtered by date, grouped by country, and sorted based on the average score.

```
--Calling the Function
--Displaying all countries within a specified date range
SELECT * FROM get_countries_with_score('2024-12-13', '2025-01-31');
```

Create Function

Function get_get_worst_operators

	datedd	country_name	operator_name	final_score	remark
1	2024-12-13	Afghanistan	AWCC	2.5483870968	Fair
2	2024-12-13	Albania	Vodafone	2.5483870968	Fair
3	2024-12-13	Algeria	ATM Mobilis	2.5483870968	Fair
4	2024-12-13	Argentina	Movistar	3.7741935484	Good
5	2024-12-13	Argentina	Telecom Personal SA	3.7741935484	Good
6	2024-12-13	Aruba	New Millenium Telecor	5	Excellent
7	2024-12-13	Australia	Optus	2.1612903226	Fair
8	2024-12-13	Australia	Telstra	2.5483870968	Fair
9	2024-12-13	Austria	Orange	2.5483870968	Fair
10	2024-12-13	Austria	T-Mobile	2.5483870968	Fair
11	2024-12-13	Austria	H3G	2.5483870968	Fair
12	2024-12-13	Austria	A1 Austria Telekom	2.5483870968	Fair
13	2024-12-13	Azerbaijan	Bakcell	2.5483870968	Fair

This function retrieves the five worst operators per country for each date based on their final_score within a given date range. It ranks operators per country and date using the RANK() function and returns the lowest-ranked ones, ordering the results by date, country, and score.

```
--Calling the Function
--Displaying all countries within a specified date range
SELECT * FROM get_countries_with_score('2024-12-13', '2025-01-31');
```

Thank You