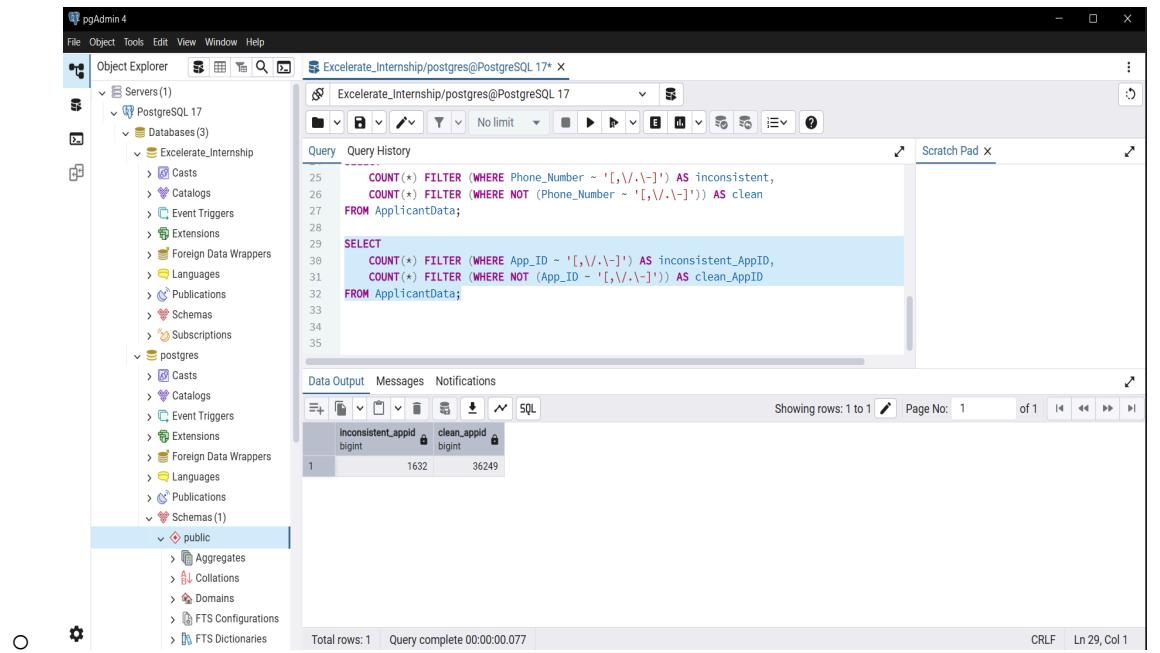


Week_1 Report_SubTeam12

1. Exploratory Data Analysis:

• Applicant Data:

- Applicant Data dataset had 37883 rows with 4 columns.
- The column names are App_ID, Country, University, Phone_Number. Every column of this file was in general format.



The screenshot shows the pgAdmin 4 interface. On the left is the Object Explorer tree, which includes a 'Servers' node, a 'PostgreSQL 17' node under it, and a 'Databases' node. Under 'Databases', there are three entries: 'Excelerate_Internship', 'postgres', and 'public'. The 'public' schema is currently selected. On the right is the main workspace, divided into several panes. The top pane is the 'Query History' pane, showing a complex SQL query. Below it is the 'Query' pane, also containing the same query. To the right of the query panes is the 'Scratch Pad' pane. At the bottom of the workspace is the 'Data Output' tab, which displays the results of the query. The results are presented in a table with two columns: 'inconsistent_appid' and 'clean_appid'. There is one row with the values 1632 and 36249 respectively. The status bar at the bottom of the pgAdmin window indicates 'Total rows: 1' and 'Query complete 00:00:00.077'.

inconsistent_appid	clean_appid
1632	36249

The screenshot shows the pgAdmin 4 interface. The left sidebar is the Object Explorer, displaying the database structure. The main area is a query editor window titled 'Excelerate_Internship/postgres@PostgreSQL 17*'. The query is:

```
23 Select Count(*) As Total_Duplicates
24 From (
25   SELECT App_ID, COUNT(*) AS occurrences
26   FROM ApplicantData
27   GROUP BY App_ID
28   HAVING COUNT(*) > 1);
```

The results show a single row with the value 7319.

	total_duplicates
1	7319

At the bottom, it says 'Total rows: 1' and 'Query complete 00:00:00.080'.

- From here we can see, App_ID has 1632 inconsistent data with 7319 duplicates which is polluting the data integrity. Also 36249 clean data.

The screenshot shows the pgAdmin 4 interface. On the left is the Object Explorer, displaying a tree structure of servers, databases, and schema objects. In the center is the Query Editor window, which contains a SQL query. The Data Output tab below the editor shows the results of the query.

```

18 COUNT(*) FILTER (WHERE App_ID IS NULL OR TRIM(App_ID) = '') AS app_id_nulls,
19 COUNT(*) FILTER (WHERE Country IS NULL OR TRIM(Country) = '') AS country_nulls,
20 COUNT(*) FILTER (WHERE University IS NULL OR TRIM(University) = '') AS university_nulls
21 COUNT(*) FILTER (WHERE Phone_Number IS NULL OR TRIM(Phone_Number) = '') AS phone_number
22 FROM ApplicantData;
23
24 SELECT
25 COUNT(*) FILTER (WHERE Phone_Number ~ '[,/.\\-]') AS inconsistent,
26 COUNT(*) FILTER (WHERE Length(Phone_Number) > 20) AS Invalid_PhoneNumber,
27 COUNT(*) FILTER (WHERE NOT (Phone_Number ~ '[,/.\\-]')) AS clean
28
29

```

	inconsistent	invalid_phonenumber	clean
1	11	3	37871

Total rows: 1 Query complete 00:00:00.077

- Here we can see, 11 inconsistent phone_number with 3 invalid phone numbers. Moreover, it has 37871 clean data.
- University column has only 1 name which is “Illinois Institute of Technology”. So it is evident that the data is of this university or related to this university.
- Top 10 most countries people applied from are India, Ghana, Nigeria, Pakistan, Iran, Bangladesh, Nepal, Ethiopia, Nigeria, South Africa.

The screenshot shows the pgAdmin 4 interface. On the left is the Object Explorer tree, which includes a 'Servers' node, a 'PostgreSQL 17' node, and a 'Databases' node containing 'Excelerate_Internship'. Inside 'Excelerate_Internship', there are several schema nodes like 'casts', 'catalogs', 'event triggers', etc., and a 'public' schema node which is currently selected. The main area contains a 'Query History' tab with a SQL query and a 'Data Output' tab showing the results of the query.

```

36 SELECT
37     Country,
38     COUNT(*) AS total_appeared
39 FROM ApplicantData
40 WHERE Country IS NOT NULL
41     AND TRIM(Country) <> ''
42     AND TRIM(Country) <> '-'
43 GROUP BY Country
44 ORDER BY total_appeared DESC

```

country	total_appeared
India	12234
Ghana	3773
Nigeria	3564
Pakistan	1502
Iran	1371
Bangladesh	997
Nepal	632
Ethiopia	539
nigeria	514
South Africa	503

● Campaign Data:

- Applicant Data dataset had 24 rows with 7 columns.
- The column names are ID, Name, Category, Intake, University, Status, and Start_Date. Every column of this file was in general format.

● Outreach Data:

- Applicant Data dataset had 37882 rows with 8 columns.
- The column names are ID, Country, University, Phone_Number. Every column of this file was in general format.
- OutreachData's Reference_ID column has 1632 inconsistent data.

The screenshot shows the pgAdmin 4 interface. On the left is the Object Explorer tree, which includes a 'Servers' node, a 'PostgreSQL 17' node, and a 'Databases' node containing 'Excelsert_Internship'. Inside 'Excelsert_Internship', there are nodes for 'Casts', 'Catalogs', 'Event Triggers', 'Extensions', 'Foreign Data Wrappers', 'Languages', 'Publications', and 'Schemas'. One schema, 'public', is selected. The main area contains a 'Query' tab with the following SQL code:

```

44 );
45
46 copy OutreachData
47 from 'E:\Data Analysis\Excelsert Internship\Week 1\Resource\Csv_files\OutreachData.csv'
48 WITH (FORMAT csv, HEADER true);
49
50 SELECT
51     COUNT(*) FILTER (WHERE Reference_ID ~ '[,/.\\-]') AS inconsistent_Reference_ID,
52     COUNT(*) FILTER (WHERE NOT (Reference_ID ~ '[,/.\\-]')) AS clean_ReferenceID
53 FROM OutreachData;
54

```

The 'Data Output' tab shows the results of the query:

	inconsistent_reference_id	clean.referenceid
1	1632	36249

Total rows: 1 Query complete 00:00:00.071 CRLF Ln 50, Col 1

2. Data Cleaning:

- **Applicant Data:**

- i. In this dataset, under “App_ID” column which is the primary key of this dataset, has null values, inconsistent values and missing values. To solve these issues, we have dropped all the rows where primary key is missing. And then found inputs like combinations of (‘, , //, “) these symbols. So, we deleted the corresponding rows which have these symbols. Moreover, we faced that there were some text data and primary key can not be text data which makes us to drop those rows too.
- ii. Country Column has lot of inappropriate values like emails, normal text and which will create an bad effect. And that is the reason I filtered out all the input not having country names and changed those inputs to “Not Mentioned” so we kept the data integrity by not dropping rows.
- iii. Phone_Number column was on “General” format which could be problematic for data modeling and is the reason why we changed it to number format.

- **Campaign Data:**

- i. In this dataset, there was a Start_Date column which we changed the format to Date.

ii. Additionally, there was a column called “name” which had informations like season, campaign type, region, and status. These informations are very necessary for the visualizations part if we can categorize them. So, we created 4 new column and named them as Season, Campaign_Region, Campaign_Type, and Campaign_Status. After that stored necessary informations into these newly created columns and if there was a nothing to put in respective column used “Not Mentioned” to avoid future sufferings when we will work on visualization.

- **Outreach Data:**

- i. “OutreachData” dataset had Reference_ID which was connected to “App_ID” of applicant data. But the difference is outreach data can have multiple same “Reference_ID” as same applicant can submit their data multiple times. So, here we kept all the duplicate values. But similar to “App_ID” cleaning process, we dropped all the rows having missing values and inconsistent values.
- ii. The dataset also had a crucial column called “Received_At” from which we created 6 new columns called Date, Year, Time, Hour, Weekday, and Month and added informations to these columns from “Received_At” column for the better visualizations and finding trends, patterns from the data.
- iii. Also added a “Outreach_ID” column which will be the primary key in this dataset and this will help in the data modeling like connecting with other dataset/tables in PostgreSQL for finding trends and patterns out of the data.

3. PostgreSql Setup Proof:

- Installed PostgreSql in my system. Importing Cleaned dataset into the PostgreSql to find trends and patterns from the data.

The screenshot shows the pgAdmin 4 interface. The left sidebar is the Object Explorer, which is currently expanded to show the 'Tables' node under the 'public' schema. The main area is a query editor titled 'Excelerate_Internship/postgres@PostgreSQL 17*'. It contains a SQL script for creating a table and copying data from a CSV file. The 'Data Output' tab at the bottom shows the result of the COPY command. The status bar at the bottom right indicates 'Query complete 00:00:00.053'.

```
FROM vbulletindata;
-- creating clean dataset
DROP TABLE IF EXISTS CleanApplicantData;
create table CleanApplicantData(
    App_ID Bigint,
    Country Varchar(56),
    University Varchar(255),
    Phone_Number Text
);
copy CleanApplicantData
from 'E:\Data Analysis\Excelerate Internship\Week 1\Resource\Cleaned_Csv_files\CleanApplicantData.csv'
WITH (FORMAT csv, HEADER true);

COPY 151175

Query returned successfully in 53 msec.
```

Total rows: Query complete 00:00:00.053 CRLF Ln 93, Col 32

View of connected Cleaned Applicant Data.

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer

PostgreSQL 17

Databases (3)

Excelerate_Internship

Casts Catalogs Event Triggers Extensions Foreign Data Wrappers Languages Publications Schemas Subscriptions

postgres

Casts Catalogs Event Triggers Extensions Foreign Data Wrappers Languages Publications Schemas

Schemas (1)

public

Aggregates Collations Domains FTS Configurations FTS Dictionaries FTS Parsers FTS Templates Foreign Tables Functions Materialized Views Operators Procedures Sequences Tables

Query History

```
86     Country Varchar(56),
87             University Varchar(255),
88             Phone_Number Text
89         );
90
91     copy CleanApplicantData
92     from 'E:\data\Analysis\Excelerate Internship\Week 1\Resource\Cleaned_Csv_files\CleanApplicantData.csv'
93     WITH (FORMAT csv, HEADER true);
94
95     Select * from CleanApplicantData;
96
97
98     DROP TABLE IF EXISTS Cleaned_CampaignData;
99     create table Cleaned_CampaignData(
```

Scratch Pad

Query

Copy

Insert

Update

Delete

Truncate

Refresh

No limit

SQL

Data Output

Messages

Notifications

Showing rows: 1 to 1000

Page No: 1 of 16

app_id	begin	country	university	phone_number
1	12345	India	Illinois Institute of Technology	9823241234
2	347397	Nigeria	Illinois Institute of Technology	7738599513
3	358065	India	Illinois Institute of Technology	920000000000
4	351333	India	Illinois Institute of Technology	13173701409
5	346435	India	Illinois Institute of Technology	917000000000
6	355959	India	Illinois Institute of Technology	920000000000
7	351520	India	Illinois Institute of Technology	183696266042
8	372165	India	Illinois Institute of Technology	919000000000
9	369273	India	Illinois Institute of Technology	916000000000
10	365995	India	Illinois Institute of Technology	918000000000
11	348627	India	Illinois Institute of Technology	919000000000
12	350814	India	Illinois Institute of Technology	919000000000
13	357845	India	Illinois Institute of Technology	919000000000
14	326070	India	Illinois Institute of Technology	919000000000

Total rows: 15175 Query complete 00:00:00.157

CRLF Ln 95 Col 1

Also connected the Cleaned_CampaignData to postgresql.

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer Excelsor_Internship/postgres@PostgreSQL 17*

Databases (3) Excelsor_Internship/postgres@PostgreSQL 17*

Excelsor_Internship Excelsor_Internship/postgres@PostgreSQL 17*

Query Scratch Pad

```

copy CleanApplicantData
from 'E:\Data Analysis\Excelsor Internship\Week 1\Resource\Cleaned_Csv_files\CleanApplic
WITH (FORMAT csv, HEADER true);

DROP TABLE IF EXISTS Cleaned_CampaignData;
create table Cleaned_CampaignData(
    ID Text,
    Name Varchar(255),
    Category Varchar(56),
    Intake Varchar(20),
    University Varchar(255),
    Status Varchar(20),
    Start_Date timestamp,
    Season Varchar(20),
    Campaign_Type Varchar(20),
    Campaign_Region Varchar(56),
    Campaign_Status Varchar(255)
);

set datestyle to MDY;
copy Cleaned_CampaignData
from 'E:\Data Analysis\Excelsor Internship\Week 1\Resource\Cleaned_Csv_files\Cleaned_Cam
WITH (FORMAT csv, HEADER true);

Select * from Cleaned_CampaignData;

```

Data Output Messages Notifications

	id	text	name	category	intake	university	status	start time
1	AANF23	GR GS FA24 Campaign- Admit, No Deposit	Post Admission	AY2024	Illinois Institute of Technology	Completed	202	
2	AND23	GR GS FA24 Campaign- Deposit No Action	Post Admission	AY2024	Illinois Institute of Technology	Completed	202	
3	BNAN...	GR GS FA24 Campaign- Deposit No I-20	Post Admission	AY2024	Illinois Institute of Technology	Completed	202	
4	BNND...	GR GS FA24 Campaign- In Progress	Pre Admission	AY2024	Illinois Institute of Technology	Completed	202	
5	CTKANF...	GR GS FA24 Campaign- Submit, Incomplete	Pre Admission	AY2024	Illinois Institute of Technology	Completed	202	
6	DANE24	GR GS Call Campaign: India ANF	Pre Admission	AY2024	Illinois Institute of Technology	Completed	202	
7	DNA24	GR GS Call Campaign: India No Deposit	Post Admission	AY2024	Illinois Institute of Technology	Completed	202	
8	FA24AND	GR GS Call Campaign: Other ANF	Post Admission	AY2024	Illinois Institute of Technology	Completed	202	
9	FA24DNA	GR GS Call Campaign: Other No Deposit	Post Admission	AY2024	Illinois Institute of Technology	Completed	202	
10	FA24DNI	GR GS CDYK Campaign: All I-20s Sent	Post Admission	AY2024	Illinois Institute of Technology	Completed	202	

Total rows: 23 Query complete 00:00:00.098

CRLF Ln 116, Col 1

Cleaned_OutreachData is also ready to find trends and patterns using PostgreSQL.

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer Databases(3) Excelsor_Internship/postgres@PostgreSQL 17*

```

118 Select * from Cleaned_CampaignData;
119
120
121 DROP TABLE IF EXISTS Cleaned_OutreachData;
122 create table Cleaned_OutreachData(
123     Outreach_ID int,
124     Reference_ID Bigint,
125     Received_At timestamp,
126     University varchar(255),
127     Cellar_Name varchar(20),
128     Outcome_1 Text,
129     Remark Text,
130     Campaign_ID Varchar(20),
131     Escalation_Required Varchar(20),
132     Date timestamp,
133     Year Varchar(10),
134     Time Time,
135     Hour int,
136     Weekday Varchar(20),
137     Month Varchar(20)
138 );
139
140 set datestyle to MDY;
141 copy Cleaned_OutreachData
142 from 'E:\Data Analysis\Excelsor Internship\Week 1\Resource\Cleaned_Csv_files\Cleaned_Out'
143 WITH (FORMAT csv, HEADER true);
144
145
146
147
148
149

```

Data Output Messages Notifications

NOTICE: table "cleaned_outreachdata" does not exist, skipping
COPY 33119

Query returned successfully in 321 msec.

Total rows: Query complete 00:00:00.321 CRLF Ln 137, Col 19

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer Databases(3) Excelsor_Internship/postgres@PostgreSQL 17*

```

136 Weekday Varchar(20),
137 Month Varchar(20)
138 );
139
140 set datestyle to MDY;
141 copy Cleaned_OutreachData
142 from 'E:\Data Analysis\Excelsor Internship\Week 1\Resource\Cleaned_Csv_files\Cleaned_Out'
143 WITH (FORMAT csv, HEADER true);
144
145 Select * from Cleaned_OutreachData;
146
147 -- checking Cleaned data is actually cleaned or not
148 -- CleanApplicantData
149 SELECT

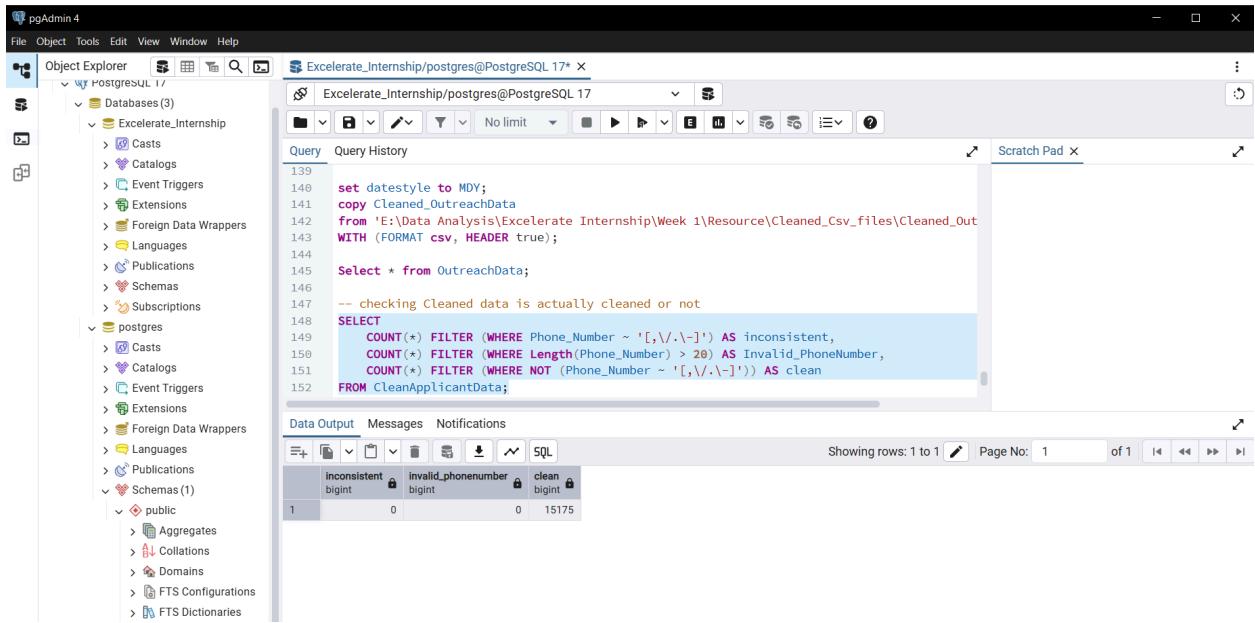
```

Data Output Messages Notifications

	outreach_id	reference_id	received_at	university	celler_name	outcome_1	remark
	integer	bigint	timestamp without time zone	character varying(255)	character varying(20)	text	text
1	1	12345	2023-04-28 12:15:00	Illinois Institute of Technology	Shailja	Connected	[null]
2	2	12345	2023-04-28 13:04:00	Illinois Institute of Technology	Shailja	Reschedule	[null]
3	3	12345	2023-05-01 11:14:00	Illinois Institute of Technology	Shailja	Connected	[null]
4	4	347397	2023-05-01 11:16:00	Illinois Institute of Technology	Isha	Not connected	[null]
5	5	347397	2023-05-01 11:18:00	Illinois Institute of Technology	Isha	Connected	[null]
6	6	358065	2023-05-01 11:19:00	Illinois Institute of Technology	Isha	Not connected	[null]
7	7	351333	2023-05-01 11:21:00	Illinois Institute of Technology	Isha	Not connected	[null]
8	8	346435	2023-05-01 11:26:00	Illinois Institute of Technology	Isha	Will Submit the docx within few days	[null]
9	9	355959	2023-05-01 11:29:00	Illinois Institute of Technology	Isha	Completed application	[null]
10	10	351520	2023-05-01 11:30:00	Illinois Institute of Technology	Shailja	Not connected	[null]
11	11	372165	2023-05-01 11:32:00	Illinois Institute of Technology	Isha	Not connected	[null]
12	12	369273	2023-05-01 11:32:00	Illinois Institute of Technology	Shailja	Completed application	[null]
13	13	365995	2023-05-01 11:33:00	Illinois Institute of Technology	Shailja	Not connected	[null]

Total rows: 33119 Query complete 00:00:00.200 CRLF Ln 145, Col 1

• Exploring the “Clean_ApplicantData”:



The screenshot shows the pgAdmin 4 interface with the following details:

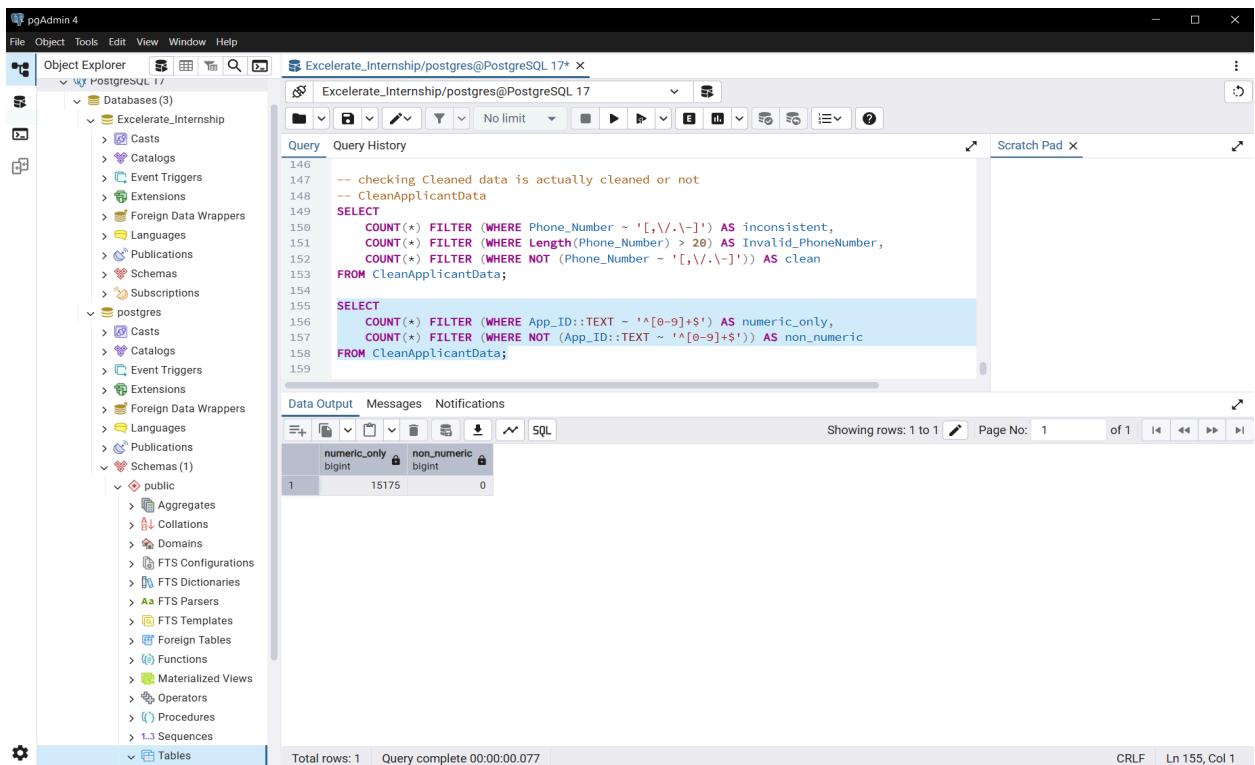
- Object Explorer:** Shows the database structure under "postresql 17" and "Excelerate_Internship".
- Query Editor:** Contains a SQL script to check for inconsistent phone numbers.
- Data Output:** A table showing the results of the query.

```

139 set datestyle to MDY;
140 copy Cleaned_OutreachData
141 from 'E:\Data Analysis\Excelerate Internship\Week 1\Resource\Cleaned_Csv_files\Cleaned_Out
142 WITH (FORMAT csv, HEADER true);
143
144 Select * from OutreachData;
145
146 -- checking Cleaned data is actually cleaned or not
147 SELECT
148     COUNT(*) FILTER (WHERE Phone_Number ~ '[,\\.-]') AS inconsistent,
149     COUNT(*) FILTER (WHERE Length(Phone_Number) > 20) AS Invalid_PhoneNumber,
150     COUNT(*) FILTER (WHERE NOT (Phone_Number ~ '[,\\.-]')) AS clean
151
152 FROM CleanApplicantData;
    
```

	inconsistent	invalid_phonenumber	clean
1	0	0	15175

No inconsistent data is found in “Phone_Number” column of cleaned applicant data.



The screenshot shows the pgAdmin 4 interface with the following details:

- Object Explorer:** Shows the database structure under "postresql 17" and "Excelerate_Internship".
- Query Editor:** Contains a SQL script to check for numeric-only and non-numeric App_IDs.
- Data Output:** A table showing the results of the query.

```

146 -- checking Cleaned data is actually cleaned or not
147 -- CleanApplicantData
148
149 SELECT
150     COUNT(*) FILTER (WHERE Phone_Number ~ '[,\\.-]') AS inconsistent,
151     COUNT(*) FILTER (WHERE Length(Phone_Number) > 20) AS Invalid_PhoneNumber,
152     COUNT(*) FILTER (WHERE NOT (Phone_Number ~ '[,\\.-]')) AS clean
153
154
155     COUNT(*) FILTER (WHERE App_ID::TEXT ~ '^[0-9]+$') AS numeric_only,
156     COUNT(*) FILTER (WHERE NOT (App_ID::TEXT ~ '^[0-9]+$')) AS non_numeric
157
158 FROM CleanApplicantData;
    
```

	numeric_only	non_numeric
1	15175	0

Total rows: 1 | Query complete 00:00:00.077 | CRLF | Ln 155, Col 1

So, it is clear that there is not any non numeric value in our primary key which is the most important column, “App_ID”.

● Exploring the “Cleaned_OutreachData”:

```

File Object Tools Edit View Window Help
Object Explorer Databases(3) Excelsor_Internship/postgres@PostgreSQL 17* X
File Object Tools Edit View Window Help
Object Explorer Databases(3) Excelsor_Internship/postgres@PostgreSQL 17* X
Query Query History
136     Weekday Varchar(20),
137     Month Varchar(20)
138 );
139
140 set datestyle to MDY;
141 copy Cleaned_OutreachData
142 from 'E:\Data Analysis\Excelsor Internship\Week 1\Resource\Cleaned_Csv_files\Cleaned_Out
143 WITH (FORMAT csv, HEADER true);
144
145 Select * from Cleaned_OutreachData;
146 Select Count(*) from Cleaned_OutreachData;
147
148 -- checking Cleaned data is actually cleaned or not
149 -- CleanApplicantData
Data Output Messages Notifications
Showing rows: 1 to 1 Page No: 1 of 1
1 count
1 33119

```

These are the callers who has been contacting with applicants.

```

172 SELECT Caller_Name, COUNT(*) AS total_callers
173 FROM Cleaned_OutreachData
174 GROUP BY Caller_Name
175 ORDER BY total_callers DESC;
176
177
178
Data Output Messages Notifications
SQL


|    | caller_name            | total_callers |
|----|------------------------|---------------|
|    | character varying (20) | bigint        |
| 1  | Rudra                  | 11564         |
| 2  | Prajwal                | 7218          |
| 3  | Jyoti                  | 4499          |
| 4  | Palak                  | 4130          |
| 5  | Isha                   | 1709          |
| 6  | Poppy                  | 1345          |
| 7  | Shruti                 | 1268          |
| 8  | Shailja                | 1171          |
| 9  | Namrata                | 111           |
| 10 | Twinkle                | 77            |
| 11 | Pranjal                | 26            |
| 12 | Mounika                | 1             |


Total rows: 12 Query complete 00:00:00.093

```

From the above picture we can see, Rudra has the most clients with 11564. Prjawal and Jyoti has the second and third most clients. So, we can say, Rudra is the top performer among the callers.

```

177      Select Distinct Count(Reference_ID) As Total_Applicants
178      from Cleaned_OutreachData;
179
180
181

```

The screenshot shows the pgAdmin 4 interface. The main area displays a SQL query and its execution results. The query is:

```

Select Distinct Count(Reference_ID) As Total_Applicants
from Cleaned_OutreachData;

```

The results table has one row with the following data:

	total_applicants
1	33119

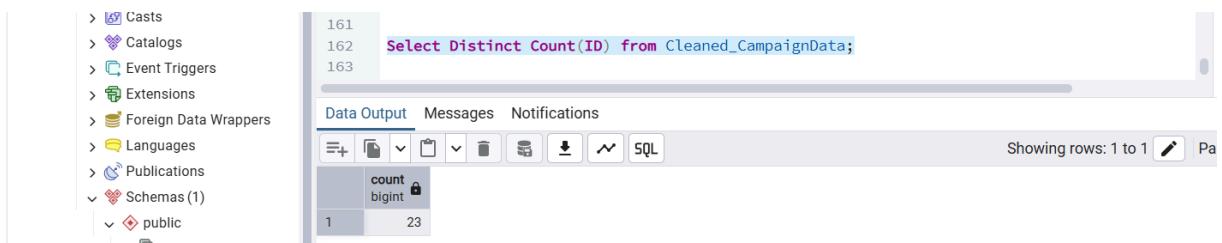
From Cleaned_OutreachData, we get to know there are 33119 applicants applied in “Illinois Institute of Technology” University through different campaigns.

• Exploring the “Cleaned_CampaignData”

The screenshot shows the pgAdmin 4 interface with a complex SQL script in the Query tab. The script includes several lines of code, such as setting datestyle, copying data from a CSV file, and selecting count(*) from the table. The results table has one row with the following data:

	count
1	23

All the rows are there in the dataset after the integration with Postgresql.

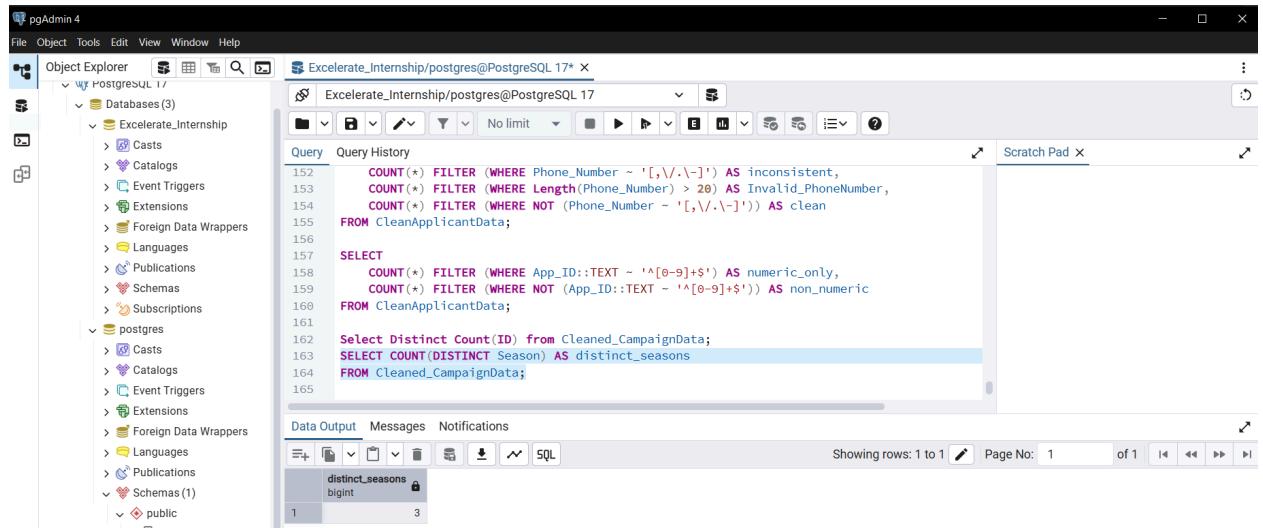


The screenshot shows the pgAdmin 4 interface. On the left is the Object Explorer pane, which lists various database objects like Casts, Catalogs, Event Triggers, Extensions, Foreign Data Wrappers, Languages, Publications, Schemas, and the public schema. The main pane displays a SQL query and its results. The query is:

```
161
162 Select Distinct Count(ID) from Cleaned_CampaignData;
163
```

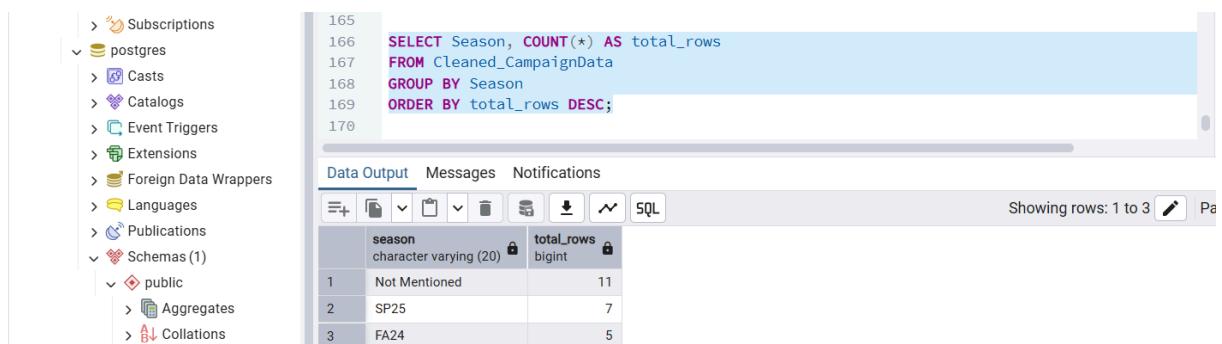
The results table shows one row with a column labeled "count" and a value of 23. The status bar at the bottom right indicates "Showing rows: 1 to 1".

The primary key of Cleaned_CampaignData is all unique as we got same row nums.



The screenshot shows the pgAdmin 4 interface. The Object Explorer pane shows the same database structure as the previous screenshot. The main pane contains a complex SQL script with multiple COUNT(*) and FILTER clauses, followed by a SELECT statement and two additional COUNT(*) queries. The results table shows one row with a column labeled "distinct_seasons" and a value of 3. The status bar at the bottom right indicates "Showing rows: 1 to 1".

There are 3 types of entries in the season column. And These are the entries of this column.



The screenshot shows the pgAdmin 4 interface. The Object Explorer pane shows the database structure. The main pane displays a query to count rows by season, ordered by total rows in descending order. The query is:

```
165
166 SELECT Season, COUNT(*) AS total_rows
167 FROM Cleaned_CampaignData
168 GROUP BY Season
169 ORDER BY total_rows DESC;
170
```

The results table shows three rows with columns "season" and "total_rows". The data is:

season	total_rows
Not Mentioned	11
SP25	7
FA24	5

The status bar at the bottom right indicates "Showing rows: 1 to 3".

● Creating Master Table:

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer

PostgreSQL 17

Databases (3)

- Excelerate_Internship
 - Casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas
 - Subscriptions
- postgres
 - Casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains

week1_postgreSQL_queries.sql*

Excelerate_Internship/postgres@PostgreSQL 17

Query Query History

```

CREATE TABLE MasterTable AS
SELECT
    a.App_ID,
    a.Country,
    a.University,
    a.Phone_Number,
    c.ID AS Campaign_ID,
    c.Name AS Campaign_Name,
    c.Season,
    c.Campaign_Type,
    c.Campaign_Region,
    o.Outreach_ID,
    o.Received_At,
    o.Outcome_1,
    o.Remark
FROM CleanApplicantData a
FULL OUTER JOIN Cleaned_OutreachData o
    ON a.App_ID = o.Reference_ID
FULL OUTER JOIN Cleaned_CampaignData c
    ON o.Campaign_ID = c.ID;

SELECT *
FROM MasterTable
LIMIT 10;

```

SQL

app_id	country	university	phone_number	campaign_id	campaign_name	session	campaign_type	campaign_region	outreach_id	received_at
1	12345	India	character varying (56)	9832341234	IANF23	GR GS SP25 Campaign-Deferrals to SP25	SP25	Campaign	Not Mentioned	2023-04-28 12:15:00
2	12345	India	Illinois Institute of Technology	9823241234	IANF23	GR GS SP25 Campaign-Deferrals to SP25	SP25	Campaign	Not Mentioned	2023-04-28 13:04:00
3	12345	India	Illinois Institute of Technology	9823241234	IANF23	GR GS SP25 Campaign-Deferrals to SP25	SP25	Campaign	Not Mentioned	2023-05-01 11:14:00
4	347397	Nigeria	Illinois Institute of Technology	773899513	IANF23	GR GS SP25 Campaign-Deferrals to SP25	SP25	Campaign	Not Mentioned	2023-05-01 11:16:00
5	347397	Nigeria	Illinois Institute of Technology	773899513	IANF23	GR GS SP25 Campaign-Deferrals to SP25	SP25	Campaign	Not Mentioned	2023-05-01 11:18:00
6	358065	India	Illinois Institute of Technology	92000000000	IANF23	GR GS SP25 Campaign-Deferrals to SP25	SP25	Campaign	Not Mentioned	2023-05-01 11:19:00
7	351333	India	Illinois Institute of Technology	13173701409	IANF23	GR GS SP25 Campaign-Deferrals to SP25	SP25	Campaign	Not Mentioned	2023-05-01 11:21:00
8	346435	India	Illinois Institute of Technology	91700000000	IANF23	GR GS SP25 Campaign-Deferrals to SP25	SP25	Campaign	Not Mentioned	2023-05-01 11:26:00
9	355959	India	Illinois Institute of Technology	92000000000	IANF23	GR GS SP25 Campaign-Deferrals to SP25	SP25	Campaign	Not Mentioned	2023-05-01 11:29:00
10	351520	India	Illinois Institute of Technology	1836926042	IANF23	GR GS SP25 Campaign-Deferrals to SP25	SP25	Campaign	Not Mentioned	2023-05-01 11:30:00

Total rows: 10 Query complete 00:00:00.082

File Object Tools Edit View Window Help

Quick search

33°C Mostly cloudy 4:44 PM 9/8/2023

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer

- PostgreSQL 17
- Databases (3)
 - Excelerate_Internship
 - Casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas
 - Subscriptions
- postgres
- Casts
- Catalogs
- Event Triggers
- Extensions
- Foreign Data Wrappers
- Languages
- Publications
- Schemas (1)
- public
- Aggregates
- Collations
- Domains
- FTS Configurations
- FTS Dictionaries
- FTS Parsers
- FTS Templates
- Foreign Tables
- Functions
- Materialized Views
- Operators
- Procedures
- Sequences
- Tables
- Trigger Functions
- Types
- Views
- Subscriptions
- test
- Login/Group Roles
- Tablespaces

Quick search

File Object Tools Edit View Window Help

Query History

```

194     c.Name AS Campaign_Name,
195     c.Session,
196     c.Campaign_Type,
197     c.Campaign_Location,
198     o.Outreach_ID,
199     o.Received_At,
200     o.Outcome_1,
201     o.Remark
202   FROM CleanedCampaignData a
203   FULL OUTER JOIN Cleaned_OutreachData o
204     ON a.App_ID = o.Reference_ID
205   FULL OUTER JOIN Cleaned_CampaignData c
206     ON o.Campaign_ID = c.ID;
207
208   SELECT *
209   FROM MasterTable
210
211   LIMIT 10;
  
```

Data Output Messages Notifications

	phone_number	campaign_id	campaign_name	season	campaign_type	campaign_region	outreach_id	received_at	outcome_1	remark
1	stitute of Technology	IANF23	GR GS SP25 Campaign Deferrals to SP25	SP25	Campaign	Not Mentioned		1 2023-04-28 12:15:00	Connected	[null]
2	stitute of Technology	IANF23	GR GS SP25 Campaign Deferrals to SP25	SP25	Campaign	Not Mentioned		2 2023-04-28 13:04:00	Reschedule	[null]
3	stitute of Technology	IANF23	GR GS SP25 Campaign Deferrals to SP25	SP25	Campaign	Not Mentioned		3 2023-05-01 11:11:400	Connected	[null]
4	stitute of Technology	IANF23	GR GS SP25 Campaign Deferrals to SP25	SP25	Campaign	Not Mentioned		4 2023-05-01 11:11:600	Not connected	[null]
5	stitute of Technology	7738599513	IANF23	GR GS SP25 Campaign Deferrals to SP25	SP25	Campaign	Not Mentioned	5 2023-05-01 11:11:800	Connected	[null]
6	stitute of Technology	92000000000	IANF23	GR GS SP25 Campaign Deferrals to SP25	SP25	Campaign	Not Mentioned	6 2023-05-01 11:11:900	Not connected	[null]
7	stitute of Technology	13179701409	IANF23	GR GS SP25 Campaign Deferrals to SP25	SP25	Campaign	Not Mentioned	7 2023-05-01 11:21:00	Not connected	[null]
8	stitute of Technology	91700000000	IANF23	GR GS SP25 Campaign Deferrals to SP25	SP25	Campaign	Not Mentioned	8 2023-05-01 11:26:00	Will Submit the docx within few days	[null]
9	stitute of Technology	92000000000	IANF23	GR GS SP25 Campaign Deferrals to SP25	SP25	Campaign	Not Mentioned	9 2023-05-01 11:29:00	Completed application	[null]
10	stitute of Technology	18369626042	IANF23	GR GS SP25 Campaign Deferrals to SP25	SP25	Campaign	Not Mentioned	10 2023-05-01 11:30:00	Not connected	[null]

Total rows: 10 Query complete 00:00:00.082

File Object Tools Edit View Window Help

Query History

```

209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
  
```

SELECT

```

    COUNT(*) FILTER (WHERE App_ID IS NOT NULL
                     AND Outreach_ID IS NOT NULL
                     AND Campaign_ID IS NOT NULL) AS full_linked,
    COUNT(*) FILTER (WHERE App_ID IS NOT NULL
                     AND Outreach_ID IS NULL
                     AND Campaign_ID IS NULL) AS applicants_only,
    COUNT(*) FILTER (WHERE App_ID IS NULL
                     AND Outreach_ID IS NOT NULL
                     AND Campaign_ID IS NOT NULL) AS outreach_only,
    COUNT(*) FILTER (WHERE App_ID IS NULL
                     AND Outreach_ID IS NULL
                     AND Campaign_ID IS NOT NULL) AS campaigns_only
  
```

FROM MasterTable;

Data Output Messages Notifications

	full_linked	applicants_only	outreach_only	campaigns_only
1	33119	16	0	0

Showing 1 rows

So, here we can see,

1. 33119 under “full_linked” which means rows where we had an App_ID, Outreach_ID, and Campaign_ID all matched.
2. 16 under “applicants_only” depicts 16 applicants exist in our applicant table but have no outreach and no campaign linked.
3. “outreach_only” has 0 which indicates that no outreach rows are floating around without applicant or campaign.

4. “campaigns_only” also has 0 shows, no campaign rows exist without outreach/applicant.

This illustrates that our Master Table (data) is very well connected and reliable where 99.95% of records are fully linked (33,119 out of 33,135). Only 16 applicants are “orphaned”. And these applicants exist in our “CleanApplicantData”, but no one reached out to them and they don’t belong to any campaign.

Furthermore, These are the 16 applicants who could not connect with master table.

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the Object Explorer with the PostgreSQL 17 database selected, showing various schema and table structures. The main area has a query editor titled "week1_postgreSQL_queries.sql*" containing the following SQL code:

```
AND Outreach_ID IS NOT NULL
AND Campaign_ID IS NULL) AS outreach_only,
COUNT(*) FILTER (WHERE App_ID IS NULL
AND Outreach_ID IS NULL
AND Campaign_ID IS NOT NULL) AS campaigns_only

FROM MasterTable;

-- 16 applicants who are Orphan or not connected with master table
SELECT
App_ID,
Country,
University,
Phone_Number
FROM MasterTable
WHERE Outreach_ID IS NULL
AND Campaign_ID IS NULL
AND App_ID IS NOT NULL;
```

Below the query editor is a Data Output tab showing the results of the last query:

app_id	country	university	phone_number
1	Pakistan	Illinois Institute of Technology	923000000000
2	Not Mentioned	Illinois Institute of Technology	440191
3	Not Mentioned	Illinois Institute of Technology	821000000000
4	South Africa	Illinois Institute of Technology	362056
5	South Africa	Illinois Institute of Technology	234000000000
6	Not Mentioned	Illinois Institute of Technology	552000000000
7	India	Illinois Institute of Technology	989000000000
8	Ethiopia	Illinois Institute of Technology	477582
9	India	Illinois Institute of Technology	916000000000
10	India	Illinois Institute of Technology	919000000000
11	Nigeria	Illinois Institute of Technology	235000000000
12	Bangladesh	Illinois Institute of Technology	405046
13	Nepal	Illinois Institute of Technology	434991
14	India	Illinois Institute of Technology	918000000000
15	India	Illinois Institute of Technology	917000000000
16	India	Illinois Institute of Technology	920000000000

Conclusion:

In this project, there are three datasets: CleanApplicantData, Cleaned_CampaignData, and Cleaned_OutreachData. And we cleaned and integrated into a single Master Table using a full outer join. The Master Table provides the full view of applicants, their associated campaigns, and outreach activities, enabling comprehensive analysis of the end-to-end process.

The results of the Master Table showed that out of 33,135 total records, 33,119 were fully connected across applicants, campaigns, and outreach, while only 16 applicants remained disconnected, which is showcasing a high level of data consistency.

The Master Table serves as a single source of truth, simplifies future analysis, ensures data quality, and supports decision-making by offering a holistic view of the applicant engagement lifecycle.