

# Data Quality Exploratory Data Analysis (EDA) Report by Team 2

## Dataset Overview

This report reviews key data quality and structural issues identified across datasets related to learners, cohorts, marketing campaigns, and user demographics. The objective is to provide clear observations, risks, and recommended mitigation strategies to improve data integrity and analytic reliability.

### Datasets Covered:

- Learner Dataset
- Content and Tracking Data
- Cohort Data
- Marketing Campaign Data
- Data Import Logs
- User Demographic Data

## Summary Stats

- The status field in the Learner dataset contains undocumented numeric codes.
- Duplicate learner\_id and apply\_date combinations detected.
- Multiple cohort assignments found per learner.
- Tracking data shows many NULL entries for questions and inconsistent naming.
- Cohort sizes range from unrealistically large to zero-day durations.
- Marketing campaigns have duplicate names and inactive campaigns with spend.
- Data import suffers from malformed rows and inconsistent ID formats.
- User demographics include missing birthdates, inconsistent gender values, and invalid emails.

## Missing & Duplicates

- Duplicate learner\_id/apply\_date pairs could bias enrollment counts.
- Missing critical fields such as tracking questions and demographic attributes.
- NULL and literal "NULL" values present in NOT NULL fields.
- Duplicate campaign names (e.g., "Copy 3", "Copy 4") affect marketing analysis.
- Malformed rows cause import failures and incomplete records.

## Outliers

- Epoch timestamps in UNIX format require conversion.
- Cohort sizes up to 100,000 and durations spanning years flagged.
- Marketing cost per result varies significantly; some campaigns show low engagement despite high reach.
- User demographic zip codes show inconsistent formats.
- Outlier cohort durations and size discrepancies need validation.

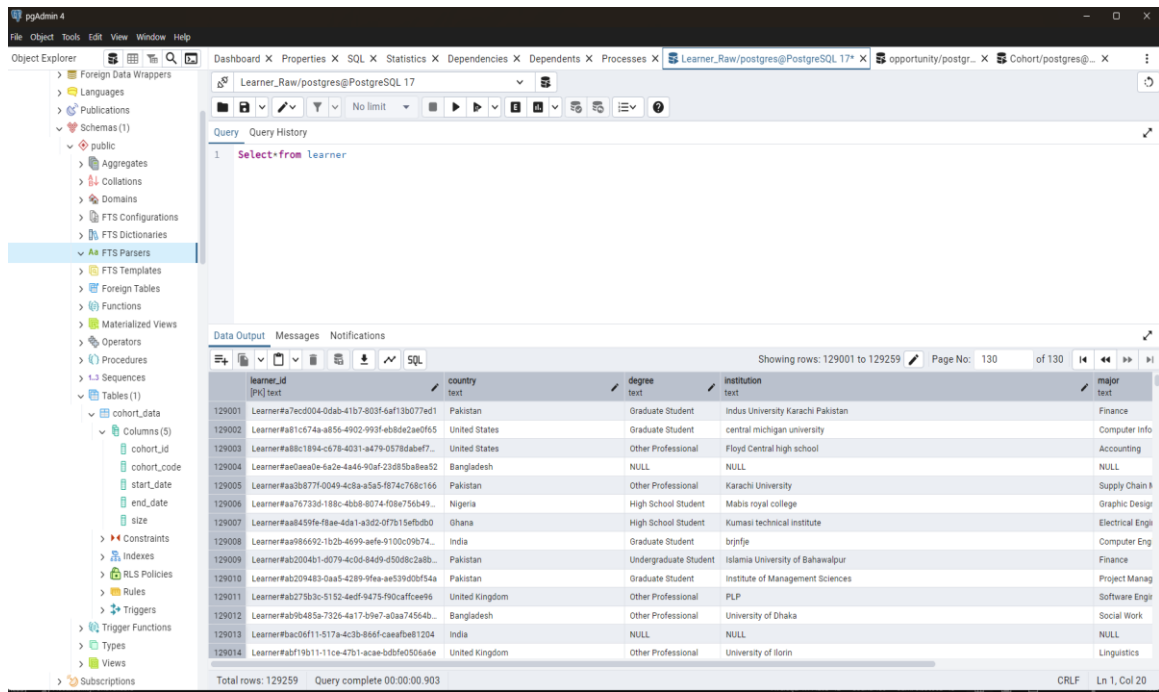
## Trend Visuals

- Learner enrollment over time, highlighting duplicates.
- Cohort size and duration distribution.
- Marketing campaign spend vs results over periods.
- Demographic data completeness trends.
- Status code frequency and anomaly detection.

## Key Insights

- Enforce data validation rules and create data dictionaries for undocumented codes.
- Implement deduplication and uniqueness constraints on learner and cohort data.
- Standardize naming conventions and clean tracking data entries.
- Set thresholds and validation rules for cohort sizes and durations.
- Optimize marketing campaign lifecycle management to avoid wasted spend.
- Improve import procedures with schema validation and pre-processing.
- Normalize and enrich demographic data to improve analytic accuracy.
- Adopt consistent timestamp and ID formatting standards across datasets.

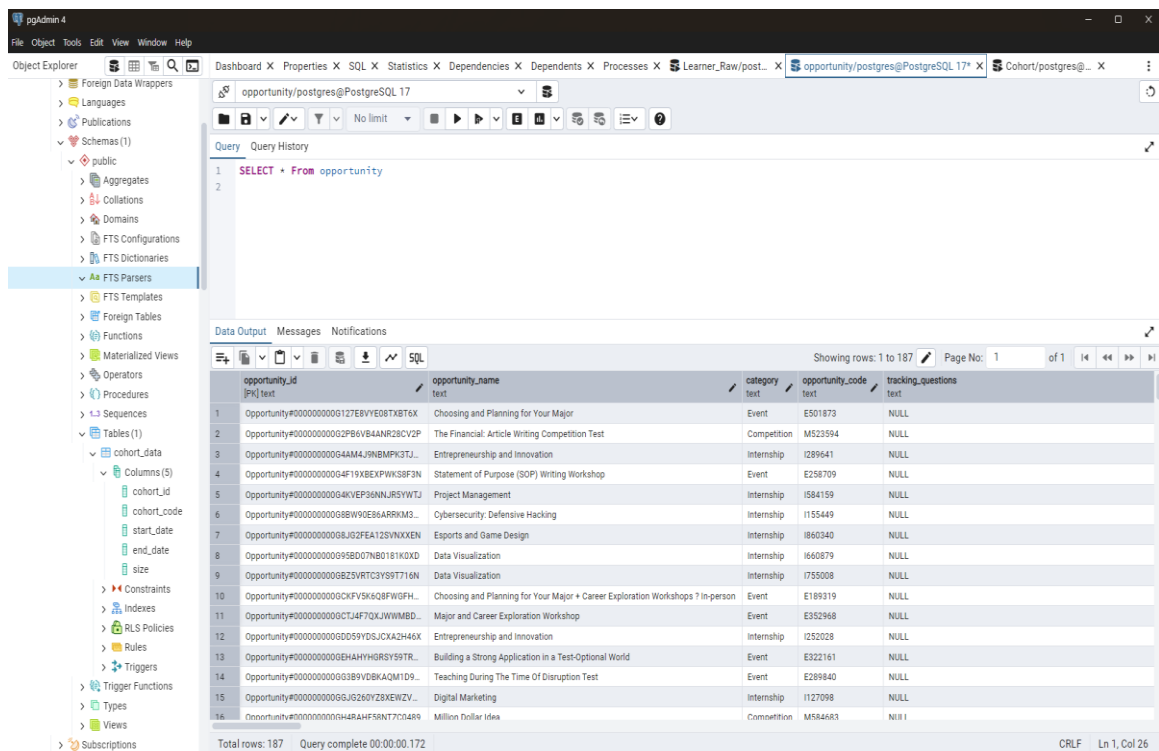
## 1. Image of Learner\_Raw\_Data successfully imported in PostgreSQL



The screenshot shows the pgAdmin 4 interface with the 'Learner' table selected in the 'public' schema. The table has 6 columns: learner\_id, country, degree, institution, and major. The data is displayed in a grid format, showing rows 129,001 to 129,259. The total number of rows is 129,259, and the query is complete in 00:00:00.903.

learner_id	country	degree	institution	major
129001	Pakistan	Graduate Student	Indus University Karachi Pakistan	Finance
129002	United States	Graduate Student	central michigan university	Computer Info
129003	United States	Other Professional	Floyd Central high school	Accounting
129004	Bangladesh	NULL	NULL	NULL
129005	Pakistan	Other Professional	Karachi University	Supply Chain M
129006	Nigeria	High School Student	Mabis royal college	Graphic Design
129007	Ghana	High School Student	Kumasi technical institute	Electrical Engi
129008	India	Graduate Student	brjnfe	Computer Eng
129009	Pakistan	Undergraduate Student	Islamia University of Bahawalpur	Finance
129010	Pakistan	Graduate Student	Institute of Management Sciences	Project Manag
129011	United Kingdom	Other Professional	PLP	Software Engi
129012	Bangladesh	Other Professional	University of Dhaka	Social Work
129013	India	NULL	NULL	NULL
129014	United Kingdom	Other Professional	University of florin	Linguistics

## 2. Image of Opportunity\_Data successfully Imported.



The screenshot shows the pgAdmin 4 interface with the 'opportunity' table selected in the 'public' schema. The table has 6 columns: opportunity\_id, opportunity\_name, category, opportunity\_code, and tracking\_questions. The data is displayed in a grid format, showing rows 1 to 187. The total number of rows is 187, and the query is complete in 00:00:00.172.

opportunity_id	opportunity_name	category	opportunity_code	tracking_questions
1	Choosing and Planning for Your Major	Event	E501873	NULL
2	The Financial: Article Writing Competition Test	Competition	M523594	NULL
3	Entrepreneurship and Innovation	Internship	I289641	NULL
4	Statement of Purpose (SOP) Writing Workshop	Event	E258709	NULL
5	Project Management	Internship	I584159	NULL
6	Cybersecurity: Defensive Hacking	Internship	I155449	NULL
7	Exports and Game Design	Internship	I860340	NULL
8	Data Visualization	Internship	I660879	NULL
9	Data Visualization	Internship	I755008	NULL
10	Choosing and Planning for Your Major + Career Exploration Workshops 7 in-person	Event	E189319	NULL
11	Major and Career Exploration Workshop	Event	E352968	NULL
12	Entrepreneurship and Innovation	Internship	I252028	NULL
13	Building a Strong Application in a Test-Optional World	Event	E322161	NULL
14	Teaching During The Time of Disruption Test	Event	E289840	NULL
15	Digital Marketing	Internship	I127098	NULL
16	Million Dollar Idea	Competition	M584683	NULL

### 3. Image of Cohort\_Data successfully imported

The screenshot shows the pgAdmin 4 interface with the 'Cohort/postgres@PostgreSQL 17' connection. The 'Tables (1)' folder is expanded, showing the 'cohort\_data' table. The table's columns are: cohort\_id (character varying (20)), cohort\_code (character varying (20)), start\_date (bigint), end\_date (bigint), and size (integer). The table contains 641 rows. The data output is displayed in a table with the following columns: cohort\_id, cohort\_code, start\_date, end\_date, and size. The data is as follows:

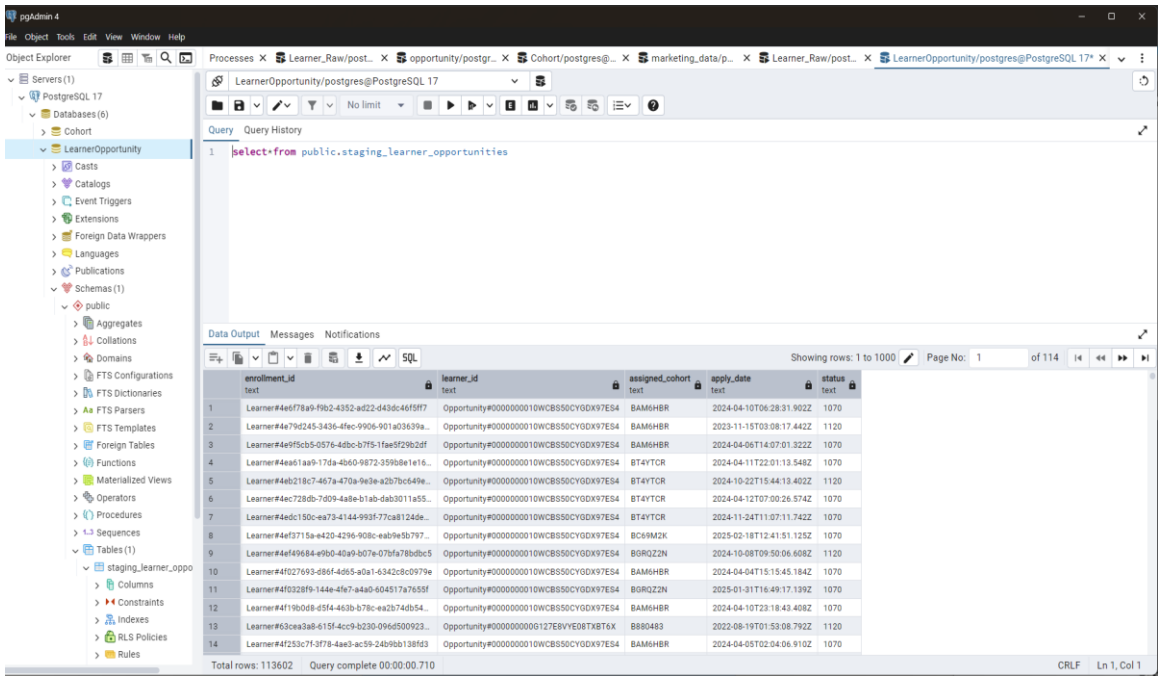
cohort_id	cohort_code	start_date	end_date	size
1 Cohort#	B456514	1680500000000	1682960000000	1500
2 Cohort#	B328821	1673850000000	1676910000000	1000
3 Cohort#	B289256	1668400000000	1671080000000	100000
4 Cohort#	B0VCB0F	1663890000000	1663890000000	40
5 Cohort#	B908347	1673240000000	1676310000000	100000
6 Cohort#	B306047	1673240000000	1676310000000	10000
7 Cohort#	B883644	1656650000000	1659240000000	1500
8 Cohort#	B280844	1668400000000	1671080000000	100000
9 Cohort#	B466039	1664770000000	1667450000000	100000
10 Cohort#	B061140	1673240000000	1676310000000	10000
11 Cohort#	B759124	1665370000000	1668530000000	1500
12 Cohort#	B304681	1664770000000	1667450000000	100000
13 Cohort#	B809432	1657890000000	1657890000000	143
14 Cohort#	B495792	1657890000000	1657900000000	1771

### 4. Image of Marketing\_Data successfully imported

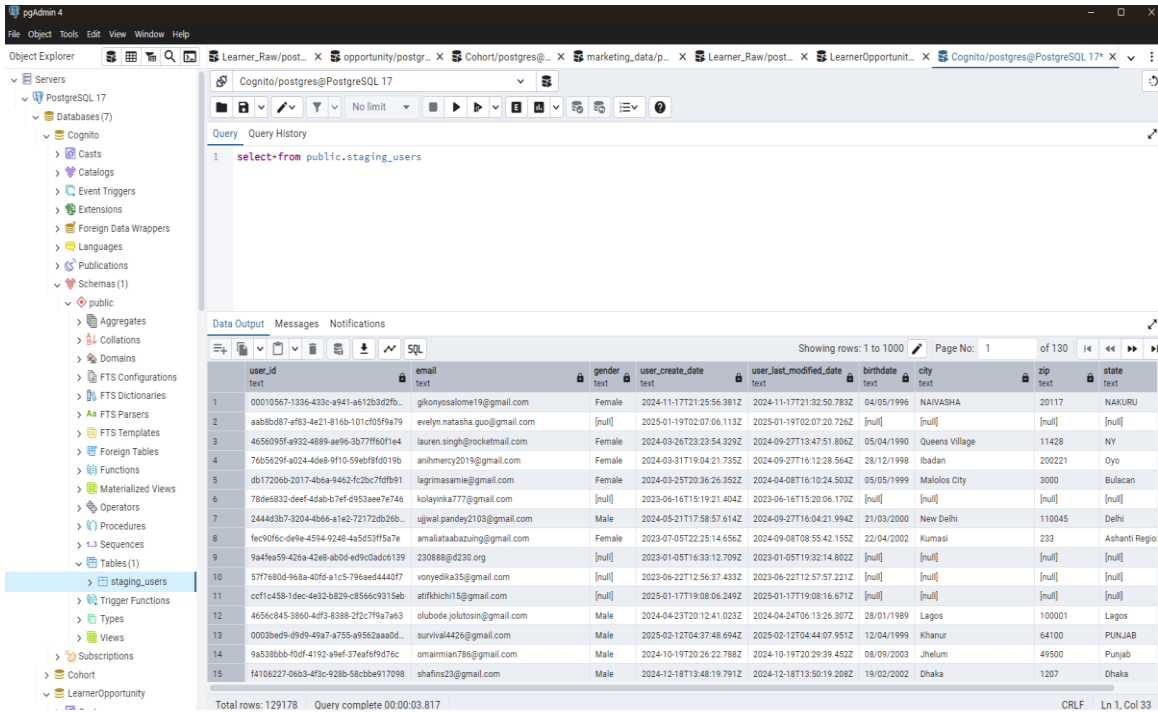
The screenshot shows the pgAdmin 4 interface with the 'marketing\_data/postgres@PostgreSQL 17' connection. The 'Tables (1)' folder is expanded, showing the 'ad\_campaign\_data' table. The table's columns are: ad\_account\_name (text), campaign\_name (text), delivery\_status (text), delivery\_level (text), reach (bigint), outbound\_clicks (integer), landing\_page\_views (integer), and result\_type (text). The table contains 148 rows. The data output is displayed in a table with the following columns: ad\_account\_name, campaign\_name, delivery\_status, delivery\_level, reach, outbound\_clicks, landing\_page\_views, and result\_type. The data is as follows:

ad_account_name	campaign_name	delivery_status	delivery_level	reach	outbound_clicks	landing_page_views	result_type
1 SLU	##B2: Digital Marketing Intern - May Ads: Website Leads Prospecting   18 to 35 years - Copy 4	completed	campaign	102962	1815	1310	Website a
2 SLU	##B2: Digital Marketing Intern - May Ads: Website Leads Prospecting   18 to 35 years - Copy 3	completed	campaign	180175	3378	2152	Website a
3 SLU	##B2: Digital Marketing Intern - May Ads: Website Leads Prospecting   18 to 35 years - Copy 3	inactive	campaign	173118	3591	2767	Website a
4 SLU	#Brand Awareness: UGC Video - March - Copy	inactive	campaign	18355415	5431	1019	Reach
5 SLU	#Data Analyst Associate Internship	inactive	campaign	2448	111	62	Website k
6 SLU	A1: Outreach Consultant Intern - March Ads: Website Leads Prospecting   18 to 35 years - Copy 2	inactive	campaign	1136229	12798	8196	Website k
7 SLU	A2: Digital Strategy Associate Intern - March Ads: Website Leads Prospecting   18 to 35 years - Copy 2	inactive	campaign	682351	9510	6912	Website k
8 SLU	A3: Data Analyst Associate Intern - March Ads: Website Leads Prospecting   18 to 35 years - Copy 2	inactive	campaign	1077778	21464	16914	Website k
9 SLU	A4: Project Management Associate Intern - March Ads: Website Leads Prospecting   18 to 35 years - Copy 2	inactive	campaign	934611	13060	8935	Website k
10 SLU	A5: Business Strategy Intern - March Ads: Website Leads Prospecting   18 to 35 years - Copy 2	inactive	campaign	999159	13337	9112	Website k
11 SLU	AeroFlow Challenge	completed	campaign	142743	1922	1149	Website a
12 SLU	AI Forensic Challenge	completed	campaign	114896	1122	818	Website a
13 SLU	AUG: AeroFlow Challenge	completed	campaign	64594	804	468	Website a
14 SLU	AUG: Power Course: Digital Wellness - Copy	inactive	campaign	169228	660	271	Website a
15 SLU	AUG_Course: CPR course - Copy	completed	campaign	148357	2037	1297	Website a
16 SLU	Awareness: Do you want to stand out?   Reel and story   Video Views	completed	campaign	1261904	3644	808	ThruPlay

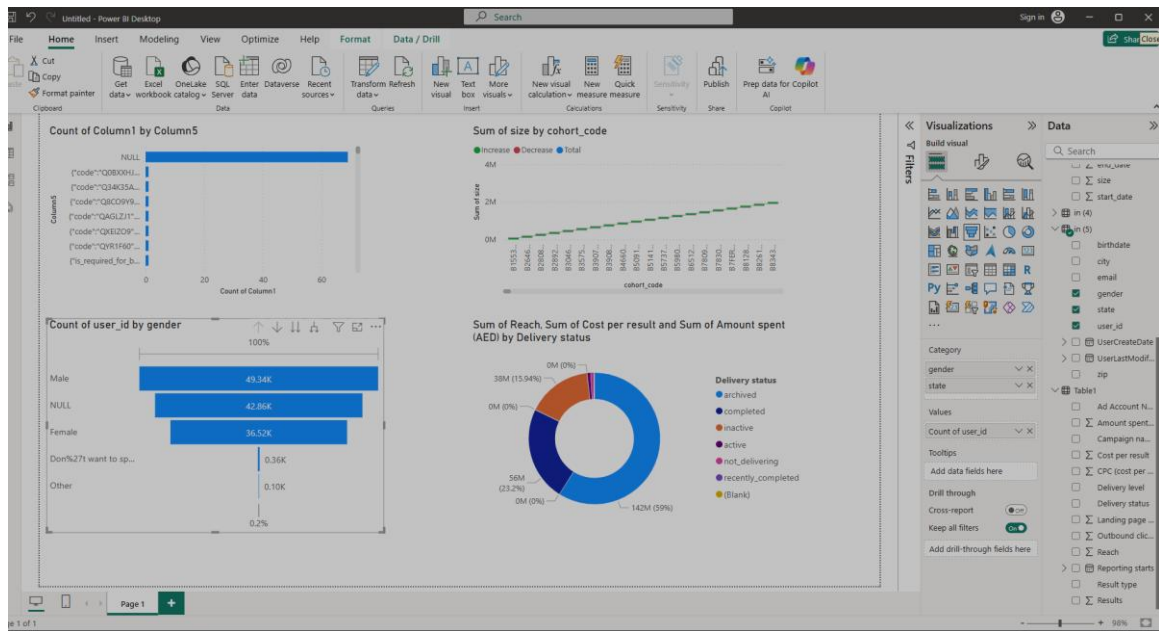
### 5. Image of LearnerOpportunity\_Raw\_Data successfully imported



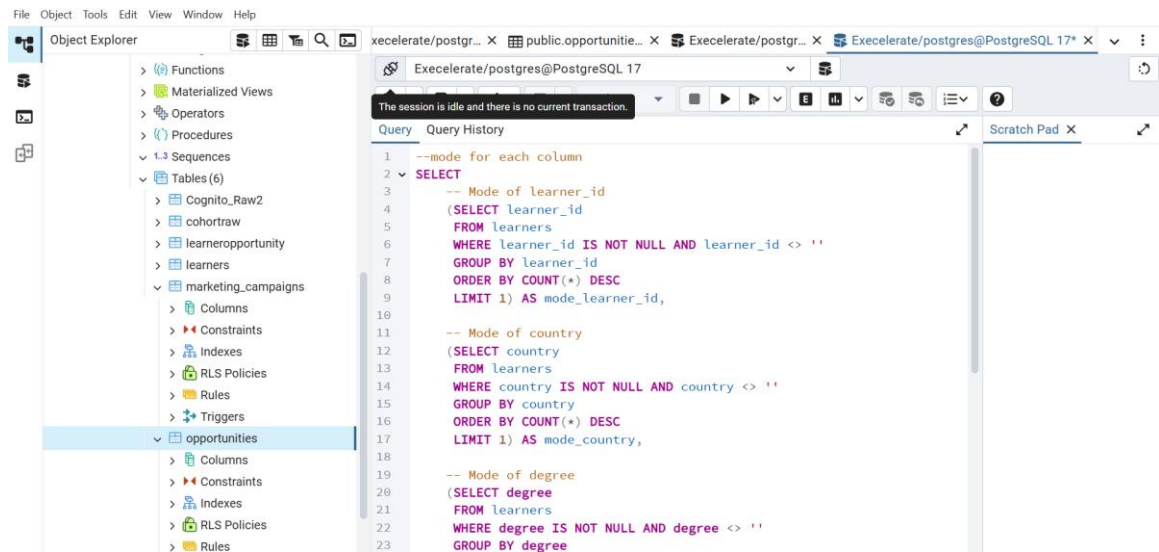
## 6. Image of Cognito\_Data successfully imported

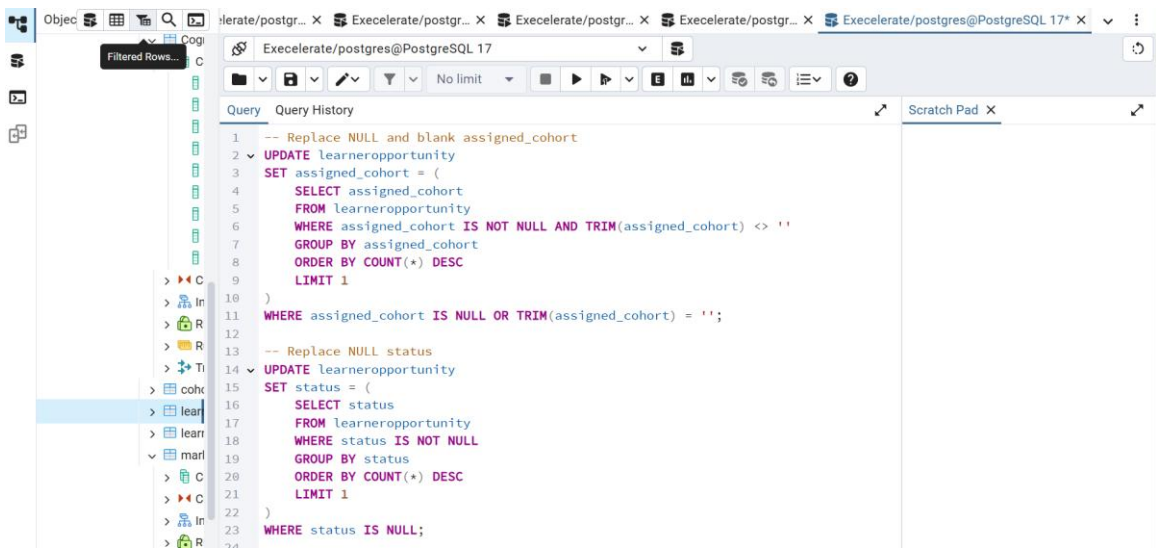


## Trend Visuals Using Power BI

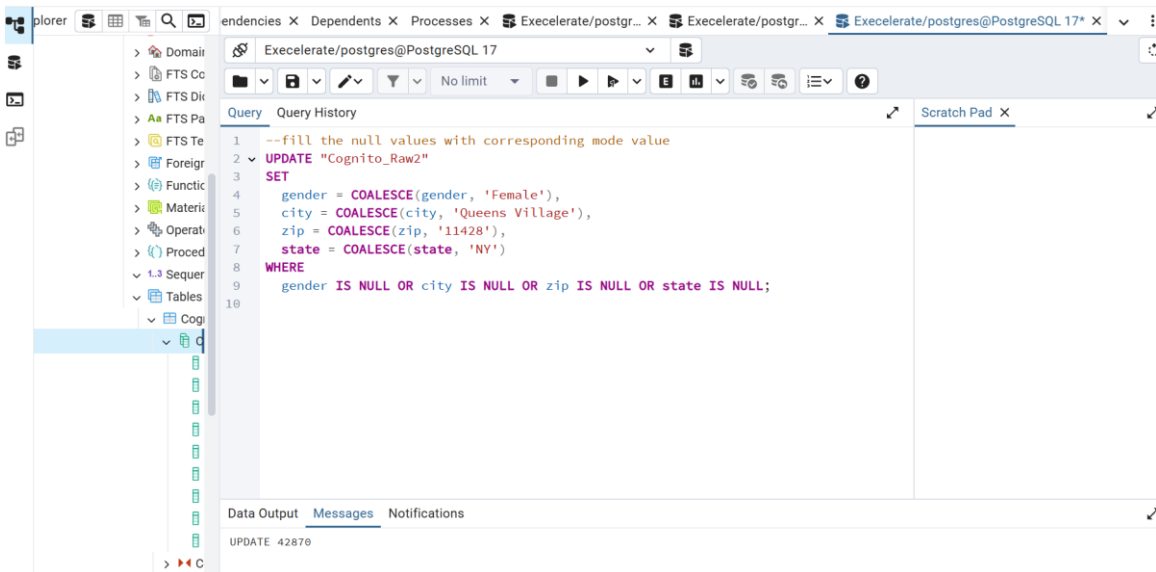


## Summarize Data Sample:





```
25 SELECT
26     q1,
27     median_size,
28     q3,
29     (q3 - q1) AS iqr,
30     (q1 - 1.5 * (q3 - q1)) AS lower_bound,
31     (q3 + 1.5 * (q3 - q1)) AS upper_bound,
32     min_size,
33     max_size,
34     avg_size,
35     outlier_count,
36     mode_cte.size AS mode_value
37 FROM size_stats, outliers, mode_cte;
```





File Object Tools Edit View Window Help

Object SQL Statistics Dependencies Dependents Processes Execelerate/postgr... Execelerate/postgres@PostgreSQL 17\*

Query Query History

```
1 --count the no. of duplicate records
2 SELECT COUNT(*) AS total_duplicate_records
3 FROM (
4     SELECT "user_id", "email", "gender", "UserCreateDate", "UserLastModifiedDate", "birthdate", "city", "zip",
5           "state",
6           COUNT(*)
7     AS duplicate_count
8     FROM "Cognito_Raw2"
9     GROUP BY "user_id", "email", "gender", "UserCreateDate", "UserLastModifiedDate", "birthdate", "city", "zip",
10            "state"
11     HAVING COUNT(*) > 1
12 ) AS duplicates;
```

Data Output Messages Notifications

Showing rows: 1 to 1 Page No: 1 of 1

total_duplicate_records
0

Object Explorer

- Cognito\_Raw2
  - Columns (9)
    - user\_id
    - email
    - gender
    - UserCreateDate
    - UserLastModifiedDate
    - birthdate
    - city
    - zip
    - state
  - Constraints
  - Indexes
  - RLS Policies
  - Rules
  - Triggers
- cohortraw
- learneropportunity
- learners
- marketing\_campaigns
  - Columns
  - Constraints
  - Indexes
  - RLS Policies
  - Rules

Object Explorer

- Cognito\_Raw2
  - Columns (9)
    - user\_id
    - email
    - gender
    - UserCreateDate
    - UserLastModifiedDate
    - birthdate
    - city
    - zip
    - state
  - Constraints
  - Indexes
  - RLS Policies
  - Rules
  - Triggers
- cohortraw
- learneropportunity
- learners
- marketing\_campaigns
  - Columns
  - Constraints
  - Indexes
  - RLS Policies
  - Rules

Execelerate/postgres@PostgreSQL 17

Query Query History Scratch Pad

```
1 --count the no. of null values in each column
2 SELECT
3     COUNT(*) FILTER (WHERE enrollment_id IS NULL) AS null_enrollment_id,
4     COUNT(*) FILTER (WHERE learner_id IS NULL) AS null_learner_id,
5     COUNT(*) FILTER (WHERE assigned_cohort IS NULL OR TRIM(assigned_cohort) = '') AS null_assigned_cohort,
6     COUNT(*) FILTER (WHERE apply_date IS NULL) AS null_apply_date,
7     COUNT(*) FILTER (WHERE status IS NULL) AS null_status
8 FROM learneropportunity;
```

Data Output Messages Notifications

Showing rows: 1 to 1 Page No: 1 of 1

null_enrollment_id	null_learner_id	null_or_blank_assigned_cohort	null_apply_date	null_status
0	0	13318	188	186