# Project Report: Automated Feed Generation, Duplicate Handling, and Comparison in PostgreSQL

**Submitted By Dhruv Ahuja**

**Date 04-09-25**

**Database PostgreSQL**

## 1. Project Objective

The goal of this project was to design and implement an automated data processing system in PostgreSQL that can:  
1. Generate test feeds (tables) dynamically with configurable rows and columns.  
2. Identify and store duplicate records from feeds.  
3. Remove duplicates and keep unique records.  
4. Verify that duplicates are removed successfully.  
5. Compare data between different feeds and store mismatches or missing records.  
6. Run automated tests to validate the above steps.

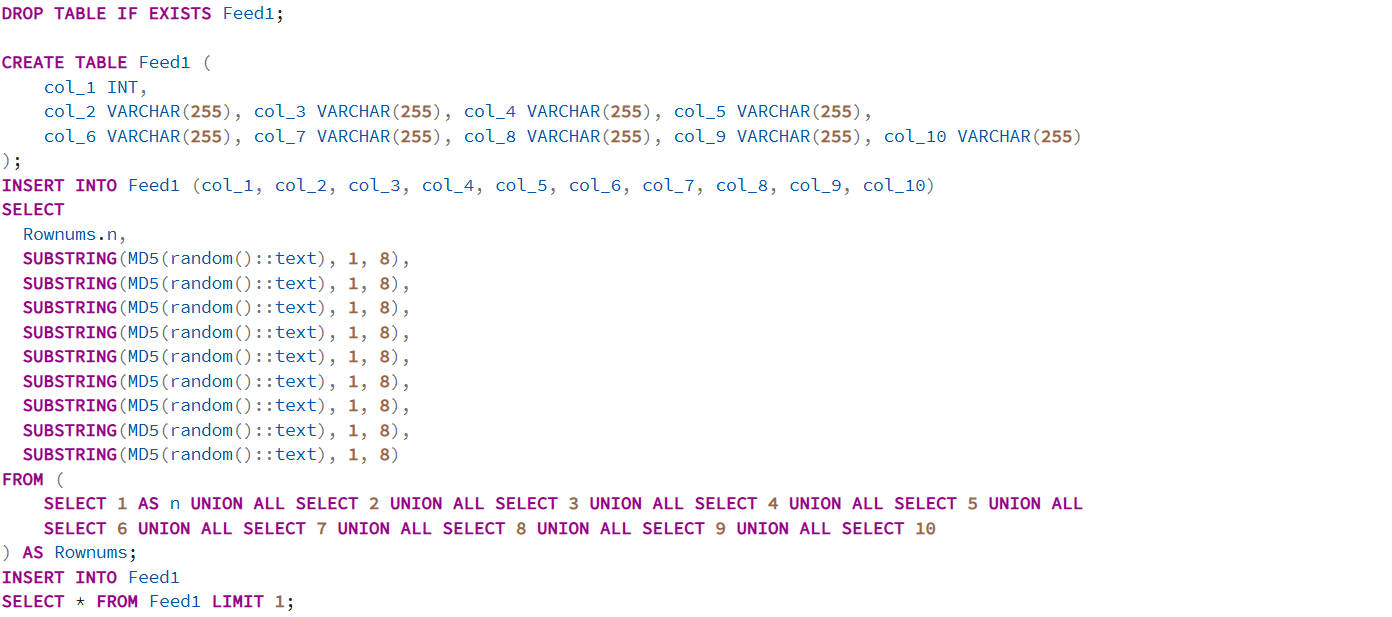
## 2. Dataset Details

- Data is generated automatically using random strings from MD5(RANDOM()).  
- Each feed (Feed1, Feed2, etc.) can have variable numbers of rows and columns.  
- Example: Feed1 had 10 columns × 10 rows, while Feed2 had 15 columns × 15 rows.  
- An extra duplicate row is inserted deliberately for testing duplicate handling.

## 3. REQ 1 – Generate Feeds

A stored procedure `generate\_feed` was created. It dynamically builds tables with random rows and columns.  
Example: `CALL generate\_feed('Feed1', 10, 10);` created Feed1 with 10 columns and 10 rows.

## SQL Script:



## Output REQ 1(Feed1)

## REQ 2 – Automated Task using Store Procedures

A table `duplicates` was created. The procedure `find\_and\_store\_duplicates` finds duplicate rows and stores them in JSON format.

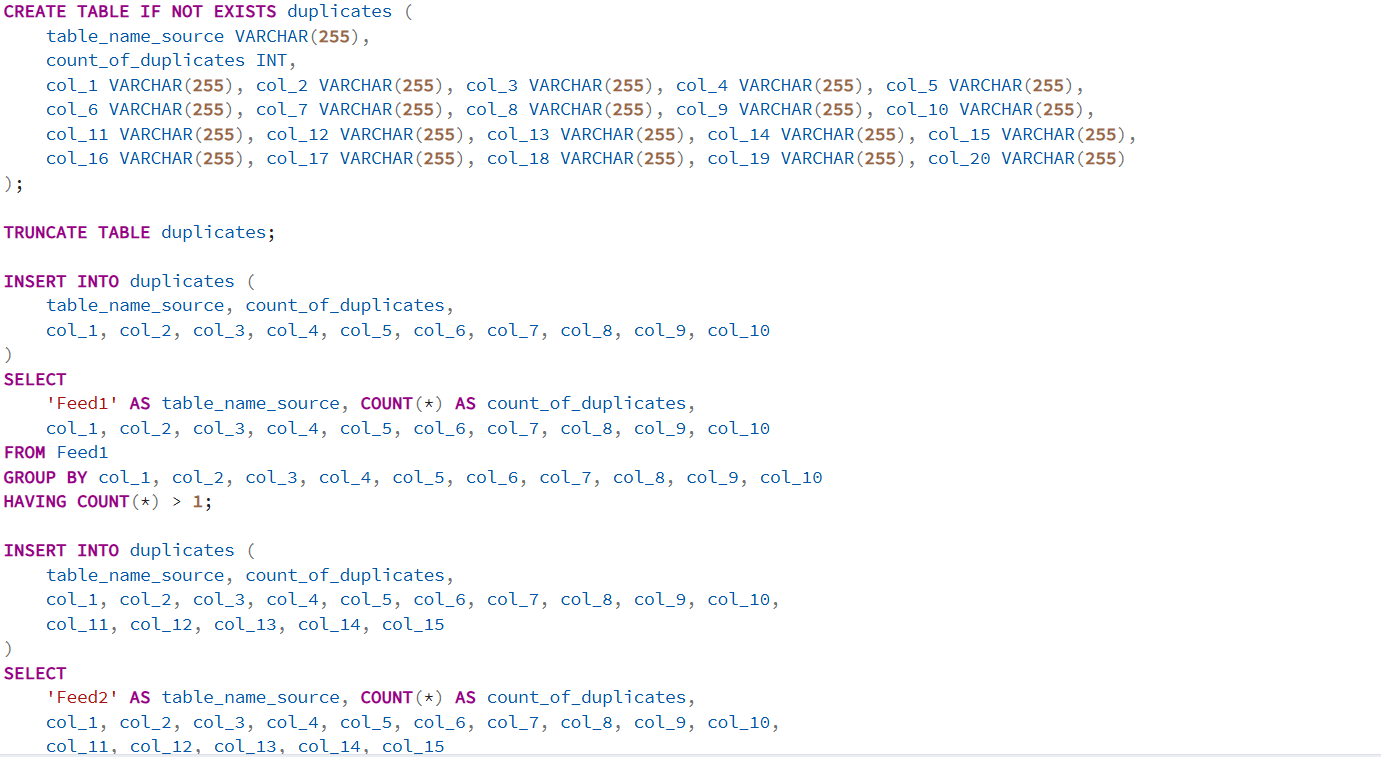
## SQL Query



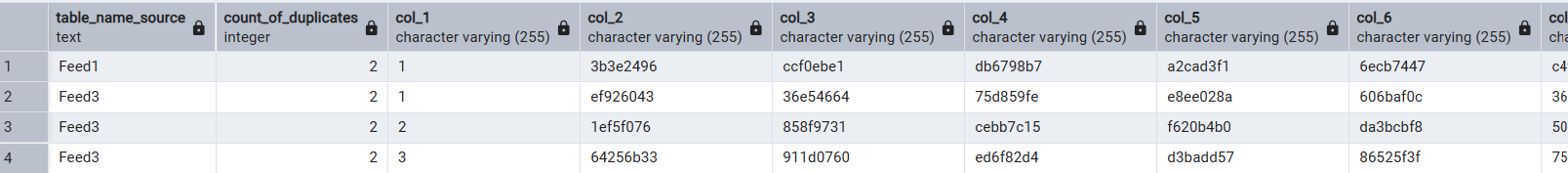
## Step 3&4 – Identify and Store Duplicates

A procedure `replace\_duplicates\_with\_unique` removes duplicate rows while keeping distinct ones.  
Example: Feed1 reduced from 11 rows → 10 rows.

## SQL query



## Output



## REQ 5&6 – Remove Duplicates and Verify

A procedure `verify\_no\_duplicates` checks the feed after cleanup.  
Result: Feed1 had 0 duplicate groups remaining.

SQL Query

A screenshot of a computer

AI-generated content may be incorrect.

## Output

A close up of a computer screen

AI-generated content may be incorrect.

## REQ 7 – Compare Feeds

A table `comparison\_results` was created. The procedure `compare\_feeds` compares two feeds row by row.  
Differences are logged in JSON with statuses: 'In source only' or 'In target only'.  
Example: Comparing Feed2 with Feed1 generated multiple mismatches.



## Output

A screenshot of a computer

AI-generated content may be incorrect.

## REQ 8&9 – Automated Testing

**Objective:** To create a comprehensive test plan covering all functionalities and to automate the execution of these tests using a SQL stored procedure.

### **Manual Test Plan**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Test Case Description** | **Expected Result** | **Test Steps** |
| **TC-01** | **Generate Feed-1** | A table named Feed1 is created with 10 columns and 11 rows (10 unique + 1 duplicate). | 1. Execute CALL generate\_feed('Feed1', 10, 10);  2. Verify the table Feed1 exists  3. Check SELECT COUNT(\*) FROM Feed1; returns 11  4. Check SELECT COUNT(\*) FROM INFORMATION\_SCHEMA.COLUMNS WHERE table\_name = 'Feed1'; returns 10. |
| **TC-02** | **Generate Feed-2** | A table named Feed2 is created with 15 columns and 16 rows. | 1. Execute CALL generate\_feed('Feed2', 15, 15);  2. Verify the table Feed2 exists.  3. Check row and column counts. |
| **TC-03** | **Generate Feed-3** | A table named Feed3 is created with 20 columns and 21 rows. | 1. Execute CALL generate\_feed('Feed3', 20, 20);  2. Verify the table Feed3 exists.  3. Check row and column counts. |
| **TC-04** | **Identify Duplicates in Feed-1** | The duplicates table should contain one record from Feed1 with a count\_of\_duplicates of 2. | 1. TRUNCATE TABLE duplicates;  2. CALL find\_and\_store\_duplicates('Feed1');  3. SELECT COUNT(\*) FROM duplicates WHERE table\_name\_source = 'Feed1'; should be 1. |
| **TC-05** | **Replace Duplicates in Feed-1** | The Feed1 table should contain only 10 unique rows. | 1. CALL replace\_duplicates\_with\_unique('Feed1');. 2. SELECT COUNT(\*) FROM Feed1; should be 10. |
| **TC-06** | **Verify No Duplicates in Feed-1** | The verification script should report that no duplicates are found. | 1. CALL verify\_no\_duplicates('Feed1');  2. The result message should indicate success. |
| **TC-07** | **Compare Feed-2 to Feed-1** | The comparison\_results table should contain records indicating which rows are unique to Feed2 and which are unique to Feed1. | 1. TRUNCATE TABLE comparison\_results;. 2. CALL compare\_feeds('Feed2', 'Feed1');  3. SELECT \* FROM comparison\_results; and manually verify the logic. |
| **TC-08** | **Generate Feed with Zero Rows** | A table should be created with the specified columns but 0 rows. | 1. CALL generate\_feed('Feed\_Zero', 5, 0);  2. SELECT COUNT(\*) FROM Feed\_Zero; should return 0. |
| **TC-09** | **Duplicate Check on Clean Table** | The find\_and\_store\_duplicates procedure should not add any records to the duplicates table. | 1. CALL replace\_duplicates\_with\_unique('Feed2');. 2. TRUNCATE TABLE duplicates;. 3. CALL find\_and\_store\_duplicates('Feed2');. 4. SELECT COUNT(\*) FROM duplicates; should be 0. |

## Query

A computer screen shot of text

AI-generated content may be incorrect.

## Output

A screenshot of a computer

AI-generated content may be incorrect.